

**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.

## **Abstract**

# Índice

<b>1. Introducción</b>	<b>5</b>
1.1. Sobre el dataset . . . . .	5
<b>2. Series temporales</b>	<b>6</b>
2.1. Modelos estacionarios y estrictamente estacionarios . . . . .	7
2.2. Descomposición de una serie temporal . . . . .	9
<b>3. Procesos autoregresivos y de media móvil</b>	<b>11</b>
3.1. Procesos ARIMA y SARIMA . . . . .	13
<b>4. Suavizado exponencial</b>	<b>18</b>
4.1. Suavizado exponencial simple . . . . .	18
4.2. Método lineal de Holt . . . . .	21
4.3. Algoritmo de Holt-Winthers . . . . .	23
<b>5. Prophet</b>	<b>26</b>
5.1. Modelos de tendencia . . . . .	27
5.2. Estacionalidad . . . . .	28
5.3. Días festivos y eventos . . . . .	30
5.4. Predicciones con Prophet . . . . .	32
<b>6. Modelos autorregresivos con árboles de decisión</b>	<b>34</b>
6.1. Método multipaso recursivo . . . . .	35
6.2. Método multipaso directo . . . . .	35
<b>7. Comparación de modelos</b>	<b>36</b>
<b>8. Conclusiones</b>	<b>37</b>
<b>Referencias</b>	<b>38</b>
<b>A. Detalles del desarrollo del trabajo</b>	<b>39</b>

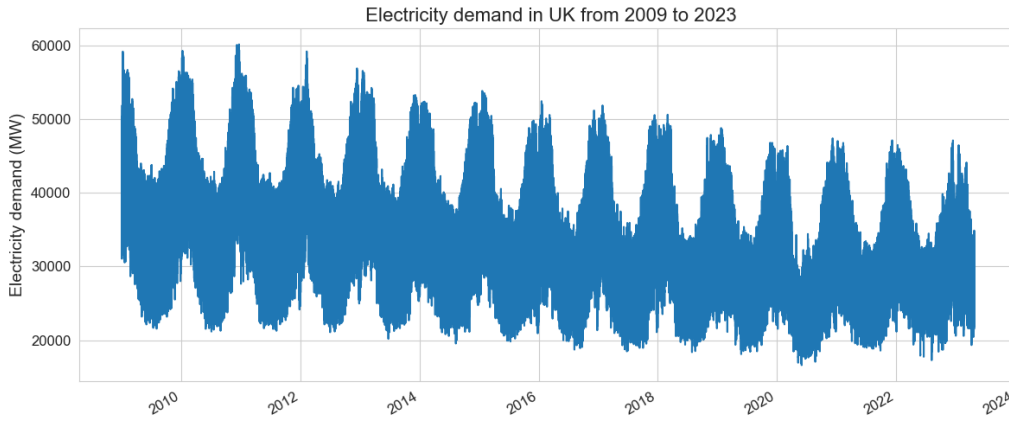
# **1. Introducción**

## **1.1. Sobre el dataset**

## 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 1 se muestra un ejemplo de serie temporal con tiempo discreto, donde las observaciones son tomadas cada treinta minutos.

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 1:** 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.  $\Omega$  es el espacio muestral,

2.  $\mathcal{F} \subset \mathcal{P}(\Omega)$  es una  $\sigma$ -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  $\{0, \pm 1, \pm 2, \dots\}$ ,  $\{1, 2, 3, \dots\}$ ,  $[0, \infty)$  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*

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

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

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

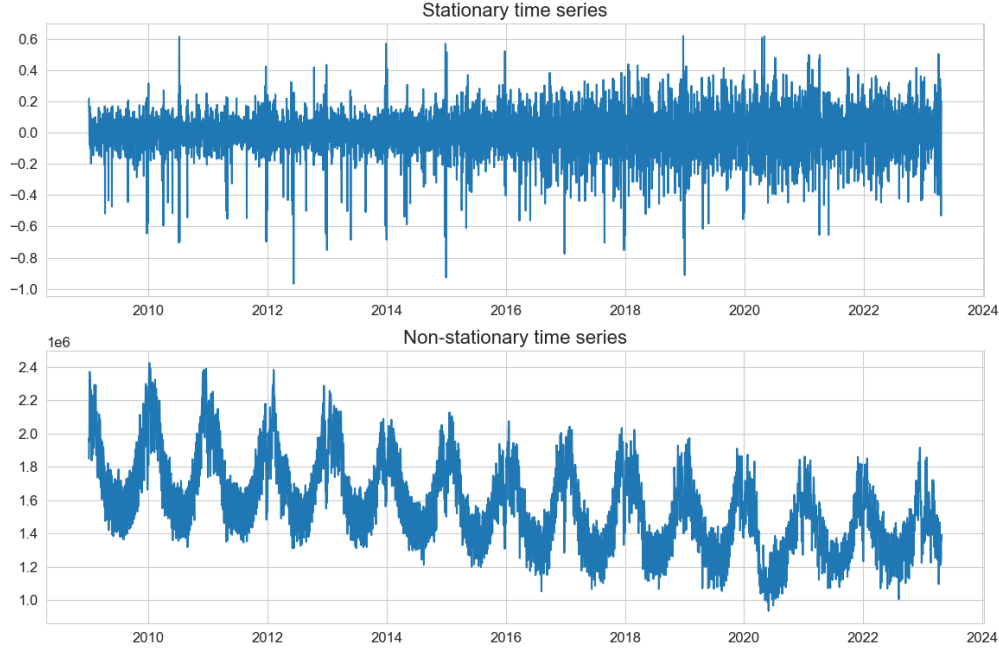
$$\rho(r, s) = \frac{\text{Cov}(X_r, X_s)}{\sqrt{\text{Var}(X_r)(\text{Var}_s)}} = \frac{\gamma(r, s)}{\sqrt{\gamma(r, r)\gamma(s, s)}}. \quad (2)$$

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

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

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

1.  $E[X_t^2] < \infty \quad \forall t \in \mathbb{Z}$
2.  $E[X_t] = \mu \quad \forall t \in \mathbb{Z}$
3.  $\gamma_X(r, s) = \gamma_X(r + t, s + t) \quad \forall r, s, t \in \mathbb{Z}$ .



**Figura 2:** Ejemplos de serie estacionaria y no estacionaria, respectivamente.

Esta definición de estacionaridad es referida habitualmente 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 constante, 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 redefinir la función de autocovarianzas de un proceso estacionario como la función de una sola variable,

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

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 autocorrelación simple de  $\{X_t\}$  se define de forma análoga como función de un solo valor para el retardo  $h$ ,

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



**Observación 2.7** *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 2.8 (Estacionaridad estricta)** *Una serie temporal  $\{X_t, t \in \mathbb{Z}\}$  se dice estrictamente estacionaria si las distribuciones conjuntas  $(X_{t_1}, \dots, X_{t_k})$  y  $(X_{t_1+h}, \dots, X_{t_k+h})$  son la misma para todos los enteros positivos  $k$  y para todo  $t_1, \dots, t_k \in \mathbb{Z}$ .*

Si  $\{X_t\}$  es estrictamente estacionaria, tomando  $k = 1$  en la Definición 2.8 se tienen que  $X_t$  tiene la 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 descomposición clásica, los datos se generan como la suma de tres efectos,

$$X_t = \mu_t + S_t + Y_t, \quad (4)$$

donde  $\mu_t$  es el *nivel de la serie* o *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

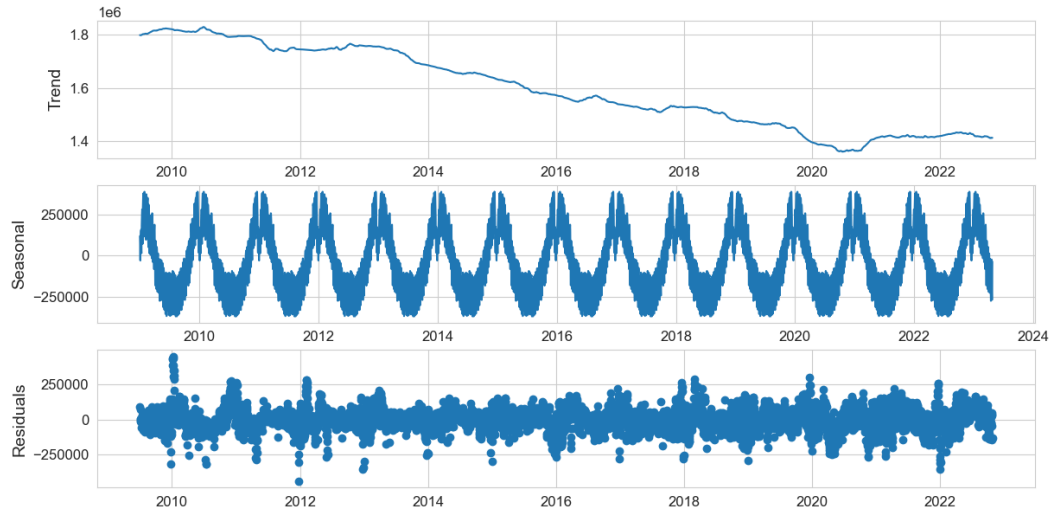
$$X_t = \mu_t + Y_t. \quad (5)$$

En la Figura 3 se muestra la descomposición aditiva de los datos. Habitualmente, el nivel  $\mu_t$  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 3:** *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 a la descomposición aditiva es la descomposición multiplicativa, que consta de los mismos componentes, y también es muy utilizada,

$$X_t = \mu_t \times S_t \times Y_t. \quad (6)$$

### 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  $\sigma^2$ ,*

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

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

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

*Si las variables aleatoria  $Y_t$  son independiente e idénticamente distribuidas con media 0 y varianza  $\sigma^2$ , entonces se conocen como*

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

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  $t$ ,*

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

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

$$BX_t = X_{t-1}. \quad (11)$$

*Además cumple las propiedades siguientes:*

1.  $B\mu = \mu$ , para  $\mu$  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

$$\phi(B)X_t = \theta(B)Y_t, \quad (12)$$

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

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

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

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 cualquier función de autocovarianza  $\gamma(\cdot)$  tal que  $\lim_{h \rightarrow \infty} \gamma(h) = 0$ , y para todo entero  $k > 0$ , es posible encontrar un proceso ARMA con función de autocovarianza  $\gamma_X(h) = \gamma(h)$   $h = 0, 1, \dots, k$ .

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

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

$$X_t = \theta(B)Y_t, \quad (15)$$

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,  $\forall h > 0$

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

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

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

$$\phi(B)X_t = Y_t \quad (16)$$

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 círculo 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  $\nabla$ , 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*

$$\phi(B)(1 - B)^d X_t = \theta(B)Y_t \quad (17)$$

*o, con la notación del operador de diferenciación*

$$\phi(B)\nabla^d X_t = \theta(B)Y_t, \quad (18)$$

donde  $\{Y_t\} \sim WN(0, \sigma^2)$ ,  $\theta(B) = 1 + \theta_1 B + \dots + \theta_q B^q$ ,  $\phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$  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*

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

donde

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

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

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

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

$$\hat{X}_T(k) = \alpha_1 X_T + \alpha_2 X_{T-1} + \dots + \alpha_T X_1.$$

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

$$\hat{X}_T(k) = E[X_{T+k} | X_1, \dots, X_T]. \quad (20)$$

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

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

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

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

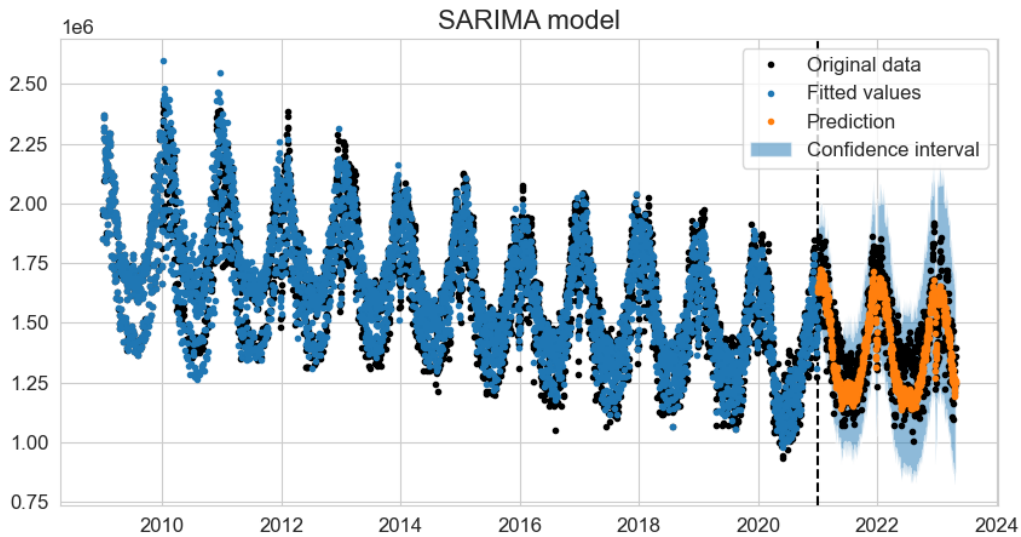
En consecuencia, para  $k \geq \max(1, q + sQ + 1 - (d + s + p + sP))$  las predicciones  $\hat{X}_T^K$  son las soluciones de la ecuación en diferencias (22) y se tiene que

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

donde

1.  $T_T(k)$  es el componente estacional solución de  $(1 - B)^{d+1}T_T(k) = 0$ .  $T_T(k)$  es un polinomio de grado  $d$  con coeficientes que se adaptan a lo largo del tiempo.
2.  $S_T(k)$  es el componente estacional solución de  $S_s(B)S_T(k)$ .  $S_T(k)$  es una función de periodo  $s$  tal que  $\sum_{j=1}^s S_T(j) = 0$ .
3.  $t_T(k)$  es el componente transitorio, solución de  $\Phi(B^s)\phi(B)t_T(k) = 0$ .  $t_T(k)$  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 4 muestra tanto los datos ajustados como las predicciones.

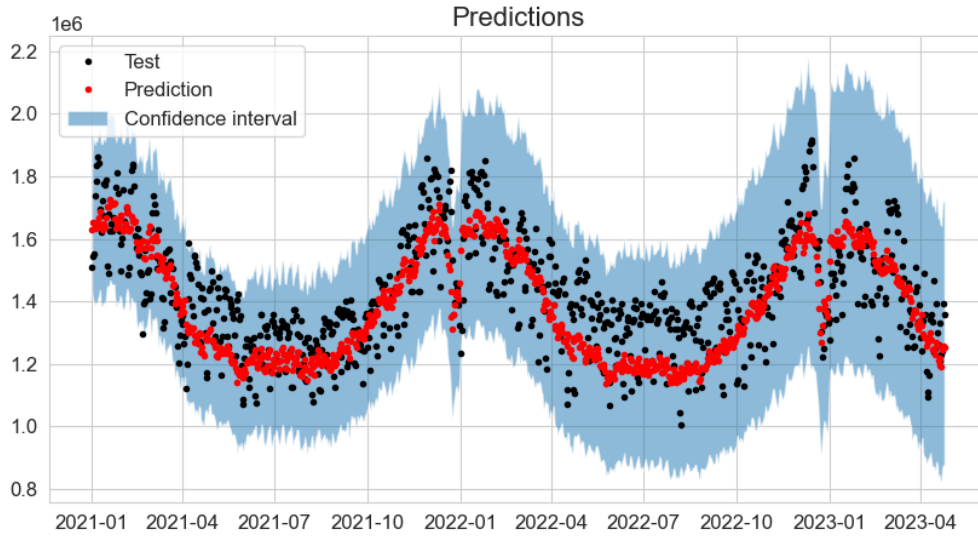


**Figura 4:** Modelo ajustado con SARIMA

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 5 se ve cómo la predicción se permanece aproximadamente en la media de todas las observaciones. El intervalo de confianza del 95 % si logra capturar todas las observaciones en su interior.

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



**Figura 5:** *Predicciones con SARIMA*

De este modelo, cabe destacar que el error medio absoluto medio es del 14,31% y su tiempo de ejecución es 23 minutos. El error 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, pues con los modelos SARIMA solo es posible ajustar a un tipo de estacionalidad.

Se ha utilizado 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  $ARIMA(1, 1, 2)_{365} \times (1, 1, 1)$ , los valores de los parámetros se encuentran en la Tabla 1. 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.

$\phi_1$	$\theta_1$	$\theta_2$	$\Phi_1$	$\Theta_1$	$\sigma^2$
0,0634	-0,2977	-0,6445	0,1278	-0,9998	393,9037

**Tabla 1:** *Parámetros de modelo SARIMA*

En la metodología de Box-Jenkins la idea es proponer diferentes modelos gracias a los



gráficos de las funciones de autocorrelación, 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 los 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 basados 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 sencillo 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 década de los 50s, 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,  $X_t$ . Si también se dispone de la observación  $X_t$ , el error de la predicción se calcula como  $X_t - \hat{X}_t$ , donde  $\hat{X}_t$  es la predicción. En el método de suavizado exponencial simple, la predicción para el siguiente tiempo se define como

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

donde  $\alpha$  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  $\alpha$  tiene un valor cercano a 1, la nueva predicción tendrá un ajuste significativo por el error en la anterior predicción. Por el contrario, cuando  $\alpha$  es cercano a 0, el nuevo pronóstico tendrá un ajuste muy pequeño.

Otra forma de escribir la ecuación (24) es

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

La predicción  $\hat{X}_{t+1}$  está basada en la ponderación de la observación más reciente,  $X_t$  con un peso de  $\alpha$  y la predicción 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:

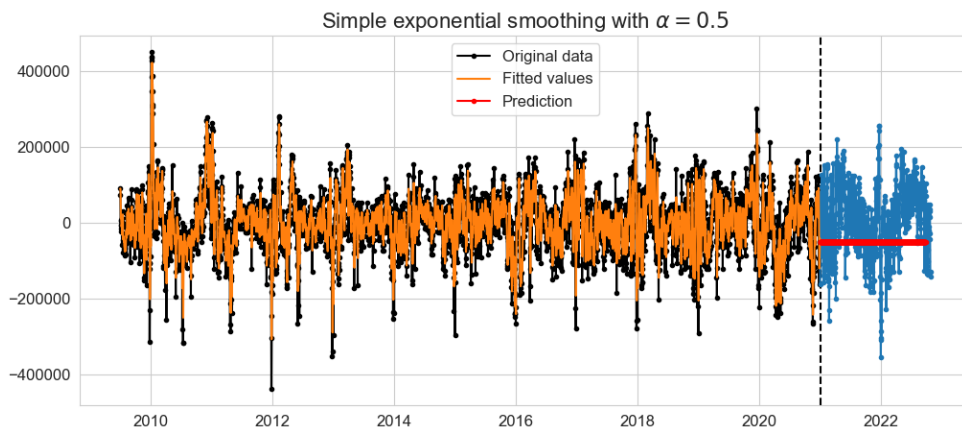
$$\begin{aligned}\hat{X}_{t+1} &= \alpha X_t + (1 - \alpha)[\alpha X_{t-1} + (1 - \alpha)\hat{X}_{t-1}] \\ &= \alpha X_t + \alpha(1 - \alpha)X_{t-1} + (1 - \alpha)^2 \hat{X}_{t-1}.\end{aligned}$$

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

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

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”.

La limitación de este método es que no puede modelar datos que presentan tendencia o componente estacional. Para aplicar este algoritmo se ha eliminado dichas componentes de los datos. En la Figura 6 se puede ver cómo el modelo ajusta a los datos, pero las predicciones son estáticas en el tiempo, con lo que no se ajusta correctamente a los posibles cambios de los datos.

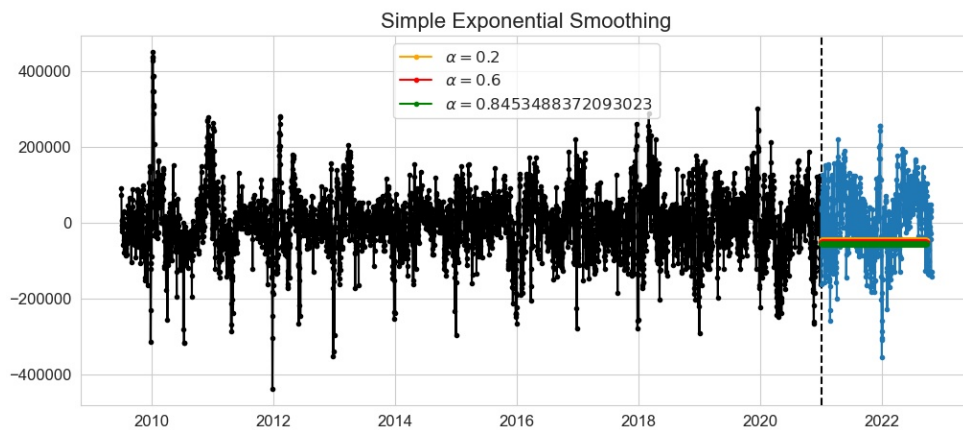


**Figura 6:** Valores ajustados y predicción de método de suavizado exponencial simple

En la Figura 7 se presentan tres modelos, en los dos primeros se fija el valor de  $\alpha$  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, con un valor por debajo de la media.

En la Tabla 2, se ve que el error absoluto medio porcentual asciende hasta el 450 %. El error alcanza valores tan altos porque los datos son cercanos a cero, es decir, al dividir por dichos valores, el error incrementa rápidamente. Por esta razón, en este caso, también se ha estudiado la suma de los errores al cuadrado (SSE). La tabla muestra que, efectivamente, el menor valor de SSE se alcanza para el valor de  $\alpha$  que ha sido optimizado. Sin embargo, el ajuste de las predicciones no es lo suficiente preciso.



**Figura 7:** Modelos del método de suavizado exponencial simple con diferentes parámetros de suavizado

	Modelo 1	Modelo 2	Modelo 3
$\alpha$	0,2	0,6	0,8453488
MAPE	382,808	423,329	456,399
SSE	$1,6106 \cdot 10^{13}$	$1,2314 \cdot 10^{13}$	$1,19017 \cdot 10^{13}$
Tiempo (s)	0,029	0,033	0,045

**Tabla 2:** Parámetros, tiempo y error del método de suavizado exponencial simple

## 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. La predicción de método lineal de Holt se basa en utilizar dos constantes  $\alpha$  y  $\beta^*$  (con valores entre 0 y 1), y tres ecuaciones

$$\text{Nivel: } l_t = \alpha X_t + (1 - \alpha)(l_{t-1} + b_{t-1}), \quad (26)$$

$$\text{Crecimiento: } b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1}, \quad (27)$$

$$\text{Predicción: } \hat{X}_{t+h|t} = l_t + b_t h. \quad (28)$$

$l_t$  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 8 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.

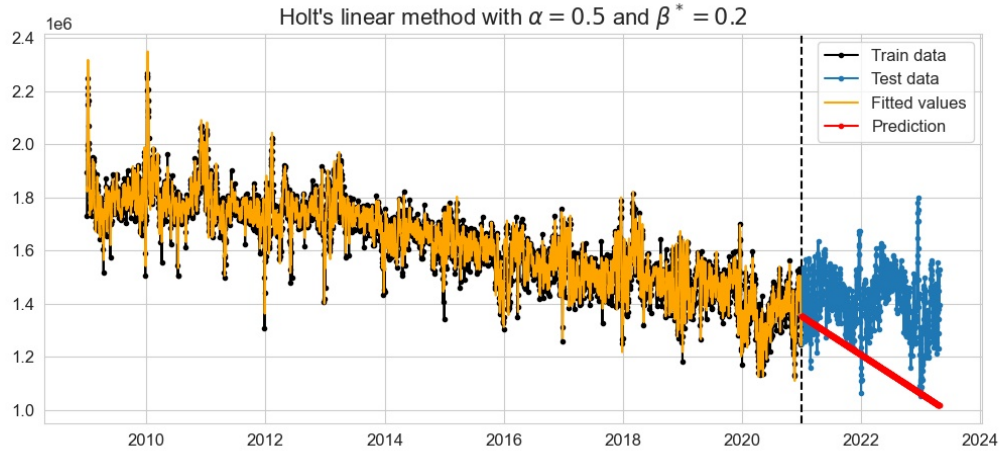
A continuación se ha probado tres modelos con las opciones disponibles de Python. El primer modelo es el presentado por las ecuaciones (26), (27) y (28). El segundo modelo ajusta la tendencia usando un modelo multiplicativo, en lugar de usar el modelo aditivo visto. Por último, el tercer modelo presenta el método amortiguado del método de Holt, en el cual se introduce un nuevo parámetro de amortiguamiento,  $\phi$ .

	Aditivo	Multiplicativo	Amortiguado
$\alpha$	0,85357	0,85357	0,877149
$\beta$	0,021886	0,02188	0,0001
$\phi$	-	-	0,99
MAPE	23,202	10,671	7,031
Tiempo (s)	0,515	0,684	0,598

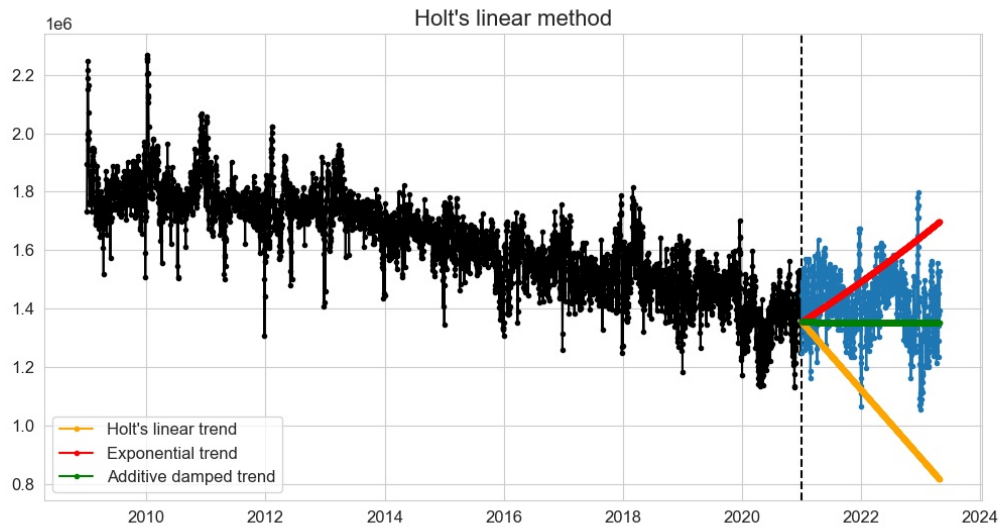
**Tabla 3:** Parámetros, error y tiempo del método de Holt

Todos los modelos optimizan los valores de  $\alpha$  y  $\beta^*$ , se puede ver los parámetros escogido en la Tabla 3. 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 9 se observa cómo

los modelos uno y dos se alejan de los valores reales del test y no predicen correctamente. El modelo amortiguado es el único que ofrece una predicción coherente con los datos presentados. La Tabla 3 muestra que el error absoluto medio porcentual se minimiza en el modelo amortiguado con un valor del 7 %.



**Figura 8:** Valores ajustados y predicción del modelo lineal de Holt



**Figura 9:** Predicciones de tres modelos con el método lineal de Holt

### 4.3. Algoritmo de Holt-Winters

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  $\alpha$ ,  $\beta^*$  y  $\gamma$ , respectivamente.

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

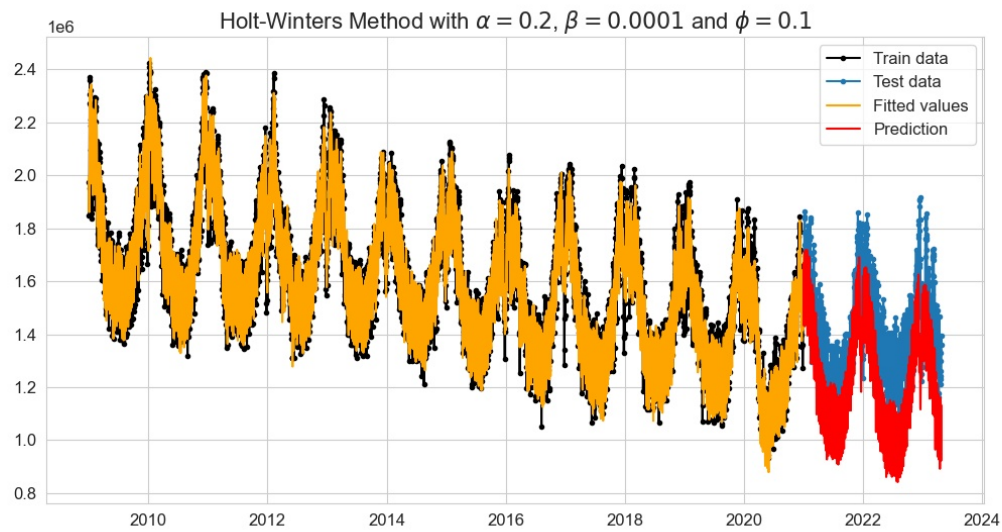
donde  $s$  es el periodo de estacionalidad y  $k = \lfloor (h - 1)/m \rfloor$ , lo que asegura que las estimaciones de los índices estacionales utilizados para la predicción provienen del último año de la muestra.

A continuación, se van a presentar modelos del algoritmo de Holt-Winters aplicados a los datos medidos diaramente y a los datos completos. La Figura 10 muestra el modelo es capaz de ajustarse a los datos del entrenamiento, sin embargo para en los datos de testeo, la predicciones parecen desplazadas hacia bajo. El modelo esperaba que el nivel de la serie siguiera bajando, pero a partir de 2020 se aprecia una ligera subida.

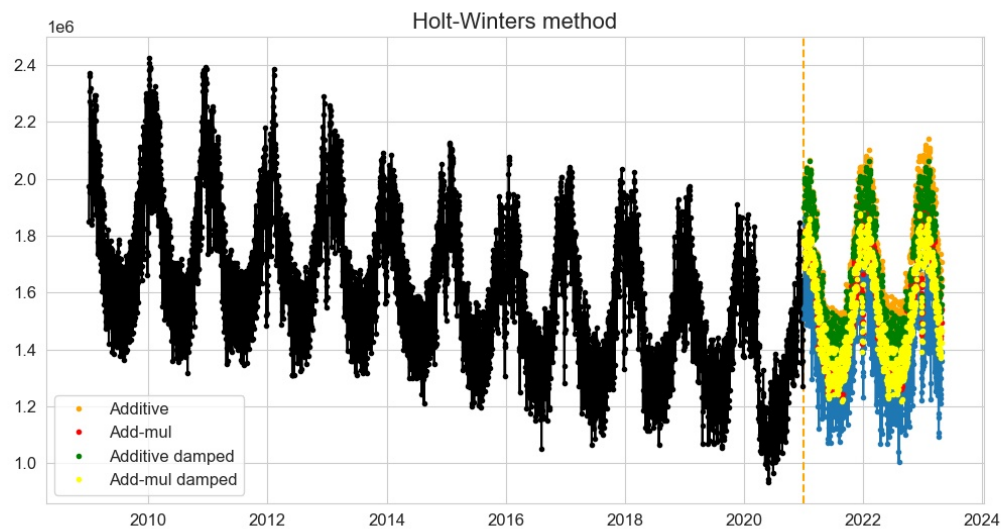
Al igual que con los otros modelos, se ha implementado las opciones disponibles, ajustado hasta cuatro modelos distintos que aplican el algoritmo de forma aditiva y multiplicativa y a su vez añadiendo o no el parámetro de amortiguamiento. En todos los métodos el algoritmo busca los parámetros óptimos. Los parámetros y resultados se pueden observar en la Tabla 4. Se ve que el modelo que minimiza el MAPE es el multiplicativo, con un valor del 17 %

	Aditivo	Multiplicativo	Ad. amortiguado	Mul. Amortiguado
$\alpha$	0,6060	0,995	0,6060	0,995
$\beta$	0,0001	0,0001	0,0001	0,0001
$\gamma$	0,3939	0,005	0,3939	0,005
$\phi$	-	-	0,99	0,99
MAPE	24,13	17,08	21,92	15,75
Tiempo (s)	0,913	01,372	0,923	0,923

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



**Figura 10:** Valores ajustados y predicción del modelo lineal de Holt-Winters

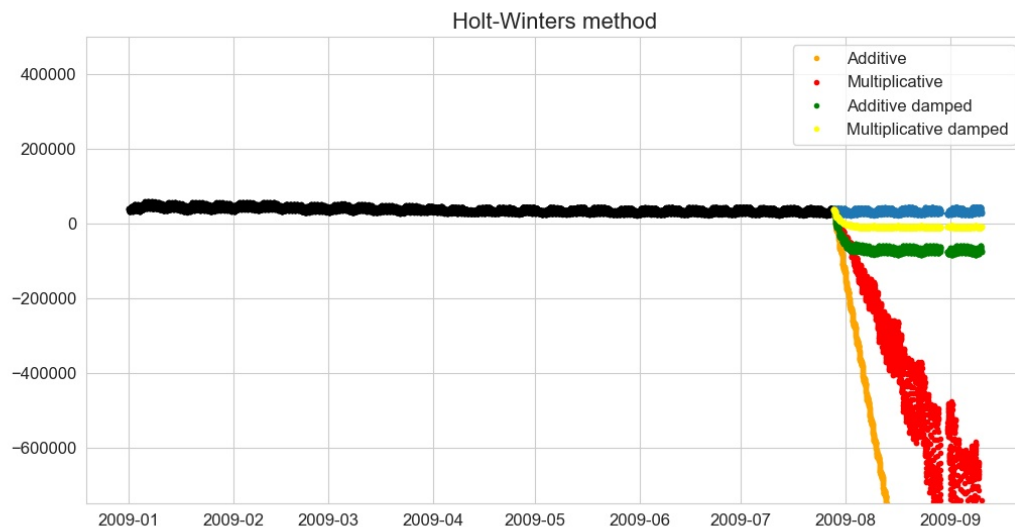


**Figura 11:** Predicciones de los cuatro modelos del método de Holt-Winters

### CAMBIAR GRÁFICO Y CORREGIR UN POCO EL TEXTO

Por otra parte, se puede ver las limitaciones de este método al aplicarse a los datos completos. El número de observaciones es tan alto que el tiempo de ejecución del algoritmo supera las 10 horas, por tanto se ha aplicado el algoritmo solamente a los últimos meses del conjunto de datos de entrenamiento. La Figura 12 muestra las predicciones del método





**Figura 12:** Predicciones del método de Holt-Winters para los datos completos

y se observa lo nefastas que son. Las predicciones se pueden dividir en dos grupos, las que incluyen el parámetro de amortiguamiento y las que no. Las que no presentan este parámetro se alejan de los valores reales a gran velocidad aumentando el error notablemente. Mientras, los métodos que presentan el parámetro de amortiguamiento son más estables, pero las predicciones de la demanda eléctrica para el mes de octubre son negativas, con lo cual no funciona correctamente.

Una limitación de este algoritmo es que no capta correctamente las predicciones a largo plazo, en este caso. Por otra parte, al aumentar el número de observaciones y a su vez el periodo de estacionalidad el tiempo de computación aumenta demasiado, pues tratando con los datos diarios en todos los casos se ejecuta en apenas un segundo.

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 metodologías de predicción:

1. Las herramientas de predicción más automáticas disponibles tienden a ser inflexibles e incapaces de ajustarse a suposiciones adicionales.
2. La herramientas de predicción más robustas requieren un análisis 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 importantes: la tendencia, la estacionalidad y los días festivos. Estos componentes son combinados en la ecuación

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t. \quad (29)$$

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 corresponde 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.
- $\varepsilon_t$  es el término del error, que engloba todos los demás cambios que no ajustan los demás componentes del modelo.

La combinación 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, una tasa de crecimiento constante por partes proporciona un modelo útil. El modelo lineal se escribe como

$$g(t) = (k + \mathbf{a}(t)^T \boldsymbol{\delta})t + (m + \mathbf{a}(t)^T \boldsymbol{\gamma}). \quad (30)$$

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  $s_j$ ,  $j = 1, \dots, S$ , donde la tasa de crecimiento puede cambiar. Se define el vector de ajustes de tasa  $\boldsymbol{\delta} \in \mathbb{R}^S$ , donde  $\delta_j$  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

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

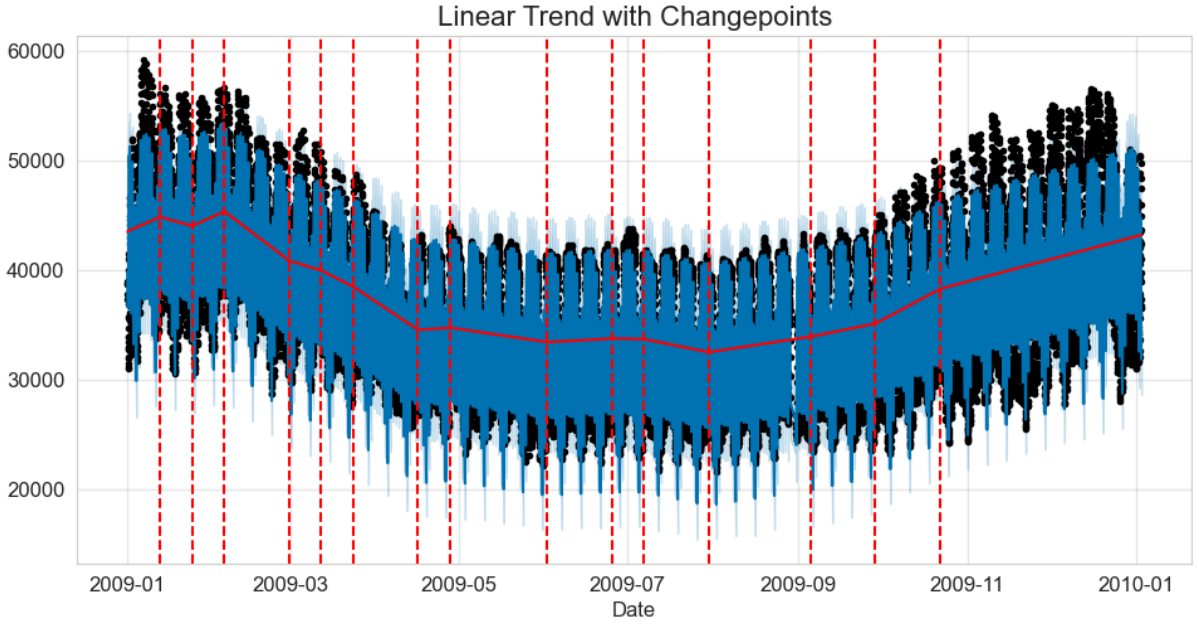
De este modo, la tasa para un tiempo dado  $t$ , es  $k + \mathbf{a}(t)^T \boldsymbol{\delta}$ . Por último, para que la curva sea continua, es necesario ajustar los segmentos entre cada punto de cambio. Al igual que para la pendiente, el parámetro de compensación es una variable base  $m$  añadiendo 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 13 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

$$g(t) = \frac{C}{1 + \exp(-k(t - m))}, \quad (31)$$

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

El modelo presentado en (31) tiene sus limitaciones. Primeramente, la capacidad de carga no es siempre constante, por tanto es necesario introducir una función  $C(t)$  que



**Figura 13:** *Modelo de tendencia lineal con puntos de cambio*

represente la capacidad a lo largo del tiempo. Por otra parte, al igual que en el modelo anterior, se plantea la tasa 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  $\gamma_j$ ,  $j = 1, \dots, S$ , con el fin de conectar correctamente los extremos de los segmentos. En este modelo, se tiene que

$$\gamma_j = \left( s_j - m - \sum_{l < j} \gamma_l \right) \left( 1 - \frac{k + \sum_{l < j} \delta_l}{k + \sum_{l \leq j} \delta_l} \right). \quad (32)$$

El modelo de crecimiento logístico por partes es

$$g(t) = \frac{C(t)}{1 + \exp(-(k + \mathbf{a}(t)^T \boldsymbol{\delta})(t - (m + \mathbf{a}(t)^T \boldsymbol{\gamma})))}. \quad (33)$$

## 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 (29).

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

$$s(t) = \sum_{n=1}^N \left( a_n \cos \left( \frac{2\pi nt}{P} \right) + b_n \sin \left( \frac{2\pi nt}{P} \right) \right), \quad (34)$$

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.

Ajustar el modelo (34) a los datos requiere estimar un total de  $2N$  parámetros,  $\beta = [a_1, b_1, \dots, a_N, b_N]^T$ . 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  $N = 3$ ,

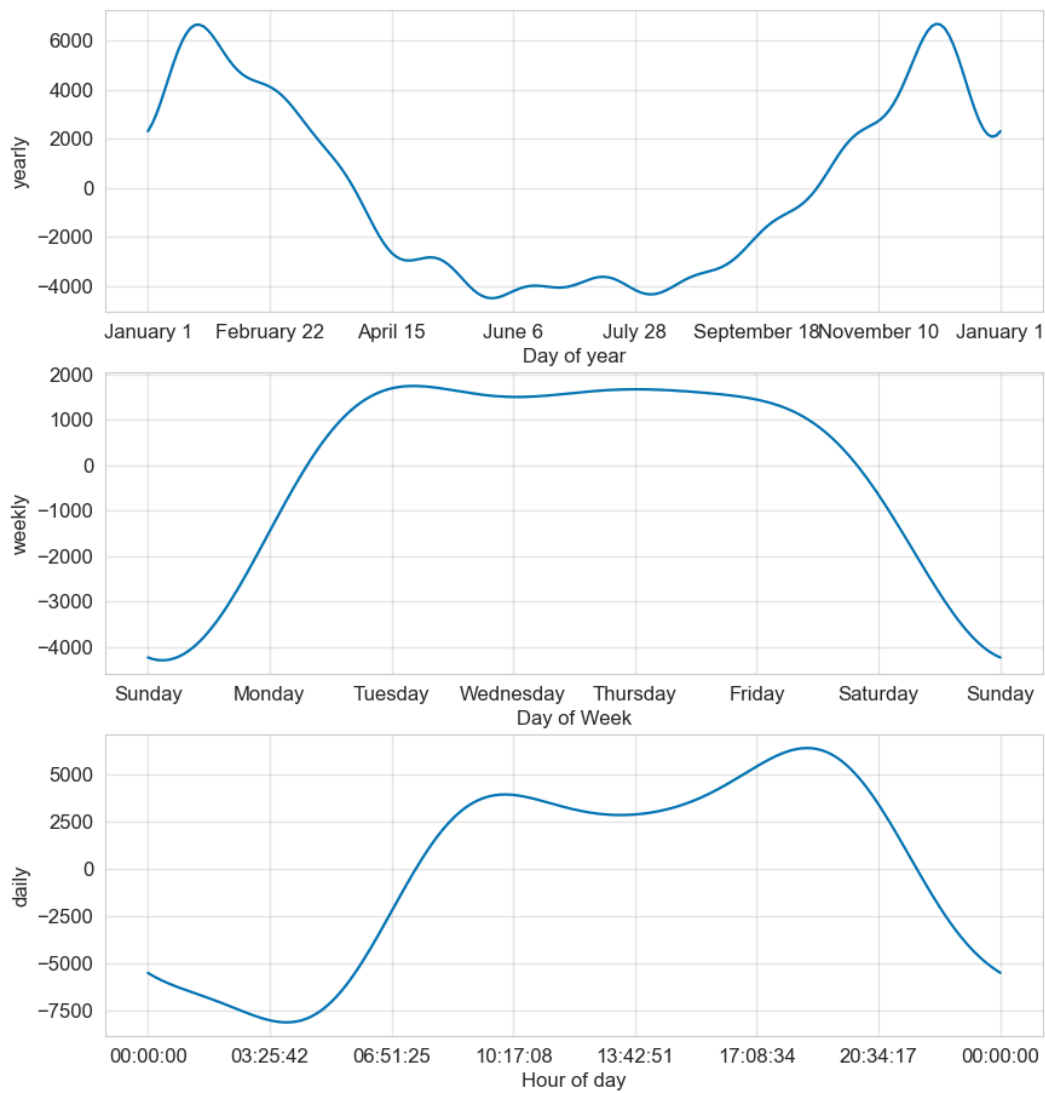
$$X(t) = \left[ \cos \left( \frac{2\pi(1)t}{7} \right), \sin \left( \frac{2\pi(1)t}{7} \right), \dots, \sin \left( \frac{2\pi(3)t}{7} \right) \right].$$

El componente estacional es

$$s(t) = X(t)\beta. \quad (35)$$

En este modelo generativo se toma  $\beta \sim \text{Normal}(0, \sigma^2)$  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 mayoría de los problemas.

El algoritmo de Prophet utilizando las series de Fourier es muy práctico a la hora de ajustar los diferentes periodos de estacionalidad. En la Figura 14 puede observarse cómo se modelan los componentes anual, semanal y diario de los datos.



**Figura 14:** Componentes de estacionalidad del modelo Prophet

### 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.

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 5 exhibe un ejemplo de cómo se

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 5:** *Días festivos en Reino Unido*

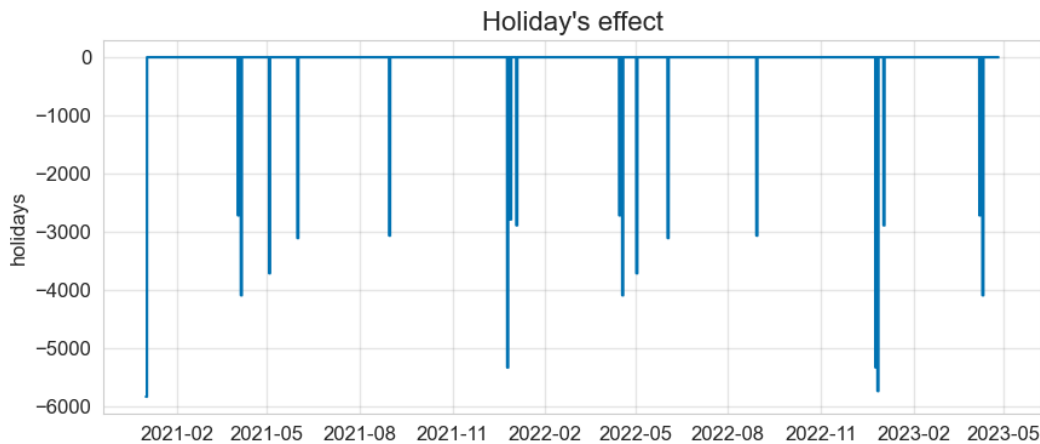
introduce la información de las vacaciones en Python. Se añade una función indicadora que represente si un tiempo  $t$  es durante el festivo  $i$ , y se le asigna a cada vacación el parámetro  $\kappa_i$ , que es el cambio en la predicción. Se crea una matriz de regresión

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

y se toma

$$h(t) = Z(t)\boldsymbol{\kappa}. \quad (36)$$

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 15 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 15:** *Efecto de los días festivos en los datos*

## 5.4. Predicciones con Prophet

En esta sección se va a estudiar la actuación del modelos presentado aplicado a los datos que se dispone. Una vez ajustados todos los componentes de la ecuación (29) el algoritmo se encarga de hacer las predicciones para cualquier valor de  $t$ .

En las Figuras 16 y 17 se muestra las predicciones del modelo. Se observa cómo este modelo sí ha sido capaz de modelar correctamente la componente estacional. Cabe resalta que, hasta el momento, ninguno de los modelos había sido de ajustar el conjunto de datos completo. Y además, se observa en la Figura 17 cómo también ajusta de forma precisa los datos medidos durante las horas del día.

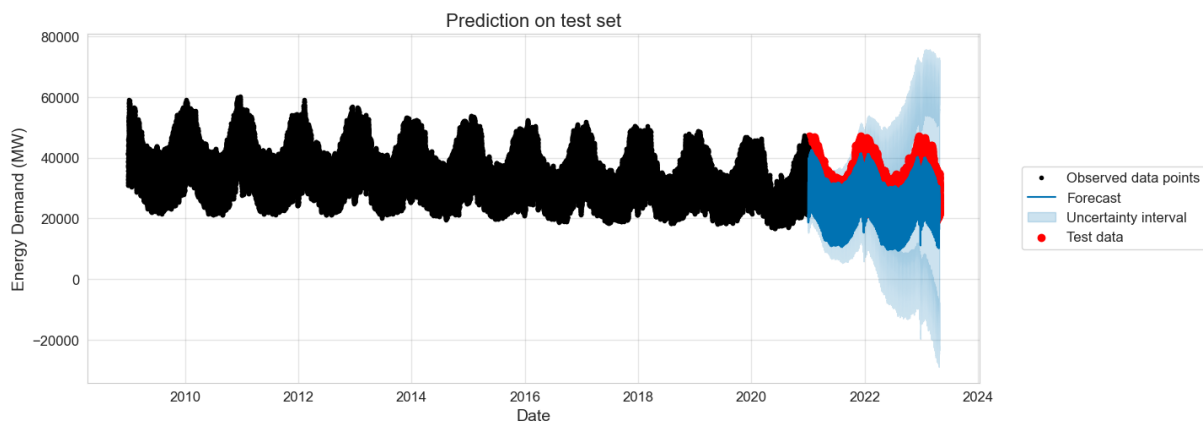
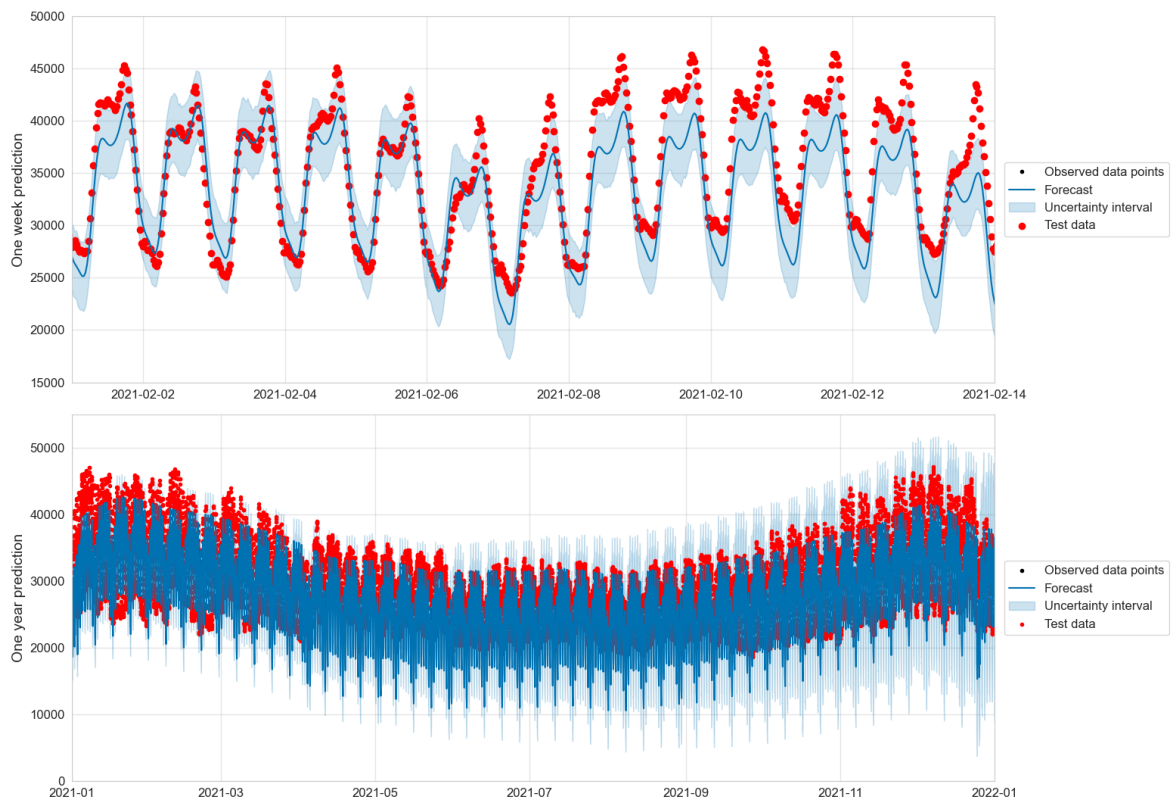


Figura 16: ...

El modelo presentado en esta sección tiene implementado los efectos de la vacaciones y gracias a la validación cruzada se ha obtenido los parámetros que minimizan en el error medio absoluto, que en este caso es 11,88 %. Hasta el momento, es el mejor modelo presentado para modelar los datos y su tiempo de computación es de 18 minutos y 23 segundos, siendo unos de los modelos que más tardan pero también el que mejor funciona.

La gran ventaja de este método es que está diseñado para que tanto analistas con grandes conocimientos en la ciencia de datos como personas no tan especializadas puedan ajustar los parámetros para tener predicciones de calidad. En este caso gracias a la validación cruzada se ha encontrado un modelo óptimo en términos de MAPE, sin embargo, con los parámetros prefijados en Python, también se obtienen buenas predicciones, aunque puedan tener menor calidad.



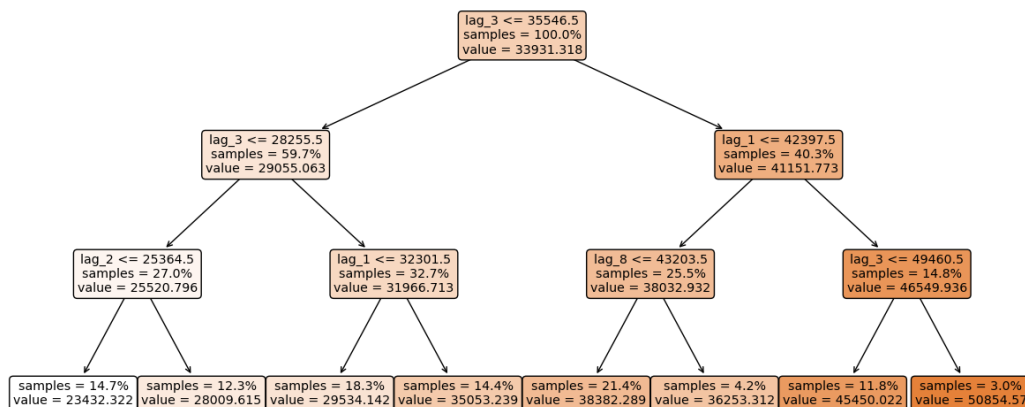


**Figura 17:** *Predicciones con el modelo Prophet*

## 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 18:** Árbol de regresión ...

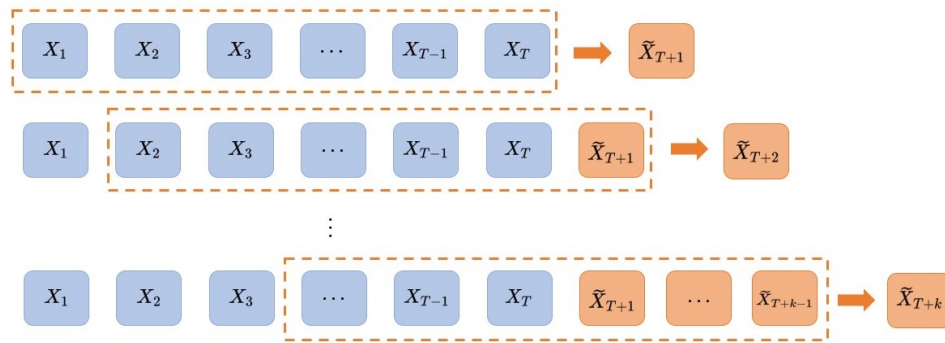
Este razonamiento se puede aplicar a las series temporales, a partir de un número de observaciones se puede pronosticar el siguiente valor de la series. Como las series temporales describen una variable continua, se utilizarán árboles de regresión en lugar de los árboles de decisión tradicionales.

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  $\{X_1, \dots, X_T\}$  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 en Python, consiste en utilizar los últimos  $n$  valores de la serie temporal para predecir el próximo,  $\tilde{X}_{T+1}$ . El siguiente paso utiliza los últimos  $n-1$  valores de la serie y el nuevo valor pronosticado  $\tilde{X}_{T+1}$  para predecir el valor de  $\tilde{X}_{T+2}$ . Así continua el algoritmo hasta obtener la predicción de  $X_{T+k}$ . Se puede ver de forma esquemática en la Figura



**Figura 19:** Esquema del método multipaso recursivo con  $n = T$

## 6.2. Método multipaso directo

Mientras el anterior algoritmo utilizaba las predicciones que él mismo generaba, este método consiste en crear un modelo para cada predicción. En este caso, si se quiere predecir el valor de  $X_{T+k}$ , se crea un modelo que tome  $n$  observaciones consecutivas de la serie temporal y prediga el valor para  $k$  paso hacia delante, como muestra la Figura 20. En el método anterior, se calculan todos los valores intermedios hasta  $X_{T+k}$ , pero en este modelo se obtiene directamente dicho valor. Si se requiere calcular todos los valores intermedios entonces es necesario entrenar  $k$  modelos distintos, lo que aumenta el coste computacional.

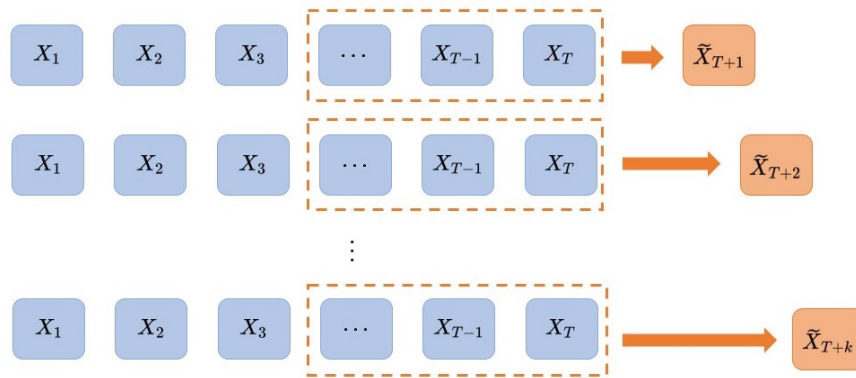


Figura 20: Esquema del método multipaso directo

## 7. Comparación de modelos

## 8. Conclusiones

## Referencias

- [1] Taylor SJ, Letham B. 2017. *Forecasting at scale*. PeerJ Preprints 5:e3190v2  
<https://doi.org/10.7287/peerj.preprints.3190v2>
- [2] Rafferty, G.,(2023). *Forecasting Time Series Data with Prophet*. Packt Publishing.
- [3] Korstanje, J., Advanced Forecasting with Python: With State-of-the-Art-Models Including LSTMs, Facebook's Prophet, and Amazon's DeepAR in Korstanje, J. (2021) pp.253-271.
- [4] Brockwell, P.J., Davis, R.A. (1987). *Time Series: Theory and Methods*. Springer.
- [5] Peña, D. (2010). *Análisis de series temporales*. Alianza Editorial.
- [6] Brockwell, P.J., Davis, R.A. (1996). *Introduction to Time Series and Forecasting*. Springer.
- [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*. Springer.

## A. Detalles del desarrollo del trabajo

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

**Tabla 6:** *Tiempo aproximado de dedicación al trabajo*

Asignatura	Páginas	Descripción
Series Temporales	...	-
Análisis de datos I	...	-
Análisis de datos II	...	-

**Tabla 7:** *Asignaturas relacionadas con el trabajo*