

Doble Grado en Estadística y Economía

Título: *Automatic forecasting* y sus aplicaciones en Big Data: una comparativa entre algoritmos

Autor: Antoni Galmés Mifsud

Director: Salvador Torra Porras

Departamento: Econometría, Estadística y Economía Española

Convocatoria: Junio 2019



UNIVERSITAT DE
BARCELONA



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat de Matemàtiques i Estadística

RESUMEN

La predicción es uno de los campos más importantes tanto de la Economía como de la Estadística. El desarrollo de técnicas de predicción automática puede permitir dotarse de información para el futuro e intentar que éste sea lo menos incierto posible para poder anticiparse mejor. Los avances tecnológicos han permitido el desarrollo de un nuevo fenómeno como es el Big Data que permite el tratamiento de grandes volúmenes de datos. En consecuencia, nuevas técnicas como las de Minería de Datos han surgido como alternativas a modelos más clásicos en el campo del *automatic forecasting*. Así pues, en este trabajo se prevé, por una parte, dar una contextualización de la predicción automática y por otra, de carácter más empírico, llevar a cabo una comparación directa entre algoritmos aplicados a diferentes tipos de datos.

Palabras clave: Aprendizaje Automático, *ARIMA*, *Automatic Forecasting*, Big Data, *ETS*, Minería de Datos, *NNAR*, Prophet, Series Temporales.

Automatic forecasting and its applications in Big Data: a comparison between algorithms

ABSTRACT

Prediction is one of the most important fields of both economics and statistics. The development of automatic forecasting techniques can enable information to be available for the future and trying to make it as uncertain as possible in order to better anticipate. Technological Innovation has allowed the development of a new phenomenon such as Big Data that enable the processing of large volumes of data. However, this capacity of managing and storing such volume of data need to go with suitable methods, which allow to recognize different patterns and therefore to provide useful information. As a result, new techniques such as Data Mining have emerged as alternatives to more classic models in the field of automatic forecasting.

Thus, in this bachelor's thesis it is envisaged, on the one hand, to provide a contextualization of automatic forecasting, and on the other hand, more empirical in nature, to carry out a direct comparison between algorithms applied to different types of data.

In the empirical part, it will be introduced a new algorithm: Prophet. It will be compared with other automatic forecasting algorithms, which come from different statistical models. All of them, except from Prophet, belong to the *forecast* package of Hyndman. For carrying out the study, it will be use three different time series, which have different characteristics among them such as length, type of trend and seasonality, type of growth...

The main purpose of this bachelor's thesis is giving to the reader a general idea of what is automatic forecasting and discover the amount of applications it has and the amount of algorithms there are. It is important to point out that there are not a winner, but each one works better depending on each data.

Keywords: ARIMA, Automatic Forecasting, Big Data, Data Mining, ETS, Machine Learning, NNAR, Prophet, Time Series.

ÍNDICE

| | |
|--|-----|
| INTRODUCCIÓN..... | 7 |
| I <i>AUTOMATIC FORECASTING</i> Y SUS APLICACIONES EN BIG DATA..... | 10 |
| 1.1 Contexto actual | 10 |
| 1.2 Comparaciones..... | 17 |
| 1.3 Aplicaciones futuras..... | 19 |
| 1.4 Futuros desarrollos | 21 |
| II COMPARACIÓN ENTRE ALGORITMOS | 23 |
| 2.1 Fuente de datos | 23 |
| 2.1.1 Índice bursátil del mercado de activos italiano | 23 |
| 2.1.2 Número de pernoctaciones mensuales en las Islas Baleares. | 26 |
| 2.1.3 Número de bicicletas municipales en uso | 28 |
| 2.2 Indicadores de exactitud de la predicción..... | 30 |
| 2.3 Algoritmos..... | 31 |
| 2.3.1 Modelos ETS | 32 |
| 2.3.2 Modelos ARIMA..... | 45 |
| 2.3.3 Prophet | 56 |
| 2.3.4. Modelos NNAR..... | 73 |
| 2.4 Comparativa..... | 81 |
| CONCLUSIONES | 86 |
| BIBLIOGRAFÍA..... | 90 |
| ANEXO..... | 92 |
| Serie diaria | 92 |
| Serie mensual | 96 |
| Serie cinco-minutal..... | 101 |

GRÁFICOS

| | |
|---|----|
| Gráfico 1: Índice Bursátil del Mercado de Activos Italiano | 25 |
| Gráfico 2: Número mensual de pernoctaciones nacionales en Baleares | 27 |
| Gráfico 3: Número de bicicletas públicas usadas durante el mes de febrero | 29 |
| Gráfico 4: Residuos del modelo ETS aplicado a la Serie diaria | 35 |
| Gráfico 5: Predicción del periodo extramuestral del modelo ETS Serie diaria | 36 |
| Gráfico 6: Errores de predicción ETS Serie diaria | 37 |
| Gráfico 7: Comparación valores reales con predicción ETS Serie diaria | 38 |
| Gráfico 8: Residuos del modelo ETS Serie mensual..... | 39 |
| Gráfico 9: Predicción periodo extramuestral ETS Serie mensual | 40 |
| Gráfico 10: Errores de predicción modelo ETS Serie mensual | 41 |
| Gráfico 11: Comparación valores reales con predicción modelo ETS Serie mensual..... | 42 |
| Gráfico 12: Predicción periodo extramuestral modelo ETS Serie cinco-minutal | 43 |
| Gráfico 13: Comparación valores reales y predicción modelo ETS Serie cinco-minutal..... | 44 |
| Gráfico 14: Características residuos modelo ARIMA Serie diaria..... | 47 |
| Gráfico 15: Predicción periodo extramuestral ARIMA serie diaria | 48 |
| Gráfico 16: Error de predicción del modelo ARIMA Serie diaria | 49 |
| Gráfico 17: Comparación valores reales con la predicción ARIMA Serie diaria | 49 |
| Gráfico 18: Características residuos modelo ARIMA Serie mensual | 50 |
| Gráfico 19: Predicción periodo extramuestral ARIMA Serie mensual..... | 51 |
| Gráfico 20: Errores de predicción modelo ARIMA Serie mensual..... | 52 |
| Gráfico 21: Comparación valores reales y predicción ARIMA Serie mensual | 53 |
| Gráfico 22: Comportamiento residuos modelo ARIMA Serie cinco-minutal | 54 |
| Gráfico 23: Predicción periodo extramuestral ARIMA Serie cinco-minutal..... | 55 |
| Gráfico 24: Comparación valores reales y predicción ARIMA Serie cinco-minutal | 56 |
| Gráfico 25: Comportamiento tendencias Prophet Serie diaria | 60 |
| Gráfico 26: Errores de predicción modelo Prophet Serie diaria | 61 |
| Gráfico 27: Comparación valores reales con predichos Prophet Serie diaria..... | 62 |
| Gráfico 28: Comportamiento tendencia Prophet automático Serie mensual | 63 |
| Gráfico 29: Error de predicción Prophet automático Serie mensual | 64 |
| Gráfico 30: Comparación predicción Prophet automático Serie mensual | 65 |
| Gráfico 31: Comportamiento tendencias Prophet modificado Serie mensual | 66 |
| Gráfico 32: Error de predicción Prophet modificado Serie mensual..... | 67 |
| Gráfico 33: Comparación predicción Prophet modificado Serie mensual..... | 68 |
| Gráfico 34: Comparación predicción Prophet automático Serie cinco-minutal | 69 |
| Gráfico 35: Comportamiento tendencia Prophet modificado Serie cinco-minutal | 70 |
| Gráfico 36: Errores predicción modelo Prophet modificado Serie cinco-minutal | 71 |
| Gráfico 37:: Comparación predicción Prophet modificado Serie cinco-minutal..... | 72 |
| Gráfico 38: Predicción periodo extramuestral modelo NNAR serie diaria | 74 |
| Gráfico 39: Comparación predicción NNAR Serie diaria..... | 75 |
| Gráfico 40: Predicción periodo extramuestral NNAR Serie Mensual | 76 |

| | |
|---|----|
| Gráfico 41: Comparación predicción modelo NNAR Serie Mensual | 76 |
| Gráfico 42: Predicción periodo extramuestral NNAR sin frecuencia..... | 77 |
| Gráfico 43: Predicción periodo extramuestral NNAR frecuencia diaria | 78 |
| Gráfico 44: Predicción modelo NNAR Serie cinco-minutal frecuencia semanal | 79 |
| Gráfico 45: Errores de predicción modelo NNAR Serie cinco-minutal | 80 |
| Gráfico 46: Comparación predicción NNAR Serie Cinco-minutal | 81 |
| Gráfico 47: Comparación Algoritmos Serie Diaria..... | 83 |
| Gráfico 48: Comparación Algoritmos Serie Mensual..... | 84 |
| Gráfico 49: Comparación algoritmos Serie Cinco-minutal..... | 85 |

TABLAS

| | |
|--|----|
| Tabla 1: Modelos de suavizado exponencial..... | 32 |
| Tabla 2: Valores del parámetro s de cada mes del modelo ETS(M,N,M) | 39 |
| Tabla 3: Comparación EPAM Serie Diaria | 82 |
| Tabla 4: Comparación EPAM Serie Mensual | 82 |
| Tabla 5: Comparación EPAM Serie Cinco-minutal..... | 82 |
| Tabla 6: Comparación entre series..... | 83 |

INTRODUCCIÓN

La predicción es uno de los campos más importantes de la Economía y la Estadística. Además de tener un pasado muy interesante, actualmente está en auge gracias al fenómeno del Big Data y a la introducción de diferentes técnicas más innovadoras como las derivadas del Aprendizaje Automático. A la vez, van surgiendo numerosos algoritmos de predicción automática, muy interesantes de aplicar y de estudiar.

En mi caso particular, desde los últimos años he sentido curiosidad por las diferencias técnicas de modelaje, desde que cursé la asignatura de Modelos Lineales, Econometría y Análisis de Series Temporales, que equivalen a las tres asignaturas de Econometría del Grado de Economía. También, como estudiante de Economía, a lo largo de la carrera me he interesado por el estudio del comportamiento de diferentes indicadores, en especial los de carácter macroeconómico, así como su impacto en diferentes sectores económicos. De esta forma, la motivación principal que me ha llevado a elegir este tema para mi último trabajo de ambos grados, es la gran vinculación que tiene en las dos áreas (Economía y Estadística) y el gran papel protagonista que ocupa en el contexto actual, donde tenemos una infinidad de investigadores tratando de conseguir numerosas innovaciones sobre el campo del análisis de datos y de la predicción.

Si nos centramos, más concretamente en el cuerpo del trabajo, éste se divide en dos partes principales y diferenciadas: una parte más teórica y otra más de carácter más práctico.

Por una parte, he creído conveniente comparar, aunque solamente sea a través de la investigación teórica, técnicas ya vistas en la asignatura de Análisis de Series Temporales con técnicas que se estudian en la asignatura optativa de Minería de Datos. Esta comparación no se ha trasladado a la parte empírica, por su enorme extensión. Cabe señalar que durante la realización de ambas asignaturas se han llegado a estudiar un gran número de métodos de predicción. Por lo tanto, he priorizado estudiar pocos algoritmos, pero realizando un análisis individual y comparativo más exhaustivo.

Además de la comparación, en la parte teórica he intentado contextualizar el *automatic forecasting* o predicción automática en la actualidad, donde el auge y la ya consolidación del fenómeno *Big Data* abre un gran abanico de nuevas posibilidades tanto en el mundo de análisis de datos, como de los modelos de predicción.

Para concluir la primera parte, he incluido una previsión del futuro basada en los avances de la Inteligencia Artificial, gracias en parte al *Big Data* y sobre todo al *Machine Learning*, y como estos avances afectarán a la población y en consecuencia al sistema económico.

La segunda parte la he enfocado más en la comparación práctica de diferentes algoritmos de *automatic forecasting*. Los objetivos planteados en esta parte son básicamente determinar si hay diferencias significativas entre los diferentes algoritmos, qué tipo de diferencias hay, analizar si se comportan de forma diferente según el tipo de serie, comprobar si las ventajas/desventajas teóricas se cumplen en los casos planteados... Posteriormente, se irán nombrando y estructurando de forma más concreta, clara y ordenada los diferentes objetivos de cada parte y, a la vista de los resultados, se irá deliberando sobre las conclusiones extraídas en cada caso.

Para ello, se contará con cuatro algoritmos de predicción automática diferentes: *ETS*, *auto.arima*, *Prophet* y *NNAR*. En un primer momento, se acordó que *Prophet* sería uno de los algoritmos protagonistas de este trabajo. De hecho, el objeto del trabajo surgió a partir de la idea de investigar este nuevo algoritmo. Además, creí que era esencial poder compararlo con otros algoritmos dedicados a la misma finalidad: predecir de forma automática (o casi automática) diferentes series temporales.

Así pues, decidí investigar el paquete *forecast* de R de Hyndman. Después de numerosas pruebas, así como investigar que podía aportar cada algoritmo de este paquete desde el punto de vista teórico, decidí decantarme por las funciones *auto.arima* (Modelos ARIMA) y *ets* (Modelos ETS), las cuales comparten unos cuantos supuestos, y también *nnetar* (Modelos NNAR) aportando una alternativa no lineal dentro del mismo paquete de R.

Además, también se contará con tres series de diferente frecuencia (diaria, mensual y cinco-minutal) así como también de diferentes características y particularidades: distintos tipos de tendencia y estacionalidad, dimensiones diferentes... Como antes he mencionado, unos de los objetivos principales es comparar el comportamiento de cada algoritmo en series diferentes. Más adelante, se presentará cada una de las series de forma mucho más detallada. Allí también, se incluirá un análisis gráfico y técnico, explicación de las principales características, *pre-processing* de los datos, la motivación de haberla elegido, los objetivos planteados *a priori*, así como una valoración final de lo que ha aportado cada una.

El software que he utilizado para el experimento es únicamente R. Por una parte, es la herramienta estadística con la que más he trabajado a lo largo de la carrera y, por

lo tanto, la que más conozco. Además, *Prophet*¹ solamente está disponible actualmente para Python y R. Al tener solamente un nivel de usuario de Python era desaconsejable su uso para el trabajo. Asimismo, el paquete *forecast* está disponible para R.

Para concluir esta introducción, me gustaría expresar una serie de agradecimientos:

En primer lugar, a mi tutor, el Dr. Salvador Torra, por guiarme y compartir su conocimiento para llevar a cabo este trabajo tanto a nivel técnico como funcional.

En segundo lugar, agradecer también a mis compañeros tanto de Economía como de Estadística, a todos mis amigos que he ido conociendo a lo largo de esta etapa en Barcelona y a mis amigos de siempre de Manacor.

Por último, me gustaría expresar mi agradecimiento a mi familia por su apoyo y por confiar siempre en mi criterio.

¹<https://facebook.github.io/prophet/>

I AUTOMATIC FORECASTING Y SUS APLICACIONES EN BIG DATA

1.1 Contexto actual

La predicción siempre ha sido un campo esencial tanto en mundo de la economía como de la empresa. Paralelamente, el fenómeno del Big Data parece haber revolucionado el mundo del análisis de datos. Cada vez son más las empresas que tienen herramientas capaces de almacenar grandes cantidades de datos que van recogiendo ya sea de información propia, de su sector, o de otros factores externos. A la vez, el concepto Minería de Datos, así como el aprendizaje automático (*Machine Learning*, en inglés) parece haber irrumpido también con fuerza. En las últimas décadas, se han ido desarrollando diferentes algoritmos y métodos de análisis basándose en el entrenamiento de un modelo.

En el ámbito empresarial, muchas compañías necesitan dotarse del análisis de datos para diferentes finalidades: ya sea de planificación, estrategia o logística. A la vez, diferentes instituciones como Gobiernos nacionales, organizaciones mundiales o instituciones reguladoras, entre otros, necesitan también dotarse del análisis de datos para decidir, implementar, controlar o modificar las diferentes políticas planteadas.

Cuando se habla de análisis de datos, también se refiere a la posterior predicción hecha a partir de la construcción de un modelo basado en datos históricos o de entrenamiento. De esta manera, este nuevo panorama que se nos presenta y que viene capitaneado por el Big Data, tiene como principal reto el hecho de encontrar buenos algoritmos de predicción, así como buenos métodos de tratamiento de datos que nos permitan principalmente aprovechar toda esta cantidad de datos que se puede llegar a almacenar. Es decir, uno de los grandes desafíos de la predicción automática es encontrar métodos útiles para transformar todo tipo de datos en la máxima información útil posible.

El objetivo de esta primera parte del trabajo es profundizar un poco más en la cuestión planteada. Con lo cual, cabe hacer una breve introducción definiendo algunos conceptos como Big Data, Minería de Datos o *Automatic Forecasting* para poder enlazarlo de la mejor manera posible y entendible para así poder enfocarlo tanto en el mundo de las empresas como en el mundo de la Macroeconomía, ya que más adelante se pasará a analizar cómo ha afectado este cambio de paradigma a la hora de hacer predicciones de los indicadores principales macroeconómicos, sociales y financieros como puede ser el PIB, Política Fiscal y Monetaria, Tipo de Interés, Población...

Como se anticipado anteriormente, Big Data es uno de los fenómenos más virales del presente y es clave para el presente y futuro desarrollo de la predicción. Big Data es un término enfocado a la existencia de datos tan grandes y complejos de analizar que se requiere de herramientas informáticas de gran potencial, no tradicionales de procesamiento de datos para tratarlos adecuadamente y así poder extraer información útil para el posterior análisis y/o predicción. Cabe destacar la existencia de tres tipos de datos masivos: estructurados, semiestructurados, no estructurados. La diferencia entre ellos es la definición de la longitud y el formato. Como se puede suponer, los datos no estructurados son los que tienen un formato más difícil de analizar a priori, con lo que es necesario encontrar métodos que sean capaces de poder extraer información útil para su posterior análisis.

A la vez, es muy común el uso de las tres Vs ² para describir las características de estos datos: Variedad (el tipo y naturaleza de los datos para ayudar a las personas a analizar los datos y usar los resultados de forma eficaz), Volumen (la cantidad de datos generados y guardados) y Velocidad (la rapidez a la cual se generan y procesan los datos para cumplir las exigencias y desafíos de su análisis).

Con lo cual el objetivo que se plantea actualmente es claro: convertir datos en información que facilite la toma de decisiones. En otras palabras: intentar convertir datos no estructurados/ semiestructurados en estructurados.

De esta forma, cabe destacar el gran avance en el sentido de la capacidad de generar y tratar con cada vez más cantidad de datos. Por ejemplo, los datos digitales en 2006 eran de 16 exabytes y en 2012 ya eran de 2837 exabytes. Ahora en 2020, se prevé que sean 40 trillones de gigabytes.

Sin embargo, el objetivo planteado anteriormente, convertir tal cantidad de información en información útil genera uno de los grandes desafíos de la actualidad y da mucho juego para futuras investigaciones. Como (West, 2013)³, son muchos los expertos que afirman que hay una falta de teoría para complementar el Big Data y, parece que los métodos o algoritmos más clásicos de predicción (*automatic forecasting*) no están a la altura de los avances de las nuevas tecnologías.

De esta manera, en el presente trabajo, más allá de intentar hacer una breve síntesis del momento que vive el *automatic forecasting*, muy influenciado por el Big Data, se quiere tratar de comparar diferentes enfoques para dotar a este nuevo

² 3Vs (Volume, variety and velocity): <https://whatis.techtarget.com/definition/3Vs>

³ Big data needs a big theory to go with it. <http://www.scientificamerican.com/article/big-data-needs-big-theory/>

fenómeno. Cuando se menciona el concepto de enfoques, se refiere a un conjunto o conjuntos de diferentes métodos que estén a la altura de la gran cantidad de datos y gran complejidad que puedan generarse para poder obtener unas buenas predicciones.

Así pues, volviendo al problema inicial, no parece que se tenga un marco de referencia conceptual claro para encarar cuestiones de complejidad. De esta forma, el Big Data pierde mucha de su potencial utilidad sin unas herramientas a su altura que le acompañen.

Un marco de referencia global, tanto matemático como predictivo, para sistemas complejos, incorporaría, en principio, las dinámicas y organización de cualquier sistema complejo en un marco de referencia cuantitativo y computable donde aprovechar los datos. Igualmente, difícilmente se llegue a hacer predicciones muy detalladas (como además se podrá ver en la parte práctica del presente trabajo) pero sí se debería poder predecir en los próximos años una probabilidad de que ocurra un determinado evento (clasificación) o cantidad (regresión).

Enfocado en el mundo empresarial, es de vital importancia que las diferentes corporaciones desarrollen las herramientas y medios necesarios para reaccionar y explotar el aumento de la disponibilidad de los datos gracias al avance ya consolidado del Big Data. Entonces, aquí es donde aparece el concepto de Minería de Datos (*Data Mining*, en inglés), ya que hay una gran necesidad de este tipo de técnicas.

De esta forma, es de vital interés antes de proseguir con la explicación, realizar una breve definición sobre en qué consiste la Minería de Datos y cuáles son sus métodos más conocidos.

La Minería de Datos es un campo que engloba por una parte la estadística y por otra, las ciencias de la computación. Consiste básicamente en la práctica de examinar una serie de bases de datos ya existentes con la intención de realizar un proceso de explotación de grandes volúmenes de datos. Su finalidad es la de descubrir patrones de comportamiento en los datos que puedan aportar información valiosa.

Normalmente, estos patrones descritos anteriormente no pueden ser detectados mediante la exploración tradicional de los datos porque las relaciones son demasiado complejas o porque simplemente hay demasiados datos. Ahí es donde este nuevo fenómeno se puede diferenciar del resto de métodos de predicción más clásicos.

De todas formas, en un principio estas técnicas fueron diseñadas para manejar datos de una magnitud mucho más reducida si se compara con el tamaño del Big Data.

Con lo cual, cabe señalar un concepto que ha ido cogiendo mucha fuerza, ya que se presenta como el nexo entre el Big Data y las técnicas de Data Mining: *Data Quality Mining*⁴. El *Quality Mining* se define como un método para la evaluación de la Calidad de los Datos inspirado en las técnicas de Minería de Datos.

Cabe destacar que la principal diferencia entre *Data Mining* y *Quality Mining* es el objetivo a alcanzar. Por una parte, la finalidad de la aplicación de técnicas de Minería de Datos es encontrar nueva información sobre los datos, ya sea patrones de comportamiento, comparaciones entre variables, predicciones en un horizonte temporal marcado, en otros. En cambio, el *Quality Mining* se centra en la evaluación de la calidad de los datos usando patrones confiables (estudiados y probados anteriormente), pero no necesariamente aportan conocimiento adicional a los datos.

Con la introducción a este nuevo concepto, se quiere hacer llegar la enorme importancia que tiene el hecho de llevar a cabo una evaluación exhaustiva a los datos previo a la predicción en sí, para evaluar su calidad, así como corregir diferentes errores que puedan tener los datos tales como valores *missing*, duplicidades, incoherencias entre atributos, inconsistencias... Todos los ejemplos que se acaban de exponer forman parte de las llamadas dimensiones de la calidad de los datos⁵.

La calidad de los datos se presenta como una fase previa al *forecasting*, pero no por eso menos importante. De esta manera, las técnicas de Data Mining no solo pueden ser útiles como algoritmos de *automatic forecasting*, sino que también en el paso previo que demanda actualmente el gran auge del fenómeno Big Data de transformar los datos almacenados en información útil para poder llevar a cabo una correcta predicción.

Gracias a la correcta aplicación de estas herramientas, el *Data Mining* se puede presentar como un candidato potencial a complementar al fenómeno Big Data a la hora de tratar grandes dimensiones de datos y sus complejas relaciones entre ellos.

Sin embargo, no hay que olvidar las técnicas más clásicas de *automatic forecasting*, ya que actualmente se siguen usando y se siguen aplicando de manera muy activa tanto en la docencia como en el ámbito empresarial. No por ser más antiguas se tengan que dejar de lado.

Uno de los más conocidos son los modelos Autorregresivos Integrados de Media Móvil, más comúnmente conocidos por la abreviatura ARIMA. El uso de ARIMA para la

⁴ Para más información consultar: Soler, S. V. (2001). *Quality Mining: A Data Mining Based Method for Data Quality Evaluation*.

⁵ Sidi, F. a. (08 de 2013). *Data quality: A survey of data quality dimensions*

predicción automática de eventos futuros es ampliamente considerado como una de las técnicas de predicción más eficientes en ciencias sociales y es usado extensivamente para series temporales. Además, algoritmos de *automatic forecasting* como el desarrollado de *Hyndman* permiten seleccionar el modelo que más se adapta a los datos históricos⁶.

Las ventajas que ofrecen los modelos ARIMA son que pueden ser fácilmente interpretables, ya que la relación entre la variable dependiente e independiente es fácilmente entendible, ya que está basada en los supuestos del modelo. No obstante, uno de los principales problemas es que alguno de sus supuestos es difícilmente aplicable a la realidad: suponen relaciones lineales entre variables dependientes e independientes, además de suponer una desviación estándar constante de los errores del modelo. Este supuesto puede dejar de ser imprescindible cuando ARIMA es sustituido por los modelos ARCH.

Las siglas ARCH significan Heterocedasticidad Condicional Auto-Regresiva. Son unos modelos muy usados en el campo de las finanzas, sobre todo en la bolsa. A diferencia de los modelos ARIMA no es necesario que cumplan el supuesto de linealidad, ya que se presentan como unos modelos de predicción no lineales. Funcionan especialmente mejor respecto a los ARIMA en series temporales de altas volatilidades, es decir, de varianzas no constantes a lo largo de la serie. Igualmente, los modelos ARCH también tienen sus problemáticas. De todas, formas como se verá más adelante también puede ser útil para construir modelos híbridos, es decir, una combinación entre este tipo de modelos y otros tales como técnicas de Minería de Datos u otras ramas del *automatic forecasting*.

Como antes se ha anunciado, los modelos ARIMA pueden formar parte de las diferentes técnicas del *automatic forecasting* gracias al paquete desarrollado por Hyndman, el paquete *forecast*. Esto no quiere decir que este señor sea el único contribuyente a poder usar este tipo de modelos. En cualquier programa (ya sea Python, R...) uno dispone de diferentes paquetes libres y accesibles a todo el mundo para trabajar con una gran variedad de métodos de predicción. Sin embargo, cabe destacar no solo su facilidad y automatización en el uso de este tipo de modelos, sino las diferentes funciones que aporta en esta librería que permiten entre otras cosas, analizar el comportamiento de las series temporales, así como aplicar otras características tales como transformaciones o validaciones de parámetros⁷.

⁶ <https://robjhyndman.com/publications/automatic-forecasting/>

⁷ Hyndman, R. J. (2007). *Automatic time series for forecasting: the forecast package for R*. (D. o. Monash University, Ed.)

Igualmente, este paquete no solamente contempla algoritmos de predicción automática tomando como base de predicción los modelos ARIMA, sino que ofrece un amplio abanico de diferentes algoritmos basados métodos de predicción más clásicos tales como ETS, ARFIMA, TBATS, NNAR...

Cada uno de estos algoritmos es más idóneo si se cumplen una serie de supuestos. Por ejemplo, los modelos ARFIMA vendrían a ser los modelos ARIMA fraccionalmente integrados. De esta forma, este tipo de modelos son más usados si la serie temporal es de memoria larga. Puede ofrecer una alternativa más flexible a la hora de investigar la dinámica de una serie. Además, permite una modelización flexible del comportamiento en baja frecuencia, con implicaciones importantes en la medición de la persistencia del shock.

Otro ejemplo, que de hecho se experimentará con él en la parte práctica son los modelos de espacio de estado de innovaciones, es decir, los modelos ETS (*Error, Trend, Seasonality*). Éstos mantienen la naturaleza intuitiva del suavizado exponencial, pero con el enfoque de modelos de espacio de estado, con las respectivas ecuaciones de medida y estado. Una de las ventajas de estos modelos es la fácil interpretación de cada uno de sus componentes. Sin embargo, también resulta tener sus limitaciones como se verá más adelante.

Asimismo, uno de los modelos más completos es el TBATS. Éstas siglas se pueden descomponer de la siguiente manera: **T** (términos Trigonométricos para la estacionalidad), **B** (transformación Box-Cox para la heterogeneidad), **A** (errores ARMA para dinámicas a corto plazo), **T** (Tendencia), **S** (Estacionalidad (*Seasonality*), incluyendo periodos múltiples y no-enteros). Una de las ventajas que tiene es que maneja estacionalidades no enteras y periodos de múltiples estacionalidades. Sin embargo, también presenta ciertas desventajas, así como limitaciones. Por ejemplo, los intervalos de predicción, a menudo son muy amplios y además el algoritmo de R es muy lento para series temporales de muchas observaciones.

Volviendo al supuesto de linealidad, como antes se ha mencionado en la parte empírica del presente trabajo no se ha decidido experimentar con los modelos ARCH porque son modelos no lineales con volatilidad cambiante. Como alternativa al supuesto de linealidad, el paquete *forecast* del profesor Hyndman propone un algoritmo que selecciona el mejor modelo NNAR⁸ para la serie introducida. NNAR responde a las siglas de Red Neuronal (*Neuronal Net*) Autorregresiva. De esta forma, este modelo se presenta

⁸ Para más información: <https://otexts.com/fpp2/nnetar.html>

como una versión no lineal de un modelo autorregresivo. En el segundo capítulo, se va a tratar de forma empírica con este algoritmo. Con lo cual, también se va a dedicar un apartado a profundizar en la teoría que envuelve este tipo de modelos.

Como se ha ido anunciando en las páginas anteriores de este trabajo, la parte práctica se centra en el estudio de tres algoritmos del paquete *forecasting* (*ets*, *auto.arima* y *nnetar*). Sin embargo, una de las principales metas propuestas con este trabajo es compararlo con un algoritmo también considerado de predicción automática, pero perteneciente a otra familia. De esta manera, se introduce el algoritmo *Prophet*⁹.

Como a alternativa a la familia del paquete *forecasting*, Facebook lanzó en febrero de 2017 un algoritmo de código abierto centrado en la predicción automática de series temporales. Perteneciente a la familia de algoritmos GAM (Modelos Aditivos Generalizados)¹⁰. En un principio, Prophet fue desarrollado para cuestiones y finalidades exclusivas de Facebook, tales como predecir la actividad del usuario en diferentes partes de la web o aplicación o para priorizar su desarrollo. En su presentación, se expusieron las principales innovaciones de este algoritmo.

Este modelo relativamente joven, viene a hacer frente a los problemas provenientes de los modelos de series temporales más tradicionales. Éstos básicamente son:

1. El intervalo de tiempo entre los datos tiene que ser el mismo a lo largo de la serie temporal.
2. Observaciones con NA's no están permitidas.
3. Estacionalidades con múltiples periodos (por ejemplo, semanal y anual a la vez), son difíciles de manejar.
4. La configuración de los parámetros a través de los expertos es necesario.

De esta forma, Prophet se presenta como un algoritmo robusto a los dos primeros problemas (intervalos de tiempo no constantes y observaciones con *missings*). A la vez, permite incluir más de un tipo de estacionalidad, además de poder personalizarla. Así también, está diseñado para ser fácil de usar sin un conocimiento experto en la predicción de series temporales o estadística.

⁹ Enlace: <https://peerj.com/preprints/3190/>

¹⁰ El modelo aditivo generalizado es extensión de los modelos tradicionales de regresión lineal, que incorpora la no linealidad y la regresión no paramétrica.

Así pues, en este caso, se podrá llevar a cabo una serie de experimentos para intentar comprobar si en esta experiencia se cumplen las promesas de este nuevo algoritmo respecto a los modelos de series temporales más clásicos.

1.2 Comparaciones

Al haber presentado hasta el momento dos diferentes enfoques para llevar a cabo predicciones, se cree interesante aportar unos cuantos ejemplos basados en diferentes experimentos en múltiples campos (desde la biología hasta el mercado financiero), donde se comparan por una parte técnicas propias de la Minería de Datos y por otra, modelos de predicción más clásicos:

Para predecir el precio del Stock, en (Adebiyi, 2014) se comparan las predicciones hecha a través de un modelo ARIMA y otra a través de una red neuronal artificial. Los resultados revelan la superioridad de las redes neuronales respecto a ARIMA, basándose en los diferentes indicadores del error. De hecho, en este *paper* se apunta a que, aunque los modelos ARIMA podrían funcionar mejor ante datos de dimensión más reducida y sobretodo de relación no tan compleja, este tipo de modelos no tienen en cuenta los shocks empíricos que las Redes Neuronales sí pueden recoger a base de entrenamiento del algoritmo.

Siguiendo con la comparación de modelos ARIMA, ésta vez se quiere predecir el flujo diario de un río en el norte de Irán. En este caso, en (Moharrampour, 2013) se compara la predicción a través de un modelo ARIMA con una predicción hecha a partir de la máquina de vector soporte (SVM), utilizando cuatro *kernels*¹¹ diferentes. Haciendo uso de la raíz del error cuadrático medio como indicador para decidir qué modelo predice mejor, se concluye, después de llevar a cabo numerosas pruebas, que funciona mejor con el método SVM.

En otros campos como es la biología, también se están llevando a cabo diferentes experimentos para poder determinar qué tipo de algoritmos de predicción automática pueden funcionar mejor. Un ejemplo lo encontramos en (Kane, 2014), donde hacen una comparación de algoritmos para el modelaje de una serie temporal. El objetivo es predecir la influencia de la gripe aviar en la muerte de un tipo de pájaro determinado. La comparación se lleva a cabo entre, por una parte, un modelo ARIMA (utilizando la función *auto.arima()* del paquete *forecast* de R) y por otra, se utiliza *Random Forest*

¹¹ En SVM la función *kernel* permite obtener una solución lineal en el espacio de características. Hay de varios tipos: Lineal, Polinomial, Sigmoid, Gaussiano...

(utilizado el paquete *randomForest*¹² de R). Para llevar a cabo la comparación, se usa el Error Cuadrático Medio. En todos los experimentos realizados, se muestra como los modelos *Random Forest* predicen con mayor exactitud.

En cuanto a Prophet, se encuentra un ejemplo de comparación con ARIMA a través de la función `auto.arima()` de R donde se tiene como objetivo predecir el valor de bitcoin. En (Yenidougan, 2018) utilizan el coeficiente de determinación R^2 como indicador cuantitativo de validación de la predicción, es decir, el porcentaje de variabilidad de los datos que está explicada por el modelo. Si se atiende a las comparaciones, Prophet tiene un R^2 de 0.94 (realmente muy bueno), mientras que el modelo ARIMA resultante tiene un coeficiente de determinación del 0.68. Por lo tanto, sí que existen diferencias bastante evidentes de precisión en la predicción de un modelo u otro.

Respecto a la alternativa no lineal de Hyndman a través de los modelos NNAR, en (Maleki, 2018) se hace una comparación entre estos modelos y los ARIMA. El objetivo es predecir los principales indicadores que afectan al tratamiento de una planta hidráulica. La principal conclusión es que ninguno de los dos modelos predice con más exactitud que los otros. ARIMA funciona mejor si se usa la función TSCV¹³. Sin embargo, si se usa el coeficiente R^2 como indicador, en la mayoría de series se obtiene un mejor coeficiente a través de los modelos NNAR.

Por otra parte, no solo se quiere dar ejemplos de comparaciones como si esto fuera una competición. A la vez, se cree relevante remarcar que una combinación de ambos tipos, implementándose a través de un modelo híbrido no solo puede ser posible, sino que muy útil. Un ejemplo lo encontramos en (Lu, 2016), donde diseñan un modelo híbrido de Redes Neuronales Artificiales y de tipo GARCH para predecir la volatilidad en el mercado energético chino.

Cabe destacar, que incluso instituciones oficiales importantes están experimentando con este tipo de comparaciones y se están pensando seriamente el uso del *Machine Learning*. El ejemplo más destacado se encuentra en un trabajo reciente del Fondo Monetario Internacional. En (Jung, 2018) se aplican tres algoritmos diferentes: *Elastic Nets*, *SuperLearner* y Redes Neuronales Recurrentes. Se aplican a diferentes variables macroeconómicas y se toman como muestra una gran número de países. Se ha podido desmostrar que los algoritmos de aprendizaje automático utilizados

¹² <https://www.rdocumentation.org/packages/randomForest/versions/4.6-14/topics/randomForest>

¹³ TSCV: *Time Series Cross Validation*. (del paquete *forecast*)

en el proyecto predijeron la evolución de la economía española, entre otras, con un 82% más de exactitud que los economistas del FMI.

En líneas generales, las ventajas que ofrecen los modelos ARIMA son que pueden ser fácilmente interpretables, ya que la relación entre la variable dependiente e independiente es fácilmente entendible, al estar basada en los supuestos del modelo. No obstante, uno de los principales problemas es que alguno de sus supuestos es difícilmente aplicable a la realidad: suponen relaciones lineales entre variables dependientes e independientes, además de suponer una desviación estándar constante de los errores del modelo. Este supuesto puede dejar de ser imprescindible cuando ARIMA es usado en conjunción con GARCH. Igualmente, los modelos GARCH también tienen sus problemáticas, siendo uno de los motivos por lo que se ha preferido optar por Prophet, basados en los Modelos Aditivos Generalizados, como alternativa a la limitación que supone el supuesto de cumplimiento de la linealidad.

Como se ha podido observar, el *automatic forecasting* es uno de los campos más importantes y que ha recibido mayor atención tanto en el sector de la estadística, como de la economía y la empresa.

Además, en el campo de la economía es ampliamente aplicado, sobre todo a la hora de predecir diferentes indicadores cuantitativos de la macroeconomía. La predicción, en este caso, principalmente se aplica a indicadores como el PIB, tasa de desempleo, tipo de interés como coste del capital, inflación... El principal objetivo reside en el hecho de actuar en consecuencia con el resultado de las predicciones llevando a cabo diferentes políticas (ya sean fiscales, monetarias o mixtas).

Por lo tanto, una correcta predicción puede contribuir a una mejor provisión, no solo de los gobiernos o instituciones para encarar los diferentes desafíos de cada nación, sino también de las empresas para tener menos incertidumbre sobre la evolución de sus sectores clave.

1.3 Aplicaciones futuras

Como antes se ha mencionado, el aprendizaje automático no solamente predice una serie temporal en el periodo extramuestral establecido, sino que sus barreras parecen ilimitadas en el futuro.

La inteligencia artificial, como tecnología de vanguardia que engloba el *Machine Learning* como uno de sus subconjuntos, es considerada como una serie de procesos en

los que una máquina imita las funciones cognitivas que los humanos asocian con otras mentes humanas. De esta forma, viene a sustituir el ser humano a la hora de aprender y resolver problemas. Entre otras cosas como antes se ha mencionado, parece que en un futuro el próximo Ministro de Economía será un algoritmo, ya que podrá detectar cualquier crisis o shock mucho antes y con mucha mayor exactitud que cualquier economista o banquero central.

Además, (Grace, 2018) prevé, entre otras muchas cosas, que:

- En 2024, la inteligencia artificial supere a la humana a la hora de traducir idiomas.
- En 2026, sea capaz de escribir comentarios de texto mejor que los alumnos de Bachillerato
- En 2027, se pueda confiar más en la Inteligencia Artificial para conducir camiones que en los seres humanos.
- En 2031, la Inteligencia Artificial sea habitual en el comercio, y supere a los vendedores humanos.
- En 2049, escriba *best sellers*.
- En 2053, pueda hacer cualquier tipo de cirugía.
- En 2057, pueda llevar a cabo sus propias investigaciones científicas.

Con lo cual, parece que va a haber no solo una revolución en nuestro día a día, sino en la forma de conseguir ingresos por parte de las familias. Si se atiende al dato predicho en el año 2031, la mayoría de puestos de trabajo quedarán obsoletos. En consecuencia, serán eliminados y sustituidos por robots, máquinas o simplemente sistemas automatizados.

De esta forma, se plantea una pregunta muy importante: ¿Qué nos depara el futuro? Una cosa está clara y es que es imprescindible tanto formarse académicamente como adaptarse lo máximo posible a los nuevos tiempos. Y este mensaje no solo va dirigido a los más jóvenes, sino a todos aquellos a los que se están dedicando o están ocupando puestos de trabajo que en un futuro cercano estarán obsoletos y completamente absorbidos por la inteligencia artificial.

Esto dará lugar al denominado desempleo estructural, íntimamente relacionado con el desarrollo tecnológico. Dos movimientos contrarrestan la pérdida de empleo que provoca el desarrollo tecnológico: el continuo aumento del empleo en el sector de los servicios, y la posibilidad de reducir la jornada laboral al conseguir mayores niveles de productividad. Sin embargo, esta revolución tecnológica parece que va muchas dimensiones más allá de simples cambios.

De esta forma, los gobiernos de todas las naciones tendrán que adoptar medidas para poder poner soluciones a las altas tasa de desempleo tecnológico que parece que se van a dar a en un futuro de cada vez menos incierto y lejano.

1.4 Futuros desarrollos

Entre otros muchos objetivos de la aproximación teórica del *automatic forecasting* y su papel en el presente y en el futuro cercano, así como sus diferentes variantes, se ha querido dar a entender que este tema engloba mucho más de lo que puede abarcar un Trabajo de Final de Grado. Así pues, entre otras muchas pretensiones hay una serie de propuestas para futuros trabajos, tanto en el ámbito de la Estadística como en el ámbito de la Economía.

En el ámbito de la Estadística, se pretende que en un futuro se pueda llevar a cabo algún trabajo de investigación más relacionado directamente con la puesta en práctica de las diferentes técnicas de Minería de Datos, tanto de aprendizaje supervisado como no supervisado, implementando tanto algoritmo de regresión como de clasificación. Si no se ha incluido en la parte práctica del presente trabajo, es por varias razones:

1. El tema principal del trabajo es *automatic forecasting*, y aunque estas técnicas pueden actuar como una alternativa, no son puramente algoritmos de predicción automática
2. Si se quiere llevar a cabo de forma correcta y con rigor, hace falta una mayor extensión y mayor tiempo de trabajo que el asignado para realizar este trabajo.
3. Se requiere de mayor conocimiento, sobre todo del campo de la informática referente al Big Data, para poder trabajar con las

herramientas idóneas para el tratamiento correcto de grandes volúmenes de datos.

En el ámbito de Economía, se pretende que en un futuro se pueda realizar un trabajo de investigación, de ámbito teórico principalmente, sobre un nuevo modelo económico para hacer frente al reemplazamiento de la mano de obra humana por los robots, gracias a los avances de la predicción automática, el aprendizaje automático, el aprendizaje profundo y la Inteligencia artificial.

II COMPARACIÓN ENTRE ALGORITMOS

Una vez finalizado el capítulo referente a lo que sería la breve explicación sobre el estado actual del *automatic forecasting* en el entorno Big Data, así como sus aplicaciones y su comparación con las herramientas más punteras a la hora de llevar a cabo diferentes predicciones, se pasa a una presentación más exhaustiva de diferentes algoritmos de *automatic forecasting* y su puesta en práctica a través de diferentes series con características muy dispares.

Este apartado tiene como principal objetivo poder llevar a cabo una comparación lo más completa posible de los diferentes algoritmos. La comparación se realizará entre cuatro algoritmos diferentes, todos ellos implementados a partir del programa estadístico R. Por lo tanto, se usarán una serie de funciones y paquetes de R que se irán anunciando a lo largo de este capítulo. Igualmente, en el presente trabajo en el apartado de anexo se incluirán todos los scripts utilizados para desarrollar todo el análisis y comparación.

Para realizar este experimento, se trabajará con tres series diferentes. A continuación, se procede a anunciarlas y a explicar con detalle, no solo sus características principales, sino también las razones que han llevado a elegir cada serie y sus objetivos esperados al aplicar *automatic forecasting*.

2.1. Fuente de datos

2.1.1 Índice bursátil del mercado de activos italiano

La primera serie trata sobre el indicador de referencia global de la Bolsa de Valores Italiana. Los datos provienen de la web de *Quandl*¹⁴.

Quandl es un sitio web muy recomendable si se quiere trabajar con diferentes series temporales de carácter económico/financiero. Solamente se tiene que crear una cuenta para tener acceso a la plataforma. Esta plataforma contiene datos, la mayoría son series temporales, tanto de carácter gratuito como Premium. Aunque para poder trabajar con las mejores series es aconsejable hacerse una cuenta Premium, sí que hay

¹⁴ Fuente de los datos: <https://www.quandl.com/data/BITAL/MFTSEMH2019-FTSE-Mini-Index-Future-March-2019>

un amplio abanico de posibilidades también con series de carácter gratuito, como esta en cuestión.

La frecuencia de los datos es diaria. Sin embargo, solo hay datos en los días laborables como es de suponer, ya que es un indicador de bolsa. Los datos de esta serie representan el valor de cierre de cada jornada.

Hay un total de 123 observaciones. Cabe destacar que es una muestra relativamente pequeña, sobre todo si se atiende a que los datos son diarios. Por lo tanto, esta serie comprende entre 4 y 5 meses de información. Más concretamente, el primer dato proviene del 24 de setiembre de 2018, mientras que el último es del 15 de marzo.

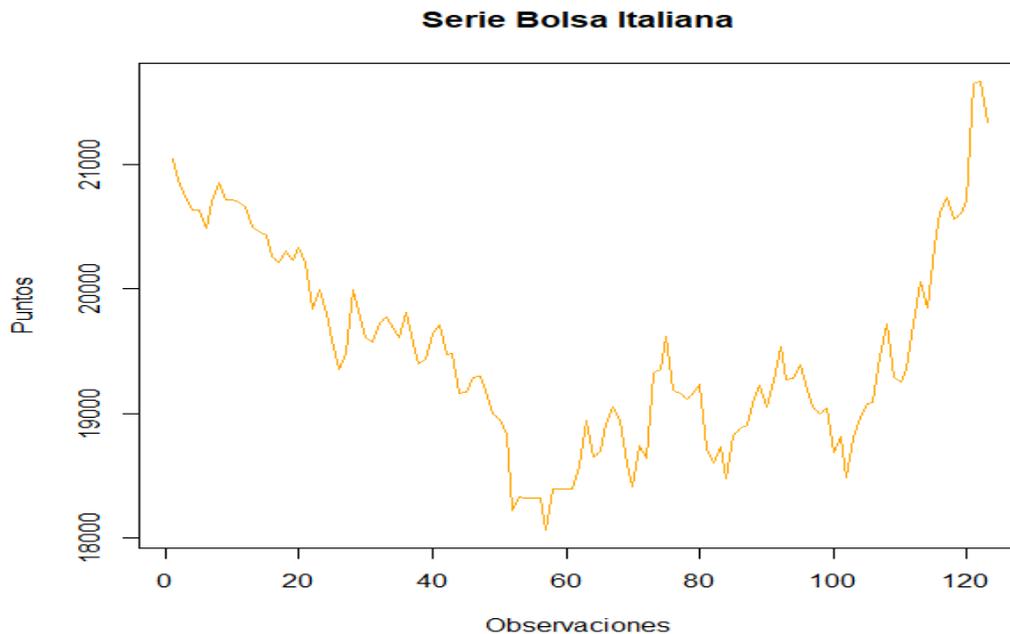
La razón al haber elegido como primer experimento una serie tan corta se debe a que se quería tener una pequeña aproximación para observar y estudiar con detenimiento como funciona cada algoritmo (en los posteriores apartados se analizará más detalladamente).

En cuanto a la división del análisis, las 86 primeras observaciones se dedican a la elaboración del modelo (elección del mejor modelo y estimación de sus parámetros) mientras que las 37 últimas serán destinadas a la validación del modelo en cuestión. Este criterio es aplicado a todos los experimentos. En términos porcentuales, es la forma más aproximada a 70% *train* y 30% *test*. La razón principal de dividir de esta forma se debe a que es un criterio muy usado en este sector.

Con lo que concierne al *pre-processing*, destacar que la base de datos original no contenía ningún valor *missing* ni incoherente. Sin embargo, ésta venía ordenada en sentido contrario, lo cual se ha tenido que arreglar, ya que no se habría podido extraer ninguna conclusión correcta. Además, no se ha retocado ninguna observación. Simplemente para poder implementar de manera correcta la función *prophet*, se ha llevado a cabo una serie de modificaciones que posteriormente se explicará, aunque ninguno de ellos afecta ni a la fecha ni al valor de las observaciones, ni mucho menos al valor de las predicciones.

A continuación, se procede a presentar gráficamente esta serie:

Gráfico 1: Índice Bursátil del Mercado de Activos Italiano



Fuente: Elaboración propia

Se puede observar como la serie va oscilando a lo largo del periodo. Sin embargo, en la primera mitad se da la presencia de una tendencia bajista (primeras 60 observaciones aproximadamente). Posteriormente, va oscilando (de la observación 60 a la 100, aproximadamente). Finalmente, se produce un incremento bastante notable en el indicador de la bolsa.

De aquí, cabe señalar que no hay una tendencia constante o permanente a lo largo de la serie, ya que en momentos sube, en otros oscila y en otros, baja. Se puede decir que solo se ve una subida clara durante el periodo extramuestral, lo cual quiere decir que los diferentes algoritmos no han sido dotados de tal información para elaborar el modelo. De esta forma, uno de los objetivos claros que se plantea en esta serie es ver si alguno de los algoritmos de automatic forecasting será capaz de poder predecir esta tendencia, o al menos, mirar cuál de ellos se ha aproximado mejor y a través de qué criterio.

Una vez presentada la primera serie y haber definido los objetivos, se presenta la siguiente.

2.1.1 Número de pernoctaciones mensuales en las Islas Baleares.

Como el título indica, esta serie recoge el número de turistas extranjeros que han pernoctado en algún hotel de las Islas Baleares. La fuente de los datos proviene del INE (Instituto Nacional de Estadística)¹⁵.

Cabe señalar que la web del INE, es otra fuente muy diversa e interesante de datos, sobre todo de series temporales. Además, está abierto a todo el mundo. De esta forma, se pueden recoger series de todo tipo de campos relacionados con las ciencias sociales y principales motores de la economía española.

La razón de haber elegido esta serie se debe a la enorme importancia que puede tener una buena predicción en el campo del turismo, ya que es un sector con mucha demanda y uno de los pilares económicos de las Islas Baleares y de España. Además, esta serie ya fue usada (de forma mucho más reducida) para realizar el trabajo de la asignatura del grado de Estadística de Análisis de Series Temporales. Con lo cual, ya se le aplicó tanto el análisis determinista como el estocástico. Sin embargo, la longitud era mucho menor, así como también el periodo extramuestral o de validación.

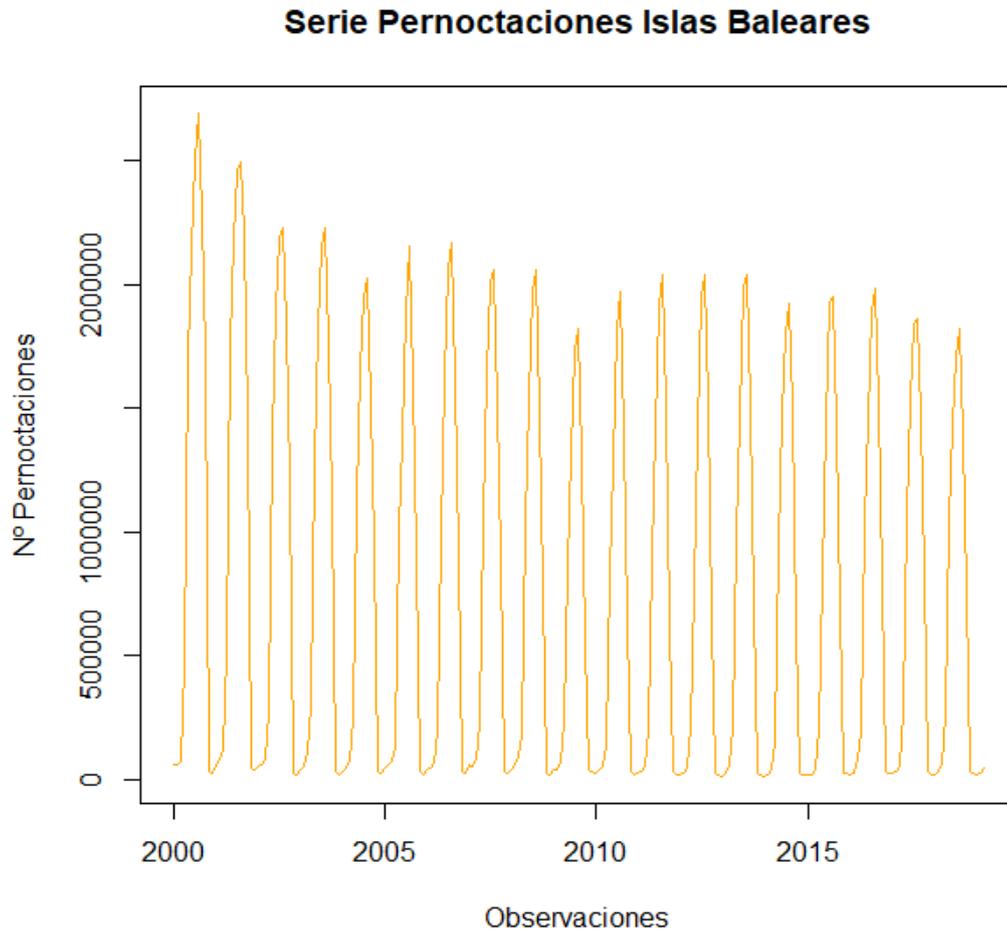
La frecuencia de los datos es mensual. De esta manera, se da un experimento con diferente frecuencia en comparación con en el anterior caso, pudiendo llevar a cabo una comparación más completa. La primera observación fecha del mes de enero del 2001, mientras que el último es de marzo del 2019. Esto quiere decir que hay un amplio intervalo de tiempo, que abarca desde el boom y la crisis económica hasta la recuperación y el turismo de masas fruto de las crisis puntuales de competidores directos como Egipto y Túnez, actualmente ya recuperados.

El total de observaciones asciende a 231. Las 161 primeras son destinadas a la elaboración y parametrización del modelo, mientras que las 70 restantes servirán para testear la predicción del modelo en cuestión. Por lo tanto, se utiliza un poco menos del 70 % para la parte de entrenamiento.

En cuanto al *pre-processing*, simplemente se han seleccionado únicamente las dos variables útiles para el análisis: fecha y valor. Cabe señalar que se ha cambiado el formato de fecha, para que encajara en el formato que se requiere en la función *prophet*. Igualmente, para el análisis mediante las diferentes funciones del paquete *forecast* se ha importado de tal manera que se da por supuesto que hay presencia de estacionalidad de periodo 12.

¹⁵ Enlace: <https://www.ine.es/dyngs/INEbase/listaoperaciones.htm>

Gráfico 2: Número mensual de pernoctaciones nacionales en Baleares



Fuente: Elaboración propia

A la vista de la gráfica 2 lo que más llama la atención es la más que evidente presencia de estacionalidad. Obviamente, el turismo es uno de los sectores que más sufre la estacionalidad. Por una parte, hay una presencia muy numerosa de turistas en temporada alta (principalmente en julio y agosto) y no tan alta en el resto del año.

Con lo cual, uno de los objetivos que se plantea en la implementación de los diferentes algoritmos de *automatic forecasting* es el comportamiento que tiene cada uno de ellos en una serie claramente estacional.

Aunque en los primeros años se ve que hay una tendencia negativa, después parece que va oscilando, sin tener muy claro si esta componente es realmente significativa. De esta forma, otro objetivo a plantearse en esta serie es muy similar a la anterior serie: como se parametrizará esta tendencia.

De todas formas, en la primera serie se encontraban cambios de tendencia muy bruscos mientras que aquí no parece que sean tan pronunciados.

Finalmente, se presenta la última serie.

2.1.2 Número de bicicletas municipales en uso

El *Bicing*¹⁶ es la red de bicis públicas de Barcelona. Esta plataforma es del Ayuntamiento de Barcelona, cuya adjudicación se llevó a cabo a través de concurso a una empresa privada.

Es un tema que tiene múltiples trabajos muy interesantes tales como las estaciones más concurridas, si la altura respecto al nivel del mar afecta (parece evidente que sí) y de qué manera, donde se produce más vandalismo en cuanto al material (estaciones, bicis...). Sin embargo, esta serie trata simplemente del número de viajes que se llevan a cabo mediante este medio de transporte, ya que las variables es el número de bicis en uso en cada momento determinado de tiempo.

La fuente de los datos proviene de la web de open data que proporciona el Ayuntamiento de Barcelona¹⁷. Es una página web muy interesante que contienen diferentes *datasets* sobre múltiples actividades que se llevan a cabo en la ciudad.

Las razones de haber elegido esta serie son varias. Por una parte, al ser usuario habitual de este servicio siempre he tenido curiosidad para estudiar diferentes características de este servicio. Por otra parte, la razón estadística se debe a que es una serie con muchas observaciones, con lo que se quiere poner a prueba la capacidad de los diferentes algoritmos de *automatic forecasting*.

La frecuencia de los datos es cinco-minutal, concretamente los datos están separados con intervalos constantes de 5 minutos. De esta forma, se quiere ampliar la comparación de los algoritmos, ya que es una frecuencia muy poco común, además de muy compleja de predecir.

Además, como más adelante se explicará con más detalle, una de las opciones que da el algoritmo Prophet es la de trabajar con datos subdiarios.

Los datos pertenecen a febrero de 2019, es decir, el primer dato es del 1 de febrero a la 00:00 horas mientras que el último es el 28 de febrero a las 23:55 horas.

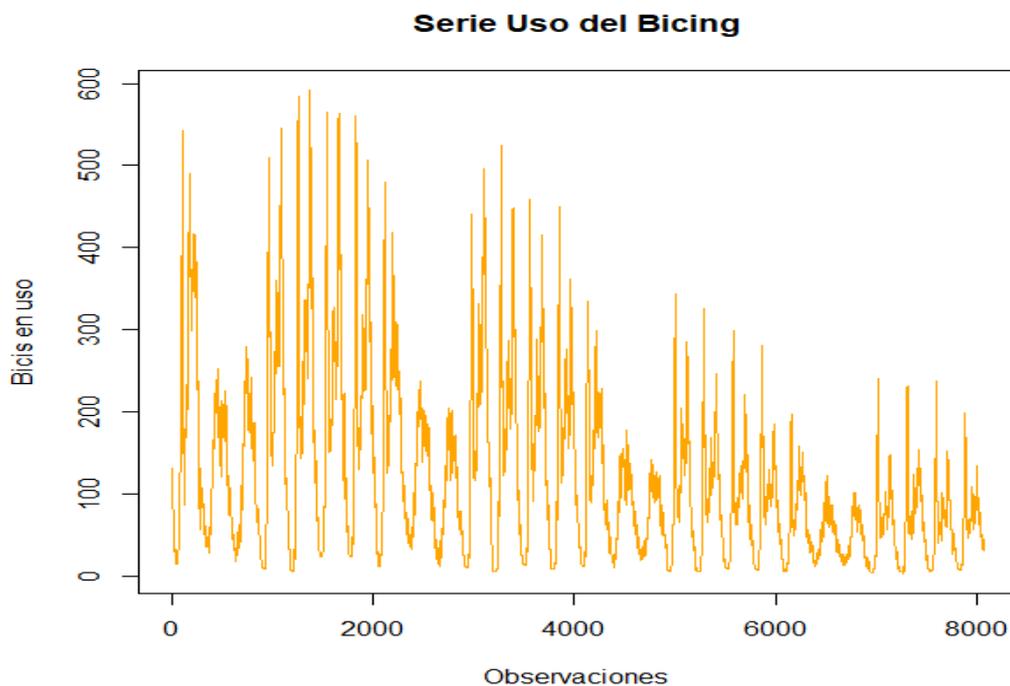
¹⁶ Enlace: <https://bicing.barcelona/>

¹⁷ Enlace: <https://opendata-ajuntament.barcelona.cat/es/>

El principal motivo de elegir el mes de febrero ha sido el *pre-processing*. Era el mes donde había menos datos *missing*. Es de vital importancia, dedicar el tiempo necesario para comprobar que no hay *missings*, ya que podría afectar al orden temporal de los datos. El único dato faltante es del 5 de febrero a las 20:35 horas. En el caso de Prophet, se ha añadido la fecha y un NA como valor en la posición correspondiente, desplazando las observaciones posteriores una posición. De esta manera, también se quiere evaluar cómo responde este algoritmo a la presencia de datos *missing*. En los demás algoritmos, se ha optado por poner el valor del día anterior a esa misma hora.

El número total de observaciones es de 8064 (28 días x 24 horas x 12 observaciones/hora). Como se puede observar, hay muchas más observaciones que en los dos anteriores casos, cosa que dificulta en gran medida la predicción. Sin embargo, se quiere probar el impacto que puede tener. Del total de observaciones, las 5760 primeras son destinadas a la parte de elaboración del modelo mientras que las restantes serán dedicadas a su posterior validación.

Gráfico 3: Número de bicicletas públicas usadas durante el mes de febrero



Fuente: Elaboración propia

Como se puede observar, es un gran reto poder predecir este tipo de serie. Parece haber una tendencia negativa, además de una presencia importante de estacionalidad, tanto diaria como semanal.

Por lo tanto, en esta serie se plantea el objetivo de intentar predecir con una mínima exactitud las diferentes componentes presentes. Al tratarse de tal dimensión tan grande de datos, cabe estudiar con detenimiento si los algoritmos de Hyndman serán capaces de poder estudiar la estacionalidad, así como también Prophet.

De esta forma, se da por concluida la primera fase de presentación de las tres series con las que se trabajará.

2.2. Indicadores de exactitud de la predicción

Para llevar a cabo la comparación entre los diferentes algoritmos de *automatic forecasting* en cada serie, es necesario hacer uso, entre otras cosas, de algún indicador cuantitativo. En este caso, se utilizará tres tipos de indicadores de error de predicción. Éstos son:

Error de predicción: Se define como la diferencia entre los valores predichos y los valores reales en horizonte temporal h .

$$e_{t+h,t} = Y_{t+h} - \hat{Y}_{t+h,t} \quad (1)$$

Error Cuadrático Medio (ECM): Se define como la media de los errores al cuadrado de cada observación en el horizonte temporal h :

$$ECM = \frac{1}{T} \sum_{t=1}^T e_{t+h,t}^2 \quad (2)$$

Error Absoluto Medio (EAM): Se define como la media de los errores en valor absoluto de cada observación en el horizonte temporal h :

$$EAM = \frac{1}{T} \sum_{t=1}^T |e_{t+h,t}| \quad (3)$$

Error Porcentual Absoluto Medio (EPAM): Se define como la media de los errores en valor absoluto en términos porcentuales respecto al valor de cada observación:

$$EPAM = \frac{100}{T} \sum_{t=1}^T \frac{|e_{t+h,t}|}{|Y_{t+h}|} \quad (4)$$

De esta forma, el EPAM será el indicador el cual se basará la valoración de las diferentes predicciones, ya que es un indicador adimensional.

Además, también se evaluará el comportamiento de los residuos del modelo. Concretamente, se utilizará el Test de *Shapiro-Wilk*. Este test se usa para contrastar la normalidad de un conjunto de datos. Se plantea como hipótesis nula que los datos siguen una distribución normal. Como hipótesis alternativa, se plantea que estos datos no siguen una distribución Normal. En cuanto a su implementación en R, se lleva a cabo a través de la función *shapiro.test()*.

Cabe añadir que, en la última serie, de frecuencia cinco-minutal, se ha utilizado el Test de *Kolmogorov-Smirnov*, muy similar y de misma utilidad que el de *Shapiro-Wilk*. La razón se debe a que en R la función *shapiro.test()* tiene un límite de 5000 observaciones en el *input*. La función de R que aplica el test es *ks.test()*. Ambas funciones son del paquete *stats*.

Siempre es muy importante evaluar la normalidad de los residuos, ya que un incumplimiento de este supuesto puede afectar severamente a la eficiencia de la estimación. A la vez, el valor de los intervalos de confianza de los parámetros del modelo puede ser solo aproximado y no exacto.

Ahora, se da paso a la presentación tanto teórica como práctica (implementación en R) de cada algoritmo de *automatic forecasting* presente en este trabajo, así como el comportamiento en el modelaje de cada una de las series.

2.3. Algoritmos

En este apartado se pueden diferenciar por una parte los algoritmos de *automatic forecasting* más clásicos, pertenecientes al paquete *forecasting* de R elaborado por Hyndman y por otra, un algoritmo de una vida muy corta y que presume de mejorar a los anteriores y de ser mucho más automático, llamado *Prophet*. Además,

se ha querido incluir un algoritmo elaborado dentro del paquete *forecast* pero basado en una red neuronal y, por lo tanto, obvia el supuesto de linealidad.

De esta forma se presenta cada algoritmo estudiado desde una perspectiva teórico, así como sus principales resultados a la hora de predecir las tres series incluidas en el presente trabajo.

2.3.1 Modelos ETS

2.3.1.1 Enfoque teórico

Está basado en innovaciones de los modelos *state space* que subyacen de los métodos de suavizado exponencial. El suavizado o alisado exponencial basa su predicción en una ponderación de las observaciones pasadas, dando más peso normalmente a observaciones más recientes. De esta forma, dependiendo del comportamiento de la serie en torno a las diferentes componentes se aplicará un método u otro. Es aplicable tanto en series estacionales como en series no estacionales.

En cuanto a los métodos de suavizado exponencial, cabe destacar que hay 9 combinaciones posibles dado que la tendencia puede ser de tres tipos (inexistente, aditiva o multiplicativa), mientras que la componente estacional puede ser de los 3 mismos tipos.

Tabla 1: Modelos de suavizado exponencial

| TENDENCIA | ESTACIONALIDAD | | |
|------------------------|----------------|----------------|-----------------------|
| | N (Ninguna) | A (Aditiva) | M (Multiplicativa) |
| N(Ninguna) | N,N | N,A | N,M |
| A (Aditiva) | A,N | A,A | A,M |
| A_d (Aditiva húmeda) | A_d,N | A_d,A | A_d,M |

Fuente: Elaboración propia

Las siglas ETS provienen de los componentes: Error, Tendencia, Estacionalidad. Cabe señalar que el error puede ser tanto aditivo como multiplicativo. De esta forma, se tiene un total de 18 combinaciones posibles.

Cabe destacar que los métodos de suavizado exponencial permiten devolver estimaciones o predicciones puntuales. El algoritmo ETS permite la introducción de los modelos ISSM.

Las siglas ISSM significa *Innovation State Space Model*. Estos modelos permiten no solo estimaciones puntuales sino también intervalos de predicción, pudiendo llevar a cabo un análisis más riguroso.

Todos los modelos ETS pueden ser escritos en la forma ISSM. Además, las versiones aditivas y multiplicativas dan la misma estimación puntual aunque diferentes intervalos de predicción.

Alguna de las combinaciones ETS puede dar lugar a modelos numéricamente inestables¹⁸.

Por ejemplo, en un alisado exponencial simple la predicción $z(t+1)$ para el paso $t+1$ es

$$\hat{z}_{T+1} = \hat{z}_T + \alpha(z_T - \hat{z}_T) = \alpha \cdot z_T + (1 - \alpha) \cdot \hat{z}_T \quad (5)$$

Matemáticamente, ISSM es especificado mediante dos ecuaciones:

La ecuación de estado de transición viene dada por:

$$l_t = F_t l_{t-1} + g_t \varepsilon_t, \quad \varepsilon_t \sim N(0,1). \quad (6)$$

Nótese que la fuerza de innovación es controlada por

$$g_t, \text{ i. e., } g_t \varepsilon_t \sim N(0, g_t^2). \quad (7)$$

La ecuación de observación viene dada por:

$$z_t = a_t^I l_{t-1} + b_t + v_t, \quad v_t \sim N(0, \sigma_t^2) \quad (8)$$

Nótese que aquí se permite un término adicional b_t , el cual puede modelar cualquier componente determinista.

¹⁸ La estabilidad numérica es una propiedad general deseada en algoritmos numéricos. Se refiere, en general, a la capacidad de resolver ecuaciones por aproximación discreta.

Aquí se describe un modelo genérico permitiendo el uso de patrones específicos propios de las series temporales usando los coeficientes F_t y a_t . El vector de innovación g_t , viene en términos de parámetros para ser aprendido. El estado inicial l_0 tiene que ser especificado. Se especifica mediante una distribución a priori Gaussiana $P(l_0)$, los parámetros de la cual son aprendidos de los datos también.

En cuanto a la estimación de ISSM, ésta se hace mediante máxima verosimilitud. Es representada de la siguiente manera:

$$L^*(\theta, x_0) = n \log\left(\sum_{t=1}^n \varepsilon_t^2 / k^2(x_{t-1})\right) + 2 \sum_{t=1}^n \log|k(x_{t-1})| \quad (9)$$

$$= -2 \log(\text{Verosimilitud}) + \text{constante}$$

De esta forma, se estiman los parámetros $\theta = (\alpha, \beta, \gamma, \phi)$ y los estados iniciales $x_0 = (l_0, b_0, s_0, s_{-1}, \dots, s_{-m+1})$ minimizando L^*

En cuanto a la selección del mejor modelo dentro de los modelos idóneos, ésta se lleva a cabo a través del Criterio de Información de Akkaike. A continuación, se presenta la fórmula:

$$\text{AIC} = -2\log(L) + 2k \quad (10)$$

Donde L es la verosimilitud y k es el número de parámetros estimados del estado inicial en el modelo.

La implementación que Hyndman lleva a cabo en el paquete *forecasting* de R de la función *ets()* se realiza a través de los siguientes pasos:

Aplica cada modelo que es apropiado a los datos.

Optimiza parámetros y valores iniciales utilizando, por defecto, la estimación por máxima verosimilitud.

Selecciona el mejor método mediante el criterio del AIC

Una vez elegido el mejor modelo, construye las predicciones usando el mejor método a través de la función *forecast()*.

Obtiene los intervalos de predicción usando Modelos SS (*State Space*).

2.3.1.2 Serie diaria

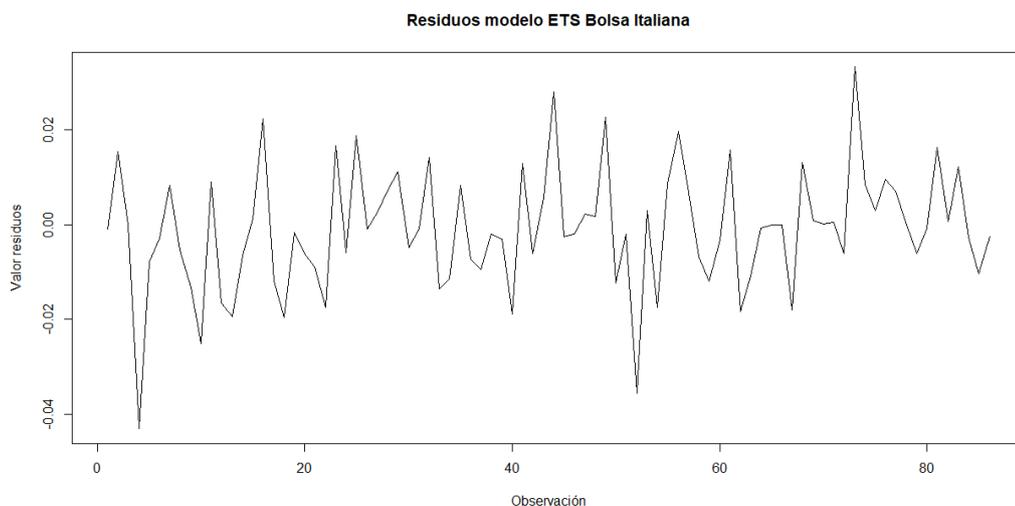
Una vez introducido brevemente las principales características del modelo ETS, así como su implementación a través del paquete *forecast* de Hyndman, se procede a ponerlo en práctica a través de las tres series antes anunciadas. De esta forma, también se evaluará si este algoritmo cumple con los objetivos propuestos de cada serie.

En primer lugar, se aplica la función *ets()* de R a la primera serie. Cabe destacar que solamente se introducen los valores, y no las fechas, ya que no hay ningún indicio de estacionalidad. Así pues, solamente nos interesa que la serie esté ordenada cronológicamente, arreglado anteriormente en el *pre-processing*.

El modelo resultante es un ETS(M,N,N), es decir, con error multiplicativo sin tendencia y sin estacionalidad. Al haber una tendencia cambiante, este algoritmo no ha detectado ningún tipo de tendencia. Además, como ya se sospechaba en la gráfica, no parece haber presencia de estacionalidad, o al menos, no para este algoritmo. El valor los parámetros de estimación del modelo son los siguientes:

- El parámetro de alisamiento α es de 0.9357
- El estado inicial I tiene un valor de 21358.4488
- La σ es de 0.0131
- El valor de AIC, criterio para decidir qué modelo es el elegido dentro de los adecuados, es de 1337.947.

Gráfico 4: Residuos del modelo ETS aplicado a la Serie diaria



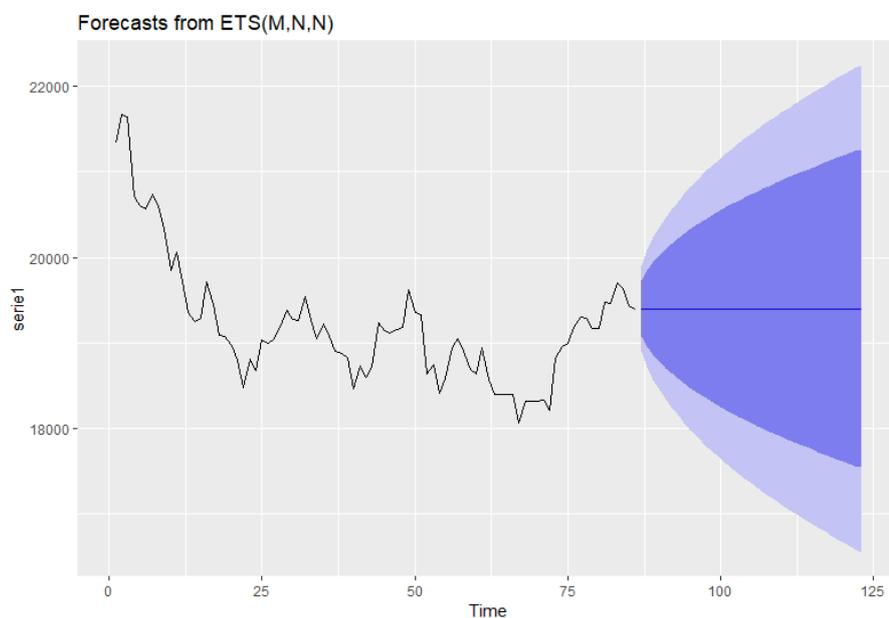
Elaboración: Fuente propia

En cuanto a la normalidad de los residuos, al aplicar el Test de Shapiro-Wilk, se puede observar como los residuos del modelo siguen una distribución Normal, ya que el p-valor es de 0.2621 (>0.05).

Una vez anunciado las principales características del modelo resultante, se va a llevar a cabo la predicción en el horizonte temporal planteado anteriormente (un total de 37 observaciones).

La representación gráfica de la predicción tanto puntual (media) como en intervalo (al 80 % en azul intenso y al 95% en azul claro) es la siguiente.

Gráfico 5: Predicción del periodo extramuestral del modelo ETS Serie diaria



Fuente: Elaboración propia

El resultado de esta predicción se ha anunciado implícitamente cuando se ha informado de que modelo ETS seguía. Al no tener tendencia ni estacionalidad, entonces predicción es la misma para todos los puntos, muy cercano al valor de la última observación del periodo muestral, ya que el parámetro α del modelo es muy cercano a 1. De esta forma, se procede a evaluar la predicción calculando los tres tipos de errores anunciado anteriormente, así como complementar el análisis a través de la comparación gráfica.

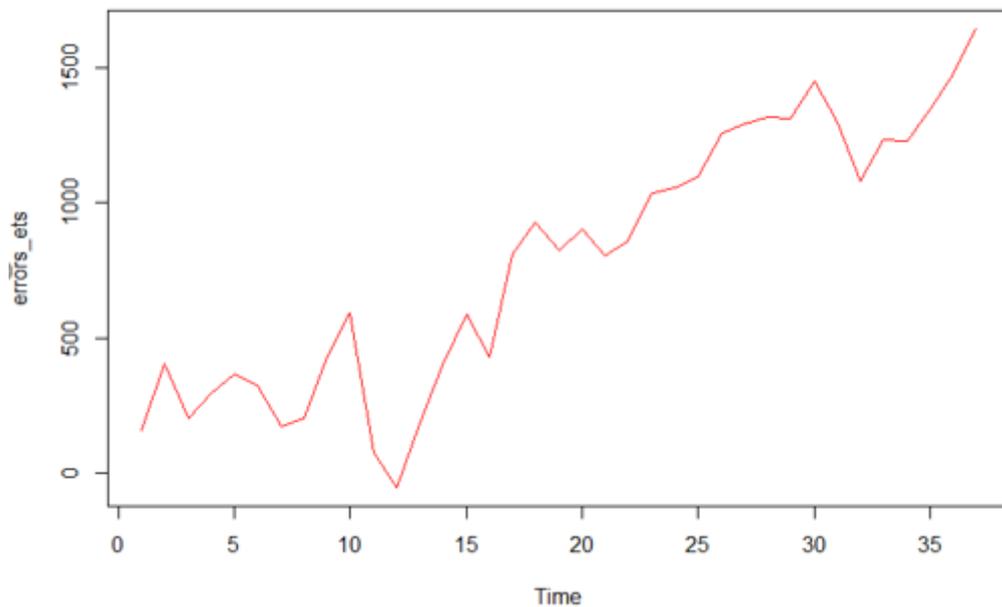
- El Error Cuadrático Medio es de 838941.8
- El Error Absoluto Medio es de 787.43

- El Error Porcentual Absoluto Medio es de un 3.84%

Aunque el EPAM es relativamente bajo, es decir, podría considerarse una buena predicción, cabe destacar que ha sido casualidad. Esta casualidad es fruto de la no muy alta volatilidad del periodo extramuestral respecto a la última observación del periodo muestral.

El siguiente gráfico muestra los errores de predicción:

Gráfico 6: Errores de predicción ETS Serie diaria



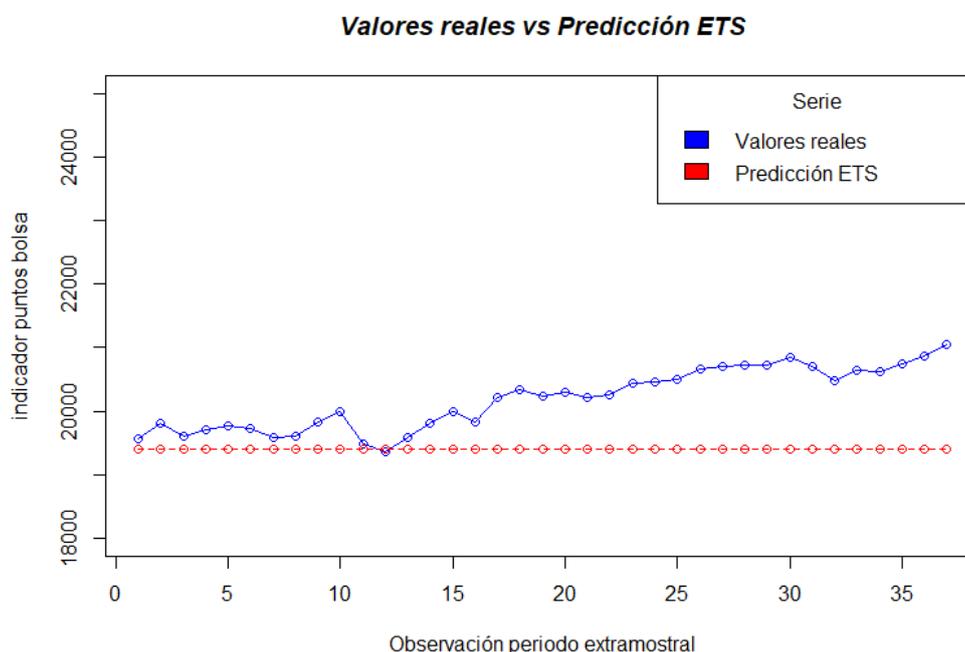
Fuente: Elaboración propia

En cuanto al contraste de hipótesis para evaluar la Normalidad de los residuos, el Test de Shapiro-Wilk indica que los residuos del periodo extramuestral no siguen una distribución Normal, ya que el p-valor es prácticamente 0.

Como se puede deducir, sigue la misma tendencia que la serie en el periodo de validación, ya que la predicción es la misma para todas las observaciones.

Finalmente, se hace una representación gráfica de los valores reales y los predichos para el periodo de validación.

Gráfico 7: Comparación valores reales con predicción ETS Serie diaria



Elaboración: Fuente propia

La conclusión que se puede extraer es que, aunque los indicadores de los errores son buenos (EPAM inferior al 5%) la interpretación gráfica no deja lugar a dudas de que este algoritmo ha realizado una mala predicción sobre la serie, ya que es constante.

En cuanto a la valoración de los objetivos, podemos concluir que el algoritmo ETS no ha conseguido hacer frente a los cambios de tendencia de la serie. Simplemente se ha limitado a concluir que no había ninguna tendencia significativa.

Una vez se ha visto como no es un buen método para predecir cambios de tendencia dentro de una misma serie, se procede a evaluar cómo se comporta delante de otra componente importa como es la estacionalidad. Para eso, se utilizará la segunda serie:

2.3.1.3 Serie mensual

Como se ha mencionado en la presentación de la segunda serie, referente a las pernoctaciones en Baleares, la serie se ha importado a R de tal forma que se señala que hay un periodo 12 (es de frecuencia mensual) para ayudar a detectar fácilmente la presencia de estacionalidad.

El modelo elegido es un ETS(M,N,M), tal y como se esperaba a vistas del gráfico.

Los parámetros de alisado son: $\alpha=0.1726$ y $\gamma=0.4093$

Los estados iniciales son: $l = 1187740.6611$

Los 12 valores del parámetro s referente al componente estacional (s_{MES}) son:

Tabla 2: Valores del parámetro s de cada mes del modelo ETS(M,N,M)

| mes | ENE | FEB | MAR | ABR | MAY | JUN | JUL | AGO | SET | OCT | NOV | DIC |
|-----|--------|-------|--------|--------|--------|--------|--------|-------|--------|--------|--------|--------|
| s | 0.0279 | 0.034 | 0.3522 | 0.0047 | 0.4307 | 0.3444 | 0.8867 | 0.327 | 0.3868 | 0.0943 | 0.0593 | 0.0521 |

Elaboración: Fuente propia

Recordemos que en caso de un modelo ETS(M,N,M) las s_t equivale a:

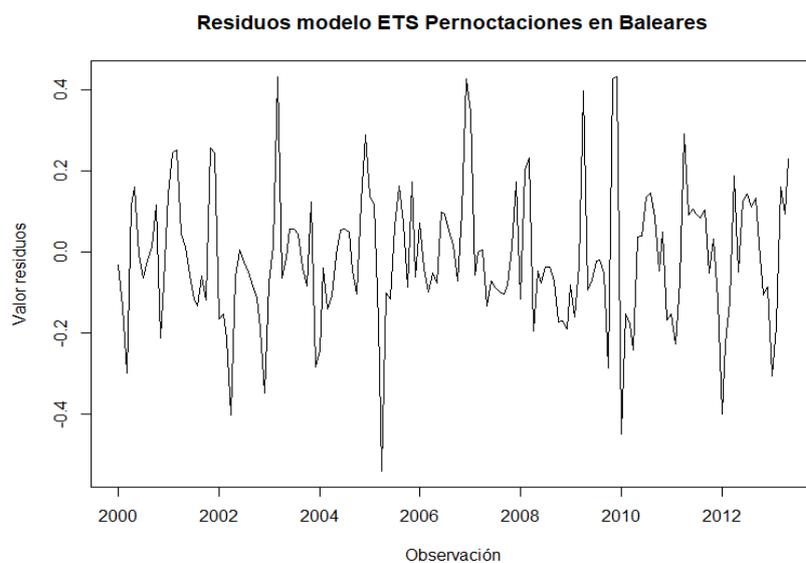
$$s_t = s_{t-m} + \gamma \mu_t \epsilon_t / \ell_{t-1} \quad (11)$$

El valor σ es de 0.1753

El valor del AIC es de 4316.706

A continuación, se representa gráficamente los residuos del modelo estimado.

Gráfico 8: Residuos del modelo ETS Serie mensual



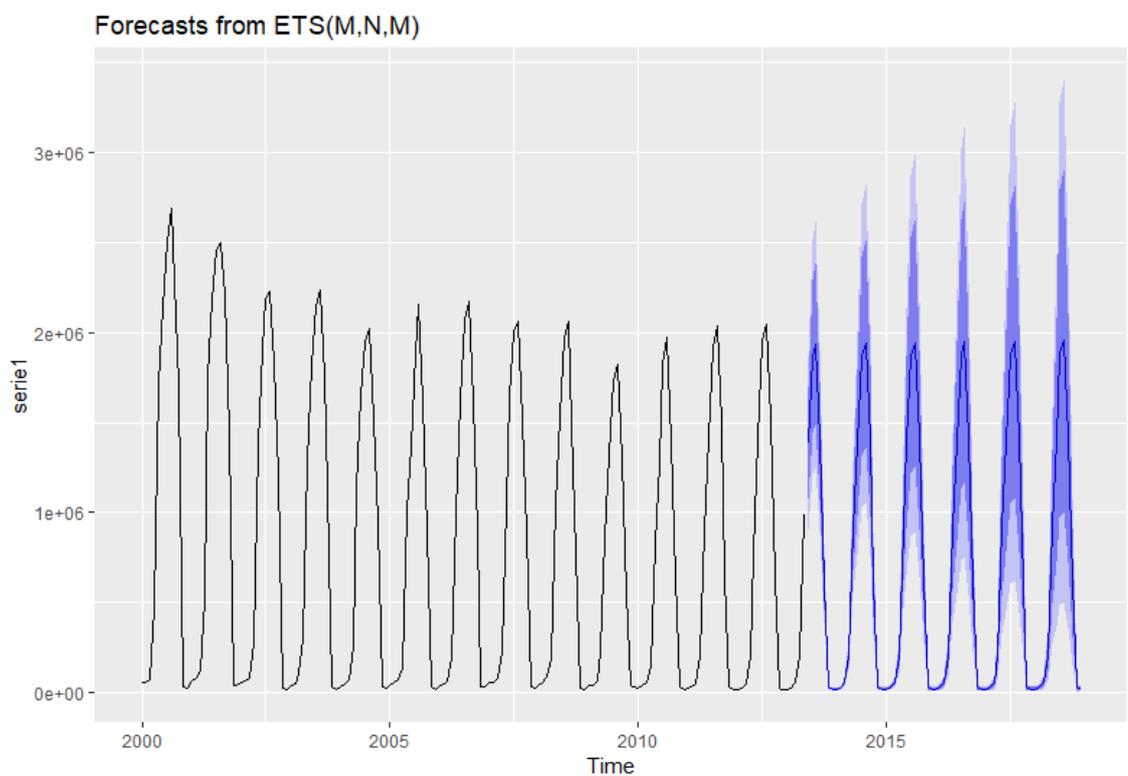
Elaboración: Fuente propia

Aunque parece que los valores oscilan entorno a una media de 0, sí que aparecen serias dudas de que la varianza sea constante. Al aplicar el Test de Shapiro-Wilk, se puede observar como los residuos del modelo no siguen una distribución Normal, ya que el p-valor es de 0.016 (<0.05).

Seguidamente, se lleva a cabo la predicción del periodo extramuestral. Recordemos que el total de observaciones de este periodo es de 67.

La predicción tanto puntual como por intervalos se ve reflejada en el siguiente gráfico:

Gráfico 9: Predicción periodo extramuestral ETS Serie mensual



Elaboración: Fuente propia

Parece que ha reconocido de forma correcta la estructura de la componente estacional a lo largo de los meses, a la vez de que también reconoce que hay una ligera tendencia negativa, al menos en los picos (julio y agosto). Antes de entrar en evaluaciones, a priori parece que el algoritmo ha hecho una muy buena predicción.

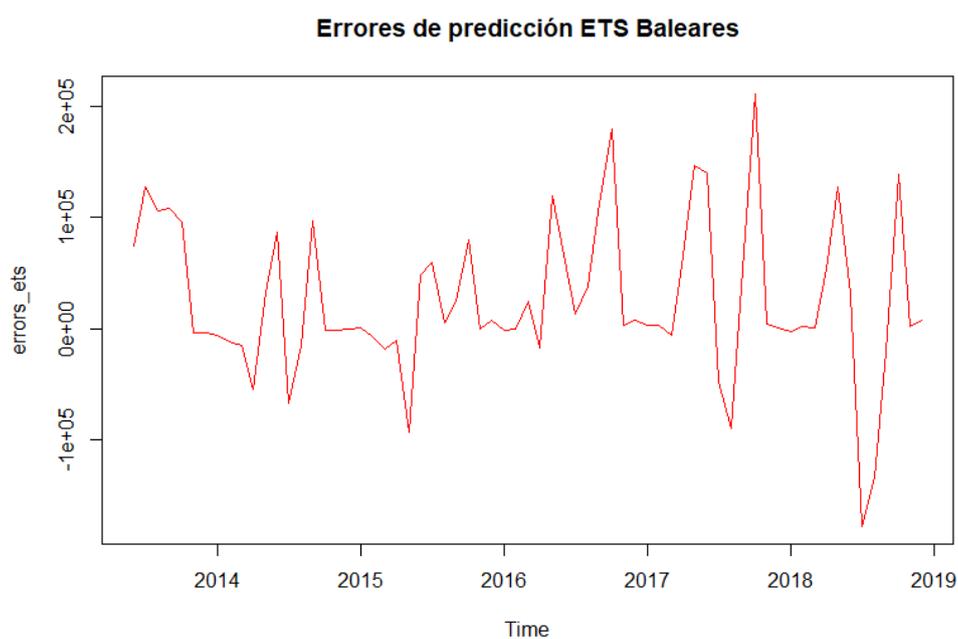
Seguidamente se pasa a evaluar tanto analíticamente como gráficamente la predicción del periodo de validación:

- El Error Cuadrático Medio es de 5414257988
- El Error Absoluto Medio es de 49581.55
- El Error Porcentual Absoluto Medio es de un 12.14%

La conclusión inicial del resultado de estos indicadores es: aunque la teoría diga que no es un buen método de predicción (ya que el EPAM es superior al 10%), al haber ya una importante cantidad de datos, sumado a que la serie es un poco más compleja, no parece que sea tan mal algoritmo en este caso.

A continuación, se presenta el análisis gráfico de la predicción:

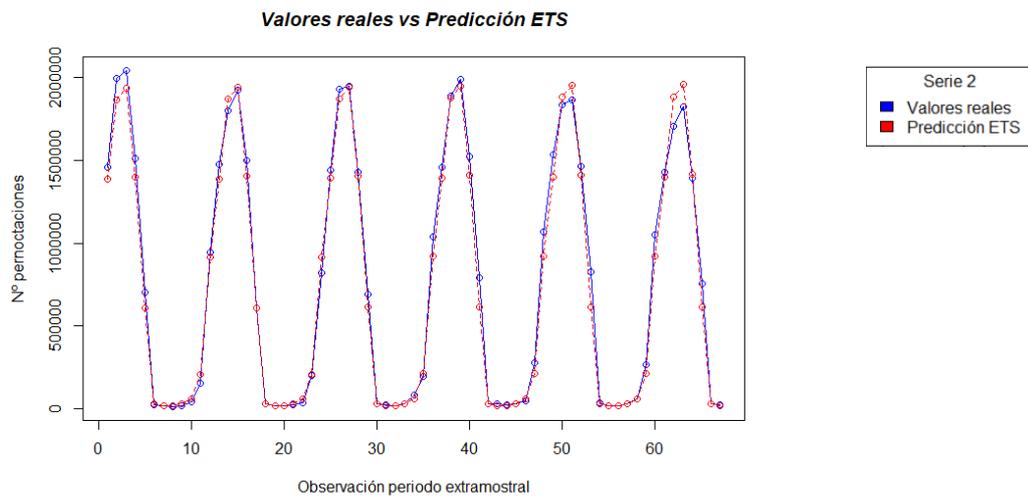
Gráfico 10: Errores de predicción modelo ETS Serie mensual



Elaboración: Fuente propia

Los errores no acaban de seguir un patrón claro, ya que la varianza es volátil. Además, no parece que los errores estén centrados a 0. En cuanto al contraste de hipótesis para evaluar la Normalidad de los residuos, el Test de Shapiro-Wilk indica que los residuos del periodo extramuestral no siguen una distribución Normal, ya que el p-valor es prácticamente 0.

Gráfico 11: Comparación valores reales con predicción modelo ETS Serie mensual



Fuente: Elaboración propia

Se puede observar como el algoritmo sigue claramente la estacionalidad. No obstante, como no detecta ningún tipo de tendencia clara, mantiene los valores constantes para cada mes en todos los años. De esta forma, el error se debe a las pequeñas variaciones entre los meses durante los años de periodo de validación.

De esta forma, se puede concluir que la función *ets()* sí que ha cumplido el objetivo de ajustar bien la estacionalidad. Sin embargo, más adelante se verá si los otros algoritmos son capaces de superarlo.

2.3.1.4 Serie cinco-minutal

Para terminar el estudio de esta función, solamente resta evaluar su capacidad predictiva a través de la última serie propuesta en este trabajo: el número de bicis Bicing en uso. Recordemos que el objetivo de esta serie era evaluar la capacidad de los diferentes algoritmos estudiados delante de un gran número de datos, así como la más que probable presencia de estacionalidades de largos periodos.

Los principales resultados del modelo son:

- Es un ETS(M,Ad,N): el error es multiplicativo, la tendencia es aditiva humedecida mientras que no estima ningún tipo de estacionalidad

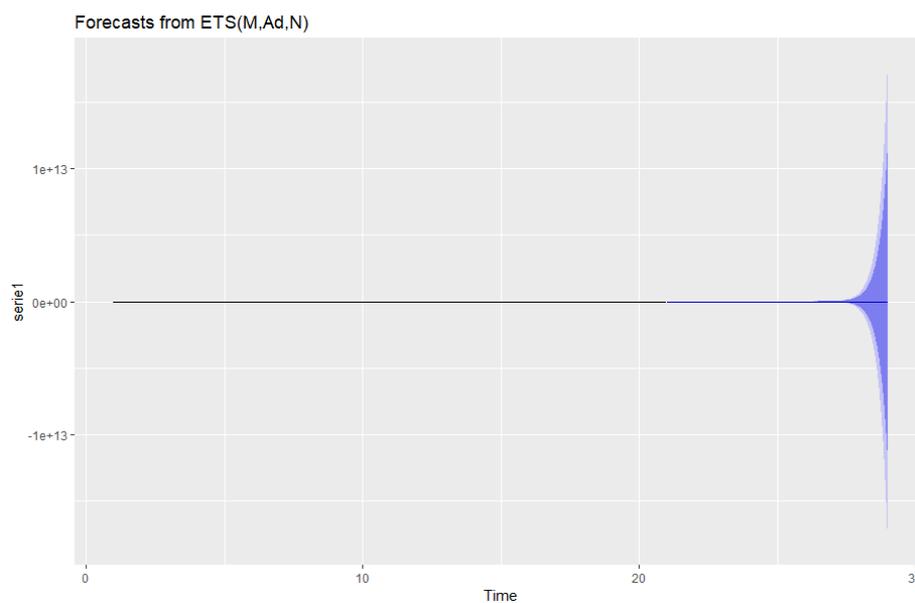
- Los parámetros de alisamiento son:
 - $\alpha = 0.9999$
 - $\beta = 0.0476$
 - $\phi = 0.8965$
- Los estados iniciales son:
 - $l = 148.0957$
 - $b = -2.6199$
- El valor σ es de 0.1124
- El valor del AIC es de 86392.95

Como se puede suponer, la predicción será una línea constante igual a la última observación del periodo muestral, ya que el parámetro α es prácticamente igual a 1. Destaca que no se considere que hay presencia de estacionalidad. El motivo es que si no se le indica lo contrario a la hora de importar la serie supone que no hay estacionalidad.

Los residuos del modelo no siguen una distribución Normal. Así lo demuestra el test de Kolmogorov-Smirnov.

Aunque ya se sabe con certeza que será una predicción totalmente errónea, se procede a su evaluación para después poder comentar y comparar con los otros algoritmos.

Gráfico 12: Predicción periodo extramuestral modelo ETS Serie cinco-minutal



Fuente: Elaboración propia

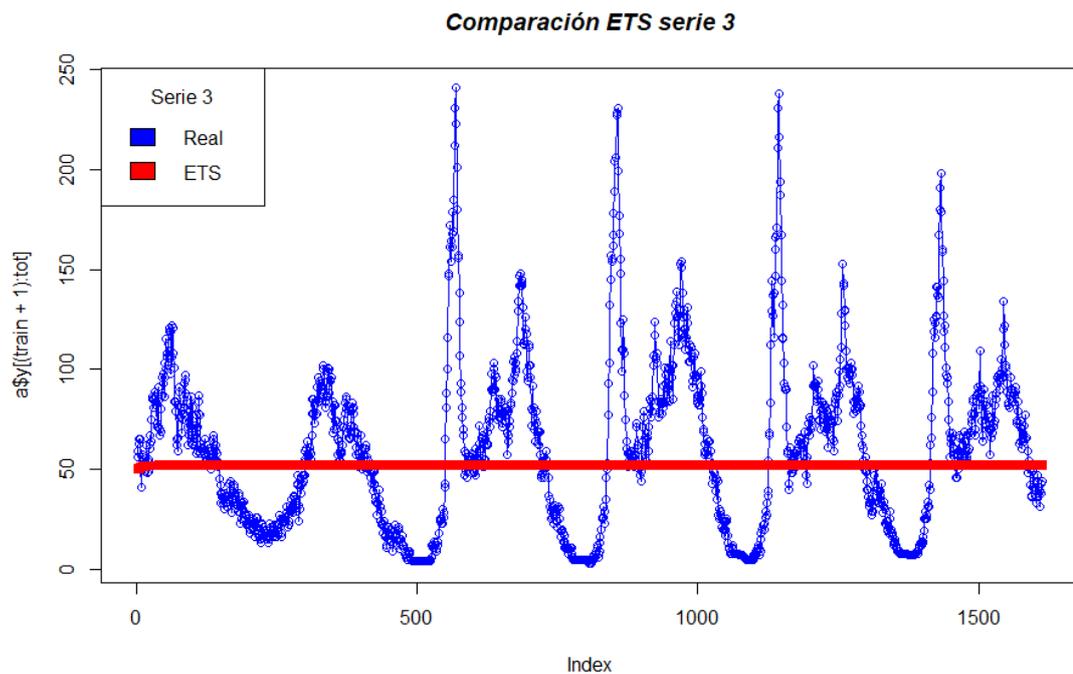
En cuanto a la evaluación de los indicadores de los diferentes errores:

- El Error Cuadrático Medio es de 1809.403
- El Error Absoluto Medio es de 32.83895
- El Error Porcentual Absoluto Medio es del 151.19%.

Más allá del EPAM altísimo, el principal defecto de este algoritmo es que no ha sido capaz de predecir la estacionalidad.

No tiene ningún sentido, ya que los intervalos se pueden suponer que abarca todos los números reales.

Gráfico 13: Comparación valores reales y predicción modelo ETS Serie cinco-minutal



Fuente: Elaboración propia

Esta función no puede manejar series con una frecuencia superior a 24. Por lo tanto, ya tenemos una de las limitaciones presentes en esta función.

2.3.2. Modelos ARIMA

2.3.2.1. Enfoque teórico

En primer lugar, se da una breve introducción de lo que son los modelos ARIMA.

Las siglas provienen de Modelo Autoregresivo (AR(p)) Integrado (I(d)) de Medias Móviles (MA(q)). De esta forma, el modelo general ARIMA (p,d,q) toma la expresión:

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)(1 - B)^d X_t = (1 - v_1 B - v_2 B^2 - \dots - v_q B^q) a_t \quad (12)$$

- p = número de términos autorregresivos
- d = número de diferencias
- q = número de términos de media móvil

Tiene dos tipos de modelo tales como el estacional y el no estacional.

Un proceso ARIMA no estacional es de la forma:

$$\phi(B)(1 - B)^d y_t = c + \theta(B)\varepsilon_t \quad (13)$$

Se necesita seleccionar apropiar el orden de los parámetros p, q, d, y si incluir c o no.

La función *auto.arima* en un proceso no estacional:

1. Selecciona el número de diferencias d vía el test de la raíz unitaria KPSS.
2. Selecciona p, q, c minimizando el AIC.
3. Usando una búsqueda *stepwise*, empezando con un modelo simple y considerando en cada paso variantes vecinas.

Un proceso ARIMA estacional es de la forma:

$$\Phi(B^m)\phi(B)(1 - B)^d(1 - B^m)^D y_t = c + \Theta(B^m)\theta(B)\varepsilon_t \quad (14)$$

Se necesita seleccionar los órdenes apropiados p, q, d, P, Q, D y decidir si incluir c.

La función *auto.arima* en un proceso estacional:

1. Selecciona el número de diferencias d vía el test de la raíz unitaria KPSS.
2. Selecciona D (número de diferencias en la parte estacional) usando el test de OCSB.
3. Selecciona p, q, P, Q, c minimizando el AIC.
4. Usa un *stepwise* empezando por un modelo simple y considerando en cada paso variantes vecinas.

Una vez se ha resumido tanto la teoría del algoritmo como sus diferentes criterios para elegir el modelo ARIMA en R, se va a proceder a su evaluación práctica a través de las tres series propuestas en este trabajo.

2.3.1.2 Serie diaria

En primer lugar, se va a trabajar con la serie de frecuencia diaria referente a la cotización de la Bolsa Italiana. Cabe destacar que no se ha especificado nada en el input de la función *auto.arima* ni se ha retocado nada adicional de esta serie temporal más allá de lo hecho en el *pre-processing*. De esta forma, la función *auto.arima()* para esta primera serie ha dado el siguiente output:

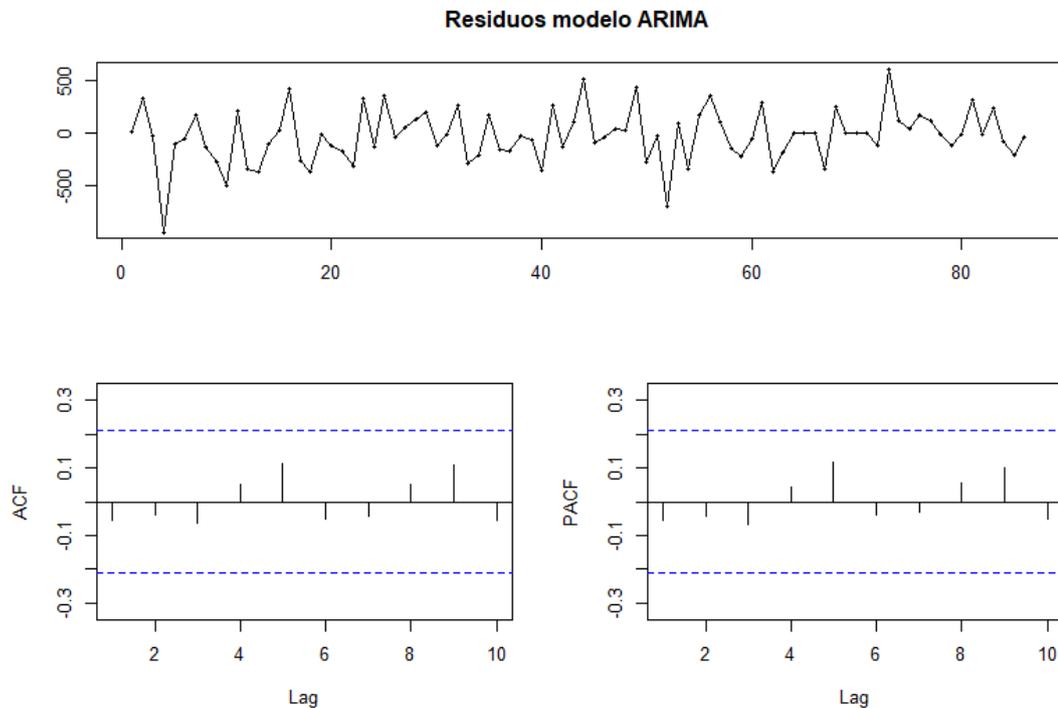
El algoritmo de Hyndman elige el modelo ARIMA no estacional (0,1,0) donde destaca únicamente una diferencia. Se procede a comprobar que tiene sentido:

Se realiza un test de estacionaridad, en este caso el Test Aumentado de Dickey-Fuller, indicando que como hipótesis alternativa se tiene que la serie es estacionaria. Con un p-valor de 0.5097 se concluye que no estacionaria. Ahora bien, después de realizar una diferencia a la misma serie se vuelve a hacer el mismo test. Ahora el p-valor es de 0.01892. En este caso, se concluye que la serie diferenciada es estacionaria. Por lo tanto, el orden d es correcto.

Además, los parámetros de media móvil y autoregresión son de orden 0. Esto significa que se encuentra en el mismo caso que el algoritmo anterior, la predicción será constante durante el periodo extramuestral. Cuadra con la gráfica ya que no parece que las observaciones estén correlacionadas ni que haya ninguna línea de tendencia continua.

A continuación, se procede a evaluar gráficamente los residuos del modelo:

Gráfico 14: Características residuos modelo ARIMA Serie diaria



Fuente: Elaboración propia

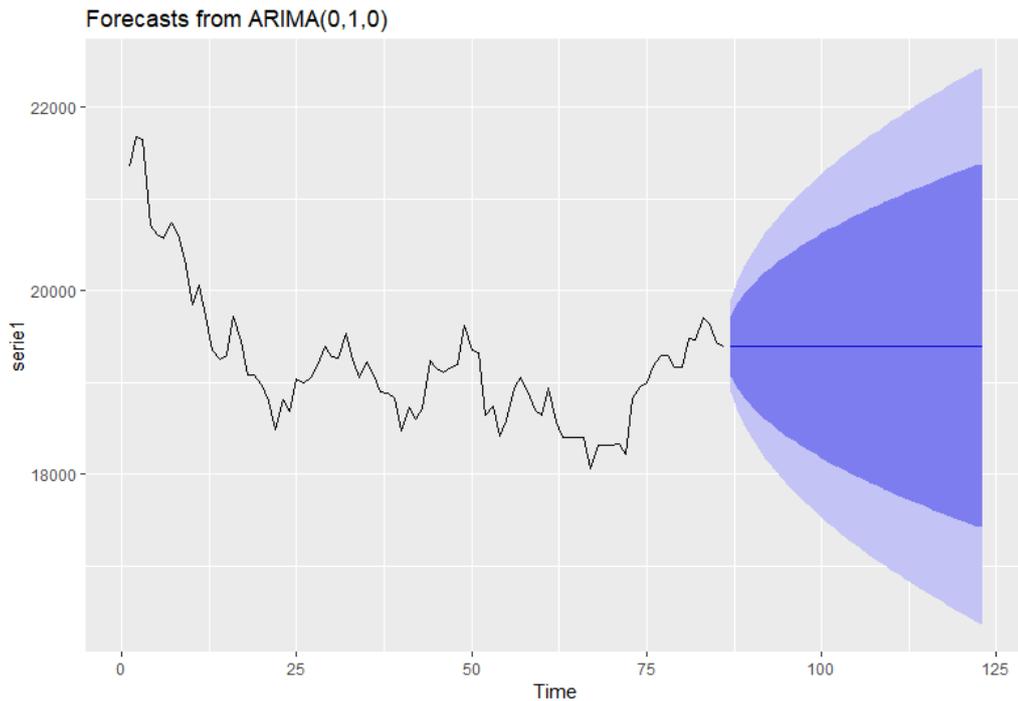
Parecen que son homocedásticos con media 0. Al aplicar el Test de Shapiro-Wilk, se puede observar como los residuos del modelo siguen una distribución Normal, ya que el p-valor es de 0.0828 (>0.05).

Además, la Función de Autocorrelación y la Función de Autocorrelación Parcial están dentro de los intervalos establecidos como estables. De esta forma, no se presenta ningún modelo ARIMA como alternativa.

La σ^2 es de 64581. El valor del AIC es de 1181.65 .

Aunque a sabiendas de la teoría y del resultado del modelo ARIMA ya se puede imaginar el resultado de la predicción (una recta constante), se va a proceder a la elaboración de la predicción del periodo extra-mostrual y a su validación.

Gráfico 15: Predicción periodo extramuestral ARIMA serie diaria



Fuente: Elaboración propia

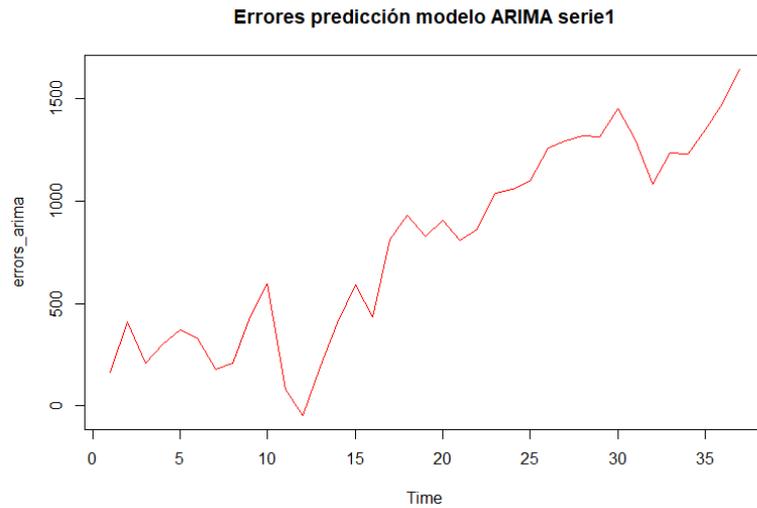
Se puede observar como en la función ETS() que la estimación puntual es una recta constante y que los intervalos de predicción son muy altos. De hecho, con un 80 % de confianza prácticamente abarcan desde el valor mínimo hasta el valor máximo.

De esta forma, se procede a la evaluación de los errores de predicción y a la comparación de los valores predichos con los reales:

- El Error Cuadrático Medio es de 844018.9
- El Error Absoluto Medio es de 790.49
- El Error Porcentual Absoluto Medio es del 3.86%

Como en el anterior caso, el EPAM indica que el modelo hace una buena predicción. Sin embargo, como se puede observar es simple casualidad, ya que como asume que no hay ningún tipo de tendencia, estos indicadores dependerán de la volatilidad real de la serie temporal en el periodo extramuestral.

Gráfico 16: Error de predicción del modelo ARIMA Serie diaria

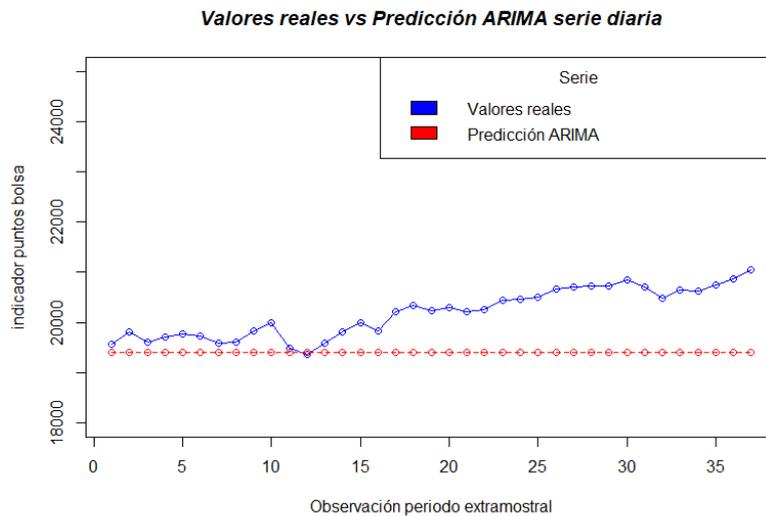


Fuente: Elaboración propia

La evolución de los errores sigue una tendencia calcada a los datos reales del periodo de validación. En cuanto al contraste de hipótesis para evaluar la Normalidad de los residuos, el Test de Shapiro-Wilk indica que los residuos del periodo extramuestral no siguen una distribución Normal, ya que el p-valor es prácticamente 0.

Para finalizar la evaluación de este algoritmo mediante la serie diaria, se presenta el gráfico de comparación entre datos reales y estimados para el periodo de test:

Gráfico 17: Comparación valores reales con la predicción ARIMA Serie diaria



Fuente: Elaboración propia

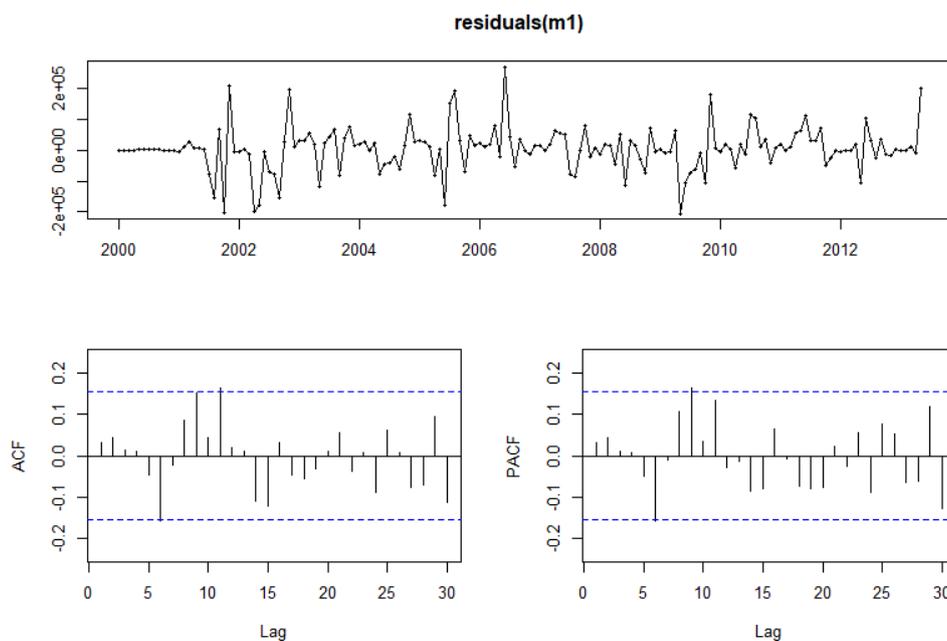
Los resultados son prácticamente idénticos al del método anterior. Con lo cual, al igual que en el caso, anterior se concluye que, a pesar de tener unos indicadores muy buenos, estos dos algoritmos no han sido capaces de parametrizar ningún tipo de tendencia. De esta manera, se tendrá que ver si Prophet es capaz de ofrecer algún tipo de estimación más útil para series con este tipo de comportamiento.

2.3.1.3 Serie mensual

Seguidamente, se pasa a evaluar esta función a través de la segunda serie (pernoctaciones en Baleares). Recordemos que el objetivo de esta serie es ver cómo responden las diferentes funciones delante de una serie claramente estacional. En este caso, se lleva a cabo el mismo proceso de preparación que en ETS. Se importa la serie de tal manera que se indica que empieza en enero del año 2000 y que es de frecuencia 12 (mensual).

El modelo ARIMA resultante de la función *auto.arima* es $(2,1,1)(2,1,2)[12]$. Por lo tanto, hay presencia tanto de parte estacional como de parte no estacional, como era fácil de prever. De esta forma, se procede a analizar los residuos:

Gráfico 18: Características residuos modelo ARIMA Serie mensual

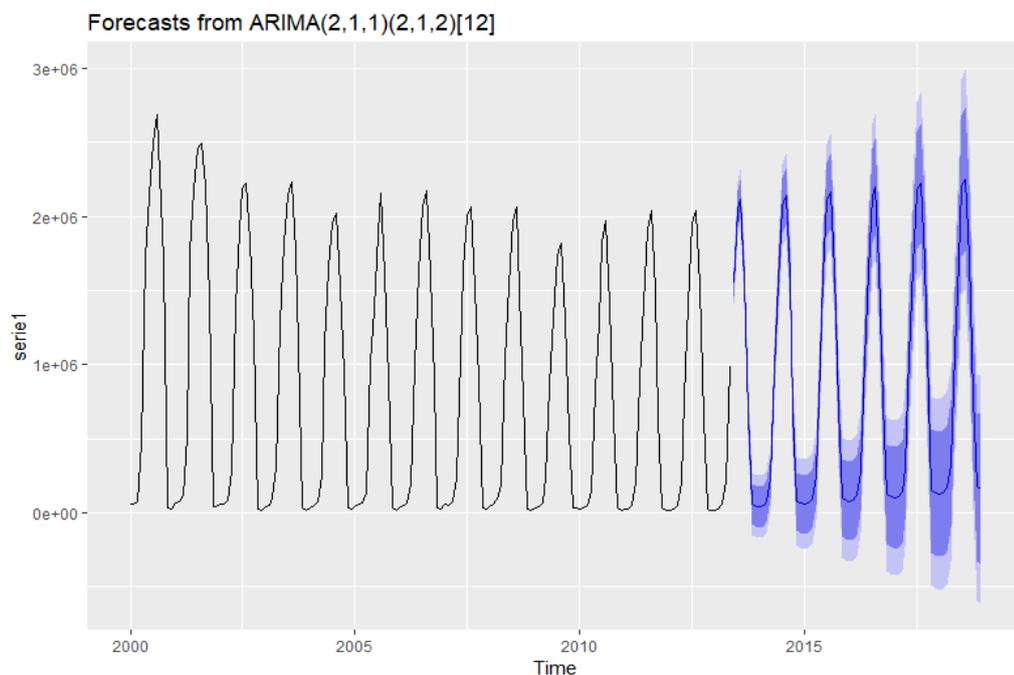


Fuente: Elaboración propia

Como se puede observar, mediante la FAS y la FAP se puede observar cómo están dentro de los intervalos, aunque muy justo. En cuanto a la distribución de los residuos, se puede deducir que la varianza no es constante. Al aplicar el Test de Shapiro-Wilk, se puede observar como los residuos del modelo no siguen una distribución Normal.

Seguidamente, se lleva a cabo la predicción para el horizonte temporal establecido anteriormente. Cabe recordar que es un total de 67 observaciones. A continuación, se presenta el gráfico correspondiente:

Gráfico 19: Predicción periodo extramuestral ARIMA Serie mensual



Fuente: Elaboración propia

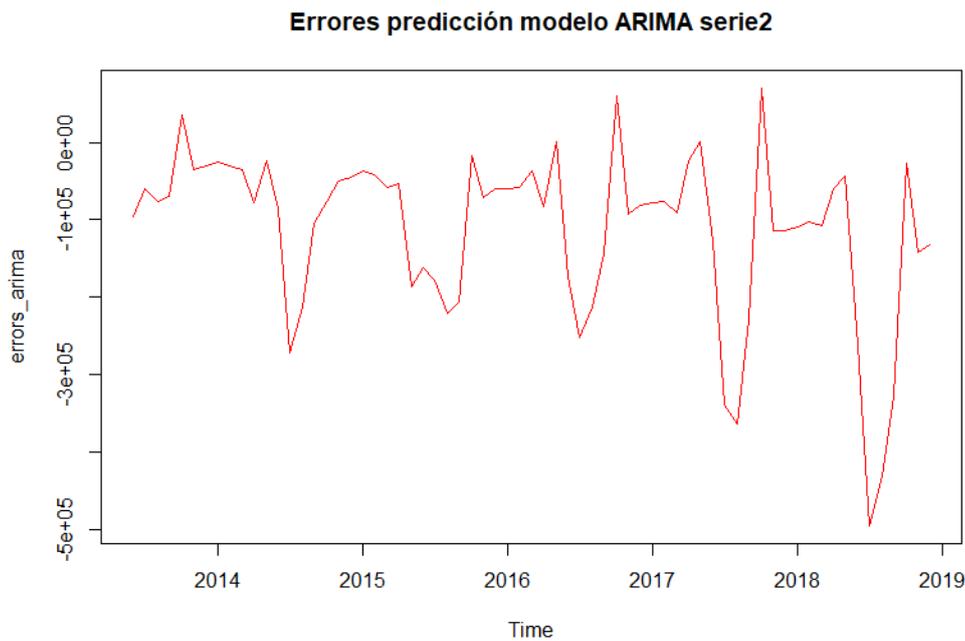
Cabe señalar que ha reconocido de forma correcta la presencia y el patrón de estacionalidad, de aquí a que se haya elegido un modelo SARIMA. A diferencia de algoritmo automático anterior, en este caso *auto.arima* predice una tendencia ligeramente positiva respecto a los últimos años del periodo de entrenamiento. Recordemos que *ets()* ha predicho una tendencia ligeramente negativa. Como ya se tiene constancia de que los datos reales muestran de que hay una tendencia bajista, parece que la validación de la predicción del algoritmo automático de ARIMA no será tan buena como en el caso de los ETS.

Igualmente, se presentan los mismos indicadores analíticos y gráficos para evaluar la exactitud de la predicción:

- El Error Cuadrático Medio es de 44184422400.
- El Error Absoluto Medio es de 212405.8.
- El Error Porcentual Absoluto Medio es del 200.397%.

Como se puede observar el EPAM es altísimo, lo que demuestra que hay una diferencia abismal entre el valor real y el predicho. Podrá observarse de mejor forma, a través de los siguientes gráficos (errores y comparación).

Gráfico 20: Errores de predicción modelo ARIMA Serie mensual

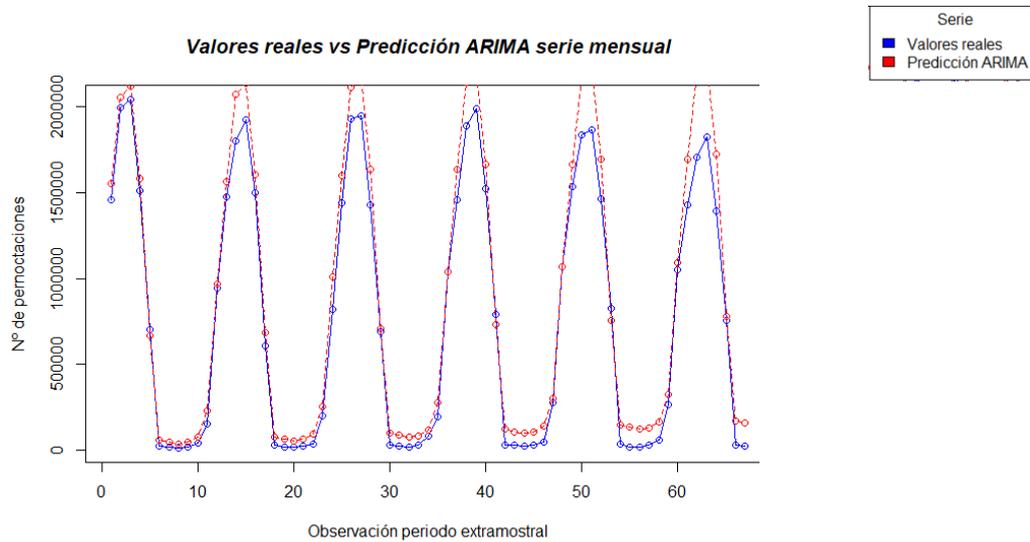


Fuente: Elaboración propia

El gráfico de errores muestra que la predicción, además de distar mucho de la realidad, sobrevalora muchísimo a las predicciones. Si se observa atentamente al gráfico, prácticamente todas las observaciones están por debajo del 0. Esto significa que el valor predicho es mayor al valor real. En cuanto al contraste de hipótesis para evaluar la Normalidad de los residuos, el Test de Shapiro-Wilk indica que los residuos del periodo extramuestral no siguen una distribución Normal, ya que el p-valor es prácticamente 0.

Para confirmar el análisis, se presenta el último gráfico:

Gráfico 21: Comparación valores reales y predicción ARIMA Serie mensual



Fuente: Elaboración propia

Efectivamente, aunque el patrón de la estacionalidad parece seguirlo con buen criterio, las predicciones del número de pernoctaciones están muy por encima de la realidad. Por lo tanto, en este apartado se puede concluir que el algoritmo de predicción automática de *auto.arima()* tampoco ha cumplido los objetivos planteados en esta serie: comportamiento para frecuencia mensual y estimación de una serie con una clara componente estacional.

De momento, si se tuviera que elegir entre los dos primeros algoritmos, cabe destacar que ETS ha predicho mejor, sobre todo la estacionalidad en el segundo caso.

2.3.1.4 Serie cinco-minutal

Ahora pues, solamente resta evaluar los modelos ARIMA en una serie cinco-minutal con una estacionalidad de periodo muy largo y un número de observaciones mucho mayor a las anteriores.

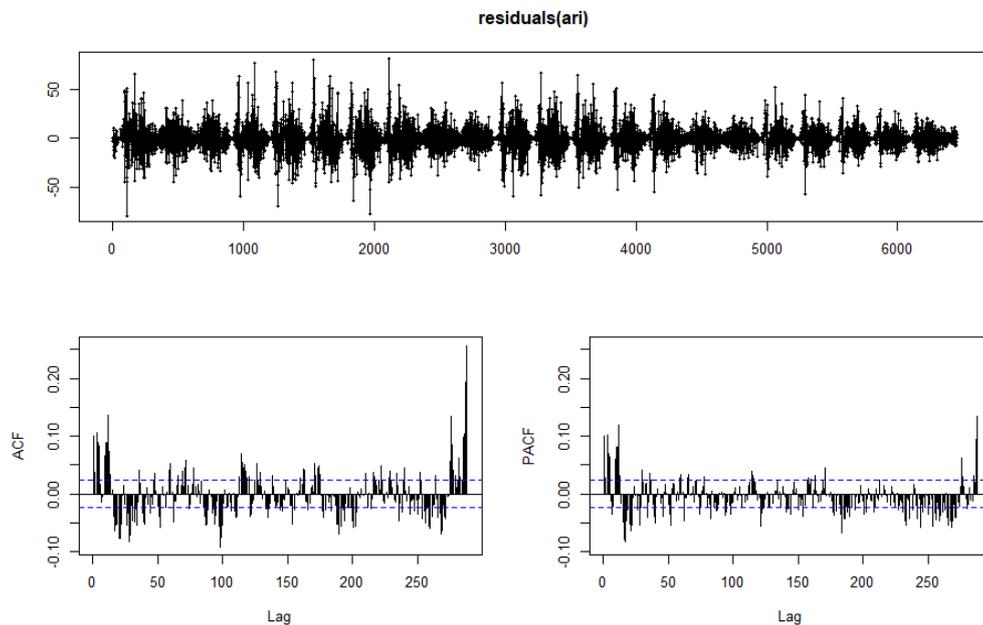
Al intentar aplicar la función *auto.arima()* a una serie de periodo 288, no se ha podido ejecutar, ya que ésta estaba tardando tanto tiempo que no se ha podido dictaminar si de verdad era capaz de llevar a cabo la predicción o en caso contrario pasaba igual que en ETS, el cual no puede predecir series de periodo superior a 24.

El algoritmo ARIMA de Hyndman selecciona un ARIMA(2,1,2) con los siguientes coeficientes:

- Ar1= 0.8801
- Ar2=-0.8574
- Ma1=-0.6761
- Ma2=0.7684
- La σ^2 estimada es 175.1
- El valor AIC es 51627.42

Seguidamente se procede a evaluar los errores:

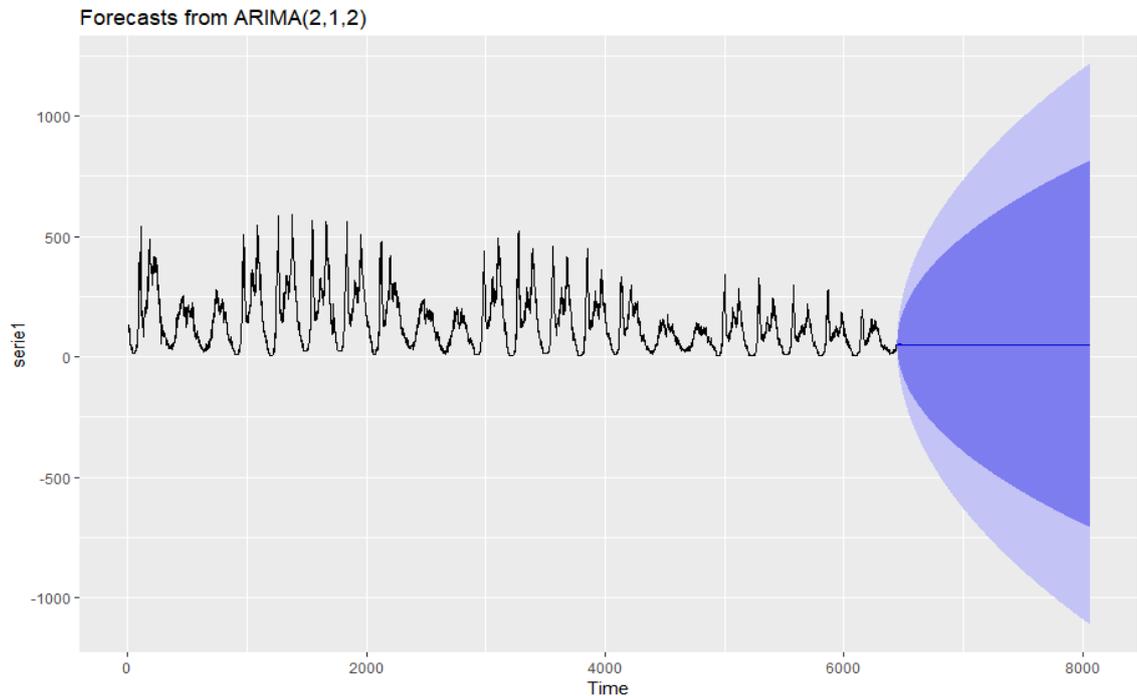
Gráfico 22: Comportamiento residuos modelo ARIMA Serie cinco-minutal



Fuente: Elaboración propia

Se puede observar claramente, como es necesario llevar a cabo un modelo SARIMA, con lo que en este caso no funcionaría el algoritmo automático. En cuanto a la normalidad de los residuos, al aplicar el Test de Kolmogorov-Smirnov se puede observar como los residuos del modelo no siguen una distribución Normal.

Gráfico 23: Predicción periodo extramuestral ARIMA Serie cinco-minutal



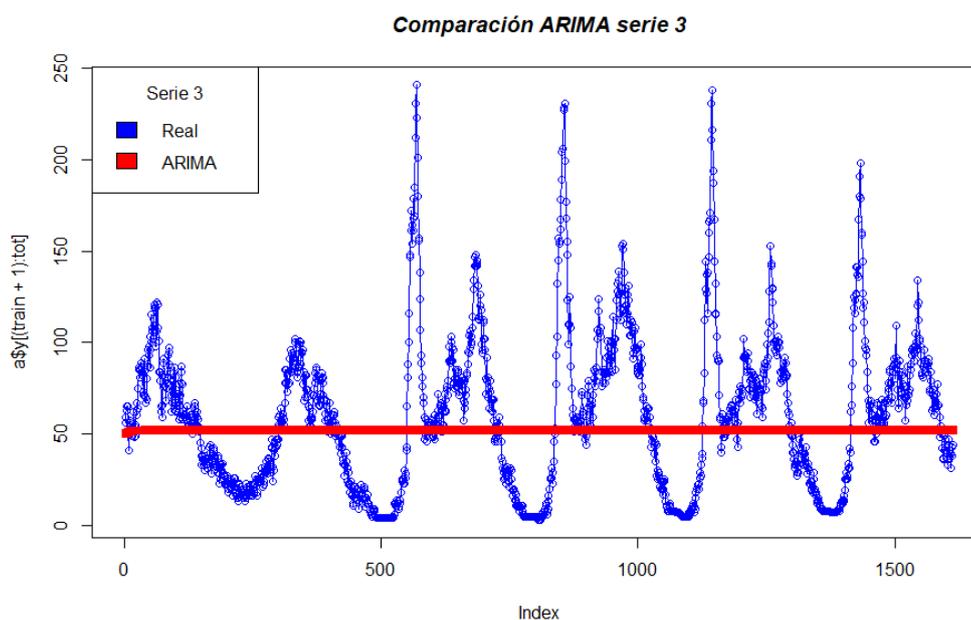
Fuente: Elaboración propia

Respecto al contraste de hipótesis para evaluar la Normalidad de los residuos, el Test de Shapiro-Wilk indica que los residuos del periodo extramuestral no siguen una distribución Normal, ya que el p-valor es prácticamente 0.

En cuanto al valor de los indicadores:

- El Error Cuadrático Medio es de 1813.039
- El Error Absoluto Medio es de 32.87
- El Error Porcentual Absoluto Medio es de 150.56%.

Gráfico 24: Comparación valores reales y predicción ARIMA Serie cinco-minutal



Fuente: Elaboración propia

2.3.2 Prophet

2.3.2.1 Enfoque teórico

Como es el algoritmo de *automatic forecasting* más reciente y con el que más se ha trabajado de forma manual en el presente trabajo, es el que se hará más hincapié tanto a nivel teórico como a nivel de implementación en R, llevando a cabo no solo comparaciones con los dos anteriores algoritmos de Hyndman sino con diferentes decisiones que se han tomado para poner de inputs en esta misma función.

Prophet es un algoritmo de código abierto desarrollado en febrero de 2017 por Facebook. Pertenece a la familia de algoritmos GAM (modelos aditivos generalizados). Se propone un modelo de regresión modular combinado con parámetros interpretables que pueden ser ajustables intuitivamente por un analista con un conocimiento avanzado en series temporales.

En este algoritmo se usa un modelo que se pueden descomponer en tres componentes principales, las cuales se estiman por separado.

$$y(t) = g(t) + s(t) + h(t) + e(t) \quad (15)$$

- $g(t)$: tendencia
- $s(t)$: componente estacional
- $h(t)$: las vacaciones (*holidays*)

El término e (*épsilon*) se refiere al término del error. Se hace el supuesto paramétrico de que este término se distribuye de forma Normal.

Como se puede observar el esquema es muy similar al modelo aditivo generalizado. La ventaja que esto conlleva es que la formulación se descompone fácilmente y es fácil de acomodar nuevos componentes al modelo si fuera necesario.

De esta forma se procede a analizar de forma resumida como se estima cada componente:

Tendencia: el tipo de crecimiento está típicamente modelado usando el modelo de crecimiento logístico, del cual su forma base sería del tipo:

$$g(t) = \frac{C}{1 + \exp(-k(t - m))} \quad (16)$$

Donde C es la capacidad de carga¹⁹, k el ratio de crecimiento y m un parámetro offset

No obstante, se introducen algunos cambios, para adaptarlo más a la realidad del modelo. Los principales cambios son:

C pasa a ser $C(t)$, es decir, varía en función del tiempo.

La tasa de crecimiento no es constante. De esta manera, se incorporan cambios de tendencia en el modelo de crecimiento a través de la definición explícita de los cambios de punto. Así pues, el modelo de crecimiento logístico es:

¹⁹ La capacidad de carga C es un número que indica el punto de saturación en una observación, es decir, el número mínimo/máximo que ésta puede alcanzar.

$$g(t) = \frac{C(t)}{1 + \exp(-(k + a(t)^T \delta)(t - (m + a(t)^T \gamma)))} \quad (17)$$

donde $a(t)$ indica si es cambio de punto o no y δ es la tasa de ajuste.

En cambio, si la tendencia es lineal se tiene que:

$$g(t) = (k + a(t)^T \delta)t + (m + a(t)^T \gamma) \quad (18)$$

Estacionalidad: Prophet hace uso de serie de Fourier para proporcionar una modelo flexible de efectos periódicos. Se pueden aproximar los efectos estacionales a través de la función:

$$s(t) = \sum_{n=1}^N (a_n \cos\left(\frac{2\pi nt}{p}\right) + b_n \sin\left(\frac{2\pi nt}{p}\right)) \quad (19)$$

El componente estacional es

$$s(t) = X(t)\beta \quad (20)$$

donde

$$\beta \sim Normal(0, \sigma^2) \quad (21)$$

La opción de todos los parámetros puede ser automatizada a través de los procedimientos de selección tales como AIC.

Vacaciones y eventos especiales

Ambas cosas pueden producir importantes choques en la predicción que pueden afectar a las diferentes series de tiempo. Además, estos choques a menudo no siguen un patrón periódico: ejemplo: Semana santa.

Prophet permite al analista proporcionar una lista de eventos pasados y futuros identificados en un único nombre: *holidays*.

Esto se lleva a cabo de una forma similar a la estacionalidad, generando una matriz de regresores.

$$Z(t) = [\mathbf{I}(t \in D_1), \dots, \mathbf{I}(t \in D_L)] \quad (22)$$

Y cogiendo

$$h(t) = Z(t)\kappa \quad (23)$$

$$\kappa \sim \text{Normal}(0, v^2). \quad (24)$$

A veces es importante incluir el efecto para un intervalo de días alrededor de las vacaciones particulares. Por eso, Prophet ofrece la posibilidad de fijar una ventana (intervalo) superior e inferior.

Una vez hecho una explicación introductoria de cómo funciona este nuevo algoritmo de predicción automática, es momento de ponerlo a prueba mediante las tres series estudiadas en el presente trabajo.

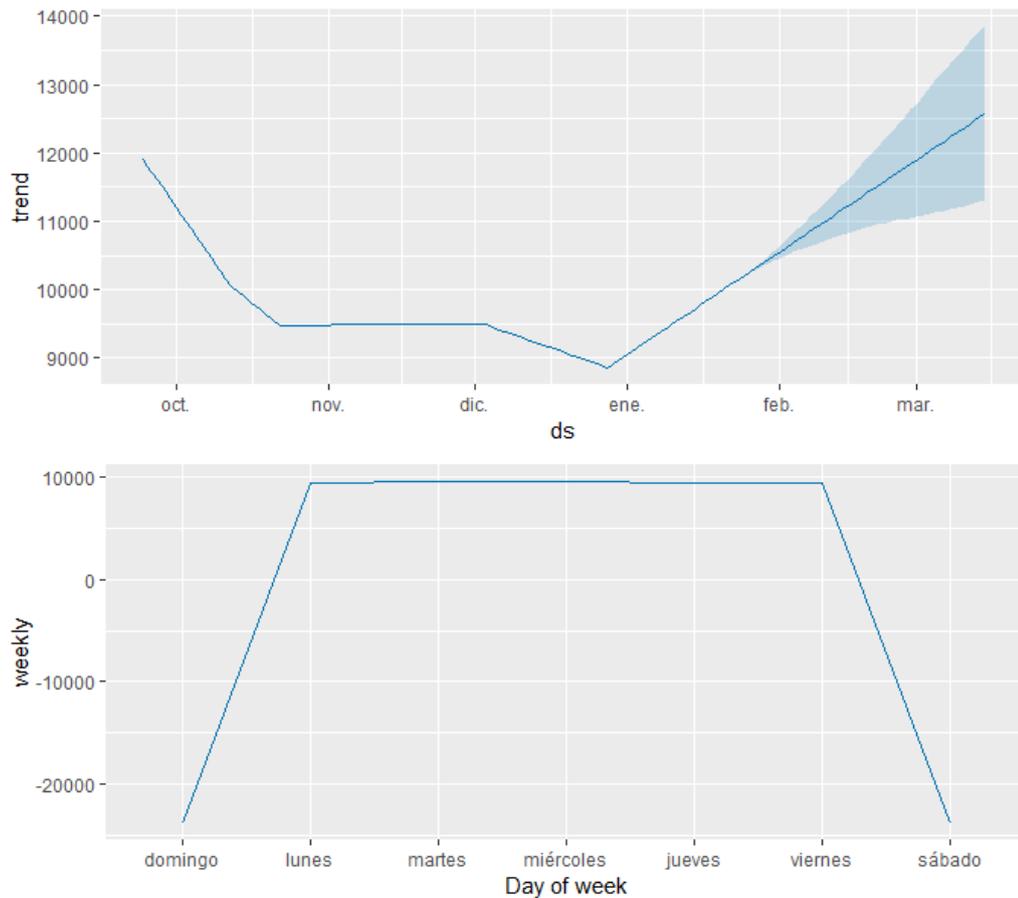
Como es el algoritmo más reciente de todos y en consecuencia el que tiene menos investigaciones en Internet, se va a hacer un análisis más profundo que en las dos funciones anteriores. Los objetivos de esta parte, aparte de comprobar si se cumplen los establecidos en cada serie es estudiar si las innovaciones introducidas por Facebook son de verdad útiles y si las ventajas que sus creadores afirman son de verdad ciertas.

2.3.2.2 Serie diaria

Con la implementación de este algoritmo en la primera serie, se quiere comprobar si de verdad se comporta como los demás algoritmos de Hyndman en los casos de que la tendencia no está claramente definida.

Cabe destacar que como *pre-processing* adicional, se ha tenido que sustituir la función `make_future_dataframe()` del paquete `prophet` por el vector de fechas correspondiente a las serie en el periodo de validación. El motivo es que la función mencionada da por defecto los días posteriores de forma seguida para predecir y, recordemos que esta serie solamente tiene valores presentes en los días laborables.

Gráfico 25: Comportamiento tendencias Prophet Serie diaria



Fuente: Elaboración propia

En cuanto a la predicción, cabe destacar que la tendencia la analiza de forma correcta. Destacar la introducción de los cambios de punto. Como se puede observar, es una gran ventaja respecto a los dos otros modelos ya que aporta una flexibilidad muy importante a la hora de analizar y predecir.

En cuanto a la predicción de la tendencia, parece que acierta al menos al detectar una tendencia alcista. Por lo que concierne al segundo gráfico, no podemos extraer ninguna conclusión útil, ya que uno de los defectos que tiene Prophet es que da por hecho de que hay datos cada día y, en nuestro caso solamente los días laborables.

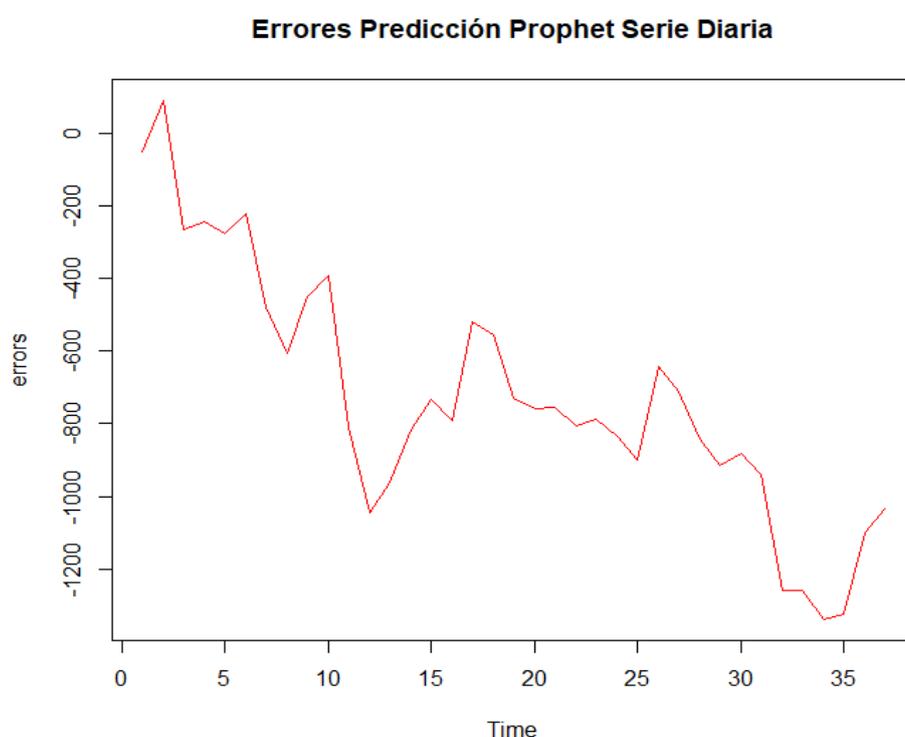
En cuanto a la normalidad de los residuos, al aplicar el Test de Shapiro-Wilk, se puede observar como los residuos del modelo siguen una distribución Normal, ya que el p-valor es de 0.8398 (>0.05).

Seguidamente, se procede a la evaluación tanto analítica, a través de los diferentes indicadores del error, como gráfica.

- El Error Cuadrático Medio es de 645298.1
- El Error Absoluto Medio es de 733.23
- El Error Porcentual Absoluto Medio es del 3.61%.

De momento, en esta primera serie se ha mejora ligeramente la precisión de la predicción, ya que el EPAM ha bajado un 0.2%.

Gráfico 26: Errores de predicción modelo Prophet Serie diaria



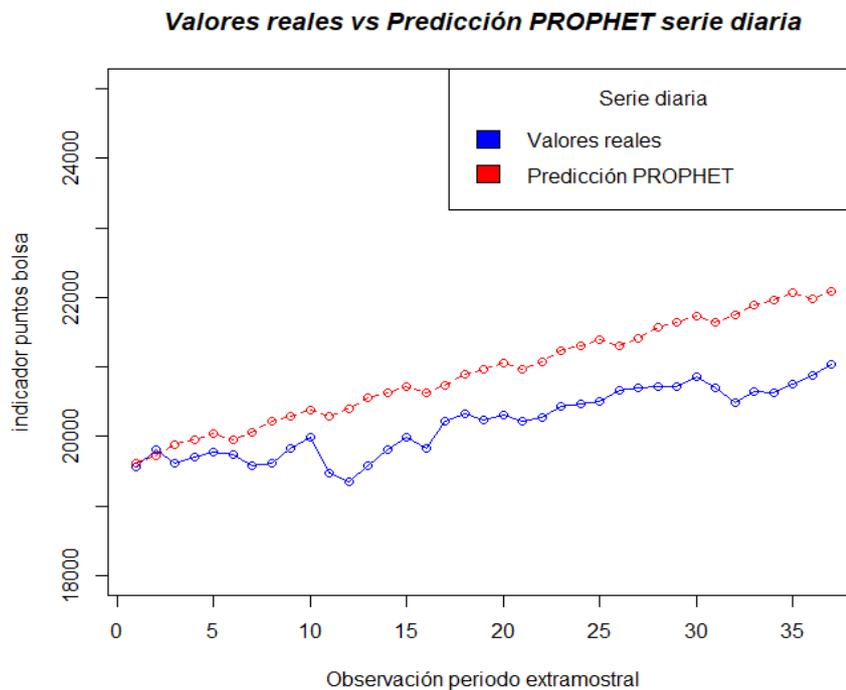
Fuente: Elaboración propia

Como indica este gráfico de errores, parece que Prophet ha sobrevalorado las predicciones, ya que la mayoría de las observaciones están por debajo del 0.

Respecto al contraste de hipótesis para evaluar la Normalidad de los residuos, el Test de Shapiro-Wilk indica que los residuos del periodo extramuestral no siguen una distribución Normal, ya que el p-valor es prácticamente 0.

En el siguiente gráfico se podrá contemplar de una manera más clara:

Gráfico 27: Comparación valores reales con predichos Prophet Serie diaria



Fuente: Elaboración propia

Efectivamente, se lleva cabo una predicción ligeramente superior a los valores reales. Sin embargo, cabe destacar que, a diferencia de los dos otros algoritmos de predicción automática, Prophet sí que ha estimado una tendencia, la cual se podría decir que prácticamente ha clavado. Lo que ha fallado es la escala, es decir, la ordenada al origen de esta predicción.

De esta forma, Prophet ha cumplido las expectativas y los objetivos planteados para la primera serie. No se ha retocado nada en el input y simplemente se ha tenido que cambiar la fecha de la predicción de los valores futuros.

Con lo cual, cabe señalar que en la primera serie ha sido el mejor predictor, el único capaz de parametrizar una tendencia y una estacionalidad, ya que además como se puede observar la serie sigue un patrón de periodo 5, correspondientes a la semana.

También se ha de destacar que Prophet ha cumplido con su propuesta estrella: la automaticidad en implementar el algoritmo.

2.3.2.3 Serie mensual

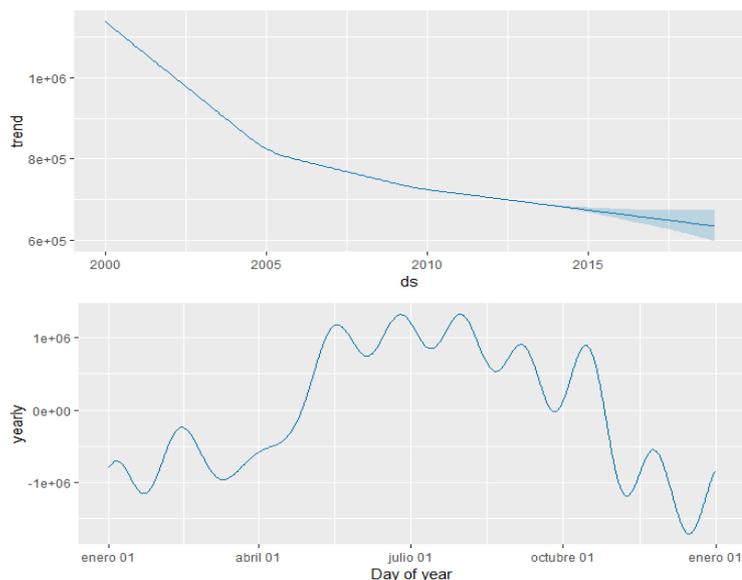
A continuación, se procede a evaluar la implementación de este algoritmo en una serie mensual y con fuerte estacionalidad como es el número de pernoctaciones en las Islas Baleares. Recordemos que el principal objetivo de esta serie es ver si los diferentes modelos de predicción automáticas predicen correctamente la estacionalidad.

En cuanto a la aplicación de Prophet en concreto, se pondrá a prueba su automatización, es decir, en una primera aproximación se supondrá que no se tiene conocimiento alguno sobre la serie y, en segundo lugar, ya se impondrá la estacionalidad mensual.

En la parte del *pre-processing*, no se ha llevado a cabo la misma sustitución que en el anterior caso (función *make_future_dataframe()*). En este caso, simplemente se ha añadido a la función el argumento de que requiere una frecuencia mensual de datos próximos. Recordemos que Prophet, por defecto, trabaja con series diarias (en una excepción pueden ser n-minutales, como se verá en el siguiente apartado).

A continuación, se procede a explicar los resultados de la predicción de Prophet, totalmente automática:

Gráfico 28: Comportamiento tendencia Prophet automático Serie mensual



Fuente: Elaboración propia

Como se puede observar, este algoritmo predice una tendencia global negativa. En cuanto, a la estacionalidad, indica que los hoteles están muchos más concurridos en los meses de verano. Con lo cual, de momento, teniendo en cuenta de que no se le ha informado de ninguna característica concreta de la serie, parece que está llevando a cabo una predicción coherente.

Ahora, es momento de resolver esta duda, es decir, evaluar concretamente la exactitud de la predicción.

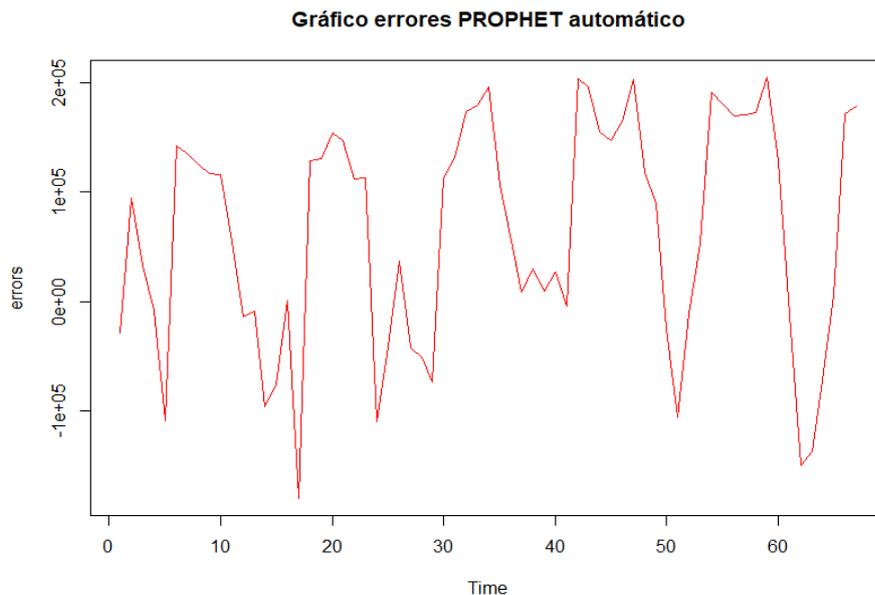
Para ello, como en los anteriores casos, se hará uso de los indicadores que miden los diferentes tipos de error, además del análisis gráfico del error y la comparación gráfica entre la serie del periodo test original y la estimada.

- El Error Cuadrático Medio es 14741567913
- El Error Absoluto Medio es 103593.2
- El Error Porcentual Absoluto Medio es del 241.27%

Es un error altísimo, peor incluso que la predicción ARIMA. Está claro que habrá que hacer unas cuantas modificaciones.

El gráfico de errores es el siguiente:

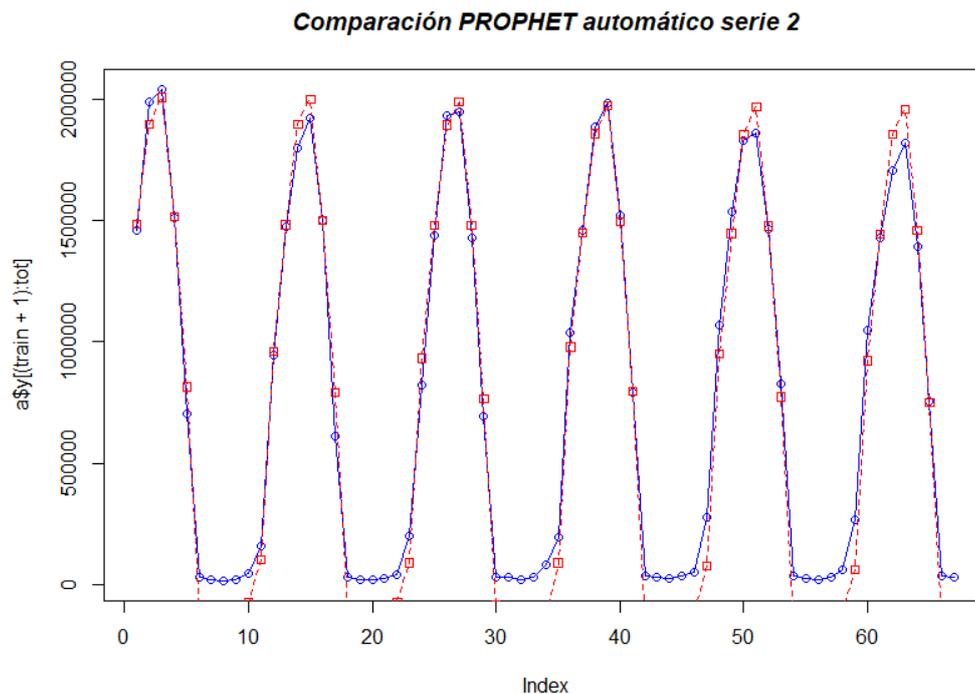
Gráfico 29: Error de predicción Prophet automático Serie mensual



Fuente: Elaboración propia

Como se puede observar, los errores llegan a límites desorbitados. Respecto al contraste de hipótesis para evaluar la Normalidad de los residuos, el Test de Shapiro-Wilk indica que los residuos del periodo extramuestral no siguen una distribución Normal, ya que el p-valor es prácticamente 0.

Gráfico 30: Comparación predicción Prophet automático Serie mensual



Fuente: Elaboración propia

Como se puede observar, Prophet predice, en primer lugar, valores negativos, lo cual es imposible por la naturaleza de la serie. En cuanto a los meses de temporada alta, no parece que lo prediga tan mal.

En cuanto a la normalidad de los residuos, al aplicar el Test de Shapiro-Wilk, se puede observar como los residuos del modelo no siguen una distribución Normal.

Igualmente, se va a intentar paliar todos estos fallos.

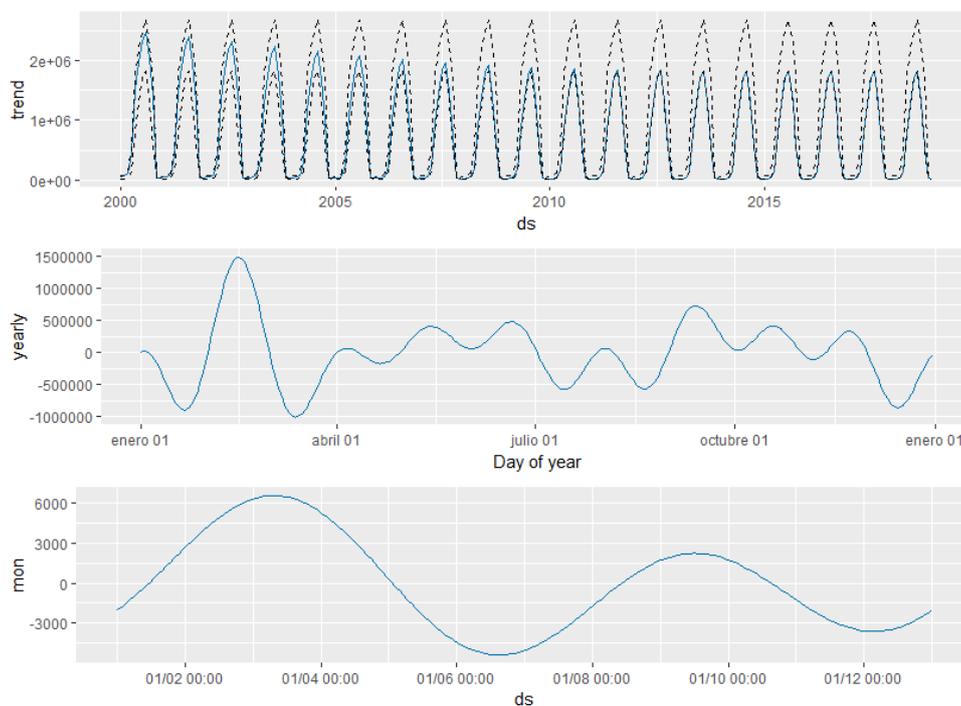
Los cambios a introducir son los siguientes:

1. A través de la función *add_seasonality()* se añade la estacionalidad de forma manual. En este caso de periodo 12. Añadimos como orden de

Fourier un 2, ya que, así como va aumentando, aunque permita mayor flexibilidad a la estacionalidad puede crear sobreajuste.

2. Se añade un tipo de crecimiento logístico. Adicionalmente se añade una capacidad máxima y mínima (valores que al llegar a este mínimo/máximo el crecimiento se satura). El principal objetivo es intentar evitar la predicción de valores negativos. Como capacidad máxima se elige el máximo de cada mes (análogamente para el mínimo).

Gráfico 31: Comportamiento tendencias Prophet modificado Serie mensual



Fuente: Elaboración propia

En este caso, la tendencia global sí tiene en cuenta la estacionalidad anual. Puede observarse los intervalos referentes a las cargas mínima y máxima

El segundo gráfico es confuso, ya que teóricamente está dedicado al día de la semana, al igual que en última es para cada observación en concreto.

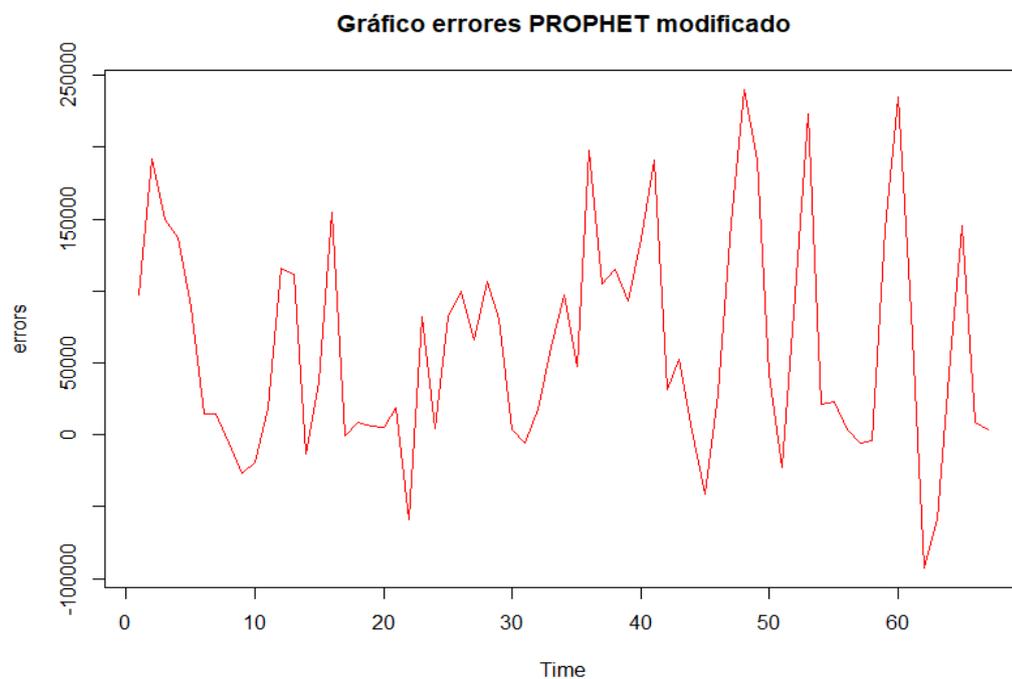
En cuanto a la normalidad de los residuos, al aplicar el Test de Shapiro-Wilk, se puede observar como los residuos del modelo no siguen una distribución Normal.

Se procede a evaluar la capacidad predictiva:

- El Error Cuadrático Medio es de 9620057224
- El Error Absoluto Medio es de 73003.2
- El Error Porcentual Absoluto Medio es de 35.5%

Se ha mejorado significativamente la exactitud de la predicción, como se puede observar si se comparan los dos EPAM's. De todas formas, sigue siendo mucho más válido el modelo ETS.

Gráfico 32: Error de predicción Prophet modificado Serie mensual



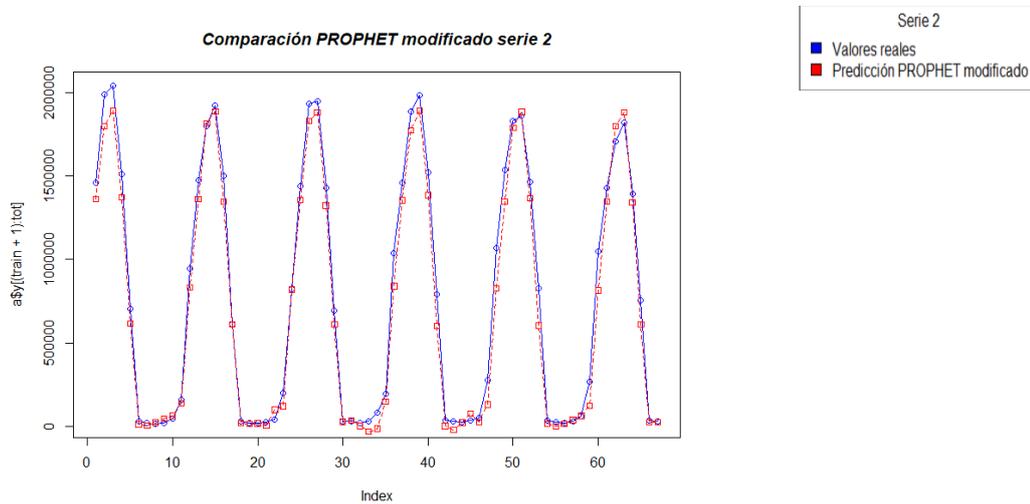
Fuente: Elaboración propia

Como se puede observar en este gráfico, parece que a diferencia de anteriores casos se ha llevado una subestimación de las observaciones en el periodo de validación.

Respecto al contraste de hipótesis para evaluar la Normalidad de los residuos, el Test de Shapiro-Wilk indica que los residuos del periodo extramuestral no siguen una distribución Normal, ya que el p-valor es prácticamente 0.

Ahora, solamente resta mirar en que meses ha fallado más la predicción, ya que se tenía expectativas de poder superar a los dos algoritmos de Hyndman estudiados en el presente trabajo.

Gráfico 33: Comparación predicción Prophet modificado Serie mensual



Fuente: Elaboración propia

A través de este gráfico, se puede ver como se ha conseguido ajustar de forma mucho más exacta la predicción en temporada baja (de finales de otoño a principios de primavera). Sin embargo, esta corrección ha ido en detrimento principalmente de los meses previos y posteriores al verano (mayo, junio, septiembre y octubre).

Se pensó en intentar modelizar estos meses como *holidays*, aunque el modelo resultante tenía una capacidad predictiva peor.

De esta manera, se puede concluir en este apartado que Prophet a veces no es tan automático como se vende, es decir, hay que tener conocimientos específicos tanto del mundo de las Series Temporales como del funcionamiento del algoritmo en cuestión.

Además, parece que en series donde la estacionalidad es muy presente además de definida de manera muy clara, parece ser más confiable otros algoritmos de predicción automática como ETS. Sin embargo, también se ha probado que la función *auto.arima* es la que predecía peor.

2.3.2.4 Serie cinco-minutal

Para concluir, el apartado de evaluación del algoritmo Prophet se utilizará la tercera serie, referente al número de bicicletas *Bicing* en uso.

Visto los resultados de los algoritmos de predicción automática anteriores aplicados a esta serie, se fija el objetivo concreto de intentar hacer una predicción más allá de una línea recta sin pendiente.

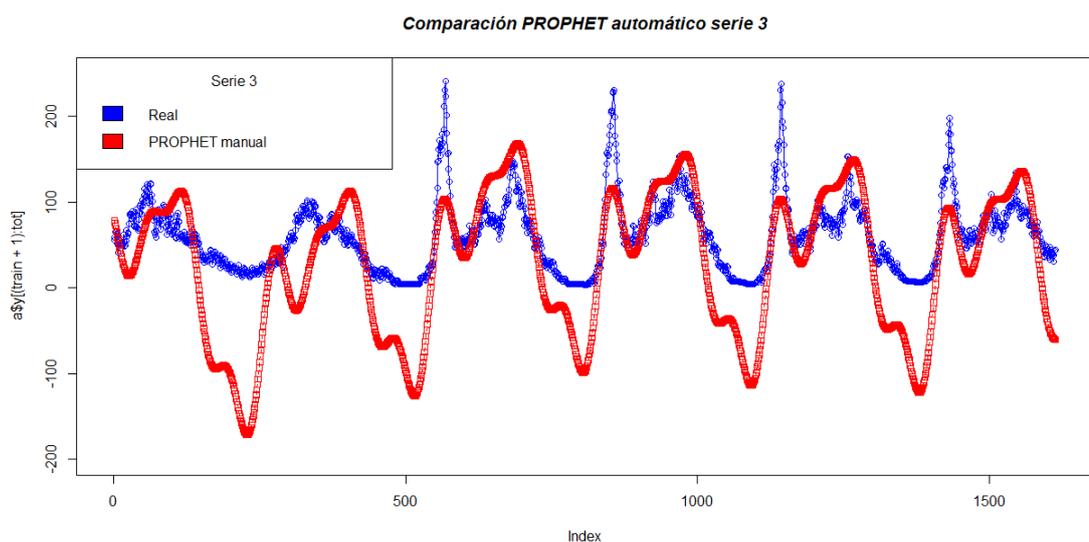
Igualmente, se cree oportuno volver a comparar la implementación de la función Prophet de manera automático y haciendo unos cuantos retoques.

Cabe nombrar que Prophet por defecto trata con datos de fechas diarias, pero también puede tratar con fechas cinco-minutales siempre y cuando estén con el formato YYYY-MM-DD HH:MM:SS. De esta forma, se ha creído interesante probar como se puede comportar una serie de intervalos intertemporales tan pequeños.

En este caso, como parece evidente de que no va a ser una buena predicción, se va a resumir los principales resultados de la implementación de forma automática.

- El Error Cuadrático Medio es de 4067.677
- El Error Absoluto Medio es de 50.42115
- El Error Porcentual Absoluto Medio es del 308.77%

Gráfico 34: Comparación predicción Prophet automático Serie cinco-minutal



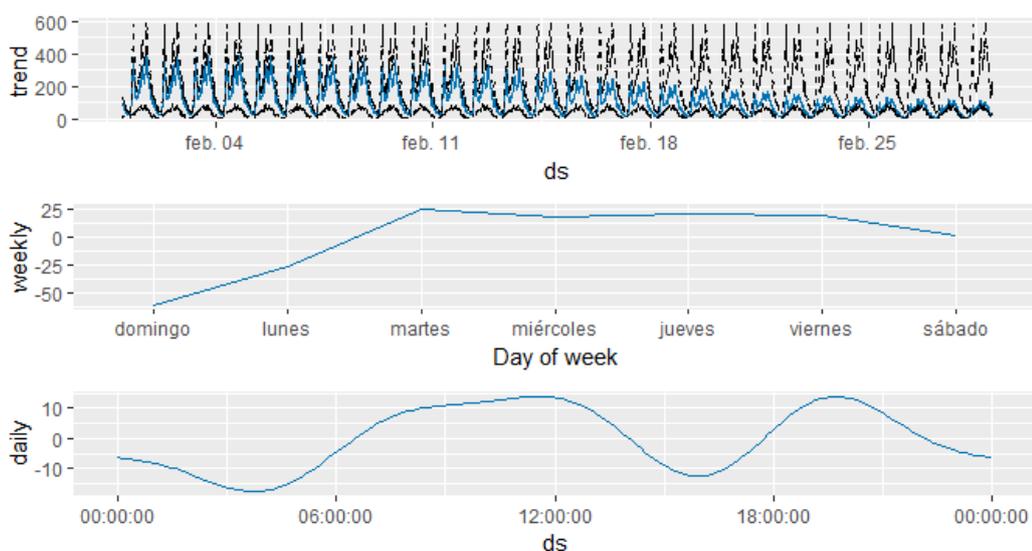
Fuente: Elaboración propia

Se puede observar como predice de forma correcta que hay estacionalidad, pero subestima muchísimo todas las observaciones. Por una parte, para las horas punta predice unos valores mucho menores y para las horas menos concurridas, valores negativos (lo cual es imposible).

De esta forma, se procede a añadir dos tipos de estacionalidad a la vista del gráfico de la serie inicial. Una de periodo 288 (diaria) y otra de periodo 2016 (de periodo semanal). La última se cree conveniente, ya que en el gráfico se ve claramente como cada semana actúa con una evolución muy similar.

En cuanto al crecimiento, se impone que es logístico y como en la serie anterior, a través de una función creada exclusivamente para esta finalidad, se impone una capacidad mínima de cada observación igual al valor mínimo de la hora y el minuto de la observación en cuestión. En el caso del máximo, se desarrolla de forma análoga.

Gráfico 35: Comportamiento tendencia Prophet modificado Serie cinco-minutal



Fuente: Elaboración propia

El siguiente gráfico muestra las diferentes líneas de tendencia. La primera es la global. Como es de suponer sigue la misma evolución que el gráfico en cuestión.

En cuanto a la segunda, enseña la evolución semanal. Afirma que el número de bicis en Domingo suele ser negativo, la cual cosa es imposible con lo que se puede concluir que no ajusta de manera coherente.

Sin embargo, en el último gráfico, sí que señala que las horas más concurridas son por la mañana y a media tarde, la cual cosa coincide con los valores reales de la serie temporal.

En cuanto a la normalidad de los residuos, al aplicar el Test de Kolmogorov-Smirnov, se puede observar como los residuos del modelo no siguen una distribución Normal.

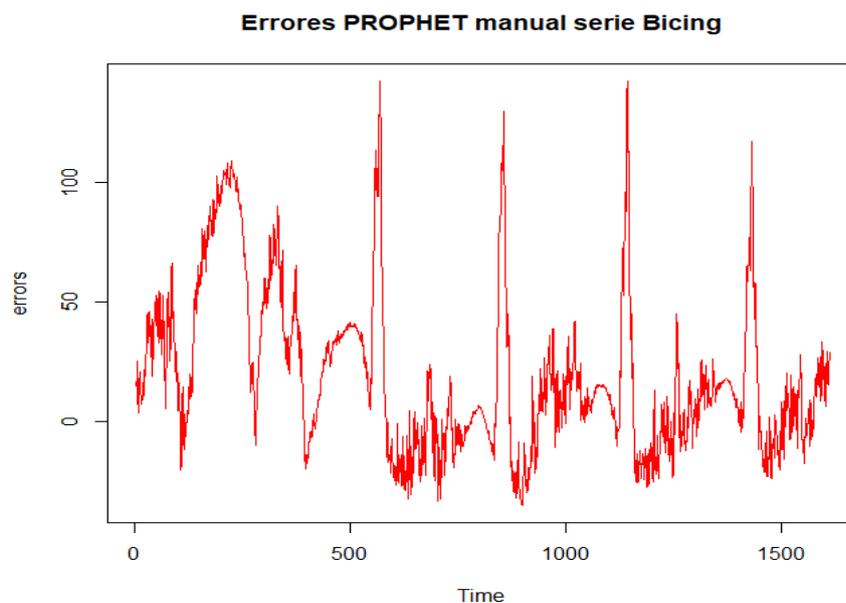
A continuación, se procede a evaluar la capacidad predictiva de este modelo:

- El Error Cuadrático Medio es de 1501.584
- El Error Absoluto Medio es de 27.206
- El Error Porcentual Absoluto Medio es del 97.16%

A través de estos indicadores, se puede concluir que la predicción no es buena. Sin embargo, sí que ha habido una mejora sustancial respecto tanto del algoritmo Prophet totalmente automático como de los demás algoritmos de predicción automática de Hyndman.

De todas formas, cabe analizar donde ha fallado principalmente la predicción. Como siempre se hará uso del gráfico de errores y de comparación de valores reales y predichos durante el periodo de validación.

Gráfico 36: Errores predicción modelo Prophet modificado Serie cinco-minutal

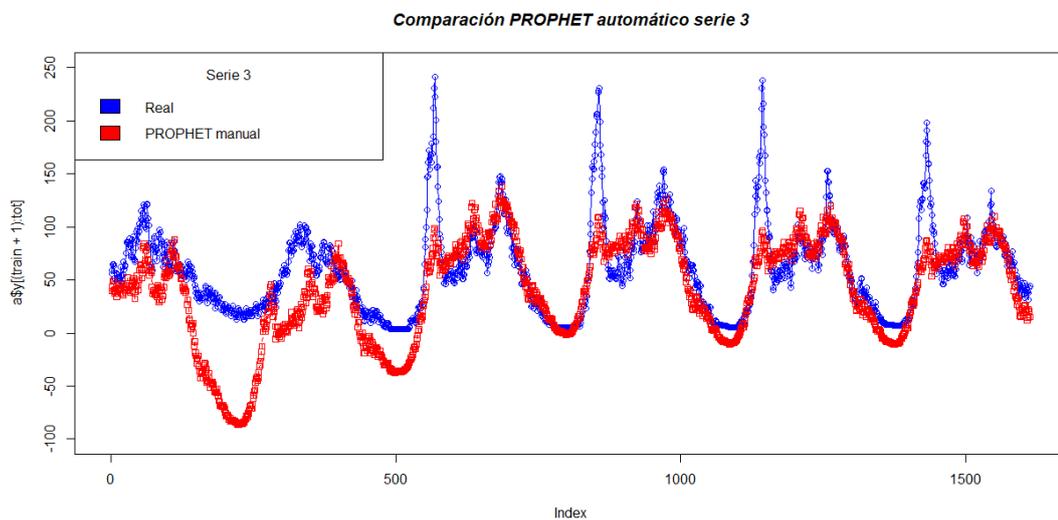


Fuente: Elaboración propia

Respecto al contraste de hipótesis para evaluar la Normalidad de los residuos, el Test de Shapiro-Wilk indica que los residuos del periodo extramuestral no siguen una distribución Normal, ya que el p-valor es prácticamente 0.

Hay principalmente unas pocas observaciones, representadas en cuatro días donde la predicción dista mucho de la realidad. Con el gráfico de comparación, se podrá ver de manera más clara.

Gráfico 37:: Comparación predicción Prophet modificado Serie cinco-minutal



Fuente: Elaboración propia

Se tiene el mismo problema que en el caso anterior: no se predicen bien los valores más bajos y más altos de cada día. Una de las opciones que se barajan para internar paliar este problema es parametrizar este tipo de observaciones a través de las vacaciones.

Cabe señalar que se intentó, pero después de llevar a cabo una investigación exhaustiva donde se examinó cada línea de código de la función Prophet se vio que solamente se pueden implementar vacaciones para datos diarios, ya que en esta función, la línea de código encargada de leer las *holidays* usa la función *as.Date()* para leer el input, lo cual automáticamente le quita la franja horaria y únicamente queda la fecha del día en cuestión.

De esta manera, se puede concluir que uno de los inconvenientes que tiene Prophet es el hecho de no poder introducir *holidays* para datos subdiarios, lo cual parece que podría ser una gran alternativa para intentar bajar el error de predicción.

2.3.4. Modelos NNAR

Como se ha anticipado en el capítulo 1 y en la introducción de este apartado, se pretende experimentar con una alternativa no lineal propuesta por Hyndman dentro del mismo paquete que los modelos ETS y ARIMA como algoritmos de automatic forecasting.

De esta manera, se introducen los modelos NNAR.

2.3.4.1 Enfoque teórico

Como sus siglas indican, este modelo proviene de una Red Neuronal. Las Redes Neuronales Artificiales son métodos de predicción basados en modelos matemáticos simples, estructura de la cual, se asemeja a una conexión entre neuronas humanas. Estos métodos permiten relaciones no lineales complejas entre la variable respuesta y sus predictores.

Existe una gran cantidad de tipos de Redes Neuronales Artificiales. Sin embargo, se procede a explicar únicamente la Autorregresiva, enfocándose en la función *nnetar*²⁰ del paquete *forecast*.

Este modelo parte de los valores retardados que son usados como inputs en un modelo de Autoregresión lineal (AR). Estos valores pueden ser usados como inputs de una red neuronal.

Hyndman solamente considera las redes *feed-forward* con solo una capa oculta.

Para series no estacionales, se usa la notación NNAR(p,k) para indicar que hay p inputs retardados y k nodos en la capa oculta de la red.

Respecto a las series estacionales la función *nnetar()* ajusta un modelo NNAR(p,P,k)[m] donde P es el orden de retardo de la parte estacional.

²⁰ Enlace: <https://pkg.robjhyndman.com/forecast/reference/nnetar.html>

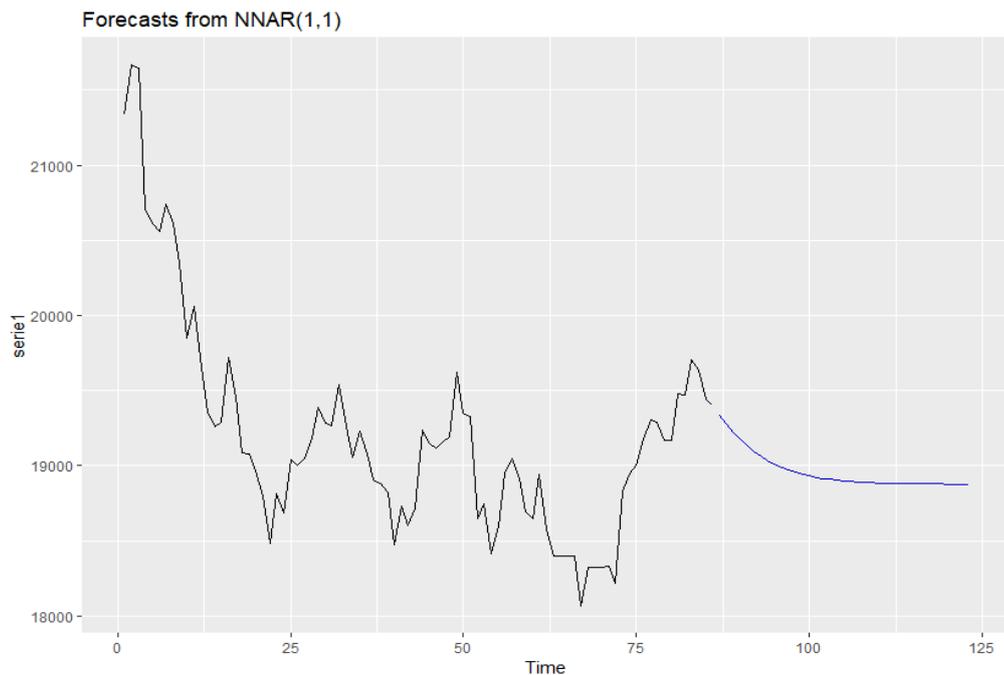
Los valores p , P y k , pueden ser introducidos manualmente o seleccionados automáticamente. Como es un modelo autorregresivo no lineal, y no es posible obtener analíticamente intervalos de predicción, se usa simulación

2.3.4.2. Enfoque empírico

En la serie diaria, el modelo resultante es un NNAR(1,1) es decir, este modelo es un *multilayer feed-forward network*, en donde tiene un solo retardo como variable independiente y un nodo en la única capa oculta. Hay un total de 4 pesos, ya que cada red es 1-1-1.

La σ^2 estimada es 56734.

Gráfico 38: Predicción periodo extramuestral modelo NNAR serie diaria



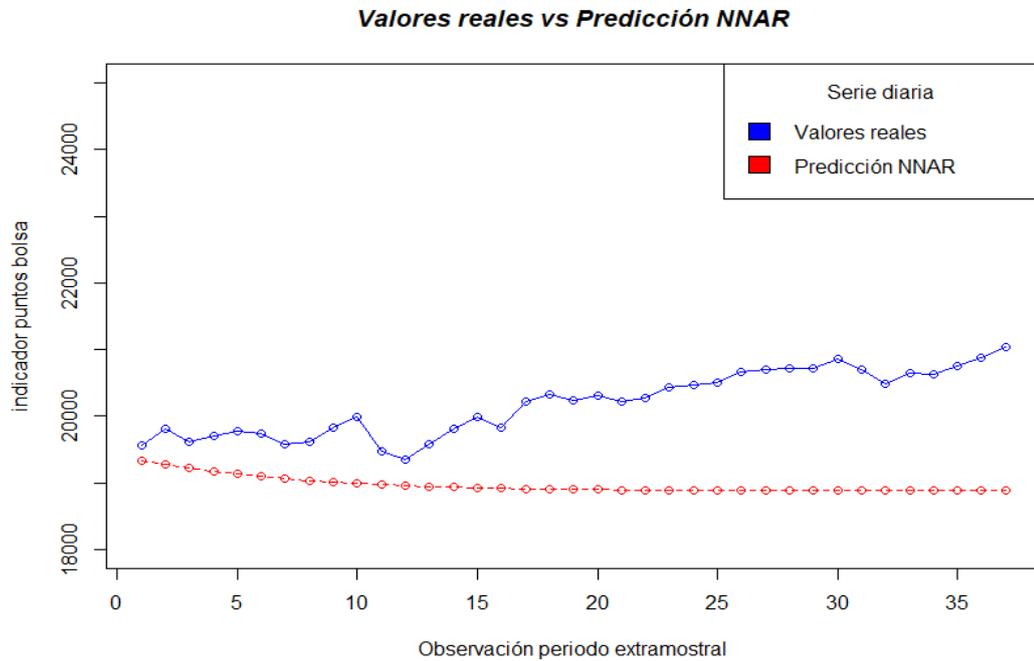
Fuente: Elaboración propia

En cuanto a la normalidad de los residuos, al aplicar el Test de Shapiro-Wilk, se puede observar como los residuos del modelo siguen una distribución Normal, ya que el p-valor es de 0.4245 (>0.05). Respecto al periodo extramuestral, el Test de Shapiro-Wilk indica que los residuos del periodo extramuestral no siguen una distribución Normal, ya que el p-valor es prácticamente 0.

- El Error Cuadrático Medio es de 1822371
- El Error Absoluto Medio es de 1227,48

- El Error Porcentual Absoluto Medio es de 6,02%

Gráfico 39: Comparación predicción NNAR Serie diaria



Fuente: Elaboración propia

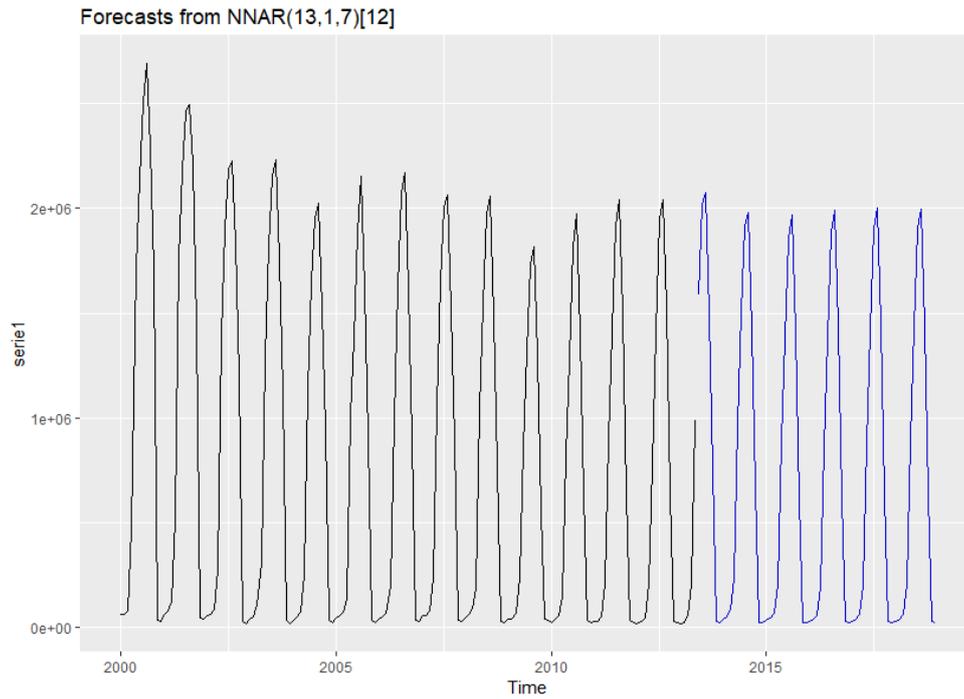
En la serie mensual, tampoco mejora los resultados de ETS. Los principales resultados a resaltar son:

- El modelo ajustado es un NNAR(13,1,7)[12]²¹
- Cada red es una 13-7-1 con 106 pesos
- La σ^2 es 488149460.

En cuanto a la normalidad de los residuos, al aplicar el Test de Shapiro-Wilk, se puede observar como los residuos del modelo no siguen una distribución Normal. Respecto al periodo extramuestral, el Test de Shapiro-Wilk indica que los residuos del periodo extramuestral no siguen una distribución Normal, ya que el p-valor es prácticamente 0.

²¹ Este modelo es un *multilayer feed-forward network*, en dónde tiene 13 retardos como variables independientes y uno estacional. Además, contiene una capa oculta de 7 nodos.

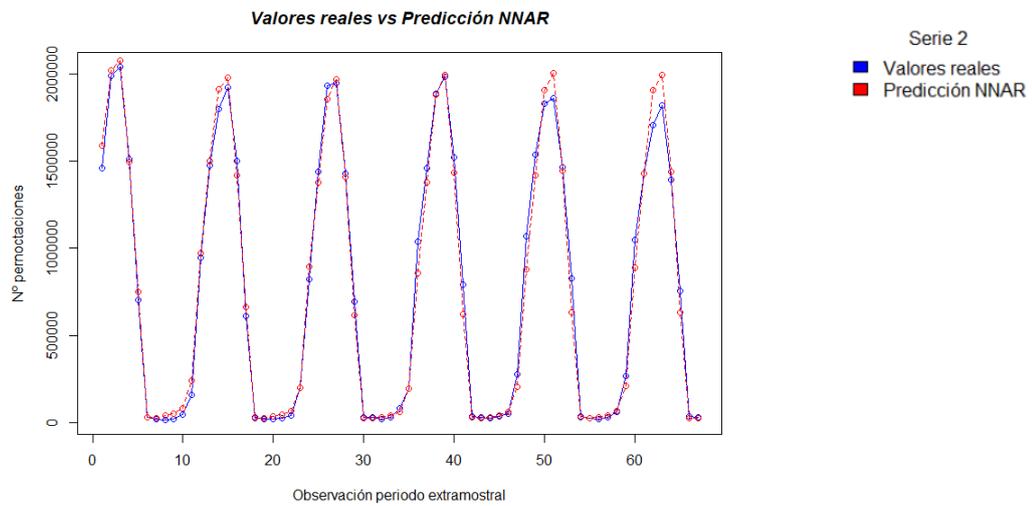
Gráfico 40: Predicción periodo extramuestral NNAR Serie Mensual



Fuente: Elaboración propia

- El Error Cuadrático Medio es de 5902573667
- El Error Absoluto Medio es de 51221,79
- El Error Porcentual Absoluto Medio es de 22,14%

Gráfico 41: Comparación predicción modelo NNAR Serie Mensual



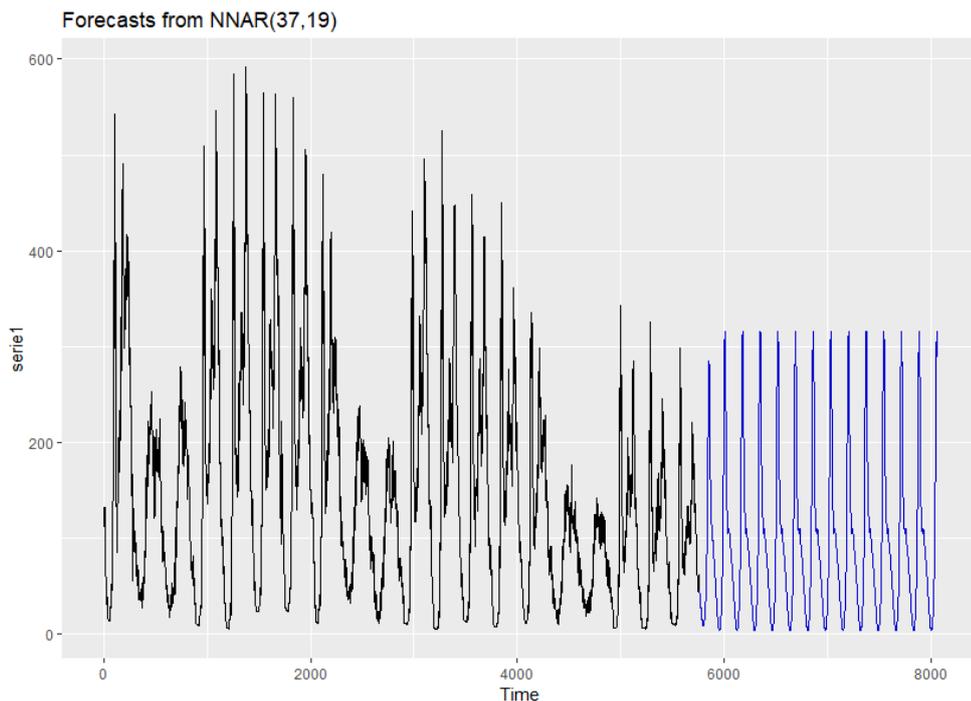
Fuente: Elaboración propia

Para acabar, se presentan las principales características del modelo resultante en la serie cinco-minutal:

En una primera aproximación, se supone que la serie no presenta estacionalidad, es decir, sin poner ningún tipo de frecuencia.

Los principales resultados son los siguientes:

Gráfico 42: Predicción periodo extramuestral NNAR sin frecuencia



Fuente: Elaboración propia

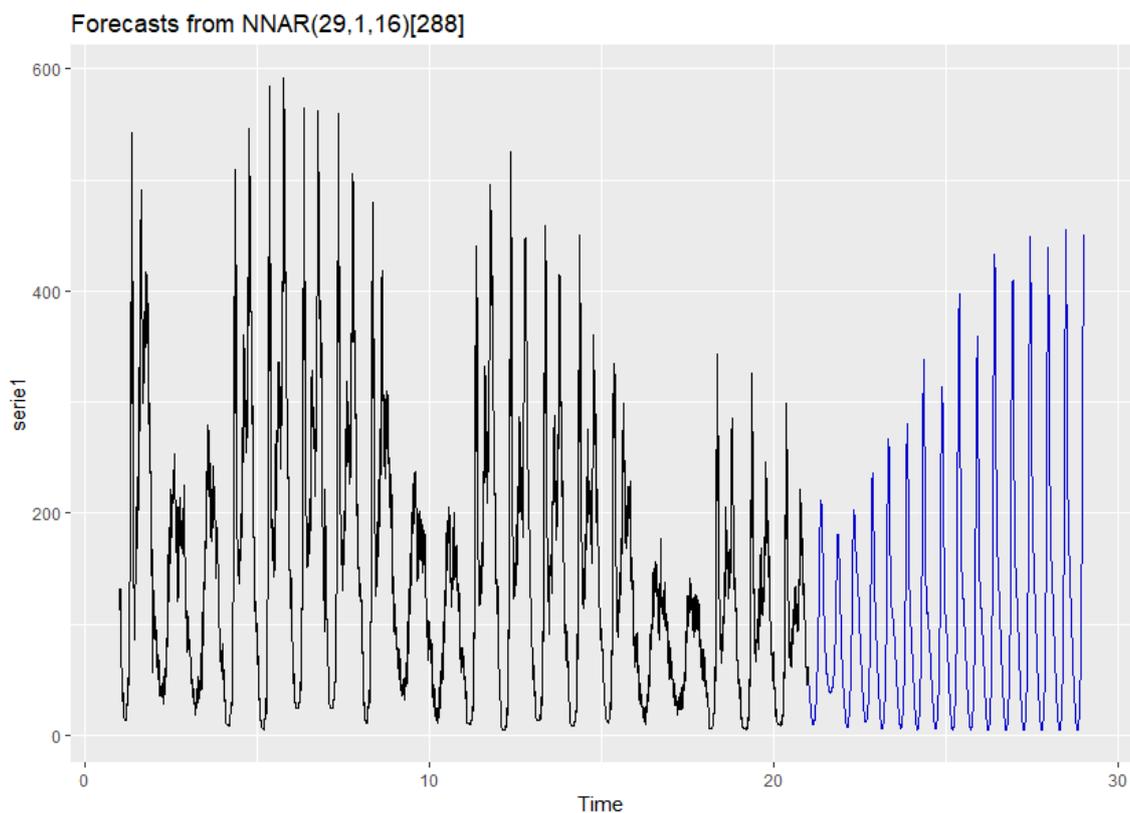
Como se puede observar, el modelo escogido es un NNAR(37,19)²². La predicción parece no ser buena. Más concretamente:

- El Error Cuadrático Medio es de 9842,79
- El Error Absoluto Medio es de 73,24
- El Error Porcentual Absoluto Medio es del 281,93%

De esta forma, se intenta ejecutar la función imponiendo una frecuencia diaria, es decir, de periodo 288. La predicción se puede observar en el siguiente gráfico:

²² Este modelo es *un multilayer feed-forward network*, en dónde tiene 37 retardos como variable independiente y 19 nodos en la primera capa oculta.

Gráfico 43: Predicción periodo extramuestral NNAR frecuencia diaria



Fuente: Elaboración propia

A primera vista, parece que la predicción es incluso peor que en el anterior caso:

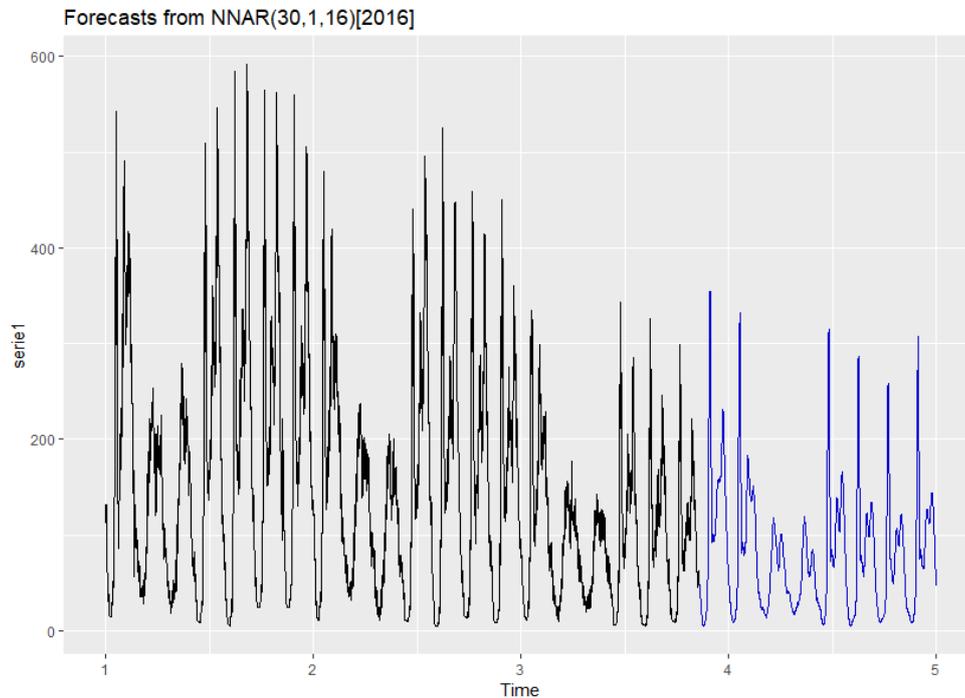
- El Error Cuadrático Medio es de 17532.8
- El Error Absoluto Medio es de 91,82452
- El Error Porcentual Absoluto Medio es de 294,06%

Aunque en este caso, la predicción sea menos precisa, se puede extraer una conclusión positiva: la función *nnetar()* puede cargar con series de presencia estacional de alto periodo.

Con lo cual, como último intento se propone aplicar este algoritmo de predicción automática suponiendo que la serie cinco-minutal tiene estacionalidad semanal, es decir, de periodo 2016 (288 x 7).

De esta forma, la predicción en el periodo extramuestral se presenta en el siguiente gráfico:

Gráfico 44: Predicción modelo NNAR Serie cinco-minutal frecuencia semanal



Fuente: Elaboración propia

A primera vista, la predicción tiene muy buena pinta, ya que la forma de la estacionalidad es muy similar. Antes de entrar a evaluar la precisión de forma analítica, se procede a describir que tipo de modelo es el elegido en este caso:

Es un NNAR (30,1,16)[2016]²³. Cada una de las redes es 31-16-1, con lo cual ya se puede considerar una red bastante compleja. Esto significa que tiene 31 retardos como variable independiente y en la capa oculta de cada red hay un total de 16 nodos.

El total de pesos es 529 y la σ^2 es 83.67

En cuanto a la normalidad de los residuos, al aplicar el Test de Kolmogorov-Smirnov, se puede observar como los residuos del modelo no siguen una distribución Normal. Respecto al periodo extramuestral, el Test de Shapiro-Wilk indica que los residuos del periodo extramuestral no siguen una distribución Normal, ya que el p-valor es prácticamente 0.

²³ Este modelo es un *multilayer feed-forward network*, en dónde tiene 30 retardos como variable independiente, 1 retardo en la parte estacional y 16 nodos en la capa oculta. La estacionalidad es de periodo 2016.

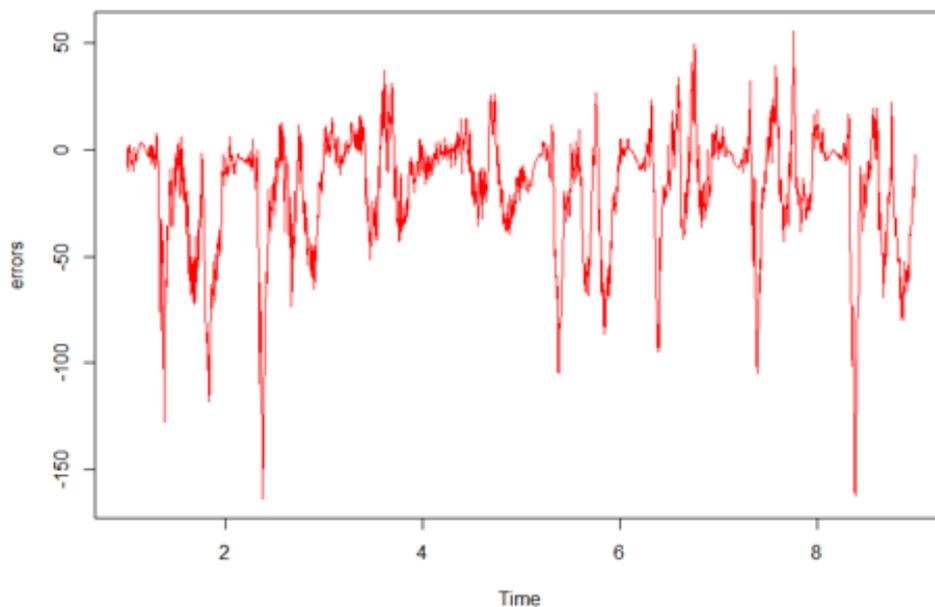
Seguidamente se presentan los siguientes resultados de los indicadores del error:

- El Error Cuadrático Medio es de 1050.5
- El Error Absoluto Medio es de 21.78
- El Error Porcentual Absoluto Medio es de 41.2%

Efectivamente, con diferencia es el algoritmo de *automatic forecasting* estudiado empíricamente en este trabajo que ha conseguido predecir con mayor exactitud la serie más compleja de todas.

A continuación, se presenta el gráfico de errores del periodo extramuestral:

Gráfico 45: Errores de predicción modelo NNAR Serie cinco-minutal

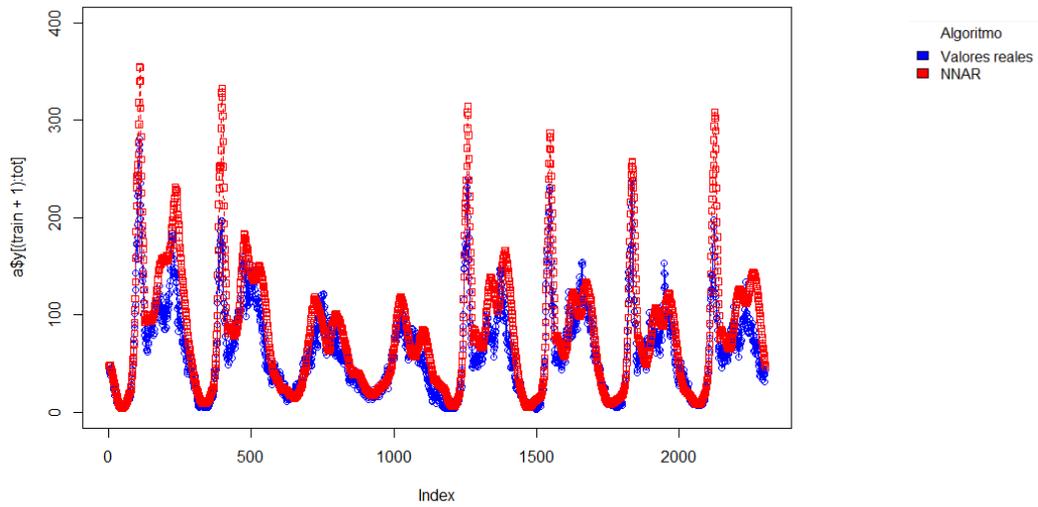


Fuente: Elaboración propia

Como se puede observar, en general parece que predice bien, ya que en la mayoría de observaciones los errores están muy cercanos al 0. Sin embargo, hay una serie de observaciones en las cuales el error es negativo y posee un valor muy alto. Con lo cual, el valor predicho es mucho más alto.

Para concluir este subapartado, se presenta gráficamente la comparación de la predicción en el periodo extramuestral:

Gráfico 46: Comparación predicción NNAR Serie Cinco-minutal



Fuente: Elaboración propia

Efectivamente en los momentos más concurridos de la semana, la red neuronal autorregresiva predice un valor mucho mayor. De todas maneras, cabe señalar que no hay valores negativos y que, durante los periodos menos concurridos de la semana, la estimación parece muy precisa.

2.4 Comparativa

En este apartado, únicamente se expondrán los resultados comparados de todos los algoritmos en cada serie.

Por una parte, se llevará a cabo una exposición analítica a través de la comparación de los EPAM de cada algoritmo. Cabe recordar que el EPAM es el único indicador de exactitud adimensional que se ha incluido en este trabajo, con lo que es el elegido para así poder comparar predicciones no solo intra-series, sino también entre series aplicando el mismo algoritmo.

Por otra parte, es de vital importancia incluir una representación gráfica donde para cada serie, hay representados la predicción de los cuatro algoritmos de automatic forecasting a la vez en el periodo extramuestral.

Así pues, se procede a presentar las siguientes tablas con los errores absolutos de predicción porcentuales:

Tabla 3: Comparación EPAM Serie Diaria

| ALGORITMO | EPAM (%) |
|------------------|-----------------|
| ETS | 3.84 |
| ARIMA | 3.86 |
| PROPHET | 3.61 |
| NNAR | 6.02 |

Fuente: Elaboración propia

Los tres primeros algoritmos tienen un EPAM aceptable y muy similar. La excepción es el modelo NNAR, aunque no alcanza un valor tampoco demasiado alto. Si la valoración global se centrara únicamente en este indicador, sería muy difícil extraer alguna conclusión.

Tabla 4: Comparación EPAM Serie Mensual

| ALGORITMO | EPAM (%) |
|-----------------------------|-----------------|
| ETS | 12.14 |
| ARIMA | 200.34 |
| PROPHET (AUTOMÁTICO) | 241.27 |
| PROPHET (MODIFICADO) | 35.5 |
| NNAR | 22.14 |

Fuente: Elaboración propia

En este caso, se observa más diferencia. Concretamente, el modelo ETS tiene un EPAM mucho menor al resto. Cabe destacar, a la vez, la enorme brecha entre el Prophet automático y el algoritmo con la inclusión de una serie de características antes mencionadas. Además, el modelo ARIMA seleccionado tiene una precisión muy baja.

Tabla 5: Comparación EPAM Serie Cinco-minutal

| ALGORITMO | EPAM (%) |
|-----------------------------|-----------------|
| ETS | 151.19 |
| ARIMA | 150.56 |
| PROPHET (AUTOMÁTICO) | 308.77 |
| PROPHET (MODIFICADO) | 97.16 |
| NNAR | 41.2 |

Fuente: Elaboración propia

En este último caso, se observan unos errores de predicción muy altos en los cuatro algoritmos. De todas formas, el modelo NNAR es el algoritmo mejor valorado con muchísima diferencia si solamente se tiene en cuenta este indicador.

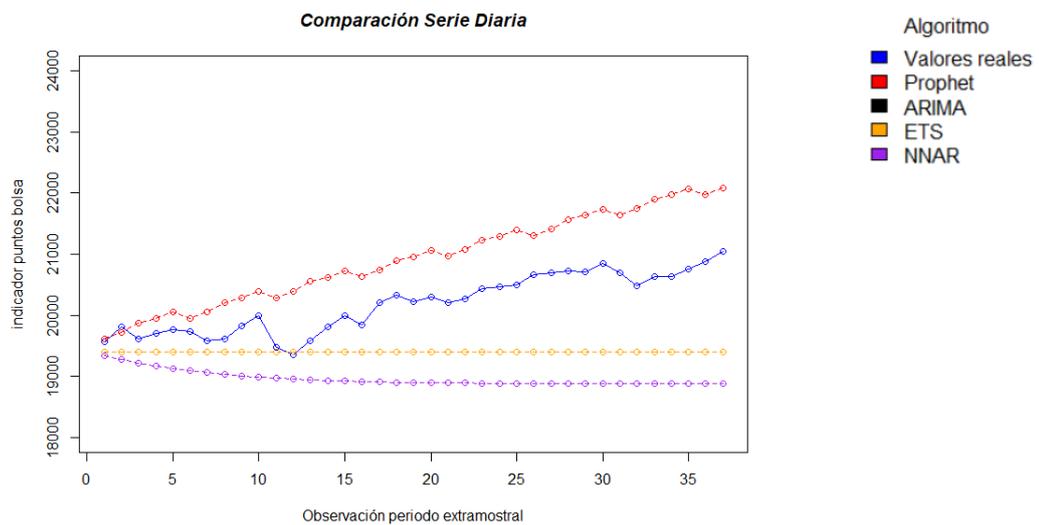
Tabla 6: Comparación entre series

| <i>SERIE</i> | ALGORITMO | EPAM (%) |
|----------------------|------------------|-----------------|
| <i>Diaria</i> | Prophet | 3.84 |
| <i>Mensual</i> | ETS | 12.14 |
| <i>Cinco-minutal</i> | NNAR | 41.2 |

Fuente: Elaboración propia

Para terminar la comparación descriptiva, es útil aprovechar la adimensionalidad del EPAM para comparar la precisión en cada serie. Como se puede observar, como más alta es la complejidad de la serie, mayor EPAM y, por lo tanto, menor precisión en la predicción.

Gráfico 47: Comparación Algoritmos Serie Diaria

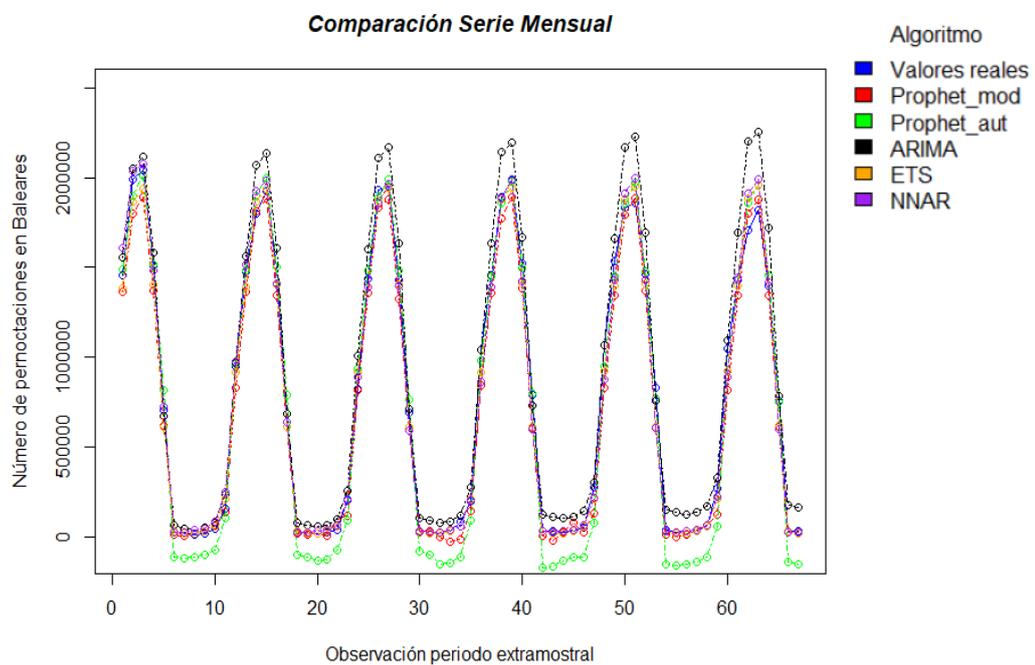


Fuente: Elaboración propia

En primer lugar, cabe señalar que, aunque no se pueda visualizar de forma correcta, la predicción ARIMA (en negro) y la predicción ETS (en amarillo) están solapadas. Esto significa que, prácticamente tiene el mismo valor en cada observación del periodo extramuestral.

En cuanto a Prophet, parece que sigue una tendencia similar a la de la serie original. Sin embargo, le falla la ordenada al origen. El peor algoritmo en este sentido es NNAR que predice una pendiente negativa cuando en realidad es positiva.

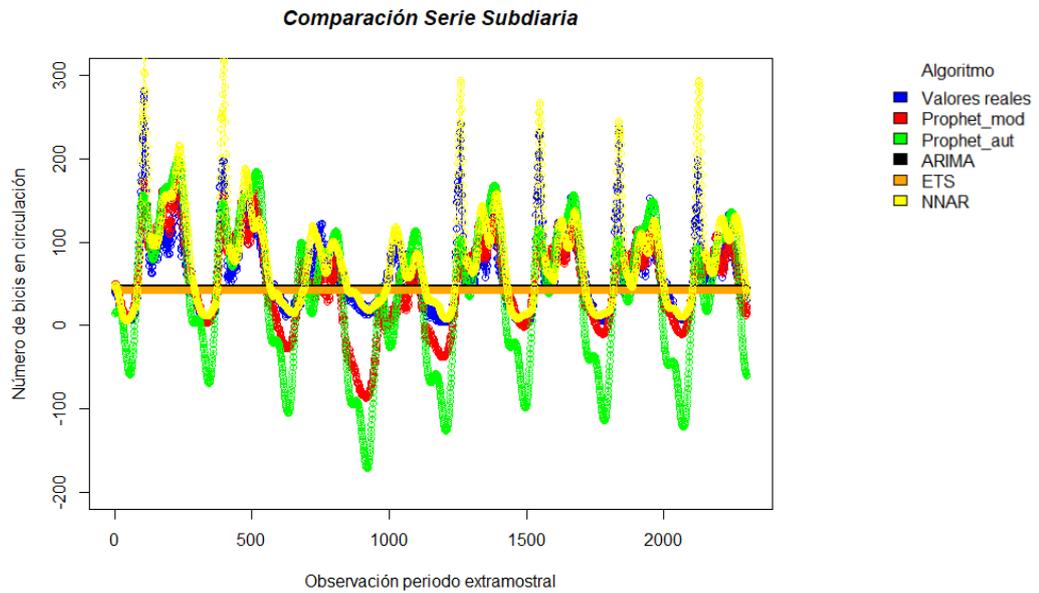
Gráfico 48: Comparación Algoritmos Serie Mensual



Fuente: Elaboración propia

Aunque sea un poco difícil de visualizar por ya la importante cantidad de observaciones de la parte test, así como la inclusión de seis funciones, sí que se puede observar que todos siguen una evolución muy similar y muy ligada a la serie original. Sin embargo, destacar el descuelgue en la temporada baja del algoritmo Prophet automático, así como el descuelgue en temporada alta del modelo ARIMA.

Gráfico 49: Comparación algoritmos Serie Cinco-minutal



Fuente: Elaboración propia

En esta predicción, cabe señalar que ARIMA y ETS vuelven a actuar de forma muy similar al primer caso (serie diaria). En cuanto a Prophet, se puede observar como el Prophet automático alcanza valores mínimos muy bajos. El mejor modelo es claramente el NNAR, ya que sigue muy de cerca la evolución de los valores reales, sobre todo en las observaciones de valor más bajo.

CONCLUSIONES

La predicción siempre ha sido uno de los temas más importantes en cualquier sector, ya que poder adivinar qué pasará en el futuro te puede proporcionar grandes ventajas. Más concretamente, los diferentes modelos de *automatic forecasting* siempre han sido muy seguidos de cerca por economistas y estadísticos para poder aplicarlos.

Recientemente, gracias al desarrollo del Big Data se pueden tratar un volumen de datos mucho mayor de lo que podríamos imaginar hasta hace poco. De todas formas, es muy importante desarrollar unas técnicas idóneas para poder traducir todo este volumen de datos en información útil para analizar y posteriormente predecir. Asimismo, es de vital importancia poder tratar con algoritmos de predicción automática capaces de poder modelar correctamente diferentes datos de evoluciones y relaciones complejas.

Por una parte, según se ha podido investigar, las diferentes técnicas de Minería de datos, aunque en un principio no fueran creadas para tal fin, parecen ser una buena alternativa a los modelos más clásicos; por otra, la introducción también de algoritmos más personalizados como Prophet dan un abanico mucho más amplio de posibilidades para el mundo del *automatic forecasting*.

Como es de suponer, estas técnicas tienen aplicaciones en múltiples sectores. Uno de ellos es el campo de la Macroeconomía, donde es crucial para la planificación de políticas y para la detección de crisis económicas, como antes se ha nombrado.

De todas formas, esta evolución puede dar lugar a otra revolución tecnológica, donde el capital humano sea sustituido prácticamente por la máquina. Ahora mismo, curiosamente se prevé que la Inteligencia Artificial, en un futuro cercano, llegue a dimensiones tales que la mayoría de trabajos actuales sean fulminados. En consecuencia, el mercado de trabajo y, por lo tanto, el sistema económico darán un cambio radical. Está por ver que como se reaccionará ante tal revolución.

Respecto a la parte empírica, así como se iban presentando tanto las diferentes series como los diversos algoritmos también se han ido planteando los distintos objetivos de la parte práctica del presente trabajo. Ahora, es momento de recordarlos y deliberar sobre si se han cumplido las expectativas generadas y, sobre todo, qué diferencias hay entre estos tres algoritmos de predicción automática.

En primer lugar, cabe señalar que, en series de tendencia no permanente, es decir, una tendencia que no presenta un crecimiento monótono, es mejor utilizar la

función Prophet. Este argumento se sostiene a través del experimento llevado a cabo con la serie referente al Índice de Activos Bursátiles de la Bolsa Italiana.

Por una parte, la función *auto.arima* y *ets* han sido incapaces de poder ajustar un modelo acorde con sus características que fuera más allá de una proyección de tendencia nula. Con el modelo NNAR, se ha predicho una pendiente negativa cuando en realidad es positiva. Cabe recordar que, en esta serie, hay cambios de tendencia bruscos, que dificultan el poder detectar un patrón a simple vista. Sin embargo, Prophet sí ha sido capaz de parametrizar una tendencia. Además, la pendiente la ha estimado de forma muy precisa, aunque ha fallado en la ordenada en el origen.

Cabe señalar que de la serie diaria se puede extraer otra lección: es muy importante combinar el análisis descriptivo con el gráfico. A la vista de los indicadores de exactitud de la predicción, los valores son muy similares, con lo que esto llevaría a pensar que los tres algoritmos con menor EPAM actúan de forma similar. Sin embargo, a la vista del análisis gráfico se puede observar que Prophet predice mucho mejor y que si ETS y ARIMA tienen un error similar a Prophet es porque la serie tiene pocas observaciones y, sobre todo porque ésta es muy poco volátil.

Otro de los objetivos planteados era el de comprobar como actuaban en series de estacionalidad claramente definida. En este caso, parece que los algoritmos del paquete *forecast* predicen con mayor precisión, con la excepción de ARIMA, que en nuestro caso empírico no ha predicho de forma tan exacta como el resto.

Para sostener esta tesis, se han probado los cuatro algoritmos con la serie mensual de muestra. El modelo más exacto es el ETS, con una diferencia bastante holgada respecto a Prophet (tercer clasificado según el criterio del EPAM, por detrás de la Red Neuronal Autorregresiva).

En cuanto a la última serie, de frecuencia cinco-minutal, cabe señalar que solamente el modelo resultante NNAR ha realizado una predicción aceptable. Eso sí, se ha tenido que suponer que la serie tenía una frecuencia de periodo 2016. No obstante, sí que a partir de este análisis de la serie cinco-minutal se han podido extraer una serie de conclusiones:

1. A partir de un gran número de datos, funcionan mucho mejor los algoritmos Prophet y NNAR, ya que pueden soportar mayor coste computacional.

2. Para series de estacionalidad de periodo alto, la función ets y auto.arima no pueden hacer frente a tal elevado coste computacional. De hecho, auto.arima solamente permite trabajar con series de frecuencia estacional de hasta periodo 24.
3. Al igual que en la serie diaria, si no hay una tendencia lineal claramente definida y, sobre todo, monótona, el modelo ajusta de ETS y ARIMA es de pendiente nula, por lo menos en los dos casos estudiados.

También, cabe hacer un paréntesis y tratar la automatización de Prophet. En su presentación oral y en su documento de presentación, antes nombrado, se muestra este algoritmo de *automatic forecasting* como una función realmente automática, en el sentido de que no hay que tener a penas conocimientos previos sobre Series Temporales. Sin embargo, en nuestro caso se puede concluir que creemos que esta afirmación no es del todo cierta. En la parte empírica se ha encontrado con dos tipos de series en las cuales ha sido necesario realizar una serie de modificaciones:

En la serie mensual, ha sido necesario añadir una estacionalidad totalmente personalizada, ya que por defecto Prophet trabaja con datos diarios. Además, se ha tenido que incluir manualmente un crecimiento logístico con las diferentes capacidades de carga en cada observación. El hecho de trabajar en una serie mensual, también nos ha obligado a añadir la frecuencia de la función *make_future_dataframe*.

En la serie cinco-minutal, se ha tenido que añadir manualmente dos tipos de estacionalidades: de frecuencia diaria y de frecuencia semanal. A la vez, también se ha tenido que imponer un crecimiento logístico con las capacidades mínimas y máximas correspondientes.

Además, como antes he mencionado la estacionalidad se ajusta a través de una serie de Fourier. De este modo, para saber qué orden aplicar hay que tener un alto conocimiento no solamente de Matemáticas, ya que como futuro economista y estadístico solamente las he estudiado como antiguo estudiante de Matemáticas, sino también de Econometría, debido a que hay que saber escoger un orden que mantenga un equilibrio entre capacidad de detección de la estacionalidad y el sobreajuste.

En cuanto a uno de sus platos estrellas, las *holidays*, cabe señalar su mal resultado en la parte empírica: en la serie mensual, se han puesto como *holidays* los meses de mayo y junio, ya que eran los peor predichos por Prophet, pero además de no salir significativo, en global ha salido peor. Como alternativa, se ha pensado en introducir los cambios de punto manualmente, pero el resultado también ha sido desfavorable.

En la serie cinco-minutal, a la vista del gráfico se ha identificado que cada día sobre las 18 horas el número de bicicletas en funcionamiento era mucho mayor. Se ha creído muy necesario y útil parametrizar este hecho a través de las *holidays* pero no ha sido posible por las limitaciones de la función *prophet*. Como consecuencia, se ha visto que las *holidays* solamente funcionan para datos diarios. Por una parte, Prophet permite trabajar con datos n-minutales (o subdiarios) pero no añadirles vacaciones.

Para concluir este trabajo, cabe resaltar que en líneas generales no hay un algoritmo de predicción automática que sea mejor en todos los sentidos. Todo está sujeto a particularidades. En general, se ha visto que si se cumplen algunos supuestos difícilmente aplicables en la mayoría de casos reales como es en la serie mensual, que la estacionalidad viene muy bien definida y apenas hay tendencia, entonces parece que los modelos más simples del paquete *forecast* pueden funcionar mejor, además de ser mucho más automáticos. De todas formas, en casos de mayor complejidad, ya sea por patrones difíciles de entrenar o una gran carga de observaciones, funciona mejor tanto Prophet como NNAR. Sin embargo, en el caso de Prophet, hay que tener un importante conocimiento tanto de su función (en R o Python) como de Series Temporales.

BIBLIOGRAFÍA

1. Adebisi, A. A. (2014). Comparison of ARIMA and artificial neural networks models for stock price prediction. (Hindawi, Ed.) *Journal of Applied Mathematics*, 2014.
2. Buono, D. a. (2018). *Big Data Econometrics: Now Casting and Early Estimates*. BAFFI CAREFIN, Centre for Applied Research on International Markets Banking.
3. GERUNOV, A. A. (2016). Automating Analytics: Forecasting Time Series in Economics and Business. (J. o. Economy, Ed.) 3(2), págs. 340--349.
4. Grace, K. a. (2018). When will AI exceed human performance? Evidence from AI experts. *Journal of Artificial Intelligence Research*, 62, págs. 729--754.
5. Hassani, H. a. (2015). Forecasting with big data: A review. *Annals of Data Science*, pp. 5--19.
6. Hyndman, R. J. (2007). *Automatic time series for forecasting: the forecast package for R*. (D. o. Monash University, Ed.)
7. Hyndman, R. J.-W. (2019). Package 'forecast'. *Online]* <https://cran.r-project.org/web/packages/forecast/forecast.pdf>.
8. Jung, J.-K. a.-M. (2018). *An Algorithmic Crystal Ball: Forecasts-based on Machine Learning*. (I. M. Fund, Ed.)
9. Kane, M. J. (2014). Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks. (B. Central, Ed.) *BMC bioinformatics*, 15(1), pág. 276.
10. Lu, X. a. (2016). Volatility forecast based on the hybrid artificial neural network and GARCH-type models. (Elsevier, Ed.) *Procedia Computer Science*, 91, págs. 1044--1049.
11. Maleki, A. a. (04 de 2018). Comparison of ARIMA and NNAR Models for Forecasting Water Treatment Plant's Influent Characteristics. *KSCE Journal of Civil Engineering*, págs. 1-13. doi:10.1007/s12205-018-1195-z
12. Moharrampour, M. a. (2013). Comparison of Support Vector Machines (SVM) and Autoregressive integrated moving average (ARIMA) in daily flow forecasting. *SCIJOUR*, 1(1), págs. 1--10.
13. Quintas, I. (2000). Modelo aditivo generalizado GAM: Regresión no lineal y no paramétrica.
14. Sidi, F. a. (08 de 2013). Data quality: A survey of data quality dimensions.
15. Soler, S. V. (2001). *Quality Mining: A Data Mining Based Method for Data Quality Evaluation*.
16. Taylor, S. J. (2018). Forecasting at scale. (T. \. Francis, Ed.) *The American Statistician*, 72(1), pp. 37--45.
17. Tibshirani, R. J. (1990). Generalized additive models. (L. C. Hall, Ed.)

18. Varian, H. R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives*, 28(2), págs. 3--28.
19. Warwick, K. (2010). El futuro de la inteligencia artificial y la cibernética.
20. West, G. (2013). Big data needs a big theory to go with it. *Scientific american*, 308(5), pág. 15.
21. Yenidougan, I. a. (2018). Bitcoin Forecasting Using ARIMA and PROPHET. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)* (pp. 621--624). IEEE.

ANEXO

```
###PASO PREVIO
```

```
library(Quandl)
library(prophet)
library(forecast)
```

Serie diaria

```
###IMPORTACIÓN
```

```
a<-Quandl("BITAL/MFTSEM2019", api_key="4wm6sn6K4cdHDfHrjAd2")
```

```
#####
```

```
## PRE-PROCESSING ####
```

```
#####
```

```
a<-a[,c(1,5)]
head(a)
tail(a)
plot(ts(a$Close))
names(a)<-c('ds','y')
```

```
####MOLT IMPORTANT COMPORVAR-HO (ORDEN DE LA SERIE)
```

```
a<-a[with(a,order(a$ds)),]
```

```
##ANÁLISIS EXPLORATORIO
```

```
plot(ts(a$y))
dim(a)
ma<-prophet(a[1:86,])
```

```
#future<-make_future_dataframe(ma,periods=37)
```

```
##sustitución de la función
```

```
future <- data.frame(a$ds)
```

```
colnames(future)<-'ds'
```

```
future
```

```
dim(future)
```

```
length(ma)
```

```
forecast<-predict(ma,future)
```

```
dim(forecast)
```

```
tail(forecast[c('ds','yhat','yhat_lower','yhat_upper')])
```

```
##gráfico tendencias
```

```
prophet_plot_components(ma,forecast)
```

```

##evaluación de la capacidad predictiva: medición del error
errors<-a$y[87:123]-forecast$yhat[87:123]
errors
ts.plot(errors,col=2,main='Errores Predicción Prophet Serie Diaria')
eqm<-sum(errors*errors)/37
eam<-sum(abs(errors))/37
epam<-sum(abs(errors)/abs(a$y[87:123]))/37
c(eqm,eam,epam*100)

##evaluación gráfica
dev.off()
plot(a$y[87:123], type="o", col="blue",ylim=c(18000,24000),ylab='indicador puntos bolsa',
      xlab='Observación periodo extramostral')
lines(forecast$yhat[87:123], type="o", lty=2, col="red")
title(main="Comparación Prophet Serie Diaria", col.main="black", font.main=4)
legend(x = "topright", legend = c("Valores reales", "Prophet"),
       fill = c("blue", "red"),
       title = "Algoritmo")

#####
##### PAQUETE FORECAST #####
#####

###preparación / pre-processing
a<-Quandl("BITAL/MFTSEM2019", api_key="4wm6sn6K4cdHdfHrjAd2")
a<-a[,c(1,5)]
names(a)<-c('ds','y')
a2<-a[with(a,order(a$ds)),]
serie1<-ts(a2$y[1:86])
serie2<-ts(a2$y[87:123])

##Estacionariedad
#install.packages('tseries')
library(tseries)
adf.test(serie1,alternative='stationary')
dserie<-diff(serie1)
adf.test(dserie,alternative='stationary')

#####
##### ARIMA #####
#####

m1<-auto.arima(serie1)

```

```

m1
###normalidad residuos
tsdisplay(residuals(m1),lag.max=10,main='Residuos modelo ARIMA')
shapiro.test(m1$residuals,pnorm)
##no parece que haya ninguna presencia de patrón de retardo que se repita.
#Por lo tanto no presentamos ninguna alternativa a este modelo ARIMA.
##predicción periodo extramuestral
pred_arima<-forecast(m1,h=37)
autoplot(pred_arima)
##evaluación capacidad predictiva: medición del error
errors_arima<-serie2-ts(pred_arima$mean)
errors_arima
ts.plot(errors_arima,col=2,main='Errores predicción modelo ARIMA serie1')
eqm<-sum(errors_arima*errors_arima)/37
eam<-sum(abs(errors_arima))/37
epam<-sum(abs(errors_arima)/abs(serie2))/37
eqm
eam
epam*100

##evaluación gráfica
dev.off()
plot(as.numeric(serie2), type="o", col="blue",ylim=c(18000,25000),ylab='indicador puntos
bolsa',
      xlab='Observación periodo extramostral')
lines(as.numeric(pred_arima$mean), type="o", lty=2, col="red")
title(main="Valores reales vs Predicción ARIMA serie diaria", col.main="black", font.main=4)
legend(x = "topright", legend = c("Valores reales","Predicción ARIMA"),
       fill = c("blue","red"),
       title = "Serie")

#####
##### ETS #####
#####
modele1<-ets(serie1)
modele1

##evaluación residuos
plot(modele1$residuals,
      col='black',main='Residuos modelo ETS Bolsa Italiana',
      ylab='Valor residuos',xlab='Observación')
shapiro.test(modele1$residuals)
names(modele1)
##predicción periodo extramuestral
pred_ets<-forecast.ets(modele1,h=37)

```

```

##representación gráfica
autoplot(pred_ets)

##evalaución capacidad predictiva:medición del error
errors_ets<-serie2-ts(pred_ets$mean)
errors_ets
ts.plot(errors_ets,col=2)
eqm<-sum(errors_ets*errors_ets)/37
eam<-sum(abs(errors_ets))/37
epam<-sum(abs(errors_ets)/abs(serie2))/37
eqm
eam
epam*100

##evaluación gráfica
dev.off()
plot(as.numeric(serie2), type="o", col="blue",ylim=c(18000,25000),ylab='indicador puntos
bolsa',
      xlab='Observación periodo extramostral')
lines(as.numeric(pred_nnetar$mean), type="o", lty=2, col="red")
title(main="Valores reales vs Predicción ETS", col.main="black", font.main=4)
legend(x = "topright", legend = c("Valores reales","Predicción ETS"),
       fill = c("blue","red"),
       title = "Serie")

#####
##### NNAR #####
#####

model<-nnetar(serie1)
##evaluación residuos del modelo
plot(model$residuals,
      col='black',main='Residuos modelo NNAR Bolsa Italiana',
      ylab='Valor residuos',xlab='Observación')
shapiro.test(model$residuals)
##predicción
pred_nnetar<-forecast(model,h=37)
##representación gráfica
autoplot(pred_nnetar)

##evaluación capacidad predictiva_medición del error
errors_ann<-serie2-ts(pred_nnetar$mean)
errors_ann
eqm<-sum(errors_ann*errors_ann)/37
eam<-sum(abs(errors_ann))/37
epam<-sum(abs(errors_ann)/abs(serie2))/37

```

```

eqm
eam
epam*100

## representación gráfica
dev.off()
plot(as.numeric(serie2), type="o", col="blue",ylim=c(18000,25000),ylab='indicador puntos
bolsa',
     xlab='Observación periodo extramostral')
lines(as.numeric(pred_nnetar$mean), type="o", lty=2, col="red")
title(main="Valores reales vs Predicción NNAR", col.main="black", font.main=4)
legend(x = "topright", legend = c("Valores reales","Predicción NNAR"),
       fill = c("blue","red"),
       title = "Serie diaria")

```

```

#####
## COMPARACIÓN GRÁFICA TODOS LOS ALGORITMOS ###
#####
dev.off()
plot(a$y[87:123], type="o", col="blue",ylim=c(18000,24000),ylab='indicador puntos bolsa',
     xlab='Observación periodo extramostral')
lines(forecast$yhat[87:123], type="o", lty=2, col="red")
lines(as.numeric(pred_arima$mean),type='o',lty=2,col='black')
lines(as.numeric(pred_ets$mean),type='o',lty=2,col='orange')
lines(as.numeric(pred_nnetar$mean),type='o',lty=2,col='purple')
title(main="Comparación Serie Diaria", col.main="black", font.main=4)
legend(x = "topright", legend = c("Valores reales","Prophet","ARIMA","ETS","NNAR"),
       fill = c("blue","red","black","orange","purple"),
       title = "Algoritmo")

```

Serie mensual

```

##IMPORTACIÓN
a<- read.csv2('balears.csv',header=T,sep=';')
##Representación gráfica
plot(ts(a$y,start=2000,frequency=12),main='Serie Pernoctaciones Islas
Balears',xlab='Observaciones',ylab='Nº Pernoctaciones',col='orange')
head(a)
tail(a)
length(a$y)

##pre-processing
a<-a[1:228,]
tot<-228
train<-161
test<- tot-train

```

test

```
#####  
##### PROPHET #####  
#####
```

```
##Preparación capacidades de carga  
mes_min<-function(serie,train,period){  
  dn<-data.frame(ncol=period)  
  min<-NULL  
  max<-NULL  
  for (i in 1:period){  
    pos<-seq(i,train,by=period)  
    min<-c(min,min(serie[pos]))  
  }  
  return(min)  
}  
mes_max<-function(serie,train,period){  
  dn<-data.frame(ncol=period)  
  min<-NULL  
  max<-NULL  
  for (i in 1:period){  
    pos<-seq(i,train,by=period)  
    max<-c(max,max(serie[pos]))  
  }  
  return(max)}  
floor<-mes_min(a$y,228,12)  
cap<-mes_max(a$y,228,12)  
fl<-rep(floor,231/12)  
fl  
ca<-rep(cap,231/12)  
ca  
length(ca)##ha de ser 228
```

```
### ma -> PROPHET AUTOMÁTICA  
### m -> PROPHET MODIFICADO
```

```
ma<-prophet(a[1:train,])  
a$floor<-fl  
a$cap<-ca  
##añadiendo crecimiento y estacionalidad  
m <- prophet(weekly.seasonality=FALSE,daily.seasonality = F, growth = 'logistic')  
m <- add_seasonality(m, name='mon', period=12,fourier.order = 2)  
##ordre de fourier augmenta -> permet a la estacionalitat canviar més ràpidament(corrent el  
risc de sobreajustament)  
m <- fit.prophet(m, a[1:train,])  
future <-make_future_dataframe(m,test,freq='month')  
future_aut<-make_future_dataframe(ma,test,freq='month')  
future  
future$floor<-fl  
future$cap<-ca  
##predicciones  
forecast <- predict(m, future)  
forecast  
forecast_aut <- predict(ma, future_aut)
```

```

forecast_aut
min(forecast$yhat)##comprovación si negativo

##gráficos tendencias
prophet_plot_components(ma, forecast_aut)
prophet_plot_components(m, forecast)

##evaluación capacidad predictiva: medición del error -> PROPHET MODIFICADO
errors<-a$y[(train+1):tot]-forecast$yhat[(train+1):tot]
errors
ts.plot(errors,col=2,main='Gráfico errores PROPHET modificado')
eqm<-sum(errors*errors)/test
eam<-sum(abs(errors))/test
epam<-sum(abs(errors)/abs(a$y[(train+1):tot]))/test
eqm
eam
epam*100
c(eqm,eam,epam*100)

##evaluación capacidad predictiva: medición del error -> PROPHET AUTOMÁTICO
errors<-a$y[(train+1):tot]-forecast_aut$yhat[(train+1):tot]
errors
ts.plot(errors,col=2,main='Gráfico errores PROPHET modificado')
eqm<-sum(errors*errors)/test
eam<-sum(abs(errors))/test
epam<-sum(abs(errors)/abs(a$y[(train+1):tot]))/test
eqm
eam
epam*100
c(eqm,eam,epam*100)

##evaluación gráfica-> PROPHET MODIFICADO
dev.off()
plot(a$y[(train+1):tot], type="o", col="blue")
lines(forecast$yhat[train+1:tot], type="o", pch=22, lty=2, col="red")
title(main="Comparación PROPHET modificado serie 2", col.main="black", font.main=4)
legend(x = "topright", legend = c("Valores reales","Predicción PROPHET modificado"),
      fill = c("blue","red"),
      title = "Serie 2")

##evaluación gráfica-> PROPHET AUTOMATICO
dev.off()
plot(a$y[(train+1):tot], type="o", col="blue")
lines(forecast_aut$yhat[train+1:tot], type="o", pch=22, lty=2, col="red")
title(main="Comparación PROPHET automático serie 2", col.main="black", font.main=4)
legend(x = "topright", legend = c("Valores reales","Predicción PROPHET automático"),
      fill = c("blue","red"),
      title = "Serie 2")

#####
### PAQUETE FORECAST ###
#####
library(forecast)

###Preparación / pre-processing
tot<-228

```

```

train=161
test<-tot-train
serie1<-ts(a$y[1:train],start=2000,frequency=12)
serie1
serie2<-ts(a$y[(train+1):tot],start=c(2013,6),frequency=12)
serie2

#####
##### AUTO.ARIMA #####
#####

m1<-auto.arima(serie1)
##evalaución residuos
dev.off()
tsdisplay(residuals(m1),lag.max=30)
shapiro.test(residuals(m1))
##predicción
pred_arima<-forecast(m1,h=67)
##representación gráfica
autoplot(pred_arima)

####evaluación capacidad predictiva:medición del error
errors_arima<-serie2-pred_arima$mean
errors_arima
ts.plot(errors_arima,col=2,main='Errores predicción modelo ARIMA serie2')
eqm<-sum(errors_arima*errors_arima)/37
eam<-sum(abs(errors_arima))/37
epam<-sum(abs(errors_arima)/abs(serie2))/37
eqm
eam
epam*100

###evaluación gráfica
dev.off()
plot(as.numeric(serie2), type="o", col="blue",ylab='Nº de pern noctaciones',
      xlab='Observación periodo extramostral')
lines(as.numeric(pred_arima$mean), type="o", lty=2, col="red")
title(main="Valores reales vs Predicción ARIMA serie mensual", col.main="black", font.main=4)
legend(x = "topright", legend = c("Valores reales","Predicción ARIMA"),
       fill = c("blue","red"),
       title = "Serie")

#####
### ETS #####
#####

mets<-ets(serie1)
mets
##evaluación residuos del modelo
plot(mets$residuals,
      col='black',main='Residuos modelo ETS Pern noctaciones en Baleares',
      ylab='Valor residuos',xlab='Observación')
shapiro.test(mets$residuals)
##predicción
pred_ets<-forecast(mets,h=67)
##representación gráfica

```

```

autoplot(pred_ets)

##evaluación capacidad predictiva: medición del error
errors_ets<-serie2-pred_ets$mean
errors_ets
ts.plot(errors_ets,col=2,main='Errores de predicción ETS Baleares')
eqm<-sum(errors_ets*errors_ets)/test
eam<-sum(abs(errors_ets))/test
epam<-sum(abs(errors_ets)/abs(serie2))/test
eqm
eam
epam*100

##evaluación gráfica
plot(as.numeric(serie2), type="o", col="blue",ylab='Nº pernoctaciones',
      xlab='Observación periodo extramostral')
lines(as.numeric(pred_ets$mean), type="o", lty=2, col="red")
title(main="Valores reales vs Predicción ETS", col.main="black", font.main=4)
legend(x = "topright", legend = c("Valores reales","Predicción ETS"),
       fill = c("blue","red"),
       title = "Serie 2")

#####
##### NNAR #####
#####

ann<-nnetar(serie1)
ann
##evaluación residuos
plot(ann$residuals,
     col='black',main='Residuos modelo NNAR Pernoctaciones en Baleares',
     ylab='Valor residuos',xlab='Observación')
shapiro.test(ann$residuals)
##predicción
pred_ann<-forecast(ann,h=67)
###representación gráfica
autoplot(pred_ann)

##evaluación capacidad predictiva: medición del error
errors_ann<-serie2-pred_ann$mean
errors_ann
ts.plot(errors_ann,col=2,main='Errores de predicción NN Baleares')
eqm<-sum(errors_ann*errors_ann)/test
eam<-sum(abs(errors_ann))/test
epam<-sum(abs(errors_ann)/abs(serie2))/test
eqm
eam
epam*100

###evaluación gráfica
plot(as.numeric(serie2), type="o", col="blue",ylab='Nº pernoctaciones',
     xlab='Observación periodo extramostral')
lines(as.numeric(pred_ann$mean), type="o", lty=2, col="red")
title(main="Valores reales vs Predicción NNAR", col.main="black", font.main=4)
legend(x = "topright", legend = c("Valores reales","Predicción NNAR"),
      fill = c("blue","red"),
      title = "Serie 2")

```

```
#####
##### COMPRARACIÓN GRÁFICA ALGORITMOS #####
#####
```

```
dev.off()
plot(a$y[(train+1):tot], type="o", col="blue",ylim=c(-100000,250000),
      ylab='Número de pernoctaciones en Baleares',
      xlab='Observación periodo extramostral')
lines(forecast$yhat[(train+1):tot], type="o", lty=4, col="red")
lines(forecast_aut$yhat[(train+1):tot], type="o", lty=4, col="green")
lines(as.numeric(pred_arima$mean),type='o',lty=4,col='black')
lines(as.numeric(pred_ets$mean),type='o',lty=4,col='orange')
lines(as.numeric(pred_ann$mean),type='o',lty=4,col='purple')
title(main="Comparación Serie Mensual", col.main="black", font.main=4)
legend(x = "topright", legend = c("Valores reales","Prophet_mod","Prophet_aut",
                                  "ARIMA","ETS","NNAR"),
       fill = c("blue","red","green","black","orange","purple"),
       title = "Algoritmo")
```

Serie cinco-minutal

```
####IMPORTACIÓN
a<-read.csv2('201902BICING.csv',header=T,sep=',')

##PREPARACIÓN
a<- a[,c(2,5)]
names(a)<-c('y','ds')
head(a)
tail(a)
###PREPROCESSING
##FALTA UN DATO -> ASIGNAR NA
c<-NULL
c<-factor(c)
levels(c)<-"2019-02-05 20:35:01"
aa<-levels(a$ds)
class(aa)
aa[1400:8063]
bb<-c(aa[1:1399],levels(c),aa[1400:8063])
bb[1399:1401]

d<-NA
cc<-c(a$y[1:1399],d,a$y[1400:8063])
cc[1399:1401]
ds<-bb
y<-cc
length(ds)
a<-data.frame(ds,y)
```

```

##comprovación
a[1399:1400,]
####plot
a$y<-ts(a$y)
plot(a$y)

##capacidades (crecimiento logistico)
dia<-c(1)
while(max(dia) <length(a$y)){
  dia<-c(dia,max(dia)+288)
}
dia

setmana<-c(1)
while(max(setmana)<length(a$y)){
  setmana<-c(setmana,max(setmana)+288*7)
}
setmana
#c<-b[-length(b)]

##detección gráfica estacionalidades/ patrones de comportamiento
dev.off()
plot(a$y)
points(dia,rep(550,length(dia)),pch=12,col='orange')
points(setmana,rep(500,length(setmana)),pch=12,col='green')

mes_min<-function(serie,train,period){
  dn<-data.frame(ncol=period)
  min<-NULL
  max<-NULL
  for (i in 1:period){
    pos<-seq(i,train,by=period)
    min<-c(min,min(serie[pos][is.na(serie[pos])!=F]))
  }
  return(min)
}

mes_max<-function(serie,train,period){
  dn<-data.frame(ncol=period)
  min<-NULL
  max<-NULL
  for (i in 1:period){
    pos<-seq(i,train,by=period)
    max<-c(max,max(serie[pos][is.na(serie[pos])!=F]))
  }
  return(max)}

floor<-mes_min(a$y,8064,288)

```

```

cap<-mes_max(a$y,8064,288)

fl<-rep(floor,8064/288)
fl
ca<-rep(cap,8064/288)
ca
length(ca)
####MOLT IMPORTANT COMPORVAR-HO
#a<-a[with(a,order(a$ds)),]
##asignación parte train/ parte test
tot<-length(a$y)
train<-6451
test<- tot-train
##cada 288
## ma -> PROPHET AUTOMÁTICO
## m -> PROPHET MODIFICADO
library(prophet)
ma<-prophet(a[1:train,])
plot(ts(a$y))

a$floor<-fl
a$cap<-ca
m <- prophet(yearly.seasonality=FALSE,growth = 'logistic')
m <- add_seasonality(m, name='hora', period=288,fourier.order = 3) ##diari
m <- add_seasonality(m, name='set', period=288*7,fourier.order = 3) ##setmanal
m <- fit.prophet(m, a[1:train,])
future<-data.frame(a$ds)
future_aut<-data.frame(a$ds)
#colnames(future)<-'ds'
future$floor<-fl
future$cap<-ca
colnames(future)<-c('ds','floor','cap')
colnames(future_aut)<-c('ds')
forecast <- predict(m, future)
forecast
forecast_aut <- predict(ma, future_aut)

###tendencia prophet automatico
prophet_plot_components(ma, forecast_aut)
###tendencia prophet modificado
prophet_plot_components(m, forecast)

#####evaluación capacidad predictiva prophet modificado
errors<-a$y[(train+1):tot]-forecast$yhat[(train+1):tot]
errors

```

```

ts.plot(errors,col=2,main='Errores PROPHET manual serie Bicing')
eqm<-sum(errors*errors)/test
eam<-sum(abs(errors))/test
epam<-sum(abs(errors)/abs(a$y[(train+1):tot]))/test
eqm
eam
epam*100

```

```

#####evaluación capacidad predictiva prophet automatico
errors<-a$y[(train+1):tot]-forecast_aut$yhat[(train+1):tot]
errors
ts.plot(errors,col=2,main='Errores PROPHET automático serie Bicing')
eqm<-sum(errors*errors)/test
eam<-sum(abs(errors))/test
epam<-sum(abs(errors)/abs(a$y[(train+1):tot]))/test
eqm
eam
epam*100

```

```

##### evaluación gráfica PROPHET modificado
dev.off()
plot(a$y[(train+1):tot], type="o", col="blue",ylim=c(-100,250))
lines(forecast$yhat[train+1:tot], type="o", pch=22, lty=2, col="red")
title(main="Comparación PROPHET modificado serie 3", col.main="black", font.main=4)
legend(x = "topleft", legend = c("Real","PROPHET manual"),
      fill = c("blue","red"),
      title = "Serie 3")

```

```

##### evaluación gráfica PROPHET automático
dev.off()
plot(a$y[(train+1):tot], type="o", col="blue",ylim=c(-100,250))
lines(forecast_aut$yhat[train+1:tot], type="o", pch=22, lty=2, col="red")
title(main="Comparación PROPHET automático serie 3", col.main="black", font.main=4)
legend(x = "topleft", legend = c("Real","PROPHET automático"),
      fill = c("blue","red"),
      title = "Serie 3")

```

```

#####
#### PAQUETE FORECAST ####
#####

```

```

##IMPORTACIÓN/ pre
a<-read.csv2('201902BICING.csv',header=T,sep=',')
a<- a[,c(2,5)]

names(a)<-c('y','ds')
head(a)
tail(a)

#### PREPARACIÓN / PRE-PROCESSING
##no sirven aqui los NA
c<-NULL
c<-factor(c)
levels(c)<-"2019-02-05 20:35:01"
aa<-levels(a$ds)
class(aa)
aa[1400:8063]
bb<-c(aa[1:1399],levels(c),aa[1400:8063])
bb[1399:1401]

d<-a$y[1400-288]###ponemos la observación del día anterior-> método ingenuo
cc<-c(a$y[1:1399],d,a$y[1400:8063])
cc[1399:1401]
ds<-bb
y<-cc
length(ds)
a<-data.frame(ds,y)

##ASIGNACIÓN TRAIN/TEST
tot<-length(a$ds)
train=5760
test<-tot-train

serie1<-ts(a$y[1:train])
serie2<-ts(a$y[(train+1):tot])
serie3<-ts(a$y[1:train],frequency = 288)### periodicitat diaria
serie4<-ts(a$y[1:5760],frequency=2016)### periodicitat setmanal

#####
### auto.arima #####
#####

ari<-auto.arima(serie1)
ari
##evaluación comportamiento residuos
tsdisplay(residuals(ari),lag.max=288)

```

```

ks.test(ari$residuals)
##predicción
pred<-forecast(ari,h=test)
##representación gráfica
autoplot(pred)

##evaluación capacidad predictiva: medición errores
errors<-serie2-ts(pred$mean)
errors
ts.plot(errors,col=2)
eqm<-sum(errors*errors)/test
eam<-sum(abs(errors))/test
epam<-sum(abs(errors)/abs(serie2))/test
eqm
eam
epam*100

## evaluación gráfica de la predicción
dev.off()
plot(a$y[(train+1):tot], type="o", col="blue")
lines(as.numeric(pred_mets$mean), type="o", pch=22, lty=2, col="red")
title(main="Comparación ARIMA serie 3", col.main="black", font.main=4)
legend(x = "topleft", legend = c("Real", "ARIMA"),
      fill = c("blue", "red"),
      title = "Serie 3")

#####
#### ETS #####
#####

mets<-ets(serie1)
mets
##evaluación comportamiento residuos del modelo
ks.test(mets$residuals,pnorm)
plot(mets$residuals)
##predicción
pred_mets<-forecast(mets,h=test)
##representación
autoplot(pred_mets)

##evaluación capacidad predictiva: medición errores
errors<-serie2-ts(pred_mets$mean)
errors
ts.plot(errors,col=2)
eqm<-sum(errors*errors)/test
eam<-sum(abs(errors))/test

```

```

epam<-sum(abs(errors)/abs(serie2))/test
eqm
eam
epam*100

##evaluación gráfica de la predicción
dev.off()
plot(a$y[(train+1):tot], type="o", col="blue")
lines(as.numeric(pred_mets$mean), type="o", pch=22, lty=2, col="red")
title(main="Comparación ETS serie 3", col.main="black", font.main=4)
legend(x = "topleft", legend = c("Real", "ETS"),
      fill = c("blue", "red"),
      title = "Serie 3")

#####
#### NNAR #####
#####

##SIN FRECUENCIA
ann3<-nnetar(serie1)
ann3
##evaluación comportamiento residuos
ks.test(ann3$residuals,dnorm)
##predicción
pred_ann3<-forecast(ann,h=test)
##gráfico predicción
autoplot(pred_ann3)

##FRECUENCIA DIARIA
ann2<-nnetar(serie3)
ann2
##evaluación comportamiento residuos
ks.test(ann2$residuals,dnorm)
##predicción
pred_ann2<-forecast(ann2,h=test)
##gráfico predicción
autoplot(pred_ann2)

##FRECUENCIA SEMANAL
ann<-nnetar(serie4)
ann
##evaluación comportamiento residuos
ks.test(ann$residuals,dnorm)

```

```

##predicción
pred_ann<-forecast(ann,h=test)
##gráfico predicción
autoplot(pred_ann)

##evaluación capacidad predictiva NO FRECUENCIA
errors<-serie2-pred_ann3$mean
errors
ts.plot(errors,col=2)
eqm<-sum(errors*errors)/test
eam<-sum(abs(errors))/test
epam<-sum(abs(errors)/abs(serie2))/test
eqm
eam
epam*100

##evaluación capacidad predictiva FRECUENCIA DIARIA
errors<-serie2-pred_ann2$mean
errors
ts.plot(errors,col=2)
eqm<-sum(errors*errors)/test
eam<-sum(abs(errors))/test
epam<-sum(abs(errors)/abs(serie2))/test
eqm
eam
epam*100

##evaluación capacidad predictiva FRECUENCIA SEMANAL
errors<-serie2-pred_ann$mean
errors
ts.plot(errors,col=2)
eqm<-sum(errors*errors)/test
eam<-sum(abs(errors))/test
epam<-sum(abs(errors)/abs(serie2))/test
eqm
eam
epam*100

##evaluación gráfica FRECUENCIA SEMANAL
dev.off()
plot(a$y[(train+1):tot], type="o", col="blue",ylim=c(0,400))
lines(as.numeric(pred_ann$mean), type="o", pch=22, lty=2, col="red")
title(main="Comparación NNAN serie subdiaria", col.main="black", font.main=4)
legend(x = "topleft", legend = c("Real","NNAN"),
      fill = c("blue","red"),

```

```

title = "Serie 3")

#####
#### COMPARACIÓN GRÁFICA ALGORITMOS ####
#####

dev.off()
plot(a$y[(train+1):tot], type="o", col="blue",ylim=c(-200,300),
     ylab='Número de bicis en circulación',
     xlab='Observación periodo extramostral')
lines(forecast$yhat[(train+1):tot], type="o", lty=4, col="red")
lines(forecast_aut$yhat[(train+1):tot], type="o", lty=4, col="green")
lines(as.numeric(pred$mean),type='o',lty=4,col='black')
lines(as.numeric(pred_mets$mean),type='o',lty=4,col='orange')
lines(as.numeric(pred_ann$mean),type='o',lty=4,col='yellow')
title(main="Comparación Serie Subdiaria", col.main="black", font.main=4)
legend(x = "topright", legend = c("Valores reales","Prophet_mod","Prophet_aut",
                                  "ARIMA","ETS","NNAR"),
       fill = c("blue","red","green","black","orange','yellow'),
       title = "Algoritmo")

#####
#### PRUEBAS HOLIDAYS ####
#####
a$ds<-as.POSIXct(as.character(a$ds), format="%Y-%m-%d %H:%M:%S")

#####holidays
sor<-NULL
se<-104:108
for (i in 1:28){
  sor<-c(sor,se+288*(i-1))
}
sor

ent<-NULL
se2<-219:229
for (i in 1:28){
  ent<-c(ent,se2+288*(i-1))
}
ent
p<-sort(c(ent,sor))

entrada <- data.frame(
  holiday = 'entrada',
  ds = a$ds[ent]

```

```

)

sortida <- data.frame(
  holiday = 'sortida',
  ds = a$ds[sor]
)

holidays <- rbind(entrada, sortida)

#a$ds<-as.POSIXlt(as.character(a$ds), format="%Y-%m-%d %H:%M:%S")

#m<-prophet(a[1:train,])
#plot(ts(a$y))

a$floor<-fl
a$cap<-ca
m <- prophet(yearly.seasonality=FALSE,growth = 'logistic',holidays = holidays)
m <- add_seasonality(m, name='hora', period=288,fourier.order = 3)
m <- add_seasonality(m, name='set', period=288*7,fourier.order = 3)

m <- fit.prophet(m, a[1:train,])
future<-data.frame(a$ds)
future$floor<-fl
future$cap<-ca
colnames(future)<-c('ds','floor','cap')
forecast <- predict(m, future)
forecast

```