Predicción de series temporales en los mercados financieros
Time series forecasting in financial markets

Trabajo de Fin de Grado
Curso 2022–2023

Autores

Pablo Lozano Martín, Jaime Pastrana García, Pablo Magno Pezo Ortiz, Jun Qiu

Director

Mercedes García Merayo

Facultad de Informática

Universidad Complutense de Madrid

Predicción de series temporales en los mercados financieros
Time series forecasting in financial markets

Trabajo de Fin de Grado

Departamento de Sistemas Informáticos y Computación

Autores

Pablo Lozano Martín, Jaime Pastrana García, Pablo Magno Pezo Ortiz, Jun Qiu

Directora

Mercedes García Merayo

Facultad de Informática

Universidad Complutense de Madrid

15 de septiembre de 2023

Agradecimientos

Quisiéramos agradecer en conjunto a Mercedes, por su paciencia y dedicación a lo largo de todo el proyecto. Así como a Manuel, por estar siempre dispuesto a ayudarnos.

Jaime Pastrana García

A mi familia, por ser mi principal apoyo. En especial, a mis padres por confiar siempre en mí, a mis hermanos por transmitirme su ilusión, y a mis abuelos por escucharme y ayudarme a encontrar el camino.

A mis amigos, porque sin ellos no sería la persona que soy hoy.

Y a todos aquellos que han contribuido en mi formación como estudiante y, sobretodo, como persona. Porque son sus valores y principios los que me han permitido llegar hasta aquí.

Pablo Magno Pezo Ortiz

Quisiera agradecer a mis padres, por su constante apoyo. A mis amigos, por todas las risas y momentos compartidos. Y a todo aquel que haya creído en mi alguna vez.

Jun Qiu

A mis padres, quienes siempre me dieron confianza. A mi hermana, por su apoyo constante, y a mi gato, quien, con su reconfortante presencia, me recordó la importancia de tomar pausas y disfrutar de los pequeños placeres de la vida.

También quiero expresar mi agradecimiento a todos los profesores que he tenido a lo largo de mi carrera académica. Sus conocimientos, orientación y mentoría han sido invaluables para mi desarrollo como estudiante y como persona.

Pablo Lozano Martín

A mi familia, por darme siempre fuerza. A mis amigos, que continuamente confiaron en mí. Y a profesores como José Jaime Ruz Ortiz y César Ruiz Bermejo, por demostrarme la belleza detrás de la informática.

Resumen

Predicción de series temporales en los mercados financieros

En las últimas décadas la inteligencia artificial ha experimentado un desarrollo notable, dado en parte por un incremento exponencial del poder computacional. En este contexto, se han desarrollado diversas técnicas que permiten que la propia máquina aprenda a predecir, a partir de unos ciertos datos. El presente trabajo tiene por objetivo utilizar dichas técnicas de aprendiza automático para el análisis, y estudio, del indicador bursátil IBEX35. Para ello, se recopilarán los datos de distintas variables que puedan tener relación directa con el valor de dicho indicador, tales como el índice Dow Jones, o el precio del oro.

Recopilados los distintos datos, se entrenarán diversos modelos de inteligencia artificial, analizándose posteriormente los resultados obtenidos por los distintos modelos para la predicción del valor del índice.

Palabras clave

Series temporales, inteligencia artificial, mercados financieros, IBEX35, aprendizaje profundo, aprendizaje automático, activos financieros.

Abstract

Time series forecasting in financial markets

In recent decades, artificial intelligence has undergone significant development, partly driven by an exponential increase in computational power. In this context, various techniques have been developed that allow the machine itself to learn to predict based on certain data.

The purpose of this study is to use these machine learning techniques for the analysis and study of the stock market indicator IBEX35. To achieve this, data from different variables that may have a direct relationship with the value of this indicator will be collected, such as the Dow Jones index or the price of gold.

Once the different data is collected, various artificial intelligence models will be trained, and subsequently, the results obtained by these different models for predicting the index value will be analyzed.

Keywords

Time series, artificial intelligence, financial markets, IBEX35, deep learning, machine learning, financial assets.

Índice

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Desde que en la década de 1950 Alan Turing sentase las bases de la computación, los científicos y matemáticos empezaron a plantearse de forma realista si era posible crear una máquina que pudiese imitar, o incluso superar, la inteligencia humana. Durante los años posteriores se producen desarrollos notables, tales como la invención, en 1951, del transistor de unión, de William Shockley, que permiten que la ciencia de la computación avance exponencialmente.

Dicho avance exponencial se puede ejemplificar en la "Ley de Moore". Enunciada en 1965 por Gordon E. Moore, afirma con una base empírica que cada dos años se duplicará el numero de transistores en un microprocesador. Si bien no es una ley en el sentido estricto de la palabra, su predicción se ha cumplido durante las décadas posteriores. No obstante, por lo visto en los últimos años, parece que su vigencia está próxima a su fin.

Con la irrupción de Internet, el computador personal, y posteriormente con la llegada de los smartphones, la computación ha pasado a estar presente en las vidas de una gran parte de la población mundial. Actualmente, aproximadamente el 64 % de la población utiliza Internet y el 68 % utiliza teléfonos móviles (Datareportal (2023)).

Dichos avances tecnológicos han posibilitado el desarrollo teórico y práctico del planteamiento mencionado, el cual dio origen al campo de la inteligencia artificial. Este término fue acuñado en 1956 durante una conferencia en Dartmouth, organizada por John McCarthy, quien más tarde recibiría el prestigioso premio Turing, la máxima distinción en el ámbito de la informática. Además de su impacto teórico, estos avances han permitido la integración de la inteligencia artificial en la vida diaria de millones de personas.

El campo de la inteligencia artificial (IA) utiliza múltiples técnicas para tratar de imitar el pensamiento humano. En el presente trabajo se hará uso del aprendizaje automático, o machine learning. El objetivo principal de dicha rama es el de desarrollar técnicas que permitan a los computadores “aprender”. Es decir, se busca que las máquinas sean capaces de realizar tareas para las cuales no han sido diseñadas explícitamente previamente, y que puedan extraer relaciones y conclusiones relevantes a partir de una serie de datos.

Esta rama tiene que ver con la estadística, ya que se basa en el análisis de diversos datos. Sin embargo, desde el punto de vista de la ciencia de la computación, se introduce

la preocupación por la complejidad computacional de los problemas. En otras palabras, la medida asintótica del tiempo que se puede tardar resolviendo un problema concreto. Existen diversos tipos de algoritmos, tales como el aprendizaje supervisado, no supervisado, por refuerzo, etc.

En el presente trabajo utilizaremos el enfoque del aprendizaje supervisado, el cual permite predecir el valor de una variable continua, en este caso, el IBEX 35, a partir de una serie de datos.

El IBEX 35 es el índice de referencia en la bolsa española. Este agrupa la capitalización bursátil de treinta y cinco empresas españolas que cotizan en el Sistema de Interconexión Bursátil Español (SIBE) en las cuatro bolsas españolas: Madrid, Barcelona, Bilbao y Valencia. Las empresas que cotizan en dicho índice no son necesariamente las más grandes, sino las que mejor cumplen con los parámetros de capitalización, liquidez y volumen negociado. Dichas empresas son elegidas por el comité asesor del IBEX, que se reúne dos veces al año. Al ser el índice el valor de referencia de la bolsa española, este se utiliza como indicador del estado del conjunto de la economía española. Es, por tanto, sensible a los diversos acontecimientos sociales, políticos y económicos que atraviese el país y tiene impacto en otros índices de la bolsa internacional.

Por consiguiente, la motivación del presente trabajo será seleccionar diversos indicadores económicos que puedan tener un impacto en el índice y estudiar cómo se pueden utilizar técnicas de aprendizaje automático para predecir el valor del IBEX 35.

1.1. Objetivos

El principal objetivo de este trabajo es el de la aplicación de distintos modelos de inteligencia artificial para tratar de predecir el valor del índice IBEX 35, comparando cuales lo hacen de forma más adecuada utilizando diversas medidas para medir la precisión de la predicción. Los objetivos se podrían resumir en:

- Obtener, y procesar los datos de las variables elegidas.
- Efectuar un análisis completo del dataset obtenido.
- Entrenar los modelos elegidos.
- Analizar los resultados obtenidos por los distintos modelos, usando para ello distintas métricas.

1.2. Plan de trabajo

Para el desarrollo de este trabajo se ha contado con un equipo conformado por cuatro personas. Se ha tratado de organizar el trabajo de forma pareja, afrontando las etapas de desarrollo de forma ágil. En esta línea, se ha prescindido de extensos análisis previos y, en su lugar, se ha trabajado en escribir esta memoria, y obtener resultados desde el principio,

mejorándolos de forma iterativa. A lo largo del desarrollo del trabajo se han mantenido diversas reuniones con la directora, en las que hemos tratado diversos aspectos, y cómo mejorar las deficiencias presentes en algunos de ellos.

Dada la naturaleza del trabajo, algunas tareas dependen estrictamente de otras para poder completarse. Por ejemplo, para analizar los resultados obtenidos es necesario haber procesado antes todos los datos, y haber entrenado los modelos.

Las distintas tareas se han organizado, de forma temporal, según el siguiente diagrama de Gantt.

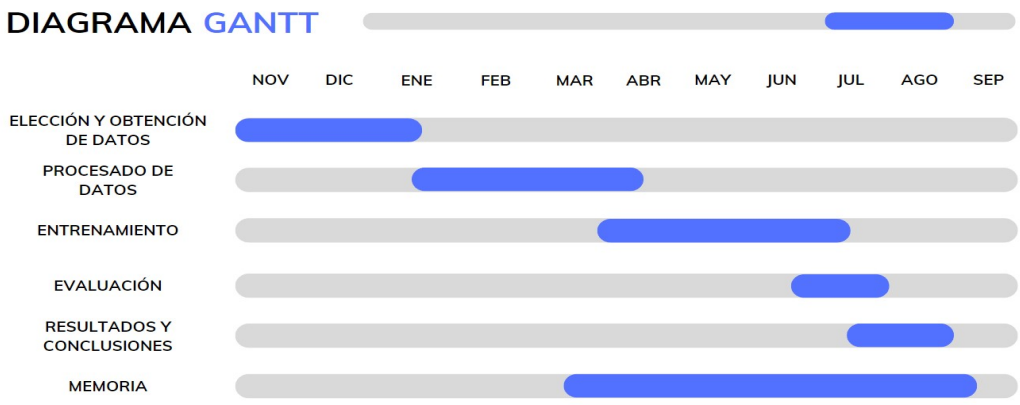

Figura 1.1: Plan de proyecto

Capítulo 2

Herramientas

En el presente capítulo se describen las herramientas empleadas para llevar a cabo tanto el preprocesado de los datos como el entrenamiento y evaluación de los modelos.

2.1. Python

Python (Van Rossum y Drake (2009)) es un lenguaje de programación ampliamente utilizado en el entorno de la IA que incluye entre sus librerías implementaciones de los modelos puestos a prueba. Además, al ser un lenguaje de muy alto nivel, permite abstraer la complejidad de los modelos con mucha facilidad, permitiéndonos centrar nuestros esfuerzos en la preparación de los datos, el ajuste de los parámetros más relevantes de los modelos y la evaluación de los resultados. También facilita la manipulación de conjuntos de datos.

2.2. Jupyter Notebook

Jupyter Notebook (Kluyver et al. (2016)) es una aplicación que permite crear documentos con secciones de código Python ejecutables. Esta herramienta es central para el desarrollo de este proyecto y nos ha permitido implementar el código Python necesario para preprocesar los datos y manipular los modelos usados. Además nos ofrece la posibilidad de incluir anotaciones en formato Markdown para facilitar la comprensión del código.

2.3. Git

Git (Git (2023)) es una herramienta de control de versiones para el desarrollo de código. Se encuentra muy extendida entre los programadores y nos permite llevar un control de los cambios que se han ido realizando en la implementación del preprocesado y el entrenamiento de los modelos.

2.4. GitHub

Github (Github (2023)) es una plataforma colaborativa basada en el sistema de control de versiones de git. Esta herramienta ha sido utilizada para almacenar y compartir el código necesario para este trabajo. Además nos permite la fácil difusión del código.

2.5. Librerías de Python

Para poder llevar a cabo este proyecto se ha necesitado hacer uso de varias librerías de Python. Estas librerías facilitan la tarea de preparación y puesta en funcionamiento de los modelos. Para ello ofrecen una serie de implementaciones tanto de dichos modelos como de operaciones necesarias para el preprocesado de los datos. A continuación se describe la utilidad de cada una de ellas en el contexto de uso que se le ha dado.

- Pandas (Pandas (2023)): permite el análisis y manipulación de datos. Utilizada para manejar y adaptar los conjuntos de datos necesarios para entrenar los modelos.
- Numpy (Harris et al. (2020)): proporciona herramientas para el manejo de vectores y matrices. Está integrada en otras librerías tales como Pandas o Matplotlib.
- Matplotlib (Hunter (2007)): sirve para crear gráficos bidimensionales. Ha sido utilizada para mostrar visualizaciones de los datos y las predicciones realizadas.
- Scikit-learn (Pedregosa et al. (2011)): librería especializada en el entrenamiento de modelos de aprendizaje automático. Se ha usado para implementar el modelo random forest.
- XGBoost (Chen y Guestrin (2016)): librería para entrenar un modelo concreto de Gradient boosting (Potenciación del gradiente).
- TensorFlow (Abadi et al. (2015)): librería para el entrenamiento de redes neuronales. Se ha utilizado para implementar los modelos de Long Short-Term Memory.
- Joblib (Joblib (2023)): librería que permite la segmentación y el guardado de funciones de Python. Se utiliza para guardar el resultado del entrenamiento de los modelos.

Capítulo 3

Preprocesado de datos

El preprocesamiento de datos es una etapa fundamental en el desarrollo de un modelo de aprendizaje automático. Esta fase de preparación es necesaria por varias razones:

- Limpieza de datos: Los conjuntos de datos reales suelen contener ruido, valores atípicos, datos nulos o inconsistentes. El preprocesado ayuda a limpiar y eliminar estos problemas, mejorando la calidad y fiabilidad de los datos utilizados para el entrenamiento.
- Normalización: Los datos pueden estar en diferentes escalas o rangos, lo que dificulta la comparación y combinación adecuada de características. La normalización es necesaria para escalar los datos y asegurar que todas las características tengan una distribución similar, lo que facilita el proceso de aprendizaje para el modelo.
- Selección y transformación de características: Los datos pueden contener una gran cantidad de características o variables, muchas de las cuales pueden no ser relevantes, o ser redundantes. El preprocesado permite seleccionar las características más relevantes y transformarlas para resaltar patrones importantes o crear nuevas características derivadas que mejoren la capacidad predictiva del modelo.
- Manejo de datos nulos: En algunos conjuntos de datos, es común que existan valores nulos. El procesamiento permite tratar adecuadamente estos datos nulos estimando como rellenar dichos valores de forma razonable.
- Reducción de dimensionalidad: En muchos casos, los conjuntos de datos pueden tener un gran número de variables, lo que puede llevar a lo que conocemos como "maldición de la dimensionalidad". Esto puede provocar que el modelo pierda capacidad predictiva. El preprocesamiento ayuda a reducir la dimensionalidad mediante diversas técnicas, lo que simplifica el modelo y reduce la complejidad computacional.

En resumen, el preprocesamiento de datos es esencial en el aprendizaje automático para garantizar que los datos utilizados para entrenar un modelo sean de alta calidad, estén en el formato adecuado y contengan las características más relevantes. Esto contribuye a mejorar

el rendimiento y la capacidad predictiva del modelo, así como a evitar problemas asociados con datos ruidosos o inconsistentes.

En el presente capítulo se abordará la cuestión de la obtención, limpieza y preprocesado de los datos, así como la conformación del conjunto de datos empleado posteriormente para entrenar el modelo.

3.1. Elección de los datos

Tras un estudio de los principales indicadores económicos, se ha decidido elegir los siguientes datos, al considerarse que son mencionados diariamente en los distintos medios de comunicación, siendo por tanto de actualidad.

Precio de la luz

En el contexto de una economía dependiente de la importación de energía a países externos, el precio de la luz suele ser representativo de algunas situaciones a nivel internacional, en especial de conflictos geopolíticos, tales como (recientemente) la guerra de Ucrania.

Precio del bono español a diez años

La cotización del bono español a 10 años es el tipo de interés o la rentabilidad que ofrece a los inversores.

Cuanto mayor sea el riesgo de invertir en dicha deuda mayor será el tipo de interés que se tendrá que ofrecer a los inversores para que la adquieran.

Precio del oro

El oro es a menudo utilizado como valor refugio, ya que se considera que no se devalúa, o lo hace en muy baja medida. Por lo tanto, en momentos de crisis económicas, políticas o sociales, es común que su precio aumente.

Euribor

Se trata de un índice de referencia publicado diariamente que indica el tipo de interés promedio al que un gran número de bancos europeos dicen concederse prestamos a corto plazo para prestárselo a terceros, particulares o empresas.

Precio del dolar con respecto al euro

Corresponde al precio al que se compra la moneda.

Índice DAX 30

Índice de referencia en Alemania, país que lidera la Unión Europea en el aspecto económico. Por tanto, tiene impacto en distintas bolsas alrededor del mundo.

Encuesta de población activa

Volumen de población activa, y de desempleados, tiene un impacto directo en la economía española, que tiene un problema endémico en cuanto a datos de empleo se refiere. La publicación de esta encuesta suele tener impacto entre los inversores.

Precio del barril de Brendt

El precio del petróleo puede ser revelador de distintos conflictos, tanto geopolíticos como económicos. Muchos países compran petróleo en previsión de posibles crisis, lo que aumentaría su valor. Otros países productores disminuyen su producción en función de cuestiones geopolíticas. Cuanta mayor sea la producción industrial, mayor será el precio del mismo.

Índice de Precios de Consumo

Es una medida estadística de la evolución de los precios de los bienes y servicios que consume la población residente en viviendas familiares en España. Está ligada a la inflación.

Índice DOW Jones

Uno de los indicadores de referencia en la bolsa de Estados Unidos, su cotización diaria tiene impacto en distintos indicadores a lo largo del mundo.

3.2. Obtención de los datos

La mayoría de indicadores han sido obtenidos a través de Investing (Investing (2023)). Sin embargo, no todos han sido obtenidas de la misma fuente por motivos de disponibilidad. Las variables obtenidas han sido nombradas siguiendo un criterio fijo consistente en señalar primero el nombre del activo en cuestión, y después la variable que representa. Las fuentes, el intervalo de fechas elegidas y las variables obtenidas han sido las siguientes:

Precio de la luz

- Intervalo: Enero 1998 - Octubre 2022
- Fuente: https://www.epdata.es/datos/precio-factura-luz-datos-estadisticas/594?accion=2
- Variables obtenidas:
    - luz_value

Precio del bono español a diez años

- Intervalo: Enero 1998 - Abril 2023
- Fuente: https://uk.investing.com/rates-bonds/spain-10-year-bond-yield-historical-data
- Variables obtenidas:
  - bono_price
  - bono_open
  - bono_high
  - bono_low
  - bono_ %

Precio del oro

- Intervalo: Enero 1998 - Abril 2023
- Fuente: https://uk.investing.com/commodities/gold-historical-data
- Variables obtenidas:
  - gold_price
  - gold_open
  - gold_high
  - gold_low
  - gold_vol
  - gold_ %

Euríbor

- Intervalo: Diciembre 1998 - Febrero 2023
- Fuente: https://www.bundesbank.de/en/statistics/money-and-capital-markets/interest-rates-and-yields/money-market-rates-651538
- Variables obtenidas:
  - euribor_value

Precio del Dólar con respecto al Euro

- Intervalo: Enero 1998 - Abril 2023
- Fuente: https://uk.investing.com/currencies/eur-usd-historical-data

- Variables obtenidas:
  - eurusd_price
  - eurusd_open
  - eurusd_high
  - eurusd_low
  - eurusd_vol
  - eurusd_ %

Índice DAX 30

- Intervalo: Enero 1998 - Abril 2022
- Fuente: https://uk.investing.com/indices/germany-30-historical-data
- Variables obtenidas:
  - dax_price
  - dax_open
  - dax_high
  - dax_low
  - dax_vol
  - dax_ %

Encuesta de población activa

- Intervalo: Marzo 2002 - Diciembre 2022
- Fuente: https://www.ine.es/prensa/epa-prensa.html
- Variables obtenidas:
  - epa_total
  - epa_activos
  - epa_ocupados
  - epa_parados
  - epa_inactivos

Precio del barril de Brendt

- Intervalo: Enero 1998 - Abril 2023
- Fuente: https://uk.investing.com/commodities/brent-oil-historical-data
- Variables obtenidas:
  - brent_price
  - brent_open
  - brent_high
  - brent_low
  - brent_vol
  - brent_ %

Índice de precios al consumo

- Intervalo: Enero 2002 - Enero 2023
- Fuente: https://www.bde.es/webbde/es/estadis/infoest/temas/sb-ipc.html
- Variables obtenidas:
  - IPC_value

Índice DOW Jones

- Intervalo: Enero 1998 - Abril 2023
- Fuente: https://uk.investing.com/indices/us-30-historical-data
- Variables obtenidas:
  - dowjones_price
  - dowjones_open
  - dowjones_high
  - dowjones_low
  - dowjones_vol
  - dowjones_ %

Indicador	Último	Apertura	Máximo
Bono español a diez años	bono_last	bono_open	bono_high
Índice DAX	dax_price	dax_open	dax_high
Índice Dow Jones	dowjones_price	dowjones_open	dowjones_high
Precio del barril de Brent	brent_price	brent_open	brent_high
Precio del oro	gold_price	gold_open	gold_high
Euro frente al Dólar	eurusd_price	eurusd_open	eurusd_high
Índice IBEX35	ibex_price	ibex_open	ibex_high

Tabla 3.1: Indicadores con variables similares redundantes

3.3. Procesado de los datos

Aunque cada elemento del conjunto de datos es distinto, y para muchos de ellos ha habido que acometer acciones especificas, es posible generalizar, para todos los elementos del conjunto de datos, las siguientes acciones.

- Todos los elementos del conjunto de datos, deberán tener un identificador que será la fecha. Para homogeneizarla, se ha utilizado el formato estándar `date` de la librería pandas.
- En linea con el punto anterior, queremos tener una fila para cada día, de tal manera que si faltasen días por algún motivo, procederemos a copiar el dato inmediatamente anterior, de forma que habrá un día por cada fila.
- En caso de que existiera alguna letra entre los datos, por ejemplo, para indicar millón (M), esta se ha suprimido, a fin de que los datos sean enteramente numéricos.
- Se ha aclarado que la coma (',') es el valor que indica los miles, y el punto los decimales, de forma que todos los valores son reconocidos como tipos numéricos.
- Se han ordenado todos los datos en orden descendente.
- En algunos casos ha sido necesario concatenar el contenido de dos archivos .csv distintos.

Existen algunos casos concretos en los que se han tenido que llevar a cabo acciones especificas, concentradas en algunas de las variables.

- En el caso de:
  - Bono español a diez años
  - Índice DAX30
  - Índice Dow Jones

- Precio del barril de Brendt
- Precio del oro
- Precio del euro frente al dolar
- Índice IBEX 35

Se ha decidido suprimir la variable que indica el porcentaje de cambio (nombre_ %), ya que se considera que no añade nueva información relevante, y puede generar problemas con la dimensionalidad.

- Para esos mismos índices, las variables *último, apertura, máximo y mínimo* , como vemos en la tabla 3.2 ofrecen información poco relevante, además de redundante. Se decide mantener la variable último, y combinar las variables máximo y mínimo de forma que se construya una medida que indique como ha variado el valor a lo largo del día. Dicha combinación se realiza mediante la siguiente formula:

$$\text{variación\_diaria} = \left(\frac{\text{máximo} - \text{mínimo}}{\text{mínimo}}\right) \times 100 \tag{3.1}$$

3.4. Conjunto de datos final

Con todo ello, el conjunto de datos final sería el siguiente:

- **Bono español a diez años:**
  - bono_price
  - bono_var
- **Encuesta de población activa:**
  - epa_total
  - epa_activos
  - epa_ocupados
  - epa_parados
  - epa_inactivos
- **Euribor:**
  - euribor_value
- **Índice DAX:**
  - dax_price
  - dax_vol
  - dax_var

- **Índice Dow Jones:**
  - dowjones_price
  - dowjones_vol
  - dowjones_var
- **Índice de precios al consumo:**
  - IPC_value
- **Precio del barril de Brent:**
  - brent_price
  - brent_vol
  - brent_var
- **Precio de la luz:**
  - luz_value
- **Precio del oro:**
  - gold_price
  - gold_vol
  - gold_var
- **Comparación del Euro frente al Dólar:**
  - eurusd_price
  - eurusd_vol
  - eurusd_var
- **Índice IBEX35:**
  - ibex_price
  - ibex_vol
  - ibex_var

Capítulo 4

Análisis descriptivo del activo

En este capítulo se analiza la evolución temporal del activo que se pretende predecir, el IBEX 35. Este índice depende de una gran cantidad de factores y, por tanto, es imposible explicar en su totalidad las causas de sus variaciones. En la Figura 4.1 se pueden ver sus valores entre el año 2002 y 2022.

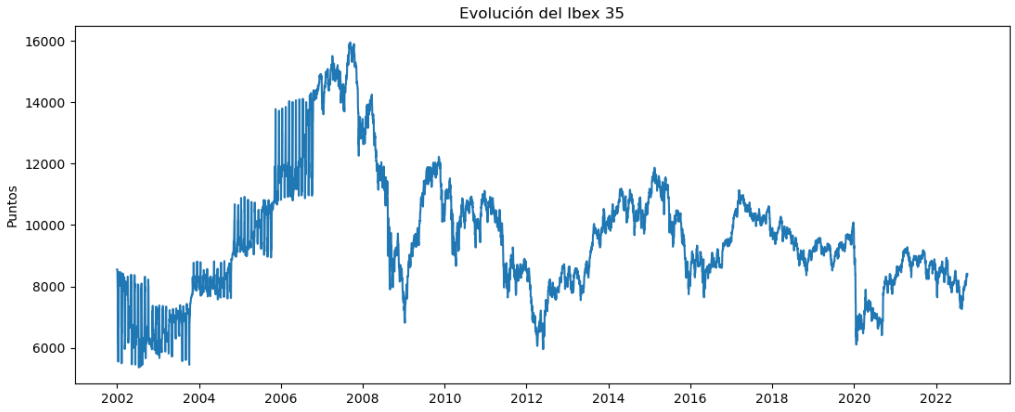

Figura 4.1: Evolución del IBEX 35

En una primera impresión, se pueden observar con claridad los efectos de la bien conocida crisis del 2008 y la pandemia del coronavirus. Ambos eventos se encuentran relacionados con las bajadas más bruscas del índice. Además, son destacables el máximo histórico de la serie, alcanzado el 8 de Noviembre de 2007 en los 15.945,7 puntos, y los mínimos relativos alcanzados en el 2012 (con 5.956,3 puntos) y 2020 (con 6.107,2 puntos).

La media se encuentra en los 9.563,96 puntos y la mediana en los 9.270,8. Estos datos parecen indicar que hay algunos valores inusualmente más altos que el resto. Esto se puede apreciar con mayor claridad en el la Figura 4.2, que muestra la distribución del IBEX 35 mediante un diagrama de caja. Como se puede observar, más de la mitad de los valores de la serie histórica se encuentran entre los 8.000 y los 11.000 puntos. Se puede ver que los valores situados por encima de los 14.000 puntos son atípicos y no representan la tendencia general de los datos. Por ello, es posible que si, de cara al entrenamiento de los modelos, se excluyen estos datos se mejore el rendimiento de sus predicciones. De este modo estaríamos

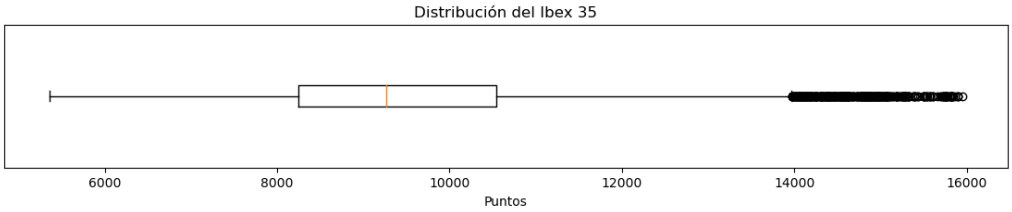

Figura 4.2: Distribución del IBEX 35 en un diagrama de caja

evitando que el modelo aprendiera patrones inusuales que probablemente no se repitan en un futuro.

En cuanto a la desviación típica, su valor es 2.012,62. Este valor viene a confirmar, tal y como se pudo ver en el diagrama de caja, que la mayor parte de valores no se alejan demasiado de la media. El diagrama de barras que aparece en la Figura 4.3 muestra la distribución de los datos y permite ver con más detalle todo lo que se ha comentado hasta el momento.

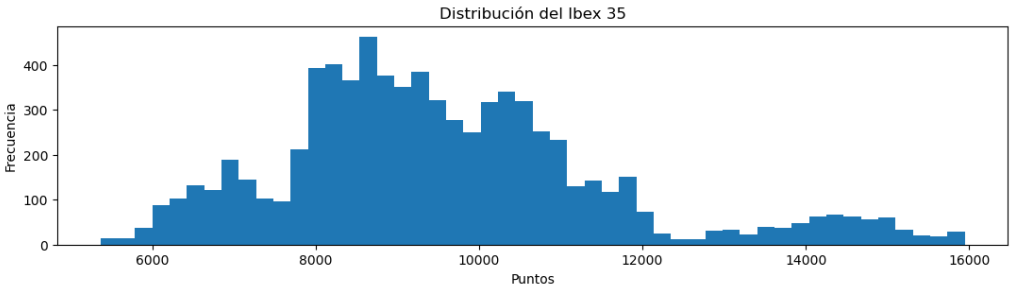

Figura 4.3: Distribución del IBEX 35 en un diagrama de barras

También es conveniente estudiar la posibilidad de que ocurra alguna estacionalidad en los datos. La Figura 4.4 muestra los valores medios del IBEX 35 para cada día de la semana. Se puede apreciar una ligera tendencia a la baja en los valores recogidos los lunes.

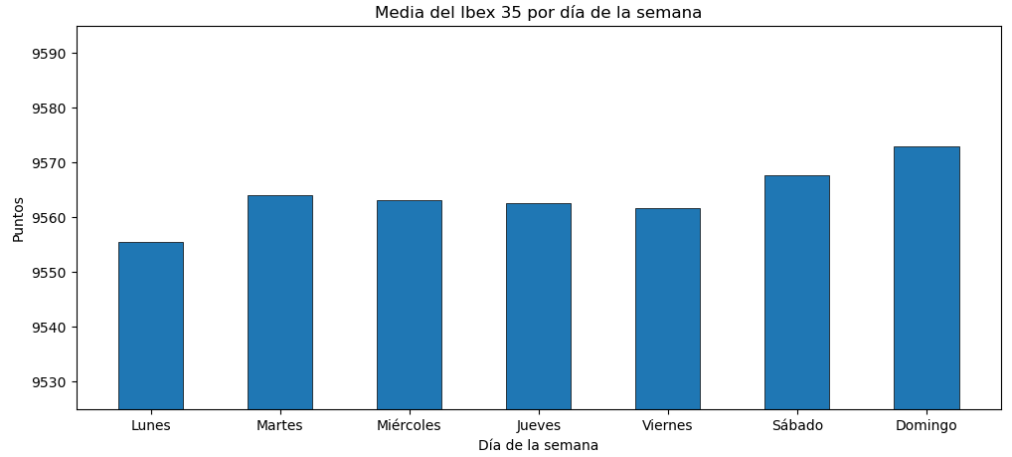

Figura 4.4: Media del IBEX 35 por día de la semana

Sin embargo, estas variaciones son mínimas, dado que no oscilan en más de 20 puntos. Por ello, se puede concluir que apenas hay relación con respecto al día de la semana en que se recogen los datos.

En cuanto al mes, podemos realizar un análisis similar mediante la gráfica correspondiente que aparece en la Figura 4.5 y que muestra los valores medios para cada mes. Como

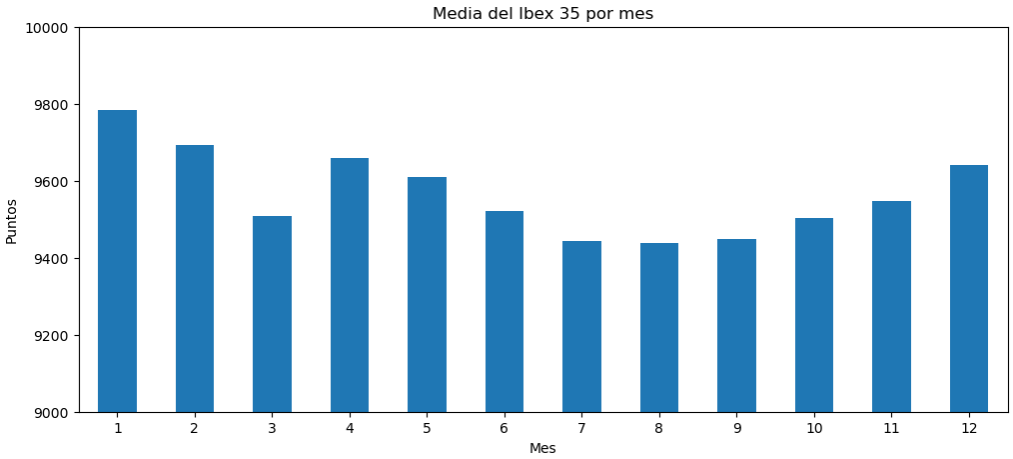

Figura 4.5: Media del IBEX 35 por mes

se puede ver, en los meses correspondientes al verano del hemisferio norte se tiene un ligero descenso en la media. Estas variaciones, aun siendo muy superiores a las apreciadas respecto a los días de la semana, tampoco son excesivamente significativas. Entre el mes de mayor media (enero) y el de menor (agosto) se tiene una diferencia de 344 puntos. Para entender con más detalle los motivos de estas variaciones en la media, se muestra en la Figura 4.6 un diagrama de cajas de la distribución de los datos cada mes. Resulta difícil, a la vista de este diagrama, explicar los motivos de las variaciones en la media. Es posible que se expliquen por valores extremos dado que la mediana, menos sensible a estos valores, no parece seguir la tendencia mencionada anteriormente. Para ver en más detalle todo esto podemos analizar la distribución del IBEX 35 en los meses de enero y julio. Estas distribuciones se muestran en las Figuras 4.7 y 4.8.

A la vista de las distribuciones de estos dos meses, se puede observar que existen valores extremos por encima de los 12.000 puntos y por debajo de los 7.500 puntos. Estos valores tienen un gran peso en el cálculo de la media y pueden justificar la mayor parte de sus variaciones. Por tanto, en rasgos generales, podemos concluir que no se aprecia ninguna tendencia estacional significativa en los datos. Otro rasgo importante a analizar es la posibilidad de que exista estacionariedad en los datos. Para ello podemos analizar la evolución de la media y la varianza a lo largo de cada año, mostradas en las Figuras 4.9 y 4.10.

Dado que tanto la media como la varianza sufren importantes variaciones con respecto a los distintos años en que se analizan, podemos decir que la serie temporal no es estacionaria. Además se puede ver como los picos más importantes en la varianza se relacionan con los

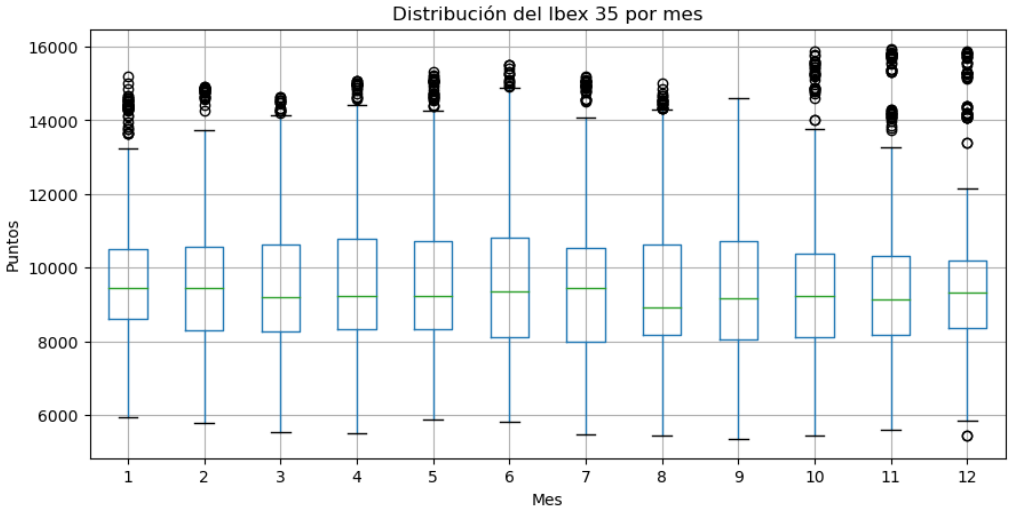

Figura 4.6: Distribución del IBEX 35 por mes

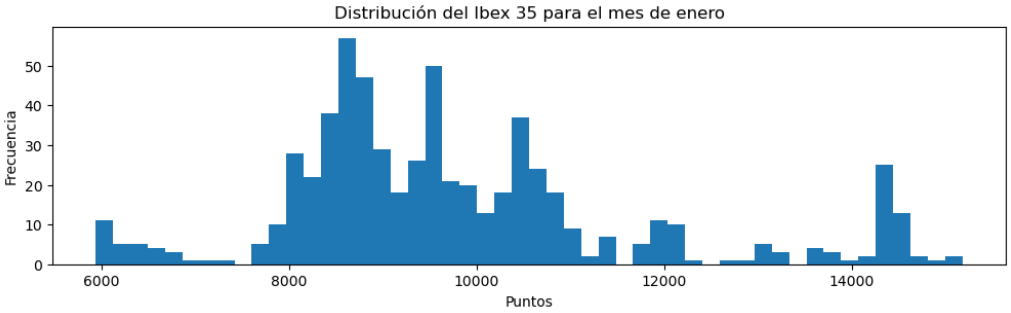

Figura 4.7: Distribución del IBEX 35 para el mes de enero

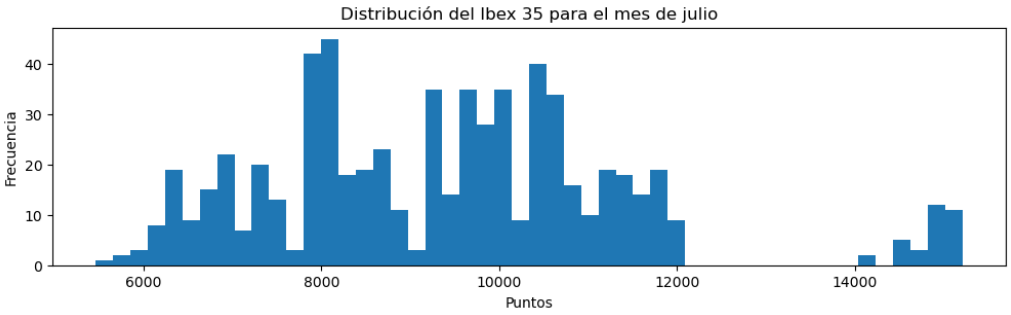

Figura 4.8: Distribución del IBEX 35 para el mes de julio

momentos de crisis económica y recesión más importantes ya mencionados al inicio de este capítulo. Esta falta de estacionariedad parece indicar que es complicado ajustar estos datos mediante una regresión.

Por último, es de especial relevancia saber si el activo a predecir se encuentra directa o inversamente correlacionado con alguna de las variables que se emplearán para ello. En caso de que el valor absoluto de la correlación con alguna variable se acerque a la unidad, esta variable será de especial relevancia a la hora de deducir la tendencia que sigue el

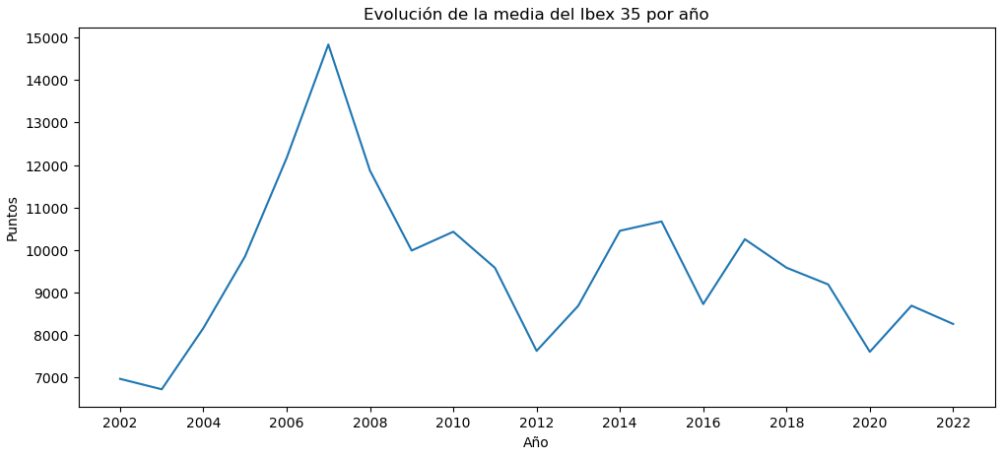

Figura 4.9: Evolución de la media del IBEX 35 por año

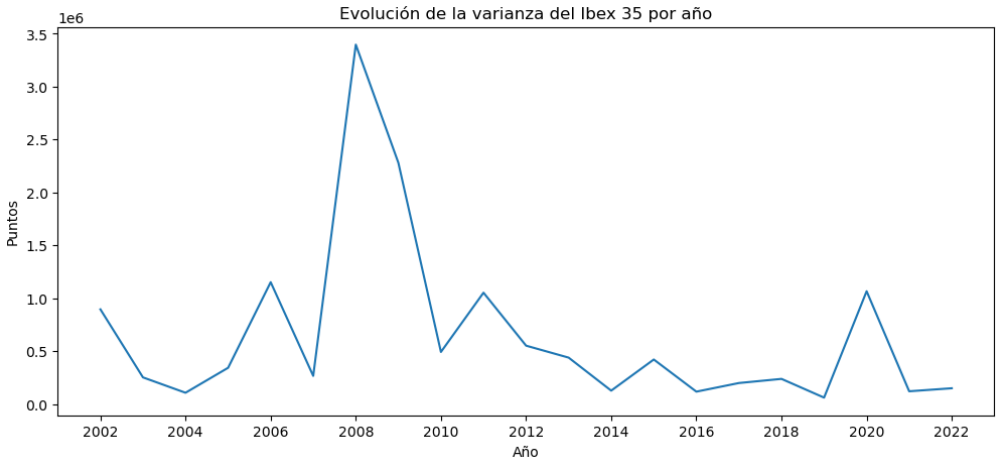

Figura 4.10: Evolución de la varianza del IBEX 35 por año

activo. Los valores de correlación de las distintas variables se muestran en la Figura 4.11.

La mayor parte de las variables muestran una correlación baja con el activo. Sin embargo, parece que la paridad entre euro y dólar, la encuesta de población activa o el Euríbor se encuentran más emparejados a la evolución del IBEX 35 que el resto de valores.

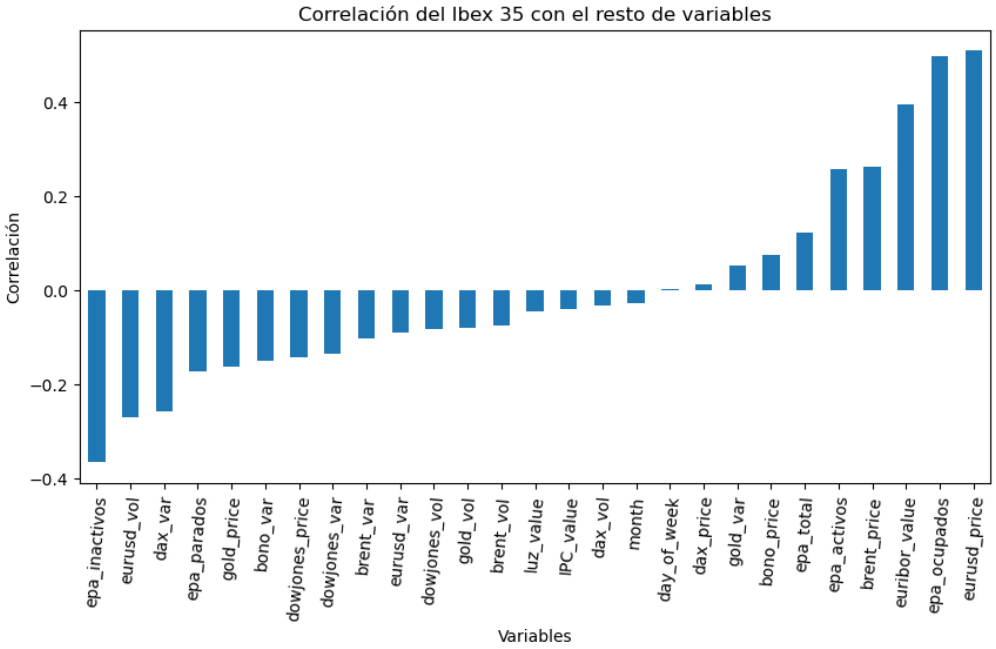

Figura 4.11: Correlación del IBEX 35 con el resto de variables

Capítulo 5

Modelos de aprendizaje

El problema que se pretende resolver es un problema de aprendizaje supervisado. Los modelos de aprendizaje supervisado son un subtipo de los modelos de aprendizaje automático. En este caso se busca entrenar un modelo que dada una entrada prediga una salida, disponiendo de un conjunto de datos que incluye la salida esperada. Este conjunto de datos sirve para entrenar y comprobar los resultados obtenidos por el modelo. En el caso de que la variable de salida sea categórica, se tratará de un problema de clasificación, y en el caso de que sea numérica, nos encontraremos ante un problema de regresión (como en este caso).

Se ha decidido experimentar con cuatro modelos distintos, dos de aprendizaje automático supervisado y dos de aprendizaje profundo. De esta manera es posible comparar los dos tipos de modelos de aprendizaje más usados actualmente para resolver este tipo de problemas. Además, dentro de cada categoría, se ha buscado experimentar con los dos modelos que mejor rendimiento pudieran tener de entre los más extendidos.

Este tipo de modelos pretenden simular el proceso de toma de decisiones necesario para predecir un valor (el activo) en función de otros (las variables). A continuación se explican los modelos puestos a prueba.

5.1. Random Forest

Este modelo se basa en la idea de combinar varios árboles de decisión para obtener una predicción final más certera que si se aplicara un solo árbol de decisión. Mantiene parte de la explicabilidad que caracteriza a los árboles de decisión al tiempo que disminuye el error cometido. Es ampliamente utilizado por la facilidad de entrenamiento y alta explicabilidad que ofrece. Por todo ello hemos considerado que este modelo es adecuado para ponerlo a prueba en la predicción de series temporales en los mercados financieros.

Para entender el funcionamiento de los Random Forests es fundamental comprender primero qué es un árbol de decisión. Un árbol de decisión consiste en la separación de casos en base a preguntas sobre los datos de entrada. De esta manera, y tras la aplicación de sucesivas preguntas, se obtiene una categoría en la que encajan dichos datos de entrada. En

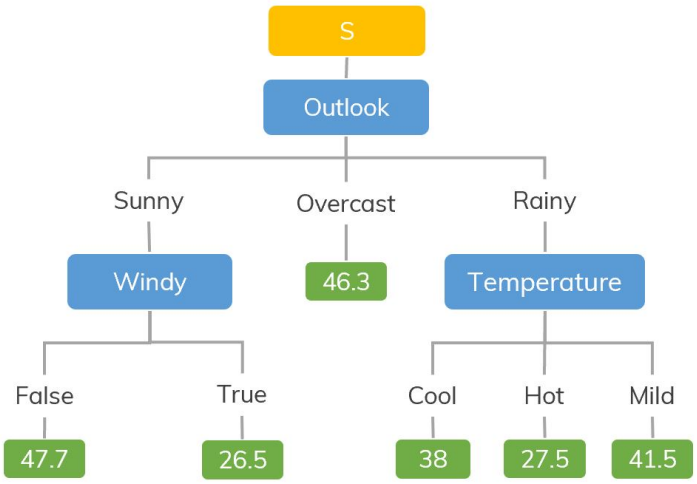

Figura 5.1: Ejemplo de árbol de decisión (Thi (2020))

base a esta categoría, se realizará una predicción u otra. Un ejemplo metafórico podría ser, por ejemplo, la toma de decisión sobre cuantos minutos salir a hacer deporte. En primer lugar, podríamos mirar por la ventana y ver si hace sol, está nublado o llueve. Este sería el primer nodo de nuestro árbol de decisión. En caso de que hiciera sol, la siguiente pregunta podría ser si hace viento. De esta manera, podríamos ir discerniendo casos y, llegado un punto, considerar si tenemos la suficiente información como para tomar una decisión final. Un caso similar se presenta en la Figura 5.1, en la cual se tiene un árbol de decisión que predice un valor numérico en función de unas condiciones ambientales observadas.

El problema de aplicar un único árbol de decisión es que, especialmente cuando el problema es complejo, tienden a aparecer errores por sobreajuste. Esto significa que el modelo aprende patrones muy concretos existentes en los datos de entrenamiento pero que no se reproducen en datos generales. Es por esto que el algoritmo de Random Forest busca combinar el resultado de entrenar múltiples árboles de decisión ligeramente distintos sobre el mismo problema, facilitando así la tarea de generalizar el conocimiento adquirido por el modelo.

Para aplicar el algoritmo de Random Forest hay tres hiperparámetros que resultan esenciales: el número de árboles que se generan, el número máximo de características o variables en las que se basa cada nodo de los árboles de decisión y la profundidad máxima del árbol. Estos hiperparámetros se deben ajustar en función de los datos, de manera que se encuentre la configuración que mejores resultados dé.

La metodología seguida para entrenar los distintos árboles y combinar sus predicciones se denomina *bagging*. Mediante este método, aplicable a cualquier modelo de aprendizaje automático, se combinan dos acciones: el *bootstrapping* y la agregación de resultados. El primer paso, correspondiente a *bootstrapping*, consiste en escoger un subconjunto de los datos de entrenamiento con la posibilidad de tener casos repetidos. Además, en este paso, el subconjunto de datos podrá tener menos variables que las del conjunto original si así se

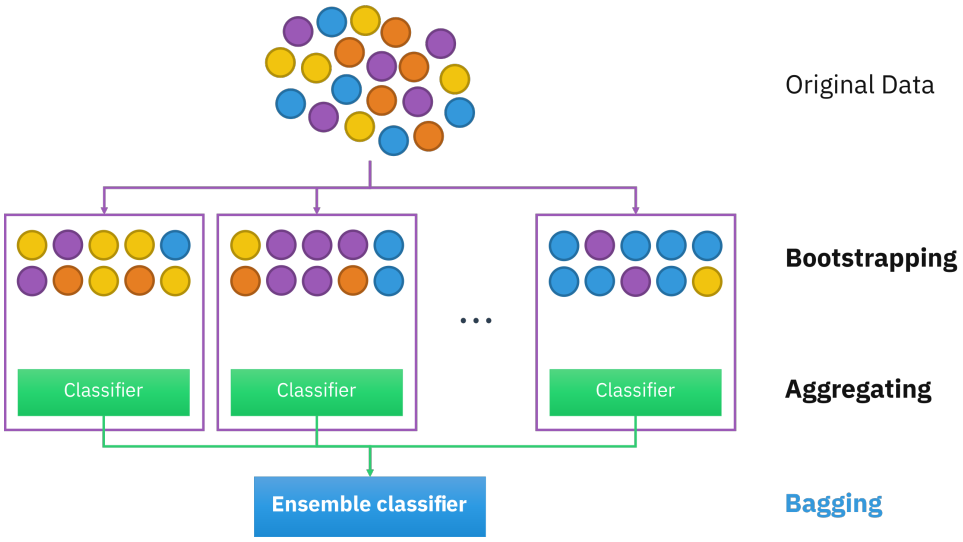

Figura 5.2: Esquema del proceso de bagging (Sirakorn (2020a))

especifica en los hiperparámetros del modelo. Tras esto, se entrenan los distintos árboles de decisión. Finalmente, se deberán combinar las soluciones que presentan los distintos árboles de decisión. Para ello, en el caso de los problemas de regresión, se tomará la media de los valores de salida. Todo este proceso se resume en la Figura 5.2.

Como ya se ha mencionado una de las ventajas de usar este modelo frente a un simple árbol de decisión es que se mantiene parte de la explicabilidad de los resultados obtenidos. Dicha explicabilidad se resume en que podemos determinar qué variables han tenido más importancia a la hora de realizar la predicción. Sin embargo, este proceso también conlleva la desventaja de tener que entrenar un modelo más complejo y costoso, lo que hace que sea más complicado interpretar el motivo de cada decisión tomada y que tengamos que emplear más tiempo para su entrenamiento.

5.2. XGBoost

XGBoost es un algoritmo que implementa la idea de potenciación del gradiente mediante *boosting*. Este algoritmo es muy popular por haber sido el empleado para ganar numerosas competiciones de aprendizaje automático en los últimos años. Especialmente ha destacado en la plataforma de Kaggle (Kaggle (2023)), que es la más popular en el contexto de las competiciones online de modelos de inteligencia artificial. Al igual que el modelo de Random Forest, consiste en combinar varios árboles de decisión, manteniendo parte de la explicabilidad que ofrecen. Sin embargo, el método de generación de estos árboles de decisión es distinto, ofreciendo una alternativa que ha demostrado ser mucho más eficaz en multitud de casos.

Al igual que en el caso de Random Forest, es esencial conocer el funcionamiento de los árboles de decisión antes de entrar a analizar el funcionamiento de este modelo de potenciación del gradiente. El motivo reside en que la idea de la que se parte es la misma,

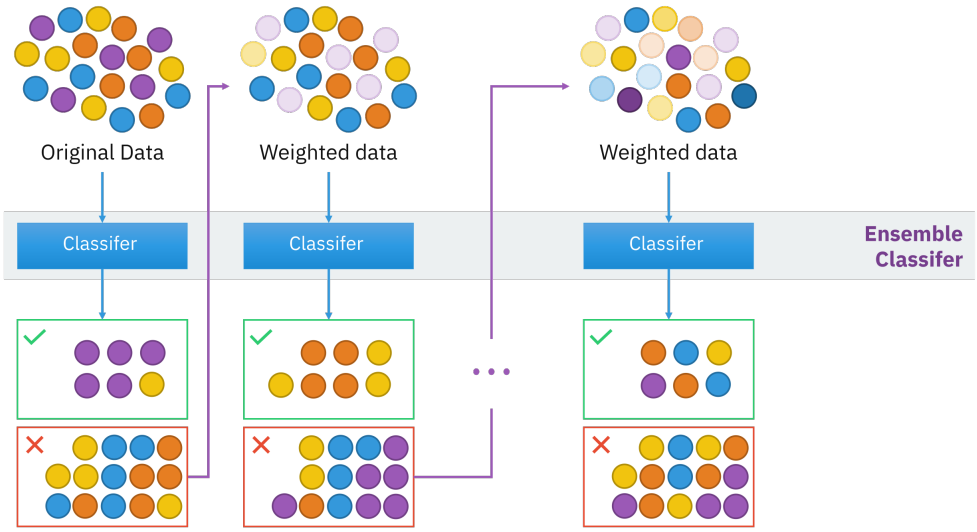

Figura 5.3: Esquema del proceso de boosting (Sirakorn (2020b))

es decir, mejorar los resultados que puede ofrecernos un simple árbol de decisión. Para ello, en este caso, se seguirá un proceso iterativo, llamado *boosting*, por el cual cada árbol de decisión supondrá una mejora sobre el anterior. Para lograr esto se aplica el algoritmo de descenso del gradiente, que progresa en busca de un mínimo local lo más cercano posible al absoluto. En la Figura 5.3 se puede ver un esquema que resume el proceso que se llevará a cabo según se ha descrito.

Uno de los hiperparámetros más relevantes de cara a la aplicación de este modelo es la tasa de aprendizaje. Este valor determinará la velocidad con la que el algoritmo de descenso del gradiente se aproximará hacia los mínimos locales. En caso de que este valor se ajuste por debajo de su valor óptimo, el algoritmo consumirá más tiempo del necesario en el proceso de entrenamiento. Además, podría localizar un mínimo local muy lejano al mínimo absoluto. En el caso contrario, si se especificara una tasa de aprendizaje excesivamente elevada, el algoritmo no será capaz de converger a una zona de mínimo coste. Esto es porque los pasos que da para avanzar hacia dicho mínimo son excesivamente grandes, haciendo imposible que el algoritmo se estabilice en la zona de coste esperada. En la Figura 5.4 se muestra de forma visual un punto cuya función de coste se pretende minimizar al buscar situarlo en las zonas de color más intenso. En este caso la tasa de aprendizaje es adecuada.

Para optimizar este parámetro también existen métodos dinámicos que van variando el valor que tiene a lo largo del propio proceso de descenso del gradiente. De esta manera se puede intentar optimizar la velocidad a la que converge el modelo al tiempo que no se pierde su capacidad para encontrar mínimos locales óptimos.

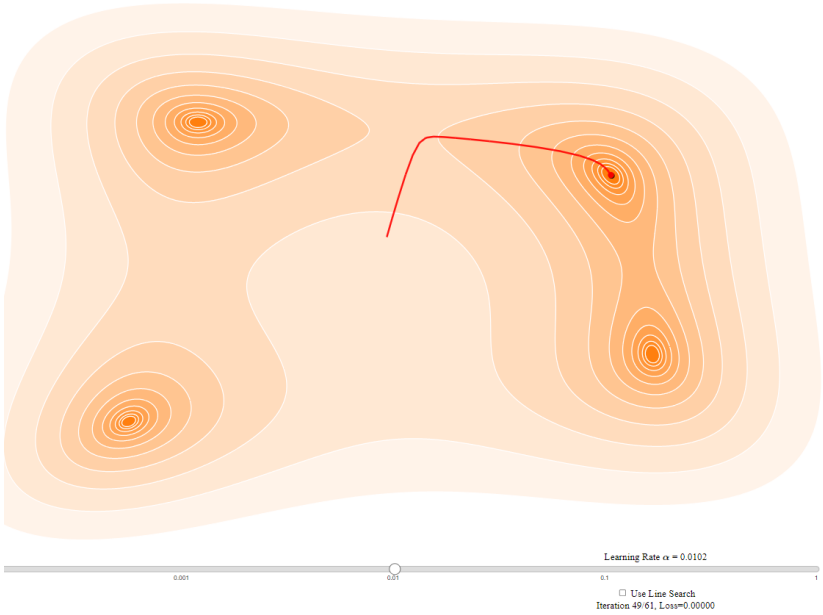

Figura 5.4: Visualización del algoritmo de descenso del gradiente (Frederickson (2023))

5.3. Modelos de aprendizaje profundo

Este tipo de modelos se encuentran dentro de la categoría de los modelos de aprendizaje automático. Sin embargo, aplican un enfoque muy distinto a los modelos tradicionalmente más usados en este contexto, suponiendo por ello una alternativa que merece la pena tratar por separado. De hecho, es en los últimos años cuando este tipo de modelos ha alcanzado su mayor popularidad y nivel de desarrollo.

Los modelos de aprendizaje profundo se basan en imitar el concepto que hace funcionar las redes neuronales biológicas. Para ello, se crean redes neuronales artificiales en las que las neuronas se suelen distribuir en capas. Caben destacar los siguientes tipos de capas:

- Capa de entrada: en ella se sitúan las variables de entrada. En realidad, esta capa no contiene neuronas, puesto que no se aplica ningún cálculo sobre las variables de entrada.
- Capa de salida: es la que agrupa a las neuronas que muestran la predicción dada por el modelo.
- Capas ocultas: son las capas intermedias que constituyen el resto de la red neuronal. Conectan la capa de entrada con la capa de salida.

Cada neurona recibe unos parámetros de entrada de la capa anterior y produce una salida. Dicha salida se puede resumir como el resultado de aplicar a los valores de entrada una suma ponderada y una posterior función de activación no lineal. Estos conceptos se pueden ver sobre la Figura 5.5, que muestra el esquema de una red neuronal de varias capas de profundidad.

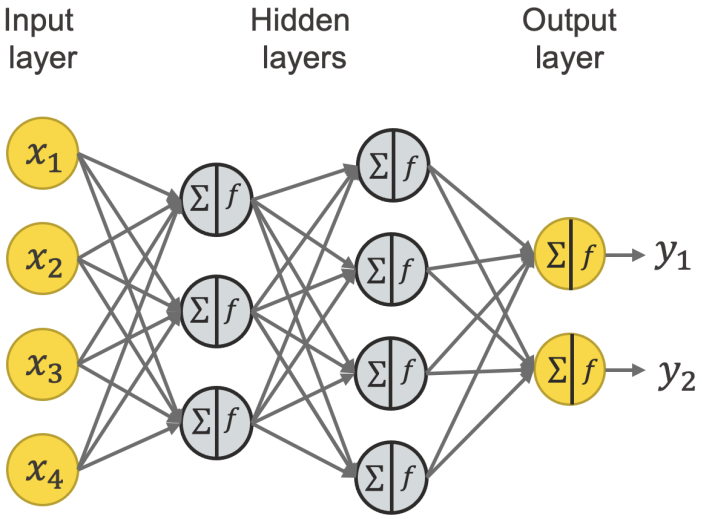

Figura 5.5: Esquema de una red neuronal (Melcher (2021))

Cada neurona, como hemos mencionado anteriormente, es capaz de, dados una serie de valores de entrada, producir una salida. Para producir dicha salida se hace uso de unos parámetros que se deberán ajustar para cada neurona. Estos parámetros no se ajustan manualmente, sino que se calibran de forma automática mediante el algoritmo de *backpropagation* (o retropropagación). Este algoritmo, durante el proceso de entrenamiento de la red neuronal, será el que, en función de los errores, irá variando estos parámetros en busca de minimizar el error. El criterio que sigue es el de, dada una salida errónea, propagar la responsabilidad hacia las neuronas que han tenido más peso en su producción. De esta manera, las neuronas que son responsables de salidas erróneas serán reajustadas. Aplicando este proceso de forma iterativa sobre los datos de entrenamiento se logrará alcanzar un punto en el que la red neuronal alcance cierto equilibrio en sus predicciones.

Cabe destacar también la baja explicabilidad que ofrecen este tipo de modelos. Dada la complejidad con la que la red neuronal almacena el conocimiento, es prácticamente imposible saber los motivos que han llevado a la red neuronal a realizar una predicción determinada. Por ejemplo, es imposible determinar la importancia que se le da a cada una de las variables de entrada para realizar una predicción final.

Para el problema que pretendemos resolver se han elegido dos modelos muy similares, Long Short-Term Memory (LSTM) (Hochreiter y Schmidhuber (1997)) y LSTM bidireccional. Se trata de modelos que pertenecen a la categoría de las Redes Neuronales Recurrentes. Se caracterizan por poseer la capacidad de recordar eventos pasados que sucedieron hace mucho tiempo, lo que los hace ideales para realizar predicciones sobre series temporales. Algunos de sus hiperparámetros más relevantes son:

- Épocas: número de veces que se pasa el conjunto de datos de entrenamiento sobre la red neuronal.

- Tamaño del lote: número de filas de datos que se usan en cada época. En nuestro caso, número de días de los que se hace uso en cada época.
- Número de neuronas de la red: permite ajustar la complejidad que queremos que tenga la red neuronal.
- Función de pérdida: mide el error obtenido entre la predicción y el valor esperado.
- Optimizador: proporciona un algoritmo para optimizar el descenso de gradiente que se aplicará sobre la red neuronal.

A continuación se describen en detalle los modelos elegidos.

5.3.1. LSTM

Las redes LSTM son un tipo de red neuronal que, tal y como se ha descrito anteriormente, poseen unas características que le permiten, a diferencia de otros tipos de redes neuronales recurrentes, mantener una memoria a largo plazo. Esto, en el contexto de la predicción de series temporales, resulta de especial utilidad, ya que permite basar las predicciones de eventos actuales en evoluciones sucedidas con mucha anterioridad.

Las redes neuronales recurrentes poseen una memoria a corto plazo que las caracteriza y las permite comprender tendencias en los datos. Esta memoria funciona gracias a que, cada vez que el modelo realiza una predicción, se almacena un estado interno de la red neuronal que será usado como parámetro en la siguiente predicción, tal y como se puede ver en la Figura 5.6. Como consecuencia de esto, un suceso acontecido en un punto concreto de la serie temporal quedará reflejado con gran intensidad en el estado interno de la red neuronal de cara a la siguiente predicción. Sin embargo, esta huella que ha dejado el suceso en el estado interno de la memoria a corto plazo se irá diluyendo conforme se vayan realizando las siguientes predicciones, dado que estas se irán superponiendo e irán cambiando la memoria a lo largo del tiempo.

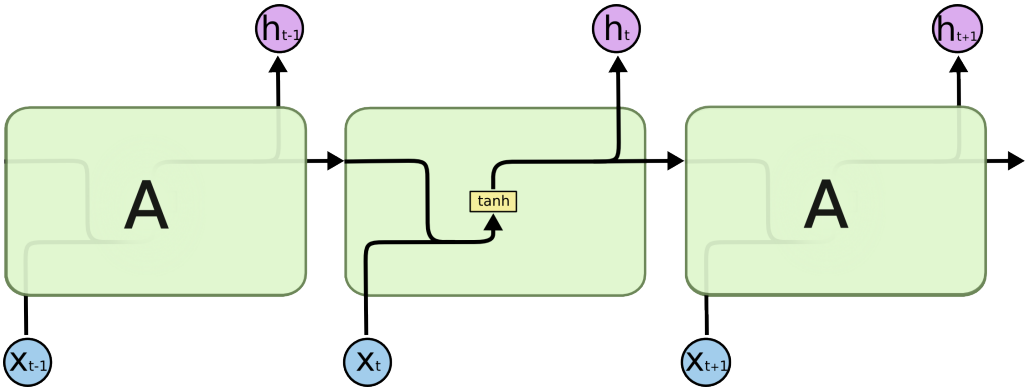

Figura 5.6: Esquema del funcionamiento interno de una red neuronal recurrente (Olah 2015a)

En base al problema descrito anteriormente surgen las redes LSTM. Estas redes, que mantienen la misma base funcional, añaden una celda adicional que mantendrá un estado interno de la red neuronal enfocado a retener información a largo plazo. Para ello, la red neuronal deberá ser capaz de distinguir, dada una predicción, si la celda de memoria debe olvidar la información almacenada, incluir información respecto a la predicción actual o modificar la predicción actual haciendo uso de la memoria a largo plazo. Para tomar estas decisiones se tienen unas estructuras especializadas en esta tarea. Estas estructuras se llaman puertas y hay 3, una para cada decisión. Cada una de ellas se describe a continuación:

- Puerta de olvido (*forget gate*): decide en qué medida se mantendrá el estado anterior de la celda de memoria de largo plazo. Para ello, teniendo como salida de la puerta un valor entre 0 y 1, multiplica el estado de la celda por dicho valor, olvidando el estado si el valor es cercano a 0 y almacenándolo si es cercano a 1.

- Puerta de entrada (*input gate*): decide qué información se añade a la celda de memoria a largo plazo. Para obtener esta información se realizan una serie de operaciones sobre la entrada.

- Puerta de salida (*output gate*): decide qué parte de la información contenida en la celda de memoria se traslada a la salida. Combina, por tanto, la información de la entrada, la celda de memoria a corto plazo y la celda de memoria a largo plazo.

En la Figura 5.7 se presenta un esquema de la estructura interna de las puertas en una red neuronal LSTM. Se puede apreciar también la aplicación de las funciones sigma (σ) y tangente hiperbólica ($tanh$) en la estructura interna de las puertas.

5.3.2. LSTM bidireccional

Una posible manera de mejorar el planteamiento ofrecido por una red LSTM estándar es añadiéndole bidireccionalidad. Esto podría ayudar a reforzar el aprendizaje del modelo haciéndolo más consciente del contexto en que se reciben los datos. Es por ello que se ha decidido experimentar con él, de modo que se pueda analizar si resulta útil aplicar dicha bidireccionalidad en el contexto de la predicción de series temporales en mercados financieros.

El modelo de red LSTM bidireccional conserva la misma base conceptual que el modelo LSTM simple. La bidireccionalidad referida a una red neuronal recurrente consiste en combinar el resultado de dos redes que se diferencian en el orden en que han recibido los datos. Una de ellas recibirá los datos en el orden temporal de los mismos, tal y como se hace habitualmente, mientras que la otra recibe los datos en orden inverso. La salida que se empareja de una red neuronal con la otra lo hace con las secciones del modelo que recibieron los mismos datos de entrada. Este procedimiento, que aparentemente es muy complejo, se puede observar de forma esquemática en la Figura 5.8.

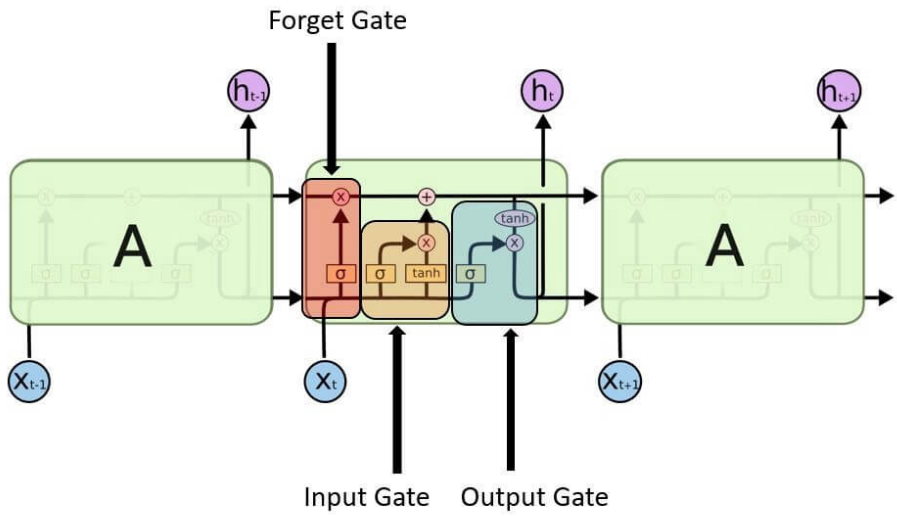

Figura 5.7: Estructura de las puertas en una red LSTM (Kalita (2022))

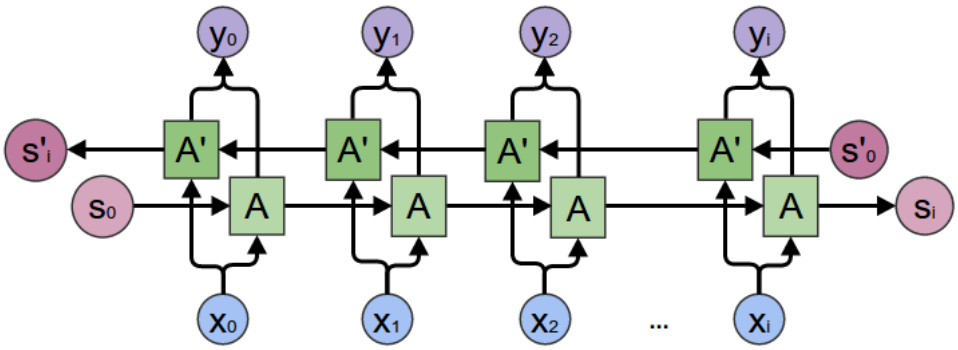

Figura 5.8: Esquema de una red neuronal recurrente bidireccional (Olah (2015b))

Es de especial relevancia el hecho de que el grafo formado por las dependencias entre las diversas componentes es un grafo acíclico. Esto es esencial debido a que de lo contrario el cómputo necesario para su entrenamiento resultaría imposible de llevar a cabo.

Más allá del mecanismo de bidireccionalidad explicado anteriormente, el entrenamiento de este tipo de redes neuronales no difiere en ningún otro aspecto. El resto de algoritmos que se aplican para el entrenamiento de una red LSTM básica se aplican de igual manera a su versión bidireccional. La ventaja de esta aproximación es la posibilidad añadida de que la red neuronal realice el proceso de aprendizaje teniendo información tanto del pasado como del futuro en el que se encuentran esos datos en la serie temporal. Esto potencialmente permite fortalecer la certeza de las predicciones al haberse tenido un contexto para el entrenamiento de las mismas. Su principal desventaja es el aumento de la complejidad computacional del modelo, dado que el número de cálculos necesarios se duplica.

Para comprender mejor cómo funciona una red LSTM bidireccional, imaginemos un

escenario de predicción de precios de acciones en el mercado financiero. Supongamos que tenemos una secuencia de precios históricos de acciones: [100, 105, 98, 110, 115]. Al aplicar una red LSTM bidireccional a esta secuencia, se procesa tanto en el orden cronológico normal como en el orden inverso. En el primer paso, la red LSTM bidireccional analiza los datos de entrada en el orden temporal original: [100, 105, 98, 110, 115]. Captura patrones y dependencias basadas en la información pasada y presente. En el segundo paso, la red LSTM bidireccional analiza los datos en orden inverso: [115, 110, 98, 105, 100]. Esto permite que la red capture patrones y dependencias basadas en información futura o posterior en la secuencia. Luego, los resultados de ambas direcciones se combinan para generar predicciones más precisas. Por ejemplo, si la red LSTM bidireccional aprende que los precios de las acciones generalmente aumentan después de un patrón específico en la secuencia, podrá utilizar ese conocimiento para hacer estimaciones más acertadas sobre el siguiente valor en la serie, incluso sin tener información futura real disponible.

Este enfoque bidireccional proporciona a la red LSTM una comprensión más completa del contexto en el que se reciben los datos, lo que puede mejorar su capacidad para predecir con mayor precisión los precios futuros de las acciones en el mercado financiero.

Capítulo 6

Entrenamiento de los modelos

En el presente capítulo se describe el proceso de entrenamiento y puesta a prueba de los modelos elegidos. Este es el paso previo a la evaluación de dichos modelos, lo que a su vez nos permitirá determinar la idoneidad de cada uno de ellos para realizar predicciones en el tipo de series temporales que tenemos.

Todos los modelos se han entrenado siguiendo el mismo esquema general. Este esquema se ha plasmado en un documento de Jupyter Notebook para cada modelo entrenado. El código incluido en dichos documentos permite configurar mediante la modificación de variables el número de días a futuro con que se quiere predecir y el número de días en base a los cuales se realiza dicha predicción. Las operaciones más importantes realizadas para entrenar los modelos se resumen en los siguientes puntos:

- Carga y adaptación de los datos al modelo. Se lee el fichero fuente de los datos y se adapta de modo que por cada fila de nuestro dataframe se tengan el activo (el valor a predecir del IBEX 35) y sus correspondientes variables. Las variables incluidas engloban los datos obtenidos en los días anteriores tanto del IBEX 35 como del resto de valores recogidos. Además se normalizan los datos y se elimina la fecha.

- Separación de los datos en conjunto de entrenamiento y conjunto de evaluación. Por tanto, el 80 % de los datos se dedicarán a entrenar el modelo y el 20 % restante a comprobar su rendimiento. La separación se hace sin aleatoriedad debido a que estamos tratando una serie de datos con dependencias temporales. La separación realizada se puede visualizar en el gráfico correspondiente a la Figura 6.1.

- Entrenamiento del modelo. Para ello se hace uso de las interfaces que nos ofrecen las herramientas de Scikit-learn y TensorFlow según el modelo que se desea entrenar. Se configuran los hiperparámetros con los que se quiere entrenar el modelo.

- Evaluación de los resultados. Se calculan varias métricas para medir el error que comete el modelo sobre los datos de evaluación y se visualizan las predicciones. Tras esto se puede decidir entrenar nuevamente el modelo con distintos valores para los hiperparámetros con el objetivo de optimizar el entrenamiento del modelo.

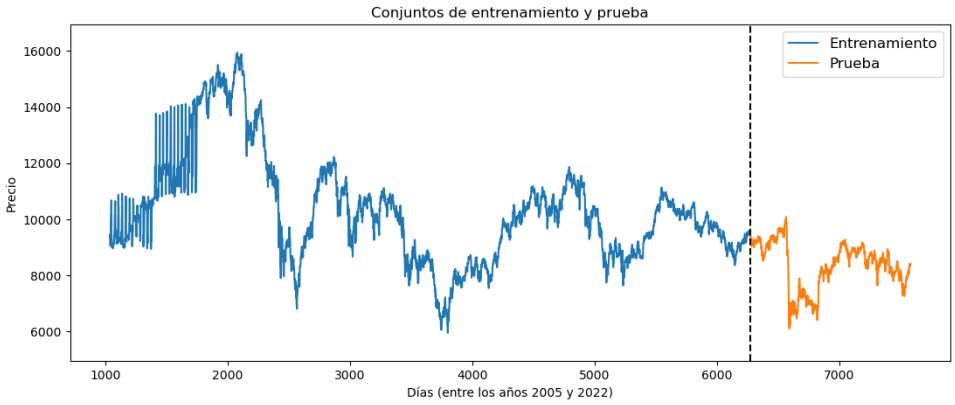

Figura 6.1: Separación de los conjuntos de entrenamiento y evaluación

- Guardado del modelo en un fichero. Esto sirve para evitar tener que volver a entrenar el mismo modelo en caso de que se quiera hacer uso de él en un futuro.

Todos los modelos entrenados mediante este esquema se han puesto a prueba en la predicción con 1, 7 y 30 días de antelación. Se toman como variables de entrada los distintos valores tomados en los 25 días anteriores al día en que se realiza la predicción.

Aunque el proceso de entrenamiento es muy similar para los distintos modelos, existen algunas características específicas de cada modelo que merecen ser mencionadas por separado. A continuación se explican para cada modelo las diferencias en su entrenamiento respecto al esquema general.

6.1. Random Forest

Para el entrenamiento de este modelo se han calibrado los siguientes hiperparámetros:

- El número de árboles
- La profundidad máxima de los árboles
- El número máximo de variables a usar en el entrenamiento de cada árbol.

Además, se han incluido el mes y el día de la semana como variables adicionales, aunque posteriormente han demostrado no resultar relevantes para el modelo.

Para calibrar los hiperparámetros, tal y como se menciona en el esquema general, se han ido probando distintos valores hasta encontrar alguno que mostrara un rendimiento adecuado. En la Figura 6.2 se muestra el resultado de calcular la raíz cuadrada del error cuadrático medio para el modelo entrenado con distintos valores en el hiperparámetro correspondiente al número de árboles de decisión generados. A la vista de estos resultados se puede ver que el valor más óptimo encontrado en este caso es de 10. Esto nos ha servido, por tanto, para encontrar un valor adecuado para el hiperparámetro con el que se ha experimentado.

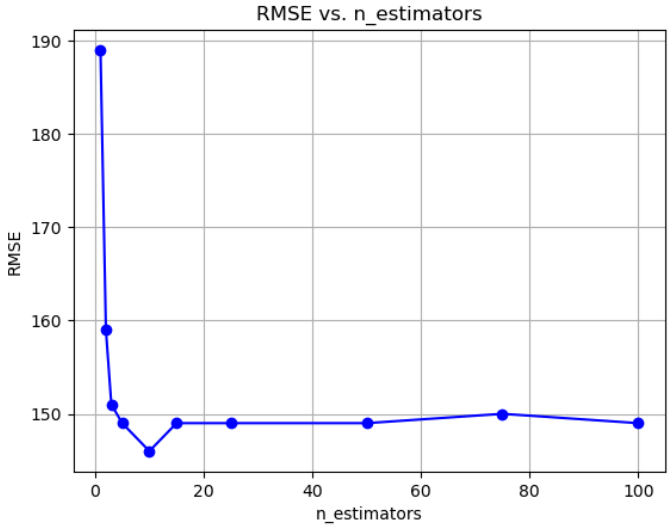

Figura 6.2: Error cuadrático medio frente a número de árboles de decisión generados

6.2. XGBoost

Para el entrenamiento de este modelo se han calibrado los siguientes hiperparámetros:

- El número de árboles máximos a generar.
- La tasa de aprendizaje.

Además, al igual que el modelo de Random Forest, se han incluido el mes y el día de la semana como variables de entrada.

Para calibrar los hiperparámetros, tal y como se menciona en el esquema general, se han ido probando distintos valores hasta encontrar alguno que mostrara un rendimiento adecuado. En la Figura 6.3 se muestra el resultado de calcular la raíz cuadrada del RMSE para el modelo entrenado con distintos valores en el hiperparámetro correspondiente al número de árboles de decisión generados. A la vista de estos resultados se puede ver que el valor más óptimo encontrado en este caso es de 100. Esto es porque es el que, siendo de los que mantienen un rendimiento adecuado, minimizan el número de árboles que se necesitan generar. Esto nos ha servido, por tanto, para encontrar un valor adecuado para el hiperparámetro con el que se ha experimentado.

6.3. LSTM

Para el entrenamiento de este modelo se han calibrado los siguientes hiperparámetros:

- El número de épocas.
- El tamaño de lote.
- El número de neuronas por capa.

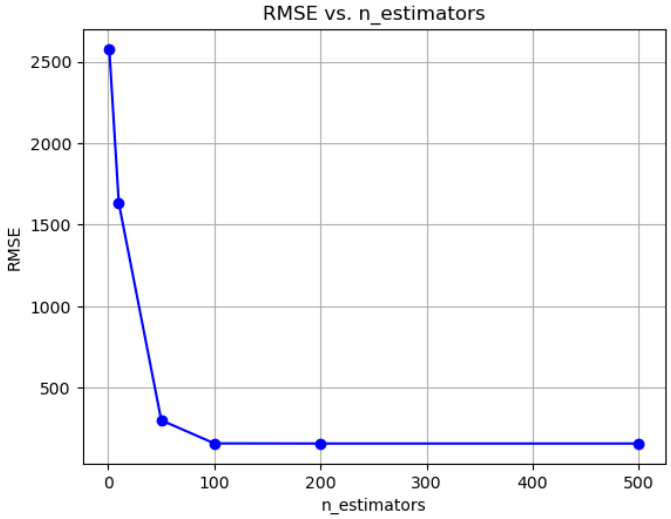

Figura 6.3: RMSE frente a número de árboles de decisión generados

- El número de capas.
- El optimizador.

Para la función de pérdida se ha hecho siempre uso del MSE.

Para poder visualizar los resultados de las distintas combinaciones de hiperparámetros se ha hecho uso de la gráfica que muestra la evolución de la función de pérdida sobre el conjunto de entrenamiento y el conjunto de test. En la Figura 6.4 se muestra la evolución obtenida para el entrenamiento más óptimo encontrado en la predicción a 7 días. En ella se alcanza un punto en el que, sin haber sobreaprendizaje, el conjunto de entrenamiento parece haber alcanzado un valor difícil de mejorar con más iteraciones. Si el conjunto de test empeorara sus resultados al tiempo que el conjunto de entrenamiento los mejorara, podríamos llegar a la conclusión de que hay sobreaprendizaje en nuestro modelo.

6.4. LSTM bidireccional

Para el entrenamiento de este modelo se han calibrado los siguientes hiperparámetros:

- El número de épocas.
- El tamaño de lote.
- El número de neuronas por capa.
- El número de capas.
- El optimizador.

Para la función de pérdida se ha hecho siempre uso del MSE.

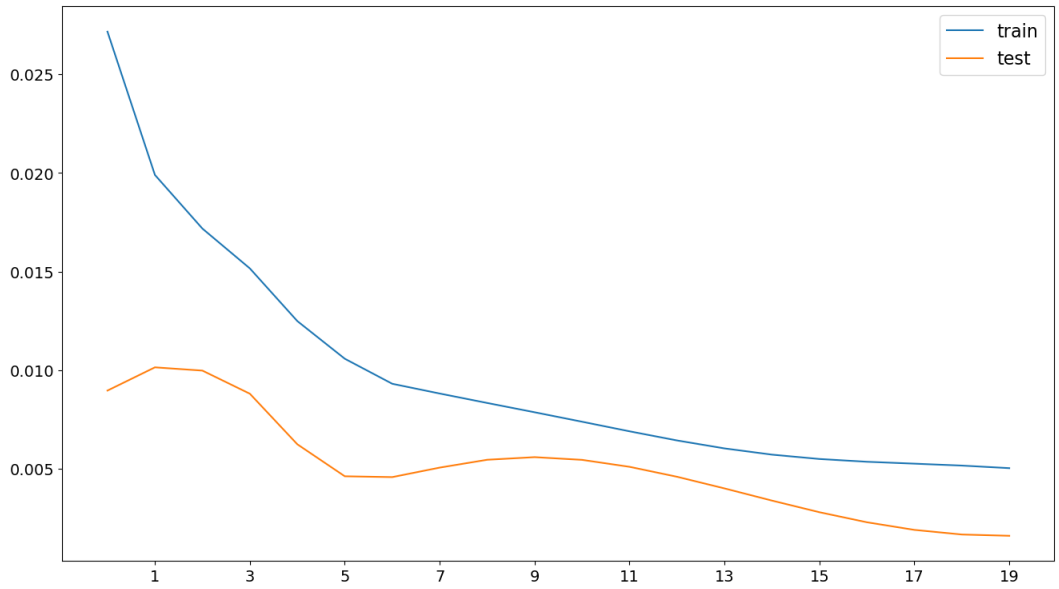

Figura 6.4: Evolución de la función de pérdida a lo largo de las épocas del entrenamiento de un modelo LSTM con una anticipación de 7 días

Al igual que en el caso del modelo LSTM, en el LSTM bidireccional se ha hecho uso de la gráfica de evolución de la función de pérdida para visualizar lo sucedido durante el entrenamiento. Esto nos permite adaptar con mayor precisión los hiperparámetros, especialmente el referente al número de épocas necesarias para alcanzar un aprendizaje óptimo sin que aparezca el problema del sobreaprendizaje. En la Figura 6.6 se muestra la evolución correspondiente al entrenamiento del modelo para predecir a 7 días.

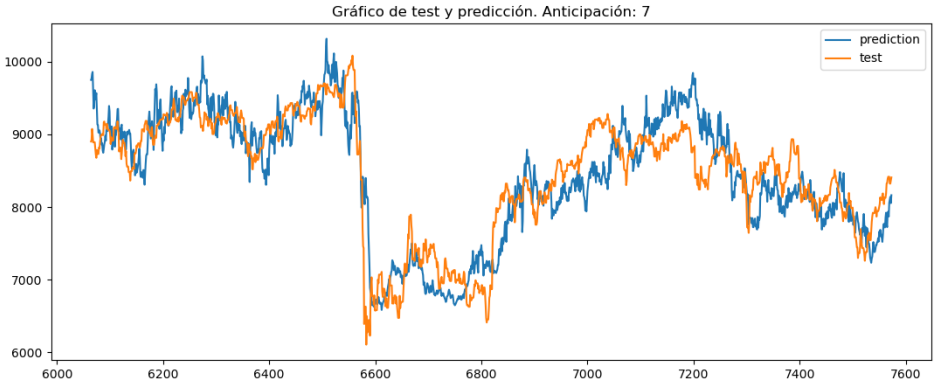

Figura 6.5: Datos predichos por el modelo de LSTM para el IBEX 35 con 7 días de anticipación

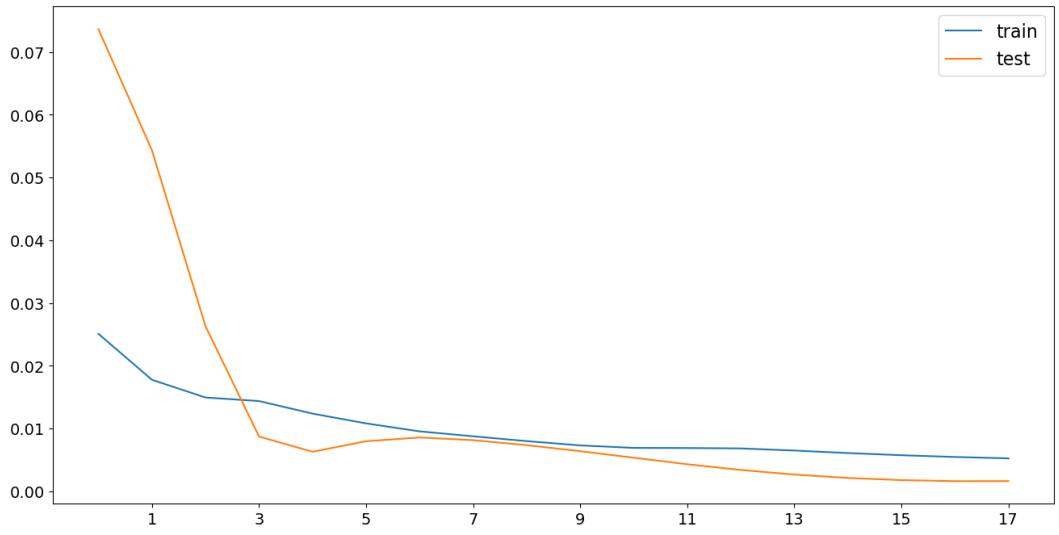

Figura 6.6: Evolución de la función de pérdida a lo largo de las épocas del entrenamiento de un modelo LSTM bidireccional con una anticipación de 7 días

Capítulo 7

Métodos de Evaluación

Tras haber realizado el entrenamiento de todos los modelos, es el momento de compararlos. Para ello precisamos de métodos de evaluación que contrasten el acierto de las predicciones de cada uno de los modelos. El modo más eficaz de comparar objetivamente varios modelos distintos es calculando métricas que sean capaces de representar el error que comete cada uno de ellos. Además, es importante realizar la comparación de los errores sobre el mismo conjunto de datos. De esta manera, en todos los modelos que hemos entrenado se han separado los datos con un conjunto del 80 % para realizar el entrenamiento y un 20 % para comprobar la capacidad de predicción del modelo y calcular las métricas de error. Es importante recordar que la separación de los datos se hace sin desordenarlos, ya que tienen dependencias temporales entre sí.

Dado que los modelos se pueden entrenar para realizar predicciones con un número de días de anticipación variable, es también adecuado realizar comparativas separadas según distintos casos de uso. Por ello, se ha decidido comparar todos los modelos con 1, 7 y 30 días de anticipación de forma individual. De este modo, podemos encontrarnos que algunos modelos sean más idóneos para las predicciones a más corto plazo mientras que otros se apliquen mejor en márgenes de tiempo más amplios.

Existen varias métricas para evaluar el error que comete un modelo al realizar las predicciones. Cada una de ellas presenta un punto de vista distinto y da más o menos importancia a cierto tipo de errores. Por ello es de especial relevancia conocer la manera en que se calculan y como afecta esto al resultado de la métrica. Cabe destacar que, a la hora de aplicarlas sobre los modelos, se hace uso de las implementaciones que ofrece Scikit-learn. A continuación se explican cada una de las métricas que se aplicarán.

7.1. Error cuadrático medio

El error cuadrático medio (MSE por sus siglas en inglés) es probablemente la métrica más usada para medir el error en problemas de regresión. Se caracteriza por penalizar en mayor medida los errores más grandes. Esto significa que si, por ejemplo, tenemos un error de 5 puntos y uno de 10, esta métrica dará mucha más importancia al error de 10 puntos

que al de 5.

La fórmula que determina el cálculo de esta métrica es la que se especifica a continuación:

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2 \tag{7.1}$$

donde $\hat{Y}$ es el vector de predicciones, Y es el vector de los verdaderos valores y n es el número de predicciones realizadas.

7.2. Raíz del error cuadrático medio

La raíz del error cuadrático medio (o RMSE por sus siglas en inglés) es una métrica muy similar al MSE. Se diferencia en que, al aplicar la raíz cuadrada al error cuadrático medio, se obtiene un valor comparable en la escala original del activo. Esto hace más sencilla la percepción del error respecto a los datos que se tienen. Al mismo tiempo, mantiene la característica de penalizar en mayor medida a los errores más grandes.

Su fórmula se presenta a continuación:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2} \tag{7.2}$$

donde $\hat{Y}$ es el vector de predicciones, Y es el vector de los verdaderos valores y n es el número de predicciones realizadas.

7.3. Error absoluto medio

El error absoluto medio (MAE por sus siglas en inglés) destaca por proporcionar una métrica que mide de forma proporcional la importancia de cada error, reuniéndolos en la media de sus valores absolutos. Su valor siempre será menor o igual al de la raíz del error cuadrático medio. Conceptualmente es la métrica más sencilla de las presentadas y el resultado es comparable en la escala original del activo.

Su fórmula resulta tal y como se muestra a continuación:

$$\frac{\sum_{i=1}^{n}|\hat{Y}_i - Y_i|}{n} \tag{7.3}$$

donde $\hat{Y}$ es el vector de predicciones, Y es el vector de los verdaderos valores y n es el número de predicciones realizadas.

7.4. Coeficiente de determinación

El coeficiente de determinación o R^2 (pronunciado R cuadrado) es una medida del error completamente distinta a las anteriores. Puede tener como resultado cualquier valor menor

o igual a 1 y resultará en 0 si las predicciones son siempre la media de los valores reales. Cuanto más cercano a 1, mejor es la predicción y menor es el error medido.

Su fórmula se calcula de la siguiente manera:

$$R^2 = 1 - \frac{\sigma_r^2}{\sigma^2} \tag{7.4}$$

donde σ_r^2 es el error cuadrático medio del modelo (o la varianza residual) y σ^2 es la varianza de las predicciones.

Capítulo 8

Resultados

Para cada uno de los modelos entrenados se han llevado a cabo una serie de experimentos utilizando diversos hiperparametros. A continuación, se analizarán los resultados de los experimentos más relevantes para cada modelo, a través de los distintos métodos de evaluación. Principalmente se hará uso de la medida RSME, al penalizar mayormente los errores más grandes, e ignorando los errores más pequeños, lo que parece deseable para la serie temporal que estamos analizando.

8.1. Random Forest

Para este modelo se han realizado diversos experimentos para 1, 7 y 30 días de anticipación. Los resultados de dichos experimentos son los detallados en las tablas 8.1, 8.2 y 8.3.

ID	**n_estimators**	**max_depth**	**rmse**
1	5	2	755
2	5	5	149
3	5	10	175
4	1	5	189
5	2	5	159
6	3	5	151
7	4	5	149
8	10	5	146
9	15	5	149
10	25	5	149
11	50	5	149
12	75	5	150
13	100	5	149

Tabla 8.1: Resultados con datos desde 2005 hasta 2022 (Anticipación: 1, num_días: 25)

ID	n_estimators	max_depth	rmse
14	100	2	747
16	100	5	307
17	100	10	336
18	1	5	379
19	2	5	368
20	3	5	335
21	5	5	319
22	10	5	316
23	15	5	312
24	25	5	305
25	50	5	309
26	75	5	308
27	100	5	307

Tabla 8.2: Resultados con datos desde 2005 hasta 2022 (Anticipación: 7, num_días: 25)

ID	n_estimators	max_depth	rmse
29	5	2	918
30	5	5	644
31	5	7	717
32	5	10	717
33	1	5	690
34	2	5	684
35	3	5	634
36	5	5	644
37	10	5	762
38	15	5	802
39	25	5	742
40	50	5	778
41	75	5	785
42	100	5	798

Tabla 8.3: Resultados con datos desde 2005 hasta 2022 (Anticipación: 30, num_días: 25)

Se puede observar que, como tendencia general, cuanto mayor es la anticipación peores son los resultados obtenidos. No parece que los resultados sean significativos ya que el activo, de media, entre un día y el siguiente cambia -0.28 puntos. Por lo que un error cuadrático medio (RSME) de 146 puntos (el mejor resultado, en el experimento número ocho) no parece una predicción adecuada.

Dado que el modelo permite una cierta explicabilidad, se pueden obtener las siguientes gráficas, que reafirman dicha hipótesis. En la figura 8.1 se pueden apreciar las diez variables

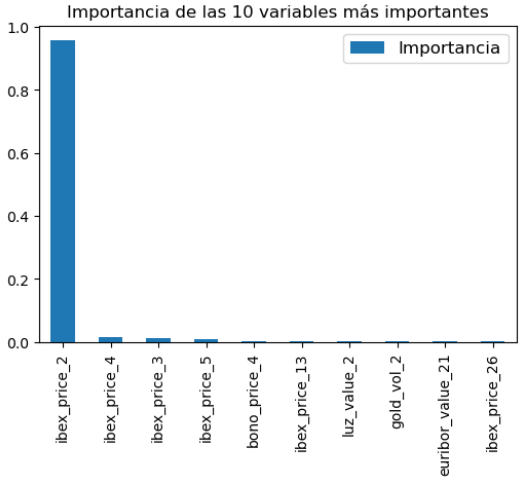

Figura 8.1: 10 variables más importantes para un modelo de Random Forest con anticipación de 1 día

más relevantes para el modelo con una anticipación de 1 día. En la figura 8.2 se aprecia la misma gráfica, para una anticipación de 7 días.

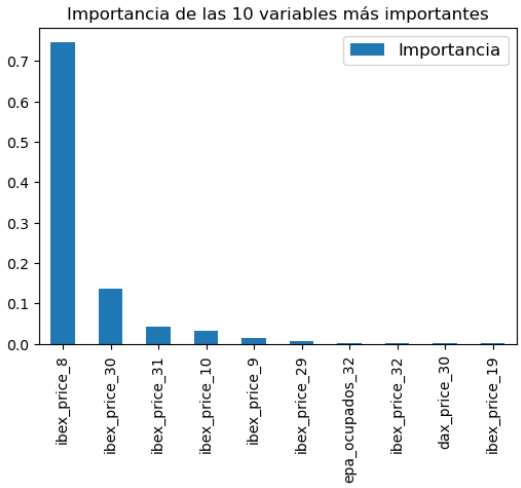

Figura 8.2: 10 variables más importantes para un modelo de Random Forest con anticipación de 7 días

Por tanto, se puede observar que apenas se está haciendo uso de ninguna otra variable que no sea el histórico de valores del IBEX 35. Esto resta utilidad al modelo, ya que lo hace equivalente a realizar predicciones del mismo valor que se ha tenido el día anterior. De hecho, haciendo el experimento de calcular el error cuadrático medio de las predicciones basadas únicamente en dar el valor del día anterior y comparando esta métrica con la obtenida en las predicciones del modelo entrenado, podemos comprobar que el modelo entrenado obtiene peores resultados. Se puede concluir, por tanto, que este modelo de aprendizaje automático no ha sido capaz de encontrar ninguna relación entre las variables de entrada y el activo más allá de que el IBEX 35 a predecir será parecido al último valor

Figura 8.3: Datos predichos por el modelo de Random Forest para el IBEX 35 con 7 días de anticipación

recogido del propio IBEX 35. En la Figura 8.3 se puede observar la tendencia descrita, en la cual los puntos de las predicciones realizadas con 7 días de anticipación sobre el conjunto de datos de evaluación marcan prácticamente los mismos datos que el propio activo marcaba unos días antes.

8.2. XGBoost

Al igual que en el modelo anterior este modelo se ha entrenado para predecir los datos a 25 días con una anticipación de 1, 7 y 30 días. Se han realizado diversos experimentos cuyos resultados estudiaremos a continuación. Para este modelo solo se presentan los mejores resultados en la tabla 8.4, al ser los resultados similares a los del modelo anterior.

ID	Anticipación	n_estimators	learning_rate	early_stopping_rounds	rmse
1	1	1000	0.05	10	158
2	1	1000	0.05	50	158
3	1	1000	0.05	100	158
4	7	200	0.05	10	393
5	7	200	0.125	10	400
6	7	200	0.025	10	412
7	30	200	0.25	10	895

Tabla 8.4: Resultados para el modelo XGBoost con datos desde 2005 hasta 2022 (Anticipación: 1,7 y 30 dias, num_días: 25)

Los mejores resultados, como ya pasaba en el modelo anterior, muestran una escasa habilidad para predecir el activo. Para los mejores resultados (experimentos 1 y 2, anticipación siete días) observamos en la figura 8.4, al igual que el modelo anterior, la variable que más se tiene en cuenta es el propio IBEX35. Sin embargo, para anticipaciones mayores, como se puede observar en la figura 8.5, el modelo otorga mayor relevancia a otros datos, tales como la encuesta de población activa. No tiene, aun así, éxito a la hora de predecir el valor del activo.

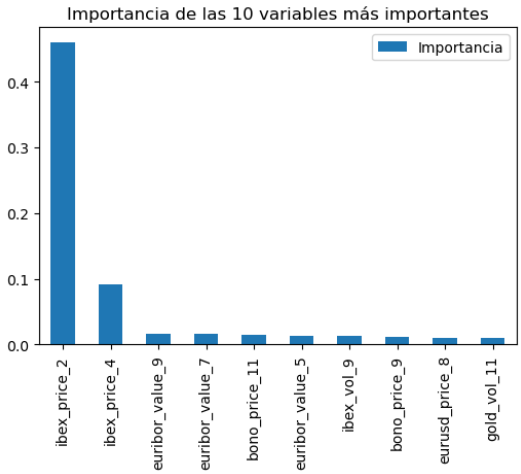

Figura 8.4: 10 variables más importantes para un modelo de XGBoost con anticipación de 1 día

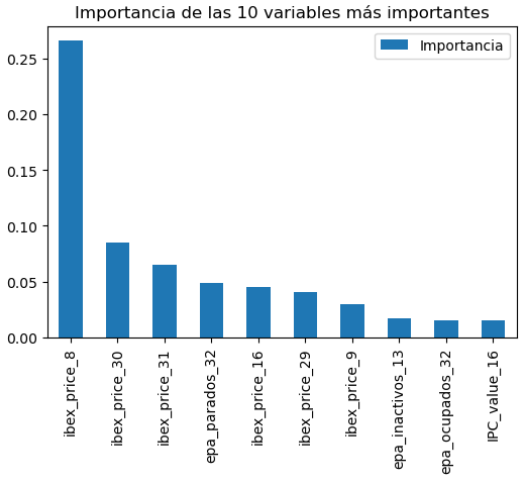

Figura 8.5: 10 variables más importantes para un modelo de XGBoost con anticipación de 7 días

Tras observar estos resultados, cabe esperar que las predicciones realizadas por el modelo sean muy parecidas a los datos más recientes del activo disponibles. Esta hipótesis se puede confirmar tras ver las predicciones sobre el conjunto de datos de evaluación, mostradas en la Figura 8.6. Las predicciones mostradas en esta figura se realizan con una

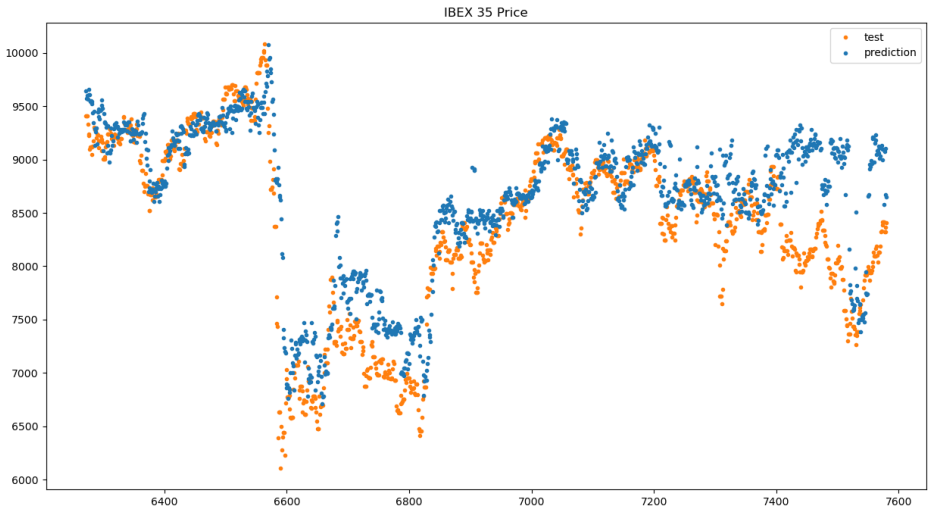

Figura 8.6: Datos predichos por el modelo de XGBoost para el IBEX 35 con 7 días de anticipación

anticipación de 7 días.

Parece por tanto evidente que el modelo no ha podido encontrar relaciones significativas entre las variables y el activo, más allá de que los valores más recientes recogidos del IBEX 35 son similares a los que se pretenden predecir.

8.3. LSTM

Para este modelo se han llevado a cabo diferentes experimentos y se ha llegado a la conclusión de que según los resultados obtenidos, el modelo LSTM tampoco obtiene resultados significativos. Incluso se observa que su rendimiento en la predicción es peor que el de los modelos de Random Forest, a pesar de ser un tipo de modelo más complejo. En la gráfica 8.7 vemos los datos predichos por el modelo, que más tarde compararemos con el modelo de tipo bidireccional.

Sin embargo, al tratarse de un modelo de caja negra, su capacidad de explicar los resultados es muy limitada. Además, debido a su complejidad, resulta difícil determinar si se ha alcanzado o no una aproximación al modelo óptimo. Por lo tanto, no podemos afirmar de manera concluyente que este modelo LSTM es menos adecuado para la predicción financiera.

Tras un análisis más detenido de los resultados, podemos observar que, en términos generales, cuanto más compleja es la red neuronal, más difícil es el proceso de entrenamiento, a medida que aumenta la complejidad de la red, la velocidad de convergencia tiende a disminuir.

Por otro lado, aunque existe la posibilidad de obtener mejores resultados con la red

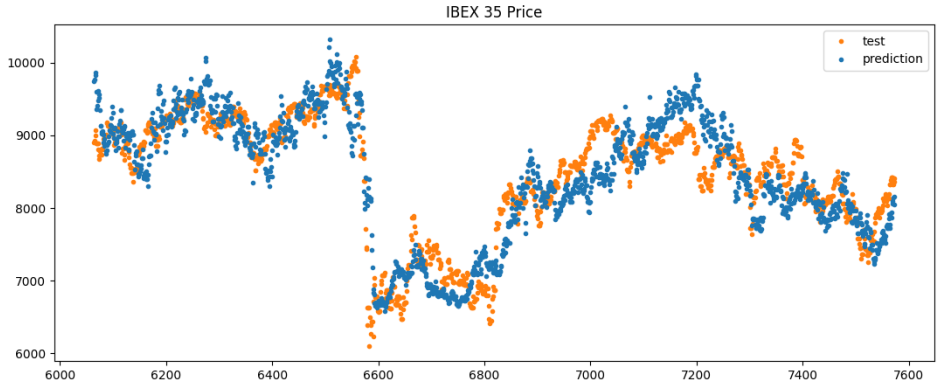

Figura 8.7: Datos predichos por el modelo de LSTM para el IBEX 35 con 7 días de anticipación

compleja, en términos generales se observa una tendencia de que un modelo más simple tiende a adaptarse mejor al problema, ya que no muestran una diferencia significativa en los resultados.

En las tablas 8.5 y 8.6 se puede observar el comportamiento del modelo según los distintos hiperparámetros. Por ejemplo, en los casos 1 y 2 de la primera tabla se comprueba que un valor alto de épocas da lugar a un resultado debido al sobreaprendizaje. Además, se puede ver en ambas que en general el optimizador Adam converge más rápido que el RMSprop. Sin embargo, en determinados casos como en la fila 9 de la tabla 8.5 éste obtiene un mejor comportamiento que el primero.

ID	**epoch**	**batch**	**capas**	**neu**	**loss**	**op**	**mse**	**rmse**	**mae**	**r2**
1	500	32	1	32	mse	adam	350765	592	504	0.4786
2	200	32	1	1	mse	adam	270462	520	401	0.5980
3	20	30	1	1	mse	adam	139975	374	306	0.7919
4	20	30	1	1	mse	RMSprop	410353	640	545	0.3901
5	22	30	1	1	mse	adam	130786	361	293	0.8056
6	8	7	1	1	mse	adam	192743	439	353	0.7135
7	20	30	1	2	mse	adam	182528	427	336	0.7287
8	20	32	1	1	mse	adam	141032	375	305	0.7903
9	30	28	1	1	mse	RMSprop	119116	345	277	0.8229
10	18	28	1	1	mse	adam	123792	351	280	0.8160

Tabla 8.5: Resultados con datos desde 2002 hasta 2022 (Anticipación: 1, num_días: 25)

ID	epoch	batch	capas	neu	loss	op	mse	rmse	mae	r2
1	20	32	1	1	mae	adam	179314	423	338	0.7336
2	350	32	1	16	mae	adam	182866	427	305	0.7283
3	500	64	1	1	mae	adam	909119	953	716	-0.3506
4	500	64	1	7	mae	RMSprop	440008	663	549	0.3463

Tabla 8.6: Resultados con datos desde 2002 hasta 2022 (Anticipación: 7, num_días: 25)

8.4. LSTM Bidireccional

Al observar los resultados, podemos llegar a una conclusión similar respecto al modelo LSTM simple. A pesar de ser una versión más sofisticada, no está obteniendo mejores resultados; sino que generalmente los resultados son peores. Una vez más, esto nos lleva a pensar que, en este contexto, es difícil encontrar una relación buena entre las entradas y la salida. En la gráfica 8.8 comprobamos que los datos predichos son bastante más desacertados que los del modelo LSTM simple mostrados en la figura 8.7.

Se puede ver en las tablas 8.7 y 8.8 el comportamiento del modelo con 1 día y 7 días de anticipación, respectivamente. Cuando se emplea más de una capa es necesario aumentar el número de neuronas de la red neuronal para conseguir un resultado óptimo.

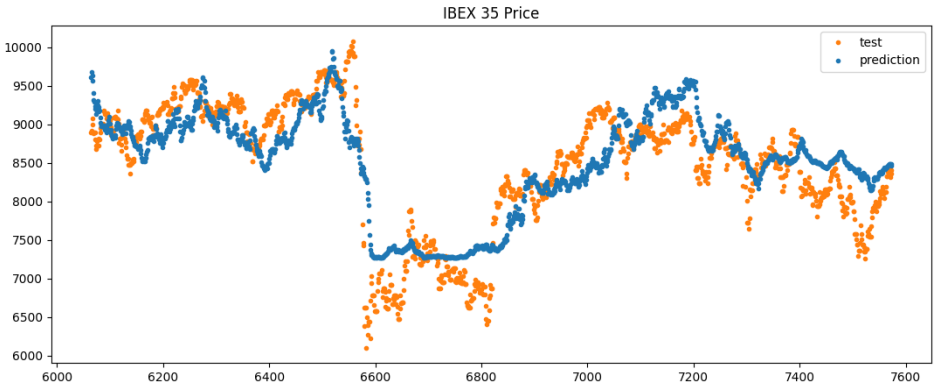

Figura 8.8: Datos predichos por el modelo de LSTM Bidireccional para el IBEX 35 con 7 días de anticipación

8.5. Comparación entre los modelos

En este apartado se tomarán los mejores experimentos realizados entre los cuatro modelos estudiados y se compararán sus resultados, teniendo en cuenta la eficiencia de cada uno de los modelos. Estos quedan representados en las tablas 8.9, 8.10, 8.11 y 8.12.

Salta a la vista que todos los modelos funcionan mejor cuando tienen una anticipación menor, algo que era previsible. También se puede comprobar que las redes neuronales más

ID	epoch	batch	capas	neu	loss	op	mse	rmse	mae	r2
1	35	30	2	12	mse	adam	147061	383	300	0.7814
2	35	60	2	12	mse	adam	117045	342	262	0.8260
3	35	90	2	12	mse	adam	144262	379	283	0.7855
4	20	30	1	1	mse	adam	157022	396	310	0.7666
5	18	30	1	1	mse	adam	154646	393	315	0.7701
6	20	32	1	1	mse	adam	151089	388	309	0.7754
7	18	28	1	1	mse	adam	165161	406	318	0.7545
8	15	28	1	1	mse	adam	142952	378	304	0.7875
9	16	28	1	1	mse	adam	141259	375	301	0.7900

Tabla 8.7: Resultados con datos desde 2002 hasta 2022 (Anticipación: 1, num_días: 25)

ID	epoch	batch	capas	neu	loss	op	mse	rmse	mae	r2
1	30	60	2	16	mse	adam	214127	462	355	0.6819
2	30	30	2	12	mse	adam	196758	443	352	0.708
3	40	60	2	28	mse	adam	236507	486	390	0.6486
4	18	30	1	1	mse	adam	181786	426	339	0.7299
5	20	30	1	1	mse	adam	205051	452	350	0.6954
6	20	32	1	1	mse	adam	239238	489	388	0.6446
7	15	32	1	1	mse	adam	236623	486	389	0.6485
8	18	32	1	1	mse	adam	181055	425	340	0.7310

Tabla 8.8: Resultados con datos desde 2002 hasta 2022 (Anticipación: 7, num_días: 25)

simples funcionan mejor para nuestro problema. Se podría razonar que el proyecto no requiere modelos con gran catnidad de variables e hiperparámetros para conseguir resultados óptimos. El modelo Random Forest consigue un valor de RMSE notablemente menor al resto y las dos variantes de LSTM, simple y bidireccional, no se comportan mejor que las demás pese a ser más complejas.

Además, hemos observado diferencias a la hora de realizar los distintos experimentos entre los últimos dos modelos. El LSTM bidireccional es más complejo, y el proceso de entrenamiento resulta más tedioso. Es necesario probar una gran cantidad de valores de hiperparámetros hasta encontrar algún resultado óptimo. Sin embargo, este proceso merece la pena en cierto modo ya que la variante bidireccional consigue un mejor rendimiento frente al LSTM simple.

anticipación	n_estimators	max_depth	rmse
1	10	5	146
7	25	5	305
30	3	5	634

Tabla 8.9: Resultados óptimos del modelo Random Forest (num_días: 25)

anticipación	n_estimators	learning_rate	early_stopping_rounds	rmse
1	1000	0.05	10	158
1	1000	0.05	50	158
1	1000	0.05	100	158
1	500	0.05	10	158
1	200	0.05	10	158
7	200	0.05	10	393
30	200	0.25	10	895

Tabla 8.10: Resultados óptimos del modelo XGBoost (num_días: 25)

anticipación	epoch	batch	capas	neu	loss	op	mse	rmse	mae	r2
1	30	28	1	1	mse	RMSprop	119116	345	277	0.8229
1	18	28	1	1	mse	Adam	123792	351	280	0.8160
7	20	32	1	1	mse	Adam	179314	423	338	0.7336
25	30	32	1	1	mse	Adam	308625	555	416	0.5424

Tabla 8.11: Resultados óptimos del modelo LSTM (num_días: 25)

anticipación	epoch	batch	capas	neu	loss	op	mse	rmse	mae	r2
1	35	60	2	12	mse	Adam	117045	342	262	0.8260
7	18	32	1	1	mse	Adam	181055	425	340	0.7310
25	30	90	2	16	mse	Adam	279614	528	355	0.5854

Tabla 8.12: Resultados óptimos del modelo LSTM Bidireccional (num_días: 25)

Capítulo 9

Conclusiones y Trabajo Futuro

Este trabajo pretende ofrecer una primera aproximación a la predicción de series temporales mediante el uso de modelos de aprendizaje automático. Para ello se ha escogido una serie temporal tan compleja como es la compuesta por los valores que va tomando el IBEX 35. Una vez escogido el activo, se han seleccionado las variables que, a nuestro juicio, más afectaban a la evolución de la serie temporal y se han entrenado cuatro modelos distintos: Random Forest, XGBoost, LSTM y LSTM bidireccional. Hemos podido ver como ninguno de ellos ha sido capaz de realizar una predicción certera debido a la excesiva complejidad del problema propuesto, obteniendo unos mejores resultados en los modelos más simples (Random Forest y XGBoost). Por otro lado se ha constatado que, al menos en este caso, la bidireccionalidad en las redes LSTM no supone una mejora significativa en las predicciones del modelo.

Se ha observado también que uno de los problemas que se afrontan a la hora de predecir un activo financiero tan complejo es la dificultad para incorporar, directa o indirectamente, todas la variables que afectan de forma relevante a los mercados financieros. El ejemplo más claro de esto es el de los anuncios de medidas económicas, aunque no es el único. Este problema podría no estar presente en otros tipos de series temporales, lo que deja la puerta abierta a la puesta a prueba de los modelos aquí vistos en otros contextos más simples.

Por tanto, una de las posibles expansiones a nuestro trabajo consiste en la prueba de estos modelos en series temporales de menor complejidad. Sería relevante averiguar si las redes LSTM son capaces de superar a modelos más simples, como los analizados en el presente trabajo, bajo condiciones de menor complejidad. Esto podría suceder en el caso de que hubiera patrones implícitos que solo puedan ser percibidos por modelos con mayor capacidad de abstracción.

Otra posible vía de trabajo sobre este proyecto es la de poner a prueba otros tipos de modelos sobre este mismo problema. En especial cabe destacar el modelo ARIMA, que es muy usado en el mundo de la predicción de series temporales y no ha llegado a ser usado en este trabajo. Tras ello, se podrían comparar los resultados obtenidos obteniendo un análisis más completo.

Capítulo 10

Aportaciones individuales

10.1. Pablo Lozano Martín

Inicialmente, establecí un repositorio en Github para el proyecto y organicé la colaboración invitando a todos los compañeros a unirse. Esto proporcionó una plataforma centralizada para el desarrollo y seguimiento del trabajo.

Más tarde, mis compañeros y yo dimos paso a la selección de los datos, que serían la base de nuestro estudio. Trabajamos juntos para garantizar que los datos elegidos fueran apropiados y relevantes para nuestro objetivo de investigación. Dividimos las variables de manera equitativa y cada compañero obtuvo los distintos datos de cada una de ellas en formato CSV (en mi caso la información del valor Dolar-Euro, el IPC y los tipos de interés del Eurosistema, del mercado primario y del secundario). Una vez completado, fue el turno del preprocesado de los datos, lo que incluyó la limpieza, transformación y preparación de estos para su posterior análisis. Este paso es fundamental para obtener resultados confiables y significativos.

Al mismo tiempo, nos dedicamos a redactar la memoria del proyecto, una parte importante de nuestra investigación. Mi papel fue destacado en la fase inicial de la memoria, donde finalicé la introducción. Esta sección es crucial para establecer el contexto y los objetivos de nuestra investigación. También colaboré activamente en la redacción de la memoria en su totalidad, lo cual implicó un trabajo minucioso en varios aspectos. Primero, contribuí a organizar la estructura del documento, asegurando una secuencia lógica y coherente de los contenidos. Además, enriquecí la memoria incluyendo hipervínculos a referencias externas citadas en nuestra bibliografía, mejorando así la profundidad y credibilidad de nuestro trabajo. También desempeñé un papel esencial en la creación de tablas y gráficos que complementaron la presentación de nuestros hallazgos y mejoraron la claridad y visualización de la información, facilitando su comprensión para nuestro equipo y futuros lectores e investigadores interesados en nuestro estudio.

Llevé a cabo un análisis detallado de los modelos LSTM y LSTM bidireccional que implementamos. A partir de los experimentos realizados, los resultados obtenidos quedaron reflejados en archivos Excel. Extraje los casos más destacables y presenté los resultados

de manera clara y concisa en forma de tablas diferenciando entre los datos calculados con una anticipación de 1, 7 y 30 días. Además, redacté las conclusiones específicas para estos modelos y contribuí al apartado de análisis y conclusiones finales del proyecto. En esta sección del estudio trabajé en estrecha colaboración con mis compañeros para analizar los modelos Random Forest y XGBoost, contribuyendo así a una evaluación completa de diferentes enfoques.

En los días finales detecté y corregí diversas erratas y errores de compilación en la memoria del proyecto, asegurándome de que la documentación estuviera libre de imperfecciones. En la última parte de resultados realicé varios cambios para mejorar la cohesión del trabajo y varios pequeños apartados en uno solo. De esta forma, las conclusiones tomadas a partir del análisis de los cuatro modelos quedaron más congruentes. También realicé un esfuerzo importante para asegurar que las referencias a las figuras estuvieran correctamente etiquetadas y enlazadas.

Es importante resaltar que todas estas contribuciones se realizaron en un entorno de trabajo colaborativo, donde el intercambio de ideas y el apoyo mutuo fueron fundamentales para alcanzar nuestros objetivos. Desde la creación del repositorio en GitHub hasta la cuidadosa selección de datos, el exhaustivo preprocesamiento de la información, el análisis detallado de los modelos y la redacción de la memoria, cada etapa fue abordada con un gran compromiso y coordinación entre los miembros del equipo.

10.2. Jaime Pastrana García

El primer paso dado en la elaboración de este proyecto consistió en la elección y obtención de los datos. La elección de tanto el activo (el IBEX 35) como las variables relacionadas fue una decisión tomada entre todos y consensuada tras discutirlo entre nosotros. Posteriormente se pasó a la obtención de dichos datos. Esta tarea se dividió entre los cuatro integrantes, correspondiéndome la descarga de los datos del DAX 30, la EPA y el precio del barril de Brent en formato CSV.

La siguiente tarea consistía en preprocesar estos datos, tarea en la que realicé una primera aproximación en python que más tarde generalizarían y perfeccionarían mis compañeros. En ella se eliminaron las datos que no eran útiles o eran anómalos, se estimaron los puntos de los que no se tenía información y se unificó el formato en el que se presentaban los datos.

Llegado este punto se comenzó la redacción de la memoria al tiempo que se comenzaron a entrenar los modelos. En cuanto a la memoria se refiere, se repartieron las responsabilidades en relación a los intereses de cada uno de los miembros del grupo. En mi caso mis responsabilidades principales fueron las de elaborar el capítulo correspondiente al análisis descriptivo y la explicación de la teoría y experimentación de los modelos de Random Forest y XGBoost. En el análisis descriptivo del activo tuve que elaborar en primer lugar varias gráficas y datos relevantes sobre los que basar mis razonamientos. Con estos datos redacté el capítulo, que fue revisado por mis compañeros. Posteriormente, entendiendo el contexto

en que se daban los datos, proseguimos con el desarrollo de los modelos. Mi implicación en este punto estuvo especialmente ligada a los modelos de Random Forest y XGBoost. En cualquier caso, todos colaboramos en los cuatro modelos con el objetivo de obtener unos resultados relevantes y una explicación clara de los modelos y su uso.

Finalmente, participé en la redacción de una breve explicación de los métodos de evaluación y redacté el capítulo sobre las conclusiones y el trabajo futuro. En este último capítulo sinteticé las ideas que habíamos desarrollado el grupo en su conjunto.

Lo aquí escrito refleja en esencia las tareas de las que he sido el principal responsable. Sin embargo, cabe destacar la colaboración e implicación de todos los miembros del grupo en todas las tareas, en mayor o menor medida según correspondiera. Esto último nos ha resultado esencial para mantener una visión de conjunto del trabajo realizado.

10.3. Pablo Magno Pezo Ortiz

Mi contribución al proyecto, como todas las de mi compañeros fue fundamental para su éxito y se llevó a cabo en varias etapas. Durante toda la elaboración del presente trabajo he asumido un rol activo en la toma de decisiones y en la ejecución de tareas. A continuación, detallaré mis contribuciones.

Comenzamos el proyecto con la elección del activo y las variables con las que buscaríamos correlaciones. Esta fase inicial requería de planificación y consideración. Participé activamente en esta parte del proceso, en la que mantuvimos diversas reuniones con la directora del proyecto para asegurarnos de que nuestras decisiones fueran fundamentadas y apropiadas. Finalmente, y tras considerar otros activos como el Dow Jones o el Eurostax, decidimos trabajar con el índice IBEX35 como nuestro activo principal.

En cuanto a la selección de variables, organizamos varias reuniones de discusión con el equipo para determinar cuáles serían las más óptimas. Durante aquellas reuniones, propuse una lista inicial de variables y, para garantizar su idoneidad, contactamos nuevamente con la profesora para verificar si necesitaban una justificación más detallada. Aclarado que no era necesario, procedimos a asignar a cada miembro del equipo la tarea de buscar y descargar los datos necesarios. En mi caso, me encargué de obtener los datos del Euribor y el precio del oro.

A su vez, creé una carpeta en Google Drive para centralizar toda la información del proyecto, donde también se creo un "log"de reuniones con la directora.

Elegidas las variables que íbamos a estudiar, y descargados sus dataset en formato .csv, participé activamente en la fase de preprocesamiento de datos, aunque no estuve involucrado en la etapa más preliminar. Durante esta etapa, detectamos que dos de las variables elegidas no eran las más apropiadas, propuse a mis compañeros sustituirlas, encargándome también de descargar sus datos. Una vez corregidas dichas variables, me hice cargo de escribir los cuadernos Python empleados para tratar los datos, así como de las decisiones técnicas a tomar. Esto incluye el procesado y limpiado de todos los dataset iniciales, así como su combinación en el dataset final.

También me encargué, posteriormente, de parte del análisis de los resultados obtenidos. En concreto, me encargué de analizar los resultados de los modelos "XGBoostz RandomForest". Esto conllevó cambios sustanciales en el capitulo de entrenamiento, a fin de evitar duplicidades.

En la parte de la memoria, me he encargado de redactar la versión inicial de la introducción, así como la que se presenta en este trabajo, que ha contado con la revisión y colaboración de otros compañeros. También he redactado el resumen, así como el capítulo de preprocesado, y el análisis final de los modelos "XGBoostz RandomForest". También me he encargado de realizar las traducciones al inglés. Todos estos avances fueron luego detallados en la memoria, en el capitulo de preprocesado, que yo me encargué de redactar.

Quisiera por último recalcar el buen clima de trabajo, cooperación y organización en general que hemos mantenido como equipo en todo momento. Fomentando la cooperación y el trabajo en equipo.

10.4. Jun Qiu

Tras la primera fase de nuestro proyecto, que consistió en la elección del activo y sus variables relacionadas con la discusión global, mi tarea fue la extracción de datos para el precio de la luz y el precio del bono español a diez años en formato CSV.

Posteriormente, durante la fase de procesamiento de datos, trabajé en colaboración con mis compañeros para perfeccionar el código que ya estaba en desarrollo, corrigiendo pequeños errores y asegurándonos de que tuviéramos un formato uniforme en nuestros datos.

En cuanto a la memoria, de acuerdo con la distribución de responsabilidades acordada, me encargué de desarrollar los apartados de los modelos LSTM y LSTM bidireccionales. Durante este proceso, colaboré estrechamente con mis compañeros para obtener una visión global de los datos y discutir qué gráficas serían más representativas, además de realizar revisiones detalladas.

Es importante destacar que a lo largo de todo el proyecto, la colaboración en equipo desempeñó un papel fundamental. Mantuvimos discusiones regulares para abordar temas clave y revisar nuestros enfoques individuales. Esta colaboración constante fue esencial para garantizar que nuestro proyecto fuera coherente y que pudiéramos abordarlo de manera efectiva.

Chapter 11

Introduction

Since the 1950s, when Alan Turing laid the foundations of computing, scientists and mathematicians began to realistically consider whether it was possible to create a machine that could mimic, or even surpass, human intelligence. In the subsequent years, notable developments occurred, such as the invention in 1951 of the junction transistor by William Shockley, which allowed the field of computer science to advance exponentially.

This exponential advancement can be exemplified by Moore's Law. Enunciated by Gordon E. Moore in 1965, it empirically states that the number of transistors in a microprocessor will double every two years. While not a law in the strict sense of the word, its prediction has held true in the decades that followed. However, recent trends suggest that its validity is nearing its end.

With the advent of the Internet, personal computers, and subsequently smartphones, computing has become a part of the lives of a significant portion of the global population. Currently, approximately 64% of the population uses the Internet and 68% use mobile phones Datareportal (2023).

These technological advancements have enabled the theoretical and practical development of the mentioned concept, which gave rise to the field of artificial intelligence. This term was coined in 1956 during a conference at Dartmouth, organized by John McCarthy, who later received the prestigious Turing Award, the highest distinction in the field of computer science. In addition to its theoretical impact, these advancements have facilitated the integration of artificial intelligence into the daily lives of millions of people.

The field of artificial intelligence (AI) employs multiple techniques to attempt to mimic human thinking. In this study, we will utilize machine learning, specifically supervised learning. The main objective of this branch is to develop techniques that enable computers to "learn." In other words, the aim is to allow machines to perform tasks for which they haven't been explicitly programmed and to extract relevant relationships and conclusions from a set of data.

This branch has connections to statistics as it relies on the analysis of various data. However, from a computer science perspective, there is a concern for the computational complexity of problems. In other words, it involves the asymptotic measure of the time it

takes to solve a specific problem. There are various types of algorithms, such as supervised learning, unsupervised learning, reinforcement learning, etc.

In this study, we will focus on supervised learning, which enables the prediction of the value of a continuous variable, in this case, the IBEX 35, based on a series of data.

The IBEX 35 is the benchmark index in the Spanish stock market. It encompasses the market capitalization of thirty-five Spanish companies listed on the Spanish Stock Exchange Interconnection System (SIBE) across the four Spanish exchanges: Madrid, Barcelona, Bilbao, and Valencia. The companies listed in this index are not necessarily the largest but rather those that best fulfill the criteria of market capitalization, liquidity, and traded volume. These companies are selected by the advisory committee of the IBEX, which meets twice a year. As the index is the reference value of the Spanish stock market, it is used as an indicator of the overall state of the Spanish economy. Therefore, it is sensitive to various social, political, and economic events in the country and has an impact on other international stock market indices.

Consequently, the motivation of this study will be to select various economic indicators that might impact the index and explore how machine learning techniques can be utilized to predict the value of the IBEX 35.

11.1. Objectives

The main objective of this work is to apply different artificial intelligence models to predict the value of the IBEX 35 index, comparing which ones do so more effectively using various measures to gauge prediction accuracy. The objectives can be summarized as follows:

- Obtaining and processing data from the selected variables.
- Conducting a comprehensive analysis of the obtained dataset.
- Training the chosen models.
- Analyzing the results obtained from different models, using various metrics for this purpose.

11.2. Project plan

For the development of this work, a team consisting of four individuals has been involved. The aim has been to organize the work evenly, tackling the stages of development in an agile manner. In this vein, extensive preliminary analyses have been dispensed with, and instead, the focus has been on writing this report and obtaining results from the outset, iteratively improving them. Throughout the course of the work's development, multiple meetings have been held with the supervisor, during which various aspects and ways to enhance shortcomings have been discussed.

Given the nature of the project, certain tasks are strictly dependent on others to be completed. For instance, to analyze the obtained results, it is necessary to have processed all the data and trained the models beforehand.

The various tasks have been organized temporally according to the following Gantt chart.

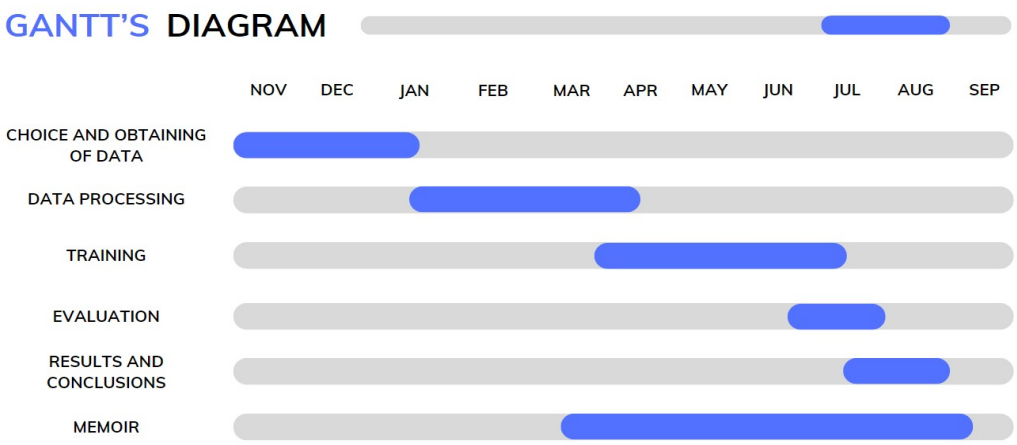

Figure 11.1: Project plan

Chapter 12

Conclusions and Future Work

This work aims to provide an initial approach to time series prediction using machine learning models. To do this, we have chosen a time series as complex as the one composed of the values taken by the IBEX 35. After selecting the asset, we have chosen the variables that, in our opinion, had the most significant impact on the evolution of the time series, and four different models were trained: Random Forest, XGBoost, LSTM, and Bidirectional LSTM. We have observed that none of them has been able to make accurate predictions due to the excessive complexity of the proposed problem, with better results obtained in the simpler models (Random Forest and XGBoost). Furthermore, it has been noted that, at least in this case, bidirectionality in LSTM networks does not lead to a significant improvement in model predictions.

It has also been observed that one of the challenges when predicting such a complex financial asset is the difficulty in directly or indirectly incorporating all the variables that significantly affect financial markets. The clearest example of this is economic policy announcements, although it is not the only one. This problem may not be present in other types of time series, leaving the door open to testing the models seen here in simpler contexts.

Therefore, one possible extension of our work is to test these models on less complex time series. It would be relevant to find out if LSTM networks can outperform simpler models, as analyzed in this work, under conditions of lower complexity. This could happen if there were implicit patterns that can only be perceived by models with a higher level of abstraction.

Another possible avenue for further work on this project is to test other types of models on the same problem. In particular, the ARIMA model, which is widely used in the field of time series prediction, has not been used in this work. Afterward, the results obtained could be compared to provide a more comprehensive analysis.9.

Bibliografía

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. y Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015. Software available from tensorflow.org.

Chen, T. y Guestrin, C. Xgboost: A scalable tree boosting system. *CoRR*, vol. abs/1603.02754, 2016.

Datareportal. Datareportal. `https://datareportal.com/global-digital-overview`, 2023.

Frederickson, B. `http://www.benfrederickson.com/numerical-optimization/`, 2023.

Git. Git. `https://git-scm.com/`, 2023.

Github. Github. `https://github.com/`, 2023.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C. y Oliphant, T. E. Array programming with NumPy. *Nature*, vol. 585(7825), páginas 357–362, 2020.

Hochreiter, S. y Schmidhuber, J. `https://www.bioinf.jku.at/publications/older/2604.pdf`, 1997.

Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, vol. 9(3), páginas 90–95, 2007.

Investing. Investing. `https://uk.investing.com/`, 2023.

Joblib. Página oficial de la librería de Python joblib. 2023. https://joblib.readthedocs.io/en/stable/#(15/06/23).

Kaggle. Kaggle. https://www.kaggle.com/, 2023.

Kalita, D. https://www.analyticsvidhya.com/blog/2022/03/an-overview-on-long-short-term-memory-lstm/, 2022.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S. y Willing, C. Jupyter notebooks – a publishing format for reproducible computational workflows. En *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (editado por F. Loizides y B. Schmidt), páginas 87 – 90. IOS Press, 2016.

Melcher, K. https://www.knime.com/blog/a-friendly-introduction-to-deep-neural-networks, 2021.

Olah, C. https://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015a.

Olah, C. http://colah.github.io/posts/2015-09-NN-Types-FP/, 2015b.

Pandas. Página oficial de la librería de Python Pandas. 2023. https://pandas.pydata.org/(15/06/23).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. y Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, vol. 12, páginas 2825–2830, 2011.

Sirakorn. https://commons.wikimedia.org/wiki/File:Ensemble_Bagging.svg, 2020a.

Sirakorn. https://commons.wikimedia.org/wiki/File:Ensemble_Boosting.svg, 2020b.

Thi. https://dinhanhthi.com/decision-tree-regression/, 2020.

Van Rossum, G. y Drake, F. L. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.