ICAI

PREDICCIÓN DE PRECIOS INTRADÍA DE MERCADOS FINANCIEROS USANDO NOTICIAS

Clave: 201909222

MADRID | Junio de 2024

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
Predicción de Precios Intradía de Mercados Financieros usando Noticias
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2023/24 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos.
El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido
tomada de otros documentos está debidamente referenciada.

Fdo.: Jaime de Clemente Fernández-Picazo
Fecha: 21/ 06/ 2024

Autorizada la entrega del proyecto
EL DIRECTOR DEL PROYECTO

Fdo.: Dr. Carlos Bellón Núñez-Mera

Agradecimientos

A mis padres,
que me han educado en el esfuerzo y el mérito y no han escatimado nunca en mi formación.
A mis hermanos y amigos,
que me han acompañado durante estos años en los mejores momentos y en los más duros.
A mis abuelos, maestros,
que me enseñaron desde pequeño el valor del conocimiento.
Al Señor,
que desde Su Casa en lo Alto me ha dado el tesón y la fuerza que me faltaba en mis momentos de flaqueza.
A todos ellos, porque sin ellos no estaría aquí ni habría acabado esta gran fase que es la universidad.

PREDICCIÓN DE PRECIOS INTRADÍA DE MERCADOS FINANCIEROS USANDO NOTICIAS

Autor: de Clemente Fernández-Picazo, Jaime.
Director: Bellón Núñez-Mera, Carlos

RESUMEN DEL PROYECTO

Este proyecto contiene una propuesta de funcionamiento de un sistema para la predicción de precios intradía de activos en mercados cotizados utilizando datos históricos y noticias. Para su creación se utilizan modelos de aprendizaje automático, en concreto redes neuronales. Con el fin de probar este sistema, se expone como caso de uso la predicción realizada para diez compañías pertenecientes al índice S&P 500.

Palabras clave: Mercados financieros, compañías cotizadas, precios intradía, Machine Learning, Aprendizaje Automático, redes neuronales, noticias, procesamiento del lenguaje natural, NLP, análisis de sentimiento

1. **Introducción**

   El presente proyecto se inscribe en un contexto de desarrollo exponencial de las tecnologías, especialmente en áreas como el Aprendizaje Profundo (Deep Learning), el Procesamiento del Lenguaje Natural (Natural Language Processing, NLP) y el Análisis de Sentimientos. Estas técnicas, fundamentales en la inteligencia artificial moderna, han sido aplicadas en múltiples campos y suponen un gran potencial para resolver problemas complejos sin intervención humana.

   En concreto, este proyecto tiene como objetivo aplicar dichas técnicas para predecir precios intradía en mercados financieros utilizando noticias, lo cual no solo tiene relevancia académica, sino también, potencialmente, un notable impacto económico.

2. **Definición del Proyecto**

   El proyecto trata de juntar dos líneas de investigación sobre las que se ha desarrollado una gran cantidad de trabajo, como son la predicción del precio de activos y el análisis de sentimiento de noticias. El trabajo pretende investigar si, aplicando técnicas de

aprendizaje profundo a estos dos campos, es posible realizar buenas predicciones. El proyecto se ha planificado considerando restricciones temporales y económicas, con una metodología estructurada que abarca desde la recopilación de datos hasta el desarrollo de modelos y la evaluación de resultados.

3. Descripción del Sistema

El sistema desarrollado se basa en dos modelos principales: un modelo base y un modelo de noticias. El modelo base utiliza técnicas de análisis técnico para predecir la evolución de los precios basándose en datos históricos de cotización. Este modelo se complementa con un modelo de noticias que trata de corregir los errores del modelo base mediante el análisis de noticias financieras. La integración de ambos modelos se realiza sumando las predicciones del modelo base y el ajuste proporcionado por el modelo de noticias, logrando así una predicción conjunta más precisa.

El desarrollo del sistema implicó la utilización de tecnologías avanzadas como TensorFlow y Keras para la construcción de los modelos de aprendizaje profundo, así como técnicas de *webscraping* para la recolección de datos de noticias. Como paso previo al desarrollo de los modelos, el preprocesamiento de datos incluyó la normalización de precios históricos y la conversión de textos de noticias en secuencias numéricas interpretables por los modelos. La implementación final requirió de recursos computacionales significativos, especialmente para el entrenamiento del modelo de noticias, que necesitó el uso de GPU para manejar su complejidad.

4. Resultados

Los resultados del proyecto se presentan en tres partes: los resultados del modelo base, los del modelo de noticias y la predicción conjunta. El modelo base mostró una capacidad significativa para predecir tendencias a corto plazo en los precios intradía utilizando datos históricos. Sin embargo, al contrario de lo que cabía esperar, su precisión no se vio mejorada al integrar las predicciones del modelo de noticias, que trataba de ajustar los errores residuales del modelo base basándose en el análisis del sentimiento de las noticias financieras. Este hecho es significativo porque, en la predicción a corto plazo presentada, resulta necesario ajustar realmente los resultados, por lo que esta falta de mejoría resulta muy relevante.

5. Conclusiones

El proyecto concluye que la utilización de técnicas de aprendizaje profundo y procesamiento del lenguaje natural para la predicción de precios intradía en mercados financieros podría, con más investigación y el desarrollo de nuevos trabajos, ser capaz de llegar a ser efectiva. Sin embargo, las limitaciones sufridas en la compleción del trabajo impidieron explotar esto para sacarle todo su potencial. En cualquier caso, la integración de modelos basados en análisis técnico y noticias permite capturar tanto las tendencias históricas como el impacto inmediato de eventos, proporcionando una herramienta poderosa para la toma de decisiones en el ámbito financiero. Dado que el proyecto enfrentó limitaciones de datos y recursos computacionales, los resultados sugieren que con mayores recursos y datos más extensos, las predicciones tienen potencial para poder llegar a mejorarse. Futuras investigaciones podrían enfocarse en la ampliación del modelo para incluir más activos financieros y en la optimización de los procesos de recolección y análisis de datos, así como en la inclusión de más datos que puedan hacer al modelo más generalista y apliccable.

INTRADAY PRICE PREDICTION IN FINANCIAL MARKETS USING NEWS

Author: de Clemente Fernández-Picazo, Jaime.
Supervisor: Bellón Núñez-Mera, Carlos

ABSTRACT

This Project contains a functional proposal for the development of a system to predict intraday prices of public assets using historical data and news articles about them. In order for this system to be created, machine learning models are used, more precisely neural networks. To prove the functioning of the system, a use case is exposed with the prediction for ten companies that are traded in the S&P 500 index.

Keywords: Financial markets, publicly traded companies, intraday prices, machine learning, neural networks, news, natural language processing, NLP, sentiment analysis

1. Introduction

This project is set in a context of exponential development of technologies, especially in areas such as Deep Learning, Natural Language Processing (NLP), and Sentiment Analysis. These techniques, fundamental in modern artificial intelligence, have been applied in multiple fields and hold great potential for solving complex problems without human intervention.

Specifically, this project aims to apply these techniques to predict intraday prices in financial markets using news, which not only has academic relevance but also potentially a significant economic impact.

2. Project Definition

The project seeks to integrate two well-established lines of research, which are asset price prediction and sentiment analysis of news articles. This study aims to explore the feasibility of making accurate predictions by applying deep learning techniques to these domains. The project has been meticulously planned, taking into account

temporal and financial constraints, and follows a structured methodology that spans from data collection to model development and performance evaluation.

3. System Description

The developed system is based on two main models: a base model and a news model. The base model uses technical analysis techniques to predict price trends based on historical quotation data. This model is complemented by a news model that attempts to correct the base model's errors through the analysis of financial news. The integration of both models is achieved by summing the predictions of the base model and the adjustment provided by the news model, thus achieving a more accurate joint prediction.

The system development involved the use of advanced technologies such as TensorFlow and Keras for the construction of deep learning models, as well as web scraping techniques for collecting news data. As a preliminary step to model development, data preprocessing included the normalization of historical prices and the conversion of news texts into numerical sequences interpretable by the models. The final implementation required significant computational resources, especially for training the news model, which needed the use of GPUs to handle its complexity.

4. Results

The project's results are presented in three parts: the baseline model results, the news model results, and the combined prediction. The baseline model demonstrated significant capability in predicting short-term trends in intraday prices using historical data. However, contrary to expectations, its accuracy did not improve when integrating the predictions of the news model, which attempted to adjust the baseline model's residual errors based on sentiment analysis of financial news. This is significant because, in the short-term prediction presented, it is necessary to truly adjust the results, making this lack of improvement highly relevant.

5. Conclusions

The project concludes that the use of deep learning techniques and natural language processing for intraday price prediction in financial markets could, with further

research and development, become effective. However, the limitations encountered during the project prevented fully exploiting its potential. Nonetheless, the integration of models based on technical analysis and news allows for capturing both historical trends and the immediate impact of events, providing a powerful tool for decision-making in the financial sector. Given the project's constraints on data and computational resources, the results suggest that with more extensive resources and data, predictions have the potential to improve. Future research could focus on expanding the model to include more financial assets and optimizing data collection and analysis processes, as well as incorporating more data to make the model more general and applicable.

Índice de la memoria

Índice de tablas

Capítulo 1. INTRODUCCIÓN

En este capítulo se realiza una introducción de este proyecto, describiendo su motivación a fin de despertar el interés del lector por el mismo.

1.1 MOTIVACIÓN DEL PROYECTO

El presente proyecto ha sido realizado en un contexto como el actual, en el que se da un desarrollo exponencial de las tecnologías. En concreto, en los últimos años se han desarrollado especialmente las tecnologías dirigidas a conseguir que, la misma tecnología, sea capaz de desarrollar conocimiento en sí misma. Aunque el desarrollo de este tipo de técnicas que han dado en llamarse, conjuntamente, aprendizaje automático (y, normalmente, más conocidas por su nombre en inglés, *Machine Learning*) daría para desarrollar de por sí un proyecto de mayor envergadura que el presente, me voy a centrar en explicar tres partes de este compendio que, por su importancia en el día a día actual y por ser las utilizadas en este proyecto, son las más relevantes a estas líneas.

Éstas no son otras que el aprendizaje profundo, el procesamiento del lenguaje natural y el análisis de sentimientos (de nuevo, más conocidas por sus nombres en inglés, *Deep Learning*, *Natural Language Processing* (NLP) y *Sentiment Analysis*, respectivamente).

Asimismo, en el desarrollo de este capítulo –y como parte de la Descripción del Proyecto—, definiré el problema al que se enfrenta el proyecto que nos atañe y daré una explicación de las razones que han llevado a elegir este campo de investigación.

1.1.1 APRENDIZAJE PROFUNDO (*DEEP LEARNING*)

El Deep Learning es un subconjunto de Machine Learning que se basa en redes neuronales con varias capas para tratar de emular el funcionamiento del cerebro humano – no llegando, de momento, a igualar su capacidad—(*¿Qué es Deep Learning? | IBM*, 2023).

De hecho, cada neurona que compone una red como las referidas arriba emula en sí misma el comportamiento de una neurona humana. Es decir, recibe una serie de estímulos, pondera la importancia de cada uno de ellos y toma una decisión en función del resultado de esta ponderación. De igual manera, las neuronas artificiales, en su versión más simple, reciben una serie de datos, les aplican unos pesos y sesgos y deciden en función de la combinación lineal de los datos con los pesos (*¿Qué es Deep Learning? | IBM*, 2023). Sin embargo, mientras que una neurona humana ya está "entrenada", una artificial es, inicialmente, aleatoria en sus decisiones. Por ello, se define una función de optimización por la cual, cada vez que la neurona recibe datos y toma una decisión, comprueba su resultado con el dato real y, dado un error o residuo entre su cálculo y la realidad, modifica sus pesos para adaptarse más al dato real.

Es importante darse cuenta de que, aunque una sola neurona sólo puede tomar decisiones para dividir un plano de manera lineal, la interconexión de muchas de ellas es capaz de tomar decisiones más complejas, pudiendo resolver problemas relevantes. Por ello, el Deep Learning es un campo de la Inteligencia Artificial que ayuda a mejorar la automatización de tareas analíticas para hacer innecesaria la intervención humana (*¿Qué es Deep Learning? | IBM*, 2023).

Es, de hecho, una tecnología que está cada vez más presente en nuestro día a día. El ejemplo más palpable son los modelos de Inteligencia Artificial Generativa, como ChatGPT o Gemini, pero reside detrás de muchos otros productos como los asistentes digitales, el reconocimiento de voz, la detección de fraudes en tarjetas de crédito o los coches autónomos (*¿Qué es Deep Learning? | IBM*, 2023).

1.1.2 PROCESAMIENTO DEL LENGUAJE NATURAL (*NATURAL LANGUAGE PROCESSING*; NLP)

Según AWS, "el procesamiento del lenguaje natural (NLP) es una tecnología de machine learning que brinda a las computadoras la capacidad de interpretar, manipular y comprender el lenguaje humano" (¿Qué Es El Procesamiento De Lenguaje Natural? - Explicación Del Procesamiento De Lenguaje Natural - AWS, s.f.). Leyendo esta definición, el ávido lector se habrá percatado de que el NLP entra también dentro de la definición anterior, por lo que se podría encuadrar dentro del Deep Learning (esto es, un modelo destinado a que una máquina pueda realizar tareas que normalmente se circunscriben al funcionamiento del cerebro humano).

La realidad es que, mientras que en muchos casos sí que es normal utilizar modelos que se podrían encuadrar dentro de la definición de aprendizaje profundo para realizar procesamiento del lenguaje natural, también es posible realizarlo mediante modelos no encuadrados en este grupo de técnicas si el lenguaje es tratado correctamente. Sin embargo, el hecho es que, en lo que respecta a este proyecto, las aplicaciones de este concepto se harán con modelos de Deep Learning.

Estas técnicas, al igual que las anteriores, se utilizan en gran cantidad de momentos de nuestro día a día como reconocimiento de voz, traducción automática o análisis de sentimientos (*¿Qué Es El Procesamiento De Lenguaje Natural? - Explicación Del Procesamiento De Lenguaje Natural - AWS*, s.f.), que es siguiente punto que se trata en esta Introducción.

1.1.3 ANÁLISIS DE SENTIMIENTOS (*SENTIMENT ANALYSIS*)

El análisis de sentimientos es un proceso por el cual se analiza un texto, normalmente proveniente de fuentes digitales a fin de determinar el tono emocional del mismo (¿Qué Es El Análisis De Opiniones? - Explicación Del Análisis De Opiniones - AWS, s.f.). Es, por

lo tanto, un subgrupo dentro del procesamiento del lenguaje natural que está destinado a descubrir la opinión de la persona a partir del texto que ha escrito.

Así, cualquier texto escrito por una persona puede ser sujeto de un análisis de sentimiento a fin de decidir cuál era la opinión de la persona que escribió el texto, lo que da más información sobre el contexto del texto en sí.

A fin de realizar este análisis de sentimiento se pueden realizar desde técnicas sencillas y manuales hasta técnicas complejas, que normalmente incluyen el procesamiento por parte de una máquina de una ingente cantidad de datos, en muchos de los casos haciendo uso de técnicas encuadrables en el aprendizaje profundo.

Si bien es cierto que el análisis de sentimientos se trata en este trabajo de manera tangencial, pues es más bien NLP lo que se hace, es cierto que uno de los fines es sacar el contexto de mercado, hecho que tiene mucho que ver con este tipo de técnicas.

1.2 DESCRIPCIÓN DEL PROYECTO

El proyecto que nos incumbe trata, en línea con lo explicado, de aplicar las técnicas descritas para tratar de solucionar un problema que, potencialmente, tiene motivaciones tanto académicas como económicas, como es la predicción del precio de productos financieros que cotizan en mercados abiertos.

Es comúnmente aceptado que los precios de los mercados financieros son, a menudo, volátiles, difíciles de predecir o, a primera vista, aleatorios. Sin embargo, estudiando su comportamiento más en profundidad se puede comprobar que el precio asignado a cada activo está determinado, principalmente, por la rentabilidad esperada del mismo, que mueve las fuerzas de la oferta y la demanda hasta determinar el precio que el mercado juzga correcto. En cualquier caso, estimar esta rentabilidad esperada ha sido siempre de extrema dificultad, pues en muchos casos ésta puede verse afectada por eventos futuros e

inciertos que hacen que, según la probabilidad que se le aplique a los mismos, dicha rentabilidad sea diferente para cada usuario, lo que redunda en un precio igualmente dispar.

Desglosando, llegamos a la conclusión de que se puede estimar el precio de un activo financiero como la agregación de cuatro componentes (*Banco BBVA - Productos financieros para personas y empresas | BBVA*, s.f.). La primera de ellas y más evidente son los factores fundamentales, que incluyen métricas de la compañía como beneficios, dividendos, diferentes ratios, etc. Todos estos datos están más o menos disponibles en el mercado y, aunque pueden ser interpretados de muchas maneras diferentes, ofrecen una base sólida para empezar a estimar un precio objetivo.

La segunda componente son los factores fundamentales. Su base es el análisis de los precios históricos del activo con el fin de tratar de predecir su comportamiento futuro. Para ello, además del precio del propio activo, se pueden utilizar variables como datos macroeconómicos, información de empresas en el mismo sector, tendencias, y demás. Este factor ofrece también una componente cuantitativa y, lo que es más importante, teóricamente cuantificable que influye en el precio del activo. Sin embargo, a más perfección de un mercado, menos efectivo se vuelve el análisis fundamental. De hecho, en mercados perfectos el precio se comporta como ruido blanco, lo que hace imposible la predicción por métodos estadísticos.

Estas dos componentes aunque, como se explica, pueden tener diferentes interpretaciones y ser más o menos difíciles de calcular, ofrecen una base cuantitativa para la estimación de un precio objetivo. Ello hace que se haya dedicado mucho tiempo y recursos a la obtención de procesos para realizar estimaciones sobre ellas y que, en la actualidad, existan modelos financieros, utilizados por aquellos que se dedican a operar en estos mercados, que explotan todo lo posible su potencial.

Pero de momento se han discutido solamente las dos primeras variables de las cuatro que se han propuesto más arriba. Las otras dos variables son puramente cualitativas, incluyendo las expectativas y el sentimiento del mercado. Podría pensar el lector que es en este punto cuando el usuario que intenta predecir el precio futuro de una acción se topa con una dificultad infranqueable y, tradicionalmente, así ha sido. Aunque se dispusiese de buenos modelos para estimar las dos variables previamente explicadas, no se podía restar el factor humano que decidiese sobre las dos introducidas en este párrafo, ya que no era sino con un amplio y profundo conocimiento del mercado y, en concreto, un conocimiento humano que se podían poner números a estas variables. Sin embargo, en la era del advenimiento de la inteligencia artificial, se presentan ante la humanidad máquinas que son capaces de realizar tareas que hace tan sólo unos años o incluso unos meses eran impensables y que, en algunos casos, simulan incluso el conocimiento humano. Este nuevo factor abre todo un nuevo horizonte de posibilidades que hace pensar en las posibles aplicaciones de estas nuevas tecnologías.

Estas tecnologías son precisamente, las que se han explicado en el apartado anterior. Los modelos creados con Inteligencia Artificial y, más concretamente, los que procesan el lenguaje a fin de sacar el contexto son de gran utilidad.

De hecho, la aplicación de esta técnica al problema presentado parece clara. Si una máquina fuese capaz de estimar la componente de expectativas y sentimiento de mercado de manera cuantitativa mediante la utilización de información que en el pasado no podía utilizar, quitaría –o más bien replicaría– la componente humana que estima lo mismo gracias a un conocimiento previo del mercado, lo que facilitaría asimismo el acceso a estos mercados.

La conclusión es que no es sino a esto que está destinado este proyecto, a tratar de estimar precios de una manera puramente automática, sin aplicar componentes cualitativas o, mejor dicho, dejando que sea la máquina quien las aplique.

Por si no fuese esto suficiente motivación para el lector, imagínese no tener necesidad de tener asesores financieros, que teniendo simplemente un modelo corriendo en un servidor con una serie de órdenes sobre el perfil del inversor se pudiese conseguir la rentabilidad deseada. Obviamente, este extremo es posible en teoría. En la práctica, este trabajo simplemente aporta un pequeño primer paso, aunque necesario, para llegar a este punto, un ejemplo que sólo utiliza una pequeña cantidad de activos y predice sobre ellos teniendo en cuenta estos cuatro componentes que se discutían al principio de este apartado.

1.3 DESCRIPCIÓN DEL TRABAJO

Aparte de llevar a cabo un estudio de las tecnologías mencionadas y de las soluciones que éstas presentan, resultaba necesaria para la compleción de este trabajo realizar un caso práctico a fin de comprobar el funcionamiento de todo lo descrito y estudiado. Como se ha mencionado, estas tecnologías se presentan como útiles para una gran cantidad de campos y son, por lo tanto, aplicables a una ingente cantidad de casos.

Sin embargo, dadas las limitaciones, tanto temporales como físicas o técnicas, que ofrece la realización de este trabajo, se ha planteado en la aplicación de este trabajo un sistema menor en el número de activos que incluye y en el tamaño de los modelos.

En cualquier caso, se ha hecho un diseño que se cree que podría ser aplicado a una mayor escala, simplemente teniendo mayores recursos de computación y tiempo para su desarrollo, así como pudiendo acceder a mejores o mayor cantidad de datos. Este diseño será explicado en las siguientes líneas, primero en alto nivel y luego explicando cada parte del mismo.

1.3.1 DESCRIPCIÓN GENERAL DEL DISEÑO

El diseño propuesto está basado en dos modelos, que definiremos como modelo base y modelo de noticias. En el primero, se entrenará un modelo que utiliza le análisis técnico

antes explicado para tratar de predecir la evolución del precio en el futuro. En el segundo se entrenará un modelo que aprenda de noticias y trate de mejorar la previsión hecha por el primero.

De todas formas, antes de poder explicar cada modelo, debemos entender el problema en profundidad. El título de este Trabajo de Fin de Grado es "Predicción de Precios Intradía en Mercados Financieros utilizando Noticias". Analicemos el título. La predicción de los precios ha sido presentada ya en varias ocasiones a lo largo de estas líneas, así como se ha explicado también dónde entran las noticias en la ecuación. Sin embargo, no ha sido aún explicado qué es y por qué se predice el precio intradía.

Es posible que el lector sepa lo que es el precio intradía, pues su propio nombre indica ya su significado. El hecho es que lo normal, cuando vemos precios históricos de acciones, veamos el precio de cierre de cada día –es decir, el último precio del día–, pero resulta obvio para cualquiera que haya seguido en algún momento la cotización de algún activo que, en el transcurso de un día, el precio de éstos fluctúa. Los precios intradía no son otra cosa que estos precios que toma la cotización de un activo entre un cierre y otro. ¿Por qué son relevantes a este proyecto? La razón es simple: si se utilizan noticias para predecir el precio de la acción, al tomar un rango tan largo como un día se pueden perder los efectos de su publicación. Si se piensa, el tiempo que se tarda en leer una noticia debe estar en el rango de los minutos y, una vez el mercado recibe y procesa esa información, toma una decisión de manera casi automática.

Es por eso que en el desarrollo de los modelos de este trabajo funcionaremos en este rango, el de los minutos. En concreto, el rango final elegido viene a asumir que son las noticias publicadas en la última hora las que definirán el posible movimiento de los precios a lo largo de los diez minutos siguientes. Es decir, se utilizarán datos de los 60 minutos anteriores para predecir los 10 siguientes. Este hecho hace que el presente trabajo no sea, en realidad, de aplicación al público en general, que no puede actuar sobre el mercado en

un plazo tan pequeño. Es, de hecho, más aplicable a los sujetos dedicados a proveer de liquidez a los mercados financieros –esto es, los llamados "*market makers*"—. Estos sujetos actúan como intermediarios entre el usuario que quiere vender y el que querrá comprar a fin de que no ocurra que, cuando alguien quiera vender, no exista comprador, lo que tensionaría el mercado. Este trabajo lo hacen a cambio de un diferencial entre el precio al que compran y al que luego venden los activos. Dado que este diferencial es muy ajustado (o, más bien, a fin de ajustarlo más correctamente), el sujeto debe tratar de saber cuál será el precio al que pueda vender en el futuro, en un plazo realmente corto. Es ésta la razón de que este trabajo sea de su interés, ya que son las personas que más necesitan saber cómo va a ser la fluctuación del precio en el corto plazo.

Con esto, ya entendemos ligeramente mejor cómo están estructurados y lo que se busca conseguir con los modelos que componen este trabajo. Así, podemos pasar a explicarlos más concretamente.

1.3.2 MODELO BASE

Este modelo, como se ha explicado anteriormente, utilizará únicamente los datos pasados sobre el precio de la acción en los 60 minutos anteriores al actual para predecir los 10 siguientes.

Es cierto que se pueden tomar muchas métricas de los precios de cotización por minuto y, de hecho, serán varias de ellas las que ayuden a predecir el precio futuro. En concreto, se utilizarán los precios de cierre, de apertura, el máximo y el mínimo de cotización alcanzados, el volumen de transacciones y el día de la semana, para proporcionar algo más de contexto al modelo.

En cuanto al diseño del modelo, se utilizarán redes neuronales con múltiples capas para tratar de sacar relaciones complejas entre todos estos datos. El diseño concreto de los

modelos se explicará en un apartado posterior, una vez que se hayan presentado y explicado las tecnologías que se utilizan para crear el mismo.

1.3.3 MODELO DE NOTICIAS

Al igual que el anterior, este modelo utiliza datos de los 60 minutos anteriores a que se procesa para predecir los 10 siguientes. Sin embargo, los datos que utiliza este modelo son radicalmente diferentes a los del modelo anterior.

Primero, los datos de entrada son las noticias publicadas en el rango descrito. Resulta palpable que una máquina no es capaz de procesar texto en su formato 'bruto' por lo que, obviamente, estas noticias no se le pasarán al modelo como las leería un humano, sino como una secuencia de números que, esta vez sí, son interpretables por la máquina.

Segundo, los datos de salida no son en sí mismos los precios de cotización de los diez minutos siguientes, sino el error cometido por el modelo anterior en esos mismos minutos –es decir, el resultado de sustraer al dato real el dato predicho por el modelo anterior–. Aunque este procedimiento pueda parecer extraño, se entiende si se piensa con algo más de profundidad.

Un modelo de aprendizaje automático bien entrenado utiliza los datos que tiene, saca las relaciones que le es posible y, con todo ello, predice de la mejor manera posible el resultado. Pero en ningún caso será capaz de utilizar información que no ha recibido. De esta forma, el modelo base podrá predecir la información que se pueda sacar de la componente técnica, pero no será capaz de predecir otras componentes. Así, una vez aprendida la componente técnica, el residuo o error cometido será la parte del precio influenciado por otras componentes, que son las que trata de aprender el modelo de noticias. Es decir, al pasarle los errores cometidos por el modelo anterior no le decimos otra cosa que "esto es lo que queda por predecir… ¿puedes predecirlo?".

Resulta así evidente que este modelo es, en esencia, bastante más complicado que el anterior, pues trata de predecir lo que el otro no ha sido capaz de hacer.

Al igual que el otro, este modelo se compone de múltiples capas de neuronas interconectadas de diferentes tipos, a lo que éste añade capas para que el modelo sea capaz de entender el texto. De igual manera, se explicará en profundidad cuando se hayan explicado sus componentes.

1.3.4 PREDICCIÓN CONJUNTA

Una vez entrenados los dos modelos, se realizará una adición de los mismos para tratar de hacer una predicción que tenga en cuenta todos los componentes explicados en la Introducción.

Dada la especial manera de construir los modelos, para realizar esta predicción conjunta es suficiente con hacer la adición de la predicción de ambos, pues el primero hace una predicción general y el segundo hace la predicción del error que comete el primero. Si ese error es calculado como la diferencia entre el dato real y el predicho, el dato predicho estimado por el conjunto podrá calcularse como la suma de la predicción inicial y la predicción del error.

Habiendo explicado todo esto, es evidente que todo lo presentado es, de momento, una abstracción de lo que se va a hacer. Pasamos ahora a explicar lo realizado, para lo cual primero hay que comprender dos puntos, que no son otros que qué se ha utilizado para realizarlo y qué se ha hecho de manera similar en el pasado. El primero de ellos sirve para dar base al trabajo, para explicar sus componentes y que el lector pueda comenzar a comprender por qué se han utilizado. El segundo, sirve para demostrar que se conoce el estado de la técnica y que estos trabajos previos han influenciad y basado el presente proyecto.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

A lo largo de este capítulo se realizará un análisis exhaustivo de las tecnologías que han sido utilizadas para el desarrollo de este proyecto. Resulta claro que existe una división inicial sencilla entre estas tecnologías, las utilizadas para la consecución de los datos que utiliza el trabajo y las que se han utilizado para su tratamiento y su utilización por medio de modelos. Aunque a primera vista pudiese parecer que las segundas resultan más relevantes que las primeras, este punto no es tan claro en realidad pues no es sino mediante la consecución de datos relevantes y de calidad que se pueden desarrollar modelo eficaces y certeros.

Pasemos, pues, primero a la explicación de las tecnologías y demás medios utilizados en el proceso de consecución de los datos, la base de este proyecto.

2.1 TECNOLOGÍAS PARA CONSECUCIÓN DE DATOS

Como se puede derivar de las líneas anteriores, las tecnologías utilizadas para conseguir los datos son, eminentemente, dos, pues dos son los tipos de datos que se tratan de conseguir, datos históricos de cotización y noticias. Sin embargo, como se explicará más adelante en el capítulo dedicado al desarrollo del sistema, la dificultad presentada para conseguir los datos, tanto de un tipo como de otro, han llevado a que sean, en realidad, más las tecnologías que se han tratado de utilizar para la consecución de los datos, siendo las dos aquí explicadas las que han sido, finalmente, utilizadas.

2.1.1 PRECIOS HISTÓRICOS: YAHOO FINANCE

Esta web se ha utilizado mediante el paquete ‘yfinance’ de Python, que es una librería extremadamente útil y poderosa desarrollada para permitir a los usuarios obtener de manera sencilla datos financieros desde la interfaz de programación de aplicaciones (API) de Yahoo Finance.

Este paquete ofrece una interfaz simple pero robusta para descargar datos financieros. Por ejemplo, permite obtener no solo los precios de cierre ajustados y no ajustados de las acciones, sino también otros datos relevantes como el volumen de acciones negociadas, el precio máximo y mínimo del período, entre otros. Estos datos, como ya se ha explicado anteriormente, pueden ser de gran utilidad para el desarrollo de modelos predictivos sobre el comportamiento de los precios, entre muchas otras aplicaciones.

Para utilizar 'yfinance' con el fin de obtener datos históricos de cotización de acciones, primero se debe instalar el paquete, lo cual se puede realizar fácilmente mediante *pip*, el gestor de paquetes de Python. Una vez instalado, el usuario puede importar y utilizar su función 'Ticker' para obtener datos de uno o múltiples símbolos de acciones en un rango de fechas específico. Este proceso es sumamente sencillo: basta con especificar el símbolo de la acción (por ejemplo, 'AAPL' para Apple Inc.), las fechas de inicio y fin del período deseado, y la periodicidad de los datos (por ejemplo, diaria, semanal, mensual o, en este caso, por minuto) (*Yfinance*, 2024).

Un aspecto destacable de 'yfinance' es su flexibilidad. El usuario puede ajustar numerosos parámetros para adecuar la consulta a sus necesidades específicas, como modificar el intervalo de tiempo de los datos, seleccionar qué tipo de precios obtener (ajustados por dividendos y splits o no ajustados), y decidir el formato en el que desean recibir los datos (por ejemplo, como un DataFrame de pandas, lo cual facilita enormemente el análisis posterior con herramientas de Python).

En resumen, 'yfinance' es una herramienta valiosa para cualquier persona interesada en el análisis financiero, proporcionando una manera directa y eficiente de acceder a datos históricos de mercado. Su facilidad de uso, junto con la rica funcionalidad que ofrece, lo convierten en una opción destacada para profesionales y aficionados del ámbito financiero que buscan realizar análisis de datos de mercado de forma efectiva y sin incurrir en costes elevados, ya que se trata de una tecnología gratuita.

Sin embargo, como toda funcionalidad gratuita, tiene sus limitaciones. En este caso, dado que la frecuencia con la que se necesitaban los datos era por minuto, el sistema ofrecía la limitación de dar datos por minuto con una antigüedad de, como máximo, un mes, teniendo además que sacarlos en períodos de, a lo sumo, 7 días por petición, lo que hacía que se tuviesen que hacer unas 5 peticiones por activo que se solicitase.

Ésta ha sido una de las razones por las que se ha demorado la consecución de los datos para realizar el trabajo, pues los dos modelos no se podían entrenar en base a datos de un solo mes. Por ello, se tuvieron que hacer peticiones sucesivas durante varios meses para después juntar los datos y utilizar estos datos conjuntos para realizar los modelos. En concreto, se consiguieron datos a partir de marzo de 2024, llegando hasta final de mayo del mismo año.

En cualquier caso, el tratarse de una herramienta gratuita y el no tener una opción diferente que ofreciese tanta funcionalidad hicieron que finalmente me decantara por esta tecnología para recopilar los datos que suponen la base para los modelos definidos.

2.1.2 NOTICIAS: NEWS API

Habiendo conseguido los datos históricos de precios, el otro paso para poder desarrollar el trabajo consistía en conseguir noticias sobre esas mismas compañías, para lo que se utilizó News API. News API es una interfaz de programación de aplicaciones (API) moderna y sofisticada que se especializa en la recopilación y distribución de noticias de diversas fuentes alrededor del mundo. Esta API está diseñada para facilitar el acceso a artículos y titulares de noticias en tiempo real, ofreciendo a los desarrolladores y analistas de datos una herramienta potente para extraer información actualizada y relevante que puede ser utilizada en una amplia gama de aplicaciones, desde la creación de agregadores de noticias hasta el análisis de tendencias y la monitorización de eventos específicos.

El uso de News API comienza con la obtención de una clave de API, que es necesaria para realizar peticiones y acceder a los datos que proporciona. Una vez que un usuario se registra y obtiene su clave, puede comenzar a realizar consultas especificando diferentes parámetros según sus necesidades. Por ejemplo, las consultas pueden filtrarse por palabras clave, fuentes de noticias específicas, idiomas, países, y fechas de publicación. Esto permite una gran flexibilidad y hace que News API sea extremadamente útil para obtener información altamente relevante y personalizada (*Documentation - News API*, s.f.).

La estructura de respuesta de News API está bien organizada y es fácil de manejar. Normalmente, incluye información detallada sobre cada artículo, como el título, la descripción, la fuente, el autor, la URL del artículo, y la fecha de publicación. Esta información se devuelve en formato JSON, un estándar de fácil manejo y ampliamente utilizado en el desarrollo web y móvil, lo que facilita la integración de los datos de News API en aplicaciones existentes o nuevas.

El primer trabajo relevante estaba titulado como "Predicción de la tendencia de las acciones utilizando análisis de sentimientos de noticias" (originalmente en inglés, "*Stock Trend Prediction using News Sentiment Analysis*", *Kalyani et al.*, 2016). Éste aborda la complejidad de la predicción de movimientos bursátiles utilizando métodos avanzados de aprendizaje automático y técnicas de minería de texto. La hipótesis subyacente del estudio es que las noticias financieras impactan significativamente en las tendencias del mercado de valores, y mediante la clasificación del sentimiento de las noticias se pueden prever estas tendencias.

La metodología adoptada implica el desarrollo de modelos de clasificación que determinan una clasificación de los artículos financieros como positivos o negativos. Sobre los métodos utilizados, se construyeron tres modelos utilizando los algoritmos de Naive Bayes, Random Forest y Máquinas de Vectores de Soporte (SVM), pudiendo derivar que Random Forest y SVM mostraron un mejor desempeño.

El estudio utilizó datos de tres años de Apple Inc., combinando precios de acciones diarios y artículos de noticias relevantes. A fin de realizar el análisis, se comenzó con la recolección y el preprocesamiento de textos para ajustarlos a un formato utilizable para los modelos descritos. Este preprocesamiento incluyó la tokenización, eliminación de palabras irrelevantes (generalmente conocidas como "*stopwords*") y ruido, y la aplicación de técnicas de *stemming*, que tratan de reducir la palabra hasta su base. Sobre esta base, se aplicaron métodos para detectar el sentimiento mediante un enfoque basado en diccionario, utilizando un conjunto de palabras clasificadas como positivas o negativas.

Una vez procesados los datos, se implementaron los modelos de clasificación. Los resultados se evaluaron mediante diversas métricas como la precisión, el *recall* y el área bajo la curva ROC (conocida como "*area under the curve*" o AUC), con una precisión del modelo general superior al 80%. Se realizó también una predicción sobre un conjunto de datos no visto aún por el modelo. Los resultados alentadores indican que el modelo puede aumentar significativamente la precisión en la predicción de tendencias de las acciones en comparación con métodos que asignan etiquetas de manera aleatoria.

Finalmente, el estudio concluye que existe una relación significativa entre el sentimiento de las noticias y las tendencias del mercado de valores. Los modelos desarrollados pueden ayudar a los inversores y analistas a tomar decisiones más informadas basadas en el análisis de sentimientos de las noticias. Para trabajos futuros, se sugiere expandir la investigación a otras empresas y examinar el uso de datos de redes sociales como Twitter para análisis similares, así como su aplicación en el trading algorítmico.

Este trabajo guarda una gran relación con el trabajo que tenemos entre manos, aunque guarda también significativas diferencias, sobre todo en los datos utilizados y las técnicas de preprocesado de los datos aplicadas.

```
[{'source': {'id': None, 'name': '9to5Mac'},
  'author': 'Seth Kurkowski',
  'title': '9to5Mac Daily: May 3, 2024 –\xa0AAPL Q2 earnings and AI promises',
  'description': 'Listen to a recap of the top stories of the day from\xa09to5Mac. 9to5Mac Daily is available\xa0on iTunes and Apple's Podcasts app,\xa0Stitcher,\xa0TuneIn,\xa0Google Pla
  'url': 'https://9to5mac.com/2024/05/03/daily-may-3-2024/',
  'urlToImage': 'https://i0.wp.com/9to5mac.com/wp-content/uploads/sites/6/2021/12/9to5Mac-Daily-art-lead.jpg?resize=1200%2C628&quality=82&strip=all&ssl=1',
  'publishedAt': '2024-05-03T17:33:42Z',
  'content': 'Listen to a recap of the top stories of the day from\xa09to5Mac. 9to5Mac Daily is available\xa0on iTunes and Apples Podcasts app,\xa0Stitcher,\xa0TuneIn,\xa0Google Play, or
  'clean_cont': 'Listen to a recap of the top stories of the day from\xa09to5Mac. 9to5Mac Daily is available\xa0on iTunes and Apples Podcasts app,\xa0Stitcher,\xa0TuneIn,\xa0Google Play,
  'errors': 'Error fetching content: Start of article not found'},
 {'source': {'id': None, 'name': 'AppleInsider'},
  'author': 'news@appleinsider.com (Mike Wuerthele)',
  'title': 'One hedge fund completely bailed out of AAPL, but another more than picked up the slack',
  'description': "Prior to Apple stock's value recovery after a better than expected quarter, one hedge fund got rid of all of its holdings in the iPhone maker, and another went in big.A
  'url': 'https://appleinsider.com/articles/24/05/16/one-hedge-fund-completely-bailed-out-of-aapl-but-another-more-than-picked-up-the-slack',
  'urlToImage': 'https://photos5.appleinsider.com/gallery/58327-118868-56908-115757-54123-109024-48167-94082-46756-91138-cook-financial-2022-xl-xl-xl-xl-xl.jpg',
  'publishedAt': '2024-05-16T10:27:55Z',
  'content': "Apple CEO Tim Cook\r\nPrior to Apple stock's value recovery after a better than expected quarter, one hedge fund got rid of all of its holdings in the iPhone maker, and ano
  'clean_cont': "Apple CEO Tim Cook\r\nPrior to Apple stock's value recovery after a better than expected quarter, one hedge fund got rid of all of its holdings in the iPhone maker, and
  'errors': 'Error fetching content: Start of article not found'},
 {'source': {'id': None, 'name': '9to5Mac'},
  'author': 'Chance Miller',
  'title': 'Warren Buffett's Berkshire Hathaway sells 13% of its Apple shares',
  'description': 'Warren Buffett's Berkshire Hathaway offloaded around 13% of its Apple holdings in Q1 2024, the conglomerate revealed this weekend. Despite selling around 115 million sh
  'url': 'https://9to5mac.com/2024/05/04/warren-buffett-berkshire-hathaway-aapl/',
...
  'urlToImage': 'https://applech2.com/wp-content/uploads/2024/05/Anker-Japan-Power-Bank-30W-Fusion-Built-In-USB-C-Cable-new-colors.jpg',
  'publishedAt': '2024-05-21T04:12:35Z',
  'content': 'Anker JapanUSBUSB-CAnker Power Bank (30W, Fusion, Built-In USB-C )\r\nAnker Japan20240521USBUSB-CAnker Power Bank (30W, Fusion, Built-In USB-C ) (A1636)\r\nAnker Power Bank
  'clean_cont': 'Anker JapanUSBUSB-CAnker Power Bank (30W, Fusion, Built-In USB-C )\r\nAnker Japan20240521USBUSB-CAnker Power Bank (30W, Fusion, Built-In USB-C ) (A1636)\r\nAnker Power B
  'errors': 'Error fetching content: Start of article not found'}]
```


Figura 1: Ejemplo de resultado de NewsAPI

En resumen, News API es una herramienta esencial para cualquier desarrollador o analista que necesite integrar datos de noticias en sus proyectos. Su facilidad de uso, junto con la amplia cobertura y la respuesta rápida, así como su coste gratuito para labores de investigación, la convierten en una solución ideal para aplicaciones que requieren acceso a información actualizada a nivel global.

En este caso en concreto, se le pasaban los tickers de las empresas que se estaban utilizando y las fechas entre las que se buscaba que estuviesen las noticias, devolviendo a los sumo 100 noticias por empresa.

Al igual que la tecnología anterior, ésta pone un límite temporal de un mes en las noticias que devuelve, por lo que será durante este tiempo durante el que se entrenará el modelo correspondiente. En este caso, los datos conseguidos fueron noticias durante el mes de mayo de 2024 por lo que, para el entrenamiento del modelo de noticias, se utilizarán tanto los datos históricos de mayo como las noticias del mismo período.

Además, la API pone un límite al número de peticiones que se pueden hacer, lo que pone también un límite a la cantidad de activos que se pueden procesar (en realidad, este límite

deja de ser relevante al darse cuenta de que los recursos computacionales establecen un límite aún menor).

2.1.3 OTRAS TECNOLOGÍAS PARA LA CONSECUCIÓN DE DATOS

Además de las dos explicadas anteriormente, se han utilizado a lo largo del desarrollo de este trabajo otras tecnologías que han sido, por una razón u otra, finalmente descartadas. Entre las más relevantes figuran Bloomberg y sus APIs y la API de OpenAI.

Como es obvio, la primera fue estudiada como posible fuente de datos históricos para las acciones. Esto era posible dado que, a pesar de su alto coste, que haría imposible que un usuario que sólo fuese a utilizarlo para un proyecto como el presente accediese a esta herramienta, la universidad pone a disposición del alumno diversos terminales para su utilización. Sin embargo, y aunque inicialmente parecía que podía ser mejor opción que la API de Yahoo Finance por ofrecer datos por minuto con una antigüedad de, a lo sumo, 140 días, finalmente fue imposible su utilización dado que establece un límite de descargas diario de 500.000 datos. Si se hacen las cuentas, si se quisiesen descargar datos de 6 variables (las antes mencionadas) durante 140 días por minuto (sólo durante las horas de cada día que está abierto el mercado), se descargarían más de 400.000 datos por empresa, lo que haría que, sólo pudiendo descargar datos de una empresa al día, se hiciese tremendamente tedioso el proceso de consecución de los datos, lo que llevó a el descarte de esta técnica.

Por parte de la segunda, fue estudiada por el formato en que devuelve News API las noticias, ofreciendo solamente un *snippet* de su contenido, una descripción de la misma y un enlace a la fuente. Esto llevó al desarrollo de un código de *webscrapping* para tratar de descargar el contenido completo. Sin embargo, cualquiera que haya hecho *webscrapping* sabrá que, cuando se quieren descargar datos automáticamente de varias webs diferentes, la generalidad a la que hay que abstraerse hace que sea complicado sacar datos con la calidad que gustaría. Por ello, exploré la posibilidad de utilizar la API de OpenAI, de pago, para

sacar de ese HTML el contenido limpio de la noticia. Aunque el código dio resultados satisfactorios, el coste que acarrearía implementarlo en la totalidad de las noticias hacía imposible su utilización pues, para la limpieza de una sola noticia, cobraba unos 80 céntimos. Por ello, finalmente se utilizó un código de *webscrapping* que trataba de, en la medida de lo posible, limpiar lo máximo el contenido de la noticia y, en los casos en que esto no fuere posible, la descripción de la misma.

2.2 TECNOLOGÍAS PARA EL DESARROLLO DE MODELOS

En esta parte se procederá a explicar los paquetes que se han utilizado para poder desarrollar los modelos explicados en la introducción. Como se puede derivar de las líneas anteriores, el lenguaje de programación utilizado ha sido Python. Para este lenguaje existen multitud de paquetes para desarrollar el potencial del aprendizaje automático. Los elegidos han sido tal por ser los que el autor conoce en mayor profundidad, así como porque son los más conocidos que desarrollan este tipo de funcionalidad. Estos son Tensorflow y su paquete de alto nivel, Keras.

2.2.1 TENSORFLOW

TensorFlow es una biblioteca de software de código abierto para computación numérica que facilita la creación de sistemas capaces de aprender y deducir a partir de los datos. Originalmente desarrollado por investigadores e ingenieros del equipo Google Brain dentro de Google's Machine Intelligence research organization, fue liberado al público en 2015 (*González*, 2022). Desde entonces, ha ganado popularidad en la comunidad de aprendizaje automático debido a su flexibilidad, escalabilidad y amplia comunidad de apoyo.

La principal fortaleza de TensorFlow radica en su capacidad para realizar cálculos intensivos de manera eficiente con ayuda de gráficos de flujo de datos, donde los nodos representan operaciones matemáticas y las aristas reflejan los datos multidimensionales (tensores) que fluyen entre ellos (*TensorFlow Core*, s.f.). Esta estructura permite a TensorFlow no solo utilizar eficientemente CPUs y GPUs para acelerar el procesamiento,

sino también soportar TPU (Tensor Processing Unit) para cálculos aún más rápidos. Esto lo convierte en una herramienta ideal para tareas que requieren manipulación intensiva de datos y computación, como entrenamiento e inferencia de modelos de redes neuronales.

TensorFlow se usa ampliamente en diversas aplicaciones que van desde la detección de objetos y el procesamiento del lenguaje natural hasta sistemas de recomendación y análisis predictivo. Su capacidad para manejar grandes volúmenes de datos y su escalabilidad lo hacen adecuado tanto para proyectos individuales pequeños como para soluciones empresariales a gran escala. Además, TensorFlow ofrece una API de alto nivel llamada Keras, que será explicada en el siguiente punto, que simplifica la creación de prototipos de modelos de aprendizaje automático con una interfaz más accesible.

Además, Google ha asegurado que TensorFlow se mantenga actualizado con las últimas tendencias y avances en aprendizaje automático, proporcionando actualizaciones regulares y mejoras en la biblioteca.

2.2.2 KERAS

Keras es una biblioteca de aprendizaje profundo en Python, diseñada para permitir la construcción rápida y sencilla de modelos de aprendizaje automático, especialmente modelos de aprendizaje profundo. Funciona como una interfaz para la biblioteca de TensorFlow, proporcionando herramientas fáciles de usar para construir y entrenar redes neuronales. Keras hace el desarrollo de modelos más accesible con su API de alto nivel y es popular entre la comunidad científica y de investigación por su simplicidad y eficiencia.

En Keras, existen dos principales tipos de modelos: el modelo secuencial y el modelo funcional. El modelo secuencial es una pila lineal de capas donde cada capa tiene exactamente un tensor de entrada y un tensor de salida. Este modelo es muy adecuado para arquitecturas simples de redes neuronales donde no hay conexiones entre capas no adyacentes o donde no se requieren múltiples entradas o salidas. Por otro lado, el modelo

funcional es más flexible y permite la creación de modelos que pueden tener múltiples entradas y salidas, y donde se pueden utilizar conexiones entre capas de manera arbitraria. Esto lo hace adecuado para arquitecturas de red más complejas, como las redes con conexiones residuales, las redes con múltiples ramas, etc. (*TensorFlow Core*, s.f.).

Las capas o *layers* en Keras son los bloques de construcción fundamentales de las redes neuronales y cada tipo de capa realiza una función específica. Aquí están algunas de las más importantes, utilizadas a lo largo de los modelos del trabajo:

- Dense: Una capa densa es una capa neuronal clásica donde cada entrada está conectada a cada salida por una función lineal, seguida típicamente por una función de activación no lineal. Es ampliamente utilizada en todo tipo de redes neuronales.

- GRU (Gated Recurrent Unit) y LSTM (Long Short-Term Memory): Estas capas son variantes de las redes neuronales recurrentes (RNN) que ayudan a mantener la información en 'memoria' por largos periodos de tiempo. Son particularmente útiles para secuencias de entrada donde la temporalidad es importante, como en el procesamiento de lenguaje natural y series temporales.

- Convolucional: Las capas convolucionales aplican un conjunto de filtros aprendibles a sus entradas. Estas capas son fundamentales en el procesamiento de imágenes, donde ayudan a la red a aprender características visuales a varios niveles de abstracción. En concreto, en este trabajo se utilizan porque se aplica el mismo concepto que el de las imágenes al añadir múltiples 'canales' con información de diferentes empresas.

- Embedding: Esta capa convierte índices enteros (que representan palabras específicas, por ejemplo) en vectores densos de un tamaño fijo y es común en el procesamiento de texto donde los modelos necesitan una representación numérica de objetos discretos. Así, ahorra recursos computacionales y mantiene mejor las relaciones entre las palabras que las soluciones basadas en *encoding* anteriores.

- TimeDistributed: Esta capa aplica una capa específica a cada segmento temporal de una entrada, lo que es útil para modelar series de tiempo cuando se desea que la misma capa actúe sobre diferentes segmentos temporales de la entrada.

- Reshape: Cambia la forma del tensor de entrada a una forma específica y es útil cuando se necesita alterar la estructura de los datos, por ejemplo, preparando el tensor para una capa convolucional después de una recurrente.

- Flatten: Aplana el tensor de entrada en un vector y es a menudo usado entre capas convolucionales y densas para transformar la representación multidimensional de los datos en 1D.

- Dropout: Se utiliza para reducir el sobreajuste en los modelos de aprendizaje automático al ignorar unidades durante la fase de entrenamiento al azar, según un parámetro indicado, lo que ayuda a hacer el modelo más generalizable.

Todas estas capas y algunas más concretas serán utilizadas a lo largo de la creación de los diferentes modelos que se presentan en este trabajo.

2.3 RECURSOS COMPUTACIONALES

Si bien todo lo anterior desarrolla una explicación de los recursos de software utilizados para realizar el trabajo, es relevante el realizar un apunte sobre los recursos de hardware computacionales que se han utilizado, pues han establecido los mayores cuellos de botella para terminarlo.

Si bien se utilizaba inicialmente un portátil para realizar los modelos, este dejó de resultar suficiente cuando se llegó al segundo de ellos, pasando a utilizar Google Colab, un servicio en la nube que permite a los usuarios escribir y ejecutar código Python en un navegador sin requerir configuración adicional. Basado en la tecnología de Jupyter Notebook, proporciona acceso gratuito a recursos computacionales, incluyendo GPUs y TPUs, lo que lo hace ideal para proyectos de aprendizaje automático y análisis de datos. Colab facilita la colaboración en tiempo real y el acceso a archivos desde Google Drive, GitHub y otros sistemas de almacenamiento externo, haciendo que sea una herramienta accesible y poderosa para educadores, investigadores y desarrolladores que buscan ejecutar, compartir y colaborar en proyectos de ciencia de datos de forma eficiente y eficaz (*Google Colab*, s.f.). La razón principal para utilizar esta tecnología es precisamente esta, el acceso a las GPUs, pues la CPU no resultaba suficiente para el desarrollo de los modelos.

2.3.1 CPU vs. GPU

Una CPU (Unidad Central de Procesamiento) y una GPU (Unidad de Procesamiento Gráfico) son componentes cruciales en los sistemas informáticos, pero se especializan en diferentes tipos de procesamiento.

La CPU, considerada el “cerebro” del ordenador, es responsable de ejecutar las instrucciones de los programas a través de sus unidades de procesamiento que son capaces de manejar una amplia variedad de tareas. Tradicionalmente, la CPU se compone de pocos núcleos con alta capacidad de procesamiento por núcleo, lo que les permite realizar tareas complejas que requieren cálculos secuenciales. Además, la CPU maneja todas las

instrucciones básicas que operan un computador, como el procesamiento del sistema operativo, la ejecución de aplicaciones software, y la gestión de dispositivos y de memoria.

Por otro lado, la GPU está diseñada específicamente para el procesamiento de gráficos. Consta de cientos o incluso miles de núcleos pequeños que trabajan en paralelo, lo que la hace excepcionalmente buena en manejar múltiples operaciones simultáneamente. Originalmente, las GPUs fueron diseñadas para acelerar la creación de imágenes en el buffer de un marco de vídeo para la salida a una pantalla. Sin embargo, su capacidad para procesar múltiples tareas simultáneamente ha llevado a su uso en áreas más allá de los gráficos, como en simulaciones científicas y aprendizaje automático, donde se necesitan cálculos masivos y simultáneos.

Las diferencias entre CPU y GPU son notables principalmente en su estructura y eficiencia en diferentes tareas. Mientras que la CPU es mejor para realizar una amplia gama de tareas generales y cálculos complejos que dependen de una ejecución rápida y secuencial, la GPU sobresale en aplicaciones donde múltiples cálculos pueden ser realizados en paralelo, como el procesamiento de gráficos y operaciones de álgebra matricial extensa. Esto hace que la GPU sea ideal para gráficos 3D, procesamiento de vídeo, y aplicaciones de inteligencia artificial, donde se pueden realizar muchos cálculos independientes simultáneamente.

En resumen, la CPU es más versátil y esencial para el funcionamiento general de la computadora, mientras que la GPU es más especializada para tareas que requieren un procesamiento paralelo masivo, como es el entrenamiento de un modelo de Deep Learning.

Capítulo 3. ESTADO DE LA CUESTIÓN

Para la presentación de las conclusiones de esta búsqueda, tendremos en cuenta el hecho de que el trabajo presentado tiene, como se ha mencionado anteriormente, dos partes principales, que no son otras que la predicción de precios utilizando históricos y la adición de un modelo que haga lo propio utilizando noticias. Por ello, se explicarán primero la literatura sobre el uso de Machine Learning para la predicción de precios y después trabajos dedicados a la utilización de noticias para predecir el precio. Finalmente, para cada apartado se acabará mencionando cómo afectan estos descubrimientos al desarrollo de este proyecto y las conclusiones que se sacan de ellos.

3.1 LITERATURA SOBRE MACHINE LEARNING EN FINANZAS

El trabajo sobre el que se ha basado parte de la revisión del estado de la técnica es "Preciado Empírico de Activos via Machine Learning" (originalmente, "*Empirical Asset Pricing via Machine Learning*", *Gu et al.*, 2018), que lleva a cabo un análisis comparativo de las técnicas que existían hasta el momento para aplicar métodos que utilizan el aprendizaje profundo a finanzas, así como una aplicación de estas técnicas a un conjunto que contiene los datos de las empresas cotizadas de EE.UU. a lo largo de los anteriores 60 años.

Cabe mencionar, como contexto y paso previo a la presentación de los aprendizajes sacados del estudio mencionado, que éste se centra en buscar la solución a un problema muy conocido en finanzas, el de conseguir medir la prima de riesgo de las acciones ("*equity risk premium*" en inglés, que es la rentabilidad por encima de aquella que ofrece un activo libre de riesgo).

Ampliamente, lo que demuestra el estudio, que compara diferentes tipos de modelos de Machine Learning para predecir el problema mencionado en el párrafo anterior, es que la

utilización de estos modelos mejora el conocimiento de los precios de los activos, y que mejora la capacidad de predicción sobre métodos econométricos clásicos.

Por otro lado, realiza un estudio de la gran cantidad de predictores que se han amontonado a lo largo del tiempo que se pueden utilizar a la hora de realizar estas predicciones. En sus modelos, utiliza un set de predictores que supera los 900, incluyendo métricas propias del activo, de su sector y del mercado en general. En concreto, explica que las variables más relevantes son aquellas que explican las tendencias de los precios, la liquidez del mercado y su volatilidad. Respecto a este punto, demuestra que, cuando se tienen tantos predictores, se hacen necesarios la penalización o la reducción de dimensionalidad, que mejoran a los modelos sin ellas.

Además de estudiar el número de predictores, también explora la posibilidad de permitir que éstos se relacionen entre ellos a la hora de realizar la predicción. Al analizarlo, llega a la conclusión de que permitir relaciones potencialmente complejas entre las variables utilizadas en el modelo mejora su capacidad de predicción.
De hecho, el estudio llega a la conclusión de que las mejores técnicas para resolver el problema propuesto son las más complejas de entre las que se presentan –esto es, árboles de decisión y redes neuronales, mejorando estas últimas a los primeros–. En concreto, su enfoque para crear modelos de Deep Learning usando redes neuronales se centra en la utilización de redes "*feed-forward*". Esto es, redes sin recurrencia ni convolucionalidad como las explicadas en el apartado anterior (es decir, simplemente tiene capas de entrada, capas ocultas y capas de salida ordenadas de manera que los datos viajan en una sola dirección).

Sorprendentemente, el estudio concluye que, al contrario de lo que cabría pensar y de lo que ocurre cuando se aplica el aprendizaje profundo a muchos otros problemas, el añadir capas a las redes neuronales no es efectivo, dado que su efectividad disminuye desde que se llega a las tres capas ocultas y en adelante.

En resumen, el estudio realiza una revisión de la técnica hasta el momento de su publicación en lo que a la aplicación del aprendizaje automático al problema de la predicción de las primas de riesgo se refiere para luego, mediante el uso de diversas técnicas, realizar un análisis sobre un conjunto de datos muy extenso (60 años). Finalmente, el estudio concluye que los mejores métodos para resolver el problema son las redes neuronales y, en menor medida, los árboles de decisión, achacando su mejor ajuste a la capacidad de estas técnicas de derivar relaciones complejas entre los datos.

Al contrario de lo que descubre el estudio anterior respecto a que la capacidad de previsión de los modelos disminuye cuando se les añaden muchas capas, un nuevo estudio por los mismos autores, llamado "Modelos de precios de activos utilizando autocodificadores" (originalmente en inglés, "*Autoenconder Asset Pricing Models*", *Gu et al.*, 2021) que, utilizando un tipo de red neuronal mucho más innovador –y mucho más complejo—, para capturar relaciones no lineales más complejas que las que los modelos utilizados en el estudio anterior pueden captar.

Los autocodificadores son un tipo de modelos basados en dos partes principales, un codificador y un decodificador. El primero recibe los datos y trata de reducir su dimensionalidad sin perder información mientras que el segundo trata de reconstruir la entrada en base a los datos codificados por el primero. Este tipo de modelos son muy utilizados para buscar casos anómalos, y han sido aplicados antes en detección de fraude (*Mitra et al.*, 2022) o reconstrucción de imágenes (*Tan et al.*, 2010), entre otras muchas aplicaciones. Además, este estudio de Gu propone la aplicación de la restricción de no arbitraje, lo que supone que no puede existir en el mercado un retorno libre de riesgo. Con este modelo de Deep Learning más complejo consiguen, de hecho, mejorar el error sobre datos no vistos por el modelo respecto a los modelos existentes hasta el momento.

Por otro lado, estudiando los rendimientos esperados utilizando el propio nivel de las mismas aparecen estudios como "Diseccionando las anomalías" (originalmente "*Dissecting anomalies*", *Fama et al.*, 2008), más centrado en momentos con rendimientos extraños o anómales a lo largo de todos los tamaños de empresa. Entre sus conclusiones más notables se encuentran que sí parece haber una mayor rentabilidad para las empresas que presentan buenos rendimientos, mientras que empresas que no son tan rentables no tienen por qué mostrar rendimientos que sean, igualmente, inusualmente bajos.

Sobre predicción de acciones en el corto plazo, aunque aún a nivel diario, destaca el trabajo titulado "Predicción de tendencias a corto plazo del mercado de acciones usando un sistema de Deep Learning" (originalmente, "*Short-term stock market prediction using a comprehensive deep learning system*", *Shen et al.*, 2020). Aparte de realizar una revisión de la cuestión tanto a nivel técnico como a nivel financiero, realiza un estudio sobre la predicción del movimiento de determinadas acciones del mercado abierto chino. Para ello, realiza un preprocesado de los datos, una selección de variables así como una reducción de dimensionalidad mediante PCA para acabar realizando un modelo basado en LSTMs. Con este modelo, trata de predecir si el nivel sube o baja, alcanzando una precisión del entorno del 95%.

3.1.1 IMPACTO EN EL DISEÑO Y LA TOMA DE DECISIONES

Los estudios mencionados vienen a validar la hipótesis de la primera parte de este trabajo, que es que la aplicación de modelos de Deep Learning puede derivar en una buena predicción de los precios de mercado.

Por otro lado, resulta relevante darse cuenta de que los estudios presentados gozan de una amplitud mucho mayor en lo que disponibilidad de recursos se refiere. Se ha explicado en el apartado anterior las dificultades afrontadas a la hora de conseguir datos con la suficiente antigüedad, lo que contrasta con los utilizados por el estudio presentado. Esto

hace que, probablemente, estos estudios puedan ser más generalistas y aplicables a otros períodos, mientras que el presentado aquí es, necesariamente, inferior en cuanto a enfoque. En la misma línea, utilizan un número ingente de variables en sus modelos mientras que, como se ha comentado, para el modelo base de este trabajo tan solo se utilizan seis, pues tantas son las que se tienen disponibles. El reducido número de variables podría indicar que no será necesario utilizar métodos de penalización o reducción de variables. Este punto se comprobará más adelante, en el capítulo dedicado a la explicación del sistema desarrollado.

3.2 LITERATURA SOBRE PREDICCIÓN DE PRECIOS A PARTIR DE NOTICIAS

El primer trabajo relevante estaba titulado como "Predicción de la tendencia de las acciones utilizando análisis de sentimientos de noticias" (originalmente en inglés, "*Stock Trend Prediction using News Sentiment Analysis*", *Kalyani et al.*, 2016). Éste aborda la complejidad de la predicción de movimientos bursátiles utilizando métodos avanzados de aprendizaje automático y técnicas de minería de texto. La hipótesis subyacente del estudio es que las noticias financieras impactan significativamente en las tendencias del mercado de valores, y mediante la clasificación del sentimiento de las noticias se pueden prever estas tendencias.

La metodología adoptada implica el desarrollo de modelos de clasificación que determinan una clasificación de los artículos financieros como positivos o negativos. Sobre los métodos utilizados, se construyeron tres modelos utilizando los algoritmos de Naive Bayes, Random Forest y Máquinas de Vectores de Soporte (SVM), pudiendo derivar que Random Forest y SVM mostraron un mejor desempeño.

El estudio utilizó datos de tres años de Apple Inc., combinando precios de acciones diarios y artículos de noticias relevantes. A fin de realizar el análisis, se comenzó con la

recolección y el preprocesamiento de textos para ajustarlos a un formato utilizable para los modelos descritos. Este preprocesamiento incluyó la tokenización, eliminación de palabras irrelevantes (generalmente conocidas como "*stopwords*") y ruido, y la aplicación de técnicas de *stemming*, que tratan de reducir la palabra hasta su base. Sobre esta base, se aplicaron métodos para detectar el sentimiento mediante un enfoque basado en diccionario, utilizando un conjunto de palabras clasificadas como positivas o negativas.

Una vez procesados los datos, se implementaron los modelos de clasificación. Los resultados se evaluaron mediante diversas métricas como la precisión, el *recall* y el área bajo la curva ROC (conocida como "*area under the curve*" o AUC), con una precisión del modelo general superior al 80%. Se realizó también una predicción sobre un conjunto de datos no visto aún por el modelo. Los resultados alentadores indican que el modelo puede aumentar significativamente la precisión en la predicción de tendencias de las acciones en comparación con métodos que asignan etiquetas de manera aleatoria.

Finalmente, el estudio concluye que existe una relación significativa entre el sentimiento de las noticias y las tendencias del mercado de valores. Los modelos desarrollados pueden ayudar a los inversores y analistas a tomar decisiones más informadas basadas en el análisis de sentimientos de las noticias. Para trabajos futuros, se sugiere expandir la investigación a otras empresas y examinar el uso de datos de redes sociales como Twitter para análisis similares, así como su aplicación en el trading algorítmico.

Este trabajo guarda una gran relación con el trabajo que tenemos entre manos, aunque guarda también significativas diferencias, sobre todo en los datos utilizados y las técnicas de preprocesado de los datos aplicadas.

El siguiente estudio, titulado "Análisis de Sentimientos de Twitter y RSS News Feeds y su Impacto en la Predicción del Mercado de Valores" (originalmente en inglés, "*Sentiment Analysis of Twitter and RSS News Feeds and its Impacto on Stock Market Prediction*",

Bharathi et al., 2017) explora la influencia de los sentimientos expresados en Twitter y los feeds RSS sobre las predicciones del mercado de valores.

El enfoque propuesto busca mejorar la precisión en la predicción del mercado de valores al integrar los puntos del índice Sensex con análisis de sentimientos derivados de noticias RSS y tweets relacionados con el mercado. Los autores desarrollaron un modelo entrenado para predecir las tasas del mercado de valores, utilizando como datos las publicaciones en Twitter y los feeds de noticias RSS de la compañía ARBK de la Bolsa de Ammán (ASE). El estudio se basa en la hipótesis de que los indicadores de nivel de stock, cuando se combinan con el análisis de sentimientos de los feeds de noticias y tweets, pueden mejorar la precisión de las predicciones del mercado de valores.

La metodología empleada incluyó la recolección de precios del índice Sensex, tweets y feeds de noticias RSS durante un período determinado. Se utilizó un algoritmo para analizar la correlación entre estos datos y los valores del mercado de acciones. El análisis de sentimientos se llevó a cabo mediante técnicas de procesamiento de lenguaje natural para clasificar los sentimientos de los tweets y las noticias como positivos, negativos o neutrales.

Los resultados experimentales demostraron una mejora significativa del 20% en la precisión de la predicción al utilizar el análisis de sentimientos en combinación con indicadores de nivel de stock tradicionales. Este avance sugiere que la integración de datos de redes sociales y noticias puede proporcionar una ventaja significativa en la predicción de los movimientos del mercado de valores.

Al igual que el anterior, este trabajo es también parecido al trabajo presente, pues trata de predecir el precio de las acciones mediante el uso de modelos de aprendizaje automático utilizando procesamiento de texto. De hecho, este artículo es, quizá, más relevante que el

anterior, pues utiliza el mismo planteamiento que el presente de hacer un modelo base prediciendo sobre precios históricos y otro prediciendo sobre las noticias.

3.2.1 IMPACTO EN EL DISEÑO Y LA TOMA DE DECISIONES

A pesar de que, como se ha visto, los dos trabajos explicados tocan en mayor o menor medida el mismo tema que se trata en este trabajo, el presente innova sobre ellos en varios puntos.

Primero, mientras que los trabajos presentados utilizan precios de cierre diarios, lo que les da capacidad para predecir sobre un histórico mucho mayor, por las razones ya explicadas en este trabajo nos centramos en los precios intradía que, en mi opinión –fundamentada anteriormente–, son capaces de capturar mejor el impacto que las noticias tienen en el precio de cotización. Además, el primero ni siquiera realiza un modelo sobre datos históricos, sino que lo realiza directamente sobre las noticias.

Segundo, los trabajos presentados, quizá por su ligera antigüedad, utilizan técnicas que han quedado superadas por las utilizadas aquí de aprendizaje profundo, capaz de aprender relaciones mucho más complejas entre los datos.

Finalmente, aunque la estructura es parecida, creo que el enfoque tomado para predecir con las noticias sobre los errores cometidos por el modelo base es mejor, por las razones explicadas anteriormente, que el tomado por el segundo trabajo.

Sin embargo, no deja de ser cierto que, sobre la base de esta búsqueda y sus resultados, se pueden estudiar o entender más correctamente los modelos, los procesos, y las razones para realizar el trabajo como se ha realizado.

Capítulo 4. DEFINICIÓN DEL TRABAJO

A lo largo de los capítulos anteriores se ha llevado a cabo una introducción al proyecto, motivándolo y explicando las razones que llevan a su desarrollo; una descripción de las tecnologías en las que se basa para solucionar el problema del apartado anterior y, finalmente, una explicación de la situación técnica actual, del trabajo que se está haciendo actualmente en este ámbito y de las decisiones que se han tomado sobre el proyecto gracias a estos descubrimientos. En el capítulo actual, en cambio, se podría decir que entramos en materia: se justifica la realización del proyecto, los objetivos que pretende alcanzar, la metodología de trabajo seguida y, finalmente, las consideraciones temporales y económicas que han afectado a su compleción.

4.1 JUSTIFICACIÓN

Como se ha adelantado en el capítulo anterior, las tecnologías utilizadas son muy nuevas pero, dada su rápida implementación, no deja de sorprender la cantidad de información que se puede encontrar sobre ellas, lo que puede dar pie a pensar que ya está todo hecho. Sin embargo, siempre quedan partes por cubrir. Así, mientras los trabajos presentados en el capítulo anterior son útiles como base para la realización del proyecto, no presentan en realidad una solución funcional o no son aplicables en su conjunto a la solución presentada en este proyecto.

De hecho, leyendo estos y más artículos sobre el tema que estamos tratando, se da uno cuenta de que el enfoque tomado, así como su implementación con tecnologías novedosas, puede merecer la pena aunque sea por explorar en el campo de aplicación más allá de lo que lo hacen los trabajos referidos. Además, cabe la esperanza de que, utilizando estas nuevas técnicas, se llegue a una solución mejor que la conseguida en estos planteamientos.

4.2 ESTIMACIÓN ECONÓMICA

Si bien parece cierto que, para el escalado de la solución presentada y, potencialmente, para su mejora, sería necesario recurrir a recursos de pago, lo cierto es que el presente trabajo ha sido realizado con herramientas que, al menos en su versión utilizada, son gratuitas.

Ello ha hecho que la realización de este trabajo haya sido prácticamente gratuita, aunque habría que excluir el hecho de que la prueba realizada con la API de OpenAI sí requería de una provisión de fondos y éstos se consumían conforme era utilizados. Sin embargo, una vez visto que, de llevar a cabo una solución utilizando este recurso se iría de presupuesto (como he dicho, costaba unos 80 céntimos procesar cada noticia), se paró de utilizar, haciendo que el gasto haya sido, finalmente, bajo.

Puede el lector opinar que, a fin de realizar el trabajo, resulta obviamente necesario el tener un ordenador portátil y que éste tiene un coste. Este hecho, si bien es cierto, se podría responder especificando que el ordenador utilizado es el mismo que he tenido a lo largo de todo el desarrollo de mi titulación y, por lo tanto, opino que ya está más que amortizado.

Capítulo 5. SISTEMA/MODELO DESARROLLADO

En la siguiente sección se presenta una solución de diseño completa. Esta sección sirve para proporcionar una descripción detallada de los dos modelos (modelo base y modelo de noticias) y la integración de los mismos que lograron la solución de diseño final. Se trabajó seguidamente en cada modelo, de acuerdo con lo expuesto en el capítulo anterior sobre la planificación del trabajo, para luego integrarlos para lograr la solución de diseño final.

5.1 ANÁLISIS DEL SISTEMA

5.1.1 MODELO BASE

El primer modelo relevante de la solución de diseño presentada es el modelo base. Como se ha explicado anteriormente, este modelo recibe secuencias de datos históricos de acciones durante los 60 minutos anteriores y predice cuál será el precio de cierre durante los 10 minutos siguientes. Sin embargo, para realizar este modelo se han seguido varios pasos:

Primero, se realizó un modelo que hacía esto mismo pero prediciendo el siguiente minuto para una sola compañía. Es decir, utilizaba los 60 minutos anteriores para predecir el siguiente. Seguidamente, este modelo pasó a cambiarse para predecir para los 10 minutos siguientes. Después, se realizaron pruebas por si fuese útil utilizar métodos para prevenir el *overfitting* y, finalmente, se realizó el modelo presentado como modelo base, que utiliza datos de 10 empresas durante 60 minutos para predecir los 10 minutos siguientes para todas las empresas. Debajo, se explican más en profundidad cada uno de esos modelos.

En cualquier caso, antes de realizar cualquier modelo es siempre necesario realizar un preprocesado de los datos. Por ello, se explicará primero este paso para, después, poder explicar cómo son cada uno de los modelos.

5.1.1.1 Preprocesado de los datos

Como se ha explicado, los datos de precios históricos están guardados en formato JSON, conteniendo un diccionario que contiene, como claves, los tickers de las empresas y, como valores, otro diccionario con cada una de las variables –incluyendo fecha, apertura, cierre, máximo, mínimo y día de la semana–. Al descargar este diccionario, lo primero que se hace es quedarse con las empresas que son objeto de análisis. Después, se elimina la variable fecha que, mientras se guarde la secuencia temporal de los datos, no ofrece información adicional al modelo y, hecho esto, se escalan los datos de manera que ofrezcan valores entre 0 y 1, manteniendo la estructura pero ofreciendo un orden constante o extrapolable para todas las empresas. Este escalado se realiza mediante un MinMaxScaler, que aplica la siguiente ecuación:

$$valor\ escalado = \frac{valor\ real - mínimo}{máximo - mínimo}$$

De esta manera, el valor máximo tendrá un valor de 1 y el mínimo de 0, estando el resto de valores entre estos dos, con lo que se consigue mantener la estructura a pesar de la normalización.

Finalmente, para preparar los datos con un formato aplicable a cada modelo, se crea una función que crea las secuencias descritas con anterioridad –es decir, listas de listas conteniendo, para cada empresa, secuencias de 60 minutos en el pasado, cada una con los datos de las 6 variables de que se dispone–, así como las secuencias de salida (punto en el que unas funciones difieren de las otras, dependiendo del número de minutos que se quieran predecir en el futuro). En la Figura 2 se puede ver un ejemplo de esta función, en concreto el aplicado para el modelo base final.

```python
def create_sequences(data, sequence_length, out_length):
    input_sequences = []
    output_sequences = []
    for i in range(len(data) - sequence_length - out_length + 1):
        seq = data[i:(i + sequence_length)]
        target = data[(i + sequence_length):(i + sequence_length + out_length), :, 5]  # Assuming Close is at index 4
        input_sequences.append(seq)
        output_sequences.append(target)
    return np.array(input_sequences), np.array(output_sequences)

sequence_length = 59
out_length = 10
X_train, y_train = create_sequences(train_data, sequence_length, out_length)
X_test, y_test = create_sequences(test_data, sequence_length, out_length)
```

Figura 2: Ejemplo de función de creación de secuencias

Además de crear las secuencias, como se puede ver en la Figura 2, también se realiza una división entre datos de entrenamiento y de test, máxima que viene a ser de aplicación conforme a las buenas prácticas del aprendizaje automático para que se pueda realizar una predicción sobre datos desconocidos para el modelo. En todos los modelos de este trabajo se utiliza una proporción de 80-20 para esta división, que también suele ser habitual en este tipo de trabajos. Cabe destacar, para finalizar, que esta división, dada la estructura temporal de los datos, no se puede realizar de manera estocástica, sino que se deben dejar los últimos tiempos para predecir.

5.1.1.2 Modelo 60-1 para una empresa

A fin de probar que el modelo propuesto podía funcionar, así como a fin de dividir en partes el trabajo y no ir directamente a hacer el modelo final, más complejo, se realizaron primeramente modelos que predecían el precio para una sola empresa. Además, simplificando al máximo el problema, se realizó primero uno que predecía para una empresa y solamente durante el siguiente minuto. Este modelo está compuesto, como se puede ver en la Figura 3 bajo estas líneas, por una capa GRU de 32 neuronas, una Flatten para linealizar el resultado de ésta y, finalmente, una Dense con una neurona para sacar la predicción.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru (GRU)                   (None, 59, 32)            3840

 flatten (Flatten)           (None, 1888)              0

 dense (Dense)               (None, 1)                 1889

=================================================================
Total params: 5729 (22.38 KB)
Trainable params: 5729 (22.38 KB)
Non-trainable params: 0 (0.00 Byte)
```

Figura 3: Resumen de modelo 60-1 para una empresa

Este modelo fue compilado y entrenado. Dado el poco tiempo que tarda en entrenar, por recibir "pocos" datos en comparación con los modelos finales, se pudieron probar diferentes parámetros para tratar de mejorarlo, como la longitud de las secuencias de entrada, el número de épocas sobre las que debía entrenar, dar más y menos neuronas o más o menos capas, etcétera. Finalmente, el presentado en la Figura 3 fue el elegido.

Una vez entrenado, se predice sobre los conjuntos de train y test para comprobar la capacidad que tiene el modelo para ajustarse a los datos, tanto a los que ya ha visto como a los que no conoce. Predecir sobre ambos conjuntos puede ser de utilidad para descubrir fenómenos como el *overfitting*, del que se hablará más tarde en este apartado. Asimismo, los resultados de estas predicciones se presentarán en el siguiente capítulo, dedicado a este fin.

5.1.1.3 Modelo 60-10 para una empresa

Comprobado el funcionamiento de este modelo, se creó otro que predijese para los 10 minutos siguientes. La estructura de este modelo es muy similar a la anterior, con la

salvedad de que la capa de salida, como se puede ver en la Figura 4, tiene 10 neuronas pues tantos son los datos que debe predecir.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru_1 (GRU)                 (None, 59, 32)            3840

 flatten_1 (Flatten)         (None, 1888)              0

 dense_1 (Dense)             (None, 10)                18890

=================================================================
Total params: 22730 (88.79 KB)
Trainable params: 22730 (88.79 KB)
Non-trainable params: 0 (0.00 Byte)
```

Figura 4: Resumen de modelo 60-10 para una empresa

De igual manera al anterior, aunque este modelo tardaba algo más tiempo en entrenar, se probaron otras variables como el número de minutos a predecir, la longitud de la secuencia para predecirlos, etcétera.

5.1.1.4 Análisis de prevención de overfitting

Tomando como modelo el trabajo de Gu et al., 2018, presentado en el estado de la cuestión, en lo que se refiere a la tendencia de las redes neuronales a producir *overfitting*, se decidió crear un modelo que explorase este punto, para ver si técnicas de prevención de este fenómeno podían ayudar a mejorar la capacidad de generalización del modelo para datos que no conoce. Aunque hay técnicas sencillas como evitar que el modelo entrene más épocas de las que necesita y otras por el estilo, las que demuestran tener más eficacia son las capas de Dropout explicadas en capítulos anteriores.

Por ello, se creó un modelo de este tipo, cuya estructura básica es similar a la de los otros pero que añade las capas mencionadas. Se puede ver un resumen de su estructura en la Figura 5.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru_2 (GRU)                 (None, 59, 32)            3840

 dropout (Dropout)           (None, 59, 32)            0

 flatten_2 (Flatten)         (None, 1888)              0

 dense_2 (Dense)             (None, 10)                18890

=================================================================
Total params: 22730 (88.79 KB)
Trainable params: 22730 (88.79 KB)
Non-trainable params: 0 (0.00 Byte)
```

Figura 5: Resumen de modelo con prevención de overfitting

Sin embargo, , los resultados obtenidos no eran los esperados, al dar coeficientes de determinación inferiores para ambos conjuntos de datos comparados con los de los otros modelos comparables, por lo que se descartó este método y se concluyó, por tanto, que el modelo no comete *overfitting*. Este hecho, como se ha apuntado en el capítulo sobre el Estado de la Cuestión, puede deberse al reducido número de variables con las que cuenta el modelo.

5.1.1.5 Modelo base final (modelo 60-10 para 10 empresas)

Finalmente, se creó el modelo final para la predicción para varias empresas del precio de cierre durante los siguientes diez minutos utilizando los 60 anteriores. Este modelo añade el concepto de convolucionalidad, explicado con anterioridad, metiendo una capa para que

el modelo generalizase para todas las empresas a la vez. Su estructura se puede comprobar en la Figura 6.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 reshape (Reshape)           (None, 59, 10, 6)         0

 time_distributed (TimeDist  (None, 59, 10, 10)        70
 ributed)

 time_distributed_1 (TimeDi  (None, 59, 100)           0
 stributed)

 gru (GRU)                   (None, 128)               88320

 dense (Dense)               (None, 100)               12900

 reshape_1 (Reshape)         (None, 10, 10)            0

=================================================================
Total params: 101290 (395.66 KB)
Trainable params: 101290 (395.66 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figura 6: Resumen de modelo base final

Igual que con los anteriores, los resultados de este modelo serán presentados en el siguiente capítulo. Sin embargo, cabe destacar aquí que este modelo ya comienza a ser más grande y, de hecho, es donde se establece el límite para el número de empresas elegidas, a fin de que no se eternizase el entrenamiento de los modelos y a fin, asimismo, de que los recursos computacionales de que se disponen fuesen suficientes para entrenarlo.

5.1.2 MODELO DE NOTICIAS

El modelo de noticias es el otro gran subsistema del sistema total propuesto en este trabajo. Como se ha explicado anteriormente, recibe secuencias de las noticias publicadas durante los 60 minutos anteriores al momento en cuestión y se predice, con estructura similar a los otros modelos, el error cometido por el modelo base.

De esta manera, se podrían distinguir tres pasos necesarios para llegar a tener un modelo de este tipo. El primero es la generación de las secuencias de entrada –esto es, crear las secuencias de noticias y estructurarlas de tal manera que sean accionables por una computadora–. El segundo es la creación de las secuencias de salida, que no son otra cosa que los residuos del modelo base. Finalmente, el tercero es la propia creación del modelo. Se pasa ahora a explicar lo realizado en cada uno de estos pasos.

5.1.2.1 Preprocesado de noticias

La creación de las secuencias de noticias es, quizá, uno de los pasos más complicados en todo el diseño, pues para él hay que tener en cuenta una gran cantidad de factores.

Por un lado, se debe decidir si se quiere que todas las noticias publicadas para cualquiera de las empresas puedan afectar al resto. En este caso, se creyó que esto simulaba mejor la realidad, por lo que se juntaron todas las noticias independientemente de su origen.

Por otro lado, como es obvio, el mercado no está siempre abierto y, por lo tanto, las noticias publicadas en el tiempo en que el mercado está cerrado no tienen impacto hasta que éste se vuelve a abrir. Por ello, se deben modificar las fechas de publicación de las noticias para tener en cuenta este hecho. Esto reviste cierta dificultad pues, no es sólo que se deba tener en cuenta que las noticias publicadas entre el cierre del mercado una tarde y su reapertura la mañana siguiente deben ser consideradas como si hubiesen sido publicadas en este momento, sino que el mercado tampoco abre los fines de semana, por lo que hay

que tener en cuenta que las noticias publicadas entre el cierre un viernes y la reapertura un lunes deben ser consideradas como si se hubiesen publicado el lunes.

Además, los propios datos históricos pueden tener ciertos agujeros en su estructura, pues la forma de conseguirlos no es perfecta. Esto afecta pero sólo en cierta manera, pues esos huecos, simplemente, no serán tenidos en cuenta.

Finalmente, existe la posibilidad de que varias noticias se haya publicado en un mismo minuto. Por lo tanto, se debe resolver este punto de alguna forma. En este caso, lo que se ha hecho es considerar todas las noticias publicadas en un mismo minuto como si fuesen una misma noticia, uniéndolas en un mismo texto separadas por un salto de línea.

Con estas ideas en mente, se pueden empezar a crear las secuencias. Sin embargo, no hay que olvidarse del hecho de que un ordenador no puede procesar texto sino números, por lo que se debe hacer esta transición. Para ello, se utiliza un tokenizador, que asigna a cada palabra un número asociado, como si fuese un diccionario. Este tokenizador debe ser 'entrenado' sobre un conjunto de palabras, que pasan a ser todas las palabras que 'conoce', descartando el resto. Obviamente, la buena práctica es utilizar sólo las palabras que se encuentren en el *corpus* de noticias del conjunto de train, y aplicar este mismo tokenizador después al conjunto de test.

Con todo esto, ya se pueden tener las secuencias generadas. En este caso, como el tamaño de vocabulario elegido eran 1000 palabras diferentes, la forma de estas secuencias era:

$$(nº\ de\ muestas, nº\ de\ minutos, tamaño\ del\ vocabulario)$$
$$= (nº\ de\ muestras, 59, 1000)$$

5.1.2.2 Cálculo de residuos

Para calcular los residuos debe, el modelo anterior, ser aplicado al tiempo nuevo (que, como se ha explicado anteriormente, es mayo de 2024), generando una serie de

predicciones. Si estas predicciones se restan a los datos reales se obtienen los errores cometidos o, lo que es lo mismo, los residuos del modelo.

Estos residuos deben, después, ser divididos en conjuntos de entrenamiento y de test, como en cualquier modelo de aprendizaje automático.

5.1.2.3 Creación del modelo de noticias

Finalmente, una vez creadas las secuencias de entrada y de salida, se pasó a la creación del modelo. Este modelo, por ser más complejo que los demás, se montó de manera funcional y no secuencial.

La estructura del modelo es comprensible si se entiende cuál es la finalidad que persigue. Lo primero es una capa de Embedding que, como se ha explicado, mapea el tamaño de vocabulario de 1000 en tan solo 128 dimensiones, lo que lo hace más manejable. Luego, el resultado es pasado a una capa de LSTMs con 128 neuronas que aprenden sobre el texto. Después, tras linealizar el resultado de esta capa, se les pasa a unas últimas capas de neuronas Dense interconectadas. La primera de ellas también sirve para sacar conclusiones y la última es para darle al resultado la forma que debe tener. En la Figura 7 se puede ver un resumen de la estructura del modelo descrito.

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 news_input (InputLayer)     [(None, 59, 1000)]        0

 embedding_layer (TimeDistr  (None, 59, 1000, 128)     128000
 ibuted)

 lstm_layer (TimeDistribute  (None, 59, 128)           131584
 d)

 flatten_layer (Flatten)     (None, 7552)              0

 dense_layer_1 (Dense)       (None, 64)                483392

 dense_layer_out (Dense)     (None, 100)               6500

 reshape (Reshape)           (None, 10, 10)            0

=================================================================
Total params: 749476 (2.86 MB)
Trainable params: 749476 (2.86 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figura 7: Resumen de modelo de noticias

Cabe destacar que este modelo sí es significativamente superior en tamaño a los anteriores y, de hecho, requería de la utilización de sistemas con GPU para su entrenamiento. Incluso en estos sistemas, requería de cierto tiempo para terminar de entrenar. Es, por tanto, el modelo último que pone restricciones al tamaño final de la lista de activos elegidos para realizar el estudio.

5.1.3 INTEGRACIÓN DE LOS SISTEMAS

Una vez que los dos subsistemas funcionaron por separado, la fase final del diseño consistió en integrarlos para testear su funcionamiento de extremo a extremo.

Para la solución de diseño final, el valor de las predicciones de uno y otro modelo se sumaron para tener las predicciones finales que, comparándose con las realizadas por el

modelo sencillo, determinarían si la adición de noticias mejora o no la predicción por medio de la serie histórica.

5.2 IMPLEMENTACIÓN

Desde la introducción inicial al proyecto propuesto hasta la solución de diseño final, hubo varios cambios por los que pasó la solución para lograr el producto final. Se tuvieron que examinar muchos *tradeoffs* como el tiempo, las restricciones presupuestarias y los conocimientos limitados a la hora de ejecutar el proyecto.

5.2.1 DECISIONES DE DISEÑO

Todas las soluciones de diseño se han realizado conforme a lo que se consideran buenas prácticas en sus campos y conforme al mejor conocimiento del autor, teniendo también en cuenta las restricciones temporales, presupuestarias y computacionales que establecían los medios de que se disponía.

Si bien inicialmente se esperaba poder hacer un modelo que tuviese en cuenta más empresas para poder sacar interrelaciones entre ellas y tener más datos de los que aprender, todas estas restricciones hicieron que, finalmente, no pudiese hacerse como era deseado. Asimismo, aunque las mejores prácticas indican que para un problema de este tipo deberían ser necesarios más datos, la imposibilidad para conseguirlos impuso el espacio temporal utilizado finalmente para crear los modelos.

5.2.2 OBSTÁCULOS DE DISEÑO

Como se ha adelantado a lo largo del texto, los obstáculos a los que me he enfrentado a la hora de poder conducir el trabajo como quería han sido, esencialmente, dos.

El primero de ellos es la dificultad de conseguir datos históricos intradía y noticias con cierta antigüedad. Tener más datos habría permitido a los modelos aprender más y

descubrir patrones que, quizá, en el período de tiempo en que, finalmente, han sido entrenados pueden no aparecer.

Sin embargo, el segundo de ellos impone otro límite a este punto. Incluso si se hubiese dispuesto de más datos, los recursos computacionales de los que se disponía hacían imposible la utilización de todos ellos, por lo que finalmente ha sido este inconveniente el que más ha pesado a la hora de realizar el trabajo.

De todas formas, me gustaría resaltar que, incluso habiendo sido enfrentados estos obstáculos, ha sido finalmente posible poder realizar el trabajo hasta el final y de manera satisfactoria. Es decir, se han podido entrenar los modelos (eso sí, más o menos aplicables) y se han podido obtener unos resultados (igualmente, más o menos significativos), por lo que, a pesar de las dificultades, éstas no han 'hundido' el proyecto, habiendo sido estos obstáculos, en mi opinión, capeados con eficacia, lo que ya de por sí considero un logro en sí mismo.

Capítulo 6. ANÁLISIS DE RESULTADOS

Si en el anterior capítulo se ha llevado a cabo un análisis de la solución a implementar y de la manera de hacerlo, en este estamos un paso más allá, habiendo completado la implementación y comprobando los resultados.

Resulta obvio que los dos capítulos, el anterior y el presente, no han ocurrido de manera absolutamente separada ni indistinta, sino que se podría decir que se ha pasado de uno al otro y viceversa, mejorando el proyecto según resultados preliminares y obteniendo los nuevos resultados de los cambios implementados. Sin embargo, al igual que en el capítulo anterior se ha mostrado el diseño final a implementar, con todos estos cambios ya implementados, en éste se presentan los resultados de dicha implementación –y no de las intermedias que han llevado hasta la última–.

Como se comentará más adelante en el capítulo sobre las Conclusiones y los Trabajos Futuros a realizar, no se pretende, ni mucho menos, decir que esta implementación sea perfecta, sino simplemente que, dadas las circunstancias temporales, técnicas y económicas, es la mejor a la que el autor ha podido llegar.

En cuanto a la estructura de este capítulo, se estructurará de manera primera en tres partes. Primero, se presentará el método utilizado para testear los resultados para luego pasar a presentar, de cada uno de los modelos, los propios resultados y finalizar haciendo un análisis de los mismos que incluya, en su caso, pequeñas recomendaciones o medidas adoptadas para mejorarlos.

Con todo, pasemos ahora a estudiar el primero de los apartados, el método utilizado para comprobar el correcto funcionamiento de los modelos.

6.1 MÉTODO

El método para comprobar los resultados de los modelos es similar entre unos y otros. En concreto, se utiliza una métrica llamada coeficiente de determinación o R^2. Ésta es una medida estadística que indica la proporción de la variación en una variable dependiente que es predecible a partir de las variables independientes en un modelo de regresión. Este coeficiente proporciona una indicación de cómo de bien los resultados observados son replicados por el modelo, basándose en la proporción de la variación total de los resultados explicada por el modelo.

El valor de R^2 varía entre $-\infty$ y 1, donde un R^2 de 1 indica que el modelo explica perfectamente la variabilidad de los datos respecto a la media, mientras que un R^2 de 0 significa que el modelo no explica mejor la variabilidad que un modelo simple que solo utiliza la media de los datos observados como predicción para todos los casos.

R^2 es ampliamente utilizado para evaluar la bondad de ajuste de un modelo de regresión, ayudando a determinar qué tan bien el modelo seleccionado se ajusta a los datos reales. Es especialmente útil en la comparación de modelos, proporcionando una métrica clara para evaluar y comparar la efectividad de diferentes modelos estadísticos o de aprendizaje automático en términos de su capacidad para explicar la variabilidad en el conjunto de datos. Es por esto que es la métrica elegida para evaluar los modelos, porque permite tanto compararlos como dar una visión por sí mismo de cuán bien se está comportando el modelo.

6.2 MODELO BASE

Pasemos ahora a comprobar los resultados del primero de los modelos. Como se ha explicado en el capítulo anterior, para terminar de realizar este modelo se realizaron primero una serie de modelos intermedios. Por tanto, se explicarán los resultados de cada uno de éstos.

6.2.1 MODELO 60-1

En la Tabla 1 se pueden comprobar los resultados obtenidos para el coeficiente de determinación.

Tabla 1: Coeficiente de determinación del modelo 60-1

	R^2
train	98.62%
test	99.53%

Estos resultados parecen, a primera vista, sorprendentes y, más sorprendente quizá, el hecho de que la determinación del test sea mayor incluso que la del entrenamiento. Esto puede deberse, simplemente, a que el conjunto de test sea realmente parecido al de entrenamiento.

Siendo aparentemente impresionantes, cabe destacar dos hechos. Primero, como se ve en la Figura 8, que incluye un extracto de las predicciones de test y los valores reales, se puede dar cuenta de que, realmente, el error es mínimo, pero también son mínimas las diferencias entre unas observaciones y otras..

	Actual	Predicted
0	0.163686	0.169282
1	0.162993	0.168320
2	0.162993	0.167251
3	0.164158	0.164415
4	0.168814	0.162992
...	...	...
2910	0.903450	0.898072
2911	0.904090	0.898157
2912	0.903450	0.902044
2913	0.899375	0.900868
2914	0.902576	0.902379

Figura 8: Extracto de resultados del modelo 60-1

Igualmente, cuando se miran las Figuras 9 y 10, donde se pueden ver graficados los valores predichos y los reales tanto para el conjunto de entrenamiento y para el de test, vemos que, efectivamente, la predicción está bien ajustada.

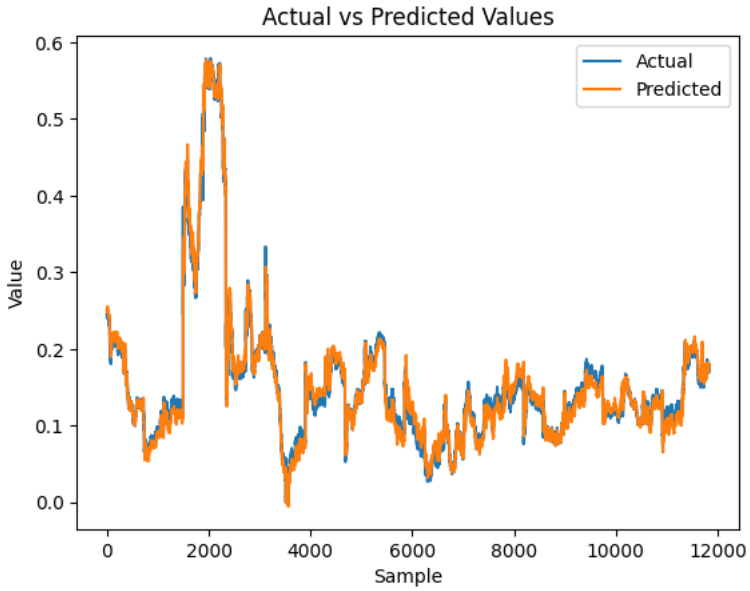

Figura 9:Valores reales y predichos del conjunto de entrenamiento del modelo 60-1

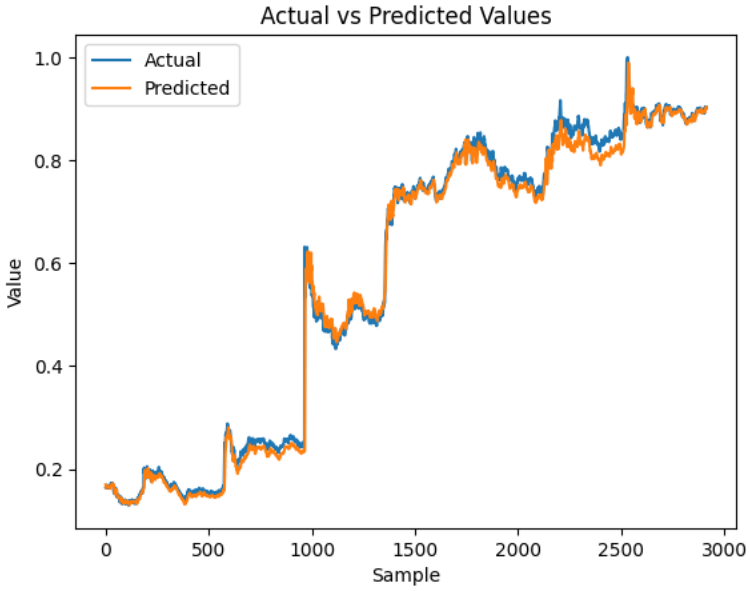

Figura 10: Valores reales y predichos del conjunto de test del modelo 60-1

Sin embargo, aunque parezca inicialmente que los resultados del modelo son realmente buenos, no hay que olvidar que, como se puede comprobar también en la Figura 8, los valores de un minuto para otro varían entre poco y nada, por lo que es normal que un modelo a tan corto plazo sea capaz de predecir con tanta precisión los modelos.

De hecho, se ha mencionado en el primer capítulo que este trabajo es de aplicación a *market makers* que quieran estimar y ajustar el *spread* que deben cobrar y es, precisamente en este aparentemente pequeño error, donde estaría la diferencia para ellos entre salir ganando o perdiendo con una operación. Lo que se quiere decir con esto es que, si bien los resultados parecen buenos, esto no sólo es normal sino que puede no ser suficiente.

6.2.2 MODELO 60-10 PARA UNA EMPRESA

De igual manera que para el modelo anterior, debajo se puede ver la Tabla 2, que muestra los coeficientes de determinación para entrenamiento y test del modelo que predice a 10 minutos para una sola empresa.

Tabla 2: Coeficientes de determinación del modelo 60-10 para una empresa

	R^2	
	train	test
Minuto 1	99.06%	99.60%
Minuto 2	99.00%	99.63%
Minuto 3	98.55%	99.42%
Minuto 4	98.35%	99.36%
Minuto 5	98.14%	99.31%
Minuto 6	98.04%	99.33%
Minuto 7	97.34%	98.66%
Minuto 8	96.70%	97.74%
Minuto 9	96.33%	97.90%
Minuto 10	95.96%	96.97%

Sigue resultando interesante que, en general, los R^2s del conjunto de test mejoran a los del conjunto de train. Otro hecho significativo es que, para los datos conocidos por el modelo –esto es, los del conjunto de entrenamiento–, la determinación va bajando conforme se predice a un horizonte superior, lo que tiene sentido dado que los datos anteriores dejan de ser tan significativos para poder estimar la evolución futura y, por otro lado, empiezan a separarse los valores de los datos utilizados como entrada y los valores en ese minuto.

Igualmente, en las Figuras 11 y 12 se pueden ver graficados los ajustes de los datos con las predicciones.

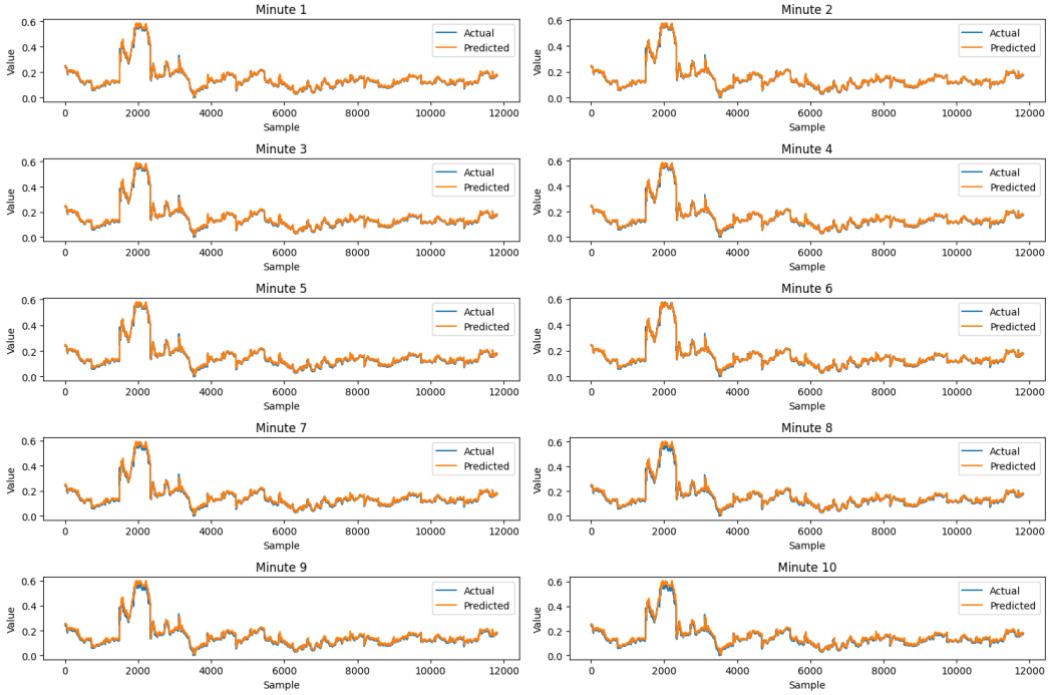

Figura 11: Valores reales y predichos del conjunto de train del modelo 60-10 para una empresa

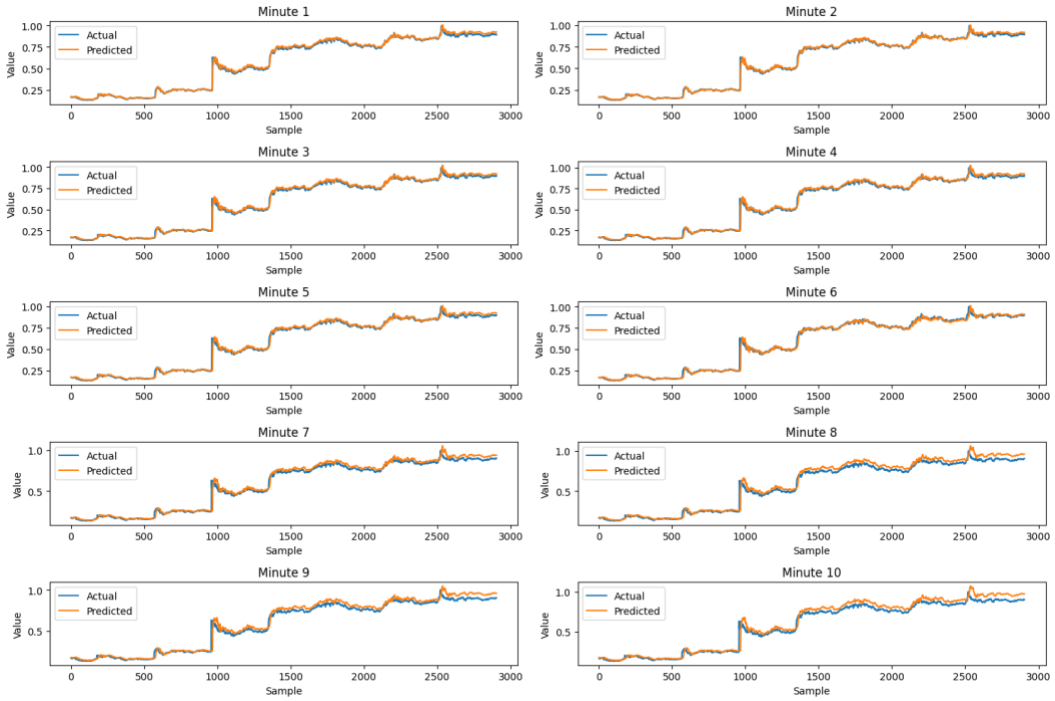

Figura 12: Valores reales y predichos del conjunto de test del modelo 60-10 para una empresa

Se puede comprobar que todos los valores predichos se ajustan significativamente a los valores reales, aunque esto se puede seguir achacando a las razones antes explicadas. Además, se puede ver en la Figura 12 que, en las predicciones para horizontes mayores, la realidad queda ligeramente por debajo de la predicción o, lo que es lo mismo, se predice un valor superior al real. Si no perdemos de vista el público al que va dirigido el trabajo, podemos darnos cuenta de cuán perjudicial puede ser este hecho, aunque parezca insignificante. Lo cierto es que si el precio predicho es unos céntimos superior al real, el *trader* puede estar perdiendo su margen o, incluso, perdiendo dinero con la operación. No se puede confirmar, sin embargo, si este es un hecho puntual sólo aplicable a esta empresa o este momento o se trata de una tendencia general del modelo.

6.2.3 MODELO BASE FINAL

Como se ha explicado anteriormente, una vez tomados los pasos anteriores para desarrollar el modelo se pasó al modelo que debía servir como final para el trabajo, similar al anterior

pero añadiendo varias empresas (10 en concreto). Los resultados del coeficiente de determinación del modelo se pueden ver bajo estas líneas, en la Tabla 3.

Tabla 3: Coeficientes de determinación del modelo 60-10 para varias empresas

	R^2	
Compañía	Media train	Media test
MMM	98.30%	98.21%
AAPL	99.28%	96.46%
MSFT	99.13%	84.93%
ABNB	98.92%	74.44%
FDX	99.31%	77.13%
AMZN	99.31%	72.28%
JNJ	99.70%	92.21%
JPM	99.57%	93.41%
PG	98.76%	90.52%
MRNA	98.87%	95.29%

Si bien es cierto que los resultados obtenidos para este modelo son significativamente inferiores, sobre todo en el conjunto de test, que los obtenidos en los modelos anteriores, también es cierto que se trata de un modelo bastante más complejo.

Dada la dificultad de presentar todas las gráficas para todos los minutos de todas las compañías, no se presentarán éstas, pero en ellas se podría ver que, en ciertos casos, la predicción no se ajusta perfectamente a, aunque sí está en el entorno de, el dato real, por lo que podríamos ver los resultados ligeramente peores del modelo.

6.2.4 RECOMENDACIONES Y/O MEDIDAS ADOPTADAS

La primera recomendación y, quizá más importante, es incrementar la cantidad de datos con la que se entrena el modelo. De hecho, inicialmente estos modelos se entrenaron con datos a lo largo de un intervalo de tiempo menor pues, como se ha dicho anteriormente, los

datos se han tenido que ir recopilando escaladamente. Cuando se entrenaron los modelos de esta manera, los resultados ofrecidos eran significativamente peores. Esto demuestra que, conforme se le dan más datos al modelo, éste es capaz de aprender relaciones no presentes en un intervalo menor, lo que hace que se vaya haciendo más generalista a datos no vistos hasta el momento.

Como sabemos por haber sido ya explicado en ocasiones anteriores, estas ligeras mejoras de predicción, aunque parezcan cuantitativamente pequeñas, pueden significar la diferencia entre que salga rentable o no la operación en el corto plazo, por lo que este extremo es de gran relevancia.

6.3 MODELO DE NOTICIAS

Habiendo visto el resultado del modelo anterior, puede darse cuenta el lector que difícilmente podrá este mejorar las predicciones. Sin embargo, como se ha comentado, es precisamente en estas pequeñas mejorías donde se encuentra el valor y, de hecho, es precisamente lo que este trabajo trata de añadir.

6.3.1 PREDICCIÓN DE RESIDUOS

Como se ha explicado en otras ocasiones, el modelo de noticias trata de predecir no los precios en sí, sino los residuos de las predicciones del modelo anterior. Por lo tanto, el modelo entrenado en el apartado anterior se aplica a datos nuevos, entrenándolo para el conjunto de train de éstos y prediciendo para ambos conjuntos, el de entrenamiento y el de test. Los resultados arrojados se pueden comprobar en la Tabla 4.

Tabla 4: Coeficientes de determinación del modelo 60-10 para varias empresas aplicado a datos nuevos

	R^2	
Compañía	Media train	Media test
MMM	99.47%	93.36%
AAPL	99.11%	-23.21%
MSFT	99.69%	96.08%

ABNB	99.49%	80.79%
FDX	99.55%	92.84%
AMZN	98.53%	95.01%
JNJ	99.08%	80.73%
JPM	99.45%	71.15%
PG	98.06%	82.95%
MRNA	99.73%	88.96%

Como se puede comprobar, el modelo reentrenado mejora en prácticamente todos los casos el coeficiente de determinación en el conjunto de entrenamiento y, en muchos de ellos, en el conjunto de test, lo que sigue apoyando la hipótesis de que, a fin de terminar de mejorar, estos modelos deberían ser entrenados con más datos. Resulta curioso el caso de AAPL, en el cual el ajuste del modelo a su conjunto de test es nefasto. Esto, sin embargo, se puede deber a que, durante el tiempo que dura el conjunto de test, la acción ha tenido un comportamiento extraño que le modelo no conoce mediante su entrenamiento previo. En cualquier caso, estos serán los datos que se utilicen para calcular los residuos y, con ellos, entrenar el modelo de noticias.

6.3.2 PREDICCIÓN DEL MODELO DE NOTICIAS

Si bien los resultados a nivel de coeficiente de determinación del modelo de noticias se presentarán a nivel de predicción conjunta –es decir, sumando las predicciones del modelo base y el de noticias—, se presenta en la Figura 13 un extracto del entrenamiento del modelo.

```
Epoch 1/100
159/162 [============================>.] - ETA: 0s - loss: 5.1318e-04Accumulating data for smo
162/162 [==============================] - 6s 9ms/step - loss: 5.0779e-04 - val_loss: 0.0016
Epoch 2/100
156/162 [============================>..] - ETA: 0s - loss: 3.7634e-04Accumulating data for smo
162/162 [==============================] - 1s 5ms/step - loss: 3.7790e-04 - val_loss: 0.0015
Epoch 3/100
157/162 [============================>.] - ETA: 0s - loss: 3.7599e-04Accumulating data for smo
162/162 [==============================] - 1s 5ms/step - loss: 3.7188e-04 - val_loss: 0.0013
Epoch 4/100
154/162 [============================>..] - ETA: 0s - loss: 3.5477e-04Accumulating data for smo
162/162 [==============================] - 1s 5ms/step - loss: 3.6136e-04 - val_loss: 0.0016
Epoch 5/100
155/162 [============================>..] - ETA: 0s - loss: 3.4880e-04Accumulating data for smo
162/162 [==============================] - 1s 5ms/step - loss: 3.5073e-04 - val_loss: 0.0013
Epoch 6/100
154/162 [============================>..] - ETA: 0s - loss: 3.5046e-04Validation loss decreased
162/162 [==============================] - 1s 6ms/step - loss: 3.4652e-04 - val_loss: 0.0014
Epoch 7/100
162/162 [==============================] - 1s 7ms/step - loss: 3.3925e-04 - val_loss: 0.0015
Epoch 8/100
162/162 [==============================] - 1s 7ms/step - loss: 3.3698e-04 - val_loss: 0.0014
Epoch 9/100
162/162 [==============================] - 1s 8ms/step - loss: 3.4163e-04 - val_loss: 0.0015
Epoch 10/100
162/162 [==============================] - 1s 7ms/step - loss: 3.3684e-04 - val_loss: 0.0013
Epoch 11/100
162/162 [==============================] - 1s 5ms/step - loss: 3.4442e-04 - val_loss: 0.0014
Epoch 12/100
162/162 [==============================] - 1s 5ms/step - loss: 3.4067e-04 - val_loss: 0.0017
Epoch 13/100
162/162 [==============================] - 1s 5ms/step - loss: 3.3974e-04 - val_loss: 0.0015
Epoch 14/100
162/162 [==============================] - 1s 6ms/step - loss: 3.2784e-04 - val_loss: 0.0015
Epoch 15/100
162/162 [==============================] - 1s 6ms/step - loss: 3.2898e-04 - val_loss: 0.0016
```

Figura 13: Proceso de entrenamiento del modelo de noticias

Como se puede ver, ni el error de validación ni el de train (representado por *loss*) bajan a lo largo de las épocas. Esto parece indicar que el modelo o bien no está aprendiendo o bien está comentiendo *overfitting*. Dado que el modelo no es capaz de bajar prácticamente nada el error desde el principio, podemos descartar el que esté cometiendo *overfitting*, por lo que podemos asumir que no está aprendiendo –es decir, que comete *underfitting*—. Las razones de ello pueden ser varias, incluyendo como posibles la falta de suficientes datos para aprender o que el modelo sea demasiado simple para ser capaz de sacar las relaciones entre los datos.

En cualquier caso, dada la imposibilidad de resolver cualquiera de estos problemas por estar limitados por obstáculos insalvables –la falta de tiempo nos impide coger más datos para entrenar sobre un conjunto mayor y la falta de recursos computacionales mejores hace que entrenar un modelo más complejo sea irrealizable—, debemos conformarnos con este resultado.

6.3.3 RECOMENDACIONES Y/O MEDIDAS ADOPTADAS

Las recomendaciones respecto a este modelo van en línea a intentar salvar los obstáculos presentados anteriormente. Es decir, se ha de tratar de conseguir más datos o de complicar el modelo hasta un punto en el que sea capaz de sacar información.

Además, resulta relevante apuntar que, como se ha mencionado anteriormente, el intervalo de tiempo en el que se ha entrenado este modelo es durante el mes de mayo de 2024. Hay varios apuntes que hacer a esto. El primero es que es un conjunto de tiempo realmente pequeño, dado que en un mes no es posible ver todos los 'tipos de noticias' que puede haber. El segundo es que este intervalo no incluye el tiempo en el que las noticias pueden afectar más significativamente al precio de las acciones, como es la época de presentación de resultados. Sería conveniente, por lo tanto, tanto incrementar la base de noticias que se tienen como realizar este mismo estudio en una época donde se crea que las noticias pueden ser más relevantes para mover los precios.

6.4 PREDICCIÓN CONJUNTA

Habiendo entrenado los dos modelos anteriores, como se ha explicado con anterioridad, se suman los resultados para obtener las predicciones conjuntas. De igual manera que para los modelos anteriores, en la Tabla 5 se presentan los resultados del coeficiente de determinación obtenido para las diferentes compañías.

Tabla 5: Coeficientes de determinación de la predicción conjunta

	R^2

Compañía	Media train	Media test
MMM	98.90%	89.73%
AAPL	99.05%	-28.61%
MSFT	99.67%	96.01%
ABNB	99.42%	82.46%
FDX	99.46%	93.31%
AMZN	98.45%	94.84%
JNJ	98.93%	81.84%
JPM	99.43%	70.29%
PG	98.89%	81.41%
MRNA	99.62%	89.99%

Puede el lector comprobar que los resultados obtenidos empeoran en, aproximadamente, la mitad de los casos las predicciones obtenidas en el conjunto de test. Sin embargo, si bien no deja de ser cierto que estas bajadas de determinación son mínimas, como se viene repitiendo son muy relevantes. También es cierto que, en los casos en que la determinación sube, lo hace por un margen igualmente exiguo.

Las recomendaciones siguen siendo las mismas que en el apartado anterior: debe mejorarse el modelo de noticias, sea mediante la consecución de más datos de noticias, la aplicación del mismo a otra época o la complicación del modelo para sacar relaciones más complejas entre los datos.

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

Para concluir esta memoria, se presentan ahora las conclusiones que se han sacado del trabajo realizado, así como posibles mejoras a futuro que, durante el desarrollo del proyecto, han surgido y que las limitaciones en cuanto a tiempo, presupuesto y recursos del proyecto que ahora concluye no han permitido realizar.

Primero, la conclusión que se saca es que el modelo de noticias, en líneas generales, ni mejora ni empeora la predicción utilizando los datos históricos de cotización. En cualquier caso, el hecho de que mejore en ciertos casos puede dar esperanza a pensar que, hecho de otra forma, podría aportar. Como se ha mencionado en varias ocasiones, cada pequeña mejoría es realmente significativa, pues puede significar la diferencia entre perder o no dinero en una operación, por lo que merece la pena explorar más este campo a fin de cerciorarse de si puede ayudar a reducir la posibilidad de pérdida y, por tanto, a ajustar los márgenes.

Por ello, los trabajos futuros van en esta línea, tratar de conseguir más datos y de hacer un modelo más complejo para poder realizar un modelo que sea de verdad significativo. De hecho, en mi opinión el factor que más influye es el tiempo, ya que, como se ha mencionado anteriormente, las noticias que se publican en un mes no pueden dar al modelo amplitud suficiente para predecir el error del modelo base en todos los casos. Por ello, lo primero que debería hacerse a futuro es seguir guardando datos, tanto de precios históricos como de noticias, para poder entrenar un modelo sobre más cantidad de datos que, esta vez sí, incluyan posibles eventualidades que el modelo prediciendo sobre un solo mes no ha presenciado. Particularmente relevante sería conseguir datos de épocas de publicación de resultados, pues es ahí donde realmente toma importancia este tipo de modelos, ya que las noticias de este tipo mueven realmente, de un momento para otro, el mercado.

Haciendo esto, los modelos (ambos) se irán haciendo cada vez más resistentes y generalistas por haber visto más posibilidades por lo que, probablemente, mejore su predicción sobre datos no vistos.

Aunque la solución de diseño se considere satisfactoria, pues se han podido completar los objetivos de desarrollo que se definieron al principio, así como validar de manera preliminar ciertas hipótesis, se puede comprobar en este mismo capítulo que aún pueden introducirse mejoras para perfeccionar el prototipo. Llegando al final de este trabajo, no se puede sino animar al lector que haya estudiado este proyecto a llevar a cabo las mejoras mencionadas y otras muchas que se le hayan ocurrido y que el autor no haya sido capaz de encontrar satisfactoriamente.

Capítulo 8. BIBLIOGRAFÍA

[1] ¿Qué es Deep Learning? (2023, mayo 16). IBM. Disponible en: https://www.ibm.com/es-es/topics/deep-learning

[2] ¿Qué es el procesamiento de lenguaje natural? - Explicación del procesamiento de lenguaje natural - AWS. (s.f.). Amazon Web Services, Inc. Disponible en: https://aws.amazon.com/es/what-is/nlp/#:~:text=Las%20tecnologías%20de%20NLP%20permiten,los%20costos%20operativos%20al%20mínimo

[3] ¿Qué es el análisis de opiniones? - Explicación del análisis de opiniones – AWS, (sin fecha). Amazon Web Services, Inc. Disponible en: https://aws.amazon.com/es/what-is/sentiment-analysis/

[4] Banco BBVA - Productos financieros para personas y empresas | BBVA, (sin fecha). Disponible en: https://www.bbva.es/estaticos/mult/Ayudas_factores_acciones.pdf_tcm924-528182.pdf

[5] yfinance. (2024, mayo 19). PyPI. Disponible en: https://pypi.org/project/yfinance/

[6] Documentation - News API. (s.f.). News API – Search News and Blog Articles on the Web. Disponible en: https://newsapi.org/docs

[7] González, L. (2022, septiembre 7). *¿Qué es TensorFlow? ¿Cómo funciona?* Aprende IA. Disponible en: https://aprendeia.com/que-es-tensorflow-como-funciona/

[8] TensorFlow Core. (s.f.). TensorFlow. Disponible en: https://www.tensorflow.org/guide?hl=es

[9] Google Colab. (s.f.). Google Colab. Disponible en: https://colab.research.google.com

[10] Gu, S., Kelly, B., & Xiu, D. (2018). Empirical asset pricing via machine learning. Review of Financial Studies/the Review of Financial Studies. Disponible en: https://doi.org/10.1093/rfs/hhaa009

[11] Gu, S., Kelly, B., & Xiu, D. (2021). Autoencoder asset pricing models. Journal of Econometrics, 222(1), 429-450. Disponible en: https://www.sciencedirect.com/science/article/pii/S0304407620301998

[12] Mitra, A., Siddhant, M., & P, G. (2022). Credit Card Fraud Detection using Autoencoders. YMER Digital. Disponible en: https://doi.org/10.37896/ymer21.06/32

[13] Tan, C., & Eswaran, C. (2010). Reconstruction and recognition of face and digit images using autoencoders. Neural Computing and Applications, 19, 1069-1079. Disponible en: https://doi.org/10.1007/s00521-010-0378-4

[14] Fama, E. F., & French, K. R. (2008). Dissecting anomalies. The journal of finance, 63(4), 1653-1678. Disponible en: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2008.01371.x

[15] Shen, J., & Shafiq, M. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. Journal of Big Data, 7. Disponible en: https://doi.org/10.1186/s40537-020-00333-6

[16] Kalyani, J., Bharathi, P. H. N., & Jyothi, P. R. (2016, July 7). Stock trend prediction using news sentiment analysis. arXiv.org. Disponible en: https://arxiv.org/abs/1607.01958

[17] Bharathi, S., Geetha, A., & Sathiynarayanan, R. (2017). Sentiment Analysis of Twitter and RSS News Feeds and Its Impact on Stock Market Prediction. International Journal of Intelligent Engineering & Systems, 10(6). Disponible en: https://inass.org/2017/2017123108.pdf

ANEXO I: DECLARACIÓN DE USO DE HERRAMIENTAS DE INTELIGENCIA ARTIFICIAL GENERATIVA EN TRABAJOS DE FIN DE GRADO

ADVERTENCIA: Desde la Universidad consideramos que ChatGPT u otras herramientas similares son herramientas muy útiles en la vida académica, aunque su uso queda siempre bajo la responsabilidad del alumno, puesto que las respuestas que proporciona pueden no ser veraces. En este sentido, NO está permitido su uso en la elaboración del Trabajo fin de Grado para generar código porque estas herramientas no son fiables en esa tarea. Aunque el código funcione, no hay garantías de que metodológicamente sea correcto, y es altamente probable que no lo sea.

Por la presente, yo, Jaime de Clemente Fernández-Picazo, estudiante del Doble Grado en Ingeniería de Tecnologías de Telecomunicación y Análisis de Negocio/Business Analytics de la Universidad Pontificia Comillas al presentar mi Trabajo Fin de Grado titulado " Predicción de Precios Intradía de Mercados Financieros usando Noticias", declaro que he utilizado la herramienta de Inteligencia Artificial Generativa ChatGPT u otras similares de IAG de código sólo en el contexto de las actividades descritas a continuación:

1. **Brainstorming de ideas de investigación:** Utilizado para idear y esbozar posibles áreas de investigación.
2. **Crítico:** Para encontrar contra-argumentos a una tesis específica que pretendo defender.

3. **Referencias:** Usado conjuntamente con otras herramientas, como Science, para identificar referencias preliminares que luego he contrastado y validado.
4. **Metodólogo:** Para descubrir métodos aplicables a problemas específicos de investigación.
5. **Interpretador de código:** Para realizar análisis de datos preliminares.
6. **Estudios multidisciplinares:** Para comprender perspectivas de otras comunidades sobre temas de naturaleza multidisciplinar.
7. **Constructor de plantillas:** Para diseñar formatos específicos para secciones del trabajo.
8. **Corrector de estilo literario y de lenguaje:** Para mejorar la calidad lingüística y estilística del texto.
9. **Sintetizador y divulgador de libros complicados:** Para resumir y comprender literatura compleja.
10. **Generador de datos sintéticos de prueba:** Para la creación de conjuntos de datos ficticios.
11. **Generador de problemas de ejemplo:** Para ilustrar conceptos y técnicas.
12. **Revisor:** Para recibir sugerencias sobre cómo mejorar y perfeccionar el trabajo con diferentes niveles de exigencia.
13. **Generador de encuestas:** Para diseñar cuestionarios preliminares.
14. **Traductor:** Para traducir textos de un lenguaje a otro.

Afirmo que toda la información y contenido presentados en este trabajo son producto de mi investigación y esfuerzo individual, excepto donde se ha indicado lo contrario y se han dado los créditos correspondientes (he incluido las referencias adecuadas en el TFG y he explicitado para que se ha usado ChatGPT u otras herramientas similares). Soy consciente de las implicaciones académicas y éticas de presentar un trabajo no original y acepto las consecuencias de cualquier violación a esta declaración.

Fecha: 21/06/2024

Firma: Jaime de Clemente Fernández-Picazo