TESIS DOCTORAL

2015

DISEÑO E IMPLEMENTACIÓN DE UN NUEVO CLASIFICADOR DE PRÉSTAMOS BANCARIOS A TRAVÉS DE LA MINERÍA DE DATOS

MAURICIO BELTRÁN PASCUAL
LICENCIADO EN CIENCIAS ECONÓMICAS Y EMPRESARIALES

DEPARTAMENTO DE ECONOMÍA APLICADA Y ESTADÍSTICA
FACULTAD DE CIENCIAS ECONÓMICAS Y EMPRESARIALES
UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

DIRECTOR: D. ÁNGEL MUÑOZ ALAMILLOS
CODIRECTOR: D. JUAN ANTONIO VICENTE VÍRSEDA

.

DEPARTAMENTO DE ECONOMÍA APLICADA Y ESTADÍSTICA
FACULTAD DE CIENCIAS ECONÓMICAS Y EMPRESARIALES
UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

DISEÑO E IMPLEMENTACIÓN DE UN NUEVO CLASIFICADOR DE PRÉSTAMOS BANCARIOS A TRAVÉS DE LA MINERÍA DE DATOS

Tesis presentada por:
MAURICIO BELTRÁN PASCUAL
LICENCIADO EN CIENCIAS ECONÓMICAS Y EMPRESARIALES

DIRECTOR: D. ÁNGEL MUÑOZ ALAMILLOS
CODIRECTOR: D. JUAN ANTONIO VICENTE VÍRSEDA

Agradecimientos

A mi madre, a mis hijos y a Maite porque son ellos los que me han enseñado el valor del trabajo, del servicio y del amor.

También expreso mi gratitud a mis dos directores de esta tesis por sus extraordinarios consejos para llevarla a cabo. A Julián Santos porque en todo momento confió en mí y en la minería de datos.

Y no me olvido de los maestros y compañeros con los que compartí tantos conocimientos en la Universidad de Baleares, donde hace ya casi una década empecé a dar mis primeros pasos en Data Mining. Os expreso mi gratitud, a vosotros: Juan José Montaño, Rafael Jiménez y Alfonso Palmer por estar ahí presentes siempre que os he necesitado.

Mi gratitud más sincera a mis actuales maestros, Diego García Morate incansable defensor de las virtudes del programa WEKA y experto en tantos lenguajes de programación y, ¡cómo no! a Francisco Javier Martínez de Pisón Ascacíbar y a su grupo EDMANS de la Universidad de La Rioja, por compartir tantos programas, modelos y algoritmos.

A mi amigo y maestro Francisco Parra por tantas sugerencias y por tantos años aprendiendo juntos y compartiendo lugar de trabajo en la Junta de Castilla y León.

A los técnicos de Caja Rioja por ayudarme a entender la base de datos aportada y, por último, a mis dos últimos jefes: a Francisco Rojas que confió en mí y que a través de tantas conversaciones aportó muchos conocimientos a mi vida, y a mi actual jefe, Jesús María Rodríguez, por sus excelentes cursos de R en los que tanto he aprendido.

Y, por supuesto, a todos los que están o han estado ahí y que han contribuido a mi formación en las técnicas de la minería de datos y, especialmente, en las técnicas de clasificación estadística, a todos desde el corazón, mi más sincero agradecimiento.

Cuando emprendas tu viaje hacia Ítaca
debes rogar que el viaje sea largo,
lleno de peripecias, lleno de experiencias.
No has de temer ni a los lestrigones ni a los cíclopes,
ni a la cólera del airado Poseidón.
Nunca tales monstruos hallarás en tu ruta
si tu pensamiento es elevado, si una exquisita
emoción penetra en tu alma y en tu cuerpo.
Los lestrigones y los cíclopes
Y el feroz Poseidón no podrán encontrarte
si tú no los llevas ya dentro, en tu alma,
si tu alma no los conjura ante ti.
Debes rogar que el viaje sea largo,
que sean muchos los días de verano;
que te vean arribar con gozo, alegremente,
a puertos que tú antes ignorabas.
Que puedas detenerte en los mercados de Fenicia,
y comprar unas bellas mercancías:
madreperlas, coral, ébano, y ámbar,
y perfumes placenteros de mil clases.
Acude a muchas ciudades del Egipto
para aprender, y aprender de quienes saben.
Conserva siempre en tu alma la idea de Ítaca:
llegar allí, he aquí tu destino.
Mas no hagas con prisas tu camino;
mejor será que dure muchos años,
y que llegues, ya viejo, a la pequeña isla,
rico de cuanto habrás ganado en el camino.
No has de esperar que Ítaca te enriquezca:
Ítaca te ha concedido ya un hermoso viaje.
Sin ella, jamás habrías partido;
mas no tiene otra cosa que ofrecerte.
Y si la encuentras pobre, Ítaca no te ha engañado.
Y siendo ya tan viejo, con tanta experiencia,
sin duda sabrás ya qué significan las Ítacas.

Kavafis

ÍNDICE

Lista de figuras, gráficos y recuadros.

Lista de tablas.

CAPÍTULO 1

PLANTEAMIENTO DE LA TESIS DOCTORAL

1. Planteamiento de la tesis doctoral.

1.1. Introducción.

El sector bancario y en general toda la industria bancaria es, sin lugar a dudas, uno de los principales actores de la economía. La función de intermediación bancaria que realizan las instituciones financieras, entre otras actividades, la llevan a cabo a través de la inversión crediticia. Al conceder créditos, estas entidades están asumiendo riesgos y, si se quiere generar rentabilidad, tienen que gestionar adecuadamente estos riesgos.

Es obvia la necesidad de comprender y, por supuesto, de administrar los diferentes tipos de riesgo que surgen de la variabilidad de los diferentes resultados financieros. En Jorion (2000) se define el riesgo como la volatilidad de los resultados esperados, generalmente el valor de los activos o pasivos de interés. Atendiendo al tipo de factores que lo generan, podemos encontrar cuatro grandes grupos: riesgo de mercado, riesgo de crédito, riesgo de negocio o estratégico y riesgo operacional.

A su vez, en el riesgo de crédito podemos identificar cuatro componentes: riesgo de default o de impago, riesgo de mercado, riesgo de liquidez y riesgo país.

Las situaciones en las que los seres humanos toman decisiones se pueden clasificar según el conocimiento y control que se tenga sobre las variables que intervienen o influencian el problema en tres categorías: certeza, riesgo (se conoce el problema y, se conocen las posibles soluciones y, aunque no se conocen con certeza los resultados que pueden arrojar, sí la probabilidad de que ocurra cada resultado) e incertidumbre (se posee información deficiente para tomar la decisión, no se tienen ningún control sobre la situación, no se conoce como puede variar o la interacción de la variables del problema y, aunque se pueden plantear diferentes alternativas de solución, no se le puede asignar probabilidad a los resultados que arrojen[1]). En la Teoría de la Decisión, suele además clasificarse la incertidumbre como estructurada (no se sabe que puede pasar entre diferentes alternativas, pero sí se conoce que puede ocurrir entre varias posibilidades) y no estructurada (no se sabe que puede ocurrir ni las probabilidades para las posibles soluciones).

[1] En 1921 se publicaron los trabajos de Keynes y Knigth (“A Treatise on Probability”. J. M. Keynes. Cambridge University) y Knight (“Risk, Uncertainty, and Profit”. Boston, MA), que distinguieron con nitidez los conceptos de riesgo, susceptible de medición al disponer de una distribución de probabilidad, y de incertidumbre, cuando no se puede asignar probabilidad a los sucesos)

El paso de situaciones de incertidumbre a situaciones de riesgo, es decir, la cuantificación de la probabilidad de que ocurra una determinada solución, es de vital importancia en la toma de decisiones económicas. En casos como el que nos ocupa, entraña la diferencia entre el éxito o el fracaso de la empresa, ya que la principal actividad de una entidad bancaria es dar créditos a clientes y sí estos no son devueltos la quiebra de dicha entidad es inminente; por ello, la disponibilidad de un buen mecanismo que aventure la probabilidad de que un cliente devuelva un crédito es de capital interés para una entidad financiera; este mecanismo debe ser además de acceso relativamente sencillo (muchos puntos de venta o clasificación dirigidos por personal no especialmente cualificado), sin perjuicio de que incorpore módulos de mayor complejidad con acceso a los centros de dirección o puntos en los que se tomen las últimas o más importantes decisiones.

Caouette *et al.* (1998) se afirma que “El próximo gran reto de los mercados financieros es el desarrollo de nuevos métodos y técnicas para valorar el riesgo de crédito”.

Podemos definir el credit scoring como los métodos estadísticos utilizados para clasificar a los solicitantes de crédito, sean o no clientes de la entidad evaluadora, entre las clases de riesgo bueno o malo, Hand y Henley (1997)). El credit scoring es un sistema o un método que a través de predicciones mide el riesgo inherente al mismo. Estos modelos llevan utilizándose varias décadas. Otros nombres con los que se conoce al credit scoring son: calificación de riesgo de insolvencia o morosidad.

Otro autor, Bessis (2002) define el riesgo de crédito como aquellas pérdidas asociadas al evento fallido del prestatario o al evento del deterioro de la calidad crediticia.

El credit scoring se erige como una metodología ya plenamente aceptada por el Comité de Basilea para la supervisión bancaria y también por los sistemas financieros europeos y norteamericano y a través de un sistema de rating interno se clasifica a los clientes de la institución financiera como clientes buenos o malos. A la hora de valorar el riesgo encontramos que los determinantes del mismo son: la probabilidad de incumplimiento (default), la Exposición y la Severidad o tasa de recuperación.

Todas las grandes corporaciones financieras poseen modelos de credit scoring, Mester (1997) y esta metodología es aplicada en la medición del riesgo de crédito para préstamos personales, hipotecarios, de consumo y fundamentalmente en préstamos a empresas.

El credit scoring se clasifica en dos grandes grupos, proactivo y reactivo:

- El scoring proactivo utiliza la información comportamental de clientes ya vinculados a la entidad para cubrir sus necesidades adicionales sin asumir riesgos mayores. La información procedente de fuentes internas se utiliza para anticiparse a las necesidades de sus clientes.
- En el scoring reactivo se utiliza el comportamiento pasado para predecir el futuro. Se explota la información de la persona existente en el sistema, tanto negativa como positivamente. Esta forma de scoring permite sintetizar en un indicador único y muy eficiente el historial de crédito de un cliente. Estos métodos sirven tanto para evaluar a clientes como a no clientes de la entidad.

En términos de lenguaje, se suele hablar de scoring cuando la evaluación se refiere a particulares, mientras que se acuña el término de rating cuando el scoring se realiza a empresas. Cuando se evalúa a individuos, se utiliza la información interna del banco más otra información generalmente socioeconómica, mientras que cuando se habla de empresas, los métodos estadísticos utilizan ratios económicos y financieros sacados del Balance y de la Cuenta de Resultados de las empresas.

Es claro que desde el principio de la crisis económica, que algunos sitúan su inicio con la caída del cuarto banco de inversión más grande de Estados Unidos, Lehman Brothers, los excesivos riesgos económicos de la banca han quedado al descubierto. El origen de la crisis financiera actual se puede fechar un año antes, en la primavera del 2007, cuando otro banco de inversión, Bear Stearns, anunciaba que algunas carteras que apostaban en deuda hipotecaria no eran rentables. Un año después este banco tuvo que ser rescatado por JP Morgan Chase con la ayuda del Tesoro norteamericano, cosa que no sucedió con Lehman. Estos acontecimientos en los mercados hipotecarios estadounidenses crearon un efecto contagio en diversos mercados a nivel mundial.

La crisis actual se desencadenó después de un prolongado periodo de bonanza económica generalizada a nivel global. Las causas de esta crisis, ni surgieron súbitamente ni podemos afirmar que se hayan corregido aún. El continuo ritmo de crecimiento de la deuda acumulada por hogares, empresas y también por las administraciones en la década pasada, que era muy elevada, ha resultado ser insostenible.

Numerosos analistas e instituciones consideran que la actual crisis se originó por la acumulación de un conjunto de fallos en el sistema financiero: fundamentalmente en una infravaloración del riesgo que asumieron a través del apalancamiento de sus

posiciones y una sobreestimación de los diferentes agentes por transferir el riesgo; también en el papel que jugaron las agencias de calificación crediticia, en la estructura de gobiernos de muchas instituciones financieras, en la normativa y en la supervisión, principalmente. Otra razón que se esgrime es el grado de complejidad que alcanzaron algunos instrumentos financieros que hicieron particularmente complejo el análisis y la gestión del riesgo a lo que hay que añadir la falta de transparencia en determinados segmentos de la banca.

Una consecuencia clara de esta crisis ha sido la restructuración bancaria. Citando a Carbó (2011) "El proceso de reestructuración bancaria en España se ha visto caracterizado por el gran énfasis en los problemas de liquidez en Europa al comienzo de la crisis, el reconocimiento (algo tardío) de los problemas de solvencia y la articulación de un proceso ordenado para reformar el sector, recapitalizarlo y adaptarlo al entorno regulatorio post-crisis". Esta restructuración ha contado con la ayuda de 100.000 millones en 2012 de los que hasta el momento se han utilizado 61.366 millones.

Unido al manifiesto proceso de deterioro del crédito, el volumen de préstamos dudosos ha ido aumentando en el periodo de crisis económica, alcanzando los 176.420 millones en junio de 2013 lo que implica que la tasa de morosidad, la proporción del crédito que se considera dudoso, alcanzó en ese mes el 11,6%.

Anteriormente al inicio de la crisis ya se habían realizado dos importantes eventos relacionados con la supervisión y la regulación bancaria: Basilea I y II. El acuerdo denominado Convergencia internacional de medición de capital y estándares de capital conocidos como Acuerdo del Comité de Basilea son recomendaciones sobre regulación y legislación bancaria que se emiten a través del Comité de supervisión bancaria.

El primer acuerdo fue tomado en el año 1988, en una reunión de los gobernadores de los bancos centrales de 13 países, el G-10 y otros tres países más, entre ellos España, donde se establecía una definición de capital regulatorio que debía de ser suficiente para hacer frente a los riesgos de crédito, mercado y tipo de cambio. El principal riesgo, el de crédito, se calculaba agrupando las exposiciones de riesgo en cinco categorías y asignando una ponderación a cada una de ellas, siendo la suma de los riesgos ponderados la que formaba los activos de riesgo.

El Comité de Basilea adoptó un estándar internacional de adecuación de capital que limitaba el apalancamiento financiero. A través de esta medida se requirió a los bancos

mantener capital suficiente para proteger a los depositantes de eventuales pérdidas y soportar el crecimiento de activos.

Basilea I estableció que el estándar de adecuación de capital, denominado Coeficiente de Adecuación Patrimonial (CAP) fuera equivalente como mínimo al 8% de los activos ponderados por riesgo. Se establecieron cinco (5) categorías de activos, cuya ponderación fue determinada en función al riesgo de crédito que conlleva cada activo: desde 0% para los activos libres de riesgo (como puede ser el efectivo y los créditos al gobierno) hasta 100% para aquellos activos con mayor riesgo (préstamos a empresas privadas y de otros activos).

Los acuerdos de Basilea I jugaron un papel notable en el fortalecimiento de los sistemas bancarios: sus recomendaciones entraron en vigor en más de 130 países.

El principal problema que se detectó es que en Basilea I no se establecía una dimensión esencial, la calidad crediticia y, por tanto, la probabilidad de incumplimiento de los distintos prestatarios. Como se ha comentado, lo que se consideraba es que los diferentes créditos tenían la misma probabilidad de incumplimiento.

Debido a las limitaciones en su definición y los cambios del sector bancario se publican, en junio de 2004, los acuerdos de Basilea II con el objetivo de establecer los requerimientos de capital necesarios para asegurar la protección de las entidades a los riesgos financieros y operativos.

El comité de Basilea II propuso nuevas recomendaciones que se apoyaron en tres pilares: el cálculo de los requisitos mínimos de capital, el proceso de supervisión de la gestión de los fondos propios y la disciplina de mercado.

En cuanto al riesgo de crédito, el comité de Basilea II introduce un enfoque más sensible al riesgo para el cálculo de requerimientos de capital, con métodos que van desde los más simples hasta los más complejos y cuya aplicación depende del grado de desarrollo de las actividades de las entidades financieras y de la infraestructura del mercado financiero. En el caso del riesgo de crédito, dentro del pilar 1 referido a requerimientos de capital, Basilea II revisó el cálculo de los activos ponderados por riesgo, y estableció básicamente dos métodos: Una alternativa, el método Estándar, sería la medición de dicho riesgo a partir de evaluaciones externas del crédito y, otra alternativa, el método basado en Calificaciones Internas, que necesitará la aprobación explícita del supervisor del banco y permitiría a los bancos utilizar sus propios sistemas de calificación interna para el riesgo de crédito.

El método estándar es el método más simple, cuya base ya se encontraba en Basilea I. La novedad en Basilea II es que este método reconoce a las calificaciones de crédito externas y permite utilizarlas para asignar un ponderador a las partidas del activo y operaciones fuera de balance. A diferencia de Basilea I, este método amplía a once las categorías en las que puede ser clasificado un activo y admite la ponderación de activos diferenciada en función de la calificación obtenida.

El método basado en calificaciones internas (*Internal Ratings-Based:* IRB) permite que el requerimiento de capital pueda basarse en calificaciones internas y estimaciones propias de los factores de riesgo. Adicionalmente, incluye técnicas de mitigación de riesgos y operaciones de titularización de activos.

En el método IRB, la entidad debe ser capaz de calcular sus pérdidas esperadas e inesperadas. El requerimiento de capital está orientado a cubrir las pérdidas inesperadas.

Para efectuar el cálculo, el modelo requiere estimar los siguientes factores de riesgo:

- Probabilidad de incumplimiento (Probability of Default: PD)
- Pérdida en caso de incumplimiento (Loss Given Default: LGD)
- Exposición al momento de incumplimiento (Exposure at Default: EAD)
- Vencimiento efectivo (*Maturity:* M).

El método IRB presenta dos variantes:

- IRB básico, en el cual las entidades están autorizadas a calcular sus estimaciones de la PD. Los parámetros de los demás componentes de riesgo son dados por el supervisor.
- IRB avanzado, cuya particularidad radica en que las entidades están facultadas a calcular los parámetros de los cuatro componentes (PD, LGD, EAD y M).

Para el uso de los métodos IRB, las entidades financieras deben contar con autorización del supervisor y cumplir unos requisitos mínimos.

Independientemente de la aplicación de Basilea II, desde la perspectiva del supervisor, es importante que las entidades financieras realicen una adecuada gestión del riesgo de crédito. En este sentido, uno de los 29 principios básicos para una supervisión bancaria efectiva, publicado por el Comité de Basilea, se refiere al riesgo de crédito (principio 17) el cual señala que: “El supervisor verifica que los bancos disponen de un

adecuado proceso de gestión del riesgo de crédito que tiene en cuenta su apetito por el riesgo, su perfil de riesgo y la situación macroeconómica y de los mercados. Esto incluye políticas y procesos prudentes para identificar, cuantificar, evaluar, vigilar, informar y controlar o mitigar el riesgo de crédito (incluido el riesgo de crédito de contraparte) en el momento oportuno. El ciclo de vida completo del crédito deberá quedar contemplado, incluida la concesión del crédito, la evaluación del crédito y la gestión continua de las carteras de préstamos e inversiones".

El riesgo de crédito puede provenir de las siguientes actividades: exposiciones dentro y fuera de balance, incluidos préstamos y anticipos, inversiones, préstamos interbancarios, operaciones con derivados, operaciones de financiación con valores y actividades de negociación. El riesgo de crédito de contraparte incluye las exposiciones al riesgo de crédito procedentes de derivados negociados en mercados no organizados (OTC) y de otros instrumentos financieros.

Tras observar y padecer las severas consecuencias de la crisis económica se desarrolla Basilea III: Marco Internacional para la medición, normalización y seguimiento del riesgo de liquidez como una serie de iniciativas promovidas por el Foro de Estabilidad Financiera y el G-20 cuyas conclusiones fueron publicadas a partir del mes de diciembre de 2010.

Las reformas diseñadas por el Comité de Basilea están basadas en las conclusiones extraídas de la reciente crisis financiera al mismo tiempo que se intenta mejorar el riesgo, el buen gobierno de las entidades financieras, reforzar su transparencia y la divulgación de la información.

En cuanto a la gestión de los riesgos está claro que existió una infravaloración del mismo que estaban asumiendo las instituciones financieras a través del apalancamiento de sus posiciones, a la vez que se sobrestimaba la capacidad de los agentes para transferir ese riesgo. El Comité de Basilea afirma que: "una de las lecciones claves es que hay que reforzar la cobertura de riesgo en el marco de capital dado que uno de los principales factores desestabilizadores fue la incapacidad de captar correctamente los mayores riesgos dentro y fuera del balance, así como las exposiciones relacionadas con sus derivados."

Estas pérdidas, según señaló el Comité, pueden desestabilizar la banca y, con ello, generar o exacerbar una desaceleración de la economía real, lo cual a su vez podría desestabilizar aún más el sector bancario. Estos vínculos destacan la importancia de que el sector bancario acumule capital defensivo cuando el crédito crece de forma

excesiva. Además, estas defensas también ayudarían a moderar la propia expansión crediticia. El documento BCBS 2011) señala que “El Comité ha examinado varias medidas adicionales que los supervisores podrían adoptar para equilibrar mejor, si se considerase necesario, la sensibilidad al riesgo y la estabilidad de los requerimientos de capital. Entre ellas, se incluye la iniciativa del Comité de Supervisores Bancarios Europeos (CEBS) de utilizar el proceso del Segundo Pilar para remediar el hecho de que, cuando las condiciones crediticias son favorables, se comprimen las estimaciones de probabilidad de incumplimiento (PD) en los requerimientos de capital calculados con el método basado en calificaciones internas (IRB), consistiendo su propuesta en tomar para las carteras del banco valores de PD estimados en condiciones recesivas”.

Esta tesis doctoral se enmarca dentro de los requerimientos de Basilea II y también de Basilea III en cuanto que lo realmente necesario es contar con una estimación del incumplimiento basada en modelos (BCBS, 2006, III, 444), “Las calificaciones internas y las estimaciones de incumplimiento y pérdida deberán ser esenciales para los bancos que utilicen el método IRB en cuanto a la aprobación de créditos, gestión de riesgos, asignaciones internas de capital y gobierno corporativo. No serán aceptables los sistemas de calificación ni las estimaciones cuyo diseño y aplicación tengan como único propósito la admisión en el método IRB y cuya utilización consista exclusivamente en el suministro de argumentos a las funciones IRB. Se entiende que los bancos no siempre emplearán en el método IRB las mismas estimaciones que utilizan en el resto de sus funciones internas. Por ejemplo, los modelos de valoración de activos probablemente utilicen las PD y LGD pertinentes a la vida útil del activo. En el caso de producirse tales discrepancias, el banco deberá documentarlas y demostrar su raciocinio ante el supervisor”.

1.2. Objetivos de la tesis doctoral.

1.2.1 Objetivo general.

El objetivo fundamental de esta tesis es disponer de un buen método estadístico que nos ayude a tomar decisiones más correctas a la hora de conceder o no un préstamo, para así mejorar la eficacia de la gestión de la entidad financiera, siendo de especial interés en una situación como la actual en la que a las entidades financieras se les está exigiendo un mayor análisis del riesgo y una mejora en la eficiencia de su gestión. Así que la verdadera motivación consiste en desarrollar modelos de credit scoring óptimos y mejores a los conocidos, de acuerdo a las exigencias de calcular la probabilidad de default que requieren los modelos de Basilea II y III y que redundará,

sin lugar a dudas, en un mayor beneficio de las instituciones al aplicar estos modelos en el proceso de concesión de créditos.

Los métodos y técnicas que se proponen en esta tesis doctoral aportan estas utilidades, además, se pueden considerar de construcción sencilla, con una semántica clara y tienen un enfoque sólido y elegante. Si bien han presentado tradicionalmente el problema de su elevado coste computacional, el avance tecnológico está contribuyendo a resolver éste de forma rápida y eficaz.

Hay que tener en cuenta que, según diferentes autores y por la investigación llevada a cabo en esta tesis doctoral, ningún método estadístico de clasificación alcanza resultados óptimos con todas las bases de datos.

El propósito de esta investigación se enmarca desde la óptica IRB que promueven los dos últimos acuerdos de Basilea así que esta tesis se centra en la búsqueda de los mejores modelos que nos ayuden a reducir el riesgo de incumplimiento de los créditos otorgados. En la elección entre los múltiples procedimientos estadísticos, a la hora de clasificar a los clientes, se han escogido aquellos modelos que satisfacen los criterios de Basilea II: **calidad explicativa, predictiva y discriminante**.

La necesidad de satisfacer el máximo de requerimientos y la alta precisión de los resultados obtenidos por diversos métodos clasificatorios nos llevan a proponer una solución basada en aquellos que son los más eficientes, por lo que en esta tesis se aborda la forma de construir un clasificador eficaz a través de una metodología adecuada y de la selección de los diferentes algoritmos utilizados en la minería de datos , cuya finalidad es conseguir más precisión entre los modelos paramétricos y no paramétricos empleados en los problemas de credit scoring.

Una importante cuestión es que tienen una significativa ventaja aquellos modelos en los que se puedan incorporar en el proceso de predicción de impago el conocimiento relevante que nos indiquen los expertos bancarios.

Con los datos reales aportados por la Caja de Ahorros de La Rioja se presenta una forma de implementar un clasificador de préstamos bancarios con los clasificadores óptimos analizados en este trabajo a través del lenguaje JAVA y cuyos comandos de programación puedes verse en los anexos. Esa aplicación informática se ofrece a los técnicos de Caja Rioja y a toda la comunidad científica.

Con la información aportada por el cliente que solicita el crédito, aplicada a la base de datos histórica que dispone el banco, el modelo sugiere al gerente una primera decisión sobre la aceptación o no de la petición del cliente (modelo de credit scoring). En este trabajo se propone un sistema de predicción que optimiza la decisión estadística que determina la clase a la que pertenecen las muestras o clientes evaluados; siempre sin olvidar que los modelos de credit scoring ayudan en un primer momento a tomar la decisión de si conceder o no el crédito, e incluso permiten justificar la misma. No obstante, junto a sus resultados, deben considerarse otras dimensiones cualitativas que necesariamente deben complementar la toma de la decisión y que no se pueden estudiar con los modelos matemáticos.

1.2.2. Objetivos específicos.

Antes de llegar a proponer la solución del mejor modelo se han abordado, en esta tesis doctoral, algunas importantes tareas relacionadas con la construcción de modelos estadísticos y con la comparación entre diferentes algoritmos de clasificación.

En la correcta aplicación de la metodología de la minería de datos a los datos aportados por la Caja de Ahorros se han optimizado varias cuestiones de este proceso, por lo que se puede afirmar que se han alcanzado otros objetivos específicos:

1. Como se ha demostrado en diversos trabajos, a la hora de aplicar los métodos de clasificación hemos de tener en cuenta cómo están distribuidas las instancias respecto a la clase. Al no estar balanceadas las clases los clasificadores estarán sesgados a predecir un porcentaje más elevado de la clase más favorecida. Dicho de otro modo, el tamaño de la muestra juega un papel determinante en la bondad de los modelos de clasificación. Cuando el desbalanceo es considerable, descubrir regularidades inherentes a la clase minoritaria se convierte en una tarea ardua y de poca fiabilidad.

   Respecto al vital tema de equilibrar las muestras como trabajo previo, antes de aplicar el modelo de clasificación, se introduce el método de muestreo denominado del cubo, no contemplado hasta ahora en ninguno de los trabajos analizados sobre credit scoring, y que presenta ventajas competitivas sobre otros procedimientos de extracción de muestras. Se reduce la muestra de la clase mayoritaria a través del método del submuestreo equilibrado del Cubo, propuesto por Deville y Tillé (2004). Entre los métodos existentes en la literatura estadística para la selección de submuestras es el denominado del

cubo el único que nos permite seleccionar una muestra equilibrada sobre variables auxiliares con probabilidades de inclusión que pueden ser iguales o no. El método del cubo selecciona únicamente las muestras cuyos estimadores de Horvitz-Thompson son iguales a los totales de las variables auxiliares conocidas.

2. En cuanto a la selección de variables significativas, a la hora de presentar modelos sencillos e interpretables, atendiendo a la lógica y a las recomendaciones del Comité de Basilea, se han utilizado métodos eficientes. La solución que nos parece más óptima y adecuada a este problema, en cuanto al número de variables utilizadas en la aplicación de los modelos y algoritmos de clasificación, es seleccionar los atributos para la clasificación a través de los métodos disponibles en la minería de datos: métodos de filtro o métodos previos y métodos basados en modelo, envolventes o de wrapper. Cuando el modelo utilizado es un red bayesiana se aborda la selección de características a través del manto de Markov. La envolvente de Markov para una variable representa el conjunto de variables de las que depende dicha variable.
3. Otro objetivo de esta tesis es comprobar cómo se comportan los modelos cuando se combinan sus predicciones agregando los modelos individuales. Los multiclasificadores son una excelente forma de integrar la información de diferentes fuentes. Esta combinación de dos o más clasificadores, en general, proporciona estimaciones más robustas y eficientes que cuando se utiliza un único clasificador. También son muy empleados porque resuelven el problema de sobreadaptación (overfitting) y es posible obtener buenos resultados con pocos datos. Esta solución, como se observa, en el capítulo de resultados, es bastante óptima.
4. Otra interesante perspectiva es estudiar la clasificación desde el punto de vista de los costes. incorporando a los métodos de clasificación una matriz de costes, como alternativa al problema del escaso acierto de la clase menos representada de los diferentes métodos de clasificación, cuando las muestras presentan un grave desequilibrio, como son los datos que se utilizan en credit scoring. En esta tesis se utiliza el método del costo-sensitivo (*cost-sensitive*). Este método se basa en la aseveración de que el precio de cometer un error de clasificación es distinto para cada clase. Es evidente que no es lo mismo conceder un crédito y no pagarlo que no concederlo cuando se debería haber ofrecido. El clasificador que se aplica para poder comparar con el resto de los algoritmos es el Metacost, Domingos, (1999). El objetivo de este

procedimiento es reetiquetar cada muestra de entrenamiento por la estimación del riesgo de Bayes. Finalmente, el clasificador se entrena con un método no basado en costes con el conjunto que ya ha sido reetiquetado.

5. Un quinto objetivo, tal y como se observa en el título de la tesis doctoral, es plasmar la implementación de los diferentes métodos de clasificación en una aplicación informática. Entre los diferentes lenguajes existentes me decanto por el JAVA, así que se facilita una aplicación realizada en este lenguaje de programación, que evalúa la petición de crédito y que se adapta a diferentes modelos de forma sencilla.

   Un importante aspecto de esta aplicación informática es poder realizar simulaciones con diferentes modelos, no sólo para ver si se le concede el crédito o no, sino que, además, se acompaña la probabilidad con la que ha sido concedida o denegada la solicitud de crédito. En el anexo se facilita el código fuente para su utilización por parte de la comunidad científica.

1.3. Esquema de la tesis.

Para cumplir los objetivos propuestos, esta tesis se organiza en nueve capítulos.

En el capítulo siguiente, se ha realizado un exhaustivo análisis sobre los trabajos realizados sobre credit scoring. El estudio del arte cubre un amplio abanico de técnicas estadísticas y nos ayuda a situar y dar importancia a esta tesis doctoral.

El tercer capítulo se aborda la metodología general de la minería de datos y se centra en explicar algunas de las técnicas y métodos de la minería de datos que abordan la clasificación y que han formado parte de la investigación de este tema. Se han recogido en este capítulo las principales técnicas estadísticas paramétricas y no paramétricas: árboles de decisión, redes neuronales, algoritmos genéticos, máquinas de vectores soporte, modelos logit y probit, análisis discriminante, redes bayesianas y multiclasificadores.

Se pone un especial énfasis en las peculiaridades metodológicas de los problemas del credit scoring, especialmente en la selección de los atributos más relevantes del conjunto de entrenamiento de la base de datos. En la literatura de selección de variables existen dos métodos generales para escoger las mejores características de la base de datos: métodos de filtro y métodos basados en modelos. En los primeros se filtran los atributos irrelevantes antes de aplicar las técnicas de minería de datos. El criterio que establece las variables óptimas se basa en una medida de calidad que se calcula a partir de los datos mismos. En los métodos basados en modelos, también

conocidos como métodos de envolvente o wrapper, la bondad de la selección de las variables se evalúa a través de un modelo utilizando, lógicamente, un método de validación. Otro aspecto metodológico esencial que se trata ampliamente en este apartado es el desbalanceo de muestras existentes en las clases a predecir y, también se pone especial énfasis en los procedimientos de evaluación de los clasificadores. También se aborda la discretización de las variables cuantitativas dado que algunos algoritmos de clasificación empleados sólo utilizan variables discretas.

Respecto a las técnicas, los árboles de decisión son particiones secuenciales de un conjunto de datos que maximizan las diferencias de la variable dependiente. Nos ofrecen una forma concisa de definir grupos que son consistentes en sus atributos pero que varían en términos de la variable dependiente. Esta herramienta puede emplearse tanto para la resolución de problemas de clasificación como de regresión: árboles de clasificación y árboles de regresión. Mediante esta técnica se representan de forma gráfica un conjunto de reglas sobre las decisiones que se deben de tener en cuenta para asignar un determinado elemento a una clase (valor de salida).

Las redes neuronales tratan de emular el comportamiento cerebral. Una red neuronal puede describirse mediante cuatro conceptos: el tipo de modelo de red neuronal; las unidades de procesamiento que recogen información, la procesan y arrojan un valor; la organización del sistema de nodos para transmitir las señales desde los nodos de entrada a los nodos de salida y, por último, la función de aprendizaje a través de la cual el sistema se retroalimenta.

Se considera una red neuronal la ordenación secuencial de tres tipos básicos de nodos o capas: nodos de entrada, nodos de salida y nodos intermedios (capa oculta o escondida).

Los nodos de entrada se encargan de recibir los valores iniciales de los datos de cada caso para transmitirlos a la red. Los nodos de salida reciben entradas y calculan el valor de salida (no van a otro nodo). En casi todas las redes existe una tercera capa denominada oculta, Este conjunto de nodos utilizados por la red neuronal, junto con la función de activación posibilita a las redes neuronales representar fácilmente las relaciones no lineales, que son muy problemáticas para las técnicas multivariantes.

Cuando se presenta un patrón de entrada X_p : $x_{p1},\ldots,x_{pi},\ldots x_{pN}$ se transmite a la red a través de los pesos w_{ji} desde la capa de entrada a la capa oculta. Las neuronas de esta capa transforman las señales a través de la función de activación proporcionando un valor de salida. Este valor se transmite a su vez a través de los pesos v_{kj} a la capa

de salida donde aplicando de nuevo la función de activación obtenemos un valor de salida.

Los fundamentos teóricos de las máquinas de vectores soporte (Support Vector Machines, SVM) fueron presentados en el año 1992 en la conferencia COLT (Computacional Learning Theory) por Boser, Guyon y Vapnik (1992) y descritos posteriormente en diversos artículos por Cortes y Vapnik [Cortes y Vapnik (1995)]; Vapnik (1998) y (2000)] a partir de los trabajos sobre la teoría del aprendizaje estadístico.

Las máquinas de vectores soporte pertenecen a la familia de los clasificadores lineales dado que inducen hiperplanos o separadores lineales de muy alta dimensionalidad introducidos por funciones núcleo o kernel. Es decir, el enfoque de las SVM adopta un punto de vista no habitual, en vez de reducir la dimensión buscan una dimensión mayor en la cual los puntos puedan separarse linealmente.

Los algoritmos genéticos propuestos por Holland (1975), suponen uno de los enfoques más originales en la minería de datos, se inspiran en el comportamiento natural de la evolución, para ello se codifica cada uno de los casos de prueba como una cadena binaria (que se asemejaría a un gen). Esta cadena se replica o se inhibe en función de su importancia, determinada por una función denominada de ajuste o fitness.

Los algoritmos genéticos son adecuados para obtener buenas aproximaciones en problemas de búsqueda, aprendizaje y optimización, Marczyk, (2004).

De forma esquemática un algoritmo genético es una función matemática que tomando como entrada unos individuos iniciales (población origen) selecciona aquellos ejemplares (también llamados genes) que recombinándose por algún método generarán como resultado la siguiente generación. Esta función se aplicará de forma iterativa hasta verificar alguna condición de parada, bien pueda ser un número máximo de iteraciones o bien la obtención de un individuo que cumpla unas restricciones iniciales.

Se abordan las redes bayesianas que también se conocen en la literatura con otros nombres: redes causales o redes causales probabilísticas, redes de creencia, sistemas probabilísticas, sistemas expertos bayesianos o también como diagramas de influencia. Las redes bayesianas son métodos estadísticos que representan la incertidumbre a través de las relaciones de independencia condicional que se establecen entre ellas, Edwards, (1998). Este tipo de redes codifica la incertidumbre

asociada a cada variable por medio de probabilidades. Siguiendo a Kadie, *et al.* (2001) afirman que una red bayesiana es un conjunto de variables, una estructura gráfica conectada a estas variables y un conjunto de distribuciones de probabilidad.

Estas redes probabilísticas automatizan el proceso de modelización probabilístico utilizando toda la expresividad de los grafos para representar las dependencias y de la teoría de la probabilidad para cuantificar esas relaciones. En esta unión se plasma de forma eficiente tanto el aprendizaje automático como la inferencia con los datos y la información disponible.

Una red bayesiana queda especificada formalmente por una dupla B = (G,Θ) donde G es un grafo dirigido acíclico (GDA) y Θ es el conjunto de distribuciones de probabilidad. Definimos un grafo como un par G= (V, E), donde V es un conjunto finito de vértices nodos o variables y E es un subconjunto del producto cartesiano V x V de pares ordenados de nodos que llamamos enlaces o aristas.

La redes bayesianas tienen la habilidad de codificar la causalidad entre las variables, por lo que han sido muy utili**z**adas en el modelado o en la búsqueda automática de estructuras causales, López *et al.* (2006). La potencia de las redes bayesianas está en su capacidad de codificar las dependencias/independencias relevantes considerando no sólo las dependencias marginales sino también las dependencias condicionales entre conjuntos de variables.

También se presentan brevemente las técnicas estadísticas clásicas más conocidas y utilizadas en la clasificación: modelos Logit, Probit y el análisis discriminante.

Los Multiclasificadores como combinación de modelos representan una excelente forma de conseguir una mayor precisión de las predicciones de nuestros modelos. La combinación de las hipótesis de los multiclasificadores es una manera de integrar la información de diferentes fuentes. Esta combinación de dos o más clasificadores, en general, como ya hemos afirmado proporciona estimaciones más robustas y eficientes que cuando se utiliza un único clasificador. También se utilizan porque resuelven el problema de sobreadaptación (overfitting) y es posible obtener buenos resultados con pocos datos. Son múltiples los estudios que se han realizado con los métodos multiclasificadores, así que podemos conocerlos en la literatura existente con muchos nombres: métodos de ensamble, modelos múltiples, sistemas de múltiples clasificadores, combinación de clasificadores, integración de clasificadores, mezcla de expertos, comité de decisión, fusión de clasificadores de aprendizaje multimodelo.

En el cuarto capítulo se detalla de forma precisa la metodología utilizada en esta tesis doctoral.

El capítulo cinco se aborda el estudio práctico de aplicación de scoring con datos de una Caja de Ahorros de La Rioja. Se presentan los resultados para todos los modelos comentados en esta tesis.

En el sexto capítulo se explica la aplicación informática implementada en JAVA, donde realmente se ve la efectividad de los métodos propuestos en esta tesis doctoral. Se introducen los datos socioeconómicos del cliente del banco o de la persona a la que se quiera evaluar y la aplicación muestra si se le concede el crédito o no, así como la probabilidad con la que se le ha clasificado.

En el séptimo capítulo se relatan las conclusiones, limitaciones y nuevas líneas de investigación relacionadas con los objetivos de esta tesis doctoral.

Los anexos muestran el código de la programación, tanto del método del Cubo como del programa Java presentado en el capítulo seis.

Finalmente, se ofrece la extensa bibliografía que ha sido utilizada en la realización de esta tesis doctoral.

CAPÍTULO 2

EL ESTADO DEL ARTE EN EL CREDIT SCORING. UNA REVISIÓN DE LOS PRINCIPALES TRABAJOS.

2. El estado del arte en el credit scoring. Una revisión de los principales trabajos.

2.1. Introducción.

Esta revisión de los trabajos relacionados con el objetivo de esta tesis, aunque no es exhaustiva, sin embargo sí cubre las principales investigaciones y trabajos técnicos que se han llevado a cabo utilizando tanto los métodos paramétricos como no paramétricos.

Las formas de enfrentarse al problema de la clasificación son variadas. La gran diversidad de técnicas existentes pueden incorporar análisis estadísticos, herramientas de minería de datos o inteligencia artificial con aprendizaje de máquina. La técnica estadística más clásica y más empleada en los problemas de credit scoring ha sido la regresión logística, que generalmente ofrece buenos resultados estadísticos. Otro enfoque clásico es sintetizar la información de la base de datos de clientes a través de reglas y de árboles de decisión; finalmente, otras aproximaciones más novedosas empleadas en los modelos de credit scoring se basan en la aplicación de redes neuronales, implementando algoritmos evolutivos, splines de regresión adaptativa, máquinas de vectores soporte o de la lógica borrosa y también se observan algunas aplicaciones a través del enfoque bayesiano.

Desde el primer trabajo donde se empleaban métodos estadísticos, Durand (1941) han transcurrido más de 70 años. Numerosos trabajos se han publicado desde entonces. Los pioneros y clásicos estudios que sirvieron para consolidar los modelos de credit scoring son: Myers y Forgy (1963), Bierman y Hauseman (1970), Orgler (1970) y Apilado *et al.* (1974). Algunos de estas referencias han servido para realizar una recopilación de estos trabajos estadísticos precursores de la situación actual, en este sentido citamos a Hand y Henley (1997) y a Thomas (2000).

A continuación realizaremos una breve revisión bibliográfica para los principales modelos paramétricos y no paramétricos.

Se denominan técnicas paramétricas de credit scoring aquellas que utilizan una función de distribución o clasificación conocida y que, por supuesto, estiman unos parámetros para explicar la variable dependiente, en este caso la concesión o no de la solicitud de crédito, de tal modo que estos parámetros de la ecuación se ajusten a las observaciones de una muestra. Estas técnicas son muy útiles si el conjunto de variables siguen una distribución propuesta. Cuando se dispone de la información se

modela con alguna técnica cuantitativa: logit, probit, análisis discriminante, modelos logarítmicos lineales, etc.

Las técnicas no paramétricas no requieren que se realicen supuestos sobre la distribución, es el otro extremo donde no se conoce, ni se supone ninguna forma concreta de la distribución, entran dentro de la filosofía de "dejar hablar a los datos" que es la forma de actuar razonable de los diferentes métodos y algoritmos de la minería de datos: redes neuronales, arboles de decisión y el resto de métodos descritos en esta tesis doctoral.

También podemos encontrar método semiparamétricos que son una vía entre los paramétricos y los no paramétricos y que participan de las ventajas e inconvenientes de ambos. En la modelización de credit scoring se puede disponer de un gran número de variables conocidas que son un subconjunto sobre la población total de las variables de los demandantes de crédito que dan lugar a métodos híbridos y que ya están poniéndose de moda en la modelización bancaria.

Antes de avanzar en el estado del arte de los modelos de credit scoring y al margen de la forma que adopten los modelos, hay que señalar dos cuestiones importantes:

1. Independientemente de las metodologías empleadas, estos modelos no gozan de aleatoriedad cuando se construye el modelo de credit scoring, dado que las muestras son muestras truncadas, ya que en la base de datos sólo se dispone de la información de los créditos concedidos sean estos devueltos o no, pero no se tiene información de los créditos denegados. Por ser la muestra truncada los estimadores de los valores poblacionales no son consistentes. Aun así, en general, podemos afirmar que las herramientas utilizadas ofrecen buenos resultados.
2. En la revisión bibliográfica no podemos decir que un modelo es mejor que otro sino que esos resultados responden a características particulares del ejemplo estudiado, a aspectos relacionados con la estructura de los datos y a las características de las variables y del tamaño y composición de la muestra utilizada, así como a la sensibilidad de la separación de las variables de clasificación.

Un amplio conjunto de técnicas han aparecido en las dos últimas décadas. De las técnicas paramétricas se ha seleccionado aquellos trabajos que utilizan el análisis discriminante y los modelos de probabilidad lineal y los modelos Logit y Probit.

Respecto a los modelos no paramétricos se recogen los resultados de algunos de los trabajos que han utilizado algunas de las técnicas más usadas: árboles de decisión, redes neuronales, máquinas de vectores soporte, modelos de lógica difusa, algoritmos genéticos y, por supuesto, los artículos que emplean las redes bayesianas, dado que en esta tesis doctoral se dan sobradas razones para la utilización de los métodos bayesianos como clasificadores óptimos, además de obtener una mayor información para la toma de decisiones a través de las probabilidades condicionales.

Hay que tener en cuenta que muchos autores proponen estudios comparativos donde se emplean tanto técnicas paramétricas como no paramétricas lo que añade un cierto grado de dificultad a la hora de ser clasificado.

2.2. Técnicas paramétricas de credit scoring.

En las técnicas paramétricas se recogen los trabajos relacionados con la regresión logística, el análisis discriminante y los modelos Probit.

2.2.1. Modelos que utilizan el análisis discriminante.

En relación con los trabajos que han utilizado el análisis discriminante se destacan en primer lugar, los trabajos pioneros antes citados de Duran (1941) y Myers y Forgy (1963). Posteriormente es Altman (1968)[2] con su conocido modelo Z el que marca una forma de proceder con variables explicativas utilizando ratios contables para determinar el fracaso empresarial, siendo muchos autores los que han empleado técnicas multivariantes en su intento de determinar la probabilidad de incumplimiento.

Señalar que los ratios empleados por Altman son todos internos, extraídos de la propia contabilidad de la empresa. Los cinco ratios utilizados como variables explicativas que emplea Altman son las siguientes: Ingresos netos/ventas, ganancias retenidas/activos, EBIT/activos, valor de mercado del patrimonio neto/valor libros deuda y ventas/activos. Posteriormente se incluyen otros variables de mercado como en el trabajo de Merton (1974)

Años después Altman y Saunder (1998) corrigen la primera estimación de la Z score cuya expresión es la siguiente:

$$Z''-Score = 3{,}25 + 6{,}56\,X_1 + 3{,}26\,X_2 + 6{,}72\,X_3 + 1{,}05\,X_4 \quad \text{.....} \quad (2.1)$$

[2] El trabajo de Altman fue precedido por el trabajo de Beaver (1966) que publica su pionero trabajo "Financial ratios as Predictor on Failure". Este trabajo adolecía de un problema al ser un modelo univariante clasificando sólo empresas ratio a ratio, existiendo la posibilidad de que una empresa sea clasificada de forma distinta por dos ratios.

Donde:

X_1=Capital de trabajo / Activo total

X_2=Reservas / Activo total

X_3=BAIT / Activo total

X_4=Capital en libros / Pasivo total

Según los autores, el modelo Z"-Score es una versión de cuatro variables del primer enfoque Z-Score. Fue diseñado para reducir las distorsiones en las puntuaciones de crédito para empresas de sectores diferentes. También encontraron que este modelo resulta extremadamente eficaz en la evaluación del riesgo de crédito de los bonos corporativos en el ámbito de los mercados emergentes, Altman *et al*. (1995).

Con el fin de estandarizar la ecuación Altman y Saunder (1998) señalan que su análisis es equivalente al de calificación de los bonos y añaden un término constante de 3,25 para el modelo; puntuaciones de cero (0) indica una D (default). Puntuaciones positivas indican clasificaciones superiores D. Los equivalentes actuales de calificación de bonos se derivan de una muestra de más de 750 estadounidenses de bonos corporativos con calificaciones promedio para cada categoría de calificación.

El estudio de Martín (1985) nos recoge un significativo conjunto de trabajos relacionados con el fracaso empresarial utilizando ratios económicos y financieros (ver tabla 2.1). Estos análisis abarcan una referencia temporal de 1972 a 1983 y como se observa los ratios utilizados como variables explicativas por los autores no son los mismos con lo que el grado de comparación entre los modelos resulta de difícil su comparación.

Posterior al año 1985 podemos citar al menos cuatro trabajos significativos que emplean el análisis discriminante como técnica de clasificación.

Un trabajo significativo es el de Falbo (1991) quien emplea 17 ratios muy utilizados en los estudios contables que calcula para 51 empresas durante tres años.

El Análisis discriminante basado en distancias lo aplican Boj *et al.* (2009) a los clientes de un banco alemán (base de datos German Credit del repositorio de la UCI). Los resultados que obtiene de pagadores y fallidos son 73,40% y 72,30% respectivamente.

Otros estudios que combinan varios métodos, incluidos el análisis discriminante son los de Esteve (2007) que aplica una red neuronal a través del algoritmo de Kohoonen

a una muestra cuya composición es del 70,5% de clientes pagadores frente a un 29,5% de clientes fallidos sobre un total de 897 observaciones. Los resultados obtenidos son de un 100% de los clientes pagadores frente a un 64% de fallidos.

Otros autores como Lee *et al.* (2002) utilizando una muestra de 2.000 observaciones de una cartera de créditos empleaando un modelo híbrido de análisis discriminante y red neuronal consiguen unos resultados de aciertos del 85,27% entre los pagadores y del 62,34% entre los fallidos.

Tabla 2.1. Modelos aplicados de análisis discriminante en la predicción de insolvencia empresarial.

Autores	Fecha publicación	Tipo, nº empresas, fecha, observación y nacionalidad	Variable observable	Variables independientes, tipo y número	Porcentaje aciertos, años antes del fracaso y tipo de muestra
Altman	1968	Empresas manufactureras, 33 saneadas y 33 fracasadas en el período 1946-1965, en USA.	Quiebra.	Capital de trabajo/activo total. Beneficios retenidos/activo total. BAI T/activo total. Valor mercado capital/valor contable de la deuda. Ventas/activo total. (5)	95%, un año, muestra inicial. 83%, dos años, muestra inicial. 96%, un año, muestra secundaria empresas fallidas. 79%, un año, muestra secundaria empresas saneadas.
Deakin	1972	Empresas industriales, 32 saneadas y 32 fracasadas en el período 1964-1970 en USA.	Quiebra.	Cash-flow/deuda total. Beneficio neto/activo total. Deuda total/activo total. Activo circulante/activo total. Activo disponible/activo/total. Capital de trabajo/activo total. Caja/activo total. Activo circulante/pasivo circulante. Activo disponible/pasivo circulante. Caja/pasivo circulante. Activo circulante/ventas. Activo disponible/ventas. Capital de trabajo/ventas. Caja/ventas. (14)	97%, un año, muestra inicial. 95,5%, dos años, muestra inicial. 95,5%, tres años, muestra inicial. 78%, un año, muestra secundaria.
Edmister	1972	Pequeñas empresas, 42 saneadas y 42 fallidas + 282 saneadas y 282 fallidas, en el período 1954-1969 en USA.	No haber devuelto un crédito a la Administración. USA.	Beneficio antes de impuestos más amortizaciones/pasivo circulante. Capital/ventas. Capital de trabajo neto/ventas. Pasivo circulante/capital. Existencias/ventas (Tend. Ascendente). Activo disponible/pasivo circulante. (Tendencia ascendente) (Tendencia descendente)	93%, conjunto de datos tres años antes de la concesión del préstamo, muestra inicial.

Tabla 2.1. Modelos aplicados de análisis discriminante en la predicción de insolvencia empresarial. Continuación

Autores	Fecha publicación	Tipo, nº empresas, fecha, observación y nacionalidad	Variable observable	Variables independientes, tipo y número	Porcentaje aciertos, años antes del fracaso y tipo de muestra
Blum	1974	Empresas industriales, 115 saneadas y 115 fracasadas, en el período 1954-68, en USA.	Quiebra.	Ratio de flujo disponible. Activo disponible neto/existencias. Cash-flow/deuda total. Patrimonio neto Mercado/deuda total. Patrimonio neto contable/deuda total. Tasa de retorno para los accionistas. Beneficio neto (Desviación estándar. Tendencia declinante. Pendiente línea tendencia). Activo disponible neto/existencias. (Desviación estándar. Tendencia declinante. Pend. Línea tendencia). (12)	93%, un año, muestra inicial. 80%, dos años, muestra secundaria. 70%, tercer, cuarto y quinto años, muestra secundaria.
Sinkey	1975	Empresas bancarias, 110 saneadas y 110 problemáticas en el período 1972-73. en USA.	Clasificación del banco como problemático según el FDIC.	Caja + valores del tesoro/activo. Préstamos /activo. Provisión pér. Pret./gastos operat. Préstamos/capital + reservas. Gastos operativos/ingresos operativos. Ingresos por préstamos/ingreso total. Ingresos valores Tesoro/ingreso total. Ingreso obligaciones estatales y locales/Ingreso total. Interés depósitos/ingreso total. Otros gastos/ingreso total. (10)	82%, un año, muestra inicial. 76%, dos años, muestra inicial. 75%, un año, muestra s/Lachen-bruch. 69%, dos años, muestra s/Lachen-bruch.
Altman y Loris	1976	Intermediarios financieros, 113 saneados y 40 fracasados, en el período 1971-73, en USA.	Liquidación forzosa.	Beneficio neto después de impuestos/activo total. Pasivo total + préstamos subord./capital. Activo total/capital neto ajust. Capital final-adiciones/capital inicial. Edad de la Empresa. Variable compuesta. (6)	90%, un año, muestra inicial. 86%, un año, muestra s/Lachen-bruch.

Tabla 2.1. Modelos aplicados de análisis discriminante en la predicción de insolvencia empresarial. Continuación

Autores	Fecha publicación	Tipo, nº empresas, fecha, observación y nacionalidad	Variable observable	Variables independientes, tipo y número	Porcentaje aciertos, años antes del fracaso y tipo de muestra
Altman, Haldeman y Narayanan	1977	Empresas manufactureras, y detallistas, 58 saneadas y 53 fracasadas, en el periodo 1969-75, en USA.	Quiebra.	BAIT/activo total. (Nivel de ratio Tendencia). BAIT/intereses deuda (log_{10}). Beneficios retenidos/activo total. Activo circulante/pasivo circulante. Capitales propios/capitales permanentes. Activo total (log_{10}).	93%, un año, muestra inicial. 89%, dos años, muestra inicial. 91%, un año, muestra s/Lachen-bruch.
Moyer	1977	Empresas industriales, 27 saneadas y 27 fracasadas, en el período 1965-75, en USA.	Quiebra.	Cash-flow/deuda total. Medida de descomposición del Balance (2)	85%, un año, muestra inicial. 83%, dos años, muestra inicial. 65%, tres años, muestra inicial.
Dambolena y Khoury	1980	Empresas manufactureras y detallistas, 23 saneadas y 23 fracasadas, en el período 1969-75, en USA.	Quiebra.	Beneficio neto/ventas. Beneficio neto/activo total. Activo fijo/patrimonio neto. (nivel de ratio y desviación típica). Deuda a LP/capital de trabajo neto. Deuda total/activo total. Existencias/capital de trabajo neto. (Desviación típica). (Modelo para cinco años, variables similares para un 3año). (7)	96%, un año, muestra inicial. 89%, tres años, muestra inicial. 87%, un año, muestra s/Lachen-bruch. 85%, un año, muestra s/Lachen-bruch.
Zollinger	1982	Empresas del sector construcción, 334 saneadas y 18 fallidas, en el período 1975-1977, en Francia.	Incidentes de pago.	Ventas. BAIT/valor de la producción. Autofinanciación/ventas. Fondo de rotación/neces. de financ. Activo neto/pasivo. Fondos propios/deudas. (6)	75%, conjunto de datos tres o cuatro años antes del impago de la deuda.

Tabla 2.1. Modelos aplicados de análisis discriminante en la predicción de insolvencia empresarial. Continuación

Autores	Fecha publicación	Tipo, nº empresas, fecha, observación y nacionalidad	Variable observable	Variables independientes, tipo y número	Porcentaje aciertos, años antes del fracaso y tipo de muestra
Richardson y Davidson	1983	Empresas que cotizaban en la American Stock Exchange, 686 saneadas y 18 fracasadas, en el año 1976, en USA.	Quiebra o suspensión de cotización en la Bolsa.	Las mismas del modelo de Altman 1968. (5)	72%, dos años, muestra simulada a partir de la muestra inicial. (25 empresas fracasadas y 700 saneadas)
El Hennawy y Morris	1983	Empresas manufactureras, de construcción distribución, 53 saneadas y 53 fracasadas, en el período 1960-71, en el Reino Unido.	Quiebra.	BAIT + amortizaciones/activo total. Deuda a LP/capital neto. Activo circulante/activo total. Variable dicotómica para el sector de construcción. Variable dicotómica para el sector de distribución. (5)	98%, un año, muestra inicial. 98%, un año, muestra secundaria. 91%, dos años, muestra inicial. 100%, dos años, muestra secundaria.

Fuente: Martín (1985)

2.2.2. Modelos de probabilidad lineal.

El precursor de la línea de investigación en modelos de credit scoring utilizando modelos de regresión lineal es Orgler (1970) cuyas variables explicativas son ratios financieros. En su estudio determina tres puntos de corte delimitando tres regiones: créditos malos, créditos buenos y créditos marginales o no decisivos. Los porcentajes de acierto para la muestra de validación son del 80,0%.

Un año más tarde el mismo autor, Orgler (1971) construye un modelo de créditos al consumo utilizando cuatro grupos de variables contables: liquidez, rentabilidad, apalancamiento y actividad.

Otros trabajos que abordan esta metodología son los de Plotnicki (2005) los de Avery *et al.* (2004).

Estos modelos dejan de utilizarse en favor de otras formas funcionales más expresivas y con mayor capacidad predictiva.

2.2.3. Aplicaciones de credit scoring con modelos Logit.

Uno de los primeros autores en realizar un análisis Logit aplicado a la banca comercial es Wiginton (1980) que a través de una muestra de 1.908 solicitudes y de pocas variables explicativas realiza un análisis comparativo entre los dos modelos paramétricos Logit y discriminante concluyendo que el análisis Logit es mejor en el porcentaje correcto de clasificaciones. El estudio lo realiza con dos submuestras de 954 observaciones. En ambas, el análisis discriminante acierta el 100% de los créditos devueltos frente a un 58,18% de los fallidos. El análisis de regresión logística arroja un porcentaje global cercano al 62% siendo mayor el porcentaje de fallidos bien clasificados en las dos submuestras utilizadas.

Tres años después de este pionero estudio, Campbell y Dietrich (1983), realizan un trabajo explicativo de los determinantes del crédito en préstamos hipotecarios.

Por su parte Gardner y Mills (1989) emplean tres modelos de regresión logística a una cartera de créditos para estudiar el efecto simultáneo de las variables seleccionadas sobre la probabilidad de caer en situación de morosidad. El porcentaje una correcta clasificación estuvo entre el 45% y el 65%.

Para préstamos personales Steenackers y Goovaerts (1989) realizan una aplicación Logit con datos de una entidad financiera belga cuyo modelo arroja una clasificación correcta del 69,60% de aciertos.

Lawrence y Arshadi (1995) utilizan la regresión logística multivariante para ver la opción más favorable en clientes que ya habían entrado en calidad de morosos. Las tres situaciones que contemplan son: ejecutar la hipoteca, renegociar el préstamo y alargar el vencimiento del préstamo.

En el cuadro siguiente adaptado de Rodríguez-Vilariño (1995) se recogen los principales trabajos que utilizan la regresión logística y que cubren el periodo 1970–1992:

Tabla 2.2. Modelos aplicados de análisis de regresión logística en la predicción de insolvencia empresarial.

MODELOS LOGÍSTICOS PARA VARIOS SECTORES									
Autor	Año publica-ción	Sector	Tamaño de la muestra	Nº Años	Nª de Ratios	Muestra de valida-ción	Model o para cada año	Exactitud %	Ratio principal
ESTADOS UNIDOS:									
Chesser	1974	Todos	126	2	15	SI	NO	75	BAIT/Activo total
Olhson	1980	Industrial	2.163	3	9	NO	SI	96	Pasivo Exigible/Activo total
Collins y Green	1982	Todos	323	-	5	SI	-	94	-
Zavgren	1983	Industrial	90	5	7	NO	SI	82	Test ácido
Hamer	1983	Manufac-turero	88	5	-	-	SI	-	Conclusiones dispares
Gentry	1985	Todos	66	3	7	SI	SI	83	Dividendos/Flujos Caja
Casey y Bartozak	1985	Todos	290	5	9	NO	SI	88	-
Lo	1986	Todos	76	3	6	NO	SI	-	Bº Neto/Activo total
Koh	1992	Todos	330	1	6	NO	-	99,9	Bº Neto/Activo total
GRAN BRETAÑA:									
Peel y Pope	1986	Todos	78	2	9	SI	SI	97	Recursos generales/Pasivo Exigible
Keasy y Watson	1987	Todos	146	3	46	SI	NO	82	Variable no financiera
Keasy y Mcguinness	1990	Industrial	86	5	16	SI	SI	86	BAIT/Ventas
BÉLGICA:									
Goghe, Joos y Vos	1992	Todos	1.109	5	-	SI	-	93	-
ESPAÑA:									
Gabás Trigo	1990	Industrial	101	10	50	SI	SI	98	-
ESTADOS UNIDOS									
Martín	1977	Banca	5.700	3	25	SI	NO	92	Préstamos Comerciales/ Préstamos Totales
Barniv y Hershbarger	1990	Seguros	56	2	20	SI	SI	91	Bº Neto/Primas Totales
Barniv	1990	Seguros	211	3	5	SI	SI	96	Medida de Descomposición del Pasivo
ESPAÑA									
Laffarga Briones	1986	Banca	48	5	15	NO	SI	93	BAT/Activo Total
Pina	1988	Banca	45	3	9	SI	SI	92	Activo Circulante/Pasivo Exigible
Rodríguez Acebes	1990	Seguros	50	3	3	NO	SI	94	Disponible/Pasi-vo a Corto

Fuente: Adaptado de Rodríguez-Vilariño (1995)

Otra comparación entre regresión logística y análisis discriminante la realizan Mures *et al.* (2005). Realizan un muestreo por conglomerados entre las cajas de ahorro, bancos y cooperativas de crédito de Castilla y León y extraen 70 clientes entre los nueve conglomerados elegidos al que aplican los modelos de regresión logística y análisis discriminante. Concluyen su estudio con una tasa alta de aciertos: el 100% en pagadores y el 88,89% para fallidos cuando aplican el análisis discriminante y el 98,08% y 94,44, para pagadores y fallidos cuando utilizan la regresión logística.

Algunos autores como Belloti y Crook (2007) incluyen el ciclo económico a la hora de aplicar el credit scoring. Inicialmente realizan una regresión logística sin variables macroeconómicas y, posteriormente, incorporan variables relacionadas con el ciclo económico: tipos de interés, Índice General Bursátil, Producto Interior Bruto, tasa de desempleo, precios de la vivienda, índice general de precios y un ratio de riqueza que incluye valores de renta fija. La conclusión a la que llegan es que la inclusión de estas variables mejora la capacidad predictiva del modelo.

En China, para créditos comerciales Yang *et al.* (2009) desarrollan un análisis logit donde alcanzan un clasificación correcta del 94,7%.

Un modelo Logit que ha sido muy utilizado por la Confederación Española de Cajas de Ahorro, y que ha servido para algunas entidades de crédito, está basado en el planteado por Siddiqui (2006). Este autor presenta una metodología apoyada en modelos Logit utilizando una agrupación óptima de atributos de las variables explicativas del riesgo. A partir de este modelo construye una herramienta de decisión que llama “tarjeta de puntuación”.

Destacar en este apartado el estudio que realiza Mallo (2011) con datos de la Caja España donde propone una alternativa a los modelos de regresión logísticos basada en las ideas de Hastie y Tibshirani (1996) a la que denomina Modelos Logísticos Lineales Híbridos de Expansiones Lineales por funciones de base, los cuales se obtienen al expandir la componente no lineal de los Modelos Logísticos Parcialmente Lineales

2.2.4. Aplicaciones de credit scoring con modelos Probit.

Entre los trabajos pioneros de este análisis se encuentra el de Boyes et al (1989) donde utilizan un muestra de 4.632 créditos para evaluar la probabilidad de impago y el beneficio esperado para cada operación de préstamo. El 80,1% de los préstamos

fueron concedidos mientras que el 19,9% restante fueron denegados. De los que se concedieron fueron calificados a priori como buenos el 52,2% por parte de la institución. En este estudio se utilizan variables personales del cliente, variables económicas y variables financieras construidas a través de ratios.

Un estudio muy similar es el de Green (1992). Este autor consideró un mayor número de variables demográficas y socioeconómicas, variables del prestamista y macroeconómicas. Considerando tres diferentes puntos de corte (0,09487, 0,12 y 0,15) alcanza los siguientes porcentajes de acierto 57,21%, 67,92% y 77,88%).

Falkenstein *et al.* (2000) trabajan en un modelo que utiliza Moody's para predecir el default y que se denomina Risk Calc^{TM}.

Años más tarde, empleando también los modelos Probit, Tsaih *et al.* (2004) desarrollan una aplicación de credit scoring.

Algunos investigadores abordan la concesión de crédiotos a través de varios modelos Probit. Jacobson y Rossbach (2003) realizan una regresión Probit para determinar la probabilidad de obtener un préstamo y emplean otro modelo Probit para obtener la probabilidad de que éste no fuera fallido. Estos autores utilizan una muestra de 13.338 registros de una entidad bancaria Sueca.

Otro autor, Bonfim (2009) utiliza diez modelos para extraer diversas conclusiones relativas a la probabilidad de impago en un estudio de una entidad financiera con datos de 30.000 empresas. Las variables explicativas surgidas de los estados contables, se complementaron con otras como la antigüedad de la empresa la productividad del capital, los activos tangibles, el volumen de negocio, las garantías del préstamo, el sector económico de actividad, así como el tamaño de la compañía. Un resultado interesante que se deduce de este estudio es que los resultados de los modelos mejoran considerablemente si se tiene en cuenta el ciclo económico.

2.3. Técnicas no paramétricas de credit scoring.

Las técnicas no paramétricas no están ligadas a ninguna forma funcional por eso también se les conoce como método de distribución libre. En estos métodos se desconoce la forma de la relación funcional pero han demostrado que son muy útiles en muestras pequeñas y en modelos no lineales.

Son muchos los modelos que se han construido en las tres últimas décadas con el desarrollo de las nuevas tecnologías que ha producido una enorme eclosión de modelos desde áreas de la informática, la ingeniería, etc. Como técnicas más representativas de modelos no paramétricos utilizadas en las aplicaciones de credit scoring, se encuentran fundamentalmente las siguientes: los modelos de programación lineal, las redes neuronales, los árboles de decisión, las máquinas de vectores soporte, la programación genética, los métodos híbridos, los multiclasificadores, los algoritmos evolutivos, los modelos que utilizan lógica borrosa y las redes bayesianas.

Este estudio bibliográfico de los principales trabajos relacionados con el estudio del arte sobre el credit scoring aplicando modelos no paramétricos se ha beneficiado de algunos documentos elaborados por otros autores. Entre ellos, quiero destacar los cinco siguientes: Ramírez (2008), Lahsana *et al.* (2010), Abdou y Pointon (2011), Keramati y Yousefi (2011) y Sadatrasoul *et al.* (2014).

2.3.1. Aplicaciones de credit scoring con modelos de programación lineal.

Una de las aplicaciones más conocidas de programación lineal es el modelo DEA (Data Envelopment Analysis) que fue desarrollado por Charnes *et al.* (1978). Tres años más tarde Fred y Glover (1981a, 1981b) demuestran que encontrar la función lineal que mejor discrimina entre grupos, dadas unas variables explicativas, podía ser considerado como un problema de programación lineal.

En las aplicaciones de credit scoring estos modelos asignan una puntuación de eficiencia financiera a cada cliente en relación al resto de los clientes que forman la muestra. Cada puntuación es ordenada de forma ascendente tomando como criterio la pérdida esperada que sufriría la entidad bancaria.

Varios autores han desarrollado esta técnica como alternativa para la predicción de impagos en la concesión de créditos: Bajgier y Hill (1982), Choo y Wedley (1985), Glover *et al.* (1988), Lam *et al.* (1993), Lam *et al.* (1996) y Emel *et al.* (2003).

Destacamos el trabajo de Lam *et al.* (1996) porque incorporan una variante a los modelos existentes incluyendo las desviaciones de los objetos (créditos) correctamente clasificados. También proponen una metodología para determinar el punto de corte óptimo en la clasificación. Con una muestra de 300 concesionarios, divididos en dos partes iguales para la estimación y posterior validación, alcanzan un

porcentaje de acierto del 99,33% en la muestra de estimación y un 93,33 en la muestra de validación, superando al porcentaje obtenido mediante el análisis discriminante.

En el sector empresarial, Emel *et al.* (2003) utilizando 46 ratios financieros de 82 empresas industriales y desarrollan un modelo de credit scoring.

Otro ejemplo ilustrativo lo encontramos en Tsai *et al.* (2009). En este estudio sobre personas que han pedido un préstamo personal en una institución financiera de Taiwán se aplica el modelo DEA-DA (Análisis envolvente de datos con análisis discriminante) y se compara con otros tres modelos: redes neuronales, regresión logística y análisis discriminante.

En esta investigación se realiza una encuesta a los prestatarios donde se consigue información sobre sus actitudes hacia el dinero (según la escala de Yamauchi y Templer´s de 1982[3]) y otras variables socio-demográficas. El total de individuos que formaron la muestra fue de 1.877 de los que 1.504 eran clientes buenos frente a 207 que resultaron ser clientes morosos.

El resultado de este estudio señala que el mejor método en cuanto al porcentaje total de clientes correctamente clasificados es el DEA-DA. Otra conclusión es que el porcentaje de aciertos del modelo se incrementa cuando se añaden las variables de la encuesta.

2.3.2. Aplicaciones de credit scoring con redes neuronales.

La principal ventaja que muestran las redes neuronales es su capacidad de generalización a partir de las observaciones reales. Además son muy robustas cuando se presentan situaciones de falta de información en los registros de las variables predictivas. Estos modelos dieron un enorme impuso al credit scoring.

Las arquitecturas más utilizadas, siguiendo a West (2000), son las siguientes:

- Mezcla de Expertos (Mixture of Expert, MOE).
- Funciones de Base Radial (Radial Basis Function, RBF).

[3] Yamouchi y Templer crearon una escala de actitudes hacia el dinero ampliamente reconocida basada en literatura clínica y teórica. Identificaron tres áreas de contenido general relacionado con el dinero y la psicología: la seguridad, la conservación y el poder del prestigio. Generaron 62 variables que más tarde redujeron a un 29 ítem que llamaron MAS (Money Attitudes Scale) basada en una muestra de adultos. A través del análisis factorial extrajeron cuatro factores: el poder del prestigio, la conservación del dinero, la desconfianza y la ansiedad que proporcionaron una evaluación fiable de las actitudes hacia de dinero

- Perceptrón Multicapa (Multi-layer perceptron, MLP).
- Learning Vector Quantification (LVQ).
- Fuzzy Adaptative Resonance (FAR).

Desde su aparición, son muchos los autores que han realizado trabajos no solo con las redes neuronales sino que han sido éstas comparadas con todo tipo de modelos, tanto paramétricos como no paramétricos. En un epígrafe posterior señalaremos algunos de estos trabajos. Ahora simplemente se referenciaran algunos de los trabajos pioneros y los más significativos relacionados con el credit scoring.

Uno los primeros trabajos comparativos de las redes neuronales con otras técnicas paramétricas de credit es debido a Davis *et al.* (1992). Dos años más tarde encontramos otros autores como Ripley (1994) y Rosenberg y Gleit (1994) que describen algunas de las aplicaciones de las redes neuronales para la gerencia del credit scoring y la detección del fraude.

Otros importantes trabajos de tres autores: West (2000), Hsieh (2005), Yu *et al* (2008 y 2009) se detall**a**n en la tabla siguiente:

Tabla 2.3. Modelos de Credit Scoring aplicando redes neuronales.

Autores	Fecha publicación	Número de observaciones	Tipo de Red Neuronal	Resultados de la investigación
WEST	2000	1000 (Entidad financiera Alemana)	MOE	75,66
			RBF	74,60
			MLP	73,28
			LVQ	63,37
			FAR	57,23
WEST	2000	690 (Entidad financiera Australiana)	MOE	86,68
			RBF	87,14
			MLP	85,84
			LVQ	82,97
			FAR	75,39
HSIEH	2005	1000 (Entidad financiera Alemana) - 690 (Entidad financiera Australiana)	Clustering + MLP	Resultados para ambas entidades en base a 10 agrupaciones
YU ET AL	2008	60 Empresas (30 pagadoras y 30 fallidas)	Análisis Regresión Logística	70,77
			MLP	73,63
			LVQ	77,84
			FAR	79,00
YU ET AL	2009	1.225 Créditos (902 pagados y 323 fallidos) – Entidad financiera Inglesa	Regresión lineal	63,38
			Regresión Logística	63,72
			RBF	68,89
			MLP	62,22
			LVQ	69,92
			FAR	80,14

Tabla 2.3. Modelos de Credit Scoring aplicando redes neuronales. Continuación.

Autores	Fecha publicación	Número de observaciones	Tipo de Red Neuronal	Resultados de la investigación
YU ET AL	2009	653 Créditos (357 concedidos y 298 rechazados) – Entidad financiera Japonesa	Regresión lineal	82,21
			Regresión Logística	83,00
			RBF	83,79
			MLP	81,03
			LVQ	80,24
			FAR	86,17
YU ET AL	2009	1.000 Créditos (700 concedidos y 300 rechazados) – Entidad financiera Alemana	Regresión lineal	66,00
			Regresión Logística	72,40
			RBF	74,00
			MLP	70,40
			LVQ	77,00
			FAR	82,00

Fuente: Elaboración propia a partir de los trabajos originales.

Un trabajo importante donde se compara la red neuronal Fuzzy ART con la regresión lineal y logística y la red neuronal perceptrón multicapa es el que llevaron a cabo Jiang y Lin (2010) sobre una muestra de clientes de créditos de consumo de un banco de Shenzhen en China. La mayor superioridad del modelo Fuzzy ART descansa en el menor porcentaje de error de tipo II. Altman, (1998) sugiere que la pérdida causada por el error de tipo II es de 20 a 60 veces más alto que el error tipo I. para los bancos comerciales que otorgan préstamos de consumo personal, la pérdida causada por el error de tipo I es sólo el beneficio que proviene de los clientes. Sin embargo, el error de tipo II no sólo hace que el banco pierda capital, también pierde los beneficios que se hacen por el préstamo del capital a los clientes.

Tabla 2.4. Resultados de la clasificación de varios modelos.

Model	Testing Samples		
	Type I Error	Type II Error	Accuracy
Linear Regression	2.67%	12.00%	92.00%
Logistic Regression	2.67%	24.00%	86.00%
BP Neural Network	2.67%	16.00%	92.67%
Fuzzy ART Model	14.67%	8.00%	88.67%

Fuente: Jiang y Lin (2010).

2.3.3. Aplicaciones de credit scoring con Árboles de Decisión.

Debido a la cómoda interpretación de los árboles de clasificación y regresión han sido bastante utilizados como modelos explicativos, bien de forma solitaria como en estudios comparativos con otros modelos.

El modelo presentado por Friedman (1977) denominado Recursive Partioning Algorithm, posteriormente mejorado con aportaciones de Breiman *et al.* (1984), Marais *et al.* (1984) y más tarde por Friedman *et al.* (1985) sirve para utilizar esta metodología a algunos autores que la aplican a problemas relacionados con credit scoring. Algunos de los pioneros que han trabajado con esta técnica son los siguientes: Makowski (1985), Coffman(1986), Carter y Catlett (1987) y Boyle *et al.* (1992).

Otro trabajo después de las recomendaciones de Basilea II es el desarrollado por Cardona (2004) con datos de un banco sudamericano, En este trabajo se presenta la utilización de los árboles de decisión como un herramienta para el cálculo de probabilidades de incumplimiento de los créditos mostrando sus ventajas y desventajas. Empleando las probabilidades obtenidas de los árboles de decisión y la severidad, se calcula el valor de pérdida esperada con la cual se realiza la provisión tal y como lo reglamenta la Superintendencia Bancaria.

Un estudio reciente de Espín y Rodríguez (2013) estudia el método CHAID como árbol de clasificación y la regresión logística a clientes sin referencias crediticias.

Tabla 2.5. Resumen de resultados de estudios sobre comparación de modelos.

AUTOR/ES	PORCENTAJE CORRECTO DE CLASIFICACIÓN					
	Análisis Discriminante o Regresión Lineal	**Modelo Logit**	**Modelo Probit**	**Programación Lineal**	**Redes Neuronales**	**Árboles de Decisiones**
Srinivasan y Kim (1987)	87,50	89,30		86,10		93,20
Boyle et al. (1992)	77,50			74,70		75,00
Henley (1995)	73,40	43,30				43,80
Desai et al. (1996)	81,12	81,70		80,75	80,46	
Arminger et al. (1997)		67,59			65,25	66,42
Desai et al. (1997)	66,50	67,30			66,40	67,30
Yobas et al. (2000)	68,40				62,00	62,30
Lee et al. (2002)	71,40	73,50			73,70 (77,00)[a]	
Malhotra y Malhotra (2003)	69,30				72,00	
Baesens (2003)[b]	88,30	87,40		88,30	88,30	90,40
Huang et al. (2006)		86,19		87,93	68,42	85,81[c] 87,06[d]
Hu y Ansell (2007)		84,31 (88,70)		85,12 (87,72)[e]	88,21 (89,51)[e]	88,29 (88,69)[e]

Tabla 2.5. Resumen de resultados de estudios sobre comparación de modelos. Continuación.

AUTOR/ES	PORCENTAJE CORRECTO DE CLASIFICACIÓN					
	Análisis Discriminante o Regresión Lineal	**Modelo Logit**	**Modelo Probit**	**Programación Lineal**	**Redes Neuronales**	**Árboles de Decisiones**
Abdou (2009)	85,42	82,81			89,58	
Abdou y Pointon (2009)	79,40	82,09			87,64	
Chuang y Lin (2009)	76,00	76,50		79,50	(86,00)[f]	77,50
Zhou et al. (2010)	72,07	77,22	77,27		75,30	70,35

[a] Modelo híbrido de red neuronal y análisis discriminante
[b] Se ha escogido el modelo de mayor porcentaje correcto de clasificación
[c] CART
[d] C4.5
[e] Resultados incorporando variables del entorno económico
[f] Red neuronal híbrida
Fuente: Lara (2010)

A continuación se muestran, en forma de cuadro, algunas ventajas y desventajas de los modelos comentados hasta ahora.

Tabla 2.6.Ventajas e Inconvenientes en algunas técnicas utilizadas en clasificación.

			Ventajas	Inconvenientes
Técnicas Paramétricas	**Lineales**	**Análisis Discriminante**	▪ Buen rendimiento para grandes muestras. ▪ Técnicamente conveniente en la estimación y mantenimiento.	▪ Problemas estadísticos y estimadores ineficientes. ▪ No arroja probabilidades de impago.
		Modelos de Probabilidad Lineal	▪ Buen rendimiento para grandes muestras. ▪ Sugieren probabilidades de impago ▪ Parámetros fácilmente interpretables.	▪ Estimadores ineficientes. ▪ Las probabilidades estimadas podrían quedar fuera del intervalo (0,1).
	No-Lineales	**Modelos Logit**	▪ Buenas propiedades estadísticas y no son estrictos con las hipótesis sobre los datos. ▪ Muestran las probabilidades de impago. ▪ Gran rendimiento respecto a la metodología y resultados.	▪ Dificultad de interpretación de los parámetros.
		Modelos Probit	▪ Buenas propiedades estadísticas y no son estrictos con las hipótesis sobre los datos. ▪ Muestran las probabilidades de impago.	▪ Dificultad de interpretación de los parámetros. ▪ Proceso de estimación relativamente complicado.
Técnicas no Paramétricas		**Programación Lineal**	▪ Apto para gran cantidad de variables. ▪ Modelo de gran flexibilidad. ▪ No requiere una especificación previa del modelo.	▪ No estima parámetros ni probabilidades de impago. ▪ Difícil comprensión. ▪ Inexactitud en la predicción.
		Redes Neuronales	▪ Gran predicción en muestras pequeñas. ▪ Modelo de gran flexibilidad. ▪ No requiere una especificación previa del modelo.	▪ No estima directamente parámetros ni probabilidades de impago. ▪ Difícil comprensión
		Árboles de decisiones	▪ El mejor rendimiento para muchos autores. ▪ Modelo de gran flexibilidad. ▪ No requiere una especificación previa del modelo.	▪ No estima parámetros ni probabilidades de impago. ▪ Difícil comprensión.

Fuente: Lara (2010).

2.3.4. Aplicaciones de credit scoring con Máquinas de Vectores Soporte.

Desde su creación, las novedosas máquinas de vectores soporte (Support Vector Machine, SMV), Vapnik, (1995), no han dejado de aplicarse en todo clase de trabajos relacionados con la clasificación por su buen desempeño. En modelos de credit scoring después del desarrollo presentado por Vapnick (1998) son varios investigadores los que han aplicado esta metodología al problema de evaluación del riesgo de crédito y han mostrado que las máquinas de vectores soporte son más precisas que otras propuestas de clasificación. Autores como Yu *et al*. (2010), Belloti y Crook (2009), Xu *et al.* (2009), Li *et al.* (2006) han empleado y comparado las máquinas de vectores soporte con otros modelos.

Para contrastar algunas características de este procedimiento con otros modelos en cuanto a la exactitud, interpretación, sencillez y flexibilidad utilizaremos el cuadro presentado por Yu *et al.* (2008).

Tabla 2.7. Comparación de metodologías de riesgo de crédito para la elaboración de scoring.

Método	Exactitud	Interpretación	Sencillez	Flexibilidad
LDA, LOG yPR	★★	★★★	★★★	★
DT	★★	★★★	★★	★
KNN	★	★★★	★★★	★
LP	★	★★★	★★	★★★
NN	★★★	★	★	★
EA	★★	★	★	★
RS	★★	★	★★	★
SVM	★★★	★★	★	★★★
Hybrid/ensemble	★★★	★	★	★★

Fuente: Yu *et al.* (2008)**.** Las siglas de los métodos se encuentran en inglés. LDA: Análisis discriminante lineal. LOG: Regresión logística. PR: Regresión Probit. DT: Árboles de decisión. KNN: K-nearest neighbors. LP: Programación Lineal. NN: Redes neuronales. EA: Algorítmos evolutivos. RS: Rough Sets. SVM: Máquinas de vectores de soporte. Hybrid/ensemble: Híbridos.

Yu *et al.* (2008) realizan un estudio con la base de datos de clientes de un banco alemán donde aplican la máquinas de vectores soporte realizando la estimación con funciones de base radial utilizando un algoritmo iterativo que denominan nearest point algorith (NPA). Los autores afirman que al comparar los resultados con otros veintidós métodos los resultados son prometedores.

Härdle *et al.* (2005, 2007, 2011) aplican esta metodología a empresas alemanas realizando un cálculo de las probabilidades de incumplimiento. A través de la puntuación proporcionada por los vectores soporte y mediante funciones monótonas

decrecientes calculan la probabilidad de incumplimiento. Afirman que los resultados alcanzados con su método resultan ser mejores que los que obtienen con el análisis discriminante y con la regresión logística.

En un trabajo interesante con las SVM realizada por Moreno y Melo (2011) y que se aborda a través de dos bases de datos, una primera base de datos de clientes tomada de Frank y Asunción (2010) y, otra segunda, relacionada con empresas de una entidad financiera de Colombia. En este trabajo los autores efectúan una comparación de la metodología SVM con la regresión logística y el análisis discriminante lineal. Los resultados que se obtienen señalan una mayor capacidad predictiva de las SMV. En el caso del banco alemán el modelo con mejor desempeño fue el SVM con kernel RBF y para el caso colombiano fueron las SVM con kernel lineal.

Tabla 2.8. Resultados para las muestras de entrenamiento y test. Precisión en la clasificación (%). Base de datos de Frank y Asunción (2010).

	Muestra de entrenamiento				Muestra de test			
	Total	Buenos clientes	Malos clientes	ROC	Total	Buenos clientes	Malos clientes	ROC
Logit-Step	79,9	84,3	69,1	76,7	69,5	73,7	61,2	67,4
Logit-Selec	78,8	91,5	48,1	69,8	72,0	84,9	46,3	65,6
LDA	80,3	84,0	71,2	77,6	69,5	72,9	62,7	67,8
SVM-RFB	85,0	96,7	56,7	76,7	75,5	91,7	43,3	67,5
SVM-RFB-U	86,4	92,1	72,5	82,3	75,5	82,0	62,7	73,3
SVM-POL	80,4	89,6	57,9	73,8	71,0	82,7	47,8	35,2
SVM-POL-U	81,3	92,1	4,9	73,5	71,0	86,5	40,3	63,4
SVM-LIN	80,9	91,0	56,2	73,6	70,5	100,0	0,0	63,8
SVM-LIN-U	81,0	91,0	56,7	73,8	71,0	85,0	43,3	64,1

Logit-Step corresponde al modelo de regresión logística seleccionado con el método stepwise de Efroymson (1960). Logit-Selec corresponde al modelo de regresión logística selecionada con stepwise utilizando errores robustos. LDA corresponde al modelo de análisis discriminante. SVM-RBF, SVM-RBF-U, SVM-POL, SVM-POL-U, SVM-LIN y SVM-LIM-U corresponden a las máquinas de vectores soporte con kernel IRBF sin umbral y con umbral, con kernel polinomial sin y con umbral respectivamente.
Fuente: Adaptado de Moreno y Melo (2011)

Tabla 2.9. Resultados para las muestras de entrenamiento y test. Precisión en la clasificación (%) Base de datos del Banco colombiano.

	Muestra de entrenamiento				Muestra de test			
	Total	Buenos clientes	Malos clientes	ROC	Total	Buenos clientes	Malos clientes	ROC
Logit-Step	82,8	92,2	44,7	68,4	82,5	89,9	53,7	71,8
Logit-Selec	82,8	92,2	35,8	65,1	84,5	93,1	51,2	72,2
LDA	82,9	94,1	37,7	65,9	84,5	93,1	51,2	72,2
SVM-RFB	90,4	99,4	54,1	76,7	85,5	95,0	48,8	71,9
SVM-RFB-U	91,9	97,8	67,9	82,9	83,0	88,1	63,4	75,7
SVM-POL	83,8	99,4	20,8	60,1	83,0	97,5	26,8	62,2
SVM-POL-U	86,0	97,0	41,5	69,3	83,0	91,2	51,2	71,2
SVM-LIN	80,5	99,5	3,8	51,7	79,5	100,0	0,0	50,9
SVM-LIN-U	82,4	97,5	21,4	59,4	74,5	76,1	68,3	72,2

Fuente: Adaptado de Moreno y Melo (2011).

2.3.5. Aplicaciones de credit scoring con Algoritmos Evolutivos.

Se incluye en este apartado los algoritmos genéticos y la programación genética, considerada esta última como una evolución de los algoritmos genéticos, Koza, (1992) y Golgberg, (1989) El rápido crecimiento de estas técnicas se han aplicado a la predicción de la bancarrota: Etemadi *et al.* (2009) y Mckee y Lensberg (2002); aplicaciones de scoring, Huang *et al.* (2007) y Huang *et al.* (2006) o a problemas de rendimiento financiero, Xia *et al.* (2000).

Huang y Tzeng (2006) proponen un modelo de programación genética cuyos resultados son comparados con otros modelos de redes neuronales, árboles de decisión, conjuntos rugosos y regresión logística aplicados a la base de datos de un banco germano y otro australiano.

En el trabajo propuesto por Huang *et al.* (2006) proponen un modelo de programación genética en dos etapas para tratar un problema de credit scoring incorporando las ventajas de las reglas IF-THEN y una función discriminante. Este modelo es comparado con un modelo simple de programación genética, un perceptrón multicapa, un árbol de clasificación y regresión (CART) y un modelo de regresión logística. Estos autores también emplean los datos del banco alemán y del australiano.

2.3.6. Aplicaciones de credit scoring con Redes Bayesianas.

Las redes bayesianas aparecen en los años noventa generando aplicaciones en temas relacionados con la gestión y la toma de decisiones en general y, de forma específica, en el análisis de fiabilidad, en el análisis de riesgos y también en el desarrollo de sistemas expertos.

Las primeras referencias generales las podemos encontrar en Pearl (1988), Henrion (1988), Morgan y Henrion (1990), Neapolitan (1990), Heckerman (1996) y Jensen (1996).

Sarkar y Sriram (2001) desarrollan a través de Redes Bayesianas, clasificadores para la alerta temprana de quiebras bancarias. Sun y Shenoy (2007) proveen una guía operacional para la construcción de un clasificador Naïve Bayes para la predicción de la bancarrota.

Baesens *et al.* (2002) aplican el clasificador bayesiano Naïve Bayes, TAN (Tree Augmented Naïve Bayes) y Redes Bayesianas utilizando una simulación Markov

Chain Monte Carlo. En ambos trabajos la estimación del modelo utiliza el método cross validation de 10 folder y la selección de variables la realizan a través del manto de Markov. Este mismo procedimiento lo utilizan Hsieh y Hung (2010) con datos de una sucursal alemana al que previamente han realizado un cluster para detectar grupos homogéneos.

Las redes Bayesianas relacionadas con el riesgo operacional se encuentran descritas en Doldán (2007).

Siguiendo la metodología de Redes Bayesianas, Beltrán *et al.* (2013) con una base de datos de una caja de ahorros de La Rioja aplican tres redes bayesianas que buscan y optimizan la métrica bayesiana a través de los algoritmos K2, HC (Hill Climbing) y TAN (Tree Augmented Naïve Bayes). Estos modelos se comparan con la regresión logística, máquinas de vectores soporte, dos modelos de redes neuronales, el algoritmo C.4.5, como árbol de clasificación y seis métodos multiclasificadores, Tanto en la fase de entrenamiento como en la de test las redes bayesianas muestran una mayor precisión.

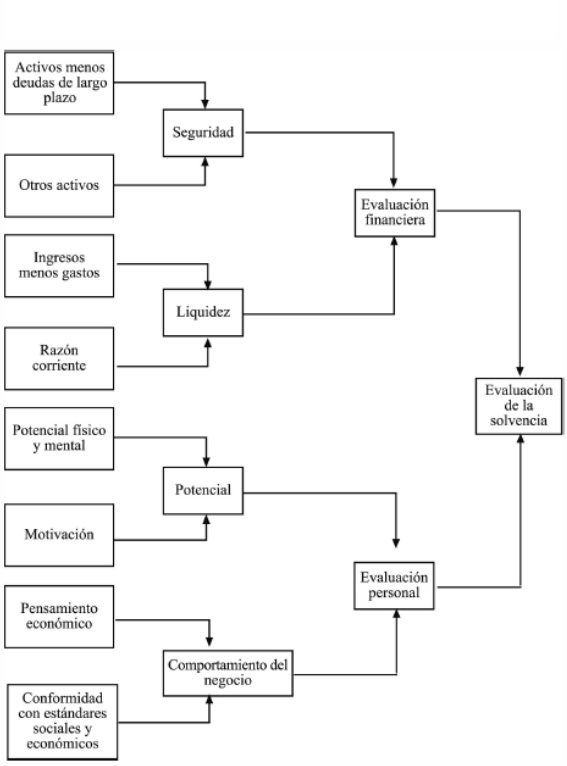
Martínez–Sánchez y Venegas-Martínez (2013) presentan un enfoque bayesiano para calcular el riesgo operacional, identificando y cuantificando los diversos factores de riesgo operacional asociados con las diferentes líneas de negocios de bancos transaccionales. El modelo de red bayesiana es calibrado a través de eventos en las líneas de negocios durante el período 2006-2009. Los autores, para calibrar el modelo incluyen fuentes de información, tanto objetivas como subjetivas, las cuales permiten capturar de forma adecuada las relaciones de causa y efecto entre los diferentes factores de riesgo.

2.3.7. Aplicaciones de credit scoring con modelos de lógica difusa (Fuzzy logit).

Para evaluar la solvencia y capacidad de pago de los solicitantes de créditos también se ha utilizado la lógica borrosa. Este campo de investigación ha abierto un nuevo campo de exploración en muchas áreas de conocimiento incluyendo los temas de análisis financieros. Particularmente, algunos autores señalan que es una buena herramienta para obtener resultados excelentes en la evaluación de créditos, Facchinetti (2001). Otras aplicaciones de los sistemas de inferencia difusa en el análisis de crédito pueden encontrarse en Malhotra y Malhotra (2002), Facchinetti *et al.* (2000), Facchinetti *et al.* (2001) y Bojadziev y Bojadziev (1997).

De Facchinetti (2001) se extrae el siguiente sistema experto para el análisis de solvencia.

Figura 2.1. Sistema experto para análisis de solvencia.



Fuente: Facchinetti (2001).

2.4. Aplicaciones de credit scoring con Modelos híbridos y estudios comparados.

Los multiclasificadores, métodos híbridos o métodos de ensemble han demostrado, en la mayor parte de los casos estudiados, que consiguen una mayor precisión que los métodos individuales, (Hung y Chen, 2009; Yu *et al*, 2008).

En un estudio muy reciente, Wang *et al.* (2011) comparan cuatro modelos individuales: Regresión logística, Árboles de decisión, Redes neuronales y máquinas de vectores soporte con tres multiclasificadores: Bagging, Boosting y Sctaking. Estos modelos son aplicados a las muestras de préstamos concedidos de créditos de tres bancos: Un banco alemán con 1.000 registros de clientes, un banco industrial y comercial de China, con ejemplos de 239 compañías y con los registros de un banco de crédito australiano. Estas tres bases de datos se encuentran disponibles en los repositorios de la web y han sido muy utilizadas por investigadores para probar diversos métodos y algoritmos de clasificación.

Los resultados que se obtienen de este estudio muestran que son mejores cuando se emplea el método Bagging, Boosting y Stacking. Bagging ajusta mejor que Boosting, además, Stacking y Bagging tienen un mejor rendimiento en los tres indicadores utilizados: precisión, error de tipo I y error de tipo II. En la tabla1.10. se muestra la precisión de los clasificadores utilizados para los datos de los tres bancos.

Zang *et al.* (2010) desarrollan un clasificador a través del método bagging al que añaden un árbol de decisión (C4.5) consiguiendo mejores resultados que otros modelos con los que comparan esta nueva contribución. Utilizan dos bases de datos del repositorio de la UCI.

Tabla 2.10. Precisión obtenida por los clasificadores (%).

	Banco australiano	Banco de China	Banco alemán
Reg. Logística	86,56	72,07	76,14
Árbol de decisión	84,39	77,85	72,10
Redes Neuronales	83,28	71,12	71,43
Maq. Vect. Soporte	85,67	67,63	76,28
Bagging (RL)	86,64	74,18	76,08
Bagging (AD)	86,30	81,07	74,92
Bagging (RN)	85,01	76,54	75,57
Bagging (MVS)	85,71	71,06	75,93
Boosting (RL)	85,56	75,12	76,14
Boosting (AD)	85,22	80,52	72,77
Boosting (RN)	83,65	71,88	73,30
Boosting (MVS)	84,19	70,37	76,30
Stacking	86,57	78,60	75,97

Fuente: Adaptado de Wang et al. (2011).

Otra forma de agregar modelos se señala en Bonilla *et al* (2003). Estos autores utilizan la base de datos presentada por Quinlan (1987) que contiene 690 registros con 14 características de los demandantes de una tarjeta de crédito, así como su comportamiento posterior a la concesión. Utilizan siete modelos, dos paramétricos: Análisis discriminante /AD) y regresión logística (LOGIT) y cinco modelos no paramétricos: dos árboles de decisión y regresión CART y C4.5, Regresión Lineal Ponderada (RLP), Splines de regresión adaptativa multivariante y una red neuronal artificial (RNA).

Para la estimación de los modelos simples utilizan el método de validación cruzada para encontrar la estructura óptima de cada uno de los modelos individuales. Se obtienen las predicciones de los mejores modelos para el conjunto de validación. Se trata de elegir una función de pérdida y encontrar el método híbrido que minimice dicha función a lo largo de los conjuntos de validación. Los resultados obtenidos se detallan en el cuadro siguiente. Una vez calculado el híbrido se computa el error de predicción sobre el conjunto de datos de test.

A continuación se muestran los errores cometidos por los siete modelos con el conjunto de datos de entrenamiento y de test.

Tabla 2.11. Tabla resumen de modelos de credit scoring.

	Errores de validación cruzada		**Errores de predicción**	
Modelos	Entrenamiento	Test	Entrenamiento	Test
AD	14,17%	13,99%	14,17%	12,22%
Logit	12,18%	14,12%	12,50%	12,22%
CART	14,6%0	14,60%	14,6%0	13,30%
RLP	12,45%	13,05%	12,33%	12,35%
RNA	12,20%	13,30%	12,33%	10,00%
C4.5	14,10%	14,80%	14,70%	13,30%
MARS	11,52%	13,33%	11,17%	12,22%

Fuente: Adaptado de Bonilla et al (1999).

En esta tabla se ve claramente que la red neuronal es la que mayor precisión alcanza. Al final del artículo añaden que “en lo que respecta al proceso de decisiones, es posible que un método que combine las predicciones de los modelos individuales podría resultar más adecuado en el problema que estamos analizando, Olmeda y Fernández, (1997) y Kumar y Olmeda, (1999).

Otro estudio basado en datos reales de una caja de ahorros española es el de Beltrán *et al.* (2011). Los métodos empleados en la clasificación de créditos son los siguientes: regresión logística, máquinas de vectores soporte, dos modelos de redes neuronales, el C.4.5 como árbol de clasificación, y también el algoritmo Metacost con y sin matriz de costes. Los resultados de todos los modelos son comparados con los que se obtienen a través de seis métodos multiclasificadores El multiclasificador Stacking se configura con cinco modelos: perceptrón multicapa, red bayesiana con el algoritmo de búsqueda K2, regresión logística, máquinas de vectores soporte y el árbol de clasificación, C4.5. En este estudio se observa como la combinación de clasificadores junto con los clasificadores que utilizan las redes bayesianas logran los mejores resultados.

2.5. Resumen y conclusiones del estado del arte de credit scoring.

Una vez analizadas las principales referencias bibliográficas, es necesario realizar una breve síntesis en relación con el objetivo de esta tesis.

Las principales bases de datos utilizadas por la gran mayoría de trabajos encontrados hacen uso, principalmente, de dos bases de datos referidas a clientes de un banco alemán y otro australiano. Estos datos son públicos y están disponibles en el repositorio de la Universidad Irvine de California, Blake et al. (1998). Otras dos bases de datos empleados con cierta frecuencia se pueden encontrar en Lyn *et al.* (2002) y en Quinlan (1987).

En la presente investigación se ha empleado una base de datos real perteneciente a una Caja de Ahorros de la Rioja, actualmente integrada en la Banca Comercial.

En los resultados que se presentan en los trabajos analizados se observa, en general, unas buenas predicciones de la clase relacionada con la devolución del crédito mientras que la predicción de la clase a la que están asociados los clientes que no devuelven el crédito no obtienen buenos resultados. Esto es una consecuencia de que las muestras utilizadas están desequilibradas. Este aspecto es fundamental para obtener buenos resultados, no sólo en la clasificación sino también para la selección del método estadístico que realiza la predicción. Esta es la razón por lo que en esta tesis se propone un método estadístico que nos ayuda a obtener muestras equilibradas de manera eficiente.

Otra cuestión importante es que, salvo en escasos estudios, los autores de los artículos analizados no realizan una adecuada selección de variables. En algunos casos las bases de datos utilizadas no contienen muchos atributos pero en otros sí. La selección de atributos es fundamental para el desarrollo de modelos precisos. En esta tesis doctoral se lleva a cabo una selección de atributos óptima desde la óptica de la construcción de modelos bayesianos

En el conjunto de los trabajos predominan aquellos que aplican métodos paramétricos clásicos como son las modelos Logit, Probit y el análisis discriminante, sin embargo, los trabajos relacionados con los métodos bayesianos y los multiclasificadores, estrechamente relacionados con el objetivo de esta tesis, no son abundantes. En el desarrollo de esta investigación se demuestra que los modelos paramétricos clásicos, aunque obtienen buenos resultados, no son los óptimos.

Respecto a las variables utilizadas en las investigaciones son muy similares cuando se trabaja con bancos de datos relacionados con créditos personales, comerciales o hipotecarios; en general utilizan variables internas del propio banco y variables socioeconómicas. Cuando las bases de datos están relacionadas con créditos a empresas, se emplean ratios contables calculados de los datos recogidos en el Balance y en la Cuenta de Resultados. El análisis comparativo de estos estudios se dificulta al emplear cada autor su propia combinación del conjunto de ratios: estructura del activo e inversión, estructura de pasivo, equilibrio financiero, liquidez, capacidad de endeudamiento, rentabilidad, productividad y eficiencia, crecimiento, riesgo, etcétera.

CAPÍTULO 3

EL PROCESO DE EXTRACCIÓN DE CONOCIMIENTO ÚTIL. ASPECTOS METODOLÓGICOS Y PRINCIPALES TÉCNICAS UTILIZADAS EN LA MINERÍA DE DATOS.

3. El proceso de extracción de conocimiento útil. Aspectos metodológicos y principales técnicas utilizadas en la minería de datos.

3.1. Diferentes aspectos metodológicos relacionados con la minería de datos.

En este epígrafe, antes de empezar a describir los diferentes métodos y algoritmos de clasificación, enmarcamos el problema de credit scoring dentro de un proceso de investigación de minería de datos. Se ofrecen diferentes perspectivas de su definición y se abordan brevemente dos metodologías generales, dado que en el capítulo cuatro se explica ampliamente la metodología aplicada en esta tesis doctoral. Aun así y antes de avanzar en este capítulo, se exponen algunos conceptos muy importantes que facilitan la comprensión de todo el proceso: equilibrado de las muestras a través del método del cubo, muestras de entrenamiento y de test, discretización de variables cuantitativas y la evaluación de modelos a través de diferentes medidas basadas en las métricas que resultan de la matriz de confusión, de la curva ROC y a través de la introducción de los costes asociados a la clasificación.

3.1.1. Introducción al Data Mining.

El proceso de credit scoring puede ser abordado como un proceso de minería de datos.

El objetivo de la minería de datos es descubrir estructuras subyacentes escondidas en las bases de datos. El conjunto de herramientas incorporadas en el Data Mining se ha revelado muy eficiente, fundamentalmente, cuando el volumen de los datos y las variables implicadas son significativamente voluminosos.

Data Mining es un proceso interactivo que combina la experiencia sobre un problema dado con variedad de técnicas tradicionales de análisis de datos y tecnología avanzada de aprendizaje automático, con el objetivo de descubrir patrones y relaciones en los datos para la realización de predicciones válidas.

La minería de datos es una parte de un proceso más general que se denomina **Descubrimiento de conocimiento en las bases de datos** (Knowledge Discovery in Databases o KDD), si bien, en la mayor parte de la bibliografía sobre el tema el concepto de Data Mining toma el significado global del proceso.

Una definición más general de la minería de datos es referirse a ella como la extracción no trivial de la información implícita, previamente desconocida y potencialmente útil, a partir de los datos.

El enfoque de estas nuevas técnicas queda magníficamente descrito por L. Breiman[4] donde sus provocadoras palabras, si bien están circunscritas en el contexto de la estadística, serían asumibles en cualquier otra área del conocimiento: "... *(son trabajos basados) en el hecho de que pueden ser una metodología útil. Todo es ad hoc, no hay máxima verosimilitud, ni minimax, ni velocidad de convergencia, ni distribuciones, ni teoría de funciones. ¿No hay nada sagrado? ¿Qué tipo de ciencia estadística es ésta? Mis felicitaciones al editor y a todos los demás implicados en esta sacrílega desviación... El desarrollo y la utilización de métodos multivariantes eficaces para ajustar datos complejos es un esfuerzo que se lleva a cabo en muchas ocasiones fuera de la estadística por grupos ínter disciplinares interesados por los resultados más que en los teoremas... El rápido crecimiento de las redes neuronales se construye en torno a una nueva clase de algoritmos para regresión y clasificación multivariante* siendo sus protagonistas principales ingenieros y especialistas en cómputo".

El Data Mining siempre intenta descubrir los patrones, perfiles y tendencias presentes y significativas ocultas en los datos trabajando con tecnologías de reconocimientos de patrones, como las redes neuronales, máquinas de aprendizaje, algoritmos genéticos, etcétera.

La minería de datos puede dar respuesta a preguntas que se plantean muy a menudo en los negocios bancarios; preguntas tales como: ¿Quiénes son mis mejores clientes?, ¿Cómo aumentar mi cuota de mercado?, ¿Quiénes son los visitantes de mi sitio web? o ¿Cómo fidelizar a mis clientes?

También se puede afirmar sobre las técnicas de Data Mining o minería datos que son un conjunto de algoritmos matemáticos y estadísticos, a veces, de enorme complejidad, que permiten descubrir y cuantificar relaciones predictivas ocultas en los datos, transformando la información disponible en conocimiento útil de negocio. Es un procedimiento automatizado, a caballo entre la información y la toma de decisiones, por parte de la dirección de la organización. Las técnicas de Data Mining permiten descubrir patrones no visibles en los datos, de manera que el analista deja trabajar al algoritmo y éste le muestra los patrones encontrados. De esta forma, el experto de negocio selecciona aquéllos que le son útiles. Se trata, pues, de un conjunto de

[4] Breiman, L. (82-90) en la discusión de : Friedman, J. H. Multivariate Adaptative Regression Splines. Invited Paper. The Annals of Statistics, vol 19, nº 1 (1991) 1-141.

algoritmos inteligentes que aplicados a un conjunto de datos sirven de soporte para la toma de decisiones.

Data Mining surge ante la necesidad de análisis de grandes bases de datos, incorporando técnicas capaces de estimar modelos con cientos de variables y miles de parámetros. Aún a pesar de la automatización del proceso de modelado con algoritmos de aprendizaje automático, se hace necesario el criterio del experto en el campo sustantivo, antes y después de la obtención del modelo. El uso de técnicas de modelado procedentes de la Estadística se puede enmarcar sin problemas en el proceso Data Mining, siendo apropiadas en el análisis de matrices de datos relativamente pequeñas.

Ningún método por sí solo soluciona la mayoría de problemas; el éxito del Data Mining requiere de la disponibilidad de un amplio repertorio de herramientas, tanto clásicas como innovadoras.

En definitiva, Data Mining es un proceso analítico eminentemente exploratorio y con un objetivo claro: la detección de nuevas estructuras en los datos, reflejadas en modelos que aporten algo más de luz sobre la influencia que tienen sobre el entorno los diferentes elementos que pertenecen a la realidad bajo estudio.

Las relaciones de la minería de datos con la estadística, la inteligencia artificial, las bases de datos, las técnicas de recuperación de la información y otros campos del conocimiento son evidentes. Muchos de sus algoritmos, procedimientos, técnicas y metodología son iguales en esencia, aunque, muchas veces, se denominen con otros nombres. No obstante, el Data Mining presenta características útiles e interesantes que lo hacen atractivo. De forma general, se puede afirmar que para la resolución de problemas, muy a menudo complejos, se utilizan varios procedimientos dada la complementariedad de las técnicas en la solución de los mismos.

Figura 3.1. Aportación de diferentes disciplinas a la minería de datos.

Fuente: Elaboración propia.

3.1.2. Metodologías generales utilizadas en el proceso del Data Mining.

La gran cantidad de datos que se crean en las actividades que desarrollamos como seres humanos se encuentran almacenados en grandes bases datos y se multiplican rápidamente. Esto es cierto para casi todos los sectores de la actividad, pero aún es más cierto en las entidades financieras cuyo negocio diario genera una ingente cantidad de información. Sin embargo, se puede afirmar que este incremento espectacular de datos almacenados no ha supuesto un significativo aumento de la información disponible para la empresa (el bosque no deja ver los árboles) ni, en general, se ha traducido en un incremento de la rentabilidad de las compañías financieras.

En los últimos años se ha desarrollado, sobre todo en las entidades bancarias, el concepto de CRM (Customer Relationship Management, Gestión de la Relación con el Cliente), tanto en su forma analítica como operacional. El Data Mining optimiza y se integra en el sistema de CRM aumentando el beneficio de las compañías que lo implementan.

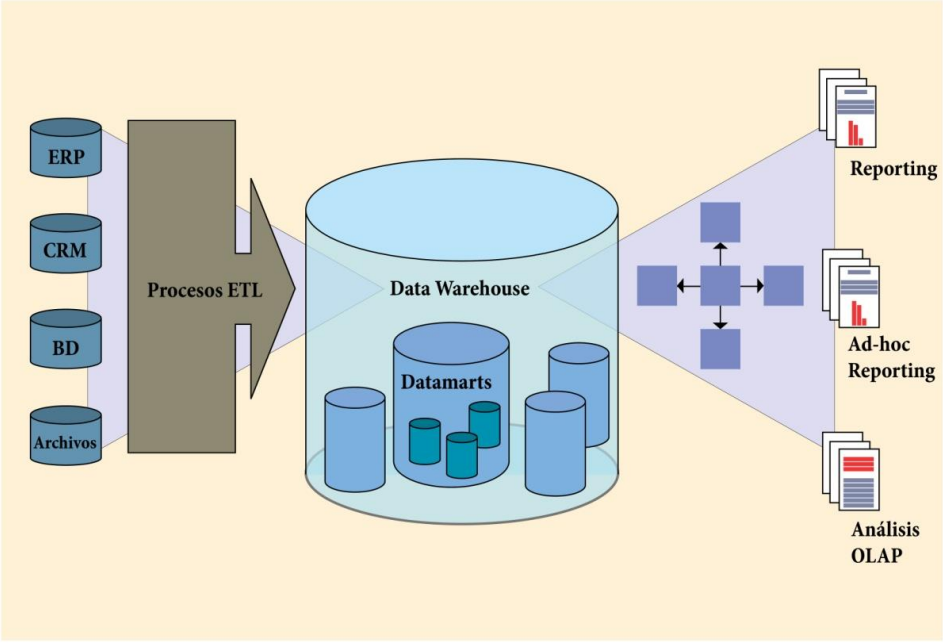
Un proyecto de minería de datos parte, como cualquier otro proyecto, de un anteproyecto donde se especifican, entre otras cosas, la formulación del problema y de los objetivos que se quieren conseguir.

Es obvia la importancia de disponer de un gran almacén de datos y de las ventajas que supone organizar toda la información disponible susceptible de ser analizada a través de un Datawarehouse.

Según la definición clásica, un Data Warehouse es una base de datos corporativa que se caracteriza por integrar y depurar información de una o más fuentes, para luego procesarla permitiendo su análisis desde infinidad de perspectivas y con grandes velocidades de respuesta. La creación de un Data Warehouse representa en la mayoría de las ocasiones el primer paso, desde el punto de vista técnico, para implantar una solución completa y fiable de Minería de Datos.

La ventaja principal de este tipo de bases de datos radica en las estructuras en las que se almacena la información (modelos de tablas en estrella, en copo de nieve, cubos relacionales... etc.). Este tipo de persistencia de la información es homogénea y fiable, y permite la consulta y el tratamiento jerarquizado de la misma (siempre en un entorno diferente a los sistemas operacionales).

Figura 3.2. Esquema de un Data Warehouse.



El término Data Warehouse fue acuñado por primera vez por Bill Inmon, y se traduce literalmente como almacén de datos. No obstante, y como cabe suponer, es mucho más que eso. Según definió el propio Bill Inmon, un Data Warehouse se caracteriza por ser:

Integrado: los datos almacenados en el Data Warehouse deben integrarse en una estructura consistente, por lo que las inconsistencias existentes entre los diversos sistemas operacionales deben ser eliminadas. La información suele estructurarse también en distintos niveles de detalle para adecuarse a las distintas necesidades de los usuarios.

Temático: sólo los datos necesarios para el proceso de generación del conocimiento del negocio se integran desde el entorno operacional. Los datos se organizan por temas para facilitar su acceso y entendimiento por parte de los usuarios finales. Por ejemplo, todos los datos sobre clientes pueden ser consolidados en una única tabla del Data Warehouse. De esta forma, las peticiones de información sobre clientes serán más fáciles de responder dado que toda la información reside en el mismo lugar.

Histórico: el tiempo es parte implícita de la información contenida en un Data Warehouse. En los sistemas operacionales, los datos siempre reflejan el estado de la actividad del negocio en el momento presente. Por el contrario, la información almacenada en el Data Warehouse sirve, entre otras cosas, para realizar análisis de tendencias. Por lo tanto, el Data Warehouse se carga con los distintos valores que toma una variable en el tiempo para permitir comparaciones.

No volátil: el almacén de información de un Data Warehouse existe para ser leído, pero no modificado. La información es por tanto permanente. De modo que, la actualización del Data Warehouse significa la incorporación de los últimos valores que tomaron las distintas variables contenidas en él sin ningún tipo de acción sobre lo que ya existía.

El proceso de Data Mining, como metodología de investigación, es ofrecida por varias empresas. Las dos metodologías más conocidas son las siguientes:

- CRISP-DM: ofrecida entre otras, además de por SPSS, por las firmas Daimler-Benz.
- SEMMA: de la casa SAS.

3.1.2.1. Metodología CRISP.

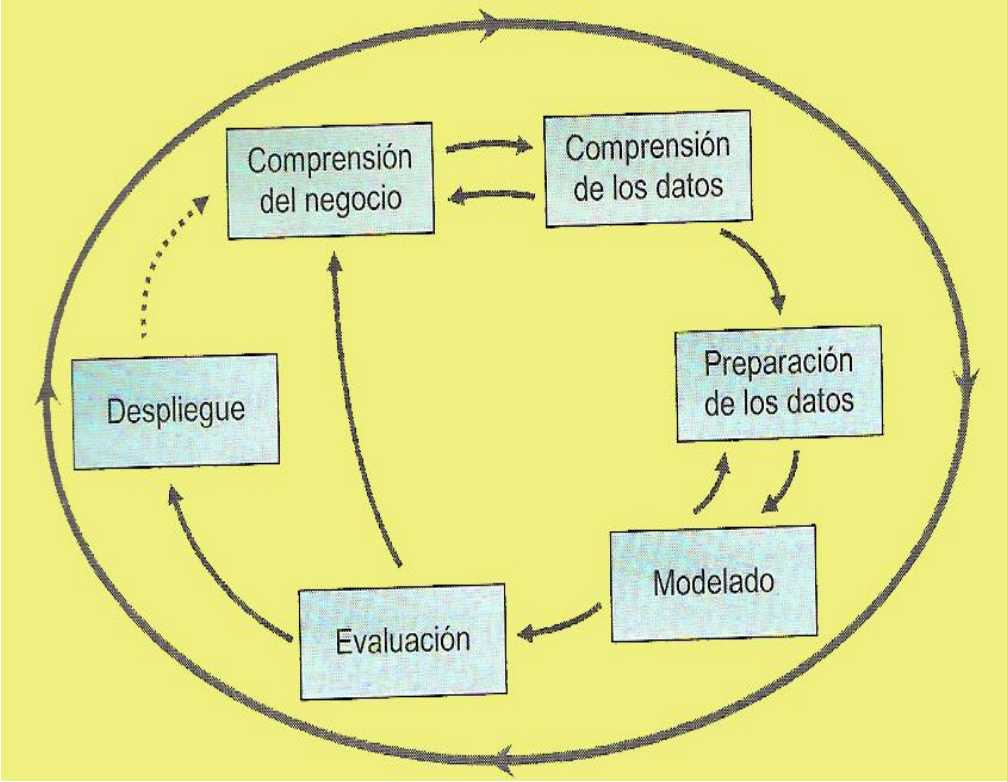
La primera metodología CRISP-DM que es el acrónimo de Cross Industry Standard Process for Data Mining. No es un software, sino una metodología y es un proyecto fundado por la Comisión Europea a través del proyecto ESPRIT en colaboración con varias empresas (IBM-SPSS, Daimler – Bentz y NCR).

Esta metodología define un modelo universal para aplicar proyectos de DM (algo parecido a un AENOR para el Data Mining). Una amplia información se puede encontrar en Chapman (2002).

Esta metodología contiene seis etapas que pretenden gestionar de forma global un proyecto de minería de datos. Las fases son las siguientes: Compresión del negocio, Comprensión de los datos, Preparación de los datos, Modelado, Evaluación y Despliegue.

El estándar incluye un modelo y una guía, estructurados en seis fases donde algunas de estas fases son bidireccionales, lo que significa que en algunas de ellas es factible que se puedan revisar parcial o totalmente los procedimientos anteriores. En el gráfico siguiente se observan las diferentes fases y los subproductos de cada fase. Un proyecto de investigación aplicando esta metodología a un proceso industrial puede estudiarse en Martínez de Pisón (2003).

Figura 3.3. Fases de CRISP – El Proceso de Data Mining de SPSS-IBM.



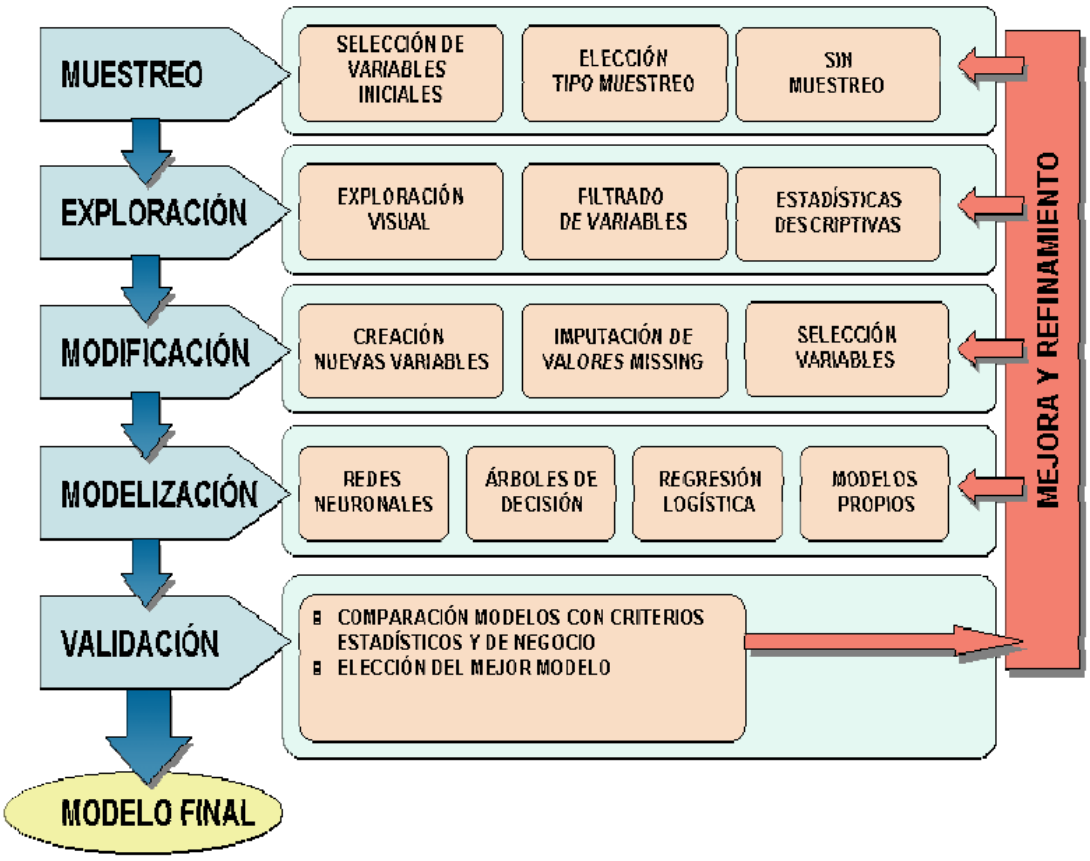
Fuente: SPSS-IBM.

3.1.2.2. Metodología SEMMA.

La compañía multinacional SAS, líder mundial en soluciones de Business Intelligence propone una metodología genérica para el análisis de datos que denomina SEMMA. Los grandes apartados que cubren esta metodología son los siguientes: Muestreo. (Sampling), Exploración. (Exploration), Modificación. (Modification), Modelización. (Modelization), Validación. (Asessment), Aplicación de resultados.

En la figura siguiente se muestran de forma sintética las principales operaciones que se realizan en cada una de las fases de la metodología de la casa.

Figura 3.4. Fases de SEMMA – El Proceso de Data Mining de SAS.



Fuente: SAS Enterprise.

3.1.3. Equilibrado de la muestra.

3.1.3.1. Introducción al balanceo de muestras.

A la hora de aplicar los métodos de clasificación hemos de tener en cuenta cómo están distribuidas las instancias respecto a la clase. Al no estar balanceadas las clases

los clasificadores estarán sesgados a predecir un porcentaje más elevado a la clase más favorecida. Este problema hay que resolverlo a través de un remuestreo de la base de datos.

La técnica más sencilla de sobremuestreo es la aleatoria simple a través de la réplica de ejemplos en la misma clase, pero este método puede ocasionar un alto sobreajuste de los clasificadores.

Como técnica más inteligente para incrementar los ejemplos de la clase minoritaria se encuentra el algoritmo SMOTE (Synthetic Minoritary Over-sampling TEchnique) originario de Chawla y otros (2002). En este método la creación de nuevas muestras se origina a través de la interpolación. En un primer paso elegimos los K vecinos más cercanos y que pertenecen a su misma clase. Posteriormente elegimos el número de muestras artificiales que se generarán y, finalmente, para generar una nueva muestra, se calcula la diferencia entre el vector de atributos bajo consideración y uno de los vecinos más cercanos de los k vecinos elegidos al azar. El resultado de la diferencia se multiplica por un valor aleatorio entre cero y uno.

El algoritmo SMOTE se ha modificado de diferentes maneras para adaptarse mejor a muchos ejemplos. Algunas de estas aportaciones son las efectuadas por Han et al (2005) que proponen el algoritmos Borderline-SMOTE para generar ejemplos positivos cercanos a una frontera. Wang et al (2006) presentan el algoritmo LLE-SMOTE (Locally Linear Embedding) que proyecta conjuntos de alta dimensionalidad a otro de menor dimensionalidad. En este espacio de reducida dimensionalidad es donde se aplica SMOTE y después los ejemplos generados son transformados a su espacio de representación original.

Otras formas de obtener una representación mayor de la clase minoritaria se basan en técnicas de agrupamiento, por ejemplo Japkowicz (2001) emplea el algoritmo de clustering k-medias sobre cada clase por separado. Los clusters resultantes se sobremuestrean aleatoriamente hasta conseguir un equilibrio entre las clases. Otro trabajo en esta línea de investigación es el de Cohen et al (2006) que también explora la generación de nuevas instancias a través de algoritmos de clustering, pero en este caso los centroides de los clusters se obtienen a través de un algoritmo aglomerativo jerárquico.

En cuanto a las técnicas de submuestreo una de las primeras propuestas para editar o filtrar las muestras de entrenamiento fue el algoritmo de Edición de Wilson (1972),

también conocido como la regla del vecino más cercano editado (Edited Nearest Neighbor). Actualmente existen muchas formas de proceder, algunas de ellas son las siguientes: a través del submuestreo aleatorio de Ho et al (2004), con submuestreo dirigido, algoritmo One-sides selection de Kubat y Matwin, (1997), con técnicas de vecindad, algoritmo Neighbordhood Cleaning Rule de Laurikkala, (2002). con submuestreo aplicando algoritmos genéticos, Kuncheva y Jain, 1999, con submuestreo por distancia, Zhanng y Mani, (2003), con submuestreo por clustering, Cohen et al, (2006), a través del aprendizaje activo de Provost, (2003). Respecto a los métodos de clasificación en entornos no balanceados que no cambian la distribución a priori de las clases nos encontramos con las soluciones a nivel de algoritmos: aprendizaje sensible al coste, algoritmos de clasificación con sesgo hacia la clase minoritaria y los clasificadores de una clase.

En esta investigación los resultados de los diferentes clasificadores que se presentan se aplican a un conjunto de datos que se han balanceado a través de un método mixto donde se aplica el método SMOTE a la clase minoritaria y se reduce la muestra de la clase mayoritaria a través del método del submuestreo equilibrado del cubo, propuesto por Deville y Tillé (2004). Este método de muestreo es el único que nos permite seleccionar una muestra equilibrada sobre variables auxiliares con probabilidades de inclusión iguales o no. El método del cubo selecciona únicamente las muestras cuyos estimadores de Horvitz-Thompson son iguales a los totales de las variables auxiliares conocidas.

Debido a la importancia en el equilibrado de la muestra, el método del cubo se describirá de forma exhaustiva en el apartado siguiente.

3.1.3.2. Método del Cubo.

La construcción de diseños estratificados es a menudo un ejercicio difícil, especialmente cuando se pretende estratificar usando un número elevado de variables cualitativas.

En muchos casos, se tiende a proceder cruzando todos los estratos de todas las variables, lo que hará que muchas de las celdas sean demasiado pequeñas para seleccionar muestras en ellas. Para solucionar este problema actualmente se utiliza el denominado muestreo equilibrado, que puede ser visto como un tipo de calibración directamente integrada en el diseño muestral.

Considérese S una muestra de tamaño n, definida como un subconjunto de una población finita U de tamaño N. Esta muestra será equilibrada si para un vector de variables auxiliares $x_k = (x_{k1}, \dots, x_{kj}, \dots, x_{kp})'$ se verifica:

$$\frac{1}{n}\sum_{k\in S} x_k = \frac{1}{N}\sum_{k\in U} x_k \equiv \frac{N}{n}\sum_{k\in S} x_k = \sum_{k\in U} x_k \qquad (3.1)$$

Lo cual significa que las medias (totales) en la muestra de las x-variables coinciden con las de la población. En otras palabras, en una muestra equilibrada el total de las x-variables son estimadas sin error.

Si la muestra S es aleatoria, se deben además satisfacer las siguientes ecuaciones de equilibrio:

$$\sum_{k\in S} \frac{x_k}{\pi_k} = \sum_{k\in U} x_k \qquad (3.2)$$

donde π_k son las probabilidades de inclusión en la muestra asociadas a cada elemento de la población.

En una muestra equilibrada y aleatoria, han de cumplirse ambas ecuaciones de equilibrio.

Si $x_k = 1\ \Delta k \epsilon U$, entonces la ecuación de equilibrio nos queda:

$$\sum_{k\in S} \frac{1}{\pi_k} = \sum_{k\in U} 1 = N \qquad (3.3)$$

donde $\sum_{k\in S} \frac{1}{\pi_k}$ es el estimador de Horvitz-Thompson del tamaño de la población N. Esta igualdad se cumple en los muestreo con equiprobabilísticos, pero es evidente que no en aquellos con probabilidades desiguales.

Cabe señalar, que el muestreo estratificado no deja de ser un caso particular de muestreo equilibrado. Así, supóngase que la población U está particionada en L estratos, $U_h, h = 1, \dots, L$, con tamaños poblacionales N_h, y que en cada uno de ellos seleccionamos una muestra aleatoria simple de tamaño n_h. En este caso, las variables auxiliares (o variables de equilibrio) serían los indicadores de pertenencia al estrato, es decir:

$$y_{kh} = \begin{cases} 1 & si\ k\ \in\ U_h \\ 0 & en\ otro\ caso \end{cases} \tag{3.4}$$

Bajo este diseño, los estimadores de Horvitz-Thompson de los tamaños poblacionales de los estratos coinciden con el valor real $\mathrm{N_h}$, lo cual en esencia es la condición de equilibrio tomando como variables auxiliares al vector $\mathrm{y_k}$. Las ecuaciones de equilibrio serán entonces:

$$\sum_{k\in S} \frac{N_h y_{kh}}{n_h} = \sum_{k\in U} y_{kh} \quad con\ h = 1, \dots, L \tag{3.5}$$

En resumen, el equilibrado de la muestra se realiza sobre los totales marginales y no sobre cada una de las celdas contenidas en una tabla de contingencia. Sin embargo, la teoría habitual de estratificación permite estratos superpuestos, ya que la estratificación debe ser una partición de la población. El método del cubo permite además equilibrar directamente en los totales de superposición de estratos, simplemente utilizando los indicadores de los estratos como variables de equilibrio.

El método del cubo se configura como una clase de algoritmos de muestreo que seleccionan muestras equilibradas considerando el conjunto de probabilidades de inclusión definidas. Se basa en una transformación aleatoria del vector de probabilidades de inclusión hasta que se obtiene una muestra tal que:

- Se cumplen exactamente las probabilidades de inclusión.
- Se cumplen lo más exactamente posible las ecuaciones de equilibrio en las variables auxiliares (expresión 2).

Las ecuaciones definidas en la expresión 2 raramente podrán cumplirse de forma exacta. Esto se conoce como problema de redondeo y viene derivado de que la selección de la muestra es un problema entero. Para ver esto, considérese una muestra aleatoria extraída de una población tal que:

- x_k son números enteros $\Delta k \epsilon U$
- $\pi_k = \frac{1}{2}\ \Delta k \epsilon S\ \rightarrow\ \frac{N}{n} = 2$
- $\sum_{k\epsilon U} x_k$ es un número impar

Entonces, según las ecuaciones de equilibrio:

$$2 \sum_{k\in S} x_k = \sum_{k\in U} x_k \tag{3.6}$$

Como vemos, el lado izquierdo de la ecuación será siempre un número par, mientras que el lado derecho es, por definición, un número impar, lo cual implica que no existirían muestras de tamaño N/2 exactamente balanceadas.

El nombre del método proviene de la representación geométrica de un diseño de muestreo. En efecto, una muestra puede ser representada por un vector de indicadores muestrales de la siguiente manera:

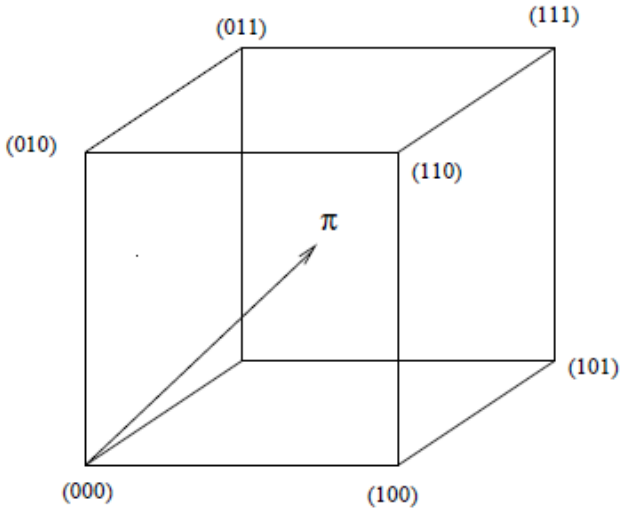
$$s = (I[1\epsilon s], I[2\epsilon s], \dots, I[k\epsilon s], \dots, I[N\epsilon s])'$$

donde

$$I[k\epsilon s] = \begin{cases} 1 & si \;\; k \,\epsilon\, s \\ 0 & si \;\; k \notin s \end{cases}$$

Entonces, una muestra puede ser vista como un vértice de un cubo N-dimensional tal como se muestra en la siguiente figura:

Figura 3.5. Posibles muestras en una población con N=2.



Fuente: Deville y Tillé (2004).

Sea p(s) = Pr(S=s) el diseño de muestreo o probabilidad de que la muestra s sea seleccionada, donde S es la muestra aleatoria y n(S) el tamaño de S. Bajo este esquema, la esperanza matemática de s se define como:

$$E(s) = \sum_{s\epsilon S} p(s)s = \pi \qquad (3.7)$$

donde π representa el vector de probabilidades de inclusión. Teniendo en cuenta esto, las ecuaciones de equilibrio definidas en la expresión 3.2 pueden escribirse como:

$$\sum_{k\epsilon U} \check{x}_k s_k = \sum_{k\epsilon U} \check{x}_k \pi_k \quad \text{para } s_k = 0,1 \;\; k\epsilon U \tag{3.8}$$

donde $\check{x}_k = \frac{x_k}{\pi_k} \; \Delta k\epsilon U$

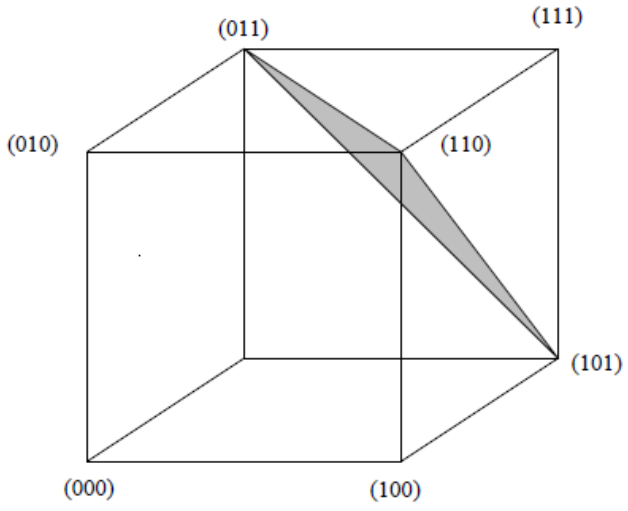
El sistema de ecuaciones definido en la expresión 3.8, con valores desconocidos s_k, define el siguiente subespacio afín en R^N de dimensión *N - p*:

$$Q = \{u\epsilon\mathbb{R}^N / \sum_{k\epsilon U} \check{x}_k u_k = \sum_{k\epsilon U} x_k\} \tag{3.9}$$

Esto nos permite reformular el problema de selección de una muestra balanceada como la elección de un vértice de un cubo N-dimensional sobre el subespacio lineal Q.

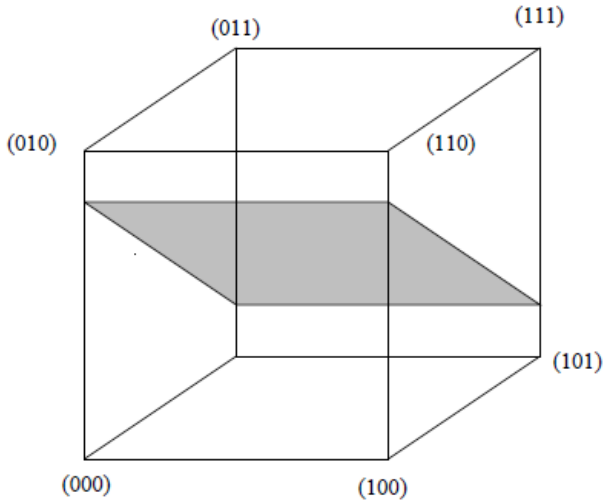
La Figura 3.6 muestra la representación gráfica del problema de la elección de una muestra de tamaño n=2 sobre una población N=3 y la Figura 3.7 el problema de redondeo descrito anteriormente.

Figura 3.6.: Posibles muestras en una población de tamaño N=3 con una restricción de tamaño de muestrsa n=2.



Fuente: Deville y Tillé (2004).

Figura 3.7. Posibles muestras en una población de tamaño N=3 con una restricción que genera un problema de redondeo.



Fuente: Deville y Tillé (2004).

El método del cubo ,Deville y Tille, (2004) se divide en dos fases: la fase de vuelo y de la fase de aterrizaje.

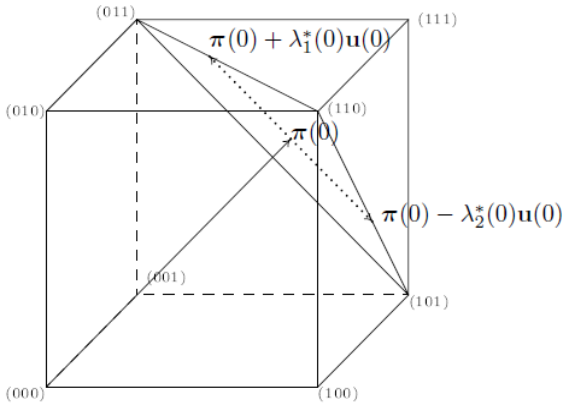
La fase de vuelo es un paseo aleatorio que comienza en el vector de probabilidades de inclusión y permanece en la intersección del cubo y el subespacio de restricciones. Este paseo aleatorio se detiene en un vértice de la intersección del cubo y el subespacio de restricciones. Al final de la fase de vuelo, si no se obtiene una muestra, la fase de aterrizaje determinará una muestra que está tan cerca como sea posible del subespacio de restricciones.

3.1.3.2.1. Fase de vuelo.

En primer lugar debemos elegir un vector $\mathrm{u}(0)$ de tal manera que $\pi + \mathrm{u}(0)$ permanezca en el subespacio de restricciones. De hecho, el método del cubo es realmente una familia de métodos que dependen de la manera en que $\mathrm{u}(0)$ sea elegido, pudiendo hacerse aleatoriamente o no.

Si desde π avanzamos en dirección de $\mathrm{u}(0)$ cruzaremos necesariamente una cara del cubo (representado por $\pi(0) + \lambda_1^*(0)\mathrm{u}(0)$ en la figura 3.8) y, análogamente, si avanzamos en dirección opuesta también cruzaremos otra cara del cubo ($\pi(0) - \lambda_2^*(0)u(0)$ en figura 3.8).

Figura 3.8. Fase de vuelo en una población de tamaño N=3 con una restricción de tamaño de muestra n=2.



Fuente: Deville y Tillé (2004).

En el primer paso, el vector $\pi(0) = \pi$ será modificado aleatoriamente a:

$$\pi(1) = \begin{cases} \pi(0) + \lambda_1^*(0)u(0) & con\ prob = p(0) \\ \pi(0) - \lambda_2^*(0)u(0) & con\ prob = 1 - p(0) \end{cases} \quad (3.10)$$

donde *p(0)* es determinado de tal manera que $\mathrm{E}[\pi(1)] = \pi(0)$, es decir, no se modifican las probabilidades de inclusión establecidas a priori en base al diseño de la muestra.

Una vez finalizado este primer paso de la fase de vuelo, hemos saltado a una de las caras del cubo, lo que significa que al menos una de las componentes de $\pi(1)$ es igual a 0 ó 1, es decir, el problema de muestreo se ha reducido de una población N=3 a un problema con N=2

Analíticamente la forma de proceder sería la siguiente:

1. Consideramos $\pi(0) = \pi$.
2. Para t = 0 hasta t=T:
   a. Generamos un vector $u(t) = [u_k(t)] \neq 0$ de tal forma que:
      - $u(t)$ está en el núcleo de la matriz $A = (x_1/\pi_1, \dots, x_k/\pi_k, \dots, x_N/\pi_N)$, es decir, $Au(t) = 0$.
      - $u_k(t) = 0$ si $\pi_k(t)$ es un número entero (0 ó 1).
   b. Calcular $\lambda_1^*(t)$ y $\lambda_2^*(t)$ como los mayores valores tales que:
      - $0 \leq \pi(t) + \lambda_1(t)u(t) \leq 1$.
      - $0 \leq \pi(t) - \lambda_2(t)u(t) \leq 1$.
   c. Calcular $\pi(\mathrm{t}+1) = \begin{cases} \pi(\mathrm{t}) + \lambda_1^*(\mathrm{t})\mathrm{u}(t) & \mathrm{con\ prob} = \mathrm{p(t)} \\ \pi(\mathrm{t}) - \lambda_2^*(\mathrm{t})\mathrm{u}(\mathrm{t}) & \mathrm{con\ prob} = 1 - \mathrm{p(t)} \end{cases}$

donde $\mathrm{p(t)} = \frac{\lambda_2^*(\mathrm{t})}{\lambda_1^*(\mathrm{t})+\lambda_2^*(\mathrm{t})}$

La fase de vuelo finalizará cuando no sea posible encontrar un vector $u(t) \neq \mathbf{0}$.

3.1.3.2.2. Fase de aterrizaje.

Si al final de la fase de vuelo las ecuaciones de equilibrio no son exactamente satisfechas, entonces debemos aplicar la fase de aterrizaje.

Sea π^* el último vector $\pi(\mathrm{t}+1)$ obtenido en la fase de vuelo. Si definimos el subesespacio:

$$U^* = \{k \epsilon U / \ 0 < \pi_k^* < 1\} \tag{3.12}$$

Es posible demostrar, Deville y Tille, (2004), que:

$$card(U^* \leq p)$$

donde p es el número de variables de equilibrio. El objetivo de la fase de aterrizaje es encontrar una muestra s, tal que:

$$E\left({}^{S}/_{\pi^*}\right) = \pi^* \tag{3.13}$$

la cual sea casi-equilibrada.

Existen dos formas de proceder:

1. La *fase de vuelo de programación lineal,* que consiste en considerar todas las posibles muestras de U^*. Esto exige asignar un coste a cada muestra. Lo más lógico sería considerar la distancia entre la muestra y el subespacio de restricciones. A continuación, se busca un diseño de muestra en U^* que minimice el coste esperado y que satisfaga las probabilidades de inclusión obtenidas en π^*. Este problema puede resuelto porque el número de muestras a considerar es razonable debido al pequeño tamaño de U^*.
2. La *fase de vuelo por supresión de variables* se puede utilizar cuando el número de variables de equilibrio es demasiado grande para que el programa lineal pueda ser resuelto por un algoritmo simplex (p>20). Con este método, al final de la fase de vuelo se elimina una variable auxiliar. A continuación, retornamos a la fase de vuelo hasta que no sea posible "moverse" en el subespacio de

restricciones. Las restricciones son entonces “relajadas” sucesivamente de acuerdo a un orden de preferencia.

La entropía del diseño de la muestra depende de la forma en que los vectores $u(t)$ son elegidos durante la fase de vuelo. Con el fin de aumentar la entropía, el vector $u(t)$ puede ser elegido aleatoriamente, o la población puede reordenarse de forma aleatoria antes de seleccionar la muestra.

3.1.4. Discretización de variables cuantitativas.

En algunos de los procedimientos utilizados en esta tesis como es el caso del modelo multinomial de redes bayesianas, los árboles de decisión o en otros procedimientos de minería de datos como las reglas de asociación es necesario transformar los valores de la variables numéricas en conjuntos de variables nominales ordenadas. Transformamos los valores en otro conjunto de intervalos disjuntos que cubren completamente del dominio de la variable continua. Los diferentes métodos que existen los podemos clasificar en cuatro grandes grupos:

- ✓ Locales o globales. Los métodos globales emplean toda la información de la variable continua en el proceso de discretización mientras que los locales solo utilizan un subconjunto de valores.
- ✓ Métodos supervisados o no supervisados. Cuando los métodos de discretización utilizan la información contenida en la clase para mejorar la csu calidad decimos que son métodos supervisados, en caso contrario son métodos no supervisados.
- ✓ Métodos top-down (separación) o bottom up (agrupamiento). Lo métodos de agrupamiento comienzan el proceso con una lista completa de todos los valores de la variable como puntos de corte. El proceso de discretización se forma mientras se van eliminando los valores. Lo métodos de separación comienzan con una lista vacía y se van agregando puntos que irán formando los intervalos.
- ✓ Métodos estáticos o dinámicos. Los estáticos reciben el nombre porque cada intervalo se forma independientemente de los demás, frente a los dinámicos que discretizan considerando simultáneamente todos los valores continuos de la variable.

Los cuatro métodos más utilizados son los siguientes: método de igual longitud, método de igual frecuencia, discretización por mínima entropía y el Chi Merge.

El método de igual longitud es el más simple de todos los métodos no supervisados. Los intervalos se construyen calculando el máximo y el mínimo de la variable y se divide por el rango en k intervalos de longitud.

El método de igual frecuencia también es supervisado. Este procedimiento construye intervalos que contienen todos el mismo número de observaciones. Experimentalmente se ha demostrado que si tomamos **k=max{1, log *I*},** donde ***I*** es el número de valores diferentes se consiguen excelentes resultados.

El método ChiMerge fue diseñado por Keber (1992) y es un tipo de discretización bottom-up supervisado ya que utiliza la información contenida en la clase donde las frecuencias relativas de las clases deben ser consistentes entre intervalos. Como su propio nombre indica en el proceso de discretización se utiliza la prueba estadística χ^2 para discriminar si dos intervalos adyacentes son independientes. Si este es el caso se juntan los intervalos y se separan, en caso contrario. La primera etapa de este algoritmo consiste en ordenar las observaciones que se pretenden discretizar y empieza el proceso colocando cada observación en un intervalo. La segunda etapa abarca dos pasos que continuaran hasta que el proceso finalice. En el primer paso se calcula el valor del estadístico χ^2 para cada para de intervalos y en el segundo paso se agrupan los intervalos con menor valor del estadístico. Este procedimiento continuará hasta que todos los valores de la χ^2 sean mayores que cierto umbral asociado al nivel se significación de la prueba estadística.

Respecto a los métodos que utilizan el concepto de mínima entropía existen en la literatura numerosos procedimientos que aplican este criterio para discretizar atributos continuos. Entre los primeros autores podemos citar a Chiu *et al.* (1990) a Kaufmann y Pfahringe (1987) y a Wong y Chiu (1987). Entre los métodos más utilizados destacan las propuestas de Catlett (1991) y Fayyard e Irani (1993). que recurren a la entropía de la clase para establecer los límites de los intervalos (cortes) en los que se dividirá el rango de un atributo continuo. En este método se seleccionan los puntos de corte de forma recursiva mediante un algoritmo de minimización de la entropía usando el criterio de Longitud de Descripción Mínima desarrollado por Suzuki (1996). Este es el procedimiento de discretización que se utiliza en esta tesis.

La entropía es la medida del desorden de un sistema mediante la incertidumbre existente ante un conjunto de casos, del cual se espera uno sólo. Sea D un conjunto de datos etiquetados con clases del conjunto C = (C_1……C_k) y frec (C_i, D) el número

de ejemplos de D con clase Ci. Entonces se define la entropía del conjunto D de la forma siguiente:

$$Ent(D) = -\sum_{i=1}^{k} \frac{frec(C_i, D)}{|D|} x \log_2 \left(\frac{frec(C_i, D)}{|D|} \right) \quad (3.14)$$

Donde $\frac{frec(C_i, D)}{|D|}$ representa la probabilidad de que se dé un ejemplo con clase C_i, y

$\log_2 \left(\frac{frec(C_i, D)}{|D|} \right)$ identifica la información que transmite un ejemplo de la clase C_i. La entropía es más máxima si todas las clases representan la misma proporción.

Utilizando la notación de Fayyad e Irani, Si el conjunto de datos lo representamos como S, un atributo como A, y el corte como T, la entropía de clase de los intervalos S_1 y S_2 inducidos por T es calculada con la siguiente expresión:

$$E(A,T;S) = \frac{|S_1|}{|S|} xEnt(S_1) + \frac{|S_2|}{|S|} Ent(S_2) \quad (3.15)$$

Donde $|S|$, $|S_1|$ y $|S_2|$ indican el número de instancias de las particiones de cada conjunto y Ent(.) es la entropía, la cual se calcula a través de la ecuación 3.16. Así, para cada atributo se selecciona el corte T entre todas las posibles particiones que minimiza $E(A,T;S)$.

Una vez establecido el corte, se aplica recursivamente esta heurística a cada una de las dos particiones resultantes (S1 y S2) hasta que se satisface un criterio de parada.

La diferencia entre el algoritmo de Catlett y la propuesta de Fayyad e Irani radica en ese criterio. Mientras el método de Catlett se detiene cuando el número de ejemplos en un intervalo es suficientemente pequeño o el número de intervalos alcanza un máximo, Fayyad e Irani usan el *principio de longitud de descripción mínima* como condición de parada, deteniendo el algoritmo si y sólo si

$$Ganancia(A,T;S) < \frac{\log_2(|S|-1)}{|S|} + \frac{\Delta(A,T;S)}{|S|} \quad (3.16)$$

Donde $Ganancia(A,T;S) = Ent(S) - E(A,T;S)$ y

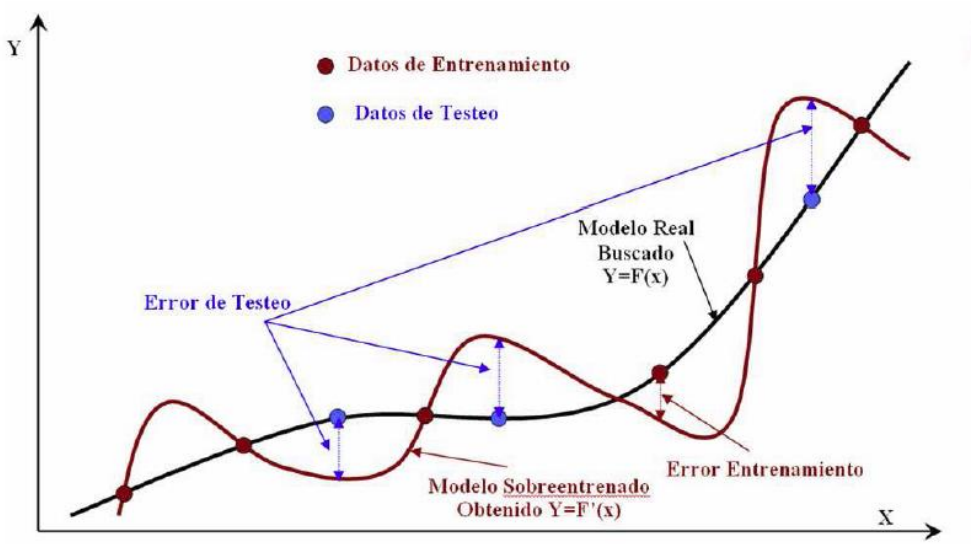
$$\Delta(A,T;S)|S| = \log_2(3^k - 2) - (k.ent(S) - k_1 Ent(S_1) - k_2.Ent(S_2)$$

y k, k_1 y k_2 son el número de clases distintas de S, S_1 y S_2 respectivamente. Este criterio puede producir intervalos muy desiguales para un mismo atributo, ya que, una vez establecido un corte, la evaluación de los dos subespacios resultantes es independiente. De este modo, zonas del espacio que presenten una baja entropía serán divididas muy pocas veces, dando intervalos relativamente grandes, mientras que en otras zonas con alta entropía, los cortes serán mucho más próximos.

3.1.5. Muestras de entrenamiento y de test.

En el proceso de estimación del modelo debemos de utilizar una estrategia para que nuestro método de clasificación se optimice. Utilizamos un conjunto de datos de entrenamiento y otro conjunto de test.

Figura 3.9. Errores de entrenamiento y de test.



Fuente: Grupo EDMANS. http://edmans.webs.com/.

3.1.5.1. Uso de muestras para el entrenamiento, validación y test.

Existen diferentes estrategias a la hora de utilizar el conjunto de datos disponible. Una práctica de muestreo, especialmente beneficiosa, es la partición (split) de la muestra en tres tablas de datos más pequeñas, que serán usadas con los siguientes fines: Training, Validación y Test.

La tabla de datos de training es utilizada para entrenar los modelos, es decir para estimar los parámetros del modelo. La tabla de datos de validación es utilizada para ajustar y/o seleccionar el mejor modelo. En otras palabras, en base a algunos criterios, el modelo con el mejor valor del criterio establecido será seleccionado. Por ejemplo, el menor error cuadrático medio de la predicción es utilizado muy a menudo. La tabla de datos de test es utilizada para comprobar el comportamiento del modelo seleccionado. Después de que se haya seleccionado el mejor modelo y se haya comprobado puede utilizarse para puntuar (score) la base de datos completa.

Cada registro en la muestra puede aparecer solamente en una de las tres tablas. Cuando subdividimos la muestra de la tabla de datos, se puede utilizar un muestreo aleatorio simple, o un muestreo estratificado. Primero, puede seleccionar aleatoriamente una pequeña fracción de los registros de una base de datos de 5-terabytes, y en general, los modelos más simples requieren menores tamaños de muestra y los modelos más complejos requieren mayores tamaños de muestra. En segundo lugar, puede usar el muestreo aleatorio de nuevo para subdividir la muestra: 40% training, 30% validación, 30% test.

3.1.5.2. Validación cruzada.

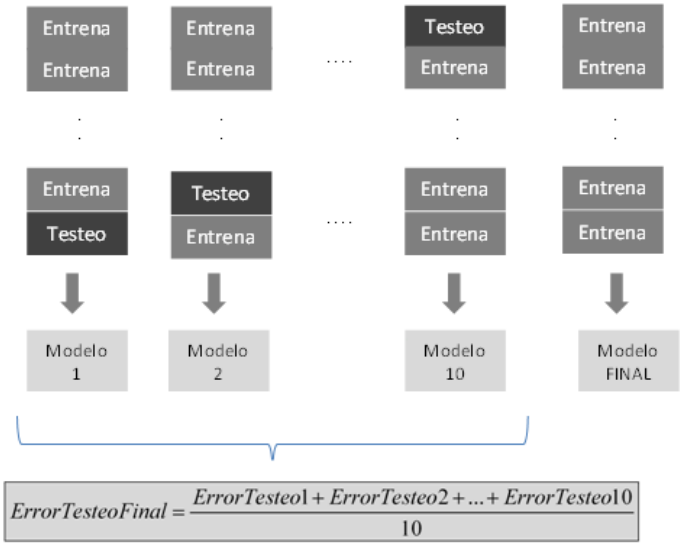
De manera ideal siempre que elaboremos un modelo lo óptimo es que lo entrenemos con un conjunto de entrenamiento y un conjunto de evaluación independientes que sean capaces de modelar la población original de manera individual.

Como esta situación en muchas ocasiones no es factible por falta de datos, dificultad de obtener un muestreo adecuado etc. Se suele aplicar un esquema de validación denominado validación cruzada que consiste en dividir el conjunto de entrenamiento en k particiones, repetir el procedimiento de entrenamiento y validación k veces, de forma que en cada una de ellas se entrene el modelo con k-1 particiones y se evalúe con la partición restante.

Los resultados finales se suelen obtener por agregación de los resultados originales.

Un esquema del desarrollo del proceso de cross validation se muestra en el gráfico *siguiente:*

Figura 3.10. Esquema de validación cross validation.



Fuente: Elaboración propia.

3.1.6. Métodos de evaluación de modelos de clasificación.

En cualquier etapa de un proceso de minería de datos resulta fundamental el poder estimar los niveles de calidad obtenidos por el modelo que hayamos construido. En función del problema y del modelo existen bastantes mecanismos de evaluación.

En credit scoring los dos grupos que deben de ser discriminados son los cumplidores y los morosos. Existirán algunos buenos pagadores que se clasificarán como malos y también habrá registros de personas morosas en la base de datos que serán clasificadas como buenos pagadores. Evaluar un modelo es encontrar aquel modelo que cometa el error mínimo en la clasificación de clientes. A continuación se exponen los métodos de evaluación más habituales utilizados en la clasificación y que se pueden agrupar en tres grupos:

- Métodos basados en métricas.
- Métodos basados en curvas ROC.
- Métodos que incorporan un matriz de costes.

3.1.6.1. Evaluación de modelos de clasificación basados en métricas.

En este problema disponemos de un conjunto de individuos del que conocemos previamente la clase a la que pertenecen. Estos individuos los evaluamos en nuestro modelo y en función de la disparidad entre los resultados obtenidos y los datos reales podemos definir diferentes métricas.

Una primera métrica es el porcentaje de acierto definido como el número de casos acertados entre el número de casos totales. Esta es la métrica más sencilla y habitual pero resulta muy engañosa en la mayor parte de las veces pues no informa nada acerca de la distribución del error entre clases. Además siempre que apliquemos esta métrica debemos tener en cuenta el balanceo entre clases, ya que si tenemos una población en la que el 90% de los individuos son de clase A y el 10% restante de clase B, cualquier algoritmo que obtenga un porcentaje de acierto inferior al 90% (decir siempre que los individuos son de clase A diremos que no aporta ningún tipo de conocimiento en la clasificación.

Para estudiar otras métricas más significativas necesitamos primero definir la Matriz de confusión. Esta matriz supone un salto cualitativo respecto al porcentaje de acierto ya que nos permite conocer la distribución del error a lo largo de las clases. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real.

Una matriz de confusión estándar tiene la siguiente estructura:

Tabla 3.1. Matriz de confusión.

		Clase clasificada como:		
		A+ (SI)	A- (NO)	Total
Estado real	A+ (SI)	Verdaderos positivos (VP) $FVP = \frac{VP}{TCP}$	Falsos negativos (FN) $FFN = \frac{FN}{TCP}$	1
	A- (NO)	Falsos positivos (FP) $FFP = \frac{FP}{TCA}$	Verdaderos negativos (TN) $FVN = \frac{VN}{TCA}$	1

Las sumas por columnas y por filas en esta matriz de confusión se corresponden a:

- TCP = VP + FN que representa el total de respuestas con presencia de la condición de interés. Es el número de individuos clasificados como que devuelven el crédito y los clasificados como que no devuelven el crédito pero su situación real es que se les otorgó el crédito solicitado.
- TCA = FP + VN. Total de respuestas con ausencia de la condición de interés.
- TRP = VP + FP. Total de repuestas positivas
- TRN = FN + VN. Total de respuestas

La precisión, exactitud o accuracy (AC) de un clasificador es el cociente entre el número de ejemplos que están bien clasificados, que se corresponde en la matriz de confusión con la suma de los elementos de la diagonal, entre el total de instancias.

$$AC = \frac{VP + VN}{N} \tag{3.17}$$

El clasificador ideal sería aquel cuyo valor de AC fuera igual a uno ya que así se verificaría que VP + VN = 1, FP = FN = 0, el clasificador no produciría instancias mal clasificadas.

La precisión se puede definir para cada clase, llamada habitualmente recall: para la clase A es el cociente de los ejemplos clasificados correctamente (verdaderos positivos) entre todos los elementos clasificados de esa clase (verdaderos positivos + falsos positivos). Para calcular la precisión de la clase B (recall de B) dividimos los registros correctamente predichos (verdaderos negativos) entre el total de los clasificados en esa clase (Verdaderos negativos + falsos negativos).

Existe una medida que combina la precisión y el recall y que se denomina F-Measure:

$$F - Measure = \frac{2 \cdot \mathrm{Pr}\,ecisión \cdot \mathrm{Re}\,call}{\mathrm{Pr}\,ecisión + \mathrm{Re}\,call} \tag{3.18}$$

En algunas disciplinas científicas son mucho más utilizados los conceptos de sensibilidad y de especificidad. La sensibilidad (S) es la probabilidad de clasificar correctamente a una instancia cuyo estado real sea la presencia de la condición de interés. Este valor también es conocido como la fracción de verdaderos positivos (FVP), recall o exactitud positiva.

$$S = \frac{VP}{VP + FN} = FVP \tag{3.19}$$

La especificidad (E) se define como la probabilidad de clasificar correctamente a un individuo cuyo estado real sea la ausencia de la condición. Es la proporción de respuestas negativas que son correctamente clasificadas:

$$E = \frac{VN}{FP + VN} = FVN \tag{3.20}$$

Dado que la sensibilidad y la especificidad son proporciones podemos construir sus intervalos de confianza asintóticos dados por:

$$S + z\alpha/2SD(E) \quad \text{y } E + z\alpha/2SD(E) \tag{3.21}$$

Donde

$$SD(S) = \sqrt{\frac{VP \cdot FN}{(VP + FN)^3}} \qquad \text{y} \qquad SD(E) = \sqrt{\frac{FP \cdot VN}{(FP + VN)^3}}$$

Otro medida de exactitud es el índice de Youden que se calcula por las diferencias de proporciones de respuestas positivas correctas e incorrectas. Es una medida que combina la sensibilidad y la especificidad por lo que este índice debería acompañar al estudio de los clasificadores que utilicen estos conceptos:

$$\gamma = FVP - FFP = S + E - 1 \tag{3.22}$$

La tasa o razón de verosimilitud (LR) es otra medida muy utilizada como forma de discriminar los clasificadores. Esta tasa de verosimilitud representa el grado de evidencia de una respuesta del clasificador a favor de la presencia de la condición de la condición con respecto a la ausencia de la condición.

Dependiendo de la respuesta del clasificador se definen dos tipos de tasas:

- Tasa de verosimilitud positiva (LRP) que es el cociente entre la sensibilidad y el complemento de la especificidad:

$$LRP = \frac{S}{1 - E} = \frac{FVP}{FFP} \tag{3.23}$$

- Tasa de verosimilitud negativa (LRN) definida como el cociente del complemento de la sensibilidad y la especificidad:

$$LRN = \frac{1-S}{E} = \frac{FFN}{FVN} \tag{3.24}$$

Las tasas de verosimilitud son tasas de probabilidad con lo que podemos calcular los intervalos de confianza:

$$\frac{S}{1-E}\exp(\pm z_{\alpha/2} SD(LRP) \text{ y } \frac{1-S}{E}\exp(\pm z_{\alpha/2} SD(LRN) \tag{3.25}$$

Donde

$$SD(LRP) = \sqrt{\frac{1-S}{VP} + \frac{E}{FP}} \qquad \text{y} \qquad SD(LRN) = \sqrt{\frac{S}{FN} + \frac{1-E}{VN}}$$

Atendiendo a los valores que tomas ambas tasas, Jaeschke *et al.* (2002) definen zonas de valores para poder ver la capacidad de discriminar de un clasificador (tabla 3.2.)

Tabla 3.2. Categorización de la tasa de verosimilitud.

< 0,1	0,1 – 0,2	0,2 – 0,5	0,5 – 2	2 – 5	5 – 10	>10
Excelente	Muy bueno	Bueno	Justo	Bueno	Muy bueno	Excelente

Fuente: Franco y Vivo (2007).

Otro índice muy utilizado es el estadístico Kappa. Es un coeficiente estadístico que determina la precisión del modelo a la hora de predecir la clase verdadera. Este estadístico está ampliamente difundido.

Se define como:

$$k = \frac{P(A) - P(E)}{1 - P(E)} \tag{3.26}$$

Siendo $P(A) = \frac{SumaDiagonales}{NúmeroCasos} = \frac{VP + VN}{N}$

P(A) es el porcentaje de casos acertados y P(E) es el porcentaje de casos cambiados. Para medir P(E) existen varias formas. El programa WEKA, utilizado en esta tesis lo proporciona de forma habitual lo calcula a través de la siguiente fórmula:

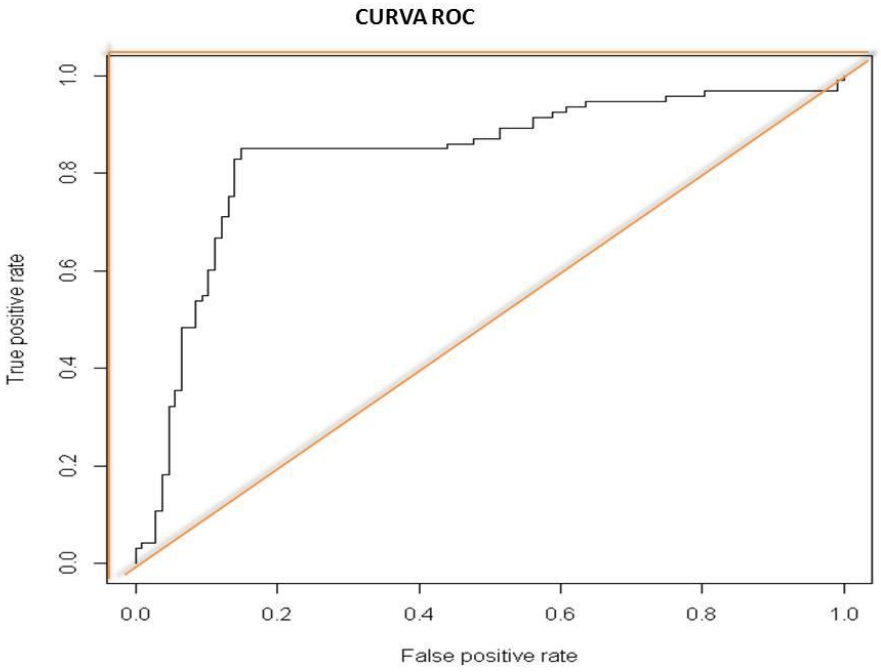
$$P(E) = \frac{(VP + FN) \cdot (VP + FP}{2N} + \frac{(VN + FP) \cdot (VN + FN}{2N} \qquad 3.27)$$

El estadístico Kappa toma valores entre cero y uno, indicando el primero la absoluta falta de concordancia y el segundo la concordancia total. Si el resultado es un valor menor que 0,4 se considera insuficiente, entre 0,4 y 0,6, la concordancia es moderada y para valores superiores a 0,6 la concordancia es elevada.

3.1.6.2. Evaluación de modelos de clasificación basados en curvas ROC.

La curva ROC (Receiver Operating Characteristic) es una representación gráfica del rendimiento de un clasificador que muestra la distribución de las fracciones de verdaderos positivos y la fracción de falsos negativos (Figura 3.11.). La curva ROC nos proporciona una herramienta visual para examinar la capacidad que dispone un clasificador para detectar correctamente a los individuos con presencia de la condición de interés en el análisis y su incapacidad para identificar los individuos del grupo de ausencia.

Figura 3.11. Curva ROC.



Fuente elaboración propia.

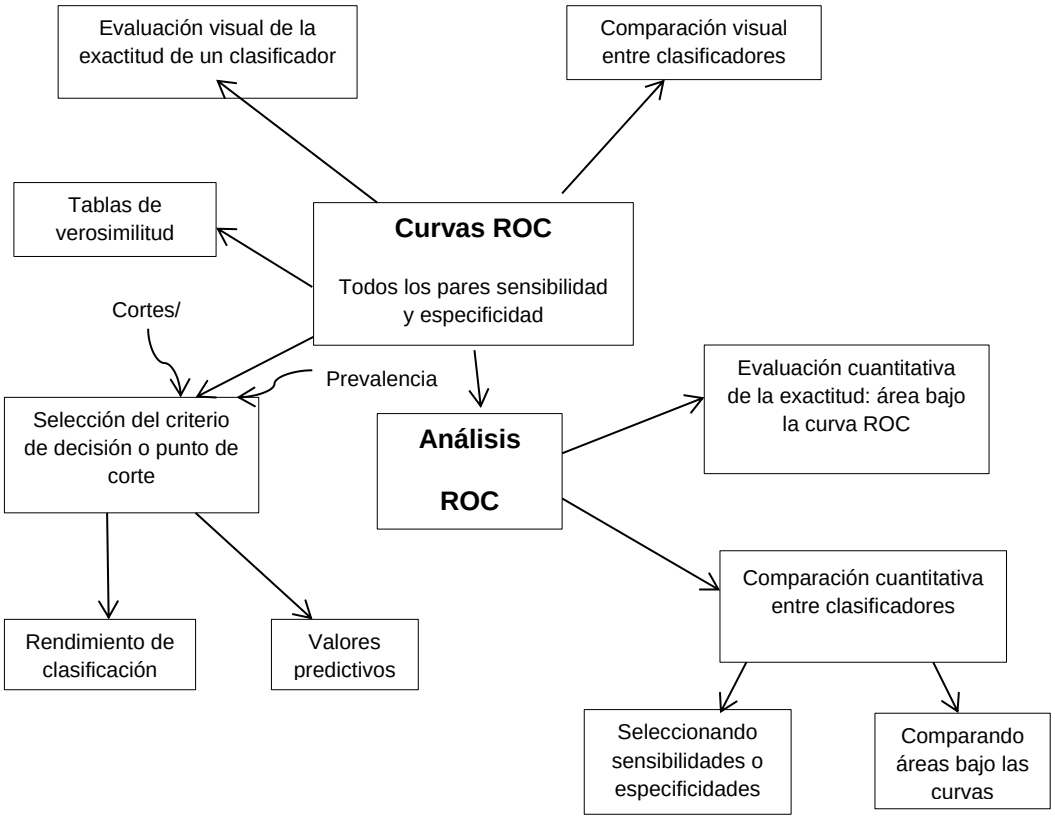
Swets y Pickett (1982) señalan tres importantes propiedades de las curvas ROC:

a. Representa un índice de exactitud intrínseca, es un índice de capacidad del clasificador par discriminar estados y seleccionando el estado correcto independientemente del criterio de selección.
b. Estas curvas refleja las probabilidades subjetivas junto con las utilidades que usualmente determina este criterio por lo que podemos decir que es un índice del criterio de decisión. Hace posible utilizar las probabilidades a priori de los posibles estados de forma más precisa y también los costes de las decisiones correctas y equivocadas para determinar el criterio óptimo de decisión de un clasificador.
c. Las curvas ROC contienen las estimaciones de probabilidades de los distintos tipos de resultados de la decisión para todos y cada uno de los criterios de decisión.

La curva ROC es el patrón de oro en muchas áreas de análisis de modelos ya que representan de forma compacta muchísima información del rendimiento de un clasificador.

Muchos autores consideran a las curvas ROC como herramientas fundamentales en la fase de evaluación de un modelo, entre ellos Zweig y Campbell (1993) como se observa en la figura 3.12.

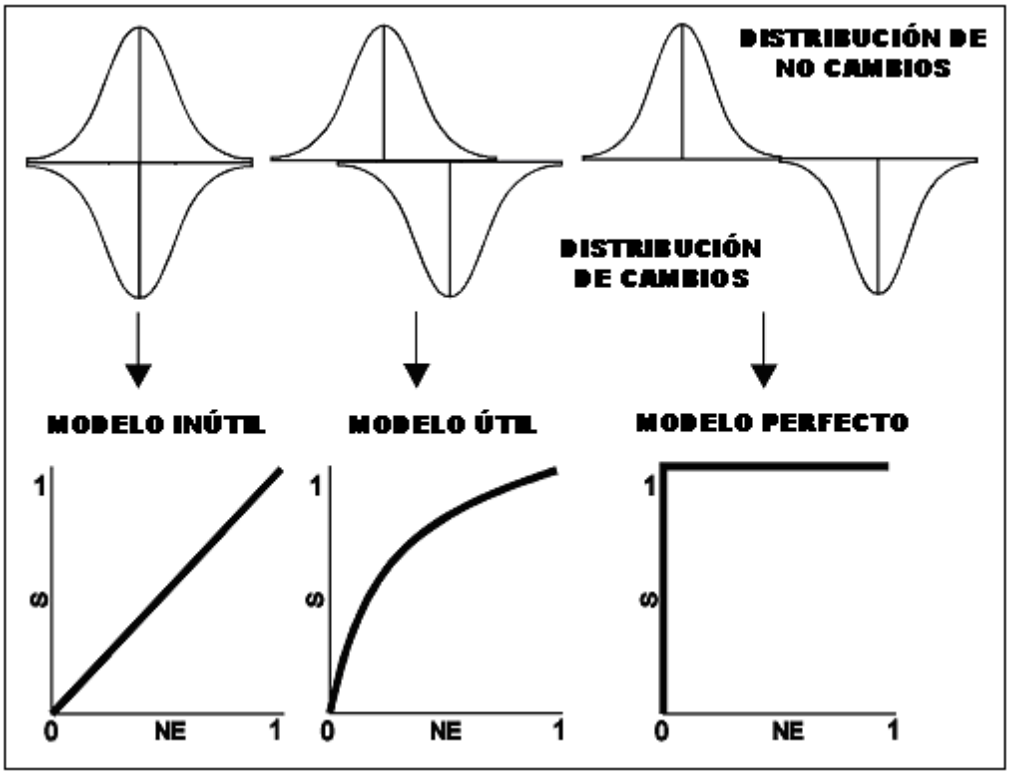
Figura 3.12. Diagrama sobre la posición central de las curvas ROC.



Fuente: Zweig y Campbell (1993). Adaptado de Franco y Vivo (2007).

La curva ROC refleja el grado de solapamiento de las estimaciones del modelo en los dos grupos de interés (cambio/no cambio). Cuando el solapamiento es total (modelo inútil), ver figura 3.14, la curva ROC recorre la diagonal positiva del gráfico, ya que para cualquier punto de corte S = NE. Cuando el solapamiento es nulo (test perfecto), la curva ROC recorre los bordes izquierdo y superior del gráfico, ya que para cualquier punto de corte, o bien S = 1, o bien NE = 0, existiendo algún punto de corte en el que S = 1 y NE = 0. En la práctica el solapamiento de valores para un grupo u otro será parcial, generando curvas ROC intermedias entre las dos situaciones planteadas (Weinstein y Fineberg, 1980).

Figura 3.13. Correspondencia entre solapamiento de las distribuciones y la curva ROC.



Fuente: Montaño, (2005).

La fracción de verdaderos positivos se conoce como sensibilidad, es decir es la probabilidad de clasificar correctamente a (1995) un individuo cuyo estado real sea definido como positivo. La especificidad es la probabilidad de clasificar correctamente a un individuo cuyo estado real sea calificado como negativo. Esto es igual a restar a uno la fracción de falsos positivos. La curva ROC permite comparar modelos a través del área bajo su curva (Figura 3.14).

Algunas de las propiedades matemáticas de estas curvas se encuentran en Pepe (2003) y en Krzanowski y Hand (2009).

- La curva ROC es una función monótona creciente en el intervalo (0,1) con límites iguales a cero o a uno:

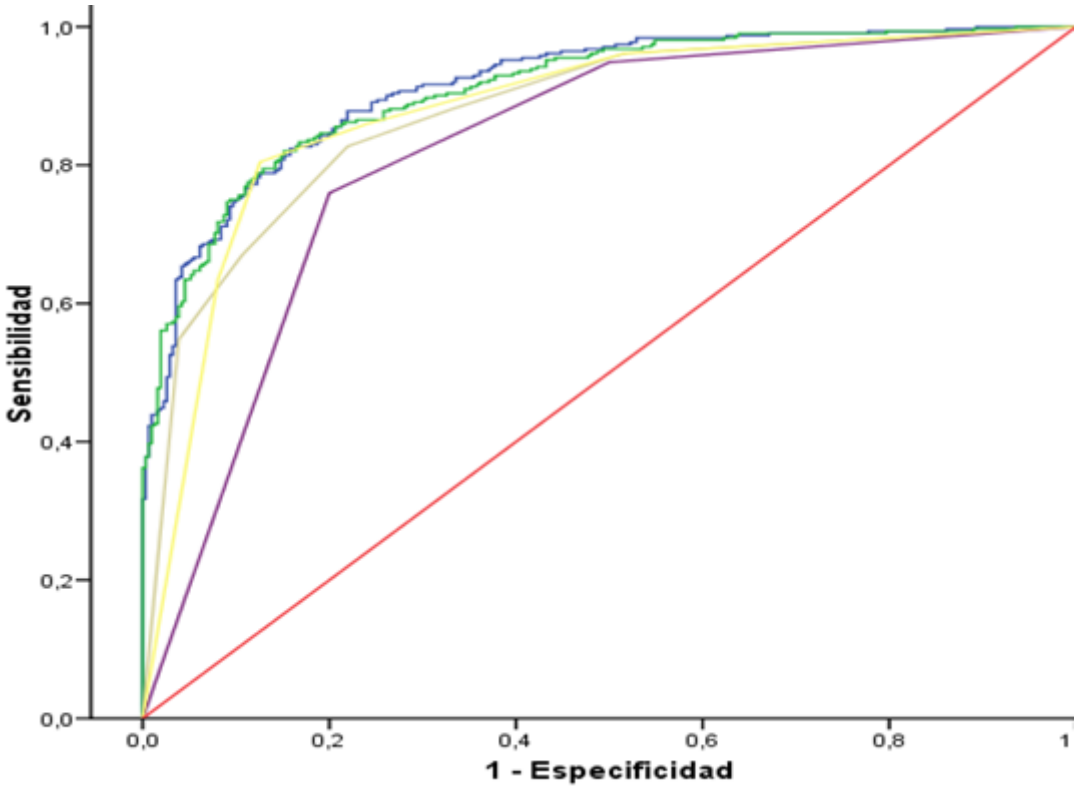
$$\lim_{t\to 0} ROC(t) = 0 \qquad \lim_{t\to 1} ROC(t) = 1$$

- Esta curva es invariante ante transformaciones monótonas estrictamente crecientes de la escala del clasificador tales como cambios lineales, logarítmicos o raíces cuadradas.
- La curva ROC es la función en (0,1) con ROC (0) = 0 y ROC(1) = 1 con pendiente:

$$\frac{\partial ROC(t)}{\partial(t)} = \frac{f_D(F_D^{-1}(t)}{f_{\bar{D}}(F_{\bar{D}}^{-1}(t)} \tag{3.28}$$

Donde f_D y $f_{\bar{D}}$ son las funciones de densidad de los resultados del marcador Y

Figura 3.14. Curvas ROC de cinco modelos de clasificación.

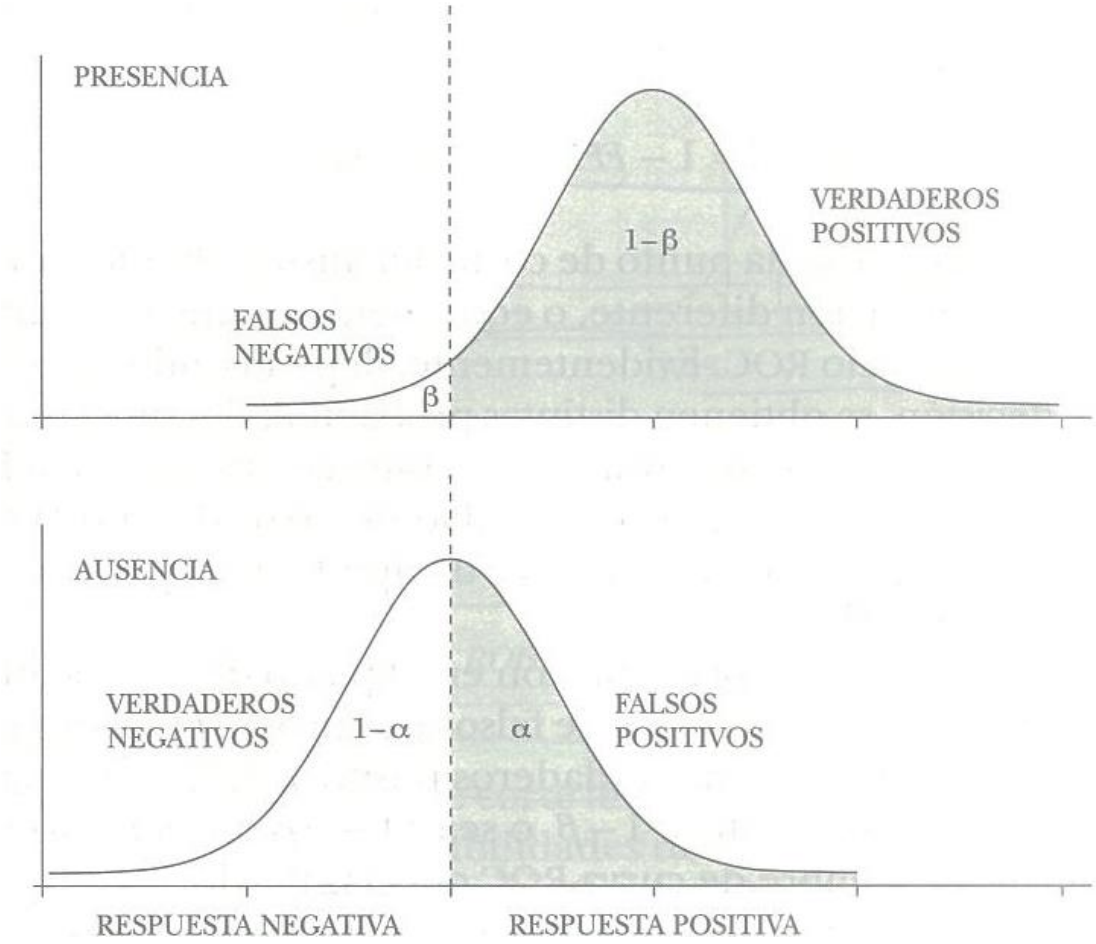


Fuente elaboración propia.

Se pueden establecer analogías interesantes entre las falsas clasificaciones de la matriz de confusión y los errores de tipo I (α) y II (β) utilizados en la contrastación estadística. Siguiendo a Franco y Vivo (2007) podemos ver en la figura 3.16. la relación entre los errores que se producen en la contrastación de hipótesis estadísticas y la sensibilidad y la especificidad de las tablas de decisión.

El error de tipo I se define como la probabilidad de rechazar la hipótesis nula siendo cierta, tomamos una decisión equivocada. Este valor es la fracción de falsos positivos que es igual a 1 menos la especificidad. FFP = (1 – especificidad). El error de tipo II se corresponde con la fracción de falsos negativos (FFN = 1 – sensibilidad). La potencia estadística viene dada por la sensibilidad (1 – β) = 1 –FFN= sensibilidad)

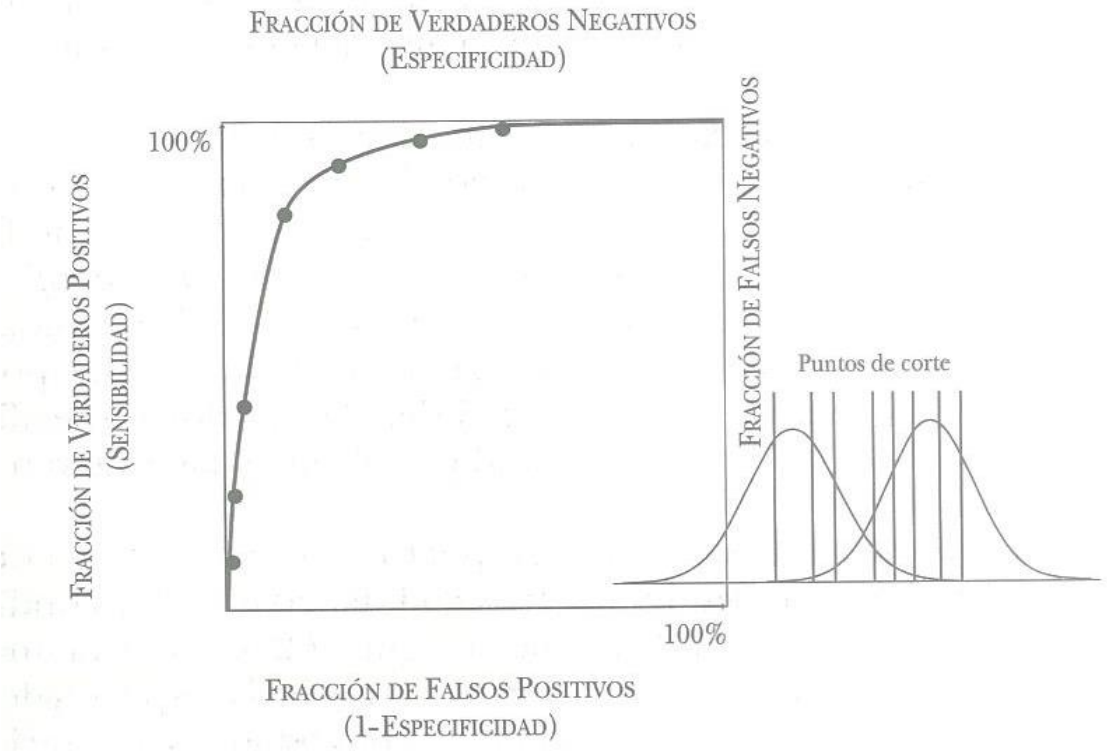
Figura 3.15. Distribuciones del clasificador según el estado de la condición.



Fuente: Franco y Vivo (2007).

Según vayamos variando los diferentes puntos de corte, es decir definiendo diferentes niveles de los tipos de error se irán produciendo diferentes puntos de clasificación, lo que es equivalente a definir distintos puntos en el espacio ROC (figura 3.17.). Para obtener la curva ROC simplemente tenemos que variar el punto de corte inferior y superior del soporte del clasificador o variable de predicción para obtener diferentes registros de α y de β y por tanto diferentes valores de sensibilidad y de especificidad que son los que se reflejan en la construcción de la curva ROC.

Figura 3.16. Curva ROC y posibles criterios de decisión



Fuente: Franco y Vivo (2007).

Como forma agregada se suele utilizar el valor del área bajo la curva ROC que se conoce con las siglas AUC. Esta medida se interpreta como la probabilidad de clasificar correctamente un par de sujetos,uno que ha realizado el cambio y otro que no, seleccionados al azar, fluctuando su valor entre 0.5 y 1 lo que nos permite evaluar el rendimiento de un modelo de manera muy precisa. El AUC de un modelo que tiene un valor de 0,5 se considera inútil ya que sólo clasifica correctamente un 50% de individuos, idéntico porcentaje al obtenido utilizando simplemente el azar. Por el contrario, el AUC de un modelo perfecto es 1, ya que permite clasificar sin error el 100% de individuos.

El problema de las curvas ROC es que en ocasiones no podemos calcularlas para determinados algoritmos.

El área bajo la curva ROC es la probabilidad de que la respuesta del clasificador sea mayor en presencia de la condición (X_p) que en ausencia de la condición (X_a):

$$AUC = P(X_p > X_a) \tag{3.29}$$

El AUC es equivalente al valor del estadísticos suma de rangos de Wilcoxon tal y como señalan Bamber (1975) y Hanley y McNeil (1982) lo que permite trasladar las propiedades del estadístico de Wilcoxon a las medidas de exactitud global de AUC. También se interpreta como un promedio de la sensibilidad para todos los valores de especificidad y, de forma análoga, como un promedio de la especificidad para todos los posibles valores de sensibilidad

Breiman *et al.* (1984) relacionan el AUC con el coeficiente de Gini. En concreto, Hand y Till (2001) lo utilizan de la forma siguiente:

$$C_{Gini} = 2.AUC - 1 \tag{3.30}$$

Existen diferentes procedimientos para calcula el área bajo la curva ROC. En Faraggi y Reiser (2002) se encuentran descritos diferentes métodos y algunas recomendaciones para situaciones particulares. Entre los procedimientos n**o** paramétricos para estimar el valor del AUC se utiliza la regla trapezoidal que estima el área sumando los trapecios formados por la unión de los puntos de la curva ROC:

$$AUC = \sum_{t=1}^{T} \frac{1}{2}(FFP_T - FFP_{t-1})(FVP_t - FVP_{t-1}) \tag{3.31}$$

Otro método no paramétrico para el cálculo del AUC se realiza a través del estadístico U de Mann-Whitney conocido también como la suma de rangos de Wilcoxon y que toma la siguiente expresión:

$$U = \sum_{i=1}^{n_p} \sum_{j=1}^{n_a} S(X_{pi}, X_{aj}) \tag{3.32}$$

Siendo:

$S(X_p, X_a)$ igual a 1 si $X_p > X_a$, igual a ½ si $X_p = X_a$ y cero si $X_p < X_a$.

n_p representan las repuestas de X_{pi} con presencia de la condición y n_a las respuestas de X_a con ausencia.

El área bajo la curva ROC siguiendo a Hanley y McNeil (1982) toma la siguiente expresión:

$$AUC = \frac{1}{n_p n_a} U \tag{3.33}$$

La expresión del error estándar de este estimador del AUC se obtiene por la siguiente fórmula:

$$SD(\hat{AUC}) = \sqrt{\frac{AUC(1-AUC) + (n_p - 1)(Q_1 - AUC^2) + (n_a - 1)(Q_2 - AUC^2)}{n_p n_a}} \tag{3.34}$$

Q_1 y Q_2 representan probabilidades y son difíciles de calcular pero Hanley y McNeil (1982) sugieren una simplificación de estos valores basados en la distribución exponencial:

$$\hat{Q}_1 = \frac{\hat{AUC}}{2 - \hat{AUC}} \quad \text{y} \quad \hat{Q}_2 = \frac{2\hat{AUC}}{1 - \hat{AUC}}$$

Otros autores como Obuchowski (1994) y DeLong *et al.* (1988) han propuesto procedimientos para el cálculo del AUC y su error estándar. También se puede calcular la varianza del área bajo la curva ROC a través de otras técnicas del tipo jacknife o bootstrap, Hanley y Hajian-Tilaky (1997)

En la literatura sobre las curva ROC se encuentra una tabla con los intervalos de los valores sobre el área bajo la curva que no sirven para determinar la capacidad global de un clasificador tal y como se observa en la tabla 3.2.

Tabla 3.3. Categorías de exactitud global de un clasificador según el AUC.

0,50-0,75	0,75-0,92	0,92-0,97	0,97-1
JUSTA	BUENA	MUY BUENA	EXCELENTE

Fuente: Franco y Vivo (2007).

Una vez que disponemos de la estimación del AUC y de su varianza podemos acompañar a las estimaciones de sus intervalos de confianza y aplicar la teoría de la

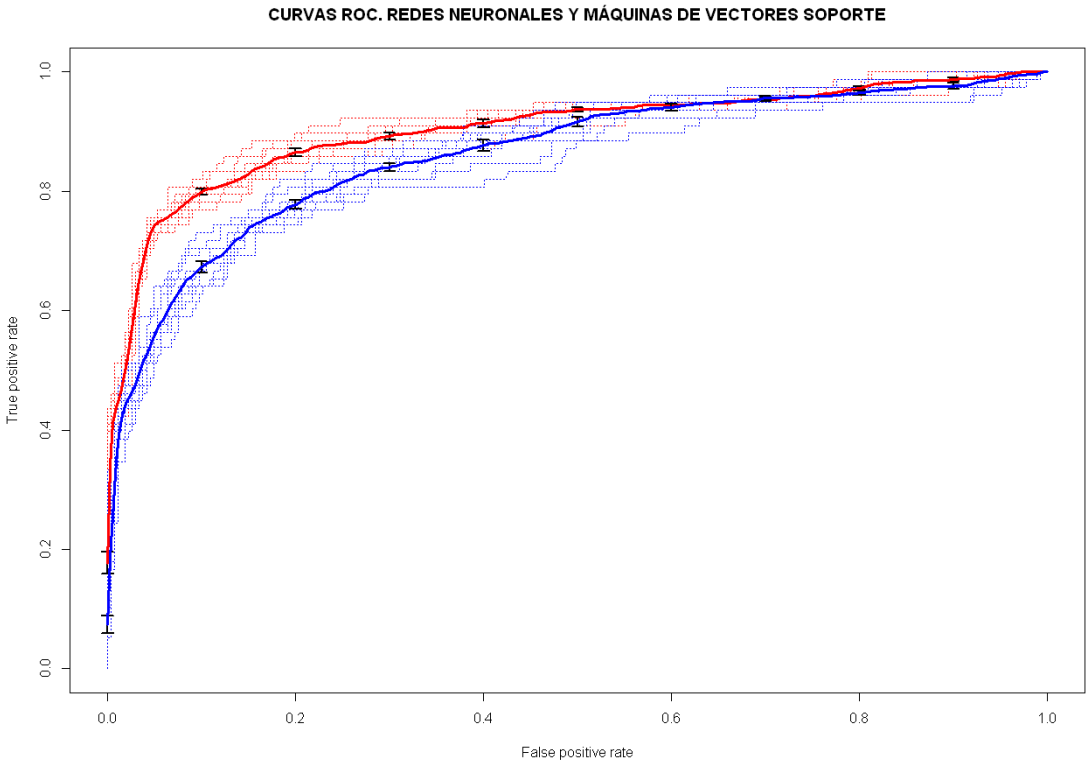
contrastación de hipótesis estadísticas para el discernimiento de los mejores clasificadores.

El intervalo de confianza para un estimador se calcula a través de esta expresión:

$$I.C.(AUC_1) = A\hat{U}C_1 \pm t_{\alpha/2} SD(A\hat{U}C_1) \qquad (3.35)$$

Donde $t_{\alpha/2}$ es el valor de la tablas de la normal para el nivel de significación α/2.

Figura 3.17. Representación de la curva ROC de dos algoritmos con intervalos de confianza.



En relación a la capacidad de predicción de diferentes modelos, dos contrastes del AUC interesantes que podemos llevar a cabo son, en primer lugar, contrastar si existen diferencias significativas entre el área bajo la curva ROC del clasificador con respecto a la mínima exactitud 05 y, en segundo término, comparar áreas bajo las curvas ROC de varios algoritmos clasificadores a través del contraste de diferencias de medias.

En el primer contraste la hipótesis nula y la alternativa y su estadístico de contraste son los siguientes:

$$H_0 : \ AUC = 0{,}5$$

$$H_1 : \ AUC \neq 0{,}5$$

$$z = \frac{\hat{AUC} - 0{,}5}{SD(\hat{AUC})} \tag{3.36}$$

El estadístico z sigue una distribución asintótica normal estándar bajo la hipótesis nula. Dado un nivel de significación α se rechaza la hipótesis nula cunado $|z| > z_{\alpha/2}$ .Rechazar la hipótesis nula implica que el clasificador tiene la capacidad de discriminar correctamente a los individuos.

Para determinar si existen diferencias significativas entre sus curvas ROC entre diferentes algoritmos clasificadores procederíamos de la forma siguiente:

En primer lugar establecemos la hipótesis nula y la alternativa:

$$H_0 : AUC_1 - AUC_2 = 0$$

$$H_1 : AUC_1 - AUC_2 \neq 0$$

Al igual que antes tenemos que establecer un estadístico de contraste z:

$$z = \frac{\hat{AUC}_1 - \hat{AUC}_2}{SD(\hat{AUC}_1 - \hat{AUC}_2)} \tag{3.37}$$

La distribución asintótica de estadístico se corresponde con una normal multivariante donde $\hat{AUC}_1$ y $\hat{AUC}_2$ son los estimadores del área bajo la curva ROC estimada para cada uno de los procedimientos de clasificación y $SD(\hat{AUC}_1 - \hat{AUC}_2)$ la desviación estándar de la diferencia. Cuando las muestras son independientes su cálculo se efectúa a través de la siguiente expresión:

$$SD(\hat{AUC}_1 - \hat{AUC}_2) = \sqrt{\hat{V}(\hat{AUC}_1) + \hat{V}(\hat{AUC}_2)} \tag{3.38}$$

Cuando las muestras están apareadas hay que incorporar el valor del coeficiente de correlación entre las áreas bajo las curvas ROC:

$$r = \frac{Cov(\hat{AUC}_1, \hat{AUC}_2)}{SD(\hat{AUC}_1) \cdot SD(\hat{AUC}_2)} \tag{3.39}$$

La expresión de la desviación estándar de la diferencia de medias incorporando el coeficiente de correlación es:

$$SD(A\hat{U}C_1 - A\hat{U}C_2) = \sqrt{\hat{V}(A\hat{U}C_1) + V(A\hat{U}C_2) - 2.r.SD(A\hat{U}C_1).SD(A\hat{U}C_2)} \quad (3.40)$$

El cálculo de coeficiente de correlación se puede llevar a cabo por el procedimiento expuesto en Hanley y McNeil (1983) de la siguiente forma:

Se representa a dos clasificadores por X e Y que son los que van a discriminar a los individuos de la población a través de una muestra con n_p individuos que presentan la condición y n_a con ausencia de la condición.

La correlación muestral entre los clasificadores en cada submuestra con presencia y ausencia de la condición lo denominamos r_p y r_a. Se calcula el promedio de las correlaciones:

$$r_m = \frac{1}{2}\left(r_p + r_a\right) \quad (3.41)$$

Igualmente calculamos el promedio del área de las curvas ROC de cada clasificador:

$$AUC_m = \frac{1}{2}(A\hat{U}C_X + A\hat{U}C_Y) \quad (3.42)$$

Para obtener el valor del coeficiente de correlación en la tabla obtenida por Hanley Y McNeil (1983) (Tabla 3.4.)

Tabla 3.4.Valores para el cálculo de la correlación entre dos AUC.

r_m \ AUC_m	0.700	0.725	0.750	0.775	0.800	0.825	0.850	0.875	0.900	0.925	0.950	0.975
0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01
0.04	0.04	0.04	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.02	0.02	0.02
0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.04	0.04	0.04	0.03	0.02
0.08	0.07	0.07	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.05	0.04	0.03
0.10	0.09	0.09	0.09	0.09	0.08	0.08	0.08	0.07	0.07	0.06	0.06	0.04
0.12	0.11	0.11	0.11	0.10	0.10	0.10	0.09	0.09	0.08	0.08	0.07	0.05
0.14	0.13	0.12	0.12	0.12	0.12	0.11	0.11	0.11	0.10	0.09	0.08	0.06
0.16	0.14	0.14	0.14	0.14	0.13	0.13	0.13	0.12	0.11	0.11	0.09	0.07
0.18	0.16	0.16	0.16	0.16	0.15	0.15	0.14	0.14	0.13	0.12	0.11	0.09
0.20	0.18	0.18	0.18	0.17	0.17	0.17	0.16	0.15	0.15	0.14	0.12	0.10
0.22	0.20	0.20	0.19	0.19	0.19	0.18	0.18	0.17	0.16	0.15	0.14	0.11
0.24	0.22	0.22	0.21	0.21	0.21	0.20	0.19	0.19	0.18	0.17	0.15	0.12
0.26	0.24	0.23	0.23	0.23	0.22	0.22	0.21	0.20	0.19	0.18	0.16	0.13
0.28	0.26	0.25	0.25	0.25	0.24	0.24	0.23	0.22	0.21	0.20	0.18	0.15
0.30	0.27	0.27	0.27	0.26	0.26	0.25	0.25	0.24	0.23	0.21	0.19	0.16
0.32	0.29	0.29	0.29	0.28	0.28	0.27	0.26	0.26	0.24	0.23	0.21	0.18
0.34	0.31	0.31	0.31	0.30	0.30	0.29	0.28	0.27	0.26	0.25	0.23	0.19
0.36	0.33	0.33	0.32	0.32	0.31	0.31	0.30	0.29	0.28	0.26	0.24	0.21
0.38	0.35	0.35	0.34	0.34	0.33	0.33	0.32	0.31	0.30	0.28	0.26	0.22
0.40	0.37	0.37	0.36	0.36	0.35	0.35	0.34	0.33	0.32	0.30	0.28	0.24
0.42	0.39	0.39	0.38	0.38	0.37	0.36	0.36	0.35	0.33	0.32	0.29	0.25
0.44	0.41	0.40	0.40	0.40	0.39	0.38	0.38	0.37	0.35	0.34	0.31	0.27
0.46	0.43	0.42	0.42	0.42	0.41	0.40	0.39	0.38	0.37	0.35	0.33	0.29
0.48	0.45	0.44	0.44	0.43	0.43	0.42	0.41	0.40	0.39	0.37	0.35	0.30
0.50	0.47	0.46	0.46	0.45	0.45	0.44	0.43	0.42	0.41	0.39	0.37	0.32
0.52	0.49	0.48	0.48	0.47	0.47	0.46	0.45	0.44	0.43	0.41	0.39	0.34
0.54	0.51	0.50	0.50	0.49	0.49	0.48	0.47	0.46	0.45	0.43	0.41	0.36
0.56	0.53	0.52	0.52	0.51	0.51	0.50	0.49	0.48	0.47	0.45	0.43	0.38
0.58	0.55	0.54	0.54	0.53	0.53	0.52	0.51	0.50	0.49	0.47	0.45	0.40
0.60	0.57	0.56	0.56	0.55	0.55	0.54	0.53	0.52	0.51	0.49	0.47	0.42
0.62	0.59	0.58	0.58	0.57	0.56	0.56	0.55	0.54	0.53	0.51	0.49	0.45
0.64	0.61	0.60	0.60	0.59	0.59	0.58	0.58	0.57	0.55	0.54	0.51	0.47
0.66	0.63	0.62	0.62	0.62	0.61	0.60	0.60	0.59	0.57	0.56	0.53	0.49
0.68	0.65	0.64	0.64	0.64	0.63	0.62	0.62	0.61	0.60	0.58	0.56	0.51
0.70	0.67	0.66	0.66	0.66	0.65	0.65	0.64	0.63	0.62	0.60	0.58	0.54
0.72	0.69	0.69	0.68	0.68	0.67	0.67	0.66	0.65	0.64	0.63	0.60	0.56
0.74	0.71	0.71	0.70	0.70	0.69	0.69	0.68	0.67	0.66	0.65	0.63	0.59
0.76	0.73	0.73	0.72	0.72	0.72	0.71	0.71	0.70	0.69	0.67	0.65	0.61
0.78	0.75	0.75	0.75	0.74	0.74	0.73	0.73	0.72	0.71	0.70	0.68	0.64
0.80	0.77	0.77	0.77	0.76	0.76	0.76	0.75	0.74	0.73	0.72	0.70	0.67
0.82	0.79	0.79	0.79	0.79	0.78	0.78	0.77	0.77	0.76	0.75	0.73	0.70
0.84	0.82	0.81	0.81	0.81	0.81	0.80	0.80	0.79	0.78	0.77	0.76	0.73
0.86	0.84	0.84	0.83	0.83	0.83	0.82	0.82	0.81	0.81	0.80	0.78	0.75
0.88	0.86	0.86	0.85	0.85	0.85	0.85	0.84	0.84	0.83	0.82	0.81	0.79
0.90	0.88	0.88	0.88	0.88	0.87	0.87	0.87	0.86	0.86	0.85	0.84	0.82

Fuente: Hanley y McNeil (1983).

3.1.6.3. Evaluación de modelos de clasificación basados en costes.

Otra alternativa disponible para comparar modelos es establecer una matriz de costes asociadas a la clasificación. Cuando utilizamos el porcentaje de acierto o de error para evaluar el desempeño de nuestros modelos de clasificación estamos suponiendo que ambos tipos de errores son equivalentes.

Los factores de riesgo de los modelos de credit scoring, es decir, los factores que están detrás de los errores tipo I (admitir como sana una operación insolvente) y tipo II (rechazar como insolvente una operación sana) no son los mismos. No es igual, en términos de coste económico clasificar a un cliente como bueno, concederle el crédito y que luego no nos lo devuelva que no conceder el crédito a una persona que es cliente. En el primer caso estamos expuestos a un caso de riesgo de crédito y, en el otro caso, incurrimos en un coste de oportunidad por la pérdida potencial de buenos clientes.

La mayoría de algoritmos de aprendizaje, por su propia naturaleza, buscan minimizar el número de errores del clasificador generado. Sin embargo, son múltiples los problemas de Aprendizaje Automático en los que los errores cometidos por el clasificador generado no tienen la misma importancia, Provost y Fawcett, (2001).

Una función de coste esperado es aquella que pondera el porcentaje de los que devuelven el crédito y los que no ponderados por sus respectivos costes. Si llamamos C_e al coste esperado la función es la siguiente:

$$C_e = \pi_{no} C_I + \pi_{si} C_{II} \tag{3.43}$$

Donde π_{no} y π_{si} es la proporción de buenos y malos pagadores y C_I y C_{II} son los costes asociados a los errores de tipo I y II.

La complejidad existente en el cálculo de los costes asociados a los dos tipos de errores es considerable dado que los factores que los afectan son difíciles de cuantificar.

Algunos componentes de C_I es la pérdida del monto del crédito otorgado al que hay que restarle los ingresos recibidos antes de pasar a la situación de moroso u otros ingresos recibidos por valores de la propiedad asegurada en el momento de la

liquidación y sumar aquellos gastos que se deriven de costes legales, costes administrativos, etcétera.

Los costes asociados al error de tipo II (C_{II}) están asociados a la pérdida de los intereses que se generarían si se hubiera concedido el préstamo del buen pagador más la pérdida o beneficio de destinar este crédito no concedido a otro cliente. Estos costes asociados a este tipo de error se pueden llamar coste de oportunidad. El verdadero coste es que, si el solicitante del crédito es un cliente del banco y no se le concede, muy probablemente, deje de ser cliente. Si el peticionario del crédito no es cliente y no se le concede el crédito casi con toda seguridad ese demandante no llegue a ser cliente de la entidad financiera a quien ha dirigido su solicitud de dinero y, desde un punto de vista más práctico, para cuantificar estos costes deberíamos de contar con información de todos los productos financieros que dejaría de consumir a lo largo del ciclo de vida del cliente. Estos costos es muy probable que cambien con el tiempo por lo que se puede concluir que, aunque se puedan establecer unos rangos en los que probablemente estén los costes es prácticamente improbable el cálculo exacto de este tipo de coste.

Una de las escasas referencias que se disponen de una matriz de coste se encuentra en los datos del banco alemán que se descargan del repositorio de la UCI y que es la siguiente:

$$C_{ij} = \begin{pmatrix} 0 & 1 \\ 5 & 0 \end{pmatrix}$$

En esta matriz los costes asociados al error de tipo I estamos suponiendo que son cinco veces mayores que los costes que involucra el error de tipo II. Otros autores, como Altman, (1998), mantienen que entre los errores de ambos tipos hay una diferencia mayor, que se cifra entre 20 y 40. Las conversaciones mantenidas con varios responsables de Cajas de Ahorro y de bancos comerciales indican que el establecimiento de esta matriz de costes para modelos de credit scoring es de una complejidad considerable y la relación entre los costes asociados a los errores de tipo I y de tipo II puede abarcar un abanico muy amplio, dependiendo del tipo de crédito concedido y de la vinculación del prestatario con el banco fundamentalmente. Esta complejidad de estimar los costes asociados a las acciones que se toman al conceder o no el crédito y su posterior evolución se agrava aún más en épocas de crisis económicas.

En general se puede afirmar que los métodos de aprendizaje sensibles al coste suelen ser adaptaciones de algoritmos existentes, como los árboles de decisión (Ting, 1998). Sin embargo, existen estrategias que son independientes del algoritmo de aprendizaje utilizado. Estas estrategias, corrientemente denominadas de meta esquemas de aprendizaje, toman como entrada un algoritmo de aprendizaje, una colección de datos de entrenamiento y una distribución de costes, y generan un clasificador basado en el algoritmo de aprendizaje y adaptado a los costes de los errores. Entre los trabajos pioneros de clasificación sensible al coste se encuentran: Turney (1995 y 2000), Ting (1998), Elkan (2001), Zadrozny y Elkan (2001), y Lizotte (2003).

Para estos modelos con costes asimétricos podemos encontrar diferentes estrategias para abordar una correcta clasificación:

- **Basadas en un umbral**. Witten y Frank, (1999), aplicable a todo algoritmo cuya salida sea un clasificador que emite valores numéricos (como probabilidades, similitudes, etc.). La idea es la siguiente: si, por ejemplo, un clasificador L asigna la clase positiva a un cliente *d* a partir de un umbral *u* (es decir, cuando L>u), el umbral se ajusta para que el clasificador sea más o menos conservador, usando para ello una submuestra de instancias de entrenamiento reservados para este fin. Utilizando el umbral, Sheng y Ling (2006) proponen un método que llaman Thresholding que, en general, produce un coste más bajo de clasificaciones incorrectas. Los autores afirman que este algoritmo convierte cualquier método no sensible al coste en sensible. Thresholding puede elegir el mejor umbral que minimiza el coste total de los errores de clasificación.
- **Ponderando las instancias**. Modificar los pesos asignados a cada clase de manera que se le da más peso a los ejemplos asociados a cometer errores más costosos. Ting, (1998) propone dar pesos a cada cliente (Instance Weighting) pero un peso mayor a los individuos de una clase (por ejemplo a la clases de los que no devuelven el crédito), con el objetivo de que el algoritmo se fije especialmente en clasificar correctamente estos ejemplares, minimizando el error sobre ellos.
- **A través del algoritmo MetaCost** de Domingos (1999). Este método es aplicable a cualquier algoritmo de aprendizaje. Esta sofisticada técnica consiste en re etiquetar la colección de entrenamiento de acuerdo con la salida de un comité de clasificadores generados por el algoritmo base usando el método de bagging, y entrenar luego un clasificador sobre la colección re etiquetada.

Algunas aportaciones interesantes en los métodos de clasificación a través de los costes las podemos encontrar en Ling *et al.* (2004). Estos autores emplean árboles de decisión con coste mínimos de. La idea que subyace es introducir un factor de costes mientras se va construyendo el árbol de acuerdo con los criterios de división que minimizan el coste total, en lugar de minimizar la entropía. En este sentido, los árboles de decisión con costos mínimos y MetaCost son similares aunque hay una gran diferencia. En los árboles de decisión con un coste mínimo, la parte más sensible a los costos, se construye directamente en el clasificador mientras que el algoritmo MetaCost puede utilizar, no sólo los árboles de clasificación sino cualquier método de clasificación: redes neuronales, máquinas de vectores soporte, redes bayesianas, etcétera.

Otro enfoque importante es el Lopez *et al.* (2010) que utilizan reglas difusas para problemas de bases de datos no balanceados. Desde esta perspectiva. El aprendizaje sensible al coste, en las bases de datos que analizan, alcanza un buen equilibrio entre las clases, mejorando la clase positiva (sensibilidad) y no perjudicando la precisión de la clase considerada negativa (especificidad).

3.1.6.3.1. Algoritmo sensible al coste: Metacost.

Una vez especificada una matriz de coste existen, como hemos comentado anteriormente, varios algoritmos que recogen la información contenida en la matriz que nos ofrecen los resultados de la clasificación.

En esta tesis utilizamos uno de los más conocidos que es el Metacost de Domingos (1999) y que goza de una característica fundamental: es independiente de la técnica de clasificación que se utilice. El algoritmo tal y como se describe por el autor consta de tres pasos:

Notaciones:

Definimos a **S** como el conjunto de entrenamiento.
L es el algoritmo de aprendizaje que utilizamos para la clasificación.
C es una matriz de costes ya tenemos especificada.

m es el número de muestras a generar.
n es el número de instancias de las muestras .

Paso 1. Para todo **i** en el rango de 1 a **m**:

(a) Crear S_i como un remuestreo de S con n ejemplos.

(b) Crear modelos M_i aplicando el algoritmo de aprendizaje a S_i.

Paso 2. Para cada ejemplo x en S

(a) Para cada clase j creamos:

$$P(j|x) = \frac{1}{\sum_i 1} \sum_i P(j|x_i M_i) \tag{3.44}$$

(b) Cambiar la clase de x a la clase k que minimiza

$$\sum_j P(j|x) C(k, j)$$

Paso 3. Crear el modelo final M mediante la aplicación de L a S.

3.2. Técnicas de clasificación de datos.

3.2.1. Árboles de decisión.

3.2.1.1. Introducción.

El problema de la clasificación o discriminación puede abordarse de varias formas. Desde el punto de vista estadístico se dispone de un amplio conjunto de elementos que pueden venir de dos o más poblaciones diferentes. Se observa un conjunto de características que vienen recogidas en una variable p-dimensional. El problema de clasificación se convierte en prever nuevos elementos de acuerdo a la información disponible.

Los árboles de decisión o de clasificación no son modelos estadísticos basados en la estimación de los parámetros de una ecuación propuesta, no tenemos que estimar un modelo estadístico formal, son más bien algoritmos para clasificar utilizando particiones sucesivas, en general binarias, en los valores de una variable cada vez.

Esta técnica de clasificación es probablemente el modelo de clasificación más utilizado y popular, según Gehrke *et al.* (1999b) y Quinlan, (1986b). Existen algunas ventajas al utilizar esta técnica frente a otros modelos, Jiménez, (2002). "Una de las ventajas más sobresalientes de los modelos de árbol de decisión es su carácter descriptivo, que permite entender e interpretar fácilmente las decisiones tomadas por el modelo, ya que tenemos acceso a las reglas que se utilizan en la tarea predictiva (aspecto no contemplado en otras técnicas, como las RNA). Además, los algoritmos utilizados para generar este tipo de modelos suelen incluir la opción de conversión de las rutas de decisión establecidas en el árbol a reglas lógicas del tipo 'si...entonces'. Con esta conversión se puede conseguir una mejor comprensión, si cabe, de las reglas predictivas del modelo."

Los árboles de decisión son particiones secuenciales de un conjunto de datos que maximizan las diferencias de la variable dependiente. Nos ofrecen una forma concisa de definir grupos que son consistentes en sus atributos pero que varían en términos de la variable dependiente. Esta herramienta puede emplearse tanto para la resolución de problemas de clasificación como de regresión: árboles de clasificación y árboles de regresión.

Desde otro punto de vista podemos asegurar que los árboles de decisión o de clasificación son un modelo de predicción surgido en el ámbito del aprendizaje

automático (Machine Learning) y de la Inteligencia artificial (Artificial Intelligence) que, partiendo de una base de datos, crea diagramas de construcciones lógicas que nos ayudan a resolver problemas.

Mediante esta técnica se representan de forma gráfica un conjunto de reglas sobre las decisiones que se deben de tener en cuenta para asignar un determinado elemento a una clase (valor de salida).

A esta técnica también se le denomina, siguiendo a Escobar (2007), segmentación jerárquica y se puede encuadrar, como técnica multivariante entre los métodos de dependencia, dado que como en el resto de los métodos estadísticos se establece una distinción entre las variables que se pretender explicar y aquellas otras que se utilizan para explicar las anteriores. La segmentación se realiza a través de un proceso (algoritmo) que está basado en criterios para identificar grupos homogéneos de una población.

La segmentación jerárquica es una técnica explicativa y descomposicional que utiliza un proceso de división secuencial, iterativo y descendiente que partiendo de una variable dependiente que se pretende explicar, forma grupos homogéneos definidos específicamente mediante combinaciones de variables independientes en las que se incluyen la totalidad de los casos recogidos en la muestra.

Los modelos basados en árboles de clasificación suelen dar buenos resultados cuando muchas de las variables de clasificación son cualitativas. Sin embargo, algunos autores afirman que, en general, no son más eficaces que los procedimientos ofrecidos por la estadística clásica cuando las variables siguen distribuciones aproximadamente normales.

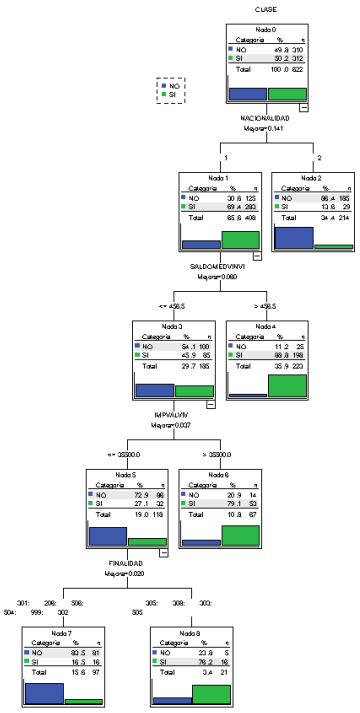
Diversos autores como Hernández *et al.* (2004) afirman que los árboles de decisión no sólo son adecuados para resolver problemas de clasificación sino que abordan eficientemente otras tareas como la regresión, el agrupamiento, o la estimación de probabilidades.

En los árboles de decisión se encuentran los siguientes componentes: nodos, ramas y hojas. Los nodos son las variables de entrada, las ramas representan los posibles valores de la variable de entrada y las hojas son los posibles valores de la variable de salida. Como primer elemento de un árbol de decisión tenemos el llamado nodo raíz que va a representar a la variable de mayor relevancia en el proceso de clasificación.

En el gráfico siguiente se muestra una partición del árbol que genera el algoritmo CART y, como se observa en la Figura 3.18, se ven cuáles son las variables de segmentación más importantes a la hora de solicitar un crédito. El objetivo es identificar las variables que ofrecen la mejor escisión entre los peticionarios del crédito. La mejor división se produce con la variable nacionalidad con un 65,4% de individuos españoles que solicitan el préstamo frente a un 34,4% de extranjeros. A continuación, considerando la variable sexo, el algoritmo encuentra que la mejor variable para dividir es, en los de nacionalidad española, la variable relacionada con el saldo medio mantenido en la entidad. El importe del valor de la vivienda y la finalidad del crédito. Así el procedimiento continúa hasta que no existen variables independientes o no existen escisiones significativas pendientes de realizar.

Como se verá en el epígrafe siguiente hay diferentes algoritmos que pueden generar diversas estructuras de árbol de decisión.

Figura 3.18. Ejemplo de árbol de clasificación. Método CART.



Fuente: Elaboración propia

3.2.1.2. Aplicabilidad de los árboles de decisión para clasificación.

En cuanto a la aplicabilidad de los árboles de decisión, en la construcción de árboles de decisión se han desarrollado varios métodos y cada uno de ellos ofrece diferentes

capacidades, en general, estos algoritmos son apropiados para problemas de clasificación que presenten las siguientes características:

- Cuando los ejemplos de aplicación vienen en forma de pares < atributos, valor>.
- Si la presentación de salida o función objetivo tiene valores discretos.
- Cuando es interesante el tipo de representación para la explotación posterior del modelo.
- Resulta conveniente si se necesitan descripciones disyuntivas.
- Los datos de aprendizaje pueden contener errores o valores nulos en algún atributo.

Esta técnica es la más utilizada por su sencillez. Los árboles de decisión ayudan a la toma de decisiones facilitando la interpretación de éstas, dado que resumen los ejemplos de partida permitiendo la clasificación de nuevos elementos siempre que no se alteren sustancialmente las condiciones iniciales. Los modelos de árboles de decisión, utilizados de forma exclusiva o en combinación con otras técnicas, se aplican a la resolución de numerosos problemas en el ámbito del marketing y, en general, son herramientas muy útiles en el control de la gestión empresarial, especialmente en las decisiones de segmentación de mercados, posicionamiento de productos y del comportamiento del consumidor, marketing directo, etcétera. Se ha empleado también en diagnósticos de enfermedades (clases) dependiendo de los síntomas (éstos representan en el modelo de árbol los atributos de entrada). También se utiliza para problemas de concesión de créditos, en la gestión de la relación con el cliente, CRM (Customer Relationship Management) y en otras múltiples actividades pertenecientes a las Ciencias Sociales en su conjunto.

3.2.1.3. Algoritmos de clasificación.

Los algoritmos que se encuentran, o bien solos o bien integrados en diferentes paquetes informáticos, son los que determinan o generan el procedimiento de cálculo que establece el orden de importancia de las variables en cada interacción. También se pueden imponer ciertas limitaciones en el número de ramas en que se divide cada nodo.

Los elementos y las herramientas de los algoritmos que determinan la construcción de un árbol son varios:

- El criterio para determinar la partición de cada nodo.

- La regla que declara un nodo terminal.
- La asignación de una clase a cada nodo terminal, lo que determina la regla de clasificación.
- Fusión: En relación a la variable dependiente, las categorías de las variables predictoras no significativas se agrupan juntas para formar categorías combinadas que sean significativas.
- Partición. Selección del punto de división. La variable utilizada para dividir el conjunto de todos los datos se elige por comparación con todas las demás.
- Poda. Se eliminan las ramas que añaden poco valor de predicción del árbol.
- La evaluación de la bondad del clasificador obtenido. La estimación de la validación del árbol y el cálculo del riesgo. Los métodos utilizados son los mismos independientemente del método que se utilice para la generación del *árbol.*

3.2.1.3.1. Particiones posibles y criterios de selección.

Lo más razonable para resolver el problema de la partición adecuada de un nodo es basarse en una tasa de error o coste de clasificación del nodo. El criterio del coste se determina a través de la denominada función de impureza, seleccionándose aquella partición que dé lugar al mayor decrecimiento de la impureza. Dependiendo de la función de impureza que se tome se tendrán distintos criterios de selección del corte óptimo. Los dos criterios más conocidos son el índice de Gini y el criterio de Twoing.

Cuando se ha definido el criterio de corte adecuado se obtiene, mediante particionamiento recursivo, sucesivas segmentaciones del conjunto de datos que cada vez se van haciendo más finas.

Una vez que ya se disponen de los criterios de partición y de asignación de cada una de las clases a cada nodo terminal el proceso terminará cuando se encuentre la regla que nos indique el instante en que el proceso de segmentación de los datos se detiene y se declara un nodo como terminal.

Los criterios de división están generalmente basados en criterios en medidas denominadas de impurezas de un nodo, entendiendo por impureza el grado en el que el nodo incluye casos de distintas clases. Así se define un nodo puro a aquel que sólo contiene casos que pertenecen a una única clase. Se considera la bondad de una partición como la medida del decrecimiento de la impureza que se consigue así la

maximización de la bondad es equivalente a la minimización de la impureza del árbol generado por la participación

La función de impureza, dado un problema de clasificación con J clases diferentes, suele ser no negativa y se define sobre las J-duplas (p_1, p_2,……p_J) donde cada valor representa la probabilidad de que un caso sea de la clase j en el subárbol actual.

La medida adaptada de Breiman *et al.* (1984) de impureza de un árbol T puede lograrse a través de las impurezas de sus hojas o nodos terminales ($\tilde{T}$ **)**

$$\phi(T) = \sum_{t \in \tilde{T}} p(t)\phi(t) \tag{3.45}$$

P(t) representa la probabilidad de que un registro dado corresponda a la hoja t y ϕ(t) es la impureza del nodo terminal t.

Cualquier función ϕ tiene las siguientes propiedades:

- ✓ Esta función ϕ posee un único máximo en (1/J, 1/J,….1/J) Esto quiere decir que la impureza de un nodo en máxima cuando los registros correspondientes a cada uno de las clases del problema es el mismo.
- ✓ La función es simétrica respecto a l conjunto de las J-duplas (p_1, p_2,……p_J).
- ✓ Un nodo se denomina puro cuando sólo contiene ejemplo de una clase (la función ϕ es igual a cero). En este caso la función ϕ alcanza sus J mínimos en ϕ(1,0,…0)….. ϕ(0,0,...1).

3.2.1.3.1.1. Ganancia de información.

Otras medidas intentan maximizar la ganancia de información que consigue el atributo A_i para ramificar el árbol de clasificación mediante la siguiente función I:

$$I(A_{ij}) = \sum_{j=1}^{Mi} p(A_{ij})H(C \mid A_{ij}) \tag{3.46}$$

La entropía es una medida de la incertidumbre que hay en un sistema, es decir, trata de medir ante una situación determinada la probabilidad de que ocurra cada uno de los posibles resultados. La entropía de clasificación se define como:

$$H(C_k \mid A_{ij}) = -\sum_{k=1}^{j} p(C_k \mid A_{ij}) \log_2 \, p(C_k \mid A_{ij}) \tag{3.47}$$

La ganancia de información que se produce al dividir T en los subconjuntos T_j viene dada por:

$$H(T) - \sum p(T_j) H(T) \tag{3.48}$$

Donde H(T) es la entropía de T.

3.2.1.3.1.2. El criterio de proporción de ganancia.

Se trata de normalizar el concepto de ganancia obtenida dado que este criterio posee el inconveniente de que favorece a los atributos o variables con muchos valores:

$$R(A_i) = \frac{H(C) - \sum_{j=1}^{M_i} p(A_{ij}) H(C \mid A_{ij})}{\sum_{j=1}^{M_i} p(A_{ij}) \log_2 (A_{ij})} \tag{3.49}$$

3.2.1.3.1.3. Índice de diversidad de Gini.

El índice de Gini es una medida de diversidad de las clases en un nodo del árbol que se utiliza. Este índice se emplea en diferentes algoritmos de árboles de clasificación:

$$G(A_i) = \sum_{j=1}^{M_i} p(A_{ij}) G(C \mid A_{ij}) \tag{3.50}$$

Siendo $G(C \mid A_{ij})$ igual a:

$$G(C \mid A_{ij}) = -\sum_{j=1}^{Mi} p(C_K \mid A_{ij}) \, p(1 - p(C_K \mid A_{ij})) \tag{3.51}$$

A_{ij} es el atributo empleado para ramificar el árbol, J es el número de clases, M_i es el de valores distintos que tiene el atributo A_i y $p(A_{ij})$ constituye la probabilidad de que A_i tome su *j*-ésimo valor y $p(C_k|A_{ij})$ representa la probabilidad de que un ejemplo sea de la clase C_k cuando su atributo A_i toma su *j*-ésimo valor.

El índice de diversidad de Gini toma el valor cero cuando un grupo es completamente homogéneo y el mayor valor lo alcanza cuando todas las $p(A_{ij})$ son contantes entonces el valor del índice es (J-1)/J

Existen otras medidas utilizadas por algunos autores: López de Mantaras (1991) sugiere una alternativa al criterio de normalización de proporción de ganancia que evita la fragmentación del conjunto de entrenamiento característica de algunas reglas de decisión. La métrica de distancia que propone es la siguiente:

$$LM(A_I) = \frac{H(C) - \sum_{j=1}^{M_i} p(A_{ij}) H(C \mid A_{ij})}{-\sum_{j=1}^{M_i} \sum_{k=1}^{J} \frac{n(C_k \mid A_{ij})}{N} \log_2 \frac{n(C_k \mid A_{ij})}{N}} \tag{3.52}$$

Otro trabajo que representa una alternativa al índice de Gini lo proponen Taylor y Silverman (1993) a cuya fórmula la llaman MPI (Mean Posterior Improvement):

$$MPI(A_i) = \prod_{j=1}^{M_i} p(A_{ij}) * \left(1 - \sum_{k=1}^{J} \frac{\prod_{j=1}^{M_i} p(C_k \mid A_{ij})}{P(C_k)} \right) \tag{3.53}$$

3.2.1.3.1.4. Otros criterios de selección.

En la literatura sobre este tema se pueden encontrar otras variaciones sobre estas medidas de impureza para que los casos de estudio no se comportan adecuadamente. En Bezal *et al.* (2001) encontramos dos medidas que son menos complejas: el criterio MaxDif y el índice generalizado de Gini (GG). Ambas medidas realizan una suma ponderada de las medidas de impureza de cada uno de los subárboles resultantes de ramificar el nodo actual del árbol

- MaxDif

$$D(A_i) = \sum_{j=1}^{M_i} p(A_{ij}) D(C \mid A_{ij}) \tag{3.54}$$

$$D(C \mid A_{ij}) = máx_k \{ p(C_k \mid A_{ij}) - p(1 - p(C \mid A_{ij})) \tag{3.55}$$

- Índice Generalizado de Gini

$$GG = \sum_{j=1}^{M_i} p(A_{ij}) GG(C \mid A_{ij}) \tag{3.56}$$

$$GG(C \mid A_{ij}) = 1 - máx_k \{ p(C_k \mid A_{ij}) \tag{3.57}$$

Estos mismos autores también proponen la utilización de un umbral de soporte mínimo para mejorar el comportamiento de los algoritmos de árboles de clasificación TDIDT (Top Down Induction Decision Trees) clásicos en presencia de ruido que nos sirva para no tener en cuenta, en la construcción del árbol, ramas poco pobladas.

Existen otros criterios a los anteriormente expuestos basados en el criterio de la impureza de los nodos que se adscriben a otras categorías, Martin, (1997): algunos de los criterios utilizan distancias o ángulos para ver las diferencias entre los diferentes subconjuntos, y otros criterios emplean medidas como la $\chi 2$ de Person entre los conjuntos de entrenamiento y las clases.

Tanto en Martin, (1997) como en Shih, (1999) se pueden encontrar estudios exhaustivos sobre distintas reglas de división.

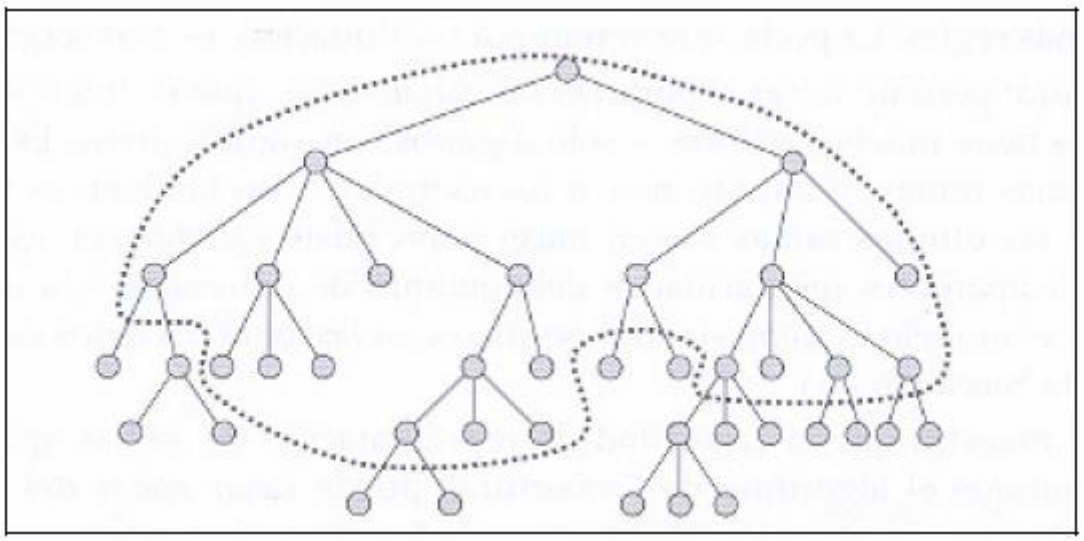
Es importante señalar que la mayor parte de las reglas de división que se han propuesto por los diferentes autores mejoran sólo de forma marginal la precisión de los árboles que se construyen pero tan sólo en situaciones muy concretas.

3.2.1.3.2. Poda en Árboles de clasificación.

Todos los algoritmos de aprendizaje de árboles de clasificación obtienen modelos más o menos complejos y consistentes respecto a la evidencia: cubre todos los ejemplos y los cubre de una forma que puede parecer óptima pero es demasiado ingenuo porque el modelo es simplemente una aproximación al concepto de aprendizaje y lo verdaderamente importante es que el modelo sirva para ejemplos nuevos, para clasificar bien al conjunto de test. Es especialmente importante si los datos contienen errores porque se ajustará el modelo a estos errores y esto perjudicará al comportamiento global del modelo, lo que se conoce como sobreajuste (overfitting), Hernández *et al.* (2004).

Para solucionar este problema es conveniente limitar el crecimiento del árbol modificando los algoritmos de aprendizaje de forma que se obtengan modelos más generales. Este proceso de poda se puede ver gráficamente en la siguiente ilustración:

Figura 3.19. Ejemplo de poda. Nodos inferiores eliminados.



Fuente: Hernández et al. (2004).

El concepto de poda en árboles de clasificación se puede dividir en dos métodos: prepoda y postpoda

Prepoda. Las reglas de parada tratan se preguntan si merece la pena seguir o detener el proceso de crecimiento del árbol por la rama actual. Se denominan reglas de prepoda ya que reducen el crecimiento y la complejidad del árbol mientras se está construyendo, diferenciándose de la reglas de post poda que se utilizan cuando ya se ha construido el árbol.

Se pueden citar tres estrategias como reglas de prepoda:

- ✓ Pureza del nodo. Si el nodo sólo contiene ejemplos o registros de una única clase de decide que la construcción del árbol ha finalizado. También se puede elegir un umbral de pureza y dejar para detener la realización del árbol de decisión.
- ✓ Cota de profundidad. Previamente a la cosntr4ucción se fija una cota que nos marque la profundidad del árbol que queremos. Cuando la alcanza se detiene el proceso.

- ✓ Umbral de soporte. Podemos parar el proceso si especificamos un número de ejemplos mínimo para los nodos ya que no consideramos fiables aquellos casos que no lleguen a alcanzar el valor fijado para el nodo.

Postpoda. Normalmente se realiza la poda del árbol una vez que este ha sido construido. Son aquellas ramas del árbol con menor capacidad las que suelen ser candidatas a ser podadas. Esta poda generalmente aumenta la capacidad de precisión del árbol. También hay que afirmar que la correcta estimación a priori del beneficio obtenido al simplificar el árbol durante su construcción resulta difícil y tan sólo se ha empleado en algunos algoritmos recientes como el denominado PUBLIC, Rastogi y Shim, (2000).

Existen dos formas de poda muy comunes utilizadas en los diferentes algoritmos: la poda por coste-complejidad y la poda pesimista.

En la poda por coste-complejidad se trata de equilibrar la precisión y el tamaño del árbol. La complejidad está determinada por el número de hojas que posee el árbol (nodos terminales)

Siguiendo la notación anteriormente utilizada llamamos T al árbol de clasificación, N al número de ejemplos de entrenamiento y M al número de instancias que se clasifican mal, entonces la medida coste-complejidad del árbol T para un parámetro de complejidad especificado α toma la siguiente expresión:

$$R_\alpha(T) = R(T) + \alpha l(T) \tag{3.58}$$

Donde R(T) es un estimador del error de T = M/N (porcentaje de instancias mal clasificadas) y $l(T)$ es el número de hoja del árbol. El parámetro *α* es desconocido.

El árbol óptimo podado será aquel que haga mínima la expresión $R_\alpha(T)$

A la hora de trabajar se genera una secuencia de árboles con los distintos valores del parámetro desconocido *α.* Tal y como se describe en Breiman *et al.* (1984) al aumentar *α* se tienden a podar menos nodos y de todos los árboles generados se escoge aquel que tenga asociado el menor error utilizando un conjunto de independiente del de entrenamiento o el método de validación cruzada.

La poda pesimista utilizada por algunos algoritmos de construcción de árboles, Quinlan, (1993) sólo utiliza el conjunto de entrenamiento para construir el árbol. Con los casos clasificados incorrectamente (E) se saca su error de sustitución (E/N). El error de sustitución de un árbol es la suma de los errores de sus hojas pero la probabilidad real del error cometido no se puede estimar exactamente pero se puede asimilar una distribución de probabilidad binomial de errores y de éxitos en N experimentos. Dado un nivel de confianza se puede establecer un intervalo de confianza para esta supuesta distribución binomial y se puede asimilar el límite superior del intervalo como cota del error en el nodo.

Para llevar a cabo la poda pesimista se podaría el árbol si el intervalo de confianza del error de resustitución incluye el error de resustitución del nodo si se trata como una hoja. Procediendo de esta forma se elimina los subárboles que no mejorar significativamente la precisión del clasificador. Esta heurística utilizada en este método suele producir buenos resultados.

3.2.1.3.3. Algoritmos para la construcción de árboles de clasificación.

A continuación se realiza una breve descripción de los principales algoritmos más utilizados por los diferentes investigadores y que podemos encontrar en la mayoría de los programas informáticos

3.2.1.3.3.1. Algoritmo AID.

El algoritmo AID (Automatic Interaction Detection) o Detección Automática de Interaciones fue uno de los más utilizados en la década de los años setenta y principios de los ochenta hasta que surgió el CHAID. Se le llama así porque la idea inicia no perseguía el objetivo de la clasificación sino que estaban centrados en las interacciones entre las variables.

Las primeras ideas de la segmentación AID fueron recogidas por Morgan y Sonquist (1963) que propusieron la utilización recursiva del análisis de la varianza con todos los pares posibles de las variables candidatas.

Este algoritmo presenta dos limitaciones muy importantes, derivadas, por una parte, del elevado número de elementos muestrales que requieren para efectuar los análisis y, por otra, de la carencia de un modelo explícito que explique o determine la relación existente entre la variable dependiente y las variables explicativas.

En el algoritmo AID las variables explicativas han de estar medidas en escalas nominales u ordinales y la variable a explicar, variable criterio o dependiente, puede medirse en un escala métrica (medida con una escala proporcional o de intervalo) o ficticia (dicotómica con valores 0 y 1).

El análisis AID constituye un análisis de la varianza secuencial que se realiza mediante divisiones dicotómicas de la variable dependiente que busca en cada etapa la partición entre las categorías de la variable independiente que maximiza la varianza intergrupos y minimiza la varianza intragrupos.

La agrupación de categorías se efectúa probando todas las combinaciones binarias posibles de las variables. Es la prueba estadística F la que se utiliza para seleccionar las mayores diferencias posibles:

La media cuadrática externa (MCE) que mide la heterogeneidad entre los grupos, es decir aquellas muestras generadas con los pronosticadores y que toma la siguiente expresión:

$$MCE = \sum_{j=1}^{J} \frac{(\bar{x}_j - \bar{x})}{J-1} \tag{3.59}$$

La siguiente medida calcula la heterogeneidad dentro de cada muestra, es decir compara a cada individuo del grupo con la media del grupo:

$$MCI = \sum_{j=1}^{J} \sum_{i=1}^{n_j} \frac{(x_{ij} - \bar{x}_j)}{n-J} \tag{3.60}$$

El cociente entre ambas fórmulas sigue una distribución F de Snedecor que se distribuye con J-1 grados de libertad bajo la hipótesis nula $H_0 = \mu_1 = \mu_2 = ...,\mu_J$

En este algoritmo, el proceso de subdivisión de la muestra en grupos dicotómicos continúa hasta que se verifica alguna de estas circunstancias:

- El tamaño de los grupos llega a un mínimo que se ha establecido de antemano.
- Las diferencias entre los valores medios de los grupos no son significativas, bien porque ninguna de las variables predictoras reduce significativamente la varianza residual, o bien porque los grupos son muy homogéneos y, por tanto, existe poca varianza intragrupos.

Las limitaciones de este algoritmo son importantes:

- Si se utilizan variables predictoras que difieren mucho en el número de categorías, el algoritmo tiende a seleccionar como más significativas y, por tanto como más explicativas, aquellas variables que posean un número más elevado de categorías.
- Las particiones resultantes dependen de la variable que es elegida en primer lugar, lo que condiciona las sucesivas particiones.
- El carácter exclusivamente dicotómico de las particiones. Particiones con tres o más ramas reducen más la varianza residual y, además, pueden permitir mejor una selección de otras variables.

3.2.1.3.3.2. Algoritmo CHAID.

Este algoritmo corrige muchas de las limitaciones del AID. Es un acrónimo de Chi-squared Automatic Interaction Detection (detector automático de interacciones mediante Ji cuadrado). Las ideas iniciales de Morgan y Sonquist (1963) fueron recogidas por otros autores que emplean, en lugar del análisis de la varianza, las tablas de contingencia y el estadístico χ^2. Algunos de estos primeros pioneros en utilizar esta técnica son Cellard *et al.* (1967), Bourouche y Tennenhaus (1972), Kass (1980) y Madgison (1989). Aunque fue diseñado para trabajar sólo con variables categóricas, posteriormente se incluyó la posibilidad de trabajar con variables categóricas, nominales, categóricas ordinales y variables continuas, permitiendo generar tanto árboles de decisión para resolver problemas de clasificación como árboles de regresión.

En este algoritmo los nodos se pueden dividir en más de dos ramas. La construcción del árbol se basa en el cálculo de la significación de un contrate estadístico como criterio para definir la jerarquía de las variables predictoras o de salida, al igual que para establecer las agrupaciones de valores similares respecto a las variables de salida a la vez que conserva inalterables todos los valores distintos. Todos los valores estadísticamente homogéneos son clasificados en una misma categoría y asignados a una única rama. Como medida estadística, si la prueba es continua, se utiliza la prueba F, mientras que si la variable predicha es categórica se utiliza la prueba Ji-cuadrado.

Para detectar si una relación es significativa se utiliza varios métodos diferentes dependiendo del tipo de variables implicadas: variable dependiente nominal, ordinal, o de intervalo.

Para el caso de que la variable dependiente sea nominal disponemos de dos test estadístico: el criterio de la χ^2 y la razón de verosimilitud (G^2).

Si cruzamos dos variables nominales, a una la llamamos X y a la otra Y, se construye la tabla de contingencia que estará formada por I filas (variable Y) y J columnas (Variable X) y en ella se encuentran las frecuencias conjuntas de ambas variables.

Tabla 3.5. Notación de una tabla de contingencia.

	X_1	X_2	..	X_I	
Y_1	n_{11}	n_{12}	..	n_{1I}	$n_{1.}$
Y_2	n_{21}	n_{22}	..	n_{2I}	$n_{2.}$
:	:	:	:	:	:
Y_J	n_{J1}	n_{J2}	..	n_{JI}	$n_{J.}$
	$n._1$	$n._2$	..	$n._I$	n

Las frecuencias marginales para cada uno de los valores *j* se obtienen a través del siguiente sumatorio:

$$n_{.j} = \sum_{i=1}^{I} n_{ij} \tag{3.61}$$

Igualmente se calculan las frecuencias marginales de los valores *i* con la fórmula siguiente:

$$n_{i.} = \sum_{j=1}^{J} n_{ij} \tag{3.62}$$

Si las categorías de la variable X y las categorías de la variable Y son independientes se cumple la siguiente condición:

$$P(I \cap J) = P(I) * P(J) \tag{3.63}$$

Las frecuencias esperadas debido a la independencia toman la siguiente expresión:

$$n_{ij}^{*} = n\frac{n_{i.}}{n}\frac{n_{.j}}{n} = \frac{n_{i.}n_{.j}}{n} \tag{3.64}$$

Una vez se hayan calculado las frecuencias empíricas y las teóricas podemos dos test estadísticos muy similares:

Definimos los residuos estandarizados a través de la siguiente expresión:

$$r_{ij}^{s} = \frac{n_{ij} - n_{ij}^{*}}{\sqrt{n_{ij}^{*}}} \tag{3.65}$$

El estadístico basado en la distribución de la χ^2 de Pearson adopta la siguiente expresión:

$$\chi^2 = \sum_{j=1}^{J}\sum_{i=1}^{I}\frac{(n_{ij} - n_{ij}^{*})^2}{n_{ij}^{*}} \tag{3.66}$$

Otra medida utilizada para la verificación entre las variables pronosticadoras (independientes) y la variable clase (dependiente) es el estadístico razón de verosimilitud que se fundamenta en el criterio de máxima verosimilitud, Haberman (1978) y Goodman (1979) que se calcula a través de la siguiente fórmula:

$$G^2 = 2\sum_{j=1}^{J}\sum_{i=1}^{I} n_{ij}\ln\frac{n_{ij}}{n_{ij}^{*}} \tag{3.67}$$

Escobar (2007) afirma que en el trabajo de comparación de modelos es el contraste a través de la G2 el que ofrece ventajas adicionales a la χ^2. Aunque los resultados son muy similares las ventajas se derivan de que la G^2 se calcula como una diferencia de las razones de verosimilitud entre dos modelos: el modelo saturado compuesto por efectos medios η de fila de columna y de asociación frente al de independencia donde sólo se consideran los efectos de fila y de columna de la tabla:

$$n_{ij} = \eta\tau^{A}_{i}\tau^{B}_{i}\tau^{AB}_{i} \tag{3.68}$$

$$n_{ij}^{*} = \eta\tau^{A}_{i}\tau^{B}_{i} \tag{3.69}$$

Si la variable dependiente toma la forma ordinal se puede considerar un contraste diferente donde se sólo se consideren los efectos columna de acuerdo a los trabajos de Goodman (1979) y de Madgison (1992) ya que son los únicos efectos que representan a la variable ordinal. La expresión del modelo es la siguiente:

$$n_{ij}^{'} = \eta \tau_{i}^{A} \tau_{i}^{B} \delta_{i}^{j} \tag{3.70}$$

Donde δ_i^j es un parámetro distinto para cada valor de la variable independiente.

Si se emplea este modelo sólo es adecuada la utilización del estadístico G^2 que ahora toma la siguiente expresión:

$$G^2 = 2\sum_{j=1}^{J}\sum_{i=1}^{I} n_{ij}^{'} \ln \frac{n_{ij}^{'}}{n_{ij}^{*}} \tag{3.71}$$

El algoritmo de segmentación de CHAID tiene tres fases: fusión, partición y detención. En la fase de fusión, cada predictor o variable independiente funde las categorías no significativas.

En la fase de partición para las variables independientes que tengan un valor p de Bonferroni ajustado significativo hay que separar el grupo del predictor que tenga el menor valor p. Cada una de las categorías mezcladas del predictor se convierte en un nuevo subgrupo del grupo padre, si ningún predictor tuviese un valor significativo entonces no separar el grupo. La fase de detención se produce cuando se analizan todos los subgrupos o cuando contengan un número demasiado bajo de casos.

El ajuste de Bonferroni, Kass (1980) y Hawing y Kass, (1982) establece que cuando se hagan B pruebas de los contraste de significación, la significación total (p_T) debe ser menor o igual a la suma de cada una de los contrastes efectuados (p_i).

$$p_T \leq \sum_{i=1}^{B} p_i \tag{3.72}$$

El número de las posibles combinaciones de las pruebas de significación (B) se calcula a través de las fórmulas de la combinatoria. Escobar (2007) contempla tres posibilidades):

Si la opción escogida es sin restricciones el número de pruebas para k grupos la fórmula utilizada es:

$$B_n = \sum_{i=0}^{k-1} (-1)^i \frac{(k-i)}{i!(k-i)!} \tag{3.73}$$

Para variables dependientes ordinales utilizando una función monótona el número de pruebas para formar k grupos también depende de del número de categorías c de la variable:

$$B_o = \binom{c-1}{k-1} \tag{3.74}$$

Si los casos perdidos se pueden fusionar con cualquier número de variables, los contrastes que se efectúan atienden a la siguiente expresión

$$B_{om} = \binom{c-1}{k-1} \frac{k-1+k(c-k)}{c-1} \tag{3.75}$$

Las ventajas del algoritmo CHAID se presentan a continuación:

- El método identifica aquellas clases o perfiles de las variables explicativas que no difieren desde el punto de vista estadístico respecto de la variable dependiente uniéndolas en el mismo nodo.
- El resultado no tiene que ser dicotómico dado que el algoritmo mantiene todas las categorías que son heterogéneas.
- El algoritmo posibilita la supresión de variables no significativas de forma segura.
- Permite conocer las variables que mantienen una fuerte interacción entre ellas.
- Cuando hay una fuerte correlación entre las variables predictoras, si se selecciona una de ellas altamente correlacionada con otras sólo se considera una. Esto supone la unión de variables desde el punto de vista de su impacto explicativo.

3.2.1.3.3.3 Algoritmo CART.

El algoritmo CART es el acrónimo de Classification And Regression Trees (Árboles de decisión y de regresión) fue diseñado por Breiman *et al.* (1984). Con este algoritmo se generan árboles de decisión binarios lo que quiere decir que cada nodo se divide en

exactamente dos ramas. Este modelo admite variables de entrada y de salida nominales, ordinales y continuas por lo que se pueden resolver tanto problemas de clasificación como de regresión.

Este algoritmo utiliza el índice de Gini para calcular la medida de impureza definido en la ecuación (3.1.6)

Este algoritmo emplea otra medida para evaluar el poder clasificador de una variable Para casos dicotómicos se trata de ver cuán diferentes son las probabilidades de los valores de la variable dependiente en cada uno de los grupos generados por el procedimiento clasificador. Esta medida en Breiman, (1984) llamada índice binario toma la siguiente expresión:

$$\phi(s,t) = \frac{p_L p_R}{4}\left[\sum_{j=1}^{J}\left|p(j\mid t_L) - \left|p(j\mid t_R)\right.\right.\right]^2 \tag{3.76}$$

Si *j* sólo toma dos valores podemos contemplarlo como un promedio al cuadrado de las diferencias absolutas de los porcentajes que presentan los dos segmentos candidatos para dividirse multiplicados por el producto de las proporciones de casos que se encuentran en cada uno de los segmentos. La expresión es la siguiente:

$$\phi(s,t) = \left[\sum_{j=1}^{2}\frac{\left|p(j\mid t_L) - \left|p(j\mid t_R)\right.\right.}{2}\right]^2 p_L p_R \tag{3.77}$$

Si la variable es cuantitativa lo que implica que estamos trabajando con árboles de regresión se emplean las fórmulas propias del cálculo de la varianza similares a las utilizadas en el algoritmo AID.

Este cálculo de la varianza del nodo parental puede explicarse de la siguiente manera:

$$S^2(t) = \frac{\sum_{i=1}^{n(t)}(y_i - \bar{y}(t))^2}{n(t)} \tag{3.78}$$

Lo que interesa en este análisis es estudiar si el predictor mejora la homogeneidad de los grupos que se forman tras la partición de la muestra en dos y no el cálculo de la heterogeneidad en sí misma, para lo cual se resta a la varianza del nodo parental

$S^2(t)$ a las de los grupos filiales formados ($S^2(t_L)$ y $S^2(t_R)$) multiplicados por la proporción de casos que existen en las particiones (p_L y p_R):

$$\Phi(S,T) = S^2(T) - p_L S^2(t_L) - p_R S^2(t_R) \tag{3.79}$$

También se pueden emplear las desviaciones medias, cuyas fórmulas son las siguientes:

$$Dm(t) = \frac{\sum_{i=1}^{n(t)} |y_i - \bar{y}(t)|}{n(t)} \tag{3.80}$$

$$\Phi(S,T) = Dm(t) - p_L Dm(t_L) - p_R Dm(t_R) \tag{3.81}$$

3.2.1.3.3.4. *A*lgoritmo QUEST.

Este procedimiento denominado QUEST es el acrónimo de Quick, Unbiased, Efficient Statistical Tree (árbol estadístico eficiente, insesgado y rápido). Este método fue propuesto por Loh y Shih (1997) que retoma las ideas previas contenidas en el trabajo de Loh y Vanichsetakul (1988) y le añaden diversas mejoras.

Este algoritmo trata de corregir y de restringir la exhaustiva búsqueda de particiones significativas que se generan tanto en los algoritmos AID y CHAID como en el CART.

Este método selecciona de forma previa la variable que segmenta mejor los datos y después realiza la división óptima de ella. Sintetizando el procedimiento, primero se elige la mejor variable predictora cuyo objetivo es que el número de categorías que poseen las variables no afecte a la elección de la mejor variable, para realizar después la mejor segmentación de la variable que ha seleccionado.

Este método CHEST sólo puede ser utilizado si la variable de salida es categórica nominal.

Además de empezar el proceso de segmentación con la selección de variables en vez de con la fusión de categorías se procede después a la mejor división de los valores de la variable elegida. Otros cambios propuestos en este algoritmo es la eliminación de la poda, la transformación de las variables cualitativas en cuantitativas. a través del procedimiento CRIMCOORD, un cambio en los valores perdidos de los clasificadores en los distintos nodos. Además, el algoritmo contiene la posibilidad de construir

particiones no binarias como CHAID y, a semejanza del método CART, el rechazo a la validación cruzada propuesta por Breiman *et al.* (1984). Respecto a estos algoritmos la diferencia está en la forma de particionar los nodos.

Los autores propusieron una clasificación arbórea basada en el análisis discriminante a la que llamaron FACT (Fast Algorithm for Classification Trees). Así una vez que se ha seleccionado la variable se procede a ver cuál es la mejor partición binaria del nodo donde nos podemos encontrar en alguno de los casos siguientes:

A. Si la variable dependiente tiene J categorías y necesitamos reducirla a 2 se realiza a través de un procedimiento de conglomerados K-means de Hartigan y Wong (1979). Como centros de los conglomerados se escogen las medias muestrales de los pronosticadores más extremos y para cada media adicional se calcula la distancia cuadrática a los centros anteriormente elegidos y se agrupan al más cercano. Se vuelven a recalculara los centros y se vuelve a asignar un grupo dependiendo de la proximidad a los nuevos centros recalculados.
B. Si la variable elegida es nominal hay que convertirla en un vector de variables ficticias empleando el análisis discriminante que convierte cada valor discreto en otro continuo con valores entre -1 y +1. El valor asignado es la puntuación discriminante que se realiza de la siguiente manera:

Suponemos que X es un variable categórica que toma los siguientes valores { c_1, c_2,…c_M}. Como se ha indicado cada valor de X es transformado primero en una variable M-multidimensional 0-1 que es un vector columna v = (v_1, v_2,…v_M)$'$ donde todos los valores son ceros excepto el componente *l* – ésimo el cual es igual a 1, donde *l* es definido implícitamente a través de X = c_l.

$V_i^{(j)}$ especifica la i – ésima observación de v en la j - ésima clase y define el M-dimensional vector columna:

$$\bar{v}^{(j)} = N_j^{-1}\sum_{i=1}^{N_j} v_i^{(j)}\ ,\ \bar{v} = N^{-1}\sum_{i=1}^{N_j} v_i^{(j)} \tag{3.82}$$

Definimos las siguientes matrices de orden M x M:

$$B = \sum_{j=1}^{J} N_j(\bar{v}^{(j)} - \bar{v})(v^{(j)} - \bar{v})' \tag{3.83}$$

$$W = \sum_{j=1}^{J} \sum_{i=1}^{N_j} (v_i^{(j)} - \bar{v}^{(j)})(v_i^{(j)} - \bar{v}^{(j)})' \tag{3.84}$$

$$T = \sum_{j=1}^{J} \sum_{i=1}^{N_j} (v_i^{(j)} - \bar{v})(v_i^{(j)} - \bar{v})' \tag{3.85}$$

Donde T= B+W

Se trata de hallar la proyección ***a'v*** que maximice la suma de cuadrados de la razón entre clases/intra clases:

$$\frac{a'Ba}{a'Wa} \tag{3.86}$$

El valor a se corresponde con el autovector que está asociado con el mayor autovalor de la matriz W'B siempre que la matriz inversa de W exista, Mardia *et al*. (1979)

Una vez que se ha dicotomizado la variable dependiente, si es el caso, y se han calculado las puntuaciones discriminantes se aplica ahora un análisis discriminante cuadrático para producir una división de la muestra por encima o por debajo de un valor calculado *d*. Tal y como se describe en Loh y Shih (1997) los pasos a seguir son los siguientes:

Definimos a $\bar{x}_A$ y S_A^2 como la media y la varianza de los elementos del grupo A y similarmente $\bar{x}_B$ y S_B^2 representan la media y la varianza de la otra clase B.

Sabemos que

$$p(A|t) = \sum_{j \in A} p(j|t) \text{ y } p(B|t) = 1 - p(A|t) \tag{3.87}$$

Tomando logaritmos a ambos lados de la ecuación obtenemos:

$$p(A|t) S_A^{-1} \phi\{(x - \bar{x}_A)/S_A = p(B|t) S_B^{-1} \phi\{(x - \bar{x}_B) \tag{3.88}$$

La solución para encontrar el punto d que nos separe los grupos es necesario resolver la ecuación de segundo grado $ax^2 + bx + c = 0$ donde los coeficientes toman las siguientes expresiones:

$$a = S_A^2 - S_B^2 \tag{3.89}$$

$$b = 2(\bar{x}_A S_A^2 - \bar{x}_B S_B^2) \tag{3.90}$$

$$c = (\bar{x}_B S_A)^2 - (\bar{x}_A C)^2 + 2S_A^2 S_B^2 \log\{p(A|t)S_B\}/\{p(B|t)S_A\} \tag{3.91}$$

Los diferentes casos que se pueden presentar son:

Si a = 0 y $\bar{x}_A \neq \bar{x}_B$ sólo existe una raíz dada por la siguiente expresión:

$$d = (\bar{x}_A + \bar{x}_B)/2 - (\bar{x}_A - \bar{x}_B)^{-1} S_A^2 \log\{p(A|t\}/\{p(B|t)\} \tag{3.92}$$

La ecuación de segundo grado no tiene solución si a = 0 y $\bar{x}_A = \bar{x}_B$

Si $a \neq 0$ Entonces nos encontramos con dos posibilidades:

Si el discriminante $b^2 - 4ac$ de la fórmula de resolución de la ecuación $d = \frac{-b \pm \sqrt{b^2 - 4ac}}{2}$ es menor que cero $d = (\bar{x}_A + \bar{x}_B)/2$

Si $b^2 - 4ac > 0$, que se verifica siempre que se cumple la igualdad p(A|t) = p(B|t) obtenemos dos soluciones diferentes en la ecuación y escogemos aquella que esté más próxima a $\bar{x}_A$.

3.2.1.3.3.5. El algoritmo C5.

El algoritmo C5 y, sobre todo, su versión no comercial, C4.5 es uno de los algoritmos más utilizados en el ámbito de los árboles de clasificación.

La forma de inferir árboles de decisión a través de este algoritmo es el resultado de la evolución del algoritmo C4.5 (Quinlan, 1993) diseñado por el mismo autor y que a su vez es el núcleo del programa pertenece a la versión ID3 (Quinlan, 1986).

Este algoritmo crea modelos de árbol de clasificación, permitiendo sólo variables de salida categórica. Las variables de entrada pueden ser de naturaleza continua o categórica.

El algoritmo básico ID3 construye el árbol de decisión de manera descendente y empieza preguntándose, ¿qué atributo es el que debería ser colocado en la raíz del

árbol? Para resolver esta cuestión cada atributo es evaluado a través de un test estadístico que determina cómo clasifica él solo los ejemplos de entrenamiento. Cuando se selecciona el mejor atributo éste es colocado en la raíz del árbol. Entonces una rama y su nodo se crea para cada valor posible del atributo en cuestión. Los ejemplos de entrenamiento son repartidos en los nodos descendentes de acuerdo al valor que tengan para el atributo de la raíz. El proceso se repite con los ejemplos para seleccionar un atributo que será ahora colocado en cada uno de los nodos generados. Generalmente el algoritmo se detiene cuando los ejemplos de entrenamiento comparten el mismo valor para el atributo que está siendo probado. Sin embargo es posible utilizar otros criterios para finalizar la búsqueda:

- Cobertura mínima de tal forma que el número de ejemplos por cada nodo está por debajo de cierto umbral.
- Pruebas estadísticas para probar si las distribuciones de las clases en los sub-árboles difieren significativamente.

Una de las maneras de cuantificar la bondad de un atributo consiste en considerar la cantidad e información que proveerá ese atributo tal y como está definido en la teoría de la información. Por tanto, este algoritmo está basado en el concepto de "ganancia de información". El C4.5 modifica el criterio de selección del atributo empleando en lugar de la ganancia, la razón de ganancia. Para definir este concepto necesitamos definir el concepto de entropía.

Si el conjunto de los registros de la base de datos T se agrupan en función de las categorías de la variable de salida S, obteniéndose una proporción p_k para cada grupo asociado a un posible resultado o categoría, la función de entropía, particularizándola en el caso del credit scoring con dos atributos de salida, se concede el crédito con probabilidad *p*, o no se concede con su probabilidad complementaria, *1-p*, y de acuerdo a la ecuación (3.1.3) toma la siguiente expresión:

$$INFO(T) = p * \log_2(p) + (1-p)\log_2(1-p) \qquad (3.93)$$

Ahora se puede expresar la ganancia de información teniendo en cuenta una variable de entrada, según ecuación (3.1.4):

$$GANANCIA(X,T) = INFO(T) - INFO(X,T) \qquad (3.94)$$

Donde

$$INFO(X,T) = \sum_{i=1}^{k} \frac{T_i}{T} * INFO(T_i) \quad (3.95)$$

INFO(X,T) nos proporciona la información aportada por la variable de salida S cuando se tiene en cuenta una variable de entrada X.

INFO(X,T_i) es la entropía de la variable de salida S en cada subconjunto T_i determinado por la k categorías (dos en la base de datos considerada en la tesis doctoral) de la variable de entrada X. T_i es el número de registros asociados a una categoría i de la variable X.

El concepto de ganancia representa la diferencia necesitada para identificar la categoría destino asociada a un elemento T y la información necesitada para identificar dicha categoría cuando se conoce el valor de una variable de entrada para ese mismo elemento, lo que esto significa es que esa variable mostrará menor incertidumbre a la hora de clasificación que el resto de variables de entrada. En el ejemplo la variable NACIONALIDAD es la que menor incertidumbre presenta o la que tiene mayor ganancia de información, por lo que será la variable que constituirá el nodo raíz.

La ganancia de información posee el inconveniente de que favorece a los atributos o variables con muchos valores por lo que este algoritmo calcula la medida siguiente:

$$GAINRATIO(X,T) = \frac{GANANCIA(X,T)}{SPLITINFO(X,T)} \quad (3.96)$$

Donde

$$SPLITINFO(X,T) = -\sum \frac{T_i}{T} * \log_2\left(\frac{T_i}{T}\right) \quad (3.97)$$

$SPLITINFO(X,T)$ es la información aportada por la división (split) del conjunto de registros T a partir de los valores de la variable de entrada.

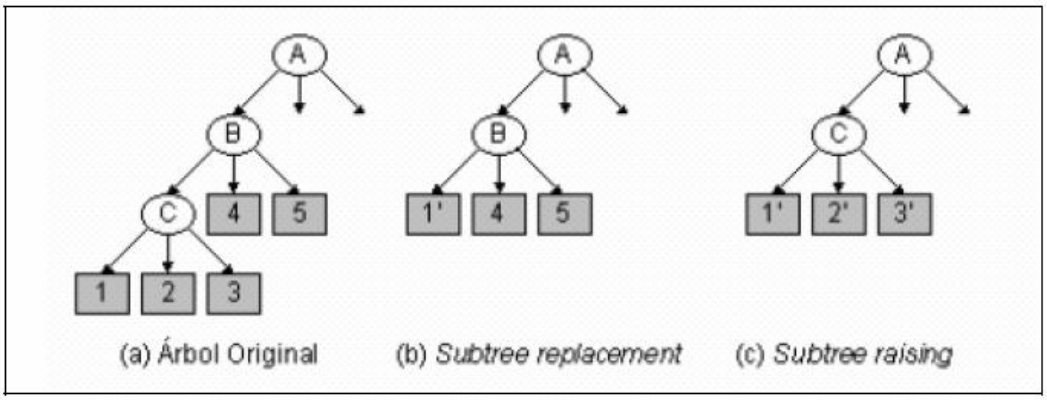
Este proceso se itera para cada una de las ramas descendientes ciñéndonos únicamente al total de registros asociados a cada rama y con las variables de entrada distintas a las utilizadas en el nodo raíz. El proceso para una rama concreta termina

cuando todos los registros de esa rama quedan perfectamente clasificados en una de las categorías de la variable de salida.

Una de las incorporaciones más novedosas de este algoritmo es la inclusión de la técnica "boosting" para la generación y combinación de múltiples modelos de clasificación. Otras interesantes aportaciones realizadas en el algoritmo C5 son que permite aplicar diferentes costes a los errores de clasificación, que se admiten ahora formatos nuevos de datos, por ejemplo fechas, horas, etc. y que se ha añadido la posibilidad de suprimir ciertos atributos marginales antes de construir el árbol para así poder reducir la dimensionalidad de la base de datos.

Los tres algoritmos más empleados por parte de la comunidad científica son el C4.5, CART y CHAID.

Figura 3.20. Tipos de operaciones de poda en C.4.5.



Fuente: Molina y García (2006).

3.2.1.3.3.6. Otros algoritmos de clasificación.

Algoritmo de construcción de árboles consolidados.

El algoritmo CTC (Construcción de árboles consolidados), Pérez (2006), se basa en las técnicas de remuestreo para construir el árbol consolidado. Primero el algoritmo genera un conjunto de muestras, posteriormente lo que hace este algoritmo es que en cada nodo va construyendo un árbol C4.5 asociado a cada muestra. Por medio de un consenso entre una serie de submuestras, eligen la variable más prometedora por la que hay que dividir ese nodo, la variable consolidada. Es decir, a cada muestra se le va a realizar un proceso por el cual decidirá cuál es la variable por la que esa muestra quiere dividir. Ese proceso está basado en un árbol de clasificación estándar como es el C4.5 (J48 en WEKA). Posteriormente teniendo todas las variables por las que las

submuestras quieren dividir se realizará una votación entre todas las variables "candidatas" y se elegirá la variable más votada, variable consolidada. Tras elegirla, todas las submuestras se dividirán obligatoriamente por esa variable consolidada, tras este paso diremos que el nodo ha sido consolidado. El proceso acabará cuando todos los nodos del árbol hayan sido procesados.

Random Forest.

La clasificación Random Forest emplea el algoritmo descrito por Breiman (2001). Este algoritmo está basado en la combinación de árboles de decisión independientes generados a partir de un vector de muestreo aleatorio que usa la misma distribución para todos los árboles de estudio. El término Random Forest se toma de la primera propuesta realizada por Ho (1995)

Este algoritmo está considerado como un clasificador bastante preciso. Trabaja bien aunque haya datos perdidos y ofrece un método para la interacción de las variables.

Decision Stum.

Es un algoritmo muy sencillo que genera un árbol de decisión de un único nivel. Utiliza un único atributo para construir el árbol de decisión. Para la selección de este atributo el algoritmo se basa en el criterio de la ganancia de información.

Admite tanto atributos numéricos como simbólicos y deben tenerse en cuenta cuatro posibles posibilidades cuando se calcula la ganancia de información, Molina y García (2006): que sea un atributo simbólico y la clase sea simbólica o que la clase sea numérica, o que sea un atributo numérico y la clase sea simbólica o que la clase sea numérica.

Finalmente, en la tabla 3.3. se ofrece una comparación entre los principales algoritmos clásicos de los árboles de clasificación mostrando algunas de sus características más relevantes:

Tabla 3.6. Características de los principales algoritmos.

CARACTERÍSTICAS DE LOS PRINCIPALES ALGORITMOS DE ÁRBOLES DE DECISIÓN						
Algoritmo	**Variables predictoras**	**Tipo de división**	**Criterio de División**	**Casos *missing***	**Método de Poda**	**Implementación**
CART (1984)	Continuas/ Discretas	Binaria	Impureza (*Gini index*)	SI	Post-	Libre Comercial
ID3 (1979)	Discretas	*n*-aria	Ganancia de información (Entropía)	NO	NO	Comercial
C4.5 (1993)	Continuas/ Discretas	Binaria/ *n*-aria	*Gain ratio* (Entropía)	SI	Pre-/Post-	Libre Comercial
J4.8	Continuas/ Discretas	Binaria/ n-aria	*Gain ratio* (Entropía)	SI	Pre-/Post-	Libre (Weka)
C5.0	Continuas/ Discretas	Binaria/ n-aria	*Gain ratio* (Entropía)	SI	Pre-/Post-	Comercial
CHAID (1975)	Discretas	*n*-aria	X^2	SI	Pre- (nivel de significancia)	Comercial

Fuente: Pérez (2006).

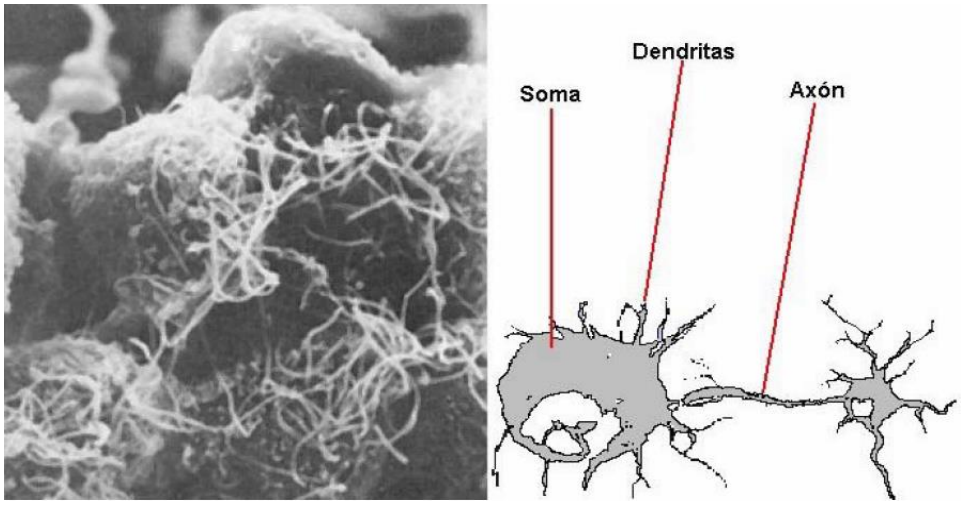
3.2.2. Redes Neuronales.

En la actualidad, las redes neuronales artificiales (RNA) constituyen un campo muy activo, fecundo y multidisciplinar. Tal y como se ha descrito en el primer capítulo uno, en el estado del arte de los modelos de credit scoring son variadas aplicaciones las que se ha desarrollado con este método comparando su efectividad con otros métodos de clasificación.

En los últimos 25 años las redes neuronales artificiales han irrumpido como una potente herramienta estadística tanto para problemas de clasificación, como de regresión o de agrupamiento. Su habilidad para procesar bases de datos con ruido o incompletas y su tolerancia a fallos permiten a estas redes operar en tiempo real por su operatividad en paralelo.

La principal virtud de una red neuroanal del tipo Perceptron Multicapa (Multilayer Perceptron) que explica su amplia utilización como técnica en el análisis de datos es que es un aproximador universal de funciones. La base matemática de esta afirmación se debe a Kolmogorov (1957). Un Perceptrón conteniendo al menos una capa oculta con suficientes unidades no lineales, tiene la capacidad de aprender virtualmente cualquier tipo de relación siempre que pueda ser aproximada en términos de una función continua (Cybenko, 1989; Funahashi, 1989; Hornik *et al.* (1989).

Figura 3.21. Micrografía ampliada de un cúmulo de neuronas y esquema de la misma.



En la parte izquierda de la figura 4.1 se puede observar un cúmulo de neuronas en el cerebro humano. Micrografía ampliada en 15.000 aumentos. En la parte derecha se muestra un esquema de una neurona. (Fuente: Brain Research Institute. UCLA en SAGAN 1980).

Las redes neuronales tratan de emular el comportamiento cerebral y están inspiradas en la estructura y funcionamiento de las redes neuronales biológicas.

Las diferencias que separan a ambas redes neuronales quedan reflejadas en el siguiente cuadro:

Tabla 3.7. Comparación del cerebro con un ordenador convencional.

	Cerebro	**Ordenador**
Velocidad de procesamiento	10^{-3} s	10^{-9} s
Modo de procesamiento	Paralelo	Serie
Número de procesadores	10^{11}	Pocos
Tipo de control del proceso	Democrático	Dictatorial
Conexiones	10000 por procesador	Pocas
Almacenamiento del conocimiento	Distribuido	En posiciones precisas
Tolerancia a fallos	Amplia	Poca o nula

Fuente: Nelson, M. M., & Illingworth, W. T. (1991).

Una red neuronal puede describirse mediante cuatro conceptos: el tipo de modelo de red neuronal; las unidades de procesamiento que recogen información, la procesan y arrojan un valor; la organización del sistema de nodos para transmitir las señales desde los nodos de entrada a los nodos de salida y, por último, la función de aprendizaje a través de la cual el sistema se retroalimenta.

3.2.2.1. Tipos de modelos de redes neuronales.

Existen actualmente más de 40 paradigmas de redes neuronales artificiales Se estima que tan sólo cuatro arquitecturas: el modelo perceptrón multicapa (MLP), los mapas autoorganizados de Kohonen, (SOFM), el vector de cuantificación (LVQ) y las redes de base radial (RBF) cubren, aproximadamente, el 90% de las aplicaciones prácticas de redes neuronales. El modelo más utilizado es el perceptrón multicapa, que abarca el 70% de las aplicaciones, dado que se ha demostrado que este modelo es un aproximador universal de funciones (Funahashi 1989).

El primer investigador que estudió el cerebro como una forma de ver el mundo de la computación fue Alan Turing, pero los primeros teóricos que concibieron los fundamentos de la computación neuronal fueron el neurofisiólogo Warren McCulloch y el matemático Walter Pitts, En 1949, Donald Hebb publica un importante libro titulado

"La organización del comportamiento" en el que establece una clara conexión entre psicología y fisiología y desarrolla un regla de cómo ocurría el aprendizaje.

Estos antecedentes sirven para que en 1957, Frank Rosenblatt desarrolle el perceptrón, que es la red neuronal más antigua, utilizándose hoy en día en aplicaciones como reconocedor de patrones. Empezando con este autor, a continuación se representa una relación de las principales redes neuronales con información sobre el tipo de aprendizaje utilizado, el año de creación y sus autores:

Tabla 3.8. Clasificación de las RNA más conocidas.

1. Supervisado
    1. Con conexiones feedforward
        - Lineales
            - Perceptrón (Rosenblatt, 1958)
            - Adaline (Widrow y Hoff, 1960)
        - Perceptrón multicapa (Multilayer perceptron) (MLP)
            - Backpropagation (Rumelhart, Hinton y Williams, 1986)
            - Correlación en cascada (Cascade correlation) (Fahlman y Lebiere, 1990)
            - Quickpropagation (Quickprop) (Fahlman, 1988)
            - Delta-bar-delta (Jacobs, 1988)
            - Resilient Propagation (RPROP) (Riedmiller y Braun, 1993)
            - Gradiente conjugado (Battiti, 1992)
        - Radial Basis Function (RBF) (Broomhead y Lowe, 1988; Moody y Darken, 1989)
            - Orthogonal Least Squares (OLS) (Chen, Cowan y Grant, 1991)
        - Cerebellar Articulation Controller (CMAC) (Albus, 1975)
        - Sólo clasificación:
            - Learning Vector Quantization (LVQ) (Kohonen, 1988)
            - Red Neuronal Probabilística (PNN) (Probabilistic Neural Network) (Specht, 1990)
        - Sólo regresión:
            - General Regression Neural Network (GRNN) (Specht, 1991)
    2. Con conexiones feedback
        - Bidirectional Associative Memory (BAM) (Kosko, 1992)
        - Máquina de Boltzman (Ackley, Hinton y Sejnowski, 1985)
        - Series temporales recurrentes
            - Backpropagation through time (Werbos, 1990)
            - Elman (Elman, 1990)
            - Finite Impulse Response (FIR) (Wan, 1990)
            - Jordan (Jordan, 1986)
            - Real-time recurrent network (Williams y Zipser, 1989)
            - Recurrent backpropagation (Pineda, 1989)
            - Time Delay NN (TDNN) (Lang, Waibel y Hinton, 1990)
    3. Competitivo

- ARTMAP (Carpenter, Grossberg y Reynolds, 1991)
- Fuzzy ARTMAP (Carpenter, Grossberg, Markuzon, Reynolds y Rosen, 1992)
- Gaussian ARTMAP (Williamson, 1995)
- Counterpropagation (Hecht-Nielsen 1987, 1988, 1990)
- Neocognitrón (Fukushima, Miyake e Ito, 1983; Fukushima, 1988)

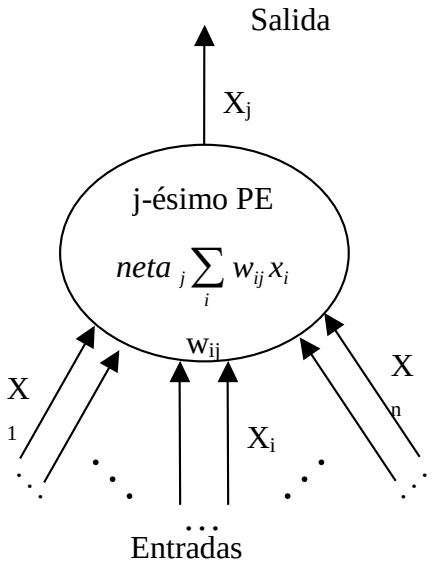
2. No supervisado
   1. Competitivo
      - Vector Quantization
        - Grossberg (Grossberg, 1976)
        - Kohonen (Kohonen, 1984)
        - Conscience (Desieno, 1988)
      - Mapa Auto-Organizado (Self-Organizing Map) (Kohonen, 1982; 1995)
      - Teoría de la Resonancia Adaptativa (Adaptive Resonance Theory, ART)
        - ART 1 (Carpenter y Grossberg, 1987a)
        - ART 2 (Carpenter y Grossberg, 1987b)
        - ART 2-A (Carpenter, Grossberg y Rosen, 1991a)
        - ART 3 (Carpenter y Grossberg, 1990)
        - Fuzzy ART (Carpenter, Grossberg y Rosen (1991b)
      - Differential Competitive Learning (DCL) (Kosko, 1992)
   2. Reducción de dimensionalidad
      - Regla de Oja (Oja, 1989)
      - Sanger (Sanger, 1989)
      - Differential hebbian (Kosko, 1992)
   3. Autoasociación
      - Autoasociador lineal (Anderson, Silverstein, Ritz y Jones, 1977)
      - Brain-State-in-a-Box (BSB) (Anderson, Silverstein, Ritz y Jones, 1977)
      - Red de Hopfield (1982)

Fuente: Montaño, (2005).

Unidades de procesamiento de la información.

El elemento básico de una red neuronal es un nodo. Es la unidad de procesamiento que actúa en paralelo con otros nodos de la red. Es similar a la neuronal del cerebro humano: acepta input y genera output. Su forma de actuar se muestra en la figura nº1; los nodos aceptan input de otros nodos. La primera tarea del nodo es procesar los datos de entrada creando un valor resumen que es la suma de todas las entradas multiplicadas por sus ponderaciones. Este valor resumen se procesa a continuación mediante una función de activación para generar una salida que se envía al siguiente nodo del sistema.

Figura 3.22. Unidad básica de una red neuronal.



Fuente: Elaboración propia.

Las cuatro funciones de activación más utilizadas son:

a) La función escalón.

Considerando que la activación debe llegar a un determinado nivel U (umbral de activación), esta función adopta la forma:

$$x_j = f_j(neta_j) = \begin{cases} 1 & \forall neta_j \geq U \\ 0 & \forall neta_j < U \end{cases}$$

Para evitar la discontinuidad se utilizan frecuentemente las tres funciones siguientes, (con mayor frecuencia las dos primeras), sin que por ello se agoten ni mucho menos todas las posibilidades.

b) Función identidad.

Es la función más sencilla.

c) Función sigmoide o x_j=$neta_j$ logística.

$$x_j = \frac{1}{1+e^{-aneta_j}} \qquad (3.98)$$

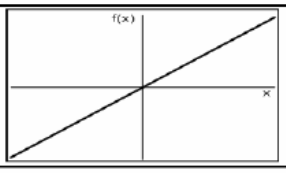
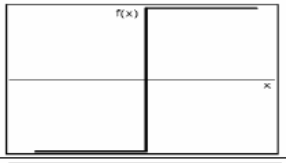
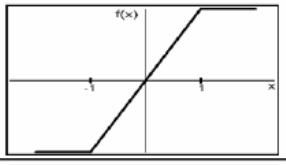
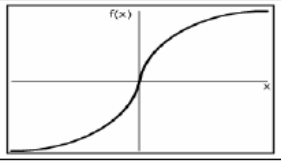
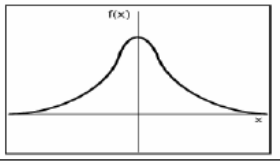
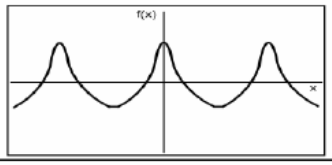
Esta función que toma valores entre 0 y 1, es conocida como la sigmoide asimétrica.

d) Tangente hiperbólica.

$$x_j = 1 - \frac{1}{e^{2neta_j} + 1} \tag{3.99}$$

Esta función varía entre –1 y 1. Es simétrica y en la literatura se la denomina frecuentemente sigmoide simétrica. Estas funciones ofrecen mayores ventajas, fundamentalmente una convergencia más rápida en el aprendizaje de la red.

Figura 3.23. Funciones activación más utilizadas en redes neuronales.

NOMBRE	FUNCIÓN	RANGO	GRÁFICA
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = sign(x)$ $y = H(x)$	$[-1, +1]$ $[0, +1]$	
Lineal a tramos	$y = \begin{cases} -1, \text{ si } x < -1 \\ x, \text{ si } +1 \le x \le -1 \\ +1, \text{ si } x > +1 \end{cases}$	$[-1, +1]$	
Sigmoidea o Logística	$y = \frac{1}{1+e^{-x}}$ $y = tgh(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = a \cdot e^{-bx^2}$	$[0, +1]$	
Sinusoidal	$y = a \cdot sen(wx + \varphi)$	$[-1, +1]$	

Fuente: Martín del Brío y Sanz (2001).

3.2.2.2. Red neuronal.

Se considera una red neuronal la ordenación secuencial de tres tipos básicos de nodos o capas: nodos de entrada, nodos de salida y nodos intermedios (capa oculta o escondida).

Los nodos de entrada se encargan de recibir los valores iniciales de los datos de cada caso para transmitirlos a la red. Los nodos de salida reciben entradas y calculan el valor de salida (no van a otro nodo). En casi todas las redes existe una tercera capa denominada oculta, Este conjunto de nodos utilizados por la red neuronal, junto con la función de activación posibilita a las redes neuronales representar fácilmente las relaciones no lineales, que son muy problemáticas para las técnicas multivariantes.

3.2.2.3. Propiedades de los sistemas neuronales.

Se puede decir que una red neuronal tiene tres ventajas que le hacen muy atractiva en el tratamiento de los datos: aprendizaje, robustez y paralelismo masivo:

- **Aprendizaje adaptativo a través de ejemplos.**

Una de las características más sobresalientes de las redes neuronales y que la aleja del resto de las técnicas multivariantes es su capacidad de aprender o de corregirse a sí misma basándose en los errores. Se puede considerar que el conocimiento se encuentra representado en los pesos de las conexiones entre las neuronas y en sus umbrales. El proceso de aprendizaje implica cierto número de cambios en estos valores de tal forma que se puede decir que “se aprende modificando los valores de los pesos y umbrales de las neuronas de la red”. Un criterio para clasificar las redes neuronales es respecto a las reglas de aprendizaje. Si el aprendizaje se basa en la existencia de un agente externo decimos que la red neuronal es supervisada, mientras que cuando no interviene el analista estamos frente a una red neuronal no supervisada. Un ejemplo de una red supervisada es el perceptrón multicapa que se describe a continuación, mientras que una red neuronal no supervisada en la red de Kohonen que también se describe en este módulo.

- **Tolerancia a fallos.**

Algunas de las capacidades de la red se pueden retener aún si ésta sufre daños. Las redes neuronales artificiales son muy robustas en el tratamiento de la información redundante e imprecisa.

- **Paralelismo masivo.**

Lo que significa que las operaciones se realizan en tiempo real. Los cómputos de la red pueden realizarse en paralelo para lo cual se pueden fabricar máquinas con hardware especial.

3.2.2.4. El perceptrón multicapa.

El método más utilizado en las aplicaciones prácticas de redes neuronales es el perceptrón multicapa, que fue popularizado por Rumelhart, (1986). Este modelo de red es conocido también como backpropagation error (propagación del error hacia atrás), también denominado método del gradiente decreciente.

La verdadera razón de su tremenda utilidad radica en su capacidad de organizar una representación interna del conocimiento en las capas ocultas de neuronas a fin de aprender la relación entre un conjunto de datos entradas y salidas. El perceptrón multicapa en un aproximador universal de funciones. La red backpropagation conteniendo al menos una capa oculta es capaz de aprender cualquier tipo de función o relación continua. Esta propiedad convierte a esta red en una herramienta de propósito general.

3.2.2.4.1. Etapa de funcionamiento.

El desarrollo de la red consta de una fase de entrenamiento y otra de funcionamiento.

En la primera etapa **c**uando se presenta un patrón de entrada X_p : $x_{p1},\ldots,x_{pi},\ldots x_{pN}$ se transmite a la red a través de los pesos w_{ji} desde la capa de entrada a la capa oculta. Las neuronas de esta capa transforman las señales a través de la función de activación proporcionando un valor de salida. Este valor se transmite a su vez a través de los pesos v_{kj} a la capa de salida donde aplicando de nuevo la función de activación obtenemos un valor de salida.

Vamos a suponer que la entrada total o neta de una neurona oculta j la expresamos cómo net_{pj}, entonces matemáticamente la podemos expresar de la siguiente manera:

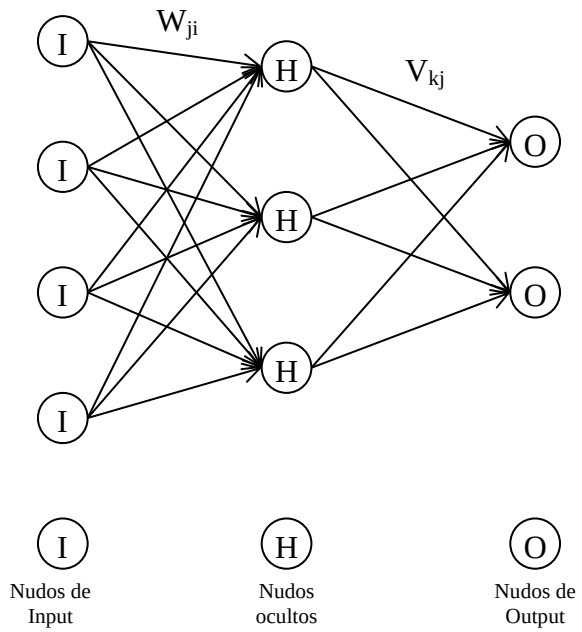
$$net_{pj} = \sum_{i=1}^{N} w_{ji} x_{pi} + \theta_j \qquad (3.100)$$

θ es el umbral de la neurona que se considera como un peso asociado a una neurona ficticia con valor de salida igual a 1.

El valor de salida de la neurona oculta j, b_{pj} lo obtenemos aplicando la función de activación f(.) sobre la entrada neta.

$$b_{pj} = f\left(net_{pj}\right) \tag{3.101}$$

Figura 3.24. Respuesta localizada de las neuronas ocultas en la RBF.



Fuente: Elaboración propia

La entrada neta que recibe una neurona de salida k la podemos expresar cómo:

$$net_{pk} = \sum_{j=1}^{L} v_{kj} b_{pj} + \theta_k \tag{3.102}$$

El valor de salida de la neurona k, y_{pk} es el siguiente:

$$y_{pk} = f\left(net_{pk}\right) \tag{3.103}$$

3.2.2.4.2. Etapa de aprendizaje.

En esta segunda etapa el objetivo que se persigue es hacer mínima la discrepancia o error entre la salida de la red y el valor real que presenta el usuario. La función que se pretende minimizar para cada patrón p viene dada por la siguiente expresión:

$$E_p = \frac{1}{2}\sum_{k=1}^{M}\left(d_{pk} - y_{pk}\right)^2 \tag{3.104}$$

Donde d_{pk} es la salida presentada por la red de la neurona k ante la presentación del patrón p. La medida general del error es la suma de todos los errores para todos los patrones.

$$E = \sum_{p=1}^{P} E_p \tag{3.105}$$

El objetivo fundamental del análisis con las RNA es minimizar el cuadrado de los errores entre el valor real y el de la variable salida dado por la ecuación 3.104. Esta función depende fundamentalmente de dos parámetros: el conjunto de variables de entrada, las variables explicativas de nuestro modelo, y el conjunto de pesos sinápticos.

Disponemos de un buen número de métodos o algoritmos que permiten estimar los parámetros del modelo de RNA minimizando E_p. Esta minimización de la función debe ser realizada por métodos de optimización no lineales. Existen diferentes enfoques a la hora de aplicar estos métodos y deben ser adaptados dependiendo de la dimensión y de la complejidad del problema. En los métodos no lineales la función E_p es una función continua y diferenciable así que podemos aplicar los métodos del gradiente. En estos métodos el proceso de búsqueda de la solución óptima usado puede ser descrito como:

$$\vartheta_{t+1} = \vartheta_t + \lambda_t \Delta_t \tag{3.105}$$

Donde ϑ_t es la solución que es viable, λ_t es el tamaño del peso y Δ_t = $M_t g_t$ es el vector de dirección. Para el vector de dirección se tiene que M_t es una matriz definida positiva y $g_t = g(\vartheta_t)$ es el vector gradiente.

En RNA la técnica más utilizada es la del gradiente decreciente, Rumelhart, (1986), denominada algoritmo backpropagation debido a la forma de modificación de los pesos. Este gradiente toma la dirección que determina el incremento más rápido en el error. Así que el error puede reducirse ajustando cada peso en la siguiente dirección:

$$-\sum_{p=1}^{P}\frac{\partial E_p}{\partial w_{ji}} \tag{3.106}$$

Entre los métodos gradientes se encuentra el método de Newton-Rapshon. Este algoritmo utiliza la inversa de la matriz hessiana como matriz M. En este procedimiento puede ocurrir que el valor del punto inicial ϑ_0 no esté próximo al punto óptimo lo que dificultaría el cálculo de la matriz hessiana, además de que puede suceder que en algunas aplicaciones esta matriz sea difícil de calcular, por lo que diferentes investigadores han desarrollado procedimientos de estimación que se conocen como métodos Quasi-Newton. Estos métodos usan una aproximación iterativa de la matriz hessiana:

$$M_{t+1} = M_t + N_t \tag{3.107}$$

Donde N_t es una matriz definida positiva, lo que garantiza que en cada paso del proceso iterativo la aproximación sea también definida positiva, al ser suma de dos matrices definidas positivas. En esta implementación tenemos que seleccionar el punto inicial ϑ_0 y una matriz M_0 que deberá ser definida positiva.

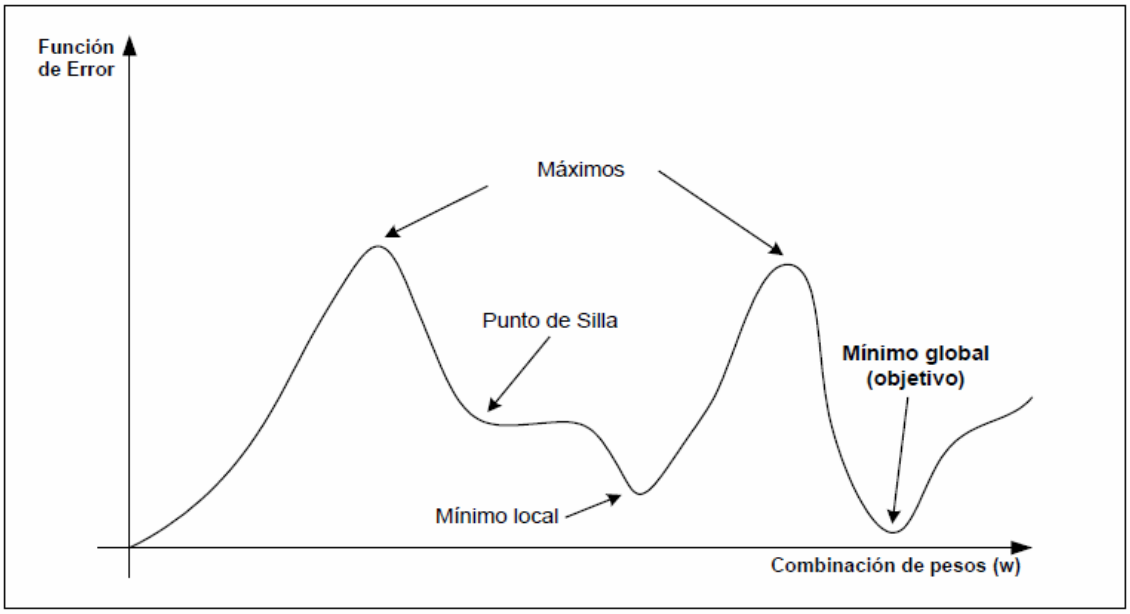
Uno de los algoritmos empleados para la estimación de la estructura de la RNA es el BFGS (Broyden-Fletcher-Goldfarb-Shanno) descrito en Fletcher (1987) que utiliza la siguiente expresión iterativa:

$$M_{t+1} = M_t\frac{\delta_t\delta_t^{'}}{\delta^{'}\nu_t} + \frac{M_t\nu_t\nu_t^{'}M_t}{\nu_t^{'}M_t\delta_t} - \nu_t^{'}M_t\nu_t\left(\frac{\delta_t}{\delta_t^{'}\nu_t} - \frac{M_t\nu_t}{\nu_t^{'}M_t\nu_t}\right)\left(\frac{\delta_t}{\delta_t^{'}\nu_t} - \frac{M_t\nu_t}{\nu_t^{'}M_t\nu_t}\right) \tag{3.108}$$

donde $\delta_t = \vartheta_{t+1} - \vartheta_t$ y $\nu_t = g(\vartheta_{t+1}) - g(\vartheta_t)$

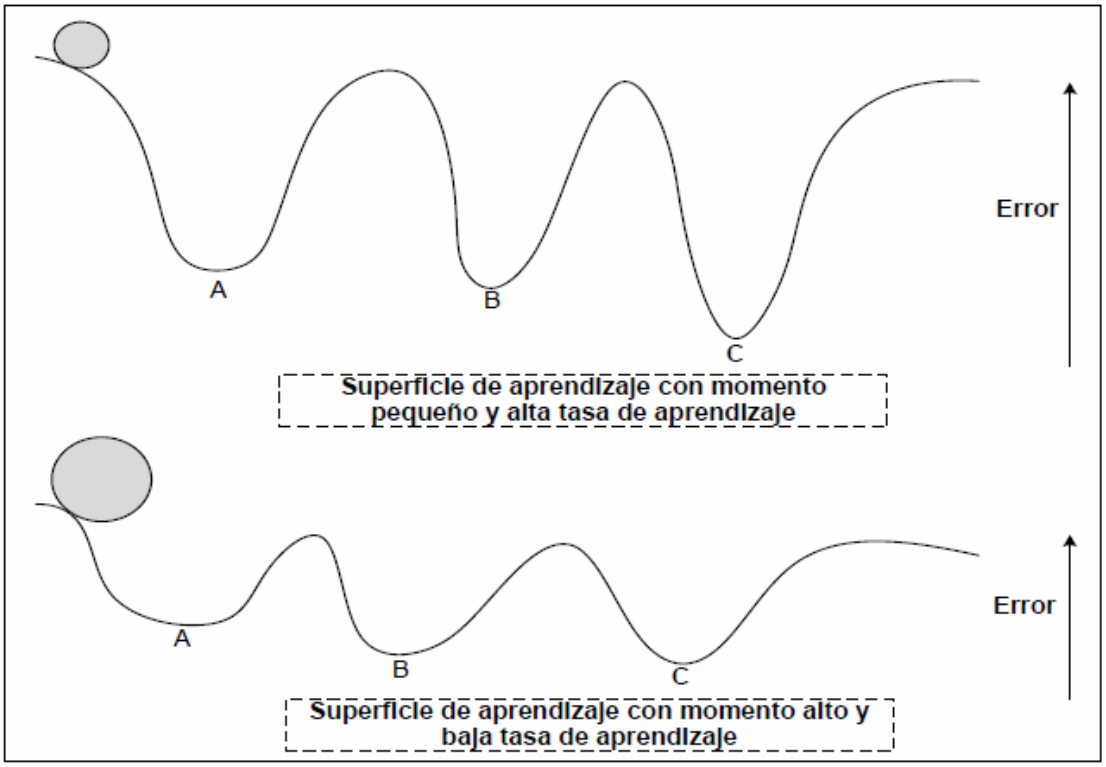
Un problema que se puede presentar al utilizar este método del gradiente conjugado es que caiga en mínimos locales. Sin embargo, como ya han señalado diversos autores, se da en contadas ocasiones cuando se trabaja con datos reales.

Figura 3.25. Complejidad en la búsqueda del mínimo global.



Fuente: Urquijo y Mendivil (2011).

Figura 3.26. Dinámica en la búsqueda del mínimo global.



Fuente: Fuente: Urquijo y Mendivil (2011).

Según Nisbet *et al.* (2009), en las redes neuronales artificiales configuradas manualmente, una tasa de aprendizaje del orden del 0,9 obtiene los mejores resultados, junto a momentos que oscilen entre 0,1 y 0,3.

Esta información que nos facilita Nisbet *et al.* (2009) no es la que mejor encaja con la red neuronal estimada en nuestro caso, donde se ajustan los mejores resultados para los datos de entrenamiento en una tasa de aprendizaje del 0,9 y un momento también alto, del 0,8 o del 0,6.

El hecho de que se pueda definir un momento con valores entre 0 y 1, tal y como señala Zhang *et al.* (1997), hace prácticamente imposible realizar una búsqueda exhaustiva para encontrar la mejor combinación, junto con la tasa de aprendizaje, implicando el diseño de RNA supervisadas una tarea que implica necesariamente cierto ensayo y error.

Otro parámetro que es necesario controlar adecuadamente es la tasa de disminución de aprendizaje, puesto que comúnmente la tasa de aprendizaje va decayendo a medida que se realiza el entrenamiento. Esto tiene el efecto importante de aplanar la superficie de búsqueda y de esta manera permitir encontrar los mínimos globales de una forma más fácil.

En la práctica la forma de modificar los pesos de forma iterativa se realiza aplicando la regla de la cadena a la expresión del gradiente y añadir una tasa de aprendizaje que se denomina η. Para una neurona de salida obtenemos la siguiente expresión:

$$\Delta v_{kj}(n+1) = \eta \sum_{p=1}^{P} \delta_{pk} b_{pj} \tag{3.109}$$

Donde δ_{pk} tiene la siguiente expresión:

y **n** nos indica la iteración.

Para el peso de una neurona oculta:

$$\Delta w_{ji}(n+1) = \eta \sum_{p=1}^{P} \delta_{pj} x_{pi} \tag{3.110}$$

donde $\delta_{p,j}$ es igual a:

$$\delta_{pj} = f'(net_{pj})\sum_{k=1}^{M}\delta_{pk}v_{kj}$$

Para acelerar el proceso de convergencia de los pesos se sugiere añadir a la expresión un factor momento,α, el cual tiene en cuenta la dirección del incremento tomado en la iteración anterior. Por ejemplo, para el peso de una *neurona de* salida, la expresión es la siguiente:

$$\Delta w_{ji}(n+1) = \alpha(n)\left(x_{pi} - w_{ji}(n)\right) \tag{3.111}$$

Y para el peso de una neurona oculta:

$$\Delta v_{kj}(n+1) = \eta\left(\sum_{p=1}^{P}\delta_{pk}b_{pj}\right) + \alpha\Delta v_{kj}(n) \tag{3.112}$$

3.2.2.4.3. Metodología de aplicación de un perceptrón multicapa.

A continuación se ofrecen una serie de pasos para la ejecución de este modelo de red, que es el modelo más extendido, en las aplicaciones prácticas:

- **Elección adecuada de las variables.**

Para obtener los mejores resultados se deberán de escoger cuidadosamente las variables. La introducción de variables irrelevantes en el modelo o que estén muy correlacionadas puede generar un sobreajuste, lo que puede provocar una disminución sensible de su capacidad de generalización.

- **Creación de los conjuntos de aprendizaje, validación y test.**

La muestra de los datos se divide generalmente en tres conjuntos: entrenamiento, validación y test. Para evitar el sobreajuste es aconsejable utilizar un segundo grupo que nos permita controlar el proceso de aprendizaje de la red neuronal. El conjunto de datos que forman el grupo de test es aconsejable porque debemos de disponer de un grupo independiente del proceso de entrenamiento de la red para probar su eficacia y que nos proporcionará una estimación insesgada del error de generalización.

- **Entrenamiento de la red neuronal.**

Para que la red efectúe el entrenamiento hay que proporcionar una serie de elementos determinados mediante ensayo y error. El grupo de validación nos permitirá optimizar este conjunto de parámetros: elección de los pesos iniciales, arquitectura de la red, tasa de aprendizaje y factor momento y las funciones de activación de las neuronas ocultas y de salida.

Para que la red empiece su etapa de entrenamiento se han de designar los pesos de las conexiones y los valores umbrales. Una forma sencilla y muy utilizada es asignar pesos pequeños elegidos de forma aleatoria en un rango de valores entre -0,5 y 0,5.

En cuanto a la arquitectura de la red es sabido que para la resolución de la mayor parte de los problemas es suficiente con utilizar una sola capa oculta. El número de neuronas de la capa de entrada está determinado por las variables predictoras. Las neuronas de la capa de salida están en función de si el problema que queremos resolver es de clasificación o estimación. Las neuronas de la capa oculta, que nos determinan la capacidad de aprendizaje de la red, se deben de elegir de forma que la red rinda de forma adecuada con el menor número de neuronas en esta capa. No hay teoría que nos lo indique pero disponemos del conjunto de validación para probar diferentes arquitecturas.

La tasa de aprendizaje controla el tamaño del cambio en el proceso de entrenamiento de la red. Si el ritmo de aprendizaje es muy pequeño la velocidad de convergencia disminuye y se puede caer en mínimos locales. Un valor muy alto puede ocasionar inestabilidades y evitar la convergencia. Algunos autores recomiendan probar con valores comprendidos entre 0,005 y 0,5.

El factor momento (**α)** permite filtrar las oscilaciones en la superficie del error que provoca la tasa de aprendizaje. El valor que suele tomar es próximo a 1.

El algoritmo backpropagation exige que las funciones de activación sean derivables para poder obtener el valor del error de las neuronas de la capa oculta y de salida. Si queremos aprovechar la capacidad que tienen las redes neuronales artificiales de aprender relaciones complejas y no lineales es imprescindible la utilización de funciones no lineales, al menos, en la capa oculta. En general se utilizan la función logística o la tangente hiperbólica.

3.2.2.5. Evaluación del rendimiento del modelo.

El último paso y el más importante es evaluar la capacidad de generalización que ha conseguido la red. Para este fin se ha reservado el tercer conjunto de datos de test. Cuando el problema es de estimación normalmente se utiliza la media cuadrática del error. Si el problema es de clasificación podemos construir una tabla de confusión y sobre los datos calcular diferentes índices de asociación. Si estamos interesados en discriminar entre dos categorías se puede hacer uso de los índices de sensibilidad, especificidad y eficacia y del análisis de curvas ROC (Receiver operating characteristic) (Palmer, Montaño y Calafat, 2000).

La expresión de la media cuadrática del error se calcula a través de la siguiente fórmula:

$$MC_{Error} = \frac{\sum_{p=1}^{P}\sum_{k=1}^{M}(d_{pk} - y_{pk})^2}{P \cdot M} \qquad (3.113)$$

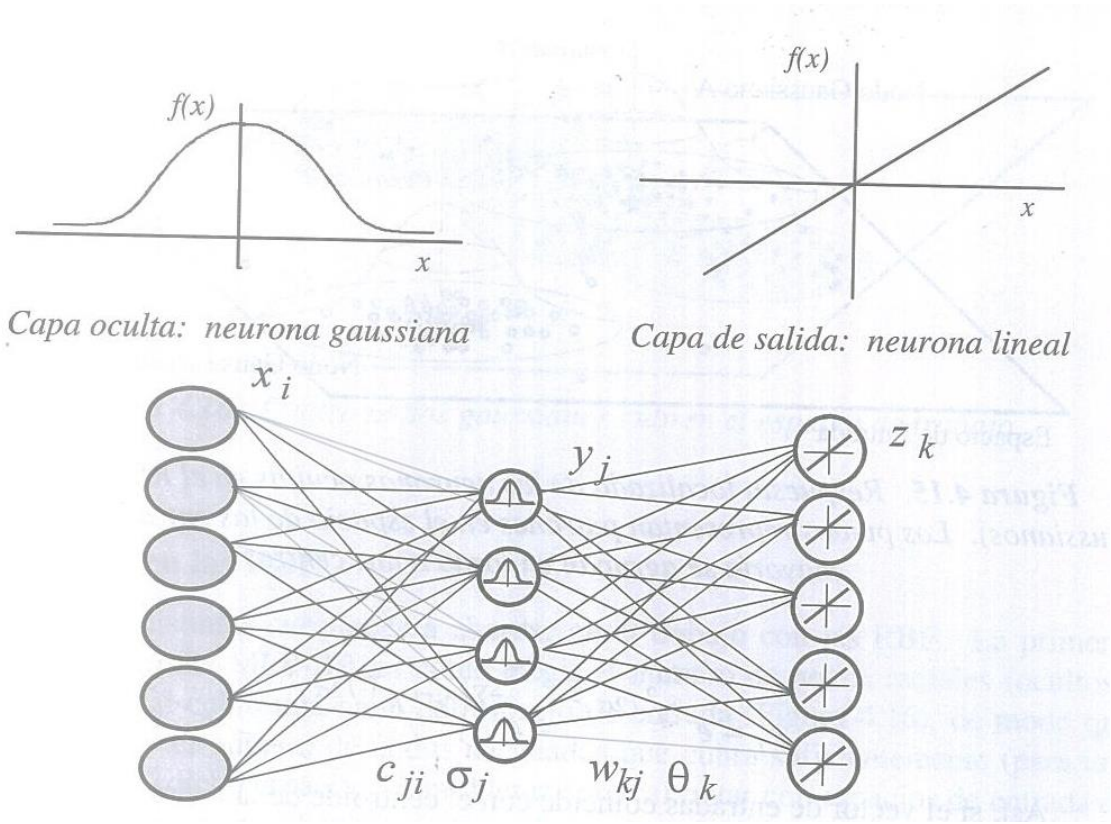
3.2.2.6. Funciones de Base Radial.

Las funciones de base radial (Radial Basis Functions o RBS) [Moody 89, Pogio 90] cuentan cada vez con más aplicaciones prácticas debido, sobre todo, a su simplicidad. Una función de base radial es una función cuya característica principal es que su respuesta disminuye (o aumenta) monótonamente con la distancia a un punto fijo llamado centro (o centroide).

Este modelo se puede considerar como un modelo híbrido de red neuronal al incorporar aprendizaje supervisado y no supervisado.

Al igual que en el perceptrón multicapa pueden modelar fácilmente sistemas no lineales en menor tiempo que el modelo clásico. Esta arquitectura al igual que del perceptrón multicapa constituyen aproximadores universales de funciones.

Figura 3.27. Arquitectura de una red neuronal RBF.



Fuente: Martín del Brío y Sanz (2001).

Este modelo de red cuenta con tres capas de neuronas: de entrada, oculta y de salida. Las neuronas de entrada envían información a la capa oculta como en el perceptrón multicapa. La diferencia fundamental se halla en como procesan la información de las neuronas de la capa oculta. En esta capa se opera en base a la distancia que separa el vector de entrada respecto del vector sináptico que cada neurona de la capa oculta almacena. A estos vectores de la capa oculta se les llama centroides. A esta cantidad se le aplica una función radial con forma guassiana. Debido a esto, en esta capa las neuronas tienen una respuesta localizada respondiendo con una intensidad apreciable cuando el vector de entradas y el centroide de la neurona pertenecen a una zona próxima en el espacio de las entradas.

De forma matemática el modelo se expresa así:

$$r_j^2 = \left\| x - c_j \right\|^2 = \sum_i \left(x_i - c_{ji} \right)^2 \qquad (3.114)$$

Donde x_i son las entradas de la red y c_{ji} representa al centroide

La salida de la neurona y_j se calcula a partir de la función de activación denominada función radial $\phi(r)$ que suele ser la función gaussiana.

$$\phi(r) = e^{-r^2/\sigma^2} \tag{3.115}$$

En esta función cuanto mayor sea σ mayor será la región que la neurona domina en torno al centroide.

La función de salida y_j es la siguiente:

$$y_j = e^{-r_j^2/2\sigma_j^2} = e^{-\sum_i (x_i - c_{ji})^2/\sigma_j^2} \tag{3.116}$$

Cuando el vector de entrada se aproxima al centroide de una neurona, ésta se activa, lo que significa que reconoce al patrón de entrada, sin embargo si el patrón de entrada es muy diferente al centroide la respuesta tiende a cero.

Las salidas de las neuronas ocultas son las entradas de las neuronas de la capa de salida z_k cuya expresión es la siguiente:

$$z_k = \sum_j w_{kj} y_j + \phi_k = \sum_j w_{kj} \phi(r_j) + \phi_k \tag{3.117}$$

Donde w_{kj} = es el peso que conecta la neurona oculta j con la salida k, y ϕ_k un parámetro adicional de la neurona k que es el umbral.

En el aprendizaje de las redes de base radial hay que tomar en cuenta, en primer lugar, el número de neuronas en la capa oculta también llamados nodos ocultos radiales. Al margen del método de prueba y error se pueden emplear algunos procedimientos más o menos automáticos como el modelo auto organizado jerárquico (Hierarchically Self organizing Learning Algorithm) de Lee y Kil (1991] para su determinación. Una vez determinado el número de nodos radiales suele emplearse un aprendizaje por etapas donde primero se realiza el entrenamiento de las neuronas ocultas y después se procede al entrenamiento de las neuronas de salida.

Existen tres enfoques para el aprendizaje de las redes:

1) Selección fija de centros, pesos y varianzas.
2) Aprendizaje auto-organizado de los centros.

3) Aprendizaje supervisado.

Los dos primeros suponen funciones gaussianas normalizadas:

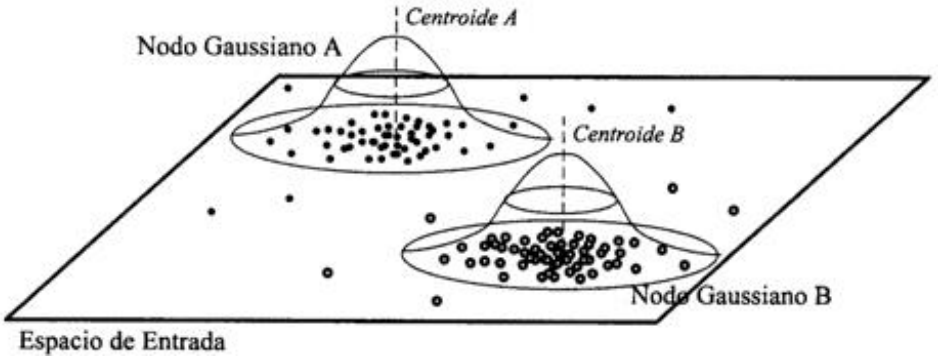
$$h_c(x) = \exp\left(-\frac{1}{\sigma^2}\|x - c\|^2\right) \tag{3.118}$$

donde **c** es el centro, **σ** es la varianza (lo que regula la anchura).

El tercer caso, para más versatilidad, considera el caso general de las gaussianas.

Las funciones de base radial se denominan así precisamente por la simetría radial de estas funciones, lo que aquí significa es que el nodo da una salida idéntica para aquellos patrones que distan lo mismo del centroide. El parámetro de normalización o factor de escala σ nos mide la anchura de la función gaussiana y equivale al radio de influencia de la neurona en el espacio de las entradas. Cuanto mayor es el valor de σ la región que la neurona domina en torno al centroide es más amplia como puede observarse en el siguiente gráfico. Los puntos en el gráfico representan patrones en el espacio de las entradas que, como se puede observan, se agrupan en su mayoría en torno a dos centros.

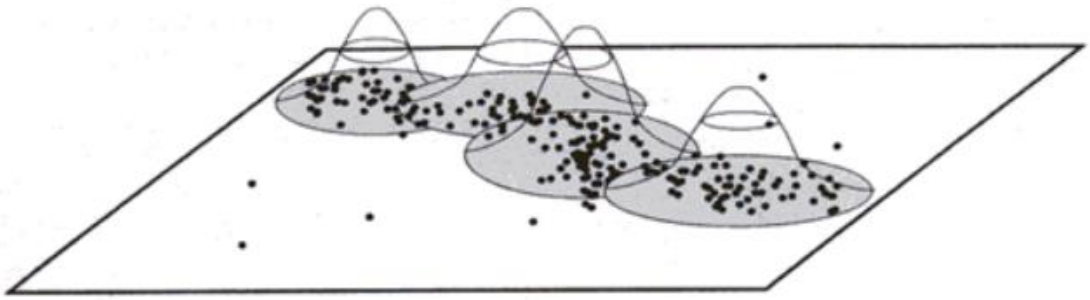
Figura 3.28. Respuesta localizada de las neuronas ocultas en la RBF.



Fuente: Martín del Brío y Sanz (2001).

En las redes de base radial cada nodo se ocupa de una zona del espacio y el conjunto de ellos debe de cubrir la zona de interés. Este proceso de cubrimiento tiene que llevarse de la manera más suave posible controlando el número de nodos de la capa oculta y con la amplitud de σ. En el grafico siguiente se observa cómo cuatro nodos gaussianos cubren el espacio de trabajo.

Figura 3.29. Nodos Gausianos recubriendo el espacio de trabajo.



Fuente: Martín del Brío y Sanz (2001).

A la familia de las redes RBF pertenecen otras arquitecturas de redes, entre otras, la Red Neuronal Probabilística (PNN, *Probabilistic Neural Network*) (Specht, 1990), la *General Regression Neural Network* (GRNN) (Specht, 1991) y la *Counterpropagation* (Hecht-Nielsen, 1987, 1988, 1990).

3.2.2.7. Comparación entre las Funciones de Base Radial y el Perceptrón Multicapa.

1. Una RBF tiene sólo una capa escondida mientras que MLP puede tener varias.
2. La capa oculta de una RBF es no lineal mientras que la capa de salida es lineal. Mientras que en un MLP como clasificador las capas escondida y de salida usualmente son no lineales; y cuando MLP se usa para síntesis funcional la capa de salida se elige como lineal.
3. El argumento de activación de la función de activación de una neurona escondida de una RBF es una distancia euclidiana entre el vector entrante y el centro de esa unidad. En MLP el argumento de la función de activación de cada neurona escondida es un producto interno del vector de entrada con el de los pesos.
4. MLPs construyen aproximaciones globales, pero las RBF construyen aproximaciones locales.

3.2.2.8. Análisis de sensibilidad e interpretación de los pesos de la red.

En las investigaciones de las redes neuronales se ha prestado mucha atención al desarrollo de las arquitecturas y desarrollo de algoritmos y existen pocas investigaciones relacionadas con los procedimientos que nos ayuden a comprender la verdadera naturaleza de las representaciones internas generadas por la red, Montaño y Palmer (2003).

Una de las principales ventajas que se han señalado de la Redes Neuronales es que se han visto como una suerte de "cajas negra" y este hecho tal y como afirma Shachmurove (2002) señala una indeterminación observacional que surge de la naturaleza autopoética o de auto organización de las Redes neuronales artificiales, lo que hace muy complejo entender como las relaciones en las capas ocultas son estimadas.

Tanto en Montaño y Palmer (2003) como en Nisbet *et al.* (2009) se observa como estudios recientes están abriendo en gran medida esa caja negra, como algunos autores la han denominado, a través del análisis de sensibilidad

Para abrir esta caja negra de las redes neuronales existen dos metodologías que nos ayudan a interpretar qué ha aprendido el perceptrón multicapa a partir del valor de los pesos y de los valores de activación de las neuronas. Con el conocimiento de estos valores lo que se pretende es conocer la importancia de cada variable independiente sobre la variable de salida, que en nuestro caso es la clase, que toma dos valores, SÍ (Se le concede el crédito) y NO (se deniega la petición de crédito.

Las dos metodologías que se comentan brevemente a continuación están basadas, la primera, en un análisis sobre los pesos de las neuronas y la segunda recibe el nombre de análisis de sensibilidad donde se estudia el efecto que produce en una variable de salida debido al cambio que se origina en una variable de entrada o bien en el error cometido.

3.2.2.8.1. Análisis basado en la magnitud de los pesos de la red.

Estos procedimientos se basan sólo en los valores de la matriz estática de pesos y cuyo propósito es determinar la influencia relativa que tiene cada variable de entrada sobre la de salida.

No es correcto suponer tal como nos lo recuerda Master (1993) que las entradas con pesos con valor absoluto mayor tengan que ser las más importantes o las de peso menor sean las menos significativas. En la literatura sobre redes neuronales existen diferentes propuestas de ecuaciones basadas en la magnitud de los pesos si bien todas ellas se caracterizan por calcular el producto de los pesos w_{ij} y v_{jk} para cada una de las neuronas ocultas: Yoon *et al.* (1989); Baba *et al.* (1990); Garson, (1991ª); Garson, (1991b); Yoon *et al.* (1993); Milne, (1995); Gedeon, (1997); Tsaih, (1999). Una

de las expresiones más utilizadas es la propuesta por Garson, (1991a, 1991b) y por Modai *et al.* (1995):

$$Q_{ik} = \frac{\sum_{j=1}^{L}\left(\frac{w_{ij}}{\sum_{r=1}^{N} w_{rj}} v_{jk}\right)}{\sum_{i=1}^{N}\left(\sum_{j=1}^{L}\left(\frac{w_{ij}}{\sum_{r=1}^{N} w_{rj}} v_{jk}\right)\right)} \qquad (3.119)$$

donde $\sum_{r=1}^{N} w_{rj}$ es la suma de los pesos de conexión entre las *i* neuronas de entrada y la neurona oculta *j*.

Este índice Q_{ik} es el porcentaje de influencias de la variable de entrada *i* sobre la salida *k* y la suma de este índice para todas las variables debe de vale el 100%

3.2.2.8.2. Análisis de sensibilidad.

El análisis de sensibilidad según Yeung *et al*. (2010) se refiere a observar cómo la salida (el output) es afectada por las entradas o inputs y/o en los pesos sinápticos tal y como hemos visto anteriormente. Sobre este importante tema podemos encontrar tres desarrollos que se comentan brevemente a continuación.

3.2.2.8.2.1. Análisis de sensibilidad basado en el error.

La función de error que se utiliza tiene la siguiente fórmula:

$$RMC_{error} = \sqrt{\frac{\sum_{p=1}^{P}\sum_{k=1}^{M}(d_{pk} - y_{pk})^2}{P \cdot M}} \qquad (3.120)$$

donde d_{pk} es la salida deseada para el patrón *p* en la neurona de salida *k*.

El método consiste, siguiendo a Frost y Karry (1999) en aplicando pequeños incrementos a las variables de entrada mientras se mantienen los valores originales del resto de variables de entrada de la red. Una vez realizado este paso se procede a

entrenar la red calculando el valor RMC error. Si seguimos este procedimiento para todas las variables podemos determinar una ordenación de las mismas: la variable de entrada que consiga el mayor RMC será considerada como la variable más influyente sobre la variable de salida.

Otras variantes de este procedimiento las podemos encontrar en Masters (1993).

3.2.2.8.2.2. Análisis de sensibilidad basado en la salida.

Citando al estudio de Engelbrecht *et al.* (1999), el análisis de sensibilidad puede ser empleado para una gran variedad de tareas como son: optimización, robustez y análisis de la relación inputs/outputs, inferencia y causalidad, aprendizaje selectivo o reducción del conjunto de inputs, entre otros.

Esta forma de estudiar la sensibilidad se basa en los efectos que se producen en las salidas debido a los cambios de las variables de entrada. Sobre la red entrenada se fija el valor de todas las entradas a su valor medio. Variando el valor de una de las variables o introduciendo ruido podemos registrar los cambios producidos en la red.

Esta forma de proceder ha sido aplicado a muchos campos del conocimiento en tareas de predicción, algunos de ellos son los siguientes: predicción del comportamiento de la bolsa (Bilge *et al.* (1993), predicción de las auto expectativas en niños, Reid *et al.* (1994) y Kashani *et al.* (1996), en el análisis de supervivencia por De Laurentiis y Ravdin, (1994), en estudios de tratamiento psiquiátrico, Modai *et al.* (1995) o en la predicción farmacológica, Opara *et al.* (1999)

Otra forma de proceder es a través de la matriz Jacobiana (S_{IK}) dado que los elementos de la misma nos proporcionan de forma analítica una medida se sensibilidad de las salidas por cambios efectuados en las variables de entrada.

Si tomamos en cuenta que una red neuronal artificial opera en un mapa no lineal y diferenciable Γ: $R^I \rightarrow R^K$ desde un vector de entradas $X = (X_1, X_2 \ldots X_i)$ hacia un vector de salidas $Y = (Y_1, Y_2, \ldots Y_K)$. La matriz Jacobiana toma la siguiente expresión:

$$S_{IK} = \begin{pmatrix} \frac{\partial Y_1}{\partial X_1} & \frac{\partial Y_1}{\partial X_2} & \cdots & \frac{\partial Y_1}{\partial X_I} \\ \frac{\partial Y_2}{\delta X_1} & \frac{\partial Y_2}{\partial X_2} & \cdots & \frac{\partial Y_2}{\partial X_I} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial Y_K}{\partial X_1} & \frac{\partial Y_K}{\partial X_2} & \cdots & \frac{\partial Y_K}{\partial X_1} \end{pmatrix}$$

En esta matriz Jacobiana de orden I x K cada fila constituye una entrada de la red y las columnas representan las salidas de la red:

$$S_{IK} = \frac{\partial y_K}{\partial x_I} = f'(net_k) \sum_{j=1}^{L} v_{JK} f'(net_J) w_{IJ} \tag{3.121}$$

Aplicando esta fórmula a una red con función de activación sigmoidal logística la sensibilidad de la salida K con respecto a la entrada tomaría la siguiente expresión:

$$S_{IK} = y_K (1 - y_K) \sum_{J=1}^{L} v_{JK} b_J (1 - b_J) w_{IJ} \tag{3.122}$$

Los valores de la matriz Jacobiana dependen de la información aprendida por la red que se encuentra distribuida en las diferentes conexiones y de las funciones de transferencia entre las diferentes capas.

Para un patrón X_p se `puede calcular la sensibilidad S_{ik} (p) partir de la esperanza matemática y de su desviación estándar que se calculan a través de las siguientes expresiones:

$$E(S_{IK}(p)) = \frac{\sum_{p=1}^{P} S_{IK}(p)}{P} \tag{3.123}$$

$$SD(S_{iIK}(p)) = \sqrt{\frac{\sum_{p=1}^{P} \left(S_{iIK}(p) - E(S_{IK}(p))\right)^2}{P-1}} \tag{3.124}$$

Se pueden citar numerosos autores que han utilizado este análisis de sensibilidad. Algunos de los más representativos son los siguientes: Hwang *et al*. (1991); Hashem, (1992); Fu y Chen, (1993); Zurada *et al*. (1994); Bishop, (1995); Rzempoluck, (1998) y

ha sido aplicado en campos tan variados como el reconocimiento de imágenes (Takenaga *et al.* (1991), la ingeniería (Guo y Uhrig, 1992; Bahbah y Girgis, 1999), la meteorología, *Castellanos et al.* (1994) y la medicina, Harrison *et al.* (1991); Engelbrecht *et al.* (1995); Rambhia et al (1999).

En Gedeon (1997) se puede encontrar una comparación de los métodos basados en la magnitud de los pesos y el análisis de sensibilidad a través del cálculo de la matriz Jacobiana. El estudio refleja que los análisis basados en propiedades dinámicas son más fiables que los análisis estáticos.

3.2.2.8.2.3. Análisis de sensibilidad numérico.

Este análisis supera algunas de las limitaciones que surgen de las restricciones encontradas en los métodos descritos en los apartados anteriores. En este sentido, el análisis basado en la magnitud de los pesos no ha demostrado ser sensible a la hora de ordenar las variables de entrada en función de su importancia sobre la salida (Garson, 1991a; Sarle, 2000). También hay que tener en cuenta que una gran variedad de estudios, sobre todo los realizados en las ciencias sociales utilizan variables discretas y/o nominales y no es muy correcto utilizar la técnica de los incrementos a las variables de entrada (Hunter, Kennedy, Henry y Ferguson, 2000). También hay que añadir que el método de la matriz Jacobiana se basa en que las variables de entrada son continuas Sarle, (2000).

Todas estas condiciones restrictivas están superadas en el nuevo método llamado sensibilidad numérica (NSA, *Numeric Sensitivity Analysis*), descrito a continuación brevemente según las notaciones y fórmulas empleadas en Montaño y Palmer (2003).

Este método está basado en el cálculo de las pendientes que se forman entre las entradas y salidas sin necesidad de realizar ningún supuesto sobre la naturaleza de las variables y siempre respetando la estructura primitiva de los datos.

El índice *NSA* está basado en el cálculo numérico de la pendiente formada entre cada par de grupos consecutivos, g_r y g_{r+1}, de x_i sobre y_k mediante la siguiente expresión:

$$NSA_{ik}(g_r) \equiv \frac{\bar{y}_k(g_{r+1}) - \bar{y}_k(g_r)}{\bar{x}_i(g_{r+1}) - \bar{x}_i(g_r)} \tag{3.125}$$

donde

$\bar{x}_i(g_r)$ y $\bar{x}_i(g_{r+1})$ son las medias de la variable x_i correspondientes a los grupos g_r y g_{r+1}, respectivamente. $\bar{y}_k(g_r)$ e $\bar{y}_k(g_{r+1})$ son las medias de la variable y_k correspondientes a los grupos g_r y g_{r+1}, respectivamente.

El valor el valor de la esperanza matemática del índice *NSA* o pendiente entre la variable de entrada *i* y la variable de salida *k* mediante:

$$E(NSA_{ik}(g_r)) = \sum_{r=1}^{G-1} NSA_{ik}(g_r) \cdot f(NSA_{ik}(g_r)) = \frac{\bar{y}_k(g_G) - \bar{y}_k(g_1)}{\bar{x}_i(g_G) - \bar{x}_i(g_1)} \qquad (3.126)$$

donde

$f(NSA_{ik}(g_r)) \equiv \frac{\bar{x}_i(g_{r+1}) - \bar{x}_i(g_r)}{\bar{x}_i(g_G) - \bar{x}_i(g_1)}$ representa la función de probabilidad del índice *NSA*

$\bar{x}_i(g_G)$ y $\bar{x}_i(g_1)$ son los valores promedio de la variable x_i para el último grupo g_G y el primer grupo g_1, respectivamente

$\bar{y}_k(g_G)$ e $\bar{y}_k(g_1)$ son los valores promedio de la variable y_k para el grupo g_G y el grupo g_1, respectivamente.

El valor de la esperanza matemática del índice *NSA* representa el efecto promedio que tiene un incremento de x_i sobre y_i. Al igual que en el caso de la matriz Jacobiana, cuanto mayor sea el valor absoluto de $E(NSA_{ik}(g_r))$, más importante es x_i en relación a y_k. El signo de $E(NSA_{ik}(g_r))$ indica si el cambio observado en y_k va en la misma dirección o no que el cambio provocado en x_i.

El cálculo de la desviación estándar del índice *NSA*, cuando las variables implicadas son de naturaleza continua, se puede realizar mediante:

$$SD(NSA_{ik}(g_r)) = \sqrt{E(NSA_{ik}^2(g_r)) - \left(E(NSA_{ik}(g_r))\right)^2} \qquad (3.127)$$

El valor de la desviación estándar se debe interpretar como el grado de oscilaciones que ha sufrido la pendiente que se establece entre x_i e y_k, de manera que a mayor valor de la desviación estándar, mayor comportamiento caótico o aleatorio tiene la función entre las dos variables implicadas.

Las conclusiones de los autores: Montaño y Palmer (2003) después de experimentar con cuatro matrices de datos y con el programa informático creado por ellos "Sensitivity Neural Network 1.0" son, en primer lugar y respecto al grado de generalidad, es que el método NSA describe mejor el efecto o la importancia de las variables, pero si éstas son cuantitativas los resultados son muy similares a los proporcionados por el método de la matriz Jacobiana. Si las variables implicadas son discretas los valores que proporciona son muy similares a los que aporta el índice de asociación Phi en el caso de variables binarias y al índice de asociación V en el caso de variables politómicas.

En el Método NSA las interpretaciones de los efectos de las variables son más sencillas porque el índice que proporciona está acotados en el intervalo [1-1] También proporciona un método gráfico que representa la función aprendida por la red entre una variable de entrada y la de salida lo que permite complementar el análisis numérico.

3.2.2.9. Redes neuronales y modelos estadísticos clásicos.

Es muy interesante observa cómo se relaciones los modelos de redes neuronales con los métodos estadísticos clásicos dado que esta comparación ofrecerá una visión más completa de la importancia de las redes neuronales como excelente clasificadores al mismo tiempo que nos motivará a utilizarlas en próximos estudios.

Una posible idea falsa acerca de las redes neuronales, que provoca diferencia entre ambos modelos, es que parece que la terminología utilizada no está en consonancia con la que se utiliza en la estadística clásica debido, fundamentalmente, a que las redes neuronales proceden del campo de la Inteligencia Artificial con aportaciones de una gran variedad de disciplinas. Sarle, (1994) y Vicino, (1998) en los estudios llevados a cabo desmienten estas diferencias y establecen muchas de las semejanzas que hay entre los modelos estadísticos clásicos y las diversas arquitecturas de las redes neuronales.

Tabla 3.9. Equivalencia en la terminología estadística y de redes neuronales.

Terminología estadística	Terminología de redes neuronales
Observación	Patrón
Muestra	Datos de entrenamiento
Muestra de validación	Datos de validación, test
Variables explicativas	Variables de entrada
Variable de respuesta	Variable de salida
Modelo	Arquitectura
Residual	Error
Error aleatorio	Ruido
Estimación	Entrenamiento, aprendizaje
Interpolación	Generalización
Interacción	Conexión funcional
Coeficientes	Pesos de conexión
Constante	Peso umbral
Regresión y análisis discriminante	Aprendizaje supervisado o heteroasociación
Reducción de datos	Aprendizaje no supervisado o autoasociación
Análisis de cluster	Aprendizaje competitivo

Fuente: Sarle (1994) y Vicino (1998).

Tabla 3.10. Equivalencia entre modelos estadísticos y modelos de red neuronal.

Modelo estadístico	**Modelo de red neuronal**
Regresión lineal múltiple	Perceptrón simple con función lineal
Regresión logística	Perceptrón simple con función logística
Función discriminante lineal	Perceptrón simple con función umbral
Regresión no lineal múltiple	Perceptrón multicapa con función lineal en la salida
Función discriminante no lineal	Perceptrón multicapa con función logística en la salida
Análisis de componentes principales	Regla de Oja Perceptrón multicapa autoasociativo
Análisis de clusters	Mapas autoorganizados de Kohonen
K vecinos más cercanos	Learning Vector Quantization (LVQ)
Regresión kernel	Funciones de Base Radial (RBF)

Fuente: Sarle, (1994).

A la vista de estos dos cuados se observa que la mayoría de redes neuronales aplicadas al análisis de datos son similares y, en algunos casos, equivalentes a modelos estadísticos muy conocidos y utilizados en la resolución de problemas de clasificación, regresión y de análisis de conglomerados.

Sí que podemos señalar una importante diferencia entre las redes neuronales y los modelos estadísticos en sus aspectos explicativos de las variables independientes sobre la variable dependiente y es que, a pesar del análisis y los esfuerzos llevados para encontrar el efecto de la importancia de las variables del modelo en las redes neuronales, como se ha explicativo en el epígrafe anterior, no parece que sea tan evidente como los son en los modelos clásicos.

Podemos encontrar diversos estudios en la literatura de redes neuronales donde se manifiestan las equivalencias entre ambas perspectivas. Algunos de estos trabajos, los principales, se referencian a continuación.

Sarle (2002) señala que un Perceptrón simple puede considerase como un Modelo Lineal Generalizado. Según Biganzoli *et al.* (1998) el concepto de discrepancia en un MLG y el concepto de función de error en un Perceptrón también son equivalentes. La función que en general se intenta minimizar, en el caso del Perceptrón es la suma del error cuadrático:

$$E = \sum_{p=1}^{P} \frac{1}{2} \sum_{k=1}^{M} (d_{pk} - y_{pk})^2 \qquad (3.128)$$

donde P hace referencia al número de patrones, M hace referencia al número de neuronas de salida, d_{pk} es la salida deseada para la neurona de salida k para el patrón p e y_{pk} es la salida obtenida por la red para la neurona de salida k para el patrón p.

Normalmente el método del Perceptrón estima los parámetros a través del criterio de los mínimos cuadrados, intentando minimizar la función E y el modelo MLG estima el modelo por el método de máxima verosimilitud. Este método también se puede aplicar a un Perceptrón en tareas de clasificación si asumimos un error con distribución de Bernoulli: Hinton, (1989), Spackman, (1992), Van Ooyen y Nienhuis, (1992); Ohno-Machado, (1997); Biganzoli *et al.* (1998). En este caso, la función de error que se intenta minimizar se denomina cross entropy (Bishop, 1995) cuya fórmula viene dada por la siguiente expresión:

$$E = -\sum_{p=1}^{P} \sum_{k=1}^{M} \left[d_{pk} \log y_{pk} + (1 - d_{pk}) \log(1 - y_{pk}) \right] \qquad (3.129)$$

Cuando se utiliza esta función de error se consigue que las salidas puedan ser interpretadas como probabilidades a posteriori, Bishop (1994)

Un modelo de regresión logística es similar a un Perceptrón simple con función de activación logística en la neurona de salida, Sarle, (1994). La función logística puede ser vista como una generalización no lineal de los MLG, Biganzoli *et al.* 1998).

La Función Discriminante Lineal de Fisher es semejante a un Perceptrón simple con función de activación umbral en la neurona de salida, Kemp *et al.* (1997).

Una red MLP compuesta por tres capas, cuya capa oculta de neuronas utiliza una función de activación no lineal, en general la función logística, puede ser vista como una generalización no lineal de los MLG, Biganzoli *et al.* (1998).

Según qué tipo de función de activación se utilice en la capa de salida, el MLP se puede orientar a la predicción o a la clasificación: En caso de utilizar la función identidad en la capa de salida, estaríamos ante un modelo de regresión no lineal Cheng y Titterington, (1994), Ripley, (1994) y Flexer, (1995). Si la función de activación en la capa de salida es la logística puede ser utilizada como una Función Discriminante no lineal Biganzoli *et al.* (1998).

Otros modelos de Redes Neuronales, a partir de los cuales también se puede establecer una clara analogía con modelos estadísticos clásicos conocido son aquellas arquitecturas de redes entrenadas mediante la regla de Oja (1982 y 1989), las cuales permiten realizar Análisis de Componentes Principales (PCA). La red backpropagation autosupervisada o MLP autoasociativo es otro modelo de red que también ha sido aplicado al PCA y a la reducción de la dimensionalidad. Esta red fue utilizada inicialmente por Cottrell *et al.* (1989)

Las RNA también han sido utilizadas en el análisis de series temporales. El modelado de una serie temporal univariante se realiza habitualmente mediante una red perceptrón multicapa usando un número determinado de términos atrasados como entradas y las previsiones como salidas, Bishop, (1995). También son interesantes las redes recurrentes que resultan de gran utilidad en la previsión de series temporales debido a que son capaces de aprender las relaciones temporales que se establecen entre patrones de entrada y salida, Elman, (1990) y Montaño *et al.* (2011).

Existen otros estudios relacionados con la regresión de Cox y redes SOM que se pueden estudiar en Montaño (2005).

3.2.3. Algoritmos genéticos y otros métodos de búsqueda.

3.2.3.1. Introducción.

Los algoritmos genéticos son uno de los métodos más comunes en minería de datos.

Se inspiran en el proceso natural de selección y evolución tal y como se describe por la teoría evolucionista de la selección natural postulada por Darwin. Los principios sobre los que se asientan los algoritmos genéticos son:

- Los individuos mejor adaptados al entorno son aquellos que tienen una probabilidad mayor de sobrevivir y, por ende, de reproducirse.
- Los descendientes heredan características de sus progenitores.
- De forma esporádica y natural se producen mutaciones en el material genético de algunos individuos, provocando cambios permanentes.

Los algoritmos genéticos son adecuados para obtener buenas aproximaciones en problemas de búsqueda, aprendizaje y optimización [Marczyk. 2004].

De forma esquemática un algoritmo genético es una función matemática que tomando como entrada unos individuos iniciales (población origen) selecciona aquellos ejemplares (también llamados genes) que recombinándose por algún método generarán como resultado la siguiente generación. Esta función se aplicará de forma iterativa hasta verificar alguna condición de parada, bien pueda ser un número máximo de iteraciones o bien la obtención de un individuo que cumpla unas restricciones iniciales.

Los algoritmos genéticos fueron propuestos por Holland (1975), quién intentando simular los procesos naturales de adaptación desarrolló por primera vez la idea de los algoritmos genéticos en los años 60; No obstante, no fue hasta 15 años más tarde cuando un pupilo suyo, David Goldberg (1989) les aplicó por primera vez a un problema real y les popularizó. En 1985 se creó la primera conferencia mundial de algoritmos genéticos ICGA que se celebra hasta el día de hoy bianualmente.

3.2.3.2. Condiciones para la aplicación de los Algoritmos Genéticos.

No es posible la aplicación en toda clase de problemas Algoritmos genéticos. Para que estos puedan aplicarse, los problemas deben cumplir las siguientes condiciones:

- El espacio de búsqueda1 debe estar acotado, por tanto ser finito.
- Es necesario poseer una función de aptitud, que denominaremos fitness, que evalúe cada solución (individuo) indicándonos de forma cuantitativa cuán bueno o mala es una solución concreta.
- Las soluciones deben ser codificables en un lenguaje comprensible para un ordenador, y si es posible de la forma más compacta y abreviada posible.

Habitualmente, la segunda condición es la más complicada de conseguir, para ciertos problemas es trivial la función de *fitness* (por ejemplo en el caso de la búsqueda del máximo de una función) no obstante, en la vida real a veces es muy complicada de obtener y, habitualmente, se realizan conjeturas evaluándose los algoritmos con varias funciones de *fitness*.

3.2.3.3. Ventajas e Inconvenientes.

3.2.3.3.1. Ventajas.

- No necesitan ningún conocimiento particular del problema sobre el que trabajan, únicamente cada ejemplar debe representar una posible solución al problema.
- Es un algoritmo admisible, es decir, con un número de iteraciones *suficiente* son capaces de obtener la solución óptima en problemas de optimización.
- Los algoritmos genéticos son bastante robustos frente a falsas soluciones ya que al realizar una inspección del espacio solución de forma no lineal (por ejemplo, si quisiéramos obtener el máximo absoluto de una función) el algoritmo no recorre la función de forma consecutiva por lo que no se ve afectada por máximos locales.
- Altamente paralelizables (es decir, ya que el cálculo no es lineal podemos utilizar varias máquinas para ejecutar el programa y evaluar así un mayor número de casos).
- Pueden ser incrustrables en muchos algoritmos de data mining para formar modelos híbridos. Por ejemplo para seleccionar el número óptimo de neuronas en un modelo de Perceptrón Multicapa.

3.2.3.3.2. Inconvenientes.

- Su coste computacional puede llegar a ser muy elevado, si el espacio de trabajo es muy grande.

[1] Recordemos que cualquier método de Data Mining se puede asimilar como una búsqueda en el espacio solución, es decir, el espacio formado por todas las posibles soluciones de un problema.

- En el caso de que no se haga un correcto ajuste de los parámetros pueden llegar a caer en una situación de *dominación* en la que se produce un bucle infinito ya que unos individuos *dominan* sobre los demás impidiendo la evolución de la población y por tanto inhiben la diversidad biológica.
- Puede llegar a ser muy complicado encontrar una función de evaluación de cada uno de los individuos para seleccionar los mejores de los peores.

3.2.3.4. Fundamentos Teóricos (Conceptos).

A continuación se explican someramente, los conceptos básicos de los algoritmos genéticos.

3.2.3.4.1. Codificación de los datos.

El primer paso para conseguir que un ordenador procese unos datos es conseguir representarlos de una forma apropiada. En primer término, para codificar los datos, es necesario separar las posibles configuraciones posibles del dominio del problema en un conjunto de estados finito.

Una vez obtenida esta clasificación el objetivo es representar cada estado de forma unívoca con una cadena de caracteres (compuesta en la mayoría de casos por unos y ceros).

A pesar de que cada estado puede codificarse con alfabetos de diferente cardinalidad[2] uno de los resultados fundamentales de la teoría de algoritmos genéticos es el teorema del esquema, que afirma que la codificación óptima es aquella en la que los algoritmos tienen un alfabeto de cardinalidad 2, es decir el uso del alfabeto binario.

El enunciado del teorema del esquema es el siguiente:

«Esquemas cortos, de bajo orden y aptitud superior al promedio reciben un incremento exponencial de representantes en generaciones subsecuentes de un Algoritmo Genético.»

Una de las ventajas de usar un alfabeto binario para la construcción deconfiguraciones de estados es la sencillez de los operadores utilizados para la modificación de estas.

[2] La longitud de las cadenas que representen los posibles estados no es necesario que sea fija, representaciones como la de Kitano para representar operaciones matemáticas son un ejemplo de esto.

En el caso de que el alfabeto sea binario, los operadores se denominan, lógicamente, operadores binarios.

Es importante destacar que variables que estén próximas en el espacio del problema deben preferiblemente estarlo en la codificación ya que la proximidad entre ellas condiciona un elemento determinante en la mutación y reproducibilidad de éstas. Es decir, dos estados que en nuestro espacio de estados del universo del problema están consecutivos deberían estarlo en la representación de los datos, esto es útil para que cuando haya mutaciones los saltos se den entre estados consecutivos. En términos generales cumplir esta premisa mejora experimentalmente los resultados obtenidos con algoritmos genéticos.

En la práctica el factor que condiciona en mayor grado el fracaso o el éxito de la aplicación de Algoritmos Genéticos a un problema dado es una codificación acorde con los datos.

Otra opción muy común es establecer a cada uno de los posibles casos un número natural y luego codificar ese número en binario natural, de esta forma minimizamos el problema que surge al concatenar múltiples variables independientes en el que su representación binaria diera lugar a numerosos *huecos* que produjeran soluciones no válidas.

Por ejemplo, tenemos 3 variables, las dos primeras tienen 3 posibles estados y la última dos, el número posible de estados es 3+3+2 = 8, combinando las 3 variables podemos codificar todo con 3 bits en comparación con los 2+2+1 = 5 bits necesarios que utilizaríamos en el caso de realizar el procedimiento anterior. En este ejemplo no sólo ahorraríamos espacio sino que además evitaríamos que se produjeran individuos cuya solución no es factible.

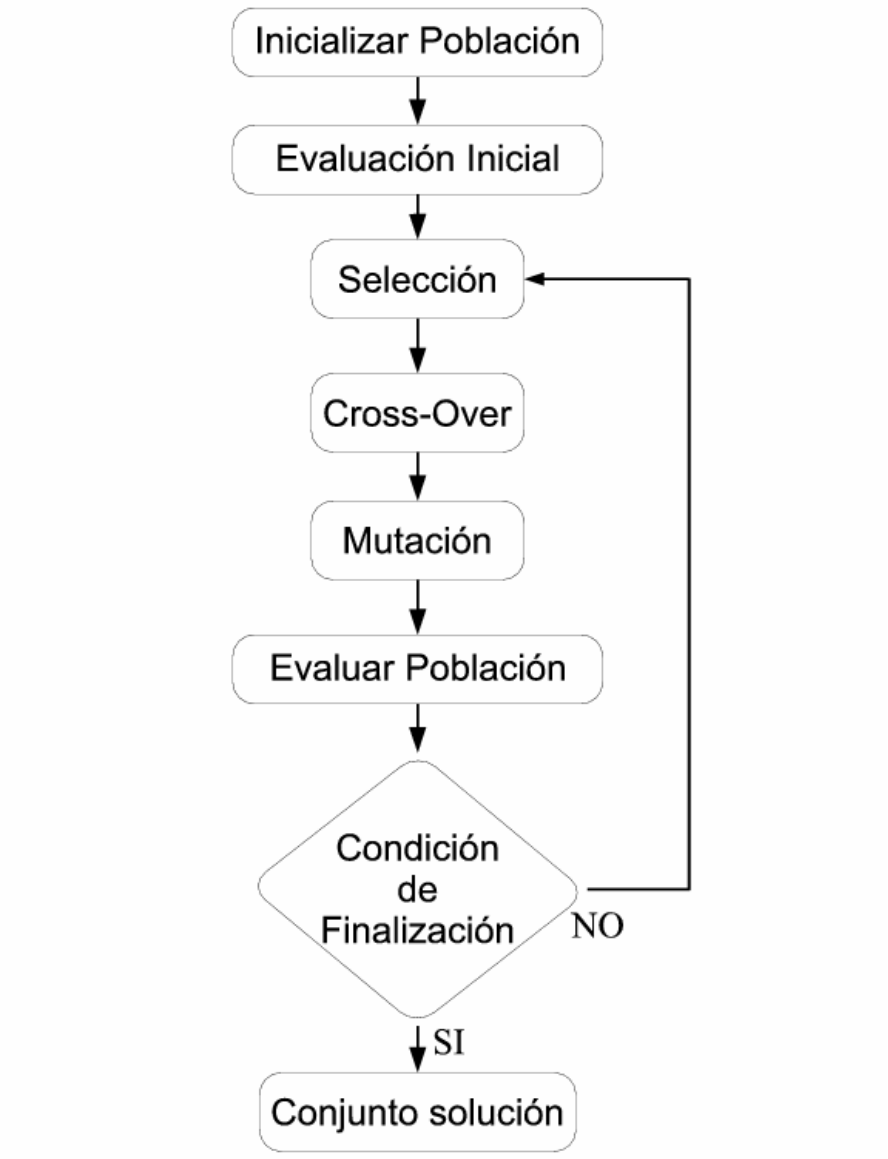
3.2.3.4.2. Algoritmo.

Un algoritmo genético implementado en pseudo código podría ser el siguiente:

Generar de forma aleatoria una serie de genes.
Mientras (condición de terminación es falsa).
{Evaluar el *fitness* de cada uno de los individuos.
Permitir a cada uno de los individuos reproducirse de acuerdo a su *fitness.*
Emparejar los individuos de la nueva población}.

Un posible esquema que puede representar una posible implementación de algoritmos genéticos se muestra en la figura 3.30.

Figura 3.30. Esquema de implementación de un algoritmo genético.



A continuación, en los siguientes apartados, se hará una descripción de las fases anteriormente expuestas:

Inicializar Población.

Como ya se ha explicado antes el primer paso es inicializar la población origen.

Habitualmente la inicialización se hace de forma aleatoria procurando una distribución homogénea en los casos iniciales de prueba. No obstante, si se tiene un conocimiento

más profundo del problema es posible obtener mejores resultados inicializando la población de una forma apropiada a la clase de soluciones que se esperan obtener.

Evaluar Población.

Durante cada iteración (generación) cada gen se decodifica convirtiéndose en un grupo de parámetros del problema y se evalúa el problema con esos datos. Pongamos por ejemplo que queremos evaluar el máximo de la función $f(x)=x^2$ en el intervalo [0,1] y supongamos que construimos cada gen con 6 dígitos ($2^6=64$), por lo que interpretando el número obtenido en binario natural y dividiéndolo entre 64 obtendremos el punto de la función que corresponde al gen (individuo).

Evaluando dicho punto en la función que queremos evaluar ($f(x)=x^2$) obtenemos lo que en nuestro caso sería el *fitness*, en este caso cuanto mayor *fitness* tenga un gen mejor valorado está y más probable es que prospere su descendencia en el futuro.

No en todas las implementaciones de algoritmos genéticos se realiza una fase de evaluación de la población tal y como aquí está descrita, en ciertas ocasiones se omite y no se genera ningún *fitness* asociado a cada estado evaluado.

Selección.

La fase de selección elije los individuos a reproducirse en la próxima generación, esta selección puede realizarse por muy distintos métodos. En el algoritmo mostrado en seudo código anteriormente el método de selección usado depende del *fitness* de cada individuo.

A continuación se describen los más comunes:

- **Selección elitista**: Se seleccionan los individuos con mayor fitness de cada generación.

La mayoría de los algoritmos genéticos no aplican un elitismo puro sino que en cada generación evalúan el *fitness* de cada uno de los individuos, en el caso de que los mejores de la anterior generación sean mejores que los de la actual éstos se copian sin recombinación a la siguiente generación.

- **Selección proporcional a la aptitud**: los individuos más aptos tienen más probabilidad de ser seleccionados, asignándoles una probabilidad de selección más alta. Una vez seleccionadas las probabilidades de selección a cada uno de los individuos se genera una nueva población teniendo en cuenta éstas.

- **Selección por rueda de ruleta**: Es un método conceptualmente similar al anterior. Se le asigna una probabilidad absoluta de aparición de cada individuo de acuerdo al *fitness* de forma que ocupe un tramo del intervalo total de probabilidad (de 0 a 1) de forma acorde a su *fitness*. Una vez completado el tramo total se generan números aleatorios de 0 a 1 de forma que se seleccionen los individuos que serán el caldo de cultivo de la siguiente generación.
- **Selección por torneo**: se eligen subgrupos de individuos de la población, y los miembros de cada subgrupo compiten entre ellos. Sólo se elige a un individuo de cada subgrupo para la reproducción.
- **Selección por rango**: a cada individuo de la población se le asigna un rango numérico basado en su *fitness*, y la selección se basa en este ranking, en lugar de las diferencias absolutas en el *fitness*. La ventaja de este método es que puede evitar que individuos muy aptos ganen dominancia al principio a expensas de los menos aptos, lo que reduciría la diversidad genética de la población y podría obstaculizar la búsqueda de una solución aceptable. Un ejemplo de esto podría ser que al intentar maximizar una función el algoritmo genético convergiera hacía un máximo local que posee un *fitness* mucho mejor que el de sus congéneres de población lo que haría que hubiera una dominancia clara con la consecuente desaparición de los individuos menos aptos (con peor *fitness*).
- **Selección generacional**: la descendencia de los individuos seleccionados en cada generación se convierte en la siguiente generación. No se conservan individuos entre las generaciones.
- **Selección por estado estacionario**: la descendencia de los individuos seleccionados en cada generación vuelven al acervo genético preexistente, reemplazando a algunos de los miembros menos aptos de la siguiente generación. Se conservan algunos individuos entre generaciones.
- **Búsqueda del estado estacionario**: Ordenamos todos los genes por su *fitness* en orden decreciente y eliminamos los últimos *m* genes, que se sustituyen por otros *m* descendientes de los demás. Este método tiende a estabilizarse y converger.
- **Selección jerárquica**: los individuos atraviesan múltiples rondas de selección en cada generación. Las evaluaciones de los primeros niveles son más rápidas y menos discriminatorias, mientras que los que sobreviven hasta niveles más altos son evaluados más rigurosamente. La ventaja de este método es que reduce el tiempo total de cálculo al utilizar una evaluación más rápida y menos selectiva para eliminar a la mayoría de los individuos que se muestran poco o nada prometedores, y

sometiendo a una evaluación de aptitud más rigurosa y computacionalmente más costosa sólo a los que sobreviven a esta prueba inicial.

Recombinación.

Recombinación también llamado *Cross-over*. La recombinación es el operador genético más utilizado y consiste en el intercambio de material genético entre dos elementos al azar (pueden ser incluso entre el mismo elemento). El material genético se intercambia entre bloques. Gracias a la presión selectiva[3] irán predominando los mejores bloques génicos.

Existen diversos tipos de *cross-over*:

- Cross-over uniforme. Se genera un patrón aleatorio en binario, y en los elementos que haya un 1 se realiza intercambio genético.
- *Cross-over de n-puntos*. Los cromosomas se cortan por n puntos y el resultado se intercambia.
- *Cross-over especializados*. En ocasiones, el espacio de soluciones no es continuo y hay soluciones que a pesar de que sean factibles de producirse en el gen no lo son en la realidad, por lo que hay que incluir restricciones al realizar la recombinación que impidan la aparición de algunas combinaciones.

Mutación.

Este fenómeno, generalmente muy raro en la naturaleza, se modela de la siguiente forma: cuando se genera un gen hijo se examinan uno a uno los bits del mismo y se genera un coeficiente aleatorio para cada uno. En el caso de que algún coeficiente supere un cierto umbral se modifica dicho bit. Modificando el umbral podemos variar la probabilidad de la mutación. Las mutaciones son un mecanismo muy interesante por el cual es posible generar nuevos individuos con rasgos distintos a sus predecesores.

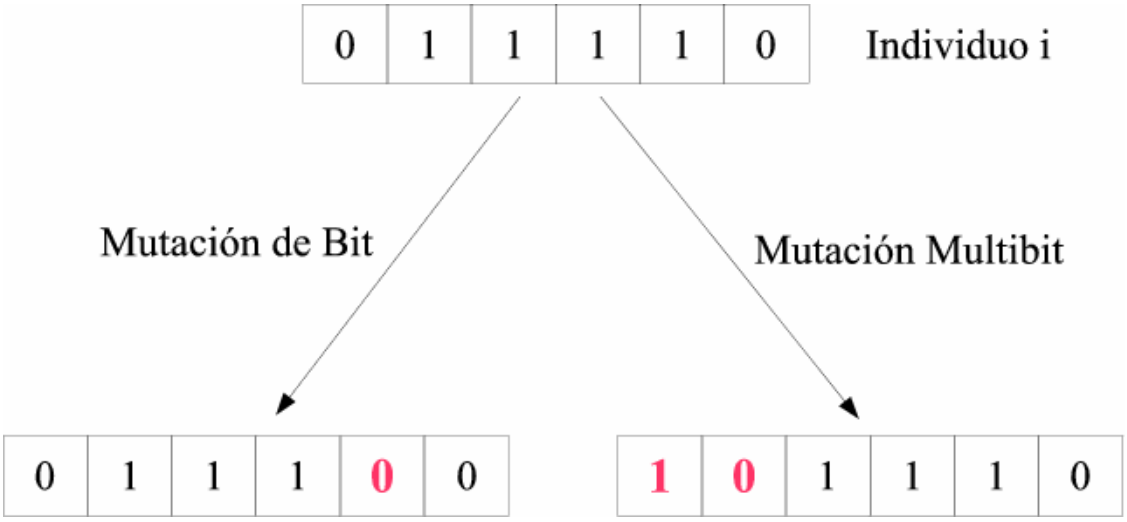
Los tipos de mutación más conocidos son:

- **Mutación de bit**: existe una única probabilidad de que se produzca una mutación de algún bit. De producirse, el algoritmo toma aleatoriamente un bit, y lo invierte.

[3] Presión Selectiva es la *fuerza* a la que se ven sometido naturalmente los genes con el paso del tiempo. Con el sucesivo paso de las generaciones los genes menos útiles estarán sometidos a una mayor presión selectiva produciéndose la paulatina desaparición de estos.

- **Mutación multibit**: cada bit tiene una probabilidad de mutarse o no, que es calculada en cada pasada del operador de mutación multibit.

Figura 3.31. Esquema de mutación multibit de un algoritmo genético.



- **Mutación de gen**[4]: igual que la mutación de bit, sólo que, en vez de cambiar un bit, cambia un gen completo. Puede sumar un valor aleatorio, un valor constante, o introducir un gen aleatorio nuevo.
- **Mutación multigen**: igual que la mutación de multibit, solamente que, en vez de cambiar un conjunto de bits, cambia un conjunto de genes. Al igual que el anterior puede sumar un valor aleatorio, un valor constante, o introducir un gen aleatorio nuevo.
- **Mutación de intercambio**: Se intercambia el contenido de dos bits/genes aleatoriamente.
- **Mutación de barajado**: existe una probabilidad de que se produzca una mutación. De producirse, toma dos bits o dos genes aleatoriamente y baraja de forma aleatoria los bits —o genes, según hubiéramos escogido— comprendidos entre los dos.
- **CREEP**: Este operador aumenta o disminuye en 1 el valor de un gen; sirve para cambiar suavemente y de forma controlada los valores de los genes.

Condición de finalización.

Una vez que se ha generado la nueva población se evalúa la misma y se selecciona a aquel individuo o aquellos que por su *fitness* se consideran los más aptos.

[4] Gen e Individuo en este contexto es lo mismo.

Seleccionados estos se toman y evalúan, y si satisfacen la condición de terminación finaliza el algoritmo.

3.2.3.4.3. Otros Operadores.

Los operadores descritos anteriormente suelen ser operadores generalistas (aplicables y de hecho aplicados a todo tipo de problemas), sin embargo para ciertos contextos suele ser más recomendable el uso de operadores específicos para realizar un recorrido por el espacio de solución más acorde a la solución buscada.

Modificadores de la longitud de los individuos.

En ocasiones las soluciones no son una combinación de todas las variables de entrada, en estas ocasiones los individuos deberán tener una longitud variable[5].

Lógicamente, en este tipo de casos, es necesario modificar la longitud de los individuos, para ello haremos uso de los operadores añadir y quitar, que añadirán o quitarán a un individuo un trozo de su carga génica (es decir, un trozo de información).

3.2.3.4.4. Parámetros necesarios al aplicar Algoritmos Genéticos.

Cualquier algoritmo genético necesita ciertos parámetros que deben fijarse antes de cada ejecución, como:

- **Tamaño de la población**: Determina el tamaño máximo de la población a obtener.

En la práctica debe ser de un valor lo suficientemente grande para permitir diversidad de soluciones e intentar llegar a una buena solución, pero siendo un número que sea computable en un tiempo razonable.

- **Condición de terminación**: Es la condición de parada del algoritmo. Habitualmente es la convergencia de la solución (si es que la hay), un número prefijado de generaciones o una aproximación a la solución con un cierto margen de error.
- **Individuos que intervienen en la reproducción de cada generación**: se especifica el porcentaje de individuos de la población total que formarán parte del acervo de padres de la siguiente generación. Esta proporción es denominada proporción de cruces.

[5] En muchas ocasiones, se realizan estudios de minería de datos sobre todos los datos existentes, encontrándose en ellos variables espúreas, es decir, variables que no aportan nada de información para el problema evaluado.

- **Probabilidad de ocurrencia de una mutación**: En toda ejecución de un algoritmo genético hay que decidir con qué frecuencia se va a aplicar la mutación. Se debe de añadir algún parámetro adicional que indique con qué frecuencia se va a aplicar dentro de cada gen del cromosoma. La frecuencia de aplicación de cada operador estará en función del problema; teniendo en cuenta los efectos de cada operador, tendrá que aplicarse con cierta frecuencia o no. Generalmente, la mutación y otros operadores que generen diversidad se suelen aplicar con poca frecuencia; la recombinación se suele aplicar con frecuencia alta.

3.2.3.4.5. Selección de atributos con Algoritmos Genéticos.

3.2.3.4.5.1. Introducción. Selección de Atributos.

Un problema muy común en cualquier estudio en el que se tenga una gran cantidad de variables es determinar qué relación hay entre las mismas y la *importancia* de éstas en el problema a tratar.

Pongamos como ejemplo el problema de calificar una persona como obesa o no.

Podemos disponer de muchas variables sobre dicha persona tales como su sexo, la raza, el color de ojos, la altura, el peso, etc. Muchas de estas variables son irrelevantes o muy poco útiles para el problema que nos ocupa, por lo que convendría descartarlas para poder disminuir el tamaño de conjunto de elementos a procesar. En este ejemplo las variables irrelevantes serían: sexo, raza y color de ojos; sin embargo, variables relevantes en este problema serían el peso y la altura.

Las ventajas obtenidas por una buena selección de atributos son:

- **Eliminar el ruido**: Eliminando el ruido aumenta la precisión de los datos, y con ello, la capacidad explicativa de las predicciones del modelo.
- **Eliminar variables irrelevantes**: Solamente atendiendo a las variables relevantes se reducen los costes de la toma de datos y el tamaño de las bases de datos.
- **Eliminar redundancias**: Evitando las redundancias se evitan problemas de inconsistencias y de información duplicada.

En términos más formales, el problema de selección de atributos es el de encontrar un subconjunto de los datos tal que aplicando un algoritmo de inducción se maximice la eficiencia de éste.

3.2.3.4.5.2. Subconjunto de atributos óptimo.

Sea un algoritmo de aprendizaje *L* y un conjunto de instancias *X* con atributos *X1, X2,…, Xn* con una distribución *D* del espacio de instancias. Se denomina subconjunto óptimo *Xopt* al subconjunto de atributos que consiguen que la eficiencia del clasificador *C=L(D)* sea máxima.

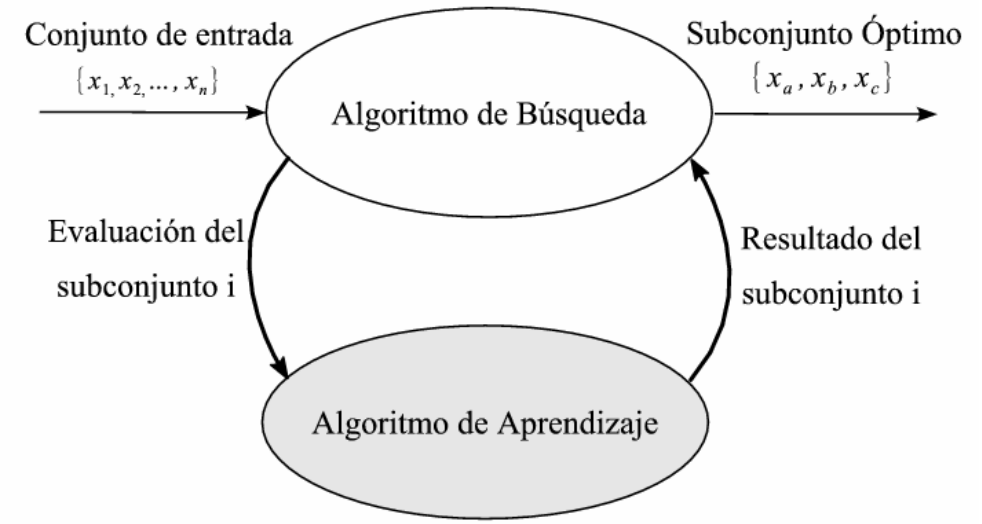
Pero, el subconjunto de atributos óptimo no tiene por qué ser único, es posible que haya combinaciones de atributos que consigan la misma eficiencia del clasificador. Un ejemplo de esto se da cuando existen dos atributos perfectamente correlacionados, en estos casos da igual el atributo que seleccionemos. Habitualmente se selecciona, si es posible, el subconjunto de atributos óptimo que sea mínimo.

Un atributo es relevante en un conjunto dado, cuando éste es significativo respecto a los demás.

El procedimiento utilizado para la selección de atributos es utilizar un método de envoltorio (*wrapper*) que se compone de un algoritmo de búsqueda, en nuestro caso Algoritmos genéticos junto a un algoritmo de aprendizaje que, en este contexto es el algoritmo que calcula el *fitness* asociado a cada uno de los individuos (subconjuntos de prueba) presentados por el algoritmo de búsqueda.

El esquema de este método se muestra en la siguiente figura.

Figura 3.32. Selección de atributos a través un algoritmo genético.



Como algoritmo de aprendizaje utilizaremos el denominado *cfssubseteval* que evalúa el *fitness* calculando la correlación entre el individuo presentado y la clasificación,

eligiendo los más correlacionados pero penalizando la intercorrelación entre los miembros de una misma iteración (generación). Este método es muy interesante ya que es un buen partidario para utilizarlo para cualquier tipo de problemas gracias a su simplicidad.

No obstante, no es ni mucho el óptimo, recordemos que en el Data Mining la eficiencia de los métodos se debe en gran medida al conocimiento del problema en sí, o al menos del dominio de este.

3.2.3.5. Conclusiones.

Los algoritmos genéticos es uno de los enfoques más originales en data mining. Su sencillez, combinada con su flexibilidad les proporciona una robustez que les hace adecuados a infinidad de problemas. No obstante, su simplicidad y sobre todo independencia del problema hace que sean algoritmos poco específicos.

Recorriendo este capítulo hemos visto los numerosos parámetros y métodos aplicables a los algoritmos genéticos que nos ayudan a realizar una adaptación de los algoritmos genéticos más concreta a un problema.

En definitiva, la implementación de esquemas evolutivos tal y como se describen en biología podemos afirmar que funciona.

3.2.4. Máquinas de vectores soporte.

3.2.4.1. Introducción.

En los últimos años, la aplicación de Máquinas de Vectores Soporte para resolver tanto problemas de clasificación como de regresión se ha incrementado notablemente debido, fundamentalmente a su alto rendimiento de forma general y su capacidad para modelar relaciones no lineales.

Los fundamentos teóricos de las máquinas de vectores soporte (Support Vector Machines, SVM) fueron presentados en el año 1992 en la conferencia COLT (Computacional Learning Theory) por Boser *et al.* (1992) y descritos posteriormente en diversos artículos por Cortes y Vapnik (1995) y Vapnik (1998 y 2000) a partir de los trabajos sobre la teoría del aprendizaje estadístico. El interés por este modelo de aprendizaje no ha parado de crecer, adquiriendo en poco tiempo cierta popularidad y en estos momentos, sobre las SVM, podemos afirmar que han alcanzado un lugar importante dentro de las técnicas utilizadas en minería de datos como disciplina de aprendizaje automático.

Las máquinas de vectores soporte pertenecen a la familia de los clasificadores lineales dado que inducen hiperplanos o separadores lineales de muy alta dimensionalidad introducidos por funciones núcleo o kernel. Es decir, el enfoque de las SVM adopta un punto de vista no habitual, en vez de reducir la dimensión buscan una dimensión mayor en la cual los puntos puedan separarse linealmente.

Las SVM, al igual que el perceptrón multicapa y las funciones de base radial se utilizan tanto para solucionar problemas de clasificación como para regresión no lineal.

3.2.4.2. Máquinas de vectores soporte con margen máximo.

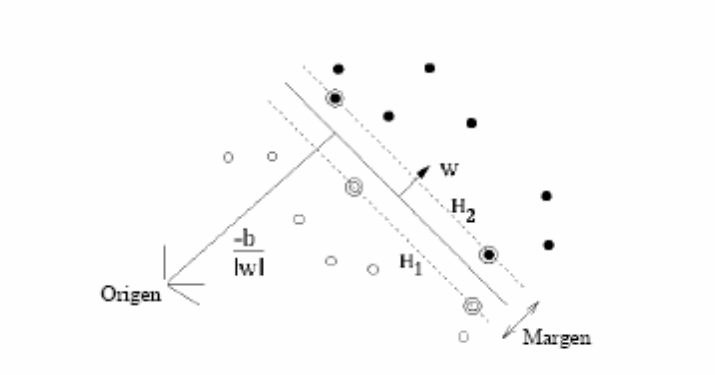
La formulación matemática de las máquinas de vectores soporte se basa en el principio de minimización estructural del riesgo, que ha demostrado ser superior al principio de minimización del riesgo empírico utilizado en muchas de las redes convencionales.

Vamos a suponer que disponemos de una muestra S de N elementos del tipo (x_i , y_i) donde $S=\{ (x_1 , z_1), (x_2 , z_2)..... (x_N , z_N)\}$.

Donde xi pertenece a un espacio de entrada y z_i toma los valores -1 y 1. x_i Є RP es el vector de variables para la observación i.

El conjunto de N datos es linealmente separable si es posible encontrar un vector w Є RP que defina un plano que separe los puntos de ambas clases.

Figura 3.33. Separación de datos con margen máximo.



Fuente: Burges (1998).

Un conjunto de observaciones se encuentra a un lado y verifica la siguiente ecuación:

$$w'x_i + b \leq -1 \qquad (3.130.a)$$

El otro conjunto de datos verifica

$$w'x_i + b \geq -1 \qquad (3.130.b)$$

Estas dos expresiones pueden escribirse como:

$$z\,(w'\,x_i + b) \geq 1 \text{ para i} = 1,2,.....N \qquad (3.130.c)$$

Sea $f(x) = w'x_i + b$ el valor del hiperplano que separa óptimamente el conjunto de puntos.

La distancia entre x_i y el hiperplano viene dada por la proyección del punto x_i en la w. Siendo w el vector ortogonal al plano.

La proyección se calcula a través de la siguiente expresión:

$$\frac{w'\,x_i}{\|w\|} \qquad (3.131)$$

$\|w\|$ es la norma en el espacio R^p.

Maximizaremos la distancia de los puntos al plano maximizando la siguiente expresión:

$$\frac{z_i\left(w' x_i + b\right)}{\|w\|} \tag{3.132}$$

Conseguiremos maximizar la anterior expresión siempre que el numerador sea positivo y el denominador lo más pequeño posible, lo que conduce a resolver un problema de programación cuadrática convexo bajo restricciones en forma de desigualdad lineal que se expresa de la siguiente forma:

$$\min \frac{1}{2}\|w\|^2$$

Sujeto a $z_i\,(w'\,x_i + b) \geq 1$ para i = 1.....N.

Se define el margen funcional para un ejemplo (x , z) con respecto a una función f como el producto $z^*f(x)$, mientras que el margen normalizado geométrico de un hiperplano como 1/||w||. La solución de una SVM lineal con margen máximo es el hiperplano que maximiza el margen geométrico sujeto o restringido a que el margen funcional sea mayor o igual que 1.

El enfoque que utilizan los métodos clásicos es proyectar los datos sobre un espacio de dimensión menor y utilizar una función no lineal para discriminar, mientras que las SVM aplican una transformación de los datos de forma que los lleve a un espacio de dimensión mayor que p, y aplicar entonces una discriminación lineal como la anterior.

La forma habitual de resolver problemas de optimización con restricciones es utilizando la teoría desarrollada en 1797 por Lagrange y extendida después, para restricciones en forma de desigualdad, por Kuhn y Tucker en 1951. Su famoso teorema nos permite obtener una alternativa que se conoce como forma dual y que es equivalente a la forma primal, pero que podemos expresar como una combinación lineal de los vectores de aprendizaje.

Introducimos N multiplicadores de Lagrange, uno para cada una de las restricciones y que denominaremos por α_1, α_2 … α_N . Para las restricciones de forma ci > 0 se

multiplican por los multiplicadores positivos y se restan de la función objetivo para formar la función de Lagrange generalizada, cuya expresión es la siguiente:

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2+\sum_{i=1}^{N}\alpha_1 z_1(w' x_i+b)+\sum_{i=1}^{N}\alpha_i \tag{3.133}$$

Derivando con respecto al vector de pesos **w** con respecto a **b** obtenemos:

$$\frac{\delta L(w,b,\alpha)}{\delta w}=0\Rightarrow w-\sum_{i=1}^{\tilde{N}} z_i\alpha_i x_i=0\Rightarrow w=\sum_{i=1}^{N} z_i\alpha_i x_i \tag{3.134}$$

$$\frac{\delta L(w,b,\alpha)}{\delta b}=0\Rightarrow \sum_{i=1}^{N} z_i\alpha_i=0 \tag{3.135}$$

Las expresiones que resultan de igualar las derivadas a cero las podemos sustituir en la función de Lagrange y obtenemos la forma dual cuya expresión resultante es la siguiente:

$$L(w,b,\alpha)=L_D=\sum_{i}^{N}\alpha_i-\frac{1}{2}\sum_{i,j}^{N} z_i z_j\alpha_i\alpha_j x_i x_j \tag{3.136}$$

Para obtener la solución en su forma dual, suponiendo que el conjunto de los datos es linealmente separable en el espacio de entrada debemos de maximizar ahora la función LD sujeta a dos restricciones:

Maximizar

$$L(w,b,\alpha)=L_D=\sum_{i}^{N}\alpha_i-\frac{1}{2}\sum_{i,j}^{N} z_i z_j\alpha_i\alpha_j x_i x_j \tag{3.137}$$

Sujeta a $\sum_{i}^{N} z_i\alpha_i$

$\alpha_i\geq 0 \quad 1\leq iN$

Entonces el vector $w^*=\sum_{i=1}^{N} z_i\alpha^*{}_i x_i$ es el vector ortogonal al hiperplano con margen geométrico máximo. Este valor está determinado por el algoritmo de entrenamiento pero en el caso del umbral *b* no es el caso aunque la forma de obtenerlo es inmediata,

tomando en la ecuación 3.137 cualquier i para el que $\alpha_i \neq 0$ (por ejemplo, $z_i=1$) obtenemos la expresión siguiente:

$$b = 1 - w' x_i \tag{3.138}$$

Obsérvese que hay un multiplicador de Lagrange para αi para cada punto de entrenamiento. Cuando se obtiene una solución, aquellos puntos para los que αi >0 se denominan ***vectores soporte*** y están sobre los hiperplanos H1 y H2. El resto de los puntos cumplen que $\alpha_i=0$. Los vectores soporte son los elementos que están más próximos a la frontera de decisión.

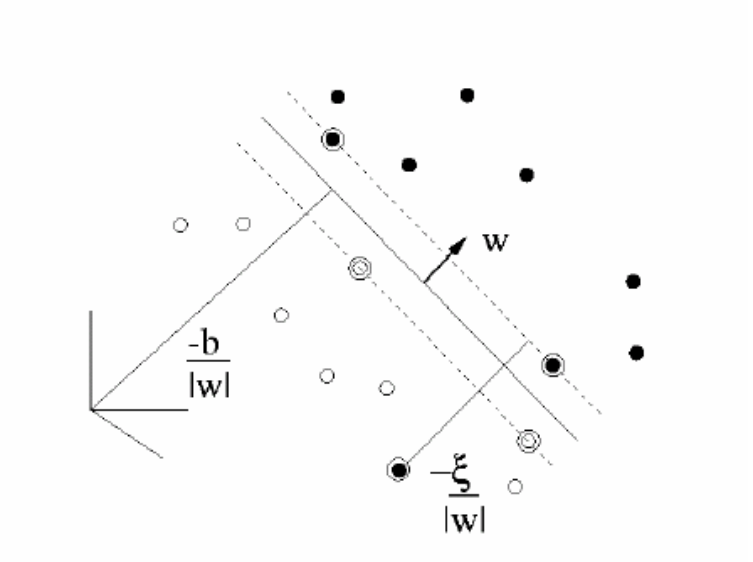
Fase de evaluación.

Cuando se ha entrenado una máquina de vectores soporte a la hora de evaluar un patrón nuevo x sólo hay que saber en qué parte de la frontera de decisión se encuentra y asignarle la etiqueta correspondiente (+1 ó -1) a través de la función **sgn(w' x + b),** donde sgn es la función signo.

3.2.4.3. Máquinas de vectores soporte con margen blando y norma 1 de las variables de holgura.

No siempre es posible encontrar una transformación de los datos que nos permita separar linealmente los datos bien sea en el espacio de entrada o en el espacio de las características (ver en el epígrafe siguiente) debido errores de medición, ejemplos mal etiquetados, valores atípicos, etcétera, lo que nos puede llevar a soluciones de las SVM que generalicen mal.

Figura 3.34. Separación de datos con margen blando.



Fuente: Burges (1998).

Cuando este es el caso nos interesa poder rebajar las restricciones impuestas en 3.137 añadiendo a la ecuación lo que se conoce como variables de holgura (slack) y que generalmente se designan con la letra griega ξ. Estas variables las introduciremos en la función objetivo a optimizar. También incluiremos un valor C o término de regulación que nos va determinar la holgura del margen blando. Esta constante C hay que fijarla de antemano.

Siguiendo la formulación de este capítulo especificaremos el modelo de forma matemática:

Partimos de las siguientes relaciones:

$w'x_i + b \geq +1-\xi_i$ para $z_i = +1$ (3.139.a)

$w'x_i + b \leq -1-\xi_i$ para $z_i = -1$ (3.139.b)

$\xi i > 0$ ξ_i i=1,……N (3.139.c)

Este problema es de nuevo un problema de programación cuadrática donde los ξi ni sus multiplicadores de Lagrange asociados aparecen en la forma dual de Wolfe y que podemos expresar de la siguiente forma:

$$\text{Maximizar } L(w,b,\alpha) = L_D = \sum_i^N \alpha_i - \frac{1}{2}\sum_{i,j}^N z_i z_j \alpha_i \alpha_j x_i x_j \qquad (3.140)$$

Sujeto a las siguientes restricciones

$$\sum_{i=1}^N z_i \alpha_i = 0$$

$0 \le \alpha_i \le C \qquad 1 \le i \le N$

La solución viene de nuevo dada por la expresión:

$$w^* = \sum_{i=1}^{N^S} z_i \alpha^*{}_i x_i \qquad (3.141)$$

Donde N^S representa el número de vectores soporte y s_i es el i-èsimo vector soporte.

La única diferencia con el caso del hiperplano óptimo es que los valores αi están acotados superiormente por C.

3.2.4.4. Máquinas de vector soporte con margen máximo en el espacio de las características. Máquinas no lineales de vectores soporte.

Los métodos anteriormente descritos se pueden generalizar al caso en el que las funciones de decisión no sean lineales para separar los datos.

Si observamos lo desarrollado en la sección anterior, por ejemplo las ecuaciones 3.139.a, 3.139.b y 3.139.c podemos ver que la SVM lineal con margen máximo sólo depende de la existencia de productos escalares $x_i.y_j$ en el espacio de los datos.

Las máquinas de vectores soporte con margen máximo en el espacio de las características se basan en la idea de realizar una transformación no lineal del espacio de entrada a otro espacio de dimensión superior que esté dotado de producto escalar. A este espacio se le conoce como espacio de Hilbert.

Podemos expresar lo dicho anteriormente suponiendo que existe una transformación no lineal del espacio de entrada en un cierto espacio de características H.

$$\phi : \Re^p \to H$$

$$x \to \phi(x)$$

Que está dotado de un producto escalar <Φ(x), Φ(z)> (H es un espacio de Hilbert).

Si el conjunto de los datos de entrada son linealmente separables entonces la SVM con margen máximo se puede obtener sustituyendo el producto escalar <x, y> por <Φ(x), Φ(z)>.

3.2.4.5. Funciones Kernel.

Hay que tener en cuenta que la dimensión del espacio necesario para separar los datos puede ser grande aumentando el coste computacional. Sin embargo, existe una forma muy efectiva de calcular los productos escalares en el espacio de las características a través de ciertas transformaciones usando las denominadas funciones núcleo (funciones kernel).

Una función kernel es una función K: X . X→ R tal que K(X, Y) = <Φ(x), Φ(z)> donde Φ es una transformación de X en un espacio de Hilbert, H.

Sin embargo, hay una gran cantidad de posibles funciones núcleo que pueden ser utilizadas para crear tal espacio de características de alta dimensional.

Algunas funciones núcleo inicialmente utilizadas y de propósito general son las siguientes:

Polinómica:

$$K(x_i, y_j) = (x_i.y_j + 1)^p \qquad (3.142)$$

Gaussiana:

$$K(x_i, y_j) = \exp\left(-\frac{\left\|x_i - y_j\right\|^2}{2\sigma^2}\right) \qquad (3.143)$$

Sigmoidal o tangente:

$$K(x_i, y_j) = \tanh(ax_i.y_j + b) \quad a, b \in R \qquad (3.144)$$

Multicuadrática inversa:

$$K(x_i, y_j) = \frac{1}{\sqrt{\left\|x_i - y_j\right\|^2 + c^2}} \quad c \geq 0 \tag{3.145}$$

Además de las funciones gaussina y sigmoide, Ivanciuc (2007) presenta otras funciones núcleo que, como las anteriores, dependen de algunos parámetros que pueden ser calculados mediante diferentes métodos empíricos o estadísticos. Estas funciones kernel se especifican de la siguiente forma:

Anova Kernel

$$K(x_i, y_j) = (\sum_i \exp(-\gamma(x_i - y_j)^d \tag{3.146}$$

Fourier Series kernel

$$K(x_i, y_j) = \frac{sen(N + 1/2)(x_i - y_j)}{sen(1/2(x_i - y_j))} \tag{3.147}$$

Spline kernel

$$K(x_i, y_j) = (\sum_{r=0}^{k} x_i^r y_j^r) + \sum_{s=1}^{N} (x_i - t_s)^k + (y_j - t_s)^k \tag{3.148}$$

Additive Kernel

$$K(x_i, y_j) = \sum_n K_n(x_i, y_j) \tag{3.149}$$

Tensor Product kernel

$$K(x_i, y_j) = \prod_n K_n(x_i, y_j) \tag{3.150}$$

En opinión de Uestuen *et al.* (2006) las funciones kernel más utilizados son la función de base radial (RBF) y las funciones de producto interno polinomio lineal. Y, dado que la naturaleza de los datos es generalmente desconocida, los especialistas opinan que es muy difícil realizar, de antemano, una elección adecuada de los núcleos mencionados. Por esta razón, además de que en la fase de construcción de modelos

el proceso de optimización consume mucho tiempo, estos autores proponen una función kernel universal basada en la función Pearson VII (PUK):

$$K(x_i, y_j) = \frac{1}{\left[1 + \left(\frac{2\sqrt{\| x_i - y_j \|^2}\ \sqrt{2^{(1/w)} - 1}}{\sigma}\right)^2\right]^w} \tag{3.151}$$

De la investigación llevada a cabo por estos científicos se concluye que el kernel que presentan es robusto y adquiere una potencia igual o incluso más fuerte en comparación con las funciones del kernel estándar, lo que conduce a un desempeño igual o mejor de la SVM. En resumen, Uestuen *et al.* (2006), afirman que el kernel PUK se puede utilizar como un núcleo universal que es capaz de servir como una alternativa genérica a la funciones del núcleo RBF lineal común y polinómico.

3.2.4.6. Aplicaciones de las SVM.

El éxito de la máquinas de vector soporte, debido fundamentalmente a su solidez teórica, ha sido constante en múltiples campos del conocimiento, de tal forma que algunos autores afirman que, en los próximos años, pueden desplazar en muchas aplicaciones a las redes neuronales (Schölkopf y Smola 2002).

Las SVM están siendo aplicadas a numerosos problemas reales en áreas como la recuperación de información, la clasificación de imágenes, clasificación y categorización de textos, análisis de biosecuencias, etcétera. Las SVM también pueden aplicarse para el caso de la regresión y en modelos de aprendizaje no supervisado.

Algunas recomendaciones que se pueden ofrecer para su correcta utilización son las siguientes:

- Antes de resolver otras cuestiones básicas hay que normalizar los datos.
- Elegir el tipo de función núcleo que va a utilizar el algoritmo.
- Resolver la dureza del margen que utilizará la SVM, escogiendo entre el modelo SVM con margen máximo o una versión con margen blando controlado por él.

3.2.5. Modelos probabilísticos de elección binaria. Regresión logística.

3.2.5.1. Introducción.

Históricamente el modelo de regresión logística, tal y como se ha visto en el estudio del estado del arte, ha sido el método estadístico más empleado para determinar la probabilidad de default en los modelos de credit scoring. En este método estadístico se trata de explicar la probabilidad de que se devuelva el crédito o no en función de todas las variables explicativas que intervienen en el modelo.

Otra forma de abordar el problema de clasificación, si la distribución de probabilidad es conocida y si se verifica la normalidad multivariante de la distribución, es utilizar el análisis clásico discriminante de Fisher como técnica ideal para abordar problemas de clasificación. En este tipo de análisis un supuesto clave es que las matrices de covarianzas y de dispersión, en principio desconocidas, sean iguales para los grupos. El incumplimiento de estos supuestos puede alterar sensiblemente la estimación de los parámetros de la ecuación discriminante.

Si todas las variables son continuas es frecuente que, aunque los datos originales no sean normales, sea posible transformar las variables para cumplir las hipótesis de aplicación del modelo. La hipótesis de normalidad multivariante es poco realista cuando algunas de las variables explicativas son discretas. Otras características que pueden afectar a la ecuación estimada son la multicolinealidad, el incumplimiento de la no linealidad del modelo y los casos atípicos presentes en los datos.

Cuando tenemos variables categóricas o las hipótesis en que se basa el análisis de Fisher no se verifican podemos utilizar, como buena alternativa, el modelo de regresión logística. La regresión logística no se enfrenta a los dos supuestos básicos tan estrictos para realizar el análisis discriminante y, además, es mucho más robusta cuando los supuestos no se verifican, lo que la hace mucho más apropiada en múltiples situaciones. Muchos investigadores la prefieren porque es similar a la regresión, sin embargo, se diferencia de la regresión múltiple en que predice directamente la probabilidad de ocurrencia de un suceso.

El odds asociado a cierto suceso se define como la razón entre la probabilidad de que ocurra dicho suceso y la probabilidad de que no ocurra. Si llamamos E a dicho suceso, en nuestro caso que el solicitante de un crédito lo devuelva, P(E) es la probabilidad de dicho suceso y O(E) al odds que le corresponde, entonces tenemos :

$$O(E) = \frac{P(E)}{1 - P(E)} \tag{3.152}$$

En su forma logarítmica este valor se denomina logit (del inglés log-unit), tomando el valor 0 cuando *P(E) = 0,5*, $-\infty$ si *P(E) = 0* y $+\infty$ si *P(E) = 1*, es decir:

$$Logit\left(P\left(E\right)\right) = \ln\left(\frac{P\left(E\right)}{1 - P\left(E\right)}\right) \tag{3.153}$$

El modelo binario logit lo podemos expresar de la siguiente manera:

$$Logit\left(P\left(E\right)\right) = \ln\left(\frac{P\left(E\right)}{1 - P\left(E\right)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p \tag{3.154}$$

Para el caso que nos ocupa, donde sólo disponemos de dos alternativas: que el cliente devuelva o no el crédito, podemos formular la ecuación anterior de esta otra forma:

$$P(Y = 1) = \frac{\exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p)}{1 + \exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p)} \tag{3.155}$$

Donde *p* es el número de variables explicativas y la variable Y es una variable dicotómica que tomará el valor 1 si el cliente devuelve el préstamo y 0 si no lo hace, es decir:

$$Y = \begin{cases} 0 & 1 - P(E) \\ 1 & P(E) \end{cases}$$

Y también de esta otra manera:

$$P(Y = 1) = \frac{1}{1 + \exp(-\beta_0 - \beta_1 x_2 - \beta_1 x_2 - ..... - \beta_n x_n)} \tag{3.156}$$

3.2.5.2. Cálculo de los parámetros del modelo de regresión logística (método de máxima verosimilitud).

El modelo de regresión logística asume que:

$Y_1=1$ con probabilidad $P_i(E) = \dfrac{e^{\sum_{j=0}^{p} b_j x_{ij}}}{1+e^{\sum_{j=0}^{p} b_j x_{ij}}}$ (3.157)

O equivalentemente:

$$Y_1=1 \text{ con probabilidad } P_i(E) = \frac{1}{1+e^{-\sum_{j=0}^{p} b_j x_{ij}}} \qquad (3.158)$$

donde $x_{i0}=1$.

Los parámetros desconocidos del modelo son: $b_0, b_1, \ldots, b_p$.

La función de verosimilitud de la muestra contendrá factores del tipo (1) si $y_i=1$, y del tipo (2) si $y_i=0$, con lo que la función de verosimilitud L será:

$$L(b_0,b_1,\ldots b_p)\frac{\prod_{i=1}^{n} e\left(y_i \sum_{j=0}^{p} b_j x_{ij}\right)}{\prod_{i=1}^{n}\left(1+e\sum_{j=0}^{p} b_j x_{ij}\right)} = \frac{e\sum_{j=0}^{p} b_j t_j}{\prod_{i=1}^{n}\left(1+e\sum_{j=0}^{p} b_j x_{ij}\right)} \qquad (3.159)$$

Con $t_j = \sum_{i=1}^{n} x_{ij} y_i$

El Ln de L queda como:

$$Ln(b_0,\ldots b_p) = \sum_{j=0}^{p} b_j t_j - \sum_{j=0}^{p} Ln\ [1+e^{\sum_{j=0}^{p} b_j x_{ij}}] \qquad (3.160)$$

Los b_j que maximizan L(.), también maximizarán Ln[L(.)] y se obtienen resolviendo el siguiente sistema de p+1 ecuaciones no lineales:

$$I^*_{j_1 j_2} = -\sum_{i=1}^{n} \frac{x_{ij} e\sum_{j=0}^{p} b_j x_{ij}}{\left[1+e\sum_{j=0}^{p} b_j x_{ij}\right]} = 0 \qquad j=0,1,\ldots,p \qquad (3.161)$$

Para resolver este sistema se usa el método de Newton-Raphson que necesita de la matriz de segundas derivadas I^*_{j1j2} que viene dada por:

$$I_{j_1 j_2} = -\sum_{i=1}^{n} \frac{x_{ij_1} x_{ij_2} e^{\sum_{j=0}^{p} b_j x_{ij}}}{\left[1 + e^{\sum_{j=0}^{p} b_j x_{ij}}\right]} = 0 \qquad j_1, j_2 = 0,1\ldots,p \qquad (3.162)$$

Llamando I_{j1j2} se tiene que I^{-1}_{j1j2} es la matriz asintótica de varianzas y covarianzas de los $b_j^{'}$s.

Un IC(1-α) para los b_j se puede obtener usando:

$$b_j \pm z_1 - \frac{\alpha}{2}\sqrt{I^{-1}_{jj}} \qquad (3.163)$$

3.2.5.3. Evaluación del modelo.

Para evaluar si cada variable individualmente contribuye significativamente al valor del modelo lo realizamos a través del estadístico de Wald, que es el cociente entre el valor del parámetro estimado de cada variable dividido entre su desviación típica:

$$W(b_j) = \frac{\left(\hat{b}_j\right)}{\sigma(b_j)} \qquad (3.164)$$

Este estadístico sigue una distribución χ^2 con un grado de libertad. Comparando este valor con el valor de las tablas, una vez fijado el error del tipo I, podemos decidir si la variable es significativa para el análisis o por el contrario su aportación es nula.

Una vez que se ha estimado el modelo y se han evaluado los coeficientes de forma individual se debe de efectuar una comprobación a nivel global, es decir, saber cuán de bueno es el ajuste de los valores predichos por el modelo a los valores realmente observados. Para decidir si la bondad del ajuste es adecuada existen varios métodos, los cuales se pueden agrupar según utilicen patrones de covariables, probabilidades estimadas o se basen en la estimación de los residuos.

En cualquier caso, se ha de partir de la especificación de una hipótesis nula y la alternativa que en un contraste global se definen de la siguiente manera:

$$H_o : p_j = \frac{\exp(\beta_0 + \beta_1 x)}{1 + \exp(\beta_0 + \beta_1 x)} \quad \forall j = 1,\ldots,J \tag{3.165}$$

$$H_1 : p_j \neq \frac{\exp(\beta_0 + \beta_1 x)}{1 + \exp(\beta_0 + \beta_1 x)} \quad \text{para algún j} \tag{3.166}$$

3.2.5.3.1. Contraste basados en patrones de las covariables.

Los dos test más utilizados para el ajuste global que comparan valores predichos y observados son la Devianza, D, (Desviación) y el estadístico χ^2 de Pearson.

3.2.5.3.1.1. Estadístico basado en la Devianza (D).

Siguiendo el trabajo de Collet (1991), para datos agrupados, en la construcción de este estadístico primero se considera la función de verosimilitud y su log-verosimilitud:

$$L(\beta_0\beta_1) = \prod_{j=1}^{J} \binom{n_j}{y_j} p_j^{yj} (1 - p_j)^{n_j - y_j} \tag{3.167}$$

$$\log L(\beta_0, \beta_1) = \sum_{j=1}^{J} \left\{ \log \binom{n_j}{y_j} + y_j \log p_j + (n_j - y_j) \log(1 - p_j) \right\} \tag{3.168}$$

Si definimos a $\hat{L}_C = L(\hat{\beta}_0, \hat{\beta}_1)$ obtenemos que la log-verosimilitud estimada como:

$$\log \hat{L}_C = \sum_{j=1}^{J} \left\{ \log \binom{n_j}{y_j} + y_j \log \hat{p}_j + (n_j - y_j) \log(1 - \hat{p}_j) \right\} \tag{3.169}$$

Donde $\hat{p}_j = \frac{\hat{y}_j}{n_j}$ representa la probabilidad estimada de la respuesta *Y=1* para el j-ésimo patrón de covariables.

Si consideramos ahora el modelo saturado como aquel modelo que se ajusta perfectamente a los datos, su log-verosimilitud vendrá dada por:

$$\log \hat{L}_F = \sum_{j=1}^{J} \left\{ \log \binom{n_j}{y_j} + y_j \log \tilde{p}_j + (n_j - y_j) \log(1 - \tilde{p}_j) \right\} \tag{3.170}$$

Ahora $\tilde{p}_j = \frac{y_j}{n_j}$ nos muestra la proporción observada de la respuesta *Y=1* para el j-ésimo patrón de covariables.

Con estas dos últimas expresiones ya podemos construir una medida, a través de su cociente, que nos sirva para ver la bondad del ajuste del modelo a los datos observados, aunque es más útil que sean comparadas multiplicadas por -2, lo que origina el llamado contraste de la devianza o contraste de Wilks.

$$D = -2\log\left(\frac{\hat{L}_C}{\hat{L}_F}\right) = -2\left(\log \hat{L}_C - \log \hat{L}_F\right) \tag{3.171}$$

$$D = 2\sum_{j=1}^{J}\left(\log\left(\frac{\tilde{p}_j}{\hat{p}_j}\right) + \left(n_j - y_j\right)\log\left(\frac{1-\tilde{p}_j}{1-\hat{p}_j}\right)\right) \tag{3.172}$$

$$D = 2\sum_{j=1}^{J}\left(y_j \log\left(\frac{y_j}{\hat{y}_j}\right) + \left(n_j - y_j\right)\log\left(\frac{n_j - y_j}{n_j - \hat{y}_j}\right)\right) \tag{3.173}$$

El estadístico de Wilks tiene una distribución asintótica χ^2 de Pearson cuyos grados de libertad vienen determinados por la dimensión ente el espacio paramétrico y la dimensión de este espacio bajo la hipótesis nula.

Por su parte, esta devianza puede expresarse como una suma de cuadrados de la siguiente forma:

$$D = \sum_{j=1}^{J} d_j^2 \tag{3.174}$$

Donde d_j es igual a:

$$d_j = signo\left(y_j - \hat{y}_j\right)\left[2\left(y_j \log\left(\frac{y_j}{\hat{y}_j}\right) + \left(n_j - y_j\right)\log\left(\frac{n_j - y_j}{n_j - \hat{y}_j}\right)\right)\right]^{1/2} \tag{3.175}$$

La verosimilitud bajo el modelo ajustado, para datos no agrupados, y en un modelo de Bernoulli se expresa así:

$$\log \hat{L}_C = \sum_{j=1}^{N} \left\{ y_j \log \hat{p}_j + \left(1 - y_i \log\left(1 - \hat{p}_j\right)\right)\right\} \tag{3.176}$$

3.2.5.3.1.2. Estadístico Chi Cuadrado de Pearson χ^2.

En la construcción del estadístico χ^2 se comparan las frecuencias observadas y esperadas bajo un modelo binomial:

$$\chi^2 = \sum_{j=1}^{J} \frac{\left(y_j - n_j \hat{p}_j\right)^2}{n_j \hat{p}_j \left(1 - \hat{p}_j\right)} = \sum_{j=1}^{J} \frac{n_j \left(y_j - \hat{y}_j\right)^2}{\hat{y}_j \left(n_j - \hat{y}_j\right)} \tag{3.177}$$

Este estadístico sigue una distribución asintótica con los mismos grados de libertad que la Devianza calculada anteriormente.

Este estadístico anterior puede calcularse, siguiendo a Hosmer como las sumas de los cuadrados:

$$X^2 = \sum_{j=1}^{J} r_j^2 \tag{3.178}$$

Donde los r_j, fueron denominados por Hosmer “residuos de Pearson”, expresados de esta forma:

$$r_j = \frac{y_j - n_j \hat{p}_j}{\sqrt{n_j \hat{p}_j \left(1 - \hat{p}_j\right)}} \tag{3.179}$$

3.2.5.3.2. Test basados en probabilidades estimadas.

Los importantes trabajos de Hosmer y Lemeshow (1980) concluyeron con la aportación de una serie de tests estadísticos para medir la bondad del ajuste basándose en la agrupación de las probabilidades estimadas por el modelo. Fueron fundamentalmente dos propuestas de estadísticos que llamaron C_g y H_g.

La construcción del estadístico C_g se basa en la agrupación de probabilidades estimadas bajo el modelo de regresión $\hat{p}_1, \hat{p}_2 ... \hat{p}_N$ y formar G grupos (normalmente 10, denominados deciles de riesgo), calculándose, para estos grupos, las frecuencias esperadas.

Tabla 3.11. Frecuencias esperadas y observadas para C_g.

	Respuesta			
	$Y=1$		$Y=0$	
Grupos	Observado	Esperado	Observado	Esperado
$\hat{p}_j < d_1$	o_{11}	e_{11}	o_{01}	e_{01}
$d_1 \le \hat{p}_j < d_2$	o_{12}	e_{12}	o_{02}	e_{02}
…	…	…	…	…
$d_{G-1} \le \hat{p}_j < d_G$	o_{1G}	e_{1G}	o_{0G}	e_{0G}
Total	o_1	e_1	o_0	e_0

El número de frecuencias para las que ocurrió el suceso y el número esperado de instancias para las que ocurrirá, y para los que no, se calculan por las siguientes fórmulas:

$$o_{1g}=\sum_{k=1}^{n_g} y_k \text{ , } o_{0g}=\sum_{k=1}^{n_g}(1-y_k) \text{ , } e_{1g}=\sum_{k=1}^{n_g}\hat{p}(x_k) \text{ y } e_{0g}=\sum_{k=1}^{n_g}(1-\hat{p}(x_k)) \quad (3.180)$$

El estadístico C_g se obtiene a través de la siguiente expresión:

$$C_g=\sum_{k=0}^{1}\sum_{g=1}^{G}\frac{(o_{kg}-e_{kg})^2}{e_{kg}} \quad (3.181)$$

Este estadístico sigue una distribución asintótica χ^2 con G -2 grados de libertad.

Dado que la construcción del cálculo de los grupos depende de los puntos de corte, se puede originar algo de confusión e inestabilidad, porque dependiendo de estos puntos, diversos programas de software estadístico obtenían diferentes resultados.

Estos investigadores, Hosmer-Lemeshow, propusieron otro test estadístico cuyos puntos de corte eran ahora fijos y estaban preestablecidos y que llamaron estadístico H_g. Si bien el número de grupos puede ser arbitrario los autores recomendaron que se utilizaran 10.

Tabla 3.12. Frecuencias esperadas y observadas para H_g .

	Respuesta			
	$Y=1$		$Y=0$	
Grupos	Observado	Esperado	Observado	Esperado
$0 \le \hat{p}_j < 0{,}1$	o'_{11}	e'_{11}	o'_{01}	e'_{01}
$0{,}1 \le \hat{p}_j < 0{,}2$	o'_{12}	e'_{12}	o'_{02}	e'_{02}
…	…	…	…	…
$0{,}9 \le \hat{p}_j < 1{,}0$	o'_{110}	e'_{110}	o'_{010}	e'_{010}
Total	o'_1	e'_1	o_0	e'_0

El estadístico H_g sigue la misma distribución que el estadístico C_g con los mismos grados de libertad.

$$H_g = \sum_{k=0}^{1} \sum_{g=1}^{10} \frac{\left(o'_{kg} - e'_{kg}\right)^2}{e'_{kg}} \tag{3.182}$$

Aunque los primeros resultados apuntaban a que H_g parecía más potente que C_g, posteriormente Hosmer y Lemeshow (1989) señalaron como más adecuado para su uso a C_g. En otro trabajo publicado por Hosmer *et al.* (1997), se recomendaba el empleo de estos estadísticos después de utilizar otros.

3.2.5.3.3. Test basados en residuos suavizados.

Los diferentes test que emplean los residuos suavizados utilizan técnicas de regresión no paramétrica que comparan con el valor estimado de las probabilidades del modelo de regresión logística.

Los trabajos pioneros fueron los elaborados por Copas (1983) que utiliza métodos no paramétricos tipo núcleo para representar la respuesta observada y suavizada frente a la covariable. Un año después, Landwehr *et al.* (1984) utilizan análisis cluster de vecinos próximos y diseñan un método gráfico para verificar la bondad del ajuste. Otros autores como Fowlkes (1987) y Azaalini y Härdle (1989) también utilizaron técnicas de suavizado para calcular la bondad del ajuste del modelo.

Todos los procedimientos propuestos por los autores señalados fueron formulados para variables de tipo continuo.

Otro test, que no depende de los patrones de las covariables, es el propuesto por Le Cessie y Howelingen (1991) que denominaron $\hat{T}_{lc}$. En este contraste, la función suavizada se obtiene a través de las funciones tipo kernel propuestos por Nadayara (1964) y Watson (1964) como una suma ponderada de los residuos de Pearson:

$$\tilde{r}(x_j) = \frac{\sum_{k=1}^{N} r(x_k) K\left(\frac{x_j - x_k}{h_N}\right)}{\sum_{k=1}^{N} K\left(\frac{x_j - x_k}{h_N}\right)} \tag{3.183}$$

En esta fórmula N representa al tamaño de la muestra y h_N es la ventana que controla el suavizado. K es la función núcleo acotada, simétrica, no negativa y normalizada. Los residuos de Pearson se obtienen de la forma siguiente:

$$r_e(x_j) = \frac{y_j - \hat{p}_j}{\sqrt{\hat{p}_j(1-\hat{p}_j)}} \tag{3.184}$$

La fórmula para medir el ajuste del modelo es la suma ponderada de los residuos suavizados:

$$\hat{T}_{lc} = N^{-1} \sum_{j=1}^{N} \tilde{r}(x_j)^2 Var\left(\tilde{r}(x_j)\right)^{-1} \tag{3.185}$$

Donde

$$Var\left(\tilde{r}(x_j)\right) = \frac{\sum_{k=1}^{N}\left(K\left(\frac{x_j - x_k}{h_N}\right)\right)^2}{\left[\sum_{k=1}^{N}\left(K\left(\frac{x_j - x_k}{h_N}\right)\right)\right]^2} \tag{3.186}$$

Obtenemos $\hat{T}_{lc} \sim c\chi^2 V$, con $c = \frac{Var(\hat{T}_{lc})}{2E(\hat{T}_{lc})}$ y $V = \frac{2E(\hat{T}_{lc})^2}{Var(\hat{T}_{lc})}$ siendo $E(\hat{T}_{lc}) = 1$ y

$$Var(\hat{T}_{lc}) = N^{-2} \sum_{j=1}^{N}\sum_{k=1}^{N}\left(\sum_{l=1}^{N} w_{jl}^2 \sum_{l=1}^{N} w_{kl}^2\right)^{-1}\left(\sum_{l=1}^{N} \frac{w_{jl}^2 w_{kl}^2 (6p_l^2 - 6p_l + 1)}{p_l(1-p_l)} + 2\left(\sum_{l=1}^{N} w_{jl} w_{kl}\right)^2\right) \tag{3.187}$$

Calculando w_{kl} con la siguiente fórmula:

$$w_{kl} = K\left(\frac{x_k - x_1}{h_N}\right) \tag{3.188}$$

3.2.5.3.4. Medidas tipo R^2.

Se dispone también de medidas similares a la regresión lineal. Una de las primeras fue propuesta por Gordon *et al.* (1979) como un promedio de la proporción de la variabilidad explicada (AVPE) que calcula la proporción media de la varianza de la probabilidad de un suceso:

$$AVPE = \frac{\text{var}\,ianza\,incondicional\,de\,y - media\,\text{var}\,ianza\,condicional\,de\,y}{\text{var}\,ianza\,incondicional\,de\,y} \quad (3.189)$$

La varianza incondicional de la variable es igual a:

$E\left[\left(y_j - p_j\right)^2 \mid x_j\right] = p_j\left(1 - p_j\right)$ y la varianza incondicional es $\overline{pq}$, con $\overline{q} = 1 - \overline{p}$ y $\overline{p} = E\left[p_j\right] = \sum_j \frac{p_j}{N}$

Sustituyendo estas igualdades obtenemos la fórmula de la medida de la bondad del ajuste:

$$AVPE = \frac{\overline{pq} - \sum_j \frac{p_j\left(1 - p_j\right)}{N}}{\overline{pq}} \quad (3.190)$$

La expresión anterior no es muy utilizada ya que el denominador puede ser en alguna ocasión cero y esta medida no está acotada superiormente.

Otra forma de computar el ajuste del modelo a través de la verosimilitud fue propuesta por varios autores: Cox y Snell (1989) y Maddala (1983) que trataron de generalizar el concepto de R^2 de los modelos de regresión lineal:

$$R_g^2 = 1 - \left(\frac{\hat{L}_c}{\hat{L}_0}\right)^{\frac{2}{n}} \quad (3.191)$$

donde $\hat{L}_C$ representa la logverosimilitud del modelo evaluado con todas las variables explicativas y $\hat{L}_0$ es la logverosimilitud del modelo cuando sólo se incluye la constante.

Posteriormente, Nagelkerke (1991) ajusta el valor de la fórmula anterior para que el máximo se iguale a la unidad:

$$\overline{R}_g^2 = \frac{R_g^2}{\max\left(R_g^2\right)} \tag{3.192}$$

donde

$$\max\left(R_g^2\right) = 1 - \left(\hat{L}_0\right)^{\frac{2}{N}} \tag{3.193}$$

3.2.5.4. Estrategias de selección de modelos.

En los modelos de regresión logística se pueden llevar a cabo diferentes estrategias de selección de variables con el objetivo de reducir el número de variables explicativas y obtener modelos más parsimoniosos.

Dado que la estimación completa de todos los modelos posibles resulta computacionalmente costosa, se recurre a estrategias de modelización destinadas a encontrar el mejor subconjunto de variables predictoras.

Las estrategias más extendidas que se pueden llevar a cabo son las siguientes;

a) Selección de las variables significativas hacia adelante, en la cual en cada etapa se añade la mejor variable clasificadora que aún no ha sido seleccionada.
b) Eliminación de variables hacia atrás. En esta estrategia se parte del conjunto completo de variables independientes y se va eliminando en cada etapa la peor variable predictora hasta que las variables que quedan en el modelo son todas ellas significativas.
c) Un procedimiento intermedio es la modelización paso a paso, en la cual se combinan las dos estrategias anteriores.

Hay que tener en cuenta que en la selección hacia adelante las variables explicativas que son incluidas en el modelo no pueden ser posteriormente eliminadas del mismo. También. en la estrategia de eliminación hacia atrás una variable que ha sido eliminada del modelo no puede volver más tarde a ser incluida en la modelización. Estos inconvenientes fueron resueltos en el proceso de construcción del modelo paso a paso, donde las variables predictoras incluidas en el modelo en una determinada etapa pueden ser excluidas del mismo, al igual que una variable excluida del modelo puede ulteriormente ser incluida.

3.2.6. Métodos Bayesianos.

3.2.6.1. Introducción.

Como seres humanos nos enfrentamos muchas veces a la incertidumbre. Los métodos y técnicas bayesianas incorporan y cuantifican esta incertidumbre añadiendo la teoría de la probabilidad. De forma coloquial podemos definir a estos procedimientos como una representación gráfica para manejar la incertidumbre en sistemas expertos.

Los métodos bayesianos actualmente se pueden considerar que son construcciones sencillas, con una semántica clara y que poseen un enfoque sólido y elegante. El problema que encuentran algunos autores es su elevado coste computacional.

Los modelos bayesianos sirven tanto para resolver problemas desde una perspectiva descriptiva como predictiva. Como método descriptivo se centra en descubrir las relaciones de dependencia/independencia. Desde esta óptica se puede afirmar que a veces complementan y/o incluso superan a las reglas de asociación. En cuanto a la función predictiva se circunscribe a las técnicas bayesianas como métodos de clasificación.

Michell (1997) nos sugiere dos razones por las que los métodos bayesianos son algunas de las técnicas que más se han utilizado en los problemas de inteligencia artificial, el aprendizaje automático y la minería de datos:

1. Constituyen un método muy válido y práctico para realizar inferencias con los datos que disponemos, lo que implica inducir modelos probabilísticos que, una vez calculados, se pueden utilizar con otras técnicas de minería de datos.
2. Son extremadamente útiles en la comprensión de otras técnicas de inteligencia artificial y minería de datos que no trabajan con las probabilidades de las que nos dotan las técnicas bayesianas. Esta combinación de métodos es muy provechosa para optimizar las soluciones de algunos problemas planteados en la minería de datos.

3.2.6.2. Teorema de Bayes e hipótesis MAP.

Para comprender estas técnicas bayesianas vamos a empezar con el teorema de Bayes. Definamos las siguientes expresiones:

P(h) es la probabilidad a priori de que se cumpla la hipótesis h. Esta probabilidad contiene el conocimiento que tenemos de que la hipótesis h es correcta.

P(h/D) es la probabilidad a posteriori de que se cumpla la hipótesis h una vez conocidos los datos D. Esta expresión refleja la influencia que tienen los datos observados sobre la hipótesis h.

P(D/h) es la probabilidad de que los datos D sean observados en un escenario en el caso de que la hipótesis h sea correcta.

Sabemos que:

$$P(h \cap D) = P(h) * P(D/h) \tag{3.194}$$

$$P(h \cap D) = P(D) * P(h/D) \tag{3.195}$$

Es decir:

$$P(h) * P(D/h) = P(D) * P(h/D) \tag{3.196}$$

Por lo tanto:

$$\underset{\text{a posteriori}}{P(h/D)} = \underset{\text{a priori}}{P(h)} * \underset{\text{Factor de corrección}}{\frac{P(D/h)}{P(D)}} \tag{3.197}$$

Observando la expresión del teorema de Bayes sabemos que P(h/D) aumenta si se incrementa P(h) y P(D/h) o disminuye P(D).

Como ya disponemos de la fórmula adecuada que nos da la probabilidad a posteriori, estamos interesados ahora en obtener aquella hipótesis más probable o hipótesis MAP (maximum a posteriori), observados los datos.

La expresión anterior la podemos escribir ahora como:

$$h_{MAP} = \arg\max_h P(h/D) = \arg\max_h \left[P(h) * P(D/h) / P(D)\right] \tag{3.198}$$

Y al ser P(D) la misma en todas las hipótesis, la obtención del máximo se calcula prescindiendo de este término:

$$h_{MAP} = \arg\max_h P(h) * P(D/h) \tag{3.199}$$

h_{MAP} es la hipótesis más probable, dados los datos observados, P(h/D).

En los problemas de clasificación disponemos de una variable clase (C) y un conjunto de variables predictoras o atributos que denominaremos A_1, A_2…A_n. Con estas especificaciones el teorema de Bayes tiene la siguiente expresión:

$$P(C/A_1, A_2, ...A_n) = \frac{P(C)P(A_1, A_2, ...A_n/C)}{P(A_1, A_2, ...A_n)} \tag{3.200}$$

En los procedimientos bayesianos la hipótesis más plausible es aquella que tiene la máxima probabilidad a posteriori dados los atributos (hipótesis MAP), cuya expresión es la siguiente:

$$\begin{aligned} c_{MAP} &= \arg\max_{C\in\Omega_C} P(c/A_1, A_2, ...A_n) = \arg\max_{C\in\Omega_C} \frac{P(c)P(A_1, A_2, ...A_n/c)}{P(A_1, A_2, ...A_n)} = \\ &= \arg\max_{C\in\Omega_C} P(c)P(A_1, A_2, ...A_n/c) \end{aligned} \tag{3.201}$$

Donde Ω_C representa el conjunto de valores que puede tomar la variable C.

En el último paso se ha eliminado el denominador debido a que sería el mismo para todas las categorías de la variable C.

Este método sencillo y claro posee un problema que es la complejidad computacional debido a que necesitamos trabajar con distribuciones de probabilidad que involucran muchas variables, lo que resulta, en la mayoría de los casos inmanejable.

3.2.6.3. Clasificador Naïve Bayes.

El desarrollo de este famoso clasificador, incluido en la gran mayoría de paquetes informáticos, se encuentra en Duda y Hart (1973). Este método parte de la suposición de que todos los atributos son independientes conocido el valor de la variable clase:

$$I(X_i X_j \mid C), \forall i, j \tag{3.202}$$

La factorización de la función de probabilidad conjunta de este modelo es de la siguiente forma:

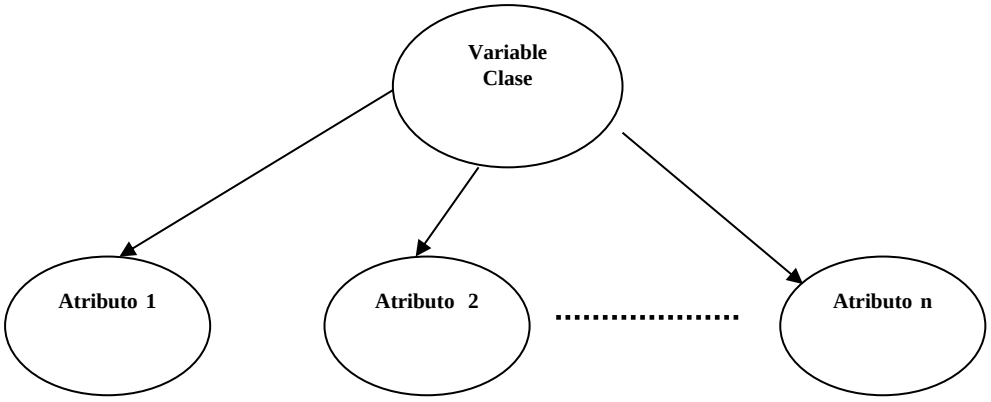
$$p(X_1, X_2 \ldots X_n, c) = P(C)\prod_{i=1}^{n} P(X_i \mid C) \qquad (3.203)$$

Este supuesto es poco realista en la mayoría de los casos pero, aun así, es uno de los más competitivos comparado con otras técnicas como las redes neuronales o los árboles de clasificación, Tsagalidis *et al.* (2008).

La estimación de los parámetros en este método, es decir, la clase o valor a devolver será la resultante de aplicar la siguiente fórmula:

$$c_{MAP} = \arg\max_{C \in \Omega_C} P(c)P(A_1, A_2, \ldots . A_n / c) = \arg\max_{C \in \Omega_C} P(c)\prod_{i=1} P(A_i / c) \qquad (3.204)$$

Figura 3.35. Esquema de representación de naïve-Bayes.



Dados los datos de entrenamiento se recorren todos esos datos y se computa la clasificación de cada uno de ellos, obteniendo $P(C_j)$ para cada clasificación posible.

Cuando los atributos son discretos, la estimación de la probabilidad condicional se extrae de la base de datos ya que son las frecuencias de aparición. Si $n(x_i, Pa(x_i))$ representa al número de registros de nuestra base de datos en el que la variable Xi toma el valor xi y a los padres de Xi lo denotamos por $Pa(x_i)$; entonces la fórmula de la probabilidad condicional viene determinada por el cociente entre el número de casos favorables y el de casos posibles:

$$P(x_i / Pa(x_i)) = \frac{n(x_i, Pa(x_i))}{n(Pa(x_i))} \qquad (3.205)$$

Cuando las muestras son pequeñas o si se realizan muestreos en el que los cruces de dimensiones son frecuentes es muy probable que los resultados obtenidos sean muy dudosos. Para atenuar este problema existen procedimientos de estimadores basados en suavizados. Uno de los más conocidos es el estimador basado en la sucesión de Laplace, que viene definido por la siguiente fórmula:

$$P(x_i / Pa(x_i)) = \frac{n(x_i, Pa(x_i)) + 1}{n(Pa(x_i)) + |alt|} \tag{3.206}$$

Ahora la estimación de la probabilidad viene expresada por el número de casos favorables + 1 dividida por el de casos totales más el número de posibilidades o alternativas.

Esta estimación asume una distribución a priori uniforme y no puede ajustarse a nuestras necesidades si es que queremos suavizar más o menos la probabilidad. Existe otra forma de resolver el cálculo de la probabilidad que es a través del m-estimador, que no es más que una generalización de la corrección de Laplace. Su expresión matemática viene dada por:

$$P(x_i / Pa(x_i)) = \frac{n(x_i, Pa(x_i)) + mf_{\Pr iori}(C)}{n(Pa(x_i)) + m} \tag{3.207}$$

Ahora el numerador son los casos favorables más una constante m multiplicada por la frecuencia de aparición a priori del evento y, el denominador es el número de casos totales más la constante m.

Cuando los datos son continuos el estimador naïve-Bayes supone que la distribución de esta variable continua sigue una distribución normal. La media aritmética y la desviación típica que caracterizan a esta distribución gaussiana se estiman a través de los datos muestrales.

$$P(A_i / c) \propto N(\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2} \right] \tag{3.208}$$

Cuando las variables continuas no siguen una distribución de probabilidad normal las estimaciones a través de este método pueden ser muy deficientes pero, en estos supuestos, se pueden aproximar a través de métodos kernel o, también, realizar la

transformación de las variables cuantitativas en otras de intervalos con lo que se pueden obtener mejores resultados.

Existen diferentes contribuciones publicadas en la literatura para mejorar este método que se pueden agrupar en dos grupos: clasificadores ingenuo extendidos y otros, más publicados más recientemente dedicados a los clasificadores ingenuos jerárquicos.

Según diversos autores, Pearl (1988), Castillo *et al.* (1997), Jensen (2001) y Cowel (2001), este método de clasificación, que en muchas aplicaciones prácticas obtiene excelentes resultados, no alcanza a considerar de forma adecuada la semántica intrínseca de las redes bayesianas.

3.2.6.4. Redes bayesianas.

Las redes bayesianas se conocen en la literatura existente con otros nombres como redes causales o redes causales probabilísticas, redes de creencia, sistemas probabilísticas, sistemas expertos bayesianos o también como diagramas de influencia. Las redes bayesianas son métodos estadísticos que representan la incertidumbre a través de las relaciones de independencia condicional que se establecen entre ellas (Edwards, 1998). Este tipo de redes codifica la incertidumbre asociada a cada variable por medio de probabilidades. Siguiendo a Kadie, Hovel y Horvitz (2001) afirman que una red bayesiana es un conjunto de variables, una estructura gráfica conectada a estas variables y un conjunto de distribuciones de probabilidad.

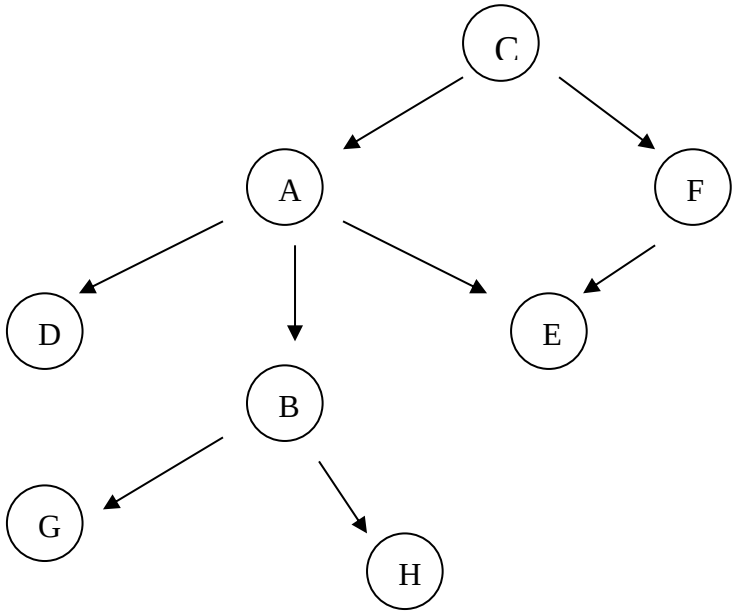
3.2.6.4.1. Definición formal.

Las redes bayesianas probabilísticas automatizan el proceso de modelización utilizando toda la expresividad de los grafos para representar las dependencias e independencias a través de la teoría de la probabilidad para cuantificar esas relaciones. En esta unión se realiza de forma eficiente tanto el aprendizaje automático como la inferencia con los datos y la información disponible

Una red bayesiana queda especificada formalmente por una dupla $B=(G,\Theta)$ donde G es un grafo dirigido acíclico (GDA) y Θ es el conjunto de distribuciones de probabilidad. Definimos un grafo como un par G= (V, E), donde V es un conjunto finito de vértices nodos o variables y E es un subconjunto del producto cartesiano V x V de pares ordenados de nodos que llamamos enlaces o aristas.

El grafo es dirigido y acíclico. Dirigido porque los enlaces entre los vértices de la estructura están orientados, por ejemplo si (A,B) € E pero (B,A) no € E diremos que hay un enlace o un arco entre los nodos y lo representamos como A→B. Cuando se dice que es acíclico es porque no pueden existir ciclos o bucles en el grafo, lo que significa que si empezamos a recorrer un camino desde un nodo no se puede regresar al punto de partida.

Figura 3.36. Topología de una red bayesiana



Las conexiones del tipo A→B indican dependencia o relevancia directa entre las variables, en este caso se indica que B depende de A o que A es la causa de B y B el efecto de A. También se dice que A es el padre y B el hijo. La ausencia de arcos entre los nodos nos está aportando una valiosa información ya que en este caso el grafo nos informa de independencia condicional.

La redes bayesianas tienen la habilidad de codificar la causalidad entre las variables por lo que han sido muy utilizadas en el modelado o en la búsqueda automática de estructuras causales (López, García y De la fuente; 2006). La potencia de las redes bayesianas está en su capacidad de codificar las dependencias/independencias relevantes considerando no sólo las dependencias marginales sino también las dependencias condicionales entre conjuntos de variables

Los grafos definen un modelo probabilístico con las mismas dependencias utilizando una factorización mediante el producto de varias funciones de probabilidad condicionada:

$$p(x_i, x_2, \ldots\ldots x_n) = \prod_{i=1}^{n} p\left(x_i \middle| padres(x_i) \right) \quad (3.209)$$

$padres(x_i)$ son las variables predecesoras inmediatas de la variable X_i en la red, precisamente $p(x_i \middle| padres(x_i))$ son los valores que se almacenan en el nodo que precede a la variable x_i

A través de la factorización las independencias del grafo son traducidas al modelo probabilístico de forma muy práctica.

Las redes bayesianas representan el conocimiento cualitativo del modelo mediante el grafo dirigido acíclico. Esta representación del conocimiento está articulada en la definición de la relaciones de dependencia/independencia. Al utilizar la representación gráfica a través del grafo hace que las redes bayesianas sean una herramienta muy poderosa y atractiva como representación del conocimiento.

3.2.6.4.2. Independencia condicional.

La estructura o topología de la red de la red bayesiana no sólo representa las dependencias entre las variables sino que describe además las independencias condicionales existentes entre ellas.

Se dice que una variables X es condicionalmente independiente de otra variable Y dada un una tercera Z, si el hecho de conocer Z hace que X e Y sean independientes. Es decir que si conozco Z, Y no tiene influencia en X.

$$P(X|Y,Z)=P(X|Z) \quad (3.210)$$

Esta condición se traduce en una red bayesiana en que cada variable es independiente de todos aquellos nodos que no son sus descendientes.

Si enumeramos los nodos de la red bayesiana X_1, X_2,…..X_i de tal forma que los cualquier nodo aparezca antes que cualquiera de sus descendientes podemos afirmar que cada variable X_i es condicionalmente independiente de las variables del conjunto {

X_1, X_2,…..X_i } conocidos los valores de sus padres. Dicho de otro modo conociendo los padres de una variable ésta se vuelve independiente del resto de sus predecesores.

Pearl (1988) especifica que la probabilidad conjunta definida como:

$$P(X_1, X_2 ... X_n) = \prod_1^n P(X_1 \mid X_2 ... X_{t-1}) \tag{3.211}$$

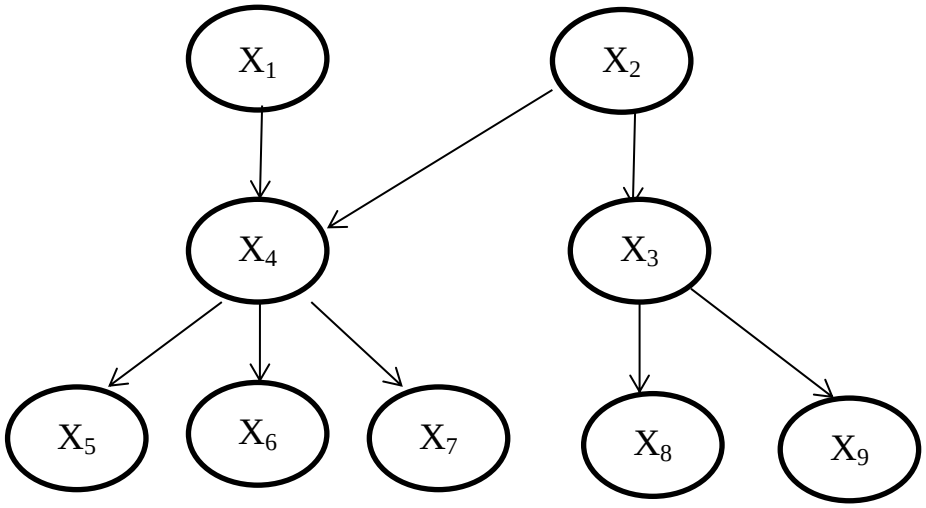
Podemos calcular la tabla de la probabilidad conjunta de todas las variables de una red bayesiana a partir de las tablas de probabilidad condicional de cada variable en función de sus padres:

$$P(X_1, X_2 ... X_n) = \prod_1^n P(X_1 \mid X_2 ... X_{t-1}) = \prod_1^n P(X_1 \mid Pa(X_1) \tag{3.212}$$

Par ilustrar este concepto tan importante vamos a calcular la probabilidad conjunta de todos los nodos que componen la siguiente red bayesiana de nueve variables definida la estructura por el siguiente diagrama:

Figura 3.37. Topología de una red con nueve parámetros.

,



Aplicando la regla de la cadena obtenemos la siguiente expresión:

$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9) = P(X_1)P(X_2|X_1)P(X_3|X_2,X_1)P(X_4|X_3,X_2,X_1)P(X_5|X_4,X_3,X_2,X_1)P(X_5|X_4,X_3,X_2,X_1)P(X_6|X_5,X_4,X_3,X_2,X_1)P(X_7|X_6,X_5,X_4,X_3,X_2,X_1)$
$P(X_8|X_7,X_6,X_5,X_4, X_3,X_2,X_1)\ P(X_9|\ X_8,X_7,X_6,X_5,X_4,\ X_3,X_2,X_1)$

Debido a que las probabilidades condicionales sólo están influenciadas por sus padres la expresión de la probabilidad conjunta de la red se reduce a la siguiente expresión:

$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9) = P(X_1)P(X_2)\ P(X_3|X_2)P(X_4|,X_2,X_1)\ P(X_5|X_4)\ P(X_6|X_4,\ P(X_7|,X_4)\ P(X_8|X_3)\ P(X_9|\ X_3)$

Como se puede observar el efecto de las probabilidades condicionales una vez dada la estructura de las relaciones entre las variables según la topología de la red se ha reducido considerablemente. La representación del conocimiento requiere ahora la estimación de muchos menos parámetros.

Un importante concepto para la construcción de una red bayesiana es el criterio ***d-separación,*** Jensen y Nielsen (2007). Se dice que dos variables distintas *A* y *B* en una red causal están d-separadas (d para grafos dirigidos) si para todos los caminos entre *A* y *B*, hay una variable intermedia *V* (distinta de *A* y *B*) tal que se cumple una de las dos proposiciones siguientes: la conexión es serial o divergente y *V* está instanciada o la conexión es convergente y ni *V* ni ninguno de sus descendientes ha recibido evidencia. Si *A* y *B* no están d -separadas se llaman d-conectadas.

Cuando *Z* d-separa *X* e *Y* en *G*, se escribe $I(X, Y \mid Z)_G$ para indicar que la relación de

independencia viene dada por el grafo *G*; en caso contrario, se escribe $D(X, Y \mid Z)_G$ para indicar que X e Y son condicionalmente dependientes dado Z en el grafo.

3.2.6.4.3. Inferencia. Propagación del conocimiento en la red bayesiana.

Una vez obtenida la red bayesiana cuando se disponen de nuevos datos, nueva información o *evidencia* necesitamos obtener nuevas conclusiones. A este proceso se le denomina inferencia probabilística o propagación del conocimiento a través de la red bayesiana.

La inferencia en redes bayesianas también conocida como actualización de creencias (belief updating) es el proceso de actualización de las probabilidades a posteriori en toda la estructura de la red, dado el conjunto de evidencias, Según se define en Kord y Nicholson (2004), en Zhang y Poole (1996) y también por otros autores se trata de un mecanismo para el cálculo de las distribuciones a posteriori de probabilidades para un conjunto de variables dado el conjunto de evidencias.

Si llamamos E a la evidencia presentada (lista de valores observados), y X es el conjunto de datos el cálculo de la probabilidad de los datos X dada la evidencia se calcula a través del teorema de Bayes:

$$P(X \mid E) = \frac{P(X)P(E \mid X)}{P(E)} \tag{3.213}$$

·El cálculo de estas probabilidades es computacionalmente intratable para problemas con muchas variables debido al elevado número de combinaciones de valores que están involucrados, sin embargo existen métodos eficientes de propagación de la evidencia. Cooper (1990) demostró que la imputación de la probabilidad a posteriori es un problema NP-difícil incluso para una variable, lo que ha originado una búsqueda de soluciones que han ido encaminadas en dos direcciones: métodos exactos y métodos aproximados. Un buen resumen de métodos eficientes que utilizan la estructura de dependencias del grafo se pueden consultar en Castillo *et al.* (1997).

Los métodos exactos calculan las probabilidades a posteriori de forma exacta, pero al ser un problema de complejidad exponencial sólo se utiliza este procedimiento cuando el tiempo de proceso es asumible o en algunas tipos especiales de redes bayesianas.

Algoritmos de propagación exacta han sido desarrollados mediante árboles de unión en Pearl (1988), Castillo *et al.* (1997), Jensen (2001), Schachter *et al.* (1994), Baldi y Soren (2001), El- Hay (2001), Jensen y Nielsen (2007), Kjærulff y Madsen (2008) y a través de propagación perezosa (lazy prop**a**gation) en Shenoy (1992) y en Madsen y Jensen (1999).

Los métodos aproximados, Salmerón (1998), se utilizan cuando la propagación exacta de probabilidades no puede llevarse a cabo en un tiempo razonable. Esto métodos pierden precisión en la solución que aportan a cambio de obtener resultados en tiempos más cortos. Generalmente estos métodos, para obtener valores aproximados de las probabilidades emplean distintas técnicas de simulación.

3.2.6.4.4. Aprendizaje en las Redes Bayesianas.

Ya se ha señalado que para obtener una red bayesiana se ha de especificar una estructura gráfica y una función de probabilidad conjunta que viene especificada por el producto de la probabilidades de cada nodo dados sus padres, lo que implica que en la mayoría de las ocasiones no se conocen ni la estructura ni las probabilidades. Esta

es la razón por la que se han desarrollado diferentes métodos de aprendizaje para obtener la red bayesiana dados los datos.

Las tareas de aprendizaje a las que se enfrentan los diferentes métodos se pueden dividir en un aprendizaje estructural y un aprendizaje paramétrico. En este sentido la mayoría de los autores afirman que las redes bayesianas tienen dos dimensiones: una cuantitativa y otra cualitativa: Cowell, *et al*., (1999); Garbolino y Taroni, (2002); Nadkarni y Shenoy, (2001, 2004); Martínez y Rodríguez, (2003)

Dimensión cuantitativa. Aprendizaje paramétrico.

No sólo estas redes bayesianas modelan cualitativamente las relaciones sino que también cuantifican y expresan de forma numérica la fuerza de esa relaciones entre las variables. Existen tres elementos que caracterizan la dimensión cuantitativa de la red bayesiana: el concepto de probabilidad, como medida del grado de creencia subjetiva relativa a un evento, un conjunto de funciones de probabilidad condicionada que definen a cada variable en el modelo y el teorema de Bayes que se utiliza para actualizar las probabilidades con base a la experiencia.

La fuerza de las relaciones entre las variables está especificada en las distribuciones de probabilidad como una medida de la creencia que tenemos sobre esas relaciones en el modelo.

El aprendizaje paramétrico consiste en hallar los parámetros asociados a la estructura de la red. Estos parámetros están constituidos por las probabilidades de los nodos raíz y de las probabilidades condicionales de las demás variables dados sus padres.

La probabilidades previas se obtienen se corresponden con la marginales de los nodos raíz y las condicionales se obtienen de las distribuciones de cada nodo con sus padres.

Dimensión cualitativa. Aprendizaje estructural.

En el aprendizaje estructural es donde se establecen las relaciones de dependencia que existen entre las variables del conjunto de datos para obtener el mejor grafo que represente estas relaciones. Este problema ya hemos señalado anteriormente es bastante complejo dado que la búsqueda de la estructura que nos represente mejor a los datos es un problema NP-completo lo que lo hace computacionalmente intratable cuando el número de variables es grande. Muchas veces se buscan algoritmos

eficientes que si bien no es lo óptimo, sí que se aproximan a la solución buscada con costes computacionales acotados, Neapolitan, (2003).

Básicamente se pueden englobar en dos tipos los métodos de aprendizaje de la estructura. Se encuentra por una parte aquellos métodos que utilizan métricas de complejidad-bondad de ajuste y algoritmos de búsqueda.

3.2.6.4.4.1. Métricas de evaluación.

La métrica define la calidad de la red bayesiana en función de los datos y el algoritmo de búsqueda tratará de encontrar la red que maximice esta métrica explorando todas las posibilidades. Téngase en cuenta que el número de posibles estructuras gráficas aumenta considerablemente con el número de variables, es un problema NP-duro, Maxwel (1996). Por ejemplo, existen 12 grafos para tres variables y se eleva a 543 si las variables son cuatro

Dependiendo de la métrica utilizada y la técnica de búsqueda existen un amplia gama de procedimientos que pueden ir desde métodos voraces simples Cooper y Herskovitz, (1992) hasta métodos que utilizan algoritmos genéticos Larrañaga *et al.,* (1996b).

Otros métodos están basados en test estadísticos para detectar las posibles dependencias/independencias presentes en los datos por lo que la red se ajustaría a estas dependencias descubiertas. Estos métodos parecen más eficientes pero pueden ser muy sensibles a los fallos en los test, especialmente cuando en el problema están involucradas muchas variables, Friedman et al*., (1999).*

También se pueden utilizar ambas estrategias para optimizar la búsqueda y construir el grafo, Campos, (2006)

Dado un conjunto de datos D=(X_1, $X_{2,...}X_n$) se define una métrica de evaluación como el procedimiento de encontrar el GDA (G^*) tal que verifique la siguiente expresión:

$$\varsigma^* = \arg máx_{\varsigma \in \varsigma^n} f(\varsigma : D) \qquad (3.214)$$

Donde

$$f(\varsigma : D) = \sum_{i=1}^{n} f_D(X_i, Pa_\varsigma(X_i))$$

$f(\varsigma : D)$ representa a la función de valuación que mide la calidad de un GDA candidato ς en relación al conjunto de datos y ς^n es el conjunto de todos los GDAs que pueden construirse con los n nodos de las variables

Para poder utilizar estas fórmulas estos métodos es necesario que la métrica sea descomponibles ante datos completos. Se dice que una métrica es descomponible, Nielsen y Jensen (2009), si la evaluación de un GDA dado es igual a la suma de los valores obtenidos de cada uno de sus nodos y sus correspondientes familias (nodo y sus padres). Esta propiedad se puedes expresar como:

$$f_D\left(X_{i,Pa\varsigma}(X_i)\right) = f_D\left(X_i Pa_\varsigma(X_i) : N_{xi,pa_\varsigma(X_i)}\right) \tag{3.215}$$

$N_{xi,pa_\varsigma(X_i)}$ son los estadísticos de las variables X_i y $Pa_\varsigma(X_i)$ calculadas sobre los datos D que se corresponden con cualquier configuración posible de $\{X_i\} \cup Pa_\varsigma(X_i)$

3.2.6.4.4.1.1. Métricas bayesianas.

Las métrica bayesianas buscan la estructura que maximiza la probabilidad de una red condicionada a la base de datos usando la fórmula de Bayes:

$$P(G \mid D) = \frac{P(D \mid G)P(G)}{P(D)} \tag{3.216}$$

P(G) es la distribución a priori de cada estructura candidata y P(D|G) se le conoce como evidencia ya que es la verosimilitud promedio que puede calcularse bajo ciertas suposiciones.

Se puede prescindir del denominador de la expresión de la fórmula de Bayes porque los datos son siempre los mismos para las distintas redes que podemos construir de un mismo problema. También se trabaja, habitualmente con los logaritmos de las fórmulas por ser más fácil a la hora de trabajar con las diferentes métricas. Por otra parte el término P(G) puede ignorarse si se utiliza una distribución uniforme

En los párrafos siguientes se van a describir las fórmulas de los principales métodos pero en primer lugar se concreta la notación común a todos ellas en la tabla 3.13.

En la métrica del procedimiento K2 de Cooper y Herskovitz, (1992) si se verifican un conjunto de condiciones tales como la independencia de los casos de la base de datos, que no existan casos perdidos o desconocidos y la uniformidad de las distribuciones de probabilidad de los parámetros de una red, podemos establecer cuál es la distribución de probabilidad conjunta de un GDA y una base de datos:

Tabla 3.13. Notación de las diferentes métricas de redes bayesianas.

Elemento	Descripción
N	Número de instancias en el conjunto de datos D
r_i	Número de estados de la variable aleatoria X_I
x_{ik}	k-ésimo valor de la variable X_I
$q_i = \prod_{x_j \in Pax_i} r_j$	Número de configuraciones posibles del conjunto de padres Pax_i de X_I
w_{ij}	j-ésima configuración de C siendo $1 \leq j \leq q_i$
N_{ijk}	Número de instancias del D en las que la variable X_I toma el k-ésimo valor x_{ik} y las variables en Pax_i toman su j-ésima configuración w_{ij}
$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$	Número de instancias en D en las que las variables en Pax_i toman su j-ésima configuración w_ij
$N_{ik} = \sum_{j=1}^{1_i} N_{ijk}$	Número de instancias en D en las que la variable X_I toma el k-ésimo valora x_{ik}.

$$f_{K2}(\varsigma : D) = \log(p(\varsigma)) + \sum_{i=1}^{n}\sum_{j=1}^{q_i}\left(\log\left(\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\right)\right) + \sum_{k=1}^{r_i}\log(N_{ijk}!) \qquad (3.217)$$

La métrica BD (Bayesian Dirichlet) es una generalización de la métrica que emplea el algoritmo de búsqueda K2 y fue propuesta por Heckerman *et al* (1994)

$$f_{BD}(\varsigma : D) = \log(p(\varsigma)) + \sum_{i=1}^{n}\sum_{j=1}^{q_i}\left(\log\left(\frac{\Gamma(\eta_{ij})}{\Gamma(N_{ij}+\eta_{ij})}\right) + \sum_{k=1}^{r_i}\log\left(\frac{\Gamma(N_{ijk}+\eta_{ijk})}{\Gamma(\eta_{ijk})}\right)\right) \qquad (3.218)$$

Donde η_{ijk} representan los hiperparámetros de la distribución a priori de Dirichlet dada la estructura de la red y $\Gamma(\ )$ es la distribución Gamma.

La especificación de los hiperparámetros η_{ijk} no es sencilla pero si realizamos la sunción de equivalencia en verosimilitud [150] el procedimiento se simplifica y la métrica así calculada recibe el nombre de métrica BDe, Heckerman *et al* (1994) y cuya

expresión es igual a la anterior pero ahora los valores se calculan a través de la siguiente expresión:

$$\eta_{ijk} = \eta \times p\left(x_{ik} w_{ij} \mid \varsigma_0\right) \tag{3.219}$$

Donde $p\left(x_{ik} w_{ij} \mid \varsigma_0\right)$ es una distribución de probabilidad asociada s la red bayesiana a priori y η es un parámetro que representa el tamaño muestral equivalente.

Otra variación muy interesante la encontramos en Buntine (1991) donde ahora la asignación de la probabilidad a priori de cada variable y sus padres se realiza a través de una distribución uniforme, lo que significa que $p\left(x_{ik} w_{ij} \mid \varsigma_0\right) = \frac{1}{r_i q_i}$. A esta métrica se la conoce con el nombre de BDeu y sólo depende del parámetro η. Su expresión matemática sustituyendo en la ecuación anterior es:

$$f_{BDeu}(\varsigma : D) = \log\left(p(\varsigma)\right) + \sum_{i=1}^{n}\sum_{j=1}^{q_i}\left(\log\left(\frac{\Gamma\left(\frac{\eta}{q_i}\right)}{\Gamma\left(N_{ij} + \frac{\eta}{q_i}\right)}\right) + \sum_{k=1}^{r_i}\log\left(\frac{\Gamma\left(N_{ijk} + \frac{\eta}{r_i q_i}\right)}{\Gamma\left(\frac{\eta}{r_i q_i}\right)}\right)\right) \tag{3.220}$$

3.2.6.4.4.1.2. Métricas basadas en la teoría de la información.

Las métricas basadas en la teoría de la información representan otra opción de evalúan la calidad de un GDA respecto del conjunto de datos en base a conceptos relativos al campo de la codificación y de la teoría de la información.

En la teoría de la información la codificación tiene como objetivo representar los mensajes utilizando el menor número de elementos posibles. En concreto el principio MDL [288] del inglés Minimun Descriptions Lenght trata de minimizar la longitud de la descripción del modelo y su capacidad para representar el conjunto de los datos,

En la redes neuronales modelos muy complejos serán aquellos donde los nodos estén densamente conectados y serán redes muy precisas, bastante ajustadas a los datos. El caso extremo sería un grafo completo. Pero hay que tener en cuenta que redes muy complejas pueden dificultar seriamente la comprensión del modelo, aumentar notablemente el tiempo de computación y producir un sobreajuste de los datos por lo que en realidad se buscan redes, más sencillas y menos precisas. El principio MDL

tratará de obtener un equilibrio entre la complejidad del modelo y la precisión frente a los datos.

La longitud de descripción mínima en las redes bayesianas incluye la longitud requerida para representar la red, además de la longitud necesaria para representar los datos dado el modelo L(D|G). Bouckaert (1995) presenta un cálculo de esta expresión a través de la negación del logaritmo de la función de verosimilitud de G dado D, definiendo esta métrica basada en el log_verosimilitud o LL (Log Likehood). Si estimamos los parámetros con la frecuencia relativa su formulación matemática se puede expresar de la siguiente manera:

$$f_{LL}(\varsigma : D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log\left(\frac{N_{ijk}}{N_{ij}}\right) \tag{3.221}$$

La fórmula anterior se puede interpretar como la cantidad de bit que son necesarios para representar a D. Se puede afirmar que cuanto mayor sea la log_verosimilitud mejor modela el grafo G la distribución en los datos D.

La longitud de la descripción de la red depende del número de parámetros libres de la factorización de la probabilidad conjunta. Es un término que se denomina complejidad de la red C(ς): Es este valor junto con una función de penalización f(N) la que permite definir las métrica más utilizadas en las aplicaciones reales.

$$C(\varsigma) = \sum_{i=1}^{n} (r_i - 1) q_i \tag{3.222}$$

Si la función de penalización es $f(N) = \frac{1}{2}\log(N)$ obtenemos la métrica MDL o métrica BIC (Bayesian Information Criterion) que propuso Gideon (1978)

$$f_{BIC}(\varsigma : D) = f_{LL}(\varsigma : D) - \frac{1}{2}\log(N) C(\varsigma) \tag{3.223}$$

Si $f(N) = 1$ entonces la métrica es la propuesta por Akaike (1974) denominada AIC (Akaike Information Criterion)

$$f_{AIC}(\varsigma : D) = f_{LL}(\varsigma : D) - C(\varsigma) \tag{3.224}$$

Otra propuesta de métrica más reciente que se puede utilizar es la MIT (Mutual Information Test) propuesta por De Campos (2006) y que tiene como expresión:

$$f_{MIT}(\varsigma : D) = \sum_{\substack{i=1 \\ Pax_i \neq \phi}}^{n} 2NI(X_i; Pax_i) - \sum_{j=1}^{q_i} \chi_{\alpha, l_{i\sigma_i^*(j)}} \tag{3.225}$$

Donde la primera parte de la ecuación incorpora la información mutua entre una variable y sus padres y lo que trata es de medir el grado de interacción entre las variables y, la segunda parte de la expresión es un término para penalizar la reden base a una serie de test de independencia basados en la χ^2.

3.2.6.4.4.2. Algoritmos de búsqueda y aprendizaje.

El considerable número de combinaciones que se producen a la hora de buscar la mejor estructura del modelo de red bayesiana ha originado que, antes de verse imposibilitado a dar una solución óptima, se considere ofrecer soluciones que satisfagan al usuario que las demanda. Esta imposibilidad de cubrir todo el espacio de búsqueda ha motivado que los algoritmos de tipo heurístico sean cada vez más empleados como valiosas herramientas donde los algoritmos exactos no son capaces de encontrarlas.

Existen multitud de algoritmos que se han desarrollado en las últimas décadas. Algunos ejemplos interesantes que se utilizan en esta tesis son los algoritmos de Búsqueda Tabú, el Enfriamiento Simulado y los Algoritmos Genéticos. Existen otros enfoques que también se muestran muy eficaces: GRASP (Greedy Randomized Adaptative Search Procedure), Algoritmos Meméticos, VNS (Búsqueda por Entornos Variables), Colonias de Hormigas, Estimación de Distribuciones (EDA), Programación por restricciones o la Búsqueda Dispersa, entre otras propuestas que se han realizado en este campo del conocimiento tan fecundo.

A continuación se describen someramente los principales algoritmos utilizados en esta tesis y que a su vez se encuentran disponibles en la mayor parte de paquetes estadísticos.

3.2.6.4.4.2.1. Algoritmo K2.

Este algoritmo basado en búsqueda y optimización de una métrica bayesiana es considerado como el predecesor y fuente de inspiración para las generaciones

posteriores. El algoritmo K2 realiza una búsqueda voraz muy eficaz para encontrar una red de calidad en un tiempo razonable (Cooper y Herskovitz, (1992). Para llevar a cabo la búsqueda previamente el algoritmo ordena los nodos (variables de entrada) de forma que los posibles padres de una variable aparecen en el orden antes que ella misma, lo que evita la generación de ciclos. Esta restricción es bastante fuerte pero al proporcionar un orden hace que el algoritmo sólo tenga que buscar los padres posibles entre las variables predecesoras.

Partiendo de que el conjunto vacío es el conjunto de padres para cada variable y siguiendo un orden establecido pasa a procesar cada variable y calcula la ganancia que se produce en la medida al introducir una variable como padre. El proceso se repite para cada nodo mientras el incremento de calidad supere un cierto umbral preestablecido. En el algoritmo también se puede limitar el número de padres de cada variable. Considerando la descomponibilidad de la red, la contribución que realiza cada variable a la calidad de la red viene dada por la siguiente expresión:

$$\sum_{j=i}^{r_i} \sum_{k=1}^{s_i} N_{ijk} \log \frac{N_{ijk}}{N_{ik}} \tag{3.226}$$

El pseudocódigo del algoritmo K2 para aprendizaje estructural de redes bayesianas requiere una muestra de datos de un conjunto previamente ordenado de nodos.

El pseudocódigo es el siguiente:

$D = \{X_1,...,X_n\}$ y un grafo disconexo B.
1: for i = 1 : n – 1 do
2: S = score (D , B);
3: for k = I + 1 : n do
4: M = score (D , B $\cup \{i \to k\}$);
5: if M > S then
S = M; B (i , k) = 1;
6: end if
7: end for
8: end for

Ensure: Un grafo dirigido acíclico B y su medida de calidad S.

3.2.6.4.4.2.2. Algoritmo B.

En este algoritmo se elimina el problema de la dependencia de la ordenación previa de los nodos lo que añade mayor complejidad, Buntine, (1991). Este algoritmo apareció

muy temprano y al igual que su predecesor utiliza un esquema voraz. El orden de complejidad computacional es mayor. Al igual que el K2 se inicia con padres vacíos y en cada etapa se añade aquel enlace que maximiza el incremento de calidad eliminando aquellos que producen ciclos. El proceso se detiene cuando la inclusión de un arco no representa ninguna ganancia o bien se obtiene una red completa.

El pseudocódigo del algoritmo B para el aprendizaje estructural de redes bayesianas es el siguiente:

Este algoritmo requiere una muestra de datos de un conjunto de nodos $D = \{X_1,...,X_n\}$ y un grafo disconexo B.

1: for i = 1 : n – 1 do
2: S = score (D , B);
3: for k = I : n do
4: M = score (D , B $\cup \{i \rightarrow k\}$);
5: if M > S $\wedge$ acyclic (B $\cup \{i \rightarrow k\}$) then
S = M; B (i , k) = 1;
6: end if
7: end for
8: end for

Ensure: Un grafo dirigido acíclico B y su medida de calidad S.

3.2.6.4.4.2.3. Algoritmo Hill Climbing.

El algoritmo Hill Climbing (HC), que se ha traducido como Ascensión de Colinas, en su forma básica realiza la selección del siguiente nodo a expandir de acuerdo con alguna medición heurística que permite estimar la distancia que queda por recorrer hasta la meta. Se trata de un procedimiento de búsqueda que parte de la solución *s* y realiza movimientos desde esa solución a otras soluciones vecinas según un definición de entorno. Se le denomina también mejora iterativa porque cada nuevo movimiento requiere que la solución sea mejor que la anterior. Una rama no tiene por qué ser explorada hasta agotarse, sino que el proceso de expansión terminará en el momento en que se encuentra un nodo sucesor que no mejora el estado actual.

Es un algoritmo local que utiliza el concepto de vecindad clásica y que parte de una solución inicial y a partir de esta se calcula el nuevo valor utilizando todas las soluciones vecinas a la solución actual y selecciona el vecino que mejor solución presenta, es decir, este algoritmo finaliza cuando no existe ningún vecino que pueda mejorar la solución vecina.

Una variante muy útil y muy empleada HC consiste en considerar todos los posibles movimientos a partir del estado actual y elegir el mejor de ellos como nuevo estado. A este método se denomina ascensión por la máxima pendiente o búsqueda del gradiente.

En este algoritmo, como todos los utilizados en la búsqueda de la estructura de la red bayesiana, es un método que aprovecha la descomponibilidad de las métricas cuando tienen que recalcular las modificaciones que se realizan en los nodos vecinos.

3.2.6.4.4.2.4. Simulated Annealing.

Este algoritmo Simulated Annealing (Recocido simulado o enfriamiento simulado), Kirkpatrick *et al.* (1983) y Cerny (1985) es una de los más antiguas metaheurísticas que ya contiene, como idea fundamental, una estrategia implícita para escapar de los mínimos locales. La idea de este algoritmo es permitir movimientos que conducen a soluciones de peor calidad pero que permitan escaparse de los mínimos locales. La probabilidad de aceptación de los movimientos va disminuyendo durante la búsqueda y va a depender de la solución inicial, *f(s)* y de un parámetro de control que modula que proporción de malas soluciones que se aceptan.

El algoritmo comienza generando una solución inicial. Esta primera solución puede ser aleatoria o creada a través de procedimientos heurísticos. También se inicializa un parámetro de temperatura T. En cada solución se escoge una solución *s´* del entorno N(s) aceptándose al solución en función de su valor objetivo, *f*(*s´),* el valor objetivo de s, *f(s)* y la temperatura. Si el valor objetivo de *s´* es mejor entonces se reemplaza. La forma de elegir este enfriamiento es crucial en este algoritmo.

El pseudocódigo del algoritmo Simulated Annealing es el siguiente:

Procedure *RecocidoSimulado()*

s ⟵ GenerarSoluciónInicial()

T ⟵ T0

While no se cumplan las condiciones de parada do

s' ⟵ EscogerAleatoriamente(N(s))

if f(s') < f(s) then

s ⟵ s'

else

Aceptar s' como nueva solución con probabilidad p(T,s',s) end if

Actualizar (T)

End while

3.2.6.4.4.2.5. Tabu Search.

La metaheurística Tabú Search (Búsqueda Tabú) está desarrollada en Glover (1986), Bouckaert (1995) y en Glover y Laguna (1997). Ese algoritmo se estructura en tres fases: preliminar, intensificación y diversificación.

En la fase de búsqueda preliminar, partiendo de la solución inicial *s* y evalúa todas las soluciones del entorno de *s, N(s*) encontrando un punto del espacio *s'* mejor que *s* o incluso peor que *s*. Esta posibilidad puede generar bucles ya que en los siguientes movimientos puede retroceder de *s'* a *s* generando ciclos. Para evitar estos ciclos el algoritmo crea una lista de movimientos prohibidos de longitud *l*. En esta lista el primer el elemento en entrar es el primero en salir, lo que puede interpretar como una memoria a corto plazo. El movimiento de *s'* a *s* queda prohibido durante los siguientes *l* movimientos.

La fase de intensificación empieza con la mejor solución encontrada limpiando la lista tabú y procediendo como en la primera fase. En la fase de diversificación vuelve a limpiarse la lista tabú y se colocan en la lista los movimientos realizados hasta el momento con mayor frecuencia. La segunda fase actúa como una lupa en las regiones más prometedoras que se encontraron en la primera fase mientras que la fase de diversificación impulsa al algoritmo a explorar regiones novedosas.

El pseudocódigo del algoritmo Tabú Search es como sigue:

Procedure *TabuSearch()*

s ⟵ GenerarSoluciónInicial()

InicializarLIstasTabú(TL_1,…,TL_r)

k ⟵ 0

while no se cumplan las condiciones de parada do

ConjuntoPermitido(s,k)⟵ { s' € N(s) | s' no viola una condición tabú o satisface algún criterio de aspiración }

s ⟵ ElegirMejor(ConjuntoPermitido(s,k))

ActualizarListaTabúYCriteriosAspiración()

k ⟵ k+1

end while

Algunas extensiones de este algoritmo es considerar la lista tabú de diversa maneras para beneficiarse de la historia de la búsqueda. Por ejemplo se han considerado memoria de soluciones élite donde sólo están las mejores, la novedad en los atributos, el número de veces que ha sido visitada una solución, la calidad o la influencia (aquellas decisiones que se muestran más importantes)

Otra solución importante desarrollada sobre este algoritmo con la idea de escapar de mínimos locales es el algoritmo denominado Búsqueda Tabú Reactive (Reactive Tabu Search), Battiti (1996). Este autor considera que la historia de la búsqueda se emplee como una guía para ajustar los parámetros de la heurística para poder detectar mínimos locales y poder actuar reajustando los parámetros de la búsqueda para poder solventarlos.

3.2.6.4.4.2.6. Algoritmos basados en test de independencia.

Los test de independencia comprueban de forma eficiente las distintas dependencias combinando las variables, aunque se suelen considerar conjuntos reducidos de las variables si el número de variables es considerable.

La idea de estos métodos es satisfacer el mayor número de independencias presentes en los datos, Neapolitan. (2004).

Cuando estas dependencias se han detectado se expresan en el grafo. Al referirnos a las dependencias e independencias de variables nos referimos a conjuntos de variables. Sean por ejemplo tres conjuntos disjuntos de variables X, Y y Z. Se dice que

X es condicionalmente dependiente de Y dado Z y lo expresamos como I(X,Y/Z) si y sólo si se cumple la siguiente igualdad:

$$p(x/z,y)=p(x/z)\Leftrightarrow p(x,y/z)=p(x/z)\,p(y/z) \qquad (3.227)$$

para todos los valores posibles x, y, z. En cualquier otro caso X e Y se dicen condicionalmente dependientes dado Z, y lo expresamos como D(X,Y/Z). La independencia condicional lleva implícita la idea de que una vez conocida Z el conocimiento de Y no altera la probabilidad de X, en otras palabras se afirma que el conocimiento de Y no añade información alguna sobre X

Uno de los algoritmos más populares basados en test de independencia es el algoritmo PC. Este algoritmo utiliza una medida denominada información mutua. El concepto de información mutua se deriva de la entropía y proporciona un excelente criterio de medida de la independencia a partir de la información que una variable tiene sobre otra. La expresión matemática de la información mutua de dos variables es la siguiente:

$$\mathrm{MI(X,Y/Z)}=\sum_{z}p(z)\sum_{x,y}p(x,y/z)\log\frac{p(x,y/z)}{p(x/z)\,p(y/z)} \qquad (3.228)$$

Las probabilidades se estiman a partir de las frecuencias relativas que se observan en el conjunto de datos y son los estimadores de máxima verosimilitud de la probabilidad.

Una vez que se ha establecido esta medida podemos comprobar la hipótesis:

$\mathrm{I(X,}Y/Z)$ a través del estadístico:

$$G = 2\,N\;\mathrm{MI(X,}Y/Z) \qquad (3.229)$$

En esta fórmula N representa el tamaño de la muestra y el estadístico G, bajo la hipótesis de independencia se distribuye como una distribución χ^2 con grados de libertad igual a $(r_x - 1)(r_y - 1)r_z$ donde r_x , r_y y r_z representan la cardinalidad de cada variable.

En el aprendizaje paramétrico se calculan los parámetros de acuerdo con la estructura del tipo de grafo aprendido en la fase estructural y la base de datos de la que disponemos: tablas de probabilidad, medias, varianzas, etcétera.

En la literatura estadística existen diferentes propuestas para detector aquellas redes que mejor representan a los datos (concepto de bondad de ajuste) y que al mismo tiempo penalizan las estructuras más complejas (concepto de penalización). A cada red se le asigna una medida de calidad que es función de la probabilidad a posteriori. Esta distribución de probabilidad a posteriori p(B/D) se expresa por la siguiente fórmula:

$$p(B \Big/ D) = p(M, \theta \Big/ D) = \frac{p(M, \theta, D)}{p(D)} \propto p(M)\, p(\theta / M)\, p(D / M, \theta) \quad (3.230)$$

Donde B = (M, θ) es la red con el grafo dirigido M y θ representa a los parámetros.

Si las redes bayesianas son multinomiales y se asumen ciertas hipótesis de las distribuciones a priori de los parámetros y una probabilidad inicial uniforme para todos los modelos se puede encontrar una medida de calidad bayesiana que se basa en el logaritmo de la verosimilitud y que incluye un término de penalización, Heckerman, (1996):

Otra medida de calidad para redes multinomiales descrita en Lam y Bacchus (1994) es la denominada MDL (Minimum Description Lenght) donde se evalúan a la vez la verosimilitud de los datos de entrenamiento D y la simplicidad, maximizando la expresión:

$$MDL(B / D) = \sum_{i=1}^{N} \log p_B(D) - r_B * \log(N / 2) \quad (3.231)$$

Donde r_B representa el número de parámetros libres asociados a la función de distribución conjunta.

Otras medidas de calidad basadas de la teoría de la información se encuentran descritas en Cheng *et al.* (1997). Algunas otras medidas maximizan la verosimilitud condicional (aprendizaje discriminativo) pero por ahora sólo son aplicables a problemas de baja dimensionalidad dado que presentan graves problemas de eficiencia computacional, Grossman y Domingos (2004), Jing *et al. (*2005).

Cuando las redes son gausianas, si se considera que la distribución de parámetros es del tipo normal-Wishart se obtienen medidas de calidad similares a las anteriores. Para más detalles véase Geiger y Heckerman (1994).

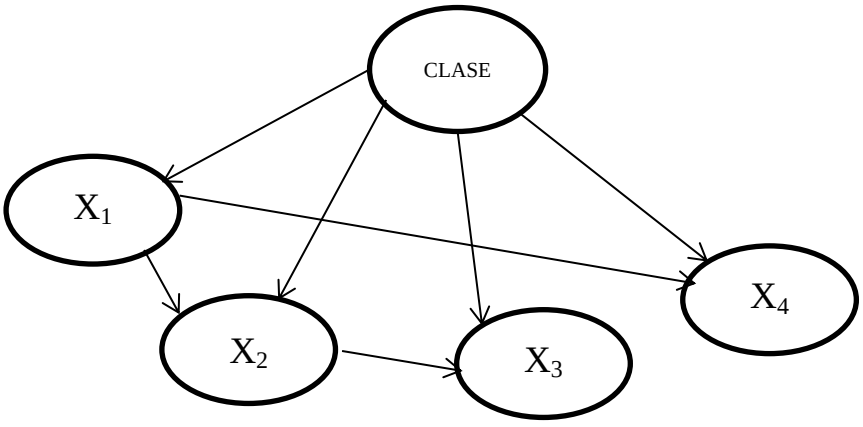
3.2.6.4.4.3. Clasificadores basados en redes bayesianas.

3.2.6.4.4.3.1. Algoritmo TAN.

A la hora de enfrentarse con la construcción de un clasificador bayesiano teniendo en cuenta las dependencias entre las variables involucradas en el problema existen alternativas más sencillas que enfrentarse a la construcción de una red bayesiana sin restricciones. En general podemos decir que estas alternativas representan una extensión del clasificador naïve-Bayes con ciertas modificaciones estructurales. La forma de proceder es construir una estructura que refleje algunas relaciones de dependencia entre los atributos manteniendo al margen la variable clase.

El algoritmo TAN del inglés Tree Augmented Network fue propuesto por Friedman y colaboradores en 1997. Este algoritmo consistió en una adaptación del algoritmo que propuso Chow-Liu (1968) El TAN utiliza el concepto de cantidad de información mutua condicionada a la variable clase, en lugar de la cantidad de información mutua en la que se basa el algoritmo de Chow-Liu. En el modelo TAN todos los atributos tienen como padre a otro atributo como mucho, además de la clase en sí, de forma que cada a tributo obtiene un arco aumentado apuntando a él.

Figura 3.38. Modelo gráfico del clasificador TAN



Dadas las variables discretas X e Y y la clase C, la cantidad de información que la variable Y nos proporciona sobre la variable X dada la variable clase es proporcionada por la siguiente expresión:

$$\mathrm{I}(\mathrm{X},\mathrm{Y}/\mathrm{C}) = \sum_{x,y,c} p(x,y/c)\log\frac{p(x,y,c)}{p(x/c)\,p(y/c)} \tag{3.232}$$

Cuando aprende la estructura del árbol entre todos los atributos, el algoritmo TAN añade la variable clase y la hace padre de todas las variables.

En Friedman y Goldszmidt (1996) se presenta un algoritmo que consta de cinco pasos:

1. En primer lugar se calcula $I(X_i;X_j \mid C)$ con $i < j$; $i,j = 1, 2, 3, \ldots\ldots, n$. Estos valores se estiman a partir de la muestra.
2. Obtenidos los valores del paso 1 se construye un grafo no dirigido completo donde los nodos se corresponden con los atributos X_1, X_2, $\ldots X_n$. Se asigna a cada arco entre los nodos X_i, X_j, un peso dado por la $I(X_i;X_j \mid C)$.
3. Se aplica el algoritmo de Krustal al grafo construido en el paso anterior con el objetivo de construir un árbol expandido máximo. Este algoritmo parte de los n(n-1)/2 pesos del grafo completo para construir un árbol con todos los nodos de tal forma que la suma de los pesos de los nodos se máxima. Para lograr esto opera en tres pasos en el que primeramente se asignan las dos aristas de mayor peso al árbol a construir. Seguidamente se examina la siguiente arista de mayor peso y se añade al grafo a no ser que forme un ciclo en cuyo caso se descarta y se busca la siguiente arista de mayor peso. Este paso se repite hasta que se hayan asignado n – 1 arista.
4. El cuarto paso consiste en transformar el árbol no dirigido resultados del paso anterior en uno que esté dirigido. para lograr esto se escoge un nodo cualquiera como raíz y asignando direcciones a todas las aristas a partir de él.
5. .El último paso para construir un modelo TAN es añadir la clase C y un arco desde C a cada atributo X_i.
6. El último paso para construir un modelo TAN es añadir la clase C y un arco desde C a cada atributo Xi.

Friedman (1996) demuestra que si el contexto en el cual los datos de entrenamiento hubieran sido generados por una estructura TAN, el algoritmo visto anteriormente es

asintóticamente correcto, lo que significa que si la muestra es suficientemente grande el algoritmo recuperará la estructura que generó los datos.

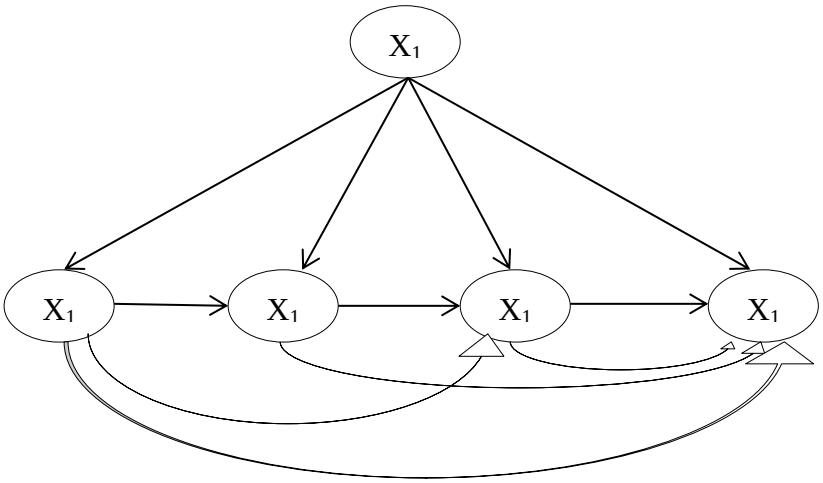
También se asegura que la estructura de red obtenida contiene la máxima verosimilitud del conjunto de todas las posibles estructuras TAN (Hernández, 2004). Por otra parte la complejidad de este algoritmo es O(n2 n), siendo n el número de atributos y N el tamaño del conjunto de entrenamiento.

Keogh y Pazzani (1999) proponen un algoritmo voraz qua va añadiendo arcos a una estructura naïve-Bayes. En cada uno de los pasos se añade un arco que mejore en mayor medida el porcentaje de instancias bien clasificadas, manteniendo la condición de que en la estructura final cada variable no tenga más de un padre.

3.2.6.4.4.3.2. Clasificadores bayesianos K-dependientes.

Un paso más en la relajación de modelo naïve-Bayes es también la generalización que realiza Sahami (1996) mediante lo que se denomina clasificadores k-dependientes (KDB)

Figura 3.39. Modelo gráfico del clasificador K dependiente.



En esta nueva estructura de red bayesiana se permite a cada variable Xi de la base de datos tener un máximo de k atributos. Su formalismo matemático sería el siguiente:

$$Pa(X_i) = \{C, X_{pa_i}\} \tag{3.233}$$

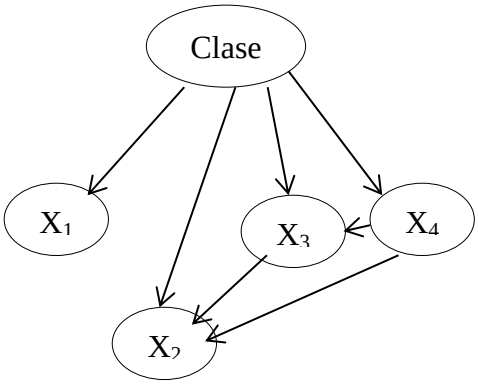
Donde X_{pa_i} es un conjunto máximo de k atributos y la clase carece de padres.

En este clasificador si variamos el valor de K nos podemos disponer de todas las combinaciones de dependencias de variables inductoras.

3.2.6.4.4.3.3. Naïve Bayes aumentado (BAN).

Este tipo de estructura denominada Naïve Bayes aumentado (Augmented Naïve Bayes) es también conocida como BAN recoge la organización de Naïve Bayes pero ahora se ve aumentada con arcos entre todas las variables pero con la única limitación de que no formen ciclos. Este tipo de estructura puede representar cualquier forma de red bayesiana.

Figura 3.40. Modelo gráfico del clasificador BAN.



3.2.6.4.4.3.4. Average One-Dependence Estimators (AODE).

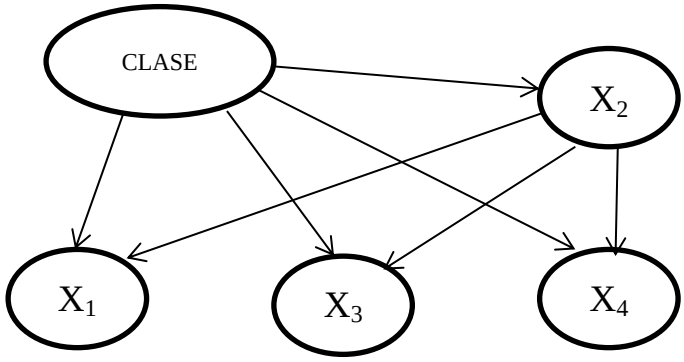
Este clasificador también está basado en el modelo Naïve Bayes pero en esta propuesta se incrementa la estructura mediante arcos relajando la suposición de independencia de las variables dada la clase que impone el método original.

Este clasificador diseñado por Webb, (2005) está siendo muy utilizado debido a su alta eficiencia y el bajo error en problemas de clasificación.

Este clasificador utiliza una estructura que está basada en el concepto de clasificador 1-dependiente de Sahami (1996). Al igual que el algoritmo TAN cada variable tiene como padre a la variable clase y como máximo a otro atributo. Sin embargo, en la selección de modelos este clasificador utiliza un conjunto de modelos, Dietterich (2000), sobre los cuales calcula una media ponderada para obtener la predicción definitiva.

De entre todos los posibles modelos de clasificadores 1-dependientes para un conjunto de datos, el clasificador AODE emplea un subconjunto que está formado por aquellos modelos en los que existe un atributo padre, que en este caso recibe el nombre de súper padre dado que hace de padre del resto de atributos. Esta estructura fue desarrollada por Keogh y Pazzani (1999 y 2002) y se conoce por sus siglas en inglés SPODE (Superparent One-Dependence Estimators). El número total estructuras que utiliza SPODE es n, una por cada variable presente en la Base de Datos en el que cada uno de ellos hace una vez de súper padre.

Figura 3.41. Grafo de un clasificador AOED.1-dependiente de tipo SPODE



Para clasificar una instancia nueva x= (x1, x2….xn) a través del conjunto de los n modelos SPODE (ensemble) utilizamos la siguiente expresión

$$\arg ma\acute{x}_{c\in\Omega_c}(\sum_{i=1}^{n}P(c,x_i)\prod_{\substack{j=1\\ j\neq 1}}P(x_j\,|c,x_i) \tag{3.234}$$

Algunas contribuciones se han orientado a la mejora de este modelo explotando la versatilidad de los modelos ensemble. Por otra parte algunos trabajos se han dirigido a plantear aproximaciones a la selección de modelos y a la ponderación de los mismos.

Las diferentes aportaciones respecto a las estrategias de selección de modelos a través de métodos wrapper se encuentran en Zheng y Webb. (2007) y Yang *et al.* (2007). Otra línea de investigación trata de clasificar nuevas observaciones realizando un selección de modelos perezosa (Lazy), Zheng y Webb. (2007)

Otras aproximaciones modelos se centran en la ponderación o pesos que se da a cada modelo en el proceso de agregación en lugar de la ponderación uniforme que

utiliza AODE. Yang *et al.* (2007) y Jiang y Zhang (2006) utilizan una media ponderada. Otras estrategias se han encaminado a utilizar ponderaciones más avanzadas a través de modelos bayesianos (Bayesian model averaging), Hoeting *et al.* (1999) o mediante mixturas lineales, Cerquides y López (2005).

Una aportación interesante la realizan Flores *et al.* (2009) que proponen un clasificador híbrido para incluir la posibilidad de utilizar todo tipo de bases de datos. En esta propuesta los autores deciden considerar cada súper-padre como discreto en su modelo correspondiente, en principio, a través de cualquier método de discretización.

Yang *et al.* (2007) realizan un amplio estudio donde analizan el rendimiento del clasificador original AODE y los comparan con distintas aproximaciones basadas en selección de modelos y ponderación de modelos y concluyen que el clasificador AODE presenta una notable robustez y estabilidad.

3.2.6.4.4.3.5. Enfoques semi naïve Bayes y clasificadores extendidos.

En este epígrafe lo que se trata es de recoger algunos de las múltiples propuestas sobre el clasificador Naïve-Bayes que consiguen de una forma u otra mejorar su exactitud y que son recogidos en la literatura como enfoque semi Naïve-Bayes.

Para organizar de alguna manera estos diferentes perspectiva se sigue y se extiende la ya conocida de Webb y Panzani (1998) donde

El conjunto de las diferentes propuestas se organizan en función de las actividades pre/post proceso que realicen. De esta manera el autor encuentra tres grandes grupos:

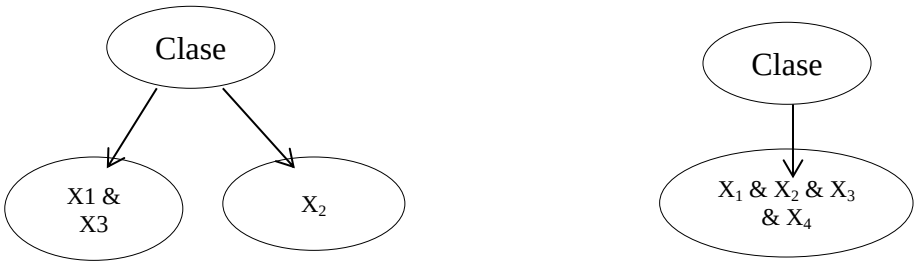
1. Aquellos procedimientos que procesan de alguna marea las variables antes de aplicar el Naïve-Bayes.
2. Enfoques que corrigen las probabilidades que arroja Naïve-Bayes.
3. Otros trabajos donde antes procesar el algoritmo del Naïve-Bayes se seleccionan el conjunto de instancias.

En el primer grupo se encuentran todos los procedimientos relacionados con la selección de variables, que ya fueron expuestos en el capítulo 4 dedicado a la metodología de esta tesis doctoral, y que de forma resumida podemos recordar que se pueden utilizar dos enfoques filter y wrapper, Kohavi y John (1997).

En relación a la selección de variables existen algunas referencias bibliográficas interesante con estrategias de tipo wrapper donde se utilizan diversa procedimientos como por ejemplo algoritmos genéticos en *Liu et al.* (2001) y en Inza *et al.* (2000). Este último autor también utiliza procedimientos de estimación denominados algoritmos de estimación de distribuciones (EDAs).

Otro enfoque con bastante buenos resultados es el que emplea Panzani (1997). En el enfoque que maneja el autor, a través de un algoritmo voraz es capaz de detectar variables irrelevantes y variables dependientes ante de emplear el modelo Naïve-Bayes.

Figura 3.42. Grafo de una Red de PANZANI.



Los algoritmos que se propones para corregir las probabilidades producidas por el método Naïve-Bayes son muay variados peros los principales se puede sintetizar en los siguientes:

- Webb y Panzani (1998) proponen un método que realiza un ajuste lineal del peso de las probailidades de cada clase denominado APNBC (Adjusted Probability Naïve-Bayes) Este ajuste se puede expresar de la siguiente manera:

$$P(C = c_i \mid X_i = x_i,...,X_n) \propto w_i P(C = c_i \prod_{k=1}^{n} P(X_k = x_k \mid C = c_i) \qquad (3.235)$$

- Ferreira *et al.* (2001) realizan una nueva aportación en el mismo sentido que los anteriores autores pero en vez de ponderar la clase permitiendo asociar pesos a particiones. En este procedimiento los ejemplos de cada variable son particionadas de forma recursiva con el objetivo de maximizar la entropía. El ajuste es el siguiente:

$$P(C = c_i \mid X_i = x_i,...,X_n) \propto w_i P(C = c_i \prod_{k=1}^{n} [P(X_k = x_k \mid C = c_i)]^{w(X_k,k,i)} \quad (3.236)$$

Donde $^{w(X_k,k,i)}$ es la función de peso para la variable X_k

En los resultados experimentales llevados a cabo con las bases de datos de la UCI, Murph y Aha (1995), para comprar este procedimiento el porcentaje alcanzado sólo supera el 0,6%.

- El algoritmo Iterative Bayes propuesto por Gama (2000) parte de la idea de mejorar la forma iterativa de calcular las probabilidades en el método original naïve-Bayes. La forma que se propone es avanzar a través del algoritmo de ascenso de colinas HC (Hill Climbing). En cada iteración se utilizan las probabilidades actuales para todas las instancias del conjunto de entrenamiento a través de la siguiente función:

  $$\frac{1}{N}\sum_{i=1}^{N}(1 - máxp(C = c_j \mid x^{(i)})) \quad (3.237)$$

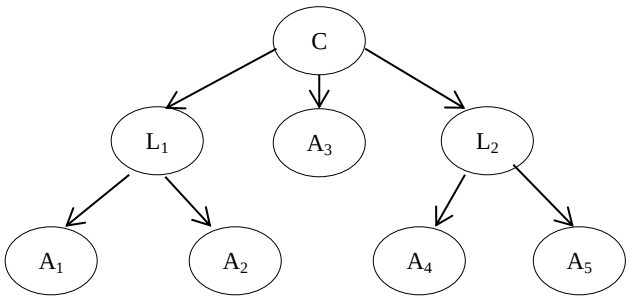
  Donde N representa el conjunto de todos los ejemplo de la base de datos y j son los valores de la clase.

  Al contrastar los resultados de este algoritmos sobre 27 conjuntos de datos los autores consiguen reducir el error medio en un 1,24%.
- El enfoque que utiliza Zaffalon (2002) se basa en conjuntos convexos de distribuciones de probabilidad (conjuntos credales) con lo que supera el requerimiento de los valores puntuales de naïve- Bayes. El naïve Bayes Credal (NBC) representa las distribuciones de probabilidad como puntos que pertenecen a regiones geométricas cerradas y acotadas y que se encuentran descritas por restricciones lineales.
- El clasificador Robust Bayesian Classifier (RBC) que propusieron Ramoni y Sebastiani (2001) es un algoritmo desarrollado para aplicar cuando en la base de datos hay valores ausentes.
- Un trabajo interesante es el de Greiner y Zhou (2002) cuya propuesta consiste en encontrar los parámetros de la distribución a través de la maximización de la verosimilitud condicional en lugar de la verosimilitud de la muestra en una red bayesiana. Se tiene en cuenta la existencia de la variable a clasificar y se propone un algoritmo de descenso por el gradiente. T

- En Robles (2003) se proponen un nuevo algoritmo denominado Internal Estimation naïve-Bayes (IENB) cuyo enfoque se centra en estimar las probabilidades necesarias para el clasificador naïve-Bayes a través de una estimación por intervalo en lugar de la estimación puntual que se aplica en el método primitivo. La búsqueda se realiza con algoritmos heurísticos de optimización (EDAs) y está guiada por la exactitud de los clasificadores.

Podemos encontrar otros algoritmos basados en Naïve Bayes y que en alguna publicación se ha denominado Clasificadores ingenuos extendidos, que también representan una buena opción de aprendizaje para resolver problemas de clasificación, como por ejemplo los Clasificadores Ingenuos Jerárquicos, Hierarchical Naïve Bayes (HBN) que están descritos en Zhang (2002) y por Langseth y Nielsen (2002). Una representación gráfica de este modelo se observa en la figura 3.43.

Figura 3.43. Ejemplo de Hierarchical Naïve Bayes.



En esta configuración se introducen variables latentes con el objetivo de relajar las hipótesis de independencia entre los atributos. Estos atributos son introducidos considerando la dependencia condicional entre ellos seleccionando la pareja de atributos que consiga la mayor información mutua.

Otras aportaciones en este sentido nos las ofrece De Campos y Castellanos (1997) introduciendo en su algoritmos restricciones estructurales derivadas del conocimiento previo o del orden de las variables, También Acid *et al.* (2005) también proponen un procedimiento donde consideran clases equivalentes tanto en términos de independencia condicional como en términos de equivalencia para reducir el espacio de búsqueda.

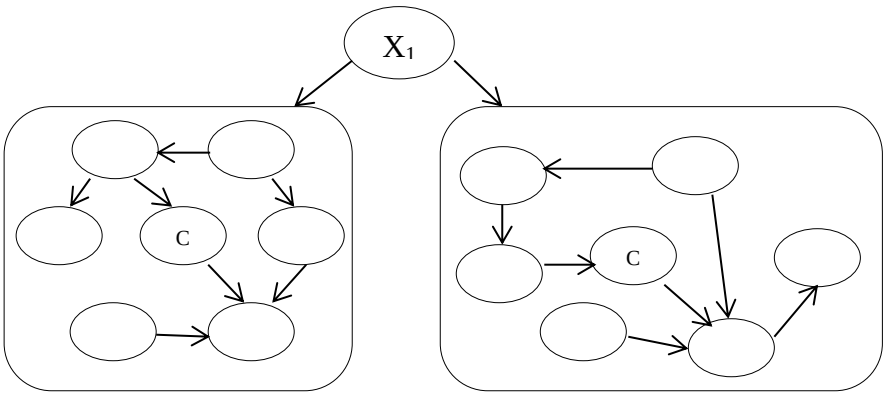
Existe una amplia literatura sobre mezclas o comités de expertos combinando las predicciones de diferentes clasificadores, Lanngseth y Nielsen (2006) y Webb *et al.* (2005).

Para concluir este epígrafe reseñar que en Larrañaga *et al.* (2005) existe una amplia información sobre el aprendizaje de clasificadores bayesianos.

3.2.6.4.4.3.6. Multiredes bayesianas.

Un avance en la modelización de datos a través de redes bayesianas es cuando consideramos un clasificador distinto para cada una de las clases tal y como se observa en el gráfico 3.44 porque una conclusión que se extrae de la observación en la construcción de las estructuras de las propuesta anteriores es que la relación es siempre la misma en cada una de las clases. Esta consideración nos permite representar independencias asimétricas, Heckerman (1991). Este autor distingue dos tipos de asimetrías, una que podemos encontrar entre las variables y los atributos, denominada asimetría de subconjunto y otra que el autor denominó hipótesis específica y que se origina cuando solo se considera la relación entre atributos. En la primera asimetría se construye una rede bayesiana para a valor de la clase a clasificar y, en la segunda, como se refleja en el grafico 3.44.

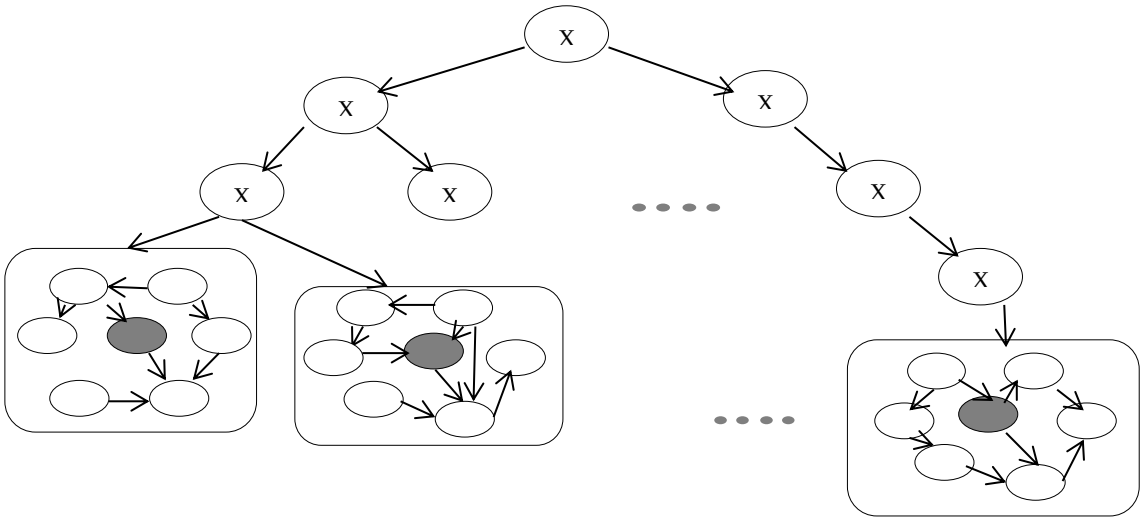
Figura 3.44. Multired bayesiana.



Esta construcción es considerada como una extensión de las redes bayesiana y el desarrolla y formalismo se puede encontrar en Geiger y Heckerman (1995).

Otro tipo de estructura multired que se encuentra en la literatura consultada es el que se conoce como redes recursivas y que se caracteriza por que se seleccionan diferentes atributos agrupados en forma de árbol donde sus extremos diferentes clasificadores bayesiana, como se puede apreciar en la figura 3.45.

Figura 3.45. Multired bayesiana recursiva.



3.2.6.4.4.3.7. Naive Bayes extendido a través de un árbol de clasificación (NBtree).

El Naïve Bayes Tree propuesto por Kohavi (1996) es un algoritmo híbrido entre un árbol de clasificación y el clasificador naïve Bayes. También podemos considerarlo como un tipo especial de multired bayesiana recursiva donde las hojas son clasificadores naïve bayes. En realidad, lo que perseguía el autor al crear esta estructura era combinar las ventajas de los árboles de decisión, Quinlan (1993) y las virtudes del clasificador Naïve Bayes.

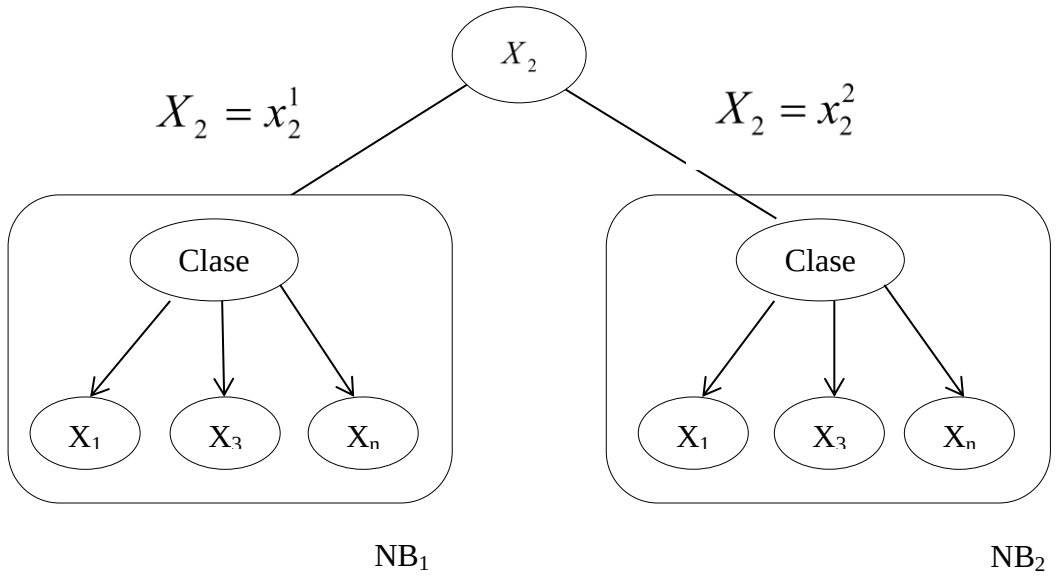
En esta estructura podemos observar las propiedades de la misma: (i) una primera observación es que, como en los algoritmos anteriores, cada nodo interno de la red representa una variable, (ii) estos nodos engendra tantos hijos o rama salientes como estados tiene la variable, (iii) todas as hojas se encuentran en el mismo nivel, (iv) en cualquier camino que se transite desde la raíz hasta las hojas no existen variables repetidas.

Un algoritmo heurístico para estimación de esta estructura se encuentra en Peña *et al.* (2002).

Es evidente que en la construcción de la estructura NBtree se pueden generar nodos terminales que contengan un reducido número de instancias por lo que algunas aportaciones intentan mejorar este problema que se resuelve a través del algoritmo

LBR (Lazy Bayesian Rule) de Zheng y Webb (2000). Las pruebas que se realizaron contrastando ambos enfoques sobre 29 conjuntos de datos del repositorio UCI demuestran la superioridad del algoritmo LBR sobre el naïve BayesTree.

Figura 3.46. Grafo de una Red NBtree.



3.2.6.4.4.4. Tipos de redes bayesianas.

Los diferentes tipos de redes bayesianas vienen determinadas por el carácter discreto o continuo de las variables involucradas en el modelo.

Redes bayesianas Gausianas.

Cuando las variables siguen una distribución normal multivariante N(μ,Σ) decimos que la red es gausiana. la función de densidad conjunta viene determinada por la siguiente expresión:

$$f(x) = (2\pi)^{-n/2} \left|\sum\right|^{-1/2} \exp\left\{-1/2 \left(x-\mu\right)^T \sum^{-1} \left(x-\mu\right)\right\} \qquad (3.238)$$

donde μ es el vector de medias n-dimensional, $\sum$ es la matriz de covarianzas n x n, $\left|\sum\right|$ es el determinante de $\sum$, y μ^T denota la traspuesta de μ.

Redes bayesianas Multinomiales.

En este tipo de red se considera que todas las variables son discretas lo que implica que todas las variables tienen un número finito de posibles estados. También suponemos que las funciones de probabilidad de cada variable condicionada a sus predecesores (padres) es también multinomial y por lo tanto están especificadas en las diferentes combinaciones de estado de las variables involucradas. La reducción de parámetros a estimar es considerable.

Redes bayesiana Mixtas.

Las redes bayesianas mixtas tienen un alto grado de complejidad a la hora de definirlas, aunque caso particulares han sido tratados. En Jordan (1998) se describe un caso en el que se permite que una variable continua tenga padres con valores discretos. Otro ejemplo lo encontramos en Castillo *et al,* (1998) donde abordan un problema utilizando variables discretas y funciones Beta.

3.2.7. Sistemas múltiples de clasificación.

3.2.7.1. Introducción a los métodos de combinación de modelos.

Una forma de conseguir una mayor precisión de las predicciones de nuestros modelos es acudir a los multiclasificadores. La combinación de las hipótesis de los multiclasificadores es una excelente forma de integrar la información de diferentes fuentes. Esta combinación de dos o más clasificadores, en general, proporciona estimaciones más robustas y eficientes que cuando se utiliza un único clasificador. También se utilizan porque resuelven el problema de sobreadaptación (overfitting) y es posible obtener buenos resultados con pocos datos.

Son múltiples los estudios que se han realizado con los métodos multiclasificadores, así que podemos conocerlos en la literatura existente con muchos nombres: métodos de ensamble, métodos híbridos de clasificación, modelos múltiples, sistemas de múltiples clasificadores, combinación de clasificadores, integración de clasificadores, mezcla de expertos, comité de decisión, fusión de clasificadores y aprendizaje multimodelo.

Existen diferentes formas de combinar conjuntos de modelos. Dietterich (2000) estableció una clasificación atendiendo a diferentes criterios:

- Manipulación de los datos entrenamiento: en estos procedimientos se construye un grupo de modelos mediante la repetición de k veces el mismo algoritmo de aprendizaje. Es importante el mecanismo de selección de los subconjuntos a partir de los datos de entrenamiento. Los métodos multiclasificadores más conocidos son: Bagging (Breiman 1996, Quinlan 1996a), Boosting (Freund y Schapire 1996; Quinlan 1986b) y Cross-Validated Committes (Parmanto *et al*. 1996).
- Manipulación de las variables de entrada: En esta técnica se altera el conjunto de atributos de entrada del algoritmo de aprendizaje. A esta familia de multiclasificadores pertenece las técnicas forest, Ho. (1995).
- Métodos aleatorios: En estas técnicas se introducen componentes aleatorios en los proceso de aprendizaje con el objetivo de obtener diferentes multiclasificadores a partir de los mismos datos. Manipulación de los datos de salida.
- Manipulación de los datos de salida. En problemas de clasificación se modifican las clases de los conjuntos de los ejemplo del conjunto de entren**a**miento.

A continuación se describen brevemente los principales métodos utilizados en minería de datos:

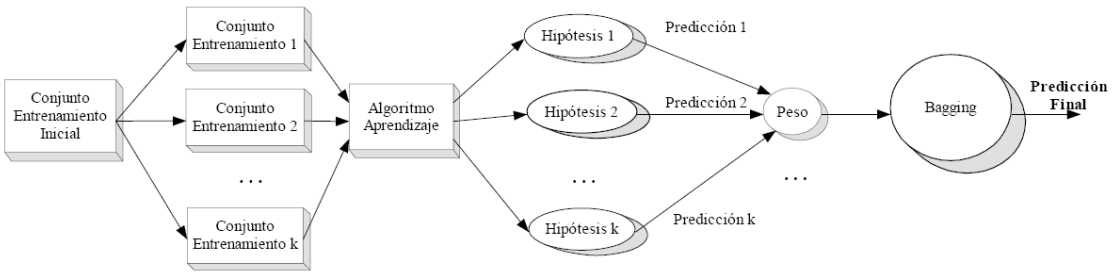
3.2.7.2. Bagging.

Este método propuesto por Breiman (1996) intenta aunar las características del Bootstrapping y de la agregación incorporando los beneficios de ambos y dándole el nombre (Bootstrap AGGregatING). En este método se generan muestras aleatorias que serán los conjuntos de entrenamiento. Las muestras se generan a través de muestreo aleatorio con reemplazamiento. Cada subconjunto de entrenamiento aprende un modelo. Su principal utilidad es que reduce la varianza existente en la generación de cada modelo.

Para clasificar un ejemplo se predice la clase de ese ejemplo para cada clasificador y se clasifica la clase con mayor voto. Es decir, este método, a la hora de emitir una decisión, recurre a la decisión mayoritaria.

La arquitectura de este modelo se corresponde con el siguiente gráfico:

Figura 3.47. Estructura del multiclasificador Bagging.



A continuación se muestra la forma de implementar este algoritmo.

Figura 3.48. Algoritmo de Bagging para clasificación.

ALGORITMO *Bagging* (k:iteraciones, *E*: conjunto de ejemplos, *A*: Algoritmo de aprendizaje)
$i \leftarrow 1$
$C \leftarrow \emptyset$
REPITE
Extrae una muestra *E'* de *n* ejemplos con remplazamiento desde *E*
// *E* contiene *n* ejemplos
$m \leftarrow A(E')$ // Aprende un modelo con *A* utilizando el conjunto *E'*
$\{M\} \leftarrow \{M\} + m$
HASTA *i=K*
FIN ALGORITMO

Para clasificar un ejemplo *e*, se predice la clase de ese ejemplo para cada clasificador de *C*, y se selecciona la clase con mayor número de votos.

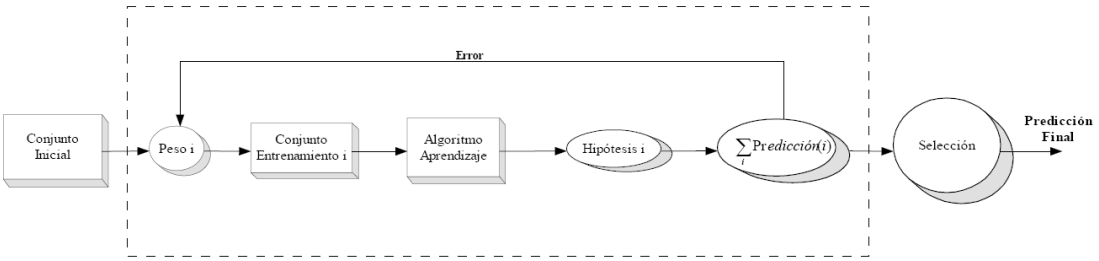
3.2.7.3. Boosting.

Este método fue propuesto por Freund y Schapire (1996). El mecanismo que proponen sus autores está basado en la asignación de un peso a cada conjunto de entrenamiento. Cada vez que se itera se aprende un modelo que minimiza la suma de los pesos de aquellos ejemplos clasificados erróneamente. Los errores de cada iteración sirven para actualizar los pesos del conjunto de entrenamiento, incrementando el peso de los mal clasificados y reduciendo el peso en aquellos que han sido correctamente clasificados.

La decisión final para un nuevo patrón de clasificación viene dada por votación mayoritaria ponderada entre los H conjuntos de entreanmiento. La ponderación de los modelos es estática.

La estructura gráfica de este método es la siguiente:

Figura 3.49. Estructura del multiclasificador Boosting.



La variante más conocida de estos algoritmos es AdaBoost que se encuentra implementado en múltiples programas. El pseudo código para su implementación se muestra en la figura 3.50.

Otra variante de este algoritmo que realiza Breiman (2001) y que denomina Arc-x4 se basa en dos aportaciones importantes: la asignación de pesos y el esquema de votación. Respecto a la primera diferencia la asignación de los pesos es mucho más simple que AdaBoost, el ajuste de los pesos es proporcional al número de errores que obtuvo el ultimo clasificador elevado a la cuarta potencia más uno. En cuanto al esquema de votación lo que realiza es combinar las decisiones individuales con la votación simple no ponderada.

Figura 3.50. Algoritmo de Adaboost M1 para clasificación.

Input:
BaseLearn – base learning algorithm
T – set of *m* training examples < $(x_1, y_1),...,(x_m y_m)>$ with labels $y_j \in Y$
I number of Boosting iterations
Initialize Distribution of weights on examples, $D_1(x_j) = 1/m$ for all $x_j \in T$

(1) For $i = 1$ to I
(2) Train base learner given the distribution $D_i, C_i = BaseLearn(T, D_i)$
(3) Calculate error of $C_{i,e_i} = \sum_{\substack{x_j \in T, \\ C_i(x_j) \neq y_j}} D_i(x_j)$
(4) If $e_i > 1/2$ or then set $I = i-1$ and abort loop
(5) Set $\beta_i = e_i / (1 - e_i)$
(6) Update weights, $D_{i+1}(x_j) = D_i(x_j)$ x $\begin{Bmatrix} \beta_t : if\ C_i(x_j) = y_j \\ 1 : otherwise \end{Bmatrix}$
(7) Normalize weights, $D_{i+1}(x_j) = \dfrac{D_{i+1}(x_j)}{\sum_{x_j \in T} D_{i+1}(x_j)}$

Output The final hypothesis, $C^*(x) = \arg\max_{y \in Y} \sum_{i:C_i(x)=y} \log \dfrac{1}{\beta_i}$

Figura 3.51. Algoritmo Arc-x4 *para clasificación.*

Entradas: M = Un conjunto de *m* patrones etiquetados: $\{x_i \mid i = 1,2,...m\}$
LEARN (algoritmo de aprendizaje)
MalClasif = Acumulador de los errores cometidos por cada clasificador
e = Total de errores cometidos por un clasificador
E = Error calculado para el clasificador D_f

Inicia
$w_f(i) = 1/m \ \forall i$ //Inicializa los pesos para cada patrón
$MalClasif_f(i) = 0 \quad \forall i$ //Inicializa acumulador de mal clasificados
Para $f = 1,...,H$
D_f = LEARN (w_f) //Construcción de D_f considerando w_f
$e_f = \sum_i$ [**1** si $D_f(x_i) \neq$ etiqueta verdadera (x_i) sino **0**] //Determina aciertos
$E_f = \sum_i w_f(i) * e_f(i)$ //Cálculo del error del clasificador D_f
Si $E_f > 0.5$ **entonces**
H=f-1
terminar //Finaliza la construcción de clasificadores
Sino
$MalClasif_f(i) = MalClasif_{f-1}(i) + e_f(i) \ \forall i$ //Actualiza mal clasificados
$w_{f+1}(i) = 1 + MalClasif_f(i)^4 \quad \forall i$ //Actualización de pesos
fin Si
$w_{f+1}(i) = w_{f+1}(i) / \sum_i w_{f+1}(i) \quad \forall i$ //Normalización de pesos
fin para
Fin

3.2.7.4. Decorate.

El método propuesto por Melville and Mooney (2003) denominado DECORATE (Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples) lo que primero realiza es un entrenamiento del primer clasificador base con todas las instancias del conjunto de entrenamiento. En las iteraciones siguientes emplea una mezcla de registros del conjunto de entrenamiento original con otras creadas de forma artificial. A la hora de generar estas instancias respeta la distribución de cada uno de los atributos, que asume son independientes. Para un atributo numérico, se calcula la media y la desviación estándar del conjunto de entrenamiento y se generan los valores de la distribución normal de Gauss. Los valores nominales se generan manteniendo la misma probabilidad que en el conjunto original. Se utiliza el suavizado de Laplace de manera que los valores de los atributos nominales no representados en la muestra tengan asignada una probabilidad de ocurrencia distinta de cero.

En las pruebas que realiza el autor con 15 bases de datos del repositorio de la UCI donde compara su procedimiento con AdaBoost, Bagging, Random Forests yJ48, usando J48 como algoritmo base del método ensamblador, obtiene unos resultados excelentes en el desempeño de su método.

Figura 3.52. Algoritmo Decorate para clasificación.

Input:
BaseLearn – basse learning algorithm
T – set of *m* training examples $<(x_1, y_1),...,(x_m, u_m)>$ with labels $y_j \in Y$
C_{size} - desired ensemble size
$I_{\max}$ - maximum number of iterations to build an ensemble
R_{size} - factor that determines number of srtificial examples to generate

(1) *i*=1
(2) *trials* = 1
(3) $C_i = BaseLearn(T)$
(4) Unitialize ensemble, $C^* = \{C_i\}$
(5) Compute ensemble error, $\varepsilon = \frac{\sum_{x_j \in T, C^*(x_j) \neq y_j} 1}{m}$
(6) While $i < C_{size}$ and $trials < I_{\max}$
(7) Generate $R_{size} \times |T|$ training examples, R, based on distribution of training data
(8) Label examples in R with probability of class labels inversely proportional to predictions of C^*
(9) $T = T \cup R$
(10) $C' = BaseLearn(T)$
(11) $C* = C* \cup \{C'\}$
(12) $T = T - R$, remove the artificial data

(13) Compute training error, e', of $C*$ as in step 5
(14) If $e' \leq e$
(15) $i = i + 1$
(16) $e = e'$
(17) Otherwise,
(18) $C* = C* - \{C'\}$
(19) *Trials*=*trials*+1

3.2.7.5. Fusión de clasificadores.

Una vez construidos los modelos la predicción de nuevos casos se realiza mediante la fusión o combinación de las predicciones de cada modelo generado.

Siguiendo a Kuncheva (2002) vamos a suponemos que estamos trabajando con un multiclasificador que incluye m miembros o modelos y que se han definido varios métodos que nos permiten unificar los m vectores de probabilidad en un único vector α, entonces, algunas estrategias de fusión para extraer la clase predicha son las siguientes.

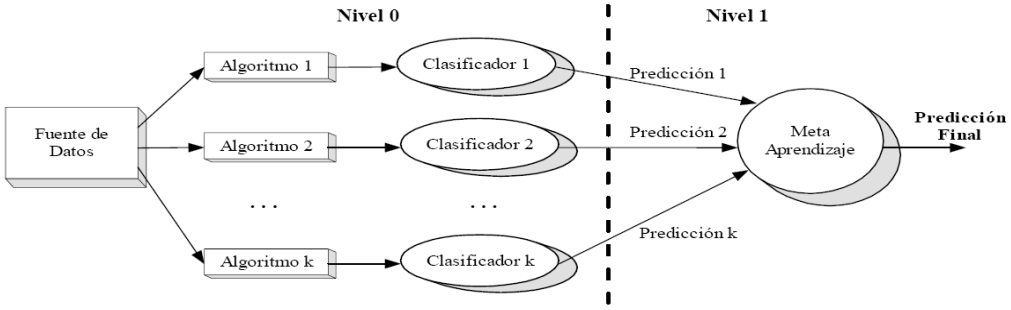
- Suma: $$\alpha = \sum_{j=1}^{m} \nu_j \quad (3.239)$$
- Media aritmética: $$\alpha = \sum_{j=1}^{m} \frac{\nu_j}{m} \quad (3.240)$$
- Producto: $$\alpha = \prod_{j=i}^{m} \nu_j \quad (3.241)$$
- Media geométrica: $$\alpha = \sqrt[m]{\prod_{j=1}^{m} \nu_j} \quad (3.242)$$
- Máximo: $$\alpha = \max_{1 \leq j \leq m} (\nu_j) \quad (3.243)$$
- Mínimo: $$\alpha = \min_{1 \leq j \leq m} (\nu_j) \quad (3.244)$$
- Mediana: $$\alpha = mediana_{1 \leq j \leq m} (\nu_j) \quad (3.245)$$

3.2.7.6. Métodos híbridos.

3.2.7.6.1. Stacking.

Este método combina múltiples clasificadores a través de diferentes algoritmos de aprendizaje. Los algoritmos de aprendizaje de la primera fase pueden ser árboles de decisión, redes neuronales, máquinas de vectores soporte, regresión logística, etcétera. En una segunda fase otro clasificador combina las salidas de los modelos de la fase anterior. La combinación de los clasificadores se realiza por mayoría. Este esquema funcionará bien cuando todos los modelos utilizados tienen una precisión aceptable.

Figura 3.53. Estructura del multiclasificador Stacking.

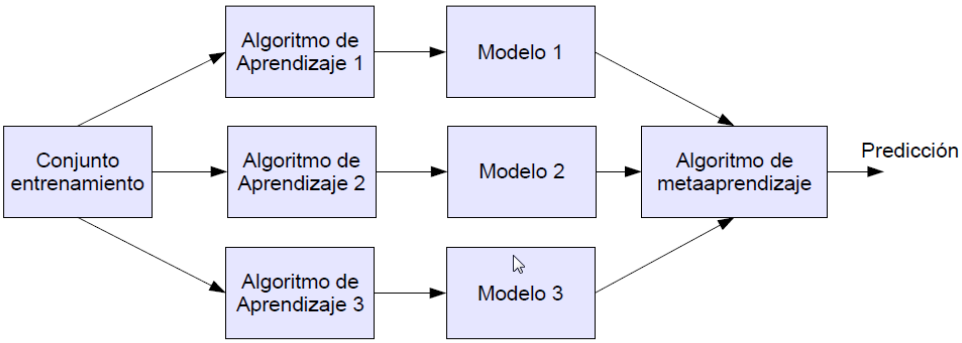


En el ejemplo que se expone se combinan primeramente tres modelos: regresión logística, una red bayesiana con K2 y como algoritmo de aprendizaje (metaclasificador) utilizaremos un árbol de decisión.

3.2.7.6.2. Cascading.

Gama y Bradzil (2000) presentan este método que nos permite mejorar las características de los árboles de decisión al incorporar nuevas particiones utilizando otros procedimientos de aprendizaje como hemos visto en el anterior multiclasificador. Cascading utiliza otros métodos de aprendizaje para crear nuevos atributos a través de redes neuronales, análisis discriminante, etc.

Figura 3.54. Estructura del multiclasificador Cascading.



CAPÍTULO 4

METODOLOGÍA APLICADA EN ESTA TESIS DOCTORAL.

4. Metodología aplicada en esta tesis doctoral.

4.1. Introducción.

Las metodologías presentadas sucintamente en el primer epígrafe del capítulo 3, tanto del proyecto CRISP-DM (Cross Industry Standard Process for Data Mining) enfocada en proceso industriales, como la metodología SEMMA (Sampling, Exploration, Modification, Modelization, Asessment) cubren todas las fases que implica una correcta metodología, pero su uso es de carácter general en los problemas de minería de datos. Si bien se entiende que las metodologías de resolución de problemas estadísticos o de minería de datos siguen un procedimiento común, cada caso en el que se trabaja dispone de una problemática especial que requiere que, en algunas etapas del proceso, se ponga especial atención en algunos asuntos de vital importancia para la óptima resolución del problema planteado. En los datos de estudio que utiliza esta tesis doctoral se presentan algunas cuestiones claves que requieren de soluciones específicas y que se detallan en los epígrafes siguientes.

4.2. Fases de la metodología aplicada en la tesis doctoral.

Los pasos metodológicos que integran el proceso de extracción de conocimiento útil y que se han llevado en esta investigación se pueden observar en la figura 4.1. Estas cinco fases se concretan en las siguientes:

- ✓ Formulación del problema. Integración de la información.
- ✓ Selección de datos, limpieza y transformación.
- ✓ Exploración y preprocesado de los datos.
- ✓ Análisis de los modelos predictivos.
- ✓ Gestión del modelo de conocimiento.

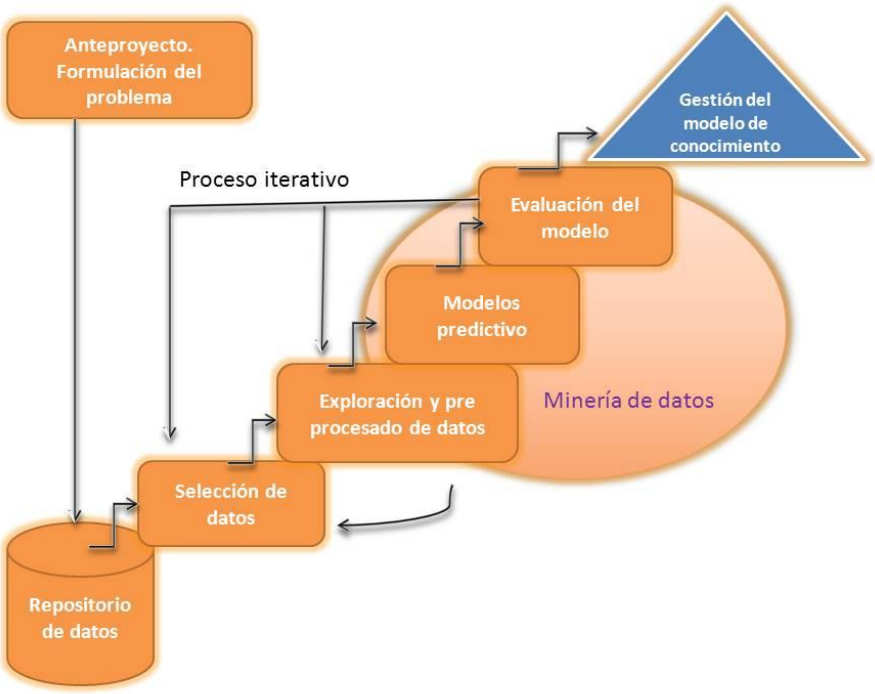
4.2.1. Formulación del problema. Integración de la información.

En la fase inicial del proyecto de tesis doctoral se contactó con el Departamento de Análisis de Caja Rioja y se contó con su colaboración para llevar a cabo una aplicación de scoring a través de los procedimientos de la minería de datos que pudiera, posteriormente, contrastarse con los métodos utilizados por la Caja de Ahorros para detectar la probabilidad de default, y cuya finalidad fuera conseguir el mejor método clasificador de los clientes que solicitan un crédito, lo que conllevaría al Banco a gestionar de forma más eficaz su negocio.

Por otra parte, el interés de esta tesis doctoral, como se ha señalado en los objetivos es conseguir, no sólo el mejor método clasificador sino también un clasificador acorde con las recomendaciones del Comité de Basilea II y III.

Definidos los objetivos a alcanzar los datos que utiliza esta tesis doctoral fueron aportados por la Caja de Ahorros de la Rioja actualmente integrada en Bankia. La base de datos contiene información de los clientes que solicitaron un crédito de consumo durante un periodo referido a los años 2010 y 2011. Estos datos fueron extraídos de su Repositorio de Datos y entregado en un fichero plano con el que se empezó a trabajar.

Figura 4.1. Fases de la metodología aplicada en la tesis doctoral.



Fuente: Elaboración propia.

4.2.2. Selección de datos, limpieza y transformación de la base de datos.

A partir del momento en que se entrega el fichero inicial se empiezan a analizar las posibles variables que constituirán el fichero con el que se iniciará la siguiente fase.

4.2.2.1. Descripción de la base de datos empleada.

El conjunto de datos facilitado por la institución contenía 1.786 registros que representan a los clientes de una Caja de Ahorros de la Rioja que demandaron un crédito entre los años 2010 y 2011. Del total de los casos, 1.609 devuelven el crédito

frente a los 177 que no reingresan el dinero prestado. La base de datos original entregada contiene diecisiete variables y sus atributos son tanto numéricos como nominales. Los atributos de cada cliente nos informan sobre diversas cuestiones: estado civil, sexo, edad, tipo de trabajo, código de profesión, situación de la vivienda, nacionalidad, etcétera, así como de otra información relacionada con el crédito: finalidad, importe solicitado, importes pendientes en su entidad bancaria y en otras, patrimonio, valor neto de la vivienda, situación de ingresos, cuotas y gastos de alquiler y préstamos, etcétera. También sabemos si el crédito se ha concedido o se ha denegado.

La composición de la Base de Datos aportada por los técnicos de Caja Rioja es la siguiente:

DATOS DE INTERVINIENTE:

NUMSOL: número de solicitud de scoring.
NUMPER: número de persona (código identificativo de la persona en la Entidad).
NUM_FAMILIA: número de componentes de la unidad familiar.
CODESTCIV: código de estado civil:

B – Casado separación de bienes.
D – Separado.
G – Casado gananciales.
H – Soltero – pareja de hecho.
I – Separado – pareja de hecho.
J – Divorciado – pareja de hecho.
K – Viudo – pareja de hecho.
R – Divorciado.
S – Soltero.
V – Viudo.

FECNACPER: fecha nacimiento.
CODSEXPER: sexo (V - Varón, M – Mujer).
TIPCNTLAB: tipo de contrato laboral:

01 – Fijo.
02 – Temporal.
03 – Temporero.
04 – Autónomo.

05 – Pensionista.

06 – Otros.

07 – Autónomo + asalariado.

CODCNO: código nacional de ocupación.

SALMED: saldo medio vista.

SALMEDNVI: saldo medio no vista.

SITVIVHBT: situación vivienda habitual:

1 – Propiedad libre de cargas.

2 – Propiedad hipotecada (mantiene).

3 – Propiedad hipotecada (cancela).

4 – Alquiler.

5 – Domicilio padres – familia.

6 – Otros.

7 – Alquiler – adquisición de vivienda.

PROFESIÓN: La profesión se presenta codificada a cuatro dígitos donde los dos últimos dígitos del código son la profesión según la Clasificación Nacional de Ocupaciones del INE (CNO).

AANCMPVIV: año compra/alquiler vivienda.

IMPVALVIV: valor neto viviendas propias.

IMPPAT: patrimonio.

IMPPMOPENENT: importe pendiente préstamos en la Entidad.

IMPPOMPENOTR: importe pendiente préstamos en otras entidades.

IMPINGFIJ: ingresos fijos anuales.

IMPINGVAR : ingresos variable anuales.

IMPOTRING: otros ingresos anuales.

IMPGASCUO: cuotas préstamos caja anuales.

IMPGASOTR: cuotas préstamos otras entidades anuales.

IMPGASALQ: gastos alquiler anuales.

IMPGASUNIFAM: gastos de otros miembros de la unidad familiar anuales.

CODNACION: código de nación. Se adjunta tabla con los códigos de nación 0017.

DATOS DE OPERACIÓN:

IMPPMO: importe del préstamo.

CODDIV: divisa de la operación.

CODPLA: plazo operación meses.

FINALID: finalidad la operación:

206 – Reformas vivienda.
301 – Compra de automóviles.
302 – Compra de electrodomésticos.
303 – Compra de ordenador y complementos informáticos.
304 – Compra de tv y otros equipos de imagen y sonido.
305 – Compra de mobiliario y decoración.
306 – Compra de motos, ciclomotores y bicicletas.
307 – Compra de vehículos de ocio.
308 - Compra de otros bienes y servicios corrientes.
404 – Compra de obras de arte, pieles y joyas.
502 – Reparación de vehículos.
504 – Financiación de servicios sanitarios.
505 – Financiación de estudios.
506 – Financiación de imprevistos familiares.
508 – Financiación de viajes y vacaciones.
509 – Financiación de celebraciones familiares.

IMPINV: importe de la inversión.
IMPCUO: importe cuota préstamo.
CODPUNSCO: puntuación scoring.
MEJTIT: mejor titular.
DICTAMEN: sanción de la operación.

En la Base de datos se realizan un conjunto de transformaciones de algunas de las variables. Estas modificaciones son las siguientes:

En la variable GASTOS se **i**ncluyen las cuotas de los préstamos contraídos en la propia Caja de Ahorros, en otras entidades bancarias, el gasto de alquiler y los gastos de los otros miembros de la unidad familiar.

El tipo de trabajo se recodifica en las siguientes categorías:

1 Fijo.
2 Temporal.
3 Autónomo.
4 Pensionista.
5 Otros.

Respecto a la profesión la base de datos original contenía 68 categorías clasificadas según la Clasificación Nacional de Ocupaciones que se han agrupado en ocho grupos:

1 Técnico superior.
2 Mando intermedio.
3 Administrativo.
4 Obrero especializado.
5 Obrero.
6 Profesión no liberal.
7 Pensionista.
8 Otras.

Combinando la profesión y el tipo de trabajo se forma una única categoría que se denomina RELACIÓN_LABORAL (Tipo de trabajo) y que contiene las siguientes categorías.

1 Técnico – mando intermedio.
2 Obrero fijo.
3 Obrero temporal.
4 Obrero fijo especializado.
5 Obrero temporal especializado.
6 Autónomo.
7 Jubilado – rentista.
8 No activo.

El Estado civil de la persona que pide el préstamo se reduce a tres categorías:

1 Casado.
2 Separado.
3 Soltero.

En cuanto a la nacionalidad, debido a la diversidad de países de procedencia que se elevaban a 34, se decide agrupar a todas las nacionalidades no españolas en una única categoría:

1 Español.
2 Extranjero.

La variable Edad se crea a partir de la fecha de nacimiento del peticionario del crédito y se expresa en años.

Debido a que en la variable finalidad del crédito existen muchas modalidades se realiza una agrupación para reducirlas a diez categorías:

206 Reformas viviendas.
301 Compra de automóviles.
302 Compra de electrodomésticos.
303 Compra de ordenador.
305 Compra de mobiliario y decoración.
308 Compra de otros bienes y servicios corrientes.
504 Financiación de servicios sanitarios.
506 Financiación de imprevistos familiares.
999 Otros.

La variable vivienda contienen las siguientes modalidades:

1. Propiedad libre de cargas.
2. Propiedad hipotecada.
3. Alquiler.
4. Domicilio con la familia.
5. Otros.

Al final del proceso de recodificación de variables se utilizan, para la construcción de los modelos, dieciséis variables explicativas de la concesión o negación del crédito y que son las siguientes:

Variables cuantitativas:

NUM_FAMILIA: número de componentes de la unidad familiar.
EDAD: años de la persona que solicita el crédito.
IMPINV: importe de la inversión.
IMPCUO: importe de la cuota del préstamo.
INGRESOS: es la suma de los ingresos anuales fijos, los variables y otros ingreso provenientes de otra fuentes.
SALMEDVINVI: consta del saldo medio a la vista y el saldo medio no vista.
IMPVALVIV: valor neto de viviendas propias.
IMPPAT: patrimonio de la persona que solicita el préstamo.
IMPPMO: importe del préstamo.
IMPORPEN: Importes pendientes de préstamos en la entidad que solicita el préstamo o en otras entidades.

PORCENPRES: Es el porcentaje del valor de lo realmente concedido en relación con lo que se solicita.

Variables cualitativas:

VIVIENDA: modalidades de la vivienda.
NACIONALIDAD: nacionalidad del que solicita el préstamo.
FINALIDAD: destino de la inversión solicitada.
ESTADO_CIVIL: estado civil de la persona que pide el préstamo.
RELACION_LABORAL: situación laboral del solicitante del crédito.
CLASE Toma el valor SÍ, si se ha devuelto el crédito y NO, si no se ha devuelto.

Las variables elegidas fueron posteriormente contrastadas con responsables de otras entidades de crédito, especialmente con técnicos de Caja España que confirmaron que eran las variables que fundamentalmente se utilizaban para la concesión de créditos personales. En los encuentros con diversos técnicos de entidades de crédito se sugiere la creación de dos nuevas variables: la relación entre ingresos y gastos y el porcentaje de lo que realmente se financia en relación con lo solicitado.

Se comprueba que los datos de la variable gastos están ausentes en la mayor parte de los registros con lo que esta variable no se toma en consideración.

4.2.3. Exploración y preprocesado de los datos.

Es bastante obvio que para obtener buenos resultados en los análisis estadísticos y de minería de datos se debe de partir de una base de datos con información consistente, completa, comprensible y limpia para que los análisis sean útiles. Es necesario, por tanto, que los datos sean analizados con conciencia.

La tarea del preprocesado de los datos tiene como objetivo obtener una vista minable, es decir, trata de quedarse con aquel conjunto de datos lo suficientemente libre de errores, ya filtrados, sin datos anómalos y cuyas variables sean lo más adecuadas al proceso de clasificación que se está manejando.

Las distintas tareas que conlleva la exploración y el preprocesado de los datos en un procedimiento de minería de datos, según varios autores, abarca un tiempo considerable, estimado entre el 70% y el 90%, del tiempo total destinado a un proyecto de minería de datos.

Este primer paso inicial es de vital importancia ya que implica realizar labores de limpieza de los datos, imputación, transformación, selección de variables y otro conjunto de tareas sin las cuales es imposible optimizar los métodos estadísticos.

Las tareas de limpieza y transformación relacionadas con la base de datos de credit scoring hasta conseguir un base de datos minable son numerosas, pero las principales se pueden resumir en las siguientes:

- ✓ Imputación de datos ausentes.
- ✓ Filtrado y eliminación de valores anómalos o outlier.
- ✓ Transformación de la Base de datos.
- ✓ Balanceo de la base de datos.
- ✓ Reducción de variables o de la dimensionalidad.
- ✓ Discretización de variables.

4.2.3.1. Imputación de datos ausentes.

En la base de datos pueden existir, por varias razones, un conjunto significativo de valores ausentes, perdidos o faltantes que pueden ser reemplazados. Tanto la fase de detección como en la de tratamiento de la información es muy importante averiguar los motivos de los datos faltantes.

Las principales razones para tratar los valores faltantes son: que el método de minería de datos no funcione bien con estos valores ausentes, o que se vayan a utilizar agregaciones de datos o variables y que estos datos ausentes no nos permitan realizarlas o bien que el método utilizado nos elimine todo el ejemplo o instancia por no encontrar la existencia del dato.

Una vez que se ha establecido las causas de los valores ausentes podemos proceder a realizar alguna de las siguientes acciones

- ✓ Ignorar, es decir, no realizar ningún tratamiento dado que la técnica que vamos a utilizar es consistente con valores ausentes, por ejemplo los árboles de decisión o las redes bayesianas.
- ✓ Eliminar el atributo si la proporción de datos ausentes es elevada. En este caso se elimina toda la columna de la base de datos.
- ✓ Eliminar los ejemplos o instancias donde se encuentran los valores faltantes.
- ✓ Reemplazar el valor. Cuando hacemos esto existen diferente procedimientos para sustituir los valores ausentes: imputación automática de casos perdidos a

través de técnicas de predicción o clasificación, sustitución del valor por otro dato preservando la media o la varianza, por la moda si son valores nominales, etcétera. Existe el algoritmo EM (Expectation Maximization) que se utiliza de forma tradicional para realizar esta operación de sustitución de valores.

4.2.3.2. Filtrado y eliminación de valores extremos u outlier.

Los valores extremos, no usuales o erróneos son aquellos cuya disposición especial es extraña respecto al comportamiento general del conjunto. Estos datos se denominan de forma general como espurios o anómalos y su tratamiento resulta esencial dado que afectan normalmente a las conclusiones y resultados finales de las investigaciones.

Las causas de encontrar valores extremos pueden provenir de sucesos anormales que pueden ser muy interesantes estudiar y un análisis minucioso puede aportarnos información muy valiosa sobre el fenómeno de estudio. Otros outliers son debidos a errores de medición por aparatos mal calibrados, por datos mal introducidos, por defectos en la base de datos o por errores de transmisión o fallos de lectura y conversión o transformación de la información. Si los datos proceden de encuestas pueden provocar muchos valores anómalos debido a formularios incorrectos o mal apuntados o valores que no se han rellenado.

Son muchas las técnicas que se han propuesto para la detección de datos anómalos. Si las muestras de datos están generadas de poblaciones con distribuciones normales multivariantes las siguientes técnicas pueden resultan muy válidas:

- ✓ Técnicas de regresión para estimar el modelo que define los datos para determinar la desviación de los puntos frente al mismo.
- ✓ Utilización de histogramas o gráficos boxplots para detectarlos gráficamente.
- ✓ Mediante el uso de los autovalores de la muestra.
- ✓ Mediante el cálculo de la distancia de Mahalanobis.
- ✓ A través del Análisis de Componentes Principales, Proyección Pursuit.
- ✓ Utilizando análisis cluster.
- ✓ etc.

Si los datos proceden de poblaciones que siguen distribuciones de probabilidad no normales, las siguientes técnicas son apropiadas:

- ✓ Proyección Sammon.

- ✓ Redes de mapas autoorganizados (SOM).
- ✓ Proyectores PCA No Lineales.
- ✓ Generative Topographic Maps (GTM) [DAS03].
- ✓ Otros proyectores No Lineales basados en redes neuronales.
- ✓ Otras técnicas de Visualización Multivariante.
- ✓ Coordenadas paralelas, dendogramas, curvas Andrews, iconos, Radviz, etc.
- ✓ Otros métodos.

El apoyo gráfico resulta fundamental para la detección de outliers:

En cuanto al tratamiento de los datos ausentes podemos utilizar varias estrategias dependiendo de las necesidades del estudio:

- ✓ Utilizar la media, moda, mediana o cualquier otro estimador robusto, o un valor que preserve la desviación estándar de la distribución de la variable, etcétera.
- ✓ Cuando el número de registros es muy numeroso simplemente se pueden eliminar aquellos que tienen datos ausentes.
- ✓ Si son series temporales podemos calcular la media del valor anterior o posterior o realizar una predicción del valor a través de otros métodos (medias móviles, modelos autorregresivos, etcétera)
- ✓ Podemos aplicar otras técnicas avanzadas de estadística o técnicas heurísticas.

4.2.3.3. Transformación de la Base de datos.

En la transformación de la base de datos se incluyen aquellas operaciones que transforman los atributos o bien se derivan nuevos atributos. También cuando las variables transforman el tipo de datos a través de la discretización o numerización o cambian el rango a través del escalado de las variables.

Estandarizar es transformar una variable aleatoria que tiene alguna distribución en una nueva variable aleatoria con distribución normal o aproximadamente normal, restando a todos los datos su media y dividiéndolos por su desviación típica. La nueva distribución tendrá media cero y desviación típica igual a uno.

$$x' = \frac{x - \bar{x}}{\sigma} \tag{4.1}$$

También se puede normalizar una variable de tal forma que el nuevo rango de valores se encuentre entre cero y uno. Lo que se consigue aplicando la siguiente fórmula:

$$x' = \frac{x - mínimo}{máximo - mínimmo} \tag{4.2}$$

El escalado de variables se lleva a cabo al dividir todos los valores de la variable por su valor máximo.

$$x' = \frac{x}{máximo} \tag{4.3}$$

Es muy importante para los procedimientos de minería de datos especificar bien el tipo de los atributos: numérico o nominal.

4.2.3.4. Balanceo de las clases.

Antes de aplicar los diferentes métodos de clasificación a la base de datos de credit scoring hemos de resolver dos cuestiones fundamentales que se abordan a continuación: balanceo de la variable clase y especificar cuál es el conjunto de variables explicativas óptimo para la clasificación. El primero es un problema de equilibrado de la muestra extraída de los clientes y el segundo un problema de selección de variables. Ambas cuestiones son de crucial importancia para el desempeño de los algoritmos de clasificación.

En los cuadros siguientes se presentan, para diversos métodos de clasificación, el porcentaje correctamente clasificado tanto para el total como para cada una de las clases.

Tabla 4.1. Muestra desbalanceada (1.609 instancia clase SI y 177 clase NO).

Técnica	CLASE SÍ (%)	CLASE NO (%)	TOTAL (%)	AREA ROC
Regresión Logística	97,5	31.3	90,9	0,777
C 4.5	97,9	35,2	91,7	0,885
Maq. Vect. Soporte	99,9	0,1	89,9	0,500
Perceptrón Mult.	94,6	35,8	88,7	0,817
Redes Base Radial	100,0	0,0	90,0	0,825
Naïve Bayes	66,4	81,6	68,0	0,832
Red Bayesiana(TAN)	95,4	49,2	90,8	0,885
Red Bayesiana(K2)	88,3	69,8	86,4	0,884
AODE1	91,9	64,8	89,1	0,894
AODE2	94,5	51,4	90,2	0,896
Metaclasificadores				
Bagging	99,3	20,7	91,4	0,879
Adaboost	97,3	39,7	91,5	0,893
Random Forest	98,4	33,5	91,9	0,846
Random Committee	99,8	14,0	91,2	0,891
RandomSubSpace	98,9	38,2	92,3	0,888
STAKING C (5 modelos)	97,7	24,6	90,4	0,772
Decorate	97,1	37,4	91,1	0,860
Metacost 1/1	97,1	43,6	91,7	0,787
Metacost 3/1	94,9	49,2	90,3	0,831
Metacost 9/1	86,9	75,4	85,8	0,828

El tamaño de la muestra juega un papel determinante en la bondad de los modelos de clasificación. Cuando el desbalanceo es considerable descubrir regularidades inherentes a la clase minoritaria se convierte en una tarea ardua y de poca fiabilidad. Japkowicz y Stephen (2002) concluyen que si los dominios son separables linealmente los modelos no son sensibles al problema del desequilibrio de las clases.

En el ejemplo que estamos tratando podemos observar en la tabla 4.1. que cuando mantenemos la base de datos con las clases desequilibradas todos los métodos presentan una importante diferencia de aciertos entre las clases.

Los métodos de clasificación favorecen en general a la clase mayoritaria salvo en el caso del clasificador bayesiano Naïves Bayes que clasifica mejor a la clase minoritaria. Se da el caso extremo en el que un clasificador, las máquinas de vectores soporte, clasifican correctamente a todos de la clase mayoritaria y a ninguno de la minoritaria. Tampoco los metaclasificadores estiman correctamente ambas clases. Solamente introduciendo un método cuyo aprendizaje sea sensible al coste se logra equilibrar la precisión de los ejemplo bien clasificados.

Tabla 4.2. Muestra equilibrada (193 ejemplos para cada clase).

Técnica	%CLASE SÍ	%CLASE NO	% CLASE TOTAL	ROC AREA
Regresión Logística	64,3	91,7	76,9	0,893
C 4.5	80,8	79,6	80,2	0,771
Maq. Vect. Soporte	73,7	75,4	74,6	0,746
Perceptrón Mult.	72,5	73,1	72,8	0,823
Redes Base Radial	75,4	74,3	74,9	0,809
Naïve Bayes	80,2	81,4	80,8	0,881
Red Bayesiana(TAN)	81,4	81,4	81,4	0,873
Red Bayesiana(K2)	80,2	81,4	80,8	0,881
AODE1	80,2	82,0	81,1	0,887
AODE2	79,6	80,8	80,2	0,885
Metaclasificadores				
Bagging	80,2	80,2	80,2	0,860
Adaboost	85,4	77,8	81,4	0,891
Random Forest	82,6	82,0	82,3	0,886
Random Committee	81,4	81,4	81,4	0,874
RandomSubSpace	79,6	83,2	81,4	0,882
STAKING C (5 modelos)	74,5	80,2	77,5	0,749
Decorate	81,4	81,4	81,4	0,816
Metacost	82,0	80,8	81,4	0,810

Las soluciones para tratar el desbalanceo se pueden encuadrar en dos grupos: soluciones a nivel de datos y a nivel de algoritmos.

Las técnicas dirigidas a modificar los datos tratan de remuestrear las tallas de entrenamiento, bien sea a través del sobremuestreo de la clase minoritaria o del submuestreo de la clase que tiene mayores instancias. Aunque estas técnicas han

demostrados su efectividad no dejan de tener ciertos inconvenientes: pueden eliminar ejemplos útiles e incrementar los costes. Otra crítica a esta estrategia se refiere al cambio que se realiza en la distribución original del conjunto de entrenamiento de los datos

En el cuadro nº 2 se expresan los resultados de diferentes clasificadores aplicados a una muestra donde se han balanceado ambas clases. La forma de extraer los registros de la clase más numerosa ha sido aleatoria. Cuando existe equilibrio de las instancias en la base de datos los porcentajes de acierto de los clasificadores para ambas clases están mucho más igualados.

El tema de muestras desbalanceadas se ha tratado extensamente y se han utilizado muchas estrategias, aunque se puede afirmar que no existe una solución concluyente sobre qué solución es mejor. Hulse et al. (2007) concluyen que la decisión sobre la mejor técnica está influenciada en gran medida por la naturaleza del clasificador y la medida de efectividad.

Otra forma que disponemos para combatir el desbalance de clases, es a través del establecimiento de una matriz de costes, lo que se ha llamado método del costo-sensitivo (cost-sensitive). Este método se basa en la aseveración de que el precio de cometer un error de clasificación debe ser distinto para cada clase. Es evidente que en este ejemplo no es lo mismo conceder un crédito y no pagarlo que no concederlo cuando se debería haber concedido.

En este trabajo el clasificador que se aplica para poder comparar con el resto de los algoritmos es el metacost (Domingos, 1999). El objetivo de este procedimientos es reetiquetar cada muestra de entrenamiento por la estimación del riesgo de Bayes. Finalmente, el clasificador se entrena con un método no basado en costes con el conjunto que ya ha sido reetiquetado.

De los 1.575 ejemplos disponibles que devolvieron el crédito se han seleccionado 312 registros a través del método del cubo. Para esta selección de los individuos las variables auxiliares utilizadas por el método del cubo han sido el estado civil, la nacionalidad, el tipo de trabajo, las condiciones de la casa y el tipo de trabajo de las personas que solicitan el crédito. El número de muestras que ha considerado este método para llegar a la solución más idónea ha sido de 77.250 muestras. En el cuadro que sigue se presenta, para la muestra elegida, los totales y los estimadores de Horvitz-Thompson (que dependen de la muestra), así como los errores absolutos y relativos, en porcentaje, entre ambos para cada variable de equilibrio.

La base de datos que finalmente se utiliza es la combinación resultante de la aplicación del método del cubo y del aumento de los registros de la clase más desfavorecida a través del método conocido como SMOTE

Los resultados de la tabla 4.3 se han obtenido con el programa R cuya programación se encuentra en el Anexo nº 1.

Tabla 4.3. Resultados del submuestreo equilibrado. Método del cubo.

VARIABLES	Totales	Estimador Horvitz Thompson	Desviación absoluta	Desviación relativa
UNO	1.609	1.609	0,00	0,00
CASADO	884	882	2,42	-0,27
SEPARADO	86	87	-0,71	0,83
SOLTERO	639	641	-1,71	0,27
ESPAÑOL	1.445	1.445	-0,21	0,01
EXTRANJERO	164	164	0,21	-0,13
LIBRE	497	496	0,81	-0,16
HIPOTECA	609	607	2,01	-0,33
ALQUILER	138	135	3,11	-2,26
FAMILIA	300	303	-3,49	1,16
OTRAS	65	67	-2,44	3,76
TECNICO	435	434	1,44	-0,33
OBRERO_FIJO	476	472	3,90	-0,82
OBRERO_TEMPORAL	159	159	0,03	-0,02
OBRERO_ESP_FIJO	161	164	-2,79	1,73
OBRERO_ESP_TEMPORAL	28	29	-0,90	3,23
AUTONOMO	155	154	0,84	-0,54
JUBILADO_RENTISTA	105	106	-0,98	0,94
NO_ACTIVO	90	92	-1,53	1,70

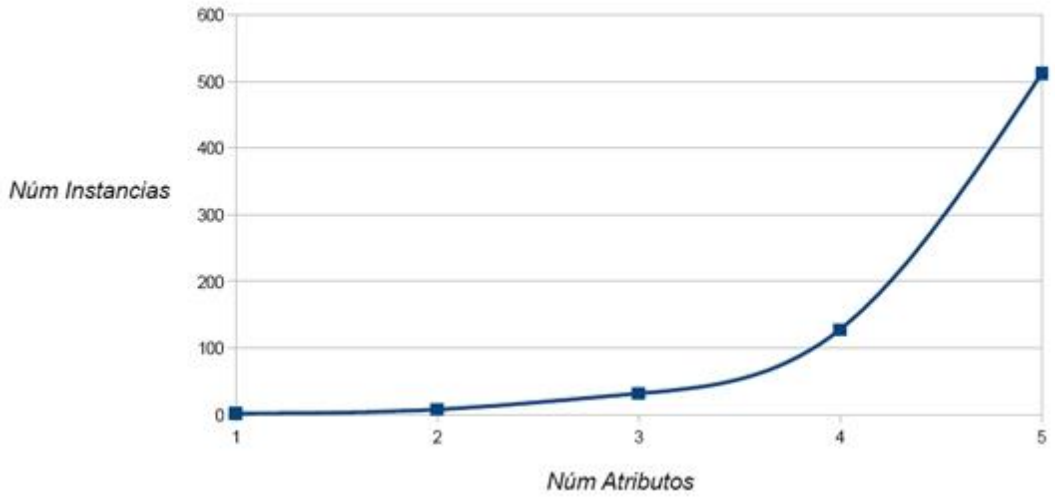
Fuente: Elaboración propia.

4.2.3.5. Reducción de variables o de la dimensionalidad.

La selección de las variables que van a formar parte del fichero inicial es una fase vital y transcendente de la minería de datos. El alto número de variables recogidas para el estudio de un fenómeno a veces es un problema para el aprendizaje si el número de

instancias o ejemplos de la muestra es reducido. Este es el problema conocido como la maldición de la multidimensionalidad.

Figura 4.2. La maldición de la dimensionalidad.



La selección de atributos es uno de los problemas más complejos al que pretende hacer frente el aprendizaje automático. El objetivo es eliminar variables redundantes, atributos espurios y, en general, todos aquellas variables donde su presencia en la base de datos no aporte un aumento de la información de la misma. En bases de datos con muchas características y pocas instancias resulta imposible construir modelos.

En el ejemplo que utiliza esta tesis doctoral donde el número de instancias o ejemplos de la muestra en una de las clases es reducido es de vital importancia reducir las variables del modelo lo que, por otra parte hará más fácil entender las relaciones existentes en las variables que explican la concesión o no de los créditos solicitados

Existen diferentes estrategias a la hora de realizar una selección de variables.

En la literatura de selección de variables existen dos métodos generales para escoger las mejores características de la base de datos: métodos de filtro y métodos basados en modelos. En los primeros se filtran los atributos irrelevantes antes de aplicar las técnicas de minería de datos. El criterio que establece las variables óptimas se basa en una medida de calidad que se calcula a partir de los datos mismos. En los métodos basados en modelos, también conocidos como métodos de envolvente o wrapper, la bondad de la selección de las variables se evalúa a través de un modelo utilizando, lógicamente, un método de validación.

En el caso de la selección de atributos debemos definir un algoritmo que evaluará cada atributo individualmente del conjunto de datos inicial, que se denomina “attribute evaluator” y un método de búsqueda que realizará una búsqueda en el espacio de posibles combinaciones de todos los subconjuntos del conjunto de atributos.

De esta forma podremos evaluar independientemente cada una de las combinaciones de atributos y, con ello, seleccionar aquellas configuraciones de atributos que maximicen la función de evaluación de atributos.

Para resolver el problema de plantear combinaciones de atributos, la función que evalúa cada subconjunto de atributos es utilizar un algoritmo de búsqueda que recorre el espacio de posibles combinaciones de una forma organizada, o adecuada al problema.

Habitualmente en las situaciones en la que se emplea selección de atributos no es posible hacer un recorrido exhaustivo en el espacio de combinaciones por lo que la selección adecuada de un algoritmo de búsqueda resulta crítica.

Además del método de las componentes principales existen dos tipos de evaluadores: evaluadores de subconjuntos o selectores (SubSetVal) y prorrateadores de atributos (AttributeEval).

Los SubSetVal necesitan una estrategia de búsqueda (Search Method) y los AttributeEval ordenan las variables según su relevancia, así que necesitan un Ranker.

Para seleccionar las variables de mayor relevancia se utilizaron varios métodos de selección de atributos disponibles en el programa WEKA (Waikato Environment for Knowledge Analysis). Para esta base de datos se utilizan, en primer lugar los Filtros, donde se seleccionan y evalúan los atributos en forma independiente del algoritmo de aprendizaje. En este caso se emplean dos algoritmos evaluadores de atributos, el “CfsSubsetEval” y el ConsistencySubsetEval. El primer algoritmo es el más sencillo, ya que puntúa a cada atributo en función de su entropía. Como algoritmos de búsqueda se utilizan cuatro métodos: Best First, algoritmos genéticos, Greedy y Tabu Search.

CfsSubsetEval evalúa un subconjunto de atributos considerando la habilidad predictiva individual de cada variable, así como el grado de redundancia entre ellos. Se prefieren los subconjuntos de atributos que estén altamente correlacionados con la clase y tengan baja intercorrelación entre ellos.

ConsistencySubsetEval: Evalúa un subconjunto de atributos por el nivel de consistencia en los valores de la clase al proyectar las instancias de entrenamiento sobre el subconjunto de atributos.

Las variables seleccionadas a través de estos dos procedimientos se encuentran recogidas en las tablas 4.4 y 4.5.

Tabla 4.4. Selección de variables a través del Atributo evaluador CFSSubEval.

Best First	Genetic Search	Greedy	Tabu Search
NUM_FAMILIA	NUM_FAMILIA	NUM_FAMILIA	NUM_FAMILIA
VIVIENDA	VIVIENDA	VIVIENDA	VIVIENDA
IMPVALVIV	IMPVALVIV	IMPVALVIV	IMPVALVIV
NACIONALIDAD	NACIONALIDAD	NACIONALIDAD	NACIONALIDAD
IMPINV	IMPPMO	IMPINV	IMPINV
IMPCUO	IMPCUO	IMPCUO	IMPCUO
SALDOMEDVINVI	SALDOMEDVINVI	SALDOMEDVINVI	SALDOMEDVINVI
TIPO_TRABAJO	TIPO_TRABAJO	TIPO_TRABAJO	TIPO_TRABAJO

Tabla 4.5. Selección de variables a través del Atributo evaluador ConsistencySubsetEval.

Best First	Genetic Search	Greedy	Tabu Search
NUM_FAMILIA	NUM_FAMILIA	NUM_FAMILIA	NUM_FAMILIA
VIVIENDA	VIVIENDA	VIVIENDA	VIVIENDA
IMPVALVIV	IMPVALVIV	IMPVALVIV	IMPVALVIV
NACIONALIDAD	IMPPAT	NACIONALIDAD	NACIONALIDAD
IMPPMO	NACIONALIDAD	IMPPMO	IMPPMO
IMPINV	IMPPMO	IMPINV	IMPINV
IMPCUO	IMPINV	IMPCUO	IMPCUO
FINALIDAD	IMPCUO	FINALIDAD	FINALIDAD
SALDOMEDVINVI	FINALIDAD	SALDOMEDVINVI	SALDOMEDVINVI
PORCENPRES	INGRESOS	PORCENPRES	PORCENPRES
CIVIL	IMPORTEPEN	CIVIL	CIVIL
TIPO_TRABAJO	SALDOMEDVINVI	TIPO_TRABAJO	TIPO_TRABAJO
	CIVIL		
	TIPO_TRABAJO		

En segundo lugar recurrimos al método Ranker para que nos facilite una lista ordenada de los atributos atendiendo a su calidad:

1. ChiSquaredAttributeEval: calcula el valor estadístico Chi-cuadrado de cada atributo con respecto a la clase y así obtiene el nivel de correlación entre la clase y cada atributo.
2. GainRatioAttributeEval: evalúa cada atributo midiendo su razón de beneficio con respecto a la clase.
3. InfoGainAttributeEval: evalúa los atributos midiendo la ganancia de información de cada uno con respecto a la clase. Previamente discretiza los atributos numéricos.
4. OneRAttributeEval: evalúa la calidad de cada atributo utilizando el clasificador OneR, el cual usa el atributo de mínimo error para predecir, discretizando los atributos numéricos.

La ordenación con la estrategia de búsqueda Ranker para todos los métodos de selección de atributos se recogen en el cuadro 4.6.

Tabla 4.6. Método Ranker con diferentes evaluadores.

ChiSquared		OneR		GainRatio		InfoGain	
Media	**Variable**	**Media**	**Variable**	**Media**	**Variable**	**Media**	**Variable**
206,5	IMPINV	74,4	IMPINV	0,171	NACIONALIDAD	0,247	IMPINV
198,6	VIVIENDA	72,9	IMPVALVIV	0,163	IMPVALVIV	0,241	VIVIENDA
178,8	SALDOMEDVINVI	71,9	SALDOMEDVINVI	0,156	SALDOMEDVINVI	0,216	SALDOMEDVINVI
171,7	IMPVALVIV	71,4	NUM_FAMILIA	0,122	IMPPMO	0,203	IMPVALVIV
156,0	IMPCUO	71,1	VIVIENDA	0,114	IMPINV	0,194	IMPCUO
149,4	IMPPMO	70,8	IMPPMO	0,113	VIVIENDA	0,171	TIPO_TRABAJO
146,6	TIPO_TRABAJO	70,5	NACIONALIDAD	0,106	IMPCUO	0,171	IMPPMO
132,1	NACIONALIDAD	70,1	TIPO_TRABAJO	0,095	NUM_FAMILIA	0,152	NACIONALIDAD
91,2	NUM_FAMILIA	67,7	CIVIL	0,084	EDAD	0,123	NUM_FAMILIA
89,0	CIVIL	66,0	IMPCUO	0,082	CIVIL	0,099	CIVIL
72,6	FINALIDAD	63,6	EDAD	0,076	INGRESOS	0,083	FINALIDAD
60,4	EDAD	62,6	FINALIDAD	0,069	TIPO_TRABAJO	0,068	EDAD
47,4	INGRESOS	58,1	INGRESOS	0,057	IMPPAT	0,054	INGRESOS
32,7	IMPORTEPEN	56,4	IMPORTEPEN	0,0394	IMPORTEPEN	0,036	IMPORTEPEN
26,9	IMPPAT	55,7	IMPPAT	0,0317	PORCENPRES	0,031	IMPPAT
17,3	PORCENPRES	52,7	PORCENPRES	0,0301	FINALIDAD	0,019	PORCENPRES

En la mayor parte de los métodos de selección de variables hay cuatro variables: PORCENPRES (relación entre lo solicitado y lo concedido), IMPPAT (Importe del patrimonio), IMPORPEN (Importes pendientes), INGRESOS, EDAD y FINALIDAD que se sitúan como las variables menos explicativas en la mayor parte de los métodos donde se emplean un Ranker

Tabla 4.7. Selección de variables. Diferentes modelos y métodos de búsqueda.

Best First	Genetic Search	Greedy	Tabu Search
ÁRBOL DE DECISIÓN			
NUM_FAMILIA	VIVIENDA	NUM_FAMILIA	NUM_FAMILIA
NACIONALIDAD	IMPPAT	NACIONALIDAD	NACIONALIDAD
IMPPMO	NACIONALIDAD	IMPPMO	IMPPMO
IMPINV	IMPINV	IMPINV	IMPINV
IMPCUO	IMPCUO	IMPCUO	IMPCUO
	INGRESOS		
	IMPORTEPEN		
	PORCENPRES		
	TIPO_TRABAJO		
RED BAYESIANA			
NUM_FAMILIA	NUM_FAMILIA	NUM_FAMILIA	IMPVALVIV
VIVIENDA	VIVIENDA	VIVIENDA	NACIONALIDAD
IMPVALVIV	NACIONALIDAD	IMPVALVIV	IMPINV
NACIONALIDAD	IMPPMO	NACIONALIDAD	IMPCUO
IMPPMO	IMPINV	IMPPMO	SALDOMEDVINVI
IMPINV	IMPCUO	IMPINV	EDAD
IMPCUO	FINALIDAD	IMPCUO	TIPO_TRABAJO
SALDOMEDVINVI	SALDOMEDVINVI	SALDOMEDVINVI	
EDAD	EDAD	EDAD	
TIPO_TRABAJO		TIPO_TRABAJO	
REGRESION LOGÍSTICA			
IMPVALVIV	NUM_FAMILIA	IMPVALVIV	IMPVALVIV
NACIONALIDAD	VIVIENDA	NACIONALIDAD	NACIONALIDAD
SALDOMEDVINVI	IMPVALVIV	SALDOMEDVINVI	SALDOMEDVINVI
TIPO_TRABAJO	IMPPAT	TIPO_TRABAJO	TIPO_TRABAJO
	IMPPMO		
	FINALIDAD		
	INGRESOS		
	IMPORTEPEN		
	SALDOMEDVINVI		
	TIPO_TRABAJO		
VECTORES SOPORTE			
VIVIENDA	VIVIENDA	NUM_FAMILIA	VIVIENDA
IMPVALVIV	IMPVALVIV	VIVIENDA	IMPVALVIV
IMPPAT	IMPPMO	IMPVALVIV	IMPPAT
NACIONALIDAD	IMPCUO	NACIONALIDAD	NACIONALIDAD
IMPPMO	FINALIDAD	IMPPMO	IMPPMO
IMPINV	INGRESOS	FINALIDAD	IMPINV
FINALIDAD	TIPO_TRABAJO	SALDOMEDVINVI	FINALIDAD
SALDOMEDVINVI		TIPO_TRABAJO	SALDOMEDVINVI
PORCENPRES			PORCENPRES
TIPO_TRABAJO			TIPO_TRABAJO
REDES NEURONALES			
IMPVALVIV	IMPVALVIV	IMPVALVIV	IMPVALVIV
IMPPAT	NACIONALIDAD	NACIONALIDAD	NACIONALIDAD
NACIONALIDAD	IMPPMO	IMPPMO	IMPPMO
IMPPMO	IMPCUO	IMPCUO	IMPCUO
IMPCUO	SALDOMEDVINVI	PORCENPRES	PORCENPRES
SALDOMEDVINVI	PORCENPRES	TIPO_TRABAJO	TIPO_TRABAJO
PORCENPRES			

Los métodos de selección de variables utilizando Wrappers (envoltorios) usan el desempeño de algún clasificador (algoritmo de aprendizaje) para determinar el conjunto de atributos óptimos (WrapperSubsetEval). Este procedimiento emplea

validación cruzada para estimar la exactitud del esquema de aprendizaje en cada conjunto.

Se han utilizado los cuatro métodos de búsqueda de los subconjuntos de variables empleados en los métodos anteriores y aplicados a cuatro métodos de clasificación: Árboles de decisión, Redes bayesianas, Regresión logística, Máquinas de vectores soporte y Redes neuronales. Las variables seleccionadas por los diferentes modelos y métodos de búsqueda están recogidas en la tabla 4.7.

Otras formas de abordar la selección de variables es a través de los algoritmos genéticos. Esta técnica propuesta por Holland, (1975), suponen uno de los enfoques más originales en la minería de datos, se inspiran en el comportamiento natural de la evolución, para ello se codifica cada uno de los casos de prueba como una cadena binaria (que se asemejaría a un gen). Esta cadena se replica o se inhibe en función de su importancia, determinada por una función denominada de ajuste o fitness.

Los algoritmos genéticos son adecuados para obtener buenas aproximaciones en problemas de búsqueda, aprendizaje y optimización (Marczyk, 2004).

Otra solución que nos parece más óptima y adecuada a este problema en cuanto a la búsqueda del conjunto óptimo de variables utilizadas en la aplicación de los modelos y algoritmos de clasificación es seleccionar los atributos para la clasificación a través de los resultados observados en el manto de Markov.

La envolvente de Markov para una variable representa el conjunto de variables de las que depende dicha variable. Así si aplicamos la envolvente o manto de Markov a una red bayesiana, definida esta envolvente como:

$$(\text{Padres}(X) \cup \text{Hijos} \cup \text{Padres}(\text{Hijos}(X))) \qquad (4.4)$$

En esta tesis doctoral de las dieciséis variables originales se han reducido a once dado que cinco de ellas no contienen información relevante una vez conocidas el resto de variables.

4.2.3.6. Discretización de variables.

En muchas ocasiones hay que convertir los valores numéricos en nominales (discretización) o un valor nominal en numérico (numerización). En general, la discretización se realiza cuando el error de medida es grande o queremos expresar las conclusiones en ciertos umbrales significativos. También debemos discretizar cuando

existen escalas diferentes. Otras veces crear intervalos es imperativo dado que algunos procedimientos necesitan que las variables numéricas sean transformadas. Dado que este proceso tiene un especial interés para algunos algoritmos utilizados en esta tesis han sido expuestas en un apartado independiente.

Es muy importante escoger el mejor método para discretizar las variables numéricas en la utilización de algunos métodos de minería de datos como las redes bayesianas o los árboles de decisión. En el capítulo anterior, en el epígrafe 3.1.4. se dedica una extensa exposición a explicar los principales métodos que existen en la literatura. En esta tesis se utiliza el criterio de Longitud de Descripción Mínima desarrollado por Suzuki (1996).

Los intervalos de las variables que se generan a través de este método se muestran en la figura 4.3.

Tabla 4.8. Intervalos de las variables. Método MDL.

Variables	**SÍ**	**NO**	**Total**	**Variables**	**SÍ**	**NO**	**Total**
Mienbros familia				**Importe cuota**			
Uno	93	136	229	Menos de 41	39	3	42
dos	1	58	59	Entre 41 y 81	36	24	60
Mas de dos	243	143	386	Entre 82 y 241	114	254	368
Valor vivienda				Entre 242 y 250	73	4	77
Menos de 27.112	111	275	386	Mayor de 250	77	54	131
Más de 27.112	225	61	286	**Ingresos**			
Importe préstamo				Menos de 23.210	236	306	542
Menos de 1.471	56	30	86	Más de 23.210	100	30	130
Entre 1.471 y 5.973	47	199	246	**Importes pendientes**			
Más de 5.973	234	108	342	Meno de 4.150	193	262	455
Saldo bancario				Mas de 4.150	143	74	217
Menos de 453	111	274	385	**Importe inversión**			
Entre 454 y 4.543	131	57	188	Menos de 565	1	9	10
más de 4.543	95	6	101	Entre 565 y 1.484	56	15	71
Edad				entre 1.485 y 5.999	40	194	234
Menos de 50	208	295	503	Entre 6.000 y 8.712	43	50	93
Más de 50	128	41	169	Más de 8.712	199	71	270

4.2.4. Análisis de los modelos predictivos.

Existen multitud de modelos tanto paramétricos como no paramétricos para llevar a cabo una clasificación de instancias. Los principales modelos utilizados en esta tesis se han descrito muy brevemente en el capítulo uno y más extensamente en el capítulo

tres. En el capítulo cinco se ofrecen los diferentes resultados de los algoritmos empleados junto con algunas medidas de calidad de las estimaciones que arrojan los diferentes métodos.

4.2.4.1. Evaluación de los modelos.

Existen diferentes formas de abordar la evaluación de modelos tal y como ha quedado expuesto en el capítulo anterior en el epígrafe 3.1.6. donde se han explicado tres métodos de evaluación de modelos: basados en métricas, en curvas ROC y a través de matrices de costes.

Una vez elegido un método de evaluación es fundamental poder disponer de procedimientos estadísticos que nos permitan realizar una comparación de los algoritmos con el objetivo de elegir aquél que mejor discrimine entre los clientes que no devuelven el crédito que se les otorgó y los que sí lo hicieron.

El contraste entre los diferentes algoritmos de clasificación lo podemos realizar a través de la prueba t por parejas mediante validación cruzada, que es la que se utiliza en todos los procedimientos en esta tesis doctoral. También podemos realizar análisis de la varianza

K-fold Cross Validation paired t Test

$$H_0 : \mu = 0 \tag{4.5}$$

$$H_1 : \mu \neq 0 \tag{4.6}$$

El error medio de la clasificación tiene la siguiente expresión:

$$m = \frac{1}{k}\sum_{i=1}^{K} p_i \tag{4.7}$$

Y la varianza como:

$$S^2 = \frac{1}{k-1}\sum_{i=1}^{K} (p_i - m)^2 \tag{4.8}$$

Se construye el siguiente estadístico m

$$\frac{\sqrt{k}(m-\mu)}{S} = \frac{\sqrt{K}m}{S} \approx t_{K-1} \tag{4.9}$$

Este estadístico **m** sigue una distribución muestral t de Student con K -1 grado de libertad.

Aceptamos la hipótesis nula si el estadístico obtenido cae dentro del intervalo de la distribución t para K – 1 grado de libertad:

$$t_{K-1} \in (-t_{\alpha/2,K-1}, t_{\alpha/2,K-1}) \tag{4.10}$$

Análisis de la varianza

La hipótesis nula es que los clasificadores presentan los mismos errores de predicción:

$$H_0 : \mu_1 = \mu_2 = \ldots = \mu_L \tag{4.11}$$

Definimos la media de los errores cometidos por el clasificador j en el fold i (X_{ij}) como sigue:

$$m_j = \frac{1}{K}\sum_{i=1}^{K} X_{ij} \tag{4.12}$$

La media y la varianza de m_j toma las siguientes expresiones:

$$m = \frac{1}{L}\sum_{j=1}^{L} m_j \qquad S^2 = \frac{1}{L-1}\sum_{j=1}^{L}(m_j - m)^2 \tag{4.13}$$

Un estimador de la varianza poblacional es el siguiente:

$$\hat{\sigma} = KS^2 \tag{4.14}$$

La suma de cuadrados Inter-Grupo (SSb) se define como:

$$SSb = K\sum_{j}(m_j - m)^2 \tag{4.15}$$

Si realizamos el cociente entre la varianza entre grupos y el estimador de la varianza poblacional obtenemos un estadístico que se distribuye como una distribución de Pearson con L – 1 grado de libertad.

$$\frac{SSb}{\hat{\sigma}^2} \sim \chi^2_{L-1} \tag{4.16}$$

Ahora definimos la Suma de cuadrados Intra Grupos (SSw) de esta forma:

$$SSw = \frac{1}{K-1}\sum_{i=1}^{K}(X_{ij} - m_j)^2 \tag{4.17}$$

$$\hat{\sigma}^2 = \frac{1}{L}\sum_{j}^{L} S_j^2 \tag{4.18}$$

$$SSw = \sum_{j}\sum_{i}(X_{ij} - m_j)^2 \tag{4.19}$$

Ahora podemos construir el estadístico que nos sirva para contrastar la hipótesis nula:

$$\frac{SSb/(L-1)}{SSw/(L(k-1))} \sim F_{L-1,L(K-1)} \tag{4.20}$$

La evaluación de los diferentes algoritmos estudiados en esta tesis se realiza a través del programa WEKA. A la hora de efectuar contrastes este software de minería de datos utiliza el test estadístico de la t de Student al que se le aplica una corrección realizada por Nadeau y Bengio (2001).

4.2.5. Gestión del modelo de conocimiento.

Una vez construido y validado el modelo, es decir, cubiertas las diferentes fases metodológicas para encontrar al mejor modelo clasificador atendiendo a los requerimientos iniciales, se ha construido una aplicación informática que permite facilitar a los gestores bancarios la decisión de conceder o no un crédito a los posibles solicitantes del mismo.

El código de la aplicación ha sido realizado en lenguaje JAVA y se encuentra en el anexo dos. Esta aplicación informática se ofrece en esta tesis doctoral para que los técnicos de la Caja de Ahorro y el resto de investigadores puedan aplicar diferentes algoritmos de clasificación a la vez de poder también realizar procedimientos de simulación.

CAPÍTULO 5

APLICACIÓN DE SCORING CON DATOS DE UNA CAJA DE AHORROS.

5. Aplicación de scoring con datos de una Caja de Ahorros.

5.1. Análisis descriptivo de la base de datos

El análisis descriptivo es una fase inicial que nos aporta una descripción de las variables que nos ofrece valiosas pistas para optimizar posteriormente los modelos a través de un conjunto de medidas estadísticas, análisis de correlaciones, contrastes de normalidad y métodos gráficos.

El análisis estadístico del conjunto de datos que contiene dieciséis variables explicativas se realiza con cada una de las cinco variables categóricas y el resto de las once restantes que son numéricas se realiza de forma conjunta. De las variables explicativas categóricas se presenta la tabla de contingencia con la variable clase (devuelve o no devuelve el crédito) y de las variables cuantitativas se ofrece información sobres sus medidas estadísticas de posición, dispersión, asimetría y curtosis y el contraste de normalidad, así como el histograma y el gráfico Q-Q- plot.

El conjunto de registros aportados por la Caja de Ahorros es de 1.788 peticionarios de créditos de los que 179 no han devuelto el crédito mientras que 1.609 sí lo han satisfecho.

Como se puede observar en la tabla 5.1 el número de morosos de la caja de ahorros alcanza el 10%. Cuando se desglosa esta información atendiendo a la ocupación resulta que el grupo de obreros con trabajo temporal alcanza el 29,3% de morosidad, mientras que los dos colectivos que son más cumplidores, técnicos - mandos intermedios y jubilados - rentistas tienen tasas de default del 2,0% y del 4,5% respectivamente.

Tabla 5.1. Estado de la devolución del crédito según el tipo de trabajo.

	Devolución del crédito					
	NO			SÍ		
	Recuento	%de la fila	%del N de la columna	Recuento	%de la fila	%del N de la columna
Técnico-mando intermedio	9	2,0%	5,0%	435	98,0%	27,0%
Obrero fijo	58	10,9%	32,4%	476	89,1%	29,6%
Obrero temporal	66	29,3%	36,9%	159	70,7%	9,9%
Obrero especializado fijo	8	4,7%	4,5%	161	95,3%	10,0%
Obrero especializado temporal	4	12,5%	2,2%	28	87,5%	1,7%
Autónomo	20	11,4%	11,2%	155	88,6%	9,6%
Jubilado-rentista	5	4,5%	2,8%	105	95,5%	6,5%
No activo	9	9,1%	5,0%	90	90,9%	5,6%
Total	179	10,0%	100,0%	1.609	90,0%	100,0%

En relación al tipo de vivienda donde residen los peticionarios del crédito son los que habitan en régimen de alquiler los que presentan los datos más elevados de morosidad, 36,7%, seguidos de los que viven en otras situaciones no consideradas 20,7% y de las que se alojan con la familia 13,3%.

Tabla 5.2. Estado de la devolución del crédito según el tipo de vivienda.

	Devolución del crédito					
	NO			SÍ		
	Recuento	%de la fila	%del N de la columna	Recuento	%de la fila	%del N de la columna
Propiedad libre de cargas	6	1,2%	3,4%	497	98,8%	30,9%
Propiedad hipotecada	30	4,7%	16,8%	609	95,3%	37,8%
Alquiler	80	36,7%	44,7%	138	63,3%	8,6%
Domicilio con la familia	46	13,3%	25,7%	300	86,7%	18,6%
Otros	17	20,7%	9,5%	65	79,3%	4,0%
Total	179	10,0%	100,0%	1.609	90,0%	100,0%

Respecto a la nacionalidad existen claras diferencias de comportamiento en relación con la devolución del crédito. Mientras que los españoles devuelven el crédito en el 94,8% de las veces, la cifra se reduce al 62,1% en el colectivo de extranjeros. De los 264 extranjeros contemplados en la muestra que solicitan un crédito, 100 son morosos.

Tabla 5.3. Estado de la devolución del crédito según la nacionalidad.

	Devolución del crédito					
	NO			SÍ		
	Recuento	%de la fila	%del N de la columna	Recuento	%de la fila	%del N de la columna
Español	79	5,2%	44,1%	1445	94,8%	89,8%
Extranjero	100	37,9%	55,9%	164	62,1%	10,2%
Total	179	10,0%	100,0%	1.609	90,0%	100,0%

El análisis de la concesión de créditos según la finalidad nos revela que el mayor número de créditos personales solicitados se relacionan con la compra de automóviles, 32,77%, seguido de préstamos relacionados con las reformas de la vivienda, 19,35% y, en tercer lugar, destaca la financiación de imprevistos familiares, 11,52%. Es esta categoría la que más morosidad presenta, 22,5%. También destacan por sus elevadas tasas de incumplimiento el grupo de pide dinero prestado para destinarlo a imprevistos familiares, 16,1% y el grupo que dedica el dinero a Otros bienes y servicios corrientes, 12,5%.

Tabla 5.4. Estado de la devolución según la finalidad del crédito.

	Devolución del crédito					
	NO			SÍ		
	Recuento	%de la fila	%del N de la columna	Recuento	%de la fila	%del N de la columna
Reformas viviendas	22	6,4%	12,3%	324	93,6%	20,1%
Compra de automóviles	49	8,4%	27,4%	537	91,6%	33,4%
Compra de electrodomésticos	4	6,1%	2,2%	62	93,9%	3,9%
Compra de ordenador	9	4,5%	5,0%	191	95,5%	11,9%
Mobiliario y decoración	9	8,4%	5,0%	98	91,6%	6,1%
Otros bienes y servicios corrientes	8	12,5%	4,5%	56	87,5%	3,5%
Finaciación servicios sanitarios	10	16,1%	5,6%	52	83,9%	3,2%
Financiación de estudios	1	1,7%	,6%	57	98,3%	3,5%
Financiación imprevistos familiares	46	22,5%	25,7%	158	77,5%	9,8%
Otros	21	22,1%	11,7%	74	77,9%	4,6%
Total	179	10,0%	100,0%	1.609	90,0%	100,0%

Respecto al estado civil son los casados los mejores representados en la muestra que representan más de la mitad de los solicitantes de créditos, 51,62%. Mientras que el colectivo de separados tan sólo representa un 5,7%.

En cuanto al incumplimiento de la obligación de devolver el crédito solicitado son las personas casadas las más cumplidoras con un 95,8%. Las personas que se encuentran separadas de su pareja así como los solteros presentan tasas de impago superiores a la media del 15,7% y del 16,3% respectivamente.

Tabla 5.5. Estado de la devolución según el estado civil.

	Devolución del crédito					
	NO			SÍ		
	Recuento	%de la fila	%del N de la columna	Recuento	%de la fila	%del N de la columna
Casado	39	4,2%	21,8%	884	95,8%	54,9%
Separado	16	15,7%	8,9%	86	84,3%	5,3%
Soltero	124	16,3%	69,3%	639	83,7%	39,7%
Total	179	10,0%	100,0%	1.609	90,0%	100,0%

Los resultados para las variables cuantitativas se han resumido en las tablas de 5.6 a 5.8 donde se recogen diferentes valores estadísticos que sintetizan la información. Para una mayor comprensión de estas variables se ha añadido otra información complementaria en el anexo número tres: intervalos de confianza, estimadores

robustos, percentiles, valores extremos, histograma y gráfico Q-Q normal para cada una de las variables.

Como se observa en la tabla 5.6 prácticamente todas las variables cuantitativas presentan valores muy elevados en el coeficiente de asimetría y en el de curtosis, lo que ya nos indican lo alejados que se encuentran de los valores de una distribución normal. La contrastación de la normalidad de las variables, que se muestra en la tabla 5.7, realizada a través del estadístico de Kolmogorov-Smirnov y del propuesto por Shapiro-Wilk señalan que ninguna de las variables consideradas sigue una distribución estadística normal.

Tabla 5.6. Estadísticos descriptivos de las variables numéricas.

	Media		Asimetría		Curtosis	
	Valor	Desv. Típica	Valor	Desv. Típica	Valor	Desv. Típica
Nº miembros familiares	2,5	1,3	0,597	0,058	0,3	0,116
Valor de la vivienda	93.647,3	108.724,5	2,347	0,058	13,4	0,116
Importe del patrimonio	9.646,9	40.669,3	7,370	0,058	71,2	0,116
Importe del préstamo	8.395,9	6.648,6	1,633	0,058	6,0	0,116
Importe de la inversión	10.559,8	12.085,3	7,073	0,058	92,5	0,116
Importe de la cuota	194,6	163,6	11,875	0,058	295,6	0,116
Ingresos	18.935,4	11.248,6	1,922	0,058	6,9	0,116
Gastos	2.781,8	3.476,9	3,030	0,058	19,2	0,116
Importes pendientes	26.474,0	52.347,7	4,330	0,058	34,5	0,116
Saldo medio	4.884,6	14.289,5	10,371	0,058	186,2	0,116
Edad	43,9	12,3	0,239	0,058	-0,5	0,116
Porcentaje prestado	121,7	88,8	14,112	0,058	261,9	0,116

La media de los miembros de la familia está en 2,5 personas. Esta variable junto con la edad del peticionario del crédito, cuya edad promedio se acerca los cuarenta años, son las que más se aproximan a una distribución normal.

Las dos variables explicativas más asimétricas del colectivo que solicitan créditos personales son el importe del pago de las cuotas y el saldo medio que se mantiene en cuenta, cuyos coeficiente de asimetría alcanzan valores elevados del 11,9 y del 10,4 respectivamente.

En cuanto a la mayor dispersión de las variables, ésta se presenta de forma más más marcada en el importe del patrimonio cuyo coeficiente de variación (desviación estándar como porcentaje de la media aritmética) es del 4,2%, en el en el saldo medio, 2,9% y en los importes pendientes, 2,0%.

Los valores más elevados de curtosis se dan fundamentalmente en el importe de la cuota, 295,6, en el porcentaje prestado, 261,9 y en el saldo medio, 186,2.

Tabla 5.7. Pruebas de normalidad para las variables numéricas.

	Kolmogorov-Smirnov (*)			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Nº miembros familiares	0,184	1.788	0,000	0,873	1.788	0,000
Valor de la vivienda	0,195	1.788	0,000	0,794	1.788	0,000
Importe del patrimonio	0,429	1.788	0,000	0,252	1.788	0,000
Importe del préstamo	0,170	1.788	0,000	0,878	1.788	0,000
Importe de la inversión	0,199	1.788	0,000	0,599	1.788	0,000
Importe de la cuota	0,161	1.788	0,000	0,595	1.788	0,000
Ingresos	0,124	1.788	0,000	0,867	1.788	0,000
Gastos	0,212	1.788	0,000	0,746	1.788	0,000
Importes pendientes	0,307	1.788	0,000	0,557	1.788	0,000
Saldo medio	0,356	1.788	0,000	0,330	1.788	0,000
Edad	0,063	1.788	0,000	0,985	1.788	0,000
Porcentaje prestado	0,404	1.788	0,000	0,218	1.788	0,000

En la tabla 5.8 se muestran los coeficientes de correlación entre las variables cuantitativas. Existen algunas relaciones significativas que medimos a través del coeficiente de correlación de Pearson. La variable número de miembros familiares tiene una alta correlación con el importe del préstamo, 0,93, el importe de la inversión, 0,93 y con el porcentaje prestado, 0,92.

El importe del préstamo muestra una ligera correlación con el importe de la inversión, 0,66 y con el importe de la cuota, 0,73. El grado de correlación entre el importe del préstamo y los importes pendientes de pago es alto, 0,92 y, finalmente, también existe un grado alto de asociación entre los importes pendientes y el saldo medio mantenido.

Tabla 5.8. Coeficientes de correlación entre las variables cuantitativas.

		Nº miembros familiares	Valor de la vivienda	Importe del patrimonio	Importe del préstamo	Importe de la inversión	Importe de la cuota	Ingresos	Importes pendientes	Saldo medio	Edad	Porcentaje prestado
Nº miembros familiares	Correlación de Pearson	1	0,202**	0,032	-0,002	-0,002	0,035	0,024	0,025	0,037	0,174**	-0,002
	Sig. (bilateral)		0,000	0,177	0,928	0,934	0,134	0,320	0,285	0,115	0,000	0,921
Valor de la vivienda	Correlación de Pearson	0,202**	1	0,167**	0,119**	0,155**	0,098**	0,329**	0,241**	0,124**	0,380**	0,094**
	Sig. (bilateral)	0,000		0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Importe del patrimonio	Correlación de Pearson	0,032	0,167**	1	0,103**	0,175**	0,138**	0,097**	0,141**	0,054*	0,169**	0,157**
	Sig. (bilateral)	0,177	0,000		0,000	0,000	0,000	0,000	0,000	0,022	0,000	0,000
Importe del préstamo	Correlación de Pearson	-0,002	0,119**	0,103**	1	0,665**	0,730**	0,119**	-0,002	0,018	0,067**	0,058*
	Sig. (bilateral)	0,928	0,000	0,000		0,000	0,000	0,000	0,921	0,459	0,005	0,014
Importe de la inversión	Correlación de Pearson	-0,002	0,155**	0,175**	0,665**	1	0,492**	0,130**	0,026	0,112**	0,088**	0,727**
	Sig. (bilateral)	0,934	0,000	0,000	0,000		0,000	0,000	0,275	0,000	0,000	0,000
Importe de la cuota	Correlación de Pearson	0,035	0,098**	0,138**	0,730**	0,492**	1	0,157**	-0,016	0,092**	0,098**	0,058*
	Sig. (bilateral)	0,134	0,000	0,000	0,000	0,000		0,000	0,500	0,000	0,000	0,014
Ingresos	Correlación de Pearson	0,024	0,329**	0,097**	0,119**	0,130**	0,157**	1	0,155**	0,195**	0,202**	0,075**
	Sig. (bilateral)	0,320	0,000	0,000	0,000	0,000	0,000		0,000	0,000	0,000	0,002
Importes pendientes	Correlación de Pearson	0,025	0,241**	0,141**	-0,002	0,026	-0,016	0,155**	1	0,006	-0,034	0,040
	Sig. (bilateral)	0,285	0,000	0,000	0,921	0,275	0,500	0,000		0,797	0,152	0,091
Saldo medio	Correlación de Pearson	0,037	0,124**	0,054*	0,018	0,112**	0,092**	0,195**	0,006	1	0,124**	0,087**
	Sig. (bilateral)	0,115	0,000	0,022	0,459	0,000	0,000	0,000	0,797		0,000	0,000
Edad	Correlación de Pearson	0,174**	0,380**	0,169**	0,067**	0,088**	0,098**	0,202**	-0,034	0,124**	1	0,071**
	Sig. (bilateral)	0,000	0,000	0,000	0,005	0,000	0,000	0,000	0,152	0,000		0,003
Porcentaje prestado	Correlación de Pearson	-0,002	0,094**	0,157**	0,058*	0,727**	0,058*	0,075**	0,040	0,087**	0,071**	1
	Sig. (bilateral)	0,921	0,000	0,000	0,014	0,000	0,014	0,002	0,091	0,000	0,003	

**. La correlación es significativa al nivel 0,01 (bilateral).

*. La correlación es significante al nivel 0,05 (bilateral).

5.2. Análisis de los modelos estadísticos.

5.2.1. Árboles de decisión.

El análisis de árboles de decisión de los principales métodos empleados en la clasificación se realiza con los programas estadísticos SPSS y con el programa WEKA (Waikato Environment for Knowledge Analysis), dado que SPSS contiene los métodos CHAID y QUEST que no están disponibles en WEKA[1], mientras que en éste existen otros algoritmos como el C 4.5, Random Forest, y REP Tree que se han demostrado opciones muy interesantes en problemas de clasificación. El método CART está disponible en ambos programas.

En el programa SPSS el máximo número de niveles del árbol que se permite es de tres para el método CHAID y de cinco para el CART y el QUEST.

Una forma de controlar la expansión de los árboles de clasificación es a través de la limitación del número de individuos en los nodos parentales y filiales. Los primeros resultados y gráficos que se presentan son con un número mínimo de individuos de cien en el nodo parental y de cincuenta para el nodo filial, y después se reduce a treinta elementos en el nodo parental y diez en el filial.

5.2.1.1. CHAID y CHAID exhaustivo.

Este primer árbol, en donde hemos exigido como condición en su construcción que al menos haya 50 elementos en los nodos hijos y cien elementos en los nodos padres, se configura con tres variables independientes: Tipo de vivienda, Saldo medio e Importe de la Inversión.

El árbol, que se muestra en la figura nº 5.1, está constituido con 11 nodos, de los que 7 son nodos terminales.

Las tablas números 5.9 y 5.10 muestran las ganancias para los nodos terminales considerando como categorías objetivo la No y la Sí devolución del préstamo respectivamente. Si observamos la primera tabla, el nodo 9 con 116 observaciones acumula el 16,7% de la muestra, donde 110 de éstas (el 94,8%, denominado porcentaje de respuesta) no han devuelto el préstamo concedido. Estos 110 individuos que no devuelven el préstamo representan el 31,8% (porcentaje de ganancia) del total de la muestra de los que no lo hacen. El índice de ganancia, calculado como el

[1] *Existen manuales excelentes sobre software WEKA como el desarrollado por García (2006)*

cociente entre el porcentaje de ganancia y el porcentaje de casos acumulados en el nodo, constituye una medida del poder discriminante del nodo, donde valores próximos al 100% señalan la máxima incertidumbre en la predicción de la categoría objetivo. En este caso, los nodos 9, 10 y 7 muestran los perfiles más característicos de las personas que no devuelven los préstamos, en contraposición a los nodos 4, 6 y 8, que recoge los perfiles de las personas que sí lo hacen. El nodo 5 es un nodo de incertidumbre. Como es lógico, el proceso de modelización ha de estar orientado a no tener nodos con índices de ganancia cercanos al 100% y, en caso de que los haya, a que éstos acumulen el menor número de casos posible.

Como puede observarse, ambas tablas nos permiten llegar a idénticas conclusiones, si bien con diferentes perspectivas.

El perfil más característico de la clase correspondiente a no devolver los préstamos es el de aquellas personas que viven en alquiler con un saldo medio igual o inferior a 133.

En las tablas números 5.11 y 5.12 se muestran las estimaciones del riesgo de clasificación errónea y la tabla de clasificación, tanto para el conjunto de datos de entrenamiento como de test. La estimación del riesgo es un indicador del modelo que facilita el programa SPSS que, en el caso de una variable explicativa numérica, constituye una medida de la varianza dentro del nodo. La estimación del riesgo se cifra en el 21,4%, es decir, se clasifican correctamente el 78,6% de los casos, en la muestra de entrenamiento, obteniendo la muestra de contraste peores resultados, con un riesgo del 34,6%.

Figura 5.1. Árbol de decisión. Método CHAID con 100 elementos parentales y 50 filiales.

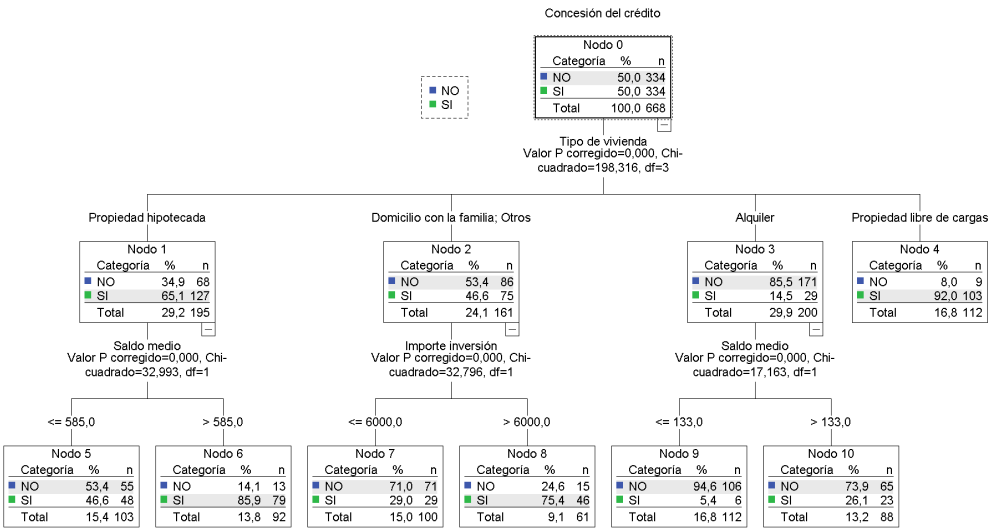


Tabla 5.9. Ganancia para los nodos. Método CHAID con 100 elementos parentales y 50 filiales. Clase NO.

	Nodo		Ganancia			
Nodo	**N**	**Porcentaje**	**N**	**Porcentaje**	**Respuesta**	**Índice**
9	112	16,8%	106	31,7%	94,6%	189,3%
10	88	13,2%	65	19,5%	73,9%	147,7%
7	100	15,0%	71	21,3%	71,0%	142,0%
5	103	15,4%	55	16,5%	53,4%	106,8%
8	61	9,1%	15	4,5%	24,6%	49,2%
6	92	13,8%	13	3,9%	14,1%	28,3%
4	112	16,8%	9	2,7%	8,0%	16,1%

Tabla 5.10. Ganancia para los nodos. Método CHAID con 100 elementos parentales y 50 filiales. Clase SÍ.

	Nodo		Ganancia			
Nodo	**N**	**Porcentaje**	**N**	**Porcentaje**	**Respuesta**	**Índice**
4	112	16,8%	103	30,8%	92,0%	183,9%
6	92	13,8%	79	23,7%	85,9%	171,7%
8	61	9,1%	46	13,8%	75,4%	150,8%
5	103	15,4%	48	14,4%	46,6%	93,2%
7	100	15,0%	29	8,7%	29,0%	58,0%
10	88	13,2%	23	6,9%	26,1%	52,3%
9	112	16,8%	6	1,8%	5,4%	10,7%

Tabla 5.11. Tabla de riesgo del método CHAID con 100 elementos parentales y 50 filiales.

Muestra	Estimación	Típ. Error
Entrenamiento	0,214	0,016
Contraste	0,346	0,093

Tabla 5.12. Resultados de la clasificación método CHAID con 100 elementos parentales y 50 filiales.

		Pronosticado		
Muestra		NO	SI	Porcentaje correcto
Entrenamiento	NO	297	37	88,9%
	SÍ	106	228	68,3%
	Porcentaje global	60,3%	39,7%	78,6%
Contraste	NO	10	2	83,3%
	SÍ	7	7	50,0%
	Porcentaje global	65,4%	34,6%	65,4%

Al disminuir el número de elementos que exigimos en los nodos padres e hijos aumentamos la complejidad del árbol. Así, si exigimos un número mínimo de 30 observaciones para los nodos padres y 10 para los hijos, el número de nodos en la construcción del árbol pasa de 11 a 29, de los que 18 son nodos terminales. También ha aumentado la profundidad del árbol construido.

Figura 5.2. Árbol de decisión. Método CHAID con 30 elementos parentales y 10 filiales.

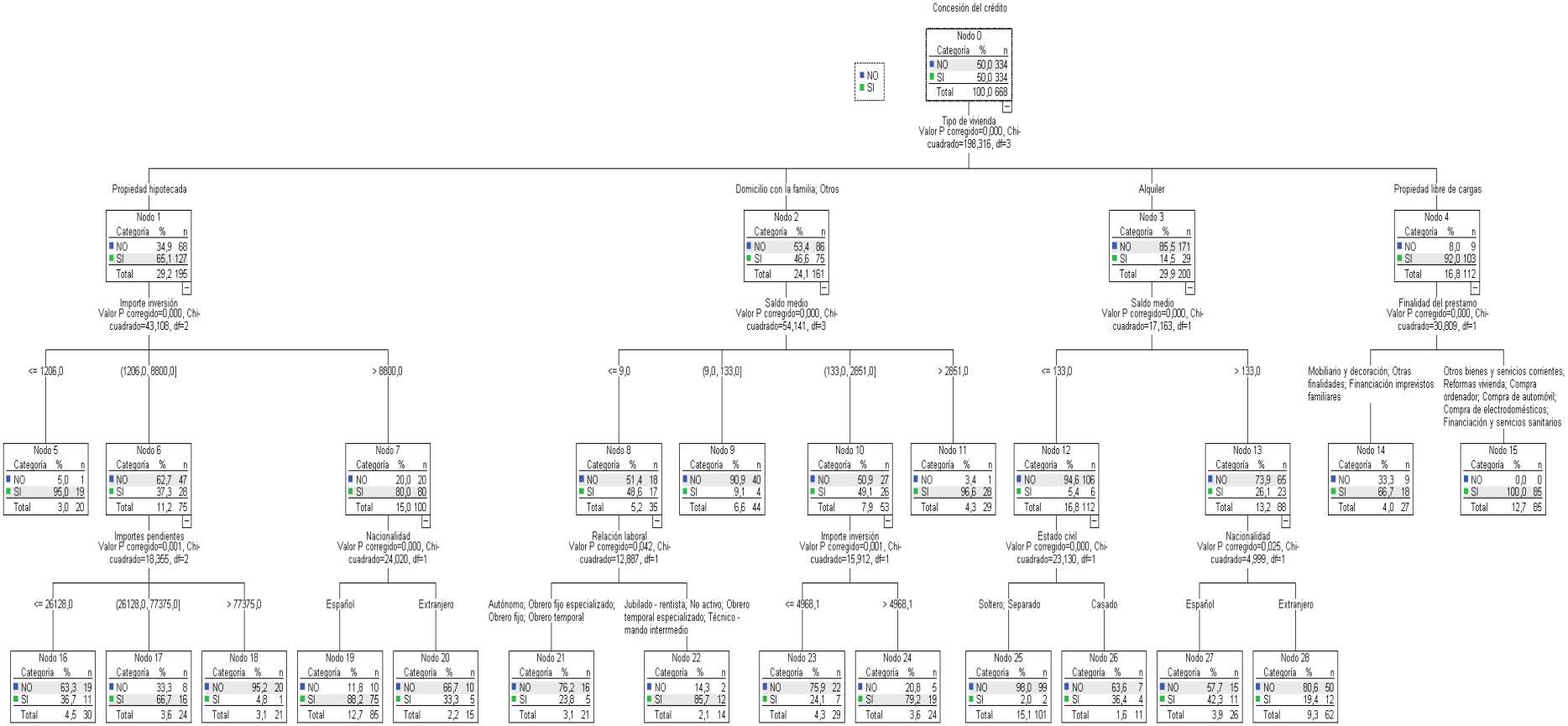


Tabla 5.13. Ganancia para los nodos. Método CHAID con 30 elementos parentales y 10 filiales. Clase NO.

	Nodo		Ganancia			
Nodo	**N**	**Porcentaje**	**N**	**Porcentaje**	**Respuesta**	**Índice**
25	101	15,1%	99	29,6%	98,0%	196,0%
18	21	3,1%	20	6,0%	95,2%	190,5%
9	44	6,6%	40	12,0%	90,9%	181,8%
28	62	9,3%	50	15,0%	80,6%	161,3%
21	21	3,1%	16	4,8%	76,2%	152,4%
23	29	4,3%	22	6,6%	75,9%	151,7%
20	15	2,2%	10	3,0%	66,7%	133,3%
26	11	1,6%	7	2,1%	63,6%	127,3%
16	30	4,5%	19	5,7%	63,3%	126,7%
27	26	3,9%	15	4,5%	57,7%	115,4%
14	27	4,0%	9	2,7%	33,3%	66,7%
17	24	3,6%	8	2,4%	33,3%	66,7%
24	24	3,6%	5	1,5%	20,8%	41,7%
22	14	2,1%	2	,6%	14,3%	28,6%
19	85	12,7%	10	3,0%	11,8%	23,5%
5	20	3,0%	1	0,3%	5,0%	10,0%
11	29	4,3%	1	,3%	3,4%	6,9%
15	85	12,7%	0	,0%	0,0%	0,0%

Tabla 5.14. Ganancia para los nodos. Método CHAID con 30 elementos parentales y 10 filiales. Clase SÍ.

	Nodo		Ganancia			
Nodo	**N**	**Porcentaje**	**N**	**Porcentaje**	**Respuesta**	**Índice**
15	85	12,7%	85	25,4%	100,0%	200,0%
11	29	4,3%	28	8,4%	96,6%	193,1%
5	20	3,0%	19	5,7%	95,0%	190,0%
19	85	12,7%	75	22,5%	88,2%	176,5%
22	14	2,1%	12	3,6%	85,7%	171,4%
24	24	3,6%	19	5,7%	79,2%	158,3%
14	27	4,0%	18	5,4%	66,7%	133,3%
17	24	3,6%	16	4,8%	66,7%	133,3%
27	26	3,9%	11	3,3%	42,3%	84,6%
16	30	4,5%	11	3,3%	36,7%	73,3%
26	11	1,6%	4	1,2%	36,4%	72,7%
20	15	2,2%	5	1,5%	33,3%	66,7%
23	29	4,3%	7	2,1%	24,1%	48,3%
21	21	3,1%	5	1,5%	23,8%	47,6%
28	62	9,3%	12	3,6%	19,4%	38,7%
9	44	6,6%	4	1,2%	9,1%	18,2%
18	21	3,1%	1	,3%	4,8%	9,5%
25	101	15,1%	2	,6%	2,0%	4,0%

Tabla 5.15. Tabla de riesgo del método CHAID con 30 elementos parentales y 10 filiales.

Muestra	Estimación	Típ. Error
Entrenamiento	0,147	0,014
Contraste	0,192	0,077

Tabla 5.16. Resultados de la clasificación método CHAID con 30 elementos parentales y 10 filiales.

		Pronosticado		
Muestra		NO	SI	Porcentaje correcto
Entrenamiento	NO	298	36	89,2%
	SÍ	62	272	81,4%
	Porcentaje global	53,9%	46,1%	85,3%
Contraste	NO	11	1	91,7%
	SÍ	4	10	71,4%
	Porcentaje global	57,7%	42,3%	80,8%

Las variables independientes que incluye el método CHAID en la construcción del árbol ha aumentado en cinco en relación al modelo anterior: Tipo de vivienda, Importe

inversión, Importes pendientes, Nacionalidad, Saldo medio, Relación laboral, Estado civil, Finalidad del préstamo.

Podemos intuir en base a las tablas de ganancias una mejora importante en el poder discriminante de los nodos, donde el nodo de máxima incertidumbre (ver tabla número 5.13) sería el 27 con un índice de ganancia de 115,4%. El nodo 25 es el que señala en este caso el perfil más característico de los que no devuelven el préstamo (vivienda en alquiler, saldo medio <=133 y estado civil soltero o separado).

La tabla de clasificación nos indica que con los datos de entrenamiento el 85,3% de los individuos han sido correctamente clasificados (es decir, el riesgo se sitúa en el 14,7%). Mayor porcentaje de acierto ha experimentado la clase de los que no devuelven el crédito, 89,2% frente al 81,4% de los que pagan el crédito concedido. Estos porcentajes de clasificación son, por tanto, mayores que en el modelo más restringido o con mayor número de individuos en los nodos.

El CHAID exhaustivo propuesto por Bigg et al. (1991) consigue que la fusión continua de pares de valores se reduzca hasta que sólo quede una dicotomía de valores. El árbol generado por este algoritmo se encuentra en la figura 5.3.

Las variables que son significativas para la construcción del árbol son las siguientes: Tipo de vivienda, Importe de la cuota, Importe de la inversión, Nacionalidad, Saldo medio, Relación laboral, Estado civil y Finalidad del préstamo.

Con el algoritmo CHAID exhaustivo el árbol ha aumentado su complejidad, ahora contiene 30 nodos de los que 19 son nodos terminales.

Las tablas de ganancias nos señalan resultados muy similares al anterior. Comparando los resultados en términos de predicción el modelo exhaustivo consigue un precisión global muy similar aunque ligeramente inferior, 84,6%. En términos de las clases alcanza mejores resultados para la clase de los que devuelven el crédito, 85,6% y menor en la que no devuelven el dinero prestado, 83,5%.

Con los datos que sirven de contraste, el método exhaustivo obtiene mejores resultados en la precisión total y también para la clase con valor igual a SÍ.

Figura 5.3. Árbol de decisión. Método CHAID exhaustivo.

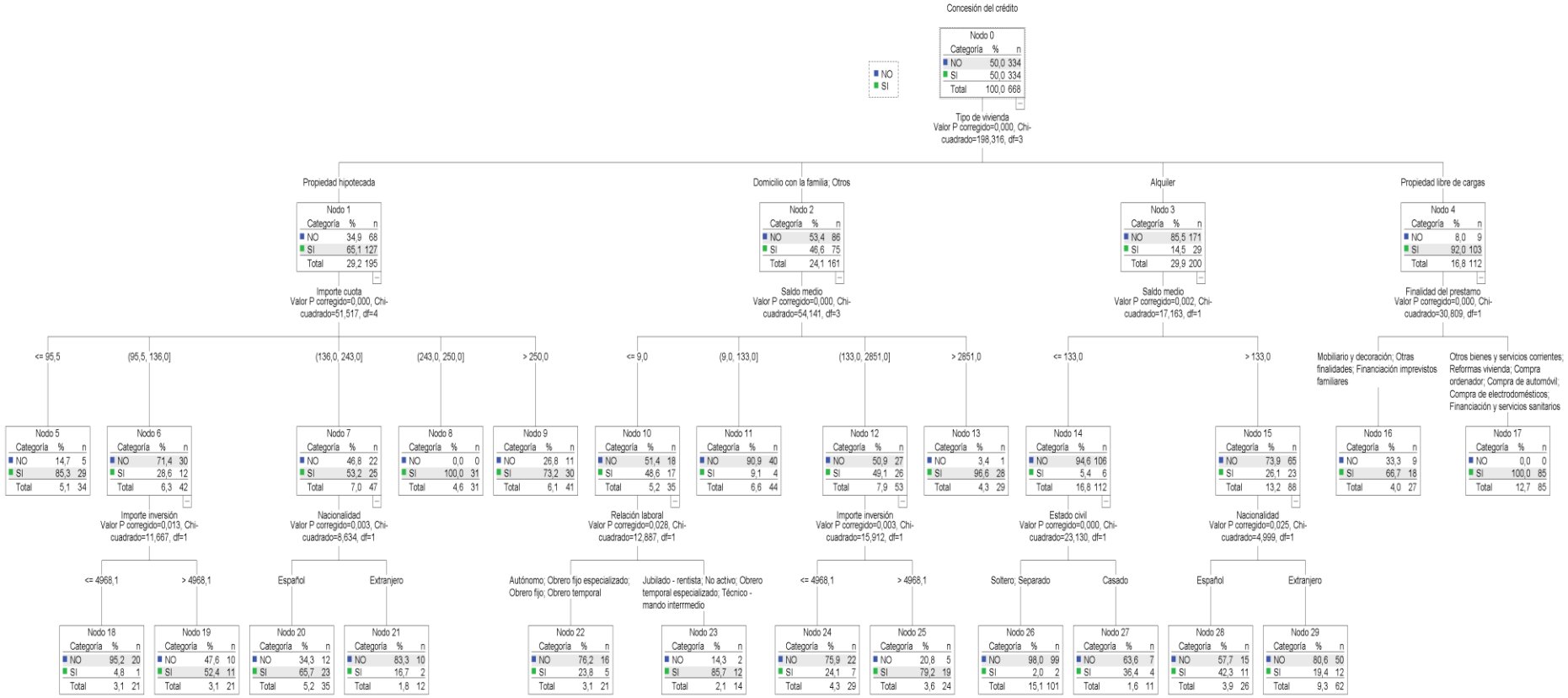


Tabla 5.17. Ganancia para los nodos. Método CHAID exhaustivo con 30 elementos parentales y 10 filiales. Clase NO.

	Nodo		Ganancia			
Nodo	**N**	**Porcentaje**	**N**	**Porcentaje**	**Respuesta**	**Índice**
26	101	15,1%	99	29,6%	98,0%	196,0%
18	21	3,1%	20	6,0%	95,2%	190,5%
11	44	6,6%	40	12,0%	90,9%	181,8%
21	12	1,8%	10	3,0%	83,3%	166,7%
29	62	9,3%	50	15,0%	80,6%	161,3%
22	21	3,1%	16	4,8%	76,2%	152,4%
24	29	4,3%	22	6,6%	75,9%	151,7%
27	11	1,6%	7	2,1%	63,6%	127,3%
28	26	3,9%	15	4,5%	57,7%	115,4%
19	21	3,1%	10	3,0%	47,6%	95,2%
20	35	5,2%	12	3,6%	34,3%	68,6%
16	27	4,0%	9	2,7%	33,3%	66,7%
9	41	6,1%	11	3,3%	26,8%	53,7%
25	24	3,6%	5	1,5%	20,8%	41,7%
5	34	5,1%	5	1,5%	14,7%	29,4%
23	14	2,1%	2	,6%	14,3%	28,6%
13	29	4,3%	1	,3%	3,4%	6,9%
17	85	12,7%	0	,0%	0,0%	0,0%
8	31	4,6%	0	,0%	0,0%	0,0%

Tabla 5.18. Ganancia para los nodos. Método CHAID exhaustivo con 30 elementos parentales y 10 filiales. Clase SÍ.

	Nodo		Ganancia			
Nodo	**N**	**Porcentaje**	**N**	**Porcentaje**	**Respuesta**	**Índice**
17	85	12,7%	85	25,4%	100,0%	200,0%
8	31	4,6%	31	9,3%	100,0%	200,0%
13	29	4,3%	28	8,4%	96,6%	193,1%
23	14	2,1%	12	3,6%	85,7%	171,4%
5	34	5,1%	29	8,7%	85,3%	170,6%
25	24	3,6%	19	5,7%	79,2%	158,3%
9	41	6,1%	30	9,0%	73,2%	146,3%
16	27	4,0%	18	5,4%	66,7%	133,3%
20	35	5,2%	23	6,9%	65,7%	131,4%
19	21	3,1%	11	3,3%	52,4%	104,8%
28	26	3,9%	11	3,3%	42,3%	84,6%
27	11	1,6%	4	1,2%	36,4%	72,7%
24	29	4,3%	7	2,1%	24,1%	48,3%
22	21	3,1%	5	1,5%	23,8%	47,6%
29	62	9,3%	12	3,6%	19,4%	38,7%
21	12	1,8%	2	,6%	16,7%	33,3%
11	44	6,6%	4	1,2%	9,1%	18,2%
18	21	3,1%	1	,3%	4,8%	9,5%
26	101	15,1%	2	,6%	2,0%	4,0%

Tabla 5.19. Tabla de riesgo del método CHAID exhaustivo.

Muestra	Estimación	Típ. Error
Entrenamiento	0,154	0,014
Contraste	0,154	0,071

Tabla 5.20. Resultados de la clasificación método CHAID exhaustivo.

Muestra		Pronosticado		
		NO	SI	Porcentaje correcto
Entrenamiento	NO	279	55	83,5%
	SÍ	48	286	85,6%
	Porcentaje global	49,0%	51,0%	84,6%
Contraste	NO	11	1	91,7%
	SI	3	11	78,6%
	Porcentaje global	53,8%	46,2%	84,6%

5.2.1.2 QUEST.

El método QUEST utiliza las siguientes variables en la construcción del árbol: Valor vivienda, Tipo de vivienda, Nacionalidad, Relación laboral, Finalidad del préstamo y Saldo medio, tal y como podemos observar en la figura nº 5.4. Este árbol está constituido por 25 nodos, de los que 12 son terminales, con una profundidad de 5 niveles. Esta representación es más compleja que con el método CHAID.

Según las tablas de ganancias, el perfil más característico de las personas que no devuelven los préstamos es valor de la vivienda menor o igual a 89.234,1 euros, viviendo con la familia o en alquiler, autónomos y obreros temporales o fijos (especializados o no) y saldo medio menor o igual a 1.754,4. Existe un nodo de máxima incertidumbre, si bien éste sólo acumula 16 casos.

Por último, el porcentaje global de clasificación correcta se sitúa en el 82,0% (riesgo igual a 18,0%), obteniendo mejores resultados en la clase de los que no devuelven los préstamos, tanto en la muestra de entrenamiento como en la de contraste, con un 85,0% y 83,3% respectivamente, presentando ésta última peores resultados, con un porcentaje de clasificación correcta global del 73,1%.

Figura 5.4. Árbol de decisión. Método QUEST.

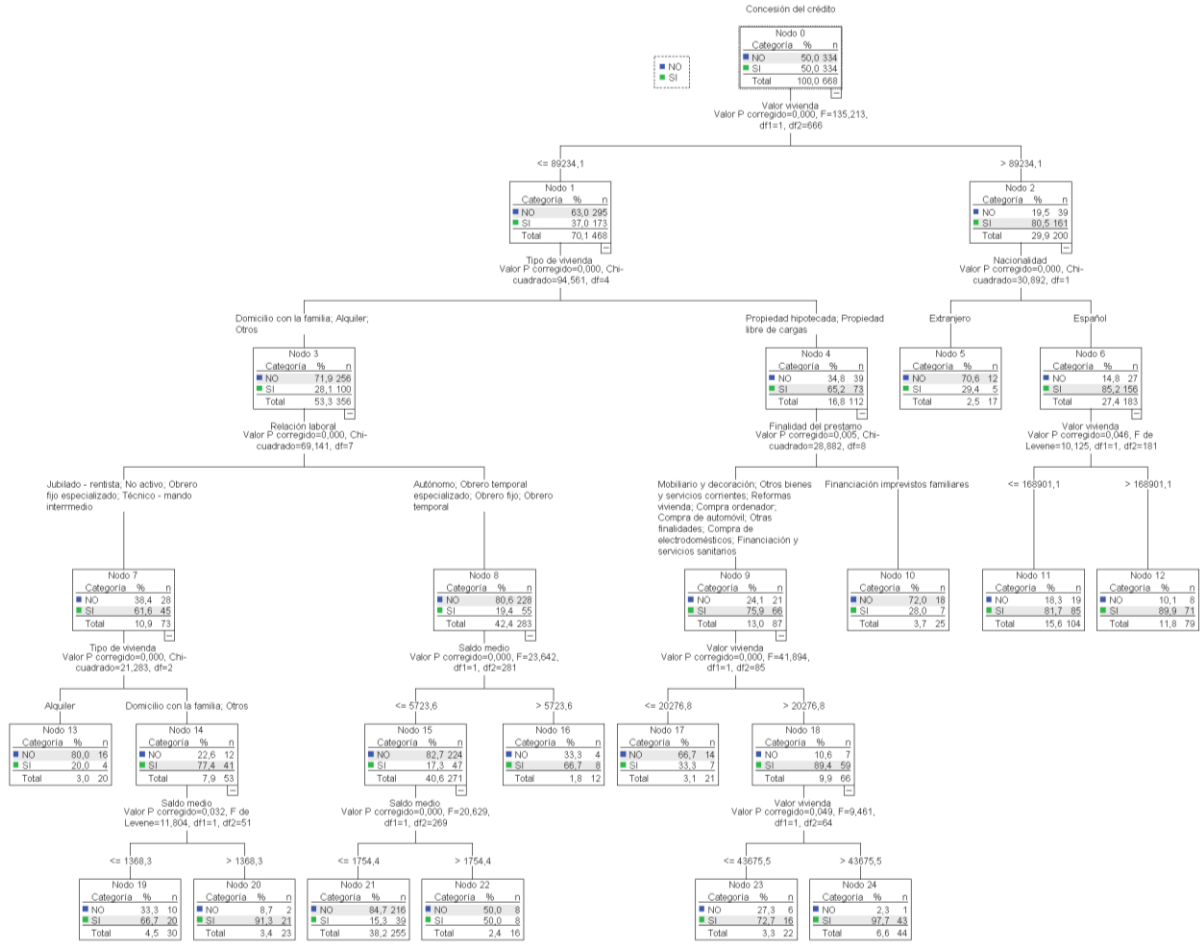


Tabla 5.21. Ganancia para los nodos. Método QUEST con 30 elementos parentales y 10 filiales. Clase NO.

	Nodo		Ganancia			
Nodo	**N**	**Porcentaje**	**N**	**Porcentaje**	**Respuesta**	**Índice**
21	255	38,2%	216	64,7%	84,7%	169,4%
13	20	3,0%	16	4,8%	80,0%	160,0%
10	25	3,7%	18	5,4%	72,0%	144,0%
5	17	2,5%	12	3,6%	70,6%	141,2%
17	21	3,1%	14	4,2%	66,7%	133,3%
22	16	2,4%	8	2,4%	50,0%	100,0%
19	30	4,5%	10	3,0%	33,3%	66,7%
16	12	1,8%	4	1,2%	33,3%	66,7%
23	22	3,3%	6	1,8%	27,3%	54,5%
11	104	15,6%	19	5,7%	18,3%	36,5%
12	79	11,8%	8	2,4%	10,1%	20,3%
20	23	3,4%	2	,6%	8,7%	17,4%
24	44	6,6%	1	,3%	2,3%	4,5%

Tabla 5.22. Ganancia para los nodos. Método QUEST con 30 elementos parentales y 10 filiales. Clase SÍ.

	Nodo		Ganancia			
Nodo	**N**	**Porcentaje**	**N**	**Porcentaje**	**Respuesta**	**Índice**
24	44	6,6%	43	12,9%	97,7%	195,5%
20	23	3,4%	21	6,3%	91,3%	182,6%
12	79	11,8%	71	21,3%	89,9%	179,7%
11	104	15,6%	85	25,4%	81,7%	163,5%
23	22	3,3%	16	4,8%	72,7%	145,5%
19	30	4,5%	20	6,0%	66,7%	133,3%
16	12	1,8%	8	2,4%	66,7%	133,3%
22	16	2,4%	8	2,4%	50,0%	100,0%
17	21	3,1%	7	2,1%	33,3%	66,7%
5	17	2,5%	5	1,5%	29,4%	58,8%
10	25	3,7%	7	2,1%	28,0%	56,0%
13	20	3,0%	4	1,2%	20,0%	40,0%
21	255	38,2%	39	11,7%	15,3%	30,6%

Tabla 5.23. Tabla de riesgo del método QUEST.

Muestra	Estimación	Típ. Error
Entrenamiento	0,180	0,015
Contraste	0,269	0,087

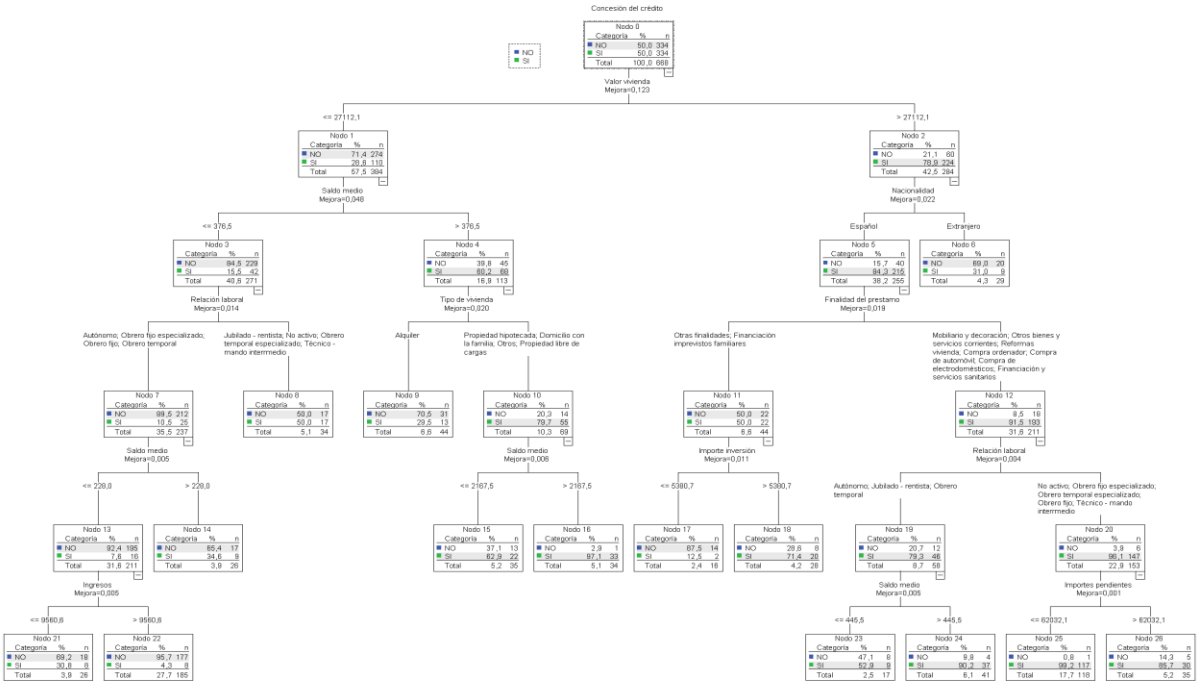
Tabla 5.24. Resultados de la clasificación método QUEST.

Muestra		Pronosticado		Porcentaje correcto
		NO	SI	
Entrenamiento	NO	284	50	85,0%
	SÍ	70	264	79,0%
	Porcentaje global	53,0%	47,0%	82,0%
Contraste	NO	10	2	83,3%
	SÍ	5	9	64,3%
	Porcentaje global	57,7%	42,3%	73,1%

5.2.1.3 CART.

El método CART construye un árbol con 27 nodos, de los que 14 son terminales. Según podemos observar en la siguiente figura, las variables utilizadas han sido: Valor vivienda, Saldo medio, Nacionalidad, Relación laboral, Tipo de vivienda, Finalidad del préstamo, Importe de la inversión, Ingresos e Importes pendientes.

Figura 5.5. Árbol de decisión. Método CART.



Atendiendo a las tablas de ganancia, tablas números 5.25 y 5.26, los perfiles más característicos, tanto de la clase de los que no devuelven los préstamos como la de los que sí lo hacen, corresponden a los nodos que acumulan el mayor número de observaciones (185 casos y 118 respectivamente). El perfil más característico de los clientes morosos es el de aquellos con un valor de la vivienda igual o inferior a 27.112,1 euros, con un saldo medio igual o inferior a 228,0, autónomos y obreros temporales o fijos (especializados o no) e ingresos mayores a 9.560,6 euros.

El riesgo de clasificación errónea en este caso es del 15,9%, clasificando por lo tanto correctamente el 84,1% de los casos, con un porcentaje ligeramente superior en la clase No del 88,0% frente al 80,2% de la clase SÍ. La muestra de contraste clasifica a nivel global peor, 76,9% de aciertos, si bien la clase SÍ presenta mejor resultado que la muestra de entrenamiento, con un porcentaje de aciertos del 91,7%.

Tabla 5.25. Ganancia para los nodos. Método CART con 30 elementos parentales y 10 filiales. Clase NO.

	Nodo		Ganancia			
Nodo	**N**	**Porcentaje**	**N**	**Porcentaje**	**Respuesta**	**Índice**
22	185	27,7%	177	53,0%	95,7%	191,4%
17	16	2,4%	14	4,2%	87,5%	175,0%
9	44	6,6%	31	9,3%	70,5%	140,9%
21	26	3,9%	18	5,4%	69,2%	138,5%
6	29	4,3%	20	6,0%	69,0%	137,9%
14	26	3,9%	17	5,1%	65,4%	130,8%
8	34	5,1%	17	5,1%	50,0%	100,0%
23	17	2,5%	8	2,4%	47,1%	94,1%
15	35	5,2%	13	3,9%	37,1%	74,3%
18	28	4,2%	8	2,4%	28,6%	57,1%
26	35	5,2%	5	1,5%	14,3%	28,6%
24	41	6,1%	4	1,2%	9,8%	19,5%
16	34	5,1%	1	,3%	2,9%	5,9%
25	118	17,7%	1	,3%	0,8%	1,7%

Tabla 5.26. Ganancia para los nodos. Método CART con 30 elementos parentales y 10 filiales. Clase SÍ.

	Nodo		Ganancia			
Nodo	**N**	**Porcentaje**	**N**	**Porcentaje**	**Respuesta**	**Índice**
25	118	17,7%	117	35,0%	99,2%	198,3%
16	34	5,1%	33	9,9%	97,1%	194,1%
24	41	6,1%	37	11,1%	90,2%	180,5%
26	35	5,2%	30	9,0%	85,7%	171,4%
18	28	4,2%	20	6,0%	71,4%	142,9%
15	35	5,2%	22	6,6%	62,9%	125,7%
23	17	2,5%	9	2,7%	52,9%	105,9%
8	34	5,1%	17	5,1%	50,0%	100,0%
14	26	3,9%	9	2,7%	34,6%	69,2%
6	29	4,3%	9	2,7%	31,0%	62,1%
21	26	3,9%	8	2,4%	30,8%	61,5%
9	44	6,6%	13	3,9%	29,5%	59,1%
17	16	2,4%	2	,6%	12,5%	25,0%
22	185	27,7%	8	2,4%	4,3%	8,6%

Tabla 5.27. Tabla de riesgo del método CART.

Muestra	Estimación	Típ. Error
Entrenamiento	0,159	0,014
Contraste	0,231	0,083

Tabla 5.28. Resultados de la clasificación método CART.

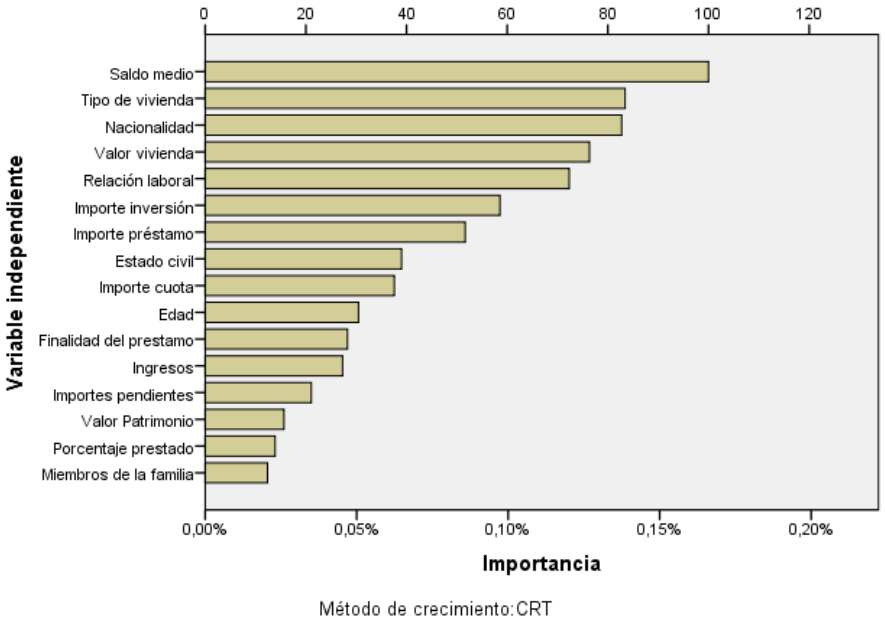
Muestra		Pronosticado		
		NO	SI	Porcentaje correcto
Entrenamiento	NO	294	40	88,0%
	SÍ	66	268	80,2%
	Porcentaje global	53,9%	46,1%	84,1%
Contraste	NO	11	1	91,7%
	SÍ	5	9	64,3%
	Porcentaje global	61,5%	38,5%	76,9%

Este método nos ofrece una medida normalizada sobre la importancia de las variables independientes en la construcción del árbol, siendo 100% la variable de mayor poder discriminante. Como puede observarse en la tabla número 5.29 las variables de mayor importancia de cara a la construcción del árbol son el saldo medio, el tipo de vivienda, la nacionalidad, el valor de la vivienda y la relación laboral (Tipo de trabajo)

Tabla 5.29. Importancia de las variables independientes. Método CART.

Variable independiente	Importancia	Importancia normalizada
Saldo medio	,166	100,0%
Tipo de vivienda	,139	83,4%
Nacionalidad	,137	82,7%
Valor vivienda	,127	76,3%
Relación laboral	,120	72,3%
Importe inversión	,097	58,6%
Importe préstamo	,086	51,7%
Estado civil	,065	39,0%
Importe cuota	,062	37,6%
Edad	,051	30,5%
Finalidad del prestamo	,047	28,3%
Ingresos	,045	27,3%
Importes pendientes	,035	21,1%
Valor Patrimonio	,026	15,7%
Porcentaje prestado	,023	13,9%
Miembros de la familia	,021	12,4%

Figura 5.6. Importancia normalizada de las variables independientes. Método CART.



A través de la Curva ROC podemos evaluar de forma conjunta los tres tipos de métodos de construcción de árboles utilizados hasta ahora. La figura número 5.7 y la tabla número 5.30 nos permite concluir unos resultados muy similares para los métodos CART y CHAID, cuyas áreas bajo la curva ROC se cifran en 0,922 y 0,921 respectivamente, y significativamente mejores que el método QUEST, el cuál obtiene sólo un área de 0,863.

Figura 5.7. Área bajo la Curva ROC. Métodos CHAID, QUEST y CART

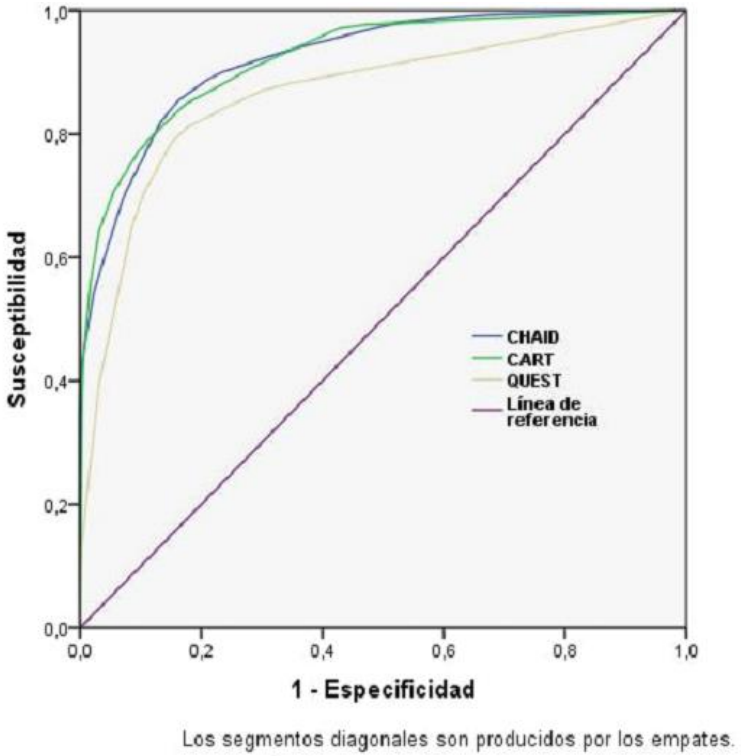


Tabla 5.30. Área bajo la curva ROC y sus intervalos. Métodos CHAID, QUEST y CART.

Área bajo la curva

Variables resultado de contraste	Área	Error típ.[a]	Sig. asintótica[b]	Intervalo de confianza asintótico al 95%	
				Límite inferior	Límite superior
CHAID	,921	,010	,000	,902	,941
CART	,922	,010	,000	,902	,941
QUEST	,863	,014	,000	,835	,891

La variable (o variables) de resultado de contraste: CHAID, CART, QUEST tiene al menos un empate entre el grupo de estado real positivo y el grupo de estado real negativo. Los estadísticos pueden estar sesgados .

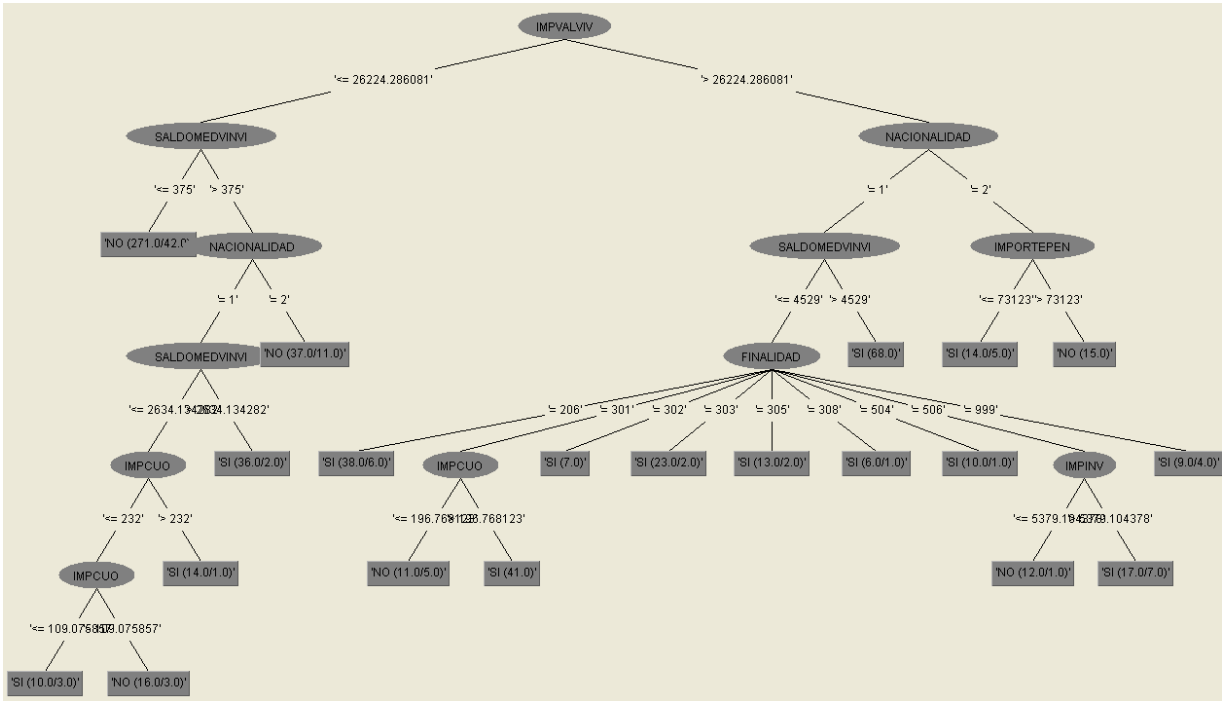
a. Bajo el supuesto no paramétrico

b. Hipótesis nula: área verdadera = 0,5

5.2.1.4. Árbol C.4.5.

La construcción del árbol de decisión generado por el algoritmo C4.5 se ha realizado con un mínimo de diez registros en los nodos filiales. Se han efectuado diferentes pruebas con los valores del parámetro de ajuste “c” y al final se ha fijado en un valor igual a cuatro. El número de nodos es de 32, de los que 20 son nodos terminales. En la figura número 5.8 se observa la construcción del árbol de decisión, donde se han utilizado las siguientes variables: Valor vivienda, Saldo medio, Nacionalidad, Importe de la cuota, Finalidad del préstamo, Importe de la Inversión e Importes pendientes.

Figura 5.8. Árbol de decisión. Método C.4.5.



Las reglas generadas por el algoritmo C4.5. son las siguientes:

```
IMPVALVIV <= 26.224,2
|  SALDOMEDVINVI <= 375: NO (271.0/42.0)
|  SALDOMEDVINVI > 375
|  |  NACIONALIDAD = 1
|  |  |  SALDOMEDVINVI <= 2.634,1
|  |  |  |  IMPCUO <= 232
|  |  |  |  |  IMPCUO <= 109,1: SI (10.0/3.0)
|  |  |  |  |  IMPCUO > 109,1: NO (16.0/3.0)
|  |  |  |  IMPCUO > 232: SI (14.0/1.0)
|  |  |  SALDOMEDVINVI > 2.634,1: SI (36.0/2.0)
|  |  NACIONALIDAD = 2: NO (37.0/11.0)
IMPVALVIV > 26.224,2
|  NACIONALIDAD = 1
|  |  SALDOMEDVINVI <= 4.529
|  |  |  FINALIDAD = 206: SI (38.0/6.0)
|  |  |  FINALIDAD = 301
|  |  |  |  IMPCUO <= 196,7: NO (11.0/5.0)
|  |  |  |  IMPCUO > 196,7: SI (41.0)
|  |  |  FINALIDAD = 302: SI (7.0)
|  |  |  FINALIDAD = 303: SI (23.0/2.0)
|  |  |  FINALIDAD = 305: SI (13.0/2.0)
|  |  |  FINALIDAD = 308: SI (6.0/1.0)
|  |  |  FINALIDAD = 504: SI (10.0/1.0)
|  |  |  FINALIDAD = 506
|  |  |  |  IMPINV <= 5.379,1: NO (12.0/1.0)
|  |  |  |  IMPINV > 5.379,1: SI (17.0/7.0)
|  |  |  FINALIDAD = 999: SI (9.0/4.0)
|  |  SALDOMEDVINVI > 4.529: SI (68.0)
|  NACIONALIDAD = 2
|  |  IMPORTEPEN <= 73.123: SI (14.0/5.0)
|  |  IMPORTEPEN > 73.123: NO (15.0)
```

Tabla 5.31. Resultados de la clasificación método C.4.5.

		Pronosticado		
		SI	NO	Porcentaje correcto
Entrenamiento	SI	260	74	77,8
	NO	60	268	81,7
Porcentaje global		47,9	51,2	79,0
Contraste	SI			
	NO	10	3	71,4
		0	12	100,0
Porcentaje global		38,5	57,7	84,6

La tabla número 5.31 muestra un porcentaje correcto de clasificación del 79,0%, clasificando ligeramente mejor a la clase NO, donde el porcentaje de aciertos se cifra en el 81,7%. La muestra de contraste clasifica correctamente a todos los individuos de la clase No.

5.2.1.5. Comparativa de los distintos métodos de construcción de árboles utilizados.

En la siguiente tabla, la número 5.32, se realiza una comparativa de los distintos métodos utilizados en construcción de árboles. Como puede observarse, en el caso del método CHAID los parámetros relativos al número mínimo de elementos parentales y filiales afectan sensiblemente a la configuración final del árbol. Así, si partimos de la premisa de identificar correctamente a la clase NO, el método con 100 elementos parentales y 50 filiales utiliza solamente tres variables frente a las ocho utilizadas por el método con 30 elementos parentales y 10 filiales, y obtiene resultados muy similares. Este método, en ambos casos, es el que obtiene el mayor porcentaje de clasificación correcta en esta clase.

Si bien existe cierta homogeneidad en la selección de variables, la relevancia de las distintas variables no es independiente del método utilizado.

Teniendo en cuenta el número de métodos en los que aparece cada una de las variables, véase tabla número 5.33, las variables más significativas serían: Saldo medio, Tipo de vivienda, Importe de la inversión, Nacionalidad y Finalidad del préstamo, donde solamente el Saldo medio aparece en todos los métodos y el resto en 5 de los 6.

En este sentido, la menos relevante sería los Ingresos, que sólo es considerada como significativa en el método CART, seguida del Importe de la cuota que aparece en CHAID exhaustivo y C4.5 y el Estado civil, señalada por CHAID con 30 elementos parentales y 10 filiales y CHAID exhaustivo.

Tabla 5.32. Comparativa de los distintos métodos de construcción de árboles de decisión: variables seleccionadas para su construcción y porcentaje correcto de clasificación.

Método de construcción del árbol	Variables seleccionadas	% correcto de clasificación	
Método CHAID con 100 elementos parentales y 50 filiales	Tipo de vivienda	No	88,9
	Saldo medio	Si	68,3
	Importe de la Inversión	Total	78,6
Método CHAID con 30 elementos parentales y 10 filiales	Tipo de vivienda		
	Importe de la inversión		
	Importes pendientes	No	89,2
	Nacionalidad	Si	81,4
	Saldo medio	Total	85,3
	Relación laboral		
	Estado civil		
	Finalidad del préstamo		
Método CHAID exhaustivo	Tipo de vivienda		
	Importe de la cuota		
	Importe de la inversión	No	83,5
	Nacionalidad	Si	85,6
	Saldo medio	Total	84,6
	Relación laboral		
	Estado civil		
	Finalidad del préstamo		
Método QUEST	Valor vivienda		
	Tipo de vivienda	No	85,0
	Nacionalidad	Si	79,0
	Relación laboral	Total	82,0
	Finalidad del préstamo		
	Saldo medio		
Método CART	Valor vivienda		
	Saldo medio		
	Nacionalidad		
	Relación laboral	No	88,0
	Tipo de vivienda	Si	80,2
	Finalidad del préstamo	Total	84,1
	Importe de la inversión		
	Ingresos		
	Importes pendientes		
Método C.4.5.	Valor vivienda		
	Saldo medio		
	Nacionalidad	No	77,8
	Importe de la cuota	Si	81,7
	Finalidad del préstamo	Total	79,0
	Importe de la Inversión		
	Importes pendientes		

Tabla 5.33. Número de métodos en los que aparece cada una de las variables seleccionadas.

Saldo medio	*6*
Tipo de vivienda	5
Importe de la inversión	5
Nacionalidad	5
Finalidad del préstamo	5
Relación laboral	4
Importes pendientes	3
Valor vivienda	3
Estado civil	2
Importe de la cuota	2
Ingresos	1

5.2.2. Redes neuronales.

Las Redes Neuronales Artificiales han mostrado ser modelos muy versátiles y han sido empleadas en múltiples campos del conocimiento. En este epígrafe del capítulo cinco se comentan los resultados obtenidos con varias estructuras de redes neuronales: Perceptrón Multicapa, el método propuesto por Fletcher (1987) conocido por las siglas de cuatro investigadores (BFGS) y que ofrece, en general, buenos resultados y las Redes de Base Radial que utilizan una función de cálculo en vez de una función de activación en las neuronas de la capa oculta.

Tabla 5.34. Comparación de modelos. Perceptrón Multicapa. Fase de Entrenamiento.

		Fase de entrenamiento						
		Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Learning Rate	Momentum	SÍ	NO	Total	SÍ	NO	Total	Valor
0,9	0,9	50,7	58,1	54,4	55,6	57,2	56,4	0,768
Con decaimiento		66,3	72,2	69,3	69,2	80,3	74,8	0,848
0,9	0,8	66,9	74,6	70,8	70,1	65,8	68,0	0,816
Con decaimiento		77,4	81,2	79,3	79,3	78,2	78,8	0,858
0,9	0,7	75,0	78,6	76,8	80,9	78,4	79,7	0,844
Con decaimiento		77,3	81,5	79,4	79,8	78,4	79,1	0,866
0,9	0,6	77,5	80,7	79,1	77,1	80,3	78,7	0,861
Con decaimiento		77,2	82,4	79,8	80,6	77,3	79,0	0,872
0,9	0,5	77,4	80,6	79,0	75,6	77,8	76,7	0,857
Con decaimiento		77,3	83,0	80,2	83,8	78,1	81,0	0,874
0,9	0,4	77,5	78,9	78,2	78,9	76,3	77,6	0,853
Con decaimiento		77,8	82,9	80,4	83,0	77,9	80,5	0,879
0,8	0,25	77,4	80,4	78,9	80,6	77,1	78,9	0,851
Con decaimiento		78,0	83,7	80,9	82,9	79,3	81,1	0,882
0,8	0,4	77,4	81,5	79,5	80,3	77,7	79,0	0,854
Con decaimiento		77,8	83,3	80,6	83,1	78,3	80,7	0,879
0,8	0,6	77,9	79,4	78,7	77,0	77,8	77,4	0,856
Con decaimiento		77,7	83,1	80,4	81,8	78,2	80,0	0,879
0,8	0,7	76,0	78,6	77,3	78,8	79,6	79,2	0,837
Con decaimiento		77,2	81,7	79,5	79,6	77,1	78,4	0,870
0,8	0,8	66,3	77,0	71,7	72,7	80,6	76,7	0,822
Con decaimiento		78,5	81,2	79,8	80,9	81,1	81,0	0,870
0,7	0,7	75,7	80,7	78,2	81,1	79,1	80,1	0,848
Con decaimiento		77,7	82,2	80,0	80,9	78,4	79,7	0,869
0,6	0,6	78,2	80,3	79,3	80,4	77,5	79,0	0,855
Con decaimiento		77,8	83,7	80,8	82,4	78,4	80,4	0,879
0,5	0,5	77,7	81,0	79,4	80,6	77,9	79,3	0,856
Con decaimiento		78,3	83,9	81,1	84,2	79,6	81,9	0,882
0,4	0,2	78,1	80,3	79,2	79,3	78,2	78,8	0,852
Con decaimiento		78,9	83,2	81,1	82,6	80,1	81,4	0,886
0,3	0,2	78,2	80,2	79,2	81,7	79,7	80,7	0,863
Con decaimiento		79,0	82,4	80,7	82,1	80,2	81,2	0,886

Los resultados que ofrecen estos tres modelos que han sido obtenidos a través del programa de minería de datos WEKA, se ofrecen tanto para la muestra de entrenamiento como para la de test.

Para el primer modelo estimado, el Perceptrón Multicapa, los resultados se encuentran en las tablas 5.34 y 5.35.

Tabla 5.35. Comparación de modelos. Perceptrón Multicapa. Fase de Test.

		Fase de test						
		Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Learning Rate	Momentum	SÍ	NO	Total	SÍ	NO	Total	Valor
0,9	0,9	78,6	50,0	65,4	64,7	66,7	65,6	0,851
Con decaimiento		0,0	100,0	46,2	0,0	46,2	21,3	0,833
0,9	0,8	85,7	100,0	92,3	100,0	85,7	93,4	0,839
Con decaimiento		85,7	100,0	92,3	100,0	85,7	93,4	0,923
0,9	0,7	57,1	83,3	69,2	80,0	62,5	28,8	0,798
Con decaimiento		78,6	91,7	84,6	91,7	78,6	85,6	0,929
0,9	0,6	50,0	83,3	65,4	77,8	58,8	69,0	0,817
Con decaimiento		71,4	83,3	76,9	83,3	71,4	77,8	0,893
0,9	0,5	57,1	100,0	76,9	100,0	66,7	84,6	0,923
Con decaimiento		64,3	91,7	76,9	90,0	68,8	80,2	0,899
0,9	0,4	71,4	100,0	84,6	100,0	75,0	88,5	0,893
Con decaimiento		64,3	83,3	73,1	81,8	66,7	74,8	0,893
0,8	0,25	92,9	100,0	96,2	100,0	92,3	96,4	0,946
Con decaimiento		64,3	91,7	76,9	90,0	68,8	80,2	0,869
0,8	0,4	85,7	100,0	92,3	100,0	85,7	93,4	0,976
Con decaimiento		64,3	91,7	76,9	90,0	68,8	80,2	0,881
0,8	0,6	85,7	83,3	84,6	85,7	83,3	84,6	0,940
Con decaimiento		78,6	91,7	84,6	91,7	78,6	85,6	0,923
0,8	0,7	64,3	75,0	69,2	75,0	64,3	70,1	0,804
Con decaimiento		78,6	100,0	88,5	100,0	80,0	90,8	0,929
0,8	0,8	78,6	75,0	76,9	78,6	75,0	76,9	0,814
Con decaimiento		78,6	100,0	88,5	100,0	0,8	90,8	0,952
0,7	0,7	50,0	91,7	69,2	87,5	61,1	75,3	0,854
Con decaimiento		78,5	91,7	84,6	91,7	78,6	85,6	0,911
0,6	0,6	71,4	100,0	84,6	100,0	75,0	88,5	0,917
Con decaimiento		64,3	91,7	76,9	90,0	68,8	80,2	0,893
0,5	0,5	78,6	100,0	88,5	100,0	80,0	90,8	0,935
Con decaimiento		64,3	91,7	76,9	90,0	68,8	80,2	0,881
0,4	0,2	78,6	100,0	88,5	100,0	80,0	90,8	0,964
Con decaimiento		64,3	91,7	76,9	90,0	68,9	80,2	0,857
0,3	0,2	78,6	100,0	88,5	100,0	80,0	90,8	0,946
Con decaimiento		64,3	91,7	76,9	90,0	68,8	80,2	0,875

Se ha realizado una simulación con diferentes valores de la tasa de aprendizaje (Learning Rate) y del momento (Momentum). La tasa de aprendizaje, en general, va disminuyendo a medida que se lleva a cabo el entrenamiento. Respecto al momento, es muy difícil realizar una simulación exhaustiva dado que este parámetro oscila entre 0 y 1.

En los cuadros se muestran resultados para dieciséis combinaciones de valores que oscilan entre el 0,9 para ambos parámetros y entre el 0,3 y 0,2 para la tasa de aprendizaje y el momento respectivamente.

En esta investigación se utilizan variables normalizadas y, para evitar que los parámetros de la red tengan valores muy grandes, se emplea una función objetivo que penaliza los valores muy altos; esta función objetivo es conocida como weight decay.

Para dilucidar que modelos son los más apropiados a la hora de clasificar se ha utilizado el contraste estadístico de la T de Student que facilita el programa WEKA. El modelo base de contrastación ha sido una red neuronal con una tasa de aprendizaje de 0,8 y un momento de 0,25 con decaimiento. Aunque en términos de tasas de acierto respecto a las clases se puede considerar que no hay diferencias estadísticamente significativas, cuando se contrasta el valor de la curva ROC si se aprecia que existe un conjunto de modelos que, atendiendo a los resultados de los test de hipótesis, podemos considerarlos significativamente peores. Estos modelos están en la tabla 5.36. marcados con un asterisco.

Tabla 5.36. Comparación de modelos. Perceptrón Multicapa.

Modelo base de contrastación: L = 0,8 y Momentum = 0,25 con decaimiento		Correctamente clasificados				Área Curva ROC	
Learning Rate	Momentum	Sí	Desviación estándar	NO	Desviación estándar	Valor	Desviación estándar
0,9	**0,9**	50,7	0,471	58,1	0,457	0,768 (*)	0,146
Con decaimiento		66,3	0,334	72,2	0,319	0,848 (*)	0,058
0,9	**0,8**	66,9	0,338	74,6	0,198	0,816 (*)	0,063
Con decaimiento		77,4	0,107	81,2	0,104	0,858	0,078
0,9	**0,7**	75,0	0,180	78,6	0,121	0,844 (*)	0,058
Con decaimiento		77,3	0,073	81,5	0,072	0,866 (*)	0,043
0,9	**0,6**	77,5	0,081	80,7	0,094	0,861 (*)	0,046
Con decaimiento		77,2	0,071	82,4	0,074	0,872	0,040
0,9	**0,5**	77,4	0,086	80,6	0,089	0,857 (*)	0,041
Con decaimiento		77,3	0,067	83,0	0,072	0,874	0,036
0,9	**0,4**	77,5	0,088	78,9	0,081	0,853 (*)	0,046
Con decaimiento		77,8	0,068	82,9	0,073	0,879	0,037
0,8	**0,25**	77,4	0,070	80,4	0,074	0,851 (*)	0,046
Con decaimiento		78,0	0,072	83,7	0,071	0,882	0,039
0,8	**0,4**	77,4	0,070	81,5	0,082	0,854 (*)	0,044
Con decaimiento		77,8	0,080	83,3	0,072	0,879	0,038
0,8	**0,6**	77,9	0,066	79,4	0,092	0,856 (*)	0,047
Con decaimiento		77,7	0,071	83,1	0,073	0,879	0,039
0,8	**0,7**	76,0	0,165	78,6	0,102	0,837 (*)	0,059
Con decaimiento		77,2	0,073	81,7	0,069	0,870	0,042
0,8	**0,8**	66,3	0,344	77,0	0,179	0,822 (*)	0,063
Con decaimiento		78,5	0,068	81,2	0,069	0,870	0,042
0,7	**0,7**	75,7	0,162	80,7	0,094	0,848 (*)	0,051
Con decaimiento		77,7	0,075	82,2	0,07	0,869	0,039
0,6	**0,6**	78,2	0,076	80,3	0,074	0,855 (*)	0,043
Con decaimiento		77,8	0,069	83,7	0,07	0,879	0,038
0,5	**0,5**	77,7	0,078	81,0	0,069	0,856 (*)	0,041
Con decaimiento		78,3	0,070	83,9	0,069	0,882	0,038
0,4	**0,2**	78,1	0,076	80,3	0,07	0,852 (*)	0,04
Con decaimiento		78,9	0,064	83,2	0,064	0,886	0,036
0,3	**0,2**	78,2	0,073	80,2	0,072	0,863 (*)	0,044
Con decaimiento		79,0	0,067	82,4	0,063	0,886	0,036

Nota: (*) Estadísticamente peor que el modelo base.

Para la estimación de redes neuronales los métodos conocidos como Cuasi-Newton son bastante utilizados. Todos los algoritmos de cuasi-Newton son aproximaciones del algoritmo de Newton-Raphson, pues usan aproximaciones numéricas de la matriz Hessiana para extraer información sobre la concavidad, dado que estimar la matriz Hessiana resulta computacionalmente costosa de obtener y, además, se requiere que la matriz sea invertible.

En los cuadros siguientes se presentan los resultados con la forma de estimación de la red neuronal a través de la propuesta efectuada por Broyden-Fletcher-Goldfarb-Shanno (BFGS) que ha demostrado un buen desempeño.

En este método se minimiza el error cuadrático medio más una función de penalización. El parámetro ridge se utiliza para controlar el tamaño de los pesos. En el programa Weka, que es con el que se han obtenido los resultados, se puede especificar el parámetro ridge y el número de neuronas de la capa oculta. Es con la combinación de estos dos parámetros con los que se han realizado las diferentes pruebas para intentar conseguir la estructura de red neuronal que mejor se adapte a la muestra de entrenamiento y a la de test. Los resultados alcanzados se pueden observar en las tablas 5.37 y 5.38.

La fase de contrastación estadística de modelos buscando el que mejor se adecúa al proceso de credit scoring se puede observar en la tabla 5.39. Tan sólo tres modelos son significativamente diferentes al resto de los modelos considerados en cuanto que el valor del área bajo la curva resulta ser más bajo: modelos con un ridge de 0,01 y con cuatro, cinco y seis neuronas.

Tabla 5.37. Modelo Perceptrón Multicapa. Método BFGS según número de neuronas y ridge Fase de entrenamiento.

		Fase de entrenamiento						
		Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Ridge	Número neuronas	SÍ	NO	Total	SÍ	NO	Total	Valor
0,01	2	76,6	83,5	80,1	81,4	80,3	0,876	0,877
	3	76,8	81,2	79,0	80,6	78,8	0,856	0,853
	4	76,1	81,2	78,7	80,5	77,7	0,849	0,840
	5	76,6	80,5	78,6	80,1	78,0	0,855	0,851
	6	77,2	79,6	78,4	81,1	78,9	0,857	0,875
0,05	2	76,7	83,6	80,2	79,1	76,7	0,882	0,867
	3	76,8	82,3	79,6	80,9	78,4	0,869	0,848
	4	78,2	81,6	79,9	80,5	77,7	0,867	0,853
	5	77,4	82,7	80,1	81,2	79,6	0,870	0,880
	6	79,2	81,9	80,6	80,8	77,8	0,872	0,862
0,1	2	76,4	83,8	80,1	82,7	77,8	0,884	0,882
	3	77,1	82,9	80,0	81,5	78,8	0,877	0,879
	4	78,4	82,5	80,5	80,7	79,2	0,876	0,883
	5	78,9	82,8	80,9	83,7	82,2	0,882	0,899
	6	78,7	82,7	80,7	81,6	80,1	0,882	0,884
0,2	2	78,4	83,6	81,0	82,0	79,8	0,885	0,887
	3	78,2	83,3	80,8	81,1	79,1	0,888	0,887
	4	79,3	83,3	81,3	83,0	80,9	0,885	0,886
	5	79,2	83,3	81,3	83,1	80,2	0,885	0,894
	6	78,9	83	81,0	83,2	78,8	0,891	0,898
1	2	79,1	83,3	81,2	81,0	79,8	0,887	0,891
	3	79,2	81,6	80,4	82,1	80,2	0,891	0,893
	4	79,2	82,4	80,8	81,6	80,1	0,890	0,893
	5	79,2	82,0	80,6	81,3	80,1	0,889	0,892
	6	79,2	82,0	80,6	81,3	80,1	0,888	0,891
2	2	79,0	81,8	80,4	81,4	80,3	0,884	0,886
	3	79,1	81,8	80,5	81,4	80,3	0,883	0,886
	4	79,3	81,6	80,5	81,6	80,1	0,883	0,887
	5	79,2	81,5	80,4	81,3	80,1	0,883	0,887
	6	79,2	81,2	80,2	81,3	80,1	0,883	0,887
10	2	79,2	81,2	80,2	78,9	77,7	0,875	0,876
	3	78,0	79,3	78,7	78,9	77,7	0,875	0,876
	4	77,9	79,4	78,7	78,9	77,7	0,875	0,876
	5	77,8	79,4	78,6	78,8	77,5	0,875	0,876
	6	77,6	79,5	78,6	78,8	77,5	0,875	0,876
15	2	77,5	79,5	78,5	78,7	77,0	0,870	0,869
	3	46,3	79,2	62,8	78,7	77,0	0,870	0,869
	4	76,2	79,3	77,8	78,6	76,8	0,870	0,869
	5	76,1	79,3	77,7	78,6	76,6	0,870	0,869
	6	76,1	79,4	77,8	78,9	76,9	0,870	0,870

Tabla 5.38. Modelo Perceptrón Multicapa. Método BFGS según número de neuronas y ridge. Fase de test.

		Fase de test						
		Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Ridge	Número neuronas	SÍ	NO	Total	SÍ	NO	Total	Valor
0,01	**2**	57,1	100,0	76,9	100,0	66,7	84,6	0,895
	3	85,7	83,3	84,6	85,7	83,3	84,6	0,888
	4	78,6	100,0	88,5	100,0	80,0	90,8	0,957
	5	64,3	100,0	80,8	100,0	70,6	85,4	0,891
	6	71,4	100,0	84,6	100,0	75,0	88,5	0,875
0,05	**2**	64,3	100,0	80,8	100,0	70,6	86,4	0,927
	3	85,7	91,7	88,5	92,3	84,6	88,8	0,920
	4	71,4	100,0	84,6	100,0	75,0	88,5	0,926
	5	85,7	100,0	92,3	100,0	85,7	93,4	0,898
	6	78,6	100,0	88,5	100,0	80,0	90,8	0,979
0,1	**2**	71,4	83,3	76,9	83,3	71,4	77,8	0,929
	3	64,3	75,0	69,2	75,0	64,3	70,1	0,780
	4	78,6	91,7	84,6	91,7	78,6	85,6	0,958
	5	78,6	91,7	84,6	91,7	78,6	85,6	0,893
	6	78,6	100,0	88,5	100,0	80,0	90,8	0,952
0,2	**2**	57,1	83,3	69,2	80,0	62,5	71,9	0,887
	3	71,4	91,7	80,8	90,9	73,3	82,8	0,887
	4	85,7	91,7	88,5	92,3	84,6	88,8	0,917
	5	78,6	91,7	84,6	91,7	78,6	85,6	0,952
	6	78,6	91,7	84,6	91,7	78,6	85,6	0,911
1	**2**	64,3	91,7	76,9	90,0	68,8	80,2	0,893
	3	71,4	83,3	76,9	83,3	71,4	77,8	0,887
	4	71,4	91,7	80,8	90,9	73,3	82,8	0,911
	5	71,4	91,7	80,8	90,9	73,3	82,8	0,911
	6	71,4	91,7	80,8	90,9	73,3	82,8	0,911
2	**2**	64,3	91,7	76,9	90,0	68,8	80,2	0,905
	3	64,3	91,7	76,9	90,0	68,8	80,2	0,899
	4	64,3	91,7	76,9	90,0	68,8	80,2	0,905
	5	64,3	91,7	76,9	90,0	68,8	80,2	0,899
	6	64,3	91,7	76,9	90,0	68,8	80,2	0,905
10	**2**	71,4	100,0	84,6	100,0	75,0	88,5	0,899
	3	71,4	100,0	84,6	100,0	75,0	88,5	0,899
	4	71,4	100,0	84,6	100,0	75,0	88,5	0,899
	5	71,4	100,0	84,6	100,0	75,0	88,5	0,899
	6	71,4	100,0	84,6	100,0	75,0	88,5	0,899
15	**2**	71,4	100,0	84,6	100,0	75,0	88,5	0,911
	3	71,4	100,0	84,6	100,0	75,0	88,5	0,911
	4	71,4	100,0	84,6	100,0	75,0	88,5	0,911
	5	71,4	100,0	84,6	100,0	75,0	88,5	0,911
	6	71,4	100,0	84,6	100,0	75,0	88,5	0,911

Tabla 5.39. Comparación de modelos. Perceptrón Multicapa. Método BFGS.

Modelo base de contrastación: 5 neuronas y ridge = 0.1		Correctamente clasificados				Área Curva ROC	
Ridge	Número neuronas	Sí	Desviación estándar	NO	Desviación estándar	Valor	Desviación estándar
0,01	**2**	76,6	7,8	83,5	7,8	0,876	0,038
	3	76,8	6,9	81,2	7,2	0,856	0,041
	4	76,1	7,7	81,2	6,4	0,849 (*)	0,043
	5	76,6	6,6	80,5	6,4	0,855 (*)	0,044
	6	77,2	6,7	79,6	6,5	0,857 (*)	0,039
0,05	**2**	76,7	6,8	83,6	7,2	0,882	0,038
	3	76,8	7,2	82,3	7,0	0,869	0,041
	4	78,2	6,7	81,6	6,7	0,867	0,040
	5	77,4	7,0	82,7	6,3	0,870	0,039
	6	79,2	7,0	81,9	6,2	0,872	0,036
0,1	**2**	76,4	6,0	83,8	6,5	0,884	0,035
	3	77,1	7,4	82,9	5,9	0,877	0,037
	4	78,4	6,8	82,5	6,8	0,876	0,042
	5	78,9	6,6	82,8	6,6	0,882	0,035
	6	78,7	7,1	82,7	6,7	0,882	0,035
0,2	**2**	78,4	6,1	83,6	6,6	0,885	0,033
	3	78,2	7,5	83,3	6,7	0,888	0,037
	4	79,3	6,5	83,3	6,2	0,885	0,035
	5	79,2	7,0	83,3	6,2	0,885	0,033
	6	78,9	6,5	83,0	5,7	0,891	0,033
1	**2**	79,1	6,1	83,3	6,4	0,887	0,035
	3	79,2	6,4	81,6	6,8	0,891	0,034
	4	79,2	6,4	82,4	6,7	0,890	0,034
	5	79,2	6,3	82,0	6,7	0,889	0,035
	6	79,2	6,3	82,0	6,7	0,888	0,035
2	**2**	79,0	6,4	81,8	6,7	0,884	0,036
	3	79,1	6,6	81,8	6,5	0,883	0,036
	4	79,3	6,8	81,6	6,5	0,883	0,036
	5	79,2	6,8	81,5	6,5	0,883	0,036
	6	79,2	6,7	81,2	6,6	0,883	0,036
10	**2**	79,2	6,7	81,2	6,6	0,875	0,038
	3	78,0	6,4	79,3	6,7	0,875	0,037
	4	77,9	6,4	79,4	6,7	0,875	0,037
	5	77,8	6,3	79,4	6,7	0,875	0,037
	6	77,6	6,5	79,5	6,7	0,875	0,037
15	**2**	77,5	6,5	79,5	6,7	0,870	0,038
	3	46,3	6,8	79,2	6,3	0,870	0,038
	4	76,2	6,6	79,3	6,3	0,870	0,038
	5	76,1	6,5	79,3	6,3	0,870	0,039
	6	76,1	6,5	79,4	6,2	0,870	0,038

Nota: (*) Estadísticamente peor que el modelo base.

En relación a los modelos de Funciones de Base Radial considerados se ha efectuado la comparación entre ellos fijando como base el modelo de cinco neuronas en la capa oculta y un ridge igual a 0,1. Mientras que para la clase SÍ todos parecen predecirla de forma parecida no ocurre lo mismo con la clase NO. Tienen un desempeño mejor los modelos con un valor del parámetro ridge más pequeño (0,01) mientras que los mayores o iguales a uno son estadísticamente peores cuando predicen la clase NO y también cuando estiman el área bajo la curva ROC.

Tabla 5.40. Comparación de modelos. Funciones de Base Radial.

Modelo base de contrastación: 5 neuronas y ridge = 0.1		Correctamente clasificados				Área Curva ROC	
Ridge	Número neuronas	Sí	Desviación estándar	NO	Desviación estándar	Valor	Desviación estándar
0,01	2	81,1	6,2	79,4	6,4	0,888	0,035
	3	81,3	6,2	80,8 (v)	6,6	0,889	0,037
	4	81,0	6,3	81,3 (v)	6,9	0,889	0,036
	5	81,0	6,3	81,6 (v)	6,9	0,888	0,036
	6	81,0	6,3	81,7 (v)	6,7	0,887	0,037
0,05	2	82,7	5,7	77,3	7,4	0,884	0,037
	3	81,9	5,8	78,1	7,2	0,886	0,036
	4	81,7	6,0	78,7	6,8	0,886	0,037
	5	81,5	6,0	79,3	6,9	0,887	0,036
	6	81,4	6,0	80,0 (v)	6,6	0,888	0,035
0,1	2	83,0	5,4	76,6	7,5	0,881	0,038
	3	82,5	5,6	76,8	7,5	0,883	0,037
	4	82,6	5,6	77,3	7,6	0,884	0,037
	5	82,0	5,8	77,8	7,3	0,885	0,037
	6	81,8	5,8	78,1	7,2	0,886	0,036
0,2	2	83,3	5,5	76,2	7,3	0,877 (*)	0,039
	3	83,1	5,7	76,3	7,7	0,880	0,038
	4	83,0	5,4	76,5	7,6	0,881	0,038
	5	82,5	5,4	76,7	7,5	0,882 (*)	0,038
	6	82,5	5,5	76,8	7,6	0,883	0,037
1	2	83,7	5,9	74,4 (*)	7,3	0,864 (*)	0,042
	3	83,8	6,0	74,0 (*)	7,4	0,886 (*)	0,041
	4	83,8	5,9	74,1 (*)	7,7	0,869 (*)	0,041
	5	83,3	5,5	74,6 (*)	7,7	0,871 (*)	0,041
	6	83,3	5,5	74,9 (*)	7,7	0,873 (*)	0,040
2	2	83,4	6,0	74,6	7,3	0,859 (*)	0,043
	3	84,0	6,1	74,0 (*)	7,7	0,861 (*)	0,043
	4	83,9	6,2	74,0 (*)	7,7	0,863 (*)	0,042
	5	83,6	6,2	74,2 (*)	7,7	0,865 (*)	0,042
	6	83,5	6,0	74,1 (*)	7,3	0,866 (*)	0,041
10	2	83,5	6,3	74,0 (*)	7,6	0,854 (*)	0,044
	3	84,3	7,0	72,2 (*)	8,7	0,855 (*)	0,044
	4	84,5	6,9	72,9 (*)	8,5	0,856 (*)	0,044
	5	84,3	6,6	73,0 (*)	8,4	0,856 (*)	0,044
	6	84,0	6,2	73,3 (*)	8,2	0,856 (*)	0,044
15	2	83,6	6,5	73,8 (*)	7,7	0,853 (*)	0,044
	3	84,3	7,0	72,1 (*)	8,8	0,854 (*)	0,044
	4	84,3	7,0	72,8 (*)	8,5	0,855 (*)	0,044
	5	84,4	6,7	72,8 (*)	8,5	0,855 (*)	0,044
	6	83,9	6,4	73,1 (*)	8,2	0,855 (*)	0,044

Nota: (v) Estadísticamente mejor que el modelo base. (*) Estadísticamente peor.

En la figura 5.9. se muestra la estructura de una red neuronal con cinco neuronas en la capa oculta mientras que en la tabla 5.41 y en la figura 5.10 se muestra la importancia de las variables calculadas siguiendo los criterios del análisis de sensibilidad reseñados en el epígrafe 3.3.2.8.

Figura 5.9. Gráfico de una red neuronal con cinco neuronas en la capa oculta.

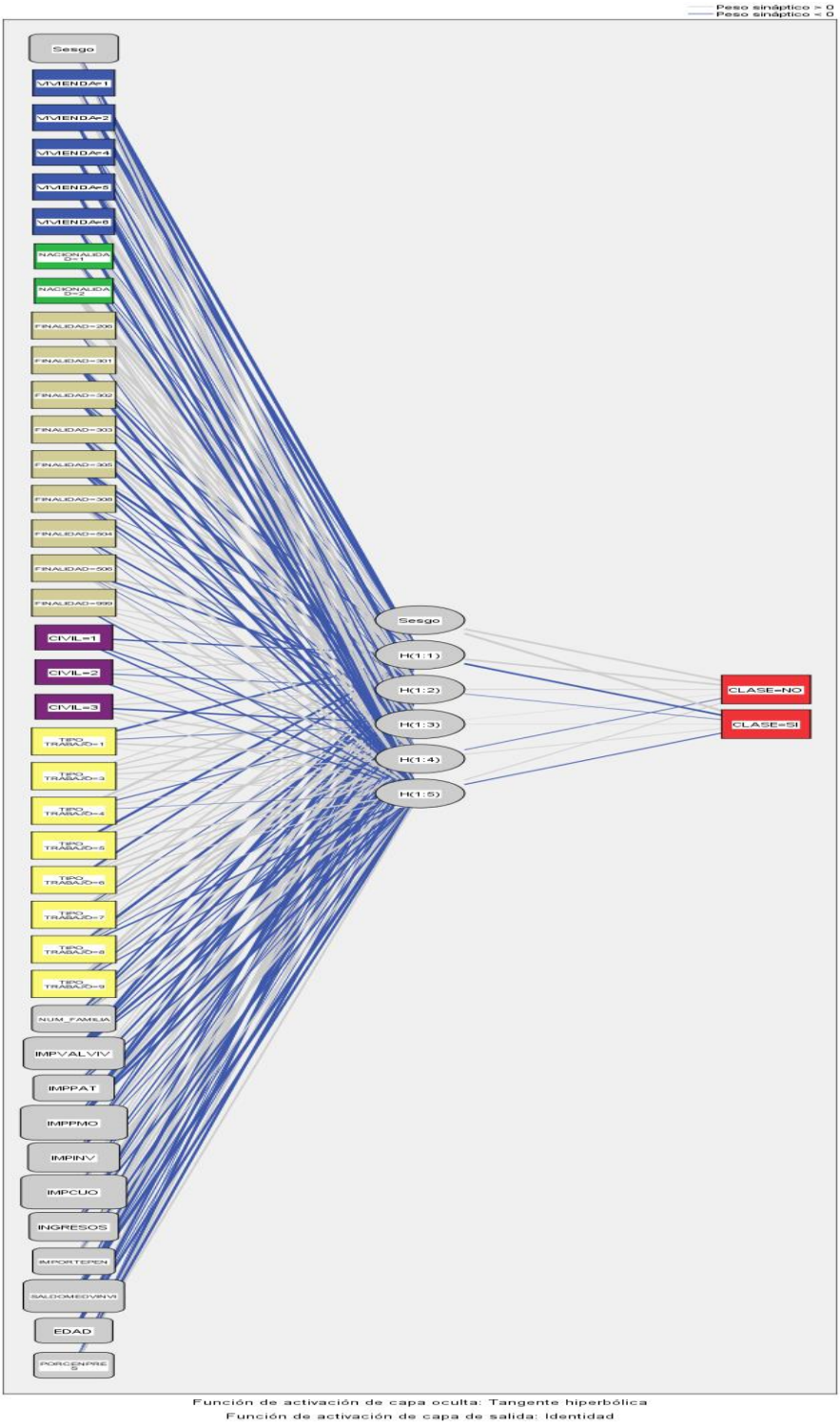
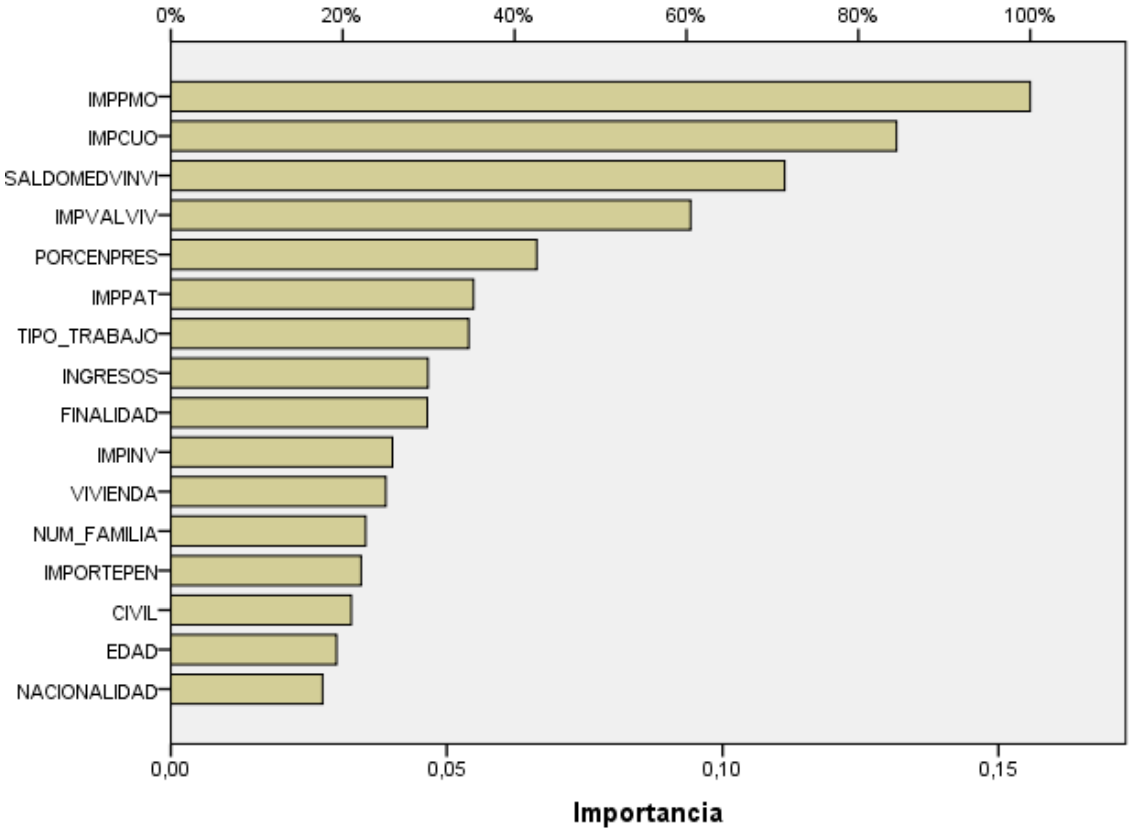


Tabla 5.41. Importancia de las variables independientes a través de un Perceptrón Multicapa.

	Importancia	Importancia normalizada
Tipo de vivienda	0,047	28,0%
Nacionalidad	0,040	23,8%
Finalidad del prestamo	0,052	30,7%
Estado civil	0,017	9,8%
Relación laboral	0,044	26,2%
Miembros de la familia	0,036	21,6%
Valor vivienda	0,048	28,4%
Valor Patrimonio	0,046	27,1%
Importe préstamo	0,166	98,5%
Importe inversión	0,064	38,0%
Importe cuota	0,168	100,0%
Ingresos	0,041	24,1%
Importes pendientes	0,041	24,2%
Saldo medio	0,106	63,0%
Edad	0,030	17,7%
Porcentaje prestado	0,055	32,8%

Figura 5.10. Importancia normalizada de las variables según el Perceptrón Multicapa.



5.2.3. Máquinas de Vectores Soporte.

Las máquinas de vectores soporte son modelos de clasificación que parecen tener mucho éxito cuando tratan de resolver problemas de clasificación y son eficientes en aquellas bases de datos donde se presentan problemas de no linealidad.

En esta tesis se ha trabajo con diversas funciones núcleo y se han ajustado dentro de ciertos intervalos sus parámetros de aprendizaje.

Como en todos los métodos utilizados en la tesis se presentan los porcentajes de clasificación para los dos valores de la clase, la precisión de estas estimaciones y la curva ROC, tanto para los registros de entrenamiento como para los de test.

En la tabla 5.42 y 5.43 se observan los resultados de la clasificación con un núcleo lineal y cuadrático. En la fase de simulación se ha trabajado con un intervalo que va desde el valor uno al diez del parámetro de aprendizaje. En la fase de entrenamiento el kernel lineal obtiene mejores resultados que el kernel cuadrático pero en la fase de test los resultados son mejores para el cuadrático.

Tabla 5.42. Comparación de modelos. Máquinas de Vectores Soporte. Polikernel lineal y cuadrático. Fase de Entrenamiento.

		Fase de entrenamiento						
		Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Modelos	Valores de C	SÍ	NO	Total	SÍ	NO	Total	Valor
Polikernel lineal	1,0	80,5	80,8	80,7	80,8	80,6	80,7	0,807
	1,2	80,2	80,2	80,2	80,2	80,2	80,2	0,802
	1,4	80,5	79,6	80,1	79,8	80,4	80,1	0,801
	1,6	80,5	80,2	80,4	80,3	80,5	80,4	0,804
	1,8	80,5	80,5	80,5	80,5	80,5	80,5	0,805
	2,0	80,6	80,8	80,7	80,8	80,6	80,7	0,807
	3,0	79,9	82,3	81,1	81,9	80,4	81,2	0,811
	6,0	80,2	83,5	81,9	83,0	80,9	81,9	0,819
	10,0	79,5	83,8	81,7	83,1	80,5	81,8	0,817
Polikernel cuadrático	1,0	77,2	80,5	78,9	79,9	78,0	78,9	0,789
	1,2	76,6	80,8	78,7	80,0	77,6	78,8	0,787
	1,4	76,9	80,2	78,6	79,6	77,7	78,6	0,786
	1,6	77,5	80,5	79,0	79,9	78,2	79,1	0,790
	1,8	77,2	81,1	79,2	80,4	78,1	79,2	0,792
	2,0	77,5	81,4	79,5	80,7	78,4	79,5	0,795
	3,0	78,1	79,3	78,7	79,1	78,4	78,7	0,787
	6,0	76,6	81,1	78,9	80,3	77,7	79,0	0,789
	10,0	77,2	81,4	79,3	80,6	78,2	79,4	0,793

Tabla 5.43. Comparación de modelos. Máquinas de Vectores Soporte. Polikernel lineal y cuadrático. Fase de Test.

		Fase de test						
		Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Modelos	Valores de C	SÍ	NO	Total	SÍ	NO	Total	Valor
Polikernel lineal	1,0	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	1,2	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	1,4	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	1,6	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	1,8	64,3	91,7	76,9	90,0	68,8	80,2	0,780
	2,0	64,3	91,7	76,9	90,0	68,8	80,2	0,780
	3,0	64,3	100,0	80,8	100,0	70,6	86,4	0,821
	6,0	71,4	100,0	84,6	100,0	75,0	88,5	0,857
	10,0	64,3	100,0	80,8	100,0	70,6	86,4	0,821
Polikernel cuadrático	1,0	71,4	91,7	80,8	90,9	73,3	82,8	0,815
	1,2	71,4	91,7	80,8	90,9	73,3	82,8	0,815
	1,4	71,4	91,7	80,8	90,9	73,3	82,8	0,815
	1,6	71,4	91,7	80,8	90,9	73,3	82,8	0,815
	1,8	71,4	91,7	80,8	90,9	73,3	82,8	0,815
	2,0	71,4	91,7	80,8	90,9	73,3	82,8	0,815
	3,0	78,6	91,7	84,6	91,7	78,6	85,6	0,851
	6,0	78,6	91,7	84,6	91,7	78,6	85,6	0,851
	10,0	85,7	91,7	88,5	92,3	84,6	88,8	0,887

También se ha probado con la función núcleo de Base Radial ensayando con los mismos valores del parámetro de aprendizaje al que se le añade otro parámetro que se denomina Gamma. Los resultados entre las clases, tanto en la fase de entrenamiento como en la de test, reflejan valores que cambian según el significado de la clase, tal como puede observarse en las tablas 5.44 y 5.45.

También se ha experimentado con otras dos funciones núcleo, el polikernel normalizado y otro basado en la función kernel universal de Pearson. Los diversos valores de clasificación se reflejan en las tablas de 5.46 a 5.49.

Con los resultados de los cinco modelos más significativos, para cada una de las funciones kernel utilizadas, se ha realizado un contraste estadístico y no se han encontrado evidencias estadísticamente significativas que puedan confirmar que alguno de ellos presente mejores resultados. Podemos afirmar entonces que ninguna de las funciones kernel utilizadas en las máquinas de vectores soporte presenta, con esta base de datos de créditos personales de clientes bancarios, un mejor desempeño en términos de clasificación estadística.

Tabla 5.44. Comparación de modelos. Máquinas de Vectores Soporte. RBF Kernel. Fase de Entrenamiento.

		Fase de Entrenamiento						
		Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Modelos	Valores de C	SÍ	NO	Total	SÍ	NO	Total	Valor
RBF Kernel Gamma=0,01	1,0	82,9	76,0	79,5	77,6	81,7	79,6	0,795
	1,2	83,2	75,7	79,5	77,4	81,9	79,7	0,795
	1,4	83,2	75,1	79,2	77,0	81,8	79,4	0,792
	1,6	83,5	74,9	79,2	76,9	82,0	79,4	0,792
	1,8	82,9	74,9	78,9	76,7	81,4	79,1	0,789
	2,0	82,9	74,9	78,9	76,7	81,4	79,1	0,789
	3,0	82,9	76,0	79,5	77,6	81,7	79,6	0,795
	6,0	79,9	79,3	79,6	79,5	79,8	79,6	0,796
	10,0	79,9	80,2	80,1	80,2	80,0	80,1	0,801
RBF Kernel Gamma=0,02	1,0	83,2	74,6	78,9	76,6	81,6	79,1	0,789
	1,2	82,6	75,1	78,9	76,9	81,2	79,1	0,789
	1,4	82,6	75,1	78,9	76,9	81,2	79,1	0,789
	1,6	82,6	76,6	79,6	78,0	81,5	79,7	0,796
	1,8	82,3	77,8	80,1	78,8	81,5	80,2	0,801
	2,0	81,4	78,7	80,1	79,3	80,9	80,1	0,801
	3,0	80,2	79,6	79,9	79,8	80,1	79,9	0,799
	6,0	80,2	79,9	80,1	80,0	80,2	80,1	0,801
	10,0	80,5	79,6	80,1	79,8	80,4	80,1	0,801
RBF Kernel Gamma=0,1	2,0	79,3	81,1	80,2	80,8	79,7	80,2	0,802
RBF Kernel Gamma=0,5	2,0	78,1	80,5	79,3	80,1	78,7	79,4	0,793

Tabla 5.45. Comparación de modelos. Máquinas de Vectores Soporte. RBF Kernel. Fase de Test.

		Fase de Test						
		Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Modelos	Valores de C	SÍ	NO	Total	SÍ	NO	Total	Valor
RBF Kernel Gamma=0,01	1,0	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	1,2	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	1,4	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	1,6	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	1,8	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	2,0	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	3,0	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	6,0	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	10,0	64,3	83,3	73,1	81,8	66,7	74,8	0,738
RBF Kernel Gamma=0,02	1,0	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	1,2	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	1,4	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	1,6	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	1,8	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	2,0	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	3,0	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	6,0	64,3	83,3	73,1	81,8	66,7	74,8	0,738
	10,0	64,3	83,3	73,1	81,8	66,7	74,8	0,738
RBF Kernel Gamma=0,1	2,0	57,1	83,3	69,2	80,0	62,5	71,9	0,702
RBF Kernel Gamma=0,5	2,0	78,6	91,7	84,6	91,7	78,6	85,6	0,851

Tabla 5.46. Comparación de modelos. Máquinas de Vectores Soporte. Polikernel normalizado. Fase de Entrenamiento.

		Fase de Entrenamiento						
		Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Modelos	**Valores de C**	**SÍ**	**NO**	**Total**	**SÍ**	**NO**	**Total**	**Valor**
Polikernel Normalizado	1,0	79,9	81,1	80,5	80,9	80,2	80,5	0,805
	1,2	79,9	81,4	80,7	81,2	80,2	80,7	0,807
	1,4	79,3	81,1	80,2	80,8	79,7	80,2	0,802
	1,6	79,0	81,7	80,4	81,2	79,6	80,4	0,804
	1,8	78,7	82,0	80,4	81,4	79,4	80,4	0,804
	2,0	78,7	82,6	80,7	81,9	79,5	80,7	0,807
	3,0	79,0	82,6	80,8	82,0	79,8	80,9	0,808
	6,0	79,0	82,6	80,8	82,0	79,8	80,9	0,808
	10,0	78,4	81,4	79,9	80,9	79,1	80,0	0,799

Tabla 5.47. Comparación de modelos. Máquinas de Vectores Soporte. Polikernel Normalizado. Fase de Test.

		Fase de Test						
		Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Modelos	**Valores de C**	**SÍ**	**NO**	**Total**	**SÍ**	**NO**	**Total**	**Valor**
Polikernel Normalizado	1,0	57,1	83,3	69,2	80,0	62,5	71,9	0,702
	1,2	64,3	91,7	76,9	90,0	68,8	80,2	0,780
	1,4	64,3	91,7	76,9	90,0	68,8	80,2	0,780
	1,6	64,3	91,7	76,9	90,0	68,8	80,2	0,780
	1,8	71,4	91,7	80,8	90,9	73,3	82,8	0,815
	2,0	71,4	91,7	80,8	90,9	73,3	82,8	0,815
	3,0	71,4	91,7	80,8	90,9	73,3	82,8	0,815
	6,0	71,4	91,7	80,8	90,9	73,3	82,8	0,815
	10,0	64,3	91,7	76,9	90,0	68,8	80,2	0,780

Tabla 5.48. Comparación de modelos. Máquinas de Vectores Soporte. Función Pearson VII (PUK). Fase de Entrenamiento.

		Fase de Entrenamiento						
		Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Modelos	**Valores de C**	**SÍ**	**NO**	**Total**	**SÍ**	**NO**	**Total**	**Valor**
Omega = 1 Sigma = 1	1,0	83,3	78,7	81,3	79,8	83,0	81,4	0,813
	1,2	83,2	79,0	81,1	79,9	82,5	81,2	0,811
	1,4	82,3	79,3	80,8	79,9	81,8	80,9	0,808
	1,6	82,0	79,9	81,0	80,4	81,7	81,0	0,810
	1,8	82,3	79,9	81,1	80,4	81,9	81,2	0,811
	2,0	82,0	79,9	81,0	80,4	81,7	81,0	0,81
	3,0	82,6	80,8	81,7	81,2	82,3	81,7	0,817
	6,0	82,3	79,9	81,1	80,4	81,9	81,2	0,811
	10,0	82,6	79,6	81,1	80,2	82,1	81,2	0,811
Omega = 1 Sigma = 2	2,0	79,9	80,8	80,4	80,7	80,1	80,4	0,804
Omega = 2 Sigma = 1	2,0	83,8	77,5	80,7	78,9	82,7	80,8	0,807
Omega = 2 Sigma = 2	2,0	79,6	80,2	79,9	80,1	79,8	79,9	0,799

Tabla 5.49. Comparación de modelos. Máquinas de Vectores Soporte. Función Person VII (PUK). Fase de Test.

		Fase de Entrenamiento						
		Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Modelos	**Valores de C**	**SÍ**	**NO**	**Total**	**SÍ**	**NO**	**Total**	**Valor**
Omega = 1 Sigma = 1	1,0	78,6	91,7	84,6	91,7	78,6	85,6	0,851
	1,2	78,6	91,7	84,6	91,7	78,6	85,6	0,851
	1,4	78,6	91,7	84,6	91,7	78,6	85,6	0,851
	1,6	78,6	91,7	84,6	91,7	78,6	85,6	0,851
	1,8	78,6	91,7	84,6	91,7	78,6	85,6	0,851
	2,0	78,6	91,7	84,6	91,7	78,6	85,6	0,851
	3,0	85,7	100,0	92,3	100,0	85,7	92,3	0,929
	6,0	85,7	100,0	92,3	100,0	85,7	92,3	0,929
	10,0	85,7	100,0	92,3	100,0	85,7	92,3	0,929
Omega = 1 Sigma = 2	2,0	78,6	91,7	84,6	91,7	78,6	85,6	0,851
Omega = 2 Sigma = 1	2,0	78,6	91,7	84,6	91,7	78,6	85,6	0,851
Omega = 2 Sigma = 2	2,0	78,6	91,7	84,6	91,7	78,6	85,6	0,851

5.2.4. Regresión logística

Para verificar si el modelo de regresión logística con las dieciséis variables seleccionadas presenta resultados aceptables tenemos que empezar observando la bondad del ajuste. En la tabla 5.50 se observan tres medidas que nos ayudan a saber hasta qué punto la regresión logística binaria se ajusta bien a los datos, es decir, evaluar de forma global la validez del modelo.

El primer valor se corresponde con el estadístico (-2LL), véase fórmula 3.171, que mide hasta qué punto un modelo se ajusta bien a los datos. El resultado de esta medición recibe también el nombre de desviación (devianza). Cuanto más pequeño sea el valor, mejor será el ajuste.

El segundo valor es el cálculo del R cuadrado de Cox y Snell, ver fórmula 3.191, y representa el coeficiente de determinación que calcula la proporción de varianza de la variable dependiente explicada por las variables predictoras (independientes). El R cuadrado de Cox y Snell se basa en la comparación del logaritmo de la verosimilitud (LL) para el modelo respecto al logaritmo de la verosimilitud para un modelo base. Los valores oscilan entre 0 y 1.

Una versión corregida del anterior R cuadrado es la que desarrolló Nagelkerke, fórmula 3.192, que ajusta la escala del estadístico para cubrir el rango completo entre cero y uno.

Tabla 5.50. Resumen del modelo de regresión logística.

Paso	-2 log de la verosimilitud	R cuadrado de Cox y Snell	R cuadrado de Nagelkerke
1	488,420[a]	0,481	0,641

a. La estimación ha finalizado en el número de iteración 7 porque las estimaciones de los parámetros han cambiado en menos de ,001.

Los valores que se observan de los coeficientes de determinación resultan un poco bajos.

Otra manera que disponemos para evaluar la bondad del ajuste de un modelo es la prueba de Hosmer-Lemeshow. La idea es que si el ajuste de los datos es bueno, un valor alto de la probabilidad predicha se asociará con el resultado “SÍ” de la variable binomial dependiente, mientras que un valor bajo de p “próximo a cero” corresponderá “en la mayoría de las veces” con el resultado “NO”.

Para cada observación del conjunto de datos, se calcula la probabilidad de la variable dependiente que predice el modelo. Un vez que estén ordenadas, las agrupamos y calculamos las frecuencias esperadas. Todas estas probabilidades se ordenan y se calculan los deciles. Con estas frecuencias estimadas y observadas podemos obtener el estadístico chi-cuadrado de Pearson (ver fórmula 3.181).

Tabla 5.51. Prueba de Hosmer y Lemeshow.

Paso	Chi cuadrado	gl	Sig.
1	18,986	8	0,015

Este estadístico de Bondad de ajuste utiliza los casos agrupados en deciles de riesgo, comparando la probabilidad observada con la probabilidad esperada dentro de cada decil. Se distribuye con 8 grados de libertad y está basado en los resultados salidos de la simulación según Tsiatis (1980). Este autor afirma que no siempre se cumplen los supuestos de esta distribución, especialmente cuando en algunos grupos los valores esperados y observados son nulos o muy pequeños (menores que cinco).

Para resolver este problema, Khan y Sempos (1989) sugieren la validez de cotejar valores esperados y observados mediante simple inspección visual y evaluar el grado de concordancia a partir del sentido común.

Tabla 5.52. Valores observados y esperados para la prueba de bondad de ajuste de Hosmer y Lemeshow.

	Concesión del crédito = NO		Concesión del crédito = SI		
	Observado	Esperado	Observado	Esperado	Total
1	66	65,551	1	1,449	67
2	60	62,244	7	4,756	67
3	62	58,698	5	8,302	67
4	54	51,955	13	15,045	67
5	38	41,337	29	25,663	67
6	28	28,361	39	38,639	67
7	17	16,209	50	50,791	67
8	7	6,849	60	60,151	67
9	0	2,548	67	64,452	67
10	2	0,248	63	64,752	65

Observando la tabla 5.52 existen algunos deciles donde las diferencias parecen ser excesivas, por ejemplo en los deciles 5, 6, 9 y 10 de la clase NO y los deciles, 3, 5 y 9 de las clase SÍ.

En la tabla número 5.53 se muestran los valores de los parámetros estimados junto a sus errores estándar, el test de significatividad y el valor del Exp(B).

Podemos ver qué variables son realmente importantes para la regresión logística atendiendo al estadístico de Wald, que contrasta si cada coeficiente de la regresión es significativamente diferente de cero:

$$Z_{Wald} = {}^{b}\!\!\diagup_{ET(b)}$$

Este estadístico sigue una distribución normal estándar. Observando la tabla 5.53 podemos descartar cinco variables que no superan el test estadístico a un nivel de significación del 5%. Estas variables son aquellas que en la columna sig (valor del p-value) superan el error del tipo I, 0,05: Número de miembros familiares, importe del valor de la vivienda, importe de la inversión, importes pendientes y edad.

Tabla 5.53. Valores de la regresión logística.

	B	E.T.	Wald	gl	Sig.	Exp(B)
NUM_FAMILIA	-0,014	0,112	0,015	1	0,902	0,986
VIVIENDA			19,621	4	0,001	
VIVIENDA(1)	2,372	0,742	10,221	1	0,001	10,716
VIVIENDA(2)	0,884	0,636	1,933	1	0,164	2,421
VIVIENDA(3)	-0,291	0,608	0,228	1	0,633	0,748
VIVIENDA(4)	0,439	0,630	0,485	1	0,486	1,551
IMPVALVIV	0,000	0,000	1,682	1	0,195	1,000
IMPPAT	0,000	0,000	0,094	1	0,760	1,000
NACIONALIDAD(1)	1,059	0,320	10,926	1	0,001	2,883
IMPPMO	0,000	0,000	7,454	1	0,006	1,000
IMPINV	0,000	0,000	0,000	1	1,000	1,000
IMPCUO	-0,004	0,001	12,820	1	0,000	0,996
FINALIDAD			28,438	8	0,000	
FINALIDAD(1)	-0,664	0,515	1,664	1	0,197	0,515
FINALIDAD(2)	0,030	0,469	0,004	1	0,948	1,031
FINALIDAD(3)	0,672	0,809	0,690	1	0,406	1,958
FINALIDAD(4)	1,016	0,608	2,790	1	0,095	2,763
FINALIDAD(5)	0,282	0,668	0,178	1	0,674	1,325
FINALIDAD(6)	-0,028	0,675	0,002	1	0,966	0,972
FINALIDAD(7)	0,472	0,676	0,486	1	0,486	1,603
FINALIDAD(8)	-1,438	0,519	7,694	1	0,006	0,237
INGRESOS	0,000	0,000	8,459	1	0,004	1,000
IMPORTEPEN	0,000	0,000	0,803	1	0,370	1,000
SALDOMEDVINVI	0,000	0,000	18,030	1	0,000	1,000
EDAD	-0,008	0,014	0,328	1	0,567	0,992
PORCENPRES	0,001	0,004	0,083	1	0,774	1,001
CIVIL			4,371	2	0,112	
CIVIL(1)	0,765	0,366	4,370	1	0,037	2,150
CIVIL(2)	0,372	0,542	0,472	1	0,492	1,451
TIPO_TRABAJO			30,068	7	0,000	
TIPO_TRABAJO(1)	1,017	0,742	1,877	1	0,171	2,764
TIPO_TRABAJO(2)	-0,480	0,635	0,570	1	0,450	0,619
TIPO_TRABAJO(3)	-0,968	0,648	2,230	1	0,135	0,380
TIPO_TRABAJO(4)	0,828	0,807	1,052	1	0,305	2,288
TIPO_TRABAJO(5)	0,864	1,012	0,730	1	0,393	2,373
TIPO_TRABAJO(6)	-0,870	0,743	1,371	1	0,242	0,419
TIPO_TRABAJO(7)	-0,602	0,859	0,491	1	0,484	0,548
Constante	-1,139	1,148	0,984	1	0,321	0,320

La tabla número 5.54 presenta los resultados de la clasificación del modelo. El porcentaje correctamente clasificado es del 86,8%, donde el porcentaje bien clasificado para la clase SÍ, 87,5%, es ligeramente mayor que el de la clase NO, 86,1%.

Tabla 5.54. Tabla de clasificación de la regresión logística.

Observado			Pronosticado					
			Muestra entrenamiento			Casos test		
			Concesión del crédito			Concesión del crédito		
			NO	SI	Porcentaje correcto	NO	SI	Porcentaje correcto
		NO	283	51	84,7	10	2	83,3
	Concesión del crédito	SI	58	276	82,6	5	9	64,3
	Porcentaje global				83,7			73,1

En la regresión logística se puede realizar una selección hacia adelante y también hacia atrás a través de varios procedimientos. Utilizando el programa SPSS, con los diferentes métodos disponibles, se han descartado las siguientes variables (tabla 5.55).

Tabla 5.55. Valores que no están en la ecuación de la regresión logística.

Variables que no están en la ecuación			
	Puntuación	gl	Sig.
NUM_FAMILIA	0,007	1	0,935
IMPVALVIV	1,557	1	0,212
IMPPAT	0,256	1	0,613
IMPINV	0,572	1	0,449
IMPORTEPEN	0,546	1	0,460
EDAD	0,048	1	0,827
PORCENPRES	0,618	1	0,432

Cuando llevamos a cabo la regresión logística sin estas siete variables los resultados que obtenemos en términos de clasificación son exactamente iguales ya que las variables suprimidas no aportan nada. Los coeficientes de los parámetros estimados se han modificado ligeramente (véase tabla 5.56):

Tabla 5.56. Valores de la regresión logística ajustada.

	B	E.T.	Wald	gl	Sig.	Exp(B)
VIVIENDA			30,000	4	,000	
VIVIENDA(1)	2,610	,700	13,895	1	,000	13,601
VIVIENDA(2)	,959	,581	2,728	1	,099	2,610
VIVIENDA(3)	-,290	,610	,226	1	,634	,748
VIVIENDA(4)	,465	,624	,556	1	,456	1,593
NACIONALIDAD(1)	1,078	,305	12,539	1	,000	2,940
IMPPMO	,000	,000	32,950	1	,000	1,000
IMPCUO	-,004	,001	13,482	1	,000	,996
FINALIDAD			28,443	8	,000	
FINALIDAD(1)	-,674	,505	1,779	1	,182	,510
FINALIDAD(2)	-,006	,462	,000	1	,990	,994
FINALIDAD(3)	,579	,808	,513	1	,474	1,784
FINALIDAD(4)	,984	,601	2,676	1	,102	2,674
FINALIDAD(5)	,274	,662	,172	1	,679	1,315
FINALIDAD(6)	-,063	,667	,009	1	,925	,939
FINALIDAD(7)	,444	,670	,439	1	,507	1,559
FINALIDAD(8)	-1,450	,512	8,023	1	,005	,234
INGRESOS	,000	,000	7,073	1	,008	1,000
SALDOMEDVINVI	,000	,000	18,544	1	,000	1,000
CIVIL			6,203	2	,045	
CIVIL(1)	,753	,303	6,170	1	,013	2,124
CIVIL(2)	,319	,531	,360	1	,548	1,376
TIPO_TRABAJO			32,190	7	,000	
TIPO_TRABAJO(1)	,988	,723	1,868	1	,172	2,687
TIPO_TRABAJO(2)	-,489	,623	,616	1	,433	,613
TIPO_TRABAJO(3)	-1,030	,636	2,620	1	,106	,357
TIPO_TRABAJO(4)	,824	,790	1,087	1	,297	2,279
TIPO_TRABAJO(5)	,838	,995	,709	1	,400	2,312
TIPO_TRABAJO(6)	-,880	,730	1,454	1	,228	,415
TIPO_TRABAJO(7)	-,722	,818	,779	1	,377	,486
Constante	-1,377	,898	2,354	1	,125	,252

La estimación de los parámetros de la regresión logística se puede realizar a través de funciones núcleo (kernel logistic regression model). El modelo se ajusta minimizando el logaritmo de la máxima verosimilitud utilizando también una función cuadrática de penalización, Le Cessie y Van (1992).

Los modelos que se han estimado han utilizado diversas funciones núcleo: polikernel lineal, cuadrático, normalizado lineal y cuadrático y funciones de base radial con diferentes valores de los parámetros que necesitan. Los resultados del porcentaje de aciertos y de errores y las curvas de los modelos, tanto para los datos de entrenamiento como de test se pueden observar en las tablas 5.57 y 5.58.

Tabla 5.57. Regresión logística a través de estimadores ridge y funciones núcleo. Fase de entrenamiento.

Regresión Logística	Fase de entrenamiento						
	Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
	SÍ	NO	Total	SÍ	NO	Total	Valor
Simple	81,7	81,4	81,6	81,5	81,7	81,6	0,887
Ridge	80,5	82,9	81,7	82,5	81,0	81,8	0,894
Polkernel lineal	80,5	82,6	81,6	82,3	80,9	81,6	0,894
Polikernel cuadrático	75,4	75,1	75,3	75,2	75,4	75,3	0,812
Polikernel normalizado	79,3	79,9	79,6	79,8	79,5	79,6	0,874
PUK	92,5	64,7	78,6	72,4	89,6	81,0	0,894
Funciones de base radial	81,7	83,2	82,5	83,0	82,0	82,5	0,903

Tabla 5.58. Regresión logística a través de estimadores ridge y funciones núcleo. Fase de Test.

	Fase de test						
	Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
	SÍ	NO	Total	SÍ	NO	Total	Valor
Simple	71,4	91,7	80,8	90,9	73,3	82,8	0,905
Ridge	64,3	83,3	73,1	81,8	66,7	74,8	0,905
Polkernel lineal	64,3	83,3	73,1	81,8	66,7	74,8	0,905
Polikernel cuadrático	85,7	100,0	92,3	100,0	85,7	93,4	0,881
Polikernel normalizado	85,7	83,3	84,6	85,7	83,3	84,6	0,917
PUK	92,9	100,0	96,2	100,0	92,3	96,4	1,000
Funciones de base radial	78,6	91,7	84,6	91,7	78,6	85,6	0,929

El test que compara los porcentajes de los diferentes modelos de regresión logística indica que estos modelos no son estadísticamente significativos con un nivel de confianza del 95%, en relación con el modelo base de la regresión logística simple, excepción hecha al modelo estimado a través de un kernel cuadrático que difiere significativamente en el valor del área bajo la curva ROC, y la función de Pearson (PUK) cuyo porcentaje correctamente clasificado de personas que devuelven el crédito es significativamente mejor que el modelo de regresión logística simple.

En la Tabla 5.59 se muestran tanto los valores estimados como su desviación típica. En el modelo que ha sido rechazado por el contraste estadístico se observan los valores estimados más bajos y los errores estándar más elevados.

Tabla 5.59. Comparación de modelos de regresión logística.

Modelo base de contrastación: regresión logística simple	Correctamente clasificados				Área Curva ROC	
	SI	Desviación estándar	NO	Desviación estándar	Valor	Desviación estándar
Simple	80,6	6,2	82,5	6,1	0,890	0,030
Ridge	79,4	6,4	82,8	6,3	0,890	0,030
Polkernel lineal	79,4	6,4	82,7	6,3	0,890	0,030
Polikernel cuadrático	74,1	9,1	76,2	6,6	0,807 (*)	0,050
Polikernel normalizado	77,6	7,3	80,4	6,5	0,864	0,040
PUK	92,5 (v)	4,4	65,0	8,3 (*)	0,896	0,040
Funciones de base radial	81,6	5,9	82,7	6,9	0,897	0,030

Nota: (v) Estadísticamente mejor que el modelo base. (*) Estadísticamente peor.

5.2.5. Redes bayesianas.

La redes bayesianas son modelos probabilísticos que nos permiten representar el conocimiento de forma gráfica y compacta utilizando los conceptos de probabilidad y causalidad entre las variables del problema de credit scoring tratado en esta tesis.

Estos modelos bayesianos arrojan unos resultados excelentes. La aplicabilidad de este tipo de modelos para el análisis de los datos y para la comprensión de las relaciones entre las variables es más que evidente. En el análisis estadístico, tanto a nivel descriptivo como explicativo, resultan muy atractivas en un amplio campo de las investigaciones científicas.

En este epígrafe seguiremos una exposición de resultados acorde con la presentación realizada en la parte teórica del capítulo tres donde se describen los algoritmos y modelos cuyos resultados se detallan en los siguientes gráficos y tablas. Partimos de las estructuras más sencillas, pero que no por ello resultan ser peores clasificadores respecto a modelos más complejos, para al final realizar, a través del Manto de Markov, una selección de variables de acuerdo a este planteamiento.

En las tablas 5.60 y 5.61 se muestran los resultados tanto para la muestra de entrenamiento como para la de test para la estructura más simple de Naïve Bayes y para otras especificaciones desarrolladas posteriormente y basadas en este primer modelo.

Tabla 5.60. Naïve Bayes y otros modelos. Fase de Entrenamiento.

	Fase de entrenamiento						
	Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Algorimo de búsqueda/Modelo	SÍ	NO	Total	SÍ	NO	Total	Valor
Naïves Bayes	55,4	88,9	72,2	83,3	66,6	75,0	0,853
Naïves Bayes con estimador Kernel	60,5	93,4	76,9	90,2	70,3	80,2	0,883
Naïves Bayes con discretización	82,3	82,9	82,6	82,8	82,4	82,6	0,907
Average One Dependec Estimator (AODE)							
AODE 1	80,8	87,4	84,1	86,5	82,0	84,3	0,925
AODE 2	82,0	88,0	85,0	87,3	83,1	85,2	0,929
NBTree	79,6	82,8	81,3	82,4	80,3	81,3	0,879
TAN (Tree Augmented Network)	83,2	84,4	83,8	84,2	83,4	83,8	0,919

El modelo de Naïve Bayes, que supone que todos los atributos son independientes conocido el valor de la variable clase, obtiene unos resultados muy mediocres: las estimaciones de los porcentajes de acierto por clases están muy descompasadas, la precisión de la clase NO es muy baja, al igual que el valor del área bajo la curva ROC.

Cuando se estiman los porcentajes de acierto a través de una función Kernel, los valores son ligeramente más elevados pero aún se mantienen las estimaciones por clases muy desequilibradas.

Sin embargo, cuando la estimación se realiza discretizando los valores de las variables continuas, las proporciones de acierto por clases resultan ser muy elevadas, así como la precisión por clases y también el valor de la curva ROC, que alcanza niveles muy aceptables.

Tabla 5.61. Naïve Bayes y otros modelos. Fase de Test.

	Fase de test						
	Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Algorimo de búsqueda/Modelo	SÍ	NO	Total	SÍ	NO	Total	Valor
Naïves Bayes	35,7	91,7	61,5	83,3	55,0	70,3	0,893
Naïves Bayes con estimador Kernel	35,7	100,0	65,4	100,0	57,1	80,2	0,964
Naïves Bayes con discretización	85,7	91,7	88,5	92,3	84,6	88,8	0,929
Average One Dependec Estimator (AODE)							
AODE 1	78,6	100,0	88,5	100,0	80,0	90,8	0,964
AODE 2	85,7	100,0	92,3	100,0	85,7	93,4	0,958
NBTree	78,6	83,3	80,8	84,6	76,9	81,1	0,940
TAN (Tree Augmented Network)	71,4	83,3	76,9	83,3	71,4	77,8	0,911

El resto de modelos considerados en este primer grupo de clasificadores bayesianos también obtienen unos resultados muy aceptables, tanto en la muestra de entrenamiento como en la de test. Especialmente significativo resulta la estructura AODE (Average One Dependence Estimator) que consigue clasificar bien al 80,2% de los que devuelven el crédito y al 85,0 de los morosos y su curva ROC alcanza un valor de 0,929. Este modelo en la fase de test clasifica correctamente a todos los registros de la clase NO y al 85,7% de la clase SÍ.

Otro modelo muy atractivo es el TAN (Tree Augmented Network) que en la fase de entrenamiento alcanza resultados equivalentes al AODE2 pero que obtiene peores predicciones en la fase de test.

Al realizar la contrastación estadística entre todos los modelos considerados, estableciendo como modelo de comparación el AODE2, se obtiene que este modelo es mejor que todos los que se han contrastado salvo con el TAN, que no difiere significativamente ni en los porcentajes de acierto ni en el valor de la curva ROC.

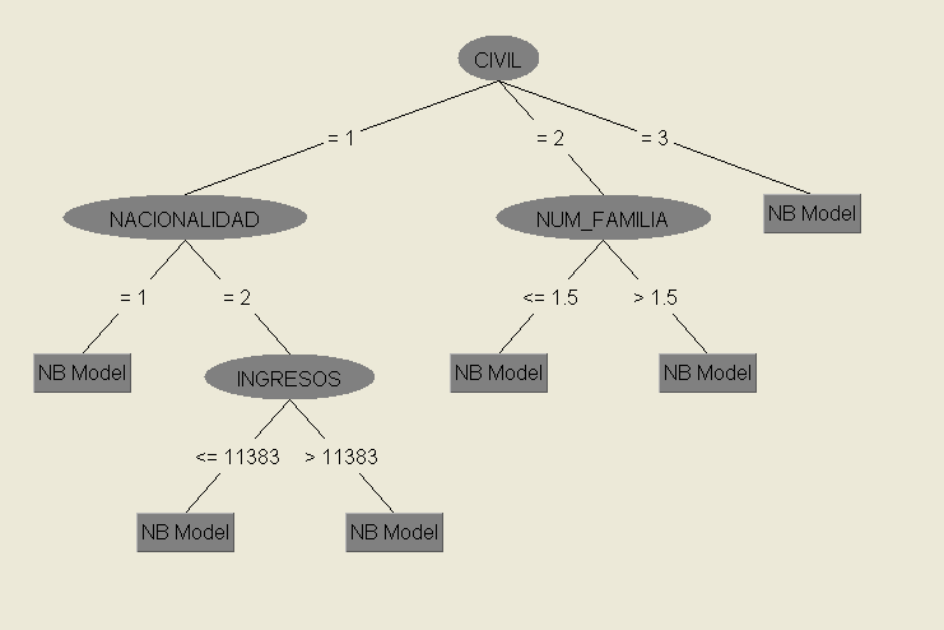
Tabla 5.62. Comparación de modelos: Naïve Bayes y variaciones.

	Correctamente clasificados				Área Curva ROC	
Modelo base de contrastacion: AODE 2	**SÍ**	**Desviación estándar**	**NO**	**Desviación estándar**	**Valor**	**Desviación estándar**
Naïve Bayes	54,4 (*)	7,8	88,5	5,6	0,852 (*)	0,04
Naïve Bayes Kernel	59,4 (*)	7,9	93,5 (v)	4,3	0,883 (*)	0,04
Naïve Bayes Discretización	82,0	6,2	83,3 (*)	6,2	0,907 (*)	0,03
AODE 1	80,7	6,3	86,9	5,5	0,922 (*)	0,03
AODE 2	81,8	6,5	87,3	5,7	0,925	0,03
Naïve Bayes Tree	79,8 (*)	6,9	82,1 (*)	6,9	0,883 (*)	0,05
TAN	83,0	5,8	84,0	6,4	0,918	0,03

Nota: (v) Estadísticamente mejor que el modelo base. (*) Estadísticamente peor.

En la figura 5.11 se muestra el grafico del Naïve Bayes extendido a un árbol de clasificación.

Figura 5.11. Estructura de red de Naïve Bayes Tree.



Los resultados de los modelos de redes bayesianas aplicando diferentes estrategias de búsqueda para obtener la estructura de dependencias/independencias se visualizan en las tablas 5.63 y 5.64.

Se han considerado los siguientes algoritmos de búsqueda: K2, ascensión de colinas (Hill Climber), recocido simulado (Simulted Anneling), Búsqueda Tabú (Tabu Search) y algunas variantes del algoritmo de ascensión de colinas (Lagd Hill Climber y Repeated Hill Climber). Todos los resultados se presentan con una estructura de los nodos condicionada a uno y dos padres y con arcos reversos, cuando el diseño de los algoritmos de búsqueda lo permiten.

Tabla 5.63. Redes bayesianas. Resultados con diferentes algoritmos de búsqueda. Fase de Entrenamiento.

	Fase de entrenamiento						
	Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Algorimo de búsqueda/Modelo	SÍ	NO	Total	SÍ	NO	Total	Valor
K2 con 1 padres	82,3	83,5	82,9	83,3	82,5	82,9	0,908
K2 con 2 padres	82,6	84,4	83,5	84,1	82,9	83,5	0,919
Hill Climber con 1 padre	82,3	83,5	82,9	83,3	82,5	82,9	0,908
Hill Climber con 2 padres	82,9	84,7	83,8	84,5	83,2	83,8	0,922
Con arcos reversos							
Hill Climber con 1 padre	82,3	83,5	82,9	83,3	82,5	82,9	0,908
Hill Climber con 2 padres	82,9	84,7	83,8	84,5	83,2	83,8	0,922
Simulated Anneling	80,5	84,4	82,5	83,8	81,3	82,5	0,914
Tabu Search con 1 padre	82,3	83,5	82,9	83,3	82,5	82,9	0,907
Tabu Search con 2 padres	83,2	83,5	83,4	83,5	83,3	83,4	0,917
Con arcos reversos							
Tabu Search con 1 padre	82,3	83,5	82,9	83,3	82,5	82,9	0,907
Tabu Search con 2 padres	83,2	83,5	83,4	83,5	83,3	83,4	0,917
LagD Hill Climber con 1 padre	82,9	82,9	82,9	82,9	82,9	82,9	0,911
LagD Hill Climber con 2 padres	82,6	84,4	83,5	84,1	82,9	83,5	0,923
Con arcos reversos							
LagD Hill Climber con 1 padre	82,3	83,5	82,9	83,3	82,5	82,9	0,9
LagD Hill Climber con 2 padres	82,9	83,8	83,4	83,7	83,3	83,4	0,920
Repeated Hill Climber con 1 padre	82,3	83,5	82,9	83,3	82,5	82,9	0,908
Repeated Hill Climber con 2 padres	82,3	85,0	83,7	84,6	82,8	83,7	0,918
Con arcos reversos							
Repeated Hill Climber con 1 padre	82,3	83,5	82,9	83,3	82,5	82,9	0,908
Repeated Hill Climber con 2 padres	82,3	84,7	83,5	84,4	82,7	83,6	0,918

Tabla 5.64. Redes bayesianas. Resultados con diferentes algoritmos de búsqueda. Fase de Test.

	Fase de test						
	Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
Algorimo de búsqueda/Modelo	SÍ	NO	Total	SÍ	NO	Total	Valor
K2 con 1 padres	85,7	91,7	88,5	92,3	84,6	88,8	0,929
K2 con 2 padres	71,4	83,3	76,9	83,3	71,4	77,8	0,893
Hill Climber con 1 padre	85,7	91,7	88,5	92,3	84,6	88,8	0,929
Hill Climber con 2 padres	71,4	100,0	84,9	100,0	75,0	88,5	0,887
Con arcos reversos							
Hill Climber con 1 padre	85,7	91,7	88,5	92,3	84,6	88,8	0,929
Hill Climber con 2 padres	71,4	100,0	84,6	100,0	75,0	88,5	0,887
Simulated Anneling	78,6	91,7	84,6	91,7	78,6	85,6	0,869
Tabu Search con 1 padre	85,7	91,7	88,5	92,3	84,6	88,8	0,929
Tabu Search con 2 padres	78,6	83,3	80,8	84,6	76,9	81,1	0,893
Con arcos reversos							
Tabu Search con 1 padre	85,7	91,7	88,5	92,3	84,6	88,8	0,929
Tabu Search con 2 padres	78,6	83,3	80,8	84,6	76,9	81,1	0,893
LagD Hill Climber con 1 padre	78,6	91,7	84,6	91,7	78,6	85,6	0,923
LagD Hill Climber con 2 padres	71,4	100,0	84,6	100,0	75,0	88,5	0,887
Con arcos reversos							
LagD Hill Climber con 1 padre	85,7	91,7	88,5	92,3	84,6	88,8	0,929
LagD Hill Climber con 2 padres	71,4	91,7	80,8	100,0	75,0	88,5	0,887
Repeated Hill Climber con 1 padre	85,7	91,7	88,5	92,3	84,6	88,8	0,929
Repeated Hill Climber con 2 padres	78,6	100,0	88,5	100,0	80,0	90,8	0,958
Con arcos reversos							
Repeated Hill Climber con 1 padre	85,7	91,7	88,5	92,3	84,6	88,8	0,929
Repeated Hill Climber con 2 padres	78,6	100,0	88,5	100,0	80,0	90,8	0,958

En general se confirma que todos los modelos de redes bayesianas obtienen unos resultados que podemos considerar excelentes. La proporción de aciertos de la clase que no devuelven el crédito es, en la mayor parte de los modelos, ligeramente superior a los que pagan el crédito concedido. El área bajo la curva ROC de todos los modelos es superior al 0,9 lo que se puede considerarse como una estimación muy buena.

Para la selección de los modelos más eficientes se recurre a la teoría de la contrastación estadística. Los resultados de los modelos seleccionados junto a los porcentajes de clasificación correcta, para ambas clases, acompañados de sus respectivas desviaciones estándar, así como del área bajo la curva ROC se muestran en la tabla 5.65.

La conclusión que se deduce es que, en términos de estimación de las proporciones de créditos fallidos y créditos pagados, ninguno de los modelos estudiados se puede considerar estadísticamente mejor que otro. Sin embargo, cuando se lleva a cabo el contraste con el valor obtenido de la curva ROC para cada uno de los modelos,

respecto al modelo base de contrastación que es la estructura de red obtenida a través del algoritmo de ascensión de colinas, algunos de ellos resultan estadísticamente peores (los señalados con un asterisco en la tabla de resultados).

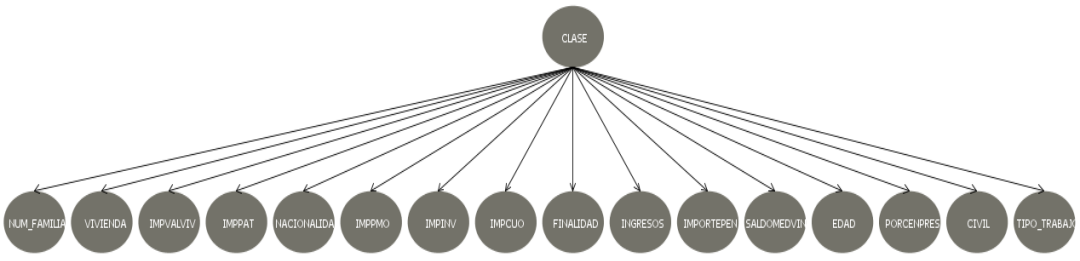
Tabla 5.65. Comparación de modelos: Redes Bayesianas con diferentes algoritmos de búsqueda.

	Correctamente clasificados				Área Curva ROC	
Modelo base de contrastacion: Hill Climber 2 padres	**SI**	**Desviación estándar**	**NO**	**Desviación estándar**	**Valor**	**Desviación estándar**
K2 con 1 padres	82,0	6,0	83,5	5,9	0,908 (*)	0,03
K2 con 2 padres	82,4	6,6	83,3	6,5	0,918	0,03
Hill Climber con 1 padre	82,0	6,2	83,5	5,9	0,908 (*)	0,03
Hill Climber con 2 padres	83,1	6,1	84,1	5,9	0,922	0,03
Simulated Anneling	82,0	6,9	84,6	6,1	0,915	0,03
Tabu Search con 1 padre	82,0	6,2	83,6	5,9	0,908 (*)	0,03
Tabu Search con 2 padres	82,7	6,2	83,3	6,2	0,916	0,03
LagD Hill Climber con 1 padre	82,6	6,0	83,2	6,4	0,909	0,03
LagD Hill Climber con 2 padres	83,0	6,2	84,3	5,8	0,922	0,03
Repeated Hill Climber con 1 padre	82,0	62,0	83,5	5,9	0,908 (*)	0,03
Repeated Hill Climber con 2 padres	82,2	5,9	84,1	6,5	0,917	0,03

Nota: (*) Estadísticamente peor que el modelo base.

A continuación, en la figura 5.18 y en la tabla 5.66, se visualizan la estructura gráfica y la tabla de probabilidades que genera una red bayesiana estimada a través del algoritmo de búsqueda denominado de ascensión de colinas, con un único padre.

Figura 5.12. Red Bayesiana. Hill Climber con 1 padre.



La riqueza y expresividad de las redes bayesianas se muestra, aparte de en su estructura gráfica, en su tabla de probabilidades. En la tabla 5.66 se han estimado las probabilidades de todas las variables explicativas de la concesión de créditos condicionadas a la variable clase.

En esta tabla de probabilidades condicionadas de las variables explicativas se encuentra información muy interesante para los responsables de la gestión de credit scoring.

Tabla 5.66. Estimaciones del valor de la probabilidad condicionada. Hill Climber con 1 padre.

NÚMERO DE MIEMBROS FAMILIARES				
CLASE	**Uno**	**Dos**	**Mayor de 2**	**Total**
SÍ	0,276	0,001	0,723	1
NO	0,404	0,171	0,425	1
IMPORTE DEL PRÉSTAMO				
CLASE	**Menor de 1.471**	**Entre 1.471 y 5.973**	**Mayor de 5.973**	**Total**
SÍ	0,165	0,139	0,696	1
NO	0,088	0,592	0,320	1
SALDO MEDIO DE LAS CUENTAS				
CLASE	**Menor de 453**	**Entre 454 y 4.543**	**Mayor de 4.543**	Total
SÍ	0,329	0,389	0,282	1
NO	0,815	0,168	0,016	1
ESTADO CIVIL				
CLASE	**Casado**	**Separado**	**Soltero**	Total
SÍ	0,550	0,055	0,395	1
NO	0,207	0,046	0,747	1
EDAD				
CLASE	**Menor de 50**	**Mayor o igual que 50**	Total	
SÍ	0,619	0,381	1	
NO	0,879	0,121	1	
IMPORTES PENDIENTES				
CLASE	**Menor de 4.150**	**Mayor o igual de 4.150**	**Total**	
SÍ	0,575	0,425	1	
NO	0,781	0,219	1	
VALOR DE LA VIVIENDA				
CLASE	**Menor de 27.112**	**Mayor o igual de 27.112**	**Total**	
SÍ	0,330	0,670	1	
NO	0,819	0,181	1	
NACIONALIDAD				
CLASE	**Español**	**Extranjero**	**Total**	
SÍ	0,897	0,103	1	
NO	0,488	0,512	1	
IMPORTE DEL PATRIMONIO				
CLASE	**Menor de 4.500**	**Mayor o igual de 4.500**	Total	
SÍ	0,810	0,190	1	
NO	0,942	0,058	1	
INGRESOS				
CLASE	**Menor de 23.210**	**Mayor o igual de 23.210**	**Total**	
SÍ	0,703	0,297	1	
NO	0,912	0,088	1	

Tabla 5.66. Estimaciones del valor de la probabilidad condicionada Hill Climber con 1 padre. (Continuación).

IMPORTE DE LA INVERSIÓN		
CLASE	**SÍ**	**NO**
Menor de 565	0,001	0,025
Entre 565 y 1.484	0,165	0,043
Entre 1.485 y 5.999	0,117	0,575
Entre 6.000 y 8.712	0,126	0,147
Mayor de 8.712	0,590	0,210
Total	1	1

TIPO DE VIVIENDA		
CLASE	**SÍ**	**NO**
Propiedad libre de cargas	0,308	**0,03**
Propiedad hipotecada	0,379	0,204
Alquiler	0,088	0,510
Domicilio con la familia	0,186	0,204
Otros	0,040	0,055
Total	1	1

IMPORTE DE LA CUOTA		
CLASE	**SÍ**	**NO**
Menos de 41	0,114	0,007
Entre 41 y 81	0,105	0,070
Entre 82 y 241	0,337	0,753
Entre 242 y 250	0,215	0,010
Mayor de 250	0,227	0,159
Total	1	1

FINALIDAD DEL CRÉDITO		
CLASE	**SÍ**	**NO**
Reformas viviendas	0,182	0,149
Compra automóvil	0,338	0,300
Compra electrodomésticos	0,049	0,022
Compra ordenador	0,114	0,034
mobiliario y decoración	0,072	0,034
Otros bien y servicios	0,046	0,028
Servicios sanitarios	0,055	0,040
Imprevistos familiares	0,081	0,303
otras finalidades	0,064	0,090
Total	1	1

TIPO DE TRABAJO (RELACIÓN LABORAL)		
CLASE	**SÍ**	**NO**
Técnico-mando internmedio	0,271	0,049
Obrero fijo	0,291	0,368
Obrero temporal	0,099	0,419
Obrero fijo especiaizado	0,099	0,037
Obrero temporal especializado	0,019	0,010
Autónomo	0,096	0,067
Jubilado rentista	0,067	0,022
No activo	0,058	0,028
Total	1	1

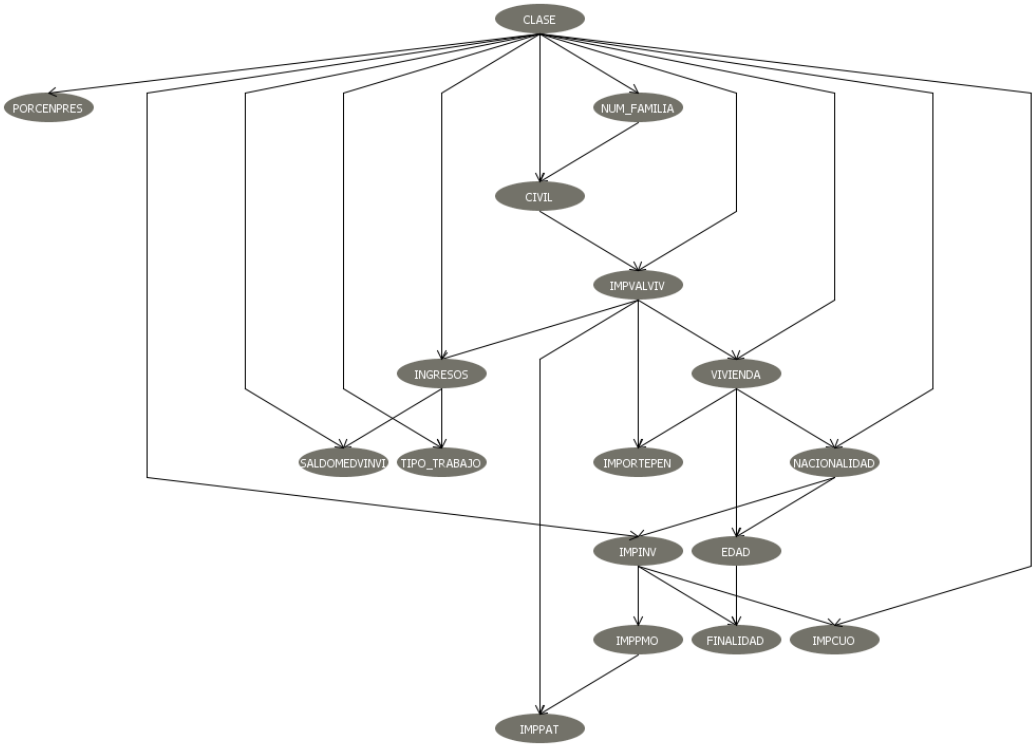
Cada algoritmo de búsqueda de los que se han utilizado genera una estructura de red bayesiana sobre la que se pueden calcular, como en la tabla anterior, las probabilidades condicionales a sus respectivos padres.

Para ver cómo aprovechar la estructura de dependencias de la red bayesiana para reducir el número de variables utilizando el Manto de Markov se va a emplear la red bayesiana permitiendo que cada variable pueda tener dos padres. En la Figura 5.13 se observa la estructura de la red que se ha obtenido utilizando el algoritmo Hill Cilmber, que parte de una red de enlaces vacía y emplea una métrica BIC (Bayesian information Criterion).

En esta red se observan las relaciones de dependencias directas e indirectas entre las variables. Entre otras muchas dependencias que se obtienen, podemos destacar la vinculación entre el importe de la cuota del crédito (IMPCUO) que se relaciona directamente con el importe de la inversión (IMPINV), el importe del préstamo (IMPPMO) que también depende del importe de la inversión o los importes pendientes

(IMPORTEPEN), relacionados estrechamente con el valor de la vivienda y con el tipo de la misma.

Figura 5.13. Red Bayesiana. Hill Climber con 2 padres.

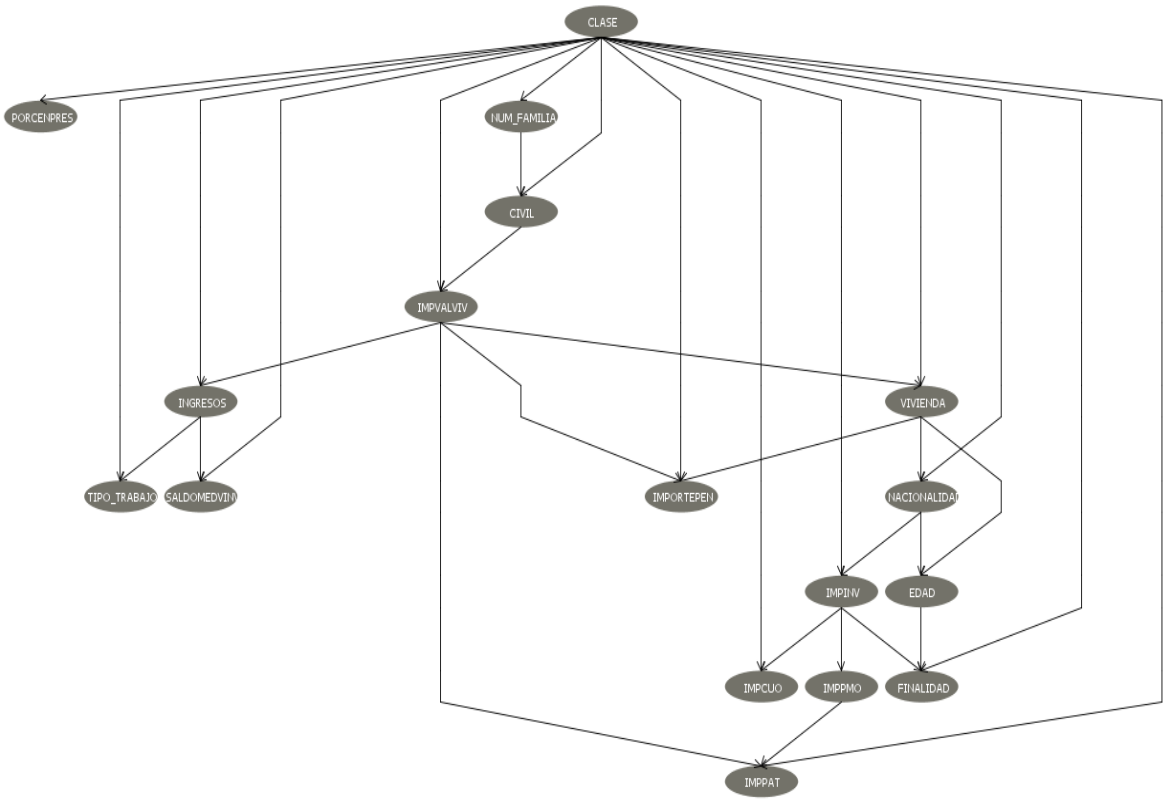


Dado que en una red bayesiana se cumple que toda variable explicativa x es independiente del resto de variables explicativas dado su manto (envolvente) de Markov, se puede seleccionar exclusivamente aquellas variables predictoras que realmente sean útiles en el modelo buscado. El Manto de Markov está formado por la unión de (padres (X) U hijos (X) U padres (hijos (X))).

Las variables que no se encuentran dentro de la envolvente de Markov son las siguientes: Importes pendientes, Edad, Finalidad, Importe del préstamo e Importe del patrimonio.

Una vez eliminadas estas variables de la base de datos se vuelve a estimar la red bayesiana con las mismas especificaciones que el modelo anterior obteniéndose otra estructura de dependencias e independencias que se refleja en la figura 5.14.

Figura 5.14. Red Bayesiana. Hill Climber con 2 padres y Manto de Markov.



Las diferentes tablas de probabilidad condicionadas a sus respectivos padres obtenidas de la estructura de la red bayesiana generada a través del algoritmo de ascensión de colinas, con dos padres, se muestran en el anexo número cuatro.

5.2.6. Multiclasificadores.

Como se ha señalado en la parte teórica del capítulo tres, actualmente existen muchas propuestas para combinar modelos. En este epígrafe se ofrecen los resultados de los procedimientos más utilizados por los diversos investigadores y que están resultando óptimos a la hora de obtener buenas predicciones. Los métodos multiclasificadores que se emplean en esta tesis son: Bagging, Boosting, Decorate, Random Forest, Random SubSpace, Vote y Stacking.

En primer lugar, se ha intentado encontrar cuáles son los mejores parámetros en relación al número de subconjuntos de entrenamiento y al metaclasificador que combine las predicciones obtenidas para proceder a realizar la clasificación a través de los métodos Bagging, Boosting, Decorate y Random SubSpace. Se ha efectuado la simulación con el método Bagging para cierto rango de valores y para los siguientes meta clasificadores: árboles de decisión, Rep Tree (el que dispone WEKA por defecto) y el C 4.5 con varias valores del factor de confianza *c*, una red bayesiana y un máquina de vectores soporte. Los resultados de esta simulación se observan en la tabla 5.67.

Tabla 5.67. Método Bagging. Fase de entrenamiento y de test.

		Fase de entrenamiento				Fase de test			
Modelo	Número de iteraciones	SI	NO	Total	ROC	SI	NO	Total	ROC
REPTree	10	84,0	84,5	84,2	0,918	71,4	100,0	84,6	0,881
	50	83,3	86,5	84,9	0,926	71,4	91,7	80,8	0,851
	100	83,0	87,4	85,2	0,925	71,4	83,3	76,9	0,857
Bayes (K, 1 padre)	10	83,7	83,2	83,4	0,908	85,0	91,7	88,5	0,935
	50	83,7	83,5	83,6	0,909	78,6	83,3	80,8	0,935
	100	83,3	83,3	83,3	0,910	78,6	91,7	84,6	0,940
C 4.5 (c = 0,25)	10	83,3	85,2	84,2	0,917	78,6	75,0	76,9	0,875
	50	85,3	83,2	84,2	0,922	71,4	75,0	73,3	0,821
	100	85,3	83,2	84,2	0,922	71,4	83,3	76,9	0,827
C 4.5 (c = 2)	10	84,9	82,9	83,8	0,918	78,6	75,0	76,9	0,875
	50	83,8	86,2	85,0	0,927	92,9	100,0	76,9	0,964
	100	85,6	85,9	85,8	0,932	92,9	100,0	96,2	0,976
Vectores soporte (c = 1)	10	83,7	82,3	83,0	0,893	71,4	83,3	76,9	0,836
	50	81,4	80,8	81,1	0,865	64,3	83,3	73,1	0,878
	100	80,8	81,7	81,3	0,881	64,3	83,3	73,1	0,877

Los modelos que se han empleado para configurar el multiclaificador Stacking y Vote son: árbol de decisión C 4.5, máquinas de vectores soporte, regresión logística, redes neuronales y redes bayesianas.

El multiclasificador Vote utiliza diferentes formas para unir las predicciones de los clasificadores que lo integran. Las diversas formas de combinar los resultados de los modelos se observan en las tablas 5.68 y 5.69.

Tabla 5.68. Comparación de modelos. Multiclasificador Vote. Fase de Entrenamiento.

	Fase de entrenamiento						
	Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
	SÍ	NO	Total	SÍ	NO	Total	Valor
Media de probabilidades	80,5	84,4	82,5	83,8	81,3	82,5	0,914
Producto de probabilidades	81,9	83,3	82,6	83,0	82,2	82,6	0,808
Votación por mayoria	81,1	85,3	83,2	84,7	81,9	83,3	0,832
Mínima probabilidad	81,1	83,3	82,6	83,0	82,2	82,6	0,808
Máxima probabilidad	82,9	79,0	81,0	79,8	82,2	81,0	0,906

Tabla 5.69. Comparación de modelos. Multiclasificador Vote. Fase deTest.

	Fase de test						
	Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
	SÍ	NO	Total	SÍ	NO	Total	Valor
Media de probabilidades	85,7	100,0	92,3	100,0	85,7	93,4	0,973
Producto de probabilidades	81,8	83,3	82,6	81,8	83,3	82,6	0,794
Votación por mayoria	85,7	100,0	92,3	100,0	85,7	93,4	0,973
Mínima probabilidad	81,8	83,3	82,6	81,8	83,3	82,6	0,794
Máxima probabilidad	85,7	83,3	84,6	85,7	83,3	84,6	0,934

Una vez elegidas las mejores opciones de los diferentes clasificadores se procede a calcular los porcentajes de clasificación y diferentes medidas para su análisis, tanto para la muestra de entrenamiento como para la de test, cuyos resultados se reflejan en las tablas 5.70 y 5.71.

Tabla 5.70. Comparación de modelos Multiclasificadores. Fase de entrenamiento.

	Fase de entrenamiento						
	Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
	SÍ	NO	Total	SÍ	NO	Total	Valor
Bagging	82,5	84,8	83,7	84,0	80,9	82,4	0,924
Adaboots	86,1	87,1	86,6	83,5	83,3	83,4	0,919
Decorate	81,6	86,3	84,0	86,4	84,1	85,2	0,915
Random Forest	82,4	85,6	84,0	85,2	84,8	85,0	0,924
Random SubSpace	82,6	85,6	84,1	84,2	80,9	82,6	0,925
Vote	80,3	84,0	82,2	83,8	81,3	82,5	0,822
Stacking	77,4	84,4	80,9	82,5	78,8	80,2	0,833

Tabla 5.71. Comparación de modelos Multiclasificadores. Fase de Test.

	Fase de test						
	Correctamente clasificados. (True Positive Rate)			Precisión			Área Curva ROC
	SÍ	NO	Total	SÍ	NO	Total	Valor
Bagging	85,7	100,0	92,3	100,0	85,7	93,4	0,970
Adaboots	85,7	100,0	92,3	100,0	85,7	93,4	0,970
Decorate	92,9	100,0	96,2	100,0	92,3	96,4	0,982
Random Forest	92,9	100,0	96,2	100,0	92,3	96,4	1,000
Random SubSpace	78,6	100,0	88,5	100,0	80,0	90,8	0,776
Vote	85,7	100,0	92,3	100,0	85,7	93,4	0,973
Stacking	85,7	91,7	88,5	92,3	84,6	88,8	0,905

Los resultados que se obtienen son excelentes: altas tasas de acierto, mejor en la fase de test, con seis de los clasificadores que aciertan todos los registros de la clase que no devuelven el crédito. También presentan elevadas estimaciones del área de la curva ROC. Los resultados que ofrecen los métodos de agregación de modelos Vote y Stacking presentan un desempeño ligeramente peor.

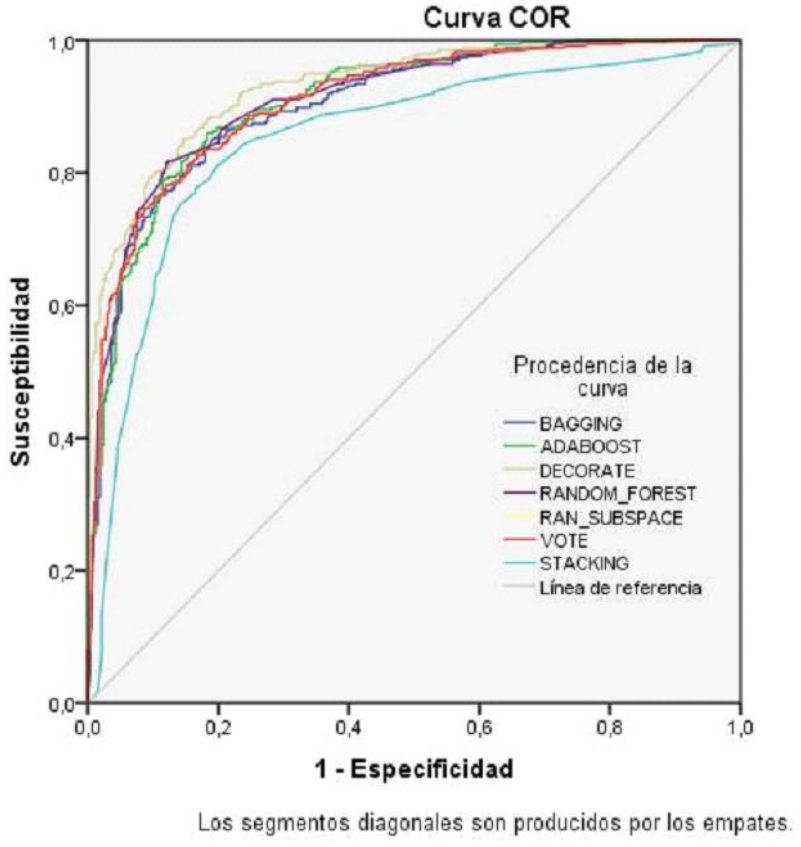
Finalmente, para proceder a comparar las diferentes opciones disponibles de modelos y poder elegir al mejor multiclaclasificador que nos ayude en la toma de decisiones respecto a la petición de créditos personales, se procede a comparar los siete modelos elegidos. Los resultados, que se resumen en la tabla 5.72, señalan que, en relación con el modelo base de contrastación que es el Bagging, el método Boosting (Adaboost en WEKA) presenta estadísticamente mejores resultados en el porcentaje de acierto de los que devuelven el crédito (Clase SÍ). Los multiclasificadores Vote y Stacking obtienen una peor estimación del área de la curva ROC. Esta curva se visualiza en la figura 5.15 para los siete multiclasificadores considerados.

Tabla 5.72. Contraste de modelos Multiclasificadores.

Modelo base de contrastación: Bagging	Correctamente clasificados				Área Curva ROC	
	Sí	Desviación estándar	NO	Desviación estándar	Valor	Desviación estándar
Bagging	82,5	6,3	84,8	6,2	0,924	0,030
Adaboots	86,1 (v)	5,8	87,1	5,7	0,919	0,033
Decorate	81,6	6,8	86,3	5,9	0,915	0,033
Random Forest	82,4	6,8	85,6	6,1	0,924	0,030
Random SubSpace	82,6	6,0	85,6	5,8	0,925	0,031
Vote	80,3	6,6	84,0	6,1	0,822 (*)	0,040
Stacking	77,4 (*)	7,6	84,4	7,2	0,833 (*)	0,052

Nota: (v) Estadísticamente mejor que el modelo base. (*) Estadísticamente peor.

Figura 5.15. Área bajo la curva COR de diversos Multiclasificadores.



5.3. Conclusiones de los análisis de los modelos.

Una vez que se han efectuado los diferentes ajustes de los modelos de clasificación y se han escogido los que mejor predecían la devolución o no del crédito se va a realizar una comparación entre ellos para averiguar cuál es el más conveniente.

Hay que tener en cuenta las especificaciones de los acuerdos de Basilea, en los que se decantan por los modelos probabilísticos y que en esta tesis son sólo dos los modelos analizados: la regresión logística y las redes bayesianas.

Además de los dos modelos probabilísticos se han escogido dieciséis modelos para realizar la contrastación entre todos ellos. El modelo que considero como base de contrastación es la regresión logística, ya que este modelo ha sido el más utilizado atendiendo a la investigación sobre el estado del arte sobre los modelos de credit scoring que se ha llevado a cabo en esta tesis doctoral y cuyos resultados se encuentran en el capítulo dos.

En la tabla número 5.73 se presentan los resultados para los diferentes modelos con 16 variables explicativas y, la siguiente tabla, la 5.74 contiene los resultados de los modelos con 11 variables debido a la reducción de cinco variables al aplicar la envolvente de Markov en la red bayesiana estimada a través del algoritmo de ascensión de colinas con dos padres, tal y como se ha visto en el epígrafe 5.2.5.

Al realizar el contraste entre la regresión logística y las diferentes estructuras de redes bayesianas encontramos que prácticamente todos los modelos de redes bayesianas son estadísticamente mejor que los modelos de regresión logística.

Los multiclasificadores también obtienen un mejor desempeño que la regresión logística.

Sólo las Máquinas de Vectores Soporte y el árbol C 4.5 obtienen peores resultados que la regresión logística en cuanto a que resultan peores modelos atendiendo al test de hipótesis sobre el valor del área de la curva ROC.

Cuando el modelo base de contrastación es la red bayesiana con el algoritmo Hill Climber con dos padres, el único modelo estadísticamente significativo superior es otra estructura bayesiana (Average One Dependence Estimators). Este modelo es el mejor en el ranking de contrastación que establece WEKA, seguidos a la par por dos modelos de redes bayesianas, TAN (Tree Aumented Netwok) y la red bayesiana con el

algoritmo de ascensión de colinas, y de tres multiclasificadores: Adaboost, Decorate y Random SubSpace.

Tabla 5.73. Resumen. Contraste de modelos con 16 variables.

	Correctamente clasificados				Área Curva ROC	
Modelo base de contrastación: Regresión Logística	**Sí**	**Desviación estándar**	**NO**	**Desviación estándar**	**Valor**	**Desviación estándar**
Árbol de decisión (C 4.5)	76,3	7,0	84,0	6,4	0,814 (*)	5,2
Red Neuronal. Perceptróm Multicapa	77,4	7,0	80,4	7,4	0,851	4,6
Perceptrón Multicapa con función de penalización	78,9	6,6	82,8	6,6	0,882	3,5
Red Neuronal. Funcion de Base Radial	81,0	6,3	81,6	6,9	0,888	3,6
Máquina de Vectores Soporte	80,2	6,5	83,5	6,3	0,819 (*)	4,2
Regresión Logística	80,6	6,2	82,5	6,1	0,890	3,4
Average One Dependence Estimator (AODE) con dos padres	81,8	6,5	87,3 (v)	5,7	0,925 (v)	3,0
Naïve Bayes con discretización	82,0	6,2	83,3	6,2	0,907	3,3
Red Bayesiana (TAN)	83,0	5,8	84,0	6,4	0,918 (v)	3,2
Red Bayesiana (K2)	82,4	6,6	83,3	6,5	0,918 (v)	3,0
Red Bayesiana (Hill Climber con 2 padres)	83,0	6,2	84,3	5,8	0,922 (v)	3,2
Red Bayesiana (Tabu Search)	82,7	6,2	83,3	6,2	0,916 (v)	3,0
Red Bayesiana (Simulated Anneling)	82,0	6,9	84,6	6,1	0,915 (v)	3,2
Bagging	82,5	6,3	84,8	6,2	0,924 (v)	3,0
AdaBoost	86,1 (v)	5,8	87,1	5,7	0,919 (v)	3,3
Decorate	81,6	6,8	86,3	5,9	0,915 (v)	3,3
Random Forest	82,4	6,8	85,6	6,1	0,924 (v)	3,0
Ramdom Subspace	82,6	6,0	85,6	5,8	0,925 (v)	3,1

Nota: (v) Estadísticamente mejor que el modelo base. (*) Estadísticamente peor.

Cuando la contrastación se lleva a cabo sobre el modelo de once variables las conclusiones son muy parecidas al de dieciséis variables. En la estimación correcta de la clase SÍ el modelo de regresión logística es superado estadísticamente por tres modelos bayesianos y por los cinco multiclasificadores. Respecto a la clase NO, solamente resulta mejor el modelo AOED y, en cuanto al área de la curva ROC, es estadísticamente peor la regresión logística como clasificador que cualquier modelo de red bayesiana o que cualquier modelo multiclasificador.

Si se realiza la contrastación con el modelo base de la red bayesiana, ningún modelo considerado es estadísticamente superior a él. En el ranking de la contrastación entre los dieciocho modelos estudiados, el primer puesto del ranking lo ocupan las redes bayesianas estimadas a través de los algoritmos de búsqueda: Simulated Annealing, Tabu Search, K2, TAN y el modelo Averaged One Dependence Estimators, el segundo lugar lo comparten, AdaBoost, Random Forest, Bagging y la red bayesiana con ascensión de colinas y, el tercer lugar, lo ocupa el multiclasificador RandomSubSpace.

Tabla 5.74. Resumen. Contraste de modelos con 11 variables.

	Correctamente clasificados				Área Curva ROC	
Modelo base de contrastación: Regresión Logística	**Sí**	**Desviación estándar**	**NO**	**Desviación estándar**	**Valor**	**Desviación estándar**
Árbol de decisión (C 4.5)	78,8	7,6	79,4	7,5	0,825 (*)	4,6
Red Neuronal. Perceptróm Multicapa	76,6	6,9	82,8	5,9	0,869	3,8
Perceptrón Multicapa con función de penalización	79,3	6,5	82,5	6,2	0,888	3,7
Red Neuronal. Funcion de Base Radial	78,2	6,6	81,0	5,9	0,875	3,9
Máquina de Vectores Soporte	78,6	6,6	79,4	6,6	0,790 (*)	4,1
Regresión Logística	78,8	6,6	81,6	6,6	0,880	3,7
Average One Dependence Estimator (AODE) con dos padres	81,4	6,2	86,4 (v)	5,3	0,921 (v)	3,0
Naïve Bayes con discretización	81,7	6,1	82,8	6,2	0,909 (v)	3,2
Red Bayesiana (TAN)	82,9 (v)	6,3	85,6	5,6	0,927 (v)	3,0
Red Bayesiana (K2)	81,9	6,5	85,0	6,1	0,925 (v)	3,1
Red Bayesiana (Hill Climber con 2 padres)	83,2 (v)	6,2	84,8	5,5	0,921 (v)	3,2
Red Bayesiana (Tabu Search)	83,3 (v)	6,1	84,6	5,7	0,923 (v)	3,1
Red Bayesiana (Simulated Anneling)	82,5 (v)	6,4	85,5	5,8	0,926 (v)	3,2
Bagging	83,6 (v)	6,2	84,2	6,1	0,921 (v)	3,2
AdaBoost	84,8 (v)	6,0	84,6	6,7	0,920 (v)	3,4
Decorate	80,1 (v)	6,1	84,3	5,8	0,902 (v)	3,6
Random Forest	82,4 (v)	6,9	83,5	6,5	0,916 (v)	3,2
Ramdom Subspace	82,8 (v)	6,0	84,2	6,5	0,918 (v)	3,4

Nota: (v) Estadísticamente mejor que el modelo base. (*) Estadísticamente peor.

CAPÍTULO 6

IMPLEMENTACIÓN DE LA APLICACIÓN DE CREDIT SCORING.

6. Implementación de la aplicación de Credit Scoring.

En este capítulo se explica la aplicación informática que se ha desarrollado para ayudar a los gerentes de los bancos a automatizar la decisión de conceder o no un crédito utilizando los mejores algoritmos de clasificación que se han encontrado en el desarrollo de esta tesis con los datos proporcionados por Caja Rioja.

Esta aplicación incluida en esta tesis se dona al colectivo científico para que pueda disponer de una herramienta sencilla para efectuar análisis así como simulaciones en sus trabajos científicos.

Esta implementación es similar a la que tienen los bancos pero con un mayor número de modelos disponibles. Sirve, por una parte, para realizar simulaciones con un conjunto de datos dado y ver el comportamiento de los diferentes algoritmos de clasificación desarrollados en esta tesis y, por otra, con la misma aplicación se puede utilizar, con los datos de un único peticionario de crédito, para ver cómo actúan los diferentes modelos clasificatorios seleccionados y con qué probabilidad se clasifica.

El lenguaje utilizado para la implementación de la aplicación informática se ha desarrollado en lenguaje JAVA. Varias son las razones por la que ha sido elegido este lenguaje de programación: es un lenguaje orientado a objetos e independiente de la plataforma. Es además un lenguaje robusto, que gestiona la memoria automáticamente, dispone de mecanismos de seguridad incorporados, además de tener una estupenda documentación.

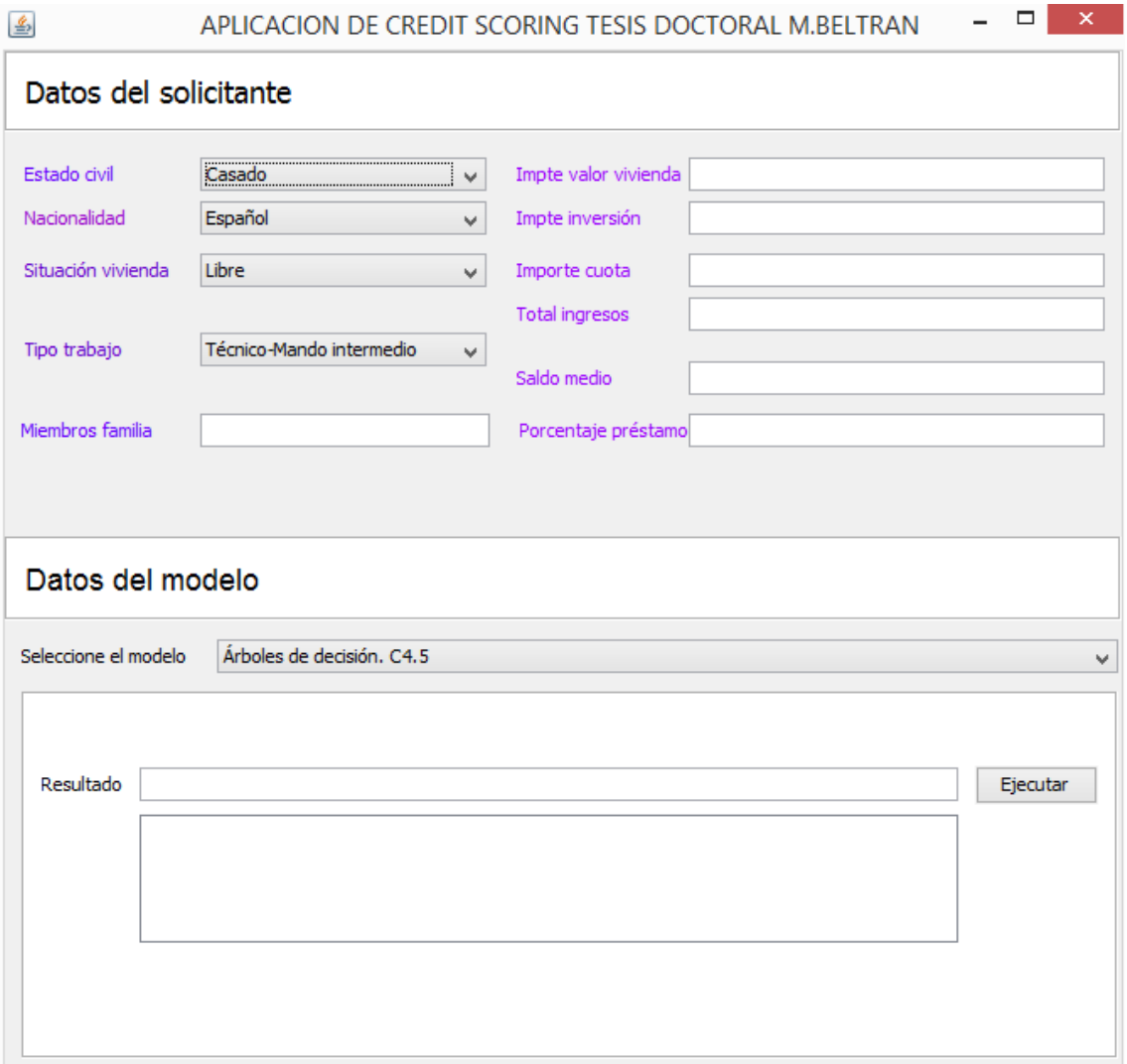
Los algoritmos de clasificación agregados en el programa WEKA (Waikato Environment for Knowledge Analysis) también están escritos en JAVA con lo que las ventajas son múltiples al poder unir los desarrollos de JAVA con las clases disponibles en weka a través del fichero weka.jar, que nos sirve de conexión, para facilitar la salida de resultados de los algoritmos de clasificación.

Los comandos de programación JAVA de esta aplicación se encuentran en el anexo nº 2.

El formulario de la aplicación de credit scoring, que se corresponde con el de la figura nº 6.1, nos permite introducir los datos de la persona que nos solicita el crédito. En la misma pantalla también se puede especificar, a través del desplegable, el modelo con el queremos realizar la clasificación. Los modelos con los que se puede experimentar son los ocho siguientes:

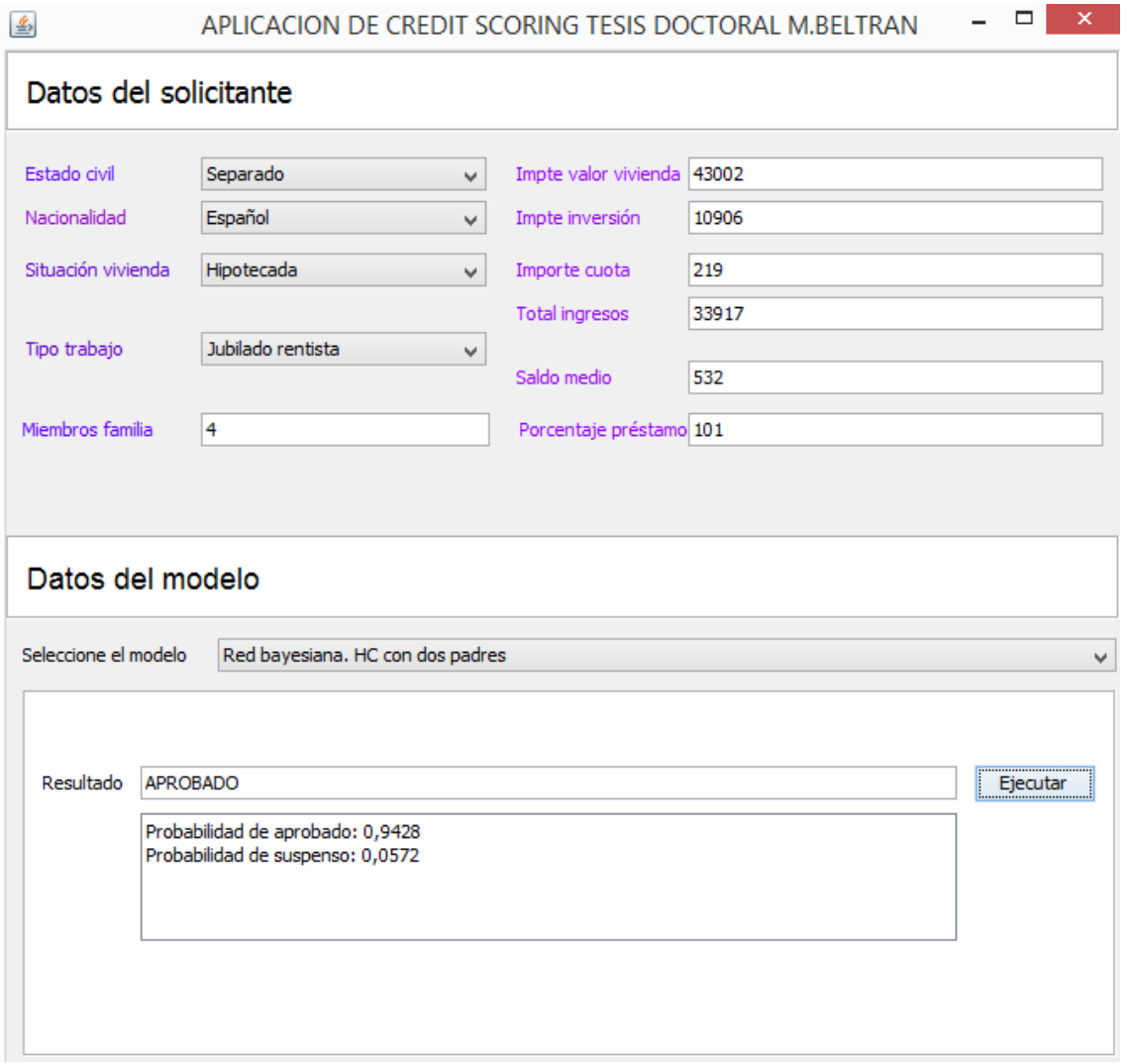
- ✓ Arbol de decisión. C 4.5.
- ✓ Naïve Bayes.
- ✓ Red Bayesiana. HC con dos padres.
- ✓ Red Neuronal.
- ✓ Regresión logística.
- ✓ Máquinas de Vectores Soporte.
- ✓ Multiclasificador Boosting.
- ✓ Multiclasicador Random Subspace.

Figura 6.1. Formulario de entrada de la aplicación de Credit Scoring.



Una vez se ha especificado el modelo podemos ejecutarlo para obtener el resultados de la calsiicación: APROBADO o DENEGADO, acompañando esta calificación con una probabilidad.asociada a la clasificación.

Figura 6.2. Datos de ejemplo de la aplicación de Credit Scoring.



Para realizar una simulación vamos a utilizar algunos de los registros del fichero de test contenidos en la tabla 6.1. Estos valores los introduciremos en la aplicación. Se cuenta con ventanas desplegables que posibilitan la introducción automática de los valores de las variables alfanuméricas. Después de elegir un modelo y de ejecutar la aplicación la salida que obtenemos es el resultado de la clasificación con la probabilidad asociada tal y como se observa en la figura 6.2.

Tabla 6.1. Simulación de la aplicación informática con algunos valores de test.

REGISTRO	NUM_FAMILIA	VIVIENDA	IMPVALVIV	NACIONALIDAD	IMPINV	IMPCUO	INGRESOS	SALDO	PORCENPRES	CIVIL	TIPO_TRABAJO	CLASE
1	1	Familia	0	Español	9.000	250	22.390	702	100	Soltero	Técnico	SÍ
2	4	Libre	60.011	Extranjero	1.061	29	15.297	956	100	Casado	Obrero temporal	SÍ
3	3	Familia	0	Extranjero	1.512	89	14.312	38	100	Casado	Obrero temporal	NO
4	1	Familia	0	Español	7.000	219	16.788	58	100	Soltero	Obrero temporal	NO
5	1	Hipotecada	90.040	Español	15.265	265	13.855	3.076	100	Soltero	Obrero fijo especializado	SÍ
6	4	Familia	0	Español	6.000	128	17.600	2.103	100	Soltero	Obrero fijo especializado	NO
7	1	Alquiler	0	Extranjero	4.000	181	14.979	0	103	Soltero	Obrero fijo especializado	NO
8	3	Hipotecada	12.020	Extranjero	5.000	103	12.623	0	100	Casado	Autónomo	SÍ
9	4	Hipotecada	43.002	Español	10.906	219	33.918	532	101	Casado	Jubilado - rentista	SÍ
10	1	Alquiler	0	Extranjero	21.000	411	29.271	9.939	114	Soltero	Jubilado - rentista	NO

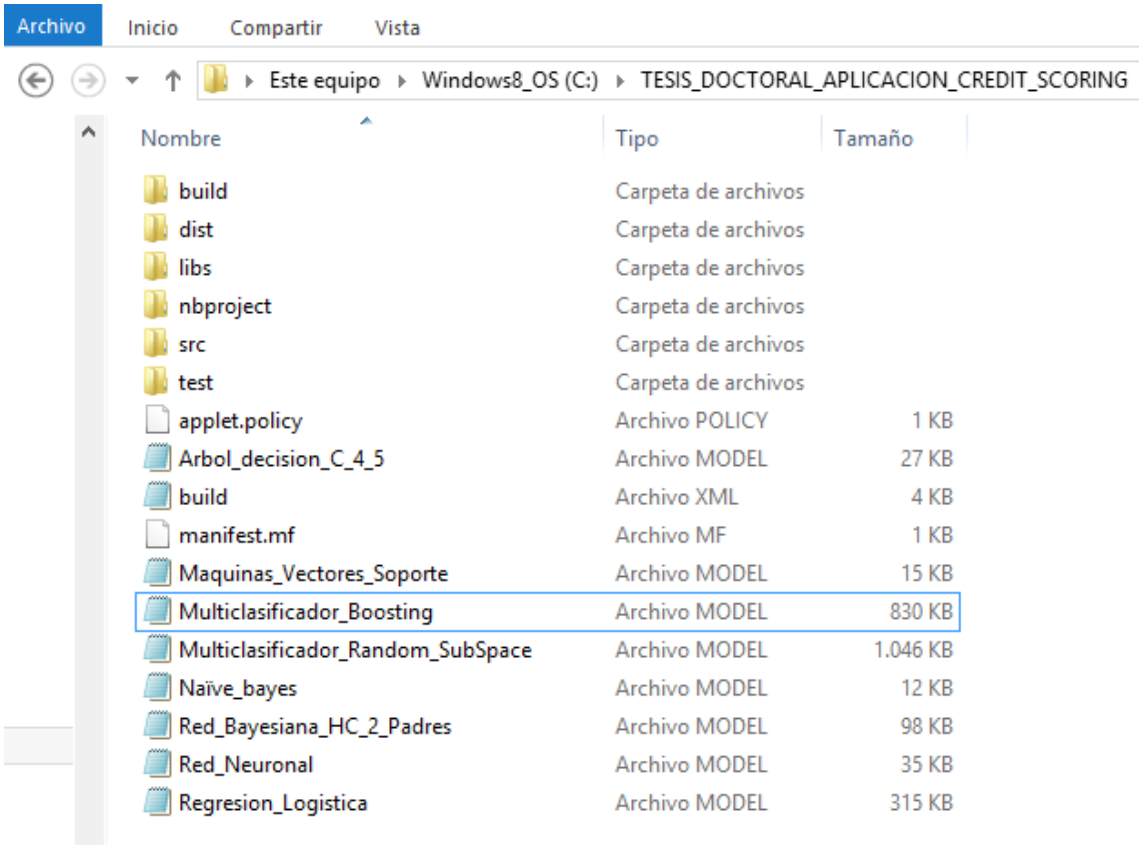
Tabla 6.2. Resultados de la decisión con la aplicación de credit scorings acompañados de su probabilidad.

	1	**2**	**3**	**4**	**5**
Árbol de decisión	A (1,0000)	D (0,8571)	D (0,8462)	D (0,9688)	A (1,0000)
Naïve Bayes	A (0,9081)	A (0,9986)	D (0,9977)	D (0,9930)	A (0,9883)
Red Bayesiana	A (0,9742)	A (0,8590)	D (0,9795)	D (0,9221)	A (0,9805)
Red Neuronal	A (0,9655)	A (0,9732)	D (0,5189)	D (0,9752)	A (0,9995)
Regresión Logística	A (0,7828)	A (0,5796)	D (0,8872)	D (0,8278)	A (0,9384)
Máquinas de Vectores Soporte	A (1,0000)	D (1,0000)	D (1,0000)	D (1,0000)	A (1,0000)
Multiclasificador Boosting	A (1,0000)	A (1,0000)	D (1,0000)	D (1,0000)	A (1,0000)
Multiclasificador Random Subspace	A (0,8346)	A (0,5814)	D (0,8284)	D (0,8500)	A (0,8835)

	6	**7**	**8**	**9**	**10**
Árbol de decisión	D (1,0000)	D (0,7429)	D (1,0000)	A (1,0000)	D (0,7429)
Naïve Bayes	A (0,5215)	D (0,9849)	D (0,9724)	A (0,9994)	A (0,7589)
Red Bayesiana	A (0,8364)	A (0,6574)	D (0,9188)	A (0,9659)	A (0,6825)
Red Neuronal	A (0,9849)	D (0,9612)	D (0,8231)	A (0,5677)	D (0,6925)
Regresión Logística	A (0,7810)	A (0,9563)	D (0,5468)	A (0,5868)	D (0,5132)
Máquinas de Vectores Soporte	A (1,0000)	D (1,0000)	A (1,0000)	A (1,0000)	D (1,0000)
Multiclasificador Boosting	A (1,0000)	D (1,0000)	D (1,0000)	A (1,0000)	D (0,9438)
Multiclasificador Random Subspace	A (0,5402)	D (0,7696)	D (0,6438)	A (0,8546)	D (0,7272)

Nota: A es que se aprueba el crédito, D es que se deniega.

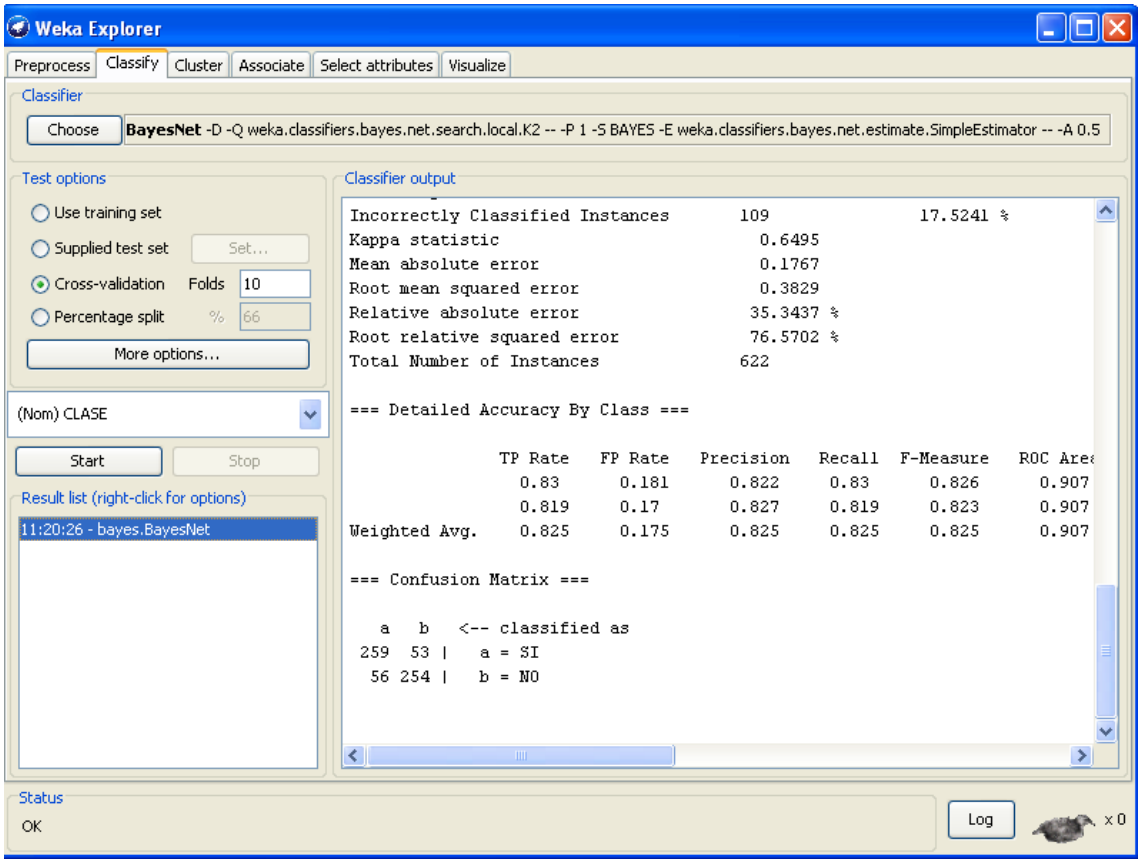
Figura 6.3. Ficheros de la aplicación de Credit Scoring.



El fichero con los datos debe de tener el formato del programa WEKA (arff). Este fichero, junto con los modelos de salida del mismo programa, tienen que ubicarse en el directorio raíz de la carpeta donde se guarde la aplicación JAVA.

Cuando se requiera un nuevo modelo de clasificación, por la entrada normalmente por la entrada de nuevos datos de los peticionarios de crédito o por otros motivos, el modelo deberá de estimarse a través del programa WEKA. En la figura 6.4 se observa cómo se utiliza el programa con un modelo de red bayesiana. El modelo resultante debe de ser guardado en la carpeta raíz donde se encuentre la aplicación informática. Para archivar el resultado se realiza a través del botón derecho del ratón posicionándose sobre el modelo (Result list- right – click for options) y se selecciona la opción de guardar modelo, que es una de las opciones disponibles. (Save model).

Figura 6.4. Resultados de la salida del programa Weka para una red bayesiana.



La aplicación puede ejecutarse de dos formas, una primera, una primera utilizando el fichero Tesis_Scoring.jar que se encuentra en la carpeta “dist” de la aplicación java, o bien a través del entorno de desarrollo de java, NetBeans IDE 7.01, que es la interface con la que se ha desarrollado la aplicación. Al emplear esta plataforma de desarrollo las posibilidades de programación se amplían considerablemente.

Figura 6.5 Interface de neatbean.

CAPÍTULO 7

CONCLUSIONES, APORTACIONES Y NUEVAS LÍNEAS DE INVESTIGACIÓN.

7. Conclusiones, aportaciones y nuevas líneas de investigación.

7.1. Conclusiones y aportaciones de la tesis doctoral.

Como resumen del estudio del arte sobre credit scoring, del análisis de los datos y de la aplicación de los modelos de minería de datos utilizados en esta tesis doctoral se extraen varias conclusiones:

Del amplio estudio bibliográfico llevado a cabo se deducen dos conclusiones claras: salvo en escasos estudios, los autores de las tesis y de los artículos analizados, no realizan una adecuada selección de variables y, otra cuestión importante, es que tampoco abundan los trabajos donde se lleve a cabo un balanceo de la base de datos adecuado cuando las clases están desequilibradas. Tanto la correcta selección de variables como el eficiente equilibrio de la muestra que se han abordado en esta investigación creo que son aportaciones interesantes realizadas en esta tesis doctoral.

También se ha comprobado, a partir del análisis de la bibliografía existente sobre credit scoring, que los métodos más utilizados para ayudar a los gestores bancarios en la toma de decisiones a la hora de conceder o no un crédito bancario han sido los modelos de regresión logística. Con los datos de préstamos bancarios personales de la Caja de Ahorros con los que se ha trabajado en esta tesis doctoral se demuestra, a través de la contrastación estadística que, en la gestión de créditos bancarios personales, los modelos logísticos, si bien ofrecen resultados aceptables, no son los modelos óptimos.

Por otra parte, entre los diferentes modelos de regresión logística lineal y algunas de sus expansiones a través de estimadores kernel que se han utilizado, cuya finalidad no es otra que superar la no linealidad del modelo, podemos afirmar que, a nivel estadístico una vez realizada la contrastación, ninguno de ellos presenta una mayor capacidad predictiva.

Una de las motivaciones de esta tesis consistió en desarrollar modelos de credit scoring óptimos y mejores a los conocidos y, en la medida de lo posible, que estén de acuerdo a las exigencias de calcular la probabilidad de default que requieren los modelos de Basilea II y III y que redundarán, sin lugar a dudas, en un mayor beneficio de las instituciones al aplicar estos modelos en el proceso de concesión de créditos.

Desde este punto de vista dos de los modelos probabilísticos ampliamente explicados en esta tesis, regresión logística y redes bayesianas, cumplen estos requisitos. Son

estos últimos las que presentan mejores resultados avalados por un mayor porcentaje de aciertos, una mayor precisión a través del área bajo la curva ROC y a través de los contrastes estadísticos efectuados, por lo que podemos afirmar que las redes bayesiana presentan un mejor desempeño que la regresión logística.

La utilización de las redes bayesianas con un óptimo equilibrado de las instancias, unido a la correcta selección del conjunto de variables explicativas para la resolución del problema del crédit scoring, nos ha conducido a obtener excelentes resultados en la fase de entrenamiento y una mayor precisión en la fase de test. Una de las razones fundamentales de que estos modelos resultan excelentes a la hora de aplicarlos al credit scoring es que el enfoque bayesiano, basado en modelos de probabilidad, emplea la teoría de la decisión para el análisis del riesgo eligiendo en cada situación que se presenta la acción que maximiza la utilidad esperada.

Como se ha demostrado en esta tesis doctoral existen muchas variantes de algoritmos de clasificación, sin embargo, los clasificadores bayesianos incorporan la información de los datos mediante modelos matemáticos o teóricos construidos a partir de la teoría de Bayes, la cual ofrece un modelo que minimiza la probabilidad total del error.

Otra importante ventaja a la hora de decidirnos por la construcción de redes bayesianas es que en estos modelos se puede incorporar el conocimiento experto de los gestores de la institución bancaria. En el modelo final propuesto, el analista puede introducir y modelar la información disponible, dado que el método que se propone es que los clasificadores bayesianos son los más eficientes de acuerdo, por una parte a los excelentes porcentajes de clasificación que alcanza tanto para la clase de los que devuelven el crédito como la de los morosos y, por otra, que también están en consonancia con el cumplimiento de las recomendaciones bancarias de Basilea II.

Otra cuestión vital es que cuando la base de datos está muy desequilibrada respecto a la variable clase, los algoritmos de minería de datos son muy ineficientes respecto a la clase más desfavorecida, lo que conlleva obtener malos resultados. Una de las aportaciones de esta tesis doctoral ha consistido en incorporar el método del Cubo para el equilibrado de la muestra. No se tiene constancia que este método haya sido utilizado en proyectos de minería de datos salvo en esta tesis. Cuando las bases de datos están desbalanceadas las mejores opciones se experimentan cuando se equilibran las muestras.

En este sentido, en esta tesis doctoral, se ha utilizado el método del submuestreo equilibrado del Cubo, propuesto por Deville y Tillé (2004). Entre los métodos existentes

en la literatura estadística para la selección de submuestras es el denominado del Cubo el único que nos permite seleccionar una muestra equilibrada sobre variables auxiliares con probabilidades de inclusión que pueden ser iguales o no. Como ya se ha comentado, el método del Cubo selecciona únicamente las muestras cuyos estimadores de Horvitz-Thompson son iguales a los totales de las variables auxiliares conocidas.

También podemos afirmar que el proceso metodológico tan detallado y exhaustivo que se ha seguido no es habitual encontrarlo en la literatura de credit scoring. Partiendo, como creo que debe de ser, del proceso metodológico seguido por CRISP (Cross Industry Standard Process for Data Mining) se ha contemplado y particularizado en esta tesis según las características específicas del sector bancario.

Otra aportación importante es la efectuada en la fase de selección de variables, cuya tarea es imprescindible para buscar modelos más sencillos e interpretables. Diferentes algoritmos de clasificación presentan sus propios métodos de selección de variables como, por ejemplo, los árboles de decisión, donde los diferentes métodos empleados señalan que, si bien existe cierta homogeneidad en la selección de variables, la relevancia de las distintas variables no es independiente del método utilizado. En la regresión logística también existen procedimientos para descartar variables irrelevantes, al igual que en las redes neuronales se pueden detectar y ordenar las variables según su poder clasificatorio a través de la sensibilidad e interpretación de los pesos de la red, como ya se ha señalado en esta tesis doctoral.

De forma general, se han estudiado diferentes procedimientos de selección de variables que se pueden concretar en escoger las mejores características de la base de datos a través de los denominados métodos de filtro y métodos basados en modelos. Sin embargo, se afirma que, para esta base de datos de concesión de créditos personales de la Caja de Ahorros, es a través de la envolvente de Markov, aplicada a las redes bayesianas, donde podemos reducir significativamente el número de variables mejorando la interpretabilidad del modelo elegido. Este es el camino que nos parece óptimo y que se ha seguido. La utilización del manto de Markov ha disminuido significativamente el número de variables de dieciséis a once, lo que simplifica y ayuda a gestionar mejor la decisión de conceder o no un préstamo.

También se puede afirmar que, para resolver el problema del credit scoring, los métodos multiclasificadores: Bagging, Boosting, Decorate, Randon Forest y Random Sub Space obtienen unos resultados excelentes, tanto en términos de precisión a la hora de clasificar ambas clases, como a través de la curva ROC, tanto en la fase de entrenamiento como en la de test, al igual que el algoritmo de árboles de decisión con el método CHAID (Chi-squared Automatic Inteaction Detection) que también goza de un grado de desempeño muy elevado.

Una aportación notable de esta tesis es la aplicación informática que acompaña a esta tesis doctoral y que, aparte de ofrecérsela a los técnicos de la Caja de Ahorros, también está disponible para la comunidad científica. Como ya se ha señalado en el capítulo seis tiene una doble finalidad, por una parte es útil para realizar simulaciones con un conjunto de datos dado y ver el comportamiento de los diferentes algoritmos de clasificación desarrollados en esta tesis o por otros que pueda diseñar el usuario y, por otra, se puede utilizar con los datos de un único peticionario de crédito, para ver cómo actúan los diferentes modelos clasificatorios seleccionados y con qué probabilidad se clasifica.

7.2. Nuevas líneas de investigación.

Es obvio y evidente que hasta ahora no se ha conseguido el método clasificador perfecto que resulte óptimo para todas las bases de datos existentes.

Como líneas futuras de investigación acordes con el tema y el objetivo de esta tesis se pueden citar al menos tres fundamentales. Una de ellas relacionada con la especificación de la función de costes, otra respecto a la investigación de métodos de estimación de redes bayesianas y, una tercera, con la óptima combinación de modelos.

Cuando el coste económico de la clasificación es diferente según las clases, como en el credit scoring, incorporar la matriz de costes en los modelos es muy conveniente. Algunos métodos como el Metacost obtienen unos resultados muy aceptables ponderando la matriz de costes ya que optimizan el análisis coste beneficio. Sin embargo, hay que encontrar una función de coste que esté consensuada por los expertos bancarios y basada en una teoría económica que lo avale. En la investigación

llevada a cabo en esta tesis doctoral y después de haber hablado con técnicos especialistas de diferentes actores del sector bancario, no se han encontrado más que datos dispersos y sin justificación suficiente sobre el verdadero coste que conlleva las dos siguientes situaciones: aprobar un crédito cuando debería haberse rechazado y el coste de denegar un crédito cuando tendría que haberse concedido. En el periodo de crisis económica al que se circunscriben los datos analizados aún es más complejo calcular los errores de tipo I y II en la concesión de créditos.

La redes bayesianas son unos excelentes modelos para aplicarlas en el credit scoring como esta tesis ha demostrado, al menos para los clientes de préstamos personales. A la hora de utilizar las redes bayesianas con variables continuas, si estas no siguen una distribución conocida como por ejemplo la distribución normal, las variables deben de ser discretizadas lo que puede originar una pérdida de información. Esperamos que en un futuro próximo los investigadores científicos avancemos en dotar de más flexibilidad a las redes bayesiana en la estimación de la estructura y de la probabilidades condicionales que dotan de más expresividad y mayor poder de explicación a las redes bayesianas.

La teoría del aprendizaje bayesiano para combinar modelos de clasificación representa una manera óptima de mejorar la aportación individual de cada uno de ellos, si se dispone de información correcta de las probabilidades a priori P(h) y de los datos dado el modelo P(D|h), pero estos valores no son fácilmente computables en problemas reales, por lo que todavía, este punto de vista de combinar modelos, no ha sido muy empleado. Aunque se ha trabajado con aproximaciones de los valores de las probabilidades, dado que es imposible conocer los valores exactos, los resultados conseguidos hasta ahora no han sido muy satisfactorios. Esta línea de investigación relacionada con la combinación de modelos sigue abierta en la minería de datos y se espera que se encuentren soluciones útiles en futuros estudios.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

Abdou, H., Pointon, J. y El Masry, A. (2008). Neural nets versus conventional techniques in credit scoring in Egyptian banking. *Expert Systems with Applications 35* (3): 1275-1292.

Ackley, D.H., Hinton, G.E. y Sejnowski, T.J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science, 9,* 147-169.

Albus, J.S. (1975). New approach to manipulator control: The Cerebellar Model Articulation Controller (CMAC). Transactions of the ASME *Journal of Dynamic Systems, Measurement, and Control*, 220-227.

Altman, E.I. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. *The Journal of Finance, 23*. 589-609.

Altman, E.I. y Saunders (1998). Credit Risk Measurement: Developments over the last 20 years. *Journal of Banking & Finance, Vol. 21*. pp. 1721–1742.

Alman, E., Hartzell, J. y Peck, M. (1995). Emerging Markets Corporate Bonds: A Scoring System. Salomon Brothers Inc, New York.

Alpaydin, E. (2004). *Introduction to Machine Learning*. The MIT Press.

Anderson, J.A., Silverstein, J.W., Ritz, S.A. y Jones, R.S. (1977). Distinctive features, categorical perception and probability learning: some applications of a neural model. *Psychological Review, 84,* 413-451.

Apilado, V., Warner, D. y Dauten, J. (1974). Evaluative Techniques in Consumer finance-experimental result an policy implications for financial institutions. *Journal of Financial and Quantitative Analysis*, 275-283.

Asunción, A. y Newman, D.J. (2007). UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. URL http://www.ics.uci.edu/_mlearn/MLRepository.html.

Avery, R., Calem, P. y Canner, G. (2004). Consumer credit scoring: do situational circumstances matter? *Journal of Banking and Finance 28*(4), pp. 835-856. Battiti, R. (1992). First and second order methods for learning: between steepest descent and Newton's method. *Neural Computation, 4*(2), 141-166.

Azzaline, A. y Härdle, W. (1989). On the use of nonparametric regression for model checking. Biometrika, 76:1-12.

Baesens, B. (2003). Developing intelligent systems for credit scoring using machine learning techniques. *status: published*.

Baldi, P. y Soren, B. (2001). Bioinformatics: The machine learning approach. *2nd Ed., Massachusetts Institute of Technology: MIT Press*: 365-369.

Bajgier, S.M. y Hill, A.V. (1982). An Experimental comparison of statistica and linear programming approaches to the discrinminate problema. Decision Sciences. Nº 13, pp. 604-618.

Bamber, D.C. (1988). The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *J. Math. Psychol*., 12, 387-415.

Battiti, R. (1996). Reactive search: Toward self-tuning heuristics. V. J. Rayward-Smith, I. H. Osman, C. R. Reeves y G.D. Smith, editors, *Modern Heuristic Search Methods,* pages 61-83. John While & Son.

Beaver, W.H. (1966). Financial ratios as predictors of failure, Empirical Research in Accounting: Selected Studies, supplement to vol.5. *Journal of Accounting Research.* 71-111.

Bellotti, T. y Crook, J. (2007). Credit scoring with macroeconomic variables using survival analysis. *Journal of the Operational Research Society, 60*(12), pp. 1699-1707.

Belloti, T. y Crook. J. (2009). Support vector machine for credit scoring and discovery of significant features. *Expert Systems with Applictions*. 36(2 , Part 2), 3302-3308.

Beltrán, M., Muñoz, A., y Muñoz, A. (2012). Un nuevo clasificador de préstamos bancarios a través de la minería de datos. *Presentado en el 50 aniversario del SEIO.* http://info.uned.es/dpto-economia-aplicada-y-estadistica/SEIO2012.pdf.

Beltrán, M., Muñoz, A. y Muñoz, A. (2013). Redes bayesianas aplicadas a problemas de credit scoring. Una aplicación práctica. *Cuadernos de Economía.* http://dx.doi.org/10.1016/j.cesjef.2013.07.0.

Berzal, F., Cubero, J.C., Cuenca, F. y Martín-Bautista, M.J. (2003). On the quest for easy-to-understand splitting rules. *Data & Knowledge Engineering, 44*(1), 31-48.

Bessis, J. (2002): *Risk Management in Banking.* Second edition. Chichester: John Wiley and sons, 496 pp.

Bierman, H. y Hausman, W. (1970). The Credit Granting Decision. *Management Science 16,* B-519-532.

Biganzoli, E., Boracchi, P., Mariani, L. y Marubini, E. (1998). Feed-forward neural networks for the analysis of censored survival data: a partial logistic regression approach. *Statistics in Medicine, 17*(10), 1169-1186.

Biggs, D., De Ville, B. y Suen, E. (1991). A Method of Choosing Multiway Partitions for Classification and Decision Tress. *Journal of Applied Statistics,* 18 (1), 49-64.

Bilge, U., Refenes, A.N., Diamond, C. y Shadbolt, J. (1993). Application of sensitivity analysis techniques to neural network bond forecasting. En A.N. Refenes (Ed.), *Proceedings of 1st International Workshop on Neural Networks in the Capital Markets* (p. 12). London: London Business School.

Bishop, C.M. (1994). Neural networks and their applications. *Review of Scientific Instruments, 65*(6), 1803-1832.

Bishop, C.M. (1995). *Neural networks for pattern recognition.* Oxford: Oxford University Press.

Blake, C., Keogh, E. y Merz, C.J. (1998). *UCI Repository of Machine Learning Databases.*http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine CA: University of California, Department of Information and Computer Science.

Bouckaert, R.R. (1995). *Bayesian belief networks: from construction to inference*. Universiteit Utrecht, Faculteit Wiskunde en Informatica.

Boj, E., Claramunt, M.M., Esteve, A. y Fortiana, J. (2009). Criterio de selección de modelo en credit scoring Aplicación del análisis discriminante basado en distancias. *Anales del Instituto de Actuarios Españoles. 3*:209–30.

Bojadziev, G. y Bojadziev, M. (1997). Fuzzy logic for business, finance and management. Singapure: World Scientific *Publishing. Journal of Computational Information Systems 6:9* (2010) 2805-2811.

Bonfim, D. (2009). Credit risk drivers: Evaluating the contribution of firm level information and of macroeconomic dynamics. *Journal of Banking & Finance.*

Bonilla, M., Olmeda, I. y Puertas, R. (2003). Modelos paramétricos y no paramétricos en problemas de credit scoring. *Revista española de financiación y contabilidad*. Vol.XXXII, nº.118.

Boser, B.E., Guyon, I.M., y Vapnik, V.N. (1992). A training algorithm for optimal margin classifiers. *En Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144-152).

Bourouche, J.M. y Tennenhaus, M. (1972). Some segmentation methods. *Metra, 7*. pp 407-418.

Boyle, M., Crook, J.N., Hamilton, R. y Thomas, L.C. (1992). Methods for Credit Scoring Applied to Slow Payers. In Thomas, L. C., Crook, J. N., Edelman, D. B. (Eds.), *Credit Scoring and Credit Control* (pp. 75-90). Oxford, UK: Clarendon.

Breiman, L., Friedman, J.H., Olshen, R.A. y Stone, C.J. (1984). *Classification and regression trees.* Monterey*, CA: Wadsworth & Brooks/Cole Advanced Book & Software*.

Breiman, L. (1996). Bagging predictors. *Machine Learning*. Kluwer Academic Publishers, Boston, Manufactured in the Netherlands, vol. 24, 2, pp. 123-140.

Breiman, L. (1998). Arcing classifiers. *The Annals of Statistical* 26 (3). Pp 801-849.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Breiman, L. (2001). *Random Forests, random features*. Berkeley: University of California.

Broomhead, D.S. y Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems, 2*, 321-355.

Buntine, W. (1991). Theory refinement on Bayesian Networks. En *Proceedings of Seventh Conference on Uncertainty in Artificial intelligence*, Los Angeles, CA (pp. 52-60). Morgan Kaufmann.

Burges, C.J. (1998). A tutorial on support vector machines for pattern recognition*. Data mining and knowledge discovery,* 2(2), 121-167.

Campbell, S. y Dietrich, J. (1984). The Determinants of Default on insured Conventional Residential Mortgage Loans, The Journal of Finance 38(5), pp. 1569-1581.

Campos, L.M. (2006). A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. *Journal of Machine Learning Research* 7:2 pp. 149-2187.

Caouette, J., Altman, E. y Narayanan, P. (1998). *Gestión del riesgo de crédito, el próximo gran desafío financiero*. Wiley Frontiers in Finance, vol. Fronteras Wiley en Finanzas, Wiley & Sons, Inc., Nueva York.

Carbó, S. (2011). Lecciones del proceso de reestructuración bancaria en España, Papeles de la Fundación, 42, titulado *Mecanismos de prevención y gestión de futuras crisis bancarias,* pp. 35-47, Fundación de Estudios Financieros. ISBN: 978-84615-5104-0.

Cardona, P. (2004). Aplicación de árboles de decisión en modelos de riesgo crediticio. *Revista Colombiana de Estadística*. Volumen 27. No. 2.

Carpenter, G.A. y Grossberg, S. (1987a). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing,* 37, 54-115.

Carpenter, G.A. y Grossberg, S. (1987b). ART2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, 26, 4919-4930.

Carpenter, G.A. y Grossberg, S. (1990). ART3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks,* 3(4), 129-152.

Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H. y Rosen, D.B. (1992). Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks, 3*, 698-713.

Carpenter, G.A., Grossberg, S. y Reynolds, J.H. (1991). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, 4, 565-588.

Carpenter, G.A., Grossberg, S. y Rosen, D.B. (1991a). ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks, 4*, 493-504.

Carpenter, G.A., Grossberg, S. y Rosen, D.B. (1991b). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks, 4*, 759-771.

Carter, C. y Carlett, J. (1987). Assessing Credit Card Applications Using Machine Learning. *IEEE Expert, 2*(3), 71–79.

Castellanos, J., Pazos, A., Ríos, J. y Zafra, J. L. (1994). Sensitivity analysis on neural networks for meteorological variable forecasting. En J. Vlontzos, J.N. Hwang y E. Wilson (Eds.), *Proceedings of IEEE Workshop on Neural Networks for Signal Processing* (pp. 587-595). New York: IEEE.

Castillo, E, Gutiérrez, J.M. y Hadi, A.S. (1997). Expert Systems and Probabilistic *Network Modela. Springer.*

Castillo, E., Gutiérrez, J.M. y Hadi, A. (1998). *Sistemas Expertos y modelos de redes Probabilísticas. Monografías de la Academia de Ingeniería.*

Catlett, J. (1991). On changing continuous attributtes into ordered discrete attributes. In *Proceedings of European Working Session on Learning*, pages 164–178. Springer-Verlag.

Cellard, J.C., Labbe, B. y Savitsky, G. (1967). Le programme ELISEE, presentation et application. Metra 3 (6), 511-519.

Cerny, V. (1985). Thermodinamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41-45.

Cerquides, J. y López, R. (2005). Robust bayesian linear classifier ensembles. In *Machine Learning*: ECML 2005, pages 72–83. Springer.

Chapman, P., Clinto, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., With, R. (2002). CRISP-DM 1.0. Step-by –step data mining guide.

Charnes, A., Cooper, W. y Rhodes, E. (1978). *Measuring Efficiency of Decision Making Units, European Journal of Operational Research:* 2, 429-444.

Chawla, N.V., Bowyer, K.W., Hall, L.O. y Kegelmeyer, W.P. (2002). SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence research*, pp.321-357.

Chen, S., Cowan, C.F.N. y Grant, P.M. (1991). Orthogonal least squares learning for radial basis function networks. *IEEE Transactions on Neural Networks*, 2, 302-309.

Cheng, B. y Titterington, D.M. (1994). Neural networks: a review from a statistical perspective. *Statistical Science, 9*(1), 2-54.

Cheng, J., Bell, D.A. y Liu, W. (1997). Learning belief networks from data: an information theory based approach. *Proceedings of the sixth international conference on Information and knowledge management*. pp 325 – 331.

Chickering, D.M. (1996). *Learning Bayesian Networks is NP-Complete*. Pp. 121-130.

Chiu, D.K.Y., Cheung, B. y Wong, A.K.C. (1990). Information synthesis based on hierarchical entropy discretization. *Experimental and Theoretical Artificial Intelligence*, 2:117–129.

Choo, E.U. y Wedley, W.C. (1985). Optimal criterion weights in repetitive multicriteria decisión making. *Journal of the Operarational Research Society*. Nº 36, pp. 983-992.

Chow, K., Liu, C.N. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, IT-14(3):462–467.

Coffman, J.Y. (1986). The Proper Role of Tree Analysis in Forecasting the Risk Behaviour of Borrowers. *Management Decision Systems*, MDS Reports 3, 4, 7 & 9.

Cohen, G., Hilario, M., Sax, H., Hugonnet, S. y Geissbuhler, A. (2006). Learning from imbalancing Data in Surveillance of Nosocomial Infection. *Artificial Intelligence in Medicine*, pp. 7-18.

Collett, D. (1991). Modelling binary data. Chapman and Hall. London.

Cooper, G.F. (1991). The computational complexity of probabilistic inference using Bayesian belief nerwork*. Artificial Intelligence* nº 42, 393-405.

Cooper, G., Herskovitz, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*. 9(4): pp.309-348.

Copas, J.B. (1983). Plotting p against x. Applied Statistics, 32:25-31.

Cortes, C. y Vapnik, V. (1995). Support-vector networks, *Machine Learning*, vol.20, pp.273–297.

Cottrell, G.W., Munro, P. y Zipser, D. (1989). Image compression by back propagation: an example of extensional programming. *En N.E. Sharkey (Ed.), Models of cognition: a review of cognitive science* (pp. 208-240). Norwood, NJ: Ablex Publishing Corp.

Cowell, R.G., David, A.P., Lauritzen, S.L. y Spiegelhalter. D.J. (1999). *Probabilistic Networks and Expert Systems*. Springer-Verlag, New York.

Cowell, R. (2001). On searching for optimal classifiers among Bayesian networks. T. Jaakkola and T. Richardson, editors, *Proceedings of the 8th International Conference on Artificial Intelligence and Statistics,* pages 175-180.

Cox, D.R. y Snell, E.J. (1989). Analysis of Binary Data. Second Edition. Chapman & Hall.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematical Control, Signal and Systems*, 2, 303-314.

Davis, R.H., Edelman, D.B. y Gammerman, A. J. (1992). Machine-Learning Algorithms for Credit-Card Applications. *Journal of Management Mathematics*, 4(1), 43-51.

DeLong, E. R., DeLong, D. M. y Clarke-Pearson, D. L. (1988). Comparing the area under two or more correlated receiver operating characteristic curve. *Biometrics*, 44(3), 837-45.

De Campos, L.M. (2006). A scoring function for learning bayesian networks based on mutual information and conditional independence tests. *The Journal of Machine Learning Research*, 7:2149–2187.

De Laurentiis, M. y Ravdin, P.M. (1994). A technique for using neural network analysis to perform survival analysis of censored data. *Cancer Letters,* 77, 127-138.

Desieno, D. (1988). Adding a conscience to competitive learning. Proceedings of the International *Conference on Neural Networks, I*, 117-124.

Deville, J.C. y Tillé, Y. (2004). Eficient balanced sampling: The cube method. *Biometrika*, 91: pp 893_912.

Didzarevich, S., Lizarraga, F., Larrañaga, P., Sierra, B. y Gallego, M.J. (1997). Statistical and machine learning methods in the prediction of bankruptcy, en Sierra Molina, G. y Bonsón Ponte, E. (Eds.): *Intelligent Technologies in Accounting and Business*, Huelva, pp. 85-100.

Dietterich, T.G. (2000). *Ensemble methods in machine learning. Multiple classifier systems, pages 1*–15. Springer.

Doldán F. (2007). Redes bayesianas y riesgo operacional. *Revista Galega de Economía* nº 16.

Domingos, P, (1999): MetaCost A general method for making classifiers cost-sensitive. In: *Fifth International Conference on Knowledge Discovery and Data Mining*, pp.155-164.

Duda, H.; Hart, P. y Stork, D. (2001). *Patternn Classifications*, 2nd ed., Wiley.

Duda, R.O.y Hart, P. E. (1973). *Pattern Classification and Scene Analysis.* New York: John Wiley & Sons.

Durand, D. (1941). *Risk Elements in Consumer Instalment Financing*. National Bureau of Economic Research.

Edwards, W. (1998). Hailfinder. Tools for and experiences with bayesian normative modeling. *American Psychologist,* 53, 416-428.

El-Hay, T. (2001). Efficient Methods for exact and aproximate inference in discrete Graphicals Models. *Master of Science Thesis, Supervisor Nir Friedman*: 17-18.

Elkan, C. (2001). The Foundations of Cost-Sensitive Learning. *In Proceedings of the Seventeenth International Conference of Artificial Intelligence*, 973-978. Seattle, Washington: Morgan Kaufmann.

Elman, J.L. (1990). Finding structure in time. *Cognitive Science,* 14, 179-211.

Emel, A., Oral, M., Reisman, A. y Yolalan, R. (2003). A credit scoring approach for the commercial banking sector. Socio-Economic Planning Sciences 37 (2): 103-123.

Engelbrecht, A.P., Cloete, I. y Zurada, J.M. (1995). *Determining the significance of input parameters using sensitivity analysis.* En J. Mira y F. Sandoval (Eds.),

Proceedings of International Workshop on Artificial Neural Networks (pp. 382-388). New York: Springer.

Engelbrecht, A.P., Fletcher, L. y Cloete, I. (1999). Variance analysis of sensitivity information for pruning multilayer feedforward. *Neural networks*.

Escobar, M. (2007). *El análisis de segmentación: técnicas y aplicaciones de los árboles de clasificación.* Centro de Investigaciones Sociológicas. Cuadernos metodológicos nº 39.

Espin-García, O. y Rodríguez-Caballero, C. (2013). Metodología para un scoring de clientes sin referencias crediticias. Cuadernos de Economía, 32(59), 139-165.

Esteve, S., Sanchís, A. y Sanchís, J.A. (2004): The Determinants of Survival of Spanish Manufacturing Firms, *Review of Industrial Organization*, 25, pp. 251-273.

Etemadi, H., Rostamy, A.A.A., y Dehkordi, H.F. (2009). A genetic programming model for bankruptcy prediction: Empirical evidence from Iran.Expert Systems with Applications, 36(2), 3199-3207.

Facchinetti, G. (2001). *Fuzzy expert systems: Economic and financial applications*. En J. Soldek y J. Pejas (Eds.), Advanced computer system (pp. 3-26). Norwell, Massachussets: Kluwer Academic Publishers.

Facchinetti, G., Bordoni, S. y Mastroleo, G. (2000). *Bank creditworthiness using fuzzy systems: A comparison with a classical analysis tecnique*. Recuperado en noviembre de 2005, de http:// citeseer.ist.psu.edu.

Facchinetti, G., Cosma, S., Mastroleo, G. y Ferretti, R. (2001). *A fuzzy credit rating approach for small firm credit orthiness evaluation in bank lending.* Recuperado en septiembre de 2005, de http://citeseer.ist. psu.edu.

Fahlman, S. E. (1988). *Faster-learning variations on back-propagation: an empirical study*. En D. Touretsky, G.E. Hinton y T. J. Sejnowski (Eds.), Proceedings of the 1988 Connectionist Models Summer School (pp. 38-51). San Mateo: Morgan Kaufmann.

Fahlman, S. E. y Lebiere, C. (1990). *The cascade-correlation learning architecture.* En D.S. Touretzky (Ed.), Advances in neural information processing systems (pp. 524-532). Los Altos, CA: Morgan Kaufmann Publishers.

Falbo, P. (1991). Credit-Scoring by enlarged discriminant models.OMEGA. *The International Journal of Management Science*. Vol. 19. Nº 4. Pp. 275-289.

Falkenstein, E., Boral, A. y Carty, V. (2000). RiskCalcTM for Private Companies: Moody's Default Model Rating Methodology. Moody's Investors Service, *Global Credit Research.*

Faraggi, D. y Reiser, B. (2002). Estimation of the Area under the ROC Curve. *Statistics in Medicine*, 21, pp. 3093 _ 3106.

Fayyard, U.M. y Irani, K.B. (1993*). Multi-interval discretization of continuous valued attributes for classification learning*. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, pp 1022-1027.San Francisco, CA: Morgan Kaufmann.

Ferreira, J.T.A.S., Denison, D.G.T. y Hand, D.J. (2001*). Weighted naive Bayes modelling for data mining. Technical report*. Department of Mathematics, Imperial College, 180 Queen's Gate, London SW7 2BZ, UK.

Ferri, C. (2004). *Multiclasificadores en minería de datos*. Dep. de Sistemes Informàtic i Computació, Universidad Politécnica de Valencia, Spain. Reunión Red Minería de Datos, Madrid.

Fletcher, R. (1987). *Practical Methods of Optimization.* John Wiley & Sons, Nueva York. 2ª edición.

Flexer, A. (1995). *Connectionist and statisticians, friends or foes*. The Austrian Research Institute for Artificial Intelligence. Recuperado 20/01/01, desde acceso FTP: Nombre del servidor: ai.univie.ac.at Archivo: oefai-tr-95-06_ps(1).ps.

Flores, M.J., Gámez J. A., Martínez A.M., Puerta J.M. (2009). GAODE and HAODE: two proposals based on AODE to deal with continuous variables, *in A. P. Danyluk, L. Bottou, M.L. Littman (eds), ICML,* vol. 382 of ACM International Conference Proceeding Series, ACM, p. 40.

Fowlkes. E.B. (1987). Some diagnostics for binary logistic regression via smoothing. *Biometrica,* 74:503-515.

Franco, M. y Vivo, J.M. (2007). *Análisis de curvas ROC. Principios básicos y aplicaciones.* Cuadernos de Estadística. Editorial La Muralla.

Frank, A. y Asuncion, A. (2010). *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science.

Freed, N. y F. Glover. (1981a). A linear programming approach to the discriminant problem. Decision Sciences. 12 68-74.

Freed, N. y Glover, F. (1981b). Simple but powerful goal programming models for discriminant problems. *European Journal of Operational Research*. 7 44-60.

Freund, Y. y Shapire, R.E. (1996). Experiments with a new boosting algoritm. *Machine Learning: Proceedings of the 13th International Conference*, July, 148-156.

Friedman, J.H. (1977). A recursive partitioning decision rule for nonparametrics classification. *IEE Transactions on Computers* pp. 404-408.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, pages 86–92, 1940.

Frydman, H., Altman, E. y Kao, D. (1985). Introducing Recursive Partitioning for Financial Classification: The Case of Financial Distress. *The Journal of Finance*, pp. 269-291.

Friedman, N., Geiger, D. y Goldszmidt, M. (1997): Bayesian Networks classifiers. *Machine Learning, 29*: pp.131-167.

Friedman, N., Getoor, L., Koller, D. y Pfeffer, A. (1999). Learning probabilistic relational models, *Proceedings of the Sixteenth International Joint Conferences on artificial Intelligence*, pp.1300-1309.

Friedman, N. y Goldszmidt, M. (1996). Building classifiers using Bayesian networks. In *Proceedings ofthe 13th National Conference on Artificial Intelligence,* pp 1277-1284, 1996.

Frost, F. y Karri, V. (1999). Determining the influence of input parameters on BP neural network output error using sensitivity analysis. En B. Verma, H. Selvaraj, A. Carvalho y X. Yao (Eds.), *Proceedings of the Third International Conference on Computational Intelligence and Multimedia Applications* (pp.45-49). Los Alamitos, CA: IEEE Computer Society Press.

Fu, L. y Chen, T. (1993). Sensitivity analysis for input vector in multilayer feedforward neural networks. En IEEE (Ed.), *Proceedings of IEEE International Conference on Neural Networks* (pp. 215-218). New York: IEEE.

Fukushima, K. (1988). Neocognitron: a hierarchical neural network model capable of visual pattern recognition. *Neural Networks, 1*(2), 119-130.

Fukushima, K., Miyake, S., y Ito, T. (1983). Neocognitron: a neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics, 13*, 826-834.

Funahashi, K. (1989). On the approximate realization of continuous mapping by neural networks. *Neural Networks*, 2, 183-192.

Gama, J. (2000). A cost-sensitive iterative Bayes. *Proceedings of Workshop on Cost-Sensitive Learning at the 17th International Conference on Machine Learning,* Stanford University, June 2000.

Gama, J. y Brazdil, P., (2000). Cascade Generalization. *Machine Learning*, Kluwer Academic Publishers, Boston, Manufactured in the Netherlands, vol. 41, 3, 315-343.

Garbolino, P. y *Taroni*, F. (2002). Evaluation of scientific evidence using bayesian networks. *Forensic Science International*, 125, pp. 149-155.

García, D. (2006). Manual de Weka. diego.garcia.morate(at)gmail.com.

Gardner, M.J. y Mills, D.L. (1989). Evaluating the likelihood of default on delinquent loans, *Financial Management*, Vol. 18 pp. 55-63.

Garson, G.D. (1991a). Interpreting neural-network connection weights. *AI Expert*, April, 47-51.

Garson, G.D. (1991b). A comparison of neural network and expert systems algorithms with common multivariate procedures for analysis of social science data. *Social Science Computer Review*, 9(3), 399-434.

Gedeon, T.D. (1997). Data mining of inputs: analysing magnitude and functional measures. *International Journal of Neural Systems*, 8(2), 209-218.

Geiger, D. y Heckerman, D. (1995). A characterization of the Dirichlet distribution with application to learning Bayesian networks. P. Besnard y S. Hanks, editors, UAI ’95:

Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence, págs. 196-207. Morgan Kaufmann.

Gehrke, J., Loh, W.Y. y Ramakrishnan, R. (1999b). Classification and regression: money can grow on trees. *Tutorial notes for ACM SIGKDD 1999 international conference on Knowledge Discovery and Data Mining*, August 15-18, 1999, San Diego, California, USA, pp. 1-73.

Geiger, D., Heckerman, D. (1994). Learning Gaussian networks. In *Proceedings of Tenth Conference on Uncertainty in Artificial Intelligence,* Seattle, WA, pp 235-243. Morgan Kaufmann.

Glover, F. (1986). Furure paths for integer programming and link to artificial Intelligence. *Comput. Oper. Res*., 13(5):533-549.

Glover, F., Keene, S. y Duea, B. (1988). A new class of models for the discriminat problema. *Financial Management.* Vol.18 pp. 55-63.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

Glover, F. y Laguna, M. (1997). *Tabu Search*. Kluver Academic Publishers, Norwell, MA, USA.

Goodman, L.A. (1979). Simple Model for Statistica Data analysis of Multivariate Observations. *Journal of the American Statistical Association*, 74: 537-552.

Gordon, T., Kannel, W.B. y Halpebn, M. (1979). Prediction of coronary disease. Journal of Chronic Diseases, 32:427- 440.

Greene, W. (1998). Sample Selection in Credit-Scoring Models. *Japan and the World Economy* 10 (3): 299-316.

Greiner, R. y Zhou, W. (2002). Structural extensión to logistic regression: discriminant parameter learning of belief net classifiers. *Proceedings of the 18th National Conference on Artificial Intelligence,* pages 167-173.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics, 23*, 121-134.

Grossman, D. y Domingos, P. (2004). Learning Bayesian Network Classifiers by Maximizing Conditional Likelihood. *21st International Conference on Machine Learning.*

Guo, Z. y Uhrig, R.E. (1992). Sensitivity analysis and applications to nuclear power plant. *En IEEE (Ed.), International Joint Conference on Neural Networks* (pp. 453-458). Piscataway, NJ: IEEE.

Haberman, S.J. (1978). *Analysis of Qualitative Data.* New York. Academic press.

Han, H., Wang, W. y Mao, B. (2005). Borderline-SMOTE: a new Over-Sampling Method in Imbalanced Data Sets Learning. *En: D.-S. Huanng; X.-P.Zhzng y G.-B. Huang (Eds.), ICICS*, volumen 3644 de LNCS, pp. 878-887.

Hand, D.J. y Henley, W.E. (1997). Statistical Classification. Methods in Costumer Credit Scoring: A review. *Journal of the Royal Statistical Association*, 160(A/ Part3), 523-541.

Hand, D.J. y Till, R.J. (2001). A simple generalization of the area under ROC curve to multiple class classification problems. *Machine Learning* 45 (2), 171-186.

Hanley, J.A. y Hajian-Tilaky, K. (1997). Sampling variability of nonparametrics estimates of the area under receiver operating characteristic curves: An update. *Acad. Radiol*., 4, 49-58.

Hanley, J.A. y McNeil, B.J. (1982). The meaning and use of the área under receiver operating characteristic (ROC) curve. *Radilogy*, 143, 29-36.

Härdle, W., Moro, R.A. y Schäfer, D. (2005). *Predicting Bankruptcy with Support Vector Machines. Statistical Tools for Finance and Insurance*. Springer Verlag, Berlin.

Härdle, W., Moro, R.A. y Schäfer, D. (2007). Estimating probabilities os default with support vector machine. *Discussion paper Series 2: Banking and Financial Studies 2007,* 18, Deutsche Bundesbank, Research Centre.

Härdle, W., Hoffmann, L. y Moro, R. (2011). Learning Machines Supporting bankruptcy. En *Statistical Tools in Finance and insurance*. Springer Verlag, Berlin.

Harrison, R.F., Marshall, J.M. y Kennedy, R.L. (1991). The early diagnosis of heart attacks: a neurocomputational approach. *En IEEE (Ed.), Proceedings of IEEE International conference on Neural Networks* (pp. 231-239). New York: IEEE.

Hartigan, J.A. y Wong, M.A. (1979). Algorithm 136. A k-means clustering algorithm. *Appl. Statist*. 28, 100.

Hashem, S. (1992). Sensitivity analysis for feedforward artificial neural networks with differentiable activation functions. *En IEEE (Ed.), International Joint Conference on Neural Networks* (pp. 419-424). New York: IEEE.

Hastie, T. y Tibshirani, R. (1996). Nonparametric regresión and clasification. PART II-nonparametric classification. En V. Cherkassky, J. H. Friedman y H. Wechsler (eds), From *Statistics to Neural Networks: Theory and Pattern Recognition Applications, Computer and System Sciences*, 136, 70-82.

Hawkins, D.M, y Kass, G. V. (1982). *Automatic Interaction Detection*, en D. M. Hawkins (ed.), Topic in Applied Multivariate Analysis, Cambridge, Cambridge University Press.

Hebb, D. (1949). *The organization of behavior*. New York, Wiley.

Hecht-Nielsen, R. (1987). Counterpropagation networks. *Applied Optics, 26*, 4979-4984.

Hecht-Nielsen, R. (1988). Applications of counterpropagation networks. *Neural Networks, 1*, 131-139.

Hecht-Nielsen, R. (1990). *Neurocomputing*. Reading, MA: Addison-Wesley.

Heckerman, D. (1991). *Probabilistic similarity nets dissertation award series*. MIT Press.

Heckerman, D., Geiger, D. y Chickering, D.M. (1994). Learning Bayesian networks: The combination of knowledge an statistical data. *In R. Lopez de Mantaras and D. Poole, editors, Tenth Conference on Uncertainty in Artificial Intelligene*, pages 293–301. Morgan- Kaufmann.

Heckerman, D. (1996). *A tutorial on learning with Bayesian networks*. Tech. Rep. Nº. MSR-TR-95-06. Redmon, WA: Microsoft Reseach.

Henrion, M. (1988). Propagating uncertainty in Bayesian networks by probabilistic logic sampling. *Uncertainty in Artificial Intelligence*, (2): 149-163.

Hernández, J., Ramírez, M. y Ferri, C. (2004). *Introducción a la minería de datos*. Ediciones Pearson Prentice Hall. Madrid. España. 680 p.

Hinton, G.E. (1989). Connectionist learning procedures. *Artificial Intelligence, 40*, 185-234.

Ho, T.K. (1995). Random Decision Forest. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, QC, 14–16 August 1995. pp. 278–282.

Hoeting, J.A., Madigan, D., Raftery, A.E. y Volinsky, C.T. (1999). Bayesian model averaging: A tutorial. *Statistical science*, pages 382–401.

Holland. J.H. (1975). Adaptation in Natural and Artificial Systems. *The University of Michigan Press (The MIT Press*, London, 1992).

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, 79*, 2554-2558.

Hornik, K., Stinchcombe, M. y White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359-366.

Hosmer, D.W. y Lemeshov, S. (1980). Goodness of _t tests for the multiple logistic regression model. *Communications in Statistics*, 10:1043-1069.

Hosmer, D.W. y Lemeshov, S. (1989). *Applied Logistic Regression.* Wiley.

Hosmer, D.W., Hosmer, T., le Cessie, S. y Lemeshow, S. (1997). A comparison of goodness of it tests for the logistic regression model. *Statistics in Medicine*, 16:965-980.

Hsieh, N.C. (2005). Hybrid mining approach in the design of credit scoring models. *Expert Systems with Applications*, 28(4), 655-665.

Hsieh, N.C. y Hung, L.P. (2010). A data driven ensemble classifier for credit scoring analysis. Expert System with Applictions, 37(1), 534-545.

Huang, J. y Ling, C.X. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299-310.

Huang, J., Tzeng, G. y Ong, C. (2006). Two-stage genetic programming (2SGP) for the credit scoring model. *Applied Mathematics and Computation* 174 (2): 1039-1053.

Huang, C., Chen, M., Wang, C. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications* 33 (4): 847-856.

Hulse, J.V., Khoshgoftaar, T.M. y Napolitano, A. (2007). Experimental perspectives on learning from imbalanced data. En: Z. Ghahramani (Rd.), ICML volume 227 de *ACM International Conference Proceeding series,* pp. 935-942.

Hung, C., y Chen, J.H. (2009). A selective ensemble based on expected probabilities for bankruptcy prediction. Expert System with Applications. 36(3), 5297-5303.

Hunter, A., Kennedy, L., Henry, J. y Ferguson, I. (2000). Application of neural networks and sensitivity analysis to improved prediction of trauma survival. *Computer Methods and Programs in Biomedicine, 62*, 11-19.

Hwang, J.N., Choi, J.J., Oh, S. y Marks, R.J. (1991). Query based learning applied to partially trained multilayer perceptron. *IEEE Transactions on Neural Networks*, 2(1), 131-136.

Inza, P., Larrañaga, R., Etxeberría y Sierra, B. (2000). Feature subset selection by Bayesian network-based optimization. *Artificial Intelligence,* 123:157-184.

Ivanciuc, O. (2007). Applications of Support Vector Machines in Chemistry. *Reviews in Computational Chemistry.* Vol. 23, pp 291-400. Wiley.

Jacobs, R.A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks, 1*(4), 295-308.

Jacobson, T. y Rossbach, K. (2003). Bank Lending Policy credit scoring and value at risk. *Journal of banking and finance* nº 27, pp. 615-633.

Jaeschke, R., Guyatt, G. y Lijmer, J. (2002). Diagnostic tets. En Guyatt, G. y Rennie, D. (eds.). *User' guides to the medical literature*. AMA Press, Chicago, 121-40.

Japkowicz, N. (2001). Concept-Learning in the Presence of Between-Class and Within-Class Imbalances. En: E. Stroulia y S. Matwin (Eds.), *Canadian Conference on AI*, volume 2056 de LNCS, pp. 67-77.

Japkowicz, N. y Stephen, S. (2002). The Class Imbalance Problem: A Systematic Study Intelligent Data*. Analysis, Journal*, volume 6, issue 5, pp: 1-32.

Jensen, F.V. (1996). *An Introduction to Bayesian* Networks. UCL Press.

Jensen, F.V. (2001). *Bayesian Networks and Decisión Graphs.* Springer-Verlag.

Jensen, F.V. y Nielsen, T.D. (2007). *Bayesian Networks and Decisions Graphs. Information Science and Statistics. Series,* Springer Verlag, New York segunda edición: 294 páginas.

Jiang, M. y Lin, S (2010). A Study of Personal Credit Scoring Models Based on Fuzzy ART. Journal of Computational Information Systems 6:9. 2805-2811.

Jiang, L. y Zhang, H. (2006). Weightily averaged one-dependence estimators. *PRICAI 2006: trends in artificial intelligence*, pages 970–974. Springer.

Jiménez, R. (2002). Aportaciones del proceso Data Mining en el análisis de datos (Memoria de investigación en el programa de doctorado del Departamento de Psicología, no publicada). Palma de Mallorca: Universidad de las Islas Baleares.

Jo, T. y Japkowicz, N. (2004). Class imbalances versus small disjuncts. *SIGKDD Explorations*, 6(1): pp.40–49.

Jordan, M.I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, 531-546.

Jordan, M.I. (1998). *Learning in Graphical Models*. Dordrecht, Netherlands: Kluwer.

Jorion, P. (2000). Valor en Riesgo, segundo. EDN, McGraw-Hill, Nueva York.

Kadie, C.M., Hovel, D. y Hovitz, E. (2001). A component-centric toolkit for modeling and inference with bayesian networks. (Tech. Rep. MSR-TR-2001-67). *Redmond, WA: Microsoft Corporation*. 2008, Vol. 13 nº 1, pp. 13-25.

Khan, H.A. y Sempos, C.T. (1989). *Statistica Methods in Epidemiology*. Osford University Press. New York.

Kashani, J.H., Nair, S.S., Rao, V.G., Nair, J. y Reid, J.C. (1996). Relationship of personality, environmental, and DICA variables to adolescent hopelessness: a neural network sensitivity approach. *Journal of American Children and Adolescent Psychiatry*, 35(5), 640-645.

Kass, G.V. (1980). An Exploratory Technique for Investigating Large Quantities of Categorical Data. *Applied Statistic*, 29. pp 119-117.

Kemp, R.A., McAulay, C. y Palcic, B. (1997). Opening the black box: the relationship between neural networks and linear discriminant functions. *Analytical Cellular Pathology, 14*, 19-30.

Keogh, E.J. y Pazzani. M. (1999). Learning augmented Bayesian classifiers: a comparison of distribution-based and non distribution-based approaches. *Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics*, pp. 225-230.

Keogh, E.J. y Pazzani, M.J. (2002). Learning the structure of augmented Bayesian classifiers. *International Journal on Artificial Intelligence Tools*, 11(04):587–601.

Keramati, A. y Yousefi, N. (2011). A Proposed Classification of Data Mining Techniques in Credit Scoring. *International Conference on Industrial Engineering and Operations Management*. Kuala Lumpur, Malaysia.

Kirkpatrick, S., Gelatt, C.D. y Vecchi, M.P. (1983). Optimization by simulated anneling, *Science*, 220(4598):671-680.

Kjærulff, U.B. y Madsen, A.L. (2008). Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis. *Springer Verlag, Series: Information Science and Statistics, New York* XVIII 318 páginas.

Kohavi, R. (1996). Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining,* pages 202-207.

Kohavi, R. y John, G.H. (1997). Wrappers for feature subset selection. *Artificial Intelligence,* 97(1-2) :273-324, 1997.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics, 43*, 59-69.

Kohonen, T. (1984). *Self-organization and associative memory*. Berlin: Springer.

Kohonen, T. (1988). Learning vector quantization. *Neural Networks, 1*, 303.

Kohonen, T. (1995). *Self-organizing maps*. Berlin: Springer-Verlag.

Kosko, B. (1992). *Neural networks and fuzzy systems*. Englewood Cliffs, NJ: Prentice-Hall.

Kolmogorov, A.N. (1957). On the representation of continuous functions of several variables by means of superpositions of continuous functions of one variable. *Doklady Akademii Nauk SSSR, 114*, 953-956.

Koza, J.R. (1992). *Genetic Programming On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

Krzanowski, W. y Hand, D. (2009). *ROC Curves for Continuous Data*. Taylor and Francis Group.

Kubat, M. y Matwin, S. (1997). Addressing the Course of Imbalanced Training Sets: One-Sided Selection. En: *D.H.Fisher* (Ed.), ICML, pp. 179-186.

Kumar, A. y Olmeda, I. (1999). A Study of Composite or Hybrid Classifieres for Knowledge Discovery. *INFORMS Journal of Computing*. Vol. 11: 267-277.

Kuncheva, L. y Jain. L.C. (1999). Nearest neighbor classifier: Simultaneous editing and feature selection. *Pattern Recognition Letters,* pp. 1149-1156.

Kuncheva, L.I. (2002). A Theorical study on six classifier fusion strategies. *IEEE Transaction on PAMI*, 24 (2), pp. 281-286.

Lam, K. F., Choo, E.U. y Wedley, W. (1993). Lineal goal programming in estimation of classification probability. *European Journal of Operational Research*. Nº 67, pp.101-110.

Lam, K.F., Choo, E.U. y Moy, W.C. (1996). Minimizing deviations from the group mean: A new linear programming approach for the two-group classification problem. *European Journal of Operational Research*. Nº 88, pp. 358-367.

Lam, W. y Bacchus, F. (1994). Learning Bayesian belief networks: an approach based on the MDL principle. *Comp. Intelligence,* 10: pp. 269-293.

Landwehr, J.M., Pregiboon, D. y Shoemaker, A.C. (1984). Graphical methods for assessing logistic regression models. Journal of the American Statistical Association, 79:61-83.

Lang, K.J., Waibel, A.H. y Hinton, G. (1990). A time-delay neural network architecture for isolated word recognition. *Neural Networks, 3*23-344.

Langley, P.W., Iba, P. y Thompson. K. (1992). An analysis of Bayesian classifiers. In *Proceedings, Tenth National. Conference on Artificial Intelligence* (pp. 223–228). Menlo Park, CA: AAAI Press.

Langseth, H. y Nielsen, T.D. (2002). *Classification using hierarchical naïve-Bayes models.* URL citeseer.ist.psu.edu/langseth02classification.html.

Lara, R.J. (2010). *La gestión de crédito en las instituciones de microfinanzas.* Tesis doctoral. Universidad de Granada.

Larrañaga, P., Poza, M., Yurramendi, Y., Murga, R.H. y Kuijpers, C.M.H. (1996). Structure learning of Bayesian networks by genetic algorithms: a performance analysis of control parameters. *Pattern Analysis and Machine Intelligence, IEEE Transactions* on Sep 1996. Volume: 18, Issue: 9. Page(s): pp. 912 – 926.

Lahsana, A., Ainon, R. y Wah, T. (2010). Credit Scoring Models Using Soft Computing Methods: A Survey. *The International Arab Journal of Information Technology,* 7(2): 29-139.

Laurikkala, J. (2002). Instance-based data reduction for improved identification of difficult small classes. *Intelligent Data Analysis*, pp.311-322.

Lawrence, E.C. y Arshadi, N (1995). A multinomial logit analysis of problem loan resolution choices in banking. Journal of Money, *Credit and Banking*, 27(1), pp. 202-216.

Le Cessie, S. y Van Houwelingen, J.C. (1991). A goodness of _t test for binary regression models, based on smoothing methods. *Biometrics,* 47:1267-1282.

Le Cessie, S. y Van Houwelingen, J.C. (1992). Ridge estimators in Logistic Regression. *Applied Statistics,* 41:191-201.

Lee, A.H., Stevenson, M.R., Wang, K. y Yau, K.K.W. (2002). Modeling Young Driver Motor Vehicle Crashes: Data with Extra Zeros. *Accident Analysis and Prevention*, 34, 4, pp. 515-521.

Lee, S. y Kil, R. (1991). A Gaussian potential function network with hierarchically self-organizing learning. *Neural Networks.* Volume 4, Issue 2, Pages 207–224.

Li, S.T., Shiue, W. y Huang, M.H. (2006) The evaluation of consumer loans using support vector machine. Expert Systems with Applications. Vol. nº 30, pp. 655-665.

Ling, C.X., Yang, Q., Wang, J. y Zhang, S. (2004). Decision trees with minimal costs. *In ICML '04: Proceedings of the twenty-first international conference on Machine earning*, page 69.

Liu, J.N.K., Li, B.N.L. y Dillon, T.S. (2001). An improved naive Bayesian classifier technique coupled with a novel input solution method. *IEEE Transactions on Systems, Man and Cybemetics - Part C: Applications and Reviews,* 31(2):249- 256.

Lyn, T., Edelman, D. y Crook, J. (2002). *Credit Scoring and its Applications*. SIAM. Filadelfia, USA.

Lizotte, D., Madani, O. y Greiner, R. (2003). Budgeted Learning of Naïve-Bayes Classifiers. *In Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*. Acapulco, Mexico: Morgan Kaufmann.

Loh, W.Y. y Shih, Y.S. (1997). Split Selection Method for Classification Trees. *Statistica Sinica,* 7:815-840.

Loh, W.Y. y Vanichsetakul, N. (1988). Tree-structured classification via generalized discriminant analysis (with discussion). J. *Amer. Statist. Assoc.* 83, 715-728.

López, J., García, J. y De la Fuente, L. (2006). Modelado causal con redes bayesianas. *Actas de las XXVII Jornadas de Automática*, 198–202.

López de Mantaras, R. (1991). A Distance-Based Attribute Selection Measure for Decision Tree Induction. *Machine Learning*, 6, pp. 81-92.

López, V., Fernández, A. y Herrera, F. (*2010).* Un primer estudio sobre el uso de aprendizaje sensible al coste con sistemas de clasificación basados en reglas difusas para problemas no balanceados**.** *In Proceedings of the III Congreso Español de Informática (CEDI 2010). III Simposio sobre Lógica Fuzzy y Soft Computing, LFSC2010 (EUSFLAT), Valencia (Spain), 459-466.*

McKee, T., Lensberg, T. (2002). Genetic programming and rough sets: A hybrid approach to bankruptcy classification. *European Journal of Operational Research 138 (2): 436-451.*

Maddala, G.S. (1983). *Limited-dependent and qualitative variables in econometrics*. Cambridge University Press.

Madgison, J. (1989). *SPSS/PC + CHAID.* Chicago, SPSSS Inc.

Madgison, J. (1992). Chi-Squared Analysis of a Scalable Dependen Variable. *Procedings of the 1992 Annual Meeting of the American Statistical Association*.

Madsen, A.L. y Jensen, F.V. (1999). Lazy propagation: A junction tree inference algorithm based or lazy evaluation. *Artificial Intelligence* 113(1-2): 203-245.

Makowski, P. (1985). Credit Scoring Branches Out: Decision Tree - *Recent Technology. Credit World*, 75, 30-37.

Malhotra, R. y Malhotra, D.K. (2002). Differentiating between good and bad credits using neurofuzzy systems. *European Journal of Operational Research*, 136, 190-211.

Malhotra, R., Malhotra, D.K. (2003). Evaluating consumer loans using Neural Networks. *Omega the International Journal of Management Science* 31 (2): 83-96.

Mallo, F. (2011). Modelos multivariantes internos de medición de riesgos de crédito acordes con Basilea II. Tesis doctoral. Universidad de Salamanca.

Marczyk, A. (2004). Genetic algorithms and evolutionary computation. *The Talk, Origins Archive.* 23 Apr. 2004. 7 Oct. 2006.

Marais, M.L; Patell, J. y Wolfson, M. (1984). The Experimental Design of Classification Models: An Application of Recursive Partitioning and Bootstrapping to Commercial Bank Loan Classifications. *Journal of Accounting Research*, pp.87-114.

Mardia, K.V., Kent, J.T. y Bibby, J.M. (1979). *Multivariate Analysis*. Academic Press, London.

Martín, B., Sanz, A. (2001). *Redes neuronales y Sistemas Borrosos*. Ed. Rama.

Martin, J.K. (1997). An Exact Probability Metric for Decision Tree Splittingand Stopping. *Machine Learning*, 28, pp. 257-291.

Martín, J.L. (1985). *El pronóstico del fracaso empresarial.* Publicaciones de la Universidad de Sevilla, Sevilla.

Martínez, I. y Rodríguez, C. (2003). Modelos gráficos. En Y. del Águila et. al. (Eds.), *Técnicas estadísticas aplicadas al análisis de datos* (pp. 217-257). Almería: Servicio de Publicaciones de la Universidad de Almería.

Martínez, J.F. y Venegas, F. (2013) Riesgo operacional en el proceso de pago del Procampo. Un enfoque bayesiano. *Revista Contaduría y Administración*. vol.58 nº 2.

Martínez de Pisón, F.J. (2003). *Optimización mediante técnicas de minería de datos del ciclo de recocido simulado de una línea de galvanizado*. *Tesis doctoral. Universidad de La Rioja.*

Melville. P. y Mooney. R.J. (2003). Constructing diverse classifier ensembles using artificial training examples. *In Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 505–510.

Merton, R.C. (1974). On the pricing of corporate debt: the risk structure of interest rates. *Journal of Finance* 29, 449–70.

Mester, L.J. (1997). What's the Point of Credit Scoring? *Business Review*, Set./Oct., pp. 3-16, Federal Reserve Bank of Philadelphia.

Milne, K. (1995). Feature selection using neural networks with contribution measures. *En IEEE (Ed.), Proceedings of Australian Conference of Artificial Intelligence* (pp. 124-136). Sydney: IEEE West Australian Section.

Mitchell, T.M. (1997). *Machin Learning*. MacGraw-Hill, 1997.

Modai, I., Saban, N. I., Stoler, M., Valevski, A. y Saban, N. (1995). Sensitivity profile of 41 psychiatric parameters determined by neural network in relation to 8-week outcome. *Computers in Human Behavior*, 11(2), 181-190.

Molina, J. y Garcías, J. (2006). *Técnicas de análisis de datos. Aplicaciones prácticas utilizando Microsoft Excel y WEKA*. Universidad Carlos III de Madrid. Madrid. España. 266 p.

Montaño, J.J. (2005). *Herramientas informáticas de redes neuronales artificiales. Aplicaciones prácticas*. Universidad de Les Illes Balears.

Montaño, J.J. y Palmer, A. (2003). Numeric sensitivity analysis applied to feedforward neural networks. *Neural Computing & Applications, 12*, 119-125.

Montaño, J.J., Palmer, A. y Fernández, C. (2002). Redes neuronales artificiales: abriendo la caja negra. *Metodología de las Ciencias del Comportamiento, 4*(1), 77-93.

Montaño, J.J., Palmer, A. y Muñoz, P. (2011). Artificial neural networks applied to forecasting time series. *Psicothema.* Vol. 23, nº 2, pp.322-329.

Moody, J. y Darken, C.J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation, 1*, 281-294.

Moreno, J.F. y Melo, L.F (2011). Pronóstico de incumplimiento de pago mediante máquinas de vectores soporte: una aproximación inicial a la gestión del riesgo de credito. *Borradores de Economía*, nº 677. Banco de la República de Colombia.

Morgan, J.N. y Sonquist, J.A. (1963). Problems in the Analysis of Survey Data and a Proposal. *Journal of the American Statistical Association*, 58: 415-434.

Morgan, M.G.y Henrion, M. (1990). *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*. Cambridge University Press, New York.

Mures, M.J., García, A., y Vallejo, M.E. (2005). Aplicación del análisis discriminante y regresión logística en el estudio de la morosidad de las entidades financieras. Comparación de resultados. *Pecvnia,* 1, 175-199.

Murphy, P.M. y Aha, D.W. (1995). *UCI repository of machine learning databases*. http://www.ics.uci.edu/~mlearn/.

Myers, J.H. y Forgy, E.W. (1963). Development of Numerical Credit Evaluation Systems. *Journal of American Statistical Association* 50, 797-806.

Nadaraya, E.A. (1964). On estimation regression. *Theory of Probability and Its Applications,* 9:141-142.

Nadeau, C. y Bengio, Y. (1999). "Inference for the Generalization Error", in S. Solla, T. Leen, Klaus-Robert Muller, eds, *Advances in Neural Information Processing Systems, MIT Press.* pp. 307-313.

Nadkarni, S., y Shenoy, P.P. (2001). A bayesian network approach to making inferences in causal maps. *European Journal of Operational Research*, 128, 479-498.

Nadkarni, S., y Shenoy, P.P. (2004). A causal mapping approach to constructing bayesian networks. *Decision Support Systems,* 38, 259-281.

Nagelkerke, N.J.D. (1991). A Note on a General Definition of the Coefficient of Determination. Biometrika 78(3): 691-692.

Neapolitan R.E. (1990). *Probabilistic Reasoning in Expert Systems: Theory and Algorithms.* John Wiley & Sons, New York.

Neapolitan, R.E. (2004). *Learning bayesian networks*. Pearson Prentice Hall.

Nelson, M.M., y Illingworth, W.T. (1991). *A practical guide to neural nets (Vol. 1).* Reading, MA: Addison-Wesley.

Nielsen, T.D., y Jensen, F.V. (2009). Bayesian networks and decision graphs. Springer Science & Business Media.

Nisbet, R., Elder IV, J. y Miner, G. (2009). *Handbook of statistical analysis and data mining applications*. Academic Press.

Obuchowski, N.A. (1994). Computing sample size for receiver operating characteristics studies. *Invest. Radiol*., 29, 238-243.

Ohno-Machado, L. y Musen, M.A. (1997b). Sequential versus standard neural networks for pattern recognition: an example using the domain of coronary heart disease. *Computational Biology in Medicine, 27*(4), 267-281.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology, 15*, 267-273.

Oja, E. (1989). Neural networks, principal components, and subspaces. *International Journal of Neural Systems, 1*, 61-68.

Olmeda, I. y Fernández, E. (1997). Hybrid classifiers for Financial Multicriteria Decision Making: The case of Bankruptcy Prediction. *Computational Economics:* 1-19.

Opara, J., Primozic, S. y Cvelbar, P. (1999). Prediction of pharmacokinetic parameters and the assessment of their variability in bioequivalence studies by artificial neural networks. *Pharmaceutical Research*, 16(6), 944-948.

Orgler, Y.E. (1970). A Credit Scoring Model for Commercial Loans. *Journal of Money, Credit and Banking*, 2(4), 435-445.

Orgler, Y.E. (1971). Evaluation of Bank Consumer Loans with Credit Scoring Models. Journal of Bank Research, 2, 31-37.

Palmer, A., Montaño, J.J. y Calafat, A. (2000). Predicción del consumo de éxtasis a partir de redes neuronales artificiales. *Adicciones, 12*(1), 29-41.

Parmanto, B.W., Munro, P. y Howard, R. (1996). Reducing Variance of Committee Prediction with Resampling Techniques. *Connection Science*. Volume 8, Issue 3-4.

Pearl, V. (1988). *Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference.* Morgan Kaufmann Publisher, Inc.

Peña, J.M., Lozano, J.A. y Larrañaga, P. (2002). Learning recursive Bayesian multinets for data clustering by means of constructive induction. *Machine Learning,* 47:63-89.

Pepe, M.S. (2003). *The Statistical Evaluation of Medical Tests for Classification and Prediction*. Oxford University Press Inc, New York.

Pérez, J.M. (2006). *Árboles Consolidados: Construcción de un árbol de clasificación basado en múltiples submuestras sin renunciar a la explicación.* Tesis doctoral. Universidad del País Vasco.

Pfahringer, B. (1987). Compression-based discretization of continuous attributes. In *Proceedings of the 20th International Conference on Machine Learning*. Morgan Kaufmann, 1995. *Intelligence*, 9(6):205–218.

Pineda, F.J. (1989). Recurrent back-propagation and the dynamical approach to neural computation. *Neural Computation, 1*, 161-172.

Plotnicki, B. (2005). Modelo de comportamiento y predicción de incumplimiento crediticio: el caso de empresas Pyme en Argentina. *Temas de Management*, Vol. 3. Pp. 15-19.

Poggio, P. y Girosi, F. (1990). Networks for Approximation and Learning, *Proc. IEEE,* Vol 78, No. 9, Sept.

Provost, F. (2003). Machine learning from imbalanced data sets 101 (Extended Abstract). *En: AAAI: Workshop on Learning with Imbalanced Data Sets*.

Provost, F. y Fawcett. T. (2001). Robust classification for imprecise environments. *Machine Learning Journal,* 42(3):203-231.

Quinlan, J.R. (1986b). Learning Decision Tree Classifiers. *ACM Computing Surveys*, 28:1, March 1996, pp. 71-72.

Quinlan, J.R. (1987). Symplifying Decision Trees. *International Journal of Man-Machine Studies.* Nº 27, pp. 221-234.

Quinlan, J.R. (1993). *C4.5: Programs for machine learning*, Morgan Kaufmann Publishers, Inc., California (USA).

Rambhia, A.H., Glenny, R. y Hwang, J. (1999). Critical input data channels selection for progressive work exercise test by neural network sensitivity analysis*. En IEEE (Ed.), IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. 1097-1100). Piscataway, NJ: IEEE.

Ramírez, A. (2008). Técnica de minería de datos aplicadas a a construcción de modelos de score crediticio: estado del arte. Universidad de Colombia.

Ramoni. M. y Sebastiani, P. (2001). Robust learning with missing data. *Machine Leaming,* 45(2):147-170.

Rastogi, R. y Shim, K. (2000). PUBLIC: A Decision Tree Classifier that integrates building and pruning. *Data Mining and Knowledge Discovery*, Vol. 4, No. 4, pp. 315-344.

Reid, J. C., Nair, S.S., Kashani, J.H. y Rao, V.G. (1994). *Detecting dysfunctional behavior in adolescents: the examination of relationships using neural networks*. En P.W. Lefley (Ed.), Proceedings of Annual Symposium of Computational Applications on Medical Care (pp. 743-746). New York: Springer.

Riedmiller, M. y Braun, H. (1993). A direct adaptive method for faster backpropagation learning: the Rprop algorithm. *IEEE International Conference on Neural Networks*, 586-591.

Ripley, B.D. (1994). Neural networks and related methods for classification. *Journal of the Royal Statistical Society, 56*(3), 409-456.

Robles, V. (2003). *Clasificación Supervisada basada en redes bayesianas. Aplicación en biología computacional*. Universidad Politécnica de Madrid, Facultad de Medicina. Tesis Doctoral.

Rodríguez – Vilariño, M.L. (1995). Predicción de la solvencia empresarial por medio del análisis logit. *Análisis Financiero* 65 (en-abr9. pp 68-78).

Rosenberg, E. y Gleit, A. (1994). Quantitative Methods in Credit Management: A Survey. *Operations Research*, 42, 589-613.

Rosenblatt, F. (1958). The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review, 65*, 386-408.

Rumelhart, D.E., Hinton, G.E. y Williams, R.J. (1986). Learning internal representations by error propagation. En D. E. Rumelhart y J.L. McClelland (Eds.), *Parallel distributed processing* (pp. 318-362). Cambridge, MA: MIT Press.

Rzempoluck, E.J. (1998). *Neural network data analysis using Simulnet. New York*. Springer-Verlag.

Sadatrasoul, S.M., Gholamian, M.R., Hajimohammadi, Z. y Hosseini,. M. (2014). Utility based Credit Scoring for Banks and Financial Institutions: Case Study of a Major Iranian Bank. *Journal of Mathematics and Computer Science*. Nº13, 281-287.

Sahami, M. (1996). Learning limited dependence bayesian classifiers. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 335–338.

Salmerón, A. (1998). Algoritmos de propagación II: Métodos de monte Carlo. En J. A. Gámez y J. M. Puerta editores, *Sistemas Expertos Probabilísticos, Ciencia y Técnica* nº 20, págs. 65-88. Universidad de Castilla-La mancha, 1 ed.

Sanger, T.D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks, 2*, 459-473.

Sarkar, S. y Sriram, R. (2001). Bayesian models for early warning of bank failures. *Management Science* 47 (11): 1457-1475.

Sarle, W.S. (1994). Neural networks and statistical models. En SAS Institute (Ed.), *Proceedings of the 19th Annual SAS Users Group International Conference* (pp.1538-1550). Cary, NC: SAS Institute.

Sarle, W.S. (2000). *How to measure importance of inputs?* Recuperado 2/11/01, desde ftp://ftp.sas.com/pub/neural/importance.html.

Sarle, W.S. (2002). *Neural network FAQ*. Recuperado 20/04/02, desde ftp://ftp.sas.com/pub/neural/FAQ.html.

SAS. (2001). A proven Data Mining Process. Dirección web: http://www.sas.com/en_us/software/analytics/enterprise-miner.htmlsemma.html

Schachter, R.D., Anderson, S.K. y Szolovits, P. (1994). Global Conditioning for Probabilistic Inference in Belief Networks. *Proceedings of the Uncertainty in AI Conference, San Francisco, CA, Morgan Kaufman*: 514–522.

Sheng, V.S. y Ling, C.X. (2006). Thresholding for making classifiers cost-sensitive. In *Proceedings of the National Conference on Artificial Intelligence* (Vol. 21, No. 1, p. 476). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

Shachmurove, Y. (2002). *Applying artificial neural networks to business, economics and finance.* University of Pennsylvania, Center for Analytic Research in Economics and the Social Sciences.

Shenoy, P.P. (1992). Valuation-based systems for Bayesian decision analysis. *Operation Research* 40(**3**): 463-484.

Shih, Y.S. (1999)*. Families of splitting criteria for classification trees.* Statistics and Computing, vol.9, no.4; Oct. 1999; pp. 309-315.

Siddiqi, N. (2006). *Credit Risk Scorecards. Developing and implementing intelligent credit scoring*. SAS Institute Inc. New Jersey. U.S.A.

Spackman, K.A. (1992). Maximum likelihood training of connectionist models: comparison with least-squares backpropagation and logistic regression. En IEEE (Ed.), *Proceedings of the 15th Annual Symposium of Computer Applications in Medical Care* (pp. 285-289). New York:

Specht, D.F. (1990). Probabilistic neural networks. *Neural Networks, 3*, 110-118.

Specht, D.F. (1991). A generalized regression neural network. *IEEE Transactions on Neural Networks, 2*, 568-576.

Spiegelhalter, D.J., Lauritzen, S.L. (1990). Sequential updating of conditional probabilities on directed graph structures. *Network,* 20, pp. 579-605.

Steenackers, A., Goovaerts, M. J. (1989). A credit scoring model for personal loans. Insurance. *Mathematics and Economics*. Nº. 8. Pp 31-34.

Sun, L., y Shenoy, P.P. (2007). Using Bayesian networks for bankruptcy prediction: Some methodological issues. *European Journal of Operational Research*, *180*(2), 738-753.

Suzuki, J. (1996). Learning Bayesian Belief Network Based on the Minimum Description Length Principle: An Efficient Algorithm Using the B&B Technique. *In Proceedings of the Thirteenth International Conference on Machine Learning,* pp 462-470.

Swets, J.A. y Pickett, R.M. (1982). *Evaluation of Diagnostic system.* Academic Press, Inc, New York.

Takenaga, H., Abe, S., Takatoo, M., Kayama, M., Kitamura, T. y Okuyama, Y. (1991). Input layer optimization of neural networks by sensitivity analysis and its application to recognition of numerals. *Transactions of the Institute of Electrical Engineers Japan*, 111(1), 36-44.

Taylor, P.C. y Silverman, B.W. (1993). Block diagrams and splitting criteria for classification trees. *Statistics and Computing*, vol. 3, no. 4, pp.163-167.

Thomas, L.C. (2000). A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. *International Journal of Forecasting*, 16, p. 149-172.

Ting, K.M. (1998). Inducing cost-sensitive trees via instance weighting. *En Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery,* pages 139-147.

Tsaih, R. (1999). Sensitivity analysis, neural networks, and the finance. *En IEEE (Ed.), International Joint Conference on Neural Networks* (pp. 3830-3835). Piscataway, NJ: IEEE.

Tsaih, R., Liu, Y., Liu, W. y Lien, Y. (2004). Credit scoring system for small business loans. *Decision Support Systems* 38 (1): 91-99.

Tsiatis, A.A. y Duncan, D.B. (1967). Estimation of the probability of an event a function of several independen variables. *Biometrica,* S4, 167-179.

Turney, P.D. (1995). Cost-Sensitive Classification: Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm. *Journal of Artificial Intelligence Research* 2:369-409.

Turney, P.D. (2000). Types of cost in inductive concept learning. *In Proceedings of the Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning*, Stanford University, California.

Uestuen, B., Melssen, W.J. y Buydens, L.M.C. (2006). Facilitating the application of Support Vector Regression by using a universal Pearson VII function based kernel. *Chemometrics and Intelligent Laboratory Systems*. 81:29-4.

Urquizo, D. y Mendivil, F. (2011). Aplicación de Redes Neuronales Artificiales para el Análisis de la Inflación en Bolivia. *4º Encuentro de Economistas de Bolivia.*

Van Ooyen, A. y Nienhuis, B. (1992). Improving the convergence of the backpropagation algorithm. *Neural Networks, 5*, 465-471.

Vicino, F. (1998). Some reflections on artificial neural networks and statistics: two ways of obtaining solutions by working with data. *Substance Use & Misuse, 33*(2), 221-231.

Vapnik, V. (2000). *The Nature of Statistical Learning Theory*. New York, Ed. Springer Verlag.

Vapnik, V. (1998). *Statistical Learning Theory*. Jhon Wiley and Sons, New York.

Vapnik, V. (1995). The Nature of *Statistical Learning Theory*. Jhon Wiley and Sons, New York.

Wan, E.A. (1990). Temporal backpropagation: an efficient algorithm for finite impulse response neural networks. En D.S. Touretzky, J.L. Elman, T.J. Sejnowski y G.E. Hinton (Eds.), *Proceedings of the 1990 Connectionist Models Summer School* (pp. 131-140). San Mateo, CA: Morgan Kaufmann.

Wang, J., Xu, M., Wang, H. y Zhang, J. (2006). Clasification of Imbalanced Data by Using the SMOTE Algorithm and locally Linear Embedding. *En: ICSP,* volume 3, pp. 16-20.

Watson, G.S. (1964). Smooth regression analysis. *Sankhya Series A*, 26:359-372.

Wang, G., Hao, J., Ma, J. y Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring.Expert systems with applications. 38(1), 223-230.

Webb, G.I., Boughton, J.R. y Wang, Z. (2005). Not so naive bayes: aggregating one-dependence estimators. *Machine Learning*, 58(1):5–24.

Webb, G.I. y Pazzani, M.J. (1998). Adjusted probability naive Bayesian induction. *Proceedings of the llth Australian Joint Conference on Artificial Intelligence,* pages 285-295.

Werbos, P.J. (1990). Backpropagation through time: What it is and how to do it. *Proceedings of the IEEE*, 78, 1550-1560.

West, D. (2000). Neural network credit scoring models, *Computers & Operations Research,* vol. 27, pp. 1131–1152.

Widrow, B. y Hoff, M. (1960). Adaptive switching circuits. En J. Anderson y E. Rosenfeld (Eds.), *Neurocomputing* (pp. 126-134). Cambridge, Mass.: The MIT Press.

Wiginton, J.C. (1980). A note on the comparison of logit and discriminant models of consumer credit behavior. *Journal of Financial and Quantitative Analysis*. Vol.15. No. 3.pp. 757-770.

Wilson, D.L. (1972). Asymptotic properties of nearest neighbor rules using edited data, IEEE Transactions on Systems, Man and Cybernetics. *IEEE Computer Society Press*, Los Alamos.

Williams, R.J. y Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation, 1*, 270-280.

Williamson, J.R. (1995). *Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps* (Informe técnico Nº CAS/CNS-95-003). Boston: Boston University, Center of Adaptive Systems and Department of Cognitive and Neural Systems.

Witten, I.H. y Frank, E. (1999). Data Mining: *Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

Wong, A.K.C. y Chiu, D.K.Y. (1987). Synthesizing statistical knowledge from incomplete mixed-mode data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(6):205–218.

Xia, Y., Liu, B., Wang, S., Lai, K.K. (2000). A model for portfolio selection with order of expected returns. *Computers & Operations Research* 27 (5): 409-422.

Xu, X., Zou, C. y Wang, Z. (2009). Credit scoring algorithm based on link analysis ranking with support vector machine. *Expert Systems with Applictions*. 36(2 , Part 2), 2625-2632.

Yamauchi, K.T., Templer, D.I. (1982), The development of a money attitude scale, *Journal of PersonalityAssessment*, Vol. 46 pp.522-8

Yang, Z., Wang, Y., Bai, Y. y Zhang, X. (2004). Measuring Scorecard Performance. *Computational Science*-ICCS LNCS 3039, 900-906.

Yang, C.H., Motohashib, K. y Chenc J.R., (2009): Are new technology-based firms located on science parks really more innovative, Evidence from Taiwan, *Research Policy*, vol. 38, p.77–85.

Yeung, D.S., Cloete, I., Shi, D., y Ng, W.W. (2010). Sensitivity Analysis for Neural Networks. *Natural Computing*.

Yang, Y., Webb, G.I., Cerquides, J., Korb, K.B., Boughton, J., y Ting, K.M. (2007). To select or to weigh: A comparative study of linear combination schemes for superparent-one-dependence estimators. *Knowledge and Data Engineering, IEEE Transactions* on, 19(12), 1652-1665.

Yoon, Y.O., Brobst, R.W., Bergstresser, P.R. y Peterson, L.L. (1989). A desktop neural network for dermatology diagnosis. *Journal of Neural Network Computing*, 1, 43-52.

Yoon, Y., Swales, G. y Margavio, T.M. (1993). A comparison of discriminant analysis versus artificial neural networks. *Journal of the Operational Research Society*, 44(1), 51-60.

Yu, L., Wang, S. y Lai, K.K. (2008). Credit risk assessment with a multistage neural network ensemble learning approach. *Expert Systems with Applications*, 34(2), 1434-1444.

Yu, L., Wang, S. y Lai, K.K. (2009). An intelligent-agent-based fuzzy group decision making model for financial multicriteria decision support: the case of credit scoring. *European Journal of Operational Research*, 195(3), 942-959.

Yu, L., Yue, W., Wang, S. y Lai, K.K. (2010). Support Vector Machine based multiagent ensemble learning for credit rik evaluation. *Expert System with Applications.* Vol nº 2, pp. 1351-1360.

Zadrozny, B. y Elkan, C. (2001). Learning and Making Decisions When Costs and Probabilities are Both Unknown. *In Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*, 204-213.

Zang, D, Zhou, X., Leung, S.C.H y Zheng, J. (2010). Vertical bgging decisión tres model for credit scoring. *Expert System with Applications*. Volume 37, Issue 12. Pages 7838–7843.

Zaffalon, M. (2002). The naive credal classifier. *Journal of Statistical Planning and Inference,* 105(1):5-21.

Zhang, G., Hu, M.Y., Patuwo, B.E. y Indro, D.C. (1999). Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis*. European journal of operational research,*116(1), 16-32.

Zhang, J. y Mani, I. (2003). kNN approach to unbalanced data distributions: a case study involving information extraction. *En ICML*: *Workshop on Learning from Imbalanced Dataset* II.

Zhang, N. y Poole, D. (1996). Exploiting Cusal Independence in Byesian Network Inference. *Journal of Artificial Intelligence Research*, v.5, p. 301-328.

Zheng, Z. y Webb, G.I. (2000). Lazy learning of Bayesian rules. *Machine Learning,* 41:53-84.

Zheng, F. y Webb, G.I. (2006). Efficient lazy elimination for averaged one-dependence estimators. *Proceedings of the 23rd international conference on Machine learning*, pages 1113–1120. ACM.

Zheng, F. y Webb, G.I. (2007). Finding the right family: parent and child selection for averaged one-dependence estimators. *Machine Learning*: ECML 2007, pages 490–501. Springer.

Zurada, J.M., Malinowski, A. y Cloete, I. (1994). Sensitivity analysis for minimization of input data dimension for feedforward neural network. *En IEEE (Ed.), Proceedings of IEEE International Symposium on Circuits and Systems* (pp. 447-450). New York: IEEE.

Zweig, M.H. y Cambell, G. (1993). Receiver–Operating Characteristic (ROC) Plots: A Fundamental Evaluation Tool in Clinical. Medicine. *Clin. Chem*., 39 (4), 561-577. [Correcciones en *Clin. Chem.,* (1993), 39, 1589].

ANEXOS

ANEXO 1

CÓDIGO EN EL PROGRAMA R DE LA APLICACIÓN DEL MÉTODO DEL CUBO.

ANEXO 1: CÓDIGO EN EL PROGRAMA R DE LA APLICACIÓN DEL MÉTODO DEL CUBO.

ANEXO 1. Código en el programa R de la aplicación del método del Cubo.

En este Anexo se explica el código del programa utilizado en R del método de extracción de muestras con el método del Cubo. Este programa utiliza la librería Sampling para obtener la selección de la muestra a través del submuestreo equilibrado donde los totales coinciden con los estimadores de Horvitz-Thompson.

```
# Cargamos el archivo "ficherotesis"
# Introducir la dirección del archivo ficherotesis.rda

load("C:/TESIS_DOCTORAL/ficherotesis.rda")


# Cargamos el paquete sampling

library(sampling)


# Codificamos las variables cualitativas como factores

as.factor(ficherotesis$CIVIL)->ficherotesis$CIVIL
as.factor(ficherotesis$NACIONALIDAD)->ficherotesis$NACIONALIDAD
as.factor(ficherotesis$TIPOTRABAJO)->ficherotesis$TIPOTRABAJO
as.factor(ficherotesis$VIVIENDA)->ficherotesis$VIVIENDA
as.factor(ficherotesis$FINALIDAD)->ficherotesis$FINALIDAD


# Tomamos como población para el muestreo aquellos que forman la clase A

sum(ficherotesis$SELEC=='A')->nA
sum(ficherotesis$SELEC=='B')->nB
ficherotesis=ficherotesis[ficherotesis$SELEC=="A",]


# Creamos las variables indicadores para cada una de las variables de equilibrio

disjunctive(ficherotesis$CIVIL)->X1
colnames(X1)<-c("casado","separado","soltero")

disjunctive(ficherotesis$NACIONALIDAD)->X2
colnames(X2)<-c("español","extranjero")

disjunctive(ficherotesis$TIPOTRABAJO)->X3
        colnames(X3)<-c("Técnico_mando_intermedio"," Obrero fijo."," Obrero
            temporal","Obrero fijo especializado","Obrero temporal especializado","
            Obrero fijo."," Autónomo","  Jubilado _rentista","  No activo")


disjunctive(ficherotesis$VIVIENDA)->X4
colnames(X4)<-c("libre","hipotecada","alquiler","domicilio_familia","otras_viviendas")
```

```
# Hemos creado también una variable que vale 1 en todas las partes (para comprobar la estimación del tamaño poblacional)

UNO=rep(1,dim(ficherotesis)[1])


# Construimos la matriz de equilibrio a partir de estas variables

X<-cbind(UNO,X1,X2,X3,X4,X5)


# Calculamos las probabilidades de inclusión.

# En este caso se trata de un m.a.s. con tamaño muestral de n=nB=167
# Por lo tanto, la prob. de inclusión de cada individuo es nB/nA; donde nA es el tamaño de la población A.

pik=rep(nB/nA,nA)


##############################################################################
## NOTA: Para modificar el tamaño muestral, cambiar el valor de nB ##
##############################################################################


# Selecionamos la muestra con la matriz de equilibrio X

#       Order=1; los datos son ordenados aleatoriamente
#       method=1; fase de aterrizaje mediante programación lineal

s=samplecube(X,pik,method=1)

muestra=cbind(ficherotesis,s)


# Una vez seleccionada la muestra, exportamos los datos a formato txt (o bien a formato csv) para poder leerlos en Excel
# Se debe modificar la dirección donde se quiere guardar el archivo.

write.table(muestra, "C:/ TESIS_DOCTORAL /muestra.txt", sep=";", col.names=TRUE, row.names=FALSE, quote=TRUE, na="NA")
```

ANEXO 2

CÓDIGO EN JAVA DE LA IMPLEMENTACIÓN DEL MODELO DE CREDIT SCORING.

ANEXO 2: CÓDIGO EN JAVA DE LA IMPLEMENTACIÓN DEL MODELO DE CREDIT SCORING.

ANEXO 2. Código en JAVA de la implementación del modelo de credit scoring.

Esta aplicación de credit scoring se ha desarrollado con NetBeans 7.01 y contiene tres clases diferenciadas siguiendo el patrón modelo, vista controlador que interactúan entre sí facilitando la comprensión del proceso.

TesisScoringApp.java es la clase donde se encuentra el método principal denominado main y que lanza la ejecución de la ventana (TesisScoringView) donde se introducen los datos del peticionario del crédito.

La segunda clase SCORER.JAVA es el modelo, que es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información. Envía a la vista aquella parte de la información que en cada momento se le solicita para que sea mostrada.

La tercera clase, Tesis_Scoring_View es la vista y tiene integrada la parte del controlador.

La vista presenta el modelo con el formato más adecuado para interactuar con el usuario.

El controlador es el que se encarga de gestionar los eventos invocando al modelo cuando se hace alguna petición en la ventana donde se registran los datos.

TesisScoringApp.java

```
 */
package tesisscoring;

import org.jdesktop.application.Application;
import org.jdesktop.application.SingleFrameApplication;

/**
 * The main class of the application.
 */
public class TesisScoringApp extends SingleFrameApplication {

    /**
     * At startup create and show the main frame of the application.
     */
    @Override protected void startup() {
        show(new TesisScoringView(this));
    }
```

```
    /**
     * This method is to initialize the specified window by injecting resources.
     * Windows shown in our application come fully initialized from the GUI
     * builder, so this additional configuration is not needed.
     */
    @Override protected void configureWindow(java.awt.Window root) {
    }

    /**
     * A convenient static getter for the application instance.
     * @return the instance of TesisScoringApp
     */
    public static TesisScoringApp getApplication() {
        return Application.getInstance(TesisScoringApp.class);
    }

    /**
     * Main method launching the application.
     */
    public static void main(String[] args) {
        launch(TesisScoringApp.class, args);
    }
}
```

SCORER.JAVA

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package tesisscoring;

import java.io.FileInputStream;
import java.io.ObjectInputStream;
import weka.classifiers.Classifier;
import weka.core.FastVector;
import weka.core.Instance;
import weka.core.Attribute;
import weka.core.Instances;

/**
 *
 * @author m
 */
public class Scorer {

    public Classifier clasificador;
    public Instances instancias;

    /*
    @attribute NUM_FAMILIA numeric
```

```
@attribute VIVIENDA {1,2,4,5,6}
@attribute IMPVALVIV numeric
@attribute NACIONALIDAD {1,2}
@attribute IMPINV numeric
@attribute IMPCUO numeric
@attribute INGRESOS numeric
@attribute SALDOMEDVINVI numeric
@attribute CIVIL {1,2,3}
@attribute TIPO_TRABAJO {1,3,4,5,6,7,8,9}
 * */
public void CargarDatos(
        float NUM_FAMILIA,
        float VIVIENDA,
        float IMPVALVIV,
        float NACIONALIDAD,
        float IMPINV,
        float IMPCUO,
        float INGRESOS,
        float SALDOMEDVINVI,
        float PORCENPRES,
        float CIVIL,
        float TIPO_TRABAJO) {

    // Creamos el vector de atributos
       FastVector atributos = new FastVector();

    //@attribute NUM_FAMILIA numeric
       atributos.addElement(new Attribute("NUM_FAMILIA"));

    //@attribute VIVIENDA {1,2,4,5,6}
       FastVector estadosVivienda = new FastVector();
       estadosVivienda.addElement("1");
       estadosVivienda.addElement("2");
       estadosVivienda.addElement("4");
       estadosVivienda.addElement("5");
       estadosVivienda.addElement("6");
        atributos.addElement(new Attribute("VIVIENDA",
  estadosVivienda));


    //@attribute IMPVALVIV numeric
       atributos.addElement(new Attribute("IMPVALVIV"));

    //@attribute NACIONALIDAD {1,2}
       FastVector estadosNacionalidad = new FastVector();
       estadosNacionalidad.addElement("1");
       estadosNacionalidad.addElement("2");
        atributos.addElement(new Attribute("NACIONALIDAD",
  estadosNacionalidad));


    //@attribute IMPINV numeric
       atributos.addElement(new Attribute("IMPINV"));
```

```
        //@attribute IMPCUO numeric
           atributos.addElement(new Attribute("IMPCUO"));

        //@attribute INGRESOS numeric
            atributos.addElement(new Attribute("INGRESOS"));


//@attribute SALDOMEDVINVI numeric
            atributos.addElement(new

Attribute("SALDOMEDVINVI"));

//@attribute PORCENPRES numeric
           atributos.addElement(new Attribute("PORCENPRES"));

        //@attribute CIVIL {1,2,3}
           FastVector estadosCivil = new FastVector();
           estadosCivil.addElement("1");
           estadosCivil.addElement("2");
           estadosCivil.addElement("3");
            atributos.addElement(new Attribute("CIVIL",
     estadosCivil));



        //@attribute TIPO_TRABAJO {1,3,4,6,7,8,9}
           FastVector estadosTipoTrabajo = new FastVector();
           estadosTipoTrabajo.addElement("1");
           estadosTipoTrabajo.addElement("3");
           estadosTipoTrabajo.addElement("4");
           estadosTipoTrabajo.addElement("5");
           estadosTipoTrabajo.addElement("6");
           estadosTipoTrabajo.addElement("7");
           estadosTipoTrabajo.addElement("8");
           estadosTipoTrabajo.addElement("9");
            atributos.addElement(new Attribute("TIPO_TRABAJO",
     estadosTipoTrabajo));



        //@attribute CLASE {SI,NO}
           FastVector estadosClase = new FastVector();
           estadosClase.addElement("SI");
           estadosClase.addElement("NO");
            atributos.addElement(new Attribute("CLASE",
     estadosClase));

        // Creamos el objeto de instancias
           this.instancias = new Instances("Datos", atributos,
0);
```

```
        double[] valores = new double[12];



        valores[0] = NUM_FAMILIA;
        valores[1] = VIVIENDA;
        valores[2] = IMPVALVIV;
        valores[3] = NACIONALIDAD;
        valores[4] = IMPINV;
        valores[5] = IMPCUO;
        valores[6] = INGRESOS;
        valores[7] = SALDOMEDVINVI;
        valores[8] = PORCENPRES;
        valores[9] = CIVIL;
        valores[10] = TIPO_TRABAJO;

        valores[11] = 0; // Variable de clase..

        this.instancias.add(new Instance(1.0, valores));
        this.instancias.setClassIndex(11);

        System.out.println(this.instancias);
    }

    public boolean CargarModelo(String rutaModelo) {
        try {
            ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(rutaModelo));
            this.clasificador = (Classifier) ois.readObject();
            ois.close();
        } catch (Exception e) {
            return false;
        }

        return true;
    }

    public int Procesar() {

        try {
            double valor =
this.clasificador.classifyInstance(this.instancias.firstInstance
());
            if (valor == 0) {
                return 0;
            } else {
                return 1;
            }
        } catch (Exception e) {
            return -1;
        }
    }
```

```
    public double[] GetProbabilidades() {
        try {
            return
this.clasificador.distributionForInstance(this.instancias.firstI
nstance());

        } catch (Exception e) {
            return new double[0];
        }
    }
}
```

TESIS_SCORING_VIEW. JAVA

```
/*
 * TesisScoringView.java
 */
package tesisscoring;

import org.jdesktop.application.Action;
import org.jdesktop.application.ResourceMap;
import org.jdesktop.application.SingleFrameApplication;
import org.jdesktop.application.FrameView;
import org.jdesktop.application.TaskMonitor;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.DecimalFormat;
import javax.swing.Timer;
import javax.swing.Icon;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JOptionPane;

/**
 * The application's main frame.
 */
public class TesisScoringView extends FrameView {

    public TesisScoringView(SingleFrameApplication app) {
        super(app);
        initComponents();
        this.getFrame().setTitle("APLICACION DE CREDIT SCORING
TESIS DOCTORAL M.BELTRAN ");
        CargarTextosCombobox();
        // status bar initialization - message timeout, idle
icon and busy animation, etc
        ResourceMap resourceMap = getResourceMap();
        int messageTimeout =
resourceMap.getInteger("StatusBar.messageTimeout");
        messageTimer = new Timer(messageTimeout, new
ActionListener() {
```

```
            public void actionPerformed(ActionEvent e) {
                statusMessageLabel.setText("");
            }
        });
        messageTimer.setRepeats(false);
        int busyAnimationRate =
resourceMap.getInteger("StatusBar.busyAnimationRate");
        for (int i = 0; i < busyIcons.length; i++) {
            busyIcons[i] =
resourceMap.getIcon("StatusBar.busyIcons[" + i + "]");
        }
        busyIconTimer = new Timer(busyAnimationRate, new
ActionListener() {

            public void actionPerformed(ActionEvent e) {
                busyIconIndex = (busyIconIndex + 1) %
busyIcons.length;

statusAnimationLabel.setIcon(busyIcons[busyIconIndex]);
            }
        });
        idleIcon = resourceMap.getIcon("StatusBar.idleIcon");
        statusAnimationLabel.setIcon(idleIcon);
        progressBar.setVisible(false);

        // connecting action tasks to status bar via TaskMonitor
        TaskMonitor taskMonitor = new
TaskMonitor(getApplication().getContext());
        taskMonitor.addPropertyChangeListener(new
java.beans.PropertyChangeListener() {

            public void
propertyChange(java.beans.PropertyChangeEvent evt) {
                String propertyName = evt.getPropertyName();
                if ("started".equals(propertyName)) {
                    if (!busyIconTimer.isRunning()) {

statusAnimationLabel.setIcon(busyIcons[0]);
                        busyIconIndex = 0;
                        busyIconTimer.start();
                    }
                    progressBar.setVisible(true);
                    progressBar.setIndeterminate(true);
                } else if ("done".equals(propertyName)) {
                    busyIconTimer.stop();
                    statusAnimationLabel.setIcon(idleIcon);
                    progressBar.setVisible(false);
                    progressBar.setValue(0);
                } else if ("message".equals(propertyName)) {
                    String text = (String) (evt.getNewValue());
                    statusMessageLabel.setText((text == null) ?
"" : text);
```

```
                    messageTimer.restart();
                } else if ("progress".equals(propertyName)) {
                    int value = (Integer) (evt.getNewValue());
                    progressBar.setVisible(true);
                    progressBar.setIndeterminate(false);
                    progressBar.setValue(value);
                }
            }
        });
    }

    @Action
    public void showAboutBox() {
        if (aboutBox == null) {
            JFrame mainFrame =
TesisScoringApp.getApplication().getMainFrame();
            aboutBox = new TesisScoringAboutBox(mainFrame);
            aboutBox.setLocationRelativeTo(mainFrame);
        }
        TesisScoringApp.getApplication().show(aboutBox);
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this
method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN:initComponents
    private void initComponents() {

        mainPanel = new javax.swing.JPanel();
        jTextFieldNUMCPMUNIFAM = new javax.swing.JTextField();
        jComboBoxEstadoCivil = new javax.swing.JComboBox();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jComboBoxNacionalidad = new javax.swing.JComboBox();
        jLabel8 = new javax.swing.JLabel();
        jComboBoxSituacionVivienda = new
javax.swing.JComboBox();
        jLabel9 = new javax.swing.JLabel();
        jComboBoxTipoTrabajo = new javax.swing.JComboBox();
        jLabel111 = new javax.swing.JLabel();
        jLabel13 = new javax.swing.JLabel();
        jTextFieldIMPVALVIV = new javax.swing.JTextField();
        jLabel15 = new javax.swing.JLabel();
        jTextFieldIMPPMO = new javax.swing.JTextField();
        jLabel17 = new javax.swing.JLabel();
        jTextFieldINGRESOS = new javax.swing.JTextField();
        jLabel20 = new javax.swing.JLabel();
```

```
        jTextFieldSALDOMDVINVI = new javax.swing.JTextField();
        jLabel21 = new javax.swing.JLabel();
        jTextFieldIMPCUO = new javax.swing.JTextField();
        jPanel1 = new javax.swing.JPanel();
        jLabel4 = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        jLabel5 = new javax.swing.JLabel();
        jPanel3 = new javax.swing.JPanel();
        jLabel6 = new javax.swing.JLabel();
        jTextFieldResultado = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTextPaneProbabilidades = new javax.swing.JTextPane();
        statusPanel = new javax.swing.JPanel();
        javax.swing.JSeparator statusPanelSeparator = new
javax.swing.JSeparator();
        statusMessageLabel = new javax.swing.JLabel();
        statusAnimationLabel = new javax.swing.JLabel();
        progressBar = new javax.swing.JProgressBar();
        jComboBoxRutaModelo = new javax.swing.JComboBox();
        jTextFieldPORCENPRES = new javax.swing.JTextField();
        jLabel22 = new javax.swing.JLabel();
        jFileChooser1 = new javax.swing.JFileChooser();

        mainPanel.setName("mainPanel"); // NOI18N

        org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(tesisscoring.Te
sisScoringApp.class).getContext().getResourceMap(TesisScoringVie
w.class);

jTextFieldNUMCPMUNIFAM.setText(resourceMap.getString("jTextField
NUMCPMUNIFAM.text")); // NOI18N

jTextFieldNUMCPMUNIFAM.setName("jTextFieldNUMCPMUNIFAM"); //
NOI18N

        jComboBoxEstadoCivil.setModel(new
javax.swing.DefaultComboBoxModel(new String[] { "Casado",
"Separado", "Soltero" }));
        jComboBoxEstadoCivil.setName("jComboBoxEstadoCivil"); //
NOI18N


jLabel1.setForeground(resourceMap.getColor("jLabel1.foreground")
); // NOI18N
        jLabel1.setText(resourceMap.getString("jLabel1.text"));
// NOI18N
        jLabel1.setName("jLabel1"); // NOI18N


jLabel2.setForeground(resourceMap.getColor("jLabel2.foreground")
); // NOI18N
```

```
        jLabel2.setText(resourceMap.getString("jLabel2.text"));
// NOI18N
        jLabel2.setName("jLabel2"); // NOI18N

        jLabel3.setText(resourceMap.getString("jLabel3.text"));
// NOI18N
        jLabel3.setName("jLabel3"); // NOI18N

        jComboBoxNacionalidad.setModel(new
javax.swing.DefaultComboBoxModel(new String[] { "EspaÃ±ol",
"Extranjero" }));
        jComboBoxNacionalidad.setName("jComboBoxNacionalidad");
// NOI18N


jLabel8.setForeground(resourceMap.getColor("jLabel8.foreground")
); // NOI18N
        jLabel8.setText(resourceMap.getString("jLabel8.text"));
// NOI18N
        jLabel8.setName("jLabel8"); // NOI18N

        jComboBoxSituacionVivienda.setModel(new
javax.swing.DefaultComboBoxModel(new String[] { "Libre",
"Hipotecada", "Alquiler", "Domicilio padres", "Otros" }));

jComboBoxSituacionVivienda.setName("jComboBoxSituacionVivienda")
; // NOI18N


jLabel9.setForeground(resourceMap.getColor("jLabel9.foreground")
); // NOI18N
        jLabel9.setText(resourceMap.getString("jLabel9.text"));
// NOI18N
        jLabel9.setName("jLabel9"); // NOI18N

        jComboBoxTipoTrabajo.setModel(new
javax.swing.DefaultComboBoxModel(new String[] { "TÃ©cnico-Mando
intermedio", "Obrero fijo", "Obrero temporal", "Obrero fijo
especializado", "Obrero temporal especializado", "AutÃ³nomo",
"Jubilado rentista", "No activo" }));
        jComboBoxTipoTrabajo.setName("jComboBoxTipoTrabajo"); //
NOI18N


jLabel111.setForeground(resourceMap.getColor("jLabel111.foregrou
nd")); // NOI18N

jLabel111.setText(resourceMap.getString("jLabel111.text")); //
NOI18N
        jLabel111.setName("jLabel111"); // NOI18N
```

```
jLabel13.setForeground(resourceMap.getColor("jLabel131.foregroun
d")); // NOI18N

jLabel13.setText(resourceMap.getString("jLabel131.text")); //
NOI18N
        jLabel13.setName("jLabel131"); // NOI18N


jTextFieldIMPVALVIV.setText(resourceMap.getString("jTextFieldIMP
VALVIV.text")); // NOI18N
        jTextFieldIMPVALVIV.setName("jTextFieldIMPVALVIV"); //
NOI18N


jLabel15.setForeground(resourceMap.getColor("jLabel151.foregroun
d")); // NOI18N

jLabel15.setText(resourceMap.getString("jLabel151.text")); //
NOI18N
        jLabel15.setName("jLabel151"); // NOI18N


jTextFieldIMPPMO.setText(resourceMap.getString("jTextFieldIMPPMO
.text")); // NOI18N
        jTextFieldIMPPMO.setName("jTextFieldIMPPMO"); // NOI18N


jLabel17.setForeground(resourceMap.getColor("jLabel171.foregroun
d")); // NOI18N

jLabel17.setText(resourceMap.getString("jLabel171.text")); //
NOI18N
        jLabel17.setName("jLabel171"); // NOI18N


jTextFieldINGRESOS.setText(resourceMap.getString("jTextFieldINGR
ESOS.text")); // NOI18N
        jTextFieldINGRESOS.setName("jTextFieldINGRESOS"); //
NOI18N


jLabel20.setForeground(resourceMap.getColor("jLabel201.foregroun
d")); // NOI18N

jLabel20.setText(resourceMap.getString("jLabel201.text")); //
NOI18N
        jLabel20.setName("jLabel201"); // NOI18N


jTextFieldSALDOMDVINVI.setText(resourceMap.getString("jTextField
SALDOMDVINVI.text")); // NOI18N
```

```
jTextFieldSALDOMDVINVI.setName("jTextFieldSALDOMDVINVI"); //
NOI18N


jLabel21.setForeground(resourceMap.getColor("jLabel211.foregroun
d")); // NOI18N

jLabel21.setText(resourceMap.getString("jLabel211.text")); //
NOI18N
        jLabel21.setName("jLabel211"); // NOI18N


jTextFieldIMPCUO.setText(resourceMap.getString("jTextFieldIMPCUO
.text")); // NOI18N
        jTextFieldIMPCUO.setName("jTextFieldIMPCUO"); // NOI18N


jPanel1.setBackground(resourceMap.getColor("jPanel1.background")
); // NOI18N

jPanel1.setBorder(javax.swing.BorderFactory.createEtchedBorder()
);
        jPanel1.setName("jPanel1"); // NOI18N


jLabel4.setBackground(resourceMap.getColor("jLabel4.background")
); // NOI18N
        jLabel4.setFont(resourceMap.getFont("jLabel4.font")); //
NOI18N
        jLabel4.setText(resourceMap.getString("jLabel4.text"));
// NOI18N
        jLabel4.setName("jLabel4"); // NOI18N

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel4,
javax.swing.GroupLayout.PREFERRED_SIZE, 167,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(482, Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
```

```
                .addContainerGap()
                .addComponent(jLabel4)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );


jPanel2.setBackground(resourceMap.getColor("jPanel2.background")
); // NOI18N

jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder()
);
        jPanel2.setName("jPanel2"); // NOI18N

        jLabel5.setFont(resourceMap.getFont("jLabel5.font")); //
NOI18N
        jLabel5.setText(resourceMap.getString("jLabel5.text"));
// NOI18N
        jLabel5.setName("jLabel5"); // NOI18N

        javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel5)
                .addContainerGap(510, Short.MAX_VALUE))
        );
        jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                .addComponent(jLabel5)
                .addContainerGap())
        );


jPanel3.setBackground(resourceMap.getColor("jPanel3.background")
); // NOI18N

jPanel3.setBorder(javax.swing.BorderFactory.createEtchedBorder()
);
```

```
        jPanel3.setName("jPanel3"); // NOI18N

        jLabel6.setText(resourceMap.getString("jLabel6.text"));
// NOI18N
        jLabel6.setName("jLabel6"); // NOI18N


jTextFieldResultado.setText(resourceMap.getString("jTextFieldRes
ultado.text")); // NOI18N
        jTextFieldResultado.setName("jTextFieldResultado"); //
NOI18N


jButton1.setText(resourceMap.getString("jButton1.text")); //
NOI18N
        jButton1.setName("jButton1"); // NOI18N
        jButton1.addMouseListener(new
java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent
evt) {
                jButton1MouseClicked(evt);
            }
        });

        jScrollPane1.setName("jScrollPane1"); // NOI18N


jTextPaneProbabilidades.setName("jTextPaneProbabilidades"); //
NOI18N
        jScrollPane1.setViewportView(jTextPaneProbabilidades);

        javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLay
out.Alignment.TRAILING)
                    .addComponent(jScrollPane1,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 486, Short.MAX_VALUE)
```

```
                    .addComponent(jTextFieldResultado,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 486, Short.MAX_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)
                .addComponent(jButton1)
                .addContainerGap())
        );
        jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addGap(42, 42, 42)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLay
out.Alignment.BASELINE)
                    .addComponent(jButton1)
                    .addComponent(jTextFieldResultado,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel6))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(67, Short.MAX_VALUE))
        );

        statusPanel.setName("statusPanel"); // NOI18N

        statusPanelSeparator.setName("statusPanelSeparator"); //
NOI18N

        statusMessageLabel.setName("statusMessageLabel"); //
NOI18N


statusAnimationLabel.setHorizontalAlignment(javax.swing.SwingCon
stants.LEFT);
        statusAnimationLabel.setName("statusAnimationLabel"); //
NOI18N

        progressBar.setName("progressBar"); // NOI18N

        javax.swing.GroupLayout statusPanelLayout = new
javax.swing.GroupLayout(statusPanel);
        statusPanel.setLayout(statusPanelLayout);
        statusPanelLayout.setHorizontalGroup(
```

```
statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Al
ignment.LEADING)
            .addGroup(statusPanelLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(statusMessageLabel)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED, 493, Short.MAX_VALUE)
                .addComponent(progressBar,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED)
                .addComponent(statusAnimationLabel)
                .addContainerGap())
            .addComponent(statusPanelSeparator,
javax.swing.GroupLayout.DEFAULT_SIZE, 663, Short.MAX_VALUE)
        );
        statusPanelLayout.setVerticalGroup(

statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Al
ignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
statusPanelLayout.createSequentialGroup()
                .addComponent(statusPanelSeparator,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED, 8, Short.MAX_VALUE)

.addGroup(statusPanelLayout.createParallelGroup(javax.swing.Grou
pLayout.Alignment.BASELINE)
                    .addComponent(statusMessageLabel)
                    .addComponent(statusAnimationLabel)
                    .addComponent(progressBar,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(3, 3, 3))
        );

        jComboBoxRutaModelo.setModel(new
javax.swing.DefaultComboBoxModel(new String[] { "arboles de
decision", "regresion logistica", "redes bayesianas" }));
        jComboBoxRutaModelo.setName("jComboBoxRutaModelo"); //
NOI18N
```

```
        jTextFieldPORCENPRES.setName("jTextFieldPORCENPRES"); //
NOI18N


jLabel22.setForeground(resourceMap.getColor("jLabel22.foreground
")); // NOI18N

jLabel22.setText(resourceMap.getString("jLabel22.text")); //
NOI18N
        jLabel22.setName("jLabel22"); // NOI18N

        javax.swing.GroupLayout mainPanelLayout = new
javax.swing.GroupLayout(mainPanel);
        mainPanel.setLayout(mainPanelLayout);
        mainPanelLayout.setHorizontalGroup(

mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alig
nment.LEADING)
            .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jPanel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(statusPanel,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGroup(mainPanelLayout.createSequentialGroup()

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.LEADING)

.addGroup(mainPanelLayout.createSequentialGroup()
                        .addGap(12, 12, 12)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.TRAILING, false)
                            .addComponent(jLabel8,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jLabel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jLabel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jLabel9,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))

.addGroup(mainPanelLayout.createSequentialGroup()
                        .addContainerGap()
```

```
                        .addComponent(jLabel111)))
                .addGap(18, 18, 18)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.TRAILING, false)

.addGroup(mainPanelLayout.createSequentialGroup()

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.LEADING, false)

.addComponent(jComboBoxSituacionVivienda, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jComboBoxNacionalidad,
0, 155, Short.MAX_VALUE)
                            .addComponent(jComboBoxEstadoCivil,
javax.swing.GroupLayout.Alignment.TRAILING, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jComboBoxTipoTrabajo,
0, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                        .addGap(18, 18, 18)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.LEADING)
                            .addComponent(jLabel21,
javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel15)
                            .addComponent(jLabel13)
                            .addComponent(jLabel20)
                            .addComponent(jLabel17))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED))

.addGroup(mainPanelLayout.createSequentialGroup()
                        .addComponent(jTextFieldNUMCPMUNIFAM)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel22)))

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.LEADING)
                    .addComponent(jTextFieldSALDOMDVINVI,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 247, Short.MAX_VALUE)
                    .addComponent(jTextFieldIMPCUO,
javax.swing.GroupLayout.DEFAULT_SIZE, 247, Short.MAX_VALUE)
                    .addComponent(jTextFieldIMPPMO,
javax.swing.GroupLayout.DEFAULT_SIZE, 247, Short.MAX_VALUE)
                    .addComponent(jTextFieldIMPVALVIV,
javax.swing.GroupLayout.DEFAULT_SIZE, 247, Short.MAX_VALUE)
```

```
                    .addComponent(jTextFieldINGRESOS,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 247, Short.MAX_VALUE)
                    .addComponent(jTextFieldPORCENPRES,
javax.swing.GroupLayout.DEFAULT_SIZE, 247, Short.MAX_VALUE))
                .addContainerGap())

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainPanelLayout.createSequentialGroup()
                .addContainerGap()

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.TRAILING)
                    .addComponent(jPanel3,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING,
mainPanelLayout.createSequentialGroup()
                        .addComponent(jLabel3)
                        .addGap(18, 18, 18)
                        .addComponent(jComboBoxRutaModelo, 0,
535, Short.MAX_VALUE)))
                .addGap(2, 2, 2))
        );
        mainPanelLayout.setVerticalGroup(

mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alig
nment.LEADING)
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(15, 15, 15)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.LEADING)

.addGroup(mainPanelLayout.createSequentialGroup()
                        .addGap(104, 104, 104)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.BASELINE)
                            .addComponent(jComboBoxTipoTrabajo,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel9)))

.addGroup(mainPanelLayout.createSequentialGroup()
```

```
.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.BASELINE)
                            .addComponent(jComboBoxEstadoCivil,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel1)
                            .addComponent(jTextFieldIMPVALVIV,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel13))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.BASELINE)
                            .addComponent(jLabel2)
                            .addComponent(jComboBoxNacionalidad,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jTextFieldIMPPMO,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel15))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.BASELINE)

.addComponent(jComboBoxSituacionVivienda,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel8)
                            .addComponent(jTextFieldIMPCUO,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel21))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.BASELINE)
```

```
                                .addComponent(jTextFieldINGRESOS,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addComponent(jLabel17))
                            .addGap(18, 18, 18)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.BASELINE)

.addComponent(jTextFieldSALDOMDVINVI,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addComponent(jLabel20))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.BASELINE)
                                .addComponent(jTextFieldPORCENPRES,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addComponent(jLabel22)

.addComponent(jTextFieldNUMCPMUNIFAM,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addComponent(jLabel111))))
                    .addGap(53, 53, 53)
                    .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.BASELINE)
                        .addComponent(jComboBoxRutaModelo,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)
                    .addComponent(jPanel3,
javax.swing.GroupLayout.PREFERRED_SIZE,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED, 59, Short.MAX_VALUE)
                .addComponent(statusPanel,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        );

        jFileChooser1.setName("jFileChooser1"); // NOI18N

        setComponent(mainPanel);
        setStatusBar(statusPanel);
    }// </editor-fold>//GEN-END:initComponents

private void jButton1MouseClicked(java.awt.event.MouseEvent evt)
{//GEN-FIRST:event_jButton1MouseClicked
        // TODO add your handling code here:

        try {
            // Variables numericas
            float NUM_FAMILIA =
Float.parseFloat(jTextFieldNUMCPMUNIFAM.getText());
            float VIVIENDA = (float)
jComboBoxSituacionVivienda.getSelectedIndex();
            float IMPVALVIV =
Float.parseFloat(jTextFieldIMPVALVIV.getText());
            float NACIONALIDAD = (float)
jComboBoxNacionalidad.getSelectedIndex();
            float IMPINV =
Float.parseFloat(jTextFieldIMPPMO.getText());
            float IMPCUO =
Float.parseFloat(jTextFieldIMPCUO.getText());
            float INGRESOS =
Float.parseFloat(jTextFieldINGRESOS.getText());
            float SALDOMEDVINVI =
Float.parseFloat(jTextFieldSALDOMDVINVI.getText());
            float PORCENPRES =
Float.parseFloat(jTextFieldPORCENPRES.getText());
            float CIVIL = (float)
jComboBoxEstadoCivil.getSelectedIndex();
            float TIPO_TRABAJO = (float)
jComboBoxTipoTrabajo.getSelectedIndex();



            Scorer scorer = new Scorer();
            scorer.CargarDatos(
                    NUM_FAMILIA,
                    VIVIENDA,
                    IMPVALVIV,
```

```
                    NACIONALIDAD,
                    IMPINV,
                    IMPCUO,
                    INGRESOS,
                    SALDOMEDVINVI,
                    PORCENPRES,
                    CIVIL,
                    TIPO_TRABAJO);

            int numero = jComboBoxRutaModelo.getSelectedIndex();
            boolean modeloOK = true;
            switch (numero) {
                case 0:
                    modeloOK =
scorer.CargarModelo(".\\Arbol_decision_C_4_5.model");
                    break;
                case 1:
                    modeloOK =
scorer.CargarModelo(".\\NaÃ¯ve_bayes.model");
                    break;
                case 2:
                    modeloOK =
scorer.CargarModelo(".\\Red_Bayesiana_HC_2_Padres.model");
                    break;
                case 3:
                    modeloOK =
scorer.CargarModelo(".\\Red_Neuronal.model");
                    break;
                case 4:
                    modeloOK =
scorer.CargarModelo(".\\Regresion_Logistica.model");
                    break;
                case 5:
                    modeloOK =
scorer.CargarModelo(".\\Maquinas_Vectores_Soporte.model");
                    break;
                case 6:
                    modeloOK =
scorer.CargarModelo(".\\Multiclasificador_Boosting.model");
                    break;
                case 7:
                    modeloOK =
scorer.CargarModelo(".\\Multiclasificador_Random_SubSpace.model"
);
                    break;
                default:
                    JOptionPane.showMessageDialog(null,
"error");
                    break;
            }

//
//
```

```
//            //joptionaqui muestra un mensage con el texto de
la ruta.
//
JOptionPane.showMessageDialog(null,".\\"+jComboBoxRutaModelo.get
SelectedItem().toString()+".model" );
//              boolean modeloOK =
scorer.CargarModelo(".\\"+jComboBoxRutaModelo.getSelectedItem().
toString()+".model");
////            boolean modeloOK =
scorer.CargarModelo(jTextFieldRutaModelo.getText());

            if (modeloOK) {
                int resultado = scorer.Procesar();
                if (resultado == -1) {
                    jTextFieldResultado.setText("Error en la
clasificacion");
                } else if (resultado == 0) {
                    jTextFieldResultado.setText("APROBADO");
                } else {
                    jTextFieldResultado.setText("SUSPENDIDO");
                }
                DecimalFormat df = new DecimalFormat("0.0000");
                double[] probabilidades =
scorer.GetProbabilidades();
                String texto = "Probabilidad de aprobado: " +
df.format(probabilidades[0]) + "\n";
                texto += "Probabilidad de suspenso: " +
df.format(probabilidades[1]);

                jTextPaneProbabilidades.setText(texto);


            } else {
                jTextFieldResultado.setText("Error al cargar el
modelo.");
            }
        } catch (Exception e) {
            jTextFieldResultado.setText("Error en datos de
entrada");
            return;
        }
}//GEN-LAST:event_jButton1MouseClicked

    // Variables declaration - do not modify//GEN-
BEGIN:variables
    private javax.swing.JButton jButton1;
    private javax.swing.JComboBox jComboBoxEstadoCivil;
    private javax.swing.JComboBox jComboBoxNacionalidad;
    private javax.swing.JComboBox jComboBoxRutaModelo;
    private javax.swing.JComboBox jComboBoxSituacionVivienda;
    private javax.swing.JComboBox jComboBoxTipoTrabajo;
    private javax.swing.JFileChooser jFileChooser1;
    private javax.swing.JLabel jLabel1;
```

```
    private javax.swing.JLabel jLabel111;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel17;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel20;
    private javax.swing.JLabel jLabel21;
    private javax.swing.JLabel jLabel22;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextField jTextFieldIMPCUO;
    private javax.swing.JTextField jTextFieldIMPPMO;
    private javax.swing.JTextField jTextFieldIMPVALVIV;
    private javax.swing.JTextField jTextFieldINGRESOS;
    private javax.swing.JTextField jTextFieldNUMCPMUNIFAM;
    private javax.swing.JTextField jTextFieldPORCENPRES;
    private javax.swing.JTextField jTextFieldResultado;
    private javax.swing.JTextField jTextFieldSALDOMDVINVI;
    private javax.swing.JTextPane jTextPaneProbabilidades;
    private javax.swing.JPanel mainPanel;
    private javax.swing.JProgressBar progressBar;
    private javax.swing.JLabel statusAnimationLabel;
    private javax.swing.JLabel statusMessageLabel;
    private javax.swing.JPanel statusPanel;
    // End of variables declaration//GEN-END:variables
    private final Timer messageTimer;
    private final Timer busyIconTimer;
    private final Icon idleIcon;
    private final Icon[] busyIcons = new Icon[15];
    private int busyIconIndex = 0;
    private JDialog aboutBox;

    private void CargarTextosCombobox() {
        jComboBoxRutaModelo.removeAllItems();
        jComboBoxRutaModelo.addItem("Ãrboles de decisiÃ³n.
C4.5");
        jComboBoxRutaModelo.addItem("NaÃ¯ve Bayes");
        jComboBoxRutaModelo.addItem("Red bayesiana. HC con dos
padres");
        jComboBoxRutaModelo.addItem("Red neuronal");
        jComboBoxRutaModelo.addItem("RegresiÃ³n Logistica");
        jComboBoxRutaModelo.addItem("Maquinas de Vectores
Soporte");
        jComboBoxRutaModelo.addItem("Multiclasificador
Boosting");
```

```
        jComboBoxRutaModelo.addItem("Multiclasificador Random
Subspace");
    }
}
```

ANEXO 3

ANÁLISIS DESCRIPTIVO DE LAS VARIABLES CUANTITATIVAS DE LA BASE DE DATOS UTILIZADA.

ANEXO 3: ANÁLISIS DESCRIPTIVO DE LAS VARIABLES CUANTITATIVAS DE LA BASE DE DATOS UTILIZADA.

ANEXO 3. Análisis descriptivo de las variables cuantitativas de la Base de Datos utilizada.

Las figuras y tablas que se encuentran en este Anexo complementan el análisis descriptivo del capítulo 5.1. A continuación se muestran las siguientes tablas y figuras: estadísticos descriptivos de la variable, estadísticos robustos, percentiles, valores extremos, el histograma y el gráfico Q-Q.

Tabla A.3.1. Estadísticos de la variable Miembros de la familia.

Descriptivos

			Estadístico	Error típ.
Nº miembros familiares	Media		2,48	,031
	Intervalo de confianza para la media al 95%	Límite inferior	2,42	
		Límite superior	2,54	
	Media recortada al 5%		2,40	
	Mediana		2,00	
	Varianza		1,746	
	Desv. típ.		1,321	
	Mínimo		1	
	Máximo		10	
	Rango		9	
	Amplitud intercuartil		3	
	Asimetría		,597	,058
	Curtosis		,251	,116

Tabla A.3.2. Estadísticos robustos de la variable Miembros de la familia.

Estimadores-M

	Estimador-M de Huber[a]	Biponderado de Tukey[b]	Estimador-M de Hampel[c]	Onda de Andrews[d]
Nº miembros familiares	2,32	2,37	2,42	2,37

a. La constante de ponderación es 1,339.

b. La constante de ponderación es 4,685.

c. Las constantes de ponderación son 1,700, 3,400 y 8,500.

d. La constante de ponderación es 1,340*pi.

Tabla A.3.3. Percentiles de la variable Miembros de la familia.

Percentiles

		Percentiles						
		5	10	25	50	75	90	95
Promedio ponderado (definición 1)	Nº miembros familiares	1,00	1,00	1,00	2,00	4,00	4,00	5,00
Bisagras de Tukey	Nº miembros familiares			1,00	2,00	4,00		

Tabla A.3.4. Valores extremos de la variable Miembros de la familia.

Valores extremos

			Número del caso	Valor
Nº miembros familiares	Mayores	1	712	10
		2	355	9
		3	606	9
		4	1000	8
		5	11	6[a]
	Menores	1	1782	1
		2	1781	1
		3	1775	1
		4	1774	1
		5	1773	1[b]

a. En la tabla de valores extremos mayores sólo se muestra una lista parcial de los casos con el valor 6.

b. En la tabla de valores extremos menores sólo se muestra una lista parcial de los casos con el valor 1.

Figura A.3.1. Histograma de la variable Miembros de la familia.

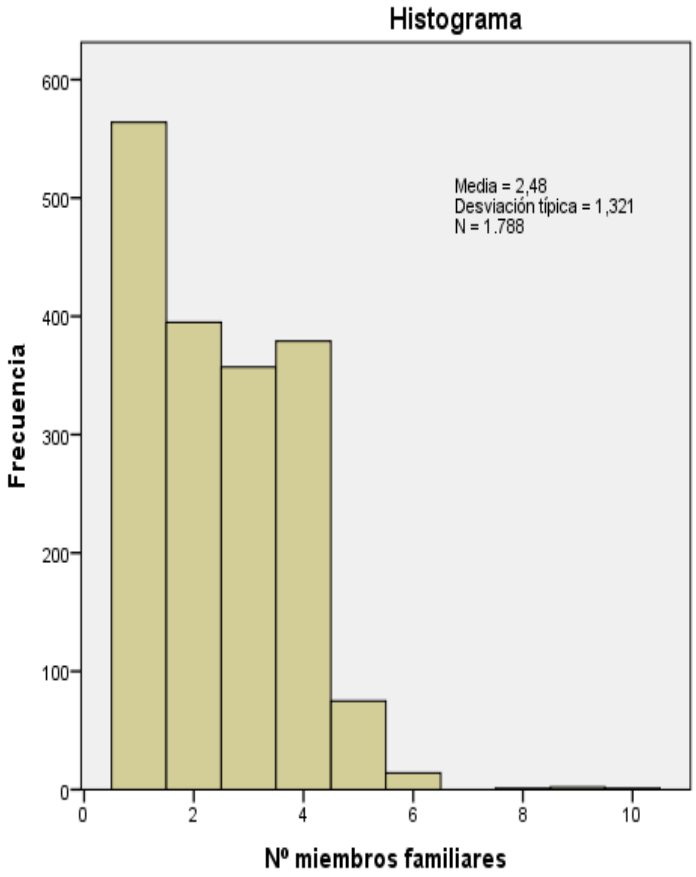


Figura A.3.2. Gráfico Q-Q normal de la variable Miembros de la familia.

.

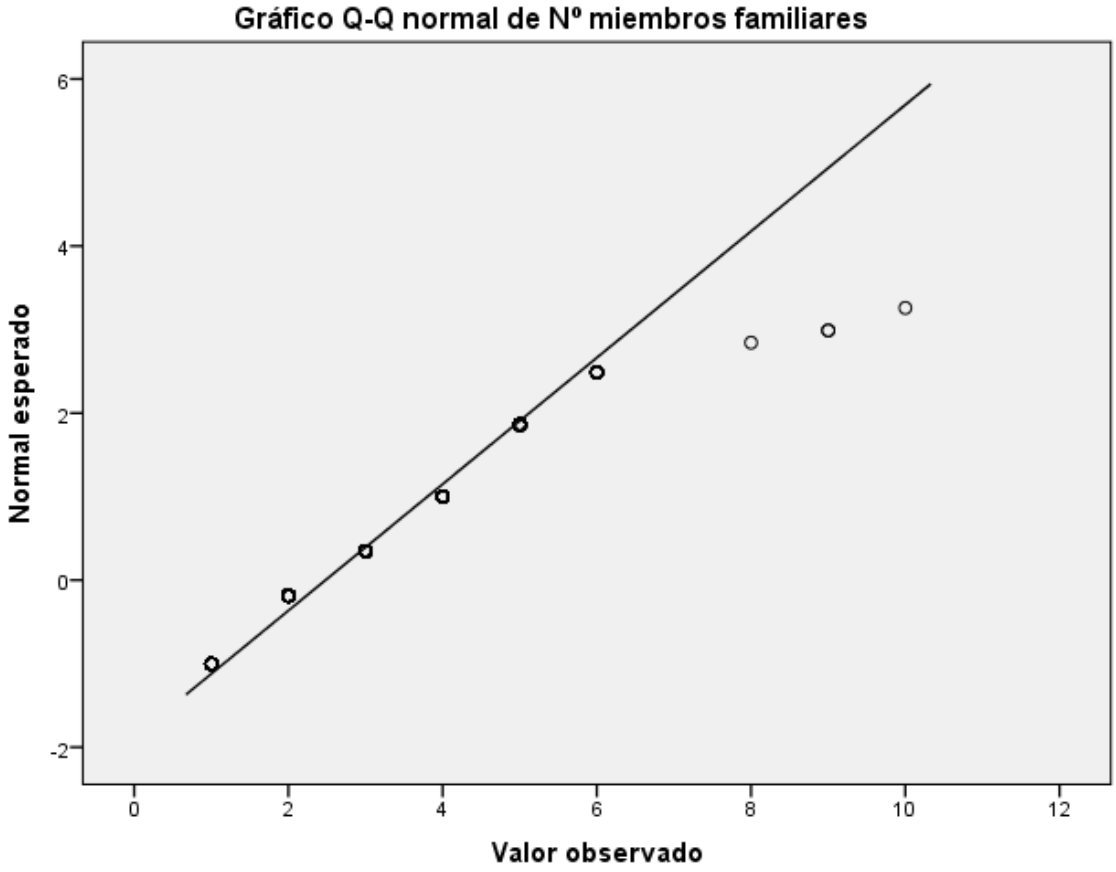


Tabla A.3.5. Estadísticos de la variable Valor de la vivienda.

Descriptivos

			Estadístico	Error típ.
Valor de la vivienda	Media		93647,35	2571,247
	Intervalo de confianza para la media al 95%	Límite inferior	88604,38	
		Límite superior	98690,31	
	Media recortada al 5%		82328,82	
	Mediana		75000,00	
	Varianza		11821022820	
	Desv. típ.		108724,527	
	Mínimo		0	
	Máximo		1280000	
	Rango		1280000	
	Amplitud intercuartil		150000	
	Asimetría		2,347	,058
	Curtosis		13,379	,116

Tabla A.3.6. Estadísticos robustos de la variable Valor de la vivienda.

Estimadores-M

	Estimador-M de Huber[a]	Biponderado de Tukey[b]	Estimador-M de Hampel[c]	Onda de Andrews[d]
Valor de la vivienda	77657,24	75088,46	81165,59	75045,50

a. La constante de ponderación es 1,339.

b. La constante de ponderación es 4,685.

c. Las constantes de ponderación son 1,700, 3,400 y 8,500.

d. La constante de ponderación es 1,340*pi.

Tabla A.3.7. Percentiles de la variable Valor de la vivienda.

Percentiles

		Percentiles						
		5	10	25	50	75	90	95
Promedio ponderado (definición 1)	Valor de la vivienda	,00	,00	,00	75000,00	150000,00	240000,00	300000,00
Bisagras de Tukey	Valor de la vivienda			,00	75000,00	150000,00		

Tabla A.3.8. Valores extremos de la variable Valor de la vivienda.

Valores extremos

			Número del caso	Valor
Valor de la vivienda	Mayores	1	1356	1280000
		2	1392	945000
		3	678	900000
		4	1478	768000
		5	281	600000[a]
	Menores	1	1788	0
		2	1782	0
		3	1781	0
		4	1779	0
		5	1776	0[b]

a. En la tabla de valores extremos mayores sólo se muestra una lista parcial de los casos con el valor 600000.

b. En la tabla de valores extremos menores sólo se muestra una lista parcial de los casos con el valor 0.

Figura A.3.3. Histograma de la variable Valor de la vivienda.

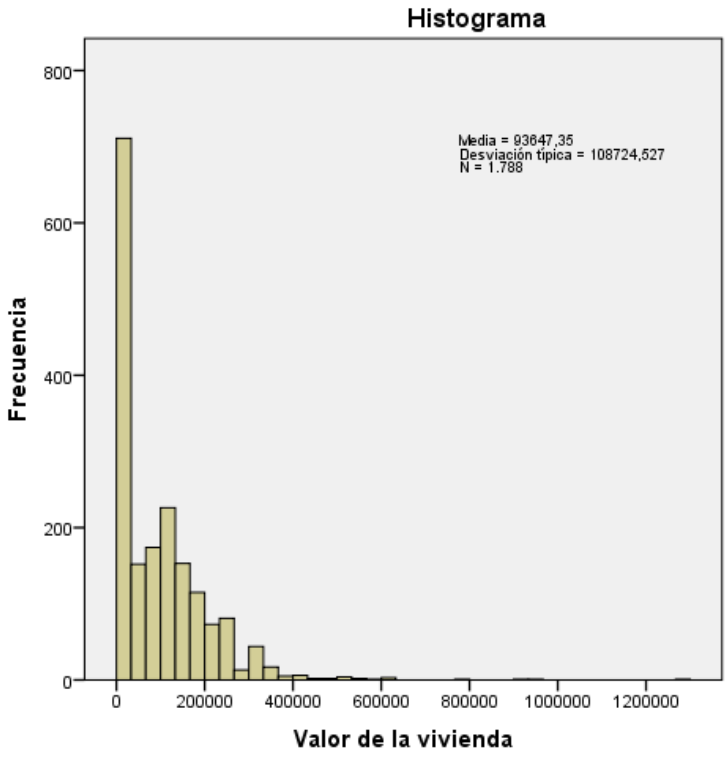
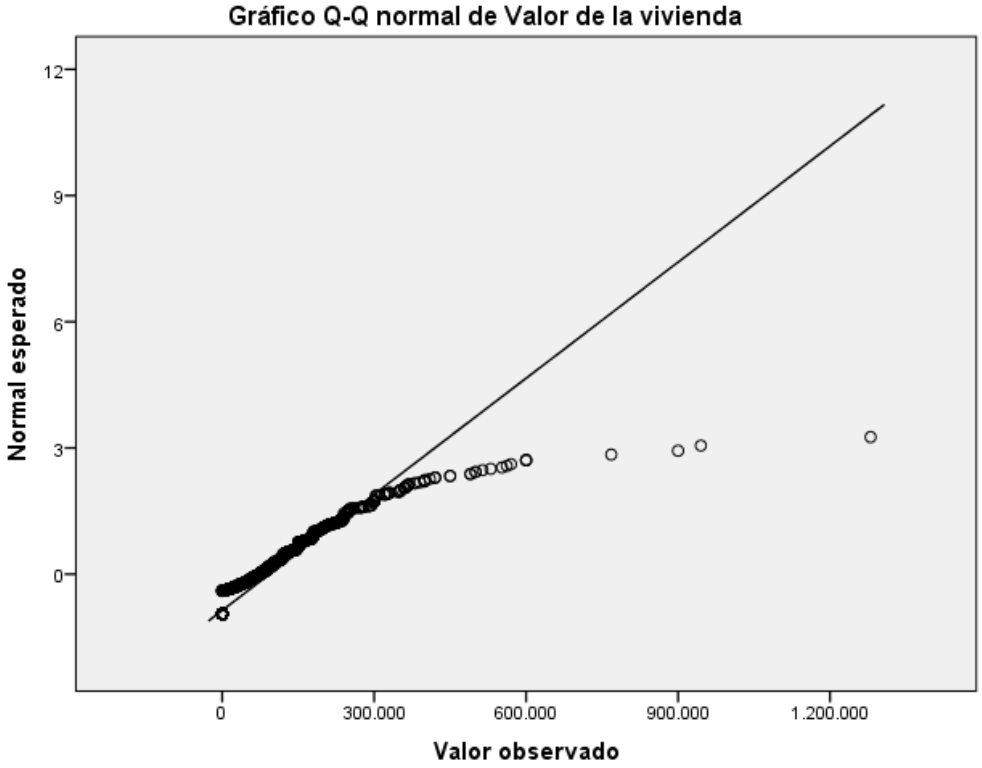


Figura A.3.4. Gráfico Q-Q normal de la variable Valor de la vivienda.



ANEXO 3: ANÁLISIS DESCRIPTIVO DE LAS VARIABLES CUANTITATIVAS DE LA BASE DE DATOS UTILIZADA.

Tabla A.3.9. Estadísticos de la variable Valor del patrimonio.

Descriptivos

			Estadístico	Error típ.
Importe del patrimonio	Media		9646,86	961,795
	Intervalo de confianza para la media al 95%	Límite inferior	7760,50	
		Límite superior	11533,23	
	Media recortada al 5%		2366,52	
	Mediana		,00	
	Varianza		1653989587	
	Desv. típ.		40669,271	
	Mínimo		0	
	Máximo		629214	
	Rango		629214	
	Amplitud intercuartil		0	
	Asimetría		7,370	,058
	Curtosis		71,174	,116

Tabla A.3.10. Estadísticos robustos de la variable Valor del patrimonio.

Estimadores-M[a]

	Estimador-M de Huber[b]	Biponderado de Tukey[c]	Estimador-M de Hampel[d]	Onda de Andrews[e]
Importe del patrimonio	.	.	.	.

a. No se pueden calcular algunos estimadores-M debido a que la distribución se centra sobre todo en la mediana.

b. La constante de ponderación es 1,339.

c. La constante de ponderación es 4,685.

d. Las constantes de ponderación son 1,700, 3,400 y 8,500.

e. La constante de ponderación es 1,340*pi.

Tabla A.3.11. Percentiles de la variable Valor del patrimonio.

Percentiles

		Percentiles						
		5	10	25	50	75	90	95
Promedio ponderado (definición 1)	Importe del patrimonio	,00	,00	,00	,00	,00	18000,00	52100,00
Bisagras de Tukey	Importe del patrimonio			,00	,00	,00		

Tabla A.3.12. Valores extremos de la variable Valor del patrimonio.

Valores extremos

			Número del caso	Valor
Importe del patrimonio	Mayores	1	1271	629214
		2	1546	480000
		3	1571	450000
		4	281	400000
		5	603	400000
	Menores	1	1788	0
		2	1787	0
		3	1786	0
		4	1785	0
		5	1784	0[a]

a. En la tabla de valores extremos menores sólo se muestra una lista parcial de los casos con el valor 0.

Figura A.3.5. Histograma de la variable Importe del patrimonio.

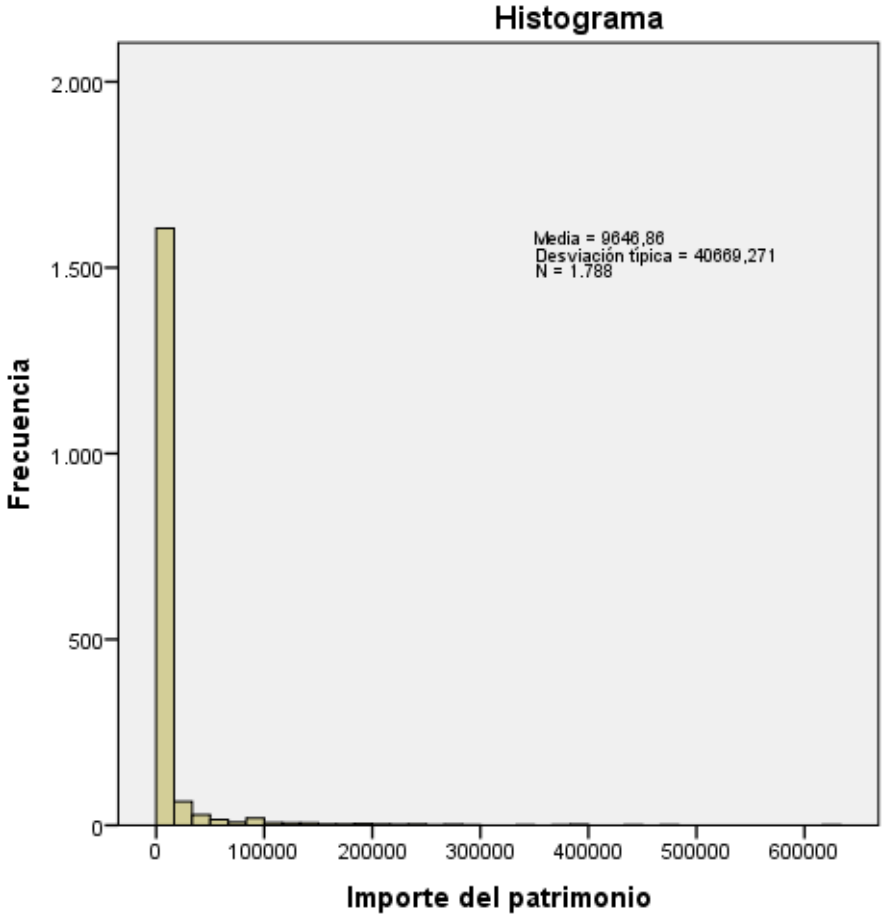


Figura A.3.6. Gráfico Q-Q normal de la variable Importe del patrimonio.

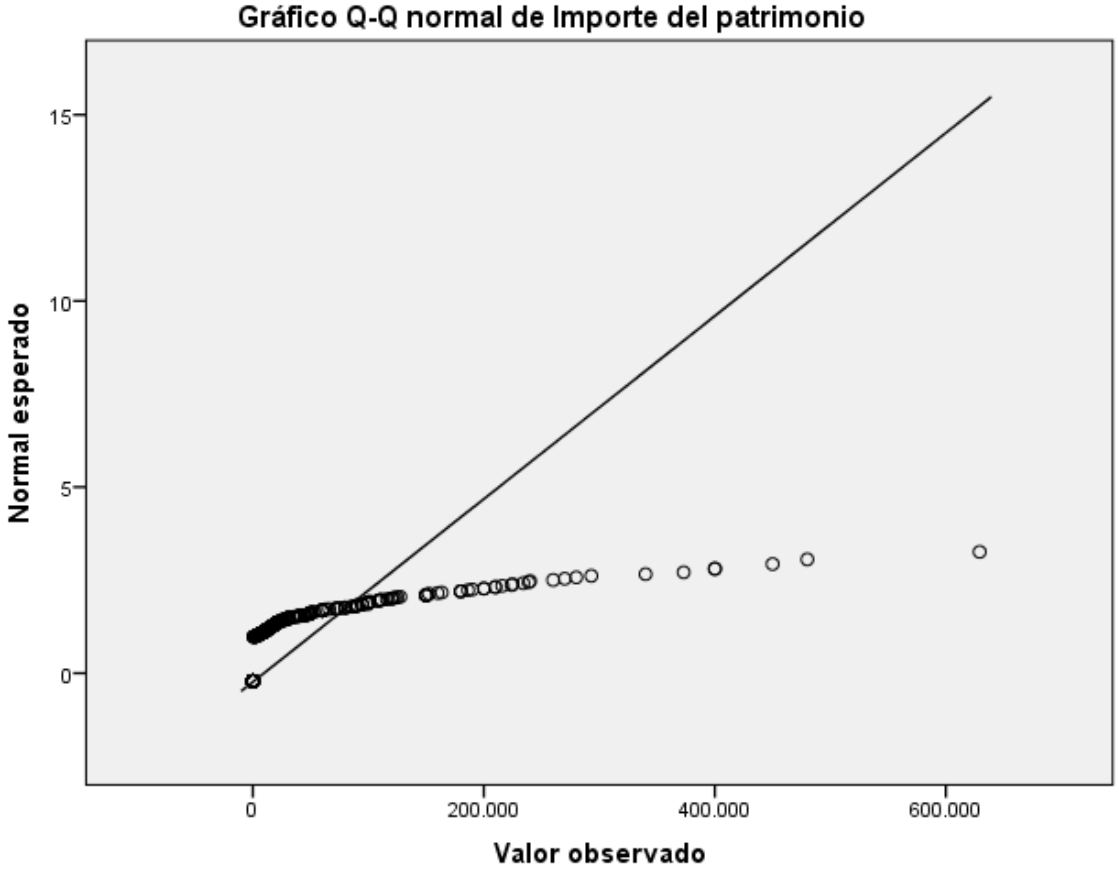


Tabla A.3.13. Estadísticos de la variable Importe del préstamo.

Descriptivos

			Estadístico	Error típ.
Importe del préstamo	Media		8395,88	157,233
	Intervalo de confianza para la media al 95%	Límite inferior	8087,50	
		Límite superior	8704,26	
	Media recortada al 5%		7808,52	
	Mediana		8000,00	
	Varianza		44203428,01	
	Desv. típ.		6648,566	
	Mínimo		285	
	Máximo		70000	
	Rango		69715	
	Amplitud intercuartil		8000	
	Asimetría		1,633	,058
	Curtosis		6,038	,116

Tabla A.3.14. Estadísticos robustos de la variable Importe del préstamo.

Estimadores-M

	Estimador-M de Huber[a]	Biponderado de Tukey[b]	Estimador-M de Hampel[c]	Onda de Andrews[d]
Importe del préstamo	7357,23	7006,95	7332,17	7001,89

a. La constante de ponderación es 1,339.

b. La constante de ponderación es 4,685.

c. Las constantes de ponderación son 1,700, 3,400 y 8,500.

d. La constante de ponderación es 1,340*pi.

Tabla A.3.15. Percentiles de la variable Importe del préstamo.

Percentiles

		Percentiles						
		5	10	25	50	75	90	95
Promedio ponderado (definición 1)	Importe del préstamo	734,48	1063,48	3000,00	8000,00	11000,00	18000,00	20000,00
Bisagras de Tukey	Importe del préstamo			3000,00	8000,00	11000,00		

Tabla A.3.16. Valores extremos de la variable Importe del préstamo.

Valores extremos

			Número del caso	Valor
Importe del préstamo	Mayores	1	61	70000
		2	113	50000
		3	1177	36000
		4	1143	35500
		5	300	35000[a]
	Menores	1	774	285
		2	1715	316
		3	1582	399
		4	1779	400
		5	985	400[b]

a. En la tabla de valores extremos mayores sólo se muestra una lista parcial de los casos con el valor 35000.

b. En la tabla de valores extremos menores sólo se muestra una lista parcial de los casos con el valor 400.

Figura A.3.7. Histograma de la variable Importe del préstamo.

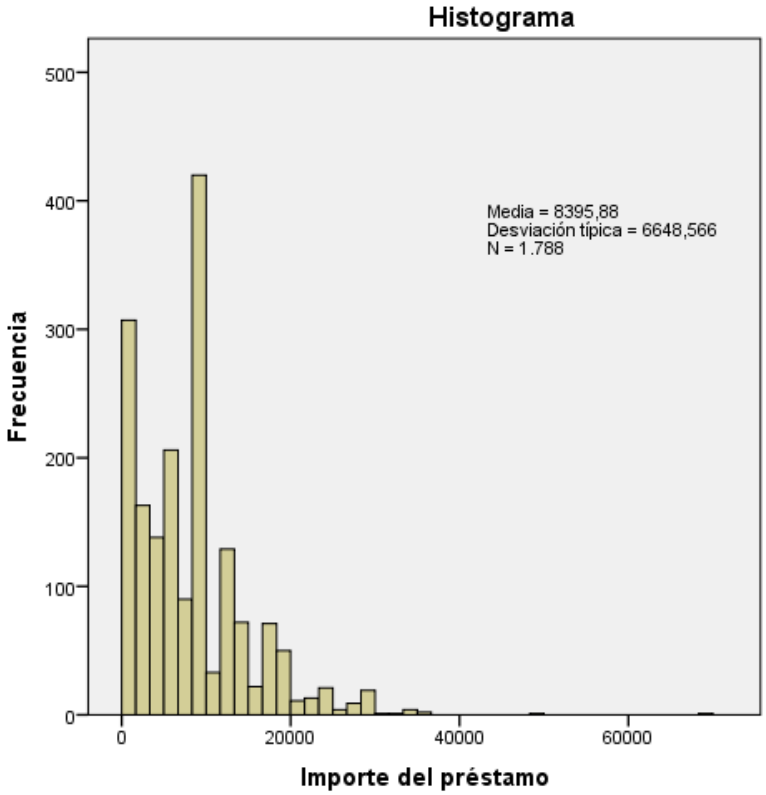


Figura A.3.8. Gráfico Q-Q normal de la variable Importe del préstamo.

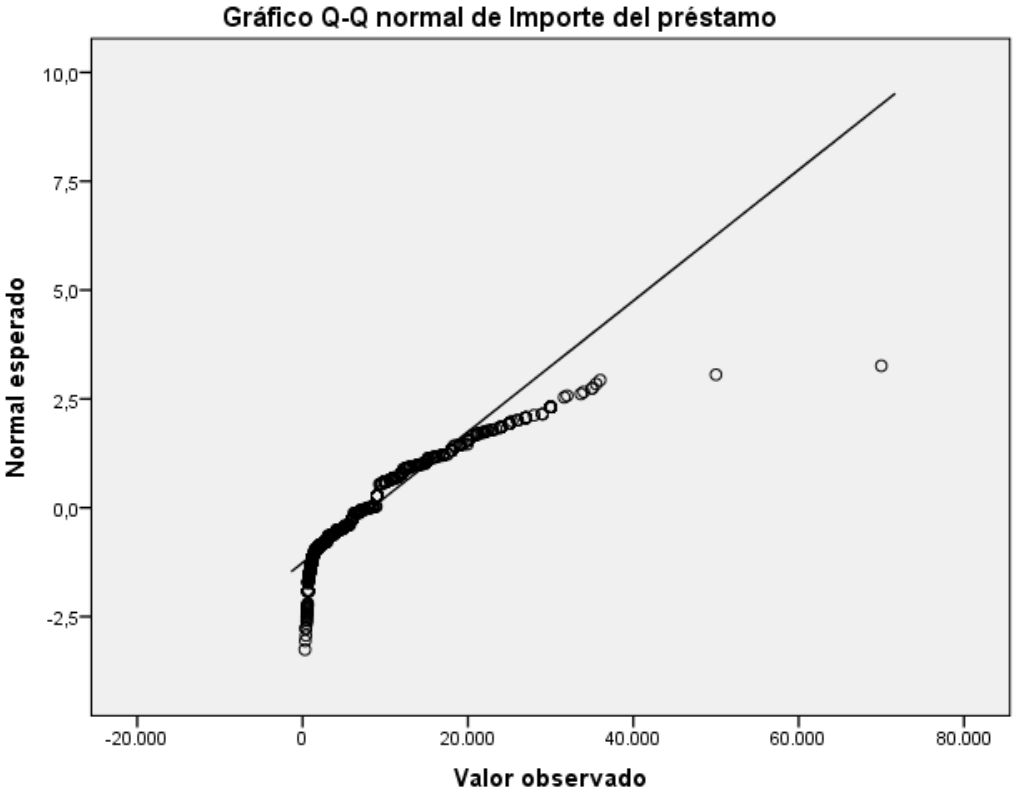


Tabla A.3.17. Estadísticos de la variable Importe de la inversión.

Descriptivos

			Estadístico	Error típ.
Importe de la inveisión	Media		10559,80	285,807
	Intervalo de confianza para la media al 95%	Límite inferior	9999,25	
		Límite superior	11120,35	
	Media recortada al 5%		9325,72	
	Mediana		9000,00	
	Varianza		146053984,7	
	Desv. típ.		12085,280	
	Mínimo		285	
	Máximo		210265	
	Rango		209980	
	Amplitud intercuartil		11500	
	Asimetría		7,073	,058
	Curtosis		92,528	,116

Tabla A.3.18. Estadísticos robustos de la variable Importe de la inversión.

Estimadores-M

	Estimador-M de Huber[a]	Biponderado de Tukey[b]	Estimador-M de Hampel[c]	Onda de Andrews[d]
Importe de la inversión	8606,81	8336,66	8875,25	8332,60

a. La constante de ponderación es 1,339.

b. La constante de ponderación es 4,685.

c. Las constantes de ponderación son 1,700, 3,400 y 8,500.

d. La constante de ponderación es 1,340*pi.

Tabla A.3.19. Percentiles de la variable Importe de la inversión.

Percentiles

		Percentiles						
		5	10	25	50	75	90	95
Promedio ponderado (definición 1)	Importe de la inveisión	734,48	1073,80	3500,00	9000,00	15000,00	21818,00	27279,89
Bisagras de Tukey	Importe de la inveisión			3500,00	9000,00	15000,00		

Tabla A.3.20. Valores extremos de la variable Importe de la inversión.

Valores extremos

			Número del caso	Valor
Importe de la invesión	Mayores	1	624	210265
		2	940	180000
		3	1546	180000
		4	507	138233
		5	255	90000
	Menores	1	774	285
		2	1715	317
		3	1582	399
		4	1779	400
		5	985	400[a]

a. En la tabla de valores extremos menores sólo se muestra una lista parcial de los casos con el valor 400.

Figura A.3.9. Histograma de la variable Importe de la inversión.

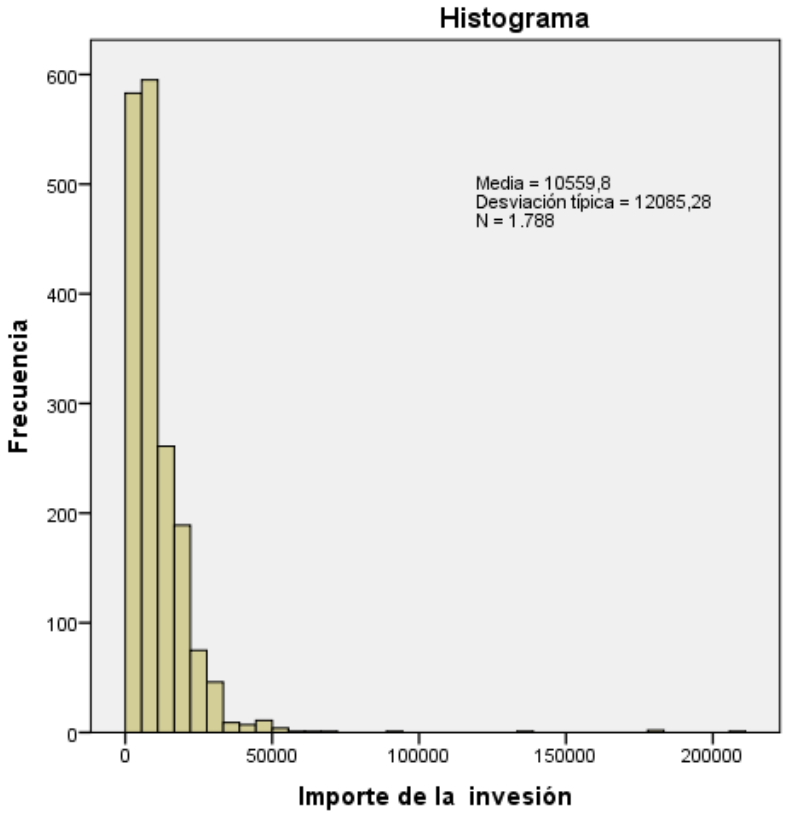


Figura A.3.10. Gráfico Q-Q normal de la variable Importe de la inversión.

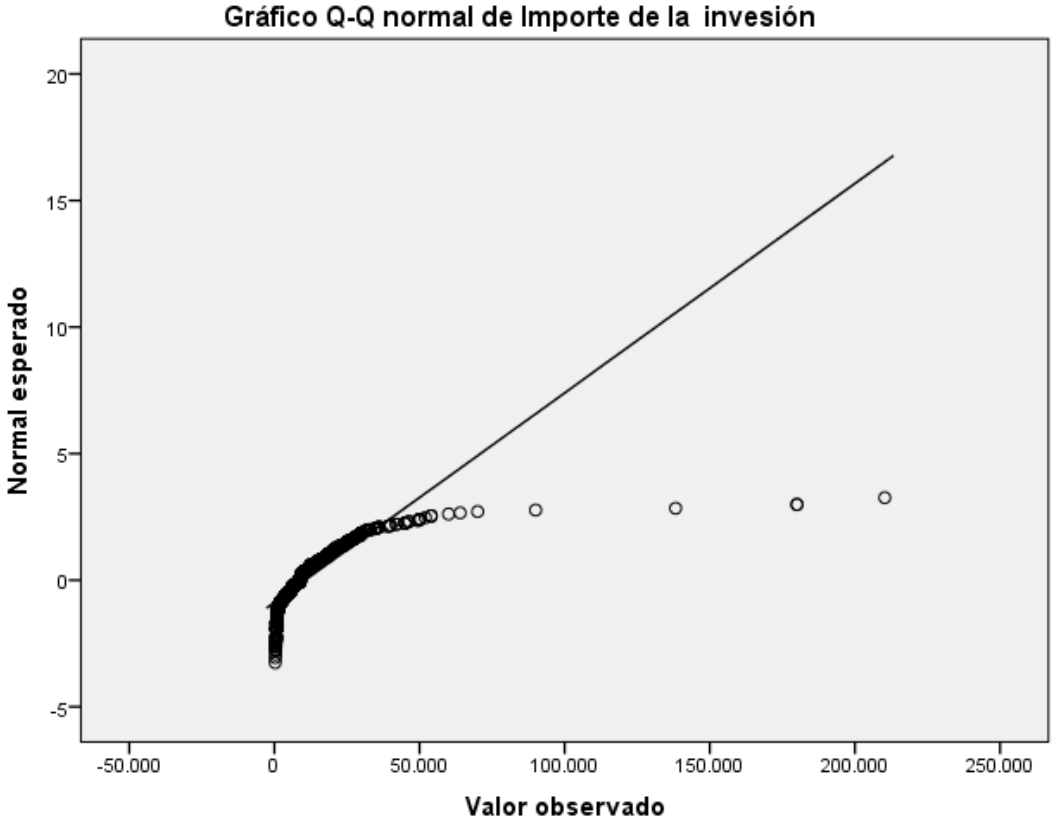


Tabla A.3.21. Estadísticos de la variable Importe de la cuota.

Descriptivos

			Estadístico	Error típ.
Importe de la cuota	Media		194,6263	3,86820
	Intervalo de confianza para la media al 95%	Límite inferior	187,0397	
		Límite superior	202,2130	
	Media recortada al 5%		184,2680	
	Mediana		188,0200	
	Varianza		26753,761	
	Desv. típ.		163,56577	
	Mínimo		8,78	
	Máximo		4567,78	
	Rango		4559,00	
	Amplitud intercuartil		152,78	
	Asimetría		11,876	,058
	Curtosis		295,646	,116

Tabla A.3.22. Estadísticos robustos de la variable Importe de la cuota.

Estimadores-M

	Estimador-M de Huber[a]	Biponderado de Tukey[b]	Estimador-M de Hampel[c]	Onda de Andrews[d]
Importe de la cuota	184,4403	176,9579	180,9005	176,8261

a. La constante de ponderación es 1,339.

b. La constante de ponderación es 4,685.

c. Las constantes de ponderación son 1,700, 3,400 y 8,500.

d. La constante de ponderación es 1,340*pi.

Tabla A.3.23 Percentiles de la variable Importe de la cuota

Percentiles

		Percentiles						
		5	10	25	50	75	90	95
Promedio ponderado (definición 1)	Importe de la cuota	24,2375	37,3700	97,2200	188,0200	250,0000	337,5460	400,7600
Bisagras de Tukey	Importe de la cuota			97,2200	188,0200	250,0000		

Tabla A.3.24. Valores extremos de la variable Importe de la cuota.

Valores extremos

			Número del caso	Valor
Importe de la cuota	Mayores	1	1724	4567,78
		2	61	2113,71
		3	1407	709,67
		4	678	701,12
		5	414	697,69
	Menores	1	1715	8,78
		2	1582	11,08
		3	1157	11,14
		4	909	12,50
		5	1292	13,75

Figura A.3.11. Histograma de la variable Importe de la cuota.

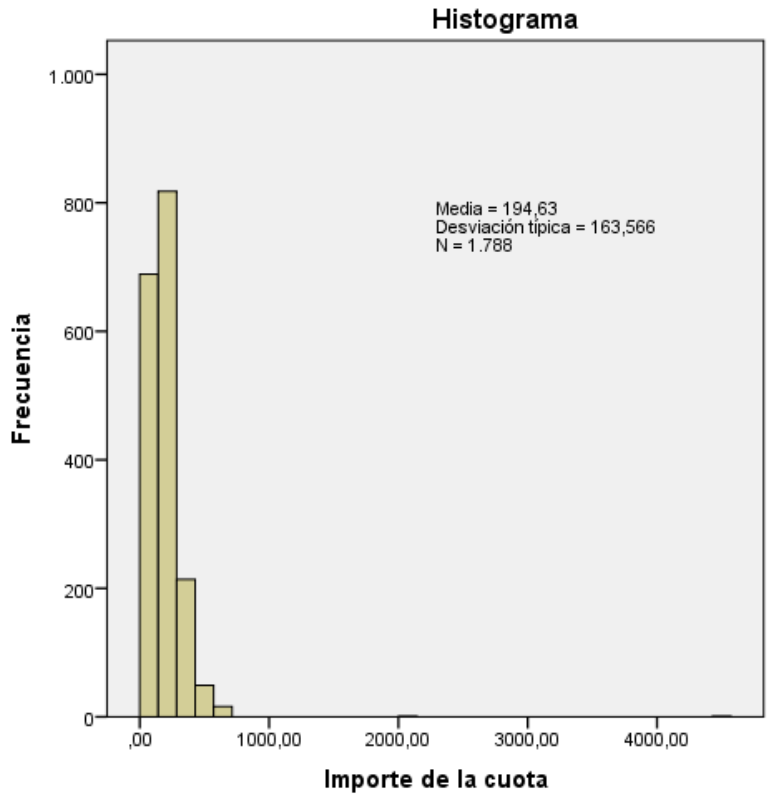


Figura A.3.12. Gráfico Q-Q normal de la variable Importe de la cuota.

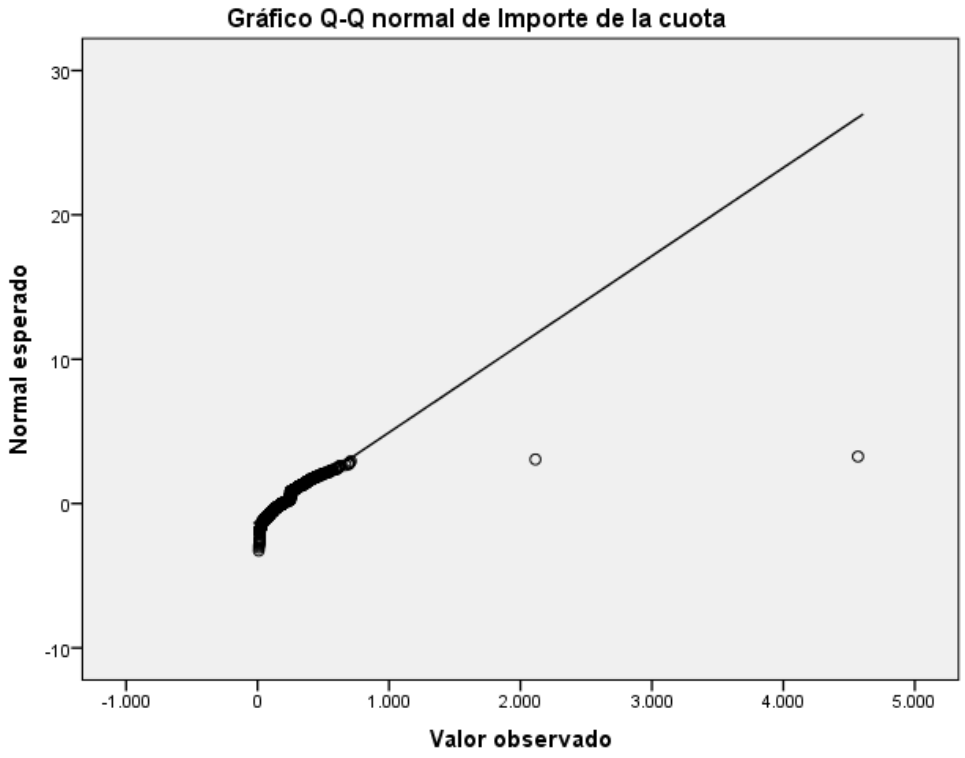


Tabla A.3.25. Estadísticos de la variable Ingresos.

Descriptivos

			Estadístico	Error típ.
Ingresos	Media		18935,40	266,020
	Intervalo de confianza para la media al 95%	Límite inferior	18413,66	
		Límite superior	19457,14	
	Media recortada al 5%		17981,51	
	Mediana		16452,97	
	Varianza		126531001,4	
	Desv. típ.		11248,600	
	Mínimo		0	
	Máximo		105978	
	Rango		105978	
	Amplitud intercuartil		10634	
	Asimetría		1,922	,058
	Curtosis		6,939	,116

Tabla A.3.26. Estadísticos robustos de la variable Ingresos.

Estimadores-M

	Estimador-M de Huber[a]	Biponderado de Tukey[b]	Estimador-M de Hampel[c]	Onda de Andrews[d]
Ingresos	17067,76	16200,21	16824,65	16183,71

a. La constante de ponderación es 1,339.

b. La constante de ponderación es 4,685.

c. Las constantes de ponderación son 1,700, 3,400 y 8,500.

d. La constante de ponderación es 1,340*pi.

Tabla A.3.27 Percentiles de la variable Ingresos.

Percentiles

		Percentiles						
		5	10	25	50	75	90	95
Promedio ponderado (definición 1)	Ingresos	5034,00	7776,86	12561,72	16452,97	23195,66	32145,36	40000,00
Bisagras de Tukey	Ingresos			12565,43	16452,97	23187,77		

Tabla A.3.28. Valores extremos de la variable Ingresos.

Valores extremos

			Número del caso	Valor
Ingresos	Mayores	1	1499	105978
		2	1356	91979
		3	400	89066
		4	1484	81679
		5	391	80687
	Menores	1	1758	0
		2	1755	0
		3	1714	0
		4	1640	0
		5	1603	0[a]

a. En la tabla de valores extremos menores sólo se muestra una lista parcial de los casos con el valor 0.

Figura A.3.13. Histograma de la variable Ingresos.

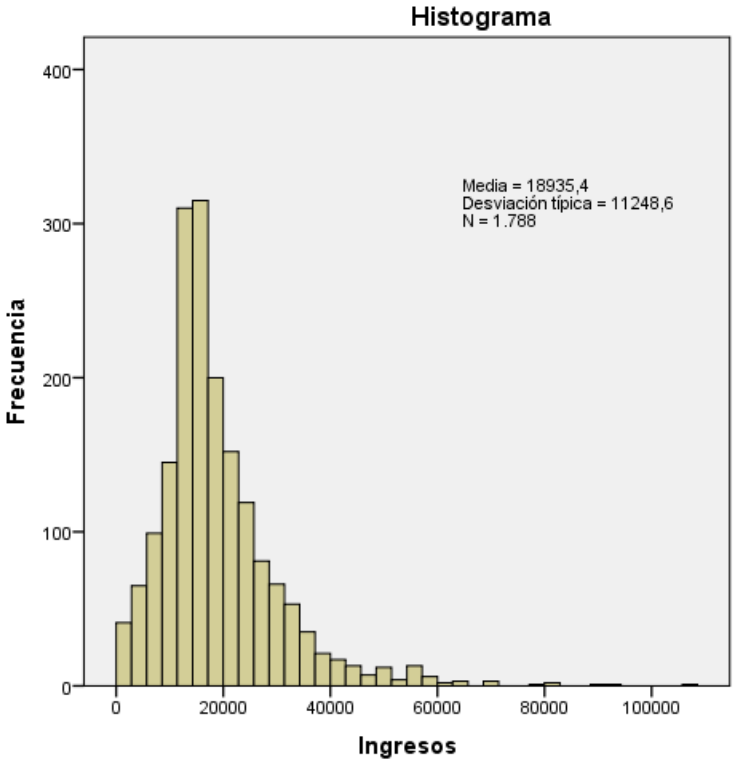


Figura A.3.14. Gráfico Q-Q normal de la variable Ingresos.

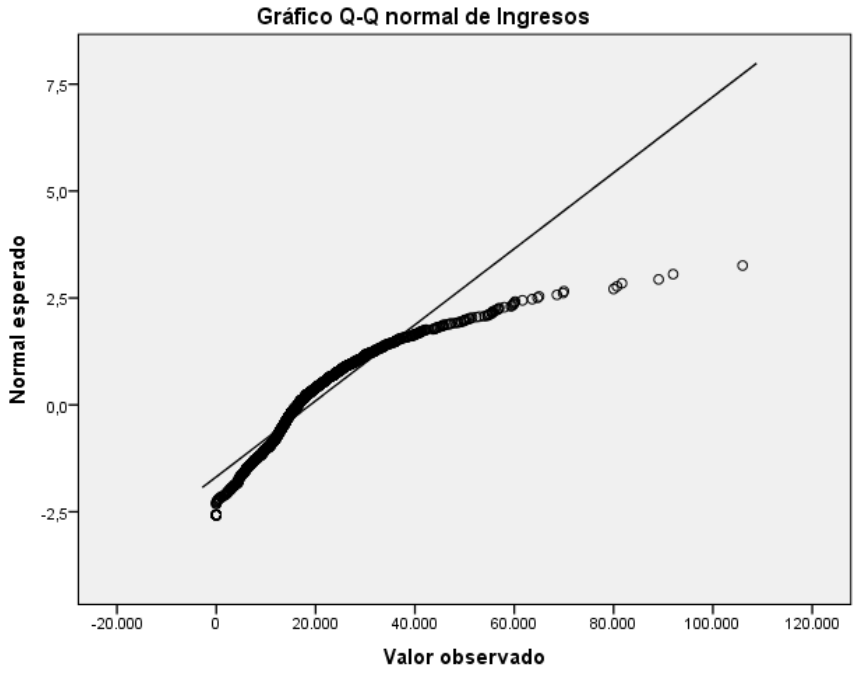


Tabla A.3.29. Estadísticos de la variable Importe pendientes.

Descriptivos

			Estadístico	Error típ.
Importes pendientes	Media		26474,02	1237,982
	Intervalo de confianza para la media al 95%	Límite inferior	24045,98	
		Límite superior	28902,06	
	Media recortada al 5%		18541,86	
	Mediana		1417,04	
	Varianza		2740285962	
	Desv. típ.		52347,741	
	Mínimo		0	
	Máximo		776367	
	Rango		776367	
	Amplitud intercuartil		32847	
	Asimetría		4,330	,058
	Curtosis		34,489	,116

Tabla A.3.30. Estadísticos robustos de la variable Importe pendientes.

Estimadores-M

	Estimador-M de Huber[a]	Biponderado de Tukey[b]	Estimador-M de Hampel[c]	Onda de Andrews[d]
Importes pendientes	1811,73	347,20	543,66	344,82

a. La constante de ponderación es 1,339.

b. La constante de ponderación es 4,685.

c. Las constantes de ponderación son 1,700, 3,400 y 8,500.

d. La constante de ponderación es 1,340*pi.

Tabla A.3.31. Percentiles de la variable Importe pendientes.

Percentiles

		Percentiles						
		5	10	25	50	75	90	95
Promedio ponderado (definición 1)	Importes pendientes	,00	,00	,00	1417,04	32847,45	86092,24	125334,97
Bisagras de Tukey	Importes pendientes			,00	1417,04	32835,88		

Tabla A.3.32. Valores extremos de la variable Importes pendientes

Valores extremos

			Número del caso	Valor
Importes pendientes	Mayores	1	1460	776367
		2	1076	437300
		3	1761	393000
		4	1142	386424
		5	1139	381924
	Menores	1	1787	0
		2	1782	0
		3	1781	0
		4	1780	0
		5	1779	0[a]

a. En la tabla de valores extremos menores sólo se muestra una lista parcial de los casos con el valor 0.

Figura A.3.15. Histograma de la variable Importes pendientes.

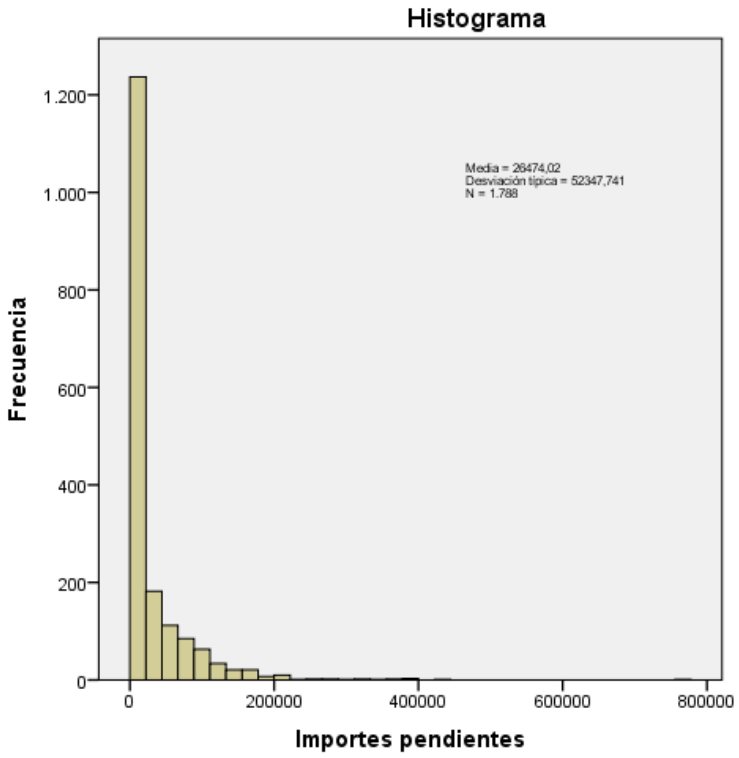


Figura A.3.16. Gráfico Q-Q normal de la variable Importes pendientes.

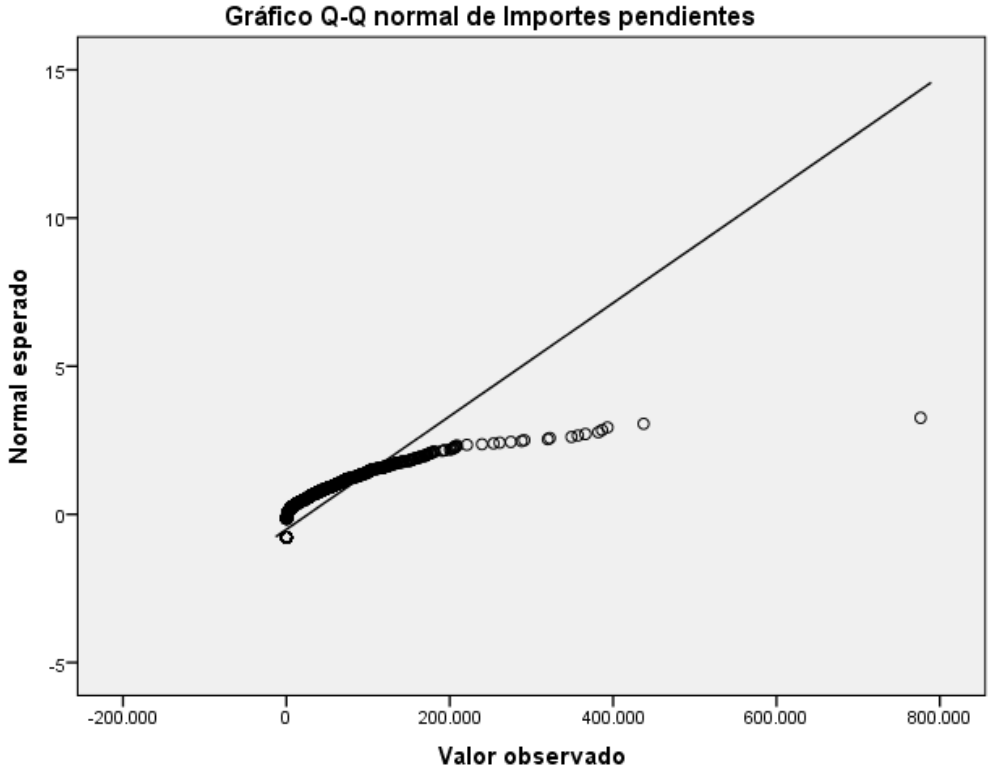


Tabla A.3.33. Estadísticos de la variable Saldo medio.

Descriptivos

			Estadístico	Error típ.
Saldo medio	Media		4884,62	337,936
	Intervalo de confianza para la media al 95%	Límite inferior	4221,83	
		Límite superior	5547,41	
	Media recortada al 5%		2645,82	
	Mediana		865,45	
	Varianza		204190598,8	
	Desv. típ.		14289,528	
	Mínimo		-920	
	Máximo		339116	
	Rango		340036	
	Amplitud intercuartil		3790	
	Asimetría		10,371	,058
	Curtosis		186,156	,116

Tabla A.3.34. Estadísticos robustos de la variable Saldo medio.

Estimadores-M

	Estimador-M de Huber[a]	Biponderado de Tukey[b]	Estimador-M de Hampel[c]	Onda de Andrews[d]
Saldo medio	1081,01	617,18	830,03	613,85

a. La constante de ponderación es 1,339.

b. La constante de ponderación es 4,685.

c. Las constantes de ponderación son 1,700, 3,400 y 8,500.

d. La constante de ponderación es 1,340*pi.

Tabla A.3.35. Percentiles de la variable Saldo medio.

Percentiles

		Percentiles						
		5	10	25	50	75	90	95
Promedio ponderado (definición 1)	Saldo medio	,00	,00	82,24	865,45	3872,67	11939,35	21842,97
Bisagras de Tukey	Saldo medio			82,60	865,45	3869,00		

Tabla A.3.36. Valores extremos de la variable Saldo medio.

Valores extremos

			Número del caso	Valor
Saldo medio	Mayores	1	1444	339116
		2	801	145937
		3	1624	132896
		4	1037	114872
		5	1022	107228
	Menores	1	301	-920
		2	1128	-507
		3	1652	-447
		4	1505	-337
		5	1635	-320

Figura A.3.17. Histograma de la variable Saldo medio.

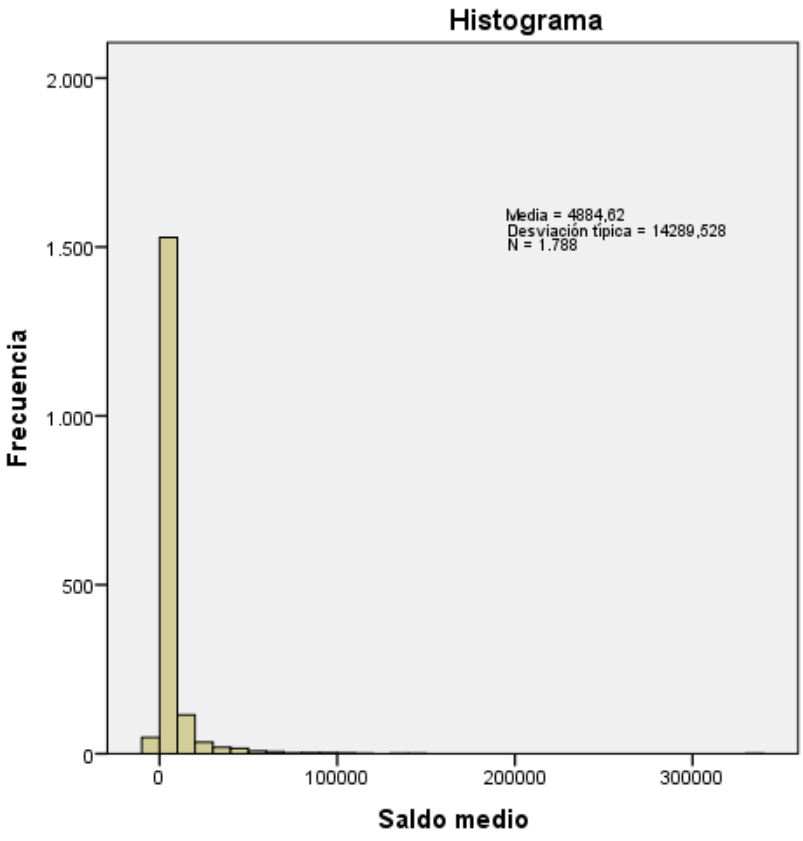


Figura A.3.18. Gráfico Q-Q normal de la variable Saldo medio.

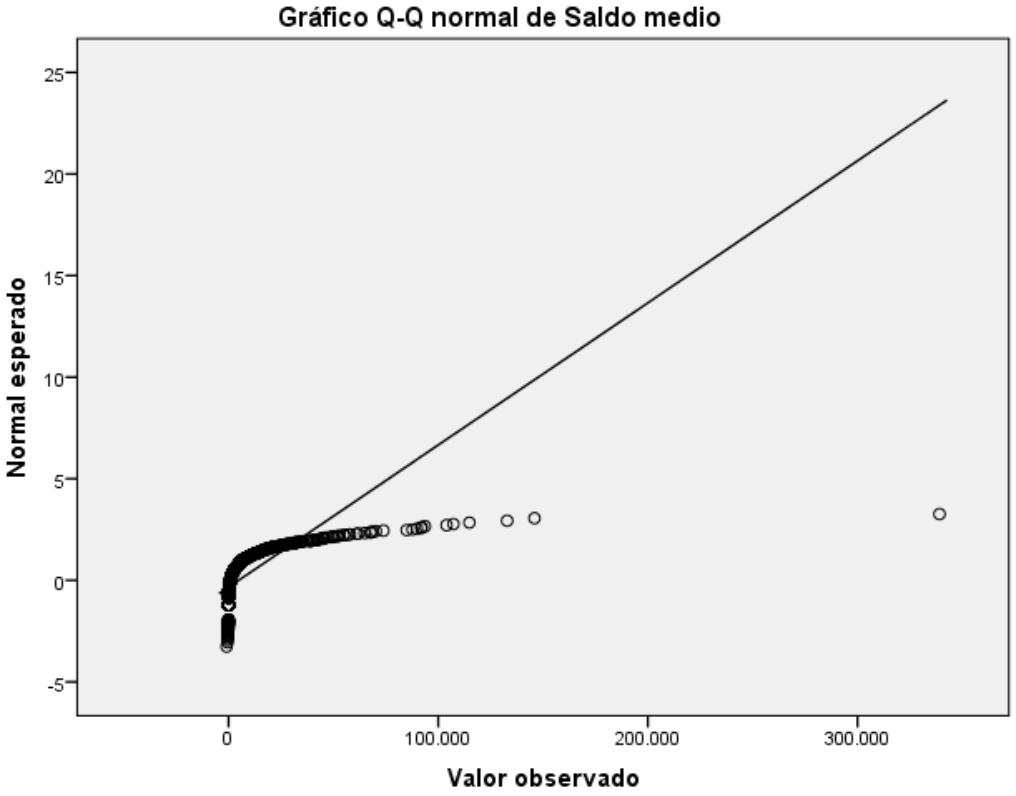


Tabla A.3.37. Estadísticos de la variable Edad.

Descriptivos

			Estadístico	Error típ.
Edad	Media		43,92	,290
	Intervalo de confianza para la media al 95%	Límite inferior	43,36	
		Límite superior	44,49	
	Media recortada al 5%		43,69	
	Mediana		44,00	
	Varianza		150,244	
	Desv. típ.		12,257	
	Mínimo		20	
	Máximo		90	
	Rango		70	
	Amplitud intercuartil		19	
	Asimetría		,239	,058
	Curtosis		-,522	,116

Tabla A.3.38. Estadísticos robustos de la variable Edad.

Estimadores-M

	Estimador-M de Huber[a]	Biponderado de Tukey[b]	Estimador-M de Hampel[c]	Onda de Andrews[d]
Edad	43,54	43,45	43,59	43,45

a. La constante de ponderación es 1,339.

b. La constante de ponderación es 4,685.

c. Las constantes de ponderación son 1,700, 3,400 y 8,500.

d. La constante de ponderación es 1,340*pi.

Tabla A.3.39. Percentiles de la variable Edad.

Percentiles

		Percentiles						
		5	10	25	50	75	90	95
Promedio ponderado (definición 1)	Edad	25,00	28,00	34,00	44,00	53,00	60,00	64,00
Bisagras de Tukey	Edad			34,00	44,00	53,00		

Tabla A.3.40. Valores extremos de la variable Edad.

Valores extremos

			Número del caso	Valor
Edad	Mayores	1	1521	90
		2	52	82
		3	12	80
		4	1393	78
		5	1073	77[a]
	Menores	1	1762	20
		2	1714	20
		3	1671	20
		4	1638	20
		5	1634	20[b]

a. En la tabla de valores extremos mayores sólo se muestra una lista parcial de los casos con el valor 77.

b. En la tabla de valores extremos menores sólo se muestra una lista parcial de los casos con el valor 20.

Figura A.3.19. Histograma de la variable Edad.

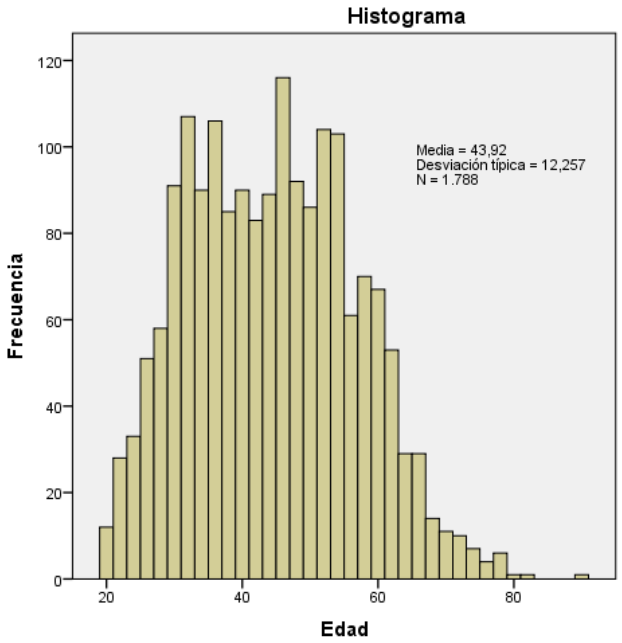
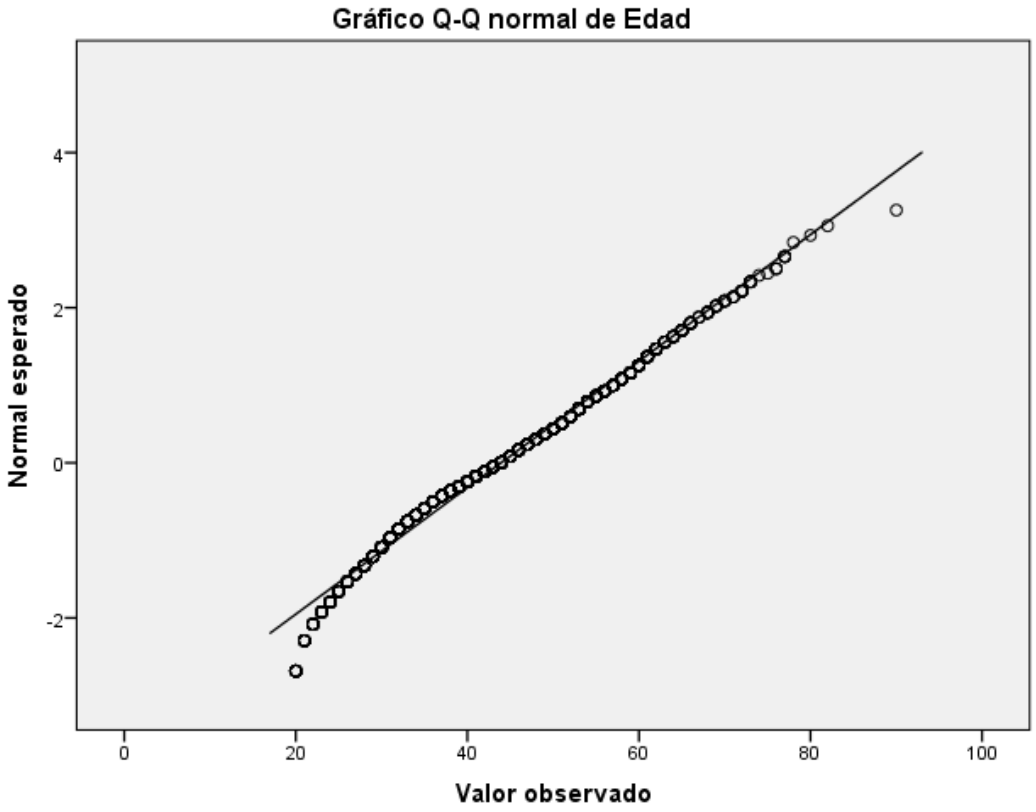


Figura A.3.20. Gráfico Q-Q normal de la variable Edad



ANEXO 3: ANÁLISIS DESCRIPTIVO DE LAS VARIABLES CUANTITATIVAS DE LA BASE DE DATOS UTILIZADA.

Tabla A.3.41. Estadísticos de la variable Porcentaje del préstamo.

Descriptivos

			Estadístico	Error típ.
Porcentanje del préstamo	Media		121,69	2,101
	Intervalo de confianza para la media al 95%	Límite inferior	117,57	
		Límite superior	125,81	
	Media recortada al 5%		110,34	
	Mediana		100,00	
	Varianza		7889,588	
	Desv. típ.		88,823	
	Mínimo		100	
	Máximo		2000	
	Rango		1900	
	Amplitud intercuartil		13	
	Asimetría		14,112	,058
	Curtosis		261,917	,116

Tabla A.3.42. Estadísticos robustos de la variable Porcentaje del préstamo.

Estimadores-M[a]

	Estimador-M de Huber[b]	Biponderado de Tukey[c]	Estimador-M de Hampel[d]	Onda de Andrews[e]
Porcentanje del préstamo	.	.	.	.

a. No se pueden calcular algunos estimadores-M debido a que la distribución se centra sobre todo en la mediana.

b. La constante de ponderación es 1,339.

c. La constante de ponderación es 4,685.

d. Las constantes de ponderación son 1,700, 3,400 y 8,500.

e. La constante de ponderación es 1,340*pi.

Tabla A.3.43. Percentiles de la variable Porcentaje del préstamo.

Percentiles

		Percentiles						
		5	10	25	50	75	90	95
Promedio ponderado (definición 1)	Porcentanje del préstamo	100,00	100,00	100,00	100,00	112,83	165,07	203,58
Bisagras de Tukey	Porcentanje del préstamo			100,00	100,00	112,80		

Tabla A.3.44. Valores extremos de la variable Porcentaje del préstamo.

Valores extremos

			Número del caso	Valor
Porcentanje del préstamo	Mayores	1	940	2000
		2	1546	2000
		3	507	1455
		4	428	1000
		5	624	876
	Menores	1	1788	100
		2	1787	100
		3	1785	100
		4	1783	100
		5	1782	100[a]

a. En la tabla de valores extremos menores sólo se muestra una lista parcial de los casos con el valor 100.

Figura A.3.21. Histograma de la variable Porcentaje del préstamo.

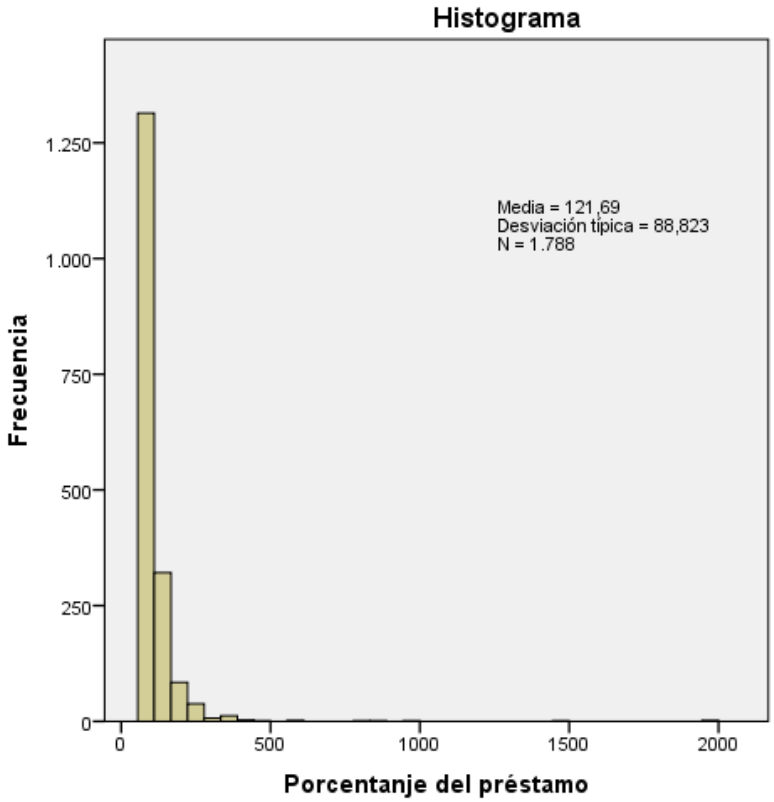
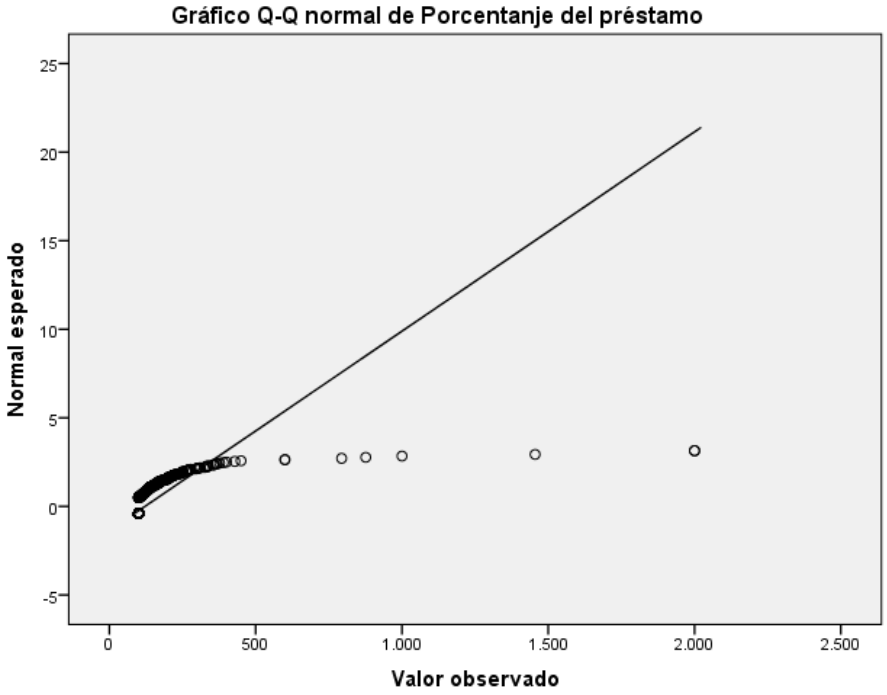


Figura A.3.22. Gráfico Q-Q normal de la variable Porcentaje del préstamo.



ANEXO 4

DISTRIBUCIONES DE PROBABILIDAD CONDICIONADA DE LAS VARIABLES DE LA RED BAYESIANA.

ANEXO 4: DISTRIBUCIONES DE PROBABILIDAD CONDICIONADA DE LAS VARIABLES DE UNA RED BAYESIANA.

Anexo 4. Distribuciones de probabilidad condicionada de las variables de una red bayesiana.

En este anexo se muestran las tablas que contienen los valores calculados de la distribución de probabilidad condicionada de las variables explicativas del modelo de credit scoring, calculadas a través de una red bayesiana que utiliza el método de búsqueda de ascensión de colinas (Hill Climber) con dos padres.

Tabla A.4.1. Distribución de probabilidad condicionada de la variable Estado civil.

		ESTADO CIVIL			
CLASE	**FAMILIA**	**Casado**	**Separado**	**Soltero**	**Total**
SÍ	**Uno**	0,005	0,102	0,893	1
SÍ	**Dos**	0,333	0,333	0,333	1
SÍ	**Mayor de 2**	0,758	0,039	0,203	1
NO	**Uno**	0,018	0,026	0,956	1
NO	**Dos**	0,111	0,009	0,880	1
NO	**Mayor de 2**	0,429	0,087	0,484	1

Tabla A.4.2. Distribución de probabilidad condicionada de la variable Valor de la vivienda.

		VALOR DE LA VIVIENDA		
CLASE	**ESTADO CIVIL**	**Menor de 27.112**	**Mayor o igual de 27.112**	**Total**
SÍ	**Casado**	0,165	0,835	1
SÍ	**Separado**	0,237	0,763	1
SÍ	**Soltero**	0,575	0,425	1
NO	**Casado**	0,450	0,550	1
NO	**Separado**	0,781	0,219	1
NO	**Soltero**	0,922	0,078	1

Tabla A.4.3. Distribución de probabilidad condicionada de la variable Ingresos.

		INGRESOS		
CLASE	VALOR DE LA VIVIENDA	Menor de 23.210	Mayor o igual de 23.210	Total
SÍ	Menor de 27.112	0,860	0,140	1
SÍ	Mayor o igual de 27.112	0,624	0,376	1
NO	Menor de 27.112	0,929	0,071	1
NO	Mayor o igual de 27.112	0,828	0,172	1

Tabla A.4.4. Distribución de probabilidad condicionada de la variable Tipo de trabajo.

		TIPO DE TRABAJO (RELACIÓN LABORAL)								
CLASE	INGRESOS	Técnico-mando internmedio	Obrero fijo	Obrero temporal	Obrero fijo especiaizado	Obrero temporal especializado	Autónomo	Jubilado rentista	No activo	Total
SÍ	Menor de 23.210	0,190	0,341	0,123	0,094	0,019	0,077	0,073	0,082	1
SÍ	Mayor o igual de 23.210	0,451	0,170	0,044	0,112	0,024	0,141	0,053	0,005	1
NO	Menor de 23.210	0,021	0,380	0,451	0,040	0,011	0,044	0,021	0,031	1
NO	Mayor o igual de 23.210	0,318	0,227	0,076	0,015	0,015	0,288	0,045	0,015	1

Tabla A.4.5. Distribución de probabilidad condicionada de la variable Saldo medio.

		SALDO MEDIO DE LAS CUENTAS			
CLASE	INGRESOS	Menor de 453	Entre 454 y 4.543	Mayor de 4.543	Total
SÍ	Menor de 23.210	0,374	0,378	0,247	1
SÍ	Mayor o igual de 23.210	0,224	0,413	0,363	1
NO	Menor de 23.210	0,850	0,135	0,015	1
NO	Mayor o igual de 23.210	0,443	0,508	0,049	1

Tabla A.4.6. Distribución de probabilidad condicionada de la variable Tipo de vivienda.

		TIPO DE VIVIENDA					
CLASE	**VALOR DE LA VIVIENDA**	**Propiedad libre de cargas**	**Propiedad hipotecada**	**Alquiler**	**Domicilio con la familia**	**Otros**	**Total**
SÍ	**Menor de 27.112**	0,013	0,093	0,262	0,511	0,120	1
SÍ	**Mayor o igual de 27.112**	0,453	0,519	0,002	0,024	0,002	1
NO	**Menor de 27.113**	0,002	0,081	0,617	0,241	0,060	1
NO	**Mayor o igual de 27.113**	0,152	0,744	0,024	0,040	0,040	1

Tabla A.4.7. Distribución de probabilidad condicionada de la variable Importes pendientes.

			IMPORTES PENDIENTES		
VIVIENDA	**VALOR DE LA VIVIENDA**	**CLASE**	**Menor de 4.150**	**Mayor o igual de 4.150**	**Total**
Propiedad libre de cargas	**Menor de 27.112**	**SÍ**	0,750	0,250	1
Propiedad libre de cargas	**Mayor o igual de 27.112**	**NO**	0,500	0,500	1
Propiedad libre de cargas	**Menor de 27.112**	**SÍ**	0,752	0,248	1
Propiedad libre de cargas	**Mayor o igual de 27.112**	**NO**	0,650	0,350	1
Propiedad hipotecada	**Menor de 27.112**	**SÍ**	0,136	0,864	1
Propiedad hipotecada	**Mayor o igual de 27.112**	**NO**	0,196	0,804	1
Propiedad hipotecada	**Menor de 27.112**	**SÍ**	0,208	0,792	1
Propiedad hipotecada	**Mayor o igual de 27.112**	**NO**	0,223	0,777	1
Alquiler	**Menor de 27.112**	**SÍ**	0,883	0,117	1
Alquiler	**Mayor o igual de 27.112**	**NO**	0,939	0,061	1
Alquiler	**Menor de 27.112**	**SÍ**	0,500	0,500	1
Alquiler	**Mayor o igual de 27.112**	**NO**	0,250	0,750	1
Domicilio con la familia	**Menor de 27.112**	**SÍ**	0,871	0,129	1
Domicilio con la familia	**Mayor o igual de 27.112**	**NO**	0,978	0,022	1
Domicilio con la familia	**Menor de 27.112**	**SÍ**	0,250	0,750	1
Domicilio con la familia	**Mayor o igual de 27.112**	**NO**	0,167	0,833	1
Otros	**Menor de 27.112**	**SÍ**	0,893	0,107	1
Otros	**Mayor o igual de 27.112**	**NO**	0,912	0,088	1
Otros	**Menor de 27.112**	**SÍ**	0,500	0,500	1
Otros	**Mayor o igual de 27.112**	**NO**	0,500	0,500	1

Tabla A.4.8. Distribución de probabilidad condicionada de la variable Nacionalidad.

		NACIONALIDAD		
CLASE	**TIPO DE VIVIENDA**	**Español**	**Extranjero**	**Total**
SÍ	**Propiedad libre de cargas**	0,995	0,005	1
SÍ	**Propiedad hipotecada**	0,895	0,105	1
SÍ	**Alquiler**	0,417	0,583	1
SÍ	**Domicilio con la familia**	0,944	0,056	1
SÍ	**Otros**	0,893	0,107	1
NO	**Propiedad libre de cargas**	0,950	0,050	1
NO	**Propiedad hipotecada**	0,601	0,399	1
NO	**Alquiler**	0,317	0,683	1
NO	**Domicilio con la familia**	0,746	0,254	1
NO	**Otros**	0,447	0,553	1

Tabla A.4.9. Distribución de probabilidad condicionada de la variable Importe de la cuota.

		IMPORTE DE LA CUOTA					
CLASE	**IMPORTE DE LA INVERSIÓN**	**Menos de 41**	**Entre 41 y 81**	**Entre 82 y 241**	**Entre 242 y 250**	**Mayor de 250**	**Total**
SÍ	**Menor de 565**	0,200	0,200	0,200	0,200	0,200	1
SÍ	**Entre 565 y 1.484**	0,670	0,235	0,078	0,009	0,009	1
SÍ	**Entre 1.485 y 5.999**	0,012	0,470	0,494	0,012	0,012	1
SÍ	**Entre 6.000 y 8.712**	0,011	0,034	0,888	0,011	0,056	1
SÍ	**Mayor de 8.712**	0,002	0,012	0,252	0,362	0,372	1
NO	**Menor de 565**	0,143	0,714	0,048	0,048	0,048	1
NO	**Entre 565 y 1.484**	0,030	0,212	0,697	0,030	0,030	1
NO	**Entre 1.485 y 5.999**	0,008	0,059	0,898	0,003	0,033	1
NO	**Entre 6.000 y 8.712**	0,010	0,010	0,806	0,010	0,165	1
NO	**Mayor de 8.712**	0,007	0,034	0,366	0,048	0,545	1

Tabla A.4.10. Distribución de probabilidad condicionada de la variable Importe del patrimonio.

			IMPORTE DEL PATRIMONIO		
IMPORTE DEL PRÉSTAMO	**VALOR DE LA VIVIENDA**	**CLASE**	**Menor de 4.500**	**Mayor o igual de 4.500**	**Total**
Menor de 1.471	**Menor de 27.112**	**SÍ**	0,853	0,147	1
Menor de 1.471	**Menor de 27.112**	**NO**	0,981	0,019	1
Menor de 1.471	**Mayor o igual de 27.112**	**SÍ**	0,863	0,138	1
Menor de 1.471	**Mayor o igual de 27.112**	**NO**	0,500	0,500	1
Entre 1.471 y 5.973	**Menor de 27.112**	**SÍ**	0,981	0,019	1
Entre 1.471 y 5.973	**Menor de 27.112**	**NO**	0,997	0,003	1
Entre 1.471 y 5.973	**Mayor o igual de 27.112**	**SÍ**	0,881	0,119	1
Entre 1.471 y 5.973	**Mayor o igual de 27.112**	**NO**	0,944	0,056	1
Mayor de 5.973	**Menor de 27.112**	**SÍ**	0,848	0,152	1
Mayor de 5.973	**Menor de 27.112**	**NO**	0,904	0,096	1
Mayor de 5.973	**Mayor o igual de 27.112**	**SÍ**	0,732	0,268	1
Mayor de 5.973	**Mayor o igual de 27.112**	**NO**	0,694	0,306	1

Tabla A.4.11. Distribución de probabilidad condicionada de la variable Finalidad del crédito.

			FINALIDAD DEL CRÉDITO									
IMPORTE DE LA INVERSIÓN	**EDAD**	**CLASE**	**Reformas viviendas**	**Compra automóvil**	**Compra electrodomésticos**	**Compra ordenador**	**Mobiliario y decoración**	**Otros bien y servicios**	**Servicios sanitarios**	**Imprevistos familiares**	**Otras finalidades**	**Total**
Menor de 565	**Menor de 50**	**SÍ**	0,111	0,111	0,111	0,111	0,111	0,111	0,111	0,111	0,111	1
Menor de 565	**Mayor o igual que 50**	**NO**	0,130	0,043	0,130	0,130	0,043	0,130	0,043	0,217	0,130	1
Menor de 565	**Menor de 50**	**SÍ**	0,111	0,111	0,111	0,111	0,111	0,111	0,111	0,111	0,111	1
Menor de 565	**Mayor o igual que 50**	**NO**	0,091	0,091	0,091	0,091	0,091	0,091	0,091	0,091	0,273	1
Entre 565 y 1.484	**Menor de 50**	**SÍ**	0,011	0,011	0,195	0,563	0,011	0,011	0,011	0,011	0,172	1
Entre 565 y 1.484	**Mayor o igual que 50**	**NO**	0,029	0,029	0,143	0,429	0,029	0,029	0,029	0,257	0,029	1
Entre 565 y 1.484	**Menor de 50**	**SÍ**	0,024	0,024	0,317	0,512	0,024	0,024	0,024	0,024	0,024	1
Entre 565 y 1.484	**Mayor o igual que 50**	**NO**	0,091	0,091	0,091	0,091	0,091	0,091	0,091	0,273	0,091	1
Entre 1.485 y 5.999	**Menor de 50**	**SÍ**	0,072	0,159	0,043	0,072	0,072	0,072	0,072	0,217	0,217	1
Entre 1.485 y 5.999	**Mayor o igual que 50**	**NO**	0,133	0,190	0,025	0,014	0,042	0,014	0,065	0,377	0,139	1
Entre 1.485 y 5.999	**Menor de 50**	**SÍ**	0,185	0,185	0,037	0,185	0,111	0,111	0,111	0,037	0,037	1
Entre 1.485 y 5.999	**Mayor o igual que 50**	**NO**	0,176	0,176	0,020	0,020	0,059	0,059	0,020	0,451	0,020	1
Entre 6.000 y 8.712	**Menor de 50**	**SÍ**	0,176	0,373	0,020	0,020	0,059	0,137	0,020	0,137	0,059	1
Entre 6.000 y 8.712	**Mayor o igual que 50**	**NO**	0,192	0,394	0,010	0,030	0,030	0,051	0,051	0,192	0,051	1
Entre 6.000 y 8.712	**Menor de 50**	**SÍ**	0,216	0,059	0,059	0,020	0,176	0,059	0,216	0,137	0,059	1
Entre 6.000 y 8.712	**Mayor o igual que 50**	**NO**	0,176	0,176	0,059	0,059	0,176	0,176	0,059	0,059	0,059	1
Mayor de 8.712	**Menor de 50**	**SÍ**	0,152	0,597	0,004	0,004	0,095	0,037	0,037	0,062	0,012	1
Mayor de 8.713	**Mayor o igual que 50**	**NO**	0,122	0,675	0,008	0,008	0,024	0,041	0,008	0,073	0,041	1
Mayor de 8.714	**Menor de 50**	**SÍ**	0,357	0,298	0,006	0,006	0,064	0,053	0,076	0,088	0,053	1
Mayor de 8.715	**Mayor o igual que 50**	**NO**	0,314	0,200	0,029	0,029	0,029	0,029	0,029	0,314	0,029	1