UNIVERSIDAD AUTONOMA DE MADRID

Escuela Politécnica Superior

20|21

Trabajo fin de grado

Aplicación de NLP y otras tecnologías para la mejora de predicción en entornos bursátiles

Javier A. Lougedo Lorente

Escuela Politécnica Superior
Universidad Autónoma de Madrid
C\Francisco Tomás y Valiente nº 11

Campus Internacional excelencia UAM CSIC+

www.uam.es

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería de Textos

TRABAJO FIN DE GRADO

Aplicación de NLP y otras tecnologías para la mejora de predicción en entornos bursátiles

Aplicación de modelos de Aprendizaje Automático al análisis bursátil

Autor: Javier A. Lougedo Lorente
Tutor: Alejandro Bellogín Kouki

junio 2021

Javier A. Lougedo Lorente
Aplicación de NLP y otras tecnologías para la mejora de predicción en entornos bursátiles

Javier A. Lougedo Lorente
C\ Francisco Tomás y Valiente Nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

A mi familia, novia y amigos que tantos cambios de humor me han soportado

La mayoría de las ideas fundamentales la ciencia son esencialmente sencillas y, por regla general pueden ser expresadas en un lenguaje comprensible para todos.

Albert Einstein

Prefacio

Este TFG nació a partir de una mezcla de ideas y conceptos que me plantearon mi cuñado y mi tutor, a fin de aplicar nuevas ideas y tecnologías a un sector tan importante a nivel económico como lo es la bolsa y el entorno bursátil.

Todo el código de este TFG, mayormente *Notebooks Jupyter* con distintas ideas y planteamientos, puede encontrarse accesible en la plataforma `tfg-lougedo.herokuapp.com`.

El propósito de este documento es defender el TFG que con tanto cariño he realizado a lo largo de estos meses, y que servirá como punto y final a mis estudios aquí en la UAM para dar paso a una, idealmente próspera, vida laboral, por la que estoy agradecido a todos mis profesores que tanto me han dado y enseñado.

Como creador y escritor de este documento, espero que te sientas cómodo durante su lectura, que disfrutes, y que idealmente, aprendas y saques algo en claro de él.

Javier A. Lougedo Lorente

Agradecimientos

En primer lugar, me gustaría agradecer a mi pareja el haber soportado todas mis subidas y bajadas a lo largo de este último año, todos mis cambios de ánimo, y por haber sido un apoyo fundamental a lo largo del desarrollo de este TFG. Gracias, Marina.

Agradecerle también a mi familia su incondicional apoyo en todo momento, por haber estado ahí siempre que me han hecho falta, y por su acometimiento a cualquier idea que quisiese escoger y llevar a cabo. Gracias mamá, y gracias, papá.

En tercer lugar, pero no por ello menos importante, agradecerle a mi cuñado, ya un hermano para mí, todas las oportunidades, ideas y apoyo constante que me ha ofrecido. Gracias, Benito.

Agradecer también a todos mis profesores, Idoia, Castells, Cantador, Aracil, Guillermo, Eloy y como no, Bellogín, por haberme facilitado todos los conocimientos que ahora tengo y haberme enseñado todo lo que sé sobre informática. Gracias, Alejandro, sobre todo a ti, por haber soportado también toda mi intensidad y haber leído todos mis correos, que prácticamente podían considerarse ensayos, y por tener la confianza que tuviste en mí, que en ocasiones ni yo mismo tenía.

Quería agradecer también, llegados a este punto, a la Escuela Politécnica Superior, que ha sido realmente como un hogar para mi estos últimos años, donde me he podido sentir cómodo, a gusto, y en resumidas cuentas, como en casa. Y en este punto, agradecer también a una persona muy especial pero que dudo que llegue a saber de esto y que ya no está en la universidad desde hace tiempo, pero gracias, Diego, por todos esos bocatas y esos mensajes de ánimo a la hora de comer antes de seguir unas horas más en la universidad. Sin ti, nada de esto habría sido posible, gracias por animarme día tras día y darme ganas de seguir siempre un poco más.

Y por último, quería agradecerle a la persona responsable de que tenga las ganas de aprender, enseñar y de que tenga la pasión por la tecnología que tengo: mi profesor, socio y amigo José Sebastián Trocolí.

Resumen

En el entorno bursátil siempre ha sido, es y será importante ser capaz de tratar de adivinar si el valor de unas determinadas acciones iba a subir, iba a bajar o iba a mantenerse. Ya fuese por mera intuición, por experiencia previa o por chivatazos y métodos ilegales, siempre ha existido esa necesidad de tratar de predecir que iba a ocurrir con el precio de las posiciones en bolsa, a fin de maximizar las probabilidades de obtener un beneficio.

Este proyecto gira entorno al acercamiento de nuevas tecnologías, ideas y métodos al entorno bursátil, a fin de maximizar el beneficio obtenido, de ser capaces de predecir con mayor detalle y exactitud qué va a suceder y tratar de implementar soluciones que nos permitan generar una mayor cantidad de beneficios, con una mayor seguridad y menor propensión al riesgo, tanto a los clientes como a las empresas involucradas.

De esta manera, pasaremos por los siguientes puntos: en primer lugar, empezaremos por una idea sencilla, basada en el estudio del precio de las acciones de una determinada empresa a lo largo del tiempo, donde podremos sacar métricas de riesgo, ganancias estimadas a priori, correlaciones con el precio de acciones de otras empresas y a partir de las cuales podremos entrenar modelos de predicción de series temporales para intuir cómo se comportarán en un futuro, una serie de ideas básicas y ya empleadas a modo de guía. Tras esto, trataremos de aplicar *NLP* para tratar de entrenar modelos más precisos, obteniendo información de Twitter y de noticias. Obtendremos información a partir de dos ideas: análisis de sentimiento de los tweets recientes relacionados con una determinada empresa, y posteriormente, trataremos de aplicar una idea más compleja, basada en el *embedding*, para tratar de ver qué tipo de tweets/noticias se asocian a una subida en bolsa y qué tipo a una bajada. Por último, compararemos los resultados obtenidos con la adición de NLP con los previamente extraídos, y compararemos con algunos de los modelos en la red.

Palabras clave

Entorno bursátil, series temporales, Procesamiento del Lenguaje Natural (NLP), Aprendizaje Automático, Sistemas de Recomendación

Abstract

In the stock exchange world, it has always been, is, and will be important to be able to guess whether the value of a certain stock was going to increase, decrease or hold. Whether it was by mere intuition, prior experience or other (maybe illegal) methods, there has always been this need to predict what would happen to the price of stock positions, in order to maximize the chances of making profit this way.

This project involves the application of new technologies, ideas and methods to the stock market environment, in order to maximize the profit obtained, not only by users but by companies, to be able to predict with greater detail and accuracy what is going to happen and to try to implement solutions and tools that allow us to generate more benefits, with greater liability and less risk.

This way, we will go through the following points/steps: first, we will start with a simple idea, based on the study of the price of the shares of a certain company over time, where we can obtain risk metrics, estimated earnings, correlations with the price of shares between companies, etc., we will be able to train time series prediction models to predict how they will behave in the future. From this point, we will try to improve the models by obtaining and using information from Twitter (or, actually, any other source we want to implement, such as news, or Facebook). We will use and process this information in two different ways: by analysing the sentiments on it and in a more complex way, by trying to apply embedding, to see which kind of tweets are related to a rise in the stock values and which kind are related to a drop. Finally, we will try to apply this new data to enhance the prediction of the previously implemented models.

Keywords

Trading, stock exchange, time series, Natural Language Processing, Machine Learning, Recommender Systems

Índice

Listas

Lista de ecuaciones

Lista de figuras

1

Introducción

Podríamos suponer que la idea de las acciones y participaciones en una empresa para compartir sus beneficios es una idea moderna. Sin embargo, el origen de esta idea se encuentra realmente en las civilizaciones más primitivas, y se desarrolla, concretamente, en la romana. Los negocios más pequeños combinarían sus fondos para permitirse barcos que cruzasen a lugares más lejanos donde vender, dándose la idea de colaboración entre empresas. En la edad media, cuando comenzó a tener lugar el comercio entre distintas civilizaciones distantes entre sí de una manera más global, fue necesario instaurar también un intercambio de moneda, para asegurar que los trueques e intercambios fuesen justos. Con el paso del tiempo, se dieron eventos que favorecieron la aparición de nuevos mercados y formas de entenderlos, pudiendo resumirse en este eje cronológico que he realizado anotando los eventos más relevantes hasta los mercados bursátiles tal y como los entendemos hoy en día.

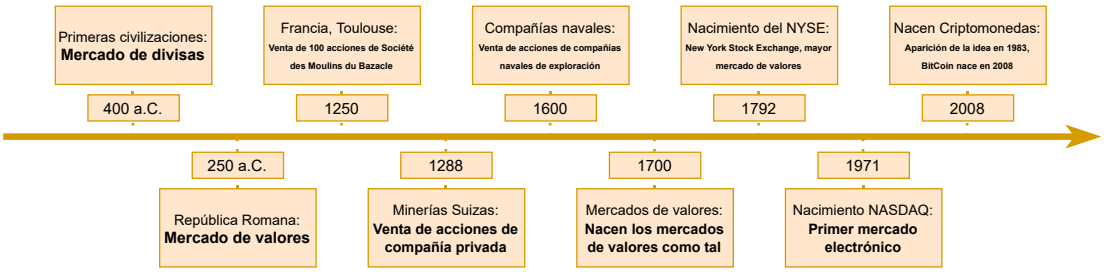


Figura 1.1: Evolución de los mercados de valores, en base a SoFiLearn [1] y Stocks [2].

El campo de los mercados de valores se ha ido modernizando cada día más y más, incorporando el uso de las nuevas tecnologías en su seno, y este TFG girará entorno al estudio de la aplicabilidad de distintas ideas, tecnologías y soluciones a este campo.

1.1. Motivación del proyecto

El mundo bursátil gira hoy día entorno a la idea del *trading* (negociación bursátil), que es la especulación sobre los instrumentos financieros a fin de obtener un beneficio a partir de los mismos [3]. Es decir, comprar acciones de determinadas empresas a un precio, y venderlas a un precio mayor.

Un *trader* (persona que se dedica al *trading*) utiliza generalmente una o varias estrategias en conjunto, que son:

1.– El análisis técnico de un activo, donde se estudian gráficas para tratar de predecir tendencias y comportamientos determinados relativos a su precio.

2.– El análisis fundamental, donde se estudia la información contable de la empresa para evaluar y predecir la tendencia de su precio (empleando generalmente noticias y rumores respecto a la empresa).

3.– El análisis macroeconómico, donde se consideran las variaciones que haya en la economía, así como su entorno económico.

4.– El análisis cuantitativo, donde se usa la estadística para predecir movimientos en los precios.

Debido a la creciente complejidad de este campo, del número de empresas y de la información disponible, así como la dificultad de manejar tales volúmenes de información, cada vez es más necesario un entorno de herramientas para facilitar el trabajo en este área y ofrezca mejores resultados.

Es por tanto que viene a la mente la aplicación de distintas estrategias, ideas y tecnologías del campo de la computación y de la IA a fin de calcular predicciones, métricas y, en resumen, facilitar el estudio del precio de los activos de una empresa, así como de la información disponible al respecto.

1.2. Objetivos

El objetivo principal de este trabajo es llevar a cabo el estudio de la aplicabilidad de distintas tecnologías del campo de la computación y de la Inteligencia Artificial al entorno bursátil, así como la preparación de un pequeño *tool-kit* para la predicción del precio de activos y su estudio. Idealmente, podrá ser utilizado tanto por individuos que busquen predecir el valor de un determinado activo, tanto por empresas que busquen predecir o anticipar el volumen de transacciones que se lleve a cabo.

1.3. Estructura del trabajo

El documento está dividido en 5 capítulos, los cuales se detallan a continuación:

1.– **Capítulo 1. Introducción**: Descripción actual de las implementaciones realizadas hasta la fecha en aplicación de Inteligencia Artificial en el entorno bursátil.

2.– **Capítulo 2. Estado del arte**: Introducción a la situación de distintas tecnologías y empresas actuales.

3.– **Capítulo 3. Diseño e Implementación**: Explicación de los módulos en los que se ha diseñado el proyecto partiendo de la base planteada en el Estado del Arte. Se justifican todas las decisiones de diseño e implementación tomadas, adjuntando los distintos diagramas explicativos del proceso.

4.– **Capítulo 4. Pruebas y resultados**: Se muestran y se evalúan los modelos e ideas implementadas, comparando las soluciones planteadas y evaluando su usabilidad y efectividad.

5.– **Capítulo 5. Conclusión y trabajo futuro**: Conclusiones obtenidas tras la realización del proyecto y análisis de viabilidad del proyecto, propuestas de modificación y ampliación para continuar la investigación en el área.

2

Estado del arte

En la presente sección se llevará a cabo el estudio del campo del *fintech* (explicado más adelante), qué tecnologías existen y se utilizan actualmente, y posteriormente se verá en qué estado se encuentran las diferentes técnicas, tecnologías e ideas aplicadas, así como su aplicabilidad. Se presentarán adicionalmente, para concluir, conceptos que puedan resultar relevantes para entender ideas más complejas explicadas más adelante, como puede ser el caso de las correlaciones.

2.1. Fintech

El origen del *fintech* es la contracción de las palabras inglesas *finance* y *technology*, que engloban básicamente los servicios de empresas del sector financiero que se ayudan de nuevas tecnologías para crear productos financieros innovadores [4].

Es el caso de *eToro*, RobinHood o Fidelity [5], plataformas orientadas al *trading* que se ayudan de estas nuevas tecnologías para facilitar intercambios en el entorno bursátil, así como el estudio de sus datos.

Actualmente, como ya se han mencionado, existen muchas plataformas orientadas al *trading*, y que ofrecen a sus usuarios la posibilidad de comprar y vender acciones. Estas buscan un punto diferenciador respecto al resto de compañías, con el que tratar de obtener una mayor cantidad de clientes y, a grandes rasgos, generar una mayor cantidad de beneficios.

Estas plataformas consisten en su mayoría de visores del precio a lo largo de los últimos tiempos de un determinado activo, así como herramientas adicionales que permiten la representación de distintas maneras de la información para el usuario. Adicionalmente, además del visor, disponen de diversos métodos para llevar a cabo la compra-venta de acciones, por las cuales obtendrán algún tipo de beneficio. Donde se trata de buscar el factor diferenciador es, principalmente, en el método de cobro de estos intereses (por volumen, por cantidad de transacciones, estáticos, dinámicos, etc) y por la funcionalidad ofrecida por su visor, así como la fiabilidad de la aplicación y su rendimiento.

Estas empresas pueden sacar beneficio de diversas formas, pero la más común es cobrar intereses

a cada usuario por cada transacción realizada (compra/venta), o un pequeño porcentaje de la misma. De esta manera, se benefician así las empresas en gran medida del volumen de transacciones que haya en un determinado día, compradas a través de ellos en su aplicación de *trading*.

Por otro lado, los usuarios se benefician de saber el precio final, y el volumen es completamente irrelevante para ellos (a no ser que tratemos de implementar un sistema de recomendación en el que el volumen pueda llegar a resultar relevante para un usuario y, concretamente, la recomendación realizada).

Cabe mencionar, llegado a este punto, que generalmente las empresas y aplicaciones de *trading* no realizan recomendaciones de ningún tipo en cuanto a la compra y venta de activos, puesto que la legislación en dicho ámbito suele ser bastante restrictiva y han de tener especial cuidado en no favorecer activos en particular recomendando su compra o su venta, si no tienen la potestad y los certificados para ello.

Actualmente, en lo relativo a predicción en bolsa, hay muy pocas plataformas que lo ofrezcan, en su totalidad de pago y con herramientas bastante crípticas y complicadas de emplear, que no siempre funcionan. Además, pese a que algunos de estos incorporan el análisis de sentimiento de una empresa como métrica visual adicional, ninguno, que yo haya sido capaz de encontrar, emplea esta información para afinar la predicción, sino que se limitan a mostrar la opinión de una empresa.

2.2. Series temporales

Una serie temporal o cronológica es una sucesión de datos medidos en determinados momentos, ordenados cronológicamente. Estas series pueden estar separadas de forma equidistante, seccionadas por espacios periódicos (un dato diario, cada minuto, cada segundo, etc), o por intervalos variables, como es el caso de las que manejaremos en este trabajo, puesto que no tenemos datos de los fines de semana o de determinados días a lo largo del año, o cuando manejemos más adelante las extraídas de los datos de Twitter, de nuevo, no tendremos datos equidistantes.

En esta sección se llevará a cabo la explicación de los distintos conceptos y tecnologías relativos a series temporales que serán aplicados para el estudio, análisis y predicción de las mismas.

2.2.1. Estudio y análisis de series temporales

En esta sección se analizan los distintos factores cuyo estudio puede resultar interesante más adelante y nos puede ofrecer información y un resumen visual de lo que está sucediendo actualmente con una determinada serie temporal. Estos distintos factores nos ayudarán más adelante a simplificar la información para el ojo humano, mucho menos preciso que nuestros ordenadores, para que un usuario pueda visualizar de manera mucho más cómoda esta información y estos pequeños detalles.

Media Móvil

Comenzaremos por la media móvil, que es, a grandes rasgos, la media de un determinado período de tiempo, que puede ir desde los dos últimos puntos de la serie temporal al número que deseemos de la misma, en función de la cantidad de puntos disponibles. De esta forma, la media móvil de los Z últimos puntos, por ejemplo, de una serie temporal en un determinado punto X, sería la media de los valores entre los puntos X y $X - Z$, siendo ese Z el número de días del cuál queramos obtener la media temporal. Esto implica también que no tendremos un valor de esos Z primeros puntos, pues necesitamos puntos que no han aparecido aún. La fórmula para calcular la media móvil sería la siguiente.

$$MovingAverage(S, N, P) = \frac{S_{P-N} + S_{P-N+1} + ... + S_P}{N} \tag{2.1}$$

Las medias móviles son así una serie temporal simplificada y suavizada en la cual cada uno de sus elementos es el promedio de un subconjunto de los datos del conjunto original. Su función es, principalmente, permitir la visualización de los datos de una manera más simplificada, analizar y observar tendencias a largo plazo y ofrecer un resumen del comportamiento de la red. Más adelante en las pruebas realizadas, concretamente en la figura 4.1 se podrá observar un buen ejemplo de la misma.

Porcentaje de cambio (retorno diario)

Esta es de las métricas más importantes y claras que podemos extraer de una serie temporal en el entorno bursátil, que llamaremos retorno diario (daily return), en lugar de porcentaje de cambio. Es, además, una métrica tremendamente sencilla, que permite estudiar las ganancias que produce un activo respecto al día anterior. Su fórmula es bastante sencilla, pues se calcula dividiendo el valor de un determinado punto por el valor que tuvo el punto anterior, tal y como se muestra a continuación.

$$ChangePercentage(S, P) = \frac{S_P}{S_{P-1}} \tag{2.2}$$

Este nos sirve para ver las ganancias que obtendrá un usuario a partir de un activo de un día para otro, y valorar así si una transacción será beneficiosa o no. También puede llegar a calcularse a partir de datos anteriores, para ver la ganancia o pérdida respecto a cualquier punto previo de la serie. Esto nos dará además una serie temporal con una forma más serrada, en la que podemos ver las ganancias cada día, porcentualmente.

Riesgo y beneficio simplificados

A partir de la serie temporal del retorno diario, podemos extraer dos parámetros que simplifican bastante la dinámica de las series temporales y que dejan de lado bastante información, pero en ocasiones esto puede resultar útil. Son el riesgo y el beneficio estimados a priori, que se calculan

como la desviación típica de la serie temporal y como su media, respectivamente.

$$PioriBenefit(S) = \frac{\sum S}{len(S)} \tag{2.3a}$$

$$Risk(S) = \frac{\sum |S - PrioriBenefit(S)|}{len(S)} \tag{2.3b}$$

De nuevo, son valores muy simplificados, pero nos sirven así principalmente para comparar de manera muy simplificada dos series temporales.

Correlaciones

La única métrica (o más bien métricas) que podría llegar a resultar más relevante que la del retorno diario es la de las correlaciones. Si antes, en la sección de riesgo y beneficio, hablábamos de una manera para comparar de manera simplificada dos series temporales (o más), ahora hablamos de una manera no solo de comparar series temporales, sino de ver como están relacionadas entre sí, en el caso de estarlo, y como se relacionan entre sí.

Es decir, yéndonos a una definición más formal en el campo de la probabilidad y la estadística: **nos indica la fuerza y la dirección de una relación lineal y proporcionalidad entre dos variables estadísticas**. Se considera que dos variables están **correlacionadas** cuando los valores de una de ellas varían sistemáticamente con respecto a los valores homónimos de la otra: si tenemos dos variables (A y B) existe correlación entre ellas si al disminuir los valores de A lo hacen también los de B y viceversa; pero también puede ocurrir lo contrario (correlación inversa) si una bajada del valor de B implica una subida de A, o viceversa. La correlación entre dos variables no implica, por sí misma, ninguna relación de causalidad.

Es decir, en el caso que nos interesa, que es el entorno bursátil, nos servirá para mostrarnos relaciones como si cuando el valor de *CocaCola* sube, baja el de *Pepsi*, si cuando el valor de *Pfizer* sube lo hace también el de Moderna o *AstraZeneca*, o si cuando el valor de *Apple* sube, el de *Microsoft* se mantiene. Todo ese tipo de relaciones “condicionales” en los precios de un activo son detectables por correlación, y esta tiene varias formas y fórmulas con las que medirse.

La fórmula general de correlación que se emplea en estos casos viene dada por la siguiente fórmula, donde *x* e *y* representan distintas series temporales, con un determinado desfase *k*.

$$r_k = \frac{\sum_{i-1}^{N-k} (x_i - \bar{x}) \cdot (y_{i+k} - \bar{y})}{\sqrt{\sum_{i=1}^{N-k} (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=k+1}^{N} (y_i - \bar{y})^2}} \tag{2.4}$$

Adicionalmente, se emplea además la correlación de *Spearman*, definida la siguiente fórmula, empleada principalmente en la correlación de volúmenes de Tweets, donde ρ define el valor de la correla-

ción de *Spearman*, d_i es la diferencia entre ambas series y *n* es el número de muestras.

$$r_s = 1 - \frac{6 \sum d_i^2}{n\left(n^2 - 1\right)} \tag{2.5}$$

Más adelante, haremos un estudio detallado de correlaciones y veremos más en detalle lo que esto implica, y que tipo de información podría llegar a aprender y extraer uno de nuestros modelos de *Machine Learning*.

2.2.2. Predicción en series temporales

El poder predecir que va a suceder a continuación es algo que ha llamado la atención de nuestra especie desde los inicios de la civilización. Ya sea vía profetización, adivinación o lectura de las estrellas, siempre se ha querido predecir eventos futuros. Todas estas predicciones, en su mayoría, siempre giran entorno a algún tipo de información o dato previo, es decir, están sustentadas en algún tipo de pilar, ya sea más o menos sólido, pero están basadas en algo, y no son triviales.

Hoy día con la cantidad de información presente en el mundo, predecir sucesos no es ya una idea irrealizable, sino que es posible llevarla a cabo, con mejores o peores resultados. De esta forma, solo es necesaria información y ciertas asunciones, que podremos realizar nosotros o nuestro modelo, en base a una serie de datos, tendencias y dinámicas.

Este problema de predicción es así uno de los problemas más habituales e interesantes en series temporales, pues resulta interesante predecir lo que va a ocurrir más adelante, ya sea para sacar algún beneficio por ello, anticipándose a dichos sucesos, o por el mero hecho de tener más datos [6].

Para ello, en el campo de las series temporales [7], existen una serie de técnicas y de modelos para llevar esto a cabo, girando principalmente entorno a la idea de que una serie temporal tiene unos parámetros inherentes, tendencias y comportamientos parametrizables, que podemos tratar de modelar.

Existen así muchos modelos para ello, observables en el estudio de Davide Burba [8] sobre el Estado del Arte en predicción de series temporales, entre los que se encuentran principalmente el modelo de red neuronal en que se basará principalmente la predicción de series temporales en este proyecto: las *LSTM*.

LSTM

A priori, resolver un problema como lo es el de predecir que va a ocurrir con una serie temporal en los siguientes puntos cuyos datos desconocemos puede parecer una tarea excesivamente compleja. Sin embargo, con un mínimo de conocimientos del funcionamiento en general del Machine Learning y

de la Inteligencia Artificial, podemos llegar a una idea que nos facilitará la comprensión y el trabajo futuro: los datos de los puntos previos son nuestros parámetros, y el siguiente punto que queremos tratar de averiguar, nuestra clase a calcular; que posteriormente formará también parte de los argumentos.

Las redes neuronales *LSTM* (siglas que corresponden a *Long Short-Term Memory*) son unas redes neuronales artificiales con arquitectura recurrente (*RNN*), que se diferencian de las redes neuronales prealimentadas (aquellas que no tienen ningún ciclo y "fluyen" únicamente hacia delante) en que tiene conexiones que **retroalimentan** la red, es decir, que forman ciclos y cuya información fluye también en sentido contrario [9]. Son utilizadas principalmente en aquellos problemas que dependen de lo que haya habido u ocurrido previamente, es decir, por ejemplo, ya no solo series temporales, que es lo que nos importa en este proyecto, sino también procesamiento de vídeo, sonido, etc., siendo utilizadas por ejemplo en reconocimiento de la escritura manual, reconocimiento de voz, y de imagen.

De esta forma, representan una arquitectura ideal para nuestro problema, pues en primer lugar, están orientadas principalmente al tratamiento y predicción en series temporales en base a lo que ha ocurrido con la serie hasta el momento; tienen una pequeña memoria que nos será muy útil, almacenando lo que ha sucedido hasta el momento; y, por último, nos permiten resolver este problema complejo de una manera bastante sencilla. Aunque si bien el problema sería resoluble empleando redes neuronales normales, o incluso algún otro tipo de modelos, el empleo de *LSTM* nos otorgará una mayor precisión y mejores resultados.

2.3. Procesamiento del Lenguaje Natural

Tras haber visto lo relativo a series temporales, llega la hora de ver lo relativo al Procesamiento del Lenguaje Natural (NLP), mediante lo cual trataremos de afinar más nuestras predicciones y obtener más información sobre el campo que vamos a estudiar.

El NLP es un campo de la computación, que enlaza IA y la lingüística, que se ocupa de las interacciones entre las computadoras y el lenguaje humano, en cualquiera de sus formas y que busca que los ordenadores entiendan el contenido de documentos escritos, a grandes rasgos.

De esta forma, para el análisis bursátil, el objetivo sería, a partir de distintos documentos y fuentes de información, extraer información que nos permita afinar las predicciones realizadas, en colaboración con los datos previamente adquiridos o por sí mismos, para tratar de conocer, modelar y estudiar de manera más precisa lo que ocurrirá con el precio de un determinado activo, o con su volumen de transacciones; para lo cual emplearemos tecnologías como el análisis de sentimiento o embedding.

Se recomienda la lectura del artículo de Cristian Rus [10] sobre GPT-3, parte de la inspiración para llevar a cabo este proyecto relacionado con el NLP.

2.3.1. Preprocesado

Antes de trabajar con cualquier conjunto de datos en NLP, es importante preprocesarlos para reducir al máximo su complejidad, su ambigüedad y facilitar al máximo posible la comprensión a nuestro sistema de la información que va a tratar. Para ello, resulta de vital importancia ser consciente del contexto en el que estamos, que variará diversos puntos clave e interpretaciones de este preprocesado, y nos permitirá así simplificar los datos que nuestro modelo manejará.

Este preprocesado, generalmente, pasa por los siguientes puntos, que más adelante en la sección 3.3.2 serán desarrollados con más detalle en el contexto:

1.– Paso a minusculas.

2.– Reemplazo de conceptos ambiguos.

3.– Eliminación de stopwords.

4.– Eliminación (o sustitución) de palabras que queramos reducir en importancia.

5.– Lematizado (conversión a su base, de Great a Good) y stemmizado (de arqueologíco a arqueolog).

Adicionalmente, se podrán añadir más pasos. De esta manera conseguimos simplificar los datos y las tareas a nuestro modelo, dándole a entender que Antonio es lo mismo que ANTONIO o antonio, que arqueológico y arqueológica son prácticamente la misma palabra (de hecho la interpretará así como la misma), o, en nuestro caso, en un contexto como Twitter, que holaaa es lo mismo que hola.

2.3.2. Vectorización

El único paso que resulta más crucial incluso que el preprocesado del texto a analizar, del cual todavía se podría llegar incluso a prescindir al haber un enorme volumen de términos, es la vectorización, que es de vital importancia para poder trabajar con los datos. Como aprendemos desde bien pronto, los ordenadores no funcionan con números, sino con números. Concretamente, a muy bajo nivel, con unos y ceros. Lamentablemente, no son capaces de manejar palabras con la misma soltura que una persona. Es por tanto que debemos buscar un formato que sea comprensible por el ordenador, pero que a la vez no haga perder importancia a nuestros datos, ni introduzca sesgos y/o ruido innecesario en nuestros datos.

Una primera idea o aproximación a dicho problema podría ser asignar un número a cada palabra del vocabulario del que dispongamos. Es una solución válida a primera vista, con la que sencillamente codificamos los datos, y no estamos perdiendo información en el proceso, ni modificando el volumen de los datos. Sin embargo, esta solución tiene un serio problema, y es que nuestro modelo rápidamente creará relaciones entre esos números. Si *patata* se codifica como **16**, *aguacate* como **32**, *primo* con **40** y *parque* como *8*, nuestro modelo llegará a la conclusión de que un *aguacate* está más próximo de *primo* que de una *patata*, que si juntamos un *aguacate* con una *patata* obtenemos un *primo*, o que una *patata* son dos *parques*, y a su vez un *aguacate* son dos *patatas*. Como cabe intuir llegados a este

punto, es una codificación simple, pero que introduce relaciones entre los datos que inicialmente no existen, y que pueden dar lugar a problemas, principalmente relacionados con las distancias entre dos términos del vocabulario.

Es entonces cuando, entendiendo que la distancia, al menos inicialmente, entre todos los términos debería de ser la misma, surge una idea relativamente sencilla, aunque dimensionalmente compleja: asignarle a cada término del vocabulario un vector equidistante al resto de vectores. Esto, que inicialmente puede parecer extremadamente críptico, es una técnica ampliamente empleada que responde al nombre de *one hot encoding*. En resumidas cuentas, a cada término del vocabulario le corresponderá un vector N dimensional, siendo N el número de palabras del vocabulario. Así, cada uno de esos vectores estará compuesto por todo ceros, menos un 1 en la posición que corresponda a dicha palabra. Esto, pese a resolver a la perfección el problema anteriormente mencionado, tiene dos pequeños problemas: en primer lugar, la dimensionalidad y complejidad de nuestros datos ha aumentado en gran medida; y en segundo lugar, que ahora todas las palabras son equidistantes, cuando realmente tampoco es así. La distancia “conceptual” o “semántica” entre una patata y una zanahoria es mucho menor, por ejemplo, que la distancia entre una zanahoria y un avión. Dos términos están muy relacionados, y un tercero no tiene nada que ver.

Este último es un problema frecuente en el campo del Natural Languaje Processing, consistente en tratar de reducir la dimensionalidad de las palabras, para tratar de captar además mejor con esto su esencia, no solo reduciendo la longitud del vector necesaria para definir cada vector, sino también captando mejor sus relaciones con otras palabras. En este campo hay mucho trabajo y actualmente Word2Vec es uno de los módulos más conocidos del campo, tal y como lo es SKLearn en el campo de la IA. Sin embargo, es un punto que en este proyecto no será trabajado, empleando únicamente la funcionalidad ofrecida por Word2Vec.

De esta forma, para traducir los datos a un formato que nuestro modelo pueda manejar, utilizaremos un vectorizador TF-IDF, que generará, en nuestro caso, un modelo para transformar los textos a una lista de vectores, tokenizándolos (dividiéndolos en palabras o términos) primero, y posteriormente, escogiendo los 500.000 términos más frecuentes y relevantes, transformando finalmente todos los datos a este nuevo formato de vectores. Con estos datos ya traducidos (cosa que deberemos hacer también con los datos reales que queramos clasificar) ya podemos, por fín, comenzar a entrenar nuestros modelos.

2.3.3. Análisis del sentimiento

El análisis de sentimiento (también conocido como *opinion mining*) es una técnica utilizada para determinar si un determinado dato (texto) ofrece una opinión positiva, neutral o negativa (aunque en ocasiones también se puede hacer un análisis más detallado y concreto por ponderaciones en determinados sentimientos, como pueden ser ira, alegría, tristeza, sorpresa, etc). Generalmente se emplea en

datos textuales a fin de ayudar a negocios y compañías a monitorizar el *feedback* que dan sus clientes, a fin de reconocer si transmiten sensaciones positivas, negativas o neutrales, y tratar de comprender así mejor las necesidades de sus clientes, siendo así muy utilizados para evaluar los comentarios de un producto en Amazon o de una app en Google Play o el App Store de iOS.

Así, el análisis de sentimiento es el proceso de detectar sentimientos/sensaciones positivos o negativos en un determinado texto. Este puede ser de distintos tipos y complejidades, que van desde una clasificación bipolar entre positivo o negativo, una similar pero granulada de muy positivo a muy negativo (pasando por positivo, neutral y negativo), clasificación emocional (triste, enfadado, alegre, eufórico, asustado, etc.) y otros tipos de análisis más variados y variopintos. En el artículo de MonkeryLearn [11] se realiza un estudio más detallado.

La idea es, por tanto, en primer lugar obtener un analizador de sentimientos, ya sea creándolo, como se llevará a cabo en este proyecto en un determinado punto, o empleando uno existente (como también se realiza más adelante).

La tecnología disponible en este campo actualmente está todavía tomando forma y en desarrollo, y está teniendo un boom en los últimos tiempos, con plataformas y APIs orientadas exclusivamente al análisis de texto, con el propio Google investigando en este campo y desarrollando una API que cada día va tomando más forma. Como ejemplos de ello, tenemos la API de Google Cloud Natural Language [1], la de DeepAI, empleada en sus cursos online de Coursera [2] y, por último, el artículo de RapidAPI [12] donde se citan diversas APIs adicionales. El uso de cualquiera de estas fue descartado, mayormente porque la gran mayoría eran de pago y no están dirigidas a un uso tan masivo para usuarios individuales.

En ese sentido, nos ayudaremos en este proyecto de grandes librerías como son *SKLearn*, *Keras* y *NLTK* para desarrollar un analizador propio, idealmente más concreto y adaptado a nuestro problema y contexto; y posteriormente emplearemos clasificadores y analizadores ya existentes, modificándolos ligeramente de ser posible para afinar más su funcionamiento a nuestro problema.

2.3.4. Embedding

Embedding se traduce al español como “incrustar”, información que de primeras no nos aporta nada. Un embedding, en el contexto actual, es un espacio vectorial en el que se puede traducir información de un espacio vectorial superior. Concretamente, en el caso que nos acata relativo al NLP, donde inicialmente cada palabra se puede asociar a un vector N dimensional (donde N es el tamaño de vocabulario) con todo ceros menos un uno en la posición de la palabra que contiene, consiste en llevar a cabo una reducción de la dimensionalidad que trae consigo una serie de grandes ventajas:

[1] https://cloud.google.com/natural-language

[2] https://deepai.org/machine-learning-model/sentiment-analysis

1.– Una reducción de la dimensionalidad que nos permite simplificar los modelos empleados en gran medida y reducir los volúmenes de entrada empleados.

2.– Que palabras que inicialmente eran equidistantes se aproximen o separen entre sí (es decir, inicialmente rey y reina estaban a la misma distancia que rey y portaaviones, y tras esta reducción de dimensionalidad, conceptos que en el vocabulario sean similares, se acercan entre sí, estando rey y reina muy próximos pero rey y portaaviones a una distancia mayor).

3.– Que palabras se asocien entre sí creando nubes de conceptos e ideas, en función del vocabulario y del problema.

De esta forma, la idea del embedding es, a grandes rasgos, pasar del enormísimo volumen de información que implican las matrices dispersas empleadas en NLP, a algo de un volumen menor, que además, aporte beneficios en el proceso. En un clasificador de sentimiento, por ejemplo, aplicando embedding, podríamos aproximar entre sí palabras que implicasen una emoción negativa o una emoción positiva, mientras que si lo que buscamos es crear un clasificador de noticias en temáticas, podríamos hacer nubes de palabras o conceptos relacionados con cada una de esas temáticas, aproximándolas entre sí. Este es un proceso complejo y para nada trivial, que requiere de una gran carga computacional y que maneja, de primeras, grandes volúmenes de datos. Es por ello que existen modelos preentrenados que traducen nuestras palabras o tokens a vectores ya embebidos y con una dimensión mucho menor que la que podría tener un *one-hot encoder* [3]., donde conceptos similares entre sí, como pueden ser el campo semántico de las verduras, o el de los animales, están próximos entre sí, mientras que conceptos que son más dispares, como pueden ser helicóptero y hormiga, se encuentran separados.

Así, la idea principal sería en este campo entrenar un sistema de embedding cuya función fuese predecir subidas o bajadas de bolsa en base a los tweets disponibles de una determinada empresa, es decir, ver que tipo de tweets se asocian a subidas o bajadas del valor de una acción, o que tipo se asocian a una variación del volumen de transacciones, con lo que poder, en base a Twitter, predecir que va a suceder, idealmente, con el precio de un activo y con su volumen de transacciones.

2.4. Sistemas de Recomendación

Los sistemas de recomendación son herramientas que hoy en día están extremadamente inmersas en nuestra tecnología. El ejemplo más claro de ello es Spotify; nos basta introducir una única canción para que el resto se vayan reproduciendo en base a nuestro gusto, ya conocido por Spotify por las canciones que vamos escuchando a lo largo del tiempo y el perfil de gustos que almacena de cada uno de sus usuarios, en base a las últimas tendencias y en base a esa primera canción que hemos querido escuchar; para que nos vaya poniendo tras esta canción muchas que nos resultan adecuadas, y que pasado cierto momento, ni siquiera somos conscientes de escuchar y de haber elegido.

[3] La estrategia que implementa es crear una columna para cada valor distinto que exista en la característica que estamos codificando y, para cada registro, marcar con un 1 la columna a la que pertenezca dicho registro y dejar las demás con 0 [13]

No solo es el caso de Spotify, sino también de YouTube, o de prácticamente la totalidad de publicidad que llega a nosotros. Hoy en día, nosotros, los usuarios, nos encontramos con recomendaciones de ítems de cualquier tipo, desde productos de Amazon a cualquier tipo de contenido en Internet.

Así pues, los sistemas de recomendación son herramientas que ayudan a los usuarios a valorar opciones o elementos de interés para personalizar su experiencia. Son así una herramienta que establece criterios y valoraciones de los usuarios para predecir que clase de elementos pueden tener utilidad o gustarle al mismo que aún no haya probado o no conozca, en base a datos proporcionados de forma directa o no por el usuario, analizando su historial para transformar todos los conocimientos en recomendaciones.

Estos sistemas de recomendación son cada día más precisos, por dos motivos clave: hay un elevado interés económico y comercial en ello; y cada día vivimos en una sociedad mas informatizada, tecnológica y en la que se nos tiene a nosotros, los usuarios, cada vez mejor modelados, y se dispone de más información de cada uno de nosotros.

Así, dado que al fin y al cabo queremos realizar predicciones de lo que va a suceder con un determinado activo para realizar recomendaciones sobre si realizar una inversión o no, resulta relevante fijarse en el entorno de los Sistemas de Recomendación, para tratar de aplicar algunas de las ideas de este campo. Es cierto que entornos como Spotify, YouTube o Amazon tienen cierta subjetividad y sus ítems no cumplen una función meramente práctica, sino que también tienen un componente estético y que, en resumidas cuentas, le puede gustar o no al usuario; mientras que los ítems que aquí valoramos, que son las participaciones de una empresa, son algo en su totalidad objetivo, a excepción de muy concretos casos, y que se reduce a que una inversión es buena o no. Sin embargo, no quita que podamos tratar de aplicar ideas de este campo, y hacer esto en base a algo propio del usuario, ya sea su personalidad, su forma de escribir tweets, sus últimas compras o las compras que se están realizando en el momento o cualquier tipo de idea que pueda llegar a ser útil.

2.5. Librerías externas

A lo largo de la realización de este proyecto, se van a emplear distintas librerías Python para llevar a cabo todo el proceso de análisis, predicción, estudio del sentimiento, etiquetado, muestreo de los resultados y demás funciones. Para ello se utilizan las librerías listadas a continuación, clasificadas en función de su uso en cuatro grandes grupos. Hay algunas adicionales, concretamente entorno al análisis de sentimiento, que fueron descartadas y no son mencionadas por evitar malgastar espacio. Estas son librerías cuyo uso era complejo o no tan completo como el de *afinn* y *nltk*.

2.5.1. Muestreo y manejo de datos

Para el muestreo de datos se han empleado las siguientes librerías:

- ***pandas*** [14], a fin de manejar los *dataset* y los diversos datos de manera cómoda, así como por su vinculación con otras librerías, comentadas más adelante.
- ***numpy*** [15], para el manejo de *arrays* en aquellos momentos que pandas no daba a basto, de manera simple.
- ***csv*** [16], para el manejo de ficheros csv y su exportado.
- ***matplotlib*** [17], a fin de realizar con ella gráficas sencillas, limpias y directas.
- ***seaborn*** [18], una librería que permite realizar gráficas mucho más complejas y elaboradas, con modelos ya preparados que facilitan su presentación y edición, vinculada con *matplotlib* y *pandas*.

2.5.2. Series temporales y predicción

En el caso de las series temporales, en predicción y análisis de las mismas se han empleado las siguientes librerías:

- ***yfinance*** [19], para la obtención de datos financieros de los activos. Es una librería de Yahoo Finance que tiene vinculación con Pandas, con lo cual no es necesaria puesto que podemos utilizar un reader especial de Pandas.
- ***keras*** [20], para el entrenamiento de los predictores de series temporales que se han empleado.
- ***sklearn*** [21], para el escalado de los datos empleados.

2.5.3. Procesado del Lenguaje Natural

En el caso del Natural Language Processing, el uso de librerías ha sido más amplio, pues se ha realizado desde el entrenamiento de un analizador de sentimiento, lo que conlleva plataformas como SKLearn o Keras; al análisis y preprocesado de texto. Las librerías y herramientas utilizadas son:

- ***keras*** [20], para el entrenamiento del modelo analizador de sentimientos, así como del tokenizado y embebido.
- ***sklearn*** [21], para la creación de modelos sencillos en análisis de sentimiento.
- ***nltk*** [22], para el preprocesado del texto, limpieza de *stopwords*, lematizado y análisis de sentimiento.
- ***afinn*** [23], para el etiquetado de sentimiento pre-entrenado.
- ***wordcloud*** [24], para la creación de nubes de palabras visuales.

2.5.4. Otras

Otras librerías para usos varios, entre las que se encuentran:

- ***scipy*** [25], para calcular correlaciones eventualmente.
- ***time*** [26], para medidas de tiempos.
- ***datetime*** [27], para traducciones de una unidad temporal a otra.

3

Diseño e implementación

En este capítulo se abordará lo relativo al diseño del proyecto y su implementación, a fin de reducir el volumen de estos dos capítulos, viendo su estructura, ciclo de vida y requisitos, así como todo lo relativo a la implementación.

3.1. Diseño

En esta sección se detallarán los módulos en los que se organiza el proyecto, las librerías externas y su influencia y relación, así como el proceso realizado para analizar y preprocesar los datos, emplearlos en aprendizaje, evaluación y posterior visualización.

3.1.1. Estructura general

En la figura 3.1 se puede observar una estructura del proyecto que se ha llevado a cabo.

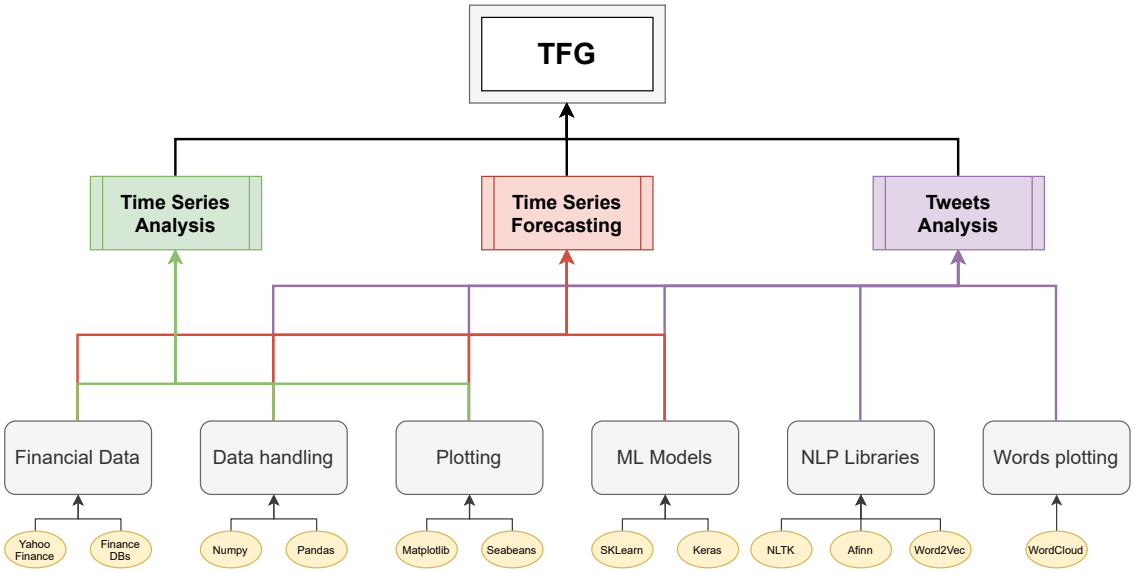


Figura 3.1: Estructura general del proyecto

En el, se observan tres secciones muy definidas, cada una de ellas relacionada con uno (o varios) de los Notebooks en los que se han realizado los distintos experimentos y donde se podrán observar ejemplos más detallados y desarrollados:

1.– Análisis de Series Temporales, que corresponde al Notebook *time_series_analysis.ipynb* del código entregado y disponible en la web comentada más adelante. En él, se lleva un estudio detallado de las series temporales, de los parámetros que se pueden extraer de ellas, de como realizar un estudio de las mismas y, por último, la extracción, obtención y muestreado de los mismos, con distintos ejemplos. Por último, se lleva a cabo el estudio de las correlaciones entre distintos activos en función de un campo.

2.– Predicción en Series Temporales, que corresponde al Notebook *time_series_prediction.ipynb* del código entregado y la web. En él, a raíz de lo anteriormente estudiado, se lleva a cabo la implementación de modelos para predecir tanto el valor de un activo como su volumen.

3.– Por último, en la sección de análisis de tweets, se lleva a cabo una serie de tareas más elaborada, toda relacionada con este campo. Corresponde a los Notebooks restantes entregados, y en ellos se lleva a cabo la implementación de un analizador de sentimientos, el análisis y empleo de otros analizadores de sentimiento, el etiquetado y procesado de los datos disponibles y su posterior estudio de correlación con los precios y volúmenes que se manejan en bolsa.

Los tres módulos superiores funcionan de forma independiente entre sí, con la colaboración de alguna rama de estudio adicional, y se centran cada uno en un area partícular. Sin embargo, se relacionan en un proyecto realista a partir de este de la forma que se observa en la figura 3.2.

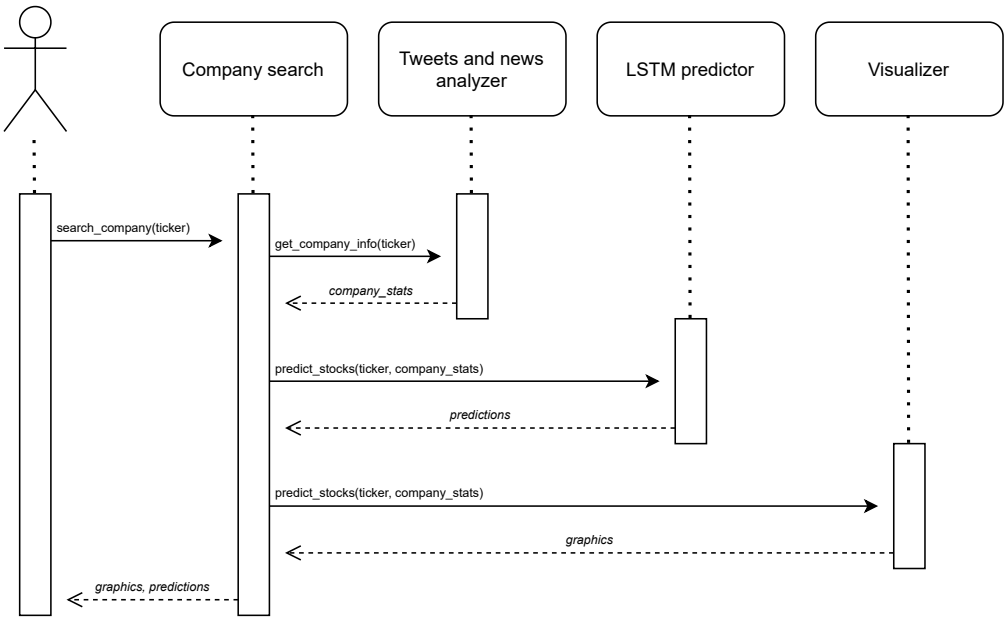


Figura 3.2: Diagrama de secuencia del proyecto

Cada uno de estos módulos contaría con un diagrama particular, que desarrollaría en el diagrama de secuencia mostrado lo que se está llevando a cabo paso a paso internamente, en detalle, que se han obviado a fin de no desaprovechar un excesivo espacio de este documento.

3.1.2. Ciclo de vida

Para el desarrollo de este proyecto se ha seguido un Ciclo de Vida Lineal Flexible (donde se podía retornar en casos excepcionales a puntos pasados para corregir cualquier tipo de error), debido a la simplicidad del producto, a la ausencia de necesidad de sincronizarse con otras personas y de dependencias, y a que los pasos finales dependían del correcto y sólido funcionamiento de los primeros. A partir del desarrollo, sin embargo, se sigue un ciclo de vida iterativo, probando distintas versiones del producto y tratando de mejorarlo, pues muchas cosas no son planificables sino que se descubren por prueba y error en los periodos de feedback, desarrollo y testing.

Así, el ciclo de vida del proyecto se podría ver representado por la figura 3.3

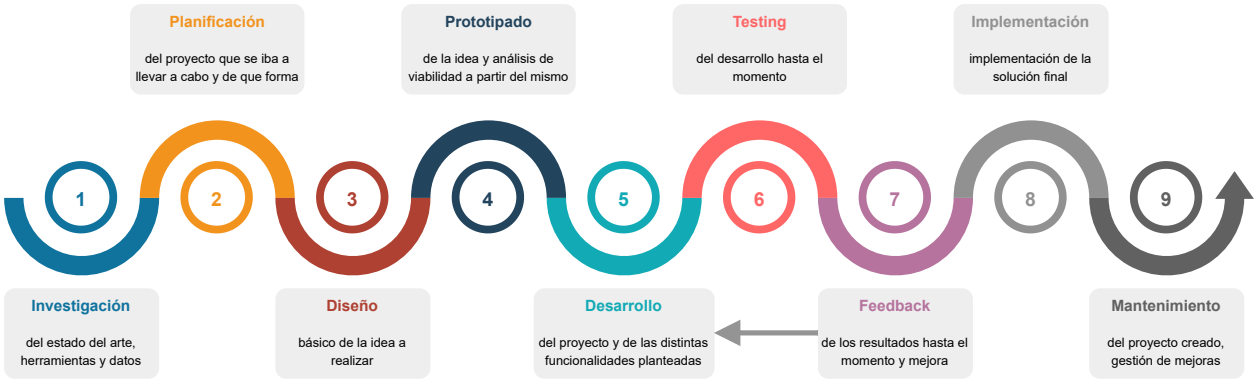


Figura 3.3: Diagrama descriptivo del ciclo de vida de este proyecto en particular.

Como se ha comentado, en este ciclo de vida, pese a ser lineal, se contempla la posibilidad de volver, en caso de ser necesario, así como de que en el periodo de desarrollo y testing habrá diversas iteraciones, tras recibir un pequeño feedback donde se analizasen los resultados obtenidos.

3.2. Requisitos

En esta sección se muestran los diferentes requisitos de este proyecto de investigación.

3.2.1. Requisitos funcionales

General:

RF-1.– El lenguaje será Python.

RF-2.– Se procurará almacenar los modelos con *pickle* para retomar el desarrollo y pruebas, en medida de lo posible, sin reiniciar el entremaiento.

RF-3.– En caso de haber algún error, deberá ser claro que lo ha habido, así como la posible resolución del mismo.

Procesado:

RF-4.– El sistema debe permitir distintas dimensiones de entrada de datos (series temporales de 10 años o de 10 meses).

RF-5.– El sistema no debe procesar tweets directamente. Los recibirá a través de un dataset previamente obtenido.

Predicción:

RF-6.– Se priorizará, en medida de lo posible, la predicción batch (todos los puntos a la vez) y no punto a punto.

RF-7.– Se priorizará la predicción en base a datos bursátiles reales antes que en base a noticias o tweets.

RF-8.– Se ofrecerán e investigarán distintos modelos al usuario, aclarando los mejores.

RF-9.– Se permitirá, de forma cómoda, modificar los parámetros de mayor importancia en el entrenamiento.

RF-10.– El sistema debe permitir variar los activos a predecir.

RF-11.– El sistema deberá predecir los $N/10$ siguientes valores de un activo/volumen a partir de los N anteriores.

RF-12.– Cuando el sistema funcione con datos adicionales a los bursátiles, lo hará considerando solo aquellos puntos con datos de ambas fuentes.

Visualización:

RF-13.– El sistema permitirá visualizar los resultados.

RF-14.– Tras cada ejecución, se facilitará al usuario un PDF con la gráfica resultante.

3.2.2. Requisitos no funcionales

RNF-1.– Todos los ficheros Notebook que faciliten las distintas pruebas y ejecuciones serán accesibles vía web, sin necesidad de instalar nada localmente.

RNF-2.– La aplicación ha de ser, así, accesible desde un navegador web.

RNF-3.– Las gráficas mostradas serán claras y no ambiguas.

RNF-4.– El tiempo de predicción básica del valor de un activo o su volumen no debe ser mayor a 3 minutos.

RNF-5.– El tiempo de predicción con análisis del lenguaje de un activo/volumen no debe ser mayor a 15 minutos.

3.3. Implementación

En este apartado se comenta el flujo del proyecto desde la obtención de los datos a la extracción de las gráficas y predicciones finales.

3.3.1. Búsqueda y obtención de los datos

A la hora de realizar el proyecto, debemos de considerar que se obtienen datos de muy diversas fuentes a fin de obtener un resultado final más preciso, adecuado y lógico. Para ello entran en juego, por tanto, una gran cantidad de datos de distintos formatos y orígenes, que tendrán que ser puestos en común. Los datos provienen de Yahoo Finance y de Kaggle [28], con otras fuentes adicionales para pruebas y entrenamientos de menor envergadura.

Así, en este caso, disponemos, por un lado, de lo que es toda la información financiera y bursátil de un determinado activo, mientras que por el otro lado, disponemos de información adicional de un activo, ajeno al entorno bursátil, que será, por un lado, como se ha realizado en el proyecto, todos los tweets sobre un determinado activo durante los últimos tiempos, o más adelante, idealmente, noticias sobre el mismo y un conjunto de datos, en resumen, que nos puedan arrojar más información sobre la fiabilidad y rentabilidad de un activo financiero.

Para ello, por el momento, aunque el proyecto es ampliable y posiblemente se ahonde en su desarrollo más en profundidad a lo largo del futuro, añadiendo una mayor cantidad y variedad de fuentes más elaboradas, se obtienen, por un lado, a fin de llevar a cabo todos los entrenamientos y pruebas pertinentes, información de un dataset de Twitter, con todos los tweets que mencionan a AAPL (Apple), GOOG/GOOGL (Google), AMZN (Amazon), TSLA (Tesla) y MSFT (Microsoft) a lo largo de los últimos cinco años, y por otro lado, la información bursátil de los activos, disponible tanto en un dataset local con los históricos de los últimos años como en la API de Yahoo Finance, vinculada con Pandas.

Los dos datasets manejados se encuentran así con los siguientes formatos, visibles en la figura 3.4 y 3.5. Adicionalmente se han empleado en ocasiones algún dataset adicional, ya sea para el entrenamiento del analizador de sentimientos, que finalmente se descartó, o para otras pruebas adicionales, aunque las dos principales fuentes de información actualmente son las dos mencionadas.

tweet_id	writer	post_date	body	comment_num	retweet_num	like_num	ticker_symbol
550441509175443456	VisualStockRSRC	1/01/2015	lx21 made $10,008 on $AAPL -Check it out! ht...	0	0	1	AAPL
550441672312512512	KeralaGuy77	1/01/2015	Insanity of today weirdo massive selling. $aap...	0	0	0	AAPL
550441732014223360	DozenStocks	1/01/2015	S&P100 #Stocks Performance $HD $LOW $S...	0	0	0	AMZN
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

Figura 3.4: Disposición del dataset de Tweets

En el caso del dataset de tweets (3.4), tenemos las columnas siguientes: **tweet_id**, que corresponde a un identificador único para cada tweet en la red (irrelevante para nosotros), **writer**, que es el autor del tweet (podría resultar relevante), **post_date**, que es el instante cuando se publicó (es de lo más relevante para este proyecto), **body**, que es el texto plano del tweet (de nuevo, extremadamente relevante), y por último, el numero de **comentarios**, **retweets** y **likes** de cada tweet (muy interesantes para un estudio más detallado de análisis de redes sociales para medir la influencia de cada miembro y poder ponderar los tweets más adelante). Por otro lado, disponen también del **ticker** de la empresa a la que mencionan en el tweet (algunos tweets mencionan a más de una, por lo que se repiten).

Y en el caso del dataset de una empresa cualquiera en bolsa (3.5), disponemos de los siguientes campos: **high**, que es el precio mayor que hubo en el día, **low**, que es el menor, **open**, que es al precio al que se abrió, **close**, que es el que tenía cuando se cerró el mercado, **volume**, que corresponde al volumen de transacciones que hubo en el día y **adj close**, que es el precio de cierre ajustado tras las diversas acciones corporativas de la empresa.

Date	High	Low	Open	Close	Volume	Adj Close
2/01/2018	86309998	85500000	86129997	85949997	22483800.0	82194328
3/01/2018	86510002	85970001	86059998	86349998	26061400.0	82576843
4/01/2018	87660004	86570000	86589996	87110001	21912000.0	83303658
...	...	...	...	...	...	...
4/05/2021	251210007	245759995	250970001	247789993	32756100.0	247789993
5/05/2021	249500000	245820007	249059998	246470001	21901300.0	246470001
6/05/2021	249860001	244690002	246449997	249729996	26491100.0	249729996

Figura 3.5: Disposición del dataset de cada activo

Estos datasets se guardan en formato CSV, principalmente, y se leen y manejan con Pandas, transformándolos ocasionalmente a arrays Numpy para calcular determinadas métricas o manejarlos de una manera más ágil, rápida y cómoda.

Cabe mencionar, antes de pasar con la siguiente sección relativa a la preparación del dataset, que los datos de los tweets son extremadamente voluminosos (nada más y nada menos que 4.336.445 tweets a lo largo de 5 años, alrededor de 2500 tweets diarios, con 1825 días. El dataset en total, tratandose de solo texto, ocupa aproximadamente 1 GB. Esto hace de él un dataset muy voluminoso, que no da problemas en su manejo, pero sí en su velocidad de ejecución.

3.3.2. Preparación del dataset

La fase de preparación de los distintos datasets es bastante variada, puesto que por un lado disponemos de datasets completamente numéricos, de los cuales valoraremos tan solo un par de series temporales basadas en fecha, y por otro lado, disponemos de un dataset (o varios, realmente, pues hay uno adicional que empleamos para el entrenamiento de un analizador de sentimiento) que están compuestos principalmente por texto. De esta forma, hablaremos de la preparación de ambos tipos de datos.

Empezamos así por los datos más relevantes, que son los financieros. Inicialmente disponemos de varias series temporales, cada una de ellas correspondientes a un determinado activo y en función de la fecha, aunque también podemos tenerlas del mismo día a distintas horas. Así, para trabajar con un solo dataset en lugar de manejar varios, podemos añadir una columna de compañía y juntarlos todos en un solo dataset, o hacer una lista de datasets accesibles con cada uno de los tickers [1], que nos darán el dataset de series temporales de cada uno de los tickers.

Generalmente nos importará solo el valor de cierre de acciones, aunque también le damos importancia en ocasiones, por otros motivos, al volumen de transacciones. Así, una de las primeras cosas que hacemos es descartar el resto de los datos para simplificar el estudio, ya que arrojan muy poca información adicional. Adicionalmente, en caso de disponer de series temporales muy largas, también nos desprenderemos de los datos más viejos, puesto que definen un mercado que no es el actual.

[1] Un símbolo bursátil o código bursátil, también conocido como **ticker**, es un código alfanumérico que sirve para identificar de forma abreviada las acciones de una determinada empresa que cotiza en un determinado mercado bursátil [29].

El trabajo con respecto a las series temporales se divide en dos líneas de trabajo: el análisis y la predicción. Durante el análisis, se realizan diversos preprocesados distintos, principalmente relacionados con el cálculo de estadísticas y visualización de las mismas, mientras que en predicción, gira entorno a mostrar un resultado de lo que sucederá en base a esos mismos datos.

De esta forma, los preprocesados que se hacen con estos datos son relativamente sencillos, llevando a cabo las operaciones comentadas con anterioridad en la sección 2.3.3.

En cuanto a los datasets de NLP, el preprocesado es bastante más complejo, en relación a los dos anteriores mencionados. Independientemente de para que vayamos a utilizarlos, ya sea para análisis de sentimiento o embedding, hemos de limpiar e higienizar los datos. Al tratarse de Tweets, hemos de considerar un preprocesado adicional al que se haría normalmente con texto. Esta limpieza previa, por tanto, lleva a cabo los siguientes pasos:

1.– Convertir el texto a minúsculas (para detectar *Good* y *good* como la misma palabra).
2.– Reemplazar URLs (todo lo que empiece por HTTP, HTTPS o WWW se reemplaza por la clave URL).
3.– Reemplazar emojis utilizando un dicionario predefinido, por claves.
4.– Reemplazar nombres de usuario por la clave USER.
5.– Reemplazar caracteres adicionales (no alfabéticos ni digitos) por espacios.
6.– Eliminación de repeticiones (3 o más letras seguidas iguales se cambian por 2, p.e. “Holaaaa” =>“Holaa”).
7.– Eliminación de palabras cortas (de longitud 1).
8.– Eliminación de stopwords y otras palabras deseadas.
9.– Lematización (conversión de palabras a su base, es decir, *Great* a *Good*).

Tras separar los datos en dos conjuntos entonces de *training* y *test*, con los datos ya higienizados y añadiendo una clase o categoría en función del objetivo que tengamos en mente, falta únicamente vectorizar antes de poder empezar a entrenar y a emplear modelos.

Posteriormente, vectorizamos como previamente se ha mencionado en la sección 2.3.2, y con esto tendríamos todos los pasos completados para poder empezar a trabajar con nuestros modelos y su entrenamiento, mediante la librería SKLearn y su módulo vectorizador *TFIDF*.

Cabe mencionar que es de vital importancia que tratemos exactamente igual los datos de entrenamiento y test, es decir, si vectorizamos, escalamos, transformamos o hacemos cualquier operación con los datos de entrenamiento antes de emplearlos en el modelo, debemos aplicar exactamente esta misma operación con los datos de test.

3.3.3. Entrenamiento de los modelos

El entrenamiento de los modelos de SKLearn y Keras se lleva a cabo con los datos anteriormente preprocesados. Llega entonces un punto importante, que es asignarles una clase a predecir a estos datos, darle un objetivo, que será distinto en función de cada problema, pero al tratarse de series

temporales, realmente, similar. En el primer caso, para las series temporales que modelan el precio de un activo en función del tiempo, hay un punto al que hemos de llegar, una conclusión, que facilita en gran medida el desarrollo de estos modelos, y es que queremos predecir un valor concreto en base a los anteriores, es decir, nuestros atributos, por ejemplo, para una red neuronal sencilla, podrían ser los últimos 50 datos de la serie temporal, y darle un peso en función de los mismos a cada uno de estos datos, a fin de calcular el siguiente, que posteriormente pasaría a formar parte de los argumentos para el siguiente punto. Es en este punto donde encajan tan bien las LSTM (sección 2.2.2, al tener memoria y estar retroalimentadas, lo que resulta en muchos sentidos ideal para nuestro problema, en primer lugar porque la salida de la primera iteración es importante para la segunda, y así sucesivamente, y en segundo lugar, porque de esta manera conseguimos "memorizar" parte de los valores, lo que resulta interesante. De esta forma, para concluir, debemos asignar adecuadamente lo que son los atributos y categorías de los datos antes de continuar apropiadamente.

En cuanto a los modelos de NLP, tenemos varios casos. En caso de aplicar Embedding para tratar de averiguar el precio de un activo, su incremento o decremento, utilizaremos el retorno diario del activo como clase, vinculada directamente a la fecha, y como atributos, se utilizarán todos los tweets que correspondan a dicho día, tokenizados y vectorizados apropiadamente. Así pues, la categoría o clase será definida por los activos y su cambio en el precio o su volumen, y los atributos el texto en sí. En caso de querer aplicar analisis de sentimiento sobre los mismos, sin embargo, no habría ya un target como tal, más allá del sentimiento en sí, sino que la salida de los datos sería otra de las entradas del modelo encargado de predecir el precio de un activo, es decir, valoraría no solo el valor de las últimas acciones sino también como ha ido variando el estado de ánimo de los tweets relacionados con las mismas en los últimos tiempos. Así, realmente, en ese segundo caso, los atributos serían los mismos, y el target sería un número que defina un estado de ánimo más positivo o negativo.

Teniendo ya en cuenta como es el entrenamiento, es hora de explicar como se llevará a cabo la visualización y muestreado de los resultados, que serán mostrados más adelante en el capítulo 4.

3.3.4. Visualización de los resultados

Para la visualización de los distintos resultados se utilizarán las ya comentadas librerías de Seabeans, Matplotlib y WordCloud. Estas librerías nos permitirán visualizar de una forma cómoda y atractiva los resultados obtenidos, tal y como se observará más adelante.

3.3.5. Evaluación de los resultados

A la hora de evaluar la calidad de los resultados se van a emplear diversas funciones. En el caso de los modelos de series temporales, el error cuadrático medio estandarizado y ponderado por la cantidad de puntos, y en el caso de los modelos con NLP, el error cuadrático medio.

4

Pruebas y resultados

En este capítulo se detallan los resultados que se han obtenido en los experimentos ya comentados y su posterior análisis. En primer lugar se comentará como es el entorno de pruebas que se ha empleado, y posteriormente se analizarán los resultados de cada una de estas ramas de trabajo, para posteriormente tratar de juntarlas en un proyecto/idea final, a modo de conclusión, y estudiar como mejora el resultado al trabajar con datos externos al entorno bursátil.

4.1. Entorno

El entorno de pruebas utilizado para un desarrollo y experimentación más rápido y sencillo, a fin de poder utilizarlo desde cualquier dispositivo y lugar con conexión a Internet, ha sido Jupyter Notebook, en Colab, ya que ofrece una gran versatilidad, rápidez en el desarrollo y una buena cantidad de recursos de manera gratuita, que además son estables.

Estos han sido realizados con Python y se pueden encontrar en la web del proyecto, en tfg-lougedo.herokuapp.com.

4.2. Análisis en bolsa

Comenzaremos viendo en esta sección todo lo relativo al estudio y análisis de series temporales que nuestro modelo podría llegar a aprender en su interior, y que pueden facilitar a un usuario a ver las tendencias que se muestran.

Un ejemplo de estas medias móviles se encuentra en la figura 4.1, donde observamos el resultado de calcular las medias móviles de 10, 20, 50 y 1000 días sobre los datos disponibles de Pfizer. Como podemos observar, la media móvil comienza ligeramente más tarde cuantos más datos necesita, y es una representación suavizada, como ya sabíamos, de lo que es la serie temporal, lo que nos permite intuir tendencias a medio y largo plazo, sin necesitar muchos más cálculos adicionales.

Por otro lado, podemos observar en la figura 4.2, el ratio de retorno diario, que viene siendo el ya

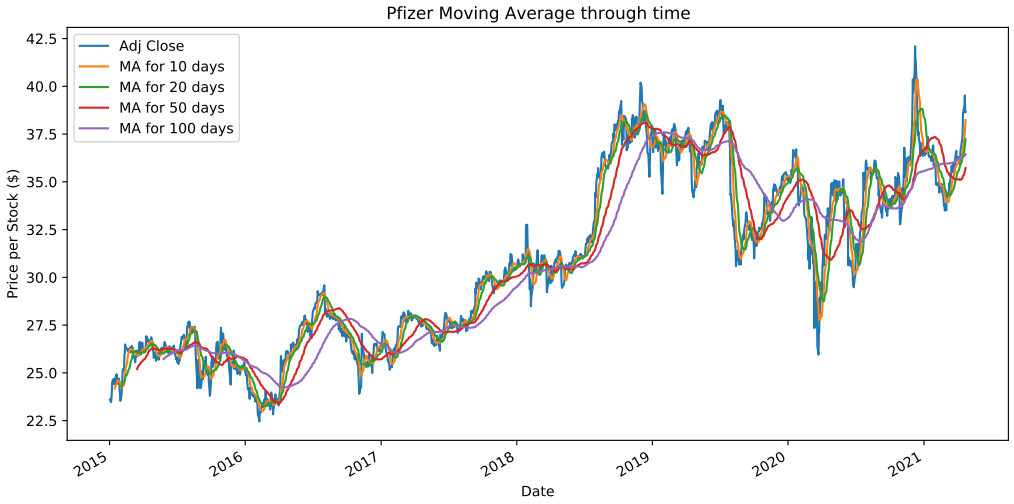


Figura 4.1: Ejemplo de media móvil de Pfizer

comentado porcentaje de cambio, corresponde a una serie temporal con forma de sierra, en la que cuesta más averiguar tendencias, al tener una forma menos uniforme. Calcular esta métrica resulta extremadamente sencillo, puesto que Pandas ya dispone de la función de *pct_change()*, que nos permitirá calcular el porcentaje de cambio entre el elemento actual y el anterior (generalmente, la fila previa del dataset que estemos manejando).

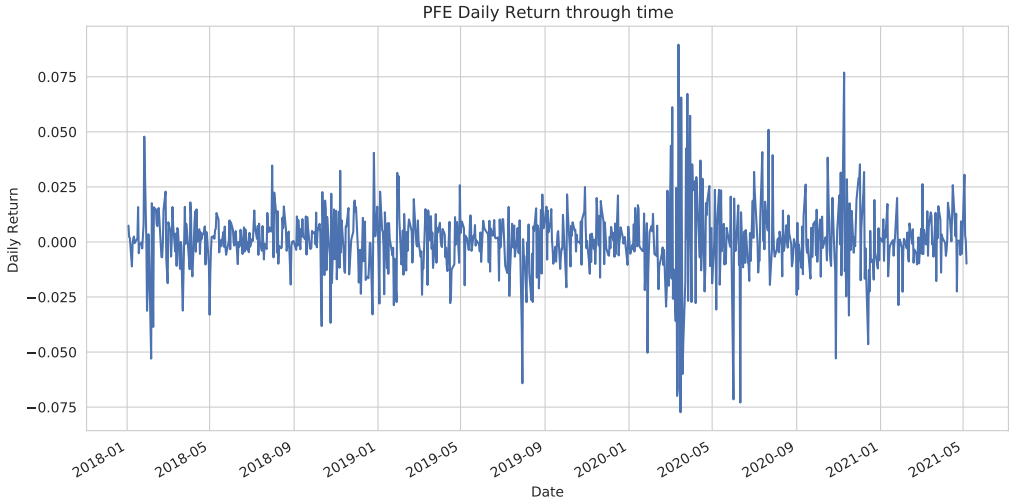


Figura 4.2: Ejemplo de Ratio de Retorno de Pfizer

Como podemos observar, en forma de serie temporal esta métrica resulta difícil de visualizar de forma lineal, aunque aporte bastante información, debido a la naturaleza de cambio de las series temporales en el entorno bursátil. Sin embargo, nos permite llevar a cabo un rápido análisis de frecuencias para saber que tipo de cambios son más habituales, así como cuales son mucho menos frecuentes, y si tiende a estar a la baja en más ocasiones, al alza pero más fuerte, o, en general, ver de forma rápida como cabe esperar que se comporte. Esto podemos representarlo mediante un histograma de frecuencias de la forma que observamos en la figura 4.3.

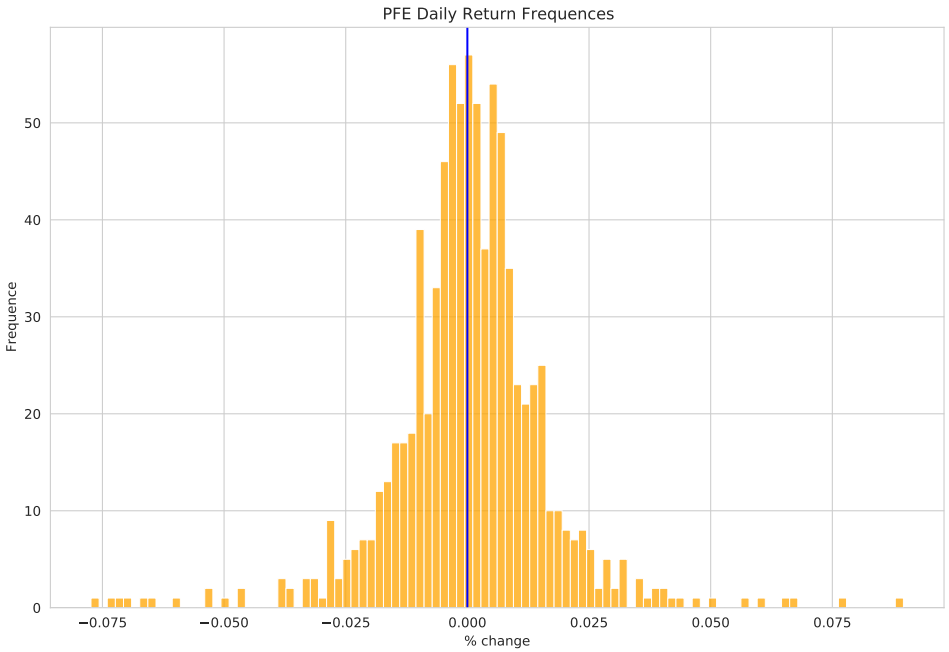


Figura 4.3: Ejemplo de histograma del Ratio de Retorno de Pfizer

En la figura observamos un histograma de frecuencias en el cual podemos ver cuantas veces aparece cada tipo de cambio en el rango de tiempo escogido. Como es de esperar, presenta una distribución normal centrada en el cero, tendiendo ligeramente a la derecha (es decir, a cambios positivos, como prácticamente cualquier activo de los que podamos estudiar, pues todos ellos buscan un mínimo de rentabilidad y tienden a crecer a la larga, o a desaparecer). Esto nos permite hacernos una idea de la fiabilidad de este activo, así como de los cambios que nos podemos esperar.

A partir de este dato, podemos obtener y comparar además con mucha más facilidad, puesto que ahora tenemos en una sola serie temporal lo que representaría el precio de un determinado activo y sus cambios, lo que nos permite ver como se comporta y relaciona con respecto a otras, con las anteriormente mencionadas correlaciones.

Para este apartado dejaremos de utilizar las acciones de Pfizer a modo de ejemplo y pasaremos a utilizar las seis acciones de cinco empresas tecnológicas bastante conocidas, que son Amazon, Apple, Google, Tesla y Microsoft. Son seis acciones y no cinco porque de Google cogeremos tanto GOOG como GOOGL, en vista a ejemplificar adicionalmente la correlación entre dos activos extremadamente relacionados. Si realizamos un pequeño análisis del ratio de retorno de cada una, nos encontramos con los resultados que observamos en la figura 4.4.

De primeras, este primer diagrama no nos aporta mucho, más allá de permitirnos ver que todas las empresas siguen una distribución normal bastante fiable. En todo caso, la única que sorprende mínimamente en relación a las demás es Tesla, puesto que tiene una distribución algo más ancha y dispar, lo cual quiere decir que da más y mayores bandazos, siendo por tanto menos fiable, generalmente, y más volátil a corto y largo plazo.

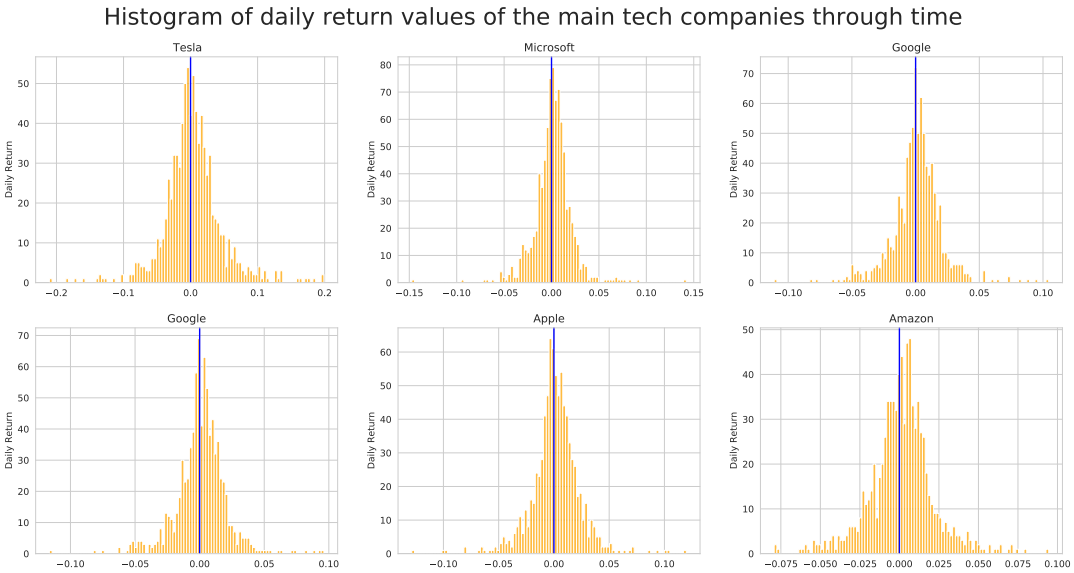
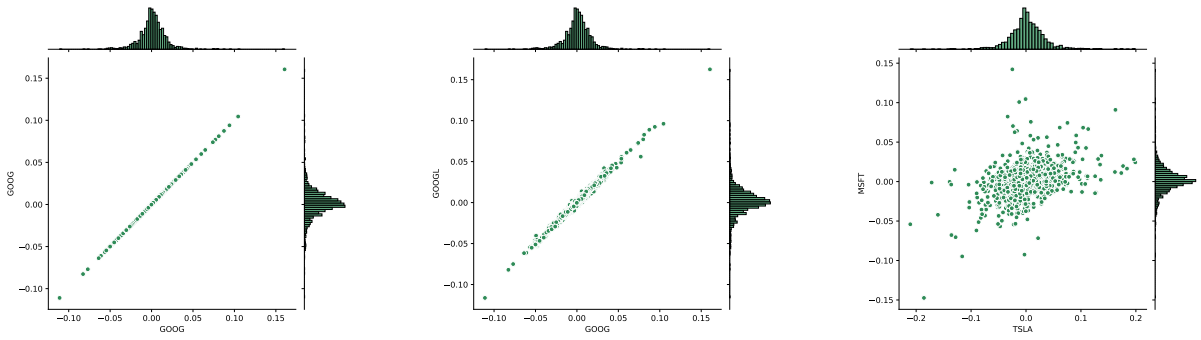


Figura 4.4: Histogramas de ratio de retorno diario

De esta forma, volviendo al concepto de correlación, queremos tratar de estudiar si las series temporales del precio de dos activos están relacionadas entre sí o no. Si realizamos ese estudio de manera individual, nos encontramos con algo como lo que se observa en las figuras 4.5(a), 4.5(b) y 4.5(c). En ellas vamos a ver, en primer lugar, como la relación de una acción consigo misma da lugar a una recta completamente derecha, como cabría de esperar, como es la relación entre dos activos muy fuertemente relacionados (como es GOOG con GOOGL, que no son el mismo activo como tal pero van de la mano), y como es la relación entre dos activos que no tienen relación en absoluto.



(a) Correlaciones GOOG-GOOG (b) Correlaciones GOOG-GOOGL (c) Correlaciones TSLA-MSFT

Figura 4.5: Correlaciones de distintas empresas tecnológicas

Como podemos observar, obtenemos una relación 1 a 1, con una recta, en el caso de Google consigo misma; algo muy próximo a una recta, como cabría esperar, con lo que es Google con sus dos posibles acciones, puesto que las dos están estrechamente relacionadas, y una nube de puntos sin aparente lógica alguna al comparar Microsoft y Tesla.

De esta forma, a mayor escala, podemos mostrar las relaciones entre las seis acciones anteriormente mencionadas, a partir de la funcionalidad ofrecida por la librería Python SeaBorn, podemos rápidamente extraer una comparativa como la observable en la figura 4.6, para estas seis empresas.

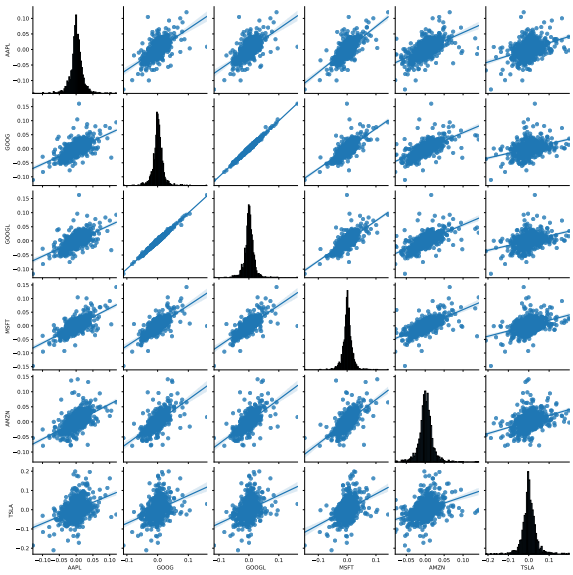
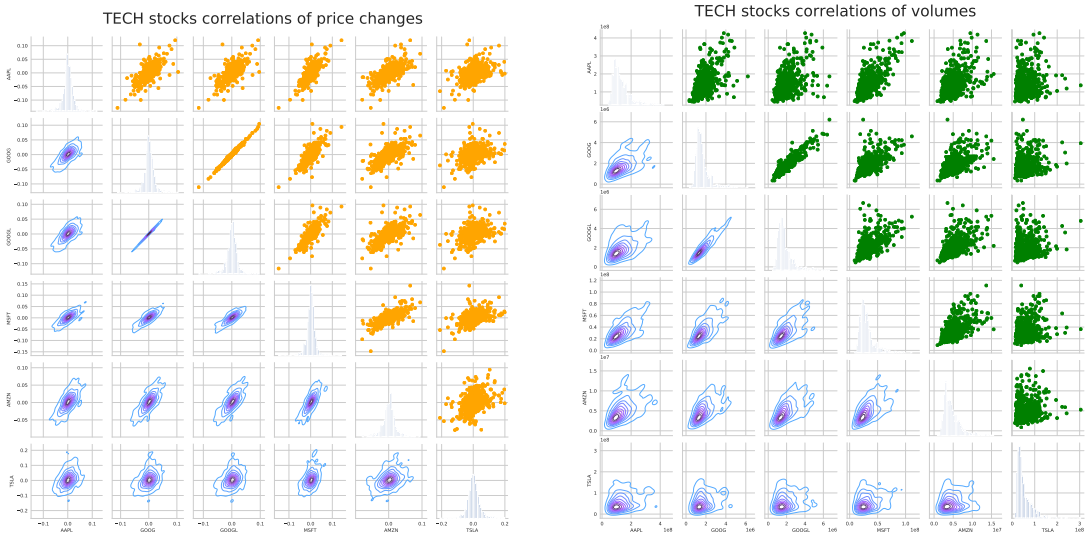


Figura 4.6: Grid simple de correlación de **cambio** entre los seis activos escogidos.

Y a partir de esta, podemos estilizarla de manera que obtengamos un resultado como los de las figuras 4.17 y 4.7(b), donde observamos las correlaciones en los cambios de precio y volumen.



(a) Correlaciones de precio

(b) Correlaciones de volumen

Figura 4.7: Correlación de **precio** y **volumen** entre los 6 activos escogidos.

En estas dos figuras, podemos comenzar a observar información de manera mucho más visual, lo que nos permite con ello extraer conclusiones a partir de los datos. En la figura 4.7(b) observamos detalles que son relevantes, pero no excesivamente. Es la relación del precio de las acciones entre sí.

Sin embargo, la que resulta verdaderamente interesante es la figura 4.17, donde podemos observar las correlaciones del cambio de los valores de las acciones. Es decir, cuando suben juntas, bajan juntas, etc. Podemos observar la estrecha y evidente relación que mantienen GOOG y GOOGL, como cabe de esperar, pero tambien intuimos otras relaciones adicionales, como la relación que tiene Microsoft con Google, o con Amazon, o el aislamiento de Tesla respecto al resto.

Esto nos permite extraer una matriz triangular (ya que las correlaciones son simétricas, no importa el orden) como la que se observa en la figura 4.8, que nos permitirá estudiar con mayor facilidad, aunque simplificando en gran medida los datos, estas relaciones.

Aquí observamos por fin de manera simplificada las correlaciones entre las seis series temporales que estamos manejando, con datos no modificados. Podemos observar una correlación estrecha como ya hemos mencionado en el caso de GOOG y GOOGL, una correlación estrecha entre Google y Microsoft y normales en el resto de los casos, que quieren decir que no hay mucha relación, aunque si alguna, por el mero hecho de que todas ellas tienden a crecer e incrementar con el tiempo.

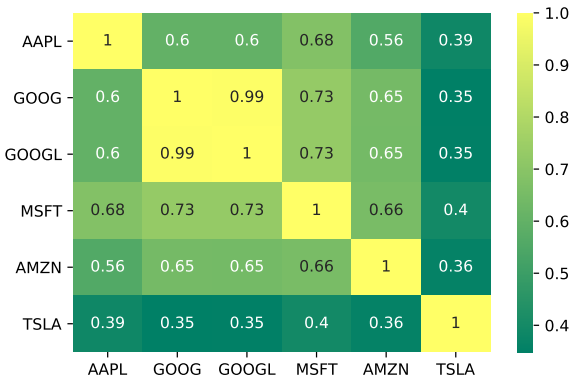


Figura 4.8: Correlaciones de **cambio** simplificadas entre los seis activos escogidos.

Sin embargo, aquí ya estamos pasando un pequeño detalle por alto a la hora de realizar un estudio tan simple, y es algo que quizás una red neuronal más compleja igual no pasa por alto: ¿que ocurre si por ejemplo, las acciones de Tesla generalmente suben cuando hay una bajada conjunta de las acciones de Microsoft y Apple pero no cuando baja una sola de ellas? Estas relaciones, con mayor o menor complejidad, se pueden dar por ejemplo en acciones como las de Pfizer, Moderna y AstraZeneca, o las de cualquier grupo de empresas relacionadas entre si, pero que no se trate de una relación de 1 a 1, sino más compleja, en la que entren en juego varios factores. Otras causalidades no tan evidentes podrían ser, por ejemplo, que las subidas en bolsa de una empresa minera estadounidense estén ligadas a las bajadas de empresas joyeras latinoamericanas, o relaciones mucho más sutiles e imperceptibles, que requerirían de un estudio extremadamente detallado y voluminoso.

De esta forma, se puede ir gestando una pequeña idea que será trabajada y explicada más adelante en la sección 4.3, relativa a los resultados de predicción, y en el capítulo 5, correspondiente a las conclusiones.

Por último, a partir de los valores de la serie temporal de retorno diario, podemos calcular una serie de estadísticas de cada serie temporal, que son el riesgo y el retorno esperado, y representarlos en una gráfica para compararlos de manera rápida en un determinado periodo de tiempo, tal y como se observa en la figura 4.9.

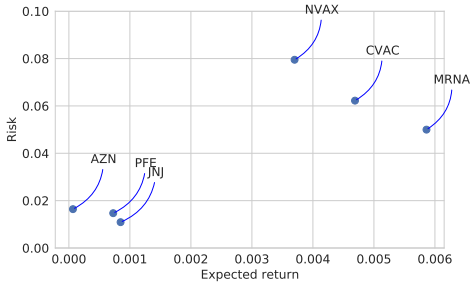


Figura 4.9: Riesgo y beneficio de las 5 empresas empleadas en el estudio

4.3. Predicción en bolsa

Visto el análisis, que es clave para entender como funcionará nuestro modelo y entrever las asunciones que podrá llevar a cabo y como podrá aprender de los datos, llega la hora de ver el punto más interesante del proyecto, que son los resultados de predicción. Para ello, emplearemos en este caso a modo de ejemplo las acciones de IBM (a fin de observar como esto no se reduce a las acciones gigantes tecnológicos sino que es generalizable, con una empresa que tiene una relación más cercana con nuestra universidad), trabajando en la predicción de valor de los activos, que puede resultar interesante para los clientes, y la predicción del volumen de transacciones, que puede resultar interesante para las empresas involucradas en la compra-venta de activos.

4.3.1. Predicción de precio de activos

Para este ejemplo, que se puede encontrar en el Notebook publicado en la web del proyecto, empleamos las redes neuronales recursivas, concretamente la LSTM, para tratar de realizar una predicción de los valores de un activo a través del tiempo. Para ello, en primer lugar, obtenemos los datos temporales de un determinado ticker, en el caso del ejemplo a continuación, el de IBM, empresa internacional con la que la UAM ha tenido la oportunidad de trabajar en reiteradas ocasiones y que nos ofrece un buen ejemplo del funcionamiento de este modelo. El resultado que obtenemos de la predicción es el observable en la figura 4.10.

Estas predicciones se realizan única y exclusivamente con los datos que se visualizan de color azul, sin llegar a computar los que se encuentran en color naranja, dando como predicción los que se encuentran en color verde. Pero, ¿como llegamos a obtener algo así?

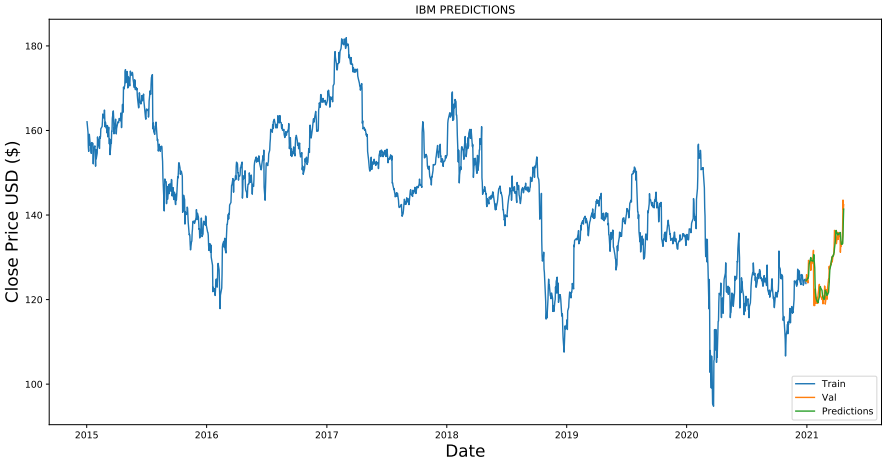


Figura 4.10: Predicciones del precio de IBM

Bien, como ya he comentado, lo primero es obtener los datos de la empresa a investigar, ya sea vía Yahoo Finance o algún otro proveedor o API en tiempo real, o del dataset que estemos empleando en su momento. Posteriormente, tras obtener esos datos, extraemos el campo (serie temporal) que queramos emplear. Este será, generalmente, el precio de cierre de cada determinado día, aunque podemos realizarlo de cualquier otra serie temporal, como veremos más adelante, pues podemos escoger cualquier otro dato relevante, ya sea el precio de apertura, cierre, o incluso el volumen de transacciones o cualquier dato que consideremos relevante de estudio.

Tras obtener la serie temporal, la partimos en dos pedazos, el que emplearemos a modo de entrenamiento, y el que emplearemos a modo de test. Posteriormente, escalamos los datos, y tras escalarlos, llega un punto clave, y es asignar una “clase” u objetivo, por así decirlo, una categoría, a cada dato de entrenamiento. Aquí es donde encontramos el primer hiperparámetro que podemos variar, pues no siempre vamos a obtener los mejores resultados con cada uno y cada serie temporal es única, como lo es cada empresa. En el caso del precio, generalmente, lo que mejor ha funcionado es considerar entre los 25 y 60 puntos anteriores, mientras que en el caso del volumen, como se verá más adelante, los mejores resultados han sido considerando entre los 10 y 20 puntos previos.

Tras construir nuestros dos conjuntos de entrenamiento y objetivo, llega el momento de crear el modelo en sí y entrenarlo. Aquí es donde entra de nuevo en juego la mano del programador, pues hay que tomar una serie de decisiones y formar una estructura apropiada y que de unos buenos resultados. Las opciones, como en las redes neuronales más simples, son muchas: desde un número más amplio de capas, con mayor profundidad, a un número más limitado de las mismas, pero con mayor tamaño. Cabe mencionar que no siempre más cantidad o más tamaño va a significar mejores resultados (aunque esta sea la tendencia general), y que los problemas que encontramos en otros modelos de aprendizaje automático también los encontramos aquí, como puede ser el overfitting. De esta forma, la elección de la estructura no es algo trivial, y es algo que dependerá en gran medida de cada una de las series temporales, tanto su número de capas como su cantidad de neuronas por capa.

Tras elegir estos parámetros, entrenamos el modelo que ya hemos estructurado, y posteriormente lo evaluamos. Los entrenamientos de estos modelos se basan, como se nos ha enseñado en Neurocomputación a lo largo de este último cuatrimestre de la carrera, en una idea muy básica, que gira entorno a reducir el error cuadrático medio en la mayor medida posible. Si conseguimos ajustar dicho error a una función, podemos ir variando sus parámetros para minimizarlo lo máximo posible, lo que dará lugar así cada vez a mejores resultados, hasta converger idealmente en el mejor resultado posible. La estructura seleccionada para este ejemplo ha sido en este caso la de la figura 4.11.

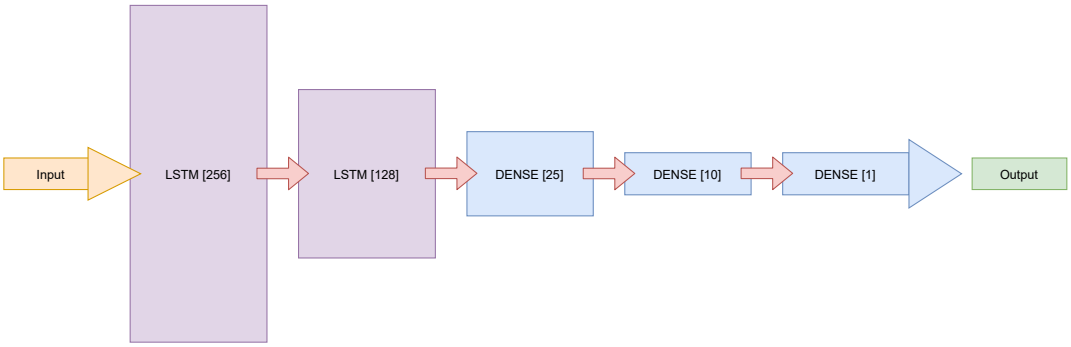


Figura 4.11: Estructura del modelo de predicción de precio de IBM

4.3.2. Predicción de volumen de transacciones

Como se ha mencionado con anterioridad, podemos predecir otros factores, más allá de únicamente el valor de los activos. Esto es importante, puesto que para los clientes generalmente resultará interesante el precio de una acción única y exclusivamente, pero para las empresas que se dedican al negocio del trading, ofreciendo aplicaciones y entornos para ello y encargándose de todo el papeleo para realizar las distintas transacciones y compras, ya sea RobinHood, Broker Naranja de ING, eToro, y otras tantas más, como puede ser incluso HeyTrade, donde actualmente trabajo, no es tan relevante el precio de una acción como lo pueden ser el volumen de transacciones. Esto se debe, principalmente, a que cobran (o pueden hacerlo) una determinada comisión por transacción. Es decir, a más transacciones, más beneficio. Con lo cual, es un campo que no debemos olvidar ni pasar por alto, por irrelevante que pueda parecer comparado con el precio de una acción.

Si tratásemos de aplicar las mismas ideas, y con la misma estructura a la predicción de volúmenes, obtendríamos inicialmente muy malos resultados (aunque seríamos capaces de predecir los grandes picos, tal y como muestra la gráfica de la figura 4.12.

Sin embargo, modificando los hiperparámetros de los que disponemos para esta serie temporal, obtenemos un resultado como el de la figura 4.13, ligeramente más precisos y válidos, que no predicen la magnitud del pico (esto no es fácil, puesto que además esta magnitud es la mayor de toda la serie temporal), pero si cuando va a haber un pico, lo cual resulta especialmente interesante.

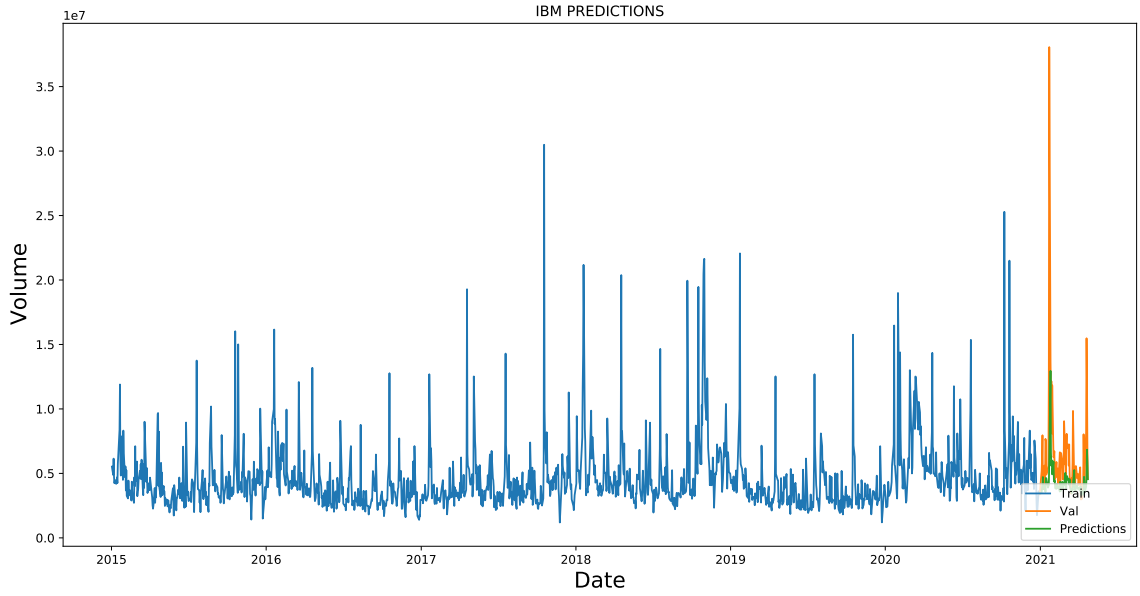


Figura 4.12: Mala predicción de volumen de ventas IBM

De esta forma, por ejemplo, empresas que se dediquen a la compraventa de acciones, podrían verse beneficiadas por esta información, ya que se podría realizar ofertas en determinadas empresas y determinados días, reduciendo la cantidad de comisiones o intereses aplicados, para así resultar más competitivas ante la competencia. Esta es una idea que exploraremos más adelante con el análisis del lenguaje natural.

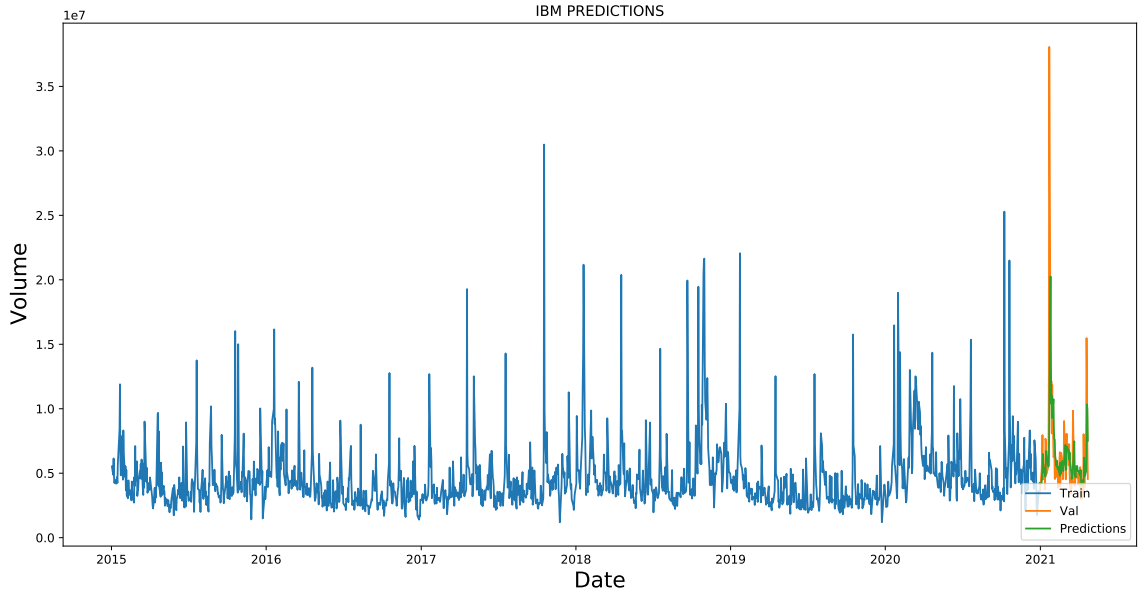


Figura 4.13: Buena predicción de volumen de ventas IBM

Hemos de considerar, aún así, que en el caso de los volúmenes, llevar a cabo la predicción es muchísimo más complejo, al carecer de tendencias y de una forma suavizada tan claras, aunque lo importante es realmente predecir que habrá un pico, más que el tamaño del mismo.

4.4. Aplicación de NLP

Este es otro de los puntos más interesantes del proyecto y es la aplicación del análisis del lenguaje natural para tratar de mejorar los resultados obtenidos anteriormente mediante la aplicación de NLP. Inicialmente, la idea sería ampliar esto a noticias, cualquier otra red social, blogs y demás información que pueda ser relevante en este campo, pero para iniciar el estudio sobre como información externa puede ayudar a precisar el estudio en mayor medida y arrojar más información sobre todo este campo, vamos a llevar a cabo un estudio inicial a partir de un dataset de Twitter con todos los Tweets sobre Apple, Google, Amazon, Tesla y Microsoft durante los años entre el 2015 y finales del 2019 [28].

4.4.1. Relación con entorno bursátil

En primer lugar vamos a observar como influye y como se relacionan ambos conjuntos de datos, es decir, las relaciones, causalidades y posibles correlaciones a priori entre uno y otro conjunto.

Podemos pensar de esta forma en varias ideas para estudiar si hay correlación entre la información de la que disponemos en Twitter y los valores que nos encontramos en Yahoo Finance. Las dos que vamos a estudiar inicialmente son las siguientes:

- Relación entre el volumen de Tweets y el volumen de transacciones.
- Relación entre el estado de ánimo de los Tweets en cuestión y el valor de las acciones.

Cabe mencionar que todos los puntos que se verán a continuación pueden mejorar con un preprocesado mucho más elaborado de los tweets, pues hemos de considerar que adicional al higienizado que se realiza, eliminando palabras y letras, y en general, conceptos, se debería de realizar un segundo higienizado y ponderado, en el que reduzcamos la cantidad de tweets, o al menos, su influencia, de los tweets de bots y similares, y demos más importancia a aquellos tweets con una mayor influencia y visibilidad, es decir, aquellos con un menor número de interacciones (likes y retweets) que, en resumidas cuentas, tendrán más influencia sobre el entorno que se está estudiando.

Así pues, si realizamos un primer estudio en el que relacionamos a priori volumen de tweets con volumen de transacciones, obtenemos los resultados de la figura 4.14 para el caso de Tesla (hay muchos más, concretamente 6, uno por cada empresa, pero se muestra el más representativo).

Como podemos observar, a priori hay una pequeña relación entre el número de Tweets y el volumen de transacciones. Cabe mencionar que en la gráfica se considera una desviación de un día.

A continuación, aplicando un pequeño análisis de sentimientos en los tweets, veremos también como se confirma la teoría de que ante sentimientos positivos de la gente respecto a una empresa se incrementa el valor de sus acciones, mientras que el sentimiento negativo no muestra generalmente una correlación del mismo tipo ni grado [30].

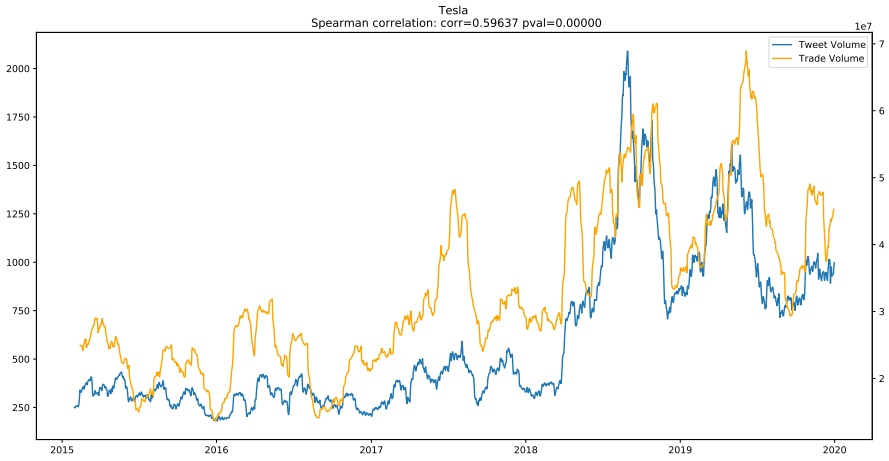


Figura 4.14: Relación entre el volumen de Tweets y el volumen de transacciones de TSLA

Tras etiquetar todos los tweets con la librería Afinn, obtenemos el resultado de correlaciones que se observa en la figura 4.15, en este caso de nuevo con Tesla.

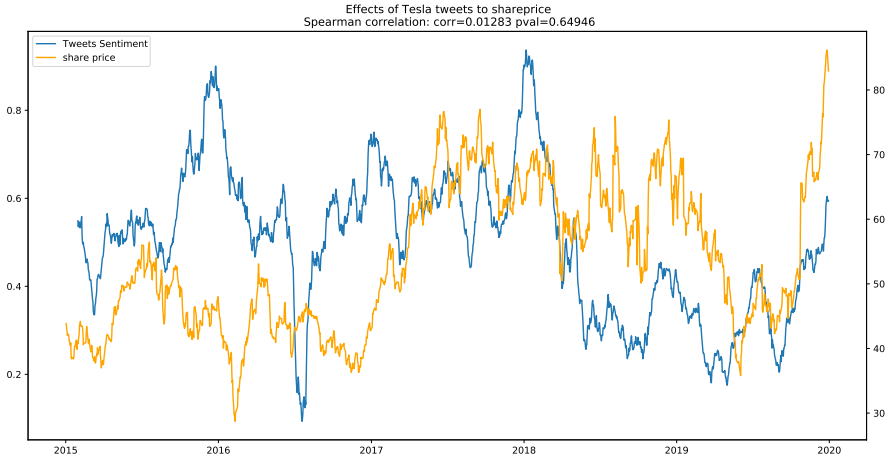


Figura 4.15: Relación entre el sentimiento de Tweets y el valor de acciones de TSLA

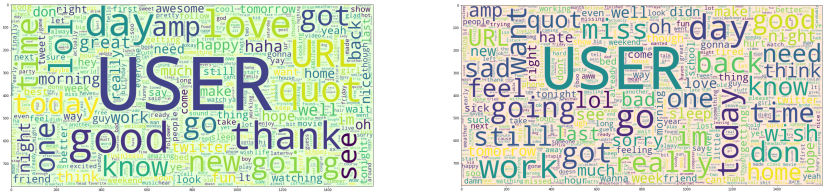
Veamos a continuación la aplicabilidad de estos datos a los modelos previamente empleados en predicción. Surgen aquí varias ramas, que son el análisis de sentimiento que acabamos de observar, para mejorar la predicción del valor, el embedding, para tratar de ver que tweets (positivos o no) se asocian a una subida en las acciones o en el número de transacciones o, por último, la influencia como mínimo de la cantidad de tweets en el día o días previos para calcular el volumen de transacciones.

4.4.2. Análisis del sentimiento

Tras aplicar el análisis de sentimiento, si tratamos de aplicar una red neuronal LSTM con dos entradas, una para la serie temporal de los activos y otra para la serie temporal del estado de ánimo, obtenemos el resultado observable en la figura 4.17(a), sobre la figura 4.17(b), donde observamos con

creces la diferencia entre considerar o no considerar los datos obtenidos vía Twitter. Se ha decidido utilizar Apple en este caso porque era la que tenía una mayor cantidad de tweets, y el preprocesado realizado en los tweets es mínimo (no se ha ponderado por usuarios, número de retweets, de likes, etc). Esto lleva a pensar que ponderando y considerando datos de otras fuentes adicionales, obtendremos aún mejores resultados, demostrando la efectividad tan solo de aplicar análisis de sentimiento.

Adicionalmente, se pueden extraer para hacer un estudio más visual diagramas como los de la figura 4.16 para cada empresa y momento del tiempo, a fin de saber que son sensaciones negativas y que son positivas, y poder modificar el sistema para cuadrarlo manual o automáticamente en aquello que sea malo o bueno.



(a) Términos positivos

(b) Términos negativos

Figura 4.16: WordClouds positivos y negativos.

Cabe mencionar que los datos mostrados en los Word Clouds son los más **frecuentes**, pero no los más **representativos**, es decir, palabras como *user* no tendrán prácticamente ponderación positiva o negativa, mientras que palabras como *happy* tendrán ponderación positiva, y palabras como *defeat*, negativa.

4.4.3. Embedding

Una idea que mejoraría en gran medida la anterior sería el empleo de embedding. Hasta ahora, lo que se ha hecho durante el proyecto para ver si un tweet era positivo o negativo (bueno, o malo) era recurrir a un análisis de sentimiento básico, llegando a modificar dicho analizador añadiendo un lexicon en el que se definían palabras con connotaciones positivas para el contexto y negativas (es decir, twekear el analizador para indicarle que palabras como bajada, pérdidas, bajos ingresos, etc son negativas, y palabras como subida, comprar, beneficio, etc., positivas).

Con esta misma línea, podemos llegar a una idea todavía mejor, que se podría llevar a cabo pero quedaba fuera de alcance para un proyecto de este calibre, que es, en resumidas cuentas, crear un analizador de Tweets (o de cualquier otro contexto, realmente, sería viable), en el que en lugar de asociar tweets a un sentimiento positivo o negativo, se asociasen tweets a una subida o a una bajada de bolsa, es decir, tratar de averiguar que tweets, que tipo de palabras y de formas de expresión de la gente van ligadas a una subida en bolsa, y cuales, por el contrario, están ligadas a una bajada.

(a) Con información de Twitter

(b) Sin información de Twitter

Figura 4.17: Comparativa entre considerar datos externos en el entrenamiento y no considerarlos.

Esto es una tarea perfectamente viable, pero realizarla en condiciones no es fácil, y con el volumen de datos del que se dispone adicionalmente requeriría de una gran cantidad de recursos y de tiempo de entrenamiento. Sin embargo, sin duda alguna, sería la mejor ruta de acción para llevar a cabo un proyecto de este estilo, puesto que es, con total seguridad, lo que mejor debería de funcionar.

De esta forma, para llevar a cabo la idea, disponemos ya de las herramientas necesarias: el conjunto de tweets, a modo de atributos, y el conjunto de stocks, a modo de clase. No consiste tanto en predecir el valor que va a tener un determinado activo a raiz de los tweets sobre la compañía, sino en ver cuales se asocian a subidas y cuales a bajadas para predecir en función de ello.

4.5. Comparativa de resultados

En este campo es difícil encontrar resultados con los que comparar, menos aún todavía con los datos temporales accesibles para poder contrastar adecuadamente (devuelven tan solo una gráfica), siendo todas las plataformas de predicción de stocks bastante crípticas y de pago en su mayoría. Aún así, podemos comparar, con las predicciones que se realizan, aunque sea visualmente, en otros proyectos independientes, como puede ser el de Serafeim Loukas [31], en la figura 4.18.

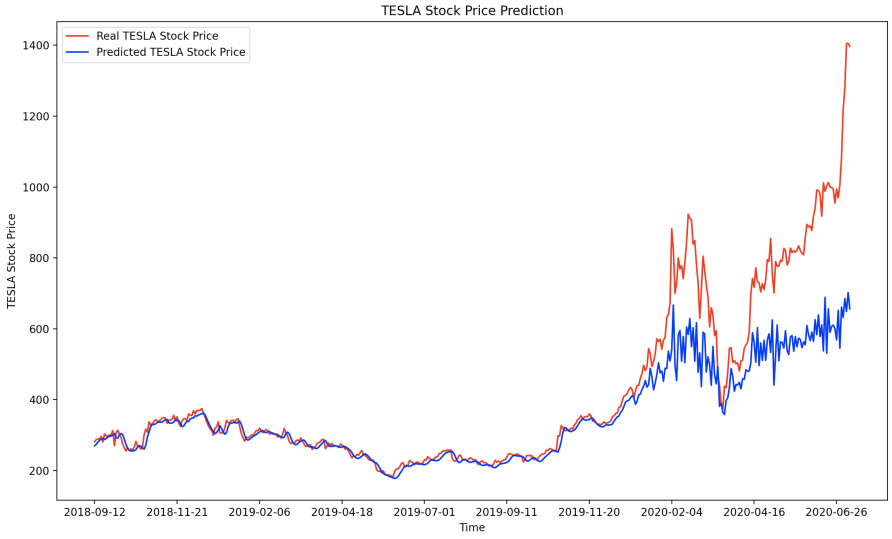


Figura 4.18: Resultados de Serafeim Loukas

Como podemos observar, ambos resultados son buenos. Pese a que el de nuestro proyecto pueda parecer mucho más preciso, cabe considerar que el de Loukas está funcionando con una mayor cantidad de tiempo, y con una empresa como Tesla, que es extremadamente volátil.

Adicionalmente, hemos observado que ya solo añadiendo el analizador de sentimiento, somos capaces de obtener unos resultados mejores a los que obtendríamos sin ellos, demostrando en el

proceso que podemos utilizar información externa al entorno bursátil (aunque relacionada, en gran medida) a la hora de llevar a cabo predicciones sobre el mismo, y que las correlaciones que hemos encontrado son suficientes para poder permitirnos aprovechar estos datos con facilidad, así como el embedding mejoraría en todavía mayor medida estos resultados. Es por ello que empresas como FinBrain ya están poniendo en uso el análisis de sentimiento en sus predicciones.

4.6. Aplicabilidad de Sistemas de Recomendación

Adicionalmente, esto se podría ver aplicado a sistemas de recomendación en el mundo del trading y el fintech [32], así como ha investigado eToro [1] con la idea de trading social. Pese a que realmente es un campo mucho más objetivo de lo que puede ser, por ejemplo, la música o el contenido audiovisual, donde los sistemas de recomendación funcionan excepcionalmente bien; eToro ya ha demostrado que es una idea, por lo menos, viable. Para ello se podrían llevar a cabo ideas que pueden resultar muy interesantes [33], tratando de monitorizar usuarios e incluso sus tweets para realizar análisis de la personalidad de los mismos y tratar de ver que empresas son más cercanas al mismo en términos de fiabilidad, riesgo, rentabilidad, y no solo “técnicos”, sino también del gusto, como puede ser el ámbito o contexto de trabajo, el campo en el que se mueve la compañía, las sensaciones que transmite a la gente, etc.

Se podrían llegar entonces a aplicar estas ideas, aun siendo un entorno más objetivo, para tratar de llevar a cabo recomendaciones para usuarios, llegando a aplicar ideas más complejas en base a las acciones que ya tenga un usuario, es decir, si tiene acciones de CocaCola recomendarle acciones de Pepsi, puesto que si baja la CocaCola tendrá la fiabilidad de tener acciones de Pepsi para que la pérdida sea menor, o lo que puede ocurrir ahora con las farmacéuticas y otras muchas empresas que como ya hemos demostrado están relacionadas entre sí en sus subidas y bajadas.

Esto se intentó llevar a cabo inicialmente en este proyecto, pero se descartó por completo su idea al ser muy limitada y, generalmente, privada, la cantidad de información al respecto, encontrando un dataset que podría hacer las veces de usuarios pero careciendo de compras de los mismos como para entrenar un sistema de recomendación en condiciones.

[1] En eToro aplican la idea de trading social, para que al igual que con la moda, por ejemplo, pueda haber expertos y gente que imite o siga la línea de esos expertos. Más información en https://www.etoro.com/trading/social/.

5

Conclusiones

En esta sección se comentan las conclusiones a las que se han llegado a través de este proyecto y su realización, el trabajo futuro que se puede llevar a cabo y expandir a raiz de este proyecto, las posibilidades de amplicación que ofrece y algunas ideas que se pueden llegar a aplicar a gran escala.

5.1. Conclusión

Con la aparición de las nuevas tecnologías, desde que nació el NASDAQ (primer mercado de valores electrónico), se ha ido modernizando cada vez más el mundo de la negociación bursátil y las herramientas empleadas en el mismo. Lo que empezó siendo, en su día, pequeños análisis estadísticos, realizados prácticamente a mano o con la ayuda rudimentaria de algún ordenador, se ha convertido en la aplicación de modelos de aprendizaje automático y de distintas herramientas y tecnologías cada vez más complejas y menos relacionadas, en apariencia, con el entorno bursátil.

Tras una exhaustiva investigación de los métodos de predicción existentes y de su empleo en casos básicos, rápidamente percibimos que se ven limitados en cierto momento por la cantidad de datos disponibles, y se intuye la necesidad de ampliarlos en busca de más precisión. El mercado de valores no es un entorno cerrado, sino abierto, y excesivamente influenciable, con lo que obtener datos externos que afecten al mismo, como puede ser a través de Tweets, e implementarlos para afinar la predicción, es, por el momento, uno de los pasos a seguir en la búsqueda de modelos más apropiados y precisos.

El aspecto más determinante ha sido ser capaz de valorar, a fin de obtener un resultado general, los valores de las acciones en bolsa, y posteriormente ampliar y mejorar los resultados con la información adicional obtenida, a fin de afinar esos resultados, pero nunca prescindir de los datos bursátiles, pues al fin y al cabo, son nuestra fuente de verdad.

Todo el código se puede encontrar en el siguiente enlace, donde se encuentra un landing y acceso facilitado a todas las implementaciones e investigación realizada, así como los datos empleados: `tfg-lougedo.herokuapp.com`.

5.2. Trabajo futuro

Surgen a partir de esta investigación una serie de ramas que se podrían llegar a valorar.

5.2.1. Posibilidades de ampliación

Hay una gran variedad de posibilidades de ampliación en este proyecto, que van desde la consideración de todo un entorno bursátil en una sola LSTM gigante, capaz de entender y valorar las series temporales de cada activo y comprender las correlaciones entre una o varias empresas, aplicandolas en predicción, al empleo de una mayor cantidad de fuentes higienizadas de datos, ya sea a partir de FaceBook, noticias de determinadas fuentes o incluso a partir de aquello que se emitiese por cualquier plataforma de streaming o televisión. Sería cuestión tan solo de implementar un interprete para esos datos y alimentarlos a nuestro predictor. Adicionalmente, se puede trabajar en la idea del embedding y ahondar más, pues al fin y al cabo, el análisis del sentimiento aplicado no es más que un embedding sencillo en el que tratamos de asociar sentimientos positivos a subidas.

De esta forma, el proyecto ofrece muchas posibilidades de ampliación, principalmente en la consideración de una mayor cantidad de fuentes, y en la implementación de una LSTM capaz de considerar y monitorizar los datos de diversas empresas simultaneamente, comprendiendo sus correlaciones y la forma en la que se influyen unas a otras, a fin de ser capaces de, en cierta manera, parametrizar el entorno bursátil y el mercado todo lo posible.

5.2.2. Ideas adicionales aplicables

Por otro lado, hay también algunas ideas adicionales que podríamos llegar a valorar, concretamente del entorno de los sistemas de recomendación, en auge últimamente. Una idea que es valorable, y que se podría añadir en el proyecto actualmente, es la de las opiniones de expertos, ponderando y dando más peso a sus veredictos o a lo que digan al respecto, por ejemplo, en Twitter. Se podría también implementar el sistema recomendador que se menciona en este proyecto, aunque inicialmente se necesitaría un tiempo para obtener los datos, y hay burocracia de por medio, pues aparentemente no es legal en España, por el momento, realizar recomendaciones a un usuario desde una aplicación de trading: se necesitan permisos especiales para ello y no pueden llevarse a cabo desde el mismo sitio donde se realiza la compra. Es por ello que plataformas como eToro, que realizan recomendaciones y ofrecen compra-venta de activos, lo hacen en base a sus usuarios, es decir, son los usuarios y no la plataforma en sí los que recomiendan. Aún así, el campo de la recomendación puede tener muchas ideas que aportar al mundo bursátil.

Bibliografía

[1] SoFiLearn, "A brief history of the stock market." https://www.sofi.com/learn/content/history-of-the-stock-market/, Jan. 2021.

[2] Wikipedia, "Stock." https://en.wikipedia.org/wiki/Stock, June 2021.

[3] A. Hayes, "What is trade?." https://www.investopedia.com/terms/t/trade.asp, Feb. 2021.

[4] R. J. Maestre, "Qué es el fintech, definición, sectores y ejemplos de startups." https://www.iebschool.com/blog/que-es-fintech-finanzas/, Oct. 2020.

[5] K. Pepi, "Best trading platforms 2021." https://tradingplatforms.com/, Jan. 2021.

[6] J. Brownlee, "What is time series forecasting?." https://machinelearningmastery.com/time-series-forecasting/, Aug. 2020.

[7] TensorFlow, "Time series." https://www.tensorflow.org/tutorials/structured_data/time_series, May 2021.

[8] D. Burba, "An overview of time series forecasting models." https://towardsdatascience.com/an-overview-of-time-series-forecasting-models-a2fa7a358fcb, Nov. 2020.

[9] C. Olah, "Understanding lstm networks." https://colah.github.io/posts/2015-08-Understanding-LSTMs/, Aug. 2015.

[10] C. Rus, "Gpt-3, el nuevo modelo de lenguaje de openai, es capaz de programar, diseñar y hasta conversar sobre política o economía." https://www.xataka.com/robotica-e-ia/gpt-3-nuevo-modelo-lenguaje-openai-capaz-programar-disenar-conv..., July 2020.

[11] MonkeyLearn, "Everything there is to know about sentiment analysis." https://monkeylearn.com/sentiment-analysis/, 2020.

[12] RapidAPI, "Top 8 best sentiment analysis apis (2021) [30+ reviewed]: Rapidapi." https://rapidapi.com/blog/sentiment-analysis-apis/, Jan. 2021.

[13] InteractiveChaos, "One hot encoding | interactive chaos." https://interactivechaos.com/es/manual/tutorial-de-machine-learning/one-hot-encoding, 2020.

[14] W. McKinney, "Pandas library." https://pandas.pydata.org/.

[15] T. Oliphant, "Numpy library." https://numpy.org/.

[16] Python, "Csv library." https://docs.python.org/3/library/csv.html.

[17] J. D. Hunter, "Matplotlib library." https://matplotlib.org/.

[18] M. Waskom, "Seaborn library." https://seaborn.pydata.org/.

[19] R. Aroussi, "Yfinance library." https://pypi.org/project/yfinance/.

[20] F. Chollet, "Keras library." https://keras.io/.

[21] D. Cournapeau, "Sklearn library." https://scikit-learn.org/stable/.

[22] N. Developers and Contributors, "Nltk library." https://www.nltk.org/.

[23] F. A. Nielsen, "Afinn library." https://pypi.org/project/afinn/.

[24] A. Mueller, "Wordcloud library." https://pypi.org/project/wordcloud/.

[25] T. Oliphant, "Scipy library." https://www.scipy.org/.

[26] Python, "Time library." https://docs.python.org/3/library/time.html.

[27] Python, "Datetime library." https://docs.python.org/3/library/datetime.html.

[28] Ömer Metin, "Tweets about the top companies from 2015 to 2020." https://www.kaggle.com/omermetinn/tweets-about-the-top-companies-from-2015-to-2020, Nov. 2020.

[29] Wikimedia, "Símbolo bursátil." https://en.wikipedia.org/wiki/Ticker_symbol, Nov. 2020.

[30] Y. Liu, "Recommendation system and retail trading,"

[31] S. Loukas, "Lstm time-series forecasting: Predicting stock prices using an lstm model." https://towardsdatascience.com/lstm-time-series-forecasting-predicting-stock-prices-using-an-l..., July 2020.

[32] P. K. M. Kanaujia, N. Behera, M. Pandey, and S. S. Rautaray, "Recommendation system for financial analytics," in *2016 International Conference on ICT in Business Industry Government (ICTBIG)*, pp. 1–5, 2016.

[33] D. Zibriczky, "Recommender systems meet finance: A literature review," 06 2016.


UAM
UNIVERSIDAD AUTONOMA
DE MADRID