Universidad Nacional de Estudios a Distancia

Escuela de Ingeniería Informática

Máster Universitario en Ingeniería y Ciencia de Datos

Optimización del servicio de recogida de residuos sólidos urbanos mediante predicción del nivel de llenado de los contenedores

Trabajo de Fin de Máster

Presentado por: Francisco Marías Ferrer

Director: Salvador Ros Muñoz

Curso académico: 2020/21 (Julio)

Contenido

Índice de figuras

Índice de tablas

Índice de Ecuaciones

1. Resumen

En el ámbito del *Internet of Things (IoT)*, las técnicas de Machine Learning nos ofrecen múltiples de posibilidades que pueden ayudar a la toma de decisiones de alto nivel. Cada vez son más las ciudades que se unen al movimiento de las Smart Cities captando datos del entorno a través de sensores instalados en campo, donde uno de los sectores clave es la recogida de residuos sólidos urbanos. El rápido crecimiento demográfico, sobre todo en grandes núcleos urbanos, junto con la sobreexplotación de recursos hace que sea necesaria una óptima gestión del servicio de recogida de residuos que supone en torno a un 70% del coste total del tratamiento de residuos.

El objetivo de este trabajo es analizar mecanismos de machine learning que sirvan para optimizar el servicio de recogida de residuos sólidos urbanos a través de técnicas de Machine Learning haciendo uso de los datos históricos del servicio enviados al Internet of Things consiguiendo no solo que las empresas sean más productivas sino consiguiendo un importante ahorro ambiental a través de una reducción de las emisiones de CO2 de los camiones reduciendo así la huella de carbono derivada del servicio.

Palabras clave: Machine Learning, Internet Of Things, optimización recogida de residuos, series temporales, ARIMA.

2. Abstract

In the field of the Internet of Things (IoT), Machine Learning techniques offer us endless possibilities that can help make high-level decisions. More and more cities are joining the Smart Cities movement by capturing data from the environment through sensors installed in the field, where one of the key sectors is the collection of municipal solid waste. Rapid population growth, especially in large urban centers, together with the overexploitation of resources makes it necessary to optimize the waste collection service, which represents around 70% of the total cost of waste treatment.

The objective of this work is to optimize the municipal solid waste collection service through Machine Learning techniques making use of the historical data of the service sent to the Internet of Things, not only making companies more productive but also achieving significant environmental savings through a reduction in CO2 emissions from trucks, thus reducing the carbon footprint derived from the service.

Keywords: Machine Learning, Internet of Things, waste collection optimization, time series, ARIMA.

3. Introducción

En la actualidad, la **humanidad se enfrenta a graves problemas ambientales** que cada día van en aumento, como el cambio climático, la contaminación, deforestación, degradación de suelos, extinción de especies y pérdida de biodiversidad entre otros muchos. Para ponerles solución tenemos que adaptarnos y tomar acciones inmediatas que contribuyan al uso sostenible de los recursos naturales y al reciclaje de los residuos generados en pro de una mejor protección del planeta y los recursos que nos ofrece.

Debido al aumento de la población mundial, el rápido desarrollo tecnológico y la fabricación de bienes de consumo de corta vida que son desechados diariamente, así como los embalajes que llevan (cajas de cartón y protecciones plásticas) no solo en los propios bienes en sí sino también, cada vez de forma más habitual, en los alimentos que compramos en los supermercados, están influyendo negativamente en el Medio Ambiente.

El **desarrollo tecnológico** está **afectando negativamente** al planeta hasta el punto en que prácticamente todas nuestras actividades cotidianas están supeditadas al uso de algún componente electrónico. SmartPhones, Tablets, SmartWatches o PCs son el ejemplo más claro de ello, pero esta evolución también tiene un punto positivo y es que nos ofrece herramientas y tecnologías que nos pueden ayudar a buscar soluciones a los problemas ambientales que el propio ser humano está causando. En este sentido la **Inteligencia Artificial (IA)** y en concreto el **Machine Learning** nos ofrece poderosas herramientas para **ayudarnos a optimizar** los procesos industriales y predecir posibles comportamientos que nos ayuden a **controlar los residuos** generados.

Si bien el término Inteligencia Artificial fue acuñado en 1956 por John McCarthy y viene evolucionando desde entonces, hay varias técnicas asociadas a la misma que han mejorado exponencialmente en los últimos años y que son vitales para nutrir un sistema de IA capaz de ayudarnos en estos retos tan complejos:

1. **Big Data**: Son una serie de técnicas utilizadas para el procesamiento de grandes volúmenes de información. Un análisis correcto de los datos puede llevar a la toma de mejores decisiones, y en el caso del medio ambiente, estas decisiones estarán relacionadas con su conservación.
2. **Machine Learning**: Son un conjunto de técnicas que, a partir del procesamiento de grandes volúmenes de información, permiten la creación de algoritmos que aprenden automáticamente de los datos y son capaces de predecir comportamientos futuros o determinar patrones.

4. Justificación

Uno de los sectores más importantes a la hora de ayudar a preservar el medio ambiente es el sector de recogida de residuos, que se encarga no solo de recoger los desechos generados, sino que deberá conseguir reciclar el mayor porcentaje de residuos posible a fin de garantizar la sostenibilidad de los recursos naturales y la reutilización de materias primas.

Hoy en día la gestión de residuos tiene un **impacto económico** en la **administración** de distintas formas, donde las principales son:

1. **Contratación del servicio**: Implica la contratación mediante la licitación pública del servicio de recogida de residuos y conlleva una enorme cantidad de gasto en contratación de personal o adquisición de vehículos y tecnología de seguimiento entre otros. De hecho, para los Ayuntamientos la recogida de residuos es el servicio que más porcentaje ocupa dentro del presupuesto municipal, situándose generalmente en una horquilla de entre el **15 y el 20 % del presupuesto anual**.
2. **Disposición de residuos**: Los residuos generados por la población serán enviados a plantas de reciclaje (en la medida en que el ciudadano colabore en la separación de las distintas fracciones) donde la administración recupera parte del dinero invertido por la venta de ese residuo o a vertederos, donde deberá pagar por depositarlo.

La correcta gestión de residuos tiene además una gran **importancia medioambiental** que incide directamente en beneficios para el medio ambiente, el clima y la salud humana. Tanto es así que el 18/08/2018 el **Parlamento Europeo**

aprobó cuatro propuestas legislativas para aumentar las tasas de reciclaje entre las que destaca un objetivo para **2025** donde al menos el **55% de los residuos municipales deberán reciclarse**. El objetivo se irá incrementando al 60% en 2030 y al 65% en 2035 *(Directiva (UE) 2018/851)*.

España se encuentra lejos de los objetivos marcados, el último informe de Eurostat *(Eurostat, 2021)* indica en 2019 únicamente se recicló un 34,7% de los residuos municipales (residuos que, además del ahorro medioambiental, suponen un ahorro económico por la venta a las plantas de reciclaje) mientras que el resto termina en vertederos o incinerado, residuos que además de los consiguientes problemas para el medioambiente generan mayor gasto para la administración.

La gestión de los residuos es pues uno de los principales retos a los que se enfrenta el ser humano desde el punto de vista medioambiental. Afortunadamente, en las últimas décadas se han producido importantes avances tanto desde el punto de vista técnico como de concienciación social en cuestiones como por ejemplo el reciclaje. Sin embargo, queda mucho por hacer.

La creciente sensorización tanto de la maquinaria como de los propios contenedores de residuos es un paso más en este avance, generando millones de datos por segundo de información relevante que viajan a través del Internet de las Cosas (IoT). Este diagnóstico es común no solo al sector de la gestión de residuos sino a cualquier otro en la actualidad. Se dispone de grandes cantidades de datos pero no se les saca todo el provecho posible. Aquí es donde el desarrollo de **modelos de *machine learning*, diseñados *ad hoc* para dar respuesta a retos concretos** del sector pueden suponer una mejora tecnológica que ayude en esos procesos de optimización tan importantes.

Hoy en día no es suficiente con hacer las cosas *medianamente bien*, hay que conjugar la viabilidad económica con un buen servicio al ciudadano y el respeto al medioambiente. Para ello las empresas gestoras de residuos deben buscar la máxima eficiencia pero respetando las citadas ligaduras sociales y medioambientales. Un ejemplo claro es la recogida de los contenedores en el momento adecuado. Si los contenedores rebosan y permanecen tiempo sin ser recogidos, generarán problemas como malos olores, incluso posibles problemas de salud, y redundará negativamente

en la concienciación social, "*¿para qué voy a molestarme en reciclar si, total, luego no lo recogen?"*. Por otro lado, recogidas innecesarias cuando los contenedores están demasiado vacíos, generan un mayor coste a la empresa (o administración) y una contaminación extra (ruidos, CO_2 de los camiones...) que hubiese sido evitable. Conseguir el equilibrio entre estas dos situaciones es complejo, y requiere el desarrollo de modelos computacionales que ayuden a lograrlo.

Sin embargo, es todavía escasa la implantación de este tipo de modelos en el software de gestión de residuos utilizado realmente, en producción, por las empresas gestoras y las administraciones públicas. Ha sido en los últimos años cuando se ha empezado a producir una importante sensorización de los distintos elementos del proceso de gestión de residuos, como la realizada por Distromel S.A. (empresa suministradora de los conjuntos de datos utilizados en este trabajo de fin de master) en muchos de sus clientes, por lo que es ahora cuando se dispone de suficientes datos para entrenar modelos de machine learning con el objetivo de lograr una gestión más eficiente. Por tanto, las empresas que dediquen esfuerzos al desarrollo de este tipo de modelos en los próximos años obtendrán un ahorro económico incrementando su productividad y contribuyendo al cuidado del medioambiente en un plazo más próximo.

Siguiendo con el caso de la recogida eficiente, los sistemas de pesaje embarcados en los vehículos así como los sensores de llenado instalados en los contenedores permiten conocer, con un cierto margen de error, el porcentaje de llenado que presenta el contenedor en un momento dado (mediante los sistemas de pesaje en el momento de la recogida y mediante los sistemas de medición de llenado en momentos programados del día). El análisis del histórico de estos datos será utilizado para desarrollar modelos de previsión de llenado a partir de los cuales se podrá planificar una recogida eficiente de los residuos.

Históricamente la administración ha licitado los servicios de recogida de residuos sólidos urbanos forzando al contratista a cumplir unas frecuencias de recogida fijas por fracción. Esto tiene sentido en determinadas fracciones que sufren mucha degradación como la orgánica que no puede pasar más de 24 horas en un contenedor por los olores y la "atracción" de roedores a los contenedores pero no en otras como por ejemplo envases, papel/cartón o vidrio, fracciones en las que el residuo

no degrada y sobre las que se podrían mejorar las frecuencias si la administración fuese más flexible en este sentido, algo que los contratistas llevan años reclamando.

Dentro todas las fracciones quizás la que más posibilidades de optimización tiene es el servicio de recogida de vidrio, donde los contenedores tienden a tener más capacidad que el resto de fracciones y que además tardan más en llenarse que el resto. Esto hace que muchas veces debido a esas frecuencias fijadas por la administración los contenedores se vacíen con porcentajes de llenado muy bajos. La existencia de un sistema capaz de indicar al contratista cuando debe ir a recoger vaciar un contenedor implicaría un importante ahorro económico para la empresa, que podrá utilizar únicamente los medios necesarios y no sobredimensionar el servicio. Este ahorro implica a su vez que la administración podrá licitar los servicios de recogida por un importe más ajustado lo que redundará evidentemente en un ahorro económico para el ciudadano. Además, no debemos olvidar que ese ahorro en los medios implicará a su vez menos emisiones de CO2 a causa de los vehículos con lo que se reducirá la huella de carbono generada por el contratista, menos tráfico en las calles, menos ruido y por consiguiente un "*ahorro medioambiental*".

En resumen, la IA ha traído muchos cambios al mundo de los negocios y gestión de las administraciones públicas, y muchos cambios están en camino. Con esto en mente, dado el contexto, en el actual entorno dinámico y cambiante en el que se encuentra la gestión de residuos y considerando el potencial derivado del uso de la IA, la aplicación de técnicas de machine learning para la comprensión y optimización de los servicios de gestión de residuos es clave en la creación de valor.

5. Contexto y estado del arte

Dada la importancia del problema de la gestión de residuos existen una serie de trabajos previos en los que se prueban distintas técnicas para la predicción de la generación de residuos municipales, de hecho, entre 1970 y 2014 aparecieron más de 80 estudios en los que se estudian métodos de predicción que podríamos clasificar en cinco grandes categorías *(Abbasi, M. et al, 2016)*:

1. Basados en métodos estadísticos descriptivos
2. Basados en técnicas de regresión
3. Basados en modelos de flujo

4. Series temporales
5. Métodos de inteligencia artificial

Los basados en métodos estadísticos descriptivos generalmente utilizan el crecimiento de la población y la renta media per-cápita como principales predictores *(Abdoli et al, 2012)*. Estos métodos, sin embargo, no son efectivos debido a las características dinámicas de la generación de residuos.

Los basados en análisis regresivos son modelos ampliamente utilizados debido a la simplicidad de las matemáticas subyacentes y la teoría estadística tan estudiada. En el análisis de regresión, la generación de residuos se asocia con variables económicas y demográficas de manera muy similar a los métodos estadísticos descriptivos *(Abdoli et al, 2012)*.

Los modelos de flujo permiten caracterizar completamente las propiedades dinámicas en el proceso de generación de residuos sólidos. Este enfoque de modelado se puede utilizar para predecir los desechos totales en lugar de los desechos recolectados *(Beigl et al., 2008)*. Aun así, podríamos ser capaces de estimar los residuos recogidos si conociésemos las tasas de reciclaje. Sin embargo, *(Hockett et al. (1995))* después de analizar los resultados obtenidos con uno de estos modelos, destacó que las comparaciones de los resultados con los datos reales de residuos observados en los niveles de agregación más altos eran cuestionables debido a la presencia de diferentes agregaciones o debido a la baja consistencia dentro de la estudios.

A diferencia de los métodos anteriores, los modelos basados en series temporales no se basan en la estimación de factores socioeconómicos, por tanto, tienen la ventaja de no necesitar la información de parámetros sociales, económicos u otros predictores. Los datos de series de tiempo de generación de residuos son de naturaleza dinámica y es posible utilizar técnicas no lineales para discernir relaciones dentro de la serie de tiempo *(Navarro, J. 2002)*.

Con los modelos basados en técnicas de inteligencia artificial, se ha demostrado que son capaces de predecir la generación de residuos a largo, medio y corto plazo *(Abbasi, M. et al, 2016)* o *(Abdoli et al, 2012)*. En los últimos años estos modelos han ganado popularidad, probando distintas técnicas para la predicción de la

generación de residuos como máquinas de soporte vectorial (SVM), sistemas adaptativos de inferencia neuro-difusa (ANFIS) y redes neuronales artificiales (ANN), con las que podemos construir sistemas complejos no lineales que hacen que sean candidatos ideales para la predicción de la generación de residuos como demuestra Abbasi en su artículo *(Abbasi, M. et al, 2016)*.

En las siguientes secciones veremos ejemplos concretos de algunos de estos modelos que aparecen en la literatura.

5.1. Redes Neuronales Artificiales (RNN)

Los modelos basados en redes neuronales artificiales son candidatos ideales para la predicción de residuos debido a la capacidad de construir con ellas sistemas complejos no lineales. *(Kannangara et al, 2018)* utiliza un modelo de red neuronal para hacer un estudio sobre la predicción de generación de residuo en un área de Canadá en el que utiliza una serie de parámetros socioeconómicos como predictores

- Fracción de población mayor de 45 años
- Ingresos medios per cápita
- Fracción de la población sin estudios superiores
- Ratio de desempleo
- Fracción de personas empleadas en agricultura, recursos, industria o construcción
- Fracción de personas con vivienda propia (versus alquileres)
- Fracción de hogares de una persona
- Fracción de población con un trabajo estable

En su estudio, cruza los datos de la generación de residuos entre los años 2002 y 2014 con los datos socio-económicos mencionados anteriormente para buscar correlaciones entre ellos y definir un modelo de aprendizaje automático a partir de los mismos. En la *Ilustración 1* podemos ver un esquema general del modelo de estudio.

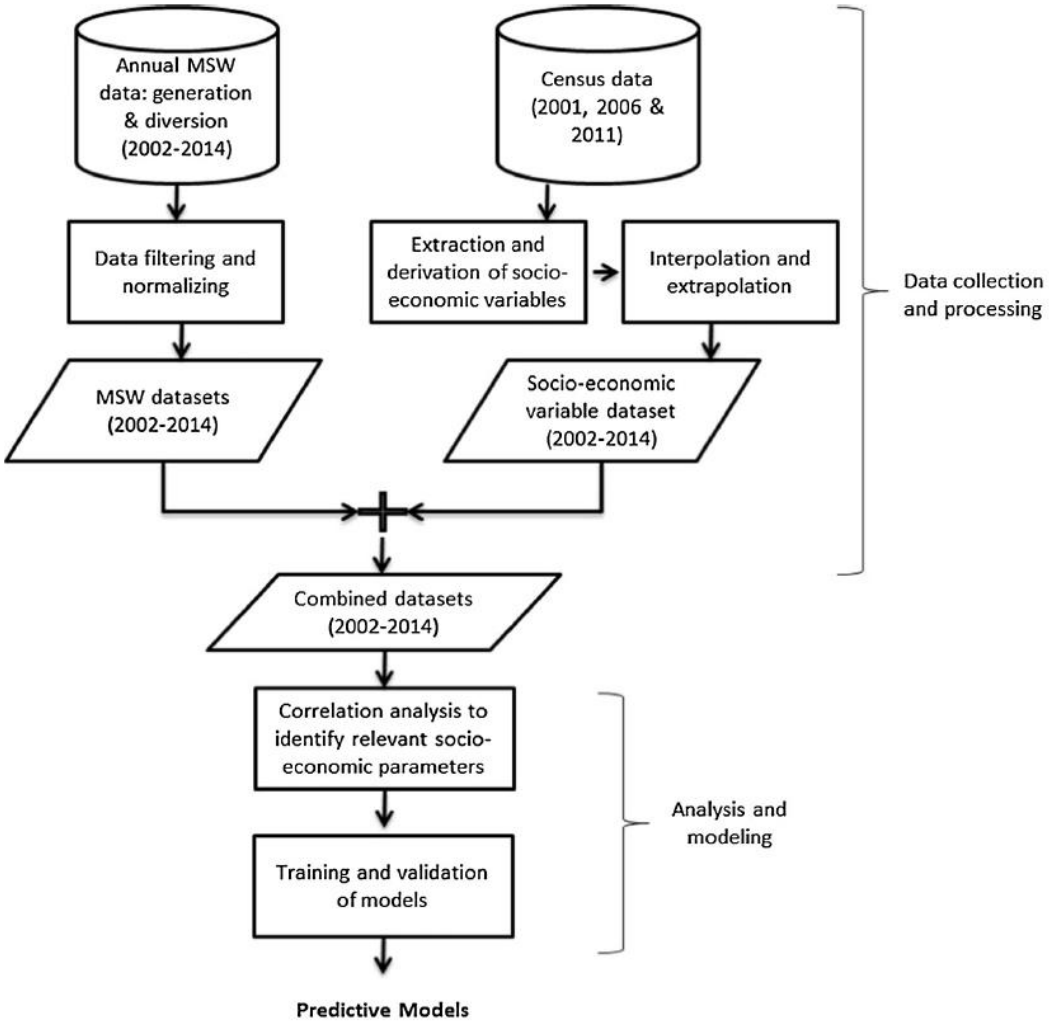


Ilustración 1. Esquema general del modelo de estudio (Kannangara et al, 2018)

Mediante la red neuronal definida en el estudio se obtuvieron errores porcentuales medios en torno al 16% para los residuos municipales, sin embargo, para el papel el error aumentó a valores entre el 32-36%. Los autores piensan que esto es debido a que con las variables socio-económicas utilizadas no se captura el comportamiento de reciclaje. En la *Tabla 1* podemos ver los resultados del estudio donde se comparan las métricas para papel y el resto de residuos municipales.

Modelo	Entrenamiento			Test		
	$\frac{RMSE}{y}(\%)$	MAPE(%)	R^2	$\frac{RMSE}{y}(\%)$	MAPE(%)	R^2
Residuos sólidos urbanos	18	15	0.83	20	16	0.72
Papel	27	26	0.56	32	32	0.35

Tabla 1. Resultados de RNN (Kannangara et al, 2018)

Estos datos demuestran la importancia de escoger las suficientes y adecuadas variables explicativas para obtener precisiones altas en la predicción de generación de residuos sólidos urbanos. Además, pese a sus capacidades de predicción, su rendimiento suele disminuir debido a la tendencia de las RNN de sobre ajuste en el conjunto de entrenamiento, mínimos locales y pobre generalización *(Abbasi, M. et al, 2016)*.

5.2. Sistemas adaptativos de inferencia neuro-difusa

Los sistemas adaptativos de inferencia neuro-difusa (ANFIS) son una técnica de modelado basada en datos que combina ANN y lógica difusa. Los sistemas ANFIS se componen de dos partes, antecedente y conclusión, que están conectadas entre sí mediante reglas difusas basadas en la forma de la red.

(Abbasi, M. et al, 2016) encuentra en la literatura tan solo 3 intentos para predecir la generación de residuos usando ANFIS, en ellos se comparara el rendimiento de la capacidad de los modelos ANN y ANFIS para predecir la generación de residuos sólidos urbanos.

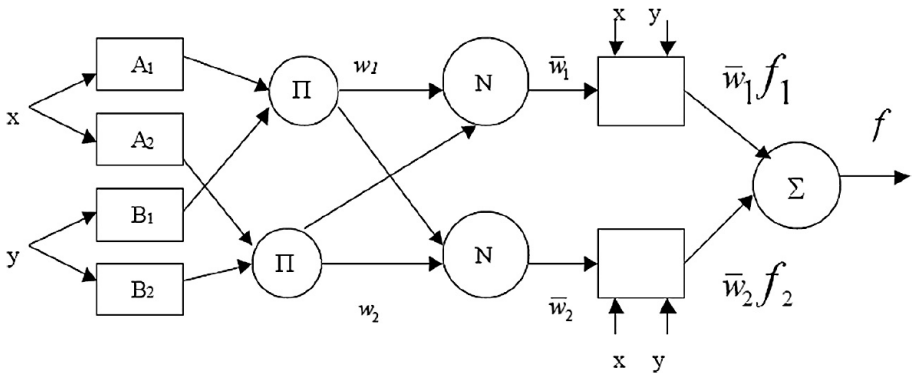


Ilustración 2. Estructura de la red ANFIS (Abbasi, M. et al, 2016)

En la *Ilustración 2* podemos ver la estructura de una red ANFIS compuesta por dos entradas, una salida y dos reglas que son:

$$Si\ x\ es\ A_1\ e\ y\ es\ B_1, entonces\ f_1 = p_1x + q_1y + r_1$$

$$Si\ x\ es\ A_2\ e\ y\ es\ B_2, entonces\ f_2 = p_2x + q_2y + r_2$$

A_i y B_i son conjuntos difusos, f_i es la salida dentro de la región difusa especificada por la regla difusa, p_i, q_i y r_i son los parámetros de diseño que se determinan durante el proceso de entrenamiento.

(Abbasi, M. et al, 2016) utiliza estos modelos para predecir la generación de residuos en la ciudad de Logan (Australia) basados en datos sociodemográficos obteniendo unas métricas:

R^2	0.98
MAE	52.16
MSE	175.88
MAPE	0.008

Tabla 2. Resultados ANFIS (Abbasi, M. et al, 2016)

En los estudios se concluye que los modelos ANFIS son más confiables que las ANN para pronósticos de generación de residuos con volúmenes de datos reducidos. El coeficiente de determinación es muy próximo a 1, lo que significa que el ajuste es muy bueno. El MAPE es prácticamente 0, indica que el modelo tiene un error porcentual medio de 0.8% por lo que su precisión será de un 99.2%.

5.3. Máquinas de soporte vectorial

(Abbasi et al., 2013) utiliza las SVM para predecir la generación semanal de residuos en la ciudad de Teherán (Irán). En el estudio concluye que las SVM se pueden utilizar para predecir la generación de residuos a corto plazo con una exactitud razonable, por lo que en el artículo *(Abbasi et al., 2016)* las vuelve a utilizar para comparar distintos modelos de inteligencia artificial para la predicción de residuos de la ciudad de Logan (Australia), en ambos casos los modelos utilizan datos sociodemográficos como predictores.

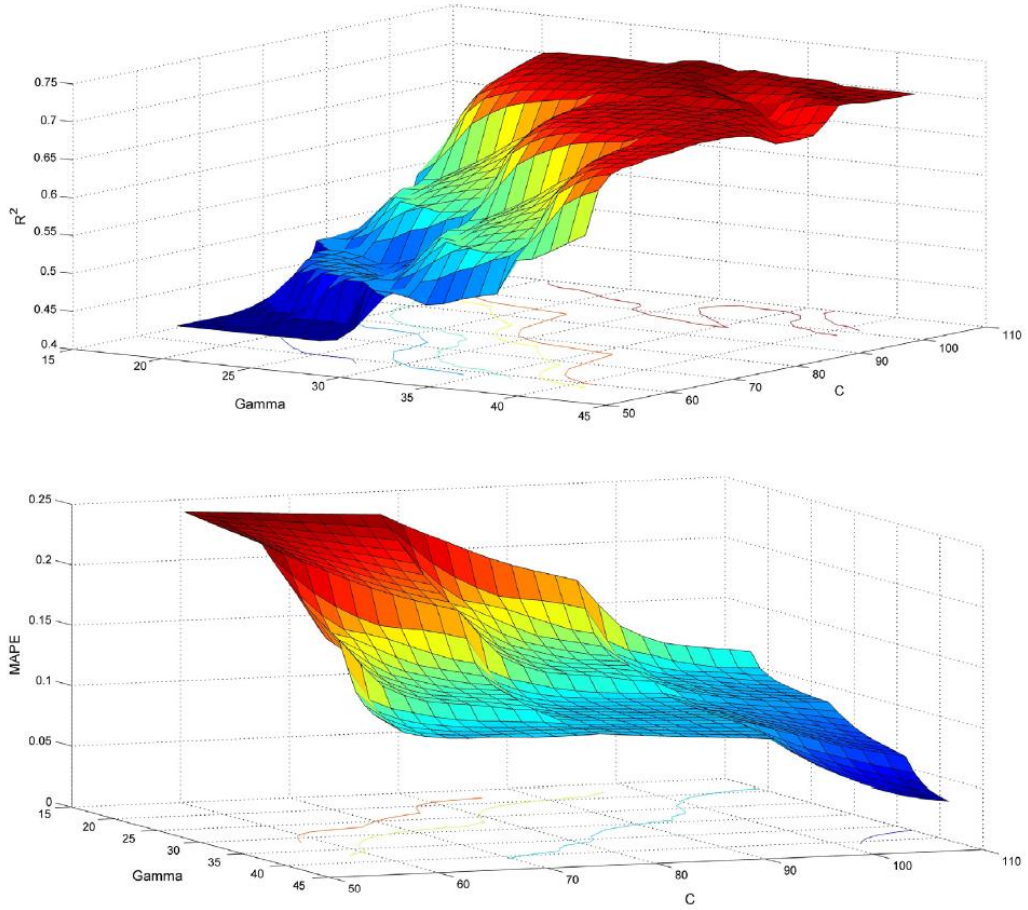


Ilustración 3. Valores MAPE y R^2 para distintas configuraciones de SVM

Para la optimización de los hiperparámetros *C, γ, ε* utiliza el método grid search que permite hacer un ajuste de cada uno de los parámetros en distintos rangos de valores buscando la mejor configuración. En la *Ilustración 3* nos muestra en un formato gráfico los distintos valores de R^2 y MAPE para los ajustes de los hiperparámetros mediante grid search.

5.4. KNN (K-nearest neighbours)

Debido a su simplicidad e ingenuidad, el algoritmo KNN ha sido ampliamente utilizado para tareas de regresión y clasificación.

La intuición subyacente para aplicar KNN a series de tiempo univariables (de una única variable) es que los procesos consistentes de generación de datos a menudo producen observaciones de patrones repetidos de comportamiento. Por lo tanto, si un patrón anterior puede identificarse como similar al comportamiento actual de la serie de tiempo, el comportamiento posterior del patrón anterior puede

proporcionar información valiosa para predecir el comportamiento en el futuro inmediato *(Abbasi et al., 2016).*

La idea por la que Abbasi utiliza este método para pronosticar la generación de residuos, es que dado un patrón cuyo valor futuro se va a predecir, el algoritmo identifica los k patrones pasados más similares y los combina, cabe decir, que hasta la fecha del artículo no se había utilizado esta técnica para este tipo de pronósticos.

Finalmente, *(Abbasi et al., 2016)* muestra en la *Tabla 3* un resumen donde compara los distintos algoritmos utilizados para el pronosticar la generación de residuos en la ciudad de Logan y las métricas obtenidas en cada uno de ellos.

	Entrenamiento				Test			
	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2
ANFIS	0.001	0.002	33.39E-06	0.99	52.16	175.18	0.008	0.98
SVM	203.03	300.70	0.05	0.93	206.42	231.99	0.033	0.71
ANN	335.03	498.43	0.07	0.83	226.50	290.55	0.037	0.46
KNN	251.80	387.21	0.06	0.88	250	308.19	0.040	0.51

Tabla 3. Comparación de métricas y modelos de inteligencia artificial (Abbasi et al., 2016).

En base en los resultados obtenidos del estudio, concluye que los modelos KNN, ANFIS y SVM pueden ofrecer buenos resultados y se pueden aplicar para establecer los modelos de predicción que podrían proporcionar un pronóstico preciso y confiable de generación de residuos sólidos urbanos. Además, los resultados sugieren que el modelo ANFIS produjo resultados más precisos que KNN y SVM ya que tiene un error porcentual absoluto medio (MAPE) más próximo a 0 que los otros modelos.

5.5. Series temporales

Todos los modelos presentados hasta ahora están basados en datos demográficos y factores socioeconómicos, sin embargo las series temporales no necesitan esta información ya que se basan en los datos medios de generación de residuos. En *(Navarro, J. 2002)* propone algunas herramientas de análisis de

predicción de series temporales para estudiar la generación de residuos sólidos urbanos. Propone una técnica de predicción basada en dinámicas no lineales y la compara con el rendimiento obtenido con el algoritmo de media móvil autorregresiva estacional (sARIMA).

En su estudio, Navarro nos explica que el modelo sARIMA está basado en la aplicación del modelo ARMA a series temporales transformadas donde los comportamientos estacionales y no estacionarios han sido eliminados. Si $\{z_t\}$ es la serie temporal original, después de algunas transformaciones se puede obtener una serie $\{x^t\}$ no estacional y estacionaria. Entonces, esta serie temporal transformada se puede ajustar con el modelo ARMA donde el valor actual de la serie temporal, x_t, se expresa como una agregación lineal de los *p* valores previos y una suma ponderada de *q* desviaciones previas (valor original menos valor ajustado de los datos anteriores) más un proceso aleatorio ε_t.

Como explica Navarro en su artículo *(Navarro, J. 2002), l*os datos de generación de residuos sólidos urbanos, suelen mostrar con frecuencia comportamientos ciclos y/o estacionales que se pueden identificar fácilmente con la función de autocorrelación (ACF) *(Box et al, 1976)*. Para mejorar los resultados de las previsiones con modelos ARMA conviene eliminar este comportamiento.

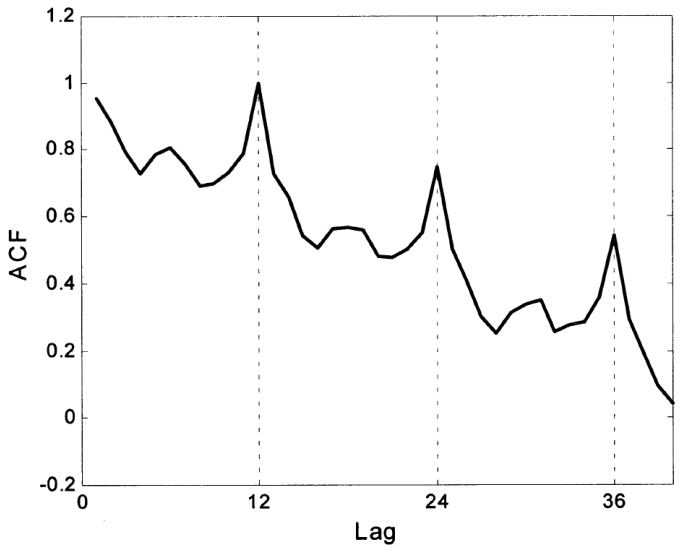


Ilustración 4. Estacionalidad en serie temporal (Navarro et al. 2002)

En la *Ilustración 4 (Navarro, J. 2002)* presenta el gráfico ACF de una serie de tiempo donde se puede observar que se detecta claramente la periodicidad estacional

donde los picos presentados en el ACF muestran una correlación en los datos cada 12 unidades de retardo. Tras eliminar este comportamiento, *(Navarro, J. 2002)* explica cómo calcular los coeficientes *(p, q)* del modelo ARMA a partir del criterio AICC que es uno de los más utilizados así como la definición de las funciones matemáticas para el modelo de predicción y el modelo no lineal que propone para comparar con sARIMA, que tiene la ventaja de que se puede aplicar directamente sobre los datos originales ya que no necesita que estos sean no estacionales.

En su estudio se centra en datos de dos ciudades Españolas y una Griega y aunque los resultados de los dos modelos son bastante buenos para predecir la cantidad de residuos generados diariamente o mensualmente en esas ciudades en general el modelo sARIMA se comporta mejor que el modelo no lineal como se puede apreciar en la *Ilustración 5*.

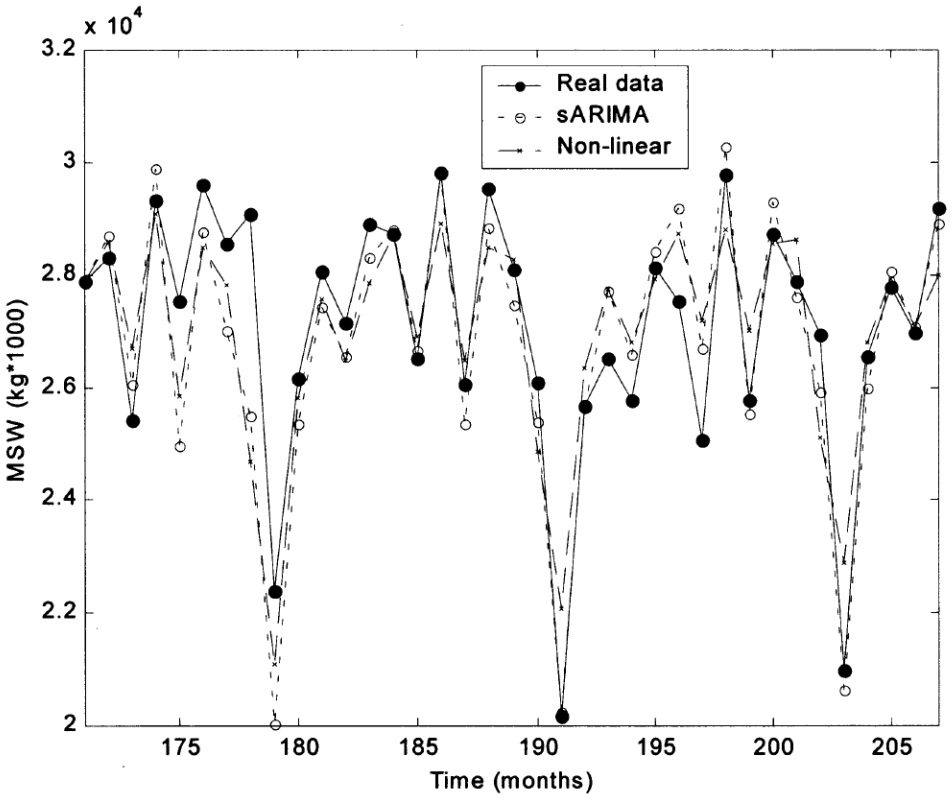


Ilustración 5. Comparación modelo sARIMA vs no-lineal (Navarro et al. 2002)

El estudio concluye comentando que los modelos permiten predecir la cantidad de residuos generados con un error relativo medio inferior al 10% en las próximas 2 semanas y que disminuye al 5% al realizar predicciones a 2 y 3 años.

5.6. Conclusiones

A lo largo de la historia se ha estudiado ampliamente la predicción de la generación de residuos sólidos urbanos y en los apartados anteriores se ha hecho un breve repaso de los mismos. Hemos visto que todos los modelos, a excepción del de series temporales utilizan datos demográficos y factores socioeconómicos para realizar las predicciones y que todos ellos tienen algo en común, realizan predicciones de la generación de residuos a nivel de áreas urbanas (barrios, municipios o ciudades).

Este tipo de predicciones resultan de gran interés al gestor de residuos en pro de calcular los medios necesarios para la prestación del servicio a lo largo del tiempo, pero no le ayudan a planificar las rutas diarias de recogida de residuos ya que le ofrecen información totalizada y no granular sobre los puntos de recogida que deberán ser recogidos diariamente para evitar desbordes, recogidas innecesarias de contenedores y en definitiva optimizar las rutas de recogida de residuos.

En este trabajo de fin de master, nuestro objetivo será el estudio de técnicas o modelos que permitan pronosticar la generación de residuos a nivel de punto de recogida, de forma que el gestor, pueda saber el momento óptimo para ir a vaciarlo. La ventaja de este modelo es que, además de ofrecer información total de la predicción de residuos de la ciudad (como suma de la predicción de cada uno de sus puntos de recogida), permite optimizar las rutas realizadas por los camiones, algo que con los otros modelos no era posible y que como hemos visto en el punto *4* significará un importante ahorro económico y medioambiental.

Para ello, nos centraremos en el uso de series temporales ya que el dataset que nos suministrará la empresa para el estudio se ajusta perfectamente a un modelo de **serie temporal univariante** sin variables exógenas. Para pronosticar las series temporales nos centraremos en el modelo ARIMA y compararemos sus resultados con una sencilla red neuronal creada para pronosticar los valores futuros de la serie.

6. Análisis de series temporales

Datos relativos a distintos tipos de negocios, como economía, ingeniería, medio ambiente, medicina y otras áreas de investigación científica a menudo se recopilan en forma de series de tiempo. Una serie de tiempo es una secuencia de observaciones

tomadas a intervalos regulares de tiempo, como por ejemplo la temperatura por hora, precios diarios de acciones de bolsa, volumen de tráfico semanal, consumo mensual de cerveza o tasas de crecimiento anual. Los principales objetivos del análisis y modelado de series de tiempo son (1) comprender la estructura dinámica o dependiente del tiempo de las observaciones de una sola serie: análisis univariante de series de tiempo y (2) determinar las principales relaciones y retroalimentación entre varias series: análisis de series de tiempo multivariables *(Peña et al, 2001).*

El conocimiento de la estructura dinámica ayudará a producir pronósticos precisos de las observaciones futuras y diseñar esquemas de control óptimos por lo que son de especial relevancia en el ámbito empresarial.

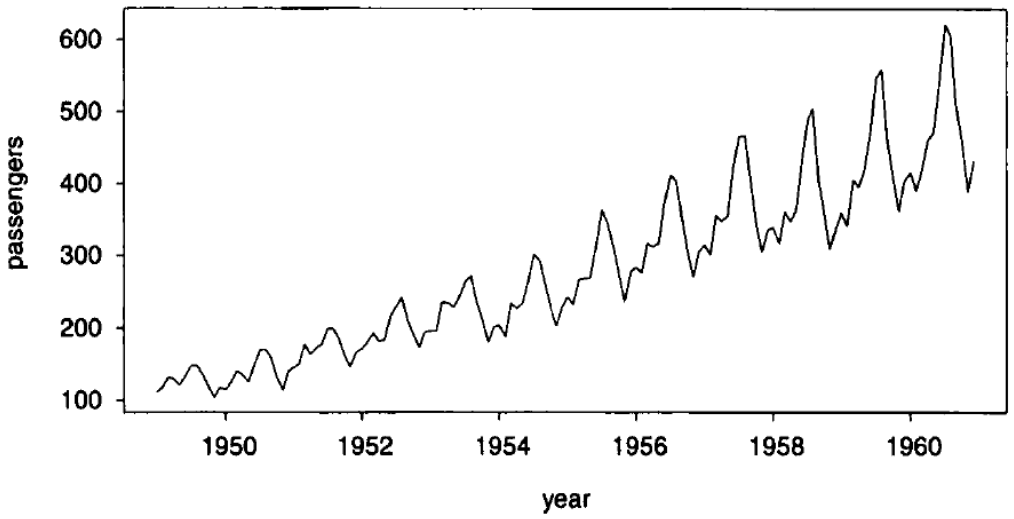


Ilustración 6. Ejemplo de serie temporal (Peña et al, 2001)

En la *Ilustración 6* vemos un ejemplo gráfico de una serie temporal, donde en el eje horizontal se presenta la línea del tiempo y en el eje vertical el valor las observaciones medidas.

En este trabajo nos vamos a centrar en el uso de modelos de series temporales para la predicción de generación de residuos sólidos urbanos por dos motivos, el primero es que no disponemos de información demográfica o socioeconómica, sino solamente los kilogramos de residuo recogido en los contenedores cada vez que se han vaciado y el segundo, porque la información que tendremos se ajusta perfectamente a la definición de una serie temporal (secuencia de observaciones tomada a intervalos regulares en el tiempo).

Por lo tanto, en primer lugar nos centraremos en la explicación teórica de cómo hacer un análisis de una serie temporal.

6.1. Técnicas de visualización de series temporales

Un primer punto de especial importancia cuando empezamos a analizar una serie temporal implica un estudio visual de la misma. Para ello hay una serie de técnicas que podemos aplicar que nos ayudarán a comprender los valores y buscar las mejores variables para aplicar los algoritmos de aprendizaje automático posteriormente.

6.1.1. Visualización de series temporales

Como punto de partida para visualización de series temporales se dibujará un gráfico con donde se representará el tiempo en el eje horizontal *X* y los valores que toma la serie para cada instante de tiempo en el eje horizontal *Y*. En la *Ilustración 7* podemos ver un ejemplo típico de representación de una serie temporal.

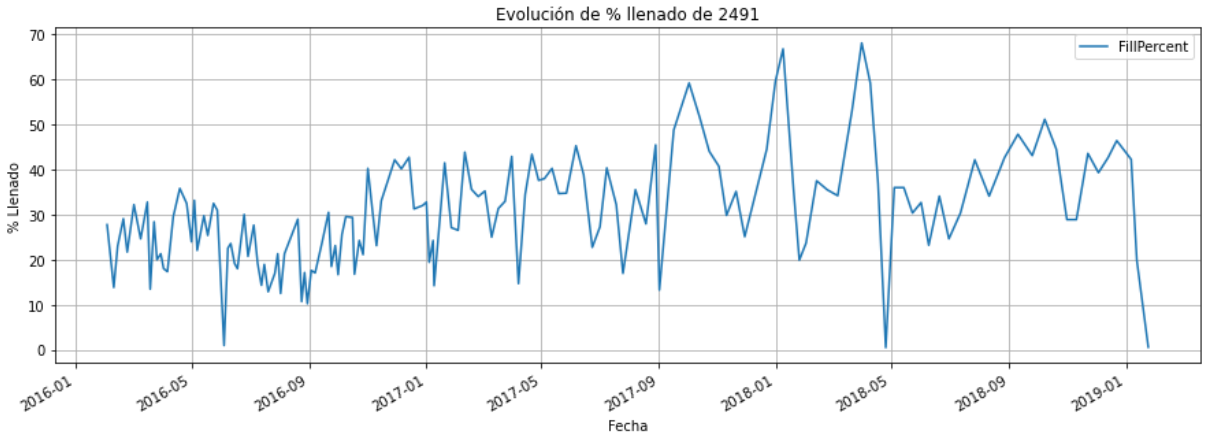


Ilustración 7. Representación de serie temporal

Mediante un gráfico estacional de la serie temporal en el que dibujemos el comportamiento anual de la misma intentaremos identificar precisamente patrones estacionales. En la *Ilustración 8* vemos un gráfico de este tipo en el que no se observan picos en ningún mes por lo que se puede concluir que la serie no presenta este tipo de comportamientos. Supongamos que la serie fuese de la cantidad de juguetes vendidos en un gran almacén por meses, veríamos que para Diciembre y Enero los valores serían mucho más altos que el resto de meses a causa de la campaña navideña por lo que nos resultaría fácilmente identificable esa estacionalidad de forma visual.

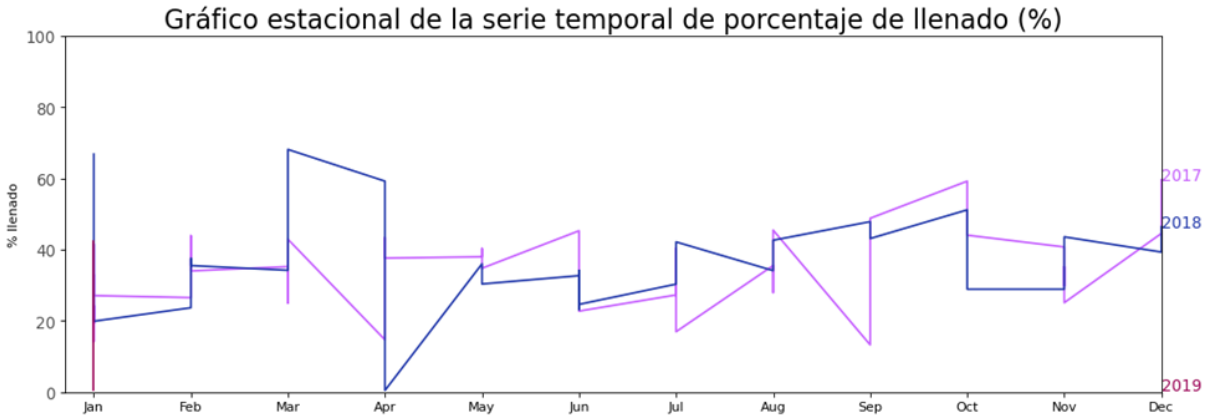


Ilustración 8. Gráfico estacional de series temporales

Aun así, podría haber algún comportamiento que hubiese escapado a nuestro análisis. Otro tipo de gráficos que resultan de utilidad para identificar patrones son los diagramas de cajas y bigotes en los que podemos agrupar los datos en distintas unidades de tiempo (por ejemplo año y por mes). Este tipo de gráficos, hacen evidentes distribuciones en los datos anuales y mensuales. En la *Ilustración 9* podremos ver que en el diagrama anual no se observa ninguna tendencia clara, sin embargo, en el diagrama mensual vemos que para los meses de Octubre y Diciembre la media de los valores es claramente superior al resto, por lo que ha aparecido una estacionalidad que no habíamos detectado en el diagrama anterior (*Ilustración 8*)

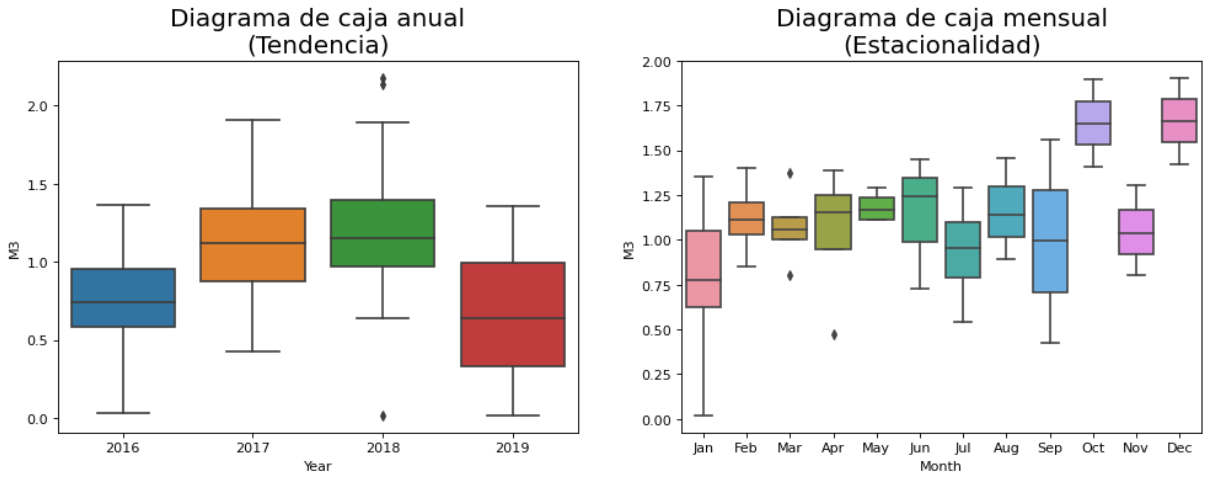
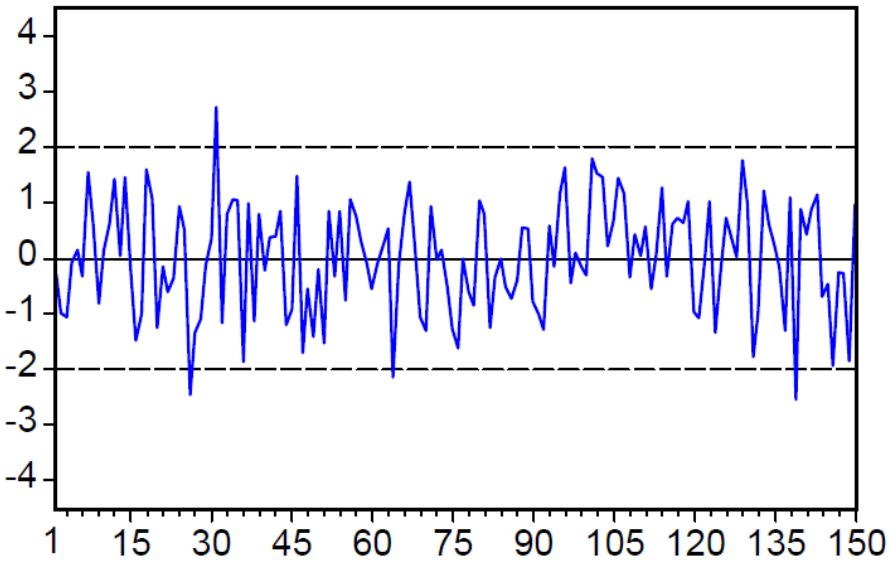


Ilustración 9. Diagrama de cajas de una serie temporal

Es muy importante en todo proceso de análisis de series temporales aplicar distintos tipos de visualizaciones que nos ayuden a comprender cómo se comporta la serie y a identificar patrones o tendencias subyacentes en los datos.

6.1.2. Ruido blanco

Debemos tener cuidado de no confundir ruido blanco con una serie temporal. El ruido blanco no guarda relación con el tiempo, por lo que puede tener propiedades estadísticas similares a una serie temporal pero, el ruido blanco siempre tomará valores aleatorios con una medida de 0. Matemáticamente lo podríamos definir como una secuencia de números aleatorios con media cero.



Ecuación 1. Señal de ruido blanco (Mauricio, J. Alberto, 2007)

6.1.3. Patrones en series temporales

Toda serie temporal puede descomponerse en 4 componentes cuya representación matemática se presenta en la *Ecuación 2*.

$$Valor = Base + Tendencia + Estacionalidad + Error$$

Ecuación 2. Componentes de serie temporal aditiva

En una serie temporal observaremos una tendencia cuando los datos muestran una pendiente creciente o decreciente en el tiempo. Si volvemos a la *Ilustración 6* podemos observar como los valores del eje vertical van creciendo a medida que avanza la línea del tiempo, por lo que existe una tendencia en los datos. Por otro lado, observaremos estacionalidad en una serie de tiempo cuando se aprecien patrones que aparecen a intervalos regulares de tiempo debido a factores estacionales. Siguiendo con la misma gráfica parece que existe un patrón que se repite anualmente, que representaría el componente estacional de la serie.

Una serie temporal no tiene porqué obligatoriamente tener tendencia y estacionalidad, podría tener solamente uno de los componentes, por lo que, la serie temporal no deja de ser una combinación de esos dos componentes más el término de error. En la *Ilustración 10* se presentan distintos tipos de visualización para una misma serie temporal utilizando distintos predictores.

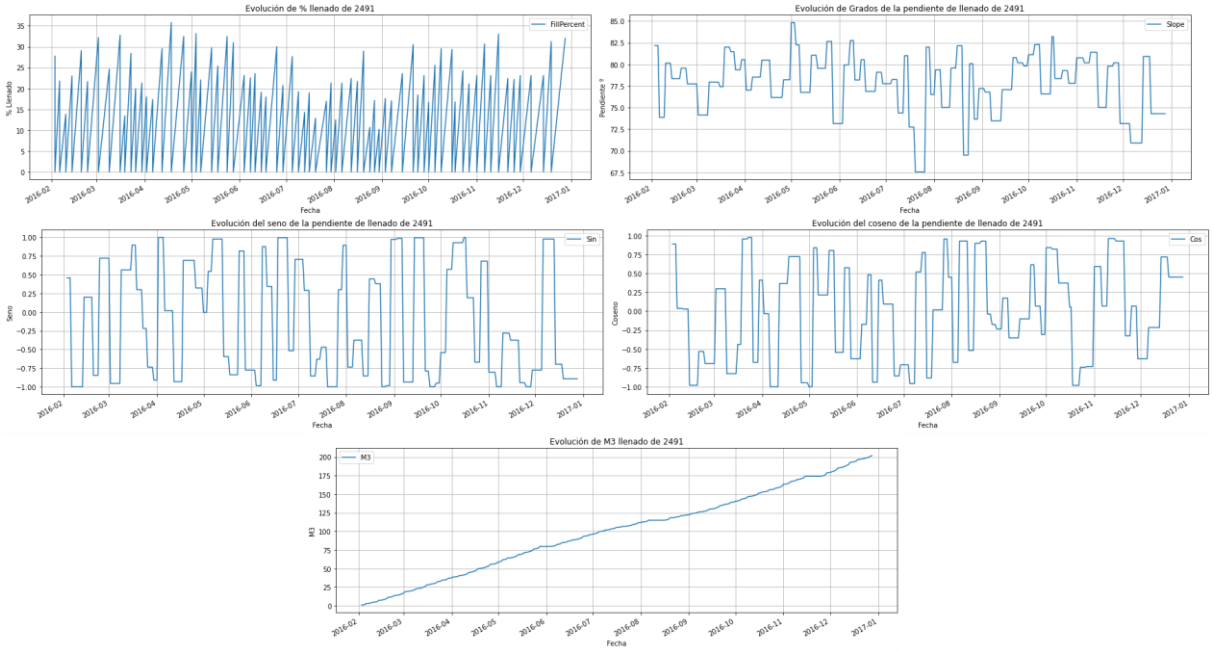


Ilustración 10. Distintas visualizaciones de la misma serie temporal con distintas variables

6.1.4. Autocorrelación y autocorrelación parcial

La autocorrelación es la correlación de una serie con sus propios rezagos. Cuando una serie está significativamente autocorrelacionada los valores anteriores de la serie (rezagos) pueden ser útiles para predecir el valor actual.

La autocorrelación parcial transmite información similar a la anterior, solo que en este caso, transmite la correlación pura de una serie y su rezago anterior, excluyendo las contribuciones de correlación de los rezagos intermedios. Podemos decir que la autocorrelación parcial del rezago (*k*) de una serie es el coeficiente de ese rezago en la ecuación de autorregresión de *Y* (ecuación de autorregresión utilizando los rezagos como predictores).

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \cdots + \alpha_n Y_{t-n}$$

Ecuación 3. Ecuación de autorregresión de Y

A modo de ejemplo, y para entender mejor el concepto de autocorrelación, supongamos que Y_t es una serie temporal y que Y_{t-1} es el retraso 1 de Y, entonces la autocorrelación de retraso 2 (Y_{t-2}) es el coeficiente α_2 en la *Ecuación 3*.

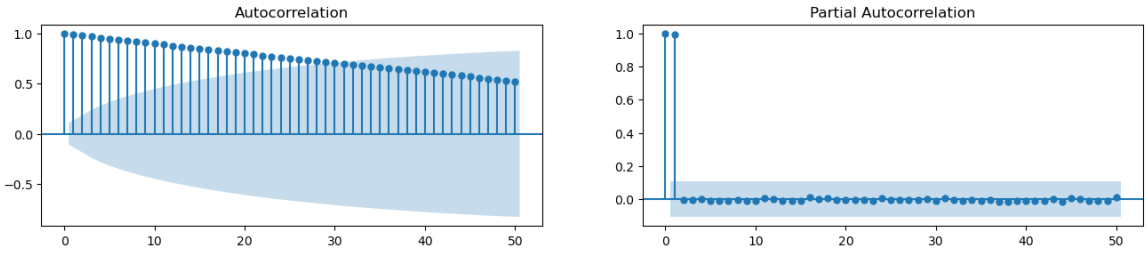


Ilustración 11. Gráficos de autocorrelación y autocorrelación parcial

En el gráfico de la *Ilustración 11*, el cual está limitado a 50 rezagos muestra la correlación con valores en el intervalo -1, 1 (recordar que los valores cercanos a 1 o -1 muestran una fuerte correlación, en cambio los cercanos a cero indican que no hay correlación. En el gráfico también se muestra el intervalo de confianza, por defecto situado en el 95%. Los valores situados por fuera del intervalo de confianza (zona azul) significarán que existe autocorrelación, mientras que si los datos se aproximan a 0, indicará que no la hay.

En la imagen utilizada como ejemplo en este apartado, donde se han dibujado las gráficas de autocorrelación y autocorrelación parcial vemos que en el gráfico de autocorrelación los 30 primeros rezagos muestran correlación, pues están fuera del intervalo de confianza, en cambio los siguientes no. Por el contrario, en el gráfico de autocorrelación parcial vemos que únicamente los dos primeros rezagos muestran esa correlación.

6.1.5. Gráficos de retardo

Los gráficos de retardo o retaso, son gráficos de dispersión de la serie temporal frente a un retraso de sí mismo que se utilizan normalmente para comprobar la autocorrelación. Si existe algún patrón en la serie significará que la serie está autocorrelacionada. En caso contrario, es probable que la serie sea ruido blanco aleatorio.

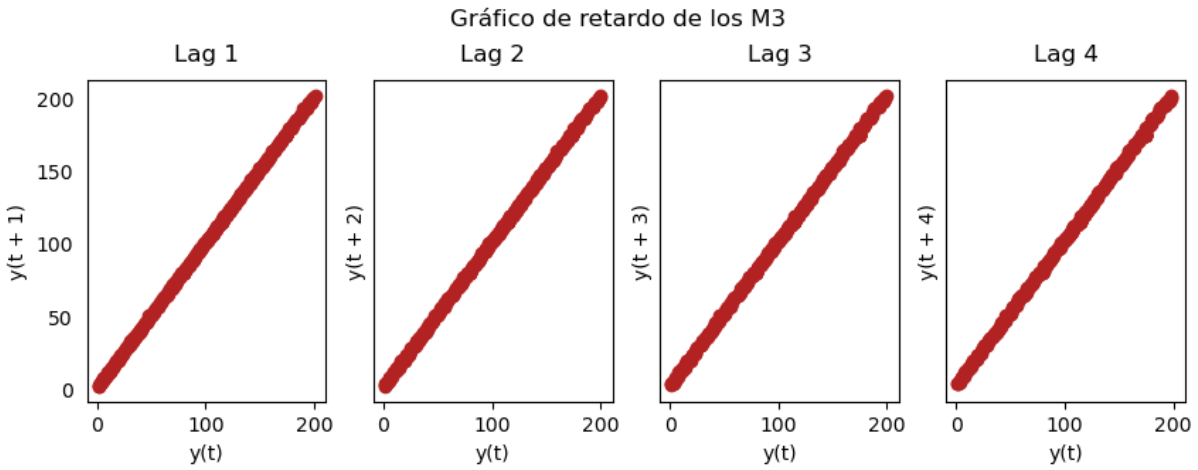


Ilustración 12. Gráfico de retardo de una señal correlacionada

En la *Ilustración 12* se han dibujado los gráficos de retardo de Y_t respecto a sus 4 primeros retrasos (Y_{t-1}, Y_{t-2}, Y_{t-3} e Y_{t-4}), como ya habíamos visto en el apartado *6.1.4* los primeros retardos de la serie están correlacionados y aquí podemos ver claramente un patrón.

En la *Ilustración 13* por el contrario se presentan las gráficas de retardo para una serie no correlacionada. En ella podemos ver que para el primer retardo pese a que se distingue un patrón aparecen puntos muy dispersos y para el cuarto retardo la señal es prácticamente ruido blanco.

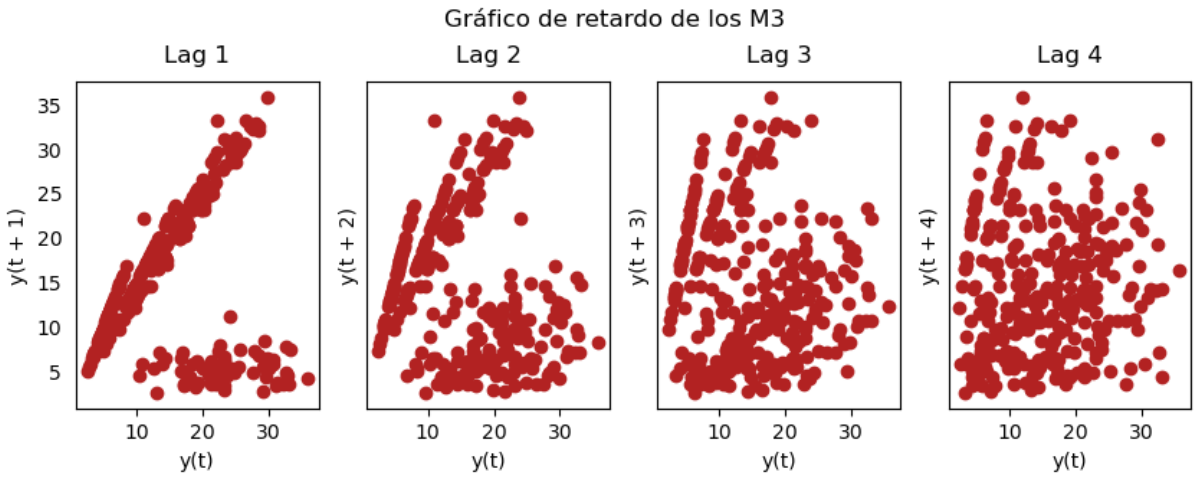


Ilustración 13. Gráficos de una señal no correlacionada

6.2. Caracterización de series temporales

En base a las propiedades de las series temporales se puede establecer una categorización de las mismas, en el siguiente apartado se describen las principales propiedades.

6.2.1. Series temporales aditivas y multiplicativas

Dependiendo de la naturaleza de la tendencia y la estacionalidad de la serie podemos diferenciarla entre una serie aditiva, donde cada observación de la serie puede expresarse como la suma de todos sus componentes o multiplicativa (ver *Ecuación 2*), donde cada observación puede expresarse como el producto de todos sus componentes (ver *Ecuación 4*).

$$Valor = Base * Tendencia * Estacionalidad * Error$$

Ecuación 4. Componentes de serie temporal multiplicativa

6.2.2. Series estacionarias y no estacionarias

La estacionariedad es otra de las propiedades de las series temporales. Una serie estacionaria será aquella en la que sus valores no son dependientes del tiempo, es decir las propiedades estadísticas de la serie como media, varianza y autocorrelación son constantes en el tiempo.

Para el pronóstico de series temporales se suelen utilizar modelos autorregresivos, que son básicamente modelos de regresión que utilizan las observaciones anteriores (rezagos) del valor de salida de la serie para predecir los nuevos valores. Sabemos que la regresión lineal funciona mejor cuando los predictores no están correlacionados entre sí por lo tanto, estacionar la serie resolvería este problema ya que eliminaría cualquier correlación existente en los datos.

6.2.3. Probar estacionariedad

Dada la importancia de la estacionareidad en las series temporales, nos interesará saber si la serie objeto del estudio es o no es estacionaria. Un posible método de prueba, un tanto rudimentario sería dividir la serie en dos o más partes contiguas y calcular para cada una las propiedades estadísticas antes mencionadas

(media, varianza y autocorrelación). Si los valores son muy diferentes entonces será bastante probable que la serie no sea estacionaria.

El segundo método, mucho más eficaz será utilizar una de las llamadas *pruebas de raíz unitaria*. Existen muchas de estas técnicas y quizás las más conocidas son:

- Test de Dickey Fuller Aumentada (ADF)
- Test de Kwiatkowski-Phillips-Schmidt-Shin (KPSS)
- Test de Philips Perron (PP)

De entre ellos el más utilizado es el test ADF, tanto en él como en el test PP la hipótesis nula implica que la serie del tiempo posee una raíz unitaria y no estacionaria, de forma que si el p-valor es menor que el nivel de significancia (0.05) se rechaza la hipótesis nula. En el test KPSS sin embargo la hipótesis nula implica que la serie temporal es estacionaria y en el test PP.

6.2.4. Estacionalidad de las series temporales

La estacionalidad de una serie temporal, como ya se vio en el punto *6.1.3* implica la repetición de patrones a intervalos regulares de tiempo. La forma más común de verificar si existe estacionalidad en una serie temporal consiste en trazar la serie y comprobar si hay patrones repetibles a intervalos regulares de tiempo (horas, días, semanas, meses, etc.) algo similar a lo que se hizo con el gráfico de cajas de la serie (ver *Ilustración 9*). Sin embargo, existe una forma mucho más rigurosa de comprobarlo mediante la función de autocorrelación (ACF) que también explicamos con mayor profundidad en el punto *6.1.4*.

Si observamos la gráfica de autocorrelación de la izquierda en la *Ilustración 14*, no se aprecia ningún patrón, a diferencia de la gráfica de la derecha donde sí que se aparecen picos cada 12 meses. Este tipo de picos suelen ser bastante difíciles de identificar ya que se distorsionan con el ruido y apenas se notan, por lo que hay que prestar especial atención para identificarlos.

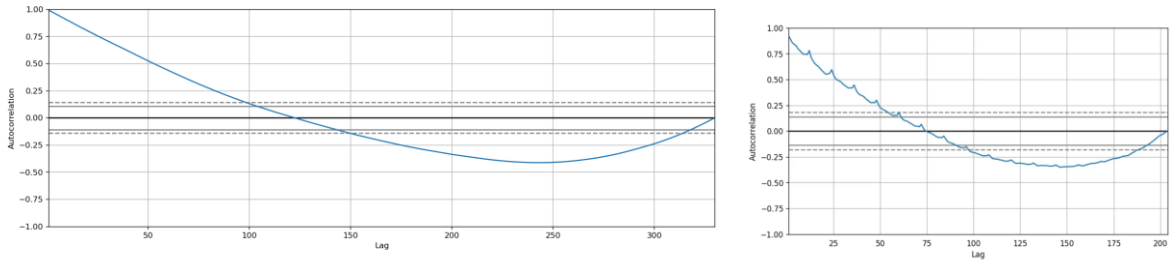


Ilustración 14. Prueba ACF de estacionalidad

Aunque en el gráfico de cajas que habíamos dibujado sí que parecía que se había cierta estacionalidad en los meses de Octubre y Diciembre, a partir del gráfico de autocorrelación no se detectan esos patrones.

Una técnica para eliminar el componente estacional de una serie temporal consiste en dividir los valores de la serie por el componente estacional obtenido a partir de la descomposición de la serie en sus 4 componentes.

Si nuestra serie presentase un componente estacional claro podríamos eliminarlo dividiendo los valores de la serie por la componente estacional.

6.3. Preprocesamiento de series temporales

Una vez vistos las principales técnicas de visualización y las propiedades de las series temporales, podemos entrar en el apartado de preprocesamiento del dataset. Como en todo proceso de machine learning es importante un buen tratamiento de la información para obtener unos buenos resultados. Los pasos más importantes de preprocesamiento en las series temporales se enumeran a continuación.

6.3.1. Valores faltantes de las series temporales

Volviendo a la definición de una serie temporal, se decía que es una serie de valores tomados a intervalos regulares de tiempo. Esto quiere decir, que todos los valores deberían existir en esos intervalos, pero en un entorno real, vamos a encontrarnos con datos faltantes, bien porque no se han tomado, bien porque no se tomaron en el momento adecuado o bien porque no se registraron, por ejemplo, porque el valor era 0 y no se consideró necesario.

Cuando trabajamos con grandes conjuntos de datos y nos enfrentamos a datos faltantes o datos nulos a los que queremos asignar un valor, solemos tomar como estrategia asignar el valor medio a esos valores, pero esta es una solución no válida para las series temporales. Dependiendo de la naturaleza de la misma existirán varios enfoques que pueden sernos útiles y los más comunes son:

- Relleno hacia atrás (backward fill, para el que podemos utilizar el método *bfill* de pandas)
- Relleno hacia delante (Forward fill, para el que podremos utilizar el método *ffill* de pandas)
- Interpolación lineal (para el que podríamos utilizar el método *interp1d* de la librería *scipy.interpolate* con *kind = "lineal"*)
- Interpolación cuadrática (para el que podríamos utilizar, el mismo método que para la lineal pero con *kind = "cubic"*)
- Media de vecinos más cercanos (para el que, como su nombre indica, habría que calcular la media de los k vecinos más cercanos)
- Media de las partes estacionales (para el que, como su nombre indica, se calcularía la medida de sus correspondientes valores estacionales)

6.3.2. Suavizado

No siempre será necesario suavizar una serie temporal, pero es algo que debemos tener muy en cuenta ya que puede resultar muy beneficioso para

- Reducir el ruido. Nos puede ayudar a filtrar el ruido de la serie temporal
- Nos puede ayudar a explicar la propia serie temporal
- Nos puede ayudar a encontrar una tendencia subyacente

Existen distintas estrategias para suavizar una serie temporal y las más utilizadas son

- Media móvil
- LOESS (LOcalized RegrESSion o regresión localizada)
- LOWESS (LOcalized Weighted regrESSion o regresión localizada ponderada)

El suavizado con media móvil es el promedio de una ventana deslizante de ancho definido. En este tipo de suavizado es muy importante escoger bien el ancho de la ventana, pues una ventana muy grande suavizaría demasiado la serie o incluso podría llegar a eliminar completamente el efecto estacional si escogemos una ventana igual a su estacionalidad.

La regresión localizada son regresiones múltiples en las proximidades de cada punto. En el paquete *statsmodels* está implementado el método *lowess* con el que podemos controlar el porcentaje de suavizado mediante el parámetro *frac* que especifica el porcentaje de puntos cercanos que se deben tomar para ajustar el modelo de regresión.

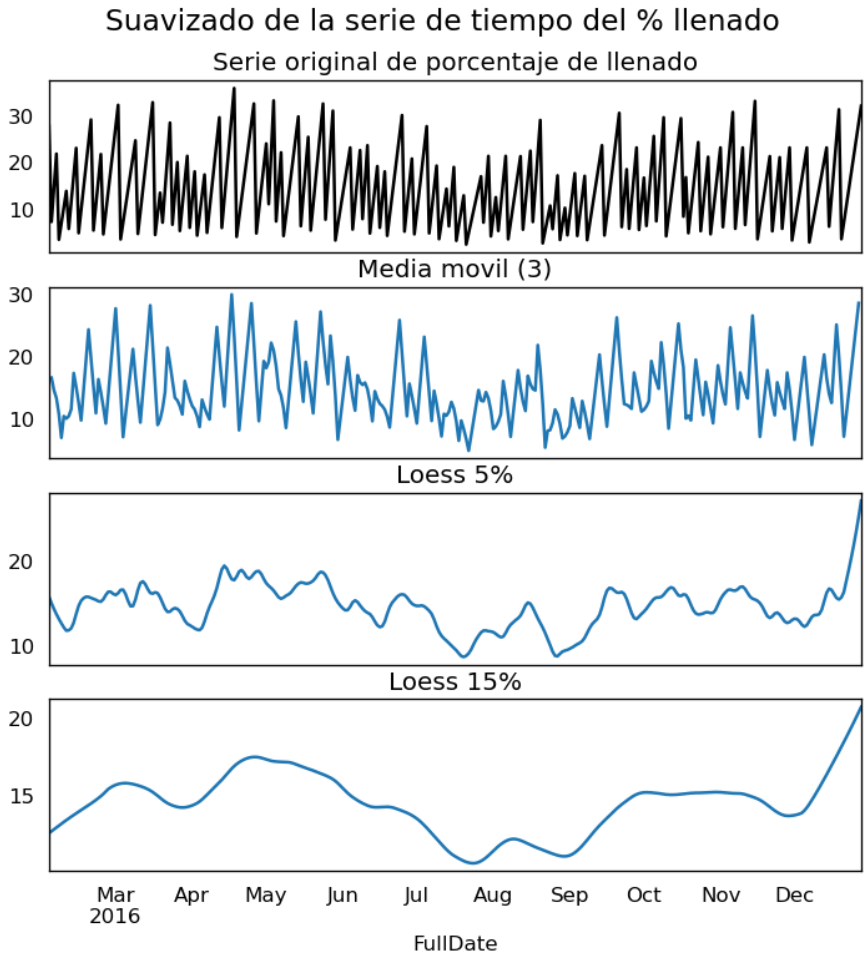


Ilustración 15. Suavizado de la serie temporal con distintas técnicas

En la *Ilustración 15* podemos comparar la serie temporal original (gráfica superior), y las series suavizadas mediante media móvil, lowess al 5% y lowess al 15% respectivamente. En los gráficos resultantes vemos la importancia de escoger

bien los valores, sobre todo, viendo las diferencias entre el suavizado lowess con distintos porcentajes.

6.3.1. Convertir serie temporal en estacionaria

Cuando la serie que estamos evaluando no es estacionaria, debemos convertirla en estacionaria y para ello, el método más común es el de diferenciación. Supongamos que *Y* es un valor para una serie en un instante *t*, la primera diferenciación sería

$$Y = Y_t - Y_{t-1}$$

Ecuación 5. Primera diferenciación

O lo que es lo mismo, la primera diferenciación implica restar a cada valor de *Y* para un instante *t* el valor de Y para el instante inmediatamente inferior.

El procedimiento consistirá en probar el test ADF, si p-value es superior a 0.05 hacer la primera diferenciación y volver a comprobar mediante el test ADF. Si la serie continua sin ser estacionaria podremos hacer diferenciaciones hasta que lo sea.

Supongamos la siguiente serie temporal [1, 3, 4, 6, 9, 14, 20], la primera diferenciación sería [1-3, 4-3, 6-4, 9-6, 14-9, 20-14] = [2, 1, 2, 3, 5, 6]. La segunda diferenciación sería [1-2, 2-1, 3-2, 5-3, 6-5] = [-1, 1, 1, 2, 1] y así sucesivamente.

6.3.2. Descomposición de componentes

Un paso muy importante en el análisis de toda serie temporal es la descomposición de la serie en los 4 componentes (base, tendencia, estacionalidad y error). Esta descomposición ayudará a identificar características difícilmente identificables que pueden ser asignadas a causas concretas, como el comportamiento general a largo plazo, factores que se repiten periódicamente, tendencias, etc.

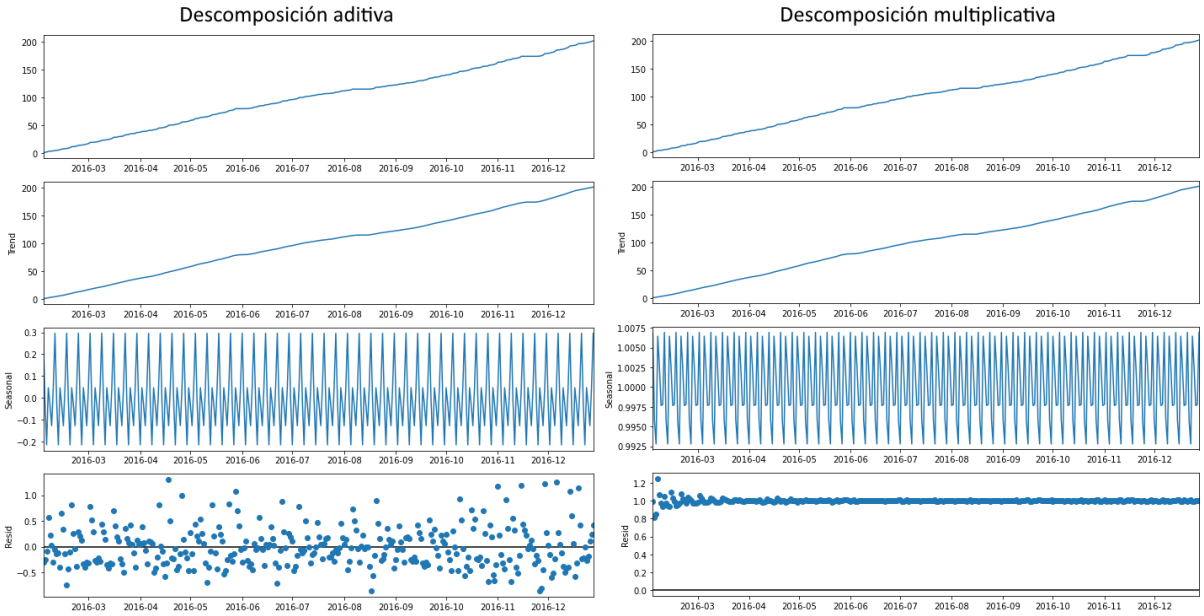


Ilustración 16. Descomposición de componentes de serie temporal

Si observamos las gráficas de la *Ilustración 16* vemos claramente cómo se han separado los componentes de tendencia y estacionalidad con ambas descomposiciones y si nos fijamos en los residuos, en la descomposición aditiva son más aleatorios que en la multiplicativa, lo cual es bueno, por lo que sería en este caso preferible escoger la descomposición aditiva para estudiar el comportamiento de esta serie en particular.

6.3.3. Reducir la tendencia

Como habíamos visto en el punto *6.1.3* la tendencia de una serie temporal supone una pendiente o decreciente de sus valores en el tiempo y es una de las componentes que dan forma a la función de serie temporal. Eliminar la tendencia de una serie implica precisamente suprimir esa pendiente y para ello existen diferentes enfoques:

1. Restar la recta de mejor ajuste a la serie de tiempo. Esta recta se puede obtener a través de un modelo de regresión lineal utilizando los pasos de tiempo como predictores.
2. Restar el componente de tendencia que se ha obtenido a través de la descomposición de la serie (ver *6.3.2*)
3. Restar la media de los valores de la serie
4. Utilizar filtros como el filtro Baxter-King o el filtro Hodrick-Prescott.

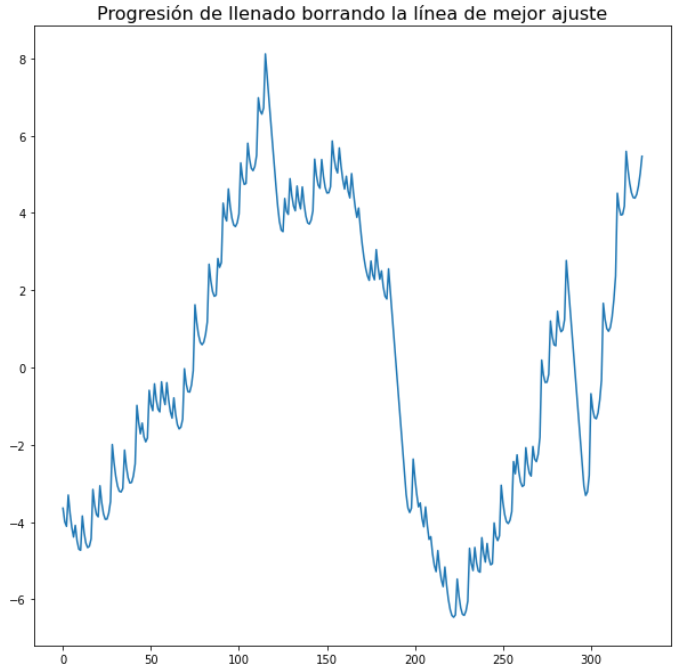


Ilustración 17. Progresión de llenado sin tendencia

En la *Ilustración 17* se muestra la serie temporal que habíamos introducido en la *Ilustración 7* a la que se ha eliminado la tendencia con el primero de los métodos (restar recta de mejor ajuste).

6.3.4. Previsibilidad

Las series temporales serán más fáciles de pronosticar cuantos más patrones regulares y repetibles presenten. Podemos cuantificar la regularidad e imprevisibilidad de las fluctuaciones de una serie del tiempo mediante la *entropía aproximada*. A valores más altos de entropía aproximada más difícil será pronosticar la serie del tiempo. Esta métrica de todas formas presenta un problema de consistencia para series pequeñas, la entropía aproximada será más baja en una serie aleatoria con pocas muestras que en una serie regular con muchas muestras. Para solucionar este problema aparece la *entropía de muestra*, esta métrica es muy similar a la anterior solo que en este caso es muy regular para series temporales con pocas muestras, de forma que esta, sí que presentará valores más bajos para series regulares con pocas muestras que para series aleatorias.

7. Modelo predictivo ARIMA

En el apartado *6* hemos estudiado las principales características y propiedades de las series temporales por lo que el siguiente paso será pronosticar sus valores

futuros. El pronóstico de series temporales es muy importante por ejemplo en empresas de fabricación, donde es importante una buena planificación para la adquisición de materia prima, producción y control de stocks. En aplicación a este trabajo de fin de master, el pronóstico será importante para determinar el momento ideal para ir a vaciar un punto de recogida, evitando viajes innecesarios que redundarán en un importante ahorro económico (gastos de personal y combustible) así como un importante ahorro medioambiental (ruido, emisiones de CO2, desgaste de los vehículos).

Cuando tratamos con series univariables utilizaremos únicamente los valores anteriores de la serie del tiempo para predecir sus valores futuros, en ese caso llamaremos al proceso **pronóstico de serie de tiempo univariante**. Si utilizamos otros predictores o variables (también conocidos como variables exógenas) para realizar el pronóstico llamaremos al proceso **pronóstico de serie de tiempo multivariable**.

Uno de los principales algoritmos para el pronóstico de series temporales univariantes es el algoritmo ARIMA, acrónimo de AutoRegresive Integrated Moving Average o promedio móvil autoregresivo integrado en castellano. Cualquier serie de tiempo no estacional que presente patrones y no sea ruido blanco puede modelarse con ARIMA, este algoritmo utiliza los valores pasados o rezagos (sus propios retrasos y errores de pronóstico retrasados) de una serie temporal para predecir los valores futuros.

Un modelo ARIMA se caracteriza por 3 términos p, d, q donde:

- **p** es el orden del término AR
- **q** es el orden del término MA
- **d** es el número de diferenciaciones necesarias para que la serie del tiempo no sea estacionaria

Cuando una serie del tiempo presenta patrones estacionales entonces podemos modelarla mediante el modelo SARIMA (Seasonal ARIMA).

7.1. p, d, q en el modelo ARIMA

p es el orden del término *autorregresivo* (AR), se refiere al número de rezagos que se utilizarán como predictores de Y

d es el número de diferenciaciones necesarias para conseguir que la serie sea estacionaria. Como ya comentamos en el punto *6.2.2* es importante que la serie sea estacionaria ya que en ese caso los predictores no estarán correlacionados entre sí, por lo que el modelo autorregresivo funcionará mejor. En el punto *6.3.1* habíamos comentado además que el método más común de convertir una serie no estacionaria en estacionaria era la diferenciación.

q es el orden del término *media móvil* (MA) y se refiere al número de errores de pronóstico retrasados que se deben incluir en el modelo.

7.2. Modelos AR y MA

Un modelo autorregresivo puro (AR) es aquel en el que Y_t depende exclusivamente de sus propios retrasos, es decir Y_t es una función de los retrasos de Y_t.

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} + \epsilon_1$$

Ecuación 6. Modelo autorregresivo puro

Donde Y_{t-1} es el retraso 1 de la serie, β_1 es el rezago 1 que estima el modelo y α es el término de intersección, también estimado por el modelo.

Por otro lado un modelo puro de media móvil (MA) es uno en el que Y_t depende únicamente de los errores de pronóstico retrasados.

$$Y_t = \alpha + \epsilon_t + \emptyset_1 \epsilon_{t-1} + \emptyset_2 \epsilon_{t-2} + .. + \emptyset_q \epsilon_{t-q}$$

Ecuación 7. Modelo de media móvil puro

Donde los términos de error son los errores de los modelos autorregresivos de los respectivos rezagos. Los errores ϵ_t y ϵ_{t-1} son los errores de las siguientes ecuaciones:

$$Y_t = \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_0 Y_{t-0} + \epsilon_t$$

Ecuación 8.Modelo autorregresivo de Y_t

$$Y_{t-1} = \beta_2 Y_{t-2} + \beta_3 Y_{t-3} + .. + \beta_0 Y_{t-0} + \epsilon_{t-1}$$

Ecuación 9. Modelo autorregresivo de Y_{t-1}

Un modelo ARIMA es aquel en el que se diferencia al menos una vez la serie temporal para hacerla estacionaria y se combinan los términos AR y MA, de forma que la ecuación se convierte en

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p+} \epsilon_t + \emptyset_1 \epsilon_{t-1} + \emptyset_2 \epsilon_{t-2} + .. + \emptyset_q \epsilon_{t-q}$$

Ecuación 10. Ecuación de modelo ARIMA

O lo que es lo mismo, una constante (α) mas una combinación lineal de *Y* (hasta p retrasos) más una combinación lineal de errores de pronóstico retrasados (hasta p retrasos).

7.3. Orden de diferenciación d

En el punto *6.2.2* hemos hablado de la importancia de que una serie sea estacionaria y en los puntos 6.2.3 y *6.3.1* hemos visto cómo se prueba si la serie es o no estacionaria mediante el test Augmented Dickey Fuller (ADF) y como se deben ir repitiendo sucesivas diferenciaciones hasta probar que la serie es estacionaria con dicho test. El valor *d* en ARIMA representa el número de diferenciaciones necesarias para convertir la serie en estacionaria.

Podríamos igualmente dibujar el gráfico de autocorrelación para cada diferenciación, si las correlaciones son positivas para muchos rezagos (más de 10), entonces la serie necesita una mayor diferenciación. Por otro lado, si la autocorrelación de retardo 1 es demasiado negativa, entonces probablemente la serie esté sobrediferenciada.

7.4. Orden del término p (AR)

Para averiguar el número requerido del término *p* en una serie temporal, podemos fijarnos en el gráfico de autocorrelación parcial que vimos en el punto *6.1.4*. Iremos aplicando diferenciaciones a la serie temporal hasta que esta sea estacionaria y que el gráfico de autocorrelación parcial muestre la mayoría de valores dentro del intervalo de confianza. En ese momento, el término *p* corresponderá al número de valores o rezagos que cruzan el límite de significancia del intervalo de confianza.

7.5. Orden del término q (MA)

Para averiguar el número requerido del término *q* seguiremos utilizando el método de la diferenciación pero, en esta ocasión, en lugar de utilizar el gráfico de autocorrelación parcial como hacíamos para obtener el término *p* nos fijaremos en el gráfico de autocorrelación siguiendo el mismo criterio que antes.

7.6. Series ligeramente por debajo o encima de la diferencia

Podría ocurrir que una serie esté ligeramente infradiferenciada pero que, una diferenciación más haga que esté sobrediferenciada. En este caso podemos jugar con los términos AR y MA.

Si la serie está ligeramente por debajo de la diferencia podemos compensarlo agregando uno o más términos AR adicionales. Por el contrario, si la serie está ligeramente sobrediferenciada lo compensaremos agregando un término MA adicional.

7.7. Construcción del modelo ARIMA

Ya sabemos cómo obtener los tres términos del modelo ARIMA (p, d, q) por lo que únicamente nos quedará ponerlo en práctica. En el paquete statsmodels está disponible el algoritmo ARIMA que tiene el parámetro *order* al cual se le deben indicar como parámetros los tres términos.

Mediante este algoritmo podremos empezar a pronosticar, aunque esto lo dejamos para más adelante, cuando entremos de lleno con nuestro dataset y las pruebas realizadas.

7.8. Obtención automática de p, d y q

Hemos visto la forma de obtener los términos p, d y q de forma manual, a partir del estudio de diferenciación de la serie y el estudio de las gráficas de autocorrelación y autocorrelación parcial pero, en la práctica hay métodos quizás más eficaces y sobre todo más automatizados. En un entorno de producción real, no podremos estudiar cada una de las propiedades del modelo paso a paso, por lo que usaremos estas técnicas para lanzar los entrenamientos y los pronósticos.

El paquete pmdarima incorpora una función *auto_arima* que realiza múltiples combinaciones de los valores p, d y q para elegir el que tenga mejor AIC (Akaike Information Criterion) por lo que realizará todo el trabajo duro por nosotros.

7.9. Gráficas de los residuos

Una vez se ha obtenido mediante auto_arima la mejor configuración de los términos (p, d, q) es conveniente estudiar los gráficos de los residuos del modelo que se presentan en la *Ilustración 18*.

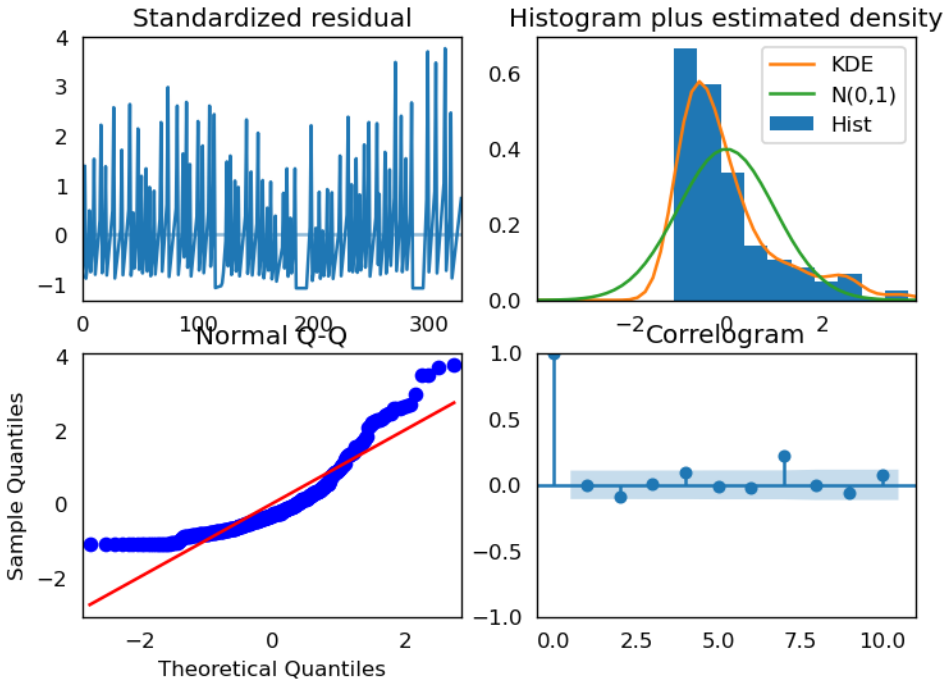


Ilustración 18. Gráficas de los residuos de auto_arima

En estas gráficas debemos observar:

- El primer gráfico (arriba izquierda) nos muestra que los residuos parecen fluctuar alrededor de cero con una varianza uniforme.
- El segundo gráfico (arriba derecha) nos muestra una gráfica de densidad de residuos, donde apreciamos una distribución normal centrada en cero.
- El tercer gráfico (abajo izquierda) muestra la desviación de los residuos, estos deben estar alineados con la línea roja. Una desviación significativa indicaría que la distribución está sesgada.
- El último gráfico (abajo derecha) también conocido como gráfico ACF (ver *6.1.4*), muestra que los errores residuales no están autocorrelacionados.

Cualquier autocorrelación implicaría la existencia de un patrón que no se explica con el modelo y por lo tanto, habría que buscar más predictores.

7.10. Variante modelo SARIMA

Una de las limitaciones que nos encontramos con el algoritmo ARIMA es que no admite estacionalidad, pero el modelo SARIMA nos ayudará a resolver este problema aplicando una diferenciación similar al modelo ARIMA, solo que en lugar de restar términos consecutivos resta los términos de la temporada anterior. El modelo se representa como (p, d, q) x (P, D, Q) donde

- P es el orden del término SAR
- D es el orden de diferenciación estacional
- Q es el orden del término SMA
- x es la frecuencia de la serie temporal

Podemos construir un modelo SARIMA con el método *auto_arima* visto en el punto *7.8*, únicamente tendremos que marcar *seasonal=True* y podremos configurar las variables *start_P*, *max_P*, *start_Q*, *max_Q*, *D* y *max_D*.

7.11. Variante modelo SARIMAX

Ya hemos visto cómo podemos utilizar el modelo ARIMA para series univariantes no estacionales y el modelo SARIMA para series univariantes estacionales. Pero, supongamos que necesitamos utilizar un predictor externo en la serie, también conocido como variable exógena, para eso podemos utilizar el modelo SARIMAX.

El único requisito para poder aplicar este modelo es que la variable exógena tendrá que tener valores para todo el periodo de la serie temporal.

Nuevamente podemos utilizar el algoritmo *auto_arima* visto en el punto *7.8* donde simplemente tendremos que configurar el parámetro *exogenus* con los valores del predictor externo.

7.12. Métricas de evaluación del modelo ARIMA

Las principales métricas de evaluación de modelos de series temporales y por consiguiente de ARIMA son métricas estadísticas que servirán para medir el desempeño de los pronósticos realizados.

Las más utilizadas son el coeficiente de determinación (R^2), el error cuadrático medio (RMSE), el error porcentual absoluto medio (MAPE), el error absoluto medio (MAE), el error medio (ME) y el error porcentual medio (MPE).

A continuación se detallará cada uno de ellos en los que debemos entender *n* como el número de observaciones, $\hat{Y}_i$ como el valor predicho por el modelo, Y_i como el valor observado e $\bar{Y}$ como el valor medio de la cantidad de residuos.

El coeficiente de determinación R^2 (*Ecuación 11*) mide la bondad del ajuste de un modelo. Contra más próximo a 1 mejor será el ajuste.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y}_i)^2}$$

Ecuación 11. Coeficiente de determinación R2

El error cuadrático medio RMSE (*Ecuación 12*) mide el error que hay entre dos conjuntos de datos (etiquetado y pronóstico). Es siempre positivo y un valor 0 indicaría que el ajuste es perfecto. Es una métrica que no sirve para comparar distintos modelos pues los valores que tome dependen de la escala del mismo.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2}$$

Ecuación 12. Error cuadrático medio (RMSE)

El error porcentual absoluto medio o MAPE (*Ecuación 13*) es una medida de precisión de la predicción. Sus valores están acotados en el intervalo 0 – 1 y refleja el porcentaje de error absoluto de la predicción, siendo deseables los valores más próximos a cero. Dado que es una medida porcentual de la precisión es válida para comparar distintos modelos y predictores.

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{Y_i - \hat{Y}_i}{Y_i}\right|$$

Ecuación 13. Error porcentual absoluto medio (MAPE)

El error absoluto medio MAE (*Ecuación 14*) muestra el promedio de la diferencia absoluta entre el valor predicho y el valor real. Igual que ocurría con RMSE no servirá para comparar distintos modelos o predictores ya que sus valores variarán en función de la escala.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i|$$

Ecuación 14. Error absoluto medio (MAE)

El error medio ME (*Ecuación 15*) por definición es igual al MAE salvo que en este caso no se mide el error absoluto sino el error real. Tiene por lo tanto las mismas carencias que el anterior.

$$ME = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)$$

Ecuación 15. Error medio (ME)

El error porcentual medio MPE (*Ecuación 16*) es muy similar al MAPE solo que la diferencia del error no es absoluta sino real y sirve para identificar si el error de pronóstico tiene un sesgo positivo o negativo. Dado que es un valor porcentual además podremos utilizarlo para comparar distintos predictores o modelos.

$$MPE = \frac{100}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{Y}_i}{Y_i}\right)$$

Ecuación 16. Error porcentual medio (MPE)

Las métricas MAPE y MPE son errores porcentuales que varían entre 0 y 1 por lo que resulta más sencillo comprender la bondad del modelo implementado. El resto de errores son cantidades, por lo que resulta más compleja su interpretación, sobre todo si pretendemos comparar los resultados entre dos series temporales diferentes.

8. Redes neuronales aplicadas a la predicción

Otro de los métodos más comunes para el pronóstico de valores en series temporales es el uso de redes neuronales y más concretamente las redes neuronales recurrentes (RNN). Las redes neuronales son capaces de aprender y extraer características automáticamente de datos sin procesar. Esto hace que se puedan utilizar para resolver problemas de predicción de series temporales donde los modelos se pueden desarrollar directamente sobre las observaciones sin la necesidad de aplicar ese pre-procesamiento (por ejemplo, normalizar datos, escalarlos o incluso convertir las series en estacionarias como ocurría con ARIMA).

Los modelos más simples de redes neuronales son capaces de realizar pronósticos muy buenos en comparación con otros modelos ingenuos o modelos ARIMA incluso sin el pre-procesamiento previo de los datos.

Las RNN por su parte, a diferencia de las redes neuronales simples donde la función de activación solamente actúa en una dirección (desde la capa de entrada hacia la capa de salida) incluyen conexiones que apuntan hacia atrás recibiendo así una retroalimentación de estados entre neuronas. En cada instante del tiempo, una neurona recurrente recibe la entrada de la capa anterior así como su propia salida del instante de tiempo anterior para generar la salida. Este tipo de redes están especializadas en procesar secuencias de tiempo donde la salida (predicción) es añadida a la siguiente entrada (podemos ver su representación matemática en la *Ecuación 17*).

$$h_t = f(h_{t-1}, X_t, \theta)$$

Ecuación 17. Ecuación de RNN

En la *Ilustración 19* podemos ver además un gráfico de una red neuronal recursiva donde las salidas de un paso del tiempo se utilizan en la entrada del siguiente, de forma que la red va aprendiendo de los pasos de tiempo anteriores.

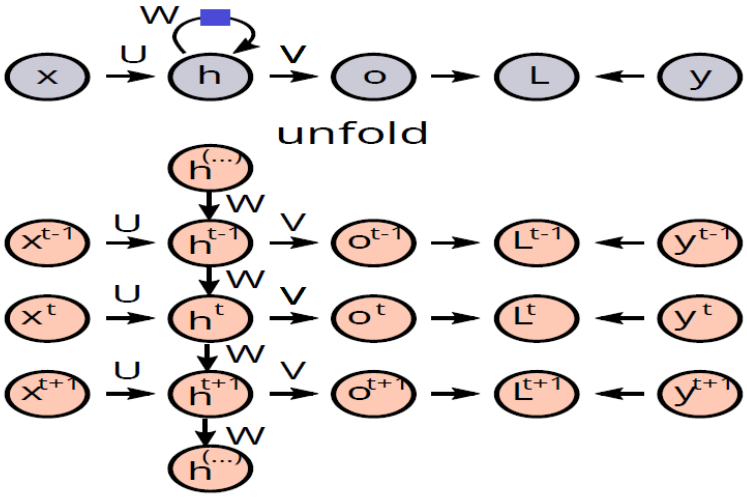


Ilustración 19. Grafo de una RNN [13]

8.1.1. Aprendizaje supervisado en red neuronal

Dado que el objetivo es entrenar la red para pronosticar una serie de tiempo a partir de valores pasados sí que es necesario una pequeña transformación de los datos para convertirlos a un formato etiquetado, donde para cada valor en un instante *T* su etiqueta será el valor para el instante *T+1*, de esta forma la red será capaz de entrenar unas entradas para ajustar los valores a las salidas proporcionadas.

Supongamos una serie temporal con los valores 1, 4, 5, 7, 8, 9, 9, 8, 1, 2, 5. El formato supervisado consistiría en dos columnas, una con los valores de la serie original y otra con los valores desplazados en una posición como podemos ver en la *Tabla 4* en la que se descartarían la primera y última fila por tener valores sin datos.

nan	1
1	4
4	5
5	7
7	8
8	9
9	9
9	8
8	1
1	2
2	5
5	nan

Tabla 4. Formato aprendizaje supervisado

8.1.2. Validación progresiva

Una técnica muy utilizada para la validación de una red neuronal utilizada para pronóstico de una serie temporal es la validación de avance (walk-forward validation).

La validación de avance es un enfoque en el que el modelo realiza un único pronóstico para el conjunto de prueba a la vez. El valor real del pronóstico se añade después al dataset de test (en los modelos más simples de redes neuronales se puede reajustar el modelo en este momento, pero en los complejos no por el alto coste computacional) para para continuar pronosticando el siguiente valor y así sucesivamente.

8.1.3. Repetir evaluación

Los modelos de redes neuronales son estocásticos, lo que significa que cada vez que se entrena un mismo modelo se obtendrá un conjunto interno de ponderaciones diferente que a su vez, generará unas métricas de rendimiento distintas.

Este comportamiento que resulta beneficioso para abordar problemas complejos ya que es capaz de adaptarse y encontrar configuraciones de alto rendimiento hace que pueda resultar complejo evaluar un modelo para un dataset. Esto nos llevará a tener que entrenar y pronosticar varias veces el mismo modelo para obtener unos valores medios de las métricas de rendimiento.

9. Objetivos concretos y metodología del trabajo

El objetivo de este trabajo de fin de master es analizar el uso de series temporales sobre un dataset con información de los pesos de los contenedores en el momento de la recogida en una ciudad de unos 250K habitantes para aprender la evolución o progresión de llenado de los contenedores a través del peso con el objetivo de ser capaces de predecir el comportamiento de llenado del mismo de forma que el gestor del servicio sea capaz de planificar las rutas de recogida en función de lo que previsiblemente va a ser necesario recoger. Esto implicaría un importante ahorro tanto económico como medioambiental como hemos visto en el apartado *4*.

9.1. Objetivos generales

Dado el estado del arte de las series temporales y el dataset que disponemos y que veremos con detalle más adelante, con este trabajo se pretenden pronosticar datos de una serie temporal univariable utilizando el algoritmo ARIMA y comparar los resultados obtenidos con una implementación sencilla de red neuronal para pronóstico de series temporales. Para ello los objetivos generales del trabajo serán:

1. Estudio del dataset: Estudiar y comprender el dataset entregado por la empresa.
2. Preprocesado de la información: Pre-procesamiento de la información para adaptarla al modelo que pretendemos estudiar.
3. Entrenamiento con modelo ARIMA: Realizar distintos experimentos y pruebas para aplicación del modelo ARIMA al dataset.
4. Entrenamiento con redes neuronales: Obtener métricas de rendimiento para el mejor modelo obtenido con ARIMA con una red neuronal sencilla.
5. Evaluación de resultados: Comparar las métricas obtenidas.

9.2. Objetivos específicos

Para alcanzar los objetivos anteriores se han establecido las siguientes metas específicas.

Estudio del dataset

- Inspeccionar y entender el dataset entregado por la empresa. Se comprobarán todos los campos y unidades de medida.
- Realizar un análisis preliminar del mismo para comprender el negocio teniendo en cuenta la experiencia profesional del autor y los conocimientos adquiridos en el Máster.
- Estudiar los trabajos anteriores en el área (visto en el punto 5)

Preprocesado de la información

- Configuración de un entorno de ejecución para las pruebas
- Trabajar los archivos entregados por la empresa para obtener otros más limpios con información más depurada y concisa

Entrenamiento con modelo ARIMA

- Seleccionar un conjunto de puntos de recogida sobre los que hacer experimentos
- Rellenar datos faltantes en las series
- Definir una o más estrategias univariantes para aplicar el modelo ARIMA
- Entrenar los modelos con el histórico de información reservando un dataset para test
- Realizar pronósticos y obtener métricas de evaluación

Entrenamiento con red neuronal

- Seleccionar mejor modelo ARIMA
- Crear red neuronal sencilla
- Adaptar datos de entrada a la red neuronal
- Entrenar los modelos con el histórico de información utilizando la técnica de validación progresiva
- Obtener métricas de evaluación

Comparación de resultados

- Comparar objetivamente las métricas obtenidas con cada modelo así como con los conjuntos de evaluación y prueba
- Identificar ventajas y desventajas de los modelos así como consideraciones relevantes en la implementación
- Obtener conclusiones específicas en base a los resultados del trabajo

9.3. Metodología

La metodología se ha basado en un enfoque de trabajo ágil e incremental. De esta forma se ha buscado poder ir trabajando pequeños paquetes de trabajo en los que se combinaba el estudio de las propiedades de las series temporales con la aplicación de los conocimientos adquiridos en el dataset preprocesado. Esta metodología me ha permitido desarrollar el proyecto de forma incremental y completarlo de manera satisfactoria.

10. Diseño del modelo de predicción

Una vez introducidos todos los conceptos ha llegado el momento de conocer el dataset proporcionado por la empresa y analizar las técnicas que debemos aplicar para pronosticar la serie temporal con ARIMA y con una red neuronal. A lo largo de este capítulo se explicarán con detalle todos los pasos seguidos durante la realización del TFM.

10.1. Plataforma Distromel

Los conjuntos de información que se utilizarán para la realización de este Trabajo de Fin de Master serán proporcionados por la empresa Distromel, S.A. Esta empresa dispone de una solución Hardware y Software para la gestión del servicio de recogida de residuos.

La parte Hardware se compone de distintos sensores que se instalan en los vehículos que prestan el servicio o en los propios contenedores y que envían los datos a través de tecnología GPRS, 3G, Narrow Band IoT,… a la nube de Distromel.

La parte Software es un conjunto de sistemas encargados de recibir y procesar los datos así como una aplicación Web donde el usuario puede consultar la información y gestionar el servicio de forma correcta, para lo cual tiene entre otros un inventario de contenedores y puntos de recogida, las fichas de personal y maquinaria, un módulo para consultar los datos relativos a recogidas, lecturas RFID, pesajes, un módulo de seguimiento de la flota de vehículos, módulo de incidencias y un módulo de planificación de rutas entre otros.

Entre las principales tecnologías Hardware utilizadas en los sensores de campo destacan:

- **RFID**: Tecnología de identificación por radiofrecuencia que permite identificar inequívocamente el contenedor sobre el que se ha realizado una operación (recogida, lavado, incidencia, etc.)
- **Pesaje**: Sistemas de pesaje dinámico (no necesitan detener la operación de carga del contenedor para estabilizar el peso) para registrar los kilos cargados en cada recogida

- **GPS**: Sistema de posicionamiento global que servirá para grabar la ruta por la que se desplazan los vehículos que prestan el servicio
- **Ultrasonidos - Láser**: Combinación de tecnologías que utilizan los sensores de volumetría para medir el porcentaje de llenado de los contenedores.
- **CanBus**: Protocolo de comunicaciones a través del cual se puede obtener la información del vehículo según el estándar J1939 de la SAE (Society of Automotive Engineers)
- **GPRS, 3G, Narrow Band IoT,…**: Diferentes tecnologías de comunicación a través de las cuales los sensores instalados en los equipos envían la información a la nube.
- **Cloud**: Centro de Procesamiento de Datos donde se registra la información recibida de todos los dispositivos

En este sentido, **Distromel** diseña, desarrolla y fabrica distintos elementos Hardware o sensores que captan información del servicio de recogida de residuos aportando información valiosa para la empresa (rutas realizadas, consumos de combustible, kilos recogidos, kilos llevados a vertedero o planta de reciclaje, etc.). Actualmente la plataforma de Distromel está siendo utilizada por más de 800 usuarios (gestores del servicio que utilizan la aplicación Web) en unas 300 delegaciones y con más de **6000 dispositivos** enviando datos al IoT. Cada uno de esos dispositivos puede tener instalado uno o varios sensores que enviarán datos a la nube, en **2018** la plataforma soportó más de **438 millones de conexiones**, en **2019** creció hasta más de **684 millones** y en **2020** ha alcanzado los **866 millones** (cada conexión puede estar compuesta por distintos tipos de información, por ejemplo, un número de posiciones Gps de la ruta junto con una lista de elevaciones o pesajes de contenedores). Esta cantidad ingente de información relacionada con la prestación del servicio de recogida de residuos se hace propicia para ser analizada con técnicas de Big Data y Machine Learning para sacar el máximo rendimiento y provecho de la misma.

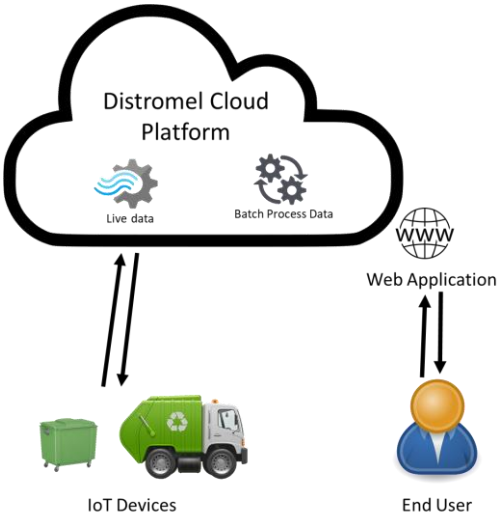


Ilustración 20. Esquema general plataforma Distromel

En la *Ilustración 20* se presenta un esquema general de la plataforma de Distromel, S.A. Los sensores que captan datos del servicio envían datos a la nube de la empresa (Distromel Cloud Platform) que contiene una serie de servicios para la ingesta de datos en tiempo real encargados de un almacenamiento y respuesta rápida a los dispositivos y una serie de servicios batch para el procesamiento de la información, encargados de refinar los datos o hacerlos más legibles para el usuario. Finalmente a través de una aplicación Web el usuario final accederá a la información generada no solo por los propios dispositivos sino por los citados servicios de plataforma o los que él mismo haya generado (órdenes de trabajo, incidencias, etc.)

Actualmente, el servicio de recogida de residuos, se realiza en base a una planificación establecida en el contrato entre la empresa de servicios y la administración. Las fracciones que más degradación de residuo sufren (orgánica, resto o rechazo) se tienden a recoger de manera frecuente, en muchos casos diariamente, mientras que las fracciones que no degradan (o no lo hacen significativamente en periodos cortos de tiempo) se recogen cada cierto número de días, estas fracciones son las que se conocen como selectivas y corresponden con los residuos reciclables como envases, papel cartón o vidrio. En este contexto el contratista genera una serie de *rutas maestras* con unas frecuencias de ejecución en función del residuo a recoger que incluyen una serie de contenedores. Durante la prestación del servicio el camión pasará y recogerá todos los contenedores indiscriminadamente sin importar su nivel de llenado.

Este modelo de recogida admite mucho margen de mejora y el objetivo de este proyecto es el estudio de los datos relativos a pesaje de los contenedores de las fracciones de recogida selectiva con la finalidad de encontrar modelos de **Machine Learning** capaces de predecir las fechas de llenado del contenedor. A partir de esta información el gestor tendrá la posibilidad de **optimizar la recogida de residuos** generando rutas dinámicas en base a lo que realmente hay que recoger y no en base a un calendario fijo como ocurre actualmente. Este modelo de recogida supondría directamente un **ahorro de costes** del servicio así como un **menor impacto ambiental** por la generación de residuos directos e indirectos.

10.2. Estudio del dataset

Para la realización del estudio la empresa proveedora de los datos nos entrega dos archivos en formato CSV, uno con el movimiento de los contenedores durante la prestación del servicio y otro que contiene los pesos de los contenedores en el momento de la recogida.

10.2.1. Movimientos de contenedores

La empresa nos entrega un fichero que contiene la información sobre el movimiento de los contenedores entre los distintos puntos de recogida entre los años 2016 y 2020, ambos inclusive con 1.338.851 registros de movimiento. El concepto de contenedor está claro, pues todos a diario hacemos uso de los mismos, pero quizás el concepto de punto de recogida o isleta no sea conocido por todos pese a que también los vemos a diario. Se entiendo por punto de recogida las ubicaciones físicas en la calle en la que se agrupan uno o varios contenedores de uno o varios residuos, estas ubicaciones además son fijas, no cambian durante la vida útil del contrato, y si lo hacen, debe ser previa autorización del Ayuntamiento de la ciudad.

Durante la vida útil de un contenedor, este se moverá entre los distintos puntos de recogida de la ciudad por distintos motivos. Imaginemos por ejemplo las fiestas patronales, en las que se concentran una serie de eventos o festejos en zonas localizadas de la ciudad. Durante la duración de las mismas se moverán una serie de contenedores procedentes de otras partes de la ciudad hacia la que se prevé mayor generación de residuo. Pero no solo se moverá para cubrir la demanda de generación de residuo, sino que en ocasiones será una constante durante los servicios de lavado de los contenedores. Imaginemos ahora el típico contenedor verde de carga trasera

(se coge por la parte trasera del camión), esos contenedores se deben lavar y generalmente se hace con una frecuencia mensual (aunque esto lo determina el Ayuntamiento durante la fase de contratación pública). El caso más típico del servicio de lavado es que el camión, un lavacontenedores, lleva una cuba donde se introduce el contenedor y automáticamente se lava con agua caliente y a presión en el interior. Este camión sale de la base con un contenedor limpio en la cuba detrás de un camión de recogida. Cuando se vacía el primer contenedor, el lavacontenedores descarga el camión que lleva limpio en la cuba y carga el contendor recién vaciado continuando la ruta tras el camión recolector y lavando el contenedor a la vez que circula. Esto lo hará durante toda la ruta de recogida, lo que implica que todos los contenedores de la ruta cambiarán de un punto de recogida a otro.

El sistema electrónico instalado en los camiones recolectores y lavacontenedores así como los transponders RFID instalados en los contenedores hacen que el sistema informático de la empresa sea capaz de detectar esos movimientos de contenedores de forma automática, por lo que tienen registrada toda esa información que es la que nos facilitan en el archivo *containers_movement_1.csv*.

El contenido del archivo se detalla en la *Tabla 5.*

Nombre columna	Tipo dato	Descripción
StartDate	Fecha/hora	Indica la fecha y hora en la que el contenedor se situó en esta ubicación
EndDate	Fecha/hora	Indica la fecha y hora en la que el contenedor abandonó esta ubicación
CollectionPointId	Entero	Código identificador del punto de recogida en el que se sitúa el contenedor
ContainerId	Entero	Código identificador del contenedor
Capacity	Entero	Indica la capacidad en litros del contenedor. Un mismo ContainerId siempre tendrá la misma capacidad

Waste	Texto	Residuo del contenedor. Un mismo ContainerId siempre tendrá el mismo residuo
Density	Entero	Densidad del residuo en $\frac{Kg}{m^3}$. Un mismo residuo siempre tendrá la misma densidad

Tabla 5. Detalles del archivo de movimiento de contenedores

10.2.2. Pesaje de los contenedores

La empresa nos entrega igualmente un fichero que contiene la información sobre cada una de las descargas o vaciados de los contenedores en el mismo periodo que el archivo anterior (2016 – 2020) en este caso con 4.357.005 registros de pesaje. Es lógico que el número de líneas sea mayor puesto que un contenedor no cambia de ubicación cada vez que se vacía, sino, como hemos explicado en el punto anterior, según las necesidades u operativa del servicio.

El sistema electrónico instalado en los camiones es capaz de detectar en el momento del vaciado del contenedor el contendor exacto que se está recogiendo (por medio de identificación por radiofrecuencia) así como el peso neto del residuo que se encuentra en el interior del contenedor (a través de sistemas de pesaje dinámico instalados en el elevador del camión) y en este archivo se encuentra disponible la información en el siguiente formato (archivo *weights_1.csv*)

Nombre columna	Tipo dato	Descripción
FullDate	Fecha/hora	Indica la fecha y hora en la que se vacía el contenedor
Weight	Entero	Indica los Kilogramos de residuo que se han vaciado del contenedor
Capacity	Entero	Indica la capacidad en litros del contenedor. Un mismo ContainerId siempre tendrá la misma capacidad

Waste	Texto	Residuo del contenedor. Un mismo ContainerId siempre tendrá el mismo residuo
Density	Entero	Densidad del residuo en $\frac{Kg}{m^3}$. Un mismo residuo siempre tendrá la misma densidad
Incidence	Texto	Descripción de la incidencia surgida en el momento de la recogida si la hay
ContainerId	Entero	Código identificador del contenedor
CollectionPointId	Entero	Código identificador del punto de recogida en el que se sitúa el contenedor en el momento de la recogida
SlideLatitude	Decimal	Latitud desplazada en formato WGS84 del punto de recogida. La latitud está desplazada aleatoriamente en el mapa para mantener el anonimato de la ciudad
SlideLongitude	Decimal	Longitud desplazada en formato WGS84 del punto de recogida. La longitud está desplazada aleatoriamente en el mapa para mantener el anonimato de la ciudad

10.2.3. Análisis preliminar

Nuestro objetivo es ser capaces de predecir la generación de residuo para optimizar las rutas de recogida, por lo que queremos bajar a un nivel de agrupación inferior al del resto de trabajos que se han presentado en el punto *5*. No nos sirve pronosticar cuánto residuo se va a generar en la ciudad o municipio, sino que nuestro objetivo es poder pronosticar cuanto residuo se va a generar en cada uno de los puntos de generación o más bien, los puntos de recogida.

Los operadores planifican las rutas por puntos de paso, los cuales corresponden a esos puntos de recogida por lo que no importa qué contenedor haya

en un momento dado, puesto que, con independencia de si el contenedor que hay instalado es el *A* o el *B* la generación de residuo por parte del ciudadano será la misma, de hecho, un contenedor *A* tendrá una progresión de llenado distinta si está ubicado en un punto de recogida de zona residencial o si está ubicado en un punto de recogida de una zona de bares. Por ese motivo, si fuésemos capaces de predecir la generación de residuo en un punto de recogida podríamos, no solo saber el momento adecuado para su vaciado sino también determinar la cantidad de contendores de un determinado residuo necesarios para un buen nivel de calidad del servicio.

De este modo, tendremos que pensar en generar un modelo ARIMA por cada punto de recogida y no uno para todos ellos, pues cada uno tendrá un comportamiento distinto en función de la zona de la ciudad en la que se encuentre. Deberemos pues procesar los archivos pensando en este tipo de modelado.

10.3. Preprocesado del dataset

Como en prácticamente la totalidad de procesos de ciencia de datos, y más concretamente en el caso que nos aplica, de procesos de aprendizaje automático, los datos en crudo deben ser preprocesados para poder trabajar con ellos mediante. En este punto se describen las principales transformaciones que se han llevado a cabo en los datos para poder continuar más adelante con el modelo ARIMA (no todos son necesarios para las redes neuronales, aunque lo veremos con más detalle en el apartado dedicado a ellas).

10.3.1. Configuración del entorno de pruebas

Para las pruebas se ha utilizado una distribución *Anaconda Navigat*or (*[4]*) en la versión 1.10.0. Se ha escogido Anaconda porque es una suite de código abierto que incorpora una serie de aplicaciones, librerías y conceptos diseñados para el desarrollo de la ciencia de datos con Python. Es una herramienta multiplataforma y básicamente funciona como un entorno de paquetes y tiene en la actualidad una colección superior a los 720.

Todos los scripts y entrenamiento de modelos se han creado a través de Jupyter Notebooks con Python 3.0.

Además, en el entorno conda se han instalado las librerías *statsmodels* (*[6]*) y *pmdarima* (*[5]*) que contienen la definición del modelo *ARIMA* y *auto_arima* que utilizaremos para pronosticar las series temporales.

10.3.2. Capacidad total de los puntos de recogida

Hemos visto que los contenedores se mueven a través de los distintos puntos de recogida y que nos interesa pronosticar la generación de residuos en cada punto de recogida por lo que en primera instancia, parece lógico conocer la capacidad total de cada punto de recogida en cada intervalo de tiempo. Dado que los contenedores se mueven de un contendor a otro y que además, cada contendor puede tener una capacidad distinta, en función del número de contenedores y de la capacidad de los mismos en un momento determinado el punto de recogida variará su capacidad total.

Para calcular la capacidad de cada una de las fracciones nos deberemos fijar en la suma de las capacidades de los contenedores de esa fracción, para ello trabajaremos en primer lugar con el fichero de movimiento de contenedores.

En este archivo en primer lugar se realiza una conversión de unidades, el proveedor nos entrega la capacidad de cada contenedor en litros, por lo que la convertiremos a metros cúbicos (m^3). Una tenemos la información en la unidad que nos interesa, el siguiente paso será separar los intervalos de cada punto de recogida y residuo, creando una fila por cada intervalo de tiempo en el que ha habido un cambio de situación en los contenedores del punto de recogida. Supongamos que el archivo nos dice en el punto de recogida *PR1* ha hay un contenedor *C1* de *3* m^3 desde el *01/01/2021* sin fecha de fin (todavía está asignado a él), un contenedor *C2* de *2,5* m^3 desde el *03/02/2021* al *01/05/2021* y un tercer contenedor *C3* de *3,5* m^3 del *25/01/2021* al *13/04/2021*. Además supongamos que *C1*, *C2* y *C3* son todos de la misma fracción. Para el objetivo de nuestro estudio nos interesa conocer la capacidad total del punto de recogida a lo largo del tiempo por lo que crearemos un proceso que cogerá esos datos y generará intervalos de tiempo en los que se ha mantenido la capacidad del punto de recogida. Con los datos anteriores obtendríamos un resultado como el que se presenta en la *Tabla 6*.

Inicio	Fin	Capacidad Total (m3)
01/01/2021	25/01/2021	3
25/01/2021	03/02/2021	6,5
03/02/2021	13/04/2021	9
13/04/2021	01/05/2021	5,5
01/05/2021		3

Tabla 6. Capacidad total del punto de recogida

Como resultado del proceso se generará un nuevo DataFrame donde tendremos registrado por cada punto de recogida y residuo la capacidad total en los distintos intervalos de tiempo. Podemos ver un ejemplo con datos reales en la *Ilustración 21*.

	CollectionPointId	StartDate	EndDate	Waste	M3
0	1887	2000-01-01 00:01:11.000	2016-02-29 16:28:39.460	ORGANICA	3.2
1	1887	2016-02-29 16:28:39.460	2016-02-29 22:43:54.533	ORGANICA	5.6
2	1887	2016-02-29 22:43:54.533	2016-03-04 22:42:08.607	ORGANICA	8.8
3	1887	2016-03-04 22:42:08.607	2016-03-05 22:36:24.097	ORGANICA	5.6
4	1887	2016-03-05 22:36:24.097	2016-03-06 22:36:03.957	ORGANICA	3.2

Ilustración 21. Capacidad total del punto de recogida 1887

10.3.3. Llenado del punto de recogida

Una vez hemos sido capaces de calcular la capacidad exacta de cada punto de recogida a lo largo de los años 2016 – 2020, como habremos visto en la *Ilustración 21*, va variando a lo largo del tiempo. Ahora bien, todavía no sabemos cuál ha sido el volumen del residuo que se ha generado y para ello, utilizaremos el archivo con información de los pesos en el momento del vaciado (*weights_1.csv)*.

El archivo tiene una columna *Incidence*, que, como hemos visto en el punto *10.2.2* contiene información relativa a la incidencia ocurrida en el momento de la descarga del contenedor. Si comprobamos los distintos valores que toma ese campo, veremos que o bien no existe incidencia o hay una *Pesada negativa* que implica que el sistema de pesaje no ha sido capaz de estabilizar el peso del contenedor y que la diferencia de peso entre el contenedor cargado y descargado ha sido negativa. Se debe realizar pues un tratamiento de estos registros, pues es imposible que exista un peso negativo de residuo. Para ello rellenaremos el peso de los registros con ese tipo de incidencias con el peso medio del residuo en ese punto de recogida.

Una vez hemos *corregido* los datos vamos a buscar la forma de calcular el porcentaje de llenado del punto de recogida en el momento de su vaciado. Para ello, el concepto que vamos a aplicar es, que si el dataset nos da información sobre el peso del residuo en el momento de su recogida y además, tenemos la densidad media del residuo así como la capacidad total del punto de recogida (gracias al procesamiento hecho en *10.3.2*) podemos calcular el % de llenado del punto de recogida justo cuando se vaciaron los contendores mediante la *Ecuación 18*.

$$Llenado\ (\%) = 100\left(\frac{\frac{Peso\ (Kg)}{Densidad\ (\frac{Kg}{m^3})}}{capacidad\ punto\ (m^3)}\right)$$

Ecuación 18. Porcentaje de llenado de un punto de recogida

Pero debemos tener cuidado, supongamos que en un punto de recogida hay dos o más contenedores de una misma fracción (pongamos 2 para que resulte más sencillo) y que ambos han sido recogidos con apenas 2 minutos de diferencia (una operación de recogida no tarda más de 90 segundos en el peor de los casos, y tan solo unos 10 en el mejor). Si únicamente vamos calculando el porcentaje de llenado del punto de recogida para cada una de los registros, calcularíamos un porcentaje P_1 para la primera descarga y un porcentaje P_2 para la segunda, cuando lo ideal sería que hubiésemos calculado un $P_T = P_1 + P_2$. Para solucionar esto definiremos una ventana deslizante de 10 minutos y todos los contenedores de un mismo punto de recogida y fracción que se hayan recogido en esa ventana contabilizarán como una única operación de vaciado en la que el peso total será la suma de los pesos de cada recogida, de esta forma se podrá calcular el porcentaje de llenado total del punto de recogida. En la *Ilustración 22* podemos ver un ejemplo de cómo quedarán los datos una vez realizado este pre-procesamiento.

	CollectionPointId	FullDate	Waste	SlideLatitude	SlideLongitude	M3	Weight	Density	FillPercent
0	1146	2018-05-19 07:02:41	VIDRIO	48.302014	-24.734623	3.0	305.0	330.0	30.808081
1	1148	2016-04-29 08:59:57	VIDRIO	48.288426	-24.700614	3.0	390.0	330.0	39.393939

Ilustración 22. Porcentaje de llenado del punto de recogida en el momento del vaciado

Dado que el dataset contiene información de distintas fracciones (vidrio, orgánica, envases) centraremos el análisis sobre la fracción vidrio por ser la que

presenta una densidad más uniforme y está sujeta a mayor margen de mejora tal y como habíamos introducido en el punto *4*.

10.4. Visualización y preprocesado de la serie temporal

Una vez procesado el fichero completo, la información ya se encuentra lista para empezar realizar un análisis de la serie temporal. Como se explicó en el punto *6,* dos tareas fundamentales en el análisis de series temporales son la visualización y preprocesado de los datos para una mayor comprensión de los datos.

En el caso que nos ocupa, el dataset final contiene información de 2731 puntos de recogida, en nuestro análisis preliminar hemos detectado que todos los puntos presentan comportamientos muy similares, por lo que vamos a suponer que el modelo que escojamos será válido para todos ellos. Para probarlo, escogeremos un subconjunto de los 150 puntos de recogida con más datos registrados al que llamaremos TOP150. Estos puntos representan el 5.49% del total pero un 34.49% de los datos de vaciado, por lo que suponen una muestra representativa de la información. Por ese motivo, en lo sucesivo trabajaremos con este subconjunto de información.

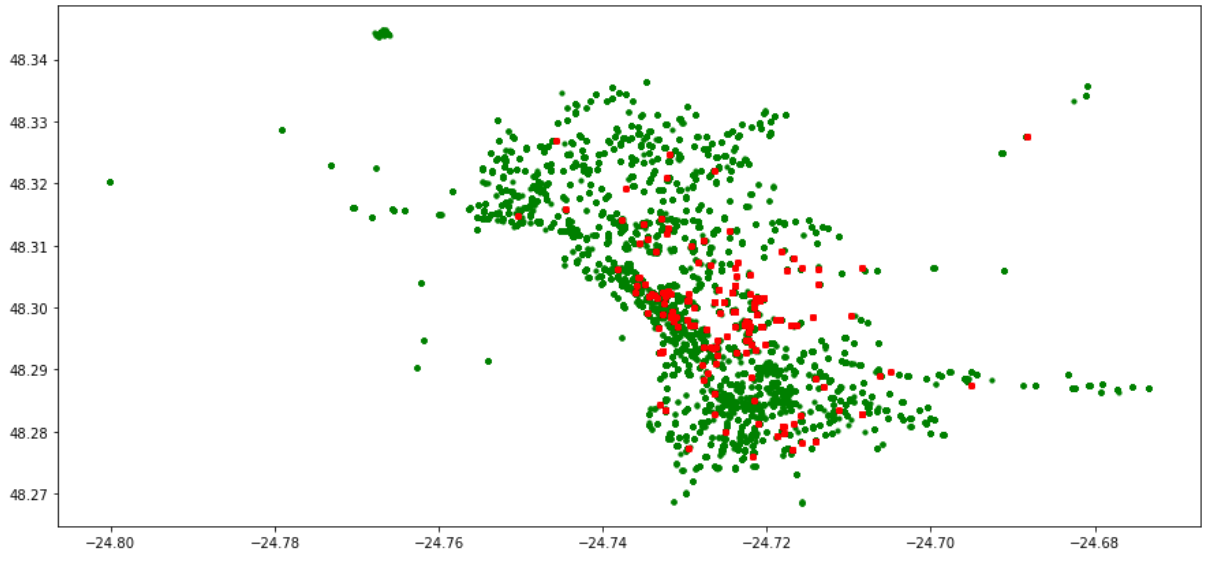


Ilustración 23. Puntos de recogida totales Vs muestreados

En la *Ilustración 23* se pueden ver representados en verde todos los puntos de recogida de la ciudad y en rojo los que se utilizan para el muestreo. Los puntos de muestras se encuentran dispersos por todo el plano de la ciudad por lo que utilizando

este pequeño subconjunto se podrá hacer una extrapolación al resto de puntos de recogida de la ciudad.

Continuaremos visualizando un histograma del TOP150 que nos ayude a comprender cómo están distribuidos los datos.

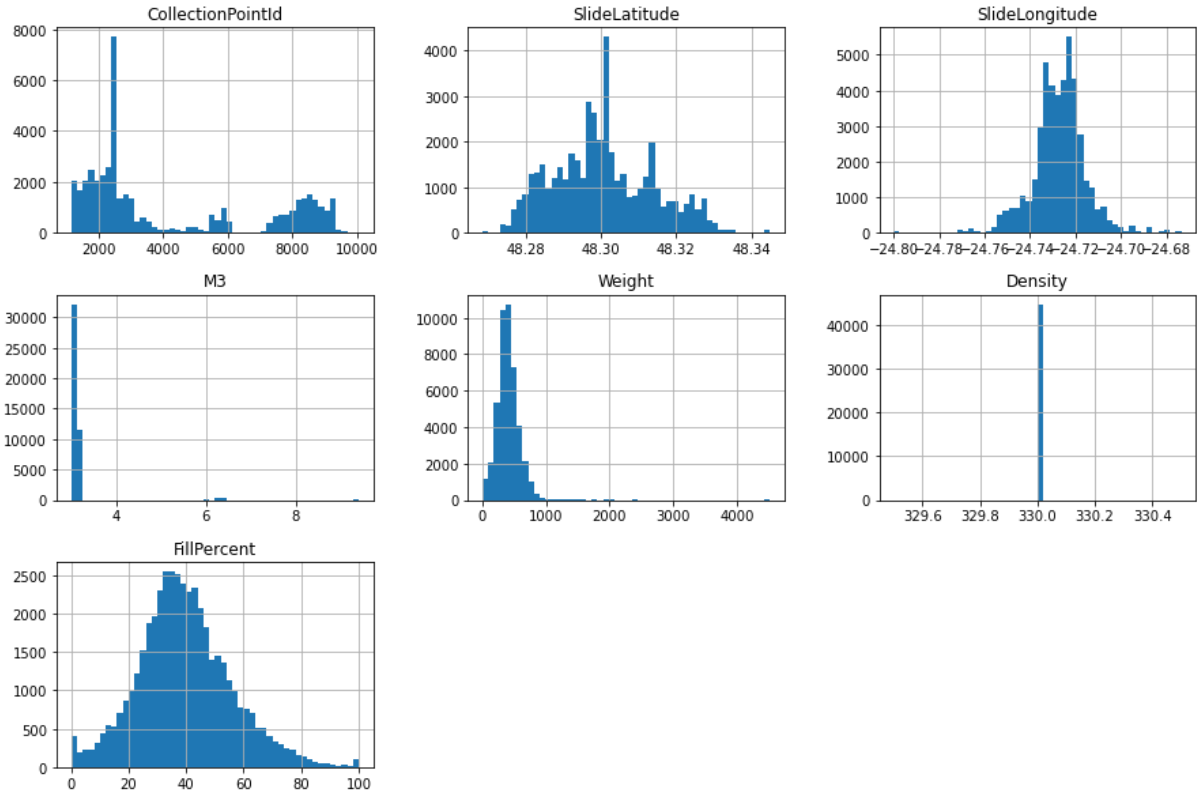


Ilustración 24. Histograma del dataset TOP150

En la *Ilustración 24* podemos ver el histograma en el que podremos apreciar:

- CollectionPointId: Dado que son los identificadores de los puntos de recogida están distribuidos en todo el rango de valores.
- Latitud y Longitud: Dado que son datos de las coordenadas de los contenedores, pese a que están desplazadas en el mapa para no identificar la ciudad objeto de estudio como es lógico ambas presentan una distribución normal centrada en el punto medio de la ciudad (desplazado).
- Capacidad (m3): Vemos que la mayor parte de valores para esta columna se encuentran por debajo de 4% salvo casos muy puntuales.
- Pesaje (Kg): Presenta una distribución normal entre 0 y 1000 kg centrado aproximadamente en 400 o 500 kg. Esto tiene lógica ya que la densidad del

vidrio (residuo con el que se realizan las pruebas) según los datos que nos proporciona la empresa es de $330\ \frac{Kg}{m^2}$ y dado el histograma de capacidades tiene sentido.

- Densidad: Dado que todo el dataset es de un único residuo y la densidad viene indicada por este, únicamente hay un único valor para esta columna.
- Porcentaje de llenado: Presenta una distribución normal con valores entre 0 y 100 como es lógico. Los datos están centrados en torno al 35% lo que significa que la gran mayoría de recogidas se están realizando cuando el punto de recogida se encuentra poco lleno.

Con el mero hecho de visualizar este histograma y más concretamente el de porcentaje de llenado se parecía que la gran mayoría de recogidas se están produciendo a niveles muy bajos de llenado del punto de recogida. Suponiendo un margen de seguridad del punto de recogida de un 20% del nivel de llenado si nos fijamos en las recogidas a menos del 80% vemos que prácticamente la totalidad de los datos se sitúan allí. Supongamos además que el momento óptimo de la recogida será antes de que el residuo supere el 80% de la capacidad del punto de recogida, pero además no debería ser inferior al 60%, para evitar recogidas innecesarias. Si observamos la cantidad de recogidas que se realizan entre el 60% y el 80% veremos que es un conjunto muy pequeño respecto a las que se realizan por debajo del 60%, luego en este caso concreto, se está infrautilizando la capacidad de los puntos de recogidas que hay en la calle. El contratista ya tendría pues información importante para aplicar una primera optimización, consistente en espaciar el tiempo entre vaciados de un punto de recogida simplemente con el pre-procesamiento de datos realizado hasta ahora.

Si representamos la serie temporal de la forma estándar (en el eje horizontal "*X*" el tiempo y en el eje vertical "*Y*" el porcentaje de llenado) podremos observar gráficas del estilo a la que se muestra en la *Ilustración 25*. En ella vemos cómo ha ido evolucionando el porcentaje de llenado del punto de recogida a lo largo del tiempo, pero debemos tener cuidado, sabemos que el punto va cambiando su capacidad total de residuo y esta gráfica solo nos muestra información porcentualizada, por lo que no sabemos en realidad si la producción de desechos ha sido mayor o menor en distintos periodos de tiempo. También podemos observar un tramo en el 2019 que parece una

recta decreciente con pendiente constante de varios meses de duración que parece un intervalo sin datos. Más adelante prestaremos más atención a este detalle.

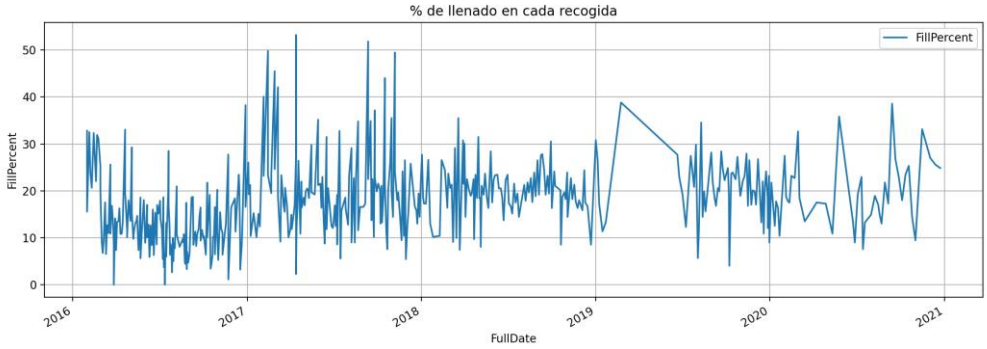


Ilustración 25. Serie temporal de % de llenado

De cara a obtener un dataset lo más limpio posible, para obtener buenos resultados con el algoritmo de predicción se limpiarán los valores anómalos u outliers de cada serie temporal.

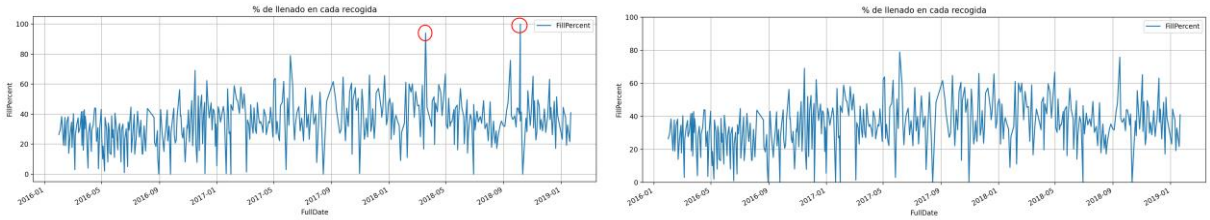


Ilustración 26. Outliers de porcentaje de llenado

En la *Ilustración 26* se puede ver en la gráfica izquierda la serie temporal de un punto de recogida antes de limpiar los valores atípicos y en la parte derecha la misma gráfica una vez limpios (los dos puntos marcados en rojo en la gráfica izquierda).

El proceso de corrección de outliers consistirá en buscar aquellos puntos que superen 3 veces la desviación media. Dado que trabajamos con una serie temporal no nos interesa borrar el valor, sino conservarlo con un valor corregido, y en nuestro caso hemos escogido el valor medio de la serie.

Continuaremos con una representación en formato calendario que nos aportará información muy valiosa. En la *Ilustración 27* vemos los vaciados de un determinado punto de recogida. En el calendario se pinta con un color, que indica según la escala

derecha el nivel de llenado (%) aquellos días en los que se ha vaciado el punto de recogida.

Ilustración 27. Calendario de vaciados de un punto de recogida

Como se puede apreciar la frecuencia de recogida era mucho mayor en 2016 que en los años siguientes, aunque no parece que eso haya afectado al nivel de llenado del punto en el momento de los vaciados, pues aunque por la escala de colores parece que se vacían cuando están algo más llenos, la diferencia es muy sutil. Este comportamiento podría deberse a que a partir de 2017 se creó un nuevo punto de recogida próximo donde los vecinos de la zona pueden depositar residuo, por lo que este se llenará más lentamente y por lo tanto, se podrá recoger más espaciadamente en el tiempo al mismo porcentaje de llenado.

Mediante este gráfico vemos también que se confirman los resultados obtenidos con el histograma de valores del porcentaje de llenado observando que el punto de recogida se está vaciando en la gran mayoría de los casos por debajo del 30% de llenado.

También podemos apreciar fácilmente que entre Febrero y Junio de 2019 hay un periodo en el que no existen datos sobre el residuo generado en el punto de recogida (ya habíamos visto un comportamiento anómalo en la *Ilustración 25*), estas situaciones generalmente se dan por dos motivos, el primero es que el vehículo que realiza la ruta se ha averiado y un vehículo de refuerzo sin sensorización está realizando su trabajo. El segundo es que el vehículo sigue siendo el mismo pero en este caso, lo que se ha averiado ha sido la propia sensorización IoT. En ambos casos el resultado es el mismo, por lo que más adelante completaremos esos intervalos de tiempo carentes de datos.

Otro punto de especial importancia que podemos apreciar en este calendario es que la frecuencia de recogida del punto está entre una y dos veces por semana aproximadamente a un 30% de información. Suponiendo que la disposición de residuo en el punto por parte de los ciudadanos es lineal podemos suponer que el punto tardará entre 2 y 3 semanas en alcanzar un 80% de llenado, que es el máximo llenado que suelen considerar los contratistas al que se debe recoger el contenedor para dar una buena impresión del servicio a los ciudadanos y a la propia administración. Esto implicará que los pronósticos que realicemos deberán dar buenos resultados en ese intervalo.

Si volvemos al problema de los intervalos sin información, vamos a diseñar un proceso para completar la información faltante.

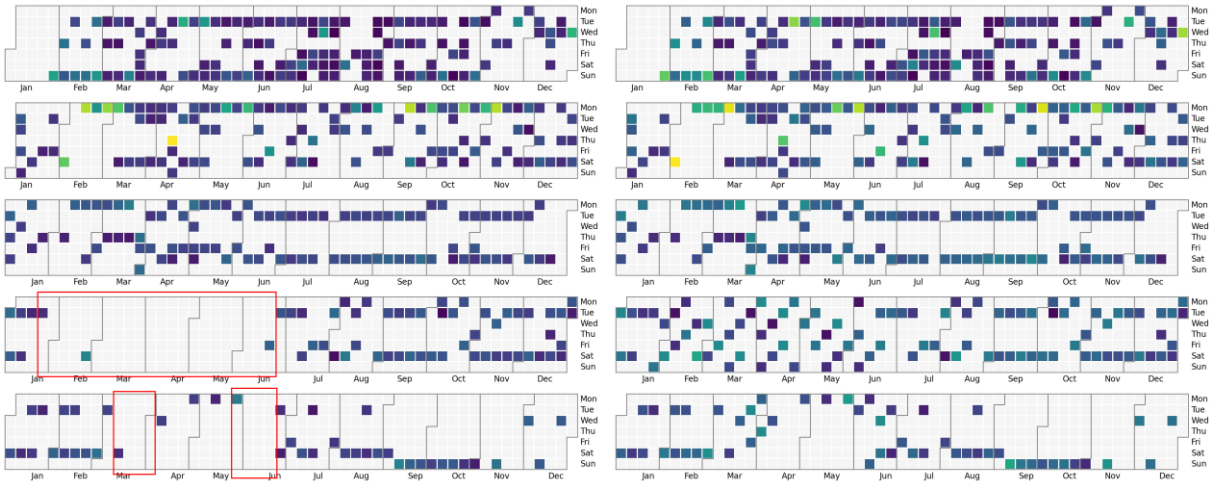


Ilustración 28. Intervalos sin información

En la *Ilustración 28* podemos ver en la parte izquierda tres intervalos de tiempo marcados en rojo en los que hay varios días sin datos. Para identificarlos se ha calculado el tiempo medio que transcurre desde un vaciado del punto de recogida hasta el siguiente y se han marcado como valores anómalos aquellos que superan 3 veces la desviación estándar. Para esos intervalos se han generado vaciados ficticios con una frecuencia igual a la media de días entre recogidas más/menos entre 1 y 2 veces la desviación media y un porcentaje de llenado igual a la media más/menos un margen de error. En la parte izquierda de la imagen vemos el resultado de este proceso.

Dado que nuestra serie temporal presenta registros a intervalos no regulares de tiempo (podemos ver el calendario de vaciados en la *Ilustración 27* para comprobarlo) se creará un proceso encargado de generar los datos diarios de llenado del punto de recogida.

¿Cómo se hará? Supongamos tres vaciados de un punto de recogida, el primero de ellos se produce el día 1 con el punto a un 75% de su capacidad. Tres días más tarde (el día 4) se vacía nuevamente, en este caso a un 50% de su capacidad. Pasados 6 días más, el punto se vuelve a vaciar, en esta ocasión al 75% de su capacidad. El porcentaje de llenado del contenedor desciende a 0 en el mismo instante en que es recogido, lo cual se representa mediante una línea discontinua en la *Ilustración 29.*

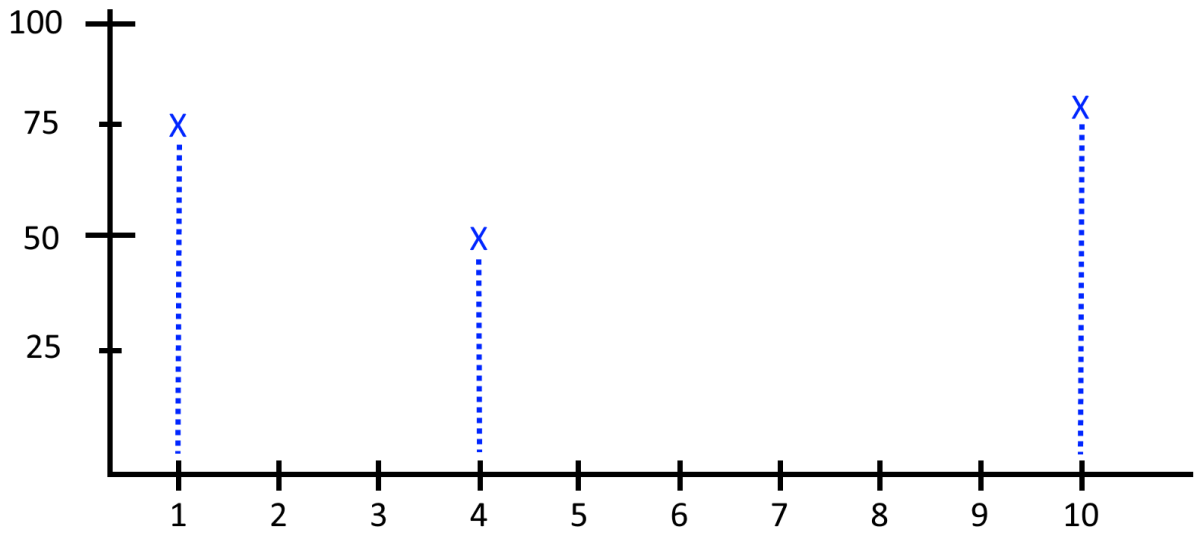


Ilustración 29. Frecuencia de muestreo irregular

Dado que no tenemos más información que las fechas y los porcentajes de llenado del punto de recogida (recordemos que trabajamos con una serie temporal univariante), podemos suponer que el llenado o generación de residuo en el punto de recogida desde una descarga a la siguiente es lineal, por lo que podemos trazar una línea desde un vaciado (al 0%) hasta el siguiente (al % de llenado anterior al vaciado) que representamos mediante una línea verde en la *Ilustración 30*, lo cual nos dará el porcentaje de llenado para cada día (puntos rojos). De esta forma, hemos transformado la serie temporal a una serie con frecuencia de muestreo diaria.

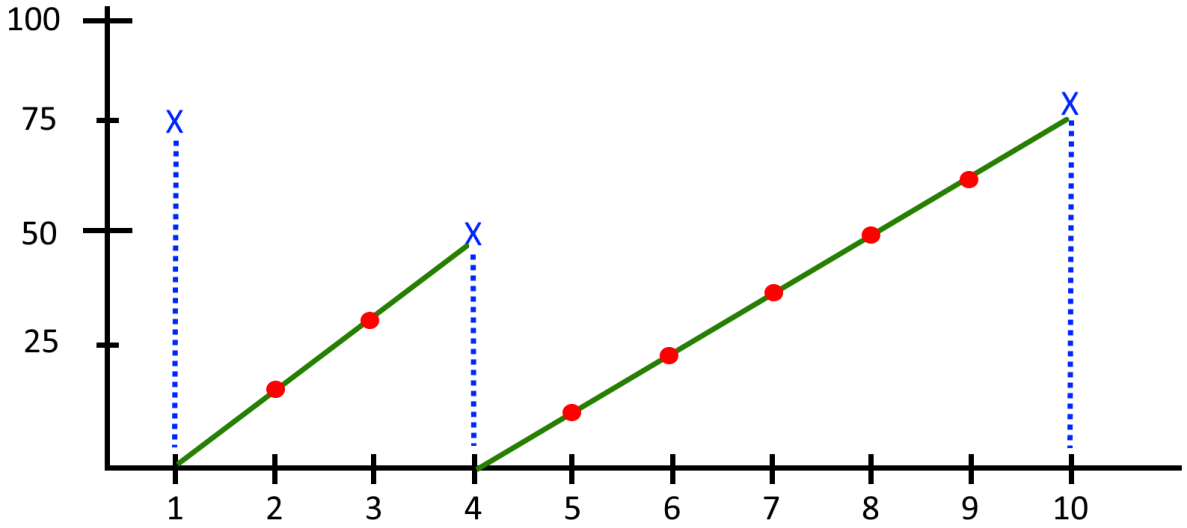


Ilustración 30. Generar frecuencia de muestreo diaria

10.5. Ajuste de parámetros, entrenamiento y pronóstico con ARIMA

Uno de los principales objetivos de este TFM era analizar el uso de ARIMA para el pronóstico del porcentaje de llenado de los puntos de recogida con la finalidad de que el gestor pueda optimizar sus rutas de vaciado de residuo. Mediante el dataset facilitado por la Distromel, S.A. y tras el preprocesado realizado, podemos construir una serie temporal univariante donde para cada día tendremos el porcentaje de llenado del punto, ahora bien, a partir de esa información podemos buscar otras variables objetivo o predictores distintos al porcentaje de llenado que también puedan servir para realizar los pronósticos que buscamos. En el estudio realizado hemos ido probando el modelo sobre distintos predictores que se explican a continuación de forma independiente hasta que hemos encontrado el que mejores resultados nos ofrece.

Siendo F_t es el porcentaje de llenado para un instante de tiempo, P_t la predicción para ese mismo instante y T_l el instante de tiempo del último vaciado, se probará el modelo sobre los siguientes predictores.

- % llenado: Indica el porcentaje de llenado del punto de recogida en el momento de su recogida. Si el modelo da buenos resultados la aplicación sería directa, ya que podríamos predecir el porcentaje de llenado para una fecha determinada.

$$F_t = P_t$$

Ecuación 19. Llenado (%) usando porcentaje de llenando como predictor

- Pendiente de llenado: A partir de la gráfica de progresión del porcentaje de llenado, podremos calcular los grados de inclinación de la recta para obtener predicciones sobre esta variable. Si da buenos resultados se podrá calcular la pendiente de llenado para una fecha determinada. En este caso, para calcular el llenando del punto de recogida en una determinada fecha, necesitaremos conocer el vaciado inmediatamente anterior para, a partir de la pendiente de la recta y el último vaciado calcular el porcentaje de llenado en esa fecha.

$$F_t = (t - T_l)tang(P_t)^1$$

Ecuación 20. Llenado (%) usando pendiente de llenando como predictor

- Seno: El concepto es el mismo que el usado con el ángulo de la recta (pendiente) como predictor solo que mediante la función *seno*. ¿Por qué probar esta función?, para escalar los valores de la pendiente en el intervalo [-1, 1] y de esta forma tener una gráfica más suavizada que la anterior.

$$P_t = \sin(\alpha)$$
$$F_t = (t - T_l)tang(\arcsin(P_t))$$

Ecuación 21. Llenado (%) usando seno de pendiente como predictor

- Coseno: Idéntico a la función seno pero con la función *coseno*.

[1] Ver funciones trigonométricas en *Anexo II (Funciones trigonométricas)*

$$P_t = \cos(\alpha)$$
$$F_t = (t - T_l)tang(\arccos(P_t))$$

Ecuación 22. Llenado (%) usando coseno de pendiente como predictor

- M^3 acumulados: En este caso y dado que sabremos los m^3 en el momento del vaciado la idea es generar una gráfica con los datos totales acumulados de residuo generados en el punto de recogida desde el primer momento en que el dataset nos da información, esta gráfica será siempre creciente. La hipótesis es que esta función crecerá progresivamente y será para ARIMA más sencilla de pronosticar que las anteriores. Dado que en este caso los pronósticos nos dirán los metros cúbicos de residuo acumulados en un momento dado, para conocer el porcentaje de llenado del punto de recogida será necesario restar la cantidad pronosticada a los metros cúbicos acumulados en la recogida anterior y dividir para la capacidad total del puto de recogida en ese momento.

$$F_t = 100\frac{M_t^3 - M_{T_l}^3}{M_{totales\ punto}^3}$$

Ecuación 23. Llenado (%) usando metros cúbicos acumulados de pendiente como predictor

10.5.1. ARIMA para porcentaje de llenado

La gráfica de la serie temporal del porcentaje de llenado tiene los valores acotados en el intervalo [0 – 100] dado que son porcentajes y presenta una forma de dientes de sierra como la que se aprecia en la *Ilustración 31* debido a que se registra el porcentaje de llenado del punto durante la operación de vaciado, por lo que en el mismo instante en que se mide ese porcentaje sabemos que automáticamente el punto queda totalmente libre de residuo, por lo que aparece ese descenso brusco hasta el 0% y por lo tanto esa característica forma.

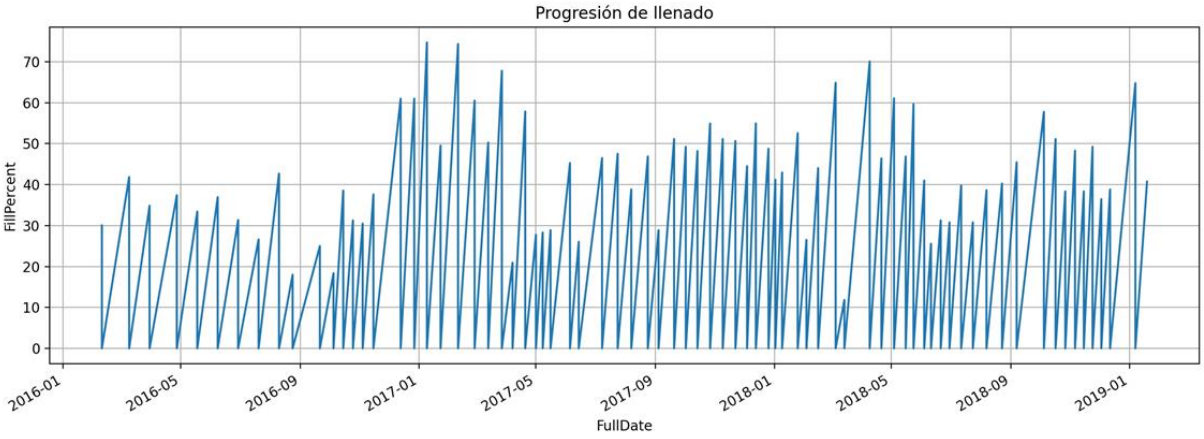


Ilustración 31. Serie temporal de porcentaje de llenado

El modelo ARIMA no deja de ser un método regresivo por lo que esos cambios bruscos harán que una serie de este tipo sea muy difícil de pronosticar por lo que en primer lugar suavizaremos la serie con el método *LOWESS* con un 5% obteniendo una gráfica como la que vemos en la *Ilustración 32* con un aspecto mucho más interesante desde el punto de vista predictivo.

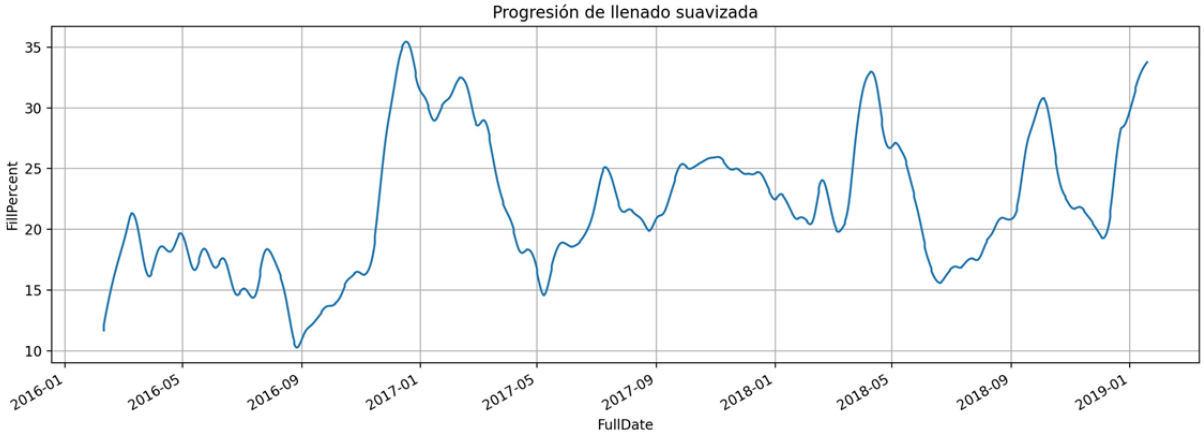


Ilustración 32. Serie temporal de porcentaje de llenado suavizada

A partir de la serie suavizada representaremos el diagrama de cajas y bigotes de la *Ilustración 33* en busca de algún patrón estacional de la serie, dado que trabajamos con datos diarios probamos a generar las cajas tanto por meses como por días de la semana, pero como podemos observar en la gráfica, no parece haber ningún patrón que demuestre estacionalidad en los datos, por lo que podremos utilizar el algoritmo ARIMA sin la componente estacional para realizar los pronósticos.

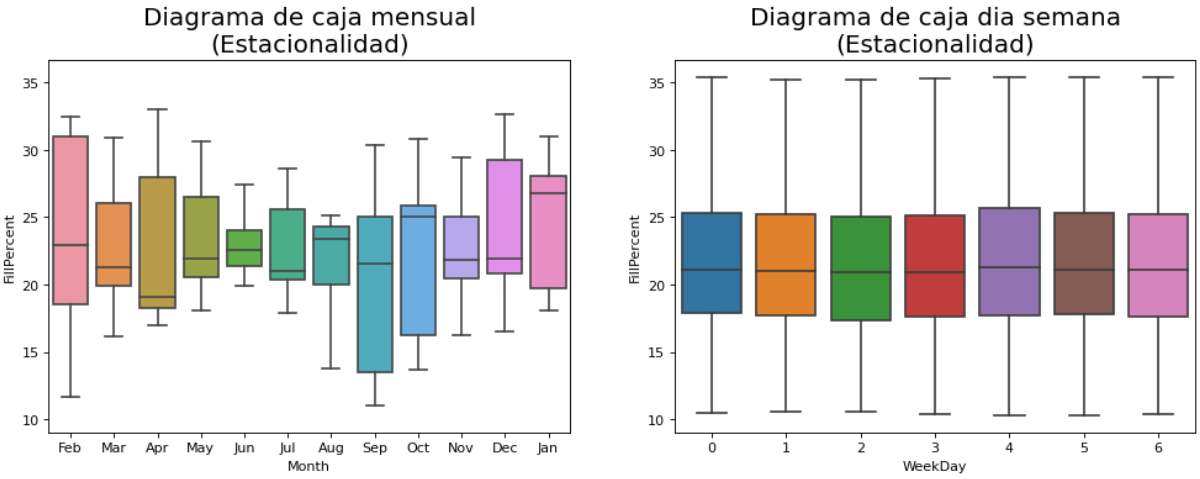


Ilustración 33. Diagrama de cajas y bigotes para porcentaje de llenado

Si observamos la gráfica de residuos (*Ilustración 34*) para el porcentaje de llenado, podemos observar como los residuos se distribuyen alrededor de 0 con una varianza uniforme, lo cual es bueno. El histograma de densidad de residuos presenta una distribución normal entre -2 y 2 centrada en cero. El tercer grafico nos muestra la desviación de los residuos, vemos que salvo unos outliers en los extremos se ajustan perfectamente a la recta roja, lo cual es muy bueno para poder pronosticar después y por último, el gráfico ACF muestra también como desaparece la correlación en pocos rezagos.

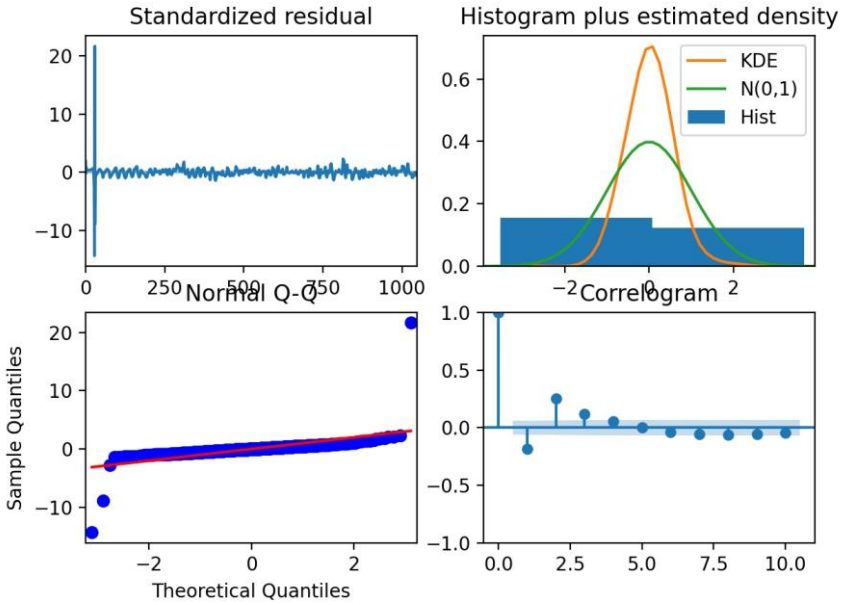


Ilustración 34. Gráfica de residuos para porcentaje de llenado

Para poder aplicar ARIMA es imprescindible que la serie sea estacionaria y para ello podemos aplicar los test que vimos en el punto *6.2.3*. Aplicando la prueba de Dickey Fuller Aumentada (ADF) sobre la serie temporal (función *adfuller* disponible en el paquete statsmodel) obtenemos un $p - valor = 0.0118$ por lo que se rechaza la hipótesis nula y se concluye que la serie es estacionaria, sin embargo el test KPSS arroja un $p - valor = 0.01$ y en este caso, se acepta la hipótesis nula por lo que concluiría que la serie no es estacionaria. Tras una primera diferenciación $p - valor_{ADF} = 4.24e^{-7}$ y $p - valor_{KPSS} = 0.1$ por lo que ahora para ambos test se concluiría que la serie es estacionaria. La librería *pdmarima* tiene un método *ndiffs* que sugiere el mejor valor de *d* para los tres tests mencionados y si probamos a ejecutarla con la serie temporal obtendremos unos valores de $d_{ADF} = 0, d_{KPSS} = 0, d_{PP} = 0$, lo que coincide con los test realizados de forma individual. Dado que hay discrepancias en dos de ellos, escogeremos un valor $d = 1$ que hará la serie estacionaria con los tres test.

En el punto *7.4* habíamos visto que el parámetro *p* corresponde al número rezagos necesarios para cruzar el umbral del intervalo de confianza en el gráfico de autocorrelación parcial (podemos dibujarla con el método *plot_pacf* del paquete *statsmodels*) mientras que para el parámetro *q* seguiremos el mismo criterio pero en este caso utilizando el gráfico de autocorrelación (podemos dibujarla con el método *plot_acf* del paquete *statsmodels*) como se explicó en el punto *7.5*. Si observamos la *Ilustración 35* vemos ambas gráficas para la primera diferenciación (parámetro *d* que hemos calculado previamente). En este caso, el parámetro *p* parece que corresponde a 2 rezagos (el primero de los puntos del *gráfico de piruleta* corresponde a los datos sin desplazar, por lo que la correlación es perfecta) mientras que el número de rezagos del gráfico de autocorrelación hasta que se cruza el umbral de confianza parece ser 1, aunque el punto es muy próximo a los límites del intervalo, por lo que podríamos dudar entre un valor del parámetro *q* de 0 o 1, siendo conservadores podríamos tomar el valor 0.

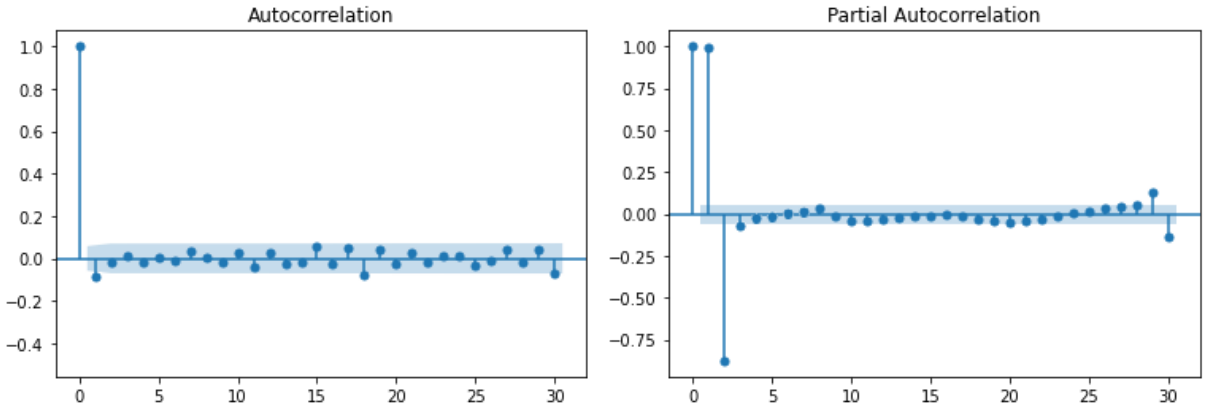


Ilustración 35. Gráfica de autocorrelación y autocorrelación parcial para el porcentaje de llenado

En base a los resultados del estudio obtenidos podríamos establecer los parámetros (p, d, q) de ARIMA con los valores (2, 1, 0). Si los comparamos con los obtenidos por el algoritmo auto_arima este nos devuelve una mejor configuración de (p, d, q) = (2, 1, 0) que coincide con los valores calculados manualmente.

Si replicamos este estudio en varios puntos de recogida, podremos observar como en la *Ilustración 36* se muestran los pronósticos realizados por ARIMA. En la mayoría de ellos vemos que los pronósticos ajustan bien en los primeros valores del conjunto de test, pero rápidamente se van alejando de la realidad. Incluso se aprecian gráficos en los que ARIMA solo ha sido capaz de dar un pronóstico que corresponde a los valores medios de la serie (gráfico esquina inferior derecha).

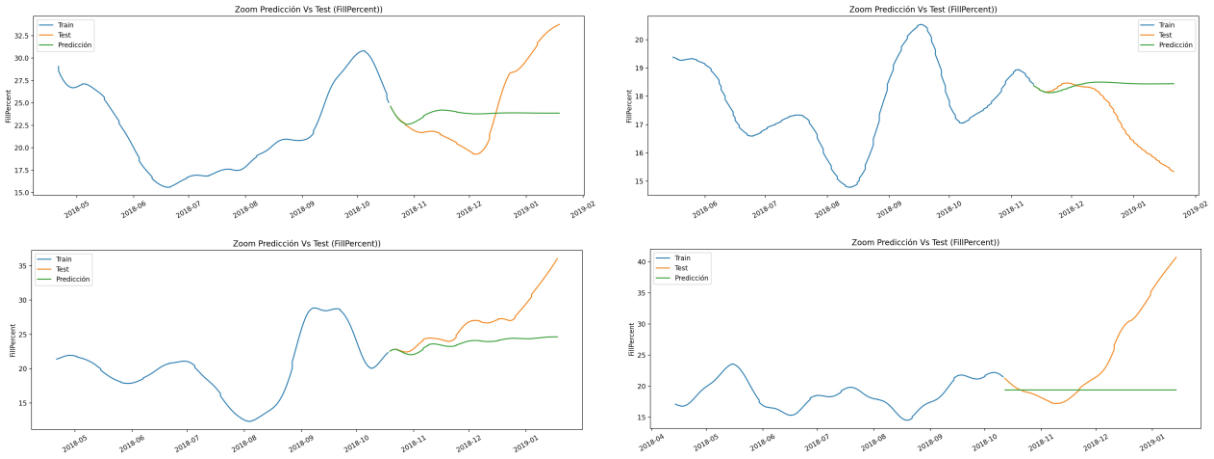


Ilustración 36. Pronóstico de la serie con porcentaje de llenado para varios puntos de recogida del TOP150

A fin de obtener unas métricas globales, se replicará el procedimiento por el dataset *TOP150*, calculando las métricas para cada punto de recogida y promediando

los resultados. En la *Tabla 7* se presenta un resumen donde se han dividido los resultados en cinco grupos para medir los valores en las predicciones a 1 semana, 2 semanas, 1 mes, 2 meses y 3 meses (recordar que, habíamos mencionado que las predicciones a 1 mes serían más que suficiente para optimizar las rutas de recogida).

Periodo de tiempo	Métrica	Media	Máximo	Mínimo	Desviación estándar
1 semana	RMSE	3.132	13.587	0.001	2.744
	MAPE	0.139	0.806	0.000	0.122
	MAE	3.063	13.567	0.001	2.742
	ME	-1.944	7.280	-13.567	3.625
	MPE	-0.066	0.806	-0.431	0.173
2 semanas	RMSE	3.339	13.539	0.014	2.729
	MAPE	0.146	0.941	0.000	0.127
	MAE	3.189	13.507	0.009	2.729
	ME	-1.969	6.786	-13.507	3.683
	MPE	-0.064	0.941	-0.429	0.181
1 Mes	RMSE	3.761	17.613	0.074	2.948
	MAPE	0.157	1.124	0.003	0.132
	MAE	3.432	14.675	0.061	2.779
	ME	-1.928	7.222	-14.675	3.847
	MPE	-0.056	1.124	-0.439	0.192
2 Meses	RMSE	4.235	15.985	0.152	2.987
	MAPE	0.185	1.765	0.006	0.184
	MAE	3.708	12.692	0.126	2.697
	ME	-1.429	7.784	-12.418	3.937
	MPE	-0.018	1.578	-0.394	0.239
3 Meses	RMSE	6.168	19.198	0.449	3.806
	MAPE	0.277	6.390	0.015	0.790
	MAE	4.853	15.838	0.318	2.923
	ME	-2.408	11.122	-15.838	4.255
	MPE	-0.019	3.825	-1.074	0.620

Tabla 7. Métricas de rendimiento para porcentaje de llenado

En los resultados podemos observar como el MAPE va creciendo a medida que aumentamos el tiempo de predicción, desde 0.139 en una semana hasta 0.277 en tres meses, lo cual implicaría que el algoritmo tiene una exactitud de un 86.1% en una semana y va disminuyendo hasta un 72.3% en tres meses. Los resultados no son especialmente buenos, además, si observamos nuevamente las imágenes *Ilustración 31* e *Ilustración 32* veremos que debemos ser cautos con la interpretación puesto que las gráficas suavizadas cambian (para este predictor) radicalmente respecto a la real (debemos suavizarla porque con la gráfica original de dientes de sierra ARIMA es incapaz de obtener resultados). Los datos originales muestran la progresión de

llenado de un punto de recogida desde un vaciado hasta el siguiente, por lo que permiten comprobar para una fecha el porcentaje de llenado al que se encuentra, sin embargo, la serie suavizada nunca llega al 0% de llenado, se ha perdido la información de los vaciados y la evolución de llenado por lo que, por muy buena que sea la métrica MAPE sobre la serie suavizada no resultará de utilidad para optimizar las rutas de recogida. Evidentemente, podríamos habernos ahorrado todo el proceso de entrenamiento y validación al comprobar la diferencia entre los datos reales y suavizados, pero hemos preferido realizar el proceso completo para comprobar el peligro que puede suponer una mala interpretación de los resultados.

Así pues, vistos los resultados podríamos concluir que el porcentaje de llenado no es un buen predictor para el objetivo de este trabajo de fin de master.

10.5.2. ARIMA para pendiente de llenado

Descartado el uso del porcentaje de llenado como predictor, lo siguiente que consideramos en nuestro estudio es generar la serie temporal no con el porcentaje sino con el ángulo de la pendiente de la gráfica. Los valores en este caso siempre estarán acotados en el intervalo [0-90] y la gráfica no tendrá esos dientes de sierra que presentaba la anterior, que creemos que los valores serán más regulares, lo que nos lleva a pensar que podría mejorar los resultados obtenidos.

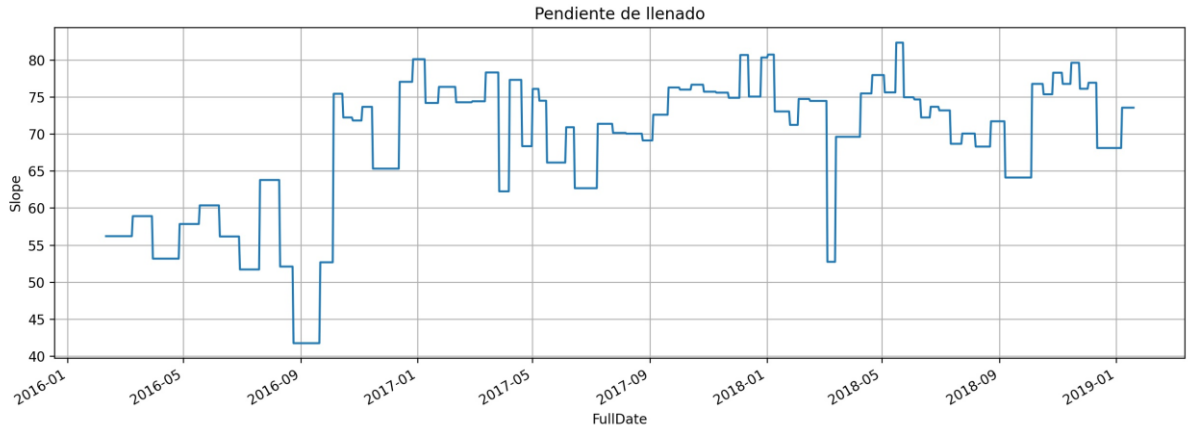


Ilustración 37. Serie temporal de pendiente de llenado

En la imagen Ilustración 37 vemos el aspecto de la gráfica de la serie temporal de la pendiente de llenado del punto de recogida. Igual que en el caso anterior suavizaremos los datos con *LOWESS* con un 5% obteniendo la gráfica de la *Ilustración 38*. Esta gráfica presenta un aspecto mucho más similar a la no suavizada

que el predictor anterior, es una gráfica más ajustada a la realidad y por lo tanto, si las métricas de rendimiento son aceptables podría ser un buen candidato.

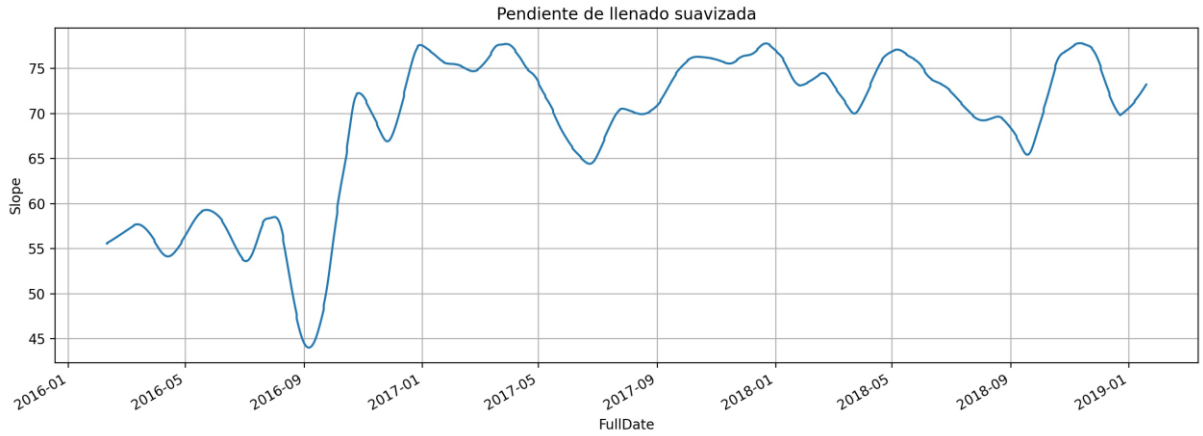


Ilustración 38. Serie temporal de pendiente de llenado suavizada

A partir de la serie suavizada representaremos el diagrama de cajas y bigotes de la *Ilustración 39* en busca de algún patrón estacional de la serie, pero como podemos observar en la gráfica, no se aprecia ninguno ni en la gráfica de valores mensuales ni en la de valores diarios, por lo que supondremos que la serie no es estacional.

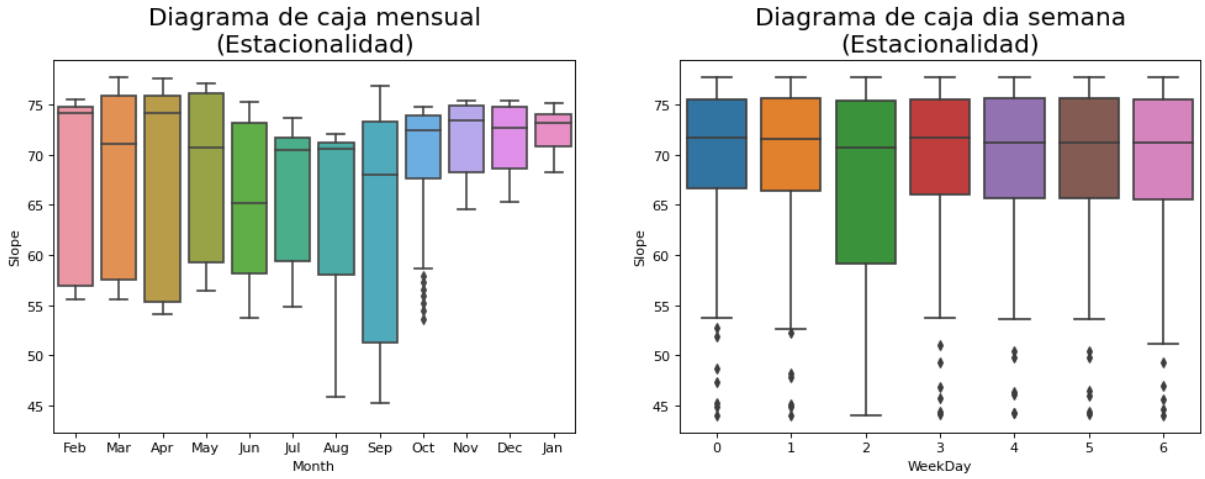


Ilustración 39. Diagrama de cajas y bigotes para la pendiente de llenado

Si observamos la gráfica de residuos (*Ilustración 40*) para el este tipo de predictor, podemos observar cómo, igual que en el caso anterior los residuos se distribuyen alrededor de 0 con una varianza uniforme y el histograma de densidad de residuos presenta una distribución normal entre -2 y 2 centrada en cero. El tercer grafico nos muestra la desviación de los residuos, vemos que salvo unos outliers en

los extremos se ajustan perfectamente a la recta roja, lo cual es muy bueno para poder pronosticar después y por último, el gráfico ACF muestra también como desaparece la correlación en pocos rezagos.

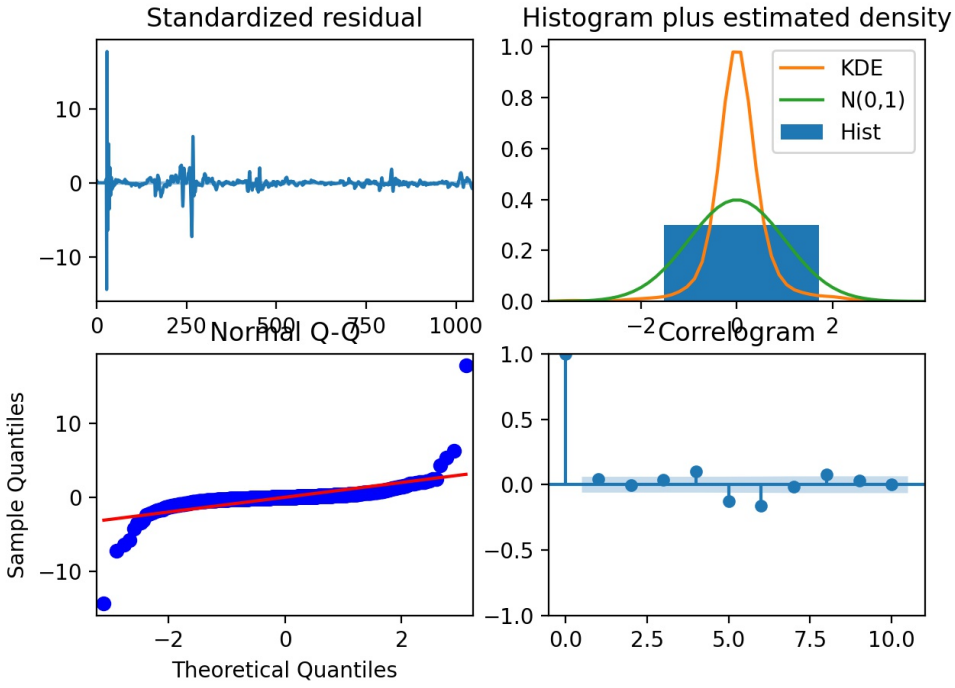


Ilustración 40. Gráfica de residuos para pendiente de llenado

Debemos comprobar que la serie es estacionaria para poder aplicar ARIMA y en caso contrario diferenciarla tantas veces como sea necesario para conseguirlo. En este caso, los valores para los test ADF y KPSS para la serie original son $p - valor_{ADF} = 0.43$ y $p - valor_{KPSS} = 0.01$ lo cual nos indica que la serie original no es estacionaria. Tras una primera diferenciación los valores cambian a $p - valor_{ADF} = 1.609e^{-12}$ y $p - valor_{KPSS} = 0.1$ por lo que tras una diferenciación, ambos test prueban la serie como estacionaria. Si lo comparamos con los resultados obtenidos por la función *ndiffs* de *pdmarima*, esta nos sugiere un mejor valor de $d_{ADF} = 1, d_{KPSS} = 1, d_{PP} = 1$. En este caso todos los test coinciden así que escogeremos un valor $d = 1$ que hará la serie estacionaria.

Si continuamos utilizando la misma técnica que en el apartado anterior para obtener los parámetros *p* y *q*, vemos que en el gráfico de autocorrelación parcial (*Ilustración 41*) todos los valores parecen próximos al intervalo de confianza, el gráfico en si es difícil de interpretar porque los límites del intervalo son muy próximos a cero

y varios de los outliers hacen que no se visualice la información claramente. Podemos dibujar el gráfico con menos "*lags*" o retrasos, pero incluso el retraso 5 hace que sea de difícil interpretación saber en qué punto cortan los puntos. Por el contrario el gráfico de autocorrelación es bastante claro y parece una elección clara un $q = 1$.

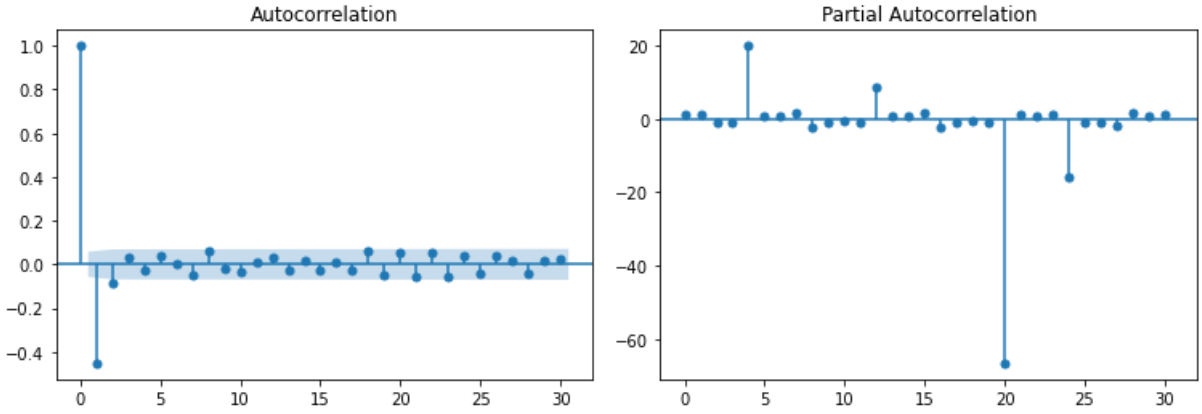


Ilustración 41. Gráfica de autocorrelación y autocorrelación parcial para la pendiente de llenado

Si probamos el método *auto_*arima en este caso obtenemos unos valores de (p, d, q) = (10, 1, 1), lo que indica que en el gráfico de autocorrelación, en el que nos resultaba tan difícil interpretar los valores hay 10 rezagos fuera de los límites hasta que uno de ellos lo cruza. Vemos como los valores para *d* y para *q* sí que han coincidido con los que hemos calculado manualmente.

Continuamos pues replicando el estudio en varios puntos de recogida, podremos observar como en la *Ilustración 42* se muestran los pronósticos realizados por ARIMA. En la mayoría de ellos vemos que los pronósticos ajustan bien en los primeros valores del conjunto de test, pero rápidamente se van alejando de la realidad. Incluso se aprecian gráficos en los que ARIMA solo ha sido capaz de dar un pronóstico que corresponde a los valores medios de la serie (gráfico esquina superior izquierda).

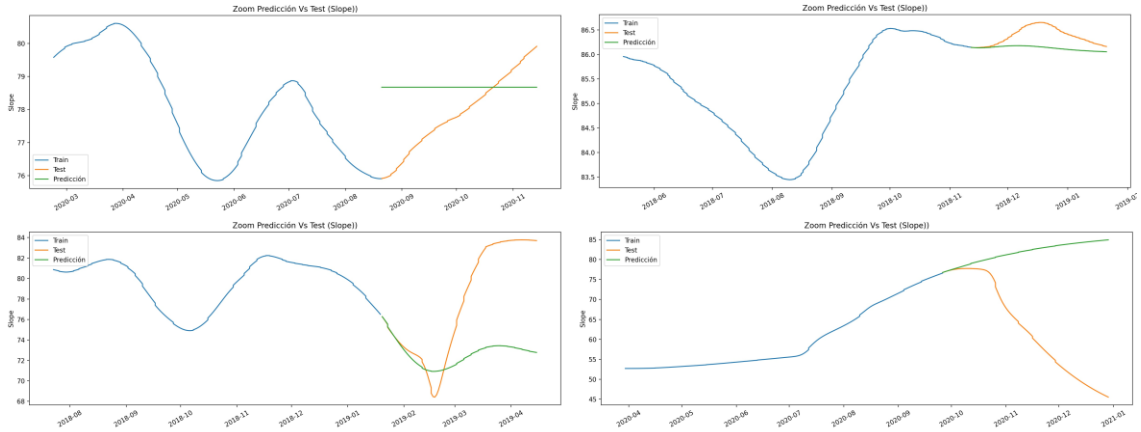


Ilustración 42. Pronóstico de la serie con pendiente de llenado para varios puntos de recogida del TOP150

En la *Tabla 8* se presentan los valores para los valores medios de las métricas de rendimiento del algoritmo utilizando la pendiente de llenado como predictor en el dataset *TOP150*. En este caso nos encontramos con exactitudes de 92,1% para predicciones a una semana y 80% para predicciones a 3 meses.

Periodo de tiempo	Métrica	Media	Máximo	Mínimo	Desviación estándar
1 semana	RMSE	4.130	27.614	0.000	5.662
	MAPE	0.079	1.209	0.000	0.160
	MAE	4.086	27.614	0.000	5.671
	ME	-0.095	27.614	-17.989	6.996
	MPE	0.020	1.209	-0.274	0.177
2 semanas	RMSE	4.386	27.554	0.012	5.664
	MAPE	0.082	1.209	0.000	0.163
	MAE	4.273	27.554	0.007	5.674
	ME	-0.152	27.554	-16.984	7.097
	MPE	0.020	1.209	-0.263	0.182
1 Mes	RMSE	4.962	31.180	0.041	5.700
	MAPE	0.090	1.257	0.000	0.177
	MAE	4.667	30.523	0.036	5.642
	ME	-0.284	30.523	-14.662	7.280
	MPE	0.020	1.257	-0.234	0.197
2 Meses	RMSE	6.140	45.269	0.132	6.424
	MAPE	0.115	2.355	0.001	0.274
	MAE	5.409	37.620	0.092	5.894
	ME	0.068	37.620	-20.232	7.767
	MPE	0.040	2.355	-0.285	0.293
3 Meses	RMSE	7.633	53.941	0.332	7.335
	MAPE	0.200	6.355	0.003	0.667
	MAE	6.486	46.040	0.273	6.459
	ME	0.508	46.008	-20.940	8.312
	MPE	0.048	3.044	-0.492	0.360

Tabla 8. Métricas de rendimiento para pendiente de llenado

Aunque en las predicciones a una semana con este predictor solo tenemos un 7,9% de error, otro parámetro en el que hay que fijarse es en la desviación estándar. En este caso obtenemos una desviación de un 16% y valores máximos de error de hasta el 120%, lo cual significa que en esos casos, los valores no se parecerán en nada a la realidad.

Podríamos concluir que en función del comportamiento del punto de recogida hay puntos para los que el predictor mantiene un porcentaje de error no demasiado alto pero hay otros en los que no ofrece seguridad, por lo que continuaremos el análisis con los siguientes predictores.

10.5.3. ARIMA para seno de la pendiente de llenado

Una vez probados los dos primeros predictores y obteniendo unas métricas que no han resultado demasiado esperanzadoras pensamos que una posibilidad será aplicar el seno o el coseno sobre la pendiente de llenado (variable anterior) ya que de esta forma los valores quedarán acotados en el intervalo [-1, 1] y ese cambio de escala, unido al suavizado de la serie puede ayudar a ARIMA a mejorar sus resultados.

En la parte superior de la *Ilustración 43* se ha dibujado la gráfica del seno de la pendiente y en la parte inferior la misma gráfica suavizada con *LOWESS* con un 5%. Vemos que como ocurría con la pendiente de llenado, el suavizado mantiene una forma similar a la original, algo que como ya hemos nombrado antes es bueno para obtener unos resultados próximos a la realidad.

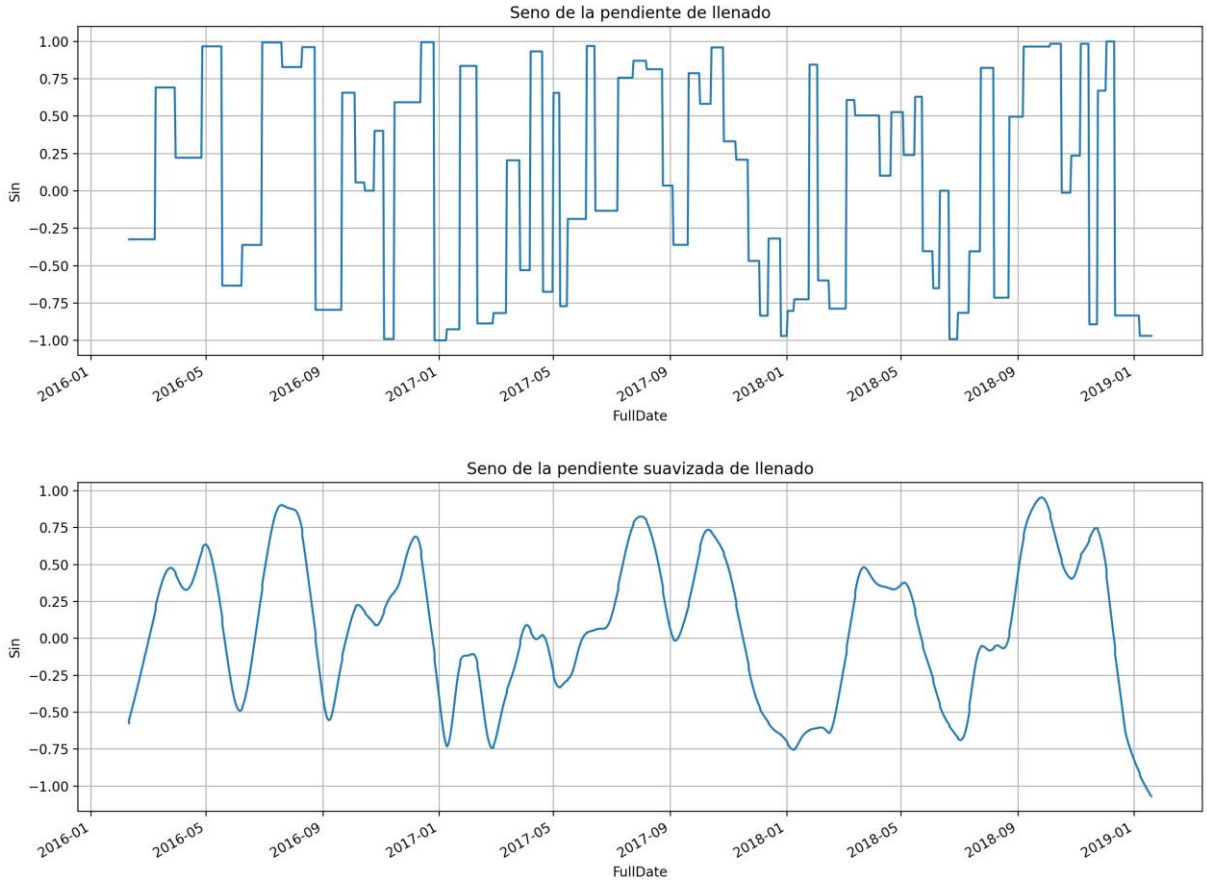


Ilustración 43. Gráfica del seno de la pendiente de llenado original y suavizada

A partir de la serie suavizada representaremos el diagrama de cajas y bigotes de la *Ilustración 44* en busca de algún patrón estacional de la serie, igual que hemos hecho con las pero como podemos observar en la gráfica, no parece haber ninguno ni a nivel mensual ni diario, por lo que podremos utilizar el algoritmo ARIMA sin la componente estacional para realizar los pronósticos.

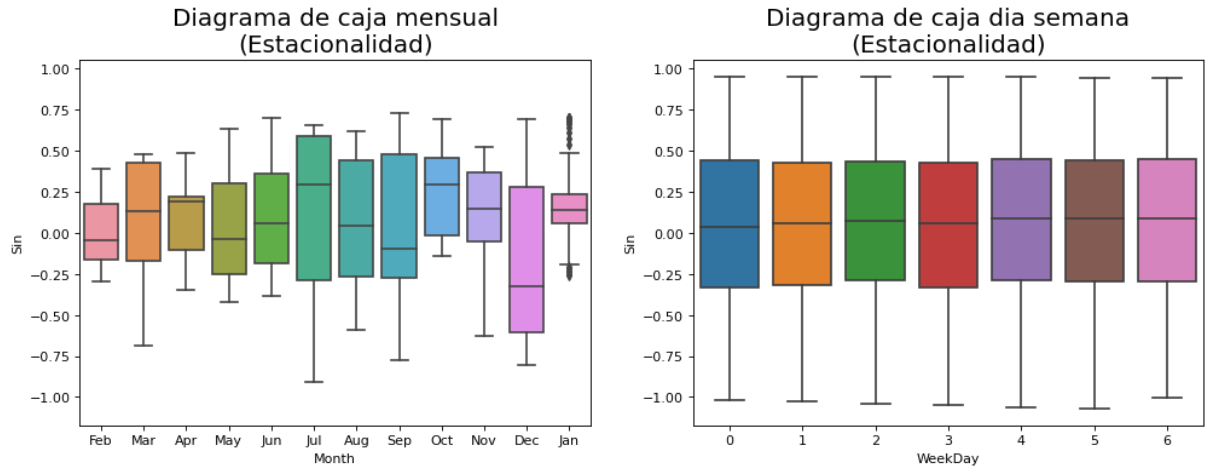


Ilustración 44. Diagrama de cajas y bigotes para el seno de la pendiente de llenado

Si observamos la gráfica de residuos (*Ilustración 45*) aunque los dos primeros gráficos presentan un aspecto similar a los de los anteriores predictores vemos que en el tercer gráfico los residuos se empiezan a desviar de la recta de mejor ajuste y que el gráfico ACF muestra una fuerte correlación pese aun con el paso del tiempo por lo que a priori, aunque habíamos da la impresión de que el resultado no mejorará el anterior.

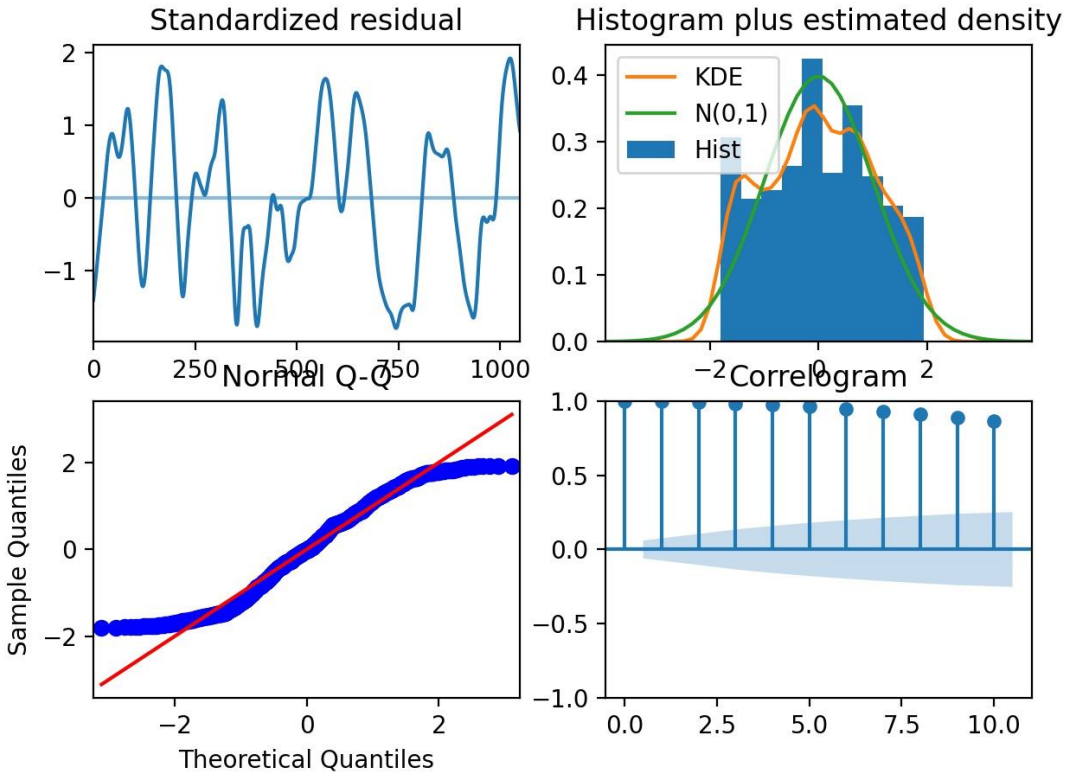


Ilustración 45. Gráfica de residuos para seno de la pendiente de llenado

Aplicaremos los mismos test que hemos usado en los anteriores apartados para comprobar si la serie es estacionaria. En este caso, los valores para los test ADF y KPSS para la serie original son $p - valor_{ADF} = 0.003$ y $p - valor_{KPSS} = 0.1$ lo que quiere decir que la serie es estacionaria sin tener que aplicar ninguna diferenciación. Al comparar los valores con la función *ndiffs* de *pdmarima*, esta nos sugiere un mejor valor de $d_{ADF} = 0, d_{KPSS} = 0, d_{PP} = 1$. Por lo que, aunque el test *PP* da un resultado distinto a los anteriores escogeremos un valor para $d = 0$.

El siguiente punto será escoger los valores para p y q a través de los gráficos PACF y ACF. En la *Ilustración 46* vemos como en el gráfico de autocorrelación parcial

hay cuatro rezagos por encima del intervalo de confianza mientras que el gráfico de autocorrelación es muy similar al que habíamos visto en el gráfico de distribución de los residuos y presenta unos 27 rezagos por encima del límite incluso con la serie estacionaria. El modelo ARIMA no espera unos valores elevados para p y q, por lo que un valor como este de q no tiene demasiado sentido. Podemos corregirlo haciendo una diferenciación, la cual mantendrá la serie estacionaria pero mejorará los valores para p y q dejándolos en *p= 2* y *q = 1*, algo más próximo a lo que se espera.

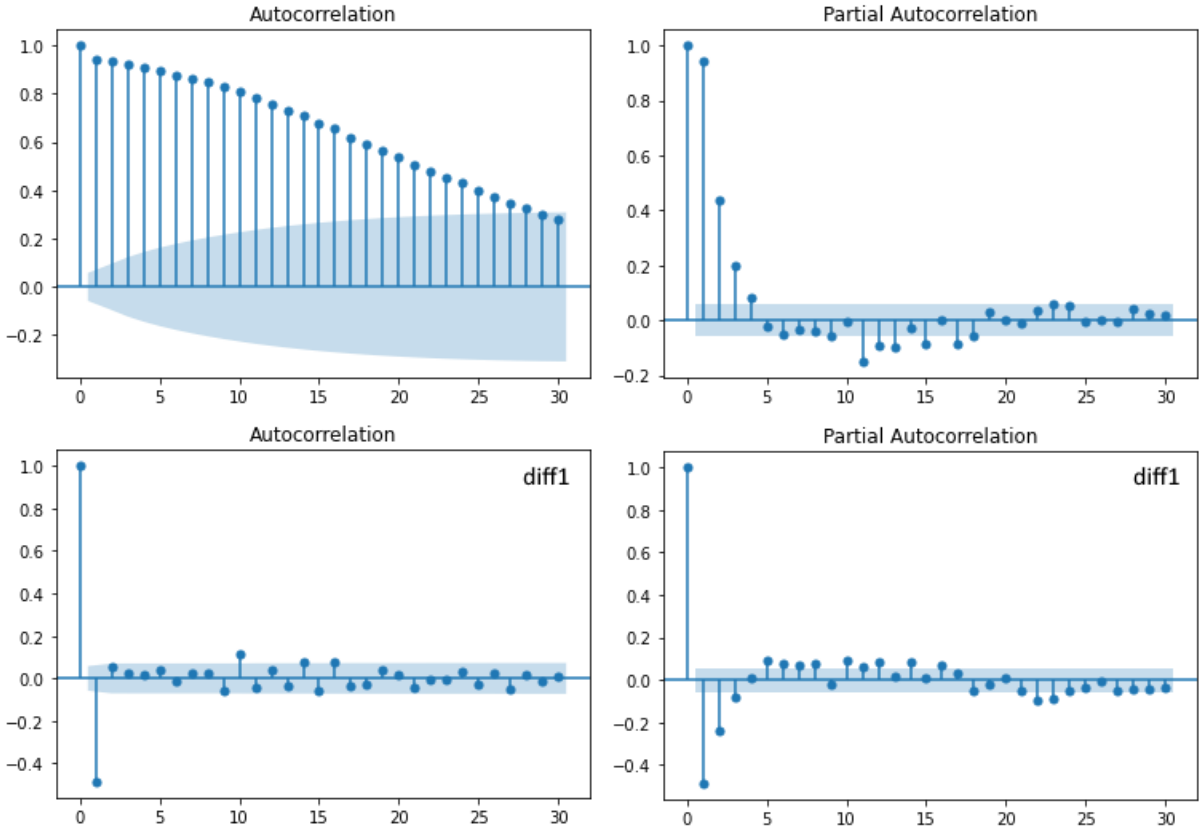


Ilustración 46. Gráfica de autocorrelación y autocorrelación parcial para el seno de la pendiente de llenado

De esta forma, y después de ajustar el valor *d* para obtener unos buenos parámetros en *p* y *q* la configuración manual quedaría como (2, 1, 1). *Auto_arima* sin embargo nos devuelve unos resultados (p, d, q) = (0, 0, 0). Vemos que dado que la serie era estacionaria sin diferenciaciones *auto_arima* la ha mantenido así y, dado que los valores de *p* y *q* eran demasiado elevados, como hemos visto en el análisis manual, ha optado por dejarlos a 0.

Todo sigue indicando que el predictor no va a arrojar buenos resultados y si trasladamos el análisis a varios puntos de recogida, podremos observar como en la

Ilustración 47 se muestran los pronósticos realizados. En la mayoría de ellos vemos que el algoritmo únicamente ha sido capaz de trazar una recta con los valores medios de la serie temporal. Si bien es cierto que la escala de valores en los que se mueve la gráfica es muy pequeña, son resultados que indican que este predicitor no es adecuado.

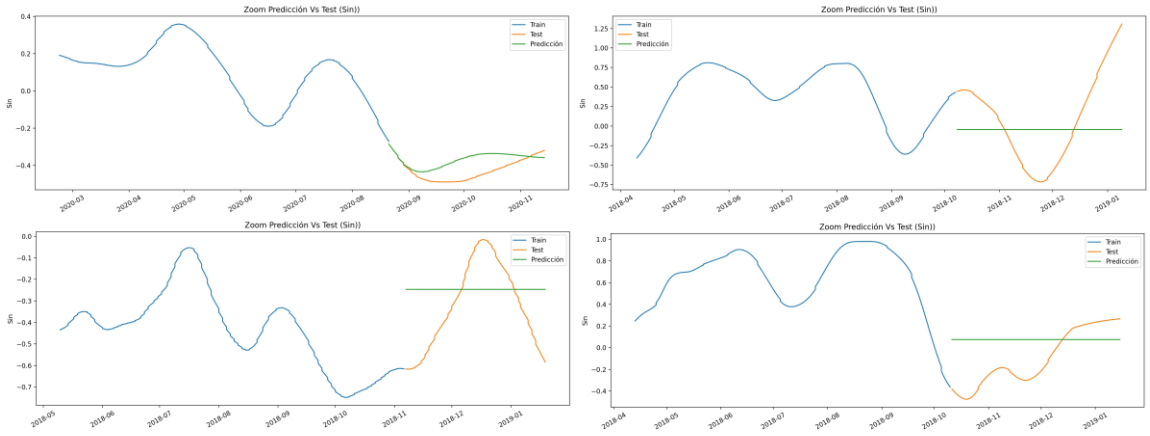


Ilustración 47. Pronóstico de la serie con seno de la pendiente para varios puntos de recogida del TOP150

Aunque ya vamos viendo que tendremos que descartar esta variable, igual que hemos hecho con las dos anteriores lanzaremos el proceso para el *TOP150* y representaremos en la *Tabla 9* los valores medios de las métricas. En este caso nos encontramos unos valores para MAPE muy altos incluso para una sola semana de pronóstico.

Periodo de tiempo	Métrica	Media	Máximo	Mínimo	Desviación estándar
1 semana	RMSE	0.324	0.962	0.000	0.266
	MAPE	1.137	9.732	0.000	1.435
	MAE	0.319	0.962	0.000	0.267
	ME	-0.013	0.932	-0.962	0.416
	MPE	-0.678	5.635	-5.181	1.171
2 semanas	RMSE	0.342	0.965	0.001	0.255
	MAPE	1.433	18.283	0.000	2.121
	MAE	0.328	0.960	0.000	0.258
	ME	-0.008	0.960	-0.936	0.416
	MPE	-0.620	8.396	-17.508	2.028
1 Mes	RMSE	0.389	0.961	0.012	0.239
	MAPE	1.626	27.739	0.017	1.688
	MAE	0.360	0.958	0.008	0.239
	ME	0.012	0.958	-0.924	0.421
	MPE	-0.822	24.662	-8.515	1.514

2 Meses	RMSE	0.442	1.019	0.022	0.216
	MAPE	2.388	62.665	0.035	2.984
	MAE	0.390	0.941	0.017	0.204
	ME	0.033	0.941	-0.895	0.389
	MPE	-0.674	53.104	-20.635	2.143
3 Meses	RMSE	0.503	1.327	0.052	0.218
	MAPE	2.306	41.725	0.093	2.638
	MAE	0.436	1.098	0.036	0.205
	ME	0.026	1.098	-0.896	0.374
	MPE	-0.660	34.682	-14.886	1.727

Tabla 9. Métricas de rendimiento para el seno de la pendiente de llenado

Volviendo a la *Ilustración 47* vemos que las predicciones, pese a que no son buenas, ya que ARIMA se limita a ofrecer el valor medio de la serie como resultado no son tan alejadas de la realidad como para ofrecer esos valores de MAPE tan elevados. ¿Por qué ocurre entonces? Porque para etiquetas cercanas a 0, dado que la resta entre valores se divide por ella, el MAPE tiende a subir produciendo una falsa sensación de desviación. De hecho, si nos fijamos en el error medio (ME) para pronósticos a 2 meses vemos que el valor medio es de 0.012 y la desviación estándar de 0.421, lo que no justifica esos valores tan altos de MAPE salvo por el caso de etiquetas cercanas a cero.

Con independencia de ello, los resultados no son buenos, por lo que continuaremos estudiando los dos predictores restantes.

10.5.4. ARIMA para coseno de la pendiente de llenado

En el punto anterior habíamos comentado que nuestra suposición era que podríamos mejorar los resultados del pronóstico de la pendiente de llenado con las funciones *seno* o *coseno* pero ya hemos visto que la función *seno* no nos servirá para nuestro propósito, lo cual nos lleva a pensar que la función *coseno* tendrá un comportamiento similar a la anterior. Aun así, analizaremos los resultados obtenidos durante las pruebas.

Para ello, como en los casos anteriores comenzaremos dibujando las gráficas de la serie temporal con sus valores reales y suavizados en la *Ilustración 48*. Los valores de la función coseno están limitados en el intervalo [-1, 1] y como vimos para la pendiente y para el seno el suavizado respeta muy bien los valores originales de la serie por lo que no debemos preocuparnos en ese sentido.

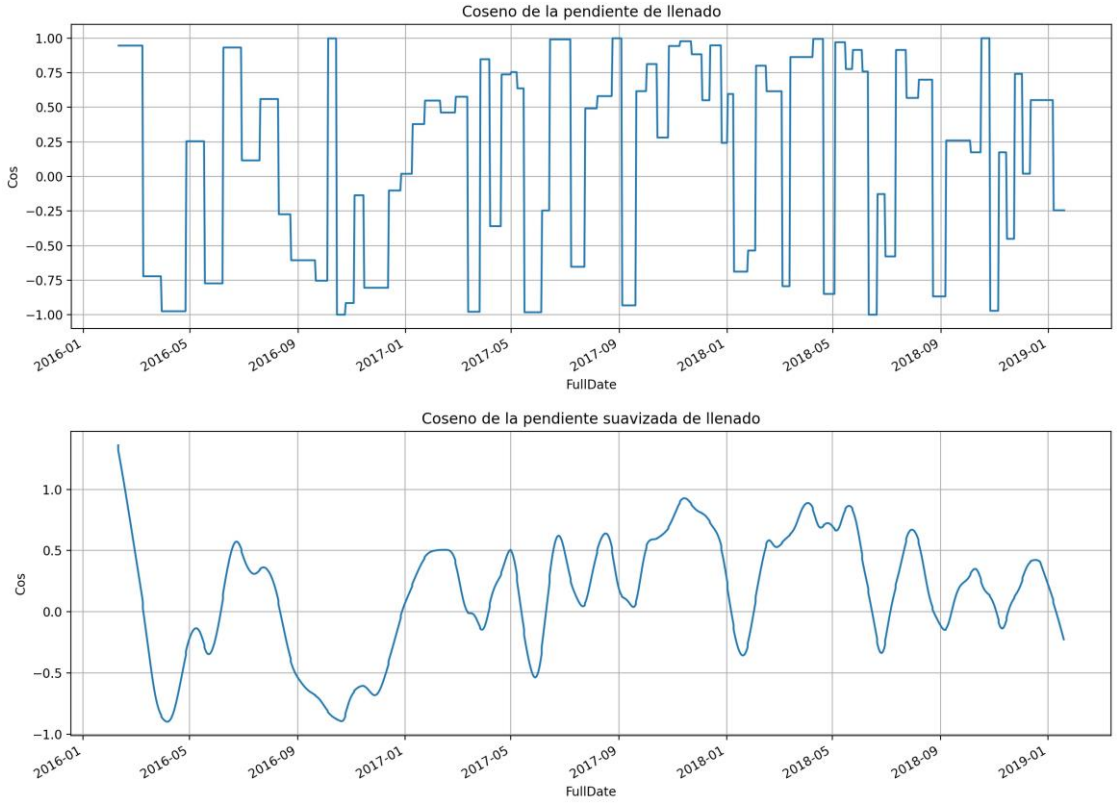


Ilustración 48. Gráfica original y suavizada del coseno de la pendiente de llenado

A partir de la serie suavizada representaremos el diagrama de cajas y bigotes de la *Ilustración 49* en busca de algún patrón estacional de la serie, pero como también ocurría con los predictores analizados con anterioridad no se observa ninguno, por lo que podremos utilizar el algoritmo ARIMA sin la componente estacional para realizar los pronósticos.

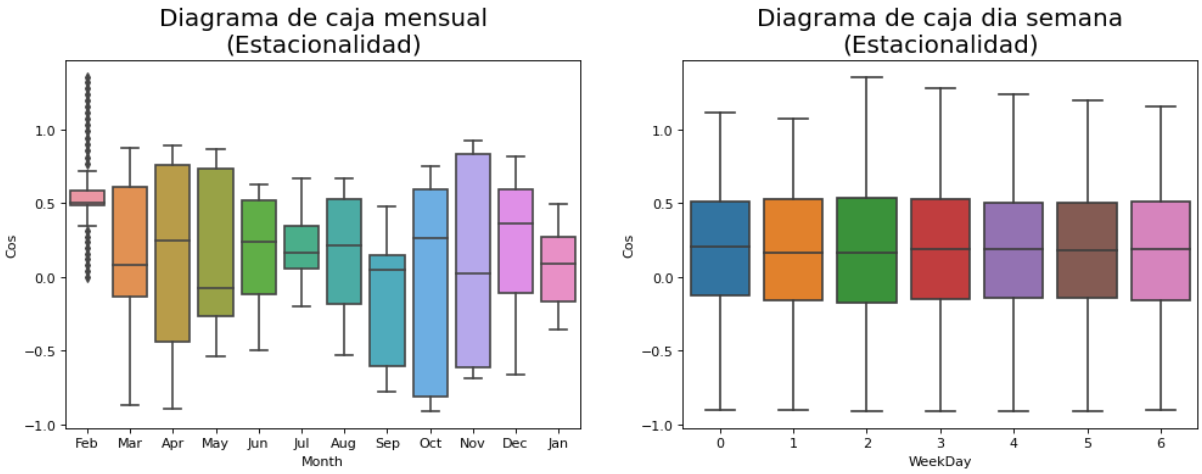


Ilustración 49. Diagrama de cajas y bigotes para coseno de la pendiente de llenado

En la gráfica de residuos (*Ilustración 50*) ocurre algo similar que para el seno. Los dos primeros gráficos presentan un aspecto adecuado pero los dos siguientes no, por lo que como ya adelantábamos al inicio de este punto sospechamos que los resultados que obtendremos serán muy similares a los de la función seno.

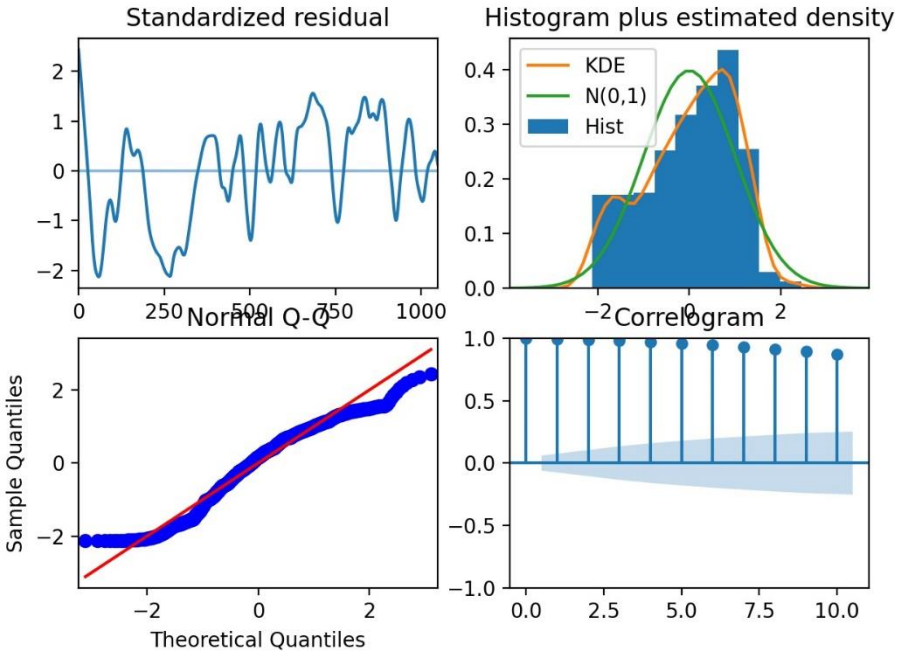


Ilustración 50. Gráfica de residuos para el coseno de la pendiente de llenado

El test *ADF* nos indica que la serie es estacionaria con una diferenciación mientras que *KPSS* nos dice que deberíamos hacer una. Son los mismos resultados que los que nos devuelve la función *ndiffs* que además, dice a través del test *PP* que la serie no necesita diferenciación para ser estacionaria.

Partiendo entonces de la base de que la serie es estacionaria sin diferenciaciones, el siguiente punto será escoger los valores para p y q a través de los gráficos PACF y ACF. En la *Ilustración 51* vemos los gráficos que, como ocurría con la función *seno*, presentan un comportamiento realmente malo para la serie sin diferenciación. Pese a que la serie es estacionaria sin diferenciaciones, igual que hicimos antes aplicaremos una diferenciación y observaremos como ahora las gráficas se parecen más a lo que esperamos. Podríamos determinar *p=3* y *q=4* para una diferenciación *d=1*.

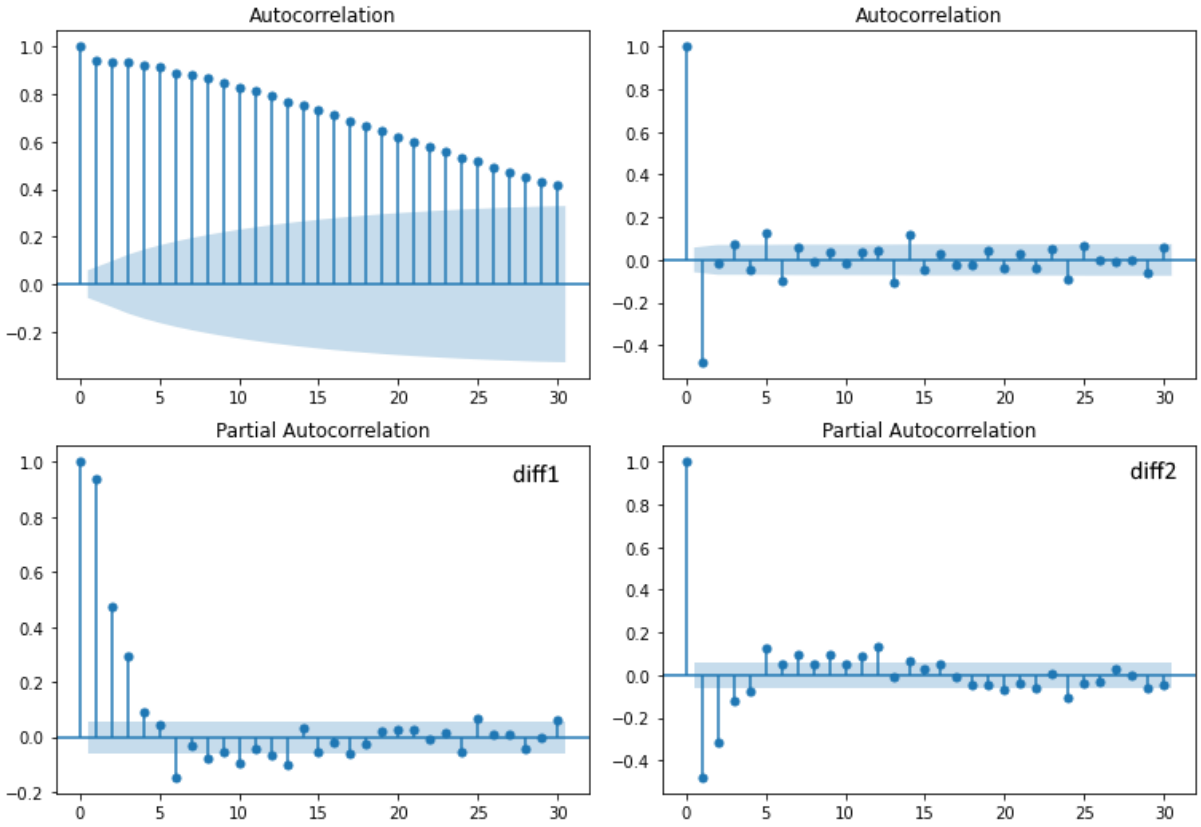


Ilustración 51. Gráfica de autocorrelación y autocorrelación parcial para el coseno de la pendiente de llenado

Sin embargo, igual que ocurrió con la función *seno* a*uto_arima* nos devuelve unos resultados (p, d, q) = (0, 0, 0). Como ocurría antes ha mantenido el parámetro *d=0* porque la serie no requería de diferenciaciones para convertirla en estacionaria y dado que no ha sido capaz de encontrar unos buenos resultados a dejado *p* y *q* con un valor 0.

Seguimos los mismos pasos que con el *seno* y todo sigue indicando que el predictor no va a arrojar buenos resultados. Si trasladamos el análisis a varios puntos de recogida, podremos observar como en la *Ilustración 52* se muestran los pronósticos realizados por ARIMA. Ahora, en todos los casos vemos que el algoritmo únicamente ha sido capaz de trazar una recta con los valores medios de la serie temporal. Si bien es cierto que la escala de valores en los que se mueve la gráfica es muy pequeña, son resultados que confirman nuestras sospechas de que el comportamiento iba a ser prácticamente igual que el de la función *seno*.

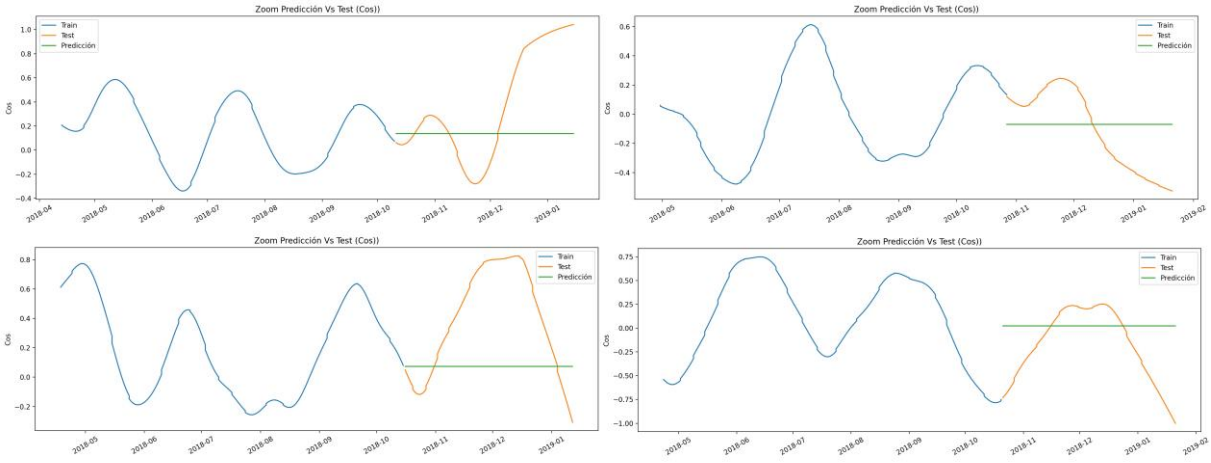


Ilustración 52. Pronóstico de la serie con seno de la pendiente para varios puntos de recogida del TOP150

Al analizar los resultados medios para *TOP150* podemos ver en la *Tabla 10* como era de esperar que nos encontramos con la misma situación que en el punto anterior. La función *coseno*, igual que el *seno*, escala los valores en el intervalo -1 1, por lo que igual que antes tenemos unos MAPE muy elevados que no corresponden con el valor medio del ME ni con su desviación estándar. Nuevamente, la cercanía a cero de las etiquetas hace que se dispare el valor del error porcentual medio absoluto.

Periodo de tiempo	Métrica	Media	Máximo	Mínimo	Desviación estándar
1 semana	RMSE	0.317	1.011	0.000	0.262
	MAPE	1.474	52.334	0.000	4.401
	MAE	0.311	1.011	0.000	0.262
	ME	-0.036	1.011	-0.928	0.406
	MPE	-0.408	52.334	-9.254	4.526
2 semanas	RMSE	0.335	1.005	0.005	0.248
	MAPE	1.781	45.157	0.003	4.426
	MAE	0.319	1.003	0.003	0.250
	ME	-0.032	1.003	-0.922	0.403
	MPE	-1.026	26.833	-41.906	4.264
1 Mes	RMSE	0.383	1.109	0.000	0.223
	MAPE	1.949	49.268	0.000	3.409
	MAE	0.348	1.103	0.000	0.221
	ME	-0.049	1.103	-0.916	0.396
	MPE	-1.300	49.268	-35.581	3.195
2 Meses	RMSE	0.443	1.120	0.005	0.204
	MAPE	1.949	27.783	0.003	2.118
	MAE	0.391	1.117	0.003	0.193
	ME	-0.034	1.117	-0.887	0.386
	MPE	-0.997	23.664	-13.089	1.801

3 Meses	RMSE	0.488	1.132	0.012	0.191
	MAPE	2.064	30.369	0.010	2.509
	MAE	0.422	1.130	0.008	0.178
	ME	0.003	1.130	-0.802	0.373
	MPE	-0.836	15.585	-17.727	1.795

Tabla 10. Métricas de rendimiento para el coseno de la pendiente de llenado

Así pues descartaremos este predictor por sus malos resultados como habíamos hecho con la función *seno*.

10.5.5. ARIMA para metros cúbicos acumulados

El último predictor con el que intentamos realizar pronósticos con ARIMA son los metros cúbicos acumulados de residuo. En este caso los valores que tome la serie no estarán contenidos en ningún intervalo de valores pues irán creciendo a medida que avance el tiempo. Como venimos haciendo con el resto de variables objetivo, el primer paso será representar las gráficas de la serie original y suavizada en la *Ilustración 53.* Se puede observar como la gráfica suavizada tiene prácticamente la misma forma que la original solo que elimina los pequeños escalones de la original. Además, si pensamos en la descomposición de componentes de una serie temporal esta gráfica presenta un componente tendencia muy marcado, ¿por qué? Precisamente porque tal y como hemos dicho al ser valores acumulados siempre irá creciendo.

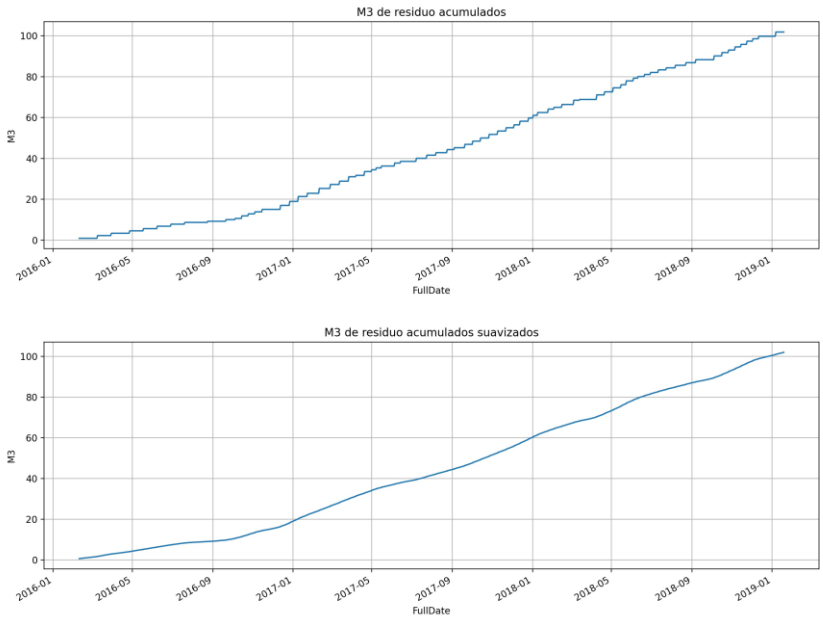


Ilustración 53. Gráfica de metros cúbicos acumulados original y suavizada

A partir de la serie suavizada representaremos el diagrama de cajas y bigotes de la *Ilustración 54* en busca de algún patrón estacional de la serie. No observamos más que el comportamiento esperado, a medida que avanzan los meses el residuo va creciendo por la tendencia de la serie. En este caso hemos dibujado el diagrama tanto para meses como para días de la semana, donde vemos, en este último caso que no aparece ningún patrón diario.

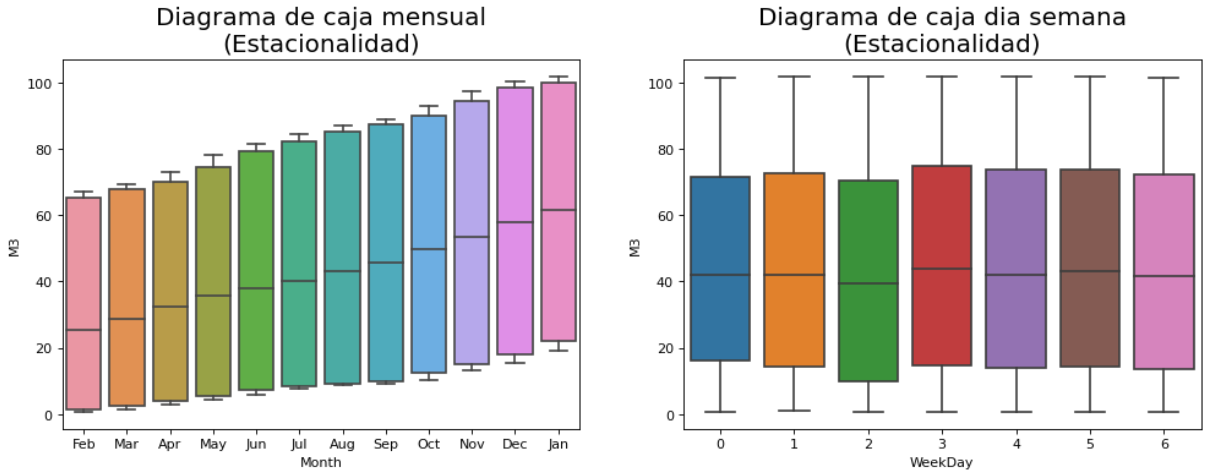


Ilustración 54. Diagrama de cajas y bigotes para los metros cúbicos acumulados

Continuamos como con el resto de predictores que hemos probado para ARIMA con la gráfica de residuos (*Ilustración 55*) para el último predictor, vemos que volvemos a tener unos buenos resultados. Los residuos se distribuyen alrededor de 0 con una varianza uniforme, el histograma de densidad de residuos presenta una distribución normal, la desviación de los residuos se ajusta salvo dos outliers a la recta de mejor ajuste y el gráfico ACF muestra también como desaparece la correlación en pocos rezagos.

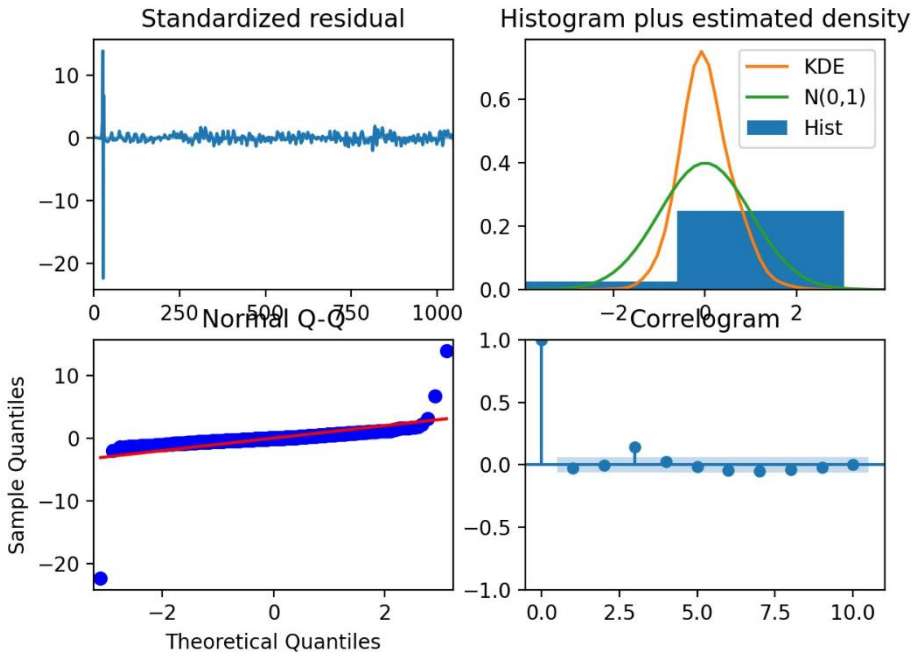


Ilustración 55. Gráfica de residuos para metros cúbicos acumulados

Debemos comprobar si la serie es estacionaria y convertirá en caso contario para poder aplicar ARIMA. Tanto con el test ADF como con el KPSS debemos ir hasta la segunda diferenciación para obtener unos valores $p - valor_{ADF} = 0.035$ y $p - valor_{KPSS} = 0.1$ que nos indican que con un grado de diferenciación 2 la serie ya se convierte en estacionaria. Idénticos valores obtenemos mediante el uso de la función *ndiffs* que sugiere en todos los casos un grado de diferenciación 2 $d_{ADF} = 2, d_{KPSS} = 2, d_{PP} = 2$, lo que coincide con los test realizados de forma individual. Dado que no hay discrepancias entre ellos, escogeremos un valor $d = 2$.

Ha llegado el momento de revisar los gráficos ACF y PACF para encontrar los parámetros *p* y *q*. En la *Ilustración 56* vemos ambas gráficas para la segunda diferenciación. En este caso, parece bastante claro que el parámetro *p* (gráfico de autocorrelación parcial) parece que corresponde a 5 rezagos hasta que entra en el intervalo de confianza mientras que *q* (gráfico de autocorrelación) corresponde a 1.

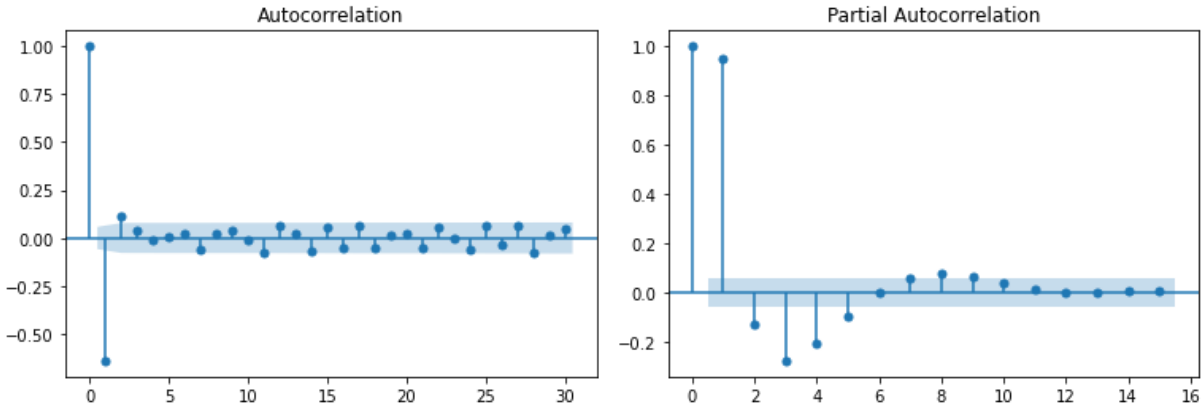


Ilustración 56. Gráfica de autocorrelación y autocorrelación parcial para los metros cúbicos acumulados

Si comparamos a configuración hallada manualmente (5, 2, 1) con la que devuelve *auto_arima* veremos que los resultados son idénticos en este caso.

Igual que en los apartados anteriores, en la *Ilustración 57* podemos observar varias predicciones realizadas por ARIMA. En la mayoría de ellos los pronósticos son prácticamente idénticos a la serie original, si bien es cierto que, como vemos en la gráfica de la esquina inferior derecha hay algún punto para el que ARIMA no ha sido capaz de ajustar correctamente y simplemente muestra una recta con los valores medios de la serie.

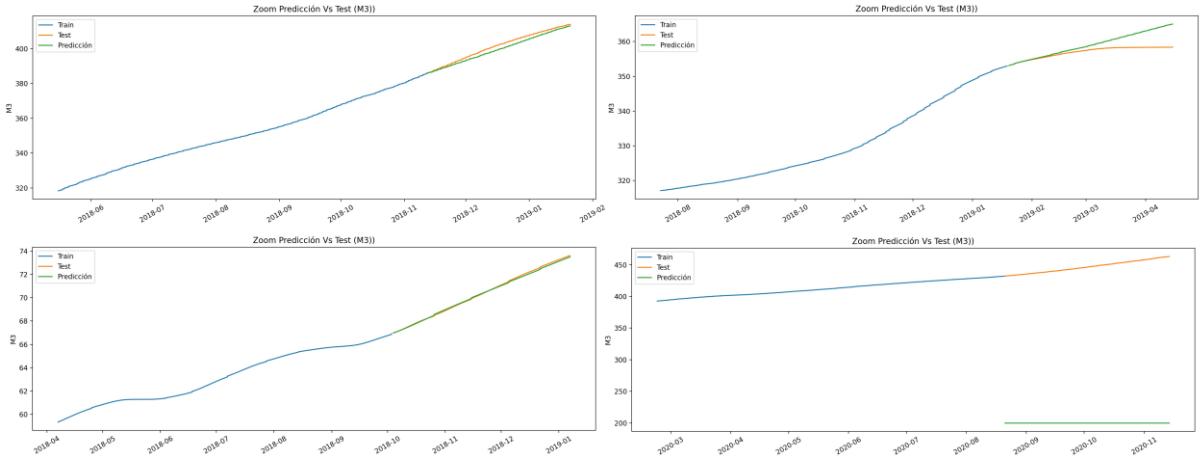


Ilustración 57. Pronóstico de la serie con metros cúbicos acumulados para varios puntos de recogida del TOP150

Nos resta probar el mismo procedimiento por el dataset *TOP150* para calcular los resultados obtenidos con los de los otros predictores. En la *Tabla 11* vemos cómo se comportan los metros cúbicos acumulados a través de sus métricas de rendimiento globales, en este caso se obtienen unos resultados mucho más interesantes que con

los predictores anteriores. Nos encontramos con una exactitud del 97.6% en 1 semana que desciende hasta 96.4% a los 3 meses.

Periodo de tiempo	Métrica	Media	Máximo	Mínimo	Desviación estándar
1 semana	RMSE	3.450	233.157	0.000	21.469
	MAPE	0.024	0.538	0.000	0.107
	MAE	3.444	233.157	0.000	21.470
	ME	-3.403	0.342	-233.157	21.476
	MPE	-0.024	0.004	-0.538	0.107
2 semanas	RMSE	3.532	234.090	0.000	21.564
	MAPE	0.025	0.539	0.000	0.107
	MAE	3.513	234.088	0.000	21.566
	ME	-3.425	0.657	-234.088	21.580
	MPE	-0.024	0.006	-0.539	0.108
1 Mes	RMSE	3.772	236.255	0.000	21.773
	MAPE	0.027	0.542	0.000	0.108
	MAE	3.708	236.242	0.000	21.781
	ME	-3.488	1.160	-236.242	21.817
	MPE	-0.024	0.012	-0.542	0.108
2 Meses	RMSE	4.394	241.241	0.000	22.232
	MAPE	0.031	0.547	0.000	0.109
	MAE	4.210	241.174	0.000	22.245
	ME	-3.631	3.505	-241.174	22.347
	MPE	-0.026	0.023	-0.547	0.110
3 Meses	RMSE	5.177	246.908	0.000	22.743
	MAPE	0.036	0.552	0.000	0.110
	MAE	4.833	246.735	0.000	22.751
	ME	-3.693	7.540	-246.735	22.964
	MPE	-0.026	0.049	-0.552	0.112

Tabla 11. Métricas de rendimiento para los metros cúbicos acumulados

Si miramos además la desviación estándar del error vemos que para una semana se sitúa en el 10.7% y aumenta hasta el 11% con tres meses. Volviendo a la *Ilustración 57* habíamos detectado un punto en el que ARIMA no había sido capaz de obtener una predicción correcta y entrega el valor medio de llenado, causando en ese caso un MAPE extremadamente elevado. Debemos preguntarnos por tanto si existen más puntos como ese que estén aumentando el MAPE general, para comprobarlo probaremos a filtrar aquellos puntos que hayan tenido un MAPE para 1 mes superior al 3%. De todos los puntos de *TOP150* un total de 142 tienen un MAPE medio menor al 3%. Si descartamos los valores de esos 8 puntos y si comprobamos el MAPE medio para el resto podremos comprobar que obtenemos unos valores mucho más cercanos al 100. De hecho, en la *Tabla 12* se presenta un resumen de la media de los MAPE

para los intervalos de tiempo pronosticados de los 142 puntos con un MAPE inferior al 3%.

Periodo de tiempo	MAPE	Precisión (%)
1 Semana	0.000403	99,96
2 Semanas	0.000984	99,90
1 Mes	0.002735	99,73
2 Meses	0.006845	99,32
3 Meses	0.011696	98,83

Tabla 12. Métricas de rendimiento para los metros cúbicos acumulados con MAPE < 3%

En el periodo de un mes de pronóstico, que como hemos comentado anteriormente sería el intervalo deseado para optimizar las rutas de recogida, ARIMA presenta una precisión media del 99,73% por lo que los resultados son realmente buenos y ahora sí que servirían para poder ofrecer al contratista un valor aproximado del llenado de cada uno de sus puntos de recogida con un 0.27% de error.

10.5.6. Auto ARIMA en el dataset representativo

En los apartados anteriores se ha detallado el proceso a seguir para analizar la serie temporal para cada uno de los predictores. Evidentemente no se ha replicado este proceso “manual” por cada uno de los puntos del dataset *TOP150* y mucho menos se podría hacer de este modo en un entorno productivo real donde los pronósticos se deben realizar de forma automática sin intervención del usuario.

Para ello hemos definido un procedimiento automatizado que se encargará de recorrer el conjunto de puntos escogido y realizar las operaciones de preprocesamiento de información, generación de gráficas que serán almacenadas en disco para su posterior consulta y entrenamiento del modelo *ARIMA* con *auto_arima*,

que como se explicó en el punto *7.8* se encargará de buscar automáticamente los mejores valores para *p, d, q*.

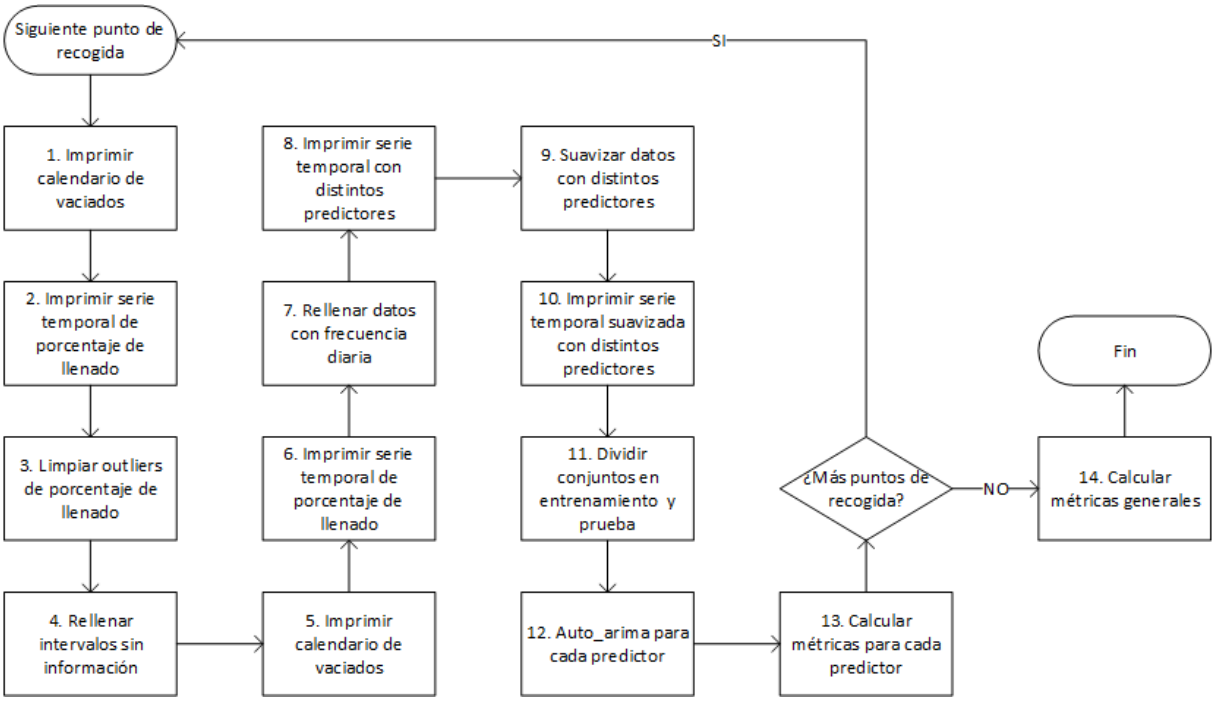


Ilustración 58. Procedimiento de auto_arima por cada punto de recogida

En la *Ilustración 58* se presenta un diagrama de flujo con los principales pasos que se seguirán en cada punto de recogida.

Merece la pena destacar que a la hora de dividir el dataset en los conjuntos de entrenamiento y prueba, se suele realizar tomando muestras aleatorias de los datos, pero en las series temporales no podemos hacerlo así, pues el orden de la muestras importa, por lo que, dado que el dataset contiene información de 4 años (2016 al 2020) tomaremos los últimos 3 meses del punto de recogida para validación y los anteriores para entrenamiento. ¿Por qué 3 meses para validación?, en el punto *10.4* hemos intuido que el punto de recogida alcanzaría el 80% de llenado en un plazo de entre 2 a 3 semanas, por lo que ese será el principal objetivo de nuestro análisis. Ahora bien, nos interesa conocer también que posibilidades o qué fiabilidad tendrían los pronósticos a más largo plazo, algo que podría servir al contratista no solo para planificar la próxima recogida de cada punto, sino las próximas *N* recogidas del próximo trimestre.

10.6. Entrenamiento y pronóstico con red neuronal

Terminado el análisis de la serie con el modelo ARIMA hemos encontrado un predictor que arroja unos resultados sorprendentemente buenos. Otro de los objetivos de nuestro trabajo era comparar los resultados con un modelo sencillo de una red neuronal para comparar los resultados entre ambos algoritmos. Para ello, nos centraremos exclusivamente en el único que ha dado buenos resultados con ARIMA, los metros cúbicos acumulados de residuo.

Crearemos una red neuronal sencilla basada en una arquitectura de perceptrón multicapa con una variable de entrada (las muestras del conjunto de entrenamiento), una capa oculta formada por 500 neuronas (se ha escogido este número para tener una buena cantidad de ellas) y una capa de salida. En el modelo de redes neuronales no será necesario buscar parámetros adicionales como los (p, d, q) de ARIMA ni que la serie sea estacionaria. El modelo de perceptrón multicapa es el modelo más sencillo de red neuronal que debemos probar antes de entrar a evaluar otros tipos de arquitecturas como LSTM, RNN o CNN entre otras.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 500)               12500
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 501
=================================================================
Total params: 13,001
Trainable params: 13,001
Non-trainable params: 0
_________________________________________________________________
None
```

Ilustración 59. Esquema red neuronal

En la *Ilustración 59* podemos ver el resultado de la operación *summary* sobre la red que muestra un resumen de su configuración. La capa oculta utiliza una activación de unidad lineal rectificada (ReLU) y la red se compila utilizando como función de pérdida el MSE con el optimizador *Adam* que se basa en el descenso del gradiente estocástico que se basa en la estimación adaptativa de momentos de primer y segundo orden.

Durante el pronóstico de los valores, se utilizará la técnica de validación progresiva que se explicó en el punto *8.1.2*. El proceso consistirá en iterar por todos los valores del conjunto de test y para cada valor hacer el pronóstico a partir de los datos anteriores. De esta forma el valor real para instante *T* se añadirá al histórico y se utilizará para pronosticar el valor *T+1*. De esta forma la red va alimentándose de los valores a medida que van apareciendo para pronosticar los siguientes.

Igual que se hizo para ARIMA, se diseñará un procedimiento encargado de entrenar un modelo independiente de red neuronal para cada punto de recogida, entrenar y pronosticar los valores para calcular así los valores medios de las métricas para el conjunto de puntos de recogida que estamos estudiando.

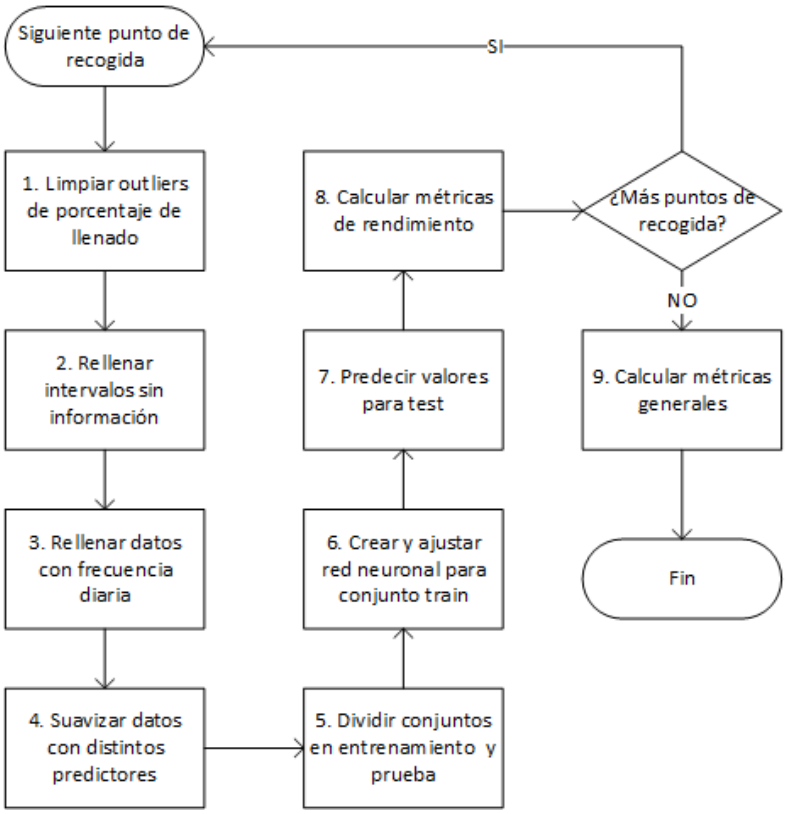


Ilustración 60. Procedimiento de red neuronal para cada punto de recogida

En la *Ilustración 60* se muestra un detalle de los principales puntos de ese procedimiento y en la *Tabla 13* se presentan los valores medios de las métricas de evaluación obtenidas para la red neuronal.

Periodo de tiempo	Métrica	Media	Máximo	Mínimo	Desviación estándar
1 semana	RMSE	0.316	2.369	0.006	0.391
	MAPE	0.003	0.051	0.000	0.006
	MAE	0.314	2.369	0.005	0.392
	ME	0.101	2.369	-1.923	0.493
	MPE	0.001	0.051	-0.030	0.007
2 semanas	RMSE	0.324	2.305	0.015	0.392
	MAPE	0.003	0.052	0.000	0.006
	MAE	0.318	2.303	0.013	0.393
	ME	0.092	2.303	-1.982	0.498
	MPE	0.001	0.052	-0.031	0.007
1 Mes	RMSE	0.335	2.130	0.014	0.381
	MAPE	0.003	0.050	0.000	0.006
	MAE	0.321	2.121	0.013	0.382
	ME	0.074	2.121	-2.035	0.491
	MPE	0.001	0.050	-0.031	0.007
2 Meses	RMSE	0.369	2.401	0.025	0.417
	MAPE	0.004	0.059	0.000	0.007
	MAE	0.345	2.276	0.021	0.406
	ME	0.073	2.276	-2.116	0.519
	MPE	0.001	0.059	-0.032	0.008
3 Meses	RMSE	0.417	3.428	0.021	0.484
	MAPE	0.004	0.069	0.000	0.008
	MAE	0.380	2.996	0.016	0.459
	ME	0.110	2.996	-2.154	0.573
	MPE	0.001	0.069	-0.032	0.009

Tabla 13. Métricas de rendimiento para la red neuronal

Podemos observar cómo se obtiene una precisión del 99,97% para pronósticos a 1 semana y desciende solamente un 0,01% hasta llegar al 99,96% en pronósticos a 3 meses. Además, la desviación estándar del MAPE es también muy constante en todo el intervalo de pronóstico situándose entre el 0,6 y el 0,8% por lo que hemos encontrado un predictor que cumple las expectativas con ambos modelos de aprendizaje.

10.7. Conclusiones de los modelos

Una vez analizados los distintos predictores que se han utilizado en el desarrollo del TFM y de los dos modelos propuestos para pronóstico de serie temporal (ARIMA y red neuronal) podemos hacer un resumen de todos ellos. El *seno* y el *coseno* de la pendiente los descartaremos ya que los valores de MAPE obtenidos distorsionarían el resto de la comparativa debido a la problemática que se ha comentado en sus correspondientes apartados. Además, a medida que se han ido

analizando cada uno de ellos se ha visto que el único que ofrece unos buenos resultados es el de metros cúbicos acumulados, pero incluiremos en la comparativa los resultados también del porcentaje y la pendiente de llenado.

En cuanto a las métricas de rendimiento utilizadas, todas las métricas son útiles a su manera, pero cuando se trata de comparar gráficas entre distintos modelos y predictores lo mejor es utilizar valores porcentuales, pues nos dan un resultado que podemos interpretar con independencia de la escala con la que se han tomado los datos, algo que no ocurre con las métricas no porcentuales como RMSE, MAE o ME. Por ello, aunque en el análisis se han recogido todas las métricas, nos fijaremos principalmente en MAPE para comparar los resultados, pues nos ofrece una visión del porcentaje de error que está cometiendo el modelo sobre los datos, algo que nos servirá para comparar no solo intervalos de tiempo sino predictores en distintas escalas entre sí. A partir del MAPE podremos además calcular la precisión del modelo mediante la *Ecuación 24*.

$$Precisión\ (\%) = 100 - MAPE$$

Ecuación 24. Precisión el modelo a partir del MAPE

Podemos resumir en una tabla la precisión de cada modelo y predictor que nos servirá para comparar gráficamente cómo se comporta cada uno de ellos.

	1 Semana	2 Semanas	1 Mes	2 Meses	3 Meses
Porcentaje de llenado (ARIMA)	86,10	85,50	84,30	81,50	72,30
Pendiente de llenado (ARIMA)	92,10	91,80	91,00	88,50	80,00
Metros cúbicos acumulados (ARIMA)	99,96	99,90	99,73	99,32	98,83
Metros cúbicos acumulados (ANN)	99,97	99,97	99,97	99,96	99,96

Tabla 14. Resumen de precisión para los 3 mejores predictores

Como podemos ver en la *Tabla 14* y también en la *Ilustración 61* los resultados de ARIMA para el porcentaje y la pendiente de llenado no son especialmente buenos, sobre todo a medida que avanzamos en el tiempo hasta los 3 meses (recordemos que la pendiente la habíamos descartado además de por los valores del MAPE porque la serie suavizada no se parecía en nada a la original y la pendiente por los valores del MAPE), sin embargo, para los metros cúbicos acumulados son muy buenos durante todo el periodo de tiempo, con un error que oscila entre el 0,04 y el 1,17%. Podemos observar también que la red neuronal ofrece unas precisiones ligeramente más

próximas al 100% manteniéndose constantes a lo largo del tiempo durante esos 3 primeros meses, con un margen de error que oscila entre el 0,03 y el 0,04%.

Merece la pena destacar que hay que tener cuidado con la interpretación del gráfico en el que aparece el porcentaje de precisión de cada predictor y algoritmo donde el eje Y se encuentra acotado al rango de valores 70 – 100%.

Sin duda alguna, el predictor que se debería utilizar en base a este análisis para utilizar como punto de partida para realizar un proceso de aprendizaje automático que ayude a la empresa a aumentar la productividad y conseguir ese ahorro económico y ambiental general que buscamos son los metros cúbicos acumulados. Dado que como se comentó con anterioridad es de especial interés la precisión en el primer mes después de la última recogida, que es cuando presumiblemente el contenedor deberá ser recogido tanto la red neuronal como ARIMA muestran un comportamiento muy similar. En este intervalo ARIMA tiene una ventaja frente a la ANN que radica básicamente en el coste computacional de la operación, pues ARIMA es mucho menos costoso que ANN en este sentido. De hecho el preprocesamiento de la información, ajuste de parámetros y entrenamiento de cada punto de recogida con el modelo ARIMA está en el orden de 8 segundos por punto, mientras que con la red neuronal nos movemos a los 35 segundos, del orden de 4 veces más tiempo (evidentemente, los timings dependerán de la capacidad computacional del Hardware empleado, pero sirve para hacernos una idea del coste que mencionamos)

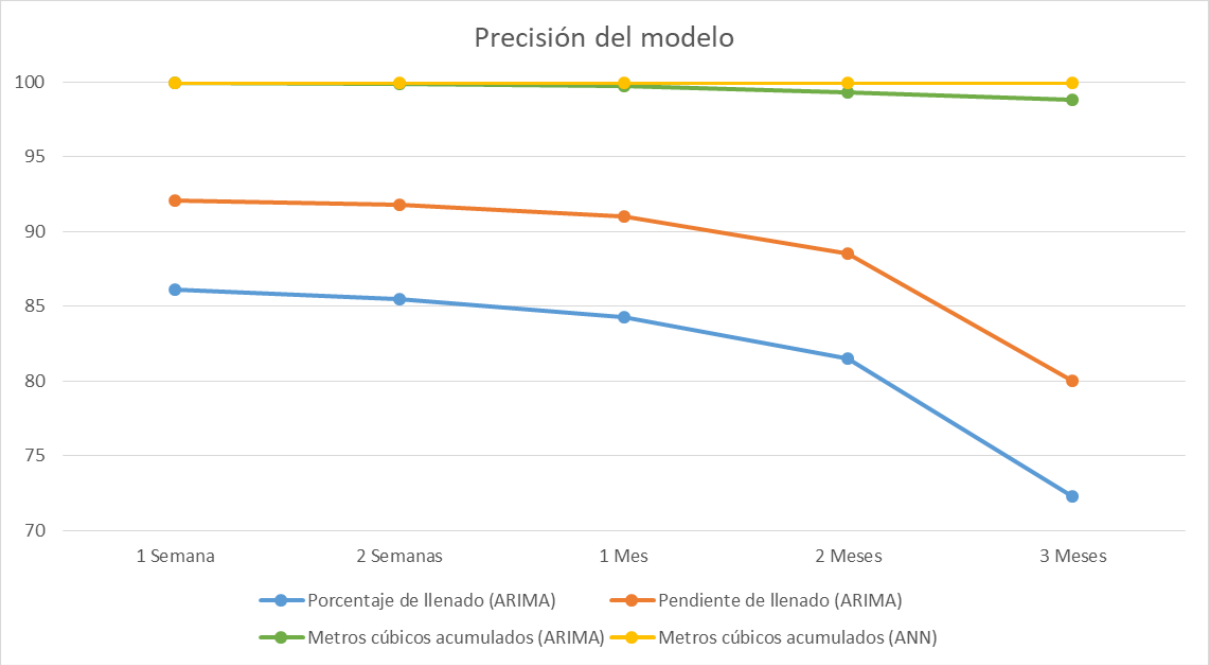


Ilustración 61. Precisión de pronósticos según el predictor y algoritmo

No debemos olvidar igualmente que si el objetivo de la empresa es pronosticar la evolución del llenando en el próximo trimestre o incluso en periodos de tiempo más amplios, podemos valorar el uso de la red neuronal, que mantendrá esos márgenes de error más bajos que ARIMA.

Volviendo a ese primer mes, entre ambos modelos hay una diferencia del margen de error del 0,01%, siendo el error máximo de ARIMA del 0,04% solamente. Si utilizamos información histórica de 3 años para los modelos, suponiendo que tenemos datos semanales de recogida, que el punto de recogida se está recogiendo a un 35% de llenado de media y que la capacidad media del punto son 3m^3 (podemos comprobarlo en el dataset) partiremos, como nos indica la *Ecuación 25* de unos metros acumulados de

$$m^3 acumulado = \left(3 años \frac{52 semanas}{año}\right)\left(3m^3 \frac{35}{100}\right) = 163{,}8m^3$$

Ecuación 25. Metros cúbicos acumulados iniciales para pronóstico

Lo cual implica que con el error obtenido nos estaríamos desviando en $0{,}065m^3$ como podemos comprobar en la *Ecuación 26*.

$$m^3 error = 163{,}8\frac{0{,}04}{100} = 0{,}065 m^3$$

Ecuación 26. Metros cúbicos de error en el pronóstico

Esto implica que en puntos de recogida con una media de 3m^3 de capacidad media tenemos un margen de error de 2,16% como se demuestra en la *Ecuación 27*, si el contratista además quiere poner un margen de error más amplio para evitar desbordes y por lo tanto mala imagen en la ciudad, podría planificar rutas cuando los puntos de recogida alcancen el 80% de su capacidad (los contratistas suelen considerar ese porcentaje de llenado como óptimo para la recogida), de forma que ese 2,16% de error sería prácticamente despreciable.

$$\%\, margen = \frac{0{,}065}{3} 100 = 2{,}16\%$$

Ecuación 27. Porcentaje de error al pronosticar sobre un punto de recogida

Así pues, podríamos concluir que sería posible optimizar el servicio de recogida en base a la previsión de llenado de los puntos de recogida utilizando los metros cúbicos acumulados con ARIMA para intervalos de tiempo pequeños (inferiores a un trimestre) y con ANN para intervalos de tiempo más amplios.

11. Conclusiones y trabajo futuro

A lo largo del trabajo se ha hecho un estudio en profundidad de las series temporales, sus propiedades, cualidades, cómo operar con ellas y de uno de los principales algoritmos para pronóstico de sus valores futuros. Se ha visto que, para una determinada serie temporal es importante escoger el mejor predictor a utilizar, de hecho, en las pruebas realizadas hemos visto que hemos descartado hasta 4 de ellos para finalmente, validar uno solo como eficaz para el objetivo del trabajo.

Nos hemos enfrentado a un problema real de una ciudad de 250K habitantes con 2731 puntos de recogida en la que se han hecho en 4 años 4.357.005 vaciados de contenedores a una media del 35% de llenado. No hace falta profundizar demasiado para ver que existe un amplio margen de mejora, de hecho, sin aplicar ninguna técnica de pronóstico el contratista podría duplicar los días entre recogidas sin miedo a que los contenedores se desbordasen, lo que implicaría una reducción de la mitad de los servicios necesarios para recoger los contenedores. Si además somos

capaces de ofrecer al contratista el nivel de llenado de los puntos de recogida con una precisión cercana al 100% (objetivo que hemos logrado en el TFM) podría optimizar notablemente sus rutas de recogida para vaciar los puntos de recogida en su punto óptimo, evitando así desplazamientos innecesarios y por consiguiente, obteniendo esos ahorros tan importantes que se han nombrado a lo largo del trabajo.

Hemos pronosticado además la generación de residuos desde un punto de vista mucho más granular que en los trabajos previos vistos en punto *5*, donde se pronosticaba la generación a un nivel de agrupación mucho mayor (áreas, distritos, barrios, etc.) y que, aunque resulta útil para planificar el servicio a gran nivel no sirven para determinar el momento ideal para vaciar un punto de recogida por lo que este estudio puede servir como punto de partida para otros trabajos de optimización de los servicios.

Hemos obtenido también unas métricas de precisión muy elevadas tanto para ARIMA como para una red neuronal sencilla, siendo en el último modelo más constantes y cercanas al 100% a medida que avanza el tiempo de pronóstico pero por unos porcentajes que podríamos considerar despreciables, por lo que, se debe valorar el intervalo de tiempo que se desea utilizar para pronóstico para estudiar si compensa el incrementar el coste computacional de la red neuronal a fin de pronosticar a más largo plazo. Podríamos concluir por tanto que hemos logrado el objetivo de este trabajo satisfactoriamente.

Debido a que la asignatura tiene evidentemente una temporalización hay un momento en que se debe detener el trabajo, aunque todavía haya aspectos susceptibles de mejora, por lo que este trabajo da pie a continuar en trabajos futuros en varias líneas de investigación.

En este TFM nos hemos centrado en entrenar cada punto de recogida como un ente independiente con su propio histórico de información sobre el que generar un modelo y a partir de él un pronóstico de la serie temporal. Ahora bien, hay una pregunta que ha quedado en el aire, ¿los puntos de recogida cercanos tienen el mismo comportamiento? o lo que es lo mismo ¿en un barrio/vecindario hay *n* puntos de recogida cuyas series temporales sean idénticas o tengan el mismo comportamiento? Si fuésemos capaces de obtener esta información podríamos evitar tener que entrenar

el algoritmo para cada punto de recogida de forma independiente, teniendo clúster de puntos para los que un mismo modelo encajaría perfectamente y por lo tanto ahorrando tiempo de cómputo.

En este trabajo nos hemos centrado en el estudio de series temporales utilizando ARIMA y haciendo una pequeña incursión en las redes neuronales para comprobar la mejora en la precisión de los algoritmos, por lo que un trabajo futuro podría ser volver a probar todos los predictores con distintos modelos de redes neuronales (LSTM, RNN, CNN, etc.) a fin de conseguir un modelo más sencillo y que necesite menos preprocesado de los datos que ARIMA.

Se ha centrado el estudio sobre los contenedores de la fracción vidrio por ser la que presenta una densidad de residuo más estable, otra posible línea de trabajo sería el estudio de las métricas de rendimiento en cada una de las fracciones del dataset.

La empresa proveedora de datos cuenta también con uno sensores capaces de medir el porcentaje de residuo en un contenedor a intervalos regulares de tiempo. La medida de estos sensores será más precisa que el porcentaje de llenado que hemos calculado a partir del peso, ya que estamos utilizando para ello densidades medidas de residuo por lo que sería interesante contar con esa información para probar los algoritmos en trabajos futuros. ¿Por qué no lo hemos probado en este? Porque la cantidad de sensores distribuidos por las ciudades se limitan a unas pocas unidades en cada ayuntamiento para pruebas de concepto, por lo que la información que se puede obtener a día de hoy no es relevante pero se prevé que en el futuro este tipo de dispositivos se instale masivamente en las ciudades con el concepto de las Smart Cities.

Todo el estudio se ha realizado utilizando series temporales univariantes, pero, se podrían intentar incluir variables exógenas (temperaturas, lluvias, festividades, etc.) para convertir la serie temporal en multivariante y comprobar si, de esta forma, ya sea con ARIMA o redes neuronales se puede mejorar la precisión de los pronósticos.

Por último, otro trabajo futuro podría ser utilizar este estudio para diseñar mecanismos de generación automática de rutas de recogida en base a los pronósticos

ofrecidos por los modelos de aprendizaje automático implementado que ayudaran al contratista a realizar su trabajo de forma más ágil y de una forma más óptima.

12. Bibliografía

[1] Abbasi, M., Abduli, M.A., Omidvar, B., Baghvand, A., 2013. *Forecasting municipal solid waste generation by hybrid support vector machine and partial least square model*. Int. J. Environ. Res. 7, 27–38. https://www.researchgate.net/publication/262298518_Forecasting_Municipal_Solid_waste_Generation_by_Hybrid_Support_Vector_Machine_and_Partial_Least_Square_Model

[2] Abbasi, M (2016). Forecasting municipal solid waste generation using artificial intelligence modelling approaches. *Waste Management*, 56, 13-22. https://doi.org/10.1016/j.wasman.2016.05.018

[3] Abdoli, M.A., Falah Nezhad, M., Salehi Sede, R., Behboudian, S., 2012. *Longterm forecasting of solid waste generation by the artificial neural networks*. Environ. Prog Sust. Energy 31, 628–636. https://www.researchgate.net/publication/227720897_Longterm_forecasting_of_solid_waste_generation_by_the_artificial_neural_networks

[4] *Anaconda | Individual Edition*. (s. f.). Anaconda. Recuperado 3 de mayo de 2021, de https://www.anaconda.com/products/individual

[5] *API Reference—Pmdarima 1.8.2 documentation*. (s. f.). Recuperado 2 de mayo de 2021, de http://alkaline-ml.com/pmdarima/modules/classes.html

[6] *API Reference—Statsmodels*. (s. f.). Recuperado 2 de mayo de 2021, de https://www.statsmodels.org/stable/api.html

[7] *ARIMA Model - Complete Guide to Time Series Forecasting in Python*. (2019, febrero 18). ML+. https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/

[8] Ayeleru, O. O., Fajimi, L. I., Oboirien, B. O., & Olubambi, P. A. (2021). *Forecasting municipal solid waste quantity using artificial neural network and supported vector machine techniques: A case study of Johannesburg, South Africa. Journal of Cleaner Production*, 289, 125671. https://doi.org/10.1016/j.jclepro.2020.125671

[9] Beigl, P., Lebersorger, S., Salhofer, S., 2008. *Modelling municipal solid waste generation: a review*. Waste Manage. 28, 200–214. https://doi.org/10.1016/j.wasman.2006.12.011

[10] Beliën, J., De Boeck, L., & Van Ackere, J. (2014). Municipal Solid Waste Collection and Management Problems: A Literature Review. *Transportation Science*, *48*(1), 78-102. https://doi.org/10.1287/trsc.1120.0448

[11] Box GEP, Jenkins GM. *Time Series Analysis: Forecasting and Control.* San Francisco: Holden-Day, 1976.

[12] Brockwell PJ, Davis RA. *Introduction to Time Series and Forecasting*. New York: Springer-Verlag, 1996

[13] Cuadra Troncoso, J.M. *Tipologías de las redes neuronales profundas*. UNED, 2021

[14] *Deep Learning Models for Univariate Time Series Forecasting*. (2018, octubre 29). https://machinelearningmastery.com/how-to-develop-deep-learning-models-for-univariate-time-series-forecasting/

[15] Directiva (UE) 2018/851, del Parlamento Europeo y del Consejo por la que se modifica la Directiva 2008/98/CE sobre los residuos. Diario Oficial de la Unión Europea. 30 de Mayo de 2018. L150, 109-140

[16] Ecoembes. (2015). *La gestión de residuos municipales (2ª edición)*. Madrid: Editorial MIC

[17] Eurostat. (2021). *Recycling rate of municipal waste*. https://ec.europa.eu/eurostat/databrowser/view/t2020_rt120/default/table?lang=en

[18] Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow (2nd edition)*. USA: O'Reilly

[19] Hockett, D., Lober, D.J., Pilgrim, K., 1995. *Determinants of per capita municipal solid waste generation in the southeastern United States*. J. Environ. Manage. 45, 205–217. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.405.1884&rep=rep1&type=pdf

[20] James, G. (2017). *An Introduction to Statistical Learning*. London: Springer

[21] Kannangara, M. (2018, Abril). Modeling and prediction of regional municipal solid waste generation and diversion in Canada using machine

learning approaches. *Waste Management,* 74, 3-15. https://doi.org/10.1016/j.wasman.2017.11.057

[22] Kolekar, K.A. (2016). A Review on Prediction of Municipal Solid Waste Generation Models. *Procedia Environmental Sciences,* 35, 238-244. https://doi.org/10.1016/j.proenv.2016.07.087

[23] Li, Z. (2018, Junio). Silicon enhancement of estimated plant biomass carbon accumulation under abiotic and biotic stresses. *Agronomy for Sustainable Development,* 38(26). https://doi.org/10.1007/s13593-018-0496-4

[24] Mauricio, José Alberto (2007). *Introducción al Análisis de series temporales*. Universidad Complutense de Madrid.

[25] Navarro, J (2002, Mayo). Time series analysis and forecasting techniques for municipal solid waste management. *Agronomy for Sustainable Development,* 38(3). https://doi.org/10.1016/S0921-3449(02)00002-2

[26] Parra, F (2019). *Estadística y Machine Learning con R*. (s. f.). Recuperado 19 de febrero de 2021, de https://bookdown.org/content/2274/portada.html

[27] Peña, D., Tiao, G. (2001). *A course in Time Serie Analysis.* New York: John Wiley & sons, inc.

[28] *Scikit-learn: Machine learning in Python – Scikit-learn 0.24.1 documentation*. (s.f.). Recuperado 19 de febrero de 2021, de https://scikit-learn.org/stable

[29] Shankar, D. (2016, Diciembre). Artificial neural network based modelling approach for municipal solid waste gasification in a fluidized bed reactor. *Waste Management,* 58, 202-213. https://doi.org/10.1016/j.wasman.2016.08.023

[30] *Time Series Analysis in Python - A Comprehensive Guide with Examples*. (2019, febrero 13). ML+. https://www.machinelearningplus.com/time-series/time-series-analysis-python/

[31] VanderPlas, J (2017). *Python Data Science Handbook*. USA: O'Reilly

[32] Wei-Meng, L. (2019). *Python Machine Learning*. Indianapolis: John Wiley & Sons, Inc.

[33] Yang, Z., Chen, H., Du, L., Lu, W., & Qi, K. (2021). *Exploring the industrial solid wastes management system: Empirical analysis of forecasting and safeguard mechanisms. Journal of Environmental Management*, 279, 111627. https://doi.org/10.1016/j.jenvman.2020.111627

13. Anexo I (auto_arima)

En el presente anexo se muestra la ejecución y resultados del método *auto_arima* del paquete pmdarima.

```
from statsmodels.tsa.arima_model import ARIMA
import pmdarima as pm

model = pm.auto_arima(
   daily_data.M3,
   start_p = 1,
   start_q = 1,
   test = 'adf',
   max_p = 3,
   max_q = 3,
   m = 1,
   d = None,
   seasonal = False,
   start_P = 0,
   D = 0,
   trace = True,
   error_action = 'ignore',
   suppress_warnings = True,
   stepwise = True)

print(model.summary())
```

```
Performing stepwise search to minimize aic
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=562.520, Time=0.15 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=559.087, Time=0.04 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=561.086, Time=0.04 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=561.086, Time=0.04 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=813.508, Time=0.02 sec

Best model:  ARIMA(0,1,0)(0,0,0)[0] intercept
Total fit time: 0.295 seconds
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  330
Model:               SARIMAX(0, 1, 0)   Log Likelihood                -277.543
Date:                Sat, 01 May 2021   AIC                            559.087
Time:                        23:07:42   BIC                            566.679
Sample:                             0   HQIC                           562.115
                                  - 330
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept      0.6111      0.048     12.818      0.000       0.518       0.705
sigma2         0.3164      0.027     11.636      0.000       0.263       0.370
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):               172.01
Prob(Q):                              0.99   Prob(JB):                         0.00
Heteroskedasticity (H):               1.35   Skew:                             1.50
Prob(H) (two-sided):                  0.12   Kurtosis:                         4.89
===================================================================================
```

En el ejemplo de código superior, hemos definido un modelo *auto_arima* al que le hemos pasado una serie de parámetros entre los que destacan

- *test*: Indicamos que debe utilizar el test *adf* para encontrar el valor óptimo de *d*
- *max_p*: Valor máximo que puede tomar el término *p*
- *max_q*: Valor máximo que puede tomar el término *q*
- *m*: Frecuencia de la serie, por defecto es igual a 1, lo que indica una frecuencia anual, una serie no estacional.
- *d*: Servirá para indicar el valor del término *d*, si lo dejamos en *None* el algoritmo lo buscará por nosotros.
- *seassonal*: Servirá para indicar si estamos trabajando con una serie estacional o no.

En el resultado de la función vemos que ha ido probando distintas combinaciones de valores de p, d, q e imprimiendo, entre otras cosas, el valor AIC para cada combinación. Al final, nos indica el mejor modelo (best model) que corresponderá con el que tenga un menor valor AIC.

Merece la pena, destacar entre los resultados de la salida del algoritmo las siguientes métricas o estadísticos:

- AIC (Akaike Information Criterion): Es un estimador del error de predicción fuera de la muestra (out of sample) y por lo tanto, de la calidad de los modelos estadísticos para un dataset. El resultado deseado para la selección de modelos consiste en encontrar el valor más bajo posible y es especialmente valioso en análisis de series temporales, donde generalmente los valores más valiosos son los más recientes.
- BIC (Bayesian Information Criterion): Igual que AIC es un criterio para la selección de modelos y de hecho ambos están estrechamente relacionados. En basa en la función de verosimilitud y se prefiere un valor más bajo para seleccionar el modelo.
- HQIC (Hannan-Quinn Information Criterion): Otro estadístico utilizado para la selección de modelos que se utiliza como alternativa al AIC y al BIC. Igual que en los anteriores se prefiere un valor más bajo para escoger el mejor modelo.

14. Anexo II (Funciones trigonométricas)

Merece la pena destacar las principales funciones trigonométricas que se utilizarán en este trabajo.

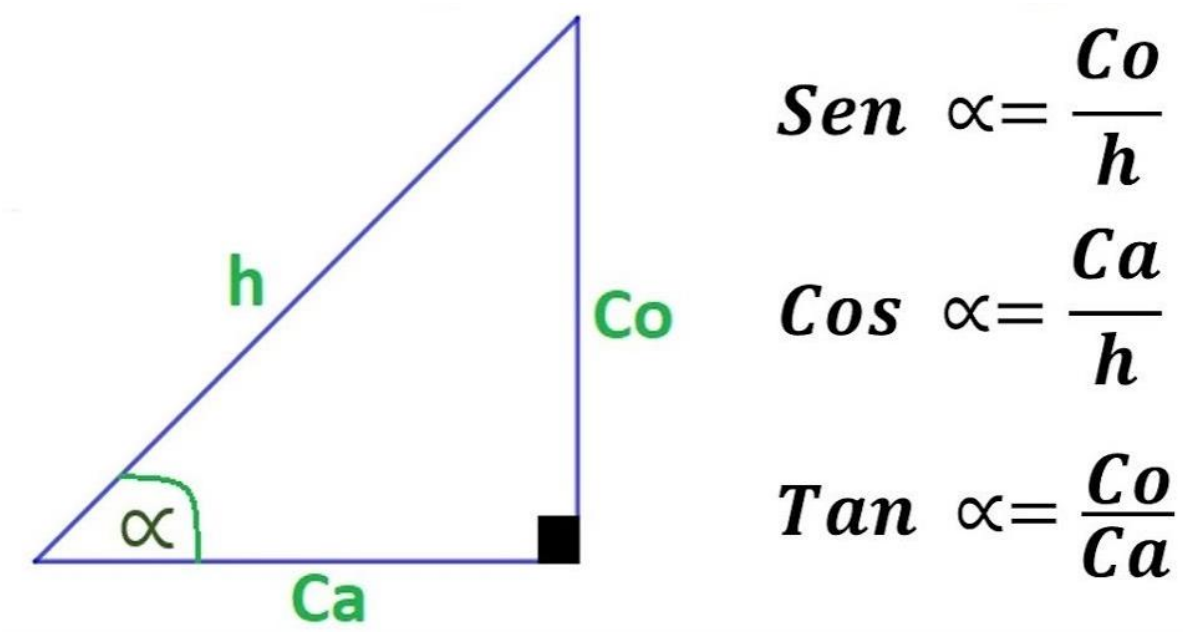


Ilustración 62. Razones trigonométricas triángulo rectángulo

15. Anexo III (Código fuente)

Tanto los archivos originales con los datos suministrados por parte de la empresa como los archivos con el código fuente en formato Jupyter Notebook se encuentran disponibles junto con la documentación entregada a la universidad para reproducir todo el proceso descrito a lo largo de este trabajo. En la *Tabla 15* podemos ver una relación de los archivos disponibles y su misión.

Nombre de archive	**Descripción**
containers_movement.csv	Archivo con la lista de movimientos de los contenedores por los puntos de recogida
weights.csv	Archivo con la lista de descargas de residuo efectuadas en los contenedores
1. CollectionPointCapacity.ipynb	Archivo con el código fuente necesario para calcular la capacidad total de los puntos de recogida en el tiempo a partir del archivo containers_movement.csv
2. CollectionPointFillPercent.ipynb	Archivo con el código fuente necesario para calcular el porcentaje de llenado de los puntos de recogida en el momento de sus vaciados a partir del archivo containers_movement.csv y weights.csv
3. Time Serie analisys.ipynb	Archivo con el código fuente con las pruebas realizadas para la descripción de las principales propiedades de las series temporales
4. WasteGenerationForecast.ipynb	Archivo con el código fuente necesario para realizar los pronósticos de progresión de llenado de los puntos de recogida así como obtención de las métricas de rendimiento

Tabla 15. Relación de archivos