

Universidad de La Habana  
Facultad de Matemática y Computación



# **Predicción de Mercado Utilizando Información de Noticias**

Autor:

**Alex Sánchez Saez, Carlos Manuel Gonzáles Peña,  
Jorge Alberto Aspiolea**

Tutores:

Trabajo de Diploma  
presentado en opción al título de  
Licenciado en (Ciencia de la Computación)

Fecha

07-07-2024

# Índice general

|  |           |
|--|-----------|
| <b>Introducción</b>  | <b>1</b>  |
| <b>1. Estado del Arte</b>  | <b>2</b>  |
| 1.0.1. Marco Teórico . . . . .   | 2         |
| 1.0.2. Revisión de la Literatura . . . . .   | 2         |
| 1.0.3. Técnicas de ensembling observadas . . . . .   | 4         |
| 1.0.4. Comparación y Análisis Crítico . . . . .  | 4         |
| 1.0.5. Identificación de Huecos en la Literatura . . . . .   | 5         |
| 1.0.6. Forecasting with Covariance . . . . .   | 5         |
| <b>2. Propuesta</b>  | <b>6</b>  |
| 2.1. Ideas Generales . . . . .   | 6         |
| 2.1.1. Red Adversarial (GAN) para predecir movimientos del mercado                                     | 6         |
| 2.1.2. Clusterización del espacio de las noticias . . . . .  | 7         |
| 2.1.3. Red Neuronal Convolutiva . . . . .  | 7         |
| 2.1.4. Árboles de Decisión y Random Forest . . . . .   | 8         |
| 2.1.5. Uso de Redes neuronales Recurrentes . . . . .   | 8         |
| 2.1.6. Redes neuronales con Arquitectura transformer . . . . .   | 8         |
| 2.1.7. Análisis de Sentimientos de las Noticias, teniendo en cuenta su<br>relevancia . . . . .         | 9         |
| <b>3. Detalles de Implementación y Experimentos</b>  | <b>10</b> |
| 3.1. Ideas Iniciales . . . . .   | 10        |
| 3.1.1. Obtención y análisis de los datos . . . . .   | 10        |
| 3.1.2. Análisis de los valores del dataset . . . . .   | 10        |
| 3.1.3. Análisis de la distribución de los datos . . . . .  | 10        |
| 3.1.4. Análisis bivariable de los datos . . . . .  | 14        |
| 3.1.5. Detección de Outlayers . . . . .  | 17        |
| 3.1.6. Limpieza y normalización de los datos . . . . .   | 18        |
| 3.1.7. Análisis de los outliers después de normalizar con la transfor-<br>mación logarítmica . . . . . | 20        |

|           |  |           |
|-----------|--|-----------|
| 3.2.      | Análisis del dataset de noticias . . . . .   | 21        |
| 3.2.1.    | Clusterización de las noticias . . . . .   | 24        |
| 3.3.      | Entrenamiento y diseño del Modelo . . . . .  | 26        |
| 3.3.1.    | Primera Iteración : Entrenando con los datos Close y Volumen                                     | 26        |
| 3.3.2.    | Segunda Iteración: Entrenando Agregando un Indicador EMA   | 27        |
| 3.3.3.    | Iteración 3 utilizando todas las columnas del dataset (Close, Open, High, Low, Volume) . . . . . | 29        |
| 3.3.4.    | Iteración 4 (Añadiendo noticias) . . . . .   | 31        |
| 3.3.5.    | Iteración 5 Agrandando el Modelo . . . . .   | 32        |
| 3.3.6.    | Iteración 6: Modificación de los hiperparámetros del modelo .                                    | 33        |
| 3.3.7.    | Iteración 8: Rectificando errores en los datos . . . . .   | 35        |
| <b>4.</b> | <b>Análisis de los resultados</b>  | <b>36</b> |
| 4.0.1.    | Resultados . . . . .   | 36        |
| 4.0.2.    | Recomendaciones . . . . .  | 36        |
|           | <b>Conclusiones</b>  | <b>37</b> |

# Introducción

Con este trabajo pretendemos contestar a la pregunta de : ¿Es posible predecir el mercado con eficacia. Este tema nos es de mucho interés y nos causa una gran curiosidad saber que requerimientos tendría en caso de que la respuesta a esta pregunta fuese afirmativa. Como es un marco demasiado amplio ya que el mercado es un componente complejo del sistema en que nos situamos y es dependiente de diversos factores de naturaleza distinta, como pueden ser las catástrofes, la influencia de personalidades o redes sociales, eventos políticos, pandemias globales etc. Es por esto que quisimos simplificar más nuestra pregunta, y la redujimos a : ¿Es posible predecir el comportamiento del mercado dada la información de noticias extraídas del mundo real?. Debido a la aparente aleatoriedad tanto del comportamiento del mercado como de las noticias, así como la dificultad de diseñar un algoritmo que se encargue de esta tarea, decidimos recurrir al aprendizaje automático (Machine Learning). Esta tecnología permite que una computadora aprenda posibles patrones y pueda predecir el comportamiento del mercado de manera aproximada.

Nuestra investigación abarcó varios temas del ámbito del aprendizaje automático, desde el estudio y análisis de varios modelos, hasta la recolección, tratamiento y preparación de los datos. Fue necesaria una profunda revisión de trabajos relacionados con esta temática, observando que muy pocos abordaban la perspectiva de utilizar noticias para dicha tarea.

Este estudio se enmarca en un ámbito mayor que es el análisis y predicción de series temporales. Las series temporales son datos secuenciales recogidos a intervalos regulares de tiempo y son fundamentales para el análisis predictivo en diversos campos, incluyendo la economía y las finanzas. La relación entre la serie de datos del mercado y las noticias es crucial, ya que las noticias pueden influir en las decisiones de los inversores y, por ende, en el comportamiento del mercado. Utilizando técnicas de "Forecasting with Covariance", podemos integrar estas variables adicionales (las noticias) en los modelos predictivos para mejorar la precisión de las predicciones.

# Capítulo 1

## Estado del Arte

En esta sección se revisa la literatura existente relacionada con la predicción del comportamiento del mercado financiero utilizando técnicas de aprendizaje automático. Esta revisión proporcionará el contexto necesario y destacará las contribuciones previas en este campo. Así como el análisis de resultados de cada una de las propuestas encontradas

### 1.0.1. Marco Teórico

El análisis de series temporales y el aprendizaje automático son dos pilares fundamentales en el campo de la predicción de mercados. En esta sección, se revisan los conceptos clave y las teorías subyacentes a estas técnicas.

### 1.0.2. Revisión de la Literatura

#### Métodos Tradicionales de Predicción de Mercados

Se identificaron dos enfoques principales para el análisis y la predicción del comportamiento del mercado: el análisis técnico y el análisis fundamental. Estos métodos se han consolidado como pilares fundamentales en la disciplina financiera debido a sus enfoques distintos pero complementarios para interpretar y prever las tendencias del mercado.

#### 1. Análisis Técnico

El análisis técnico se centra en la identificación de patrones en los datos históricos del mercado, tales como precios, volúmenes de transacciones y otras variables derivadas. Este enfoque se basa en la premisa de que los movimientos de los precios no son aleatorios, sino que siguen tendencias y patrones que pueden ser identificados y utilizados para predecir movimientos futuros. Los

analistas técnicos emplean una variedad de herramientas y técnicas, incluyendo gráficos, indicadores técnicos y modelos matemáticos, para detectar señales de compra y venta en el mercado. Este enfoque es particularmente útil en el corto plazo, donde los patrones históricos pueden repetirse con mayor frecuencia.

## 2. Análisis Fundamental

Por otro lado, el análisis fundamental se basa en la evaluación del estado financiero de una empresa y su entorno económico para determinar el valor intrínseco de sus acciones. Este enfoque examina factores como los estados financieros, la gestión de la empresa, las condiciones del sector y las tendencias macroeconómicas. Los analistas fundamentales creen que el valor de una acción está determinado por el rendimiento económico y financiero de la empresa, y que el mercado eventualmente reflejará este valor en el precio de la acción. Este tipo de análisis es esencial para inversiones a largo plazo, ya que proporciona una visión más profunda de los factores subyacentes que afectan el rendimiento de una empresa.

Los Modelos Más utilizados en la literatura son:

1. SVM (Support vector Machine) para regresión en un enfoque multi etapa.
2. computational efficient functional link artificial neural network (CELANN) un modelo de red neuronal con una única capa oculta que permite obtener un mejor performance .
3. Long Short Term Memory (LSTM) un tipo de red neuronal recurrente (RNN) Optimizada para corregir el problema de desvanecimiento de gradientes en estas (RNN) lo que permite utilizar modelos mas grandes .
4. Modelo de Medias Móviles (Simple Moving Average ) (SMA) (Modelo estadístico).
5. Media Móvil Integrada Auto-regresiva (Auto-Regresive Integrated Moving Average)(ARIMA) (Modelo Estadístico).
6. método Holt-Winters, también conocido como suavización exponencial triple, (Modelo Estadístico).
7. LSTM Convolucionales

### 1.0.3. Técnicas de ensembling observadas

1. XBoosting, usando Desission Tree y Random Forest
2. Gradient Boosting
3. ADA Boosting

### Aplicaciones de Aprendizaje Automático en Finanzas

A pesar de ser capaces de capturar no linealidades y comportamientos mas complejos en los datos de series temporales financieras los algoritmos de Aprendizaje de Máquina no consiguen capturar toda la complejidad del comportamiento del mercado debido a la gran cantidad de elementos de la realidad que interfieren en el mercado y la esencia casi aleatoria de dichos datos. Tienen en su mayoría el defecto de ser computacionalmente intensos y de requerir grandes volúmenes de datos , lo cuál no representa un problema debido a que existen muchos datos financieros con disponibilidad. Modelos más potentes como las redes neuronales carecen de explicabilidad , por lo que no es posible extraer los patrones que detectan dichos modelos . Otros trabajos basados en modelos como randon forest y arboles de decision , tienen mayor explicabiliad , pero presentan un rendimiento menor incluso que los modelos estadísticos clásicos. Modelos como las medias móviles , ARIMA y sus variantes son los más utilizados en la actualidad ya que requieren de tareas computacionalmente menos demandantes y aportan resultados comparables alos ofrecidos por los modelos de aprendizaje.

### Uso de Noticias y Datos No Estructurados

En general el uso de noticias para relacionar los movimientos del mercado con los acontecimientos del mundo real es escaso en la bibliografía consultada, pero hay enfoques similares utilizando datos sobre política y relacionando las probabilidades de subir o bajar valores de acciones en dependencia de eventos ocurridos en el mundo real

### 1.0.4. Comparación y Análisis Crítico

En general para poder predecir efectivamente la esencia dinámica del comportamiento del mercado y su relación a eventos externos como eventos políticos, eventos sociales como por ejemplo el aislamiento social provocado por la pandemia , catástrofes naturales, influencias sociales etc, serían necesaios modelos muy potentes que sepan capturar la esencia de estos eventos , lo cuál lleva a la necesidad de una cantidad de datos muy grande y variables, necesitando de diferentes fuentes y datos de

naturaleza muy distinta, lo cuál dificulta el tratamiento y modelación de estos datos para su uso. Por otra parte modelos complejos capaces de establecer relaciones entre estos datos requerirían de una amplia cantidad de recursos para tener un correcto funcionamiento, esto supone un freno en el desarrollo de herramientas para llevar a cabo predicciones efectivas en el ámbito financiero. Por su parte los modelos estadísticos siguen dando resultados cuanto menos comparables con los modelos de aprendizaje de máquina aunque estos cuentan con la limitación de no poder capturar en su mayoría no linealidades presentes en el histórico de los datos ni poder establecer una relación entre estos y datos de naturaleza diferente como las noticias u otros eventos.

#### **1.0.5. Identificación de Huecos en la Literatura**

Aunque se ha avanzado en el uso de aprendizaje automático para la predicción del mercado, la literatura muestra una falta de estudios que integren de manera efectiva noticias y datos no estructurados en los modelos predictivos.

#### **1.0.6. Forecasting with Covariance**

El "Forecasting with Covariance" implica la incorporación de múltiples series temporales relacionadas, permitiendo que las variaciones en una serie (como las noticias) informen las predicciones en otra serie (como los precios del mercado). Este enfoque reconoce que los mercados no operan de manera aislada, sino que están influenciados por una amplia gama de factores externos. Al incluir estas variables adicionales, nuestros modelos pueden captar mejor las dinámicas complejas que afectan los movimientos del mercado.



# Capítulo 2

## Propuesta

El objetivo central de nuestra investigación es encontrar la manera de relacionar los precios de las acciones en el mercado con factores externos, específicamente a través del análisis de noticias. Nuestra premisa inicial se centra en hallar una representación adecuada para las noticias que nos permita extraer su contenido y determinar su relación con la variación en el precio del activo en cuestión.

En este contexto, nos enfocamos en acciones de empresas tecnológicas, particularmente en Apple. El propósito es desarrollar un modelo que pueda identificar y cuantificar el impacto de las noticias sobre las fluctuaciones del precio de las acciones de Apple. Para lograr esto, empleamos técnicas avanzadas de procesamiento del lenguaje natural (NLP) y análisis de sentimientos, con el fin de extraer información relevante de las noticias y correlacionarla con los movimientos en el mercado bursátil.

### 2.1. Ideas Generales

#### 2.1.1. Red Adversarial (GAN) para predecir movimientos del mercado

El proceso comenzaría con la recopilación y organización cronológica de las noticias, asociándolas con las fechas correspondientes y etiquetándolas según su impacto en el precio de la acción en ese momento del tiempo, es decir, si contribuyeron a una subida o bajada del precio del activo.

Con esta información, procederíamos a entrenar un predictor. Este predictor sería capaz de tomar como entrada los embeddings de las noticias y predecir el efecto que estas tendrán sobre el precio de las acciones.

Posteriormente, utilizando datos del comportamiento histórico del precio de las acciones, entrenaríamos un discriminador. Este discriminador tendría la tarea de evaluar si el comportamiento descrito por las noticias es congruente con los patrones

observados en el comportamiento histórico del precio de las acciones. En esencia, el discriminador verificaría la validez y precisión del predictor, asegurando que las predicciones se alineen con las tendencias históricas observadas.

Una vez entrenado el modelo este sería capaz de relacionar efectivamente el contenido de las noticias con los patrones de comportamiento del precio de un activo en el mercado, es decir el movimiento de este.

No llevamos a cabo esta alternativa debido a la dificultad para encontrar un dataset de noticias etiquetado debidamente con dada una noticia su efecto en el mercado y por cuestiones del tiempo disponible para el desarrollo de esta investigación

### 2.1.2. Clusterización del espacio de las noticias

Otra alternativa que tuvimos en cuenta fue realizar una clusterización del espacio de las noticias (conjunto de noticias en nuestro dataset) para así agruparlas por características comunes, luego de esto clasificar los centroides encontrados por su aporte, positivo, negativo o neutro respecto al activo que queremos predecir, así para clasificar una noticia solo tendríamos que ver a que centroide está más cercana (a que cluster pertenece) y clasificar según dicho centroide; es decir, cada cluster tendría una clasificación de acción positiva, neutra o negativa y asumiríamos que todas las noticias que están presentes en ese cluster también, la representación de las noticias podría ser tanto en embeddings de las mismas como modelos más sencillos como bag of words o tf-idf si se requiriera de un análisis del corpus de noticias. Una vez hecho esto podríamos construir un dataset sintético donde colocáramos el movimiento del mercado en cada fecha y la acción de su noticia correspondiente a la fecha. Con esto entrenaríamos un regresor usando alguno de los modelos disponibles como una regresión lineal o una red neuronal. Esta idea fue descartada ya que introduce el sesgo de que todas las noticias de un cluster tienen el mismo efecto en el activo, aunque sean noticias semánticamente iguales, no quiere decir que tengan el mismo efecto en el mercado.

### 2.1.3. Red Neuronal Convolutiva

Otra de las ideas que tuvimos presente fue la de utilizar una red neuronal convolutiva que captara estructuras más complejas de la serie de tiempo representada por las noticias y el comportamiento del activo a predecir. Esto inspirado en la literatura citada requeriría de construir un dataset que involucrara tanto al estado de la moneda en cada fecha, como a la noticia presente en esa fecha, permitiendo al modelo capturar patrones y estructuras subyacentes en ventanas o márgenes de tiempo. Para llevar a cabo esta solución chocamos con la problemática de encontrar una forma de modelar este escenario ya que por ejemplo las noticias pueden estar afectando el

valor del activo un determinado tiempo, el cuál es muy difícil de predecir. Además de la dificultad que implica determinar el tamaño de las ventanas de tiempo y como estructurar el dataset; por esto, esta idea fue descartada.

#### 2.1.4. Árboles de Decisión y Random Forest

Otra de las ideas latentes en la literatura consultada que nos llamó bastante la atención fue la idea de utilizar random forest a través de ensembles como Gradient Boosting y Bagging que fueron analizados en la bibliografía encontrada. Estos enfoques son potentes pues los árboles de decisión son capaces de capturar características específicas de cada conjunto de datos y al combinar el conocimiento obtenido por varios de estos modelos se pueden capturar comportamientos más complejos en la estructuración de la relación Noticia-Mercado. Esto aportaría una mayor explicabilidad del modelo propuesto y aportaría en el caso de Bagging la capacidad de entrenar estos modelos por separado y después utilizar el conocimiento de estos en conjunto. Cada uno de los random forest se entrenaría con un fragmento del dataset donde capturaría diferentes características tanto de las noticias como del comportamiento del mercado, ayudando así a relacionar estas variables y permitiendo emerger patrones y características más complejas. Este enfoque tiene un problema con la tendencia al sobreajuste que haría que el modelo aprendiera muy bien de los datos de entrenamiento pero no sea capaz de generalizar.

#### 2.1.5. Uso de Redes neuronales Recurrentes

Otro acercamiento a la resolución del problema planteado es el uso de redes neuronales recurrentes, las cuáles permiten procesar series de tiempo con mayor eficacia ya que le permite al modelo no solo aprender de los datos de entrada si no que puede aprender de datos anteriormente aprendidos en estados anteriores. Este enfoque tiene el problema del desvanecimiento de gradiente, lo cuál limita la capacidad de estos modelos para aprender de series temporales de largo plazo. Para abordar este problema, se han desarrollado arquitecturas más avanzadas como las redes LSTM (Long Short-Term Memory) y GRU (Gated Recurrent Unit), que están diseñadas específicamente para manejar mejor las dependencias a largo plazo, estas agregan una puerta de olvido que permite a la red neuronal solo utilizar la información necesaria y olvidar la innecesaria.

#### 2.1.6. Redes neuronales con Arquitectura transformer

Los transformers, aunque efectivos en muchas tareas de procesamiento de secuencias, no son ideales para la predicción de series de tiempo financieras debido a varias

limitaciones. En primer lugar, su complejidad dificulta la interpretabilidad, esencial en el ámbito financiero. Además, tienen una alta propensión al sobreajuste debido a su gran cantidad de parámetros, lo que puede resultar en un rendimiento deficiente con datos no vistos. Requieren grandes volúmenes de datos, lo cual es problemático en el contexto financiero donde los datos pueden ser limitados y obsoletos rápidamente. Su alta demanda computacional es otro inconveniente, especialmente para aplicaciones en tiempo real. Aunque buenos para capturar dependencias a largo plazo, los transformers pueden tener dificultades con los patrones cíclicos y estacionales típicos de las series financieras, y son sensibles al ruido característico de estos datos. Por último, la incorporación de conocimiento específico del dominio financiero es más compleja en transformers que en modelos tradicionales de series temporales.

### **2.1.7. Análisis de Sentimientos de las Noticias, teniendo en cuenta su relevancia**

Haciendo un Análisis de sentimiento por las noticias (con un modelo pre entrenado) podríamos determinar si esta habla bien o mal de la moneda en cuestión , esto en principio debería verse reflejado en un alza o decaimiento del valor de la moneda, este coeficiente de afectación del contenido de la noticia respecto a la moneda se vería limitado o potenciado por la relevancia de la noticia, siguiendo alguna métrica como la cantidad de lectores que tenga la cadena que la notifica , métricas de audiencia y alcance de las mismas etc. Con esto podríamos introducir este nivel de afectación de la noticia al valor de la moneda en la serie de tiempo del movimiento de precios de la moneda o activo del mercado a analizar. Entonces con un modelo LSTM que tenga como entrada todos estos datos podríamos predecir efectivamente el valor de la moneda. Este enfoque es muy convincente y se encontró en la literatura trabajos que abordan el problema de forma muy similar. Pero haciendo esto agregamos el sesgo de afirmar que una noticia que hable bien sobre una moneda o activo haga que este aumente o decremente su valor de mercado. También se incluye la problemática de encontrar las métricas reales de alcance de estas noticias , podría utilizarse una simulación para obtener estos parámetros de forma aproximada, pero esto agregaría el error cometido por la simulación al no tratarse de datos reales, lo cuál lo hace poco fiable. En trabajos futuros se podría utilizar nuestro anterior trabajo simulador de redes sociales, para ser utilizado en la generación de dichas métricas.

## Capítulo 3

# Detalles de Implementación y Experimentos

### 3.1. Ideas Iniciales

#### 3.1.1. Obtención y análisis de los datos

Para obtener datos financieros con información relevante como el precio de apertura ( El precio de las primeras operaciones realizadas en el día , coincide con el precio de cierre del día anterior), Precio de cierre ( Precio de las últimas operaciones realizadas en el día ) y los precios más altos y más bajos que parecen a lo largo del día. Este dataset lo obtuvimos de Kaggle Yahoo Stock Prediction. Haciendo un análisis de estos datos encontramos que tiene 7 columnas (Open, High, Low, Close, Volume, Dividends, Stock Splits) y 6165 filas

#### 3.1.2. Análisis de los valores del dataset

Haciendo un análisis sobre los datos presentes en el dataset llegamos a que no hay datos cuyo valor sea nulo, También encontramos que todos los datos son numéricos. Para evitar insertar ruido en los datos eliminamos los features Dividends y Stock Splits , pues estos no aportan información relevante para la predicción y solo agregan ruido a los datos.

#### 3.1.3. Análisis de la distribución de los datos

Hallando la asimetría o sesgo de los datos (Skewness) para detectar anomalías en la distribución de los datos

Shape: (6165, 7)

|                           | Open       | High       | Low        | Close      | Volume    | Dividends | Stock Splits |
|---------------------------|------------|------------|------------|------------|-----------|-----------|--------------|
| date                      |            |            |            |            |           |           |              |
| 2000-01-03 00:00:00-05:00 | 0.791669   | 0.849227   | 0.767607   | 0.844981   | 535796800 | 0.0       | 0.0          |
| 2000-01-04 00:00:00-05:00 | 0.817145   | 0.835073   | 0.763833   | 0.773740   | 512377600 | 0.0       | 0.0          |
| 2000-01-05 00:00:00-05:00 | 0.783176   | 0.834601   | 0.777515   | 0.785063   | 778321600 | 0.0       | 0.0          |
| 2000-01-06 00:00:00-05:00 | 0.801105   | 0.807709   | 0.717125   | 0.717125   | 767972800 | 0.0       | 0.0          |
| 2000-01-07 00:00:00-05:00 | 0.728448   | 0.762417   | 0.720900   | 0.751094   | 460734400 | 0.0       | 0.0          |
| ...                       | ...        | ...        | ...        | ...        | ...       | ...       | ...          |
| 2024-06-28 00:00:00-04:00 | 215.770004 | 216.070007 | 210.300003 | 210.619995 | 82542700  | 0.0       | 0.0          |
| 2024-07-01 00:00:00-04:00 | 212.089996 | 217.509995 | 211.919998 | 216.750000 | 60402900  | 0.0       | 0.0          |
| 2024-07-02 00:00:00-04:00 | 216.149994 | 220.380005 | 215.100006 | 220.270004 | 58046200  | 0.0       | 0.0          |
| 2024-07-03 00:00:00-04:00 | 220.000000 | 221.550003 | 219.029999 | 221.550003 | 37369800  | 0.0       | 0.0          |
| 2024-07-05 00:00:00-04:00 | 221.649994 | 226.449997 | 221.649994 | 226.339996 | 58797569  | 0.0       | 0.0          |

6165 rows x 7 columns

Figura 3.1: Data Set escogido

Index: 6165 entries, 2000-01-03 00:00:00-05:00

Data columns (total 7 columns):

| # | Column       | Non-Null Count | Dtype   |
|---|--------------|----------------|---------|
| 0 | Open         | 6165 non-null  | float64 |
| 1 | High         | 6165 non-null  | float64 |
| 2 | Low          | 6165 non-null  | float64 |
| 3 | Close        | 6165 non-null  | float64 |
| 4 | Volume       | 6165 non-null  | int64   |
| 5 | Dividends    | 6165 non-null  | float64 |
| 6 | Stock Splits | 6165 non-null  | float64 |

dtypes: float64(6), int64(1)

memory usage: 514.4+ KB

Figura 3.2: Análisis de la nulidad en los datos

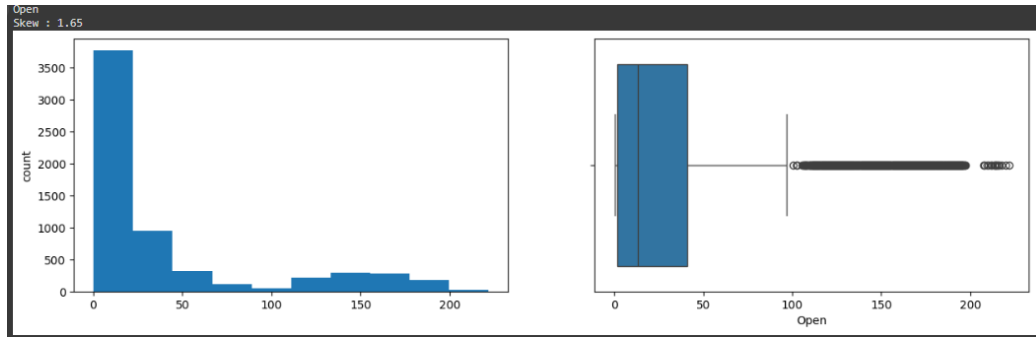


Figura 3.3: Distribución de la característica Open

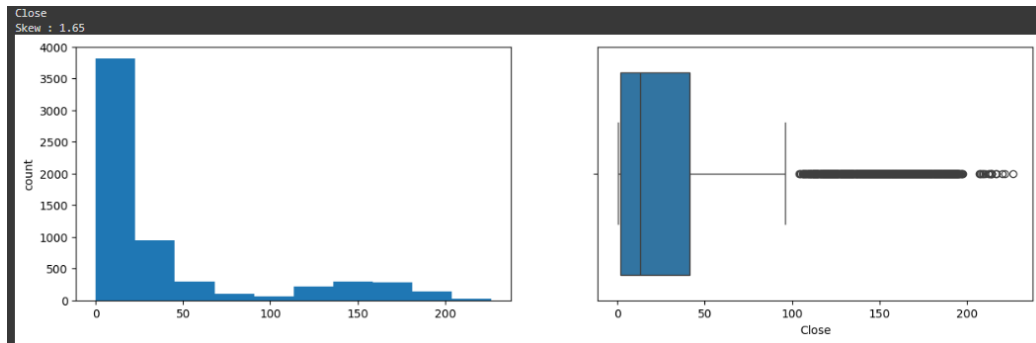


Figura 3.4: Distribución de la característica Closed

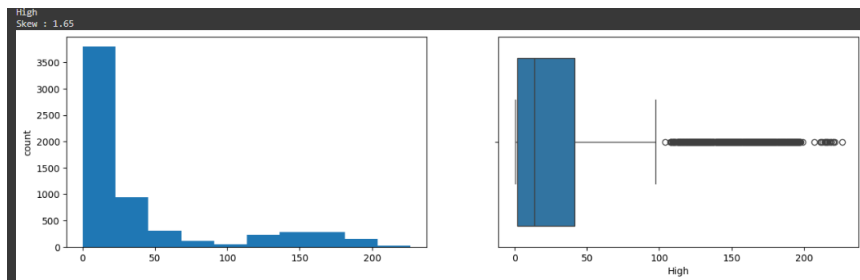


Figura 3.5: Distribución de la característica High

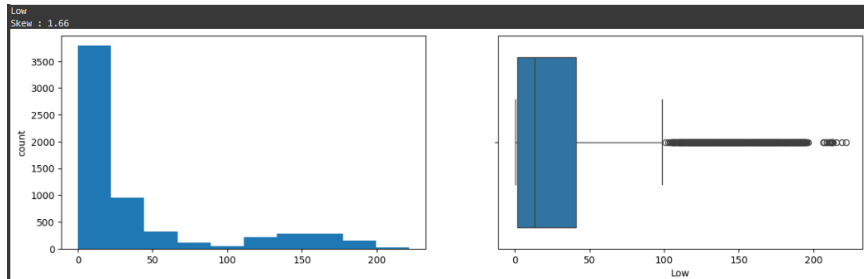


Figura 3.6: Distribución de la característica Low

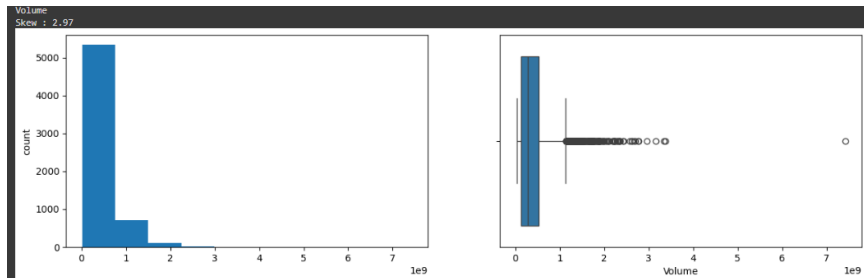


Figura 3.7: Distribución de la característica Volume



Figura 3.8: Comportamiento de la característica Close a lo largo del tiempo



Podemos observar en los datos un sesgo positivo; los valores en su mayoría son bajos y hay una cola larga que se extiende a la derecha. lo cuál indica que hay un número significativo de valores altos en la distribución, pero con menor frecuencia. El gráfico de caja y bigotes (boxplot) a la derecha muestra que la mediana está cerca del primer cuartil, con una cola larga extendiéndose hacia valores más altos. En la mayoría de casos el sesgo fluctúa alrededor de 1.65 lo cuál indica una asimetría hacia valores altos, Esto sugiere que aunque hay una gran cantidad de valores bajos, hay valores muy altos que están afectando la media y extendiendo la cola de distribución, por lo que es necesario normalizar los datos.

#### **3.1.4. Análisis bivariable de los datos**

Con este Análisis pretendemos encontrar relaciones entre las diferentes características presentes en los datos para encontrar así relaciones, ruidos, etc.

##### **Utilizando un grafico de pares para ver relaciones entre las diferentes variables**

En general la relacion entre las variables es lineal , notando no linealidades sobre todo en variables relacionadas con el volumen de las operaciones

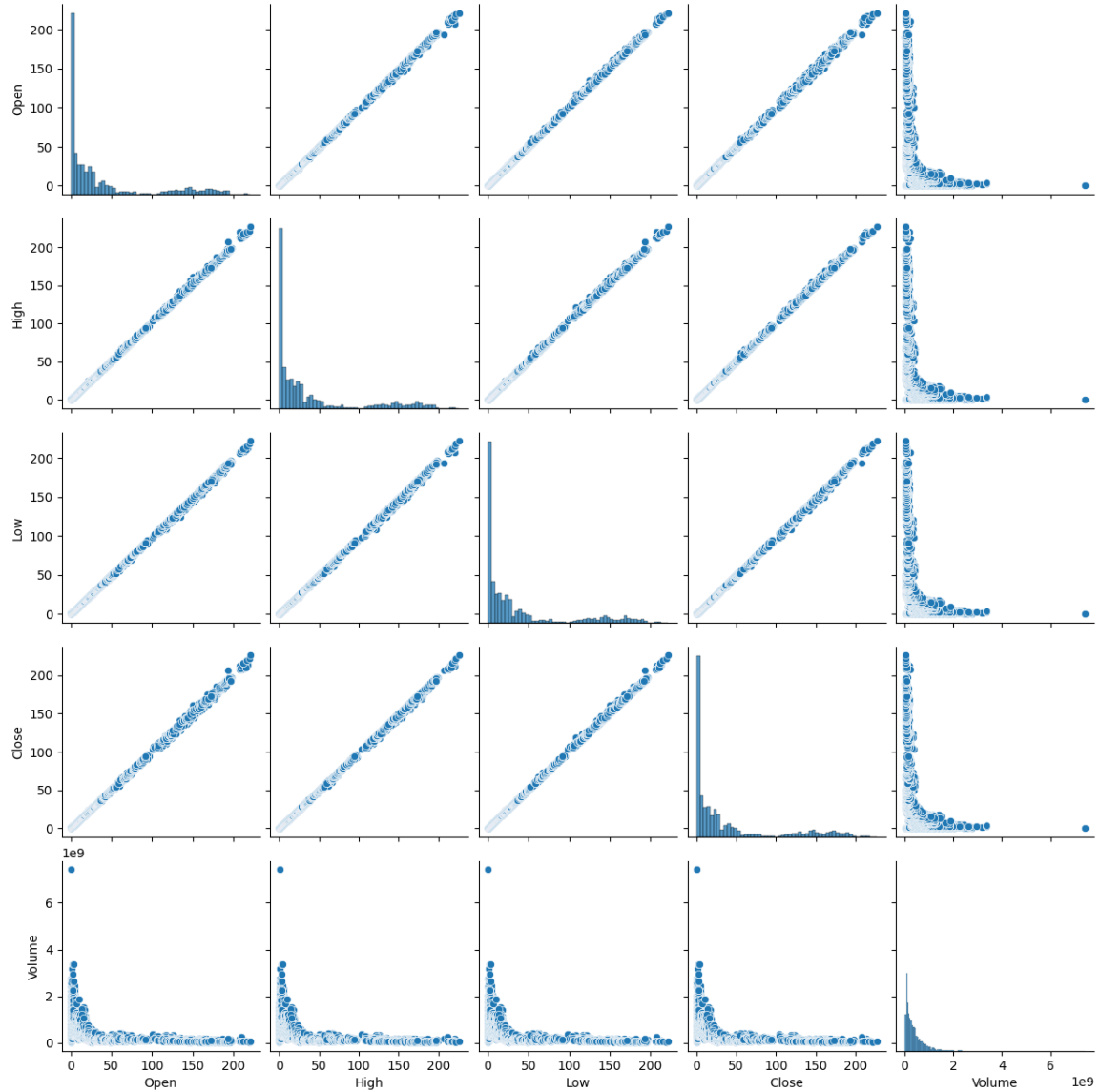


Figura 3.9: Gráfico de pares con los datos

Analizando la gráfica de pares llegamos a la conclusión de la mayoría de datos tienen relaciones lineales entre sí, salvo por el volumen el cuál presenta relaciones no lineales respecto al resto de variables.

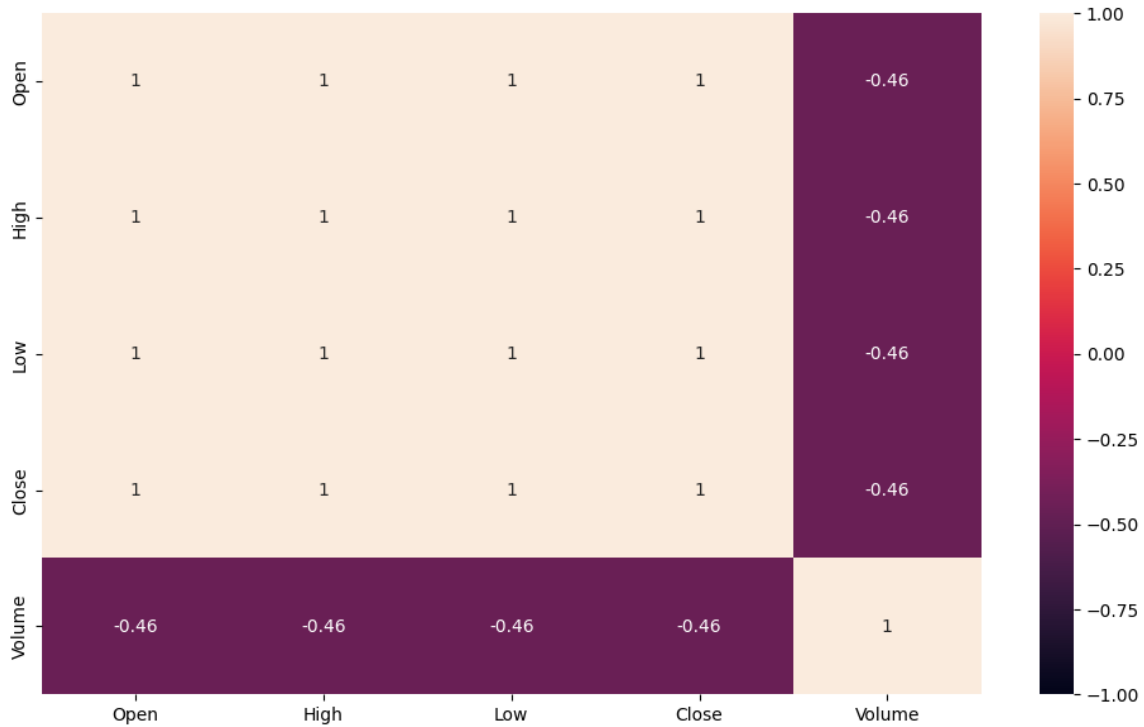
**Análisis de correlación entre los datos utilizando un mapa de calor**

Figura 3.10: Mapa de calor de las variables del dataset

Es notable la correlación negativa del volumen al resto de características capturadas en el dataset, mientras que el resto mantienen una correlación lineal entre sí. Esto corrobora la información sacada de la gráfica de pares analizada anteriormente

### 3.1.5. Detección de Outlayers

#### Utilizando Rango Intercuartílico (IQR)

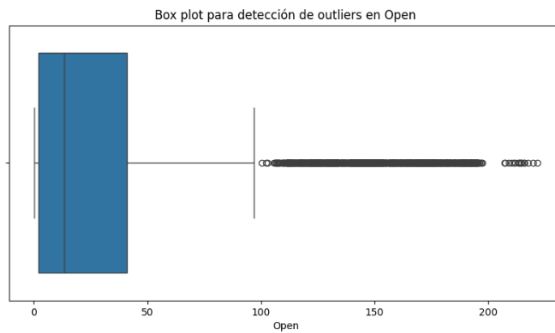


Figura 3.11: Detección de Outliers en Open

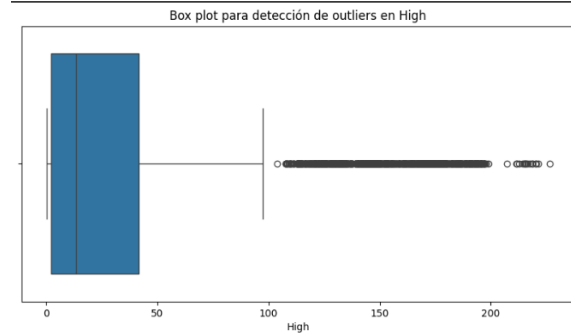


Figura 3.12: Detección de Outliers en High

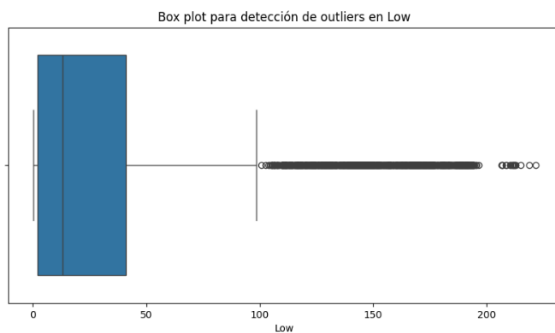


Figura 3.13: Detección de Outliers en Low

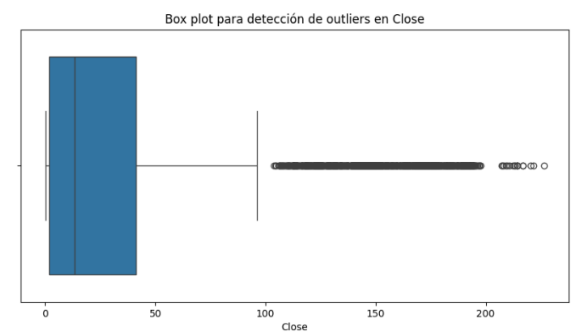


Figura 3.14: Detección de Outliers en Close

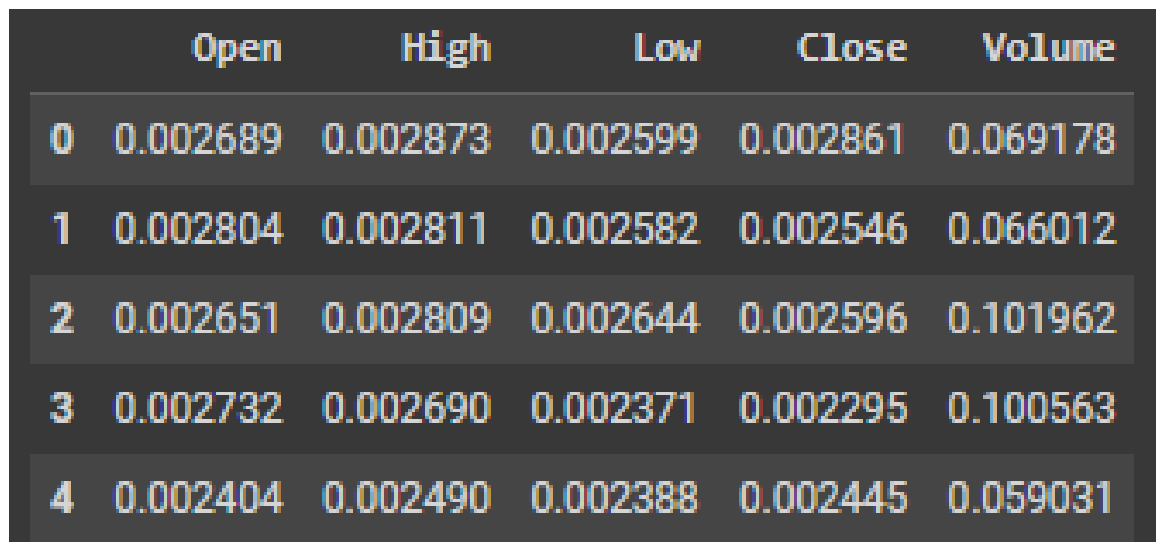
Analizando la gráfica de bigote notamos una gran cantidad de valores moviéndose a la derecha de cada dato, alejándose de la media, lo que indica una gran cantidad de valores desproporcionados al resto, mucho mayores que el resto, este comportamiento es común dada la naturaleza de estos datos financieros la cuál tiende a ser muy volátil y dinámica en el tiempo. Las variables Open, High, Low y Close parecen tener distribuciones similares, todas con un número significativo de outliers en el extremo superior. Esto sugiere que las variaciones extremas en los precios pueden estar correlacionadas entre estas diferentes medidas.

### 3.1.6. Limpieza y normalización de los datos

Para Mantener los datos normalizados exploramos varias alternativas de métodos de normalización

#### MinMax Scaling

Escala los datos Para Mantenerlos en un rango específico, en este caso (0,1). Tiene el problema de ser susceptible a los outliers



|   | Open     | High     | Low      | Close    | Volume   |
|---|----------|----------|----------|----------|----------|
| 0 | 0.002689 | 0.002873 | 0.002599 | 0.002861 | 0.069178 |
| 1 | 0.002804 | 0.002811 | 0.002582 | 0.002546 | 0.066012 |
| 2 | 0.002651 | 0.002809 | 0.002644 | 0.002596 | 0.101962 |
| 3 | 0.002732 | 0.002690 | 0.002371 | 0.002295 | 0.100563 |
| 4 | 0.002404 | 0.002490 | 0.002388 | 0.002445 | 0.059031 |

Figura 3.15: Resultados de la normalización con MinMax

#### Standard Scaler

Transforma los datos para que tengan una media de 0 y una desviación estándar de 1. Menos sensible a outliers que Min-Max Scaling

|   | Open      | High      | Low       | Close     | Volume   |
|---|-----------|-----------|-----------|-----------|----------|
| 0 | -0.673803 | -0.672878 | -0.673852 | -0.672678 | 0.372172 |
| 1 | -0.673336 | -0.673134 | -0.673922 | -0.673982 | 0.311264 |
| 2 | -0.673959 | -0.673143 | -0.673669 | -0.673775 | 1.002914 |
| 3 | -0.673630 | -0.673630 | -0.674786 | -0.675018 | 0.976000 |
| 4 | -0.674961 | -0.674451 | -0.674717 | -0.674396 | 0.176954 |

Figura 3.16: Resultado de la normalización usando Standard Scaler

### Transformación logarítmica

La transformación logarítmica permite reducir los sesgos y hacer los datos más normales, lo que ayuda a manejar los outliers, esta normalización será la que utilizaremos en nuestra implementación

### 3.1.7. Análisis de los outliers después de normalizar con la transformación logarítmica

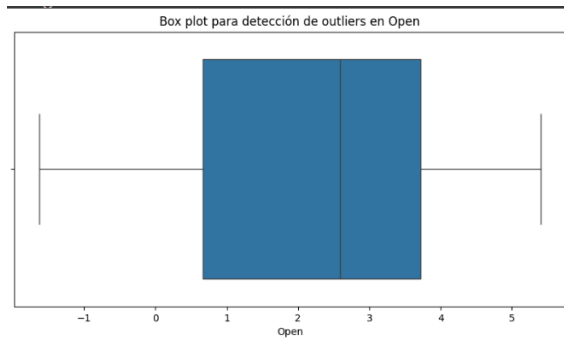


Figura 3.17

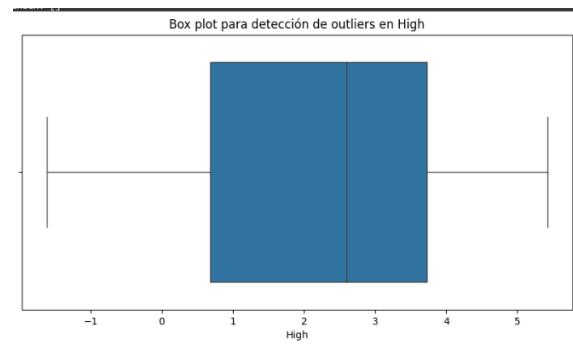


Figura 3.18

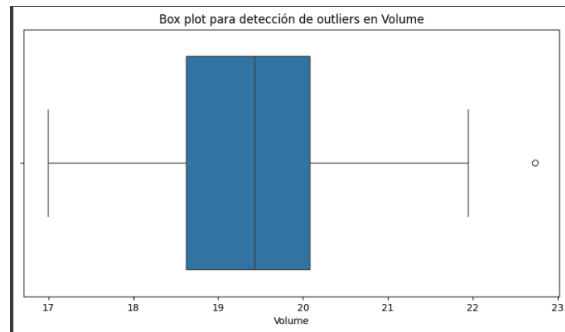


Figura 3.19

se puede notar como se reduce la cantidad de outliers luego de la normalización y se obtiene una distribución más normal de los datos

## Distribución de los datos luego de normalizar

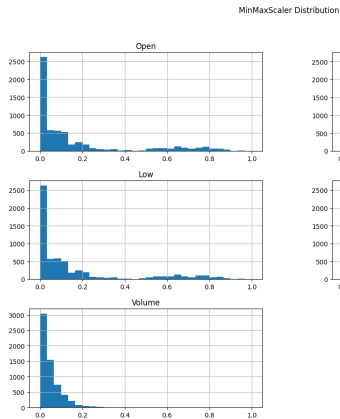


Figura 3.20: Distribución de los datos después de MinMax

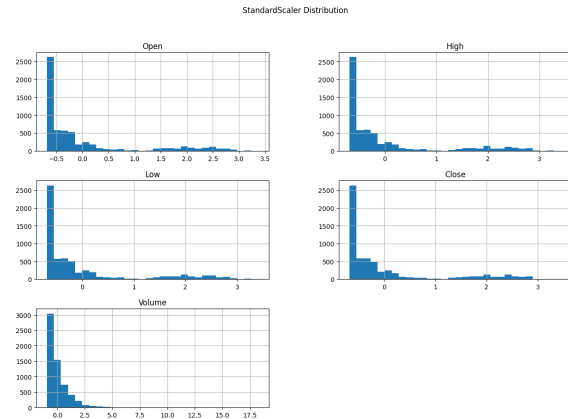


Figura 3.21: Distribución de los datos después de Standard Scaler

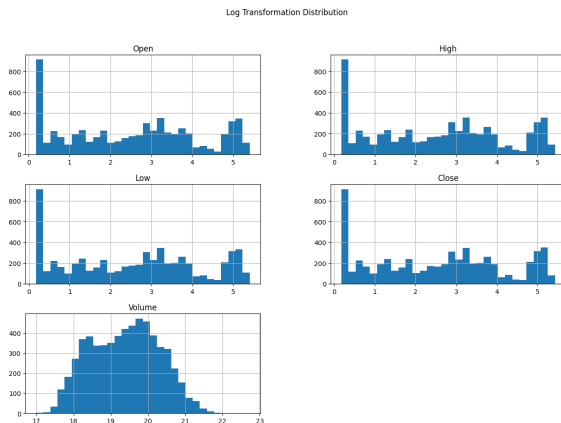


Figura 3.22: Distribución de los datos después de Transformación Logarítmica

La transformación logarítmica resuelve el problema del sesgo en los datos y los outliers. Lo cual hace que los datos tengan escalas semejantes y elimina el ruido que agregan los datos con tamaños muy grandes que es el principal problema que nos habíamos encontrado.

## 3.2. Análisis del dataset de noticias

El dataset de noticias en nuestro caso, de igual manera fue extraído de Kaggle y contiene noticias de los últimos años de la empresa tecnológica Apple. EL dataset



contiene 12 columnas y 15975 entradas. Para nuestro trabajo sólo nos quedaremos con el contenido de la noticia que es el que queremos vincular al movimiento de los precios. El resto de columnas son valores que no tienen sentido para este trabajo como Label, Ticket, Category que no aportan información relevante. También hay dato que redundan como el Precio, El Volumen etc.

| Unnamed: 0 | ticker | category | title   |   | content  | Open      | High      | Low       | Close     | Adj Close | Volume    | label |
|------------|--------|----------|---------|---|--|-----------|-----------|-----------|-----------|-----------|-----------|-------|
| Date       |        |          |         |   |  |           |           |           |           |           |           |       |
| 2020-01-27 | 0      | AAPL     | opinion | Apple Set To Beat Q1 Earnings Estimates Tech ...  | Technology giant Apple NASDAQ AAPL is set ...    | 77.514999 | 77.942497 | 76.220001 | 77.237503 | 75.793358 | 161940000 | 0     |
| 2020-01-27 | 1      | AAPL     | opinion | Tech Daily Intel Results Netflix Surge Appl...    | The top stories in this digest are Intel's N...  | 77.514999 | 77.942497 | 76.220001 | 77.237503 | 75.793358 | 161940000 | 0     |
| 2020-01-27 | 2      | AAPL     | opinion | 7 Monster Stock Market Predictions For The Wee... | S P 500 SPY \nThis week will be packed with e... | 77.514999 | 77.942497 | 76.220001 | 77.237503 | 75.793358 | 161940000 | 0     |

Figura 3.23: Dataset de noticias

### Eliminación de datos innecesarios y redundantes

Notamos que todas las fechas estaban en tipo de dato Object y que a diferencia de las fechas en los datos de los precios estas no tenían la hora del día, por lo que fue necesario transformar estos datos para poder así vincular cada fecha en que el activo se investigó con su respectiva noticia vigente en ese día. También nos encontramos con muchas fechas si noticias y muchas noticias repetidas a lo largo del dataset, las cuáles eliminamos para evitar así redundancias en los datos.

```
Cantidad de noticias: 15975
Cantidad de fechas con noticias: 1654
Noticias sobrantes: 14321
```

Figura 3.24: Fechas y noticias redundantes

Finalmente nos quedamos con un dataset de 1654 entradas que solo contiene la fecha y la noticia en sí

### Procesamiento del contenido de las noticias con técnicas de NLP

Para procesar las noticias empleamos técnicas de NLP como la eliminación de stopwords, signos de puntuación, tokenización de los contenidos de las noticias y Entrenamos un Modelo Word2Vect para así obtener una representación de embedding de el contenido de dichas noticias. Esto nos permite una representación numérica y cómoda para entrenar nuestro modelo y una forma de capturar relaciones más complejas entre los precios y el contenido de las noticias.

| Date                  |   |
|-----------------------|---|
| 2012-07-23            | Summer Heat Scorches Europe And U S Europe fl...  |
| 2012-07-24            | Market Bait And Switch That is the sound we ar... |
| 2012-07-27            | Will AAPL Fall From The Tree Apple s AAPL ...     |
| 2012-07-30            | Bulls Snatch Victory From Jaws of Defeat Last ... |
| 2012-07-31            | What s Driving China s Real Estate Rally Par...   |
| ...                   | ...   |
| 2020-01-21            | How to download Apple Card data into a spreads... |
| 2020-01-22            | Dow Jones News IBM Reports Strong Results Ap...   |
| 2020-01-23            | Apple Boosts Chip Orders From Main Foundry Sup... |
| 2020-01-24            | Fiscal policies of main Irish parties vying fo... |
| 2020-01-27            | Apple Set To Beat Q1 Earnings Estimates Tech ...  |
| 1654 rows × 1 columns |   |

Figura 3.25: Dataset Resultante

|            |   | new   | processed  | embedding |
|------------|---|---|--|-----------|
| Date       |   |   |  |           |
| 2012-07-23 | Summer Heat Scorches Europe And U S Europe fl...  | [summer, heat, scorches, europe, u, europe, fl... | [0.3377519, 0.84812295, 0.7480908, -0.22768843,    |           |
| 2012-07-24 | Market Bait And Switch That is the sound we ar... | [market, bait, switch, sound, going, hear, soo... | [0.7784091, 0.16130532, 0.5994031, -0.6621038,...  |           |
| 2012-07-27 | Will AAPL Fall From The Tree Apple s AAPL ...     | [aapl, fall, tree, apple, aapl, sales, third, ... | [-0.0186733, 0.32305655, 0.29652885, -0.745922...  |           |
| 2012-07-30 | Bulls Snatch Victory From Jaws of Defeat Last ... | [bulls, snatch, victory, jaws, defeat, last, w... | [0.4676175, 0.6291028, 0.59835887, -0.5777845,...  |           |
| 2012-07-31 | What s Driving China s Real Estate Rally Par...   | [driving, china, real, estate, rally, part, 3,... | [0.553934, 0.73153806, 0.6304709, -0.3310698, ...  |           |
| ...        | ...   | ...   | ...  |           |
| 2020-01-21 | How to download Apple Card data into a spreads... | [download, apple, card, data, spreadsheet, app... | [0.21420306, 0.723705, 0.5310207, -0.27709982,...  |           |
| 2020-01-22 | Dow Jones News IBM Reports Strong Results Ap...   | [dow, jones, news, ibm, reports, strong, resul... | [0.31232607, 0.7350899, 0.6199543, -0.386247, ...  |           |
| 2020-01-23 | Apple Boosts Chip Orders From Main Foundry Sup... | [apple, boosts, chip, orders, main, foundry, s... | [0.20302458, 0.86192226, 0.59886396, -0.296175...  |           |
| 2020-01-24 | Fiscal policies of main Irish parties vying fo... | [fiscal, policies, main, irish, parties, vying... | [0.43426958, 0.66431063, 0.5781885, -0.483632,...  |           |
| 2020-01-27 | Apple Set To Beat Q1 Earnings Estimates Tech ...  | [apple, set, beat, q1, earnings, estimates, te... | [0.35244656, 0.7231707, 0.62173194, -0.4401080,... |           |

Figura 3.26: Resultado de aplicación de técnica nlp al dataset de noticias

### 3.2.1. Clusterización de las noticias

Con el objetivo de observar el agrupamiento de las noticias dado su contenido y cuán distantes o agrupadas están unas de otras decidimos crear clusters con el objetivo de observar esto. Utilizamos el algoritmo K-Means con diferentes k. Para determinar cuál sería el valor óptimo del hiperparámetro K hicimos un análisis del índice de silueta.

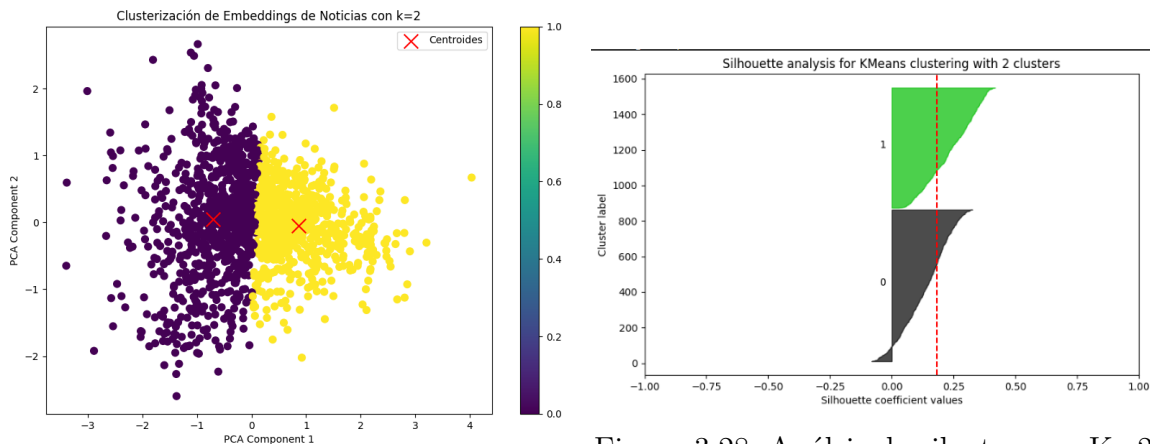


Figura 3.27: Clusters para K=2

Figura 3.28: Análisis de silueta para K=2

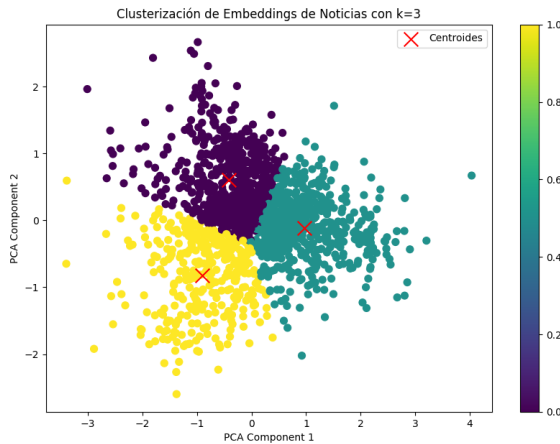


Figura 3.29: Clusters para K=3

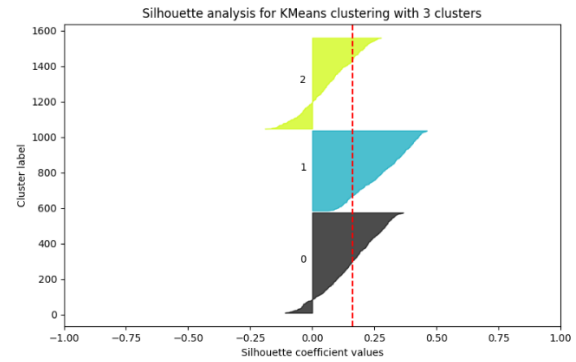


Figura 3.30: Análisis de silueta para K=3

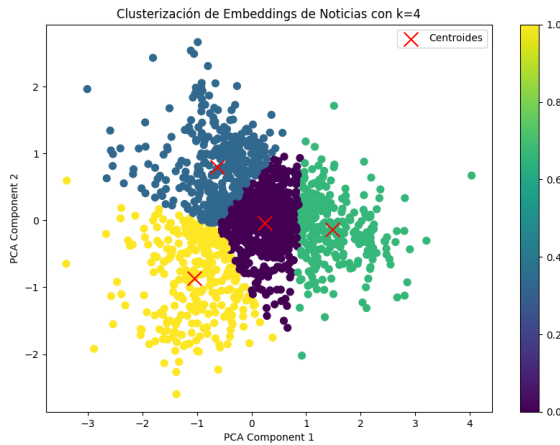


Figura 3.31: Clusters para K=4

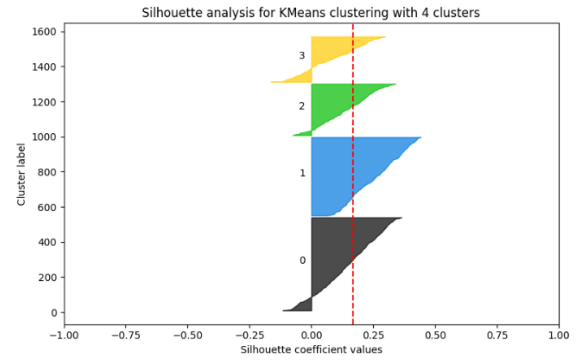


Figura 3.32: Análisis de silueta para K=4

Después de este análisis llegamos a la conclusión de que la mejor forma de clustear el conjunto de noticias es con 2 clusters, pues mantiene consistente la distancia media de los elementos del cluster. En un análisis posterior de los centroides pretendíamos analizar el contenido de los centroides para determinar si eran noticias que afectaban positiva o negativamente a los precios de las noticias y así clasificar las noticias en base a esto, por cuestiones ya mencionadas esta idea fue descartada en el presente trabajo.

Por cuestiones de tiempo y complejidad de estos algoritmos descartamos la idea de hacer un análisis utilizando modelos como DBSCAN o HDBSCAN que serían más

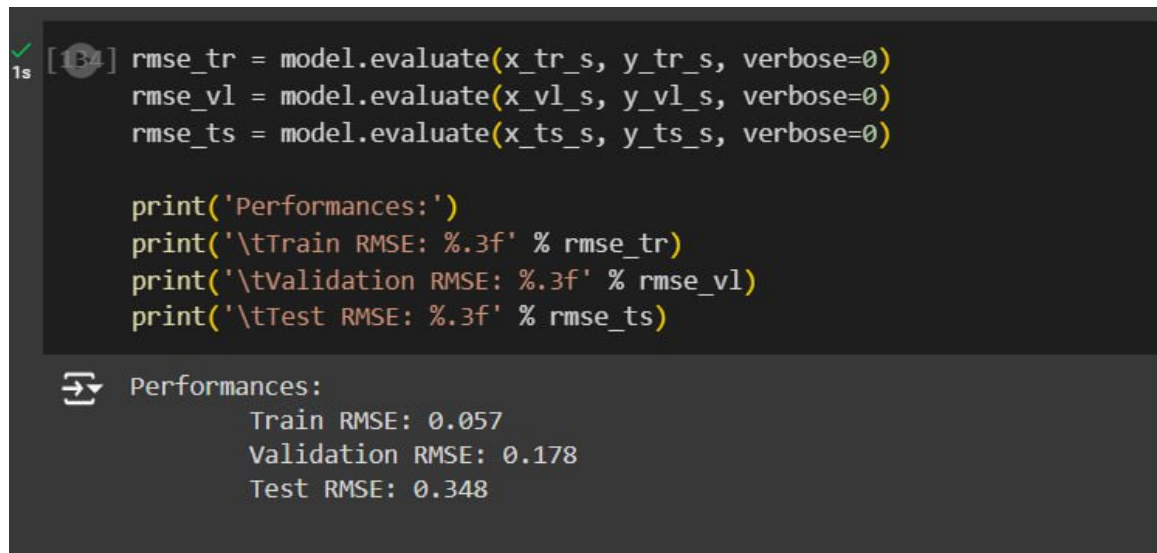
adecuados dada la naturaleza no lineal de los datos textuales como el contenido de las noticias y su capacidad de encontrar grupos basandose en la densidad y proximidad que sería un enfoque beneficioso ya que las noticias que tratan temas semejantes entre sí tienden a estar agrupadas pues usan regularmente palabras muy similares, cosa que los embeddings son capaces de captar en los vectores resultantes. También temas muy presentes en los datos , tienden a estar más concentrados, creando puntos de alta densidad, mientras que temas menos tratados tienden a estar más dispersos.

### 3.3. Entrenamiento y diseño del Modelo

El modelo escogido es una red neuronal recurrente usando la arquitectura LSTM que permite un mejor análisis de series temporales. Hicimos varias pruebas entrenando al modelo con diferentes conjuntos del dataset, agregando datos extras como los indicadores EMA.

Inicialmente Tenemos un Modelo LSTM cuyos datos de entrada sólo pertenecen a los registros históricos de movimiento del activo que queremos predecir, este modelo tiene 128 neuronas en las capas ocultas y un batch size de 256

#### 3.3.1. Primera Iteración : Entrenando con los datos Close y Volumen



```
[134]: rmse_tr = model.evaluate(x_tr_s, y_tr_s, verbose=0)
      rmse_vl = model.evaluate(x_vl_s, y_vl_s, verbose=0)
      rmse_ts = model.evaluate(x_ts_s, y_ts_s, verbose=0)

      print('Performances:')
      print('\tTrain RMSE: %.3f' % rmse_tr)
      print('\tValidation RMSE: %.3f' % rmse_vl)
      print('\tTest RMSE: %.3f' % rmse_ts)
```

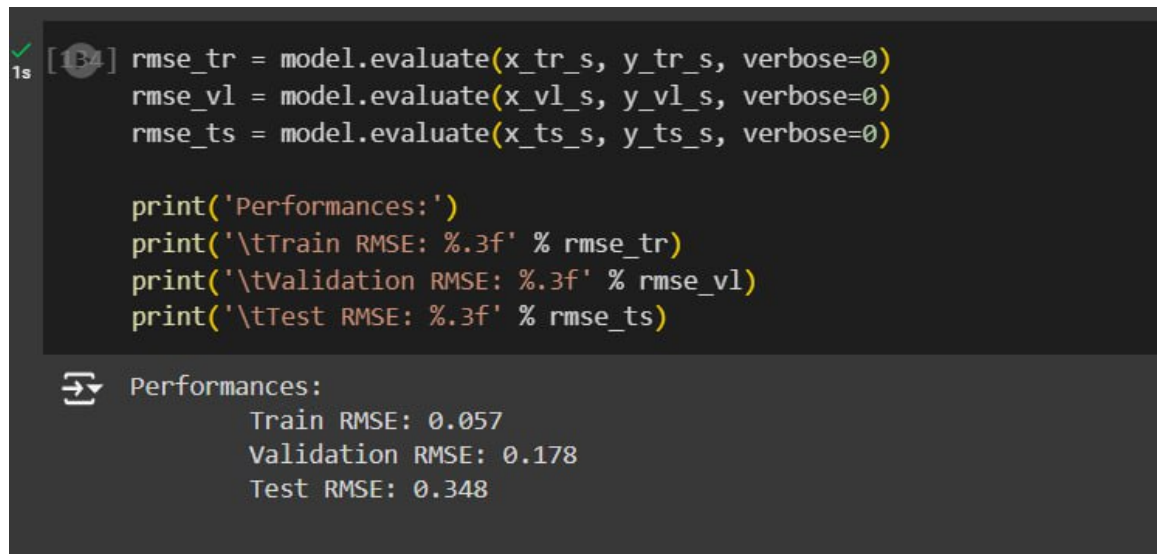
Performances:

|                  |       |
|------------------|-------|
| Train RMSE:      | 0.057 |
| Validation RMSE: | 0.178 |
| Test RMSE:       | 0.348 |

Figura 3.33: Resultados del entrenamiento

Tiene una buena consistencia entre los datos de entrenamiento, aunque aún no vincula la información relacionada con las noticias.

### 3.3.2. Segunda Iteración: Entrenando Agregando un Indicador EMA



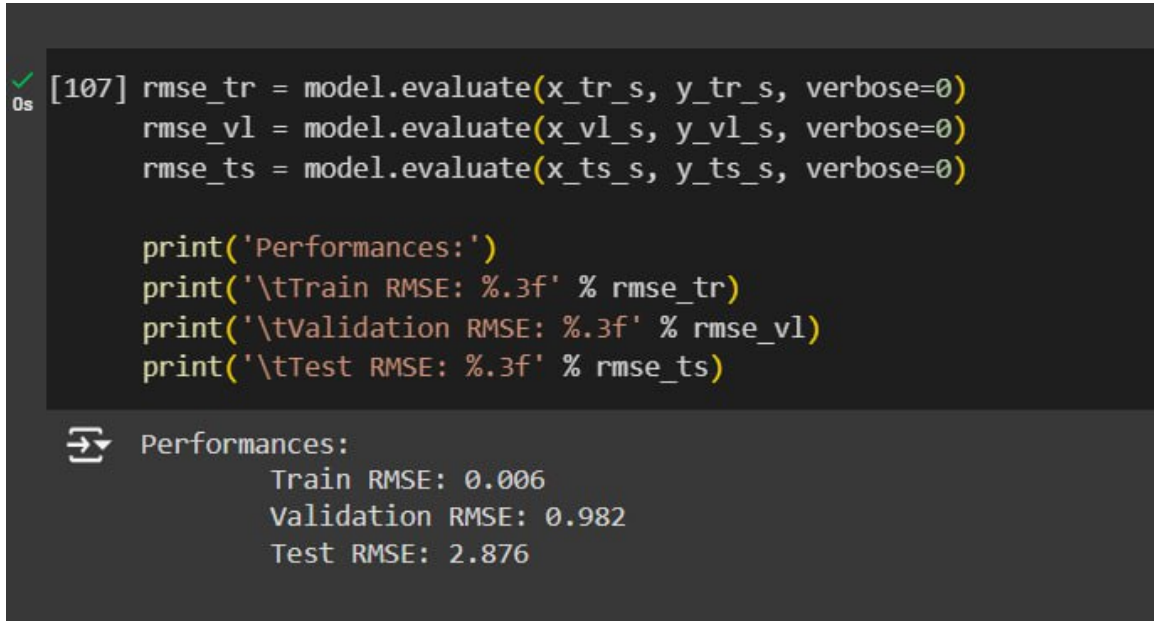
```
[134]: rmse_tr = model.evaluate(x_tr_s, y_tr_s, verbose=0)
rmse_vl = model.evaluate(x_vl_s, y_vl_s, verbose=0)
rmse_ts = model.evaluate(x_ts_s, y_ts_s, verbose=0)

print('Performances:')
print('\tTrain RMSE: %.3f' % rmse_tr)
print('\tValidation RMSE: %.3f' % rmse_vl)
print('\tTest RMSE: %.3f' % rmse_ts)
```

⇒ Performances:  
Train RMSE: 0.057  
Validation RMSE: 0.178  
Test RMSE: 0.348

Figura 3.34

Aquí se nota una diferencia entre el resultado con el dataset de entrenamiento y de prueba, por lo que concluimos que el indicador EMA introducido está cegando los datos y añadiendo ruido.



```
[107] rmse_tr = model.evaluate(x_tr_s, y_tr_s, verbose=0)
      rmse_vl = model.evaluate(x_vl_s, y_vl_s, verbose=0)
      rmse_ts = model.evaluate(x_ts_s, y_ts_s, verbose=0)

      print('Performances:')
      print('\tTrain RMSE: %.3f' % rmse_tr)
      print('\tValidation RMSE: %.3f' % rmse_vl)
      print('\tTest RMSE: %.3f' % rmse_ts)
```

↵ Performances:  
Train RMSE: 0.006  
Validation RMSE: 0.982  
Test RMSE: 2.876

Figura 3.35

### 3.3.3. Iteración 3 utilizando todas las columnas del dataset (Close, Open, High, Low, Volume)

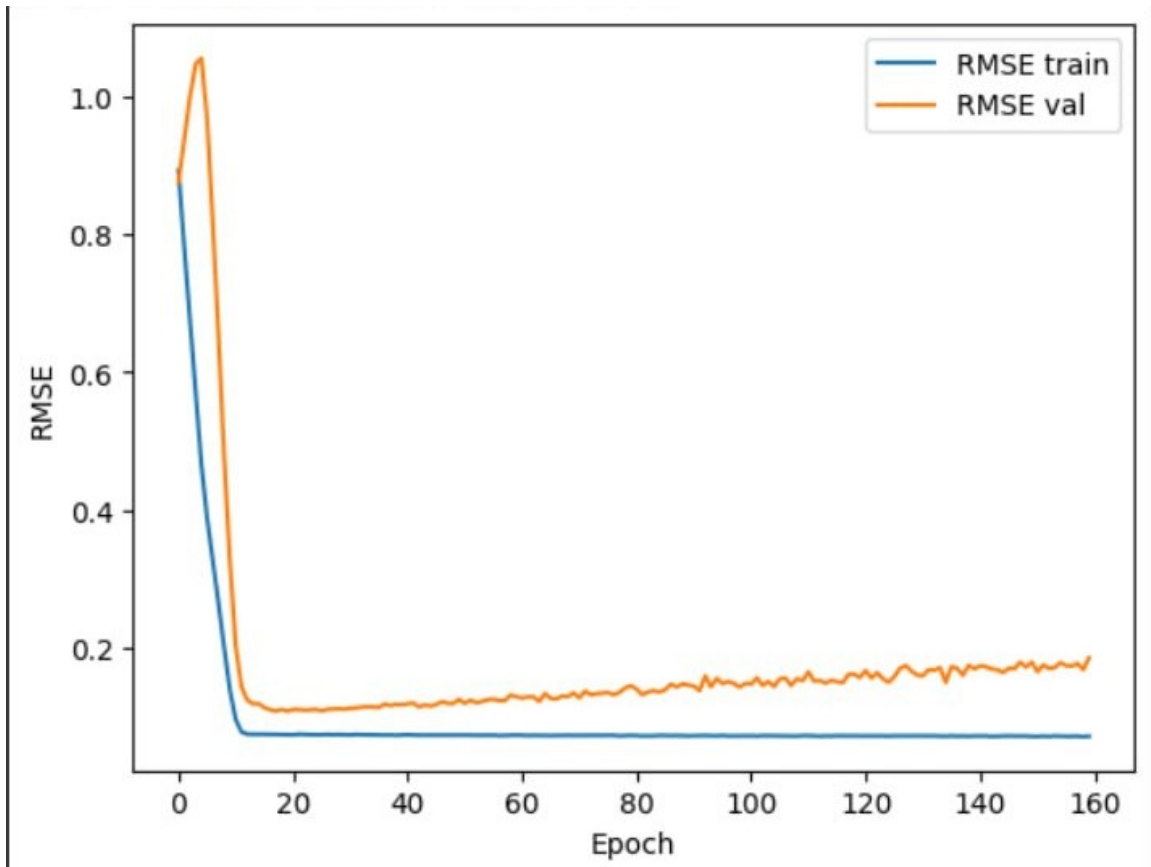


Figura 3.36: Resultados del entrenamiento



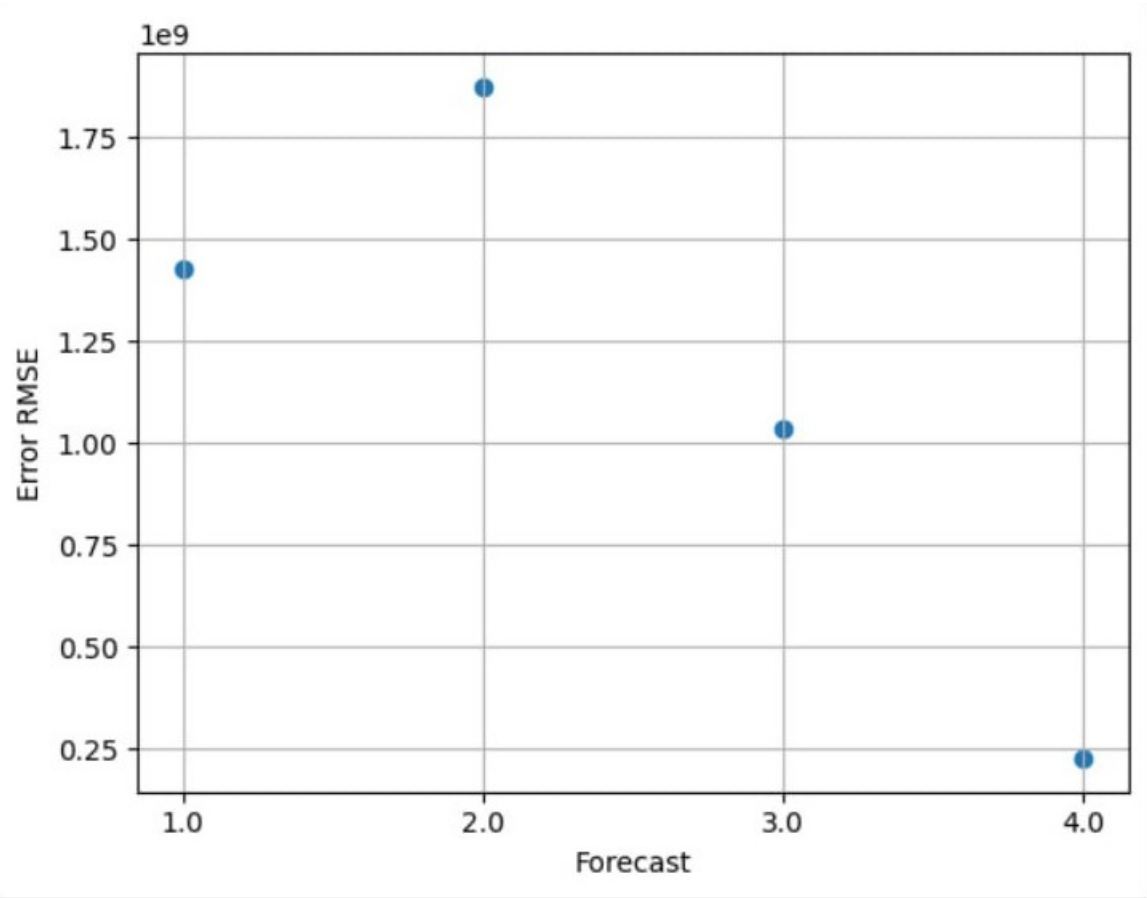
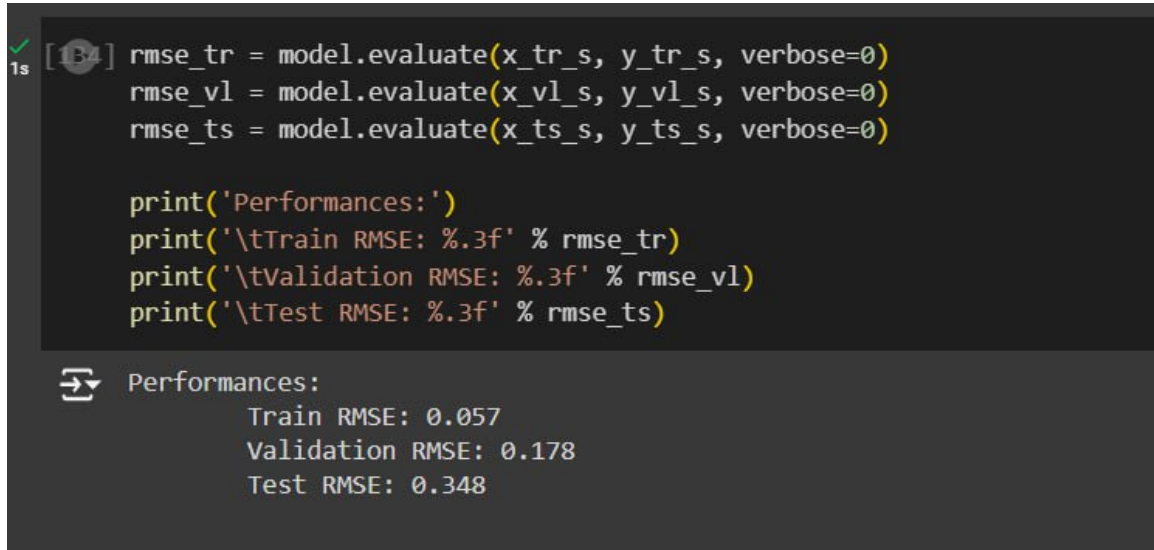


Figura 3.37: Errores después del entrenamiento



```
✓ [134] rmse_tr = model.evaluate(x_tr_s, y_tr_s, verbose=0)
rmse_vl = model.evaluate(x_vl_s, y_vl_s, verbose=0)
rmse_ts = model.evaluate(x_ts_s, y_ts_s, verbose=0)

print('Performances:')
print('\tTrain RMSE: %.3f' % rmse_tr)
print('\tValidation RMSE: %.3f' % rmse_vl)
print('\tTest RMSE: %.3f' % rmse_ts)
```

⇒ Performances:  
Train RMSE: 0.057  
Validation RMSE: 0.178  
Test RMSE: 0.348

Figura 3.38

### 3.3.4. Iteración 4 (Añadiendo noticias)

Para tratar de encontrar la relación semántica entre la evolución del precio y las palabras o contenido de las noticias, utilizamos WordToVect para hallar el embedding de cada noticia, tomamos cada componente del embedding como un feature. Nuestro modelo es ahora una red Neuronal con arquitectura LSTM con un vector de entrada de tamaño 105, 256 neuronas en las capas ocultas y un learning rate de  $5 \times 10^{-11}$ . Hicimos una separación 80, 10, 10, donde 80% es entrenamiento 10% es validación y el otro 10% es de test. Obtuvimos los resultados siguientes:

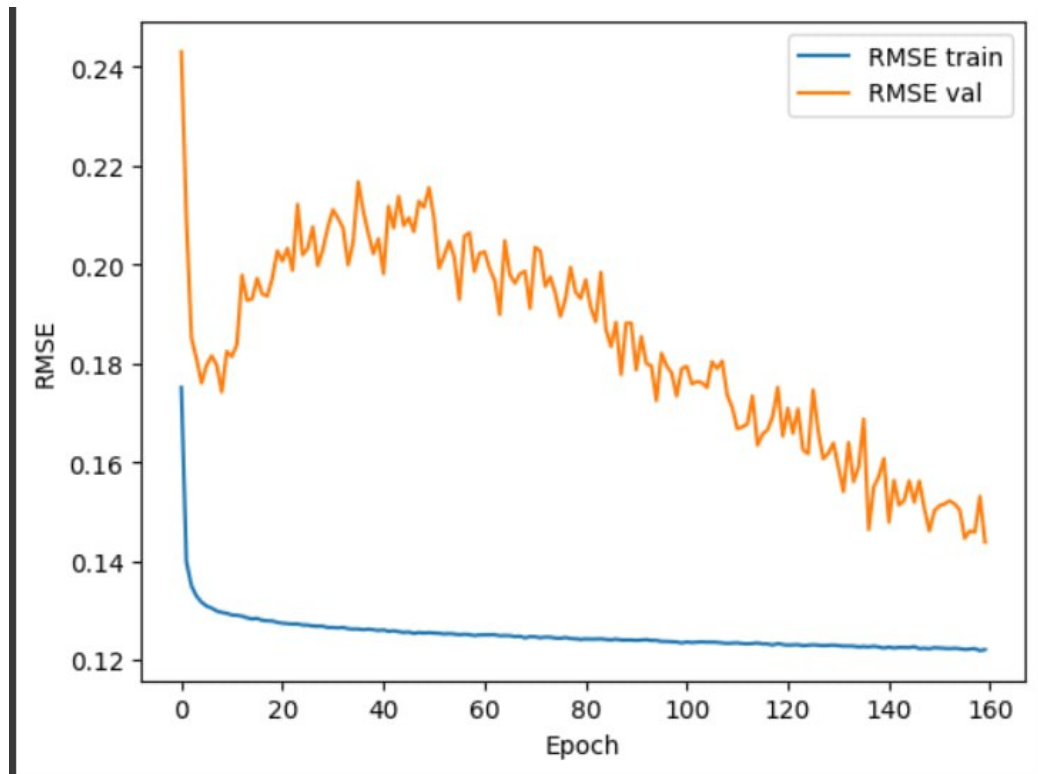


Figura 3.39: Resultados Obtenidos después del entrenamiento

### 3.3.5. Iteración 5 Agrandando el Modelo

Después de Probar con varias configuraciones y parámetros decidimos ampliar el modelo agregando más neuronas en las capas ocultas y aumentando el número de iteraciones EPOCH ( número de veces que el algoritmo de aprendizaje recorre todo el conjunto de datos de entrenamiento.) a 400. Haciendo Esto no mejoraron los resultados, por el contrario disminuyó la precisión del modelo.

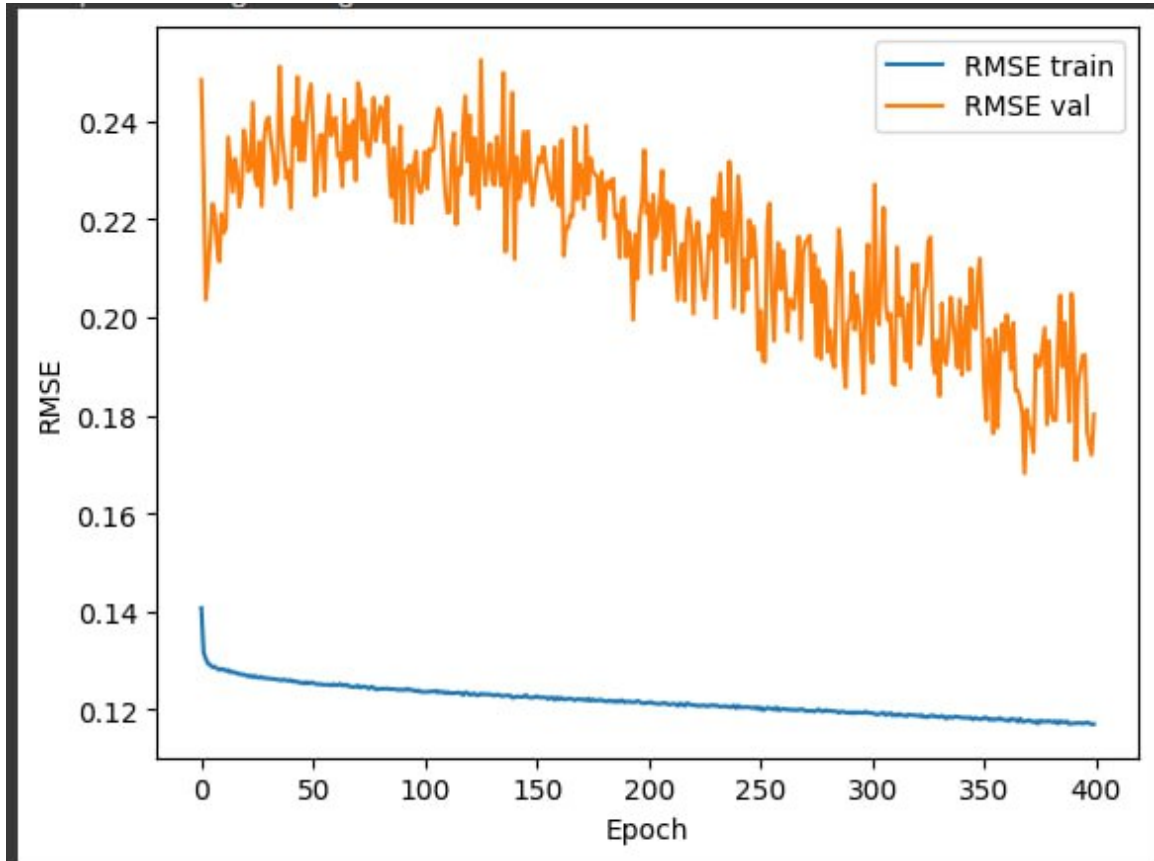


Figura 3.40: Resultados de agrandar el modelo

### 3.3.6. Iteración 6: Modificación de los hiperparámetros del modelo

Aquí colocamos 150 neuronas en las capas ocultas y redujimos el EPOCH en 200

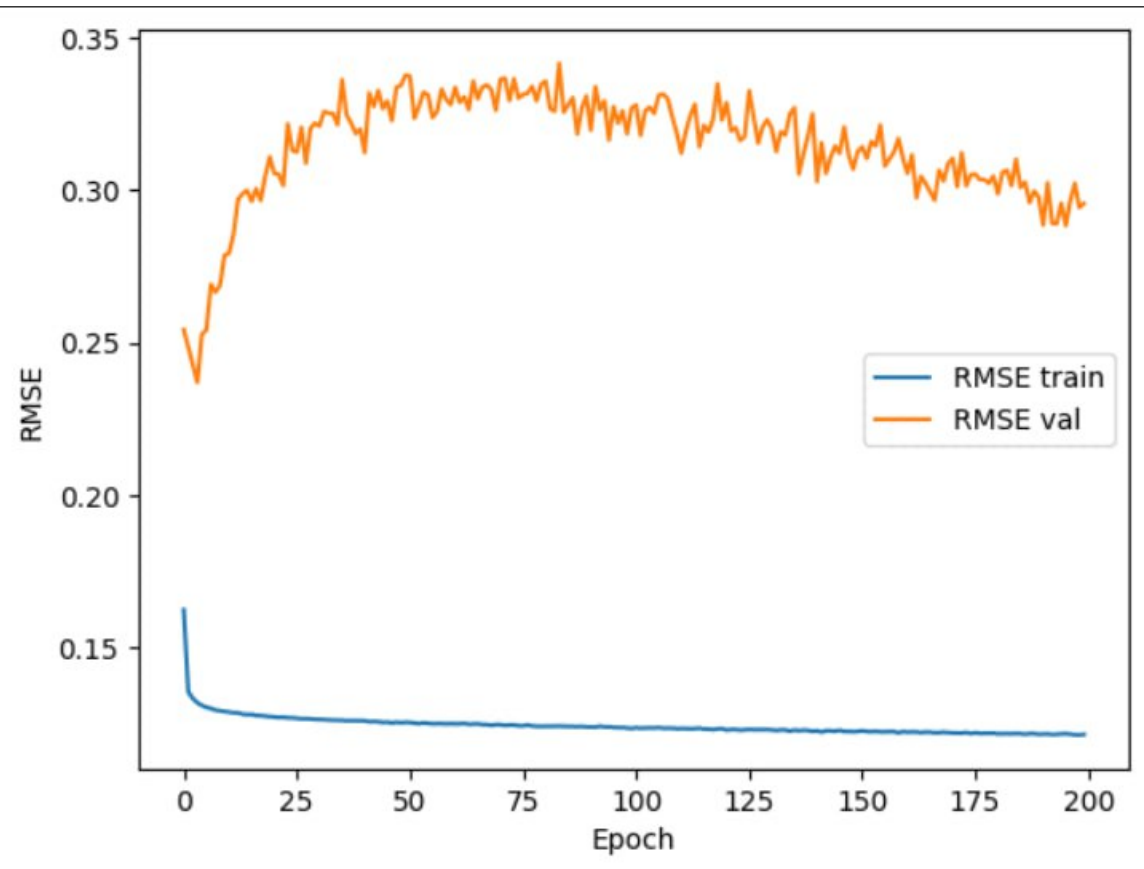


Figura 3.41: Resultados

```
Performances:  
Train RMSE: 0.082  
Validation RMSE: 0.293  
Test RMSE: 0.422
```

Figura 3.42

### 3.3.7. Iteración 8: Rectificando errores en los datos

Después de varias iteraciones y muchas pruebas cambiando los hiperparámetros del modelo notamos que existían en los datos un faltante de noticias. El dataset de noticias está desde julio del 2012 hasta febrero del 2020, mientras que el de los precios estaba desde el 2000 hasta el 2024. Por esto cambiamos y acortamos el dataset para hacer coincidir estos rangos de fecha , dejándonos los siguientes resultados.

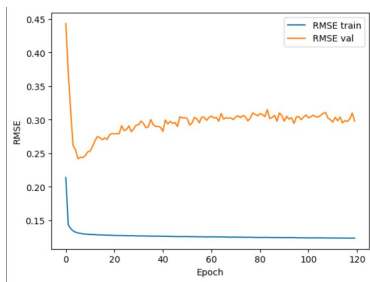


Figura 3.43: Prueba 1

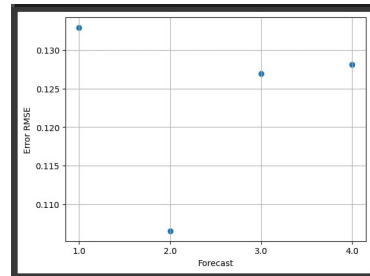


Figura 3.44: Errores en la predicción de varios parámetros

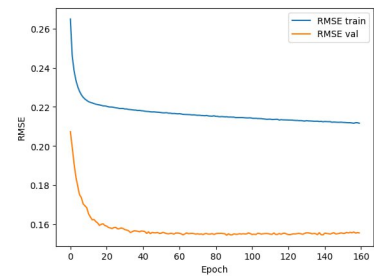


Figura 3.45: Prueba 2

Se hace notar la baja capacidad de generalización del modelo, así como la baja calidad de los resultados obtenidos.

# Capítulo 4

## Análisis de los resultados

### 4.0.1. Resultados

Después de varias iteraciones de experimentación e intentos de optimizar nuestro modelo para resolver la tarea de predicción del mercado utilizando noticias como elemento de soporte para este análisis, observamos que la forma en que relacionamos las noticias con los precios solo introdujo ruido en el proceso. Por tanto, es necesario encontrar una representación más adecuada para llevar a cabo esta tarea. Al explorar la bibliografía consultada, encontramos diversos enfoques para resolver este problema, cada uno con sus pros y sus contras. Entre todos los artículos revisados, hubo una idea que nos llamó particularmente la atención: la de proporcionar explicabilidad a los resultados, lo cual se aleja del enfoque que tomamos, pero es un factor crucial para comprender si estos modelos son realmente capaces de captar la complejidad del sistema financiero y relacionarlo efectivamente con datos del mundo real.

### 4.0.2. Recomendaciones

Adicionalmente, consideramos que futuros trabajos deberían enfocarse en la integración de técnicas de aprendizaje profundo con métodos de interpretación de modelos. Esto no solo aumentaría la precisión de las predicciones, sino que también facilitaría la comprensión de cómo las noticias y otros factores externos influyen en el mercado. La incorporación de análisis de sentimientos y el uso de modelos híbridos que combinan datos estructurados y no estructurados podrían ofrecer una perspectiva más holística y precisa. De esta manera, no solo se mejoraría la exactitud de los modelos predictivos, sino que también se incrementaría la confianza en su aplicabilidad práctica en entornos financieros reales.

# Conclusiones

La realización de esta investigación nos confrontó con la compleja y errática realidad de intentar predecir entornos tan volátiles como el mercado financiero. Este desafío representa un reto significativo para los modelos de aprendizaje actuales. Al analizar los resultados de varios estudios y aplicar nuestras propias metodologías, constatamos la dificultad inherente y la imperiosa necesidad de desarrollar herramientas que integren variables externas, como las noticias, en los sistemas de predicción.

Nuestra investigación nos sirvió como base para comprender mejor la utilización y los problemas presentes en las soluciones basadas en *machine learning*. Revela que, aunque existe una abundancia de datos y una considerable inversión en este campo, los resultados no son tan sobresalientes como en otras áreas que también emplean técnicas de aprendizaje automático.

En resumen, si predecir el mercado fuese una tarea sencilla, todos seríamos ricos. Sin embargo, esto mismo cambiaría el mercado una vez más, creando nuevos desafíos que resolver. También pudimos notar que nuestro enfoque para resolver el problema estaba incorrecto y no abordaba correctamente la solución del problema.



## Referencias Bibliográficas

- 1. Optimizing LSTM for time series prediction in Indian stock market**  
Anita Yadav, C K Jha, Aditi Sharan  
*Procedia Computer Science*, 167:2091-2100, 2020  
International Conference on Computational Intelligence and Data Science  
ISSN: 1877-0509  
DOI: <https://doi.org/10.1016/j.procs.2020.03.257>  
URL: <https://www.sciencedirect.com/science/article/pii/S1877050920307237>
- 2. Stock price index movement classification using a CEFLANN with extreme learning machine**  
Rajashree Dash, P. K. Dash  
In *2015 IEEE Power, Communication and Information Technology Conference (PCITC)*, pages 22-28, 2015  
Keywords: Indexes, Training, Artificial neural networks, Computational efficiency, Computational modeling, Neurons, Testing, ELM, CEFLANN, RBF, Chebyshev FLANN, Stock Price index, Technical Indicators  
DOI: 10.1109/PCITC.2015.7438176
- 3. Deep Learning for Stock Market Prediction**  
M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, S. Shahab  
*Entropy*, 22(8):840, 2020  
URL: <https://www.mdpi.com/1099-4300/22/8/840>
- 4. Explainable stock prices prediction from financial news articles using sentiment analysis**  
Shilpa Gite, Hrituja Khatavkar, Ketan V. Kotecha, Shilpi Srivastava, Priyam Maheshwari, Neerav Pandey  
*PeerJ Computer Science*, 7, 2021  
URL: <https://api.semanticscholar.org/CorpusID:231827830>
- 5. HighFrequency Trading and Financial TimeSeries Prediction with Spiking Neural Networks**  
Spiking neural networks (SNN)  
2021  
URL: <https://onlinelibrary.wiley.com/doi/10.1002/wilm.10927>
- 6. A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data Driven Networks**

LSTM (Redes Neuronales Recurrentes), ARIMA (Predictor Estadístico)

2023

URL: <https://www.mdpi.com/1999-5903/15/8/255>

**7. Redes Neuronales Transformers aplicada a la predicción de activos financieros**

Transformers

2021

URL: <https://youtu.be/8fOR1tqQF6I?si=zmmUSImGtotl-eR5>

**8. Time Series Data Analysis for Stock Market Prediction**

ARIMA, HOTL-Winters, SMA

2020

URL: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3563111](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3563111)

Aquí esta nuestra tabla de revisión bibliográfica