

**PREDICCIÓN DE TENDENCIA DEL PRECIO DE
ACTIVOS FINANCIEROS:
ANÁLISIS DE GRÁFICOS DE VELAS CON
APRENDIZAJE PROFUNDO**

María Paula Basigalup

Tutor: Ramiro Gálvez - Escuela de Negocios, Universidad Torcuato Di Tella

Julio 2022

Resumen

Existe un fuerte interés por parte de la industria financiera por tener predicciones precisas del comportamiento de los precios de los activos. El objetivo de esta tesis es ver si las técnicas modernas de procesamiento, análisis y predicción con imágenes pueden ser utilizadas con gráficos de velas para automatizar la identificación de patrones que permitan predecir la tendencia del precio de acciones a corto plazo.

En el presente trabajo se realizan experimentaciones con diversas metodologías, alternando entre varios conjuntos de datos, métodos de clasificación, y algoritmos predictivos. También se contrasta el desempeño de algoritmos de redes neuronales prediciendo tendencias de precios mediante el análisis de gráficos de velas, contra algoritmos más tradicionales de aprendizaje automático prediciendo la misma variable en base a indicadores técnicos.

Los resultados obtenidos por algoritmos de redes neuronales al analizar gráficos de velas superan al azar - lo cual es significativo dada la naturaleza desafiante de este problema predictivo - y al análisis tradicional de indicadores técnicos mediante algoritmos de aprendizaje automático. Esto sugiere que a partir de los gráficos de velas podría plantearse un modelo con cierta capacidad predictiva, y sería viable incorporarlo en estrategias de trading.

STOCK PRICE TREND PREDICTION: A DEEP LEARNING APPROACH ON CANDLESTICK CHART ANALYSIS

María Paula Basigalup

Thesis Advisor: Ramiro Gálvez - School of Business, Universidad Torcuato Di Tella

July 2022

Abstract

There is a strong interest in the financial industry to have accurate predictions of the behavior of stock prices. The purpose of this thesis project is to find out if modern image processing, analysis and prediction techniques can be used with candlestick charts to automate the identification of patterns that can predict short-term stock price trends.

In this paper, experiments are carried out with various methodologies, alternating between several data sets, classification methods and predictive algorithms. The performance of neural network algorithms predicting price trends by analyzing technical candlestick charts is additionally compared with traditional machine learning algorithms predicting the same variable based on technical indicators.

The results obtained when analyzing candlestick charts with neural network algorithms outperformed chance - which is significant given the challenging nature of the predictive problem - and more traditional analysis of technical indicators using machine learning algorithms. This suggests that, based on candlestick chart analysis, a deep learning framework with certain predictive capacity could be proposed, and it could be incorporated in trading strategies.

Índice

Resumen	1
Abstract	2
Índice	3
Índice de Tablas	5
Índice de Figuras	7
1. Introducción	10
1.1 Contexto	10
1.1.1 Mercado de capitales	10
1.1.2 Inteligencia Artificial	11
1.2 Propuesta de Trabajo	12
1.3 Trabajo Relacionado	12
1.4 Estructura de Trabajo	13
2. Metodología	14
2.1 Datos	14
2.1.1 Gráficos de velas	14
2.1.2 Procesamiento	15
2.2 Modelado	17
2.2.1 Uso de Aprendizaje Profundo con imágenes	17
2.2.2 Variable a predecir	21
2.2.3 Métricas de evaluación	24
2.2.4 Esquema de validación	26
3. Experimentación	27
3.1 Experimento 1: Identifying Candlestick Patterns using Deep Learning	27
3.1.1 Metodología original	28
3.1.2 Metodología corregida	32
3.1.3 Metodología alternativa	35
3.1.4 Conclusiones	70
3.2 Experimento 2: Using Deep Learning Neural Networks and Candlestick Chart Representation To Predict Stock Market	72
3.2.1 Metodología del paper	73
3.2.2 Metodología del Experimento 2	76
3.2.3 Replicación de resultados	77
3.2.4 Error	81
3.2.5 Experimentación	81
3.2.6 Conclusiones	90

3.3 Experimento 3: Predicción de tendencia de precios de activos financieros mediante el análisis de indicadores técnicos con Aprendizaje Automático	91
3.3.1 Metodología Experimento 3	92
3.3.2 Experimentación	97
3.3.3 Conclusiones	110
4. Discusión	111
4.1 Resumen de Resultados	111
4.2 Limitaciones y trabajo futuro	114
4.3 Conclusión	116
5. Bibliografía	117
6. Anexo	120
6.1 Listas de acciones utilizadas	120
6.1.1 Acciones de India	120
6.1.2 Acciones de Estados Unidos	120
6.1.3 Acciones de Indonesia	132
6.2 Experimento 2 - Código original del paper	133
6.2.1 Github del autor	133
6.2.2 Historial de ediciones	133
6.3 Experimento 3	137
6.3.1 Listado de atributos	137
6.3.2 Experimentación	141

Índice de Tablas

Tabla 1: Ejemplo de estructura de datos de cotización histórica	15
Tabla 2: Experimento 1 - Resultados con metodología original	30
Tabla 3: Experimento 1 - Partición de datos corregida	32
Tabla 4: Experimento 1 - Resultados con metodología corregida	33
Tabla 5: Experimento 1 - Resultados con clasificación alternativa ($k = 10$, $p = 0.025$, $t = 0.1$)	37
Tabla 6: Experimento 1 - Resultados con clasificación alternativa ($k = 5$, $p = 0.025$, $t = 0.05$)	39
Tabla 7: Experimento 1 - Resultados con media móvil simple de 9 días	44
Tabla 8: Experimento 1 - Resultados con media móvil simple de 20 días	46
Tabla 9: Experimento 1 - Resultados con medias móviles de 5 y 20 días	49
Tabla 10: Experimento 1 - Resultados con medias móviles de 4, 9 y 18 días	52
Tabla 11: Experimento 1 - Resultados con Bandas de Bollinger por período de 20 días	56
Tabla 12: Experimento 1 - Resultados con Bandas de Bollinger por período de 9 días	58
Tabla 13: Experimento 1 - Resultados con datos alternativos	64
Tabla 14: Experimento 1 - Resultados con datos alternativos, clasificación original y bandas de Bollinger por período de 20 días	66
Tabla 15: Experimento 1 - Resultados con datos alternativos, clasificación alternativa ($k=10$, $p=0.025$, $t=0.1$) y bandas de Bollinger por período de 20 días	68
Tabla 16: Resultados del Experimento 1 con datos de India	70
Tabla 17: Resultados del Experimento 1 con datos de Estados Unidos	71
Tabla 18: Experimento 3 - Conjuntos de datos utilizados	73
Tabla 19: Experimento 2 - Resultados con datos de Indonesia + Clasificación paper + CNN	78
Tabla 20: Experimento 2 - Resultados con datos de Indonesia + Clasificación inicial + CNN	80
Tabla 21: Experimento 2 - Resultados con datos de Indonesia + Clasificación paper + Resnet34	83
Tabla 22: Experimento 2 - Resultados con datos de India + Clasificación paper + CNN	85
Tabla 23: Experimento 2 - Resultados con datos de India + Clasificación paper + Resnet34	87
Tabla 24: Experimento 2 - Resultados con datos de India + Clasificación paper + Resnet34 + Bandas de Bollinger de 20 días	89
Tabla 25: Resultados del Experimento 2 con datos de Indonesia	91
Tabla 26: Resultados del Experimento 2 con datos de India	91
Tabla 27: Experimento 3 - Hiperparámetros de algoritmo XGBoost	97

Tabla 28: Experimento 3 - Resultados para datos de India + modelo con todos los atributos e hiperparámetros predeterminados	99
Tabla 29: Experimento 3 - Resultados para datos de India + modelo con atributos selectos e hiperparámetros predeterminados	100
Tabla 30: Experimento 3 - Resultados para datos de India + modelo con todos los atributos e hiperparámetros optimizados	102
Tabla 31: Experimento 3 - Resultados para datos de India + modelo con atributos selectos e hiperparámetros optimizados	103
Tabla 32: Experimento 3 - Resultados para datos de EE.UU. + modelo con todos los atributos e hiperparámetros predeterminados	105
Tabla 33: Experimento 3 - Resultados para datos de EE.UU.+ modelo con atributos selectos e hiperparámetros predeterminados	106
Tabla 34: Experimento 3 - Resultados para datos de EE.UU. + modelo con todos los atributos e hiperparámetros optimizados	108
Tabla 35: Experimento 3 - Resultados para datos de EE.UU. + modelo con atributos selectos e hiperparámetros optimizados	109
Tabla 36: Resultados del Experimento 3 con datos de India	111
Tabla 37: Resultados del Experimento 3 con datos de Estados Unidos	111
Tabla 38: Resultados de experimentación con datos de India	113
Tabla 39: Resultados de experimentación con datos de Estados Unidos	114
Tabla 40: Lista de acciones de India	121
Tabla 41: Lista de acciones de Estados Unidos	121
Tabla 42: Lista de acciones de Indonesia	133
Tabla 43: Experimento 3 - Lista de atributos	138

Índice de Figuras

Figura 1: Interpretación de las velas	14
Figura 2: Ejemplo de gráfico de velas con volumen	15
Figura 3: Ejemplo de gráfico de velas con volumen	16
Figura 4: Ejemplo de gráfico de velas con volumen y medias móviles de 5 y 20 días	16
Figura 5: Ejemplo de gráfico de velas con volumen y bandas de Bollinger de 20 días	16
Figura 6: Esquema de red neuronal profunda	17
Figura 7: Descomposición de imagen RGB 4x4x3	18
Figura 8.a: Ejemplo de ecuación de convolución	18
Figura 8.b: Ejemplo de operación de convolución con imagen en escala de grises	19
Figura 9: Ejemplo de la arquitectura de una CNN	19
Figura 10: Ejemplo de curva ROC	25
Figura 11: Esquema de split de conjuntos de datos de series temporales	26
Figura 12: Experimento 1 - Extracto del código original donde se ejecutan la clasificación y generación de imágenes	29
Figura 13: Experimento 1 - AUC-ROC con metodología original	31
Figura 14: Experimento 1 - Matriz de confusión con metodología original	31
Figura 15: Extracto del código original donde se realiza la partición de datos aleatoria	32
Figura 16: Experimento 1 - AUC-ROC con metodología corregida	33
Figura 17: Experimento 1 - Matriz de confusión con metodología corregida	34
Figura 18: Experimento 1 - AUC-ROC con clasificación alternativa ($k=10, p=0.025, t=0.1$)	37
Figura 19: Experimento 1 - Matriz de confusión con clasificación alternativa ($k=10, p=0.025, t=0.1$)	38
Figura 20: Experimento 1 - AUC-ROC con clasificación alternativa ($k=5, p=0.025, t=0.05$)	39
Figura 21: Experimento 1 - Matriz de confusión con clasificación alternativa ($k=5, p=0.025, t=0.05$)	40
Figura 22: Experimento 1 - AUC-ROC con media móvil simple de 9 días	44
Figura 23: Experimento 1 - Matriz de confusión con media móvil simple de 9 días	45
Figura 24: Experimento 1 - AUC-ROC con media móvil simple de 20 días	46
Figura 25: Experimento 1 - Matriz de confusión con media móvil simple de 20 días	47
Figura 26: Método de doble cruce	48
Figura 27: Experimento 1 - AUC-ROC con medias móviles de 5 y 20 días	49
Figura 28: Experimento 1 - Matriz de confusión con medias móviles de 5 y 20 días	50

Figura 29: Método de triple cruce con medias móviles de 4, 9 y 18 días	51
Figura 30: Experimento 1 - AUC-ROC con medias móviles de 4, 9 y 18 días	53
Figura 31: Experimento 1 - Matriz de confusión con medias móviles de 4, 9 y 18 días	53
Figura 32: Ejemplo de envolventes porcentuales	54
Figura 33: Bandas de Bollinger alrededor de una media móvil de 20 días	55
Figura 34: Experimento 1 - AUC-ROC con Bandas de Bollinger por período de 20 días	57
Figura 35: Experimento 1 - Matriz de confusión con Bandas de Bollinger por período de 20 días ...	57
Figura 36: Experimento 1 - AUC-ROC con Bandas de Bollinger por período de 9 días	59
Figura 37: Experimento 1 - Matriz de confusión con Bandas de Bollinger por período de 9 días	59
Figura 38: Experimento 1 - AUC-ROC con datos alternativos	65
Figura 39: Experimento 1 - Matriz de confusión con datos alternativos	65
Figura 40: Experimento 1 - AUC-ROC con datos alternativos, clasificación original y bandas de Bollinger por período de 20 días	67
Figura 41: Experimento 1 - Matriz de confusión con datos alternativos, clasificación original y bandas de Bollinger por período de 20 días	67
Figura 42: Experimento 1 - AUC-ROC con datos alternativos, clasificación alternativa ($k=10$, $p=0.025$, $t=0.1$) y bandas de Bollinger por período de 20 días	69
Figura 43: Experimento 1 - Matriz de confusión con datos alternativos, clasificación alternativa ($k=10$, $p=0.025$, $t=0.1$) y bandas de Bollinger por período de 20 días	69
Figura 44: Experimento 2 - Extracto del código donde se realiza la clasificación mencionada	75
Figura 45: Experimento 2 - Arquitectura de la CNN propuesta en el paper	75
Figura 46: Experimento 2 - AUC-ROC con datos de Indonesia + Clasificación paper + CNN	77
Figura 47: Experimento 2 - Matriz de confusión con datos de Indonesia + Clasificación paper + CNN	78
Figura 48: Experimento 2 - AUC-ROC con datos de Indonesia + Clasificación inicial + CNN	79
Figura 49: Experimento 2 - Matriz de confusión con datos de Indonesia + Clasificación inicial + CNN	80
Figura 50: Experimento 2 - AUC-ROC con datos de Indonesia + Clasificación paper + Resnet34	83
Figura 51: Experimento 2 - Matriz de confusión con datos de Indonesia + Clasificación paper + Resnet34	83
Figura 52: Experimento 2 - AUC-ROC con datos de India + Clasificación paper + CNN	85
Figura 53: Experimento 2 - Matriz de confusión con datos de India + Clasificación paper + CNN	85
Figura 54: Experimento 2 - AUC-ROC con datos de India + Clasificación paper + Resnet34	87
Figura 55: Experimento 2 - Matriz de confusión con datos de India + Clasificación paper + Resnet34	87
Figura 56: Experimento 2 - AUC-ROC con datos de India + Clasificación paper + Resnet34 + Bandas de Bollinger de 20 días	89

Figura 57: Experimento 2 - Matriz de confusión con datos de India + Clasificación paper + Resnet34 + Bandas de Bollinger de 20 días	89
Figura 58: Experimento 3 - Matriz de confusión para datos de India + modelo con todos los atributos e hiperparámetros predeterminados	98
Figura 59: Experimento 3 - Matriz de confusión para datos de India + modelo con atributos selectos e hiperparámetros predeterminados	100
Figura 60: Experimento 3 - Matriz de confusión para datos de India + modelo con todos los atributos e hiperparámetros optimizados	101
Figura 61: Experimento 3 - Matriz de confusión para datos de India + modelo con atributos selectos e hiperparámetros optimizados	103
Figura 62: Experimento 3 - Matriz de confusión para datos de EE.UU.+ modelo con todos los atributos e hiperparámetros predeterminados	104
Figura 63: Experimento 3 - Matriz de confusión para datos de EE.UU. + modelo con atributos selectos e hiperparámetros predeterminados	106
Figura 64: Experimento 3 - Matriz de confusión para datos de EE.UU. + modelo con todos los atributos e hiperparámetros optimizados	107
Figura 65: Experimento 3 - Matriz de confusión para datos de EE.UU. + modelo con atributos selectos e hiperparámetros optimizados	109
Figura 66: Experimento 2 - Resumen del historial de ediciones	133
Figura 67: Experimento 2 - Commit inicial (09/07/2018)	134
Figura 68: Experimento 2 - Edición #1: "fixed preprocess not works correctly" (13/04/2019)	135
Figura 69: Experimento 2 - Edición #2: "fixed labelling" (13/04/2019)	136
Figura 70: Experimento 3 - Espacio de hiperparámetros definido	141
Figura 71: Experimento 3 - Importancia de atributos del modelo base con hiperparámetros predeterminados	142
Figura 72: Experimento 3 - Atributos seleccionados del modelo base con hiperparámetros predeterminados	143
Figura 73: Experimento 3 - Importancia de atributos del modelo base con hiperparámetros predeterminados	144
Figura 74: Experimento 3 - Atributos seleccionados del modelo base con hiperparámetros predeterminados	145

1. Introducción

1.1 Contexto

1.1.1 Mercado de capitales

Personas de todo el mundo invierten en activos financieros con fines especulativos. Estos inversores gestionan su cartera de inversión en gran medida en función del valor esperado futuro de los activos, comprando un activo cuyo valor se espera que aumente y vendiendo o tomando posiciones de short para los que se espera que caiga. De esta manera, existe un fuerte interés por parte de la industria financiera por tener predicciones precisas de las tendencias en los precios.

La predicción de los precios/retornos futuros de los activos representa un problema desafiante de predicción, pues sumado a que existen muchos factores que afectan el precio de los instrumentos financieros (por ejemplo las noticias, el rendimiento de la empresa, el rendimiento de la industria, el sentimiento de los inversores, factores económicos, entre otros) se tiene que el mismo surge a partir de la interacción de múltiples agentes que buscan maximizar sus retornos futuros.

Eugene Fama (1970) sostiene que un mercado es eficiente cuando el precio de un activo en el mercado refleja toda la información disponible que existe (histórica, pública y privada), y considera que cualquier noticia o evento futuro que pueda afectar a la cotización de un activo hará que el precio se ajuste tan rápido, que sea imposible obtener un beneficio económico del mismo. Dado esto, se considera una pérdida de tiempo y dinero el intentar hacer un análisis de los valores, ya que no habrá ningún activo infravalorado o sobrevalorado en el mercado (J. López Abellán, 2018).

Aún así, en el mundo financiero hay una larga tradición de utilizar esquemas analíticos para intentar predecir rendimientos o el comportamiento de diferentes activos financieros. Uno de ellos es conocido como el análisis técnico, y se originó en EEUU a fines del siglo XIX con la Teoría de Dow¹ planteada por Charles Henry Dow, luego impulsada por Ralph Nelson Elliott con su Teoría de las Ondas de Elliott². Más concretamente, el análisis técnico sostiene que por medio de indicadores y una lectura adecuada de los gráficos de precio y de volumen negociado de las acciones se puede tomar decisiones en el mercado bursátil, sin considerar los fundamentos que mueven a la economía y a las firmas. Esta postura es confrontada por un segundo esquema conocido como análisis fundamental, cuya visión consiste en predecir el futuro de los instrumentos financieros desde la información geopolítica, económica y teórica, entre otras.

Mark P. Taylor y Helen Allen (1992) muestran, a través de cuestionarios, que el análisis técnico es utilizado como fuente de información primaria o secundaria por más del 90% de

¹ Teoría que permite comprender el comportamiento del mercado utilizando un conjunto de reglas técnicas respecto al análisis de los precios de cierre de gráficos diarios.

² Teoría basada en el estudio de las ondas o patrones que forman los precios del mercado financiero en su formación gráfica.

los agentes de mercado en Londres, y que el 60% considera que el análisis técnico es al menos tan importante como el análisis fundamental. Esto demuestra que efectivamente el análisis técnico resulta indispensable al momento de operar en el mercado financiero.

Dentro del análisis técnico, una estrategia de predicción se basa en lo que se conoce como análisis de gráficos de evolución de los activos, siendo el gráfico de velas el más popular de ellos. Existe literatura que sugiere que distintos patrones en los mismos se asocian a subas o bajas de los precios de los activos: “muchas veces es posible anticipar reacciones, normalmente con una visión de corto plazo. [...] Este tipo de gráfico aporta una información inestimable para el planteamiento de las tácticas y los momentos de realizar las inversiones.” (Codina, 2011)

1.1.2 Inteligencia Artificial

“La visión por computadora (computer vision) es un campo de la inteligencia artificial que entrena a las computadoras para interpretar y comprender el mundo visual. Usando imágenes digitales de cámaras y videos y modelos de aprendizaje profundo, las máquinas pueden identificar y clasificar objetos con precisión, y luego reaccionar a lo que ‘ven’.” (Jara, 2021)

La tecnología de visión artificial ha recorrido un largo camino desde su aparición en la década de 1950. En los últimos años se ha producido una base muy grande en la tecnología de análisis y reconocimiento de imágenes, y predicción con imágenes, principalmente impulsada por avances en las técnicas de deep learning.

Hoy en día, se encuentran ejemplos de uso de estas tecnologías en la mayoría de las industrias. Según Business Wire (2016), se estimó que su mercado de software y hardware alcanzaría los \$48.6 billones de dólares este año. Forbes (Marr, 2019) destaca algunos de estos casos, entre ellos los vehículos autónomos desarrollados por Tesla, BMW, Volvo y Audi, que a través de cámaras, radares y sensores capturan imágenes para detectar objetos y señales y conducir con seguridad. Otra aplicación ampliamente utilizada es la del reconocimiento facial, China está a la vanguardia en este campo, ya que según Reuters (Goh, 2019) la usan para el trabajo policial, los portales de pago, los controles de seguridad en el aeropuerto, entre otros. La industria de la salud presenta una gran cantidad de usos para la visión artificial ya que el 90% de todos los datos médicos se basan en imágenes: desde la habilitación de nuevos métodos de diagnóstico médico para analizar radiografías, mamografías y otras exploraciones hasta el control de pacientes para identificar problemas de manera temprana y predecir patologías (Ho, Zhao, Chen et al, 2022).

1.2 Propuesta de Trabajo

Existen muchas herramientas para ayudar a las personas a pronosticar los precios de las acciones, como modelos de regresión estadística, análisis de series de tiempo y árboles de clasificación. Sin embargo, las mismas rara vez consideran los patrones que los analistas técnicos estudian en los gráficos bursátiles.

El claro avance y adopción de la visión artificial en múltiples ámbitos, plantea la duda de si estas técnicas modernas de procesamiento, análisis y predicción con imágenes pueden ser utilizadas con gráficos de velas para automatizar la identificación de patrones y predecir de alguna manera el comportamiento del precio de los activos financieros a corto plazo.

Algunas personas ya han comenzado a trabajar en esta hipótesis, y sus trabajos presentan resultados notoriamente alentadores. En esta tesis nos propusimos inspeccionar esos modelos, ver si efectivamente sus conclusiones son acertadas, y en caso de serlo, evaluar posibles mejoras.

1.3 Trabajo Relacionado

Hay literatura incipiente respecto al uso de algoritmos de inteligencia artificial para predecir el comportamiento de los precios de los activos mediante imágenes de sus gráficos de velas. Entre varios trabajos que fueron inspeccionados, a continuación se mencionan los dos que resultaron más interesantes y se analizaron en este trabajo.

En el paper *Using Deep Learning Neural Networks and Candlestick Chart Representation to Predict Stock Market* (Kusuma, Ho, Kao, Ou, Hua, 2019) se evalúan varios algoritmos de aprendizaje automático y su mejor modelo de redes convolucionales alcanza una precisión de predicción mediante gráficos de velas que supera el 90%.

En el blog *Identifying Candlestick Patterns using Deep Learning* (Shah, 2020) publicado en la página *Towards Data Science* se presenta un modelo pre entrenado que alcanza una performance bastante buena de 62%.

Cuando analizamos dichos trabajos y evaluamos las metodologías implementadas en los mismos, encontramos que efectivamente estos experimentos tenían problemas metodológicos y que al ser corregidos baja dramáticamente la performance de sus algoritmos.

1.4 Estructura de Trabajo

El resto de este documento se encuentra estructurado principalmente en tres secciones:

- Metodología: se presentarán los datos, métodos y conceptos a utilizar.
- Experimentación: se abordarán los tres experimentos explorados en este trabajo.
- Discusión: se planteará un resumen de resultados de la experimentación y las limitaciones afrontadas, como así también trabajo futuro, y por último la conclusión final del trabajo.

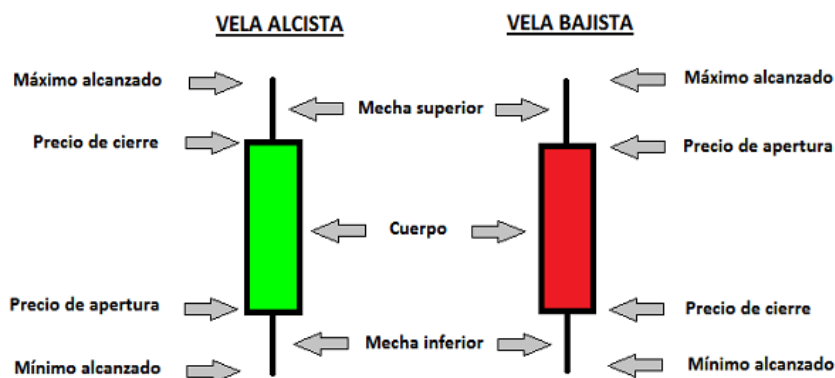
2. Metodología

2.1 Datos

2.1.1 Gráficos de velas

El gráfico de velas (también conocido como “*Candlestick*” en inglés) es un gráfico que se utiliza para describir los movimientos de precios de las acciones durante un período de tiempo determinado (días, horas, etc.). Ayuda a entender la relación entre el precio máximo y el mínimo así como la apertura y el cierre, de una manera más visual. El color de cada vela indica si en el periodo de cada vela el precio de apertura (el que tuvo al comienzo del periodo) fue mayor o menor al de cierre (el que tuvo a final del periodo). La Figura 1 ilustra cómo debe interpretarse cada vela en un gráfico de velas.

Figura 1: Interpretación de las velas (Fuente: Regader, 2019)



Otra información importante a incluir en el gráfico es el volumen. El volumen representa la cantidad total de actividad comercial en ese mercado para ese día. Es el número total de acciones ordinarias que cambian de manos en un día determinado en el mercado de valores. El volumen se registra mediante una barra vertical en la parte inferior del gráfico debajo de la barra de precios de ese día. (Murphy, 1999)

Figura 2: Ejemplo de gráfico de velas con volumen (TradingView, 2022)



2.1.2 Procesamiento

Como se verá más adelante en la etapa de experimentación, abordaremos varios experimentos que seguirán metodologías diferentes. Sin embargo, algo que todos tendrán en común será la estructura de los datos que alimentarán a cada modelo.

Estaremos trabajando con precios históricos de activos financieros cuya cotización es pública. Entre cada experimento irán variando los mercados, conjuntos de acciones a utilizar y períodos temporales estudiados, pero la lógica aplica a todos por igual.

Para cada activo analizado se utilizará la siguiente información: fecha, símbolo, precio máximo, precio mínimo, precio de apertura, precio de cierre, precio de cierre ajustado y volumen. Accederemos a dicha información a través de la API de Yahoo Finance, obteniendo los datos estructurados de la siguiente manera para cada acción y período temporal especificado:

Tabla 1: Ejemplo de estructura de datos de cotización histórica

Date	Open	High	Low	Close	Adj Close	Volume	ticker
1/2/2009	3.067143	3.251429	3.041429	3.241071	2.795913	746015200	AAPL
1/5/2009	3.3275	3.435	3.311071	3.377857	2.913912	1181608400	AAPL
1/6/2009	3.426786	3.470357	3.299643	3.322143	2.865849	1289310400	AAPL
1/7/2009	3.278929	3.303571	3.223572	3.250357	2.803923	753048800	AAPL

Habr  una instancia de preprocesamiento de los datos, donde la informaci n mencionada ser  manipulada para crear una variedad de gr ficos de vela para cada acci n, los cuales ser n guardados como im genes para ser procesados como los inputs de los modelos. Utilizando la librer a *mplfinance*³, desarrollamos un c digo de armado de gr ficos lo suficientemente vers til como para poder jugar con las variantes de los gr ficos, con lo cual es posible agregar al gr fico de velas otros componentes adicionales a partir de los datos mencionados: barras de volumen, medias m viles de los precios de cierre y bandas de Bollinger⁴ (especificando los per odos temporales de las medias por supuesto). A continuaci n se muestran ejemplos de im genes de gr ficos producidos con nuestro c digo:

Figura 3: Ejemplo de gr fico de velas con volumen



Figura 4: Ejemplo de gr fico de velas con volumen y medias m viles de 5 y 20 d as



Figura 5: Ejemplo de gr fico de velas con volumen y bandas de Bollinger de 20 d as



³ <https://pypi.org/project/mpl-finance/>

⁴ Como veremos m s adelante, las bandas de Bollinger son envolventes trazadas a una desviaci n est ndar de distancia por encima y por debajo de una media m vil simple del precio de cierre.

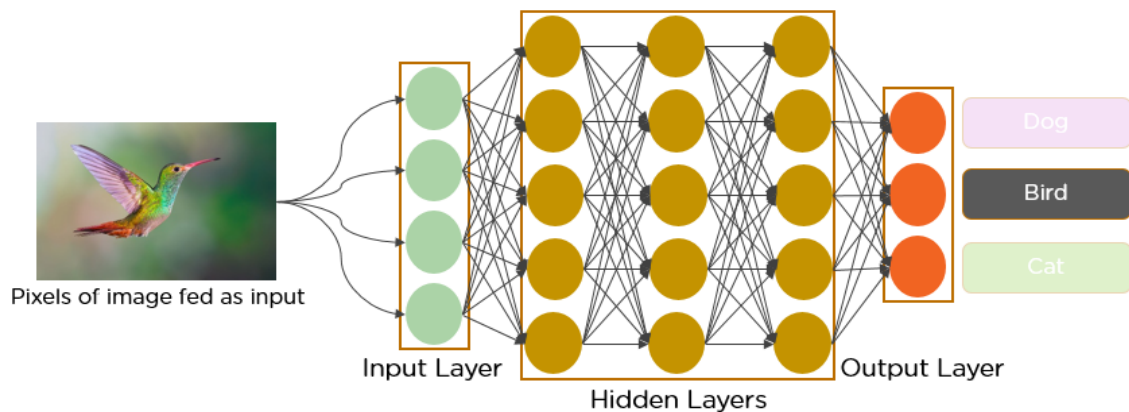
2.2 Modelado

2.2.1 Uso de Aprendizaje Profundo con imágenes

El aprendizaje profundo es un tipo de aprendizaje automático que replica la estructura y funcionamiento del cerebro humano, utilizando redes neuronales artificiales para realizar cálculos sofisticados en grandes cantidades de datos.

Una red neuronal profunda consta de neuronas artificiales, también conocidas como nodos. Estos nodos se apilan uno al lado del otro en tres capas: capa de entrada (primera capa de la red neuronal), capa oculta (todas las capas intermedias de la red neuronal) y capa de salida (última capa de la red neuronal).

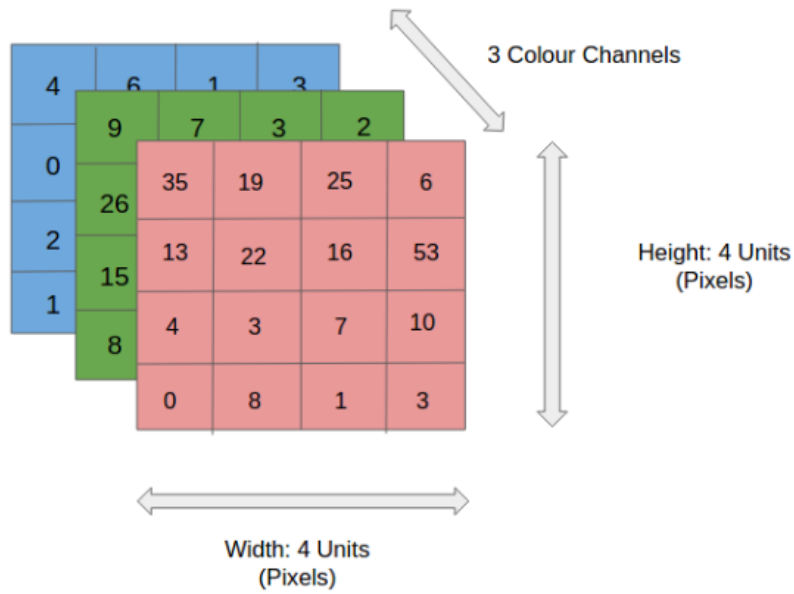
Figura 6: Esquema de red neuronal profunda (Fuente: Mandal, 2021)



La red neuronal convolucional (CNN) es la clase de red neuronal profunda que mayor avance tuvo en los últimos años en el campo de visión artificial (Computer Vision), y es la más comúnmente aplicada en problemas de análisis de imágenes, por lo que estaremos utilizando este tipo de redes en nuestros modelos. Esta red puede tomar una imagen de entrada, asignar importancia (pesos y sesgos aprendibles) a varios aspectos/objetos en la imagen y ser capaz de diferenciar uno del otro. El preprocesamiento requerido en la CNN es mucho menor en comparación con otros algoritmos de clasificación, ya que mientras que en los métodos tradicionales los regresores están diseñados a mano, con suficiente capacitación, la CNN tiene la capacidad de aprender estos filtros/características (Saha, 2018).

Para entender cómo una imagen puede ser pasada como input a estos modelos, tengamos en cuenta que una imagen no es más que una matriz de valores de píxeles, en el caso de una imagen a color la misma tiene tres planos (RGB: rojo, verde y azul), mientras que en el caso de una imagen en escala de grises tiene un solo plano (negro). En nuestro caso, estaremos trabajando con imágenes a color.

Figura 7: Descomposición de imagen RGB 4x4x3 (Fuente: Saha 2018)



La CNN utiliza una técnica especial llamada convolución. La convolución es una operación matemática sobre dos funciones que produce una tercera función que expresa cómo la forma de una es modificada por la otra. Esto se hace utilizando un filtro (también conocido como Kernel), que es una pequeña matriz cuyos valores se denominan pesos (Parveez, Iriondo, 2021).

Figura 8.a: Ejemplo de ecuación de convolución (Fuente: Parveez, Iriondo, 2021)

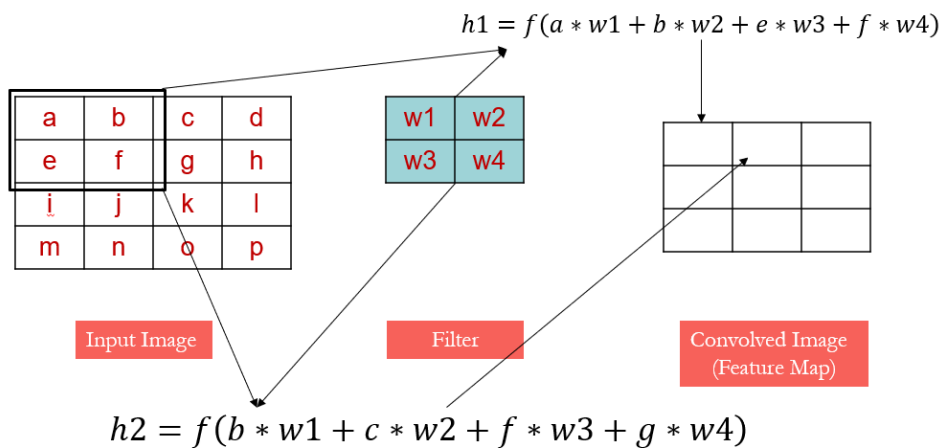
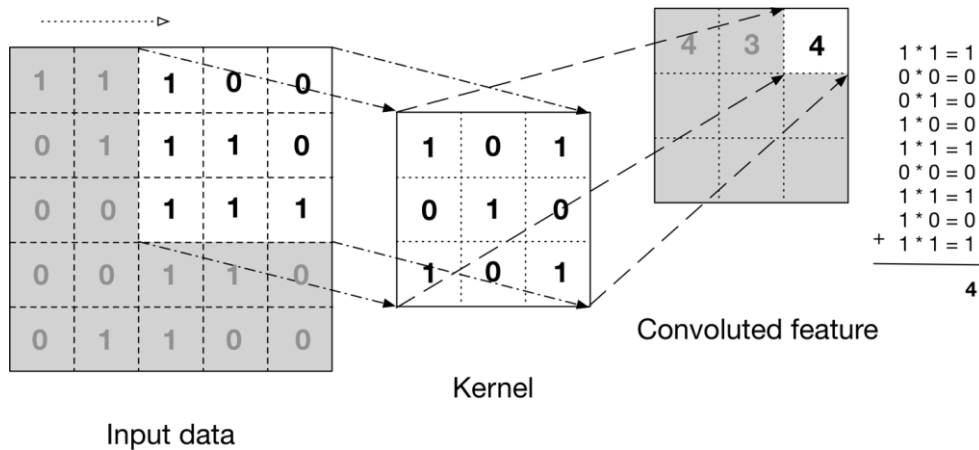


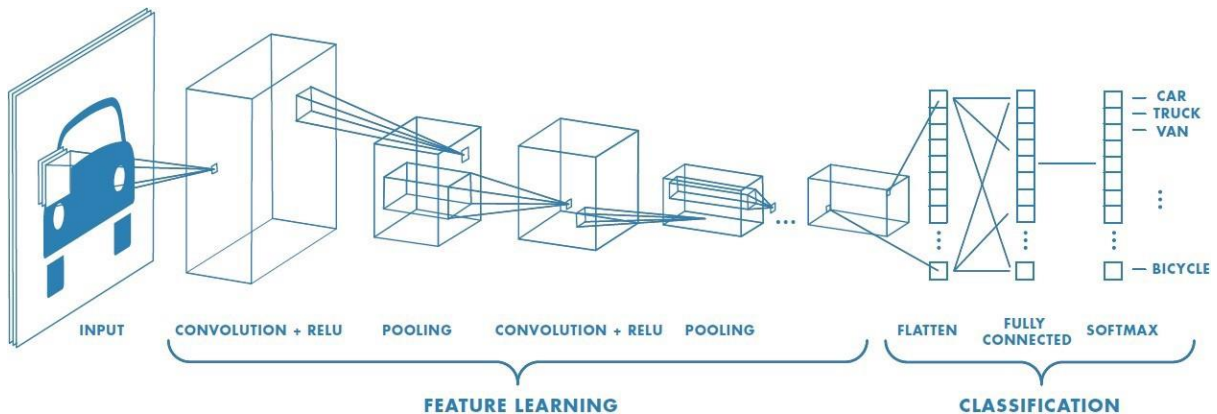
Figura 8.b: Ejemplo de operación de convolución con imagen en escala de grises (Fuente: Mandal, 2021)



La arquitectura de una CNN suele constar de dos instancias clave:

- Una etapa de extracción de características (feature extraction), donde la herramienta de convolución separa e identifica las distintas características de una imagen para su análisis.
- Una etapa de clasificación, donde una capa totalmente conectada (fully connected) toma el resultado del proceso de convolución y predice la clase de la imagen en función de las características recuperadas anteriormente.

Figura 9: Ejemplo de la arquitectura de una CNN (Fuente: Saha 2018)



Entrando más en detalle, las CNN suelen componerse de tres tipos de capas: capas convolucionales, capas de agrupación y capas totalmente conectadas. A continuación especificamos los componentes de la CNN:

- Capas de convolución

Esta es la primera capa de la CNN que es responsable de la extracción de las diferentes características de las imágenes de entrada. La misma tiene varios filtros que realizan la operación convolucional. Las capas convolucionales se aplican a entradas bidimensionales.

- Capa de agrupación (pooling) o de muestreo (subsampling)

Una capa de agrupación generalmente se aplica después de una capa convolucional. Fundamentalmente, la capa de agrupación se utiliza para reducir la dimensionalidad de la imagen, reduciendo así el número de parámetros y el tamaño espacial en la red, y finalmente los gastos computacionales.

- Capa totalmente conectada (Fully connected)

Esta capa es la suma de todas las entradas y ponderaciones que determinan la predicción final, que representa la salida de la última capa de agrupación. Se utiliza para conectar las neuronas entre dos capas separadas, realizando clasificación en base a las características extraídas por las capas anteriores (Parveez, Iriondo, 2021).

- Dropout

Para evitar el overfitting (cuando un modelo funciona bien con datos de entrenamiento pero no con datos nuevos), se utiliza una capa de dropout, en la que se eliminan algunas neuronas de la red neuronal durante la fase de entrenamiento, lo que da como resultado un modelo más pequeño.

- Funciones de activación

La función de activación devuelve una salida que será generada por la neurona dada una entrada o conjunto de entradas. Cada una de las capas que conforman la red neuronal tienen una función de activación que permitirá reconstruir o predecir (Freire, Silva, 2019).

2.2.2 Variable a predecir

El problema en cuestión podría ser atacado como un problema de regresión (predecir el precio de mañana, por ejemplo) o de clasificación (predecir si mañana el precio sube o baja, por ejemplo).

Decidimos abordarlo como un problema de clasificación binaria, y predecir si el precio ‘sube’ o ‘baja’ dentro de algún rango de tiempo (llamaremos “Up” y “Down” a las clases que tomará la variable Y de aquí en adelante). Teniendo esto en cuenta, inevitablemente la próxima decisión que se debe tomar es cómo armar la variable a predecir.

Nótese la complejidad del problema de predicción que estamos afrontando, donde la construcción de la variable objetivo juega un rol importante, o incluso determinante, en la performance que pudiera alcanzar el modelo predictivo que se desarrolle.

A lo largo de los diversos experimentos utilizaremos y evaluaremos las dos siguientes alternativas de variable objetivo:

a. Comportamiento del precio del día siguiente

La alternativa más simple, básicamente partiendo del día i , predecir si el precio de cierre de mercado subirá o bajará al día siguiente, $i+1$:

$$Y_i = \begin{cases} Down & \text{si } C_{i+1} > C_i \\ Up & \text{si } C_{i+1} \leq C_i \end{cases} \quad \text{donde } C \text{ es el precio de cierre} \quad (1)$$

b. Tendencia del precio dentro de los próximos días

Predecir la tendencia del precio de una acción en los próximos días (período a definir), podría llegar a tener más utilidad desde un punto de vista de inversión, con lo cual también nos interesa definir una variable Y que clasifique dicha tendencia. Ahora bien, ¿cómo se define la *tendencia* del precio de un activo financiero? O mejor dicho, ¿se puede identificar específicamente lo que implica dicho término?

La variable objetivo que buscamos predecir no es un valor absoluto, sino que es la construcción de una interpretación conceptual. Por lo tanto, no hay una única forma de definirla, sino que puede diferir según el criterio de cada persona.

Luís Torgo (2017) explora este tema en el capítulo 5 ‘Predicting Stock Market Returns’ de su libro *Data Mining with R - Learning with Case Studies*, y plantea la interpretación y construcción para la *tendencia* del precio de un activo financiero descrita a continuación⁵.

⁵ Texto original en inglés, traducido al español por Paula Basigalup.

Se busca obtener una estimación de la tendencia del mercado en los próximos días, para tomar acciones comerciales de compra/venta en base a dicha predicción. Asumimos que si los precios varían más de $p\%$, consideramos que esto vale la pena en términos de trading (cubrimos costos de transacción, etc.). En este contexto, queremos que nuestro modelo pronostique si este margen es alcanzable en los próximos k días.

Describiremos una variable, calculada con los datos de cotizaciones diarias, que se puede ver como un indicador (un valor) de la tendencia en los próximos k días. El valor de este indicador debe estar relacionado con la confianza que tenemos de que el margen objetivo p será alcanzable en los próximos k días. En esta etapa es importante notar que cuando mencionamos una variación de $p\%$, nos referimos a por encima o por debajo del precio actual. La idea es que variaciones positivas nos lleven a comprar, mientras que las variaciones negativas generarán acciones de venta. El indicador que estamos proponiendo resume la tendencia como un valor único, positivo para tendencias de precios al alza y negativo para tendencias a la baja.

Aproximamos el precio diario promedio como:

$$\bar{P}_i = \frac{C_i + H_i + L_i}{3} \quad (2)$$

donde C_i , H_i , L_i son las cotizaciones de cierre, máxima y mínima para el día i , respectivamente.

Sea V_i el conjunto de k variaciones porcentuales (a menudo llamadas rendimientos aritméticos) del precio de cierre de hoy contra los precios promedio de los siguientes k días:

$$V_i = \left\{ \frac{P_{i+j} - C_i}{C_i} \right\}_{j=1}^k \quad (3)$$

Nuestra variable indicadora es la suma total de las variaciones de este conjunto cuyo valor absoluto está por encima de nuestro margen objetivo $p\%$:

$$T_i = \sum \{v \in V_i: v > p\% \vee v < -p\%\} \quad (4)$$

La idea general de la variable T es señalar periodos de k -días que tienen varios días con precios medios diarios claramente por encima de la variación objetivo. Valores positivos altos de T significan que hay varios precios promedio diarios que son $p\%$ más altos que el precio de cierre de hoy. Tales situaciones son buenas indicaciones de potenciales oportunidades para emitir una orden de compra, ya que tenemos buenas expectativas de que los precios suban. Por otro lado, valores muy negativos de T sugieren acciones de venta, dado que los precios probablemente bajen. Los valores alrededor de cero pueden ser causados por periodos con precios "planos" o por variaciones positivas y negativas conflictivas que se anulan entre sí. (p. 244)

Usaremos el valor T como variable objetivo y trataremos de obtener modelos que estimen este valor usando información del predictor, para luego 'traducir' las

predicciones de nuestro modelo a señales comerciales. Esto significa decidir sobre los umbrales de los valores T predichos que conducirán a cualquiera de las tres posibles acciones comerciales. Realizaremos esta transformación utilizando los siguientes valores:

$$\text{señal} = \begin{cases} \text{sell} & \text{si } T < -0.1 \\ \text{hold} & \text{si } -0.1 \leq T \leq 0.1 \\ \text{buy} & \text{si } T > 0.1 \end{cases} \quad (5)$$

La selección de los valores 0.1 y -0.1 es puramente heurística y también podemos utilizar otros umbrales. Aun así, estos valores significan que durante el período de 10 días utilizado para generar los valores T , hubo al menos cuatro precios diarios promedio que están un 2.5% por encima del cierre actual ($4 \times 0.025 = 0.1$).

Si se decide utilizar otros valores, se debe tener en cuenta que valores absolutos demasiado altos generarán menos señales, mientras que valores demasiado pequeños pueden llevarnos a operar con variaciones demasiado pequeñas del mercado, incurriendo así en un mayor riesgo. (p. 251)

Así, Torgo construye una variable de “señal de trading” que puede tomar tres valores según la tendencia del mercado: “sell” (vender posición) si identifica una tendencia de precios *a la baja*, “hold” (mantener posición) si no identifica tendencia, y “buy” (comprar posición) si identifica una tendencia de precios *en alza*.

Como nuestro objetivo es construir una variable de tendencia de precios en caso de haberla, dejaremos de lado las observaciones para las cuales no se identifique que haya una tendencia de precios, y así estaremos teniendo una variable binaria Y que tome valores “Down” cuando identifique una tendencia de precios *a la baja* y “Up” cuando identifique una tendencia de precios *en alza*:

$$Y_i = \begin{cases} \text{Down} & \text{si } T_i < -t \\ \text{Up} & \text{si } T_i > t \end{cases} \quad (6)$$

De esta manera, las observaciones con variaciones poco sustanciales ($-t \leq T \leq t$) son descartadas al momento de clasificar tendencias.

Volveremos a revisar esta definición de tendencia en la etapa de experimentación, al contrastar otras interpretaciones de tendencias presentadas por otros autores.

2.2.3 Métricas de evaluación

Para evaluar los modelos y poder compararlos utilizaremos dos métricas clásicas para performance de clasificación: accuracy y área bajo la curva ROC (AUC-ROC).

La accuracy es el número de predicciones correctas como una proporción de todas las predicciones hechas. Toma valores entre 0 y 100%, y se busca obtener el mayor valor posible.

$$accuracy = \frac{\text{número de predicciones correctas}}{\text{número total de predicciones}} \quad (7)$$

Esta es una métrica de evaluación común para los problemas de clasificación, y es la utilizada principal y/o únicamente en los trabajos que analizamos, por esta razón la tendremos en cuenta. Sin embargo, pondera a las clases por igual, por lo que se ve afectada por la distribución de las clases, lo cual presenta un problema en casos de distribución de clases desbalanceadas.

Consideramos que esto es un defecto no menor, ya que la predicción del mercado financiero suele ser un tema complejo en el cual no esperamos obtener valores muy elevados de accuracy, simplemente superar al azar (50%) sería considerado un logro en este campo⁶. Es por esto que, aunque nuestros datos no tendrán una distribución de clases muy desbalanceada (rondará entre 50%-50% y 40%-60% para los varios conjuntos de datos que estaremos utilizando), al estar trabajando con valores de accuracy que serán buenos con sólo superar el 50%, podría resultar que un algoritmo resuelva predecir todas las observaciones como la clase mayoritaria para así maximizar esta métrica.

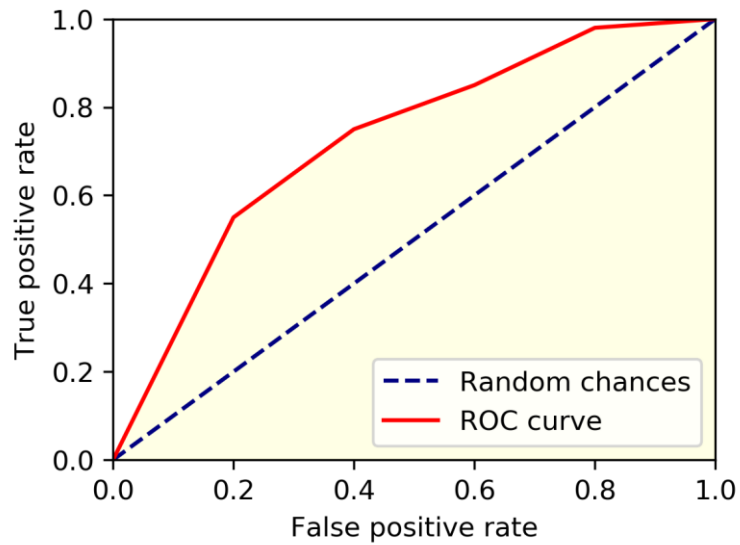
Es por esto que también evaluaremos los modelos según el área bajo la curva ROC (AUC-ROC), otra conocida métrica de performance que ha ido ganando cada vez más adhesión debido a sus propiedades que la hacen especialmente útil para dominios con distribución de clases desiguales.

Como su nombre indica, esta métrica se obtiene calculando el área bajo la curva ROC (acrónimo de Receiver Operating Characteristic, o Característica Operativa del Receptor). En una curva ROC, la tasa de verdaderos positivos (TPR) se representa a lo largo del eje y y la tasa de falsos positivos (FPR) se muestra en el eje x. Cada punto a lo largo de la curva corresponde a los valores de TPR y FPR que se obtienen cuando se modifican los umbrales de decisión.

Un buen modelo de clasificación debe ubicarse lo más cerca posible de la esquina superior izquierda del diagrama, mientras que un modelo que hace conjeturas aleatorias debe residir a lo largo de la diagonal principal, conectando los puntos (TPR=0, FPR=0) y (TPR=1, FPR=1). La predicción aleatoria significa que un registro se clasifica como clase positiva con una probabilidad fija p , independientemente de sus atributos (Tan, Steinbac, Kumar, 2006).

⁶ “Fama y Blume (1966) admiten que en el corto plazo se puede dar la situación en que toda la información disponible no esté incorporada correctamente a los precios, permitiendo que algunos inversores obtengan, en consecuencia, beneficios superiores al promedio del mercado, originado en esta ineficiencia temporal. Sostienen que estas variaciones de corto plazo no pueden predecirse, siendo el beneficio extraordinario, generado producto del azar.” (Quiroga, 2017)

Figura 10: Ejemplo de curva ROC (Fuente: Bui, 2020)



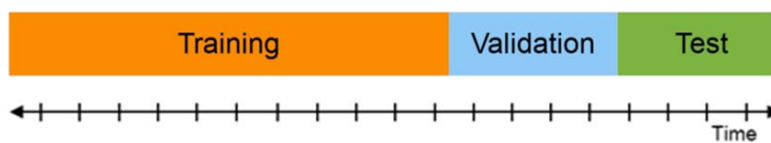
Para obtener un número que nos diga qué tan buena es nuestra curva, calculamos el área bajo la curva ROC, o puntaje ROC-AUC. Cuanto más arriba a la izquierda esté la curva, mayor será el área y, por lo tanto, mayor será la puntuación ROC-AUC.

Bajo esta lógica, un modelo perfecto tendría el área bajo la curva ROC máxima, que es igual a 1. Por el contrario, un modelo que simplemente realiza conjeturas aleatorias, tendría el peor valor de área bajo la curva ROC, que sería 0,5. Un modelo con área bajo la curva por debajo de 0,5 estaría interpretando mal las clases. Se concluye entonces que un modelo estrictamente mejor que otro tendría un área más grande bajo la curva ROC.

2.2.4 Esquema de validación

Como estamos tratando con series de tiempo, respetaremos la temporalidad de los datos al momento de dividirlos en conjuntos de entrenamiento, validación y evaluación. Debido a que utilizaremos diferentes conjuntos de datos según cada experimento, especificaremos la partición de cada uno en sus debidas secciones (dentro de la sección de Experimentación). De momento nos limitaremos a decir que en todos los casos los datos estarán ordenados por fecha, y la partición entre conjuntos se hará respetando el orden cronológico de las observaciones, como se observa en el esquema a continuación.

Figura 11: Esquema de split de conjuntos de datos de series temporales



Resulta de vital importancia hacer una correcta partición de los datos teniendo en cuenta su temporalidad, para cerciorarnos de que no habrá *data leakage* en nuestros modelos y los resultados obtenidos sean válidos/confiables. Kaufman, Rosset y Perlich (2014) exponen en su paper *Leakage in Data Mining: Formulation, Detection, and Avoidance*:

Considerado ‘uno de los diez principales errores de minería de datos’, *leakage* es esencialmente la introducción de información sobre la variable objetivo, que no debería estar legítimamente disponible para entrenar el modelo.

En algunos dominios, como la predicción de series temporales, donde normalmente solo se dispone de una única serie histórica que mide el fenómeno de interés para el análisis, esta forma de *leakage* es endémica y se conoce comúnmente como *data snooping / dredging*.

Modelar con *leakage* es perjudicial en muchos niveles: es una fuente de mala generalización y sobreestimación del rendimiento esperado.

3. Experimentación

En esta sección se desarrollarán tres experimentos. Los dos primeros intentan replicar trabajos mencionados anteriormente en la Introducción, mientras que el tercero va a tratar de abordar la problemática del primer experimento utilizando una metodología diferente:

- Experimento 1: *Identifying Candlestick Patterns using Deep Learning* (Shah, 2020)
- Experimento 2: *Using Deep Learning Neural Networks and Candlestick Chart Representation to Predict Stock Market* (Kusuma, Ho, Kao, Ou, Hua, 2019)
- Experimento 3: Predicción de tendencia de precios de activos financieros mediante el análisis de indicadores técnicos con modelo de Aprendizaje Automático

A continuación, en la subsección de cada experimento se detallan las características y algunas decisiones metodológicas específicas de los mismos.

3.1 Experimento 1: Identifying Candlestick Patterns using Deep Learning

En la página de *Towards Data Science* se presenta un prometedor artículo, *Identifying Candlestick Patterns using Deep Learning* (Shah, 2020), que plantea la posibilidad de armar fácilmente un modelo predictivo que entrena redes neuronales sobre gráficos de vela para predecir tendencias alcistas o bajistas de los precios de los activos financieros, y asegura que obtiene resultados lo suficientemente buenos (62% de accuracy) como para poder reconocer patrones que se dan en caso de cada tendencia.

Dicho experimento utiliza la librería *fast.ai*⁷, que permite una implementación simple y rápida de los modelos de redes neuronales. Adicionalmente, el autor ha puesto a disposición su código, con lo cual pasamos a replicarlo.

Al hacerlo nos encontramos con un error metodológico que comprometería los resultados alcanzados por el autor. En consecuencia, veremos cómo cambian los resultados al arreglar dicho error y evaluaremos varias alternativas para mejorar el modelo y su desempeño.

⁷ <https://www.fast.ai/>

3.1.1 Metodología original

a. Planteo original

Los datos a utilizar son precios diarios históricos de 10 acciones indias de alta capitalización⁸, entre enero de 2018 y septiembre de 2020, extraídos de la Bolsa Nacional de Valores de India mediante la API *nsepy*⁹.

Shah realiza la partición de los datos entre conjuntos de entrenamiento y validación de manera aleatoria, dejando 80% de los datos para entrenamiento y 20% para validación. No dispone de un tercer conjunto de datos para evaluación del modelo. (Retomaremos este punto más adelante)

En este caso, la variable a predecir, Y , es la *tendencia* del precio de una acción. El autor define la lógica de clasificación según la variación de precios de cierre de los próximos 5 días. En particular, la variable binaria Y tomará los siguientes valores:

- “Up”: Si en algún punto dentro de los próximos 5 días la variación de precio es superior a 3.7% del precio de cierre de hoy (por su tendencia alcista)
- “Down”: Si en algún punto dentro de los próximos 5 días la variación de precio es inferior a -3.7% (por su tendencia bajista)

Los casos que presenten variaciones poco significativas y no entren en las categorías ‘Up’ o ‘Down’ se desestiman, es decir se descartan y no cuentan como observaciones válidas para pasarle al modelo.

Se itera por todas las series de precios de todas las acciones, y para cada observación:

- Según el criterio de clasificación descrito anteriormente, se le asigna la clase ‘Up’ o ‘Down’, o se saltea el caso, según corresponda.
- Para las observaciones clasificadas como ‘Up’ o ‘Down’, se produce el gráfico de velas, con el atributo adicional de las barras de volumen de negociación diario, para una ventana de tiempo de 20 días anteriores a dicha observación.

⁸ Lista de acciones disponible en el [Anexo](#).

⁹ <https://nsepy.xyz/>

Figura 12: Experimento 1 - Extracto del código original donde se ejecutan la clasificación y generación de imágenes

```
equities=['BAJFINANCE', 'RELIANCE', 'INFY', 'HDFC', 'HDFCBANK', 'HDFCLIFE', 'ZEEL', 'DELTACORP', 'ITC', 'ASIANPAINT']
for equity in equities :
    df_pricing=obtain_data(equity,date(2018,1,1),date(2020,9,1))
    df=df_pricing.copy()
    df=df.reset_index(drop=True)
    n_days=5
    fraction_movement=0.037
    df['Trend']=None
    for i in range(len(df)):
        try :
            for n in range(n_days):
                if df.loc[i,'close_price'] - df.loc[i+n,'close_price'] >= fraction_movement*df.loc[i,'close_price']:
                    df.loc[i,'Trend'] = 'Down'
                    if i >=20 :
                        fig=plot_candles(df_pricing[i-20:i],volume_bars=True)
                        fig.savefig('Candle Data/Down/{0}{1}.png'.format(df_pricing['Symbol'][i],i),dpi=70)
                        print('Down',i,n)
                        break
                elif df.loc[i+n,'close_price'] - df.loc[i,'close_price'] >= fraction_movement*df.loc[i,'close_price'] :
                    df.loc[i,'Trend']='Up'
                    if i >20 :
                        fig=plot_candles(df_pricing[i-20:i],volume_bars=True)
                        fig.savefig('Candle Data/Up/{0}{1}.png'.format(df_pricing['Symbol'][i],i),dpi=70)
                        print('Up',i,n)
                        break
                else :
                    df.loc[i,'Trend']= 'No Trend'
            except :
                print(i)
                pass
```

Se aborda el problema de predicción con el algoritmo Resnet34, una red neuronal convolucional de 34 capas que se suele utilizar como un modelo de clasificación de imágenes de última generación (Solawetz, 2020). Este es un modelo que se ha entrenado previamente en el conjunto de datos de ImageNet¹⁰, una base de datos que tiene más de 14 millones de imágenes pertenecientes a más de 20.000 clases diferentes. Otra particularidad de esta red es que toma residuos de cada capa y los usa en las capas conectadas posteriores. Según el autor, con dicho modelo se alcanza una accuracy de cerca de 62% en el conjunto de validación.

En el artículo se evalúa el desempeño del modelo únicamente según la accuracy. Sin embargo, como ya mencionamos en la metodología, también resulta interesante tener presente la distribución de las clases en los conjuntos de datos y la capacidad del modelo para distinguir ambas clases. Es por esto que adicionalmente estaremos evaluando el área bajo la curva ROC en nuestro análisis.

Como paso adicional, el autor juega con las activaciones de redes en las imágenes bien clasificadas por el modelo, para encontrar patrones en ambas clases de imágenes. Esto lo reproduce a través de la generación de *heatmaps*. A efectos del análisis de este trabajo, no nos adentraremos en esta parte de su código.

¹⁰ <https://www.image-net.org>

b. Replicación de resultados

Como punto de inicio de nuestra investigación, procedemos a replicar el trabajo realizado en el código del artículo recién descrito para comprobar que se alcanzan los resultados afirmados por Shah en validación (ya que no utiliza datos para testing). Descargamos los datos especificados y entrenamos el modelo, obteniendo los resultados expuestos a continuación.

Resultados preliminares

Tabla 2: Experimento 1 - Resultados con metodología original

Modelo	Original
train_class_distribution	
Down	46%
Up	54%
valid_class_distribution	
Down	56%
Up	44%
epochs	10
train_loss	0.3
valid_loss	0.7
Resultados Validación	
Accuracy	69%
AUC-ROC	74%

Figura 13: Experimento 1 - AUC-ROC con metodología original

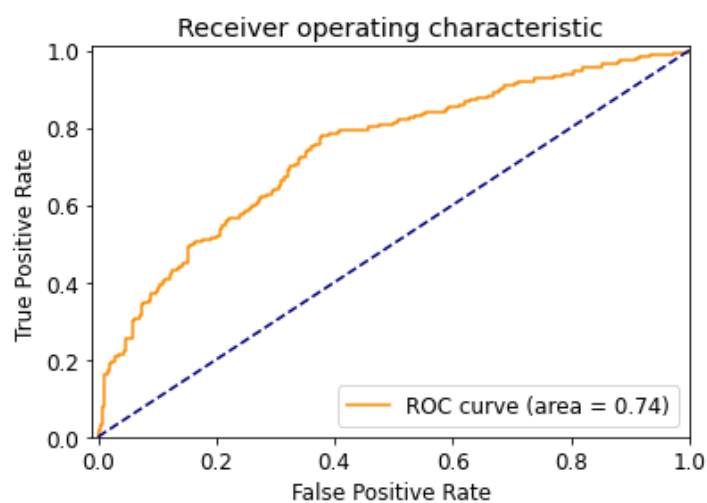
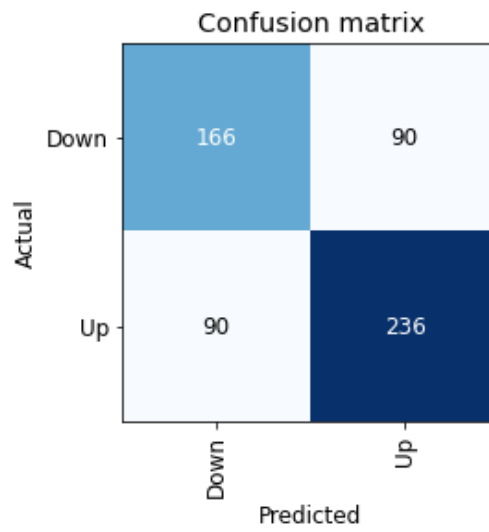


Figura 14: Experimento 1 - Matriz de confusión con metodología original



Obtenemos resultados parecidos a los indicados en el artículo, donde se menciona una accuracy del 62% mientras que en esta corrida obtenemos 69%. También observamos que las clases tienen una distribución bastante pareja, y se obtiene un 74% en la métrica AUC-ROC.

c. Error

Como mencionamos anteriormente, al inspeccionar el código que entrena el modelo predictivo, nos encontramos con que los datos se están dividiendo aleatoriamente en conjuntos de entrenamiento y validación, lo cual implica que se está filtrando información del futuro al entrenamiento del modelo, y por lo tanto el mismo estaría sobreestimando los resultados. Esto anula por completo la validez del modelo planteado y los resultados alcanzados.

Figura 15: Extracto del código original donde se realiza la partición de datos aleatoria

```
np.random.seed(42)
data = ImageDataBunch.from_folder(path, train=".", valid_pct=0.2,
    ds_tfms=get_transforms(flip_vert=False,
        max_lighting=0.1,
        max_zoom=1.05,
        max_warp=0.,
        max_rotate=3),
    size=224, num_workers=4
).normalize(imagenet_stats)
```

3.1.2 Metodología corregida

Para solucionar esta situación, tomamos los precios diarios históricos de las 10 acciones indias entre enero de 2018 y septiembre de 2020, y procedemos a hacer una partición adecuada del conjunto de datos de la siguiente manera:

Tabla 3: Experimento 1 - Partición de datos corregida

Dataset	Fechas
entrenamiento	01/2018 - 02/2020
validación	03/2020 - 05/2020
evaluación	06/2020 - 08/2020

Cabe mencionar que no solo realizamos la adecuada división de los datos teniendo en cuenta el eje temporal, sino también que al generar los gráficos evitamos que se transfiera información de un período al otro entre conjuntos de datos (por ejemplo, en el training set no

se generan gráficos para observaciones cuya tendencia se define según días futuros que se encuentran en el validation set).

Una vez que tenemos la data inicial dividida correctamente entre los conjuntos de datos, clasificamos y generamos nuevamente las observaciones (gráficos/imágenes) para alimentar al modelo predictivo.

Resultados corregidos

Con la partición de datos correcta, se obtienen realmente los siguientes resultados:

Tabla 4: Experimento 1 - Resultados con metodología corregida

Modelo	Original c/split correcto
train_class_distribution	
Down	47%
Up	53%
valid_class_distribution	
Down	36%
Up	64%
test_class_distribution	
Down	34%
Up	66%
epochs	10
train_loss	0.26
valid_loss	1.25
Resultados Validación	
Accuracy	50%
AUC-ROC	57%
Resultados Testing	
Accuracy	53%
AUC-ROC	54%

Figura 16: Experimento 1 - AUC-ROC con metodología corregida

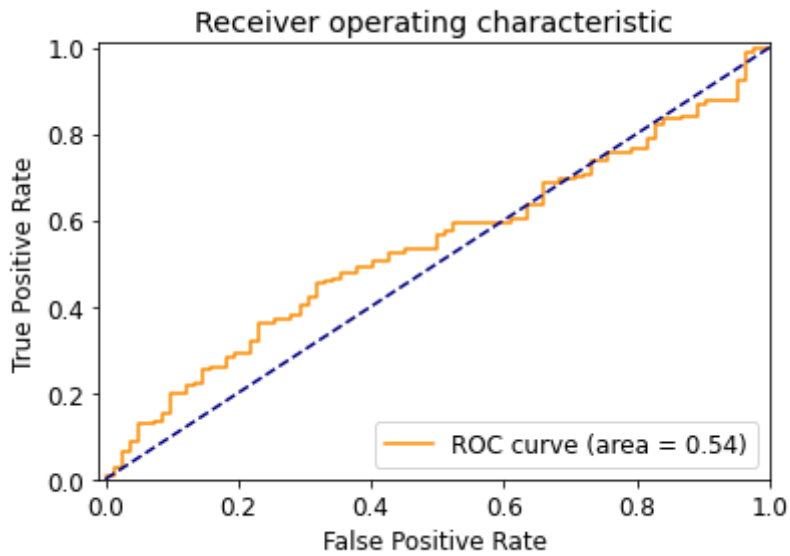
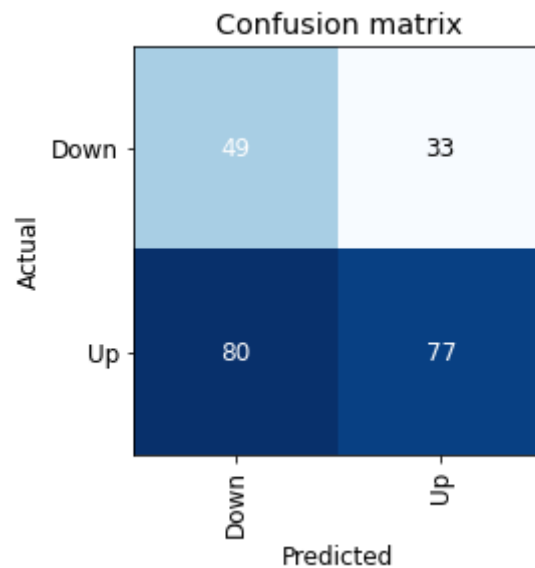


Figura 17: Experimento 1 - Matriz de confusión con metodología corregida



Queda expuesto así que el real desempeño de este modelo es efectivamente bastante más pobre, y se refleja claramente en la matriz de confusión la poca precisión de sus predicciones, la cual desciende a 50% en validación (donde Shah reportó un 62%) y 53% en test. El área bajo la curva ROC desciende a su vez a 57% en validación y 54% en test.

Se puede concluir que gracias a un error conceptual no menor, un artículo publicado en un blog de ciencia de datos falsamente declara que es posible generar un modelo predictivo razonablemente fiable usando redes neuronales para analizar gráficos de velas. Sin embargo dicho modelo, al ser formulado correctamente, predice prácticamente al azar (lo cual tiene bastante más sentido que el elevado resultado adjudicado).

3.1.3 Metodología alternativa

Al resolver el error de la partición de datos en conjuntos de entrenamiento y validación, el modelo no obtiene resultados satisfactorios, y predice al igual que el azar. Ahora nos encontramos en un punto que es realista para un problema de negocios en este campo. Con un buen esquema de validación, tomamos esto como punto de partida para experimentar con diferentes variantes de la metodología buscando mejorar el desempeño del modelo.

A continuación se describen algunas posibles modificaciones y sus resultados.

a. Método de Clasificación

Como ya vimos en la sección anterior, parte de la complejidad de este problema está ligada a la definición de la variable objetivo, y en consecuencia la performance del modelo también.

Consideramos la posibilidad de que tal vez haya una manera alternativa para construir la variable a predecir, en este caso la *tendencia* del precio de un activo financiero, que resulte mejor en términos del desempeño del modelo. Revisamos entonces la clasificación que planteamos en la sección de Metodología, basada en la teoría de Luís Torgo (2017).

Aproximamos el precio diario promedio como:

$$\bar{P}_i = \frac{C_i + H_i + L_i}{3} \quad (8)$$

donde C_i , H_i , L_i son los precios de cierre, máximo y mínimo para el día i , respectivamente.

Sea V_i el conjunto de k variaciones porcentuales del precio de cierre de hoy contra los precios promedio de los siguientes k días:

$$V_i = \left\{ \frac{P_{i+j} - C_i}{C_i} \right\}_{j=1}^k \quad (9)$$

Nuestra variable indicadora es la suma total de las variaciones de este conjunto cuyo valor absoluto está por encima de nuestro margen objetivo $p\%$:

$$T_i = \sum \{v \in V_i: v > p\% \vee v < -p\%\} \quad (10)$$

Así, obtenemos nuestra variable a predecir Y :

$$Y_i = \begin{cases} Down & si T_i < -t \\ Up & si T_i > t \end{cases} \quad (11)$$

De esta manera, como veníamos viendo hasta el momento, las observaciones con variaciones poco sustanciales ($-t \leq T \leq t$) son descartadas al momento de clasificar tendencias.

Experimentación: Clasificación Alternativa (k = 10, p = 0.025, t = 0.1)

Procedemos a modificar el código original siguiendo este método alternativo de definición de la variable objetivo y clasificación de los gráficos de velas, utilizando los valores planteados por Luis Torgo (2017):

$$k = 10, p = 0.025, t = 0.1$$

$$Y_i = \begin{cases} Down & si T_i < -0.1 \\ Up & si T_i > 0.1 \end{cases} \quad (12)$$

Al entrenar nuevamente el modelo de redes planteado por Shah - con la correcta partición de los conjuntos de datos de entrenamiento y validación, y la nueva clasificación -, se obtienen los siguientes resultados:

Tabla 5: Experimento 1 - Resultados con clasificación alternativa ($k = 10$, $p = 0.025$, $t = 0.1$)

Modelo	Clasificación alternativa ($k=10$; $p=0.025$; $t=0.1$)
train_class_distribution	
Down	44%
Up	56%
valid_class_distribution	
Down	39%
Up	61%
test_class_distribution	
Down	30%
Up	70%
epochs	10
train_loss	0.2
valid_loss	1.29
Resultados Validación	
Accuracy	50%
AUC-ROC	50%
Resultados Testing	
Accuracy	48%
AUC-ROC	51%

Figura 18: Experimento 1 - AUC-ROC con clasificación alternativa ($k = 10$, $p = 0.025$, $t = 0.1$)

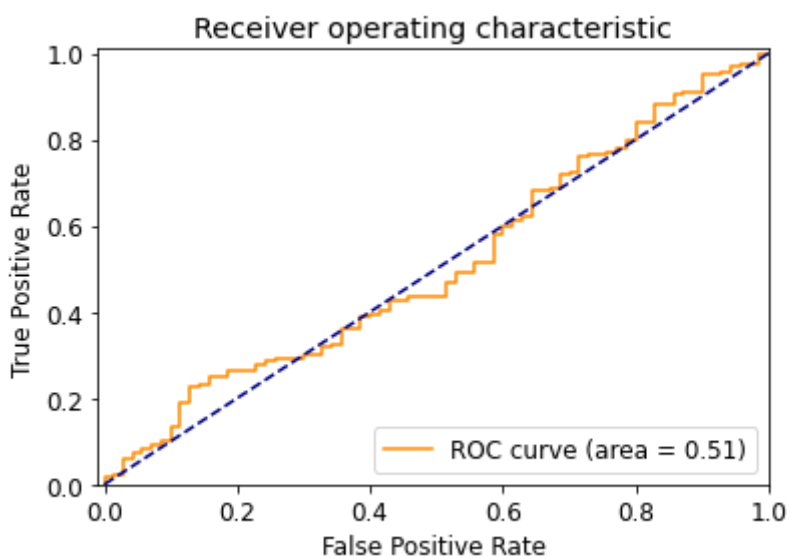
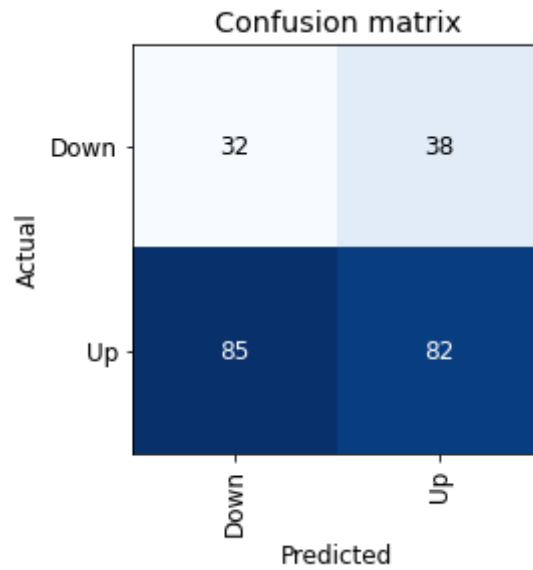


Figura 19: Experimento 1 - Matriz de confusión con clasificación alternativa ($k = 10, p = 0.025, t = 0.1$)



Los resultados alcanzados no parecen mejorar con el nuevo método de clasificación aplicado, obteniendo una accuracy de 48% y AUC-ROC de 51%. El modelo no logra superar al azar, e incluso consigue un peor desempeño según la métrica AUC-ROC.

Experimentación: Clasificación Alternativa ($k = 5, p = 0.025, t = 0.05$)

Antes de descartar este enfoque, consideramos probarlo nuevamente con diferentes valores a los utilizados por Luis Torgo (2017). Tomando en consideración que el período futuro estudiado en el artículo de Shah es de 5 días, adaptamos los valores para definir la tendencia en período k más corto, y por consiguiente ajustamos la variable t también:

$$k = 5, p = 0.025, t = 0.05$$

$$Y_i = \begin{cases} Down & \text{si } T_i < -0.05 \\ Up & \text{si } T_i > 0.05 \end{cases} \quad (13)$$

Al entrenar el modelo con los nuevos parámetros se obtienen los siguientes resultados:

Tabla 6: Experimento 1 - Resultados con clasificación alternativa ($k = 5$, $p = 0.025$, $t = 0.05$)

Modelo	Clasificación alternativa ($k=5$; $p=0.025$; $t=0.05$)
train_class_distribution	
Down	47%
Up	53%
valid_class_distribution	
Down	39%
Up	61%
test_class_distribution	
Down	28%
Up	72%
epochs	10
train_loss	0.26
valid_loss	1.29
Resultados Validación	
Accuracy	47%
AUC-ROC	53%
Resultados Testing	
Accuracy	50%
AUC-ROC	52%

Figura 20: Experimento 1 - AUC-ROC con clasificación alternativa ($k = 5$, $p = 0.025$, $t = 0.05$)

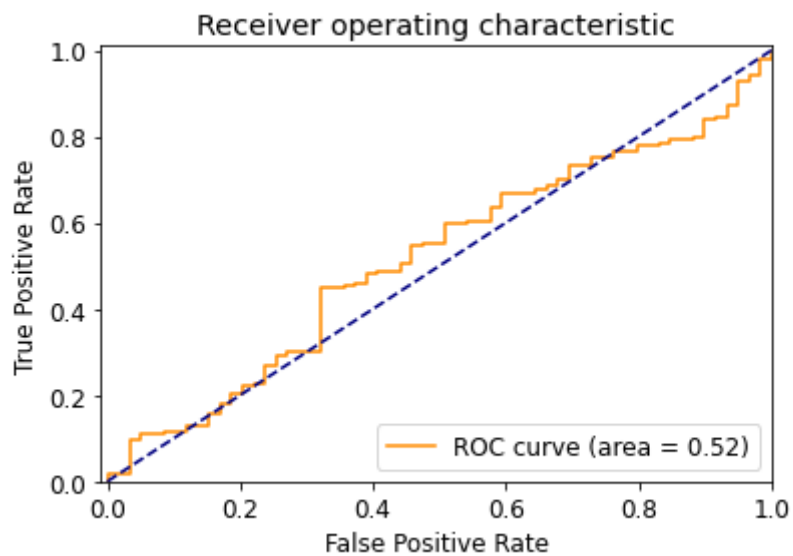
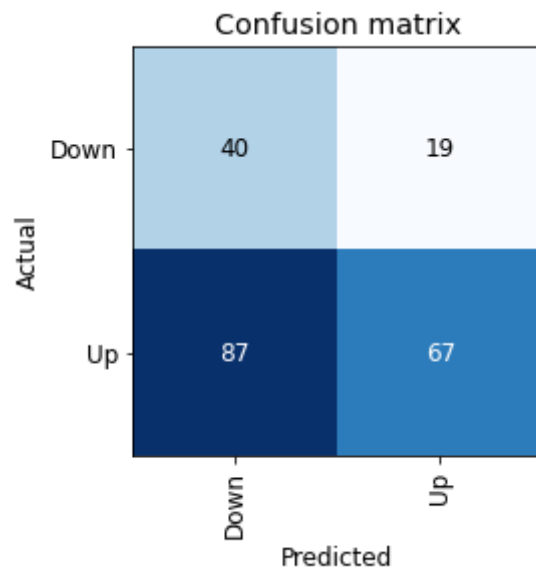


Figura 21: Experimento 1 - Matriz de confusión con clasificación alternativa ($k = 5$, $p = 0.025$, $t = 0.05$)



Pareciera haber una leve mejora en ambas métricas, aunque aún tenemos un modelo equivalente al azar. El desempeño del modelo aún no resulta satisfactorio con las modificaciones experimentadas en cuanto al criterio de clasificación, por lo que evaluaremos otros enfoques para mejorar su performance.

b. Generación de Imágenes

En el análisis técnico se estudian diferentes herramientas e indicadores. En principio el artículo se centra en analizar gráficos de velas con el adicional de las barras de volumen. Consideramos que este es un buen punto de inicio, pero a su vez esto podría no estar aportando suficiente información al modelo predictivo para que el mismo pueda llegar a buenos resultados, por lo que parecería conveniente explorar la posibilidad de agregar más información a las imágenes mediante algún indicador adicional.

A lo largo de esta etapa de experimentación ('Generación de imágenes'), volveremos a utilizar la construcción de la variable Y originalmente utilizada en el artículo, dejando de lado las alternativas planteadas en base al trabajo de Torgo (2017).

Medias Móviles

En su libro *Technical Analysis of the Financial Markets* (1999), Murphy destaca una métrica que resulta particularmente interesante para sumar a nuestro modelo: la media móvil. Expone su atractivo de la siguiente manera:

La media móvil es uno de los indicadores técnicos más versátiles y ampliamente utilizados. Debido a la forma en que está construida y al hecho de que puede cuantificarse y probarse con tanta facilidad, es la base de muchos sistemas mecánicos de seguimiento de tendencias que se utilizan en la actualidad.

El análisis chartista es en gran parte subjetivo y difícil de probar. Como resultado, el mismo no se presta tan bien a la informatización. Las reglas de la media móvil, por el contrario, se pueden programar fácilmente en una computadora, que luego genera señales específicas de compra y venta. Si bien dos técnicos pueden estar en desacuerdo sobre si un patrón de precio dado es un triángulo o una cuña, o si el patrón de volumen favorece al lado alcista o bajista, las señales de tendencia de la media móvil son precisas y no están abiertas a debate.

Propósito

Las fluctuaciones diarias en los precios de las acciones, los precios de las materias primas y las tasas de cambio pueden ser grandes. Los promedios móviles atenúan estas fluctuaciones, quitando énfasis pero a veces distorsionando las fluctuaciones. Los analistas técnicos utilizan medias móviles para suavizar los datos erráticos, lo que facilita ver la verdadera tendencia subyacente. (Kirkpatrick II, Dahlquist, 2010)

Sin embargo, la media móvil no predice la acción del mercado en el mismo sentido que intenta hacerlo el análisis de gráficos estándar. Es un seguidor, no un líder. Nunca anticipa; solo

reacciona. La media móvil sigue un mercado y nos dice que ha comenzado una tendencia, pero sólo después del hecho. (Murphy, 1999)

Queda entonces clara la utilidad de dicha métrica ya que estaría aportando información nueva y diferente a nuestro modelo, complementando perfectamente a la que aportan las velas, por lo que podría ser una incorporación enriquecedora para nuestro análisis.

Cálculo

La media móvil simple, o la media aritmética, es el tipo utilizado por la mayoría de los analistas técnicos. Como implica la primera palabra, es un promedio de cierto cuerpo de datos. Por ejemplo, si se desea un promedio de 10 días de los precios de cierre, se suman los precios de los últimos 10 días y el total se divide por 10. Se usa el término móvil porque solo se usan los precios de los últimos 10 días en el cálculo. Por lo tanto, el cuerpo de datos a promediar (los últimos 10 precios de cierre) avanza con cada nuevo día. (Murphy, 1999)

Determinación del período

Kirkpatrick II y Dahlquist (2010) explican la importancia de la selección del período a utilizar:

Debido a que las medias móviles se pueden calcular para varios períodos de tiempo, ¿cuál período conviene usar? Por supuesto, un período de tiempo más largo incluye más observaciones de datos y, por lo tanto, más información. Al incluir más datos en el cálculo del promedio móvil, los datos de cada día se vuelven relativamente menos importantes en el cálculo. Por lo tanto, un gran cambio en el valor de un día no tiene un gran impacto en la media móvil más larga. Esto puede ser una ventaja si este gran cambio es un valor atípico irregular de un día particular en los datos.

Sin embargo, si este gran movimiento representa el comienzo de un cambio significativo en la tendencia, toma más tiempo hasta que el cambio de tendencia subyacente sea perceptible. Por lo tanto, la media móvil más larga es más lenta para detectar cambios de tendencia, pero es menos probable que indique falsamente un cambio de tendencia debido a una irregularidad a corto plazo en los datos.

Así que hay una cuestión de costo-beneficio aquí. El truco consiste en encontrar la media que sea lo suficientemente sensible para generar señales tempranas, pero lo suficientemente insensible para evitar la mayor parte del ruido aleatorio. (Murphy, 1999)

Experimentación: Media móvil simple de 9 días

Habiendo considerado las cuestiones teóricas mencionadas previamente, probaremos incorporar una media móvil simple a nuestro modelo predictivo para ver si aporta la información adicional necesaria para mejorar el desempeño del mismo.

La literatura no especifica un número de días que sea comúnmente utilizado para medias móviles simples, sino que el número elegido por cada analista técnico puede variar bastante. Por otro lado, esto sí sucede en el caso de utilizar múltiples medias móviles, y también en cuanto al uso de Bandas de Bollinger hay un número popularmente usado, como veremos más adelante. Es por esto que nos decidimos por estudiar medias móviles simples de dos períodos a corto plazo suficientemente diferentes entre sí: de 9 días y de 20 días, ya que más adelante estos números se volverán a repetir cuando veamos los siguientes casos.

Procedemos entonces a producir nuevamente todos los gráficos que serán alimentados en formato de imágenes al modelo predictivo, con el adicional de la media móvil simple de 9 días. Veremos que la cantidad de imágenes varía levemente, esto se debe a que con el cálculo de la media móvil algunas observaciones pueden quedar descartadas por no haber suficientes observaciones previas en el set de datos para poder construir e incorporar la media (en este caso se suman 9 días a los 20 anteriormente requeridos para iniciar a graficar la ventana de 20 días de una observación).

Una vez obtenidos los nuevos datos, entrenamos el modelo predictivo y obtenemos los resultados expuestos a continuación.

Tabla 7: Experimento 1 - Resultados con media móvil simple de 9 días

Modelo	Media móvil de 9 días
train_class_distribution	
Down	47%
Up	53%
valid_class_distribution	
Down	39%
Up	61%
test_class_distribution	
Down	39%
Up	61%
epochs	10
train_loss	0.25
valid_loss	1.48
Resultados Validación	
Accuracy	46%
AUC-ROC	45%
Resultados Testing	
Accuracy	57%
AUC-ROC	56%

Figura 22: Experimento 1 - AUC-ROC con media móvil simple de 9 días

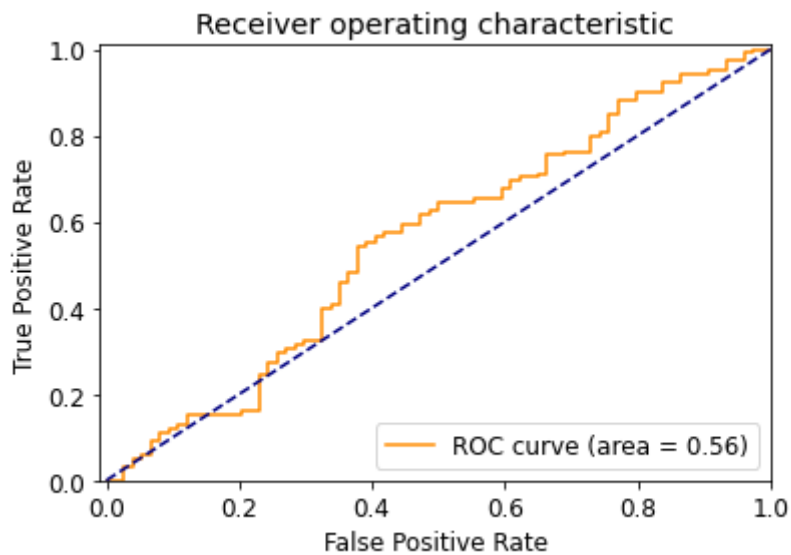
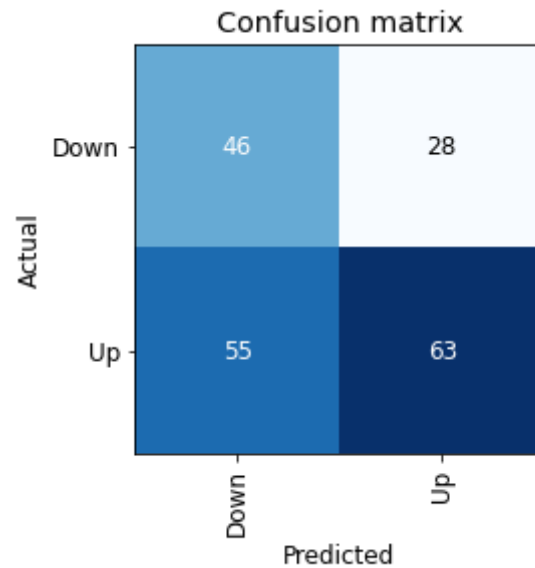


Figura 23: Experimento 1 - Matriz de confusión con media móvil simple de 9 días



Obtenemos resultados interesantes, por un lado los resultados en validación por debajo del 50% para ambas métricas, mientras que al evaluar el modelo con los datos de testing el modelo tiene un desempeño bastante más elevado.

No queda claro si el modelo está aprendiendo efectivamente con la información adicional incorporada, por lo que exploraremos otras opciones en busca de resultados más confiables.

Experimentación: Media móvil simple de 20 días

Probamos si con un período de 20 días la media móvil aporta información más clara al modelo predictivo. Producimos nuevamente todos los gráficos con el adicional de la media móvil simple de 20 días. Nuevamente, la cantidad de imágenes varía levemente, ya que en este caso se suman 20 días a los 20 anteriormente requeridos para iniciar a graficar la ventana de 20 días de una observación (40 días previos requeridos en total para graficar una observación).

Una vez obtenidos los nuevos datos, entrenamos el modelo predictivo y obtenemos los resultados expuestos a continuación.

Tabla 8: Experimento 1 - Resultados con media móvil simple de 20 días

Modelo	Media móvil de 20 días
train_class_distribution	
Down	47%
Up	53%
valid_class_distribution	
Down	40%
Up	60%
test_class_distribution	
Down	38%
Up	62%
epochs	10
train_loss	0.22
valid_loss	1.36
Resultados Validación	
Accuracy	48%
AUC-ROC	51%
Resultados Testing	
Accuracy	57%
AUC-ROC	61%

Figura 24: Experimento 1 - AUC-ROC con media móvil simple de 20 días

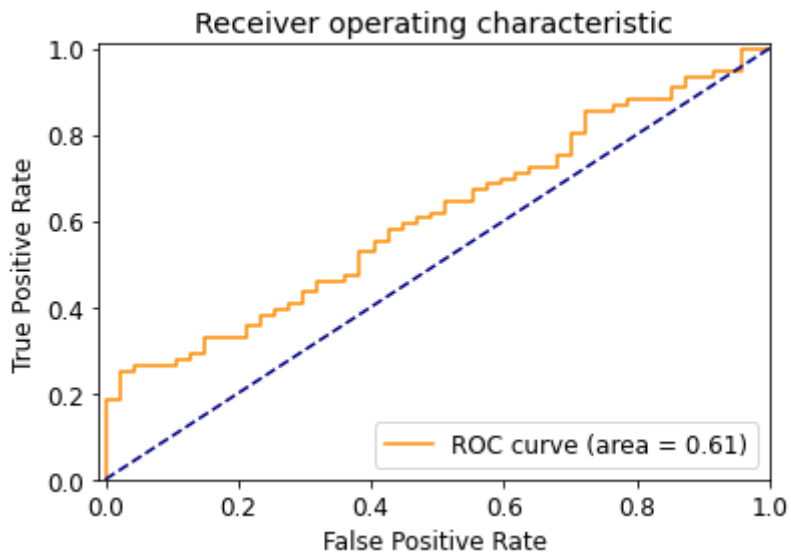
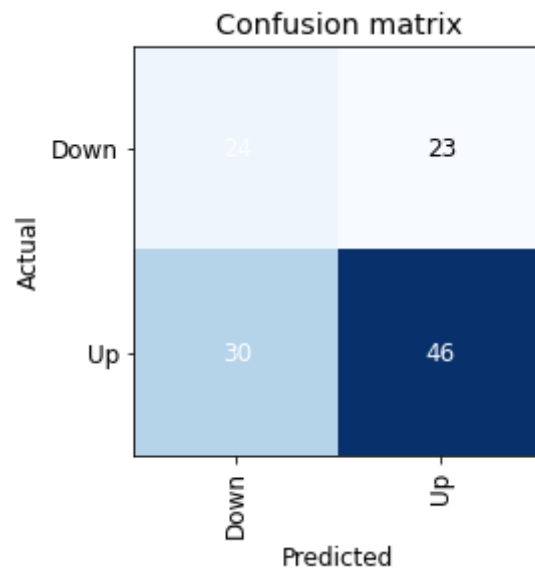


Figura 25: Experimento 1 - Matriz de confusión con media móvil simple de 20 días



El período de 20 días pareciera aportar información un poco más clara al modelo, ya que el mejoran los resultados tanto en validación como testing, aunque sigue habiendo una brecha entre los resultados obtenidos en ambas instancias.

Múltiples medias móviles

El análisis no se limita a la información proporcionada por una sola media móvil. La consideración simultánea de varias medias móviles de varias longitudes puede aumentar el conjunto de información de los analistas.

Un nivel de soporte¹¹ o resistencia¹² a menudo ocurre donde se cruzan dos medias móviles. Cuando el promedio móvil más corto cruza por encima del más largo, a menudo se toma como una señal de compra mecánica, o al menos una señal de que la tendencia del precio es al alza. Asimismo, se considera una señal de venta cuando el más corto cae por debajo del más largo. Muchas estrategias exitosas de medias móviles usan las medias móviles como principal determinante de la tendencia y luego usan medias móviles a más corto plazo, ya sea como puntos finales o como señales. En algunos casos, las medias móviles se usan para determinar la tendencia y luego los patrones de gráficos se usan como señales de entrada y salida. (Kirkpatrick II, Dahlquist, 2010)

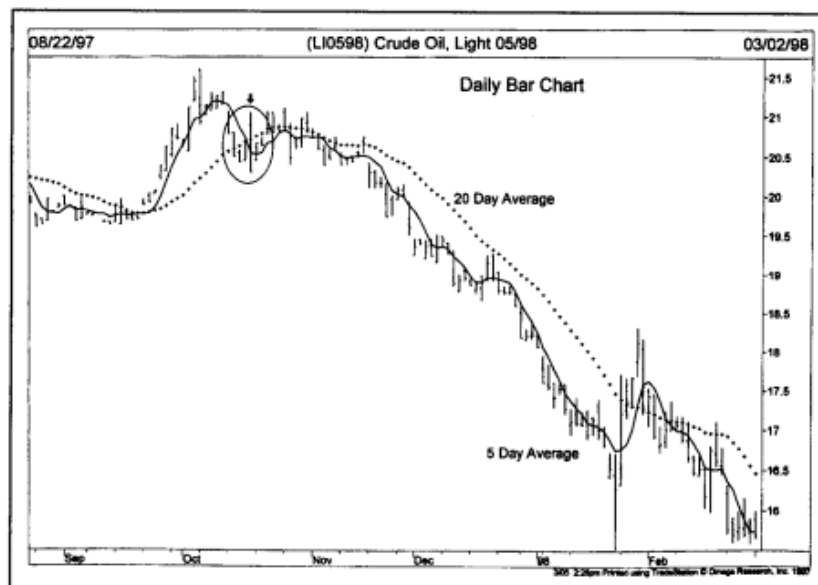
¹¹ “Soporte: nivel en el que el precio frena su descenso para volver a subir.” (Blanco Garzón, 2022)

¹² “Resistencia: nivel en el que el precio deja de subir para comenzar con su descenso.” (Blanco Garzón, 2022)

Uso de dos promedios, o el método de doble cruce

Esta técnica se llama el método de doble cruce. Esto significa que se produce una señal de compra cuando el promedio más corto cruza por encima del más largo. Por ejemplo, dos combinaciones populares son los promedios de 5 y 20 días y los promedios de 10 y 50 días. En el primero, se produce una señal de compra cuando el promedio de 5 días cruza por encima de 20 y una señal de venta cuando el 5 día se mueve por debajo de 20. [...] Esta técnica de usar dos promedios juntos retrasa el mercado un poco más que el uso de un solo promedio, pero produce menos señales falsas. (Murphy, 1999)

Figura 26: Método de doble cruce (Fuente: Murphy, 1999)



Experimentación: Medias móviles de 5 y 20 días

La combinación de medias móviles parece una mejor alternativa que utilizar una simple. Esperando que esto aporte la información adicional necesaria al modelo, procedemos a poner en práctica el método de doble cruce con la popular combinación de medias móviles de 5 y 20 días.

Nuevamente generamos las nuevas imágenes para alimentar el modelo, y se sostiene aquí también la leve variación de observaciones debido al armado de los gráficos con observaciones para las que se cuente con información de hasta 20+20 días previos en el conjunto de datos correspondiente (20 por la ventana del gráfico, y otros 20 por el cálculo de la media móvil más larga, en total 40 días).

Una vez obtenidos los nuevos datos, entrenamos el modelo predictivo y obtenemos los resultados expuestos a continuación.

Tabla 9: Experimento 1 - Resultados con medias móviles de 5 y 20 días

Modelo	Medias móviles de 5 y 20 días
train_class_distribution	
Down	47%
Up	53%
valid_class_distribution	
Down	40%
Up	60%
test_class_distribution	
Down	38%
Up	62%
epochs	10
train_loss	0.22
valid_loss	1.38
Resultados Validación	
Accuracy	53%
AUC-ROC	57%
Resultados Testing	
Accuracy	51%
AUC-ROC	51%

Figura 27: Experimento 1 - AUC-ROC con medias móviles de 5 y 20 días

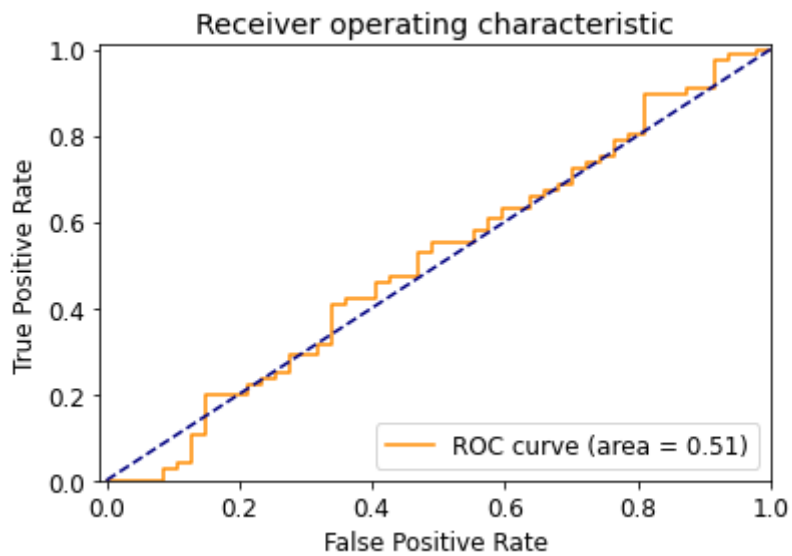
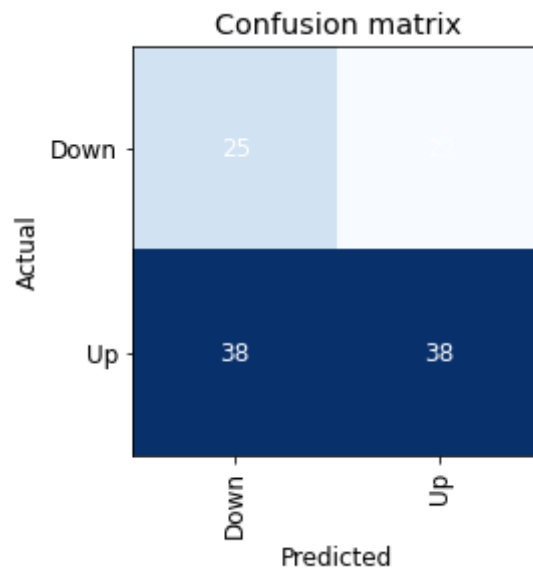


Figura 28: Experimento 1 - Matriz de confusión con medias móviles de 5 y 20 días



Con la incorporación de medias móviles de 5 y 20 días, obtenemos una mejoría en ambas métricas en la etapa de validación pero al evaluar el modelo con el conjunto de datos de testing, ambas descienden a 51% y quedamos nuevamente cercanos al azar.

Procedemos a ver qué información adicional podemos incorporar al modelo para ayudarlo a predecir aún mejor.

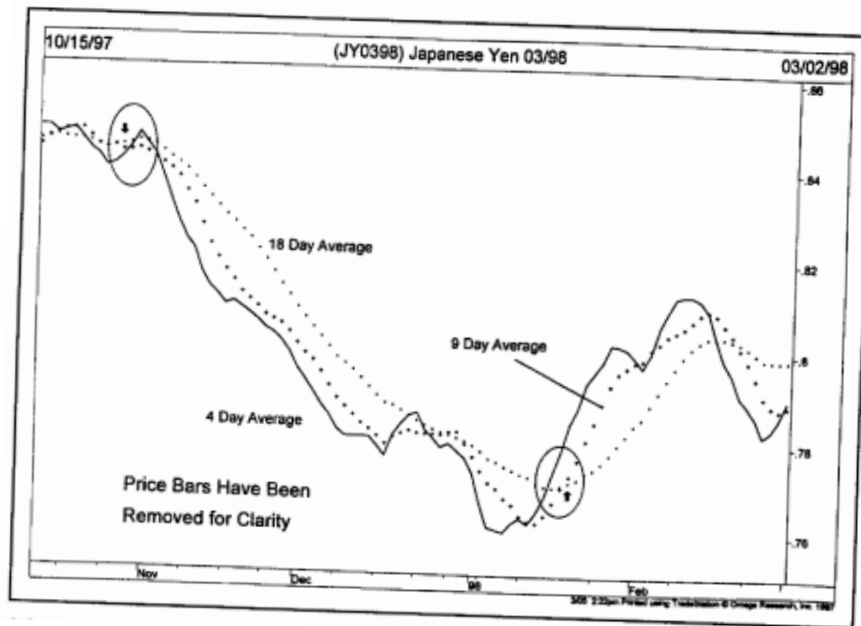
Uso de tres promedios, o el método de cruce triple

Kirkpatrick II y Dahlquist (2010) decretan que el sistema de cruce triple más utilizado es la popular combinación de promedio móvil de 4-9-18 días.

Murphy (1999) explica cómo utilizar el sistema de medias móviles de 4-9-18 días de la siguiente manera:

Ya se ha explicado que cuanto más corta es la media móvil, más se acerca a la tendencia del precio. Entonces, es lógico pensar que el más corto de los tres promedios - el de 4 días - seguirá la tendencia más de cerca, seguido por el de 9 días y luego el de 18. En una tendencia alcista, por lo tanto, la alineación adecuada sería para el promedio de 4 días estar por encima del día 9, que está por encima del promedio de 18 días. En una tendencia bajista, el orden se invierte y la alineación es exactamente la opuesta. Es decir, el día 4 sería el más bajo, seguido del día 9 y luego el promedio de 18 días. (Ver figura 29)

Figura 29: Método de triple cruce con medias móviles de 4, 9 y 18 días (Fuente: Murphy, 1999)



Se produce una alerta de compra en una tendencia bajista cuando el día 4 cruza por encima del 9 y el 18. Se produce una señal de compra confirmada cuando el día 9 cruza por encima del 18. Esto sitúa al día 4 sobre el día 9, que está por encima del día 18. Puede ocurrir cierta mezcla durante las correcciones o consolidaciones, pero la tendencia alcista general permanece intacta. Algunos traders pueden obtener ganancias durante el proceso de mezcla y algunos pueden usarlo como una oportunidad de compra. Obviamente, hay mucho espacio para la flexibilidad aquí en la aplicación de las reglas, dependiendo de la agresividad con la que uno quiera operar.

Cuando la tendencia alcista se revierte a la baja, lo primero que debe ocurrir es que el promedio más corto (y más sensible) - el de 4 días - cae por debajo del día 9 y del día 18. Esto es solo una alerta de venta. Algunos traders, sin embargo, podrían usar ese lanzamiento inicial como razón suficiente para comenzar a liquidar posiciones largas. Luego, si el siguiente promedio más largo, el día 9, cae por debajo del día 18, se da una señal de venta corta confirmada.

Experimentación: Medias móviles de 4, 9 y 18 días

Ponemos en práctica ahora el método de triple cruce con la popular combinación de medias móviles de 4, 9 y 18 días esperando pasarle más y mejor información al modelo predictivo.

Nuevamente generamos las nuevas imágenes para alimentar el modelo, y se sostiene aquí también la leve variación de observaciones debido al armado de los gráficos con observaciones para las que se cuente con información de hasta 20+18 días previos en el

conjunto de datos correspondiente (20 por la ventana del gráfico, y otros 18 por el cálculo de la media móvil más larga, en total 38 días).

Una vez obtenidos los nuevos datos, entrenamos el modelo predictivo y obtenemos los resultados expuestos a continuación.

Tabla 10: Experimento 1 - Resultados con medias móviles de 4, 9 y 18 días

Modelo	Medias móviles de 4, 9 y 18 días
train_class_distribution	
Down	47%
Up	53%
valid_class_distribution	
Down	41%
Up	59%
test_class_distribution	
Down	39%
Up	61%
epochs	10
train_loss	0.23
valid_loss	1.77
Resultados Validación	
Accuracy	43%
AUC-ROC	44%
Resultados Testing	
Accuracy	46%
AUC-ROC	51%

Figura 30: Experimento 1 - AUC-ROC con medias móviles de 4, 9 y 18 días

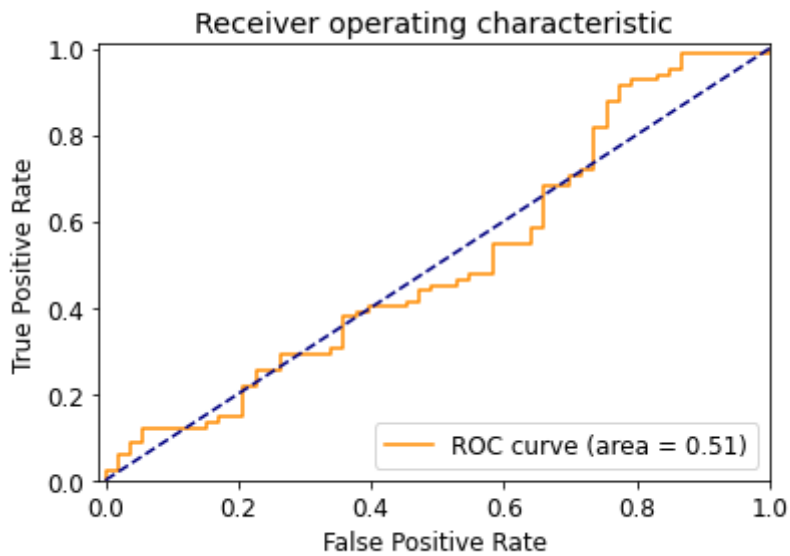
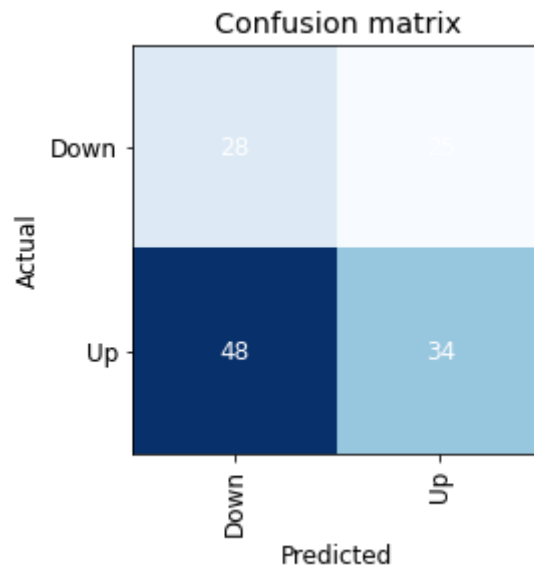


Figura 31: Experimento 1 - Matriz de confusión con medias móviles de 4, 9 y 18 días



Los resultados obtenidos son peores que los alcanzados anteriormente con el modelo de medias de 5 y 20 días. Pareciera ser que la nueva información que incorporamos con la tercera media confundió más al modelo.

Procedemos a evaluar otras opciones para mejorar la incorporación de medias móviles. En particular, nos interesa lo que sugiere John Murphy (1999): La utilidad de una única media móvil puede mejorarse rodeándola de envolventes.

Canales envolventes y bandas

La media móvil simple representa el centro de la tendencia del precio de una acción, y los precios reales tienden a oscilar alrededor de ese promedio móvil. El movimiento del precio se centra en la media móvil pero cae dentro de una banda o canal envolvente alrededor de la media móvil. Al determinar la banda dentro de la cual los precios tienden a oscilar, el analista puede determinar mejor el rango en el que se espera que fluctúe el precio. (Kirkpatrick II, Dahlquist, 2010)

Envolventes porcentuales

Los sobres de porcentaje se pueden usar para ayudar a determinar cuándo un mercado se ha extendido demasiado en cualquier dirección. En otras palabras, nos dicen cuándo los precios se han desviado demasiado de su línea de media móvil. Para ello, los sobres se colocan en porcentajes fijos por encima y por debajo de la media. (Murphy, 1999)

Figura 32: Ejemplo de envolventes porcentuales (Fuente: Kirkpatrick II, Dahlquist, 2010)



Uno de los principales problemas con los sobres de porcentaje fijo es que no tienen en cuenta la volatilidad cambiante del precio subyacente. Durante una tendencia lateral, cuando la volatilidad suele disminuir, la acción del precio puede estar contenida dentro de una banda relativamente estrecha. Sin embargo, cuando comienza la tendencia, la volatilidad a menudo se expande y luego creará señales falsas utilizando un sobre de porcentaje fijo. Para combatir este problema, se desarrolló el concepto de bandas que se ajustan por volatilidad. (Kirkpatrick II, Dahlquist, 2010)

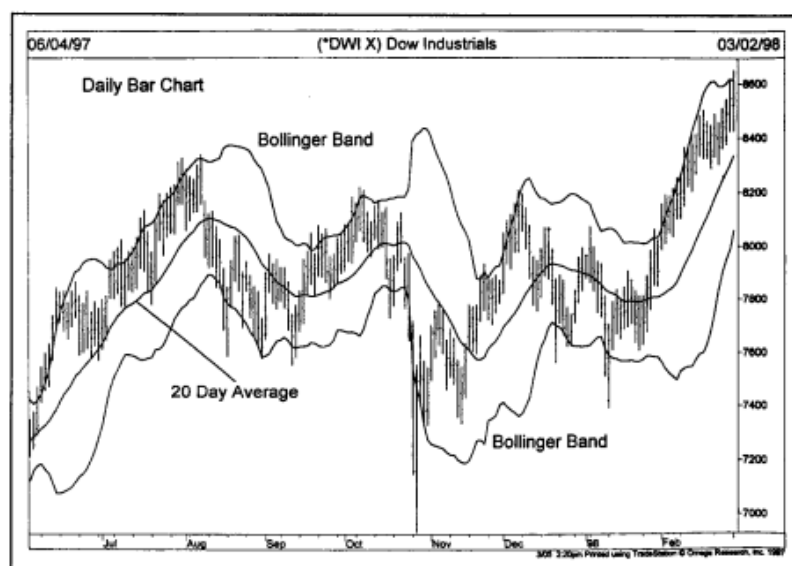
Bandas de Bollinger

Las bandas también son envolventes alrededor de un promedio móvil pero, en lugar de tener un tamaño fijo, se calculan para ajustarse a la volatilidad del precio alrededor de la media móvil. Por lo tanto, se contraen cuando los precios se calman y se expanden cuando los precios se vuelven volátiles. La banda más utilizada es la Banda de Bollinger, llamada así por John Bollinger (2002). (Kirkpatrick II, Dahlquist, 2010)

Murphy (1999) explica su construcción, interpretación y utilidad:

Se colocan dos bandas alrededor de una media móvil similar a la técnica de la envolvente porcentual. Excepto que las Bandas de Bollinger se colocan dos desviaciones estándar por encima y por debajo de la media móvil, que suele ser de 20 días. La desviación estándar es un concepto estadístico que describe cómo se dispersan los precios alrededor de un valor promedio. El uso de dos desviaciones estándar garantiza que el 95% de los datos de precios se encuentren entre las dos bandas comerciales. Como regla general, se considera que los precios están sobreextendidos al alza (sobre comprados) cuando tocan la banda superior. Se consideran sobreextendidos a la baja (sobre vendidos) cuando tocan la banda inferior. (Ver Figura 33)

Figura 33: Bandas de Bollinger alrededor de una media móvil de 20 días (Fuente: Murphy, 1999)



La forma más sencilla de usar las Bandas de Bollinger es usar las bandas superior e inferior como objetivos de precios. En otras palabras, si los precios rebotan en la banda inferior y superan el promedio de 20 días, la banda superior se convierte en el precio objetivo superior. Un cruce por debajo del promedio de 20 días identificaría la banda inferior como el objetivo a la baja. En una fuerte tendencia alcista, los precios normalmente fluctúan entre la banda superior y el promedio de 20 días. En ese caso, un cruce por debajo del promedio de 20 días advierte de un cambio de tendencia a la baja.

Experimentación: Bandas de Bollinger por período de 20 días

En lugar de combinar múltiples medias móviles resulta prometedor combinar una media móvil simple con Bandas de Bollinger. En particular, nos interesará probar con el parámetro más usado que es el período de 20 días.

Generamos las nuevas imágenes para alimentar el modelo, y se sostiene aquí también la leve variación de observaciones debido al armado de los gráficos con observaciones para las que se cuente con información de hasta 20+20 días previos en el conjunto de datos correspondiente (20 por la ventana del gráfico, y otros 20 por el cálculo de la media móvil, en total 40 días).

Una vez obtenidos los nuevos datos, entrenamos el modelo predictivo y obtenemos los resultados expuestos a continuación.

Tabla 11: Experimento 1 - Resultados con Bandas de Bollinger por período de 20 días

Modelo	Bandas de Bollinger de 20 días
train_class_distribution	
Down	47%
Up	53%
valid_class_distribution	
Down	40%
Up	60%
test_class_distribution	
Down	38%
Up	62%
epochs	10
train_loss	0.21
valid_loss	1.07
Resultados Validación	
Accuracy	59%
AUC-ROC	61%
Resultados Testing	
Accuracy	57%
AUC-ROC	61%

Figura 34: Experimento 1 - AUC-ROC con Bandas de Bollinger por período de 20 días

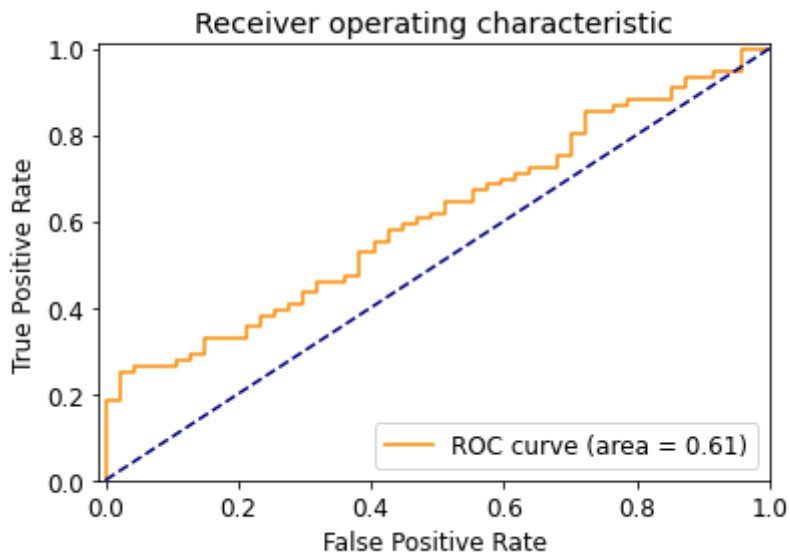
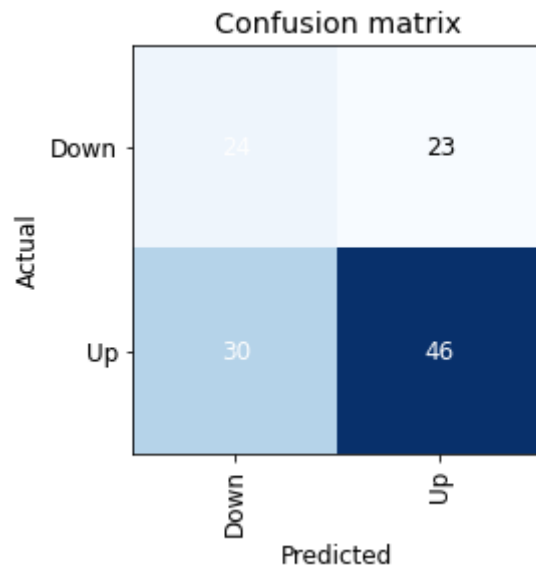


Figura 35: Experimento 1 - Matriz de confusión con Bandas de Bollinger por período de 20 días



Finalmente obtenemos una mejora más consistente. Los mejores resultados obtenidos hasta el momento, con accuracy y AUC-ROC rondando el 60%, tanto en validación como en test.

Experimentación: Bandas de Bollinger por período de 9 días

Nos interesaría ver qué resultados se obtendrían con un período más cortoplacista, por lo que volvemos a probar con un período de 9 días, pero en este caso combinando la media móvil con Bandas de Bollinger.

Generamos las nuevas imágenes para alimentar el modelo, y se sostiene aquí también la leve variación de observaciones debido al armado de los gráficos con observaciones para las que se cuente con información de hasta 20+9 días previos en el conjunto de datos correspondiente (20 por la ventana del gráfico, y otros 9 por el cálculo de la media móvil, en total 29 días). Veremos que la cantidad de observaciones lógicamente coincide con la de los conjuntos de datos creados anteriormente para el caso de la media móvil simple de 9 días.

Una vez obtenidos los nuevos datos, entrenamos el modelo predictivo y obtenemos los resultados expuestos a continuación.

Tabla 12: Experimento 1 - Resultados con Bandas de Bollinger por período de 9 días

Modelo	Bandas de Bollinger de 9 días
train_class_distribution	
Down	47%
Up	53%
valid_class_distribution	
Down	39%
Up	61%
test_class_distribution	
Down	39%
Up	61%
epochs	10
train_loss	0.2
valid_loss	1.22
Resultados Validación	
Accuracy	53%
AUC-ROC	57%
Resultados Testing	
Accuracy	49%
AUC-ROC	53%

Figura 36: Experimento 1 - AUC-ROC con Bandas de Bollinger por período de 9 días

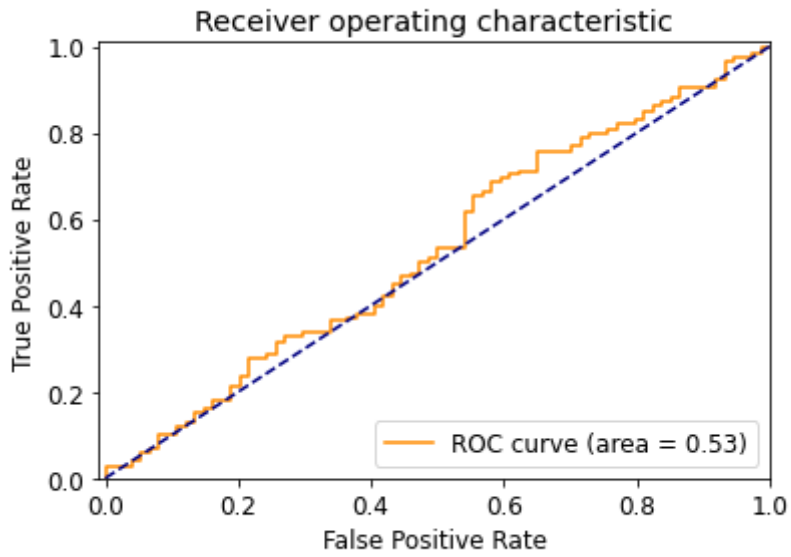
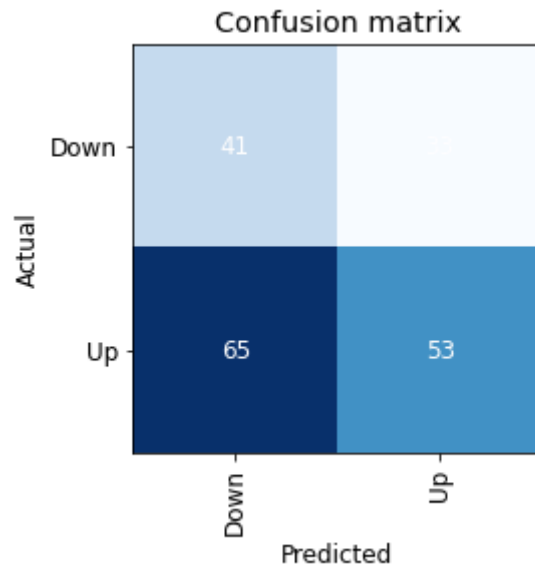


Figura 37: Experimento 1 - Matriz de confusión con Bandas de Bollinger por período de 9 días



Los resultados nuevamente caen, aunque se mantiene un poco la consistencia entre resultados de validación y test.

Se ve claramente que la combinación de velas con volumen, media móvil simple y bandas de Bollinger aporta mejor información al modelo predictivo que las demás alternativas exploradas hasta el momento.

c. Selección de Datos

El buen desempeño de cualquier modelo reside principalmente en los datos. Es por esto que venimos probando muchas variantes en su generación, ya sea desde el método de clasificación de los gráficos hasta el armado y contenido de los mismos. Sin embargo, hasta el momento estuvimos realizando variaciones con la data inicial existente, ya que fue la elegida y utilizada en el artículo en cuestión.

En esta instancia damos un paso más atrás para cuestionar la data subyacente de la que parte el modelo: precios diarios históricos de 10 acciones indias de alta capitalización, entre enero de 2018 y septiembre de 2020.

Probablemente el autor del artículo haya utilizado un conjunto de datos reducido priorizando la rápida ejecución de modelos. Sin embargo, con el objetivo de armar un modelo predictivo que pueda aprender lo mejor posible y obtener resultados satisfactorios y confiables, estar trabajando con datos limitados puede ser inconveniente.

Abordaremos dos cuestiones respecto a los datos:

- Cantidad

Actualmente se utilizan muy pocos datos, ya que se toman precios de tan solo 10 acciones indias, a lo largo de un año y medio. Es difícil que con tan poca información el modelo pueda encontrar todos los patrones o comportamientos posibles, generalizar y aprender de ellos.

- Calidad

Adicionalmente, al estar aprendiendo con información de 10 acciones indias, las conclusiones que derivan del entrenamiento con estos datos podrían no ser replicables a otras acciones o mercados. La muestra podría no estar siendo representativa.

Estas sospechas se resolverían utilizando más información, tanto de un período de tiempo más prolongado como de más acciones. También nos interesaría probar acciones que pudiesen ser más representativas del mercado mundial, para obtener conclusiones más generales.

Definición de los nuevos datos

En la plataforma financiera TradingView¹³ se establece que el mercado de valores de los Estados Unidos cubre alrededor del 75% de capitalización de mercado bursátil mundial, lo

¹³ <https://es.tradingview.com/markets/indices/quotes-us/>

que lo convierte en el mayor mercado de valores del mundo. También afirman que los índices bursátiles de Estados Unidos se pueden usar como una buena representación de diferentes sectores de la economía. Sería interesante utilizar las acciones que componen dichos índices para nuestro nuevo set de datos.

En particular, nos interesan el S&P500 y Nasdaq-100, dos índices bursátiles ponderados por capitalización de mercado, y por lo tanto compuestos por acciones de gran capitalización.

El S&P 500 es una representación del mercado de Estados Unidos, que abarca alrededor de 80%-85% del mercado de capitales del país, afirmó J. Gunzberg (2020), Directora Gerente de S&P Dow Jones Indices. Este índice tiene 500 componentes que representan un conjunto diverso de empresas de múltiples industrias pertenecientes a los 11 principales sectores económicos definidos por el Estándar de Clasificación de la Industria Global (GICS) (S&P Dow Jones Indices, 2021):

- Energía
- Materiales
- Industrial
- Consumo Discrecional
- Productos de Primera Necesidad
- Salud
- Finanzas
- Tecnologías de la Información
- Servicios de Comunicación
- Servicios de Utilidad Pública
- Bienes Inmobiliarios

El Nasdaq-100 incluye 100 de las empresas no financieras nacionales e internacionales más grandes que cotizan en el mercado de valores Nasdaq en función de la capitalización de mercado, según lo establece su página web.¹⁴ Está compuesto por activos en varios sectores, excluyendo los servicios financieros. Una gran parte del índice cubre el sector tecnológico, que representa el 56% del peso del índice. El siguiente sector más grande es el de servicios al consumidor, representado por empresas como cadenas de restaurantes, minoristas y servicios de viajes. Completan el índice los sectores de la salud, la industria y las telecomunicaciones. (Chen, 2021)

Como entre ambos índices predominan acciones de diferentes sectores, y a su vez los criterios que definen la inclusión de una empresa en cada índice son diferentes, la combinación de acciones de ambos índices podría generar un conjunto de datos prometedor al complementarse entre sí, alcanzando así nuestro objetivo de cantidad y calidad.

Por último, decidimos recolectar los precios históricos de dichas acciones entre 2009 y 2019. Queremos cubrir un período de tiempo más largo, para captar la mayor cantidad de movimientos posibles y que el modelo pueda generalizar. Dicho período es posterior a la crisis económica de 2008, y previo al desencadenamiento de la pandemia por Covid-19 de fines de

¹⁴ <https://www.nasdaq.com/nasdaq-100>

2019 (cuyo impacto económico se empezó a sentir recién en 2020), por lo cual estaríamos evitando así patrones excepcionales y posible ruido en los datos.

Armado de los nuevos datos

Selección de acciones

Fabozzi, Focardi y Kolm (2006) advierten:

Uno de los defectos más comunes (y peligrosos) de los datos muestrales son los llamados sesgos de supervivencia. El sesgo de supervivencia es consecuencia de seleccionar series temporales, en particular series temporales de precios de activos, en función de criterios que se aplican al final del período. Por ejemplo, suponga que una muestra contiene 10 años de datos de precios para todas las acciones que están en el S&P 500 hoy y que existieron durante los últimos 10 años. Esta muestra, aparentemente bien formada, está sin embargo sesgada: la selección, de hecho, se hace sobre las acciones de las empresas que están hoy en el S&P 500, es decir, aquellas empresas que han 'sobrevivido' en una forma lo suficientemente buena como para seguir siendo en el agregado S&P 500. El sesgo proviene del hecho de que muchas de las empresas sobrevivientes atravesaron con éxito un período difícil. Sobrevivir a la dificultad es una forma de reversión a la media que produce beneficios comerciales. Sin embargo, en el momento de la crisis era imposible predecir qué empresas en crisis habrían sobrevivido.

Para evitar el sesgo de supervivencia seleccionamos las acciones que componían ambos índices al comienzo del período de estudio, de modo que no ingrese información no válida al modelo. No existe un histórico de los componentes de los índices, por lo que obtenemos la lista de acciones que componían al S&P 500 y Nasdaq-100 al 01/01/2009 por nuestra cuenta a partir de la información disponible.

Para ambos índices, extraemos a través de la agencia Reuters¹⁵ la siguiente información:

- Lista de constituyentes del índice a la fecha (datos actuales)
- Lista de movimientos históricos de entradas y salidas de empresas al índice desde sus comienzos hasta la fecha presente (histórico de ajustes)

Esta información es suficiente para llegar a la lista de acciones que necesitamos. Para ello, partimos de la lista de componentes actuales y vamos deshaciendo los ajustes históricos hasta la fecha inicial de lo que será nuestro dataset: 01/01/2009.

Esto lo hacemos con ambos índices, y luego juntamos ambas listas. Claramente hay acciones que se encuentran en ambos índices, así que el número total de acciones es menor a la suma de la cantidad de componentes de ambos índices (600).

¹⁵ <https://www.reuters.com/>

Extracción y definición final de los datos

Luego descargamos de Yahoo Finance los precios históricos entre 2009 y 2019 para todas las acciones de la lista. Para algunas acciones no se encuentra información, esto puede deberse a varias razones, como que una empresa haya quebrado, o haya sido eliminada del índice por baja capitalización, o se haya fundido con otra compañía. De momento desestimamos estos casos ya que contamos con información completa de 394 acciones¹⁶, la cual nos parece bastante para una primera prueba de concepto.

Partición datos entre conjuntos de entrenamiento, validación y evaluación

Una vez que tenemos todos los datos, realizamos la partición entre conjuntos de entrenamiento, validación y evaluación. Como vimos anteriormente, lo hacemos respetando la temporalidad de los datos para evitar filtrar información entre conjuntos.

Con este criterio, los primeros 8 años (2009-2017) son utilizados para datos de entrenamiento, mientras el año consiguiente (2018) es utilizado para validar y el último año (2019) para testear. La partición de los datos es aproximadamente de 80% entrenamiento, 10% validación y 10% testeo.

Clasificación de los datos

El próximo paso es clasificar la data, esto lo hacemos con el [método original planteado en el artículo](#), ya que hemos visto que los métodos alternativos no obtuvieron resultados superiores, y aparte nos interesa ver primero el efecto de la modificación de los datos en el desempeño del modelo predictivo, *ceteris paribus*.

Generación de gráficos

Pasamos a generar la data que efectivamente alimenta al modelo predictivo: los gráficos, que guardamos en formato de imagen. Como ahora contamos con un volumen de datos mucho mayor, modificamos levemente la manera en la cual generamos los gráficos. En particular, en lugar de generar un gráfico para toda observación que se encuentre en un conjunto de datos, para cada acción se seleccionan aleatoriamente 12 observaciones de cada año de data disponible, para las cuales generamos el gráfico correspondiente.

Esto no afectaría la temporalidad de los datos, ya que dentro de cada imagen se conserva la temporalidad, pero sí haría que las imágenes estén menos correlacionadas entre sí, lo cual

¹⁶ Lista de acciones utilizadas disponible en el [Anexo](#).

esperamos que sume a la calidad de los datos. Anteriormente, si había una tendencia o comportamiento predominante en una acción, al generarse gráficos de corrido para todas las observaciones, se obtenían muchos gráficos similares, que no aportaban mucha información nueva al modelo, y a su vez asignaban más peso a dicho patrón al momento de entrenar. Ahora, estaríamos alimentando al modelo con información más diversa, con distribución o peso más equitativo para cada tipo de patrón que pudiera existir. Suponemos que esto podría ayudar al algoritmo a distinguir más patrones presentes en la data.

Entrenamiento del modelo con los nuevos datos

Contamos con la nueva data presentada con el mismo formato que la anterior, por lo que pasamos directamente a correr el modelo sin necesidad de hacer nada más.

Experimentación: Datos alternativos

Tabla 13: Experimento 1 - Resultados con datos alternativos

Modelo	Datos alternativos
train_class_distribution	
Down	46%
Up	54%
valid_class_distribution	
Down	58%
Up	42%
test_class_distribution	
Down	47%
Up	53%
epochs	10
train_loss	0.54
valid_loss	0.82
Resultados Validación	
Accuracy	49%
AUC-ROC	51%
Resultados Testing	
Accuracy	51%
AUC-ROC	51%

Figura 38: Experimento 1 - AUC-ROC con datos alternativos

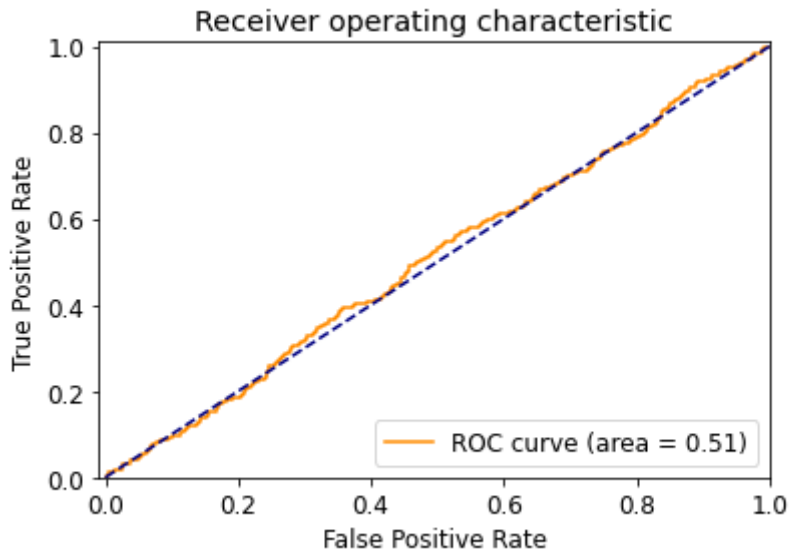
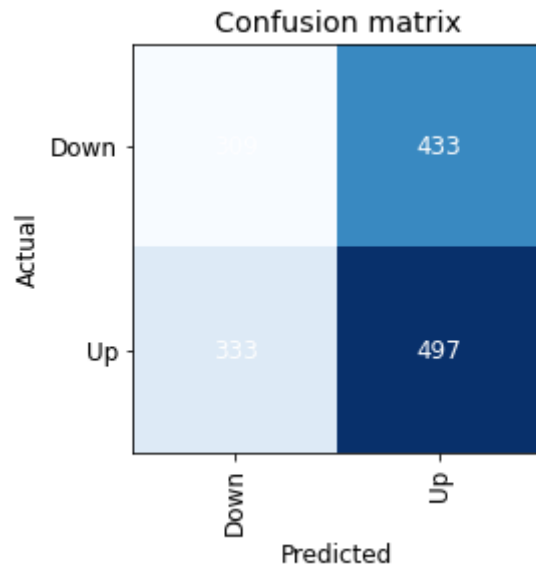


Figura 39: Experimento 1 - Matriz de confusión con datos alternativos



El modelo no obtiene un buen desempeño, pareciera ser que los nuevos datos no aportan información clara. Tanto accuracy como AUC-ROC dan 51%.

d. Implementación Conjunta

En los previos experimentos se evidenció que partiendo de malos resultados, al agregar la información necesaria al modelo de la manera correcta, se puede potenciar el desempeño del mismo. Por esto, en esta última instancia probamos combinar algunas de las diferentes alternativas que fuimos explorando, para ver si en conjunto logran mejorar el desempeño del modelo predictivo utilizando los datos alternativos recolectados.

Experimentación: Datos alternativos, clasificación original y bandas de Bollinger por período de 20 días

Tabla 14: Experimento 1 - Resultados con datos alternativos, clasificación original y bandas de Bollinger por período de 20 días

Modelo	Datos alternativos + Bandas de Bollinger de 20 días
train_class_distribution	
Down	45%
Up	55%
valid_class_distribution	
Down	57%
Up	43%
test_class_distribution	
Down	49%
Up	51%
epochs	10
train_loss	0.53
valid_loss	0.83
Resultados Validación	
Accuracy	49%
AUC-ROC	50%
Resultados Testing	
Accuracy	52%
AUC-ROC	53%

Figura 40: Experimento 1 - AUC-ROC con datos alternativos, clasificación original y bandas de Bollinger por período de 20 días

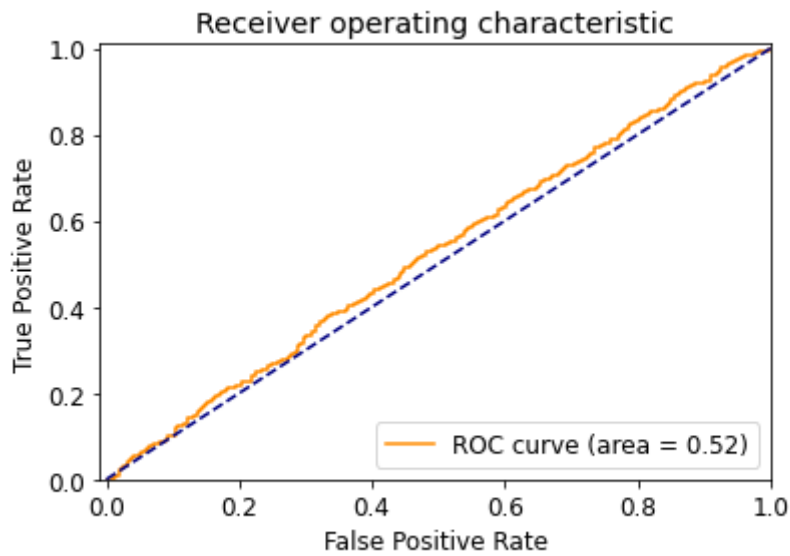
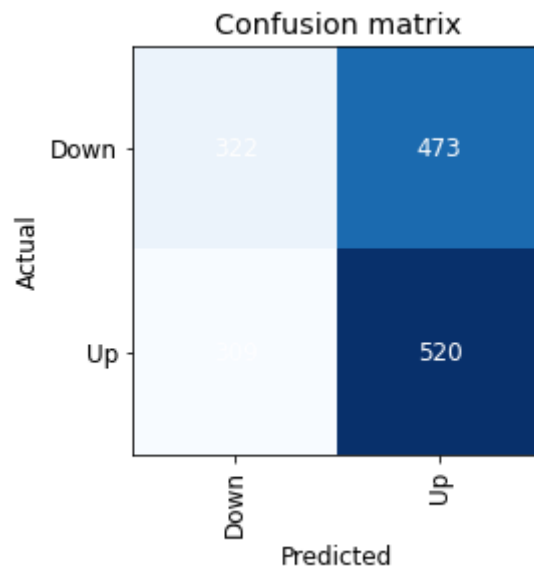


Figura 41: Experimento 1 - Matriz de confusión con datos alternativos, clasificación original y bandas de Bollinger por período de 20 días



Vemos que las bandas de Bollinger por un período de 20 días que tan útiles resultaron al aplicarlas a los datos originales, en este caso no logran prácticamente ninguna diferencia en la performance del modelo con los nuevos datos.

Consideramos que podría ser interesante probar de nuevo un método alternativo para clasificar los datos, por lo que agregamos este condimento al experimento.

Experimentación: Datos alternativos, clasificación alternativa ($k=10$, $p=0.025$, $t=0.1$) y bandas de Bollinger por período de 20 días

Tabla 15: Experimento 1 - Resultados con datos alternativos, clasificación alternativa ($k=10$, $p=0.025$, $t=0.1$) y bandas de Bollinger por período de 20 días

Modelo	Datos alternativos + Clasificación alternativa ($k=10$, $p=0.025$, $t=0.1$) + Bandas de Bollinger de 20 días
train_class_distribution	
Down	43%
Up	57%
valid_class_distribution	
Down	55%
Up	45%
test_class_distribution	
Down	45%
Up	55%
epochs	10
train_loss	0.55
valid_loss	0.82
Resultados Validación	
Accuracy	49%
AUC-ROC	51%
Resultados Testing	
Accuracy	53%
AUC-ROC	52%

Figura 42: Experimento 1 - AUC-ROC con datos alternativos, clasificación alternativa ($k=10$, $p=0.025$, $t=0.1$) y bandas de Bollinger por período de 20 días

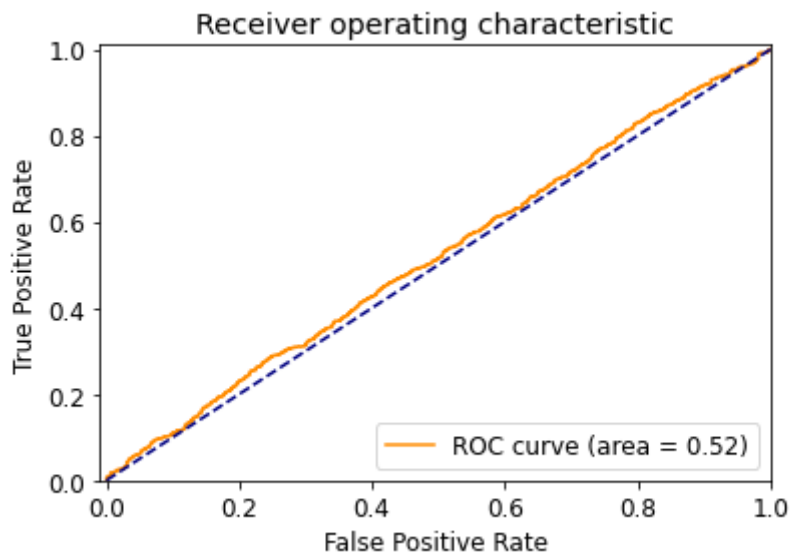
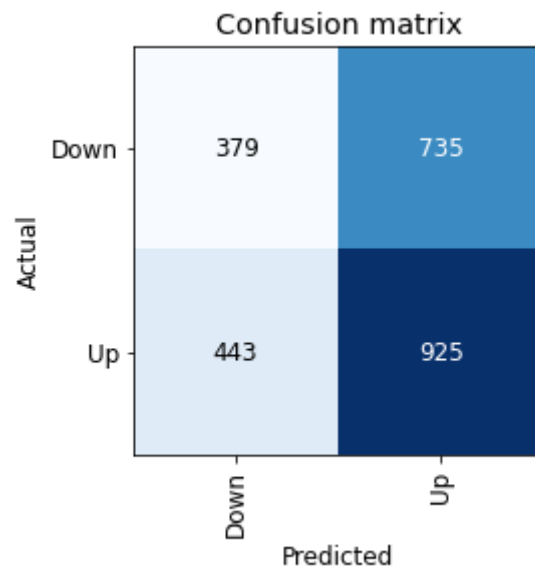


Figura 43: Experimento 1 - Matriz de confusión con datos alternativos, clasificación alternativa ($k=10$, $p=0.025$, $t=0.1$) y bandas de Bollinger por período de 20 días



Una vez más, vemos que los resultados se sostienen. Este modelo apenas logra superar al azar al entrenar con los nuevos datos, clasificados de manera alternativa, y con el adicional de Bandas de Bollinger de 20 días. Prevalece el modelo con datos de India, clasificación original y Bandas de Bollinger de 20 días como el mejor modelo.

3.1.4 Conclusiones

En primer lugar, se evidenció la importancia de saber con qué datos se está trabajando, y cómo tratar con ellos. Por un simple error se puede llegar a resultados y conclusiones desacertados. En este caso, por no respetar la temporalidad de los datos al dividir los conjuntos de entrenamiento y validación, se filtró información futura en la instancia de entrenamiento, lo cual tiene el peligroso efecto de sobreestimar el rendimiento esperado del modelo. Esto se apreció al corregir dicho error y comprobar que el desempeño real del modelo era peor.

Tabla 16: Resultados del Experimento 1 con datos de India

Modelo				Validación		Test	
Algoritmo	Variable Y	Clasificación	Características	Accuracy	AUC-ROC	Accuracy	AUC-ROC
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Split de datos incorrecto + Velas + volumen	69%	74%	-	-
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen	50%	57%	53%	54%
Resnet34	Tendencia próximos 5 días	“Alternativa Torgo”	Velas + volumen	47%	53%	50%	52%
Resnet34	Tendencia próximos 10 días	“Alternativa Torgo”	Velas + volumen	50%	50%	48%	51%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen + media móvil de 9 días	46%	45%	57%	56%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen + media móvil de 20 días	48%	51%	57%	61%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen + medias móviles de 5 y 20 días	53%	57%	51%	51%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen + medias móviles de 4, 9 y 18 días	43%	44%	46%	51%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen + bandas de Bollinger de 20 días	59%	61%	57%	61%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen + bandas de Bollinger de 9 días	53%	57%	49%	53%

Tabla 17: Resultados del Experimento 1 con datos de Estados Unidos

Modelo				Validación		Test	
Algoritmo	Variable Y	Clasificación	Características	Accuracy	AUC-ROC	Accuracy	AUC-ROC
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen	49%	51%	51%	51%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen + bandas de Bollinger de 20 días	49%	50%	52%	53%
Resnet34	Tendencia próximos 10 días	“Alternativa Torgo”	Velas + volumen + bandas de Bollinger de 20 días	49%	51%	53%	52%

En segundo lugar, vimos que hay muchas maneras de abordar este (o cualquier) problema predictivo. El algoritmo de redes neuronales que usamos para reconocer patrones en el comportamiento de los precios de acciones y predecir su tendencia, no obtenía resultados tan prometedores luego de corregir el error mencionado anteriormente. Experimentamos con una variedad de alternativas con el objetivo de pasarle más información, y de manera más útil, al algoritmo. Probamos muchos enfoques, y tras combinar conocimientos de ciencia de datos con conceptos financieros, descubrimos cuál información adicional confundía más al modelo, y cual lo ayudaba. En particular, agregando bandas de Bollinger sobre una media móvil de 20 días a los gráficos de velas y volumen, obtuvimos los mejores resultados, los cuales superan tanto a los resultados originales como también al azar. Esto último es un progreso muy importante.

Aún así es importante destacar que el problema de predicción es complejo y que el algoritmo de predicción, aún cuando logra superar el azar, nunca tiene performance como la que se sugería en el artículo original.

3.2 Experimento 2: Using Deep Learning Neural Networks and Candlestick Chart Representation To Predict Stock Market

El paper *Using Deep Learning Neural Networks and Candlestick Chart Representation to Predict Stock Market* (Kusuma, Ho, Kao, Ou, Hua, 2019) disponible en arXiv¹⁷ aborda el problema de predicción del mercado financiero mediante el uso de redes neuronales profundas y gráficos de velas, logrando resultados muy prometedores en mercados asiáticos. Los autores adicionalmente ofrecen un marco de apoyo para la toma de decisiones de inversión mediante la implementación de su modelo predictivo en una herramienta disponible en la web¹⁸ que proporciona indicaciones sugeridas sobre la dirección futura del precio de las acciones.

Este trabajo ya está siendo citado por la literatura, en Google Scholar ya cuenta con 26 citas a la fecha (mayo 2022), muchas hechas por artículos publicados en revistas con referencias. El hecho de que el paper haya sido citado varias veces, podría indicar que el trabajo ha sido revisado y aceptado por otros adeptos al tema. Sin embargo, obtener una precisión mayor al 90%¹⁹ al predecir el mercado financiero parece demasiado bueno para ser verdad, e inevitablemente genera sospechas. Por último, los autores de este trabajo exploran una serie de algoritmos de redes neuronales profundas diseñadas a medida del problema predictivo en cuestión.

Resulta muy interesante por lo tanto adentrarnos en la metodología seguida en este trabajo para comprobar su validez y a su vez explorar la alternativa a la utilización de redes neuronales pre-entrenadas estudiada en el experimento anterior, para ver si la construcción de una red neuronal convolucional es la clave del éxito.

Lamentablemente al mirar detalladamente el código en cuestión descubriremos que nuestras sospechas eran ciertas y que la clave del “éxito” reside principalmente en un error conceptual.

¹⁷ <https://arxiv.org>

¹⁸ <http://140.138.155.216/deepcandle/>

¹⁹ El resultado mayor a 90% que promete el paper es sobre el conjunto de validación.

3.2.1 Metodología del paper

El paper trabaja con datos de dos mercados (en paralelo), en caso de observar diferencias de resultados entre mercados a lo largo de su investigación. En particular, los autores optan por utilizar precios diarios históricos de 50 acciones de Taiwan y de 10 acciones de Indonesia, entre enero del 2000 y diciembre de 2017. Adicionalmente utilizan datos de dos índices para realizar evaluaciones independientes de los modelos para ambos mercados: Yuanta/P-shares para Taiwan y Jakarta Composite Index para Indonesia. Obtienen todos los precios históricos a través de la API de Yahoo Finance.

Realizan la partición de conjuntos de datos respetando la temporalidad de los mismos:

Tabla 18: Experimento 3 - Conjuntos de datos utilizados

Dataset	Entrenamiento	Validación	Independiente
TW50	2000/01/01 - 2016/12/31	2017/01/01 - 2018/06/14	2017/01/01 - 2018/06/14
ID10	2000/01/01 - 2016/12/31	2017/01/01 - 2018/06/14	2017/01/01 - 2018/06/14

Una peculiaridad de este paper es que no se destina ninguna sección a explicar algo que resulta sumamente importante para cualquier problema predictivo: la construcción de la variable a predecir. En la sección de Metodología de su artículo, los autores mencionan que dicha variable tomará valores binarios: “Up” o “Down”, según el precio suba o baje en un *futuro cercano*²⁰. Dicho futuro cercano sería el próximo día, según un detalle adicional que encontramos en la Introducción²¹. Al adentrarnos en el código confirmamos que efectivamente se está haciendo esto. Por lo tanto, en este caso se trabaja con la alternativa de construcción de la variable Y más simple que planteamos en la sección de Metodología de esta tesis, básicamente partiendo del día i , predecir si el precio de cierre de mercado subirá o bajará al día siguiente, $i+1$:

$$Y_i = \begin{cases} Down & \text{si } C_{i+1} > C_i \\ Up & \text{si } C_{i+1} \leq C_i \end{cases} \quad \text{donde } C \text{ es el precio de cierre} \quad (14)$$

²⁰ “our candlestick charts are feed as input into some deep learning neural networks model to find the best model for stock market prediction, and the outputs will be binary class to indicate the stock price will going up or down in the near future.” (Sección 4, página 4)

²¹ “check whether it will be going up or going down in the next day” (Sección 1, página 2)

Figura 44: Experimento 2 - Extracto del código donde se realiza la clasificación mencionada (Fuente: Kusuma, Ho, Kao, Ou, Hua, 2019)

```
105 def createLabel(fname, seq_len):
106     # python preprocess.py -m createLabel -l 20 -i stockdatas/EWT_training5.csv
107     print("Creating label . . .")
108     # remove existing label file
109     filename = fname.split('/')
110     # print("{} - {}".format(filename[0], filename[1][:-4]))
111     removeOutput("{}_label_{}.txt".format(filename[1][:-4], seq_len))
112
113     df = pd.read_csv(fname, parse_dates=True, index_col=0)
114     df.fillna(0)
115
116     df.reset_index(inplace=True)
117     df['Date'] = df['Date'].map(mdates.date2num)
118     for i in range(0, len(df)):
119         c = df.ix[i:i + int(seq_len), :]
120
121         starting = 0
122         endvalue = 0
123         label = ""
124
125         if len(c) == int(seq_len)+1:
126             starting = c["Close"].iloc[-2]
127             endvalue = c["Close"].iloc[-1]
128             # print(f'endvalue {endvalue} - starting {starting}')
129             if endvalue > starting:
130                 label = 1
131             else:
132                 label = 0
```

Se construyen las imágenes con una serie de variantes, para ver si alguna resulta más útil para la tarea de predicción. De esta manera, se construyen gráficos de velas alternando entre combinaciones con y sin barras de volumen adicionales, con dos tamaños de imagen (dimensiones de 20 y de 50 pixeles) y con tres ventanas de tiempo (de 5, 10 y 20 días).

Los autores prueban una serie de algoritmos de redes de aprendizaje profundo y evalúan su desempeño en términos de sensibilidad, especificidad, precisión y MCC. Finalmente se implementa el algoritmo de mejor performance en un servidor web disponible en <http://140.138.155.216/deepcandle/> . Allí ofrecen una herramienta con la cual los usuarios pueden predecir el mercado de valores en el futuro cercano, sin un conocimiento profundo de la informática.

A efectos de nuestro análisis nos enfocaremos únicamente en el modelo que obtiene el mejor desempeño, una red neuronal convolucional hecha a medida de la siguiente manera: 4 capas convolucionales 2D, 4 capas de max pooling y 3 dropouts.

Figura 45: Experimento 2 - Arquitectura de la CNN propuesta en el paper (Fuente: Kusuma, Ho, Kao, Ou, Hua, 2019)

Input
Conv2D-32 ReLU
max-pooling
Conv2D-48 ReLU
max-pooling
Dropout
Conv2D-64 ReLU
max-pooling
Conv2D-96 ReLU
max-pooling
Dropout
Flatten
Dense-256
Dropout
Dense-2

3.2.2 Metodología del Experimento 2

Tenemos dos objetivos:

- 1) Corroborar la metodología y resultados planteados por el paper
- 2) Evaluar el efecto en la performance al utilizar otra construcción de variable a predecir, y también al utilizar otro algoritmo de redes neuronales.

Para poder realizar una comparación provechosa con la metodología y resultados del Experimento 1, trabajamos de la siguiente manera:

- Datos: corroboramos resultados de validación con el modelo CNN con los datos de Indonesia, pero adicionalmente también probaremos la metodología de este paper con los datos de India que venimos trabajando en el experimento anterior.
- Imágenes: Generamos imágenes de gráficos de velas con volumen para períodos de tiempo de 20 días y tamaño 200 píxeles, ya que son las características que veníamos utilizando en el Experimento 1.
- Clasificación: Aplicaremos la definición de variable a predecir utilizada en este paper (donde Y describe el comportamiento del precio de cierre del día siguiente), para ver qué efectos tiene en la performance con datos de India también.
- Predicción: Aplicaremos el modelo de CNN construido por los autores de este paper (utilizando la librería de machine learning *PyTorch*²²), y adicionalmente también estaremos usando el modelo pre entrenado Resnet34 que veníamos trabajando en el experimento anterior para comparar su desempeño.

²² <https://pytorch.org>

3.2.3 Replicación de resultados

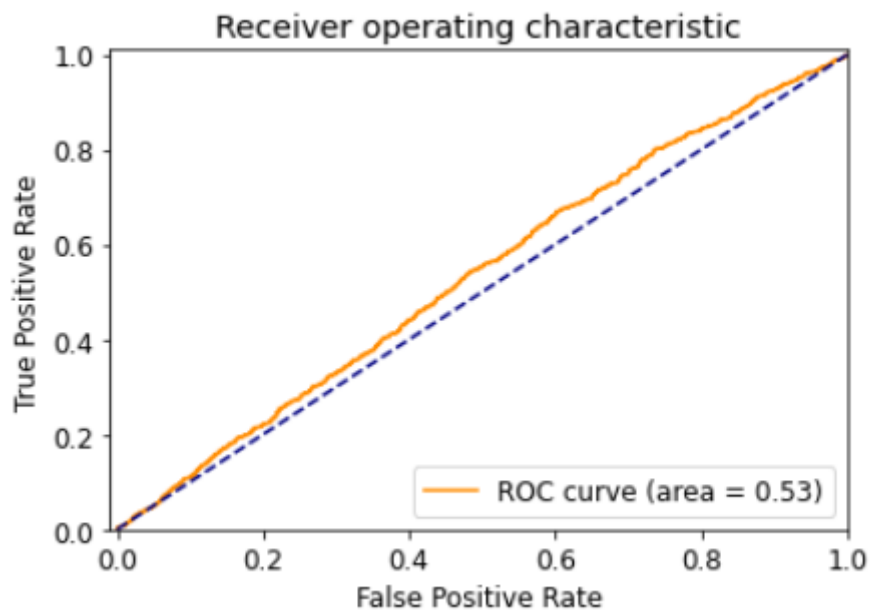
Una vez más, como punto de inicio de nuestra investigación, replicamos el trabajo realizado en el paper según la metodología descrita, con los datos de acciones de Indonesia²³, para comprobar si se alcanza el resultado afirmado por los autores: 92% de accuracy en validación.

Sin embargo, al hacer esto obtenemos resultados completamente diferentes:

Tabla 19: Experimento 2 - Resultados con datos de Indonesia + Clasificación paper + CNN

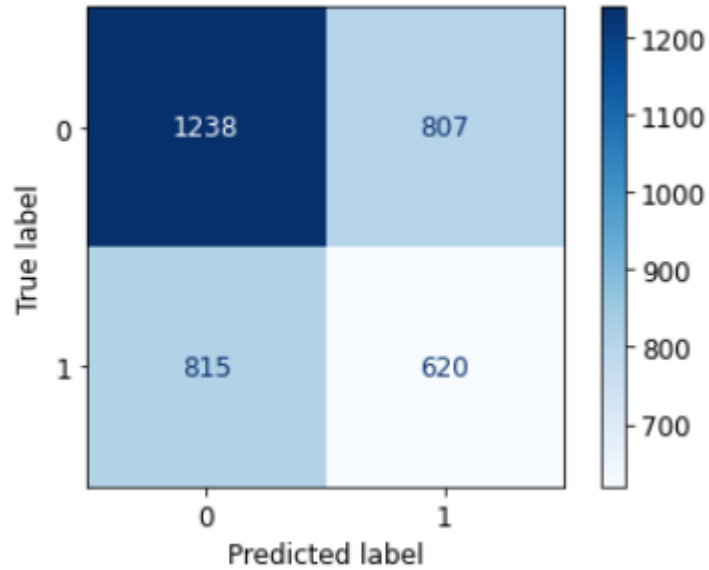
Modelo	Indonesia + Clasificación paper + CNN
train_class_distribution	
Down	57%
Up	43%
valid_class_distribution	
Down	59%
Up	41%
Resultados Validación	
Accuracy	53%
AUC-ROC	53%

Figura 46: Experimento 2 - AUC-ROC con datos de Indonesia + Clasificación paper + CNN



²³ Lista de acciones utilizadas disponible en el [Anexo](#).

Figura 47: Experimento 2 - Matriz de confusión con datos de Indonesia + Clasificación paper + CNN



Cuando intentamos replicar los resultados utilizando el código disponible en github del autor, no se llega al resultado sugerido. Al revisar el código vemos que el mismo ha sido editado, y al ver las ediciones del código descubrimos que la versión del código anterior a la fecha de publicación de este artículo tiene una definición de Y diferente a la descrita en el paper.²⁴

En particular, dicha definición construye la variable a predecir de la siguiente manera:

$$Y_i = \begin{cases} Down & \text{si } C_{i+1} > C_{i-(k-1)} \\ Up & \text{si } C_{i+1} \leq C_{i-(k-1)} \end{cases} \quad (15)$$

donde C es el precio de cierre y k la ventana del gráfico

Por lo tanto, en lugar de comparar el precio del día i contra el del día siguiente $i+1$, compara el precio del principio de la ventana de tiempo analizada con el gráfico de velas contra el precio del día siguiente $i+1$. En nuestro caso, estamos trabajando con una ventana de 20 días.

Cuando usamos esta definición, llegamos a la performance expuesta en el paper:

²⁴ En el [Anexo](#) se incluye el detalle de las ediciones en el código.

Tabla 20: Experimento 2 - Resultados con datos de Indonesia + Clasificación inicial + CNN

Modelo	Indonesia + Clasificación inicial + CNN
train_class_distribution	
Down	44%
Up	56%
valid_class_distribution	
Down	40%
Up	60%
Resultados Validación	
Accuracy	93%
AUC-ROC	98%

Figura 48: Experimento 2 - AUC-ROC con datos de Indonesia + Clasificación inicial + CNN

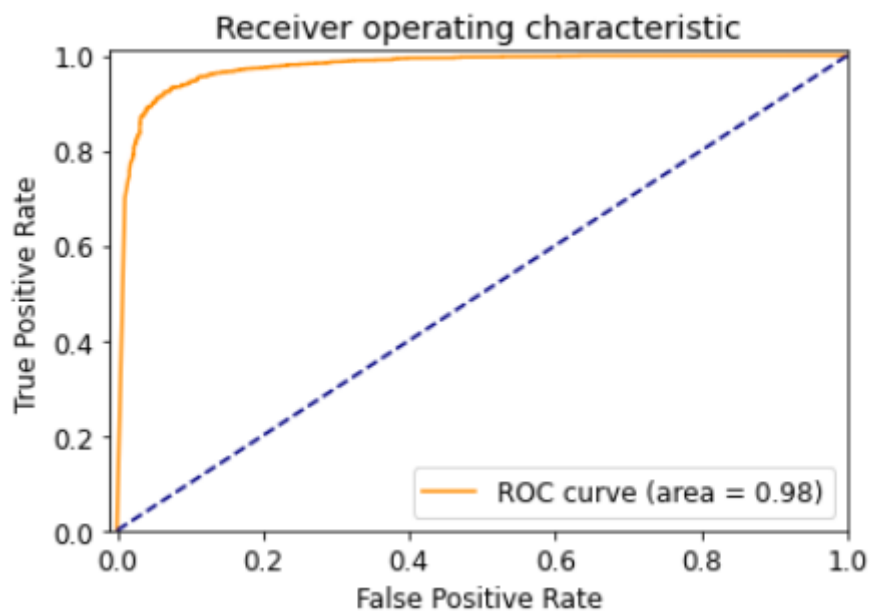
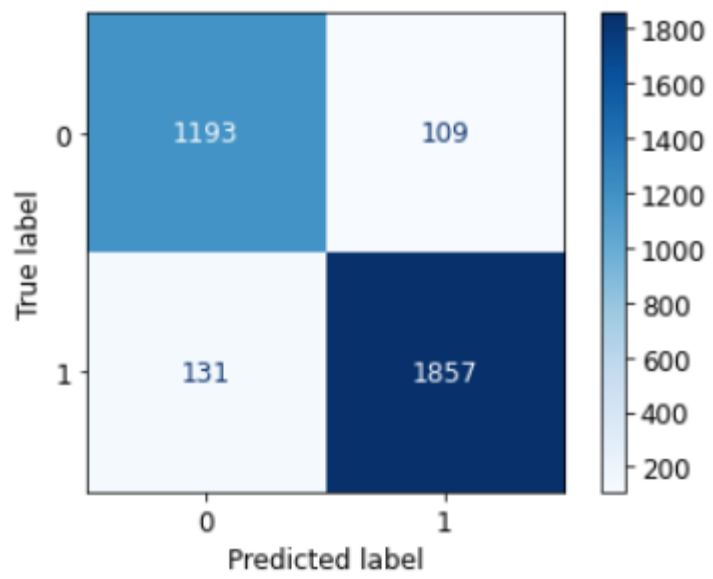


Figura 49: Experimento 2 - Matriz de confusión con datos de Indonesia + Clasificación inicial + CNN



3.2.4 Error

Los autores publicaron un paper que presenta resultados que adjudican a una metodología, pero que en realidad alcanzaron con otra. Y esta *otra* clasificación, llamémosla clasificación “inicial”, comprende una decisión metodológica extraña: comparar el comienzo del período del gráfico de velas contra el día siguiente al mismo.

Esa es una mala forma de armar la Y , porque estamos tratando con datos que son autorregresivos. Entonces si por ejemplo el precio de la acción viene subiendo, generalmente no va a bajar repentinamente por debajo del inicio del periodo. Entonces este problema de predicción es muy simple, ya con ver como va el gráfico es fácil inferir como sigue comportándose el precio.

Este problema de predicción no solamente es simple o tiene problemas estadísticos, sino que tampoco tiene mucha utilidad de negocios. Al trader no le interesa si “mañana va a estar más alta la acción de lo que estuvo hace un mes”, si está evaluando su decisión de inversión al día de hoy.

Cuando se corrige esta metodología y se utiliza el criterio de clasificación *correcto* planteado en el paper, la performance cae drásticamente, alcanzando una accuracy de 53%.

Nuevamente nos encontramos en el punto de partida que ronda el azar.

3.2.5 Experimentación

Seguimos adelante con el método de clasificación planteado en el paper y que consideramos correcto, que partiendo del día i predice si el precio de cierre de mercado subirá o bajará al día siguiente $i+1$:

$$Y_i = \begin{cases} Down & \text{si } C_{i+1} > C_i \\ Up & \text{si } C_{i+1} \leq C_i \end{cases} \quad \text{donde } C \text{ es el precio de cierre} \quad (16)$$

Ya vimos el desempeño del modelo con datos de Indonesia y el algoritmo de CNN utilizados en el paper. En esta sección evaluaremos adicionalmente el desempeño del modelo con el algoritmo Resnet34 y datos de India utilizados en el Experimento 1.

a. Datos de Indonesia (Paper)

Utilizando el criterio correcto planteado por el paper (precio de día i contra día $i+1$) y datos de Indonesia, probamos predecir con el algoritmo Resnet34 que veníamos utilizando en el Experimento 1. Seguimos comparando resultados en la instancia de validación como se hizo en el paper con los datos disponibles de Indonesia.

Experimentación: Datos de Indonesia, clasificación paper (correcta) y Resnet34

Tabla 21: Experimento 2 - Resultados con datos de Indonesia + Clasificación paper + Resnet34

Modelo	Indonesia + Clasificación paper + Resnet34
Datos	Indonesia
Imágenes	Velas + volumen
Algoritmo	Resnet34
Clasificación	Paper
train_class_distribution	
Down	57%
Up	43%
valid_class_distribution	
Down	59%
Up	41%
Resultados Validación	
Accuracy	56%
AUC-ROC	53%

Figura 50: Experimento 2 - AUC-ROC con datos de Indonesia + Clasificación paper + Resnet34

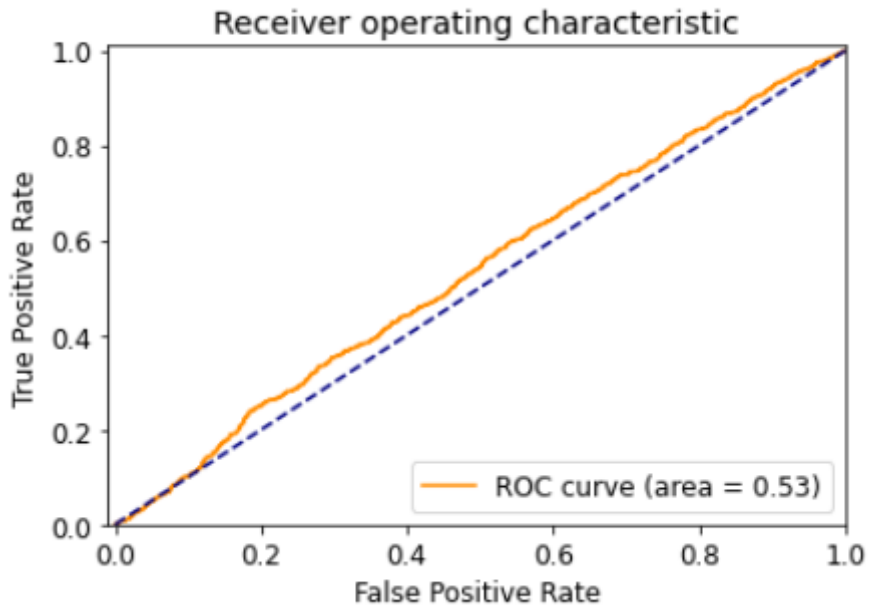
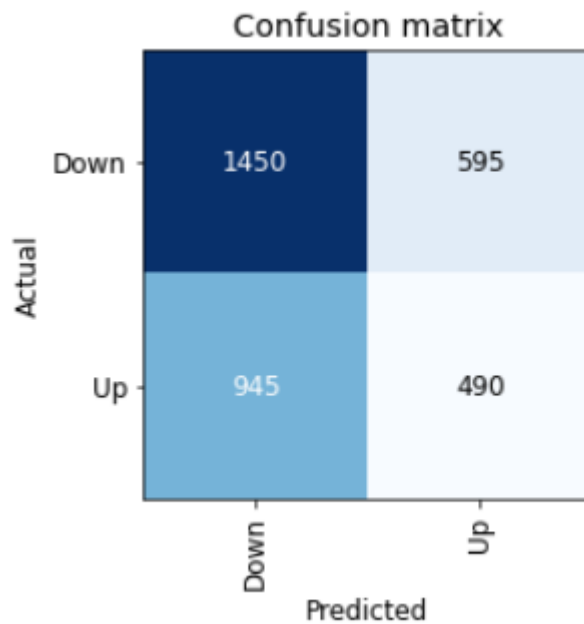


Figura 51: Experimento 2 - Matriz de confusión con datos de Indonesia + Clasificación paper + Resnet34



En términos de AUC-ROC obtenemos 53% al igual que con el modelo CNN, sin embargo pasamos de un 53% a un 56% en términos de accuracy. El modelo Resnet34 pareciera alcanzar levemente mejores resultados que la CNN planteada en el paper.

Veamos si esto se mantiene con datos de India.

b. Datos de India (Experimento 1)

Ahora pasamos a evaluar el desempeño con datos de India, para los cuales contamos con conjuntos de datos de entrenamiento, validación y testing.

Experimentación: Datos de India, clasificación paper (correcta) y CNN

Partimos por replicar la metodología del paper con los datos de India: con la definición de Y correcta y el algoritmo de CNN.

Tabla 22: Experimento 2 - Resultados con datos de India + Clasificación paper + CNN

Modelo	India + Clasificación paper + CNN
train_class_distribution	
Down	50%
Up	50%
valid_class_distribution	
Down	43%
Up	57%
test_class_distribution	
Down	49%
Up	51%
Resultados Validación	
Accuracy	45%
AUC-ROC	44%
Resultados Testing	
Accuracy	48%
AUC-ROC	47%

Figura 52: Experimento 2 - AUC-ROC con datos de India + Clasificación paper + CNN

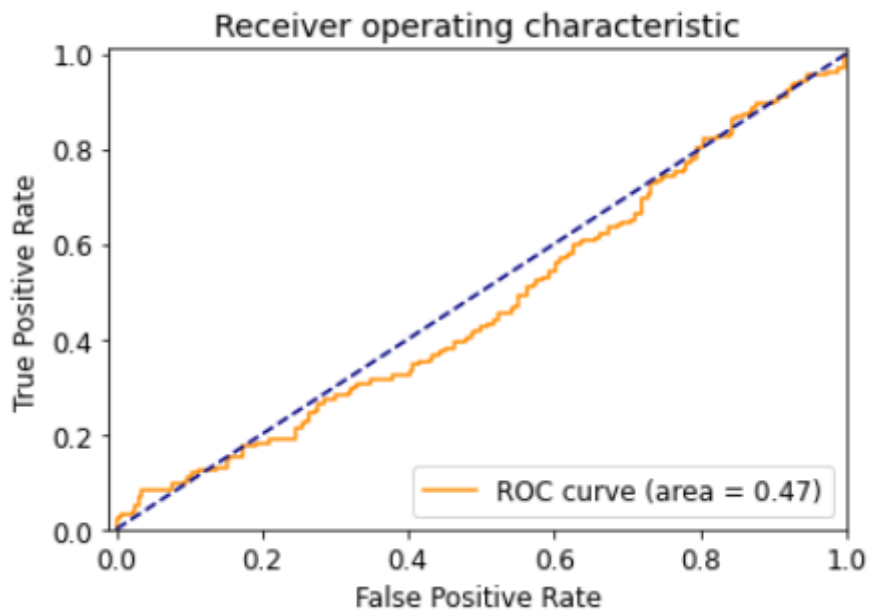
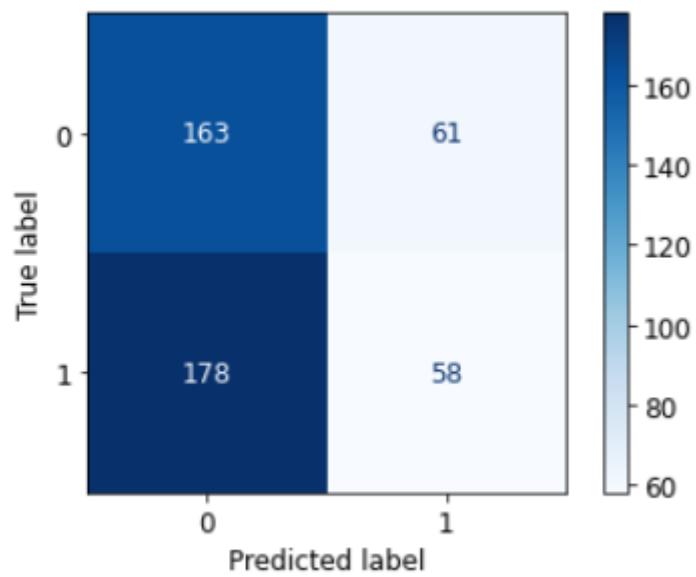


Figura 53: Experimento 2 - Matriz de confusión con datos de India + Clasificación paper + CNN



Se mantiene el pobre desempeño alcanzado con el algoritmo de CNN para datos de India, en este caso peor que el azar.

Experimentación: Datos de India, clasificación paper (correcta) y Resnet34

Veamos si los resultados mejoran con el algoritmo Resnet.

Tabla 23: Experimento 2 - Resultados con datos de India + Clasificación paper + Resnet34

Modelo	India + Clasificación paper + Resnet34
train_class_distribution	
Down	50%
Up	50%
valid_class_distribution	
Down	43%
Up	57%
test_class_distribution	
Down	49%
Up	51%
Resultados Validación	
Accuracy	51%
AUC-ROC	49%
Resultados Testing	
Accuracy	51%
AUC-ROC	50%

Figura 54: Experimento 2 - AUC-ROC con datos de India + Clasificación paper + Resnet34

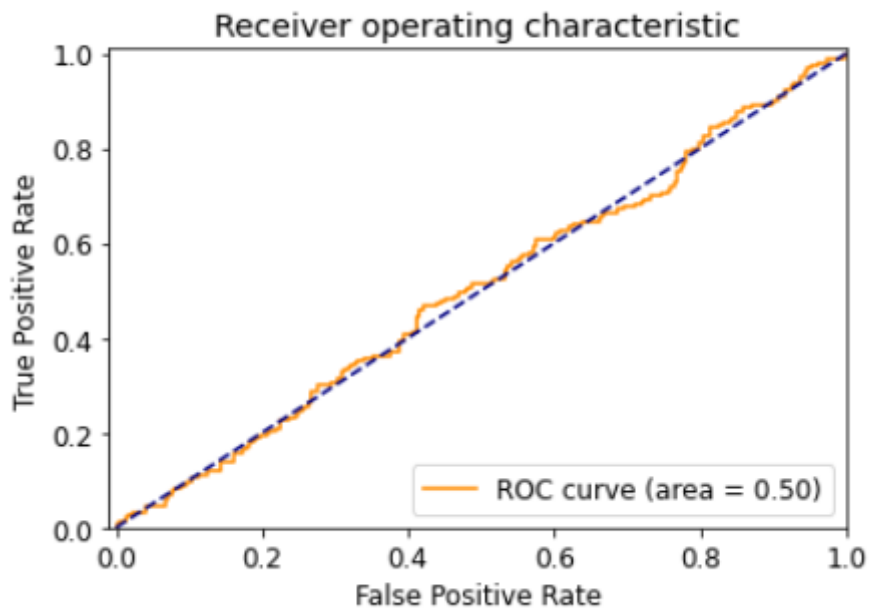
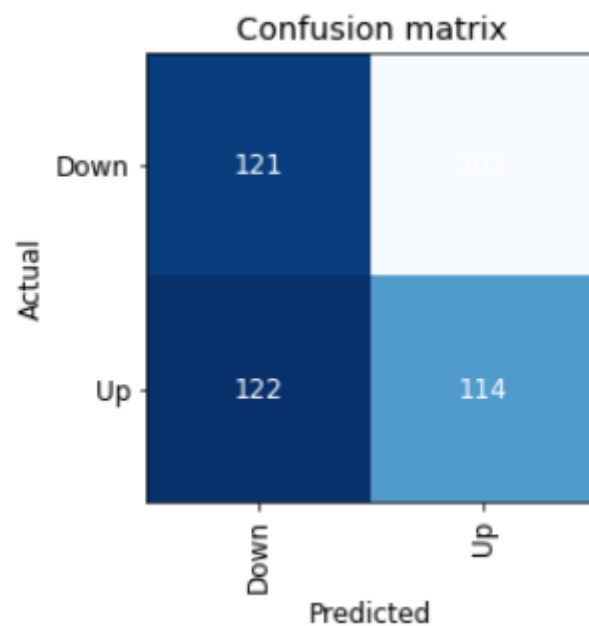


Figura 55: Experimento 2 - Matriz de confusión con datos de India + Clasificación paper + Resnet34



Efectivamente, con el algoritmo Resnet34 se alcanzan mejores resultados. De todas maneras, simplemente se logra alcanzar al azar, tampoco es un desempeño muy bueno. Los resultados obtenidos en el Experimento 1 con los mismos datos y algoritmo, pero el criterio de clasificación que toma a la variable Y como tendencia, siguen siendo superiores.

Experimentación: Datos de India, clasificación paper (correcta), Resnet34 y bandas de Bollinger por período de 20 días

En el Experimento 1 se evidenció que agregando bandas de Bollinger por un período de 20 días a los gráficos de velas y volumen, se lograba pasar información útil al modelo predictivo, obteniendo el mejor modelo hasta el momento.

Resulta interesante evaluar si agregando este indicador a los gráficos en este caso podrá mejorar los resultados de igual manera. Dicho de otra forma, probaremos el mejor modelo del Experimento 1 con la variable Y del paper del Experimento 2.

Tabla 24: Experimento 2 - Resultados con datos de India + Clasificación paper + Resnet34 + Bandas de Bollinger de 20 días

Modelo	India + Clasificación paper + Resnet34 + Bandas de Bollinger de 20 días
train_class_distribution	
Down	50%
Up	50%
valid_class_distribution	
Down	45%
Up	55%
test_class_distribution	
Down	30%
Up	70%
Resultados Validación	
Accuracy	51%
AUC-ROC	50%
Resultados Testing	
Accuracy	48%
AUC-ROC	51%

Figura 56: Experimento 2 - AUC-ROC con datos de India + Clasificación paper + Resnet34 + Bandas de Bollinger de 20 días

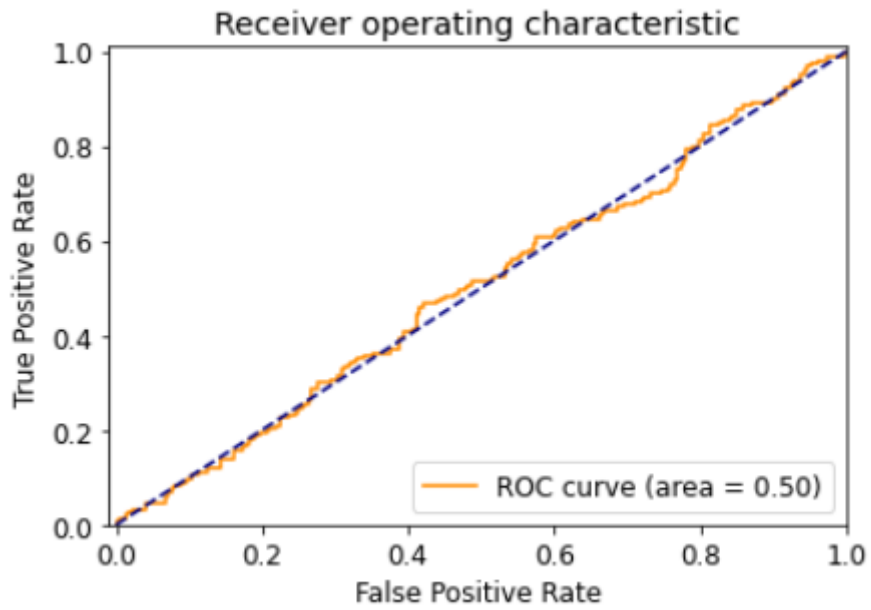
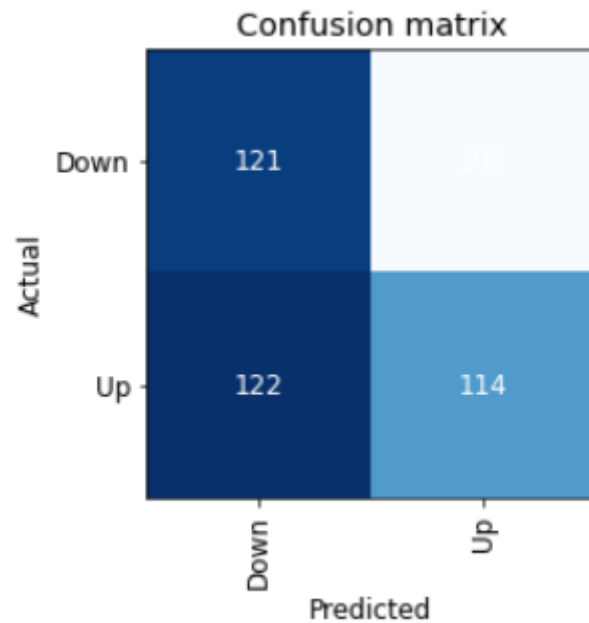


Figura 57: Experimento 2 - Matriz de confusión con datos de India + Clasificación paper + Resnet34 + Bandas de Bollinger de 20 días



En este caso las bandas de Bollinger no logran aportar información adicional que le resulte útil al algoritmo Resnet34 con este método de clasificación.

3.2.6 Conclusiones

Kusuma, Ho, Kao, Ou y Hua han publicado un paper con información ambigua y engañosa, ya que presentan resultados que adjudican a una metodología pero alcanzaron con otra. Comprobamos que los gráficos de velas permiten predecir una variable binaria que no es interesante a nivel económico, pero que al ser pasada a una definición significativa, se cae la performance del modelo.

Hasta ahora no encontramos ninguna combinación metodológica que logre predecir mejor que el azar. El modelo Resnet34 del Experimento 1 alcanza resultados levemente mejores que la red neuronal convolucional presentada como mejor modelo en este paper. Replicamos la estructura de la variante del Experimento 1 que mejor funcionó, agregando bandas de Bollinger de 20 días a los gráficos de velas y volumen, con la variable Y definida en el Experimento 2. Este modelo, que en el Experimento 1 funcionaba, con los mismos datos, al tratar de predecir la nueva variable Y, pierde su utilidad. Por lo tanto, predecir la variable Y construída como el comportamiento del precio del día siguiente pareciera ser más complejo que predecir la variable Y construída como la tendencia del precio en los próximos días, ya que dejando las demás variables constantes empeoran los resultados.

Tabla 25: Resultados del Experimento 2 con datos de Indonesia

Modelo				Validación	
Algoritmo	Variable Y	Clasificación	Características	Accuracy	AUC-ROC
CNN	Comportamiento precio día siguiente contra inicio de ventana (19 días antes)	Inicial incorrecta	Velas + volumen	93%	98%
CNN	Comportamiento precio día siguiente	Paper Experimento 2	Velas + volumen	53%	53%
Resnet34	Comportamiento precio día siguiente	Paper Experimento 2	Velas + volumen	56%	53%

Tabla 26: Resultados del Experimento 2 con datos de India

Modelo				Validación		Test	
Algoritmo	Variable Y	Clasificación	Características	Accuracy	AUC-ROC	Accuracy	AUC-ROC
CNN	Comportamiento precio día siguiente	Paper Experimento 2	Velas + volumen	45%	44%	48%	47%
Resnet34	Comportamiento precio día siguiente	Paper Experimento 2	Velas + volumen	51%	49%	51%	50%
Resnet34	Comportamiento precio día siguiente	Paper Experimento 2	Velas + volumen + bandas de Bollinger de 20 días	51%	50%	48%	51%

3.3 Experimento 3: Predicción de tendencia de precios de activos financieros mediante el análisis de indicadores técnicos con Aprendizaje Automático

Este experimento es diferente a los anteriores ya que no tiene la motivación en un paper, pero sí en la literatura.

Hasta ahora, a lo largo de la experimentación hemos encontrado resultados positivos en algunos casos pero en muchos otros negativos. ¿Cuál es el problema? ¿Es que estamos tratando con una problemática muy difícil de predecir, o que los candlesticks no están conteniendo información relevante a los fines predictivos?

Esta es la interrogante que nos proponemos aclarar con este experimento. Para ello, dejaremos de lado las imágenes de gráficos de velas y recurriremos a una metodología más tradicional de Machine Learning para abordar este problema de clasificación.

Existe toda una rama de la literatura que sigue la estrategia de utilizar algoritmos de aprendizaje automático en donde la tendencia del precio de instrumentos financieros se predicen en base a indicadores técnicos. Particularmente hay dos papers que motivan este experimento. En el paper *Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting* (Dey, Saha, Kumar, Basak, 2016) se aborda este problema predictivo y se demuestra que se pueden alcanzar resultados muy buenos con algoritmos de Machine Learning e indicadores técnicos, hace una comparación performance de los mismos y concluye que mediante XGBoost se alcanzan resultados especialmente superadores. Adicionalmente, en el paper *Predicting Stock and Stock Price Index Movement Using Trend Deterministic Data Preparation and Machine Learning Techniques* (Patel, Shah, Thakkar, Kotecha, 2014) se revela que al preprocesar los indicadores técnicos y pasarlos al modelo como variables discretas deterministas de tendencias se obtienen mejores resultados que al utilizarlos como valores numéricos continuos, que es como normalmente se presentan.

En esta sección vamos a ver un experimento donde replicaremos el problema que presentamos en el experimento 1, predecir tendencias de precios de acciones de India, pero en lugar de usar como atributo predictivo los candlesticks vamos a usar indicadores técnicos.

3.3.1 Metodología Experimento 3

a. Datos

Dataset

Continuaremos trabajando con el dataset de precios históricos de acciones Indias, para poder comparar los resultados alcanzados en los tres experimentos. Adicionalmente trabajaremos también con el dataset alternativo que armamos en el Experimento 1 con acciones de la bolsa de Estados Unidos para seguir evaluando si hay diferencias de performance según los datos que se utilicen.

Esquema de validación

Como ya sabemos, ambos datasets ya se encuentran divididos en conjuntos de entrenamiento, validación y testing respetando la temporalidad de los datos.

Ingeniería de atributos

En cuanto al procesamiento e ingeniería de los datos, en esta oportunidad estaremos trabajando principalmente con indicadores técnicos. Ya tuvimos una breve introducción conceptual a los mismos en el Experimento 1, aunque en ese caso los visualizamos de manera gráfica.

Los indicadores técnicos son formulaciones de expresiones matemáticas y estadísticas derivadas a partir de precios máximo, mínimo, de apertura y de cierre, y volumen de negociación de los instrumentos financieros. Por lo tanto, se pueden resumir como números, los cuales pueden ser calculados e incorporados fácilmente a nuestros datos.

Vamos a tener una matriz X formada por todos los atributos que calculemos y una Y a predecir. Combinaremos regresores continuos y discretos (binarios: tomarán valor 0 o 1), principalmente los podemos agrupar en tres grupos:

- VARIABLES REFERIDAS A FECHAS - BINARIAS

Como venimos trabajando en los experimentos anteriores, el campo fecha original no es pasado directamente al modelo, sino que es utilizado indirectamente mediante el esquema de validación.

Pero la fecha puede llegar a contener información que resulta importante cuando hablamos de trading. Por ejemplo, si estamos parados en un día particular de la semana o del mes, o en un cuatrimestre del año, etc.

Por lo tanto, antes de descartar la fecha, generamos varias variables binarias que toman valor 1 si se cumplen y 0 si no. Por ejemplo, la variable 'viernes' toma valor 1 si una observación pertenece a una fecha que cae en día viernes, y sino 0.

- Indicadores técnicos numéricos - continuos

Calculamos una serie de indicadores que se desprenden de la teoría tradicional del análisis técnico, según las fórmulas ya establecidas. Los mismos toman valores numéricos, por lo que estamos hablando de variables continuas. Entre ellos se encuentran retornos diarios, índices de fuerza, indicador MACD, oscilador estocástico, índice de fuerza relativa (RSI), entre otros. Adicionalmente, en los casos que es posible, normalizamos algunos indicadores para que no se vean afectados por temas de escala, por ejemplo la distancia entre las bandas de Bollinger y el precio de cierre ajustado (normalizadas por el precio).

- Variables categóricas armadas en base a indicadores técnicos - binarias

En base a los indicadores técnicos, generamos una serie de variables categóricas que indiquen ciertos comportamientos en base a los indicadores originales, ya que de esta forma se podría estar pasando información más útil al modelo, como argumentan Patel, Shah, Thakkar y Kotecha (2014). Otra razón por la cual pasamos muchos indicadores técnicos a formato categórico es que los mismos se veían afectados por la escala, y por más de que estamos trabajando con 10 acciones de alta capitalización, no resulta del todo correcto pasar datos mezclados al modelo de acciones cuyas escalas de precios y negociación pudiesen diferir, porque de este modo se podrían estar generando falsas señales.

Algunos ejemplos interesantes son las variables que indican si el retorno de ese día se encuentra por encima del promedio de los últimos 5 días, si el volumen de negociación está más alto que hace 20 días, si una media móvil de 10 días está por encima de la media móvil de 20 días, si el índice direccional es positivo, si según el ADX hay una tendencia fuerte/no hay tendencia, o si según las bandas de Bollinger el activo está sobrevendido/sobrecomprado.

Todas las variables de indicadores técnicos fueron construidas según la teoría existente, a efectos del interés de este trabajo no nos adentraremos en detalles, simplemente nos limitaremos a mencionar que todas las definiciones de estos indicadores se encuentran en la sección de Análisis Técnico de Investopedia²⁵. Gran parte de la implementación en python de la ingeniería de atributos sobre indicadores técnicos está basada en el procedimiento expuesto por Yibin Ng en su trabajo *Forecasting Stock Prices using XGBoost* (2021).

Adicionalmente incluimos el listado completo de los 125 atributos y su categoría de datos (binario/continuo) en el [Anexo](#).

²⁵ <https://www.investopedia.com/technical-analysis-basic-education-4689655>

b. Variable a predecir

Utilizaremos la construcción de la variable Y que mejor performó hasta el momento: la definición planteada en el artículo del Experimento 1.

Recordemos que en este caso, la variable a predecir, Y , es la *tendencia* del precio de una acción. El autor define la lógica de clasificación según la variación de precios de cierre de los próximos 5 días. En particular, la variable binaria Y tomará los siguientes valores:

- “Up”: Si en algún punto dentro de los próximos 5 días la variación de precio es superior a 3.7% del precio de cierre de hoy (por su tendencia alcista)
- “Down”: Si en algún punto dentro de los próximos 5 días la variación de precio es inferior a -3.7% (por su tendencia bajista)

c. Predicción

Algoritmo predictivo

Extreme Gradient Boosting, o XGBoost, es una implementación eficiente de código abierto del algoritmo de gradient boosting, una clase de algoritmo de aprendizaje automático de ensamble que puede ser utilizado para problemas de modelado predictivo de clasificación o regresión. Los ensambles se construyen a partir de modelos de árboles de decisión, los cuales se agregan de manera secuencial al ensamble y se ajustan para corregir los errores de predicción cometidos por modelos anteriores. (Brownlee, 2021)

XGBoost fue diseñado para mejorar en gran medida la eficiencia computacional (velocidad de ejecución) y el rendimiento de los modelos de aprendizaje automático, y según Dey, Saha, Kumar, y Basak (2016) es efectivamente el algoritmo de aprendizaje automático que mejor desempeña en este tipo de problema predictivo. Es por esto que elegimos construir el modelo de este experimento con dicho algoritmo.

Selección de atributos

La selección de características o atributos ayuda a reducir el overfitting, elimina características redundantes y evita confundir al clasificador. XGBoost es un algoritmo muy robusto y suele saber muy bien cómo manejarse con los atributos que tiene a disposición, y no suele ser afectado por la inclusión de atributos redundantes.

De todas maneras, como parte de nuestra experimentación, buscando refinar nuestro modelo y mejorar su desempeño, probaremos implementar una selección automática de regresores que maximicen los resultados alcanzados con los datos de validación (en términos de AUC-ROC), a partir de la importancia que XGBoost asigna a los atributos al entrenar.

Optimización de hiperparámetros

Los hiperparámetros son importantes para los algoritmos de aprendizaje automático, ya que controlan directamente el comportamiento de los algoritmos durante el entrenamiento y tienen un efecto significativo en el rendimiento de los modelos de aprendizaje automático. (Wu, Chen, Zhang, Xiong, Lei, Deng, 2019)

El objetivo de la optimización de hiperparámetros es encontrar los hiperparámetros de un algoritmo de aprendizaje automático determinado que arroje el mejor rendimiento medido en un conjunto de validación. (Koehrsen, 2018)

Hay muchos enfoques para el proceso de búsqueda de los mejores hiperparámetros, comúnmente conocido como *fine tuning* o ajuste/optimización de hiperparámetros, divididos entre métodos de búsqueda informada y no informada.

En los métodos de búsqueda no informada se prueban todas las combinaciones de hiperparámetros pertenecientes a un subespacio predefinido por el usuario, de forma repetida e independiente, es decir que cada búsqueda no informa ni predispone las búsquedas consiguientes. Ejemplos populares de estos métodos son Grid Search y Random Search. El problema con la búsqueda no informada es que es ineficiente, toma relativamente mucho tiempo construir todos los modelos, muchos de los cuales pueden resultar irrelevantes.

Los métodos de búsqueda informada resuelven este problema, ya que cada combinación de hiperparámetros que se prueba en un modelo, informa y determina los valores que se probarán en el modelo posterior. Entre este tipo de métodos, también conocidos como autoML, se encuentra el enfoque de optimización bayesiana para el ajuste de hiperparámetros que utilizaremos en nuestro experimento. Al evaluar los hiperparámetros que parecen más prometedores a partir de resultados anteriores, los métodos bayesianos pueden encontrar mejores configuraciones del modelo que Random Search o Grid Search en menos iteraciones. (Koehrsen, 2018)

Los hiperparámetros de XGBoost que optimizaremos en este experimento son:

Tabla 27: Experimento 3 - Hiperparámetros de algoritmo XGBoost

Hiperparámetro	Descripción	Rango	Valor predeterminado
alpha (alias: reg_alpha)	Término de regularización de pesos L1.	$[0; \infty]$	0
colsample_bytree	Porcentaje de columnas muestreadas (al azar) para construir un árbol.	$(0; 1]$	1
gamma (alias: min_split_loss)	Mínima reducción del error necesaria en una hoja para generar una nueva partición.	$[0; \infty]$	0
lambda (alias: reg_lambda)	Término de regularización de pesos L2.	$[0; \infty]$	1
learning_rate (alias: eta)	Proporción que aprende de cada árbol.	$[0; 1]$	0.3
max_depth	Máxima profundidad de los árboles.	$(0; \infty]$	6
min_child_weight	Cantidad mínima exigida de observaciones por hoja.	$[0; \infty]$	1
n_estimator (alias: num_boosting_rounds)	El número máximo de árboles generados y el número máximo de iteraciones.	$(0; \infty]$	100

3.3.2 Experimentación

En esta sección aplicaremos la metodología mencionada anteriormente.

Para cada set de datos (India y EE.UU.) probaremos cuatro variantes del algoritmo XGBoost:

- Modelo base con hiperparámetros predeterminados y todos los atributos
- Modelo con hiperparámetros predeterminados y selección de atributos importantes
- Modelo con hiperparámetros optimizados y todos los atributos
- Modelo con hiperparámetros optimizados y selección de atributos importantes

a. Datos de India

Experimentación: Modelo base con todos los atributos e hiperparámetros predeterminados

En primer lugar probamos el modelo de XGBoost básico con todos los atributos e hiperparámetros, para ver cómo performa.

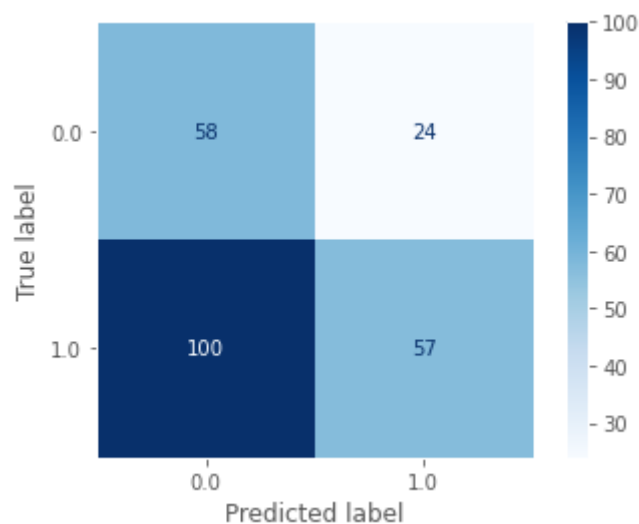
Obtenemos los siguientes resultados²⁶.

²⁶ La lista de importancia de atributos está disponible en el [Anexo](#).

Tabla 28: Experimento 3 - Resultados para datos de India + modelo con todos los atributos e hiperparámetros predeterminados

Modelo	India + todos los atributos + hiperparámetros predeterminados
train_class_distribution	
Down	47%
Up	53%
valid_class_distribution	
Down	36%
Up	64%
test_class_distribution	
Down	34%
Up	66%
Resultados Validación	
Accuracy	54%
AUC-ROC	55%
Resultados Testing	
Accuracy	48%
AUC-ROC	54%

Figura 58: Experimento 3 - Matriz de confusión para datos de India + modelo con todos los atributos e hiperparámetros predeterminados



Vemos que incluso con este modelo, que suele alcanzar buenos resultados en otros problemas de clasificación, estamos parados nuevamente en valores que rondan el azar. Veamos si al hacer algún ajuste adicional cambian los resultados.

Experimentación: Modelo con atributos selectos e hiperparámetros predeterminados

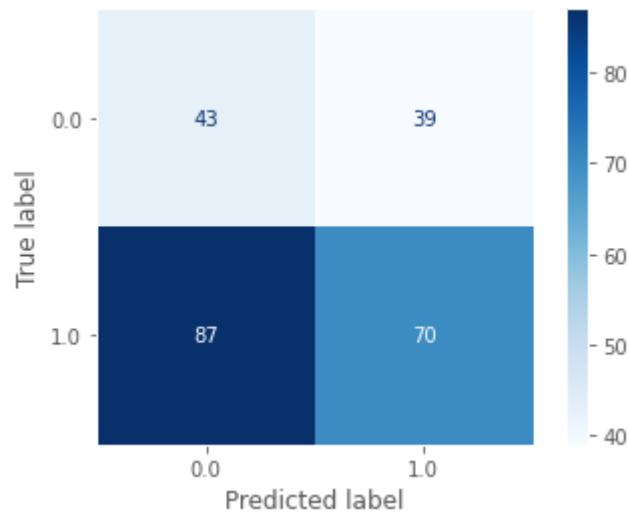
Probamos seleccionar los atributos²⁷ que mejor performance alcanzan en validación y al re-entrenar el modelo con los mismos llegamos a los siguientes resultados en testing:

Tabla 29: Experimento 3 - Resultados para datos de India + modelo con atributos selectos e hiperparámetros predeterminados

Modelo	India + atributos selectos + hiperparámetros predeterminados
train_class_distribution	
Down	47%
Up	53%
valid_class_distribution	
Down	36%
Up	64%
test_class_distribution	
Down	34%
Up	66%
Resultados Validación	
Accuracy	59%
AUC-ROC	60%
Resultados Testing	
Accuracy	47%
AUC-ROC	49%

²⁷ El detalle de los atributos seleccionados se encuentra en el [Anexo](#).

Figura 59: Experimento 3 - Matriz de confusión para datos de India + modelo con atributos selectos e hiperparámetros predeterminados



Pareciera ser que incluso empeoran los resultados de esta manera. Comprobamos que XGBoost sabe manejarse inteligentemente con todos los atributos.

Experimentación: Modelo con todos los atributos e hiperparámetros optimizados

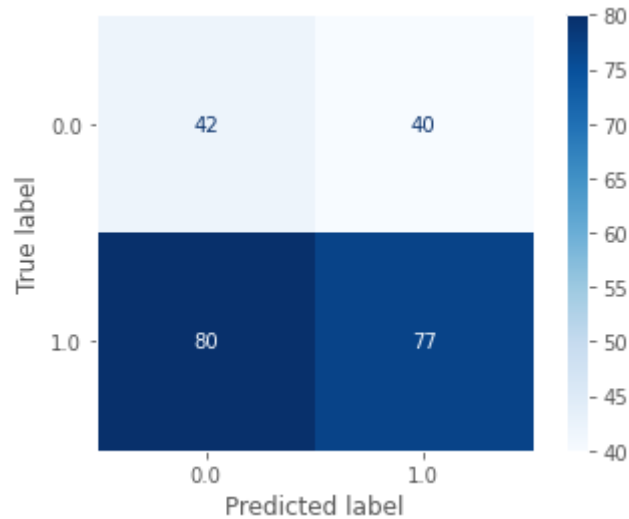
Veamos ahora qué sucede al mantener todos los atributos y optimizar hiperparámetros mediante optimización bayesiana.²⁸

²⁸ Los valores de los hiperparámetros óptimos obtenidos se encuentran detallados en el [Anexo](#).

Tabla 30: Experimento 3 - Resultados para datos de India + modelo con todos los atributos e hiperparámetros optimizados

Modelo	India + todos los atributos + hiperparámetros optimizados
train_class_distribution	
Down	47%
Up	53%
valid_class_distribution	
Down	36%
Up	64%
test_class_distribution	
Down	34%
Up	66%
Resultados Validación	
Accuracy	51%
AUC-ROC	51%
Resultados Testing	
Accuracy	50%
AUC-ROC	50%

Figura 60: Experimento 3 - Matriz de confusión para datos de India + modelo con todos los atributos e hiperparámetros optimizados



Mejora levemente la accuracy, pero decae AUC-ROC. El modelo aún no resulta útil, es equivalente al azar.

Experimentación: Modelo con atributos selectos e hiperparámetros optimizados

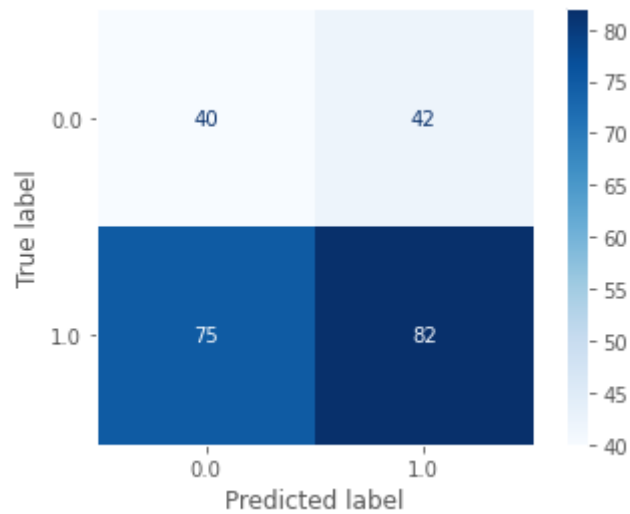
Probamos a ver si optimizando hiperparámetros²⁹ hay diferencia en la performance al utilizar simplemente los atributos selectos.

Tabla 31: Experimento 3 - Resultados para datos de India + modelo con atributos selectos e hiperparámetros optimizados

Modelo	India + atributos selectos + hiperparámetros optimizados
train_class_distribution	
Down	47%
Up	53%
valid_class_distribution	
Down	36%
Up	64%
test_class_distribution	
Down	34%
Up	66%
Resultados Validación	
Accuracy	61%
AUC-ROC	60%
Resultados Testing	
Accuracy	51%
AUC-ROC	51%

²⁹ Los valores de los hiperparámetros óptimos obtenidos se encuentran detallados en el [Anexo](#).

Figura 61: Experimento 3 - Matriz de confusión para datos de India + modelo con atributos selectos e hiperparámetros optimizados



Obtenemos una mejora casi imperceptible en ambas métricas, los resultados siguen dejando mucho que desear.

Con datos de India por ahora no logramos superar al azar o los resultados obtenidos en el Experimento 1.

b. Datos de Estados Unidos

A continuación replicamos los experimentos ya vistos para India, con los datos alternativos que armamos en el Experimento 1 con precios de acciones de Estados Unidos.

Experimentación: Modelo base con todos los atributos e hiperparámetros predeterminados

En primer lugar probamos el modelo de XGBoost básico con todos los atributos e hiperparámetros, para ver cómo performa.

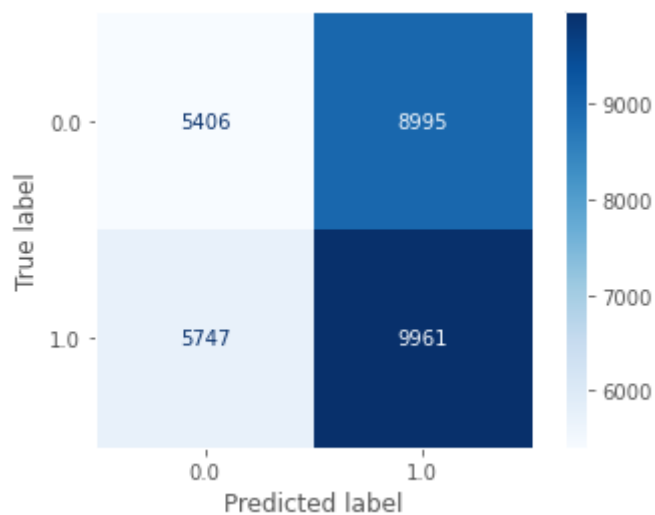
Obtenemos los siguientes resultados³⁰.

³⁰ La lista de importancia de atributos está disponible en el [Anexo](#).

Tabla 32: Experimento 3 - Resultados para datos de EE.UU. + modelo con todos los atributos e hiperparámetros predeterminados

Modelo	EE.UU.+ todos los atributos + hiperparámetros predeterminados
train_class_distribution	
Down	46%
Up	54%
valid_class_distribution	
Down	58%
Up	43%
test_class_distribution	
Down	48%
Up	52%
Resultados Validación	
Accuracy	46%
AUC-ROC	49%
Resultados Testing	
Accuracy	51%
AUC-ROC	50%

Figura 62: Experimento 3 - Matriz de confusión para datos de EE.UU.+ modelo con todos los atributos e hiperparámetros predeterminados



Nuevamente partimos de valores que rondan el azar. Veamos si al hacer ajustes adicionales cambian los resultados.

Experimentación: Modelo con atributos selectos e hiperparámetros predeterminados

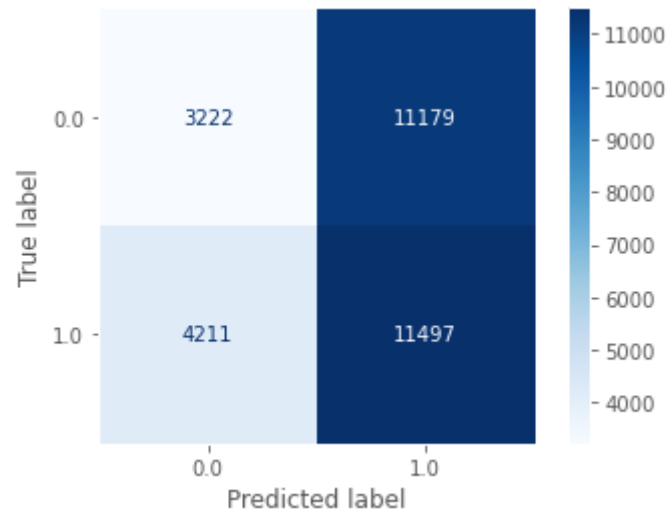
Probamos seleccionar los atributos³¹ que mejor performance alcanzan en validación y al re-entrenar el modelo con los mismos llegamos a los siguientes resultados en testing:

Tabla 33: Experimento 3 - Resultados para datos de EE.UU.+ modelo con atributos selectos e hiperparámetros predeterminados

Modelo	EE.UU.+ atributos selectos + hiperparámetros predeterminados
train_class_distribution	
Down	46%
Up	54%
valid_class_distribution	
Down	58%
Up	43%
test_class_distribution	
Down	48%
Up	52%
Resultados Validación	
Accuracy	46%
AUC-ROC	50%
Resultados Testing	
Accuracy	49%
AUC-ROC	48%

³¹ El detalle de los atributos seleccionados se encuentra en el [Anexo](#).

Figura 63: Experimento 3 - Matriz de confusión para datos de EE.UU. + modelo con atributos selectos e hiperparámetros predeterminados



Otra vez, comprobamos que empeoran los resultados de esta manera.

Experimentación: Modelo con todos los atributos e hiperparámetros optimizados

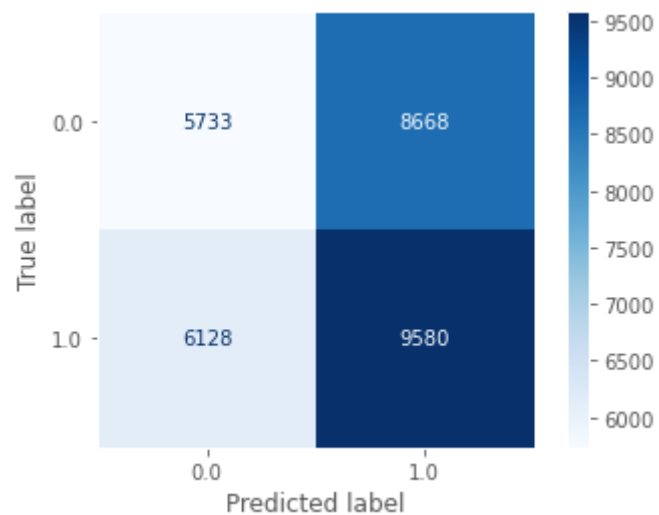
Veamos ahora qué sucede al mantener todos los atributos y optimizar hiperparámetros mediante optimización bayesiana.³²

³² Los valores de los hiperparámetros óptimos obtenidos se encuentran detallados en el [Anexo](#).

Tabla 34: Experimento 3 - Resultados para datos de EE.UU. + modelo con todos los atributos e hiperparámetros optimizados

Modelo	EE.UU. + todos los atributos + hiperparámetros optimizados
train_class_distribution	
Down	46%
Up	54%
valid_class_distribution	
Down	58%
Up	43%
test_class_distribution	
Down	48%
Up	52%
Resultados Validación	
Accuracy	47%
AUC-ROC	50%
Resultados Testing	
Accuracy	51%
AUC-ROC	50%

Figura 64: Experimento 3 - Matriz de confusión para datos de EE.UU. + modelo con todos los atributos e hiperparámetros optimizados



Mejora levemente frente a resultados anteriores, pero el modelo aún no resulta útil, es equivalente al azar.

Experimentación: Modelo con atributos selectos e hiperparámetros optimizados

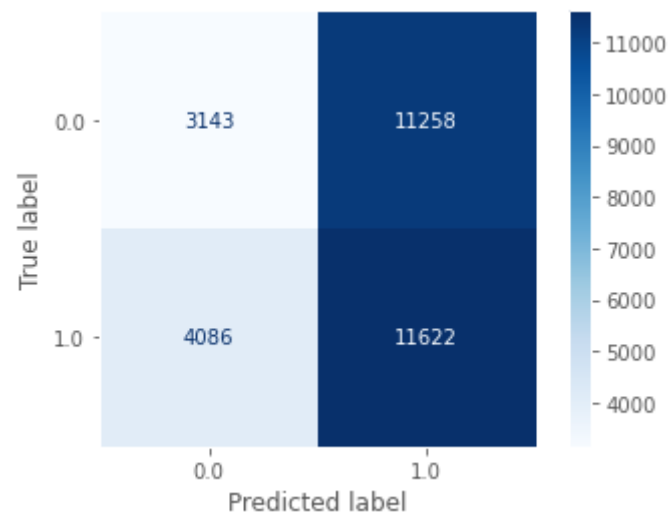
Probamos a ver si optimizando hiperparámetros³³ hay diferencia en la performance al utilizar simplemente los atributos selectos.

Tabla 35: Experimento 3 - Resultados para datos de EE.UU. + modelo con atributos selectos e hiperparámetros optimizados

Modelo	EE.UU. + atributos selectos + hiperparámetros optimizados
train_class_distribution	
Down	46%
Up	54%
valid_class_distribution	
Down	58%
Up	43%
test_class_distribution	
Down	48%
Up	52%
Resultados Validación	
Accuracy	46%
AUC-ROC	50%
Resultados Testing	
Accuracy	49%
AUC-ROC	48%

³³ Los valores de los hiperparámetros óptimos obtenidos se encuentran detallados en el [Anexo](#).

Figura 65: Experimento 3 - Matriz de confusión para datos de EE.UU. + modelo con atributos selectos e hiperparámetros optimizados



Los resultados vuelven a empeorar. Con datos de Estados Unidos tampoco logramos superar al azar o los resultados obtenidos en el Experimento 1.

3.3.3 Conclusiones

En este experimento probamos una alternativa validada por la literatura: utilizar algoritmos de Aprendizaje Automático para la predicción de tendencia de precios a partir de atributos de indicadores técnicos.

A continuación se presenta el resumen de resultados:

Tabla 36: Resultados del Experimento 3 con datos de India

Modelo				Validación		Test	
Algoritmo	Variable Y	Clasificación	Características	Accuracy	AUC-ROC	Accuracy	AUC-ROC
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Todos los atributos + Hiperparámetros predeterminados	54%	55%	48%	54%
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Atributos selectos + Hiperparámetros predeterminados	59%	60%	47%	49%
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Todos los atributos + Hiperparámetros optimizados	51%	51%	50%	50%
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Atributos selectos + Hiperparámetros optimizados	61%	60%	51%	51%

Tabla 37: Resultados del Experimento 3 con datos de Estados Unidos

Modelo				Validación		Test	
Algoritmo	Variable Y	Clasificación	Características	Accuracy	AUC-ROC	Accuracy	AUC-ROC
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Todos los atributos + Hiperparámetros predeterminados	46%	49%	51%	50%
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Atributos selectos + Hiperparámetros predeterminados	46%	50%	49%	48%
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Todos los atributos + Hiperparámetros optimizados	47%	50%	51%	50%
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Atributos selectos + Hiperparámetros optimizados	46%	50%	49%	48%

Con esta alternativa no obtenemos resultados predictivos, y de hecho llegamos a resultados peores que los alcanzados en el Experimento 1. Lo que esto sugiere es que el problema abordado es un problema complejo de predicción, pero que de alguna manera en el Experimento 1 los gráficos de velas están logrando captar algo que no está logrando captar esta metodología.

4. Discusión

4.1 Resumen de Resultados

En el trabajo presentado hemos replicado un par de trabajos que aseguraban que es posible alcanzar buenos resultados mediante el análisis de gráficos de velas con algoritmos de redes neuronales, para la predicción de la tendencia o comportamiento de los precios de activos financieros en el corto plazo. Al hacerlo, encontramos importantes errores metodológicos, y al corregirlos vimos que la capacidad predictiva de los modelos caía dramáticamente.

Una vez corregidos dichos errores, probamos diferentes alternativas de modelado para abordar el problema predictivo en cuestión. Encontramos que en algunos escenarios una alternativa de análisis de gráficos de velas con algoritmos de redes neuronales funciona, en particular el método que alcanza mejores resultados en nuestra experimentación es el análisis de gráficos de velas y volumen con bandas de Bollinger de 20 días mediante el algoritmo Resnet34, que predice la tendencia de los precios de acciones del mercado indio con accuracy de 57% y una métrica AUC-ROC de 61%, superando al azar.

Cuando probamos abordar este problema con otra metodología más tradicional, prediciendo la misma variable con un algoritmo de Aprendizaje Automático que se alimenta puramente con indicadores técnicos, no llegamos a ningún modelo que prediga bien.

Pareciera que hay cierto potencial en los gráficos de velas porque logran predecir algo que los indicadores técnicos no, pero está lejos de ser lo que se afirma en las fuentes con las que trabajamos en esta oportunidad.

Por último, dada la diferencia entre los resultados obtenidos con datos de diferentes mercados, pareciera ser que en mercados tan líquidos como Estados Unidos se cumpla más la hipótesis de mercado eficiente y efectivamente no sirvan estas metodologías, pero en los emergentes sí.

Tabla 38: Resultados de experimentación con datos de India

Modelo				Validación		Test	
Algoritmo	Variable Y	Clasificación	Características	Accuracy	AUC-ROC	Accuracy	AUC-ROC
Resnet34	Tendencia próximos 10 días	“Alternativa Torgo”	Velas + volumen	50%	50%	48%	51%
Resnet34	Tendencia próximos 5 días	“Alternativa Torgo”	Velas + volumen	47%	53%	50%	52%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen	50%	57%	53%	54%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen + media móvil de 9 días	46%	45%	57%	56%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen + media móvil de 20 días	48%	51%	57%	61%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen + medias móviles de 5 y 20 días	53%	57%	51%	51%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen + medias móviles de 4, 9 y 18 días	43%	44%	46%	51%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen + bandas de Bollinger de 20 días	59%	61%	57%	61%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen + bandas de Bollinger de 9 días	53%	57%	49%	53%
Resnet34	Comportamiento precio día siguiente	Paper Experimento 2	Velas + volumen	51%	49%	51%	50%
Resnet34	Comportamiento precio día siguiente	Paper Experimento 2	Velas + volumen + bandas de Bollinger de 20 días	51%	50%	48%	51%
CNN	Comportamiento precio día siguiente	Paper Experimento 2	Velas + volumen	45%	44%	48%	47%
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Todos los atributos + Hiperparámetros predeterminados	54%	55%	48%	54%
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Atributos selectos + Hiperparámetros predeterminados	59%	60%	47%	49%
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Todos los atributos + Hiperparámetros optimizados	51%	51%	50%	50%
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Atributos selectos + Hiperparámetros optimizados	61%	60%	51%	51%

Tabla 39: Resultados de experimentación con datos de Estados Unidos

Modelo				Validación		Test	
Algoritmo	Variable Y	Clasificación	Características	Accuracy	AUC-ROC	Accuracy	AUC-ROC
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen	49%	51%	51%	51%
Resnet34	Tendencia próximos 5 días	Original Experimento 1	Velas + volumen + bandas de Bollinger de 20 días	49%	50%	52%	53%
Resnet34	Tendencia próximos 10 días	“Alternativa Torgo”	Velas + volumen + bandas de Bollinger de 20 días	49%	51%	53%	52%
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Todos los atributos + Hiperparámetros predeterminados	46%	49%	51%	50%
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Atributos selectos + Hiperparámetros predeterminados	46%	50%	49%	48%
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Todos los atributos + Hiperparámetros optimizados	47%	50%	51%	50%
XGBoost	Tendencia próximos 5 días	Original Experimento 1	Atributos selectos + Hiperparámetros optimizados	46%	50%	49%	48%

4.2 Limitaciones y trabajo futuro

La problemática abordada tiene tantas aristas que por más de haber realizado muchas instancias variadas de experimentación, siguen quedando cosas por probar, que podemos agrupar en las siguientes categorías:

- Imágenes/gráficos

En el paper del experimento 3.2 los autores mencionan que realizaron pruebas variando las dimensiones de las imágenes, por lo cual podría hacerse algo similar para explorar si la calidad de la imagen tiene alguna incidencia en el resultado predictivo.

También sería interesante seguir probando combinaciones adicionales de indicadores técnicos que se puedan incorporar en los gráficos, para ver si se obtienen mejores resultados aún.

Se observa que dentro de los indicadores que probamos hasta ahora, las medias móviles y bandas de Bollinger de períodos de 20 días obtuvieron los mejores resultados. Esto podría dar lugar a pensar que con indicadores contruídos con períodos de tiempo más largos podrían obtenerse más resultados favorecedores, con lo cual sería lo próximo a probar en el futuro.

Adicionalmente sería aconsejable probar incorporar otros indicadores que hasta el momento no agregamos. Por ejemplo otro tipo de medias móviles, como las exponenciales, cuyos atributos resultaron bastante importantes según el algoritmo de XGBoost del último experimento.

Por último, el código de python que genera los gráficos se apalanca en la librería *mplfinance*, en principio se hizo así para facilitar la rápida implementación de indicadores técnicos a los gráficos, pero a lo largo de la experimentación se hicieron notorias ciertas limitaciones respecto a la calidad y variedad de indicadores que podíamos graficar con la misma. De cara a futuras experimentaciones debería considerarse implementar alguna otra librería que pueda resultar más flexible y completa, o alguna otra manera de generar estos gráficos.

- Variable a predecir

Como venimos viendo a lo largo del trabajo, la definición de la variable a predecir juega un rol clave en el desempeño de cualquier modelo que se quiera desarrollar, y especialmente en este caso. Vimos que predecir la dirección del precio al día siguiente fue más difícil que predecir tendencia, por lo tanto podríamos seguir probando definiciones alternativas de la tendencia, y jugar con su temporalidad también.

- Modelos

Comenzamos experimentando con algunos de los algoritmos más populares por su robustez y desempeño. Fue un buen punto de partida, pero claramente existe una amplia variedad de redes neuronales que podríamos implementar, sería interesante ver cuál se adapta mejor a este problema predictivo. El próximo paso podría ser realizar una prueba con otro algoritmo de red neuronal más profunda.

- Datos

En el trabajo terminamos enfocando el análisis en el conjunto de datos de acciones Indias para poder comparar los experimentos entre sí, y porque no observamos que hubiese grandes diferencias al trabajar con los demás conjuntos de datos. Sin embargo, no se descarta la utilidad de seguir probando otros datos para ver si con alguno se obtienen mejores resultados.

Por ejemplo, tal vez resulte útil trabajar con acciones de algún sector industrial más específico en lugar de acciones representativas de toda la economía de un país, ya que podrían darse comportamientos de precios que sean característicos de cada sector. De esto se puede desprender como trabajo futuro generar modelos predictivos independientes por sectores económicos.

También podría resultar interesante jugar con el período temporal de los conjuntos de datos, para ver si al modelo le es más útil estudiar comportamientos en un período más acotado o no a la hora de identificar patrones para predecir tendencias en el corto plazo.

4.3 Conclusión

Emprendimos este proyecto con el objetivo de ver si las técnicas modernas de procesamiento, análisis y predicción con imágenes pueden ser utilizadas con gráficos de velas para automatizar la identificación de patrones y predecir de alguna manera el comportamiento del precio de los activos financieros a corto plazo, un tema de gran interés en el mundo financiero.

Llevamos adelante una serie de experimentaciones con diversas metodologías, alternando entre varios conjuntos de datos, métodos de clasificación o armado de la variable a predecir, y algoritmos predictivos.

Finalmente contrastamos el desempeño de algoritmos de redes neuronales prediciendo tendencias de precios mediante el análisis de gráficos técnicos, contra algoritmos más tradicionales de aprendizaje automático prediciendo la misma variable en base a indicadores técnicos.

Efectivamente con el análisis de gráficos de velas obtuvimos resultados que superaron al azar y al análisis más tradicional de indicadores técnicos, lo cual sugiere que con los gráficos se estaría captando más información que resulta útil a la hora de identificar tendencias de precios de acciones.

Esta tesis refuerza la idea de que el problema de predecir el retorno de activos financieros es muy difícil, sin embargo encontramos algunas variantes que sugieren que a partir de los gráficos de velas podría plantearse un modelo con cierta capacidad predictiva. Esto parece algo viable para ser incorporado en estrategias de trading.

5. Bibliografía

- E. F. Fama, (1970). *Efficient capital markets: A review of theory and empirical work*. The Journal of Finance, Vol. 25, Papers and Proceedings of the Twenty-Eighth Annual Meeting of the American Finance Association New York, N.Y. December, 28-30, 1969, p. 383-417.
- J. López Abellán, (2018). *Hipótesis del mercado eficiente*. Economipedia. <https://economipedia.com/definiciones/hipotesis-del-mercado-eficiente.html>
- M. P. Taylor, H. Allen, (1992). *The use of technical analysis in the foreign exchange market*. Journal of International Money and Finance, Volume 11, Issue 3, Pages 304-314.
- J. Codina, (2011). *Manual de análisis técnico. 7ª ed. revisada y actualizada*. Inversor Ediciones, S.L., ISBN 978-84-15304-03-6, pág 127.
- D. Jara, (2021). "Computer Vision", ¿Qué es? ¿Qué valor genera? ¿En qué industria se aplica? LinkedIn. <https://www.linkedin.com/pulse/computer-vision-que-es-valor-genera-en-industria-se-aplica-jara?originalSubdomain=es>
- Business Wire (2016). *Computer Vision Hardware and Software Market to Reach \$48.6 Billion by 2022, According to Tractica*. Business Wire. <https://www.businesswire.com/news/home/20160620005440/en/Computer-Vision-Hardware-and-Software-Market-to-Reach-48.6-Billion-by-2022-According-to-Tractica>
- B. Marr, (2019). *7 Amazing Examples Of Computer And Machine Vision In Practice*. Forbes. <https://www.forbes.com/sites/bernardmarr/2019/04/08/7-amazing-examples-of-computer-and-machine-vision-in-practice/?sh=5b353ea41018>
- B. Goh, (2019). *CONTEXTO-China extiende el reconocimiento facial a móviles, tiendas y hogares*. Reuters. <https://www.reuters.com/article/china-tecnolog-a-idESKBN1Y60Y3>
- Ho, C., Zhao, Z., Chen, X.F. et al, (2022). *A promising deep learning-assistive algorithm for histopathological screening of colorectal cancer*. Sci Rep 12, 2222. <https://doi.org/10.1038/s41598-022-06264-x>
- R. M. I. Kusuma, T.-T. Ho, W.-C. Kao, Y.-Y. Ou, and K.-L. Hua, (2019). *Using Deep Learning Neural Networks and Candlestick Chart Representation to Predict Stock Market*, arXiv preprint arXiv:1903.12258. <https://arxiv.org/abs/1903.12258>
- S. Shah, (2020). *Identifying Candlestick Patterns using Deep Learning*. Towards Data Science. <https://towardsdatascience.com/identifying-candlestick-patterns-using-deep-learning-b7d706726874>
- JH. Chen, YC Tsai, (2020). *Encoding candlesticks as images for pattern classification using convolutional neural networks*. Financial Innovation 6, 26. <https://doi.org/10.1186/s40854-020-00187-0>

- J. Regader, (2019). *Introducción a las Velas Japonesas: significado y tipos*. Bolsa Expertos. <https://www.bolsaexpertos.com/que-son-las-velas-japonesas/>
- J. J. Murphy, (1999). *Technical Analysis of the Financial Markets*. New York Institute of Finance.
- M. Mandal, (2021). *Introduction to Convolutional Neural Networks (CNN)*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>
- S. Saha, (2018). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. Towards Data Science. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- S. Parveez, R. Iriondo, (2021). *Convolutional Neural Networks (CNNs) Tutorial with Python*. Towards AI. <https://pub.towardsai.net/convolutional-neural-networks-cnns-tutorial-with-python-417c29f0403f>
- E. Freire, S. Silva, (2019). *Redes neuronales*. Medium. <https://bootcampai.medium.com/redes-neuronales-13349dd1a5bb>
- L. Torgo, (2017). *Data Mining with R: Learning with Case Studies, Second Edition*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series.
- E. Quiroga, (2017). *Eficiencia en los mercados financieros y predicción de precios de los activos*. Ciencias Administrativas.
- E. F. Fama, M. E. Blume, (1966). *Filter Rules and Stock-Market Trading*. Journal of Business, Volume 39, Issue 1, Part 2: Supplement on Security Pricing, 226-241.
- P. Tan, M. Steinbac, V. Kumar, (2006). *Introduction to Data Mining*. Pearson.
- H. Bui, (2020). *ROC Curve Transforms the Way We Look at a Classification Problem*. Towards Data Science. <https://towardsdatascience.com/a-simple-explanation-of-the-roc-curve-and-auc-64db32d75541>
- S. Kaufman, S. Rosset & C. Perlich, (2014). *Leakage in Data Mining: Formulation, Detection, and Avoidance*. https://www.cs.umb.edu/~ding/history/470_670_fall_2011/papers/cs670_Tran_PreferredPaper_LeakingInDataMining.pdf
- J. Solawetz, (2020). *Use Resnet34 for Image Classification*. Roboflow. <https://blog.roboflow.com/custom-resnet34-classification-model/>
- C. D. Kirkpatrick II, J. R. Dahlquist, (2010). *Technical Analysis: The Complete Resource for Financial Market Technicians*. FT Press.
- E. Blanco Garzón, (2022). *Qué son los Soportes y Resistencias y cómo identificarlos*. Admiral Markets. <https://admiralmarkets.com/es/education/articles/forex-strategy/soporte-y-resistencia>
- J. Gunzberg, T. Edwards, (2020). *Why is the S&P 500 Relevant Globally?* Market Realist. <https://marketrealist.com/2018/10/why-is-the-sp-500-relevant-globally/>

- S&P Dow Jones Indices, (2021). *Clasificación Industrial Global Estándar (GICS®). Metodología.* S&P Global. <https://www.spglobal.com/spdji/es/documents/index-policias/methodology-gics-spanish.pdf>
- J. Chen, (2021). *Nasdaq 100 Index.* Investopedia. <https://www.investopedia.com/terms/n/nasdaq100.asp>
- F. J. Fabozzi, S. M. Focardi, P. N. Kolm, (2006). *Financial Modeling of the Equity Market: From CAPM to Cointegration.* John Wiley & Sons.
- S. Dey, S. Saha, Y. Kumar, S. Basak, (2016). *Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting.* Research Gate.
- J. Patel, S. Shah, P. Thakkar, K. Kotecha, (2014). *Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques.* Science Direct.
- Y. Ng, (2021). *Forecasting Stock Prices using XGBoost (Part 3/5).* Medium. <https://medium.com/ai-trading-labs/forecasting-stock-prices-using-xgboost-part-3-3-c7b13d7a84df>
- J. Brownlee, (2021). *Ensemble Learning Algorithms with Python: Make Better Predictions with Bagging, Boosting, and Stacking.* Machine Learning Mastery.
- J. Wu, X.Y. Chen, H. Zhang, L.D. Xiong, H. Lei, S.H. Deng, (2019). *Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization.* Journal of Electronic Science and Technology. P.26-40.
- W. Koehrsen, (2018). *A Conceptual Explanation of Bayesian Hyperparameter Optimization for Machine Learning.* Towards data science. <https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f>

6. Anexo

6.1 Listas de acciones utilizadas

6.1.1 Acciones de India

Lista de 10 acciones de alta capitalización de la Bolsa de Valores de India, utilizadas en el artículo analizado en el Experimento 1.

Tabla 40: Lista de acciones de India

Símbolo (ticker)
BAJFINANCE.NS
RELIANCE.NS
INFY.NS
HDFC.NS
HDFCBANK.NS
HDFCLIFE.NS
ZEEL.NS
DELTACORP.NS
ITC.NS
ASIANPAINT.NS

6.1.2 Acciones de Estados Unidos

Lista de 394 acciones de la Bolsa de Valores de Estados Unidos, que formaban parte de los índices SP500 y Nasdaq100 al 01/01/2009.

Tabla 41: Lista de acciones de Estados Unidos

Símbolo (ticker)
A
AAPL
ABC
ABT
ADBE
ADI
ADM

ADP
ADSK
AEE
AEP
AES
AFL
AIG
AIV
AIZ
AKAM
ALL
AMAT
AMD
AMGN
AMP
AMT
AMZN
AN
ANF
ANTM
AON
APA
APD
APH
ATI
ATVI
AVB
AVY
AXP
AZO
BA
BAC
BAX
BB

BBBY
BBY
BDX
BEN
BF-B
BIDU
BIG
BIIB
BK
BKR
BLL
BMY
BSX
BXP
C
CAG
CAH
CAT
CB
CBRE
CCEP
CCL
CF
CHK
CHKP
CHRW
CI
CIEN
CINF
CL
CLX
CMA
CMCSA
CME

CMI
CMS
CNP
CNX
COF
COG
COP
COST
CPB
CRM
CSCO
CSX
CTAS
CTSH
CTXS
CVS
CVX
D
DD
DE
DFS
DGX
DHI
DHR
DIS
DISH
DOV
DRI
DTE
DUK
DVA
DVN
DXC
EA

EBAY
ECL
ED
EFX
EIX
EL
EMN
EMR
EOG
EQR
EQT
ETFC
ETN
ETR
EXC
EXPD
EXPE
F
FAST
FCX
FDX
FE
FHI
FHN
FIS
FISV
FITB
FLEX
FLIR
FLR
FLS
FSLR
FTI
GD

GE
GHC
GILD
GIS
GL
GLW
GME
GNW
GOOGL
GPC
GPS
GRMN
GS
GT
GWW
HAL
HAS
HBAN
HD
HES
HIG
HOG
HOLX
HON
HPQ
HRB
HSIC
HST
HSY
HUM
IBM
ICE
IFF
IGT

ILMN
INFY
INTC
INTU
IP
IPG
ISRG
ITT
ITW
IVZ
J
JBHT
JBL
JCI
JEF
JNJ
JNPR
JPM
JWN
K
KBH
KDP
KEY
KIM
KLAC
KMB
KO
KR
KSS
L
LB
LBTYA
LEG
LEN

LH
LHX
LIN
LLY
LM
LMT
LNC
LOGI
LOW
LRCX
LSI
LUV
M
MA
MAR
MAS
MAT
MBI
MCD
MCHP
MCK
MCO
MDLZ
MDP
MDT
MET
MKC
MMC
MMM
MNST
MO
MRK
MRO
MRVL

MS
MSFT
MSI
MTB
MTW
MU
MUR
MXIM
NBL
NBR
NDAQ
NE
NEE
NEM
NI
NKE
NLOK
NOC
NOV
NSC
NTAP
NTRS
NUE
NVDA
NWL
NYT
ODP
OMC
ORCL
ORLY
OXY
PAYX
PBCT
PBI

PCAR
PCG
PDCO
PEAK
PEG
PEP
PFE
PFG
PG
PGR
PH
PHM
PKI
PLD
PM
PNC
PNW
PPG
PPL
PRU
PSA
PXD
QCOM
QRTEA
R
RF
RHI
RL
ROK
ROST
RRC
RRD
RSG
RTX

RYAAY
SBUX
SCHW
SEE
SHW
SITC
SJM
SLB
SLM
SNA
SO
SPG
SPGI
SRCL
SRE
STLD
STT
STX
STZ
SWK
SWN
SYK
SYY
T
TAP
TDC
TEL
TER
TEVA
TFC
TGNA
TGT
THC
TIF

TJX
TMO
TPR
TROW
TRV
TSN
TT
TXN
TXT
UNH
UNM
UNP
UPS
URBN
USB
VAL
VAR
VFC
VIAV
VLO
VMC
VNO
VRSN
VRTX
VTRS
VZ
WAT
WBA
WEC
WFC
WHR
WM
WMB
WMT

WU
WY
WYND
WYNN
X
XEL
XLNX
XOM
XRAY
XRX
YUM
ZBH
ZION

6.1.3 Acciones de Indonesia

Lista de 10 acciones de alta capitalización de la Bolsa de Valores de Indonesia, utilizadas en el paper analizado en el Experimento 2.

Tabla 42: Lista de acciones de Indonesia

Símbolo (ticker)
TLKM.JK
BBCA.JK
HMSP.JK
BBRI.JK
ASII.JK
BMRI.JK
UNVR.JK
GGRM.JK
BBNI.JK
UNTR.JK

6.2 Experimento 2 - Código original del paper

6.2.1 Github del autor

<https://github.com/rosdyana/Going-Deeper-with-Convolutional-Neural-Network-for-Stock-Market-Prediction>

6.2.2 Historial de ediciones

Figura 66: Experimento 2 - Resumen del historial de ediciones

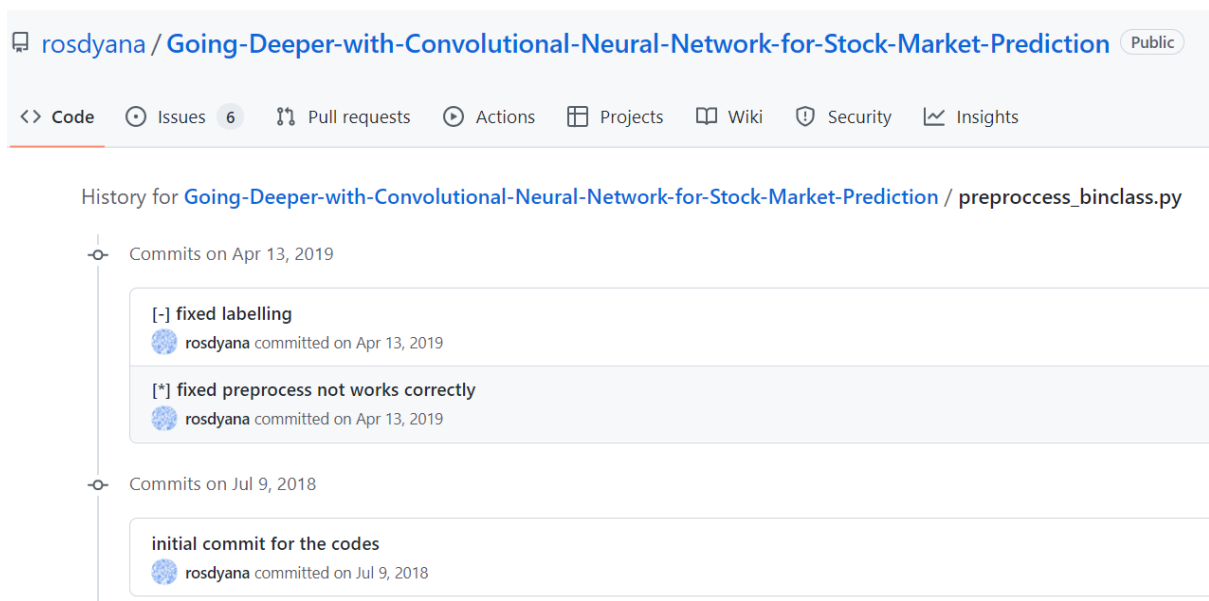


Figura 67: Experimento 2 - Commit inicial (09/07/2018)

```
102 + def createLabel(fname, seq_len):
103 +     # python preprocess.py -m createLabel -l 20 -i stockdatas/EWT_training5.csv
104 +     print("Creating label . . .")
105 +     # remove existing label file
106 +     filename = fname.split('/')
107 +     # print("{} - {}".format(filename[0], filename[1][:-4]))
108 +     removeOutput("{}_label_{}.txt".format(filename[1][:-4], seq_len))
109 +
110 +     df = pd.read_csv(fname, parse_dates=True, index_col=0)
111 +     df.fillna(0)
112 +
113 +     df.reset_index(inplace=True)
114 +     df['Date'] = df['Date'].map(mdates.date2num)
115 +     for i in range(0, len(df)):
116 +         c = df.ix[i:i + int(seq_len), :]
117 +         starting = 0
118 +         endvalue = 0
119 +         label = ""
120 +         # print("len(c) is {}".format(len(c)))
121 +         # print(c)
122 +         # label 1 if price in the end of period higher then the start point of period
123 +         # otherwise it will be 0
124 +         if len(c) == int(seq_len) + 1:
125 +             for idx, val in enumerate(c['Close']):
126 +                 # print(idx,val)
127 +                 if idx == 0:
128 +                     starting = float(val)
129 +                 if idx == len(c) - 1:
130 +                     endvalue = float(val)
131 +             if endvalue > starting:
132 +                 label = 1
133 +             else:
134 +                 label = 0
135 +             with open("{}_label_{}.txt".format(filename[1][:-4], seq_len), 'a') as the_file:
136 +                 the_file.write("{}-{},{}\n".format(filename[1][:-4], i, label))
137 +                 the_file.write("\n")
138 +     print("Create label finished.")
```

Figura 68: Experimento 2 - Edición #1: "fixed preprocess not works correctly" (13/04/2019)

```

102 102     def createLabel(fname, seq_len):
103 103         # python preprocess.py -m createLabel -l 20 -i stockdatas/EWT_training5.csv
104 104         print("Creating label . . .")
105 105         # remove existing label file
106 106         filename = fname.split('/')
107 107         # print("{} - {}".format(filename[0], filename[1][:-4]))
108 108         removeOutput("{}_label_{}.txt".format(filename[1][:-4], seq_len))
109 109
110 110         df = pd.read_csv(fname, parse_dates=True, index_col=0)
111 111         df.fillna(0)
112 112
113 113         df.reset_index(inplace=True)
114 114         df['Date'] = df['Date'].map(mdates.date2num)
115 115         for i in range(0, len(df)):
116 -         c = df.ix[i:i + int(seq_len), :]
116 +         c = df.ix[i:i + int(seq_len) + 1, :]
117 117             starting = 0
118 118             endvalue = 0
119 119             label = ""
120 -             # print("len(c) is {}".format(len(c)))
121 -             # print(c)
122 -             # label 1 if price in the end of period higher then the start point of period
123 -             # otherwise it will be 0
120 +
124 121             if len(c) == int(seq_len) + 1:
125 122                 for idx, val in enumerate(c['Close']):
126 123                     # print(idx, val)
127 -                     if idx == 0:
128 -                         starting = float(val)
129 124                     if idx == len(c) - 1:
125 +                         starting = float(val)
126 +                         if idx == len(c):
130 127                             endvalue = float(val)
131 128                     if endvalue > starting:
132 129                         label = 1
133 130                     else:
134 131                         label = 0
135 132                 with open("{}_label_{}.txt".format(filename[1][:-4], seq_len), 'a') as the_file:
136 133                     the_file.write("{}-{},{}".format(filename[1][:-4], i, label))
137 134                     the_file.write("\n")
138 135         print("Create label finished.")

```

Figura 69: Experimento 2 - Edición #2: "fixed labelling" (13/04/2019)

```

102 105 def createLabel(fname, seq_len):
103 106     # python preprocess.py -m createLabel -l 20 -i stockdatas/EWT_training5.csv
104 107     print("Creating label . . .")
105 108     # remove existing label file
106 109     filename = fname.split('/')
107 110     # print("{} - {}".format(filename[0], filename[1][:-4]))
108 111     removeOutput("{}_label_{}.txt".format(filename[1][:-4], seq_len))
109 112
110 113     df = pd.read_csv(fname, parse_dates=True, index_col=0)
111 114     df.fillna(0)
112 115
113 116     df.reset_index(inplace=True)
114 117     df['Date'] = df['Date'].map(mdates.date2num)
115 118     for i in range(0, len(df)):
116 -         c = df.ix[i:i + int(seq_len) + 1, :]
119 +         c = df.ix[i:i + int(seq_len), :]
120 +
117 121         starting = 0
118 122         endvalue = 0
119 123         label = ""
120 -
121 -         if len(c) == int(seq_len) + 1:
122 -             for idx, val in enumerate(c['Close']):
123 -                 # print(idx, val)
124 -                 if idx == len(c) - 1:
125 -                     starting = float(val)
126 -                 if idx == len(c):
127 -                     endvalue = float(val)
124 +
125 +                 if len(c) == int(seq_len)+1:
126 +                     starting = c["Close"].iloc[-2]
127 +                     endvalue = c["Close"].iloc[-1]
128 +                     # print(f'endvalue {endvalue} - starting {starting}')
128 129             if endvalue > starting:
129 130                 label = 1
130 131             else:
131 132                 label = 0
132 133             with open("{}_label_{}.txt".format(filename[1][:-4], seq_len), 'a') as the_file:
133 134                 the_file.write("{}-{},{}".format(filename[1][:-4], i, label))
134 135                 the_file.write("\n")
135 136     print("Create label finished.")

```


6.3 Experimento 3

6.3.1 Listado de atributos

Tabla 43: Experimento 3 - Lista de atributos

Atributo	Tipo de dato
january	binario
february	binario
march	binario
april	binario
may	binario
june	binario
july	binario
august	binario
september	binario
october	binario
november	binario
december	binario
monday	binario
tuesday	binario
wednesday	binario
thursday	binario
friday	binario
saturday	binario
sunday	binario
Q1	binario
Q2	binario
Q3	binario
Q4	binario
first_trading_day_of_month	binario
last_trading_day_of_month	binario

daily_ret	continuo
daily_ret_positive	binario
daily_ret_above2	binario
daily_ret_above5	binario
daily_ret_above10	binario
daily_ret_below-2	binario
daily_ret_below-5	binario
daily_ret_below-10	binario
ret_above_avg5	binario
ret_above_avg10	binario
ret_above_avg20	binario
ret_above_lag1	binario
ret_above_lag2	binario
ret_above_lag3	binario
ret_lag1_above_lag2	binario
ret_lag2_above_lag3	binario
vol_above_avg5	binario
vol_above_avg10	binario
vol_above_avg20	binario
vol_above_lag1	binario
vol_above_lag2	binario
vol_above_lag3	binario
vol_lag1_above_lag2	binario
vol_lag2_above_lag3	binario
force_index	continuo
force_index_2	continuo
force_index_13	continuo
positive_force_index	binario
positive_force_index_2	binario
positive_force_index_13	binario
sma_above20	binario
sma_above50	binario

sma_above100	binario
sma_above200	binario
sma_10above20	binario
sma_10above50	binario
sma_10above100	binario
sma_10above200	binario
sma_cut20	binario
sma_cut50	binario
sma_cut100	binario
sma_cut200	binario
sma_10cut20	binario
sma_10cut50	binario
sma_10cut100	binario
sma_10cut200	binario
sma_cut20down	binario
sma_cut50down	binario
sma_cut100down	binario
sma_cut200down	binario
sma_10cut20down	binario
sma_10cut50down	binario
sma_10cut100down	binario
sma_10cut200down	binario
ema_above20	binario
ema_above50	binario
ema_above100	binario
ema_above200	binario
ema_10above20	binario
ema_10above50	binario
ema_10above100	binario
ema_10above200	binario
ema_cut20	binario
ema_cut50	binario

ema_cut100	binario
ema_cut200	binario
ema_10cut20	binario
ema_10cut50	binario
ema_10cut100	binario
ema_10cut200	binario
ema_cut20down	binario
ema_cut50down	binario
ema_cut100down	binario
ema_cut200down	binario
ema_10cut20down	binario
ema_10cut50down	binario
ema_10cut100down	binario
ema_10cut200down	binario
macd	continuo
macd_crossover	binario
macd_crossoverdown	binario
stochastic_fast	continuo
stochastic_slow	continuo
stochastic_fastcutslow	binario
stochastic_fastcutslowdown	binario
stochastic_overs	binario
stochastic_verb	binario
rsi	continuo
rsi_ob	binario
rsi_os	binario
atr_signal	binario
atr_signaldown	binario
adx_strength	binario
adx_trendless	binario
adx_bull	binario
adx_bear	binario

bollinger_upp_dist	continuo
bollinger_low_dist	continuo
bollinger_ob	binario
bollinger_os	binario

6.3.2 Experimentación

Tras una serie de pruebas terminamos trabajando con el siguiente espacio de hiperparámetros en este experimento (en los casos donde se lleva a cabo la optimización):

Figura 70: Experimento 3 - Espacio de hiperparámetros definido

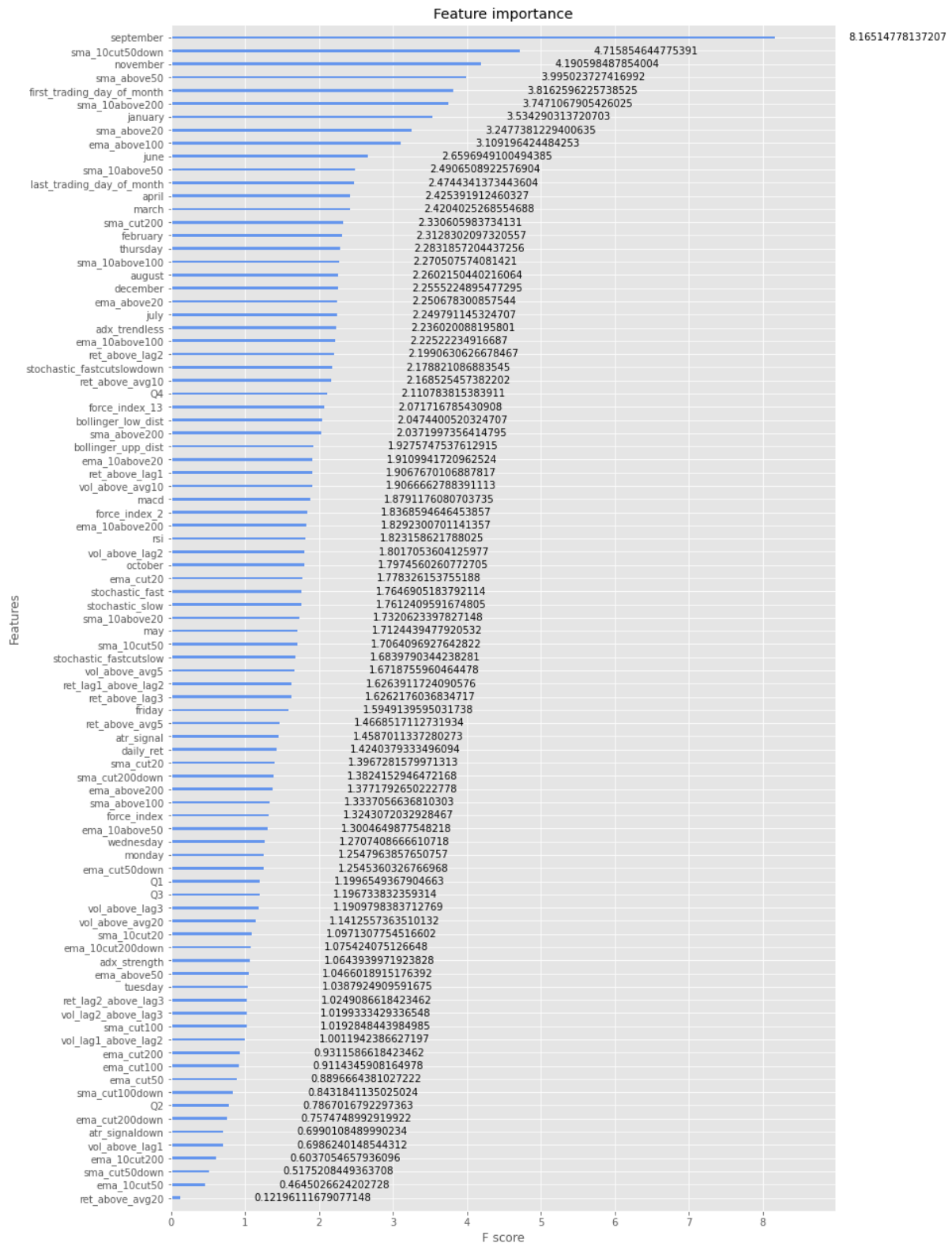
```
# Definimos espacio de hiperparámetros a evaluar
space={'colsample_bytree' : hp.uniform('colsample_bytree', 0.5,1),
      'gamma': hp.uniform ('gamma', 1,9),
      'learning_rate': hp.uniform('learning_rate', 0.01, 0.4),
      'max_depth': hp.quniform("max_depth", 3, 18, 1),
      'min_child_weight' : hp.quniform('min_child_weight', 0, 10, 1),
      'n_estimators': 500,
      'reg_alpha' : hp.quniform('reg_alpha', 10,180,1),
      'reg_lambda' : hp.uniform('reg_lambda', 0,1),
      }
```

Datos de India

Experimentación: Modelo base con todos los atributos e hiperparámetros predeterminados

Importancia de atributos:

Figura 71: Experimento 3 - Importancia de atributos del modelo base con hiperparámetros predeterminados



Experimentación: Modelo con atributos selectos e hiperparámetros predeterminados

Atributos seleccionados:

Figura 72: Experimento 3 - Atributos seleccionados del modelo base con hiperparámetros predeterminados

```
total features kept: 11
Index(['january', 'june', 'september', 'november',
      'first_trading_day_of_month', 'sma_above20', 'sma_above50',
      'sma_10above50', 'sma_10above200', 'sma_10cut50down', 'ema_above100'],
      dtype='object')
```

Experimentación: Modelo con todos los atributos e hiperparámetros optimizados

Hiperparámetros seleccionados:

```
{'colsample_bytree': 0.8306262456774398,
  'gamma': 8.389777920825608,
  'learning_rate': 0.3129390423661329,
  'max_depth': 14.0,
  'min_child_weight': 7.0,
  'reg_alpha': 10.0,
  'reg_lambda': 0.9640934368227408}
```

Experimentación: Modelo con atributos selectos e hiperparámetros optimizados

Hiperparámetros seleccionados:

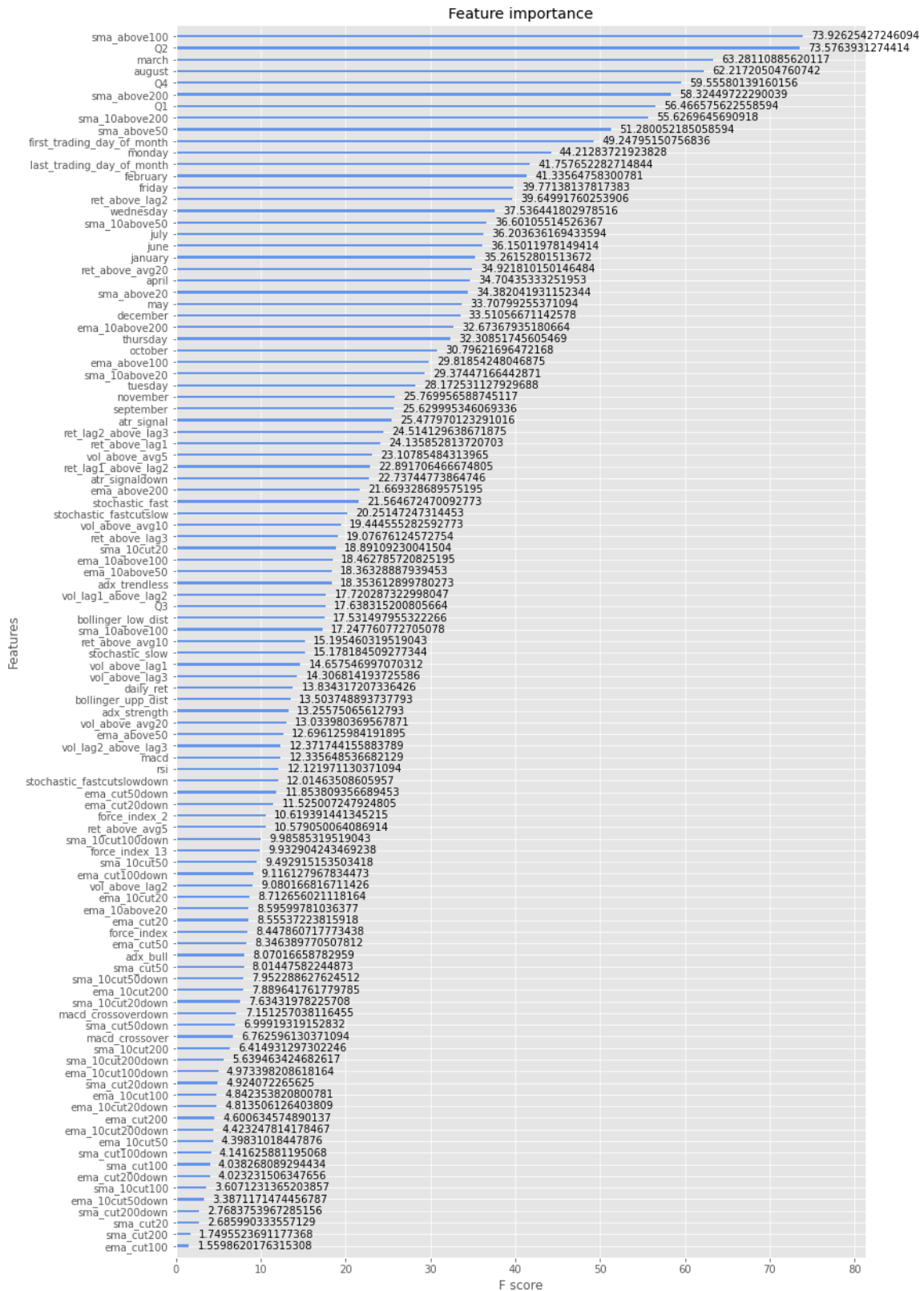
```
{'colsample_bytree': 0.6940868599938681,
  'gamma': 7.737955930217203,
  'learning_rate': 0.15730075937167928,
  'max_depth': 16.0,
  'min_child_weight': 6.0,
  'reg_alpha': 13.0,
  'reg_lambda': 0.17482380964713085}
```

Datos de Estados Unidos

Experimentación: Modelo base con todos los atributos e hiperparámetros predeterminados

Importancia de atributos:

Figura 73: Experimento 3 - Importancia de atributos del modelo base con hiperparámetros predeterminados



Experimentación: Modelo con atributos selectos e hiperparámetros predeterminados

Atributos seleccionados:

Figura 74: Experimento 3 - Atributos seleccionados del modelo base con hiperparámetros predeterminados

```
total features kept: 9
Index(['march', 'august', 'Q1', 'Q2', 'Q4', 'sma_above50', 'sma_above100',
      'sma_above200', 'sma_10above200'],
      dtype='object')
```

Experimentación: Modelo con todos los atributos e hiperparámetros optimizados

Hiperparámetros obtenidos:

```
{'colsample_bytree': 0.6292403386629198,
  'gamma': 8.325753519220914,
  'learning_rate': 0.35144753019986597,
  'max_depth': 13.0,
  'min_child_weight': 6.0,
  'reg_alpha': 25.0,
  'reg_lambda': 0.04985871047209681}
```

Experimentación: Modelo con atributos selectos e hiperparámetros optimizados

Hiperparámetros obtenidos:

```
{'colsample_bytree': 0.6085524028451332,
  'gamma': 5.060951212614636,
  'learning_rate': 0.1920011431883298,
  'max_depth': 6.0,
  'min_child_weight': 2.0,
  'reg_alpha': 68.0,
  'reg_lambda': 0.07743734593052315}
```