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 290 km, 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 una buena organización del personal, volumen y frecuencia de trenes, es crucial a la hora de ofrecer el mejor servicio a los ciudadanos. Para realizar esta optima organización, realizaremos un estudio del volumen de pasajeros con tarjeta transporte que difrutan del Metro mensualmente, todo esto apoyandonos en datos meteorológicos, lo ideal sería tener datos con una frecuancia diaria o incluso por horas/minutos, disintguidos por lineas, para un óptimo análisis y provechamiento del modelo. Debido a que no tengo acceso a estos datos, lo realizamos mensualmente, aunque un estudio de estas características podría seguir las mismas lineas, con una mayor frecuencia de la variable temporal y con la diferencia del desglose por línea, lo que nos llevaría a utilizar un método de reducción de la dimensionalidad, por ejemplo clustering, para optimizar los tiempos de

entrenamiento y la producción de predicciones. Quizá un método adecuado para un problema de estas características sea uno basado en la densidad de las predicciones.

```
En nuestro conjunto de datos, tenemos 25 columnas y 396 filas, a continuación mostramos
una breve muestra del conjunto.
```

fecha	volumenMetro	tmed	prec	racha
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 mensuales 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 pequeño análisis exploratorio de las variables input y de la variable objetivo. La variable objetivo, en nuestro caso es el Volumen de pasajeros del Metro de Madrid, se trata de una variable númerica, que varia con el tiempo, por lo que nos encontramos ante un problema de series temporales. Además, tenemos un conjunto de variables input, que analizaremos para determinar si nos producen una mejora del modelo y por tanto un menor error en nuestras predicciones. En nuestro conjunto de datos, tenemos diferenciadas dos tipos de variables input.

Variable de tiempo(fecha), desde el año 2000 hasta el año 2020, la elección de estos años, es debido a que anterior al año 2000 el uso del Metro difiere mucho al actual. Además, en el año 2020 el volumen de pasajeros cayó drásticamente, debido al confinamiento domiciliario del año 2020. En una puesta a producción, lo ideal sería tener datos agrupados por días o incluso por horas, ya que así podríamos estimar los picos de pasajeros y determinar el volumen de trenes y trabajadores, la frecuencia de los vagones, etc. Además, enriquecería nuestro modelo, pues por lo general, a mayor cantidad de datos, mayor precisión podríamos alcanzar. Otra de las ideas para este análisis diario, sería introducir las varibles de "calendario", es decir, si se trata de un día laborables, cuándo es festivo, estacionalidad, picos horarios por entrada/salida del trabajo, etc. Todo esto, aunque en

nuestro modelo, no hemos podido utilizarlo, en la BBDD propuesta, si existe una tabla de calendario, donde hay un maestro de fechas, con todas estas variables.

También hemos recurrido a los datos meteorológicos, agrupados mensualmente y promediados por las 3 estaciones meteorológicas distintas que existen en la ciudad de madrid. Estos datos los facilita la AEMET con su API y contiene un apartado para desarrolo de aplicaciones en streaming, lo que ayudaría a utilizar un modelo en producción casi en tiempo real si lo requiere.

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

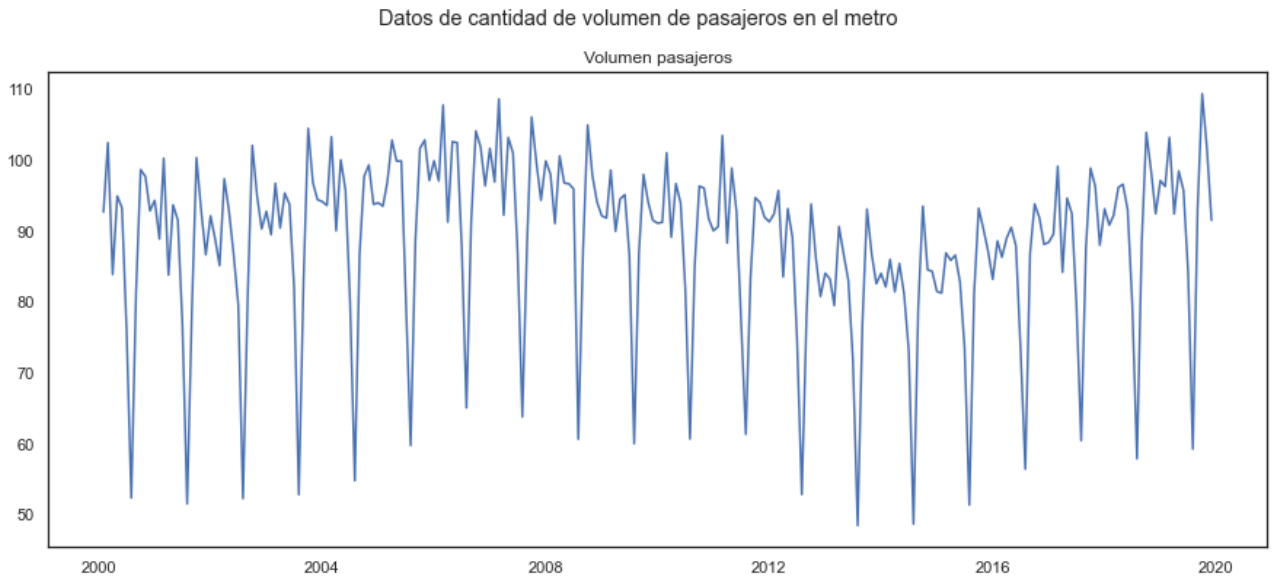
En el gráfico vemos como nos encontramos ante una serie casi en su totalidad estacionaria, hemos tenido que lidiar con una mala extracción de los datos, pues en el año 2010 aparecía una gran caida de pasajeros. La decisión ha sido promediar los datos del año anterior y posterior. Veamos que significa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

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.



Analizando la definición y gracias al gráfico, vemos como actualmente nuestra serie no es estacionaria, pues la media y la varianza no son constantes en el tiempo, debido a la tendencia sinusoidal de la series. Además, vemos como la serie presenta una cierta periodicidad de los datos, pues la forma que toma la serie es similar en todo el tiempo.

Para poder aplicar el modelo ARIMA, debemos "eliminar" esta componente estacional, mediante por ejemplo la diferenciación, normalización, transformación de la serie por logaritmos... Con estos ajustes buscamos obtener un ruido blanco.

Decimos que un proceso estocástico es un ruido blanco si

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

Por lo tanto, nuestro objetivo para poder 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, todo esto, confirma la no estacionalidad de nuestra serie.

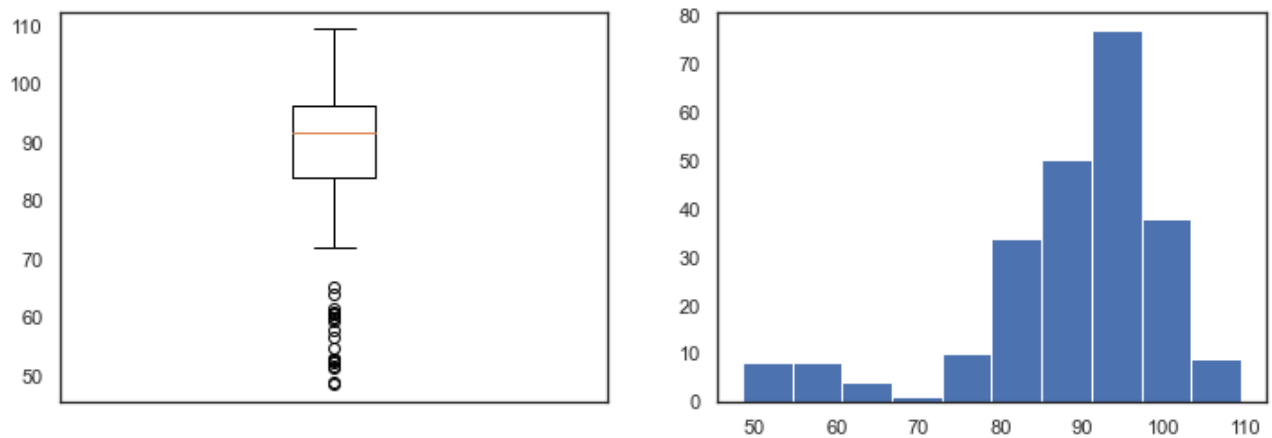
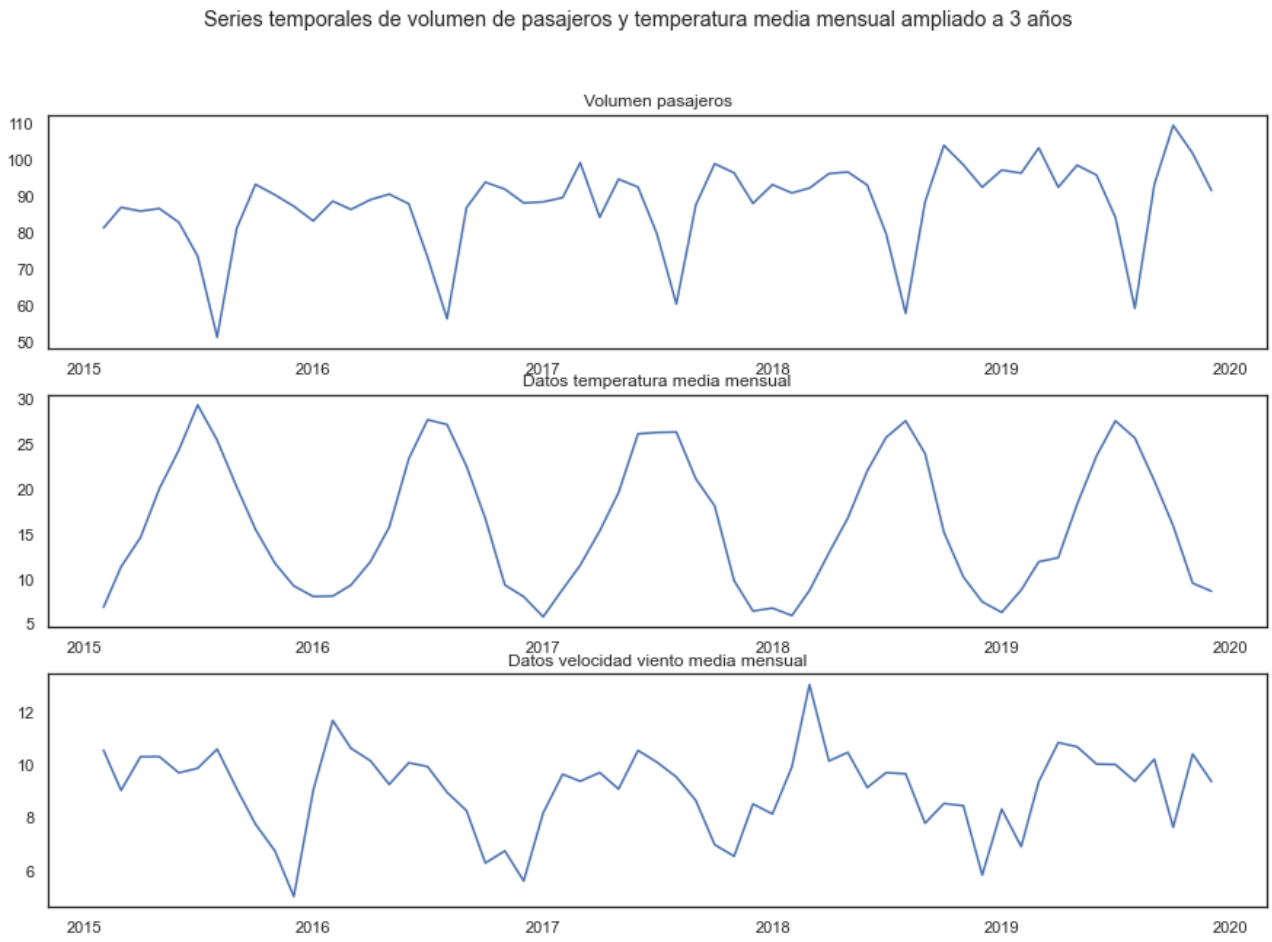


Fig 1.: Gráfico boxplot e histograma del volumen de Pasajeros del metro

Variables climatológicas

Las variables climatológicas, entre otras cosas, nos ayudan a percibir en que momento del año nos encontramos, pues la temperatura media no es la misma en Febrero que en Julio. Esto nos hace pensar que ayudará a nuestro modelo a producir predicciones más exactas. Para este análisis, hemos contrastado todas las variables climatológicas contra la variable objetivo, aquí vamos a mostrar un par, para ver la relación que puede llegar a tener este tipo de variables con la objetivo.



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.

Esto nos llevaría a pensar que si empieza aumenta la velocidad del viento en el mes actual, los proximos meses empezará la subida de las temperaturas y por lo tanto, disminuirá el volumen de pasajeros, algo que nos ayuda a predecir, las fluctuaciones de la serie objetivo.

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.

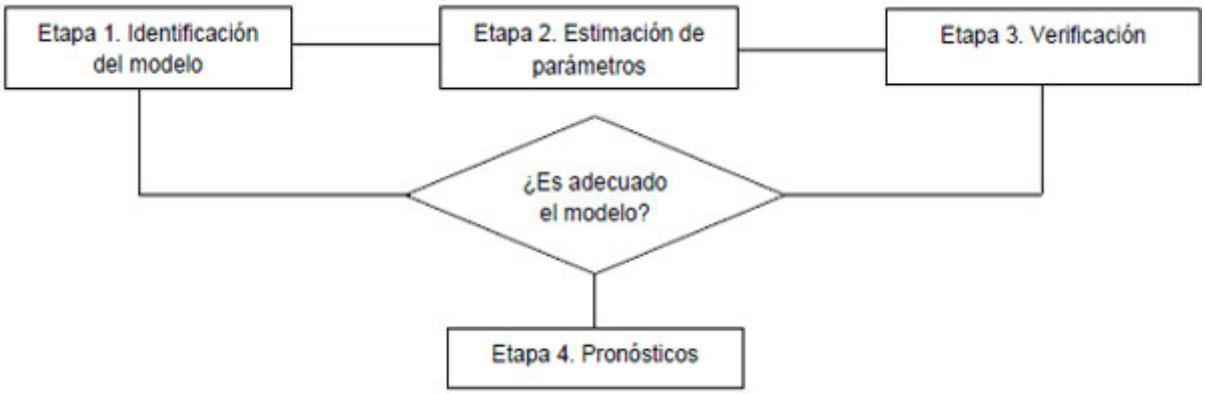
Decimos que aplicamos una diferenciación de orden k a una serie, cuando teniendo X_t le restamos la observación de k instantes anteriores, es decir, X_{t-k}.

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.

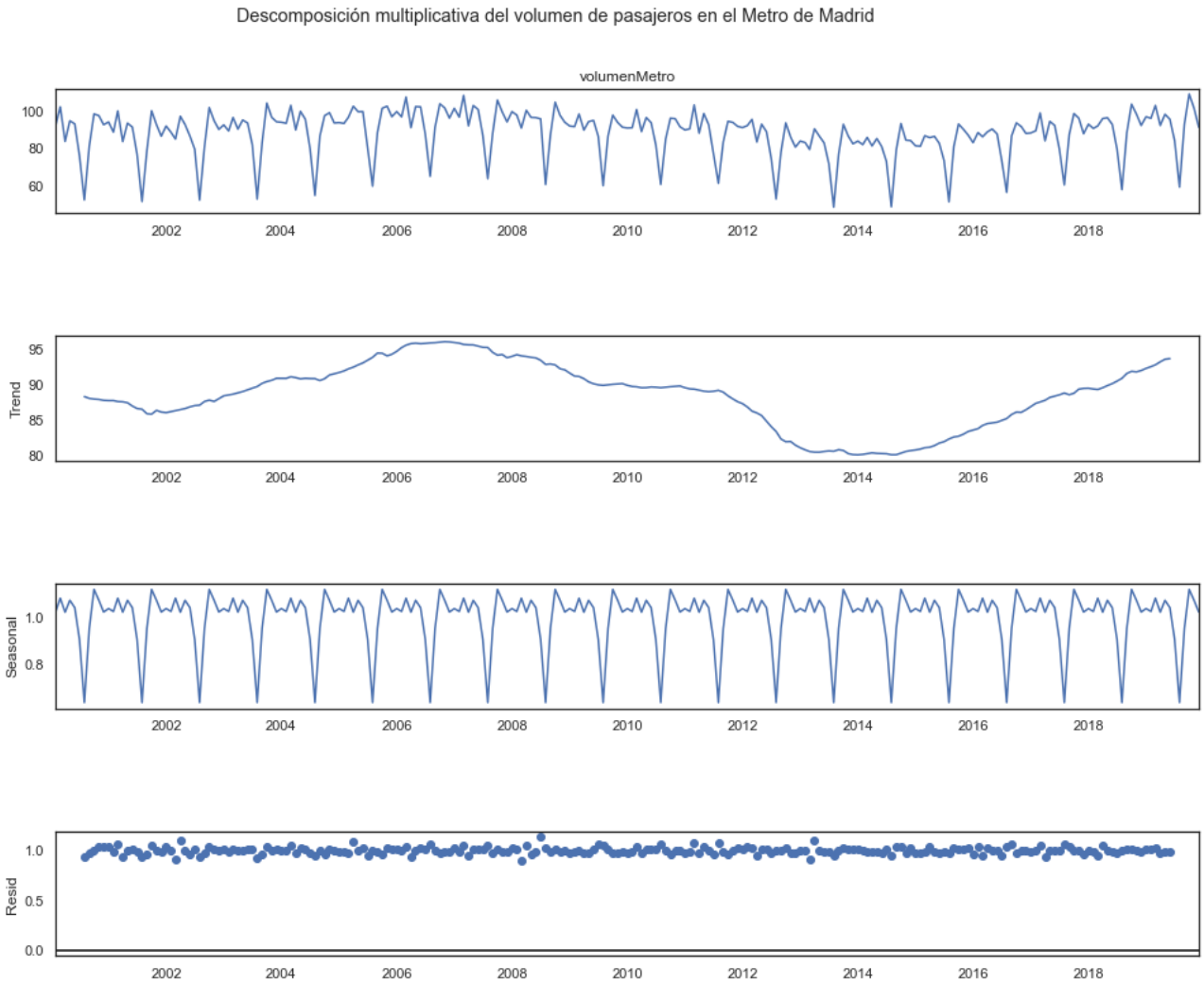
```
Esquema metodología Box-Jenkins
```

Out[12]:



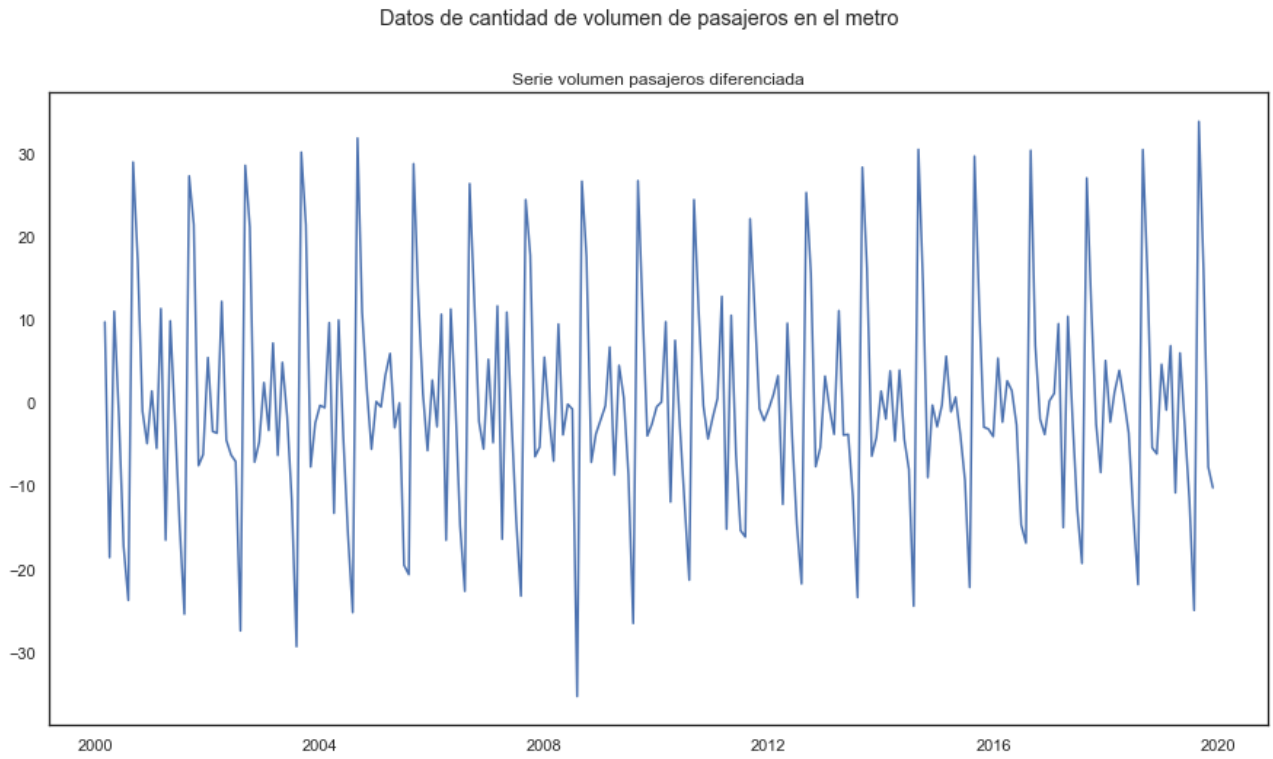
Serie y descomposición estacional

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

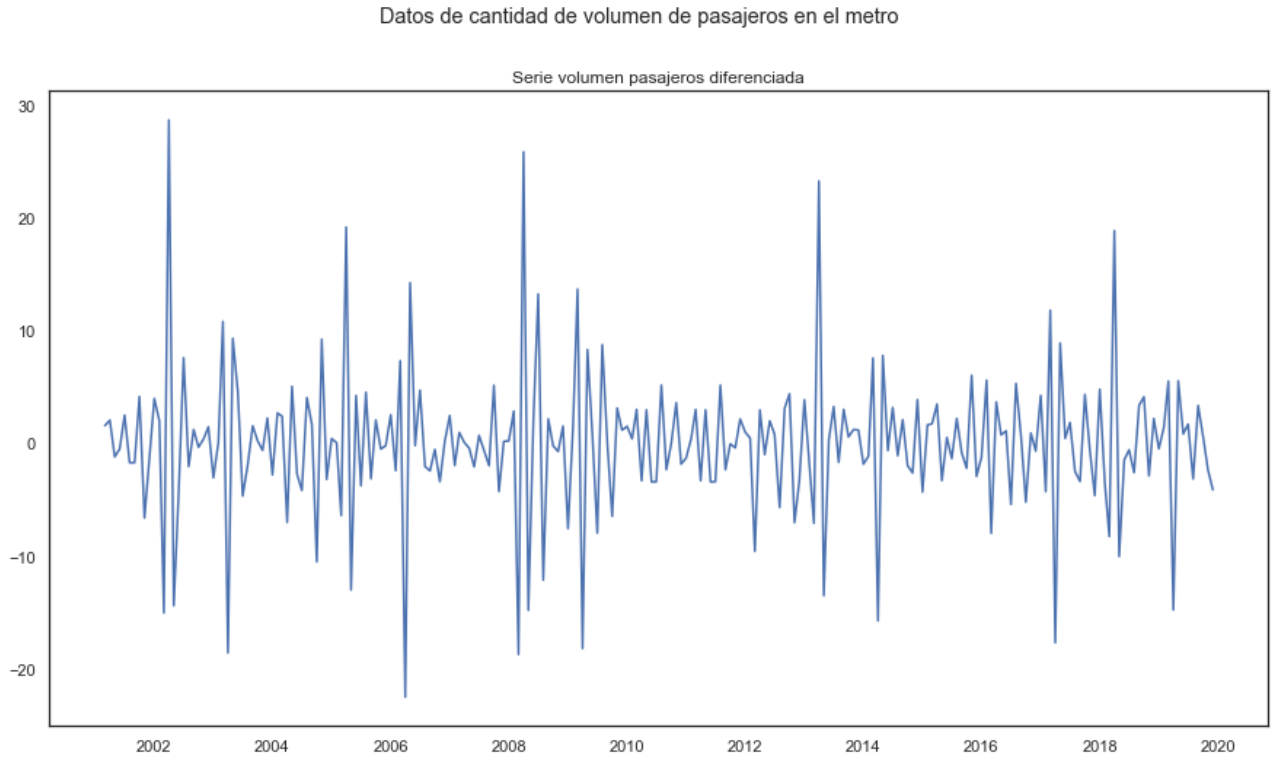


La serie ya hemos analizado que tiene una periodicidad de 12 meses, esta descomposición nos afirma que efectivamente esto sucede. Además como habíamos percibido la tendencia tiene un comportamiento sinusoidal, esto nos indica que vamos a tener que aplicar una diferenciación con 12 instantes anteriores. 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.



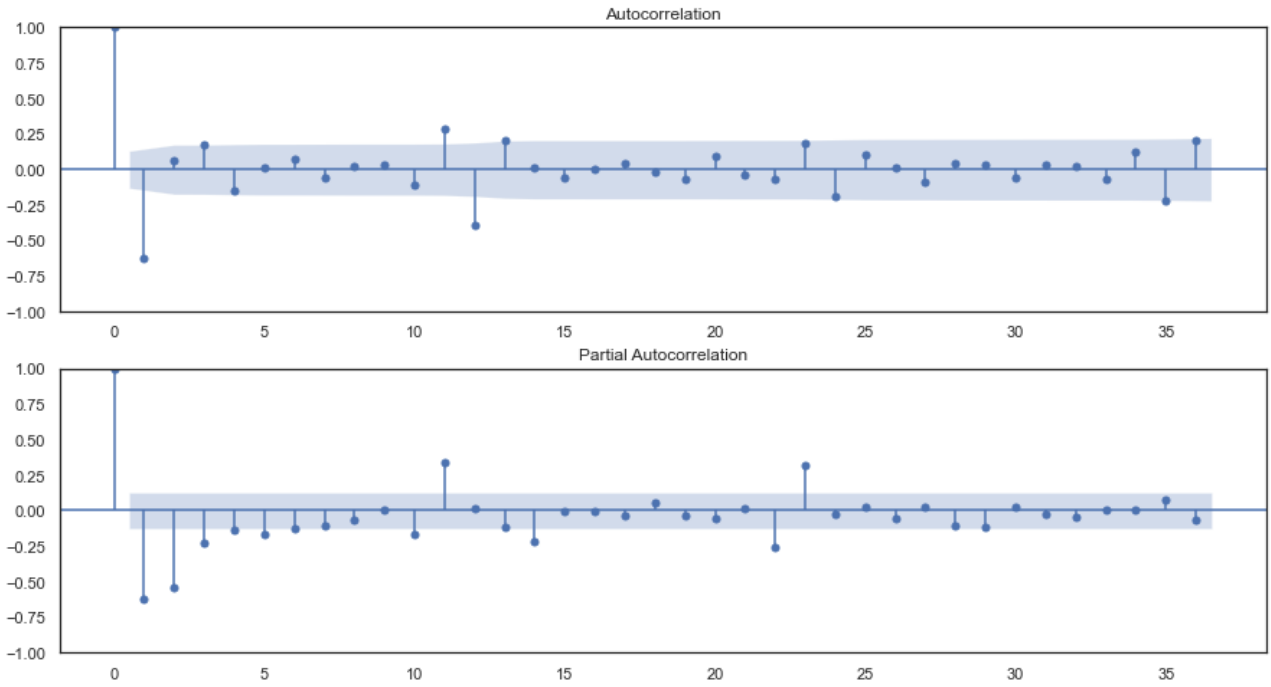
Aplicando esta primera diferenciacion hemos centrado la media en 0 por lo que hemos conseguido parte de las hipótesis para poder aplicar un modelo ARIMA. Pero vemos que sigue existendo periodicidad en la serie, esto se debe a que no hemos aplicado la diferenciación estacional. Vamos a aplicarle sobre 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 hemos 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.



Si nos fijamos en la gráfica de las gráfica, en la parte superior tenemos las autocorrelaciones simples de la serie diferenciada y en la parte inferior tenemos las autocorrelaciones parciales. Estas gráficas son las que nos ayudan a estimar en instantes de tiempo debemos reflejar en nuestros parámetros del modelo ARIMA. Si nos fijamos en las parciales, vemos como las 4 primeras autocorrelaciones no se encuentran dentro de la banda y cada 12 instantes tenemos un pico, que refleja la componente estacional. Además, tenemos que reflejar los los autocorrelogramas de la simple.

Estamación de parámetros y representación de errores

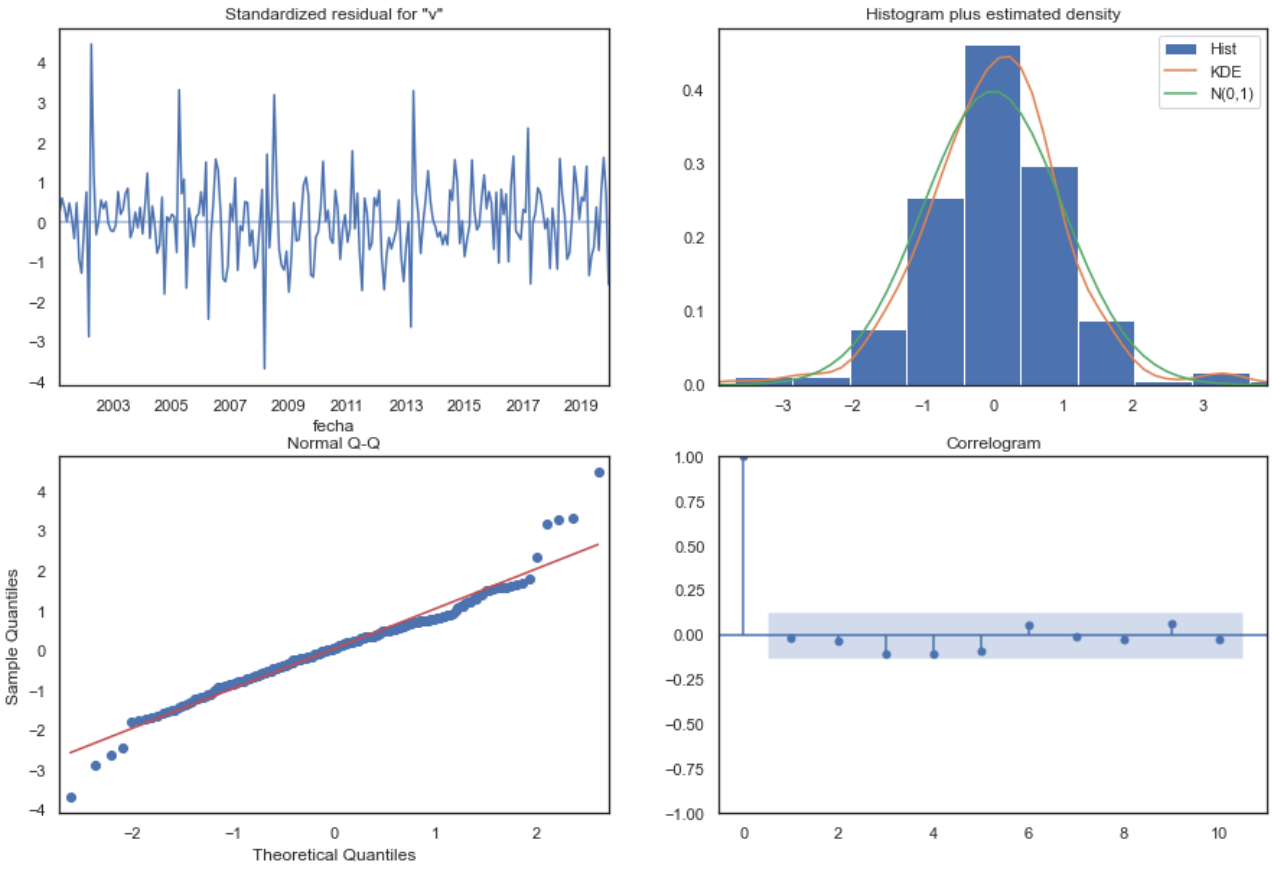
Tendremos un primer modelo $ARIMA(4,1,0)(0,1,2)_{12}$, donde $(4,1,0)$ representa los 4 autcorrelogramas que salen del parcial, el 1 para representar la primera diferenciación. Y en la componente estacional tenemo un $(0,1,2)_{12}$ donde el 1 se refiere a la diferenciación de orden 12 (la estacional) 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                -60
9.211
Date:                               Sun, 20 Feb 2022   AIC                           123
2.422
Time:                                       09:25:26   BIC                           125
6.365
```

```
Sample:                                 02-01-2000   HQIC                          124
2.084
                                      - 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-step).
```

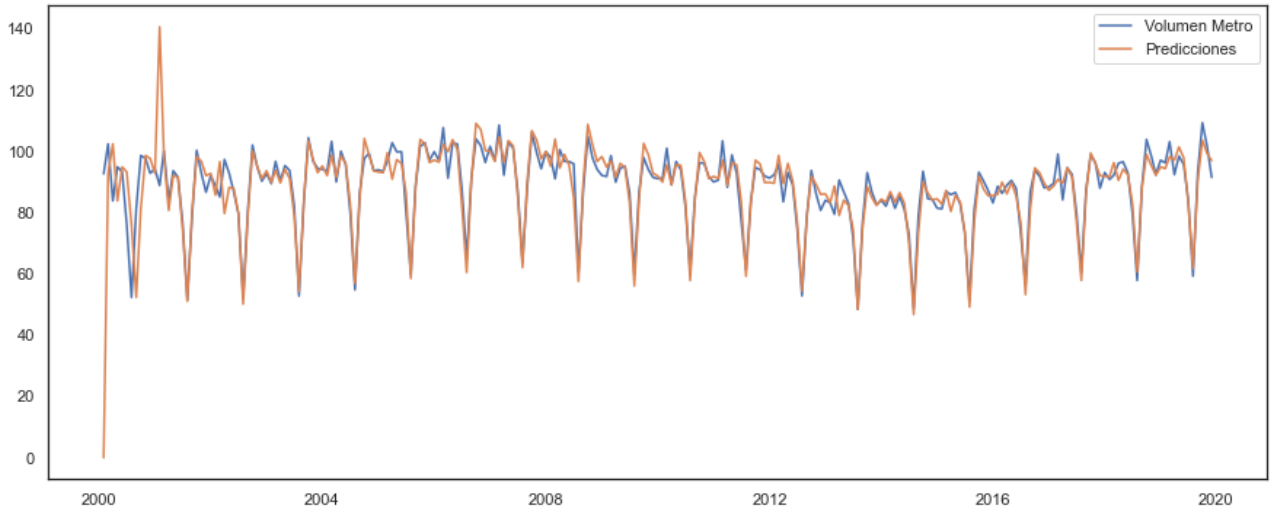
Vemos como el p-valor es 0.189 para la componente autoregresiva y de 0.193 para la parte de medias móviles, ambos mayores que 0.05 pues podemos confirmar que se cumple la hipótesis de que los residuos están incorrelados. Debajo vemos como para el primer momento el autocorrelograma es 1 y después desciende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



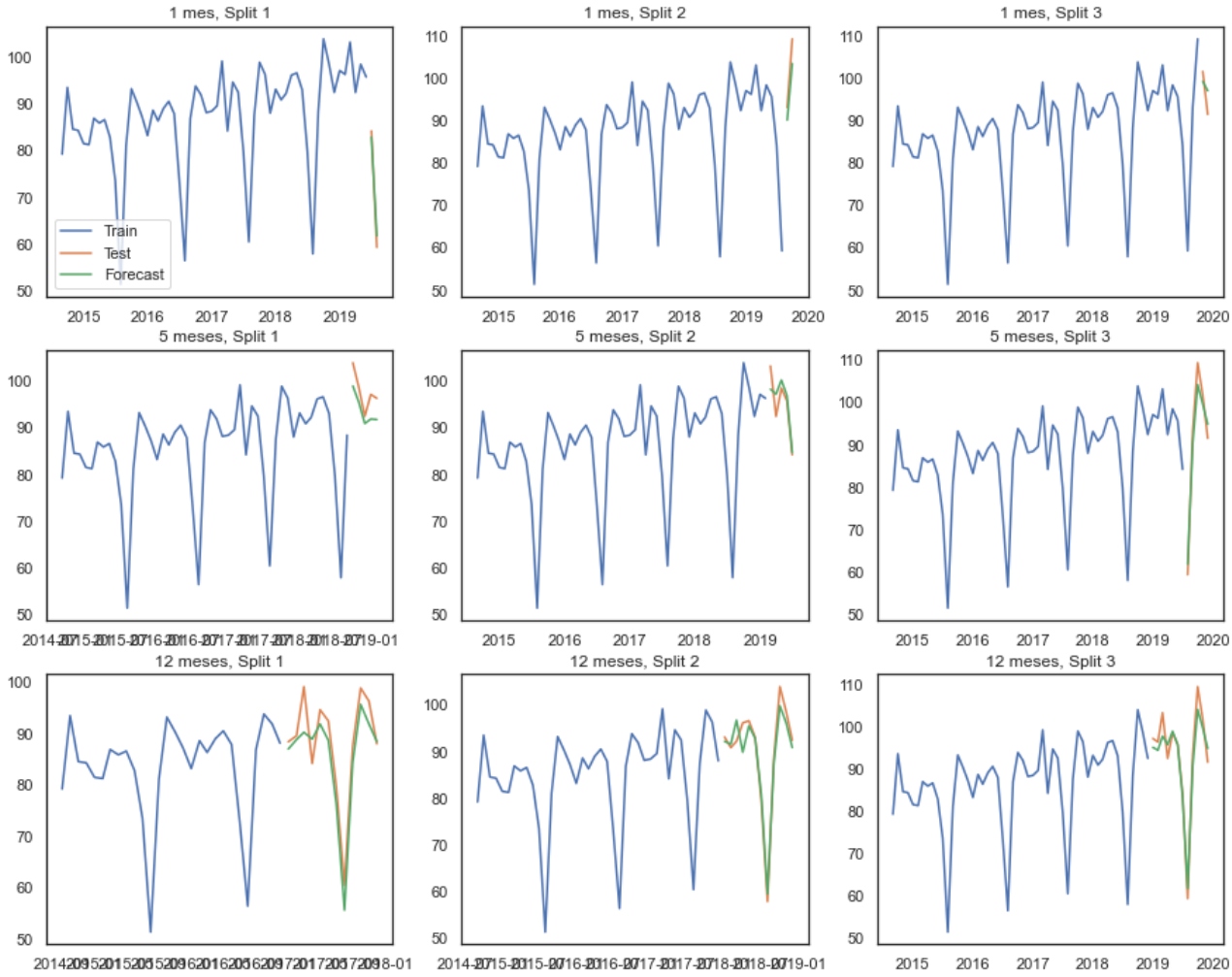
Además, ahora nuesta distribución se asemeja a una distribución $N(0,1)$, pues fiajandonos en el

gráfico de quantiles, vemos como exceptuando los extremos, la distribución está muy relacionada con la normal.

También hemos pintado la gráfica de nuestra serie y las predicciones que se obtienen con nuestro modelo. Vemos como aunque en ciertos momentos falla, sobre todo en los primeros años, esto es normal, ya que nuestro modelo aprende de los datos históricos y se basa en momentos anteriores, cuando aún no hay datos, hasta que aprende como es la serie, pasan un par de años, apartir de ahí y sobre todo en la parte final de la serie, las estimaciones son bastantes aproximadas a la realidad.



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.

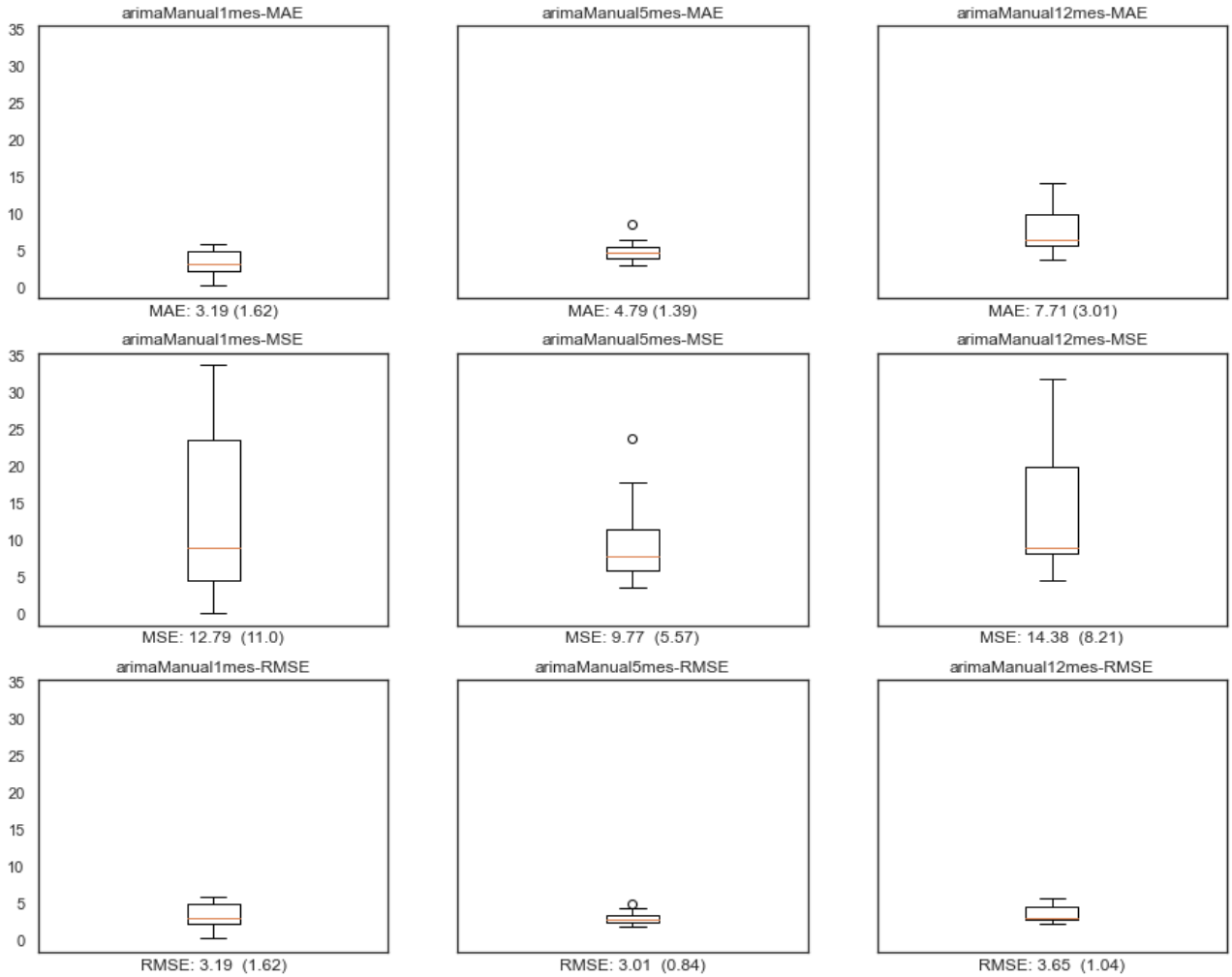
Si nos fijamos en la siguiente tabla, vemos los errores medios cometidos en cada una de las iteraciones por mes, en esta ocasion, hemos ejecutado el C-V con 15 splits para coger una media más real, ya que con solo 3 splits quizá no representa del todo bien la realidad.

Para analizar los errores, debemos recordar que nuestra escala hace refencia a miles de personas. 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, 20 Feb 2022   AIC                           1
324.738
```

```
Time:                                        09:28:34   BIC                            1
352.516
Sample:                                             0   HQIC                           1
335.933
                                                - 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-step).
```

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. Realizamos la misma validación que para el modelo manual, guardando los errores y pudiendo así poder hacer una comparativa.

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 un proceso de selección backward, es decir, utilizando todas las variables y poco a poco descartando las menos significativas, las variables escogidas para apoyar a la predicción fueron: tMax, tMaxStd, pressMinStd.

```
                                SARIMAX Results
==========================================================================================
==
Dep. Variable:                                 y   No. Observations:                  2
39
Model:             SARIMAX(3, 1, 1)x(1, 0, 1, 12)   Log Likelihood                -653.4
30
```

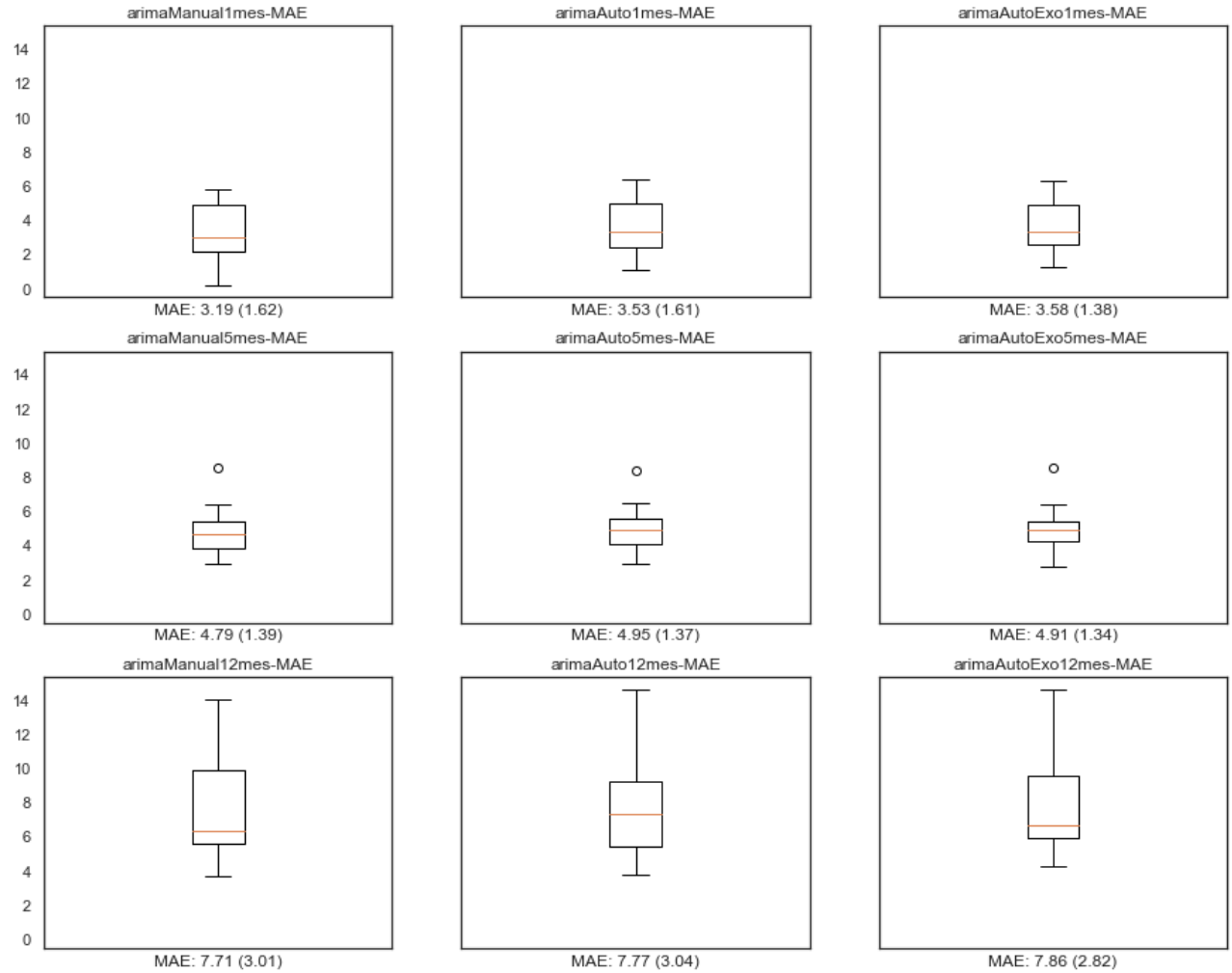
```
Date:                               Sun, 20 Feb 2022   AIC                            1326.8
61
Time:                                     09:34:15   BIC                            1361.5
84
Sample:                                 02-01-2000   HQIC                           1340.8
55
                                      - 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-step).
```

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/horas.

Capítulo 3: Elección del modelo y puesta en producción

Vamos a analizar el MAE para poder hacer una comparación mejor. Hemos escogido esta métrica pues es más robusta que el resto y detecta bien si un modelo produce predicciones muy malas. Vamos a comprar el modelo Manual sin variables exogenas, los modelos AutoArima sin y con variables exógenas.



Analizando los gráficos, nos fijamos como cuantos más meses querramos predecir, más error vamos a cometer, pero esto ya lo habíamos percibido anteriormente. Lo importante es fijarnos en que aunque la media de los errores es muy similar en ambos modelos, la desviación típica no lo es, teniendo una desviación típica inferior en el modelo AutoArima con variables exógenas. Esto es relevante debido a la métrica que hemos elegido. Como ya comentamos, esta métrica es sensible ante outliers, pues al realizar un máximo, detecta bien cuando el modelo comete mucho error y esto nos conduce a elegir el modelo con variables Exogenas, que para todas las predicciones tiene una menor varianza y aunque la media del error en algún momento es mayor, queremos un modelo que estime lo mejor posible para todas las situaciones y esto nos lo indica una media parecida y una varianza más baja.

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.
Finalmente para la toma de decisiones podríamos utilizar un dashboard para mostrar los valores y poder decidir a través de la visualización de los datos. La propuesta de un desboard de este estilo debe ser algo sencillo, con información suficiente como para hacer una buena toma de decisiones.

```
Propuesta de dashboard
```

Out[34]:

He querido plasmas una comparativa de el supuesto momento actual, en este caso he escogido el año 2019 y con las predicciones que obtuvimos de nuestro modelo, hemos dibujado el próximo año, al menos lo que se espera. Además, hemos incluido un histórico de como ha ido evolucionando nuestra serie a lo largo de los últimos 20 años. Podemos encontrar KPI's del volumen total que se espera en el periodo, el máximo y el mínimo y en las gráficas vemos en que puntos se alcanza cada

uno de ellos. Todo esto con un juego de colores acorde al logo de nuestra marca y sin leyendas, ya que los títulos nos sugieren a que momento hacen referencia cada color.

Tableau, además nos permite compartir mediante Tableau Server el dash con otras personas de la compañía, algo que facilita la toma de decisiones. En nuestro caso, como muestra, hemos utilizado Tableau public, para compartir nuestro análisis.

https://public.tableau.com/views/AnlisisMtrodemadrid/Dashboard1?:language=es-ES&:display_count=n&:origin=viz_share_link

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.

Conclusiones

El volumen de pasajeros mensual del metro está claro que tiene una componente estacional muy marcada, además tiene una tendencia cíclica de crecimiento y decrecimiento, lo que hace que un modelo tradicional como es el ARIMA, funcione correctamente. La falta de precisión en los datos ha provacdo un error elevado, con un mayor detalle podríamos haberlo reducido y además se podría realizar un estudio más exhaustivo del volumen de pasajeros, pero probablemente, tuviésemos que escalar a un modelo de redes LSTM para poder afinar en las predicciones cuando el volumen de datos aumenta. El análisis de este estudio, además, está agrupado por todas las lineas de metro, un correcto clustering de las series temporales de cada linea, sería crucial a la hora de minimizar costes computacionales y rapidez del modelo, pues estimar tres o cuatro clusters no es comparable con las trece lineas que existen actualmente, con sus respectivos datos. Esto ocasiona un aumento de la base de datos considerable.

A lo largo de este estudio, me he dado cuenta de la importancia de los pequeños detalles a la hora de crear un modelo de ML. Durante este proyecto he buscado información de todo tipo acerda de las series temporales, como crear una base de datos sencilla SQL con python, la extracción de datos públicos mediante API o modelado de estos para que encajen en nuestro conjunto de datos y sean consumibles para nuestro modelo. La utilización de git, como medio de control de Versiones, que aunque en este caso el trabajo era individual, te das cuenta como unos simples comandos pueden facilitar la toma de decisiones mediante distintos caminos, o directamenta guardar de manera segura el proyecto en un repositorio. La elección de lenguaje de programación, que aunque un análisis estadístico de series temporales como el propuesto hubiese sido más eficiente R como core del proyecto, la versatilidad que me ha ofrecido los cuadernos de jupyter, han sido decisivos a la hora de escribir la memoria y por tanto, desarrollar el trabajo en python.

La utilización de herramientas de BI no solo como análisis final a la hora de plasmar los resultados de tu modelo de una manera sencilla y visual, que facilite la toma de decisiones, si no que para hacer el EDA ha sido útil esta herramienta, pues sin tener que programar visualizas una base de datos de menera rápida, para después poder argumentar de manera directa en el documento.

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

T. Warren Liao, Clustering of time series data—a survey, Pattern Recognition, Volume 38, Issue 11, 2005, Pages 1857-1874, ISSN 0031-3203, https://doi.org/10.1016/j.patcog.2005.01.025.

A.M. Alonso, J.R.Berrendero, A.Hernández, A.Juster, Time series clusterin based on forecast densities - A survey, Computational statistics & data analysis, Volume 51, 2006, Pages 762-776, https://doi.org/10.1016/j.csda.2006.04.035