UNIVERSITAT D'ALACANT
Facultat de Ciències
Facultad de Ciencias

FACULTAD DE CIENCIAS

GRADO EN MATEMÁTICAS

TRABAJO FIN DE GRADO

CURSO ACADÉMICO 2022-2023

TÍTULO:

MÉTODOS NO CLÁSICOS DE SERIES TEMPORALES EN DATA SCIENCE

AUTOR:

MARINA PEÑALVER RIPOLL

Resumen

Las series temporales son un conjunto de datos ordenados que comparten un factor temporal. El aumento en los datos disponibles ha provocado un aumento en la demanda de analistas de datos capaces de interpretar dicha información. El análisis de series temporales se centra en modelar su comportamiento y utilizar los modelos para producir predicciones.

En este trabajo, se ha estudiado las características más relevantes de las series temporales, profundizando en la descomposición de dichas series y diferentes métodos de predicción.

Primeramente, se ha presentado las definiciones básicas de proceso estocástico y estacionariedad, así como la descomposición tradicional de las series, para introducir los métodos clásicos de series temporales. En particular, se ha desarrollado los procesos de media móvil integrados autorregresivos estacionales (SARIMA), que surgen a partir de los procesos de medias móviles (MA) y los procesos autorregresivos (AR). También se estudia los modelos de suavizado exponencial, centrándose el algoritmo de Holt-Winters.

La segunda parte del trabajo se centra en algunos de los métodos de predicción surgidos recientemente. El algoritmo de Prophet, desarrollado por Facebook en 2017, es muy eficiente a la hora de trabajar con series temporales con fuertes relaciones de estacionalidad. Además, se analizan procesos autorregresivos utilizando métodos de Machine Learning.

Finalmente, se ha comparado los diferentes métodos empleados teniendo en cuenta la precisión de las predicciones, señalando las diferencias entre ellos. Todo el código usado se ha desarrollado en exclusiva para este trabajo y se encuentra disponible en el repositorio de GitHub de la autora.

Palabras clave: Predicción, SARIMA, Suavizado exponencial, Método de Holt-Winters, Prophet, Machine Learning.

Abstract

Índice

1. Introducción **5**

1.1. Sobre el dataset 6

2. Series temporales **10**

2.1. Modelos estacionarios y estrictamente estacionarios 11

2.2. Descomposición de una serie temporal 13

3. Procesos autoregresivos y de media móvil **15**

3.1. Procesos ARIMA y SARIMA 17

4. Suavizado exponencial **22**

4.1. Suavizado exponencial simple 22

4.2. Método lineal de Holt 25

4.3. Algoritmo de Holt-Winthers 27

5. Prophet **30**

5.1. Modelos de tendencia 31

5.2. Estacionalidad 32

5.3. Días festivos y eventos 34

5.4. Prediciones con Prophet 35

6. Modelos autorregresivos con árboles de decisión **40**

6.1. Método multipaso recursivo 41

7. Comparación de modelos **45**

8. Conclusiones **47**

Referencias **48**

A. Detalles del desarrollo del trabajo **50**

1. Introducción

El mundo se actualiza en cada segundo, cada vez son más los dispositivos capaces de registrar y almacenar información. Como consecuencia, los métodos y estrategias de análisis de datos han evolucionado para intentar entender y aprovechar dichos datos. El análisis de datos se ha convertido en una disciplina fundamental en el mundo actual, ya que gracias a él es posible entrenar modelos de aprendizaje para tomar decisiones, optimizar procesos o predecir resultados.

Una rama clave del análisis de datos es el análisis de series temporales, que son conjuntos de datos que se recopilan secuencialmente en intervalos de tiempo. Estos datos pueden ser observaciones de variables en diferentes momentos, como ventas diarias o registros de tráfico web. El análisis de series temporales se enfoca en descubrir patrones, tendencias y relaciones en estos datos a lo largo del tiempo.

A mediados del siglo XX empezaron a surgir diferentes métodos con el fin de modelar las series temporales. Dos de los procesos más populares en el día de hoy son los procesos autorregresivos (AR) y de medias móviles (MA), juntándolos se obtienen los procesos de media móvil autorregresivos. El componente autoregresivo se refiere a la dependencia lineal que existe entre los valores pasados de una serie temporal y sus valores futuros, mientras que el componente de media móvil se utiliza para modelar los efectos de los errores. De esta forma se consigue modelar un tipo de procesos muy importante, los procesos estacionarios.

La evolución de los procesos no se quedó ahí, siguió evolucionando e incorporando nuevos componentes hasta que se propuso los modelos SARIMA (*seasonal autoregressive integrated moving average*), además de las propiedades de los modelos ARMA también se le añade un componente estacional y los procesos integrados. Aunque estos no son los únicos procesos que surgieron, existen muchos otros como los VARIMA, ARARMA, pero este proyecto solo se centra en los SARIMA.

Otros de los métodos que tuvo una gran repercusión en el mundo de las series temporales son el el suavizado exponencial, trabajo de Brown, Holt y Winters en los años 50 y 60. La técnica de suavizado exponencial se basa en la premisa de que los valores recientes de una serie temporal tienen más relevancia para la predicción que los valores más antiguos. De este modo, la predicciones se hacen asignando pesos exponencial decrecientes a medida que se avanza el tiempo. Los métodos que se estudian en este trabajo son el método es alisado exponencial simple, el método lineal de Holt y el método del Holt-Winters, aunque existen muchos más.

Al igual que en todos los campos de la ciencia, el análisis de series temporales también ha evolucionado y nuevos modelos han sido creados. Muchos de estos nuevo métodos han surgido para compensar las limitaciones que presentan los métodos clásicos, que no son capaces de modelar los datos actuales, debido a su colosal dimensión o a otros factores. En este trabajo se estudia dos métodos que surgen recientemente.

En 2017, Facebook lanzó al público una algoritmo muy potente llamado Prophet. Prophet tiene la ventaja de ser capaz de trabajar con datos de gran tamaño y modelar diferentes tipos de estacionalidad, lo cual es un punto débil en los métodos clásicos. Además, este algoritmo incorpora un componente muy interesante, los efectos de los días festivos, ya que hay ciertos días del año que pueden afectar a negocios y empresas. El objetivo de este trabajo es estudiar los componentes que se esconden detrás de este algoritmo e investigar cómo funciona, además de usarlo en un caso práctico.

El último método que se estudia son los modelos autorregresivos basados árboles de decisión. A partir de los árboles de decisión es posible crear otros nuevos algoritmos de predicción mediante la agrupación de árboles. En este trabajo se va a estudiar, métodos como *Random Forest* o *XGBoost* modificados para poder predecir series temporales.

Por último, se compara la actuación de todos los métodos aplicados a un mismo conjunto de datos, viendo las limitaciones y las ventajas de cada uno.

1.1. Sobre el dataset

El conjunto de datos "Electricity consumption UK 2009-2023" es un dataset formado por mediciones de la demanda eléctrica en Reino Unido desde enero de 2009 hasta abril de 2023. Se trata de un conjunto de datos público disponible en Kaggle (https://www.kaggle.com/datasets/albertovidalrod/electricity-consumption-uk-20092022). Los datos han sido tomado de forma diaria en intervalos de 30 minutos, siendo 48 observaciones al día, es decir en total hay 250800 observaciones. La Tabla 1 muestra cómo están dispuestos los datos.

A la hora de trabajar con los datos solo se ha tomado la variable que representa la demanda elétrica, aunque el dataset es muy completo, para el análisis de series temporales es suficiente utilizar solo una variable, además de su identificador de fecha y hora.

A la hora de trabajar con los datos se ha dividido en dos partes. La primera es el dataset completo con las 48 observaciones al día. El objetivo de este proyecto es encontrar modelos capaces de ajustar los estos datos, sin embargo la gran mayoría de los métodos están limitados y el tiempo de ejecución es muy alto si se trabaja con los datos completos.

Date	Electricity Demand
01-01-2009 00:00	38704
01-01-2009 00:30	38965
⋮	⋮
25-04-2023 23:00	26244
25-04-2023 23:30	25407

Tabla 1: *Primeras y últimas observaciones del dataset*

Por esta razón, se ha resumido los datos, sumando los valores de las 48 observaciones, obteniendo observaciones diarias.

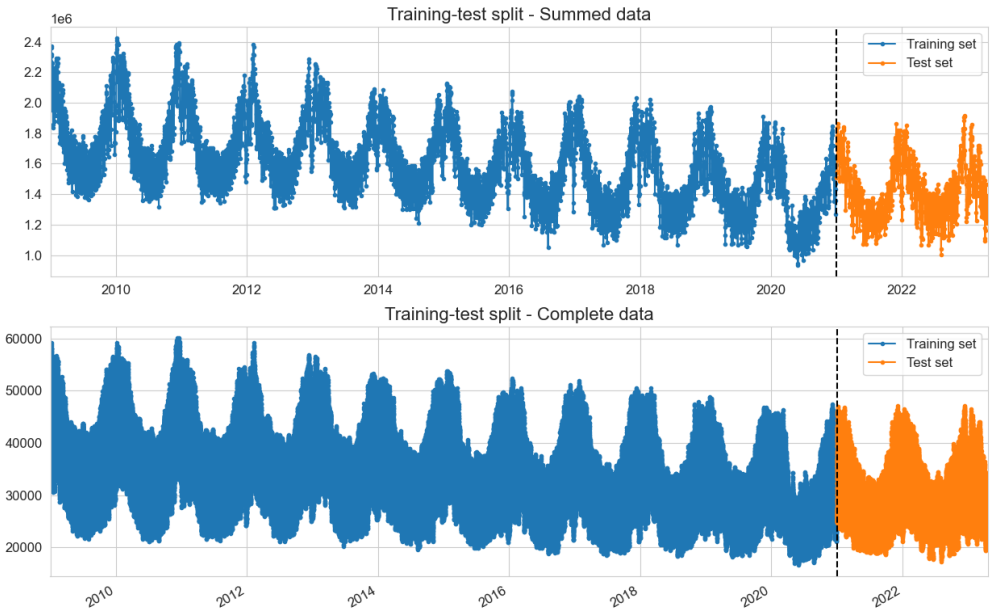

Figura 1: *Separación en conjuntos train y test para los datos sumados y completos, respectivamente*

Para aplicar los modelos se ha dividido el conjuntos en dos partes: conjunto de entranamiento (train) y conjunto de testeo (test). El conjunto train consiste en los datos desde 2009 hasta 2020, inclusive, y los datos del conjunto de test son todas la observaciones en adelande del 1 de enero de 2021. La Figura 1 muestra la separación de los datos resumidos y los datos completos.

En la figura se ve que hay patrones que se repiten anualmente, pero también hay patrones que se repiten mensualmente y semanalmente, y en el caso del dataset completo,

diariamente. Se puede la estacionalidad diaria en la Figura 2.

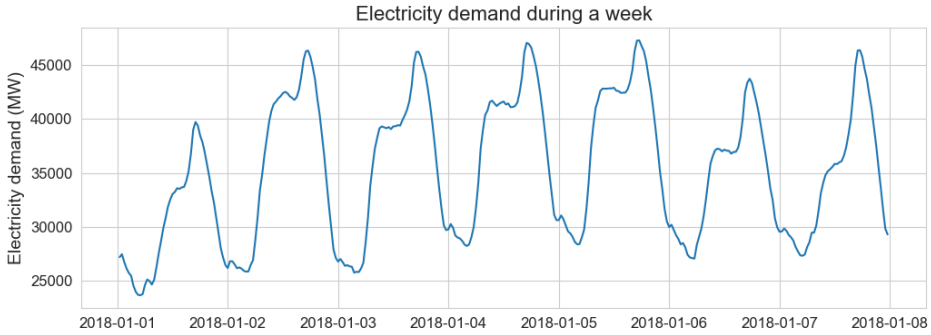

Figura 2: *Estacionalidad diaria de los datos*

Asimismo, la estacionalidad no es el único factor a tener en cuenta a la hora de modelar series temporales. En el caso del algoritmo Prophet de Facebook, también tiene en cuenta los efectos de los días festivos y las vacaciones. Hay fechas señadalas que pueden afectar a los negocios o empresas que no son un día fijo del año, como el día de la madre o el *Black Friday*.

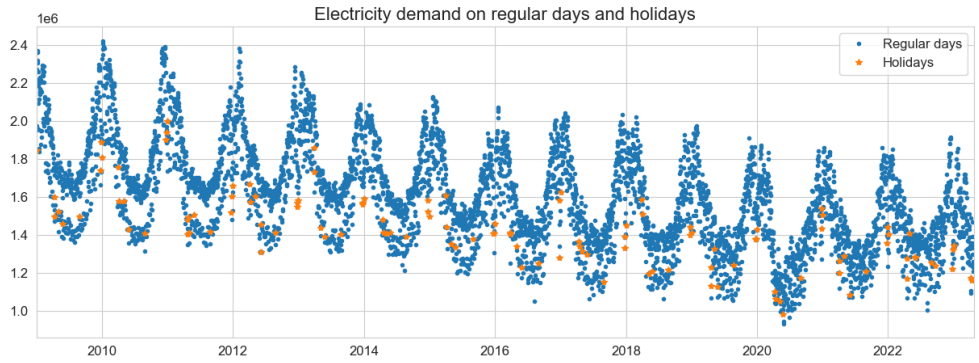

Figura 3: *Días festivos en los datos diarios*

Esta clase de eventos pueden modificar los valores que toma una serie temporal. La Figura 3 muestra los días festivos que contiene el dataset, mientras que la Figura 4 es una comparación de la demanda eléctrica de los días festivos y los días regulares. Se puede observar que para los días festivos, en general, la demanda eléctrica disminuye.

A la hora de evaluar y comparar los modelos se ha utlizado el error absoluto medio

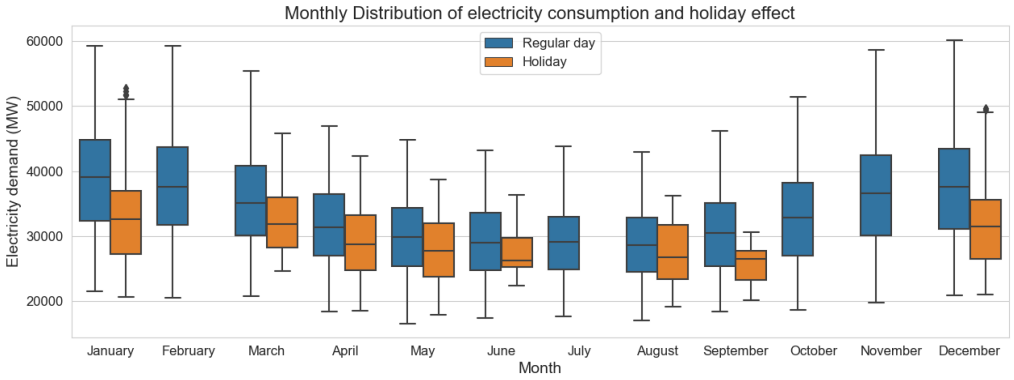

Figura 4: *Comparación de la demanda eléctrica para días regulares y festivos*

porcentual (MAPE) y el error cuadrático medio (MSE). Las fórmulas de estos errores son

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{y_i} \cdot 100,$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2,$$

donde y_i son los valores reales e $\hat{y}_i$ son los valores pronosticados para $i = 1, \ldots, n$, $n \in \mathbb{N}$. En este proyecto, los error se caclulan a partir de los datos del conjunto test y los valores pronosticados por los modelos.

Todos estos efectos se estudian a lo largo del trabajo y en todo momento se pretende comprender cómo afectan estos componentes y encontrar los modelos óptimos capaces de ajustarse correctamente a los datos. Todos los pasos y código empleado para realizar los modelos y gráficos del trabajo se encuentra en GitHub (https://github.com/MarinaPenalver/TFG).

2. Series temporales

Una serie temporal es un conjunto de observaciones x_t, cada una registrada en un tiempo específico t. Una serie temporal con tiempo discreto es aquella en la que el conjunto de tiempo en los que se realizan las observaciones, T_0, es un conjunto discreto. Si los datos son medidos en intervalos fijados, por ejemplo mensual o anualmente, entonces es una serie temporal de tiempo discreto. En la Figura 5 se muestra un gráfico de los datos, que es una serie temporal de tiempo discreto.

Este trabajo se centra en las series temporales de tiempo discreto, sin embargo también existen las de tiempo continuo. Las series temporales de tiempo continuo se obtienen cuando las observaciones son registradas de forma continua a lo largo del tiempo, por ejemplo, cuando $T_0 = [0, 1]$. En este caso la notación pasa a ser $X(t)$, para así poder especificar que las observaciones son registradas de forma continua.

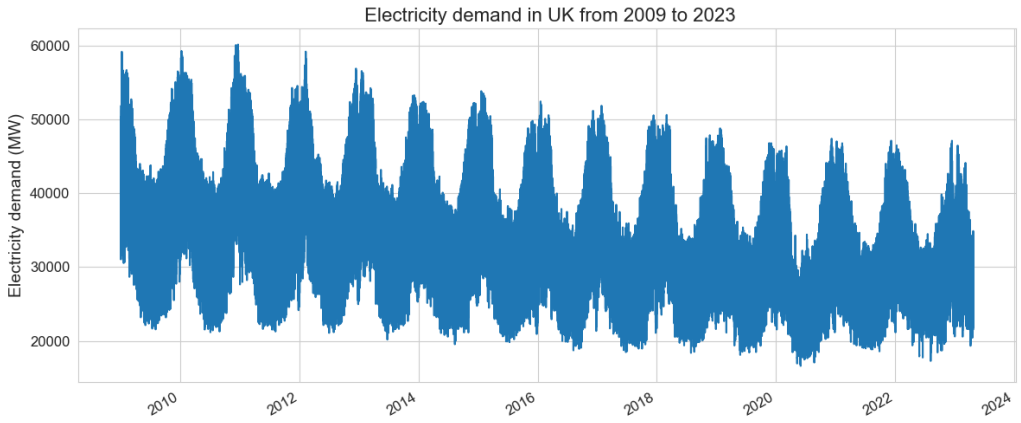

Figura 5: *Serie temporal con tiempo discreto*

Se supone que cada observación x_t es una realización de cierta variables aleatoria X_t. Una serie temporal $\{x_t,\, t \in T_0\}$ es una realización de una familia de variables aleatorias $\{X_t,\, t \in T_0\}$. Estas consideraciones sugieren modelar los datos como una realización (o parte de una realización) de un proceso estocástico $\{X_t,\, t \in T\}$ con $T_0 \subseteq T$.

Definición 2.1 *Un proceso estocástico es una familia de variables aleatorias* $\{X_t,\, t \in T\}$ *definidas en el espacio de probabilidad* $(\Omega, \mathcal{F}, P)$*, donde*

1. Ω *es el espacio muestral,*
2. $\mathcal{F} \subset \mathcal{P}(\Omega)$ *es una* σ*-álgebra y*

3. *P es la probabilidad, $P : \mathcal{F} \longrightarrow [0, 1]$.*

En el análisis de series temporales el conjunto de índices T es un conjunto de puntos, normalmente $\{0, \pm1, \pm2, \dots, \}$, $\{1, 2, 3, \dots\}$, $[0, \infty)$ o $(-\infty, \infty)$. Aunque también se puede definir procesos estocásticos en los cuales T no es un subconjunto de $\mathbb{R}$.

Observación 2.2 *El término serie temporal se usa frecuentemente para referirse tanto a los datos como al proceso del cual es una realización.*

2.1. Modelos estacionarios y estrictamente estacionarios

A la hora de estudiar un número finito de variables, lo más común es usar la matriz de covarianzas para obtener información sobre la dependencia entre ellas. En el caso de las series temporales, es necesario extender este concepto para tratar con conjuntos infinitos de variables aleatorias. Esta extensión la proporciona la función de autocovarianzas.

Definición 2.3 *Si $\{X_t,\, t \in T\}$ es un proceso estocástico tal que $E(X_t^2) < \infty$ para todo $t \in T$, entonces la función de autocovarianzas $\gamma_X(\cdot, \cdot)$ de $\{X_t\}$ se define como*

$$\gamma(r, s) = Cov(X_r, X_s) = E[(X_r - E[X_r])(X_s - E[X_s])], \quad r, s \in T. \tag{1}$$

Definición 2.4 *Si $\{X_t,\, t \in T\}$ es un proceso estocástico tal que $E(X_t^2) < \infty$, entontes*

1. *La función de autocorrelación simple (fas) se define como*

$$\rho(r, s) = \frac{Cov(X_r, X_s)}{\sqrt{Var(X_r)(Var_s)}} = \frac{\gamma(r, s)}{\sqrt{\gamma(r, r)\gamma(s, s)}}. \tag{2}$$

2. *La función de autocorrelación parcial (fap) se define como*

$$\alpha(r, s) = Corr(X_r, X_s \mid X_{r+1}, X_{r+2}, \dots, X_{s-1}) \quad \forall r < s. \tag{3}$$

Definición 2.5 *Una serie temporal $\{X_t,\, t \in \mathbb{Z}\}$ se dice que es estacionaria si*

1. *$E[X_t^2] < \infty \ \forall t \in \mathbb{Z}$*

2. *$E[X_t] = \mu \ \forall t \in \mathbb{Z}$*

3. *$\gamma(r, s) = \gamma(r + t, s + t) \ \forall r, s, t \in \mathbb{Z}$.*

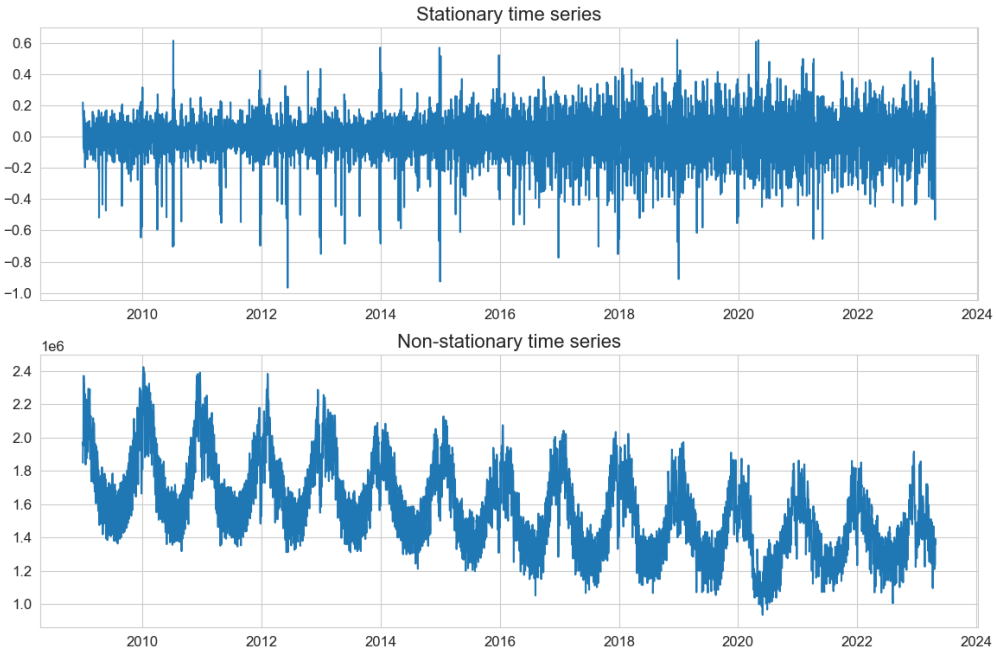

Figura 6: *Ejemplos de serie estacionaria y no estacionaria, respectivamente.*

Esta definición de estacionaridad es referida habitualmete como estacionaridad débil. En este trabajo el término de estacionaridad se referirá siempre a la Definición 2.5.

La Figura 2 muestra dos ejemplos de series temporales: una estacionaria y otra no estacionaria. En el primer gráfico se observa que la media es cero y la varianza es constate, mientras que, en la segunda gráfica se ve que a lo largo de los años la media disminuye y hay patrones de repetición, es decir la media y la varianza no son constantes.

Observación 2.6 *Si* $\{X_t,\, t \in \mathbb{Z}\}$ *es estacionaria entonces* $\gamma_X(r,s) = \gamma_X(r-s,0)$ *para todo* $r, s \in \mathbb{Z}$. *Por tanto, es conveniente redifinir la función de autocovarianzas de un proceso estacionario como la función de una sola variable,*

$$\gamma_X(h) = \gamma_X(h,0) = Cov(X_{t+h}, X_t) \quad \textit{para todo } t, h \in \mathbb{Z}.$$

La función $\gamma_X(\cdot)$ *hace referencia a la función de autocovarianzas de* $\{X_t\}$ *y* $\gamma_X(h)$ *a su valor para el retardo* h. *La función de autcorrelación simple de* $\{X_t\}$ *se define de forma análoga como función de un solo valor para el retardo* h,

$$\rho_X(h) := \frac{\gamma_X(h)}{\gamma_X(0)} = Corr(X_{t+h}, X_t) \quad \forall t, h \in \mathbb{Z}.$$

La definición de estacionaridad se ha dado para el caso en que $T = \mathbb{Z}$. No es difícil definir la estacionaridad usando un conjunto de índices más general, pero en este caso, no es necesario extender la definición. Si se quiere modelar un conjunto de datos $\{x_t,\, t \in T \subset \mathbb{Z}\}$ como una realización de un proceso estacionario, se puede considerar como parte de una realización de un proceso estacionario $\{X_t,\, t \in \mathbb{Z}\}$.

Definición 2.7 (Estacionaridad estricta) *Una serie temporal $\{X_t,\, t \in \mathbb{Z}\}$ se dice estrictamente estacionaria si las distribuciones conjuntas $(X_{t_1}, \ldots, X_{t_k})$ y $(X_{t_1+h}, \ldots, X_{t_k+h})$ son la misma para todos los enteros positivos k y para todo $t_1, \ldots, t_k \in \mathbb{Z}$.*

Si $\{X_t\}$ es estrictamente estacionaria, tomando $k = 1$ en la Definición 2.7 se tienen que X_t tiene las misma distribución para todo $t \in \mathbb{Z}$. En particular, si $E[X_t^2] < \infty$, se tiene que $E[X_t]$ y $Var[X_t]$ son constantes. Además, para $k = 2$ se tiene que X_{t+h} y X_t tienen la misma distribución conjunta para todo $h \in \mathbb{Z}$. Por tanto, la estacionaridad estricta implica estacionaridad (débil).

2.2. Descomposición de una serie temporal

En el método de descomosición clásica, los datos se generan como la suma de tres efectos,

$$X_t = \mu_t + S_t + Y_t, \tag{4}$$

donde μ_t es el *componente de tendencia*, S_t es conocido como el *componente estacional* y, por último, Y_t es el componente puramente aleatorio o *innovación*. En el caso de no presentar componente estacional, la descomposición se reduce a

$$X_t = \mu_t + Y_t. \tag{5}$$

En la Figura 7 se muestra la decomposición aditiva de los datos. Habitualmente, el nivel μ_t se modela mediante un polinomio del tiempo de orden menor o igual a dos, y la estacionalidad como una función periódica, que verifica la condición

$$S_t = S_{t-s},$$

donde s es el periodo de la función. Una serie mensual con estacionalidad anual tiene periodo $s = 12$ meses, ya que se supone que los *coeficientes estacionales*, S_t, se repiten cada 12 observaciones y una serie diaria con estacionalidad semanal tiene $s = 7$ días.

La innovación es un componente aleatorio que recoge todos los demás efectos que actúan sobre la serie. Se supone que las variables aleatorias Y_t tienen una estructura estable

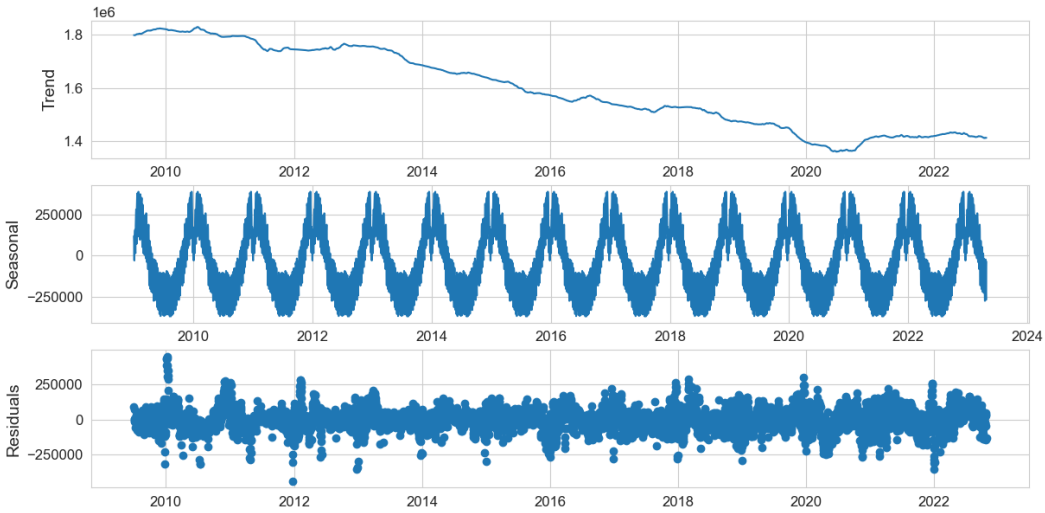

Figura 7: *Descomposición aditiva de los datos*

a lo largo del tiempo: media cero, varianza constante y distribución normal. Además, las observaciones correspondientes a dos periodos distintos de tiempo son independientes, es decir, conocer el valor Y_t no proporciona ninguna información sobre el posible valor de Y_{t+1}.

Una alternativa la descomposición aditiva es la descomposición multiplicativa, que consta de los mismos componentes, y también es muy utlizada,

$$X_t = \mu_t \times S_t \times Y_t. \tag{6}$$

3. Procesos autoregresivos y de media móvil

En esta sección se estudia una clase muy importante de series temporales, definida en términos de ecuaciones en diferencias lineales con coeficientes constantes. La imposición de esta estructura adicional define una familia de procesos estacionarios, los llamados procesos de media móvil autoregresivos o ARMA. Antes de estudiar dichos procesos, es necesario introducir un nuevo tipo de proceso, el proceso de ruido blanco.

Definición 3.1 *El proceso* $\{Y_t\}$ *se conoce como proceso de ruido blanco con media* 0 *y varianza* σ^2*,*

$$\{Y_t\} \sim WN(0, \sigma^2), \tag{7}$$

si $\{Y_t\}$ *tiene media* 0 *y función de autocovarianza*

$$\gamma(h) = \begin{cases} \sigma^2 & si\ h = 0 \\ 0 & si\ h \neq 0 \end{cases} \tag{8}$$

Si las variables aleatoria Y_t *son independiente e identicamente distribuidas con media* 0 *y varianza* σ^2*, entonces se conocen como*

$$\{Y_t\} \sim IID(0, \sigma^2). \tag{9}$$

Los procesos de ruido blanco permiten generar una clase muy amplia de procesos estacionarios gracias a ecuaciones en diferencias lineales.

Definición 3.2 *El proceso* $\{X_t,\ t \in T\}$ *se dice* $ARMA(p, q)$ *si es estacionario y para todo* t*,*

$$X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} = Y_t + \theta_1 Y_{t-1} + \cdots + \theta_q Y_{t-q}, \tag{10}$$

donde $\{Y_t\} \sim WN(0, \sigma^2)$*. Se dice que* $\{X_t\}$ *es un proceso* $ARMA(p, q)$ *con media* μ *si* $\{X_t - \mu\}$ *es un proceso* $ARMA(p, q)$*.*

La ecuación (10) puede escribirse de forma más compacta, introduciendo la notación del operador de retardo.

Definición 3.3 *Se llama operador de retardo a* B*, definido por*

$$BX_t = X_{t-1}. \tag{11}$$

Además cumple las propiedades siguientes:

1. $B\mu = \mu$*, para* μ *constante.*
2. $BaX_t = aBx_t = aX_{t-1}$*, para* a *constante.*

3. Es lineal, es decir, $B(af(t) + bg(t)) = af(t-1) + bg(t-1)$.

4. $B^k X_t = \underbrace{B \cdots B}_{k} X_t = X_{t-k}$.

Utilizando la notación del operador de retardo, la ecuación (10) puede escribirse como

$$\phi(B)X_t = \theta(B)Y_t, \tag{12}$$

donde $\phi(B)$ y $\theta(B)$ son polinomios de grado p y q en el operador de retardo,

$$\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p, \tag{13}$$

$$\theta(B) = 1 + \theta_1 B + \cdots + \theta_q B^q. \tag{14}$$

A $\phi(B)$ se le conoce como el operador del proceso autoregresivo u operador AR y a $\theta(B)$ se le conoce como el operador del proceso de media móvil u operador MA. El proceso ARMA(p, q) es estacionario si, y solo si, el módulo de las raíces de $\phi_p(B) = 0$ son mayores a 1.

Observación 3.4 *Para cualquien función de autocovarianza* $\gamma(\cdot)$ *tal que* $\lim_{h\to\infty} \gamma(h) = 0$*, y para todo entero* $k > 0$*, es posible encontrar un proceso ARMA con función de autocovarianza* $\gamma_X(h) = \gamma(h)$ $h = 0, 1 \ldots, k$.

Por razones como esta, la familia de los procesos ARMA juega un papel muy importante en la modelación de series temporales. Además, su estrucutura lineal conduce a un teoria simple de predicciones lineales.

Ejemplo 3.5 (Proceso MA(q)) *Si* $\phi(B) \equiv 1$*, entonces*

$$X_t = \theta(B)Y_t, \tag{15}$$

y a este proceso se le conoce como proceso de media móvil de orden q *o MA*(q) *por sus siglas en inglés,* moving average*. Se tiene que* $\{X_t\}$ *es estacionaria ya que, tomando* $\theta_0 = 1$ *y* $\theta_j = 0$ *para* $j > q$*, se tiene que,* $\forall h > 0$

$$E[X_t] = \sum_{j=0}^{q} \theta_j E[Y_{t-j}] = 0 \quad y$$

$$\gamma_X(h) = Cov(X_{t+h}, X_t) = \begin{cases} \sigma^2 \sum_{j=h}^{q} \theta_j \theta_{j-h} & si\ h \leq q, \\ 0 & si\ h > q. \end{cases}$$

Ejemplo 3.6 (Proceso AR(p)) *Si* $\theta(B) \equiv 1$*, entonces*

$$\phi(B)X_t = Y_t \tag{16}$$

y a este proceso se le conoce como proceso autorregresivo de orden p *o AR*(p)*. Este proceso es estacionario solo si las raíces de* $\phi(B) = 0$ *están fuera del cículo unidad.*

3.1. Procesos ARIMA y SARIMA

Se puede definir nuevos procesos no estacionarios a partir de la Definición 3.2. Para ello, se incorpora raíces unitarias al proceso ARMA a través de un operador.

Definición 3.7 *El operador de diferenciación, denotado por* ∇*, se define como*

$$\nabla = 1 - B.$$

De igual modo, se define el operador de diferenciación de orden s u operador de diferenciación estacional como $\nabla_s = 1 - B^s$. Con los operadores se puede definir un nuevo proceso.

Definición 3.8 *Se dice que* $\{X_t\}$ *es un proceso integrado de orden* h*,* $I(h)$*, si al diferenciar* $\{X_t\}$ h *veces se obtiene un proceso estacionario.*

Uniendo los procesos estudiados, es decir los ARMA(p, q) con los procesos integrado, se obtiene los procesos de media móvil integrados autorregresivos o ARIMA.

Definición 3.9 *Se dice que* $\{X_t\}$ *es un proceso* $ARIMA(p, d, q)$ *si*

$$\phi(B)(1 - B)^d X_t = \theta(B)Y_t \tag{17}$$

o, con la notación del operador de diferenciación

$$\phi(B)\nabla^d X_t = \theta(B)Y_t, \tag{18}$$

donde $\{Y_t\} \sim WN(0, \sigma^2)$*,* $\theta(B) = 1 + \theta_1 B + \cdots + \theta_q B^q$*,* $\phi(B) = 1 - \phi_1 B - \cdots - \phi_q B^p$ *y las raíces de* $\phi(B) = 0$ *tienen módulo mayor que 1.*

Por la estructura de los procesos ARIMA, se sabe que son no estacionarios, sin embargo se puede definir $W_t = \nabla^d X_t$ que es un proceso ARMA(p, q) y sí es estacionario. Los procesos ARIMA se pueden extender aún más introduciendo el operador de diferenciación estacional.

Definición 3.10 $\{X_t\}$ *es un proceso ARIMA multiplicativo estacional o SARIMA, denotado* $\{X_t\} \sim ARIMA(P, D, Q)_s \times (p, d, q)$*, si*

$$\Phi(B^s)\phi(B)\nabla_s^D \nabla^d X_t = \theta(B)\Theta(B^s)Y_t, \tag{19}$$

donde

- $\{Y_t\} \sim WN(0, \sigma^2)$*,*
- ∇_s^D *representa la diferenciación estacional,*

- ∇^d *representa la diferenciación regular*
- $\Phi(B^s) = 1 - \Phi_1 B^s - \cdots - \Phi_P B^{sP}$ *es el operador AR estacional,*
- $\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$ *es el operador AR,*
- $\Theta(B^s) = 1 + \Theta_1 B^s + \cdots + \Theta_Q B^{sQ}$ *es el operador MA estacional,*
- $\theta(B) = 1 + \theta_1 B + \cdots + \theta_q B^q$ *es el operador MA.*

En la siguiente parte de esta seccción se va a explicar cómo se calculan las predicciones del modelo SARIMA y se va a estudiar los resultados obtenidos de los datos.

Se cononce como $\hat{X}_T(h)$ a un predictor de X_{T+h} obtenido como función de los T valores observados, es decir con origen T y horizonte de la predicción h. El predictor de X_{T+h} con origen T y horizonte de predicción a $\hat{X}_T(h)$. Si la predicción es lineal entonces se puede escribir como

$$\hat{X}_T(h) = \alpha_1 X_T + \alpha_2 X_2 + \cdots + \alpha_T X_1.$$

El predictor queda definido al determinar el valor de las constantes $\alpha_1, \alpha_2, \ldots \alpha_T$. El predictor lineal que minimiza el error cuadrático medio, $\text{MSE}(\hat{X}_T(h)) = E[(\hat{X}_T(h) - X_{T+h})^2 \mid X_1, \ldots, X_T\,]$, es

$$\hat{X}_T(h) = E[X_{T+h} \mid X_1, \ldots, X_T\,]. \tag{20}$$

Supóngase que $\{X_t\} \sim \text{ARIMA}(P, D, Q)_s \times (p, d, q)$, se ha observado $x_1, x_2, \ldots, x_T$ y se pretende calcular la predicción de X_{T+h}, con $h \in \mathbb{N}$. Para $D = 1$, es posible calcular la predicciones de forma sencilla pues se tiene que el predictor linear de (20) cumple la ecuación

$$\Phi_P(B^s)\phi_p(B)\nabla_s\nabla^d\hat{X}_T(h) = 0, \quad h > q + sQ. \tag{21}$$

Defininiendo el operador estacional puro, $S_S(B) = 1 + B + \cdots + B^{s-1}$, la ecuación (21) se puede escribir como

$$\Phi_P(B^s)\phi_p(B)S_s(B)\nabla^{d+1}\hat{X}_T(h) = 0, \quad h > q + sQ. \tag{22}$$

En consecuencia, para $h \geq \max(1, q + sQ + 1 - (d + s + p + sP))$ las predicciones $\hat{X}_T(h)$ son las soluciones de la ecuación en diferencias (22) y se tiene que

$$\hat{X}_T(h) = T_T(h) + S_T(h) + t_T(h), \tag{23}$$

donde

1. $T_T(h)$ es el componente estacional solución de $(1-B)^{d+1}T_T(h) = 0$. $T_T(h)$ es un polinomio de grado d con coeficientes que se adaptan a lo largo del tiempo.

2. $S_T(h)$ es el componenente estacional solución de $S_s(B)S_T(h)$. $S_T(h)$ es una función de periodo s tal que $\sum_{j=1}^{s} S_T(j) = 0$.

3. $t_T(h)$ es el componente transitorio, solución de $\Phi(B^s)\phi(B)t_T(h) = 0$. $t_T(h)$ está determinado por las raíces de los operadores regular y estacional autorregresivos.

La implementación en Python de este método se ha hecho gracias al paquete *statsmodels*, aplicado a los datos diarios, ya que si se tomara el dataset completo el tiempo de ejecución ascendería hasta horas o días. La Figura 8 muestra tanto los datos ajustados como las predicciones.

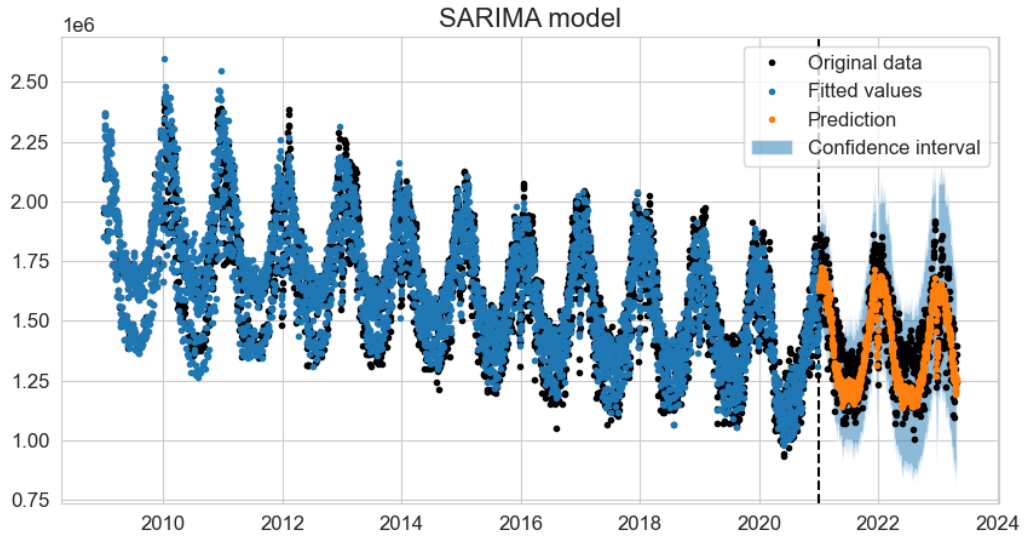

Figura 8: *Modelo ajustado con SARIMA*

En el conjunto de entrenamiento, los datos se ajustan correctamente, especialmente en el primer año de la serie, pero según se avanza en el tiempo, la serie no alcanza los valores más altos y más bajos que provoca la estacionalidad.

Exactamente lo mismo ocurre en los datos de testeo, pero en la Figura 9 se ve cómo la predicción se permanece aproximadamente en la media de todas la observaciones. El intervalo de confianza del 95 % si logra capturar todas la observaciones en su interior.

Además, conforme se avanza en el tiempo, el intervalo de confianza también aumenta su amplitud, siendo cada vez menos preciso.

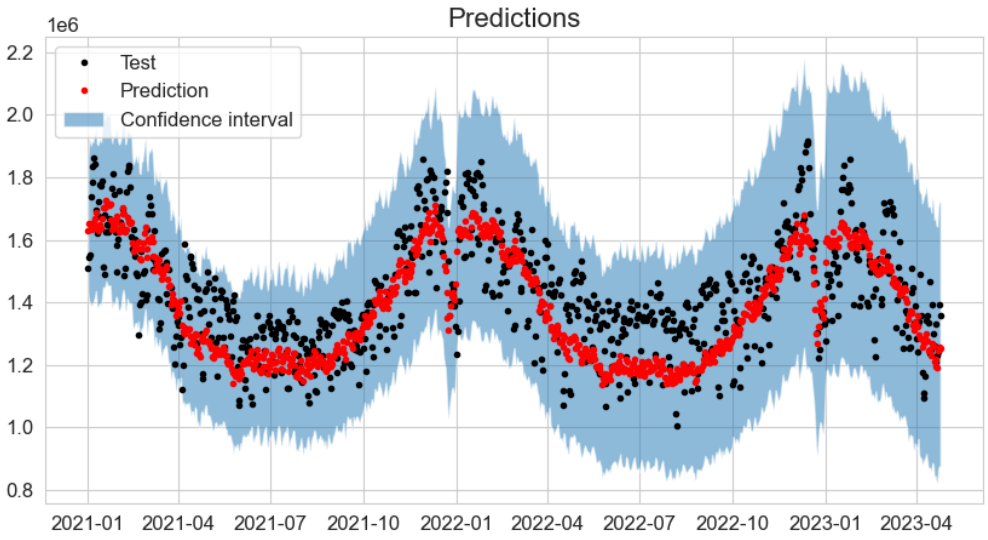

Figura 9: *Predicciones con SARIMA*

De este modelo, cabe destacar que el error absoluto medio porcentual es del 14,31 %, el error cuadrático medio es $1{,}761 \cdot 10^{10}$ y su tiempo de ejecución es 23 minutos. El error porcentual no es significativamente grande, pero el modelo no es capaz de capturar los patrones estacionales que no son anuales, pues en este caso también hay estacionalidad semanal y mensual, y con los modelos SARIMA solo es posible ajustar a un tipo de estacionalidad.

Se ha utlizado la metodología de Box-Jenkins para identificar el modelo SARIMA que se ajusta a los datos. El modelo presentado en la imagen es un proceso $\text{ARIMA}(1,1,2)_{365} \times (1,1,1)$, los valores de los parámetros se encuentran en la Tabla 2. La limitación de este método es su tiempo de ejecución. Al ser el periodo de estacionalidad $s = 365$, la duración de computación del algoritmo es grande, además si se aumenta el número de parámetros de los procesos AR y MA, el tiempo de computación es tan alto que es inviable calcularlo.

ϕ_1	θ_1	θ_2	Φ_1	Θ_1	σ^2
0,0634	−0,2977	−0,6445	0,1278	−0,9998	393,9037

Tabla 2: *Parámetros de modelo SARIMA*

En la medología de Box-Jenkins la idea es proponer diferentes modelos gracias a los gráficos de las funciones de autocorrelación para después escoger el que mejor se ajuste, pero en este caso, no se ha podido computar todos los modelos, por tardar demasiadas horas. Si se trabaja con estacionalidad $s = 7$ o $s = 30$ para datos semanales o mensuales, respectivamente, el algoritmo obtiene la solución en segundos o minutos. No obstante, el objetivo es trabajar con los datos disponibles y estacionalidad que le corresponde, de forma que este modelo no es adecuado en este caso.

En consecuencia, el modelo SARIMA funciona correctamente, pero existen otros modelos con menor coste computacional que podrían ajustar lo datos de forma más precisa, teniendo en cuenta diferentes tipos de estacionalidad de forma simultánea. En las siguientes secciones se presenta algunos de esos métodos.

4. Suavizado exponencial

El suavizado exponencial o alisado exponencial describe una clase de métodos de predicción. De hecho, muchos de los métodos de predicción más exitosos están basandos en el concepto del suavizado exponencial. Hay una gran variedad de métodos basados en esta familia, cada una con la propiedad de que los pronósticos son combinaciones ponderadas de observaciones pasadas, donde las observaciones recientes tienen más peso que las observaciones más alejadas. El nombre "suavizado exponencial" refleja el hecho de que los pesos caen exponencialmente según las observaciones son más lejanas.

En este capítulo se va a estudiar tres de los métodos más reconocidos en la familia de suavizado exponencial: el método de suavizado simple, el método lineal de Holt y el método de Holt-Winters. Todos estos algoritmos están implementados en Python en el paquete *statsmodels*.

4.1. Suavizado exponencial simple

El modelo de suavizado exponencial simple recibe su nombre por ser el más secillo de esta familia. Esta técnica solo se puede aplicar a datos que no presentan tendencia ni patrones estacionales.

El método de *simple exponential smoothing*, resultado del trabajo de Brown en la decada de los 50, toma la predicción del periodo previo y lo ajusta usando el error de predicción.

Si se supone que se dispone de las observaciones hasta $T-1$ y se quiere predecir el próximo valor de la serie temporal X_T, entonces el error de la predicción es $X_T - \hat{X}_T$, siendo $\hat{X}_T$ la predicción. En el método de suavizado exponencial simple la predicción para el siguiente tiempo se define como

$$\hat{X}_{t+1} = \hat{X}_t + \alpha(X_t - \hat{X}_t), \tag{24}$$

donde α es una constante entre 0 y 1.

La nueva predicción es la predicción anterior sumando un ajuste para el error de la última predicción. Cuando α tiene un valor cercano a 1, la nueva predicción tendrá un ajuste significativo por el error en la anterior predicción. Por el contrario, cuando α es cercano a 0, el nuevo pronóstico tendrá un ajuste muy pequeño.

Otra forma de escribir la ecuacion (24) es

$$\hat{X}_{t+1} = \alpha X_t + (1-\alpha)\hat{X}_t. \tag{25}$$

La predicción $\hat{X}_{t+1}$ está basada en la ponderación de la observación más reciente, X_t con un peso de α y la predición más reciente, $\hat{X}_t$ con peso de $1-\alpha$. Las implicaciones del suavizado exponencial se pueden observar más fácilmente si se expande la ecuación (25) sustituyendo $\hat{X}_t$ por sus componentes, como sigue:

$$\begin{aligned}\hat{X}_{t+1} &= \alpha X_t + (1-\alpha)[\alpha X_{t-1} + (1-\alpha)\hat{X}_{t-1}] \\ &= \alpha X_t + \alpha(1-\alpha)X_{t-1} + (1-\alpha)^2\hat{X}_{t-1}.\end{aligned}$$

Si se repite este proceso de sustitución de $\hat{X}_{t-1}$ por sus componentes, $\hat{X}_{t-2}$ por sus componentes, y así, se llega al resultado

$$\begin{aligned}\hat{X}_{t+1} = \alpha X_t + \alpha(1-\alpha)X_{t-1} + \alpha(1-\alpha)^2 X_{t-2} + \alpha(1-\alpha)^3 X_{t-3} \\ + \alpha(1-\alpha)^4 X_{t-4} + \cdots + \alpha(1-\alpha)^{t-1} X_1 + (1-\alpha)^t \hat{X}_1.\end{aligned}$$

Por tanto $\hat{X}_{T+1}$ representa un proceso de media móvil ponderado con todas las observaciones pasadas y los pesos decreciendo exponencialmente, de ahí el nombre de "suavizado exponencial".

El pronóstico para cualquier horizonte h se calcula como

$$\hat{X}_{t+h|t} = \hat{X}_{t+1}, \tag{26}$$

es decir, la predicciones a largo plazo son planas, por esta razón el método de suavizado exponencial simple se utiliza con datos que no presentan ni tendencia ni estacionalidad ni otros patrones.

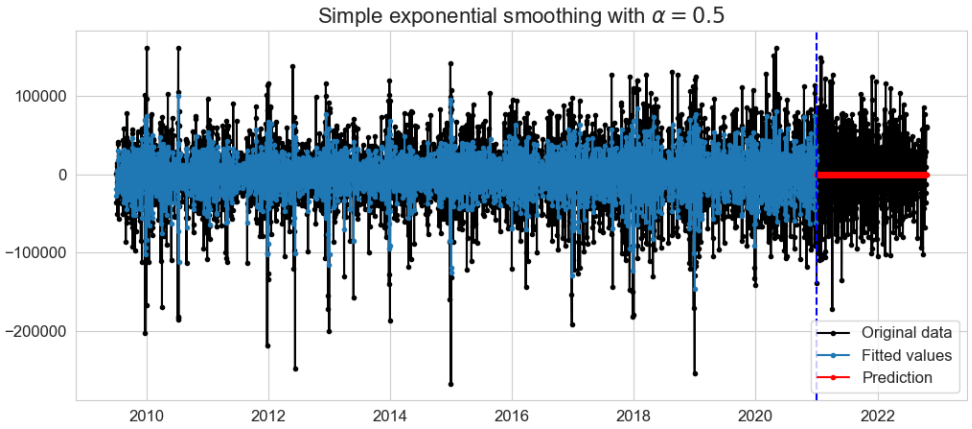

Figura 10: *Valores ajustados y predicción de método de suavizado exponencial simple*

Los datos disponibles presentan tendencia y diferentes patrones de estacionalidad, por tanto, para aplicar este algoritmo, se ha eliminado estos componentes. En la Figura 10 se puede ver cómo el modelo ajusta a los datos y las predicciones son estáticas en el tiempo, con lo que no se ajusta correctamente a los posibles cambios de los datos.

En la Figura 11 se presentan tres modelos, en los dos primeros se fija el valor de α como 0,2 y 0,6, respectivamente, y el tercer modelo busca el valor óptimo. Se observa en el gráfico que todas las predicciones son estáticas y, además, toma un valor aproximado a la media ,0.

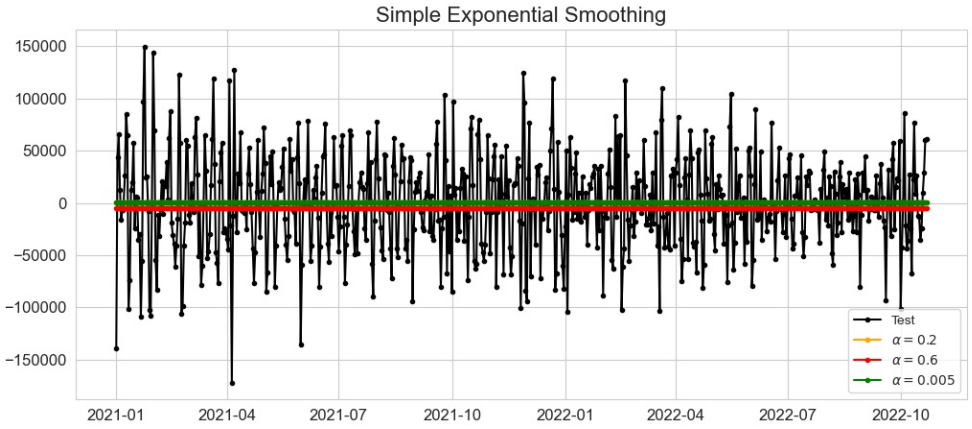

Figura 11: *Modelos del método de suavizado exponencial simple con diferentes parámetros de suavizado*

	Modelo 1	Modelo 2	Modelo 3
α	0,2	0,6	0,005
MAPE	102,09	153,66	99,83
MSE	$2{,}070 \cdot 10^9$	$2{,}095 \cdot 10^9$	$2{,}070 \cdot 10^9$
Tiempo (s)	0,014	0,0107	0,0175

Tabla 3: *Parámetros, tiempo y error del método de suavizado exponencial simple*

En la Tabla 3, se ve que el error absoluto medio porcentual asciende hasta el 150 %. El error alcanza valores tan altos porque los datos son cercanos a cero, es decir, al dividir por dichos valores, el error incrementa rápidamente. Se observa también que el error cuátrico medio se miniza para los valores de $\alpha = 0{,}2$ y para el valor de α optimizado, como cabe esperar. Sin embargo, el ajuste de las predicciones no es lo suficiente preciso.

4.2. Método lineal de Holt

Holt extendió el suavizado exponencial simple al suavizado exponencial lineal para permitir el pronósticos con datos que presentan tendencias. También recibe el nombre de método de alisado exponencial doble. La predicción de método lineal de Holt se basa en utilizar dos constantes α y β^* (con valores entre 0 y 1), y tres ecuaciones

$$\text{Nivel:} \quad l_t = \alpha X_t + (1-\alpha)(l_{t-1} + b_{t-1}), \tag{27}$$

$$\text{Crecimiento:} \quad b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}, \tag{28}$$

$$\text{Predicción:} \quad \hat{X}_{t+h|t} = l_t + b_t h. \tag{29}$$

l_t denota una estimación del nivel de la serie en el instante t y b_t denota una estimación de la pendiente (o crecimiento) de la serie en el instante t.

La ventaja de este modelo es que puede trabajar con datos con tendencia, por ello, se ha transformado los datos eliminando la componente estacional, para así aplicar este método.

La Figura 12 muestra los valores ajustados y la predicción del modelos lineal de Holt para los valores $\alpha = 0{,}5$ y $\beta^* = 0{,}2$. Sin embargo, se observa cómo el valor de la predicción desciende, alejándose de los valores reales.

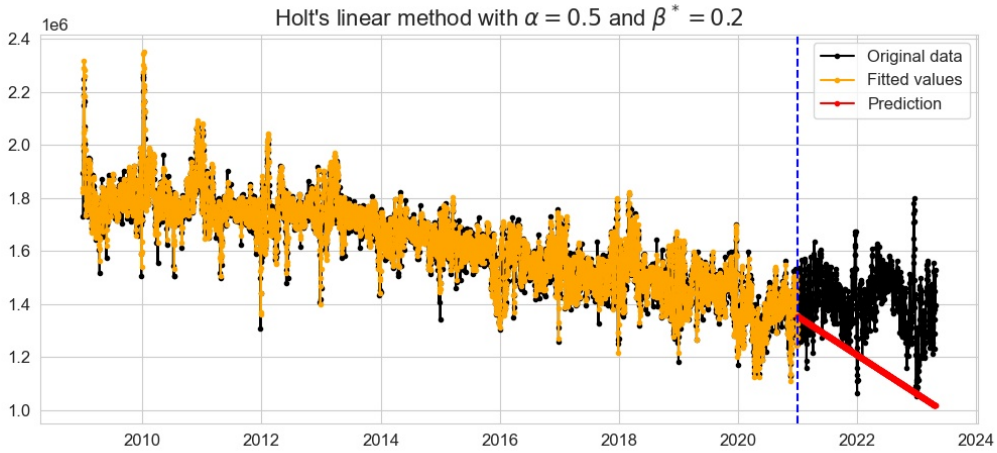

Figura 12: *Valores ajustados y predicción del modelo lineal de Holt*

A continuación se ha probado tres modelos con las opciones disponibles de Python. El primer modelo es el presentado por las ecuaciones (27), (28) y (29). El segundo modelo ajusta la tendencia usando un modelo multiplicativo, en lugar de usar el modelo aditivo

visto. Por último, el tercer modelo presenta el método amortiguado del método de Holt, en el cual se introduce un nuevo parámetro de amortiguamiento, ϕ.

	Aditivo	Multiplicativo	Amortiguado
α	0,85357	0,85357	0,877143
β	0,021886	0,02188	0,0001
ϕ	-	-	0,99
MAPE	23,068	10,865	7,027
MSE	$1{,}390 \cdot 10^{11}$	$3{,}569 \cdot 10^{10}$	$1{,}509 \cdot 10^{10}$
Tiempo (s)	0,187	0,201	0,161

Tabla 4: *Parámetros, error y tiempo del método de Holt*

Todos los modelos optimizan los valores de α y β^*, se puede ver los parámetros escogido en la Tabla 4. Los modelos se ejecutan en menos de un segundo, por tanto el tiempo de computación no es un problema en este caso. Sin embargo, en la Figura 13 se observa cómo el modelo aditivo y multiplicativo se aleja de los valores reales del test y no predice correctamente. El modelo amortiguado es el único que ofrece una predicción coherente con los datos presentados, aunque las prediciones son estáticas. La Tabla 4 muestra que los errores MAPE y MSE se minimizan en el modelo amortiguado con valores de 7 % y $1{,}509 \cdot 10^{10}$, respectivamente.

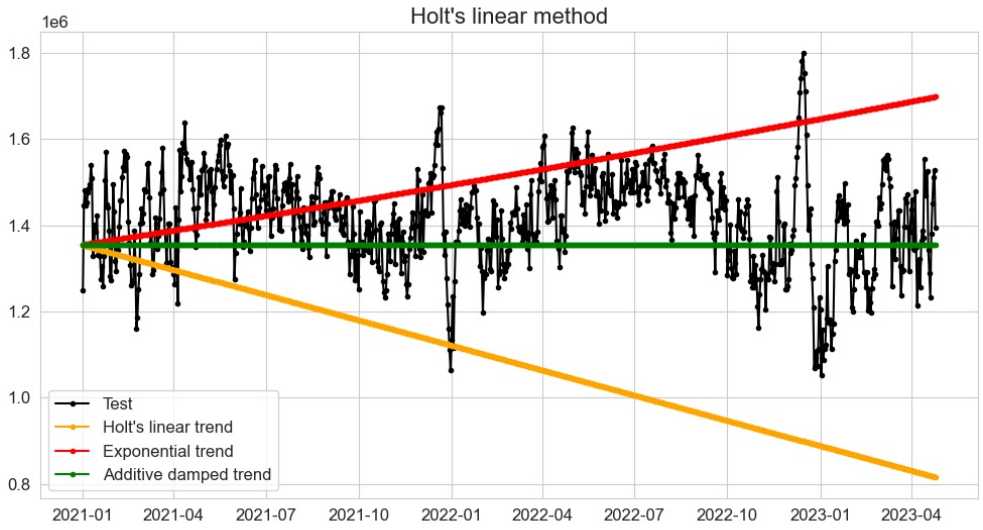

Figura 13: *Predicciones de tres modelos con el método lineal de Holt*

4.3. Algoritmo de Holt-Winthers

Holt y Winters extendieron el método de Holt para incorporar el componente estacional. El método de Holt-Winters consta de tres ecuaciones de suavizado: una para el nivel l_t, otra para la tendencia b_t y la última para el componente estacional s_t. A cada ecuación le corresponde un parámetro de suavizado α, β^* y γ, respectivamente.

$$\begin{aligned} \text{Nivel:} \quad & l_t = \alpha(X_t - s_{t-s}) + (1-\alpha)(l_{t-1} + b_{t-1}), \\ \text{Crecimiento:} \quad & b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}, \\ \text{Estacionalidad:} \quad & s_t = \gamma(X_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-s}, \\ \text{Predicción:} \quad & \hat{X}_{t+h|t} = l_t + b_t h + s_{t+h-s(k+1)}, \end{aligned}$$

donde s es el periodo de estacionalidad y $k = \lfloor (h-1)/m \rfloor$, lo que asegura que las estimaciones de los índices estacionales utilizados para la predicción provienen del último año de la muestra. También recibe el nombre de método de alisado exponencial triple.

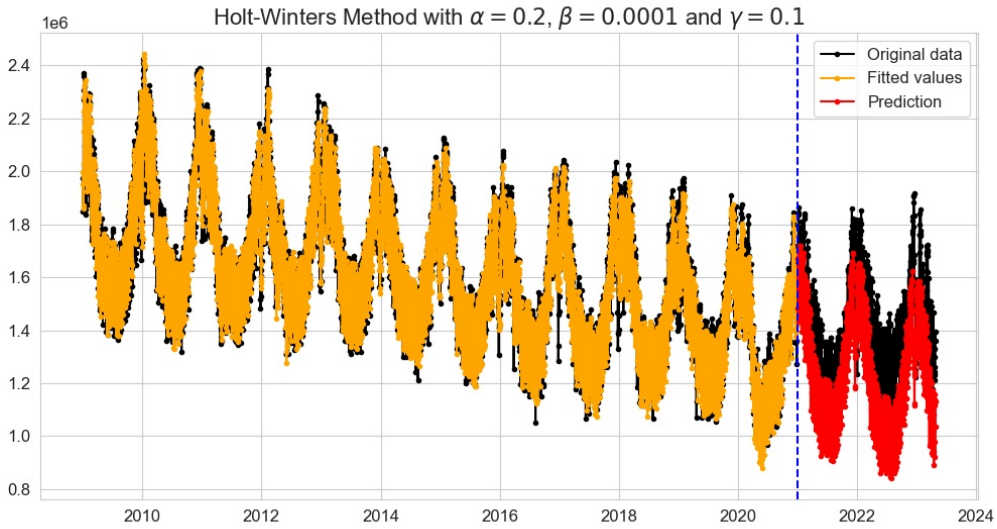

Figura 14: *Valores ajustados y predicción del modelo lineal de Holt-Winters*

A continuación, se van a presentar modelos del algorimo de Holt-Winters aplicados a los datos medidos diaramente y a los datos completos. La Figura 14 muestra el modelo es capaz de ajustarse a los datos del entrenenamiento, sin embargo para en los datos de testeo, la predicciones parecen desplazadas hacia bajo. El modelo esperaba que el nivel de la serie sigiuera bajando, pero a partir de 2022 se aprecia una ligera subida, que el modelo no ha sido capaz de predecir.

	Aditivo	Multiplicativo	Ad. amortiguado	Mul. Amortiguado
α	0,6060	0,995	0,6060	0,995
β	0,0001	0,0001	0,0001	0,0001
γ	0,3939	0,005	0,3939	0,005
ϕ	-	-	0,99	0,99
MAPE	24,13	17,08	21,92	15,75
MSE	$1{,}013 \cdot 10^{11}$	$4{,}105 \cdot 10^{10}$	$7{,}772 \cdot 10^{10}$	$3{,}074 \cdot 10^{10}$
Tiempo (s)	2,705	4,272	2,903	4,352

Tabla 5: *Parámetros, error y tiempo del método de Holt-WInters*

Al igual que con los otros modelos, se ha implementado las opciones disponibles, ajustado hasta cuatro modelos distintos que aplican el algoritmo de forma aditiva y multiplicativa y a su vez añadiendo o no el parámetro de amortiguamiento. En todos los métodos el algoritmo busca los parámetros óptimos. Los parámetros y resultados se pueden observar en la Tabla 5. Se ve que el modelo que minimiza el MAPE y el MSE es el multiplicativo amortiguado, con un valores de 15 % y $3{,}07 \cdot 10^{10}$, respectivamente.

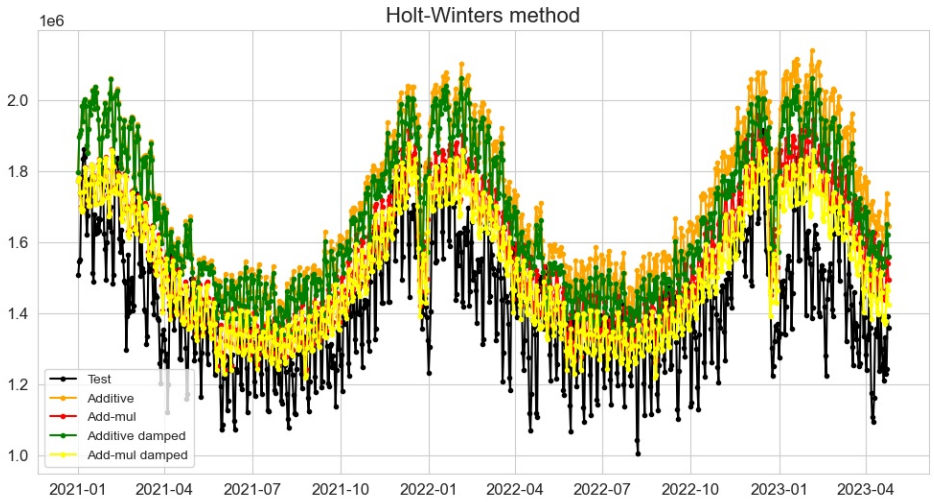

Figura 15: *Predicciones de los cuatro modelos del método de Holt-Winters*

Las predicciones en este caso capturan correctamente el componente estacional que se repite cada año, como se ve en la Figura 15. Sin embargo, la amplitud de las oscilaciones del conjunto test es mayor que en la predicciones, se ve cómo las predicciones no han sabido capturar los valores más bajos de la serie. No obstante, de forma general, la predicciones

si son buenas con un error del 15 %. Además, en este caso, el tiempo de computación de varios segundos, pero si se aumenta el número de datos y el perido de estacionalidad, este algoritmo puede estar compilando durante horas.

También se ha aplicado este algoritmo al dataset completo, pero como el tiempo de computación era demasiado alto, se ha reducido con conjunto de entramiento tomando solamente el último año, 2020. De este modo, se ha intentado modelar la estacionalidad mensual que presentan los datos. En la Figura 16 se observa cómo el algoritmo sí se adapta a los datos del conjunto de entrenamiento, sin embargo su actuación en el conjunto de test es muy deficiente.

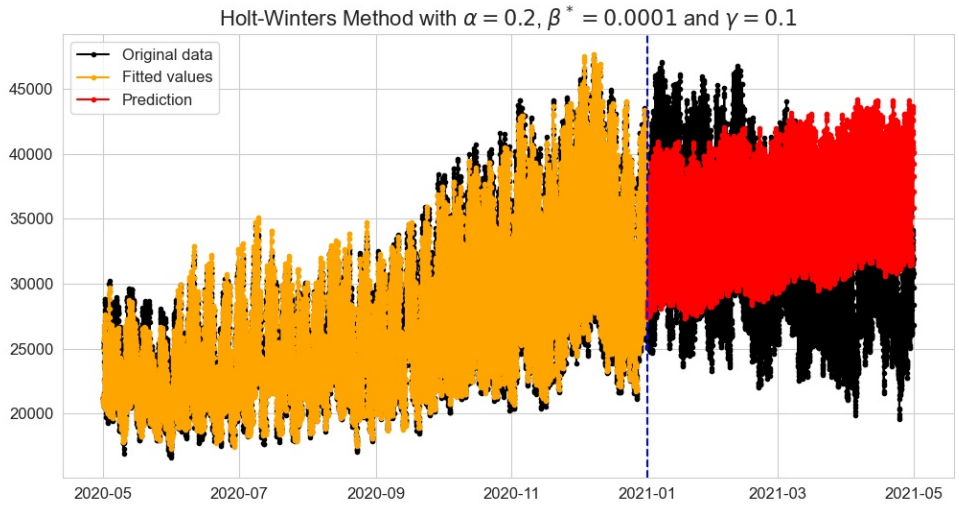

Figura 16: *Predicciones del método de Holt-Winters para los datos compeltos*

En cuanto a las predicciones, es método no modela correctamente la estacionalidad presente en los datos. Además, en cuanto a la tendencia, se ve que en el train hay cierto crecimiento que provoca que en las predicciones también también asciendan con el paso del tiempo, sin embargo, se ve que los valores reales descienden. También cabe resaltar la amplitud de las oscilaciones de la predicción, que son menores a los valores reales. De este modo, se ve que no logra ajustar a los valores más extremos que presenta la serie. En este caso el tiempo de computación es y el MAPE es 23,63 %.

El algoritmo funciona correctamente modelando los datos diarios, sin embargo aún no se ha encontrado un método de series temporales clásico o de alisado exponencial que permita ajustar a los datos completos. Por ello, en las próximas secciones se presentarán nuevos métodos que conseguirán cumplir dicho objetivo.

5. Prophet

Prophet es un software de código abierto que fue desarrollado internamente en *Facebook* (ahora conocido como *Meta*) por Sean J. Taylor y Ben Lethan, para afrontar dos de los problemas más comunes en las metodologias de predicción:

1. Las herramientas de predicción más automaticas disponibles tienden a ser inflexibles e incapaces de ajustarse a suposiciones adicionales.

2. La herramientas de predicción más robustas requieres un analisita especializado en la ciencia de datos.

Facebook experimentó demasiada demanda de pronósticos de alta calidad, la cual los analistas no podía proporcionar. En 2017, Facebook lanzó Prophet al público como software de código abierto.

El modelo de Prophet consiste en usar un modelo de series temporales descomponible que tiene en cuenta tres factores importates: la tendencia, la estacionalidad y los días festivos. Estos componentes son combinados en la ecuación

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t. \tag{30}$$

El valor de y pronosticado por el modelo en el momento t viene dado por la función $y(t)$. Esta función se descompone en cuatro sumandos:

- $g(t)$ se corresponede con la componente de crecimiento o la tendencia general, que modela los cambios no periódicos.

- $s(t)$ representa la componente estacional, que es la suma de todos los componentes periódicos.

- $h(t)$ representa los efectos de los días festivos, que ocurre en calendarios irregulares durante uno o más días.

- ε_t es el término del error, que engloba todos los demás cambios que no ajustan los demás componentes del modelo.

La combiandción de estos componentes es todo lo que Prophet requiere para construir las predicciones. Para comprender cómo funciona Prophet es necesario desglosar y estudiar cada uno de los componentes.

5.1. Modelos de tendencia

Con el fin de ajustar el componente de tendencias se implementan dos tipos de modelo: un modelo lineal y un modelo logístico. Para pronosticar problemas que no muestran un crecimiento de saturación, el modelo lineal se escribe como

$$g(t) = (k + \mathbf{a}(t)^T\boldsymbol{\delta})t + (m + \mathbf{a}(t)^T\boldsymbol{\gamma}). \tag{31}$$

La variable k es la tasa de crecimiento. Este es un modelo lineal definido a trozos, es decir, la pendiente cambia como una función de t, por ello, a la pendiente k se le añade $\mathbf{a}(t)^T\boldsymbol{\delta}$.

Se supone que existen S puntos de cambio en los tiempos s_j, $j = 1, \ldots, S$, donde la tasa de crecimiento puede cambiar. Se define el vector de ajustes de tasa $\boldsymbol{\delta} \in \mathbb{R}^S$, donde δ_j es el cambio en la tasa que ocurre en el tiempo s_j. De esta forma, para cualquier tiempo t la tasa es la suma de una tasa base k con todos los ajustes que ocurren hasta este tiempo, es decir, $k + \sum_{j:t>s_j} \delta_j$. Para representarlo de forma más compacta se define el vector $\mathbf{a}(t) \in \{0, 1\}^S$ tal que

$$a_j(t) = \begin{cases} 1, & \text{si } t \geq s_j, \\ 0, & \text{en otro caso.} \end{cases}$$

De este modo, la tasa para un tiempo dado t, es $k + \mathbf{a}(t)^T\boldsymbol{\delta}$. Por último, para que la curva sea continua, es necesario ajustar los segmentos entre cada punto de cambio. Al igual que para la pendiente, el parámetro de compensación es una variable base m añadiendo todos los cambios hasta el tiempo t, es decir, $m + \mathbf{a}(t)^T\boldsymbol{\gamma}$, donde $\boldsymbol{\gamma}$ es un vector de ajustes de compensación. En este modelo, con el fin de que la curva sea continua, $\boldsymbol{\gamma}$ se toma de forma que $\gamma_j = -s_j\delta_j$. La Figura 17 muestra un modelo de tendencia lineal con puntos de cambio ajustado a los datos.

Por otro lado, para series temporales que sí presentan un crecimiento de saturación, el factor más importante del proceso es modelar cómo ha crecido la población y cómo se espera que siga creciendo. En este caso se emplea un modelo de crecimiento logístico, dado por la ecuación

$$g(t) = \frac{C}{1 + \exp\left(-k(t - m)\right)}, \tag{32}$$

donde C es la capacidad de los datos, k la tasa de crecimiento y m un parámetro de compensación.

El modelo presentado en (32) tiene sus limitaciones. Primeramente, la capacidad de carga no es siempre constante, por tanto es necesario introducir una función $C(t)$ que represente la capacidad a lo largo del tiempo. Por otra parte, al igual que en el modelo

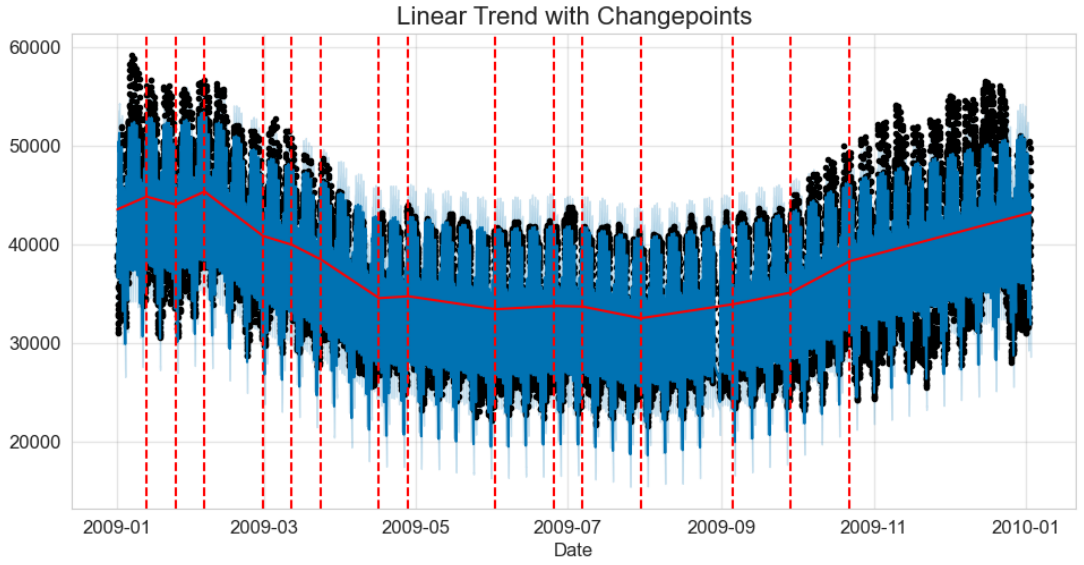

Figura 17: *Modelo de tendencia lineal con puntos de cambio*

anterior, se plantea la tasa de de crecimiento como $k + \mathbf{a}(t)^T\boldsymbol{\delta}$ y el parámetro de compensación $m + \mathbf{a}(t)^t\boldsymbol{\gamma}$. En este caso, cambian los valores de γ_j, $j = 1, \ldots, S$, con el fin de conectar correctamente los extremos de los segementos. En este modelo, se tiene que

$$\gamma_j = \left(s_j - m - \sum_{l<j} \gamma_l\right)\left(1 - \frac{k + \sum_{l<j}\delta_l}{k + \sum_{l\leq j\delta_l}}\right). \tag{33}$$

El modelo de crecimiento logístico por partes es

$$g(t) = \frac{C(t)}{1 + \exp\left(-(k + \mathbf{a}(t)^T\boldsymbol{\delta})(t - (m + \mathbf{a}(t)^T\boldsymbol{\gamma})\right)}. \tag{34}$$

También se puede especificar el número de puntos de cambio. Se supone $\delta_j \sim \text{Laplace}(0, \tau)$, donde τ contra la flexibilidad del modelo para alterar su tasa. Según el valor de τ se acerca a 0, el ajuste se reduce al cremimiento lineal o logístico estándar, es decir, no es función definida por partes.

5.2. Estacionalidad

Los datos de series temporales a menudo exhiben periodicidad, especialmente con datos comerciales, donde a menudo hay ciclos anuales, semanales y diarios. Prophet puede modelar un número ilimitado de dichos componentes periódicos en su término de estacionalidad, $s(t)$, de la ecuación (30).

Prophet se basa en la series de Fourier para ajustar el componente estacional. Sea P el periodo regular que se espera en la serie temporal, los efectos estacionales se aproximan mediante la fórmula

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right), \tag{35}$$

que consiste en una serie de Fourier estándar. En este caso no se considera el término independiente porque a su vez también se ajusta el componente de tendencia.

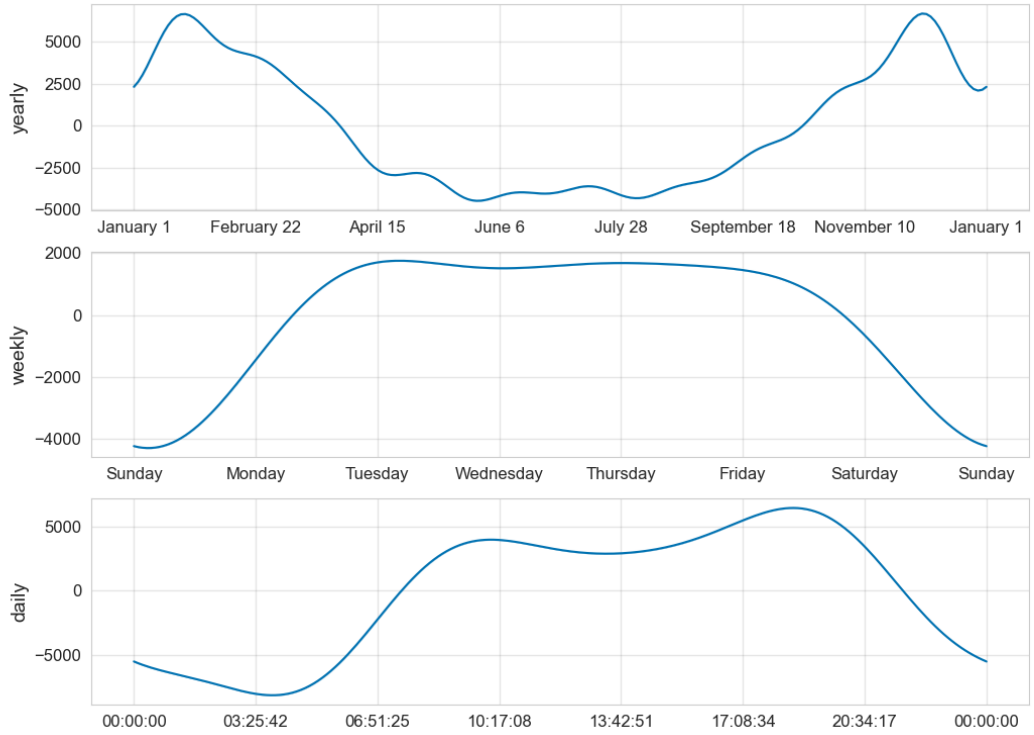

Figura 18: *Componentes de estacionalidad del modelo Prophet*

Ajustar el modelo (35) a los datos requiere estimar un total de $2N$ parámetros, $\boldsymbol{\beta} = [a_1, b_1, \ldots, a_N, b_N]^T$. Esta estimación se hace a partir de una matriz de estacionalidad con los vectores para cada valor de t de los datos históricos y los datos futuros. Por ejemplo, para estacionalidad semanal y $N = 3$,

$$X(t) = \left[\cos\left(\frac{2\pi(1)t}{7}\right), \sin\left(\frac{2\pi(1)t}{7}\right), \ldots, \sin\left(\frac{2\pi(3)t}{7}\right) \right].$$

El componente estacional es

$$s(t) = X(t)\boldsymbol{\beta}. \tag{36}$$

En este modelo generativo se toma $\boldsymbol{\beta} \sim \text{Normal}(0, \sigma^2)$ para imponer un suavizado previo a la estacionalidad. Aumentar N permite ajustar los patrones de los datos que cambian rápidamente, sin embargo este aumento también incrementa el riesgo de sobreajustar el modelo. Para datos con patrones anuales y semanales se ha visto que $N = 10$ y $N = 3$, respectivamente, funciona correctamente para la mayoria de los problemas. La obtención de estos parámetros se puede automatizar usando un procedimineto de selección de moldelo, como puede ser el criterio AIC (criterio de información de Akaike).

El algoritmo de Prophet utlizando las series de Fourier es muy práctico a la hora de ajustar los diferentes periodos de estacionalidad. En la Figura 18 puede observarse cómo se modelan los componentes anual, semanal y diario de los datos. Se puede ver que en los meses centrales del año, la demanda eléctrica alcanza el mímino. Por otro lado, los días de la semana con menor demanda son sábado y domingo y las horas del día con menor demanda es durante las horas de la madrugada.

5.3. Días festivos y eventos

Las componentes del modelo Prophet presentadas hasta ahora se corresponden con las de la descomposición clásica de la series temporal. Los analistas que trabajaban en *Facebook* decidieron añadir también una nueva componente que modela los efectos de los días festivos, ya que las vacaciones tienen un gran efecto en las actividades de negocios.

Holiday	Date
New Year's Day	01-01-2009
New Year's Day	01-01-2010
Good Friday	10-04-2009
Good Friday	02-04-2010
Good Friday	22-04-2011

Tabla 6: *Días festivos en Reino Unido*

La incorporación de una lista de días festivos en el modelo se simplifica asumiendo que los efectos de los días festivos son independientes. Para cada día festivo i, sea D_i el conjunto de las fechas pasadas y futuras de dicho festivo, la Tabla 6 exhibe un ejemplo de cómo se introduce la información de las vacaciones en Python. Se añade una función indicadora que represente si un tiempo t es durante el festivo i, y se le asigna a cada vacación el parámetro κ_i, que es el cambio en la predicción. Se crea una matriz de regresión

$$Z(t) = [\mathbf{1}(t \in D_1), \ldots, \mathbf{1}(t \in D_L)]$$

y se toma

$$h(t) = Z(t)\boldsymbol{\kappa}. \tag{37}$$

Al igual que con la estacionalidad, se considera que a priori $\boldsymbol{\kappa} \sim \text{Normal}(0, \nu^2)$. A menudo es importante añadir los efectos de los días festivos en un umbral alrededor de los mismos. Para ello se incluyen parámetros adicionales para los días circundantes a los festivos, tratando estos días como festivos. La Figura 19 es un gráfico con los efectos de las vacaciones en el modelo, se observar que para los días festivos la demanda eléctrica tiende a disminuir.

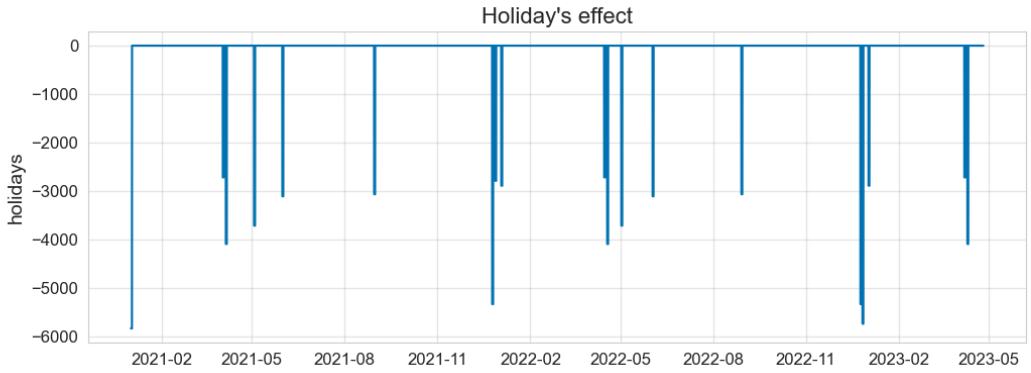

Figura 19: *Efecto de los días festivos en los datos*

5.4. Prediciones con Prophet

Se supone que se observa la serie temporal $\{X_1, X_2, \ldots, X_T\}$. En esta sección se va a estudiar la actuación del modelo presentado aplicado a los datos que se dispone. Una vez ajustados todos los componentes de la ecuación (30) el algoritmo se encarga de hacer las predicciones para cualquier valor de t.

Los componentes que modelan la estacionalidad y las vacaciones vistos se aplican al modelo sin ningún cambio. Sin embargo, la predicción de la tendencia es diferente, si el modelo se extrapola al futuro entonces la tendencia tendria una tasa constate.

En el modelo se supone que hay S puntos de cambio de la tendencia, por tanto se proponen futuros puntos de cambio dispuesto de forma aleatoria δ_j, de forma que la

frecuencia media de los puntos de cambio coincide con

$$\begin{cases} \delta_j = 0 \text{ con probabilidad } \frac{T-S}{S}, \\ \delta_j \sim \text{Laplace}(0, \lambda) \text{ con probabilidad } \frac{S}{T}, \end{cases} \qquad \forall j > T,$$

donde λ se puede estimar con la estimación de máxima verosimilitud, es decir, $\lambda = \frac{1}{S}\sum_{j=1}^{S} |\delta_j|$. De esta forma, se asume que la tendencia continua presentando cambios con la misma frequencia y magnitud que la que ha tenido en el conjunto de entrenamiento.

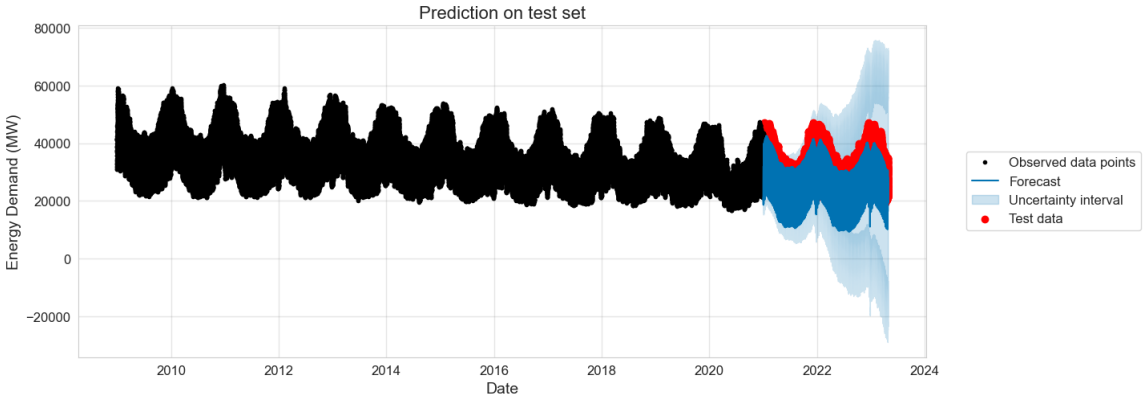

Figura 20: *Predicciones con el modelo Prophet*

El algoritmo de Prophet es capaz de modelar los datos teniendo en cuenta los diferentes tipos de estacionalidad que otros modelos no han podido manejar. La Figura 20 muestra las predicciones del modelo Prophet con los parámetros por defecto.

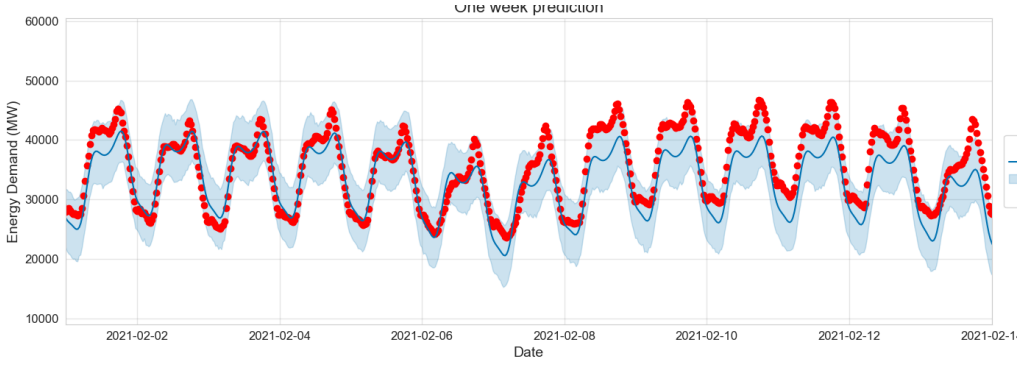

Figura 21: *Predicciones con el modelo Prophet para una semana aleatoria*

Al fijarse en una semana aleatoria, como muestra la Figura 21, se ve cómo este algoritmo sí ha sabido adapatarse a las variaciones de los datos durante un día. En este caso el tiempo de computación es de 19 minutos, no es un algoritmo que se ejecute de forma

inmediata, pero dada la magnitud de los datos y la precisión del modelo, es un tiempo razonable. En este caso, el MAPE es del 11,88 %, siendo hasta el momento el modelo capaz de adaptar lo datos con menor error.

A continuación, se ha implementado el mismo modelo, pero añadiendo también los efectos de los días festivos. Como se ha visto, por lo general en este dataset los días festivos provocan que haya un descenso en el consumo eléctrico.

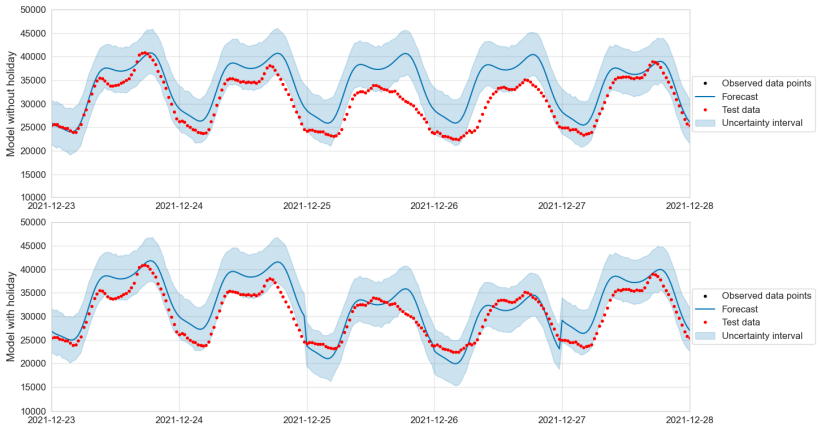

Figura 22: *Comparación de modelo con y sin efecto vacacional*

Se puede comparar el modelo con el componente vacacional y el primer modelo, observado la Figura 22. Se puede ver que el segundo modelo lo ajustar de forma más precisa, pues en el primer modelos, los valores reales ni si quiera caen en el intervalo de confianza del 95 % para los días 26 y 27 de diciembre.

En este caso, el algortimo ha tardado 23 minutos y el error es 11,86 %. El error no ha mejorado significativamente respecto al anterior, pero sí ha ajustado correctamente los posibles efectos vacacionales.

El último modelo que se propone es resultado de aplicar validación cruzada a los datos. La Figura 23 muestra la división realizada para la validación cruzada, se ha tomado solamente dos divisiones para que que el tiempo de computación a la hora de buscar los parámetros no ascienda demasiado. Ajustar el modelo final ha tenido un tiempo de ejecución de unos 20 y los parámetros ajustados son los relacionados con los componentes explicados anteriormente.

En este caso lo parámetros que se ha intentado optimizar son *changepoint_prior_scale*, *seasonality_prior_scale* y *holidays_prior_scale*, cada parámetro se corresponde a los valores vistos τ, σ^2 y ν^2, respectivamente. Estos parámetros permiten modular la fuerza

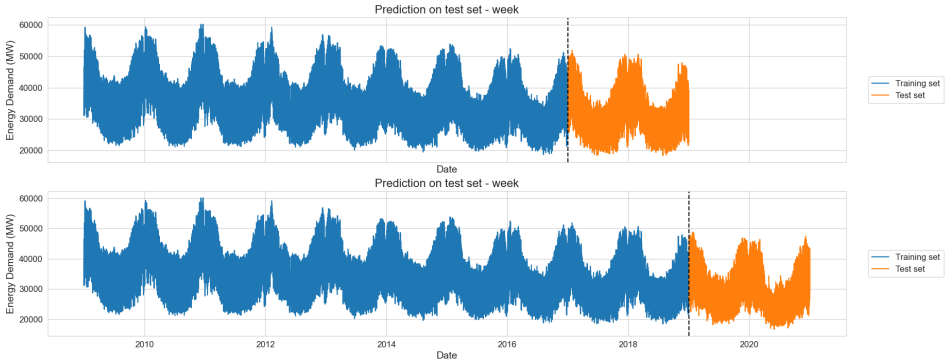

Figura 23: *Separación de los datos para validación cruzada*

de la componentes del modelo, es decir, los valores más grandes permiten que el modelo se ajuste a fluctuaciones más grandes, mientras que los valores más pequeños amortiguan la componente.

changepoint_prior_scale (τ)	seasonality_prior_scale (σ^2)	holidays_prior_scale (ν^2)
0.001	2.5	10

Tabla 7: *Parámetros obtenidos con validación cruzada*

Los resultados de la validación cruzada se encuentran en la Tabla 7. En Python, los valores predeterminados de estas variables son $\tau = 0{,}05$, $\sigma^2 = 10$ y $\nu^2 = 10$, a partir de estos valores se va a interpretar los obtenidos. Respecto a los puntos de cambios, se ha tomado un valor pequeño, es decir permite poco cambios de tendencia, de esta forma también se evita el sobreajuste del modelo. Además se ah visto que la tendencia de este modelo no varía significativamente e incluso puede modelarse con un recta. Teniendo encuenta la estacionalidad, el modelo toma un valor menor al fijado, en otra palabras, es conviente no darle tanto peso a los cambios estacionales.

Por último, respecto al componente que modela los días festivos, la validación cruzada ha dado el mismo valor que el fijado por Python, es decir, el valor fijado es suficiente para ajustar los datos en los días festivos.

En la Figura 24 se observa la actuación del modelo con los parámetros escogidos. La primera gráfica muestra dos semanas aleatoria del conjunto de testeo, las dos primeras semanas de febrero de 2021. Se ve cómo el modelo consigue ajustar a los cambios horarios que ocurren durante el día, además de las fluctaciones que ocurren a lo largo de la semana. Por ejemplo, se observa una baja en el día 7 que sigue de una subida los siguientes días y el modelo es capaz de capturar esos cambios.

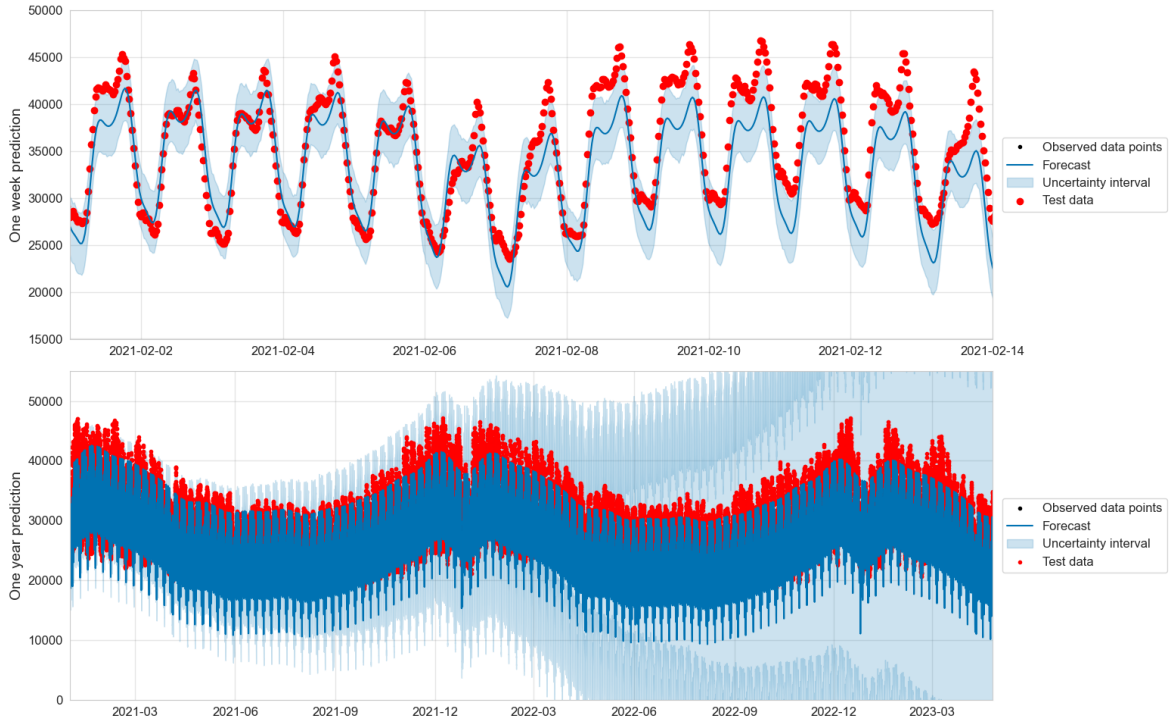

Figura 24: *Predicciones del modelo optimizado*

La siguiente gráfica muestra las predicciones de todo el conjunto test. Se ve como el modelo ajusta los cambios estacionales alnuales, que presentan un pico durante las vacaciones de navidad. El modelo se adapata correctamente, pero no predice los valores más altos de la serie, ya que predice una bajada de la tendendia. Se puede cómo también el intervalo de confianza aumenta rápidamente a la vez que las predicciones son más futuras.

El error absoluto medio porcentual final de este modelo es de 11,83, ha mejorado respecto a los otros en un 0,06 %, y el tiempo de ejecución es de 6 minutos y 58 segundos.

6. Modelos autorregresivos con árboles de decisión

Los árboles de decisión son modelos de aprendizaje automático y una técnica de análisis predictivo que utiliza para tomar decisiones o predecir resultado. Se basa en la idea de dividir un problema en varias decisiones más pequeñas y más simples, representadas en forma de un árbol.

Un árbol de regresión es una variante de los árboles de decisión que se utiliza para realizar tareas de regresión, es decir, para predecir valores numéricos en lugar de realizar clasificaciones. A diferencia de los árboles de clasificación, donde las hojas representan clases o categorías, en un árbol de regresión, las hojas contienen valores numéricos que representan las predicciones o estimaciones de una variable continua.

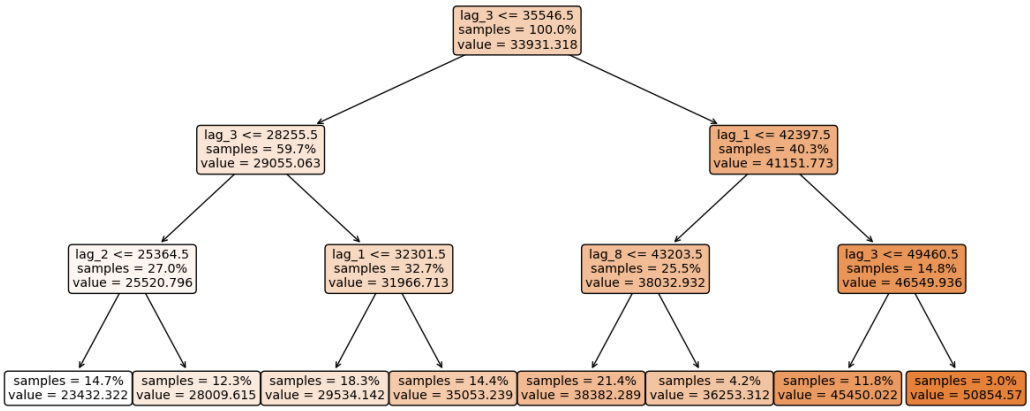

Figura 25: *Árbol de regresión*

Este razonamiento se puede aplicar a las series temporales, a partir de un número de observaciones se puede pronosticar un nuevo valor de la serie. Como las series temporales describen una variable continua, se utlizarán árboles de regresión en lugar de los árboles de decisión tradicionales. La Figura 25 representar un árbol de regresión que utilizan las diez últimas observaciones de la serie para predecir el siguiente valor, es decir, con los datos $\{X_{T-9}, \ldots, X_{T-1}, X_T\}$ calcula la predicción $\hat{X}_{T+1}$.

Con los árboles de decisión se puede predecir el valor de una variable objetivo de estudio, sin embargo en el análisis de series temporales es interesante estudiar las predicciones en un intervalo futuro, para poder estudiar su evolución. Por esta razón surgen dos tipos de métodos que permiten generar este tipo de predicciones múltiples, los método multi-paso o *multi-step*. En Python, la implementación de estos métodos es a través del paquete *skforecast*.

6.1. Método multipaso recursivo

Supóngase que se dispone de datos de una serie temporal $\{X_1, \ldots, X_T\}$ y se quiere predecir el valor de X_{T+k}. El funcionamiento de los árboles de decisión permite entrenar el modelos para predecir el siguiente valor de la serie, es decir, si se quiere predecir el valor X_{T+k} se requiere conocer el valor de X_{T+k-1}, pero para conocer este valor es necesario tener el de X_{T+k-2}, y así.

Este algoritmo es muy sencillo de implementar y consiste en utlizar los últimos n valores de la serie temporal para predecir el próximo, $\hat{X}_{T+1}$. El siguiete paso utliza los últimos $n-1$ valores de la serie y el nuevo valor pronosticado $\hat{X}_{T+1}$ para predecir el valor de $\hat{X}_{T+2}$. Así continua el algoritmo hasta obtener la predicción de X_{T+k}, se puede ver de forma esquemática en la Figura 26.

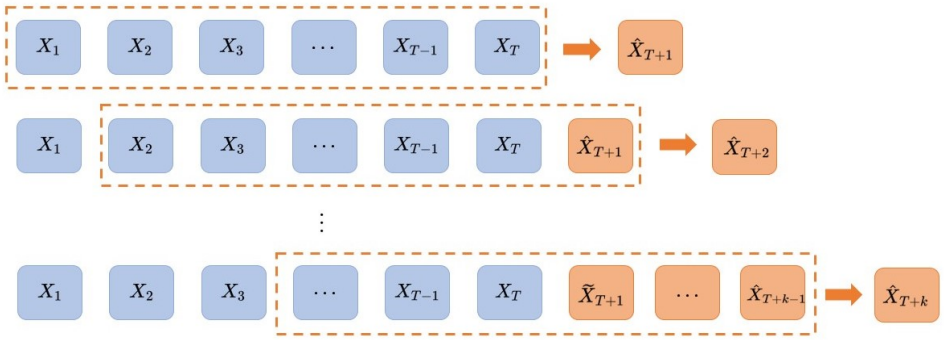

Figura 26: *Esquema del método multipaso recursivo con $n = T$*

El modelo se aplica a dos algoritmos que utilizan muchos árboles de regresión para hacer predicciones: Random Forest y Extreme Gradient Boosting (XGBoost).

Un Random Forest se basa en la combinación de múltiples árboles de regresión individuales para realizar predicciones más precisas y estables. Cada árbol se entrena de forma independiente en una muestra aleatoria del conjunto de datos de entrenamiento, utilizando diferentes subconjuntos de características. Luego, durante la fase de predicción, cada árbol emite su propia predicción en el caso de regresión, se calcula el promedio de las predicciones de todos los árboles.

Por otra parte, XGBoost combina múltiples árboles de decisión de forma iterativa. En cada iteración, ajusta un nuevo árbol para corregir los errores residuales del modelo anterior y actualiza las predicciones sumando las contribuciones de todos los árboles ajustados. Utiliza técnicas de regularización para controlar el sobreajuste y mejorar la generalización.

La implementación de Python permite elegir el número de retardos o lags, n, con los que se quiere trabajar, es decir, con cuántas observaciones se va a entrenar el modelo. En este caso se ha probado dos valores $n = 48$, $n = 48 \cdot 7$ y $n = 48 \cdot 30$, es decir los modelos se entrenan con los datos del día anterior, de la semana anterior y del mes anterior, respetivamente.

	RF1	RF2	XGB1	XGB2	XGB3
MAPE	18,82	19,03	13,128	19,03	14,97
MSE	$5{,}02 \cdot 10^7$	$4{,}49 \cdot 10^7$	$2{,}42 \cdot 10^7$	$4{,}49 \cdot 10^7$	$2{,}93 \cdot 10^7$
Tiempo	96,238	635,34	17,30	623,78	2821,58

Tabla 8: *Comparación de los modelos autorregresivos*

En la Tabla 8 se ha llamado a los modelos por sus iniciales y el número corresponde a los lags con los que se ha trabajado, es decir, 1, 2 y 3 son un día, una semana y un mes, respectivamente. El tiempo de ejecución del modelo Random Forest trabajando con un mes de retardo es muy alto así que no se ha incluido.

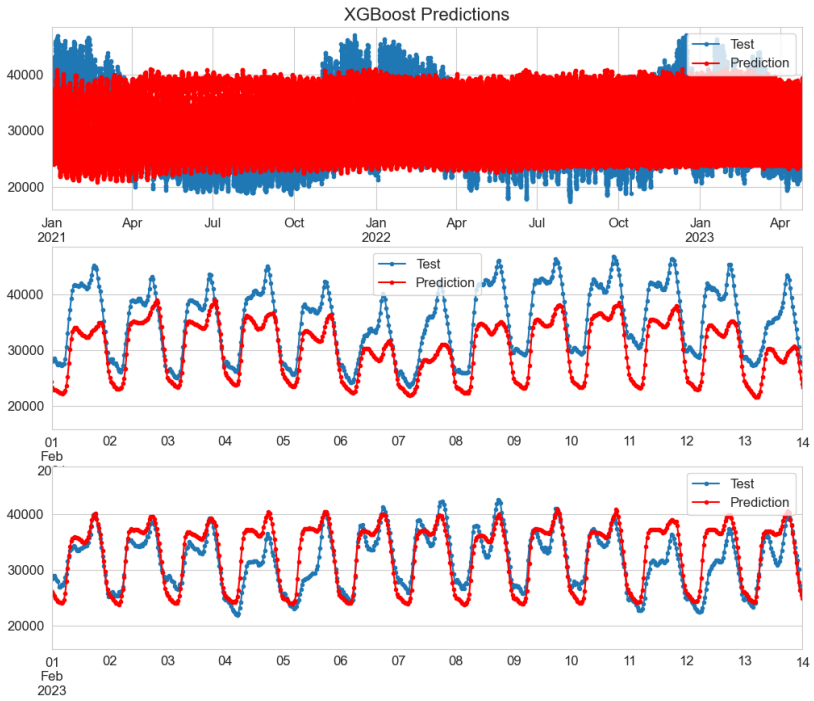

Figura 27: *Predicción con el modelo XGBoost*

Se ve que según aumenta la complejidad del modelos, tambiés aumenta su tiempo de ejecución, como es lógico. Por otra parte, el error MAPE alcanza valores entre 10 % y 15 %, cercanos a los valores del modelos Prophet. Todos los modelos presentan gráficas similares, en este caso la Figura 27 muestra el modelo que utiliza XGBoost y los datos del mes anterior para predecir.

Se puede observar, que el modelos no es capaz de predecir los patrones de estacionalidad que ocurren a lo largo de los años, aunque sí capta la fluctuaciones que ocurren durante las horas del día. Como modelo a corto plazo predice correctamente las primeras observaciones del conjunto test, sin embargo, a largo plazo las predicciones no son capaces de captar las diferencias entre los meses del años, ya que la predicciones siempre se encuentran en el mismo nivel.

Sin duda estos algoritmos son muy útiles, pero es importante ajustar el número de retardos. En este caso el número de retardos que mejor ha funcionado se corresponde con la estacionalidad mensual. A la hora de aplicar el algoritmo es recomentable ajustar el número de lags de forma que alcance al menos un ciclo de estacionalidad, pero teniendo en cuenta que para valores demasiados grandes, el algoritmo aumenta el tiempo de computación.

Una de las ventajas de este método es que también se puede estudiar la importancia de las variables, al igual que se puede estudiar en los árboles de decisión. La Tabla 9 muestra la impotancia de los retardos ordenados de forma descendente. Se puede ver que la variable más importante es la obsevación anterior, lo cual tiene sentido, ya que de una observación a otra no hay mucho cambio. El resto de variables tiene una importancia menor al 1 % del total, es decir, lo que escoge en predcir es la observación anterior sumando los pequeños efectos que puedan tener el resto de variables, por ello, las predicciones parecen estaticas, solo teniendo en cuenta los cambios diarios y no a largo plazo.

Variable	Importancia
lag 1	0.9814
lag 3	0.0040
lag 333	0.0024
lag 334	0.0024

Tabla 9: *Importancia de los lags en el modelo ordenado*

Todo esto motiva a buscar los parámetros que permitan ajustar correctamente los tipos de estacionalidad. Se va a trabajar con el modelo XGBoost por ser el algoritmo con

mejor tiempo de ejecución.

Ajustando los parámetros que ajusta la tasa de aprendizaje del modelo XGBoost, se consigue que se capte la estacionalidad anual, pero solamente en los primeros meses del test, como se ve en la Figura 28. Los modelos presentados logran capturar la primera subida y la primera bajada del año 2022, pero conforme avanzan los años, la predicciones quedan estáticas.

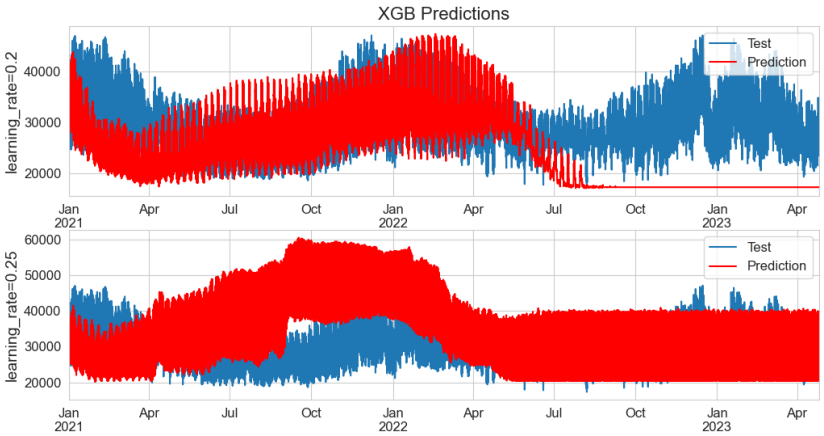

Figura 28: *Modelos XGBoost ajustados*

En este caso, el error MAPE asciende hasta más del 25 %, siendo el más alto de los modelos autorregresivos, pero también es el único en captar la estacionalidad anual. Por otra parte, los tiempo de ejecución son de alrededor de los 20 minutos.

El cambio con los modelos anteriores es que ahora el primer retardo recoge una impotancia del 70 %. Los lags 334, 669 y 3 ahora tienem una importancia del 1 %, pero el resto de variables son menores al 1 %.

7. Comparación de modelos

En este trabajo se ha estudiado cuatro modelos de predicción para series temporales, dos de ellos clásicos y los otros dos más recientes.

En cuantos a los modelos clásicos, procesoso SARIMA y método de suavizado exponencial, cabe destacar su limitación a la hora de trabajar con los datos. La gran dimensión de las variables y el tamaño del periodo de estacionalidad ($s = 48 \cdot 365$) ha provocado que el tiempo de computación se tan alto, hasta llegar a ser impracticable.

Sin embargo, no todo son contras a estos métodos. Aplicados a los datos reducidos sí funcionan correctamente y son capaces de modelizar la estacionalidad. Para casos con menor embergadura de datos y tamaño de estacionalidad inferior, los métodos funcionan correctamente.

Una diferencia clara entre los métodos clásicos y los no clásicos es que los métodos clásicos vistos no son capaces de trabajar con varios tipos de estacionalidad. Se ha visto que el algoritmo de Prophet puede modular la estacionalidad diaria, semanal, mensual y anual a la vez, lo que hace de este método una herramienta muy potente y eficaz. Además, incorpora un componente único, el componente vacacional.

La Tabla 10 muestra los componentes que possen y carecen los modelos estudiados, salvo los modelos autorregresivos, que no se basan en modelos de descomposición.

		Tendencia	Estacionalidad	Vacaciones
SARIMA		✓	✓	✗
Exponential Smoothing	Simple	✗	✗	✗
	Doble	✓	✗	✗
	Triple	✓	✓	✗
Prophet		✓	✓	✓

Tabla 10: *Componentes de los modelos*

Los modelos que siguen métodos autorregresivos no parten de una descomposición de series temporales, por lo que es más complicado medir su capacidad. El ajuste de estos modelos depende de los parámetros del modelos escogido y de los retardos usados. Cuanto mayor sea el número de lags, el modelso tendrá más datos para apredender sobre la tendecia o estaciodalidad de la serie, sin embargo el tiempo de ejecución aumenta mucho.

Encontrar un término medio entre una buena predicción y un buen tiempo de ejecución ha sido un punto clave en este trabajo. No cabe duda de que el modelo que ha cumplido

con las expectativas y ha conseguido ajustarse mejor a los datos es el modelo Prophet.

En este trabajo se ha encontrado un modelo que minimiza el error medio absoluto porcentual, con un valor del 10,67 %, se puede ver en la Figura 29. Cabe señalar que solamente los algoritmos autorregresivos y Prophet han modelado el dataset completo. Este algoritmo es el que mejor ha funcionado con los datos diponibles en todo momento, además presenta una forma sencilla de ajustar los parámetros.

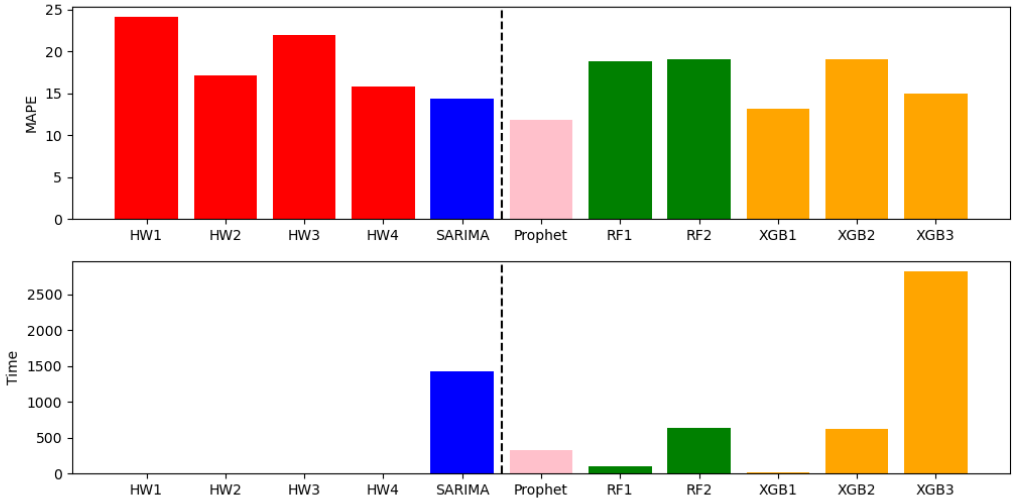

Figura 29: *Comparación según el MAPE y el tiempo de los métodos*

8. Conclusiones

A lo largo de este trabajo se ha estudiado diferentes tipos de modelos. Ha sido una gran labor de investigación del funcionamiento de los algoritmos así de aprender a implementarlos en Python. Los modelos SARIMA y de alisado exponencial ofrecen buenas predicciones, pero funcionan mejor con datos de menor dimensión.

Se ha visto que los modelos más eficaces son los más recientes, destacando la actuación del modelo Prophet. El objetivo del análisis de series temporales es la predicción de valores futuros de la serie, se ha tomado el modelo de Prophet, ajustandolo a todos los datos disponibles para predecir el valor de la demanda eléctrica durante todo el año 2023.

La predicción se muestra en la Figura 30. Se espera una subida de la demanda eléctrica durante el tercer trimestre de 2023 y una brusca en las tres útimas semanas del año. Los valores de la demanda fluctuan entre los valores 20000 y 35000 MW a lo largo del año, hasta la subida que asciende hasta los 40000 MW.

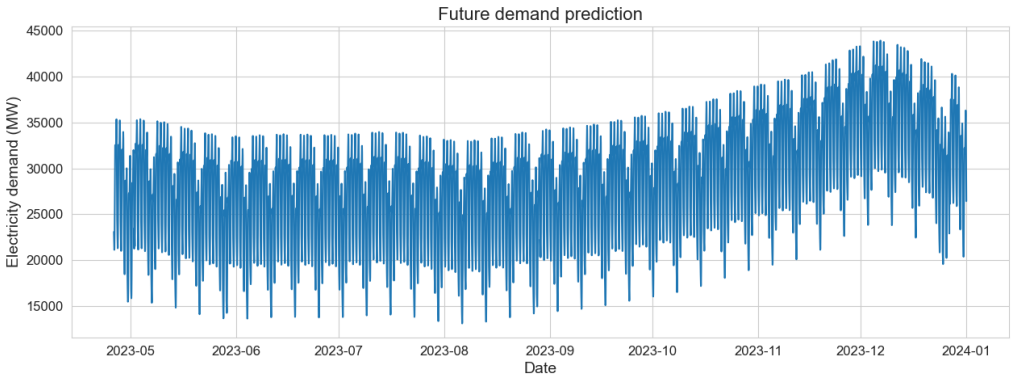

Figura 30: *Predicción de la demanda elétrica en Reino Unido durante 2023*

El mayor beneficio de realizar este trabajo ha sido aprende a modelar un caso real de series temporales, viendo las limitaciones y las ventajas de cada método. Así también, profundizando el las bases matemáticas que se esconden detrás y aplicando lo aprendido.

Referencias

[1] Brockwell, P.J., Davis, R.A. (1987). *Time Series: Theory and Methods*. Springer.

[2] Peña, D. (2010). *Análisis de series temporales*. Alianza Editorial.

[3] Brockwell, P.J., Davis, R.A. (1996). *Introduction to Time Series and Forecasting*. Springer.

[4] Box, G. E. P., Jenkins, G. M, Reinsel, G.C, Ljung, G.M. (1970). *Time series analysis: Forecasting and control*. Wiley.

[5] De Gooijer J.G. and Rob J. Hyndman R.J. (2006). *25 years of time series forecasting*. International Journal of Forecasting **22(3)**, 443-473

[6] Wang, S., Li, C. and Lim, A. (2021). *Why Are the ARIMA and SARIMA not Sufficient*.

[7] Hyndman, R.J., Athanasopoulos, G. (2014). *Forecasting: principles and practice*. OTexts.

[8] Hyndman, R., Koehler, A., Ord, K., Snyder, R. (2008). *Forecasting with Exponential Smoothing*. Wiley.

[9] Amat, R., Escobar, J., Skforecast: forecasting series temporales con Python y Scikit-learn. https://www.cienciadedatos.net/documentos/py27-forecasting-series-temporales-python-scikitlearn (Consultado el 1 de Julio de 2023).

[10] Amat, R., Escobar, J., Forecasting series temporales con gradient boosting: Skforecast, XGBoost, LightGBM y CatBoost. https://www.cienciadedatos.net/documentos/py39-forecasting-series-temporales-con-skforecast-xgboost-lightgbm-catboost (Consultado el 1 de Julio de 2023).

[11] Friedman, J.H., Tibshirani, R., Hastie, T. The Elements of Statistical Learning (2001) pp. 587-604. Convexity in Nonlinear Optimization in Aragón, F.J., Goberna, M.A., López, M.A. and Rodríguez, M.L. (2019) pp.55-89. Jerome H. Friedman, Robert Tibshirani y Trevor Hastie

[12] Taylor SJ, Letham B. 2017. *Forecasting at scale*. PeerJ Preprints 5:e3190v2 https://doi.org/10.7287/peerj.preprints.3190v2

[13] Rafferty, G.,(2023). *Forecasting Time Series Data with Prophet*. Packt Publishing.

[14] Korstanje, J., Advanced Forecasting with Python: With State-of-the-Art-Models Including LSTMs, Facebook's Prophet, and Amazon's DeepAR (2021) pp.253-271.

A. Detalles del desarrollo del trabajo

Todo el códido empleado en el trabajo para realizar modelos, obtener resultados y dibujar gráficos se encuentra en la carpeta de GitHub (https://github.com/MarinaPenalver/TFG). El código se ha dividido en diferentes notebooks, organizados según las secciones del proyecto. Los datos usado han sido extraídos de Kaggle, del enlace https://www.kaggle.com/datasets/albertovidalrod/electricity-consumption-uk-20092022.

Tarea	Tiempo (horas)
Recopilación de materiales	...
Estudio de bibliografía	...
Elaboración de resultados gráficos/numéricos	...
Redacción de la memoria	...
Total	150

Tabla 11: *Tiempo aproximado de dedicación al trabajo*

Asignatura	Páginas	Descripción
Series Temporales	10-21	Definición de series temporales y procesos estacionarios. Procesos AR, MA, ARMA, ARIMA y SARIMA. Metodología de Box-Jenkins
Análsis de datos I	General	Relación general con diferentes aspectos tratados.
Análsis de datos II	39-44	Modelos de árboles de decisión y sus variaciones.

Tabla 12: *Asignaturas relacionadas con el trabajo*