MÁSTER EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE MÁSTER

MODELO FINTECH DE CORRELACIÓN DE LA TENDENCIA MEDIÁTICA Y EL VALOR DE LA EMPRESA.

Juan Alfageme Puga

Director: Dr. Antonio García de Garmendia

Madrid, 2023

MODELO FINTECH DE CORRELACIÓN DE LA TENDENCIA MEDIÁTICA Y EL VALOR DE LA EMPRESA.

Autor: Alfageme Puga, Juan.
Director: García de Garmendia , Dr. Antonio.
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

1. Introducción

Creación de un modelo matemático de predicción de la variación bursátil de una empresa mediante el estudio de noticias relevantes al sector de actuación de la empresa y el impacto sobre el valor de dicha empresa.

2. Definición del proyecto

El proyecto en primera instancia estudiará las tendencias mediáticas y las palabras más relevantes con respecto a las noticias de actualidad.

Una vez analizadas dichas palabras o parejas de palabras, se creará un primer modelo matemático que extraiga la información relevante de una noticia, la reduzca sin perder información, la agrupe y clasifique según la frecuencia de aparición en dicha noticia y la relevancia de cada palabra para el texto, ya sea negativa, neutra o negativa. Por último, cogerá esta información y construirá un vector único que un modelo de redes neuronales pueda leer.

En línea con el output del primer modelo matemático, se entrenará un modelo de redes neuronales que teniendo como input el vector único previo y una base de datos histórica de valores bursátiles, consiga predecir la variación del valor de una empresa basándose en la información extraída de la noticia y dicha base de datos.

Para demostrar la eficacia del modelo, se estudiará un caso práctico que arroje resultados dónde se verifique la menor incertidumbre ante movimientos bursátiles.

3. Hoja de ruta

La hoja de ruta del proyecto estará dividida en tres grandes partes: estudio de la información relevante, extracción y clasificación de la información, y resultado de variación de una acción bursátil:

A. La primera tarea constará de un análisis de las palabras dentro de una noticia que suponen una mayor tendencia mediática dentro de la población. Esta tarea permitirá crear una matriz que será utilizada por el primer modelo matemático para extraer solo la información relevante.
B. El primer modelo matemático extraerá las palabras relevantes, las reducirá de manera automática para agrupar palabras similares (con distinto género y número o distintos sufijos), las clasificará según su frecuencia de aparición en el artículo y les dará una puntuación de sentimiento basada en cuán relevante es cada palabra dependiendo de su frecuencia y su unicidad. Por último, mediante un modelo de machine learning

construirá un vector único que englobe toda esta información y sea legible el siguiente modelo.

C. El último paso será entrenar un modelo de redes neuronales que consiga dar un resultado de variación bursátil de una empresa. Este modelo tendrá como inputs una base de datos histórica de valores bursátiles de las empresas, y el vector creado en el primer modelo. Mediante redes neuronales, el modelo predecirá qué variación tendrá el valor de una empresa como respuesta a una determinada noticia de actualidad.

4. Conclusiones

El objetivo de este proyecto es conseguir un modelo que pueda ser utilizado para el estudio del mercado bursátil. El hecho de conseguir un modelo que reduzca la incertidumbre de una acción bursátil va a suponer un ahorro de dinero invertido por parte de los brokers y traders de bolsa. Otro objetivo importante es conseguir obtener una relación temporal con la variación bursátil. Es decir, conseguir saber a partir de cuánto tiempo y durante cuánto una noticia tiene repercusión y por tanto ese valor va a variar de manera anormal, y a partir de cuando una acción se puede considerar no afectada por una determinada noticia y por tanto vuelve a la normalidad.

FINTECH CORRELATION MODEL BETWEEN MEDIA TREND AND COMPANY VALUE.

Author: Alfageme Puga, Juan.
Director: García de Garmendia , Dr. Antonio.
Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

1. Introduction

Creation of a mathematical model to predict the stock market variation of a company by studying news relevant to the company's sector and its impact on the company's value.

2. Project definition

The project will first study the media trends and the most relevant words with respect to current news.

Once these words or pairs of words have been analyzed, a first mathematical model will be created to extract the relevant information from a news item, reduce it without losing information, group and classify it according to the frequency of appearance in the news item and the relevance of each word for the text, whether negative, neutral, or negative. Finally, it will take this information and build a single vector that a neural network model can read.

In line with the output of the first mathematical model, a neural network model will be trained to predict the change in the value of a company based on the information extracted from the news and the database.

In order to demonstrate the effectiveness of the model, a practical case will be studied that yields results where the lowest uncertainty in the face of stock market movements is verified.

3. Road map

The roadmap of the project will be divided into three main parts: study of the relevant information, extraction and classification of the information, and the variation result of a stock market action:

- The first task will consist of an analysis of the words within a news item that represent a higher media trend within the population. This task will allow the creation of a matrix that will be used by the first mathematical model to extract only the relevant information.
- The first mathematical model will extract the relevant words, automatically reduce them to group similar words (with different gender and number or different suffixes), classify them according to their frequency of occurrence in the article and give them a sentiment score based on how relevant each word is depending on its frequency and uniqueness. Finally, using a machine learning model will build a unique vector that encompasses all this information and makes the following model readable.
- The last step will be to train a neural network model that manages to give the result of stock market variation of a company. This model will have as input a historical

database of companies' stock market values, and the vector created in the first model. By means of neural networks, the model will predict what variation a company's value will have in response to a specific news item.

4. Conclusions

The objective of this project is to obtain a model that can be used to study the stock market. The fact of obtaining a model that reduces the uncertainty of a stock market action will save money invested by brokers and stock market traders. Another important objective is to obtain a temporal relationship with stock market variation. That is to say, to know for how long and for how much time a piece of news has an impact and therefore that value will vary abnormally, and from when a stock can be considered unaffected by a certain piece of news and therefore returns to normal.

Índice de la memoria

Índice de ilustraciones

Índice de tablas

Glosario

Machine Learning (Aprendizaje automático): Un subcampo de la inteligencia artificial que se enfoca en desarrollar algoritmos y modelos que permiten a las computadoras aprender y mejorar su rendimiento en tareas específicas a partir de datos y experiencia previa en lugar de programación explícita.

Deep Learning (Aprendizaje profundo): Una subrama del aprendizaje automático que utiliza redes neuronales artificiales profundas para modelar y resolver tareas complejas, como reconocimiento de imágenes, procesamiento de lenguaje natural y juegos.

Bag-of-words model (Modelo de bolsa de palabras): Un modelo de procesamiento de lenguaje natural que representa un documento como un conjunto no ordenado de palabras, ignorando la estructura gramatical y el orden de las palabras. Se utiliza para el análisis de texto y la recuperación de información.

Noun-phrase model (Modelo de frases sustantivas): Un enfoque de procesamiento de lenguaje natural que se centra en identificar y analizar frases sustantivas o grupos de palabras que funcionan como sustantivos en una oración.

N-gram model (Modelo N-grama): Un modelo de lenguaje que analiza secuencias de N palabras en un texto para predecir la siguiente palabra en una secuencia. Los N-gramas son útiles en tareas de procesamiento de texto como la predicción de texto y la traducción automática.

N-Word combination model (Modelo de combinación de N palabras): Un modelo que

utiliza combinaciones de N palabras en lugar de N-gramas para analizar el texto o el discurso. Puede ser útil en tareas de análisis de sentimientos o clasificación de texto.

2-Word combination model (Modelo de combinación de 2 palabras): Un modelo que utiliza combinaciones de dos palabras contiguas para analizar el texto o el discurso.

Porter Stemmer (Modelo de agrupación de palabras por la raíz): Un algoritmo que reduce palabras a su forma raíz, eliminando sufijos y afijos, utilizado en procesamiento de lenguaje natural para normalizar el texto y simplificar palabras relacionadas.

Dictionary-based approach (Enfoque basado en diccionario): Un enfoque que utiliza diccionarios o bases de datos predefinidas para realizar tareas de procesamiento de lenguaje natural, como la traducción o la detección de emociones en el texto.

Harvard-IV-4 psychosocial dictionary (Diccionario psicosocial de Harvard-IV-4): Un diccionario especializado utilizado en investigación psicosocial que contiene términos y puntuaciones asociadas a las dimensiones psicológicas y emocionales.

Feature selection without exogenous market feedback (Selección de características sin retroalimentación del mercado exógeno): En el contexto del análisis financiero, se refiere a la selección de características o variables para un modelo de predicción sin tener en cuenta la retroalimentación del mercado externo.

Feature selection employing exogenous market feedback (Selección de características empleando retroalimentación del mercado exógeno): En el contexto financiero, se refiere a la selección de características para un modelo de predicción teniendo en cuenta la retroalimentación o información externa del mercado.

TF-IDF (Term Frequency-Inverse Document Frequency): Un método de ponderación

utilizado en recuperación de información y análisis de texto para evaluar la importancia de una palabra en un documento en relación con una colección de documentos. Ayuda a identificar palabras clave significativas.

Support Vector Machine (Máquina de Soporte Vectorial): Un algoritmo de aprendizaje automático utilizado en clasificación y regresión que busca encontrar un hiperplano de separación óptimo entre dos clases de datos.

Support vector regression (Regresión de vectores de soporte): Una variante de las Máquinas de Soporte Vectorial utilizada para problemas de regresión, donde se busca encontrar una función de regresión que se ajuste a los datos de entrenamiento.

Brownian motion model (Modelo de movimiento browniano): Un modelo matemático que describe el comportamiento aleatorio de partículas en suspensión en un fluido, como el movimiento de las partículas de polvo en el agua.

Deep bidirectional long short-term memory (DBLSTM) (Memoria a corto y largo plazo bidireccional profunda): Un tipo de red neuronal recurrente (RNN) que utiliza una arquitectura bidireccional y capas profundas para procesar secuencias de datos, como texto o series temporales.

Recurrent neural networks (RNN) (Redes neuronales recurrentes): Un tipo de red neuronal que tiene conexiones recursivas y se utiliza para modelar secuencias de datos, como texto o audio.

Long-short term memory (LSTM) (Memoria a corto y largo plazo - LSTM): Un tipo de unidad de memoria en una red neuronal recurrente que permite retener información durante largos períodos y es especialmente útil para tratar problemas de secuencias largas.

Multiplication and Accumulation Operations (MACs) (Operaciones de Multiplicación y Acumulación - MACs): En el contexto del hardware de las redes neuronales, se refiere a las operaciones de multiplicación y suma realizadas en unidades de procesamiento para calcular el resultado de una red neuronal.

Chi-square (Chi-cuadrado): Una prueba estadística que se utiliza para determinar si existe una asociación significativa entre dos variables categóricas en un conjunto de datos. Se utiliza comúnmente en el análisis de tablas de contingencia.

Bi-normal Separation (BNS) (Separación bi-normal - BNS): Un enfoque estadístico utilizado en el análisis de características para la selección de características relevantes en problemas de clasificación. BNS mide la separación de las distribuciones de características para diferentes clases y se utiliza para evaluar la importancia de cada característica en la clasificación.

Word2Vec model (Modelo Word2Vec): Es un modelo de procesamiento de lenguaje natural que convierte palabras en vectores numéricos para capturar relaciones semánticas y similitudes entre palabras. Ayuda en tareas como el análisis de texto y la búsqueda de palabras relacionadas en grandes conjuntos de datos.

NumPy array (Arreglo NumPy): Es la estructura de datos principal que ofrece para almacenar y operar con datos multidimensionales de manera eficiente.

Gensim: Una biblioteca de procesamiento de lenguaje natural en Python que se utiliza para implementar algoritmos de modelado de temas y vectores de palabras, incluyendo Word2Vec y doc2vec.

TensorFlow: Una plataforma de código abierto desarrollada por Google para el aprendizaje automático y la inteligencia artificial. Permite la creación y entrenamiento de

redes neuronales y otros modelos de aprendizaje automático de manera eficiente.

Kernel (Núcleo): En el contexto del aprendizaje automático y las máquinas de soporte vectorial, un kernel es una función matemática que calcula el producto escalar de los vectores de características en un espacio de alta dimensión. Los kernels se utilizan para transformar datos en un espacio donde es más fácil realizar clasificaciones o regresiones.

Capítulo 1. INTRODUCCIÓN

En un mundo cada vez más impulsado por la información y la tecnología, el procesamiento de grandes cantidades de datos en tiempo real se ha convertido en un tema de vital importancia en la toma de decisiones en los mercados financieros. Las mejoras en las tecnologías han supuesto un incremento en la búsqueda de métodos de reducción de la incertidumbre relacionados con los movimientos de acciones bursátiles, siendo una tarea compleja que involucra una multitud de factores, desde eventos económicos hasta noticias y eventos globales. En este contexto, el presente proyecto de fin de máster se enfoca en desarrollar un sistema que consiga realizar un análisis de sentimientos relacionado con la reacción de las personas ante diferentes noticias. Mediante la utilización de técnicas de aprendizaje profundo basadas en redes neuronales se analizarán noticias y se ofrecerán sugerencias sobre el comportamiento futuro de las acciones de una empresa en el mercado, reduciendo la incertidumbre.

La volatilidad de los mercados financieros ha aumentado en las últimas décadas, y la información juega un papel crucial en la toma de decisiones de inversión. Los inversores buscan apoyándose en la tecnología identificar patrones y señales que para poder anticipar los movimientos del mercado. En este contexto, las noticias financieras y económicas son una gran fuente de información que supone un enorme impacto en los precios bursátiles.

El objetivo principal de este proyecto es desarrollar un modelo algorítmico capaz de recopilar y extraer la información relevante de noticias financieras, utilizando algoritmos de procesamiento y clasificación de la información. Posteriormente, se aplicará un algoritmo de redes neuronales, basado en redes neuronales recurrentes (RNN), para

analizar el sentimiento de las noticias y predecir si las acciones de una empresa específica tienen más probabilidades de subir, bajar o mantenerse estables en el corto-medio plazo.

Para llevar a cabo este proyecto, se utilizarán técnicas avanzadas de machine learning[1] y deep learning[2], aprovechando la capacidad de las redes neuronales para capturar patrones complejos en bases de datos.

El desarrollo de un programa de este tipo tiene un impacto significativo en el mundo bursátil, al proporcionar a los inversores una herramienta capaz de reducir la incertidumbre en la toma de decisiones. Además, este proyecto se enmarca en un contexto de grandes avances tecnológicos y cambios en los mercados financieros, donde la capacidad de procesar y analizar grandes volúmenes de datos de manera eficiente es fundamental.

En las secciones siguientes de este trabajo, se explorará en detalle la literatura relevante sobre la que se ha apoyado este proyecto, el modelo teórico de este proyecto explicado paso por paso y por último un caso práctico de la implementación de este programa, dando unos resultados como muestra de su viabilidad, así como su evaluación y potenciales aplicaciones en el mundo real.

[1] En español aprendizaje automático. Ver glosario
[2] En espalo aprendizaje profundo. Ver glosario.

1.1 ¿QUÉ ES LA INFORMACIÓN RELEVANTE?

El primer paso de este proyecto es tener una idea clara de lo que se considera información relevante. Dicha información se describe como la parte de un texto o noticia que supone un impacto en el lector y por tanto, da información sobre el contexto de la noticia.

Hoy en día la información es abundante y se encuentra presente en todos los ámbitos de nuestra vida: televisión, redes sociales, internet, etc. De ahí que saber extraer la información relevante suponga una tarea complicada a la vez que muy necesaria para los sistemas tecnológicos de procesamiento de información. El saber diferenciar la información relevante de la información mundana va a facilitar el trabajo de nuestro algoritmo y por tanto ayudará a dar unos resultados más precisos, al no tener perturbaciones derivadas de excesos de información.

Es por eso por lo que la primera parte del modelo teórico, de extracción y clasificación de la información relevante, va a jugar un papel diferencial a la hora de conseguir una mayor precisión en el modelo de redes neuronales, pues el entrenamiento de dicho modelo será más fácil y con menos perturbaciones.

Capítulo 2. ESTADO DEL ARTE

Este capítulo comentará la literatura relevante al respecto, indagando en los distintos modelos útiles para cada fase del proyecto y analizando los fallos de dichos proyectos que han servido de motivación para realizar este proyecto.

El desarrollo de los lenguajes de programación está incrementando la creación de modelos matemáticos que estudien los mercados bursátiles.

Tras examinar literatura relevante en relación a este trabajo, se ha podido extraer información relevante sobre modelos matemáticos en línea con el trabajo. La búsqueda de la literatura existente se ha basado en dos ramificaciones: cómo extraer y clasificar información de bases de datos de noticias, y cómo utilizar valores bursátiles junto a información sobre noticias para obtener una predicción de variación de un valor bursátil.

2.1 ESTADO DEL ARTE SOBRE LA EXTRACCIÓN Y CLASIFICACIÓN DE LA INFORMACIÓN

En relación con el primer bloque se ha encontrado información relacionada con distintas maneras de tratar una base de datos de noticias, para convertir cada pieza de información en un vector reducido[3] que otro modelo matemático pueda comprender, sin perder la información relevante de la noticia. Se ha indagado sobre cuáles son los temas relevantes[4]

[3] Michael Hagenau, 2013, Robert P. Schumaker H. C., 2009, Bellintani, Kristof Coussement, 2008
[4] The Thought Tree, 2021, Basilone, 2021

en el mundo bursátil, tanto los temas con más transcendencia como los indicadores financieros con más peso. Lo que se va a buscar en este trabajo es partir de una base de datos de noticias, conseguir extraer la información relevante relacionada con la categoría en la que se encuentra la noticia, el titular de la noticia y una descripción corta de la noticia basada en modelos matemáticos que extraigan las palabras relevantes de cada noticia.

Para esa extracción de las palabras relevantes de una noticia, se han estudiado distintos modelos matemáticos de extracción, como puede ser el *Bag-of-words model*, *Noun phrases model*, *Dictionary-based* o *N-gram*[5]. También se han estudiado softwares más específicos como el software *Stanford Parser*[6], utilizado en el *Noun phrases model*. Estos distintos modelos, los cuales se analizarán en profundidad en este capítulo, buscan palabras específicas dentro de la noticia y anotan qué palabras contiene el cuerpo de la noticia y con qué frecuencia se mencionan estas palabras. Cada modelo las extrae de una manera distinta, siendo algunos a priori más efectivos que otros para el caso a realizar en el trabajo, aunque se estudiarán distintos modelos más en profundidad para comparar su funcionamiento.

El segundo bloque consiste en coger valores bursátiles de una empresa de cierto sector, coger un vector de información resultante del modelo matemático del primer bloque y conseguir crear un modelo que arroje un resultado de predicción de la variación de esa acción en el mercado. Sobre el estudio de este modelo hay literatura con enfoques muy distintos, sin haber un modelo que tenga hasta el momento una eficacia mucho mayor que los demás. Algunos de los estudiados mediante literatura relevante son: Brownian motion

[5] Matthew Butler, 2009, Michael Hagenau, 2013
[6] Software > Stanford Parser, s.f.

model[7], Kaleckian model[8], Deep Bidirectional Long Short-Term Memory[9] y multi-regression model[10]. El fin de este segundo y último modelo matemático es arrojar un resultado de variación de la acción en el mercado bursátil para así reducir la incertidumbre asociada a un valor bursátil (50% de probabilidad de acierto en saber si el valor de una acción sube o baja).

A continuación, se indagará en los distintos modelos matemáticos que podrían ser útiles para las distintas etapas de nuestro proyecto. En este capítulo del estado del arte simplemente se explicarán cada uno de ellos en base a la literatura existente, para poder asentar las bases que permitan en los siguientes capítulos del modelo teórico explicar qué modelo se utilizará en cada paso y porqué.

Para las primeras etapas del modelo matemático, que incluyen la extracción y clasificación de la información, dos artículos se han estudiado en profundidad, por su completitud. Estos artículos son *"Automated news reading: Stock price prediction based on financial news using context-capturing features"* de Michael Hagenau[11] y *"Textual analysis of stock market prediction using breaking financial news: The AZFin text system"* de Robert P. Schumacker[12]. Ambos artículos realizan un estudio parecido al propuesto en la primera parte de nuestro proyecto, de ahí que se hayan cogido de base de apoyo a la hora de comparar las distintas herramientas plausibles para cada etapa de este primer modelo. Por tanto, la literatura relevante descrita sobre este primer modelo matemático sigue una línea parecida a la de estos dos artículos, cogiendo información de

[7] Rajpal, 2018
[8] Miranda, 2017
[9] Yinghao Ren, 2020, Early, 2022, Jie Wu, 2017
[10] Bellintani, s.f.
[11] Michael Hagenau, 2013
[12] Robert P. Schumaker, 2009

estos artículos para describir cada modelo,pero utilizando otra literatura propia en la cual se ha indagado más para cada tipo de modelo matemático propuesto, ya que estos artículos proponen una manera muy correcta de llevar a cabo esta extracción y clasificación de la información de las noticias, pero sin aportar demasiada explicación acerca de cada tipo de modelo.

2.1.1 LITERATURA RELEVANTE ACERCA DE LA EXTRACCIÓN DE INFORMACIÓN

La literatura relevante estudiada en este apartado comentará distintos modelos de extracción de la información relevante de un artículo de noticias. Estos modelos lo que buscan es extraer la información que se considere relevante, limpiando la parte de la noticia que no aporte ningún valor, como pueden ser preposiciones, artículos, determinantes, conjunciones y adjetivos, verbos y nombres que no tengan connotación relevante con respecto al cuerpo de la noticia.

1. El primero de los modelos es el modelo bolsa de palabras[13][14]. Es un método que se utiliza en el procesado del lenguaje para representar documentos ignorando el orden de las palabras. En este modelo, cada documento parece una bolsa que contiene algunas palabras. Este método anota todas las palabras separadas y cuenta la frecuencia de cada palabra en el documento. Aunque es un método sencillo y utilizado, este modelo presenta distintos problemas para el trabajo. En primer lugar, este método no tiene en cuenta el orden de las palabras, algo a tener en cuenta en un documento. Segundo, trata las palabras individualmente, lo que reduce la

[13] Del inglés Bag-of-words model. Ver glosario.
[14] Brownlee, 2017

posibilidad de extraer el significado de grupos de palabras que tienen un significado junto distinto del que tienen cuando aparecen por separado. Por último, el modelo trata todas las palabras por igual, derivando en un exceso de información no relevante al incluir artículos, preposiciones y determinantes.

2. El segundo de los modelos se denomina modelo de frases nominales[15], el cual sigue un enfoque parecido al modelo de bolsas de palabras, pero teniendo solo en cuenta las frases que incluyen sustantivos. Utiliza el software Stanford Parser[16], un software de Java implementado por la universidad de Stanford. Este modelo sigue acarreando el primer problema del modelo bag-of-words, pues no tiene en cuenta el orden de las palabras en el documento, y no trabaja con grupos de palabras que puedan tener un significado específico y relevante en el documento. Además, al ser un paquete de Java implementado por la universidad de Stanford, está principalmente desarrollado para ciertos idiomas: inglés, chino, árabe y alemán. También hay una opción de tener el software en español, pero es más limitada.

3. El modelo de N-gramas[17][18] hace una suposición simplificadora de que la probabilidad de la siguiente palabra en una secuencia depende solo de una ventana de tamaño fijo de m palabras anteriores. Este modelo es muy efectivo porque permite recolectar n palabras seguidas, consiguiendo recoger significados que una sola palabra no puede obtener. Además, este modelo se fija solo en palabras

[15] Del inglés Noun-phrase model. Ver glosario.
[16] Anon., s.f..
[17] Del inglés N-gram model. Ver glosario.
[18] Matthew Butler, 2009

específicas que se incluyan en el programa, lo que da una gran posibilidad a combinar palabras relevantes, las cuales se han estudiado previamente en este trabajo, con extracción de la noticia sin perder información.

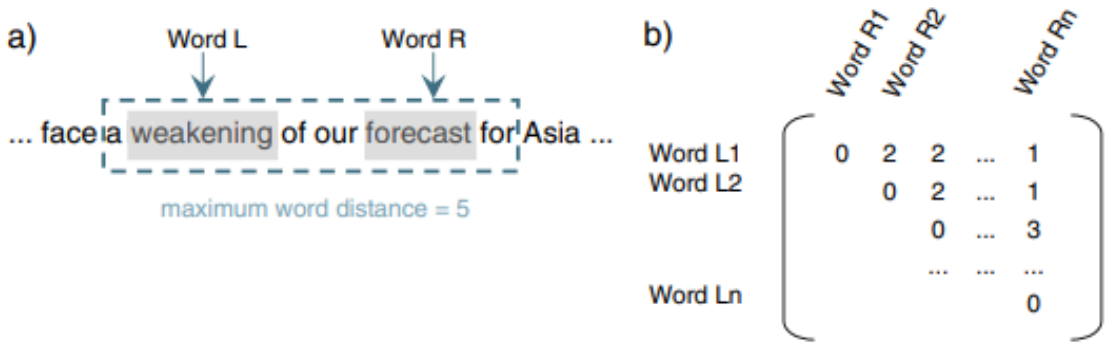

Ilustración 1: Metodología de utilización del modelo 2-word combination..Fuente (Michael Hagenau, 2013).

- El modelo de combinación de N-palabras[19] deriva del modelo de N-gramas, pero con la diferencia de que el grupo de n palabras que se buscan en el texto no tienen que estar seguidas, y pueden estar separadas entre sí una distancia de m palabras. La metodología de un modelo específico para 2 palabras está ilustrada en la Ilustración 1. Así, este modelo incluye grupos de palabras relevantes, pero sin la necesidad de tener que estar de manera sucesiva en el texto, evitando así completamente palabras sin contexto como pueden ser artículos, preposiciones o determinantes. Sin embargo, esta flexibilidad aumentada tiene un costo. Los modelos de

[19] Del inglés N-Word combination model. Ver glosario.

combinación de n palabras son mucho más costosos computacionalmente que los modelos de n-gramas, ya que requieren buscar en todas las combinaciones posibles de n palabras en la secuencia. Además, los modelos de combinación de n palabras pueden sufrir problemas de esparcidad cuando el cuerpo contiene muchas combinaciones de palabras raras o no vistas. Para reducir esos costes y los problemas de sobreinformación debido a las múltiples combinaciones posibles al haber n palabras en la secuencia, se introduce el modelo de combinación de 2-palabras[20], que como se puede observar en la ilustración gráfica de la Ilustración 1, este modelo específico del modelo de combinación de N-palabras restringe el número de n palabras a solo dos, consiguiendo extraer del modelo parejas de palabras, simplificando el modelo de combinación de palabras al solo tener que buscar parejas de palabras y no agrupaciones mayores.

2.1.2 LITERATURA RELEVANTE ACERCA DEL MODELO DE REDUCCIÓN DE INFORMACIÓN PORTER STEMMER[21].

Este modelo, el cual se comenta de manera superficial en el artículo de Michael Hagenau, ayuda a reducir información redundante, agrupando parejas de palabras que varíen en género o número, extraídas en el primer paso de extracción de la información, para poder realizar una clasificación de los grupos de palabras basado en la frecuencia de dichas

[20] Del inglés 2-Word combination model. Una especificación del N-Word combination model.
[21] Modelo de agrupación de palabras por la raíz. Ver glosario.

parejas. El modelo de Porter Stemmer[22] es un algoritmo sencillo, el cual no sería de vital importancia utilizarlo, de ahí que en nuestro otro artículo principal[23] no se comente este paso. Aun así, es cierto que este paso ayuda a limpiar y reducir información, algo importante cuando se trabaja con cantidades masivas de datos.

El funcionamiento de este modelo se basa en coger solo las raíces de las palabras, agrupando las palabras de distinto género y número mediante la agrupación de los términos con la misma raíz.

El Porter Stemmer sigue un conjunto de reglas y algoritmos para realizar el "stemming". Aplica una serie de transformaciones a las palabras para eliminar sufijos y terminaciones comunes. El objetivo es reducir las palabras a su forma central, llamada raíz, eliminando prefijos y sufijos pero preservando el significado de la palabra. El algoritmo se divide en 4 fases:

Fase 1: Normalización de plurales y eliminación de terminaciones comunes. Esta fase se centra en eliminar terminaciones en plural como “s” y “es”.

Fase 2: Normalización adicional y eliminación de terminaciones. Esta fase busca terminaciones impersonales como pueden ser las pertenecientes a los infinitivos, gerundios y participios, y ciertas terminaciones sin valor añadido a la palabra.

Fase 3a: Tratamiento de prefijos y sufijos específicos. Esta fase elimina los prefijos, y sufijos más comunes.

[22] Willett, 2006
[23] Robert P. Schumaker, 2009

Fase 3b: Ajuste de prefijos y sufijos adicionales. Esta fase busca los prefijos y sufijos más largos y menos comunes, al ser estos más fácil de encontrar una vez se han eliminado los prefijos y sufijos simples (fase 3a) y aplica reglas adicionales según la estructura de la palabra.

Fase 4: Eliminación de terminaciones en la etapa final. Esta última etapa elimina terminaciones específicas que no se hayan eliminado antes, para terminar de limpiar la raíz de la palabra, siempre y cuando se cumplan ciertas condiciones.

El Porter Stemmer es un algoritmo que se basa puramente en reglas y no depende de ninguna base de conocimientos externos o datos lingüísticos. Es un modelo sencillo, rápido y eficiente; aunque a veces puede arrojar raíces de palabras que no son reales. A pesar de sus limitaciones, sigue siendo un algoritmo de "stemming" popular y ampliamente utilizado debido a su simplicidad y efectividad en muchas aplicaciones de procesamiento de texto.

2.1.3 LITERATURA RELEVANTE SOBRE LA CLASIFICACIÓN DE LA INFORMACIÓN

El último paso consiste en la clasificación de la información. Esta etapa es de gran relevancia, pues dota de un valor cuantitativo a las palabras y al conjunto del texto. Los dos principales artículos comentados previamente tienen una serie de pasos para la clasificación de dicha información. Aunque no indagan mucho en los modelos, cada uno de los dos artículos utilizan técnicas distintas en alguna de las etapas de la clasificación, y parecidas en otras de las etapas.

Una de las primeras opciones que se estudian son:

1. **Enfoque basado en diccionario[24]:** Este método se basa en el procesamiento del lenguaje natural (PLN) que se apoya en el uso de diccionarios o léxicos para analizar y procesar texto. Consiste en asociar puntuaciones de sentimiento, etiquetas de partes de la oración u otra información relevante junto a cada colección de palabras y sus significados asociados. Este método se puede utilizar para realizar un análisis de sentimiento, en el cual se determinar el sentimiento o tono emocional de palabras o grupos de palabras, basándose en diccionarios de textos que tienen asociados puntuaciones positivas y negativas a cada palabra. Teniendo las puntuaciones individuales se pueden destacar cuáles son las palabras más polarizadas y cuál es el sentimiento general del texto.

   Dicho enfoque puede ser efectivo en ciertos contextos, pero también tiene limitaciones. Por ejemplo, puede tener dificultades con palabras que no están presentes en el diccionario, como se puede comprobar en estos artículos[25], donde se estudia las limitaciones de las palabras y su puntuación de sentimiento y se comprueba que la puntuación puede verse influenciada por el contexto en el que se utilizan las palabras. Este enfoque se basa solo en el "diccionario psicosocial de Harvard-IV-4"[26], un diccionario psicosocial desarrollado por el Programa de Análisis Lingüístico de la Universidad de Harvard. Es una herramienta que contiene una lista extensa de palabras y términos relacionados con aspectos psicológicos y sociales.

[24] Del inglés Dictionary-based approach. Ver glosario
[25] Tim Loughran, 2011, Paul C. Tetlock, 2008
[26] Del inglés Harvard-IV-4 psychosocial dictionary. Ver glosario

2. **Selección de características sin retroalimentación del mercado exógeno[27][28]:** Este enfoque se basa en clasificar las palabras según su puntuación de sentimiento en el contexto de predicción del valor bursátil. Se busca etiquetar cada palabra con información sentimental asociada, clasificación mediante un modelo de machine learning de las palabras más relevantes en el texto teniendo en cuenta la frecuencia de cada una de ellas y su puntuación de sentimiento.

3. **Selección de características utilizando retroalimentación del mercado exógeno[29]:** Este modelo es una extensión del anterior, pero utilizando un modelo estadístico para calcular la puntuación de sentimiento de cada palabra basándose en información exterior histórica sobre el mercado, para poder dar una puntuación de sentimiento que se vaya ajustando con los cambios de tendencia que puede haber en el mercado.

A continuación, se presenta una descripción de los pasos necesarios en este enfoque. Para el modelo anterior que no utiliza retroalimentación del mercado, el enfoque es el mismo pero sin la utilización de modelos estadísticos que resulten en puntuaciones de sentimiento para cada palabra:

a. **Recopilación de datos del mercado:** Obtener datos del mercado relevantes para el análisis de sentimiento y la predicción del valor bursátil. Los más importantes incluyen índices financieros, precios de acciones,

[27] Del inglés Feature selection without exogenous market feedback. Ver glosario.
[28] Marco Castangia, 2021, Edna S. Solano, 2022.
[29] Del inglés Feature selection employing exogenous market feedback. Ver glosario.

noticias económicas, entre otros. Estos datos deben tener un marco temporal coherente con las noticias utilizadas para extraer las palabras relevantes.

b. **Selección de características:** Utilización de técnicas de selección de características para identificar las palabras más relevantes y las características del mercado más influyentes para la clasificación de sentimiento. En esta parte es en la que el modelo sin opiniones del mercado se distingue del que utiliza opiniones del mercado, pues este último incluye métodos estadísticos para seleccionar las características más importantes, evaluando la relación entre dichas palabras relevantes y los datos del mercado.

Hay un último modelo de clasificación muy interesante, el cual puede ser utilizado también para dar una clasificación de las palabras. Este modelo es el último de los modelos de clasificación que no se engloba dentro de los modelos de machine learning. Este modelo se separa de los otros tres anteriores pues puede ser utilizado junto a alguno de los otros, como se analiza en este artículo[30]. El modelo matemático se llama frecuencia de término – frecuencia inversa de documento[3132].

Dicho algoritmo es una estadística numérica utilizada para evaluar la importancia de una palabra dentro de un documento en una colección o corpus de documentos. Se utiliza

[30] Michael Hagenau, 2013
[31] Del inglés TF-IDF (Term Frequency-Inverse Document Frequency). Ver glosario.
[32] Karabiber, 2023, Hans Christian, 2016.

habitualmente en recuperación de información y minería de textos para determinar la relevancia de las palabras en un documento concreto.

El modelo TF-IDF incorpora dos componentes:

1. **Frecuencia de términos (TF):** Este componente mide la frecuencia de un término dentro de un documento. Cuantifica la frecuencia con la que un término aparece en un documento y suele calcularse como el recuento bruto del término en el documento. Sin embargo, también pueden utilizarse otras variantes como el escalado logarítmico o la frecuencia booleana. Un valor más alto de TF indica una mayor aparición del término en el documento.

$$TF = \frac{nº\ de\ veces\ que\ el\ término\ aparece\ en\ el\ documento}{nº\ total\ de\ términos\ en\ el\ documento} \qquad [1]$$

2. **Frecuencia inversa del documento (IDF):** Este componente mide la importancia de un término en todo el corpus de documentos. Su objetivo es identificar los términos que son únicos o poco frecuentes en el corpus. La IDF se calcula dividiendo el número total de documentos del corpus por el número de documentos que contienen el término. El valor resultante se suele escalar logarítmicamente para amortiguar el efecto de los términos muy raros. Un valor IDF más alto indica que el término es menos común en el corpus. Las palabras exclusivas de un pequeño porcentaje de documentos (por ejemplo, los términos de la jerga técnica) reciben valores de importancia más altos que las palabras comunes a todos los documentos (por ejemplo, a, el, y).

$$IDF = \log\left(\frac{nº\ de\ documentos\ en\ el\ corpus}{nº\ de\ documentos\ en\ el\ corpus\ que\ contienen\ el\ término}\right) \quad [2]$$

La puntuación TF-IDF de un término en un documento se obtiene multiplicando el valor TF por el valor IDF. Cuanto mayor es la puntuación TF-IDF, más importante es el término en ese documento concreto.

$$TF - IDF = TF * IDF \quad [3]$$

El modelo TF-IDF ayuda a identificar la importancia de las palabras dentro de los documentos dando mayor peso a los términos que son frecuentes en el documento pero raros en el corpus. Esto permite distinguir los términos importantes que caracterizan el contenido de un documento de los términos comunes que aparecen en varios documentos.

El TF-IDF se utiliza ampliamente en diversas tareas de procesamiento del lenguaje natural (PLN), como la recuperación de información, la clasificación de documentos, el resumen de textos y la extracción de palabras clave. Proporciona una forma de representar datos textuales en forma numérica que puede utilizarse como entrada para algoritmos de aprendizaje automático.

2.1.4 LITERATURA RELEVANTE SOBRE CLASIFICACIÓN DE LA INFORMACIÓN MEDIANTE MACHINE LEARNING Y PARECIDOS

Este apartado del modelo va a englobar los modelos que cogerán la información clasificada anteriormente y la convertirán en un vector legible por un modelo de redes neuronales. La literatura relevante acerca de estos modelos nos arroja una inclinación por los modelos de machine learning que se verán a continuación, siendo los que arrojan mejores resultados en los distintos artículos, aunque también se ha encontrado algún modelo matemático que no incluye machine learning.

Esta primera revisión de la literatura va a analizar un algoritmo probabilístico en vez de un modelo de machine learning:

1. **Método de clasificación bayesiana[33]:** Un clasificador Naive Bayes es un algoritmo probabilístico de aprendizaje automático basado en el teorema de Bayes y en el supuesto de independencia condicional entre características. Se utiliza habitualmente en tareas de clasificación, sobre todo en el procesamiento del lenguaje natural y el análisis de textos. La suposición "ingenua" en Naive Bayes se refiere a la suposición de que todas las características son independientes entre sí, dada la etiqueta de clase. A pesar de esta suposición simplificadora, los clasificadores Naive Bayes han demostrado un buen rendimiento en muchas aplicaciones del mundo real.

[33] Mücahid Mustafa Saritas, 2019, Peter A. Flach, 2004

A continuación se muestra una descripción general de cómo funciona un clasificador Naive Bayes:

a. **Teorema de Bayes:** El clasificador se basa en el teorema de Bayes, que establece que la probabilidad de un suceso (en este caso, una etiqueta de clase) dada la evidencia (las características) es proporcional a la probabilidad de la evidencia dada el suceso, multiplicada por la probabilidad previa del suceso. Matemáticamente, puede representarse como:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \qquad [4]$$

- *Donde A representa una etiqueta de clase.*
- *Donde B representa las características.*

b. **Entrenamiento**: Durante esta fase, el clasificador aprende las probabilidades necesarias para la clasificación. Estima las probabilidades previas de cada clase calculando la proporción de casos que pertenecen a cada clase en el conjunto de entrenamiento.

c. **Probabilidades de las características:** El clasificador también estima las probabilidades condicionales de las características para cada clase. En el caso de la clasificación de textos, estas probabilidades suelen calcularse a partir de la frecuencia de aparición de cada característica (palabra) en cada clase. Para las características continuas, pueden utilizarse funciones de densidad de probabilidad (como las distribuciones gaussianas).

d. **Clasificación:** Para clasificar una nueva instancia, el clasificador calcula la probabilidad de cada clase a partir de las características observadas mediante el teorema de Bayes. Multiplica la probabilidad a priori de cada clase por el producto de las probabilidades condicionales de las características observadas en función de esa clase.

e. **Regla de decisión:** El clasificador asigna la etiqueta de clase con la probabilidad más alta como la clase predicha para la nueva instancia. Esta regla de decisión se conoce como regla del máximo a posteriori (MAP).

Los clasificadores Naive Bayes son eficientes desde el punto de vista computacional y requieren una cantidad relativamente pequeña de datos de entrenamiento en comparación con otros algoritmos. Son especialmente útiles cuando la hipótesis de independencia se cumple razonablemente bien o cuando se dispone de pocos datos de entrenamiento. Sin embargo, es posible que su rendimiento no sea óptimo si se incumple el supuesto de independencia o si existen fuertes dependencias entre las características.

En general, los clasificadores Naive Bayes se utilizan ampliamente en tareas de clasificación de texto, como el análisis de sentimientos, el filtrado de spam y la categorización de documentos, entre otras.

A continuación, se van a analizar los métodos de machine learning más utilizados. En nuestros dos principales artículos, los cuales han servido medianamente de guía para revisar la distinta literatura relevante, acaban con la implementación de este algoritmo. Se van a exponer dos métodos muy similares, puesto que el segundo es una variación de regresión del primero:

1. **Máquina de vectores de soporte (SVM)[34][35]:** Este modelo es un potente algoritmo de aprendizaje automático supervisado que se utiliza para tareas de clasificación y regresión. Resulta especialmente eficaz cuando se trabaja con conjuntos de datos complejos que presentan una clara separación entre clases o cuando se necesitan límites de decisión no lineales.

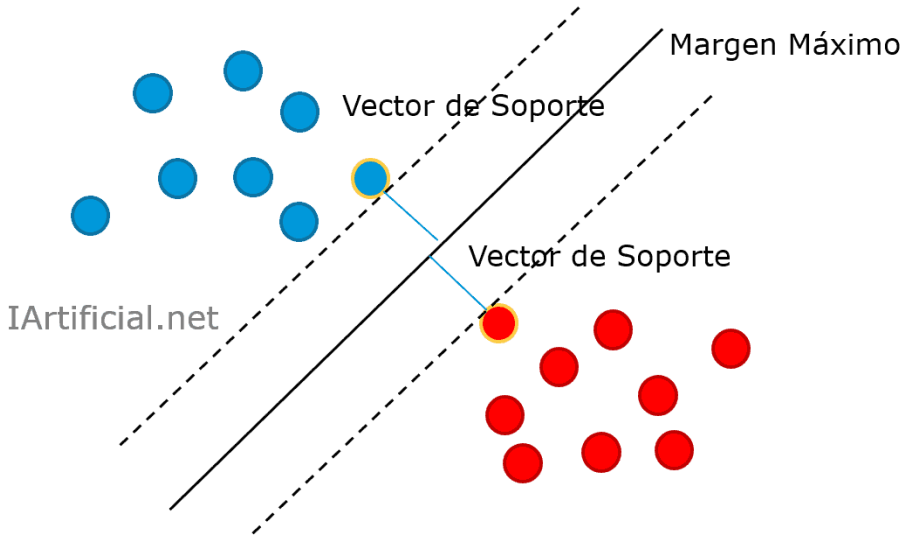

Ilustración 2: Ilustración gráfica de la máquina de vectores de soporte (SVM). Fuente IArtifical.net

La idea principal detrás de SVM es encontrar el hiperplano óptimo que mejor separa los puntos de datos en diferentes clases. El hiperplano se define como la frontera de decisión que maximiza el margen, que es la distancia entre el

[34] Gabriel Pui Cheong Fung, 2002, Joachims, 1998
[35] Del inglés Support Vector Machine, Ver glosario.

hiperplano y los puntos de datos más cercanos de cada clase, conocidos como vectores de soporte.

Estos son los conceptos clave y los pasos que intervienen en SVM:

a. **SVM lineal:** En el caso de datos linealmente separables, el objetivo de la SVM es encontrar el mejor hiperplano que separe las clases con el mayor margen. El hiperplano viene determinado por una combinación lineal de las características de entrada.

b. **SVM no lineal:** SVM también puede tratar datos separables de forma no lineal transformando las características originales en un espacio de mayor dimensión. Esto se hace mediante una técnica llamada el truco del kernel. La función kernel calcula el producto interno entre los vectores de características transformados, lo que permite a SVM operar implícitamente en este espacio de mayor dimensión sin calcular explícitamente la transformación.

c. **Margen y vectores de soporte:** SVM busca maximizar el margen, que es la distancia entre el hiperplano y los puntos de datos más cercanos de cada clase. Los puntos de datos más cercanos a la frontera de decisión se denominan vectores de soporte y desempeñan un papel crucial en la definición del hiperplano.

d. **Margen suave y regularización:** En el mundo real, los datos no siempre son perfectamente separables. SVM puede manejar estos casos mediante la introducción de un margen suave, lo que permite algunos errores de clasificación. El equilibrio entre la maximización del margen y la clasificación errónea se controla mediante un parámetro de regularización, conocido como C. Un valor más alto de C permite menos clasificaciones erróneas, pero puede conducir a un exceso de ajuste, mientras que un valor más bajo de C permite un margen más amplio, pero puede dar lugar a más clasificaciones erróneas.

e. **SVM para clasificación:** En la clasificación SVM, los puntos de datos se asignan a diferentes clases en función del lado de la frontera de decisión en el que caen. Las nuevas instancias se pueden clasificar determinando a qué lado del hiperplano pertenecen.

SVM tiene varias ventajas, incluyendo su capacidad para manejar datos de alta dimensión, manejar problemas tanto lineales como no lineales, y proporcionar un límite de decisión claro. Sin embargo, SVM puede ser sensible a los valores atípicos y a los grandes conjuntos de datos, ya que el entrenamiento del modelo puede ser costoso desde el punto de vista informático.

En general, SVM es un algoritmo versátil ampliamente utilizado en diversos ámbitos, como la clasificación de imágenes, la clasificación de textos, la

bioinformática y las finanzas, debido a su robustez y flexibilidad en el manejo de conjuntos de datos complejos.

2. **Regresión de vectores de soporte (SVR)[36][37]:** Es una variante de las máquinas de vectores de soporte (SVM) que se utiliza para tareas de regresión en lugar de clasificación. Mientras que SVM se centra en encontrar un hiperplano que separe mejor las clases, SVR pretende encontrar un hiperplano que se aproxime a una función continua minimizando el error entre los valores objetivo previstos y reales.

   Aquí están los puntos clave para entender acerca de la regresión de vectores de soporte:

   a. **Tarea de regresión:** SVR se utiliza para resolver problemas de regresión donde el objetivo es predecir una variable objetivo continua en lugar de clases discretas. Se utiliza comúnmente en escenarios como la predicción del mercado de valores, la estimación del precio de la vivienda y el análisis de series temporales.

   b. **Margen y banda de épsilon (ε)**: De forma similar a SVM, SVR busca maximizar el margen. Sin embargo, en lugar de definir el margen como una brecha entre las clases, SVR introduce una distancia margen **(ε)** de modo que todos los datos estén dentro de esa banda o tubo alrededor de la

[36] Guareño, s.f.
[37] Del inglés Support vector regression. Ver glosario.

función de regresión. Se considera que los puntos de datos que caen dentro de este tubo tienen un error o pérdida aceptable.

c. **Función de pérdida y épsilon:** La función de pérdida en SVR está diseñada para penalizar los puntos de datos que caen fuera del tubo insensible a épsilon. El parámetro épsilon **(ε)** determina la anchura del tubo y define el umbral a partir del cual los errores se consideran significativos. Los puntos de datos dentro del tubo o los que caen dentro de la región épsilon no se penalizan.

d. **Vectores de soporte:** En SVR, los vectores de apoyo son los puntos de datos que se encuentran en los límites del tubo o banda. Son los puntos de datos que más influyen en la colocación del hiperplano y la función de regresión.

e. **Truco del kernel:** SVR, similar a SVM, puede emplear el truco del kernel para manejar problemas de regresión no lineal. Al transformar las características de entrada en un espacio de mayor dimensión, SVR puede encontrar relaciones no lineales y aproximar funciones complejas.

f. **Hiperparámetros:** SVR incluye varios hiperparámetros que deben ajustarse para un rendimiento óptimo. Estos incluyen la elección del kernel, el parámetro de regularización (C), el valor épsilon y parámetros

específicos del kernel, como el grado del kernel polinómico o la anchura del kernel gaussiano.

g. **Predicción:** Una vez entrenado, el SVR puede predecir los valores objetivo de las nuevas instancias basándose en los valores de sus características. Los valores previstos se encuentran en los límites o dentro del tubo insensible a épsilon.

La regresión por vectores de soporte es una potente técnica que ofrece flexibilidad en el tratamiento de diversos problemas de regresión. Al incorporar el margen y el concepto de vectores de soporte, SVR puede encontrar una función de regresión que generalice bien a datos no vistos. Es especialmente útil cuando se trata de relaciones complejas y conjuntos de datos con ruido o valores atípicos.

2.2 ESTADO DEL ARTE SOBRE PREDICCIÓN DE VARIACIÓN DE UN VALOR BURSÁTIL

En relación con este segundo modelo se han encontrado estudios que abordan distintos enfoques de predicción de los valores bursátiles, desde enfoques estadísticos hasta algoritmos de aprendizaje supervisado.

2.2.1 ESTADO DEL ARTE MODELO DE MOVIMIENTO BROWNIANO

El primer modelo interesante en relación con el proyecto es el modelo de movimiento Browniano[38][39], también conocido como proceso de Wiener. Este modelo matemático puede emplearse como elemento fundamental en los modelos financieros para estimar el valor futuro de una acción. A continuación se exponen los requisitos necesarios para la utilización de este modelo en este proyecto:

1. **Dinámica del precio de las acciones:** El modelo de movimiento browniano supone que los rendimientos logarítmicos de una acción siguen un camino aleatorio. Es decir, los movimientos futuros del precio de una acción pueden representarse como una serie de incrementos o cambios aleatorios en los rendimientos logarítmicos.

2. **Movimiento browniano geométrico:** Para incorporar la volatilidad al modelo, se suele utilizar el movimiento browniano geométrico. Supone que el precio de las acciones sigue una tendencia determinista y está influido por un componente aleatorio. El modelo puede expresarse como:

$$dS_t = \mu S_t dt + \sigma S_t dW_t \qquad [5]$$

Donde:

- dS es el cambio infinitesimal en el precio de la acción.
- S es el precio actual de la acción.

[38] Del inglés Brownian motion model. Ver glosario.
[39] Rajpal, 2018

- μ es la rentabilidad esperada o tasa de deriva de la acción.
- dt es el incremento infinitesimal de tiempo.
- σ es la volatilidad o desviación típica de la rentabilidad de la acción.
- dW es el incremento de un proceso de Wiener (variable aleatoria) que representa el componente aleatorio.

3. **Estimación de precios futuros:** Mediante la simulación de múltiples trayectorias del precio de la acción utilizando el movimiento browniano geométrico, se puede estimar el valor futuro de la acción. Para ello se suelen utilizar métodos numéricos como la simulación de Monte Carlo. Las simulaciones generan una gama de posibles trayectorias de precios futuros en función de la tasa de deriva, la volatilidad y el horizonte temporal asumidos.

4. **Evaluación de las distribuciones de probabilidad:** A partir de las trayectorias de precios simuladas, puede analizar la distribución de probabilidad de los precios futuros de las acciones. Esto le permite estimar la probabilidad de que la acción alcance determinados niveles o supere umbrales específicos.

5. **Análisis de sensibilidad:** El modelo puede ajustarse cambiando los parámetros como la tasa de deriva y la volatilidad para realizar un análisis de sensibilidad. Esto ayuda a evaluar el impacto de diferentes supuestos en los precios futuros estimados de las acciones.

El modelo de movimiento browniano geométrico (GBM) se utiliza ampliamente para describir la dinámica de los precios de las acciones en la modelización financiera. Se trata de una ampliación del modelo básico de movimiento browniano que incorpora una tendencia y una volatilidad deterministas. El modelo GBM supone que los rendimientos logarítmicos de una acción siguen un camino aleatorio con deriva y volatilidad.

En este modelo, el término de deriva ($\mu S_t dt$) representa la tasa de crecimiento esperada del precio de las acciones a lo largo del tiempo. Recoge factores como los dividendos, el crecimiento de los beneficios y las tendencias generales del mercado. El término de volatilidad ($\sigma S_t dW_t$) representa las fluctuaciones aleatorias del precio de las acciones debidas al ruido del mercado y a acontecimientos inesperados.

El término estocástico, dW, sigue un proceso de Wiener o movimiento browniano estándar, que representa el componente aleatorio del modelo. Tiene la propiedad de incrementos independientes e idénticamente distribuidos, lo que significa que los cambios futuros del precio de las acciones no están influidos por su trayectoria pasada.

El modelo GBM supone que los rendimientos logarítmicos futuros del precio de las acciones se distribuyen normalmente. Esto implica que las variaciones porcentuales del precio de las acciones a lo largo de pequeños intervalos de tiempo se distribuyen de forma log-normal. Normalmente, se supone que los rendimientos logarítmicos son estacionarios, lo que significa que su media y su varianza permanecen constantes a lo largo del tiempo.

Utilizando el modelo GBM, se pueden simular las trayectorias futuras del precio de las acciones discretizando el tiempo y actualizando iterativamente el precio de las acciones basándose en la deriva y los componentes aleatorios. Las técnicas de simulación Monte

Carlo se emplean habitualmente para generar un gran número de trayectorias de precios simuladas. Estas simulaciones permiten estimar las distribuciones futuras de los precios de las acciones, las probabilidades de alcanzar determinados niveles y otras medidas estadísticas.

Aunque el modelo GBM se utiliza ampliamente y proporciona un marco útil para comprender la dinámica de los precios de las acciones, tiene limitaciones. Supone una deriva y una volatilidad constantes a lo largo del tiempo, lo que puede no ser cierto en la realidad. El modelo no tiene en cuenta los saltos o discontinuidades en los precios de las acciones, que pueden producirse debido a acontecimientos noticiosos o perturbaciones repentinas del mercado.

El estudio del artículo[40] aporta unos resultados de incertidumbre útiles para analizar la eficacia de este modelo. Este estudio predijo las variaciones bursátiles de las acciones de Apple cogiendo un número variable de valores bursátiles al cierre del día, con el fin de comprobar la eficacia y analizar si el uso de plazos más largos o más cortos mejora las predicciones, utilizando desde 20 días atrás hasta 1000 días atrás. Se realizó el estudio utilizando los supuestos de normalidad y Cauchy, variando los supuestos de distribución. Se puede comprobar en la Tabla 1 como la menor incertidumbre se alcanza en 100 días, mientras que el mínimo error cuadrático medio en 60 días. Esto enseña que la sobre información no siempre va a ayudar a arrojar valores más ciertos, al exponerse el estudio a un mayor número de variables y provocar confusiones en la muestra.

Sample size	MSENormal	P^Normal	MSECauchy	P^Cauchy

[40] Rajpal, 2018

20	1.68844	0.5252202	1.696666	0.5152122
30	1.676837	0.5188907	1.667149	0.5132637
40	1.67217	0.5056497	1.666231	0.5092817
50	1.665432	0.5202593	1.65844	0.5144919
60	1.66206	0.5174939	1.65229	0.5101709
70	1.665484	0.5216503	1.657948	0.503268
80	1.66508	0.5307629	1.659285	0.5139459
90	1.673651	0.5201812	1.669046	0.5004119
100	1.676367	0.5326716	1.673352	0.5128205
200	1.714528	0.5181191	1.708248	0.5163934
300	1.782596	0.5081154	1.773526	0.5153291
400	1.859956	0.5080264	1.853068	0.5028329
500	1.941562	0.5208127	1.93728	0.5178394
600	2.018263	0.5218978	2.014919	0.5177268
700	2.105357	0.5192519	2.104171	0.5181518
800	2.202516	0.5157159	2.202301	0.5157159
900	2.299643	0.5173053	2.296151	0.5166873
1000	2.390633	0.5158103	2.386399	0.5158103

Tabla 1: Tabla resultados modelo de movimiento Browniano. Estudio del artículo (Rajpal, 2018)

Este modelo se basa en la hipótesis de distribución normal de los precios de las acciones debido a su deriva y volatilidad, por lo que es un modelo que no alcanza una gran eficacia en predicciones debidas a sucesos extraños, como puede ser la aparición de una noticia que suponga un cambio drástico, parte del objetivo de este proyecto.

Como se puede analizar en la Tabla 1, los resultados arrojados por el artículo demuestran que este modelo estadístico apenas reduce la incertidumbre, con resultados ligeramente por encima del 50% para ambos estudios, tanto el de la distribución normal como el de la sucesión de cauchy, siendo esta segunda la que obtiene los mejores resultados en cuanto a incertidumbre, siendo la menor un 53,27% de probabilidad de acierto, en el caso de 100 días.

2.2.2 ESTUDIO DEL ARTE MODELO DE REGRESIÓN MÚLTIPLE

El segundo modelo de predicción de valores bursátiles es un modelo de regresión múltiple, el cual puede utilizarse para analizar la relación entre múltiples variables independientes y la variable dependiente de los precios de las acciones. En este contexto, la variable dependiente serían los precios de las acciones observados, y las variables independientes podrían incluir diversos factores que pueden influir en los precios de las acciones. El artículo[41] formula un modelo de regresión múltiple para los precios de las acciones:

[41] Bellintani, s.f.

$$Y = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 X_1} + \boldsymbol{\beta_2 X_2} + \cdots + \boldsymbol{\beta_n X_n} + \varepsilon \quad [6]$$

Donde:

- Y representa la variable dependiente, que es el precio de las acciones.
- X1, X2, ..., Xn son las variables independientes, como indicadores económicos, métricas específicas de la empresa o factores de mercado que se cree que afectan a los precios de las acciones.
- β0 es el intercepto o término constante.
- β1, β2, ..., βn son los coeficientes que miden el impacto de cada variable independiente en el precio de las acciones.
- ε es el término de error, que representa la variación no explicada del precio de las acciones.

Para utilizar el modelo de regresión múltiple en la predicción del precio de las acciones, las variables independientes (X1, X2, ..., Xn) deben seleccionarse cuidadosamente en función de su relevancia potencial y su impacto en el precio de las acciones. Estas variables pueden incluir factores como los tipos de interés, la inflación, los beneficios de las empresas, los índices de mercado o indicadores específicos del sector.

Los coeficientes (β1, β2, ..., βn) estimados por el modelo de regresión representan el cambio esperado en el precio de las acciones para un cambio de una unidad en la variable independiente correspondiente, suponiendo que todas las demás variables permanecen constantes. Los coeficientes positivos indican una relación positiva entre la variable independiente y el precio de las acciones, mientras que los coeficientes negativos indican una relación inversa.

Para el estudio de este artículo se utilizaron las siguientes variables independientes:

- Rendimiento del mercado a 1 día.
- Cierre del mercado a 1 día.
- Relevancia del artículo.
- Tipo de artículo (1=positivo, -1=negativo, 0=neutral).

También se incluyeron dos variables de control, que en principio no deberían influir en el valor de la acción:

- Urgencia del artículo.
- Longitud del artículo.
- Número de compañías mencionadas en el artículo.

Este artículo tiene un mayor parecido con el objetivo de nuestro proyecto, pues incluye datos de mercado (los valores de cierre y rendimiento) y datos de noticias (el tipo de noticia y la relevancia del artículo).

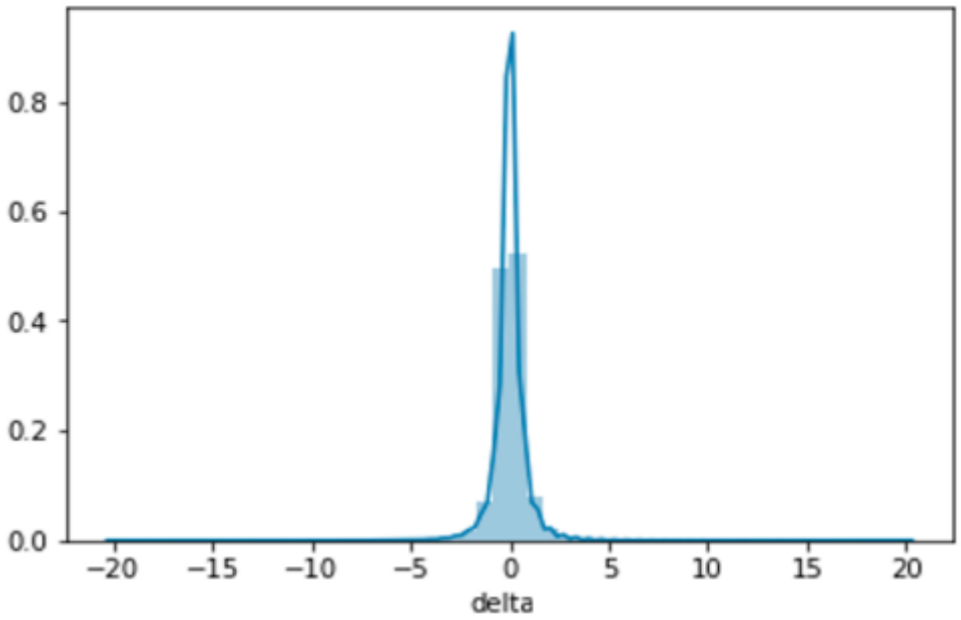

Ilustración 3: Distribución del precio de las acciones a lo largo del día. Fuente (Bellintani, s.f.)

La Ilustración 3 muestra que los deltas de las acciones se distribuyen normalmente con un valor medio de μ=0,0055 y una desviación típica de σ= 0,941, lo que indica que, entre 2007 y 2016, el precio de las acciones tendió a aumentar 0,0055 dólares al día, con una varianza de 0,94 dólares al día. También se puede deducir del gráfico que tiene una curtosis extremadamente alta, lo que implica que el punto del gráfico es agudo y sus colas están muy próximas a la media. Esto pone de relieve el hecho de que los precios intradía tienden a centrarse cerca de la media (y de 0, dado lo cerca que está μ de 0).

	coef	std err	t	P>|t|	[0.025	0.975]
Lagged Variable	17.8335	0.027	672.649	0.000	17.782	17.885
1 Day Market Return	-7.1779	0.031	-233.168	0.000	-7.238	-7.118
1 Day Market Close	-0.0003	0.000	-1.762	0.078	-0.001	3.89e-05
Article Urgency	0.0014	0.001	1.769	0.077	-0.000	0.003
Article Length	-6.897e-07	2.4e-07	-2.868	0.004	-1.16e-06	-2.18e-07
Number of Companies Mentioned	6.192e-06	0.000	0.043	0.965	-0.000	0.000
Sentiment Class	-0.0003	0.002	-0.123	0.902	-0.005	0.004
Adjusted Negative	-0.0818	0.007	-12.496	0.000	-0.095	-0.069
Adjusted Neutral	0.0111	0.004	2.646	0.008	0.003	0.019
Adjusted Positive	0.0173	0.005	3.256	0.001	0.007	0.028

Ilustración 4: Resultados estudio regresión múltiple, valores de las variables independientes.

Fuente (Bellintani, s.f.).

Uno de los resultados interesantes en el que los deltas de las poblaciones se compararon con numerosos valores independientes se pueden observar en la Ilustración 4. Se determina que el sentimiento negativo ajustado tuvo, en efecto, un efecto negativo en el tamaño del delta de las acciones, con un βNegativo ajustado=-0,082. La lógica que subyace al sentimiento positivo se fiel a la teoría, ya que el modelo arrojó un βPositivo ajustado=0,017. Neutral ajustado tuvo un efecto positivo en el precio de una acción con βNeutral ajustado=0,011. Las noticias neutrales tuvieron un efecto ligeramente positivo

en la empresa, ya que los inversores potencialmente pensaban "Ninguna noticia es una buena noticia". Todas las cifras anteriores tenían valores P<0,01 lo que significa que todas son estadísticamente significativas para el modelo. Esto significa que la tendencia de las acciones como respuesta a cualquier noticia (o a ninguna) va a ser siempre creciente, mientras que la variación debida a una noticia negativa va a tener una influencia mucho mayor (0.082 >> 0.017). Por último, estudiando la variable rezagada en el estudio, el principal componente del poder predictivo del modelo, que procede de la rentabilidad no ajustada al mercado del día anterior. Con βVariable Retrasada=17,8. Esto significa que a medida que aumenta el precio de cierre del día anterior, el precio de cierre del día siguiente tiende a aumentar también, lo que supone que la tendencia bursátil va a ser siempre positiva y creciente. Como conclusión, la tendencia normal de los valores bursátiles va a ser alcista, pero la aparición de una noticia de carácter negativo va a suponer una variación negativa mucho más pronunciada.

OLS Regression Results

Dep. Variable:	Delta	R-squared:	0.221
Model:	OLS	Adj. R-squared:	0.220
Method:	Least Squares	F-statistic:	8.124e+04
Date:	Sat, 02 Feb 2019	Prob (F-statistic):	0.00
Time:	13:02:47	Log-Likelihood:	-4.8837e+06
No. Observations:	3733316	AIC:	9.767e+06
Df Residuals:	3733303	BIC:	9.768e+06
Df Model:	13		
Covariance Type:	nonrobust		

Ilustración 5: Resultados estudio regresión múltiple con variable dependiente el delta. Fuente (Bellintani, s.f.).

El coeficiente de determinación (R^2) puede adquirir resultados que oscilan entre 0 y 1. Así, cuando adquiere resultados más cercanos a 1, mayor resultará el ajuste del modelo a la variable que se pretende aplicar para el caso en concreto. Por el contrario, cuando adquiere resultados que se acercan al valor 0, menor será el ajuste del modelo a la variable que se pretende aplicar y, justo por eso, resultará dicho modelo menos fiable. Fijándose en la Ilustración 5, los modelos dan un valor de R^2=R_adj^2=0.22, por lo que se podría decir que este modelo no explica los cambios de valor por las noticias, ya sea por la forma en que se agruparon las noticias o porque realmente no tienen valor real al final del día.

OLS Regression Results

Dep. Variable:	10 Day Market Return	R-squared:	0.004
Model:	OLS	Adj. R-squared:	0.004
Method:	Least Squares	F-statistic:	1358.
Date:	Fri, 08 Feb 2019	Prob (F-statistic):	0.00
Time:	13:46:38	Log-Likelihood:	-8.7345e+06
No. Observations:	3733316	AIC:	1.747e+07
Df Residuals:	3733304	BIC:	1.747e+07
Df Model:	12		
Covariance Type:	nonrobust		

Ilustración 6: Resultados estudio regresión múltiple con variable dependiente el rendimiento de mercado a 10 días. Fuente (Bellintani, s.f.).

Por último, si nos fijamos en el último estudio que se realizó en el artículo, correspondiente al rendimiento del mercado a 10 días, expuesto en la Ilustración 6, se puede ver como el coeficiente de determinación es R^2= 0.004, siendo incluso mucho menor que en el caso anterior. Estos resultados hacen al autor afirmar que las noticias no tienen un efecto en la volatilidad de los valores bursátiles en un rango de 10 días, una afirmación interesante a tener en cuenta en nuestro proyecto.

2.2.3 ESTADO DEL ARTE DEL MODELO DE RED PROFUNDA BIDIRECCIONAL DE MEMORIA A LARGO PLAZO[42]

La red profunda bidireccional de memoria a largo plazo (BLSTM)[43] es un tipo de arquitectura de red neuronal que se engloba dentro de las redes neuronales recurrentes[44] (RNN). Está diseñada para resolver las limitaciones de las RNN tradicionales a la hora de capturar dependencias a largo plazo en datos secuenciales. Las BLSTM se utilizan ampliamente en diversas aplicaciones, como el procesamiento del lenguaje natural, el reconocimiento del habla y el análisis de series temporales, en las que los datos de entrada tienen dependencias temporales o secuenciales.

Los conceptos clave de una memoria profunda bidireccional a corto plazo son:

- **Redes neuronales recurrentes (RNN):** Las RNN son una clase de redes neuronales artificiales diseñadas específicamente para procesar datos secuenciales introduciendo el concepto de estados ocultos o celdas de memoria. Cada neurona de una RNN tiene un "estado oculto", que actúa como memoria, permitiendo a la red mantener información de pasos temporales anteriores y utilizarla para procesar entradas posteriores en la secuencia.

[42] Del inglés Deep bidirectional long short-term memory (DBLSTM). Ver glosario.
[43] Early, 2022, Yinghao Ren, 2020, Jie Wu, 2017.
[44] Del inglés Recurrent neural networks (RNN). Ver glosario.

- **Memoria de largo y corto plazo (LSTM)[45]:** La LSTM es un tipo especializado de RNN que supera los problemas de gradiente explosivo y evanescente que suelen presentar las RNN tradicionales. La LSTM introduce tres tipos de compuertas: compuerta de entrada, compuerta de olvido y compuerta de salida, que le permiten controlar el flujo de información que entra y sale de la célula de memoria. Esto permite a las LSTM aprender y recordar eficazmente dependencias a largo plazo en datos secuenciales.

- **Bidireccional:** El aspecto bidireccional de las BLSTM es lo que las diferencia de las LSTM normales. En una LSTM estándar, la información fluye sólo en una dirección, del pasado al futuro. En cambio, una BLSTM procesa la secuencia de entrada en ambas direcciones simultáneamente, del pasado al futuro y del futuro al pasado. Este procesamiento bidireccional permite a la red captar información tanto del contexto pasado como del futuro, mejorando su capacidad para comprender las dependencias y el contexto de los datos.

- **Aprendizaje profundo:** El término "profundo" en "BLSTM profundo" se refiere al concepto de apilar múltiples capas de BLSTM unas sobre otras. Los modelos de aprendizaje profundo con múltiples capas tienen la ventaja de aprender representaciones más complejas a partir de los datos, lo que puede mejorar el rendimiento en diversas tareas.

[45] Del inglés long-short term memory (LSTM). Ver glosario.

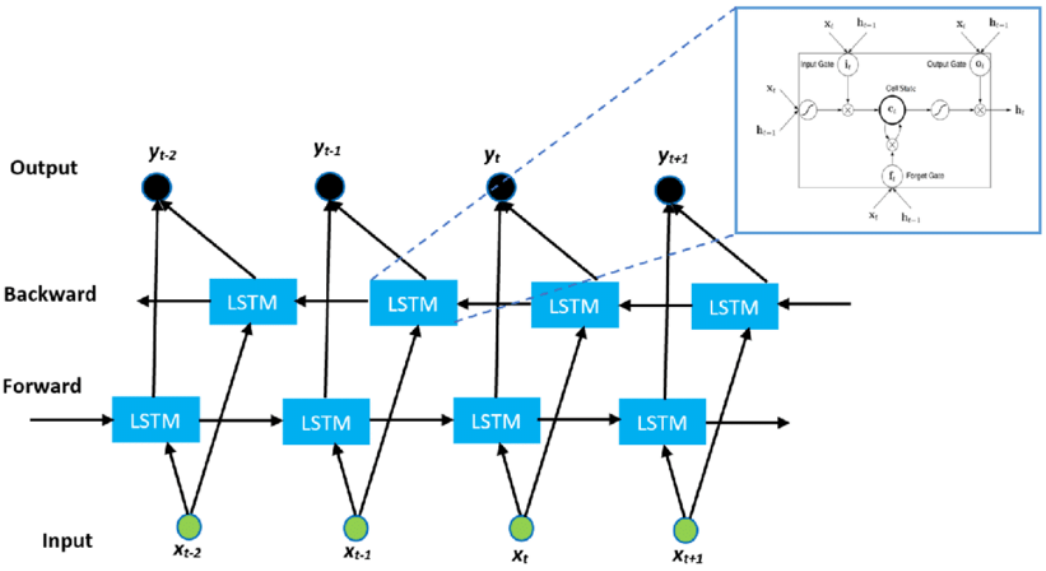

Ilustración 7: Esquema de una red profunda bidireccional de memoria a largo plazo. Fuente Researchgate.net

En la Ilustración 7 se puede observar un esquema de una red bidireccional de memoria a largo plazo. Esta ilustración muestra solo una capa de la red, por lo que la red profunda completa estaría formada por distintas capas como esta. El flujo de trabajo de una red profunda bidireccional de memoria a largo plazo, como se puede observar en la Ilustración 7 es el siguiente:

1. Los datos de entrada se presentan como una secuencia secuencial o de series temporales.
2. Los datos se introducen en la primera capa de la red BLSTM, que procesa la secuencia tanto hacia delante como hacia atrás.
3. La salida de la primera capa pasa a las capas siguientes, formando una arquitectura profunda.
4. Cada capa refina la representación aprendida de la capa anterior, lo que permite a la red aprender patrones complejos y dependencias en los datos.

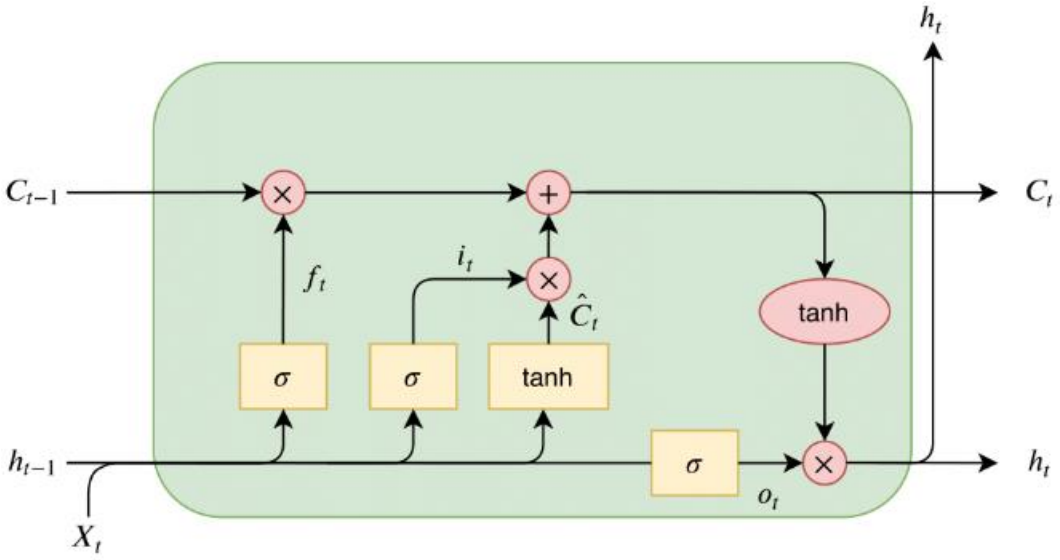

Ilustración 8: Esquema de cada célula LSTM. Fuente Researchgate.net

Arquitectura célula LSTM:

- X_t: Paso de tiempo de entrada.
- h_t: Salida.
- C_t: Estado de la célula.
- f_t: Puerta de olvido.
- i_t: Puerta de entrada.
- o_t: Puerta de salida.
- $\hat{C}_t$: Estado de la célula interna.

En la Ilustración 8 se puede observar el funcionamiento de cada célula LSTM[46]. Una célula LSTM consta de tres puertas y una célula de memoria interna. Cada puerta es un conjunto de capas neuronales que controlan el flujo de información que entra o sale de la célula de memoria. Las tres puertas de una célula LSTM son:

[46] Ingolfsson, 2021

1. **Puerta de entrada (i_t):** Esta puerta controla qué información nueva debe añadirse a la célula de memoria interna. Analiza la nueva entrada y la memoria anterior para decidir cuánta y qué información debe almacenarse.

2. **Puerta de olvido (f_t):** La puerta del olvido determina qué información debe ser descartada u olvidada de la célula de memoria interna. Evalúa la relevancia de la información almacenada en la célula de memoria anterior y decide qué cantidad de esa información debe conservarse.

3. **Puerta de salida (o_t):** La puerta de salida regula cuánta información de la celda de memoria interna debe utilizarse para la salida actual. Determina la relevancia de la información almacenada en la célula de memoria y controla cuánta de esa información debe pasarse como salida.

El funcionamiento de cada puerta en una célula LSTM es:

1. **Puerta de entrada:**
   - Se calcula una "puerta de entrada", que va de 0 a 1, y que representa qué nueva información de entrada debe añadirse a la célula de memoria.
   - La puerta de entrada se calcula mediante una función sigmoidea que procesa la combinación lineal de la nueva entrada y la salida de la célula anterior (estado oculto).

- A continuación, se calcula un vector candidato que representa la información que podría añadirse a la célula de memoria. Este vector se genera aplicando una función tangente hiperbólica (tanh) a la combinación lineal de la nueva entrada y el estado oculto anterior.
- Finalmente, la puerta de entrada se multiplica por el vector candidato para obtener la información que se añadirá a la celda de memoria.

$$i_t = \sigma(W_i * [h_{t-1}, X_t] + b_i) \quad [7]$$

2. **Puerta de olvido:**

- La puerta del olvido determina qué información de la celda de memoria anterior debe conservarse y qué información debe descartarse.
- Se calcula una "puerta de olvido", que va de 0 a 1, y que representa la cantidad de información que debe olvidarse de la celda de memoria anterior.
- La puerta del olvido se calcula mediante una función sigmoidea que procesa la combinación lineal de la nueva entrada y el estado oculto anterior.
- A continuación, la puerta del olvido se multiplica por el contenido de la celda de memoria anterior para obtener la información que se retendrá en la celda de memoria actual.

$$f_t = \sigma(W_f * [h_{t-1}, X_t] + b_f) \quad [8]$$

3. **Puerta de salida:**

- La puerta de salida controla cuánta información de la celda de memoria debe utilizarse para generar la salida actual (nuevo estado oculto).
- Se calcula una "puerta de salida", que va de 0 a 1, y que representa qué parte de la información almacenada en la celda de memoria se utilizará para la salida.
- La puerta de salida se calcula mediante una función sigmoidea que procesa la combinación lineal de la nueva entrada y el estado oculto anterior.
- A continuación, se aplica una función tangente hiperbólica (tanh) al contenido de la celda de memoria actual.
- Por último, la puerta de salida se multiplica por el contenido de la celda de memoria procesada por la función tanh para obtener la salida actual (nuevo estado oculto).

$$o_t = \sigma(W_o * [h_{t-1}, X_t] + b_o) \qquad [9]$$

Además, el estado de la célula interna se calcula como:

$$\hat{C}_t = tanh(W_c * [h_{t-1}, X_t] + b_c) \qquad [10]$$

Y por último, el estado de la célula y la salida se calculan respectivamente:

$$C_t = i_t * \hat{C}_t + f_t * C_{t-1} \qquad [11]$$

$$h_t = o_t \times \tanh(C_t) \qquad [12]$$

Al igual que en cada red neuronal, se conectan pesos y sesgos a cada puerta. Estas matrices de pesos se utilizan en combinación con la optimización basada en el gradiente para que la célula LSTM aprenda. .Las matrices de pesos y los sesgos pueden verse respectivamente en las ecuaciones anteriores como W_f, b_f, W_i, b_i, W_o, b_o, y W_c, b_c .

La dimensión de cada matriz de pesos viene determinada por:

$$C_{LSTM} * (F_d + C_{LSTM}) \quad [13]$$

- C_{LSTM} representa el número de celdas de cada capa de LSTM.
- F_d representa la dimensión de las características del valor de entrada.

La dimensión de cada matriz de sesgos es directamente la dimensión de C_{LSTM}. Por tanto, el número total de parámetros en una capa de una red LSTM es:

$$4 * (C_{LSTM} + F_d + C_{LSTM}^2 + C_{LSTM}) \quad [14]$$

La inferencia LSTM puede reducirse a dos multiplicaciones entre matrices. La primera puede simplificarse como:

$$[X_t|h_{t-1}|1] \times \begin{bmatrix} W \\ b \end{bmatrix} \quad [15]$$

W es la matriz de pesos de cada célula LSTM que se compone de W_f, W_i, W_o y W_c que se utilizan en las ecuaciones para las puertas y el estado de la célula. Como también se ha comentado justo arriba, b es la matriz de sesgo, que se compone de b_f, b_i, b_o y b_c.

La última multiplicación de matrices es la necesaria para calcular C_t y h_t. Obsérvese también que estas multiplicaciones siguientes son puntuales. Pueden reducirse a

$$C_{LSTM} * T\ MACs^{47} \qquad [16]$$

donde T es la longitud de la serie temporal.

Poniendo todo esto junto, el número total de MACs en una capa LSTM es:

$$((F_d + C_{LSTM} + 1) * 4 * C_{LSTM} + C_{LSTM}) * T \qquad [17]$$

Uno de los artículos[48] más relevantes que se han estudiado acerca de este modelo utiliza un indicador llamado BIAS que calcula la diferencia porcentual entre un índice de mercado o precio de cierre y una media móvil. De este modo se puede obtener el índice que refleja el grado de desviación entre el precio y su media móvil en un determinado período de tiempo, y la posibilidad de que el precio retroceda o repunte por desviarse de la tendencia de la media móvil cuando el precio fluctúa violentamente, así como la fiabilidad de que el precio se mueva dentro del rango normal de fluctuación para formar un potencial continuo. La fórmula de cálculo es la siguiente:

$$BIAS = \frac{P_i - P}{P_i} * 100\% \qquad [18]$$

Donde:

- P representa el valor de cierre de la acción el día de la noticia.

[47] Del inglés Multiplication and Accumulation Operations. Ver glosario.
[48] Yinghao Ren, 2020

- P_i representa la media del valor de la acción i días después de que la noticia ocurriera.

Por tanto, un valor BIAS>0 supone una tendencia alcista de esa acción y viceversa. Por último, para obtener unos resultados más exactos, convierten el problema en una clasificación binaria utilizando LSTM. Para ello, utilizan un umbral cambiante para no confundir el movimiento bursátil debido a la tendencia general con el movimiento bursátil debido explícitamente al contenido de una noticia. Así, se cogerá un umbral de subida o bajada en los i días desde la aparición de la noticia y si la variación es menor al umbral, se considerará una noticia pesimista, mientras que si la variación es mayor al umbral, se considerará una noticia optimista. Las noticias que se supongan como pesimistas arrojarán un valor de -1 y las optimistas un valor de 1.

Por último, este artículo arroja unos resultados comparando diferentes índices y diferentes modelos de predicción.

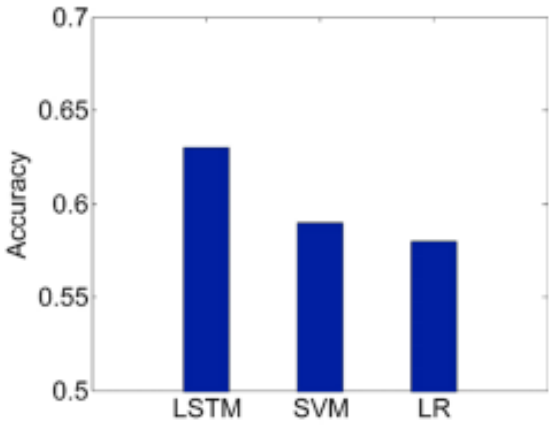

Ilustración 9: Precisión de la predicción bursátil basándose en el índice BIAS utilizando distintos modelos. Fuente (Yinghao Ren, 2020)

Como se puede apreciar en la Ilustración 9, se comparan tres modelos matemáticos: red profunda bidireccional de memoria a largo plazo (LSTM), modelo de máquinas de

vectores de soporte (SVM) y modelo de regresión múltiple (LR). Se puede apreciar que el LSTM es el que más precisión arroja, por lo que es el más adecuado para predicción de valores bursátiles.

En la Ilustración 10 se utiliza como indicador el ratio entre el precio de cierre de mercado y el precio de apertura de mercado. Si se comparan con los resultados de la Ilustración 9 se puede apreciar como la precisión es ligeramente superior utilizando el indicador BIAS, por lo que sería más conveniente utilizarlo en un modelo.

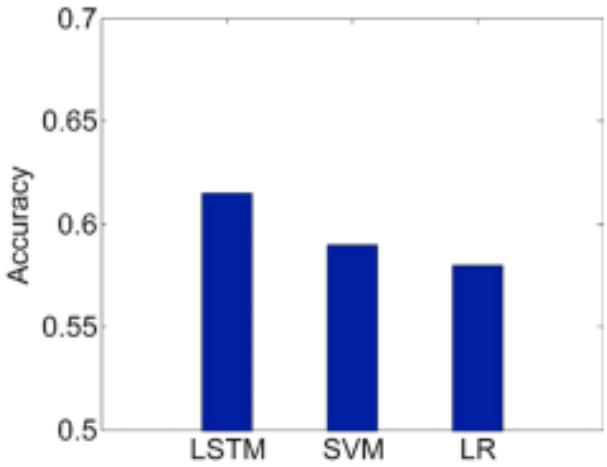

Ilustración 10: Precisión de la predicción bursátil basándose en el ratio utilizando distintos modelos. Fuente (Yinghao Ren, 2020)

En resumen, una célula LSTM utiliza las puertas de entrada, olvido y salida para controlar qué información se añade, retiene o utiliza para generar la salida en cada paso temporal. Estas puertas permiten a la LSTM aprender y retener patrones a largo plazo en secuencias de datos, lo que la hace especialmente adecuada para tareas que implican relaciones temporales o dependencias a largo plazo, como el procesamiento de textos, el análisis de series temporales y la traducción automática.

2.3 CONCLUSIONES

El estudio de la literatura relevante relacionada con el proyecto nos ha permitido estudiar los aciertos y fallos de los proyectos anteriores, para poder aprender de dichos errores y crear un modelo propio como necesidad ante la falta de un modelo completo que realice un análisis sentimental completo. Las distintas literaturas estudiadas servirán como ayuda para el desarrollo de nuestro modelo.

Capítulo 3. MODELO TEÓRICO DE EXTRACCIÓN Y CLASIFICACIÓN

Este capítulo comentará los distintos algoritmos utilizados para extraer y clasificar información relevante. Como se ha comentado en capítulos previos, un buen desarrollo de este modelo será de gran importancia a la hora de evitar sobre ajustes y mejorar la precisión del modelo de redes neuronales.

La función de este primer modelo matemático consistirá en extraer la información relevante dentro de una noticia, agrupar dicha información sin tener en cuenta género y número de las palabras para reducir la cantidad de información extraída, clasificar cada noticia en base al número de veces que aparecen las palabras relevantes y relacionarlo con la relevancia de cada palabra, creando una escala de palabras positivas y negativas y asociando valores a cada una para crear un vector único para cada noticia y por último una conversión de dicho vector para que el segundo modelo matemático lo pueda procesar. Este primer modelo matemático se va a subdividir en distintos módulos que configurarán cada función específica. Los distintos módulos son: extracción de la información, reducción de la información, clasificación de la información (I y II) y creación de un vector.

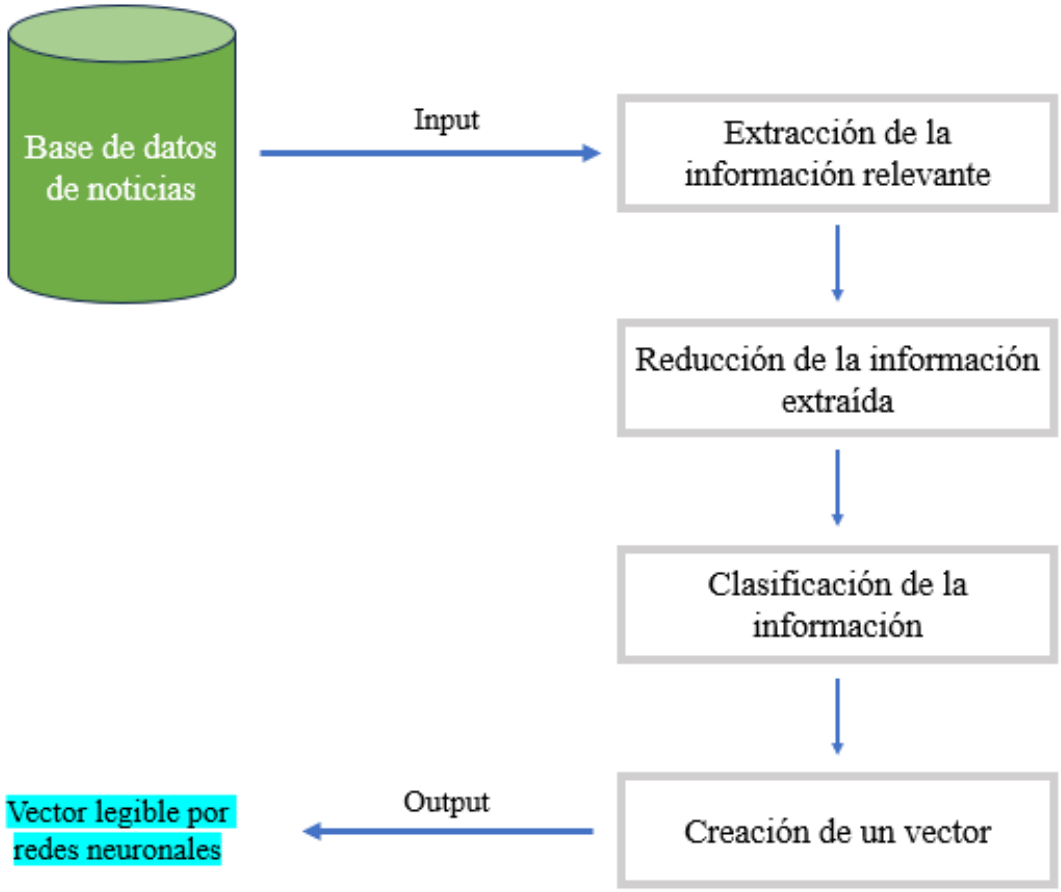

Ilustración 11: Esquema primer modelo teórico. Fuente elaboración propia, 2023.

3.1 EXTRACCIÓN DE LA INFORMACIÓN

En base a la Tabla 2, el modelo de combinación de 2-palabras[49] será el utilizado en nuestro proyecto. Como se comenta en un artículo de estudio sobre este modelo[50], el número de n palabras se restringe a dos puesto que siendo tres, el modelo arroja una infinidad de combinaciones, reduciendo su frecuencia, suponiendo el estudio de los grupos de palabras más frecuentes en el artículo menos esclarecedor en cuanto a resumir el tema de la noticia.

[49] Del inglés 2-Word combination model. Una especificación del N-Word combination model. Ver glosario.
[50] Michael Hagenau, 2013

El modelo de combinación de 2-palabras consistirá en la creación de una matriz con todos los grupos de palabras relevantes en una noticia, dicha matriz se determina en el primer paso del proyecto, el estudio de la información relevante. Una vez se ha creado esta matriz, el algoritmo recorrerá la noticia y extraerá los grupos de palabras pertenecientes a la matriz, los cuales se encuentren a una separación inferior o igual a m, número de separación entre palabras que será una variable que podrá modificarse. Por último, el algoritmo devolverá una clasificación de las parejas de palabras más utilizadas, y la frecuencia de aparición en el texto de cada una de ellas.

Classification performance for different feature types measured by R^2.

Feature type	Data I: DGAP			Data set II: EuroAdhoc		
	Freq-based feature reduction	Chi^2-based feature selection	BNS-based feature selection	Freq-based feature reduction	Chi^2-based feature selection	BNS-based feature selection
Single words I: based on dictionary	4.7%	–	–	0.1%	–	–
Single words II: retrieved from corpus	4.7%	4.7%	5.9%	0.4%	1.0%	0.4%
2-Gram[a]	1.3%	8.5%	4.4%	0.3%	2.7%	2.5%
2-Word combinations	4.9%	15.3%	20.2%	0.9%	7.4%	9.4%
Noun phrases	3.8%	6.2%	4.6%	0.4%	0.3%	2.1%

[a] Performance of 3-Gram was slightly weaker than 2-Gram and is therefore not listed. 3-Gram suffer from a high number of combinations causing a rapid decrease in actual frequencies per feature.

Tabla 2: Clasificación de rendimiento de los distintos modelos por coeficiente de determinación

Fuente. (Michael Hagenau, 2013).

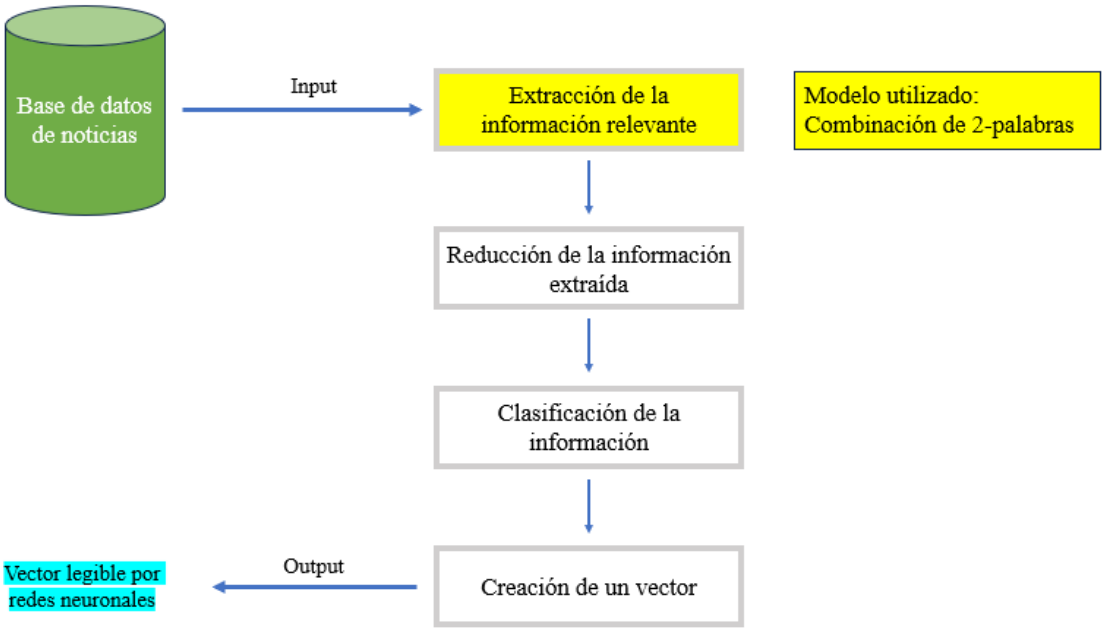

Ilustración 12: Esquema primer modelo teórico con la primera etapa detallada. Fuente elaboración propia,, 2023.

3.2 REDUCCIÓN DE LA INFORMACIÓN

Este segundo paso reducirá información redundante, agrupando parejas de palabras que varíen en género o número, extraídas en el primer paso de extracción de la información, para poder realizar una clasificación de los grupos de palabras basado en la frecuencia de dichas parejas. Para esta etapa, se utilizará el modelo Porter Stemmer que se ha comentado en el estado del arte. En esta etapa se cogerán los términos de la clasificación de los grupos de palabras generados en la primera etapa y se agruparán aquellos con la misma raíz, modificando la lista de las palabras más frecuentes, junto a su frecuencia y las combinaciones de palabras con las que han aparecido. Así, se podrá entender de manera ordenada los temas más relevantes sobre los que trata dicha noticia.

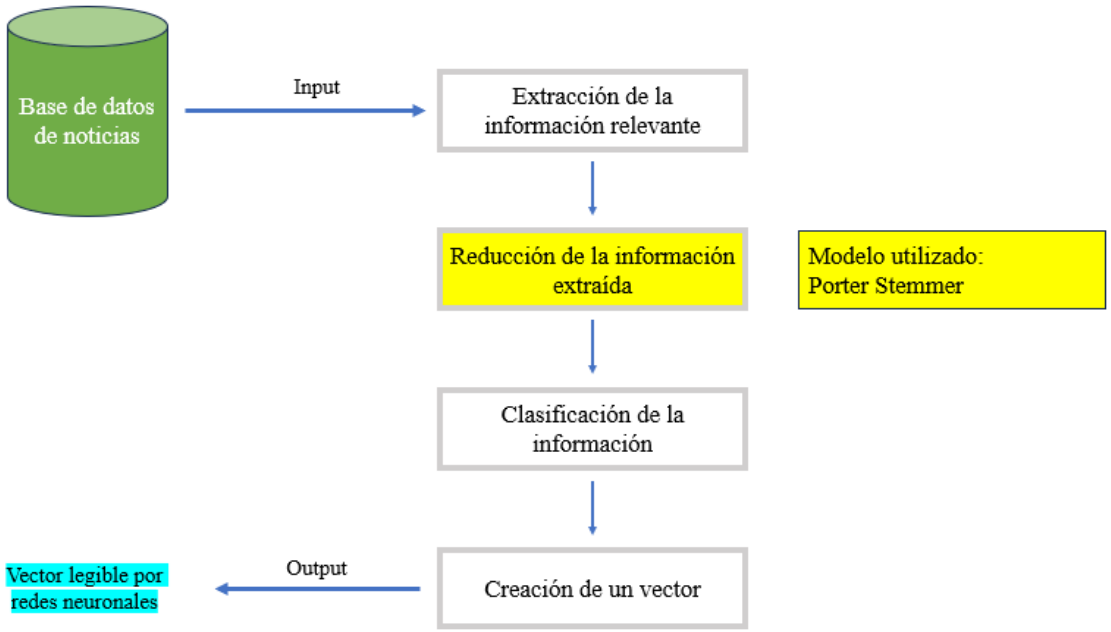

Ilustración 13: Esquema primer modelo teórico con la segunda etapa detallada. Fuente elaboración propia, 2023.

3.3 CLASIFICACIÓN DE LA INFORMACIÓN I

En esta etapa se clasificará la información extraída, asignando unos valores de relevancia a cada palabra o parejas de palabras, para poder resaltar de las palabras relevantes cuáles son las que hay que tener más en cuenta en cada artículo específico. A la hora de la clasificación, hay tres distintos modelos que se pueden desarrollar para dicha clasificación. Tras haber explicado en el estado del arte los tres modelos y debido a su completitud, el modelo que se va a utilizar para este proyecto es el de selección de características utilizando retroalimentación del mercado exógeno. Este modelo es el más completo pues incluye retroalimentación del mercado al coger valores bursátiles de cada empresa, por lo que será el que se utilizará.

Como se ha comentado anteriormente, la diferencia de este método con el que no incluye retroalimentación del mercado es la incorporación de un método estadístico. Los dos métodos estadísticos más válidos para este caso son el Chi-cuadrado y la separación bi-normal.

1. **Chi-cuadrado[51]:** Chi-cuadrado es un modelo estadístico que se basa en la siguiente ecuación:

$$\chi^2 = \sum_{j=1}^{4} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \qquad [13]$$

El modelo compara la frecuencia observada Oi de la característica i dentro del conjunto de mensajes positivos con su frecuencia esperada Ei, y normaliza la desviación al cuadrado. Esta desviación se calculará para los cuatro posibles resultados j, es decir, característica en mensaje positivo/negativo y característica no en mensaje positivo/negativo. La suma de las cuatro desviaciones normalizadas constituye el estadístico χ^2. Utilizando este estadístico χ^2 cada característica recibe un valor por mayor o menor desviación respecto a lo esperado. Las palabras que suelen influir en las decisiones de los inversores reciben valores más altos. Las palabras que influyen menos en la decisión, ya que aparecen uniformemente en los mensajes positivos y negativos, recibirán valores más bajos. Para evaluar si una palabra tiene mayor o menor poder explicativo, calculamos el p-valor basándonos en la prueba Chi-cuadrado. Cortamos la lista de

[51] Del inglés Chi-square. Ver glosario.

características con un valor p del 5%, es decir, obtenemos una lista de características con un nivel de confianza de al menos el 95% de que el inversor medio basa su decisión de inversión también en estas características.

2. **Separación bi-normal[52]:** Este segundo modelo estadístico mide la separación entre la prevalencia de las características en la clase de mensajes positivos y la clase de mensajes negativos. Curiosamente, se utiliza poco en la literatura, pero ofrece resultados superiores para un mayor número de características. La ecuación de BNS se define como:

$$BNS = F^{-1}\left(\frac{O_{i,pos}}{pos}\right) - F^{-1}\left(\frac{O_{i,neg}}{neg}\right) \qquad [14]$$

donde F^{-1} es la función de probabilidad acumulativa inversa de la distribución normal (es decir, la puntuación z) y siendo *pos* el número de mensajes positivos y *neg* el número de mensajes negativos. En correspondencia con Chi-cuadrado, Oi,pos (Oi,neg) denota la característica de frecuencia observada i dentro del conjunto de mensajes positivos (negativos). Para evitar el valor indefinido $F^{-1}(0)$, el cero se sustituye por un número muy pequeño, es decir, 0,0005. La principal diferencia estructural del BNS es que sólo se centra en las palabras que aparecen realmente en un documento, a diferencia del Chi-cuadrado, que

[52] Del inglés Bi-normal Separation (BNS). Ver glosario.

también incluye el recuento de características que no aparecen en un mensaje.

Un artículo[53] ya realizó un estudio sobre el rendimiento de ambos modelos estadísticos, arrojado en Tabla 2**Error! Reference source not found.**, donde se puede observar como este último modelo estadístico, el BNS, tiene un mayor rendimiento en la clasificación de puntuaciones de sentimiento en palabras extraídas de un texto. Será por tanto este modelo estadístico el que se utilice en nuestro proyecto.

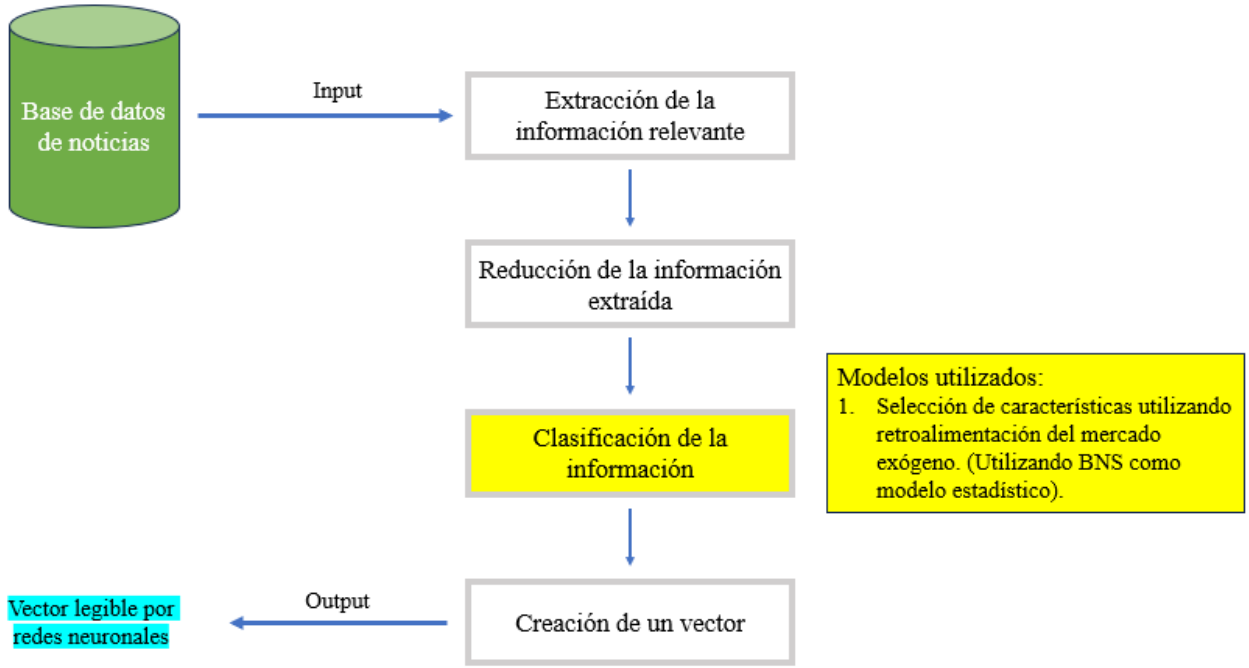

Ilustración 14: Esquema primer modelo teórico con la tercera etapa detallada I. Fuente elaboración propia, 2023.

[53] Michael Hagenau, 2013

Como se ha revisado en el estado del arte, se va a introducir un algoritmo de clasificación que es compatible con el modelo anterior de selección de características utilizando retroalimentación del mercado exógeno. Dicho modelo, el de frecuencia de término-frecuencia inversa de documento aportará una mayor calidad de clasificación a la clasificación del modelo de retroalimentación. El modelo teórico que se va a utilizar puede esquematizarse como:

1. **Cálculo de la TF (frecuencia de términos):** Calcular la frecuencia de términos de cada palabra del documento. Este recuento de la frecuencia de términos ya se ha realizado en el paso de extracción de la información, por lo que solo habrá que extraerlo de la base de datos previa de las palabras relevantes.

2. **Calcular la IDF (frecuencia inversa del documento):** Calcule la frecuencia inversa del documento para cada palabra. Se cogerá la base de datos extraída en el primer paso del modelo matemático, la cual contiene dividido por noticias, todas las palabras relevantes de cada una de ellas.

3. **Calcule el TF-IDF:** Multiplicar el valor TF de cada palabra por su valor IDF. El TF-IDF asigna un peso a cada palabra en función de su frecuencia en el documento y su rareza en todos los documentos. Las palabras con puntuaciones TF-IDF más altas se consideran más relevantes para el documento.

4. **Incorporar palabras relevantes y puntuaciones sentimentales:** Integrar la lista de palabras relevantes obtenida a partir del feature selection employing exogenous feedback (clasificación de la información I) junto los valores de TF-IDF obtenidos.

5. **Determinación de la relevancia:** Comparar las puntuaciones TF-IDF de las palabras relevantes con sus puntuaciones de sentimiento. Este paso evalúa la importancia y el sentimiento de las palabras relevantes dentro del documento. Las puntuaciones TF-IDF más altas combinadas con puntuaciones sentimentales positivas sugieren una mayor relevancia y un sentimiento positivo, mientras que las puntuaciones TF-IDF más altas combinadas con puntuaciones sentimentales negativas sugieren una mayor relevancia y un sentimiento negativo.

6. **Análisis e interpretación:** Analizar la relevancia de las palabras en función de sus puntuaciones TF-IDF y sentimentales. Identifique las palabras más relevantes que contribuyen significativamente al sentimiento del documento. Tener en cuenta el contexto del documento y el ámbito específico de los mercados financieros para interpretar los resultados con precisión.

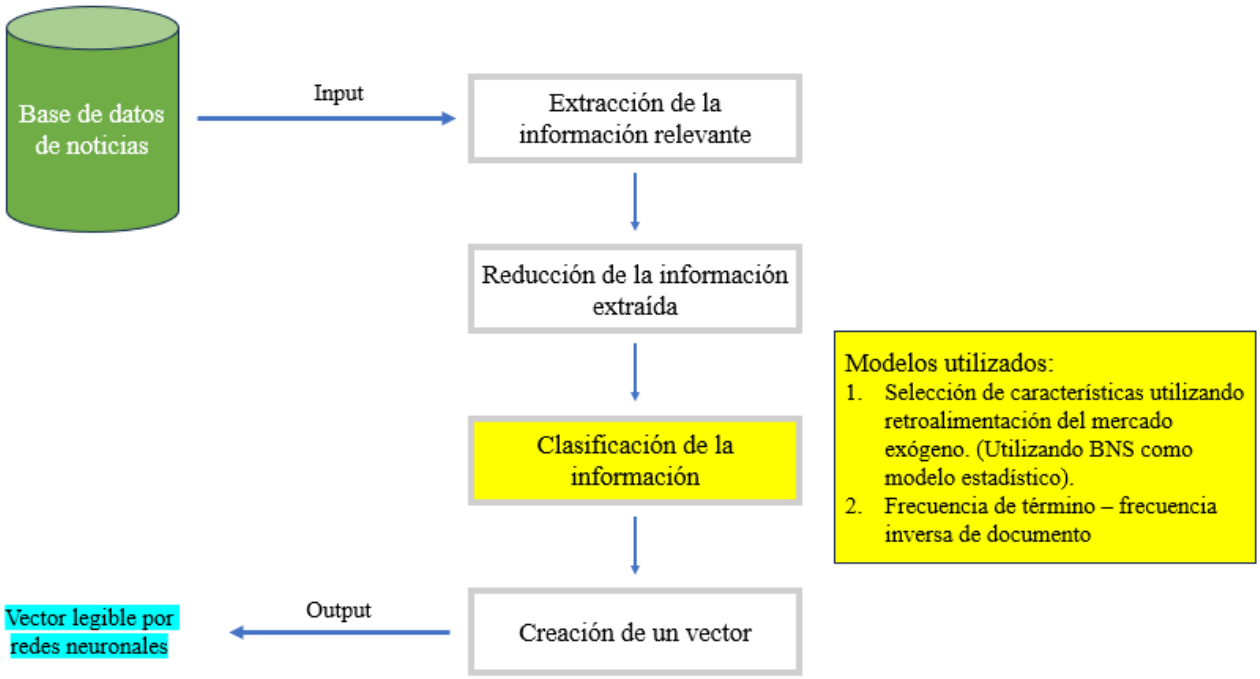

Ilustración 15: Esquema primer modelo teórico con la tercera etapa detallada II. Fuente elaboración propia, 2023.

3.4 CREACIÓN DE UN VECTOR

El último paso de este primer modelo consiste en la creación de un vector asociado a la información de la noticia previamente extraída y tratada, para que dicho vector sea legible por el modelo de redes neuronales perteneciente al siguiente paso del proyecto.

Para la creación de este vector se van a exponer tres diferentes métodos: el método de clasificación bayesiana, y dos métodos algorítmicos de aprendizaje supervisado (machine learning), muy parecidos entre sí, que son el modelo de máquinas de vectores de soporte (SVM) y el modelo de regresión de vectores de soporte (SVR).

El primer modelo de clasificación Naive Bayes, para su implementación en este proyecto, sería necesario convertir cada instancia de los datos extraídos y clasificados previamente en un vector de características adecuado para la clasificación Naive Bayes. En este caso, puede representar la presencia o ausencia de palabras o frases relevantes como características binarias. Por ejemplo, si una palabra aparece en el artículo, el valor de la característica correspondiente es 1; de lo contrario, es 0. Esto crea una matriz de características dispersa. Sin embargo, al no ser un algoritmo de aprendizaje supervisado como sí lo son los otros dos, este modelo de clasificación bayesiana tiene una mayor limitación en este proyecto con respecto a los otros dos modelos, al tener una menor robustez por el hecho de ser un algoritmo probabilístico, es por eso por lo que este método no será el utilizado.

Comparando ahora los dos modelos de aprendizaje supervisado, y como se ha analizado en el estado del arte, el SVR es un derivado de SVM utilizado para datos continuos en vez de clases, por lo que es idóneo para nuestro proyecto de predicción de valores bursátiles. Por tanto, el modelo de regresión de vectores de soporte (SVR) será el utilizado en nuestro proyecto.

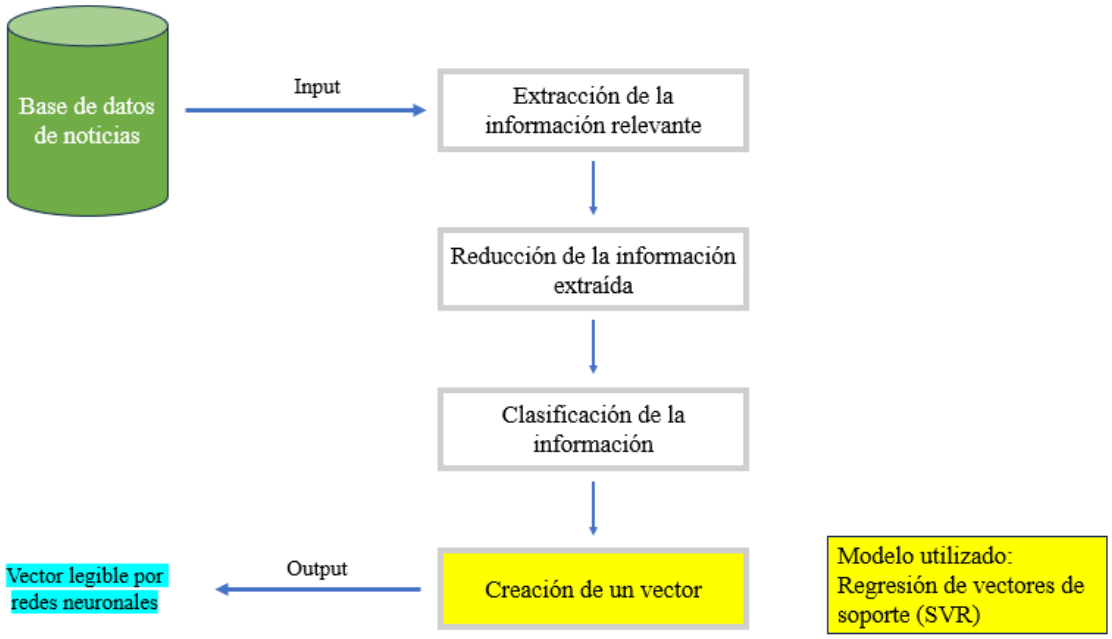

Ilustración 16: Esquema primer modelo teórico con la última etapa detallada. Fuente elaboración propia, 2023.

3.5 CONCLUSIONES

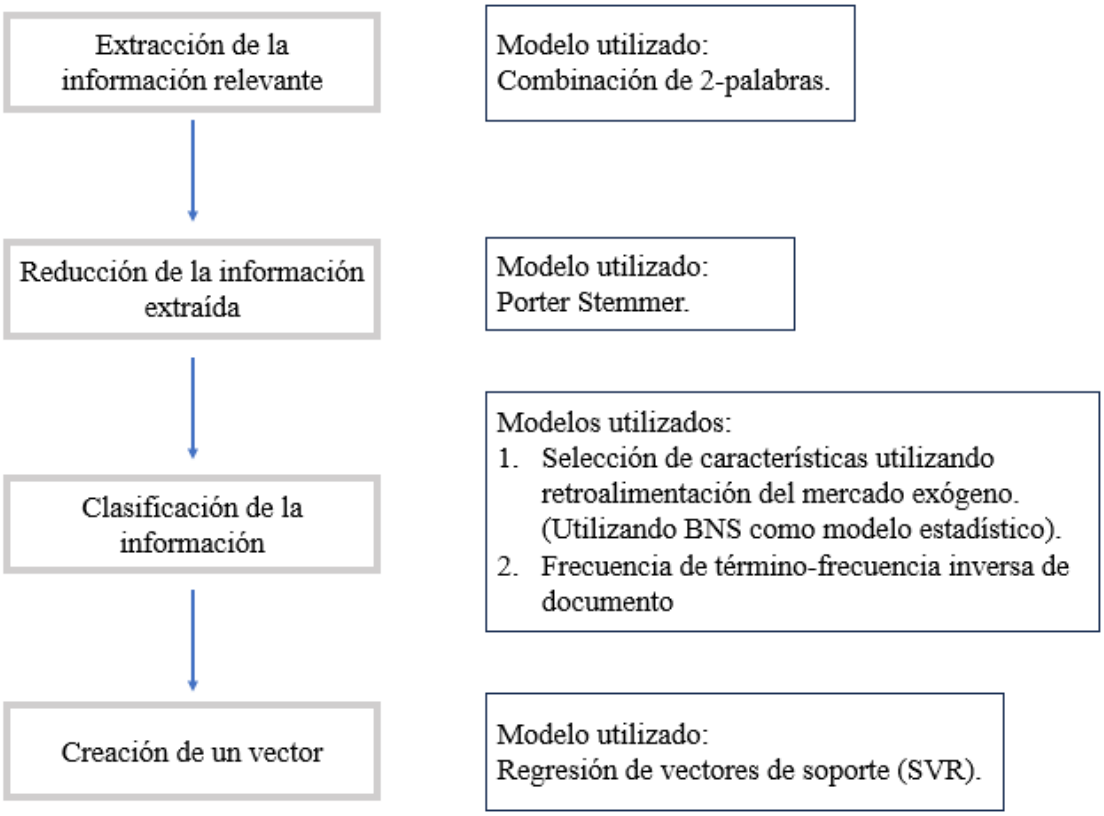

Ilustración 17: Esquema primer modelo teórico detallado. Fuente elaboración propia, 2023.

En resumen, el primer modelo teórico lo que hará será limpiar cada noticia para extraer solo la información importante. Posteriormente, hará una clasificación de dicha información importante y por último la convertirá a un vector legible por el segundo modelo matemático.

Como se puede apreciar en la Ilustración 17, los modelos que se utilizarán en cada paso quedan detallados, habiendo sido explicados cada uno de ellos en su respectiva sección.

Capítulo 4. MODELO TEÓRICO PREDICCIÓN Y VARIACIÓN DE VALORES BURSÁTILES

Este capítulo indagará en los distintos algoritmos plausibles para desarrollar la principal parte de nuestro proyecto, la relacionada con las redes neuronales. Utilizando la ayuda de la literatura relevante, se elaborará el modelo que mejor se ajuste a las necesidades del proyecto, el cual consiga reducir en mayor medida la incertidumbre asociada a los valores de salida.

Este segundo modelo será el encargado de aportar la solución final del proyecto. El modelo recibirá como entradas el vector creado en el primer modelo matemático, junto con información histórica de valores bursátiles de inicio y cierre de distintas empresas, los cuales coge de una base de datos. Este modelo, teniendo la información de la noticia extraída por el primer modelo, y junto con la base de datos, arrojará un resultado de predicción de variación del valor bursátil de una empresa en bolsa, como consecuencia de la reacción de la sociedad ante las distintas noticias. Se utilizarán datos para entrenar el modelo y poder reducir la incertidumbre de variación bursátil, arrojando resultados con una incertidumbre menor que el 50%, la cual sería la incertidumbre estándar de variación de una acción bursátil sin haber realizado ningún modelo (la probabilidad de que la cotización de una empresa sube o baje en un día es a priori y sin estudiar ninguna variable, un 50%).

Como se ha visto en el estado del arte, hay distintos algoritmos que pueden aportar un valor a este modelo. Se estudiarán y se verán su capacidad de aplicación en este proyecto, para poder en última instancia elegir el modelo más adecuado con el que se llevará a cabo el proyecto.

4.1 MODELO DE MOVIMIENTO BROWNIANO[54][55]

Como se ha comentado en el estado del arte, el modelo de movimiento browniano puede emplearse utilizarse en los modelos financieros para estimar el valor futuro de una acción.

Es importante tener en cuenta que el modelo de movimiento browniano es una simplificación de la dinámica de los precios de las acciones en el mundo real y asume una deriva y una volatilidad constantes, que pueden no cumplirse en la práctica.

Por tanto, este modelo de movimiento browniano es una simplificación estadística del algoritmo que se busca en este proyecto, al no tener más que dos variables ajustables para conseguir resultados específicos para cada caso. Como se comenta en el capítulo de literatura relevante, este modelo se basa en la hipótesis de distribución normal de los precios de las acciones debido a su deriva y volatilidad, por lo que es un modelo que no alcanza una gran eficacia en predicciones debidas a sucesos extraños, como puede ser la aparición de una noticia que suponga un cambio drástico, parte del objetivo de este proyecto.

Como se ha comentado en los resultados de la literatura relevante, este modelo estadístico apenas reduce la incertidumbre estándar que se supone (50%), debido a su poca capacidad

[54] Del inglés Brownian motion model. Ver glosario.
[55] Rajpal, 2018

de perfeccionamiento de las variables. Es por eso y por la poca capacidad de variabilidad del modelo ante cambios drásticos por lo que este modelo se desestimará para el proyecto, pues se busca tener unos resultados superiores a este 53%.

4.2 MODELO DE REGRESIÓN MÚLTIPLE[56]

Este modelo, comentado en la literatura relevante y estudiado sus resultados, arroja unas conclusiones de gran importancia para el proyecto. Estas, en las cuales se ha indagado en el estado del arte se podrían resumir en:

- El mercado bursátil tiene una tendencia normal alcista, lo que supone que ante el caso de ninguna anomalía, el valor de cierre de una acción debería incrementar ligeramente de un día a otro.
- El sentimiento de las noticias es un factor a tener en cuenta, sacando dos principales conclusiones. La primera, la falta de noticias relevantes o la aparición de noticias de sentimiento neutro (ni positivo ni negativo), van a suponer un crecimiento de los valores bursátiles. La segunda, la aparición de una noticia de sentimiento negativo va a derivar en una variación negativa mayor que en el caso contrario con una noticia de sentimiento positivo.
- Las noticias tienen una relevancia en los valores bursátiles a corto plazo, afirmando que no tienen influencia en dichos valores a los 10 días y en adelante.

[56] Bellintani, s.f.

4.3 MODELO DE RED PROFUNDA BIDIRECCIONAL DE MEMORIA A LARGO PLAZO[57]

Como se ha estudiado en la literatura relevante, este modelo es el más adecuado para nuestro proyecto, al arrojar unos datos de precisión mayores que los otros modelos que se han estudiado en este trabajo. El hecho de que sea una red neuronal permite al modelo tener un aprendizaje que será necesario para obtener una mayor precisión en nuestro modelo.

Así, el indicador BIAS explicado en la literatura relevante será muy útil al poder ver la influencia de una noticia en un periodo de tiempo determinado, eliminando el problema de fluctuaciones relacionadas con la noticia pero con una duración muy pequeña como para que sea relevante. Este problema, el cual se ha comentado en modelos anteriores como algo a estudiar, se consigue eliminar en esta red neuronal.

[57] Early, 2022, Yinghao Ren, 2020, Jie Wu, 2017.

4.4 CONCLUSIONES

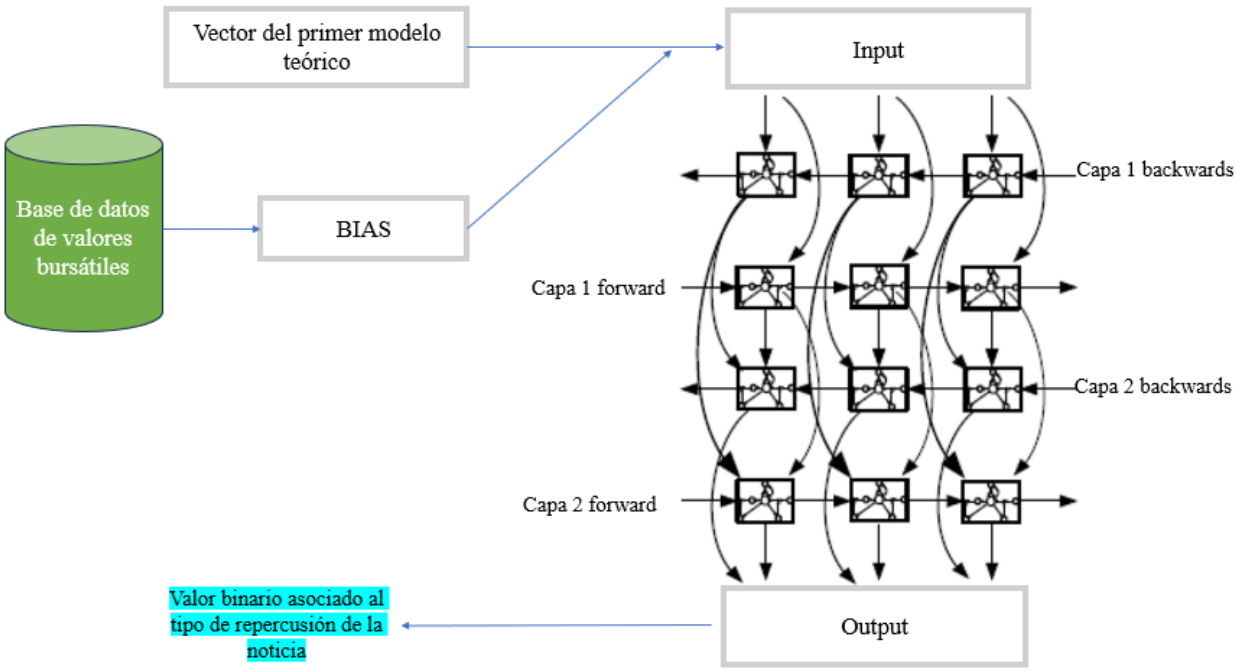

Ilustración 18: Esquema segundo modelo teórico detallado. Fuente elaboración propia, 2023..

Este segundo modelo matemático será el encargado de dar la solución final. Después de haber evaluado distintos modelos y basándose en estudios ya realizados, los cuales se han visto en la literatura relevante, el modelo que se utilizará será una red neuronal profunda bidireccional de memoria a largo plazo. Esta red, como se puede observar en el esquema de la Ilustración 18, tendrá como entrada un vector relacionado con la información relevante de la noticia, proveniente del primer modelo teórico; y una base de datos de valores bursátiles de apertura y cierre. Como indicador utilizará una variación porcentual entre el valor bursátil de una empresa el día de la noticia y la media de su valor durante los i días siguientes, dicho indicador se ha comentado en este proyecto como BIAS. El

resultado arrojará un valor binario que será 1 para noticias positivas, 0 para noticias sin impacto y -1 para noticias negativas.

Capítulo 5. CASO PRÁCTICO

En este capítulo se presentará el caso práctico basado en el modelo teórico, y se estudiarán sus resultados. El fin de este capítulo es plasmar en un caso práctico el modelo teórico desarrollado previamente, para observar la validez del modelo creado y por tanto la de este proyecto. El programa se codificará en lenguaje Python, compilando y ejecutando en Visual Studio Code.

5.1 BASES DE DATOS UTILIZADAS

Para el desarrollo de este caso práctico, el programa se va a fijar en una de las principales empresas españolas que cotizan en el IBEX 35, ENDESA. Todos los valores que se obtengan tendrán relación directa con esta empresa, para enmarcar el caso práctico en un sector específico.

Se utilizarán bases de datos para el entrenamiento del modelo de redes neuronales y para su posterior validación (test).

La primera base de datos será de noticias. Se cogerán un número de noticias relacionadas con Endesa y se le asignará un número de noticia a cada uno (id), para poder relacionar cada noticia con sus pertinentes datos bursátiles. Dichas noticias están sacadas directamente de los periódicos y contienen el cuerpo completo de la noticia. Habrá una base de datos de noticias de entrenamiento del modelo (news_training_dataset) y otra base de datos para la validación (news_validation_dataset). Al ser una base de datos creada manualmente, el número de noticias no va a ser muy elevado, utilizando solo diez noticias para el entrenamiento y tres noticias para la validación. Este programa permitirá

buscar la palabra o grupo de palabras relevantes que se consideren pertinentes, lo que dará resultados muy diferentes. Así, con unas base de datos mucho más amplia y una matriz de palabras relevantes con gran conexión entre las palabras, se conseguirán unos resultados óptimos. Para este caso práctico se han utilizado dos palabras, relacionadas con Endesa: Inversión y renovables. Estas dos palabras serán sobre las que se asentará la clasificación de la tokenización del cuerpo de las noticias, del cual se hablará en el siguiente apartado.

La segunda base de datos de noticias utilizará el BIAS, fórmula que se comentó en la literatura relevante, y base de datos que se utilizará como indicador de la red neuronal. Como se puede observar en la **Error! Reference source not found.**, el BIAS coge el valor bursátil de Endesa de cierre el día de la noticia y la media de los valores bursátiles durante los siguientes 4 días. Así, el valor BIAS será un porcentaje de crecimiento o decrecimiento del valor de la acción. Este porcentaje se sumará al crecimiento general del IBEX 35 durante ese mismo periodo de tiempo, resultando en un valor total de subida o bajada (crecimiento) del precio de la acción bursátil, el cual será el valor que se incluya en la base de datos que se utilizará como indicador del modelo. Al igual que en la anterior base de datos, cada BIAS irá asociado a un id, y habrá dos bases de datos, una para entrenamiento y otra para validación.

ENDESA					
	Noticias training				
	valor cierre día de la noticia:	valor medio 4 dias	BIAS	Variación IBEX 35	Crecimiento
27/08/2023	19,13	19,2425	0,58%	-1,16%	1,74%
26/07/2023	19,515	19,57125	0,29%	0,43%	-0,14%
09/07/2023	19,29	19,51	1,13%	2,38%	-1,25%
07/07/2023	19,43	19,4425	0,06%	-0,16%	0,23%
04/07/2023	19,86	19,59375	-1,36%	-3,67%	2,31%
04/07/2023	19,86	19,59375	-1,36%	-3,67%	2,31%
22/06/2023	19,402	19,506	0,53%	0,30%	0,23%
11/05/2023	18,416	18,4902	0,40%	0,09%	0,31%
09/05/2023	18,494	18,441	-0,29%	0,55%	-0,84%
28/04/2023	18,865	18,82775	-0,20%	-2,18%	1,98%
	noticias validation				
	valor cierre día de la noticia:	valor medio 4 dias	BIAS	Variación IBEX 35	Crecimiento
18/07/2023	19,345	19,5375	0,99%	1,21%	-0,22%
28/06/2023	19,8648	19,74745	-0,59%	1,70%	-2,29%
22/06/2023	19,402	19,506	0,53%	0,30%	0,23%

Tabla 3: Bases de datos BIAS entrenamiento y validación. Fuente elaboración propia, 2023.

5.2 TOKENIZACIÓN[58] DEL CUERPO DE LAS NOTICIAS

El primer paso, perteneciente al primer modelo teórico, se basará en un modelo de red neuronal para aprender asociaciones de palabras a partir de un gran corpus de texto. El modelo Word2vec[59] es una técnica de inteligencia artificial que permite el análisis algorítmico de textos mediante la conversión de palabras en vectores numéricos. Este principio básico se denomina incrustación de palabras y es un medio probado de poner el texto en una forma matemáticamente detectable. Word2vec es un modelo creado por Google en 2013 y se utiliza para producir word embeddings[60]. Estos modelos son poco

[58] La tokenización es el proceso de dividir un texto en unidades más pequeñas llamadas "tokens".
[59] Del inglés Word2vec model. Ver glosario.
[60] Del español incrustaciones de palabras o vectores de palabras.

profundos, se entrenan redes neuronales de dos capas para reconstruir contextos lingüísticos de palabras. Word2vec toma como entrada un gran corpus de texto y produce un espacio vectorial, típicamente de varios cientos de dimensiones, asignando cada palabra única en el corpus a un vector correspondiente en el espacio. Los word embeddings están colocados en el espacio vectorial de forma que las palabras que comparten contextos comunes en el corpus están localizadas cerca unas de otras en el espacio.

Este modelo Word2vec está incluido en la librería Gensim[61], por lo que se inicializa únicamente llamándolo y añadiendo 3 apartados de información:

- Window: Este parámetro establece la ventana máxima para la predicción de palabras objetivo en función de las palabras de contexto.
- Min_count: Este parámetro establece el número mínimo de veces que una palabra debe aparecer en el corpus de entrenamiento para ser considerada en el modelo.
- Workers: Este parámetro indica cuántos núcleos de CPU se utilizarán para entrenar el modelo en paralelo.

Una vez se ha inicializado el modelo, se coge el corpus de la noticia, se tokeniza el texto y se crean los vectores de palabras (word embeddings) para cada token. Por último, este modelo se entrena y se guarda en una dirección o ruta. La lista de vectores de palabras se convierte en un NumPy array[62] para poder ser tratados por el segundo modelo de redes neuronales.

[61] Librería de Python. Ver glosario.
[62] Del español arreglo NumPy. Ver glosario.

Este modelo tiene un diccionario propio con las relaciones de las palabras según su contexto. De ahí que una vez entrenado el modelo se le pueda pedir que busque palabras similares a una palabra específica o un grupo de palabras, y el programa te dará las palabras en el corpus con mayor similitud a la palabra escogida, adjuntando también una proporción de similitud con dicha palabra.

5.3 MODELO DE REDES NEURONALES

El segundo y último modelo será un modelo BLSTM (bidirectional long-short term memory), el cual tendrá como entrada las noticias tokenizadas y dará como output un valor de BIAS esperado, tras haber sido entrenado. El modelo de redes neuronales BLSTM se utilizará mediante las bibliotecas tensorflow[63] y keras[64], ampliamente utilizadas para el desarrollo de redes neuronales y aprendizaje profundo en Python. Se cogerán distintas clases y funciones de dichas bibliotecas.

El programa en primer lugar cargará el anterior modelo de tokenización y preparará los vectores asociados y ajustará sus longitudes para que concuerden con las longitudes del indicador BIAS.

Posteriormente, el código cargará la base de datos de entrenamiento de BIAS y las convertirá en un NumPy array, igual que con la base de datos de las noticias.

A continuación, se creará el modelo con las especificaciones necesarias, siendo un modelo bidireccional y con una salida lineal. Al ser la salida buscada un valor continuo,

[63] Biblioteca de Python. Ver glosario.
[64] Biblioteca de Python. Ver glosario

pues es un porcentaje con decimales, el código configura la capa de salida del modelo de red neuronal para realizar una regresión.

Por último, se cargará y entrenará el modelo, teniendo como valores métricos la pérdida y el error absoluto medio (EAM).

Una vez se ha entrenado el modelo, se verificará su validez con las bases de datos de validación. Para ello, se realizará el mismo proceso de código que en el entrenamiento, cambiando las bases de datos por las de validación, y se pedirá al modelo que prediga los valores de las etiquetas o indicadores (los pertenecientes a la base de datos de BIAS). Se utilizará la función y_pred=model.predict(). Una vez haya predicho unos valores, se compararán con los reales para ver la eficacia del modelo.

5.4 RESULTADOS DEL CASO PRÁCTICO

A la hora de estudiar los resultados, para los casos lineares de regresión, hay que fijarse en la pérdida y el error absoluto medio.

- **Pérdida:**
    - **Definición**: La pérdida es una medida que cuantifica la diferencia entre las predicciones del modelo y los valores reales (etiquetas o targets) en un problema de aprendizaje automático. Es una función que evalúa qué tan bien el modelo está haciendo sus predicciones en comparación con los valores reales.
    - **Interpretación:** Una pérdida baja indica que las predicciones del modelo son cercanas a las etiquetas reales, lo que significa que el modelo está

aprendiendo bien los patrones en los datos. Una pérdida alta indica que las predicciones del modelo están lejos de las etiquetas reales, lo que sugiere un rendimiento deficiente del modelo.

- **Optimización:** Durante el entrenamiento, el objetivo es minimizar esta pérdida. Los algoritmos de optimización, como el descenso del gradiente, ajustan los parámetros del modelo para reducir la pérdida.

- **Error Absoluto Medio (EAM):**
  - **Definición:** El error absoluto medio es una métrica que mide la magnitud promedio de las diferencias absolutas entre las predicciones del modelo y los valores reales. En otras palabras, es el promedio de las distancias absolutas entre las predicciones y las etiquetas reales.
  - **Interpretación:** Un EAM más bajo indica que las predicciones del modelo tienen una magnitud promedio más cercana a las etiquetas reales.

Es una métrica que proporciona una medida directa de cuán precisas son las predicciones en términos de las unidades de la variable objetivo.

- **Fórmula:** El EAM se calcula sumando las diferencias absolutas entre cada predicción y el valor real, y luego dividiendo por el número total de ejemplos:

$$EAM = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Donde:

- y_i es el valor de la predicción.
- x_i es el valor real.
- n es el número de ejemplos.

Resultados training output linear 5 noticias	
Pérdida	1,1358
EAM	0,8363
Pérdida validación	4,4793
EAM validación	2,1163

Tabla 4: Resultados training output lineal 5 noticias. Fuente elaboración propia, 2023.

Resultados training output linear 10 noticias	
Pérdida	1,5880
EAM	1,1100
Pérdida validación	2,2440
EAM validación	1,4469

Tabla 5: Resultados training output lineal 10 noticias. Fuente elaboración propia, 2023.

El rendimiento del modelo se examina fijándose tanto en los datos de entrenamiento como en los datos de validación. Para examinar ambas, se ha entrenado el modelo primero con solo cinco noticias y luego con diez noticias. Estudiando los resultados de la Tabla 4 y Tabla 5:

El modelo parece estar aprendiendo bastante bien los datos de entrenamiento, ya que la pérdida y el error absoluto medio en entrenamiento son bajos. Al ser un modelo financiero que predice el porcentaje de crecimiento del valor de una empresa al decimal, tener una pérdida del 1,588 o un error absoluto del 1,11 son valores razonables.

Sin embargo, se aprecia una diferencia significativa entre las métricas de validación cuando se utiliza una base de datos el doble de grande. La pérdida y el error absoluto medio en validación para el caso de cinco noticias son mucho más altos que en entrenamiento, lo que podría indicar que el modelo está sobre ajustando los datos de entrenamiento y tiene dificultades para generalizar a datos nuevos. En cambio, al aumentar la base de datos, se puede ver como los valores de validación se van acercando mucho más a los valores de entrenamiento, y como se reducen prácticamente a la mitad tanto la pérdida como el EAM. Al final, lo que se busca es que el modelo se entrene correctamente para poder obtener unos buenos resultados de validación, pues como se ha comentado antes, la validación son los resultados del modelo ante valores no conocidos por dicho modelo. En ese sentido, se puede apreciar como la mejora es significativa, indicando que el funcionamiento del modelo está siendo el correcto.

Estos resultados son coherentes con nuestro modelo, pues el hecho de que nuestra base de datos tenga tan pocos datos (solo diez noticias como se ha comentado antes), hace que las predicciones de la validación no sean fiables al no haber entrenado al modelo lo

suficiente. En contraposición, el hecho de que las métricas de los entrenamientos sean bajas significa que el modelo sí que es válido.

Habría que añadir también que la salida de este modelo es un valor continuo y no discreto, por lo que el error va a ser mucho mayor puesto que la precisión tiene que ir al decimal.

Por último, se han incluido en el modelo las bases de datos de validación. Es aquí donde se mide el modelo y se puede ver el error entre los valores predichos por el modelo y los valores reales. Al igual que en el caso anterior, se realizará con cinco noticias y posteriormente con diez noticias.

Resultados validation output linear 5 noticias	
ECM	2,1000
EAM	0,9700

Tabla 6: Resultados validation output lineal 5 noticias. Fuente elaboración propia, 2023.

Resultados validation output linear 10 noticias	
ECM	2,9200
EAM	1,2100

Tabla 7: Resultados validation output lineal 10 noticias. Fuente elaboración propia, 2023.

Para ver los resultados de validación del modelo lineal se van a estudiar el error cuadrático medio (ECM) y el error absoluto medio (EAM).

Lo primero que hay que analizar es el hecho de que los errores sean ligeramente mayores al incrementar la base de datos de la noticia. Esto se debe a que la base de datos de diez

noticias comienza a reflejar unos valores similares a los obtenidos en la etapa de entrenamiento, lo que indica que el modelo coge una forma más real. A partir de igualar las métricas con las del modelo de entrenamiento, el incrementar la base de datos va a suponer comenzar a reducir estos errores. En contraposición, las métricas del modelo con cinco noticias difieren mucho de los valores de entrenamiento, por lo que esos valores se podrían considerar todavía poco fiables al ser la base de datos mucho más pequeña.

Si comparamos el ECM con alguno de los modelos estudiados en la literatura relevante, como puede ser el modelo browniano[65], se puede observar como el mínimo valor de dicho modelo, que se puede ver en la Tabla 1, es de 1,65229 un valor ligeramente inferior al obtenido por nuestro modelo. Comparando otros valores de ese estudio se puede ver cómo hay valores parecidos al obtenido en nuestro modelo. Si se tiene en cuenta que ese modelo tiene unas bases de datos mucho mayor, lo que ha permitido entrenarlo con mayor precisión, se puede confirmar que el modelo es válido y muy útil para el objetivo del proyecto.

Resultados training output discrete	
Pérdida	0,4465
Precisión	0,7500
Pérdida validación	0,0124
Precisión validación	1,0000

Tabla 8: Resultados training output discreto. Fuente elaboración propia, 2023.

[65] El perteneciente al artículo de (Rajpal, 2018).

Tras haber estudiado los resultados del modelo, y para verificar su validez y tener más datos de estudio sobre dicho programa, se ha escrito una modificación del principal modelo BLSTM. Dicho modelo, cuyos resultados se encuentran en la Tabla 8, sigue teniendo como entrada la base de datos de noticias, y se incluirá como entrada también la base de datos de BIAS. Así, el indicador de este modelo será ahora un valor discreto de tres clases: 0,1 y 2. Dicho indicador, perteneciente a una base de datos llamada "labeled_training_dataset" y "labeled_validation_dataset" es un análisis sentimental sobre los datos del BIAS. Al ser la salida discreta, el valor métrico del error absoluto medio no se utiliza y en cambio se utiliza la precisión, siendo mejor para el modelo cuanto más alta sea dicha precisión.

El análisis funciona de la siguiente manera:

- Si el crecimiento es mayor de un 0,5% se considera que la noticia ha influido positivamente en el valor de la compañía, y se asigna un 2 al valor de salida.
- Si el crecimiento se encuentra entre [-0,5%-0,5%], se considera que la noticia no ha influido en el valor de la compañía, y se asigna un 1 al valor de salida.
- Si el crecimiento es menor de un -0,5% se considera que la noticia ha influido negativamente en el valor de la compañía, y se asigna un 0 al valor de salida.

Este modelo simplificado permite predecir si el valor de la empresa va a crecer, mantenerse estable o disminuir. Es cierto que este modelo es muy sencillo de entrenar puesto que al tener el valor BIAS de cada noticia, la predicción es directa. Aun así, esto nos permite ver que el funcionamiento del modelo es el correcto y que los errores obtenidos en el primer modelo se deben únicamente a la falta de entrenamiento del modelo.

Una vez se han estudiado los resultados del entrenamiento del modelo, se van a abordar ahora los resultados pertenecientes al periodo de validación o test.

Resultados validation output discrete	
Precisión	1,0000
Recall	1,0000
F1-score	1,0000

Tabla 9: Resultados validation output discreto. Fuente elaboración propia, 2023.

Por último, los resultados con los datos de validación dan unos valores perfectos en todas las métricas, como se puede observar en la Tabla 9. Esto, como se ha comentado justo arriba, indica que el modelo funciona perfectamente; aunque es cierto que la simplicidad de este modelo permite obtener estos valores.

5.5 CONCLUSIONES

Tras el desarrollo del caso práctico, se ha podido analizar la funcionalidad del modelo. La comparación entre los distintos entrenamientos del modelo y del tipo de salida, como han sido el entrenamiento entre cinco y diez noticias y entre una salida lineal y discreta, ha arrojado un resultado que verifica que el modelo sigue una línea de aprendizaje correcta. El hecho de que la base de datos fuera manual ha impedido que se pudiera entrenar el modelo de una manera más precisa, y por tanto obtener unos resultados realmente precisos. Aun así, el objetivo de este caso práctico era desarrollar un modelo que fuera válido para el propósito del proyecto, y que en un futuro se pudiera seguir entrenando para conseguir convertirlo en un programa completamente fiable.

Capítulo 6. CONCLUSIONES Y PASOS FUTUROS

Para concluir este trabajo, se realizará un resumen de los objetivos que se habían propuesto y se valorará el cumplimiento de dichos objetivos basándose en los resultados obtenidos en el caso práctico. Por último, se comentarán pasos futuros recomendables para seguir perfeccionando el modelo.

Los principales objetivos son:

4. **Creación de un modelo necesario para la predicción bursátil:** El principal objetivo de este proyecto no era otro que, como se vió en la literatura relevante, había una necesidad de crear un modelo completo que englobara distintas funcionalidades de otros modelos, para crear un programa que fuera capaz de realizar todo el proceso completo, desde la extracción de la información relevante hasta la predicción de crecimiento del valor de una empresa. Este modelo ha sido capaz de juntar distintos proyectos para conseguir un modelo general de gran dimensión y precisión.
5. **Creación de una base de datos para el modelo:** Este proyecto ha incluido bases de datos propias, indicando las características importantes que han de tener esas bases de datos, tanto las de noticias como las de valores bursátiles (BIAS), para poder dar al modelo información organizada y de gran ayuda para entrenarlo.
6. **Conseguir un modelo que reduzca la incertidumbre de una acción bursátil:** Analizando este primer objetivo, y tomando los resultados del caso práctico como referencia, se puede afirmar que los valores obtenidos de errores absolutos y

cuadráticos, al igual que la pérdida, precisión, recall y F1; arrojan resultados muy interesantes y completamente válidos. Por tanto, se puede afirmar que este modelo reduce la incertidumbre de una acción bursátil, teniendo en cuenta que una mayor base de datos va a propiciar un mejor entrenamiento del modelo y por tanto una reducción incluso mayor de dichos errores, lo que supone una mejora en la precisión.

7. **Conseguir obtener una relación temporal con la variación bursátil:** Este segundo objetivo, el cual es más complejo; también ha sido abordado en este proyecto. Al haber tenido como indicador al valor BIAS, el cual coge los valores bursátiles el día de la noticia y luego coge la media de los valores x días después (4 días en nuestro caso práctico) va a permitir estudiar la relación temporal, pues solo va a haber que estudiar cuantos días hacen falta para que todos los crecimientos bursátiles se encuentren dentro del rango de [-0,5% - 0,5%] que se atribuye a una no influencia de la noticia en el valor de la empresa.

6.1 PASOS FUTUROS

Habiendo comentado los principales objetivos, toca analizar qué componentes se podrían mejorar para mejorar la precisión y reducir los errores del modelo.

5. Crear una base de datos mucho mayor que permita entrenar con más precisión al modelo.
6. Incluir algunos aspectos del primer modelo teórico en la tokenización del cuerpo de las noticias, para crear así un primer modelo más robusto que únicamente el

Word2Vec, lo que supondrá una ayuda al modelo BLSTM al evitar sobre ajustes en el modelo por exceso de información o por tener información no precisa.

7. Crear un modelo automático que obtenga la base de datos BIAS sin tener que incluir los valores de manera manual. Uno de los principales problemas de no haber creado bases de datos más grandes se ha debido a la necesidad de incluir el valor BIAS de manera manual. Un programa que cogiera una base de datos de valores bursátiles históricos y teniendo el día que salió la noticia y el número x de días que se van a utilizar, consiga obtener el valor BIAS de crecimiento teniendo en cuenta el crecimiento en esas fechas de la bolsa en la que cotice la empresa que se quiere estudiar.

Capítulo 7. BIBLIOGRAFÍA

Anon., 2021. *The Thought Tree.* [En línea]

Available at: https://thethoughttree.com/blog/how-to-predict-the-stock-market/

Anon., s.f. *FXStreet.* [En línea]

Available at: https://www.fxstreet.com/economic-calendar

Anon., s.f. *Software > Stanford Parser.* [En línea]

Available at: https://nlp.stanford.edu/software/lex-parser.shtml#:~:text=The%20parser%20can%20read%20various,relations%20(typed%20dependency)%20format.

Basilone, R., 2021. The Impact of News on Stock Market Investors.

Bellintani, L., s.f. The Effect of News on Intra-Day Stock Prices & their Volatility. *Student Economic Review,* Volumen 33.

Brownlee, J., 2017. *A Gentle Introduction to the Bag-of-Words Model.* [En línea]

Available at: https://machinelearningmastery.com/gentle-introduction-bag-words-model/

Early, J., 2022. *Towards Data Science.* [En línea]

Available at: https://towardsdatascience.com/understanding-the-outputs-of-multi-layer-bi-directional-lstms-13ad99a80dd3

Edna S. Solano, P. D. C. M. A., 2022. Solar Radiation Forecasting Using Machine Learning and Ensemble Feature Selection. *Energies,* 25 September.

Gabriel Pui Cheong Fung, J. X. Y. W. L., 2002. News Sensitive Stock Trend Prediction. *Advances in knowledge discovery and data mining*, 06 May.pp. 481-493.

Guareño, J. J. M., s.f. SUPPORT VECTOR REGRESSION: PROPIEDADES Y APLICACIONES.

Hans Christian, M. P. A. D. S., 2016. SINGLE DOCUMENT AUTOMATIC TEXT SUMMARIZATION USING TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF). *ComTech*, December.pp. 285-294.

Ingolfsson, T. M., 2021. Insights into LSTM architecture. *Thorir Mar Ingolfsson.*

Jie Wu, D.-Y. H. L. X. H. L., 2017. Denoising Recurrent Neural Network for Deep Bidirectional LSTM Based Voice Conversion.

Joachims, T., 1998. Making Large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning.*

Karabiber, F., 2023. *TF-IDF — Term Frequency-Inverse Document Frequency.* [En línea]
Available at: https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/#:~:text=Using%20scikit%2Dlearn-,What%20is%20TF%2DIDF%3F,%2C%20relative%20to%20a%20corpus).

Kristof Coussement, D. V. d. P., 2008. Improving customer complaint management by automatic email classification using linguistic style features as predictors. *Decision Support Systems*, 44(4), pp. 870-882.

Marco Castangia, A. A. L. B. E. M. E. P., 2021. A compound of feature selection techniques to improve solar. *Expert Systems With Applications*, 20 April.Volumen 178.

Matthew Butler, V. K., 2009. Financial Forecasting Using Character N-Gram Analysis and Readability Scores of Annual Reports.. *Lecture Notes in Computer Science*, Volumen 5549.

Michael Hagenau, M. L. D. N., 2013. Automated news reading: Stock price prediction based on financial news using context-capturing features. *Decision Support Systems*, 55(3), pp. 685-697.

Miranda, F. V., 2017. Inversión y financiarización: un modelo matemático kaleckiano. *Tiempo Económico*, 12(35).

Mücahid Mustafa Saritas, A. Y., 2019. Performance Analysis of ANN and Naive Bayes Classification Algorithm. *International Journal of Intelligent Systems and Applications in Engineering.*

Paul C. Tetlock, M. S.-T. a. S. M., 2008. More Than Words: Quantifying Language. *THE JOURNAL OF FINANCE , VOL. LXIII, NO. 3.*

Peter A. Flach, N. L., 2004. Naive Bayesian Classification of Structured Data. *Machine Learning*, Volumen 57, p. 233–269.

Rajpal, R., 2018. Mathematical Modeling in Finance. *International Journal of Scientific & Engineering Research*, 9(11).

Robert P. Schumaker, H. C., 2009. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Transactions on Information Systems*, 27(2), p. 1–19.

Robert P. Schumaker, N. M., 2018. Analysis of Stock Price Movement Following Financial News. *Communications of the IIMA*, 16(1).

Shailendra Kumar Singh, S. P., 2015. Sentiment Analysis of Social Issues and Sentiment Score Calculation of Negative Prefixes. *International Journal of Applied Engineering Research,* pp. 1694-1699.

Tetlock, P. C., 2007. Giving Content to Investor Sentiment: The Role of Media in the Stock Market. *The journal of finance,* 62(3), pp. 1139-1168.

Tim Loughran, B. M., 2011. When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *The Journal of Finance 66,* pp. 35-65.

Willett, P., 2006. The Porter stemming algorithm: Then and now. *Program Electronic Library and Information Systems.*

Yinghao Ren, F. L. Y. G., 2020. Impact of News on the Trend of Stock Price Change: an Analysis based on the Deep Bidirectiona LSTM Model. *Procedia Computer Science,* Volumen Volume 174, pp. Pages 128-140.

ANEXO I: CÓDIGO DE PROGRAMACIÓN DEL CASO PRÁCTICO

```
#!/usr/bin/env python
# coding: utf-8

# In[8]:


from gensim.models import Word2Vec
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import pandas as pd
import numpy as np

# Download NLTK resources (only required once)
nltk.download('punkt')
nltk.download('stopwords')

# Load stopwords
stop_words = set(stopwords.words('spanish'))

# Preprocess the news data
corpus = []  # List to store preprocessed news articles

# CSV file named 'news_training_dataset.csv' with a
'summary' column
data = pd.read_csv('news_training_dataset.csv')

# Initialize the model
model = Word2Vec(window=50, min_count=1, workers=4)

# List to store embeddings for each news summary
news_embeddings = []

for summary in data['summary']:
    if isinstance(summary, str):  # Check if 'summary' is a
non-null string
        tokens = word_tokenize(summary.lower())  # Tokenize
the text
        filtered_tokens = [token for token in tokens if
token.isalpha() and token not in stop_words]  # Filter out
non-alphabetic tokens and stopwords
        corpus.append(filtered_tokens)
```

```
        # Calculate embeddings for each token in the
summary
        embeddings = [model.wv[word] if word in model.wv
else np.zeros(model.vector_size) for word in
filtered_tokens]
        news_embeddings.append(embeddings)  # Store the
embeddings


# Build the vocabulary of the Word2Vec model
model.build_vocab(corpus)

# Train the Word2Vec model
model.train(corpus, total_examples=len(corpus), epochs=10)

# Save the Word2Vec model to a file
model.save("word2vec_model.bin")

path_to_your_word2vec_model = "word2vec_model.bin"

# Save the path_to_your_word2vec_model to a file
with open("word2vec_model_path.txt", "w") as f:
    f.write(path_to_your_word2vec_model)

# Convert the list of embeddings to a numpy array
max_sequence_length = 100
news_dim = model.vector_size
X_news_word_embeddings = []
for embeddings in news_embeddings:
    padded_embeddings = embeddings[:max_sequence_length] +
[np.zeros(news_dim)] * (max_sequence_length -
len(embeddings))
    X_news_word_embeddings.append(padded_embeddings)
X_news_word_embeddings = np.array(X_news_word_embeddings)

# Save the news embeddings as a numpy array
np.save("news_embeddings.npy", X_news_word_embeddings)

# Find similar words
similar_words = model.wv.most_similar(['inversion',
'renovables'])

print(similar_words)


# In[14]:


import numpy as np
import tensorflow as tf
```

```
from tensorflow.keras.layers import Input, LSTM,
Bidirectional, Dense, Concatenate, RepeatVector
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.sequence import
pad_sequences
from gensim.models import Word2Vec
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import pandas as pd
import nltk



# Preprocess the news data
corpus = []  # List to store preprocessed news articles

# Load or create your Word2Vec model
model_w2v = Word2Vec.load("word2vec_model.bin")

# Query similar words dynamically
query_words = ['inversion', 'renovables']
num_similar_words = 10
#most_similar_output =
model_w2v.wv.most_similar(query_word,
topn=num_similar_words)
#similar_words = [word for word, _ in most_similar_output]

similar_words_combined = []

for query_word in query_words:
    most_similar_output =
model_w2v.wv.most_similar(query_word,
topn=num_similar_words)
    similar_words = [word for word, _ in
most_similar_output]
    similar_words_combined.extend(similar_words)

# Assuming you have a CSV file named 'noticias-economicas-
sample.csv' with a 'summary' column
data = pd.read_csv('news_training_dataset.csv')
summaries = data['summary']

# Initialize a list to store the lengths of each sequence
sequence_lengths = []

input_embeddings_list = []

max_sequence_length = 100  # Define the maximum sequence
length
news_dim = model_w2v.vector_size  # Get the dimension of
word embeddings
```

```
# Loop through each summary and process it
for summary in summaries:
    if isinstance(summary, str):
        tokens = word_tokenize(summary.lower())  # Tokenize
the text
        filtered_tokens = [token for token in tokens if
token.isalpha() and token not in stop_words]  # Filter out
non-alphabetic tokens and stopwords
        corpus.append(filtered_tokens)

        # Calculate embeddings for each token in the
summary
        embeddings = [model_w2v.wv[word] if word in
model_w2v.wv else np.zeros(model_w2v.vector_size) for word
in filtered_tokens]
        input_embeddings_list.append(embeddings)  # Store
the embeddings
        sequence_lengths.append(len(embeddings))  # Store
the length of this sequence



# Pad sequences to the maximum length
X_news_word_embeddings =
pad_sequences(input_embeddings_list,
maxlen=max_sequence_length, dtype='float32',
padding='post', truncating='post',
value=np.zeros(news_dim))



# Load the bias values from the bias database CSV file
bias_data = pd.read_csv('bias_training_dataset.csv')
bias_values = bias_data['bias_value'].values.reshape(-1, 1)


y_labels = np.array(bias_values)


# Iterar a través de cada palabra en la lista
for query_word in query_words:
    # Get word vectors for the current query_word
    vector = model_w2v.wv[query_word]

    # Find similar words for the current query_word
    similar_words = model_w2v.wv.most_similar(query_word)

# Get word vectors
#vector = model_w2v.wv[query_word]
```

```
# Find similar words
#similar_words = model_w2v.wv.most_similar(query_word)

# Rest of your existing code
num_samples = X_news_word_embeddings.shape[0]
seq_length = max_sequence_length


news_input = Input(shape=(seq_length, news_dim),
name='news_input')


news_lstm = Bidirectional(LSTM(64,
return_sequences=True))(news_input)


output_layer = Dense(1, activation='linear')(news_lstm)

# Compile and train the model
model = Model(inputs=[news_input], outputs=output_layer)
model.compile(optimizer='adam', loss='mean_squared_error',
metrics=['mean_absolute_error'])

model.fit({'news_input': X_news_word_embeddings}, y_labels,
epochs=10, batch_size=32, validation_split=0.2)


# In[13]:


from sklearn.metrics import accuracy_score,
precision_score, recall_score, f1_score
from sklearn.metrics import mean_squared_error,
mean_absolute_error, r2_score


# Assuming you have a CSV file named 'validation_data.csv'
with a 'summary' column
data = pd.read_csv('news_validation_dataset.csv')
summaries = data['summary']

# Initialize a list to store the lengths of each sequence
sequence_lengths = []

input_embeddings_list = []

max_sequence_length = 100  # Define the maximum sequence
length
news_dim = model_w2v.vector_size  # Get the dimension of
word embeddings
```

```
# Loop through each summary and process it
for summary in summaries:
    if isinstance(summary, str):
        tokens = word_tokenize(summary.lower())  # Tokenize
the text
        filtered_tokens = [token for token in tokens if
token.isalpha() and token not in stop_words]  # Filter out
non-alphabetic tokens and stopwords
        corpus.append(filtered_tokens)

        # Calculate embeddings for each token in the
summary
        embeddings = [model_w2v.wv[word] if word in
model_w2v.wv else np.zeros(model_w2v.vector_size) for word
in filtered_tokens]
        input_embeddings_list.append(embeddings)  # Store
the embeddings
        sequence_lengths.append(len(embeddings))  # Store
the length of this sequence

# Pad sequences to the maximum length
X_val_news_word_embeddings =
pad_sequences(input_embeddings_list,
maxlen=max_sequence_length, dtype='float32',
padding='post', truncating='post',
value=np.zeros(news_dim))

# Load the bias values from the bias database CSV file
bias_data = pd.read_csv('bias_validation_dataset.csv')
y_val = bias_data['bias_value'].values.reshape(-1, 1)


# Generate predictions for the validation data
y_pred = model.predict({'news_input':
X_val_news_word_embeddings})

# Convert predictions to labels
y_pred_labels = np.argmax(y_pred, axis=1)

# Get the number of samples and sequence length
n_samples, seq_length, news_dim =
X_val_news_word_embeddings.shape

# Initialize lists for per-sequence metrics
mse_per_sequence = []
mae_per_sequence = []

# Calculate metrics for each sequence
for i in range(n_samples):
    # Extract the current sequence and corresponding
predictions
```

```
    current_sequence = X_val_news_word_embeddings[i]
    current_sequence_pred = y_pred[i]

    # Get the unique value from y_val[i]
    unique_value = y_val[i][0]

    # Repeat the unique value to match the sequence length
    y_val_i_adjusted = np.full_like(current_sequence_pred,
unique_value)

    # Calculate MSE for the current sequence
    mse = mean_squared_error(y_val_i_adjusted,
current_sequence_pred)

    # Calculate MAE for the current sequence
    mae = mean_absolute_error(y_val_i_adjusted,
current_sequence_pred)


    # Append metrics to respective lists
    mse_per_sequence.append(mse)
    mae_per_sequence.append(mae)


# Calculate the average metrics
mse_promedio = np.mean(mse_per_sequence)
mae_promedio = np.mean(mae_per_sequence)


# Print evaluation results
print(f'Mean Squared Error (MSE): {mse_promedio:.2f}')
print(f'Mean Absolute Error (MAE): {mae_promedio:.2f}')

# In[ ]:
```

ANEXO II: OBJETIVOS DE DESARROLLO SOSTENIBLE (ODS)

Este trabajo se encuentra alineado con alguno de los principales objetivos de desarrollo sostenible. El ODS con el cual se encuentra más alineado este trabajo es:

4. **ODS 8: Trabajo Decente y Crecimiento Económico:** La predicción de los valores bursátiles va a promover un crecimiento económico sostenible y la posibilidad de creación de empleo en el sector financiero.