Universidad Siglo 21

Trabajo Final de Grado. Trabajo de Investigación en Matemática.

Carrera: Licenciatura en Matemática

Clasificación con Machine Learning para la predicción de la tendencia alcista de acciones en BYMA

Autor: Germán Cáceres Bignotti

Legajo: VLMA000611

Tutor: Mgtr. Victor Omar Palazzesi

Corrientes, noviembre de 2025

Resumen

La predicción de la tendencia alcista de los precios de las acciones en mercados volátiles y complejos, como el mercado de capitales argentino, es un desafío significativo para los inversores. Este contexto expone las limitaciones de los métodos econométricos tradicionales para capturar las no linealidades inherentes a los datos financieros. Ante esto, el Aprendizaje Automático emerge como un paradigma más robusto. El objetivo de esta investigación fue evaluar y comparar la efectividad y el rendimiento de tres algoritmos de clasificación -Regresión Logística, Árbol de Decisión y Bosque Aleatorio-, en la predicción de la tendencia alcista a 80 días bursátiles para los precios de las acciones de servicios financieros argentinos cotizantes en Bolsas y Mercados Argentinos S.A.. La investigación fue cuantitativa, no experimental, de alcance descriptivo-correlacional y longitudinal. Se emplearon datos históricos de siete empresas, procesados para generar 19 indicadores técnicos. Los modelos fueron entrenados, ajustados y validados mediante un análisis walk-forward y métricas de matriz de confusión. Los resultados revelaron un intercambio crucial entre confiabilidad y costo computacional. El algoritmo de Bosque Aleatorio demostró la máxima confiabilidad si bien a expensas de un mayor costo computacional. En contraste, el algoritmo de Regresión Logística ofreció una Precisión máxima con un costo computacional mínimo. Estos hallazgos aportan un nuevo conocimiento académico sobre la predictibilidad de la tendencia alcista del sector financiero de Bolsas y Mercados Argentinos S.A. y establecen una base para el desarrollo de herramientas de significativa capacidad predictiva en el mercado de capitales argentino.

Palabras clave: Aprendizaje Automático, algoritmos, clasificación, predicción, tendencia alcista, acciones, precios, BYMA, Regresión Logística, Árbol de Decisión, Bosque Aleatorio, servicios financieros.

Abstract

Predicting upward trends in stock prices in volatile and complex markets, such as the Argentine capital market, is a significant challenge for investors. This context highlights the limitations of traditional econometric methods in capturing the non-linearities inherent in financial data. In response, Machine Learning emerges as a more robust paradigm. The objective of this research was to evaluate and compare the effectiveness and performance of three classification algorithms -Logistic Regression, Decision Tree, and Random Forest- in predicting the upward trend over 80 trading days for the prices of Argentine financial service stocks listed on Bolsas y Mercados Argentinos S.A.. The research was quantitative, non-experimental, descriptive-correlational, and longitudinal. Historical data from seven companies were processed to generate 19 technical indicators. The models were trained, fitted, and validated using walk-forward analysis and confusion matrix metrics. The results revealed a crucial trade-off between reliability and computational cost. The Random Forest algorithm demonstrated the highest reliability, albeit at the expense of higher computational cost. In contrast, the Logistic Regression algorithm offered maximum Precision with minimal computational cost. These findings contribute new academic knowledge on the predictability of the upward trend in the financial sector of Bolsas y Mercados Argentinos S.A. and establish a basis for the development of tools with significant predictive capacity in the Argentine capital market.

Keywords: Machine Learning, algorithms, classification, prediction, upward trend, bullish trend, stocks, shares, prices, BYMA, Logistic Regression, Decision Tree, Random Forest, financial services.

Introducción

Antecedentes Relevantes y Problema de Investigación

La predicción de los precios de las acciones ha sido una tarea intrínsecamente compleja y un área de intensa investigación en la economía y las finanzas durante décadas. La volatilidad y la dificultad para anticipar la tendencia futura de los precios han representado un desafío significativo para los inversores y analistas financieros. En un blog del Fondo Monetario Internacional (FMI), Catalán, Deghi y Qureshi (2024) señalaron que la incertidumbre y las incógnitas aumentan el riesgo de volatilidad en los mercados financieros, lo que representa un desafío para la estabilidad. Mientras que en un artículo Cattlin (2023) explicó que la volatilidad es un indicador del nivel de miedo en el mercado y que la incertidumbre hace que los movimientos de precios sean erráticos e impredecibles. Otra fuente definió la volatilidad como una medida estadística de la dispersión de los rendimientos de un activo y menciona que es influenciada por factores como los indicadores económicos, los cambios políticos y normativos, y los eventos globales (*¿Qué es la volatilidad de mercado y cómo funciona?*, s.f.). Este problema ha sido más agudo en mercados emergentes como el de capitales argentino, donde la alta volatilidad se ha constituido como una característica recurrente, influenciada por factores macroeconómicos, políticos y la dinámica propia del mercado. Un análisis de UBS (*¿Por qué invertir en mercados emergentes?*, 2023) indicó que los mercados emergentes se caracterizan por ciclos económicos más volátiles, instituciones más débiles y la propensión a la inestabilidad política. Estos factores han resultado en una

mayor volatilidad general de los activos, lo que los hace una propuesta de alto riesgo pero con potencial de alta rentabilidad para los inversores (*Mercados emergentes explicados*, 2025).

Según los autores, los enfoques tradicionales para la predicción de precios de activos se basaban en el análisis de informes financieros e históricos. Sin embargo, con el aumento masivo de datos financieros, estas metodologías han sido complementadas por técnicas de machine learning (ML). En el ámbito de la fijación de precios de derivados, los modelos clásicos se han construído con suposiciones poco prácticas, lo que llevó a los profesionales a ajustar los resultados según su propio juicio. En contraste, los algoritmos de ML han tenido la capacidad de abordar más matices con muy pocas suposiciones teóricas y fueron utilizados de manera efectiva incluso en un entorno con fricciones (Tatsat, Puri y Lookabaugh, 2021, pp. 4, 114). Tradicionalmente, este desafío se ha abordado a través de métodos de análisis técnico y fundamental. Sin embargo, la creciente disponibilidad de datos históricos y el avance de la inteligencia artificial (IA) y el ML han abierto nuevas vías para desarrollar modelos predictivos más sofisticados, capaces de capturar patrones complejos que escapan a la detección humana (Pashankar, S., Shendage, J., & Pawar, J., 2024)

La predicción de los movimientos de precios en los mercados financieros constituye un desafío formidable que se encuentra en la intersección de la economía, las finanzas y la ciencia de datos. Los mercados financieros son sistemas dinámicos y complejos, caracterizados por una interacción constante de innumerables participantes que operan bajo información asimétrica y sesgos de comportamiento. Esta naturaleza intrínseca dificulta enormemente la tarea de pronosticar precios, lo que históricamente ha llevado a la creencia de que tales predicciones son, en el mejor de los casos, inútiles.

En el ámbito de la teoría financiera académica, el punto de partida fundamental para comprender la dinámica de los precios de los activos es la Hipótesis de los Mercados Eficientes (HME). Según Fama (1970), en un mercado eficiente, los precios de las acciones reflejan plenamente toda la información disponible, lo que implicaría que los cambios de precios siguen un camino aleatorio y que es imposible superar el rendimiento del mercado de forma consistente mediante el análisis técnico o fundamental.

No obstante, la evidencia empírica contemporánea y el desarrollo de las Finanzas Conductuales sugieren que los mercados no siempre operan bajo una racionalidad perfecta. Shiller (2003) argumenta que factores psicológicos, el comportamiento gregario y las respuestas emocionales de los inversores generan ineficiencias y anomalías en los precios que la teoría clásica no logra explicar. Estas ineficiencias se traducen en patrones y tendencias que, a diferencia de lo postulado por la HME, podrían ser detectables. Al respecto, Malkiel (2003) sostiene que, si bien los mercados son mayoritariamente eficientes, la existencia de episodios de irracionalidad permite que modelos computacionales avanzados identifiquen patrones.

La literatura financiera tradicional, anclada en modelos econométricos, ha tendido a simplificar la complejidad de los mercados financieros mediante el uso de técnicas estadísticas ingenuas y sobreajustadas que resultan ser impotentes en el entorno de las finanzas de alta tecnología contemporáneas (López de Prado, 2018). Estas metodologías clásicas, que a menudo se basan en supuestos de linealidad y distribuciones normales, no logran capturar las no linealidades inherentes y la alta dimensionalidad de los datos de mercado. El fracaso de estos enfoques no necesariamente implica que los mercados sean

inherentemente impredecibles, sino que las herramientas utilizadas son inadecuadas para la tarea. La necesidad de un nuevo paradigma que pueda manejar la complejidad de los mercados es evidente. Por ello, el Aprendizaje Automático (ML, por sus siglas en inglés -machine learning-) ha emergido como una solución potencial, ofreciendo un marco robusto para analizar grandes flujos de datos y descubrir patrones que escapan a las técnicas estadísticas convencionales (López de Prado, 2018).

La evolución de la predicción financiera ha pasado de la econometría tradicional a un enfoque moderno de Aprendizaje Automático, lo que representa un cambio paradigmático más que una simple actualización de herramientas. La econometría se ha centrado históricamente en modelos de regresión lineal para predecir retornos futuros, utilizando predictores como ratios financieros y variables de sentimiento de mercado. Este enfoque se basa en la construcción de modelos a partir de una teoría económica preexistente, como el Capital Asset Pricing Model (CAPM) o la Arbitrage Pricing Theory (APT) (López de Prado, 2018).

En contraste, el Aprendizaje Automático en finanzas, particularmente lo que se ha denominado como finanzas AI-first o sin modelo, opera bajo una premisa diferente. En lugar de buscar validar una teoría, este enfoque utiliza algoritmos que son inherentemente theory- and model-free para permitir que los datos revelen sus propios patrones y relaciones. La literatura sugiere que estos algoritmos de ML pueden descubrir ineficiencias estadísticas que las metodologías econométricas, con sus supuestos restrictivos, no pueden. Un ejemplo de esto se observa en la predicción de la dirección del movimiento de precios, donde los algoritmos de ML han demostrado ser capaces de superar el rendimiento de la regresión de mínimos cuadrados ordinarios, lo que apunta a la capacidad de los modelos de ML para

capturar complejidades no lineales que desafían las suposiciones de los métodos tradicionales. Este cambio de enfoque subraya la idea de que la verdadera imprevisibilidad del mercado puede ser, en realidad, un reflejo de la impotencia de las herramientas clásicas para capturar su verdadera naturaleza (López de Prado, 2018).

La aplicación de modelos tradicionales de regresión a los datos financieros se ve comprometida por las características estilizadas que exhiben estos datos y que contradicen los supuestos subyacentes de muchos de estos modelos. Un supuesto central, por ejemplo, es que los retornos de los activos siguen una distribución normal. Sin embargo, la evidencia empírica demuestra que los retornos financieros son a menudo asimétricos y, crucialmente, presentan colas pesadas o leptocurtosis (Heterocedasticidad y curtosis, 2025).

Esta leptocurtosis implica que la probabilidad de que ocurran eventos extremos (tanto positivos como negativos, conocidos como shocks de mercado) es mucho mayor de lo que predice una distribución normal (Heterocedasticidad y curtosis, 2025). Los datos también muestran heterocedasticidad, que es la variación de la volatilidad a lo largo del tiempo, y una fuerte dependencia temporal que a menudo no es lineal (Serrano Bautista y Mata Mata, 2018). Estas propiedades de los datos no son meras anomalías estadísticas, sino manifestaciones de la complejidad subyacente del mercado que los modelos econométricos lineales luchan por capturar (López de Prado, 2018).

El surgimiento de metodologías de ML ofrece una ventaja distintiva en este contexto. Dado que muchos algoritmos de ML, como los modelos basados en árboles, no hacen suposiciones a priori sobre la distribución de los datos, pueden capturar estas complejidades sin violar sus propios supuestos (López de Prado, 2018). La capacidad de trabajar con

distribuciones con colas pesadas y heterocedasticidad convierte lo que serían puntos débiles para los modelos lineales en un terreno fértil para el modelado avanzado. En lugar de ser un obstáculo, la naturaleza no normal de los retornos financieros se convierte en una oportunidad para que los modelos de ML demuestren su superioridad en la identificación de patrones y en la gestión del riesgo de cola.

El avance en la IA ha sido impulsado por innovaciones tecnológicas, desarrollos algorítmicos, la disponibilidad de big data y la creciente capacidad de cómputo, lo que ha llevado a cambios fundamentales en muchas industrias (Hilpisch, 2021, p. 11). A diferencia de la econometría tradicional, ML se ha enfocado en la capacidad predictiva de los datos para revelar patrones por sí mismos, incluso aquellos que son imperceptibles para el análisis humano (MIOTI, 2024). Este cambio de paradigma ha permitido construir modelos que se adaptan a la complejidad inherente de los mercados, como en Bolsas y Mercados Argentinos S.A. (BYMA), un mercado en particular dinámico. Si bien se han realizado estudios sobre la aplicación de técnicas de ML a la predicción de precios de acciones en diversos mercados internacionales (Gómez Plaza, S. D., 2025), existe una carencia de literatura académica en la investigación que evalúe y compare la eficacia de algoritmos de clasificación específicos en el mercado de capitales argentino.

La presente investigación busca atender esa carencia de literatura académica al evaluar y comparar la eficacia en la aplicación de tres algoritmos de clasificación de ML conocidos como Regresión Logística (RL), Árbol de Decisión (AD) y Bosque Aleatorio (BA), en la predicción de la tendencia alcista a 80 días bursátiles para los precios de las acciones de empresas argentinas del sector de servicios financieros que cotizan en BYMA. Este enfoque genera un nuevo conocimiento y aportes de valor práctico, proporcionando a los

analistas financieros e inversores una herramienta potencial para mejorar la toma de decisiones. El estudio de la aplicación de estos modelos en un nicho de mercado específico ofrece una nueva perspectiva sobre la dinámica de los precios de las acciones argentinas del sector de servicios financieros que cotizan en BYMA. La investigación se guía por los siguientes dos interrogantes:

- ¿En qué medida los algoritmos de clasificación de RL, AD y BA son efectivos para predecir la tendencia alcista a 80 días bursátiles de los precios de las acciones argentinas de servicios financieros que cotizan en BYMA?

- Entre los tres algoritmos de clasificación de ML (RL, AD y BA), ¿cuál ofrece el mejor rendimiento en términos de métricas relevantes para la predicción de la tendencia alcista a 80 días bursátiles de los precios de las acciones argentinas de servicios financieros cotizantes en BYMA?

La comparación entre el algoritmo de RL, el AD y el BA determina cuál de ellos es el más idóneo para la tarea de predicción de la tendencia alcista a 80 días bursátiles de los precios en el contexto planteado. Esto hace que la investigación se diferencie de trabajos anteriores en el área temática al centrarse en el sector financiero con oferta pública en BYMA y en un mercado geográfico específico, lo que genera conocimientos y aportes de valor práctico. El trabajo al tener potencial para ser una herramienta que aporte información y conocimiento, podría ser utilizada por analistas financieros e inversores para la toma de decisiones, permitiéndoles identificar con mayor precisión la tendencia alcista a 80 días bursátiles para los precios de las acciones argentinas de servicios financieros. Los hallazgos de este trabajo podrían ser de utilidad para los actores del mercado financiero argentino,

proporcionando una base para el desarrollo de herramientas predictivas que puedan mejorar la toma de decisiones estratégicas.

Además para el presente trabajo, la HME representa un desafío teórico, ya que si el sector de servicios financieros del mercado local (BYMA) fuera eficiente en su forma débil (Pruebas de Forma Débil: el subconjunto de información de interés son los historiales de precios -o rendimientos- pasados), el uso de datos históricos y algoritmos de ML no debería poseer valor predictivo o tendría una insignificante capacidad predictiva sobre la tendencia alcista de las acciones.

A continuación se desarrollan la parte de los antecedentes de esta investigación que se enfocan en la naturaleza de los tres algoritmos clasificación de ML (RL, AD y BA) y la evidencia empírica de su rendimiento comparativo.

La RL es una técnica fundamental que modela la probabilidad de que una instancia pertenezca a una clase, y su ventaja es su interpretabilidad y velocidad de ajuste a los datos (Abdulhafedh, 2022). Sin embargo, su capacidad para capturar las relaciones complejas y no lineales inherentes a los datos del mercado de valores es limitada, lo que a menudo la sitúa en desventaja frente a modelos más avanzados (Zheng et al., 2024). La RL es un modelo de clasificación lineal que utiliza la función logística para predecir la probabilidad de que una instancia pertenezca a una clase (alcista o bajista) (*Cómo implementar un modelo de clasificación con Python*, 2025).

El AD es un modelo no lineal que divide el espacio de características en regiones basadas en una serie de decisiones binarias (*Cómo implementar un modelo de clasificación*

con Python, 2025). También se puede definir al AD como un algoritmo de aprendizaje supervisado no paramétrico que se utiliza tanto para problemas de clasificación como de regresión. Se caracteriza por su estructura en forma de diagrama de flujo, lo que facilita su visualización e interpretación. Los AD tienen la capacidad de manejar relaciones no lineales y pueden ser ajustados a un conjunto de datos con relativa rapidez. No obstante, una de sus principales debilidades es la alta propensión al sobreajuste (overfitting), lo que hace que su rendimiento en datos no vistos sea a menudo deficiente, ya que el modelo se ha ajustado demasiado al ruido presente en los datos de entrenamiento (Abdulhafedh, 2022).

Para mitigar las limitaciones del AD, se desarrollaron modelos de conjunto (ensemble learning) como el BA -Random Forest- (Abdulhafedh, 2022). Este algoritmo se basa en la construcción de múltiples árboles de decisión y la combinación de sus salidas para mejorar la precisión predictiva y reducir el sobreajuste (Dev, 2024). Su funcionamiento se apoya en dos mecanismos claves: El bootstrapping, que crea subconjuntos aleatorios de datos para entrenar cada árbol, y la aleatoriedad de las características, que considera sólo un subconjunto aleatorio de variables en cada división de nodo (Zheng et al., 2024). La predicción final en un problema de clasificación se determina por el "voto de la mayoría" de los árboles individuales (Dev, 2024). La combinación de estos factores confiere al BA una mayor robustez y una capacidad superior para manejar datos complejos y ruidosos, lo que lo convierte en un candidato ideal para el análisis de los mercados financieros (Abdulhafedh, 2022).

Numerosos estudios han comparado el rendimiento de estos algoritmos en la predicción de precios de acciones, estableciendo un consenso emergente sobre la superioridad de los modelos de conjunto. Un estudio de Wu (2023) comparó un AD simple, un BA y XGBoost, concluyendo que el BA fue consistentemente el modelo con el mejor rendimiento,

alcanzando una precisión de predicción superior al 99% y el Error Medio Cuadrático (MSE) más bajo para tres acciones estadounidenses analizadas. De manera similar, una investigación de Lubis y Samsudin (2025) sobre el mercado de valores de Indonesia demostró la efectividad del BA, logrando una alta precisión del 98% en la predicción de movimientos de precios de acciones, lo que valida su aplicabilidad en contextos de mercados emergentes. En otro trabajo, el BA alcanzó tasas de precisión del 80-99% y se mantuvo más estable en las predicciones a largo plazo en comparación con la RL y otros algoritmos lineales (Zheng et al., 2024). En términos de efectividad general, investigaciones recientes han demostrado que modelos como la Regresión Logística, las Redes Neuronales Artificiales y las Máquinas de Vectores de Soporte son capaces de predecir los movimientos direccionales de diversos índices bursátiles con una tasa de precisión superior al 70% (Ayyildiz & Iskenderoglu, 2024). Estos hallazgos demuestran que, como regla general, los modelos que mitigan el sobreajuste y consideran la no linealidad de los datos tienden a ser más eficaces.

Actualmente la literatura también presenta resultados que sugieren que el rendimiento de un modelo no es universal y depende de las características específicas del conjunto de datos y del mercado. Por ejemplo, un análisis comparativo de la Regresión Lineal, el BA y la Regresión de Vectores de Soporte encontró que la Regresión Lineal tuvo un desempeño superior en un caso particular de predicción de precios de acciones (Pashankar et al. (2024). Otro estudio sobre la predicción del precio máximo de las acciones de Intel Corporation determinó que un Modelo de Regresión Lineal Múltiple, tras la eliminación de datos atípicos, superó al Árbol de Regresión al presentar un error significativamente menor (Carpio Vargas et al., 2023). Estos hallazgos contradictorios en los estudios resaltan la importancia del contexto y subrayan que un algoritmo que funciona bien en un mercado o con un conjunto de datos no necesariamente lo hará en otro, lo cual puede ser un indicio de la carencia de

portabilidad o sobreajuste de algunos de estos algoritmos entrenados en los estudios publicados. El rendimiento no es una característica intrínseca del algoritmo, sino una función de su idoneidad para las particularidades del entorno.

Objetivo General y Específicos

Si bien se han realizado comparaciones exhaustivas en mercados como el de Estados Unidos e Indonesia, entre otros, el análisis de estos antecedentes revela que la literatura carece de un estudio que evalúe y compare la efectividad de los algoritmos de RL, AD y BA específicamente para la predicción de la tendencia alcista a 80 días bursátiles de los precios en el mercado de capitales argentino, y más aún, en el sector de empresas de servicios financieros con oferta pública en BYMA. La alta volatilidad de este mercado representa un entorno de prueba único para la robustez de estos modelos, y el estudio de un sector específico permite generar un conocimiento detallado y práctico que no se obtendría de una investigación de carácter más general. Por lo tanto, la presente investigación no sólo contribuye a la literatura académica al atender esta carencia específica, sino que también ofrece un aporte de valor práctico al proporcionar una base para el desarrollo de herramientas predictivas que pueden mejorar la toma de decisiones estratégicas de analistas e inversores en el mercado de capitales argentino.

Bajo el marco de la HME y las Finanzas Conductuales, el presente manuscrito se posiciona en la intersección de estas dos corrientes. Al proponer el uso de modelos de clasificación para predecir la tendencia alcista del sector de servicios financieros en BYMA, se asume implícitamente que el mercado argentino presenta ciertos grados de ineficiencia informativa o patrones conductuales que pueden ser capturados mediante el aprendizaje

automatizado. De este modo, tácitamente, no sólo se busca evaluar los modelos de clasificación de ML, sino también explorar si, mediante el procesamiento avanzado de variables financieras, es posible poseer una significativa capacidad predictiva sobre la tendencia alcista de las acciones en el contexto del mercado bursátil local.

En este contexto, el objetivo general de esta investigación es realizar un análisis comparativo de la efectividad y el rendimiento de los tres algoritmos de clasificación (RL, AD y BA) en la predicción de la tendencia alcista a 80 días bursátiles para los precios de las acciones de empresas argentinas de servicios financieros con oferta pública en BYMA. Para lograrlo, se proponen los siguientes objetivos específicos:

1. Recolectar los datos históricos de las acciones de empresas argentinas de servicios financieros con oferta pública en BYMA.

2. Construir los predictores y procesar los datos históricos a utilizar en los tres modelos.

3. Implementar y entrenar los modelos de clasificación de machine learning de Regresión Logística, Árbol de Decisión y Bosque Aleatorio, utilizando los datos procesados.

4. Validar los modelos de clasificación de machine learning de Regresión Logística, Árbol de Decisión y Bosque Aleatorio, utilizando los datos procesados.

5. Comparar la efectividad y el rendimiento de los tres modelos entrenados y validados mediante las métricas, tablas, gráficos y matrices de confusión, para identificar el

algoritmo más adecuado para la predicción de la tendencia alcista a 80 días bursátiles de los precios de las acciones de empresas argentinas de servicios financieros con oferta pública en BYMA.

Métodos

1-Alcance

El alcance de esta investigación se centra en la aplicación de tres algoritmos de clasificación de ML (RL, AD y BA), en la predicción de la tendencia alcista a 80 días bursátiles para los precios de las acciones de empresas argentinas del sector de servicios financieros que cotizan en BYMA y sus datos históricos de negociación diaria se encuentran disponibles en la API de Data912. Por ende la investigación tiene un alcance descriptivo-correlacional, buscando primero describir y comparar los rendimientos de los algoritmos de RL, el AD y el BA, para luego determinar cuál de ellos es el más idóneo para la tarea de predicción de la tendencia alcista a 80 días bursátiles de los precios en este contexto.

2-Enfoque

El enfoque de la investigación es cuantitativo, basándose en la recolección de datos numéricos, procesamiento y su análisis estadístico para medir y comparar, teniendo los modelos entrenados y validados. Se ha adoptado un diseño no experimental, ya que el investigador no manipula las variables, sino que observa y analiza los fenómenos financieros en su contexto natural, tal como se presentan en los datos históricos del mercado. Adicionalmente, el estudio es de carácter longitudinal, ya que se apoya en la observación de

variables a lo largo del tiempo para construir predictores/indicadores. Esta ingeniería de características captura la evolución histórica de la serie, lo cual es inherente a la naturaleza del problema de clasificación basado en series temporales, a pesar de que el modelo final no modele la secuencia estricta.

3-Diseño y Tipo de la Investigación

El diseño de la investigación será no experimental, de alcance descriptivo-correlacional y de tipo longitudinal. El enfoque de la investigación es cuantitativo, dado que se basará en la medición de variables numéricas y en el uso de técnicas de análisis de datos para cuantificar las relaciones entre variables predictivas y la variable de resultado. Este enfoque permitirá una evaluación objetiva del rendimiento de los modelos mediante métricas de desempeño.

En la Tabla 1 se describe la planificación de la investigación, ordenada según los objetivos específicos, detallando las actividades respectivas a cada objetivo, los resultados que deben obtenerse una vez finalizadas las tareas y el análisis y/o decisión que se recomienda tomar.

Tabla 1. *Planificación de la ejecución de la investigación.*

Objetivos específicos	Actividades	Resultados	Análisis y decisión
1-Obtener datos históricos de las acciones de empresas argentinas de servicios financieros con oferta pública en	1.A-Desarrollar una función para persistir los datos mediante la consulta a la API de Data912. 1.B-Persistir los	1.1-Función de persistencia de datos, en producción. 1.2-8 datasets persistidos, de la	Si la calidad de los datos no es la requerida se debe ajustar el/los datasets para que permitan cumplimentar con

Objetivos específicos	Actividades	Resultados	Análisis y decisión
BYMA.	datasets requeridos mediante la función. 1.C-Realizar data mining sobre los datos revisando y confirmando la calidad de los datos	negociación diaria de los precios y volúmenes, cada uno perteneciente a una empresa de servicios financieros en el período que empieza el 2 de enero de 2001 y finaliza el 12 de septiembre de 2025.	los requerimientos de los objetivos de la investigación, en otro caso se da por cumplido el objetivo 1 y se continúa con las actividades del objetivo 2.
2-Construir los predictores y procesar los datos históricos a utilizarse en los tres modelos.	2.A-Desarrollar la función de los predictores que calcule los valores de estos para cada día del dataset y los persista. 2.B-Desarrollar la función que define el objetivo a predecir (tendencia alcista a 80 días bursátiles de los precios) y persista los targets. 2.C-Desarrollar la función que crea y persiste las muestras a entrenar y validar por los tres modelos de clasificación. Además esta función debe permitir normalizar los valores calculados por los predictores. 2.D-Persistir los datasets de las muestras normalizadas a entrenar y validar	2.1-Función de los predictores, en producción. 2.2-Función que define el objetivo a predecir, en producción. 2.3-Función que crea y persiste las muestras a entrenar y validar por los tres modelos de clasificación, y normaliza los valores calculados por los predictores, en producción. 2.4-Datasets de entrenamiento y validación creados, normalizados y persistidos.	Si los datos fueron persistidos por las tres funciones desarrolladas en el objetivo 2, de acuerdo a lo esperado, se da por cumplido el objetivo 2 y se continúa con las actividades del objetivo 3.

Objetivos específicos	Actividades	Resultados	Análisis y decisión
	por los tres modelos de clasificación.		
3-Implementar y entrenar los modelos de clasificación de machine learning de Regresión Logística, Árbol de Decisión y Bosque Aleatorio, utilizando los datos procesados.	3.A-Desarrollar la función que entrena, ajusta los hiper-parámetros y persiste los tres modelos de clasificación entrenados con los datos procesados. 3.B-Persistir los tres modelos de clasificación entrenados y con los hiper-parámetros ajustados.	3.1-Función que entrena, ajusta los hiper-parámetros y persiste los tres modelos de clasificación entrenados, en producción. 3.2-Los tres modelos de clasificación entrenados y con los hiper-parámetros ajustados, persistidos.	Si los tres modelos entrenados y ajustados fueron persistidos por la función desarrollada en el objetivo 3, de acuerdo a lo esperado, se da por cumplido el objetivo 3 y se continúa con las actividades del objetivo 4.
4-Validar los modelos de clasificación de machine learning de Regresión Logística, Árbol de Decisión y Bosque Aleatorio, utilizando los datos procesados.	4.A-Desarrollar la función que predice en base a la muestra de validación y persiste las predicciones para los tres modelos de clasificación. 4.B-Esta función también debe calcular y persistir las métricas, tablas, gráficos y matrices de confusión necesarias para evaluar el rendimiento de los tres modelos de clasificación. 4.C-Persistir las métricas, tablas, gráficos y matrices de confusión de los	4.1-Función que predice en base a la muestra de validación y persiste las predicciones, métricas, tablas, gráficos y matrices de confusión, en producción. 4.2-Los tres modelos de clasificación validados y sus métricas, tablas, gráficos y matrices de confusión, persistidas.	Si las predicciones de los tres modelos validados y sus métricas, tablas, gráficos y matrices de confusión fueron persistidas por la función desarrollada en el objetivo 4, de acuerdo a lo esperado, se da por cumplido el objetivo 4 y se continúa con las actividades del objetivo 5.

Objetivos específicos	Actividades	Resultados	Análisis y decisión
	tres modelos de clasificación entrenados y validados, luego del ajuste de sus hiper-parámetros.		
5-Comparar la efectividad y el rendimiento de los tres modelos entrenados y validados mediante las métricas, tablas, gráficos y matrices de confusión, para identificar el algoritmo más adecuado para la predicción de la tendencia alcista a 80 días bursátiles de los precios de las acciones de empresas argentinas de servicios financieros con oferta pública en BYMA.	5.A-Hacer el análisis de métricas, tablas, gráficos y matrices persistidas y describir las diferencias o similitudes entre la efectividad y rendimiento de los tres modelos de clasificación entrenados y validados.	5.1-Los tres modelos de clasificación entrenados, ajustados y validados son comparados según la proximidad y el tipo de proximidad (positiva / negativa) de los sesgos (alcistas) de la muestra utilizada para validar y los resultados de las predicciones de los modelos entrenados y ajustados. 5.2-Los tres modelos de clasificación entrenados, ajustados y validados son comparados según la proximidad y el tipo de proximidad (positiva / negativa) de los errores tipo I y errores tipo II de la muestra utilizada para validar y los resultados de las predicciones de los modelos entrenados y ajustados. 5.3-Los tres modelos de clasificación entrenados,	Los modelos validados con predicciones que tengan un Sesgo (Alcista) más próximo al Sesgo (Alcista) de la muestra utilizada para validar, serán los modelos con mayor efectividad y rendimiento. Si los Sesgos (Alcistas) de las predicciones de los modelos tienen entre sí, una proximidad absoluta menor al cinco por ciento (5%), los modelos con mayor efectividad y rendimiento serán aquellos que tengan la mayor cantidad de métricas (Positivos detectados -Aciertos Alcistas-, Exactitud, Precisión, Sensibilidad y F1-Score) con mayores valores o mejores según lo que signifique cada una. Cuando las diferencias absolutas entre las seis métricas nombradas (Sesgo (Alcista),

Objetivos específicos	Actividades	Resultados	Análisis y decisión
		ajustados y validados son comparados según la cantidad de Aciertos Alcistas, Exactitud, Precisión, Sensibilidad y F1-Score de la muestra utilizada para validar y los resultados de las predicciones de los modelos entrenados y ajustados.	Positivos detectados -Aciertos Alcistas-, Exactitud, Precisión, Sensibilidad y F1-Score), sean menores al uno por ciento (1%) se considerará que los modelos comparados tienen una efectividad y rendimiento similar. Los Errores tipo I y tipo II se observarán de forma descriptiva para comprender la relación que existe con el sesgo de las predicciones y el resto de las métricas.

4-Población

La población de este estudio se define como el conjunto de todas las empresas de servicios financieros que cotizan en BYMA al momento de realizar el estudio, debido a que existen empresas del sector que han retirado la oferta pública de acciones, como también existe otro grupo que tiene un menor historial de datos históricos debido al momento en que realizaron su oferta pública de acciones o la inexistencia de negociación diaria de acciones en determinados momentos desde su oferta pública inicial de acciones.

La oferta pública de acciones en BYMA permite a los inversores comprar y vender la participación en el mercado de las empresas de servicios financieros, las cuales son principalmente entidades bancarias y el propio BYMA. En total son ocho empresas de

servicios financieros cuyas acciones cotizan en BYMA, las cuales tienen la siguiente información con respecto a su negociación diaria:

- la fecha de negociación,
- el precio de apertura,
- el precio de cierre,
- el precio mínimo,
- el precio máximo y
- el volumen de acciones negociadas.

A continuación se listan estas ocho empresas, según el rubro financiero al cual pertenecen, por el nombre de su razón social y ticker identificatorio de la especie cotizante:

❖ Bancos:

1. Banco BBVA Argentina S.A. (BBAR)
2. Banco Hipotecario S.A. (BHIP)
3. Banco Macro S.A. (BMA)
4. Banco Patagonia S.A. (BPAT)
5. Banco Supervielle S.A. (SUPV)

❖ Otros servicios financieros:

6. Bolsas y Mercados Argentinos S.A. (BYMA): La propia compañía que opera el mercado bursátil tiene oferta pública de sus acciones.
7. Grupo Financiero Galicia S.A. (GGAL)
8. Banco de Valores S.A. (VALO)

Estas empresas representan el universo de activos en el mercado de capitales argentino sobre los cuales se desea realizar la predicción de la tendencia alcista a 80 días bursátiles para los precios.

5-Muestra y Participantes

El muestreo de esta investigación es no probabilístico intencional, donde la muestra está constituida por un subconjunto de la población de empresas argentinas del sector de servicios financieros con oferta pública en BYMA. La selección de la muestra se basó en criterios de disponibilidad de datos históricos de negociación diaria de acciones para asegurar su representatividad y la calidad de la información. Debido a que la API de Data912 no tiene una base histórica del Banco de Valores S.A. (VALO), y otras fuentes de acceso libre y gratuito presentan problemas de calidad de datos, no se utilizó la misma para el entrenamiento y validación de los modelos de clasificación. Al mismo tiempo no es relevante la no inclusión de la misma en la muestra porque a partir de 2019, Banco de Valores S.A. y su sociedad controlante, Grupo Financiero Valores S.A., iniciaron un proceso de reorganización societaria, mediante el cual Banco de Valores S.A. absorbería a su sociedad controlante. En el marco del mencionado proceso, Banco de Valores S.A. solicitó autorización para ingresar al régimen de la oferta pública por acciones, la cual fue otorgada por la Comisión Nacional de Valores (CNV) el 3 de mayo de 2021. En noviembre de 2021 la CNV aprobó, sujeto al cumplimiento de ciertas condiciones, la fusión por absorción de Banco de Valores S.A. con su sociedad controlante Grupo Financiero Valores S.A. Cumplidas dichas condiciones, la fecha de efectiva reorganización fue el 3 de enero de 2022. Todos estos eventos afectaron la calidad de la información en esos períodos siendo que las bases históricas que se ha podido

acceder inician el 9 de julio de 2007 y no consideraron estos eventos en el ajuste de precio cierre.

Para este estudio, los participantes son las siete acciones de las empresas del sector financiero seleccionadas para la muestra:

1. Banco BBVA Argentina S.A. (BBAR),
2. Banco Hipotecario S.A. (BHIP),
3. Banco Macro S.A. (BMA),
4. Banco Patagonia S.A. (BPAT),
5. Banco Supervielle S.A. (SUPV),
6. Bolsas y Mercados Argentinos S.A. (BYMA),
7. Grupo Financiero Galicia S.A. (GGAL).

Dado que estas empresas tienen oferta pública de parte de su capital accionario, no involucra sujetos humanos, y por lo tanto no requiere la obtención de consentimiento informado ni la aprobación de un comité de ética. La investigación se enfocará en la comparación de la efectividad y el rendimiento de los tres algoritmos de clasificación (RL, AD y BA) en la predicción de la tendencia alcista a 80 días bursátiles para los precios de estos activos.

La muestra de datos se recolectó para el período comprendido entre el 2 de enero de 2001 y el 12 de septiembre de 2025. Los tipos de datos históricos de negociación diaria obtenidos de la API de Data912 incluyeron:

1. fecha de negociación,
2. precio de apertura,
3. precio máximo,

4. precio mínimo,
5. precio de cierre,
6. volumen de acciones negociadas,
7. ratio de retorno diario (tanto por uno)

6-Instrumentos

Para la recolección, procesamiento y análisis de los datos, se emplearon diversas herramientas y recursos.

Herramientas de Recopilación de Datos:

- API de Data912: Se utilizó la API de Data912 para obtener series de tiempo de datos históricos de empresas argentinas de servicios financieros con oferta pública en BYMA.
- Requests (v. 2.32.5): La librería de Python Requests (Reitz, 2023) se usó para interactuar con la API de Data912 y realizar las solicitudes HTTP para la obtención de los datos.

Entorno de Programación y Librerías:

El análisis se realizó en un entorno de programación Python (v. 3.13.7), utilizando las siguientes librerías:

- Pandas (v. 2.3.3): Empleada para la manipulación y el análisis de datos (Pandas Development Team, 2025), incluyendo la gestión de estructuras de datos como DataFrames.
- NumPy (v. 2.3.3): Utilizada para operaciones matemáticas y numéricas eficientes (NumPy Developers, 2024).

- Scikit-learn (v. 1.7.2): Esta librería de Aprendizaje Automático fue central para el proyecto (Pedregosa et al., 2011). Se usaron los siguientes módulos:
    - sklearn.preprocessing.StandardScaler: para la estandarización de las variables.
    - sklearn.linear_model.LogisticRegression: para la implementación del algoritmo de Regresión Logística.
    - sklearn.tree.DecisionTreeClassifier: para la implementación del algoritmo de Árbol de Decisión.
    - sklearn.ensemble.RandomForestClassifier: para la implementación del algoritmo de Bosque Aleatorio.
    - sklearn.metrics: Se utilizaron confusion_matrix, ConfusionMatrixDisplay y sus métodos, para evaluar el rendimiento de los modelos de clasificación.
    - sklearn.model_selection.RandomizedSearchCV: para encontrar la combinación óptima de hiper-parámetros de los modelos de clasificación.
    - sklearn.model_selection.GridSearchCV: para la optimización en forma exhaustiva de los hiper-parámetros.
- Matplotlib (v. 3.10.6): Utilizada para la visualización de datos (The Matplotlib Development Team, 2025), incluyendo gráficos y la representación de las matrices de confusión.
- IPython.display.Image v. 9.6.0 (Pérez & Granger, 2007) y graphviz v. 0.21 (Ellson et al., 2002): Empleadas en conjunto para la visualización del Árbol de Decisión.
- os y pickle: La librería estándar de Python os se usó para la gestión de archivos, mientras que pickle se utilizó para serializar y guardar los modelos entrenados, facilitando su posterior uso y análisis.
- Scipy (v. 1.16.2): Utilizada para asegurar que la búsqueda de los hiper-parámetros sea eficiente (Virtanen et al., 2020). Se usó el siguiente módulo:

    - stats: en este contexto, para definir las distribuciones de probabilidad necesarias para el parámetro param_distributions de RandomizedSearchCV.
- Seaborn (v. 0.13.2): Esta librería se usó para mostrar correlaciones entre múltiples variables y crear gráficos (Waskom, 2024).

7-Análisis de los Datos

Preprocesamiento de Datos y Variables

La función de Python desarrollada calcula 19 indicadores a utilizarse como predictores de los tres modelos de clasificación y una variable de referencia para la predicción de la tendencia alcista a 80 días bursátiles de los precios de las acciones de empresas argentinas del sector de servicios financieros que cotizan en BYMA. Esta variable es identificada como la ventana de tiempo a predecir por los modelos (en inglés se la referencia con las letras “fw”, relativas a "forward”, que en finanzas se relaciona con flujos futuros).

Retorno Futuro Normalizado (fw). Esta es la variable de referencia para predicción y se utiliza como el valor objetivo que el modelo intentará predecir, indicando el retorno que se obtendría al mantener la posición durante un periodo igual a $ventana$ (80 días bursátiles).

C_t Precio de cierre en el período t.

$C_{t+ventana}$ Precio de cierre $ventana$ períodos en el futuro.

$ventana$ Número de períodos de la ventana de predicción (80 días bursátiles).

$$fw_t = \frac{C_{t+ventana}}{C_t} - 1 \qquad (1)$$

Los criterios de selección para los indicadores, considerando que el objetivo es comparar tres algoritmos de clasificación (AD, RL y BA) para predecir la tendencia alcista a 80 días bursátiles ($fw > 0$) de los precios de las acciones de servicios financieros de BYMA, se basan en la relevancia, la no redundancia y la capacidad predictiva de cada indicador.

Criterio de Selección de la Variable Objetivo. El primer y más crucial paso es la correcta definición de la variable a predecir (Y).

Variable Objetivo (Y): Tendencia Alcista (Binaria)

Base: Se utiliza el indicador fw (Retorno futuro normalizado a 80 días bursátiles).

Criterio: Se convierte el retorno futuro continuo (fw) en una variable binaria de clasificación:

$$Y_t = \begin{cases} 1 \text{ (Alcista)} & \text{si } fw_t > 0 \\ 0 \text{ (Bajista/Lateral)} & \text{si } fw_t \leq 0 \end{cases} \quad (2)$$

Los tres algoritmos de clasificación requieren una variable dependiente categórica o binaria. Predecir la dirección (alcista) es el objetivo directo del problema.

Se ha elegido una ventana de 80 días bursátiles buscando reducir la autocorrelación de errores y multicolinealidad entre la variable objetivo y los indicadores, lo cual puede reducir y/o condicionar el rendimiento predictivo del algoritmo de clasificación de RL. Ninguno de los indicadores contiene ventanas de 80 días bursátiles, sólo tienen ventanas de 5, 10, 20, 60 y 120 días. La multicolinealidad también afecta al algoritmo del AD y en menor medida al BA.

A continuación se describen los indicadores, sus fórmulas y criterios de selección, recordando que la muestra de datos que se recolectó contiene las series de tiempo de precio

de apertura, máximo, mínimo, cierre y volumen diarios. Para reducir la autoexplicación de los indicadores de precios, como la autocorrelación de errores y multicolinealidad entre la variable objetivo y los indicadores, se utilizó para la construcción de este tipo de indicador la media diaria del precio de apertura, máximo, mínimo y cierre ($meanOHLC$) en lugar de solamente el precio de cierre, debido a que la tendencia alcista que se trata de predecir depende directamente de este.

A-Indicadores de Precio (Tendencia -Trend Following-). Los indicadores basados en precios miden la posición relativa de la tendencia a diferentes plazos, esencial para evaluar la dinámica del mercado. De este tipo de indicador se usaron 10 indicadores de cruces de medias móviles simples de la media OHLC, los cuales se detallan en el siguiente subtítulo:

A.1-Cruce de Medias Móviles Simples (SMA) de la Media OHLC ($cruce\_sma\_p_1\_p_2$***).*** Este indicador representa la razón entre una media móvil simple (SMA) de corto período (p_1) y una SMA de largo período (p_2), normalizada al restarle 1 (uno). Se utiliza para medir la tendencia relativa. Un valor positivo indica que la media de corto plazo está por encima de la de largo plazo (tendencia alcista) y viceversa. La SMA se calcula sobre la serie de la media diaria del precio de apertura, máximo, mínimo y cierre ($meanOHLC$).

$M_{OHLC,\,i}$ Valor de $meanOHLC$ en el período i.

$SMA(M_{OHLC},\ p)$ Media móvil simple de $meanOHLC$ con período p.

p_1 Período corto (Se calculó para los períodos de 5, 10, 20 y 60 días).

p_2 Período largo (Se calculó para los períodos de 10, 20, 60 y 120 días).

Fórmula de la Media Móvil Simple (SMA):

$$SMA(M_{OHLC}, p)_t = \frac{1}{p} \sum_{i=t-p+1}^{t} M_{OHLC,i} \quad (3)$$

Fórmula del Indicador $cruce\_sma\_p_1\_p_2$:

$$cruce\_sma\_p_1\_p_2 = \frac{SMA(M_{OHLC}, p_1)_t}{SMA(M_{OHLC}, p_2)_t} - 1 \quad (4)$$

Se seleccionaron cruces que cubren la relación a corto-medio plazo (ej. 5/20, 10/60) y la relación corto-largo plazo (ej. 5/120, 20/120). Este indicador captura directamente la relación de tendencia de la media OHLC. Un valor positivo (cruce alcista) sugiere un impulso que, sí es fuerte, podría sostenerse por 80 días bursátiles. Proporciona diferentes visiones del mercado.

Es importante considerar que algunos cruces están correlacionados (ej. 5/10 y 5/20 son casi idénticos). La correlación de las variables (indicadores y variable objetivo) puede verse con mayor detalle en la Tabla 2.

La RL y el AD son sensibles a la multicolinealidad. El BA es menos sensible y la redundancia sólo afecta la interpretabilidad.

B-Indicadores de Volumen y Derivados Estadísticos. Los indicadores de volumen son esenciales para evaluar la fuerza y convicción detrás de los movimientos de precios, y en la Tabla 2 se puede observar su baja a casi nula correlación con los indicadores de precio, aportando un comportamiento distinto en las variables predictoras utilizadas por los tres algoritmos de clasificación. De este tipo de indicador se usaron 3 sub-tipos:

- 4 indicadores de medias móviles simples del volumen,
- 1 indicador de volumen en balance y
- 4 indicadores de desviación estándar normalizada del OBV,

que se detallan en los siguientes subtítulos:

B.1-Media Móvil Simple del Volumen ($volume\_mean\_p$ ***).*** Es la media aritmética del volumen negociado durante un período de p días. Mide el volumen promedio en el período y se utiliza para suavizar la fluctuación del volumen diario.

V_i Volumen en el período i.

p Período de cálculo (Se calculó para los períodos de 5, 10, 20 y 60 días).

$$volume\_mean\_p = \frac{1}{p} \sum_{i=t-p+1}^{t} V_i \qquad (5)$$

Se selecciona para períodos cortos y medios. Indica si el volumen actual es significativamente mayor que el promedio reciente. El volumen alto es necesario para sustentar grandes movimientos de precio, crucial para una predicción a 80 días bursátiles.

B.2-Volumen en Balance -On Balance Volume (OBV)- (obv) ***.*** El OBV es un indicador de impulso que relaciona el volumen con los cambios de precio. Se calcula sumando o restando el volumen total del día al acumulado, dependiendo de si el precio de cierre subió o bajó.

OBV_t Valor del OBV en el período t.

OBV_{t-1} Valor del OBV en el período el período anterior.

V_t Volumen de la negociación en el período t.

C_t Precio de cierre en el período t.

C_{t-1} Precio de cierre en el período anterior.

$$OBV_t = \begin{cases} OBV_{t-1} + V_t & \text{si } C_t > C_{t-1} \\ OBV_{t-1} - V_t & \text{si } C_t < C_{t-1} \\ OBV_{t-1} & \text{si } C_t = C_{t-1} \end{cases} \quad (6)$$

Se incluye directamente el valor del OBV o su tasa de cambio. El OBV es un indicador de impulso que ayuda a confirmar la tendencia. Los incrementos en el OBV junto con alzas de precio sugieren una fuerte acumulación.

B.3-Desviación Estándar Normalizada del OBV (obv_std_p). Es la desviación estándar móvil del OBV multiplicada por la raíz cuadrada del período. Esta normalización se realiza a menudo en ML para ajustar la desviación estándar de una serie de tiempo a la escala de la raíz del tiempo. Mide la volatilidad (dispersión) del indicador OBV en el período p.

OBV_i Valor del OBV en el período i.

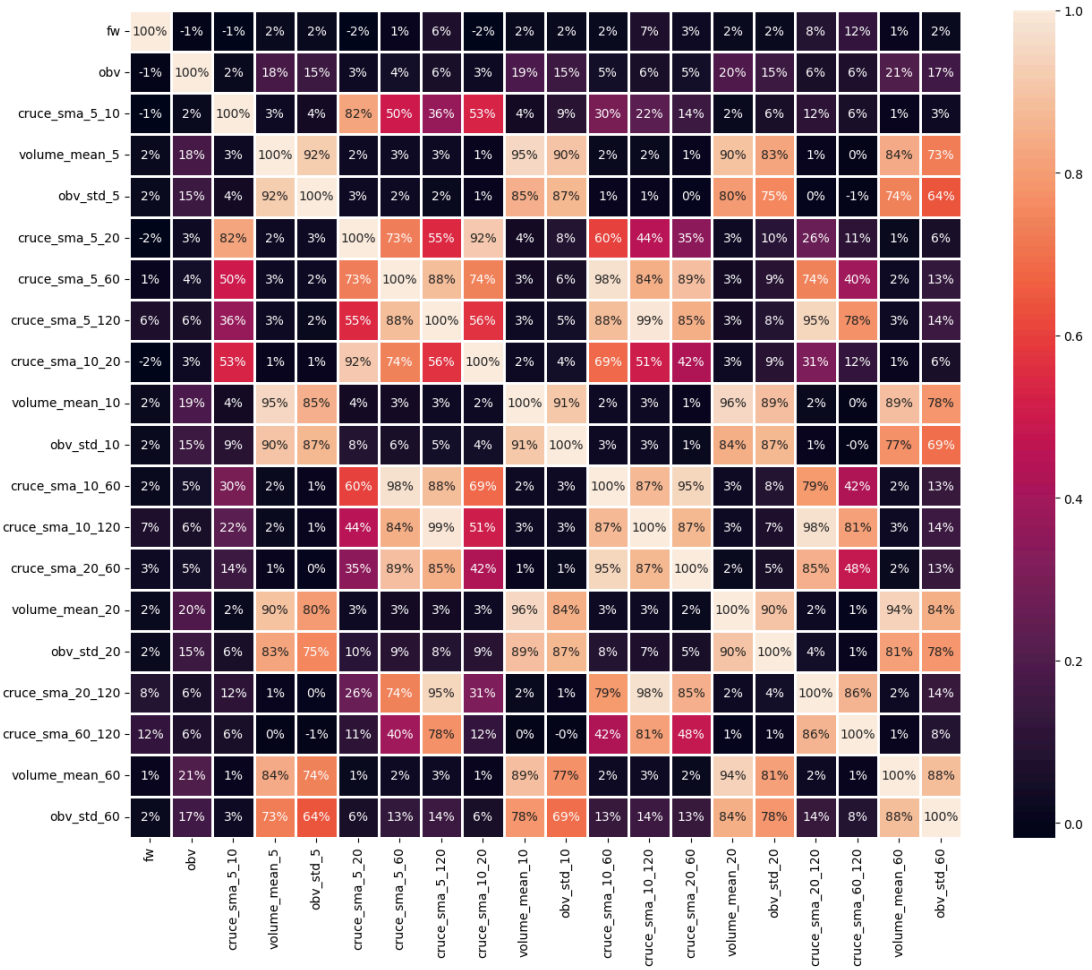
$\overline{OBV}_p$ Media del OBV en el período p.

p Período de cálculo (Se calculó para los períodos de 5, 10, 20 y 60 días).

$$obv\_std\_p = \sqrt{\frac{1}{p-1} \sum_{i=t-p+1}^{t} (OBV_i - \overline{OBV_p})^2} \cdot \sqrt{p} \quad (7)$$

Se selecciona la desviación estándar para períodos cortos a medios. Mide la volatilidad o dispersión del impulso del volumen. Un aumento en esta volatilidad puede ser un precursor de un cambio de tendencia.

Tabla 2. *Matriz de coeficientes de correlación de variables (fw y 19 indicadores).*



Resumen de la Estrategia de Selección. La estrategia debe ser la de construir un conjunto de features que capturen las tres dimensiones clave:

- Impulso/Fuerza de Tendencia: obv, $cruce\_sma\_p_1\_p_2$
- Volatilidad/Riesgo: obv_std_p
- Liquidez/Participación: $volume\_mean\_p$

Debido a que el algoritmo de clasificación de la RL es un modelo lineal sensible a la escala de las variables, antes de entrenar los tres modelos todos los indicadores continuos pertenecientes a la muestra de entrenamiento fueron normalizados usando Z-score. Esta escala fue utilizada para normalizar los valores de los indicadores de la muestra de

validación. El AD y el BA son invariantes a la escala, pero de igual forma se utilizaron estas muestras escaladas para su entrenamiento y validación.

División, Normalización y Tratamiento del Data Leakage en Muestras de Entrenamiento y Validación

División. La división de los datos se realiza de forma secuencial por cada archivo (ticker) y luego se concatenan todos los conjuntos. Los parámetros de la devisión son:

- El 20% de los datos de cada archivo se reserva inicialmente para el conjunto de validación.
- Se introduce un buffer de 120 filas (períodos) de datos entre el final del conjunto de entrenamiento y el inicio del conjunto de validación para evitar data leakage sobre la muestra de validación.

Normalización (Escalado). La normalización se realiza mediante el estandarizador de scikit-learn (StandardScaler).

1. Ajuste (fit) en el conjunto de entrenamiento: Se calcula la media y la desviación estándar para cada indicador exclusivamente en el conjunto de datos de entrenamiento, y luego se transforma usando esta información.
2. Transformación en el conjunto de validación: Se estandariza el conjunto de validación utilizando las medias y desviaciones estándar calculadas previamente.

Tratamiento del Data Leakage. El código implementa dos mecanismos principales para mitigar la filtración de datos (data leakage):

1. Prevención en el escalado: El tratamiento más importante se realiza en la normalización: los datos solamente son ajustados en el conjunto de entrenamiento y luego aplicado al conjunto de validación. Si se ajusta el conjunto validación usando los propios datos, se estaría incorporando información del conjunto de validación al proceso de preprocesamiento, lo que podría llevar a una evaluación artificialmente optimista. Al usar las estadísticas del conjunto de entrenamiento, se simula mejor un escenario real donde los datos de validación son verdaderamente nuevos.
2. Prevención por dependencia temporal (Buffer): El buffer está diseñado para prevenir el leakage debido a la autocorrelación o dependencia temporal de los indicadores (común en series de tiempo financieras). Al dejar 120 filas vacías entre el último dato de entrenamiento y el primer dato de validación, se evita que el modelo use los indicadores con mayor rezago en su cálculo, cuyos cálculos pudieron ser influenciados directamente por el mismo período de tiempo que la variable objetivo del conjunto de validación. Esto asegura una separación temporal limpia entre el entrenamiento y la validación.

La validación de modelos en finanzas requiere un tratamiento especial debido a la naturaleza secuencial y temporal de los datos. La validación cruzada estándar (k-fold cross-validation), que divide el conjunto de datos de forma aleatoria, puede fallar en datos financieros debido a la fuga de información (leakage) entre los conjuntos de entrenamiento y prueba. Si las observaciones de entrenamiento y prueba están demasiado cerca en el tiempo, una estrategia podría aprender de la información futura de una manera que no es replicable en el mundo real (López de Prado, 2018), para contrarrestar esto una técnica avanzada es Purged K-fold Cross-Validation. Esta técnica soluciona el problema de leakage introduciendo un embargo entre los conjuntos de entrenamiento y prueba. Esto significa que se excluye un

período de tiempo de los datos entre el final del conjunto de entrenamiento y el inicio del conjunto de prueba, asegurando la independencia temporal (López de Prado, 2018).

Manejo del Balance de las Muestras

Las muestras no fueron afectadas por métodos de oversampling u undersampling, debido a que la muestra de entrenamiento tiene un sesgo alcista de 61.50% y la muestra de validación un sesgo alcista de 77.07%.

Dada la finalidad de comparar modelos de clasificación para predecir la tendencia alcista a 80 días bursátiles en precios de acciones de BYMA, las muestras no fueron afectadas por métodos de oversampling o undersampling, porque se priorizó mantener la integridad del escenario real y la comparabilidad objetiva de los algoritmos.

El desequilibrio de clases (61.50% sesgo alcista en entrenamiento y 77.07% sesgo alcista en validación) es una característica inherente de los mercados financieros, que históricamente tienden a ser alcistas (sesgo de tendencia).

Al no modificar la distribución, se entrenó y validó los tres algoritmos con el contexto real donde operarán. Si fuese balanceado los datos artificialmente a 50%/50% o en proporción, los algoritmos podrían obtener métricas infladas o aprender patrones que son irrelevantes para el entorno real de las empresas de servicios financieros que cotizan en BYMA.

Un desequilibrio de 61.50% vs 38.50% no es tan extremo como para inutilizar los algoritmos de clasificación utilizados. La clase minoritaria (tendencia bajista) representa una porción significativa del conjunto de entrenamiento (38.50%). Los modelos pueden aprender de esta cantidad de datos sin necesidad de manipulación. Las técnicas de balanceo se consideran necesarias para este tipo de datos y objetivos cuando la clase minoritaria cae por debajo del 10% o 5%.

Modelado y Optimización

Se utilizaron tres algoritmos de clasificación de aprendizaje supervisado (ML), detallados en los siguientes subtítulos, en conjunto con los métodos de optimización que se le aplicaron:

Árbol de Decisión. Es un modelo no lineal que divide el espacio de características en regiones basadas en una serie de decisiones binarias (*Cómo implementar un modelo de clasificación con Python*, 2025). Su fortaleza es la interpretabilidad, pero es propenso al sobreajuste (Aco Tito, Hancco Condori & Pérez Vera, 2023). La configuración **inicial** con la que fue entrenado el modelo del AD se puede ver en la Tabla A1 del Apéndice.

La configuración de hiperparámetros del algoritmo de la clase DecisionTreeClassifier y la mejora en los resultados se deben a la aplicación de técnicas de Ajuste de Hiperparámetros (Hyperparameter Tuning -(HT)-) mediante las clases RandomizedSearchCV y GridSearchCV, lo que determinó la configuración **final** del modelo del AD que se puede ver en la Tabla A2 del Apéndice.

Los AD son los bloques de construcción de los modelos de conjunto más populares. Son algoritmos versátiles que pueden realizar tanto tareas de clasificación como de regresión. Sin embargo, si se dejan sin restricciones, son propensos al sobreajuste (overfitting), lo que aumenta la varianza de las predicciones y los hace menos fiables en datos nuevos. Para mitigar esto, se utilizan hiperparámetros de regularización como max_depth (la profundidad máxima del árbol), min_samples_split (el número mínimo de muestras para que un nodo pueda ser dividido) y max_leaf_nodes (el número máximo de nodos hoja). Reducir la

complejidad del árbol a través de estos hiperparámetros ayuda a regularizar el modelo y a mejorar su capacidad de generalización (López de Prado, 2018).

La expresión 2^n, donde n es la profundidad del AD, proporciona el número máximo teórico de nodos terminales (o nodos hoja) que el árbol puede albergar. Un árbol de decisión es, por naturaleza, una estructura binaria, lo que significa que en cada nivel, un nodo se divide en un máximo de dos nodos nuevos. De esta manera, en el nivel $n + 1$ (profundidad n), pueden existir hasta 2^n nodos terminales. En resumen, 2^n representa la cantidad de cajas de decisión finales que se podrían obtener si el árbol se dividiera de forma perfectamente balanceada en cada nivel hasta alcanzar la profundidad establecida. Esta cifra es un límite superior teórico. En la práctica, un árbol real casi siempre contendrá menos nodos hoja que este máximo, dado que el proceso de división se detiene debido a varios criterios:

- Se alcanza el número mínimo de muestras requerido en un nodo.
- Se cumple el criterio de pureza para el nodo.
- Se llega a la profundidad máxima (`max_depth`) establecida.

La **profundidad máxima** del AD y de los árboles del BA utilizados se seleccionó de forma experimental mediante una heurística, que se puede representar con la siguiente fórmula:

$$Profundidad = \log_2\left(\frac{T}{30}\right) \qquad (8)$$

Donde:

- Profundidad: Es la profundidad máxima.
- T: Es el tamaño de la muestra de entrenamiento.

Esta fórmula heurística tiene como **objetivo** garantizar que cada nodo hoja tenga, en promedio, un **mínimo de 30 observaciones**, utilizando el logaritmo en base 2 debido a la naturaleza binaria de los AD.

Al dividir el tamaño total de la muestra (T) por el número máximo teórico de nodos terminales (2^n), se obtiene la cantidad mínima de datos que existiría por nodo terminal ($\frac{T}{2^n}$). Esta proporción se relaciona con el número mínimo de datos necesario por nodo terminal para evitar el sobreajuste (overfitting). Si la cantidad de datos por nodo terminal es menor a 30, se recomienda disminuir la profundidad del árbol o aumentar la cantidad de datos de entrenamiento, en este caso las funciones desarrolladas optan por disminuir la profundidad del árbol/es. Para prevenir el sobreajuste y debido a que fue el mejor resultado obtenido para el AD y el BA luego de realizar el HT, se dejó fija la siguiente **condición** en la función desarrollada, que dice que n debe ser el **máximo valor** que cumple con la **relación** $\frac{T}{2^n} \geq 30$, la cual se puede expresar de la siguiente forma:

$$max\_depth = n = \max\{k \in \mathbb{N}_0 \mid \frac{T}{2^k} \geq 30\} \qquad (9)$$

Donde:

- n: Es la profundidad máxima del árbol (el valor que se busca).
- T: Es el tamaño total de la muestra de entrenamiento.
- k: Representa una posible profundidad del árbol.
- N_0: Denota el conjunto de números naturales ({0,1,2,3,…}), ya que la profundidad de un árbol es un entero no negativo.

- $max\{...\}$: Significa que n es el valor máximo de k que satisface la condición.
- $\frac{T}{2^k} \geq 30$: Es la relación de restricción, que establece que la división de la muestra (T) entre el número máximo teórico de nodos terminales (2^k) debe ser mayor o igual a 30.

La relación ideal, no obstante, es $\frac{T}{2^n} \geq 400$, pero no se lo consideró debido a la cantidad de datos que se tiene para entrenar los modelos. En conclusión, la selección de la profundidad óptima es un equilibrio esencial entre la capacidad de aprendizaje del modelo y el riesgo de sobreajuste.

La adopción de la cifra 30 es porque:

- Evita el ruido: Asegura que el nodo hoja no esté basado en una sola o unas pocas observaciones, que podrían ser ruido o valores atípicos. Un tamaño de $n \geq 30$ es a menudo considerado el mínimo requerido para que una muestra empiece a exhibir una distribución medianamente estable de acuerdo al Teorema del Límite Central (TLC).
- Fórmula Heurística: La fórmula (8) se basa en la idea de que, idealmente, se quiere un árbol que sea lo más profundo posible (para capturar patrones) sin que el número promedio de muestras por hoja caiga por debajo de 30, ya que esto dispararía el riesgo de sobreajuste.

Aunque los AD no requieren explícitamente la normalidad (son algoritmos no paramétricos), el uso de $n \geq 30$ garantiza que la subpoblación en el nodo terminal sea de un tamaño razonable para cualquier inferencia local que se haga. Si bien el TLC es más relevante para la inferencia estadística (pruebas de hipótesis, intervalos de confianza), su

adopción en ML simplemente refuerza la idea de que $n = 30$ es el umbral para una muestra con propiedades estadísticas mínimamente estables.

El valor de 400 representa una práctica aún más conservadora que se utiliza en entornos donde la precisión estadística de cada estimación es crucial, como en ciertas aplicaciones financieras o médicas donde un alto número de muestras asegura la solidez de la predicción de la hoja. Y además indirectamente está alineada con los principios fundamentales de la Ley de los Grandes Números y el TLC: cuantas más muestras se tienen, más estable y fiable es la estadística calculada (media o moda), lo que se traduce en un modelo con mejor capacidad de generalización y menor sobreajuste.

Mediante las clases RandomizedSearchCV y GridSearchCV se exploraron diferentes combinaciones de hiperparámetros (incluido el criterion) y seleccionaron la que produjo el mejor rendimiento según la métrica de evaluación definida en validación cruzada de 5 Pliegues (5-Fold Cross-Validation). La métrica de evaluación prefirió las configuraciones de hiperparámetros que luego de predecir y validar obtenían un Sesgo (Alcista) más próximo al valor del Sesgo (Alcista) de la muestra utilizada para validar (77.07%).

Primero se utilizó RandomizedSearchCV para seleccionar aleatoriamente combinaciones de hiperparámetros del espacio de búsqueda y evaluarlas mediante validación cruzada (Cross-Validation). Una vez que se identificó un área prometedora, se aplicó GridSearchCV realizando una búsqueda exhaustiva de todas las combinaciones de hiperparámetros especificadas en una rejilla (grid) más pequeña y enfocada.

Al evaluar todas las combinaciones, el proceso de GridSearchCV (etapa final de la optimización) determinó que la configuración óptima para el modelo era criterion='gini', (manteniendo max_depth=9, random_state=0 y el resto de los hiperparámetros en su configuración por defecto).

El cambio del hiper parámetro clave es el criterio de impureza (criterion):

- Sin Tuning = entropy - Ganancia de Información (Information Gain): Mide el desorden o la incertidumbre, buscando maximizar la reducción de entropía.
- Con Tuning= gini - Índice de Gini: Mide la probabilidad de clasificar incorrectamente un elemento seleccionado al azar.

El hecho de que el modelo con criterion=gini (Con HT) haya mostrado una mejora en el rendimiento sugiere que, para los datos específicos utilizados, el Índice de Gini fue una métrica más efectiva para guiar las divisiones del árbol, resultando en un modelo con mejor poder predictivo.

La optimización de hiperparámetros mejoró el rendimiento del modelo modificando varias métricas, pero solamente se detalla el sesgo alcista de la validación que es la métrica de evaluación que se utilizó, el cual pasó de ser 57.5% Sin Tuning, a ser 66.44% Con Tuning.

En resumen, la aplicación de RandomizedSearchCV y GridSearchCV permitió descubrir que cambiar el criterio de impureza de entropy a gini resultaba en un modelo que se ajustaba mejor a los datos y demostraba una capacidad predictiva superior, especialmente en la identificación de casos verdaderos positivos (aciertos en la predicción de tendencia alcista).

Regresión Logística. Es un modelo de clasificación lineal que utiliza la función logística para predecir la probabilidad de que una instancia pertenezca a una clase -alcista o

bajista- (*Cómo implementar un modelo de clasificación con Python*, 2025). Sirve como un benchmark interpretable para comparar modelos más complejos (Aco Tito et al., 2023). El modelo de RL fue entrenado **inicialmente** con la configuración que se puede ver en la Tabla A1 del Apéndice.

La configuración **final** de hiperparámetros de la clase LogisticRegression, que se puede ver en la Tabla A2 del Apéndice, se obtuvo tras la aplicación de técnicas de Ajuste de Hiperparámetros (HT) mediante las clases RandomizedSearchCV y GridSearchCV, logrando una mejora en el balance de los resultados.

La mejora en los resultados del algoritmo de RL se debe a la optimización de sus hiperparámetros a través de la aplicación combinada de RandomizedSearchCV y GridSearchCV. Estas clases exploraron diferentes combinaciones y seleccionaron la que produjo el mejor rendimiento, según la métrica de evaluación definida en validación cruzada de 5 Pliegues (5-Fold Cross-Validation).

La métrica de evaluación prioriza las configuraciones con menor valor en la sumatoria de las métricas de Sensibilidad y Precisión, lo que permitió maximizar el error tipo II y minimizar el error tipo I, según las características del comportamiento de las muestras utilizadas.

Los principales cambios en los hiperparámetros fueron:

- class_weight pasó de ser None (pesos uniformes) a 'balanced'. Al establecer class_weight='balanced'`, el algoritmo automáticamente ajusta los pesos de las clases inversamente proporcionales a sus frecuencias en los datos de entrenamiento. Esto es

crucial para conjuntos de datos desequilibrados, ya que penaliza los errores en la predicción de la clase minoritaria (bajista), forzando al modelo a prestar más atención a esos casos y mitigando el sesgo hacia la clase mayoritaria.

- solver pasó de `'lbfgs'` a 'newton-cholesky'. El solver define el algoritmo de optimización utilizado. 'newton-cholesky' (una variante del método de Newton) puede ser más robusto o converger más rápido para ciertos problemas, aunque su elección está estrechamente ligada a su compatibilidad con otros parámetros (como penalty='l2').
- max_iter Aumentó de 100 a 124. Se incrementó el número máximo de iteraciones permitidas para que el solver converja. Este leve aumento indica que el proceso de ajuste pudo haber requerido más pasos para alcanzar la convergencia con los nuevos parámetros (especialmente class_weight='balanced'`).

Al evaluar todas las combinaciones, el proceso de GridSearchCV (etapa final de la optimización) determinó que la configuración óptima para el modelo era: class_weight='balanced', solver='newton-cholesky', y max_iter=124 (manteniendo C=1.0, random_state=0 y el resto de los hiperparámetros por defecto).

La optimización de hiperparámetros generó un cambio significativo en la estrategia predictiva del modelo, pero solamente se detalla el sesgo alcista de la validación, métrica que se está usando como referencia para evaluar los modelos. El sesgo alcista de la validación pasó de ser 98.60% Sin Tuning, a ser 70.51% Con Tuning.

La aplicación de los ajustes de hiperparámetros (principalmente class_weight='balanced'`) transformó el modelo:

- Modelo Sin Tuning: Exhibía un sesgo extremo hacia la clase mayoritaria (alcista, '1'). Su Sensibilidad para la clase '1' era casi perfecto (98.82%), pero era virtualmente inútil para predecir la clase minoritaria (bajista, '0'), con una Sensibilidad de sólo 2.14%. El modelo era excesivamente simple, predecía casi siempre alcista.
- Modelo Con Tuning: Logró un balance mucho mayor en su capacidad predictiva. Aunque la Exactitud general disminuyó de 76.2% a 61.91%, la Sensibilidad de la clase minoritaria (bajista) mejoró drásticamente de 2.14% a 31.24%. Este cambio indica que el modelo ahora es una herramienta más útil y robusta para la clasificación binaria, ya que aumentaron los aciertos en la clase dominante, que es de interés para el objetivo la investigación, y tienen menor cuantía de Falsos Positivos, a la vez que tiene una capacidad significativa (mejorada) para identificar la clase minoritaria, que es la más difícil de predecir, pero que no es de interés para el objetivo la investigación.

Bosque Aleatorio. Es un método de ensemble que combina múltiples árboles de decisión para mejorar la precisión y la robustez del modelo (*Cómo implementar un modelo de clasificación con Python*, 2025). Esta técnica reduce la varianza y mitiga el sobreajuste, haciéndolo una herramienta poderosa para la predicción de precios (Gómez Plaza, S. D., 2025).

Los modelos de conjunto (Ensemble Methods), como los que combinan las predicciones de múltiples modelos, han demostrado ser altamente efectivos en la predicción de precios de acciones (Ampomah et al., 2020). La lógica detrás de estos métodos es que, incluso si los clasificadores individuales son **aprendices débiles** que sólo rinden un poco mejor que el azar, la combinación de sus predicciones puede formar un **aprendiz fuerte** con una precisión mucho mayor, siempre y cuando sus errores no estén correlacionados. El éxito

de estos modelos se basa en la diversidad de sus componentes (López de Prado, 2018).La configuración **inicial** con la que fue entrenado el modelo del BA se puede ver en la Tabla A1 del Apéndice.

La configuración de hiperparámetros del algoritmo de la clase RandomForestClassifier y la mejora en los resultados se deben a la aplicación de técnicas de Ajuste de Hiperparámetros (HT) mediante las clases RandomizedSearchCV y GridSearchCV, lo que determinó la configuración **final** del modelo del BA que se puede ver en la Tabla A2 del Apéndice. A través de las instancias de ambas clases, se exploraron diversas combinaciones de hiperparámetros -como el número de estimadores (n_estimators), la cantidad máxima de características (max_features), el tamaño máximo de la muestra (max_samples) y el criterio de impureza (criterion)- para seleccionar la configuración que produjo el mejor rendimiento, de acuerdo con la métrica de evaluación definida en la validación cruzada de 5 Pliegues (5-Fold Cross-Validation). La métrica de evaluación prefirió las configuraciones de hiperparámetros que luego de predecir y validar obtenían un Sesgo (Alcista) más próximo al valor del Sesgo (Alcista) de la muestra utilizada para validar (77.07%).

Primero se utilizó RandomizedSearchCV para seleccionar aleatoriamente combinaciones de hiperparámetros del espacio de búsqueda y evaluarlas mediante validación cruzada (Cross-Validation). Una vez que se identificó un área prometedora, se aplicó GridSearchCV realizando una búsqueda exhaustiva de todas las combinaciones de hiperparámetros especificadas en una rejilla (grid) más pequeña y enfocada.

Al evaluar todas las combinaciones, el proceso de GridSearchCV (etapa final de la optimización) determinó que la configuración óptima para el modelo era criterion='gini', n_estimators=150, max_features=4, max_samples=0.85 (manteniendo max_depth=9, random_state=0 y el resto de los hiperparámetros en su configuración por defecto).

El cambio de los hiperparámetros clave son el número de estimadores, el criterio de impureza, la cantidad de características a utilizar en cada estimador y el tamaño de la muestra de la muestra de entrenamiento a utilizar en cada estimador:

- n_estimators (Árboles) = 500: Mayor cantidad de árboles puede reducir la varianza, pero aumenta el tiempo de entrenamiento. 150: Un valor más bajo fue suficiente para un buen rendimiento, reduciendo la complejidad y el tiempo de cómputo.
- criterion (Criterio de impureza) = entropy - Ganancia de Información (Information Gain): Busca maximizar la reducción de entropía. gini - Índice de Gini: Mide la probabilidad de clasificar incorrectamente un elemento seleccionado al azar.
- max_features (Features por división): La reducción de 5 a 4 features consideradas en cada división incrementa la aleatoriedad del modelo. Esto típicamente ayuda a reducir la correlación entre los árboles individuales, mejorando la diversidad del ensemble y mitigando el sobreajuste (overfitting).

El hecho de que el modelo con criterion='gini', n_estimators=150, y max_features=4 (Con HT) haya mostrado una mejora en el rendimiento sugiere que, para los datos específicos utilizados, el Índice de Gini fue una métrica más efectiva para guiar las divisiones de los árboles, y la reducción en la cantidad de árboles (n_estimators), la limitación de features por división (max_features=4), y la adición de max_samples=0.85 ayudaron a optimizar el balance entre sesgo y varianza, resultando en un modelo con mejor poder predictivo.

La optimización de hiperparámetros mejoró el rendimiento del modelo modificando varias métricas, pero solamente se detalla el sesgo alcista de la validación que es la métrica de evaluación que se utilizó, el cual pasó de ser 71.17% Sin Tuning, a ser 73.78% Con Tuning.

En resumen, la aplicación de RandomizedSearchCV y GridSearchCV permitió descubrir que cambiar el criterio de impureza de `entropy` a `gini`, reducir la cantidad de features consideradas en cada división de 5 a 4 para aumentar la robustez, junto con la optimización de otros parámetros como n_estimators y max_features, resultó en un modelo que se ajustaba mejor a los datos y demostraba una capacidad predictiva superior, especialmente en la identificación de casos verdaderos positivos (aciertos en la predicción de tendencia alcista).

Validación y Evaluación del Rendimiento de los Modelos

Para evaluar el rendimiento de los modelos de clasificación de Aprendizaje Automático se utilizaron métricas de desempeño derivadas de la matriz de confusión. Estas métricas son esenciales para cuantificar la efectividad de los algoritmos en la predicción de la tendencia alcista a 80 días bursátiles de los precios de las acciones. La matriz de confusión organiza los resultados de la clasificación en cuatro categorías:

1. **Positivos detectados**: El modelo predijo un resultado positivo y el resultado real fue positivo.
2. **Falsos negativos** (**Error tipo II**): El modelo predijo un resultado negativo pero el resultado real fue positivo.
3. **Negativos detectados**: El modelo predijo un resultado negativo y el resultado real fue negativo.

4. **Falsos positivos** (**Error tipo I**): El modelo predijo un resultado positivo pero el resultado real fue negativo.

A partir de estas cuatro categorías, se calcularon las siguientes métricas para cada modelo:

- **Sensibilidad** (**Recall**): Esta métrica mide la capacidad del modelo para identificar correctamente todos los casos positivos reales. Se calcula como la proporción de los positivos detectados sobre el total de positivos reales. Es la probabilidad de que un evento positivo sea correctamente pronosticado como positivo. La fórmula es:

$$Sensibilidad = \frac{Positivos\ detectados}{Positivos\ detectados + Falsos\ Negativos} \qquad (10)$$

- **Exactitud** (**Accuracy**): Representa la proporción de predicciones correctas totales (tanto positivas como negativas) sobre el total de mediciones. Se calcula como el cociente entre las Mediciones Correctas (Positivos detectados + Negativos detectados) y el Total de Mediciones (Positivos detectados + Falsos Positivos + Negativos detectados + Falsos Negativos). La fórmula es:

$$Exactitud = \frac{Mediciones\ Correctas}{Total\ de\ Mediciones} \qquad (11)$$

- **Precisión** (**Precision**): Esta métrica indica la proporción de predicciones positivas que fueron realmente correctas. Se calcula como la proporción de los positivos detectados sobre el total de predicciones positivas realizadas. La fórmula es:

$$Precisión = \frac{Positivos\ detectados}{Positivos\ detectados + Falsos\ Positivos} \qquad (12)$$

- **Puntuación F1** (**F1-score**): Es la media armónica de la precisión y la sensibilidad. Esta métrica es especialmente útil cuando se busca un equilibrio entre precisión y

sensibilidad y es una medida más robusta que la exactitud en casos de clases desbalanceadas. La fórmula es:

$$F1\ score = 2 \times \frac{Precisión \times Sensibilidad}{Precisión + Sensibilidad} \quad (13)$$

- **Sesgo** (**Alcista**): Esta métrica indica la proporción de predicciones positivas detectadas como correctas e incorrectas sobre el total de mediciones. Se calcula como el cociente entre la sumatoria de Positivos detectados y Falsos Positivos y el Total de Mediciones (Positivos detectados + Falsos Positivos + Negativos detectados + Falsos Negativos). La fórmula es:

$$Sesgo\ (Alcista) = \frac{Positivos\ detectados + Falsos\ Positivos}{Total\ de\ Mediciones} \quad (14)$$

Además del análisis de estas métricas, los resultados se validaron usando un análisis walk-forward. Este método simuló un entorno real al entrenar el modelo en una ventana de datos (80 días bursátiles) que se desplaza y validar su rendimiento en datos futuros no vistos. Esta técnica proporciona una evaluación más realista de la capacidad del modelo para generalizar a nuevas condiciones de mercado que un backtest estático. La validación walk-forward es crucial para mitigar el sobreajuste en backtesting, un problema común en la investigación financiera donde una estrategia se ajusta demasiado a un conjunto de datos histórico limitado, lo que lleva a un rendimiento deficiente en datos nuevos e imprevistos (Brandimarte, 2014).

Para visualizar el desempeño de los modelos, se utilizaron tablas construidas a partir de la librería pandas, matrices de confusión y otros gráficos pertinentes a cada situación de análisis construidos con la librería Matplotlib.

Criterio para Comparar (Ranquear) los Algoritmos de Clasificación según su Objetivo Predictivo. El **criterio** que permitió determinar la **efectividad** y el **mejor rendimiento** entre los tres algoritmos de clasificación de ML (AD, RL y BA), para predecir la tendencia alcista de los precios de las acciones argentinas de servicios financieros que cotizan en BYMA, se basó en los siguientes **dos pasos** y **tres condiciones excluyentes**, que al finalizar su ejecución obtienen por resultado el **Ranking de los Algoritmos de Clasificación según su Objetivo Predictivo**:

1. **Ordenar los modelos** de **mayor a menor efectividad y rendimiento**, siendo los **primeros** modelos aquellos que tienen un **Sesgo (Alcista)** en las **predicciones** con una **mayor proximidad absoluta** al **Sesgo (Alcista)** de la **muestra** utilizada para validar.
2. **Comparar por par de modelos** en base a las siguientes **tres condiciones excluyentes**:
    a. Si al comparar el par de modelos, las **proximidades absolutas** entre **iguales métricas** de las **predicciones**, para las seis métricas:
        - Sesgo (Alcista)
        - Precisión
        - Sensibilidad
        - F1- Score
        - Positivos detectados
        - Exactitud

    son **menores** al **uno por ciento (1%)**: Los dos modelos comparados se considera que tienen una *efectividad y rendimiento* ***similar***.

b. Si al comparar el par de modelos, las **predicciones** de ambos modelos tienen un **Sesgo (Alcista)** con una **proximidad absoluta entre sí menor** al **cinco por ciento (5%)**: Entre los dos modelos se ranquea como modelo con ***mayor*** *efectividad y rendimiento* al que tiene la **mayor cantidad de métricas con mayores valores o mejores** según el significado de cada una:
    - Precisión
    - Sensibilidad
    - F1- Score
    - Positivos detectados
    - Exactitud
c. Si al comparar el par de modelos, las **predicciones** de ambos modelos tienen un **Sesgo (Alcista)** con una **proximidad absoluta entre sí mayor o igual** al **cinco por ciento (5%)**: Entre los dos modelos se ranquea como modelo con ***mayor*** *efectividad y rendimiento* al que tiene el **mayor valor de Sesgo (Alcista)**.

Los Errores tipo I y tipo II se observaron de forma descriptiva para comprender la relación que existe con el sesgo de las predicciones y el resto de las métricas, lo que ayuda en gran medida a responder el interrogante: *¿Prefieres perder capital o perder una oportunidad?*. En la mayoría de las estrategias de inversión o trading, la prioridad suele ser la **preservación del capital**. Por lo tanto, casi siempre se prefiere minimizar el riesgo de pérdida de capital sobre la pérdida de oportunidad.

Por lo tanto, la respuesta común es: preferir tener una mayor cantidad de Errores Tipo II (Falsos Negativos) que Errores Tipo I (Falsos Positivos). En un escenario donde se invierta en tendencias alcistas y no en bajistas, esto se traduce en lo siguiente:

- Minimizar Error Tipo I (Falso Positivo): Se busca que el modelo sea muy preciso (Precisión) en sus predicciones de la clase Alcista. Se prefiere que el modelo sólo diga Alcista cuando está muy seguro, para evitar hacer una mala inversión (**preservación del capital**).
- Aceptar más Error Tipo II (Falso Negativo): Se está dispuesto a perder algunas oportunidades de ganancias (cuando el modelo dice Bajista pero la acción realmente sube) a cambio de protegerse de la pérdida de capital causada por un Falso Positivo (**mayor costo de oportunidad**).

En resumen se prioriza la Precisión (Precision) sobre la Sensibilidad (Recall) de la clase Alcista.

Resultados

En este apartado, se presentan y describen los hallazgos que se obtuvieron a partir del análisis de los datos de las siete acciones de empresas argentinas del sector de servicios financieros que cotizan en BYMA. Los resultados se organizaron en correspondencia con los objetivos específicos planteados en la investigación. El análisis se centró en la evaluación de la efectividad y el rendimiento de los tres algoritmos de clasificación de ML:

- AD,
- RL y
- BA.

El análisis de los datos se realizó en un entorno de programación Python, utilizando la librería Scikit-learn para la implementación de los algoritmos y el cálculo de métricas de desempeño. Se emplearon métricas de desempeño derivadas de la matriz de confusión para cuantificar la efectividad de los algoritmos en la predicción de la tendencia alcista a 80 días bursátiles para los precios de las acciones. Además, se utilizó un análisis walk-forward para validar los resultados, simulando un entorno real y evaluando la capacidad de los modelos para generalizar a nuevas condiciones de mercado.

Para cumplir el objetivo general de esta investigación se efectivizaron los siguientes cinco objetivos específicos planificados:

Objetivo Específico N° 1

Se recolectaron los datos históricos de las acciones de siete empresas argentinas de servicios financieros con oferta pública en BYMA. El detalle del dataset persistido, sin procesar los datos históricos, se ve en la Tabla 3:

Tabla 3. *Detalle del dataset de datos históricos sin procesar.*

Ticker	Fecha de inicio	Fecha de Fin	Días de negociación	Tamaño del dataset
BBAR	2 de enero de 2001	12 de septiembre de 2025	6035	6035 filas y 6 columnas
BHIP	12 de mayo de 2004	12 de septiembre de 2025	5072	5072 filas y 6 columnas
BMA	2 de enero de	12 de	5970	5970 filas y 6

Ticker	Fecha de inicio	Fecha de Fin	Días de negociación	Tamaño del dataset
	2001	septiembre de 2025		columnas
BPAT	23 de julio de 2008	12 de septiembre de 2025	4071	4071 filas y 6 columnas
SUPV	6 de junio de 2017	12 de septiembre de 2025	2017	2017 filas y 6 columnas
BYMA	31 de mayo de 2018	12 de septiembre de 2025	1776	1776 filas y 6 columnas
GGAL	2 de enero de 2001	12 de septiembre de 2025	6038	6038 filas y 6 columnas

La sumatoria de las filas del dataset de cada ticker nos da por resultado un dataset con 30979 filas 6 columnas de las siguiente información de negociación diaria:

1. precio de apertura,
2. precio máximo,
3. precio mínimo,
4. precio de cierre,
5. volumen de acciones negociadas,
6. ratio de retorno diario (tanto por uno)

El dataset de cada ticker está indexado por la fecha de negociación (año-mes-día).

Objetivo Específico N° 2

Se construyeron 19 predictores:

- 10 indicadores de cruces de medias móviles simples de la media OHLC (son todas las combinaciones de 2 medias móviles simples de la media OHLC de 5, 10, 20, 60 y 120 días de negociación),
- 4 indicadores de medias móviles simples del volumen (de 5, 10, 20 y 60 días de negociación),
- 1 indicador de volumen en balance y
- 4 indicadores de desviación estándar normalizada del OBV (de 5, 10, 20 y 60 días de negociación).

La función de Python que se desarrolló calculó los 19 indicadores que se utilizaron como predictores de los tres modelos de clasificación y una variable que se utilizó como referencia para la predicción de la tendencia alcista a 80 días bursátiles de los precios de las acciones de empresas argentinas del sector de servicios financieros que cotizan en BYMA. Esta variable se identificó como la ventana de tiempo (fw) que los modelos tenían por objetivo predecir, la cual se relacionó con una máscara que se nombró bajo el título de target (variable categórica binaria -1 = Alcista y 0 = Bajista/Lateral-).

La división de los datos se realizó de forma secuencial por cada archivo (ticker) y luego se concatenaron todos los conjuntos. El 20% de los datos de cada archivo se reservó inicialmente para el conjunto de validación. Y se introdujo un buffer de 120 filas (días de negociación) de datos entre el final del conjunto de entrenamiento y el inicio del conjunto de validación para evitar data leakage sobre la muestra de validación.

Antes de entrenar los tres modelos, los 19 indicadores continuos pertenecientes a la muestra de entrenamiento (80% de los datos de cada archivo) fueron normalizados usando Z-score.

Objetivo Específico N° 3

Se realizaron los entrenamientos de los modelos de clasificación de ML (AD, RL y BA) utilizando los datos procesados, mediante la función de Python que sirve para entrenar y ajustar sus hiperparámetros. El detalle de la configuración **inicial** que tuvieron los hiperparámetros de cada modelo se puede ver en la Tabla A1 del Apéndice.

Luego de realizar el ajuste de hiperparámetros (HT), mediante las clases RandomizedSearchCV y GridSearchCV, se obtuvo la configuración **final** de los hiperparámetros de cada modelo, que se puede ver en la Tabla A2 del Apéndice.

La optimización de hiperparámetros (HT) mejoró el rendimiento de los tres modelos de clasificación (AD, RL y BA), modificando varias métricas y aproximando el Sesgo (Alcista) de las predicciones luego de validarlo al **Sesgo (Alcista) de 77.07%** de la muestra de validación. Estas mejoras de las métricas pueden observarse en la Tabla 4, Tabla 5 y Tabla 6.

Tabla 4. *Métricas del modelo AD Sin HT y Con HT.*

Modelo	Sesgo (Alcista)	Precisión	Sensibilidad	F1-score	Exactitud	Positivos detectados
AD Sin HT	57.50%	73.99%	55.20%	63.23%	50.52%	42.55%
AD Con HT	66.44%	74.85%	64.53%	69.30%	55.95%	49.73%

Tabla 5. *Métricas del modelo RL Sin HT y Con HT.*

Modelo	Sesgo (Alcista)	Precisión	Sensibilidad	F1-score	Exactitud	Positivos detectados
RL Sin HT	98.60%	77.24%	98.82%	86.71%	76.65%	76.16%
RL Con HT	70.51%	77.64%	71.04%	74.19%	61.91%	54.75%

Tabla 6. *Métricas del modelo BA Sin HT y Con HT.*

Modelo	Sesgo (Alcista)	Precisión	Sensibilidad	F1-score	Exactitud	Positivos detectados
BA Sin HT	71.16%	76.43%	70.58%	73.39%	60.55%	54.39%
BA Con HT	73.78%	76.20%	72.95%	74.54%	61.60%	56.22%

Objetivo Específico N° 4

Se realizaron las validaciones de los modelos de clasificación de ML con HT del AD, la RL y el BA utilizando los datos procesados, y se persistieron sus resultados, los cuales se pueden ver en las matrices de confusión en el conjunto validación de la:

- **AD**: Tabla 7, Tabla 8, Tabla 9, Tabla 10
- **RL**: Tabla 11, Tabla 12, Tabla 13, Tabla 14
- **BA**: Tabla 15, Tabla 16, Tabla 17, Tabla 18

Tabla 7. *Matriz de confusión del modelo AD.*

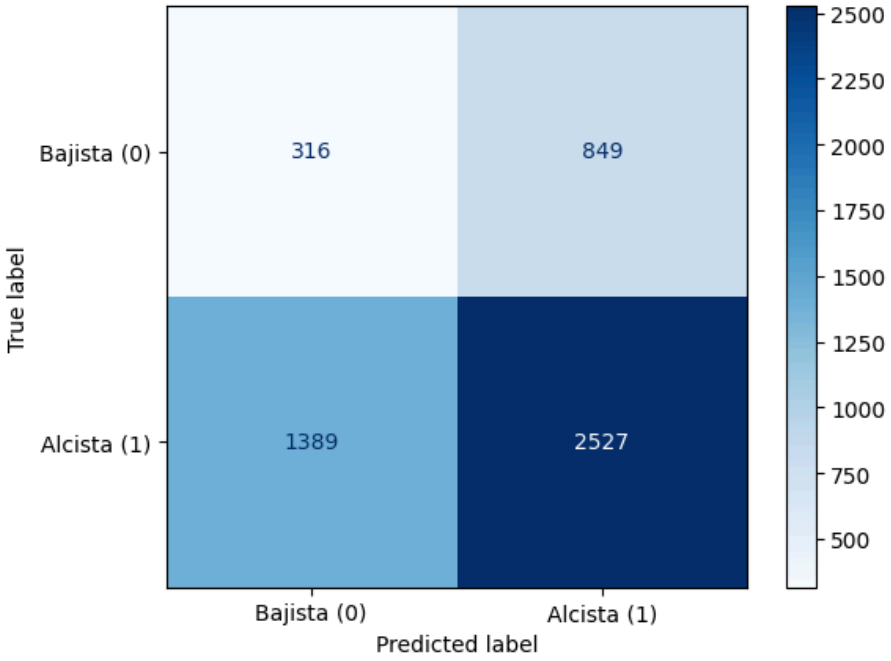


Tabla 8. *Matriz de confusión normalizada al total de la muestra del modelo AD.*

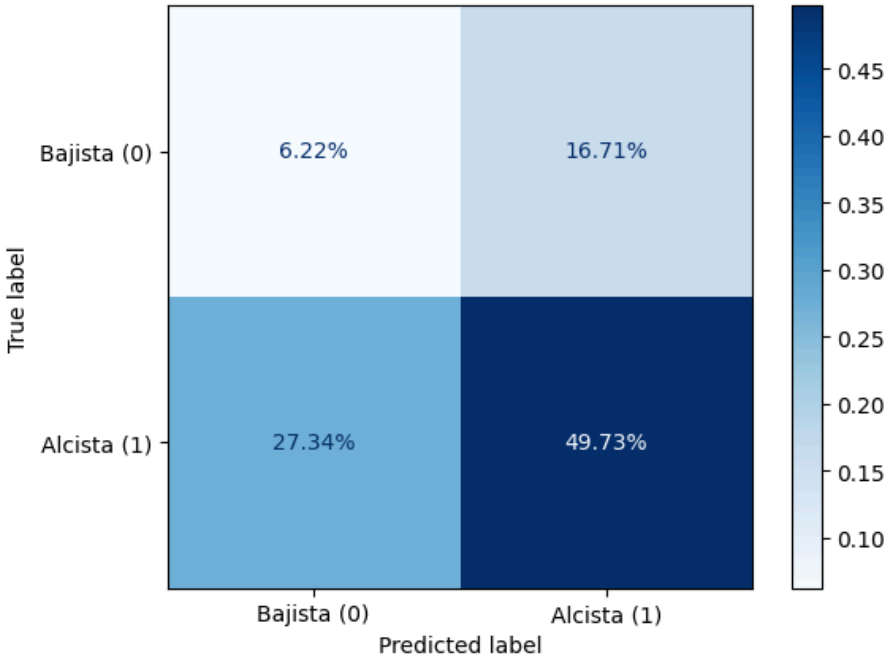


Tabla 9 *Matriz de confusión del modelo AD: precisión de predicción por clase.*

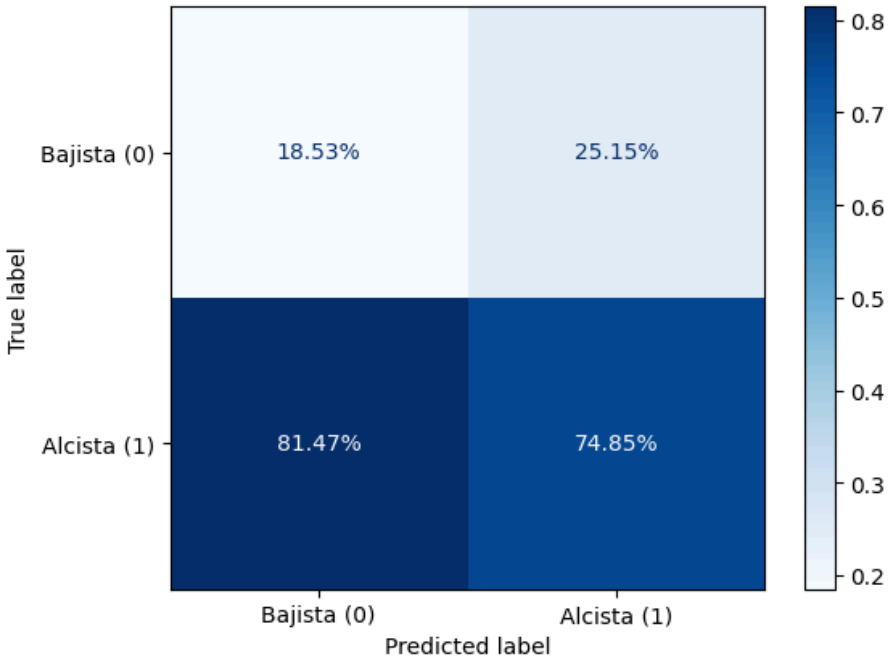


Tabla 10. *Matriz de confusión del modelo AD: exhaustividad (recall) del modelo por condición real.*

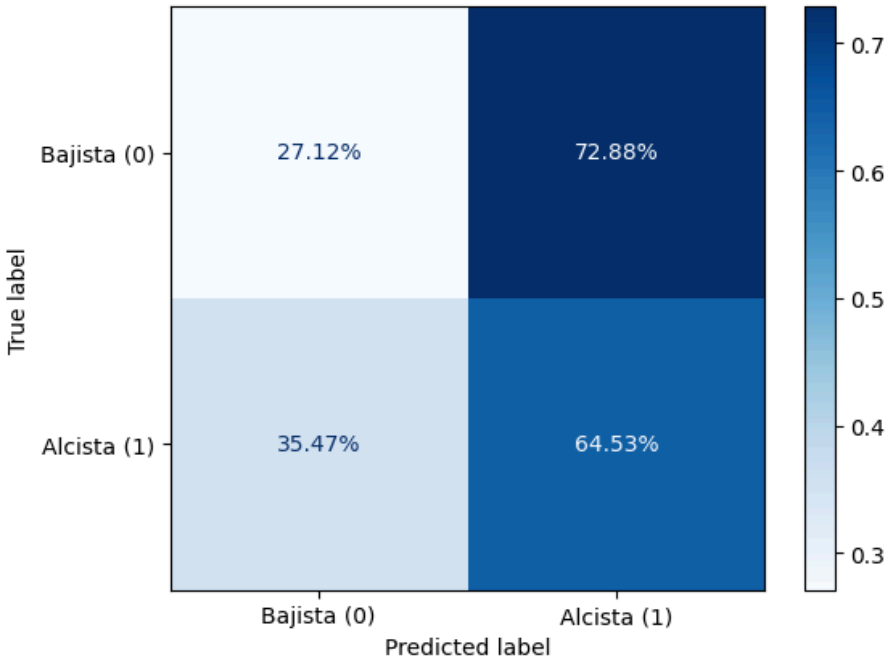


Tabla 11. *Matriz de confusión del modelo RL.*

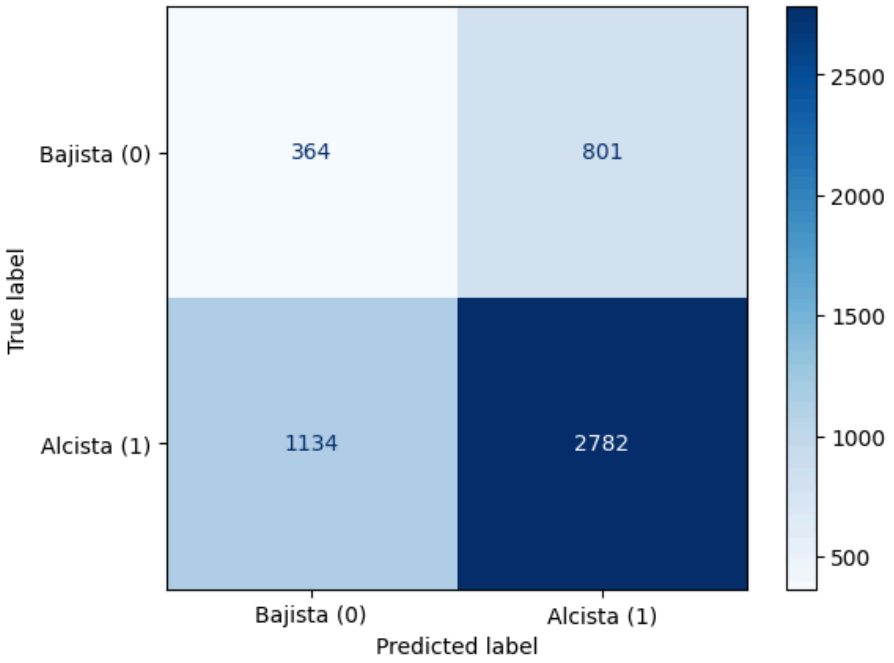


Tabla 12. *Matriz de confusión normalizada al total de la muestra del modelo RL.*

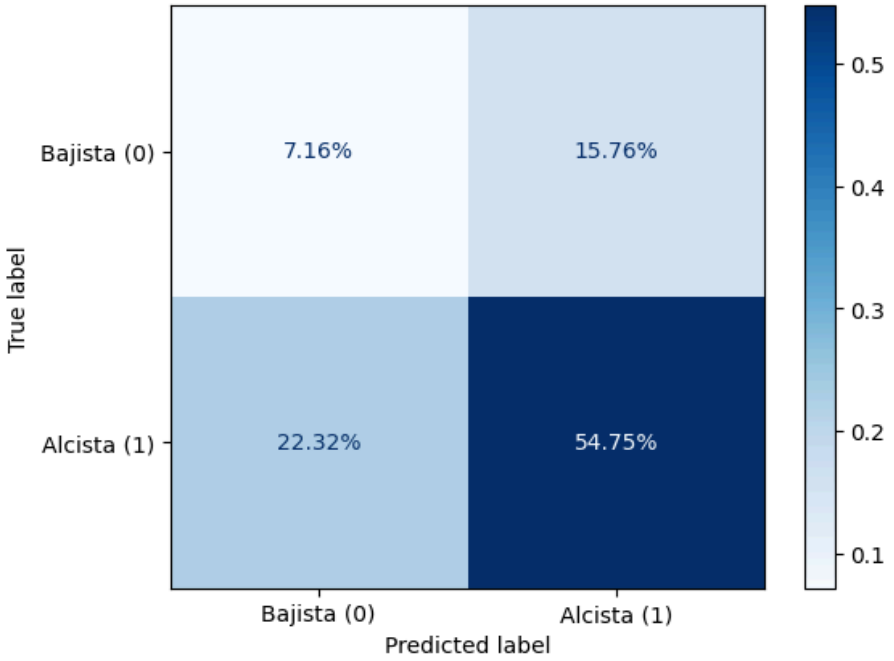


Tabla 13. *Matriz de confusión del modelo RL: precisión de predicción por clase.*

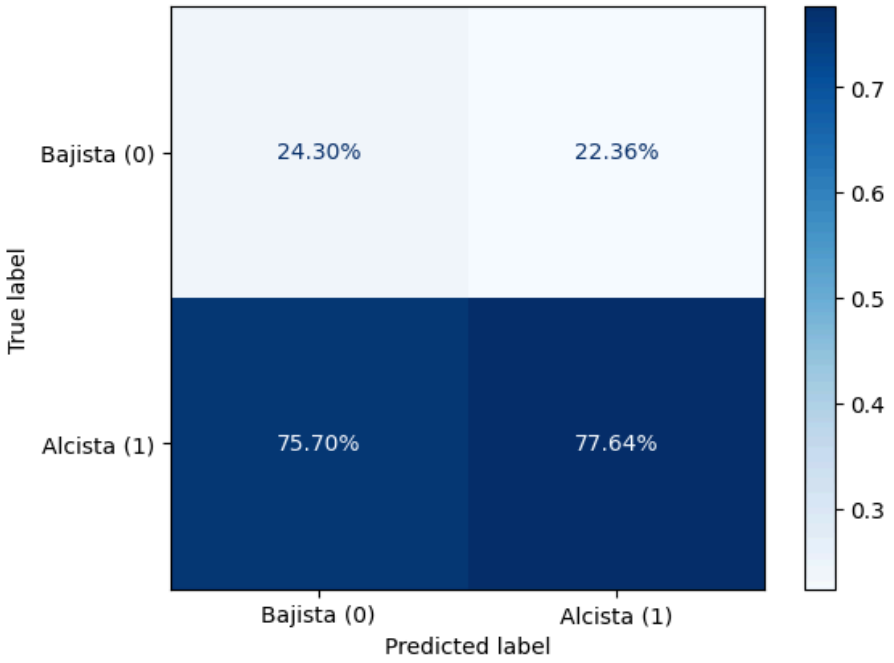


Tabla 14. *Matriz de confusión del modelo RL: exhaustividad (recall) del modelo por condición real.*

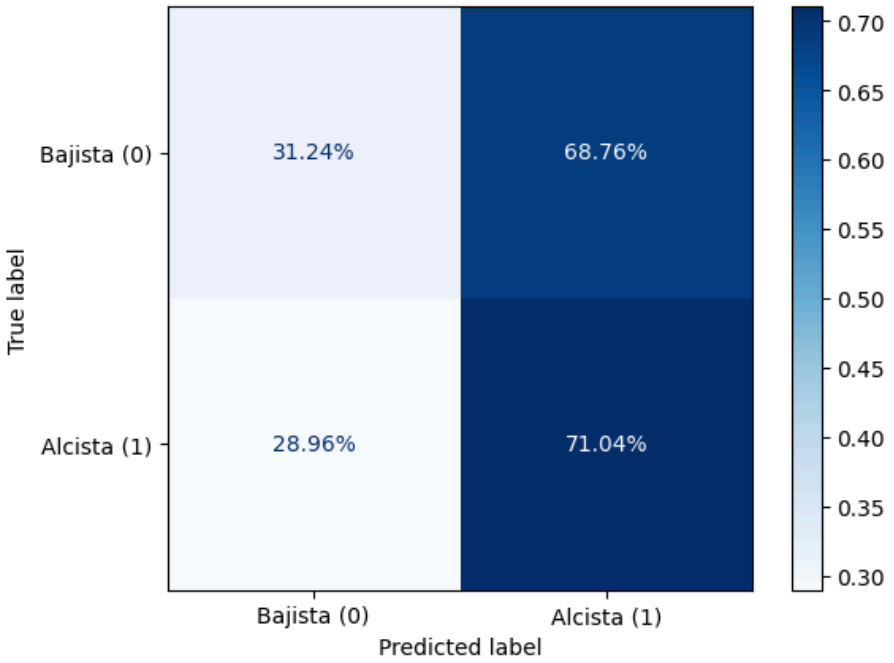


Tabla 15. *Matriz de confusión del modelo BA.*

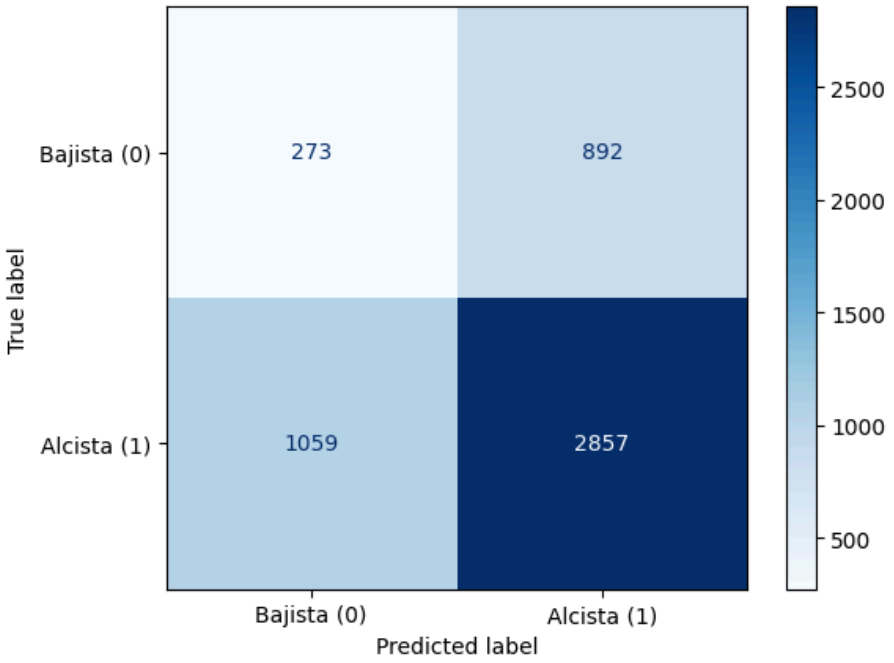


Tabla 16. *Matriz de confusión normalizada al total de la muestra del modelo BA.*

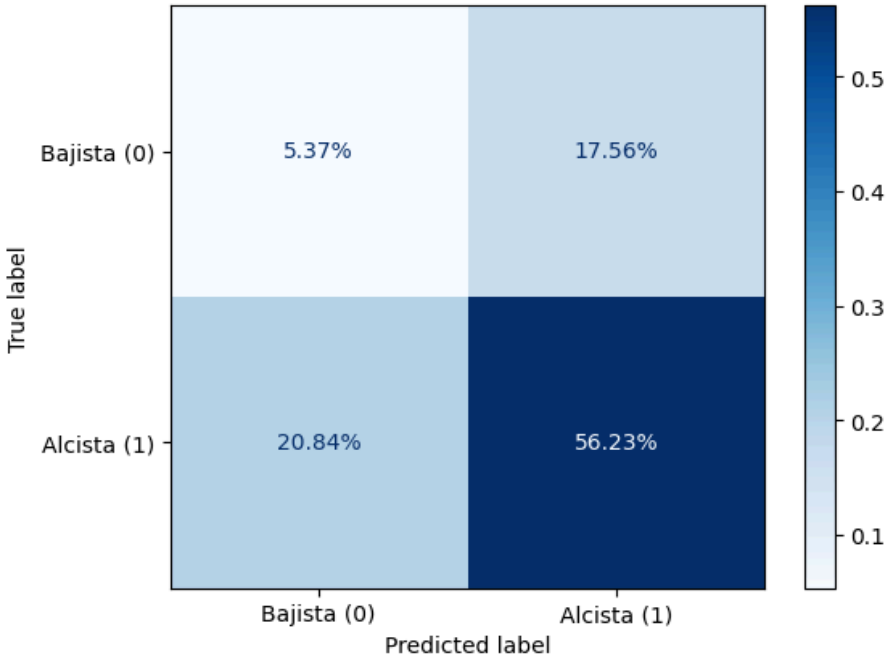


Tabla 17. *Matriz de confusión del modelo BA: precisión de predicción por clase.*

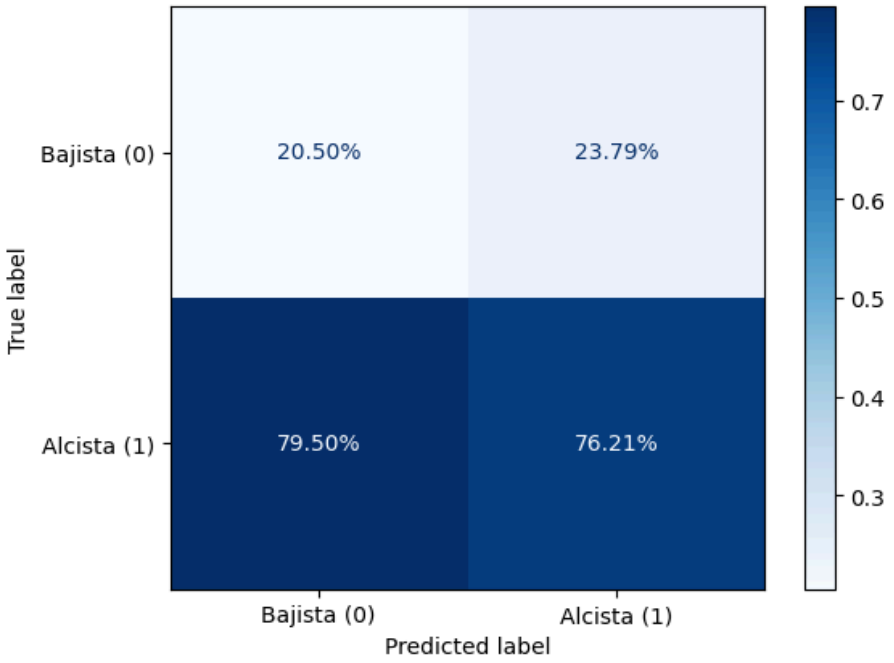
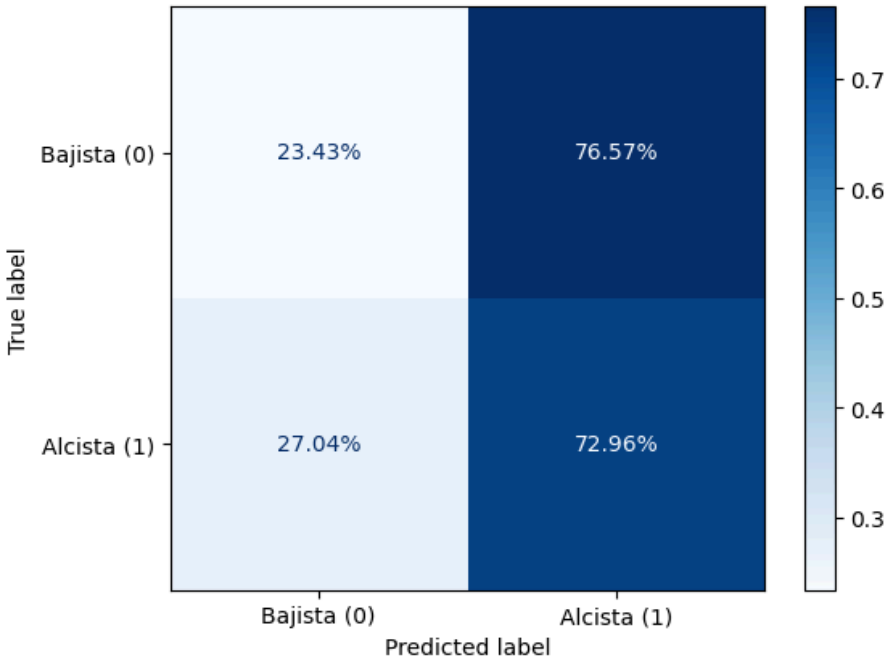


Tabla 18. *Matriz de confusión del modelo BA: exhaustividad (recall) del modelo por condición real.*



Objetivo Específico N° 5

Según el **criterio** de comparación definido en el subtítulo "Validación y Evaluación del Rendimiento de los Modelos" para determinar la **efectividad** y el **mejor rendimiento** entre los tres modelos de clasificación de ML (AD, RL y BA) entrenados y validados para la predicción de la tendencia alcista a 80 días bursátiles de los precios de las acciones de empresas argentinas de servicios financieros con oferta pública en BYMA, se obtuvo la Tabla 19 de los tres algoritmos con sus respectivos valores de métricas, ranqueada desde el encabezado de la tabla hacia el pie de la misma, de mayor a menor efectividad y de mejor a peor rendimiento.

Tabla 19. *Ranking de los algoritmos de clasificación de ML (AD, RL y BA) según su efectividad y rendimiento.*

Modelo	Sesgo (Alcista)	Precisión	Sensibilidad	F1-score	Exactitud	Positivos detectados
1-Bosque Aleatorio	73.78%	76.20%	72.95%	74.54%	61.60%	56.22%
2-Regresión Logística	70.51%	77.64%	71.04%	74.19%	61.91%	54.75%
3-Árbol de Decisión	66.44%	74.85%	64.53%	69.30%	55.95%	49.73%

Rendimiento del Código

Las funciones que procesaron los datos, entrenaron, validaron y persistieron los resultados obtenidos de los tres algoritmos de clasificación de ML (AD, RL y BA) -cuyo objetivo fue la predicción de la tendencia alcista a 80 días bursátiles para las acciones de empresas argentinas de servicios financieros con oferta pública en BYMA- fueron evaluadas desde otra perspectiva, en cuanto a su rendimiento computacional. Para ello, se utilizó una

notebook Hewlett Packard con procesador Intel Core i7 y 16 GB de RAM, ejecutando Python 3.13.7 en un entorno de Visual Studio Code, el cual proporciona compatibilidad con los kernels de Jupyter Notebooks. Las métricas de tiempo y memoria fueron obtenidas mediante los comandos mágicos %time, %timeit y %memit de la biblioteca **`memory_profiler`**. Es muy importante tener en cuenta que los algoritmos de clasificación fueron evaluados con los hiperparámetros ajustados (Con HT) según se ha detallado en el objetivo N°3.

Se utilizó el resultado de **%time** para poder evaluar el rendimiento mediante una visión rápida (cuellos de botella) y una medición precisa (velocidad algorítmica) para el proceso de procesamiento de los datos, entrenamiento, validación y persistencia de los resultados de cada uno de los tres algoritmos:

Tabla 20. *Análisis del cuello de botella (%time - Wall Time vs. CPU Time).*

Algoritmo	CPU Time	Wall Time	Diferencia (Espera)	Interpretación
AD	1.27 s	7.97 s	6.70 s	Limitado por I/O. El proceso pasó ≈84% del tiempo esperando (E/S, lectura de datos, persistencia).
RL	672 ms	772 ms	100 ms	Limitado por CPU. La RL mostró una espera mínima (≈13%), lo que indica que es muy eficiente en el manejo de I/O dentro del proceso.
BA	7.04 s	18.7 s	11.66 s	Limitado por I/O. El BA fue

Algoritmo	CPU Time	Wall Time	Diferencia (Espera)	Interpretación
				el más afectado, pasando ≈62% del tiempo esperando.

La RL es, con diferencia, el algoritmo más eficiente en términos de Wall Time y el que menos tiempo pierde en operaciones de E/S. Los algoritmos AD y BA son lentos debido a la alta dependencia de operaciones de I/O, no de la complejidad del cálculo puro.

Se utilizó el resultado de **%timeit** para reportar el rendimiento promedio más fiable y demostrar la estabilidad de la métrica, lo que refleja la medición y precisión algorítmica para el proceso de procesamiento de los datos, entrenamiento, validación y persistencia de los resultados de cada uno de los tres algoritmos:

Tabla 21. *Medición y precisión algorítmica (%timeit).*

Algoritmo	Tiempo Promedio	Desviación Estándar (std. dev.)	Observación
AD	3.5 s	± 100 ms	Velocidad media, alta estabilidad de la medición.
RL	705 ms	± 107 ms	El más rápido por un gran margen, aunque la desviación es relativamente alta para el valor medio (variación del 15%).
BA	8.49 s	± 1.76 s	El más lento y el más inestable (≈20% de variación), lo que sugiere una mayor sensibilidad a la carga del sistema o a

Algoritmo	Tiempo Promedio	Desviación Estándar (std. dev.)	Observación
			los datos.

La RL es el algoritmo más rápido cuando se aísla el ruido del sistema, demostrando que su lógica interna es la más eficiente para la tarea de predicción. El BA es el más lento, lo cual es predecible debido a que entrena múltiples árboles.

Se utilizó el resultado de **%memit** para cuantificar el impacto del código en la gestión de recursos de la RAM y poder analizar su uso en el proceso de procesamiento de los datos, entrenamiento, validación y persistencia de los resultados de cada uno de los tres algoritmos:

Tabla 22. *Análisis del uso de memoria.*

Algoritmo	Memoria Pico	Incremento Neto	Observación
AD	256.52 MiB	14.42 MiB	Incremento neto moderado.
RL	260.93 MiB	16.06 MiB	Memoria Pico e Incremento ligeramente superior al AD.
BA	267.53 MiB	20.29 MiB	Mayor consumo y mayor incremento. El BA utiliza más memoria de forma persistente.

Los tres algoritmos son eficientes y tienen un consumo de memoria razonable (picos por debajo de 300 MiB). El BA es el más demandante en memoria, lo cual es esperado al requerir almacenar múltiples estructuras de árbol. Todos dejan un incremento neto bajo (< 21 MiB), lo que sugiere que no hay fugas de memoria significativas en el proceso.

El análisis de rendimiento computacional revela que la elección del algoritmo tiene un impacto directo en la velocidad y estabilidad del proceso completo de entrenamiento y validación:

1. La RL es la más eficiente y rápida: Obtiene el mejor rendimiento en tiempo real (Wall Time = 772 ms) y en rendimiento puro (CPU Time = 672 ms). En contextos de despliegue rápido o entrenamiento frecuente, la RL es la opción superior.
2. Los algoritmos AD y BA están limitados por I/O: Su gran Wall Time sugiere que el principal cuello de botella de las funciones de los modelos no está en el entrenamiento y la validación (cálculo), sino en las operaciones de acceso a datos y persistencia de resultados (E/S).
3. El BA es el más costoso: Es el más lento en rendimiento puro (CPU Time = 7.04 s), el más inestable (≈20% de variación con respecto al Tiempo Promedio), y el que consume más memoria (Incremento Neto = 20.29 MiB). Este costo debe justificarse con una ganancia significativamente superior en la precisión predictiva.

Si la precisión entre los modelos es comparable, la RL ofrece el mejor equilibrio entre rendimiento y eficiencia de recursos. Si se requiere el uso del BA por su desempeño superior en la métrica de Sesgo Alcista, la optimización debe enfocarse en reducir los tiempos de E/S del flujo de trabajo de datos.

Discusión

La presente investigación se propuso atender a la carencia de literatura académica en la aplicación y comparación de algoritmos de ML en el mercado de capitales argentino,

específicamente en la predicción de la tendencia alcista a 80 días bursátiles para las acciones de servicios financieros que cotizan en BYMA. Los resultados obtenidos permitieron evaluar la efectividad de los tres modelos de clasificación implementados -Árbol de Decisión (AD), Regresión Logística (RL) y Bosque Aleatorio (BA)-, y determinar cuál se adapta mejor a la dinámica de este mercado de alta volatilidad.

Fortalezas y Limitaciones

La fortaleza de este trabajo radicó en la rigurosa metodología de validación walk-forward y en el tratamiento explícito del data leakage, lo que garantiza la fiabilidad de las métricas obtenidas. Además el aporte o esencia de la contribución de la investigación es la aplicación y comparación rigurosa de algoritmos de ML en un contexto de mercado emergente de alta complejidad. Esto tiene como consecuencia teórica el refuerzo del paradigma de las finanzas **AI-first** o sin modelo, demostrando que los algoritmos como el BA, con un enfoque no paramétrico y basados en datos (Non-parametric and Data-driven), al ser inherentemente theory-free, pueden descubrir ineficiencias estadísticas que los modelos econométricos restrictivos no logran capturar.

Sin embargo, la principal limitación es que la investigación se restringió a la comparación de sólo tres algoritmos de clasificación y se enfocó exclusivamente en el mercado de BYMA y, específicamente, en el sector de servicios financieros, lo que circunscribe la generalización de los resultados a otros mercados emergentes o sectores. Adicionalmente, la alta volatilidad intrínseca del mercado argentino, influenciada por factores macroeconómicos y políticos, representa una dificultad inherente que, aunque los modelos de

ML gestionan mejor que las técnicas clásicas, sigue afectando las métricas como referencias estables para los modelos entrenados comparados.

Análisis de la Efectividad Comparativa: La Paradoja de los Modelos

El análisis comparativo, basado en la Tabla 19 de resultados finales, estableció que los tres algoritmos son efectivos, superando a la elección aleatoria. El BA aseguró el primer puesto del ranking, lo que significa que es el modelo de mejor rendimiento global para el objetivo de esta investigación. Este resultado es consistente con la evidencia empírica que valida la superioridad de los modelos de conjunto (ensemble) en la captura de la no-linealidad de las series financieras (Ampomah, Qin, & Nyame, 2020), donde el BA mitiga la varianza y el potencial sobreajuste de modelos más simples (Wu, 2023).

Autores como Wu (2023), Pashankar et al. (2024), y Lubis y Samsudin (2025) también han reportado que los enfoques basados en Random Forest (Bosque Aleatorio) superan consistentemente a modelos más sencillos como la Regresión Lineal o Logística en tareas de predicción de precios o tendencias. La robustez del Bosque Aleatorio, lograda mediante el bagging y la aleatorización de características, resultó ser esencial para manejar la alta dimensionalidad y la complejidad no-lineal de las series de precios de las acciones de las siete empresas de servicios financieros que cotizan en BYMA.

Sin embargo, la clasificación final revela una paradoja crítica en la comparación de rendimiento que debe ser expuesta, y que constituye el principal punto de discusión de los resultados:

1. **RL y la Exactitud y Precisión**: Se observó que la RL es ligeramente superior al BA en las métricas de Exactitud y Precisión. Este resultado anómalo, donde un modelo lineal casi igualó a un modelo ensemble en métricas clave, sugiere que la relación subyacente entre los 19 predictores técnicos y la variable objetivo posee un componente de linealidad a considerar sobre las muestras que se utilizaron para el entrenamiento y validación.
2. **BA y el Sesgo Alcista**: La decisión de otorgar el primer puesto al BA se justifica por su desempeño superior en la métrica de Sesgo Alcista (según el Criterio preestablecido para Ranquear). A pesar de su ínfima inferioridad en Exactitud y Precisión, el BA presentó un Sesgo Alcista más aceptable, lo que indica que su predicción de la frecuencia de la clase positiva (alcista) se alinea de manera más realista con la frecuencia real de la muestra. Este control del sesgo es esencial para la implementación práctica, ya que evita la sobreestimación o subestimación de las oportunidades de compra, cumpliendo así con el objetivo predictivo.

Similares resultados se observan en el estudio de Ayyildiz e Iskenderoglu (2024) sobre la predicción de movimientos direccionales en siete índices bursátiles de países desarrollados. Los autores compararon algoritmos como la Regresión Logística, el Árbol de Decisión, el Bosque Aleatorio y las Redes Neuronales Artificiales (RNA). Aunque encontraron que las RNA exhibieron el mayor rendimiento promedio, determinaron que la Regresión Logística fue el mejor algoritmo para índices específicos como el NIKKEI 225, el CAC 40, y el TSX , demostrando que la efectividad superior de un algoritmo lineal aún puede presentarse en ciertos entornos de mercado.

Consecuencias Teóricas y Aplicaciones Prácticas: El Rendimiento del Código

Las consecuencias teóricas de este ranking (BA > RL) son que, si bien los modelos de conjunto son superiores en la capacidad de generalización (control del Sesgo Alcista), un modelo simple como la RL puede alcanzar un rendimiento de Precisión casi idéntico en mercados con dinámicas potencialmente más sencillas o impulsadas por la magnitud de las features.

A nivel de aplicaciones prácticas, el análisis del Rendimiento del Código impone un **trade-off fundamental**:

- BA ofrece la máxima confiabilidad (gracias al valor de la métrica del Sesgo Alcista muy próximo al valor de la muestra de validación) a cambio de un mayor costo computacional (lento entrenamiento y validación).
- RL ofrece una Precisión máxima a cambio de un costo computacional mínimo (algoritmo ligero y rápido). Es necesario considerar que una alta Precisión es a menudo priorizada para minimizar las pérdidas asociadas a señales de compra fallidas.

Por lo tanto, en escenarios de inversión donde la latencia de predicción es crítica o donde los recursos de hardware son limitados, la RL se presenta como una solución de eficiencia óptima, siendo el modelo práctico a elegir. Este dilema entre Precisión y Rendimiento del Código es una implicación directa y valiosa para la toma de decisiones tecnológicas en el mercado de capitales argentino.

Conclusión

La investigación cumplió con su objetivo general, permitiendo la implementación, entrenamiento y validación de los tres modelos de clasificación de ML para la predicción de la tendencia alcista a 80 días bursátiles de los precios de acciones de servicios financieros en BYMA. El logro de los objetivos específicos permitió: (1) recolectar y procesar los datos históricos de las acciones; (2) construir los 19 predictores/indicadores necesarios; (3) implementar, entrenar y ajustar los hiperparámetros de los modelos del AD, RL y BA, (4) validar su rendimiento utilizando un riguroso análisis walk-forward, y (5) compararlos mediante el criterio de ranking preestablecido usando métricas relevantes.

El trabajo se sintetiza en las siguientes conclusiones que responden directamente a los interrogantes de la investigación:

- **Efectividad de los Algoritmos** (Respuesta al primer interrogante): Los algoritmos de clasificación de RL, AD y BA son efectivos en diferentes grados para predecir la tendencia alcista a 80 días bursátiles, lo que demuestra la existencia de patrones predictivos extraíbles en este segmento del mercado argentino (las medidas se pueden observar en la Tabla 19 y la presentación de resultados de los subtítulos Objetivo Específico N° 4 y Rendimiento del Código).

- **Mejor Rendimiento** (Respuesta al segundo interrogante): Entre los tres algoritmos, el BA ofrece el mejor rendimiento general según el criterio de ranking preestablecido. El BA es el modelo más confiable y robusto para la predicción de la tendencia alcista a 80 días bursátiles de los precios de las acciones argentinas de servicios financieros

cotizantes en BYMA, en términos de todas las métricas utilizadas en el criterio para comparar (ranquear) los algoritmos de clasificación según su objetivo predictivo.

La principal conclusión que se deriva es que el algoritmo BA se estableció como el modelo más efectivo (primer puesto en el ranking) para la predicción de la tendencia alcista a 80 días bursátiles para las acciones de servicios financieros en BYMA. Esta investigación se constituye como un aporte esencial al aplicar y validar métodos de ML directamente en el contexto geográfico y sectorial del mercado argentino, atendiendo a la carencia de literatura académica sobre el tema específico.

Principales Hallazgos e Implicaciones Prácticas

Estos hallazgos atienden a la carencia de literatura académica específica al aplicar y comparar estos algoritmos directamente en el contexto geográfico y sectorial del mercado argentino.

El modelo identificado (BA) no sólo proporciona un nuevo conocimiento académico sobre la predictibilidad de este mercado, sino que también ofrece un aporte de valor práctico fundamental, sirviendo como una base para el desarrollo de herramientas predictivas que pueden mejorar la toma de decisiones estratégicas de analistas e inversores en la identificación de oportunidades condicionadas por la tendencia alcista del sector financiero a mediano plazo (80 días bursátiles).

Además, la exploración de los resultados obtenidos, ejemplifica cómo los modelos de aprendizaje automático (ML), al realizar un procesamiento avanzado de variables financieras,

constituyen una herramienta de significativa capacidad predictiva para anticipar movimientos en el contexto del mercado bursátil local.

Aportes y Recomendaciones para Futuros Estudios

La principal contribución de la investigación es la validación empírica en el contexto geográfico y sectorial del mercado argentino de que el Bosque Aleatorio es superior a la RL y AD, aunque se descubre que la Regresión Logística ofrece un rendimiento casi idéntico en las métricas con una eficiencia computacional radicalmente superior.

Se recomienda que los estudios futuros profundicen en las siguientes líneas de investigación:

- **Modelos avanzados y comparación de eficiencia**: Ampliar la comparación para incluir modelos de ensemble más avanzados, como XGBoost o Extra Trees Classifier (Ampomah et al., 2020), que han demostrado un rendimiento superior en ciertos contextos de predicción de precios de acciones, y realizar un análisis de costo-beneficio formal que pondere la ganancia marginal de precisión predictiva del BA frente al costo de latencia de la RL, para ofrecer una recomendación definitiva sobre la implementación del modelo más eficiente en una aplicación específica.

- **Aportes de datos no estructurados**: El análisis se basó exclusivamente en indicadores de precios y volúmenes (datos técnicos). Una línea de investigación valiosa sería incorporar variables cualitativas o features derivadas de datos no estructurados, como el análisis de sentimiento de noticias financieras y redes sociales,

para evaluar si estos factores exógenos mejoran el rendimiento predictivo de la clasificación.

- **Extensión Sectorial y del Muestreo**: La generalización de los resultados está limitada al sector de servicios financieros de BYMA. Sería beneficioso replicar el modelo en otros sectores (ej.: energía, agro, etc.) que cotizan en el mismo mercado para determinar la portabilidad y la universalidad de la efectividad de los modelos. Con este enfoque también sería valioso expandir la investigación a nuevas muestras del mismo sector para realizar pruebas de los modelos.

Agradecimientos

De manera muy especial y profundamente personal, deseo expresar mi gratitud al Ing. Juan Pablo Pisano. Más allá de coincidir con él desde su rol de docente, el poder relacionarnos a través del proceso de aprendizaje sobre el gran mundo de la inteligencia artificial hizo que se convirtiera en el motor conceptual y la base analítica sobre la que se construyó parte de esta investigación. Fue su enfoque pragmático, conceptual y distinto al enfoque tradicional que dotó de singularidad a este trabajo final de grado. Su influencia marcó un antes y un después en mi desarrollo, transformando mi comprensión y aplicación práctica de áreas troncales utilizadas en la investigación. Logré asimilar y aplicar cada temática desde una perspectiva práctica y conceptual, acortando esa brecha entre la teoría y la realidad que a veces propone el sistema educativo, manteniendo el foco en la visión que me inspiró. Muchas gracias Juanpi, por tu influencia constante, paciencia, generosidad y por tu invaluable rol como mentor quant. Este trabajo es un pequeño reflejo de cómo algunas personalidades hacen la diferencia en la vida de otras personas.

Referencias

Abdulhafedh, A. (2022). Comparison between Common Statistical Modeling Techniques Used in Research, Including: Discriminant Analysis vs Logistic Regression, Ridge Regression vs LASSO, and Decision Tree vs Random Forest. *Open Access Library Journal, 9*: e8414. https://doi.org/10.4236/oalib.1108414

Aco Tito, A. E., Hancco Condori, B. O., & Pérez Vera, Y. (2023). Análisis comparativo de Técnicas de Machine Learning para la predicción de casos de deserción universitaria. *Revista Ibérica de Sistemas e Tecnologias de Informação, (51)*, 84–98. doi:10.17013/risti.51.84-98

Ampomah, E. K., Qin, Z., & Nyame, G. (2020). Evaluation of Tree-Based Ensemble Machine Learning Models in Predicting Stock Price Direction of Movement. *Information, 11*(6), 332. https://doi.org/10.3390/info11060332

Ayyildiz, N., & Iskenderoglu, O. (2024). How effective is machine learning in stock market predictions?, *Heliyon, 10*(1), e24123. https://doi.org/10.1016/j.heliyon.2024.e24123

Brandimarte, P. (2014). *Handbook in Monte Carlo Simulation: Applications in Financial Engineering, Risk Management, and Economics*. Wiley.

Carpio Vargas, E. E., Mayda-Huanca, A. R., Espinoza, G., Mamani Vilca, E., & Ccolqque Ruiz, B. M. (2023). La mejor opción para predecir el precio máximo de las acciones de Intel Corporation: ¿Árbol de regresión o regresión lineal múltiple?. *RIQCHARY: Revista de investigación en ciencias contables, financieras y administrativas, 5*(1), 117-124. Recuperado de https://www.researchgate.net/publication/374984543_La_mejor_opcion_para_predecir_el_precio_maximo_de_las_acciones_de_Intel_Corporation_Arbol_de_regresion_o_Regresion_Lineal_Multiple

Catalán, M., Deghi, A., & Qureshi, M. S. (2024, 15 de octubre). *Cómo la alta incertidumbre económica podría amenazar la estabilidad financiera mundial*. Blog del FMI. Recuperado de https://www.imf.org/es/Blogs/Articles/2024/10/15/how-high-economic-uncertainty-may-threaten-global-financial-stability

Cattlin, C. (2023, 19 de abril). *¿Qué es la volatilidad de los mercados financieros?*. FOREX.com. Recuperado de https://www.forex.com/es/news-and-analysis/volatilidad-en-mercados-financieros/

Cómo implementar un modelo de clasificación con Python. (2025, 25 de mayo). El Mundo de los Datos. Recuperado de https://elmundodelosdatos.com/como-implementar-un-modelo-de-clasificacion-con-python/

Dev, A. (2024, 22 de junio). *Data Science: Understanding Random Forest Machine Learning Model for Stock Price Prediction with Bajaj Finance Stock Data*. Medium. Recuperado de https://medium.com/@nikaljeajay36/data-science-understanding-random-forest-machine-learning-model-for-stock-price-prediction-with-88d1b2f93047

Ellson, J., Gansner, E., Koutsofios, L., North, S. C., & Woodhull, G. (2002). Graphviz—open source graph drawing tools. *Lecture Notes in Computer Science, 2265*, 483-490.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance, 25*(2), 383-417. doi:10.2307/2325486

Gómez Plaza, S. D. (2025). *Valoración de alternativas de inversión en el mercado de renta fija colombiano con algoritmos de machine learning* [Tesis de maestría, Universidad EAFIT]. Repositorio Institucional Universidad EAFIT. Recuperado de

https://repository.eafit.edu.co/server/api/core/bitstreams/98453bbc-9590-4003-b7f8-287d1578e05f/content

Heterocedasticidad y curtosis: evaluación de las desviaciones de la normalidad. (2025, 30 de abril). Recuperado de https://fastercapital.com/es/contenido/Heterocedasticidad-y-curtosis--evaluacion-de-las-desviaciones-de-la-normalidad.html

Hilpisch, Y. (2021). *Artificial Intelligence in Finance: A Python-Based Guide*. O'Reilly Media, Inc.

López de Prado, M. (2018). *Advances in financial machine learning*. Wiley.

Lubis, M. A., & Samsudin, S. (2025). Using the Random Forest Method in Predicting Stock Price Movements. *DINDA, 5*(1). https://doi.org/10.20895/dinda.v5i1.1765

Malkiel, B. G. (2003). *The efficient market hypothesis and its critics. Journal of Economic Perspectives, 17*(1), 59-82. doi:10.1257/089533003321164958

Mercados emergentes explicados. (2025, 16 de abril). Plus500. Recuperado de https://www.plus500.com/es/newsandmarketinsights/emerging-markets-explained

MIOTI. (2024, 24 de octubre). *Cómo invertir en Bolsa usando Inteligencia Artificial*. MIOTI. Recuperado de https://mioti.es/es/blog-como-invertir-en-bolsa-usando-inteligencia-artificial/

NumPy Developers (2024). *NumPy 2.3.3 Release Notes*. NumPy. Recuperado de https://numpy.org/devdocs/release/2.3.3-notes.html

Pandas Development Team. (2025). *What's new in 2.3.3* (versión 2.3.3). Pandas. Recuperado de https://pandas.pydata.org/docs/whatsnew/v2.3.3.html

Pashankar, S., Shendage, J., & Pawar, J. (2024). Machine Learning Techniques For Stock Price Prediction - A Comparative Analysis Of Linear Regression, Random Forest, And Support Vector Regression. *Journal of Advanced Zoology, 45*(S-4), 118–127.

Recuperado de https://www.researchgate.net/publication/378836384_Machine_Learning_Techniques_For_Stock_Price_Prediction_-_A_Comparative_Analysis_Of_Linear_Regression_Random_Forest_And_Support_Vector_Regression

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

Pérez, F., & Granger, B. E. (2007). IPython: A system for interactive scientific computing. *Computing in Science & Engineering, 9*(3), 21-29.

¿Por qué invertir en mercados emergentes? (2023, 19 de abril). UBS Global Wealth Management. Recuperado de https://www.ubs.com/global/es/wealthmanagement/latamaccess/market-updates/articles/por-que-invertir-mercados-emergentes.html

¿Qué es la volatilidad de mercado y cómo funciona? (s.f.). Capital.com. Recuperado de https://capital.com/es-int/learn/essentials/market-volatility

Reitz, K. (2023). *Requests: HTTP for Humans* (v. 2.32.5) [Software de computación]. Recuperado de https://requests.readthedocs.io

Serrano Bautista, R., & Mata Mata, L. (2018). Valor en Riesgo mediante un modelo heterocedástico condicional α-estable. *Revista mexicana de economía y finanzas, 13*(1). https://doi.org/10.21919/remef.v13i1.257

Shiller, R. J. (2003). From efficient markets theory to behavioral finance. *Journal of Economic Perspectives, 17*(1), 83-104. doi:10.1257/089533003321164967

Tatsat, H., Puri, S. y Lookabaugh, B. (2021). *Machine Learning and Data Science Blueprints for Finance: From Building Trading Strategies to Robo-Advisors Using Python*. O'Reilly Media, Inc.

The Matplotlib Development Team. (2025). *Matplotlib* (Versión 3.10.6) [Software de computación]. Recuperado de https://matplotlib.org

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods, 17*(3), 261-272.

Waskom, M. L. (2024). *mwaskom/seaborn: v0.13.2* [Software de computación]. doi:10.5281/zenodo.10651324

Wu, Y. (2023). Stock Price Prediction Based on Simple Decision Tree Random Forest and XGBoost. *BCP Business & Management, 38,* 3383-3388. https://doi.org/10.54691/bcpbm.v38i.4311

Zheng, J., Xin, D., Cheng, Q., Tian, M., & Yang, L. (2024). *The Random Forest Model for Analyzing and Forecasting the US Stock Market in the Context of Smart Finance*. Recuperado de arXiv. https://doi.org/10.48550/arXiv.2402.17194

Apéndice

Configuraciones Técnicas de los Modelos al Inicio y al Final del HT

Para garantizar la transparencia y replicabilidad del experimento, se establecieron configuraciones base para realizar el entrenamiento de cada uno de los algoritmos seleccionados. En la Tabla A1 se detallan las especificaciones técnicas **iniciales** empleadas para el AD, la RL y el BA. Posteriormente, estos valores sirvieron como punto de partida para el proceso de Ajuste de Hiperparámetros (HT). La Tabla A2 expone la configuración **final** obtenida mediante la búsqueda de hiperparámetros.

Tabla A1. *Configuración inicial de los modelos entrenados sin HT.*

DecisionTreeClassifier()	**LogisticRegression**()	**RandomForestClassifier**()
criterion = 'entropy' splitter = 'best' max_depth = 9 min_samples_split = 2 min_samples_leaf = 1 min_weight_fraction_leaf = 0.0 max_features = None random_state = 0 max_leaf_nodes = None min_impurity_decrease = 0.0 class_weight = None ccp_alpha = 0.0 monotonic_cst = None	penalty = 'l2' dual = False tol = 0.0001 C = 1.0 fit_intercept = True intercept_scaling = 1 class_weight = None random_state = 0 solver = 'lbfgs' max_iter = 100 multi_class = 'deprecated' verbose = 0 warm_start = False n_jobs = None l1_ratio = None	n_estimators = 500 criterion = 'entropy' max_depth = 9 min_samples_split = 2 min_samples_leaf = 1 min_weight_fraction_leaf = 0.0 max_features = 5 max_leaf_nodes = None min_impurity_decrease = 0.0 bootstrap = True oob_score = False n_jobs = -1 random_state = 0 verbose = 0 warm_start = False class_weight = None ccp_alpha = 0.0 max_samples = None monotonic_cst = None

Nota. Los hiperparámetros detallados corresponden a los argumentos (parameters) de las clases de la librería Scikit-Learn v. 1.7.2 para el lenguaje Python **(Pedregosa et al., 2011)**.

Tabla A2. *Configuración final de los modelos entrenados con HT.*

DecisionTreeClassifier()	**LogisticRegression**()	**RandomForestClassifier**()
criterion = 'gini' splitter = 'best' max_depth = 9 min_samples_split = 2 min_samples_leaf = 1 min_weight_fraction_leaf = 0.0 max_features = None random_state = 0 max_leaf_nodes = None min_impurity_decrease = 0.0 class_weight = None ccp_alpha = 0.0 monotonic_cst = None	penalty = 'l2' dual = False tol = 0.0001 C = 1.0 fit_intercept = True intercept_scaling = 1 class_weight = 'balanced' random_state = 0 solver = 'newton-cholesky' max_iter = 124 multi_class='deprecated' verbose = 0 warm_start = False n_jobs = None l1_ratio = None	n_estimators = 150 criterion = 'gini' max_depth = 9 min_samples_split = 2 min_samples_leaf = 1 min_weight_fraction_leaf = 0.0 max_features = 4 max_leaf_nodes = None min_impurity_decrease = 0.0 bootstrap = True oob_score = False n_jobs = -1 random_state = 0 verbose = 0 warm_start = False class_weight = None ccp_alpha = 0.0 max_samples = 0.85 monotonic_cst = None

Nota. Los hiperparámetros detallados corresponden a los argumentos (parameters) de las clases de la librería Scikit-Learn v. 1.7.2 para el lenguaje Python (Pedregosa et al., 2011).