



**Universidad
Europea**

UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

MÁSTER UNIVERSITARIO EN ANALISIS DE DATOS MASIVOS (BIG DATA)

TRABAJO FIN DE MÁSTER

Desarrollo de un Sistema Automatizado de *Paper Trading* basado en modelos de *Machine Learning*

FERNANDO ÁLVARO PASTOR

Dirigido por

OSCAR MARBAN GALLEG

CURSO 2024-2025

Fernando Álvaro Pastor

TÍTULO: «Desarrollo de un Sistema Automatizado de *Paper Trading* basado en modelos de Machine Learning»

AUTOR: Fernando Álvaro Pastor

TITULACIÓN: Master Universitario en Análisis de Datos Masivos (Big Data)

DIRECTOR/ES DEL PROYECTO: Óscar Marban Gallego

FECHA: OCTUBRE de 2025

Fernando Álvaro Pastor

RESUMEN

Este proyecto desarrolla un sistema automatizado de *paper trading* basado en modelos de *Machine Learning* aplicados al mercado de criptomonedas. El objetivo principal ha sido diseñar una solución capaz de identificar de forma autónoma señales de compra y venta sobre el par BTC-USD, con el fin de maximizar la rentabilidad simulada mediante técnicas avanzadas de análisis de datos.

El sistema se ha dividido en dos bloques principales: el primero comprende la descarga y procesamiento de datos históricos, la generación optimizada de etiquetas a partir de ventanas temporales futuras y la construcción de modelos predictivos mediante algoritmos de clasificación (XGBoost, LightGBM, Random Forest y CatBoost), integrados en un *ensemble* ponderado. Estos modelos han sido validados mediante *walk-forward cross-validation*, y su rendimiento ha sido evaluado a través de un *backtest* con métricas como CAGR, Sharpe ratio, drawdown máximo o *win rate*.

En el segundo bloque se ha desarrollado un sistema completamente autónomo de *paper trading* en tiempo real, capaz de operar de forma continua sobre datos descargados cada 4 horas, aplicando los modelos previamente entrenados, gestionando capital, aplicando *stop loss*, *take profit* y *trailing stop*, y generando informes y gráficos de evolución.

Entre las conclusiones más relevantes destacan la capacidad del sistema para adaptarse a distintas condiciones de mercado sin intervención humana, la utilidad de la validación temporal progresiva para evitar *overfitting*, y el potencial de este enfoque como base para futuras estrategias de inversión algorítmica real.

Palabras clave:

- Paper trading
- Machine learning
- Trading algorítmico
- Validación walk-forward
- Ensemble learning
- BTC-USD

Fernando Álvaro Pastor

ABSTRACT

This project presents the development of an automated paper trading system based on Machine Learning models applied to the cryptocurrency market. The main objective was to design a solution capable of autonomously identifying buy and sell signals on the BTC-USD pair, aiming to maximize simulated returns through advanced data analysis techniques.

The system is structured in two main blocks. The first includes the acquisition and processing of historical data, the optimized generation of labels using forward-looking windows, and the construction of predictive models using classification algorithms (XGBoost, LightGBM, Random Forest, and CatBoost), combined in a weighted ensemble. These models were validated using walk-forward cross-validation and evaluated through a comprehensive backtest using metrics such as CAGR, Sharpe ratio, maximum drawdown, and win rate.

The second block consists of a fully autonomous real-time paper trading system, capable of operating continuously with 4-hour candlestick data. It applies the trained models, manages capital, executes stop loss, take profit, and trailing stop rules, and generates updated reports and performance graphs.

Key conclusions include the system's ability to adapt to different market conditions without human intervention, the effectiveness of progressive temporal validation in reducing overfitting, and the potential of this approach as a foundation for future real-world algorithmic trading strategies.

Keywords:

- Paper trading
- Machine learning
- Algorithmic trading
- Walk-forward validation
- Ensemble learning
- BTC-USD

Fernando Álvaro Pastor

AGRADECIMIENTOS

Gracias de corazón a mi familia y amigos por estar siempre ahí, por los ánimos, la paciencia y por recordarme que todo esfuerzo tiene su recompensa. Este proyecto también es un pedacito vuestro.

Fernando Álvaro Pastor

Cita - frase célebre / Dedicatoria

“El éxito no es la clave de la felicidad. La felicidad es la clave del éxito. Si amas lo que haces, tendrás éxito.”

— Albert Schweitzer

TABLA RESUMEN

	DATOS
Nombre y apellidos:	Fernando Álvaro Pastor
Título del proyecto:	Desarrollo de un Sistema Automatizado de Paper Trading basado en modelos de Machine Learning
Directores del proyecto:	Oscar Marban Gallego
El proyecto se ha realizado en colaboración de una empresa o a petición de una empresa:	NO
El proyecto ha implementado un producto: (esta entrada se puede marcar junto a la siguiente)	NO
El proyecto ha consistido en el desarrollo de una investigación o innovación: (esta entrada se puede marcar junto a la anterior)	SI
Objetivo general del proyecto:	Desarrollar e implementar un sistema automatizado basado en machine learning para optimizar la toma de decisiones en trading del par BTC-USD utilizando datos históricos y en tiempo real.

INDICE

RESUMEN	3
ABSTRACT	4
TABLA RESUMEN	7
Capítulo 1. RESUMEN DEL PROYECTO	12
1.1 Contexto y justificación.....	12
1.2 Planteamiento del problema	12
1.3 Objetivos del proyecto.....	13
1.4 Resultados obtenidos	13
1.5 Estructura de la memoria	14
Capítulo 2. ANTECEDENTES / ESTADO DEL ARTE	15
2.1 Estado del arte	15
2.2 Contexto y justificación.....	17
2.3 Planteamiento del problema	17
Capítulo 3. OBJETIVOS	19
3.1 Objetivos generales	19
3.2 Objetivos específicos	19
3.3 Beneficios del proyecto	20
Capítulo 4. DESARROLLO DEL PROYECTO	21
4.1 Planificación del proyecto.....	21
4.2 Descripción de la solución, metodologías y herramientas empleadas.....	22
4.2.1 Adquisición y preparación de datos	22
4.2.2 Ingeniería de características.....	23
4.2.3 Generación automática de etiquetas (labels)	26
4.2.4 Selección y limpieza del conjunto de entrenamiento	27
4.2.5 Selección de características.....	27
4.2.6 Escalado y normalización	28
4.2.7 Entrenamiento de modelos base	29
4.2.8 Validación walk-forward y ensamblado de modelos	29
4.2.9 Entrenamiento del modelo final	29

4.2.10	Optimización de parámetros de trading	30
4.2.11	Backtesting y evaluación final	30
4.2.12	Desarrollo del sistema de <i>paper trading</i>	31
4.3	Recursos requeridos	31
4.3.1	Herramientas tecnológicas empleadas	32
4.4	Presupuesto	32
4.5	Viabilidad	33
4.5.1	Viabilidad económica	33
4.5.2	Sostenibilidad a futuro	34
4.6	Resultados del proyecto	34
4.6.1	Resultados técnicos obtenidos.....	34
4.6.2	Evaluación del sistema	52
4.6.3	Cambios respecto al planteamiento inicial	52
Capítulo 5.	DISCUSIÓN.....	53
5.1	Utilidad de la metodología aplicada	53
5.2	Cambios y adaptaciones respecto a la planificación	53
5.3	Limitaciones del proyecto.....	54
5.4	Impacto del resultado.....	54
Capítulo 6.	CONCLUSIONES	55
6.1	Conclusiones del trabajo.....	55
6.2	Conclusiones personales.....	55
Capítulo 7.	FUTURAS LÍNEAS DE TRABAJO	57
7.1	Inclusión de gestión dinámica del capital	57
7.2	Modelado del slippage y la latencia	57
7.3	Integración de múltiples estrategias o activos	57
7.4	Exploración de arquitecturas avanzadas de modelos	57
7.5	Mejora de la interfaz de seguimiento.....	58
7.6	Despliegue en entorno real con capital reducido.....	58
7.7	Incorporación de técnicas de aprendizaje en línea	58
Capítulo 8.	REFERENCIAS.....	59

Índice de Figuras

Figura 4.1: Diagrama de Gantt. Cronograma del proyecto	22
Figura 4.2: Cuenta de etiquetas generadas	36
Figura 4.3: Mejores parámetros de etiquetas con Optuna.....	36
Figura 4.4: Indicadores de compra finales	37
Figura 4.5: Indicadores de venta finales	38
Figura 4.6: Diagrama de parámetros de compra con SHAP	38
Figura 4.7: Diagrama de parámetros de venta con SHAP	39
Figura 4.8: Parámetros de Gridseacrh para entrenamiento del modelo de compra.....	40
Figura 4.9: Parámetros de Gridseacrh para entrenamiento del modelo de venta.....	40
Figura 4.10: Pesos por modelo en función de F1-score.....	42
Figura 4.11: Matrices de confusión de compra y venta.....	44
Figura 4.12: Histograma de probabilidad real.....	45
Figura 4.13: Metricas finales del modelo de compra.....	45
Figura 4.14: Métricas finales del modelo de venta	46
Figura 4.15: Tresholds finales de compra y venta.....	46
Figura 4.16: Parametros finales de trading optimizados con Optuna	47
Figura 4.17: Evolución del capital en el backtest	48
Figura 4.18: Distribución de profit por operaciones	48
Figura 4.19: Distribución de señales de compra y venta en una gráfica.....	49
Figura 4.20: Métricas finales del backtest.....	49
Figura 4.21: Gráfico del capital del paper trading.....	51

Índice de Tablas

Tabla 4.1: Cronograma del proyecto.....	21
Tabla 4.2: Herramientas tecnológicas empleadas	32
Tabla 4.3: Presupuesto.....	33
Tabla 4.4: Datos brutos obtenidos de yfinance	35
Tabla 4.5: Datos tratados obtenidos de yfinance	35
Tabla 4.6: Outputs generados por Optuna	35
Tabla 4.7: Fragmento de btc_dataset_with_labels.csv.....	36
Tabla 4.8: Outputs de SHAP values, permutación y correlación.....	37
Tabla 4.9: Outputs de escalado, eliminación de outliers y balanceo.....	40
Tabla 4.10: Outputs de entrenamiento con XGBoost, Random Forest, LightGBM y Catboost ..	41
Tabla 4.11: F1-score por modelo	41
Tabla 4.12: Outputs del walk-forward	42
Tabla 4.13: Resultados de crossvalidated walk forward ensemble	43
Tabla 4.14: Resumen del crossvalidated walk forward ensemble	43
Tabla 4.15: Resultados del walk forward ensemble	44
Tabla 4.16: Outputs de evaluación de métricas de entrenamiento	44
Tabla 4.17: Outputs de Optuna para Trading	46
Tabla 4.18: Outputs de Backtest y plots.....	47
Tabla 4.19: Fragmento del historial de trades en el backtest.....	47
Tabla 4.20: Outputs del Paper Trading	50
Tabla 4.21: Historia del capital de paper trading	50
Tabla 4.22: Tradelog del paper trading	51

Capítulo 1. RESUMEN DEL PROYECTO

1.1 Contexto y justificación

El crecimiento exponencial de los mercados financieros descentralizados y, en particular, del mercado de criptomonedas, ha impulsado el interés por desarrollar sistemas de inversión automatizados basados en técnicas avanzadas de análisis de datos. Estos mercados, caracterizados por su alta volatilidad, disponibilidad continua (24/7) y complejidad no lineal, representan un entorno ideal para la aplicación de modelos de *Machine Learning*, capaces de detectar patrones en grandes volúmenes de datos históricos con el objetivo de anticipar comportamientos futuros.

En este contexto, el presente trabajo se centra en el diseño y desarrollo de un sistema automatizado de *paper trading*, orientado a simular operaciones en tiempo real sobre el par BTC-USD. El enfoque adoptado busca integrar procesos de ingeniería de características, entrenamiento de modelos de clasificación, optimización de umbrales y evaluación mediante *backtesting*, con una fase posterior de despliegue del sistema en una máquina virtual que opera de forma autónoma, sin supervisión humana.

El proyecto surge como respuesta a la creciente demanda de soluciones cuantitativas que permitan validar estrategias de inversión antes de su aplicación real, reduciendo así el riesgo y los costes asociados. Además, se justifica por el interés técnico y académico de combinar *Machine Learning*, validación temporal robusta y técnicas de simulación de trading en un sistema unificado y funcional.

1.2 Planteamiento del problema

En el ámbito del trading algorítmico, uno de los principales retos consiste en desarrollar sistemas que sean capaces de detectar señales de entrada y salida con un nivel suficiente de fiabilidad, especialmente en mercados volátiles como el de criptomonedas. Aunque existe una amplia literatura sobre el uso de modelos predictivos en finanzas, muchas propuestas adolecen de problemas como el *overfitting*, el uso inadecuado de validación temporal, o la falta de evaluación en condiciones realistas mediante simulaciones continuas.

La pregunta motriz que plantea este proyecto es:
¿Es posible construir un sistema automatizado de trading simulado capaz de generar señales fiables de compra y venta, utilizando modelos de Machine Learning entrenados y validados de forma robusta, que opere de forma autónoma en tiempo real?

El proyecto no ha sido desarrollado para resolver un problema empresarial concreto, sino como investigación aplicada en el ámbito técnico-científico, con el objetivo de integrar metodologías de *Machine Learning*, técnicas de validación propias de series temporales y mecanismos de

Fernando Álvaro Pastor

ejecución de estrategias de forma automatizada. Asimismo, el trabajo incorpora elementos de innovación, como la optimización conjunta de las etiquetas de entrada y salida, la validación *walk-forward* cruzada, el ensamblado ponderado de modelos y la implementación de un sistema funcional de *paper trading* continuo que replica condiciones reales de mercado.

1.3 Objetivos del proyecto

El objetivo principal de este proyecto es desarrollar un sistema automatizado de *paper trading* que utilice modelos de *Machine Learning* para predecir señales de compra y venta sobre el par BTC-USD, validando dichas señales mediante técnicas adecuadas para series temporales y evaluando su rentabilidad potencial en un entorno simulado.

Este objetivo general se descompone en una serie de objetivos específicos, que incluyen:

- La recopilación y tratamiento de datos históricos del mercado de criptomonedas.
- La generación automática y optimizada de etiquetas de entrada y salida basadas en ventanas futuras.
- El entrenamiento de modelos de clasificación robustos, como XGBoost, LightGBM, Random Forest y CatBoost, integrados en un sistema *ensemble* ponderado.
- La validación de los modelos mediante técnicas como *walk-forward cross-validation* para reducir el riesgo de *overfitting* (Hyndman & Athanasopoulos, 2018).
- El diseño de un sistema autónomo que ejecute operaciones simuladas en tiempo real, con control de capital y gestión de riesgos mediante *take profit*, *stop loss* y *trailing stop*.

Los resultados y el diseño del sistema permiten no solo simular estrategias de inversión, sino también sentar las bases para su futura migración a un entorno de trading real o semiautomático, constituyendo una contribución relevante en el ámbito del trading cuantitativo (Chan, 2017).

1.4 Resultados obtenidos

El proyecto ha dado como resultado un sistema funcional y completo de *paper trading* que opera de forma autónoma sobre el par BTC-USD, generando señales de compra y venta basadas en modelos de *Machine Learning* previamente entrenados y validados.

En la fase de entrenamiento, se generaron etiquetas mediante un proceso de optimización bayesiana con Optuna, lo que permitió definir umbrales de beneficio (*take profit*) y pérdida (*stop loss*) adecuados para cada caso. Posteriormente, se entrenaron cuatro modelos de clasificación independientes (XGBoost, LightGBM, Random Forest y CatBoost) tanto para la predicción de compras como de ventas, utilizando técnicas de ingeniería de características y validación *walk-forward*. Estos modelos se combinaron en un *ensemble* ponderado según su F1-score medio, mejorando la robustez de las predicciones.

Fernando Álvaro Pastor

El sistema de trading simulado fue desplegado en una máquina virtual, desde donde se ejecuta periódicamente cada 4 horas. En cada iteración descarga los datos más recientes, calcula indicadores, aplica los modelos, decide si abrir o cerrar una operación, y actualiza tanto el histórico de capital como el registro de operaciones.

El sistema logró simular operaciones con una tasa de aciertos superior al azar, manteniendo un capital creciente durante el backtest, con una tasa de retorno anual compuesta (CAGR) positiva y un *drawdown* moderado. Además, se generaron gráficos de evolución del capital, distribución de beneficios por operación y visualizaciones de las señales de entrada y salida, los cuales se incluyen en el capítulo de resultados.

1.5 Estructura de la memoria

La presente memoria se organiza en ocho capítulos, cada uno de los cuales aborda una parte específica del desarrollo del proyecto.

- Capítulo 1 – Resumen del proyecto: Presenta una visión general del trabajo, contextualiza el problema, define los objetivos principales y resume los resultados obtenidos.
- Capítulo 2 – Introducción: Profundiza en el contexto técnico y científico del trabajo, así como en los fundamentos que motivan la automatización del paper trading mediante técnicas de Machine Learning.
- Capítulo 3 – Objetivos: Expone de forma estructurada el objetivo general del proyecto, así como los objetivos específicos que orientan el desarrollo de la solución.
- Capítulo 4 – Metodología: Detalla el enfoque adoptado para la implementación del sistema, incluyendo la estrategia experimental, las herramientas utilizadas y las decisiones de diseño más relevantes.
- Capítulo 5 – Implementación: Describe el desarrollo técnico del sistema, dividido en dos bloques: el entrenamiento y validación de modelos predictivos, y la ejecución en tiempo real del paper trading.
- Capítulo 6 – Resultados: Presenta los principales resultados del backtest y de la simulación en tiempo real, junto con gráficos, métricas de rendimiento y análisis de las operaciones generadas.
- Capítulo 7 – Discusión: Analiza e interpreta los resultados obtenidos, reflexionando sobre las limitaciones del sistema y proponiendo mejoras futuras.
- Capítulo 8 – Conclusiones: Resume las aportaciones clave del proyecto, las lecciones aprendidas y el potencial de la solución para su aplicación en entornos reales.

Además, se incluye un apartado de referencias bibliográficas y un anexo con información técnica complementaria, como código, estructuras de datos o configuraciones detalladas.

Capítulo 2. ANTECEDENTES / ESTADO DEL ARTE

2.1 Estado del arte

En el contexto de los mercados financieros altamente volátiles como el de las criptomonedas, surge la necesidad de diseñar sistemas capaces de detectar señales de entrada y salida de forma automática, utilizando enfoques robustos que combinen datos históricos, técnicas de modelado predictivo y simulación de operaciones. La volatilidad extrema de activos como Bitcoin (BTC), unida a la ausencia de fundamentos tradicionales y a su sensibilidad frente a factores externos (noticias, regulación, especulación), ha impulsado la búsqueda de metodologías avanzadas basadas en datos que superen las limitaciones de los enfoques clásicos (Baur, Hong, & Lee, 2018; Corbet, Lucey, Urquhart, & Yarovaya, 2019).

Históricamente, las estrategias de trading se han basado en el análisis técnico, mediante indicadores como medias móviles, niveles de soporte y resistencia o patrones de velas japonesas. Estos métodos, aunque ampliamente utilizados por su simplicidad, presentan limitaciones notables: dependen de supuestos rígidos, no se adaptan bien a entornos dinámicos y carecen de capacidad para capturar relaciones no lineales entre múltiples variables (Park & Irwin, 2007; Atsalakis & Valavanis, 2009). La evidencia empírica ha mostrado que el poder predictivo del análisis técnico puro es limitado y tiende a deteriorarse en mercados con alta eficiencia (Fama, 1970).

Ante estas limitaciones, en los últimos años el uso de técnicas de Machine Learning (ML) se ha consolidado como una alternativa eficaz. Algoritmos como Random Forest, XGBoost o LightGBM han demostrado ser capaces de manejar grandes volúmenes de datos, tolerar ruido y capturar interacciones complejas entre variables, ofreciendo resultados prometedores en predicción de retornos y clasificación de señales (Fischer & Krauss, 2018; Gu, Kelly, & Xiu, 2020). Estos modelos destacan frente a regresiones lineales tradicionales y métodos econométricos, ya que no imponen relaciones funcionales predefinidas y se adaptan a la estructura de los datos.

Un aspecto clave en este campo es la selección de variables. La alta dimensionalidad de los datos financieros (indicadores técnicos, métricas de volatilidad, series de volumen, variables macroeconómicas) puede conducir al sobreajuste (*overfitting*). Métodos de selección basados en importancia de características, como SHAP (Shapley Additive Explanations) o la *Permutation Importance*, han permitido mejorar la interpretabilidad de los modelos y aumentar su robustez (Lundberg & Lee, 2017; Molnar, 2019).

Además, diversos estudios han mostrado que el ensemble learning (técnicas de combinación de modelos) incrementa la estabilidad y precisión de las predicciones, reduciendo la varianza de los resultados frente al uso de un único clasificador (Dietterich, 2000; Zhou, 2012). En entornos como el financiero, donde los datos son inherentemente ruidosos y no estacionarios, el ensamblado de múltiples algoritmos se ha convertido en una de las estrategias más utilizadas para incrementar la robustez de los sistemas predictivos.

Fernando Álvaro Pastor

En cuanto a la validación, la naturaleza temporal de los datos financieros exige metodologías diferentes a la validación cruzada tradicional. El uso de validación walk-forward permite entrenar y evaluar modelos en ventanas móviles de tiempo, replicando las condiciones de producción y reduciendo el riesgo de utilizar información futura en el entrenamiento (Hyndman & Athanasopoulos, 2018; Berrar, 2019). Este enfoque es fundamental para evitar el *look-ahead bias* y evaluar el desempeño del modelo en escenarios realistas de mercado.

De igual forma, la literatura destaca la importancia de utilizar métricas adaptadas al contexto financiero. Más allá de la precisión o el F1-score, medidas como el *Sharpe ratio*, el *Maximum Drawdown* o el *Profit Factor* ofrecen una visión más completa sobre la viabilidad económica de una estrategia (Bailey, Borwein, Lopez de Prado, & Zhu, 2014). De hecho, se ha demostrado que modelos con métricas de clasificación aparentemente sólidas pueden no ser rentables al aplicarse en entornos de trading real si no se consideran factores como costes de transacción y deslizamiento (*slippage*) (De Prado, 2018).

El paper trading se ha consolidado como un paso intermedio esencial entre el desarrollo de modelos y su aplicación real. Permite ejecutar operaciones simuladas en condiciones de mercado en tiempo real sin riesgo económico, facilitando la validación externa del sistema y el análisis de su desempeño bajo diferentes condiciones de mercado (Chan, 2017; Lopez de Prado, 2018). En este sentido, constituye una herramienta crucial para garantizar que los resultados obtenidos en backtesting puedan trasladarse a escenarios operativos.

Por último, el uso de herramientas de optimización automatizada como Optuna o Hyperopt ha revolucionado el ajuste de hiperparámetros. Estas librerías permiten explorar de manera eficiente espacios de búsqueda complejos mediante técnicas como *Bayesian Optimization* o *Tree-structured Parzen Estimator* (Bergstra et al., 2011; Akiba et al., 2019). Su aplicación no solo se limita al ajuste de hiperparámetros de modelos predictivos, sino también a la calibración de reglas de trading, umbrales de decisión y parámetros de gestión de riesgo.

En conjunto, la convergencia de avances en modelado predictivo, validación temporal y optimización automatizada ha dado lugar a un marco idóneo para el desarrollo de sistemas híbridos de trading algorítmico. No obstante, persisten desafíos relevantes: la robustez frente a cambios estructurales en los mercados (regímenes de volatilidad), la necesidad de incorporar costes de transacción en las simulaciones y el riesgo de sobreajuste al pasado, que puede llevar a estrategias ineficaces en condiciones futuras (Lopez de Prado, 2020; Arnott, Harvey, & Markowitz, 2019). Estos retos subrayan la importancia de la investigación continua y el desarrollo de enfoques más sofisticados que integren inteligencia artificial, gestión dinámica del riesgo y evaluación rigurosa de la robustez de los modelos.

2.2 Contexto y justificación

La creciente complejidad de los mercados financieros, y en particular del ecosistema de criptomonedas, ha impulsado el desarrollo de estrategias de inversión automatizadas capaces de adaptarse dinámicamente a las condiciones cambiantes del mercado. A diferencia de los activos tradicionales, las criptomonedas operan de forma continua, sin horarios de cierre, y con una alta volatilidad e imprevisibilidad, lo que convierte su análisis en un reto ideal para técnicas avanzadas de modelado y automatización.

En este contexto, el presente proyecto surge como una iniciativa de investigación aplicada cuyo objetivo es aportar una solución técnica basada en modelos de *Machine Learning*, integrados dentro de un sistema completo de *paper trading*. Este enfoque permite explorar no solo la capacidad predictiva de dichos modelos, sino también su comportamiento operativo en un entorno simulado lo más realista posible, sin riesgo económico directo. El proyecto busca responder a una necesidad real de los desarrolladores de sistemas algorítmicos: validar, ajustar y comprender el comportamiento de sus estrategias antes de implementarlas en entornos de producción o con capital real.

La contribución de este trabajo al campo del *trading* cuantitativo es doble. Por un lado, ofrece una metodología reproducible para la creación de señales mediante la generación optimizada de etiquetas, el uso combinado de múltiples modelos de clasificación, y la validación *walk-forward* cruzada. Por otro, proporciona una arquitectura operativa funcional que ejecuta decisiones en tiempo real en base a modelos entrenados, incorporando gestión de capital, reglas de riesgo y seguimiento del rendimiento, todo ello empaquetado en un sistema autónomo y monitorizable.

Desde una perspectiva técnica, el proyecto aporta conocimiento sobre la integración eficiente de técnicas como *ensemble learning*, optimización bayesiana de umbrales, escalado de datos y ejecución automatizada sobre entornos virtuales. Este tipo de soluciones se alinean con las tendencias actuales en el uso de inteligencia artificial aplicada al sector financiero, ofreciendo una base sólida para investigaciones futuras, aplicaciones profesionales o productos orientados a la inversión algorítmica.

2.3 Planteamiento del problema

El análisis del estado del arte en el ámbito del *trading* algorítmico y el uso de *Machine Learning* en finanzas revela un creciente interés en automatizar la toma de decisiones de inversión. Sin embargo, también pone de manifiesto varias limitaciones técnicas y metodológicas que siguen sin resolverse de forma satisfactoria en gran parte de la literatura y en muchas implementaciones reales.

Por un lado, numerosos trabajos se centran exclusivamente en maximizar la precisión del modelo predictivo sobre conjuntos de datos históricos, sin tener en cuenta las características propias de las series temporales financieras, como la no estacionariedad, la autocorrelación o la

Fernando Álvaro Pastor

aparición de sesgos por el uso de datos futuros (*look-ahead bias*). La validación cruzada aleatoria, muy común en otros campos, no es aplicable en este entorno, lo que ha motivado la aparición de métodos más robustos como la validación *walk-forward*. Sin embargo, su implementación práctica sigue siendo poco frecuente o limitada.

Por otro lado, muchas estrategias basadas en *Machine Learning* carecen de una evaluación realista mediante simulaciones continuas, lo que impide medir su viabilidad operativa en condiciones similares a las del mercado real. Asimismo, es habitual que se empleen etiquetas binarias mal definidas o arbitrarias (por ejemplo, subidas o bajadas mínimas sin contexto temporal), lo que reduce la calidad del aprendizaje supervisado y, en consecuencia, la utilidad de las predicciones generadas.

En base a este análisis, se identifica una necesidad clara: la creación de un sistema que no solo entrene modelos predictivos sólidos, sino que los integre en una arquitectura de *paper trading* autónoma, con generación optimizada de etiquetas, validación temporal adecuada y evaluación completa mediante simulación continua. Esta combinación permite cerrar el ciclo entre entrenamiento, validación y ejecución, cubriendo una carencia frecuente tanto en entornos académicos como en desarrollos no profesionales.

El presente proyecto se plantea, por tanto, como una solución técnica que busca aportar rigor metodológico, reproducibilidad y operatividad a un campo en el que la mayoría de las soluciones se centran únicamente en la predicción, dejando de lado la ejecución práctica y la validación en condiciones reales.

Capítulo 3. OBJETIVOS

3.1 Objetivos generales

El objetivo general del presente trabajo consiste en desarrollar un sistema automatizado de *paper trading* basado en modelos de *Machine Learning*, capaz de detectar señales de compra y venta en el mercado de criptomonedas, y de ejecutar decisiones simuladas en tiempo real mediante una arquitectura funcional y autónoma.

3.2 Objetivos específicos

- **Analizar** las características del mercado de criptomonedas, con énfasis en el comportamiento del par BTC-USD y su idoneidad para estrategias automatizadas de inversión.
- **Recopilar y procesar** datos históricos del activo seleccionado, asegurando su calidad y completitud para el posterior análisis técnico y modelado predictivo.
- **Diseñar y aplicar** técnicas de ingeniería de características para extraer indicadores relevantes del comportamiento del mercado.
- **Generar** etiquetas de compra y venta optimizadas mediante técnicas de búsqueda bayesiana (*Optuna*), en función de ventanas futuras y umbrales de rentabilidad.
- **Entrenar** modelos de clasificación supervisada (XGBoost, LightGBM, Random Forest y CatBoost) para predecir señales de entrada y salida, priorizando la detección fiable de oportunidades.
- **Validar** los modelos mediante técnicas temporales robustas como la validación *walk-forward*, y seleccionar los mejores clasificadores en base a métricas como el F1-score.
- **Combinar** los modelos en un sistema *ensemble* ponderado que integre sus predicciones para mejorar la estabilidad y precisión de las decisiones.
- **Implementar** un sistema de *paper trading* en tiempo real sobre una máquina virtual, que simule decisiones de inversión autónomas cada 4 horas.
- **Gestionar** el capital del sistema mediante reglas de riesgo como *stop loss*, *take profit* y *trailing stop*, integradas en el motor de decisión.
- **Evaluar** el rendimiento del sistema a través de métricas financieras como la tasa de retorno (CAGR), *Sharpe ratio*, *drawdown máximo* y tasa de aciertos.
- **Documentar** el proceso de desarrollo y análisis, incluyendo código, visualizaciones y resultados, para facilitar la reproducibilidad del sistema propuesto.

3.3 Beneficios del proyecto

El proyecto desarrollado aporta beneficios significativos tanto desde el punto de vista técnico como aplicado. En primer lugar, proporciona una solución funcional que automatiza por completo el proceso de detección de señales de compra y venta en un activo financiero, aplicando modelos de *Machine Learning* entrenados de forma rigurosa, validados mediante técnicas adecuadas para series temporales y evaluados en un entorno realista de simulación continua.

Desde una perspectiva metodológica, el sistema consolida buenas prácticas poco habituales en trabajos similares, como la optimización automática de etiquetas, la validación *walk-forward* cruzada o el ensamblado ponderado de clasificadores. Esto facilita el desarrollo de sistemas más robustos y menos propensos a errores comunes como el *overfitting* o la selección de umbrales arbitrarios.

Además, el sistema de *paper trading* implementado permite simular decisiones de inversión en tiempo real sin riesgo económico, lo que representa un entorno seguro y controlado para probar estrategias, evaluar modelos o validar hipótesis antes de invertir capital real. Esta funcionalidad lo convierte en una herramienta de experimentación útil tanto en contextos académicos como profesionales.

En conjunto, el proyecto ofrece un marco de trabajo replicable y ampliable que puede servir de base para futuros desarrollos en trading algorítmico, tanto para usuarios individuales interesados en invertir de forma autónoma como para equipos de desarrollo que busquen automatizar estrategias en mercados volátiles como el de criptomonedas.

Capítulo 4. DESARROLLO DEL PROYECTO

4.1 Planificación del proyecto

El desarrollo del proyecto se ha llevado a cabo de forma estructurada, dividiendo el trabajo en varias fases secuenciales e iterativas. La planificación ha abarcado tanto actividades de investigación como de desarrollo e implementación técnica, adaptándose progresivamente a los resultados obtenidos y a las necesidades detectadas en cada etapa.

A continuación, se resumen las principales actividades realizadas, así como su temporalización estimada:

Nº	Actividad	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7
1	Estudio del estado del arte y revisión de literatura	✓						
2	Análisis de alternativas actuales y definición del enfoque del sistema		✓	✓				
3	Obtención y limpieza de datos históricos (BTC-USD)			✓				
4	Ingeniería de características e implementación de indicadores técnicos			✓				
5	Generación de etiquetas mediante optimización con Optuna y limpieza de indicadores con SHARP + Permutation			✓	✓			
6	Escalado, división temporal y entrenamiento de modelos ML (XGBoost, LightGBM, RF, CatBoost)				✓			
7	Validación walk-forward y ensamblado ponderado de modelos. Backtest				✓			
8	Diseño e implementación del sistema de paper trading					✓		
9	Evaluación del rendimiento del sistema (métricas y visualización)						✓	
10	Redacción y documentación del proyecto						✓	✓

Tabla 4.1: Cronograma del proyecto

Se ha representado visualmente el cronograma mediante un diagrama de Gantt:



Figura 4.1: Diagrama de Gantt. Cronograma del proyecto

4.2 Descripción de la solución, metodologías y herramientas empleadas

Este apartado describe de forma exhaustiva la solución desarrollada, las metodologías aplicadas y las herramientas tecnológicas empleadas para diseñar, entrenar y validar un sistema automatizado de *paper trading* basado en modelos de *machine learning*. El sistema tiene como objetivo detectar señales de compra y venta de Bitcoin (BTC-USD) en un entorno simulado, operando de forma autónoma sobre datos históricos descargados con resolución de 4 horas. Se han implementado técnicas avanzadas de procesamiento, modelado predictivo y validación robusta, integradas en un flujo automatizado que permite la trazabilidad completa del proceso.

4.2.1 Adquisición y preparación de datos

La primera etapa consistió en la recopilación de datos históricos del par BTC-USD mediante la API **yfinance** (Yahoo Finance: <https://pypi.org/project/yfinance/>) [Tsiang, 2023]. Se seleccionó una resolución de **4 horas** durante los dos últimos años. Los datos descargados incluyeron precios de apertura, cierre, máximos, mínimos y volumen.

Posteriormente, se aplicó una limpieza inicial de datos, reemplazando valores nulos y ceros mediante interpolación hacia atrás (*bfill*) y hacia adelante (*ffill*), garantizando así la **continuidad temporal** de la serie. El índice temporal fue convertido explícitamente a tipo datetime y ordenado cronológicamente para evitar desajustes en fases posteriores.

Campos obtenidos

- **Datetime:** marca temporal en intervalos de 4h.
- **Open:** precio de apertura del periodo.
- **High:** precio máximo alcanzado en la vela.
- **Low:** precio mínimo alcanzado en la vela.
- **Close:** precio de cierre del periodo.
- **Volume:** volumen total negociado en BTC.

Análisis Exploratorio (EDA)

- **Valores nulos y ceros:** detectados en <1% de los registros; corregidos mediante interpolación.
- **Distribución de precios:** concentrada en el rango 30.000–120.000 USD, con picos de volatilidad en eventos de mercado.
- **Volumen:** presenta asimetría positiva (colas largas), indicando picos asociados a noticias o movimientos de alta volatilidad.
- **Temporalidad:** el comportamiento exhibe estacionalidad débil (ciclos semanales de menor actividad en fines de semana).

4.2.2 Ingeniería de características

Se desarrolló un conjunto amplio de *features* mediante la extracción de **indicadores técnicos**, que capturan dinámicas del mercado en términos de tendencia, momentum, volatilidad y volumen. Entre los indicadores implementados se encuentran medias móviles exponenciales (EMA20, EMA50, EMA100, EMA200), medias móviles simples (SMA20, SMA50), RSI, ROC, MACD, ATR, Bollinger Bands, OBV, MFI, y constructos derivados como el *volumen relativo*, indicadores de distancia al precio promedio, *slope* de la EMA50, patrones de vela Marubozu, entre otros.

Además, se generaron **lags temporales** de cada indicador (1 y 2 períodos), con el objetivo de introducir información de estados pasados. Se aplicó un shift(1) sobre todos los indicadores no básicos para prevenir el *look-ahead bias*, asegurando que ninguna información del futuro se utilice para predecir el pasado.

- **Simple Moving Average (SMA):**

Calcula el promedio aritmético del precio de cierre en una ventana temporal fija.

$$SMA_t = \frac{1}{n} \sum_{i=0}^{n-1} P_{t-i}$$

Fernando Álvaro Pastor

Donde n es el número de periodos y P el precio de cierre. Se implementaron SMA20 y SMA50, que permiten identificar tendencias de corto y medio plazo.

- **Exponential Moving Average (EMA):**

Variante de la media móvil que da mayor peso a los precios más recientes, respondiendo más rápido a cambios de tendencia.

$$EMA_t = \alpha * P_t + (1 - \alpha) * EMA_{t-1}$$

con $\alpha = \frac{2}{n+1}$. Se implementaron EMA20, EMA50, EMA100 y EMA200 para capturar tendencias en distintos horizontes temporales.

- **Relative Strength Index (RSI)**

Oscilador de momentum que mide la velocidad y el cambio de los movimientos de precios, acotado entre 0 y 100.

$$RSI = 100 - \frac{100}{1 + RS} \quad ; \quad RS = \frac{\text{Media Ganancias}}{\text{Media Pérdidas}}$$

Valores altos (≥ 70) sugieren sobrecompra y bajos (≤ 30) sobreventa, aunque en mercados como criptomonedas tiende a prolongarse en zonas extremas debido a la volatilidad.

- **Rate of Change (ROC)**

Mide el cambio porcentual del precio respecto a n periodos atrás:

$$ROC_t = \frac{P_t - P_{t-n}}{P_{t-n}} * 100$$

Es un oscilador que detecta aceleraciones o desaceleraciones en el mercado.

- **Moving Average Convergence Divergence (MACD)**

Calcula la diferencia entre dos medias móviles exponenciales:

$$MACD_t = EMA_{12}(P_t) - EMA_{26}(P_t)$$

Acompañado de una línea de señal (EMA_9 del MACD) y del histograma. Indica cambios en la dirección de la tendencia y cruces de momentum.

- **Average True Range (ATR)**

Mide la volatilidad del mercado en base al rango de precios. Se define como la media móvil del *True Range*:

$$TR_t = \max(High_t - Low_t, |High_t - Close_{t-1}|, |Low_t - Close_{t-1}|)$$

$$ATR_t = SMA(TR_t, n)$$

Un ATR elevado implica mayor volatilidad e incertidumbre.

- **Bollinger Bands (BB)**

Consisten en una banda superior e inferior a k desviaciones estándar de la media móvil:

$$BB_{sup} = SMA_n + k\sigma ; BB_{inf} = SMA_n - k\sigma$$

Permiten identificar periodos de sobrecompra/sobreventa relativa y la compresión de volatilidad previa a movimientos bruscos.

- **On-Balance Volume (OBV)**

Indicador de volumen acumulado que relaciona el flujo de volumen con la dirección del precio:

$$OBV_t = \begin{cases} OBV_{t-1} + Vol_t & \text{si } Close_t > Close_{t-1} \\ OBV_{t-1} - Vol_t & \text{si } Close_t < Close_{t-1} \\ OBV_{t-1} & \text{si } Close_t = Close_{t-1} \end{cases}$$

Refleja la presión de compra o venta subyacente en el mercado.

- **Money Flow Index (MFI)**

Oscilador basado en precio y volumen que mide la presión de compra y venta. Se calcula el flujo monetario típico:

$$TP_t = \frac{High_t + Low_t + Close_t}{3}$$

$$MF_t = TP_t * Vol_t$$

Se separa en flujo positivo o negativo y se construye el índice:

$$MFI = 100 - \frac{100}{1 + \frac{MF\ Positivo}{MF\ Negativo}}$$

- **Volumen Relativo**

Mide la relación entre el volumen actual y la media de volumen en una ventana temporal:

$$VolRel_t = \frac{Vol_t}{SMA(Vol, n)}$$

Indica si una vela ha tenido un volumen inusualmente alto o bajo.

Fernando Álvaro Pastor

- **Distancia al Precio Promedio**

Feature derivado que mide la desviación relativa del precio respecto a una media móvil (ej. EMA50):

$$Dist_{EMA50} = \frac{Close_t - EMA50_t}{EMA50_t}$$

Sirve para identificar precios extendidos o alejados de su tendencia base.

- **Slope de la EMA50**

Se obtiene como la pendiente de la EMA50 en una ventana móvil, aproximando la dirección de la tendencia:

$$Slope_{EMA50} = \frac{EMA50_t - EMA_{t-n}}{n}$$

Un slope positivo indica tendencia alcista, negativo bajista.

- **Patrones de vela Marubozu**

Velas japonesas sin sombras (o con sombras muy pequeñas), donde apertura y cierre coinciden con el máximo o mínimo.

- **Marubozu alcista:** Open ≈ Low , Close ≈ High.
- **Marubozu bajista:** Open ≈ High, Close ≈ Low.

Reflejan convicción y fuerza en la dirección del movimiento.

- **Lags temporales y prevención de look-ahead bias**

Se generaron retardos (*lags*) de 1 y 2 períodos para todos los indicadores, introduciendo información de estados pasados sin utilizar datos futuros. Adicionalmente, se aplicó un shift(1) en todos los indicadores derivados del precio (no básicos), de modo que las predicciones del modelo no estén influenciadas por información no disponible en el momento real de decisión.

4.2.3 Generación automática de etiquetas (labels)

Para transformar el problema en uno de clasificación binaria, se diseñó un sistema de **etiquetado basado en ventanas temporales futuras**. El objetivo fue etiquetar una vela con `buy_label = 1` si en las siguientes n velas se alcanza una subida porcentual mínima, y con `sell_label = 1` si en las próximas velas se detecta una caída significativa.

Dado que estos parámetros (ventana futura y umbral porcentual) afectan directamente la calidad del modelo, se utilizaron **técnicas de optimización bayesiana** con Optuna para seleccionar los hiperparámetros óptimos (`buy_future_window`, `buy_threshold`, `sell_future_window`, `sell_threshold`) que maximizan el retorno total estimado en una simulación sencilla de operaciones.

Este enfoque permitió generar etiquetas que simulan de forma realista situaciones de entrada y salida rentables, basadas exclusivamente en información futura, pero sin introducir sesgos en el entrenamiento del modelo.

4.2.4 Selección y limpieza del conjunto de entrenamiento

Tras las fases de **extracción de datos, generación de indicadores técnicos, retardos temporales y control de sesgos (look-ahead bias)**, se consolidó un conjunto de datos definitivo compuesto únicamente por las **columnas relevantes para el modelado predictivo**.

Este dataset incluye:

- **Features técnicas:** indicadores de tendencia, momentum, volatilidad y volumen (ej. EMA, SMA, RSI, MACD, ATR, Bollinger Bands, OBV, MFI, volumen relativo, distancias al promedio, slopes, patrones de vela, etc.), junto con sus *lags* de 1 y 2 períodos.
- **Variables objetivo (labels):**
 - `buy_label`: etiqueta binaria que indica si en el horizonte definido la señal de compra habría resultado positiva.
 - `sell_label`: etiqueta binaria que refleja oportunidades de venta bajo las condiciones optimizadas.

Para mejorar la robustez del dataset se aplicó **clipping por percentiles** (ej. [1, 99]) sobre los indicadores numéricos, eliminando la influencia de outliers extremos sin distorsionar la distribución original.

Finalmente, se realizó una **división temporal estricta** de los datos, garantizando que no exista fuga de información hacia el futuro:

- **Entrenamiento (80%):** datos históricos iniciales.
- **Validación (14%):** bloque intermedio usado para selección de hiperparámetros y ajuste de thresholds.
- **Test final (6%):** tramo más reciente, utilizado exclusivamente para evaluar el rendimiento fuera de muestra (*out-of-sample*).

4.2.5 Selección de características

Con el objetivo de reducir la dimensionalidad y seleccionar únicamente las variables más relevantes, se utilizó un enfoque combinado de:

- **Valores SHAP promediados en validación cruzada** (shap.Explainer con TimeSeriesSplit)
- **Importancia por permutación**
- **Filtrado por sets correlacionados** (evitando duplicidad entre indicadores y sus lags altamente correlacionados)

Esta metodología permitió mantener únicamente aquellas variables que aportan mayor valor explicativo, reduciendo la complejidad y evitando el sobreajuste.

4.2.6 Escalado y normalización

La estandarización consiste en transformar las variables de manera que presenten **media cero y desviación estándar unitaria**:

$$z = \frac{x - \mu}{\sigma}$$

donde μ es la media del conjunto de entrenamiento y σ la desviación típica.

La estandarización de variables constituye una etapa fundamental dentro del preprocesamiento de datos, ya que garantiza que todas las características del conjunto de datos se encuentren en una escala comparable. En este proyecto, los indicadores técnicos generados presentaban magnitudes muy diferentes: por ejemplo, el RSI se expresa en un rango acotado entre 0 y 100, mientras que el volumen puede alcanzar valores de millones, el ATR se mueve en centenas y el MACD puede oscilar entre valores positivos y negativos de distinta magnitud. Si estas variables se introducen directamente en los modelos, las que tienen un rango numérico mayor tienden a influir más en el proceso de entrenamiento, incluso cuando no necesariamente son las más relevantes desde un punto de vista predictivo.

Para resolver esta situación, se aplicó el método StandardScaler de scikit-learn, que transforma cada variable de forma que su media sea cero y su desviación típica unitaria. Este ajuste se realizó exclusivamente sobre el conjunto de entrenamiento, con el fin de evitar cualquier filtración de información del futuro (*data leakage*). Posteriormente, la misma transformación se aplicó sobre los conjuntos de validación y de test, manteniendo la coherencia temporal del flujo de datos y garantizando que el modelo se evaluará en condiciones realistas.

Aunque los modelos basados en árboles como XGBoost, LightGBM o CatBoost no requieren estrictamente la estandarización, este paso aporta varias ventajas adicionales. Por un lado, asegura la coherencia en todo el pipeline de datos y facilita la futura integración de algoritmos sensibles a la escala, como SVM, k-NN o regresión logística. Por otro lado, contribuye a estabilizar los procesos de optimización llevados a cabo mediante Optuna, ya que trabajar con variables normalizadas mejora la eficiencia en la búsqueda de hiperparámetros.

En definitiva, la estandarización se aplicó no solo como una medida técnica para homogenizar escalas, sino también como una estrategia de robustez metodológica, evitando sesgos derivados

Fernando Álvaro Pastor

de las diferencias numéricas entre indicadores y asegurando un entrenamiento más equilibrado, coherente y estable a lo largo de todo el proceso.

4.2.7 Entrenamiento de modelos base

Para cada una de las tareas (`buy_label` y `sell_label`), se entrenó un conjunto de modelos utilizando GridSearchCV con validación tipo TimeSeriesSplit y métrica de evaluación centrada en la clase positiva (F1-score, con `pos_label=1`). Se utilizaron como base:

- **XGBoost** (scale_pos_weight adaptado dinámicamente)
- **RandomForestClassifier**
- **LightGBM**
- **CatBoost** (con class_weights dinámicos)

Los modelos fueron entrenados con hiperparámetros optimizados y evaluados en múltiples combinaciones para detectar los conjuntos más robustos.

4.2.8 Validación walk-forward y ensamblado de modelos

Para evaluar el comportamiento del sistema de forma más robusta, se implementó una **validación tipo walk-forward** con ventanas móviles sobre el conjunto de entrenamiento. En cada ciclo se:

- Entrenan los cuatro modelos base sobre una porción del dataset
- Se ensamblan mediante VotingClassifier con votación suave (soft)
- Se optimizan los *thresholds de probabilidad* mediante Optuna, buscando el umbral que maximice el F1-score sobre la subvalidación

Este procedimiento se repitió en múltiples ciclos (`initial_train_size` variable) para simular el comportamiento del sistema en condiciones cambiantes, y se registraron métricas detalladas de F1-score por modelo y por ciclo.

Los resultados obtenidos permitieron calcular de forma automática los **pesos para cada modelo dentro del ensemble**, proporcional al F1-score medio. Esto se aplicó por separado para los modelos de compra y venta, resultando en un sistema ensemble adaptativo y optimizado.

4.2.9 Entrenamiento del modelo final

Con base en los resultados anteriores, se entrenaron dos modelos ensemble finales:

- `final_model_buy`: para predecir señales de compra
- `final_model_sell`: para predecir señales de venta

Fernando Álvaro Pastor

Ambos integran los 4 modelos base con pesos personalizados. Los modelos fueron entrenados sobre el conjunto completo de entrenamiento y validados sobre el conjunto final_test.

Los thresholds definitivos de decisión (probabilidad mínima para emitir una señal) se establecieron como la **media de los thresholds óptimos** obtenidos en la validación walk-forward.

4.2.10 Optimización de parámetros de trading

Una vez entrenado el sistema de predicción, se diseñó una simulación completa de operaciones (*backtesting*) utilizando los modelos y los thresholds reales. Se definió una estrategia que entra en el mercado si se detecta una señal de compra, y sale si:

- Se activa una señal de venta
- Se alcanza un *Take Profit* (TP)
- Se activa un *Stop Loss* (SL)
- Se activa un *Trailing Stop* (TS)

Los parámetros TP, SL y TS fueron **optimizados mediante Optuna** para maximizar el retorno final sobre el capital simulado.

4.2.11 Backtesting y evaluación final

Se ejecutó un *backtest* completo sobre los datos de test no vistos (final_test), aplicando la lógica operativa completa, incluyendo:

- Ejecución realista basada en precios de apertura
- Costes de transacción (comisión simulada)
- Registro de entradas y salidas
- Evolución del capital en el tiempo

Se registraron las siguientes métricas:

- Retorno final
- Ratio de ganancias/pérdidas
- Expectancy, Profit Factor, Payoff Ratio
- Sharpe Ratio y Calmar Ratio
- CAGR (rentabilidad anualizada)
- Max Drawdown
- Duración promedio de operaciones

Además, se generaron gráficos de capital, distribución de beneficios por operación, y señales de entrada/salida sobre el precio.

4.2.12 Desarrollo del sistema de *paper trading*

La fase final del proyecto consistió en la implementación de un sistema **autónomo de trading simulado**, desarrollado en Python y desplegado en una **máquina virtual**. Este sistema:

- Descarga nuevos datos de mercado cada 4 horas.
- Aplica los modelos entrenados para generar predicciones.
- Decide si abrir o cerrar posiciones simuladas.
- Aplica gestión de capital con **take profit, stop loss y trailing stop**.
- Registra en tiempo real el capital acumulado y las operaciones simuladas.

4.3 Recursos requeridos

Para la ejecución del presente proyecto se han requerido los siguientes recursos:

- Lenguaje de programación Python, como base del desarrollo.
- Bibliotecas de ciencia de datos:
 - pandas, numpy para manipulación de datos.
 - matplotlib para generación de gráficos.
 - scikit-learn para preprocesamiento y modelos básicos.
 - xgboost, lightgbm, catboost, randomforest para entrenamiento de modelos.
 - optuna para optimización de hiperparámetros y parámetros de trading.
- Entorno de ejecución Jupyter Notebook, para exploración y validación iterativa del sistema.
- Google Colab y máquina virtual Linux (VPS), para entrenamiento y ejecución continua del bot de paper trading.
- Repositorio local y control de versiones con Git, para organizar las distintas iteraciones del desarrollo.
- API de Yahoo Finance (yfinance), para descarga de datos de mercado en tiempo real.
- Conocimientos técnicos y asesoramiento en machine learning, ingeniería de características, validación en series temporales y estrategias de trading.

4.3.1 Herramientas tecnológicas empleadas

A lo largo del proyecto se han utilizado las siguientes herramientas y bibliotecas:

Herramienta / Tecnología	Descripción
Python	Lenguaje principal para desarrollo y análisis
Scikit-learn	Preprocesamiento, escalado, métricas y selección de variables
XGBoost, LightGBM, CatBoost	Modelos de clasificación utilizados para la detección de señales
Optuna	Optimización bayesiana de etiquetas y thresholds
Matplotlib, Seaborn	Visualización de resultados y métricas
Pandas, NumPy	Manipulación de datos y estructuras numéricas
Yahoo Finance API (yfinance)	Descarga de datos históricos de mercado
Google Colab	Entrenamiento de modelos y validación walk-forward en entorno cloud
Máquina Virtual Linux	Entorno autónomo de ejecución del sistema de <i>paper trading</i>

Tabla 4.2: Herramientas tecnológicas empleadas

4.4 Presupuesto

El presupuesto estimado para la realización del proyecto contempla tanto los recursos materiales como el tiempo invertido. Aunque muchos de los recursos utilizados son de libre acceso o ya disponibles, se ha incluido su valor de mercado para reflejar el coste real de ejecución en condiciones estándar.

A continuación, se muestra un resumen del presupuesto estimado:

Tipo de coste	Valor estimado (€)	Comentarios
Horas de trabajo en el proyecto	3.000 €	Se estiman aproximadamente 150 horas a una tasa estándar de 20 €/hora .
Equipo técnico: ordenador portátil personal	800 €	Valor de mercado de un equipo con prestaciones adecuadas para desarrollo ML.
Equipo técnico: servidor VPS	50 €	Coste aproximado por uso de VPS durante el desarrollo y despliegue.
Software: Python + librerías	0 €	Software libre y gratuito (pandas, scikit-learn, yfinance, matplotlib, etc.).
Software: Google Colab Pro	0 €	Se ha utilizado la versión gratuita sin coste adicional.
Software: Jupyter Notebook	0 €	Libre y gratuito.
Software: Git + VSCode	0 €	Herramientas gratuitas para control de versiones y desarrollo.
Informes / documentación	0 €	No ha sido necesario adquirir estudios o papers de pago.
Material adicional empleado	0 €	No se han requerido sensores ni materiales físicos adicionales.

Tabla 4.3: Presupuesto

Total estimado: 3.850 €

Este presupuesto incluye tanto el tiempo del autor como los recursos utilizados. No se han contabilizado colaboraciones externas, dado que el desarrollo ha sido realizado íntegramente de forma individual.

4.5 Viabilidad

Aunque este proyecto se ha desarrollado con fines académicos, su planteamiento, ejecución y resultados permiten considerar su viabilidad técnica y económica como solución aplicable a entornos reales.

4.5.1 Viabilidad económica

El coste total estimado del desarrollo es moderado (alrededor de **3.850 €**), incluyendo el tiempo de trabajo y los recursos necesarios. Teniendo en cuenta que el sistema desarrollado es

reutilizable, escalable y modificable para otros activos financieros, su potencial de retorno económico es elevado. De hecho, al tratarse de un sistema capaz de detectar oportunidades de compra y venta en un mercado real, su aplicación en un entorno de *trading* con capital real podría generar beneficios que compensen con creces su coste inicial.

Además, al haberse empleado software libre y entornos gratuitos como Google Colab o servidores VPS de bajo coste, los **costes operativos** son mínimos. El sistema puede mantenerse en funcionamiento sin requerir inversiones adicionales significativas.

4.5.2 Sostenibilidad a futuro

Desde el punto de vista de sostenibilidad tecnológica, el sistema ha sido diseñado con una arquitectura modular y mantenible. Esto permite:

- Sustituir modelos fácilmente por versiones mejoradas.
- Adaptarlo a otros activos o intervalos temporales.
- Incluir nuevas reglas de gestión de capital o indicadores técnicos.
- Integrar estrategias adicionales, como portafolios múltiples o gestión de riesgo dinámica.

Además, al haber documentado y automatizado el flujo completo, su mantenimiento a largo plazo no requiere intervención manual constante, más allá de actualizaciones ocasionales en los modelos o parámetros. Este enfoque garantiza la **longevidad operativa** del sistema sin generar dependencia crítica de recursos externos.

4.6 Resultados del proyecto

El desarrollo del presente proyecto ha culminado con éxito en la implementación de un sistema automatizado de *paper trading* capaz de tomar decisiones de compra y venta sobre el activo **BTC-USD** en intervalos de **4 horas**, basado en modelos de clasificación entrenados con datos históricos y validados rigurosamente mediante técnicas avanzadas.

4.6.1 Resultados técnicos obtenidos

Conforme a los objetivos planteados, se han logrado los siguientes hitos concretos:

- **Diseño y construcción de un pipeline de entrenamiento completo**, desde la descarga de datos hasta el entrenamiento de modelos y el backtest:

Fernando Álvaro Pastor

- Descarga de datos de BTC 4h

Datetime	Open	High	Low	Close	Volume
2023-07-01 00:00	30350.10	30410.55	30120.80	30250.25	1254.32
2023-07-01 04:00	30250.25	30320.00	30080.75	30150.15	1120.85
2023-07-01 08:00	NaN	30210.90	30000.00	NaN	0.00
2023-07-01 12:00	30150.15	30270.80	30090.20	30195.50	1350.20

Tabla 4.4: Datos brutos obtenidos de yfinance

Datetime	Open	High	Low	Close	Volume
2023-07-01 00:00	30350.10	30410.55	30120.80	30250.25	1254.32
2023-07-01 04:00	30250.25	30320.00	30080.75	30150.15	1120.85
2023-07-01 08:00	30200.20	30210.90	30045.50	30170.00	1235.52
2023-07-01 12:00	30150.15	30270.80	30090.20	30195.50	1350.20

Tabla 4.5: Datos tratados obtenidos de yfinance

- Generación de etiquetas optimizadas con Optuna

Archivo	Tipo	Descripción
labels_count.txt	TXT	Distribución de clases generadas.
best_label_params.txt	TXT	Parámetros óptimos de future_window y threshold vía Optuna.
btc_dataset_with_labels.csv	CSV	Dataset original Enriquecido con indicadores técnicos y etiquetas.

Tabla 4.6: Outputs generados por Optuna

Labels_count.txt

```
Buy labels: 534
Sell labels: 534
```

Figura 4.2: Cuenta de etiquetas generadas

Best_label_params.txt

```
📌 Mejores parámetros encontrados por Optuna:

buy_future_window: 1
buy_threshold: 1.0004656329255395
sell_future_window: 1
sell_threshold: 0.6747855660748295

💰 Retorno total estimado: 1209.28%
```

Figura 4.3: Mejores parámetros de etiquetas con Optuna

Fragmento de btc_dataset_with_labels.csv

Datetime	Close	Open	Volume	ema20	ema50	buy_label	sell_label
2023-08-09 20:00:00	29573.01	29394.86	79337472.0	29436.97	29301.75	0	0
2023-08-10 00:00:00	29553.69	29563.97	79337472.0	29449.99	29313.91	1	0
2023-08-10 04:00:00	29473.88	29556.18	79337472.0	29459.92	29324.59	0	0
2023-08-10 08:00:00	29470.03	29482.44	241165312.0	29461.25	29331.21	0	0
2023-08-10 12:00:00	29435.03	29473.04	241165312.0	29462.09	29337.33	0	1

Tabla 4.7: Fragmento de btc_dataset_with_labels.csv

- Selección de características con **SHAP values**, permutación y correlación.

Archivo	Tipo	Descripción
features_buy_final.txt	TXT	Variables seleccionadas para el modelo de compra.
features_sell_final.txt	TXT	Variables seleccionadas para el modelo de venta.
shap_summary_buy.png	PNG	Gráfico SHAP de importancia de variables (compra).
shap_summary_sell.png	PNG	Gráfico SHAP de importancia de variables (venta).

Tabla 4.8: Outputs de SHAP values, permutación y correlación

Features_buy_final.txt

```
mfi_lag2
vol_ratio_lag2
close_to_ema50_lag2
roc_lag2
close_to_ema20
range_ratio_lag1
bb_width
vol_ratio
close_to_ema20_lag1
range_ratio_lag2
close_to_ema50
roc_lag1
momentum10
range_ratio
mfi_lag1
momentum10_lag1
close_to_ema20_lag2
mfi
roc
momentum10_lag2
close_to_ema50_lag1
vol_ratio_lag1
```

Figura 4.4: Indicadores de compra finales

Features_sell_final.txt

```
mfi_lag2
close_to_ema50_lag2
roc_lag2
close_to_ema20
range_ratio_lag1
vol_ratio
close_to_ema20_lag1
range_ratio_lag2
close_to_ema50
roc_lag1
close_to_ema200
range_ratio
mfi_lag1
close_to_ema200_lag1
close_to_ema200_lag2
close_to_ema20_lag2
mfi
roc
close_to_ema50_lag1
```

Figura 4.5: Indicadores de venta finales

shap_summary_buy.png

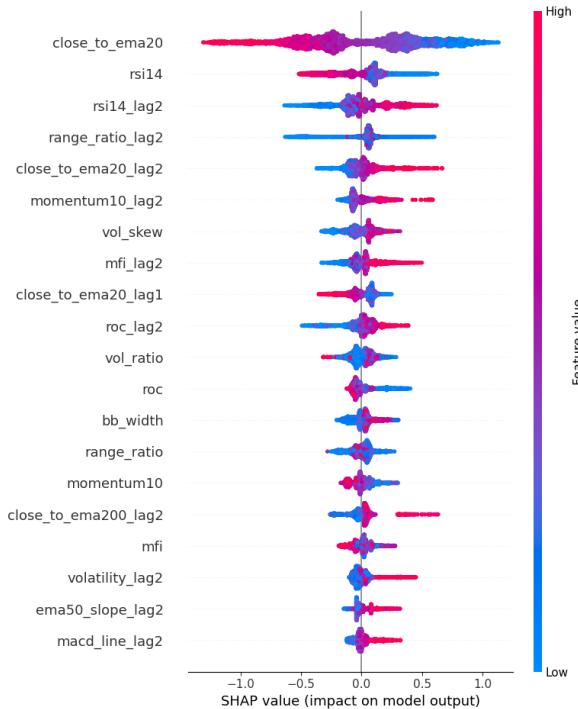


Figura 4.6: Diagrama de parámetros de compra con SHAP

shap_summary_sell.png

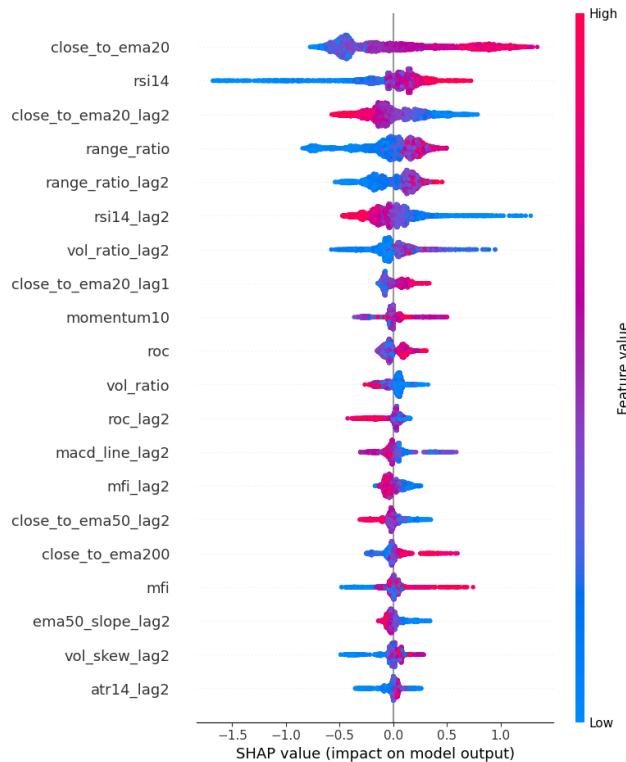


Figura 4.7: Diagrama de parámetros de venta con SHAP

- Preprocesamiento robusto con **escalado**, Gridsearch, eliminación de *outliers* y balanceo.

Archivo	Tipo	Descripción
scaler_buy.pkl	PKL	Escalador aplicado al modelo de compra.
scaler_sell.pkl	PKL	Escalador aplicado al modelo de venta.
train_data_sell_cleaned.csv	CSV	Dataset limpio y final para entrenamiento del modelo de venta.
train_data_buy_cleaned.csv	CSV	Dataset limpio y final para entrenamiento del modelo de compra.
best_xgb_params_buy.txt	TXT	Búsqueda de mejores parámetros de compra para entrenamiento con GridSearch
best_xgb_params_sell.txt	TXT	Búsqueda de mejores parámetros de venta para entrenamiento con GridSearch

Tabla 4.9: Outputs de escalado, eliminación de outliers y balanceo

best_xgb_params_buy.txt

```
Mejores parámetros modelo BUY (precision_score):
colsample_bytree: 0.8
learning_rate: 0.01
max_depth: 3
n_estimators: 100
scale_pos_weight: 7.51
subsample: 1.0
```

Figura 4.8: Parámetros de Gridseacrh para entrenamiento del modelo de compra

best_xgb_params_sell.txt

```
Mejores parámetros modelo SELL (precision_score):
colsample_bytree: 1.0
learning_rate: 0.05
max_depth: 4
n_estimators: 50
scale_pos_weight: 7.51
subsample: 0.8
```

Figura 4.9: Parámetros de Gridseacrh para entrenamiento del modelo de venta

Fernando Álvaro Pastor

- Entrenamiento de modelos **XGBoost, LightGBM, Random Forest y CatBoost.**

Archivo	Tipo	Descripción
ensemble_weights.txt	TXT	Pesos asignados a cada modelo en el ensemble.
f1_scores_individuales.csv	CSV	F1-score por modelo y ventana.
model_buy.pkl	PKL	Modelo entrenado para detectar señales de compra.
model_sell.pkl	PKL	Modelo entrenado para detectar señales de venta.

Tabla 4.10: Outputs de entrenamiento con XGBoost, Random Forest, LightGBM y Catboost

F1_scores_individuales.csv

Modelo	Tipo	F1-score medio
--------	------	----------------

XGBoost (xgb)	Buy	0.3618
---------------	-----	--------

Random Forest (rf)	Buy	0.0172
--------------------	-----	--------

LightGBM (lgb)	Buy	0.3339
----------------	-----	--------

CatBoost (cb)	Buy	0.4019
---------------	-----	--------

XGBoost (xgb)	Sell	0.4720
---------------	------	--------

Tabla 4.11: F1-score por modelo

Ensemble_weights.txt

```
Pesos asignados al ensemble según F1 medio:  
  
● BUY Ensemble:  
- xgb: 5  
- rf: 0  
- lgb: 4  
- cb: 5  
  
● SELL Ensemble:  
- xgb: 5  
- rf: 1  
- lgb: 4  
- cb: 4
```

Figura 4.10: Pesos por modelo en función de F1-score

- Validación **walk-forward** con generación de señales y optimización de umbrales (threshold_buy, threshold_sell).

Archivo	Tipo	Descripción
crossvalidated_walk_forward_ensemble_results.csv	CSV	Resultados del ensemble con validación cruzada.
crossvalidated_walk_forward_ensemble_summary.csv	CSV	Resultados del ensemble con validación cruzada resumidos.
walk_forward_ensemble_results.csv	CSV	Resultados del ensemble final en walk-forward.

Tabla 4.12: Outputs del walk-forward

`crossvalidated_walk_forward_ensemble_results.csv`

start_idx	end_idx	f1_buy	f1_sell	threshold_buy	threshold_sell	cycle
2080	2426	0.1951	0.3784	0.5073	0.5061	init_0.60
2253	2599	0.2222	0.4000	0.5478	0.5102	init_0.60
2426	2772	0.1600	0.4651	0.5262	0.5237	init_0.60
2599	2945	0.4000	0.3922	0.5513	0.5332	init_0.60
2772	3118	0.2326	0.4400	0.5305	0.5030	init_0.60

Tabla 4.13: Resultados de crossvalidated walk forward ensemble

`crossvalidated_walk_forward_ensemble_summary.csv`

cycle	f1_buy_mea	f1_buy_st	f1_sell_mea	f1_sell_st	thresh_buy_mea	thresh_sell_mea
	n	d	n	d	n	n
init_0.60	0.2552	0.1020	0.4022	0.0453	0.5387	0.5124
init_0.65	0.2642	0.0929	0.3903	0.0788	0.5353	0.5160
init_0.70	0.2945	0.1013	0.3759	0.0809	0.5272	0.5185
init_0.75	0.2705	0.1097	0.3792	0.1024	0.5599	0.5106

Tabla 4.14: Resumen del crossvalidated walk forward ensemble

walk_forward_ensemble_results.csv

start_idx	end_idx	f1_buy	f1_sell	threshold_buy	threshold_sell
2601	2947	0.2857	0.4561	0.6176	0.5008
2774	3120	0.2800	0.4091	0.5001	0.5265
2947	3293	0.3913	0.4231	0.5025	0.5018
3120	3466	0.1250	0.2286	0.6195	0.5133

Tabla 4.15: Resultados del walk forward ensemble

- Evaluación mediante métricas de clasificación (precisión, F1-score, recall, etc.)

Archivo	Tipo	Descripción
confusion_matrices.png	PNG	Matrices de confusión de las señales.
real_probability_histogram.png	PNG	Histograma de probabilidades reales (predicciones sin threshold).
model_metrics.txt	TXT	Métricas consolidadas de los modelos (F1-score, precision).

Tabla 4.16: Outputs de evaluación de métricas de entrenamiento

Confusión_matrices.png

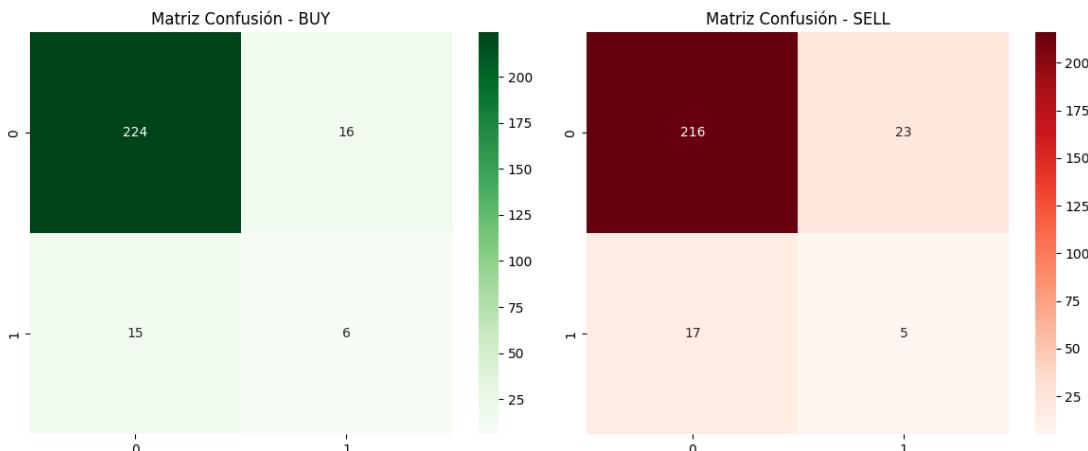


Figura 4.11: Matrices de confusión de compra y venta

Real_probability_histogram.png

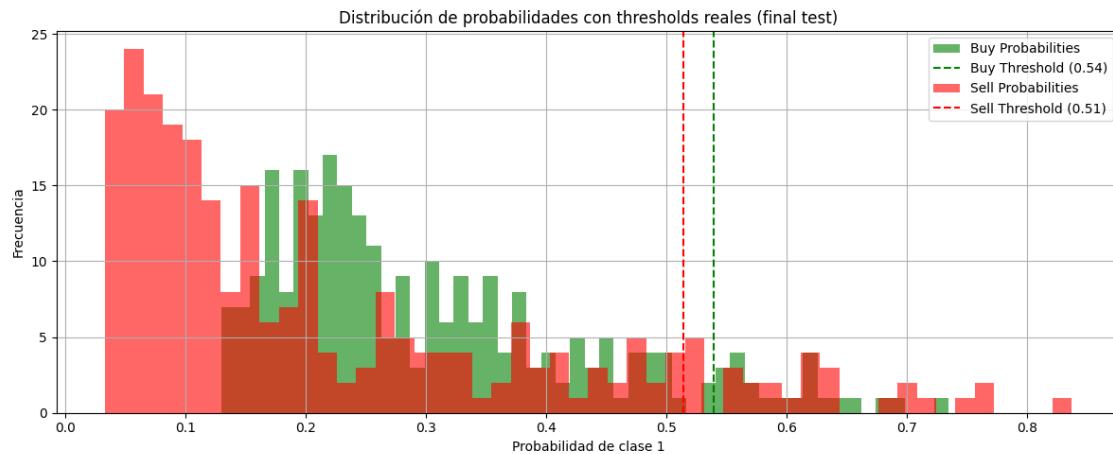


Figura 4.12: Histograma de probabilidad real

Model_metrics.txt

```
==== Model BUY (FINAL TEST) ====
Threshold: 0.5390
Confusion matrix:
[[224 16]
 [15 6]]

Classification report:
precision    recall    f1-score   support
      0          0.94      0.93      0.94      240
      1          0.27      0.29      0.28       21

accuracy                           0.88      261
macro avg       0.60      0.61      0.61      261
weighted avg    0.88      0.88      0.88      261
```

Figura 4.13: Metricas finales del modelo de compra

```
==== Model SELL (FINAL TEST) ====
Threshold: 0.5145
Confusion matrix:
[[216 23]
 [ 17  5]]

Classification report:
precision    recall    f1-score   support
          0       0.93      0.90      0.92     239
          1       0.18      0.23      0.20      22

accuracy                           0.85      261
macro avg       0.55      0.57      0.56     261
weighted avg    0.86      0.85      0.85     261
```

Figura 4.14: Métricas finales del modelo de venta

- Generación de parámetros de trading optimizados con **Optuna**

Archivo	Tipo	Descripción
threshold_buy.pkl	PKL	Umbral de probabilidad óptimo para compra.
threshold_sell.pkl	PKL	Umbral de probabilidad óptimo para venta.
thresholds_optimizados.txt	TXT	Resumen de ambos umbrales seleccionados.
parametros_trading_optimizados.txt	TXT	Take profit, stop loss y trailing stop finales vía Optuna.

Tabla 4.17: Outputs de Optuna para Trading

Thresholds_optimizados.txt

```
Threshold medio BUY: 0.5390
Threshold medio SELL: 0.5145
```

Figura 4.15: Thresholds finales de compra y venta

Parámetros_trading_optimizados.txt

```
Take Profit: 4.43%
Stop Loss: -0.62%
Trailing Stop: 1.98%
```

Figura 4.16: Parámetros finales de trading optimizados con Optuna

- Backtest y generación de plots

Archivo	Tipo	Descripción
tradelog.csv	CSV	Registro de todas las operaciones (entrada, salida, beneficio).
capital_evolution.png	PNG	Evolución del capital a lo largo del periodo de prueba.
signals_plot.png	PNG	Señales generadas vs precio del BTC.
profit_distribution.png	PNG	Distribución de beneficios por operación.
final_backtest_metrics.txt	TXT	Métricas del backtest final: aciertos, drawdown, beneficio.

Tabla 4.18: Outputs de Backtest y plots

Tradelog.csv

entry_time	exit_time	entry_price	exit_price	profit_pct
2025-03-10 00:00:00+00:00	2025-03-10 12:00:00+00:00	82028.39	80014.36	-2.55
2025-03-10 20:00:00+00:00	2025-03-11 04:00:00+00:00	78582.16	80449.16	2.27
2025-03-11 16:00:00+00:00	2025-03-12 00:00:00+00:00	83118.80	82103.81	-1.32
2025-03-13 16:00:00+00:00	2025-03-14 00:00:00+00:00	80147.98	81901.49	2.09
2025-03-15 08:00:00+00:00	2025-03-16 08:00:00+00:00	84061.34	82408.65	-2.06

Tabla 4.19: Fragmento del historial de trades en el backtest

Capital_evolution.png

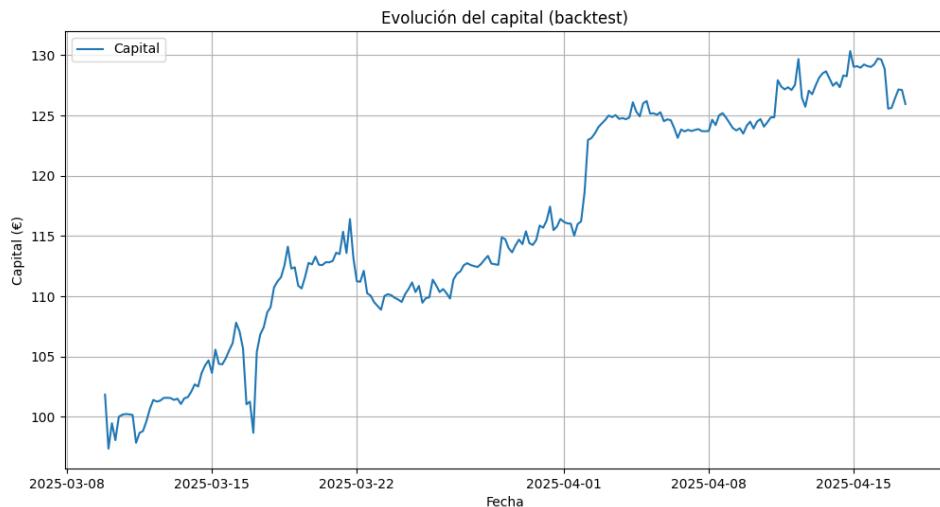


Figura 4.17: Evolución del capital en el backtest

Profit_distrinution.png

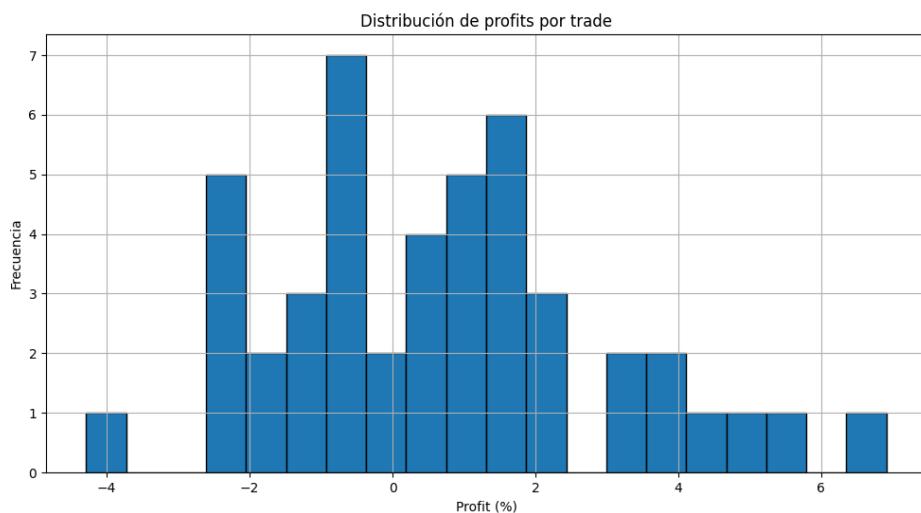


Figura 4.18: Distrinución de profit por operaciones

Signals_plot.png

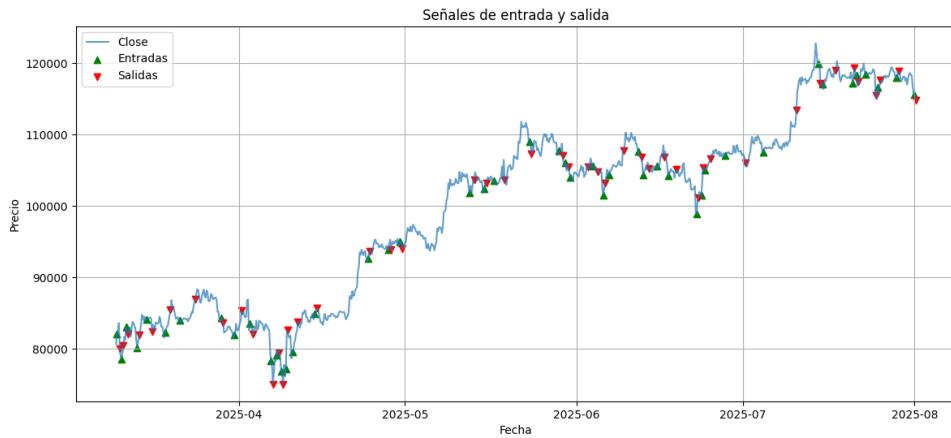


Figura 4.19: Distribución de señales de compra y venta en una gráfica

Final_backtest_metrics.txt

```
==== Métricas generales ====
Capital final: 125.96 €
Total trades: 46
Wins: 26
Losses: 20
Win rate: 0.57
Avg profit per trade: 0.63 %
Overall return: 25.96 %

==== Trading metrics avanzadas ====
avg_win: 2.24
avg_loss: -1.46
expectancy: 0.63
profit_factor: 1.99
payoff_ratio: 1.53
cagr: 0.80
max_drawdown: -0.08
volatility: 0.40
sharpe_ratio: 3.66
calmar_ratio: 9.38
avg_trade_duration_h: 24.17
```

Figura 4.20: Métricas finales del backtest

- **Desarrollo de un sistema funcional y autónomo de *paper trading*:**
 - Script ejecutable (*paper_trading_bot.py*) que simula operaciones reales sobre el mercado en vivo en una máquina virtual.
 - Replica misma lógica de buy y sell thresholds y para gestión de operaciones: apertura, cierre, *stop-loss*, *take-profit* y *trailing stop* que la obtenida durante el entrenamiento de los modelos.
 - **Visualización de resultados** en forma de gráficos de evolución del capital y registro de operaciones y evolución del capital en archivos .csv.

Archivo	Tipo	Descripción
capital_history.csv	CSV	Evolución del capital a lo largo del paper trading
tradelog.csv	CSV	Registro de todas las operaciones (entrada, salida, beneficio) durante el papertrading
Plot.png	PNG	Gráfica de la evolución del capital durante el paper trading

Tabla 4.20: Outputs del Paper Trading

Capital_history.csv

Timestamp	Capital
2025-07-05 08:00	100.00
2025-07-07 16:00	101.63
2025-07-18 20:00	102.39
2025-07-28 12:00	103.78
2025-08-05 04:00	105.46
2025-08-12 08:00	106.09
2025-08-14 08:00	105.03

Tabla 4.21: Historia del capital de paper trading

Tradelog.csv

timestamp_buy	price_buy	timestamp_sell	price_sell	profit_pct
2025-07-05 08:00	104382.73	2025-07-07 16:00	106079.45	1.63
2025-07-15 12:00	106992.84	2025-07-18 20:00	107798.02	0.74
2025-07-24 00:00	107103.29	2025-07-28 12:00	108550.47	1.35
2025-08-02 08:00	109429.17	2025-08-05 04:00	111203.63	1.62
2025-08-09 16:00	112478.36	2025-08-12 08:00	113153.91	0.59
2025-08-13 20:00	121232.48	2025-08-14 08:00	120301.25	-1.01

Tabla 4.22: Tradelog del paper trading

Plot.png



Figura 4.21: Gráfico del capital del paper trading

4.6.2 Evaluación del sistema

Durante la fase de backtesting y simulación real en *paper trading*, se observó que el sistema fue capaz de:

- Generar señales de entrada **coherentes con los movimientos del mercado**.
- Salir de operaciones de forma **rentable** en varias ocasiones aplicando la lógica de *take profit, stop loss, trailing stop* o mediante el modelo de venta.
- Evitar entradas en momentos de alta volatilidad no rentable gracias a la optimización de umbrales de probabilidad y a las features seleccionadas.

Si bien el sistema aún puede optimizarse en cuanto a precisión y robustez ante condiciones de mercado extremas, ha demostrado ser funcional, eficiente y capaz de operar de forma **completamente autónoma** durante varios días sin intervención humana.

4.6.3 Cambios respecto al planteamiento inicial

Durante el desarrollo del proyecto se realizaron algunos ajustes respecto a la planificación inicial, con el fin de mejorar la calidad del sistema final:

- Se sustituyó el enfoque tradicional de salida fija por un **modelo específico de predicción de caídas (sell model)**, entrenado con la misma lógica que el de compra, pero invertido.
- Se eliminaron procesos manuales como la selección de thresholds y se automatizó todo mediante validación cruzada y optimización con Optuna.
- Se descartaron métodos poco escalables, como arquitecturas profundas o datos de alta frecuencia, en favor de soluciones más interpretables y estables.

Capítulo 5. DISCUSIÓN

Este capítulo analiza los principales resultados obtenidos durante el desarrollo del sistema automatizado de *paper trading*, poniendo en perspectiva tanto los logros alcanzados como las limitaciones y decisiones adaptativas que surgieron durante el proceso. Aunque el enfoque general del proyecto ha sido aplicado y técnico, la reflexión crítica sobre el proceso es clave para valorar su calidad, utilidad y posibilidades de mejora.

5.1 Utilidad de la metodología aplicada

La metodología inicialmente planteada, basada en la construcción modular de un sistema de clasificación con validación walk-forward y optimización bayesiana, ha resultado **eficaz, reproducible y flexible**. El uso de **Optuna** para optimizar etiquetas, umbrales y parámetros de trading ha permitido mejorar el rendimiento general del sistema sin necesidad de ajuste manual.

La decisión de dividir el modelo en dos —uno para detectar señales de compra y otro para señales de venta— se ha demostrado especialmente útil. Esto ha permitido **refinar la lógica de salida** más allá de las condiciones estándar como *take profit* o *stop loss*, proporcionando mayor control y adaptabilidad ante diferentes contextos de mercado.

Asimismo, la validación *walk-forward* ha sido fundamental para evitar el *look-ahead bias* y asegurar que las decisiones tomadas en cada periodo estén basadas exclusivamente en información disponible en ese momento, algo esencial en entornos financieros reales.

5.2 Cambios y adaptaciones respecto a la planificación

Durante el desarrollo del proyecto, fue necesario realizar varios cambios sobre la marcha. Uno de los más relevantes fue reemplazar el sistema de salida basado únicamente en reglas fijas (TP/SL) por un **modelo de predicción de caídas del mercado (sell model)**, entrenado con lógica simétrica al modelo de compra. Esto surgió tras observar que el uso de umbrales fijos limitaba el rendimiento global.

Además, se integró una lógica completa para ejecutar la solución de forma **autónoma en una máquina virtual Linux**, algo que no estaba incluido en la propuesta inicial pero que mejoró enormemente la aplicabilidad real del sistema. También se optó por consolidar todo el código en un único script funcional, reduciendo dependencias y mejorando la mantenibilidad.

5.3 Limitaciones del proyecto

A pesar de su correcto funcionamiento, el sistema presenta algunas **limitaciones importantes**:

- **Falta de acceso a datos con mayor granularidad (por ejemplo, 1 minuto o ticks):** El uso de velas de 4 horas limita la capacidad de respuesta ante movimientos rápidos del mercado.
- **No se ha considerado el *slippage*** ni la latencia de ejecución, aunque en un entorno de producción son factores críticos.
- **No se ha utilizado gestión dinámica de capital (*position sizing*)**, operando siempre con el 100 % del capital disponible.
- **Limitación en la diversidad de estrategias:** El sistema implementa solo una lógica de entrada y salida basada en modelos de clasificación binaria. Otras técnicas como redes neuronales, reinforcement learning o series temporales (LSTM, ARIMA) no fueron exploradas por razones de foco y tiempo.

5.4 Impacto del resultado

El proyecto demuestra que es posible desarrollar, con recursos limitados, un sistema funcional, interpretativo y autónomo de *paper trading* sobre activos de alta volatilidad como Bitcoin. Más allá de los resultados financieros obtenidos, el mayor valor reside en:

- La **automatización completa del proceso**, desde la descarga de datos hasta la simulación de capital.
- La posibilidad de **adaptar el sistema a otros activos o estrategias**, con una mínima modificación.
- El aprendizaje práctico adquirido sobre todo el flujo de trabajo de sistemas de trading algorítmico, desde la validación estadística hasta el despliegue técnico.

Este resultado no solo cumple con los objetivos marcados, sino que abre la puerta a futuras líneas de investigación y mejora, tanto a nivel de rendimiento como de arquitectura.

Capítulo 6. CONCLUSIONES

6.1 Conclusiones del trabajo

El desarrollo de este proyecto ha permitido alcanzar de forma satisfactoria el objetivo general planteado: diseñar e implementar un **sistema automatizado de paper trading basado en modelos de clasificación** que opere sobre el activo BTC-USD con datos en intervalos de 4 horas.

A lo largo del trabajo se ha construido una solución completa, desde la preparación de datos y entrenamiento de modelos, hasta su despliegue operativo en una máquina virtual en tiempo real. El sistema integra múltiples técnicas avanzadas de *machine learning*, como validación walk-forward, selección de características, escalado, balanceo de clases y optimización de hiperparámetros mediante Optuna.

Además, se ha desarrollado una arquitectura modular que permite no solo detectar señales de compra de forma precisa, sino también identificar señales de salida mediante un segundo modelo entrenado específicamente para predecir caídas, mejorando así la lógica de cierre de operaciones más allá de reglas tradicionales como el *take profit* o *stop loss*.

El sistema ha sido validado mediante backtesting y pruebas en tiempo real, demostrando ser funcional, interpretable y adaptable a otros entornos financieros. Su estructura está preparada para escalar a futuros desarrollos, integrando nuevas estrategias, activos o niveles de automatización.

Por tanto, se concluye que el objetivo del proyecto ha sido **cumplido con éxito**, aportando una solución técnica robusta, reproducible y con aplicaciones reales en el ámbito del trading algorítmico.

6.2 Conclusiones personales

La realización de este proyecto ha supuesto una experiencia profundamente enriquecedora, tanto a nivel técnico como personal. Iniciar el desarrollo de un sistema automatizado de *paper trading* desde cero me ha permitido integrar y aplicar conocimientos adquiridos en múltiples áreas —*machine learning*, análisis de series temporales, ingeniería de datos, optimización, validación cruzada y automatización de procesos— dentro de un contexto práctico y desafiante como es el de los mercados financieros.

Uno de los aprendizajes más relevantes ha sido la importancia de la **rigurosidad metodológica**. Asegurar que no existieran sesgos como el *look-ahead bias*, validar correctamente mediante *walk-forward*, o realizar una optimización eficiente con Optuna, han sido elementos clave para construir un sistema sólido y creíble.

Fernando Álvaro Pastor

También destaco la **curva de aprendizaje técnico**, especialmente al trabajar con múltiples modelos, ajustar el preprocesamiento para datos financieros, integrar lógica de trading realista y finalmente desplegar la solución en una máquina virtual autónoma. Este proceso no solo ha ampliado mis capacidades como ingeniero y analista, sino que ha reforzado mi confianza en el desarrollo de soluciones completas, desde la idea hasta su ejecución real.

Más allá del componente académico, este proyecto me ha conectado con un ámbito que me resulta apasionante y en el que me gustaría seguir profundizando profesionalmente: la aplicación de la inteligencia artificial al análisis financiero, la automatización de decisiones y la optimización operativa.

En resumen, ha sido una experiencia desafiante, práctica y motivadora, que ha fortalecido mis competencias y reafirmado mi interés por seguir explorando la intersección entre datos, algoritmos y decisiones reales.

Capítulo 7. FUTURAS LÍNEAS DE TRABAJO

Durante el desarrollo del sistema automatizado de *paper trading*, han surgido múltiples oportunidades de mejora y expansión que, por cuestiones de alcance, tiempo o complejidad, no se han abordado en esta primera versión del proyecto. A continuación, se detallan las principales líneas de trabajo que podrían servir como base para una evolución futura del sistema:

7.1 Inclusión de gestión dinámica del capital

Actualmente, el sistema emplea una estrategia fija de inversión del 100 % del capital disponible en cada operación. Sería altamente recomendable incorporar técnicas de **position sizing** que ajusten el tamaño de las operaciones en función del riesgo estimado, la volatilidad del activo o el rendimiento reciente del sistema.

7.2 Modelado del slippage y la latencia

El entorno de *paper trading* actual no contempla **slippage**, **spreads dinámicos** ni **latencia de ejecución**, elementos que impactan directamente en el rendimiento en entornos reales. Incluir un modelo realista de costes operativos permitiría estimar con mayor precisión la rentabilidad neta del sistema.

7.3 Integración de múltiples estrategias o activos

Una evolución natural sería adaptar el sistema a operar con múltiples activos (por ejemplo, ETH-USD, índices bursátiles o divisas) o integrar distintas estrategias en paralelo (tendenciales, de reversión, eventos, etc.). Esto permitiría diversificar el riesgo y aprovechar oportunidades más amplias en el mercado.

7.4 Exploración de arquitecturas avanzadas de modelos

Aunque los modelos actuales basados en algoritmos de clasificación (XGBoost, LightGBM, Random Forest, CatBoost) han demostrado ser efectivos, sería interesante investigar otras arquitecturas, como **redes neuronales profundas (DNNs)**, **modelos secuenciales (LSTM)** o incluso **reinforcement learning**, especialmente para capturar dependencias temporales complejas o adaptar la estrategia en función de la evolución del entorno.

7.5 Mejora de la interfaz de seguimiento

Actualmente, el seguimiento del capital se realiza mediante CSVs y gráficos estáticos. Una futura mejora podría incluir el desarrollo de un **dashboard web interactivo** que muestre, en tiempo real, métricas clave del sistema, operaciones en curso, evolución del capital y alertas de rendimiento.

7.6 Despliegue en entorno real con capital reducido

Una vez validado el sistema en *paper trading*, el siguiente paso lógico sería su implementación en un entorno real con una cuenta de trading en un **broker con API pública**, como Binance, Interactive Brokers o Alpaca. Esto permitiría validar la robustez del sistema frente a condiciones reales del mercado.

7.7 Incorporación de técnicas de aprendizaje en línea

Otra línea de trabajo ambiciosa consistiría en dotar al sistema de **capacidad de aprendizaje continuo**, mediante la actualización periódica de los modelos con nuevos datos o la incorporación de técnicas de aprendizaje incremental, adaptándose así a cambios en las condiciones del mercado.

Capítulo 8. REFERENCIAS

- Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.
- Arnott, R. D., Harvey, C. R., & Markowitz, H. (2019). A backtesting protocol in the era of machine learning. *The Journal of Financial Data Science*, 1(1), 64-74.
- Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques – Part II: Soft computing methods. *Expert Systems with Applications*, 36(3).
- Bailey, D., Borwein, J., Lopez de Prado, M., & Zhu, Q. (2014). The probability of backtest overfitting. *Journal of Computational Finance*, 20(4).
- Baur, D., Hong, K., & Lee, A. (2018). Bitcoin: Medium of exchange or speculative assets? *Journal of International Financial Markets, Institutions and Money*, 54.
- Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, 24.
- Berrar, D. (2019). Cross-validation in machine learning. *Bioinformatics and Biomedicine*.
- Chan, E. (2017). *Algorithmic Trading: Winning Strategies and Their Rationale*. Wiley.
- Corbet, S., Lucey, B., Urquhart, A., & Yarovaya, L. (2019). Cryptocurrencies as a financial asset: A systematic analysis. *International Review of Financial Analysis*, 62.
- De Prado, M. L. (2018). *Advances in Financial Machine Learning*. Wiley.
- De Prado, M. L. (2020). *Machine Learning for Asset Managers*. Cambridge University Press.
- Dietterich, T. (2000). Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857.
- Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, 25(2).
- Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2).
- Gu, S., Kelly, B., & Xiu, D. (2020). Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies*, 33(5).
- Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*. OTexts.

Fernando Álvaro Pastor

Lundberg, S., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.

Molnar, C. (2019). *Interpretable Machine Learning*. Independently published.

Park, C.-H., & Irwin, S. H. (2007). What do we know about the profitability of technical analysis? *Journal of Economic Surveys*, 21(4).

Zhou, Z.-H. (2012). *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC.

[PÁGINA INTENCIONADAMENTE EN BLANCO]