Facultad de Ciencias Económicas y Empresariales
ICADE

MODELOS DE MACHINE LEARNING PARA LA PREDICCIÓN DE RENTABILIDAD EMPRESARIAL: UN ANÁLISIS DEL ÍNDICE S&P 500

Clave: 202002653
Autor: Beatriz Tejedor Canet
Director: Carlos Miguel Vallez Fernández

MADRID | Junio 2025

RESUMEN:

Este estudio examina el uso de técnicas de Machine Learning (ML) para poder predecir la rentabilidad futura de empresas del índice S&P 500, utilizando modelos de clasificación binaria. Se crearon varios algoritmos de predicción, entre ellos regresión logística, árboles de decisión, random forest, gradient boosting y redes neuronales, aplicados a un conjunto de datos financieros. La base de datos incluye datos de más de 500 compañías y abarca tanto variables financieras del año 2023 como datos históricos de cotización correspondientes al período de 2013 a 2018.

El objetivo principal de esta investigación es determinar qué modelo ofrece la mayor precisión predictiva, ofreciendo de esta manera una herramienta valiosa para empresas de private equity e inversores en la identificación ágil y automatizada de oportunidades de inversión. Como objetivo complementario, se analizará el efecto de diversas variables financieras y sectoriales en la rentabilidad, mediante un estudio exploratorio enfocado en identificar los elementos más cruciales en el rendimiento futuro de la empresa.

La metodología del estudio adopta un enfoque cuantitativo, que comprende un proceso exhaustivo de extracción, transformación y carga (ETL), limpieza y normalización de datos, análisis exploratorio y entrenamiento de modelos con técnicas como validación cruzada y ajuste de hiperparámetros. Se evaluarán los modelos empleando métricas de rendimiento como precisión, sensibilidad, especificidad, AUC-ROC y matriz de confusión, para comparar el rendimiento relativo de los modelos.

Este análisis no solo favorece la creación de instrumentos predictivos eficientes en el sector financiero, sino que también inaugura nuevas áreas de investigación en el empleo de algoritmos sofisticados —como modelos de aprendizaje profundo— para incrementar la exactitud, capacidad de adaptación y utilidad de las predicciones en diversos contextos económicos y sectores de negocios.

PALABRAS CLAVE

Machine Learning, Modelos de Clasificación, Predicción Financiera, Árboles de Decisión, Logit Regression, Gradient Boosting, Redes Neuronales, ETL (Extracción, Transformación y Carga), Rentabilidad Empresarial, S&P 500, Inversión, Evaluación de Modelos, Private Equity, Análisis Exploratorio.

ABSTRACT

This study examines the use of Machine Learning (ML) techniques to predict the future profitability of companies in the S&P 500 index, using binary classification models. Several predictive algorithms were developed, including logistic regression, decision trees, random forest, gradient boosting, and neural networks, applied to a financial dataset. The database includes information on more than 500 companies and covers both financial variables from the year 2023 and historical stock price data corresponding to the period from 2013 to 2018.

The main objective of this research is to determine which model offers the highest predictive accuracy, thus providing a valuable tool for private equity firms and investors in the agile and automated identification of investment opportunities. As a complementary objective, the effect of various financial and sectoral variables on profitability will be analysed through an exploratory study focused on identifying the most crucial elements in the future performance of the company.

The methodology of the study adopts a quantitative approach, involving an exhaustive process of extraction, transformation and loading (ETL), data cleaning and normalisation, exploratory analysis and model training with techniques such as cross-validation and hyperparameter fitting. Models will be evaluated using performance metrics such as accuracy, sensitivity, specificity, AUC-ROC and confusion matrix to compare the relative performance of models.

This analysis not only favours the creation of efficient predictive tools in the financial sector, but also opens up new areas of research in the use of sophisticated algorithms - such as deep learning models - to increase the accuracy, adaptability and usefulness of predictions in various economic contexts and business sectors.

KEY WORDS:
Machine Learning, Classification Models, Financial Forecasting, Decision Trees, Logit Regression, Gradient Boosting, Neural Networks, ETL (Extraction, Transformation and Loading), Corporate Profitability, S&P 500, Investment, Model Evaluation, Private Equity, Exploratory Analysis.

ÍNDICE DEL CONTENIDO:

ÍNDICE DE TABLAS Y GRÁFICOS:

1. INTRODUCCIÓN

1.1 OBJETIVOS

El objetivo principal de este trabajo es crear y examinar modelos predictivos de clasificación basados en técnicas de Machine Learning (ML), con el fin de determinar cuál de estos muestra un rendimiento superior en la proyección de la rentabilidad futura de las compañías que integran el índice S&P 500. Se pretende comparar la efectividad de diversos algoritmos —tales como árboles de decisión, random forest, gradient boosting, regresión logística y redes neuronales— y seleccionar aquel que proporcione los mejores resultados en cuanto a precisión, robustez y capacidad para generalizar. Se espera que los modelos generados constituyan una herramienta estratégica de alto valor para empresas de private equity e inversores, facilitando la automatización del análisis financiero y la identificación eficiente de oportunidades de inversión y adquisición.

Como objetivo complementario, se busca analizar el impacto de distintas variables financieras y sectoriales en la capacidad predictiva de los modelos. A través de un análisis exploratorio riguroso, se pretende identificar los factores más determinantes en el rendimiento futuro de las empresas del S&P 500. Se espera que variables como los ratios de rentabilidad, apalancamiento financiero o crecimiento de ingresos presenten una influencia significativa sobre las predicciones, permitiendo construir un perfil más preciso de las características asociadas a empresas de alto desempeño. De este modo, el estudio aspira a aportar un enfoque integral que ayude a los agentes financieros a fundamentar sus decisiones en patrones consistentes y datos empíricamente validados.

1.2 JUSTIFICACIÓN DEL TEMA

En una era marcada por rápidos avances tecnológicos y una economía en constante evolución, el Machine learning (ML) se ha convertido en un pilar fundamental para el análisis predictivo y para la toma decisiones.

El aprendizaje supervisado, sin embargo, ha sido clave para predecir la dinámica de las empresas, especialmente al aplicar algoritmos para predecir el crecimiento, desempeño y salida de empresas al mercado. Según Bargagli-Stoffi, Niederreiter y Riccaboni (2021), los algoritmos de aprendizaje supervisado son especialmente efectivos en contextos

empresariales, ya que pueden captar relaciones no lineales complejas en datos con alta dimensionalidad, ofreciendo así ventajas frente a modelos econométricos más tradicionales.

Al centrarse en el índice S&P 500, este análisis se basa en la información financiera exhaustiva de las compañías más influyentes de Estados Unidos. Estas compañías, constituyen una parte considerable del valor total del mercado y juegan un rol fundamental en la economía mundial.

Desde la automatización de análisis complejos hasta la creación de percepciones predictivas a gran escala, el Machine Learning no solo potencia la eficiencia en los negocios, sino que también cambia nuestra forma de relacionarnos con los datos, facilitando una toma de decisiones proactiva. Además, estos algoritmos son líderes en predecir la quiebra y el riesgo financiero. Según Alanis, Chava & Shah (2022) "los algoritmos de ML parecen ser particularmente adecuados para la predicción de quiebras, ya que pueden manejar un gran número de covariables, sobresalir en descubrir relaciones no lineales complejas y seleccionar predictores importantes mediante un enfoque basado en datos, sin depender de las creencias previas de los investigadores" (p. 1).

Asimismo, la integración de modelos de redes neuronales artificiales (ANN) con indicadores financieros y macroeconómicos ha demostrado ser particularmente efectiva para proyectar el rendimiento financiero. Estos modelos, especialmente aquellos con múltiples capas ocultas, han mostrado una capacidad significativa para capturar relaciones complejas y no lineales entre variables clave, como los precios históricos de los activos y factores macroeconómicos, mejorando sustancialmente la precisión en las predicciones financieras (Instituto O'Higgins, 2025).

El propósito de este estudio es desarrollar modelos predictivos utilizando diversos métodos de clasificación, con el objetivo de anticipar la rentabilidad futura de estas empresas líderes. Estos modelos de Machine Learning proporcionarán a las empresas e inversores una base sólida para tomar decisiones más informadas y estratégicas. En un contexto donde la automatización es clave, estos modelos no solo permitirán decisiones más precisas y bien fundamentadas, sino que también agilizarán el proceso, ahorrando tiempo y optimizando recursos para las organizaciones.

1.3 METODOLOGÍA

Este análisis se basa en un enfoque cuantitativo, utilizando técnicas de Machine Learning (ML) para predecir y clasificar la rentabilidad futura de las compañías que integran el índice S&P 500. El procedimiento metodológico se organiza en diversas etapas secuenciales, diseñadas para optimizar la preparación de los datos y garantizar la precisión y fiabilidad de los modelos desarrollados.

En primer lugar, se llevó a cabo el proceso de extracción, transformación y carga de datos (ETL). Para ello, se integraron fuentes de información financiera y bursátil que combinan datos actuales y registros históricos. Esta fase incluyó la limpieza de datos, la eliminación de valores atípicos y duplicados, así como la normalización y estandarización de las variables financieras relevantes, asegurando la coherencia y comparabilidad entre las observaciones.

Posteriormente, se realizó un análisis exploratorio con el propósito de comprender la estructura del conjunto de datos e identificar posibles redundancias o correlaciones relevantes entre las variables. Además, se definió la variable objetivo del estudio, de naturaleza binario, que clasifica a las compañías según si su rentabilidad futura supera determinados umbrales establecidos, facilitando la posterior tarea de clasificación.

Una vez preparados los datos, se llevó a cabo la creación de los modelos predictivos utilizando algoritmos de Machine Learning. Entre los métodos utilizados se encuentran árboles de decisión, random forest, gradient boosting, regresión logística y redes neuronales. Estos modelos fueron entrenados y ajustados mediante técnicas de validación cruzada y optimización de hiperparámetros, con el objetivo de maximizar su rendimiento y evitar problemas de sobreajuste.
La elaboración de los modelos se basó en un método iterativo de mejora continua, inspirado en los principios Agile, que facilitó la efectuación de modificaciones graduales en base a los resultados alcanzados en cada ciclo de evaluación.

Finalmente, se llevó a cabo una etapa de evaluación completa, donde se analizaron los resultados a través de métricas clásicas de clasificación, como la precisión (accuracy), la sensibilidad (recall), la especificidad, el área bajo la curva ROC (AUC-ROC) y la matriz

de confusión. Estas métricas proporcionaron una base objetiva para comparar el desempeño de los diferentes algoritmos y seleccionar el modelo más adecuado para predecir la rentabilidad empresarial futura. Además, se validaron los modelos empleando grupos de datos independientes, garantizando su habilidad para generalizarse en situaciones reales.

1.4 ESTRUCTURA

La estructura ha sido cuidadosamente diseñada para garantizar claridad y facilitar la comprensión del lector. Este enfoque estructural permite explorar de manera efectiva cómo los modelos de Machine Learning pueden predecir qué empresas tendrán mayor rentabilidad en el futuro, cumpliendo así con el objetivo principal del estudio.

El primer capítulo, correspondiente a la introducción, establece los fundamentos del trabajo. En él se delimita el alcance del análisis, se exponen los objetivos principales y secundarios, se justifica la relevancia del tema abordado, destacando el impacto potencial de poder anticipar qué empresas tendrán un mejor desempeño financiero. Asimismo, se ofrece una visión general de la metodología empleada, subrayando el enfoque cuantitativo adoptado y las herramientas de análisis utilizadas. Finalmente, se presenta una descripción general de la estructura del TFG, anticipando el contenido de los capítulos siguientes.

El segundo capítulo está dedicado al marco teórico y la revisión de la literatura. Esta sección profundiza en conceptos clave relacionados con la rentabilidad empresarial, los factores que influyen en su evolución, y la aplicación de técnicas de Machine Learning en contextos financieros. Se revisan estudios previos relevantes, se analizan los principales modelos de clasificación utilizados en la literatura y se identifican tanto sus aplicaciones prácticas como sus limitaciones. Esta revisión crítica permite establecer una base teórica sólida sobre la cual se sustenta el análisis empírico posterior, y al mismo tiempo, señala líneas de investigación emergentes en el ámbito del análisis predictivo financiero.

El tercer capítulo corresponde al análisis empírico, núcleo central de esta investigación. En primer lugar, se describe el proceso de obtención y preparación de los datos,

incluyendo la fase de extracción, transformación y carga (ETL). A continuación, se expone detalladamente la metodología empleada para el desarrollo y evaluación de los modelos de Machine Learning, destacando las métricas de rendimiento utilizadas y el enfoque comparativo adoptado. Finalmente, se presentan los resultados obtenidos y se analizan las diferencias en el desempeño de los modelos, identificando cuál ofrece la mayor capacidad predictiva y qué variables tienen un mayor peso en la determinación de la rentabilidad futura.

El cuarto y último capítulo recoge las conclusiones del estudio, ofreciendo una síntesis de los hallazgos más relevantes y sus implicaciones prácticas para inversores y empresas del sector financiero. Asimismo, se discuten las principales limitaciones del trabajo y se proponen futuras líneas de investigación que podrían ampliar y enriquecer los resultados alcanzados. El documento se completa con los anexos, que incluyen el código desarrollado, y la bibliografía utilizada, proporcionando al lector todos los elementos necesarios para la réplica, comprensión o ampliación del análisis presentado.

2. REVISION DE LA LITERATURA Y MARCO TEÓRICO

2.1 Conceptualización de la rentabilidad empresarial

La rentabilidad empresarial es un concepto clave en la evaluación del desempeño de una organización, especialmente en el contexto de los mercados financieros como el S&P 500. En términos generales, la rentabilidad según Lizcano Álvarez & Castelló Taliani (2004) es “la capacidad que tiene una empresa para generar un excedente a partir de los recursos invertidos en su actividad económica, de producción o de intercambio” (p.10). Este concepto va más allá de la manera de obtención de los beneficios contables, ya que incorpora una dimensión de eficiencia: cuánto valor se genera por cada unidad de recurso invertido. Desde esta perspectiva, la rentabilidad permite valorar si una empresa está gestionando de manera eficaz sus activos y capital para producir resultados económicos sostenibles (Economipedia, 2024).

Es fundamental establecer una distinción conceptual entre ganancia y rentabilidad en el análisis del desempeño financiero empresarial. La ganancia constituye un valor absoluto que representa el beneficio neto obtenido en un periodo determinado, mientras que la

rentabilidad se define como un valor relativo que pone en relación dicha ganancia con los recursos utilizados para generarla, tales como los activos totales, la inversión inicial o el patrimonio. Según CEUPE (2022), la utilidad se refiere a las ganancias obtenidas directamente de un producto después de descontar todos los gastos, mientras que la rentabilidad evalúa si un negocio es viable a largo plazo considerando las utilidades y el costo total de la inversión. En este sentido, dos empresas podrían presentar un beneficio neto equivalente; sin embargo, aquella que lo alcanza utilizando un menor volumen de recursos presentará una rentabilidad superior. Esta distinción resulta esencial en el análisis financiero, ya que permite comparar la eficiencia relativa de organizaciones con tamaños, estructuras o niveles de inversión diferentes.

La rentabilidad puede analizarse desde diferentes enfoques, principalmente el económico, el financiero y, más recientemente, el social. La rentabilidad económica evalúa el rendimiento en función de todos los recursos económicos invertidos, sin importar su origen (propios o ajenos), y se asocia habitualmente con indicadores como el Return on Assets (ROA), que mide el beneficio obtenido por cada unidad de activo total. Por otro lado, la rentabilidad financiera se centra en el rendimiento del capital propio invertido por los accionistas, empleando como principal indicador el Return on Equity (ROE), que relaciona el beneficio neto con el patrimonio neto de la empresa (Universidad Europea de Madrid, s.f.).

A estas dimensiones clásicas se suma la rentabilidad social, que incorpora el valor creado para la sociedad por las actividades empresariales. Este enfoque reconoce que las empresas no solo deben maximizar el retorno financiero, sino también deben generar impactos positivos en su entorno social y ambiental, siguiendo principios como los de la triple cuenta de resultados. Según Ramírez Orellana (2006), una organización solo puede sobrevivir a medio-largo plazo si resulta económicamente viable, medioambientalmente sostenible y socialmente responsable.

En este trabajo se adopta un enfoque centrado en la rentabilidad financiera, ya que permite evaluar de forma directa la capacidad de una empresa para generar valor para sus accionistas a partir del capital propio invertido. Este criterio es especialmente relevante en el contexto de análisis de inversiones y estrategias de private equity, donde la rentabilidad financiera facilita comparaciones homogéneas entre empresas del índice

S&P 500 y contribuye a identificar aquellas con mayor potencial de generar retornos sostenibles. No obstante, en el desarrollo empírico del trabajo también se incorporan otras variables financieras clave —como la rentabilidad sobre activos (ROA), márgenes de beneficio o ratios de endeudamiento— con el fin de construir modelos más robustos. Esta decisión se apoya en el estudio de Nagy, Valaskova, Kovalova y Macura (2024), que demuestra cómo la rentabilidad del S&P 500 está condicionada por múltiples factores macroeconómicos y financieros, reforzando la necesidad de integrar distintos indicadores en el análisis para captar de forma más precisa el comportamiento empresarial y su capacidad de generación de valor en contextos económicos dinámicos.

En el contexto actual, las técnicas de Machine Learning (ML) y Deep Learning (DL) están transformando el análisis financiero, particularmente en la predicción de la rentabilidad futura de las empresas. Estas metodologías permiten modelar relaciones no lineales complejas entre múltiples variables financieras, lo que representa una ventaja frente a los enfoques tradicionales basados en supuestos lineales. Anand, Brunner, Ikegwu y Sougiannis (2019) demuestran que modelos como los árboles aleatorios (Random Forests) y el Gradient Boosting superan en capacidad predictiva a los modelos de regresión lineal tradicional, al capturar interacciones complejas entre ratios financieros y cambios en la rentabilidad a lo largo del tiempo. Del mismo modo, recientes autores como Artene y Domil (2025) emplean redes neuronales artificiales para predecir resultados financieros, evidenciando que los sistemas basados en DL logran una mayor precisión al analizar grandes volúmenes de datos contables y económicos. En conjunto, estos estudios evidencian que las técnicas de ML y DL no solo mejoran la precisión de las predicciones, sino que también generan información estratégica clave para la toma de decisiones en contextos empresariales e inversores cada vez más dinámicos y competitivos.

2.2 Factores determinantes de la rentabilidad futura

La rentabilidad futura de una empresa ha sido ya ampliamente estudiada en la literatura financiera contemporánea, ya que su predicción influye directamente a las decisiones de inversión, planificación estratégica y evaluación del valor de las empresas. Esta se ha analizado desde enfoques cualitativos, como la teoría de los recursos y capacidades, la estrategia competitiva o los ciclos económicos, y también desde metodologías cuantitativas, basadas en datos contables, macroeconómicos y técnicas de inteligencia artificial.

Desde una perspectiva interna, la rentabilidad futura depende en gran medida de factores como la estructura operativa, el gobierno corporativo, la eficiencia productiva y la capacidad de innovación. Según Aguinis y Glavas (2020), una gestión empresarial que incorpore criterios de sostenibilidad y responsabilidad social contribuye al fortalecimiento de la reputación corporativa y a relaciones sólidas con los stakeholders, lo cual se traduce en un mejor desempeño financiero en el medio y largo plazo.

Por otro lado, la teoría de los recursos y capacidades, propuesta por Barney (1991), sostiene que las ventajas competitivas sostenibles, como el conocimiento especializado, la eficiencia operativa y las rutinas organizativas, son fundamentales para explicar las diferencias significativas en la rentabilidad futura de las empresas. Esta perspectiva ha sido reafirmada en investigaciones recientes que destacan la influencia positiva de los recursos intangibles en el desempeño financiero. Por ejemplo, un estudio centrado en la industria hotelera española encontró que el stock de activos intangibles, como la reputación de marca y el capital humano, mantiene una relación significativa con la rentabilidad empresarial, medida a través del retorno sobre activos (ROA). Este hallazgo subraya la relevancia de los recursos intangibles como determinantes clave de la rentabilidad sostenible en sectores intensivos en conocimiento.

En relación, a los factores externos, el entorno en el que operan también influye significativamente en su rentabilidad futura. Factores macroeconómicos como el crecimiento del PIB, las tasas de interés, la inflación o el ciclo económico afectan el desempeño financiero, especialmente en empresas cotizadas de mercados desarrollados como el S&P 500 (Nagy et al., 2024). Las empresas tienden a tener mejores márgenes y crecimiento en periodos expansivos, mientras que enfrentan mayores desafíos durante las recesiones.

Además de los factores macroeconómicos, el comportamiento sectorial también desempeña un papel crucial en la rentabilidad de las empresas. Algunas industrias presentan una rentabilidad estructuralmente mayor debido a barreras de entrada, economías de escala o niveles de competencia reducidos. Por ejemplo, los sectores tecnológicos y farmacéuticos del S&P 500 han mostrado históricamente mejores márgenes que el industrial o el de consumo básico. Según un análisis de T. Rowe Price (2024), las acciones de crecimiento, especialmente en tecnología, han mantenido

márgenes de beneficio elevados durante años, en parte debido a las "trincheras tecnológicas" que protegen a las principales empresas de la competencia. Esto sugiere que las barreras tecnológicas y las ventajas competitivas sostenibles contribuyen significativamente a la rentabilidad superior en estos sectores.

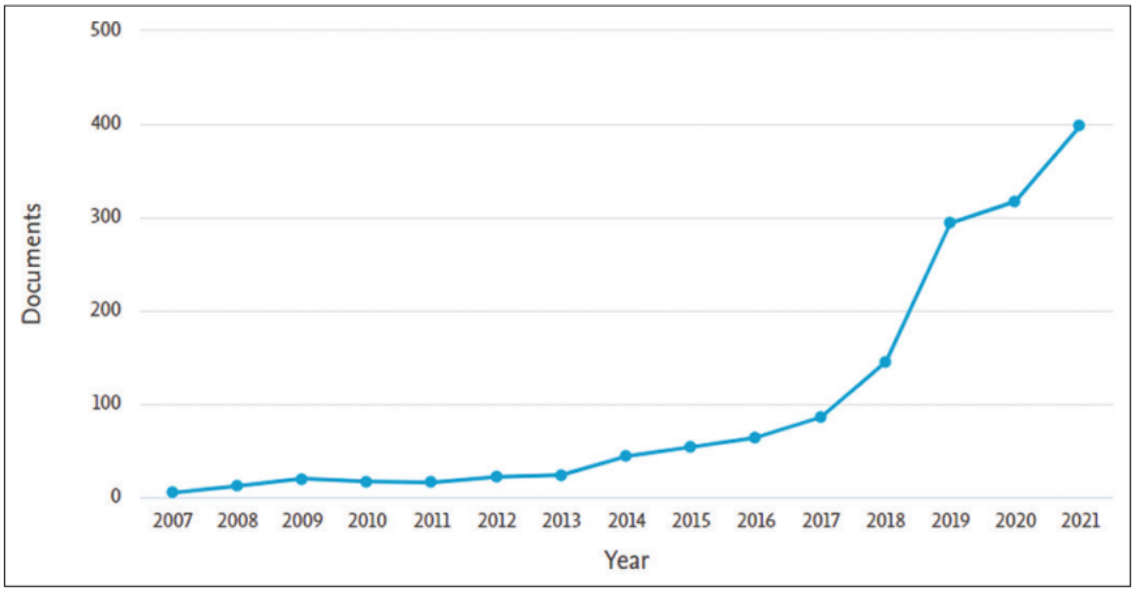
En definitiva, la rentabilidad futura de una empresa está determinada por una combinación compleja de factores internos y externos. Desde las capacidades organizativas y los recursos intangibles hasta las condiciones macroeconómicas y sectoriales, cada elemento influye de forma directa en el desempeño financiero sostenible. Tanto la literatura teórica como los estudios empíricos recientes coinciden en que aquellas empresas que gestionan eficazmente sus recursos innovan de manera constante y operan en sectores estratégicos con barreras competitivas relevantes tienden a obtener rentabilidades superiores a largo plazo. Esta visión multidimensional de la rentabilidad no solo enriquece el análisis financiero, sino que también aporta un marco más sólido para la toma de decisiones en contextos de inversión, como el que representa el índice S&P 500.

2.3 Estudios previos de predicción financiera basados en ML

En los últimos años, el uso de técnicas de Machine Learning (ML) ha transformado el análisis financiero, especialmente en áreas como la predicción de rentabilidad, la evaluación de riesgos crediticios, la detección de quiebras empresariales y la anticipación de movimientos en los mercados bursátiles. Estas herramientas permiten procesar grandes volúmenes de datos históricos y detectar patrones complejos y no lineales que los modelos estadísticos tradicionales no logran captar con la misma precisión.

La revisión de bases de datos científicas, como Scopus, evidencia un crecimiento exponencial en la aplicación del aprendizaje automático (Machine Learning, ML) en los ámbitos empresarial y financiero. El gráfico 1 muestra la evolución del número de publicaciones relacionadas con el uso del ML en estos sectores financieros entre 2007 y 2021, donde se observa un incremento de apenas 5 documentos en 2007 a más de 400 en 2021. Este patrón refleja un interés académico creciente y sostenido por parte de la comunidad investigadora en el potencial de estas tecnologías para transformar la toma de decisiones financieras (Ajibade et al., 2024, traducción propia).

Gráfico 1. Número de publicaciones sobre ML en negocios y finanzas entre 2007 y 2021



Fuente: Adaptado de Ajibade et al. (2024)

Este incremento en la literatura especializada ha estado acompañado por un enfoque particular en técnicas de aprendizaje supervisado, dada su eficacia en tareas predictivas dentro del sector financiero. El aprendizaje supervisado se basa en el uso de datos etiquetados para entrenar modelos capaces de realizar predicciones sobre nuevos escenarios. Su aplicación abarca desde la clasificación del riesgo crediticio hasta la predicción de la rentabilidad de activos financieros y la detección de fraudes.

Según Masini, Medeiros y Mendes (2020), los modelos supervisados han demostrado una elevada eficacia en la predicción de variables financieras clave, al reducir errores y mejorar la capacidad de generalización en entornos dinámicos. Estos modelos integran información como ratios financieros, series históricas de precios y métricas operativas para estimar la probabilidad de éxito financiero. Diversos estudios han profundizado en la aplicación de estas técnicas. Por ejemplo, Fuzail et al. (2023) analizaron el uso de Redes Neuronales Artificiales (ANN), Máquinas de Vectores de Soporte (SVM) y Redes Neuronales Recurrentes (RNN) para la predicción de precios bursátiles, evidenciando mejoras significativas en la precisión de las predicciones.

No obstante, a pesar de los avances, persisten desafíos importantes. Muchos modelos presentan limitaciones frente a datos ruidosos, alta volatilidad o cambios estructurales en

los mercados financieros. Además, gran parte de la investigación se ha centrado en contextos o geografías específicas, lo que resalta la necesidad de estudios comparativos entre mercados emergentes y desarrollados.

Se espera que la evolución futura de estas técnicas permita desarrollar modelos más robustos**,** interpretables y adaptados al dinamismo del entorno financiero actual. En este contexto, el presente estudio busca contribuir a esta línea de investigación, aplicando modelos de aprendizaje supervisado para la predicción de la rentabilidad futura en empresas del índice S&P 500, ofreciendo una perspectiva práctica y comparativa que permita identificar oportunidades de inversión con mayor precisión.

2.4 Aplicaciones prácticas de clasificación en predicciones financieras

La actual revolución digital ha consolidado las técnicas de Machine Learning (ML), y en particular los modelos de clasificación**,** como herramientas esenciales en el análisis financiero. En un entorno caracterizado por la volatilidad, la complejidad de los mercados y el crecimiento exponencial de datos, estos modelos han demostrado un notable potencial para generar predicciones precisas y respaldar decisiones estratégicas. A diferencia de los métodos estadísticos tradicionales —que requieren supuestos estrictos y suelen limitarse a relaciones lineales— los algoritmos de clasificación permiten identificar patrones ocultos, modelar interacciones complejas y adaptarse dinámicamente a cambios en el mercado.

Diversos estudios han evidenciado la superioridad de estos enfoques frente a las metodologías convencionales. Por ejemplo, Sun, Li y Huang (2014) aplicaron algoritmos como árboles de decisión**,** redes neuronales y máquinas de vectores de soporte (SVM) para la predicción de bancarrotas empresariales, obteniendo resultados significativamente mejores que los que generaban los modelos logit tradicionales. En la misma línea, Lessmann et al. (2015) realizaron una revisión comparativa de 41 técnicas de clasificación aplicadas al riesgo crediticio, concluyendo que los modelos de ML superan de manera sistemática a los enfoques econométricos en términos de precisión.

Además, informes sectoriales refuerzan estas evidencias prácticas. Según Infomineo (2024), algoritmos como los árboles de decisión y las redes neuronales destacan por su capacidad para identificar relaciones no lineales en datos financieros, permitiendo una mejor comprensión y predicción de fenómenos económicos complejos. Por su parte, el

Grupo Stefanini (2023) señala que los modelos de ML no solo mejoran la precisión de las proyecciones financieras, sino que también automatizan procesos analíticos avanzados, incrementando la eficiencia operativa y facilitando decisiones en tiempo real, especialmente en sectores como la banca y las Fintech. Finalmente, Majka (2024) destaca la capacidad de estos modelos por su capacidad de adaptación continua, ajustando sus parámetros a medida que se incorporan nuevos datos, lo que les permite mantener su rendimiento predictivo en contextos financieros cambiantes.

En resumen, las aplicaciones prácticas de los modelos de clasificación en el ámbito financiero han demostrado ser una alternativa eficaz y adaptable frente a los métodos tradicionales. Esta capacidad de manejar grandes volúmenes de información, identificar patrones complejos y ajustarse a dinámicas de mercado convierte a estas técnicas en herramientas clave para la predicción de rentabilidad, tal como se propone en el presente estudio aplicado a las empresas del índice S&P 500.

2.5 Limitaciones y desafíos en la predicción financiera mediante técnicas de ML

A pesar de las numerosas ventajas que ofrecen las técnicas de Machine Learning (ML) en el ámbito financiero, estas presentan también importantes limitaciones y desafíos que deben ser considerados, especialmente cuando se aplican en contextos reales caracterizados por una alta exigencia y responsabilidad. Estos obstáculos no solo afectan la fiabilidad y precisión de los modelos, sino también su utilidad práctica en la toma de decisiones estratégicas.

Uno de los principales retos asociados al uso de técnicas avanzadas de ML es la falta de interpretabilidad. Según Guidotti et al. (2018), modelos como las redes neuronales profundas o los algoritmos ensemble (como Random Forest o XGBoost) operan frecuentemente como auténticas "cajas negras", en las que el proceso que conduce a una determinada predicción resulta difícil de comprender para los usuarios finales. Esta opacidad representa un desafío crítico en el sector financiero, donde la transparencia y la trazabilidad son requisitos indispensables para garantizar decisiones fundamentadas y cumplir con las normativas regulatorias.

Otro desafío relevante es la fuerte dependencia de los modelos respecto a la calidad y representatividad de los datos utilizados en su entrenamiento. Datos incompletos, desbalanceados o incorrectamente etiquetados pueden derivar en resultados engañosos o

poco generalizables. Además, tal como advierten Barocas, Hardt y Narayanan (2019), existe el riesgo de que estos modelos reproduzcan y amplifiquen sesgos históricos presentes en los datos, incluyendo sesgos por género, edad o ubicación geográfica. En contextos financieros, esta problemática puede traducirse en decisiones éticamente cuestionables, particularmente en áreas sensibles como la concesión de préstamos, la evaluación de riesgos en seguros o la inversión automatizada.

Por tanto, aunque las técnicas de ML suponen un avance significativo respecto a los métodos estadísticos tradicionales, es fundamental complementarlas con buenas prácticas en la gestión de datos, una cuidadosa selección de variables, y procesos rigurosos de validación ética. Solo así es posible garantizar que el uso de estos modelos no sea únicamente eficaz desde el punto de vista técnico, sino también responsable y alineado con principios de equidad, transparencia y sostenibilidad en la toma de decisiones financieras.

3. ANÁLISIS EMPÍRICO

Para antes del desarrollo de estos modelos, se realizó inicialmente un análisis exploratorio de los datos con el objetivo de comprender la estructura, calidad y principales características del conjunto de datos financieros históricos de las empresas que conforman el índice S&P 500. Esta etapa fue esencial para identificar patrones relevantes, gestionar valores atípicos, tratar datos faltantes y preparar adecuadamente el data set para su posterior uso en los modelos predictivos. El procedimiento se inició con una valoración inicial de las variables existentes, que comprendían indicadores financieros esenciales como ingresos, márgenes de beneficio, ratios de endeudamiento, además de datos históricos de cotización de acciones, entre otros. En esta etapa se comprobó la existencia de valores nulos, inconsistencias y posibles outliers que pudieran comprometer la calidad del análisis.

Posteriormente, se llevó a cabo un análisis más detallado mediante la generación de visualizaciones descriptivas, tales como histogramas, boxplots (diagramas de caja) y gráficos de dispersión. Estas herramientas facilitaron la identificación de la distribución de las variables, la detección de relaciones significativas entre ellas y la localización de valores atípicos que requerían un tratamiento específico antes de proceder con la fase de modelado.

3.1 DATOS

Para el desarrollo del análisis empírico y la construcción de los modelos, se emplearon dos data sets principales: S&P 500 Companies Description, con información descriptiva y financiera de las empresas del índice correspondiente al año 2023, y All Stocks 5 Yrs, que recopila datos históricos de precios de acciones, volumen de negociación y otros indicadores clave correspondientes al período 2013–2018.

La elección de estos conjuntos de datos responde a la necesidad de combinar información financiera actual con datos históricos de comportamiento bursátil, con el objetivo de construir modelos de predicción de rentabilidad futura. El dataset S&P 500 Companies Description 2023 proporciona una fotografía actualizada de la situación financiera de las empresas, mientras que All Stocks 5 Yrs (2013–2018) permite observar patrones de evolución en los precios y volúmenes de negociación a lo largo del tiempo. Esta combinación facilita la identificación de relaciones entre características estructurales de las compañías y su rendimiento bursátil posterior, lo que constituye la base del análisis predictivo desarrollado.

Previamente al análisis exploratorio, se realizó una estandarización de las unidades en el dataset S&P 500 Companies Description, ya que varios valores —como la capitalización de mercado e ingresos— se encontraban expresados en distintas escalas, utilizando millones (M) y billones (B). Para asegurar la coherencia y facilitar comparaciones directas, todos los valores fueron convertidos al sistema anglosajón, donde un billion equivale a mil millones (10^9).

3.2 ANÁLISIS EXPLORATORIO VARIABLES

Una vez estandarizadas las métricas en una escala homogénea, se procedió a realizar un análisis exploratorio de las variables incluidas en el conjunto de datos. Este proceso se estructuró en tres fases principales: cálculo de estadísticas descriptivas, análisis de correlación y visualización de relaciones entre variables.

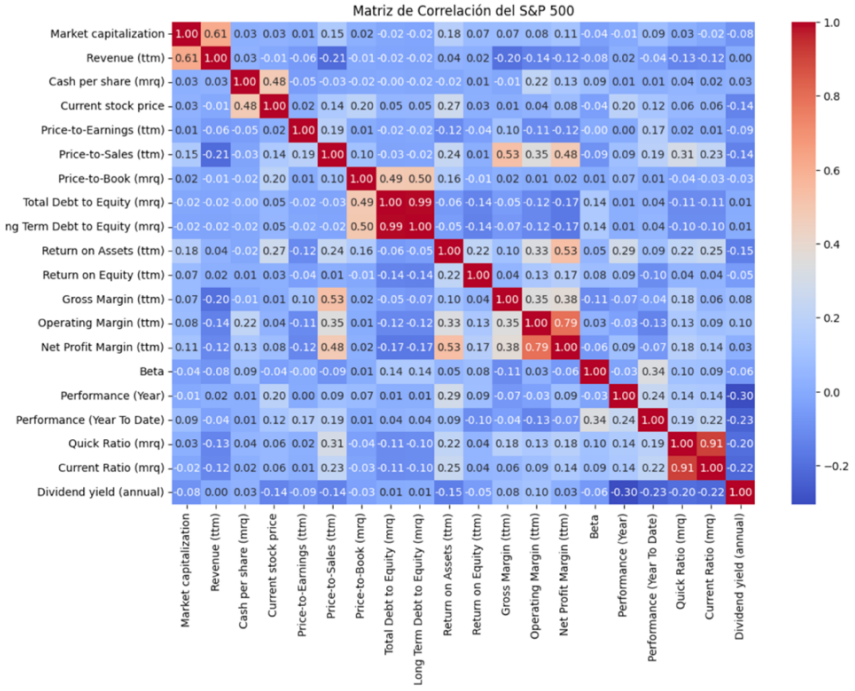
En una primera etapa, se calcularon las estadísticas descriptivas de las principales variables numéricas. Entre los estadísticos obtenidos se incluyen la media, mediana, desviación estándar, así como los valores mínimo y máximo de indicadores clave como la capitalización bursátil y los ingresos. Este análisis permitió identificar la existencia de

valores atípicos, evaluar la dispersión de los datos y establecer rangos de referencia para un tratamiento posterior adecuado de las observaciones extremas.

En la segunda fase, se construyó una matriz de correlación con el fin de analizar la fuerza y dirección de las relaciones lineales entre las variables numéricas. Este enfoque resultó especialmente útil para detectar asociaciones significativas entre métricas financieras y para anticipar posibles problemas de multicolinealidad en las etapas de modelización. La matriz fue visualizada mediante un mapa de calor (heatmap), una herramienta que permite representar gráficamente la intensidad de las correlaciones.

En esta representación gráfica, los tonos rojo oscuro indican una fuerte correlación positiva (valores próximos a +1), mientras que los tonos azul oscuro reflejan correlaciones negativas significativas (valores cercanos a -1). Los colores más claros señalan relaciones débiles o inexistentes (valores próximos a 0). A continuación, se expone el heatmap de la matriz de correlación, cuya interpretación resulta fundamental para la selección de variables y la prevención de problemas de multicolinealidad en los modelos posteriores.

Gráfico 2. Matriz de correlación de las variables financieras numéricas del data set S&P 500 Companies Description



Fuente: Elaboración propia a partir del DataSet

El Gráfico 2 evidencia algunas correlaciones significativas entre las variables financieras analizadas. Destaca la fuerte relación entre el Gross Margin y el Operating Margin (0.79), reflejando cómo ambas métricas capturan aspectos similares de la eficiencia operativa. Por este motivo, se decidió conservar únicamente el Operating Margin, al ofrecer una visión más completa del rendimiento empresarial.

De igual manera, se observó una correlación casi perfecta (0.99) entre Total Debt to Equity y Long Term Debt to Equity. Dado que la primera variable incorpora tanto la deuda a corto como a largo plazo, se optó por eliminar la segunda, priorizando una perspectiva más integral de la estructura financiera. Una situación comparable se presentó entre el Quick Ratio y el Current Ratio (correlación de 0.91), donde se seleccionó el Quick Ratio por su enfoque más conservador al excluir inventarios, siendo más adecuado para contextos de análisis de liquidez y riesgo.

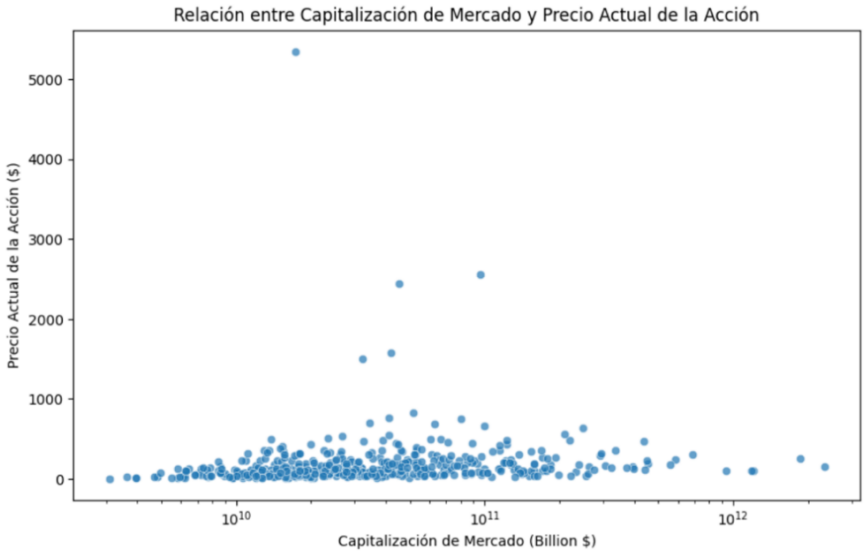
Por otro lado, se identificaron correlaciones moderadas, como la existente entre Price-to-Sales y Gross Margin (0.53), o entre Return on Equity (ROE) y Net Profit Margin (0.45). Estas relaciones, al no ser excesivamente elevadas, permiten mantener ambas variables en el modelo, dado que aportan información complementaria sin generar redundancias significativas.

En contraste, el Dividend Yield (annual) mostró correlaciones bajas o negativas con la mayoría de las variables (por ejemplo, -0.30 con el rendimiento anual), lo que indica que este indicador refleja una dimensión diferente del comportamiento financiero, asociada habitualmente a empresas consolidadas que priorizan la distribución de dividendos frente al crecimiento. De forma similar, la variable Beta**,** que mide la volatilidad relativa respecto al mercado, presentó correlaciones inferiores a 0.15, aportando así una perspectiva adicional en el análisis del riesgo.

Como resultado de estos hallazgos, se procedió a depurar las variables altamente correlacionadas, eliminando aquellas redundantes —como Long Term Debt to Equity, Current Ratio y Operating Margin— y priorizando las métricas más representativas. Esta optimización busca reducir la colinealidad, mejorar la interpretabilidad del modelo y maximizar su rendimiento predictivo. Siguiendo con la parte del análisis exploratorio de relaciones entre variables clave, lo siguiente que se consideró examinar gráficamente fue

la asociación entre la capitalización de mercado y el precio actual de las acciones de las empresas que integran el índice S&P 500.

Gráfico 3. Relación entre Capitalización de Mercado y Precio Actual de las Acciones en el S&P 500.



Fuente: Elaboración propia a partir del DataSet

El gráfico 3 se fundamenta en el interés por comprender hasta qué punto el valor bursátil total de una empresa se relaciona con el precio unitario de sus acciones, dos variables fundamentales en la evaluación financiera corporativa. Para ello, se utilizó un gráfico de dispersión (scatter plot) en el que se representa cada empresa como un punto, situando la capitalización de mercado en el eje X (expresada en escala logarítmica para mejorar la visibilidad de empresas con menor tamaño relativo) y el precio actual de la acción en el eje Y.

El uso de la escala logarítmica en el eje horizontal fue crucial para evitar que las empresas con menor capitalización quedaran visualmente agrupadas y su análisis se viera distorsionado por la presencia de valores extremos. Esta transformación permitió una representación más equitativa del conjunto de empresas, facilitando la identificación de patrones y outliers.

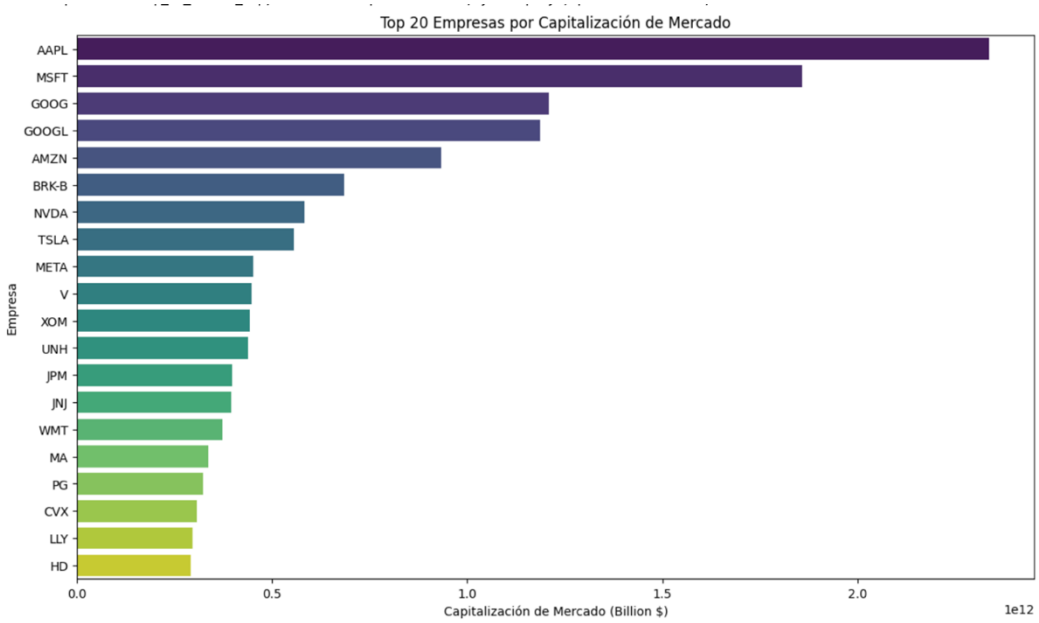
La visualización revela que la mayoría de las empresas se concentran en rangos bajos o medios tanto de capitalización como de precio por acción, lo que indica que, si bien

existen corporaciones con valores excepcionalmente altos, estas no constituyen la mayoría del índice. También se aprecian outliers con precios de acción o capitalizaciones particularmente elevadas, los cuales podrían ejercer un peso desproporcionado en la construcción de modelos predictivos. Por ello, se identificaron como observaciones críticas que podrían requerir tratamiento específico en fases posteriores del análisis.

A pesar de que el gráfico sugiere una ligera correlación positiva entre ambas variables, no se observa una relación lineal clara. Este hallazgo refuerza la conveniencia de emplear técnicas de aprendizaje automático (machine learning) capaces de capturar relaciones complejas y no lineales entre las variables, superando así las limitaciones de enfoques econométricos tradicionales.

Siguiendo con el análisis y con el objetivo de identificar a los actores más influyentes dentro del índice S&P 500, se elaboró un gráfico de barras horizontales que muestra las 20 empresas con mayor capitalización de mercado, expresada en billones de dólares.

Gráfico 4. Las 20 Empresas Más Valiosas del S&P 500 por Capitalización de Mercado



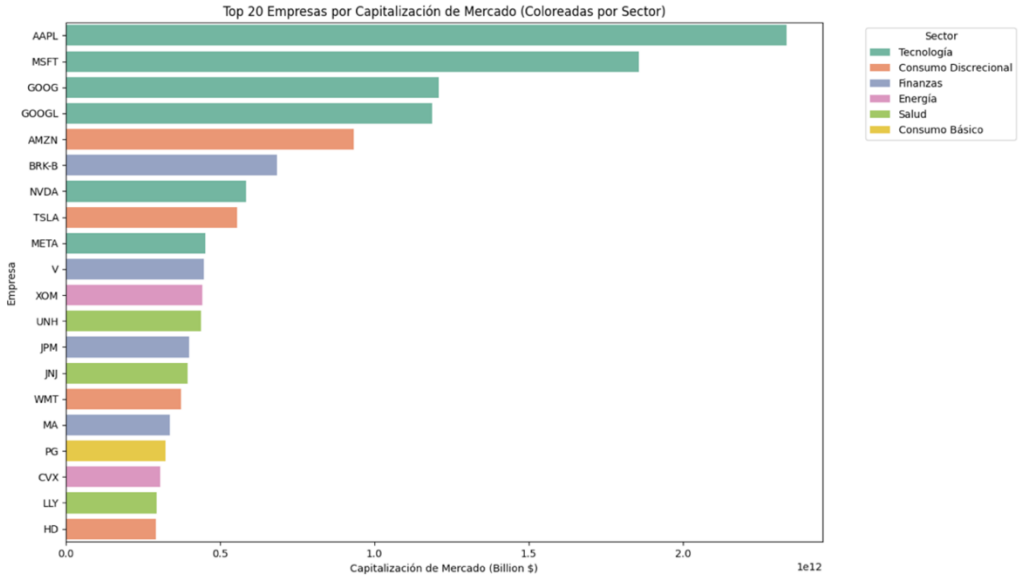
Fuente: Elaboración propia a partir del DataSet

Esta visualización se basa en datos correspondientes al periodo comprendido entre el 1 de enero de 2023 y finales de diciembre de 2023, lo que proporciona una fotografía representativa del cierre anual de ese ejercicio fiscal.

El gráfico revela que Apple (AAPL) y Microsoft (MSFT) lideraban el ranking de capitalización en ese momento, seguidas por Alphabet Inc., representada a través de sus dos tipos de acciones: GOOG y GOOGL. Estas empresas reflejan el peso dominante del sector tecnológico durante ese periodo. Aunque sus posiciones siguen siendo relevantes en 2025, es importante señalar que los datos no reflejan los cambios más recientes en las valoraciones de mercado. No obstante, la información resulta válida para el análisis histórico y la construcción de modelos, ya que recoge tendencias estructurales clave.

Además, las empresas fueron clasificadas por sectores, lo que permitió visualizar el dominio del sector tecnológico, seguido de otros como consumo discrecional**,** finanzas y salud. Esta clasificación refuerza la hipótesis de que el sector al que pertenece una empresa podría actuar como una variable explicativa relevante en la predicción de su rentabilidad futura. Por ello, tras identificar las empresas con mayor capitalización de mercado dentro del índice S&P 500, se consideró incorporar una clasificación sectorial para enriquecer el análisis.

Gráfico 5. Top 20 Empresas del S&P 500 por Capitalización de Mercado, Clasificadas por Sector



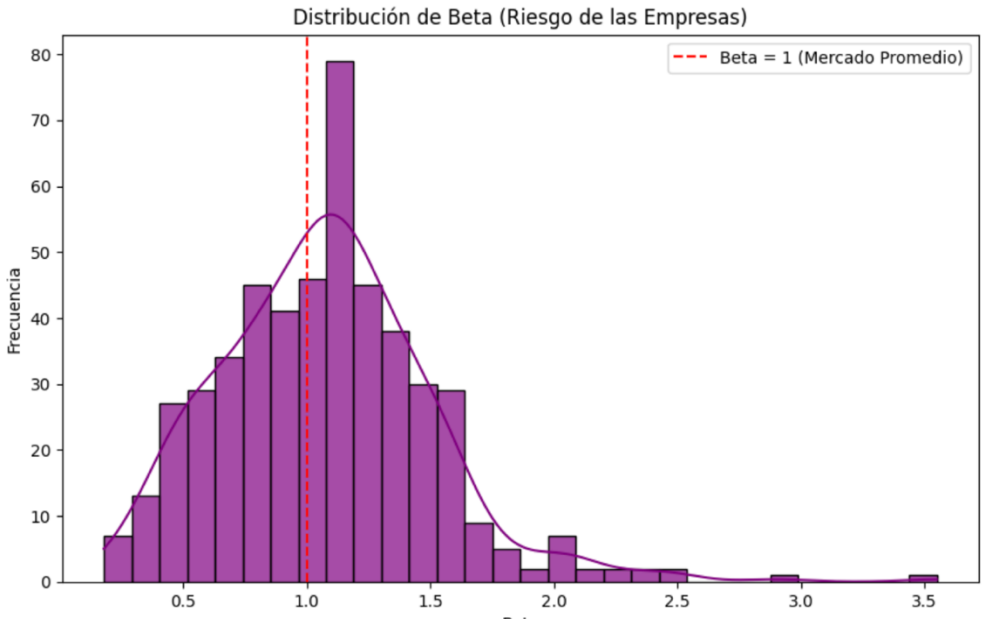
Fuente: Elaboración propia a partir del DataSet

El gráfico 5 que se presenta muestra las 20 compañías más valiosas del índice, diferenciadas visualmente por colores según el sector al que pertenecen. Esta visualización permite observar no solo el peso individual de cada empresa, sino también cómo se distribuyen los valores de mercado entre las distintas industrias.

El uso de codificación por colores facilitó significativamente las comparaciones tanto dentro de cada sector como entre sectores, evidenciando la concentración del valor bursátil en determinadas áreas, especialmente en el sector tecnológico. Además, esta categorización por sector es de gran relevancia en el contexto de los modelos de predicción, ya que el sector económico al que pertenece una empresa puede influir significativamente en su comportamiento financiero y, por ende, en su rentabilidad futura. Incorporar esta variable como predictor permitirá capturar patrones sectoriales que no serían detectables mediante análisis puramente numéricos.

Tras identificar la distribución sectorial de las principales empresas del índice, se consideró relevante examinar la distribución del riesgo sistemático en el conjunto de compañías que integran el S&P 500.

Gráfico 6. Distribución del Coeficiente Beta en el S&P 500: Análisis del Riesgo de las Empresas



Fuente: Elaboración propia a partir del DataSet

En el gráfico 6 se analizó el coeficiente Beta, una métrica clave en finanzas que mide la sensibilidad de los rendimientos de una acción frente a las variaciones del mercado. El objetivo de este análisis fue identificar el comportamiento relativo al riesgo de las empresas del índice y evaluar la posible influencia de este indicador sobre la rentabilidad futura, en el marco del desarrollo de modelos predictivos.

El gráfico presenta un histograma de frecuencia que representa el número de empresas según sus valores de Beta, complementado con una curva de densidad que permite visualizar la forma de la distribución. En el eje X se encuentran los valores de Beta, mientras que el eje Y indica la frecuencia absoluta, es decir, la cantidad de empresas que presentan un coeficiente dentro de cada intervalo.

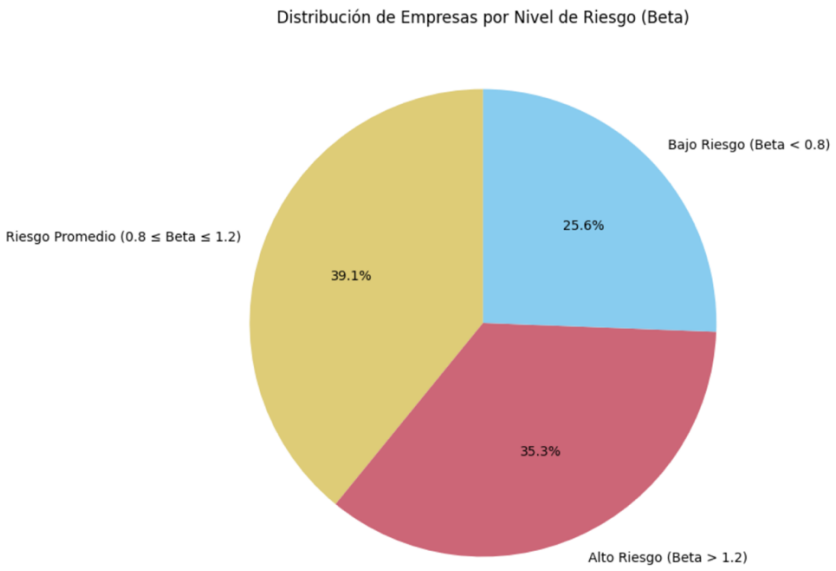
La línea discontinua roja marca el valor de Beta = 1, que representa el riesgo promedio del mercado. Se observa que la mayoría de las empresas presentan valores de Beta cercanos a 1, lo que sugiere que su comportamiento sigue de forma relativamente estable las fluctuaciones del mercado general. Sin embargo, también se identifican empresas con Betas significativamente superiores a 2, que implican una alta volatilidad, y otras con Betas por debajo de 0.5, que reflejan un perfil más conservador o defensivo frente a los movimientos del mercado.

Esta representación permitió comprender la estructura de riesgo del índice de manera global, así como detectar asimetrías en la distribución, que tiende a estar sesgada hacia la derecha (mayor presencia de valores altos). Este hallazgo sugiere la presencia de empresas con un perfil especulativo o muy expuesto a la volatilidad.

Incorporar el coeficiente Beta como variable explicativa en los modelos de machine learning puede resultar altamente relevante, ya que permite capturar relaciones entre el nivel de riesgo y el comportamiento futuro de la rentabilidad. Además, dándole una perspectiva más práctica este análisis resulta de gran utilidad para inversores institucionales y empresas de private equity, ya que les permite identificar perfiles de compañías que, pese a presentar un coeficiente Beta bajo —y por tanto menor riesgo sistemático—, podrían mostrar un alto potencial de rentabilidad.

De forma análoga, también facilita la localización de empresas altamente volátiles que podrían ser atractivas en estrategias de inversión orientadas a crecimiento agresivo o especulativo. Con el objetivo de profundizar en el análisis del riesgo sistemático, se llevó a cabo el siguiente gráfico, que nos muestra la distribución de empresas del índice por niveles de riesgo.

Gráfico 7. Distribución de Empresas del S&P 500 por Niveles de Riesgo (Beta)



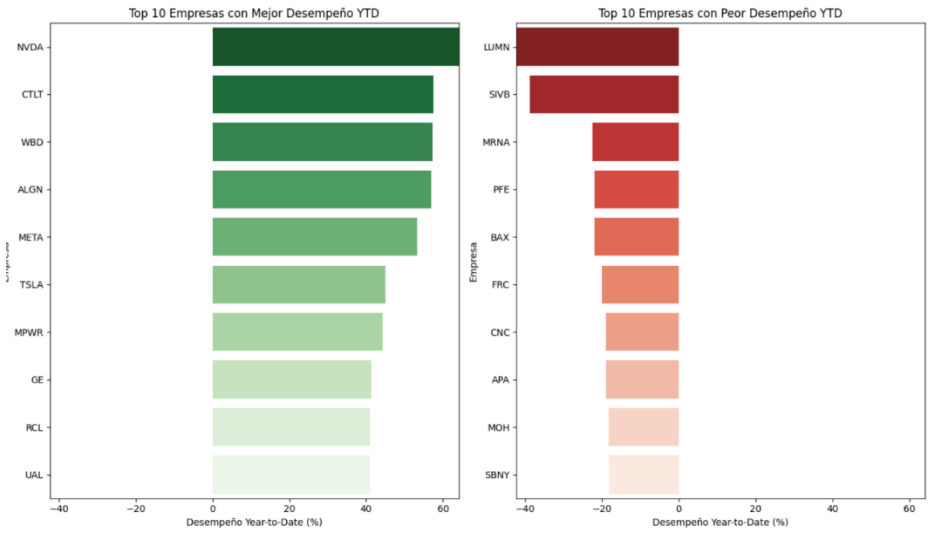
Fuente: Elaboración propia a partir del DataSet

Este análisis reveló que el 39.1% de las empresas presentan un riesgo promedio, con un coeficiente Beta cercano al mercado (Beta = 1), lo que indica un comportamiento alineado con el rendimiento general del índice. Estas empresas representan un perfil de estabilidad y son indicativas de la diversidad dentro del S&P 500. Por otro lado, el 25.6% de las empresas se clasificaron como de bajo riesgo, mostrando menor volatilidad que el mercado promedio. Estas empresas suelen estar asociadas a sectores más defensivos, como salud o consumo básico, y pueden ser más atractivas durante períodos de incertidumbre económica. Finalmente, el 35.3% de las empresas se identificaron como de alto riesgo, destacándose por su mayor volatilidad y sensibilidad a los movimientos del mercado, características típicas de sectores como tecnología o consumo discrecional.

Esta clasificación por niveles de riesgo resultó particularmente útil para complementar los modelos predictivos, ya que el riesgo sistemático puede ser un factor determinante en la rentabilidad futura. Las empresas de alto riesgo, por ejemplo, tienen mayor potencial de crecimiento en mercados alcistas, pero también están más expuestas a pérdidas en mercados bajistas. Por su parte, las empresas de bajo riesgo ofrecen una mayor estabilidad, convirtiéndose en opciones atractivas para inversores más conservadores.

Con el objetivo de complementar el análisis de las métricas financieras y obtener una visión más dinámica del comportamiento reciente de las empresas que componen el índice S&P 500, se elaboró un gráfico comparativo que representa las 10 empresas con mejor desempeño Year-to-Date (YTD) y las 10 con peor desempeño YTD. Este enfoque permite identificar patrones relevantes vinculados al rendimiento reciente del mercado, facilitando la detección de compañías que han experimentado un crecimiento sobresaliente o, por el contrario, han enfrentado dificultades significativas a lo largo del año.

Gráfico 8. Top 10 Empresas del S&P 500 con Mejor y Peor Desempeño Year-to-Date (YTD)



Fuente: Elaboración propia a partir del DataSet

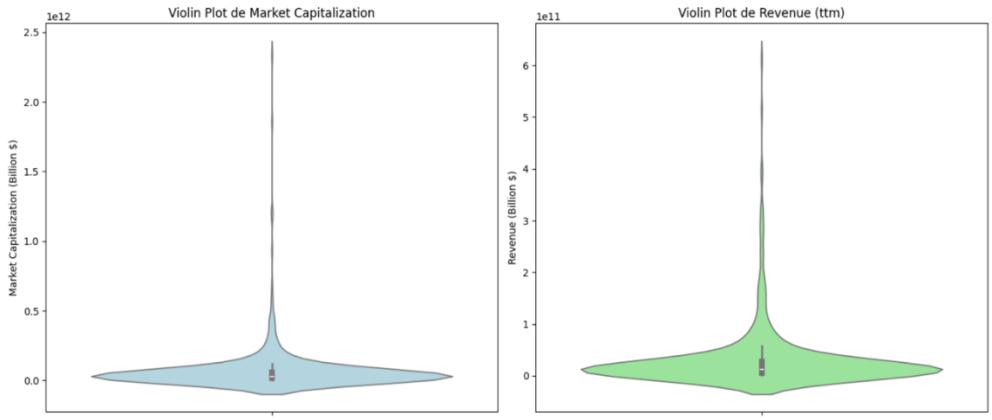
El gráfico 8 revela diferencias significativas entre los dos grupos, destacando cómo el contexto sectorial y los eventos recientes influyen en el rendimiento de las empresas. Por un lado, las empresas con el mejor desempeño, encabezadas por NVIDIA (NVDA), pertenecen mayoritariamente a sectores dinámicos como la tecnología y el consumo

discrecional. Este crecimiento puede estar impulsado por factores como avances tecnológicos, mayor demanda en sus productos o servicios, o un entorno de mercado favorable. Estas empresas han experimentado incrementos significativos, con variaciones positivas superiores al 40%, posicionándose como líderes en sus respectivas industrias.

Destacamos que como los datos de los gráficos comprenden el periodo de 2023, un año marcado por el auge de la inteligencia artificial y el renovado interés por empresas tecnológicas. En particular, NVIDIA se benefició del creciente uso de sus chips en los modelos de inteligencia artificial generativa, lo que impulsó significativamente su cotización. Asimismo, otras compañías como Tesla o Meta mostraron fuertes recuperaciones tras un 2022 más volátil.

En contraste, las empresas con peor desempeño, lideradas por Lumen Technologies (LUMN) y Silicon Valley Bank (SIVB)**,** han enfrentado caídas notables en su rendimiento. Estas disminuciones pueden estar relacionadas con desafíos sectoriales, como cambios regulatorios, fluctuaciones en la demanda o problemas específicos de cada empresa. En el caso de SIVB, su desplome se vincula directamente con la crisis bancaria que afectó a varias entidades financieras regionales en los primeros meses de 2023, lo que generó un fuerte impacto en la confianza del mercado. Y, por último, para finalizar el análisis empírico de las principales métricas financieras del S&P 500, se elaboraron dos gráficos de tipo violin plot que representan la distribución de la capitalización de mercado y los ingresos totales (revenue ttm) de las empresas del índice.

Gráfico 9. Distribución de la Capitalización de Mercado e Ingresos Totales (Revenue) en el S&P 500.



Fuente: Elaboración propia a partir del DataSet

A diferencia de los boxplots tradicionales, estos gráficos permiten visualizar simultáneamente la dispersión, los valores centrales y la densidad de probabilidad, ofreciendo una representación más rica y detallada de las variables.

Ambas distribuciones muestran una alta asimetría positiva, con una gran concentración de empresas en valores bajos o moderados, y una cola superior pronunciada. Esta cola recoge a un pequeño grupo de compañías significativamente más grandes en términos de ingresos y valor bursátil, que destacan como outliers en el conjunto del índice.

Estas empresas atípicas no solo presentan una escala muy superior al resto, sino que también podrían ejercer un efecto desproporcionado en los modelos predictivos, especialmente si no se normalizan adecuadamente las variables. Por tanto, esta visualización refuerza la necesidad de aplicar técnicas de tratamiento de escalas y de considerar la heterogeneidad estructural del índice.

Finalmente, este análisis proporciona una base sólida para la selección de variables relevantes, paso previo fundamental en la construcción de los modelos de machine learning. Esta selección buscará optimizar la precisión de las predicciones sin comprometer la interpretabilidad del modelo ni su capacidad de generalización.

3.3 ANÁLISIS DE LAS VARIABLES SELECCIONADAS:

A partir del análisis de la matriz de correlación de las variables financieras, se procedió a depurar el conjunto de datos con el objetivo de eliminar variables redundantes y priorizar aquellas con mayor relevancia analítica. Como resultado de este proceso, se seleccionaron 17 variables clave, fundamentadas tanto en su frecuencia de uso en la literatura económica y financiera, como en su disponibilidad y calidad dentro del data set original**.**

Estas variables abarcan dimensiones fundamentales de la estructura financiera empresarial, incluyendo indicadores de valoración, rentabilidad, riesgo**,** apalancamiento y liquidez, lo que permite construir una visión integral del desempeño corporativo. La selección de estas métricas busca maximizar el poder explicativo de los modelos predictivos, preservando al mismo tiempo la interpretabilidad económica de los resultados. A continuación, se define la variable objetivo utilizada que nos permite clasificar a las empresas según su rentabilidad futura.

A. VARIABLE OBJETIVO (TARGET):

La variable objetivo utilizada en este estudio fue construida a partir del segundo conjunto de datos (*All Stocks in 5 yrs*), que contiene información histórica de precios de cierre de las acciones. Esta variable es de tipo binaria e indica si una empresa ha sido rentable (1) o no rentable (0) durante el periodo analizado. Para su construcción, se calculó el retorno porcentual entre el primer y el último precio de cierre disponible para cada empresa.

Esta fue la siguiente fórmula utilizada para calcular la variable objetivo:

Tabla 1. Fórmula utilizada para calcular la variable objetivo

$$\text{Return\_future} = \left(\frac{\text{Precio final} - \text{Precio inicial}}{\text{Precio inicial}} \right) \times 100$$

Fuente: Fuente Propia

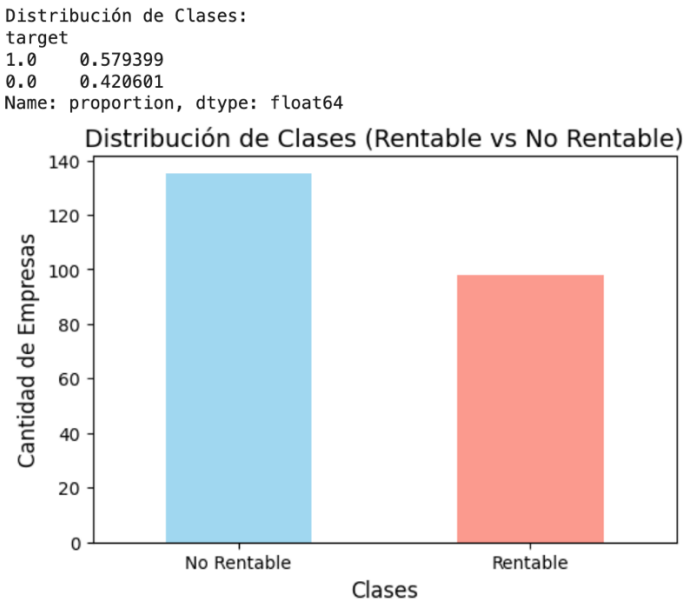
Con el fin de evitar una clasificación arbitraria y reducir el sesgo que podría generar una definición absoluta de rentabilidad, se optó por una estrategia basada en percentiles. En concreto, se utilizaron los percentiles 30 y 70 como puntos de corte. Estos límites de percentiles se utilizaron para separar de forma más nítida las empresas claramente rentables de aquellas que presentaron bajo desempeño. Además, esta selección de límites permite mantener un número suficiente de observaciones en cada clase, asegurando así el equilibrio estadístico necesario.

Aquellas empresas cuyo retorno se situó por debajo del percentil 30 fueron clasificadas como no rentables (0), mientras que las que se ubicaron por encima del percentil 70 fueron consideradas rentables (1). Las empresas en el rango intermedio fueron excluidas del modelo para asegurar una mayor diferenciación entre clases y así facilitar un entrenamiento más robusto del algoritmo de clasificación.

Este enfoque, común en estudios financieros y predictivos, permite definir clases bien separadas y reducir la ambigüedad en los casos marginales, lo que mejora tanto la precisión del modelo como su interpretabilidad.

Posteriormente, se analizó la distribución de las clases resultantes. Como muestra la siguiente gráfica, aproximadamente el 42% de las compañías fueron clasificadas como rentables, mientras que el 58% restante correspondió a empresas no rentables. Aunque la distribución no es perfectamente equilibrada, el desbalance es moderado y no se consideró necesario aplicar técnicas adicionales de balanceo de clases.

Gráfico 10. Distribución de clases



Fuente: Elaboración propia a partir del DataSet

Esta distribución sirvió como base para la construcción de los modelos predictivos presentados en los apartados siguientes, asegurando que las métricas de evaluación consideraran adecuadamente el comportamiento de ambos grupos.

La definición y análisis de la variable objetivo constituyen un paso esencial para garantizar la solidez del enfoque predictivo adoptado. Sin embargo, la calidad de los modelos no solo depende de una adecuada definición de la salida, sino también de la correcta selección de las variables explicativas que alimentarán el proceso de aprendizaje. A continuación, se detallan las características financieras seleccionadas como predictores, fundamentales para modelar de forma precisa el comportamiento futuro de las empresas del índice S&P 500.

B. VARIABLES EXPLICATIVAS:

Tabla 2. Descripción de cada variable financiera seleccionada

Variable	Descripción
Market Capitalization	Representa el valor total en dólares del mercado de las acciones en circulación de una empresa. Se utiliza frecuentemente para estimar el tamaño de una compañía, en lugar de basarse en las ventas o el valor total de los activos. En procesos de adquisición, la capitalización bursátil ayuda a determinar si una empresa objetivo supone una oportunidad de valor razonable para el adquirente.
Revenue	Es el dinero generado por las operaciones comerciales normales, calculado como el precio medio de venta multiplicado por el número de unidades vendidas. Es la cifra de ingresos brutos a la que se restan los costes para determinar los ingresos netos.
Cash per Share	Es el dinero generado por las operaciones normales de la empresa, calculado como el precio medio de venta multiplicado por el número de unidades vendidas. Es la cifra de ingresos brutos a la que se restan los costes para determinar los ingresos netos.
Current Stock Price	Es el precio de venta más reciente de una acción, divisa, materia prima o metal precioso que se negocia en bolsa y es el indicador más fiable del valor actual de ese valor.
Price-to-Earnings (P/E)	Mide el precio de las acciones de una empresa en relación con sus beneficios por acción (EPS). A menudo denominado múltiplo de precio o de beneficios, el ratio P/E ayuda a evaluar el valor relativo de las acciones de una empresa.
Price-to-Sales (P/S)	Es un coeficiente de valoración que compara el precio de las acciones de una empresa con sus ingresos. Es un indicador del valor que los mercados financieros atribuyen a cada dólar de ventas o ingresos de una empresa.
Price-to-Book (P/B)	Compara la capitalización bursátil de una empresa con su valor contable y localizar empresas infravaloradas.
Total Debt to Equity	Se calcula dividiendo el pasivo total de una empresa por el total de fondos propios. Este ratio ayuda a determinar cuánta deuda utiliza una empresa para financiar sus operaciones en comparación con el capital que han invertido los accionistas.
Return on Assets (ROA)	Es el porcentaje de ventas que queda después de contabilizar el COGS y los gastos de explotación normales, y se evalúa en relación con los costes y gastos. Se analiza en comparación con los activos para ver la eficacia de una empresa en el despliegue de activos para generar ventas y beneficios.
Return on Equity (ROE)	Es un ratio clave para los accionistas, ya que mide la capacidad de una empresa para rentabilizar sus inversiones en capital. El ROE, calculado como el beneficio neto dividido por los fondos propios, puede aumentar sin inversiones adicionales de capital.
Operating Margin	Es el porcentaje de ventas que queda después de contabilizar el COGS y los gastos de explotación normales.
Net Profit Margin	Es una medida de los beneficios (o ingresos netos) que genera una empresa.
Beta	Es la segunda letra del alfabeto griego utilizada en finanzas para denotar la volatilidad o el riesgo sistemático de un valor o una cartera en comparación con el mercado, normalmente el S&P 500, que tiene una beta de 1,0. Los valores con betas superiores a 1,0 se interpretan como más volátiles que el S&P 500.
Performance (Year)	Es una medida subjetiva de la capacidad de una empresa para utilizar los activos de su actividad principal y generar ingresos, en un año.
Performance (To Date)	Es una medida subjetiva de la capacidad de una empresa para utilizar los activos de su actividad principal y generar ingresos, hasta la fecha.
Quick Ratio	Mide la capacidad de una empresa para hacer frente a sus obligaciones corrientes (pagaderas en el plazo de un año) con el total de sus activos corrientes, como efectivo, cuentas por cobrar e inventarios. Mide la capacidad de una empresa para hacer frente a sus obligaciones a corto plazo con sus activos más líquidos y, por tanto, excluye los inventarios de su activo Corriente.
Dividend Yield (Annual)	Es un ratio financiero que muestra cuánto paga una empresa en dividendos cada año en relación con el precio de sus acciones.

Fuente: Elaboración propia a partir de los datos de (Investopedia, s.f., traducción propia).

3.4 METODOLOGÍA ANALÍTICA:

Definidas las variables relevantes y la variable objetivo, se procedió a la implementación de los modelos predictivos siguiendo un enfoque metodológico riguroso. El desarrollo del análisis se llevó a cabo utilizando el lenguaje de programación Python en el entorno colaborativo Google Colab, apoyándose en librerías especializadas como pandas para la manipulación de datos, scikit-learn para la construcción y evaluación de modelos de clasificación, y TensorFlow para el diseño de redes neuronales.

En primer lugar, se aplicaron transformaciones adicionales sobre el conjunto de datos con el fin de optimizar su adecuación para el entrenamiento de los algoritmos. Las variables numéricas fueron sometidas a un proceso de normalización, asegurando así que todas las características operaran en una escala comparable y evitando que aquellas con magnitudes mayores dominaran el entrenamiento. De manera complementaria, las variables categóricas fueron transformadas mediante codificación One-Hot, permitiendo su inclusión en los modelos sin introducir sesgos asociados al orden implícito que podría implicar otro tipo de codificación. Tras este preprocesamiento, el conjunto de datos fue dividido en dos subconjuntos independientes: un conjunto de entrenamiento (80% de las observaciones) y un conjunto de prueba (20%), con el objetivo de validar los modelos sobre datos no vistos previamente y evaluar su capacidad de generalización.

En cuanto al desarrollo de los modelos predictivos, se implementaron cinco algoritmos de clasificación ampliamente reconocidos en la literatura especializada: regresión logística, árbol de decisión, random forest, gradient boosting y red neuronal multicapa. Para cada uno de ellos se llevó a cabo un ajuste sistemático de hiperparámetros mediante la técnica de búsqueda en rejilla (Grid Search). Este procedimiento permitió explorar distintas combinaciones de parámetros críticos —como la profundidad máxima de los árboles, el número de estimadores o la tasa de aprendizaje— a fin de optimizar el rendimiento de cada modelo y reducir el riesgo de sobreajuste.

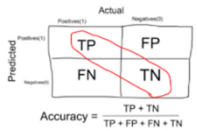
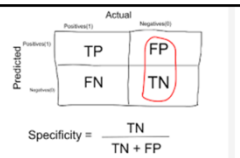
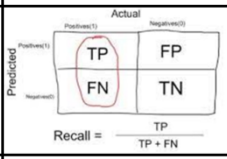
La validación de los modelos se realizó aplicando un esquema de validación cruzada estratificada de cinco particiones (5-fold Stratified Cross-Validation), el cual asegura que la proporción de clases se mantenga constante en cada partición, proporcionando así una estimación más robusta y menos sesgada del desempeño de los modelos.

Para evaluar el rendimiento predictivo, se recurrió a diversas métricas de clasificación que permiten una valoración integral de los resultados: precisión (accuracy), sensibilidad (recall), especificidad, área bajo la curva ROC (AUC-ROC) y el análisis de la matriz de confusión. Estas métricas no solo proporcionan información sobre el porcentaje de aciertos generales, sino que también permiten examinar el comportamiento de los modelos frente a los errores de tipo I y tipo II, aspectos fundamentales en problemas de clasificación binaria.

Finalmente, el modelo seleccionado como óptimo fue aquel que logró el mejor equilibrio entre precisión, sensibilidad y área bajo la curva ROC, garantizando así un desempeño robusto y una adecuada capacidad de generalización en el contexto de predicción de la rentabilidad futura de las empresas del índice S&P 500.

A continuación, se presentan las principales métricas utilizadas para la evaluación del desempeño de los modelos de clasificación, junto con su definición y la fórmula matemática correspondiente.

Tabla 3. Precisión, Especificidad, Sensibilidad y Matriz de Confusión

MÉTRICAS	DEFINICIÓN	FÓRMULA MATEMÁTICA
PRECISIÓN	**Porcentaje total de elementos clasificados correctamente.**	
ESPECIFICIDAD	**Número de ítems correctamente identificados como negativos fuera del total de negativos.**	
SENSIBILIDAD	**Número de elementos identificados correctamente como positivos del total de positivos verdaderos.**	
MATRIZ DE CONFUSIÓN	**Tabla que describe el rendimiento de un modelo supervisado de Machine Learning en los datos de prueba, donde se desconocen los verdaderos valores.**	**Verdaderos positivos (VP):** Casos correctamente identificados como positivos. **Falsos positivos (FP):** Casos incorrectamente identificados como positivos. **Verdaderos negativos (VN):** Casos correctamente identificados como negativos. **Falsos negativos (FN):** Casos incorrectamente identificados como negativos.

Fuente: Elaboración propia a partir de Data, S. B. (2023, 18 de octubre).

3.5 Modelos predictivos

Con el conjunto de datos preprocesado y la metodología de validación definida, se procedió a la construcción y evaluación de los modelos de Machine Learning seleccionados.

Modelo Random Forest

El primer modelo que implementamos en el estudio fue el Random Forest, este algoritmo es un método de aprendizaje supervisado basado en la construcción de múltiples árboles de decisión, cuyas predicciones se combinan para generar una salida agregada más robusta y precisa. Esta técnica mejora la estabilidad del modelo, reduce la varianza y mitiga el riesgo de sobreajuste. Además, se adapta bien a datos complejos y de alta dimensión, siendo especialmente útil en contextos con muchas variables correlacionadas.

Según Louppe (2014), su fortaleza radica en su capacidad para manejar automáticamente la selección de variables y ofrecer medidas de importancia de predictores, lo que lo convierte en una herramienta poderosa tanto para predicción como para interpretación de datos complejos.

Implementamos el modelo Random Forest utilizando la biblioteca de aprendizaje automático Scikit-learn, aprovechando su capacidad para manejar variables complejas y múltiples interacciones entre ellas.

Los resultados iniciales obtenidos por este modelo indican un rendimiento destacable, alcanzando una precisión general del 85,1%. Este valor refleja que aproximadamente el 85% de las empresas del conjunto de prueba fueron correctamente clasificadas como "Rentables" o "No Rentables".

Si analizamos en profundidad, la matriz de confusión muestra un desempeño particularmente fuerte en la detección de empresas rentables, con una sensibilidad (recall) del 96%. Esto implica que prácticamente todas las empresas rentables fueron identificadas correctamente, lo que es clave para aplicaciones prácticas donde se busque maximizar oportunidades de inversión. En contraste, la clasificación de empresas no

rentables presentó una sensibilidad del 70%**,** indicando cierto margen de mejora en la clasificación de esta categoría. Sin embargo, el resultado sigue siendo adecuado, y la precisión general del modelo se mantiene alta.

Tabla 4. Resultados Random Forest

```
Random Forest:
Accuracy: 0.851063829787234
Confusion Matrix:
 [[14  6]
 [ 1 26]]
Classification Report:
              precision    recall  f1-score   support

         0.0       0.93      0.70      0.80        20
         1.0       0.81      0.96      0.88        27

    accuracy                           0.85        47
   macro avg       0.87      0.83      0.84        47
weighted avg       0.86      0.85      0.85        47
```

Fuente: Elaboración propia a partir del DataSet

Para poder mejorar, aún más la capacidad predictiva del modelo, posteriormente se procedió a la optimización mediante el ajuste de hiperparámetros del Random Forest. Estos hiperparámetros son configuraciones específicas del modelo que afectan directamente a su rendimiento y su generalización. Para ello, realizamos una búsqueda exhaustiva de combinaciones posibles utilizando un método llamado Grid Search. Este evalúa distintas configuraciones para encontrar aquella que maximice la precisión del modelo.

Estos hiperparámetros, incluyen el número de árboles generados en el modelo, la profundidad máxima permitida para cada árbol, el mínimo número de observaciones necesarias para dividir un nodo, el mínimo número de observaciones requeridas para formar una hoja y el ajuste automático del peso de las clases para contrarrestar posibles desequilibrios. Seleccionamos estos hiperparámetros porque tienen una influencia directa sobre la complejidad y capacidad del modelo para adaptarse correctamente a los datos, buscando evitar errores comunes como el sobreajuste o una generalización deficiente.

Una vez aplicados los ajustes mediante Grid Search, observamos una mejora en los resultados del modelo optimizado de Random Forest. La precisión general aumentó del

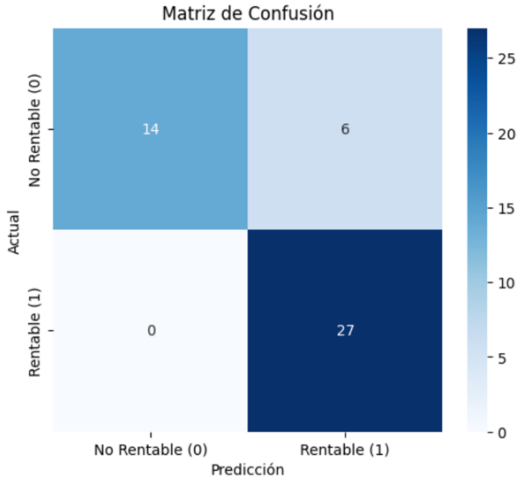
85,1% inicial a un 87,2%, lo que indica una mejor capacidad de clasificación en el conjunto de prueba.

Tabla 5. Resultados Random Forest con hiperparámetros

```
Random Forest ajustado:
Accuracy: 0.8723404255319149
Confusion Matrix:
 [[14  6]
 [ 0 27]]
Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      0.70      0.82        20
         1.0       0.82      1.00      0.90        27

    accuracy                           0.87        47
   macro avg       0.91      0.85      0.86        47
weighted avg       0.90      0.87      0.87        47
```

Fuente: Elaboración propia a partir del DataSet

Tabla 6. Matriz de confusión del Random Forest con hiperparámetros



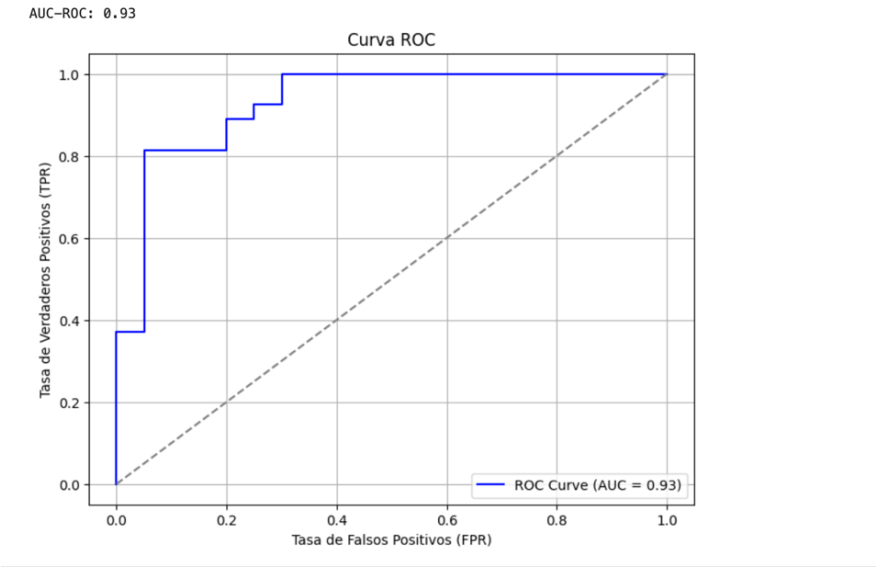
Fuente: Elaboración propia a partir del DataSet

Analizando la matriz de confusión del modelo ajustado, se observa que todas las empresas clasificadas como "Rentables" fueron correctamente identificadas (27 casos), lo que implica un recall del 100% para la clase rentable. Este resultado es especialmente valioso en contextos de inversión, ya que garantiza que ninguna empresa con alto retorno futuro haya sido omitida por el modelo.

Por otro lado, el modelo clasificó correctamente a 14 de las 20 empresas "No Rentables", presentando un recall del 70% para esta clase. Se produjeron 6 falsos positivos, es decir, 6 empresas no rentables fueron clasificadas incorrectamente como rentables. Aunque este margen de error es asumible, indica un posible sesgo hacia la predicción de rentabilidad, lo cual puede explicarse por la priorización del modelo en maximizar la identificación de oportunidades de inversión.

Finalmente, la curva ROC nos ofrece una evaluación más profunda del desempeño general del modelo, mostrando la capacidad del algoritmo para distinguir correctamente entre empresas rentables y no rentables a lo largo de diferentes niveles de probabilidad. La métrica AUC asociada a esta curva es del 0.93, un resultado excelente, ya que una puntuación cercana a 1 indica una capacidad predictiva muy alta. Esto implica que nuestro modelo tiene una gran habilidad para diferenciar correctamente entre empresas rentables y no rentables.

Gráfico 11. Curva AUC-ROC para Random Forest con hiperparámetros



Fuente: Elaboración propia a partir del DataSet

En definitiva, ambas evaluaciones confirman la fiabilidad y solidez del modelo Random Forest optimizado, reforzando su utilidad como herramienta efectiva para la predicción y toma de decisiones financieras sobre empresas potencialmente rentables en el futuro.

Modelo árbol de decisión

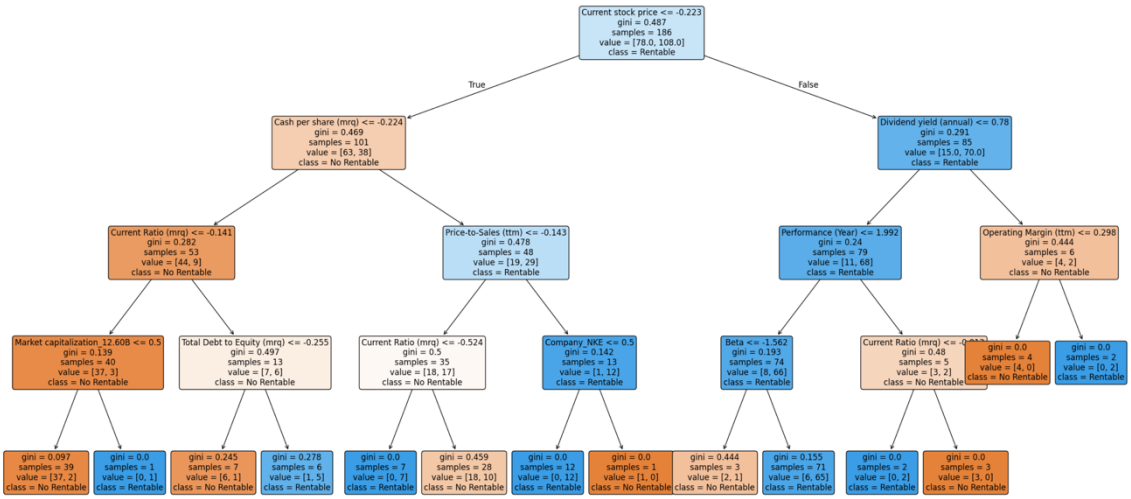
Siguiendo con el análisis, se ha implementado un modelo de árbol de decisión, es un modelo de clasificación ampliamente utilizado por su estructura jerárquica clara y su alto grado de interpretabilidad. Este algoritmo divide iterativamente el conjunto de datos en subconjuntos más homogéneos mediante reglas de decisión basadas en los valores de las variables independientes. Como explica Quinlan (1986), cada nodo interno representa una prueba sobre un atributo, y cada hoja terminal una predicción de clase, lo que facilita la comprensión del razonamiento seguido por el modelo. Además, su transparencia lo convierte en una herramienta valiosa en contextos donde la explicación del proceso de decisión es tan importante como la precisión del resultado (Safavian & Landgrebe, 1991).

El árbol de Decisión generado en este análisis identifica inicialmente la característica "Current stock price" (Precio actual de la acción) como la variable más importante para dividir las empresas en rentables y no rentables.

En este caso, el árbol identifica inicialmente la variable “Current stock Price” (Precio actual de la acción) como el mejor criterio para realizar la primera partición. Las empresas con precios más bajos tienden a clasificarse como “No Rentables”**,** mientras que aquellas con precios más altos se asocian con la categoría “Rentable”.

A partir de esta división principal, el modelo utiliza sucesivamente otras variables relevantes para afinar la clasificación, como “Cash per share” (Efectivo por acción), “Price-to-Sales**”**, “Dividend yield”, “Performance Year” y “Operating Margin”, entre otras. Cada nodo adicional refleja una evaluación lógica basada en indicadores financieros clave, buscando maximizar la pureza de las clasificaciones y separar con mayor precisión las empresas rentables de las no rentables.

Gráfico 12. Representación del ML de árbol de decisión



Fuente: Elaboración propia a partir del DataSet

Analizando los resultados cuantitativos, el modelo alcanza una precisión general del 74,4%, un rendimiento aceptable pero inferior al del modelo de Random Forest, que alcanzó un 85,1%. Esto sugiere que, si bien el árbol de decisión ofrece una excelente capacidad interpretativa, en términos de exactitud predictiva puede verse superado por métodos más robustos como los ensambles.

Tabla 7. Resultados árbol de decisión

```
Matriz de Confusión:
 [[17  3]
 [ 9 18]]

Reporte de Clasificación:
              precision    recall  f1-score   support

         0.0       0.65      0.85      0.74        20
         1.0       0.86      0.67      0.75        27

    accuracy                           0.74        47
   macro avg       0.76      0.76      0.74        47
weighted avg       0.77      0.74      0.75        47
```

Fuente: Elaboración propia a partir del DataSet

Aunque presenta un buen desempeño en la detección de empresas no rentables (recall del 85%), su capacidad para identificar correctamente a las rentables es menor (recall del 67%). Esta diferencia se debe a que el Random Forest combina múltiples árboles, lo que

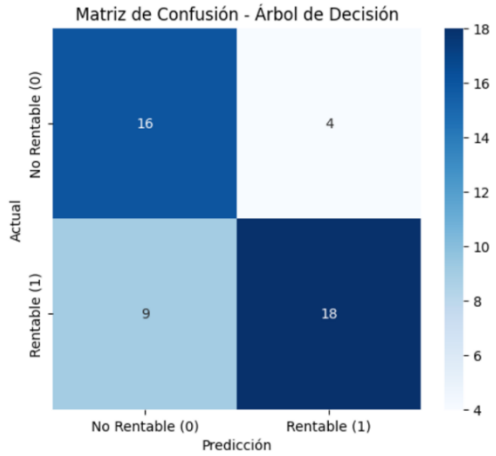
reduce el sobreajuste y mejora la estabilidad del modelo. En contraste, el Árbol de Decisión, si bien es útil por su interpretabilidad, tiene un rendimiento predictivo más limitado.

Para poder mejorar esta precisión, introducimos hiperparámetros al modelo. Este ajuste incluye el control de la profundidad máxima del árbol, buscando limitar su complejidad y reducir el riesgo de sobreajuste; también se definió un mínimo número de muestras requerido para realizar cada división del árbol, así como un número mínimo de muestras necesarias en cada hoja final del mismo. Finalmente, se evaluaron diferentes criterios para determinar la calidad de las divisiones internas del árbol, específicamente mediante los índices "gini" y "entropía".

A pesar de este esfuerzo de optimización, los resultados del Árbol de Decisión ajustado no presentaron una mejora significativa respecto al modelo inicial. De hecho, la precisión general del modelo descendió al 63,8%, mostrando limitaciones claras en su capacidad predictiva. Esta situación puede explicarse por una excesiva restricción del modelo, provocada por hiperparámetros demasiado conservadores que impidieron al árbol capturar relaciones complejas presentes en los datos. Por ejemplo, limitar en exceso la profundidad máxima o imponer umbrales elevados para la división de nodos puede conducir a un subajuste (underfitting), donde el modelo es incapaz de aprender adecuadamente los patrones del conjunto de entrenamiento.

La matriz de confusión muestra que el modelo identificó correctamente a 14 empresas como "No Rentables" y 19 como "Rentables", pero cometió 14 errores en total. Esto sugiere una capacidad predictiva moderada.

Tabla 8. Matriz de confusión del árbol de decisión con hiperparámetros



Fuente: Elaboración propia a partir del DataSet

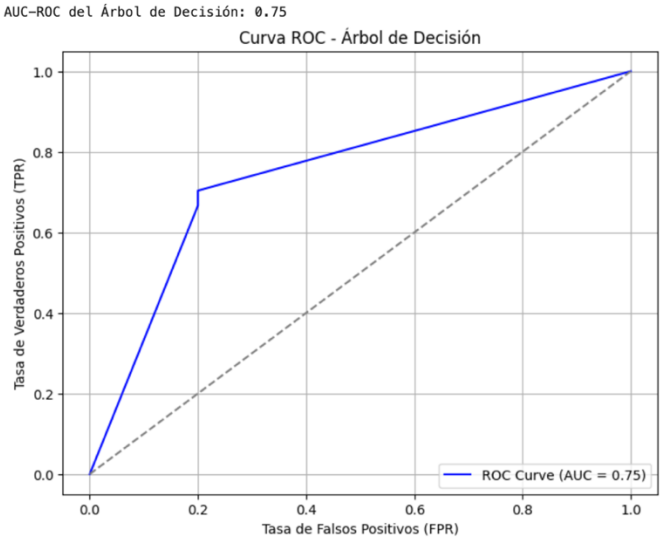
Tabla 9. Resultados Árbol de decisión con hiperparámetros

```
Accuracy del Arbol Ajustado: 0.6382978723404256

Mejores Hiperparámetros: {'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 4, 'min_samples_split': 10}

Reporte de Clasificación Ajustado:
              precision    recall  f1-score   support

         0.0       0.57      0.65      0.60        20
         1.0       0.71      0.63      0.67        27

    accuracy                           0.64        47
   macro avg       0.64      0.64      0.64        47
weighted avg       0.65      0.64      0.64        47
```

Fuente: Elaboración propia a partir del DataSet

Además, el resultado dado a través de la curva ROC con un AUC de 0.75 refleja que, aunque el modelo tiene cierta capacidad de distinguir entre empresas rentables y no rentables, esta capacidad es moderada y claramente inferior al modelo Random Forest optimizado anteriormente.

Gráfico 13. Curva AUC-ROC para Árbol de decisión con hiperparámetros



Fuente: Elaboración propia a partir del DataSet

Estos resultados nos permiten concluir que, aunque el Árbol de Decisión es una herramienta valiosa para la interpretación visual y sencilla los resultados, pero a la vez presenta limitaciones en precisión y estabilidad comparado con modelos más robustos, como el Random Forest ajustado previamente.

Tras evaluar estos dos modelos anteriores, se procedió a entrenar modelos más avanzados.

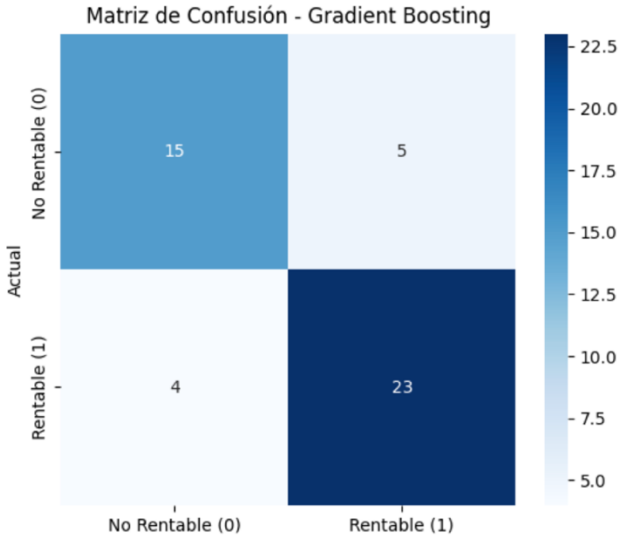
Modelo Gradient Boosting

"El Gradient Boosting es una técnica de aprendizaje automático utilizada para tareas de regresión y clasificación. Construye modelos secuencialmente, cada uno de los cuales intenta corregir los errores del anterior. A diferencia de otros algoritmos que se centran en un único modelo, Gradient Boosting combina varios modelos débiles (normalmente árboles de decisión) para formar un modelo predictivo sólido". (Kashyap, 2020, traducción propia).

En este caso, el modelo fue construido ajustando directamente los hiperparámetros clave, como la tasa de aprendizaje (learning rate), el número de estimadores (n_estimators) y la profundidad máxima del árbol (max_depth), con el objetivo de lograr un equilibrio entre precisión y riesgo de sobreajuste. Los resultados obtenidos muestran un rendimiento superior al del árbol de decisión individual, aunque ligeramente inferior al modelo de

Random Forest optimizado. Esto respalda la eficacia del enfoque basado en el ensamblado secuencial de árboles como estrategia robusta de predicción (Kashyap, 2020, traducción propia).

Tabla 10. Matriz de confusión Gradient boosting con hiperparámetros



Fuente: Elaboración propia a partir del DataSet

Tabla 11. Resultados Gradient Boosting con Hiperparámetros

Reporte de Clasificación - Gradient Boosting:

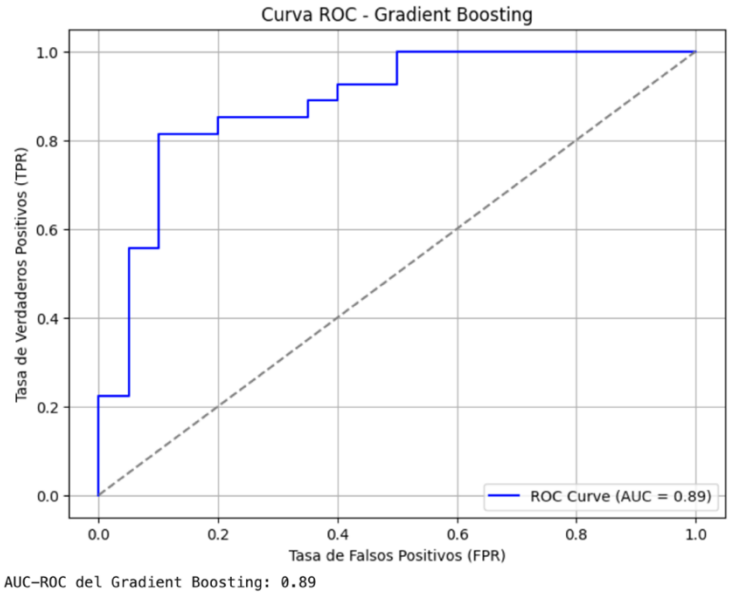
	precision	recall	f1-score	support
0.0	0.79	0.75	0.77	20
1.0	0.82	0.85	0.84	27
accuracy			0.81	47
macro avg	0.81	0.80	0.80	47
weighted avg	0.81	0.81	0.81	47

Fuente: Elaboración propia a partir del DataSet

Los resultados obtenidos a través de la matriz de confusión del modelo de Gradient Boosting evidencian un rendimiento robusto tras el ajuste de hiperparámetros. Obteniendo una precisión del 81%, clasificando correctamente la mayoría de las observaciones tanto de empresas rentables como no rentables. Destaca especialmente su capacidad de detección de empresas rentables, con una sensibilidad del 85.2%**,** lo que

refuerza su utilidad en contextos donde es crucial identificar oportunidades de rentabilidad futura.

Gráfico 14. Curva AUC-ROC Gradient Boosting



Fuente: Elaboración propia a partir del DataSet

La curva ROC del modelo Gradient Boosting, alcanza un AUC de 0.89, lo que indica una excelente capacidad de discriminación entre empresas rentables y no rentables. Este valor refleja un equilibrio sólido entre la tasa de verdaderos positivos y falsos positivos, posicionando al modelo como una opción eficaz para predicción binaria en contextos financieros.

En comparación, el árbol de decisión obtuvo un AUC inferior, evidenciando una menor capacidad para capturar patrones complejos. Aunque el Random Forest optimizado presentó una precisión general superior, su AUC fue ligeramente más bajo (0,87), lo que sugiere que el Gradient Boosting ofrece una mejor diferenciación entre clases en términos globales.

Después de evaluar modelos más complejos como el árbol de decisión, Random Forest y Gradient Boosting, se decidió implementar también un modelo de regresión logística con el fin de establecer un punto de comparación con una técnica estadística clásica, ampliamente reconocida por su simplicidad e interpretabilidad.

Modelo de Regresión Logística

"El modelo de regresión logística es una técnica estadística utilizada para predecir la probabilidad de que una observación pertenezca a una de dos categorías posibles, a partir de un conjunto de variables independientes. A diferencia de una regresión lineal, en este modelo se utiliza una función sigmoide que transforma la salida en una probabilidad entre 0 y 1, lo que lo convierte en una herramienta eficaz para problemas de clasificación binaria "(AprendeIA, 2022). Esta capacidad para modelar relaciones no lineales, junto con su interpretabilidad, hace que sea ampliamente utilizado en análisis financieros y de riesgo.

En este estudio, se implementó optimizando hiperparámetros mediante una búsqueda en rejilla (GridSearchCV), incluyendo el coeficiente de regularización C, el tipo de penalización (l1 o l2), el solver y el ajuste de pesos por clase (class_weight='balanced') para manejar posibles desequilibrios. Esta configuración permitió adaptar el modelo a los datos disponibles, equilibrando la precisión predictiva con una elevada interpretabilidad.

Tabla 12. Resultados Regresión Logística con Hiperparámetros

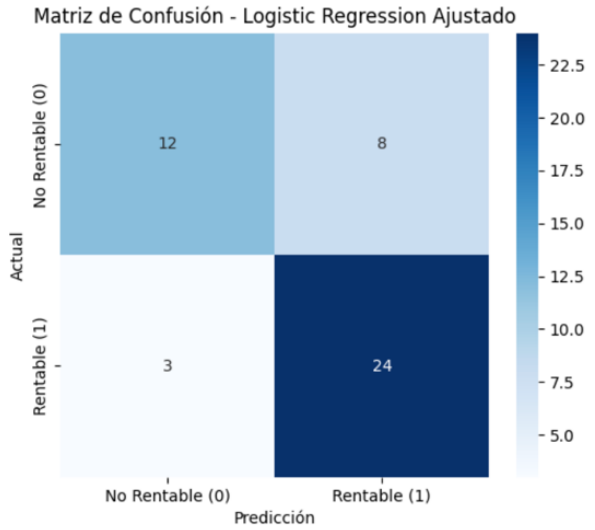
Reporte de Clasificación con Mejores Hiperparámetros:

	precision	recall	f1-score	support
0.0	0.80	0.60	0.69	20
1.0	0.75	0.89	0.81	27
accuracy			0.77	47
macro avg	0.78	0.74	0.75	47
weighted avg	0.77	0.77	0.76	47

AUC-ROC con Mejores Hiperparámetros: 0.82

Fuente: Elaboración propia a partir del DataSet

Tabla 13. Matriz de confusión Regresión Logística con hiperparámetros

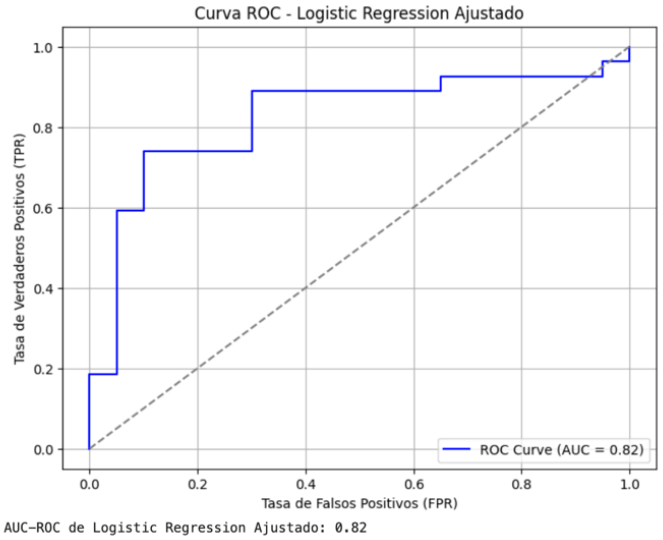


Fuente: Elaboración propia a partir del DataSet

La matriz de confusión mostró que el modelo clasificó correctamente a 24 empresas rentables y 12 no rentables, con un total de 11 errores de clasificación. El modelo demostró una mayor capacidad para identificar correctamente empresas rentables (recall = 0.89), lo cual es particularmente valioso en este estudio, ya que el objetivo es anticipar las empresas con mayor probabilidad de ser rentables en el futuro.

El reporte de clasificación reveló un equilibrio razonable entre las métricas clave, con una precisión del 75% para la clase rentable y un F1-score de 0.81. Estos resultados confirman que la regresión logística ajustada ofrece un modelo fiable, interpretable y con buen rendimiento, lo que la convierte en una herramienta útil para la predicción de rentabilidad empresarial.

Gráfico 15. Curva AUC-ROC Regresión Logística con hiperparámetros



Fuente: Elaboración propia a partir del DataSet

Este enfoque permitió mejorar la capacidad predictiva del modelo, alcanzando una precisión global del 77% y un AUC-ROC de 0.82, lo que indica una buena capacidad de discriminación entre empresas rentables y no rentables.

Modelo Redes Neuronales

Por último, exploramos enfoques más complejos, el modelo de Red Neuronal. Este modelo está compuesto de dos capas ocultas con 64 y 32 neuronas respectivamente, ambas con función ReLU, que permite capturar relaciones no lineales de forma eficiente y evita el problema del desvanecimiento del gradiente.

La capa de salida utiliza la función sigmoide, adecuada para problemas de clasificación binaria, con el objetivo de predecir si una empresa será rentable o no

Para prevenir el sobreajuste, se incorporaron capas de dropout con una tasa del 30% después de cada capa oculta. Esta técnica desactiva aleatoriamente una fracción de las neuronas en cada iteración durante el entrenamiento, favoreciendo la generalización del modelo. El modelo se entrenó durante 50 épocas, un valor que permite un entrenamiento suficiente para que el modelo aprenda patrones relevantes sin caer en sobre entrenamiento. Este número se eligió tras pruebas preliminares, donde se observó que la función de pérdida se estabilizaba antes de las 50 épocas.

Además, se utilizó un tamaño de lote de 16**,** ya que este valor permite un equilibrio adecuado entre velocidad de entrenamiento y precisión del gradiente. Por último, se utilizó el optimizador Adam con una tasa de aprendizaje de 0.0001, elegida tras pruebas con valores más altos como 0.001, donde se observó una menor estabilidad.

La matriz de confusión mostró un rendimiento bastante balanceado: el modelo clasificó correctamente a 23 empresas rentables y 16 no rentables, con solo 8 errores en total. Esto refleja tanto una buena sensibilidad (recall = 0.85 para rentables) como especificidad (recall = 0.80 para no rentables), valores cruciales en contextos financieros donde la clasificación errónea puede tener un impacto significativo.

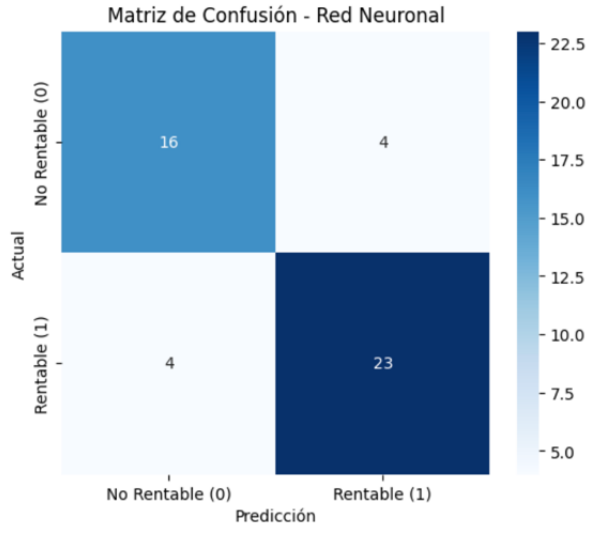
Tabla 14. Matriz de confusión Red Neuronal con hiperparámetros

Reporte de Clasificación – Red Neuronal:

	precision	recall	f1-score	support
0.0	0.80	0.80	0.80	20
1.0	0.85	0.85	0.85	27
accuracy			0.83	47
macro avg	0.83	0.83	0.83	47
weighted avg	0.83	0.83	0.83	47

Fuente: Elaboración propia a partir del DataSet

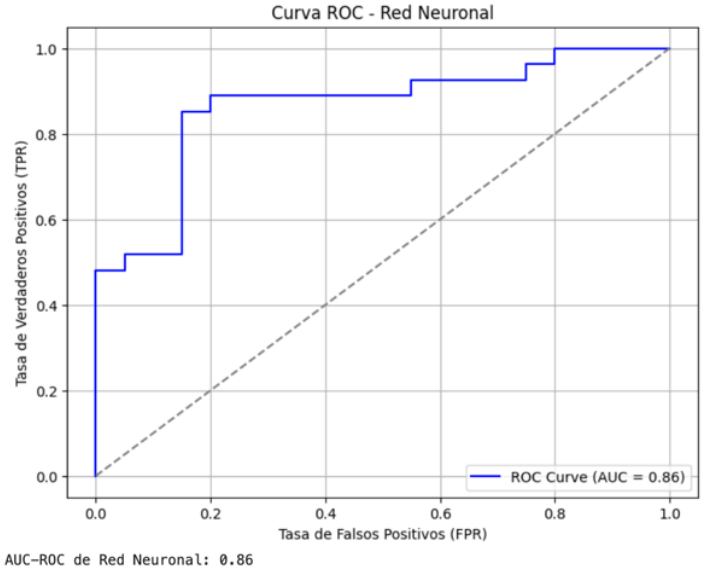
Tabla 15. Matriz de confusión Red Neuronal con hiperparámetros



Fuente: Elaboración propia a partir del DataSet

Se obtuvo una precisión de un 83%, y los resultados de la Curva AUC-ROC nos dieron 86%, lo que indica una alta capacidad discriminativa del modelo para distinguir entre empresas rentables y no rentables.

Gráfico 16. Curva AUC-ROC Red Neuronal



Fuente: Elaboración propia a partir del DataSet

Desde el punto de vista interpretativo, aunque las redes neuronales presentan menor transparencia que otros modelos como los árboles de decisión o la regresión logística, su capacidad para modelar relaciones complejas no lineales y su buen rendimiento justifican su uso en tareas de predicción financiera como la presente. A continuación, se presenta una tabla resumen que recoge los principales resultados obtenidos por cada uno de los modelos predictivos, lo que permite comparar su rendimiento de forma clara y estructurada.

Tabla 16. Resumen de los Resultados de los Modelos

Modelo	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Random Forest	**87%**	81%	**100%**	**90%**	**93%**
Decision Tree	72%	82%	67%	73%	75%
Gradient Boosting	79%	81%	81%	81%	84%
Logistic Regression	77%	75%	89%	81%	82%
Neural Network	83%	**83%**	89%	86%	85%

Fuente: Elaboración Propia a partir de los resultados de los modelos Predictivos

4. CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACION

Tras la implementación y comparación de cinco modelos de aprendizaje automático, se concluye que el modelo de Random Forest ha demostrado ser el que ofrece mejor rendimiento global para predecir la rentabilidad futura de las empresas del índice S&P 500. Este modelo destaca por su elevada precisión (87%**)**, su alta capacidad discriminativa (AUC-ROC = 93%) y, especialmente, por su notable sensibilidad en la identificación de empresas rentables. Estas características lo convierten en una herramienta particularmente valiosa en el ámbito financiero, donde las decisiones deben orientarse a maximizar el retorno esperado minimizando el riesgo. Adicionalmente, su robustez frente al sobreajuste y su capacidad para modelar interacciones complejas entre múltiples variables refuerzan su idoneidad para tareas de clasificación en contextos empresariales.

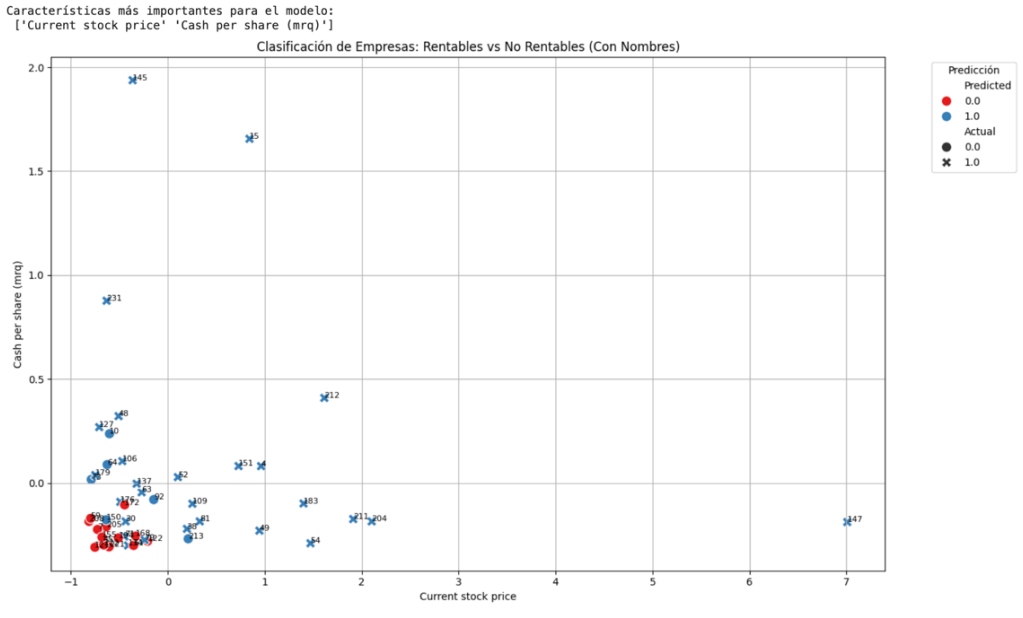
Aunque otros modelos como Gradient Boosting (AUC = 84%) y la Red Neuronal (AUC = 85%) ofrecieron resultados sólidos, su mayor complejidad computacional y menor interpretabilidad los hacen menos adecuados para escenarios donde se requiere transparencia y buena generalización**.** Por su parte, el árbol de decisión, si bien aporta claridad interpretativa, presentó el rendimiento más bajo en términos de precisión y estabilidad, incluso tras el ajuste de hiperparámetros.

En consecuencia, el modelo Random Forest optimizado se posiciona como la opción más equilibrada y fiable dentro de este análisis empírico, siendo la alternativa más recomendable para tareas de clasificación binaria en el ámbito financiero, especialmente en aquellos escenarios donde se pretende anticipar la rentabilidad futura de empresas a partir de indicadores fundamentales.

Dado que el modelo de Random Forest ha sido seleccionado como el más adecuado en función de su desempeño global, resulta relevante analizar cómo clasifica visualmente las empresas en función de las variables que más contribuyen a su capacidad predictiva. Para ello, se ha procedido a representar gráficamente la clasificación realizada sobre el conjunto de prueba, utilizando las dos variables más influyentes identificadas por el propio modelo: el precio actual de la acción (Current Stock Price) y el efectivo por acción (Cash per-Share). En esta visualización, cada empresa se muestra como un punto

individual, cuyo color refleja la predicción del modelo (azul para rentables y rojo para no rentables), mientras que la forma del marcador indica su clasificación real. Esta codificación cruzada permite observar visualmente tanto los aciertos como los errores del modelo. El gráfico revela que las empresas clasificadas como rentables tienden a agruparse en zonas con mayor precio por acción y niveles elevados de efectivo por acción, reforzando así la lógica del modelo que asocia estos indicadores con una mayor probabilidad de rentabilidad futura. Esta representación no solo facilita la evaluación cualitativa del desempeño del modelo, sino que también aporta transparencia interpretativa sobre cómo esta toma sus decisiones.

Gráfico 17. Distribución de las empresas en el modelo Random Forest



Fuente: Elaboración propia a partir del DataSet

De cara a futuras investigaciones, sería pertinente explorar la aplicación de estos modelos predictivos en otros sectores económicos o geografías, como mercados emergentes o pequeñas y medianas empresas, cuyos patrones de rentabilidad pueden diferir significativamente. Asimismo, podría enriquecerse el análisis incorporando variables cualitativas o no estructuradas, como el sentimiento del mercado extraído de noticias financieras o redes sociales mediante técnicas de procesamiento del lenguaje natural (NLP).

Por último, una línea especialmente relevante sería evaluar la capacidad del modelo entrenado con datos históricos para anticipar resultados reales en el tiempo presente, comparando sus predicciones con datos actuales del año 2025. Esta validación temporal permitiría determinar en qué medida el modelo mantiene su precisión en entornos dinámicos, proporcionando así evidencia adicional sobre su aplicabilidad práctica en los procesos de toma de decisiones de inversión.

5. DECLARACIÓN USO HERRAMIENTA DE IA

Declaración de Uso de Herramientas de Inteligencia Artificial Generativa en Trabajos Fin de Grado

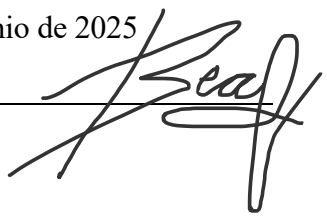
ADVERTENCIA: Desde la Universidad consideramos que ChatGPT u otras herramientas similares son herramientas muy útiles en la vida académica, aunque su uso queda siempre bajo la responsabilidad del alumno, puesto que las respuestas que proporciona pueden no ser veraces. En este sentido, NO está permitido su uso en la elaboración del Trabajo fin de Grado para generar código porque estas herramientas no son fiables en esa tarea. Aunque el código funcione, no hay garantías de que metodológicamente sea correcto, y es altamente probable que no lo sea.

Por la presente, yo, Beatriz Tejedor Canet, estudiante de Business Analytics y Relaciones Internacionales de la Universidad Pontificia Comillas al presentar mi Trabajo Fin de Grado titulado "Modelos de Machine Learning para la predicción de rentabilidad empresarial: Un Análisis del índice S&P 500", declaro que he utilizado la herramienta de Inteligencia Artificial Generativa ChatGPT u otras similares de IAG de código sólo en el contexto de las actividades descritas a continuación:

1. **Brainstorming de ideas de investigación:** Utilizado para idear y esbozar posibles áreas de investigación.
2. **Referencias:** Usado conjuntamente con otras herramientas, como Science, para identificar referencias preliminares que luego he contrastado y validado.
3. **Metodólogo:** Para descubrir métodos aplicables a problemas específicos de investigación.
4. **Interpretador de código:** Para realizar análisis de datos preliminares.
5. **Estudios multidisciplinares:** Para comprender perspectivas de otras comunidades sobre temas de naturaleza multidisciplinar.
6. **Constructor de plantillas:** Para diseñar formatos específicos para secciones del trabajo.
7. **Corrector de estilo literario y de lenguaje:** Para mejorar la calidad lingüística y estilística del texto.
8. **Sintetizador y divulgador de libros complicados:** Para resumir y comprender literatura compleja.
9. **Revisor:** Para recibir sugerencias sobre cómo mejorar y perfeccionar el trabajo con diferentes niveles de exigencia.
10. **Traductor:** Para traducir textos de un lenguaje a otro.

Afirmo que toda la información y contenido presentados en este trabajo son producto de mi investigación y esfuerzo individual, excepto donde se ha indicado lo contrario y se han dado los créditos correspondientes (he incluido las referencias adecuadas en el TFG y he explicitado para que se ha usado ChatGPT u otras herramientas similares). Soy consciente de las implicaciones académicas y éticas de presentar un trabajo no original y acepto las consecuencias de cualquier violación a esta declaración.

Fecha: 16 de junio de 2025

Firma: 

6. BIBLIOGRAFÍA:

Bargagli-Stoffi, F. J., Niederreiter, J., & Riccaboni, M. (2021). Supervised learning for the prediction of firm dynamics. En M. Riccaboni & F. J. Bargagli-Stoffi (Eds.), *Data Science for Economics and Finance* (pp. 19–41). Springer. https://doi.org/10.1007/978-3-030-66891-4_2.

Alanis, E., Chava, S., & Shah, A. (2022). *Benchmarking machine learning models to predict corporate bankruptcy*. arXiv. https://arxiv.org/abs/2212.12051.

Instituto O'Higgins. (2025). *El impacto de la inteligencia artificial y el machine learning en la valoración de activos financieros*. https://www.instituto-ohiggins.edu.ec/wp-content/uploads/2025/02/El-Impacto-de-la-Inteligencia-Artificial-y-el-Machine-Learning-en-la-Valoracion-de-Activos-Financieros.pdf.

Lizcano Álvarez, J., & Castelló Taliani, E. (2004). *Rentabilidad Empresarial*. Cámara de Comercio de España. https://www.camara.es/sites/default/files/publicaciones/rentab_emp.pdf.

Economipedia. (2024). *Rentabilidad: Qué es y qué tipos hay*. https://economipedia.com/definiciones/rentabilidad.html.

CEUPE. (2022). *Diferencia entre utilidad y rentabilidad*. https://www.ceupe.do/blog/diferencia-entre-utilidad-y-rentabilidad.html.

Universidad Europea de Madrid. (s.f.). *Rentabilidad económica y rentabilidad financiera: método de descomposición de DuPont*. https://www.academia.edu/42455954/Rentabilidad_econ%C3%B3mica_y_rentabilidad_financiera_m%C3%A9todo_de_descomposici%C3%B3n_de_Dupont.

Ramírez Orellana, A. (2006). *La RSC y la triple cuenta de resultados*. Recuperado de https://www.researchgate.net/publication/40969495_La_RSC_y_la_triple_cuenta_de_resultados.

Nagy, M., Valaskova, K., Kovalova, E., & Macura, M. (2024). Drivers of S&P 500's profitability: Implications for investment strategy and risk management. *Economies*, 12(4), 77. https://doi.org/10.3390/economies12040077.

Anand, V., Brunner, R., Ikegwu, K., & Sougiannis, T. (2019). *Predicting profitability using machine learning (S&P Global Market Intelligence Research Paper)*. SSRN. https://doi.org/10.2139/ssrn.3466478.

Artene, A. E., & Domil, A. E. (2025). Neural networks in accounting: Bridging financial forecasting and decision support systems. *Electronics, 14*(5), 993. https://doi.org/10.3390/electronics14050993.

Aguinis, H., & Glavas, A. (2017). *On corporate social responsibility, sensemaking, and the search for meaningfulness through work*. Journal of Management. https://hermanaguinis.com/pdf/JOMCSR2019.pdf.

Barney, J. B. (1991). *Firm resources and sustained competitive advantage*. Journal of Management, 17(1), 99–120. Recuperado de https://josephmahoney.web.illinois.edu/BA545_Fall%202022/Barney%20(1991).pdf.

Ramón Dangla, R., & Bañón Calatrava, C. (2022). *Stock de activos intangibles y rentabilidad empresarial: El caso de la industria hotelera española (2008–2019)*. Recuperado de https://www.researchgate.net/publication/359609867_Stock_de_activos_intangibles_y_rentabilidad_empresarial_El_caso_de_la_industria_hotelera_espanola_2008-2019.

T. Rowe Price. (2024). *Do high margins justify high valuations?* https://www.troweprice.com/en/us/insights/do-high-margins-justify-high-valuations.

Ajibade, S.-S. M., Jasser, M. B., Alebiosu, D. O., Al-Hadi, I. A. A.-Q., Al-Dharhani, G. S., Hassan, F., & Gyamfi, B. A. (2024). *Uncovering the dynamics in the application of machine learning in computational finance: A bibliometric and social network analysis*. International Journal of Economics and Financial Issues, 14(4), 299–315. Recuperado de https://ideas.repec.org/a/eco/journ1/2024-04-32.html.

Masini, R. P., Medeiros, M. C., & Mendes, E. F. (2021). Machine learning advances for time series forecasting. *Journal of Economic Surveys*, 37(1), 76–111. https://doi.org/10.1111/joes.12429.

Fuzail, M., Abid, M. K., Rehman, M., & Aslam, N. (2023). *Financial Prices Prediction of Stock Market using Supervised Machine Learning Models*. VF AST Transactions on Software Engineering. https://www.researchgate.net/publication/371043240.

Sun, J., Li, H., & Huang, Q. H. (2014). Predicting financial distress and corporate failure: A review from the state-of-the-art definitions, modeling, sampling, and featuring approaches. *Knowledge-Based Systems, 57*, 41–56. https://doi.org/10.1016/j.knosys.2013.12.006.

Lessmann, S., Baesens, B., Seow, H. V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research, 247*(1), 124–136. https://doi.org/10.1016/j.ejor.2015.05.030.

Infomineo. (2024). *Machine Learning Models vs. Statistical Models*. Recuperado de https://infomineo.com/data-analytics/machine-learning-models-vs-statistical-models/.

Stefanini Group. (2023). *Machine Learning Models for Precise Predictive Analytics*. Recuperado de https://stefanini.com/en/insights/news/machine-learning-models-for-precise-predictive-analytics.

Majka, M. (2024). *The Impact of Machine Learning on Financial Modeling*. LinkedIn. Recuperado de https://www.linkedin.com/pulse/impact-machine-learning-financial-modeling-marcin-majka-gzacf.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). *A survey of methods for explaining black box models*. ACM Computing Surveys, 51(5), 93:1–93:42. Recuperado de https://www.researchgate.net/publication/322976218_A_Survey_of_Methods_for_Explaining_Black_Box_Models.

Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness and Machine Learning: Limitations and Opportunities*. https://fairmlbook.org.

Data, S. B. (2023, 18 octubre). Machine Learning: Seleccion Métricas de clasificación - sitiobigdata.com. sitiobigdata.com. https://sitiobigdata.com/2019/01/19/machine-learning-metrica-clasificacion-parte-3/.

Chen, J. (2022, December 28). *15 financial ratios every investor should use*. Investopedia. https://www.investopedia.com/articles/stocks/06/ratios.asp

Louppe, G. (2014). *Understanding Random Forests: From Theory to Practice* (Doctoral dissertation, Université de Liège). arXiv:1407.7502. https://arxiv.org/abs/1407.7502

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81–106. https://doi.org/10.1007/BF00116251

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3), 660–674. https://doi.org/10.1109/21.97458

Kashyap, P. (2020). *A comprehensive guide to Gradient Boosting and Regression in Machine Learning: Step by Step*. Medium. https://medium.com/@piyushkashyap045/a-comprehensive-guide-to-gradient-boosting-and-regression-in-machine-learning-step-by-step-faa17fbd0e2c (traducción propia).

AprendeIA. (2022). *Regresión logística: el algoritmo para clasificar eventos binarios*. https://aprendeia.com/algoritmo-regresion-logistica-machine-learning-teoria/ (traducción propia).

7. ANEXO:

7.1 ANÁLISIS EXPLORATORIO DEL DATA SET:

```
import os
import pandas as pd
import numpy as np

# Get the current working directory
current_directory = os.getcwd()
print(f"Current working directory: {current_directory}")

# Construct the file path relative to the current directory
file_path = os.path.join(current_directory,
'snp500_companies_description 2.xlsx')

# Check if the file exists
if not os.path.exists(file_path):
    file_path = input("Please enter the correct path to the Excel
file: ")

# Load the Excel file
datos_df = pd.read_excel(file_path, sheet_name='Datos')

# Function to clean numeric columns
def clean_numeric_column(column, is_percentage=False):
    """
    Limpia una columna:
    - Convierte porcentajes (%).
    - Maneja sufijos como B (billion) y M (million).
    """
    column_cleaned = column.replace("-", np.nan)  # Replace "-"
with NaN
    column_cleaned = column_cleaned.str.replace("%", "",
regex=False)  # Remove percentages
    column_cleaned = column_cleaned.str.replace("B", "e9",
regex=False)  # Convert "B" to e9
    column_cleaned = column_cleaned.str.replace("M", "e6",
regex=False)  # Convert "M" to e6
    column_cleaned = pd.to_numeric(column_cleaned, errors='coerce',
downcast='float')  # Convert to float
    if is_percentage:
        column_cleaned = column_cleaned / 100  # Divide by 100 for
percentages
    return column_cleaned

# Select relevant columns
selected_columns = [
```

```
    'Company', 'Market capitalization', 'Revenue (ttm)', 'Cash per
share (mrq)',
    'Current stock price', 'Price-to-Earnings (ttm)', 'Price-to-
Sales (ttm)',
    'Price-to-Book (mrq)', 'Total Debt to Equity (mrq)',
    , 'Return on Assets (ttm)',
    'Return on Equity (ttm)', 'Operating Margin (ttm)',
    'Net Profit Margin (ttm)', 'Beta', 'Performance (Year)',
    'Performance (Year To Date)','Quick Ratio (mrq)','Dividend
yield (annual)'
]
cleaned_data = datos_df[selected_columns]

# Apply cleaning to problematic columns
problematic_columns = [
    'Revenue (ttm)','Market capitalization' ,'Cash per share
(mrq)', 'Price-to-Earnings (ttm)',
    'Price-to-Book (mrq)', 'Total Debt to Equity (mrq)', 'Return on
Assets (ttm)',
    'Return on Equity (ttm)', 'Operating Margin (ttm)',
    'Net Profit Margin (ttm)', 'Beta', 'Performance (Year)',
    'Performance (Year To Date)','Quick Ratio (mrq)','Dividend
yield (annual)'
]

for column in problematic_columns:
    cleaned_data[column] =
clean_numeric_column(cleaned_data[column], is_percentage=('%' in
column))

# Impute missing values with the median
for column in cleaned_data.columns[1:]:  # Exclude 'Company'
    if cleaned_data[column].isnull().sum() > 0:
        cleaned_data[column].fillna(cleaned_data[column].median(),
inplace=True)

# Save the cleaned dataset to a new Excel file
output_path = os.path.join(current_directory,
'snp500_cleaned_data.xlsx')
cleaned_data.to_excel(output_path, index=False)

print(f"Limpieza completada. El archivo limpio ha sido guardado
como: {output_path}")
```

```
import pandas as pd
import matplotlib.pyplot as plt
```

GRÁFICO 2: MATRIZ DE CORRELACIÓN

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Cargar los datos desde el archivo Excel
file_path = "snp500_cleaned_data.xlsx"  # Este es el cleaned hecho
(del código de arriba, cambiarlo si se hacen modificaciones)
data = pd.read_excel(file_path, sheet_name="Sheet1")

# 1. Estadísticas descriptivas
descriptive_stats = data.describe()
print("Estadísticas descriptivas:")
print(descriptive_stats)

# 2. Matriz de correlación
correlation_matrix = data.corr(numeric_only=True)  # Usar solo
columnas numéricas

# Visualizar la matriz de correlación
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f",
cmap="coolwarm", cbar=True)
plt.title("Matriz de Correlación del S&P 500")
plt.show()

# 3. Visualización de relaciones
# Ejemplo: Relación entre capitalización de mercado y precio actual
de la acción
plt.figure(figsize=(10, 6))
sns.scatterplot(data=data, x="Market capitalization", y="Current
stock price", alpha=0.7, edgecolor="w")
plt.title("Relación entre Capitalización de Mercado y Precio Actual
de la Acción")
plt.xlabel("Capitalización de Mercado (Billion $)")
plt.ylabel("Precio Actual de la Acción ($)")
plt.xscale("log")  # Escala logarítmica para mejor visualización
plt.show()
```

GRÁFICO 3:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Cargar los datos desde tu archivo Excel
file_path = "snp500_cleaned_data.xlsx"  # Cambia esta ruta si el
archivo tiene otro nombre o ubicación
```

```
data = pd.read_excel(file_path, sheet_name="Sheet1")

# Asegurarnos de que la columna "Market capitalization" sea
numérica
data["Market capitalization"] = pd.to_numeric(data["Market
capitalization"], errors="coerce")

# Seleccionar las 20 empresas con mayor capitalización de mercado
top_20_market_cap = data.nlargest(20, "Market
capitalization")[["Company", "Market capitalization"]]

# Crear el gráfico de barras
plt.figure(figsize=(14, 8))
sns.barplot(data=top_20_market_cap, x="Market capitalization",
y="Company", palette="viridis", orient='h')
plt.title("Top 20 Empresas por Capitalización de Mercado")
plt.xlabel("Capitalización de Mercado (Billion $)")
plt.ylabel("Empresa")
plt.show()
```

GRÁFICO 4:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Cargar los datos desde tu archivo Excel
file_path = "snp500_cleaned_data.xlsx"  # Cambia esta ruta si el
archivo tiene otro nombre o ubicación
data = pd.read_excel(file_path, sheet_name="Sheet1")

# Supongamos que tienes una columna llamada "Sector" en tu dataset
# Si no la tienes, puedes añadirla manualmente para las empresas
top 20.
# Aquí hay un ejemplo ficticio de sectores para estas empresas:
sector_mapping = {
    "AAPL": "Tecnología",
    "MSFT": "Tecnología",
    "GOOG": "Tecnología",
    "GOOGL": "Tecnología",
    "AMZN": "Consumo Discrecional",
    "BRK-B": "Finanzas",
    "NVDA": "Tecnología",
    "TSLA": "Consumo Discrecional",
    "META": "Tecnología",
    "V": "Finanzas",
    "XOM": "Energía",
    "UNH": "Salud",
    "JPM": "Finanzas",
```

```
    "JNJ": "Salud",
    "WMT": "Consumo Discrecional",
    "MA": "Finanzas",
    "PG": "Consumo Básico",
    "CVX": "Energía",
    "LLY": "Salud",
    "HD": "Consumo Discrecional",
}

# Añadir la columna de sector al dataset
data["Sector"] = data["Company"].map(sector_mapping)

# Seleccionar las 20 empresas con mayor capitalización de mercado
top_20_market_cap = data.nlargest(20, "Market
capitalization")[["Company", "Market capitalization", "Sector"]]

# Crear el gráfico de barras coloreando por sector
plt.figure(figsize=(14, 8))
sns.barplot(
    data=top_20_market_cap,
    x="Market capitalization",
    y="Company",
    hue="Sector",
    dodge=False,  # Mantener las barras agrupadas
    palette="Set2"
)
plt.title("Top 20 Empresas por Capitalización de Mercado
(Coloreadas por Sector)")
plt.xlabel("Capitalización de Mercado (Billion $)")
plt.ylabel("Empresa")
plt.legend(title="Sector", bbox_to_anchor=(1.05, 1), loc='upper
left')
plt.tight_layout()  # Ajustar el diseño para que la leyenda no
corte el gráfico
plt.show()
```

GRÁFICO 5:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Cargar los datos desde tu archivo Excel
file_path = "snp500_cleaned_data.xlsx"  # Cambia esta ruta si el
archivo tiene otro nombre o ubicación
final_data = pd.read_excel(file_path, sheet_name="Sheet1")

# Convertir la columna 'Beta' a tipo numérico, ignorando errores
```

```
final_data["Beta"] = pd.to_numeric(final_data["Beta"],
errors="coerce")

# Resumen estadístico de Beta
beta_stats = final_data["Beta"].describe()

# Visualización de distribución de Beta
plt.figure(figsize=(10, 6))
sns.histplot(data=final_data, x="Beta", bins=30, kde=True,
color='purple', alpha=0.7)
plt.title("Distribución de Beta (Riesgo de las Empresas)")
plt.xlabel("Beta")
plt.ylabel("Frecuencia")
plt.axvline(x=1, color='red', linestyle='--', label='Beta = 1
(Mercado Promedio)')
plt.legend()
plt.show()

# Identificar empresas con mayor y menor riesgo
high_risk_companies = final_data[final_data["Beta"] >
1.5][["Company", "Beta", "Market capitalization"]]
low_risk_companies = final_data[final_data["Beta"] <
0.8][["Company", "Beta", "Market capitalization"]]

# Mostrar empresas con alto riesgo
print("Empresas con Mayor Riesgo (Beta > 1.5):")
print(high_risk_companies)

# Mostrar empresas con bajo riesgo
print("Empresas con Menor Riesgo (Beta < 0.8):")
print(low_risk_companies)
```

GRÁFICO 6:

```
import pandas as pd
import matplotlib.pyplot as plt

# Cargar los datos desde tu archivo Excel
file_path = "snp500_cleaned_data.xlsx"  # Cambia esta ruta si es
necesario
final_data = pd.read_excel(file_path, sheet_name="Sheet1")

# Convertir la columna 'Beta' a tipo numérico, ignorando errores
final_data["Beta"] = pd.to_numeric(final_data["Beta"],
errors="coerce")

# Clasificar empresas según el nivel de Beta
final_data["Risk Level"] = pd.cut(
```

```
    final_data["Beta"],
    bins=[0, 0.8, 1.2, float("inf")],
    labels=["Bajo Riesgo (Beta < 0.8)", "Riesgo Promedio (0.8 ≤
Beta ≤ 1.2)", "Alto Riesgo (Beta > 1.2)"]
)

# Contar el número de empresas en cada categoría
risk_distribution = final_data["Risk Level"].value_counts()

# Crear el gráfico de tarta
plt.figure(figsize=(8, 8))
plt.pie(
    risk_distribution,
    labels=risk_distribution.index,
    autopct="%1.1f%%",
    startangle=90,
    colors=["#DDCC77","#CC6677" ,"#88CCEE"]  # Colores
personalizados #88CCEE" (azul) "#DDCC77" (amarillo) "#CC6677"
(rojo)
)
plt.title("Distribución de Empresas por Nivel de Riesgo (Beta)")
plt.show()
```

GRÁFICO 7:

```
# Asegurarnos de que la columna 'Performance (Year To Date)' sea
numérica
final_data["Performance (Year To Date)"] =
pd.to_numeric(final_data["Performance (Year To Date)"],
errors="coerce")

# Top 10 empresas con mejor desempeño Year-to-Date
best_performance = final_data.nlargest(10, "Performance (Year To
Date)")[["Company", "Performance (Year To Date)", "Market
capitalization"]]

# Top 10 empresas con peor desempeño Year-to-Date
worst_performance = final_data.nsmallest(10, "Performance (Year To
Date)")[["Company", "Performance (Year To Date)", "Market
capitalization"]]

# Determinar los límites comunes del eje x
x_min = min(worst_performance["Performance (Year To Date)"].min(),
0)  # Valor mínimo
x_max = max(best_performance["Performance (Year To Date)"].max(),
0)   # Valor máximo
# Crear paletas personalizadas (de más fuerte a más suave)
best_colors = sns.color_palette("Greens", n_colors=10)[::-1]  #
Invertir para que el color más fuerte esté arriba
```

```
worst_colors = sns.color_palette("Reds", n_colors=10)[::-1]

# Gráfico de las mejores y peores empresas por desempeño Year-to-
Date
plt.figure(figsize=(14, 8))

# Mejor desempeño
plt.subplot(1, 2, 1)
sns.barplot(data=best_performance, x="Performance (Year To Date)",
y="Company", palette="Greens", orient='h')
plt.title("Top 10 Empresas con Mejor Desempeño YTD")
plt.xlabel("Desempeño Year-to-Date (%)")
plt.ylabel("Empresa")
plt.xlim(x_min, x_max)  # Usar los mismos límites

# Peor desempeño
plt.subplot(1, 2, 2)
sns.barplot(data=worst_performance, x="Performance (Year To Date)",
y="Company", palette="Reds", orient='h')
plt.title("Top 10 Empresas con Peor Desempeño YTD")
plt.xlabel("Desempeño Year-to-Date (%)")
plt.ylabel("Empresa")
plt.xlim(x_min, x_max)  # Usar los mismos límites

plt.tight_layout()
plt.show()
```

GRÁFICO 8:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Cargar el archivo Excel
file_path = "snp500_cleaned_data.xlsx"
final_data = pd.read_excel(file_path, sheet_name="Sheet1")

# Crear violin plots para 'Market capitalization' y 'Revenue (ttm)'
plt.figure(figsize=(14, 6))

# Violin plot de 'Market capitalization'
plt.subplot(1, 2, 1)
sns.violinplot(data=final_data, y="Market capitalization",
color="lightblue", inner="box")
plt.title("Violin Plot de Market Capitalization")
plt.ylabel("Market Capitalization (Billion $)")
plt.xlabel("")
```

```
# Violin plot de 'Revenue (ttm)'
plt.subplot(1, 2, 2)
sns.violinplot(data=final_data, y="Revenue (ttm)",
color="lightgreen", inner="box")
plt.title("Violin Plot de Revenue (ttm)")
plt.ylabel("Revenue (Billion $)")
plt.xlabel("")

plt.tight_layout()
plt.show()
```

7.2 CREACIÓN DE LOS MODELOS DE CLASIFICACIÓN DE ML:

```
import pandas as pd

# Función para calcular la rentabilidad futura
def calculate_future_return(historical_data):
    # Obtener precios iniciales y finales por empresa
    close_prices =
historical_data.groupby('Name')['close'].agg(['first',
'last']).reset_index()
    # Calcular el retorno porcentual
    close_prices['return_future'] = ((close_prices['last'] -
close_prices['first']) / close_prices['first']) * 100
    return close_prices

# Función para clasificar empresas como rentables o no rentables
def classify_companies(close_prices, q1=0.30, q3=0.70):
    # Calcular percentiles
    lower_threshold = close_prices['return_future'].quantile(q1)
    upper_threshold = close_prices['return_future'].quantile(q3)
    # Crear columna target
    close_prices['target'] = close_prices['return_future'].apply(
        lambda x: 1 if x > upper_threshold else (0 if x <
lower_threshold else None)
    )
    # Filtrar valores intermedios
    return close_prices.dropna(subset=['target'])

# Función para unir datos financieros con clasificaciones
def merge_datasets(financial_data, close_prices):
    # Unir los datasets en la columna 'Name' y eliminar duplicados
    merged_data = pd.merge(financial_data, close_prices[['Name',
'target']], left_on='Company', right_on='Name', how='inner')
    # Eliminar columnas redundantes
    merged_data = merged_data.drop(columns=['Name'])
    return merged_data
```

```
# Cargar los datos
financial_data_path = "snp500_cleaned_data.xlsx"  # Ajusta esta
ruta si es necesario
historical_data_path = "all_stocks_5yr.xlsx"  # Ajusta esta ruta si
es necesario

financial_data = pd.read_excel(financial_data_path)
historical_data = pd.read_excel(historical_data_path)

# Calcular rentabilidad futura
close_prices = calculate_future_return(historical_data)

# Clasificar empresas
close_prices = classify_companies(close_prices)

# Unir con los datos financieros
merged_data = merge_datasets(financial_data, close_prices)

# Guardar el dataset final
output_path = "merged_data_for_modeling.xlsx"
merged_data.to_excel(output_path, index=False)

print(f"Archivo final con target guardado en: {output_path}")
```

```
# Revisar la distribución de las clases
print("Distribución de Clases:")
print(merged_data['target'].value_counts(normalize=True))

# Visualización de la distribución
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 4))
merged_data['target'].value_counts().plot(kind='bar',
color=['skyblue', 'salmon'], alpha=0.8)
plt.title('Distribución de Clases (Rentable vs No Rentable)',
fontsize=14)
plt.xlabel('Clases', fontsize=12)
plt.ylabel('Cantidad de Empresas', fontsize=12)
plt.xticks(ticks=[0, 1], labels=['No Rentable', 'Rentable'],
rotation=0)
plt.show()
```

```
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Identificar columnas numéricas (excluyendo la columna objetivo)
```

```
numeric_columns = merged_data.select_dtypes(include=['float64',
'int64']).drop(columns=['target']).columns

# Inicializar el escalador
scaler = StandardScaler()

# Aplicar la normalización solo a las columnas numéricas
merged_data[numeric_columns] =
scaler.fit_transform(merged_data[numeric_columns])

# Verificar las transformaciones
print("Variables normalizadas:\n",
merged_data[numeric_columns].head())

# Identificar columnas categóricas
categorical_columns =
merged_data.select_dtypes(include=['object']).columns
print("Columnas categóricas:\n", categorical_columns)

# Aplicar One-Hot Encoding a las columnas categóricas
merged_data = pd.get_dummies(merged_data,
columns=categorical_columns, drop_first=True)

# Verificar el dataset final después de la normalización y
codificación
print("Dimensiones del dataset final:", merged_data.shape)
print("Primeras filas del dataset final:\n", merged_data.head())
```

MODELO MACHINE LEARNING – RANDOM FOREST

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score,
confusion_matrix
from sklearn.model_selection import train_test_split

# Separar características (X) y variable objetivo (y)
X = merged_data.drop(columns=['target'])
y = merged_data['target']

# Dividir el dataset en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42, stratify=y)

# Modelo Random Forest
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)

# Predicciones con Random Forest
```

```
rf_predictions = rf_model.predict(X_test)

# Evaluación de Random Forest
print("Random Forest:")
print("Accuracy:", accuracy_score(y_test, rf_predictions))
print("Confusion Matrix:\n", confusion_matrix(y_test,
rf_predictions))
print("Classification Report:\n", classification_report(y_test,
rf_predictions))
```

RANDOM FOREST – CON HIPERARÁMETROS

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report, accuracy_score,
confusion_matrix

# Definir los hiperparámetros a ajustar
param_grid = {
    'n_estimators': [100, 200, 300],           # Número de árboles
    'max_depth': [None, 10, 20, 30],          # Profundidad máxima
de los árboles
    'min_samples_split': [2, 5, 10],          # Muestras mínimas
para dividir un nodo
    'min_samples_leaf': [1, 2, 4],            # Muestras mínimas en
una hoja
    'class_weight': [None, 'balanced']        # Balancear clases
automáticamente
}

# Crear el modelo base de Random Forest
rf_model = RandomForestClassifier(random_state=42)

# Configurar GridSearchCV
grid_search = GridSearchCV(
    estimator=rf_model,
    param_grid=param_grid,
    cv=5,                         # Validación cruzada de 5
particiones
    scoring='accuracy',           # Métrica de evaluación
    n_jobs=-1,                    # Usar todos los procesadores
disponibles
    verbose=2                     # Mostrar detalles del progreso
)

# Entrenar el modelo con búsqueda de hiperparámetros
grid_search.fit(X_train, y_train)

# Imprimir los mejores parámetros encontrados
```

```
print("Mejores parámetros encontrados:", grid_search.best_params_)

# Usar el mejor modelo encontrado para hacer predicciones
best_rf_model = grid_search.best_estimator_
rf_predictions = best_rf_model.predict(X_test)

# Evaluación del modelo ajustado
print("\nRandom Forest ajustado:")
print("Accuracy:", accuracy_score(y_test, rf_predictions))
print("Confusion Matrix:\n", confusion_matrix(y_test,
rf_predictions))
print("Classification Report:\n", classification_report(y_test,
rf_predictions))

# Añadir las predicciones al conjunto de prueba
X_test_with_predictions = X_test.copy()
X_test_with_predictions['Actual'] = y_test
X_test_with_predictions['Predicted'] = rf_predictions

# Mostrar empresas clasificadas como rentables (1) y no rentables
(0)
rentables =
X_test_with_predictions[X_test_with_predictions['Predicted'] == 1]
no_rentables =
X_test_with_predictions[X_test_with_predictions['Predicted'] == 0]

print("\nEmpresas clasificadas como rentables (1):")
print(rentables)

print("\nEmpresas clasificadas como no rentables (0):")
print(no_rentables)
```

GRÁFICO RANDOM FOREST – CLASIFCACIÓN DE EMPRESAS

```
import matplotlib.pyplot as plt
import seaborn as sns

# Añadir el nombre de las empresas al DataFrame
visualization_data = X_test_with_predictions.copy()
visualization_data['Name'] = X_test.index  # Asegúrate de que el
índice sea el nombre de las empresas

# Seleccionar las dos características principales
feature_importances = best_rf_model.feature_importances_
important_features = pd.DataFrame({
    'Feature': X_train.columns,
    'Importance': feature_importances
}).sort_values(by='Importance', ascending=False)
```

```
top_features = important_features.head(2)['Feature'].values
print("Características más importantes para el modelo:\n",
top_features)

# Crear un gráfico de dispersión con etiquetas de empresas
plt.figure(figsize=(14, 8))

scatter = sns.scatterplot(
    data=visualization_data,
    x=top_features[0],
    y=top_features[1],
    hue='Predicted',
    style='Actual',
    palette='Set1',
    s=100
)

# Añadir etiquetas con los nombres de las empresas
for line in range(visualization_data.shape[0]):
    plt.text(
        x=visualization_data[top_features[0]].iloc[line],
        y=visualization_data[top_features[1]].iloc[line],
        s=visualization_data['Name'].iloc[line],
        fontsize=8
    )

# Configuración del gráfico
plt.title("Clasificación de Empresas: Rentables vs No Rentables
(Con Nombres)")
plt.xlabel(top_features[0])
plt.ylabel(top_features[1])
plt.legend(title="Predicción", bbox_to_anchor=(1.05, 1), loc='upper
left')
plt.grid(True)
plt.tight_layout()

# Mostrar el gráfico
plt.show()
```

MATRIZ DE CONFUSIÓN & CURVA ROC – RANDOM FOREST

```
from sklearn.metrics import confusion_matrix, roc_auc_score,
roc_curve, auc
import matplotlib.pyplot as plt
import seaborn as sns

# 1. Matriz de Confusión
```

```
conf_matrix = confusion_matrix(y_test, rf_predictions)

# Visualizar la matriz de confusión con un heatmap
plt.figure(figsize=(6, 5))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
xticklabels=['No Rentable (0)', 'Rentable (1)'], yticklabels=['No
Rentable (0)', 'Rentable (1)'])
plt.title("Matriz de Confusión")
plt.xlabel("Predicción")
plt.ylabel("Actual")
plt.show()

# 2. AUC y Curva ROC
# Calcular las probabilidades de las clases para AUC-ROC
rf_probabilities = best_rf_model.predict_proba(X_test)[:, 1]  #
Probabilidades para la clase positiva (1)

# Calcular el AUC-ROC
roc_auc = roc_auc_score(y_test, rf_probabilities)
print(f"AUC-ROC: {roc_auc:.2f}")

# Calcular la curva ROC
fpr, tpr, thresholds = roc_curve(y_test, rf_probabilities)

# Visualizar la curva ROC
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='blue', label=f'ROC Curve (AUC =
{roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='gray', linestyle='--')  # Línea de
referencia (clasificador aleatorio)
plt.title("Curva ROC")
plt.xlabel("Tasa de Falsos Positivos (FPR)")
plt.ylabel("Tasa de Verdaderos Positivos (TPR)")
plt.legend(loc="lower right")
plt.grid(True)
plt.show()
```

MODELO DE MACHINE LEARNING – ARBOL DE DECISIÓN

```
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
import matplotlib.pyplot as plt

# Entrenar el modelo (puedes ajustar max_depth para mejorar la
legibilidad)
dt_model = DecisionTreeClassifier(random_state=42, max_depth=4)  #
Limita la profundidad si quieres más claridad
```

```
dt_model.fit(X_train, y_train)

# Ajustar tamaño según profundidad
plt.figure(figsize=(24, 12))  # Puedes aumentar si el árbol es muy
grande
plot_tree(
    dt_model,
    feature_names=X_train.columns,
    class_names=['No Rentable', 'Rentable'],
    filled=True,
    rounded=True,
    fontsize=12
)
plt.title("Árbol de Clasificación", fontsize=16)
plt.tight_layout()  # Asegura que no se recorte
plt.show()

# Evaluación
dt_predictions = dt_model.predict(X_test)
print("Accuracy del Árbol de Clasificación:",
accuracy_score(y_test, dt_predictions))
print("\nMatriz de Confusión:\n", confusion_matrix(y_test,
dt_predictions))
print("\nReporte de Clasificación:\n",
classification_report(y_test, dt_predictions))
```

ÁRBOL DE DECISIÓN – CON HIPERPARÁMETROS

```
from sklearn.model_selection import GridSearchCV

# Definir los hiperparámetros a ajustar
param_grid = {
    'max_depth': [5, 10, 20, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'criterion': ['gini', 'entropy']
}

# Configurar GridSearchCV para el Árbol de Clasificación
grid_search_dt = GridSearchCV(
    estimator=DecisionTreeClassifier(random_state=42),
    param_grid=param_grid,
    cv=5,
    scoring='accuracy',
    verbose=2,
    n_jobs=-1
)

# Entrenar el modelo con búsqueda de hiperparámetros
```

```
grid_search_dt.fit(X_train, y_train)

# Mejor modelo encontrado
best_dt_model = grid_search_dt.best_estimator_

# Evaluar el modelo ajustado
dt_predictions_adjusted = best_dt_model.predict(X_test)
print("Accuracy del Árbol Ajustado:", accuracy_score(y_test,
dt_predictions_adjusted))
print("\nMejores Hiperparámetros:", grid_search_dt.best_params_)
print("\nReporte de Clasificación Ajustado:\n",
classification_report(y_test, dt_predictions_adjusted))
```

MATRIZ DE CONFUSIÓN & CURVA ROC – ÁRBOL DE DECISIÓN

```
from sklearn.metrics import confusion_matrix, roc_auc_score,
roc_curve, classification_report
import matplotlib.pyplot as plt
import seaborn as sns

# 1. Matriz de Confusión
conf_matrix = confusion_matrix(y_test, dt_predictions_adjusted)

# Visualizar la matriz de confusión
plt.figure(figsize=(6, 5))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
xticklabels=['No Rentable (0)', 'Rentable (1)'], yticklabels=['No
Rentable (0)', 'Rentable (1)'])
plt.title("Matriz de Confusión - Árbol de Decisión")
plt.xlabel("Predicción")
plt.ylabel("Actual")
plt.show()

# 2. AUC y Curva ROC
# Obtener probabilidades para calcular el AUC
dt_probabilities = best_dt_model.predict_proba(X_test)[:, 1]  #
Probabilidades para la clase positiva (1)

# Calcular el AUC
roc_auc = roc_auc_score(y_test, dt_probabilities)
print(f"AUC-ROC del Árbol de Decisión: {roc_auc:.2f}")

# Calcular la curva ROC
fpr, tpr, thresholds = roc_curve(y_test, dt_probabilities)

# Graficar la curva ROC
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='blue', label=f'ROC Curve (AUC =
{roc_auc:.2f})')
```

```
plt.plot([0, 1], [0, 1], color='gray', linestyle='--')  # Línea de
referencia
plt.title("Curva ROC - Árbol de Decisión")
plt.xlabel("Tasa de Falsos Positivos (FPR)")
plt.ylabel("Tasa de Verdaderos Positivos (TPR)")
plt.legend(loc="lower right")
plt.grid(True)
plt.show()
```

MODELO RANDOM FOREST – GRADIENT BOOSTING CON HIPERPARÁMETROS

```
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import classification_report, accuracy_score,
confusion_matrix, roc_auc_score, roc_curve
import matplotlib.pyplot as plt
import seaborn as sns

# Entrenar el modelo de Gradient Boosting
gb_model = GradientBoostingClassifier(random_state=42)
gb_model.fit(X_train, y_train)

# Realizar predicciones
gb_predictions = gb_model.predict(X_test)
gb_probabilities = gb_model.predict_proba(X_test)[:, 1]
from sklearn.model_selection import GridSearchCV

# Definir los hiperparámetros a ajustar
param_grid = {
    'n_estimators': [100, 200, 300],         # Número de árboles
    'learning_rate': [0.01, 0.05, 0.1],     # Tasa de aprendizaje
    'max_depth': [3, 5, 7],                 # Profundidad máxima de
los árboles
    'min_samples_split': [2, 5, 10],        # Muestras mínimas para
dividir un nodo
    'min_samples_leaf': [1, 2, 4]           # Muestras mínimas en
una hoja
}

# Configurar GridSearchCV
grid_search_gb = GridSearchCV(
    estimator=GradientBoostingClassifier(random_state=42),
    param_grid=param_grid,
    cv=5,
    scoring='accuracy',
    verbose=2,
    n_jobs=-1
)
```

```
# Entrenar el modelo con búsqueda de hiperparámetros
grid_search_gb.fit(X_train, y_train)

# Mejor modelo encontrado
best_gb_model = grid_search_gb.best_estimator_

# Evaluar el modelo ajustado
gb_predictions_adjusted = best_gb_model.predict(X_test)
gb_probabilities_adjusted = best_gb_model.predict_proba(X_test)[:,
1]
roc_auc_adjusted = roc_auc_score(y_test, gb_probabilities_adjusted)

print("Accuracy del Gradient Boosting Ajustado:",
accuracy_score(y_test, gb_predictions_adjusted))
print("\nMejores Hiperparámetros:", grid_search_gb.best_params_)
print("\nReporte de Clasificación Ajustado:\n",
classification_report(y_test, gb_predictions_adjusted))
print(f"AUC-ROC del Gradient Boosting Ajustado:
{roc_auc_adjusted:.2f}")
```

MATRIZ DE CONFUSIÓN & CURVA ROC – GRADIENT BOOSTING

```
from sklearn.metrics import confusion_matrix, roc_auc_score,
roc_curve, classification_report
import matplotlib.pyplot as plt
import seaborn as sns

# 1. Matriz de Confusión
# Obtener la matriz de confusión
conf_matrix = confusion_matrix(y_test, gb_predictions)

# Visualizar la matriz de confusión como heatmap
plt.figure(figsize=(6, 5))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
xticklabels=['No Rentable (0)', 'Rentable (1)'], yticklabels=['No
Rentable (0)', 'Rentable (1)'])
plt.title("Matriz de Confusión - Gradient Boosting")
plt.xlabel("Predicción")
plt.ylabel("Actual")
plt.show()

# 2. Curva ROC y AUC
# Calcular el AUC
roc_auc = roc_auc_score(y_test, gb_probabilities)

# Calcular los valores de la curva ROC
fpr, tpr, thresholds = roc_curve(y_test, gb_probabilities)
```

```
# Visualizar la curva ROC
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='blue', label=f'ROC Curve (AUC =
{roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='gray', linestyle='--')  # Línea de
referencia (clasificación aleatoria)
plt.title("Curva ROC - Gradient Boosting")
plt.xlabel("Tasa de Falsos Positivos (FPR)")
plt.ylabel("Tasa de Verdaderos Positivos (TPR)")
plt.legend(loc="lower right")
plt.grid(True)
plt.show()

# Imprimir el AUC-ROC
print(f"AUC-ROC del Gradient Boosting: {roc_auc:.2f}")

# 3. Reporte de Clasificación
print("\nReporte de Clasificación - Gradient Boosting:\n")
print(classification_report(y_test, gb_predictions))
```

MODELO LOGIT REGRESSION CON HIPERPARÁMETROS:

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score,
confusion_matrix, roc_auc_score, roc_curve
import matplotlib.pyplot as plt
import seaborn as sns

# Crear el modelo de Logistic Regression
logit_model = LogisticRegression(random_state=42, max_iter=1000)
logit_model.fit(X_train, y_train)

# Realizar predicciones
logit_predictions = logit_model.predict(X_test)
logit_probabilities = logit_model.predict_proba(X_test)[:, 1]  #
Probabilidades para la clase 1
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV

# Definir el grid de hiperparámetros
param_grid = {
    'C': [0.01, 0.1, 1, 10, 100],                # Regularización
    'penalty': ['l1', 'l2'],                     # Tipo de
regularización
    'solver': ['liblinear', 'saga'],             # Solvers
compatibles
    'class_weight': [None, 'balanced']           # Balanceo de
clases
}
```

```
# Configurar GridSearchCV
grid_search = GridSearchCV(
    LogisticRegression(random_state=42, max_iter=1000),
    param_grid,
    cv=5,  # Validación cruzada
    scoring='accuracy',
    n_jobs=-1,
    verbose=2
)

# Entrenar con búsqueda de hiperparámetros
grid_search.fit(X_train, y_train)

# Obtener los mejores hiperparámetros
best_params = grid_search.best_params_
print("Mejores hiperparámetros:", best_params)
```

MATRIZ DE CONFUSIÓN & CURVA AUC-ROC – LOGIT REGRESSION:

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, roc_curve, auc

# 1. Matriz de Confusión
# Obtener la matriz de confusión
conf_matrix = confusion_matrix(y_test, logit_predictions)

# Visualizar la matriz de confusión como heatmap
plt.figure(figsize=(6, 5))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
xticklabels=['No Rentable (0)', 'Rentable (1)'], yticklabels=['No
Rentable (0)', 'Rentable (1)'])
plt.title("Matriz de Confusión - Logistic Regression Ajustado")
plt.xlabel("Predicción")
plt.ylabel("Actual")
plt.show()

# 2. Curva ROC
# Calcular valores de la curva ROC
fpr, tpr, thresholds = roc_curve(y_test, logit_probabilities)
roc_auc = auc(fpr, tpr)

# Visualizar la curva ROC
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='blue', label=f'ROC Curve (AUC =
{roc_auc:.2f})')
```

```
plt.plot([0, 1], [0, 1], color='gray', linestyle='--')  # Línea de
referencia
plt.title("Curva ROC - Logistic Regression Ajustado")
plt.xlabel("Tasa de Falsos Positivos (FPR)")
plt.ylabel("Tasa de Verdaderos Positivos (TPR)")
plt.legend(loc="lower right")
plt.grid(True)
plt.show()

# Imprimir el valor de AUC-ROC
print(f"AUC-ROC de Logistic Regression Ajustado: {roc_auc:.2f}")
```

MODELO RED NEURONAL:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import classification_report, roc_auc_score,
roc_curve, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Construcción del modelo de red neuronal
nn_model = Sequential([
    Dense(64, activation='relu', input_dim=X_train.shape[1]),  #
Primera capa oculta con 64 neuronas
    Dropout(0.3),                                               #
Dropout para prevenir sobreajuste
    Dense(32, activation='relu'),                              #
Segunda capa oculta con 32 neuronas
    Dropout(0.3),
    Dense(1, activation='sigmoid')                             #
Capa de salida (sigmoide para clasificación binaria)
])

# Compilación del modelo
nn_model.compile(
    optimizer=Adam(learning_rate=0.001),                       #
Optimizador Adam con tasa de aprendizaje
    loss='binary_crossentropy',                               #
Función de pérdida para clasificación binaria
    metrics=['accuracy']                                      #
Métrica de evaluación
)

# Entrenamiento del modelo
history = nn_model.fit(
    X_train, y_train,
```

```
    validation_split=0.2,                                    #
División para validación
    epochs=50,                                               #
Número de épocas
    batch_size=16,                                           #
Tamaño del batch
    verbose=1                                                #
Nivel de detalle
)
```

MATRIZ DE CONFUSIÓN & CURVA AUC-ROC – RED NEURONAL:

```
# Evaluación del modelo
nn_predictions = (nn_model.predict(X_test) >
0.5).astype(int).flatten()
nn_probabilities = nn_model.predict(X_test).flatten()

print("Reporte de Clasificación - Red Neuronal:\n")
print(classification_report(y_test, nn_predictions))

# Matriz de Confusión
conf_matrix = confusion_matrix(y_test, nn_predictions)
plt.figure(figsize=(6, 5))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
xticklabels=['No Rentable (0)', 'Rentable (1)'], yticklabels=['No
Rentable (0)', 'Rentable (1)'])
plt.title("Matriz de Confusión - Red Neuronal")
plt.xlabel("Predicción")
plt.ylabel("Actual")
plt.show()

# Curva ROC y AUC
roc_auc = roc_auc_score(y_test, nn_probabilities)
fpr, tpr, thresholds = roc_curve(y_test, nn_probabilities)
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='blue', label=f'ROC Curve (AUC =
{roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='gray', linestyle='--')  # Línea de
referencia
plt.title("Curva ROC - Red Neuronal")
plt.xlabel("Tasa de Falsos Positivos (FPR)")
plt.ylabel("Tasa de Verdaderos Positivos (TPR)")
plt.legend(loc="lower right")
plt.grid(True)
plt.show()

# Imprimir el AUC-ROC
print(f"AUC-ROC de Red Neuronal: {roc_auc:.2f}")
```