



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE MÁSTER

*Predicción de series temporales utilizando inteligencia artificial:
Desde modelos clásicos hasta redes neuronales recurrentes*

Máster Universitario en Dirección TIC para la Defensa

ALUMNO: Rubén Palacios Sánchez

DIRECTORES: Francisco Troncoso Pastoriza

CURSO ACADÉMICO: 2025-2026

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE MÁSTER

*Predicción de series temporales utilizando inteligencia artificial:
Desde modelos clásicos hasta redes neuronales recurrentes*

Máster Universitario en Dirección TIC para la Defensa
Especialidad de Sistemas y Tecnologías de la Información

UniversidadeVigo

RESUMEN

Este trabajo analiza la aplicación de técnicas de inteligencia artificial a la predicción de series temporales bursátiles, comparando modelos estadísticos clásicos con enfoques modernos de aprendizaje automático y aprendizaje profundo. El objetivo principal es evaluar si estos modelos pueden aportar una ventaja real en la predicción de movimientos de precios y en la toma de decisiones de inversión.

El estudio comienza revisando los fundamentos de las series temporales, describiendo conceptos clave como estacionariedad, estacionalidad, autocorrelación, ruido blanco y caminata aleatoria, así como los principales problemas que aparecen al trabajar con datos financieros reales. A partir de esta base, se analizan distintos modelos predictivos, desde métodos tradicionales como ARIMA, ARIMAX y SARIMAX, hasta algoritmos de aprendizaje automático como Random Forest, XGBoost y LightGBM, y finalmente modelos de redes neuronales recurrentes, en particular LSTM y GRU.

Todo el desarrollo se implementa en Python, utilizando datos bursátiles reales obtenidos a través de la API de Interactive Brokers. Los datos se preparan mediante procesos de limpieza, construcción de indicadores técnicos y normalización para garantizar un entrenamiento estable de los modelos. La evaluación se apoya en un enfoque práctico basado en backtesting, simulando estrategias de inversión bajo distintos rangos porcentuales de movimiento.

PALABRAS CLAVE

Series temporales, inteligencia artificial, mercados financieros, aprendizaje profundo, data science.

AGRADECIMIENTOS

Quiero agradecer a mi tutor Francisco Troncoso Pastoriza su orientación y disponibilidad a lo largo de todo el desarrollo de este trabajo, que ha sido clave para enfocar correctamente los objetivos y la metodología.

Del mismo modo, agradezco su apoyo continuo y sus observaciones críticas, que han contribuido de manera decisiva a la calidad final del proyecto.

CONTENIDO

Contenido	1
Índice de Figuras	4
Índice de Tablas.....	7
1 Introducción y objetivos	8
1.1 Contexto	8
1.2 Objetivos	9
1.3 Desafíos actuales.....	10
1.4 Estructura del documento.....	11
2 Fundamentos de Series Temporales y Predicción.....	13
2.1 Introducción a Series Temporales.....	13
2.2 Características fundamentales de las series temporales.....	14
2.3 Desafíos de las series temporales.....	16
3 Métodos de Predicción: Teoría y Aplicación.....	19
3.1 Introducción a modelos predictivos	19
3.2 Modelos estadísticos tradicionales.....	19
3.2.1 Modelo AR: Autorregresivo	19
3.2.2 Modelo MA: Medias Móviles	20
3.2.3 Modelo ARMA: Autorregresivo de Medias Móviles.....	20
3.2.4 Modelo ARIMA: Autorregresivo Integrado de Medias Móviles	21
3.2.5 Modelos ARMAX y ARIMAX: ARMA y ARIMA con variables exógenas.....	22
3.2.6 Modelo SARIMAX: ARIMAX con componente estacional.....	22
3.3 Modelos de aprendizaje automático.....	24
3.3.1 Introducción al aprendizaje supervisado en series temporales	24
3.3.2 Árbol de decisiones.....	25
3.3.3 Random Forest.....	26
3.3.4 Gradient Boosting Machines	27
3.4 Modelos de aprendizaje profundo.....	29
3.4.1 Introducción al aprendizaje profundo y redes neuronales	29
3.4.2 RNN: Redes Neuronales Recurrentes.....	31
3.4.3 LSTM: Long Short-Term Memory	35
3.4.4 GRU: Gated Recurrent Unit	38
4 Obtención y Preparación de los Datos	41

4.1 Descripción del proceso de preparación de datos	41
4.2 Fuentes y obtención de datos bursátiles.....	42
4.3 Definición e importancia del análisis técnico	43
4.4 Construcción de indicadores técnicos relevantes.....	44
4.5 Limpieza y transformación de los datos	44
4.6 Normalización de los datos	45
5 Metodología de Evaluación de Modelos y Estrategias	46
5.1 La importancia del análisis temporal en la estimación de precios bursátiles.....	46
5.2 Evaluación de la Estrategia de Inversión y Definición de Criterios Operación.....	48
5.3 Métricas de Evaluación y Análisis de Resultados	50
6 Modelados Predictivos: Implementación y Resultados.....	55
6.1 Implementación de modelos tradicionales de series temporales	55
6.1.1 Proceso Auto-ARIMA	55
6.1.2 ARIMA: implementación y resultados.....	56
6.1.3 SARIMAX: implementación y resultados.....	57
6.2 Implementación de modelos de aprendizaje automático	58
6.2.1 Random Forest: implementación y resultados.....	58
6.2.2 XGBoost: implementación y resultados	59
6.2.3 LightGBM: implementación y resultados	60
6.3 Implementación de modelos de aprendizaje profundo	61
6.3.1 GRU: implementación y resultados.....	61
6.3.2 LSTM: implementación y resultados.....	62
7 Discusión y Análisis Comparativo con Data Science	64
7.1 Comparación entre diferentes modelos de predicción	64
7.2 Comparación entre diferentes indicadores.....	65
7.2.1 Stocks.....	65
7.2.2 Num lookback.....	66
7.2.3 Num forecast.....	67
7.2.4 Batch size.....	68
7.2.5 Epochs.....	69
7.2.6 Learning rate.....	70
7.2.7 Variable a predecir.....	71
7.3 Correlación entre diferentes parámetros	72
7.3.1 Gráficos de dispersión (scatterplots)	72
7.3.2 Diagramas de cajas (boxplots).....	73
7.3.3 Matriz de correlación	73

8 Backtesting y Evaluación de Estrategias de Inversión.....	75
8.1 Definición e importancia del <i>backtesting</i>	75
8.2 Diseño de estrategias basadas en predicciones	76
8.3 Evaluación de rentabilidad y riesgo en datos históricos	81
9 Conclusiones y Líneas Futuras de Investigación	86
9.1 Principales conclusiones del trabajo	86
9.2 Implicaciones prácticas para estrategias de inversión.....	87
9.3 Limitaciones y mejoras futuras en modelos y datos	87
10 Bibliografía.....	90
Anexo I: Abreviaturas y acrónimos.....	95
Anexo II: INDICADORES TÉCNICOS	97
Anexo III: INDICADORES TÉCNICOS.....	99

ÍNDICE DE FIGURAS

Figura 1 Ejemplos gráficos de estacionariedad [11]	15
Figura 2 Estacionariedad en forma débil.....	15
Figura 3 Ventas de helados en función de los meses [13].....	16
Figura 4 Saltos entre los valores de la serie temporal	21
Figura 5 “Snowboarding” en Google Trends.....	23
Figura 6 Etiquetado de datos con aprendizaje supervisado [27]	25
Figura 7 Ejemplo de árbol de decisión para estimar el precio de una vivienda [27]	26
Figura 8 Estructura general de un modelo Random Forest [30]	27
Figura 9 Algoritmo de <i>Gradient Boosting</i> [33].....	28
Figura 10 Esquema de un perceptrón con tres entradas y una salida [35]	29
Figura 11 Estructura de una red neuronal [35].....	30
Figura 12 Analogía Red Neuronal FeedForward (FNN) y Red Neuronal Recurrente (RNN) [37].	31
Figura 13 Despliegue de una Red Neuronal Recurrente (RNN) en el tiempo [37].....	32
Figura 14 <i>Backpropagation</i> a través del tiempo (BPTT) en una RNN para $t = 4$ [37]	33
Figura 15 Descenso del Gradiente y Problemas de <i>Vanishing</i> y <i>Exploding Gradient</i> [37]	35
Figura 16 Celda LSTM con compuertas de entrada (i), olvido (f) y salida (o) [38]	36
Figura 17 Estructura de una capa LSTM con múltiples celdas conectadas en secuencia [39]	38
Figura 18 Celda de la GRU con compuertas de actualización (z) y reinicio (r) [42].....	39
Figura 19 Estructura de una capa GRU con múltiples celdas conectadas en secuencia [43].....	40
Figura 20 Comparación entre sobreajuste y entrenamiento con buena generalización.	49
Figura 21 Ejemplo de estructura de resultados y métricas de evaluación por fichero porcentual ...	52
Figura 22 Ejemplo de predicción de precios predichos vs precios reales	53
Figura 23 Ejemplo de predicción de porcentaje de cambio vs porcentaje de cambio real.....	54
Figura 24 Estadísticas del modelo ARIMA aplicadas a precios de cierre en cada rango porcentual.	56
Figura 25 Estadísticas del modelo ARIMA aplicadas a porcentajes de cambio en cada rango porcentual.....	56
Figura 26 Estadísticas del modelo SARIMAX aplicadas a precios de cierre en cada rango porcentual.....	57
Figura 27 Estadísticas del modelo SARIMAX aplicadas a porcentajes de cambio en cada rango porcentual.....	57
Figura 28 Estadísticas del modelo Random Forest aplicadas a precios de cierre en cada rango porcentual.....	58
Figura 29 Estadísticas del modelo Random Forest aplicadas a porcentajes de cambio en cada rango porcentual.....	58

Figura 30 Estadísticas del modelo XGBoost aplicadas a precios de cierre en cada rango porcentual.....	59
Figura 31 Estadísticas del modelo XGBoost aplicadas a porcentajes de cambio en cada rango porcentual.....	59
Figura 32 Estadísticas del modelo LightGBM aplicadas a precios de cierre en cada rango porcentual.....	60
Figura 33 Estadísticas del modelo LightGBM aplicadas a porcentajes de cambio en cada rango porcentual.....	60
Figura 34 Estadísticas del modelo GRU aplicadas a precios de cierre en cada rango porcentual. ..	61
Figura 35 Estadísticas del modelo GRU aplicadas a porcentajes de cambio en cada rango porcentual.....	62
Figura 36 Estadísticas del modelo LSTM aplicadas a precios de cierre en cada rango porcentual.	62
Figura 37 Estadísticas del modelo LSTM aplicadas a porcentajes de cambio en cada rango porcentual.....	63
Figura 38 Resultados comparativos de los modelos en la predicción de precios de cierre.....	64
Figura 39 Resultados comparativos de los modelos en la predicción del porcentaje de cambio.....	65
Figura 40 Resultados comparativos de los activos en la predicción de precios de cierre.....	65
Figura 41 Resultados comparativos de los activos en la predicción del porcentaje de cambio.....	66
Figura 42 Resultados comparativos de los diferentes <i>num lookback</i> en la predicción de precios de cierre.....	66
Figura 43 Resultados comparativos de los diferentes <i>num lookback</i> en la predicción del porcentaje de cambio	67
Figura 44 Resultados comparativos de los diferentes <i>num forecast</i> en la predicción de precios de cierre.....	67
Figura 45 Resultados comparativos de los diferentes <i>num forecast</i> en la predicción del porcentaje de cambio	68
Figura 46 Resultados comparativos de los diferentes <i>batch size</i> en la predicción de precios de cierre.....	68
Figura 47 Resultados comparativos de los diferentes <i>batch size</i> en la predicción del porcentaje de cambio.....	69
Figura 48 Resultados comparativos de los diferentes <i>epochs</i> en la predicción de precios de cierre	69
Figura 49 Resultados comparativos de los diferentes <i>epochs</i> en la predicción del porcentaje de cambio.....	70
Figura 50 Resultados comparativos de los diferentes <i>learning rate</i> en la predicción de precios de cierre.....	70
Figura 51 Resultados comparativos de los diferentes <i>learning rate</i> en la predicción del porcentaje de cambio	71
Figura 52 Resultados comparativos de las diferentes variables a predecir	71
Figura 53 Correlación entre diferentes variables	73

Figura 54 Diagramas de cajas entre diferentes variables73

Figura 55 Matriz de correlación74

ÍNDICE DE TABLAS

Tabla 1 Estructura de los datos bursátiles descargados.....	43
Tabla 2 Tabla de estructura de resultados para la evaluación de modelos.....	50
Tabla 3 Resultados mensuales de las operaciones según la predicción de los precios de cierre	78
Tabla 4 Resultados mensuales de las operaciones según la predicción del porcentaje de cambio ..	80
Tabla 5 Resultados anuales de las operaciones según la predicción de los precios de cierre.....	80
Tabla 6 Resultados anuales de las operaciones según la predicción del porcentaje de cambio.....	81
Tabla 7 Resultados mensuales de la estrategia real según la predicción del precio de cierre.....	84
Tabla 8 Resultados mensuales de la estrategia real según la predicción del porcentaje de cambio.	85
Tabla 9 Resultados anuales de la estrategia real según la predicción del precio de cierre.....	85
Tabla 10 Resultados anuales de la estrategia real según la predicción del porcentaje de cambio ...	85

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Contexto

La predicción de series temporales constituye un pilar fundamental en el desarrollo de la Industria 4.0, donde los datos se han convertido en el principal activo estratégico. En un entorno caracterizado por la digitalización, la automatización y la interconexión de sistemas, la capacidad de anticipar comportamientos futuros a partir de datos históricos resulta esencial para la toma de decisiones inteligentes y la optimización de procesos industriales.

La integración de tecnologías como el Internet de las cosas (IoT), la inteligencia artificial (IA), la computación en la nube y el análisis masivo de datos ha permitido generar y procesar volúmenes sin precedentes de información temporal. Estas tecnologías no solo facilitan la monitorización en tiempo real, sino que posibilitan la construcción de modelos predictivos capaces de detectar patrones, prever fallos y ajustar dinámicamente la producción.

En este contexto, las series temporales adquieren una gran relevancia, aplicándose en ámbitos como el mantenimiento predictivo, la gestión de la energía, la planificación de la demanda, la detección de anomalías o la optimización logística. Su correcta utilización contribuye a mejorar la eficiencia, reducir costes operativos y aumentar la resiliencia de las cadenas de valor.

En definitiva, la predicción de series temporales representa una herramienta clave para transformar los datos industriales en conocimiento útil, impulsando una nueva era de fabricación inteligente basada en la anticipación, la adaptabilidad y la toma de decisiones basada en evidencia [1].

Asimismo, la predicción de series temporales trasciende el ámbito industrial y encuentra amplias aplicaciones en el sector financiero. Se utiliza para el pronóstico de precios de activos, la gestión del riesgo, la detección de anomalías en transacciones y la optimización de carteras, convirtiéndose en una herramienta esencial para la toma de decisiones estratégicas basadas en datos.

La inteligencia artificial (IA) aplicada a las finanzas ha sido durante décadas un área de investigación de gran interés. Las aplicaciones clásicas de la inteligencia artificial en finanzas y economía (como en mercados financieros tradicionales, trading, banca, seguros, gestión del riesgo, regulación o marketing) han evolucionado hacia una nueva generación de tecnologías financieras (*FinTech*), que abarcan desde monedas digitales inteligentes hasta gestión de activos y patrimonio, pagos, préstamos, auditoría, contabilidad, regulación y riesgos.

En este contexto, el término "finanzas" engloba un amplio espectro de áreas como los mercados de capitales, operaciones de trading, banca, seguros, préstamos e inversiones, gestión de activos y riesgos, marketing financiero, cumplimiento normativo, pagos, contratos, auditoría, infraestructura financiera, *blockchain*, operaciones financieras, servicios financieros, ciberseguridad y ética financiera.

Actualmente, finanzas y economía (*EcoFin*) se encuentran cada vez más interconectadas y retroalimentadas por los avances en inteligencia artificial y ciencia de datos (*AIDS: Artificial*

Intelligence and Data Science). Esta transformación paradigmática ha sido impulsada principalmente por el progreso en estas tecnologías emergentes, que están revolucionando la economía, los servicios financieros, los medios de comunicación, la tecnología y la sociedad en general.

El conjunto de técnicas *AIDS* incluye tanto enfoques clásicos (como lógica, planificación, representación del conocimiento, modelado estadístico y matemático, optimización, sistemas expertos, simulación, procesamiento de lenguaje natural, visión por computador o reconocimiento de patrones) como métodos modernos basados en aprendizaje automático, representación de datos, minería de datos, aprendizaje profundo, computación cognitiva e incluso computación cuántica.

Estas técnicas no solo redefinen los productos y servicios de las finanzas modernas, sino que habilitan una nueva generación de soluciones inteligentes, personalizadas, interconectadas y seguras. Este ecosistema da lugar a lo que se denomina *smart FinTech*, que abarca desde banca inteligente (*smart banking*), seguros, préstamos y trading inteligentes, hasta gestión de riesgos, regulación, contabilidad y gobernanza automatizada. De aquí surgen subdisciplinas como *BankingTech*, *WealthTech*, *PayTech*, *RiskTech*, *RegTech*, entre otras.

Aunque existen múltiples revisiones en la literatura sobre la aplicación de inteligencia artificial en las finanzas, muchas se centran en técnicas o problemas específicos, como análisis de series temporales, procesamiento de lenguaje natural, detección de fraudes, scoring crediticio o trading algorítmico. Sin embargo, son escasos los trabajos que abordan de forma integral el ecosistema completo de técnicas, áreas de negocio y sus interrelaciones [2].

Con el auge de la tecnología y de la digitalización global, la predicción del mercado bursátil ha entrado en una era tecnológicamente avanzada, transformando los antiguos modelos de inversión. Con el continuo aumento de la capitalización de mercado, la compra-venta de acciones se ha convertido en un foco de interés para numerosos inversores financieros. A lo largo del tiempo, muchos analistas e investigadores han desarrollado herramientas y técnicas destinadas a anticipar el comportamiento de los precios bursátiles, facilitando así la toma de decisiones estratégicas.

Los modelos de predicción más avanzados permiten a los investigadores utilizar datos no tradicionales, como la información textual proveniente de plataformas sociales, para mejorar la precisión de sus análisis. En particular, la aplicación de enfoques de aprendizaje automático, como el análisis de datos textuales y los métodos de ensamblado (*ensemble methods*), ha contribuido significativamente al aumento de la exactitud en las predicciones.

No obstante, el análisis y la predicción del mercado bursátil continúan siendo un área de investigación compleja y desafiante, debido a la naturaleza dinámica, errática y caótica de los datos financieros [3].

1.2 Objetivos

El objetivo del presente Trabajo de Fin de Máster se centra en el estudio y la aplicación de diferentes técnicas de predicción de series temporales, abarcando métodos tradicionales y avanzados basados en Inteligencia Artificial. Inicialmente, se realizará un análisis exhaustivo de los métodos clásicos, con especial atención a modelos estadísticos como *ARIMA* (*AutoRegressive Integrated Moving Average*) y sus variantes. Se estudiarán los supuestos, ventajas y limitaciones de estos modelos, aplicándolos a distintos conjuntos de datos para evaluar su capacidad predictiva.

Posteriormente, el trabajo avanzará hacia técnicas basadas en árboles de decisión, particularmente algoritmos de ensamble (que combina múltiples modelos) como *Random Forest* y *Gradient Boosting Machines* (GBM). Aunque estos métodos no están diseñados específicamente para datos secuenciales, mediante la selección adecuada de características, se explorará su viabilidad y rendimiento en tareas de predicción temporal. Esta fase servirá como transición hacia métodos más complejos, destacando la importancia de la selección de variables.

La etapa final del proyecto estará dedicada a modelos de aprendizaje profundo, enfocándose en redes neuronales recurrentes (RNN) y sus variantes modernas como LSTM (*Long Short-Term Memory*) y GRU (*Gated Recurrent Unit*). Se analizarán sus arquitecturas, su capacidad para capturar dependencias de largo plazo y su comportamiento frente a series temporales complejas y no lineales.

Todo el desarrollo del trabajo se implementará utilizando el lenguaje de programación Python, haciendo uso de librerías especializadas como *statsmodels*, *scikit-learn*, *tensorflow* o *pytorch*. El proyecto se dividirá en dos grandes bloques: una primera parte centrada en la programación y desarrollo de los modelos, incluyendo el tratamiento de datos, la construcción y entrenamiento de los algoritmos, y una segunda parte destinada al análisis y comparación de los resultados, evaluando la precisión de las predicciones y la eficiencia de los métodos aplicados.

Finalmente, se presentarán las conclusiones basadas en el análisis comparativo de los resultados, destacando las fortalezas y debilidades de cada enfoque. Se discutirán también posibles líneas futuras de investigación, como la integración de redes neuronales convolucionales (CNN) en series temporales o el uso de modelos de atención como *Transformers* adaptados a datos secuenciales.

Estos modelos se podrán extrapolar a cualquier tipo de serie temporal, y la idea es que sea la base para poder entrenar cualquier estructura de datos, pero se realizará con activos financieros como acciones o divisas, analizando el precio y el retorno de éstos, haciendo una predicción buscando una ventaja estadística y obteniendop una estrategia rentable que pueda ser automatizada.

1.3 Desafíos actuales

La inteligencia artificial ha sido uno de los temas más discutidos en los últimos años. Durante la última década, diversas industrias a nivel mundial han comenzado a implementar nuevas tecnologías basadas en inteligencia artificial con el fin de mejorar la eficiencia y precisión de sus sistemas. En el ámbito de la salud, por ejemplo, los cirujanos asistidos por inteligencia artificial están revolucionando los tratamientos médicos, mientras que los vehículos autónomos están transformando radicalmente la industria automotriz.

Hasta hace poco, la inteligencia artificial parecía una tecnología propia del futuro. Sin embargo, con la aparición de plataformas generativas como ChatGPT, su integración en la vida cotidiana se ha vuelto tangible, ofreciendo a millones de personas la oportunidad de experimentar cómo esta tecnología puede mejorar distintos aspectos de sus vidas.

A medida que se profundiza en el abanico de posibles aplicaciones de la IA, su incorporación al mercado bursátil se perfila como una realidad cada vez más alcanzable. Dos capacidades fundamentales de la inteligencia artificial (el procesamiento masivo de datos y la predicción) resultan especialmente relevantes en este ámbito. A través de modelos analíticos que incluyen el análisis de sentimientos y el análisis en tiempo real, la inteligencia artificial puede mitigar desafíos clave como la volatilidad del mercado y la limitada accesibilidad para ciertos inversores.

Estos avances, impulsados por el aprendizaje automático y sus distintas ramas, podrían transformar de forma radical el funcionamiento de los mercados financieros.

El mercado bursátil enfrenta desafíos persistentes, como ineficiencias, volatilidad y barreras de entrada, que dificultan su accesibilidad y fiabilidad para los inversores. Este trabajo analiza cómo la inteligencia artificial puede abordar dichos retos, transformando la investigación, la automatización y la capacidad predictiva en los mercados financieros.

La pregunta central que guía esta investigación es: **“¿Cómo puede la inteligencia artificial mejorar la eficiencia del mercado, la toma de decisiones por parte de los inversores y la accesibilidad al mercado bursátil, y cuáles son los desafíos éticos y regulatorios asociados?”**

A través de una revisión sistemática de la literatura, se examinan técnicas de procesamiento de datos como el análisis de grandes volúmenes de información (*big data*) y el aprendizaje automático,

fundamentales para el desarrollo de modelos de análisis impulsados por inteligencia artificial, tales como el análisis en tiempo real y el análisis de sentimientos.

Asimismo, se evalúa el papel de la inteligencia artificial en la automatización y predicción dentro del mercado financiero, destacando su aplicación en la gestión de carteras, el análisis predictivo y la mitigación del riesgo. Se ha puesto especial énfasis en técnicas avanzadas de aprendizaje automático como el *deep learning*, el *reinforcement learning*, *random forest* y la inteligencia artificial generativa.

Los hallazgos de la investigación indican que la inteligencia artificial contribuye significativamente a mejorar la eficiencia del mercado, aumenta la precisión en la toma de decisiones y amplía el acceso al mercado para una mayor diversidad de inversores. No obstante, persisten desafíos éticos y regulatorios importantes, tales como la rendición de cuentas, la equidad y la protección de datos, los cuales deben ser abordados para garantizar una implementación exitosa.

En conclusión, aunque la inteligencia artificial tiene el potencial de transformar el mercado bursátil en un entorno más eficiente e inclusivo, su adopción exitosa exige una integración responsable y una supervisión regulatoria sólida [4].

1.4 Estructura del documento

El presente Trabajo de Fin de Máster se organiza en secciones que abarcan desde los fundamentos teóricos hasta la aplicación práctica y las conclusiones finales.

- **Sección 1 – Introducción y objetivos:** Expone el contexto general de la Industria 4.0 y la relevancia de la predicción de series temporales en entornos industriales y financieros. Presenta los objetivos del trabajo, el alcance de la investigación y los principales desafíos actuales en la aplicación de la inteligencia artificial al análisis de datos bursátiles.
- **Sección 2 – Fundamentos de series temporales y predicción:** Introduce los conceptos básicos sobre series temporales, sus características fundamentales (como estacionariedad, autocorrelación y estacionalidad) y los principales retos asociados a su análisis, estableciendo la base conceptual necesaria para los modelos posteriores.
- **Sección 3 – Métodos de predicción: teoría y aplicación:** Describe los principales enfoques teóricos empleados en la predicción temporal. Se analizan los modelos estadísticos clásicos (AR, MA, ARMA, ARIMA, ARIMAX, SARIMAX) y su formulación matemática, sentando las bases para su comparación con técnicas más avanzadas.
- **Sección 4 – Obtención y preparación de los datos:** Detalla el proceso de recopilación, limpieza y transformación de los datos bursátiles utilizados en el estudio. Explica las fuentes de información, la construcción de indicadores técnicos y los métodos de normalización aplicados antes del modelado.
- **Sección 5 – Metodología de evaluación de modelos y estrategias:** Define los criterios de evaluación empleados para medir el rendimiento de los modelos predictivos. Incluye métricas como MAE, MAPE, precisión, recall y F1-score, así como el diseño de estrategias de inversión basadas en las predicciones.
- **Sección 6 – Modelados predictivos: implementación y resultados:** Presenta la implementación práctica de los modelos desarrollados. Se incluyen los resultados obtenidos con los modelos estadísticos, de aprendizaje automático (Random Forest, XGBoost, LightGBM) y de aprendizaje profundo (RNN, LSTM, GRU).
- **Sección 7 – Discusión y análisis comparativo con Data Science:** Realiza un análisis comparativo entre los diferentes modelos, parámetros e indicadores. Se estudian correlaciones, comportamientos de entrenamiento y sensibilidad de los modelos frente a distintas configuraciones.

- **Sección 8 – Backtesting y evaluación de estrategias de inversión:** Aplica los modelos desarrollados a datos históricos para evaluar su viabilidad en estrategias reales de inversión. Analiza la rentabilidad y el riesgo, validando la efectividad de las predicciones mediante simulaciones retrospectivas.
- **Sección 9 – Conclusiones y líneas futuras de investigación:** Resume los principales hallazgos del trabajo, identifica las fortalezas y limitaciones de cada enfoque y propone futuras líneas de investigación, como la integración de CNNs o Transformers en el análisis de series temporales financieras.

2 FUNDAMENTOS DE SERIES TEMPORALES Y PREDICCIÓN

2.1 Introducción a Series Temporales

La predicción del futuro ha sido una inquietud constante para la humanidad desde tiempos antiguos, como lo demuestran registros históricos de las primeras civilizaciones. Esta fascinación surge de la necesidad de anticiparse en contextos marcados por la incertidumbre, donde prever lo que vendrá puede ofrecer ventajas cruciales.

En la actualidad, la predicción de series temporales se ha consolidado como una herramienta clave en numerosas disciplinas como la ciencia, la ingeniería, la economía, la sociología o la medicina. La posibilidad de anticipar la evolución de determinados indicadores tiene un impacto profundo en diversos ámbitos de la vida cotidiana, desde la gestión económica y la seguridad pública hasta la previsión de fenómenos climáticos extremos.

Aunque estas técnicas ya se aplican con éxito en sectores como la economía o la meteorología, todavía existen áreas donde su potencial no ha sido plenamente explotado. Tal es el caso de los campos que trabajan con datos geotécnicos y estructurales, donde la aplicación de modelos de series temporales puede representar un gran avance para la investigación, el monitoreo y la toma de decisiones. Su uso permitiría mejorar la seguridad, optimizar operaciones y facilitar una asignación más eficiente de recursos en proyectos de ingeniería y construcción [5].

Una serie temporal es una sucesión de N observaciones (datos) dispuestas de forma cronológica y espaciadas de manera regular en el tiempo. Estas observaciones pueden referirse a una única característica (lo que se conoce como serie univariante o escalar) o a múltiples características simultáneamente (denominada serie multivariante o vectorial) de una misma unidad analizada en distintos momentos [6].

El análisis de datos orientados en el tiempo y la predicción de valores futuros en una serie temporal son algunos de los desafíos más relevantes que enfrentan los analistas en numerosos campos, que van desde las finanzas y la economía, hasta la gestión de operaciones productivas, el análisis de políticas sociales y políticas públicas, o la evaluación del impacto humano y de las decisiones gubernamentales sobre el medio ambiente.

Por ello, existe un amplio abanico de profesionales (en áreas como finanzas, economía, ciencias, ingeniería, estadística y políticas públicas) que necesitan comprender los conceptos fundamentales del análisis y la predicción de series temporales. Lamentablemente, la mayoría de los libros introductorios de estadística o gestión operativa dedican muy poco espacio (cuando lo hacen) al tratamiento de datos temporales y ofrecen escasa orientación sobre cómo realizar predicciones.

Una predicción es una estimación sobre uno o varios eventos futuros. Como señaló en su momento Niels Bohr, hacer buenas predicciones no siempre es tarea fácil.

La predicción es un problema central en múltiples disciplinas, incluyendo el mundo empresarial e industrial, la administración pública, la economía, las ciencias ambientales, la medicina, las ciencias sociales, la política y las finanzas. Los problemas de predicción suelen clasificarse según el horizonte temporal en tres categorías: corto plazo, medio plazo y largo plazo.

Las predicciones a corto plazo buscan anticipar eventos que ocurrirán en los próximos días, semanas o pocos meses. Los de medio plazo suelen abarcar entre uno y dos años, mientras que los de largo plazo pueden proyectarse a varios años vista. Las predicciones de corto y medio plazo son esenciales en tareas como la planificación operativa, la elaboración de presupuestos o la selección de proyectos de I+D. Por otro lado, las predicciones a largo plazo son fundamentales para la planificación estratégica.

En general, las predicciones de corto y medio plazo se basan en identificar, modelar y extrapolar patrones a partir de datos históricos. Dado que estos datos suelen mostrar cierta inercia (es decir, tienden a evolucionar de forma gradual), los métodos estadísticos resultan especialmente eficaces en este contexto.

La mayoría de los problemas de predicción se apoyan en datos recogidos en forma de series temporales, es decir, secuencias cronológicas de observaciones sobre una determinada variable de interés [7].

En el contexto del análisis de series temporales, es importante distinguir entre **predicción (prediction)** y **pronóstico (forecasting)**:

- La **predicción** se refiere a la estimación de valores pasados o presentes desconocidos dentro de los datos disponibles, por ejemplo, durante la fase de validación del modelo utilizando subconjuntos de datos históricos.
- El **pronóstico** implica proyectar valores hacia el futuro, es decir, estimar observaciones aún no ocurridas, basándose en patrones aprendidos por el modelo.

Mientras que la predicción sirve para evaluar la capacidad del modelo de generalizar, el pronóstico representa su aplicación práctica en escenarios reales.

No obstante, en este trabajo se empleará mayoritariamente el término "predicción" por simplicidad terminológica, ya que la distinción entre ambos conceptos no afecta de forma significativa al objetivo ni a la interpretación de los resultados.

2.2 Características fundamentales de las series temporales

Las características fundamentales de una serie temporal permiten describir su comportamiento y facilitan su modelado. El análisis conjunto de estas propiedades es esencial para seleccionar técnicas de predicción adecuadas y mejorar la precisión de los modelos.

- **Ruido blanco:** El ruido blanco es único dentro de las series temporales porque es completamente aleatorio y carece de cualquier estructura. A diferencia de otras series que pueden mostrar tendencias, ciclos o autocorrelación, el ruido blanco es impredecible y presenta un comportamiento caótico [8]. La falta de autocorrelación en el ruido blanco va a implicar que no hay una relación clara entre valores pasados y presentes, por tanto, el ruido blanco va a ser una secuencia de valores aleatorios.
- **Caminata aleatoria (Random walk):** Una caminata aleatoria es un tipo de serie temporal en el que el valor actual es igual al valor previo más un cambio aleatorio hacia arriba o hacia abajo. De manera más formal, se puede definir según la Ecuación 1, donde $\mathbf{x}_t$ es el periodo actual, $\mathbf{x}_{t-1}$ el periodo anterior y $\boldsymbol{\varepsilon}_t$ el residuo aleatorio [9]:

$$x_t = x_{t-1} + \varepsilon_t \quad (1)$$

Por lo que los residuos son ruido blanco y no se pueden predecir, y esto sugiere que las mejores estimaciones para los valores de hoy serán los valores de ayer de la serie temporal. En general, si una serie temporal se asemeja a una caminata aleatoria, los valores de la serie temporal no pueden predecirse con gran precisión.

- **Estacionariedad:** Una serie temporal es estacionaria cuando es estable a lo largo del tiempo, es decir, cuando la **media** y la **varianza** son **constantes** en el tiempo, además, **no presenta tendencia**. Por lo que una serie no estacionaria será aquella en la que su tendencia y/o variabilidad cambian en el tiempo. Los cambios en la media determinan una tendencia a crecer o decrecer a largo plazo, por lo que no oscilaría alrededor de un valor constante (que es la media) [10]. En la Figura 1 se muestra diferentes ejemplos gráficos de la estacionariedad:

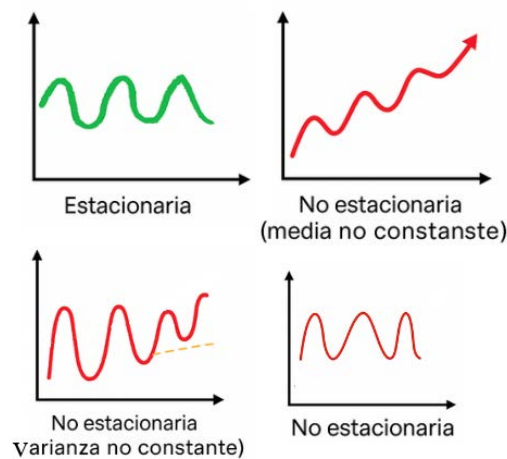


Figura 1 Ejemplos gráficos de estacionariedad [11]

Si en una serie estacionaria obtenemos 2 secuencias de datos consecutivos, con la misma cantidad de datos, aunque con el punto inicial diferentes, estas 2 secuencias de datos tendrán covarianzas iguales, por lo que esta serie debería llamarse **débilmente estacionaria** o **estacionaria en covarianza**, tal y como se muestra en la Figura 2 [12]:

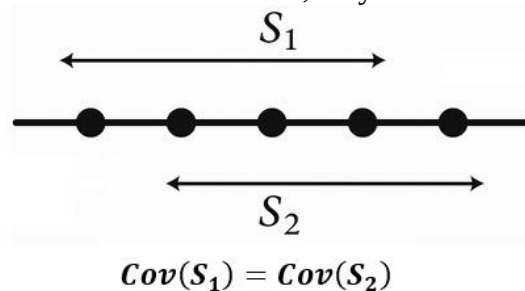


Figura 2 Estacionariedad en forma débil

- **Estacionalidad:** Una serie temporal se considera estacional cuando los eventos que ocurren en ella se repiten en intervalos regulares de tiempo. Es decir, patrones que se presentan cada día a la misma hora, o cada año en un mes específico. La estacionalidad es fácil de entender: ciertos productos tienen más ventas en épocas específicas, como helados

en verano o ropa de abrigo en invierno, como se puede mostrar gráficamente en la Figura 3 [13].

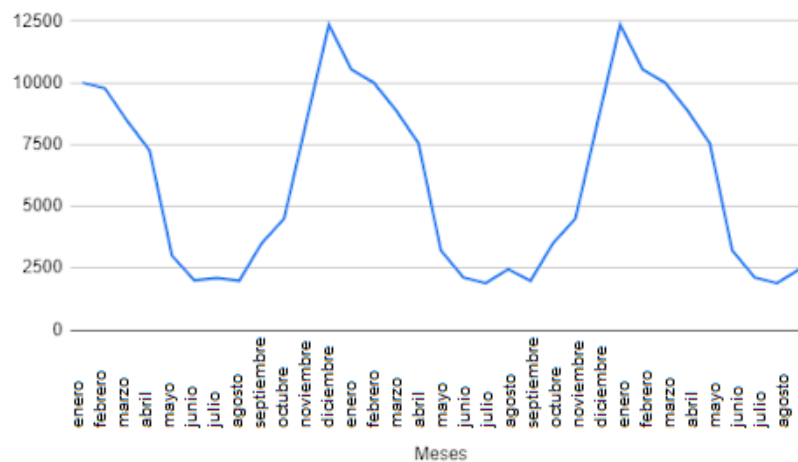


Figura 3 Ventas de helados en función de los meses [13]

- **Autocorrelación:** La autocorrelación en una serie temporal es la relación que existe entre los valores de la misma variable en diferentes momentos, de modo que un valor en el tiempo t está influido por valores anteriores, como $t-1$, $t-2$, etc. Este vínculo indica que las observaciones no son independientes y que datos cercanos en el tiempo tienden a parecerse. En muchos análisis, basta con medir la autocorrelación de retardo uno para detectar esta dependencia [14]. Si solo queremos determinar la relación directa entre la serie de tiempo y su versión retrasada, necesitamos calcular la **autocorrelación parcial**. La función de autocorrelación parcial es una medida de la correlación entre observaciones de una serie de tiempo que se encuentran separadas por k unidades de tiempo (y_t y y_{t-k}), después de ajustarse para la presencia de los demás términos de desfase más corto (y_{t-1} , y_{t-2} , ..., y_{t-k-1}) [15].

2.3 Desafíos de las series temporales

El análisis de series temporales es una herramienta poderosa utilizada para anticipar patrones y comportamientos futuros a partir de datos históricos. No obstante, como ocurre con cualquier técnica analítica, presenta ciertas dificultades y limitaciones que deben ser consideradas. En esta sección, se abordan algunos de los principales retos que enfrentan los analistas al aplicar este enfoque con fines predictivos:

- **No estacionariedad:** Uno de los supuestos clave en el análisis de series temporales es la estacionariedad, es decir, que las propiedades estadísticas de la serie (como la media y la varianza) se mantienen constantes a lo largo del tiempo. Sin embargo, los datos reales normalmente presentan no estacionariedad, lo que complica la aplicación de modelos clásicos. En muchos escenarios del mundo real, los datos suelen mostrar tendencias, estacionalidades u otras formas de comportamiento no estacionario. Para abordar este problema, se recurre a técnicas como la diferenciación, la transformación logarítmica o la eliminación de tendencias, aunque su efectividad puede variar según el caso. *Ejemplo:* Si analizamos las ventas mensuales de una cadena minorista y observamos una tendencia creciente sostenida, los datos no son estacionarios. En tales situaciones, puede ser necesario transformar la serie o emplear modelos más sofisticados que toleren dicha tendencia. Otro ejemplo es el análisis de precios de acciones que, debido a una tendencia prolongada al alza o baja, requieren transformación para ser modelados de forma efectiva.

- **Datos faltantes:** Las series temporales con frecuencia presentan valores faltantes, lo cual puede dificultar el análisis. Estas ausencias pueden deberse a errores de medición, problemas técnicos, fallos en sensores o simplemente a la falta de recolección de datos en ciertos momentos. Tratar con valores ausentes requiere precaución, ya que puede afectar significativamente la precisión de los modelos. Entre las técnicas más utilizadas para imputación se encuentran: el llenado hacia adelante o hacia atrás, la media, la interpolación y métodos más avanzados como imputaciones múltiples. *Ejemplo:* Si en una serie diaria de temperatura faltan varios días por errores en los sensores, el modelo puede perder precisión si no se imputan o corrigen esos valores ausentes adecuadamente.
- **No estacionalidad o tendencia:** Las series temporales suelen contener patrones estacionales y tendencias de largo plazo que deben ser correctamente identificados y modelados. La estacionalidad se refiere a fluctuaciones periódicas previsibles (por ejemplo, mensuales o anuales), mientras que la tendencia refleja una dirección general creciente o decreciente a lo largo del tiempo. Ignorar estos componentes puede llevar a predicciones inexactas. *Ejemplo:* En una serie sobre reservas hoteleras, si hay un aumento recurrente en verano y una caída en invierno, es esencial modelar esa estacionalidad para obtener predicciones realistas. Para tratar estos patrones se puede recurrir a técnicas de descomposición estacional, ajuste estacional u otros métodos especializados.
- **Valores atípicos y anomalías:** Las series temporales están expuestas a la presencia de valores extremos que se desvían notablemente del patrón general. Estos *outliers* pueden distorsionar la estructura estadística del conjunto de datos y comprometer la precisión de los modelos. Es crucial detectarlos adecuadamente e interpretar si se trata de errores o de eventos significativos. *Ejemplo:* Un pico inusual en el precio de una acción debido a una noticia inesperada puede actuar como valor atípico. Si el modelo no lo considera, las predicciones resultantes pueden ser erróneas. Existen diversas estrategias para tratar estas observaciones, incluyendo pruebas estadísticas, inspección visual o algoritmos específicos de detección.
- **Sobreajuste (*overfitting*):** Esta limitación ocurre cuando un modelo se ajusta demasiado a los datos históricos, capturando incluso el ruido y las fluctuaciones aleatorias. Aunque el modelo puede mostrar un excelente desempeño sobre los datos de entrenamiento, su capacidad de generalizar a nuevos datos se ve afectada. Para evitarlo, se pueden aplicar técnicas de regularización, reducir la complejidad del modelo o utilizar conjuntos de validación adecuados para evaluar su rendimiento real. *Ejemplo:* Un modelo que aprende perfectamente los precios diarios de una acción durante un mes puede fallar al predecir el mes siguiente porque se ha ajustado demasiado a los datos pasados, incluyendo el ruido.
- **Longitud de los datos:** La cantidad de datos disponibles es un factor crítico en el análisis de series temporales. Series muy cortas pueden dificultar la detección de patrones estacionales o la validación del modelo, mientras que series excesivamente largas pueden aumentar la complejidad computacional o requerir técnicas específicas para segmentación o resumen de datos. Es importante encontrar un equilibrio entre longitud y calidad de los datos para lograr resultados confiables. *Ejemplo:* Si solo se cuenta con tres meses de datos para predecir el comportamiento anual de ventas, el modelo no podrá capturar patrones estacionales importantes y sus resultados serán poco fiables.

Si bien el análisis de series temporales es una metodología muy útil para prever comportamientos futuros, es imprescindible reconocer y gestionar adecuadamente sus limitaciones. Problemas como la no estacionariedad, la presencia de tendencias y la estacionalidad, la presencia de valores atípicos o anomalías, los datos faltantes, el riesgo de sobreajuste y la limitación en la longitud de las observaciones, pueden comprometer la validez de los resultados. La aplicación de técnicas adecuadas para abordar estos retos puede mejorar significativamente la calidad de los pronósticos y fortalecer la toma de decisiones basada en datos [16].

3 MÉTODOS DE PREDICCIÓN: TEORÍA Y APLICACIÓN

3.1 Introducción a modelos predictivos

Cuando analizamos una serie temporal, lo que realmente nos interesa es poder mirar hacia adelante y predecir el futuro. Para eso, se usan los datos que ya se tienen, tratando de encontrar un modelo que encaje bien con ellos. Ese modelo será como una herramienta que, entendiendo el comportamiento pasado, nos ayude a estimar valores futuros. Pero para que funcione, necesitamos conocer bien la serie: si tiene tendencias, si sigue un patrón que se repite o si cambia mucho de un momento a otro. Sin ese análisis previo, corremos el riesgo de elegir un modelo equivocado y obtener predicciones poco útiles.

También es muy importante la calidad de la información que usamos. No se trata solo de tener muchos datos, sino de que sean correctos y representen bien la realidad. Con datos limpios y bien organizados, es mucho más fácil encontrar un modelo que realmente funcione, ya sea uno sencillo o algo más avanzado. Cuantos más detalles tengamos sobre la serie y su contexto, mejores serán las predicciones y más confianza tendremos para tomar decisiones basadas en ellas.

3.2 Modelos estadísticos tradicionales

3.2.1 Modelo AR: Autorregresivo

En series temporales a menudo se observan similitudes entre valores pasados y valores presentes, esto se debe a que se encuentra una correlación dentro de dichos datos.

La autocorrelación describe la relación entre una variable consigo misma, y nos va a definir el llamado modelo autorregresivo (AR).

Un modelo AR es una herramienta estadística utilizada en el análisis de series temporales que aprovecha los valores pasados para estimar valores futuros. En esencia, describe el valor presente de la serie como una combinación lineal de ciertas observaciones anteriores, intentando así reflejar la relación que existe entre los datos actuales y los pasados, tal y como se muestra en la Ecuación 2, donde x_t es el periodo actual, x_{t-1} el periodo anterior, c una constante, ϕ el coeficiente que el modelo aprende de los datos y ε_t un residuo aleatorio, además, $|\phi| < 1$:

$$x_t = c + \phi_1 \cdot x_{t-1} + \varepsilon_t \quad (2)$$

En cuanto al residuo, significa que generalmente van a ser diferencias impredecibles, y no se tienen en cuenta, son impredecibles (es la diferencia entre nuestra predicción y el valor correcto).

Para un modelo AR de orden n , su fórmula sería como se muestra en la Ecuación 3 :

$$AR(n) = c + \varphi_1 \cdot x_{t-1} + \varphi_2 \cdot x_{t-2} + \dots + \varphi_n \cdot x_{t-n} + \varepsilon_t \quad (3)$$

Si se retrocede demasiado se puede estar recogiendo ruido que ya no es relevante para el término de la predicción real, ya que más coeficientes habrá que estimar, siendo el modelo más complejo y con más posibilidades de que haya coeficientes no significativos [17].

3.2.2 Modelo MA: Medias Móviles

Un método para identificar la tendencia subyacente en un conjunto de datos (suavizando los picos y valles) es el modelo de medias móviles.

Una media móvil consiste en una secuencia de promedios calculados a partir de datos históricos. Se pueden obtener para distintos intervalos de tiempo, como un promedio móvil de tres meses, de siete días o de cuatro trimestres. El procedimiento básico de cálculo es el mismo en todos los casos [18].

El modelo de medias móviles (MA) es un enfoque estadístico utilizado para modelar y pronosticar datos de series temporales. Suponer este modelo implica asumir que el valor actual de la serie depende de forma lineal tanto de las perturbaciones aleatorias presentes como de las ocurridas en el pasado. A diferencia de los modelos autorregresivos (AR), que utilizan valores pasados de la propia serie, los modelos MA se basan en errores pasados, tal y como se muestra en la Ecuación 4, donde x_t es el periodo actual, ε_t el residuo en el periodo actual, ε_{t-1} el residuo en el periodo anterior, c una constante y θ el coeficiente que el modelo aprende de los datos, además, $|\theta| < 1$:

$$x_t = c + \theta_1 \cdot \varepsilon_{t-1} + \varepsilon_t \quad (4)$$

En un modelo MA(q), el parámetro ‘q’ indica el número de términos de error rezagados que se consideran en la ecuación del modelo. Al incorporar errores de pasos de tiempo anteriores, los modelos MA(q) ayudan a suavizar las irregularidades en los datos y ofrecen una mejor comprensión de las tendencias subyacentes.

La forma general de un modelo MA(q) se expresa como se muestra en la Ecuación 5 [19]:

$$MA(q) = c + \varphi_1 \cdot x_{t-1} + \theta_1 \cdot \varepsilon_{t-1} + \theta_2 \cdot \varepsilon_{t-2} + \dots + \theta_q \cdot \varepsilon_{t-q} + \varepsilon_t \quad (5)$$

Se van a ir a ir generando residuos ' ε' ', que se convertirán en una nueva variable, y los tendremos para cualquier periodo.

3.2.3 Modelo ARMA: Autorregresivo de Medias Móviles

Un modelo Autorregresivo de Media Móvil (ARMA) combina las características de los modelos AR y MA para representar de manera más completa la dinámica de una serie temporal. De forma general, un modelo ARMA(p, q) se expresa como se muestra en la Ecuación 6, donde x_t, x_{t-1} son los valores del periodo actual y periodo anterior, $\varepsilon_t, \varepsilon_{t-1}$ son los errores del periodo actual y periodo anterior, φ_p representa qué parte de los valores del periodo anterior es relevante para explicar el actual, θ_q representa qué parte de los errores del periodo anterior es relevante para explicar el actual y c es una constante, además, $|\varphi| < 1$ y $|\theta| < 1$:

$$ARMA(p, q) = c + \varphi_1 \cdot x_{t-1} + \dots + \varphi_p \cdot x_{t-p} + \theta_1 \cdot \varepsilon_{t-1} + \dots + \theta_q \cdot \varepsilon_{t-q} + \varepsilon_t \quad (6)$$

Se denomina ARMA (p, q), donde ‘p’ representa los términos autorregresivos y ‘q’ los términos de media móvil. Por lo tanto, el modelo tiene en cuenta tanto los valores de periodos anteriores como los residuos de esos mismos periodos, combinando ambas fuentes de información para describir mejor la serie temporal. El modelo ARMA no funciona para series **no estacionarias** porque asume que las propiedades estadísticas, como la media y la varianza, permanecen constantes en el tiempo [20].

3.2.4 Modelo ARIMA: Autorregresivo Integrado de Medias Móviles

ARIMA son las siglas de *Autoregressive Integrated Moving Average* (media móvil integrada autorregresiva) y es una técnica utilizada para el análisis de series temporales y la estimación de valores futuros [21].

Se aplicará el método ARIMA si los datos de la serie temporal **no son estacionarios**. Como ya se explicó al previamente, se conocen los conceptos de autorregresión y medias móviles. La “I” en ARIMA significa “Integrado” y hace referencia al proceso de hacer la diferencia de los datos no estacionarios un cierto número de veces (d) para lograr que se vuelvan estacionarios. De esta forma, es posible transformar datos no estacionarios en datos estacionarios, facilitando su análisis.

Esta parte integrada del modelo explica el número de **diferencias no estacionarias** que debemos examinar para establecer estacionariedad. Va a transformar los datos para que sean estacionarios, ya que el modelo ARMA no funciona bien para datos no estacionarios.

Para el modelo ARIMA, lo importante va a ser la diferencia entre los valores en el periodo ‘t’ y el anterior que le precede. Básicamente, estima los saltos entre los valores, tal y como se muestra en la Figura 4:

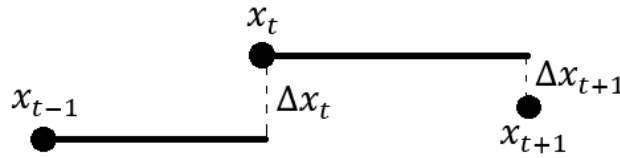


Figura 4 Saltos entre los valores de la serie temporal

En cada paso va a estimar los saltos entre los valores. En cierto sentido supone una especie de elemento aleatorio para el conjunto de datos, que modela las diferencias de un periodo al siguiente, en lugar de los valores por sí mismo. Entonces, para el modelo, los datos van a ser las diferencias de un periodo a otro $\Delta x_t, \Delta x_{t-1} \dots$

El propósito es garantizar la estacionariedad, sin importar qué tipo de datos son los subyacentes, integrando ‘d’ veces la serie temporal para construir una nueva serie temporal y luego ajustarla con un modelo ARMA(p,q). Por lo que el método ARIMA, realmente es un ARMA aplicado a una nueva serie temporal que ha sido obtenida transformando o integrando la serie de tiempo original.

La ecuación de un modelo simple ARIMA(1,1,1) se muestra en la Ecuación 7, donde x_t, x_{t-1} son los valores del periodo actual y periodo anterior, $\epsilon_t, \epsilon_{t-1}$ son los errores del periodo actual y periodo anterior, ϕ_1 representa qué parte de los valores del periodo anterior es relevante para explicar el actual, θ_1 representa qué parte de los errores del periodo anterior es relevante para explicar el actual, c es una constante y $\Delta_t = x_t - x_{t-1}$ es la disparidad entre valores en periodos consecutivos, además, $|\phi| < 1$ y $|\theta| < 1$:

$$\Delta x_t = c + \phi_1 \cdot \Delta x_{t-1} + \theta_1 \cdot \epsilon_{t-1} + \epsilon_t \quad (7)$$

Por lo que el modelo ARIMA(p,d,q) es un ARMA(p,q) para una nueva serie temporal (que ahora es estacionaria) generada mediante integración.

Si solo hay 1 integración, es decir, si el orden de integración en ARIMA es 1 (ARIMA(p,1,q)), la nueva serie temporal representa las diferencias entre los valores. Si hay 2 integraciones (ARIMA(p,2,q)), la nueva serie temporal sería las diferencias entre la serie de diferencias entre los valores, y a esta nueva serie temporal se le aplicaría el modelo ARMA(p,q).

Por lo que un modelo ARIMA(p,2,q) sobre los valores es lo mismo que un ARIMA(p,1,q) sobre la diferencia entre los valores.

La generalización del modelo ARIMA(p, d, q), expresado en la Ecuación 8, donde si $d = 1$: $\Delta_t = x_t - x_{t-1}$ y si $d = 2$: $\Delta_t = x_t - x_{t-2}$, permite extender la formulación básica de ARIMA(1, 1, 1) a un número cualquiera de términos autorregresivos y de media móvil, así como a un grado de integración 'd' determinado. De esta forma, la expresión resultante combina en una sola ecuación los efectos de la autorregresión, la media móvil y la integración, capturando con mayor flexibilidad la dinámica de la serie temporal:

$$ARIMA(p, d, q) = c + \varphi_1 \cdot \Delta x_{t-1} + \dots + \varphi_n \cdot \Delta x_{t-n} + \theta_1 \cdot \varepsilon_{t-1} \dots + \theta_q \cdot \varepsilon_{t-q} + \varepsilon_t \quad (8)$$

Es importante saber que en cada integración se va a perder una observación, es decir, si se hace más de una integración, se perderían varias observaciones, porque al tratar con valores integrados, como se tienen que coger valores pasados y hacer esos cálculos, se va a perder información, siendo esto inevitable.

Por ejemplo, si se va a hacer una integración teniendo el día 1, día 2, día 3, etc..., que son nuestros datos originales, si se va a hacer una integración para el primer día, se consideraría el día anterior para considerar esa integración, pero como no hay día anterior porque ese es el primer día de los datos, entonces el resultado de esa primera integración ya le faltaría el dato el primer día. Para el día 2 y el día 3 sí, porque para el día 2 sí se tiene el día anterior, que es el día 1, y para el día 3 se tiene el día 2, que es el día anterior, pudiéndose calcular. Por lo que el día 1 se tiene un valor faltante, y si hiciéramos otra integración, ser perdería el día 2, perdiendo un valor más por cada integración al principio de la serie [22].

3.2.5 Modelos ARMAX y ARIMAX: ARMA y ARIMA con variables exógenas

El modelo ARIMAX, cuyas siglas corresponden a *AutoRegressive Integrated Moving Average with eXogenous inputs* (Media Móvil Integrada Autorregresiva con entradas exógenas), es una versión avanzada del modelo ARIMA. Su principal diferencia radica en que incorpora variables exógenas, es decir, factores externos que pueden influir en la serie temporal analizada. Al integrar esta información adicional, el modelo puede aprovechar más contexto y, con ello, mejorar de forma significativa la precisión de las predicciones.

La ecuación del modelo ARMAX se muestra en la Ecuación 9, donde $\beta \cdot Y$ son las variables exógenas:

$$ARMAX(p, q) = c + \beta \cdot Y + \varphi_1 \cdot x_{t-1} + \dots + \varphi_n \cdot x_{t-n} + \theta_1 \cdot \varepsilon_{t-1} \dots + \theta_q \cdot \varepsilon_{t-q} + \varepsilon_t \quad (9)$$

La ecuación del modelo ARIMAX se muestra en la Ecuación 10 donde $\beta \cdot Y$ son las variables exógenas:

$$ARIMAX(p, d, q) = c + \beta \cdot Y + \varphi_1 \cdot \Delta x_{t-1} + \dots + \varphi_n \cdot \Delta x_{t-n} + \theta_1 \cdot \varepsilon_{t-1} \dots + \theta_q \cdot \varepsilon_{t-q} + \varepsilon_t \quad (10)$$

Las variables exógenas son predictores o factores externos que no forman parte directa de la serie temporal, pero que pueden influir de manera importante en ella. Al incluir estas variables, los modelos ARMAX y ARIMAX logran un análisis más completo y una capacidad de predicción más precisa [23].

3.2.6 Modelo SARIMAX: ARIMAX con componente estacional

Entre los diferentes enfoques disponibles, el modelo SARIMAX (*Seasonal Autoregressive Integrated Moving Average* con variables exógenas) destaca como una herramienta potente para

modelar y predecir tanto las tendencias como las variaciones estacionales en datos temporales, incorporando además variables externas al análisis para mejorar la precisión de las predicciones.

Todos los modelos que se han visto hasta ahora van a tener un equivalente estacional: SAR, SMA, SARMA, SARIMA, SARIMAX. La ‘S’ que se está agregando para representar el aspecto estacional.

Tal y como se expuso previamente, la estacionalidad ocurre cuando ciertos patrones aparecen periódicamente, tal y como se muestra en la Figura 5:

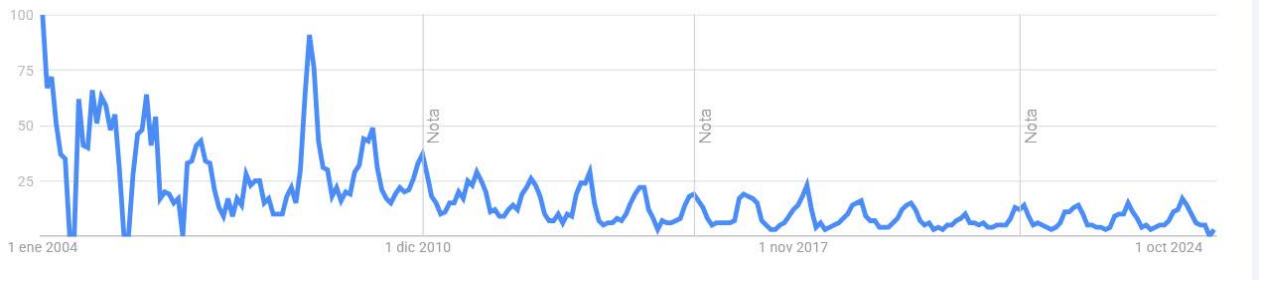


Figura 5 “Snowboarding” en Google Trends

Por lo tanto, un modelo simple autorregresivo (AR) no describirá bien estos datos. Si se observa la ecuación del modelo AR de orden 1 ($x_t = c + \phi_1 \cdot x_{t-1} + \varepsilon_t$), con x_{t-1} solo se está considerando el valor del periodo anterior para estimar el periodo actual, por lo que, si tenemos una frecuencia mensual por ejemplo, donde se observa un pico y un valle en ciertos meses como puede ser diciembre en el ejemplo de la Figura 5, esos valores estarían estimados solo a partir de noviembre, que es el periodo anterior, infraestimando lo que va a haber en diciembre. Por otro lado, los valores de febrero podrían estar sobreestimándose en base a los valores de enero (ya que hay muchas más búsquedas en enero que en febrero normalmente).

En este caso se necesitaría incluir a los valores registrados durante el periodo en el que se suele hacer ese deporte anterior al periodo presente. Eso significa no coger solo el mes anterior, sino incorporar esos meses de invierno del año anterior, o de años anteriores.

Es como tener una serie de tiempo que se extiende más en el pasado que la original, es decir, la serie de tiempo original contiene valores separados por 1 mes, mientras que la estacional, contendrá valores entre 12 meses atrás incluso 24 meses atrás, tal y como se muestra en la Ecuación 11:

$$x_t = c + \phi_1 \cdot x_{t-1} + \phi_{12} \cdot x_{t-12} + \phi_{24} \cdot x_{t-24} + \varepsilon_t \quad (11)$$

Si el enfoque parte de medias móviles, es decir de los errores, en este caso habría que incluir los errores de esos periodos atrás, tal y como se muestra en la Ecuación 12:

$$x_t = c + \theta_1 \cdot \varepsilon_{t-1} + \theta_{12} \cdot \varepsilon_{t-12} + \theta_{24} \cdot \varepsilon_{t-24} + \varepsilon_t \quad (12)$$

El modelo SARIMAX tendría los siguiente parámetros: (p,d,q)(P,D,Q,s), siendo:

- (p,d,q): Órdenes no estacionales.
- (P,D,Q,s): Órdenes estacionales.
- P,D,Q: Variaciones estacionales de (p,d,q).
- s: Parámetro estacional.

El parámetro estacional ‘s’ va a ser una cantidad de periodos que tienen que pasar para que la tendencia reaparezca. Por ejemplo, si la frecuencia de los datos son horas, y hay un patrón que se repite todos los días, tendríamos s=24. Si s=1, cada periodo es su propio ciclo, por lo tanto, se

supondrá que no hay estacionalidad. Por lo que si $s=1$, no se está agregando el componente de estacionalidad al modelo, y el patrón sería simplemente autocorrelación.

Si tenemos que implementar por ejemplo el modelo SARIMAX(1,0,2)(2,0,1,5), donde se ha quitado la parte de integración con $d=0$ y $D=0$ y sin tener en cuenta variables exógenas (sería un modelo SARMA), la ecuación quedaría como se muestra en la Ecuación 13, donde $\Phi_1 \cdot (x_{t-5} + \varphi_1 \cdot x_{t-6}) + \Phi_2 \cdot (x_{t-10} + \varphi_1 \cdot x_{t-11})$ es la componente AR, $\Theta_1 \cdot (\varepsilon_{t-5} + \theta_1 \cdot \varepsilon_{t-6} + \theta_2 \cdot \varepsilon_{t-7})$ es la componente MA, $(x_{t-5} + \varphi_1 \cdot x_{t-6})$ representa 1 retraso en AR(1,0,2) con estacionalidad 5 (2,0,1,5), Φ_2 es el segundo coeficiente en AR(2,0,1,5), $(x_{t-10} + \varphi_1 \cdot x_{t-11})$ representa estacionalidad 5 (2,0,1,5), por lo que se le resta 5 a x_{t-5} , Θ_1 es el primer coeficiente de MA(2,0,1,5) y $(\varepsilon_{t-5} + \theta_1 \cdot \varepsilon_{t-6} + \theta_2 \cdot \varepsilon_{t-7})$ representa 2 retrasos en MA (1,0,2):

$$x_t = c + \varphi_1 \cdot x_{t-1} + \theta_1 \cdot \varepsilon_{t-1} + \theta_2 \cdot \varepsilon_{t-2} + \Phi_1 \cdot (x_{t-5} + \varphi_1 \cdot x_{t-6}) + \Phi_2 \cdot (x_{t-10} + \varphi_1 \cdot x_{t-11}) + \Theta_1 \cdot (\varepsilon_{t-5} + \theta_1 \cdot \varepsilon_{t-6} + \theta_2 \cdot \varepsilon_{t-7}) + \varepsilon_t \quad (13)$$

Aunque en este caso se ha presentado un modelo sin integración ni variables exógenas, la misma estructura puede ampliarse para incluir estos elementos cuando la naturaleza de la serie lo requiera. De esta forma, es posible adaptar el modelo para trabajar con datos no estacionarios mediante la integración, así como incorporar factores externos que puedan influir en la serie temporal, aumentando su capacidad de análisis y predicción en muchos casos [24].

3.3 Modelos de aprendizaje automático

3.3.1 Introducción al aprendizaje supervisado en series temporales

La Inteligencia Artificial (IA) ha incrementado su relevancia en el panorama empresarial actual gracias a la integración de capacidades en máquinas. La IA describe la inteligencia demostrada por las máquinas, en contraste con la inteligencia natural presente en los seres vivos.

Hoy en día, la IA es ampliamente conocida gracias a las técnicas de Aprendizaje Automático (*Machine Learning*, ML). En este campo, el rendimiento de una máquina depende directamente de su capacidad de aprendizaje; por lo tanto, mejorar su desempeño está siempre ligado a perfeccionar su forma de aprender. Estos comportamientos de aprendizaje se inspiran en el conocimiento derivado de la inteligencia de los organismos vivos.

En el camino hacia el éxito del aprendizaje automático, el dato es el recurso esencial. El ML destaca por su implementación a través de tres enfoques de aprendizaje: **supervisado**, **no supervisado** y **por refuerzo**.

- En el aprendizaje supervisado, se busca encontrar la relación entre variables independientes (atributos de entrada) y variables dependientes (atributos objetivo o de salida).
- En el aprendizaje no supervisado, a diferencia del supervisado, el objetivo es formar grupos o clústeres a partir de los datos, sin conocer de antemano las etiquetas.
- El aprendizaje por refuerzo se basa en un sistema de recompensas o retroalimentación para guiar el comportamiento del modelo.

En el contexto de este trabajo, orientado a la predicción de precios bursátiles, el enfoque de mayor interés será el **aprendizaje supervisado**. Este método permite utilizar datos históricos (como precios pasados, volúmenes de negociación e indicadores técnicos) para entrenar modelos capaces de estimar valores futuros. La razón de esta elección radica en que el aprendizaje supervisado trabaja con pares de entrada-salida conocidos, lo que facilita la identificación de relaciones entre variables pasadas y el comportamiento futuro del activo, incrementando así el potencial predictivo del modelo [25].

En el aprendizaje supervisado, se construye un conjunto de entrenamiento de observaciones $T = (x_i, y_i)$ con $i=1, \dots, N$, $N=1$. Los valores de entrada observados (x_i, y_i) se introducen también en un sistema, conocido como algoritmo de aprendizaje, el cual produce salidas $f(x_i)$ en respuesta a dichas entradas.

Este algoritmo tiene la capacidad de ajustar su relación entrada/salida en función de las diferencias $y_i - f(x_i)$ entre las salidas reales y las generadas por el modelo.

Al finalizar el proceso de entrenamiento, se espera que las salidas producidas por el sistema y las reales sean lo suficientemente similares como para resultar útiles ante cualquier conjunto de entradas que probablemente se presenten en la práctica [26].

En el aprendizaje supervisado, los algoritmos entrenan un modelo para detectar patrones en un conjunto de datos que cuenta con características (*features*) y etiquetas (*labels*). Una vez entrenado, el modelo puede predecir las etiquetas de nuevos datos a partir de sus características, tal y como se muestra en la Figura 6 [27]:

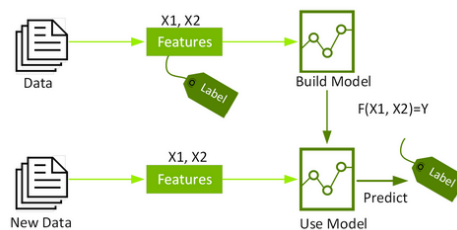


Figura 6 Etiquetado de datos con aprendizaje supervisado [27]

3.3.2 Árbol de decisiones

Un árbol de decisiones se trata de un diagrama de flujo que empieza con una idea principal y luego se ramifica según las consecuencias de las decisiones. Se denomina “árbol de decisiones” por su semejanza con un árbol con muchas ramas.

Este método se usa para realizar un análisis que consiste en delinear de forma gráfica los posibles resultados, costes y consecuencias de una decisión compleja. Se puede usar un árbol de decisiones para calcular el valor esperado de cada resultado en función de las decisiones tomadas y sus respectivas consecuencias. Luego, se pueden comparar los diferentes resultados para determinar rápidamente cuál será el mejor plan de acción. También se puede usar un árbol de decisiones para resolver problemas, gestionar costes e identificar oportunidades.

En este método se suelen utilizar los siguientes términos:

- **Ramificaciones alternativas:** son líneas que se desprenden de una decisión. Cada rama indica un posible resultado o decisión que se deriva de la decisión inicial.
- **Nodos de decisión:** indican una decisión que se está tomando en el árbol. Todos los árboles de decisiones comienzan con un nodo de decisión.
- **Nodos de oportunidad:** muestran varios resultados posibles.
- **Nodos terminales o finales:** indican el resultado final de una decisión.

En un diagrama de árbol de decisiones se combinan estos símbolos con notas que detallan cada decisión y sus posibles resultados, así como cualquier dato relevante para explicar las ganancias o las pérdidas. Se puede dibujar el árbol de decisiones de forma manual o se puede usar una herramienta de diagrama de flujo para crearlo digitalmente [28].

El árbol de decisiones construye un modelo que predice la etiqueta evaluando una secuencia de reglas del tipo “si (condición), entonces (acción); de lo contrario (acción alternativa)” sobre las características, buscando el menor número de reglas para maximizar la probabilidad de acierto. Pueden usarse para clasificación (predecir una categoría) o regresión (estimar un valor numérico continuo). Por ejemplo, un árbol puede estimar el precio de una vivienda (etiqueta) a partir del tamaño y el número de habitaciones (características), como se muestra en la Figura 7 [27]:

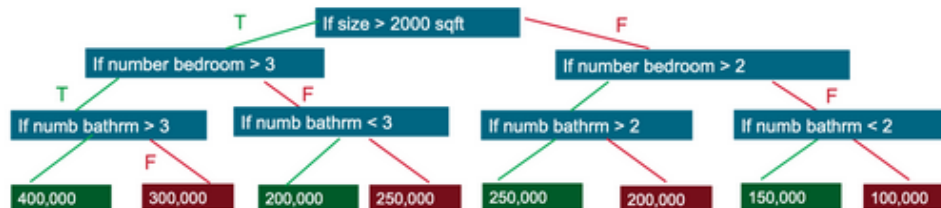


Figura 7 Ejemplo de árbol de decisión para estimar el precio de una vivienda [27]

3.3.3 Random Forest

Random Forest es un método de *ensemble* de aprendizaje automático utilizado tanto para clasificación como para regresión, que combina múltiples árboles de decisión entrenados sobre distintas muestras aleatorias para mejorar la precisión y reducir el sobreajuste. Se basa en el principio de *bagging* y en la selección aleatoria de predictores en cada división, lo que produce modelos más robustos y menos correlacionados.

Un método de *ensemble* es una técnica que combina varios modelos base para producir una predicción final más precisa y estable que la de un único modelo. La idea es aprovechar la diversidad de modelos para reducir errores, ya sea promediando sus predicciones (*bagging*), corrigiéndose de forma secuencial (*boosting*) o combinando distintas estrategias.

El *bagging* es una técnica de *machine learning*, que consiste en crear muchos modelos, generalmente árboles o modelos pequeños o *weak learners*, para hacer la predicción como media de todos estos modelos.

Los *Random Forest* están compuestos por un gran número de árboles de decisión individuales que funcionan como un *ensemble*. Se consideran clasificadores no paramétricos, ya que no requieren establecer parámetros de aprendizaje. Representan una mejora sobre el método *bagging* (acrónimo de *bootstrap aggregating*, técnica que entrena múltiples modelos sobre distintas muestras aleatorias con reemplazo) al introducir descorrelación entre ellos.

El *bootstrapping* es una técnica estadística que consiste en crear muchas muestras, del mismo tamaño que la población original, mediante muestreo aleatorio con reemplazo. Gracias al *bootstrapping* se cuenta con muchas muestras diferentes sobre las que se puede trabajar, sin necesidad de tener que buscar más datos. Estas muestras pueden servir tanto como para estimar un estadístico como, en el caso de *Random Forest*, para entrenar un árbol de decisión.

Al igual que en *bagging*, se construye un gran número de árboles de decisión a partir de muestras *bootstrap* (conjuntos de datos generados al seleccionar aleatoriamente observaciones del conjunto original, permitiendo repeticiones). Cada árbol individual genera una predicción para la clase, y la clase que recibe más votos se convierte en la predicción final del modelo.

La diferencia clave con *bagging* radica en el proceso de división de los nodos: en *bagging*, se consideran todos los predictores en cada división, mientras que en *Random Forest* se selecciona un subconjunto aleatorio de predictores como candidatos para la división, generalmente calculado como la raíz cuadrada del número total de predictores.

Aunque esta elección aleatoria de predictores pueda parecer contra intuitiva, promediar resultados de árboles no correlacionados reduce la varianza mucho más que promediar árboles altamente correlacionados. Por lo que en *Random Forest* los árboles son entrenados con muestras diferentes gracias al *bagging* y, además, utilizan conjuntos distintos de características al realizar sus predicciones [29].

La estructura básica de un *Random Forest* puede representarse como se muestra en la Figura 8:

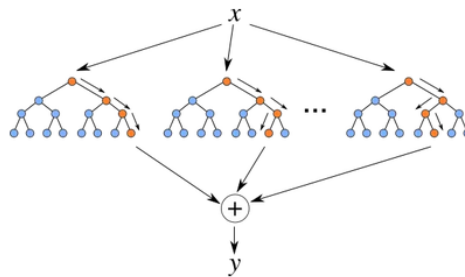


Figura 8 Estructura general de un modelo Random Forest [30]

Random Forest es uno de los modelos de *Machine Learning* más conocidos y utilizados. Es un modelo que tiene muchas ventajas: sirve tanto para clasificación como regresión, es robusto a *outliers*, no tiende al *overfitting* y suele tener buena capacidad predictiva.

Como ya se ha mencionado, *Random Forest* se basa en árboles de decisión, los cuales eligen las divisiones (*splits*) midiendo la impureza de los datos. En cada nodo, el algoritmo prueba distintas variables y diferentes puntos de corte para cada una. Cada corte divide los datos en dos grupos, se calcula la impureza de cada grupo y luego la impureza total como una media ponderada. Este proceso se repite para todas las variables y se elige el corte que produzca el grupo más “puro”. Después, el procedimiento se repite de forma recursiva hasta completar el árbol.

Los árboles de decisión presentan una limitación importante: si no se podan o no se restringe su profundidad, tienden a sobreajustar (*overfitting*); pero, si se podan en exceso o se limita demasiado su crecimiento, pueden incurrir en subajuste (*underfitting*), reduciendo así su capacidad predictiva.

En conclusión, *Random Forest* es un modelo potente, versátil y elegante, capaz de superar varias limitaciones de los árboles de decisión gracias al *bootstrapping* y a la selección aleatoria de variables. No obstante, no es infalible: puede verse afectado por variables correlacionadas o por *overfitting*, y requiere seguir buenas prácticas como la limpieza de datos, el análisis de correlaciones y el ajuste de hiperparámetros [31].

3.3.4 Gradient Boosting Machines

El *gradient boosting* es uno de los algoritmos de aprendizaje automático más utilizados para trabajar con datos tabulares, que son aquellos organizados en filas y columnas, como en una hoja de cálculo o una base de datos, donde cada fila representa una observación y cada columna una característica o variable. Tiene la capacidad de identificar relaciones no lineales entre la variable objetivo y las características, y además ofrece gran versatilidad, ya que puede manejar valores faltantes, *outliers* y variables categóricas de alta cardinalidad (aquellas que pueden tomar un número muy grande de valores o categorías diferentes) sin requerir un tratamiento especial previo [32].

Una de sus implementaciones más conocidas es el *Gradient Boosting Decision Trees* (GBDT), que aplica esta técnica específicamente sobre árboles de decisión como modelos base, combinándolos de forma secuencial para mejorar el rendimiento predictivo. Ofrece entrenamiento en paralelo y es una de las herramientas más utilizadas para resolver problemas de regresión, clasificación y ranking.

El GBDT es un algoritmo de aprendizaje basado en un conjunto (*ensemble*) de árboles de decisión, similar al *Random Forest*, que se utiliza tanto para clasificación como para regresión. Los algoritmos de *ensemble* combinan múltiples modelos de aprendizaje automático para obtener un modelo final más preciso y robusto.

Tanto *Random Forest* como GBDT construyen un modelo formado por varios árboles de decisión, pero difieren en la forma en que estos árboles se entrenan y se combinan.

Random Forest utiliza una técnica llamada *bagging* para construir árboles de decisión completos en paralelo a partir de muestras *bootstrap* aleatorias del conjunto de datos. La predicción final se obtiene promediando las predicciones de todos los árboles.

El término *gradient boosting* proviene de la idea de *boosting* o potenciación: mejorar un modelo débil combinándolo con varios otros modelos débiles para obtener un modelo globalmente más fuerte. El *gradient boosting* es una extensión de esta técnica, en la que el proceso de generar modelos débiles de forma aditiva se formaliza como un algoritmo de descenso por gradiente sobre una función objetivo. En este método, cada nuevo modelo se entrena con resultados objetivo ajustados para minimizar los errores, y dichos objetivos se calculan a partir del gradiente del error (de ahí el nombre *gradient boosting*) con respecto a la predicción, tal y como se muestra en la Figura 9:

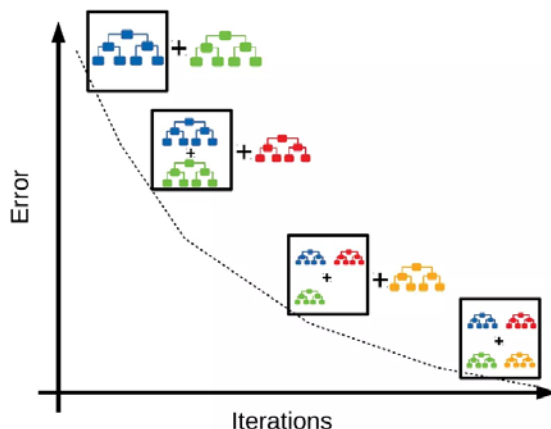


Figura 9 Algoritmo de *Gradient Boosting* [33]

Los GBDT entrenan de forma iterativa un conjunto (*ensemble*) de árboles de decisión poco profundos, donde cada iteración utiliza los residuos de error del modelo anterior para ajustar el siguiente. La predicción final es una suma ponderada de las predicciones de todos los árboles. En términos generales, el *bagging* de *Random Forest* tiende a reducir la varianza y el sobreajuste (*overfitting*), mientras que el *boosting* de GBDT busca reducir el sesgo y el subajuste (*underfitting*) [27].

3.3.4.1 XGBoost: Extreme Gradient Boosting

XGBoost (*Extreme Gradient Boosting*) es una implementación escalable y de alta precisión del algoritmo de *gradient boosting*, diseñada para maximizar el rendimiento y la velocidad de cómputo en modelos basados en árboles potenciados. A diferencia del GBDT tradicional, en el que los árboles se construyen de forma secuencial, XGBoost permite construirlos en paralelo. Utiliza una estrategia nivel por nivel (*level-wise*), analizando los valores del gradiente y empleando sumas parciales para evaluar la calidad de cada división posible en el conjunto de entrenamiento.

Cabe destacar que XGBoost y los modelos de aprendizaje automático que implementa ofrecen una combinación sobresaliente de rendimiento predictivo y velocidad de procesamiento en comparación con otros algoritmos [27].

3.3.4.2 LightGBM: Light Gradient Boosting Machine

LightGBM (*Light Gradient Boosting Machine*), desarrollado por Microsoft, es otro *framework* de *gradient boosting* que ha ganado gran popularidad gracias a su eficiencia y velocidad. Está diseñado para ser ligero en el uso de memoria y manejar de forma eficaz grandes conjuntos de datos con características dispersas. Emplea un enfoque de aprendizaje basado en histogramas y está optimizado para el entrenamiento distribuido, lo que lo hace especialmente adecuado para entornos de alto rendimiento.

LightGBM emplea una estrategia de crecimiento *leaf-wise* (hoja por hoja), expandiendo el árbol nodo por nodo, lo que le permite ser más eficiente en memoria y en el manejo de grandes volúmenes de datos, especialmente cuando son dispersos. Además, soporta variables categóricas de forma nativa, evitando la necesidad de *one-hot encoding* (técnica para convertir variables categóricas en columnas binarias (0/1), creando una columna independiente para cada categoría posible) y reduciendo así la dimensionalidad y el consumo de recursos.

Gracias a su enfoque de aprendizaje basado en histogramas, LightGBM logra una paralelización eficiente de forma horizontal, lo que se traduce en una mayor velocidad de entrenamiento. Esto lo convierte en una opción especialmente rápida en comparación con otros métodos de *gradient boosting*, manteniendo un rendimiento competitivo incluso en *datasets* muy grandes [34].

3.4 Modelos de aprendizaje profundo

3.4.1 Introducción al aprendizaje profundo y redes neuronales

Las redes neuronales son uno de los paradigmas de programación más fascinantes creados hasta la fecha. En el enfoque convencional de programación, se le indica al ordenador exactamente qué hacer, dividiendo los problemas grandes en muchas tareas pequeñas y bien definidas que la máquina puede ejecutar fácilmente. En cambio, en una red neuronal no se le dice al ordenador cómo resolver el problema: aprende a partir de datos observacionales y encuentra por sí misma la solución más adecuada.

La idea de aprender automáticamente a partir de datos resulta muy prometedora. Sin embargo, hasta 2006 no sabíamos cómo entrenar redes neuronales para superar a los métodos tradicionales, salvo en unos pocos problemas muy específicos. Lo que cambió en 2006 fue el descubrimiento de técnicas para entrenar las llamadas redes neuronales profundas (*deep neural networks*). Estas técnicas, ahora conocidas como aprendizaje profundo (*deep learning*), han seguido evolucionando y hoy en día alcanzan un rendimiento sobresaliente en áreas como visión por computadora, reconocimiento de voz y procesamiento del lenguaje natural. Grandes compañías como Google, Microsoft y Facebook las utilizan a gran escala.

Para definir qué es una red neuronal, es importante comenzar explicando un tipo de neurona artificial conocido como perceptrón. Este modelo fue desarrollado en las décadas de 1950 y 1960 por el científico Frank Rosenblatt, inspirado en trabajos previos de Warren McCulloch y Walter Pitts.

El funcionamiento de un perceptrón se basa en recibir varias entradas binarias $x_1, x_2 \dots$ y generar una única salida binaria. Este modelo evalúa las entradas según ciertos pesos asignados y, mediante una regla de decisión, determina si la salida será un 0 o un 1, tal y como se muestra en la Figura 10:

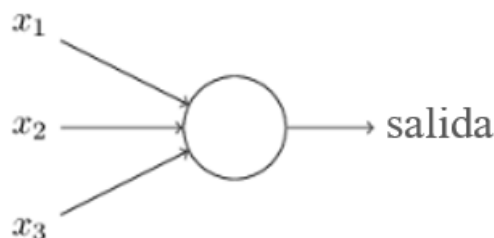


Figura 10 Esquema de un perceptrón con tres entradas y una salida [35]

En el ejemplo mostrado, el perceptrón cuenta con tres entradas: x_1, x_2, x_3 aunque en general puede tener más o menos. Rosenblatt propuso una regla sencilla para calcular la salida, introduciendo el concepto de pesos $\omega_1, \omega_2 \dots$, que son valores reales que indican la importancia de cada entrada en el resultado final. La salida de la neurona, que puede ser 0 o 1, se determina evaluando si la suma

ponderada $\sum_j \omega_j x_j$ es menor o mayor que un umbral. Este umbral, al igual que los pesos, es un valor real y constituye uno de los parámetros clave de la neurona. En términos algebraicos, el funcionamiento puede describirse de manera más precisa como se muestra en la Ecuación 14:

$$salida = \begin{cases} 0 & \text{if } \sum_j \omega_j x_j \leq \text{umbral} \\ 1 & \text{if } \sum_j \omega_j x_j > \text{umbral} \end{cases} \quad (14)$$

Un perceptrón es un modelo matemático que toma decisiones a partir de un conjunto de entradas binarias. Cada entrada está asociada a un peso, el cual representa la importancia relativa de esa variable en la decisión final. El perceptrón calcula la suma ponderada de todas las entradas y la compara con un valor umbral previamente definido.

Si la suma ponderada es mayor que el umbral, el perceptrón produce una salida de 1; en caso contrario, la salida será 0. Los pesos y el umbral son parámetros ajustables que determinan el comportamiento del modelo. Cambiar estos valores permite modificar las condiciones bajo las cuales el perceptrón genera una salida positiva o negativa.

De esta forma, el perceptrón puede adaptarse a diferentes patrones de decisión simplemente ajustando la importancia de cada entrada y el criterio de activación, funcionando como un clasificador lineal capaz de separar datos en dos categorías distintas.

El perceptrón no pretende ser un modelo completo del proceso de toma de decisiones humanas. Sin embargo, ilustra cómo es posible ponderar distintos tipos de evidencia para llegar a una decisión. Este enfoque permite asignar diferentes niveles de importancia a cada entrada y establecer un criterio objetivo para determinar el resultado.

Resulta razonable pensar que, al combinar múltiples perceptrones en una red más compleja, se puedan modelar procesos de decisión mucho más sutiles y sofisticados, capaces de abordar problemas más variados y con un mayor nivel de precisión, tal y como se muestra en la Figura 11:

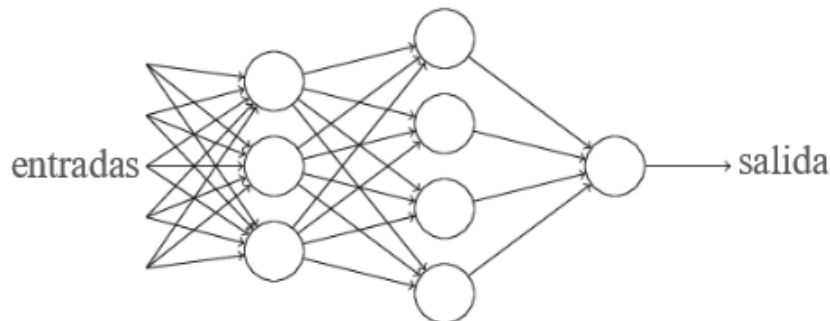


Figura 11 Estructura de una red neuronal [35]

En esta red neuronal, la primera columna de perceptrones (lo que se denomina la primera capa), toma decisiones simples evaluando la evidencia proveniente de las entradas. Los perceptrones de la segunda capa, en cambio, toman sus decisiones a partir de los resultados generados por la primera capa, lo que les permite operar en un nivel más complejo y abstracto. De forma similar, los perceptrones de la tercera capa pueden realizar evaluaciones aún más elaboradas. Así, una red de perceptrones con múltiples capas es capaz de llevar a cabo procesos de toma de decisiones sofisticados.

Cabe aclarar que, aunque cada perceptrón tiene una única salida, en el diagrama parece tener varias debido a que esa salida se utiliza como entrada para múltiples perceptrones de la siguiente capa. Este tipo de representación gráfica es más clara y práctica que dibujar una única línea que luego se ramifique.

En conclusión, las redes neuronales representan un poderoso paradigma de aprendizaje automático capaz de modelar relaciones complejas y no lineales mediante la interconexión de múltiples capas de

perceptrones u otras unidades artificiales. Su fortaleza radica en la capacidad de aprender de los datos de forma autónoma, ajustando sus pesos y parámetros para optimizar la toma de decisiones. Gracias a esta estructura jerárquica, pueden extraer y combinar patrones desde niveles simples hasta representaciones altamente abstractas, lo que las convierte en herramientas fundamentales para tareas como el reconocimiento de imágenes, el procesamiento del lenguaje natural y la predicción en entornos complejos [35].

Para la predicción de series temporales, especialmente en el contexto de precios bursátiles, se emplearán redes neuronales recurrentes (RNN) debido a su capacidad para manejar datos secuenciales y capturar dependencias a lo largo del tiempo.

3.4.2 RNN: Redes Neuronales Recurrentes

La predicción y clasificación de series temporales son problemas comunes y desafiantes en el aprendizaje profundo, ya que a menudo requieren identificar dependencias complejas entre entradas separadas en el tiempo. Las Redes Neuronales Recurrentes (RNN) tienen, en teoría, la capacidad de manejar estas dependencias temporales gracias a su memoria a corto plazo implementada mediante conexiones recurrentes (retroalimentación). Sin embargo, en la práctica, su entrenamiento resulta complicado cuando se necesita memoria a largo plazo.

Las RNN son un tipo de modelo de conexiones que poseen un estado interno o memoria a corto plazo gracias a sus conexiones recurrentes de retroalimentación, lo que las hace adecuadas para abordar problemas secuenciales como la clasificación, predicción y generación de secuencias [36].

Las RNN surgieron para superar las limitaciones de las redes neuronales tradicionales, como las Redes Neuronales *FeedForward* (FNN), en el procesamiento de datos secuenciales. Las FNN procesan cada entrada de manera independiente a través de varias capas ocultas, sin tener en cuenta el orden ni el contexto de las demás entradas. Esto impide que capturen las dependencias entre los datos, lo que las hace poco adecuadas para tareas que requieren procesar información en secuencia, como el modelado de lenguaje, la traducción automática, el reconocimiento de voz, el análisis de series temporales, entre otras aplicaciones.

Las RNN resuelven estas limitaciones incorporando conexiones recurrentes que permiten que la información fluya de un instante temporal al siguiente. Gracias a este mecanismo, mantienen una memoria interna en la que la salida de cada paso se reutiliza como entrada en el paso siguiente. Esto posibilita que el modelo aproveche la información previa, aprenda dependencias temporales y procese secuencias de longitud variable de manera efectiva.

En la Figura 12 se muestra la diferencia entre ambas redes neuronales:

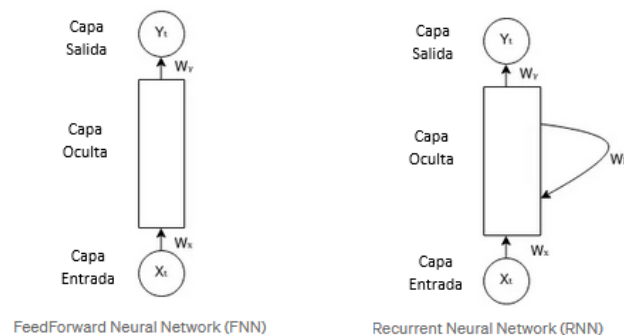


Figura 12 Analogía Red Neuronal FeedForward (FNN) y Red Neuronal Recurrente (RNN) [37]

Para una comprensión más clara del concepto de las RNN, en la Figura 13 se muestra su diagrama desplegado.

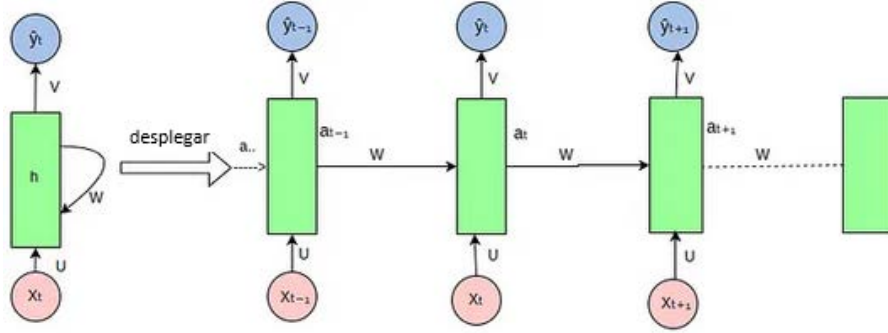


Figura 13 Despliegue de una Red Neuronal Recurrente (RNN) en el tiempo [37]

Una RNN recibe un vector de entrada ' x ' y genera un vector de salida ' y ' procesando los datos de manera secuencial, de izquierda a derecha. En cada paso de tiempo, actualiza su estado oculto y produce una salida. La red utiliza el mismo conjunto de parámetros en todos los pasos temporales, lo que implica que las matrices ' U ', ' V ' y ' W ' se mantienen constantes a lo largo de todo el proceso.

Aquí, ' U ' corresponde a los pesos que conectan la capa de entrada ' x ' con la capa oculta ' h '; ' W ' representa los pesos que conectan los estados ocultos entre sí; y ' V ' los que conectan la capa oculta ' h ' con la capa de salida ' y '. Esta compartición de parámetros permite a la RNN capturar de manera eficiente las dependencias temporales y procesar datos secuenciales reteniendo información de entradas pasadas en su estado oculto actual.

En cada paso de tiempo ' t ', el estado oculto ' a_t ' se calcula a partir de la entrada actual ' x_t ', el estado oculto previo ' a_{t-1} ' y los parámetros del modelo, tal como se muestra en la fórmula de la Ecuación 15, donde a_t es la salida generada por la capa oculta en el paso de tiempo t , x_t es la entrada en el paso de tiempo t , θ es el conjunto de parámetros entrenables (pesos y sesgos), U es la matriz de pesos que regula las conexiones desde la entrada hasta la capa oculta ($U \in \theta$), W es la matriz de pesos que regula las conexiones de la capa oculta consigo misma (conexiones recurrentes; $W \in \theta$), V es el peso asociado a la conexión entre la capa oculta y la capa de salida ($V \in \theta$), a_{t-1} es la salida de la capa oculta en el paso de tiempo $t-1$, b es el vector de sesgos para la capa oculta ($b \in \theta$) y f es la función de activación:

$$a_t = f(a_{t-1}, x_t, \theta) \text{ que se puede escribir también como } a_t = f(U \cdot x_t + W \cdot a_{t-1} + b) \quad (15)$$

La salida en cada paso de tiempo ' t ', denotada como ' y_t ' se calcula a partir de la salida del estado oculto ' a_t ' utilizando la fórmula de la Ecuación 16, donde $\hat{y}_t$ es la salida predicha en el paso de tiempo t , v es la matriz de pesos que regula las conexiones desde la capa oculta hasta la capa de salida y c es el vector de sesgos para la capa de salida:

$$\hat{y}_t = f(a_t, \theta) \text{ que se puede escribir también como } \hat{y}_t = f(V \cdot a_t + c) \quad (16)$$

Para un número finito de pasos de tiempo $T=4$, es posible desplegar el grafo computacional de una RNN, como se muestra en la Figura 13, aplicando la Ecuación 15 un total de $T-1$ veces: $a_4 = f(a_3, x_4, \theta)$, la cual se puede expandir como se muestra a continuación:

$$a_4 = f(U \cdot x_4 + W \cdot a_3 + b)$$

$$a_3 = f(U \cdot x_3 + W \cdot a_2 + b)$$

$$a_2 = f(U \cdot x_2 + W \cdot a_1 + b)$$

Pudiendo escribir la fórmula de salida según la Ecuación 16 como $\hat{y}_4 = f(V \cdot a_4 + \theta)$.

Lo que hace que una RNN aprenda es el proceso de **backpropagation**, que ajusta sus parámetros internos (pesos y sesgos) en función del error cometido entre la salida predicha y el valor objetivo real. El objetivo es minimizar la función de pérdida y, con ello, mejorar el rendimiento del modelo.

En las Redes Neuronales Recurrentes se utiliza una variante específica llamada *Backpropagation Through Time* (BPTT), en la que el error se propaga hacia atrás a lo largo de toda la secuencia, desde el último paso de tiempo hasta el inicial $t=1$.

Este procedimiento se compone de dos fases principales: la *forward pass*, en la que se calculan las salidas, y la *backward pass*, en la que se retropropaga el error para actualizar los parámetros.

- **Forward Pass:** durante la fase de propagación hacia adelante, la RNN procesa la secuencia de entrada a lo largo del tiempo, desde $t=1$ hasta $t=n$, donde n es la longitud de la secuencia de entrada. En cada paso de propagación hacia adelante, se realizan los cálculos de las fórmulas de la Ecuación 17, donde **tanh** es la función de activación que comprime los valores en el rango $[-1,1]$ y **softmax** es la función que convierte un vector de valores en una distribución de probabilidades:

$$\begin{aligned} a_t &= U \cdot X_t + W \cdot a_{t-1} + b \\ a_t &= \tanh(a_t) \\ \hat{y}_t &= \text{softmax}(V \cdot a_t + c) \end{aligned} \quad (17)$$

Después de procesar toda la secuencia, la RNN genera una secuencia de salidas predichas $\hat{y}_t = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_T]$, la función de pérdida MSE (Mean Squared Error Loss) se calcula comparando la salida predicha ' $\hat{y}$ ' en cada paso de tiempo con la salida objetivo real ' y ', y viene dada por la Ecuación 18:

$$L(y, \hat{y}) = \frac{1}{t} \cdot \sum (y_t - \hat{y}_t) \quad (18)$$

- **Backward Pass:** En el algoritmo *Backpropagation Through Time* (BPTT), esta fase consiste en calcular los gradientes de la función de pérdida con respecto a los parámetros de la red (U , W , V y los sesgos) a lo largo de cada paso de tiempo. Explorando el concepto de BPTT calculando los gradientes de la pérdida en el paso de tiempo $t=4$, en la Figura 14 se ilustra el proceso de *backpropagation* correspondiente a ese cuarto paso:

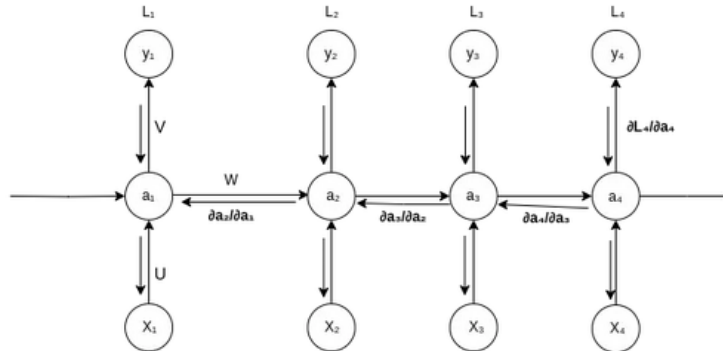


Figura 14 *Backpropagation* a través del tiempo (BPTT) en una RNN para $t = 4$ [37]

La pérdida ' L ' es una función del valor predicho ' $\hat{y}$ ', por lo que, aplicando la regla de la cadena, la derivada $\frac{\partial L}{\partial V}$ se puede expresar como:

$$\frac{\partial L}{\partial V} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial V}$$

Aplicando la regla de la cadena de derivadas, $\frac{\partial L}{\partial W}$ se puede expresar de la siguiente forma: La pérdida en el cuarto paso temporal depende de ' $\hat{y}$ ', ya que la pérdida se calcula como una función de ' $\hat{y}$ ', la cual a su vez está influenciada por el estado oculto ' a_4 ' del paso actual.

El estado ' a_4 ' está determinado tanto por ' W ' como por ' a_3 ', y ' a_3 ' se conecta con ' a_2 ' y con ' W '.

Asimismo, ' a_2 ' depende de ' a_1 ' y nuevamente de ' W ', lo que muestra cómo la influencia de ' W ' se propaga hacia atrás a través de los diferentes pasos temporales:

$$\frac{\partial L_4}{\partial W} = \left(\frac{\partial L_4}{\partial \hat{y}_4} \cdot \frac{\partial \hat{y}_4}{\partial a_4} \cdot \frac{\partial a_4}{\partial W} \right) + \left(\frac{\partial L_4}{\partial \hat{y}_4} \cdot \frac{\partial \hat{y}_4}{\partial a_4} \cdot \frac{\partial a_4}{\partial a_3} \cdot \frac{\partial a_3}{\partial W} \right) + \left(\frac{\partial L_4}{\partial \hat{y}_4} \cdot \frac{\partial \hat{y}_4}{\partial a_4} \cdot \frac{\partial a_4}{\partial a_3} \cdot \frac{\partial a_3}{\partial a_2} \cdot \frac{\partial a_2}{\partial W} \right) + \left(\frac{\partial L_4}{\partial \hat{y}_4} \cdot \frac{\partial \hat{y}_4}{\partial a_4} \cdot \frac{\partial a_4}{\partial a_3} \cdot \frac{\partial a_3}{\partial a_2} \cdot \frac{\partial a_2}{\partial a_1} \cdot \frac{\partial a_1}{\partial W} \right)$$

Del mismo modo, $\frac{\partial L}{\partial U}$ se puede expresar como:

$$\frac{\partial L_4}{\partial U} = \left(\frac{\partial L_4}{\partial \hat{y}_4} \cdot \frac{\partial \hat{y}_4}{\partial a_4} \cdot \frac{\partial a_4}{\partial U} \right) + \left(\frac{\partial L_4}{\partial \hat{y}_4} \cdot \frac{\partial \hat{y}_4}{\partial a_4} \cdot \frac{\partial a_4}{\partial a_3} \cdot \frac{\partial a_3}{\partial U} \right) + \left(\frac{\partial L_4}{\partial \hat{y}_4} \cdot \frac{\partial \hat{y}_4}{\partial a_4} \cdot \frac{\partial a_4}{\partial a_3} \cdot \frac{\partial a_3}{\partial a_2} \cdot \frac{\partial a_2}{\partial U} \right) + \left(\frac{\partial L_4}{\partial \hat{y}_4} \cdot \frac{\partial \hat{y}_4}{\partial a_4} \cdot \frac{\partial a_4}{\partial a_3} \cdot \frac{\partial a_3}{\partial a_2} \cdot \frac{\partial a_2}{\partial a_1} \cdot \frac{\partial a_1}{\partial U} \right)$$

Aquí se están sumando los gradientes de la pérdida a lo largo de todos los pasos temporales, lo que representa la diferencia clave entre el enfoque de *backpropagation* a través del tiempo (BPTT) y el *backpropagation* tradicional.

Las RNN tienen limitaciones importantes que afectan su capacidad de aprendizaje, especialmente durante el proceso de *backpropagation*. Uno de los problemas más conocidos es el **desvanecimiento del gradiente (*vanishing gradient*)**, que ocurre cuando los gradientes se vuelven demasiado pequeños al propagarse hacia atrás en el tiempo. Esto dificulta que la red capture dependencias a largo plazo de manera efectiva; aunque el entrenamiento pueda converger, puede tardar muchísimo en hacerlo. Por otro lado, el **problema de gradiente explosivo (*exploding gradient*)** surge cuando los gradientes crecen excesivamente, provocando inestabilidad numérica, desviando al modelo de la solución óptima y dificultando que la red converja hacia un mínimo global. Se puede ver gráficamente en la Figura 15:

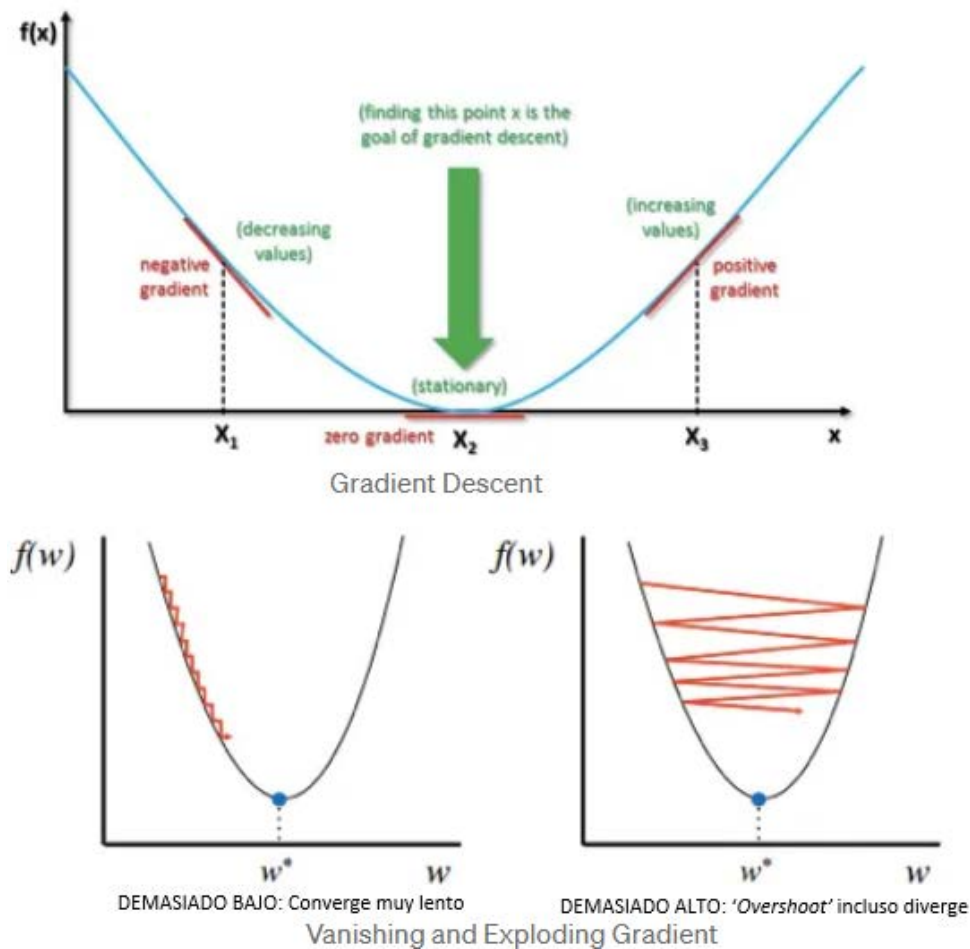


Figura 15 Descenso del Gradiente y Problemas de *Vanishing* y *Exploding Gradient* [37]

Este problema ha sido afrontado mediante enfoques como el uso de estructuras especializadas de neuronas, como las celdas de memoria de las redes **LSTM** (*Long Short-Term Memory*) y las **GRU** (*Gated Recurrent Unit*), que permiten manejar dependencias a largo plazo manteniendo un flujo constante del error hacia atrás; métodos de optimización de segundo orden, que preservan el gradiente estimando su curvatura; o inicializaciones aleatorias informadas, que facilitan el entrenamiento utilizando únicamente *momentum* y *gradient descent* estocástico [37].

3.4.3 LSTM: Long Short-Term Memory

La *Long Short-Term Memory* (LSTM) es un tipo especial de red neuronal recurrente (RNN) diseñada para superar el problema del *vanishing gradient* o *exploding gradient* que afecta a las RNN convencionales.

Gracias a ello, las LSTM pueden aprender de secuencias largas y con retardos temporales desconocidos entre eventos relevantes, resultando útiles para tareas de clasificación, procesamiento y predicción de series temporales.

Una red LSTM está compuesta por bloques LSTM que incluyen tres compuertas principales: entrada, salida y olvido. Estas compuertas permiten:

- **Recordar** un valor durante un tiempo arbitrariamente largo.
- **Olvidarlo** cuando deja de ser relevante.
- **Producirlo** como salida cuando es necesario.

Su diseño permite preservar el error durante el *backpropagation* a lo largo del tiempo y de las capas. Mantener un error más estable hace posible que la red siga aprendiendo incluso a lo largo de cientos o miles de pasos temporales.

Las LSTM manejan la información mediante una **celda con compuertas** que funciona de forma análoga a la memoria de un ordenador:

- Puede almacenar, escribir o leer información.
- Decide cuándo hacerlo mediante compuertas que se abren o cierran. A diferencia de la memoria digital, estas compuertas son análogas, implementadas con multiplicaciones elemento a elemento usando funciones sigmoideas (valores en el rango $[0, 1]$). Esto las hace diferenciables, lo que permite entrenarlas mediante *backpropagation*.

Cada compuerta filtra la información que recibe según su importancia, utilizando pesos propios que se ajustan durante el aprendizaje de la red. Así, la celda **aprende cuándo dejar pasar, bloquear o eliminar datos** a través del ciclo de predicción, *backpropagation* del error y ajuste de pesos con descenso de gradiente.

En las celdas LSTM, la suma y la multiplicación cumplen roles diferentes:

- En lugar de calcular el nuevo estado multiplicando el estado previo por la entrada, **los combina sumando**, lo que ayuda a mantener un error constante en *backpropagation* profundas.
- Distintos conjuntos de pesos se usan para las compuertas de entrada, salida y olvido.
- La compuerta de olvido actúa como una función identidad: si está abierta, el estado actual se multiplica por uno y se propaga al siguiente paso temporal.

Aunque su objetivo es conectar eventos distantes con una salida final, las LSTM necesitan la compuerta de olvido. Por ejemplo: Si se analiza un corpus de texto y se llega al final de un documento, es posible que el siguiente texto no tenga relación con el anterior. En ese caso, conviene reiniciar la celda de memoria a cero antes de procesar el nuevo documento.

El funcionamiento interno de una celda LSTM se ilustra en la Figura 16, donde se aprecian sus principales compuertas de control, mostrando cómo las compuertas de entrada, olvido y salida regulan el flujo de información a través de la celda:

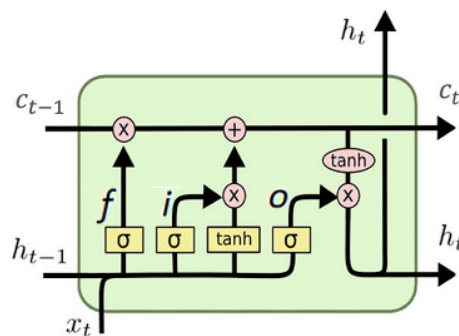


Figura 16 Celda LSTM con compuertas de entrada (i), olvido (f) y salida (o) [38]

Una celda LSTM no se limita a calcular la suma ponderada de las entradas y aplicar una función no lineal; en cambio, cada unidad j -ésima mantiene una memoria ' c_t^j ' en el instante ' t '.

La función de activación de la LSTM se presenta en la Ecuación 19, donde h_t^j es la salida o estado oculto de la unidad j en el instante de tiempo t dentro de una celda LSTM, o_t^j es el valor de la compuerta de salida para la unidad j en el instante t , que regula qué parte del estado de la celda se envía como salida, $\tanh(\cdot)$ es la función de activación tangente hiperbólica que introduce no linealidad de actualización, que limita valores entre -1 y 1 y c_t^j es el estado de la memoria de la celda para la unidad j en el instante t :

$$h_t^j = o_t^j \cdot \tanh(c_t^j) \quad (19)$$

La compuerta de salida ' o_t^j ' regula cuánto del contenido de la memoria se expone en la salida. Usando V_o como una matriz diagonal, su valor se calcula mediante la fórmula de la Ecuación 20, donde o_t^j es el valor de la compuerta de salida para la unidad j en el instante t, $\sigma(\cdot)$ es la función de activación sigmoide que limita valores entre 0 y 1, W_o es la matriz de pesos que conecta la entrada x_t con la compuerta de salida, x_t es el vector de entrada en el instante t, U_f es la Matriz de pesos que conecta el estado oculto previo h_{t-1} con la compuerta de salida, h_{t-1} es el estado oculto previo al instante t, V_o es la matriz diagonal de pesos que conecta el estado de la celda c_t con la compuerta de salida, c_t es el estado de la memoria de la celda en el instante t y j es el índice que identifica la neurona dentro de la capa oculta:

$$o_t^j = \sigma(W_o \cdot x_t + U_f \cdot h_{t-1} + V_o \cdot c_t)^j \quad (20)$$

La memoria ' c_t^j ' en una celda LSTM se actualiza combinando dos procesos: por un lado, se olvida parcialmente el contenido previo, y por otro, se incorpora una nueva memoria candidata ' $\tilde{c}_t^j$ '. El nivel de olvido lo regula la compuerta de olvido ' f_t^j ', mientras que la cantidad de nueva información añadida está controlada por la compuerta de entrada. Este mecanismo se describe en la Ecuación 21, donde f_t^j es el valor de la compuerta de olvido para la unidad j en el instante t, $\sigma(\cdot)$ es la función de activación sigmoide que limita valores entre 0 y 1, W_f es la matriz de pesos que conecta la entrada x_t con la compuerta de olvido, x_t es el vector de entrada en el instante t, U_o es la matriz de pesos que conecta el estado oculto previo h_{t-1} con la compuerta de olvido, h_{t-1} es el estado oculto previo al instante t, V_i es la matriz diagonal de pesos que conecta el estado previo de la celda c_{t-1} con la compuerta de olvido, c_{t-1} es el estado de la memoria de la celda en el instante anterior t-1, y j es el índice que identifica la neurona dentro de la capa oculta:

$$f_t^j = \sigma(W_f \cdot x_t + U_o \cdot h_{t-1} + V_i \cdot c_{t-1})^j \quad (21)$$

Y la cantidad de nueva memoria que se incorpora está controlada por la compuerta de entrada ' i_t^j ', que se calcula con la fórmula de la Ecuación 22, donde i_t^j es el valor de la compuerta de entrada para la unidad j en el instante t, $\sigma(\cdot)$ es la función de activación sigmoide que limita valores entre 0 y 1, W_i es la matriz de pesos que conecta la entrada x_t con la compuerta de entrada, U_i es la matriz de pesos que conecta la entrada x_t con la compuerta de entrada, h_{t-1} es el estado oculto previo al instante t, V_f es la matriz diagonal de pesos que conecta el estado previo de la celda c_{t-1} con la compuerta de entrada, c_{t-1} es el estado de la memoria de la celda en el instante anterior t-1 y j es el índice que identifica la neurona dentro de la capa oculta:

$$i_t^j = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + V_f \cdot c_{t-1})^j \quad (22)$$

Controlada por estas compuertas, la memoria existente se actualiza utilizando las fórmulas representadas en la Ecuación 23, donde $\tilde{c}_t^j$ es la memoria candidata de la unidad j en el instante t, $\tanh(\cdot)$ es la función de activación tangente hiperbólica que introduce no linealidad de actualización y que limita valores entre -1 y 1, W_c es la matriz de pesos que conecta la entrada x_t con la memoria candidata, x_t es el vector de entrada en el instante t, U_c es la matriz de pesos que conecta el estado oculto previo h_{t-1} con la memoria candidata, h_{t-1} es el estado oculto previo al instante t, c_{t-1}^j es el estado de la memoria de la celda en el instante anterior t-1, f_t^j es el valor de la compuerta de olvido para la unidad j en el instante t, c_{t-1}^j es el estado previo de la memoria de la unidad j, i_t^j es el valor de la compuerta de entrada para la unidad j en el instante t, y j es el índice que identifica la neurona dentro de la capa oculta:

$$\begin{aligned}\tilde{c}_t^j &= \tanh(W_c \cdot x_t + U_c \cdot h_{t-1})^j \\ c_t^j &= f_t^j \cdot c_{t-1}^j + i_t^j \cdot \tilde{c}_t^j\end{aligned}\quad (23)$$

De esta forma, se obtiene el estado de memoria actualizado que la LSTM empleará en los siguientes pasos de procesamiento.

Si se quiere visualizar el funcionamiento de una LSTM a lo largo de varios pasos temporales, se puede representar como una secuencia de celdas conectadas, donde cada una procesa una entrada en un instante específico y transmite sus estados a la siguiente. Este encadenamiento se ilustra en la Figura 17:

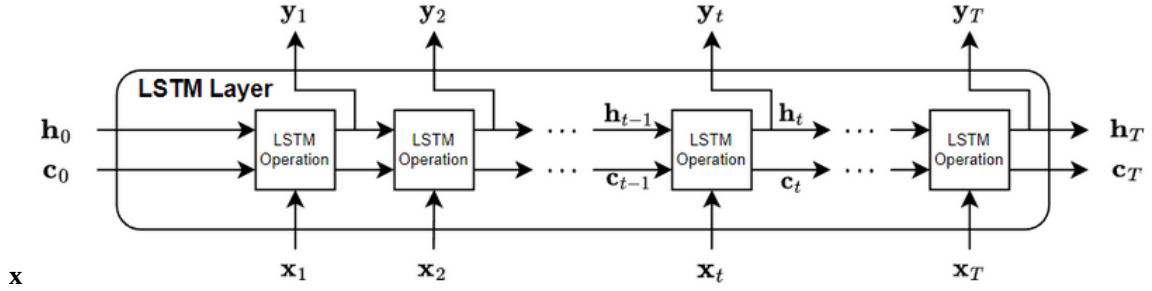


Figura 17 Estructura de una capa LSTM con múltiples celdas conectadas en secuencia [39]

En conclusión, las redes LSTM ofrecen una arquitectura altamente eficiente para el tratamiento de secuencias extensas, ya que permiten retener la información clave y descartar la que no aporta valor de forma controlada. Su estructura basada en compuertas de entrada, olvido y salida les brinda la capacidad de superar las limitaciones de las RNN convencionales, manteniendo la estabilidad del aprendizaje y garantizando predicciones precisas incluso cuando las relaciones temporales abarcan un gran número de pasos [40].

3.4.4 GRU: Gated Recurrent Unit

Las **redes Gated Recurrent Unit (GRU)** destacan en tareas de aprendizaje secuencial y superan los problemas de *Vanishing* y *Exploding Gradient* presentes en las RNN tradicionales al aprender dependencias a largo plazo [41].

GRU es una variante simplificada de la LSTM. En lugar de tener compuertas de olvido e ingreso separadas, las fusiona en una sola llamada “compuerta de actualización” y añade otra denominada “compuerta de reinicio”. Este diseño da como resultado un modelo más sencillo que las LSTM tradicionales y que, además, está ganando popularidad de forma creciente.

Una GRU, al igual que una LSTM, regula el flujo de información dentro de la unidad, pero sin contar con una celda de memoria independiente, tal y como se muestra en la Figura 16:

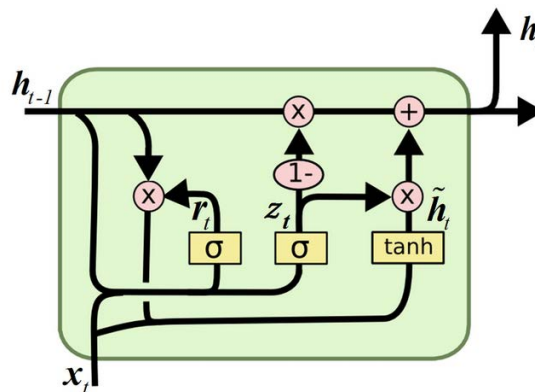


Figura 18 Celda de la GRU con compuertas de actualización (z) y reinicio (r) [42]

La activación ' h_t^j ' de la GRU en el instante ' t ' se calcula como una interpolación lineal entre la activación previa ' h_{t-1}^j ' y la activación candidata ' $\tilde{h}_t^j$ ', tal y como se muestra en la Ecuación 24, donde h_t^j es la activación o estado oculto de la unidad j en el instante de tiempo t , z_t^j es la compuerta de actualización, h_{t-1}^j es la activación o estado oculto de la unidad J en el instante de tiempo anterior $t-1$, $\tilde{h}_t^j$ es la activación candidata de la unidad J en el instante T , antes de combinarse con el estado previo mediante la compuerta de actualización, y j es el índice que identifica la neurona dentro de la capa oculta:

$$h_t^j = (1 - z_t^j) \cdot h_{t-1}^j + z_t^j \cdot \tilde{h}_t^j \quad (24)$$

La compuerta de actualización ' z_t^j ' determina qué proporción de la activación de la unidad se reemplaza o ajusta en cada instante, tal y como se muestra en la Ecuación 25, donde z_t^j es la compuerta de actualización, $\sigma(\cdot)$ es la función de activación sigmoide que limita valores entre 0 y 1, W_z es la matriz de pesos que conecta la entrada actual x_t con la compuerta de actualización, U_z es la matriz de pesos que conecta el estado oculto previo h_{t-1} con la compuerta de actualización, h_{t-1} es el estado oculto previo al instante t y j es el índice que identifica la neurona dentro de la capa oculta:

$$z_t^j = \sigma(W_z \cdot x_t + U_z \cdot h_{t-1})^j \quad (25)$$

La activación candidata ' $\tilde{h}_t^j$ ' se calcula de forma análoga al procedimiento usado para la compuerta de actualización, tal y como se muestra en la Ecuación 26, donde $\tilde{h}_t^j$ es la activación candidata de la unidad j en el instante de tiempo t , $\tanh(\cdot)$ es la función de activación tangente hiperbólica que introduce no linealidad de actualización y limita valores entre -1 y 1 W es la matriz de pesos que conecta la entrada actual x_t con la activación candidata, x_t es el vector de entrada en el instante t , U es la matriz de pesos que conecta el estado oculto previo con la activación candidata, r_t es el vector de la compuerta de reinicio que controla qué parte del estado previo se considera, $\cdot *$ representa la multiplicación elemento a elemento (Hadamard) entre r_t y h_{t-1} , h_{t-1} es el estado oculto previo al instante t y j es el índice que identifica la neurona dentro de la capa oculta:

$$\tilde{h}_t^j = \tanh(W \cdot x_t + U_z \cdot (r_t * h_{t-1}))^j \quad (26)$$

Cuando la compuerta de reinicio está desactivada $r_t^j = 0$, la unidad "olvida" su estado anterior, lo que equivale a procesar el primer símbolo de una secuencia de entrada. El valor de esta compuerta se calcula con la fórmula de la Ecuación 27, donde r_t^j es el valor de la compuerta de reinicio para la unidad j en el tiempo t , $\sigma(\cdot)$ es la función de activación sigmoide que limita valores entre 0 y 1, x_t es el vector de entrada en el instante t , W_r es la matriz de pesos que conecta la entrada x_t con la compuerta de reinicio, U_r es la matriz de pesos que conecta el estado oculto previo h_{t-1} con la compuerta de reinicio, h_{t-1} es el estado oculto previo al instante t y j es el índice que identifica la neurona dentro de la capa oculta:

$$r_t^j = \sigma(W_r \cdot x_t + U_r \cdot h_{t-1})^j \quad (27)$$

La compuerta de actualización ' z ' regula cuánto influye el estado anterior en el momento actual. Las unidades que manejan dependencias a corto plazo tienden a activar la compuerta de reinicio ' r ', mientras que aquellas con dependencias a largo plazo mantienen activa la compuerta de actualización ' z '.

Al igual que con LSTM, si se quiere visualizar el funcionamiento de una GRU a lo largo de varios pasos temporales, se puede representar como una secuencia de celdas conectadas, donde cada una procesa una entrada en un instante específico y transmite sus estados a la siguiente. Este encadenamiento se ilustra en la Figura 19:

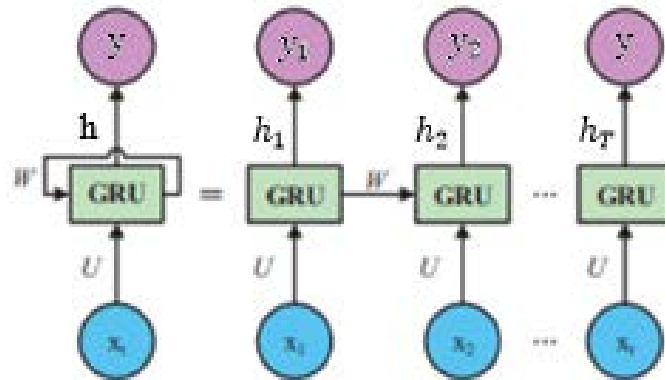


Figura 19 Estructura de una capa GRU con múltiples celdas conectadas en secuencia [43]

En conclusión, las redes GRU proporcionan una arquitectura simplificada y eficiente para el manejo de secuencias, al integrar las funciones de memoria y olvido en un conjunto reducido de compuertas. Esta estructura compacta, basada en las compuertas de actualización y reinicio, les permite mantener el flujo de información relevante y descartar la irrelevante con menor complejidad computacional que las LSTM. Gracias a ello, las GRU logran un aprendizaje estable y eficaz, ofreciendo un excelente rendimiento en tareas donde las dependencias temporales pueden abarcar múltiples pasos, pero se busca optimizar el tiempo de entrenamiento y el uso de recursos [40].

4 OBTENCIÓN Y PREPARACIÓN DE LOS DATOS

4.1 Descripción del proceso de preparación de datos

La preparación de los datos puede considerarse la etapa más importante dentro de un proyecto de predicción de series temporales. Aunque a menudo recibe menos atención en la literatura, suele ser la tarea más costosa y que consume más tiempo. La preparación de datos, también conocida como preprocesamiento, consiste en transformar los datos en bruto en un formato adecuado para el modelado. La mayoría de los modelos requieren que los datos estén en formato numérico, y prácticamente todas las implementaciones cumplen con esta expectativa. Por ejemplo, si los datos contienen categorías o etiquetas en lugar de números, será necesario transformarlos a valores numéricos. Además, muchos algoritmos tienen supuestos específicos sobre los datos, como la escala, la distribución de probabilidades o las relaciones entre variables de entrada, lo que obliga en ocasiones a realizar ajustes adicionales para que cumplan con estas condiciones.

La filosofía que subyace en la preparación de los datos es descubrir la mejor manera de exponer la estructura oculta del problema al algoritmo de aprendizaje. Esto implica un proceso iterativo de prueba y error, aplicando diferentes técnicas de transformación hasta encontrar aquellas que mejor funcionan. Aunque los algoritmos de aprendizaje automático que se emplean hoy en día han sido desarrollados durante décadas, el aspecto que cambia en cada proyecto es el conjunto de datos utilizado. En muchos casos, un investigador puede ser el primero en aplicar un conjunto de datos concreto para un problema de predicción, lo que convierte a la preparación de los datos en un paso crítico para obtener resultados válidos.

El reto principal radica en que cada conjunto de datos es único: difiere en el número de variables (decenas, cientos o miles), en el tipo de variables (numéricas, categóricas, binarias, etc.), en la escala de los valores, en la presencia de valores atípicos y en su distribución en el tiempo. Esta diversidad hace que hablar de “preparación de datos” de manera general sea un desafío. Por ello, algunos enfoques se centran en casos específicos, mientras que otros proponen métodos generales aplicables a distintos contextos.

En este trabajo, se adoptará una visión práctica y aplicada de la preparación de datos para predicción bursátil. El proceso incluirá desde la obtención de datos históricos y el cálculo de indicadores técnicos relevantes, hasta la limpieza y transformación de los mismos para ajustarlos a los requerimientos de los modelos estadísticos, de aprendizaje automático y de aprendizaje profundo que se implementarán. De esta manera, la preparación de los datos no será un paso aislado, sino un componente central que garantice la fiabilidad de las predicciones y la validez del análisis posterior.

Cada conjunto de datos es único y altamente específico para el problema en cuestión. Sin embargo, a pesar de esta diversidad, existen suficientes similitudes entre distintos proyectos de modelado predictivo como para identificar una secuencia general de pasos y tareas comunes que suelen repetirse. Este proceso permite situar la preparación de datos dentro de un marco más amplio, teniendo en cuenta tanto las características del proyecto antes de esta fase como la posterior evaluación de los algoritmos de aprendizaje automático.

En este trabajo se entiende la preparación de datos no como un paso aislado, sino como una parte esencial del ciclo completo de modelado predictivo. De este modo, se busca optimizar la exposición de la estructura subyacente del problema para que los algoritmos de aprendizaje puedan capturarla de manera más efectiva [44].

4.2 Fuentes y obtención de datos bursátiles

La calidad y procedencia de los datos bursátiles constituye un aspecto fundamental en cualquier proyecto de predicción financiera, ya que de ello depende la fiabilidad de los modelos y la posibilidad de aplicar posteriormente los resultados en escenarios reales de inversión. Existen múltiples fuentes abiertas de información financiera, entre las que destaca **Yahoo Finance**, ampliamente utilizada por la comunidad académica y profesional debido a su facilidad de acceso y disponibilidad gratuita. Sin embargo, este tipo de fuentes presentan ciertas limitaciones, como la menor precisión en datos intradía o no proporcionar los datos en tiempo real.

Con el objetivo de garantizar la máxima calidad de los datos y asegurar la posibilidad de trasladar los resultados del análisis a estrategias operativas en mercados reales, en este trabajo se utilizarán datos proporcionados por **Interactive Brokers (IB)**, uno de los brokers más reconocidos a nivel internacional. El acceso a la información se realizará a través de su **API oficial en Python**, en combinación con la plataforma **IB Gateway** o **Trader WorkStation (TWS)**, lo que permitirá obtener datos históricos de forma automática y organizada.

Interactive Brokers ofrece los precios **ajustados**, lo que significa que los valores históricos se modifican para reflejar ciertos eventos corporativos y así mantener la coherencia en la serie temporal. Este ajuste busca que los precios antiguos sean comparables con los actuales y que no aparezcan saltos artificiales que en realidad no corresponden a cambios en la valoración real de la empresa.

En este caso concreto, los precios están ajustados por *splits*, pero no por dividendos (Yahoo Finance ofrece precios ajustados por *splits* y por dividendos). El ajuste por *splits* significa que, cuando una empresa divide sus acciones (por ejemplo, 2:1), los precios históricos se recalculan para mantener la consistencia de la serie temporal. Esto evita saltos artificiales que podrían distorsionar los modelos de predicción. En cambio, la no corrección por dividendos implica que las caídas de precio tras el pago de dividendos permanecen reflejadas en la serie, lo que debe tenerse en cuenta en el análisis, ya que representa un factor real del mercado.

Además, se descargarán directamente los indicadores **OHLCV** (Open, High, Low, Close, Volume), que representan precios de apertura, máximos, mínimos, cierres y volúmenes de negociación. Estos cinco elementos constituyen la base de prácticamente cualquier análisis bursátil, ya que permiten tanto el cálculo de indicadores técnicos (como medias móviles, RSI o volatilidad) como la construcción de modelos de series temporales y de aprendizaje automático.

En este trabajo se emplearán **datos diarios**, ya que esta frecuencia aporta un equilibrio adecuado entre detalle y estabilidad en el análisis de series temporales. Al trabajar con precios diarios, es posible identificar patrones y tendencias significativas del mercado sin el exceso de ruido característico de los intervalos intradía, lo que favorece una visión más clara de la evolución de los precios. Además, los datos diarios permiten un proceso de toma de decisiones más informado y estratégico, al ofrecer un registro consistente de la dinámica bursátil que facilita reconocer tendencias, medir rendimientos y evaluar la eficacia de diferentes estrategias. De este modo, los datos diarios se convierten en una

herramienta clave para garantizar un análisis robusto y replicable, con mayor aplicabilidad práctica en la predicción y en el diseño de estrategias de inversión [45].

Se empleará un activo financiero individual como referencia para el estudio, lo que puede corresponder a una acción representativa del mercado. Por ejemplo, un caso habitual es el análisis de **AAPL (Apple Inc.)**, una de las compañías más negociadas en los mercados bursátiles internacionales. El uso de un activo concreto permite validar la metodología de predicción en un entorno realista, asegurando que los resultados obtenidos puedan extrapolarse posteriormente a otros instrumentos financieros con características similares.

Para facilitar la comprensión, en la Tabla 1 se resumen los principales indicadores bursátiles que serán utilizados como base del análisis. Estos datos constituyen la información fundamental de cualquier serie financiera, ya que reflejan tanto el comportamiento del precio en cada sesión como el volumen negociado. Su correcta interpretación y utilización es esencial para la construcción de modelos de predicción y para el cálculo posterior de indicadores técnicos que enriquecerán las series temporales:

	Fecha	Open	High	Low	Close	Volume
0	2023-01-02	150.0	153.0	149.5	152.0	1200000
1	2023-01-03	152.3	154.2	151.0	151.5	1350000
2	2023-01-04	151.8	152.5	150.2	152.2	1100000
3	2023-01-05	153.5	155.0	152.0	154.5	1450000
4	2023-01-06	154.0	156.0	153.2	155.5	1600000

Tabla 1 Estructura de los datos bursátiles descargados

4.3 Definición e importancia del análisis técnico

El análisis técnico es una de las metodologías más empleadas en los mercados financieros para estudiar y anticipar el comportamiento de los precios. Se basa principalmente en el estudio en profundidad de los gráficos, partiendo de la premisa de que toda la información relevante (económica, política, social o psicológica) ya se encuentra reflejada en el precio de los activos. A diferencia del análisis fundamental, que se enfoca en las causas que influyen en los mercados, el análisis técnico se centra en las consecuencias, es decir, en el movimiento del precio resultante de la interacción entre oferta y demanda. De este modo, se convierte en una herramienta práctica para interpretar lo que está ocurriendo en el mercado y evaluar hacia dónde puede dirigirse en el futuro.

Los analistas técnicos utilizan esta metodología para tomar decisiones de inversión basadas en la lectura de gráficos y en la identificación de patrones repetitivos que suelen anticipar los movimientos del mercado. Estos elementos permiten a los inversores disponer de un marco de referencia más objetivo para decidir si posicionarse al alza o a la baja en un activo financiero.

El análisis técnico se sostiene en tres principios fundamentales:

- El primero es que **todo está descontado en el precio**: se asume que cualquier información con capacidad de afectar a un activo ya está incorporada en su cotización.
- El segundo principio es que **los precios se mueven en tendencias**, lo cual convierte la identificación temprana de dichas tendencias en un aspecto clave para aprovechar al máximo los movimientos del mercado.

- El tercer principio sostiene que **la historia tiende a repetirse**, basado en el comportamiento recurrente de los participantes del mercado ante circunstancias similares, lo que genera patrones reconocibles en los gráficos.

En definitiva, el análisis técnico constituye una herramienta esencial en los mercados financieros, ya que ofrece a los inversores la posibilidad de adaptar sus estrategias de acuerdo con su propia filosofía de inversión y nivel de aversión al riesgo. Además, brinda la flexibilidad de trabajar en diferentes marcos temporales, desde minutos hasta meses, lo que lo hace aplicable tanto al trading de corto plazo como a la inversión más prolongada. Gracias a su enfoque visual e interpretativo, el análisis técnico permite a cualquier participante del mercado (independientemente de sus recursos) analizar con mayor claridad la acción del precio, mejorar la toma de decisiones y establecer bases sólidas para construir sistemas de inversión consistentes y disciplinados [46].

4.4 Construcción de indicadores técnicos relevantes

La aplicación del análisis y la predicción de series temporales a los mercados bursátiles resulta especialmente relevante dentro del Análisis Técnico, el cual se basa en valores históricos para construir indicadores que permiten identificar posibles tendencias en los precios de las acciones. En la práctica, la mayoría de estos indicadores suelen evaluarse de manera gráfica, y su impacto directo en la calidad de la predicción de precios no ha sido suficientemente estudiado hasta el momento [47].

En este trabajo se emplearán algunos de los indicadores técnicos más utilizados en los mercados financieros, ya que proporcionan señales valiosas para interpretar la dirección de los precios, identificar cambios de tendencia y establecer puntos de entrada y salida. Entre ellos, destaca la media móvil exponencial (EMA), que permite suavizar los precios y detectar tendencias con mayor sensibilidad a los cambios recientes del mercado. Asimismo, se incluirán otros indicadores como el índice de fuerza relativa (RSI) o las Bandas de Bollinger, que permiten detectar posibles rupturas de tendencia [48].

A partir de estos indicadores técnicos, se obtendrán variables que servirán para alimentar los modelos ya presentados en este trabajo, con el objetivo de realizar predicciones sobre la evolución de los precios. De este modo, indicadores como la EMA (*Exponential Moving Average*), el RSI (*Relative Strenght Index*) y las Bandas de Bollinger no solo se emplearán para el análisis descriptivo, sino que se integrarán directamente en el proceso de modelado, aportando información clave para evaluar su capacidad predictiva y su utilidad práctica en el diseño de estrategias de inversión.

En el **Anexo A.2** se muestran los indicadores técnicos que se van a emplear en el presente trabajo.

4.5 Limpieza y transformación de los datos

La detección y corrección de datos sucios constituye uno de los problemas más persistentes en el análisis de datos, ya que ignorarlo puede conducir a resultados imprecisos y a la toma de decisiones poco confiables. En los últimos años, tanto la industria como la academia han incrementado su interés en los desafíos de la limpieza de datos, desarrollando nuevas abstracciones, interfaces, enfoques de escalabilidad y técnicas estadísticas para mejorar estos procesos. Asimismo, se subraya la importancia de considerar cómo los procesos de limpieza influyen en los resultados posteriores del análisis estadístico, lo que abre nuevas perspectivas para desarrollar enfoques más robustos y adaptativos en el tratamiento de datos [49].

Para preparar los datos de este trabajo, el primer paso será realizar un proceso de limpieza eliminando los registros duplicados y descartando aquellas columnas que presenten valores 'NaN' (no es un número válido). Esta etapa es fundamental para asegurar que el análisis posterior no se vea sesgado por inconsistencias o datos incompletos. Asimismo, se debe considerar que los registros de fechas en la serie temporal no son completamente continuos, ya que existen saltos correspondientes a fines de semana y días festivos en los que no se registran operaciones en los mercados financieros.

Posteriormente, al generar variables derivadas de la serie temporal, como por ejemplo medias, porcentajes de cambio u otros indicadores técnicos, se producirán inevitablemente valores faltantes en los primeros periodos debido al propio cálculo de estas transformaciones. Estos registros incompletos serán eliminados con el fin de evitar inconsistencias y garantizar que las variables obtenidas puedan utilizarse correctamente en los modelos. De esta manera, se asegura que los datos finales empleados en el análisis sean coherentes, representativos y adecuados para el estudio planteado.

4.6 Normalización de los datos

Antes de introducir los datos en el modelo, es necesario aplicar un proceso de normalización. Este paso asegura que todas las variables estén en una misma escala y que ninguna predomine sobre las demás únicamente por la magnitud de sus valores. Por ejemplo, si el precio de una acción oscila entre 100 y 600, el RSI entre 0 y 100 y el volumen entre 1M y 100M, el modelo podría centrarse en el volumen solo por sus números más grandes, aunque no sea la variable más importante. Escalar los datos elimina este sesgo y equilibra la contribución de cada variable.

Las funciones de activación utilizadas en redes neuronales, como *sigmoid* o *tanh*, son especialmente sensibles al rango de entrada. Valores demasiado altos o bajos pueden saturarlas, lo que dificulta el aprendizaje. Al normalizar los datos a un rango controlado (habitualmente entre 0 y 1), el entrenamiento resulta más rápido y estable.

Otra ventaja clave es que se evitan problemas numéricos y de convergencia. En modelos como LSTM, los gradientes pueden volverse explosivos o desvanecerse cuando las variables no están equilibradas, lo que ralentiza el entrenamiento o incluso lo interrumpe.

Existen distintos métodos de escalado. **MinMaxScaler** es útil cuando se quiere mantener la forma de la serie y los datos están acotados y no excesivamente ruidosos. En cambio, **StandardScaler** se aplica mejor cuando los datos siguen una distribución normal y se busca centrar los valores en torno a cero. Para predicciones con series temporales, MinMaxScaler suele ofrecer mejores resultados.

No obstante, es necesario considerar ciertas limitaciones. La presencia de un solo **outlier** puede alterar significativamente la normalización, y si los datos nuevos caen fuera del rango original utilizado para escalar, será necesario recalcular el ajuste. Por ello, el tratamiento adecuado de los valores extremos y la revisión periódica del escalado forman parte fundamental del preprocesamiento.

En conclusión, **MinMaxScaler** constituye una herramienta fundamental en el preprocesamiento de datos, especialmente adecuada cuando se requiere mantener una escala consistente entre todas las variables. Su aplicación permite optimizar el rendimiento de modelos como las redes neuronales al mejorar la estabilidad del entrenamiento y la calidad de las predicciones. Teniendo en cuenta sus ventajas y limitaciones, será el método de normalización empleado en este trabajo [50].

5 METODOLOGÍA DE EVALUACIÓN DE MODELOS Y ESTRATEGIAS

5.1 La importancia del análisis temporal en la estimación de precios bursátiles

La predicción de los patrones futuros de movimiento de los precios bursátiles ha sido un tema ampliamente investigado en la literatura. Aunque los defensores de la hipótesis de mercado eficiente sostienen que es imposible anticipar con precisión los precios de las acciones, existen posturas que sostienen que, con una formulación y un modelado adecuados, es posible realizar predicciones con un nivel de exactitud considerablemente alto. Esta segunda corriente se ha enfocado en el desarrollo de modelos estadísticos, econométricos y de *machine learning* robustos, sustentados en una cuidadosa selección de variables y en la elección de formas funcionales o modelos de pronóstico apropiados. Parte de la literatura propone el uso de análisis y descomposición de series temporales para estimar valores futuros de las acciones, presentando múltiples enfoques que siguen esta metodología.

Asimismo, existe una gran cantidad de estudios dedicados al análisis técnico de los movimientos de precios, donde se emplean indicadores técnicos como Bandas de Bollinger, convergencia/divergencia de medias móviles (MACD) o índice de fuerza relativa (RSI) entre otros. También se utilizan patrones reconocidos como el “hombro-cabeza-hombro”, su versión invertida, triángulos, banderas o líneas de Fibonacci, los cuales ayudan a los operadores a tomar decisiones informadas. Estos métodos ofrecen representaciones visuales que facilitan a los inversores comprender la posible dirección del precio.

En este trabajo, se propone un enfoque granular para la predicción de precios y patrones de movimiento, combinando análisis técnico con métodos estadísticos, de *machine learning* y de *deep learning*. Se exploran diversas técnicas de clasificación y regresión para pronósticos a corto plazo, comparando su rendimiento en la estimación tanto del valor como de la dirección del precio de las acciones, así como los porcentajes de cambio de los precios.

El objetivo es utilizar datos de precios tomados en intervalos diarios de la Bolsa de Valores y desarrollar un marco robusto de pronóstico del movimiento de precios. Este enfoque granular busca modelar la dinámica inherente del mercado y ajustarse para ofrecer estimaciones inmediatas, enfocándose en un marco orientado al *trading* de corto plazo, más que en tendencias de largo plazo [51].

En el presente trabajo, se abordará el desarrollo de un sistema de predicción orientado al análisis bursátil, utilizando como base dos series temporales distintas: por un lado, los precios de cierre diarios y, por otro, los porcentajes de cambio de dichos precios. Los precios de cierre constituyen una serie con una clara tendencia, ya que reflejan la evolución a lo largo del tiempo del valor de una acción en el

mercado. En cambio, los porcentajes de cambio (calculados como la variación porcentual del precio de cierre respecto al día anterior) tienden a comportarse como series más cercanas a la estacionariedad, es decir, sin una tendencia definida y con una media y varianza relativamente constantes a lo largo del tiempo. Este tipo de series suele capturar la volatilidad y las fluctuaciones diarias del mercado, siendo de gran utilidad para el modelado de riesgos y la creación de estrategias de corto plazo.

Para la predicción de ambas series, se empleará un enfoque progresivo que abarcará desde métodos tradicionales de series temporales hasta arquitecturas avanzadas de *deep learning*. En primer lugar, se aplicarán modelos estadísticos, destacando **ARIMA** como una de las técnicas más consolidadas para el pronóstico univariante, y su versión extendida **SARIMAX**, que permite incorporar tanto la estacionalidad como variables exógenas que puedan influir en la evolución de la serie (por ejemplo, indicadores técnicos o factores macroeconómicos).

Posteriormente, se recurrirá a modelos de aprendizaje automático diseñados para problemas de regresión, comenzando con **Random Forest**, un método basado en conjuntos de árboles de decisión que destaca por su capacidad para manejar relaciones no lineales y datos con ruido. A continuación, se explorarán modelos de *Gradient Boosting Machines*, concretamente **XGBoost** y **LightGBM**, ampliamente reconocidos por su rendimiento en competiciones de *data science* y por su capacidad para optimizar predicciones a partir de grandes volúmenes de datos y características diversas.

En la etapa final, el trabajo se enfocará en el uso de modelos de aprendizaje profundo, específicamente redes neuronales recurrentes (RNN) y sus arquitecturas más avanzadas. Se comenzará con las **GRU** (Gated Recurrent Unit), que simplifican la estructura de las LSTM manteniendo un rendimiento competitivo en series con dependencias temporales. El proceso culminará con el uso de **LSTM** (Long Short-Term Memory), cuya arquitectura compleja, basada en compuertas de control de flujo de información, les permite capturar patrones de largo plazo y manejar eficazmente problemas de desaparición de gradiente. Este último modelo se considera el candidato con mayor potencial de éxito debido a su probada eficacia en contextos financieros y de series temporales.

La metodología propuesta comenzará con la obtención y preparación de los datos históricos, así como el cálculo de indicadores técnicos relevantes (*features*), entre los que se incluirán medias móviles, índices de volatilidad y RSI (Índice de Fuerza Relativa), entre otros. Posteriormente, se llevará a cabo un proceso exhaustivo de filtrado y limpieza para garantizar la calidad de los datos de entrada. A partir de ahí, se entrenarán los modelos mencionados, evaluando su capacidad predictiva mediante métricas adecuadas para cada tipo de problema (MAE, RMSE, MAPE, entre otras).

Con los resultados obtenidos, se elaborará un análisis estadístico comparativo que permita evaluar el rendimiento relativo de cada modelo y seleccionar los más prometedores. Además, se presentará un bloque de *data science* orientado al análisis comparativo, en el que se examinarán las diferencias de rendimiento entre arquitecturas, indicadores y estrategias propuestas. Esto permitirá identificar qué combinaciones ofrecen mejores resultados y, en consecuencia, definir líneas futuras de trabajo. El estudio concluirá con la exposición de las principales conclusiones y con la propuesta de mejoras y posibles líneas de investigación orientadas a incrementar la precisión de las predicciones y la eficacia de las estrategias generadas.

Las predicciones a más de un día vista, por ejemplo a 10 días, pueden abordarse de dos formas distintas. Una opción consiste en predecir de manera iterativa, estimando un día y utilizando ese resultado como entrada para predecir el siguiente. La alternativa es generar directamente una predicción conjunta de los 10 días futuros. En este trabajo se adopta este segundo enfoque, tratando la predicción de los días siguientes como un único bloque, lo que simplifica el proceso y evita la propagación de errores entre predicciones consecutivas.

Finalmente, se llevará a cabo un **backtesting**, es decir, la simulación de estrategias de inversión aplicadas retrospectivamente sobre datos históricos, con el objetivo de estimar su viabilidad y rentabilidad potencial si se aplicaran en un entorno de mercado real. Este proceso permitirá validar no solo la precisión de los modelos, sino también su aplicabilidad práctica en contextos de trading.

5.2 Evaluación de la Estrategia de Inversión y Definición de Criterios Operación

Para evaluar el rendimiento de los modelos propuestos en este trabajo, es necesario establecer criterios claros de entrada y salida en las operaciones, de forma que sea posible medir la efectividad de las predicciones en un contexto real de inversión. En este caso, las entradas se generarán en dos posibles momentos:

- Al cierre de la sesión bursátil (día de la predicción).
- En la apertura de la siguiente sesión (siguiente sesión al día de la predicción).

En cuanto al cierre de las operaciones, se considerarán 2 criterios de salida distintos, que son:

- Cerrar la operación en cuanto el precio supere un umbral porcentual previamente definido por la predicción (si predice que va a subir un 3% en los días que se establezca, hará una muestra buscando un 1%, otra buscando un 2% y otra buscando un 3% de subida).
- Mantenerla abierta hasta el cierre de sesión en el que se alcance el porcentaje definido por la predicción.

Por simplicidad, en este trabajo se mostrarán los resultados estableciendo **la entrada en el cierre del de la sesión bursátil** (día de la predicción) y **la salida en el cierre del día cuyo porcentaje de la operación supere el umbral porcentual**, o el último día de la predicción en caso de que no lo alcance.

Con el fin de cubrir diferentes escenarios de predicción, se generarán **10 ficheros de salida** correspondientes a movimientos en rangos desde $\pm 1\%$ hasta $\pm 10\%$. Esto significa que, si el modelo estima una subida del 5%, se abrirá una posición alcista en los ficheros correspondientes (los ficheros que representan un porcentaje del 1% al 5%); en caso de que no se alcance el objetivo marcado, la operación se cerrará en el último día de la ventana de predicción (por ejemplo, el décimo día si la muestra abarca 10 jornadas), calculando en ese momento el porcentaje final de la operación. Por el contrario, si el modelo determina que no habrá una subida superior a cierto umbral (ejemplo: 7%), entonces no se abrirán posiciones para los porcentajes superiores (8, 9 o 10%).

Además, se realizarán predicciones considerando **dos series de tiempo distintas**: los precios de cierre y los porcentajes de cambio diario (lo cual se asemeja más a una serie estacionaria). La comparación de ambas permitirá evaluar cuál de las dos configuraciones ofrece mayor consistencia y precisión a la hora de generar señales útiles para la toma de decisiones de inversión. El esquema de resultados derivado de este procedimiento quedará estructurado de tal forma que, para cada fichero de porcentaje, se disponga de la información necesaria para estudiar la efectividad de las señales emitidas por los modelos y valorar su aplicabilidad en la creación de estrategias de trading.

El procedimiento consiste en primer lugar, realizar una división del conjunto de datos en **entrenamiento, validación y prueba (test)**. Esta segmentación permite evaluar de forma estructurada la capacidad del modelo para generalizar a datos no vistos. Durante el ajuste de hiperparámetros, se reserva un porcentaje de los datos de entrenamiento para validación. El objetivo de esta validación es comparar la función de pérdida en entrenamiento y validación para detectar posibles casos de *overfitting*. Cuando ambas curvas presentan comportamientos similares, se asume que el modelo está aprendiendo patrones útiles; si en cambio la pérdida en entrenamiento sigue disminuyendo mientras que la de validación aumenta, se interpreta como una señal de sobreajuste, donde el modelo memoriza en lugar de generalizar.

En la Figura 20 se muestra un ejemplo de sobreajuste vs buen aprendizaje en el entrenamiento del modelo:

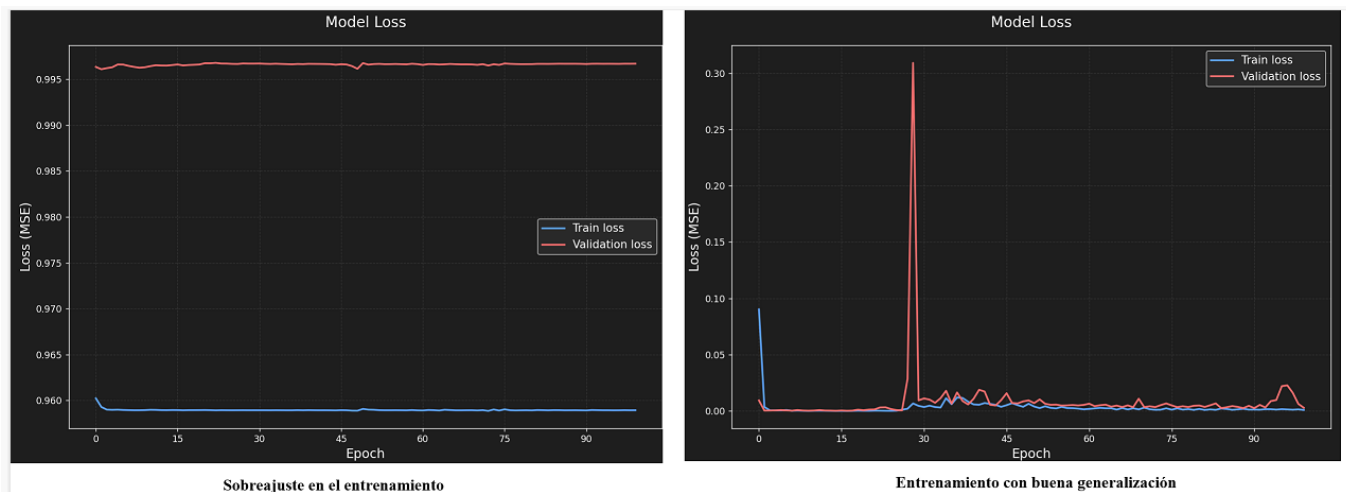


Figura 20 Comparación entre sobreajuste y entrenamiento con buena generalización.

Este análisis permite descartar configuraciones de hiperparámetros inadecuadas, por ejemplo, combinaciones de número de neuronas, tamaño del *batch*, o número de *epochs*, y concentrar los recursos computacionales en aquellas que presentan mayor potencial de generalización. El proceso se repite hasta encontrar un conjunto de parámetros candidatos a ofrecer un buen rendimiento.

Una vez determinados los posibles mejores hiperparámetros, se entrena el modelo sobre el conjunto de entrenamiento completo y se evalúa sobre el conjunto de prueba. Esta evaluación proporciona una medida objetiva del rendimiento del modelo. A partir de esta evaluación se genera un fichero de resultados que resume el comportamiento del modelo bajo condiciones controladas, permitiendo una primera validación de su utilidad práctica.

Con los hiperparámetros seleccionados, se procede a la fase central del trabajo: un **procedimiento iterativo de predicción en el tiempo**, diseñado para imitar cómo se utilizaría el modelo en un entorno real. Este enfoque consiste en lo siguiente: para cada fecha disponible, se entrena el modelo utilizando exclusivamente los datos históricos hasta ese día, y a continuación se realiza una predicción para el día siguiente. Una vez se conoce el valor real, este se incorpora al conjunto histórico, y se repite el proceso para la siguiente fecha. Este ciclo permite generar predicciones en tiempo simulado, siempre con la información que realmente estaría disponible en el momento de tomar la decisión.

Aunque sería posible guardar el modelo entrenado y reutilizarlo en cada iteración, en este trabajo se utiliza **Keras**, que guarda los pesos del modelo, pero **no conserva el estado interno de la red**, especialmente relevante en arquitecturas como **LSTM**, **GRU** o **RNN**. Dicho estado incluye, por ejemplo, los valores de las celdas de memoria o los estados ocultos que contienen información sobre las secuencias procesadas [52]. Esta limitación impide continuar el entrenamiento directamente desde el punto anterior, lo que hace que la opción más precisa sea **reentrenar el modelo desde cero en cada iteración**, utilizando todos los datos acumulados hasta la fecha actual.

Aunque este proceso es computacionalmente intensivo, garantiza un entorno de predicción más fiel a las condiciones reales del mercado, donde las decisiones se toman de forma secuencial y con información limitada al pasado. A lo largo de este proceso, se genera un nuevo fichero de resultados que contiene las predicciones diarias y los errores asociados, permitiendo analizar la evolución del rendimiento del modelo a lo largo del tiempo.

Finalmente, este fichero de predicciones servirá como base para la creación y evaluación de **estrategias de inversión automatizadas**, que se diseñarán en función de la calidad y la consistencia de las predicciones generadas. Al comparar las decisiones sugeridas por el modelo con la evolución

real del mercado, se podrán extraer conclusiones sobre su utilidad práctica, su robustez ante diferentes escenarios y su posible integración en sistemas de trading algorítmico.

Por lo que, tanto la evaluación sobre el conjunto de prueba (*test*) como el procedimiento de iteración diaria generan sus respectivos ficheros de resultados, que contienen las predicciones del modelo y las métricas asociadas. No obstante, debido a las exigencias computacionales del enfoque iterativo, se utilizarán inicialmente los resultados obtenidos mediante el conjunto de prueba para explorar de forma más rápida y eficiente múltiples combinaciones de hiperparámetros. Estos ficheros permiten una primera comparación entre modelos y facilitan la selección de posibles configuraciones. Sin embargo, las estrategias de inversión finales se diseñarán a partir de los resultados generados por la iteración diaria de fechas, ya que este enfoque reproduce de forma más realista cómo funcionaría el modelo en un entorno real, haciendo predicciones para cada día únicamente con los datos que estaban disponibles hasta ese momento, tal como ocurriría en la práctica. Por tanto, estos resultados constituyen la base más adecuada para evaluar el rendimiento práctico del sistema y diseñar estrategias aplicables a escenarios reales.

El esquema de los datos de resultados con los que evaluaremos la validez del modelo y a partir de los cuales generaremos estrategias se muestra en la Tabla 2. Cada fichero asociado a un umbral porcentual ($\pm 1\% \dots \pm 10\%$) tendrá la misma estructura, lo que facilita la comparación sistemática entre umbrales y criterios:

stock	predicted_feature	dirección_entrada	resultado	fecha_entrada	precio_entrada
AAPL	Close	ALCISTA	EXITOSA	2010-01-04	7.64
AAPL	porcentaje_cambio	BAJISTA	PERDIDA	2020-10-28	111.2
AAPL	porcentaje_cambio	ALCISTA	EXITOSA	2023-04-12	160.1
AAPL	Close	ALCISTA	PERDIDA	2024-03-08	170.73

fecha_salida	precio_salida	arquitectura	layer_size	n_lookback	n_forecast	epochs	batch_size	porcentaje_operacion
2010-01-06	7.53	LSTM	256	60	10	5	1	1.46
2020-10-29	115.32	GRU	128	150	1	5	1	-3.57
2023-04-13	165.56	LSTM	256	60	1	5	1	3.41
2024-03-11	172.75	LSTM	256	60	10	5	1	-1.19

Tabla 2 Tabla de estructura de resultados para la evaluación de modelos

En la Tabla 2 se muestran algunas filas seleccionadas al azar a modo de ejemplo, pero conviene aclarar que los ficheros de resultados no se generan de forma aleatoria, sino a partir de una configuración concreta de variables del modelo. Entre los parámetros utilizados se encuentran, por ejemplo, un valor de $n_lookback = 60$ y un $n_forecast = 10$, junto con la definición de la variable objetivo a predecir, que puede ser el precio de cierre (*Close*) o el porcentaje de cambio. Con esta configuración se realizan las predicciones sobre la serie temporal completa, obteniéndose un fichero que recoge los resultados para prácticamente todas las fechas. No obstante, aquellas predicciones cuya magnitud no supera el umbral porcentual establecido se descartan, de modo que solo se incorporan al fichero de porcentaje correspondiente los registros que cumplen el nivel mínimo de variación definido.

5.3 Métricas de Evaluación y Análisis de Resultados

La evaluación de cada modelo no solo se limitará al cálculo de la rentabilidad de las operaciones simuladas, sino que se apoyará en métricas de desempeño ampliamente utilizadas en el campo del aprendizaje automático. En particular, se medirá el porcentaje de éxito en las operaciones alcistas y bajistas para cada rango porcentual, así como métricas estadísticas como la **precisión (precision)**, la **exhaustividad (recall)** y el **F1-score** para las operaciones alcistas. De manera complementaria,

también se calcularán métricas equivalentes para las operaciones bajistas: el **valor predictivo negativo** (*Negative Predictive Value*, NPV) y la **especificidad** (*Specificity o True Negative Rate*, TNR), junto con su correspondiente **F1-neg**.

Para ello, se empleará la matriz de confusión como herramienta central de análisis. En este contexto, las operaciones alcistas se considerarán como predicciones positivas, mientras que las bajistas serán catalogadas como negativas. De esta manera, los verdaderos positivos (TP) corresponderán a las operaciones que el modelo predijo como alcistas y que efectivamente lo fueron; los verdaderos negativos (TN) serán aquellas que predijo como bajistas y realmente terminaron a la baja; los falsos positivos (FP) reflejarán los casos en los que el modelo predijo una subida pero en realidad se produjo una caída; y finalmente, los falsos negativos (FN) representarán los casos en los que el modelo predijo una bajada pero en realidad se produjo una subida.

A partir de esta matriz, se podrán calcular distintas métricas de desempeño:

- **Precisión (Precision):** Mide qué proporción de las operaciones predichas como alcistas realmente lo fueron, es decir, la exactitud de las predicciones positivas. Su fórmula se representa en la Ecuación 28:

$$precision = \frac{TP}{TP+FP} \quad (28)$$

- **Exhaustividad (Recall):** Indica qué proporción de todas las operaciones alcistas reales fueron correctamente identificadas por el modelo, midiendo su capacidad para no dejar escapar oportunidades relevantes. Su fórmula se representa en la Ecuación 29:

$$recall = \frac{TP}{TP+FN} \quad (29)$$

- **Medida F1 (F1-score):** Representa el promedio armónico entre la precisión y el recall, proporcionando un equilibrio entre ambas métricas, especialmente útil cuando las clases no están balanceadas. Su fórmula se representa en la Ecuación 30:

$$F1\text{-score} = 2 * \frac{precision*recall}{precision+recall} \quad (30)$$

- **Valor predictivo negativo (NPV):** mide la proporción de operaciones que el modelo predijo como bajistas y que efectivamente terminaron a la baja. Su fórmula se representa en la Ecuación 31:

$$NPV = \frac{TN}{TN+FN} \quad (31)$$

- **Especificidad (Specificity o True Negative Rate, TNR):** refleja la capacidad del modelo de identificar correctamente las bajistas, evitando confundirlas con alcistas. Su fórmula se representa en la Ecuación 32:

$$TNR = \frac{TN}{TN+FP} \quad (32)$$

- **F1-score para bajista (F1-neg):** Al igual que en el caso positivo, combina NPV y TNR como media armónica. Su fórmula se representa en la Ecuación 33:

$$F1\text{-neg} = 2 * \frac{NPV*TNR}{NPV+TNR} \quad (33)$$

Además, al repetir este análisis en cada fichero de porcentaje, se obtendrá una visión comparativa del rendimiento de los modelos según distintos niveles de exigencia. Esto es especialmente importante, dado que no resulta equivalente predecir correctamente un cambio del 1% que anticipar un movimiento del 10%. El esquema de resultados será documentado de manera detallada para cada rango, constituyendo la base sobre la cual se validará la utilidad de los modelos en la construcción de estrategias de inversión sólidas y comparables.

En la Figura 21 se muestra un ejemplo de cómo se van a representar los resultados, donde se presentan las estadísticas correspondientes a cada fichero de resultados, siendo **Forecast 1** las estadísticas de las predicciones de 1%... y así sucesivamente:



Figura 21 Ejemplo de estructura de resultados y métricas de evaluación por fichero porcentual

Como ya se ha mencionado, el presente trabajo realizará predicciones tanto sobre **precios de cierre** como sobre **porcentajes de cambio**, lo que permitirá evaluar la eficacia de los modelos bajo dos enfoques distintos de series temporales. Para ello, se emplearán diferentes acciones y parámetros de entrenamiento con el fin de comparar los resultados y determinar cuáles ofrecen un mejor desempeño en la construcción de estrategias de inversión.

Los parámetros que se van a utilizar son los siguientes:

- **Núm lookback:** Representa la cantidad de pasos temporales previos que el modelo utilizará como entrada para generar la predicción, determinando así la memoria del sistema.
- **Num forecast:** Corresponde al número de pasos futuros que se desea predecir; en el caso del porcentaje de cambio, este valor será siempre 1, ya que la predicción se centra en el movimiento inmediato del mercado.
- **Batch size:** Especifica la cantidad de muestras procesadas antes de que el modelo actualice sus pesos, lo cual impacta en la velocidad de entrenamiento y la estabilidad de la convergencia.
- **Epochs:** Cantidad de veces que el modelo recorrerá el conjunto de datos durante el entrenamiento, lo que influye directamente en su capacidad de aprendizaje y ajuste.
- **Layer size:** En el caso de modelos basados en redes neuronales, que determina el número de neuronas en cada capa oculta y, por tanto, la complejidad de la red y su capacidad de capturar patrones no lineales en los datos.

Este conjunto de parámetros será objeto de un proceso sistemático de ajuste y comparación en distintos escenarios experimentales, de modo que se puedan analizar con detalle sus efectos sobre el comportamiento de los modelos. No se trata únicamente de observar la precisión de las predicciones en términos absolutos, sino de identificar cómo cada configuración concreta influye en la estabilidad, la capacidad de generalización y la sensibilidad del sistema frente a diferentes condiciones de mercado. Para ello, se plantearán múltiples combinaciones de valores para los parámetros de *lookback*, *forecast*, *batch size*, *epochs* y *layer size*, evaluando su impacto tanto en la variable de cierre como en el

porcentaje de cambio. Los resultados obtenidos serán registrados y comparados, con el propósito de identificar aquellas configuraciones que permitan desarrollar estrategias de inversión más consistentes y aplicables en distintos contextos de mercado.

En cuanto a las **predicciones con precios de cierre**, el procedimiento se asemeja a lo ilustrado en la Figura 22, donde el modelo genera una proyección de los valores futuros de cierre a partir de los datos históricos. Este tipo de predicciones busca capturar las tendencias generales del mercado y es especialmente útil para evaluar el comportamiento global de un activo en horizontes temporales de corto o mediano plazo. En este escenario, se analiza cómo el modelo es capaz de ajustarse a los movimientos del precio y hasta qué punto anticipa cambios en la dirección de la tendencia. Al comparar la predicción con los valores reales, se pueden medir tanto la precisión de las proyecciones como la utilidad de estas para la toma de decisiones de inversión:



Figura 22 Ejemplo de predicción de precios predichos vs precios reales

Por otro lado, cuando se trabaja con **porcentajes de cambio**, el enfoque difiere y se orienta a medir las variaciones relativas entre un cierre y el siguiente. Tal como se representa en la Figura 23, las predicciones ya no se centran en valores absolutos del precio, sino en estimar si el mercado experimentará una subida o bajada en términos porcentuales. Este método resulta particularmente valioso para estrategias de trading más dinámicas, donde interesa identificar movimientos puntuales que generen oportunidades de entrada y salida rápidas. Al considerar únicamente un horizonte de predicción de un paso (*num forecast = 1*), el análisis se concentra en la inmediatez del cambio, lo que permite evaluar con mayor claridad la capacidad del modelo para anticipar fluctuaciones a corto plazo:

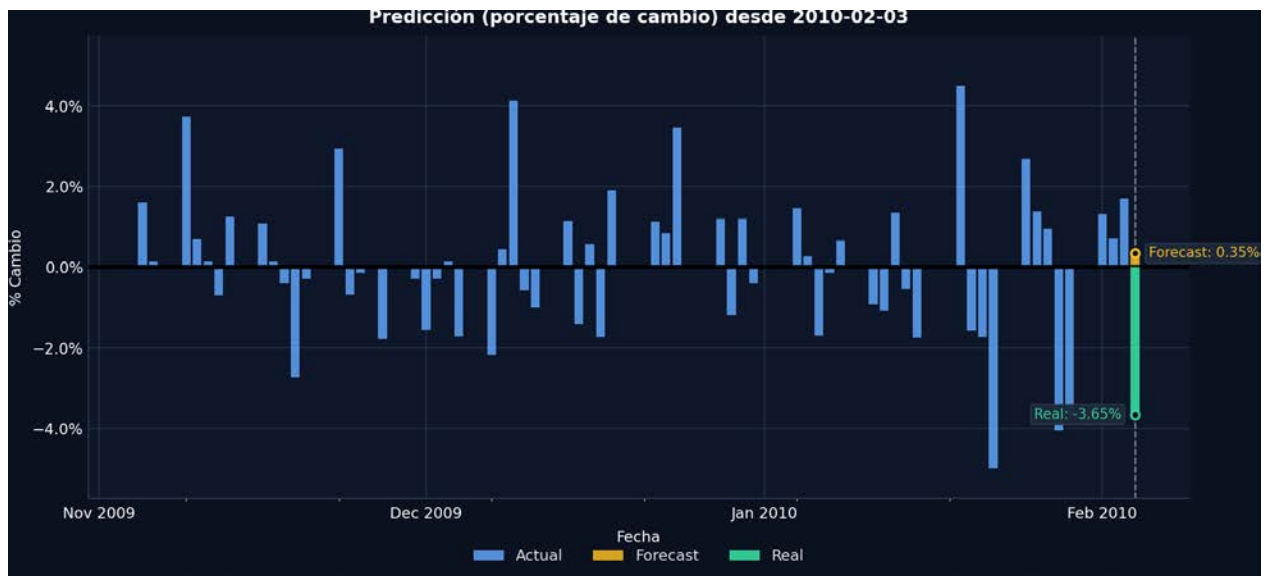


Figura 23 Ejemplo de predicción de porcentaje de cambio vs porcentaje de cambio real

De esta manera, la combinación de ambos enfoques (precios de cierre y porcentajes de cambio) ofrece una visión más completa y complementaria del comportamiento del mercado. Mientras el primero aporta información sobre tendencias y trayectorias, el segundo brinda señales tácticas sobre movimientos inmediatos, lo que permite fortalecer el diseño de estrategias de inversión tanto de carácter estratégico como operativo.

Se evaluarán diferentes **modelos y arquitecturas** con el fin de determinar cuáles presentan un mejor desempeño en la predicción de los movimientos del mercado. Para ello, se presentarán las estadísticas de cada modelo en términos de porcentaje de acierto, acompañadas de métricas adicionales como precisión, *recall* y *F1-score* para las alcistas y NPV, TNR y *F1-neg* para las bajistas, que permitirán una evaluación más rigurosa de su capacidad predictiva. Estos resultados se documentarán de manera sistemática, lo que facilitará la comparación entre modelos bajo diferentes configuraciones y parámetros.

Posteriormente, se llevará a cabo un proceso de **análisis comparativo de modelos (data science)**, en el que se contrastarán no solo los niveles de acierto, sino también la robustez y estabilidad de cada aproximación frente a distintos escenarios de mercado. Este análisis permitirá identificar patrones en el comportamiento de los modelos según las variables utilizadas (por ejemplo, precios de cierre o porcentajes de cambio) y los parámetros definidos (como *num lookback*, *num forecast*, *batch size*, *epochs* o *layer size*).

Finalmente, se aplicará un **backtesting** al modelo que haya demostrado los mejores resultados. Este procedimiento consiste en simular la aplicación de la estrategia generada por dicho modelo sobre datos históricos, evaluando así su desempeño en condiciones reales de mercado. Con ello, se podrá medir la rentabilidad, consistencia y riesgo asociado a la estrategia propuesta. Los resultados obtenidos se compararán con el índice de referencia **S&P 500**, lo que permitirá evaluar de manera objetiva si la estrategia generada aporta un valor añadido respecto a seguir un índice de mercado ampliamente reconocido.

6 MODELADOS PREDICTIVOS: IMPLEMENTACIÓN Y RESULTADOS

6.1 Implementación de modelos tradicionales de series temporales

6.1.1 Proceso Auto-ARIMA

Para la implementación de los modelos ARIMA y SARIMAX no se realizará una búsqueda manual de parámetros, sino que se empleará un enfoque automatizado.

Como ya se ha mencionado en este trabajo, la construcción de un modelo ARIMA requiere elegir adecuadamente los parámetros (p,d,q) , donde ‘p’ corresponde a los términos autorregresivos (AR), ‘d’ al grado de diferenciación necesario para transformar la serie en estacionaria, y ‘q’ a los términos de medias móviles (MA). Para determinar estos valores, se suele recurrir al análisis de la autocorrelación (ACF) y la autocorrelación parcial (PACF), así como a la inspección de la serie en busca de estacionariedad. No obstante, este procedimiento puede resultar laborioso y en cierta medida subjetivo, ya que requiere interpretar gráficos y probar distintas combinaciones de parámetros. A partir de esta necesidad surge el modelo **Auto-ARIMA**, que automatiza la selección de parámetros óptimos mediante criterios estadísticos, facilitando la construcción de modelos más robustos y eficientes.

La función Auto-ARIMA no solo selecciona automáticamente los valores más adecuados para ‘p’ y ‘q’, sino que también es capaz de estimar valores faltantes, aplicar diferenciación estacional, detectar valores atípicos y generar previsiones. Dado que un usuario riguroso puede estar interesado tanto en la serie original libre de *outliers* como en las previsiones correspondientes, el algoritmo proporciona como salida tanto la serie depurada de valores atípicos como las predicciones asociadas a ella [53].

El algoritmo **auto-ARIMA** sigue una serie de pasos para construir el modelo:

- En primer lugar, verifica si es necesario aplicar diferenciación e identifica el valor óptimo del parámetro ‘d’.
- Transforma la serie en estacionaria calculando las diferencias entre observaciones consecutivas.
- Con los criterios de información seleccionados y el valor de ‘d’, explora diferentes combinaciones de (p,d,q) hasta encontrar los valores más adecuados de ‘p’ y ‘q’.
- Finalmente, con los parámetros óptimos obtenidos, genera la previsión aplicando los términos autorregresivos y de medias móviles sobre la serie ya diferenciada.

En el caso de auto-ARIMA estacional (auto-SARIMA), el proceso se ejecuta considerando ciclos estacionales en lugar de periodos independientes. En el caso de auto-SARIMAX, los parámetros correspondientes a las componentes estacionales se representan como (P,D,Q), tal y como se vio previamente en este trabajo. El algoritmo auto-SARIMA desarrolla los siguientes pasos:

- Determina la duración del ciclo estacional si esta no fue proporcionada.
- Verifica si es necesario aplicar diferenciación e identifica los órdenes óptimos para ‘d’ y ‘D’.
- Con dichos valores, transforma la serie en estacionaria calculando las diferencias correspondientes.
- A partir de los criterios de información seleccionados, explora múltiples combinaciones de (p,d,q,P,D,Q) hasta encontrar los valores más adecuados para cada parámetro.
- Finalmente, con los parámetros óptimos determinados, el algoritmo genera la previsión aplicando tanto los componentes autorregresivos como los de medias móviles, en sus partes estacional y no estacional [54].

6.1.2 ARIMA: implementación y resultados

Se van a mostrar los resultados estadísticos obtenidos con el modelo **ARIMA**, aplicados tanto a la predicción de los precios de cierre como a la predicción de los porcentajes de cambio.

En la Figura 24 se presentan las estadísticas de las predicciones realizadas sobre los precios de cierre, desglosadas por cada rango porcentual considerado:

Forecast 1					Forecast 2					Forecast 3				
Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación	Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación	Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación
Forecast 1	ALCISTA	172	54.07%	63.50	Forecast 2	ALCISTA	27	70.37%	39.02	Forecast 3	ALCISTA	9	88.89%	40.43
Forecast 1	BAJISTA	122	52.46%	31.67	Forecast 2	BAJISTA	19	57.89%	2.02	Forecast 3	BAJISTA	10	70.00%	11.31
Forecast 1	TOTAL	294	53.40%	95.17	Forecast 2	TOTAL	46	65.22%	61.04	Forecast 3	TOTAL	19	79.95%	51.74

Matriz de Confusión					Matriz de Confusión					Matriz de Confusión				
Real \ Pred →		ALCISTA	BAJISTA		Real \ Pred →		ALCISTA	BAJISTA		Real \ Pred →		ALCISTA	BAJISTA	
ALCISTA		93	58		ALCISTA		19	8		ALCISTA		8	3	
BAJISTA		79	64		BAJISTA		8	11		BAJISTA		1	7	
Precision: 0.54		Recall: 0.52		F1-score: 0.58	Precision: 0.70		Recall: 0.70		F1-score: 0.70	Precision: 0.89		Recall: 0.73		F1-score: 0.80
NPV: 0.52		TNR: 0.45		F1-Neg: 0.48	NPV: 0.58		TNR: 0.58		F1-Neg: 0.58	NPV: 0.70		TNR: 0.88		F1-Neg: 0.78

Forecast 4					Forecast 5				
Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación	Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación
Forecast 4	ALCISTA	4	100.00%	22.22	Forecast 5	ALCISTA	2	100.00%	12.35
Forecast 4	BAJISTA	5	80.00%	12.60	Forecast 5	BAJISTA	2	50.00%	-6.14
Forecast 4	TOTAL	9	88.89%	34.82	Forecast 5	TOTAL	4	75.00%	6.41

Matriz de Confusión					Matriz de Confusión				
Real \ Pred →		ALCISTA	BAJISTA		Real \ Pred →		ALCISTA	BAJISTA	
ALCISTA		4	1		ALCISTA		2	1	
BAJISTA		0	4		BAJISTA		0	1	
Precision: 1.00		Recall: 0.80		F1-score: 0.89	Precision: 1.00		Recall: 0.67		F1-score: 0.80
NPV: 0.80		TNR: 1.00		F1-Neg: 0.89	NPV: 0.50		TNR: 1.00		F1-Neg: 0.67

Figura 24 Estadísticas del modelo ARIMA aplicadas a precios de cierre en cada rango porcentual.

Los resultados reflejan se obtiene un mayor porcentaje de acierto en las señales alcistas, aunque no genera muchas operaciones. En el fichero de resultados se aprecia que al incrementar el umbral porcentual exigido aumenta la probabilidad de éxito, pero a costa de una reducción progresiva del número de operaciones disponibles.

En la Figura 25 se muestran las estadísticas de las predicciones de los porcentajes de cambio, igualmente distribuidas por cada rango porcentual:

Forecast 1				
Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación
Forecast 1	ALCISTA	3	66.67%	8.95
Forecast 1	BAJISTA	4	75.00%	2.35
Forecast 1	TOTAL	7	71.43%	11.30

Matriz de Confusión				
Real \ Pred →		ALCISTA	BAJISTA	
ALCISTA		2	1	
BAJISTA		1	3	
Precision: 0.67		Recall: 0.67		F1-score: 0.67
NPV: 0.75		TNR: 0.75		F1-Neg: 0.75

Figura 25 Estadísticas del modelo ARIMA aplicadas a porcentajes de cambio en cada rango porcentual.

En el caso de las predicciones basadas en porcentajes de cambio, el modelo apenas genera operaciones y no supera variaciones superiores al 1%, lo que limita su capacidad para capturar movimientos de mayor magnitud.

6.1.3 SARIMAX: implementación y resultados

Se van a mostrar los resultados estadísticos obtenidos con el modelo **SARIMAX**, aplicados tanto a la predicción de los precios de cierre como a la predicción de los porcentajes de cambio.

En la Figura 26 se presentan las estadísticas de las predicciones realizadas sobre los precios de cierre, desglosadas por cada rango porcentual considerado:

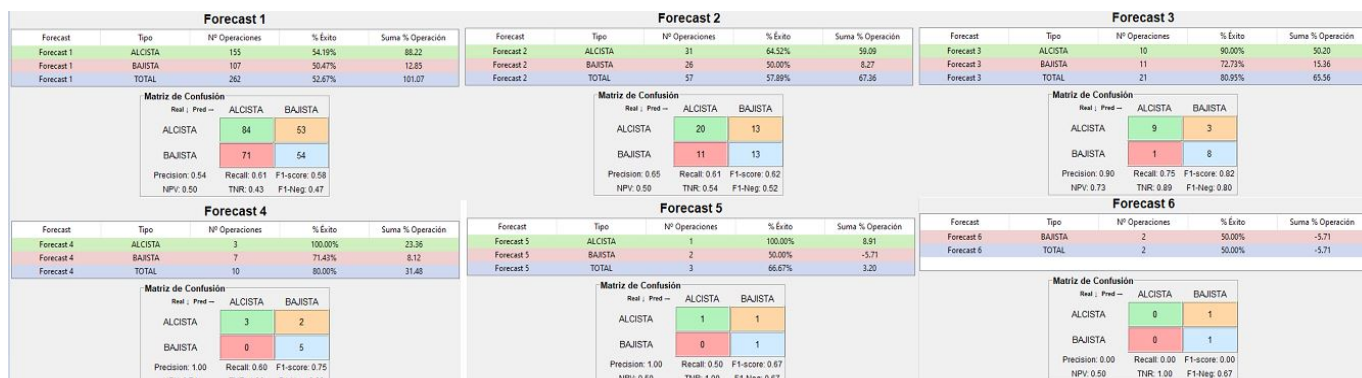


Figura 26 Estadísticas del modelo SARIMAX aplicadas a precios de cierre en cada rango porcentual.

Los resultados reflejan se obtiene un mayor porcentaje de acierto en las señales alcistas, aunque no genera muchas operaciones. En el fichero de resultados se aprecia que al incrementar el umbral porcentual exigido aumenta la probabilidad de éxito, pero a costa de una reducción progresiva del número de operaciones disponibles.

En la Figura 27 se muestran las estadísticas de las predicciones de los porcentajes de cambio, igualmente distribuidas por cada rango porcentual:



Figura 27 Estadísticas del modelo SARIMAX aplicadas a porcentajes de cambio en cada rango porcentual.

En el caso de las predicciones basadas en porcentajes de cambio, el modelo apenas genera operaciones y no supera variaciones superiores al 2%, lo que limita su capacidad para capturar movimientos de mayor magnitud.

6.2 Implementación de modelos de aprendizaje automático

6.2.1 Random Forest: implementación y resultados

Se van a mostrar los resultados estadísticos obtenidos con el modelo **Random Forest**, aplicados tanto a la predicción de los precios de cierre como a la predicción de los porcentajes de cambio.

En la Figura 28 se presentan las estadísticas de las predicciones realizadas sobre los precios de cierre, desglosadas por cada rango porcentual considerado:



Figura 28 Estadísticas del modelo Random Forest aplicadas a precios de cierre en cada rango porcentual.

Los resultados reflejan se obtiene un mayor porcentaje de acierto en las señales alcistas. En el fichero de resultados se aprecia que al incrementar el umbral porcentual exigido aumenta la probabilidad de éxito, pero a costa de una reducción progresiva del número de operaciones disponibles.

En la Figura 29 se muestran las estadísticas de las predicciones de los porcentajes de cambio, igualmente distribuidas por cada rango porcentual:



Figura 29 Estadísticas del modelo Random Forest aplicadas a porcentajes de cambio en cada rango porcentual.

Los resultados reflejan se obtiene un mayor porcentaje de acierto en las señales alcistas. En el fichero de resultados no se aprecia que al incrementando el umbral porcentual exigido aumente la probabilidad de éxito, pero sí una reducción progresiva del número de operaciones disponibles.

6.2.2 XGBoost: implementación y resultados

Se van a mostrar los resultados estadísticos obtenidos con el modelo **XGBoost**, aplicados tanto a la predicción de los precios de cierre como a la predicción de los porcentajes de cambio.

En la Figura 30 se presentan las estadísticas de las predicciones realizadas sobre los precios de cierre, desglosadas por cada rango porcentual considerado:

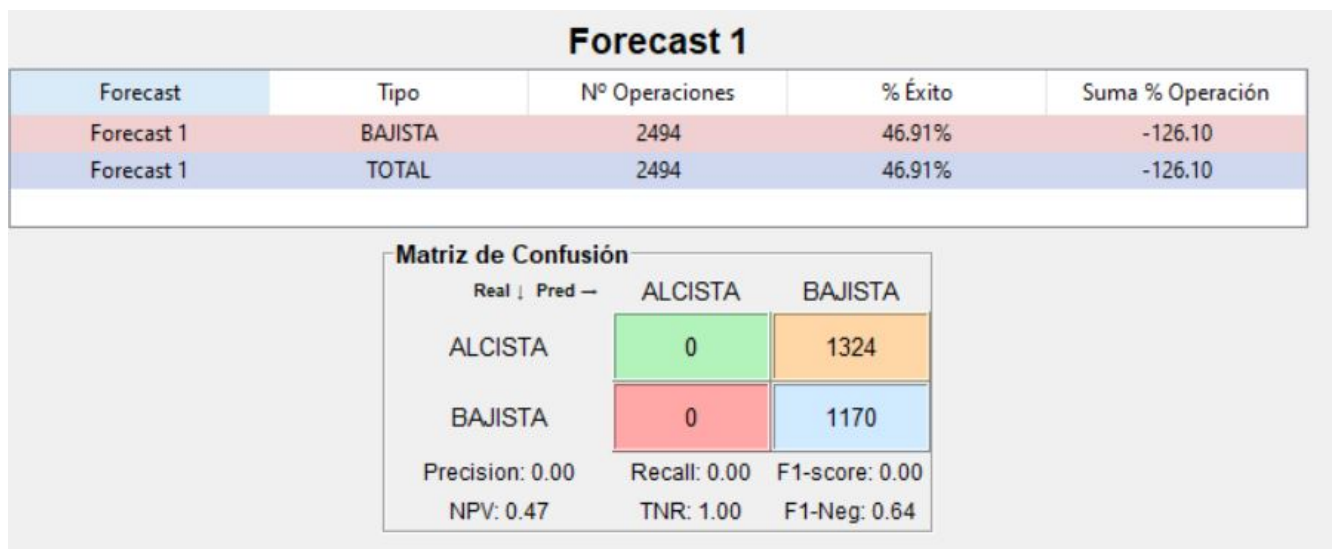


Figura 30 Estadísticas del modelo XGBoost aplicadas a precios de cierre en cada rango porcentual.

Los resultados muestran que no es capaz de realizar predicciones para precios de cierre.

En la Figura 31 se muestran las estadísticas de las predicciones de los porcentajes de cambio, igualmente distribuidas por cada rango porcentual:

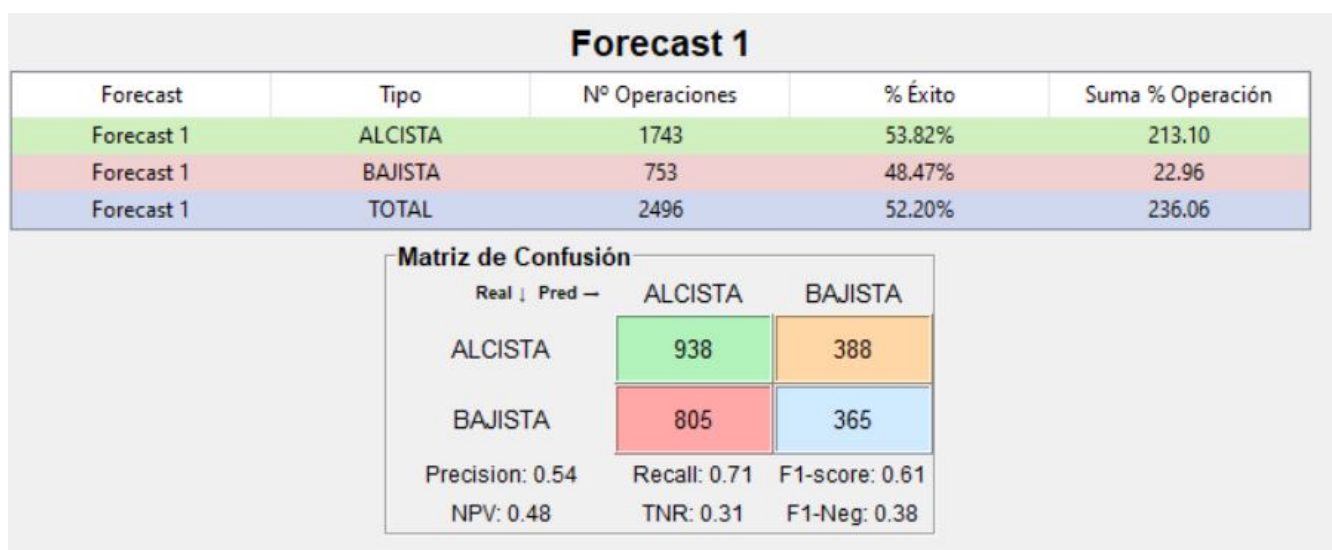


Figura 31 Estadísticas del modelo XGBoost aplicadas a porcentajes de cambio en cada rango porcentual.

En el caso de las predicciones basadas en porcentajes de cambio, el modelo apenas genera operaciones y no supera variaciones superiores al 1%, lo que limita su capacidad para capturar movimientos de mayor magnitud.

6.2.3 LightGBM: implementación y resultados

Se van a mostrar los resultados estadísticos obtenidos con el modelo **LightGBM**, aplicados tanto a la predicción de los precios de cierre como a la predicción de los porcentajes de cambio.

En la Figura 32 se presentan las estadísticas de las predicciones realizadas sobre los precios de cierre, desglosadas por cada rango porcentual considerado:

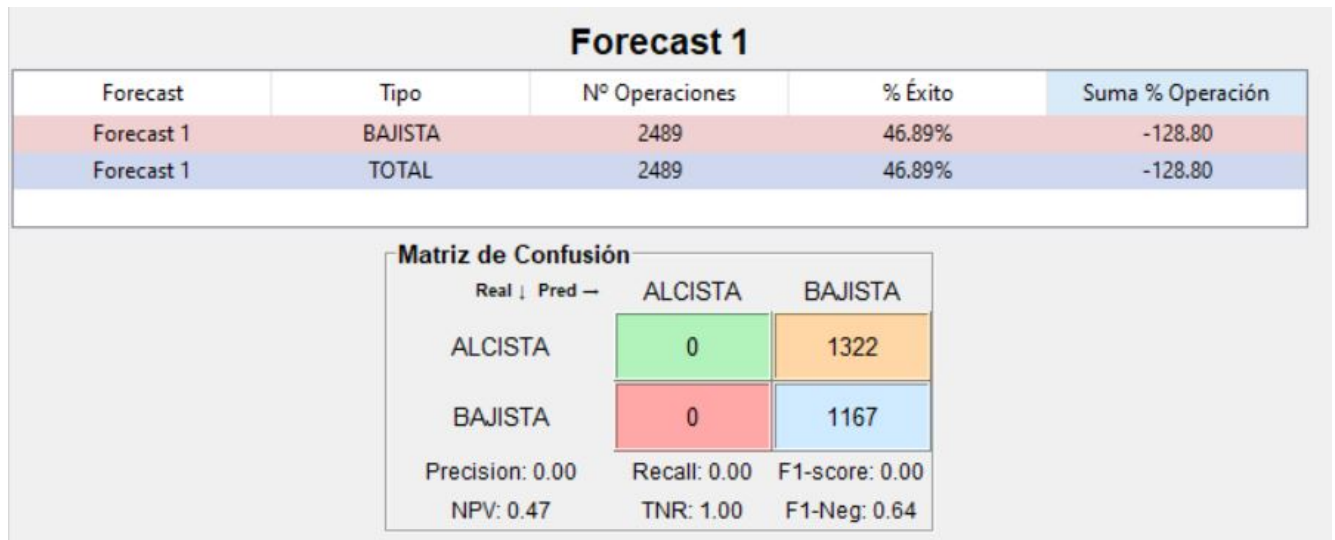


Figura 32 Estadísticas del modelo LightGBM aplicadas a precios de cierre en cada rango porcentual.

Los resultados muestran que no es capaz de realizar predicciones para precios de cierre.

En la Figura 33 se muestran las estadísticas de las predicciones de los porcentajes de cambio, igualmente distribuidas por cada rango porcentual:

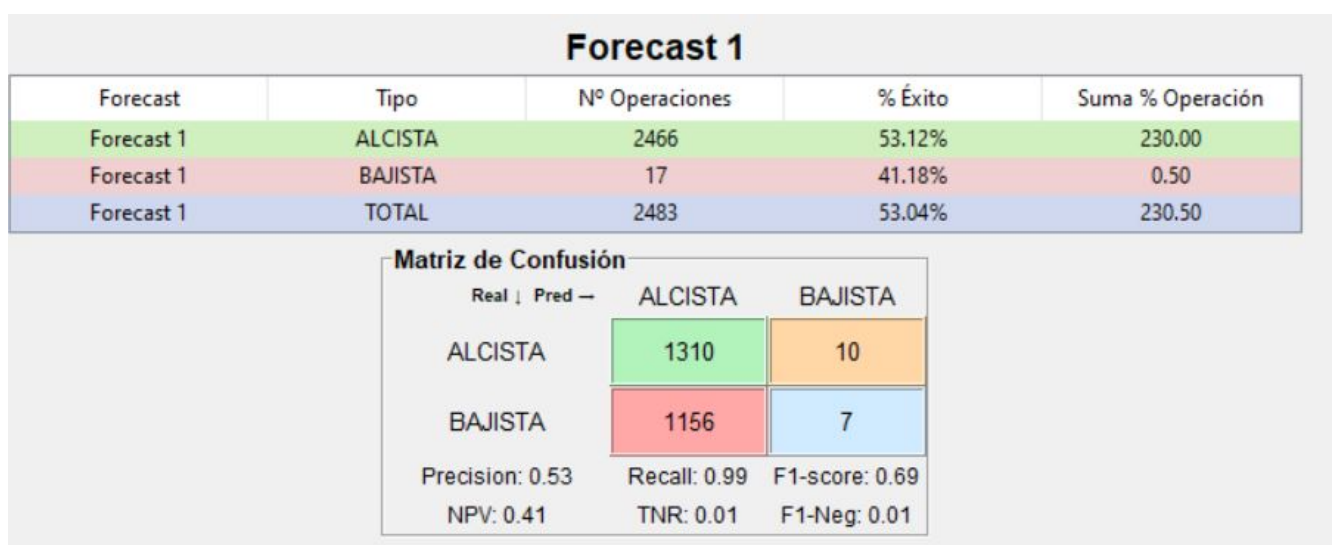


Figura 33 Estadísticas del modelo LightGBM aplicadas a porcentajes de cambio en cada rango porcentual.

En el caso de las predicciones basadas en porcentajes de cambio, el modelo apenas genera operaciones y no supera variaciones superiores al 1%, lo que limita su capacidad para capturar movimientos de mayor magnitud.

6.3 Implementación de modelos de aprendizaje profundo

6.3.1 GRU: implementación y resultados

Se van a mostrar los resultados estadísticos obtenidos con el modelo **GRU**, aplicados tanto a la predicción de los precios de cierre como a la predicción de los porcentajes de cambio.

En la Figura 34 se presentan las estadísticas de las predicciones realizadas sobre los precios de cierre, desglosadas por cada rango porcentual considerado:



Figura 34 Estadísticas del modelo GRU aplicadas a precios de cierre en cada rango porcentual.

Los resultados reflejan se obtiene un mayor porcentaje de acierto en las señales alcistas. En el fichero de resultados se aprecia que al incrementar el umbral porcentual exigido aumenta la probabilidad de éxito, pero a costa de una reducción progresiva del número de operaciones disponibles. Se observan muchas operaciones, lo que nos muestra que la mayoría de las veces la predicción va a generar un resultado.

En la Figura 35 se muestran las estadísticas de las predicciones de los porcentajes de cambio, igualmente distribuidas por cada rango porcentual:

Forecast 1					Forecast 2					Forecast 3				
Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación	Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación	Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación
Forecast 1	ALCISTA	130	52.31%	13.28	Forecast 2	ALCISTA	12	75.00%	11.72	Forecast 3	ALCISTA	6	50.00%	3.62
Forecast 1	BAJISTA	49	53.06%	16.87	Forecast 2	BAJISTA	8	50.00%	-4.66	Forecast 3	BAJISTA	1	0.00%	-0.23
Forecast 1	TOTAL	179	52.51%	30.15	Forecast 2	TOTAL	20	65.00%	7.06	Forecast 3	TOTAL	7	42.86%	3.59
Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 68 23 BAJISTA 62 26 Precision: 0.52 Recall: 0.75 F1-score: 0.62 NPV: 0.53 TNR: 0.30 F1-Neg: 0.38					Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 9 4 BAJISTA 3 4 Precision: 0.75 Recall: 0.69 F1-score: 0.72 NPV: 0.50 TNR: 0.57 F1-Neg: 0.53					Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 3 1 BAJISTA 3 0 Precision: 0.50 Recall: 0.75 F1-score: 0.60 NPV: 0.00 TNR: 0.00 F1-Neg: 0.00				
Forecast 4					Forecast 5					Forecast 6				
Forecast 4	ALCISTA	4	50.00%	4.87	Forecast 5	ALCISTA	1	100.00%	8.91	Forecast 6	ALCISTA	1	100.00%	8.91
Forecast 4	BAJISTA	0	0.00%	0.00	Forecast 5	BAJISTA	0	0.00%	0.00	Forecast 6	BAJISTA	0	0.00%	0.00
Forecast 4	TOTAL	4	50.00%	4.87	Forecast 5	TOTAL	1	100.00%	8.91	Forecast 6	TOTAL	1	100.00%	8.91
Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 2 0 BAJISTA 2 0 Precision: 0.50 Recall: 1.00 F1-score: 0.67 NPV: 0.00 TNR: 0.00 F1-Neg: 0.00					Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 1 0 BAJISTA 0 0 Precision: 1.00 Recall: 1.00 F1-score: 1.00 NPV: 0.00 TNR: 0.00 F1-Neg: 0.00					Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 1 0 BAJISTA 0 0 Precision: 1.00 Recall: 1.00 F1-score: 1.00 NPV: 0.00 TNR: 0.00 F1-Neg: 0.00				

Figura 35 Estadísticas del modelo GRU aplicadas a porcentajes de cambio en cada rango porcentual.

Los resultados reflejan se obtiene un mayor porcentaje de acierto en las señales alcistas. En el fichero de resultados se aprecia que al incrementar el umbral porcentual exigido aumenta la probabilidad de éxito, pero a costa de una reducción progresiva del número de operaciones disponibles.

6.3.2 LSTM: implementación y resultados

Se van a mostrar los resultados estadísticos obtenidos con el modelo **LSTM**, aplicados tanto a la predicción de los precios de cierre como a la predicción de los porcentajes de cambio.

En la Figura 36 se presentan las estadísticas de las predicciones realizadas sobre los precios de cierre, desglosadas por cada rango porcentual considerado:

Forecast 1					Forecast 2					Forecast 3				
Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación	Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación	Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación
Forecast 1	ALCISTA	1059	54.49%	264.04	Forecast 2	ALCISTA	907	55.35%	285.01	Forecast 3	ALCISTA	785	56.18%	429.09
Forecast 1	BAJISTA	3653	48.02%	-187.72	Forecast 2	BAJISTA	3310	46.44%	-99.85	Forecast 3	BAJISTA	3383	47.15%	-31.36
Forecast 1	TOTAL	4712	47.82%	76.32	Forecast 2	TOTAL	4417	48.27%	185.16	Forecast 3	TOTAL	4168	48.92%	397.73
Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 577 1972 BAJISTA 492 1681 Precision: 0.54 Recall: 0.23 F1-score: 0.32 NPV: 0.46 TNR: 0.78 F1-Neg: 0.58					Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 582 1890 BAJISTA 405 1630 Precision: 0.55 Recall: 0.21 F1-score: 0.31 NPV: 0.46 TNR: 0.80 F1-Neg: 0.59					Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 441 1788 BAJISTA 344 1595 Precision: 0.56 Recall: 0.20 F1-score: 0.29 NPV: 0.47 TNR: 0.82 F1-Neg: 0.60				
Forecast 4					Forecast 5					Forecast 6				
Forecast 4	ALCISTA	699	59.19%	544.69	Forecast 5	ALCISTA	358	60.54%	318.50	Forecast 6	ALCISTA	466	59.87%	502.86
Forecast 4	BAJISTA	3270	46.64%	-150.33	Forecast 5	BAJISTA	3175	45.92%	-268.61	Forecast 6	BAJISTA	3085	45.41%	-418.07
Forecast 4	TOTAL	3969	48.77%	394.36	Forecast 5	TOTAL	3733	48.03%	270.89	Forecast 6	TOTAL	3551	47.31%	84.79
Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 396 1745 BAJISTA 273 1625 Precision: 0.59 Recall: 0.18 F1-score: 0.28 NPV: 0.47 TNR: 0.85 F1-Neg: 0.60					Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 335 1717 BAJISTA 223 1458 Precision: 0.60 Recall: 0.16 F1-score: 0.26 NPV: 0.46 TNR: 0.87 F1-Neg: 0.60					Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 279 1684 BAJISTA 187 1401 Precision: 0.60 Recall: 0.14 F1-score: 0.23 NPV: 0.45 TNR: 0.88 F1-Neg: 0.60				
Forecast 7					Forecast 8					Forecast 9				
Forecast 7	ALCISTA	401	60.10%	478.30	Forecast 8	ALCISTA	333	60.06%	444.38	Forecast 9	ALCISTA	278	60.07%	414.10
Forecast 7	BAJISTA	2977	45.38%	-452.19	Forecast 8	BAJISTA	2884	45.74%	-367.85	Forecast 9	BAJISTA	2905	45.80%	-389.99
Forecast 7	TOTAL	3378	47.13%	26.11	Forecast 8	TOTAL	3217	47.32%	76.53	Forecast 9	TOTAL	3083	46.90%	24.11
Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 241 1626 BAJISTA 160 1351 Precision: 0.60 Recall: 0.13 F1-score: 0.21 NPV: 0.45 TNR: 0.88 F1-Neg: 0.60					Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 208 1555 BAJISTA 133 1319 Precision: 0.60 Recall: 0.11 F1-score: 0.19 NPV: 0.45 TNR: 0.91 F1-Neg: 0.61					Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 167 1526 BAJISTA 111 1279 Precision: 0.60 Recall: 0.10 F1-score: 0.17 NPV: 0.45 TNR: 0.92 F1-Neg: 0.61				
Forecast 10					Forecast 10					Forecast 10				
Forecast 10	ALCISTA	218	62.39%	364.99	Forecast 10	ALCISTA	218	62.39%	364.99	Forecast 10	ALCISTA	218	62.39%	364.99
Forecast 10	BAJISTA	2715	45.45%	-446.46	Forecast 10	BAJISTA	2715	45.45%	-446.46	Forecast 10	BAJISTA	2715	45.45%	-446.46
Forecast 10	TOTAL	2933	46.71%	-81.47	Forecast 10	TOTAL	2933	46.71%	-81.47	Forecast 10	TOTAL	2933	46.71%	-81.47
Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 136 1481 BAJISTA 82 1234 Precision: 0.62 Recall: 0.08 F1-score: 0.15 NPV: 0.45 TNR: 0.94 F1-Neg: 0.61					Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 136 1481 BAJISTA 82 1234 Precision: 0.62 Recall: 0.08 F1-score: 0.15 NPV: 0.45 TNR: 0.94 F1-Neg: 0.61					Matriz de Confusión Real \ Pred → ALCISTA BAJISTA ALCISTA 136 1481 BAJISTA 82 1234 Precision: 0.62 Recall: 0.08 F1-score: 0.15 NPV: 0.45 TNR: 0.94 F1-Neg: 0.61				

Figura 36 Estadísticas del modelo LSTM aplicadas a precios de cierre en cada rango porcentual.

Los resultados reflejan se obtiene un mayor porcentaje de acierto en las señales alcistas. En el fichero de resultados se aprecia que al incrementar el umbral porcentual exigido aumenta la probabilidad de éxito, pero a costa de una reducción progresiva del número de operaciones disponibles. Se observan muchas operaciones, lo que nos muestra que la mayoría de las veces la predicción va a generar un resultado.

En la Figura 37 se muestran las estadísticas de las predicciones de los porcentajes de cambio, igualmente distribuidas por cada rango porcentual:

PREDICCIÓN DE SERIES TEMPORALES UTILIZANDO INTELIGENCIA ARTIFICIAL: DESDE MODELOS CLÁSICOS HASTA REDES NEURONALES RECURRENTE

Forecast 1				
Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación
Forecast 1	ALCISTA	102	53.92%	28.47
Forecast 1	BAJISTA	28	46.43%	-5.38
Forecast 1	TOTAL	130	52.31%	23.09

Matriz de Confusión

Real \ Pred	ALCISTA	BAJISTA
ALCISTA	55	15
BAJISTA	47	13

Precision: 0.54

Recall: 0.79

F1-score: 0.64

NPV: 0.45

TNR: 0.22

F1-Neg: 0.30

Forecast 2				
Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación
Forecast 2	ALCISTA	10	40.00%	1.70
Forecast 2	BAJISTA	3	66.67%	-0.15
Forecast 2	TOTAL	13	46.15%	1.55

Matriz de Confusión

Real \ Pred	ALCISTA	BAJISTA
ALCISTA	4	1
BAJISTA	6	2

Precision: 0.40

Recall: 0.80

F1-score: 0.53

NPV: 0.67

TNR: 0.25

F1-Neg: 0.36

Forecast 3				
Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación
Forecast 3	ALCISTA	4	25.00%	-1.94
Forecast 3	BAJISTA	1	100.00%	0.05
Forecast 3	TOTAL	5	40.00%	-1.89

Matriz de Confusión

Real \ Pred	ALCISTA	BAJISTA
ALCISTA	1	0
BAJISTA	3	1

Precision: 0.25

Recall: 1.00

F1-score: 0.40

NPV: 1.00

TNR: 0.25

F1-Neg: 0.40

Forecast 4				
Forecast	Tipo	Nº Operaciones	% Éxito	Suma % Operación
Forecast 4	ALCISTA	2	0.00%	-1.28
Forecast 4	BAJISTA	1	100.00%	0.05
Forecast 4	TOTAL	3	33.33%	-1.23

Matriz de Confusión

Real \ Pred	ALCISTA	BAJISTA
ALCISTA	0	0
BAJISTA	2	1

Precision: 0.00

Recall: 0.00

F1-score: 0.00

NPV: 1.00

TNR: 0.33

F1-Neg: 0.50

Figura 37 Estadísticas del modelo LSTM aplicadas a porcentajes de cambio en cada rango porcentual.

Los resultados reflejan se obtiene un mayor porcentaje de acierto en las señales alcistas. En el fichero de resultados se aprecia que al incrementar el umbral porcentual exigido aumenta la probabilidad de éxito, pero a costa de una reducción progresiva del número de operaciones disponibles, observando que sólo tiene sentido la predicción del umbral del 1%.

7 DISCUSIÓN Y ANÁLISIS COMPARATIVO CON DATA SCIENCE

7.1 Comparación entre diferentes modelos de predicción

La comparación entre los modelos de predicción se realizará tomando como referencia dos enfoques distintos: los precios de cierre y los porcentajes de cambio. El criterio será el porcentaje de éxito obtenido en las operaciones, diferenciando entre predicciones alcistas y bajistas. Esto permitirá evaluar la capacidad de cada modelo para identificar correctamente la dirección de los movimientos del mercado en distintos escenarios.

En la Figura 38 se muestran los resultados comparativos de cada modelo tras realizar predicciones sobre los precios de cierre:

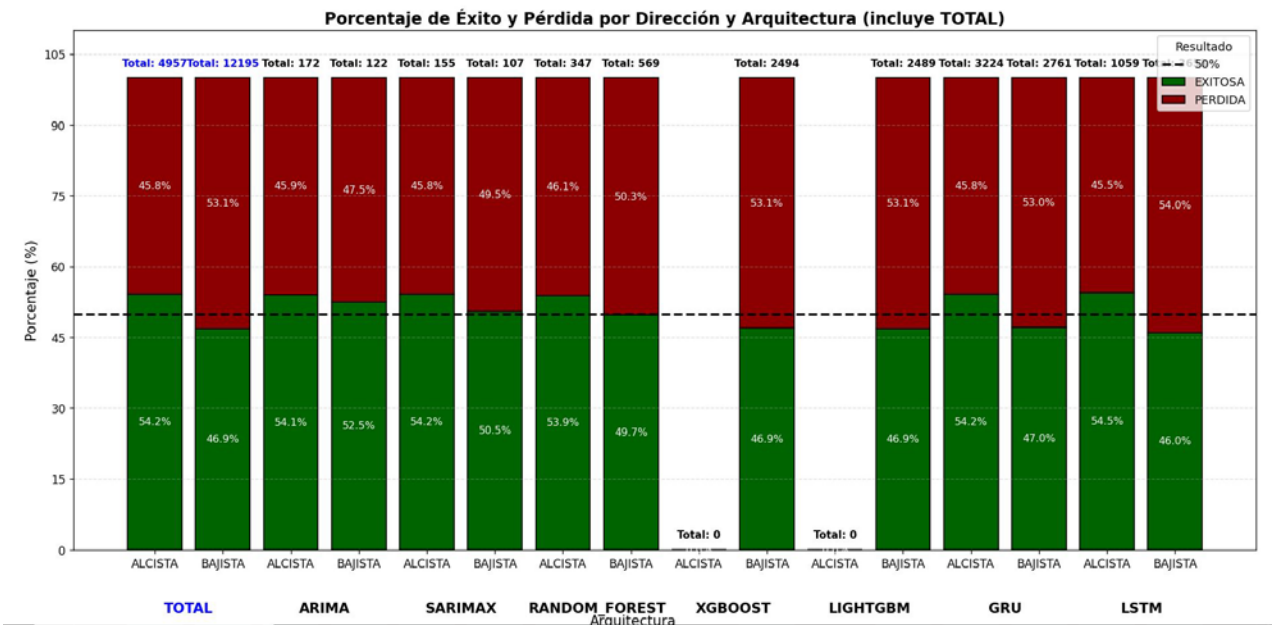


Figura 38 Resultados comparativos de los modelos en la predicción de precios de cierre

Se observa que el método con mayor porcentaje de acierto es el LSTM para operaciones alcistas.

En la Figura 39 se muestran los resultados comparativos de cada modelo tras realizar predicciones sobre los porcentajes de cambio:

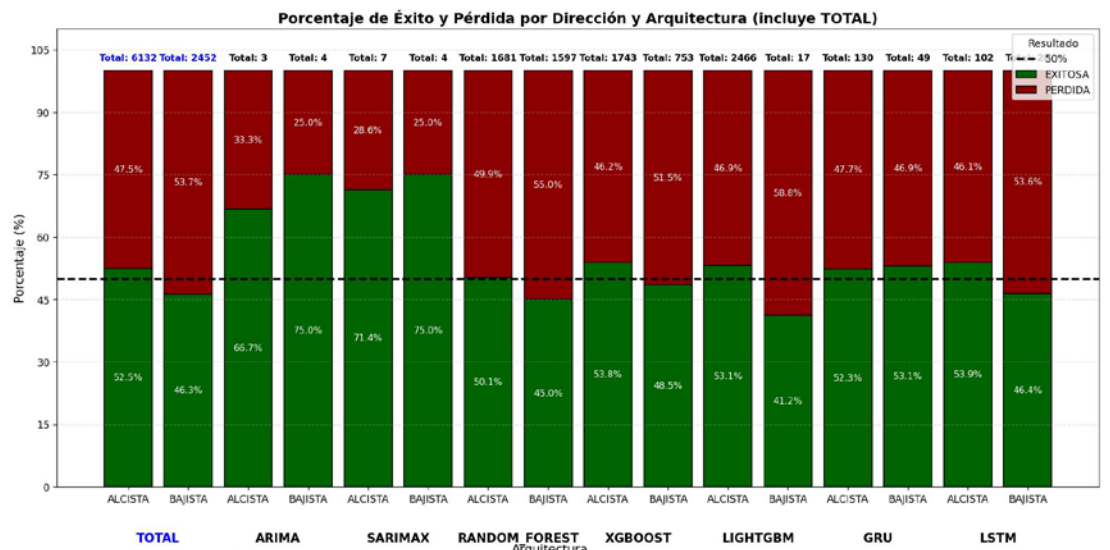


Figura 39 Resultados comparativos de los modelos en la predicción del porcentaje de cambio

Se observa que la arquitectura con mayor porcentaje de acierto es el LSTM para operaciones alcistas, ya que, aunque haya arquitecturas con mejor porcentaje de aciertos, no tienen suficiente número de muestras.

7.2 Comparación entre diferentes indicadores

Se procederá a la evaluación de diferentes indicadores aplicados con distintos valores de sus parámetros, tanto sobre precios de cierre como sobre porcentajes de cambio. El criterio será el porcentaje de éxito en las operaciones, distinguiendo entre predicciones alcistas y bajistas.

7.2.1 Stocks

En la Figura 40 se muestran los resultados comparativos de cada acción tras realizar predicciones sobre los precios de cierre, para comparar si se comporta de una forma parecida independientemente del activo:

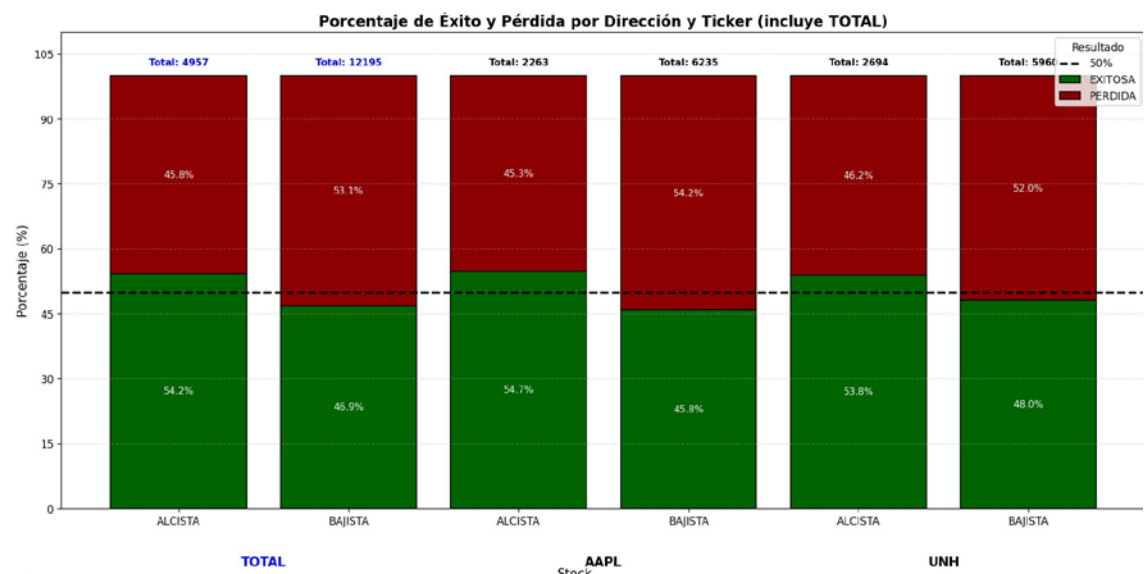


Figura 40 Resultados comparativos de los activos en la predicción de precios de cierre

En la Figura 41 se muestran los resultados comparativos de cada acción tras realizar predicciones sobre el porcentaje de cambio, para comparar si se comporta de una forma parecida independientemente del activo:

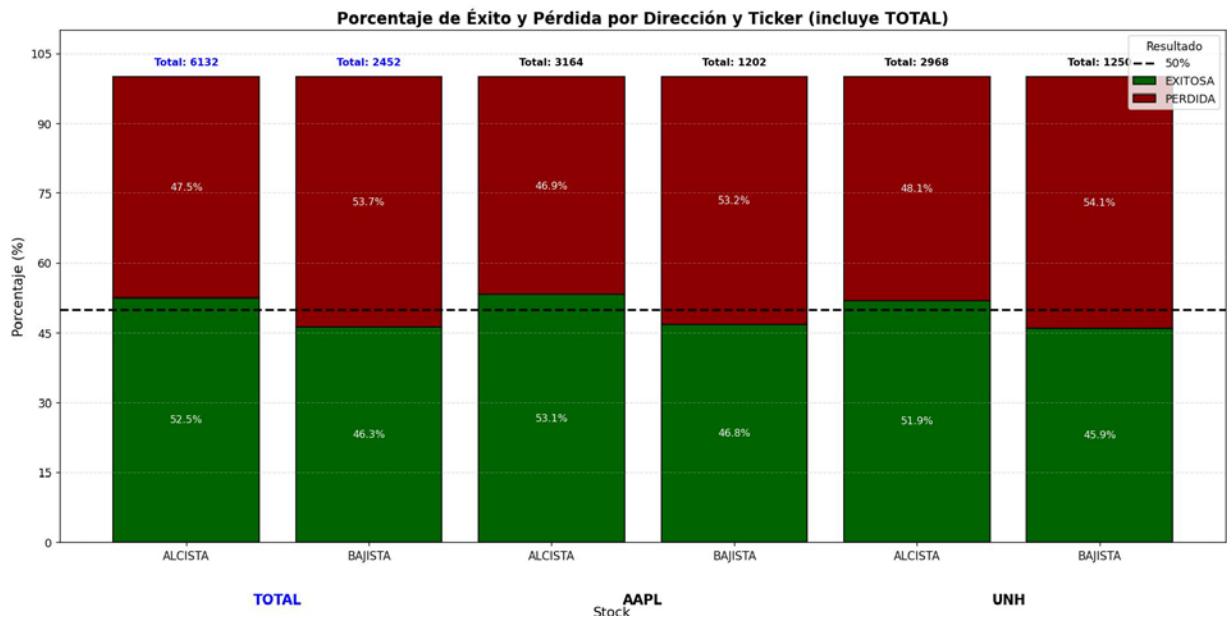


Figura 41 Resultados comparativos de los activos en la predicción del porcentaje de cambio

7.2.2 Num lookback

En la Figura 42 se muestran los resultados comparativos de cada *num lookback* tras realizar predicciones sobre los precios de cierre:

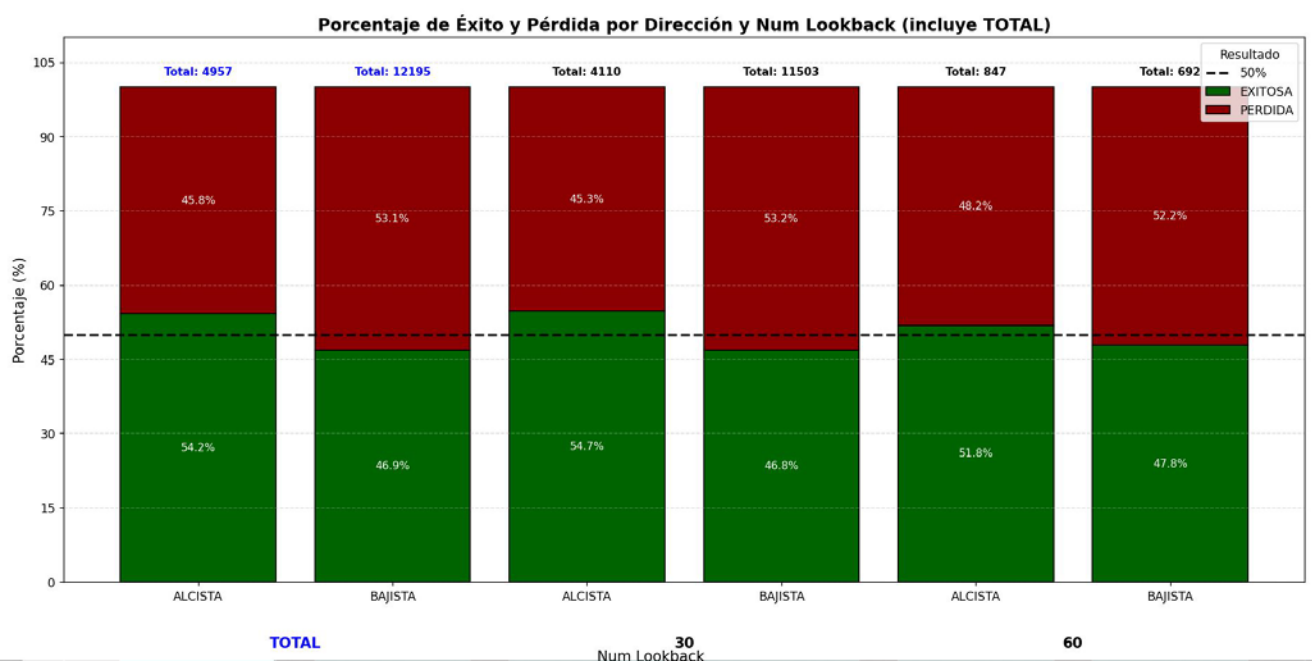


Figura 42 Resultados comparativos de los diferentes *num lookback* en la predicción de precios de cierre

Se observa que el *num_lookback* con mayor porcentaje de acierto es el 30 para operaciones alcistas.

En la Figura 43 se muestran los resultados comparativos de cada *num lookback* tras realizar predicciones sobre el porcentaje de cambio:

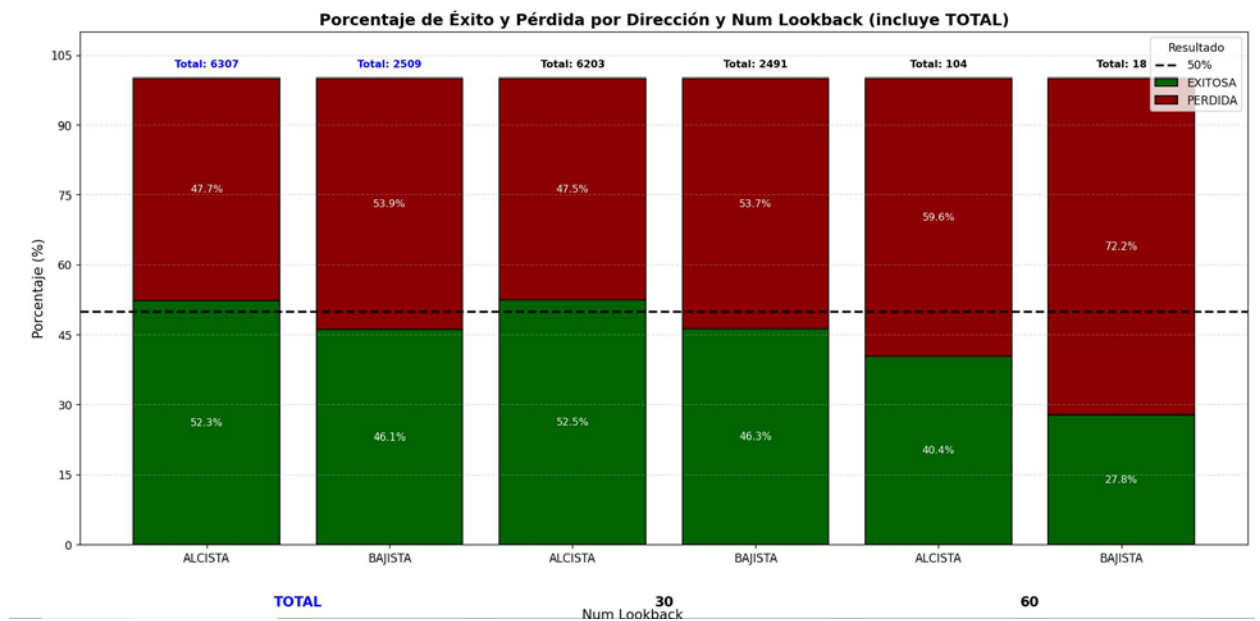


Figura 43 Resultados comparativos de los diferentes *num lookback* en la predicción del porcentaje de cambio

Se observa que el *num_lookback* con mayor porcentaje de acierto es el 30 para operaciones alcistas.

7.2.3 Num forecast

En la Figura 44 se muestran los resultados comparativos de cada *num forecast* tras realizar predicciones sobre los precios de cierre:

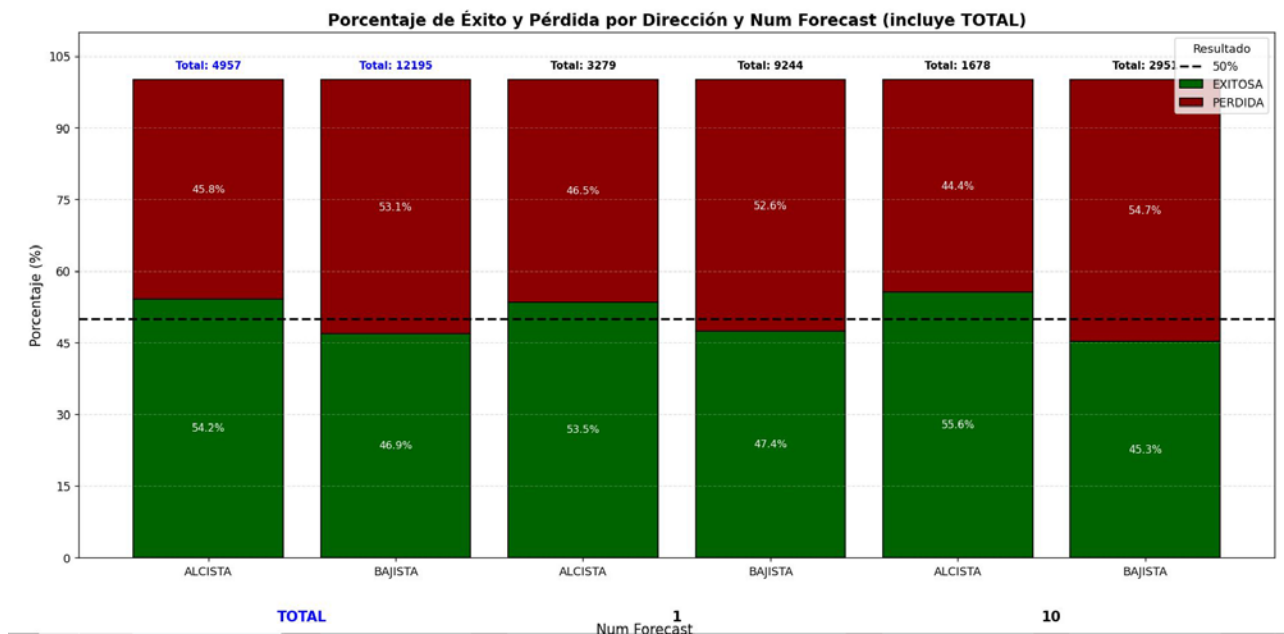


Figura 44 Resultados comparativos de los diferentes *num forecast* en la predicción de precios de cierre

Se observa que el *num_forecast* con mayor porcentaje de acierto es el 10 para operaciones alcistas.

En la Figura 45 se muestran los resultados comparativos de cada *num forecast* tras realizar predicciones sobre el porcentaje de cambio:

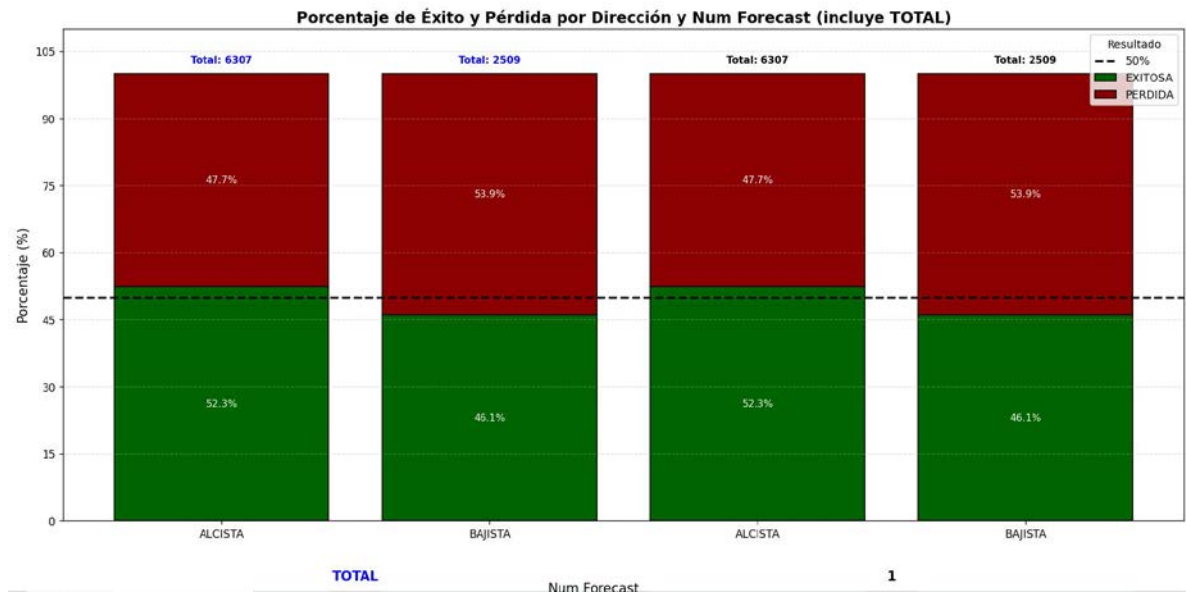


Figura 45 Resultados comparativos de los diferentes *num forecast* en la predicción del porcentaje de cambio

En el caso de predicción de porcentaje de cambio, solo habrá un *num_forecast* posible, que es al día siguiente, debido a que solo se ha programado la posibilidad de predecir la siguiente muestra.

7.2.4 Batch size

En la Figura 46 se muestran los resultados comparativos de cada *batch size* tras realizar predicciones sobre los precios de cierre:

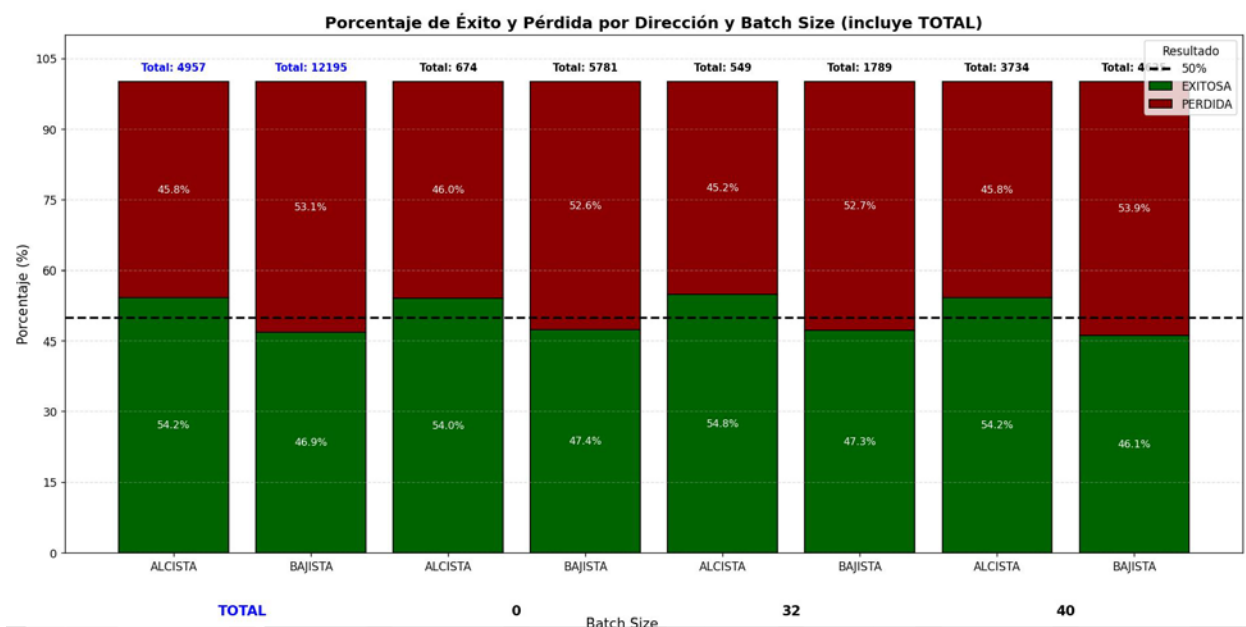


Figura 46 Resultados comparativos de los diferentes *batch size* en la predicción de precios de cierre

Se observa que el *batch_size* con mayor porcentaje de acierto es el 32 para operaciones alcistas.

En la Figura 47 se muestran los resultados comparativos de cada *batch size* tras realizar predicciones sobre el porcentaje de cambio:

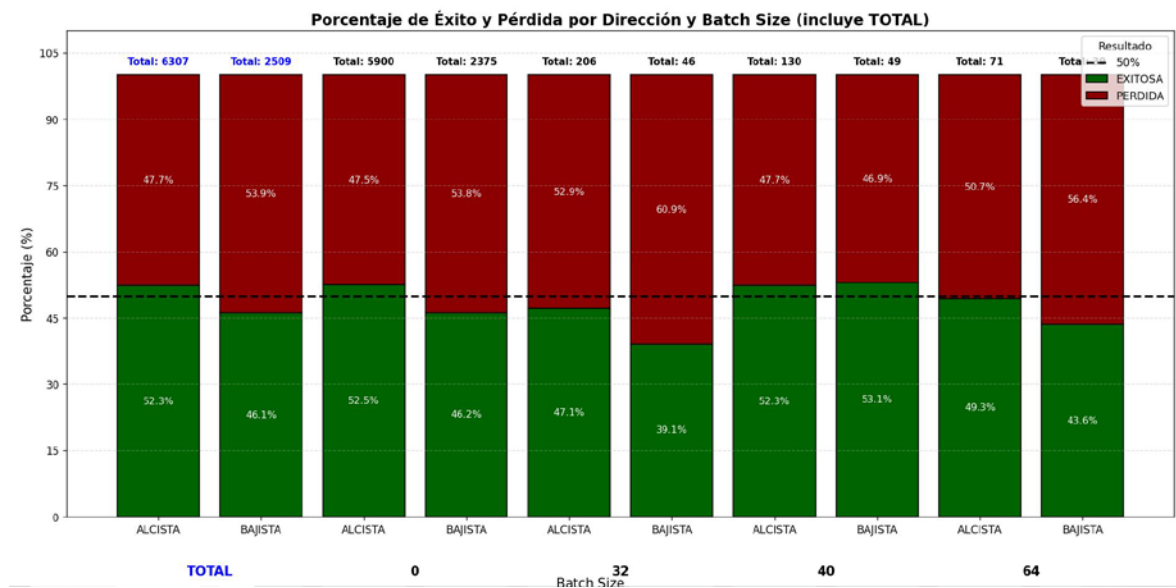


Figura 47 Resultados comparativos de los diferentes *batch size* en la predicción del porcentaje de cambio

Se observa que el *batch size* con mayor porcentaje de acierto es el 40 para operaciones alcistas.

Se resalta que el *batch size* de 0 solo es aplicable para arquitecturas que no utilizan *batch size*, que son todas menos las basadas en Redes Neuronales (LSTM y GRU).

7.2.5 Epochs

En la Figura 48 se muestran los resultados comparativos de cada *epochs* tras realizar predicciones sobre los precios de cierre:

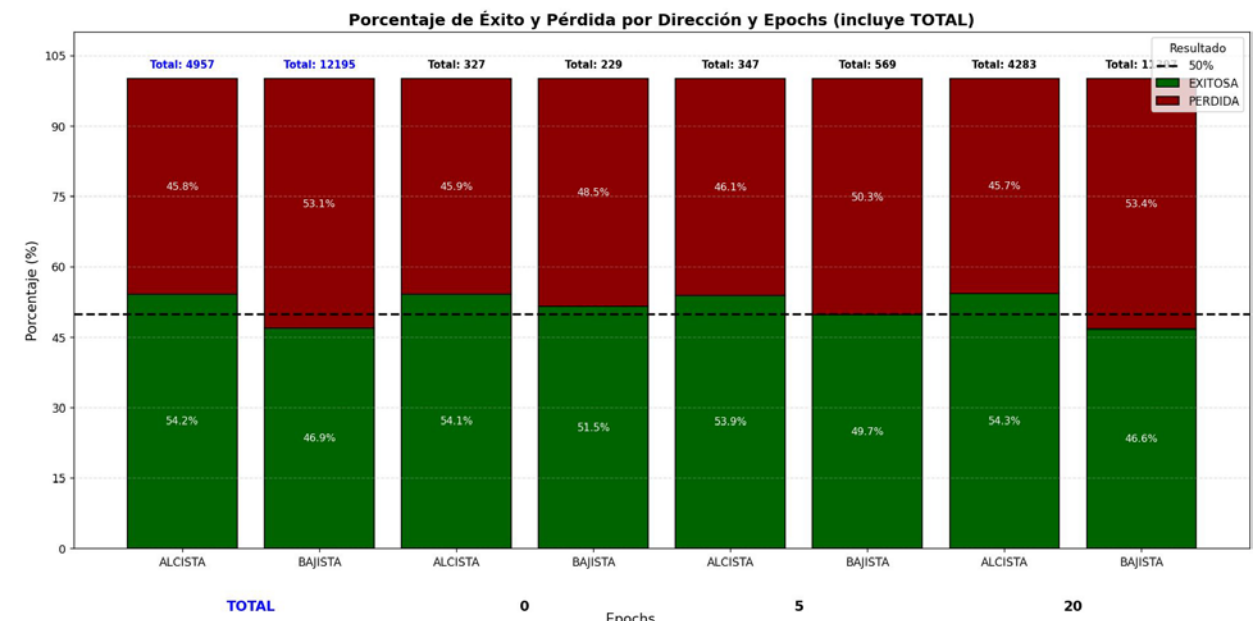


Figura 48 Resultados comparativos de los diferentes *epochs* en la predicción de precios de cierre

Se observa que *epochs* con mayor porcentaje de acierto es el 20 para operaciones alcistas.

En la Figura 49 se muestran los resultados comparativos de cada *epochs* tras realizar predicciones sobre el porcentaje de cambio:

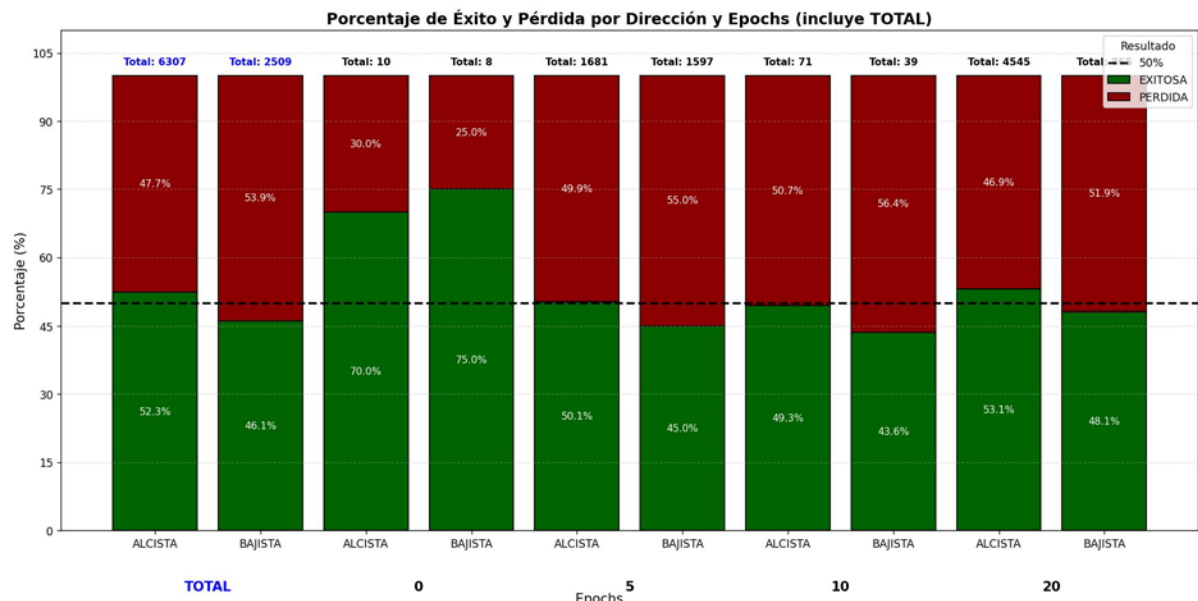


Figura 49 Resultados comparativos de los diferentes *epochs* en la predicción del porcentaje de cambio

Se observa que *epochs* con mayor porcentaje de acierto es el 20 para operaciones alcistas.

Se resalta que el *epochs* de 0 solo es aplicable para arquitecturas que no utilizan *epochs*, que son todas menos las basadas en Redes Neuronales (LSTM y GRU).

7.2.6 Learning rate

En la Figura 50 se muestran los resultados comparativos de cada *learning rate* tras realizar predicciones sobre los precios de cierre:

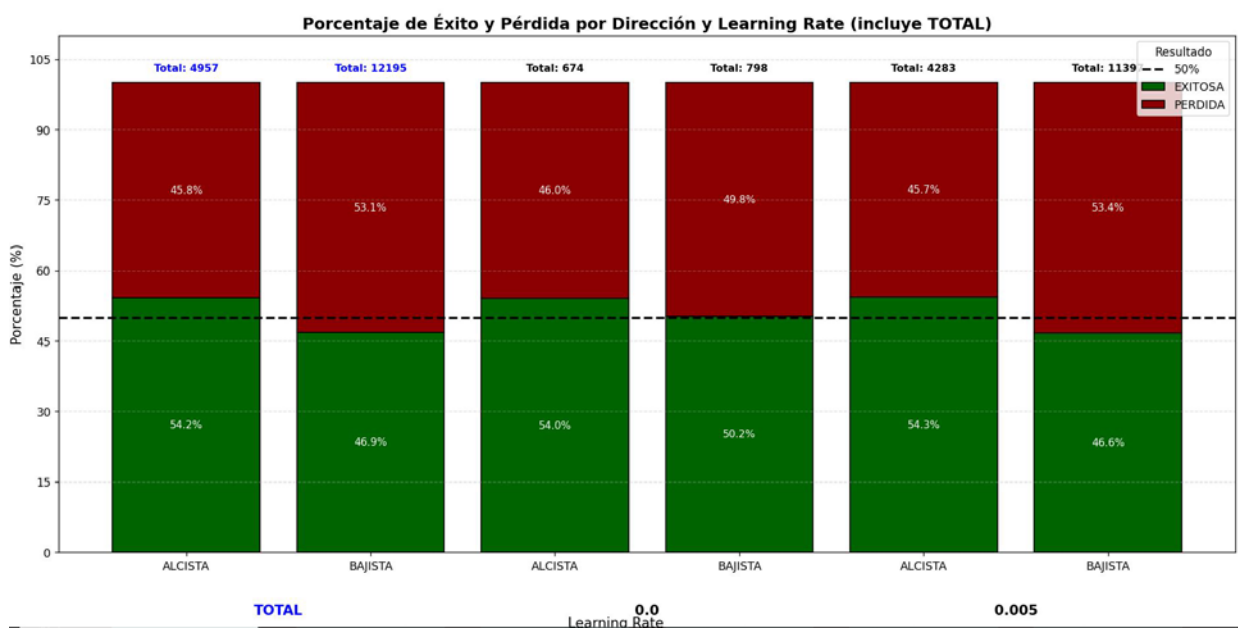


Figura 50 Resultados comparativos de los diferentes *learning rate* en la predicción de precios de cierre

Se observa que el *learning rate* con mayor porcentaje de acierto es el 0.005 para operaciones alcistas.

En la Figura 51 se muestran los resultados comparativos de cada *learning rate* tras realizar predicciones sobre el porcentaje de cambio:

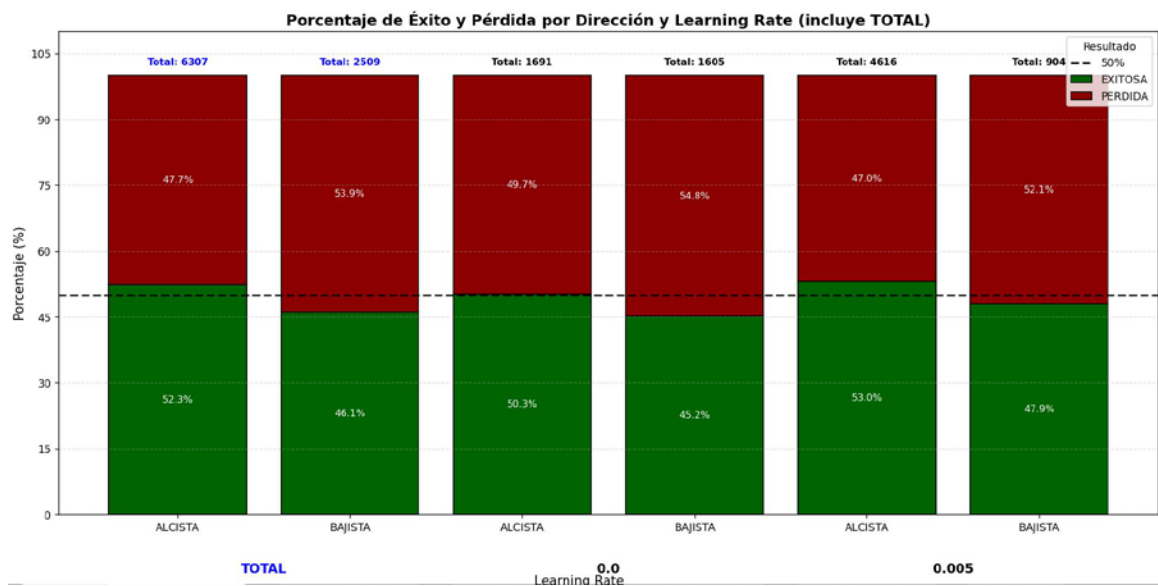


Figura 51 Resultados comparativos de los diferentes *learning rate* en la predicción del porcentaje de cambio

Se resalta que el *learning_rate* de 0 solo es aplicable para arquitecturas que no utilizan *learning_rate*, que son todas menos las basadas en Redes Neuronales (LSTM y GRU).

7.2.7 Variable a predecir

Por último, se llevará a cabo una comparación entre las variables consideradas como objetivo de predicción (el **precio de cierre** y el **porcentaje de cambio**) con el fin de evaluar cuál de ellas ofrece un mejor desempeño en términos de ajuste y capacidad predictiva. De esta manera, será posible determinar si alguno de los dos enfoques resulta más adecuado para el modelo propuesto, ya sea porque capta mejor la dinámica de la serie o porque proporciona pronósticos más precisos y estables.

En la Figura 52 se muestran los resultados comparativos de cada variable a predecir:

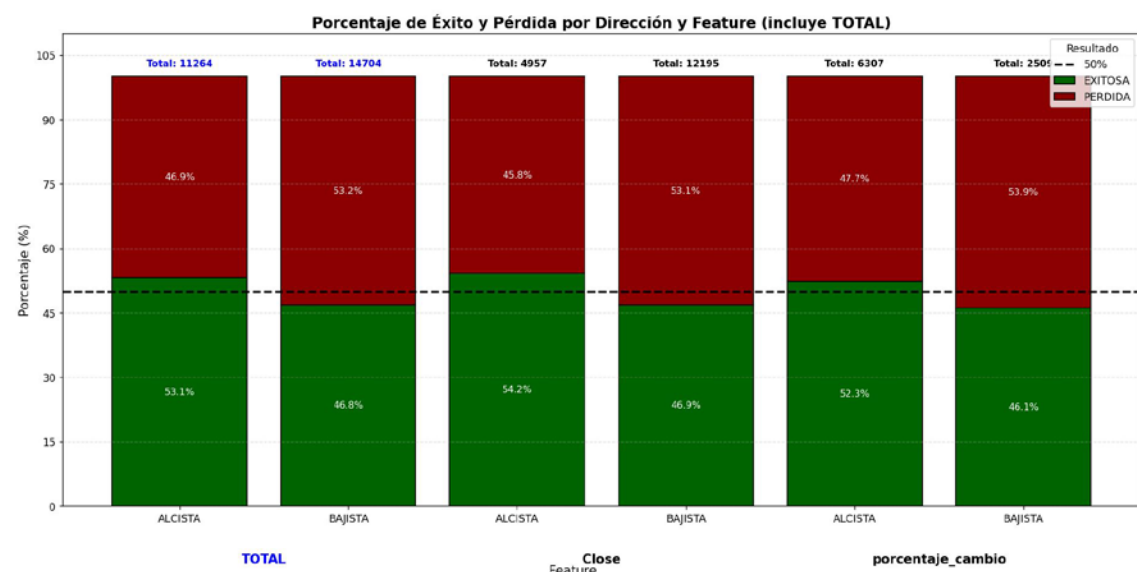


Figura 52 Resultados comparativos de las diferentes variables a predecir

Se observa que, a la hora de predecir los precios de cierre, se comporta mejor que predecir los porcentajes de cambio.

7.3 Correlación entre diferentes parámetros

En este apartado se realizará un análisis de la correlación entre los diferentes parámetros empleados en las predicciones, con el objetivo de identificar qué relaciones pueden influir en un mayor porcentaje de éxito. Este estudio permitirá detectar dependencias significativas o redundancias entre las variables, aportando información valiosa para optimizar la selección de parámetros y mejorar la consistencia de los resultados. Además, este enfoque facilitará la construcción de estrategias más eficientes, centradas en aquellos factores que realmente aportan valor predictivo.

7.3.1 Gráficos de dispersión (scatterplots)

Un gráfico de dispersión es una herramienta estadística que permite representar la relación entre dos variables continuas, ubicando cada punto en el plano según sus coordenadas en los ejes **X** e **Y**. A través del patrón que forman los puntos, es posible identificar si existe correlación entre las variables, así como su dirección (positiva o negativa) y el tipo de relación (lineal o no lineal). Este tipo de gráfico resulta especialmente útil para explorar dependencias y tendencias entre variables, facilitando un análisis visual claro y directo [55].

En la Figura 53 se muestran los resultados comparativos entre varias variables y su correlación:

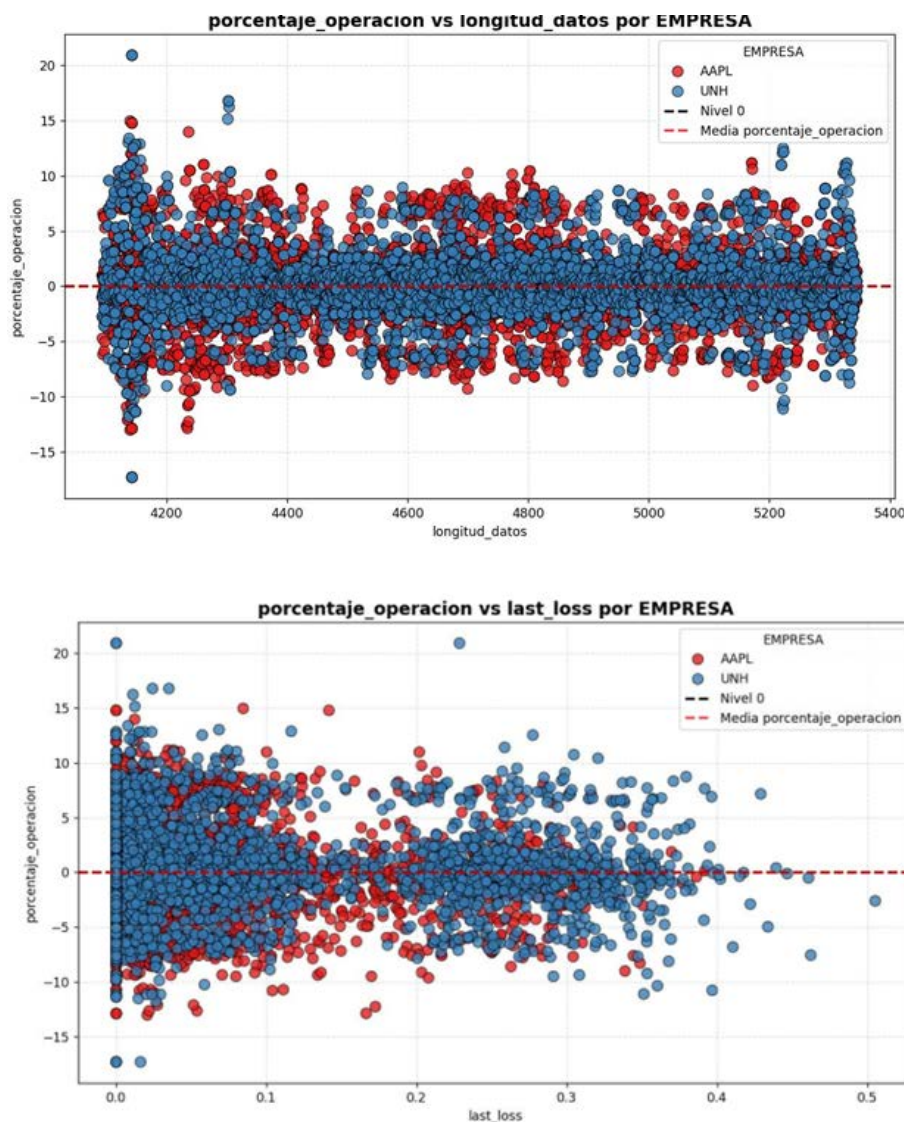


Figura 53 Correlación entre diferentes variables

No se observa una correlación entre el porcentaje de operación y la longitud de datos por empresa, que son la cantidad de datos históricos utilizados para hacer la predicción, lo que indica que por más datos que tengamos para entrenar, no llega a generar mejores predicciones.

Tampoco se observa una correlación entre el porcentaje de operación y el valor de pérdida resultante de cada predicción, lo que indica que los valores de pérdida resultantes de la fase de entrenamiento no son relevantes, y son variaciones muy pequeñas.

7.3.2 Diagramas de cajas (boxplots)

Un *boxplot*, es una herramienta gráfica que permite resumir y visualizar la distribución de un conjunto de datos de manera compacta. Este tipo de gráfico destaca valores estadísticos clave, como la mediana y los posibles valores atípicos (*outliers*), lo que facilita la identificación de patrones y anomalías dentro de los datos. Generalmente, se utiliza con variables de escala métrica, como la edad de una persona, el consumo anual de electricidad o la temperatura.

La caja central del gráfico representa el rango intercuartílico, es decir, el intervalo donde se concentra la mayor parte de los datos. Dentro de esta caja aparece una línea que indica la mediana. Los “bigotes” del boxplot se extienden hasta el valor máximo y mínimo observados que no se consideran *outliers*, mostrando así el rango total de los datos sin valores extremos. De esta forma, el *boxplot* permite identificar rápidamente la tendencia central, la variabilidad y la presencia de valores anómalos, resultando muy útil para comparar la distribución de varias muestras o grupos de datos [56].

En la Figura 54 se muestran diferentes diagramas de cajas entre diferentes variables:

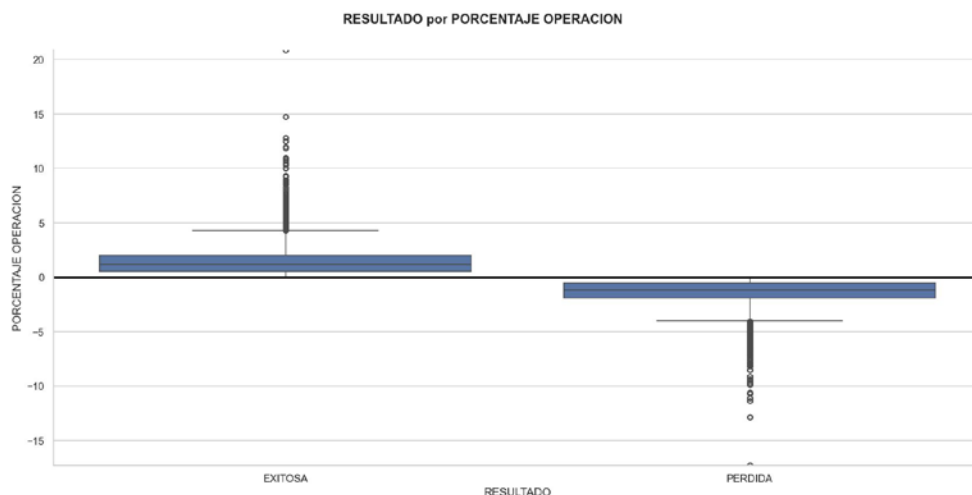


Figura 54 Diagramas de cajas entre diferentes variables

Vemos que tanto las operaciones exitosas como perdidas nos dan un porcentaje similar en valor absoluto, lo que se espera que las pérdidas no generen un porcentaje total mayor que las ganancias.

7.3.3 Matriz de correlación

La matriz de correlación es una herramienta estadística utilizada para cuantificar y comparar las relaciones entre diferentes variables de un conjunto de datos. En ella se muestran, en formato tabular, las correlaciones por pares entre todas las combinaciones posibles de dos variables. Cada celda de la matriz contiene el denominado coeficiente de correlación, calculado entre la variable de la fila y la de la columna correspondiente.

Este coeficiente toma valores entre -1 y 1 e indica el tipo y la intensidad de la relación entre las variables. Un valor positivo refleja una correlación directa: a medida que una variable aumenta, la otra también lo hace. Cuanto más cercano a 1, más fuerte es esta relación. En cambio, un valor negativo muestra una correlación inversa: el incremento en una variable implica la disminución en la otra. Por último, un valor de 0 señala ausencia de correlación.

De esta manera, la matriz de correlación permite visualizar de forma rápida y sencilla las dependencias existentes en los datos, sirviendo como base para análisis posteriores, como la selección de modelos. Además, facilita la detección de problemas como la multicolinealidad (situación en un análisis de regresión donde las variables independientes están altamente correlacionadas entre sí), que puede afectar a los modelos de regresión al distorsionar la estimación de los parámetros [57].

En la Figura 55 se muestra la matriz de correlación entre todas las variables:

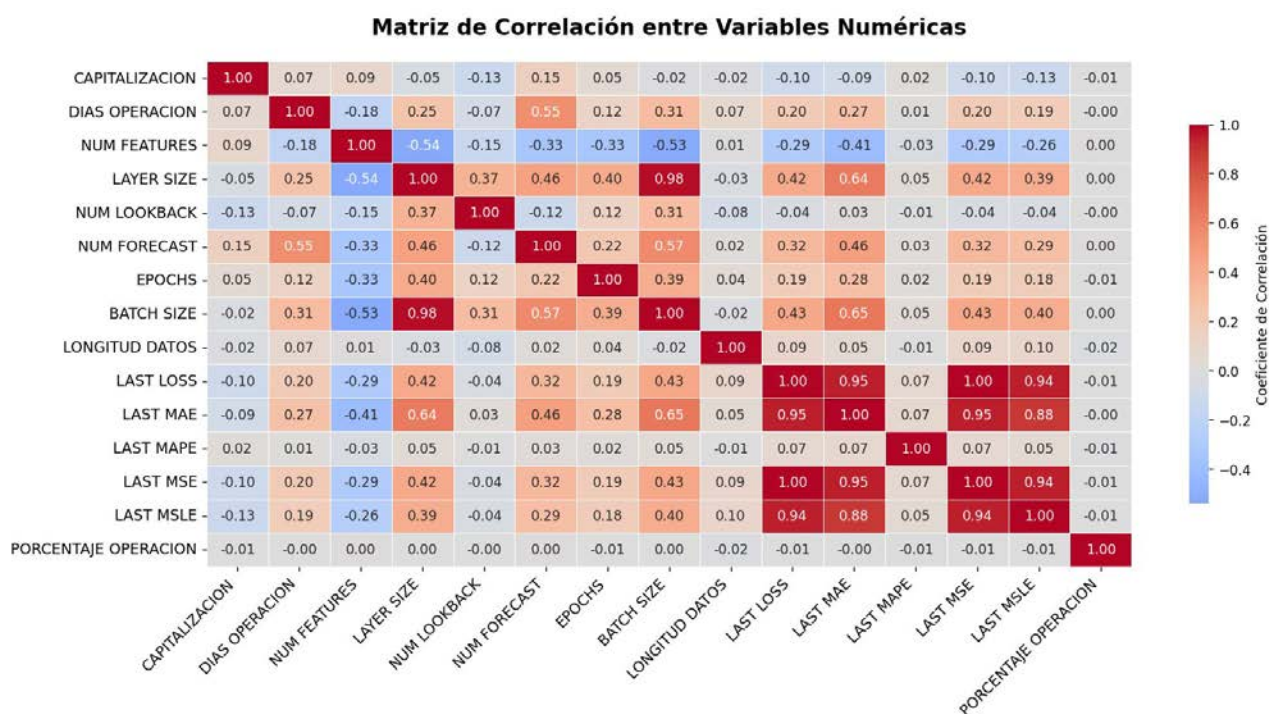


Figura 55 Matriz de correlación

Se observa la correlación entre todos los indicadores, pudiendo generar gráficos de dispersión para observar la verdadera correlación entre 2 variables y pudiendo hacer mejores filtros. Los datos son interpretables, que los errores LAST MSE esté correlacionado con LAST MAE no transmite nada, pero que LAST MAE esté correlacionado con *batch size* nos podría indicar que aumentando el *batch size* se podrían generar errores menores, y teóricamente mejores resultados.

8 BACKTESTING Y EVALUACIÓN DE ESTRATEGIAS DE INVERSIÓN

8.1 Definición e importancia del *backtesting*

El término *backtesting* se utiliza en finanzas con distintos significados. De forma general, puede referirse a la evaluación del desempeño hipotético que habría tenido una estrategia de inversión aplicada a datos históricos, o puede referirse a la validación de modelos de riesgo financiero mediante la comparación entre pronósticos históricos de riesgo y las pérdidas y ganancias realmente observadas. En este caso, el enfoque se centra en la evaluación del que habría tenido una estrategia de inversión aplicada a datos históricos [58].

En el ámbito del análisis financiero con datos del mercado de valores, el *backtesting* es una herramienta clave para lograr explicabilidad y aumentar las posibilidades de adopción real de la investigación. El *backtesting* utiliza datos históricos para evaluar retrospectivamente la viabilidad de un modelo y generar confianza en su uso futuro. La lógica subyacente es sencilla: una estrategia que ha mostrado buenos resultados en el pasado tiene mayores probabilidades de funcionar bien en el futuro, y lo contrario ocurre si los resultados históricos han sido deficientes.

Diversos estudios han analizado la aplicación del aprendizaje profundo a los mercados financieros, destacando que los casos más relevantes para la comunidad investigadora se encuentran en el trading algorítmico y la gestión de portafolios, debido a la complejidad de los datos bursátiles. Algunos trabajos se han centrado en la reproducibilidad, aunque en muchos, pese a presentar métricas financieras, no se evidencia el uso de *backtesting* ni se aborda su aplicabilidad práctica.

En este trabajo se hará uso del *backtesting* como herramienta para validar los resultados obtenidos a partir de las predicciones generadas por los modelos implementados. De esta manera, se podrá comprobar si las estrategias modeladas presentan un comportamiento consistente y aplicable en escenarios reales [59].

Las estrategias modernas de inversión se basan en la identificación de patrones que puedan cuantificarse y aprovecharse de manera sistemática. Por ejemplo, es posible diseñar algoritmos que obtengan beneficios a partir de fenómenos como el *momentum*, es decir, la tendencia de muchos activos a mostrar rachas prolongadas de ganancias o pérdidas, más allá de lo que cabría esperar de valores que siguieran un comportamiento aleatorio. Una de las ventajas de esta sistematización es que dichos algoritmos pueden someterse a *backtesting*, lo cual consiste en simular históricamente cómo

habría funcionado una estrategia en el pasado. Este tipo de pruebas resulta muy valioso, ya que permite evaluar la relación riesgo/retorno antes de comprometer capital real [60].

Sin embargo, este tipo de backtesting no está exento de limitaciones, siendo el **sobreajuste (overfitting)** uno de los problemas más relevantes. Cuando ocurre, la estrategia muestra un rendimiento excepcional en los datos históricos porque se ha ajustado demasiado a ellos, capturando ruido en lugar de patrones genuinos del mercado. Como consecuencia, al aplicarla en tiempo real, el desempeño suele ser mucho peor al esperado, lo que genera decisiones de inversión poco fiables y un riesgo elevado de pérdidas.

En particular, las estrategias más complejas (aquellas que incorporan un mayor número de reglas de trading, filtros y parámetros) parecen ser más propensas a sufrir problemas de sobreajuste en los *backtesting* [61].

A pesar de todo, el *backtesting* resulta fundamental, ya que permite evaluar la posible eficacia y rentabilidad de las estrategias antes de aplicarlas en el mercado real. Al simular condiciones históricas y poner a prueba la estrategia sobre datos pasados, se obtiene una visión más precisa de cómo habría funcionado en distintos escenarios, ofreciendo así una referencia útil para decidir si merece la pena implementarla [62].

8.2 Diseño de estrategias basadas en predicciones

En el presente trabajo no se lleva a cabo una selección de indicadores técnicos con el objetivo de identificar patrones y explotarlos de forma sistemática. En su lugar, el enfoque se centra en la **predicción de valores futuros** mediante la aplicación de distintos modelos sobre diversos activos financieros. Esta aproximación reduce considerablemente el riesgo de *overfitting*, ya que la validez de un modelo puede evaluarse de manera más robusta: si resulta realmente efectivo, debería demostrar un desempeño consistente tanto en operaciones alcistas como bajistas, y además en diferentes tipos de activos.

A partir de los ficheros de resultados obtenidos en el proceso de modelado (un ejemplo de los cuales se presenta en la Tabla 2), se procede a la construcción de estrategias de inversión. Dichas estrategias permiten evaluar, de manera sistemática, cómo se habrían comportado las decisiones de trading bajo distintos escenarios y marcos temporales. El objetivo es no solo medir la rentabilidad de las estrategias propuestas, sino también contrastarlas frente a un referente sólido del mercado como lo es el índice S&P 500.

El análisis de los resultados se lleva a cabo en diferentes horizontes temporales: diario, mensual y anual. No obstante, con el fin de simplificar la interpretación y evitar una sobrecarga de información, en este trabajo se presentan únicamente los comportamientos de manera mensual y anual. Este enfoque permite identificar tendencias más claras y eliminar parte del ruido propio de las variaciones diarias, ofreciendo una visión más estructurada sobre la eficacia de las estrategias.

Se mostrarán los resultados incluyendo métricas relevantes como el **porcentaje de éxito** (esto es, la proporción de operaciones rentables frente al total), el **porcentaje de todas las operaciones** y, de forma comparativa, el comportamiento del índice **S&P 500** en ese mismo periodo. De esta manera, es posible evaluar no solo el rendimiento absoluto de las estrategias, sino también su capacidad relativa para superar a un índice de referencia ampliamente reconocido en los mercados financieros.

Según los resultados del análisis de data science, se observa que las mejores variables predictivas, para los casos analizados, se obtienen al trabajar con operaciones alcistas y emplear una arquitectura LSTM. La configuración que ofrece un mejor rendimiento utiliza una ventana temporal de 30 observaciones históricas (`num_lookback = 30`) y un horizonte de predicción de 10 pasos (`num_forecast = 10`). El caso de la predicción del porcentaje de cambio, el único valor posible para esta variable es 1.

En función de la variable a predecir, se aprecian diferencias en los parámetros de entrenamiento. Cuando el objetivo del modelo es el precio de cierre, los mejores resultados se alcanzan con un batch_size de 32, mientras que para la predicción del porcentaje de cambio el rendimiento mejora con un batch_size de 40. En ambos casos, el entrenamiento se realiza con 20 epochs y un learning_rate de 0.05, parámetros que proporcionan un compromiso adecuado entre estabilidad y capacidad de aprendizaje.

Asimismo, para el análisis se han utilizado los ficheros correspondientes a las predicciones del 3 % en el caso del precio de cierre y del 1 % en el caso del porcentaje de cambio, ya que ambos presentan un número de operaciones aceptable y un porcentaje de acierto suficientemente representativo para extraer conclusiones fiables.

Por último, los resultados indican que las predicciones basadas en precios de cierre esperan ofrecer un mejor rendimiento frente a las basadas en porcentaje de cambio. No obstante, esta ventaja deberá validarse mediante un backtesting posterior, que permitirá comprobar si la mejora observada a nivel predictivo se traduce en un comportamiento más eficiente.

Los resultados se analizarán diferenciando entre las **predicciones del precio de cierre** y las del **porcentaje de cambio**, con el fin de evaluar el rendimiento de cada enfoque por separado. Tal como se ha señalado previamente, la **entrada** en la operación se establece al cierre de la sesión bursátil correspondiente al día de la predicción. La **salida**, en cambio, se produce al cierre del día en que el porcentaje acumulado de la operación supera el umbral porcentual definido.

En la Tabla 3 se muestran los resultados **mensuales** obtenidos a partir del total de operaciones analizadas, considerando específicamente las predicciones sobre los **precios de cierre**. Se presentan los resultados de cada periodo, donde la penúltima columna muestra el promedio y la última el total acumulado sin distinguir entre periodos:

FECHA INICIO	FECHA FIN	TOTAL OPERACIONES	Nº EXITOSAS	% EXITOSAS (VS PERDIDAS)	TOTAL PORCENTAJE OPERACION	% VARIACION SP
2020-01-01	2020-01-31	32	19	59,38	12,99	-0,04
2020-02-01	2020-02-29	48	19	39,58	-79,84	-7,92
2020-03-01	2020-03-31	86	45	52,33	126,27	-16,12
2020-04-01	2020-04-30	54	35	64,81	67,25	12,7
2020-05-01	2020-05-31	61	33	54,1	30,76	4,76
2020-06-01	2020-06-30	48	31	64,58	16,16	1,36
2020-07-01	2020-07-31	50	34	68	60,88	5,89
2020-08-01	2020-08-31	44	32	72,73	61,31	6,98
2020-09-01	2020-09-30	50	20	40	-45,63	-3,74
2020-10-01	2020-10-31	47	20	42,55	-32,89	-2,49
2020-11-01	2020-11-30	44	26	59,09	25	9,65
2020-12-01	2020-12-31	50	39	78	105,52	3,7
2021-01-01	2021-01-31	47	19	40,43	-16,37	-1,02
2021-02-01	2021-02-28	52	19	36,54	-47,13	1,1
2021-03-01	2021-03-31	45	30	66,67	38,17	2,07
2021-04-01	2021-04-30	46	31	67,39	65,97	5,29
2021-05-01	2021-05-31	38	18	47,37	-6,22	0,66
2021-06-01	2021-06-30	41	24	58,54	25,82	2,33
2021-07-01	2021-07-31	35	25	71,43	22,06	2,44
2021-08-01	2021-08-31	37	21	56,76	27,87	3,19

2021-09-01	2021-09-30	43	14	32,56	-50,13	-4,66
2021-10-01	2021-10-31	41	30	73,17	40,28	7,02
2021-11-01	2021-11-30	42	23	54,76	4,79	-0,97
2021-12-01	2021-12-31	34	25	73,53	53,04	4,62
2022-01-01	2022-01-31	57	29	50,88	-0,82	-5,27
2022-02-01	2022-02-28	51	26	50,98	3,53	-2,95
2022-03-01	2022-03-31	63	39	61,9	33,9	3,76
2022-04-01	2022-04-30	38	15	39,47	-43,12	-8,78
2022-05-01	2022-05-31	57	31	54,39	-12,19	-0,37
2022-06-01	2022-06-30	50	21	42	-35,24	-8,25
2022-07-01	2022-07-31	35	28	80	70,23	9,21
2022-08-01	2022-08-31	25	13	52	10,6	-3,8
2022-09-01	2022-09-30	14	4	28,57	-14,58	-9,24
2022-10-01	2022-10-31	18	14	77,78	24,55	8,13
2022-11-01	2022-11-30	27	14	51,85	-8,88	5,56
2022-12-01	2022-12-31	23	8	34,78	-35,36	-5,76
2023-01-01	2023-01-31	25	9	36	-16,7	6,74
2023-02-01	2023-02-28	24	8	33,33	-27,6	-2,51
2023-03-01	2023-03-31	25	12	48	-4,6	3,71
2023-04-01	2023-04-30	21	13	61,9	10,4	1,6
2023-05-01	2023-05-31	12	5	41,67	2,1	0,56
2023-06-01	2023-06-30	16	9	56,25	5,97	6,48
2023-07-01	2023-07-31	12	5	41,67	-3,1	3,27
2023-08-01	2023-08-31	36	18	50	7,33	-1,63
2023-09-01	2023-09-30	29	12	41,38	-18,67	-4,74
2023-10-01	2023-10-31	41	22	53,66	-5,02	-2,13
2023-11-01	2023-11-30	36	28	77,78	44,24	9,13
2023-12-01	2023-12-31	58	34	58,62	9,3	4,57
2024-01-01	2024-01-31	70	26	37,14	-48,44	2,16
2024-02-01	2024-02-29	55	23	41,82	-49,05	5,22
2024-03-01	2024-03-31	49	22	44,9	-10,21	3,27
2024-04-01	2024-04-30	62	28	45,16	-20,2	-3,86
2024-05-01	2024-05-31	66	41	62,12	54,72	5,06
2024-06-01	2024-06-30	48	32	66,67	65,76	3,53
2024-07-01	2024-07-31	20	16	80	23,31	1
2024-08-01	2024-08-31	14	7	50	1,41	2,34
2024-09-01	2024-09-30	18	4	22,22	-12,18	4,25
2024-10-01	2024-10-31	26	14	53,85	1,19	-0,89
2024-11-01	2024-11-30	33	21	63,64	23,04	5,96
2024-12-01	2024-12-31	46	27	58,7	-8,09	-2,58
2025-01-01	2025-01-31	0	0	0	0	2,69
2020-01-01	2025-01-31	39,59	21,48	52,88	8,58	1,18
2020-01-01	2025-01-01	2415	1310	54,24	523,46	96,4

Tabla 3 Resultados mensuales de las operaciones según la predicción de los precios de cierre

En la Tabla 4 se muestran los resultados **mensuales** obtenidos a partir del total de operaciones analizadas, considerando específicamente las predicciones sobre el **porcentaje de cambio**. Se presentan los resultados de cada periodo, donde la penúltima columna muestra el promedio y la última el total acumulado sin distinguir entre periodos:

FECHA INICIO	FECHA FIN	TOTAL OPERACIONES	Nº EXITOSAS	Nº PERDIDAS	% EXITOSAS (VS PERDIDAS)	TOTAL % OPERACION	% VARIACION SP
2020-01-01	2020-01-31	11	7	4	63,64	2,38	-0,04
2020-02-01	2020-02-29	6	1	5	16,67	-3,99	-7,92
2020-03-01	2020-03-31	6	4	2	66,67	12,17	-16,12
2020-04-01	2020-04-30	17	9	8	52,94	8,26	12,7
2020-05-01	2020-05-31	15	10	5	66,67	9,07	4,76
2020-06-01	2020-06-30	7	4	3	57,14	0,83	1,36
2020-07-01	2020-07-31	1	0	1	0	-0,2	5,89
2020-08-01	2020-08-31	7	5	2	71,43	5,43	6,98
2020-09-01	2020-09-30	5	2	3	40	-7,45	-3,74
2020-10-01	2020-10-31	0	0	0	0	0	-2,49
2020-11-01	2020-11-30	5	2	3	40	-1,19	9,65
2020-12-01	2020-12-31	2	2	0	100	1,76	3,7
2021-01-01	2021-01-31	3	2	1	66,67	0,17	-1,02
2021-02-01	2021-02-28	1	1	0	100	0,13	1,1
2021-03-01	2021-03-31	2	0	2	0	-1,41	2,07
2021-04-01	2021-04-30	0	0	0	0	0	5,29
2021-05-01	2021-05-31	1	0	1	0	-0,53	0,66
2021-06-01	2021-06-30	0	0	0	0	0	2,33
2021-07-01	2021-07-31	1	1	0	100	1,96	2,44
2021-08-01	2021-08-31	0	0	0	0	0	3,19
2021-09-01	2021-09-30	0	0	0	0	0	-4,66
2021-10-01	2021-10-31	0	0	0	0	0	7,02
2021-11-01	2021-11-30	0	0	0	0	0	-0,97
2021-12-01	2021-12-31	0	0	0	0	0	4,62
2022-01-01	2022-01-31	0	0	0	0	0	-5,27
2022-02-01	2022-02-28	1	1	0	100	2,01	-2,95
2022-03-01	2022-03-31	0	0	0	0	0	3,76
2022-04-01	2022-04-30	1	0	1	0	-2	-8,78
2022-05-01	2022-05-31	2	2	0	100	1,01	-0,37
2022-06-01	2022-06-30	4	1	3	25	-0,61	-8,25
2022-07-01	2022-07-31	3	0	3	0	-2,43	9,21
2022-08-01	2022-08-31	3	0	3	0	-6,34	-3,8
2022-09-01	2022-09-30	4	2	2	50	0,75	-9,24
2022-10-01	2022-10-31	5	3	2	60	-0,28	8,13
2022-11-01	2022-11-30	1	1	0	100	1,47	5,56
2022-12-01	2022-12-31	2	0	2	0	-3,77	-5,76
2023-01-01	2023-01-31	4	3	1	75	2,2	6,74
2023-02-01	2023-02-28	3	1	2	33,33	-4,24	-2,51

2023-03-01	2023-03-31	1	1	0	100	0,84	3,71
2023-04-01	2023-04-30	2	2	0	100	1,36	1,6
2023-05-01	2023-05-31	0	0	0	0	0	0,56
2023-06-01	2023-06-30	0	0	0	0	0	6,48
2023-07-01	2023-07-31	2	1	1	50	-0,58	3,27
2023-08-01	2023-08-31	0	0	0	0	0	-1,63
2023-09-01	2023-09-30	0	0	0	0	0	-4,74
2023-10-01	2023-10-31	0	0	0	0	0	-2,13
2023-11-01	2023-11-30	0	0	0	0	0	9,13
2023-12-01	2023-12-31	0	0	0	0	0	4,57
2024-01-01	2024-01-31	0	0	0	0	0	2,16
2024-02-01	2024-02-29	0	0	0	0	0	5,22
2024-03-01	2024-03-31	1	0	1	0	-1,81	3,27
2024-04-01	2024-04-30	0	0	0	0	0	-3,86
2024-05-01	2024-05-31	0	0	0	0	0	5,06
2024-06-01	2024-06-30	0	0	0	0	0	3,53
2024-07-01	2024-07-31	0	0	0	0	0	1
2024-08-01	2024-08-31	0	0	0	0	0	2,34
2024-09-01	2024-09-30	0	0	0	0	0	4,25
2024-10-01	2024-10-31	0	0	0	0	0	-0,89
2024-11-01	2024-11-30	1	0	1	0	-1,69	5,96
2024-12-01	2024-12-31	0	0	0	0	0	-2,58
2025-01-01	2025-01-31	0	0	0	0	0	2,69
2020-01-01	2025-01-31	2,13	1,11	1,02	26,81	0,22	1,18
2020-01-01	2025-01-01	130	68	62	52,31	13,28	96,4

Tabla 4 Resultados mensuales de las operaciones según la predicción del porcentaje de cambio

En la Tabla 5 se muestran los resultados **anuales** obtenidos a partir del total de operaciones analizadas, considerando específicamente las predicciones sobre los **precios de cierre**. Se presentan los resultados de cada periodo, donde la penúltima columna muestra el promedio y la última el total acumulado sin distinguir entre periodos:

FECHA INICIO	FECHA FIN	TOTAL OPERACIONES	Nº EXITOSAS	Nº PERDIDAS	% EXITOSAS (VS PERDIDAS)	TOTAL % OPERACION	% VARIACION SP
2020-01-01	2020-12-31	614	353	261	57,49	347,78	18,33
2021-01-01	2021-12-31	501	279	222	55,69	158,15	28,73
2022-01-01	2022-12-31	458	242	216	52,84	-7,38	-18,18
2023-01-01	2023-12-31	335	175	160	52,24	3,65	26,71
2024-01-01	2024-12-31	507	261	246	51,48	21,26	25,59
2020-01-01	2024-12-31	483	262	221	53,95	104,69	16,24
2020-01-01	2025-01-01	2415	1310	1105	54,24	523,46	96,4

Tabla 5 Resultados anuales de las operaciones según la predicción de los precios de cierre

En la Tabla 6 se muestran los resultados **anuales** obtenidos a partir del total de operaciones analizadas, considerando específicamente las predicciones sobre el **porcentaje de cambio**. Se

presentan los resultados de cada periodo, donde la penúltima columna muestra el promedio y la última el total acumulado sin distinguir entre periodos:

FECHA INICIO	FECHA FIN	TOTAL OPERACIONES	Nº EXITOSAS	Nº PERDIDAS	% EXITOSAS (VS PERDIDAS)	TOTAL % OPERACION	% VARIACION SP
2020-01-01	2020-12-31	82	46	36	56,1	27,07	18,33
2021-01-01	2021-12-31	8	4	4	50	0,32	28,73
2022-01-01	2022-12-31	26	10	16	38,46	-10,19	-18,18
2023-01-01	2023-12-31	12	8	4	66,67	-0,42	26,71
2024-01-01	2024-12-31	2	0	2	0	-3,5	25,59
2020-01-01	2024-12-31	26	13,6	12,4	42,25	2,66	16,24
2020-01-01	2025-01-01	130	68	62	52,31	13,28	96,4

Tabla 6 Resultados anuales de las operaciones según la predicción del porcentaje de cambio

8.3 Evaluación de rentabilidad y riesgo en datos históricos

Hasta este punto se han analizado los resultados globales de las predicciones mediante un *backtesting*, presentando sus estadísticas agregadas y evaluando el comportamiento del modelo al considerar la totalidad de las operaciones generadas. Este análisis inicial permite tener una visión general del potencial de los modelos, pero también muestra sus limitaciones, ya que en un portafolio real no es posible ejecutar todas las operaciones disponibles, pues resultaría poco práctico e irreal. En la práctica, es necesario seleccionar un conjunto específico de operaciones y, a partir de ellas, diseñar una estrategia de inversión más realista y aplicable al mercado.

Lo que diferencia este trabajo es precisamente la **creación de estrategias a partir del *backtesting* de los resultados obtenidos**. Gracias al uso de técnicas de *data science*, ha sido posible seleccionar de manera sistemática los parámetros más adecuados, lo que genera un conjunto de resultados más robustos y sobre el cual se pueden construir estrategias con mayor realismo. Con estas estrategias se analizan los rendimientos de forma diaria, mensual y anual; aunque, por simplicidad, en este estudio se presentan únicamente los resultados mensuales y anuales.

Conviene destacar que un *backtesting* global con resultados modestos no implica necesariamente que las estrategias derivadas sean poco efectivas, ya que al no poder ejecutarse todas las operaciones y tener cada una un peso diferente, la selección final puede conducir a un desempeño mejor en el mercado real. De la misma forma, también puede ocurrir lo contrario: que un *backtesting* muy favorable no se traduzca en resultados igualmente buenos al convertirlo en una estrategia concreta.

Estas estrategias permiten **simular condiciones cercanas a la realidad del mercado** y conocer los resultados que se habrían obtenido en los periodos considerados. Para ello, se calcula la rentabilidad y el riesgo de cada estrategia, teniendo en cuenta factores que afectan directamente al rendimiento neto, como las comisiones de intermediación, el *slippage* y los impuestos (aplicando los tramos impositivos vigentes en España sobre las ganancias obtenidas). El *slippage* hace referencia a la diferencia entre el precio esperado de ejecución de una operación y el precio real al que finalmente se ejecuta, fenómeno habitual en mercados volátiles o de menor liquidez que puede reducir de forma

significativa el beneficio de una operación. Además, se incorpora la consideración de un capital inicial de inversión, junto con los intereses que aplicaría el bróker en caso de emplear apalancamiento financiero.

El riesgo se evalúa mediante métricas ampliamente utilizadas en *backtesting* y gestión de portafolios, como el **máximo drawdown**, que mide la mayor caída acumulada desde un máximo histórico hasta un mínimo posterior, reflejando la peor pérdida experimentada por la estrategia. También se utiliza el **maximum time to recovery** de ese máximo drawdown, que indica el tiempo necesario para que una estrategia recupere su valor tras una pérdida significativa, proporcionando así una visión temporal del riesgo. Ambas métricas permiten comprender no solo cuánto puede perder una estrategia, sino también cuánto tarda en recuperarse, lo que resulta fundamental para evaluar la viabilidad de una inversión a largo plazo.

Finalmente, todas estas métricas y resultados se contrastan con el comportamiento del índice S&P 500, utilizado como referencia (*benchmark*) del mercado. Además de los valores numéricos, se representará gráficamente la evolución de cada estrategia frente al índice, mostrando la rentabilidad acumulada, la volatilidad y las caídas sufridas. De este modo, el análisis trasciende la validación estadística y se acerca a la realidad operativa de los mercados, comprobando la rentabilidad y consistencia de las estrategias en distintos escenarios y reforzando su aplicabilidad práctica.

Como recordatorio, el análisis se basa en operaciones alcistas con arquitectura LSTM, usando `num_lookback = 30` y `num_forecast = 10`, con diferencias en `batch_size` según la variable objetivo (32 para precio de cierre y 40 para porcentaje de cambio), manteniendo 20 epochs y un `learning_rate` de 0.05. Para los experimentos se emplean los ficheros de predicción del 5 % en precios de cierre y del 1 % en porcentaje de cambio, al presentar un volumen de operaciones y un nivel de acierto adecuados, quedando la validación final supeditada al backtesting posterior.

El backtesting se realizará considerando un *slippage* del 0.1 % en cada operación y unas comisiones del 0.01 %. Asimismo, se tendrán en cuenta los tramos impositivos vigentes en España, mientras que no se considerará el *spread*, al tratarse de activos de muy alta capitalización y elevada liquidez. La simulación se llevará a cabo partiendo de un capital inicial de 100.000 dólares y teniendo como máximo 2 operaciones en cartera, dividiendo el capital por operación en partes iguales.

En la Tabla 7 se muestran los resultados **mensuales** obtenidos a partir del de la estrategia real, considerando específicamente las predicciones sobre los **precios de cierre**. Se presentan los resultados de cada periodo, donde la penúltima columna muestra el promedio y la última el total acumulado sin distinguir entre periodos:

FECHA INICIO	FECHA FIN	BENEFICIO ANTES DE IMPUESTOS	% ANTES DE IMPUESTOS	MÁX DRAWDOWN	% VARIACION SP	MÁX DRAWDOWN SPY
2020-01-01	2020-01-31	-1709,82	-1,71	-5,52	-0,04	-3,09
2020-02-01	2020-02-29	-9578,59	-9,58	-12,41	-7,92	-15,79
2020-03-01	2020-03-31	6668,28	6,67	-10,99	-16,12	-30,05
2020-04-01	2020-04-30	37764,12	37,76	-0,97	12,7	-7,05
2020-05-01	2020-05-31	9722,2	9,72	-1,6	4,76	-7,22
2020-06-01	2020-06-30	7443,1	7,44	-0,2	1,36	-8,25
2020-07-01	2020-07-31	2592,56	2,59	0	5,89	-3,32
2020-08-01	2020-08-31	-1486,97	-1,49	-1,34	6,98	-1,64
2020-09-01	2020-09-30	4117,07	4,12	0	-3,74	-10,5
2020-10-01	2020-10-31	-648,38	-0,65	-2,52	-2,49	-8,88
2020-11-01	2020-11-30	2934,15	2,93	-0,64	9,65	-3,81
2020-12-01	2020-12-31	1180	1,18	0	3,7	-4,34
2021-01-01	2021-01-31	-551,14	-0,55	0	-1,02	-4,56

PREDICCIÓN DE SERIES TEMPORALES UTILIZANDO INTELIGENCIA ARTIFICIAL:
DESDE MODELOS CLÁSICOS HASTA REDES NEURONALES RECURRENTE

2021-02-01	2021-02-28	9,69	0,01	-1,23	1,1	-4,04
2021-03-01	2021-03-31	877,83	0,88	0	2,07	-4,87
2021-04-01	2021-04-30	640,9	0,64	-2,36	5,29	-1,75
2021-05-01	2021-05-31	-900,05	-0,9	-3,75	0,66	-4,45
2021-06-01	2021-06-30	-5018,11	-5,02	-3,75	2,33	-2,21
2021-07-01	2021-07-31	18,96	0,02	-1,06	2,44	-3,64
2021-08-01	2021-08-31	4238,23	4,24	-0,17	3,19	-2,46
2021-09-01	2021-09-30	1769,78	1,77	-1,73	-4,66	-5,25
2021-10-01	2021-10-31	11013,79	11,01	-0,12	7,02	-2,3
2021-12-01	2021-12-31	2897,7	2,9	-2,13	4,62	-4,26
2022-01-01	2022-01-31	971,19	0,97	-1,34	-5,27	-12,34
2022-02-01	2022-02-28	59,96	0,06	0	-2,95	-10,36
2022-03-01	2022-03-31	-470	-0,47	0	3,76	-5,89
2022-04-01	2022-04-30	-1910	-1,91	0	-8,78	-10,18
2022-05-01	2022-05-31	6079,65	6,08	0	-0,37	-11,43
2022-06-01	2022-06-30	-3829,15	-3,83	-9,22	-8,25	-12,87
2022-07-01	2022-07-31	208,93	0,21	-4,17	9,21	-5,02
2022-08-01	2022-08-31	-733,82	-0,73	-0,98	-3,8	-8,28
2022-09-01	2022-09-30	-5625,28	-5,63	-4,97	-9,24	-12,28
2022-10-01	2022-10-31	7116,5	7,12	-5,86	8,13	-8,26
2022-11-01	2022-11-30	9085,47	9,09	-2,23	5,56	-5,53
2022-12-01	2022-12-31	-8923,5	-8,92	-8,25	-5,76	-8,28
2023-01-01	2023-01-31	12969,82	12,97	-0,89	6,74	-3,24
2023-02-01	2023-02-28	2019,92	2,02	-1,88	-2,51	-5,9
2023-03-01	2023-03-31	6016,41	6,02	-2,83	3,71	-6,58
2023-04-01	2023-04-30	-3194,55	-3,19	-3,18	1,6	-2,87
2023-05-01	2023-05-31	-2868,13	-2,87	-3,51	0,56	-3,32
2023-06-01	2023-06-30	2381,87	2,38	0	6,48	-2,8
2023-07-01	2023-07-31	1649,72	1,65	0	3,27	-1,72
2023-08-01	2023-08-31	-1293,11	-1,29	-2,18	-1,63	-5,3
2023-09-01	2023-09-30	-1449,5	-1,45	0	-4,74	-6,59
2023-10-01	2023-10-31	-1192,93	-1,19	-2,65	-2,13	-6,6
2023-11-01	2023-11-30	1627,07	1,63	0	9,13	-1,61
2023-12-01	2023-12-31	896,84	0,9	-1,13	4,57	-1,7
2024-01-01	2024-01-31	-2676,5	-2,68	-1,99	2,16	-1,82
2024-02-01	2024-02-29	-148,21	-0,15	-2,85	5,22	-2,54
2024-03-01	2024-03-31	2050,91	2,05	-0,13	3,27	-1,88
2024-04-01	2024-04-30	6043,09	6,04	-2,45	-3,86	-5,82
2024-05-01	2024-05-31	1520,69	1,52	-2,59	5,06	-2,2
2024-06-01	2024-06-30	6777,8	6,78	-1,8	3,53	-1,33
2024-07-01	2024-07-31	3680,54	3,68	0	1	-4,9
2024-09-01	2024-09-30	-997,82	-1	-0,32	4,25	-3,81
2024-10-01	2024-10-31	-3516,58	-3,52	-3,6	-0,89	-2

2024-11-01	2024-11-30	-600,89	-0,6	-0,5	5,96	-2,72
2024-12-01	2024-12-31	-3481,8	-3,48	-4,96	-2,58	-4,3
2020-01-01	2024-12-31	1762,76	1,76	-2,22	1,18	-5,79
2020-01-02	2024-12-30	151539,64	151,54	-13,19	95,3	-35,25

Tabla 7 Resultados mensuales de la estrategia real según la predicción del precio de cierre

En la Tabla 8 se muestran los resultados **mensuales** obtenidos a partir del de la estrategia real, considerando específicamente las predicciones sobre los **porcentajes de cambio**. Se presentan los resultados de cada periodo, donde la penúltima columna muestra el promedio y la última el total acumulado sin distinguir entre periodos:

FECHA INICIO	FECHA FIN	BENEFICIO ANTES DE IMPUESTOS	% ANTES DE IMPUESTOS	MÁX DRAWDOWN	% VARIACION SP	MÁX DRAWDOWN SPY
2020-01-01	2020-01-31	1139,54	1,14	-1,28	-0,04	-3,09
2020-02-01	2020-02-29	-1976,51	-1,98	-2,38	-7,92	-15,79
2020-03-01	2020-03-31	5889,14	5,89	0	-16,12	-30,05
2020-04-01	2020-04-30	3911,34	3,91	-1,45	12,7	-7,05
2020-05-01	2020-05-31	4598,73	4,6	-1,36	4,76	-7,22
2020-06-01	2020-06-30	363,64	0,36	-2,4	1,36	-8,25
2020-07-01	2020-07-31	-100	-0,1	0	5,89	-3,32
2020-08-01	2020-08-31	1008,31	1,01	0	6,98	-1,64
2020-09-01	2020-09-30	-2240,59	-2,24	-0,83	-3,74	-10,5
2020-11-01	2020-11-30	-1074,07	-1,07	-0,1	9,65	-3,81
2020-12-01	2020-12-31	1361,92	1,36	0	3,7	-4,34
2021-01-01	2021-01-31	78,94	0,08	0	-1,02	-4,56
2021-02-01	2021-02-28	65	0,07	0	1,1	-4,04
2021-03-01	2021-03-31	-703,76	-0,7	-0,32	2,07	-4,87
2021-05-01	2021-05-31	-265	-0,27	0	0,66	-4,45
2021-07-01	2021-07-31	980	0,98	0	2,44	-3,64
2022-02-01	2022-02-28	1005	1	0	-2,95	-10,36
2022-04-01	2022-04-30	-1000	-1	0	-8,78	-10,18
2022-05-01	2022-05-31	505,56	0,51	0	-0,37	-11,43
2022-06-01	2022-06-30	-310,73	-0,31	-0,6	-8,25	-12,87
2022-07-01	2022-07-31	-1205	-1,21	0	9,21	-5,02
2022-08-01	2022-08-31	-3141,85	-3,14	-2,41	-3,8	-8,28
2022-09-01	2022-09-30	351,54	0,35	-1,82	-9,24	-12,28
2022-10-01	2022-10-31	-159,65	-0,16	-1,84	8,13	-8,26
2022-11-01	2022-11-30	735	0,73	0	5,56	-5,53
2022-12-01	2022-12-31	-1876,73	-1,88	-0,7	-5,76	-8,28
2023-01-01	2023-01-31	1072,91	1,07	0	6,74	-3,24
2023-02-01	2023-02-28	-2110,49	-2,11	-1,33	-2,51	-5,9
2023-03-01	2023-03-31	420	0,42	0	3,71	-6,58
2023-04-01	2023-04-30	681,14	0,68	0	1,6	-2,87
2023-07-01	2023-07-31	-292,81	-0,29	0	3,27	-1,72
2024-03-01	2024-03-31	-905	-0,91	0	3,27	-1,88
2024-11-01	2024-11-30	-845	-0,84	0	5,96	-2,72

2020-01-01	2024-11-30	180,62	0,18	-0,57	0,86	-7,09
2020-01-02	2024-12-30	5554,66	5,55	-10,83	95,3	-35,25

Tabla 8 Resultados mensuales de la estrategia real según la predicción del porcentaje de cambio

En la Tabla 9 se muestran los resultados **anuales** obtenidos a partir del de la estrategia real, considerando específicamente las predicciones sobre los **precios de cierre**. Se presentan los resultados de cada periodo, donde la penúltima columna muestra el promedio y la última el total acumulado sin distinguir entre periodos:

FECHA INICIO	FECHA FIN	BENEFICIO DESPUÉS DE IMPUESTOS	% DESPUÉS DE IMPUESTOS	MÁX DRAWDOWN	% VARIACION SP	MÁX DRAWDOWN SPY
2020-01-01	2020-12-31	57614,43	57,61	-12,23	18,33	-35,25
2021-01-01	2021-12-31	14328,39	14,33	-8,3	28,73	-5,8
2022-01-01	2022-12-31	2795,04	2,8	-10,58	-18,18	-26,63
2023-01-01	2023-12-31	17077,94	17,08	-7,55	26,71	-10,62
2024-01-01	2024-12-31	9880,51	9,88	-7,53	25,59	-9,71
2020-01-01	2024-12-31	20339,26	20,34	-9,24	16,24	-17,6
2020-01-02	2025-01-01	137319,33	137,32	-13,35	94,58	-35,25

Tabla 9 Resultados anuales de la estrategia real según la predicción del precio de cierre

En la Tabla 10 se muestran los resultados **anuales** obtenidos a partir del de la estrategia real, considerando específicamente las predicciones sobre los **porcentajes de cambio**. Se presentan los resultados de cada periodo, donde la penúltima columna muestra el promedio y la última el total acumulado sin distinguir entre periodos:

FECHA INICIO	FECHA FIN	BENEFICIO DESPUÉS DE IMPUESTOS	% DESPUÉS DE IMPUESTOS	MÁX DRAWDOWN	% VARIACION SP	MÁX DRAWDOWN SPY
2020-01-01	2020-12-31	10607,34	10,61	-5,31	18,33	-35,25
2021-01-01	2021-12-31	119,57	0,12	-0,97	28,73	-5,8
2022-01-01	2022-12-31	-5055,41	-5,06	-6,4	-18,18	-26,63
2023-01-01	2023-12-31	-260,84	-0,26	-2,11	26,71	-10,62
2024-01-01	2024-12-31	-1742,35	-1,74	-0,85	25,59	-9,71
2020-01-01	2024-12-31	733,66	0,73	-3,13	16,24	-17,6
2020-01-02	2025-01-01	3039,85	3,04	-10,83	94,58	-35,25

Tabla 10 Resultados anuales de la estrategia real según la predicción del porcentaje de cambio

9 CONCLUSIONES Y LÍNEAS FUTURAS DE INVESTIGACIÓN

9.1 Principales conclusiones del trabajo

Los resultados del backtesting muestran que el comportamiento del modelo no se traduce de forma directa en el mismo porcentaje de acierto cuando se aplica dentro de una estrategia operativa. Aunque a nivel predictivo el modelo es capaz de identificar determinadas tendencias, en la práctica estas pueden prolongarse en la dirección equivocada durante varias operaciones consecutivas. Esto provoca rachas de fallos que no aparecen reflejadas de forma tan clara en las métricas puramente estadísticas, y explica por qué el rendimiento real de la estrategia difiere del observado en la fase de validación del modelo.

Al analizar el comportamiento por tipo de mercado, se observa que el modelo se desempeña mejor en escenarios alcistas que en bajistas. Esta asimetría es coherente con el propio diseño del enfoque y con la naturaleza de los datos, ya que las tendencias alcistas suelen ser más persistentes y estables, mientras que las bajistas tienden a ser más abruptas e impredecibles. Además, tanto en el conjunto total del backtesting como al aplicar las predicciones a estrategias concretas, se aprecia una correlación relevante con el S&P 500, lo que indica que parte del rendimiento está ligado al comportamiento general del mercado y no únicamente a la capacidad predictiva del modelo.

En cuanto a la variable objetivo, los resultados muestran diferencias claras. Los modelos basados en porcentaje de cambio han ofrecido resultados pobres y no se comportan como buenos predictores con los parámetros estudiados. Por el contrario, la predicción de precios de cierre sí presenta un comportamiento más consistente y podría utilizarse como base para la generación de señales, especialmente en combinación con otros filtros o activos.

Utilizando precios de cierre, el modelo logra superar al S&P 500 en términos de rentabilidad neta tras impuestos durante el periodo analizado, que se adopta como índice de referencia. Además, presenta un máximo drawdown total y medio inferior, lo que implica una menor caída máxima del capital en los momentos más adversos del mercado. Esta combinación de mayor rentabilidad y mejor control del riesgo resulta especialmente relevante: no solo mejora el resultado agregado frente al índice, sino que también reduce la exposición a pérdidas severas. En la práctica, esto sugiere que el enfoque es sólido y escalable, y que su aplicación a carteras con mayor volatilidad o a un universo más amplio de activos podría reforzar aún más la ventaja observada frente al S&P 500.

Además, es importante señalar que los datos utilizados se ajustan únicamente por splits, tal y como se indicó anteriormente, y no por dividendos. Esta decisión implica una ligera pérdida de rentabilidad en las pruebas, que en algunos activos podría suponer en torno a un 2 % adicional real de beneficio si se incorporaran los dividendos, o una cantidad variable en función de los dividendos de cada activo.

9.2 Implicaciones prácticas para estrategias de inversión

Este trabajo es aplicable a un entorno de mercado real, ya que el planteamiento permite realizar la predicción de los activos justo un minuto antes del cierre del mercado y ejecutar las compras necesarias en función del número máximo de operaciones permitidas en cartera. De este modo, el proceso de decisión y ejecución puede integrarse de forma directa en un flujo operativo diario, alineado con las restricciones habituales de gestión de cartera.

También sería posible aplicar el modelo entrando en la apertura del día siguiente, utilizando las predicciones generadas al cierre. Esta alternativa no se ha desarrollado en detalle en este trabajo para no ampliar excesivamente su alcance, aunque el código se ha diseñado de forma flexible para soportar ambos enfoques sin cambios estructurales relevantes.

Asimismo, la implementación contempla la posibilidad de gestionar las salidas de las posiciones cerrándolas cuando el precio alcanza el umbral máximo definido por el porcentaje correspondiente en cada fichero. Esto permite adaptar la estrategia tanto a entradas en apertura como a ejecuciones cercanas al cierre, manteniendo la coherencia del modelo y facilitando su uso en escenarios reales de trading automatizado.

9.3 Limitaciones y mejoras futuras en modelos y datos

Como en todo estudio aplicado a los mercados financieros, el trabajo desarrollado presenta ciertas limitaciones que conviene señalar, así como posibles líneas de investigación futuras que podrían ampliar o complementar los resultados obtenidos.

Una de las principales consideraciones está relacionada con el criterio de entrada y salida de las operaciones. En este estudio, se ha definido que la entrada se produce al cierre del día en el que se realiza la predicción, mientras que la salida se establece en el cierre de la sesión en la que se cumple la condición de porcentaje fijada (o en el último día si nunca se alcanza dicho umbral). Este planteamiento es razonablemente realista, ya que es posible generar la predicción instantes antes del cierre del mercado y ejecutar la orden en ese mismo momento. No obstante, este no constituye el único enfoque posible. En futuras investigaciones podría evaluarse la efectividad de alternativas más cercanas a la operativa de ciertos inversores, como establecer la entrada en la apertura de la sesión siguiente al día de la predicción, o permitir que la salida se produzca en el mismo instante en que el umbral se alcanza intradía, sin esperar al cierre. Estos enfoques no invalidan el criterio adoptado en este trabajo, pero permitirían contrastar variantes y enriquecer la visión sobre el comportamiento de las estrategias. Por otra parte, se podrán hacer pruebas para predicciones de más años y con más combinaciones de variables, así como para una mayor variedad de activos.

Otra limitación importante se encuentra en la consideración del *slippage*, entendido como la diferencia entre el precio esperado de ejecución y el precio real al que finalmente se lleva a cabo la operación. En este estudio, se ha aplicado un valor fijo, cuya cifra utilizada se ha calculado a partir de estadísticas de miles de operaciones, lo que la hace bastante representativa. Sin embargo, en la práctica el *slippage* puede variar de forma significativa en función del activo, el volumen de capital invertido o las condiciones del mercado. Algo similar ocurre con las comisiones, que también pueden fluctuar según el intermediario financiero o el nivel de operativa del inversor. Una mejora natural sería incorporar valores dinámicos de *slippage* y comisiones, ajustados a cada contexto, lo que aumentaría el realismo de los resultados.

Más allá de estas limitaciones iniciales, el trabajo abre la puerta a diversas líneas de mejora orientadas a aproximar aún más las estrategias a la operativa real de los mercados. Una de ellas es la inclusión de mecanismos de gestión dinámica del riesgo, como los *trailing stops*. Este tipo de orden actúa como un *stop loss* móvil, ajustándose de manera automática conforme el precio evoluciona

favorablemente. De este modo, se asegura una parte de las ganancias y, al mismo tiempo, se limitan las pérdidas potenciales. Incorporar esta práctica permitiría acercar el modelo a estrategias de corte profesional y mejorar la relación rentabilidad/riesgo (aunque automatizarlo puede llevar a bastantes recursos computacionales si se tienen muchas operaciones abiertas en el portafolio, además de una mayor repercusión del *slippage*).

Otra línea de mejora se centra en la gestión del capital invertido. En este estudio se optó por una división equitativa del capital entre todas las operaciones, lo cual facilita el análisis comparativo, pero no refleja el comportamiento de una cartera optimizada. En futuras investigaciones sería interesante aplicar técnicas de asignación de capital más sofisticadas, ajustando la inversión en función de la volatilidad del activo, su correlación con el resto de la cartera o la confianza en la señal de predicción. Esto permitiría construir portafolios más diversificados y adaptados a diferentes perfiles de riesgo.

Asimismo, la inclusión de condiciones de mercado constituye otra vía de mejora. Actualmente, el modelo genera señales y ejecuta operaciones de forma sistemática, sin discriminar entre entornos favorables o adversos. Introducir filtros basados en tendencias claras, niveles de volatilidad o fases de mercado específicas (alcista, bajista o lateral) permitiría optimizar las entradas y salidas, reduciendo la exposición en escenarios menos atractivos y aumentando la eficiencia general de las estrategias.

Otra línea de investigación futura podría enfocarse en la integración de un mayor número de indicadores técnicos, lo que permitiría una evaluación más robusta y precisa de los mercados financieros. Además, se podría explorar el análisis de datos no estructurados provenientes de noticias y redes sociales, utilizando técnicas de procesamiento de lenguaje natural y aprendizaje automático, para predecir movimientos del mercado a partir de tendencias sentimentales y eventos relevantes en tiempo real.

En cuanto a las líneas metodológicas, también existen oportunidades relevantes de investigación. Una de ellas es explorar arquitecturas más complejas de aprendizaje profundo. Aunque en este trabajo se han utilizado redes LSTM por su idoneidad para el tratamiento de series temporales, en futuras investigaciones podría explorarse el uso de otras arquitecturas como redes neuronales convolucionales (CNN) o de arquitecturas más complejas, como los *Transformers*. Este tipo de modelos se caracterizan por su capacidad para manejar secuencias largas y patrones más sofisticados, aunque requieren un mayor preprocesamiento de datos y un ajuste más delicado de los hiperparámetros e indicadores. Su incorporación no garantiza necesariamente mejores resultados, pero sí representa una vía prometedora para profundizar en el análisis de predicciones financieras con enfoques de aprendizaje profundo más avanzados.

Finalmente, una línea prometedora es el uso de modelos de heterocedasticidad, como ARCH y GARCH, que permiten analizar y modelar la volatilidad dinámica de los mercados financieros. Integrar estas herramientas dentro de la estrategia facilitaría ajustar de forma adaptativa la exposición al riesgo, invirtiendo menos capital en fases de alta volatilidad y aumentando la inversión en periodos de mayor estabilidad. Este ajuste dinámico permitiría que las estrategias se aproximen más a la lógica de gestión de riesgo aplicada en entornos financieros reales.

En conjunto, la consideración de estas limitaciones y la exploración de las líneas mencionadas proporcionan un marco claro para enriquecer el trabajo aquí presentado. Incorporar *trailing stops*, aplicar técnicas de gestión de capital más flexibles, introducir filtros de mercado, explorar arquitecturas neuronales avanzadas y modelizar la volatilidad con enfoques especializados son pasos que pueden dotar a las estrategias de mayor robustez, realismo y aplicabilidad práctica.

Todo este trabajo puede extrapolarse a otros tipos de series temporales, como el mantenimiento predictivo, donde las técnicas de gestión de riesgos, la modelización de volatilidad y las arquitecturas neuronales pueden ayudar a anticipar fallos y optimizar intervenciones. Esta línea de investigación abre rutas futuras para adaptar las estrategias desarrolladas a dominios donde la predicción y la robustez son críticas, ampliando su aplicabilidad y valor práctico.

10 BIBLIOGRAFÍA

- [1] C. P.-I. J. B. Nataliia Kashpruk, «Time Series Prediction in Industry 4.0: A Comprehensive Review and Prospects for Future Advancements,» *Applied Sciences*, vol. 13, nº 22, p. 12374, 2023.
- [2] L. Cao, «AI in Finance: Challenges, Techniques, and Opportunities,» *ACM Computing Surveys (CSUR)*, vol. 55, nº 3, pp. 1-38, 2022.
- [3] M. B. M. T. A. S. S. S. S. A. H.-C. K. Nusrat Roul, «Stock Market Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions,» *Electronics*, vol. 10, nº 21, p. 2717, 2021.
- [4] N. Kotecha, «Artificial Intelligence in the Stock Market: The Trends and Challenges Regarding AI-Driven Investments,» *Open Journal of Business and Management*, vol. 13, nº 2, 2025.
- [5] J. M. Fidalgo, «Diseño, Desarrollo y Evaluación de Modelos de Predicción de Series Temporales Aplicados al Monitoreo Geotécnico y Estructural,» Universitat Oberta de Catalunya, Barcelona, 2024.
- [6] J. A. Mauricio, *Introducción al Análisis de Series Temporales*, Madrid: Universidad Complutense de Madrid, 2013.
- [7] C. L. J. M. K. Douglas C. Montgomery, *Introduction to Time Series Analysis and Forecasting*, Segunda ed., Hoboken, NJ: Wiley, 2015.
- [8] S. Amrane, «Medium,» Medium Corporation, 9 2 2025. [En línea]. Available: <https://sirineamrane.medium.com/what-is-white-noise-in-stationary-time-series-part-1-introduction-193df0af7d77>. [Último acceso: 11 8 2025].
- [9] QuantStart, «QuantStart,» QuantStart, [En línea]. Available: <https://www.quantstart.com/articles/White-Noise-and-Random-Walks-in-Time-Series-Analysis/>. [Último acceso: 11 9 2025].
- [10] S. Krishnan, «Medium,» Medium, 18 10 2021. [En línea]. Available: <https://medium.com/codex/what-is-stationarity-in-time-series-how-it-can-be-detected-7e5dfa7b5f6b>. [Último acceso: 11 08 2025].
- [11] L. H. B. Filho, «Análise Macro,» Análise Macro, 07 07 2023. [En línea]. Available: <https://analisemacro.com.br/estatistica-e-econometria/estacionariedade-de-series-temporais/>.

[Último acceso: 11 08 2025].

- [12] RPubS by RStudio, «RPubS,» RStudio, [En línea]. Available: https://rpubs.com/elias_jurgen/492089. [Último acceso: 11 08 2025].
- [13] I. Silva, «Alura LATAM,» Alura LATAM, 05 04 2023. [En línea]. Available: <https://www.aluracursos.com/blog/tipos-de-estacionalidad>. [Último acceso: 11 08 2025].
- [14] Universidad de Sevilla, «Autocorrelación en series temporales,» Universidad de Sevilla, 06 09 2011. [En línea]. Available: <http://innoevalua.us.es/files/perpage/@disenoslongitudinalesdeinvestigacionaplicados@autocorrelacion.pdf>. [Último acceso: 11 08 2025].
- [15] PTC, «PEjemplo: correlación y autocorrelación parcial,» PTC, [En línea]. Available: https://support.ptc.com/help/mathcad/r8.0/es/index.html#page/PTC_Mathcad_Help/example_correlation_and_partial_autocorrelation.html. [Último acceso: 11 08 2025].
- [16] FasterCapital, «FasterCapital,» FasterCapital, 26 junio 2024. [En línea]. Available: <https://fastercapital.com/es/tema/desaf%C3%ADos-y-limitaciones-del-an%C3%A1lisis-de-series-temporales-en-la-previsi%C3%B3n.html>. [Último acceso: 26 junio 2024].
- [17] TigerData, «Understanding Autoregressive Time-Series Modeling,» TigerData, 21 10 2024. [En línea]. Available: <https://www.tigerdata.com/learn/understanding-autoregressive-time-series-modeling>. [Último acceso: 11 08 2025].
- [18] ACCA, «Time series and moving averages,» Association of Chartered Certified Accountants, [En línea]. Available: <https://www.accaglobal.com/gb/en/student/exam-support-resources/fundamentals-exams-study-resources/f5/technical-articles/time-series.html>. [Último acceso: 11 08 2025].
- [19] B. Borah, «Medium,» 29 06 2024. [En línea]. Available: <https://medium.com/@bhargavborah020/understanding-moving-average-ma-models-in-time-series-analysis-d18e443f15cc>. [Último acceso: 11 08 2025].
- [20] K. Mishra, «Medium,» A Medium Corporation, 24 05 2022. [En línea]. Available: <https://iamkartikya.medium.com/time-series-modeling-13f076d6d6ca>. [Último acceso: 12 08 2025].
- [21] IBM, «IBM Think,» IBM, 24 05 2024. [En línea]. Available: <https://www.ibm.com/es-es/think/topics/arima-model>. [Último acceso: 12 08 2025].
- [22] Ç. Çakıroğlu, «Medium,» Medium, 30 03 2024. [En línea]. Available: https://medium.com/@cagri_cakiroglu/arima-arma-moving-average-and-autoregressive-6ed2efef4aa9. [Último acceso: 12 08 2025].
- [23] GeeksforGeeks, «GeeksforGeeks,» GeeksforGeeks, 23 07 2025. [En línea]. Available: <https://www.geeksforgeeks.org/artificial-intelligence/what-is-an-arimax-model/>. [Último acceso: 12 08 2025].
- [24] DataScientest, «DataScientest,» DataScientest, 18 03 2024. [En línea]. Available: <https://datascientest.com/en/sarimax-model-what-is-it-how-can-it-be-applied-to-time-series>. [Último acceso: 12 08 2025].
- [25] R. Verma, V. Nagar y S. Mahapatra, «Introduction to supervised learning,» *Wiley Online*

Library, 2021.

- [26] T. Hastie, R. Tibshirani y J. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*, New York, NY: Springer New York, 2008.
- [27] NVIDIA, «NVIDIA Glossary,» NVIDIA Corporation, 2025. [En línea]. Available: <https://www.nvidia.com/en-us/glossary/xgboost/>. [Último acceso: 13 08 2025].
- [28] Team Asana, «Asana,» Asana, Inc., 21 02 2025. [En línea]. Available: <https://asana.com/es/resources/decision-tree-analysis>. [Último acceso: 13 08 2025].
- [29] P. Sadorsky, «Random Forests Approach to Predicting Clean Energy Stock Prices,» *Journal of Risk and Financial Management*, vol. 14, n° 2, p. 48, 2021.
- [30] Viswa, «Medium,» A Medium Corporation, 15 08 2023. [En línea]. Available: <https://medium.com/@vk.viswa/surface-level-understanding-of-random-forest-regression-3bac37e06f3e>. [Último acceso: 13 08 2025].
- [31] A. Fernández, «anderfernandez.com,» [En línea]. Available: <https://anderfernandez.com/blog/guia-completa-random-forest/>. [Último acceso: 13 08 2025].
- [32] T. Masui, «Medium,» A Medium Corporation, 20 01 2022. [En línea]. Available: <https://medium.com/data-science/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502>. [Último acceso: 13 08 2025].
- [33] A. Minaro, «Theodo Data & AI,» Theodo, 27 09 2023. [En línea]. Available: <https://data-ai.theodo.com/en/technical-blog/mastering-lightgbm-unravelling-the-magic-behind-gradient-boosting>. [Último acceso: 13 08 2025].
- [34] Data Overload, «Medium,» A Medium Corporation, 06 12 2023. [En línea]. Available: <https://medium.com/@data-overload/comparing-xgboost-and-lightgbm-a-comprehensive-analysis-9b80b7b0079b>. [Último acceso: 13 08 2025].
- [35] M. A. Nielsen, *Neural networks and deep learning*, San Francisco, CA, USA: Determination Press, 2015.
- [36] J. Koutnik, K. Greff, F. Gomez y J. Schmidhuber, «A clockwork rnn,» *International conference on machine learning*, pp. 1863-1871, 2014.
- [37] S. Poudel, «Recurrent Neural Network (RNN) Architecture Explained,» *Medium*, 2023.
- [38] DataScientest, «DataScientest,» DataScientest, 20 05 2024. [En línea]. Available: <https://datascientest.com/es/memoria-a-largo-plazo-a-corto-plazo-lstm>. [Último acceso: 15 08 2025].
- [39] MathWorks, «MathWorks,» MathWorks, 2025. [En línea]. Available: <https://la.mathworks.com/discovery/lstm.html>. [Último acceso: 14 08 2025].
- [40] R. Rana, «Gated recurrent unit (GRU) for emotion classification from noisy speech,» *arXiv*, 2016.
- [41] G. Shen, Q. Tan, H. Zhang, P. Zeng y J. Xu, «Deep learning with gated recurrent unit networks for financial sequence predictions,» *Procedia Computer Science*, vol. 131, pp. 895-903, 2018.
- [42] L. A. G. López, «Predicción de series de tiempo usando redes neuronales recurrentes LSTM y GRU,» Universidad Nacional Autónoma de México, Ciudad de México, 2021.

- [43] F. K. H. O. K. H. E. C. Humaira Kamal, «Performance Anomaly Detection in Web Services: An RNN-Based Approach Using Dynamic Quality of Service Features,» *IEEE Transactions on Services Computing*, vol. 14, n° 3, p. 1–14, 2021.
- [44] P. Mondal, L. Shit y S. Goswami, «Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices,» *International Journal of Computer Science, Engineering and Applications*, vol. 4, n° 2, p. 13, 2014.
- [45] J. Brownlee, *Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python*, Machine Learning Mastery, 2020.
- [46] Hantec Markets, «Hantec Markets,» Hantec Markets, 2024. [En línea]. Available: <https://hmarkets.com/es/learn-to-trade/learning-hub/trading-journal/>. [Último acceso: 18 08 2025].
- [47] Enbolsa, [En línea]. Available: <https://www.enbolsa.net/analisis-tecnico/>. [Último acceso: 18 08 2025].
- [48] F. B. Oriani y G. P. Coelho, «Evaluating the impact of technical indicators on stock forecasting,» *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1-8, 2016.
- [49] Club de Capitales, «Club de Capitales,» Club de Capitales, 2025. [En línea]. Available: <https://clubdecapitales.com/indicadores-de-analisis-tecnico-en-elliott-wave/>. [Último acceso: 18 08 2025].
- [50] X. Chu, I. F. Ilyas, S. Krishnan y J. Wang, «Data cleaning: Overview and emerging challenges,» *Proceedings of the 2016 International Conference on Management of Data*, p. 2201–2206, 2016.
- [51] K. K. Rangi, «Medium,» A Medium Corporation, 25 08 2024. [En línea]. Available: <https://medium.com/@iamkamleshurangi/how-min-max-scaler-works-9fbb9347da>. [Último acceso: 22 08 2025].
- [52] Keras, «Keras,» Keras Team, [En línea]. Available: https://keras.io/api/models/model_saving_apis/model_saving_and_loading/#save-method. [Último acceso: 15 09 2025].
- [53] Perforce, «Perforce Blog,» Perforce Software, 05 08 2020. [En línea]. Available: <https://www.perforce.com/blog/ims/auto-arima>. [Último acceso: 19 08 2025].
- [54] SAP, «SAP Help Portal,» SAP SE, [En línea]. Available: https://help.sap.com/docs/SAP_INTEGRATED_BUSINESS_PLANNING/feae3cea3cc549aa9d9de7d363a83e6/c9b4470ccc0b47b7bed31d18c76f2201.html. [Último acceso: 19 08 2025].
- [55] J. Frost, «Scatterplots: Using, Examples, and Interpreting,» Statistics by Jim, [En línea]. Available: <https://statisticsbyjim.com/graphs/scatterplots/>. [Último acceso: 18 08 2025].
- [56] DATAtab, «DATAtab,» DATAtab, [En línea]. Available: <https://datatab.net/tutorial/box-plot>. [Último acceso: 18 08 2025].
- [57] Towards Data Science, «Towards Data Science,» Medium, 13 11 2024. [En línea]. Available: <https://towardsdatascience.com/demystifying-the-correlation-matrix-in-data-science-6b8a4482b6e2/>. [Último acceso: 19 08 2025].

- [58] P. Christoffersen, «Backtesting,» *SSRN Electronic Journal*, 2008.
- [59] K. Olorunnimbe y H. Viktor, «Deep learning in the stock market—a systematic survey of practice, backtesting, and applications,» *Artificial Intelligence Review*, vol. 56, nº 3, pp. 2057-2109, 2023.
- [60] D. H. Bailey, J. Borwein, M. Lopez de Prado y Q. J. Zhu, «The probability of backtest overfitting,» *Journal of Computational Finance (Risk Journals)*, 2015.
- [61] A. Suhonen, M. Lennkh y F. Perez, «Quantifying backtest overfitting in alternative beta strategies,» *SSRN*, 2017.
- [62] Skilling Writers Team, «Skilling,» Skilling Ltd., 03 06 2024. [En línea]. Available: <https://skilling.com/eu/en/blog/trading-terms/backtesting/>. [Último acceso: 19 08 2025].

ANEXO I: ABREVIATURAS Y ACRÓNIMOS

ML	Machine Learning
DL	Deep Learning
GBM	Gradient Boosting Machines
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
CNN	Convolutional Neural Networks
IA	Inteligencia Artificial
AIDS	Artificial Intelligence and Data Science
I+D	Investigación y Desarrollo
AR	AutoRegressive
MA	Moving Average
ARIMA	AutoRegressive Integrated Moving Average
ARIMAX	AutoRegressive Integrated Moving Average with eXogenous variables
SARIMAX	SeasonalAutoRegressive Integrated Moving Average with eXogenous variables
GBDT	Gradient Boosting Decision Trees
XGBoost	Extreme Gradient Boosting
LightGBM	Light Gradient Boosting Machines
FNN	Feedforward Neural Networks
BPTT	BackPropagation Through Time
MACD	Moving Average Convergence Divergence
RSI	Relative Strenght Index
MSE	Mean Squared Error
MAE	Mean Absolute Error
MASE	Mean Absolute Scaled Error
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percent Error
API	Application Programing Interface
IB	Interactive Brokers
TWS	Trader WorkStation

S&P 500	Standard & Poor's 500
OHLCV	Open, High, Low, Close, Volume
EMA	Exponential Moving Average
NaN	Not a Number
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives
NPV	Negative Predictive Value
TNR	True Negative Rate
ACF	AutoCorrelation Function
PACF	Partial AutoCorrelation Function
ARCH	Autoregressive Conditional Heteroskedasticity
GARCH	Generalized Autoregressive Conditional Heteroskedasticity

ANEXO II: INDICADORES TÉCNICOS

Datos del activo principal

- Close: Precio de cierre del activo.
- High: Precio máximo del día.
- Low: Precio mínimo del día.
- Open: Precio de apertura del día.
- Volume: Volumen negociado del activo.

Indicadores técnicos sobre el activo

- porcentaje_cambio: Variación porcentual diaria del precio.
- midpoint: Punto medio entre máximo y mínimo del día.
- SMA_10: Media móvil simple de 10 días del cierre.
- SMA_50: Media móvil simple de 50 días del cierre.
- EMA_10: Media móvil exponencial de 10 días del cierre.
- EMA_50: Media móvil exponencial de 50 días del cierre.
- RSI_14: Índice de fuerza relativa de 14 días (mide sobrecompra/sobreventa).
- MACD: Diferencia entre EMA(12) y EMA(26) (tendencia).
- MACD_signal: Señal de 9 días del MACD (para cruces).
- Volatilidad_10: Desviación estándar de los retornos de 10 días.
- ROC_10: Rate of Change, variación porcentual respecto a 10 días atrás.

Mercado SPY (ETF S&P 500)

- Close_SPY, High_SPY, Low_SPY, Open_SPY: Precios de SPY.
- Volume_SPY: Volumen negociado del SPY.

Mercado TLT (ETF bonos 20+ años)

- Close_TLT, High_TLT, Low_TLT, Open_TLT: Precios de TLT.
- Volume_TLT: Volumen negociado de TLT.

Mercado IWM (ETF Russell 2000, small caps)

- Close_IWM, High_IWM, Low_IWM, Open_IWM: Precios de IWM.
- Volume_IWM: Volumen negociado de IWM.

Mercado IEF (ETF bonos 7–10 años)

- Close_IEF, High_IEF, Low_IEF, Open_IEF: Precios de IEF.
- Volume_IEF: Volumen negociado de IEF.

Mercado VIX (Índice de volatilidad implícita)

- Close_VIX, High_VIX, Low_VIX, Open_VIX: Valores diarios del VIX.
- Volume_VIX: Volumen de contratos/ETF ligados al VIX.

SECTOR (ETF sectorial específico que uses)

- Close_SECTOR, High_SECTOR, Low_SECTOR, Open_SECTOR: Precios diarios del ETF sectorial.
- Volume_SECTOR: Volumen negociado del ETF sectorial.

ANEXO III: INDICADORES TÉCNICOS

Código utilizado para obtención de los ficheros de porcentaje, en el siguiente enlace:

https://github.com/rpalaciossa/curso-series-temporales/blob/master-DIRETIC/VAL_TEST_PREDICCION_fechas_Close_Close.py