UNIVERSIDAD MIGUEL HERNÁNDEZ

FACULTAD DE CIENCIAS SOCIALES Y JURÍDICAS DE ELCHE

GRADO EN ESTADÍSTICA EMPRESARIAL

TRABAJO FIN DE GRADO

CURSO ACADÉMICO 2018/2019

ESTRATEGIAS DE DECISIÓN PARA LA SELECCIÓN DE MODELOS DE PREDICCIÓN

Aurora Mula Leal

Tutor: José Vicente Segura Heras

ÍNDICE

1. RESUMEN

En este trabajo vamos a analizar los resultados obtenidos en la competición M4 por una combinación lineal de predicciones presentada al concurso. Para cada serie consideramos dos predicciones, obtenidas a partir de los datos sin transformar y transformados mediante logaritmo, combinando ambas mediante la inversa de sus errores de ajuste (sMAPE). El análisis del sMAPE a posteriori nos lleva a concluir que una selección más eficiente entre los tres tipos de predicciones propuestas para cada serie hubiera mejorado considerablemente dicho error. Hemos incluido también como posible opción el método Naïve.

Se propone en este trabajo un árbol de decisión, a partir de los errores de ajuste y los parámetros asociados a cada tipo de predicción (basados en el sMAPE), para seleccionar la mejor opción posible de las 4 incluidas para cada serie.

También se ha valorado el papel de otros parámetros como el estadístico U de Theil o el tamaño de la serie.

Palabras clave: Predicción, Sistema de apoyo a la toma de decisiones, series temporales, Competición M4.

2. ABSTRACT

This paper analyze the results obtained in the M4-Competition by a linear combination of predictions submitted to the competition. For each time series we consider two forecasts, obtained from the raw data and its log-transformed time series. A linear combination of both forecasts is built using as weights the inverse of the averaged fitting errors (sMAPE). The analysis of the sMAPE leads us to conclude that a more efficient selection among the three types of proposed forecasts for each series would have considerably improved this error. We have also included the Naïve method as a possible option.

A conditional tree is proposed in this paper, based on the adjustment errors associated with each type of forecast (based on the sMAPE), to select the best possible option from the 4 included for each series.

The role of other parameters such as Theil's U statistic or the size of the series has also be assessed.

Keywords: Forecast, decision support system (DSS), time series, M4-Competition.

3. INTRODUCCIÓN

3.1. ANTECEDENTES HISTÓRICOS DE LA COMPETICIÓN M

Las competiciones propuestas por el Dr. Spyrus Makridakis son estudios empíricos mediante los cuales se compara el rendimiento y la precisión de diferentes métodos de predicción en series temporales. En ellas participan numerosos expertos en el ámbito de predicción. El concurso consiste en lo siguiente: cada uno de los expertos realiza sus predicciones y, una vez realizadas, estas se evalúan y se comparan con los resultados de los otros expertos, así como con algunos de los métodos más simples, utilizados como puntos de referencia[1].

La primera competición se realizó en 1982 y es conocida como Competición M. En esta primera competición se utilizaron 1001 series temporales y 15 métodos de predicción (incluyendo nueve variaciones de estos). Una vez realizado el concurso, algunas de las principales conclusiones fueron las siguientes:

- Los métodos de predicción estadísticamente más sofisticados y complejos no proporcionan necesariamente predicciones más precisas que los métodos simples.
- La clasificación relativa del rendimiento de los diferentes métodos varía en función de la medida de precisión que se esté utilizando.
- Cuando varios métodos se combinan se obtiene una precisión mayor (en promedio) en comparación con cada método individual.
- La precisión de los diversos métodos depende de la longitud del horizonte de predicción involucrado.

Estos primeros resultados de la Competición M fueron posteriormente verificados y replicados por otros investigadores, los cuales obtuvieron conclusiones similares a las cuatro anteriores[2]. Además, estudios adicionales en los que utilizaron series de datos distintas, han demostrado la validez de las conclusiones de la Competición M. A pesar de ello, también se recibieron críticas de numerosos estadísticos, quienes refutaban

[1] https://www.mcompetitions.unic.ac.cy/m-competitions/
[2] https://www.mcompetitions.unic.ac.cy/m1/

algunas de las conclusiones establecidas. Fueron estas mismas críticas las que motivaron las siguientes competiciones M2 y M3. (Makridakis & Hibon, 2000)

La segunda competición, conocida como Competición M2, se llevó a cabo a una mayor escala que la primera. Se publicó una convocatoria de participación en el *International Journal of Forecasting*, se realizaron anuncios en el *International Symposium of Forecasting* y se enviaron invitaciones personales a diferentes expertos en metodología de series temporales.

La Competición M2 utilizó 29 series temporales, 23 de las cuales eran de las cuatro empresas colaboradoras en la competición y 6 series eran macroeconómicas.

La competición se realizó en tiempo real con el propósito de simular mejor las predicciones en los siguientes 3 aspectos:

1. Permitir que los expertos combinen su método de predicción con un criterio personal.
2. Permitir que los expertos contacten con las compañías colaboradoras para recopilar información y solicitar datos con el fin de obtener mejores predicciones.
3. Permitir que los expertos revisen sus predicciones para un próximo ejercicio y aprendan del trabajo realizado.

La organización de la competición fue la siguiente: en el verano de 1987 se envió el primer conjunto de datos a los expertos participantes. En octubre de ese año se enviaron los datos actualizados. A finales de noviembre, los expertos debían enviar las predicciones. Un año después se envió a los participantes un análisis de las predicciones y debían presentar un próximo pronóstico en noviembre de 1998. Finalmente, la evaluación de las predicciones se realizó a partir de abril de 1991, cuando las empresas colaboradoras conocían los datos reales.

Los resultados de la Competición M2 se publicaron en un artículo en 1993 en el que se confirmaban las conclusiones obtenidas en la primera competición. Por otro lado, muchos de los expertos participantes escribieron artículos detallando su experiencia en el

concurso. Sin embargo, los estadísticos teóricos seguían ignorando las implicaciones producidas en estas competiciones[3].

La tercera competición, Competición M3, pretendía replicar y ampliar las características de las competiciones anteriores a través de la inclusión de más expertos (en particular investigadores en el ámbito de redes neuronales), más métodos y más series temporales.

En esta competición se utilizaron un total de 3003 series temporales. Éstas incluían series anuales, trimestrales, mensuales, diarias y otras. Además, las series pertenecían a los siguientes dominios: micro, industria, macro, finanzas, demografía y otros[4].

Con el fin de garantizar suficientes datos para poder desarrollar un modelo preciso, se establecieron umbrales mínimos para el número de observaciones: 14 para las series anuales, 16 para las trimestrales, 48 para series mensuales y 60 para otras series. Además, en la Competición M3 se incluyeron todos los métodos utilizados en la Competición M más siete nuevos de las áreas de redes neuronales (Makridakis & Hibon, 2000).

En la Competición M3 se utilizaron cinco medidas para evaluar la precisión de los diferentes métodos de predicción: error porcentual absoluto medio simétrico (MAPE simétrico), clasificación promedio, error porcentual absoluto simétrico mediano (APE simétrico mediano), mejor porcentaje y RAE mediana[4].

Una vez más, las conclusiones de la Competición M3 fueron similares a las de las competiciones anteriores utilizando un conjunto de datos nuevo y más ampliado. (Makridakis & Hibon, 2000).

[3] https://www.mcompetitions.unic.ac.cy/m2/
[4] https://www.mcompetitions.unic.ac.cy/m3/

3.2. COMPETICIÓN M4

La Competición M4 es la continuación de las tres competiciones anteriores iniciadas hace más de 45 años, cuyo propósito era aprender cómo mejorar la precisión de las predicciones y cómo aplicar ese aprendizaje en la teoría y la práctica del ámbito predictivo. (Makridakis & Spilotis, 2018).

La Competición M4 se anunció en noviembre de 2017 y comenzó el 1 de enero de 2018. Ésta se dio por finalizada el 31 de mayo de 2018 y utilizó un conjunto extenso y diverso de series temporales para identificar los métodos de predicción más precisos. Se utilizaron 100.000 series temporales de la vida real con el fin de encontrar respuestas precisas y convincentes. Además, se incorporaron todos los métodos de predicción principales, incluidos los basados en Inteligencia Artificial (*Machine Learning*, ML), así como los estadísticos tradicionales[5].

Se registraron 248 personas, muchas de las cuales representaban a equipos de escuelas de negocios y/o finanzas, de departamentos o centros de investigación universitarios, de empresas de software de predicción o de analistas financieros, entre otros. En el apartado de predicciones puntuales sólo 50 propuestas fueron válidas. Esto implicó una tasa de participación del 20% sobre el total de registros.

El objetivo más importante de la Competición M4 ha sido "aprender cómo mejorar la precisión de la predicción y cómo se puede aplicar ese aprendizaje para avanzar en la teoría y la práctica de la predicción", proporcionando beneficios para aquellos que estén interesados en el campo (Makridakis & Spilotis, 2018).

La M4 es una competición abierta cuyas series están disponibles tanto en el sitio M4[6] como en el paquete de R M4comp2018 (Montero-Manso, Netto, & Talagala, 2018). Además, se ha solicitado a la mayoría de los participantes que depositen el código utilizado para generar los pronósticos en GitHub[7], con una descripción detalla de sus métodos. Esto implica que los individuos interesados y las organizaciones podrán descargar y utilizar la mayoría de los métodos M4, para beneficiarse de su mayor

[5] https://www.mcompetitions.unic.ac.cy/m4/
[6] https://www.mcompetitions.unic.ac.cy/the-dataset/[1]
[7] https://github.com/M4Competition/M4-methods

precisión. Además, los investigadores académicos pueden analizar los factores que afectan en la precisión de las predicciones para comprenderlos mejor y concebir nuevas formas de mejorar la misma.

Los cinco hallazgos principales de la M4 fueron los siguientes (Makridakis & Spilotis, 2018):

- Las combinaciones de métodos predominaron en la M4. De los 17 métodos más precisos, 12 eran combinaciones.
- Un enfoque "híbrido" que utilizó características estadísticas y de ML fue el que produjo, tanto las predicciones, como los intervalos de predicción más precisos.
- El segundo método más preciso combinaba siete métodos estadísticos y uno ML.
- Los dos métodos más precisos lograron especificar correctamente el 95% de los intervalos de predicción.
- Los seis métodos puros de ML que se presentaron en la M4 tuvieron un desempeño deficiente, ya que ninguno fue más preciso que el Comb (método utilizado como punto de referencia) y solo uno fue más preciso que el Naïve2.

Como conclusión global a partir de todo lo anterior, se puede decir que la precisión de los métodos estadísticos individuales o de ML es baja, mientras que los enfoques híbridos y las combinaciones de métodos son el camino a seguir para mejorar la precisión de las predicciones. Con todo ello, las 100.000 series temporales de la Competición M4 se convertirán en un campo de prueba en el que los investigadores puedan experimentar y descubrir nuevos enfoques de pronósticos cada vez más precisos (Makridakis & Spilotis, 2018).

Es esta línea en la que se enmarca este trabajo, aunque centrado en las 23.000 series anuales de la Competición M4. En el apartado de Metodología se presenta el método utilizado para obtener las predicciones para dicha competición, y se complementa con un modelo, basado en árboles de decisión, para decidir cuál debería ser el modelo seleccionado de los cuatro citados para proponer las predicciones. El apartado 5 muestra los resultados obtenidos con esta metodología, mientras que en el apartado 6 se presentan las conclusiones de este trabajo.

4. METODOLOGÍA

4.1. MODELO PROPUESTO

El modelo utilizado para predecir las series temporales involucradas en la Competición M4 es SIOPRED (Bermúdez, Segura, & Vercher, 2008). Este sistema de soporte de predicción aplica un esquema basado en la optimización que calcula conjuntamente los parámetros de suavizado y las condiciones iniciales para un método de suavizado exponencial generalizado. El sistema resuelve varios problemas de programación no lineal y utiliza un enfoque de multicriterio difuso (Bermúdez, Segura, & Vercher, 2006) que garantiza predicciones precisas para el horizonte de planeación.

Los resultados que se muestran aquí se basan en la combinación de dos predicciones proporcionadas automáticamente por SIOPRED. En particular, para cada serie, SIOPRED considera los datos sin procesar y los mismos transformados mediante el logaritmo neperiano, y proporciona, para cada uno, predicciones puntuales aplicando el modelo aditivo de Holt-Winters con tendencia amortiguada (Bermudez, Segura, & Vercher, 2007) (Vercher, Corberan-Vallet, Segura, & Bermúdez, 2012). Una vez realizado lo anterior, se construye una combinación lineal de ambas predicciones utilizando como ponderación la inversa de los respectivos errores de ajuste (sMAPE a priori).

A continuación, se explica de forma detallada los pasos a seguir para resolver los problemas de optimización:

1. **Técnicas multicomienzo:**
   - Consideramos 12 combinaciones de los parámetros de suavizado como puntos iniciales del algoritmo de optimización para evitar caer en mínimos locales (α, β, Φ)

	1	2	3	4	5	6	7	8	9	10	11	12
α	0,01	0,30	0,30	0,30	0,30	0,30	0,50	0,70	0,70	0,70	0,70	0,90
β	0,01	0,001	0,30	0,30	0,30	0,50	0,50	0,001	0,001	0,30	0,70	0,90
Φ	0,90	0,30	0,30	0,50	0,70	0,70	0,90	0,10	0,70	0,50	0,90	0,95

- Para las componentes iniciales de nivel (F_0) y, tendencia (b_0) utilizamos la propuesta de cálculo de Makridakis, S, (Makridakis, Wheelwright, & Hyndman, 1997),

$$F_0 = D_1$$
$$b_0 = \frac{D_2 - D_1}{2}$$

- Para el valor de compromiso λ que utilizamos para resolver el problema multiobjetivo difuso consideramos cuatro valores iniciales (0,1; 0,4; 0,7; 0,9). Resolvemos, por tanto, 12 problemas no lineales, 4 para cada mejor solución obtenida con una medida de error.

2. Problema de optimización no lineal con un único objetivo.

- Resolvemos 12 problemas de programación no lineal para cada medida de error de ajuste (SMAPE, RMSE, MAD) mediante la dll Solver. Esta librería utiliza el método de optimización no lineal GRG2 desarrollado por Leon Lasdon del MSIS Department de la Universidad de Texas en Austin y Allan Waren, del Departamento de Computación y Ciencias de la Información de la Universidad del Estado de Cleveland, e implementado por Daniel Fylstra de Frontline Systens y John Whatson de Software Engines.

- El número de variables es 5 al considerar la tendencia amortiguada, y las cotas de las mismas:

0	≤	α	≤	0,3
0	≤	β	≤	0,3
0	≤	Φ	≤	1
-∞	≤	b_0	≤	+∞
-∞	≤	F_0	≤	+∞

- Las funciones objetivo φ a minimizar son:
  - $SMAPE = \frac{200}{n}\sum_{j=1}^{n}\frac{|D_{j+1}-(F_j+b_j)|}{|D_{j+1}|+|F_j+b_j|}$

- $RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(D_{j+1} - (F_j + b_j))^2}$
- $MAD = \frac{1}{n}\sum_{j=1}^{n}\left|D_{j+1} - (F_j + b_j)\right|$

- Las restricciones del problema son:
  - $\beta \leq \phi$

- Las ecuaciones recursivas que nos relacionan todas las variables en cada periodo, para j=0, …, n-1, son:
  - $e_{j+1} = D_{j+1} - (F_j + b_j)$
  - $F_{j+1} = \alpha(D_{j+1}) + (1-\alpha)(F_j + b_j)$
  - $b_{j+1} = \beta(F_{j+1} - F_j) + (\phi - \beta)b_j$

3. Problema de optimización no lineal multiobjetivo difuso.

- Resolvemos 12 problemas de programación no lineal mediante la dll Solver además de los 36 anteriores.

- Resolvemos el siguiente sistema de ecuaciones para obtener los valores de c_i asociados a cada uno de los errores de ajuste considerados:

$$0{,}5 = \frac{1 - e^{\left\{-b_i \frac{Máximo(\varphi_{i,j}) - Mediana(\varphi_{i,j})}{Máximo(\varphi_{i,j}) - Mínimo(\varphi_{i,j})}\right\}}}{1 - e^{\{-b_i\}}} \quad i = 1,2,3; j = 1, \dots, 12$$

Donde $\varphi_{i,j}$ es el valor obtenido para la función objetivo i en el problema no lineal j.

- Añadimos tres restricciones al problema de optimización no lineal y maximizamos el valor de λ, restringido al intervalo [0,1]:

$$\frac{1 - e^{\left\{-b_i \frac{Máximo(\varphi_{i,j}) - Mediana(\varphi_{i,j})}{Máximo(\varphi_{i,j}) - Mínimo(\varphi_{i,j})}\right\}}}{1 - e^{\{-b_i\}}} \geq \lambda \quad i = 1,2,3; j = 1, \dots, 12$$

4. Otras medidas calculadas:

- $U1 = \sqrt{\frac{\sum_{j=2}^{n}\left(\frac{D_{j+1}-a_j}{D_j}\right)^2}{\sum_{j=2}^{n}\left(\frac{D_{j+1}-D_j}{D_j}\right)^2}}$ U de Theil para el método Naïve

Donde $a_j = Fj + bj$

El índice de Theil es una medida de precisión de un modelo de predicción que compara el modelo propuesto con el método Naïve.

4.2. ÁRBOL DE DECISIÓN

4.2.1. ORIGEN

El almacenamiento y análisis de datos se ha convertido en una tarea rutinaria de los sistemas de información de las organizaciones. Esto es aún más evidente en las empresas de la nueva economía, el comercio, la telefonía, el marketing directo, etc. Los datos almacenados son un tesoro para las organizaciones, es donde se guardan las interacciones con los clientes o la contabilidad de sus procesos internos, es decir, representan la memoria de la organización. Sin embargo, no es suficiente con tener memoria, es necesario pasar a la acción inteligente sobre los datos para extraer la información que almacenan. Este es el objetivo de la Minería de Datos.

Pero ¿qué es exactamente la Minería de Datos? Se puede definir como: "Iterative process of extracting hidden predictive patterns from large data- bases, using AI technologies as well as statistics techniques" (Mena, 1999). Esta definición abarca las dos raíces principales de la Minería de Datos: la Inteligencia Artificial (en particular *Machine learning*) y la Estadística.

A lo largo de la historia se han desarrollado gran cantidad de técnicas de Minería de Datos capaces de abordar cualquier problema sobre análisis de datos. Estas técnicas están basadas principalmente en métodos estadísticos y algunos ejemplos podrían ser: el análisis factorial descriptivo, las técnicas de clustering, las series temporales o las redes

bayesianas. Entre las mismas podemos destacar uno de los principales modelos de clasificación y predicción para cantidades ingentes de datos: los árboles de decisión (Aluja, 2001).

La representación gráfica, mediante el ordenamiento de un árbol, suele ser la mejor opción para mostrar una solución cuando se plantean problemas de decisión "secuenciales" o "encadenados", es decir, aquellos problemas en los que cada posible alternativa conllevará una nueva decisión. La utilización de un diagrama de árbol surge de la Teoría de Juegos de John von Neumann y Oskar Morgenster (1944), quienes recurren a este tipo de gráfico para representar la estructura temporal de un juego en forma extensiva (desde el principio -*primera jugada*-, hasta el final -*última jugada*-), como por ejemplo ante un juego de suma cero entre dos jugadores, como puede ser el denominado "Pares y nones" [8].

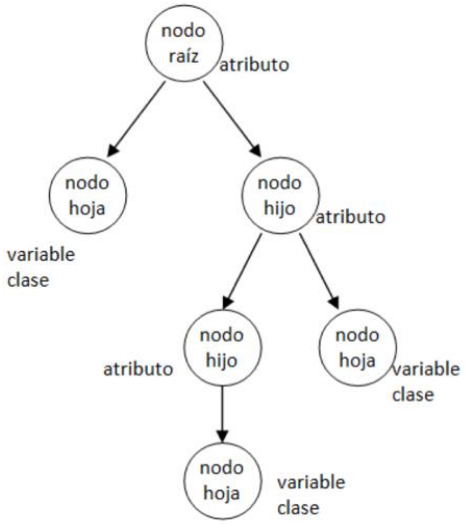
4.2.2. DESCRIPCIÓN Y TIPOS

Un árbol de decisión es una herramienta analítica para la selección, estructuración y evaluación de problemas bajo un ambiente de incertidumbre. Permite además evaluar planes de acción, efectuar valoración de consecuencias, obtener cálculo de probabilidades y establecer simulaciones. La construcción de árboles de decisión se basa en la existencia de escenarios donde los agentes actúan interactiva y consecutivamente, es decir, implican un análisis dinámico basado en la sucesión de eventos consecutivos donde el conjunto de acciones futuras se determina por acciones y decisiones presentes. (César & Molina, 2016).

El propósito que se persigue es que en cada evaluación sucesiva de una función de decisión se reduzca la incertidumbre en la identificación del patrón desconocido. Su principal ventaja es la facilidad de interpretación (Goddard, Cornejo, Martínez, Martínez, Rufiner, & Acevedo, 1995).

[8] https://www.academia.edu/14386108/Teor%C3%ADa_de_la_Decisión_-_El_árbol_de_decisión_-_Su_aplicación_en_situaciones_de_decisión_con_alternativas_interdependientes

Figura 1: Estructura general de un árbol de decisión. Fuente Externa



Un árbol se representa gráficamente por un conjunto de nodos (variables de entrada), hojas (valores de la variable salida) y ramas (grupos de entradas en las variables de entrada). El nodo principal o raíz es el atributo a partir del cual se inicia el proceso de clasificación; los nodos internos corresponden a cada una de las preguntas acerca del atributo en particular del problema. Cada posible respuesta se representa mediante un nodo hijo. Las ramas que salen de cada uno de estos nodos se encuentran etiquetadas con los posibles valores del atributo. Los nodos hoja o nodos finales corresponden a una decisión, la cual coincide con una de las variables clase del problema a resolver. (Figura 1) (Barrientos, et al., 2009).

"Los árboles de decisión también son útiles, porque no sólo permiten considerar el riesgo en cada una de las etapas, sino que te ayudan a diseñar la mejor respuesta, dado un resultado determinado (si ocurre x, habría que hacer z). Vincular acciones y opciones a los resultados de eventos inciertos, a través de árboles de decisión, permite a las empresas considerar cómo actuar hoy ante riesgos y circunstancias diferentes. Como consecuencia, las empresas están preparadas para cualquier resultado que pueda suceder, y así, no verse sorprendidas" (César & Molina, 2016).

La partición se puede realizar con muchos enfoques diferentes y, por lo tanto, existen una gran variedad de algoritmos para construir estos árboles: AID, CHAID, CART, CTREE, C4, Random Forest, etc. Todos ellos pueden clasificarse en aquellos que hacen regresión, aquellos que hacen clasificación y aquellos que hacen ambas técnicas. Existen diferentes características que difieren para cada algoritmo: divisiones binarias o divisiones múltiples, el criterio de parada (cuando el árbol ya no crece) o la filosofía de cómo determinar qué variable se debe tomar para el siguiente paso y donde dividirla (Molnar, 2013).

4.2.3. CTREE

Dentro de los árboles de decisión encontramos un tipo especial llamado árboles condicionales (CTREE-Conditional Tree). En los árboles de inferencia condicional la selección de las variables se realiza en dos fases, primero se formula una hipótesis global de independencia en términos de hipótesis parciales. Es decir, se evalúa si existe dependencia entre la variable respuesta y cada una de las variables explicativas, en caso de no poder rechazar la hipótesis nula de independencia planteada se detiene el proceso recursivo. En contraposición, si la hipótesis global de independencia es rechazada, el siguiente paso es medir el nivel de asociación entre la variable dependiente y cada una de las variables explicativas, lo cual permite generar nuevas divisiones del árbol de manera secuencial (Hothorn, Hornik, & Zeileis, 2006). Algunas de las ventajas de esta herramienta son: la fácil interpretación de los modelos debido a la estructura del árbol resultante (Molnar, 2013) y su versatilidad en el caso de que existan relaciones no lineales y en el manejo de variables numéricas y categóricas de forma simultánea.

Estos modelos de árboles de inferencia condicional surgen con el objetivo de solucionar básicamente dos graves problemas: el sobreajuste y el sesgo de selección de covariables con muchas divisiones posibles o valores perdidos de los modelos de árboles tipo CART (Classification and Regression Trees) (Hothorn, Hornik, & Zeileis, 2006).

Como se afirma anteriormente, existen diferentes tipos de árboles de decisión, sin embargo, en este trabajo se va a realizar un árbol de clasificación mediante inferencia condicional.

4.2.4. ALGORITMO

Denotamos por $X = (X_1, X_2, \ldots, X_m)$ el conjunto de variables de entrada y por Y la variable respuesta:

1. Para cada variable de entrada X_i se realiza el siguiente contraste de hipótesis:

$$\left.\begin{array}{l} \text{H0: La variable } X_i \text{ es independiente de la variable Y} \\ \text{H1: La variable } X_i \text{ presenta asociación con la variable Y} \end{array}\right\}$$

El nivel de confianza suele estar entre el 95% y el 99,9% de tal manera que 1-α = (0.95-0.999). Como consecuencia, cuanto menor sea α, más estricto será el criterio para rechazar la hipótesis nula y elegir la variable X_i (Martín, 2017). El algoritmo empieza con todo el conjunto de datos y comprueba si se puede dividir mediante el contraste de hipótesis anterior. Si se rechaza la hipótesis nula, se elige la variable con asociación más fuerte respecto a la variable respuesta y comienza la división binaria. Los pasos se repiten dentro de las dos nuevas divisiones (Molnar, 2013).

a) Si se rechaza la hipótesis nula para una única X_i, esta variable de entrada se selecciona para realizar el siguiente corte.
b) Si se rechaza la hipótesis nula para más de una variable de entrada, se selecciona para el siguiente corte aquella que cuente con mayor asociación con Y, es decir, tenga el menor p-valor.
c) Si la hipótesis nula no puede ser rechazada para ninguna de las variables de entrada, se paran los cortes y por lo tanto se detiene el crecimiento del árbol.

2. En el caso de que no se detenga el crecimiento del árbol es debido a que existe una variable X_i para realizar el siguiente corte. Esta variable se divide en dos subconjuntos, a través de un valor V, que deben ser mutuamente excluyentes. Este valor V pertenece a X_i y se elige de tal modo que la discrepancia entre el subgrupo que contiene V y el que no sea la máxima posible. De esta manera, se obtienen los subconjuntos $\phi_1 = \{X_i \leq V\}$ y $\phi_2 = \{X_i > V\}$.

3. Se repiten los pasos 1 y 2 hasta que se cumpla el criterio de parada cuando no se pueda rechazar la hipótesis nula. Debe tenerse en cuenta que cuanto mayor sea α, será más fácil rechazar la hipótesis nula y el árbol resultante será de menor profundidad (Martín, 2017).

4.2.5. CTREE EN R

Como se ha señalado en el apartado anterior, en este trabajo se realiza un árbol de clasificación mediante el algoritmo de árboles condicionales. Este algoritmo se ejecuta mediante el comando *ctree* de la librería *partykit* de R.

Ctree es una clase no paramétrica de árboles de regresión que incorporan modelos de regresión estructurados en árbol con procedimientos de inferencia condicional. Es aplicable a todo tipo de problemas de regresión, incluyendo variables de respuesta nominales, ordinales, numéricas, censuradas, multivariantes y escalas de medición arbitrarias de las covariables. Esta función se encuentra con sus versiones mejoradas en el paquete *partykit* (Hothorn, Hornik, & Zeileis, 2015). *Partykit* presenta un conjunto de herramientas con infraestructura para representar, resumir y visualizar una amplia gama de modelos de regresión y de clasificación con estructura de árbol.

La forma genérica de la función es la siguiente:

ctree(formula, data, subset, weights, na.action = na.pass, offset, cluster, control = ctree.control(...), ytrafo = NULL, coverged = NULL, scores = NULL, doFit = TRUE,..)

donde:

- *formula:* descripción del modelo a ser ajustado.
- *data:* dataset que contiene las variables del modelo.
- *subset:* vector opcional que especifica un subconjunto de observaciones que se utilizarán en el proceso de ajuste.
- *weights:* vector opcional de pesos para ser utilizados en el proceso de ajuste. Solo se permiten pesos de valores enteros no negativos.
- *na.action*: una función que indica lo que debería suceder cuando los datos contienen un valor perdido. El valor por defecto es na.pass que no realiza ningún cambio sobre los valores perdidos.
- *offset*: un vector opcional de valores de compensación
- *cluster:* factor opcional que indica clusters independientes.
- *control*: lista de parámetros de control en el ajuste. Se añaden mediante la función ctree_control()
- *ytrafo:* lista opcional de funciones que se aplican a las variables respuesta antes de probar su asociación con las variables explicativas. Esta transformación solo se realiza una vez para el nodo raíz y no tiene en cuenta los pesos. Alternativamente, ytrafo puede ser una función de datos y pesos. En este caso, la transformación se calcula para cada nodo con los pesos correspondientes.

- *coverged:* es una función opcional para verificar los criterios definidos por el usuario antes de implementar las divisiones.
- *scores:* lista opcional de puntuaciones para adjuntar a factores ordenados
- *doFit:* si es FALSO, el árbol no está ajustado.

Todas las características detalladas anteriormente son experimentales y la interfaz de usuario decide si efectuar algún cambio en cualquiera de ellas o no, en función de sus preferencias.

Es interesante destacar de entre todas ellas la función *ctree_control*, que es aquella que se encarga de controlar varios parámetros del ajuste los árboles de inferencia condicional en *ctree.* A continuación, se detalla la forma genérica que adquiere la función y se describen algunas de las opciones de esta.

ctree_control(teststat = c("quadratic", "maximum"), splitstat = c("quadratic", "maximum"),splittest = FALSE,testtype = c("Bonferroni", "MonteCarlo", "Univariate", "Teststatistic"),pargs = GenzBretz(),nmax = c(yx = Inf, z = Inf), alpha = 0.05, mincriterion = 1 - alpha, logmincriterion = log(mincriterion), minsplit = 20L, minbucket = 7L, minprob = 0.01, stump = FALSE, lookahead = FALSE, MIA = FALSE, nresample = 9999L, tol = sqrt(.Machine$double.eps),maxsurrogate = 0L, numsurrogate = FALSE, mtry = Inf, maxdepth = Inf, multiway = FALSE, splittry = 2L, intersplit = FALSE, majority = FALSE, caseweights = TRUE, applyfun = NULL, cores = NULL, saveinfo = TRUE, update = NULL, splitflavour = c("ctree", "exhaustive"))

De entre los argumentos de *ctree_control* encontramos *teststat,* que indica el tipo de prueba estadística que se desea aplicar para la selección de variables o *splitstat,* que especifica el tipo de prueba estadística que se aplicará para la selección de los puntos de división, entre muchos otros (Hothorn & Zeileis, 2011).

4.2.6. VARIABLES Y MODELO

Una vez detallada la función *ctree* con sus respectivas características, se procede a describir las variables implicadas en el modelo resultante.

En nuestro caso, hemos etiquetado la variable respuesta como variable *grupo,* y presenta 4 niveles, es decir, 4 posibles resultados en función del método que tiene asociado el menor error (sMAPE) a posteriori en cada una de las series:

1. Método Naïve: este método utiliza como predicción el dato observado en el periodo anterior.
2. Datos: utilizamos el esquema de predicción SIOPRED sobre los datos originales.
3. Logaritmo de los datos: utilizamos el esquema de predicción SIOPRED sobre los datos transformados mediante el logaritmo.
4. Método M4: planteamos una combinación ponderada de las predicciones obtenidas en los casos 2 y 3.

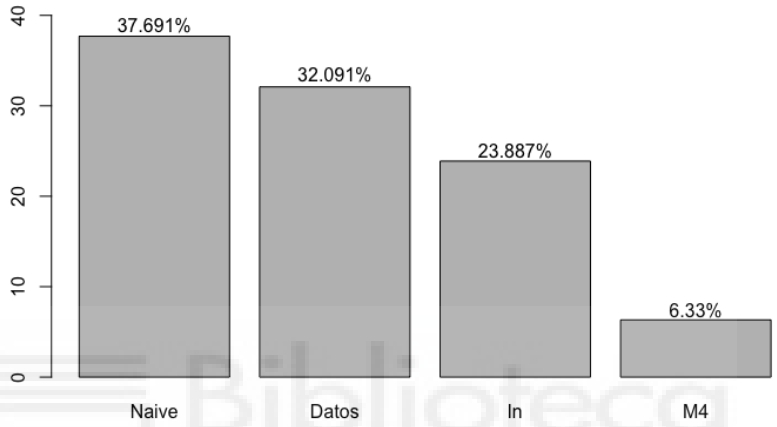


Figura 2: Clasificación de las series anuales según el método que produce el menor sMAPE a posteriori. Elaboración propia.

En muchas series el sMAPE a posteriori es muy parecido con las cuatro estrategias planteadas, siendo las diferencias de milésimas. No obstante, se ha preferido contemplar aquella estrategia asociada al mínimo.

Las principales variables explicativas, medidas en una escala numérica, que se han utilizado en la construcción del modelo son las siguientes:

- dsMAPEa: error a priori para las predicciones con los datos originales.
- lnsMAPEa: error a priori para las predicciones con el logaritmo de los datos.
- M4sMAPEa: error a priori para las predicciones del modelo M4.
- d_alfa: parámetro del modelo de suavizado asociado al nivel medio con los datos originales.
- d_beta: parámetro del modelo de suavizado asociado a la tendencia con los datos originales.
- dϕ: parámetro del modelo de suavizado asociado a la tendencia amortiguada con los datos originales.

- ln_alfa: parámetro del modelo de suavizado asociado al nivel medio con el logaritmo de los datos.
- ln_beta: parámetro del modelo de suavizado asociado a la tendencia con el logaritmo de los datos.
- lnϕ: parámetro del modelo de suavizado asociado a la tendencia amortiguada con el logaritmo de los datos.
- n: número de observaciones de cada serie

Una vez detalladas cada una de las variables, se procede a realizar el modelo. En primer lugar, se necesita un set de entrenamiento para generar un modelo predictivo y un set de prueba para comprobar la eficacia de este modelo. Se utilizan dos funciones de la librería *dplyr*:

- *sample_frac():* se obtiene un subconjunto de datos de entrenamiento. En esta función se introduce el porcentaje de datos que se desea obtener, en este caso el 70%.
- *setdiff:* se obtiene el subconjunto de datos complementario al anterior, en este caso el 30% restante.

Con los datos de entrenamiento, se van creando diferentes modelos con las distintas variables disponibles. En el siguiente apartado se describen los resultados obtenidos.

5. RESULTADOS

En este apartado se procede a detallar las diferentes pruebas que se han desarrollado sobre las 23.000 series anuales de la competición M4. Con los análisis previos oportunos, las pruebas realizadas nos llevan a la selección de un árbol de decisión como la herramienta más adecuada, capaz de clasificar cada serie temporal en función del mejor modelo de predicción de entre los cuatro propuestos.

En primer lugar, se considera oportuno crear diferentes variables con el fin de averiguar si estas aportan una información adicional a las variables originales. Para los tres tipos de predicciones de cada serie, se calcula el sMAPE a priori asociado únicamente a los 5 últimos periodos en el ajuste, así como también el correspondiente al considerar del 6º al 10º de los últimos periodos. Por otro lado, también se añade el método Naïve como posible alternativa de predicción, para este método también se calcula el error cometido. Con todas estas variables resultantes, se intentan ajustar diferentes modelos de regresión que expliquen el comportamiento de los errores a posteriori. Estos modelos no consiguen explicar más de un 40% de la variabilidad de los sMAPE a posteriori y, tras analizar los residuos, se llega a la conclusión de que ninguno de ellos es válido.

Con todo ello, se decide tomar otra vía de análisis: ahora nuestro punto de partida está en un 10,85 de sMAPE a posteriori. Es decir, si para cada serie nos quedáramos con el menor error a posteriori de las 4 posibles alternativas, se conseguiría reducir el sMAPE de un 15,40 a un 10,85. Hay que tener en cuenta que la mejor propuesta en el bloque de series anuales del concurso presentó un sMAPE de 13,18.

Para intentar alcanzar este objetivo, se empiezan a considerar diferentes alternativas. En primer lugar, se clasifican las series en función de sus tamaños muestrales y en función de los errores a priori, con el fin de observar cómo afectan estas variables a los errores. De esta manera se van creando diferentes tablas descriptivas que nos resumen los errores cometidos en función de estas clasificaciones. Finalmente, se llega a la conclusión de que se está construyendo una especie de árbol de decisión.

Este es el motivo que nos lleva a la selección de un modelo de árbol de inferencia condicional como mejor alternativa para alcanzar el objetivo planteado. La función utilizada se explica en el apartado anterior y, como se afirma en el mismo, se definen diferentes modelos para construir el árbol, los cuales quedan resumidos en la tabla 1.

Tabla 1: Resultados de los modelos ajustados con la función ctree(). Elaboración propia

	sMAPE		
MODELO	Entrenamiento	Prueba	Total
d_α + d_β+ dϕ	14,27	14,43	14,318
d_α + d_β+ dϕ+ dU1	14,21	14,49	14,294
d_α + d_β+ln_α+ln_β +dϕ+ lnϕ	14,34	14,58	14,412
ln_α + ln_β+ lnϕ	14,52	14,84	14,616
n + lnsMAPEa + lnsMAPE5 + M4sMAPEa	14,56	14,85	14,647
n+ dsMAPE5 + lnsMAPEa + M4sMAPEa	14,44	14,86	14,566
dsMAPEa + dsMAPE5 + lnsMAPEa	14,6	14,98	14,714
min + dsMAPEa + dsMAPE5 + lnsMAPEa	14,6	14,98	14,714
dsMAPEa + lnsMAPEa + M4sMAPEa	14,69	15,08	14,807
n + dsMAPEa + lnsMAPEa + M4sMAPEa	14,69	15,1	14,813
dsMAPEa + lnsMAPEa + M4sMAPEa + dsMAPE5 + lnsMAPE5 + M4sMAPE5	14,71	15,17	14,848
dsMAPEa + dsMAPE5 + dsMAPE10	14,93	15,27	15,032
d_β + ln_β	14,97	15,33	15,078
dsMAPEa + lnsMAPEa + M4sMAPEa + dsMAPE5 + lnsMAPE5 + M4sMAPE5 + dsMAPE10 + lnsMAPE10 + M4sMAPE10	14,87	15,38	15,023
dsMAPE5 + lnsMAPE5 + M4sMAPE5	15,22	15,7	15,364
dsMAPE10 + lnsMAPE10+ M4sMAPE10	15,59	15,87	15,674
d_α + ln_α	15,87	16,11	15,942

* sMAPEM4p(entrenamiento)=15.34; sMAPEM4p(prueba)=15.51

La columna *Entrenamiento* recoge el error cometido en las 16.100 series que conforman el conjunto de entrenamiento. La columna *Prueba* hace referencia al error cometido en el set de prueba (6900 series) mediante el modelo *ctree*. La columna *Total* recoge el error total ponderado que cometeríamos aplicando este criterio.

De entre todos los modelos propuestos, se decide escoger el primero como mejor alternativa. Sin embargo, se puede observar que el sMAPE total cometido con el primer modelo es ligeramente superior al sMAPE del segundo modelo, a pesar de ello, se considera que la diferencia de errores es despreciable en comparación a la complejidad que adquiere el modelo añadiendo una nueva variable.

Finalmente, como se aprecia en la tabla 1, el modelo escogido es aquel formado por α, β y ϕ con los datos originales. A partir de este modelo, el árbol resultante es el que se muestra en la figura 3.

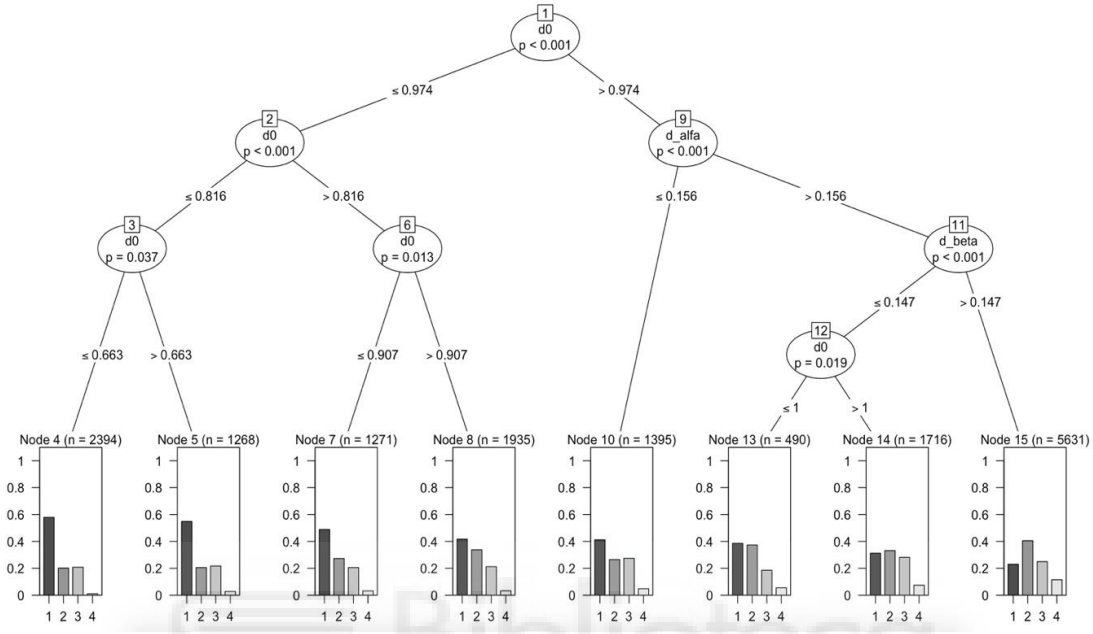


Figura 3: Resumen modelo ctree. Elaboración propia

En cada nodo de entrada se realiza un test de independencia de las variables explicativas (α, β y ϕ) con la variable respuesta (grupo), para cada uno de ellos se obtiene un pvalor. Si se rechaza la hipótesis nula, esta variable de entrada se utiliza para realizar la siguiente partición en el árbol. Cuando la hipótesis nula no puede ser rechazada para ninguna variable de entrada, se detiene el crecimiento del árbol y se obtienen los nodos finales. El funcionamiento árbol queda explicado de una forma más detallada en el apartado 4.2.

En el árbol resultante mediante nuestro modelo se obtienen 8 nodos de clasificación. Se observa que para los nodos 4, 5, 7, 8, 10 y 13 existe una mayor probabilidad de clasificación de las series en el grupo 1; mientras que los nodos 14 y 15 clasifican las series en el grupo 2 con una mayor probabilidad. Es decir, el árbol de inferencia condicional clasifica las 16.100 series del set de entrenamiento de la siguiente manera: 8.753 series son ajustadas mediante el método Naïve (grupo 1) y 7.347 son ajustadas con los datos (grupo 2).

Sin embargo, cada nodo final del árbol representa la probabilidad de clasificación en cada grupo y se puede observar que esta no representa más del 0,6 en ningún de ellos. Es decir, no existe una probabilidad de clasificación segura, es posible que se ajuste alguna serie mediante un método que no sea el más adecuado. Por ejemplo, en el nodo 4, existe un 60% de clasificación en el grupo 1, sin embargo, el grupo 2 y 3 presentan un 20% de probabilidad de que alguna serie de dicho nodo pueda ser ajustada mediante alguno de ellos. Por lo tanto, existirán algunas series que se clasifiquen en el grupo 1 de forma errónea, esto mismo ocurre con el grupo 2.

Todo esto puede resumirse de una manera más detallada en la Tabla 2. De las 8.753 series clasificadas en el método Naïve, 2.295 series se ajustarían mejor mediante los datos, 1.921 se ajustan mejor mediante el logaritmo de los datos y 260 con el modelo M4. Por otro lado, de las 7.347 series ajustadas con los datos según nuestro modelo, 1.835 series deberían ajustarse mediante el método Naïve, 1.887 series con el logaritmo de los datos y 771 con el modelo M4. De esta forma estaríamos clasificando de forma correcta el 44,29% de las series. Es importante destacar que en muchas ocasiones los errores a posteriori cometidos con varias de las estrategias comentadas anteriormente son muy parecidos, por lo que optar por cualquiera de ellas es igual de válido. A pesar del error cometido en el ajuste de las series, este modelo es el que proporciona mejores resultados en comparación con todas las pruebas realizadas.

Tabla 2: Ajuste predicciones set de entrenamiento. Elaboración propia

Ajuste	1	2	3	4
1	4277	2295	1921	260
2	1835	2854	1887	771
3	0	0	0	0
4	0	0	0	0

Por otro lado, también se puede obtener la misma tabla para las predicciones del set de prueba. La clasificación se observa en la tabla 3 y presenta también un 44,29% de acierto.

Tabla 3: Ajuste predicciones set de prueba. Elaboración propia

Predicciones	1	2	3	4
1	1805	981	851	103
2	752	1251	835	322
3	0	0	0	0
4	0	0	0	0

6. CONCLUSIONES

Hemos encontrado una herramienta de clasificación que nos permite mejorar la selección de la mejor estrategia de predicción para series temporales sin estacionalidad. La combinación de técnicas estadísticas y de machine learning permite mejorar los resultados finales y va en la línea de lo que proponen algunos autores (Makridakis, Spiliotis, & Assimakopoulos, 2018). El modelo de selección ajustado nos permite, a priori, añadir fácilmente la estacionalidad como un posible parámetro más del mismo, aunque estas pruebas las dejamos para un trabajo posterior.

No obstante, debemos de profundizar en varios aspectos como la estrategia de transformación de los datos originales, los criterios de combinación de predicciones o los tipos de series, para acercarnos a la selección óptima del método de predicción.

7. BIBLIOGRAFÍA

1. Aluja, T. (2001). La minería de datos, entre la estadística y la inteligencia artificial. *Qüestiió* , *25* (3), 479-498.
2. Barrientos, R., Cruz, N., Acosta, H., Rabatte, I., Gogeascoechea, M., Pavón, P., y otros. (2009). Árboles de decisión como herramienta en el diagnóstico médico. *Revista Médica de la Universidad Veracruzana* , 19-24.
3. Bermúdez, J., Segura, J., & Vercher, E. (2006). A decision support system methodology for forecasting of time series based on soft computing. *Computational Statistics & Data Analysis* , *51* (1), 177-191.
4. Bermúdez, J., Segura, J., & Vercher, E. (2008). SIOPRED: a prediction and optimisation integrated system for demand. *TOP* , *16* (2), 258-271 .
5. Bermudez, J., Segura, J., & Vercher, E. (2007). Holt-winters forecasting: An alternative formulation applied to UK air passenger data. *Journal of Applied Statistics* , *34* (9), 1075-1090.
6. César, J., & Molina, J. (2016). Breves consideraciones acerca de la importancia de los árboles de decisión en el análisis de carteras. *Revista de la Facultad de Ciencias Económicas y Administrativas* , *27* (1), 11-33.
7. Goddard, J., Cornejo, J., Martínez, F., Martínez, A., Rufiner, H., & Acevedo, R. (1995). Redes Neuronales y Árboles de Decisión: Un Enfoque Híbrido. *Memorias del Symposium Internacional de Computación* , 1-7.
8. Hothorn, T., & Zeileis, A. (2011). partykit: A Toolkit for Recursive Partytioning. *R package vignette version 0.1-2* , 1-50.
9. Hothorn, T., Hornik, K., & Zeileis, A. (2015). ctree: Conditional inference trees. *The Comprehensive R Archive Network* , 1-7.
10. Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics* , *15* (3), 651-674.
11. Makridakis, S., & Hibon, M. (2000). The M3-Competition: results, conclusions and implications. *International Journal of Forecasting* , 451-476.
12. Makridakis, S., & Spilotis, E. (2018). The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting* , 802-808.
13. Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *Plos One* , *13* (3), e0194889.
14. Makridakis, S., Wheelwright, S., & Hyndman, R. (1997). *Forecasting: Methods and Applications.* New York: Wiley.
15. Martín, B. (2017). *Predicción semanal de precios de la energía eléctrica utilizando bosques aleatorios.* Madrid: Universidad Politécnica de Madrid.
16. Mena, J. (1999). *Data Mining your Website.* Boston: Digital Press.
17. Molnar, C. (2013). Recursive partitioning by conditional inference. *Seminar paper. Department of Statistics University of Munich* .
18. Montero-Manso, P., Netto, C., & Talagala, T. (2018). M4comp2018: Data from the M4-Competition. *R package version 0.1.0* , 0.
19. Vercher, E., Corberan-Vallet, A., Segura, J., & Bermúdez, J. (2012). Initial conditions estimation for improving forecast accuracy in exponential smoothing. *TOP* , *20* (2), 517-533.