Predicting Price Fluctuations Using Sentiment Analysis and Machine Learning , *Support Vector Machine*, *Logistic Regression*, and Naive Bayes

ANDRÉS SOTO , JANICE MAMANI, AND JAVIER AYALA

Compiled November 24, 2023

En este estudio, se investiga la aplicación de técnicas de aprendizaje automático para analizar la relación entre la evolución de precios y el sentimiento expresado en Twitter. El objetivo principal de este proyecto es extraer el sentimiento de los tweets para luego poder relacionarlos con los movimientos de precios de 12 activos cotizados en el NYSE. Se evalúan y comparan métodos como Support Vector Machines, Multi-Layer Perceptron y Random Forest, resaltando su eficacia en la clasificación de sentimientos y su impacto en los mercados financieros. Este trabajo proporciona un análisis cualitativo detallado de estos métodos de clasificación, ofreciendo una perspectiva crítica sobre la idoneidad de los modelos propuestos para abordar la tarea de relacionar el sentimiento expresado en las redes sociales con la evolución de precios en los mercados financieros.

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCTION

En la última década, se ha evidenciado un crecimiento exponencial en la difusión de las redes sociales, hasta el punto de que su uso se ha convertido en una práctica diaria generalizada. Estas plataformas han alcanzado un nivel de influencia tal, que el material compartido en ellas puede tener un impacto significativo en nuestras conductas, incluyendo la forma en que nos vestimos, nos comportamos e incluso cómo pensamos. Además, estas plataformas reflejan intensamente el sentir de los usuarios a través de sus publicaciones. Por lo tanto, las decisiones que tomamos en nuestra rutina diaria se ven ahora influenciadas por lo que se difunde a través de estas aplicaciones.

La influencia de estas plataformas también se extiende al ámbito de las decisiones de inversión, lo cual representa un desvío de las acciones racionales, que deberían determinarse por la información fundamental disponible para los inversores. Un ejemplo ilustrativo es el incidente asociado a "GameStop", donde un grupo de inversores minoristas se coordinó a través de foros en línea, específicamente desde la plataforma Reddit, con el propósito de adquirir acciones de empresas que habían experimentado una notable baja en su valor. Entre las preferencias de inversión estaba GameStop, una empresa dedicada a la venta de videojuegos cuyo valor en el mercado había declinado considerablemente durante años. A mediados de 2020, sus acciones cotizaban alrededor de 4 dólares. Sin embargo, con la participación coordinada de inversores minoristas, hacia finales de ese año, el valor ascendió a 16 dólares. Para enero de 2021, alcanzó los 350 dólares por acción. Este caso es un ejemplo de cómo, a través de las redes sociales, los inversores frecuentemente comparten información que puede desencadenar una secuencia de compras sucesivas de acciones, propiciando un aumento abrupto en los precios.

La rama de estudio de las finanzas que se ocupa de analizar el mercado financiero en equilibrio se llama "Asset Pricing". En términos simples, es el campo de las finanzas que estudia la determinación del precio de los activos. Uno de los subcampos de esta área abarca el "sentimiento", que consiste en considerar la "irracionalidad" o el "comportamiento" de los inversores en sus expectativas, las cuales determinan el precio de los activos.

Por consiguiente, este proyecto propone relacionar el sentimiento reflejado en las publicaciones de Twitter sobre los precios de determinadas acciones. Intweetivamente, un sentimiento positivo se relaciona con un incremento en la compra de los activos, ya que las expectativas están siendo guiadas de manera positiva, elevando los precios. Por otro lado, un sentimiento negativo se relacionaría con reducciones en el nivel de precios, debido a que las expectativas están siendo influenciadas negativamente. Todo ello se llevará a cabo empleando herramientas de aprendizaje automático (machine learning).

El presente informe sigue la siguiente estructura: en la sección 2 se aborda la definición de “sentiment" en el marco teórico financiero; la sección 3 revisa los antecedentes y la literatura existente en relación con el machine learning y el "sentiment". Posteriormente, la sección 4 proporciona una descripción de las bases de datos empleadas para el proyecto y detalla su tratamiento. La sección 5 se dedica al análisis de los modelos de machine learning utilizados, incluyendo una breve descripción del código implementado. A continuación, en la sección 6, se lleva a cabo

una comparación de los resultados obtenidos por los modelos con la evolución de los precios, centrándose en un grupo de muestra seleccionado. Finalmente, la sección 7 presenta las conclusiones del estudio.

2. ENFOQUE CONCEPTUAL EN FINANZAS

Previo a abordar la relación entre el sentimiento del mercado y la evolución de los precios, resulta pertinente establecer inicialmente la definición del término *sentiment*. Uno de los objetivos del Asset Pricing, una rama de estudio en Finanzas, es calcular el precio de los activos financieros en el mercado. Para formalizar y entender el concepto de "sentiment", consideremos una economía cuyo estado en el momento t se resume en un vector de estado X_t. Un vector de parámetros ρ determina la ley de movimiento de X_t. Supongamos que X_t es observable para los inversores en t, y que J_t denota el conjunto de información generado por las observaciones de los inversores de X. La economía presenta N activos que producen un vector de pagos $D_t(X_t)$. Sea $M_t(X_t, \phi)$ el factor de descuento estocástico (SDF) que refleja las preferencias de un inversor representativo con parámetros de preferencia ϕ. Este inversor tiene conocimiento de ϕ pero no necesariamente de ρ. El grado de conocimiento de ρ por parte del inversor es importante para la fijación del precio de los activos porque afecta a su capacidad para prever realizaciones futuras de X y, por tanto, los futuros pagos de los activos y el SDF, que dependen de X.

La mayoría de los estudios empíricos sobre la valoración de activos de corte transversal se basan en la teoría de la valoración de activos de expectativas racionales según la tradición de Lucas (1978) para derivar predicciones del modelo y una hipótesis nula. Según las expectativas racionales, los inversores conocen los parámetros ρ. Por lo tanto, los inversores conocen el proceso de movimiento del estado de la economía. El marco econométrico de gran parte de la valoración empírica de activos se basa en este sólido supuesto. Recogiendo los rendimientos (brutos) $R_{it+1} = \left(\frac{P_{it+1}+D_{it+1}}{P_{it}}\right)$ en el vector $\mathbf{R}_{t+1}$, los inversores ponen precio a los activos según

$$\mathrm{E}\left[\mathbf{R}_{t+1}(\mathbf{X}_{t+1})M_{t+1}(\mathbf{X}_{t+1}, \phi)|J_t\right] = \mathbf{1}$$

donde la expectativa condicional $\mathrm{E}[\cdot|J_t]$ es evaluada bajo la densidad $f(\mathbf{X}_{t+1}|J_t, \rho)$. De forma intweetiva, la información disponible para los agentes es evaluada de manera racional para poder tener una expectativa diciplinada sobre el futuro.

El sentiiento en el asset pricing de los activos consiste en que la distribución de probabilidad de $\mathbf{X}_{t+1}$ percibida por los inversionistas en el tiempo t puede desviarse de las expectativas racionales. En este caso

$$\mathrm{E}^s\left[\mathbf{R}_{t+1}(\mathbf{X}_{t+1})M_{t+1}(\mathbf{X}_{t+1}, \phi)|J_t\right] = \mathbf{1}$$

se cumple con $\mathrm{E}^s[.|J_t]$ evaluada en la densidad subjetiva percibida por los inversionistas (Wilksch,2023) .

3. ANTECEDENTES

En esta sección, se examinan cuatro documentos relevantes en relación con el tema del proyecto. Además, se detallan diversas técnicas de machine learning que se han empleado en dichos trabajos, las cuales son consideradas en la elección de los modelos a implementar en el presente proyecto.

A. *"Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning"*

Pregunta de investigación: ¿Es posible utilizar el análisis de sentimiento y modelos de aprendizaje automático, como redes neuronales, máquinas de vectores de soporte y random forest (RF), para realizar predicciones precisas en este mercado altamente volátil y complejo?

Métodos de Machine Learning:

- **Análisis de sentimientos:** El análisis de sentimientos se define en este contexto como el proceso de medir el grado de placer o desagrado de una experiencia emocional en el texto de los tweets relacionados con criptomonedas. Se utiliza una técnica llamada Valence Sentiment Analysis, que cuantifica este grado de placer o desagrado. Se utiliza un diccionario [VADER] que está diseñado para Twitter. Cuando se aplica VADER a un texto de tweet, produce un vector con valores normalizados para los siguientes aspectos: sentimiento positivo, sentimiento neutro, sentimiento negativo y sentimiento compuesto.
- **Feature Vectors:** El vector de características V se utiliza para predecir la dirección del mercado (aumento o disminución de precios) en función de varios aspectos del mercado y el análisis de sentimiento de las redes sociales.
- **Support Vector Machines:** Las SVM buscan encontrar el hiperplano de decisión que mejor separa las clases en un espacio de alta dimensionalidad. Las funciones de kernel permiten realizar esta separación incluso en espacios donde los datos no son linealmente separables. Los vectores de soporte son los ejemplos de entrenamiento que son fundamentales para determinar el hiperplano de decisión y, por lo tanto, son esenciales en la clasificación.
- **Multi-Layer Perceptron:** El Multi-Layer Perceptron (MLP) es un tipo de red neuronal que consta de al menos tres capas de nodos. Los MLP pueden utilizar la retropropagación (backpropagation) y el aprendizaje supervisado para el entrenamiento. Como tal, pertenecen a la clase de redes neuronales de retropropagación (Back Propagation, BP). La función de un MLP se puede expresar de manera simple como F() = Rm > Ro, donde m es el tamaño de la dimensión del vector de características y o es el tamaño de la dimensión del objetivo.
- **Random Forest (RF):** Son metaestimadores que ajustan una serie de árboles de decisión en diferentes submuestras del conjunto de datos. Los Bosques Aleatorios utilizan un enfoque de conjunto, combinando predictores basados en árboles, donde cada árbol depende de los valores de un vector aleatorio con una distribución uniforme para todos los árboles en el bosque.

Resultados:

- **Bitcoin (Tabla 3):**
  - MLP fue el modelo con mejor rendimiento para Bitcoin, con una precisión de 0.76.
  - Tanto SVM como RF lograron predecir.
  - Los datos de Twitter por sí solos no pudieron utilizarse para predecir el movimiento del mercado en ningún modelo y su inclusión empeora los resultados de SVM y RF.

- **Ethereum (Tabla 4):**
  - El mejor modelo de rendimiento fue MLP y ninguno otro pudo reducir el peso del azar. También MLP fue el único que aprovechó los datos de mercado y de Twitter para mejorar precisión.
- **Ripple (Tabla 5):**
  - MLP fue el mejor modelo con una precisión del 0.64.
  - SVM también superó al azar por un margen pequeño cuando se utilizaron solo datos de Twitter.
  - Los datos de Twitter pudieron predecir el mercado por sí solos al utilizar el modelo SVM con una precisión del 0.53 y puntuaciones de precisión del 0.6.
- **Litecoin (Tabla 6):**
  - SVM fue el mejor modelo de rendimiento con una precisión de 0.66 y una puntuación de precisión del 0.8.
  - RF tuvo un rendimiento ligeramente mejor que MLP cuando se utilizaron datos de Twitter y datos de mercado.
  - Los datos de Twitter pudieron superar al azar.
  - Los datos de Twitter pudieron predecir el mercado por sí solos al utilizar los modelos MLP y RF.

Descripción de su base de datos: La data de mercado fue sacada de 65 criptoactivos de la página cryptocompare.com con un seguimiento de 80 días de data histórica. Por otro lado, la data social fue sacada de Twitter con el siguiente criterio:

1. Los tweets se han creado durante el período de tiempo del estudio,
2. Los tweets contienen el nombre (por ejemplo, “bitcoin”) o el símbolo bursátil (por ejemplo, “btc”) de una de las monedas analizadas.
3. Los tweets están escritos en inglés, ya que el diccionario solo funciona así.
4. No se permiten tweets duplicados, aunque se permiten retweets, ya que estos pueden indicar una tendencia sentimental.

En promedio, se rescataron 345,000 tweets por día y se llegó a la cantidad de 20,789,572 tweets al final. La figura 1 muestra el gráfico de SVM.

B. *“Twitter Sentiment Classification using Distant Supervision”*

Pregunta de investigación: Desarrollar un método efectivo para clasificar automáticamente el sentimiento en los mensajes de Twitter utilizando etiquetas ruidosas, lo que es una contribución novedosa.

Método de Machine learning

Naive Bayes: Es un modelo simple que funciona bien en la categorización de texto. En este caso, se utiliza un modelo multinomial Naive Bayes.

$$c^* =_c P_{NB}(c|d)$$

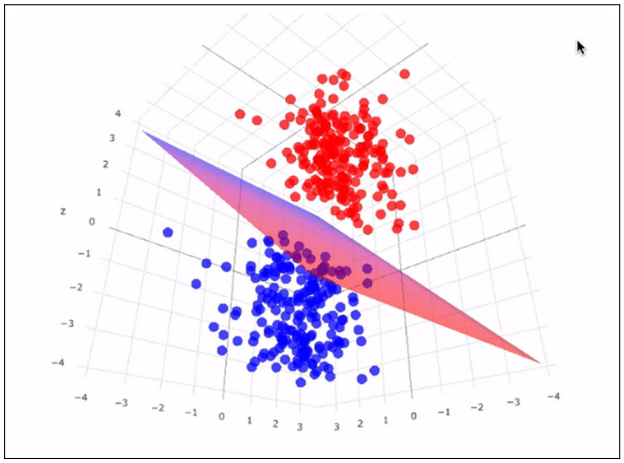

Fig. 1. Support Vector Machine Graphic

$$P_{NB}(c|d) := \frac{P(c)\prod_{i=1}^{m} P(f_i|c)^{n_i(d)}}{P(d)}$$

El proceso implica asignar una clase, denotada como "c", a un tweet llamado "d". La asignación se realiza utilizando una fórmula que involucra la probabilidad condicional de la clase "c" dado el tweet "d". La fórmula incluye características (representadas como "f") y cuenta el número de veces que cada característica "f" se encuentra en el tweet "d". Existen en total "m" características.

Los parámetros ”$P(c)$” y ”$P(f_i|c)$” se obtienen a través de estimaciones de máxima verosimilitud y se menciona que se utiliza un suavizado de tipo "add-1" para características no vistas o inéditas.

Maximum Entropy: El modelo está basado en características. En un escenario de dos clases, es similar a usar regresión logística para encontrar una distribución sobre las clases. A diferencia del Naive Bayes, MaxEnt no realiza suposiciones de independencia para sus características, lo que significa que puedes agregar características como bigramas y frases sin preocuparte por superposiciones de características.

$$P_{ME}(c|d,\lambda) = \frac{exp\left(\sum_i \lambda_i f_i(c,d)\right)}{\sum_{c'} exp\left(\sum_i \lambda_i f_i(c',d)\right)}$$

El modelo de MaxEnt se representa mediante una fórmula en la que "c" es la clase, "d" es el tweet y "λ" es un vector de pesos. Estos vectores de peso determinan la importancia de una característica en la clasificación. Un peso más alto indica que la característica es un indicador sólido para la clase. Los vectores de peso se encuentran mediante optimización numérica de los valores lambda para maximizar la probabilidad condicional.

Support Vector Machines: Las Máquinas de Soporte Vectorial (SVM) son otra técnica popular de clasificación. En este caso, utilizan el software SVM*light* con un núcleo lineal. Los datos de entrada consisten en dos conjuntos de vectores de tamaño "m". Cada entrada en el vector corresponde a la presencia de una característica. Por ejemplo, cuando se usa un extractor de características unigramas, cada característica es una palabra individual que se encuentra en un tweet. Si la característica está presente, el valor es 1; si está ausente, el valor es 0. Utilizan la presencia de características en lugar de un recuento, lo que evita tener que escalar los datos de entrada y acelera el procesamiento en general.

Resultados

- **Unigramas:** El extractor de características de unigramas es la forma más simple de obtener características de un tweet. Estos resultados son muy similares a otros, estos eran un 81.0%, 80.4% y 82.9% de precisión para Naive Bayes, MaxEnt y SVM, respectivamente. Esto se asemeja mucho a los resultados de 81.3%, 80.5% y 82.2% para el mismo conjunto de clasificadores.
- **Bigramas:** Utilizan bigramas para lidiar con tweets que contienen frases negadas como "no es bueno" o "no es malo". Los bigramas tienden a ser muy escasos y la precisión general disminuye en el caso de MaxEnt y SVM. Incluso el problema de la escasez de palabras individuales en clases de equivalencia, no ayuda. El problema de la escasez se puede observar en tweets como:

  > "@stellargirl *I looooooooovvvvvveee my Kindle2. No es que el DX sea genial, pero el 2 es fantástico por sí mismo.*"

  MaxEnt dio probabilidades iguales a las clases positivas y negativas en este caso porque no hay un bigrama que incline la polaridad en ninguna dirección.
- **Partes del Discurso:** Utilizamos etiquetas de partes del discurso (POS) como características porque la misma palabra puede tener muchos significados diferentes según su uso. Por ejemplo, "over" como verbo puede tener una connotación negativa. "Over" también puede usarse como sustantivo para referirse al cricket, lo que no lleva una connotación positiva o negativa. Sin embargo, las etiquetas de POS no fueron útiles. La siguiente tabla **??** muestra los indicadores de accuracy.

Table 1. Classifier Accuracy

Features	Naive Bayes	MaxEnt	SVM
Unigram	81.3	80.5	82.2
Bigram	81.6	78.8	82.8
Unigram + Bigram	N/A	82.7	81.6
Unigram + POS	N/A	79.9	81.9

Base de datos

Se realizó un proceso de scrapping para recolectar la data mediante un API que se puede descargar de https://nlp.stanford.edu/software/classifier.shtml. El problema es que la información recolectada no está presente en el documento ni en github. Sin embargo, tenemos algunas especificaciones de la "Data" recolectada.

El API de Twitter impone un límite de 100 tweets en una respuesta para cualquier solicitud. El proceso de obtención de datos utiliza un "scraper" con la capacidad de definir la frecuencia de las solicitudes.

Los datos de entrenamiento utilizados en este estudio abarcan desde el 6 de abril de 2009 hasta el 25 de junio de 2009. Luego, se aplican diversos filtros a los datos de entrenamiento:

1. Se eliminan los emoticonos que se detallan en la Tabla 3. Esto se hace para mejorar el proceso de entrenamiento, ya que, de lo contrario, los clasificadores MaxEnt y SVM tienden a dar una importancia excesiva a los emoticonos, lo que afecta negativamente la precisión.
2. Se descartan los tweets que contienen tanto emoticonos positivos como negativos.
3. Se eliminan los "retweets", que son reenvíos de tweets de otros usuarios a una cuenta diferente.
4. Se excluyen los tweets que contienen ":P" debido a un problema en el API de Twitter.
5. Se eliminan los tweets duplicados, ya que ocasionalmente el API de Twitter puede devolver tweets idénticos.

Luego de la limpieza de datos, se seleccionan los primeros 800,000 tweets con emoticonos positivos y 800,000 tweets con emoticonos negativos, lo que resulta en un conjunto de entrenamiento de 1,600,000 tweets. Por otro lado, los datos de prueba son recopilados manualmente a través de la aplicación web. Este conjunto de pruebas consta de 177 tweets negativos y 182 tweets positivos. La siguiente tabla **??** muestra la lista de emoticones. Asimismo, la tabla **??** muestra el gráfico de Naives Bayes.

Table 2. List of Emoticons

Emoticons mapped to :)	Emoticons mapped to :(
:)	:(
:-)	:-(
:D	
;)	
=)	

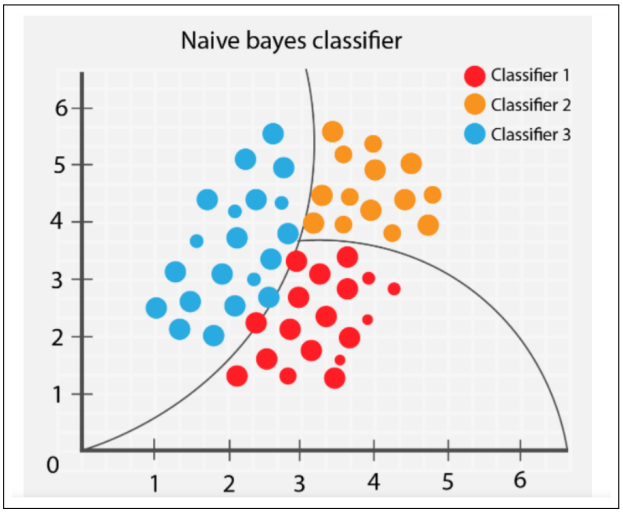

Fig. 2. Naive Bayes Graphic

C. *"Stock Prediction Using Twitter Sentiment Analysis"*

Pregunta de investigación: Existe una correlación significativa entre el estado de ánimo público y el sentimiento del mercado financiero. Específicamente, se investiga cómo las emociones y el estado de ánimo de las personas, expresadas a través de Twitter, pueden influir en la toma de decisiones en el mercado de activos, representado por el DJIA

Método de Machine learning: La técnica desarrollada por los autores es la siguiente: Los datos originales del DJIA se someten

a un proceso de preprocesamiento, mientras que los mensajes de Twitter se analizan mediante un algoritmo de análisis de sentimientos, generando valores emocionales para cuatro clases distintas por día (calmado, feliz, alerta y amable). Estos valores emocionales y los datos del DJIA procesados se introducen en un marco de aprendizaje de modelos, el cual utiliza SOFNN para desarrollar un modelo predictivo de los valores futuros del DJIA. Para realizar el análisis de sentimiento, los autores crearon su propia lista de palabras basada en el cuestionario Profile of Mood States (POMS). Además, se filtro la data de twitter que solo contenía las palabras "feel", "makes me", "I´m" o "I am". Por último, se utilizó un algoritmo contador de palabras para encontrar el puntaje de cada palabra POMS.

$$\text{score of a word} = \frac{\text{\# of times the word matches tweets in a day}}{\text{\# of total matches of all words}}$$

Resultados:Con el objetivo de elaborar el modelo de aprendizaje y la predicción se utilizaron diferentes algoritmos como regresión lineal, SVM, Redes Neuronales Difusas Autoorganizativas (SOFNN).

Los resultados reflejan que el ánimo público puede ser identificado de forma eficiente a partir de las numerosas publicaciones en Twitter empleando métodos básicos de procesamiento de lenguaje. Además, de las diversas emociones observadas, solamente la calma y la alegría tienen un efecto causal sobre el DJIA en un intervalo de 3 a 4 días.

Finalmente, los investigadores indican que el análisis omite varios factores. El conjunto de datos no refleja de manera precisa el estado emocional general, ya que se limita a usuarios de Twitter que se expresan en inglés. Una mayor precisión en la correlación podría lograrse si se analiza directamente el estado de ánimo real. Se plantea la hipótesis de que el ánimo de las personas influye en sus decisiones financieras, aunque no hay una correlación directa entre los inversores en bolsa y los usuarios frecuentes de Twitter. No obstante, existe una correlación indirecta, ya que las decisiones de inversión podrían estar influenciadas por el ánimo general de las personas cercanas, es decir, el sentir colectivo. Estas áreas persisten como direcciones para futuras investigaciones.

Base de datos: En el estudio se emplearon dos bases de datos. Una de ellas comprende información relativa al mercado de activos, derivada de datos del DJIA desde junio de 2009 hasta diciembre del mismo año. Esta base de datos integra los precios de apertura, máximos, mínimos y cierre diarios. Paralelamente, se utilizó información pública proveniente de Twitter que abarca más de 476 millones de tweets recopilados durante el periodo comprendido entre junio de 2009 y diciembre de 2009. Los datos de Twitter contienen detalles como la fecha de publicación, el nombre del usuario y el contenido del tweet. Para los propósitos del proyecto, se realizó una segmentación de los datos en función de los días correspondientes. El gráfico **??** muestra el SOFNN.

D. *"PyFin-sentiment: Towards a machine-learning-based model for deriving sentiment from financial tweets"*

Pregunta de investigación: ¿Cómo podemos diseñar un modelo funcional que pueda extraer el sentiment del autor de posts de redes sociales relacionadas a las finanzas ? ¿Cuál es el rendimiento de este artefacto modelo en comparación con los modelos existentes del ámbito de los textos relacionados con las finanzas o de las publicaciones genéricas en redes sociales? ¿Puede un modelo pequeño y específico del dominio superar a los LLM más genéricos? ¿Cómo cambia el rendimiento de los modelos

??

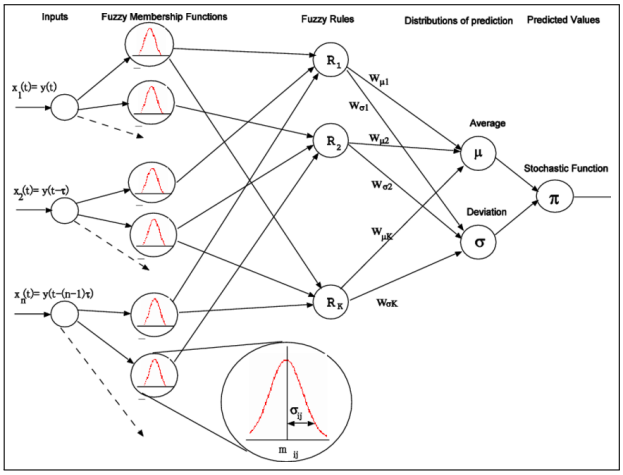

Fig. 3. Self-Organising Fuzzy Neural Network (SOFNN) Graphic

entrenados en mensajes de Twitter cuando se aplican a mensajes de StockTwits?

Método de Machine learning: El autor identifica 3 distintas formas para poder extraer el sentimiento de los textos:

- **Dictionary based models:** Los modelos de análisis de sentimiento basados en diccionarios utilizan listas de palabras o frases con asignaciones de puntuaciones de sentimiento realizadas por humanos. Estos modelos clasifican palabras como positivas o negativas y luego calculan el sentimiento de un documento sumando las puntuaciones de todas las palabras. Aunque esta metodología es explicativa y computacionalmente eficiente en la inferencia, tiene desventajas. La compilación del diccionario por humanos es un proceso que consume mucho tiempo y las decisiones sobre escalas y puntuaciones pueden influir en el rendimiento del modelo final. Además, este enfoque puede fallar si el texto no contiene muchas de las palabras en la lista, lo que es común en contenidos de redes sociales con errores tipográficos, jerga y emojis. Modelos no adaptados a este tipo de contenido tienden a clasificar los textos como neutros por la falta de coincidencias.

- **Machine learning model:** Los modelos de aprendizaje automático pueden entrenarse con grandes conjuntos de datos etiquetados para evaluar el impacto positivo o negativo de las palabras en el sentimiento. Esto permite una optimización directa del objetivo. Estos modelos se centran en analizar frases o documentos cortos en lugar de asignar puntuaciones de sentimiento por palabra. Algunos modelos comunes incluyen máquinas de vectores de soporte, Naive Bayes, modelos basados en árboles y regresión logística. Por ejemplo, las SVM pueden lograr una precisión de alrededor del 75% en la clasificación de publicaciones financieras como "alcistas" o "bajistas". Para tweets genéricos, la precisión puede llegar hasta el 83% con diferentes modelos.

- **Deep learning models:** En el campo del Procesamiento del Lenguaje Natural (PLN), la mayoría de los modelos líderes en áreas como la respuesta a preguntas o la generación de texto se basan en el aprendizaje profundo. Los investigadores han empezado a aplicar este enfoque al análisis de

sentimientos, que implica clasificar textos. Actualmente, la mayoría de los modelos de aprendizaje profundo en PLN utilizan grandes modelos de lenguaje (LLM) adaptados a tareas específicas. Estos LLM son redes neuronales de gran tamaño entrenadas con grandes cantidades de datos. Al aprender patrones complejos del lenguaje natural, pueden ser utilizados para una variedad de tareas, incluso diferentes a la de su entrenamiento original. Representaciones del texto aprendidas por LLM, como "Bidirectional Encoder Representations from Transformers" (BERT), pueden ser afinadas por una sola capa de red neuronal para diversas tareas.

Resultados: Para cada una de las tres clases de sentimiento, la Tabla **??** enumera los tokens con los mayores coeficientes. Estas palabras, si están presentes en un documento, tienen el mayor efecto en la predicción a favor de cada una de las clases. Observamos que el modelo ha aprendido un vocabulario específico del dominio, en el que palabras como "comprar", "comprar", "correr" indican un sentimiento positivo, y palabras como "tirar", "bajar" o "corto" indican un sentimiento negativo. Además, ha aprendido que patrones numéricos como "123C" o "123P" (opción de compra u opción de venta con un precio de ejercicio de 123 $) expresan un sentimiento positivo o negativo, respectivamente.

Class	Tokens associated with largest coefficients
Bullish	run, buy, rip, cal, call, 999c, bull, ulli, bul, llis, lish, ath, 🚀, up, buy
Neutral	tick, play, hart, name, hit, |, =, real, ser, ▦, 9:9, sur, er?, or, chat,
Bearish	fall, earl, dump, dum, rish, lowe, dow, shor, low, red, hort, 999p, down, los, put,

Fig. 4. Resultados

Este proyecto también analiza las palabras más recurrentes en relación con sus respectivas categorías. Para la presentación de estos resultados, se emplea una nube de palabras (wordcloud) implementada en Python. Workcloud proporciona una representación visual donde las palabras más frecuentes se presentan de manera prominente, ofreciendo así una rápida percepción de los términos más relevantes en cada categoría. El código del Workcloud se detallará más adelante.

Base de datos: Se recopilaron 10,000 tweets sobre finanzas e inversión y se asignó manualmente un sentimiento de mercado a cada uno, representando la opinión del inversor sobre la rentabilidad futura de una acción. Se demostró que los modelos existentes entrenados en dominios cercanos tienen dificultades con este tipo de análisis debido al lenguaje especializado. Por lo tanto, se diseñó, entrenó e implementó un nuevo modelo de sentimiento que supera a todos los modelos previos al ser evaluado con mensajes de Twitter. En otra plataforma, nuestro modelo tiene un rendimiento comparable a los modelos basados en BERT. Además, se logró este resultado con costos de entrenamiento e inferencia considerablemente más bajos gracias al diseño simple del modelo. Se ha publicado el código como una biblioteca de Python para facilitar su uso por parte de futuros investigadores y profesionales.

4. BASE DE DATOS

En esta sección, se describe la base de datos que se utilizará. La base de datos contiene tweets desde el 2021-09-30 hasta 2022-09-29. Los datos son de alta frecuencia, pues la base de datos contiene 89793 oervaciones, por lo que cada día contiene 110

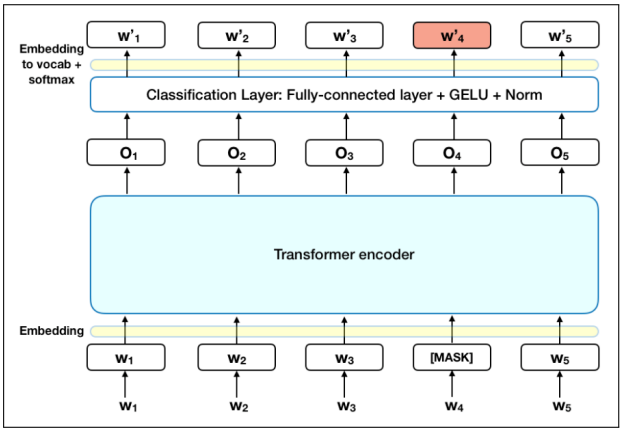

Fig. 5. Berth

tweets en promedio. Cada vector de la base de datos contiene la fecha y hora del tweet, el texto completo del tweet, el nombre de la acción o activo del que se habla y el nombre de la empresa. Según Kaggle, que es la página donde se obtuvo la información, este dataset sirve para estudios experimentales con análisis de sentimiento, predicción de precios de los activos y explorar la conexión entre el sentimiento público y el movimiento en el precio del activo. Además, se cuenta con la base de datos de los precios de cada activo, donde cada observacion contiene la fecha y el precio ajustado al cierre del día. [1]

A. Preprocesamiento

La limpieza de datos asegura la calidad de un conjunto de datos, permitiendo así obtener conclusiones confiables. El objetivo principal de la limpieza es proporcionar al modelo datos de entrada que se asemejen lo más posible al lenguaje natural, facilitando así la tarea de clasificación por sentimiento.

La limpieza de la base de datos se hará como en el paper de "PyFin-sentiment: Towards a machine-learning-based model for deriving sentiment from financial tweets". Este paper realiza el preprocesamiento de una manera sistemática: Se empieza eliminando todos los hipervínculos de los tweets, ya que no constituyen lenguaje natural. Esto será importante para las operaciones de filtrado posteriores, que se basan en el recuento de palabras. A continuación, eliminamos todos los duplicados del conjunto de datos. Filtramos dos tipos de duplicados. En primer lugar, filtramos los duplicados basándonos en los ID de los tweets en caso de que la API (el application programming interface de Twitter) devuelva resultados duplicados. En segundo lugar, eliminamos todos los tweets con textos duplicados de más de 5 palabras, ya que gran parte del contenido de Twitter lo generan robots que publican el mismo tweet varias veces. Elegimos este umbral porque los tweets cortos duplicados pueden ser mensajes legítimos (por ejemplo: "compré $TSLA"). Sin embargo, si se duplican dos tweets de más de cinco palabras, lo más probable es que se trate de un mensaje repetitivo publicado por una cuenta automatizada. En el presente proyecto, se realiza un análisis del market sentiment.

A continuación, filtramos los tweets en función del número

[1] El precio ajustado al cierre del dia contiene correcciones por el reparto de dividendos

de hashtags y cashtags. Una inspección manual revela que los tweets de spam suelen utilizar muchos hashtags o cashtags diferentes para aparecer en el mayor número de búsquedas posible. Por lo tanto, excluimos todos los tweets que contienen cinco o más cashtags u ocho o más hashtags. En este punto, sin embargo, los datos que quedan siguen conteniendo numerosos tweets spam. La mayoría de ellos son tweets más cortos con relativamente muchos hashtags o cashtags, pero no los suficientes para ser eliminados por el filtro anterior. Por lo tanto, imponemos otro filtro basado en la relación entre cashtags y palabras, hashtags y palabras, y menciones de otros usuarios y palabras. Exigimos que cada uno de estos ratios sea inferior o igual a 0.5, de modo que un tweet contenga al menos tantas palabras como cashtags, hashtags y menciones. Por último, la única forma de tweets no deseados que sigue representando una cantidad significativa de datos son los tweets sobre criptomonedas.

Antes de utilizar los datos para entrenar los modelos, los preprocesamos para mejorar el aprendizaje de patrones generalizables. ransformamos todas las menciones de nombres de usuario en "@user", sustituimos todos los dígitos por el número "9", convertimos todos los caracteres de nueva línea en espacios y nos aseguramos de que el texto esté en minúsculas. Estos pasos de preprocesamiento son esenciales para mitigar el riesgo de que los modelos de aprendizaje automático se ajusten en exceso a los patrones presentes en los datos de entrenamiento.

El gráfico 13 muestra los tweets de la base de tados original en la columna “Tweets”, y los tweets después de la limpieza a la base de datos en la columna “cleanTweet”

	Date	Tweet	Stock Name	Company Name	cleanTweet
0	2022-09-29 23:41:16+00:00	Mainstream media has done an amazing job at br...	TSLA	Tesla, Inc.	mainstream media has done an amazing job at br...
1	2022-09-29 23:24:43+00:00	Tesla delivery estimates are at around 364k fr...	TSLA	Tesla, Inc.	tesla delivery estimates are at around 999k fr...
3	2022-09-29 22:40:07+00:00	@RealDanODowd @WholeMarsBlog @Tesla Hahaha why...	TSLA	Tesla, Inc.	@user @user @user hahaha why are you still try...
4	2022-09-29 22:27:05+00:00	@RealDanODowd @Tesla Stop trying to kill kids,...	TSLA	Tesla, Inc.	@user @user stop trying to kill kids, you sad ...
5	2022-09-29 22:25:53+00:00	@RealDanODowd @Tesla This is you https://t.co/...	TSLA	Tesla, Inc.	@user @user this is you
...	...	...	...	...	...
80784	2021-10-13 16:47:19+00:00	XPeng P7 tops Sept sales among new EV makers i...	XPEV	XPeng Inc.	xpeng p9 tops sept sales among new ev makers i...
80786	2021-10-11 03:51:36+00:00	NIO reached 100,000 vehicle rolled off the pro...	XPEV	XPeng Inc.	nio reached 999,999 vehicle rolled off the pro...
80790	2021-10-01 04:43:41+00:00	Our record delivery results are a testimony of...	XPEV	XPeng Inc.	our record delivery results are a testimony of...
80791	2021-10-01 00:03:32+00:00	We delivered 10,412 Smart EVs in Sep 2021, rea...	XPEV	XPeng Inc.	we delivered 99,999 smart evs in sep 9999, rea...
80792	2021-09-30 10:22:52+00:00	Why can XPeng P5 deliver outstanding performan...	XPEV	XPeng Inc.	why can xpeng p9 deliver outstanding performan...

Fig. 6. Dataset in Python

5. MODELOS DE MACHINE LEARNING Y TF-IDF

En esta sección, se exponen los resultados de cada modelo, junto con las wordclouds que muestran las palabras más recurrentes asociadas a cada clasificación. Los modelos recibirán la data vectorizada, esa tarea es realizada por la metodologia Term Frequency-Inverse Document Frequency TF-IDF. Los resultados son ilustrados usando *Workcloud* para cada modelo. Los modelos de Machine Learning son aplicados con la librería sklearn.

La herramienta TF-IDF es usado para el procesamiento de lenguaje natural y recuperación de información para evaluar la importancia de una palabra en un texto en relación con una colección más grande de textos. Se utiliza para vectorizar textos de tal forma que los modelos de Machine Learning puedan entender. Los componentes del TF-IDF son los siguientes

1. Term Frequency (TF): Mide la frecuencia con la que una palabra específica aparece en un texto, y por lo tanto, su importancia en el mismo. Se calcula de la siguiente manera:

$$TF(t,f) = \frac{\text{\# veces que aparece el término t en el texto f}}{\text{\# de términos del texto f}}$$

2. Item docuemnt frequency (IDF): Mide la importancia de una palabra en el conjunto de documentos. Palabras que aparecen en muchos documentos tendrán un IDF más bajo, ya que se consideran menos informativas. Su fórmula es la siguiente:

$$IDF(t,D) = ln\left(\frac{\text{\# total de textos en la coleccin D}}{\text{\# de textos que contienen el término t}}\right) + 1$$

3. TF-IDF: Finalmente, la puntuación asignada para cada término t en el texto f en la colección de textos D es igual a:

$$TF - IDF(t,f,D) = TF(t,f) * IDF(t,D)$$

Previamente a la presentación, se proporcionará una explicación detallada sobre la herramienta de generación de nubes de palabras, denominada "WordCloud". Para ello se utiliza la librería con el mismo nombre. En primera instancia, se define una lista específica de palabras que se excluyen de la representación visual generada por WordCloud. Estas son: "new", "car", "think", "elon", "week", "one"," day", "year", "today" ,"amp" ,"appl" ,"amzn" ,"fb" ,"like", "see", "stock", "my", "so", "was", "V", "are", "is", "user", "and", "but", "if", ""of", "to", "on", "for", "s", "as", "wa", "with", "or", "in", "it", "you", "that", "at", "is", "this", "from", "a", "what", "i". La razón detrás de esta exclusión radica en evitar la visualización de palabras que, a pesar de su frecuencia, no aportan significativamente a la interpretación del sentimiento de los tweets según los modelos utilizados. Este enfoque contribuye a presentar visualmente únicamente las palabras más relevantes y asociadas al sentimiento expresado en los tweets. Luego, con el input “text” se concatena todos los tweets limpios (columna ’cleanTweet’) correspondientes al sentimiento actual. Después, la clase wordcloud con su atributo generate(text) genera la nube de palabras basada en el texto combinado para el sentimiento actual. Finalmente, se procede a mostrar el gráfico del workcloud con la libreria matplotlib.

A. Support Vector Machine

Decidimos utilizar SVM ya que es una herramienta poderosa para el " Análisis de sentimientos" debido a su habilidad para manejar grandes cantidades de palabras y frases complejas. Es particularmente eficaz en situaciones donde las relaciones entre las palabras y los sentimientos no son claras o directas. Para el desarrollo de este modelo se realizó los siguientes pasos (IBM,2021)

Inicialmente, transformamos los tweets en una representación numérica utilizando un vectorizador TF-IDF. Esta técnica nos permitió convertir el texto en una matriz de características, resaltando la importancia relativa de cada palabra dentro del conjunto de datos, limitando el número de características a 3000 para mantener un equilibrio entre detalle y manejabilidad computacional. EL parámetro de penalización del término de error C es fijado en 3. El input *class_weight* es fijado en “balanced”, ya que ajusta automáticamente los pesos de las clases inversamente proporcional a su frecuencia, lo que puede ser útil cuando las clases están desequilibradas, como es

en este caso.

Posteriormente, integramos un clasificador de Máquina de Soporte Vectorial (SVM) con un kernel radial y un ajuste de peso de clase equilibrado. Este clasificador es conocido por su eficacia en encontrar la división óptima en un espacio de características, esto permite distinguir entre diferentes categorías de sentimientos en los tweets.

Para entrenar y evaluar el modelo, dividimos los datos en conjuntos de entrenamiento y prueba, manteniendo un 20% de los datos para pruebas a fin de evaluar la precisión del modelo en datos no vistos. Esta división se realizó de manera aleatoria pero controlada para garantizar la consistencia en las pruebas.

Una vez entrenado el modelo con los datos de entrenamiento, lo aplicamos al conjunto de prueba para realizar predicciones. Estas predicciones fueron luego comparadas con los valores reales para evaluar el rendimiento del modelo. Utilizamos métricas estándar como el informe de clasificación y la precisión para medir la eficacia del modelo en clasificar correctamente los tweets según su sentimiento. Este enfoque metodológico nos proporcionó una evaluación robusta y detallada de la capacidad del modelo para interpretar y clasificar datos de texto complejos.

Class	Precision	Recall	F1-score	Support
Negative	0.67	0.52	0.58	60
Neutral	0.69	0.60	0.64	57
Positive	0.59	0.75	0.66	81
Accuracy			0.64	198
Macro avg	0.65	0.62	0.63	198
Weighted avg	0.65	0.64	0.63	198

Table 3. Classification report

La precisión del modelo en la clasificación de sentimientos varía entre las categorías: es más alta para los tweets neutrales (69%), seguida por los negativos (67%) y más baja para los positivos (59%). Esto implica que el modelo es mejor para identificar correctamente los tweets que son neutrales, mientras que tiende a cometer más errores al clasificar los tweets positivos.

Para asegurar la consistencia de nuestros resultados por cada categoría se realizó el proceso de "wordcloud". Estos fueron los resultados:

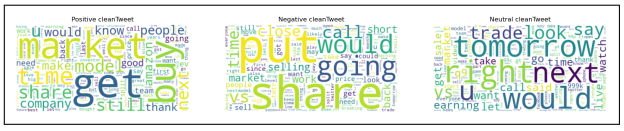

Fig. 7. Cloud

B. KNeighborsClassifier

El clasificador K-Nearest Neighbors (KNN) es útil en el análisis de sentimientos debido a su enfoque directo que clasifica los textos según la similitud con ejemplos previamente etiquetados. Funciona bien cuando los patrones de sentimiento son consistentes y los datos son densamente agrupados, ya que puede identificar y asignar sentimientos basándose en la proximidad a sus vecinos más cercanos (Díaz,2021).

Se aplico TF-IDF como en todos los casos para crear los feature vectors. De igual forma, el parámetro *max_features*es fijado en 3000, el cual limita el número de características, es decir, solo se considerarán las 3000 palabras más frecuentes en el conjunto de datos. El parámetro *n_neighbors* es fijado en 5, el cual indica que se considerarán los 5 vecinos más cercanos para realizar la clasificación

Una vez representados los textos, aplicamos un algoritmo que clasifica cada entrada basándose en la similitud con sus vecinos más cercanos. Este algoritmo selecciona un número pequeño de textos cercanos para determinar la categoría de sentimiento más probable, basándose en la premisa de que textos con sentimientos similares se encuentran próximos en el espacio de características.

Para validar la eficacia de nuestra metodología, dividimos los datos en un conjunto para entrenamiento y otro para pruebas, reservando un 20% para la evaluación. Después de entrenar nuestro modelo con los datos de entrenamiento, realizamos predicciones en el conjunto de pruebas para medir la precisión de nuestra técnica. Utilizamos medidas estándar de evaluación para cuantificar el rendimiento del modelo, proporcionando una comprensión clara de su capacidad para clasificar correctamente los sentimientos. Este método nos ofrece una forma pragmática y basada en ejemplos para entender cómo los patrones de lenguaje se correlacionan con diferentes emociones en los textos.

	Precision	Recall	F1-score	Support
Negative	0.54	0.48	0.51	58
Neutral	0.46	0.55	0.50	66
Positive	0.43	0.39	0.41	74
Accuracy			0.47	198
Macro avg	0.48	0.47	0.47	198
Weighted avg	0.47	0.47	0.47	198

Table 4. Classification report

la precisión en la clasificación de las etiquetas "negative", "neutral" y "positive" es del 54%, 46% y 43% respectivamente. Estos valores indican la proporción de predicciones correctas para cada clase en relación con todas las predicciones positivas para esa clase. En general, la precisión promedio ponderada es del 47%, lo que significa que el modelo tiene un rendimiento moderado en la clasificación de las tres clases

Para asegurar la consistencia de nuestros resultados por cada categoría se realizó el proceso de "wordcloud". Estos fueron los resultados:

Fig. 8. Cloud

C. LogisticRegression

Decidimos utilizar esta modelo pues es efectivo para análisis de sentimientos debido a su capacidad para manejar clasificaciones binarias, que es esencial en este contexto donde los sentimientos generalmente se categorizan como positivos o negativos. Este modelo matemático predice la probabilidad de una variable dependiente (como la polaridad del sentimiento) basada en una o más variables independientes (como las palabras o frases en un texto), lo que lo hace adecuado para entender y predecir las tendencias de los sentimientos en grandes volúmenes de texto (DATAtab,2021)

Cabe mencionar que seguimos utilizando un paso previo para crear los features vectors mediante TF-IDF. El modelo tiene los siguientes parámetros: *('logistic', LogisticRegression(C=5.0, class_weight='balanced'))*:
En esta etapa del pipeline, se utiliza el clasificador de regresión logística. El parámetro *C* es fijado en 5, el cual controla la inversa de la fuerza de regularización. Un valor más alto de C indica una regularización más débil. class_weight='balanced' ajusta automáticamente los pesos de las clases inversamente proporcional a su frecuencia.

Luego, aplicamos la Regresión Logística, un modelo robusto y eficiente para clasificación, con ajustes específicos para manejar mejor las características de nuestros datos. Este modelo se emplea para discernir patrones y tendencias en los datos, permitiendo así clasificar los sentimientos en categorías definidas.

Para garantizar la fiabilidad y generalización del modelo, dividimos los datos en conjuntos de entrenamiento y prueba. El conjunto de entrenamiento se utiliza para enseñar al modelo a reconocer y entender los patrones de sentimiento, mientras que el conjunto de prueba sirve para evaluar su rendimiento y precisión en datos no vistos.

Finalmente, evaluamos la eficacia del modelo utilizando métricas estándar como el informe de clasificación y la precisión. Estas métricas nos proporcionan una visión clara de cómo el modelo maneja las diferentes categorías de sentimientos y su capacidad general para realizar predicciones precisas.

	Precision	Recall	F1-score	Support
Negative	0.55	0.57	0.56	58
Neutral	0.56	0.48	0.52	66
Positive	0.59	0.65	0.62	74
Accuracy		0.57		
Macro Avg		0.57		
Weighted Avg		0.57		

Table 5. Model Classification Report

Los resultados de la columna "Precisión" en la tabla reflejan cómo el modelo clasificó cada categoría de sentimiento. Para la categoría "Negativo", la precisión fue del 55%, lo que indica que el modelo identificó correctamente el 55% de los casos negativos. En la categoría "Neutral", la precisión fue ligeramente superior, alcanzando el 56%, lo que sugiere una capacidad similar del modelo para identificar correctamente los casos neutrales. Finalmente, para la categoría "Positivo", la precisión fue del 59%, la más alta entre las tres categorías, mostrando una mejor capacidad del modelo para identificar correctamente los casos positivos

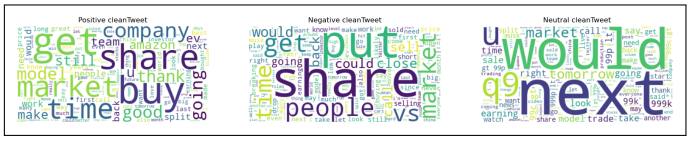

Fig. 9. Cloud

D. Naive Bayes

La fortaleza de Naive Bayes reside en su capacidad para aprender rápidamente a asociar ciertas palabras con sentimientos positivos o negativos a partir de los datos de entrenamiento, que a menudo son abundantes para las tareas de análisis de sentimientos. Además, puede hacer predicciones con un alto grado de precisión incluso con un conjunto de datos pequeño, lo que es útil para aplicaciones donde la toma de decisiones rápida es crucial. El problema de ello, es que nuestra base de datos tampoco consideramos que sea tan pequeña paraque disfrute de este beneficio. (Blackbox,2020)

Cabe mencionar que seguimos utilizando un paso previo para crear los features vectors mediante TF-IDF. La vectorización tiene el sigueinte código:

$$('tfidf', \text{TfidfVectorizer}(max\_df = 0.9, min\_df = 3))$$

Primero, El parámetro max_df con valor 0.9, elimina las palabras que aparecen en más del 90% de los textos, lo que ayuda a eliminar términos demasiado frecuentes que pueden no ser informativos. Segundo, el parámetro min_df con valor 3, elimina las palabras que aparecen en menos de 3 documentos, lo que ayuda a eliminar términos poco frecuentes que pueden ser ruido.

Posteriormente, se empleó el clasificador Naive Bayes Multinomial, adecuado para trabajar con las características de frecuencia obtenidas y conocido por su buen desempeño en la clasificación de texto. Se dividió el conjunto de datos en dos: uno para el entrenamiento del modelo y otro para la prueba, manteniendo una distribución aleatoria y controlada.

El modelo se entrenó con el conjunto de datos de entrenamiento, ajustándose a las relaciones entre las características vectorizadas y las etiquetas de sentimiento correspondientes. Tras el entrenamiento, se procedió a realizar predicciones sobre el conjunto de prueba para evaluar la capacidad del modelo de generalizar y reconocer los sentimientos en nuevos datos.

Finalmente, se evaluó el desempeño del modelo utilizando métricas estándar como el informe de clasificación, que proporciona una visión detallada de la precisión, el recall y la

puntuación F1 para cada clase de sentimiento, y la precisión general, que resume el porcentaje de predicciones correctas sobre el total.

	Precision	Recall	F1-score	Support
Negative	0.59	0.17	0.27	58
Neutral	0.69	0.30	0.42	66
Positive	0.41	0.85	0.56	74
Accuracy			0.47	
Macro Avg	0.56	0.44	0.42	198
Weighted Avg	0.56	0.47	0.43	198

Table 6. Model Classification Report

El análisis de la columna de precisión en el informe de clasificación revela una variabilidad notable en la capacidad del modelo para identificar cada clase correctamente. Con una precisión del 59% para la clase negativa, el modelo muestra una probabilidad moderada de identificar correctamente los tweets negativos. La clase neutral, con una precisión del 69%, indica que el modelo es relativamente más preciso al clasificar los tweets neutrales. Sin embargo, la precisión de la clase positiva desciende al 41%, sugiriendo que el modelo confunde con frecuencia los tweets positivos con otras emociones.

Por otro lado, para asegurar el valor del reporte, se decidio emplear una nube de datos

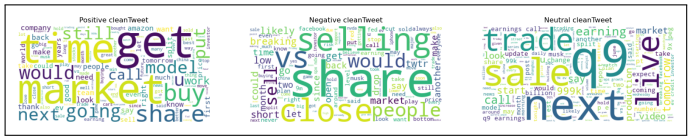

Fig. 10. Cloud

6. EJEMPLIFICACIÓN DEL MOVIMIENTO DE LOS PRECIOS CON LOS SENTIMENTALS

La metodología diseñada para estudiar la evolución del precio de un activo financiero en relación con el sentimiento del mercado expresado en los tweets, implica un proceso analítico estructurado. Inicialmente, se filtra la información de precios históricos para el activo seleccionado a partir de una fecha específica, ajustando el rango de tiempo de interés. Este conjunto de datos se reduce para incluir solo las entradas pertinentes a nuestras condiciones, lo que permite un análisis focalizado en el período y el activo de interés.

Paralelamente, se realiza un filtrado de los datos de tweets no etiquetados, seleccionando aquellos que corresponden al activo y la fecha en cuestión. Este paso es crucial para sincronizar el análisis de sentimiento con la evolución del precio del activo. Se lleva a cabo un conteo de los tweets clasificados como positivos, negativos y neutrales, proporcionando así una medida cuantitativa del estado de ánimo del mercado.

A continuación, se procede a visualizar la evolución del precio mediante un gráfico de series temporales, lo que ofrece una representación gráfica de la tendencia del precio a lo largo del tiempo. Además, se crea un gráfico de barras para ilustrar la distribución de los sentimientos expresados en los tweets, facilitando una comparación visual entre la percepción del mercado y el desempeño del precio del activo.

En última instancia, esta metodología busca correlacionar la información de sentimiento con los movimientos de precio, asumiendo que las percepciones y emociones del mercado reflejadas en las redes sociales pueden tener un impacto en las fluctuaciones del mercado financiero.

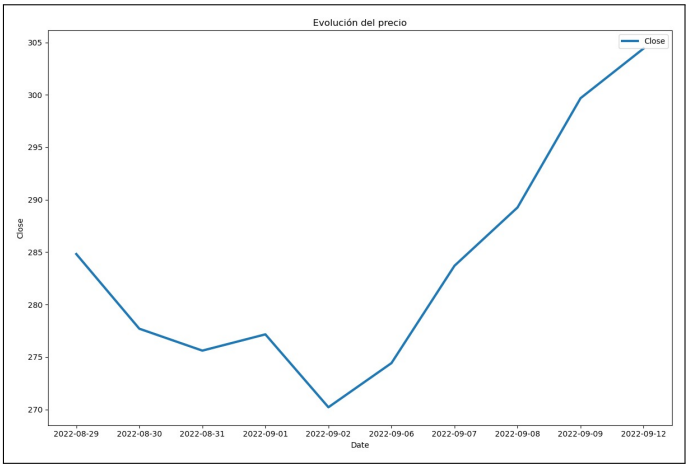

Fig. 11. Price´s Evolution

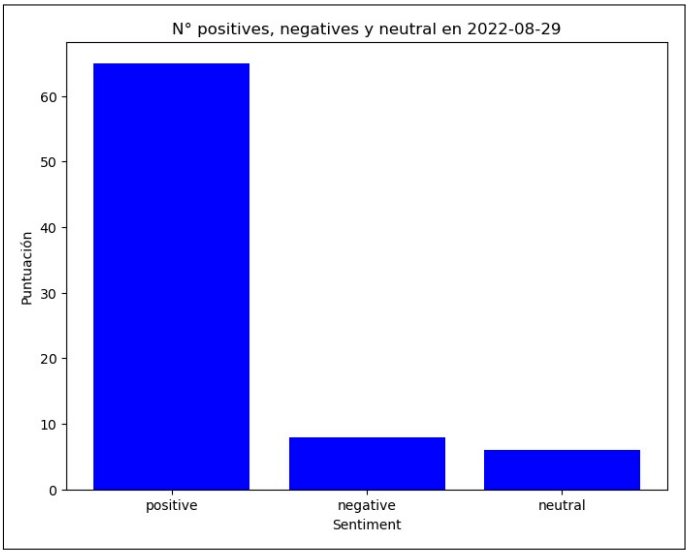

Fig. 12. Number of sentiments

7. CONCLUSIONES

El sentiment del mercado ha sido estudiado de manera empírica por las finanzas. Muchos investigadores de Data Science han considerado al sentiment como variable explicativa de los precios. Algunos modelos que se podrían utilizar para este objetivo son

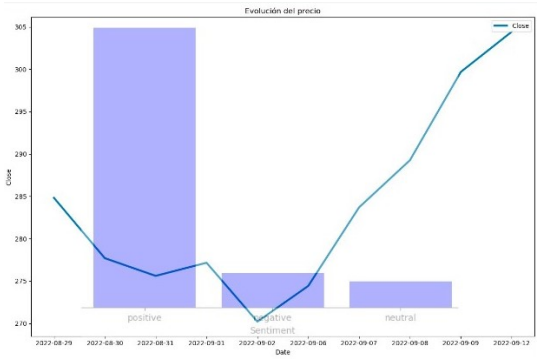

Fig. 13. Explanation of Results

los presentados en este proyecto: SVM, KNeighborsClassifier, LogisticRegression y Naive Bayes. Los practicioners están considerando la variable sentiment en sus modelos de predicción. Aún así, el análisis de lenguaje natural a través de inteligencia artifical aún están siendo desarrollados (general sentiment vs. market sentiment, sarcasmo, etc).

8. BIBLIOGRAFIA

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N project report, Stanford, 1(12), 2009.

Mittal, A., & Goel, A. (2012). Stock prediction using twitter sentiment analysis. Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf), 15, 2352.

Valencia, F., Gómez-Espinosa, A., & Valdés-Aguirre, B. (2019). Price movement prediction of cryptocurrencies using sentiment analysis and machine learning. Entropy, 21(6), 589.

Wilksch, M., & Abramova, O. (2023). PyFin-sentiment: Towards a machine-learning-based model for deriving sentiment from financial tweets. International Journal of Information Management Data Insights, 3(1), 100171.

IBM (2021). Funcionamiento de SVM. Recuperado de: https://www.ibm.com/docs/es/spss-modeler/saas?topic=models-how-svm-works

Díaz (2021). Algoritmo KNN – cómo funciona y ejemplos en Python. Recuperado de: https://www.themachinelearners.com/algoritmo-knn/

DATAtab (2021). Regresión logística. Recuperado de: https://datatab.es/tutorial/logistic-regression

Blackbox (2020). MODELOS NAIVE BAYES: PRECISIÓN E INDEPENDENCIA. Recuperado de: https://theblackboxlab.com/2022/03/30/modelos-naive-bayes/