# MODELO ANALÍTICO DE APRENDIZAJE AUTOMÁTICO PARA LA PREDICCIÓN EN CORTO PLAZO DEL SENTIDO DEL S&P 500

Manuela Piedrahita Vasco
Camilo Llerena Riascos

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Asesor
Hernan Felipe Garcia Arias, Doctor (PhD) en Ingeniería

Universidad de Antioquia
Facultad de Ingeniería
Especialización en Analítica y Ciencia de Datos
Medellín, Antioquia, Colombia
2023

| | |
|---|---|
| **Cita** | Piedrahita Vasco y Llerena Riascos [1] |
| **Referencia**<br>Estilo IEEE (2020) | [1] M. Piedrahita Vasco y C. Llerena Riascos, “Modelo analítico de aprendizaje estadístico para la predicción en corto plazo del sentido del S&P500”, Trabajo de grado especialización, Especialización en Analítica y Ciencia de Datos, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2023. |

creative commons BY NC SA

Especialización en Analítica y Ciencia de Datos, Cohorte IV.

Centro de Investigación Ambientales y de Ingeniería (CIA)

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia

Sistema de Bibliotecas

Centro de Documentación Ingeniería (CENDOI)

**Repositorio Institucional:** http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co
Rector: John Jairo Arboleda Céspedes.
Decano: Julio Cesar Saldarriaga Molina
Jefe departamento: Diego José Luis Botia Valderrama.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

TABLA DE CONTENIDO

RESUMEN ..... 6
ABSTRACT ..... 7
I. INTRODUCCIÓN ..... 8
II. DESCRIPCIÓN DEL PROBLEMA ..... 10
A. Problema de Negocio ..... 10
B. Aproximación desde la Analítica de Datos ..... 11
C. Formulación de pregunta de investigación ..... 12
III. OBJETIVOS ..... 13
A. Objetivo general ..... 13
B. Objetivos específicos ..... 13
IV. MARCO TEÓRICO ..... 14
A. Modelos ..... 14
B. Métricas ..... 20
V. METODOLOGÍA ..... 22
A. Pipeline Principal ..... 22
B. BASELINE ..... 24
C. Validación ..... 25
D. Iteraciones y Evolución ..... 25
E. Herramientas ..... 26
VI. RESULTADOS ..... 27
A. Datos ..... 27
B. DATASET ..... 27
C. Descriptiva ..... 28
D. Análisis de características ..... 29
VII. CONCLUSIONES ..... 37
VIII. REFERENCIAS ..... 38

# LISTA DE FIGURAS

Figura 1. Funcionamiento de las funciones de base no lineales en modelos de clasificación lineal……. 15
Figura 2. Diagrama de red neuronal…… 18
Figura 3. Modelo de ensamble de árboles……. 19
Figura 4: Pipeline Principal……..... 23
Figura 5: Procesamiento de datos……... 24
Figura 6: Subconjuntos de entrenamiento y testeo……. 28
Figura 7: Estructura de datos de entrada a modelo…… 28
Figura 8: Primera diferencia aplicada a la serie de tiempo…… 29
Figura 9. Análisis de correlación de variables ……30
Figura 10. Valores p, prueba de hipótesis de la correlación ……..……31
Figura 11. Curva Roc Modelo Red Neuronal MLP…… 34
Figura 12. Curva Roc Modelo Xgboost ……..……35
Figura 13. Gráfico de Curvas ROC de los modelos implementados ……36

# LISTA DE TABLAS

TABLA 1. Hiper parámetros Grid Search Regresión Logística…….. 32
TABLA 2. Hiper parámetros utilizados en GridSearch para Red Neuronal…….. 33
TABLA 3. Resultados mejores hiper parámetros Red Neuronal MLP…….. 33
TABLA 4. Hiper parámetros utilizados en GridSearch para Xgboost…… 34
TABLA 5. Resultados mejores hiper parámetros Xgboost…… 35
TABLA 6. Resumen de métricas finales en modelos implementados. ……..36

## RESUMEN

Se propone el análisis de tres modelos de machine learning, regresión logística, redes neuronales, y Xgboost, con el objetivo de encontrar un modelo analítico que aporte en la predicción en corto plazo del sentido del S&P 500. Desde la analítica de datos los modelos de operaciones de compra y venta de activos financieros consideran diferentes desafíos, la operación en el mercado por parte de los inversionistas está supeditada evidentemente a la volatilidad del mercado. A pesar de que en una operación de trading el usuario haga uso de múltiples herramientas para proteger su inversión, la volatilidad del mercado siempre será un factor a considerar y el riesgo estará latente, por ello, lo que se busca es disminuir la incertidumbre de la próxima operación. En consecuencia, los algoritmos aquí tratados están encaminados a dotar de herramientas técnicas al inversionista para que se logre maximizar beneficios económicos. El resultado obtenido sugiere que el modelo que se debe utilizar para tales fines es regresión logística ya que obtuvo el mejor desempeño, F1 socore 0,94 y AUC 0,98.

***Palabras clave*** — **regresión logística, redes neuronales, Xgboost, S&P 500.**

## ABSTRACT

We propose the analysis of three machine learning models, logistic regression, neural networks, and Xgboost, to find an analytical model that contributes to the short-term prediction of the direction of the S&P 500. From data analytics, the models of buying and selling operations of financial assets consider different challenges, the operation in the market by investors is subject to market volatility. Although the user uses multiple tools to protect his investment in a trading operation, market volatility will always be a factor to consider, and the risk will be latent. Therefore, what is sought is to reduce the uncertainty of the next operation. Consequently, the algorithms discussed here aim to provide investors with technical tools to maximize economic benefits. The result suggests that logistic regression is the model for such purposes since it obtained the best performance, F1 score of 0.94, and AUC of 0.98.


***Keywords*** — **logistic regression, neural networks, Xgboost, S&P 500**

# I. INTRODUCCIÓN

El S&P 500 es uno de los índices bursátiles más importantes del mundo y se compone de las 500 empresas más grandes cotizadas en las bolsas de valores de Estados Unidos. La predicción del sentido del S&P 500 (es decir, si subirá o bajará) en el corto plazo es un desafío clave para muchos inversores y traders.

Una forma de abordar este problema es mediante el uso de modelos de aprendizaje automático (Machine Learning). En particular, los modelos de aprendizaje supervisado pueden ser utilizados para predecir si el S&P 500 subirá o bajará en un período de tiempo específico. Esta predicción se constituye en una operación de compra y venta binarias en el mercado bursátil y representa una forma de inversión en la que se hace una apuesta sobre si el precio de un activo subirá o bajará en un plazo determinado (Vesga Bermejo & Chacón Ojeda, 2019). Algunas de las herramientas utilizadas para predecir el comportamiento del mercado son: análisis técnico, esta herramienta utiliza gráficos y datos históricos de los precios para identificar patrones y tendencias, este análisis se basa en la suposición de que los movimientos del mercado se repiten en patrones predecibles y, por lo tanto, se pueden utilizar para predecir futuros movimientos de precios. Análisis fundamental, esta herramienta utiliza datos financieros y económicos para evaluar el valor intrínseco de una empresa o activo financiero, es decir se basa en la suposición de que el valor de una empresa o activo financiero se puede medir a través de sus fundamentos financieros y económicos. Por último, análisis de sentimiento, que se basa en el estudio de las emociones y comportamientos de los inversores para predecir el comportamiento del mercado.

Algunos de los desafíos asociados con el uso de Machine Learning en las operaciones de compra y venta de activos financieros incluyen la necesidad de una gran cantidad de datos de calidad para entrenar los algoritmos, la complejidad de los modelos y la necesidad de ajustar y reajustar los modelos de forma constante para adaptarse a los cambios del mercado. Tanto el análisis técnico como el Machine Learning pueden ser herramientas útiles en las operaciones de compra y venta de activos financieros, pero cada uno tiene sus propios desafíos. Es importante que los operadores consideren cuidadosamente cuál herramienta es la más adecuada para sus necesidades y que mantengan una perspectiva a largo plazo al tomar decisiones de inversión (Francés Monedero, 2020).

El flujo de trabajo para la implementación de modelos analíticos de machine learning está comprendido así: a partir de las 5.826 observaciones se desarrolla un preprocesamiento de la información, luego se realiza una segmentación de los datos en subconjuntos de entrenamiento, testeo en el caso de la aplicación del modelo Regresión logística, Xgboost y Redes neuronales. Finalmente, a partir de las métricas consideradas a lo largo de este documento se evalúa la precisión de los pronósticos generados en cada uno de los modelos para determinar de esta forma, el más apropiado.

Se evidencia a lo largo de este documento el desempeño de cada uno de los modelos analizados por medio de métricas en comparación con los datos reales de la cotización y gráficas que evidencian la distancia entre los pronósticos obtenidos. Una vez el modelo ha sido entrenado, se puede utilizar para hacer predicciones sobre si el S&P 500 subirá o bajará en el futuro cercano con un 80% de probabilidad de acierto. Sin embargo, es importante tener en cuenta que los modelos de aprendizaje automático no son infalibles y siempre existirá un margen de error en las predicciones. Por lo tanto, es importante utilizar el modelo en conjunto con otras herramientas de análisis de mercado y tener una estrategia de inversión bien definida antes de tomar cualquier decisión de inversión basada en las predicciones del modelo.

# II. DESCRIPCIÓN DEL PROBLEMA

Los mercados financieros juegan un rol fundamental en el desarrollo de la economía global. A través de índices bursátiles como el Standard & Poor's 500 (S&P 500), Nasdaq, Dow Jones, entre otros, que representan portafolios conformados por las empresas más grandes en los Estados Unidos, es posible observar las tendencias que marcan la economía, no solo estadounidense, sino también global. Esto ocurre puesto que las empresas que hacen parte de estos portafolios ejercen en varios países alrededor del mundo. El S&P 500 actúa como un barómetro del mercado bursátil global y es un indicador clave de la salud económica, su rendimiento es seguido de cerca por inversores, analistas y economistas en todo el mundo (FAMA, 1991).

En lo que respecta al funcionamiento de inversión, una de las estrategias que existen y que será el foco del presente trabajo, es la denominada operación binaria y comienza cuando un inversor elige un activo financiero y decide si el precio subirá o bajará en un plazo específico (usualmente horas o días). Luego, realiza una inversión en base a su predicción y espera a que el plazo expire. Si la predicción es correcta, el inversor obtiene un retorno sobre su inversión, que puede ser del 70% al 90% en algunos casos. Si la predicción es incorrecta, el inversor pierde la cantidad invertida.

Las operaciones binarias son importantes porque ofrecen una forma relativamente simple y accesible para que los inversores participen en el mercado bursátil. Además, el riesgo y la recompensa están claramente definidos, lo que permite a los inversores tomar decisiones informadas sobre cuánto dinero están dispuestos a arriesgar. Sin embargo, es importante tener en cuenta que las operaciones binarias también tienen riesgos asociados y pueden ser una forma de inversión muy especulativa. Por lo anterior toma relevancia la aplicación de modelos analíticos de aprendizaje para que los inversores cuenten con información proyectada a futuro y con ello formular una estrategia clara antes de participar en este tipo de operaciones.

## *A. Problema de Negocio*

La operación en el mercado por parte de los inversionistas está supeditada evidentemente a la volatilidad del mercado, y este factor logra poner en aprietos a esta amplia comunidad alrededor del mundo. Por ello, múltiples herramientas están disponibles en las plataformas digitales de

trading para contrarrestar el efecto de la volatilidad frente a las posiciones que toman los inversionistas en el mercado. No obstante, aunque en una operación de trading el usuario haga uso de múltiples herramientas técnicas para proteger su inversión y asegurar el éxito de su operación, la volatilidad del mercado siempre será un factor a considerar y el riesgo siempre estará latente, por ello, lo que se busca, es disminuir la incertidumbre del próximo movimiento, es decir, que el problema de negocio al que se enfrentan los inversionistas es fundamentalmente el siguiente: estando en el tiempo t, debe obtener la predicción del valor de la tasa cambiaria en los tiempo t+1, t+2, t+3 … t+n, y con ello lograr las mejores condiciones del mercado para la implementación de su estrategia de trading y de esta forma maximizar sus ganancias (Cermeño & María José Roa, 2014).

*B. Aproximación desde la Analítica de Datos*

La conceptualización matemática de las herramientas y algoritmos de machine learning usadas para el análisis de datos, en muchos casos, sus diseños fueron elaborados en el siglo pasado y solo hasta hoy, gracias al boom tecnológico que impacta nuestra sociedad, se tiene la oportunidad de hacer uso de ellas. Puntualmente, en el caso de las redes neuronales, su génesis tuvo lugar a mediados del siglo X y sus primeras implementaciones se desarrollaron para la consideración de funciones linealmente separables (Corredor & Montoya, 2022).

Desde la analítica de datos los modelos de operaciones de compra y venta de activos financieros consideran desafíos y limitaciones en aspectos como la gran cantidad de datos. Para entrenar modelos de manera efectiva, se requiere una cantidad sustancial de datos históricos cuyo procesamiento puede consumir mucho tiempo, además, no todos los datos pueden ser relevantes o útiles para predecir el comportamiento del mercado. Los modelos no siempre proporcionan un rendimiento perfecto, estos pueden ser sensibles a cambios en los datos y a condiciones del mercado que pueden afectar la precisión y confiabilidad de las predicciones. Además, el mercado financiero está influenciado por una serie de factores no cuantificables, como eventos geopolíticos, cambios regulatorios o sorpresas económicas. Estos factores pueden afectar significativamente los precios de los activos y pueden ser difíciles de incorporar en un modelo de machine learning.

Dado lo anterior, se busca la construcción de modelos predictivos basados en aprendizaje de máquina y aprendizaje profundo, que predicen el comportamiento futuro del mercado. Su validación, aunque proviene de datos reales no puede estar supeditada únicamente al periodo que comprende el subconjunto de datos de entrenamiento y testeo, por el contrario, se debe ampliar sobre un conjunto de datos posterior a la serie de tiempo considerada, para realizar el contraste entre el pronóstico y los datos reales en el periodo mencionado anteriormente y verificar su precisión. En consecuencia, los algoritmos que se discuten a continuación están encaminados a dotar de herramientas técnicas al inversionista para la operación que pretende desarrollar y que, a partir de ello, logre obtener y maximizar beneficios económicos. No obstante, el marco de referencia para el comparativo de las predicciones será detallado en secciones posteriores.

### *C. Formulación de pregunta de investigación*

Con base en la anterior descripción del problema y los modelos de analítica existentes para abordar este tipo de problemáticas, se formula la siguiente pregunta de investigación:

*¿Cómo crear un modelo analítico de aprendizaje automático para la predicción en corto plazo del sentido del S&P 500?*

# III. OBJETIVOS

## *A. Objetivo general*

Desarrollar un modelo analítico para la predicción en corto plazo del sentido del S&P 500 utilizando técnicas de aprendizaje automático.

## *B. Objetivos específicos*

- Realizar un análisis de relevancia de características en series de tiempo S&P 500 utilizando enfoques de correlación y estadísticos.
- Implementar algoritmos de aprendizaje automático para la predicción en corto plazo de la tendencia en las señales S&P 500 utilizando enfoques de clasificadores lineales y no lineales.
- Evaluar la validez predictiva de los modelos de tendencia en corto plazo del S&P 500 utilizando métricas comúnmente utilizadas en el estado del arte para series financieras.

# IV. MARCO TEÓRICO

## *A. Modelos*

### ***Regresión logística***

En este caso se tomó como referencia para el desglose matemático de este modelo a Christopher M. Bishop (Bishop & Nasrabadi, 2006). Para un problema de clasificación de dos clases, la probabilidad de la clase xx se puede representar como una función sigmoide logística que actúa como una función lineal de $C_1$. De manera similar, para el caso de múltiples clases, la probabilidad posterior de la clase $Ck$ está dada por una transformación softmax de una función lineal de $x$. Con respecto a las elecciones de las densidades condicionales de clase $p(x|C_k)$, se utiliza la máxima verosimilitud, para encontrar los parámetros de las densidades al igual que las probabilidades a priori de clase $p(Ck)$, utilizando posteriormente el teorema de Bayes para encontrar las probabilidades de clase posteriores. Sin embargo, existe un enfoque alternativo el cual consiste en utilizar la forma funcional del modelo lineal generalizado explícitamente y determinar sus parámetros utilizando la máxima verosimilitud. Un algoritmo eficiente para esto es el algoritmo de Mínimos cuadrados ponderados iterativamente o IRLS Por sus siglas en inglés.

En la Figura 1 se muestra cómo actúan las funciones de base no lineales sobre modelos de clasificación lineal. El gráfico de la izquierda muestra el espacio original $(x1, x2)$ junto con los puntos de las dos clases en rojo y azul. En este espacio se definen dos funciones de base "Gaussianas" $\omega1(x)$ y $\omega2(x)$, cuyos centros se denotan con las cruces verdes y contornos con los círculos verdes. El gráfico de la derecha muestra la transformación en un nuevo espacio $(\omega1, \omega2)$ junto con la frontera lineal obtenida por el modelo de Regresión logística.

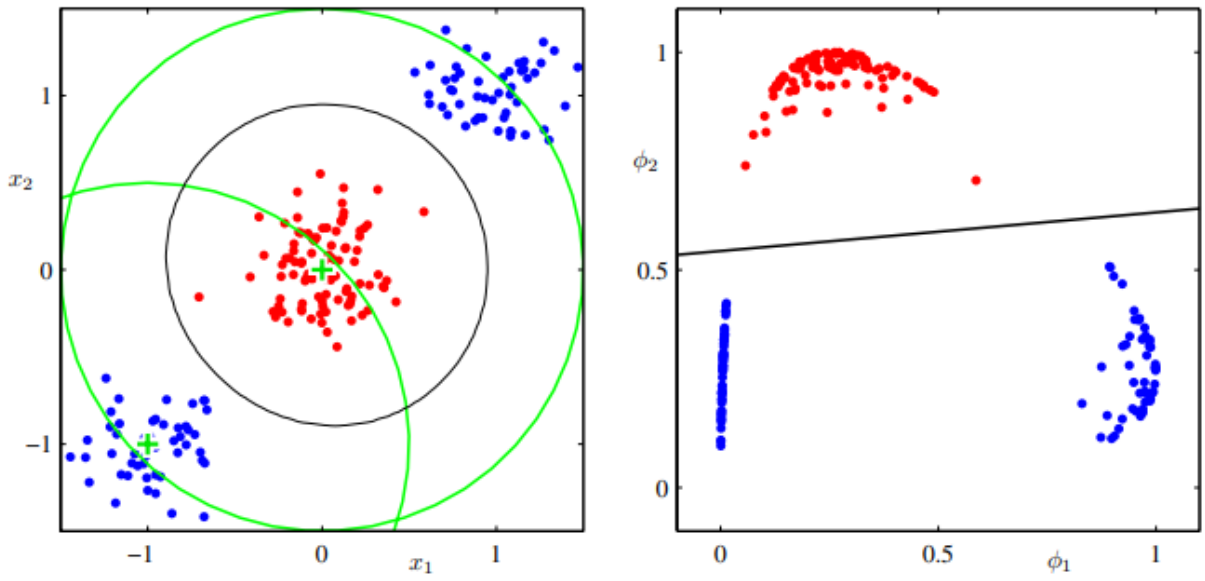


Figura 1. Funcionamiento de las funciones de base no lineales en modelos de clasificación lineal. Tomado de (Bishop & Nasrabadi, 2006).

Bajo suposiciones generales, la probabilidad posterior de la clase $C_1$ se puede escribir como una función sigmoide logística que actúa sobre una función lineal del vector de características φ, de modo que:

$$p(C_1|\varphi) = y(\varphi) = \sigma(\boldsymbol{w}^T\varphi) \quad (1)$$

Con $p(C_2|\varphi) = 1 - p(C_1|\varphi)$. En terminología estadística esto se conoce como *Regresión logística*, aunque se debe enfatizar que este modelo es más para clasificación que para regresión.

***Redes Neuronales***

Siguiendo la interpretación matemática de Christopher M. Bishop (Bishop & Nasrabadi, 2006), El término "Red neuronal" se remonta a los intentos por encontrar representaciones matemáticas del procesamiento de información en sistemas biológicos (McCulloch y Pitts, 1943; Widrow y Hoff, 1960; Rosenblatt, 1962; Rumelhart et al., 1986). Se ha utilizado para cubrir muchos modelos diferentes, muchos han sido cuestionados con respecto a su plausibilidad biológica. Sin embargo, el siguiente enfoque se centra en las Redes neuronales como modelos eficientes para el reconocimiento estadístico de patrones, en especial en las de tipo perceptrón multicapa quienes han demostrado ser de mayor valor práctico.

Inicialmente se considera la forma funcional del modelo de red, incluyendo la parametrización específica de las funciones de base y posteriormente se determinan los parámetros de la red dentro de un marco de máxima verosimilitud, lo que deriva en un problema de optimización no lineal. Esto requiere la evaluación de derivadas de la función de log-verosimilitud con respecto a los parámetros de la red, obteniéndose eficientemente con la técnica de retropropagación de errores. Los modelos lineales para regresión y clasificación se basan en combinaciones lineales de funciones de base no lineales fijas $\varphi_j(\boldsymbol{x})$, y tienen la siguiente forma:

$$y(\boldsymbol{x}, \boldsymbol{w}) = f(\sum_{j=1}^{M} w_j \varphi_j(\boldsymbol{x})) \quad (2)$$

Donde $f$ es una función de activación no lineal cuando es clasificación, $y$ la función identidad en el caso de regresión. El objetivo es extender el modelo haciendo que las funciones de base $\varphi_j(\boldsymbol{x})$ dependan de parámetros y luego hacer que estos parámetros se puedan ajustar junto con los coeficientes $w_j$ durante el entrenamiento. Las Redes neuronales utilizan como función de base la ecuación (2), de manera que cada función de base es en sí misma una función no lineal de una combinación lineal de las entradas, donde los coeficientes son parámetros adaptables.

Esto lleva a que el modelo básico de una red neuronal, se puede describir como una serie de transformaciones funcionales. Inicialmente se construyen M combinaciones lineales de las variables de entrada $x1, \ldots, xD$ de la siguiente forma:

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \quad (3)$$

Donde j=1..., M, y el superíndice (1) indica que los parámetros corresponden a la primera capa, siendo $w_{ji}^{(1)}$ los pesos y $w^{(1)}{}_{j0}$ los sesgos. El término $a_j$ se conoce como activación. Esta se transforma utilizando una función de activación no lineal y diferenciable *h* para obtener

$$z_j = h(a_j) \quad (4)$$

Estas cantidades $z_j$ son las salidas de las funciones de base en (2), que en Redes neuronales se llaman unidades ocultas. Las funciones no lineales h se escogen generalmente como funciones sigmoides como la sigmoide logística o la "*tanh*". Siguiendo la ecuación (2), estos valores se combinan linealmente nuevamente para obtener las activaciones de las unidades de salida.

$$a_k = \sum_{i=1}^{D} w_{kj}^{(2)} z_j + w_{k0}^{(2)} \quad (5)$$

Donde $k1, \ldots, k$ corresponde al número total de salidas. La ecuación (5) se refiere a la transformación de la segunda capa de la red. Finalmente, las activaciones de las unidades de salida se transforman con una función de activación adecuada dando como resultado un conjunto de salidas de la red $y_k$. La elección de la función de activación es determinada por la naturaleza de los datos y la distribución de las variables objetivo. Así, para problemas de regresión estándar, la función de activación es la identidad, por lo que $y_k = a_k$. De forma similar para muchos problemas de clasificación binaria, cada activación de unidad de salida se transforma usando una función sigmoide logística de manera que

$$y_k = \sigma(a_k) \tag{6}$$

Siendo,

$$\sigma(a) = \frac{1}{1+exp\,(-a)} \tag{7}$$

Por último, para problemas de clasificación multiclase, se utiliza una función de activación *softmax*. Combinando las diversas etapas se llega a la función de red general que, para funciones de activación de unidades de salida sigmoidales, toma esta forma:

$$y_k(\boldsymbol{x}, \boldsymbol{w}) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right) \tag{8}$$

Donde el conjunto de todos los pesos y sesgos han sido agrupados en un vector *w*. Así que un modelo de red neuronal es simplemente una función no lineal de un conjunto de variables de entrada $\{x_i\}$ a un conjunto de variables de salida $\{y_k\}$ controlada por un vector $\boldsymbol{w}$ de parámetros ajustables.

La Figura 2 muestra un diagrama de red neuronal. Las variables de entrada, ocultas y de salida se representan mediante nodos, y los parámetros de peso se representan mediante conexiones entre los nodos, en los que los parámetros de sesgo se indican mediante enlaces que provienen de variables adicionales de entrada y ocultas $x_0$ $y$ $z_0$. Las flechas indican la dirección del flujo de información a través de la red durante la propagación hacia adelante

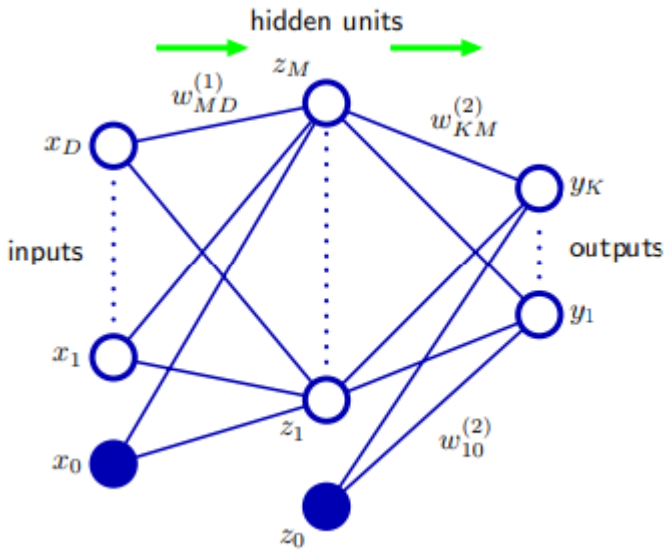


Figura 2. Diagrama de red neuronal. Tomado de (Bishop & Nasrabadi, 2006)

***Xgboost***

Los boosting trees son un método de aprendizaje automático muy efectivo y bastante utilizado. El Xgboost es un sistema escalable de extremo a extremo de los boosting trees, Este modelo es utilizado por los científicos de datos para lograr resultados de vanguardia en problemas que implican desafíos. Propone un algoritmo con un algoritmo novedoso consciente de la dispersión de los datos y un esquema de cuantiles ponderados para el aprendizaje de árboles aproximados. Además, cuenta con patrones de acceso a la memoria caché, compresión de datos y particionamiento para así lograr escalar más de miles de millones de datos utilizando menos recursos (Chen & Guestrin, 2016).

Objetivo de aprendizaje regularizado:

Para un conjunto de datos dado con n muestras y m características $D = \{(x_i, y_i)\}$ $(|D| = n, x_i \in R\ m, y_i \in R)$, un modelo de ensamble de árboles (figura 7), utiliza K funciones aditivas para predecir la salida.

$$\hat{y}_i = \varphi(\boldsymbol{x}_i) = \sum_{k=1}^{K} f_k(\boldsymbol{x}_i), \quad f_k \in F \tag{9}$$

Donde $F = \{f(\boldsymbol{x}) = w_{q(\boldsymbol{x})}\}(q: R^m \to T, w \in R^T)$ es el espacio de regresión de árboles (también conocido como CART). En esta expresión $q$, representa la estructura de cada árbol que le asigna un dato al índice correspondiente de la hoja. $T$ corresponde al número de hojas del árbol, $f_k$

representa una estructura de árbol independiente *q* y pesos de hojas *w*. En el caso de regresión, cada árbol tiene una puntuación continua en cada una de las hojas, representando con *wi* la puntuación de la i-ésima hoja, Para un ejemplo dado, se usarán las reglas de decisión en los árboles (dadas por q) para clasificar en las hojas y calcular la predicción final sumando la puntuación en las hojas correspondientes (dadas por w), como lo muestra la Figura 3. Con el objetivo de aprender el conjunto de funciones utilizadas en el modelo se minimiza el siguiente objetivo regularizado:

$$L(\omega) = \sum_i l(\hat{y}_i - y_i) + \sum_k \Omega(f_k), \ donde\ \Omega(f) = \gamma T + \frac{1}{2}\lambda \|\boldsymbol{w}\|^2 \qquad (10)$$

Donde l es una función de pérdida convexa diferenciable que mide la diferencia entre la predicción $\hat{y}_i$ y la variable de salida $y_i$. El segundo término, $\Omega$, penaliza la complejidad del modelo (es decir, las funciones de los árboles de regresión). El término adicional de regularización permite suavizar los pesos finales aprendidos para evitar el sobreajuste. Intuitivamente, el objetivo regularizado tenderá a seleccionar un modelo que emplee funciones simples y predictivas. Cuando el parámetro de regularización se establece en cero, el objetivo vuelve al tradicional boosting de árbol de gradiente.

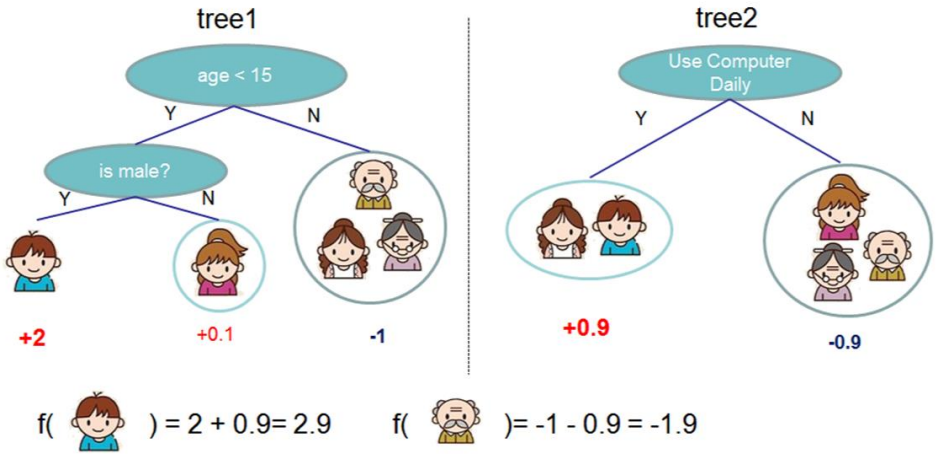


Figura 3. Modelo de ensamble de árboles. Tomado de Chen & Guestrin, 2016

Gradient Tree Boosting

El modelo de árboles que describe la ecuación (10) tiene incluido funciones como parámetros que no se pueden optimizar de manera tradicional en un espacio euclidiano, de esta manera se debe entrenar de manera aditiva. Teniendo que sea $\hat{y}_i^{(t)}$ la predicción la predicción de la i-ésima instancia en la t-ésima iteración, se agrega ft para minimizar el siguiente objetivo:

$$L^t = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(\boldsymbol{x}_i)) + \Omega(f_t) \quad (11)$$

Lo cual significa que se adiciona la $f_t$ que más mejora la ecuación (10), es decir el modelo de árboles.

*B. Métricas*

Considerando la variable de salida a pronosticar, en este caso la subida o bajada de la variable Close en un periodo de tiempo t + 1, es decir una salida de clasificación binaria. Por esta razón las métricas de desempeño a utilizar son F1 score y curva ROC.

F1 Score: Antes de definir la métrica de F1 Score es necesario remitirse a las definiciones de precisión y sensibilidad (Recall). La precisión se refiere a la tasa de verdaderos positivos en relación con el total de casos clasificados como positivos.

$$Precisión: \frac{Verdaderos\ Positivos\ (TP)}{Verdaderos\ Positivos\ (TP) + Falsos\ Positivos\ (FP)} \quad (12)$$

Recall es la tasa de verdaderos positivos (True Positive Rate, TPR). Es la proporción de casos positivos correctamente identificados, en relación con el total de positivos reales.

$$Recall = \frac{Verdaderos\ Positivos}{Verdaderos\ Positivos + Falsos\ Negativos} \quad (13)$$

El F1 score es una medida que combina la precisión y el recuerdo (recall) del modelo, se calcula con la media armónica entre estas dos medidas. Esta medida de calificación del modelo permite identificar la capacidad del modelo para clasificar correctamente las dos salidas del modelo. Toma valores entre o y 1, siendo 1 un mejor modelo frente a un modelo con calificación F cerca de 0, su ecuación es la siguiente:

$$F1 = \frac{2 * (precision * recall)}{precision + recall} \quad (14)$$

*Curva ROC:* La curva característica del receptor (receiver operating characteristic curve, ROC) es un resumen comúnmente utilizado para evaluar el equilibrio entre la sensibilidad y la especificidad. Es una representación gráfica de la sensibilidad frente a la especificidad a medida que variamos los parámetros de una regla de clasificación. La sensibilidad entendida como la tasa de Verdaderos Positivos, y la especificidad como la tasa de Falsos Positivos o tasa de Verdaderos Negativos.

La curva ROC muestra la relación entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) del modelo. TPR describe la tasa a la que el clasificador predice como "positivo" a las observaciones que son "positivas". FPR describe la tasa a la que el clasificador predice como "positivo" a las observaciones que en realidad son "negativas". Los clasificadores perfectos tienen una TPR de 1 y una FPR de 0 (Mathwork, 2022).

# V. METODOLOGÍA

## *A. Pipeline Principal*

En la Figura 4 se observan los diferentes pasos desarrollados durante este proyecto, que se divide en tres etapas principales: Adquisición y preprocesamiento de los datos, implementación de modelos de machine learning y deep learning y finalmente evaluación de las métricas. El primer paso que se debe hacer para la implementación de los diferentes modelos, en especial cuando se tienen diferentes variables predictoras de la variable de salida, es el escalamiento de los datos, éste permite reducir ruido, que el rango de valores de las variables sea definido y eliminar sesgos dados por las distribuciones propias de cada variable.

En el caso de los algoritmos de Deep Learning, estos son muy sensibles a las diferencias en rangos de las diferentes variables, así como el algoritmo Xgboost implementado en la clasificación binaria aquí propuesta como objeto de estudio. Por estas razones se realizó un escalamiento de los datos, tomando valores entre 0 y 1. Debido al análisis de correlación entre las variables iniciales del dataset (High, Low, Close, Volume), se decide utilizar una única variable para modelar la subida o bajada del precio de cierre (Close) en el periodo $t+1$.

Como hipótesis inicial, se plantea que el incremento o disminución de la variable Close puede ser explicada por ella misma tomando $\boldsymbol{n}$ periodos iniciales. De esta manera se crean rezagos de la variable Close utilizando la función “shift” de pandas (ver Figura 6).

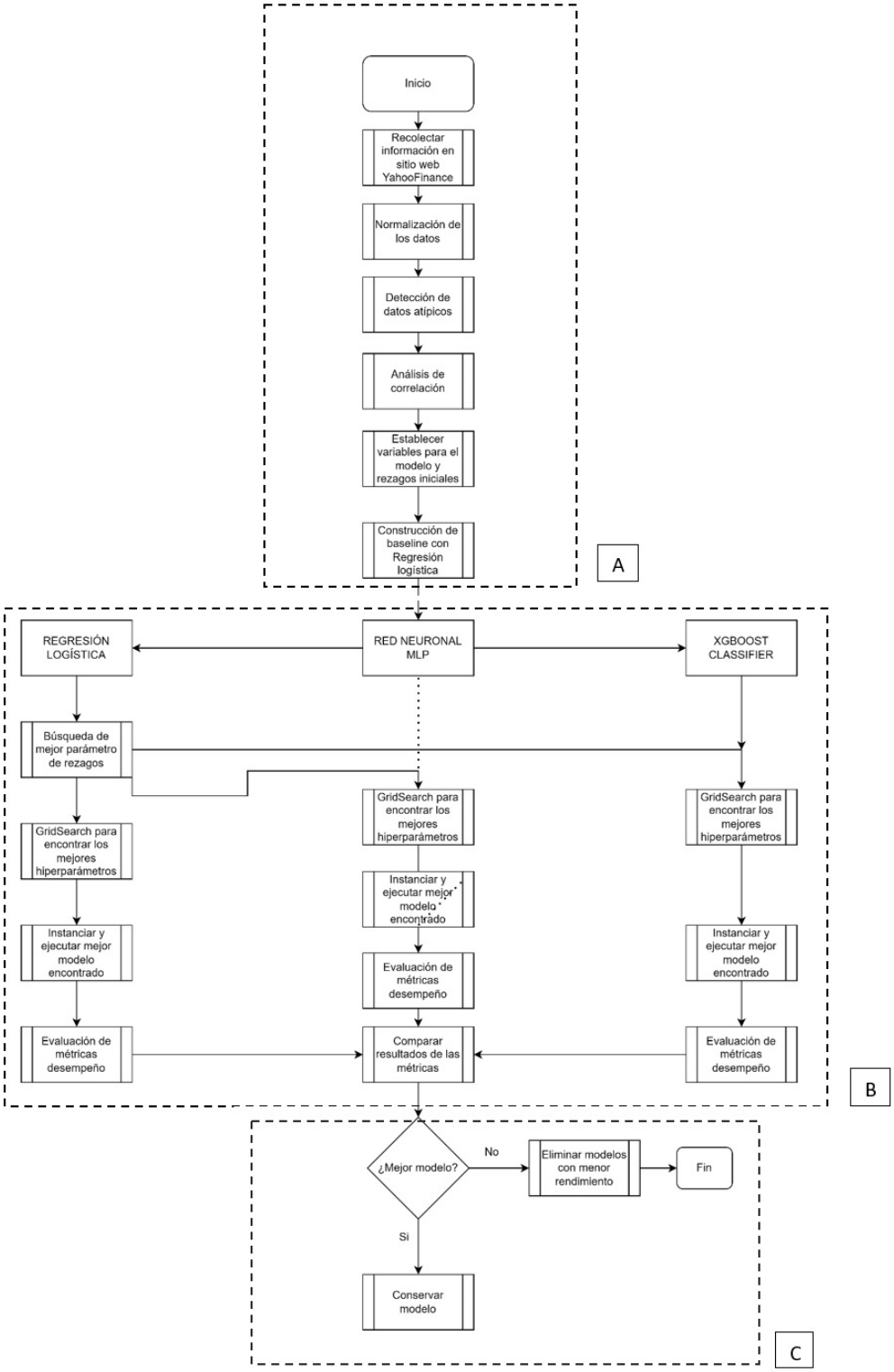


Figura 4: Pipeline Principal. Fuente: Elaboración propia

| Date | Close | t-1 | t-2 | t-3 |
|---|---|---|---|---|
| 2000-01-06 00:00:00-05:00 | 1403.449951 | 1402.109985 | 1399.420044 | 1455.219971 |
| 2000-01-07 00:00:00-05:00 | 1441.469971 | 1403.449951 | 1402.109985 | 1399.420044 |
| 2000-01-10 00:00:00-05:00 | 1457.599976 | 1441.469971 | 1403.449951 | 1402.109985 |
| 2000-01-11 00:00:00-05:00 | 1438.560059 | 1457.599976 | 1441.469971 | 1403.449951 |
| 2000-01-12 00:00:00-05:00 | 1432.250000 | 1438.560059 | 1457.599976 | 1441.469971 |

Figura 5: Procesamiento de datos. Fuente: Elaboración propia

Posteriormente se implementó el algoritmo de Regresión logística como línea base considerando un número inicial de rezagos y sin previa optimización de hiperparámetros. El recuadro A indiaca el preprocesamiento de los datos el cual consistió en normalización, detección de atípicos, y correlación, que permitió seleccionar las variables a tratar en el modelo. El recuadro B describe la metodología utilizada para la búsqueda óptima del mejor parámetro de rezagos, procediendo después a la consideración de los modelos de Redes Neuronales y Xgboost. Finalmente se realizó una evaluación de las métricas para determinar el mejor modelo, como se muestra en el recuadro C, de la Figura 5.

*B. BASELINE*

La primera iteración se realizó con el algoritmo de clasificación Regresión logística y a los datos no se les aplicó ninguna técnica de normalización o balanceo, inicialmente los algoritmos se trabajan con hiperparámetros por defectos.

El reto en este punto es identificar los mejores datos para entrenar los modelos, pues solo se podrá generar eficiencia en los algoritmos si se trabaja con la información correcta. La matriz de correlación es una de las técnicas utilizadas en machine learning para seleccionar las variables correctas, (Zhou, 2021) se debe aclarar que la matriz no hace una selección definitiva de variables, simplemente es un concepto matemático que se encarga de medir la relación entre todas las variables. En el ejercicio como se ve en la Figura 4, al utilizar esta herramienta se encuentra una correlación alta de 0.99 entre las variables open, high, low, y close. Con el objetivo de enfocar el análisis y delimitar el alcance del presente trabajo se toma la decisión de eliminar de los datos las

variables open, high, y low, teniendo en cuenta la gran similitud que poseen sus datos y la relación en la información suministrada.

La importancia de los ejercicios realizados en estas primeras iteraciones sirvió como fundamentos para el inicio de una búsqueda de hiperparámetros en los modelos que se desarrollarán.

*C. Validación*

Se podría validar los modelos realizando una repartición de los datos aleatoriamente en dos grupos (Entrenamiento y evaluación), lo cual se puede considerar el medio más práctico y simple; sin embargo, condicionan la estimación del error debido a que su valor es dependiente al conjunto de datos que se utiliza en el entrenamiento y validación. Los datos ya organizados son divididos en entrenamiento y validación (train-test) 70% - 30% esto se realiza con una función muy práctica de sklearn train_test_split(). La validación cruzada se realizará una vez sea seleccionado el mejor modelo, esto permitirá que independientemente de la partición de datos que se haya hecho, sea posible entrenar y validar con todo el conjunto de datos completos. También llamado como cross-validation consiste en dividir los datos de entrenamiento en iteraciones.

*D. Iteraciones y Evolución*

En la primera iteración el enfoque fue la limpieza de datos y se aplicaron técnicas de normalización, detección de datos atípicos, visualización de distribución, correlación, y eliminación de columnas que no aporten al análisis. Con el objetivo de realizar una primera evaluación de la calidad de datos y el comportamiento de la aplicación de un modelo de clasificación, la segunda iteración se basó en la valoración del modelo de Regresión logística sobre los datos sin normalizar y utilizando hiperparámetros por defecto. El principio de esta iteración es identificar un primer resultado con el cual se van a comparar los resultados subsiguientes, y se constituirá en el modelo base a superar en cuanto a las métricas de desempeño, dicho en otras palabras este paso proporciona una estructura básica para el modelo final, establece expectativas para la precisión y el rendimiento del modelo final, ahorra tiempo y recursos al proporcionar una estructura básica en lugar de comenzar desde cero, y facilita la comprensión de los datos y el problema que se está tratando de resolver (Goodfellow et al., 2016)

En las siguientes iteraciones, ya con el dataset organizado se trabaja con tres modelos, Regresión logística, Redes Neuronales y Xgboost, todos con el cálculo de los mejores hiperparametros. El modelo de Regresión logística metodológicamente aporta desde su interpretabilidad, lo que significa que los desarrolladores pueden entender fácilmente cómo el modelo realiza las predicciones, esto es importante en aplicaciones donde se necesitan explicaciones detalladas de cómo se toman las decisiones. Por su parte las Redes Neuronales tienen la capacidad de aprender características complejas de los datos de entrada, lo que puede mejorar la precisión del modelo de clasificación, al aprender características complejas, las Redes neuronales pueden identificar patrones en los datos que otros modelos pueden perder (LeCun et al., 2015). Finalmente, el modelo Xgboost puede mejorar la precisión del modelo de clasificación en comparación con otros modelos al optimizar el modelo mediante la combinación de múltiples árboles de decisión y minimizar el error en cada paso (Kelleher et al., 2018).

En cada modelo se realiza la búsqueda de los mejores hiperparámetros, se calculan las métricas Accuracy, curva ROC, F1 y se realiza validación cruzada. A partir de este análisis comparativo se selecciona el mejor modelo.

*E. Herramientas*

La principal herramienta utilizada en este proyecto es Google Collaboratory, que está basado en el proyecto de código abierto Jupyter. Al hacer uso de esta herramienta, se le proporciona al usuario una máquina virtual para la ejecución de código Python en la versión 3.7.13. Google Colab genera un ambiente de librerías preinstaladas en el espacio asignado, que el usuario solo debe importar bajo la sintaxis de Python.

- Para la creación del modelo se utilizó:
- Librerías de python - Pandas - Numpy - Scikit-learn, Xgboost

Respecto a la visualización en todas las etapas del procesamiento y en cada uno de los experimentos desarrollados, se implementó la librería Matplotlib y Seaborn.

# VI. RESULTADOS

## *A. Datos*

Los datos utilizados en este proyecto representan el cierre de cotización diaria del índice S&P500, estos son públicos y para obtenerlos se debe instalar el paquete de Yahoo Finance (!pip install yfinance) e importar la librería *yfinance as yahooFinance.* El usuario determina la fecha de inicio y final para el respectivo análisis y con ello el aplicativo se conecta a la base de datos que contiene seis columnas: date, open, high, low, close, y volume.[1]

La columna "close", corresponde con el cierre de la cotización del día anterior, de acuerdo con una revisión preliminar de la base de datos, esta puede ser igual al valor de la columna "apertura" para del día siguiente, sin embargo, esto no se cumple en todos los casos. Las columnas "high" y "low" corresponden con la cotización del S&P 500 diariamente y por último la columna "volume" corresponde a la cantidad de operaciones de compra y ventas realizadas en un día.

De acuerdo a las características inherentes de la data (serie de tiempo) y de los algoritmos para su modelamiento, este proceso se desarrolla con base en la columna "Close", y consecuentemente las métricas de testeo y validación frente a los datos reales son realizadas en contraste con dicha columna.

## *B. DATASET*

En consideración del dataset y la condición de serie de tiempo que presenta los datos, la creación de los subconjuntos de datos debe realizarse sin alterar el orden secuencial de los registros en el tiempo, de esta forma los algoritmos usados para el modelamiento de los datos logran reconocer los patrones secuenciales existentes y definir de esta forma un pronóstico cercano a la cotización del tiempo t+1.

Para el entrenamiento de datos inicialmente se ha determinado seleccionar 5 etapas temporales diferentes, cada una con un total de 252 observaciones. En la Figura 6 se indica esta definición.

[1] Yahoo Finance *S&P 500 (^GSPC)*: https://Finance.Yahoo.Com/Quote/%5EGSPC/History?P=%5EGSPC.




Figura 6: Subconjuntos de entrenamiento y testeo. Fuente: Elaboración propia

Adicional a lo anterior, se construye un dataframe con base en una observación en el tiempo *t*, y a partir de la búsqueda del mejor resultado en términos *grid_search.best_score* se generan veintiún (21) periodos rezagados para cada observación, es decir, cada cotización de cierre (variable a predecir) contiene las cotizaciones de cierre de los últimos veintiún días (variables predictoras), tal como se observa en la Figura 7.

| Date | Close | t-1 | ... | t-12 | t-13 | t-14 | t-15 | t-16 | t-17 | t-18 | t-19 | t-20 | t-21 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2000-02-02 00:00:00-05:00 | 1409.119995 | 1409.280029 | ... | 1465.150024 | 1449.680054 | 1432.250000 | 1438.560059 | 1457.599976 | 1441.469971 | 1403.449951 | 1402.109985 | 1399.420044 | 1455.219971 |
| 2000-02-03 00:00:00-05:00 | 1424.969971 | 1409.119995 | ... | 1455.140015 | 1465.150024 | 1449.680054 | 1432.250000 | 1438.560059 | 1457.599976 | 1441.469971 | 1403.449951 | 1402.109985 | 1399.420044 |
| 2000-02-04 00:00:00-05:00 | 1424.369995 | 1424.969971 | ... | 1455.900024 | 1455.140015 | 1465.150024 | 1449.680054 | 1432.250000 | 1438.560059 | 1457.599976 | 1441.469971 | 1403.449951 | 1402.109985 |
| 2000-02-07 00:00:00-05:00 | 1424.239990 | 1424.369995 | ... | 1445.569946 | 1455.900024 | 1455.140015 | 1465.150024 | 1449.680054 | 1432.250000 | 1438.560059 | 1457.599976 | 1441.469971 | 1403.449951 |
| 2000-02-08 00:00:00-05:00 | 1441.719971 | 1424.239990 | ... | 1441.359985 | 1445.569946 | 1455.900024 | 1455.140015 | 1465.150024 | 1449.680054 | 1432.250000 | 1438.560059 | 1457.599976 | 1441.469971 |

5 rows × 22 columns

Figura 7: Estructura de datos de entrada a modelo. Fuente: Elaboración propia

*C. Descriptiva*

Dada la condición de serie de tiempo que tienen los datos analizados, resulta innecesario realizar gráficas para la construcción de un análisis exploratorio de los datos, dado que no aporta información relevante al análisis de la serie de tiempo. En su lugar, se presentan a continuación pruebas estadísticas para determinar la factibilidad del modelamiento del proceso estocástico. La condición de estacionariedad de una serie de tiempo define si esta presenta media y varianza constante a través del periodo de tiempo analizado. La condición de estacionariedad define si dicha serie de tiempo pudo o no considerar un modelo autorregresivo para predecir su comportamiento. Para ello, la estacionariedad es evaluada con la prueba de hipótesis de Dicky - Fuller en un nivel de confianza del 95%, donde:

$Ho$: *La serie de tiempo no es estacionaria*

$Hi$: *La serie de tiempo es estacionaria*

Este contraste de hipótesis arrojó un p-valor de 0.9913, siendo éste mayor al nivel de confianza del 95% establecido para la prueba de hipótesis. En consecuencia, la evidencia estadística señala el rechazo de la hipótesis nula, es decir su media y varianza si cambian en el tiempo y la serie de tiempo no es estacionaria. Existen transformaciones de los datos que permiten convertir en estacionaria una serie de tiempo que no lo es, puntualmente se utiliza la técnica de diferenciación que está definida como la resta entre el periodo t y el t-1. En la Figura 8 se muestra gráficamente cómo queda determinada la nueva serie de tiempo.

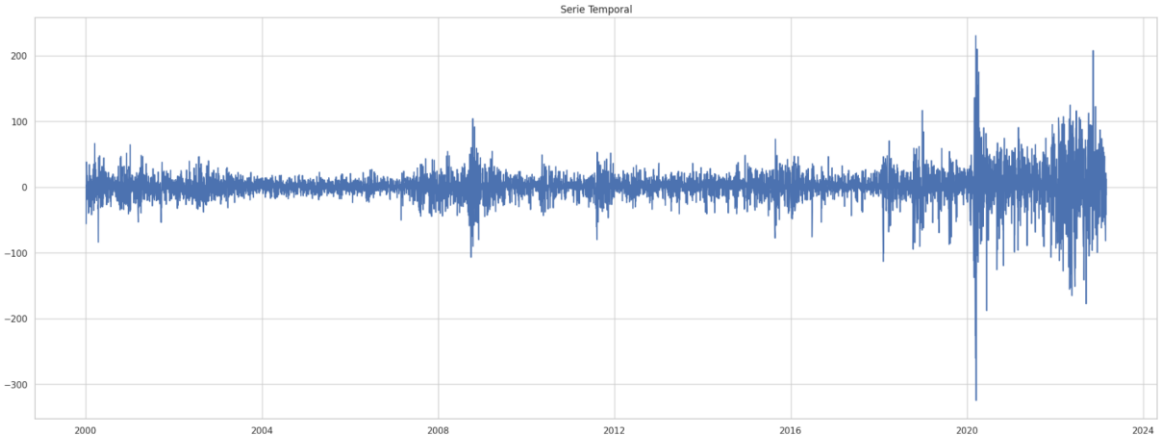


Figura 8: Primera diferencia aplicada a la serie de tiempo. Fuente: Elaboración propia

Al aplicar nuevamente la prueba Dicky - Fuller a la serie de tiempo con la primera diferencia se encuentra que el valor-p es cercano a 0, rechazando la hipótesis nula. Uno de los aspectos más importantes para la definición de los parámetros del modelo autorregresivo, es la estimación de la autocorrelación de la serie de tiempo que determinan sus rezagos en el tiempo. Esta medida es útil como punto de referencia para predeterminar en los modelos de deep learning los retrasos en el tiempo.

*D. Análisis de características*

Inicialmente se decide eliminar del dataset original, la variable “Dividends” y “Stock Splits” ya que como supuesto inicial no se considera alguna relación con la variable “Close”, posteriormente se procede a realizar un análisis de correlación entre las variables “Close”, “Open”, “High”, “Low” y “Volume”.

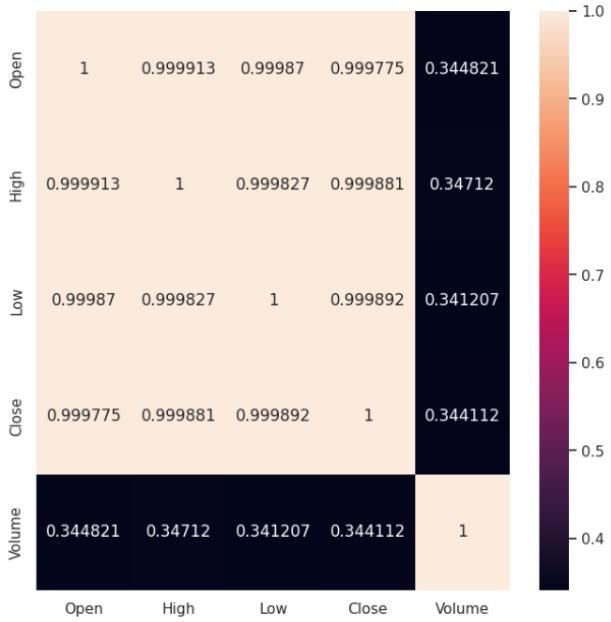


Figura 9. Análisis de correlación de variables. Fuente: Elaboración propia.

La Figura 9, nos muestra el resultado del análisis de correlación entre las variables, donde se observa una correlación superior al 0.9 entre las variables “Close”, “Open”, “High” y “Low”, es decir que la información contenida entre estas variables es similar, por lo que procedemos a continuar el análisis únicamente con la variable Close. La Figura 10, muestra los resultados de los valores p, para la prueba de correlación. Al ser inferior a 0.05, se puede concluir que si existe dicha relación entre las anteriores variables y de esta manera se puede tomar la decisión de no incluir las variables similares en el análisis.

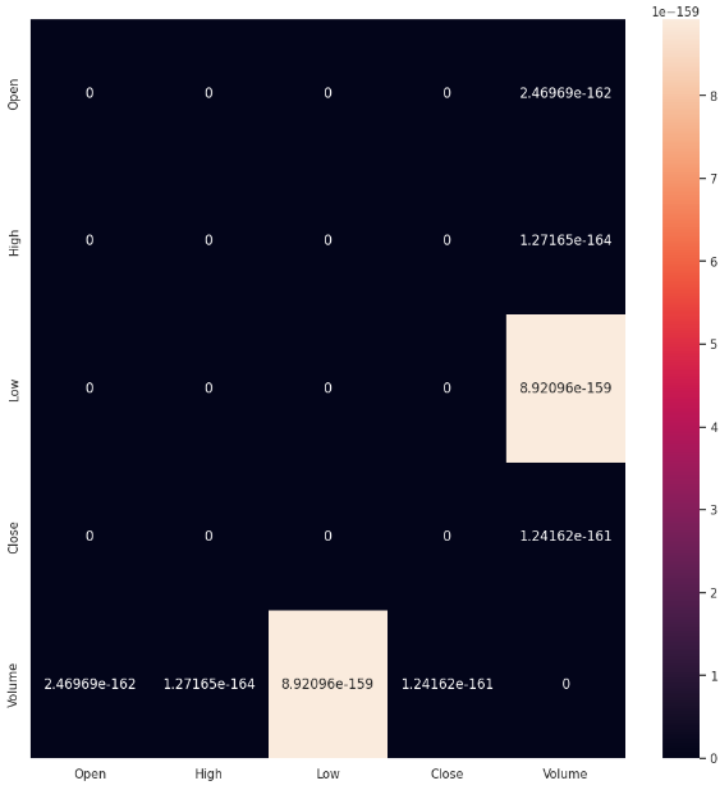


Figura 10. Valores p, prueba de hipótesis de la correlación. Fuente: Elaboración propia.

### *Modelo Baseline*

Teniendo la variable “Close” como la única a tomar en cuenta para predecir la subida o bajada al dia siguiente de ella misma, se procede a rezagar dicha variable, con una escogencia arbitraria inicial de 3 rezagos, utilizando un modelo de Regresión logística sin optimización de hiper parámetros. Este modelo inicial arrojó una calificación F1 de 0.8, el cual se toma como base para evaluar los modelos implementados posteriormente.

### *Modelo Regresión logística*

Se realizó un Grid Search, en donde se buscó obtener los mejores hiper parámetros en cuanto al parámetro C, y la norma de penalidad. Se adiciona al Grid Search la búsqueda del número de rezagos a considerar de la variable “Close” que optimiza los resultados. La Tabla 1 muestra los valores que se tuvieron en cuenta para los diferentes parámetros. El parámetro C controla el sobreajuste en el modelo, entre más pequeño mayor es la regularización, evaluando este parámetro en la norma de penalidad l1 o l2.

**TABLA 1.**

Hiper parámetros Grid Search Regresión Logística.Fuente: Elaboración propia

| Número de Rezagos | Parámetro C | Penalidad |
|---|---|---|
| 3 | 0,1 | l1 |
| 14 | 1 | l2 |
| 21 | 10 | |

Con respecto al Grid Search, se utilizó el método Time Series Split de la librería de Scikit-learn, ya que de la manera como fue modelado este problema, se requería de una partición de los datos que no fuera aleatoria sino por el contrario, los datos fueran contiguos, en este caso el conjunto de train se conforma de tramos de 200 datos contiguos de un total del 15% del dataset inicial.

El mejor resultado de la validación cruzada se obtuvo con 21 rezagos de la variable de salida, un valor de C de 0.1 y una penalidad l2. Posteriormente se utilizaron los hiper parámetros hallados en el Grid Search y entrenando el modelo de regresión logística con los datos de entrenamiento definidos inicialmente, se obtuvo una calificación F1 de 0.94 y un área bajo la curva de 0.98 para la curva ROC que se muestra en la Figura 10.

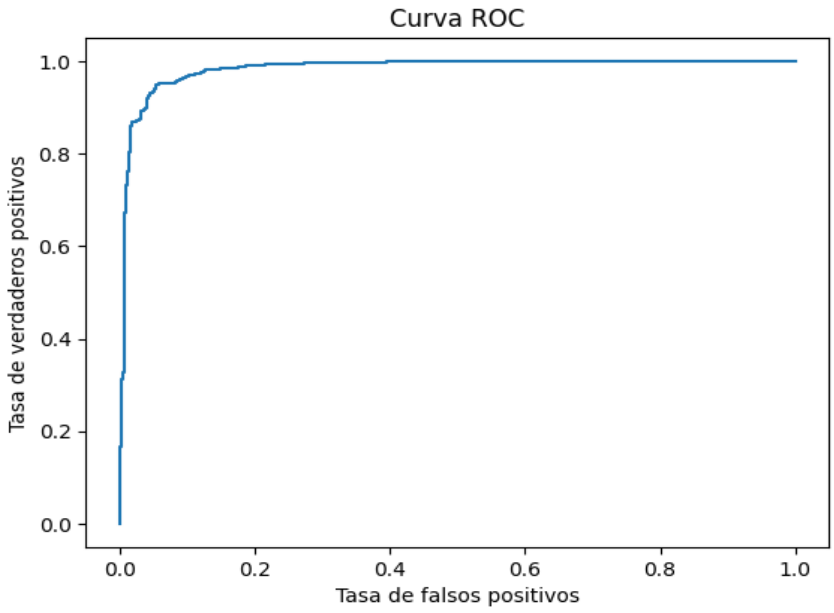


Figura 10. Curva Roc Modelo Regresión logística. Fuente: Elaboración propia.

***Modelo con Red Neuronal***

Realizando la optimización de los resultados con los diferentes rezagos de la variable "Close", se decide fijar 21 rezagos para evaluar tanto el modelo de Red Neuronal como el modelo Xgboost. Con respecto a la red neuronal también se realizó una validación cruzada o Grid Search con la combinación de hiper parámetros que se muestra en la Tabla 2.

**TABLA 2.**

Hiper parámetros utilizados en GridSearch para Red Neuronal. Fuente: Elaboración propia.

| Hidden layer sizes | Activation | Alpha | Learning rate | Solver | Max iter |
|---|---|---|---|---|---|
| 100 | logistic | 0.001 | constant | adam | 1000 |
| 200 | relu | 0.01 | invscaling | | 2000 |

Los mejores resultados se obtuvieron con la arquitectura de red y los hiperparámetros mostrados en la tabla 3, los cuales implican que el tamaño de la capa oculta es 100, la función de activación utilizada es la logística, y un valor de Alpha igual a 0.001 que indica una tasa de aprendizaje pequeña. Dando como resultado un 0,95 de calificación f1 para el GridSearch.

**TABLA 3.**

Resultados mejores hiper parámetros Red Neuronal MLP. Fuente: Elaboración propia.

| Hidden layer sizes | Activation | Alpha | Learning rate | Solver | Max iter |
|---|---|---|---|---|---|
| 100 | logistic | 0.001 | constant | adam | 1000 |

Al utilizar estos hiperparámetros para el entrenamiento de la red neuronal con los datos de entrenamiento y testeo se obtuvo una calificación f1 de 0.93 y un área bajo la curva ROC de 0.98. Dados estos resultados y observando la Figura 11 se puede inferir que el resultado presenta un excelente rendimiento del modelo de clasificación binaria, debido a que es capaz de reconocer las dos clases de la variable de salida (alta capacidad de discriminación), es decir implica que la tasa de falsos positivos (Especificidad) es baja en comparación con la tasa de verdaderos positivos (Sensibilidad).

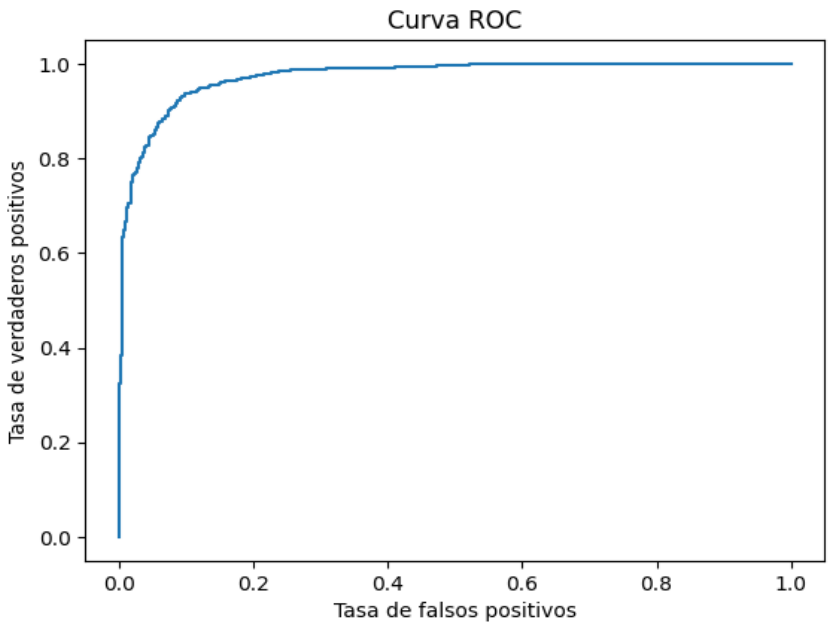


Figura 11. Curva Roc Modelo Red Neuronal MLP. Fuente: Elaboración propia.

***Modelo con Xgboost***

Para realizar el GridSearch y obtener dentro de un conjunto de hiper parámetros, aquellos que optimicen los resultados del modelo, se probaron los valores mostrados en la tabla 4.

**TABLA 4.**

Hiper parámetros utilizados en GridSearch para Xgboost. Fuente: Elaboración propia.

| n_estimators | Max_depth | Learning-rate | Gamma | Subsample | colsample_bytree |
|---|---|---|---|---|---|
| 100 | 3 | 0,01 | 0 | 0,5 | 0,5 |
| 200 | 5 | 0,1 | 0,1 | 1 | 1 |

Los mejores resultados se obtuvieron con los hiper parámetros mostrados en la Tabla 5, teniendo un 0,88 de calificación f1 con dicha combinación para el GridSearch. Finalmente implementando estos hiper parámetros en el modelo xgboost para el dataset con la partición inicial se obtuvieron resultados de 0.74 para la calificación F1 y 0.6 en el área bajo la curva.

**TABLA 5.**

Resultados mejores hiper parámetros Xgboost. Fuente: Elaboración propia.

| n_estimators | Max_depth | Learning-rate | Gamma | Subsample | colsample_bytree |
|---|---|---|---|---|---|
| 100 | 3 | 0,01 | 0 | 0,5 | 0,5 |

La Figura 11, muestra la curva ROC obtenida para el modelo de Xgboost luego de hacer la optimización de hiperparámetros. Su resultado indica que el modelo tiene cierta capacidad de discriminar entre las diferentes salidas, sin embargo; para lo que se pretende lograr se descarta este modelo como definitivo.

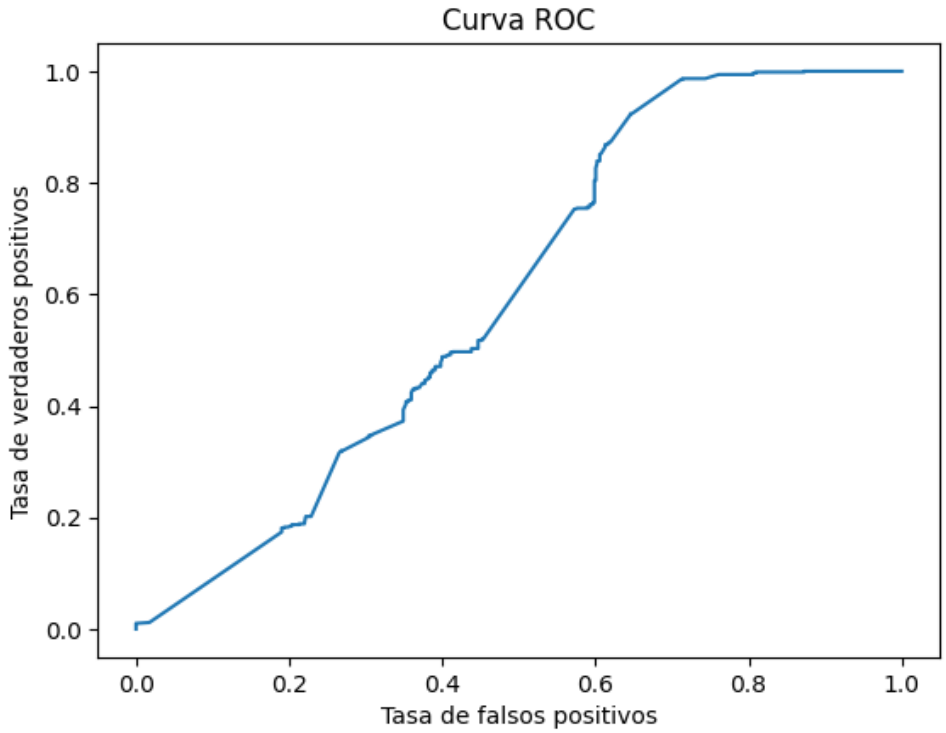


Figura 12. Curva Roc Modelo Xgboost. Fuente: Elaboración propia.

Se consolida el proceso de experimentación en la tabla 6, en la que se describen las métricas de evaluación, en este caso la calificación F1 y el área bajo la curva ROC (AUC). Con base en esta información, se observa que los modelos de regresión logística y Red Neuronal MLP presentan mejores resultados en las métricas que el Xgboost, sin embargo; superior en el caso de la regresión logística sin ser un método de aprendizaje profundo como lo son las redes neuronales.

**TABLA 6.**

Resumen de métricas finales en modelos implementados. Fuente: Elaboración propia.

| MODELO | F1 SCORE | AUC |
|---|---|---|
| Regresión Logística | 0,94 | 0,98 |
| Red Neuronal MLP | 0,93 | 0,98 |
| Xgboost | 0,74 | 0,6 |

La Figura 13, muestra en el mismo gráfico las curvas de los tres modelos implementados. En un modelo con buen desempeño, su curva debe aproximarse a la esquina superior izquierda, lo cual sucede con las curvas de los modelos regresión logística y Red Neuronal MLP, las cuales están muy cercanas entre sí pero más alejadas de dicha esquina del modelo de la red. La curva del modelo Xgboost es muy cercana a la diagonal, lo que indica que no diferencia con muy buen desempeño entre las clases.

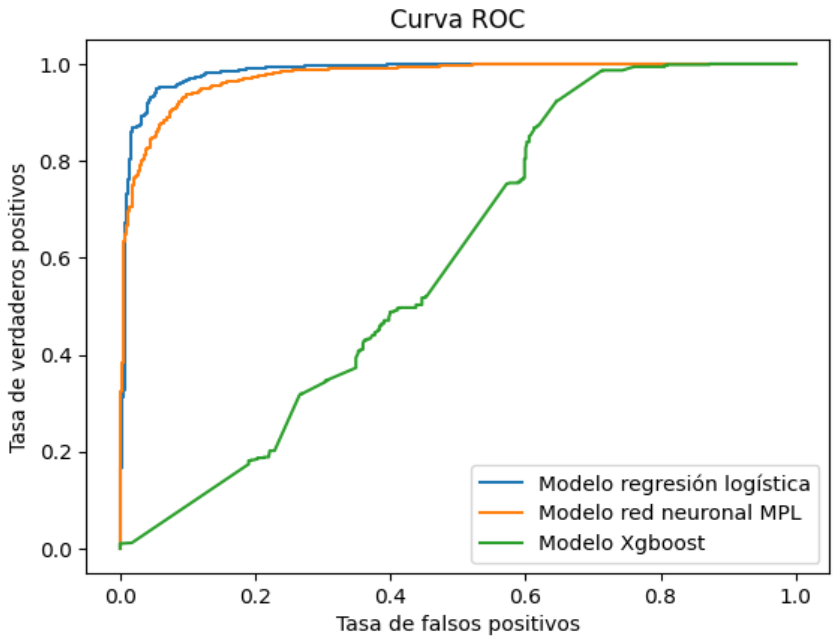


Figura 13. Gráfico de Curvas ROC de los modelos implementados. Fuente: Elaboración propia.

Finalmente dadas los resultados de las métricas F1 y AUC, y las curvas ROC, el modelo con mejor desempeño es el de Regresión Logística, muy cercano en resultado se encuentra la Red Neuronal MLP y finalmente con unas métricas muy inferiores a los demás se encuentra el modelo Xgboost.

# VII. CONCLUSIONES

- El análisis de correlación permitió identificar que las variables close, open, high, low, aportan la misma información con respecto a la variable de respuesta. Con base en este resultado se decidió continuar el estructuramiento del modelo utilizando la variable close, lo que se traduce en menor procesamiento de datos. En términos prácticos en los mercados financieros, las inversiones en lapsos de tiempo muy cortos, tomar en cuenta todas las variables si representan información relevante para una predicción acertada.
- Es de anotar que la búsqueda del rezago juega un papel importante en el resultado de los modelos, esto se evidencia en un mejor rendimiento del modelo de regresión logística, una vez se pasa de un valor arbitrario a un valor óptimo de rezagos.
- El modelo Xgboost es muy eficiente a nivel de procesamiento y convergencia, sin embargo, fue el modelo que obtuvo la menor calificación.
- Las redes neuronales implican gran capacidad de cómputo debido a que son modelos paramétricos, y puede llegar a sobrentrenarse y caer en mínimos locales, lo cual se traduce en una gran inversión de tiempo y de recursos computacionales para llegar a un resultado óptimo. Lo anterior genera una limitación en la toma de decisión del inversionista que debe generarse de manera diaria y oportuna.
- El modelo que se debe utilizar para fines de la predicción en el corto plazo es regresión logística ya que obtuvo el mejor desempeño de manera muy eficiente y rápida, de esta manera se desarrolla un modelo analítico para la predicción en corto plazo del sentido del S&P 500 utilizando técnicas de aprendizaje automático.

**Trabajo Futuro**

- Se evidencia una oportunidad en explorar variables como la de volumen, que aporta una condición de mercado de compra y venta que puede generar robustez en modelos futuros.
- La aplicación de enfoques de modelos diferentes de ML puede mejorar la interpretabilidad de este tipo de problemas, dando más herramientas de análisis que contribuyan en la construcción de soluciones optimas y consistentes en el tiempo.

# VIII. REFERENCIAS

Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning: Vol. 4 No. 4, p. 738 (New York: springer).

Cermeño, R., & María José Roa. (2014). Desarrollo financiero, crecimiento y volatilidad: una breve revisión de la literatura reciente. Estudios Económicos, 1(57), 85–105.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. https://doi.org/10.1145/2939672.2939785

Corredor, C., & Montoya, C. (2022). Aplicación de modelos de deep learning y machine learning para el pronóstico de la serie de tiempo del par de divisas euro - dólar. Universidad de Antioquia.

Fama, E. F. (1991). Efficient Capital Markets: II. The Journal of Finance, 46(5), 1575–1617. https://doi.org/10.1111/j.1540-6261.1991.tb04636.x

Francés Monedero, T. (2020). Impacto del machine learning en el sistema financiero. Universidad Pontificia Comillas.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

Kelleher, J. D., Tierney, B., & Tierney, B. (2018). Data science: an introduction.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

Mathwork. (2022). Evaluación del rendimiento de los modelos de clasificación de machine learning.

Vesga Bermejo, C. A., & Chacón Ojeda, Y. L. (2019). Opciones binarias en portafolios óptimos de inversión a corto plazo estructurados en Forex. Universidad Autónoma de Bucaramanga UNAB.

Zhou, Z.-H. (2021). Machine learning. Springer Nature.