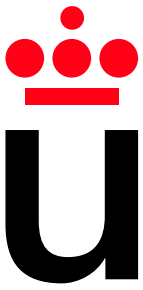


Universidad
Rey Juan Carlos

Escuela Técnica Superior
de Ingeniería Informática

Grado en Matemáticas

Curso 2022-2023

Trabajo Fin de Grado

PREDICCIÓN DEL GASTO EN CONSUMO UTILIZANDO LA TÉCNICA RANDOM FOREST

Autor: Ramón Pacheco Murillo
Tutor: Javier Martínez Moguerza

Agradecimientos

Mientras escribo las últimas líneas de este trabajo, no puedo dejar de pensar en todas aquellas personas que me han acompañado en este largo recorrido y que, sin saberlo, me han guiado hasta aquí.

Quisiera darle las gracias a mi familia, a mis padres y a mi hermano, por darme la oportunidad de formarme, por atenderme, cuidarme y quererme incondicionalmente, por inculcarme unos valores de los que estoy y estaré siempre orgulloso. En definitiva, gracias por ser los pilares de mi vida.

A mis amigos de la infancia, porque tengo la suerte de contar con ellos a pesar de que hemos crecido y tomado caminos diferentes en la vida. Me llena de alegría saber que seguiremos construyendo recuerdos juntos.

A todos los profesores que han tenido que aguantarme, en especial, a mi profesora de matemáticas del I.E.S Julio Verne, Rosa María Sebastián de Santos que, con su trabajo y entrega despertó en mí la curiosidad por las matemáticas. Gracias por haber creído en mí cuando ni siquiera yo lo hacía. Gracias, también, a mi tutor Javier por haber hecho posible este trabajo.

A mis compañeros del doble grado: Lucía Salgado, Alberto Gutiérrez, Marta Jiménez, Daniel Pulido, Amelia Arribas, Javier Alarcón, Marta Camacho y Víctor Calvo, porque sin ellos estos últimos años hubieran sido aún más insoportables.

Aunque me encantaría poder mencionar individualmente a todas aquellas personas que han sido parte de esta etapa y que, sin duda alguna también han dejado una huella significativa en mí, esa tarea es imposible pues, cada sonrisa compartida, cada gesto amable y cada palabra de aliento han permitido de alguna manera este momento. A todas ellas les deseo que sean felices porque, ¿qué más podría desearles yo a estas personas que tanto aprecio?

Ramón Pacheco Murillo

Resumen

En este trabajo, se utiliza un modelo Random Forest para realizar predicciones de la evolución del gasto en consumo final de los hogares e Instituciones Sin Fines de Lucro al Servicio de los Hogares (ISFLSH) a partir de la base de datos BDREMS [3].

Utilizando dos enfoques de entrenamiento distintos, uno que no permite la ampliación de los datos de entrenamiento según se avanza en el tiempo y, otro que sí permite su actualización, se discute el desempeño del modelo para predecir la serie temporal y se comparan los resultados obtenidos con cada estrategia. Además, se varía el parámetro del número de árboles del Random Forest para contrastar cómo afecta a las predicciones del modelo.

Antes de aplicar esta técnica y mostrar los resultados obtenidos, se realiza un breve estudio de los fundamentos del análisis de series temporales donde se aplica un modelo ARIMA a la serie del gasto en consumo final.

Servirá este estudio para comprender las bases del análisis de series temporales y aplicar un modelo basado en aprendizaje automático a la predicción de series temporales.

Palabras clave:

- Análisis de series temporales
- Random Forest
- Predicciones
- Modelo ARIMA
- Consumo
- Economía

Índice general

Capítulo 1

Introducción

1.1. Contexto y alcance

La capacidad de hacer predicciones precisas en economía es indispensable para la toma de decisiones de las personas, desde aquellas que debe tomar un pequeño empresario hasta las de los altos cargos de la Administración Pública. Existen ciertas variables que, por su gran importancia en el funcionamiento de la economía, requieren de estimaciones lo más precisas posible para poder informar a la sociedad de su posible evolución futura. Una de ellas es el consumo, ya que representa una gran parte del gasto total de los hogares y, por lo tanto, tiene un gran impacto en la producción, el empleo y el crecimiento económico. No obstante, predecir el consumo no es una tarea fácil pues en él influyen variables económicas, demográficas, psicológicas (a través del comportamiento de los consumidores) ... etc.

El desarrollo de modelos basados en Aprendizaje Automático y su capacidad para tratar eficientemente grandes cantidades de datos, propicia que se hayan aplicado en el campo de la economía. Sin embargo, cuando se aplican a series temporales se enfrentan a diversos desafíos [20]. Los valores de una serie temporal pueden depender de valores pasados, es decir, puede existir una dependencia temporal que el algoritmo de aprendizaje automático debe ser capaz de capturar para generar predicciones precisas. Además, una serie temporal puede presentar también variaciones estacionales (pensemos por ejemplo en un aumento en las ventas de un producto en determinadas épocas del año) que son difíciles de captar por los modelos o verse afectadas por factores externos (como la pandemia de COVID-19). Por otra parte, el entrenamiento de los modelos requiere de una gran cantidad de datos que, para algunas series temporales, no siempre están plenamente disponibles.

Bajo este enfoque, es interesante estudiar cómo se puede aplicar modelos basados en aprendizaje automático a la predicción de series temporales. En

particular, en este trabajo, se expone una metodología para aplicar la técnica de Random Forest a este tipo de datos y, se utiliza, para obtener predicciones de la evolución del gasto en consumo final de los hogares e Instituciones Sin Fines de Lucro al Servicio de los Hogares (ISFLSH). Con el objetivo de comprender mejor el problema con el que tratamos, antes de aplicar la técnica y presentar los resultados, se desarrolla un breve estudio de los fundamentos del análisis de series temporales donde se aplica un modelo $ARIMA$ a la serie del gasto en consumo final.

1.2. Estructura del documento

Tras introducir el contexto y el alcance del problema que nos ocupa, los contenidos principales de esta memoria se estructuran de la siguiente manera:

- En el capítulo 2, se desarrolla un breve estudio de los fundamentos del análisis de series temporales, incluyendo el desarrollo de un modelo $ARIMA$ para la serie temporal del gasto en consumo final de los hogares e Instituciones Sin Fines de Lucro al Servicio de los Hogares (ISFLSH).
- El capítulo 3 contiene la base teórica que permite comprender la técnica de Random Forest y su uso para la predicción de datos de serie temporal.
- En el capítulo 4 se explica detalladamente la metodología seguida para obtener predicciones de la evolución del gasto en consumo de los hogares e ISFLSH utilizando la técnica Random Forest. Además, se presentan los resultados del análisis y las comparaciones entre las distintas alternativas exploradas.
- Por último, el capítulo 5 recoge las conclusiones, limitaciones y trabajos futuros del estudio realizado.

1.3. Objetivos

El objetivo de este trabajo es utilizar la técnica de Random Forest para realizar predicciones de la evolución del gasto en consumo final de hogares e ISFLSH. A partir de las series temporales de un amplio conjunto de variables económicas, se desarrolla un modelo Random Forest que permite tratar con estos datos y predecir el gasto en consumo final de los hogares e ISFLSH. Los resultados del modelo nos permitirán discutir la idoneidad de la metodología aplicada y la precisión de las predicciones obtenidas.

A continuación, se exponen los objetivos más específicamente:

1. Comprender los conceptos básicos y métodos estadísticos tradicionales aplicados al estudio de series temporales.

2. Aplicar el Método de Box-Jenkins para analizar un caso real e implementar su desarrollo a través de código en *R*.

3. Revisar conceptos básicos sobre Aprendizaje Automático y árboles C.A.R.T que nos permitan entender la técnica de Random Forest.

4. Establecer una estrategia para poder aplicar la técnica de Random Forest a series temporales e implementarla para obtener predicciones de la evolución del gasto en consumo final de hogares e ISFLSH.

5. Presentar los resultados obtenidos con la ayuda de gráficas generadas con código en *R*.

1.4. Metodología

Para alcanzar los objetivos anteriormente expuestos, se sigue la siguiente metodología de investigación:

Primero, se realiza una revisión de los fundamentos del análisis de series temporales, centrándonos en métodos estacionarios lineales, que nos permite desarrollar e implementar en R un modelo ARIMA para simular el comportamiento del gasto en consumo final de los hogares e ISFLSH en caso de no haber ocurrido la pandemia de COVID-19. Además, gracias a este análisis podemos comprender los rasgos fundamentales del estudio de series temporales para poder adaptar mejor la técnica Random Forest a datos de este tipo.

A continuación, tras analizar los aspectos teóricos de la técnica Random Forest, se implementa un modelo Random Forest para predecir el gasto en consumo final de los hogares e ISFLSH a partir de la base de datos BDREMS. Utilizando dos enfoques de entrenamiento distintos, uno que no permite la ampliación de los datos de entrenamiento conforme se avanza en el horizonte temporal y, otro que sí permite su actualización hasta el periodo anterior al de la predicción, contrastamos:

- Si la técnica de Random Forest es capaz de capturar correctamente la tendencia temporal de los datos.
- Si el segundo enfoque de entrenamiento produce mejores resultados que el primero.

Para ello, se utilizan tres métricas de error - Raíz del error cuadrático medio (R.M.S.E), Error absoluto medio (M.A.E) y Error absoluto medio porcentual (M.A.P.E) – y se acompañan los resultados de representaciones gráficas de los ajustes y tablas. Por último, se varía uno de los parámetros del modelo – el número de árboles del Random Forest - para ver cómo afecta a su desempeño y se compara los resultados obtenidos con el modelo ARIMA que se desarrolla inicialmente.

Capítulo 2

Análisis de series temporales.

En este capítulo se introducen una serie de conceptos básicos cuya comprensión se hace necesaria antes de realizar cualquier análisis de series temporales. Partiendo de estas nociones, se realiza un breve estudio de los métodos estadísticos tradicionales aplicados al estudio de series temporales, en particular, nos centramos en los siguientes métodos estacionarios lineales: modelos autorregresivos, modelos de medias móviles, modelos mixtos, procesos integrados y procesos integrados estacionales multiplicativos.

A continuación, se procede a la explicación del método de Box-Jenkins y a su aplicación práctica. Se analiza los datos de consumo de los hogares e Instituciones sin fines de lucro al servicio de los hogares (ISFLSH) a través de un modelo $ARIMA$ y se propone una posible evolución del consumo final en ausencia pandemia de COVID-19 en base al modelo.

Evidentemente, este capítulo no pretende ser una revisión exhaustiva y con todo detalle de todos los métodos estadísticos tradicionales aplicados a las series temporales.[1] Se trata más bien de un breve estudio que nos permite entender qué son las series temporales, cuáles son sus componentes y cómo pueden pueden analizarse para intentar establecer predicciones atendiendo a la evolución histórica de la serie. Comprender bien estas cuestiones nos permitirá trasladar las características fundamentales del estudio de series temporales al método de aprendizaje automático que utilizaremos en el capítulo 4.

[1] Para un análisis más detallado pueden tomarse las referencias que han servido como bibliografía de este capítulo: [21],[18] y [11].

2.1. Conceptos básicos

Definición 1. *Un proceso estocástico es un conjunto de variables aleatorias* $\{y_t\}$ *donde el índice* t *toma valores de un cierto conjunto* $C = \{1, 2, ..., T\}$.

Definición 2. *Una serie temporal es una secuencia de observaciones* $(\hat{y_1}, \hat{y_2}, ..., \hat{y_T})$ *de un proceso estocástico, donde los elementos de* C *están ordenados y corresponden a instantes equidistantes del tiempo.*

Definición 3. *Sea* $\{y_t\}$ *un proceso estocástico, la función de medias del proceso viene dada por:*

$$E(y_t) = \mu_t, \qquad t = 1, 2, ..., T \tag{2.1}$$

Además, se dice que el proceso es estable en la media si todas las variables tienen la misma media:

$$E(y_t) = \alpha = cte, \qquad t = 1, 2, ..., T \tag{2.2}$$

Definición 4. *Sea* $\{y_t\}$ *un proceso estocástico, la función de varianzas del proceso viene dada por:*

$$Var(y_t) = \gamma, \qquad t = 1, 2, ..., T \tag{2.3}$$

Definición 5. *Sea* $\{y_t\}$ *un proceso estocástico, la función de autocovarianzas del proceso se define como:*

$$\gamma(t, t+i) = Cov(y_t, y_{t+i}) = E[(y_t - \mu_t)(y_{t+i} - \mu_{t+i})] \tag{2.4}$$

Además, se dice que el proceso es estable en autocovarianza si:

$$\gamma(t_1, t_1 + i) = \gamma(t_2, t_2 + i) = \gamma_i \quad \forall t_1, t_2 \in C, \quad \forall i \in \mathbb{Z} \tag{2.5}$$

Definición 6. *Sea* $\{y_t\}$ *un proceso estocástico, la función de autocorrelación simple del proceso se define como:*

$$\rho(t, t+i) = \frac{Cov(y_t, y_{t+i})}{\sigma_t \sigma_{t+i}} = \frac{\gamma(t, t+i)}{\gamma^{1/2}(t, t)\gamma^{1/2}(t+i, t+i)} \quad \forall i \in \mathbb{Z} \tag{2.6}$$

Definición 7. *Sea* $\{y_t\}$ *un proceso estocástico, el coeficiente de correlación parcial entre* y_t *y* y_{t-k} *se define como la relación lineal existente entre* y_t *y* y_{t-k} *obtenida de la siguiente manera:*

$$y_t = \beta_1 y_{t-1} + \beta_2 y_{t-2} + ... + \beta_{k-1} y_{t-(k-1)} + u_t \tag{2.7}$$

$$y_{t-k} = \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + ... + \alpha_{k-1} y_{t-(k-1)} + v_t \tag{2.8}$$

El coeficiente de correlación parcial de orden k - ϕ_{kk} - viene determinado por:

$$\phi_{kk} = \frac{Cov(u_t, v_t)}{\sqrt{Var(u_t)}\sqrt{Var(v_t)}} \tag{2.9}$$

Considerando ϕ_{kk}, $\forall k \in \mathbb{N}$ *obtenemos la función de autocorrelación parcial.*

Definición 8. *Sea $\{y_t\}$ un proceso estocástico, se dice estacionario en sentido estricto si la distribución conjunta de cualquier conjunto de variables no cambia al trasladar las variables en el tiempo, es decir:*

$$F(y_i, y_j, ..., y_k) = F(y_{i+h}, y_{j+h}, ..., y_{k+h}) \tag{2.10}$$

Definición 9. *Sea $\{y_t\}$ un proceso estocástico, se dice estacionario en sentido débil si es estable en media y en autocovarianza.*

Definición 10. *Se dice que un proceso estocástico $\{y_t\}$ es de ruido blanco si cumple:*

1. $E(y_t) = 0$
2. $Var(y_t) = \sigma^2$
3. $Cov(y_t, y_{t+k}) = 0, \quad k = \pm 1, \pm 2, ...$

Definición 11. *Sean $(\hat{y_1}, \hat{y_2}, ..., \hat{y_T})$ observaciones relativas a un proceso estocástico estacionario, entonces la media del proceso puede estimarse a través de la media muestral:*

$$\overline{y} = \frac{1}{T} \cdot \sum_{t=1}^{T} y_t \tag{2.11}$$

Definición 12. *Sean $(\hat{y_1}, \hat{y_2}, ..., \hat{y_T})$ observaciones relativas a un proceso estocástico estacionario, entonces la función de autocorrelación simple definida anteriormente puede estimarse a través de la función de autocorrelación muestral:*

$$r_k = \frac{\sum_{t=1}^{T}(y_t - \overline{y})(y_{t-k} - \overline{y})}{\sum_{t=1}^{T}(y_t - \overline{y})^2} \quad con \quad k = 1, 2, \tag{2.12}$$

En el caso de la función de autocorrelación parcial, podemos usar el estimador de Mínimos Cuadrados Ordinarios.

Hasta ahora nos hemos estado refiriendo a la distribución marginal de y_t, es decir, $f(y_t)$ sin incorporar la información anterior al momento t. No obstante, para el trabajo de predicción nuestro interés está en la distribución condicionada $f(y_t \mid y_{t-1}, y_{t-2}, ..., y_{t-k})$ donde conocemos los k valores anteriores del proceso.

2.2. Descomposición de series temporales. Modelos clásicos.

El análisis clásico de series temporales está basado en el modelo de descomposición donde las series temporales están constituidas por cuatro elementos básicos:

$$Y_t = f(C_t, T_t, S_t, E_t) \tag{2.13}$$

- Tendencia (T_t): es el comportamiento a largo plazo de la serie.
- Componente cíclica (C_t): son fluctuaciones a medio plazo en torno a la tendencia con un período y amplitud de cierta regularidad.
- Componente estacional (S_t): movimientos regulares de la serie que tienen una periodicidad inferior a un año.
- Componente irregular o ruido (E_t): variaciones de la serie que no responden a un comportamiento sistemático o regular. Constituye la parte que no es explicada por la tendencia, el ciclo y la estacionalidad.

Para poder separar estas cuatro componentes es necesario conocer la forma en que se relacionan entre ellas, algo que en la práctica, resulta imposible de averiguar con plena certeza. No obstante, podemos recurrir a los modelos de descomposición estacional de la serie generalmente admitidos:

1. Aditivo:
$$Y_t = C_t + T_t + S_t + E_t \tag{2.14}$$
2. Multiplicativo:
$$Y_t = C_t \cdot T_t \cdot S_t \cdot E_t \tag{2.15}$$

Es cierto que, en ocasiones, analizando preliminarmente la serie temporal podemos determinar el modelo a emplear. Por ejemplo, en las series temporales con una marcada tendencia, si al representar la serie temporal gráficamente observamos que la magnitud de las fluctuaciones de las serie no se ven afectadas por la tendencia, se puede emplear el esquema aditivo. Si por el contrario, la magnitud de las fluctuaciones aumentan al hacerlo la tendencia (y viceversa) entonces, se puede emplear el esquema multiplicativo.

2.3. Procesos no estacionarios

Muchas series temporales reales no presentarán estacionariedad debido a la existencia de variaciones estacionales, varianza no constante o tendencia. En algunos casos, será posible transformar la serie en otras que se comporten de manera similar a una serie estacionaria y así usar las herramientas de análisis existentes para estas últimas. En esta sección, se muestran algunos ejemplos de la transformación a seguir según la tendencia de la serie.

2.3.1. Procesos no estacionarios homogéneos

Tendencia lineal

Supongamos que la serie temporal $\hat{y}_t$ sigue una tendencia lineal de la siguiente manera:
$$\hat{y}_t = \alpha + \beta t + u_t \quad con \quad t = 1, 2, ..., T \tag{2.16}$$

donde u_t cumple las hipótesis del modelo clásico de regresión lineal. Entonces se tiene:

$$\begin{aligned} \hat{y_t} &= \alpha + \beta t + u_t \\ \hat{y_{t-1}} &= \alpha + \beta(t-1) + u_{t-1} \end{aligned} \tag{2.17}$$

Restando ambas expresiones obtenemos:

$$\hat{y_t} - \hat{y_{t-1}} = \beta + u_t - u_{t-1} \tag{2.18}$$

De donde podemos deducir que, aplicando la primera diferencia de la serie: $Z_t \equiv \hat{y_t} - \hat{y_{t-1}}$ eliminamos la tendencia lineal.

Tendencia polinómica

Consideremos la siguiente función polinómica de grado p que representa la tendencia polinómica:

$$f(t) = \alpha + \beta_1 t + \beta_2 t^2 + \ldots + \beta_p t^p \tag{2.19}$$

Supongamos ahora que la serie temporal presenta la tendencia polinómica más sencilla, es decir, tendencia cuadrática:

$$\hat{y_t} = \alpha + \beta_1 t + \beta_2 t^2 + u_t \tag{2.20}$$

Aplicando la primera diferencia se tiene:

$$Z_t = \hat{y_t} - \hat{y_{t-1}} = \beta_1 - \beta_2 + 2\beta_2 t + u_t - u_{t-1} \tag{2.21}$$

Como la expresión anterior es una expresión lineal en t, aplicando la segunda diferencia eliminamos la tendencia:

$$Z_t - Z_{t-1} = 2\beta_2 + u_t - 2u_{t-1} + u_{t-2} \tag{2.22}$$

En general, si una serie temporal presenta una tendencia polinómica de grado p, eliminamos la tendencia tomando diferencias sucesivas de la serie p veces. Si un proceso no estacionario puede ser transformado en estacionario después de k operaciones de diferencias, se dice que es *homogéneo - o integrado - de orden k.*

Tendencia exponencial

Supongamos que la serie temporal sigue una tendencia exponencial de la siguiente forma:

$$\hat{y_t} = a \cdot e^{rt} \cdot e^{u_t} \quad con \quad t = 1, 2, ..., T \tag{2.23}$$

En este caso tenemos:

$$\begin{aligned} \hat{y_t} &= a \cdot e^{rt} \cdot e^{u_t} \\ \hat{y_{t-1}} &= a \cdot e^{r(t-1)} \cdot e^{u_{t-1}} \end{aligned} \tag{2.24}$$

Dividiendo la primera expresión entre la segunda obtenemos:

$$\frac{\hat{y_t}}{\hat{y_{t-1}}} = \frac{a \cdot e^{rt} \cdot e^{u_t}}{a \cdot e^{r(t-1)} \cdot e^{u_{t-1}}} \tag{2.25}$$

$$\frac{\hat{y_t}}{\hat{y_{t-1}}} = e^{r} \cdot e^{u_t - u_{t-1}} \tag{2.26}$$

De donde tomando logaritmos se tiene:

$$\ln \frac{\hat{y_t}}{\hat{y_{t-1}}} = r + u_t - u_{t-1} \tag{2.27}$$

$$\ln \hat{y_t} - \ln \hat{y_{t-1}} = r + u_t - u_{t-1} \tag{2.28}$$

Y por tanto, podemos concluir que, $Z_t \equiv \ln \hat{y_t} - \ln \hat{y_{t-1}}$ elimina la tendencia exponencial.

2.3.2. Corrección de las variaciones estacionales

Para eliminar las variables estacionales, primero debemos eliminar la tendencia de la serie y, a continuación, recurrir a un proceso denominado diferenciación estacional. Basándonos en el modelo clásico aditivo de descomposición, operamos según la frecuencia de los datos:

1. Si los datos son mensuales:

$$Z_t = y_t - y_{t-12} \tag{2.29}$$

2. Si los datos son trimestrales:

$$Z_t = y_t - y_{t-4} \tag{2.30}$$

3. Si los datos son cada n-meses ($n < 12$):

$$Z_t = y_t - y_{t-\frac{12}{n}} \tag{2.31}$$

Si tras aplicar el procedimiento, las variaciones estacionales persisten, podemos aplicar las diferencias sucesivas que sean necesarias para eliminarlas.

2.4. Modelos estacionarios lineales

Más adelante estudiaremos el enfoque de Box-Jenkins para analizar y predecir series temporales estacionarias aplicando ciertos modelos. Antes de hacerlo, necesitamos introducir el modelo autorregresivo, el de medias móviles, el modelo mixto y los procesos integrados.

2.4.1. Modelo autorregresivo de orden p : $AR(p)$

Para aligerar la notación usada en este apartado introducimos el operador de retardo B definido sobre la variable aleatoria y_t como:

$$B \cdot y_t = y_{t-1} \tag{2.32}$$

Definición 13. *Un proceso estocástico $\{y_t\}$ describe un modelo autorregresivo de orden p si:*

$$y_t = \delta + \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t \tag{2.33}$$

Con δ constante, $\varepsilon_t \sim N(0, \sigma^2)$. Utilizando el operador de retardos se tiene:

$$\phi(B) \cdot y_t = \delta + \varepsilon_t \tag{2.34}$$

$$con \quad \phi(B) = 1 - \sum_{i=1}^{p} \phi_i B^i \tag{2.35}$$

Sabemos que un proceso estacionario cumple:

$$E(y_t) = \mu = cte < \infty \qquad con \quad t = 1, 2, ..., T \tag{2.36}$$

$$\gamma(t, t+k) = E[(y_t - \mu)(y_{t+k} - \mu_{t+k})] = \gamma_k < \infty \tag{2.37}$$

Tomando esperanzas en la expresión del $AR(p)$ se tiene:

$$\mu = \delta + \sum_{i=1}^{p} \phi_i \mu \tag{2.38}$$

$$\mu = \frac{\delta}{1 - \sum_{i=1}^{p} \phi_i} \tag{2.39}$$

Y por tanto, de la última ecuación se deduce que si el proceso es estacionario, entonces:

$$1 - \sum_{i=1}^{p} \phi_i \neq 0 \tag{2.40}$$

Esta condición, sin embargo, es necesaria pero no suficiente. La condición necesaria y suficiente de estacionariedad de $AR(p)$ requiere que las raíces de la ecuación $\phi(B) = 0$ estén fuera del círculo de radio unitario, en otras palabras, deben tener módulos superiores a la unidad.

Las características más relevantes de este modelo son:

1. El modelo $AR(p)$ siempre es invertible.

2. Condición de estacionariedad: el modelo será estacionario si las raíces de $\phi(B)$ se encuentran fuera del círculo de radio unitario.
3. Ni las autocovarianzas ni las autocorrelaciones se anulan.
4. Si representamos la función de autocorrelación simple podremos apreciar infinitos valores con una tendencia amortiguada.
5. La autocorrelaciones parciales se anulan cuando consideramos un desfase temporal mayor al orden del modelo.

Ejemplo modelo autorregresivo de orden 2: $AR(2)$

Utilizando la función `arima.sim(.)` del paquete `stats`, podemos generar un conjunto de observaciones que sigan el siguiente modelo autorregresivo:

$$y_t = 0{,}5 \cdot y_{t-1} + 0{,}45 \cdot y_{t-2} + \varepsilon_t \tag{2.41}$$

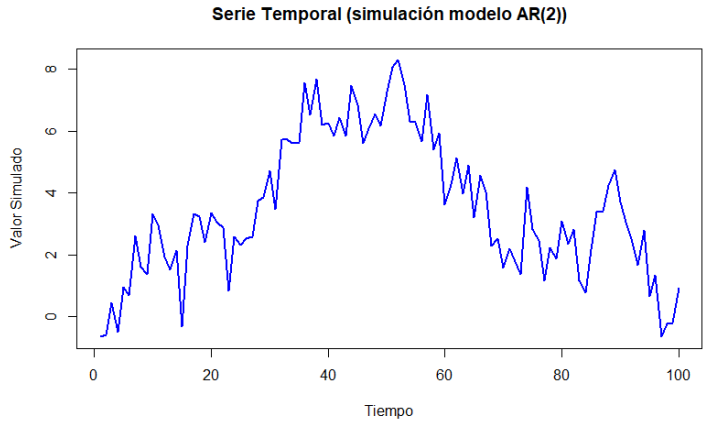


Figura 2.1: Ejemplo modelo autorregresivo de orden 2: $AR(2)$

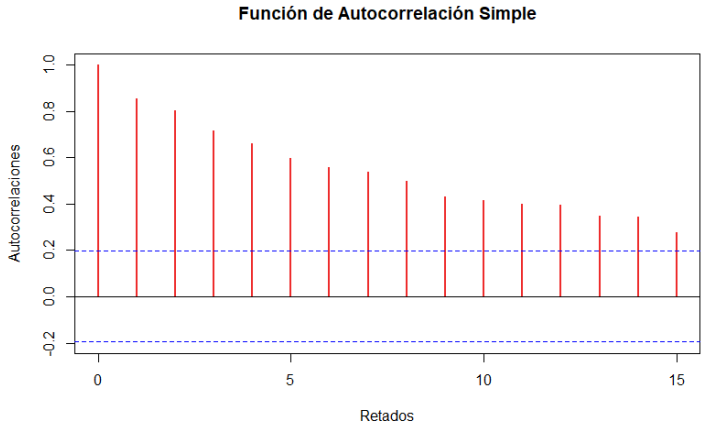


Figura 2.2: F.A.S del modelo autorregresivo de orden 2: $AR(2)$

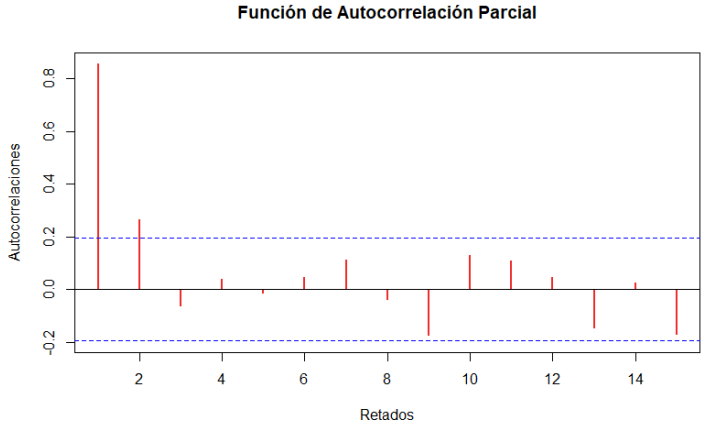


Figura 2.3: F.A.P del modelo autorregresivo de orden 2: $AR(2)$

2.4.2. Modelo de medias móviles de orden q : $MA(q)$

Definición 14. *Sea $\{y_t\}$ un proceso estocástico. El modelo de medias móviles de orden $q - MA(q)-$ se define como:*

$$y_t = \mu - \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t \tag{2.42}$$

Con θ_i tal que $i \in \{1...q\}$, μ constante, $\varepsilon_t \sim N(0, \sigma^2)$. Utilizando el operador de retardos se tiene:

$$y_t = \mu + \theta(B) \cdot \varepsilon_t \tag{2.43}$$

$$con \quad \theta(B) = 1 - \sum_{i=1}^{q} \theta_i B^i \tag{2.44}$$

Tomando esperanzas en la expresión anterior:

$$E(y_t) = \mu \tag{2.45}$$

Por tanto, la varianza del proceso quedará determinada por:

$$\begin{aligned} Var[y_t] &= Var[\mu - \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t] = Var[\mu] + Var[\sum_{i=1}^{q} \theta_i \varepsilon_{t-i}] + Var[\varepsilon_t] = \\ &= Var[\sum_{i=1}^{q} \theta_i \varepsilon_{t-i}] + Var[\varepsilon_t] = \sum_{i=1}^{q} \theta_i^2 Var[\varepsilon_{t-i}] + Var[\varepsilon_t] = \\ &= \sigma^2(1 + \theta_1^2 + \theta_2^2 + ... + \theta_q^2) \end{aligned} \tag{2.46}$$

como $\sum_{i=1}^{q} \theta_i^2$ es convergente, por ser q finito, entonces se cumplen las condiciones de estacionariedad.

Las características más importantes de este modelo son:

1. El modelo $MA(q)$ siempre es estacionario.
2. Condición de invertibilidad: el modelo será invertible si las raíces de $\theta(B)$ se encuentra fuera del círculo de radio unitario.
3. La memoria del proceso coincide con su orden, ya que las autocovarianzas se anulan caundo consideramos un desfase temporal mayor al orden del modelo.
4. La autocorrelaciones paraciales presentan un comportamiento que tiende a cero.

Ejemplo modelo media móvil de orden 2: $MA(2)$

Utilizando la función `arima.sim(.)` del paquete `stats`, podemos generar un conjunto de 100 observaciones que siguen el siguiente modelo de media móvil:

$$y_t = -0{,}7 \cdot \varepsilon_{t-1} + 0{,}2 \cdot \varepsilon_{t-2} + \varepsilon_t \tag{2.47}$$

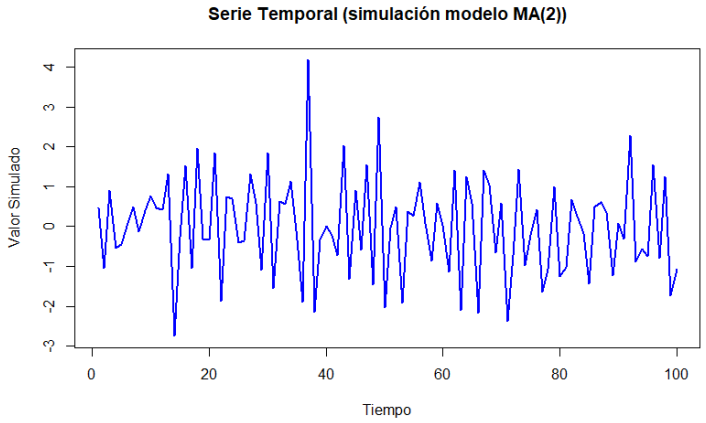


Figura 2.4: Ejemplo modelo media móvil de orden 2: $MA(2)$

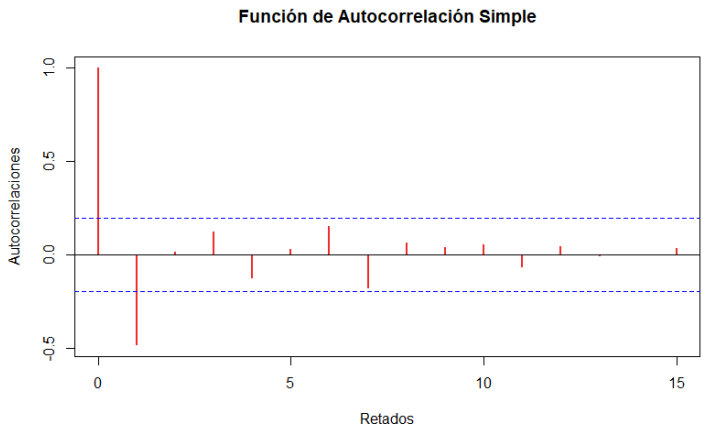


Figura 2.5: F.A.S modelo media móvil de orden 2: $MA(2)$

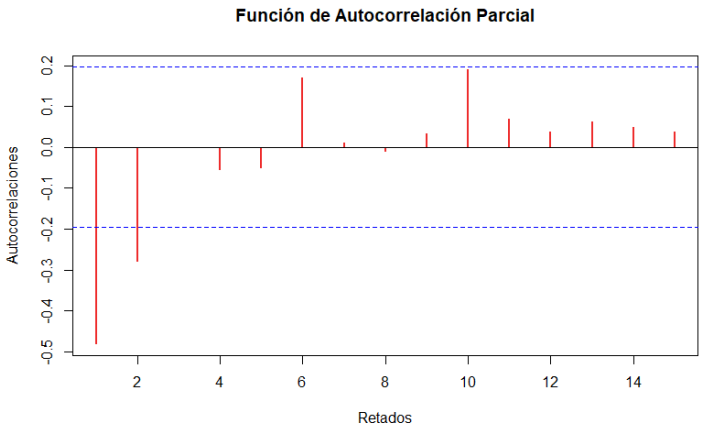


Figura 2.6: F.A.P modelo media móvil de orden 2: $MA(2)$

2.4.3. Modelo mixto: $ARMA(p, q)$

Definición 15. *Sea $\{y_t\}$ un proceso estocástico. El modelo autorregresivo de medias móviles $ARMA(p, q)$ se define como:*

$$y_t = \delta + \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} \tag{2.48}$$

Con δ constante, $\varepsilon_{t-j} \sim N(0, \sigma^2)$.

Tomando esperanzas en la expresión anterior:

$$\mu = \delta + \sum_{i=1}^{p} \phi_i \mu \tag{2.49}$$

$$\mu = \frac{\delta}{1 - \sum_{i=1}^{p} \phi_i} \tag{2.50}$$

Y por tanto, de la última ecuación se deduce que si el proceso es estacionario, entonces:

$$1 - \sum_{i=1}^{p} \phi_i \neq 0 \tag{2.51}$$

Utilizando el operador de retardos y suponiendo que la media es nula, consideremos:

$$\begin{aligned} \phi(B) &= 1 - \sum_{i=1}^{p} \phi_i B^i \\ \theta(B) &= 1 - \sum_{j=1}^{q} \theta_j B^j \end{aligned} \tag{2.52}$$

Por tanto, el modelo puede escribirse como:

$$y_t = \frac{\theta(B)}{\phi(B)} \cdot \varepsilon_t; \quad \frac{\phi(B)}{\theta(B)} \cdot y_t = \varepsilon_t; \tag{2.53}$$

Del procedimiento anterior podemos extraer las siguientes conclusiones:

1. Un $ARMA(p, q)$ que es estacionario, es equivalente a un proceso de medias móviles $MA(\infty)$ con $p+q$ coeficientes independientes - parte izquierda de la ecuación. Recordemos que la condición de estacionariedad implica que las raíces de $\phi(B) = 0$ deben encontrarse fuera del círculo de radio unitario.

2. Un $ARMA(p, q)$ que es invertible, puede expresarse como un modelo autorregresivo $AR(\infty)$ con $p+q$ coeficientes independientes - parte derecha de la ecuación. Recordemos que la condición de invertibilidad implica que las raíces de $\theta(B) = 0$ deben encontrarse fuera del círculo de radio unitario.

3. La función de autocorrelaciones parciales no se anulará - aunque tiende a cero - puesto que el modelo contiene al modelo de medias móviles como un caso especial.

Ejemplo modelo mixto: $ARMA(1, 1)$

Utilizando la función `arima.sim(.)` del paquete `stats`, generamos un conjunto de 100 observaciones que siguen el siguiente modelo mixto:

$$y_t = 0{,}7 \cdot y_{t-1} + 0{,}35 \cdot \varepsilon_{t-1} + \varepsilon_t \tag{2.54}$$

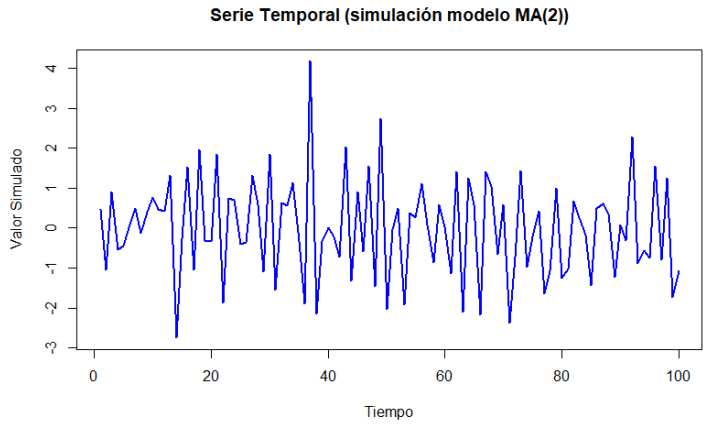


Figura 2.7: Ejemplo modelo mixto: $ARMA(1, 1)$

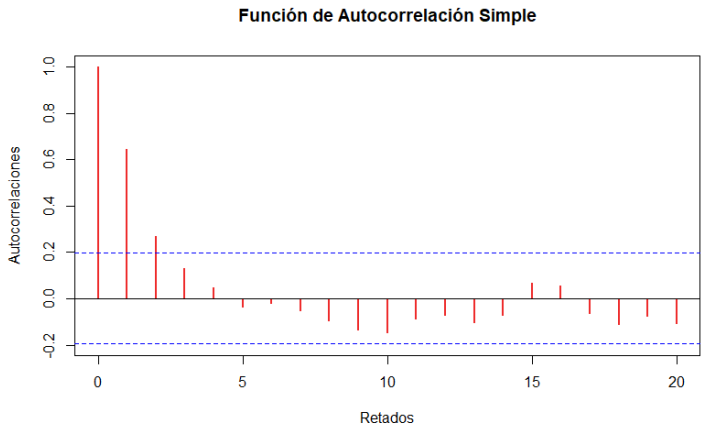


Figura 2.8: F.A.S modelo mixto: $ARMA(1,1)$

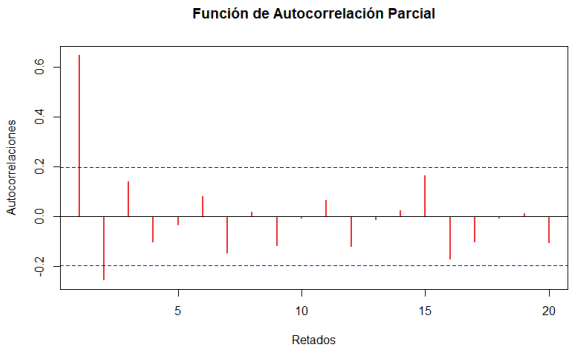


Figura 2.9: F.A.P modelo mixto: $ARMA(1,1)$

2.4.4. Procesos integrados: $ARIMA(p,d,q)$

Definición 16. *Sea* $\{y_t\}$ *un proceso estocástico. El modelo* $ARIMA(p,d,q)$ *sigue la ecuación:*

$$\begin{aligned}(1-\sum_{i=1}^{p}\phi_i B^i)\cdot(1-B)^d y_t &= \delta + (1-\sum_{j=1}^{q}\theta_j B^j)\cdot\varepsilon_t \\ \phi(B)\cdot\nabla^d y_t &= \delta + \theta(B)\cdot\varepsilon_t\end{aligned} \tag{2.55}$$

donde:

- p *es el orden de la parte autorregresiva estacionaria*
- d *es el orden de integración del proceso*

- q *es el orden de la parte media móvil*
- $\varepsilon_t \sim N(0, \sigma^2)$

En presencia de estacionalidad, podemos incorporarla de manera multiplicativa en el modelo anterior:

2.4.5. Proceso integrado estacional multiplicativo: $SARIMA(p, d, q) \times (P, D, Q)_s$

Definición 17. *Sea* $\{y_t\}$ *un proceso estocástico. El modelo* $SARIMA(p, d, q) \times (P, D, Q)_s$ *sigue la ecuación:*

$$(1 - \sum_{k=1}^{P} \Phi_k B^{s \cdot k}) \cdot (1 - \sum_{i=1}^{p} \phi_i B^i) \cdot (1 - B^s)^D \cdot (1 - B)^d \cdot y_t = (1 - \sum_{l=1}^{Q} \Theta_l B^{s \cdot l}) \cdot (1 - \sum_{j=1}^{q} \theta_j B^j) \cdot \varepsilon_t$$

$$\Phi_P(B^s) \cdot \phi_p(B) \cdot \nabla_s^D \cdot \nabla^d \cdot y_t = \Theta_Q(B^s) \cdot \theta_q(B) \cdot \varepsilon_t \tag{2.56}$$

donde:

- $\Phi_P(B^s)$ *es el operador autorregresivo estacionario de orden* P.
- $\phi_p(B)$ *es el operador autorregresivo regular de orden* p.
- $\Theta_Q(B^s)$ *es el operador media móvil estacional de orden* Q.
- $\phi_p(B)$ *es el operador media móvil regular de orden* q.
- $\nabla_s^D = (1 - B^s)^D$ *representa las diferencias estacionales.*
- $\nabla^d = (1 - B)^d$ *representa las diferencias regulares.*
- $\varepsilon_t \sim N(0, \sigma^2)$

Para decidir si los datos observados siguen un modelo $ARIMA$ o $SARIMA$, podemos aplicar el método de Box-Jenkins que se explica a continuación.

2.4.6. Método de Box-Jenkins

El método de Box-Jenkins es una técnica de modelado y predicción de series temporales. Basado en 4 pasos, permite identificar el modelo y los parámetros que debemos seguir para analizar los datos observados [19].

1. **Identificación del modelo:** El primer paso consiste en analizar la estructura de la serie temporal para intentar identificar el modelo más adecuado para describir los datos. Para ello, utilizamos mecanismos como las gráficas de las funciones de autocorrelación y autocorrelación parcial, la posible existencia de tendencia, de componente estacional...etc. Si los datos presentan tanto componente estacional como tendencia, un modelo $SARIMA(p, d, q) \times (P, D, Q)_s$ será más adecuado. Si solamente presentan tendencia, se puede usar el modelo $ARIMA(p, d, q)$.

2. **Identificación de los órdenes del modelo:** Tras analizar las características de la serie temporal, establecemos uno de los dos modelos y, procedemos a identificar los órdenes, es decir, los valores de (p, d, q) en el caso de un $ARIMA$ y los de $(p, d, q); (P, D, Q)$ en el caso del $SARIMA$.

   Primero, para fijar los valores de d o D, procedemos iterativamente a diferenciar $(\nabla \cdot y_t = (1 - B) \cdot y_t)$ tantas veces como sea necesario hasta que la varianza sea menor a la varianza calculada en el modelo inicial.

   A continuación, inspeccionamos las funciones de autocorrelación muestral y parcial de la serie temporal diferenciada en los diferentes retardos 1,2,3,... para obtener los valores p y q. En el caso de presencia de estacionalidad, debemos analizar también los retardos de $s, s \cdot 2, s \cdot 3, ...$ para establecer los valores de P y Q.

   También podemos fijar estos valores usando criterios de información. Por ejemplo, usando el Criterio de Información de Akaike (AIC), que viene dado por:

   $$AIC = -2 \cdot ln(L) + (ln(n) + 1) \cdot k \tag{2.57}$$

   donde $ln(L)$ es el logaritmo neperiano de la verosimilitud del modelo, n es el número de observaciones y k es el número de parámetros. Bajo este criterio, seleccionaremos aquel cuyo AIC sea más bajo.

3. **Estimación de los coeficientes del modelo:** Tras la asignación de los órdenes, estimamos los coeficientes maximizando la función de verosimilitud obteniendo así aquellos parámetros del modelo que maximizan la probabilidad de observar los datos disponibles. Podemos maximizar la función de verosimilitud mediante los estimadores de mínimos cuadrados no lineales, que podemos hallar usando el algoritmo de Levenberg-Marquardt.

4. **Validación del modelo ajustado:** Para valorar el modelo que hemos ajustado, debemos comprobar que se cumplen los requisitos que acompañan a su especificación. En la práctica, esto implica analizar que se cumplen las hipótesis relativas a los coeficientes y residuos estimados. Además, también es recomendable comprobar si deberíamos incluir parámetros adicionales (a través del método de sobreajuste) o, por el contrario, valorar la eliminación de los parámetros poco significativos [18].

2.5. Aplicación práctica: Estimación del consumo final.

El Instituto Nacional de Estadística proporciona datos trimestrales del gasto en consumo final para la elaboración del Producto Interior Bruto a través de la vía de la demanda. Accediendo a las series de resultados detallados, obtenemos los datos del gasto en consumo final para el periodo 1995-2022 expresados

en millones de euros. Este periodo incluye los primeros trimestres de la pandemia del coronavirus COVID-19. Representando los datos gráficamente, podemos apreciar una fuerte caída en el consumo que coincide con la implantación del confinamiento:

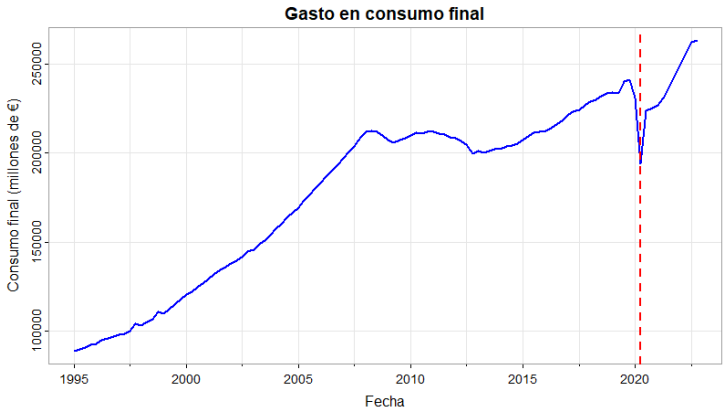


Figura 2.10: Gasto en consumo final (1995-2022). Fuente: I.N.E

Asumiendo que no hubiera existido otro cambio sustancial en ausencia del coronavirus, ¿cómo hubiera evolucionado el consumo final si no hubiera existido el confinamiento? Para responder a esta pregunta, planteamos un modelo sencillo con ayuda del software estadístico *R*.

Podríamos utilizar estos datos del I.N.E, pero dado que en una sección posterior usaremos los datos de consumo final de los hogares e Instituciones sin fines de lucro al servicio de los hogares procedentes de la base de datos $BDREMS$, para mantener la consistencia, usaremos los de esta última. Las variables que contiene la base de datos proceden de fuentes estadísticas oficiales y, por tanto, nuestro análisis tampoco diferirá demasiado del que resultaría de haber usado directamente los datos del I.N.E. Así, obtenemos datos del consumo final de los hogares e Instituciones sin fines de lucro al servicio de los hogares desde el primer trimestre de 1980 hasta el último trimestre del año 2020.

En esta sección, se analiza la estacionariedad de la serie y se realizan las modificaciones oportunas para lograr su estacionariedad. A continuación, se ajusta un modelo $ARIMA$ sobre los datos para obtener una posible evolución del consumo en ausencia del COVID-19 y se valora su desempeño.

Análisis de estacionariedad.

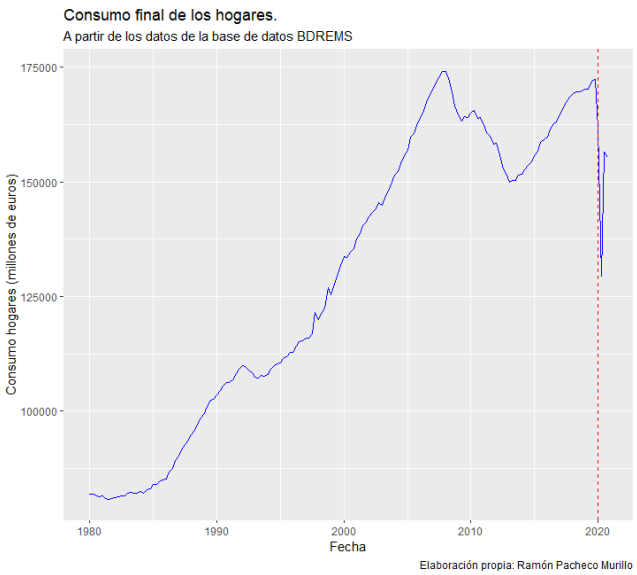


Figura 2.11: Gasto en consumo final (1980-2020). Fuente: BDREMS a partir de estadísticas oficiales.

En la representación gráfica de nuestros datos no se observa un comportamiento estable alrededor de un valor constante, lo que induce a pensar que la serie no es estacionaria. A continuación, se muestran las funciones de autocorrelación simple y autocorrelación parcial para nuestra serie:

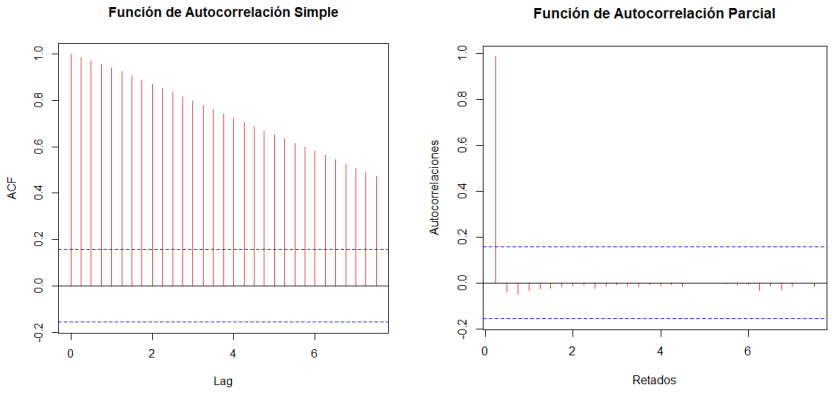


Figura 2.12: Funciones de autocorrelación.

En la figura 2.14, podemos apreciar que los coeficientes son positivos y decrecen lentamente hacia el cero, indicando que es posible que se tenga que tomar diferencias en la serie para convertirla en estacionaria. Para contrastar si nuestra intuición es cierta, aplicamos el Test de Dickey-Fuller aumentado a nuestros datos, obteniendo el siguiente resultado:

```
> #Test de dickey-fuller
> adf.test(serieTraining)

Augmented Dickey-Fuller Test

data:  serieTraining
Dickey-Fuller = -1.9913, Lag order = 5, p-value = 0.58
alternative hypothesis: stationary
```

La función utilizada en el software R toma como hipótesis nula H_0: *la serie no es estacionaria*. Dado que el contraste muestra un $p\text{-}valor = 0{,}58 > 0{,}05$ no se rechaza la hipótesis nula y no se puede afirmar que es estacionaria.

Aplicamos la primera diferencia sobre nuestros datos para tratar de corregir el posible problema de no estacionariedad:

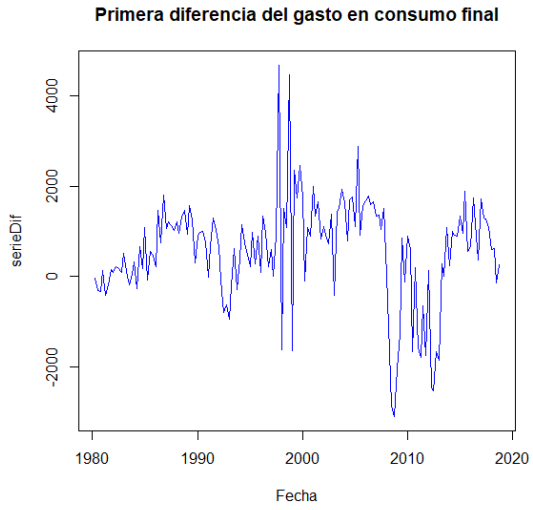


Figura 2.13: Primera diferencia para el gasto en consumo final

A continuación, se muestran las funciones de autocorrelación simple y autocorrelación parcial para nuestra serie:

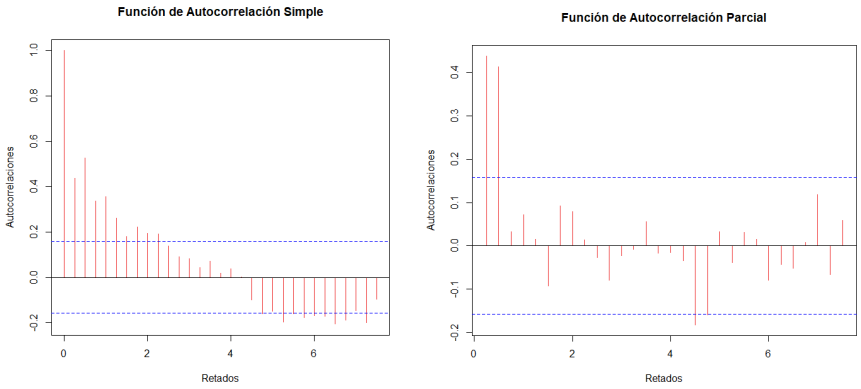


Figura 2.14: Funciones de autocorrelación.

Recurrimos nuevamente al Test de Dickey-Fuller aumentado para contrastar la hipótesis de estacionariedad:

```
> adf.test(serieDif)

Augmented Dickey-Fuller Test

data:  serieDif
Dickey-Fuller = -3.4305, Lag order = 5, p-value = 0.05205
alternative hypothesis: stationary
```

Aunque en este caso, el *p-valor* se acerca mucho al nivel al que rechazaríamos H_0, se sigue cumpliendo $= 0,005205 > 0{,}05$ y, por tanto, no se rechaza la hipótesis nula. Procedemos entonces a aplicar la segunda diferencia sobre la serie:

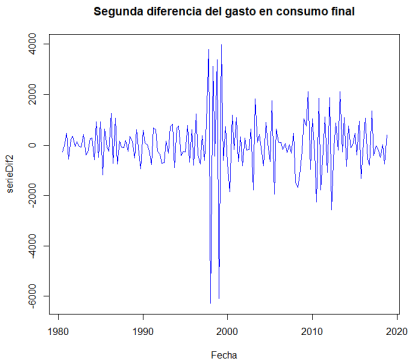


Figura 2.15: Segunda diferencia para el gasto en consumo final

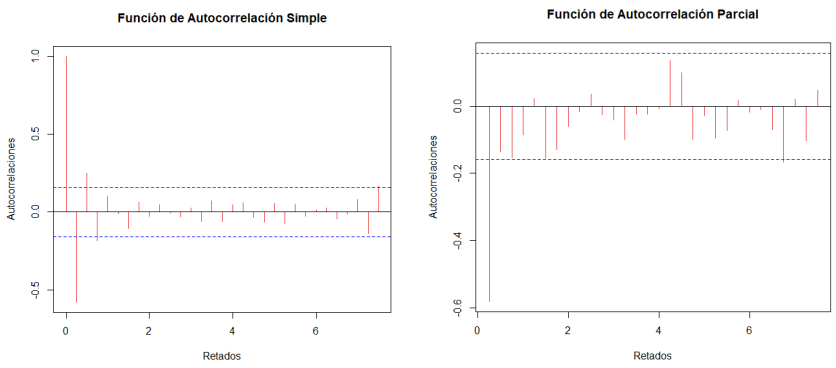


Figura 2.16: Funciones de autocorrelación.

Los resultados del Test de Dickey-Fuller aumentado para la segunda diferencia son los siguientes:

```
> adf.test(serieDif2)

Augmented Dickey-Fuller Test

data:  serieDif2
Dickey-Fuller = -6.7302, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(serieDif2) : p-value smaller than printed p-value
```

En este caso, se tiene un *p-valor* $= 0,01 < 0{,}05$ y, por tanto, rechazamos H_0 al 5 % de nivel de significación en favor de la hipótesis alternativa H_1: *la serie es estacionaria.*

Ajuste modelo ARIMA.

Para establecer el modelo adecuado, utilizaremos la función `auto.arima(x)` del paquete `forecast` en *R* [14]. Introduciendo los datos como argumento de la función, obtenemos el modelo *ARIMA* que mejor se ajusta a nuestros datos. En nuestro caso, es un $ARIMA(2,1,0)$:

```
> modelo_uno<-auto.arima(serieTraining)
> summary(modelo_uno)
Series: serieTraining
ARIMA(2,1,0) with drift

Coefficients:
```

```
         ar1     ar2     drift
      0.2549  0.4129  543.5491
s.e.  0.0724  0.0726  223.7536

sigma^2 = 908774:  log likelihood = -1282
AIC=2571.99   AICc=2572.26   BIC=2584.17

Training set error measures:
                    ME     RMSE      MAE          MPE      MAPE      MASE         ACF1
Training set 6.876334 940.9953 623.3039 0.008456465 0.4837478 0.1689025 -0.01393289
```

Es decir:

$$\begin{aligned}(1-\sum_{i=1}^{2}\phi_i B^i)\cdot(1-B)y_t &= \delta+\varepsilon_t \\ \phi(B)\cdot\nabla y_t &= \delta+\varepsilon_t\end{aligned} \tag{2.58}$$

Para comprobar si los residuos del modelo que nos sugiere la función son de ruido blanco, ejecutamos el test de Ljung-Box.

En este caso, tenemos como hipótesis nula - H_0 : *los residuos son ruido blanco*. A continuación, se muestran los resultados obtenidos por la función `Box.text(residuos,type = "Ljung-Box")` aplicada a nuestros datos:

```
modelo_grafica<-sarima(serieTraining,2,1,0)
> Box.test(residuals(modelo_grafica$fit),type = "Ljung-Box")

Box-Ljung test

data:  residuals(modelo_grafica$fit)
X-squared = 0.03087, df = 1, p-value = 0.8605
```

Se tiene un *p-valor* $= 0,8605 >$ 0,05 luego no rechazamos H_0 y, como la cifra del *p-valor* es muy alta, es muy posible que los residuos del modelo sí sean de ruido blanco. Para concluir este contraste utilizamos la función `sarima(datos,p,d,q)`.

La función genera automáticamente una serie de gráficas 2.17 que nos permiten profundizar y concluir el último contraste:

```
modelo_grafica<-sarima(serieTraining,2,1,0)
```

- El primer gráfico - *Standarized residuals* - muestra que los residuos se comportan de manera independiente alrededor del cero.
- En el gráfico de la función de autocorrelación simple - *ACF of Residuals* - todos los valores aparecen dentro de las bandas de significación (las líneas discontinuas azules).

- El gráfico cuantil-cuantil - *Normal Q-Q Plot of Std Residuals* - nos permite verificar si la distribución de los residuos del modelo siguen una distribución normal. Podemos apreciar que, aproximadamente, los puntos aparecen alineados y por tanto, siguen una distribución normal.
- Los p-valores del test de ruido blanco de Ljung-Box - *p values for Ljung-Box statistic* - muestran que no se rechaza la hipótesis nula - H_0 : *los residuos del modelo son ruido blanco* a ninguno de los niveles de significación habituales.

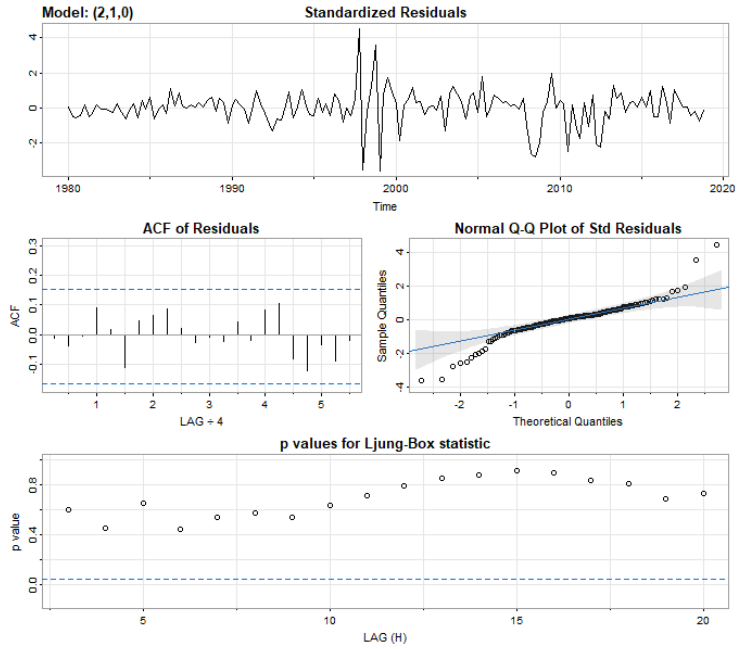


Figura 2.17: Gráficas - sarima(serieTraining,2,1,0)

Antes de proceder con las predicciones, tenemos que comprobar que el ajuste del modelo $ARIMA(2,1,0)$ a los datos de entrenamiento y test es óptimo pues, si no lo es, siguiendo la metodología de Box-Jenkins se debería realizar un análisis más profundo de la serie temporal para proponer un modelo más adecuado.

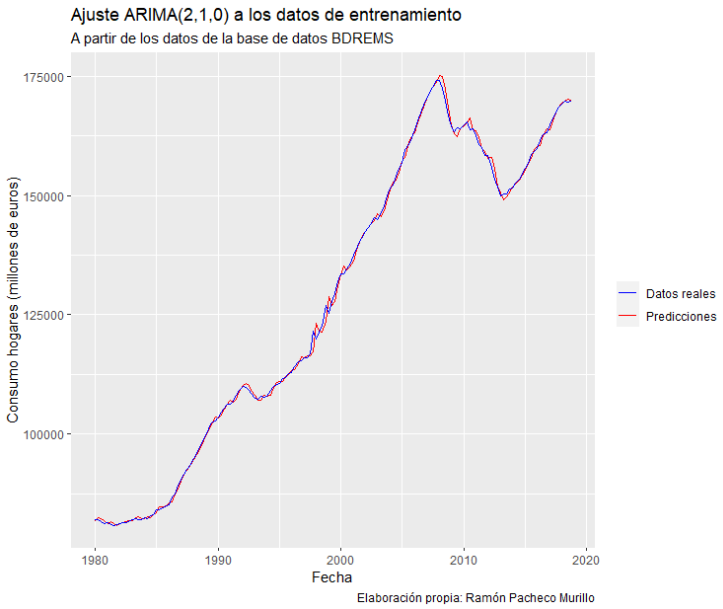


Figura 2.18: Ajuste $ARIMA(2,1,0)$ a los datos de entrenamiento.

Gráficamente se observa que el ajuste para los datos de entrenamiento es óptimo. La función `auto.arima(x)` ajusta tomando el modelo que menor AIC presenta, mostrando también el resto de medidas de error para el conjunto de entrenamiento:

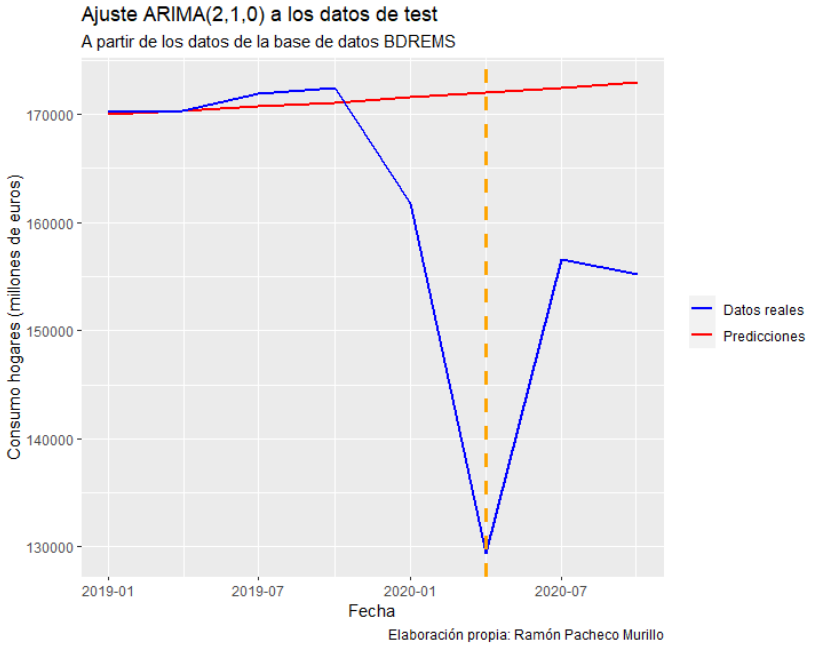
```
Training set error measures:
                    ME     RMSE      MAE          MPE      MAPE      MASE         ACF1
Training set 6.876334 940.9953 623.3039 0.008456465 0.4837478 0.1689025 -0.01393289
```

Tomando este modelo como base, realizamos predicciones para los 8 siguientes trimestres (dos años), obteniendo:

Horizonte	Predicción	L. Inf.(80 %)	L. Sup.(80 %)	L. Inf.(95 %)	L. Sup.(95 %)
2019 - T1	170.012,4	168.790,7	171.234,1	168.144	171.880,9
2019 - T2	170352.1	168391.8	172312.5	167354.0	173350.3
2019 - T3	170699.2	167813.9	173584.4	166286.6	175111.7
2019 - T4	171108.4	167359.1	174857.8	165374.3	176842.6
2020 - T1	171536.6	166912.6	176160.6	164464.8	178608.4
2020 - T2	171995.3	166537.1	177453.5	163647.6	180343.0
2020 - T3	172469.6	166203.1	178736.1	162885.8	182053.4
2020 - T4	172960.4	165922.3	179998.5	162196.6	183724.3

(donde L.Inf y L.Sup denotan los valores de los intervalos de confianza a sus respectivos niveles de significación).

Representando las predicciones y los valores reales gráficamente obtenemos:



La línea vertical discontinua naranja marca la observación correspondiente al primer trimestre tras el confinamiento. Recordemos que nuestro objetivo era simular qué podría haber pasado con el consumo en ausencia de la pandemia. Por tanto, nos interesa ver lo que ocurre antes de este periodo cuya última observación se sitúa en el último trimestre de 2019. Se aprecia que el modelo propuesto logra capturar la tendencia previa razonablemente, y por tanto, la simulación puede ser tenida en cuenta (error absoluto medio porcentual - 7.7 %).

En los datos de la tabla, también podemos ver que, a medida que nos alejamos en el horizonte de predicción, el intervalo de confianza para ambos niveles de significación se hace más amplio. Evidentemente, este comportamiento no debería sorprendernos, pues a medida se avanza en el tiempo, la componente autorregresiva del modelo incorpora los valores estimados anteriores con sus respectivos errores.

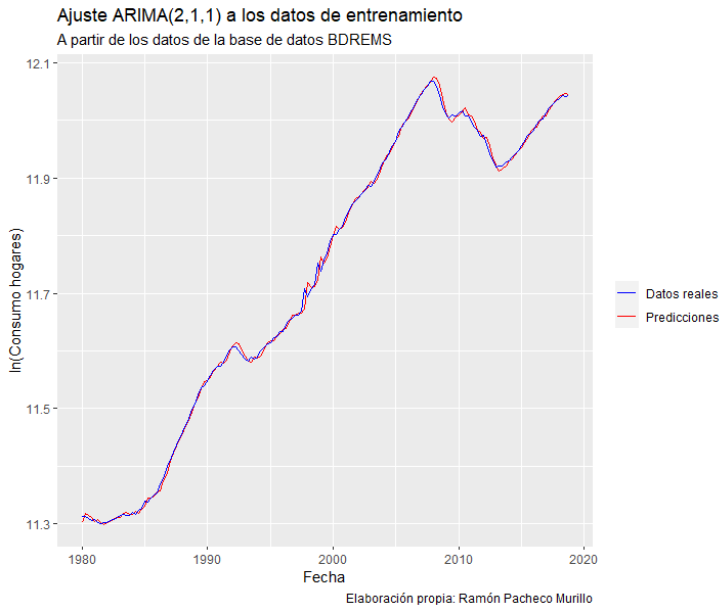
Por último, exploramos brevemente los resultados obtenidos si se aplica antes una transformación logarítmica a la serie temporal con el objetivo de estabilizar la varianza. En este caso, la función `auto.arima(x)` nos indica que el modelo que mejor se ajusta es un $ARIMA(2,1,1)$:

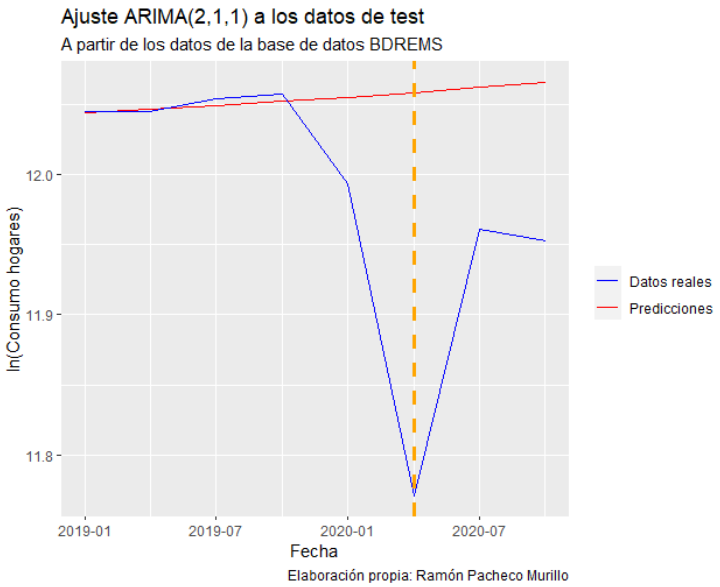
```
> modelo_log
Series: log(serieTraining)
ARIMA(2,1,1) with drift
```

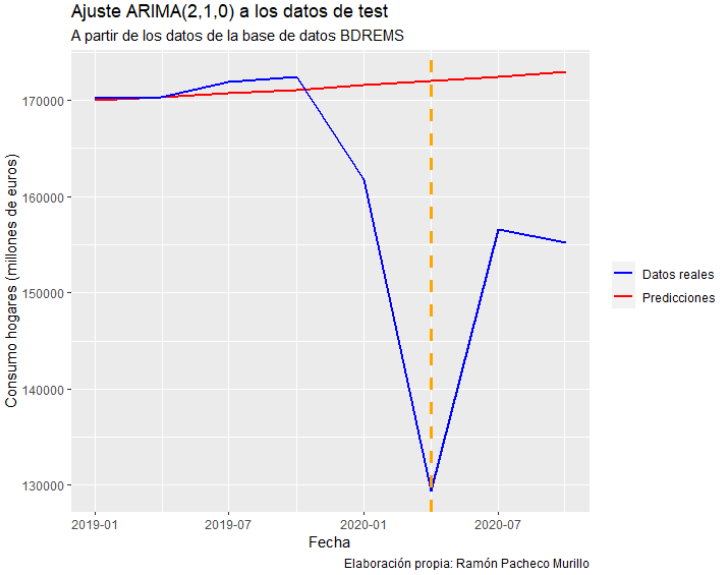
```
Coefficients:
         ar1     ar2      ma1   drift
      0.4904  0.3145  -0.3432  0.0044
s.e.  0.1781  0.1100   0.1892  0.0019

sigma^2 = 5.334e-05:  log likelihood = 545.54
AIC=-1081.08   AICc=-1080.68   BIC=-1065.86
```

A continuación, se comprueba que el ajuste del modelo $ARIMA(2,1,1)$ a los datos de entrenamiento es óptimo. Además, las predicciones obtenidas para los datos de test no difieren en exceso del análisis presentado anteriormente dándole cierta validez:




Ajuste ARIMA(2,1,1) a los datos de test
A partir de los datos de la base de datos BDREMS
ln(Consumo hogares)
12.0
11.9
11.8
2019-01
2019-07
2020-01
2020-07
Fecha
Datos reales
Predicciones
Elaboración propia: Ramón Pacheco Murillo


Ajuste ARIMA(2,1,0) a los datos de test
A partir de los datos de la base de datos BDREMS
Consumo hogares (millones de euros)
170000
160000
150000
140000
130000
2019-01
2019-07
2020-01
2020-07
Fecha
Datos reales
Predicciones
Elaboración propia: Ramón Pacheco Murillo

Capítulo 3

Técnica de Random Forest

En este capítulo nos centramos en el método de Random Forest, una técnica de aprendizaje automático supervisado de clasificación y regresión [5] [6]. Recordemos que nuestro objetivo último es utilizar esta técnica con series temporales y discutir su desempeño en el ajuste y predicción de los datos. Sin embargo, antes de poder aplicar el método a observaciones reales, es necesario comprender el funcionamiento de los árboles de clasificación y decisión e indagar brevemente en cuestiones como el propio concepto de aprendizaje supervisado o los tipos que nos podemos encontrar.

Además, puesto que esta herramienta no fue específicamente diseñada para mantener la estructura temporal de los datos, será necesario tener en cuenta una serie de consideraciones que permiten su uso y que se abordan en las siguientes secciones.

En definitiva, este capítulo contiene la base teórica que permite comprender la herramienta de aprendizaje automático que utilizaremos para analizar en el siguiente capítulo 4 los datos de consumo de los hogares e Instituciones sin fines de lucro al servicio de los hogares (ISFLSH).

3.1. Introducción al Aprendizaje Automático

Formalmente, podemos entender un problema que resolvemos mediante aprendizaje automático como un problema de optimización matemática:

$$F(X, Y, \beta; \lambda) \longrightarrow \beta \tag{3.1}$$

donde:

- X es el conjunto de datos de entrada, es decir, los datos que introducimos en el modelo para poder ser procesados.
- Y es la variable respuesta que produce el modelo tras procesar los datos de entrada.
- β es un vector de parámetros que depende de los datos de entrada y de la clase de modelo utilizado.
- $F(\cdot, \beta)$ es la función objetivo a optimizar - con respecto a los parámetros β - mediante un algoritmo.
- λ son los hiper-parámetros que controlan características internas del modelo y que influyen en el rendimiento del mismo. Por ejemplo, el número de capas ocultas en una red neuronal, el número de árboles a utilizar en un modelo de bosques aleatorios...etc.

Partiendo de un problema y datos de entrada determinados, deberemos elegir una estrategia para tratar los datos a través del modelo apropiado. Es importante tener en cuenta que los algoritmos de optimización que usados por el modelo pueden generar soluciones no óptimas si sus parámetros no están correctamente ajustados o no existe convexidad en la función objetivo. Sobre esto último, en la formulación propuesta en (3.1), no se ha asumido que $F(\cdot, \beta)$ fuera una función convexa. Si ese fuera el caso, el problema podía haberse planteado como un problema de minimización donde $F(\cdot, \beta)$ representa la función de costes, y su convexidad garantizaría la existencia de una solución única. No obstante, se ha decidido usar una formulación más general pues, en la mayoría de los casos, esta asunción no se cumple pudiendo existir, puntos silla y mínimos locales [9].

Supongamos que se tienen m observaciones de n características. Cada observación, denotada por x_i, será un vector perteneciente a $\mathbb{R}^n$ y tendrá asociada una variable de respuesta y (asumiendo que estamos en un caso de aprendizaje supervisado). Por tanto, matricialmente se tiene:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \qquad X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix} \tag{3.2}$$

Aunque en el conjunto de datos de entrada se tengan n características, esto no implica que el modelo se limite únicamente al uso de estas, pues los modelos de aprendizaje automático también pueden incluir interacciones entre las variables, transformaciones mediante funciones...etc. Los datos incluidos en X e Y pueden ser de diversos tipos - como booleanos, categóricos y numéricos - y deberán ser divididos en función de las etapas que aplicaremos:

Habiendo establecido la estrategia para abordar el problema a través de aprendizaje automático y dividido los datos, se procede al entrenamiento, calibración y validación del modelo:

- **Entrenamiento:** En esta etapa, se usa el conjunto de datos de entrenamiento para ajustar los parámetros del modelo de manera que se minimice el error cometido al realizar las estimaciones.
- **Calibración:** Para entrenar el modelo, se han establecido previamente los valores de los hiper-parámetros que influyen tanto en la manera en la que se entrena al modelo, como en la forma en la que se realiza la predicción. Estos valores no cambian durante la fase de entrenamiento, pero sí podemos cambiarlos en esta fase, con el objetivo de obtener aquellos valores que maximicen la precisión del modelo y evitemos fenómenos como el sobreajuste.
- **Evaluación:**A la hora de construir un modelo de aprendizaje automático, uno de los principales objetivos a perseguir es que su rendimiento en datos que no son de la muestra sea óptimo. Así, en esta fase se toman los datos de prueba y se evalúa, atendiendo a distintas métricas, si el modelo es capaz de hacer predicciones precisas sobre estos datos. Si el desempeño no es óptimo, se deberá considerar la realización de ajustes adicionales en las fases previas del modelo y, en caso, de perseverar los resultados, ver si el modelo elegido o la estructura de los datos es la adecuada.

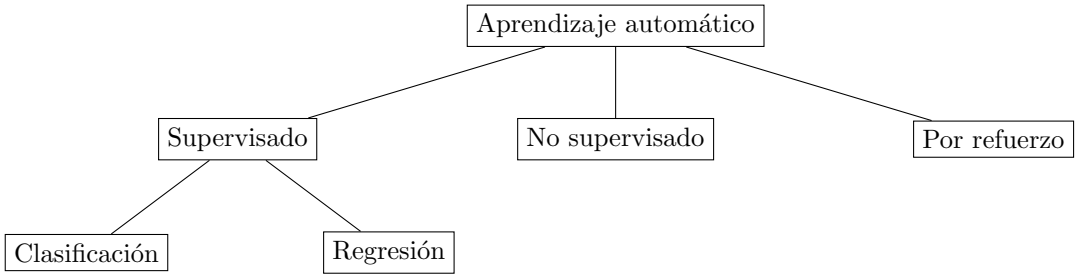
3.1.1. Tipos de aprendizaje automático

En general, podemos distinguir tres tipos de aprendizaje automático [10]:

- **Aprendizaje supervisado:** este enfoque es análogo a la modelización clásica donde se pretender predecir el valor de una variable respuesta Y a partir de unos datos de entrada X. Para ello, necesitaremos un conjunto de datos de entrenamiento donde a cada observación $x_i \in X$ le corresponde un valor de $y_i \in Y$. Los datos de entrenamiento permiten que los algoritmos aprendan la relación existente entre los datos de entrada y la

variable respuesta, de manera que, al introducir datos nuevos puedan realizar las predicciones más acertadas posibles. Si la variable respuesta es discreta y pretende categorizar los datos atendiendo a diferentes etiquetas, estamos ante un problema de clasificación. Si por el contrario es una variable continua, estamos ante un problema de regresión. No obstante, esta clasificación no es exhaustiva pues, en algunos casos, la variable respuesta puede ser difusa.

- **Aprendizaje no supervisado:** este enfoque está centrado en examinar la estructura de los datos de entrada para poder encontrar patrones presentes en los datos y ser organizados en base a ellos. Por ejemplo, dado un conjunto de datos de entrada, podemos agruparlos en función de su similitud respecto a una característica concreta. En este caso, no tenemos una variable respuesta asociada a cada observación, es decir, los datos no están etiquetados.
- **Aprendizaje por refuerzo:** este enfoque está basado en el mecanismo de prueba y error, donde un agente, normalmente una máquina o un programa informático, aprende mediante la recompensa negativa o positiva que recibe al realizar diferentes acciones. Se puede interpretar como un bucle donde un agente situado en un determinado estado, debe decidir si transitar a nuevos estados (con la retroalimentación negativa o positiva que el cambio de estado conlleve) para maximizar la recompensa acumulada.



3.1.2. Consideraciones particulares para series temporales

Cuando aplicamos modelos basados en aprendizaje automático sobre datos de sección cruzada, la pérdida de la estructura temporal que pueda estar implícitamente presente en ellos no supone un problema. En otras palabras, con este tipo de datos, podemos dividir la muestra para el entrenamiento, calibración y test de manera aleatoria sin ningún tipo de percance, pues podemos asumir que son variables aleatorias independientes e idénticamente distribuidas.

Sin embargo, al tratar con series temporales, es muy probable que exista una dependencia temporal entre los valores pasados y futuros de la serie o la presencia de patrones estacionales y tendencias a largo plazo. Si no se tiene en cuenta esto

en la división de los datos para el entrenamiento, calibrado y test del modelo, este podría no capturar estas características, generando predicciones inexactas. Para afrontar este inconveniente, podemos recurrir al método de *sliding window* o ventana deslizante y a sus variantes.

Método de *sliding window* o ventana deslizante

El primer paso para aplicar este método es establecer el tamaño de la ventana deslizante y el paso de avance que vamos a considerar (que usualmente es uno). En cada iteración, se toma la siguiente observación y se añade a la ventana, pero dado que el tamaño de esta es fijo, se debe eliminar la primera. El modelo será entrenado con las observaciones de la ventana correspondiente a ese momento, devolviendo una predicción en base a estos. En este caso, las observaciones dentro de cada ventana se toman de manera secuencial y no aleatoria. De esta manera generamos múltiples muestras de entrenamiento a partir de una única serie temporal donde permanece la dependencia entre ellas [13]. Veamos un ejemplo.

Consideremos los datos de gasto en consumo final en millones de euros facilitados por el I.N.E entre 2005 y 2006:

{169.616, 173.258, 176.424, 180.083, 183.321, 187.317, 190.242, 193.284}

Supongamos que nuestra ventana es de 4 valores, que el paso de avance es unitario, y que la predicción del modelo en la ventana i se denota por $\hat{c_i}$. Para la primera iteración se tiene:

- Datos de entrenamiento: (169.616, 173.258, 176.424, 180.083) $\rightarrow \hat{c_1}$

Dado que el paso de avance es uno, para la segunda iteración tomamos la siguiente observación del conjunto inicial, y eliminamos la primera de la ventana. Con estos nuevos datos de entrenamiento generamos la predicción:

- Datos de entrenamiento: (173.258, 176.424, 180.083, 183.321) $\rightarrow \hat{c_2}$

Para la tercera iteración, volvemos a tomar la siguiente observación del conjunto inicial, y eliminamos la primera de la ventana. Con estos nuevos datos de entrenamiento generamos la predicción correspondiente:

- Datos de entrenamiento: (173.258, 176.424, 180.083, 183.321) $\rightarrow \hat{c_3}$

Se prosigue hasta que ya no queden elementos en el conjunto inicial.

Es importante destacar que, al usar este método, el tamaño de la ventana y el paso de avance influirán en la capacidad del modelo. Por un lado, si se utilizan ventanas formadas por un número demasiado pequeño de observaciones, será más complicado para el modelo captar los patrones a largo plazo presentes en la serie temporal. Por otro, la elección de un tamaño muy grande provocaría

una reducción en el número de muestras de entrenamiento. En definitiva, para fijar estos parámetros es necesario buscar un equilibro entre la cantidad de datos disponibles para el entrenamiento del modelo y su capacidad para captar patrones a largo plazo.

3.2. Árboles de clasificación y regresión.

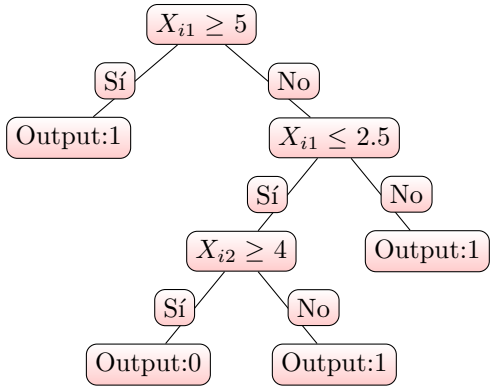
Un árbol de clasificación y regresión (C.A.R.T - por sus siglas en inglés) es un algoritmo de aprendizaje automático supervisado basado en decisiones binarias [5]. Cuando se está ante de un problema de regresión, el árbol será de regresión. Si estamos ante un problema de clasificación, el árbol es de clasificación.

Partiendo de un conjunto de datos inicial, en cada nodo del árbol se establece un criterio para dividir las observaciones en función del cumplimiento o no del mismo, hasta llegar a un nodo terminal. En el nodo terminal se determinará el valor de la variable de respuesta. Veamos un ejemplo.

Supongamos que se tienen n observaciones y cada observación tiene dos características:

$$X_i = (X_{i1}, X_{i2}) \quad i \in 1...n \tag{3.3}$$

Consideremos el siguiente árbol de clasificación:



Se trata de un árbol de clasificación con un nodo raíz ($X_{i1} \geq 5$, caracterizado por no tener ramas entrantes y ser aquel donde comienza el árbol), dos nodos internos o de decisión ($X_{i1} \leq 2{,}5$ y $X_{i2} \geq 4$ donde se evalúan los datos) y cuatro nodos terminales que contienen los posibles resultados. El resultado de aplicar el árbol de clasificación a las siguientes observaciones de muestra sería:

- $X = (6, 5)$. En este caso, la variable respuesta sería 1 pues la condición del primer nodo se cumple $6 \geq 5$, llegando directamente a un nodo terminal.
- $X = (1, 1)$. Esta observación no cumple la condición del primer nodo ($1 \ngeq 5$), pero sí la del segundo ($2{,}5 \geq 1$). Como se tiene $1 \ngeq 4$, en el tercer nodo, se llega al nodo terminal que le asigna el valor 1.

- $X = (0, 7)$. Esta observación no cumple la condición del primer nodo ($0 \ngeq 5$), pero sí la del segundo ($2{,}5 \geq 0$). Como se tiene $7 \geq 4$, en el tercer nodo, se llega al nodo terminal que le asigna el valor 0.

- $X = (3{,}5, 9)$. En este caso, la variable respuesta sería 1 pues la condición del primer nodo no se cumple $3{,}5 \ngeq 5$, y la del segundo nodo tampoco $3{,}5 \nleq 2{,}5$, llegando directamente a un nodo terminal con valor 1.

Describir el funcionamiento de un árbol de decisión dado es sencillo, no obstante, a la hora de construir uno surgen tres preguntas fundamentales:

- ¿Cómo deberían establecerse las divisiones de los datos? En otras palabras, ¿cómo establecemos los nodos de decisión?

- ¿Bajo qué criterio dejamos de hacer particiones y situamos un nodo terminal?

- En el caso de los árboles de regresión, ¿qué valor se le asigna a los datos que lleguen a un determinado nodo terminal?

De estas preguntas se deduce la necesidad de establecer por un lado, un criterio para determinar qué tan buena es una partición y, por otro, para determinar cuándo deja de ser necesario realizar una nueva división de los datos. A continuación, se explica un procedimiento recursivo para obtener un árbol de clasificación.

3.2.1. Árbol de clasificación.

A finales de la década de los 50, se establece un método para la construcción de árboles de decisión de manera recursiva usando un algoritmo de divide y vencerás [2]:

- Si todos los elementos del conjunto de entrenamiento son de la misma clase, entonces el árbol está formado por un solo nodo terminal etiquetado con el nombre de esa clase.

- Supongamos que en los datos de entrenamiento hay distintas clases. Si se llega a un nodo donde se cumple un criterio de parada (o todos los datos en ese nodo son de la misma clase), entonces el nodo es una hoja del árbol. En caso contrario, se selecciona un criterio basado en los valores que toman las características de los datos para ramificar el árbol de decisión. De esta manera, se va dividiendo el conjunto de entrenamiento en subconjuntos según el criterio del nodo y para cada subconjunto, se vuelve a aplicar el mismo procedimiento anterior.

- En el caso de los árboles de clasificación, el criterio de parada quedará determinado por la pureza del nodo medida a través del métodos como la ganancia de información, el criterio de proporción de ganancia o el índice de diversidad de Gini (este último es el que se usa en los C.A.R.T).

Un nodo se dice puro si todo elemento del subconjunto de datos que le corresponden son de la misma clase. Evidentemente, es muy complicado obtener nodos completamente puros, lo que se buscará es obtener grupos lo más homogéneos posible.

3.2.2. Árbol de regresión.

En el caso de los árboles de regresión se produce una serie de variaciones [2]:

- Recordemos que en los problemas de aprendizaje automático de regresión el objetivo es predecir el valor numérico de una variable respuesta. Así, mientras el procedimiento de división recursiva de los datos en subconjuntos homogéneos hasta alcanzar un nodo terminal permanece igual que en el caso de los árboles de clasificación, la manera de realizar dicha división varía.
- En el caso de los árboles de regresión se va a buscar la variable y el punto de corte de los datos que maximice la reducción de la suma al cuadrado de los residuos. Para ello, se toma todas las combinaciones posibles entre variables predictoras y puntos de corte como división y se calcula la suma al cuadrado de los residuos para cada una de ellas. Por último, se selecciona la variable y valor de corte que maximiza la reducción en la suma al cuadrado de los residuos.
- El valor del nodo terminal vendrá determinado por la media de los valores de la variable respuesta de los datos que terminen en el nodo.
- No se debe olvidar que la selección de la mejor variable y valor de corte para la división dependerá no solamente de los datos, sino también de parámetros del modelo fijados a priori como la profundidad máxima del árbol o el número de observaciones mínimas que debe tener cada nodo terminal.

Considerando, m observaciones en el conjunto de entrenamiento, formalmente se tiene:

$$(x_{1,1}, x_{1,2}, \cdots, x_{1,n}, y_i) \quad con \quad i \in 1, ..., m \tag{3.4}$$

En cada nodo del árbol de regresión, se debe decidir la variable (x_j) y el punto de corte (k) que minimicen el error cuadrático medio:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.5}$$

Fijados una variable x_j y un punto de corte k, establecer la condición $x_j \geq k$ implica dividir los datos en dos nodos:

$$\begin{aligned} N_R &= \{i : x_{ij} > k\} \\ N_L &= \{i : x_{ij} < k\} \end{aligned} \tag{3.6}$$

El MSE de cada nodo vendrá determinado por:

$$\begin{aligned} MSE_{N_R} &= \frac{1}{|N_R|} \sum_{i=1}^{|N_R|} (y_i - \hat{y}_i)^2 \\ MSE_{N_L} &= \frac{1}{|N_L|} \sum_{i=1}^{|N_L|} (y_i - \hat{y}_i)^2 \end{aligned} \tag{3.7}$$

Así en cada nodo, se probará con todas las posibles variables x_j y valores de corte k, seleccionando aquella condición $x_j \geq k$ que minimice el MSE de nodos en los que divide los datos. Por tanto, el árbol de regresión final será aquel que minimice el MSE en todos sus nodos dados los parámetros fijados a priori como la profundidad máxima del árbol o el número de observaciones mínimas que debe tener cada nodo terminal.

3.2.3. Ventajas e inconvenientes.

Usar árboles de regresión y clasificación permite identificar relaciones complejas entre predictores y la variable respuesta de manera eficiente y rápida. Además, manejan bien la presencia de outliers en los datos iniciales e interpretar cómo ha llegado el algoritmo al resultado final es más sencillo que en otros modelos basados en aprendizaje automático.

Sin embargo, también son frecuentes su tendencia al overfitting, alta varianza o empeoramiento del rendimiento de los árboles de clasificación cuando en los datos iniciales predomina una de las clases.

3.3. Random Forest.

El random forest es una técnica que combina los árboles C.A.R.T y el método de *bagging* [6].

El método de *bagging* permite obtener múltiples muestras a partir de un conjunto de entrenamiento tomando observaciones al azar con reemplazamiento. Con estas observaciones se crea un conjunto del mismo tamaño que los datos de entrenamiento originales.

Creando y combinando los árboles de regresión de cada uno de los subconjuntos resultantes de aplicar el método de *bagging* a los datos de entrenamiento, se obtiene el random forest.

Utilizar el método de *bagging* nos permite conseguir predicciones más precisas que el uso de un árbol individual. La predicción de un único árbol individual podría estar sujeta a fenómenos como el overfitting, o simplemente, ser una mala predicción. El método de Random Forest toma la media de las predicciones de

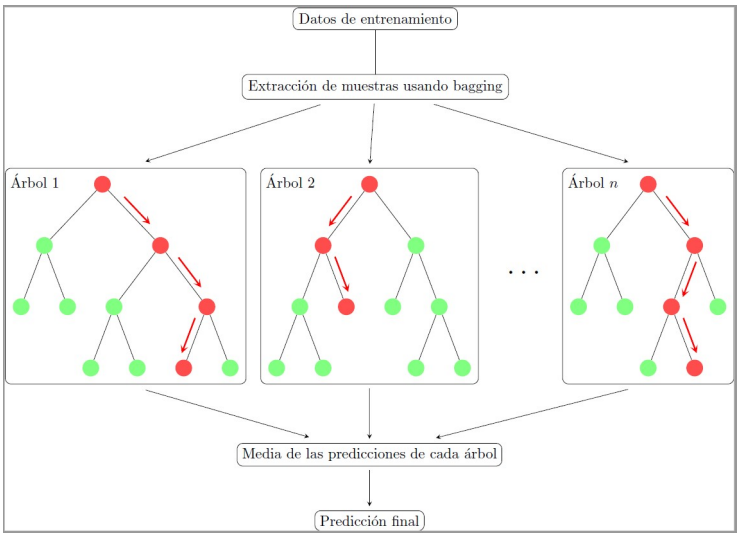


Figura 3.1: Funcionamiento Random Forest. Elaboración propia a partir de: `https://tikz.net/random-forest/`

los árboles individuales que ha construido, disminuyendo el peso que tiene una mala predicción en el resultado final.

Los parámetros más importantes de un Random Forest son:

- **Número de árboles de regresión:** aumentar este valor puede mejorar la precisión del modelo a cambio de un tiempo de ejecución mayor. Según Hastie [12], los resultados suelen estabilizarse en el rango (250,500).
- **Profundidad máxima permitida en el árbol:** utilizar una profundidad muy alta puede llevar a un error de sobreajuste del modelo, mientras que, fijar un valor demasiado bajo puede conducir a un modelo excesivamente simple y, por tanto, menos preciso.
- **Número mínimo de observaciones en los nodos terminales:** si se establece un número demasiado bajo puede llevar a un sobreajuste del modelo, afectando negativamente a la capacidad predictiva del mismo.
- **Número máximo de variables a considerar en cada división:** como hemos visto anteriormente, en la parte recursiva de la construcción del árbol de regresión considerábamos todas las posibles variables x_j para establecer la división. Podemos limitar el número de variables a considerar si ayuda a reducir el sobreajuste del modelo y, por tanto, aumentar su desempeño sobre datos nuevos. Por ejemplo, se puede eliminar de la base de datos aquellas características que sepamos con certeza que no influyen en la variable resultado cuando tengamos más variables que observaciones.

Capítulo 4

Desarrollo del análisis y resultados

En este capítulo, aplicamos la técnica de Random Forest anteriormente explicada para predecir los datos de consumo de los hogares e Instituciones Sin Fines de Lucro al Servicio de los Hogares (ISFLSH).

Aprovechando que los árboles de regresión son eficaces tratando con un amplio conjunto de variables y que, el sofware *R* nos permite trabajarlas con comodidad, consideraremos un conjunto de variables económicas recogidas en la base de datos *REMSDB* que incluye la serie histórica del consumo.[1]

Aplicando dos estrategias de entrenamiento del modelo, una que permite la expansión de los datos de entrenamiento y otra que no, [2] analizamos los resultados del desempeño [3] del Random Forest con cada estrategia, primero por separado y después estableciendo comparaciones. Acompañaremos este análisis con tablas y representaciones gráficas que permiten representar adecuadamente los resultados.

También exploramos cómo cambia el comportamiento del modelo en función del número de árboles del Random Forest. Por último, compararemos con las estimaciones obtenidas en la sección 2.5.

[1] En consonancia con otros estudios que usan la técnica Random Forest para predecir variables económicas e incluyen una amplia base de datos y no únicamente la de la serie histórica de la variable a predecir [1], [4], [9].

[2] El primer enfoque se denominará dinámico. El segundo, estático

[3] A través de tres métricas de error: Raíz del error cuadrático medio (R.M.S.E), Error absoluto medio (M.A.E) y Error absoluto medio porcentual (M.A.P.E)

4.1. Uso de la técnica Random Forest para la estimación del consumo final.

En la sección 2.5 hemos aplicado un modelo estadístico usando el método de Box-Jenkins para estimar la evolución del consumo final a partir de su evolución histórica. En este apartado, el objetivo perseguido será conseguir una estimación del consumo final de los hogares, pero a partir de los datos históricos de varias variables macroeconómicas que nos permita aprovechar la técnica de RF.

4.1.1. Descripción de la base de datos.

La base de datos *REMSB* contiene las series de las variables macroeconómicas utilizadas por J.E. Boscá, A. Díaz, R. Doménech, J. Ferri, E. Pérez y L. Puch para desarrollar un Modelo Macroeconómico de Expectativas Racionales para España. Aunque pudiera parecer que la base de datos está diseñada exclusivamente para este uso, los autores advierten que, dada la multitud de variables recogidas y sus óptimas características, puede ser usada para múltiples labores relacionadas con la simulación a medio plazo de variables macroeconómicas [3].

Se puede agrupar las variables presentes en la base de datos en cinco categorías:

- Agregados nacionales relacionados con la producción y la demanda.
- Variables relacionadas con la población y el mercado laboral.
- Variables monetarias y financieras.
- Agregados económicos gubernamentales relevantes para la macroeconomía.
- Otras variables económicas relevantes que no se incluyen en los grupos anteriores.

Los datos de *REMSDB* son agregados nacionales de carácter trimestral. Cuando la frecuencia de la serie en las estadísticas oficiales no era trimestral, los autores usaron técnicas apropiadas y específicas para ajustar cada serie. Además, los autores procedieron a la desestacionalización de los datos.

Dado que la base de datos sigue actualizándose, es importante mencionar que la que se usa en este trabajo corresponde a aquella cuya última actualización data de diciembre de 2021. Por tanto, se tienen datos trimestrales desde 1980 hasta el tercer trimestre de 2021.

Partiendo de esta base de datos, se han eliminado aquellas variables sobre las que no se disponía de datos para todos los períodos. Además, también se ha suprimido aquellas variables que ya se consideraban suficientemente representadas en el modelo y, así, evitar un posible overfitting. Por ejemplo, la base de datos original contiene las siguientes 4 variables: Importación de bienes y

servicios, Importación de bienes de consumo, Importación de bienes de capital e Importación de bienes intermedios, donde la primera resulta de la suma de las tres últimas y, por tanto, podemos quedarnos con las tres últimas o con la primera. Por último, es importante mencionar que la variable a predecir por nuestro modelo RF es el Consumo final de los hogares y la Instituciones sin fines de lucro al servicio de los hogares (ISFLSH) y no, el Consumo final de las administraciones públicas.

Finalmente, aplicando los criterios anteriormente mencionados, de 78 variables disponibles en la base de datos original nos quedamos con 62 variables - cuyo listado completo, puede consultarse en el anexo A. Entendiendo como observación un vector $X_j = (x_{1j}, ..., x_{62j})$ de 62 variables para el trimestre j correspondiente, la base de datos sobre la que aplicaremos el RF dispone de 164 observaciones.

4.1.2. Estrategia de aplicación.

El primer paso para llevar a cabo nuestro análisis consiste en dividir la base de datos en dos: la parte de entrenamiento y la parte de test. Los datos de entrenamiento suponen el 73 % de las observaciones y los datos de test contienen el 27 % restante. Cronológicamente, la fracción de entrenamiento abarca desde el primer trimestre de 1980 al último trimestre de 2009 y, la parte de test, del primer trimestre de 2010 al último de 2020.

Utilizando el paquete `randomForest` en R crearemos un Random Forest de regresión entrenado con los respectivos datos de entrenamiento . La versión estándar de la función `randomForest` trabaja con 500 árboles y, en cada árbol, limita a $n = j/3$, el número de variables con las que prueba en cada división [16]. A continuación, se probará el desempeño del árbol con la parte de datos de test.

Además, trabajando con R podemos saber cuál es el número de árboles que minimiza el error cuadrático medio del RF. Así ajustaremos este parámetro y podremos hacer comparaciones con el primer modelo implementado.

Dado que la porción de datos de entrenamiento es fija, podemos decir que esta parte del análisis es en cierto modo estática. En otras palabras, estamos viendo qué tan bien se comporta el modelo para predecir los datos de test sin alimentar los datos de entrenamiento con datos adicionales. Para ilustrar mejor esta idea, supongamos que estamos en el último trimestre de 2009, y queremos predecir la evolución del consumo para la década 2010-2020. Dado que los datos relativos a la década 2010-2020 aún no serían observados, no podemos actualizar los datos de entrenamiento con nuevas observaciones y, utilizamos el modelo RF entrenado con información histórica. Este análisis y sus respectivos resultados de esta parte se presentan en 4.2.1.

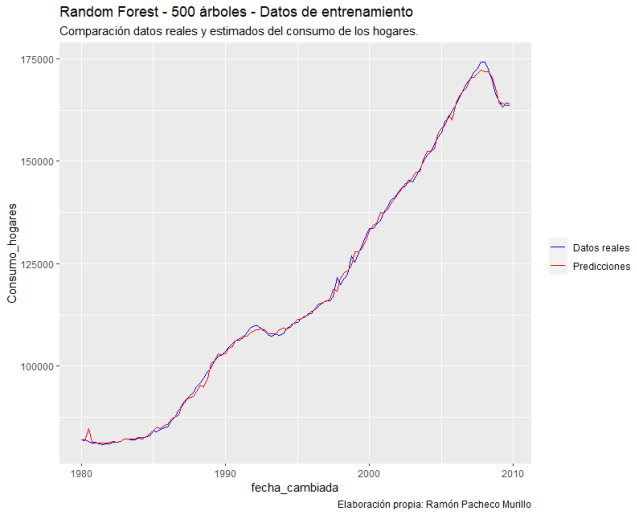
Ahora bien, dado que estamos trabajando con datos de series temporales también es interesante explorar otra manera de realizar las predicciones. Asumiendo que la recogida de la información de las variables fuera instantánea, podemos hacer predicciones cuyo horizonte temporal sea únicamente el siguiente trimestre a partir de la información histórica recogida hasta el trimestre anterior. De esta manera, los datos de entrenamiento van creciendo conforme avanzamos en el tiempo y podremos ajustar un modelo RF para cada predicción. Al igual que en el apartado estático, variaremos el parámetro correspondiente al número de árboles. Este análisis y sus respectivos resultados se presentan en 4.2.2.

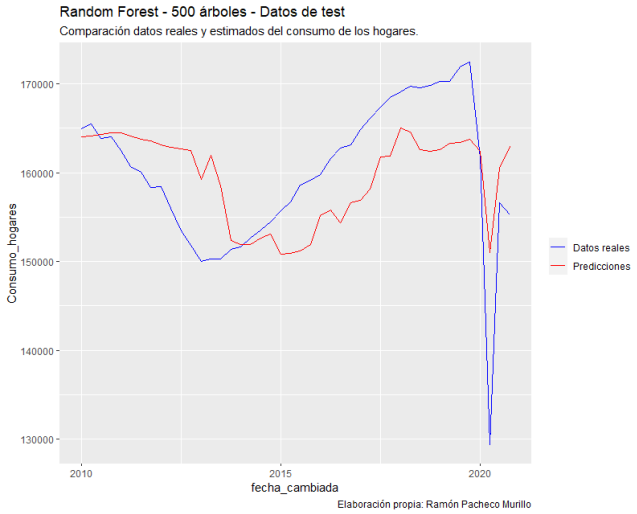
4.2. Resultados.

4.2.1. Resultados parte estática.

El ajuste del RF de 500 árboles a los datos de entrenamiento es óptimo, tal y como muestra la representación gráfica de los datos reales y las predicciones generadas por el modelo (ver siguiente página). La división de los datos en las partes de entrenamiento y test, respetando el orden cronológico de la serie temporal, ha permitido que el modelo RF haya sido capaz de captar perfectamente la tendencia de los datos.

Para los datos de test, esperamos un desempeño peor del modelo, pero manteniendo su habilidad de ajustarse a la serie temporal considerablemente bien. Tal y como podemos observar en la representación gráfica de los resultados para los datos de test, aunque el modelo no es tan bueno como para los datos de entrenamiento, sí logra capturar perfectamente la tendencia de los datos (ver siguiente página). Este comportamiento nos confirma que, siguiendo los pasos adecuados, sí se puede aplicar la técnica RF para obtener predicciones de datos con componente temporal.

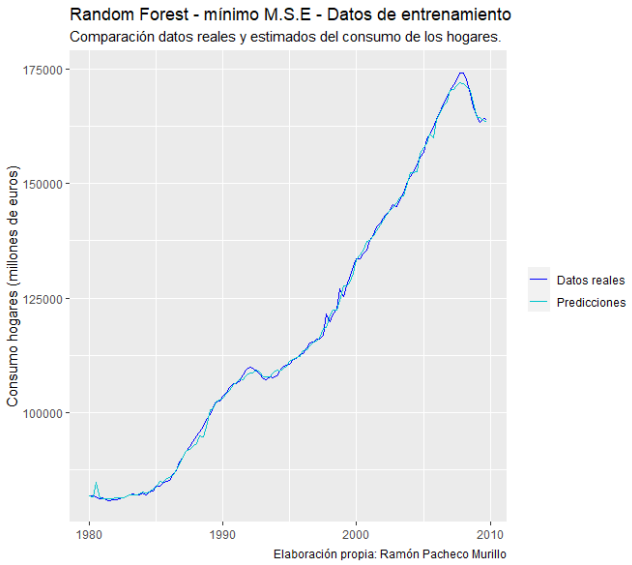


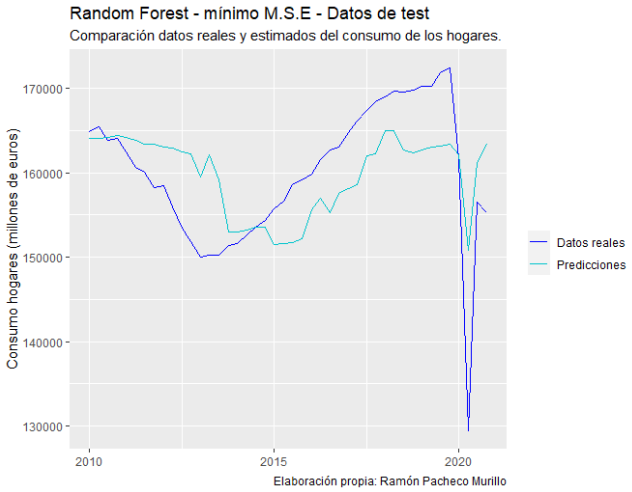


A continuación, se muestran 3 medidas de error (raíz cuadrada del error cuadrático medio - RMSE, error absoluto medio - MAE y error absoluto medio porcentual - MAPE) correspondientes al desempeño del *RF*:

	RMSE	MAE	MAPE
Train	942.59	683.32	0.57 %
Test	6900.7	5687.99	3.61 %

Dado que el software R nos permite hallar el árbol que genera el menor error cuadrático medio a través de la sentencia `which.min(rf_500$mse)`, procedemos a trabajar con este modelo RF, ver su desempeño y comparar con el modelo anterior.





Al igual que ha ocurrido con el modelo anterior, el ajuste del modelo a los datos de entrenamiento es mejor que el de los datos de test. No obstante, ambos logran capturar la evolución de la serie y las predicciones ante datos nuevos son considerablemente buenas.

A continuación, se muestran las 3 medidas de error utilizadas anteriormente correspondientes al desempeño del último modelo RF:

	RMSE	MAE	MAPE
Train	941.64	669.51	0.56 %
Test	6772.5	5529.12	3.51 %

Comparando las medidas de error de los dos modelos, podemos observar que efectivamente el desempeño del último modelo estimado es mejor:

Anterior	RMSE	MAE	MAPE	Último	RMSE	MAE	MAPE
Train	942.59	683.32	0.57 %	Train	941.64	669.51	0.56 %
Test	6900.7	5687.99	3.61 %	Test	6772.5	5529.12	3.51 %

Gráficamente se obtiene:

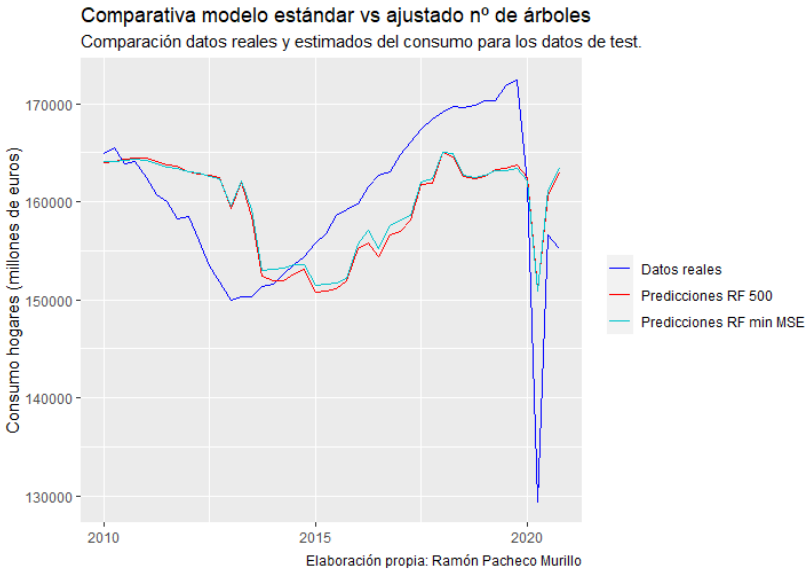


Figura 4.1: Comparativa modelo estándar vs ajustado

4.2.2. Resultados parte dinámica.

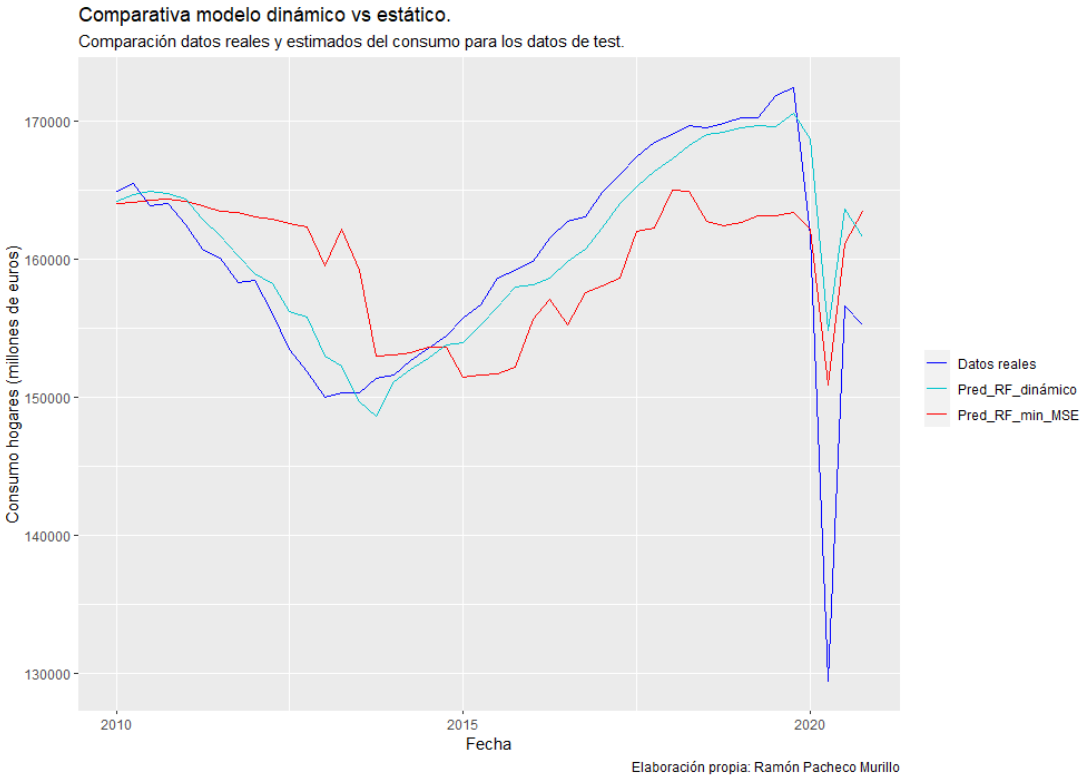
En esta sección exploramos un supuesto más realista pues actualizamos el modelo con datos observados conforme se avanza temporalmente. Aprovechando las estructuras recursivas que ofrece el software *R*, podemos implementar esta idea usando un `for` para ir actualizando el conjunto de entrenamiento, construir un modelo *RF* con estos datos y así obtener una estimación para el trimestre inmediatamente siguiente usando el modelo:

```
  for (i in 0:43) {
#Hacemos los datos de training más grandes cada vez que se ejecuta el bucle
datos_training<-df_datos[1:(120+i),]
#Ajustamos los datos de test para que corresponda
#con el trimestre inmediatamente siguiente
datos_test<-df_datos[(121+i),]
#Quitamos la variable Consumo_hogares de los datos de test
datos_test<-select(datos_test,1:2,4:63)
#Estimamos el modelo
modelo_rf_iterativo=randomForest(Consumo_hogares~., data=datos_training,
  type="regression")
#Hacemos la predicción
predicciones4<- predict(modelo_rf_iterativo,datos_test)
#Guardamos la predicción en orden
vector_resultados[i+1]<-predicciones4
}
```

Con la manera de actualizar los datos aquí presentada, esperamos un mejor desempeño del modelo en la parte de test. Así, comparamos las medidas de error del modelo *RF* con menor M.S.E (calculado en la parte estática) con las cifras que arroja el último modelo estimado, obteniendo:

Comparativa en datos de test	RMSE	MAE	MAPE
Modelo estático	6772.5	5529.12	3.51 %
Modelo dinámico	4587.44	2544.80	1.67 %

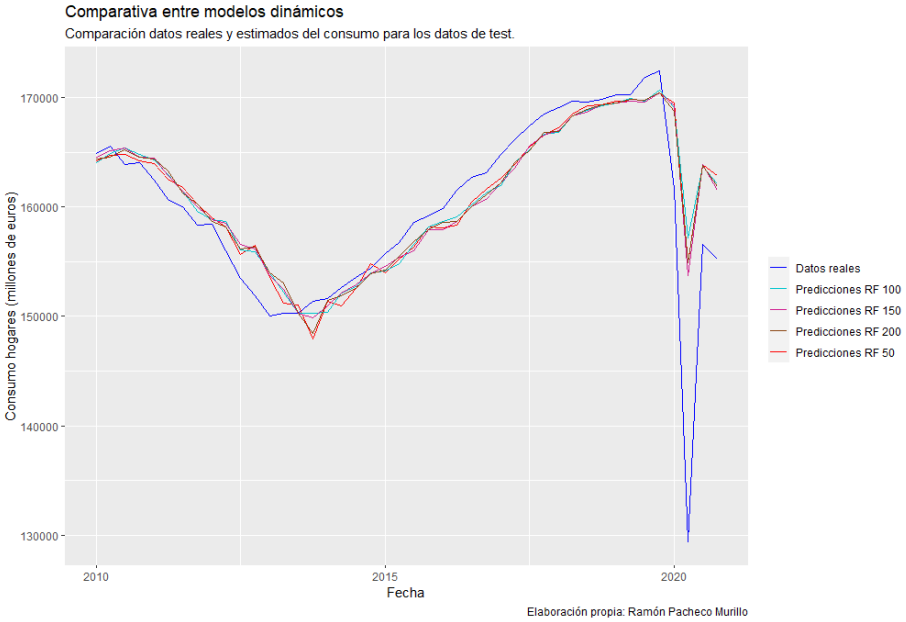
La reducción como consecuencia de entrenar el modelo de manera dinámica es muy considerable en todas las métricas utilizadas (por ejemplo, el *M.A.P.E* se reduce más de la mitad). Este comportamiento corresponde con el esperado pues, por un lado, el modelo tiene más datos a su disposición y, por otro, la predicción se realiza para el trimestre inmediatamente siguiente disminuyendo su error. En la siguiente página, se aprecia gráficamente este resultado:

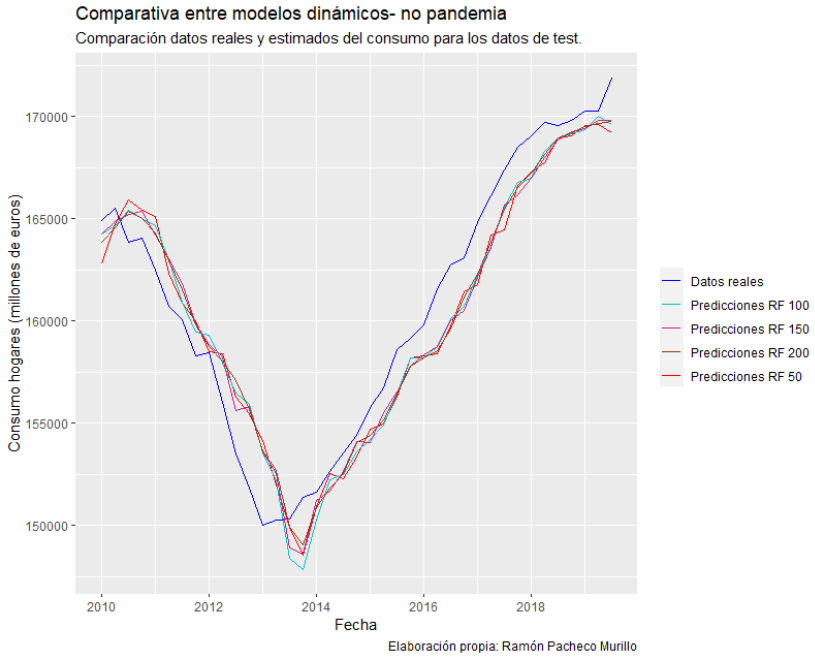


También merece la pena explorar la dimensión del número de árboles y comprobar si existe mucha diferencia entre ellos. Consideremos pues, el método de entrenamiento dinámico y ajustemos un modelo *RF* con árboles de tamaño 50, 100, 150, 200 y 500 respectivamente. Se obtienen los siguientes resultados:

Comparativa en datos de test	RMSE	MAE	MAPE
50 árboles	4559.37	2621.02	1.72 %
100 árboles	4397.99	2573.25	1.69 %
150 árboles	4449.891	2552.363	1.68 %
200 árboles	4337.099	2535.72	1.67 %
500 árboles	4587.44	2544.80	1.67 %

A medida que aumentamos el número de árboles en el modelo, mejor se ajustan las predicciones a los datos reales. Sin embargo, llega un momento a partir del cual añadir más árboles genera una ganancia adicional mínima, estabilizándose así el error. En la siguiente página se presentan los resultados gráficamente.


Comparativa entre modelos dinámicos
Comparación datos reales y estimados del consumo para los datos de test.
Consumo hogares (millones de euros)
170000
160000
150000
140000
130000
2010
2015
2020
Fecha
Datos reales
Predicciones RF 100
Predicciones RF 150
Predicciones RF 200
Predicciones RF 50
Elaboración propia: Ramón Pacheco Murillo

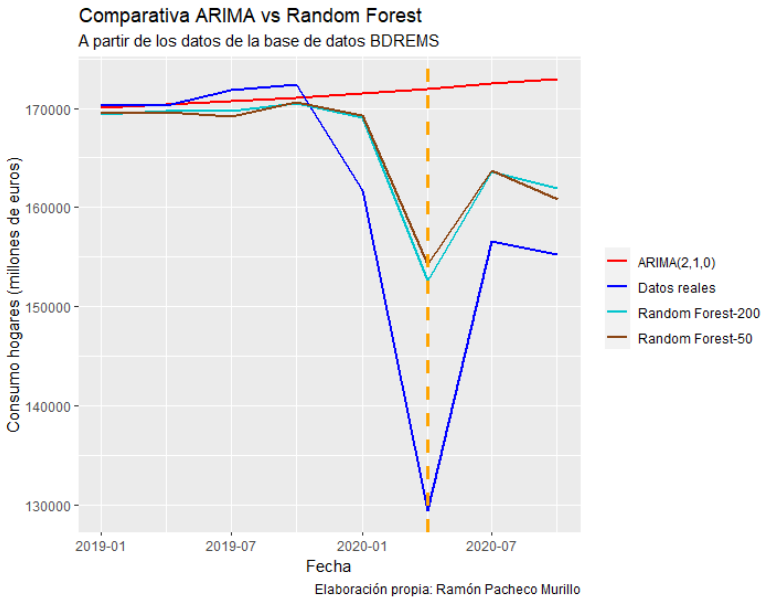

Comparativa entre modelos dinámicos- no pandemia
Comparación datos reales y estimados del consumo para los datos de test.
Consumo hogares (millones de euros)
170000
165000
160000
155000
150000
2010
2012
2014
2016
2018
Fecha
Datos reales
Predicciones RF 100
Predicciones RF 150
Predicciones RF 200
Predicciones RF 50
Elaboración propia: Ramón Pacheco Murillo

4.2.3. Resultados comparativa ARIMA.

Por último, dado que se dispone de los valores de las predicciones realizadas con el modelo ARIMA $(2, 1, 0)$ desarrollado en la sección 2.5, se procede a comparar las estimaciones entre modelos *RF* y el ARIMA. En cuanto a las medidas de error, se tiene:

Comparativa en datos de test	RMSE	MAE	MAPE
50 árboles	4559.37	2621.02	1.72 %
200 árboles	4337.099	2535.72	1.67 %
ARIMA(2,1,0)	17606.06	11103.88	7.77 %

El desempeño del modelo *RF* basado en aprendizaje automático es claramente mejor al modelo ARIMA. Sin embargo, recordemos que este modelo ARIMA tenía dos componentes autorregresivos. Como las predicciones que había realizado el ARIMA para los trimestres anteriores fueron al alza, debido a la componente autorregresiva, las predicciones siguientes del modelo también lo fueron y, por tanto, no puede captar la caída en el consumo consecuencia del confinamiento. También debemos reconocer que el modelo ARIMA era univariante y únicamente disponía de datos de la serie histórica del consumo. El modelo *RF* por su parte dispone de información de más variables, afectando positivamente a su precisión. Representando gráficamente la comparativa obtenemos:



Capítulo 5

Conclusiones

Los algoritmos basados en aprendizaje automático requieren de un tratamiento especial para poder trabajar adecuadamente con datos de series temporales [19]. En este trabajo, se ha utilizado la técnica de Random Forest para establecer predicciones del gasto en consumo final de los hogares e ISFLSH a través de dos enfoques distintos.

El primer enfoque consiste en entrenar el modelo sin permitir la ampliación de los datos de entrenamiento conforme avanza el tiempo. En la sección 4.2.1, se ha comprobado que la técnica de RF ha sido capaz de capturar perfectamente la evolución temporal de los datos, tanto en la parte de entrenamiento, como en las predicciones de test donde no se observa un comportamiento anómalo. Además, se ha ajustado el número de árboles que utiliza el modelo RF para obtener aquel que minimiza el error cuadrático medio en las predicciones.

El primer enfoque utilizado nos confirma que, con el tratamiento adecuado, la técnica de RF puede utilizarse para realizar predicciones de series temporales. Sin embargo, los resultados en la parte de test son modestos. Con el objetivo de mejorar las predicciones en los datos de test, introducimos el segundo enfoque, que permite la ampliación de los datos de entrenamiento conforme se avanza en el tiempo.

En la sección 4.2.2, observamos una gran mejoría con respecto al primer enfoque en las tres métricas de error con las que se ha evaluado el ajuste del modelo. La nueva estrategia de entrenamiento permite ajustar mejor los parámetros del modelo - pues vamos introduciendo información nueva - lo que provoca un mejor desempeño en las predicciones. Variando el parámetro del número de árboles del RF, se observa un mejor rendimiento a medida que aumentamos su valor (n) pero que tiende a estabilizarse cuando el valor es muy grande ($n \to \infty$). [1]

[1] En línea con lo mencionado por [12]

Por último, comparamos el modelo $ARIMA$ que hemos desarrollado para iniciarnos en el estudio del análisis de series temporales con el modelo Random Forest. La comparativa muestra que la técnica Random Forest produce mejores resultados. No obstante, debemos reconocer que el modelo ARIMA utilizaba únicamente los datos de la serie histórica del consumo, por lo que una parte de ese mejor desempeño podría deberse a ello.

En definitiva, la metodología expuesta en este trabajo nos ha permitido utilizar la técnica de Random Forest para obtener predicciones razonables de la evolución del gasto en consumo final de los hogares e ISFLSH. Hemos alcanzado así los objetivos previamente definidos en la sección 1.3 dedicando aproximadamente 500 horas, equivalentes a 15 créditos ECTS y cuyo detalle puede consultarse en el cronograma de trabajo (A.3). A continuación, se proponen una serie de posibles trabajos futuros relacionados con el estudio aquí desarrollado:

- Aplicar la técnica Random Forest a otra variable económica estudiando el efecto de la variación de otros parámetros - como la profundidad máxima permitida en el árbol - en las predicciones.
- Utilizar la técnica Random Forest para extraer las variables más importantes que afectan a un determinado fenómeno económico a través de la importancia y estimar un modelo econométrico en base a ellas.
- Establecer una metodología para utilizar otra técnica de aprendizaje automático - como las redes neuronales recurrentes - para hacer predicciones de una serie temporal determinada.

Bibliografía

[1] G.S. Araujo and W.P. Gaglianone, «Machine learning methods for inflation forecasting in Brasil: new contenders versus classical models», *The Banco Central do Brasil Working Papers*, 561, 2022. Disponible aquí.

[2] F. Berzal, «ART - un método alternativo para la construcción de árboles de decisión», *Universidad de Granada*, págs. 15-35, 2002. Disponible aquí.

[3] J.E. Boscá, A. Bustos, A. Díaz, R. Doménech, J. Ferri, E. Pérez and L. Puch «The REMSDB Macroeconomic Database of The Spanish Economy», *International Economics Institute, University of Valencia*, 2007. Disponible aquí.

[4] O.Biau and A.D'Elia, «Euro area GDP forecasting using large survey datasets: a random forest approach», *European Commission*, 2010.

[5] L.Breiman, J.Friedman, C.J. Stone and R.A. Olshen «Classification and Regression Trees», *Taylor & Francis Group*, 1984.

[6] L.Breiman, «Random Forests», *Machine Learning*, 45, págs. 5-32, 2001.

[7] M.Camacho, S.Ramallo and M. R. Marín, «Árboles de decisión en economía: una aplicación a la determinación del precio de la vivienda», Nuevos métodos de predicción económica con datos masivos, págs. 61-93, *Funcas*, 2021. Disponible aquí.

[8] Á. Caro and D. Peña, «Predicción de series temporales económicas con datos masivos: perspectiva, avances y comparaciones», Nuevos métodos de predicción económica con datos masivos, págs. 5-32, *Funcas*, 2021. Disponible aquí.

[9] C. Chakraborty and Andreas Joseph, «Machine learning at central banks», Staff Working Paper No. 674 , *Bank of England*, 2017.

[10] Cuatro 54 [Online], «Los principales tipos de aprendizaje automático» Disponible aquí.

[11] C. Esparza, «Series temporales»,*CSIC*, págs. 3-30. Disponible aquí.

[12] T. Hastie, R. Tibshirani and G. Friedman «The elements of statistical learning: data mining, inference and prediction», *Springer*, págs. 587-605, 2009. Disponible aquí.

[13] H.S. Hota, R. Handa and A.K. Shrivas «Time Series Data Prediction Using Sliding Window Based RBF Neural Network», *International Journal of Computational Intelligence Research*, págs. 1145-1156, 2017. Disponible aquí.

[14] R.J.Hyndman et al, «Forecasting Functions for Time Series and Linear Models», *Package forecast*, 2023. Disponible aquí.

[15] R.J.Hyndman and G.Athanasopoulos, «Forecasting: Principles and Practice», *Texts*, 2018. Disponible aquí

[16] A. Liaw and M. Wiener, «Breiman and Cutler's Random Forests for Classification and Regression», *Package randomForest*, 2022. Disponible aquí.

[17] M.C. Medeiros, G.F.R. Vasconcelos, A. Veiga and E. Zilberman «Forecasting Inflation in a Data-Rich Environment: The Benefits of Machine Learning Methods», *Journal of Business & Economic Statistics*, 2019. Disponible aquí.

[18] J.M. Otero, «Econometría: Series temporales y predicción», *Editorial AC*, págs. 201-281, 1993.

[19] A.R.S. Parmezan, V.M.A. Souza and G.E.A.P.A Batista «Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model», *Information Sciences*, 2019. Disponible aquí.

[20] L.Pascual and E. Ruiz, «Predicciones de series temporales basada en Machine Learning: aplicaciones económicas y financieras», Nuevos métodos de predicción económica con datos masivos, págs. 189-214, *Funcas*, 2021. Disponible aquí.

[21] D. Peña, «Análisis de series temporales», *Alianza Editorial*, 2010.

Apéndice A

Anexo

A.1. Listado completo de variables utilizadas BDREMS.

- Producto Interior Bruto a precios de mercado (mill. € cons).
- Gasto en consumo final de los hogares y de las ISFSH (mill. € cons).
- Formación Bruta de Capital Fijo (mill. € cons).
- Exportaciones de bienes y servicios (mill. € cons).
- Importaciones de bienes y servicios (mill. € cons).
- Variación de existencias (mill. € cons).
- Stock de capital (mill. € cons).
- Deflactor P.I.B pm (2015=1).
- Deflactor del Gasto en Consumo final de las AAPP (2015=1).
- Deflactor de la F.B.C.F (2015=1).
- Deflactor de las exportaciones de bienes y servicios (2015=1).
- Deflactor de las importaciones de bienes y servicios (2015=1).
- Precios exteriores (2015=1).
- P.I.B clientes resto del mundo.
- Población (miles de personas).
- Población activa (miles de personas).
- Población de 16 años o más (miles de personas).
- Vacantes (miles de personas).

- Ocupados (miles de personas).
- Ocupados: puestos de trabajo equivalente a tiempo completo (miles de personas).
- Asalariados (miles de personas).
- Asalariados: puestos de trabajo equivalente a tiempo completo (miles de personas).
- Tasa de paro (%).
- Horas trabajadas (miles de horas).
- Remuneración de asalariados total (mill. € corr).
- Impuestos netos sobre los productos (mill. € cons).
- Energía (mill. € cons).
- Índice del precio de la energía (2015=1).
- Agregado monetario (M1) (mill. € corr).
- Agregado monetario (M3) (mill. € corr).
- Tipo de interés EEUU a 3 meses (%).
- Tipo de cambio nominal (USD por €).
- Tipo de cambio nominal (€ por USD).
- Deuda AAPP (mill. € corr).
- Activos financieros netos economía nacional (mill. €).
- Total recursos de las AA.PP (mill. €).
- Producción de mercado (mill. €).
- Pagos por otra producción no de mercado (mill. €).
- Impuestos sobre la producción y las importaciones (mill. €).
- Rentas de la propiedad (mill. €).
- Impuestos corrientes sobre la renta, el patrimonio, etc.(mill. €).
- Cotizaciones sociales (mill. €).
- Otras transferencias corrientes (mill. €).
- Transferencias de capital (mill. €).
- Total empleos de las AA.PP. (mill. €).

- Consumos intermedios AA.PP. (mill. €).
- Remuneración de los asalariados AA.PP. (mill. €).
- Otros impuestos sobre la producción. (mill. €).
- Subvenciones (mill. €).
- Rentas de la propiedad (mill. €).
- Impuestos corrientes sobre a la renta a pagar por las AA.PP. (mill. €).
- Prestaciones sociales distintas de las transferencias en especie (mill. €).
- Transferencias sociales en especie relacionadas con el gasto en productos suministrados a los hogares por productores de mercado (mill. €).
- Otras transferencias corrientes (mill. €).
- Transferencias de capital (mill. €).
- Adquisiciones menos cesiones de activos no financieros no producidos (mill. €).
- Adquisiciones menos cesiones de activos no financieros no producidos (mill. €).
- Capacidad/necesidad de financiación (mill. €).
- Capacidad/necesidad de financiación excluidas las ayudas a entidades financieras (mill. €).
- Prestaciones por desempleo (mill. €).
- Stock de capital público (mill. €).

A.2. Acceso al código del análisis

El código e instrucciones de uso están disponibles en el siguiente repositorio de Github: `https://github.com/ramonpac/TFG_Matematicas`. Para que funcione el código será necesario adaptar las rutas de importación a la ruta local que se esté usando, instalar las librerías necesarias y cargar el workspace de *R*: `imagen_resultados_TFG.RData` para poder replicar los resultados.

A.3. Cronograma organización del tiempo

Como no es posible insertar el cronograma por partes - ya que dificultaría su entendimiento - se adjunta completo en horizontal en la página siguiente y se recomienda hacer zoom.

TFG Matemáticas

Ramón Pacheco Murillo
Planificación temporal

nienzo: lu, 1/16/2023

1

GRÁFICO GANTT SIMPLE de Vertex42.com
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

TAREA	INICIO	FIN
Concretar tema del TFG + elección software	**16-1-23**	**31-1-23**
Búsqueda de tema para el trabajo (opción seleccionada: series temporales y aprendizaje automático)	16-1-23	19-1-23
Búsqueda de métodos de aprendizaje automático aplicables a series temporales	19-1-23	21-1-23
Pruebas de código con Redes Neuronales Recurrentes en Python (opción descartada)	21-1-23	25-1-23
Pruebas de código con la técnica Random Forest (RF) en R (opción seleccionada)	25-1-23	30-1-23
Establecimiento de los objetivos del TFG	30-1-23	31-1-23
Revisión de la literatura I	**1-2-23**	**16-2-23**
Conceptos básicos de Series Temporales	1-2-23	5-2-23
Modelos estacionarios lineales	6-2-23	13-2-23
Método de Box-Jenkins	14-2-23	16-2-23
Revisión de la literatura II	**17-2-23**	**9-3-23**
Conceptos básicos de Aprendizaje Automático	17-2-23	21-2-23
Base teórica de la Técnica Random Forest	22-2-23	26-2-23
Consideraciones para adaptar RF a datos de series temporales	27-2-23	4-3-23
Lectura de estudios que apliquen RF a la predicción de series temporales	5-3-23	9-3-23
Revisión de estudios aplicando RF a la predicción de variables económicas	27-2-23	9-3-23
Búsqueda de datos para realizar el estudio	**10-3-23**	**20-3-23**
Construcción de una amplia base de datos desde cero	10-3-23	14-3-23
Prueba de código para el modelo univariante usando la base de datos construida	15-3-23	15-3-23
Prueba de código para el modelo RF usando la base de datos construida (resultados deficientes)	16-3-23	16-3-23
Encotramos la base de datos BDREMS	17-3-23	17-3-23
Lectura, exploración y tratamiento de la base de datos BD rems	17-3-23	18-3-23
Prueba de código para el modelo univariante usando la base de datos BDREMS	19-3-23	19-3-23
Prueba de código para el modelo RF usando la base de datos BDREMS	20-3-23	20-3-23
Modelo ARIMA	**21-3-23**	**8-4-23**
Generación del código del análisis con R (paquetes stats y Forecast)	21-3-23	28-3-23
Representaciones gráficas con R	28-3-23	28-3-23
Redacción del apartado correspondiente de la memoria	28-3-23	8-4-23
Modelo Ranfom Forest	**1-4-23**	**30-4-23**
Establecimiento de las estretegias de entrenamiento (parte estática vs dinámica)	1-4-23	3-4-23
Generación del código del análisis con R para la parte estática (paquete randomForest)	3-4-23	10-4-23
Correcciones del código de la parte estática	10-4-23	12-4-23
Representaciones gráficas con R de la parte estática	12-4-23	13-4-23
Generación del código del análisis con R para la parte dinámica (paquete randomForest)	13-4-23	20-4-23
Correcciones del código de la parte dinámica	20-4-23	25-4-23
Representaciones gráficas con R de la parte dinámica	25-4-23	29-4-23
Ultimas correcciones/feedback	29-4-23	30-4-23
Redacción de los apartados correspondientes de la memoria	3-4-23	30-4-23
Redacción del resto de la memoria	**1-4-23**	**7-5-23**

Inserte nuevas filas ENCIMA de esta