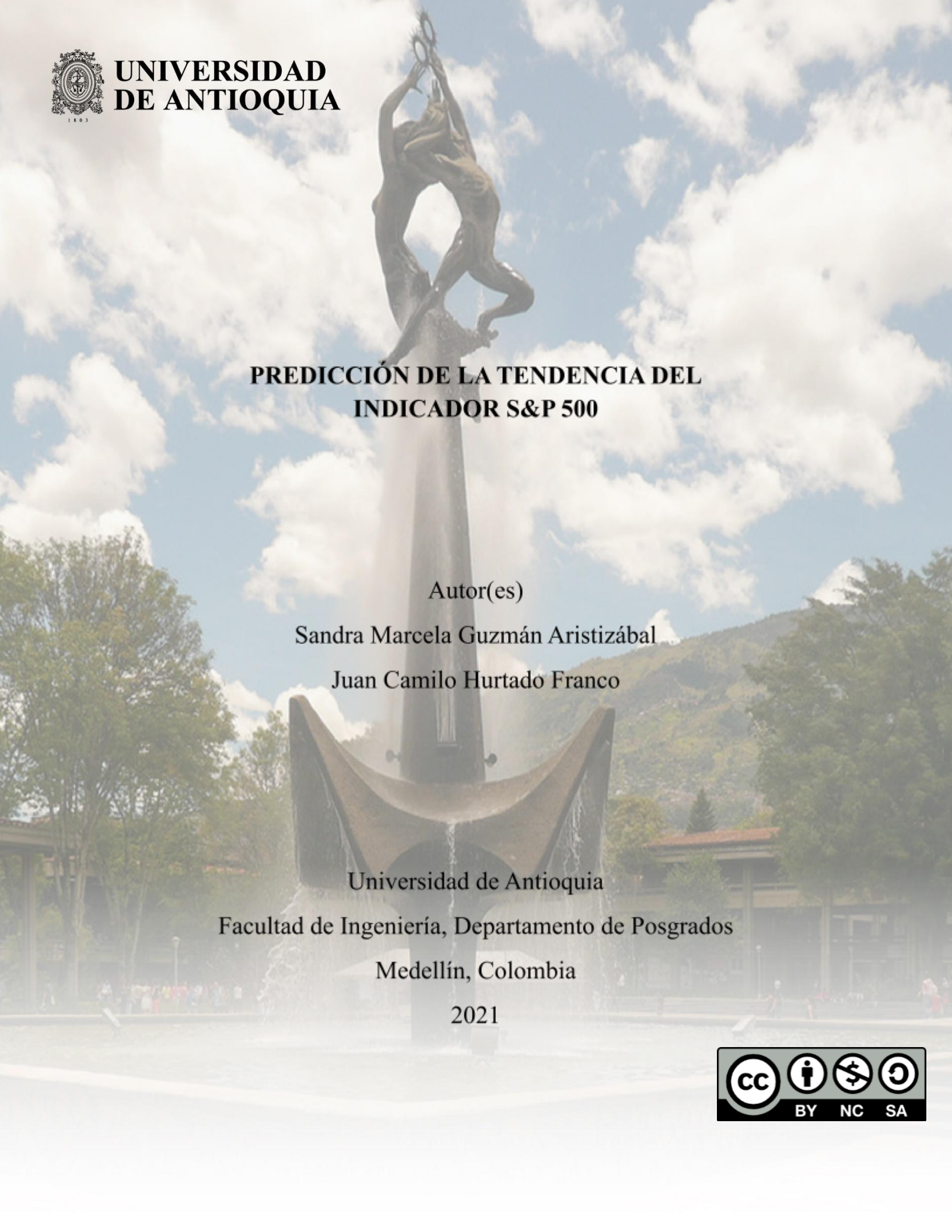


UNIVERSIDAD DE ANTIOQUIA

PREDICCIÓN DE LA TENDENCIA DEL INDICADOR S&P 500

Autor(es)

Sandra Marcela Guzmán Aristizábal

Juan Camilo Hurtado Franco

Universidad de Antioquia

Facultad de Ingeniería, Departamento de Posgrados

Medellín, Colombia

2021

Predicción de la tendencia del indicador S&P 500

Sandra Marcela Guzmán Aristizábal

Juan Camilo Hurtado Franco

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:

Especialista en Analítica y Ciencia de Datos

Asesores (a):

Javier Fernando Botía Valderrama

Doctor en Ingeniería Electrónica

Universidad de Antioquia

Facultad de Ingeniería, Departamento de Posgrados

Medellín, Colombia

2021

Predicción de la tendencia del indicador S&P 500.

Sandra Marcela Guzmán Aristizábal[1]
marcela.guzman1@udea.edu.co

Juan Camilo Hurtado Franco[1]
juan.hurtado3@udea.edu.co

Resumen

El mercado de valores es conocido y caracterizado por su gran complejidad y volatilidad. Las *Redes Neuronales de Largo y Corto Plazo* (LSTM) son ampliamente utilizadas en diversas aplicaciones por su capacidad de abordar problemas de alta dimensionalidad y no-linealidad, resultando particularmente atractivas para predecir *series temporales financieras*. En esta monografía, se explora un método para realizar la predicción de la tendencia de cierre del indicador *S&P 500* con un horizonte de pronóstico de 1 día, adecuando el problema de interés a un problema de clasificación binaria; 1 si la predicción del indicador es creciente y 0 si es decreciente. Finalmente, se evalúan algunas pruebas de hipótesis para concluir sobre la estacionalidad de la serie temporal y se analizan los resultados más relevantes de las implementaciones, entre los cuales sobresalen niveles de exactitud de 52.51% y 64.04% para los modelos LSTM y Regresión Logística respectivamente.

Palabras clave: *Redes Neuronales de Largo y Corto Plazo* (*Long Short Term Memory - LSTM*), *Series temporales financieras*, *S&P 500*.

Abstract

The stock market is known for its extreme complexity and volatility. The Long Short-Term Memory Neural Networks (LSTM) are widely used in various applications due to their ability to address high dimensionality and non-linear problems, being particularly attractive for predicting financial time series. In this paper, a method is explored to perform the prediction of the closing trend of the S&P 500 indicator with a forecast horizon of 1 day, adapting the problem of interest to a binary classification problem; 1 if the prediction is to increase and 0 if the opposite. Some hypothesis tests are evaluated to conclude on the stationarity of the time series and the most relevant results of the implementations are analyzed, among which Accuracy levels of 52.51% and 64.04% stand out for the LSTM and Logistic Regression models respectively.

Key words: *Long Short Term Memory - LSTM*, *Financial time series*, *S&P 500*.

[1] Especialización en Analítica y Ciencia de Datos. Facultad de Ingeniería. Universidad de Antioquia
Dirección: Calle 67 No. 53 - 108

1. Introducción

La predicción de los mercados financieros, ha sido un campo ampliamente analizado y estudiado con un gran interés tanto académico e investigativo como comercial. Hace varios años y de manera evidente, se ha observado que la tendencia de un gran número de disciplinas y aplicaciones busca eliminar la ineficiencia, los sesgos y la imprecisión de confiar únicamente en la experiencia personal y la intuición para el análisis y el juicio.

Realizar tareas de regresión o clasificación de series financieras es un ejercicio complejo en sí mismo como consecuencia de la naturaleza caótica del mercado de valores; los movimientos son dinámicos, aleatorios, irracionales, ruidosos y no lineales. De igual manera, las variables y las salidas están profundamente relacionadas con los acontecimientos macroeconómicos e incluso políticos [1].

Existen esencialmente tres enfoques de pensamiento relacionados con este campo. El **análisis fundamental** evalúa el valor de las empresas mediante un marco general de factores cualitativos y cuantitativos, comparando este resultado con el valor del mercado para identificar oportunidades de inversión. De esta manera, se estudia todo lo que puede afectar al valor de la acción, desde factores macroeconómicos como el estado de la economía y las condiciones del sector hasta factores microeconómicos como la eficiencia de la gestión en las empresas.

El **análisis técnico**, no tiene en cuenta la información de las empresas y, en su lugar, evalúa las inversiones identificando las oportunidades de negociación mediante el análisis de tendencias y patrones estadísticos de la actividad comercial, como el movimiento de los precios y el volumen.

El tercer enfoque, y de alguna manera el marco en el que reside esta monografía, es la predicción de valores bursátiles a través del **aprendizaje automático** y el **trading algorítmico**, donde las redes neuronales son el modelo más utilizado debido a su relativa eficiencia para adaptarse a la incertidumbre y al ruido de las variables [2].

Es claro que cada vez es más necesario un método inteligente, numérico y eficaz para dirigir las operaciones bursátiles. Con el rápido y vertiginoso desarrollo de la **Inteligencia Artificial**, la aplicación del **Deep Learning**, la naturalización de lenguajes de programación gratuitos y la amplia cultura colaborativa en estos campos, problemas y aplicaciones que hace un par de años resultaban increíblemente difíciles de abordar, ahora no requieren de un conocimiento tan especializado por lo menos para su exploración y experimentación.

Esta monografía tiene entonces como objetivo explorar la posibilidad de predecir la tendencia de los movimientos de cierre del indicador bursátil S&P 500 y con un horizonte de predicción de 1 día.

El alcance propuesto en este documento se estructura de la siguiente manera. En la sección 2, se establece el marco teórico sobre los conceptos más relevantes de las series financieras así como las generalidades de los modelos utilizados. En la sección 3, se presenta la metodología para la adquisición de la información, la generación de indicadores técnicos y el proceso de entrenamiento, prueba y validación. En la sección 4, se analizan los resultados obtenidos en la experimentación para finalmente presentar la discusión y concluir sobre los mismos en la sección 5.

2. Marco teórico.

Existen diversos enfoques para abordar las tareas de predicción y clasificación en series de tiempo. Antes de la masificación de las técnicas de Deep Learning, el modelado y la predicción se habían concentrado principalmente en el campo de las regresiones estadísticas (ARIMA) o en cualquier modificación de las mismas.

Estudios previos en este campo han demostrado que si bien los modelos tradicionales para enfrentar las series temporales proporcionan un poder predictivo decente, existe un marcado límite que muchos métodos de Machine Learning han logrado sortear gracias a la capacidad de interpretar relaciones no lineales y los patrones emergentes de los diversos indicadores técnicos.

En esta sección, se da un contexto general del funcionamiento de los modelos utilizados y algunas definiciones relevantes del mercado de valores que permiten una mayor comprensión durante el análisis.

2.1. Efficient Market Hypothesis (EMH).

La Hipótesis del Mercado Eficiente (EMH) es una teoría de gran relevancia dentro de la economía financiera. Formalizada y publicada por Eugene Fama en 1970, afirma que los mercados de valores son eficientes en el sentido de que los precios de las acciones reflejan plenamente toda la información sobre las mismas, y que las acciones siempre se negocian a su valor razonable, lo que hace imposible la infravaloración y la sobrevaloración de las empresas públicas. Por lo tanto, debería ser imposible superar al mercado, lo que significa que la única forma de alcanzar rendimientos superiores es aumentar el riesgo de la inversión.

En su artículo de los años 70, Fama introdujo tres variantes diferentes de su hipótesis que representan distintos grados de eficiencia en el mercado: *forma débil*, *semifuerte* y *fuerte*.

La variante de eficiencia de forma débil, afirma que es imposible predecir los precios futuros analizando los movimientos pasados, es decir, no se pueden obtener rendimientos y retornos sostenidos a largo plazo utilizando datos históricos, sin embargo gracias al gran avance en el poder computacional y el desarrollo de herramientas estadísticas robustas, algunas técnicas de análisis fundamental pueden producir rendimientos muy interesantes en este tipo de mercados.

Según esta teoría, los precios de las acciones no pueden predecirse analizando sus movimientos anteriores, esto significa esencialmente que los precios del mercado de valores no pueden predecirse debido a la naturaleza aleatoria de sus movimientos.

La teoría del paseo aleatorio (**Random Walk Hypothesis**) está muy vinculada a la hipótesis del mercado eficiente y es de gran importancia mencionarla en cualquier intento o estudio para explorar la predicción de los movimientos del mercado de valores. En este artículo, realizaremos algunas pruebas que abordan un poco esta teoría y que permiten interpretar de manera más clara los resultados obtenidos.

La variante de eficiencia semifuerte, implica que la introducción de nueva información pública sobre una acción dará lugar a un ajuste muy rápido y sin sesgo del precio de la misma. Esta variante sugiere entonces que no es posible alcanzar retornos en el largo plazo mediante el análisis técnico o fundamental, ya que el comportamiento de este tipo de mercado hace que las técnicas de los análisis anteriores no pudieran aprovechar la naturaleza aleatoria y sesgada del mismo.

La variante de eficiencia fuerte del mercado, implica que el mercado es realmente eficaz y que los precios de las acciones reflejan toda la información sobre el valor, tanto privada como pública. Este grado de eficiencia sugiere que es imposible conseguir retornos consistentes durante un largo periodo de tiempo, ya que la única forma de obtener grandes retornos de los portafolios es asumiendo un riesgo alto.

La **Hipótesis del Mercado Eficiente** (EMH) y su principal implicación de que el mercado es imbatible, es ampliamente debatida y estudiada en la actualidad. Existen gran cantidad de artículos dedicados a este tema y con diferente rigor estadístico y matemático.

En este artículo, nos apoyaremos en algunos aspectos de esta teoría para comprender con un mayor grado de profundidad el comportamiento del indicador S&P 500 y su implicación en los resultados obtenidos.

2.2. Indicador S&P 500.

Standard & Poor's es una de las mayores agencias de calificación crediticia, que asigna calificaciones a empresas y países considerando la deuda que emiten en una escala de AAA a D, indicando su grado de riesgo de inversión.

El popular índice S&P 500 es quizás el producto más conocido de Standard & Poor's. Este índice pondera la capitalización bursátil de las 500 mayores empresas que cotizan en la bolsa de los Estados Unidos. El índice está considerado como el mejor indicador de la renta variable estadounidense y es ampliamente utilizado y referenciado.

Una de las limitaciones del S&P y de otros índices ponderados, surge cuando existe sobrevaloración de los índices, es decir, cuando suben más de lo que sus **fundamentos** justifican. Si una acción tiene una gran ponderación en el índice y está sobrevalorada, suele inflar el valor global o el precio del índice. [4]

2.3. Regresión Logística.

Es un algoritmo de predicción que utiliza variables independientes para predecir una variable de interés. Las variables independientes pueden ser numéricas o categóricas, pero este algoritmo tiene la particularidad de que la variable dependiente debe ser categórica. La regresión logística es un modelo estadístico que utiliza la función logística para modelar una probabilidad condicional, es decir, calculamos la probabilidad condicional de la variable dependiente Y, dada la variable independiente X.

2.4. Redes Neuronales Artificiales.

Las redes neuronales artificiales son modelos que representan el cerebro, donde un gran número de neuronas envían señales entre sí. Se denominan neuronales por su origen en un modelo simplificado de la neurona humana, sin embargo, el uso moderno de las redes neuronales ya no se basa en estas inspiraciones biológicas. En su lugar, una ANN (**Artificial Neural Network**) es una red de pequeñas unidades de computación, cada una de las cuales toma un vector de valores de entrada y produce un único valor de salida mediante un cálculo.

En esencia, las redes neuronales artificiales pueden ilustrarse de forma simplificada como grafos con conexiones ponderadas. El tipo más sencillo de redes neuronales artificiales es la red multicapa en la que las unidades se conectan sin ningún ciclo y las salidas de cada capa de unidades de cálculo se pasan a la siguiente capa sin pasar nada de vuelta.

Las unidades de cálculo propiamente dichas consisten en funciones matemáticas, con entradas multiplicadas por pesos adaptativos. El nodo se activa si la suma de las entradas ponderadas satisface las funciones matemáticas definidas de acuerdo a diversas funciones.

2.5. LSTM – Long Short Term Memory.

El término **memoria a largo y corto plazo** (LSTM) en este contexto, se refiere a una arquitectura desarrollada para superar los problemas de desvanecimiento del gradiente inherentes al entrenamiento de las RNN (**Recurrent Neural Network**) tradicionales. En resumen, estos problemas surgen por las dependencias a largo plazo, cuando una célula tiene que recordar algo durante un largo periodo de tiempo en las secuencias de entrenamiento.

Cuando se entrena una red neuronal artificial utilizando métodos basados en el gradiente y el *backpropagation*, en muchos casos el gradiente se volverá progresivamente más pequeño, impidiendo que un peso cambie su valor. Esto se convierte en un problema, ya que los cálculos del proceso utilizan números de precisión finita.

Las LSTM resuelven parcialmente este problema al permitir que los gradientes fluyan sin cambios a través de compuertas que mantienen los valores que se van considerando importantes para el modelo.

2.6. Augmented Dickey-Fuller Test (ADF).

La comprobación de la estacionalidad de una serie temporal es una actividad que se realiza con frecuencia en los modelos autorregresivos. Si bien se discutirá con más detalle los conceptos de estacionalidad y diferenciación en la sección 3, es relevante mencionar que la prueba ADF es fundamentalmente una prueba de significación estadística, es decir, se realiza una prueba de hipótesis que da como resultado un estadístico de prueba y un *p-value* que permite arrojar una conclusión [6]. De manera muy general, podríamos resumir el test de la siguiente manera:

Hipótesis Nula: H_0: $\gamma(gamma) = 0 \rightarrow$ Serie no estacionaria.

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \Phi \Delta Y_{t-p} + e_t \quad (1)$$

Donde:

Δy_t: Es el delta término a término de la información que compone la serie de tiempo.

α: Es una constante de los modelos estocásticos. (En su forma más general)

βt: Es el factor de tendencia en los modelos autorregresivos.

γ: Constante de la prueba de hipótesis para determinar si la serie temporal sigue un patrón de **Random Walk** (camino aleatorio) o **estacionario**.

$\Phi \Delta Y_{t-p}$: $\phi_1 \Delta y_{t-1} + \phi_2 \Delta y_{t-2} + \ldots \phi_p \Delta y_{t-p}$ y ϕ representa los parámetros del modelo.

e_t: Es el factor de tendencia en los modelos autorregresivos.

2.7. Wald-Wolfowitz Test (WW).

También conocido como prueba de corridas, es un procedimiento estadístico que examina si una cadena de datos se produce al azar a partir de una distribución específica. En series temporales, esta prueba es importante para determinar si los comportamientos bursátiles del conjunto de datos bajo estudio se generan de manera aleatoria o si por el contrario se ven afectados por una variable subyacente. La prueba Kolmogorov-Smirnov ha sido referenciado por muchos estadísticos como una mejor herramienta para detectar diferencias entre distribuciones. Sin embargo, el principal interés de esta prueba es demostrar si los datos de la muestra que se está probando representan dependencia entre sí. Se podría resumir de la siguiente manera:

H_0:Cada elemento de la secuencia es independientemente extraído de la misma distribución aleatoria.

3. Metodología.

De manera general, la obtención de los resultados que se discutirán en las siguientes secciones, siguió el siguiente proceso:

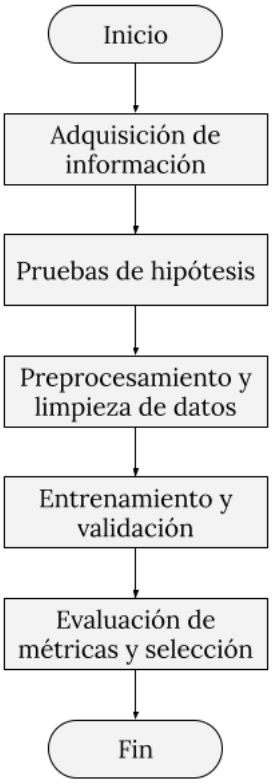


Figura 1. Diagrama de flujo metodología.

3.1. Adquisición de información.

La información primaria o 'cruda' se obtuvo a través de *Yahoo Finance* [8] utilizando la librería *pandas-datareader* en Python. Esta librería permite al usuario descargar los valores históricos de diversos indicadores bursátiles y crear un objeto tipo *DataFrame* para su visualización y manipulación. Los datos recogidos abarcan desde el 16 de mayo de 2011 hasta el 14 de mayo de 2021.

Los datos importados representan un día de trading y contienen atributos como el precio de apertura, cierre, máximo diario y mínimo diario del S&P 500 además de 50 indicadores relevantes de la bolsa, entre índices, divisas, futuros y acciones, incluyendo el valor del indicador en cuestión del día anterior.

3.2. Pruebas de hipótesis.

Como se mencionó en el marco teórico, una de las principales actividades que se deben realizar cuando se analizan series de tiempo es la verificación de la estacionalidad o aleatoriedad de la misma. Antes de establecer la metodología para las pruebas de hipótesis, detallar el concepto de estacionalidad es de gran importancia.
La estacionalidad de las series temporales se refiere a las series cuyas propiedades estadísticas no cambian con el tiempo. Esto no debe confundirse con un sentido más general de "cambio", es decir, lo que no cambia con el tiempo es la forma en que varía la propia serie. En otras palabras menos confusas: la media, la varianza y la estructura de autocorrelación de una serie temporal permanecen constantes.

Esto es primordial en el ámbito de la estadística de series temporales, ya que muchas pruebas y procesos estadísticos sólo funcionan con datos estacionarios. De acuerdo a esto, cualquier serie que exhiba un comportamiento no estacionario debe transformarse para serlo.

Estas pruebas se encuentran implementadas en la librería *statsmodels* [9] a través de las funciones `adfuller` y `runstest_1samp`.

3.3. Preprocesamiento y limpieza de datos.

Dentro del alcance de esta monografía está la evaluación de la incidencia de diversos indicadores de la bolsa y su relevancia en el poder de predicción de la tendencia del índice S&P 500. Sin embargo, la misma naturaleza del mercado hace que estas variables ganen o pierdan relevancia con el tiempo. Es por esto que uno de los pasos consiste en la limpieza de valores nulos o inexistentes que surgen al importar datos históricos de indicadores con diferente antigüedad.

Con el fin de lograr un proceso de experimentación un poco más robusto y entender el impacto que tienen las características seleccionadas para el ajuste de los modelos, se realizará la experimentación con 5 datasets:

A. Dataset X0: X1 + X2.
B. Dataset X1: Indicadores de la bolsa + S&P 500 del día anterior.
C. Dataset X2: Atributos del S&P 500 (High, Low, Open, Adj Close, Volume).
D. Dataset X3: X2 + X5.
E. Dataset X4: X1 + X2 + X5.
F. Dataset X5: Precio de cierre del S&P 500 (Adj Close) + Indicadores técnicos de la librería *Technical Analysis ta* [10].

3.4. Entrenamiento y validación.

Dado que el método de *Cross Validation K-Fold* no es útil en las series de tiempo, utilizamos en su lugar la validación cruzada "**walk-forward**" y de esta manera realizamos el ajuste de hiperparámetros. Además, para proporcionar una evaluación imparcial de nuestro modelo final, retenemos el último 20% de nuestros datos como un conjunto de prueba independiente. Este conjunto de datos sólo se utiliza una vez que nuestro modelo está completamente entrenado (utilizando los conjuntos de entrenamiento y validación).

3.5. Evaluación de métricas y selección.

El objetivo final de nuestros modelos es predecir la dirección del indicador S&P 500 como una clasificación binaria. De manera consecuente, el rendimiento de los modelos se evaluará a través del *Accuracy*, *Recall*, y *F-score*.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F_1 = \frac{2PR}{P + R} \quad (4)$$

Donde:

TP: True Positives
TN: True Negatives
FP: False Positives
FN: False Negatives
P: Precision
R: Recall

4. Resultados y análisis.

4.1. Pruebas de hipótesis.

Para la ejecución de las pruebas de hipótesis se analizaron los valores de cierre del indicador S&P 500 de acuerdo a los parámetros requeridos por cada prueba. El nivel de significancia seleccionado fue del 5% (α = 0.05) para concluir sobre cada prueba.

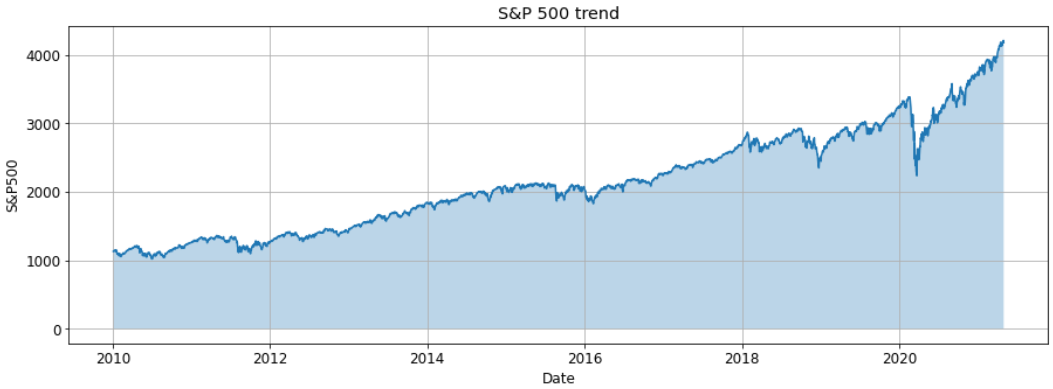


Figura 2. Comportamiento precio de cierre indicador S&P 500.

Test	Variable	p-value	Conclusión
ADF-1	S&P 500 (Adj Close)	0.9963	Se acepta H_0. Serie no estacionaria.
ADF-2	S&P 500 (return)	$2.47x10^{-21}$	Se rechaza H_0. Serie estacionaria.
WW-1	S&P 500 (direction)	0.0218	Se rechaza H_0. No es una serie aleatoria.

Tabla 1. *Resultados pruebas de hipótesis.*

Como se puede observar en la tabla anterior, el *p-value* de la prueba ADF-1 es mayor al nivel de significancia α = 0.05. Por esta razón se acepta la hipótesis nula y se puede concluir que los precios de cierre (*Adj Close*) **tienen un comportamiento no estacionario**.

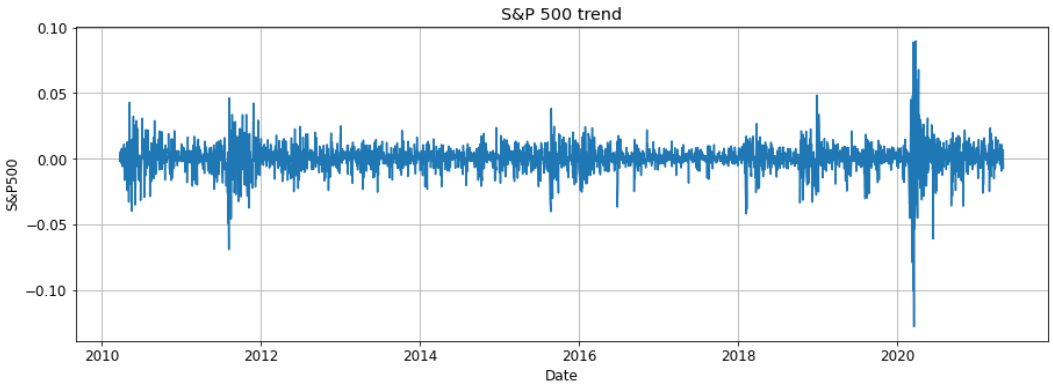


Figura 3. *Comportamiento de los retornos del precio de cierre indicador S&P 500.*

Es necesario transformar los datos para que el entrenamiento de los modelos tenga el efecto esperado y se logren establecer las relaciones existentes entre las características del modelo y la dirección de la tendencia sin perder desempeño por los ruidos generados por la no estacionalidad. Esto se logra a través de la siguiente transformación:

$$Logarithmic\ return = ln\left(\frac{P_d}{P_{d-1}}\right) \quad (5)$$

Donde P_d es el precio de cierre del indicador en el día *d*.

Como se puede observar en la Figura 3 y de acuerdo al *p-value* la prueba ADF-2, **la nueva transformación tiene un comportamiento estacionario**.

Para interpretar correctamente la prueba WW-1 (Wald-Wolfowitz) es necesario hacer una breve explicación acerca de la *teoría del paseo aleatorio* o **Random Walk Hypothesis**, como es conocida. De manera muy resumida, se podría decir que esta teoría afirma que los cambios en los precios de las acciones tienen la misma distribución y son independientes entre sí. La teoría del paseo aleatorio sugiere que las acciones siguen una trayectoria aleatoria e impredecible lo que desestima cualquier método para predecir los precios de las acciones de manera sistemática. Por lo anterior, cualquier precio de las acciones, tendencia o información anterior no puede utilizarse para predecir el movimiento futuro.

Para realizar esta prueba, es necesario considerar la tendencia del indicador y convertirla en una secuencia binaria, posteriormente esta característica se convierte en una entrada importante para el entrenamiento de los modelos. Con un *p-value* de 0.0218, **se rechaza la hipótesis nula permitiendo concluir que el comportamiento de la tendencia no es independiente entre sí.**

4.2. Experimentación.

4.2.1. Algoritmos de la librería scikit-learn.

Se realizaron 694 experimentos con diferentes algoritmos de la librería para resolver el problema de predicción de la clase (1: si el precio del S&P subirá, 2: caso contrario):

- lor: *Logistic Regression*.
- dtc: *Decision Tree Classifier*.
- rfc: *Random Forest Classifier*.
- gnb: *Gaussian Naive Bayes*.
- svc: *Support Vector Machine*.
- gbc: *Gradient Boosting Classifier*.
- mlp: *Multi-layer Perceptron Classifier*.

Con el fin de realizar la validación sobre los datos, la cantidad de registros -días- en un experimento fue distribuida en varias particiones de acuerdo al parámetro *Period* (cantidad de días para cada partición) y a *test_size* (porcentaje que indica el tamaño de los datos con los cuales se realiza la validación del modelo), reciclando la información de validación (test data) de la partición P_i en la data de entrenamiento de la partición P_{i+1}. Se obtuvieron un total de *6074 particiones* entre los diferentes experimentos con valores de exactitud -calculados para los datos de validación- entre 20.83% y 79.17%.

Para cada experimento, se presentó el promedio de la exactitud obtenido en sus particiones respecto a la utilización de medias móviles (0: no se utilizan, 1: se utilizan sólo las medias móviles de los datos para el entrenamiento, 2: se utilizan tanto los datos como las medias móviles) y PCA (*True*: si fue aplicado en el experimento, *False*: caso contrario) y la asignación del número de medias (*n_mean*), número de PCA (*n_pca*) y conjunto de datos utilizado. Las ***Figuras* 4 y 5** presentan los principales resultados.

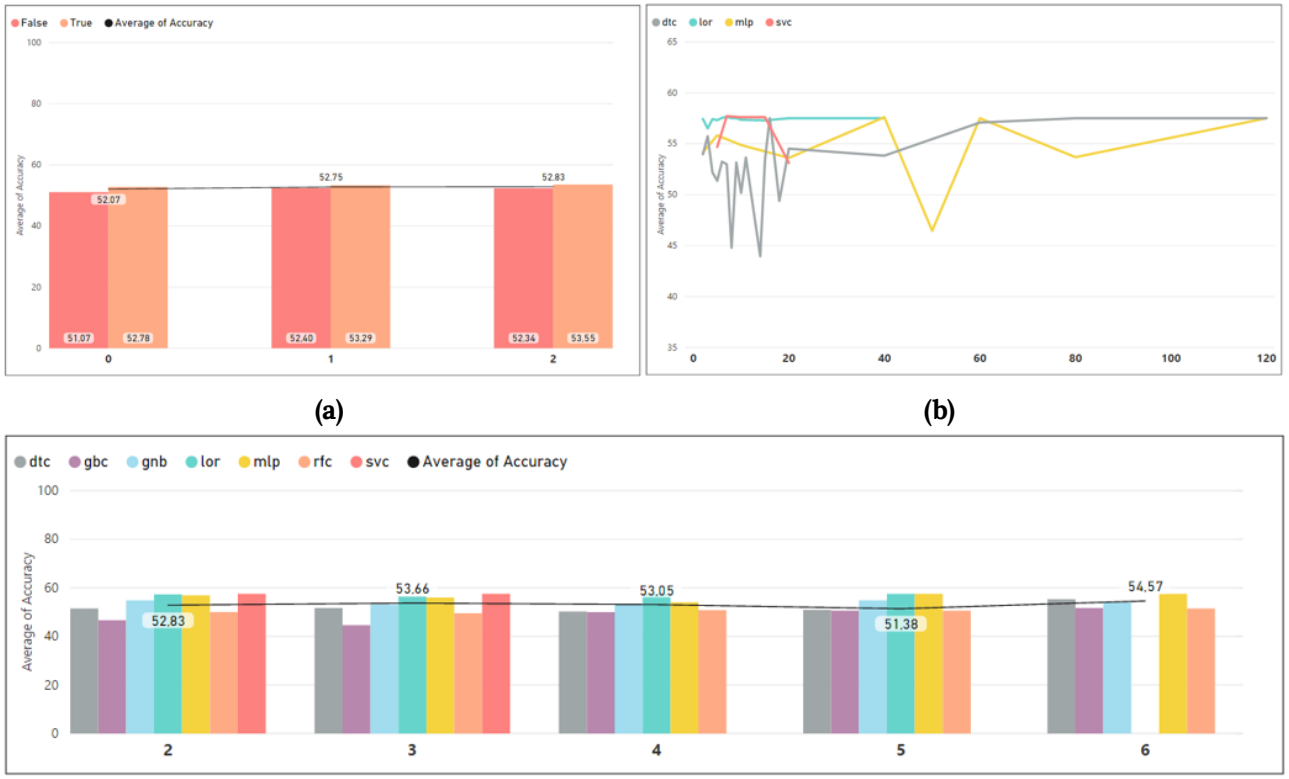


Figura 4. *Exactitud promedio por uso de medias para PCA True o False (a), número de medias móviles para diferentes algoritmos (b), número de PCA según algoritmo (c).*

El uso de medias móviles en la experimentación refleja un pequeño aumento de la exactitud (***figura* 4a**). La utilización de PCA (PCA True), especialmente con 3 y 6 componentes (*n_pca*), también mejora la métrica.

Los algoritmos *lor*, *mlp* y *svc*, exhiben mejor desempeño independientemente del número de componentes utilizado (***figura 4c***), dataset (***figura 5c***) o periodo (***figura 5d***); y la cantidad de días por partición (*Period* o periodo) connota una relación directa al incremento de la exactitud.

Ahora bien, es menester mencionar que en un gran número de las particiones de los experimentos, el modelo predijo, para todas las entradas de testeo, que el precio del S&P500 subiría (en experimentos con el algoritmo *mlp* -que está entre los mejores desempeños-, por ejemplo, fue muy frecuente). Ello se debe, en gran medida, a la tendencia alcista del indicador (***figura 2***).

Figura 5. *Exactitud promedio por número de medias usadas para los conjuntos de datos* X0, X1, X2 *(a), número de medias para PCA True o False (b), algoritmo para cada conjunto de datos (c), algoritmo para diferentes periodos usados (d).*

4.2.2. Algoritmos de redes neuronales secuenciales (librería TensorFlow-Keras).

Pretendiendo reducir la tendencia de predicción al alza del precio del indicador en los modelos, se optó por la estrategia de predecir el precio del mismo en lugar de su tendencia (alza o baja) y, a partir de la comparación entre precios determinar la etiqueta. Así, por ejemplo, si la predicción del precio resulta en un valor mayor al precio del día anterior, se etiqueta como 1 (sube), de lo contrario 0 (baja).

Cada modelo, sea LSTM, GRU o Simple RNN, consta de 3 capas secuenciales con 25, 15 y 5 neuronas, respectivamente; de las cuales, las dos primeras, tienen función de activación lineal (*linear*) y, la de la última, depende de la asignación al momento de la experimentación.

Para evaluar el desempeño del modelo respecto a la etiqueta obtenida se utilizó como métrica de desempeño *Accuracy*. No obstante, también se consideró el *error* MAE obtenido en la predicción del precio.

Se realizaron un total de 734 experimentos con modelos secuenciales, divididos en cinco grupos de experimentación diferentes:

Grupo A: Con 106 experimentos LSTM, en los cuales se prueban diferentes optimizadores, funciones de pérdida y de activación (para la última capa del modelo, solamente), así como los conjuntos de datos X0, X1, X2.

Grupo B: 106 experimentos GRU con algunos de los parámetros que obtuvieron mejor desempeño en A. Se utilizan los conjuntos de datos X0.

Grupo C: 25 experimentos SimpleRNN con los mismos parámetros usados en B y los conjuntos de datos X0.

Grupo D: 65 experimentos sobre X3, X4 y X5 con los parámetros de los mejores obtenidos en A y B, según la métrica Accuracy.

Grupo E: 415 experimentos sobre X0, X1, X2, X3, X4 y X5 para 240, 480 y 720 registros.

En todos los grupos de experimentos se varió el parámetro *look back* en múltiplos de 5, entre 5 y 40 y, para los Grupos A hasta D, se utiliza la totalidad de registros de los datos (2516).

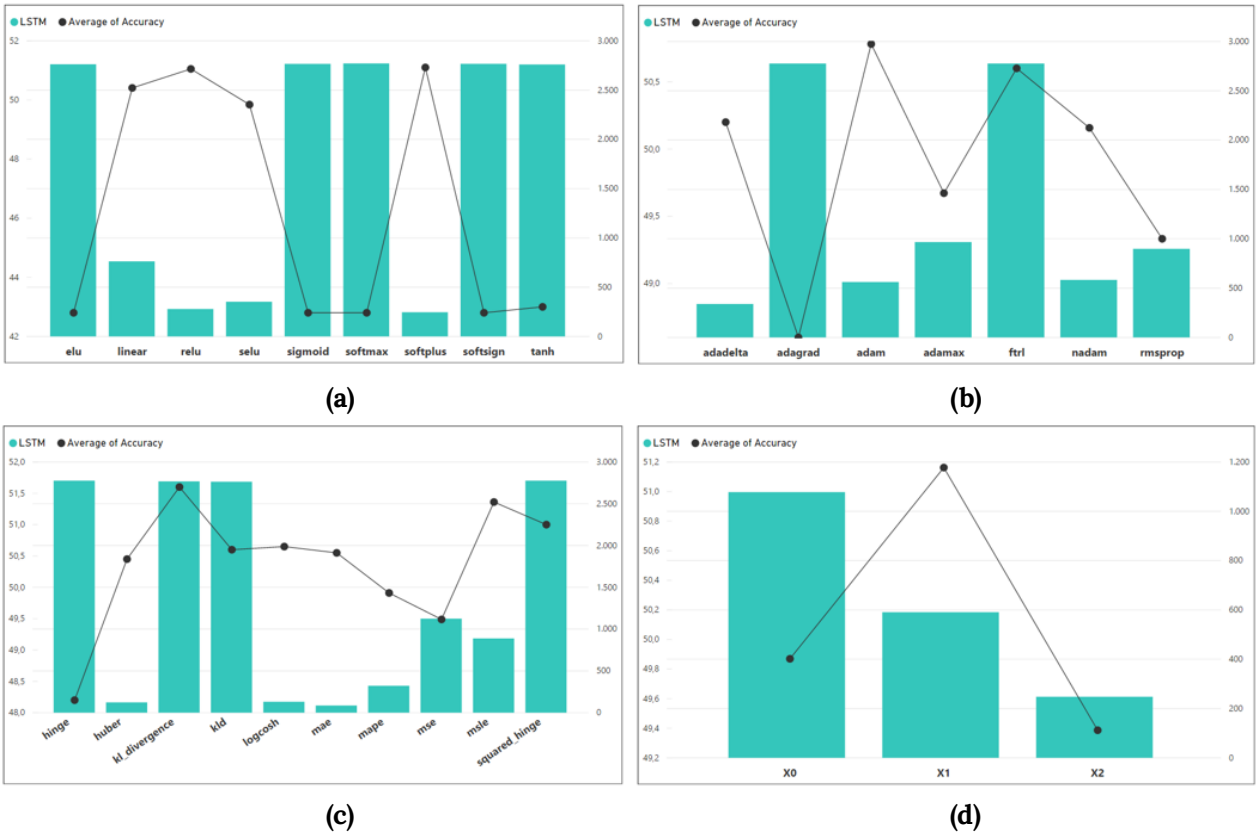


Figura 6. *Mae y exactitud promedio para experimentos del Grupo A, por función de activación (a), optimizador (b), función de pérdida (c) y conjunto de datos (d).*

En el Grupo A se evidenció que las funciones de activación con mejores resultados tanto en *Accuracy* como en MAE fueron *linear*, *relu*, *selu* y *softplus*. Entre los optimizadores resaltaron *Adam* y *ftrl* por sus altos valores de *Accuracy* y *adadelta*, *Adam* y *nadam*, por sus bajos MAE. Las funciones de pérdida *huber*, *logcosh* y *mae* tuvieron los más bajos valores de MAE con sobresalientes *Accuracy*. El conjunto de datos con mejor ejecución fue el X1.

X	Distribution	Accuracy	Val_Loss	Val_Mae	Look_Back	Layer_Type	Activation	Optimizer	Loss	Time
X0	Counter({1: 269, 0: 231})	55.80	5.61E+04	2.03E+02	5	LSTM	linear	adam	mse	36.98
X0	Counter({1: 257, 0: 218})	54.74	2.94E+06	1.65E+03	30	LSTM	linear	adam	mse	73.41
X1	Counter({1: 244, 0: 241})	54.43	4.62E-02	6.46E+02	20	LSTM	linear	adam	msle	38.46
X0	Counter({1: 269, 0: 231})	54.20	6.91E+00	1.93E+02	5	LSTM	linear	rmsprop	mape	44.55
X1	Counter({1: 253, 0: 247})	54.20	6.61E-02	7.84E+02	5	LSTM	linear	adam	msle	28.44

X	Distribution	Accuracy	Val_Loss	Val_Mae	Look_Back	Layer_Type	Activation	Optimizer	Loss	Time
X2	Counter({1: 281, 0: 219})	51.00	2.61E+01	2.68E+01	5	LSTM	linear	adam	logcosh	45.11
X2	Counter({1: 252, 0: 248})	51.20	1.05E+00	2.85E+01	5	LSTM	linear	adam	mape	44.82
X2	Counter({1: 277, 0: 223})	47.80	1.36E+00	3.72E+01	5	LSTM	selu	nadam	mape	45.47
X2	Counter({1: 279, 0: 221})	52.20	1.48E+00	4.06E+01	5	LSTM	selu	adam	mape	27.56
X2	Counter({1: 261, 0: 239})	51.40	1.53E+00	4.19E+01	5	LSTM	relu	adam	mape	44.79

Tabla 2. Experimentos del Grupo A con mejores resultados según *Accuracy* (superior) y Mae (inferior). El atributo Time de la tabla especifica el tiempo de ejecución del experimento en segundos y, Distribution, la cantidad de datos de validación por etiqueta (1: predijo que el precio sube, 0: caso contrario).

Para la experimentación con GRU (Grupo B) y SimpleRNN (Grupo C), seleccionamos los parámetros que aportaron mejor *Accuracy* en el Grupo A. La ***figura* 7** compara su comportamiento para cada algoritmo secuencial.

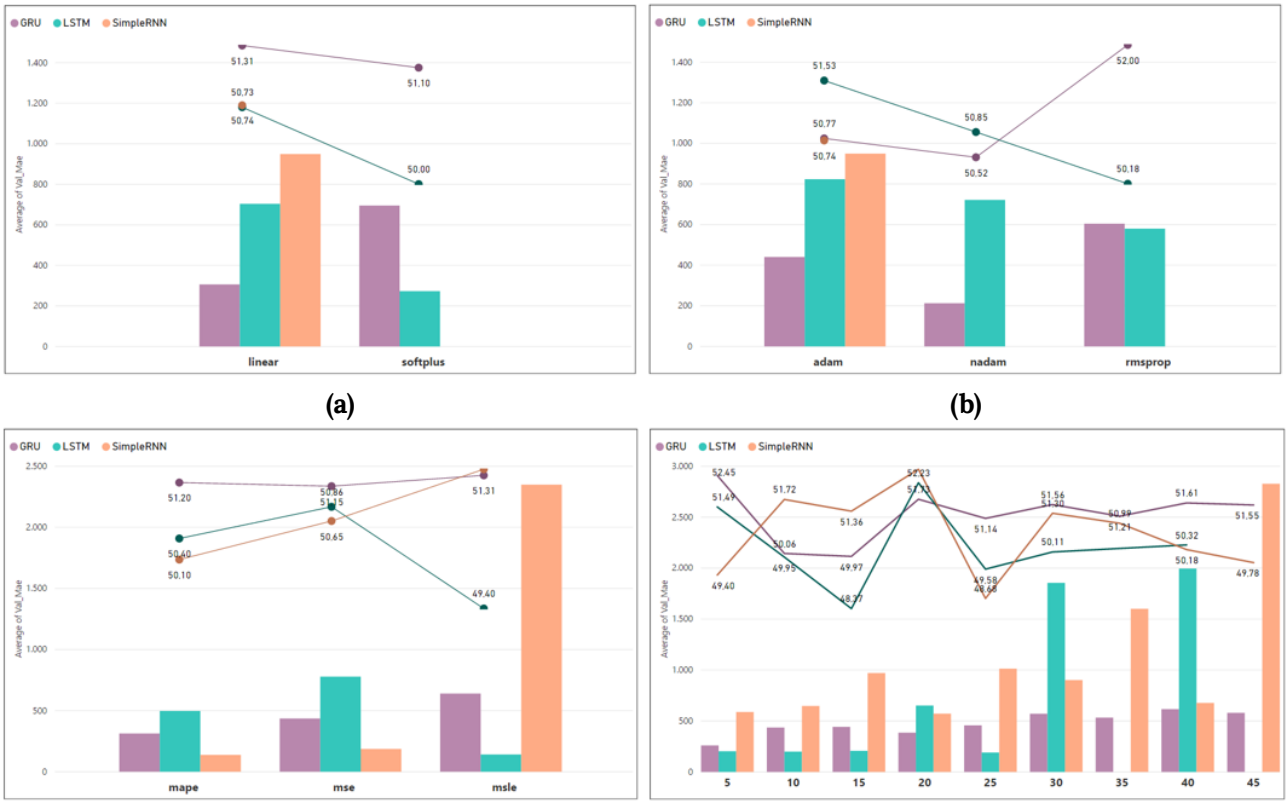


Figura 7. *Mae (columnas) y Accuracy promedio (líneas) para experimentos de los grupos A, B y C por función de activación (a), optimizador (b), función de pérdida (c) y look back (d).*

A nivel general, GRU mostró mayor estabilidad frente al cambio de los parámetros (verbigracia, aunque existe una clara relación entre el incremento del *look back* y el MAE, GRU fue menos susceptible a ella), así como los más altos niveles de *Accuracy*, mientras que LSTM los mejores MAE.

X	Distribution	Accuracy	Val_Loss	Val_Mae	Look_Back	Layer_Type	Activation	Optimizer	Loss	Time
X0	Counter({1: 283, 0: 202})	56.70	2.09E-03	9.84E+01	20	GRU	linear	rmsprop	msle	37.69
X0	Counter({1: 269, 0: 231})	55.80	5.61E+04	2.03E+02	5	LSTM	linear	adam	mse	36.98
X0	Counter({1: 274, 0: 201})	55.79	1.99E-02	3.52E+02	30	GRU	linear	nadam	msle	77.98
X0	Counter({1: 251, 0: 209})	55.65	1.87E+01	5.14E+02	45	GRU	linear	rmsprop	mape	107.66
X0	Counter({1: 274, 0: 196})	55.53	6.92E+00	1.94E+02	35	GRU	linear	rmsprop	mape	86.70

X	Distribution	Accuracy	Val_Loss	Val_Mae	Look_Back	Layer_Type	Activation	Optimizer	Loss	Time
X2	Counter({1: 252, 0: 248})	51.20	1.05E+00	2.85E+01	5	LSTM	linear	adam	mape	44.82
X2	Counter({1: 271, 0: 229})	50.60	1.68E+00	4.60E+01	5	LSTM	softplus	adam	mape	45.30
X2	Counter({1: 269, 0: 231})	49.40	1.92E+00	5.27E+01	5	LSTM	linear	rmsprop	mape	26.65
X2	Counter({1: 259, 0: 241})	46.60	2.22E+00	6.10E+01	5	LSTM	linear	nadam	mape	45.46
X2	Counter({1: 302, 0: 198})	49.20	5.60E+03	6.14E+01	5	LSTM	linear	adam	mse	45.07

***Tabla* 3.** Experimentos de los grupos A, B y C con mejores resultados según *Accuracy* (superior) y MAE (inferior).

En el Grupo D se evaluaron los conjuntos de datos X3, X4 y X5 con función de activación lineal, optimizadores *Adam* y *rmsprop* y funciones de pérdida *mse* y *msle* para observar si incluyendo nuevos features se podría mejorar la exactitud. Y, a pesar de que se alcanzó un experimento con *Accuracy* del 57.42%, en un nivel más amplio, los mejores experimentos no superan la métrica alcanzada con el conjunto de datos X0.

X	Distribution	Accuracy	Val_Loss	Val_Mae	Look_Back	Layer_Type	Activation	Optimizer	Loss	Time
X3	Counter({1: 240, 0: 225})	57.42	7.40E+03	7.29E+01	40	GRU	linear	adam	mse	146.75
X3	Counter({1: 258, 0: 232})	54.69	1.65E+04	9.18E+01	15	GRU	linear	adam	mse	45.96
X3	Counter({1: 246, 0: 244})	54.69	8.80E+04	2.26E+02	15	LSTM	linear	adam	mse	40.55
X4	Counter({1: 256, 0: 219})	54.53	5.09E+05	6.15E+02	30	LSTM	linear	adam	mse	70.27
X5	Counter({1: 263, 0: 227})	54.49	1.13E+04	7.19E+01	15	GRU	linear	adam	mse	45.00

X	Distribution	Accuracy	Val_Loss	Val_Mae	Look_Back	Layer_Type	Activation	Optimizer	Loss	Time
X3	Counter({1: 256, 0: 219})	50.32	4.53E+03	4.87E+01	30	GRU	linear	adam	mse	70.48
X3	Counter({1: 252, 0: 248})	50.80	4.64E+03	4.87E+01	5	GRU	linear	adam	mse	45.19
X3	Counter({1: 269, 0: 226})	49.09	5.59E+03	5.75E+01	10	GRU	linear	adam	mse	45.21
X5	Counter({1: 263, 0: 227})	54.49	1.13E+04	7.19E+01	15	GRU	linear	adam	mse	45.00
X3	Counter({1: 240, 0: 225})	57.42	7.40E+03	7.29E+01	40	GRU	linear	adam	mse	146.75

***Tabla* 4.** Experimentos del Grupo D con mejores resultados según Accuracy (superior) y Mae (inferior).

Con el propósito de determinar la incidencia del tamaño de los datos en el modelo, se llevaron a cabo experimentos con 240, 480 y 720 registros (recordar que los modelos se entrenan con el 80% de éstos registros y se valida con el 20% restante) que ilustraron una relación inversa entre la cantidad de registros y el *Accuracy* -ver ***Figura* 8** del Grupo E-; contraria a la relación entre esa cantidad y el MAE, que fue directa.

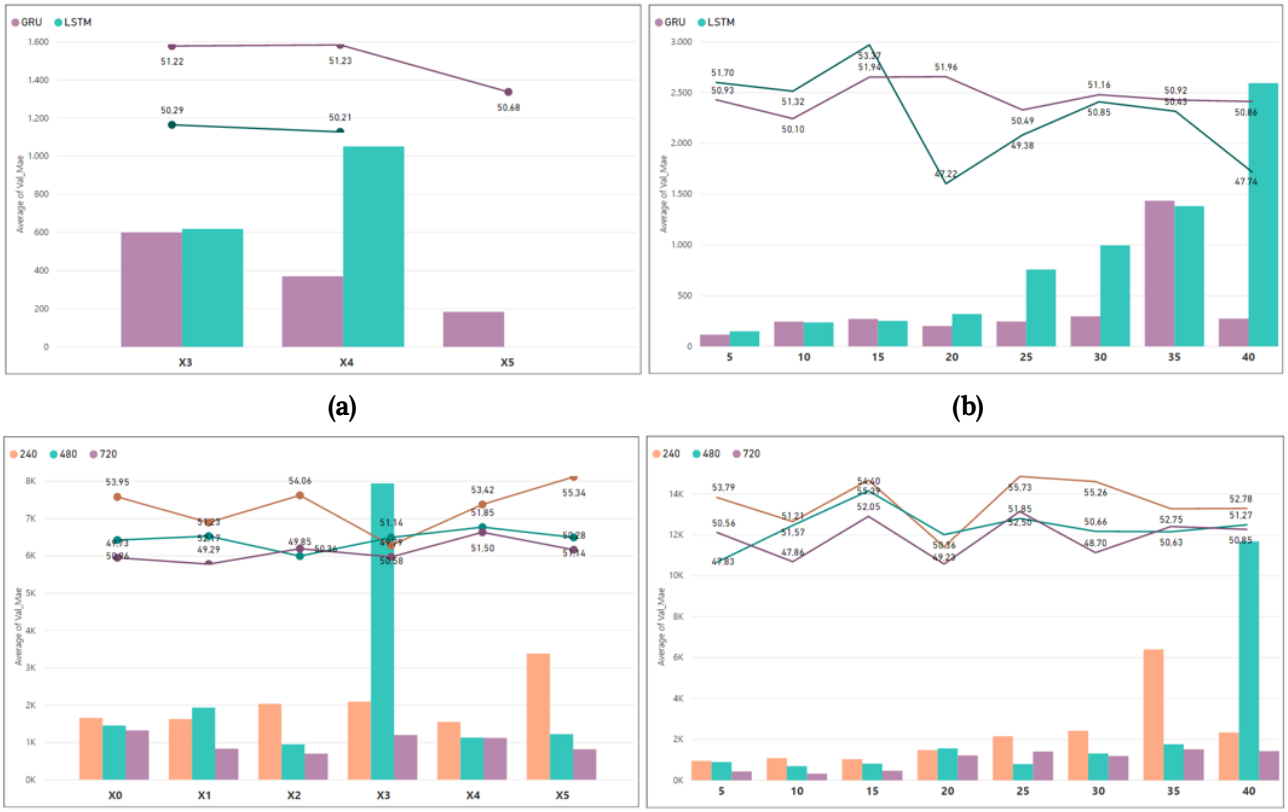
De otro lado, para los diferentes parámetros utilizados (activación *linear*, optimizadores *Adam*, *nadam* y *rmsprop*, pérdida *mse* y *msle*), el *look back* de 15 obtuvo la mejor exactitud promedio, entre los MAE más pequeños de la muestra.

X	X_Shape	Distribution	Accuracy	Val_Loss	Val_Mae	Look_Back	Layer_Type	Activation	Optimizer	Loss	Time
X5	240	Counter({1: 8, 0: 6})	71.43	2.90E+09	4.96E+04	35	LSTM	linear	adam	mse	14.92
X0	240	Counter({0: 10, 1: 9})	68.42	1.50E+07	3.87E+03	30	LSTM	linear	adam	mse	14.61
X5	240	Counter({1: 21, 0: 13})	67.65	8.42E-02	1.05E+03	15	GRU	linear	nadam	msle	15.75
X2	240	Counter({1: 5, 0: 4})	66.67	1.99E+06	1.16E+03	40	LSTM	linear	adam	mse	16.48
X3	240	Counter({1: 14, 0: 10})	66.67	1.49E+07	3.87E+03	25	LSTM	linear	adam	mse	14.29

X	X_Shape	Distribution	Accuracy	Val_Loss	Val_Mae	Look_Back	Layer_Type	Activation	Optimizer	Loss	Time
X0	480	Counter({1: 35, 0: 22})	66.67	1.07E+01	2.86E+03	40	GRU	linear	rmsprop	msle	46.24
X3	480	Counter({1: 36, 0: 26})	66.13	1.59E-01	1.18E+03	35	GRU	linear	rmsprop	msle	25.21
X1	480	Counter({0: 45, 1: 37})	64.63	1.43E+00	1.88E+03	15	GRU	linear	rmsprop	msle	17.26
X4	480	Counter({1: 49, 0: 33})	64.63	1.52E-01	1.08E+03	15	GRU	linear	nadam	msle	18.43
X5	480	Counter({1: 45, 0: 37})	64.63	1.26E-01	8.56E+02	15	GRU	linear	nadam	msle	17.52

X	X_Shape	Distribution	Accuracy	Val_Loss	Val_Mae	Look_Back	Layer_Type	Activation	Optimizer	Loss	Time
X4	720	Counter({1: 67, 0: 63})	60.77	1.06E-02	2.88E+02	15	GRU	linear	nadam	msle	21.75
X0	720	Counter({0: 59, 1: 46})	60.00	1.09E+07	3.30E+03	40	LSTM	linear	adam	mse	46.03
X4	720	Counter({1: 88, 0: 32})	59.17	9.99E+06	3.15E+03	25	LSTM	linear	adam	mse	25.82
X4	720	Counter({1: 80, 0: 40})	59.17	4.19E+00	2.24E+03	25	GRU	linear	nadam	msle	46.73
X4	720	Counter({0: 67, 1: 48})	58.26	2.97E-01	1.23E+03	30	GRU	linear	rmsprop	msle	46.11

***Tabla* 5.** Experimentos del Grupo E con mejores resultados según *Accuracy* para diferente cantidad de registros.



(a) **(b)**

(c) **(d)**

Figura 8. *Mae (columnas) y Accuracy promedio (líneas) por conjuntos de datos (gráficos de la izquierda) y look back (gráficos de la derecha), para experimentos de los grupos D (gráficos superiores) y E (gráficos inferiores)*

4.3. Discusión.

Un aspecto importante de los resultados obtenidos, radica en su uso en entornos reales. A lo largo del documento se han expuesto numerosas razones por las que los modelos de aprendizaje automático todavía no han dominado los mercados de valores, y a su vez varios factores por los cuales todavía no se consideran decisivos a la hora de tomar decisiones sobre portafolios de inversión.

La amplia investigación sobre métodos estadísticos y tecnológicos para sacar provecho de las series temporales financieras y la creciente disponibilidad de software y hardware han dado lugar a conceptos más aplicados como el High Frequency Trading (HFT) donde los altos volúmenes compensan los bajos márgenes de cada transacción.

En este artículo, el estudio estuvo centrado en la predicción diaria de la tendencia del precio de cierre del indicador S&P 500. Los enfoques que se dieron a las características durante el entrenamiento y la validación de los modelos, no sólo dan claridad sobre la dificultad que existe en encontrar relaciones complejas entre las variables, sino también la necesidad de incorporar otro tipo de información (incluso no estructurada).

Modelo	Test Accuracy	Dataset
Logistic Regression	64.04%	X5
LSTM	55.80%	X0
GRU	57.42%	X3
MLP	58.96%	X1

Tabla 6. Resumen resultados más relevantes.

Los resultados que se pueden observar en la tabla 6 evidencian una baja exactitud y refuerzan muchas de las premisas que se establecieron en el marco teórico y en la introducción. Es importante mencionar que aunque las clases para el periodo seleccionado, no tienen un desbalance significativo (54.92% up y 45.08% down) fue frecuente encontrar durante los entrenamientos algunos modelos en que la gran mayoría de predicciones eran hacia arriba (clase 1) y se obtenían buenos resultados en *accuracy* pero sin lograr establecer buenas relaciones entre las variables dependientes y la clase.

Con los modelos secuenciales elegidos a través de investigaciones anteriores y que indican sus bondades para este tipo de aplicaciones, no se observaron resultados que soporten la capacidad predictiva con relaciones de alta complejidad con la que sí se desempeñan en aplicaciones de predicción de texto e incluso traducción. La gran cantidad de variantes y ajustes que son posibles para este tipo de modelos establece una gran cantidad de experimentos que por ahora exceden el alcance de este artículo, entre los cuales se destacan los llamados mecanismos de atención o incluso las más recientes Redes Neuronales Transformers.

5. Conclusiones.

Como se mencionó en la introducción de este documento, la predicción de series temporales -en particular en lo que respecta a los movimientos bursátiles- es un reto increíblemente difícil que ha sido investigado a fondo a lo largo de la historia.

Además, cuando se analizan resultados aparentemente concluyentes, éstos podrían ser consecuencia de conjuntos de datos engañosamente adecuados, o resultantes de un exceso de ajuste en determinado periodo de tiempo analizado. Para el enfoque que se le dio a los modelos de Deep Learning de resolver el problema de regresión e interpretar los resultados como una tendencia binaria, fue una clara muestra de la importancia de incluir características que sí le confieran al modelo la capacidad de inferir, ya que lo evidenciado fue que a pesar de tener errores muy bajos en la experimentación para cualquiera de las funciones de pérdida (MAE, RMSE, etc.) al traducir esas predicciones a una secuencia binaria, el modelo no lograba predecir nada muy diferente al valor de cierre del día anterior.

El desempeño de los modelos mejora, regularmente, con la utilización de estadísticos que recogen la historia de de los datos como la media móvil, volatilidad, momento, rastreo de señal, entre otros. En el caso de los algoritmos típicos de clasificación, la transformación de los datos en sus componentes principales (PC) no tiene incidencias negativas.

Sin embargo, el modelo de Regresión Logística obtuvo resultados notables en cuanto a exactitud o *accuracy* y una buena distribución de clases (57.26% up y 42.74% down). Esto permite inferir que para el conjunto de datos seleccionado, y a pesar de las grandes limitaciones que este tipo de modelos pueden tener, con los hiperparámetros seleccionados el clasificador logra desprenderse del desbalance de clases y arrojar predicciones basadas en las relaciones establecidas durante el entrenamiento alcanzado un score total de 62.40% en entrenamiento y 64.04% en prueba.

La elección de un rezago -*look back*- múltiplo de 5, en los modelos de algoritmos secuenciales empleados, muestra mejores resultados. Esto podría explicarse en que el Mercado de Futuros donde se transa el S&P 500 funciona cinco de los siete días de la semana. Se observó que valores elevados del parámetro derivan en el incremento del error en la predicción del precio del indicador. Cabe destacar que para el modelo GRU, el comportamiento del error es menos sensible al incremento de dicho parámetro.

Los modelos secuenciales suelen tener un mejor comportamiento ante el incremento de la cantidad de registros en los datos de entrada. Este comportamiento se sostiene en la experimentación realizada -ver ***Figura 8d*** donde se observa que el MAE sube cuando aumentan los registros-. Empero, con el modelo diseñado, en el cual se predice el precio y finalmente se etiqueta con base en su tendencia (alza o baja), el desempeño de la métrica seleccionada -*Accuracy*-, es mejor con menor número de registros.

Si bien se ha mencionado en varias ocasiones que la predicción del precio de las acciones sigue siendo una tarea difícil incluso después de simplificar el problema a una tendencia de precios binaria y aplicar potentes modelos de aprendizaje profundo, con la elaboración de este artículo es posible orientar algún trabajo futuro, concretamente para determinar los conjuntos de características relevantes y las arquitecturas de los modelos que logren mejores resultados.

Como trabajo futuro, se establecen las siguientes direcciones:

- Por un lado podrían realizarse validaciones formales de la robustez de los modelos secuenciales cuando la información de entrada tiene pocos registros, es bien conocido que las redes neuronales requieren grandes cantidades de información para obtener niveles de desempeño satisfactorios durante el proceso de entrenamiento; esto permitiría evaluar el mismo modelo en diversos rangos de tiempo.

- Un ejercicio interesante que puede abordar la situación de pocos datos, sería la adquisición de los precios de cierre con frecuencias más elevadas (cada hora o incluso cada minuto) donde diferentes patrones pueden emerger y tener espacios de predicción y retroalimentación más frecuentes.

- Continuar enriqueciendo el conjunto de datos. La posibilidad de incluir features o características no técnicas, la información más relevante que se utilizan en los análisis fundamentales de las empresas podrían tener un impacto importante en las predicciones.

- Continuar con la experimentación de diferentes modelos, como el mecanismo de atención en redes LSTM, Redes Neuronales Gráficas (Graph Neural Networks) o incluso las Redes Neuronales Transformers mencionadas anteriormente. Para los modelos clásicos de Machine Learning, es menester introducir una estrategia en la cual los algoritmos no queden atrapados en el comportamiento alcista del indicador.

6. Referencias bibliográficas.

[1] BERGSTRÖM, C., & HJELM, O. (2019, 10 15). *Impact of Time Steps on Stock Market Prediction with LSTM*. Impact of Time Steps on Stock Market Prediction with LSTM. https://www.diva-portal.org/smash/get/diva2:1361305/FULLTEXT01.pdf

[2] SEGAL, T. (2021, 03 11). *Fundamental Analysis*. Investopedia. https://www.investopedia.com/terms/f/fundamentalanalysis.asp

[3] DUEMIG, D. (s.f.). *Predicting stock prices with LSTM Networks*. Predicting stock prices with LSTM Networks. http://cs230.stanford.edu/past-projects/

[4] KHANDELWAL, R. (2020, 4 6). Quick and Easy Explanation of Logistic Regression. Towards Data Science. https://towardsdatascience.com/quick-and-easy-explanation-of-logistics-regression-709df5cc3f1e

[5] PRABHAKARAN, S. (n.d.). Augmented Dickey Fuller Test (ADF Test) – Must Read Guide. Machine Learning Plus. https://www.machinelearningplus.com/time-series/augmented-dickey-fuller-test/

[6] KENTON, W. (2020, 03 17). Runs Test. Investopedia. https://www.investopedia.com/terms/r/runs_test.asp

[7] Yahoo Finance. (s.f.). Yahoo Finance. https://finance.yahoo.com/

[8] Statsmodels. (n.d.). Statsmodels. https://www.statsmodels.org/stable/index.html

[9] LOPEZ, D. (s.f.). Technical Analysis Library in Python. Technical Analysis Library in Python. https://technical-analysis-library-in-python.readthedocs.io/en/latest/index.html#

[10] ZOU, Z. (s.f.). Using LSTM in Stock prediction and Quantitative Trading. Using LSTM in Stock prediction and Quantitative Trading. http://cs230.stanford.edu/projects_winter_2020/reports/32066186.pdf

[11] CHALVATZIS, C., & HRISTU-VARSAKELIS, D. (2019, 05 09). High-performance stock index trading: making effective use of a deep long short-term memory network. https://arxiv.org/pdf/1902.03125.pdf

[12] Repositorio GitHub. (s.f.). https://github.com/chelaguza/S-P500.git