Motivación

El Metro de Madrid fué inagurado en 1919 por el rey Alfonso XIII, aquella primera "red" de Metro constaba únicamente con ocho paradas, desde la Puerta del Sol hasta Cuatro Caminos. Tuvo tal éxito el nuevo medio de transporte en la ciudad que fué usado por más de 14 millones de usuarios.

Actualmente el Metro de Madrid, es la segunda red de metro mas extensa de la Unión Europea y la cuerta del mundo, consta de 13 líneas con 278 paradas distribuidas por toda la ciudad, creando una gran red de transporte de casi 290km, estableciendo la red de transporte más eficiente de la capital.

Línea	Terminales	Longitud	Estaciones
0	Pinar de Chamartín - Valdecarros	20,8 km	31
1	Las Rosas - Cuatro Caminos	14 km	20
2	Villaverde Alto - Moncloa	16,4 km	18
3	Argüelles - Pinar de Chamartín	16 km	23
4	Alameda de Osuna - Casa de Campo	23,2 km	32
5	Circular	23,5 km	28
6	Hospital de Henares - Pitis	31,2 km	29
7	Nuevos Ministerios - Aeropuerto T4	16,5 km	8
8	Paco de Lucía - Arganda del Rey	38,0 km	26
9	Hospital Infanta Sofía - Puerta del Sur	39,9 km	31
10	Plaza Elíptica - La Fortuna	5,3 km	7
11	MetroSur (Circular)	40,7 km	28
12	Ópera - Príncipe Pío	1,1 km	2
13	-	285,1 km	278

```
Tabla extraída de wikipedia
```

En la Comunidad de Madrid (CAM) el transporte público preferido por los Madrileños es el Metro y en los tiempos que nos encontramos (de pandemia), resultaría interesante estimar el volumen de pasajeros que recibirá el Metro en distintos instantes del tiempo.

Apoyados en la temperatura, viento, presión atmosférica y cantidad de rayoUV, vamos a intentar predecir la cantidad de viajeros en el metro de manera mensual (debido a que no he encontrado datos diarios, o incluso por horas).

```
Tenemos 25 columnas y 396 filas
```

fecha	volumenMetro	tmed	prec	racha

	volumenMetro	tmed	prec	racha
fecha				
2000-02-01	92.705	10.748276	5.448276	951.093103
2000-03-01	102.479	12.080645	6.777419	944.209677
2000-04-01	83.902	10.683333	6.273333	935.190000
2000-05-01	94.966	17.993750	12.743750	939.990323
2000-06-01	93.300	24.047917	16.800000	943.486667

Hemos recogido los datos desde Enero de 2000 hasta Diciembre de 2019 de volumen de pasajeros, datos climatológicos como temperatura, viento y presión medias y sus desviaciones típicas menusuales. Los datos han sido extraidos del banco de datos del ayuntamiento de Madrid y de la AEMET.

Hemos tenido que realizar una imputación de algunos de los datos, pues había datos nulos de presión atmosférica. El método utilizado, al tratarse de una serie temporal, ha sido la interpolación que ofrece la librería pandas.

Capítulo 1: Análisis explotatorio de los datos (EDA)

Lo primero que debemos hacer es un pequño análisis exploratorio de las variables input y de la variable objetivo. Decimos que una variable objetivo, cuando es la elegida para estimar y las variables input, son las que se basará nuestro modelo para realizar las predicciones. En nuestro conjunto de datos, tenemos diferenciadas tres tipos de variables.

Variable de tiempo(fecha), hasta el año 2020 ya que debido a la pandemia, los datos se han visto alterados de manera muy fuerte y tener un frecuencia mensual, no tendríamos datos suficientes para hacer una buena estimación, es por esto, que se ha optado recurir a datos hasta el 2020, también a tener en cuenta esta frecuencia mensual, esto se debe a que los datos que disponemos sobre el volumen de pasajeros en el metro de Madrid es mensual. En una puesta en producción los datos podrían ser en streaming, con una actualización de minutos, probablemente gestionado en un entorno spark o con un procesado en bach. Hemos propuesto una pequeña base de datos SQL con tres tablas, que contienen el maestro de fechas, donde determina si es laborable, festivo, fin de semana... Otra tabla con los datos meteorológicos, agrupados por la variable tiempo, aunque en la ciudad de Madrid, por ejemplo existen 3 estaciones disponibles de donde sacar datos, estos datos los facilita la AEMET con su API y contiene un apartado para desarrolo de aplicaciones en streaming. Y por su puesto el volumen de pasajeros, que mediante los tornos ya instalados en las paradas de Metro se podría llevar el conteo casi en tiempo real. Por lo que, aunque la aplicación que propongo está gestionada en mensual, se podría extrapolar a un entorno en streaming o incluso para análisis en bach.

El modelo propuesto es un modelo ARIMA, ya que se ha comprobado en diversos estudios que para variables como la que vamos a estudiar tiene un gran rendimiento, además de su fácil interpretación. Además, al encontrarnos ante una serie temporal con carácter estacionario va el método clásico ARIMA tendrá un gran rendimiento.

Volumen de pasajeros

Estos datos han sido extraidos de Banco de datos del ayuntamiento de Madrid el dato viene informado miles de viajeros que están registrados en la agencia de viajeros de la CAM.

	count	mean	std	min	25%	50%	75%	max
volumenMetro	239.0	88.589019	12.173598	48.479	84.0535	91.535	96.3655	109.412

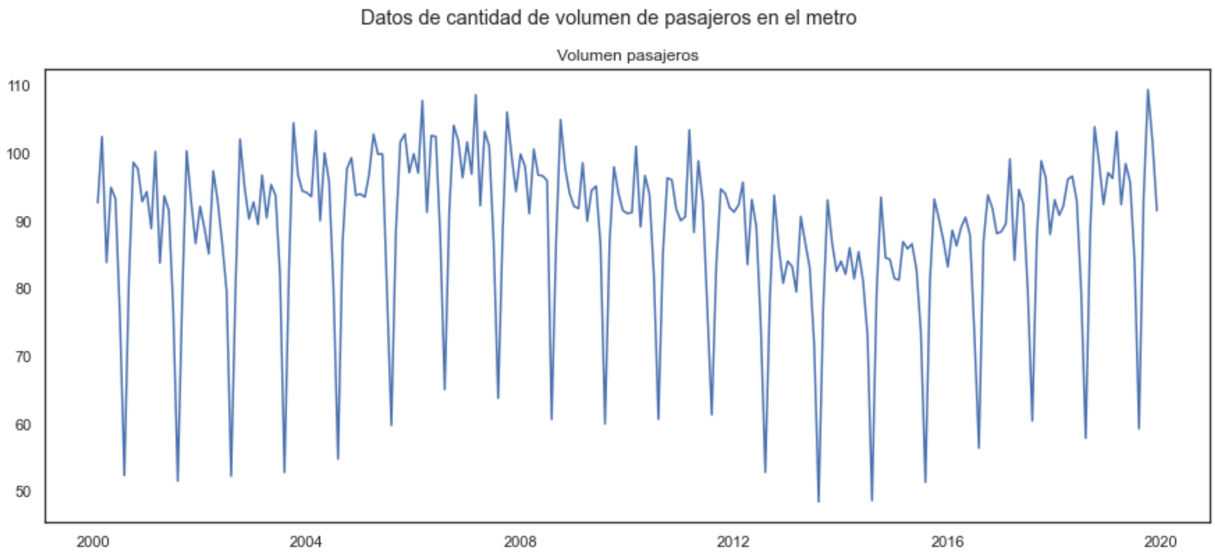
En el gráfico vemos como nos encontramos ante una serie casi en su totalidad estacionaria, hemos tenido que lidiar con una mala extracción de los datos, pues en el año 2010 aparecía una gran caida de pasajeros. La decisión ha sido utilizar la media del año anterior y posterior, es decir, la media entre los datos de 2019 y 2021.

Definición: Decimos que un proceso estocástico es estarionario en el sentido estricto cuando las distribuciones marginales de cualquier conjunto de k variables son idénticas, en distribución y en parámetros.

Para nuestro estudio, nos vasta que el proceso sea estacionario en sentído débil, es decir

$$\begin{cases} \mu_t = \mu \ \ \forall t \\ \sigma_t^2 = \sigma^2 \ \ \forall t \\ Cov(X_t, X_{t+k}) = E[(x_t - \mu)(x_{t-k} - \mu)] = \gamma_k \ \ \forall k \end{cases}$$

Esto quiere decir, que tanto media como varianza permanecen constantes con el tiempo y la covariancia, entre dos variables de la serie depende sólo de su separación en el tiempo.



Vemos como la serie presenta una cierta periodicidad de los datos, pues la forma que toma la serie es similar en todo el tiempo. Presentando cierta tendencia sinusoidal, si queremos aplicar el modelo ARIMA, debemos "eliminar" esta componente estacional, mediante por ejemplo normalizando podríamos obtener un ruido blanco.

Decimos que un proceso estocástico es un ruido blanco si

$$E[X(t)] = 0 \qquad V[X(t)] = \sigma^2 \qquad \gamma_k = 0$$

El objetivo es ajustar el modelo ARIMA es que el error producido sea un ruido blanco, esto significará que nuestro modelo está bien ajustado.

Actualmente, vemos con un gráfico de cajas y bigotes y un histográma, que la distribución de nuestra serie no es simétrica y esto se debe a que aunque tenemos una periodicidad de los datos muy clara, existe tendencia, provocando que la distribución de la serie no sea simétrica, con una media distante a la mediana y una desviación tipìca elevada. Por lo que nuestra serie no es estacionaria.

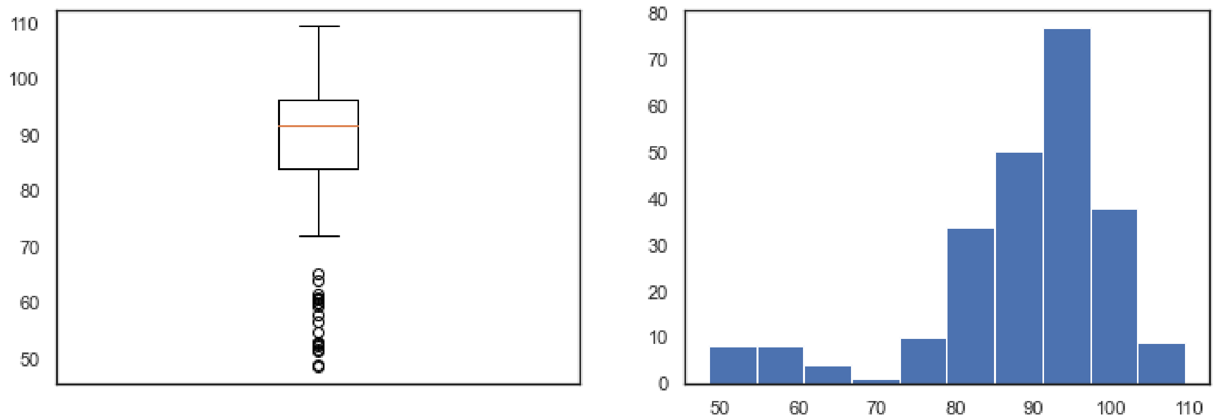
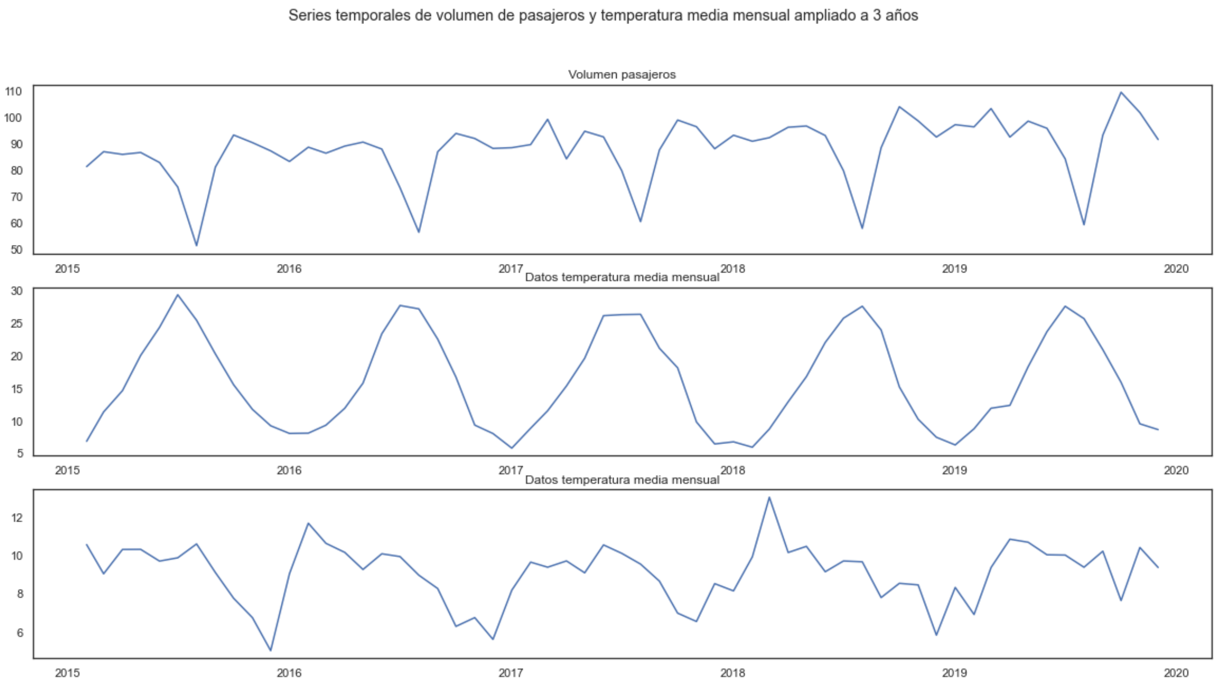


Fig 1.: Gráfico boxplot e histograma del volumen de Pasajeros del metro

Variables climatológicas

Vamos a exponer la comparación de un par de series de la variable objetivo con alguna exogena de apoyo, aunque el estudio se ha realizado en todas las variables. Vamos a aplicar un filtro de fecha de 5 años para poder comprar mejor.



Vemos como claramente en las épocas de mas calor, hay menos usuarios de Metro que en los meses más fríos. Parece que la temperatura media puede ayudarnos a estimar. Si nos fijamos en el viento sin embargo, cuando se producen disminuciones en la valocidad del viento, vemos como en general, se producen los aumentos de volumen de pasajeros. Esto es debido a que parece que va con un ligero retraso en su periodicidad, ya que en julio/agosto se produce un aumento de la temperatura, con una disminución del volumen de pasajeros, pero el viento se mantiene constante para estos meses. Es a partir de septiembre cuando se produce esta disminución de la velocidad media del viento y una disminución de la temperatura y el aumento del volumen de pasajeros.

Capítulo 2: Modelización predictiva

Modelo ARIMA

Destacamos los modelos clásicos:

- Decimos que un modelo es AR(p) (Autoregresivo de orden p), cuando las autocorrelaciones simples decrecen de manera exponencial y existen p autocorrelaciones distintas de 0.
- Decimos que un modelo es MA(q) (Medias móviles), cuando las autocorrelaciones simples decaen y se cortan de forma rápida, sin embargo las autocorrelaciones parciales decrecen exponencialmente.
- Decimos que un modelo es ARMA(p,q), cuando comparten las características de ambos modelos.

Definimos modelo ARIMA (Autoregresivo integrado de medias móviles), como el modelo estadístico que utiliza variaciones y regresiones de datos estadísticos con el fin de encontrar patrones para una predicción hacia el futuro. Como ya hemos explicado, las series estacionarias son las que tienen media 0, por lo tanto, un proceso no estacionario lo llamaremos proceso integrado si al hacer una diferenciación se obtienen procesos estacionarios.

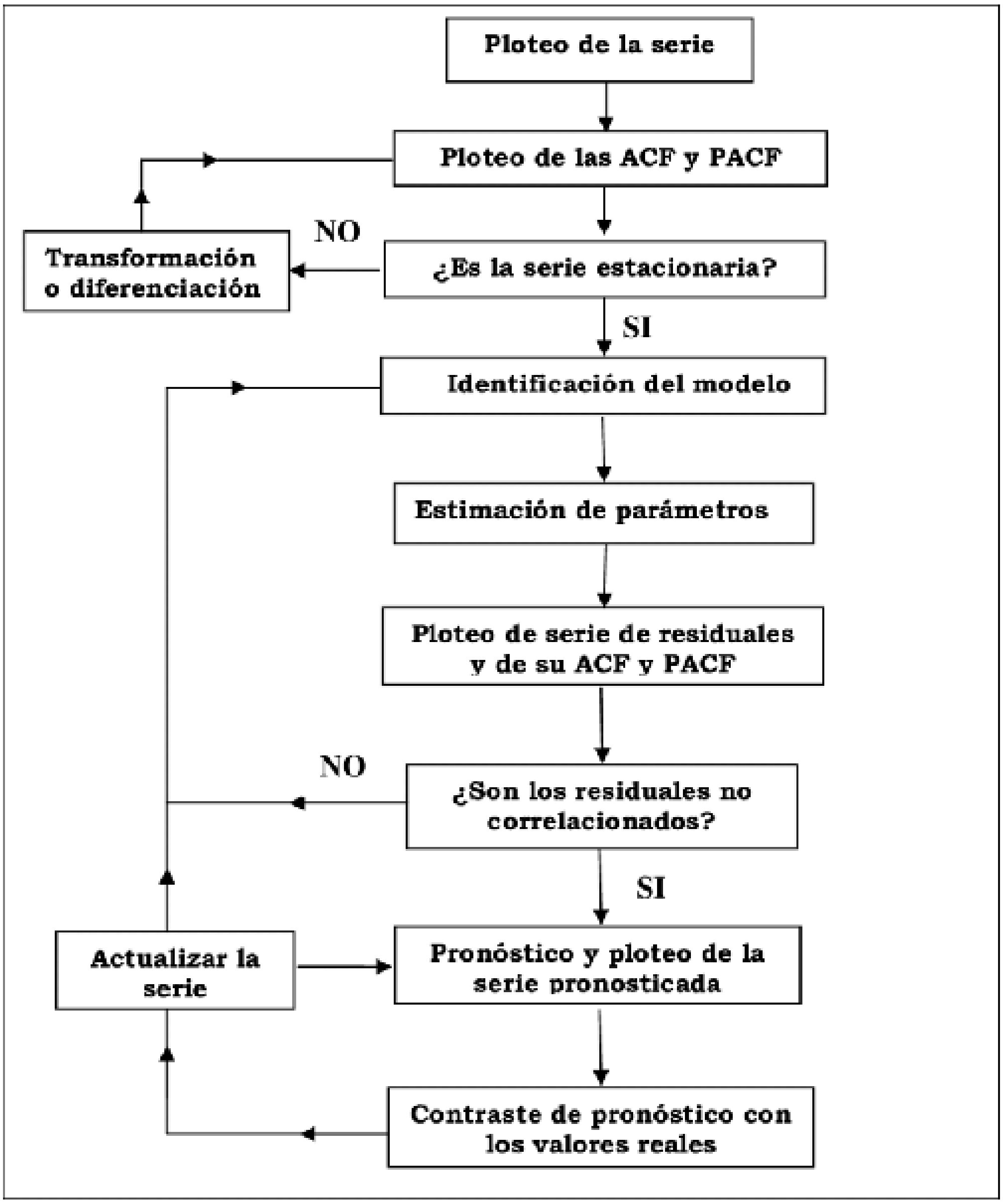
Decimos que aplicamos una diferenciación de orden k a una serie, cuando teniendo X_t le restamos la observación de k instantes anteriores, es decir, X_{t-k}.

Vamos a hacer una diferenciación de la serie para ver si conseguimos eliminar la media igual a cero y así poder aplicar un modelo $ARIMA(p,d,q)(P,D,Q)_s$.

Vamos a utilizar la metodología Box-Jenkis para ajustar el modelo lo más fielmente posible, para realizar unas buenas predicciones.

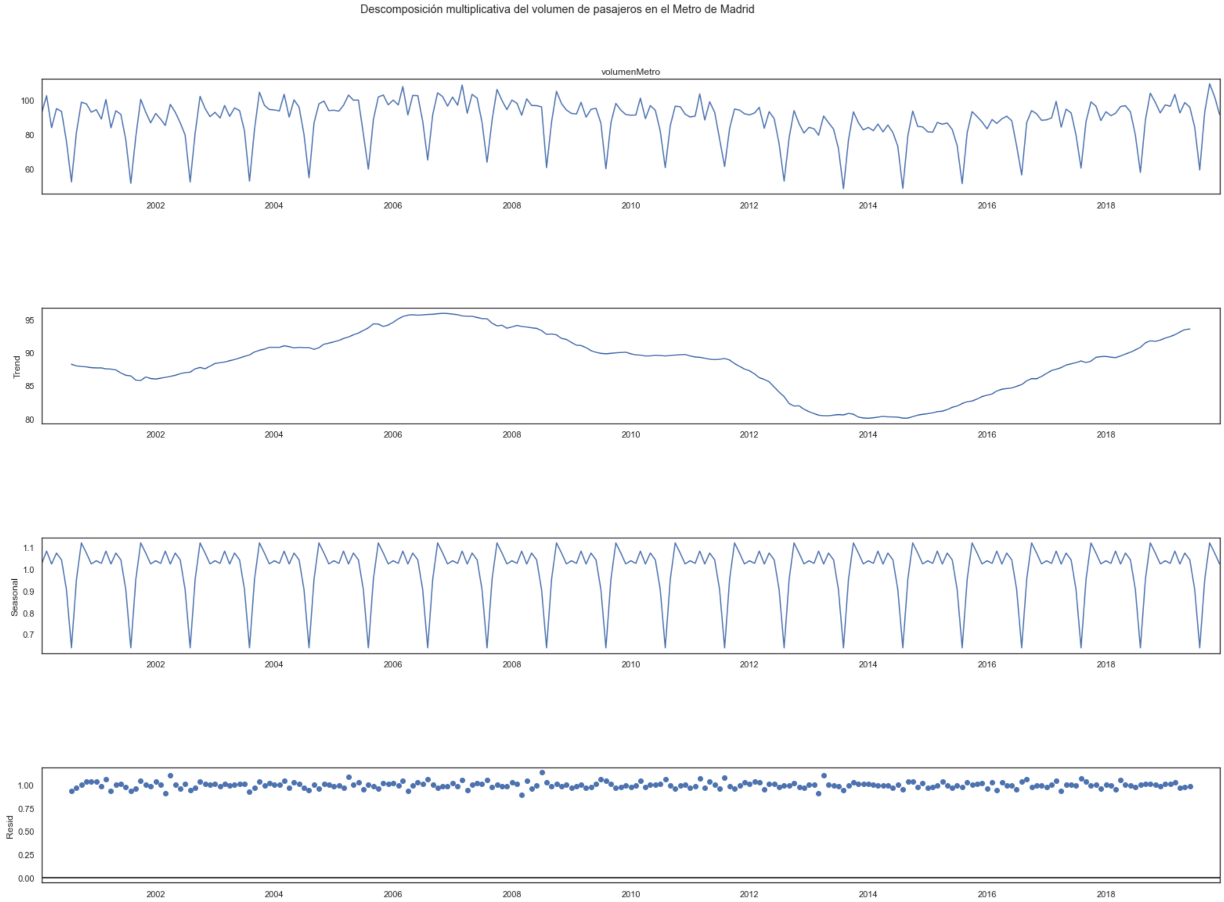
```
Esquema metodología Box-Jenkins
```

Out[12]:



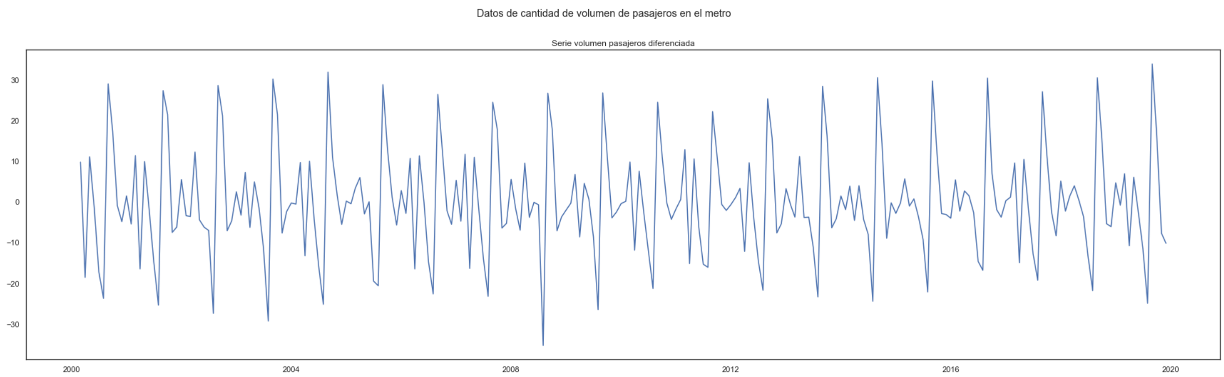
Serie y descomposición estacional

Vamos a graficar la serie con su descomposición estacional, así podemos ver bien la tendencia, los coeficientes de estacionalidad y el error.

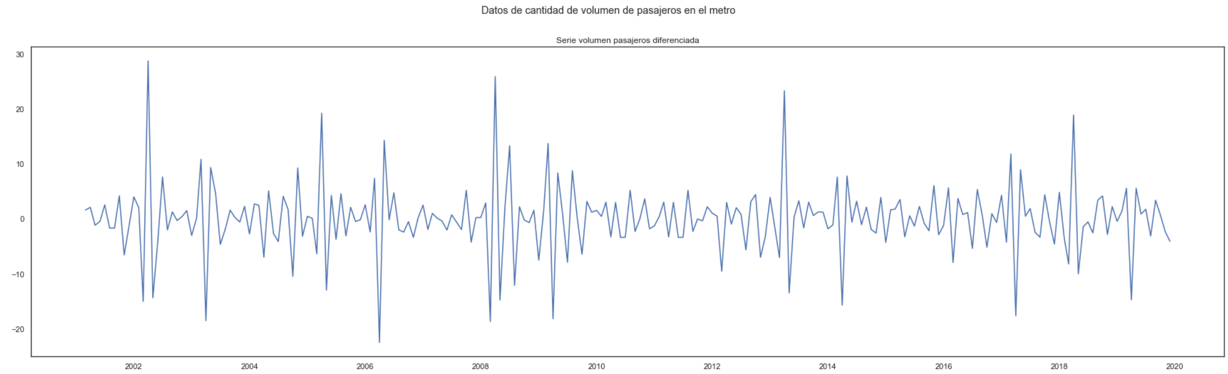


La serie ya hemos analizado que tiene una periodicidad de 12 meses, esta descomposición nos afirma que efectivamente esto sucede. Además como habíamos percibido la tendencia tiene un comportamiento sinusoidal, esto nso indica que vamos a tener que aplicar una diferenciación con 12 instantes anteriores. Si nos fijamos en la gráfica de la estacionalidad vemos como en el mes de julio/agosto se produce un decaimiento de más del 30% en el volumen de pasajeros y se alcanza un pico en el mes de septiembre con un aumento del 40% con respecto al instante anterior, lo que significa un aumento del 10% con respecto a la media. Si nos fijamos en los residuos, estan en torno al 1 con una variación pequeña.

Vamos a ver la serie diferenciada.

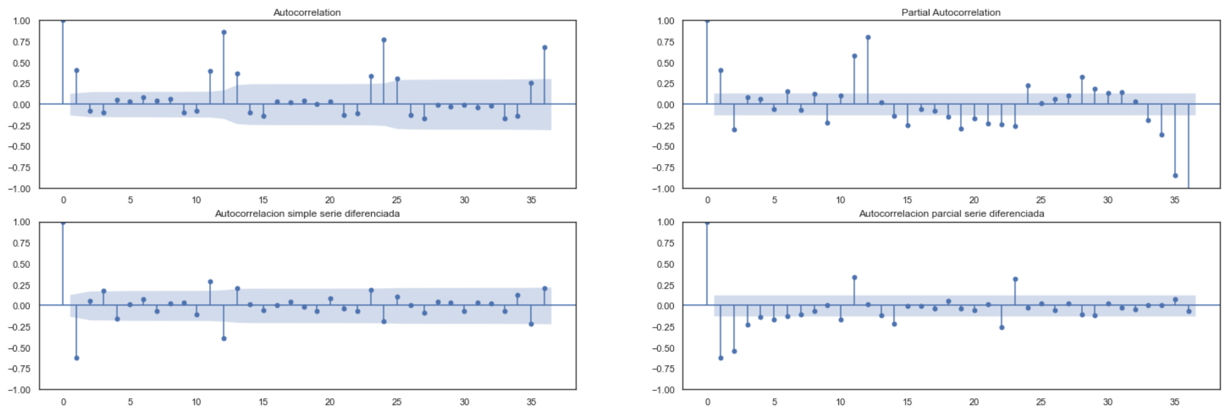


Aplicando esta primera diferenciacion hemos centrado la media en 0 por lo que hemos conseguido parte de las hipótesis para poder aplicar un modelo ARIMA. Pero vemos que sigue existendo periodicidad en la serie, esto se debe a que no hemos aplicado la diferenciación estacional. Vamos a aplicarle sobre esta diferenciación una diferenciación estacional de orden 12 para ver si conseguimos eliminar este efecto periódico que presenta la serie.



Vemos como al aplicar esta segunda diferenciación de orden 12 hemo eliminado la periodicidad de la serie. Parece que estamos ante unas buenas condiciones para aplicar el modelo, para estimar bien los parámetros del modelo ARIMA, necesitamos fijarnos en los autocorrelogramas de la serie ya diferenciada, pues es lo que nos indicará como estimar bien los parámetros para nuestro modelo ARIMA.

Ya sabemos que tenemos que marcar la diferenciación de orden 1 y de orden 12 en la componente estacional.



Vemos como el aplicar logaritmos y diferenciar la serie, hemo eliminado practiamente la autocorrelación. Debemos corregir las autocorrelaciones que no hemos sido capaces de aproximar a 0, eso lo conseguimos modelando los parámetros del ARIMA.

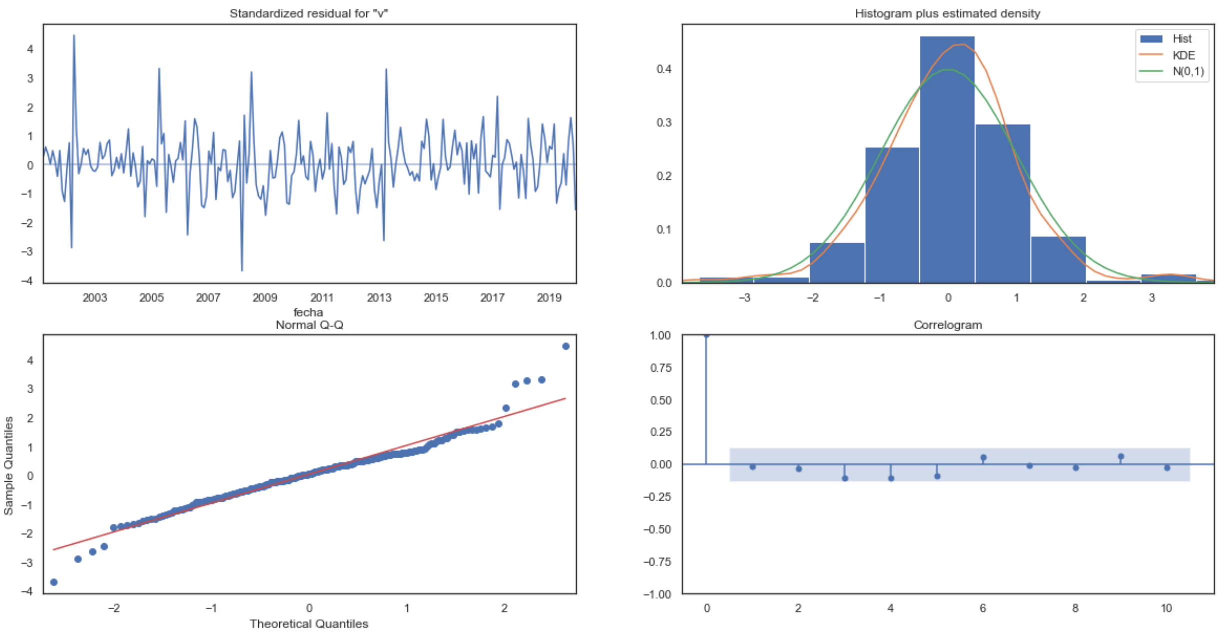
Estamación de parámetros y representación de errores

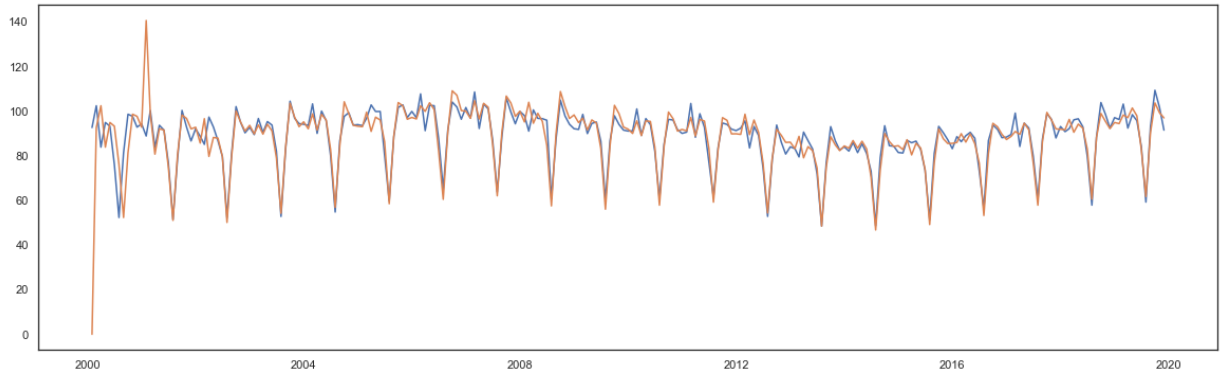
Tendremos un primer modelo $ARIMA(4,1,0)(0,1,1)_{12}$, donde $(4,1,0)$ representa los 4 autcorrelogramas que salen del parcial, el 1 para representar la primera diferenciación. Y en la componente estacional tenemo un $(0,1,2)_{12}$ donde el 1 se refiere a la diferenciación de orden 12 (la estacional) y el 2 la modelización de la autocorrelación de orden que aparece en el momento 12 y en el 24.

```
                                      SARIMAX Results                                      
==========================================================================================
=========
Dep. Variable:                           volumenMetro   No. Observations:                  
239
Model:             ARIMA(4, 1, 0)x(0, 1, [1, 2], 12)   Log Likelihood                  
-609.211
Date:                                Sun, 06 Feb 2022   AIC                             
1232.422
Time:                                        19:11:16   BIC                             
1256.365
```

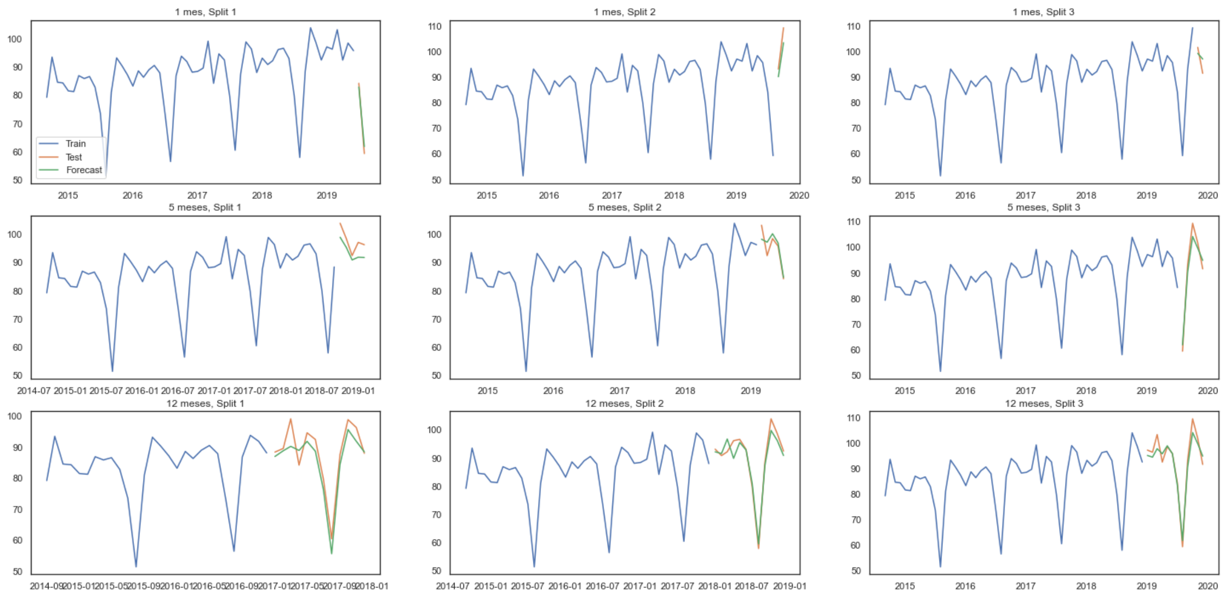
```
Sample:                                         02-01-2000   HQIC
1242.084
                                              - 12-01-2019
Covariance Type:                                       opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.9932      0.062    -15.940      0.000      -1.115      -0.871
ar.L2         -0.7359      0.124     -5.914      0.000      -0.980      -0.492
ar.L3         -0.2790      0.118     -2.372      0.018      -0.509      -0.049
ar.L4         -0.1164      0.089     -1.314      0.189      -0.290       0.057
ma.S.L12      -0.6814      0.071     -9.572      0.000      -0.821      -0.542
ma.S.L24      -0.0893      0.069     -1.303      0.193      -0.224       0.045
sigma2        12.2282      1.080     11.327      0.000      10.112      14.344
===================================================================================
Ljung-Box (L1) (Q):                   0.09   Jarque-Bera (JB):                78.94
Prob(Q):                              0.76   Prob(JB):                         0.00
Heteroskedasticity (H):               0.66   Skew:                             0.31
Prob(H) (two-sided):                  0.07   Kurtosis:                         5.83
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-ste
p).
```

Vemos como el p-valor es 0.189 para la componente autoregresiva y de 0.193 para la parte de medias móviles, ambos mayores que 0.05 pues podemos confirmar que se cumple la hipótesis de que los residuos están incorrelados. Debajo vemos como para el primer momento el autocorrelograma es 1 y después desciende en todos sus valores por debajo del intervalo de confianza, pues podemos asumirlos como 0.





Vamos a realizar predicciones para 1, 5 y 12 meses, de tal forma que le aplicaremos un C-V de time series de 3 splits a cada una de las predicciones. Este C-V se trata de crear ventanas de tiempo que van cogiendo de menos a más conjunto de train y siempre guardan una ventana como test, para poder hacer la validación.



En el gráfico vemos como la linea azul representa el conjunto de Train, si nos fijamos en los ejes para el primer split y para el resto, la serie temporal termina en distintos momentos, esto se debe a la variación del conjunto de train, como ya habíamos comentado. La lina naranja simpre es de la misma longitud, de 1, 5 y 12 meses en función de los que vayamos a predecir y la usamos como referencia para ver la calidad de nuestras predicciones. Y la verde representa a las predicciones que hace nuestro modelo. Como vemos, hay en ocasiones donde ajusta realmente bien y en otros momentos no es capaz de ajustarse tanto. Aunque parece a simple vista que la prediccion no es muy mala.

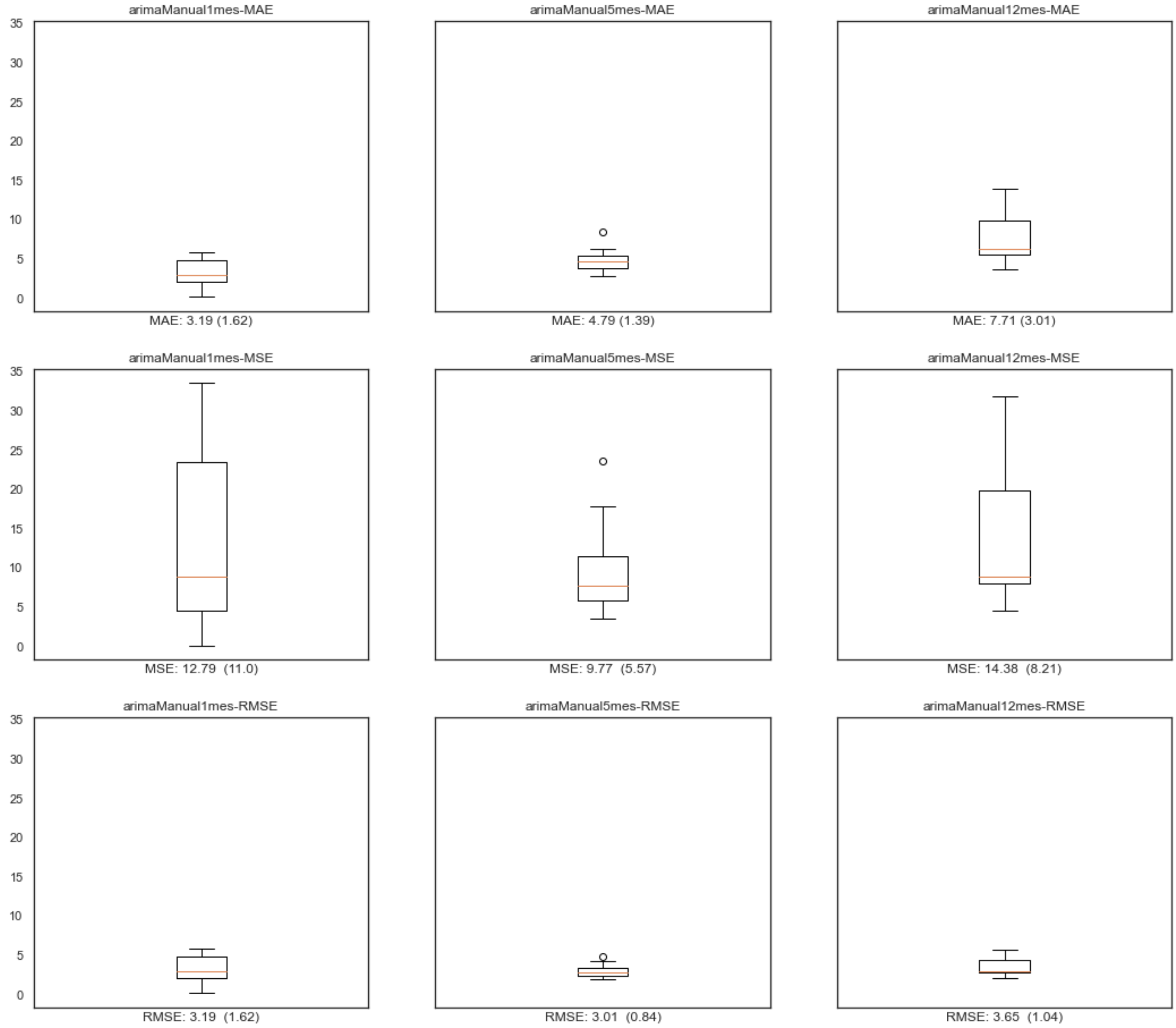
Si nos fijamos en la siguiente tabla, vemos los errores medios cometidos en cada una de las iteraciones por mes, en esta ocasion, hemos ejecutado el C-V con 15 splits para coger una media más real, ya que con solo 3 splits quizá no representa del todo bien la realidad.

Vamos a analizar el MAE, MSE y RMSE, para las predicciones de 1, 5 y 12 meses. Siendo n el numero total de observaciones, $\hat{y}_i$ el valor predico e y_i el valor observado en el instante i, donde $i = 1, 2, \ldots, n$. Se tiene que:

El Error absoluto medio (MAE) viene dado por $\frac{1}{n} \cdot \sum \|\hat{y}_i - y_i)\|$ con $i = 1, \ldots, n$

El Error cuadrático medio (MSE) viene dado por $\frac{1}{n} \cdot \sum (\hat{y}_i - y_i)^2$ con $i = 1, \ldots, n$

El Raiz del error cuadrático medio (MSE) viene dado por $\sqrt{\frac{1}{n} \cdot \sum(\hat{y}_i - y_i)^2}$ con $i = 1, \ldots, n$



En el gráfico vemos la comparativa de los distintos errores por meses, además estamos mostrando el valor medio y la desviación estandar en cada uno de los gráficos.

Si analizamos los errores absolutos, el que mejor parado sale es la prediccion a un mes, tiene una media menor y además la desviación típica no es muy elevada. Sin embargo si nos fijamos en los errores cuadráticos, vemos como la mejor estimación es para 5 meses tanto en media como en varianza es el error más bajo. Parece que nuestro modelo estima mejor para 1 y 5 meses que para 12, esto es algo lógico, pues cuando más tiempo queramos estimar, más error deberíamos cometer.

Modelo AutoArima

Hemos realizado este mismo estudio con el modelo Autoarima que ofrece la librería pmdarima, la librería ofrece una funcion auto_arima, que itera sobre los distintos valores que pueden tomar los parámetros, mostrando cuál es el mejor modelo ARIMA para esta serie.

```
                                      SARIMAX Results
====================================================================================
===========
Dep. Variable:                                      y   No. Observations:
239
Model:             SARIMAX(3, 1, 1)x(1, 0, [1, 2], 12)   Log Likelihood
```

```
-654.369
Date:                            Sun, 06 Feb 2022   AIC
1324.738
Time:                                    19:14:30   BIC
1352.516
Sample:                                         0   HQIC
1335.933
                                            - 239
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.2684      0.092     -2.932      0.003      -0.448      -0.089
ar.L2         -0.0697      0.074     -0.938      0.348      -0.215       0.076
ar.L3          0.1395      0.069      2.019      0.043       0.004       0.275
ma.L1         -0.7460      0.068    -11.032      0.000      -0.879      -0.613
ar.S.L12       0.9988      0.001   1114.357      0.000       0.997       1.001
ma.S.L12      -0.6739      0.072     -9.376      0.000      -0.815      -0.533
ma.S.L24      -0.1259      0.066     -1.903      0.057      -0.255       0.004
sigma2        11.7712      1.008     11.677      0.000       9.795      13.747
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):                75.29
Prob(Q):                              0.94   Prob(JB):                         0.00
Heteroskedasticity (H):               0.64   Skew:                             0.07
Prob(H) (two-sided):                  0.05   Kurtosis:                         5.75
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-ste
p).
```

En esta ocasión nos ha recomendado un modelo $ARIMA(3,1,1)(1,0,(1,2))_{12}$ esto quiere decir que ha producido raíces complejas en el modelo. El p-valor asociado a la parte estacional es mayor que 0.05 por lo que podemos aceptar la hipótesis de que las autocorrelaciones son ruido blanco.

Vamos a guardar los errores como en el modelo que hemos propuesto manualmente, para posteriormente comprarlas.

Modelo ARIMA con variables exógenas

Vamos a utilizar el modelo autoarima de nuevo pero esta vez utilizaremos las variables exógenas para ver si podemos mejorar la calidad de nuestras predicciones. Este modelo hemos ido aplicando un método stepwise para la elección de las variables. Además, hemos probado con variables creadas por nosotros, como la desviación estandar por mes de las temperaturas o las velocidades del viento. Otras de las variables que hemos probado añadiendo la serie direfenciada una vez y estacional, la media, la mediana y la desviación estandar móviles en los periodos 6 y 12 meses.

Después de este proceso de selección, las variables escogidas para apoyar a la predicción fueron: tMax, tMaxStd, pressMinStd.

```
                                     SARIMAX Results
=====================================================================================
======
Dep. Variable:                                y   No. Observations:
```

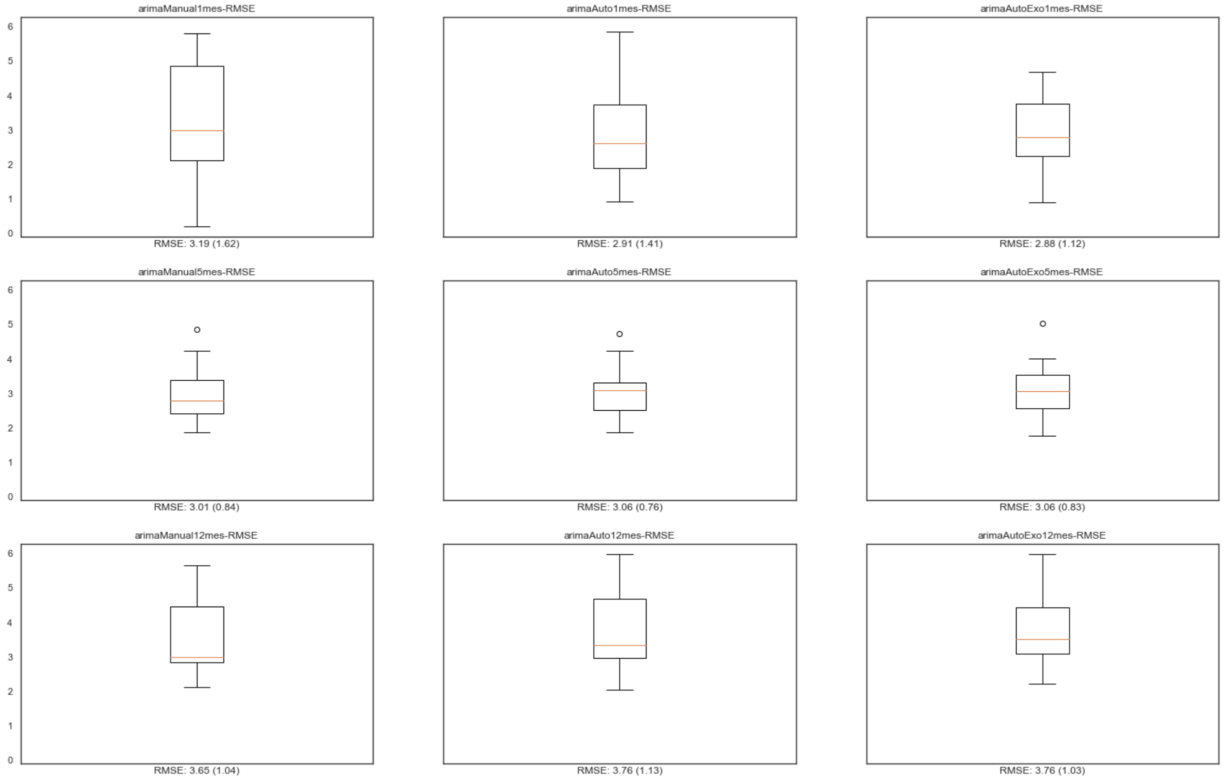
```
239
Model:             SARIMAX(3, 1, 1)x(1, 0, 1, 12)   Log Likelihood                -6
53.430
Date:                            Sun, 06 Feb 2022   AIC                           13
26.861
Time:                                    19:47:26   BIC                           13
61.584
Sample:                                02-01-2000   HQIC                          13
40.855
                                     - 12-01-2019
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
tmax          -0.0023      0.045     -0.051      0.960      -0.090       0.085
tmaxStd        0.0065      0.046      0.142      0.887      -0.084       0.097
presMinStd     1.0649      0.615      1.730      0.084      -0.141       2.271
ar.L1         -0.2992      0.154     -1.941      0.052      -0.601       0.003
ar.L2         -0.0848      0.162     -0.523      0.601      -0.403       0.233
ar.L3          0.1845      0.100      1.850      0.064      -0.011       0.380
ma.L1         -0.7189      0.113     -6.348      0.000      -0.941      -0.497
ar.S.L12       0.9984      0.001   1043.740      0.000       0.997       1.000
ma.S.L12      -0.7643      0.059    -12.859      0.000      -0.881      -0.648
sigma2        11.6855      1.093     10.692      0.000       9.543      13.828
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):                55.40
Prob(Q):                              0.91   Prob(JB):                         0.00
Heteroskedasticity (H):               0.66   Skew:                             0.05
Prob(H) (two-sided):                  0.07   Kurtosis:                         5.36
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-ste
p).
```

Una vez escogidas las variables, el modelo autoArima, nos recomienda utilizar un modelo $ARIMA(3,1,1)(1,0,1)_{12}$, vamos a comprar y analizar cuál funciona mejor para el problema que hemos planteado. No vamos a utilizar redes neuronales, ya que al no tener un data set suficientemente grande la predicción no es buena, pero una red neuronal LSTM podría funcionar muy bien con datos diario o incluso mejor por minutos u horas.

Capítulo 3: Elección del modelo y puesta en producción

Vamos a analizar el RMSE para poder hacer una comparación mejor. Hemos escogido esta medida de error, ya que es una de las más recomendadas para predicciones de este estilo.

Vamos a comprar el modelo Manual sin variables exogenas, los modelos AutoArima sin y con variables exógenas.



Analizando el gráfico, vemos como para la estimación del primer mes, el mejor modelo parece ser el AutoArima con variables exogenas. Sin embargo para las estimaciones de 5 y 12 meses, parece que la mejor estimación la tiene el modelo Manual. En media de las 3 estimaciones, parece que es mejor el modelo AutoArima con variables exogenas, ya que aunque para las estimaciones de 5 y 12 meses tiene un mejor performance el modelo Manual, la diferencia no es tan significativa como para el resto de estimaciones. De forma global un RSME no parece un buen resultado.Es probable que debido a la poca información del dataset no podamos obtener un mejor performance. Independientemente de esto, como todoterreno es mejor el modelo AutoArima con variables exogenas, pero para una estimación a largo plazo, parece funcionar mejor el modelo Manual, que además es más sencillo de explicar, lo que haría que fuese un buen modelo para poder poner en producción, aunque juega con la desventaja que es menos polivalente, debido a que un modelo AutoArima, puede aprender y estimar nuevos parámetros para la siguiente predicción en función de los datos que van llegando a la tabla de entrenamiento del modelo.

Puesta en producción

Como método de trabajo para una puesta en producción del modelo, recomiendo el sugerido en Introducing MLOps, donde describe el flujo de trabajo para la puesta en producción de un modelo de ML. Este método esta basado en el ya conocido para otras aplicaciones de desarrollo como DevOps. La idea es integrar en un mismo equipo todos los componentes del flujo de trabajo de un modelo. Desde la extracción de los datos, mediante una ETL, que en nuestro caso sería una querie recurrente llamando a una base de datos y haciendo una unión de distintas tablas para obtener la información que necesitamos. Después una vez tenemos los datos, se prodería a un análisis de los datos como el expuesto en este documento, como es un modelo ARIMA, el método descrito por Bob-Jenkins es ideal para nuestra solución. Para encontrar el mejor modelo, los pasos a seguir serían el EDA, podríamos hacer un primer featuring engineering, donde buscaríamos variables escondidas en nuestro conjunto de datos, haríamos uno o dos primeros modelos, donde nos daríamos cuenta que variables tienen más importancia y esto nos llevaría a un segundo análisis de las variables, para ver si podemos utilizar algún método de transformación o aplicar transformaciones que aumenten el poder predictivo de nuestro modelo. Después de este segundo proceso de transformación, estaríamos en disposición de intentar encontrar el mejor modelo, con una comparativa de distintos modelos y haciendo una elección en función de las necesidades. Si el modelo es lo suficiente bueno, podríamos desarrollar una aplicación que hiciese llamadas a nuestra tabla y rellenase la columna con las predicciones. Esto podría ser mediante un repositorio de trabajo y creación de aplicaciones mediante el repositorio, después se procedería a automatizar la lectura y predecir valores.

Cuando nuestro modelo ya está en producción debemos monotorizar los resultados y si fuera necesario volver a empezar el flujo para adaptarlo, esto es esencial, pues para crear un modelo potente, es necesario ir adaptándolo, pues el entorno de preproducción es distinto al entorno de producción y no trabajará con un conjunto de datos tan pequeño.

Bibliografía

Treveil, M. & the Dataiku Team. (2020). Introducing MLOps (1.a ed., Vol. 1). O'Reilly.

Zheng, A., & Casari, A. (2018). Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. O'Reilly Media.

Hunter, J., Dale, D., Firing, E., Droettboom, M., & The Matplotlib development team. (2002-2021). Matplotlib. Matplotlib 3.5.1 documentation. https://matplotlib.org/stable/index.html

Mauricio, J.A. (2007) Introducción al análisis de Series Temporales. UCM https://www.ucm.es/data/cont/docs/518-2013-11-11-JAM-IAST-Libro.pdf

Kutzkov, K. (2022, 4 enero). ARIMA vs Prophet vs LSTM for Time Series Prediction. Neptune.Ai. https://neptune.ai/blog/arima-vs-prophet-vs-lstm

pmdarima.arima.auto_arima — pmdarima 1.8.4 documentation. (2022). Pmandarina Documentation. https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.auto_arima.html

Introduction — statsmodels. (2022). StatsModels API. https://www.statsmodels.org/dev/index.html

```
[NbConvertApp] Converting notebook memoria.ipynb to latex
[NbConvertApp] Support files will be in memoria_files\
[NbConvertApp] Making directory memoria_files
[NbConvertApp] Making directory memoria_files
[NbConvertApp] Making directory memoria_files
[NbConvertApp] Making directory memoria_files
[NbConvertApp] Making directory memoria_files
[NbConvertApp] Making directory memoria_files
[NbConvertApp] Making directory memoria_files
[NbConvertApp] Making directory memoria_files
[NbConvertApp] Making directory memoria_files
[NbConvertApp] Making directory memoria_files
[NbConvertApp] Making directory memoria_files
[NbConvertApp] Making directory memoria_files
[NbConvertApp] Making directory memoria_files
[NbConvertApp] Writing 52888 bytes to memoria.tex
```