



UNIVERSIDADE DA CORUÑA

Facultade de Economía e Empresa

Trabajo de fin de máster

Construcción de un modelo de predicción para activos financieros

Creación de una red neuronal recurrente LSTM
con machine learning

Jorge Briones Viqueira

Tutor: Xosé Manuel Martínez Filgueira

Máster en Banca y Finanzas

Curso académico 2022/2023

Trabajo de Fin de Máster presentado en la Facultad de Economía e Empresa de la Universidade da Coruña para la obtención del Máster en Banca y Finanzas

Resumen

La predicción de la cotización futura de los activos es un objetivo ambicioso en el mundo de los mercados financieros. Las mejoras tecnológicas de los últimos años han originado una carrera entre los grandes fondos de inversión para desarrollar algoritmos más eficientes que mejoren sus rentabilidades anuales. A través de la inteligencia artificial se están creando modelos predictivos cuya misión es batir al mercado a partir del análisis, casi a tiempo real, de múltiples variables.

Este trabajo analiza la inteligencia artificial (IA), centrada en la rama del aprendizaje automático o *machine learning* en inglés, para investigar el funcionamiento de los modelos predictivos en la bolsa de valores. A lo largo del trabajo se examina la IA desde un enfoque teórico, centrado principalmente en el aprendizaje profundo (*Deep learning*) y en las redes neuronales, para abordar una implementación práctica de una red neuronal recurrente de tipo LSTM. A partir de los datos históricos de los *blue chips* de España, se construye, entrena y evalúa un modelo que realiza pronósticos de las cotizaciones futuras. Se calculan métricas de error para estudiar el rendimiento de los modelos y se presentan los resultados predichos en una comparativa respecto a los precios reales de las acciones estudiadas.

Palabras clave: Inteligencia artificial, aprendizaje automático, aprendizaje profundo, modelo predictivo, LSTM.

Número de palabras: 18.906

Abstract

Predicting the future price of financial assets is an ambitious goal in the world of financial markets. Technological improvements in recent years have led to a race among large investment funds to develop more efficient algorithms to improve their annual returns. Through artificial intelligence, predictive models are being created whose mission is to beat the market based on the analysis, almost in real time, of multiple variables.

This paper analyses artificial intelligence (AI), focusing on the branch of machine learning to investigate the performance of prediction models in the stock market. Throughout the paper, AI is examined from a theoretical approach, mainly focused on deep learning and neural networks, to address a practical implementation of a recurrent neural network of LSTM type. Based on historical data of Spanish blue chips, a model that performs forecasts of future stock prices is built, trained and evaluated. Error metrics are calculated to study the performance of the models and the predicted results are presented in a comparison with respect to the actual prices of the stocks studied.

Keywords: Artificial Intelligence, machine learning, deep learning, prediction model, LSTM.

Índice

1. Introducción.....	9
2. La inteligencia artificial.....	11
2.1. ¿Qué es la Inteligencia Artificial?	11
2.2. Aplicaciones de la Inteligencia Artificial	12
2.3. La inteligencia artificial en el sector financiero.....	13
3. Aprendizaje Automático (Machine Learning).....	15
3.1. Tipos de aprendizaje automático	16
3.2. Principales técnicas de machine learning	17
3.2.1. Redes neuronales.....	17
3.2.2. Regresión lineal	20
3.2.3. Regresión logística.....	20
3.2.4. Agrupación en clústeres.....	21
3.2.5. Árboles de decisión	21
3.2.6. Bosques aleatorios	22
3.3. Desafíos y limitaciones del <i>Machine Learning</i>	22
4. Deep Learning	24
4.1. La evolución del Deep Learning	24
4.2. Tipos de redes neuronales	25
4.2.1. Redes Neuronales Convolucionales (CNN).....	25
4.2.2. Redes Neuronales Recurrentes (RNN)	26
4.2.3. Redes Neuronales Generativas Antagónicas (GAN).....	27
4.2.4. Redes neuronales Transformadoras (<i>Transformer</i>)	27
5. Planteamiento del problema	30

5.1.	Series temporales.....	30
5.2.	Elección del activo financiero.....	32
5.3.	Diseño de la solución.....	33
6.	Implementación práctica	35
6.1.	Uso de librerías.....	36
6.2.	Análisis y modelado de datos.....	37
6.3.	Tratamiento de los datos históricos.....	39
6.4.	Construcción de la red neuronal recurrente y ajuste de los hiperparámetros.....	40
6.5.	Entrenamiento y prueba de la red neuronal.....	43
6.6.	Predicción de las cotizaciones futuras	45
6.7.	Métricas de rendimiento.....	45
7.	Resultados del modelo	47
7.1.	Descarga y visualización de los precios históricos	47
7.1.1.	Datos históricos de Inditex.....	47
7.1.2.	Datos históricos de Iberdrola	48
7.1.3.	Datos históricos del Banco Santander.....	49
7.1.4.	Datos históricos del BBVA.....	50
7.1.5.	Datos históricos de Amadeus.....	52
7.2.	Construcción y entrenamiento del modelo	53
7.3.	Test y evaluación del rendimiento del modelo.....	56
7.3.1.	Test y predicción sobre Inditex	56
7.3.2.	Test y predicción sobre Iberdrola.....	57
7.3.3.	Test y predicción sobre Santander.....	58
7.3.4.	Test y predicción sobre BBVA.....	59
7.3.5.	Test y predicción sobre Amadeus	60
7.4.	Predicciones futuras.....	61

7.4.1.	Predicción futura de Inditex	61
7.4.2.	Predicción futura de Iberdrola	62
7.4.3.	Predicción futura de Santander	63
7.4.4.	Predicción futura de BBVA	64
7.4.5.	Predicción futura de Amadeus	65
8.	Estrategia de inversión	66
8.1.	Análisis del mercado	66
8.2.	Correlación con el mercado	68
8.3.	Método de inversión	70
9.	Conclusiones	73
Anexo	76
Bibliografía	82

Índice de figuras

Ilustración 1. Relación entre IA, ML, Deep Learning y Red Neuronal.....	15
Ilustración 2. Representación de una neurona artificial de McCulloch-Pitts	18
Ilustración 3. Red neuronal artificial profunda	18
Ilustración 4. Comparativa de los modelos con mayor número de parámetros	28
Ilustración 5. Componentes teóricas de una serie temporal.....	31
Ilustración 6. Proceso para la creación del modelo de machine learning	34
Ilustración 7. Los 10 lenguajes de programación más utilizados en 2023.....	35
Ilustración 8. Librerías importadas en el programa	36
Ilustración 9. Intervalo de tiempo escogido para el entrenamiento del modelo	39
Ilustración 10. Normalización de los datos históricos para el entrenamiento.....	40
Ilustración 11. Sentencias para la construcción del modelo con capas LSTM	41
Ilustración 12. Sentencia para el entrenamiento del modelo creado	42
Ilustración 13. Intervalo de tiempo escogido para el testeo del modelo	43
Ilustración 14. Preparación de los datos para la prueba del modelo	44
Ilustración 15. Realización de predicciones para la prueba del modelo	44
Ilustración 16. Realización de las predicciones futuras	45
Ilustración 17. Cálculo de las métricas de rendimiento	46
Ilustración 18. Cabecera de los datos históricos de Inditex.....	47
Ilustración 19. Cabecera de los datos históricos de Iberdrola	48
Ilustración 20. Cabecera de los datos históricos de Santander.....	50
Ilustración 21. Cabecera de los datos históricos de BBVA.....	51
Ilustración 22. Cabecera de los datos históricos de Amadeus	52
Ilustración 23. Beneficios y pérdidas posibles con la compra de una call y una put	71

Índice de gráficos

Gráfico 1. Evolución de las acciones de las grandes tecnológicas en el índice NASDAQ	32
Gráfico 2. Evolución de las acciones de las grandes empresas españolas en el IBEX 35	33
Gráfico 3. Formas de ahorro de los españoles en los últimos en el año 2021. Fuente: ECF y CNMV	39
Gráfico 4. Evolución del precio de cierre ajustado de las acciones de Inditex (2013-2023)	48
Gráfico 5. Evolución del precio de cierre ajustado de las acciones de Iberdrola (2013-2023)	49
Gráfico 6. Evolución del precio de cierre ajustado de las acciones de Santander (2013-2023)	50
Gráfico 7. Evolución del precio de cierre ajustado de las acciones de BBVA (2013-2023)	51
Gráfico 8. Evolución del precio de cierre ajustado de las acciones de Amadeus (2013-2023)	53
Gráfico 9. Visualización de la construcción del modelo en tiempo de ejecución ..	53
Gráfico 10. Visualización del entrenamiento del modelo creado, en tiempo de ejecución	54
Gráfico 11. Evolución de la pérdida a medida que avanza el entrenamiento	55
Gráfico 12. Iteraciones del entrenamiento	55
Gráfico 13. Test del modelo con precios históricos de Inditex.....	56
Gráfico 14. Test del modelo con precios históricos de Iberdrola	57
Gráfico 15. Test del modelo con precios históricos de Santander.....	58
Gráfico 16. Test del modelo con precios históricos de BBVA.....	59
Gráfico 17. Test del modelo con precios históricos de Amadeus	60

Gráfico 18. Predicciones futuras de los precios de las acciones de Inditex.....	61
Gráfico 19. Predicciones futuras de los precios de las acciones de Iberdrola	62
Gráfico 20. Predicciones futuras de los precios de las acciones de Santander....	63
Gráfico 21. Predicciones futuras de los precios de las acciones de BBVA.....	64
Gráfico 22. Predicciones futuras de los precios de las acciones de Amadeus IT Group	65
Gráfico 23. Test del modelo con precios históricos del Ibex 35.....	67
Gráfico 24. Predicciones futuras de los precios del índice Ibex 35	68
Gráfico 25. Correlación entre IBEX35, Inditex, Iberdrola, Santander, BBVA y Amadeus (2013-2023)	69

1. Introducción

El *machine learning*, traducido al español como aprendizaje automático, es un campo de la ciencia computacional cuyo objetivo es “enseñar” a los ordenadores a reconocer patrones dentro de los datos, introducidos como *inputs*, para devolver predicciones sobre estos, los *outputs*.

En los últimos años, los avances tecnológicos han propiciado que el término de *machine learning* se haya propagado y sea un vocablo reconocido por personas ajenas al mundo de las ciencias de la computación. Este concepto está directamente relacionado con la que ha sido la palabra de moda en España en el año 2022, la Inteligencia Artificial o IA (FundéuRAE, 2022). De esta forma, podemos clasificar el *machine learning* (ML) como una rama de la IA.

La inteligencia artificial tiene como objetivo crear un sistema inteligente autónomo que sea capaz de emular el comportamiento de los seres humanos para el análisis y la resolución de problemas.

En los últimos años, las empresas se han dado cuenta de que disponen de una cantidad ingente de datos y han decidido sacarles partido. En sectores como el sanitario, el financiero, el transporte y, por supuesto, el tecnológico, se ha evidenciado como el análisis de los datos proporciona herramientas para tomar decisiones sobre el negocio. En el mundo académico han surgido numerosos másteres y cursos con los títulos de “Big Data” o “Data Science”, y, cada vez, son más las empresas que buscan perfiles en este ámbito (BBVA, 2023).

El presente proyecto tiene como objetivo la investigación de cómo sacar partido a la gran cantidad de datos históricos de los que se dispone acerca de los mercados financieros. Con ese propósito se estudia en profundidad el aprendizaje profundo y las redes neuronales, para su puesta en práctica mediante la creación de un modelo que pueda predecir la cotización futura de los activos financieros.

El estado del arte en la construcción de este tipo de modelos de predicción se encuentra en constante evolución, con un crecimiento exponencial en los últimos años. El creciente enfoque en la aplicación de técnicas de aprendizaje profundo como método para identificar patrones y tendencias a largo plazo son áreas de investigación activas y de interés para pequeños y grandes inversores.

Para llevar a cabo este estudio y su posterior implementación práctica, el trabajo se divide en dos bloques principales: el teórico y el práctico. El primer bloque se subdivide a su vez en tres capítulos. En el primero se expone la historia y estado del arte de la inteligencia artificial. En el segundo capítulo se analizan los tipos y las técnicas diferentes del aprendizaje automático, así como los desafíos futuros que se presentan. En el último capítulo de este bloque, se estudia el aprendizaje profundo y se detallan los diferentes tipos de redes neuronales existentes hasta la fecha.

El segundo bloque se centra en la aplicación práctica de la teoría vista con anterioridad. Se comienza planteando el problema para intentar encontrar la solución más eficiente. A continuación, se procede a la implementación práctica mediante una red LSTM en Python, alimentada con los precios históricos de las acciones de algunas de las mayores empresas del IBEX 35. Se construye, se entrena y se predice con el modelo para medir su rendimiento y analizar posteriormente sus resultados. En el siguiente apartado, se presentan y explican de manera visual los resultados obtenidos, tanto en las pruebas como en las predicciones futuras. En el último capítulo, se desarrolla una estrategia de inversión para aprovechar las predicciones del modelo.

Finalmente, se exponen las conclusiones, repasando todos los temas tratados en el trabajo y los resultados obtenidos, para emitir un veredicto sobre la viabilidad del modelo de predicción creado.

2. La inteligencia artificial

El nacimiento de la inteligencia artificial se produce como consecuencia de diversos avances científicos en el campo de la computación. En 1854, George Boole introduce los fundamentos de la aritmética computacional que son conocidos como el álgebra de Boole. A principios del siglo XX, comienzan a aparecer referencias a robots como “El hombre de hojalata” del libro “El maravilloso Mago de Oz”, aunque, no es hasta 1921, cuando nace este término en una obra de teatro de Karel Čapek. Más de una década después, en 1936, nace el concepto de algoritmo a raíz de un estudio del famoso padre de la computación moderna, Alan Turing. En los años de la Segunda Guerra Mundial se publican las leyes de la robótica en el cuento “Círculo vicioso” de Isaac Asimov y, en 1950, Alan Turing escribe una prueba para diferenciar una persona de una máquina, conocida como el Test de Turing (National Geographic, 2019).

En 1956, se acuña el término de inteligencia artificial por parte del informático John McCarthy y, unos cuantos meses después, se diseña la primera red neuronal artificial por parte del psicólogo Frank Rosenblatt utilizando las investigaciones de McCulloch y Pitts sobre la creación de redes neuronales en ordenadores (Ramírez, 2018).

En las décadas siguientes surgieron toda clase de inteligencias artificiales como la famosa supercomputadora “Deep Blue”, que derrotó al gran maestro de ajedrez Gary Kasparov y la noticia dio la vuelta al mundo. Hoy en día, la inteligencia artificial ha cosechado tantos avances que ha sustituido la mano de obra humana en muchos trabajos cotidianos y continúa con una progresión sin límites a la vista.

2.1. ¿Qué es la Inteligencia Artificial?

La inteligencia artificial no es otra cosa que la simulación del funcionamiento del cerebro humano en los ordenadores. Para lograr este objetivo, se hace uso de los conocidos algoritmos. La RAE define el algoritmo como un “conjunto ordenado y finito de operaciones que permite hallar la solución de un problema” (Garate, 2021). Un ejemplo cotidiano de un algoritmo podría ser, por ejemplo, una receta de cocina, donde, a partir de unos valores de entrada (ingredientes) se realizan una serie de pasos (proceso) para obtener el plato final (salida).

Hasta este punto, se podría pensar que una inteligencia artificial es semejante a cualquier programa informático, pero, a diferencia de este, la IA tiene capacidad para

aprender de los datos proporcionados por una persona o de forma autónoma. Con estos datos, la IA posee la capacidad de variar su comportamiento en función de los datos recibidos.

Generalmente podemos dividir la inteligencia artificial en IA débil (IAD) e IA fuerte (IAF). A pesar de los avances tecnológicos del último lustro, no se puede afirmar que se haya conseguido programar una IA fuerte. La IAD se desarrolla para un caso de uso específico como puede ser la capacidad de jugar al ajedrez e ir mejorando cada partida jugada. Estas IA están orientadas a problemas concretos, son reactivas, inflexibles y no pueden salirse de la tarea para la que están programadas. En cambio, una IAF puede pensar y comprender los problemas para encontrar soluciones óptimas en diferentes contextos sin la intervención de un ser humano.

2.2. Aplicaciones de la Inteligencia Artificial

La inteligencia artificial se está expandiendo por numerosos campos en los últimos años. Además de la ciencia de computación o las finanzas, a la que se dedicará un apartado más adelante, la IA se usa ampliamente en campos tan variados como la medicina, la industria, el transporte o las telecomunicaciones. En la vida cotidiana se hace uso de la inteligencia artificial diariamente sin percatarse de ello. Desde los asistentes de voz como Alexa de Amazon o Siri de Apple, los smartphones, el buscador de Google, los *chatbots*, la domótica o una aplicación de geolocalización como Google Maps, funcionan con tecnologías que aplican la inteligencia artificial (Thompson, 2019).

En los portales de venta en internet como Amazon aparecen recomendados productos que guardan una estrecha relación con los intereses de los usuarios, en la plataforma musical Spotify aparecen recomendaciones similares a los gustos musicales de los usuarios o una plataforma de *streaming* como Netflix tiene la capacidad de recomendarte una serie o película en función de lo que considera los gustos de los clientes.

En este último año, han aparecido una serie de IA generativas que, a pesar de que ya existían, sus resultados no eran tan buenos como ahora. El caso más conocido es el ChatGPT, un chat que usa una IA para generar texto. La mejor manera para explicar de que se trata es hacerle la pregunta al propio chat. La respuesta sobre esta cuestión es la siguiente:

“ChatGPT es un modelo de lenguaje desarrollado por OpenAI, basado en la arquitectura GPT-3.5. Es un sistema de inteligencia artificial que utiliza el aprendizaje

automático para generar respuestas a preguntas y participar en conversaciones de manera similar a un humano.

El modelo GPT (Generative Pre-trained Transformer) fue entrenado con grandes cantidades de texto para aprender patrones y estructuras del lenguaje. Utiliza un enfoque de "pre-entrenamiento y ajuste fino" para generar respuestas coherentes y contextualmente relevantes.

ChatGPT se ha entrenado en una amplia gama de temas y puede proporcionar información, sugerencias, opiniones y llevar a cabo tareas específicas dentro de sus capacidades. Sin embargo, es importante tener en cuenta que, aunque ChatGPT puede ser útil y proporcionar información precisa en muchos casos, también puede cometer errores y generar respuestas que no son del todo precisas o confiables. Por lo tanto, es recomendable verificar la información importante con fuentes adicionales cuando sea necesario (ChatGPT, 2023)."

2.3. La inteligencia artificial en el sector financiero

En el ámbito de las finanzas son numerosas las aplicaciones que han transformado el sector. En la banca se ha extendido el uso cotidiano de IA para la gestión del riesgo de crédito. En el sector bancario y en el sector de los seguros se utilizan diferentes aplicaciones con inteligencia artificial para evaluar la solvencia previa a la concesión de un préstamo, detectar fraudes o predecir el deterioro de solvencia de un prestatario o el riesgo de *default*.

También, es ampliamente usado en la composición y gestión de carteras debido a su velocidad para analizar e interpretar grandes conjuntos de datos. En la composición se centra en buscar la optimización de una cartera para maximizar rentabilidades y reducir el riesgo mediante la incorporación de múltiples variables de entrada. Por otro lado, en la gestión de carteras se pueden programar estrategias de trading automatizado mediante IA para analizar patrones y tomar decisiones casi de forma instantánea. Un claro ejemplo de esta práctica es el High Frequency Trading (HFT).

La negociación de alta frecuencia, o HFT por sus siglas en inglés, es un tipo de negociación donde el operador es una máquina autónoma y opera comprando y vendiendo activos en los mercados financieros. Este tipo de negociación no es ni mucho menos reciente, ya que su uso data de más de 20 años atrás (Alubankudi & Tapia, 2014).

Respecto a los 2000's, la velocidad de transacción ha pasado de algunos segundos a tiempos realmente pequeños como microsegundos. En el año 2007 en el índice bursátil estadounidense NASDAQ, los tiempos rondaban los 2 o 3 ms (Hasbrouck & Saar, 2011, p. 2), mientras que apenas unos años después, en ese mismo índice, se podían encontrar tiempos entre envío y respuesta menores a un microsegundo (Ye et al., 2012, p. 34). Algunos autores sugieren que los HFT tienen un efecto positivo en el mercado bursátil por el aumento de la eficiencia del mismo pero provocan un efecto negativo al aumentar la volatilidad de los precios (Latza et al., 2014).

Un famoso caso, ejemplo de esta situación, fue el conocido como *flash crash* el 6 de mayo de 2010. Ese día, el Dow Jones, uno de los índices más importantes del mundo, llegó a caer casi 1000 puntos en la jornada (aproximadamente una caída del 9%). En pocos minutos el índice retrocedió por encima de los 600 puntos los cuales recuperó minutos más tarde. La SEC abrió una investigación acerca de este asunto y se encontraron evidencias de la manipulación del mercado mediante la automatización de fuertes órdenes de venta para disminuir los precios de los activos y poder comprarlos a precios mínimos. Pese a que los HFT habían emitido las órdenes, fue un agente quién había manipulado el sistema (elEconomista.es, 2010).

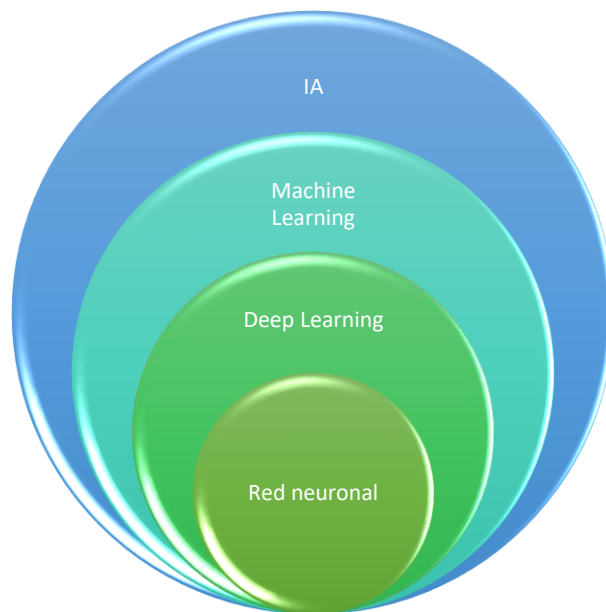
Además de las mejoras en tiempos de respuesta, se han mejorado los algoritmos empleados y se ha implementado la inteligencia artificial para tomar decisiones cercanas al tiempo real. Las innovaciones tecnológicas posibilitan acciones como el arbitraje automático al replicar un índice con ETF, el análisis de las nuevas noticias en numerosos portales de internet o el análisis de sentimientos en redes sociales como Twitter, donde se catalogan los tuits acerca de un índice o valor para predecir el sentido que tomará la cotización en un plazo determinado.

3. Aprendizaje Automático (Machine Learning)

El aprendizaje automático (ML) es el proceso mediante el cual se usan modelos matemáticos de datos para ayudar a un equipo a aprender sin instrucciones directas (Microsoft Azure, 2023). Mediante el uso de algoritmos se crean modelos de datos con capacidad de hacer predicciones de forma similar a como los realizaría un ser humano. A menudo, estos modelos adquieren mayor precisión que las personas y los tiempos de predicción son sumamente eficientes en comparación con cualquier persona. El modelo creado es escalable y continúa mejorando con la inserción de nuevos datos.

Una vez vistos los conceptos de IA y *machine learning* uno se puede preguntar qué diferencia existe entre ellos. Como se definió anteriormente, se puede decir que el *machine learning* es una rama o subconjunto de la inteligencia artificial y ahí radica su estrecha relación. A su vez, el *deep learning* (concepto que se definirá más adelante) es una técnica específica del *machine learning* que usa las redes neuronales interconectadas para simular el funcionamiento del cerebro humano.

Ilustración 1. Relación entre IA, ML, Deep Learning y Red Neuronal.



Elaboración propia

3.1. Tipos de aprendizaje automático

Existen diferentes tipos de Machine Learning que varían según el autor que se consulte, pero, en esencia, podrían resumirse en: aprendizaje supervisado, en el no supervisado, en el semi-supervisado y en el aprendizaje por refuerzo (Bobadilla, 2021, p. 14).

- Aprendizaje supervisado: El entrenamiento consiste en el uso de ejemplos etiquetados previamente por una persona. Por ejemplo, si queremos entrenar un modelo de reconocimiento de imágenes podemos poner fotos de animales y etiquetar cada animal por su nombre para que el modelo pueda predecir que animal aparece en una imagen sin haberla visto anteriormente. Este tipo de aprendizaje es relativamente sencillo, pero requiere de una gran cantidad de datos y horas de entrenamiento en función de las categorías que se quieran entrenar. En el sector financiero se utilizan, por ejemplo, para la categorización de movimientos bancarios o para la detección de fraude en las transacciones.
- Aprendizaje no supervisado: A diferencia del anterior, el no supervisado usa información sin etiquetar. Al no disponer de información acerca de los ejemplos introducidos como valores de entrada, el modelo debe tener la capacidad de reconocer patrones comunes en los datos para hacer el etiquetado de las entradas. Se usa habitualmente en problemas de agrupamiento o *clustering*, por ejemplo, para realizar una segmentación de clientes o para agrupar productos por categorías en un portal de comercio electrónico.
- Aprendizaje semi-supervisado: En el aprendizaje semi-supervisado se puede encontrar una combinación de las dos técnicas anteriores, teniendo conjuntos de datos de entrada etiquetados y sin etiquetar. Este tipo es óptimo para modelos donde la cantidad de datos a entrenar es enorme y su tiempo de etiquetado haría imposible realizar la tarea. Cuando se introducen los datos sin etiquetar, el propio algoritmo los etiqueta, siendo este proceso conocido como *pseudoetiquetado*.
- Aprendizaje por refuerzo: El algoritmo avanza pasos y recibe recompensas o penalizaciones en función de los resultados. Actualmente, muchos de los sistemas de inteligencia artificial comienzan su desarrollo usando este tipo de aprendizaje. Pese a sus grandes pros debido a su relación con el aprendizaje propio de los seres vivos y a los resultados positivos que se

obtienen en la industria de los videojuegos, este tipo de aprendizaje no suele ser eficiente para tareas específicas y no suele aplicarse a nivel empresarial. El motivo principal es que el desarrollador del algoritmo es propenso a condicionarlo al aplicarle las recompensas o penalizaciones en función de sus gustos o su pensamiento.

3.2. Principales técnicas de machine learning

Tras conocer mejor los métodos que usa un modelo de machine learning para aprender, hay que adentrarse en los algoritmos más usados para la creación de los modelos. La clasificación se realiza principalmente con paradigmas desarrollados por modelos estadísticos como la regresión logística o lineal y por la inteligencia artificial, como son las redes neuronales. Los algoritmos más empleados se basan en redes neuronales, regresiones lineales y logísticas, agrupaciones en clústeres, árboles de decisión y bosques aleatorios (IBM, s. f.).

3.2.1. Redes neuronales

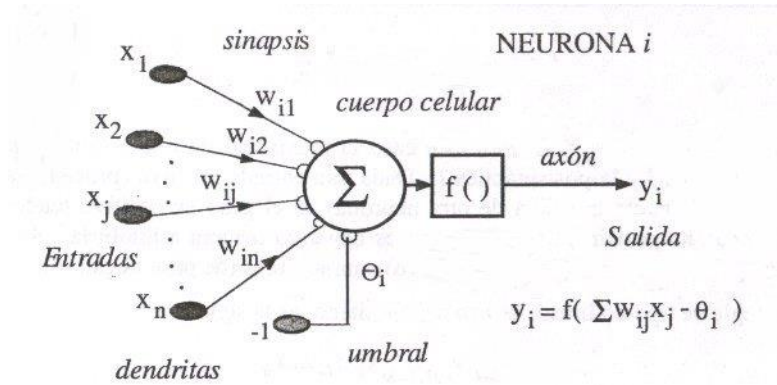
Las redes neuronales artificiales (RNA) son modelos matemáticos que se desarrollaron con el objetivo de simular el comportamiento de las neuronas del cerebro humano.

El primer modelo neuronal conocido fue propuesto por McCulloch y Walter Pitts en el año 1943. Este modelo fue la base para el desarrollo de otros modelos de redes neuronales hasta nuestros días. A pesar de que el modelo neuronal biológico no es adecuado para describir las operaciones realizadas en el cerebro por las neuronas humanas, este primer modelo ha servido de inspiración para la creación de nuevos modelos y ha sido fundamental para el desarrollo del campo de la computación (Prieto Meléndez et al., 2000, p. 6).

La red neuronal artificial se estructura de forma similar a las neuronas del cerebro humano y, por tanto, cuentan con niveles jerárquicos. Cada nivel de la jerarquía procesa la información y la transmite al siguiente nivel. El elemento más básico de la red neuronal es la neurona en su forma artificial, denominada nodo. Este recibe un conjunto de entradas ponderadas, que son el resultado de multiplicar estas entradas por sus respectivos pesos sinápticos. Los pesos determinan la importancia relativa de cada entrada en el cálculo de la salida del nodo y la intensidad de interacción entre las conexiones neuronales. Después, se aplica una función de activación para determinar si el nodo se activa o no, como la sigmoide o la función ReLU (Rectified Linear Unit). Al aplicar la

función a la suma ponderada de las entradas junto al sesgo se obtiene la salida del nodo. Esta salida se trasmite al resto de nodos conectados en la red para la propagación de la información por el conjunto (Parra, 2019, Capítulo 6).

Ilustración 2. Representación de una neurona artificial de McCulloch-Pitts

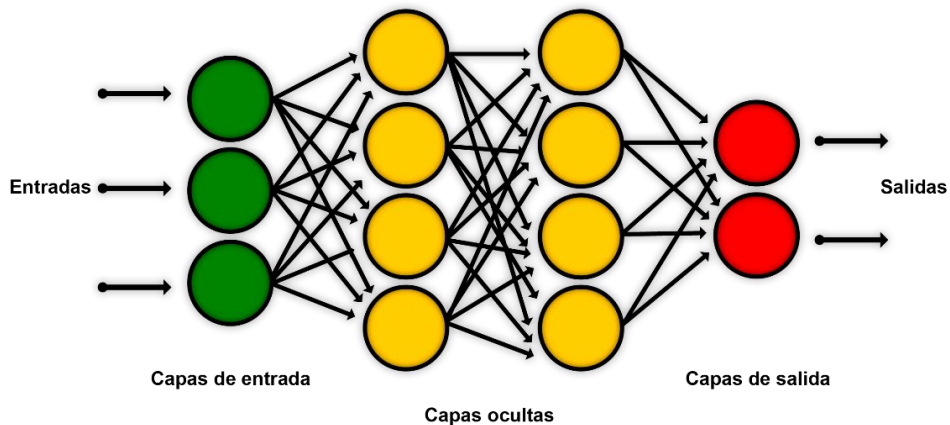


Fuente: www.medium.com, La estructura de los datos (Jiménez, 2018).

La red neuronal está compuesta por una capa de entrada dónde se reciben los *inputs* del sistema (el *input* viene en forma de señal numérica), una capa oculta para el procesamiento de la información introducida y una capa de salida dónde se recibe la información procesada y se devuelve un valor, el *output*.

Las redes neuronales funcionan mediante la interconexión de nodos y pueden contar con múltiples capas ocultas.

Ilustración 3. Red neuronal artificial profunda



Elaboración propia

Las redes neuronales son eficaces capturando patrones entre los datos y se utilizan cada vez más en el reconocimiento y la creación de imágenes o en el procesamiento del lenguaje natural (NLP). Este último, es una tecnología de *machine learning* que brinda a las computadoras la capacidad de interpretar, manipular y comprender el lenguaje humano (AWS Amazon, 2023). Mediante la réplica de un procesamiento de texto de manera similar al entendimiento humano, el programa puede analizar ingentes cantidades de texto en diferentes idiomas o dialectos y procesar estos datos para extraer información.

El enfoque en el desarrollo de algoritmos y técnicas para el análisis, así como la generación de lenguaje humano se utiliza ampliamente en la mayoría de las grandes empresas de todo el mundo. Estamos acostumbrados a hacer uso de sus aplicaciones prácticas sin darnos cuenta. Algunas de sus aplicaciones prácticas incluyen:

- **Búsqueda de información:** NLP se utiliza para mejorar los resultados de búsqueda en motores de búsqueda y plataformas de redes sociales. Por ejemplo, cuando introducimos una palabra mal escrita en el buscador de Google y éste nos muestra una sugerencia de búsqueda interpretando que nuestro texto de búsqueda podría estar mal escrito. También se utiliza en la predicción de una búsqueda, cuando sugiere o se completa automáticamente la frase de búsqueda a partir de unas letras o palabras introducidas.
- **Asistentes virtuales:** NLP se utiliza en asistentes virtuales como Siri, Alexa y Google Assistant para entender las preguntas y comandos de los usuarios. De esta forma se procesa rápidamente la información y se obtienen respuestas casi de forma instantánea.
- **Análisis de sentimientos:** NLP se utiliza para analizar los sentimientos de los usuarios en redes sociales, comentarios en línea y otras formas de comunicación. Con esta aplicación se pueden desde realizar campañas electorales, hasta analizar los tweets de las personas acerca de una marca o sobre los sentimientos en los mercados de valores. Por ejemplo, el índice de miedo y codicia para las criptomonedas (Crypto Fear & Greed Index) recibe los sentimientos de los usuarios de diferentes redes sociales y los condensa junto con otras variables, otorgándole un 15% (a la actividad en RRSS) de ponderación en la elaboración de este índice (Alternative.me, s. f.).

- Traducción automática: NLP se utiliza para traducir automáticamente textos de un idioma a otro. La capacidad de procesamiento ha mejorado en gran medida las traducciones debido a que puede interpretar contextos y no solo traducir literalmente los textos por palabras de forma individual.
- *Chatbots*: NLP se utiliza para desarrollar *chatbots* que pueden entender y responder a las preguntas de los clientes en una variedad de contextos. Hoy en día, los *chatbots* han sustituido a muchos de los operadores humanos en la resolución de problemas recurrentes de fácil respuesta o solución.
- Procesamiento automático de documentos: NLP se utiliza para automatizar tareas como la clasificación, la indexación y la extracción de información de documentos. La enorme información contenida en documentos antiguos sin indexar como en los nuevos documentos, se puede procesar y, con ello, extraer información que puede llegar a ser muy valiosa.

3.2.2. Regresión lineal

La regresión es una técnica usada por el *machine learning* supervisado para la predicción cuando la variable objetivo es continua. En el caso de la regresión lineal, el algoritmo trata de ajustar una línea recta sobre los datos. La resolución del problema se realiza mediante el método de mínimos cuadrados, minimizando la suma de los cuadrados de los errores.

El principal problema de este algoritmo es la asunción de que existe una combinación lineal entre las variables, lo cual no tiene que ser necesariamente cierto. La más común es la regresión lineal simple que se caracteriza por contar con una sola variable independiente.

3.2.3. Regresión logística

La regresión logística es otra técnica usada por el *machine learning* supervisado, pero, a diferencia de la regresión lineal, se utiliza, principalmente, en problemas de clasificación con variables discretas. La variable objetivo tan sólo puede tomar dos valores, como "Sí" y "No", o "0" y "1".

Mediante una función logística, la función sigmoide, se realiza una estimación de la probabilidad de que cada observación pertenezca a una categoría o clase. La llamada sigmoide, toma un valor real escalado entre los valores de 0 y 1. Dentro de esos valores, se establece un umbral y la observación se clasifica en una categoría u otra dependiendo de si el valor es menor o mayor que el umbral establecido.

Para estimar los diferentes coeficientes del modelo, el algoritmo de regresión logística usa el método conocido como máxima verosimilitud en el cual el estimador tomará como valor del parámetro aquel que produciría una muestra con la mayor probabilidad. El objetivo de la fórmula empleada es maximizar esta función de verosimilitud.

Este método es eficiente y obtiene tiempos de computación bajos en comparación con otros tipos de algoritmos más complejos, como las redes neuronales artificiales. Es usado en multitud de aplicaciones como la detección de *spam*.

La diferencia respecto a la regresión simple radica en la salida de la predicción. Por un lado, la regresión lineal predice valores continuos, y, por otro lado, la logística devuelve valores discretos.

3.2.4. Agrupación en clústeres

El conocido como *clustering*, anteriormente mencionado en el apartado de aprendizaje no supervisado es otra de las principales técnicas de *machine learning*. Mediante este método, el conjunto de datos se divide en grupos de datos similares entre sí y se diferencia del resto de grupos. Mediante esta exploración se identifican anomalías y se realizan predicciones en base a la similitud. Es una opción recomendable para explorar datos y detectar correlaciones que no se deducen a simple vista en un conjunto de datos.

El método de *clustering* más conocido es el algoritmo *K-means*. Este algoritmo basa en la minimización de la suma de los cuadrados de las distancias entre los objetos y el centroide, que es el punto que representa cada uno de los clústeres.

Esta agrupación se utiliza ampliamente como método para segmentar a la clientela o para agrupar documentos u otras formas posibles de agrupaciones.

3.2.5. Árboles de decisión

Los árboles de decisión son otra técnica de *machine learning* supervisado que se emplean para resolver problemas de clasificación o regresión (valores numéricos). Utilizan la famosa técnica de computación de divide y vencerás y se estructuran en forma de árbol en función de sus características, de ahí el nombre. Estas características son representadas mediante un nodo y se unen mediante ramificaciones (decisiones) con otros nodos del árbol.

Los árboles de decisión se construyen recursivamente, al dividir el conjunto de datos en subconjuntos de menor tamaño en base a sus características más relevantes. Esta división se realiza, o bien, minimizando el error de regresión, o bien, eligiendo la característica más adecuada.

A diferencia de las redes neuronales, estos árboles son fáciles de diseñar e interpretar, pero, a menudo, tienden al sobreajuste del modelo si no se tiene cuidado con la profundidad de este.

3.2.6. Bosques aleatorios

Los bosques aleatorios guardan una estrecha relación con los árboles de decisión y son otra técnica de *machine learning* supervisado. Conocidos por su nombre en inglés como *random forest*, destacan por su precisión, flexibilidad y menor complejidad que otros métodos. Este método es una combinación de árboles de decisión que se entrenan mediante muestras aleatorias de un conjunto de datos y se combinan aleatoriamente sus características. Cada árbol realiza una clasificación y se combinan sus respuestas para devolver una predicción final.

A diferencia de los árboles de decisión, tienen mayor resistencia al sobreajuste y manejan datos con múltiples dimensiones. Sus aplicaciones principales son ampliamente usadas en la actualidad, como la clasificación de imágenes o la detección de fraudes.

3.3. Desafíos y limitaciones del *Machine Learning*

Las aplicaciones de *machine learning*, si bien ofrecen grandes promesas, también enfrentan desafíos y limitaciones significativas en el mundo real. Uno de estos desafíos es la vulnerabilidad al sesgo y al error humano. Al basarse en datos previos, los algoritmos pueden contener sesgos inherentes, conduciendo a resultados discriminatorios. Un claro ejemplo, se encuentra, por ejemplo, en el sector bancario con el uso del *machine learning* en la concesión de una hipoteca, que representa uno de los mayores riesgos para un banco, el riesgo de crédito.

“El mayor rendimiento predictivo de ML también puede conllevar efectos positivos en la inclusión financiera, aunque estos resultados deben ser valorados junto con nuevos riesgos que vendrían de la mano de esta tecnología, como son especialmente la discriminación (sesgos) entre clases sociales (Alonso-Robisco & Carbo, 2023, p. 79)”.

Otro de los aspectos desafiantes de los modelos de *machine learning* es la nula o difícil auditabilidad. Debido a la enorme complejidad de los algoritmos, sus resultados no pueden ser fácilmente explicados o comprendidos por los seres humanos. Al encontrarse con esta opacidad, no existe transparencia en el modelo y no se puede comprender en ocasiones como se alcanza un resultado.

Afortunadamente, el desarrollo cada vez mayor de este campo, se complementa con el desarrollo de herramientas y directrices para las buenas prácticas. Además, se cuenta con mecanismos de supervisión y auditoría de las tomas de decisiones basadas en el machine learning.

4. Deep Learning

El *deep learning*, o aprendizaje profundo en castellano, proporciona un enfoque multicapa para aprender representaciones de datos. Esto se realiza construyendo una red neuronal con múltiples capas. A diferencia de los conocidos modelos superficiales que cuentan con solo 1 o 2 capas para procesar e interpretar los datos, los modelos profundos cuentan con 3 o más capas. Con una sola capa, cualquier red neuronal posee la habilidad de aprendizaje, pero, la incorporación de un mayor número de capas (capas ocultas) le permite tener mayor precisión y optimización. Aunque el hecho de tener más capas no siempre es mejor ya que puede empeorar notablemente el tiempo de ejecución y producir sobreajustes.

Como ocurre con otros algoritmos de aprendizaje automático, las redes neuronales profundas (DNN, por sus siglas en inglés) logran el aprendizaje al asignar características a objetivos a través de un proceso de transformación de datos básicos y retroalimentación de señales. Sin embargo, las DNN se destacan al priorizar el aprendizaje de sucesivas capas de representaciones significativas. A pesar de que pueda parecer un tema imponente, el concepto subyacente es bastante sencillo y ha demostrado ser sumamente eficaz en una amplia gama de problemas, como la clasificación de imágenes, el reconocimiento del habla y la conducción autónoma.

4.1. La evolución del Deep Learning

El *deep learning* surge con el concepto de redes neuronales en los años 40, pero, estuvo congelado en el tiempo sin apenas avances significativos hasta los años 80, donde resurgió el interés por este campo. En esta década, se desarrollaron algoritmos de aprendizaje profundo con un problema insalvable para la época, la falta de potencia computacional. Con los avances computacionales del siglo XXI se han producido avances significativos, sobre todo, a raíz del aumento de potencia de las GPU.

La conocida empresa estadounidense Nvidia define a la computación acelerada por GPU como “el uso de una unidad de procesamiento de gráficos en combinación con la CPU (*Central Processing Unit*) para acelerar el funcionamiento de las aplicaciones de *deep learning*, análisis e ingeniería (NVIDIA, s. f.)”. Esta innovación viene de la mano de Nvidia en el año 2007 y supuso un momento clave en la historia del aprendizaje profundo. El desarrollo de la GPU fue impulsado por la creciente demanda de la industria

de los videojuegos en aquella época y ha sido derivado a campos científicos por sus notables resultados para el cálculo, en comparación con el funcionamiento de una CPU. Tal es la diferencia, que el minado de criptomonedas se realiza, principalmente, mediante la combinación de GPU, llamado también *rig* de minería.

Además de la mejora en el *hardware*, la disponibilidad de grandes conjuntos de datos etiquetados como, por ejemplo, el proyecto de ImageNet que cuenta con decenas de millones de imágenes etiquetadas a mano, ha servido como base para entrenar y mejorar los modelos de aprendizaje profundo. En 2012, se logró un hito importante en el aprendizaje profundo respecto a este proyecto, ya que una red neuronal, AlexNet, ganó el desafío de clasificación de imágenes de ImageNet. El equipo liderado por Geoff Hinton de la Universidad de Toronto consiguió un 85% de precisión e hizo que las miradas de la comunidad tecnológica se interesaran por este tema innovador basado en una vieja idea (The Economist, 2016).

Pese a los enormes avances en la última década, el Deep Learning es todavía un campo inmaduro y con grandes posibilidades y aplicaciones por explorar.

4.2. Tipos de redes neuronales

Las redes neuronales son un subconjunto del *machine learning* y están estructuradas en capas de entrada, capas ocultas y capas de salida, como hemos visto anteriormente. Sin embargo, existen diferentes tipos de redes neuronales en función de su objetivo o caso de uso. Estos tipos han ido surgiendo y evolucionando a través del tiempo. En el siglo XX surgieron las conocidas redes neuronales recurrentes, seguidas de las redes neuronales convolucionales, ya bien entrado en el siglo XXI. Sin embargo, la explosión de este campo en los últimos años proviene del nacimiento de las redes neuronales generativas adversariales (GAN), en el año 2014, o las redes neuronales transformadoras (modelo *Transformer*), publicadas en un *paper* de trabajadores de Google en el año 2017 (Merino, 2023).

4.2.1. Redes Neuronales Convolucionales (CNN)

La arquitectura de estas redes se fundamenta en la aplicación de filtros convolucionales para extraer las características más significativas de las imágenes (capa convolucional). En la siguiente capa, la capa de agrupación, se reduce la dimensionalidad de las características extraídas, de tal forma que se conserva su información esencial. Por último, se utilizan capas completamente conectadas para realizar la clasificación de las imágenes. Este tipo de redes son usadas para detectar

características y patrones en imágenes como se demostró con la red neuronal AlexNet. Las CNN logran esta enorme precisión debido al etiquetado de cada píxel en una imagen para comprender su estructura y contexto de cara a una posterior clasificación. Su capacidad de reconocimiento de patrones visuales se aprovecha en áreas tan innovadoras como el reconocimiento facial o la conducción autónoma.

4.2.2. Redes Neuronales Recurrentes (RNN)

Las redes neuronales recurrentes se caracterizan por el uso de las series de tiempo. Como hemos visto anteriormente, son ampliamente usados en tareas de procesamiento de lenguaje natural (NLP). A diferencia de las redes neuronales retroalimentadas (*feed-forward*) convencionales cuyas conexiones se mueven en una única dirección sin formar un ciclo, las RNN están dotadas de conexiones recurrentes que les permiten mantener información contextual a lo largo del tiempo. Esta propiedad es esencial para abordar tareas que requieren un procesamiento secuencial y la comprensión de dependencias temporales. Uno de los principales problemas que tienen las RNN es que están limitadas a un entrenamiento secuencial. La imposibilidad de realizar el procesamiento de datos en paralelo ralentiza enormemente el proceso.

Dentro de las RNN existen variantes de arquitecturas relevantes:

- Redes neuronales con memoria a largo plazo (LSTM): Nace en el 1997 y se trata de la arquitectura más popular y la empleada en la realización práctica de este trabajo. Este tipo de arquitectura se caracteriza por la capacidad de recordar datos dentro de la secuencia y preservarlo durante varios ciclos. Este hito se ha conseguido mediante la adición de celdas de memoria en las capas ocultas de la red neuronal profunda. Estas celdas contienen tres puertas principales: de entrada, de salida y de olvido. Mediante estas puertas se regula el flujo de información en la celda de memoria para evitar los problemas de desvanecimiento típico de las redes neuronales recurrentes tradicionales.
- Redes neuronales recurrentes bidireccionales (BRNN): Las redes bidireccionales, como su propio nombre indican, suponen una extensión de las redes neuronales recurrentes unidireccionales, donde la información fluye a través de una única dirección. Su arquitectura se basa en la combinación de dos RNN en la que una se encarga del procesamiento directo de la información y la otra realiza la etapa a la inversa. Cada una de las redes aborda los datos de entrada en una dirección. Al acceder a la

información en toda la escala temporal, posee la capacidad para capturar dependencias contextuales en ambas direcciones.

- Unidades recurrentes con puertas (GRU): Se introducen en 2014 como respuesta al problema común de las RNN del desvanecimiento del gradiente. Las GRU mantienen la capacidad de las LSTM para capturar dependencias a largo plazo, pero, su arquitectura es más simple. Su funcionamiento es parecido al de las redes LSTM, de hecho, se las considera una variante avanzada de estas, pero, a diferencia de estas últimas, cuentan con tan sólo dos puertas: una puerta para el restablecimiento y otra para la actualización. Esta red consta de dos RNN, donde una codifica una secuencia de símbolos y la otra decodifica la representación en otra secuencia. El modelo tiene la capacidad de aprender una representación semántica y sintáctica significativa de las frases lingüísticas (Cho et al., 2014), lo que hace una herramienta poderosa para el procesamiento del lenguaje natural y la generación de texto.

4.2.3. Redes Neuronales Generativas Antagónicas (GAN)

Las redes neuronales generativas antagónicas se utilizan para la generación de imágenes sintéticas. Los datos creados son semejantes a una distribución de datos de entrenamiento dada. La arquitectura de este tipo de red contiene dos redes neuronales profundas enfrentadas, una red neuronal generativa y otra discriminativa. Por una parte, la generadora crea instancias de datos nuevas y, por otra parte, la discriminadora tiene la tarea de discernir las instancias de datos reales del *dataset* de entrenamiento, de los datos de ejemplo falsos creados por la generadora.

A partir de un espacio latente aleatorio, el generador crea muestras sintéticas y el discriminador filtra entre las muestras reales y las sintéticas generadas. Estas subredes operan de manera adversarial, de ahí su nombre (redes neuronales generativas antagónicas o adversariales). A través de este proceso de competencia entre las dos subredes se generan todo tipo de imágenes realistas, voz o la traducción de texto en imágenes, por ejemplo. La mejora mutua que propicia la feroz competencia permite a las GAN generar datos realistas y de alta calidad (MathWorks, s. f.).

4.2.4. Redes neuronales Transformadoras (*Transformer*)

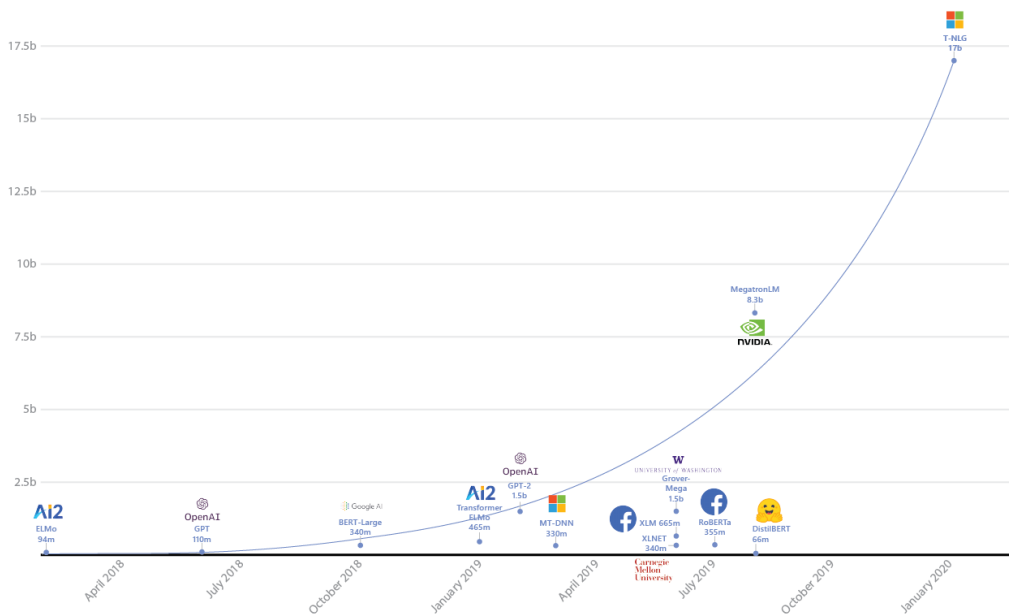
La arquitectura de red simple del *Transformer* o Transformador está basada únicamente en mecanismos de atención, prescindiendo por completo de las recurrencias y las convoluciones (Vaswani et al., 2017). La idea de este tipo de redes

surge del *paper* titulado “Attention Is All You Need”. Este artículo académico publicado por varios miembros de Google (en su mayoría) estableció los principios claves para el exitoso desarrollo de las inteligencias artificiales en estos últimos meses. de modelos como el GPT-4 (versión actual) de la empresa OpenAI.

En los Transformers se añade una nueva capa al modelo, conocida como capa de atención. La clave del funcionamiento de esta innovadora red neuronal es que examina una gran cantidad de datos filtrando los detalles más relevantes para establecer un contexto. Además, esta nueva red permite paralelizar los procesos, a diferencia de las clásicas RNN, permitiendo el entrenamiento o la ejecución con múltiples GPU para acelerar el proceso.

Dentro de estas redes *Transformer* hay que destacar la empresa en auge OpenAI. Esta empresa elevó la inteligencia artificial a otro nivel a través de este tipo de arquitectura. Para el entrenamiento del modelo GPT-3, su modelo estrella, se usaron 175 billones (americanos) de parámetros, una ingente cantidad en comparación con los anteriores mayores modelos, como Turing-NLG de Microsoft, que tenía 17 billones (americanos) de parámetros (Alarcón, 2020).

Ilustración 4. Comparativa de los modelos con mayor número de parámetros



Nota. La gráfica muestra la notable diferencia entre Turing-NLG de Microsoft y otros modelos predecesores con grandes cantidades de parámetros en su entrenamiento. Fuente: www.microsoft.com , figura adaptada de DistilBERT (Sanh et al., 2019, fig. 1).

Además del *chatbot* conocido como ChatGPT basado en GPT-3, OpenAI cuenta con otras IA, también basadas en el mismo modelo, como DALL-E, para la generación de imágenes a partir de texto. Como respuesta al famoso *chatbot*, Google sacó su propia versión, Google Bard, basado en LaMDA. Cabe destacar que, si bien la empresa OpenAI es la pionera en el lanzamiento de un *chatbot* tan preciso y entrenado con tantos miles de millones de parámetros, se debe recordar el mérito de los investigadores de Google al publicar el artículo académico que ha revolucionado la era digital en el último lustro.

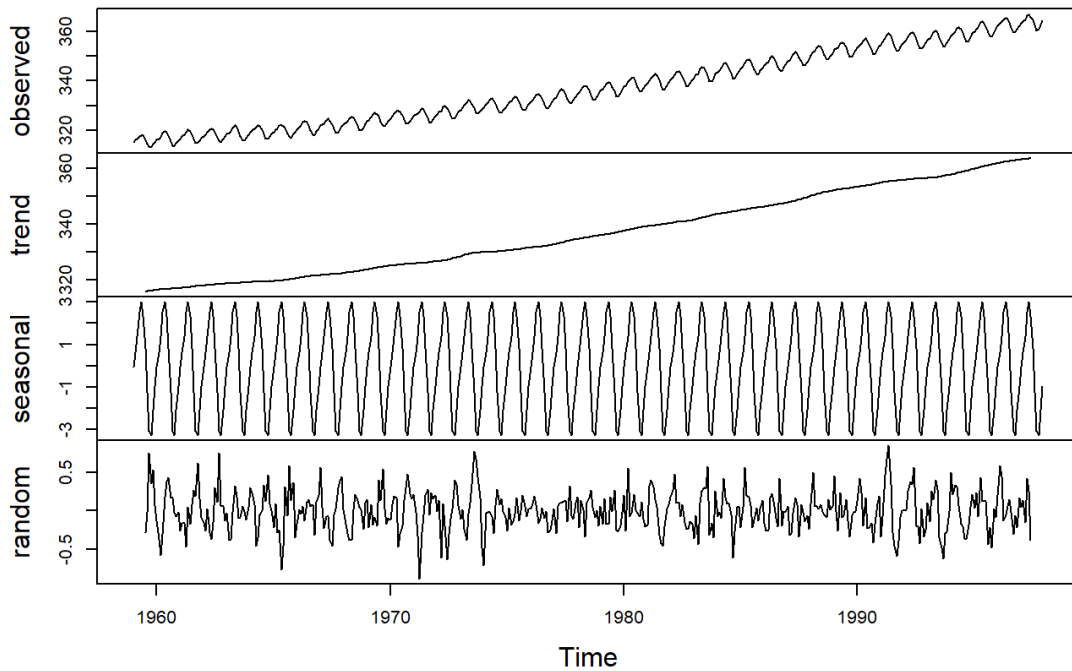
5. Planteamiento del problema

El objetivo que persigue este trabajo es la construcción de un modelo de predicción para activos financieros. La Comisión Nacional del Mercado de Valores (CNMV) define al activo financiero como un valor que acredita la titularidad de determinados derechos políticos, económicos, etc. (CNMV, 2023). Este estudio, se va a centrar únicamente en un activo financiero, las acciones. Pese a que el estudio está referenciado a este activo en concreto, por supuesto, cualquier derivado cuyo activo subyacente sea la acción estudiada se vería de la misma forma influido en la práctica. Se debe dejar claro que el modelo no devuelve recomendaciones de inversión, si no que estudia las posibles cotizaciones futuras a partir de datos históricos relevantes, para tratar de encontrar patrones que puedan repetirse en el tiempo.

5.1. Series temporales

Las cotizaciones de las acciones son series temporales de tipo *stock*, ya que están referidas a una determinada fecha. Las series temporales son el conjunto de observaciones de los datos en intervalos regulares de tiempo (Parra, 2019, Capítulo 8). La teoría clásica divide las series de tiempo en cuatro componentes teóricas que, con la combinación de ellas, producen la evolución de la serie. Estas componentes son: la tendencia, que representa el movimiento en largos periodos de tiempo; las variaciones cíclicas, que son oscilaciones periódicas con media 0 o cercana al nulo; las variaciones estacionales, que son oscilaciones periódicas en cortos plazos de tiempo; los movimientos irregulares, que carecen de periodicidad y son perturbaciones aleatorias sin regularidad.

Ilustración 5. Componentes teóricas de una serie temporal



Nota. En la ilustración se pueden apreciar 3 de las 4 componentes teóricas de una serie temporal: la tendencia (trend), la estacionalidad (seasonal) y los movimientos regulares (random), además de las observaciones (observed). Fuente: Estadística y Machine Learning con R (Parra, 2019, Capítulo 8.3).

Las series temporales se pueden clasificar, dependiendo de sus características, en estacionarias o no estacionarias. Una serie temporal es estacionaria cuando se repiten ciertos patrones a lo largo de la escala del tiempo. En el caso de las series no estacionarias, tienen patrones de repetición, pero no están definidos en un intervalo temporal como es el caso de la cotización de una acción en la bolsa. Según Francisco Parra (2019, Capítulo 8), las técnicas predictivas sobre las series temporales se agrupan en métodos cualitativos y métodos cuantitativos. En este problema, el método más adecuado parece ser el cuantitativo, ya que, a diferencia del cualitativo, sus datos históricos pueden proporcionar información sobre el comportamiento futuro de la acción.

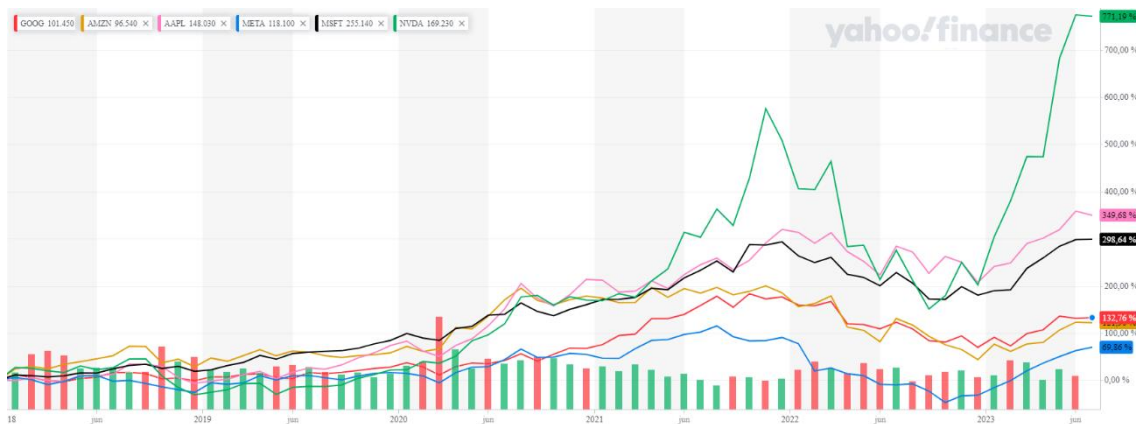
En el ámbito financiero las series suelen ser inestables y, por ello, los modelos más simples de estadística, que resuelven exitosamente otros problemas, carecen de sentido para esta cuestión. En las series temporales de tipo *stock*, se encuentran características como: los clústeres de volatilidad, que son periodos volátiles con grandes volúmenes de negociación, mezclados con otros periodos de escasa volatilidad y menor volumen; el efecto Leverage, donde se incrementa la rentabilidad del capital propio mediante el uso de deuda; el efecto de los periodos sin mercado, donde la información

con el mercado cerrado influye notablemente en la siguiente apertura del mismo; el efecto de comunicación de eventos, donde las comunicaciones corporativas como los informes financieros u otras noticias relevantes provocan alteraciones en la cotización.

5.2. Elección del activo financiero

En un primer momento se pensó en el estudio de los gigantes tecnológicos (Big Tech): Alphabet (Google), Amazon, Apple, Meta (anterior Facebook), Microsoft y, adicionalmente, Nvidia. Aunque la conocida empresa fabricante de tarjetas gráficas (principalmente) no suele incluirse en este grupo, su cotización actual, cercana al trillón de dólares, la sitúa como un gigante del sector tecnológico. Además, este crecimiento exponencial en la valoración de la empresa es resultado en gran parte de la evolución y revolución de las IA generativas y sus cálculos a través de las GPU que fabrica. No obstante, la evolución de la cotización de estas empresas en el último lustro, marcada por un crecimiento exponencial, no permite estudiar el correcto funcionamiento del modelo predictivo.

Gráfico 1. Evolución de las acciones de las grandes tecnológicas en el índice NASDAQ

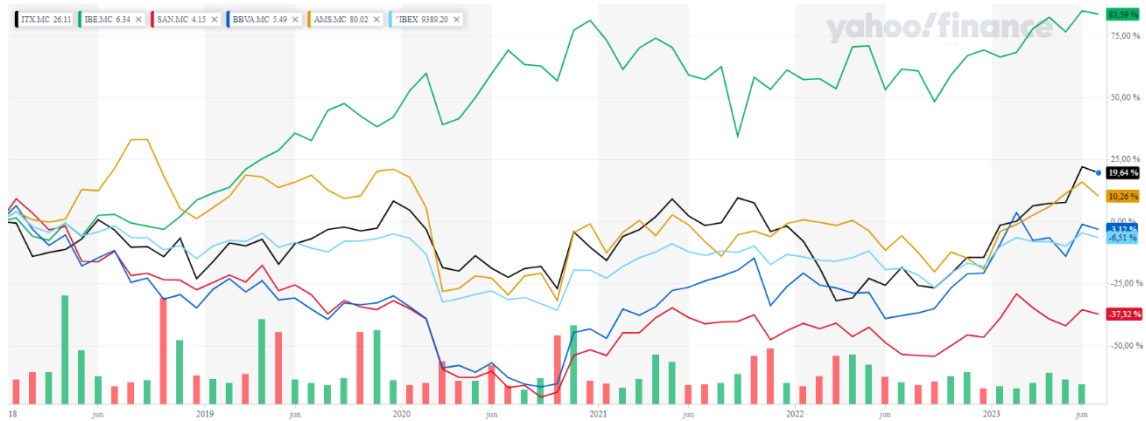


Nota. En el gráfico se puede ver el crecimiento exponencial de las acciones durante los últimos 5 años en el NASDAQ de: Nvidia (en verde), Apple (en rosa), Microsoft (en negro), Google (en rojo), Amazon (en amarillo) y Meta (en azul). Fuente: Elaboración propia a partir de datos extraídos de Yahoo Finance.

En lugar de las gigantes del índice NASDAQ 100, se ha preferido el estudio de las *blue chips* españolas. Se conocen como las *blue chips* a las empresas con mayor capitalización bursátil y, en España, serían, en junio de 2023: Inditex, Iberdrola, Banco Santander, BBVA y Amadeus IT Group. Históricamente, un blue chip por excelencia era Telefónica, pero, su mal rendimiento en bolsa a lo largo de la última década le ha hecho

retroceder varios puestos. En resumen, nos encontramos con: una empresa líder en el sector textil, Inditex; una gran empresa del sector energético, Iberdrola; dos grandes bancos a nivel europeo, Santander y BBVA; y una empresa del sector tecnológico y turístico, Amadeus IT Group.

Gráfico 2. Evolución de las acciones de las grandes empresas españolas en el IBEX 35



Nota. En el gráfico se puede apreciar la evolución de los 5, además del propio índice, en los últimos años. Se encuentran ordenados de arriba hacia abajo por rendimiento: Iberdrola (en verde), Inditex (en negro), Amadeus (en amarillo), BBVA (en azul oscuro), el índice Ibex 35 (en azul claro) y el Banco Santander (en rojo). Fuente: Elaboración propia a partir de datos extraídos de Yahoo Finance.

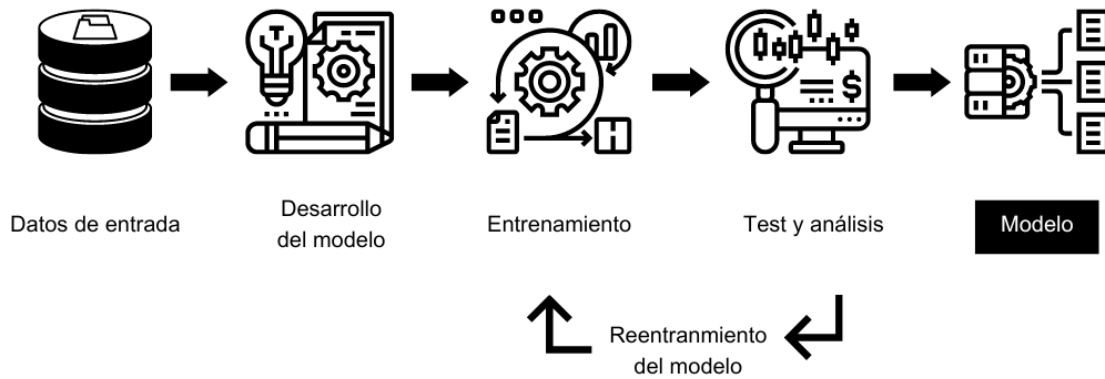
5.3. Diseño de la solución

Una vez vista la teoría acerca de las series temporales relacionadas con la bolsa de valores, y, después de haber seleccionado los valores para el estudio del modelo, el siguiente paso es el diseño de la solución a la cuestión planteada.

Debido a la dificultad en la predicción de las acciones en Bolsa, se debe recurrir a soluciones más potentes y complejas. Por ello, se ha optado por la creación de redes neuronales recurrentes.

Los pasos por efectuar en la implementación de la solución son: introducir los datos históricos; crear el modelo y entrenarlo con esos datos; probar, analizar y reentrenar el modelo; predecir valores futuros; guardar el modelo para su reutilización.

Ilustración 6. Proceso para la creación del modelo de machine learning



Elaboración propia.











Con el diseño teórico de la solución, el siguiente paso es ponerse manos a la obra para seleccionar el lenguaje y las herramientas necesarias para su implementación práctica. Perceptiblemente, es necesario el uso de un *hardware* con la suficiente potencia para realizar los cálculos que exige el modelo. El tiempo que demore la resolución dependerá enormemente tanto del *software* como del *hardware* utilizado en la práctica.

6. Implementación práctica

Las cuestiones acerca de la construcción de un modelo para la predicción de activos financieros con *machine learning* son numerosas. En primer lugar, el lenguaje de programación seleccionado para esta implementación ha sido Python. Tradicionalmente, los lenguajes más usados para estos casos de uso suelen ser R o Python. Si bien R está más orientado al análisis estadístico, Python es más multidisciplinar. Asimismo, la ejecución del código en R es más lenta, aunque posee una potente visualización de datos. Ambos lenguajes ofrecen una amplitud de librerías y herramientas para la construcción rápida y relativamente sencilla de estos modelos.

Python es el lenguaje de programación más popular del mundo en la actualidad. Pese a aparecer a principios de los años 90, su facilidad de aprendizaje, sus enormes posibilidades al ser un lenguaje gratuito, de código abierto y multiparadigma, y, su explosión actual por las opciones que ofrece a científicos de datos y desarrolladores de inteligencias artificiales, le ha permitido posicionarse entre los lenguajes más usados y demandados (*TIOBE Index*, 2023).

Ilustración 7. Los 10 lenguajes de programación más utilizados en 2023

Jun 2023	Jun 2022	Change	Programming Language	Ratings	Change
1	1		 Python	12.46%	+0.26%
2	2		 C	12.37%	+0.46%
3	4	▲	 C++	11.36%	+1.73%
4	3	▼	 Java	11.28%	+0.81%
5	5		 C#	6.71%	+0.59%
6	6		 Visual Basic	3.34%	-2.08%
7	7		 JavaScript	2.82%	+0.73%
8	13	▲	 PHP	1.74%	+0.49%
9	8	▼	 SQL	1.47%	-0.47%
10	9	▼	 Assembly language	1.29%	-0.56%

Nota. En la ilustración puede verse un top de popularidad de los lenguajes de programación, que está encabezado por Python, seguido de C, C++, Java, etc. Fuente: www.tiobe.com (*TIOBE Index*, 2023).

6.1. Uso de librerías

En el desarrollo del programa se han empleado multitud de librerías, optando por cada una de ellas para un propósito específico. Se define una librería en programación como al conjunto de archivos de código utilizados en el desarrollo de *software*. Permiten al usuario implementar funciones predefinidas para agilizar el proceso de escritura y la reutilización de código.

Ilustración 8. Librerías importadas en el programa

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import datetime as dt
import yfinance as yf
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from keras.models import Sequential
from keras.layers import Dense, Dropout, LSTM
```

Elaboración propia

Las librerías seleccionadas para la construcción del modelo son:

- Numpy: Esta librería sirve para el procesamiento numérico en Python. Mediante su uso se pueden realizar operaciones matemáticas de alto rendimiento con grandes conjuntos de datos numéricos como son los precios históricos de una acción.
- Matplotlib: Es una biblioteca que permite la generación de gráficos para visualizar los datos. Proporciona una amplia gama de estilos y opciones de personalización, lo que la convierte en una herramienta versátil para la representación visual de datos.
- Pandas: Esta librería es una de las más populares y permite la manipulación y el análisis de los datos. Mediante su uso se pueden hacer tareas de limpieza, filtrado o transformación, entre otras.
- Datetime: Permite el trabajo con valores de tiempo y fechas. Es usada como contador del tiempo que demora la ejecución de un programa.
- Yfinance: Permite obtener de forma directa los precios históricos u otra información financiera del portal especializado Yahoo Finance dentro del

mismo programa. A diferencia de la descarga manual de un fichero de Excel o '.csv', esta librería permite manejar rápidamente los datos e incluso visualizarlos en la propia ejecución.

- Sklearn: Scikit-learn es la librería usada para el aprendizaje automático. Permite generar modelos de aprendizaje supervisado y no supervisado, evaluarlo y seleccionar las características.
- Keras: Es la API de alto nivel para entrenar y construir modelos de *deep learning* a través de Tensorflow. Introduce la clase 'Sequential' para la construcción del modelo neuronal multicapa. También introduce otras clases como 'LSTM' para el manejo de redes neuronales de este tipo.
- Tensorflow: Es una biblioteca de código abierto que permite preparación previa de los datos para la creación de modelos de *machine learning* y su implementación.

6.2. Análisis y modelado de datos

El modelo creado recibirá como datos de entrada los precios históricos de las empresas seleccionadas para el estudio. Cuando hablamos de precios históricos nos referimos a las cotizaciones que ha tomado en el intervalo del pasado escogido. Para la implementación del modelo se ha elegido un período temporal de 10 años, que se cree suficiente para capturar patrones en la curva de cotizaciones.

Dentro de los precios históricos de una acción, se selecciona el precio diario o precio de una sesión que, a su vez, puede dividirse en precio de inicio (*Open*), precio máximo de la sesión (Máx.), precio mínimo de la sesión (Mín.), precio de cierre (*Close*), precio de cierre ajustado (*Adjusted Close*). Además de los precios expuestos, Yahoo Finance devuelve la fecha correspondiente a cada sesión y el volumen de negociación de esta.

En este modelo se ha optado por usar una única variable de entrada (junto con la fecha) para reducir la complejidad de este y poder visualizar su potencial, antes de embarcarse en modelos más complejos cuyo funcionamiento no tiene por qué ser necesariamente mejor. La variable escogida es el precio de cierre ajustado para cada sesión. Yahoo Finance define este valor como el precio de cierre al ajustar la distribución de dividendos y *splits* de la acción en un periodo (Yahoo Finance, s. f.). La principal razón para realizar este ajuste en el precio de cierre se debe a la necesidad de eliminar el impacto que tienen estos eventos corporativos que podrían distorsionar la evolución del valor a lo largo del tiempo. De esta manera, podemos relacionar los precios entre sí,

como la suma del precio de la sesión anterior más las fluctuaciones del valor en ese día, ya sean positivas o negativas.

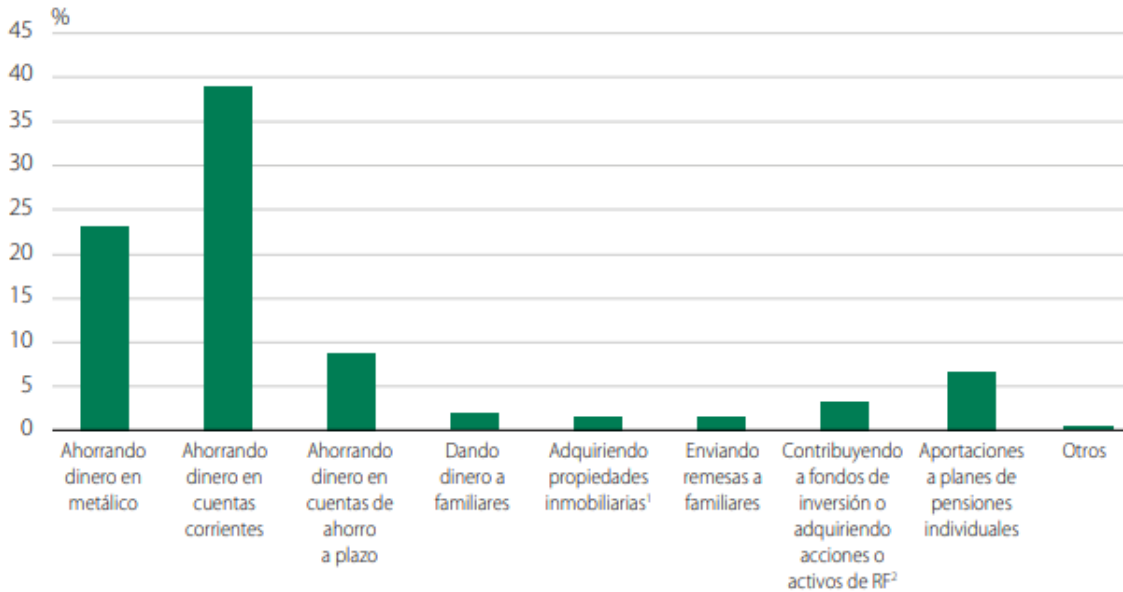
Además de los precios históricos de la acción, se podrían añadir derivados de estos, como son los indicadores técnicos. Estos proporcionan una información adicional acerca del comportamiento del valor de la acción. El uso de estos podría ayudar a capturar patrones o señales de compra o venta en el valor. Algunos de los indicadores técnicos más usados son: el indicador de fuerza relativa (RSI), el indicador de divergencia de medias móviles (MACD), las bandas de Bollinger o el oscilador estocástico, entre otros.

Otra variable de entrada a considerar podría ser la inclusión del volumen de negociación de los activos durante la sesión. Sin embargo, la agregación de esta variable sin contar con la volatilidad del mercado podría producir ruido en el sistema. Dado que las empresas escogidas son multinacionales con una muy alta capitalización en bolsa, el volumen no será tan relevante como si se escogieran *medium caps* o *small caps* (compañías de mediana o baja capitalización).

Otra de las alternativas al modelo podría ser la introducción de datos financieros de la propia compañía como pueden ser: los ingresos; la ratio precio-beneficio o PER, el beneficio por acción (BPA); la volatilidad en relación con el mercado (Beta); u otros datos relevantes. Además, otros modelos de *machine learning* más avanzados incorporan datos macroeconómicos relevantes como el PIB del país al que pertenece la compañía, la tasa de desempleo de ese país o región, los tipos de interés del banco central, la inflación o IPC o las estadísticas de producción y fabricación, entre muchos otros.

En los últimos años, el auge de las redes sociales ha permitido que se puedan realizar análisis de sentimiento para saber opiniones y predecir comportamientos de los inversores a través de sus publicaciones y respuestas en foros económicos especializados o incluso en la red social Twitter. No obstante, como se puede observar en el gráfico 3, tan sólo un 3% de los ahorradores españoles compran acciones o participaciones en fondos de inversión. Además, de ese 3% se infiere que una gran parte no posee una cuenta en la red social o es parte de los usuarios inactivos, que en España suponen el 87,1% de los usuarios de Twitter según el IX informe de los perfiles en redes sociales en España en 2023 de la consultora de marketing The Social Media Family. Con los datos sobre la mesa, pensamos que no supone una variable relevante la medición del sentimiento en las redes sociales sobre empresas cotizadas españolas.

Gráfico 3. Formas de ahorro de los españoles en los últimos en el año 2021. Fuente: ECF y CNMV



Nota. El gráfico muestra algunas de las formas de ahorro de los encuestados en España. Se destaca el ahorro en metálico y en cuentas corrientes (23,3% y 38,8% respectivamente) y el ahorro en fondos, acciones o renta fija con poco más de un 3%. Fuente: ECF y CNMV (Maté et al., 2021, fig. 14).

6.3. Tratamiento de los datos históricos

Los precios de cierre ajustados para el entrenamiento del modelo pertenecen al intervalo entre el 1 de enero de 2013 y el 1 de enero de 2023. Se ha escogido este intervalo de 10 años para tener una imagen precisa de la evolución de la cotización durante un largo periodo de tiempo. Cabe destacar, que los tipos de interés en la eurozona se mantuvieron en 0 o cercanos a este valor durante casi todo el periodo, influyendo notablemente en la renta variable. Además, la crisis económica y sanitaria producida a raíz de la Covid-19 en primavera del año 2020, supuso una fuerte caída de los precios en la gran mayoría de acciones e índices mundiales. Pese a esta caída, la mayoría de estos valores consiguió recuperar los niveles previos a la crisis a lo largo del año 2021.

Ilustración 9. Intervalo de tiempo escogido para el entrenamiento del modelo

```
start = dt.datetime(2013, 1, 1)
end = dt.datetime(2023, 1, 1)
```

Elaboración propia

Primeramente, hay que descargar los datos históricos mediante el uso de las funciones predefinidas de la librería 'yfinance'. Además de la descarga de datos y previo al tratamiento de estos para el posterior entrenamiento del modelo, se debe comprobar si existen valores NA (valores faltantes). Mediante el uso de la función 'dropna' se eliminarían aquellos valores no existentes que pueden perjudicar el entrenamiento.

En segundo lugar, se deben normalizar los datos históricos. Para ello, se escalan todos los datos a un valor entre 0 y 1. De esta forma se consiguen que todos los valores de la serie de tiempo escogida se acoten entre esos valores.

Ilustración 10. Normalización de los datos históricos para el entrenamiento

```
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data['Adj Close'].values.reshape(-1, 1))
```

Elaboración propia

Se debe aclarar que esta función para escalar los datos no tiene en cuenta las particularidades de los datos financieros, como la volatilidad y la distribución de los precios.

A continuación, es el momento de definir los días de predicción del algoritmo. Estos días, que en realidad son sesiones de mercado abierto, es el tiempo que la red neuronal mira hacia atrás para predecir el valor del próximo día de cotización futuro. En este caso, se ha optado por 60 sesiones, lo que equivaldría a cerca de unos 90 días naturales, un trimestre de cotizaciones.

6.4. Construcción de la red neuronal recurrente y ajuste de los hiperparámetros

En este punto es el momento de hacer uso de la librería 'keras' importada. Con ella, se construye el modelo a partir de una red neuronal básica secuencial que consta de tres capas LSTM (Long Short-Term Memory), tres capas de abandono o *dropout* y una capa '*Dense*'.

En la primera capa LSTM se indica que el número de unidades en esa capa son 100 ('units=100') y que se deben devolver las secuencias completas en lugar de solo el último paso de tiempo con la sentencia '*return_sequences=True*'. Además, se establece

la forma de la entrada de los datos para introducir secuencias unidimensionales con `'input_shape=(x_train.shape[1], 1)'`.

En la primera capa de `'Dropout'` y las demás de este tipo, se intenta disminuir el sobreajuste del modelo con un valor de 0,2. Este valor, indica que se desactivará el 20% de las unidades de la capa anterior de manera aleatoria durante el entrenamiento, ayudando a prevenir el sobreajuste que se mencionó anteriormente.

Las siguientes capas del modelo funcionan de manera similar a las dos primeras, aunque pueden diferir en los hiperparámetros empleados.

La última capa añadida es la capa `'Dense'` cuyo valor es de una sola unidad y se utiliza para generar la salida final del modelo.

Finalmente, el modelo se compila usando el optimizador `'adam'`, que es un algoritmo de optimización popular para redes neuronales. La función de pérdida seleccionada es `'mean_squared_error'`, que es comúnmente utilizada para problemas de regresión. El objetivo del modelo será minimizar esta función de pérdida durante el entrenamiento.

Ilustración 11. Sentencias para la construcción del modelo con capas LSTM

```
# Construcción del modelo
model = Sequential()
# Primera capa LSTM
model.add(LSTM(units=50, return_sequences=True, input_shape=(x_train.shape[1], 1)))
model.add(Dropout(0.2))
# Segunda capa LSTM
model.add(LSTM(units=50, return_sequences=True))
model.add(Dropout(0.2))
# Tercera capa LSTM
model.add(LSTM(units=50))
model.add(Dropout(0.2))
model.add(Dense(units=1))
model.compile(optimizer='adam', loss='mean_squared_error')
```

Elaboración propia

El modelo tiene multitud de variaciones atendiendo a las infinitas posibilidades de configuración mediante la modificación de los hiperparámetros. Los hiperparámetros afectan al comportamiento y al rendimiento del modelo y su ajuste corresponde al desarrollador, quién decide la combinación entre precisión y tiempo de ejecución del algoritmo. Algunos de estos parámetros son la tasa de aprendizaje, el número de capas

ocultas, el número de unidades por capa o el tamaño del lote y las épocas de entrenamiento, que se detallan a continuación.

Para el entrenamiento del modelo se emplea el método 'model.fit' con el *dataset* de entrenamiento y las etiquetas de destino, además de las épocas y el tamaño del lote.

Ilustración 12. Sentencia para el entrenamiento del modelo creado

```
model.fit(x_train, y_train, epochs=25, batch_size=32)
```

Elaboración propia

En la sentencia se indica la cantidad de veces que el modelo recorrerá todo el *dataset* de entrenamiento durante este, con la definición del número de épocas ('epochs'). También, se especifica el tamaño del lote utilizado, que son los ejemplos de entrenamiento utilizados en cada paso de actualización de los hiperparámetros del modelo. Mientras dure el proceso del entrenamiento, el modelo ajustará gradualmente los parámetros utilizando el optimizador 'adam', ya definido en este apartado. Además, el modelo tratará de minimizar la función de pérdida durante la compilación con la también mencionada función del error cuadrático medio (`mean_squared_error`).

La elección de estos parámetros requiere múltiples pruebas y medición de tiempos de compilación. Para este caso de uso, se ha probado con hiperparámetros cercanos a los definidos buscando obtener un equilibrio entre el rendimiento del modelo y un tiempo de ejecución pequeño. El ajuste mediante continuas pruebas de estos parámetros requiere de un extenso estudio que no se va a tratar en este trabajo. De todas formas, algunos de los valores que se pueden modificar para medir la precisión y el tiempo del modelo son:

- Las unidades en las capas. Se puede aumentar o disminuir este valor para ajustar la capacidad del modelo. Al aumentarlo, se puede permitir que el modelo capture patrones más complejos en los datos, pero también puede conllevar un sobreajuste.
- La tasa de *dropout*. Se puede modificar para controlar la regularización del modelo e intentar evitar el sobreajuste.
- El optimizador. Existen otros optimizadores proporcionados por la librería 'Keras' como el 'sgd' (descenso de gradiente estocástico) o el 'rmsprop'

(propagación de la raíz del error cuadrático medio). El uso de uno u otro puede aumentar la precisión de los resultados.

- La función de pérdida ('loss'). Además de la función escogida existen otras variantes como 'mean_absolute_error' (error absoluto medio) o 'binary_crossentropy' (entropía cruzada binaria) para problemas de clasificación. Dependiendo del caso de uso se puede optar por una u otra, atendiendo siempre al rendimiento del modelo.
- Las épocas. El modelo puede realizar más iteraciones para ajustar los pesos y buscar patrones si se aumenta este valor, pero podría provocar un sobreajuste al conjunto de entrenamiento si el valor es excesivo. Al reducir el número de épocas se acelera el entrenamiento, pero puede impedir que el modelo aprenda patrones y puede provocar un subajuste.
- El tamaño del lote. El uso de un tamaño del lote grande acelera el entrenamiento al procesar más muestras paralelamente, pero, puede requerir mayores recursos computacionales. Por el contrario, al disminuir su tamaño se ralentiza el entrenamiento bruscamente permitiendo al modelo generalizar mejor.

6.5. Entrenamiento y prueba de la red neuronal

Una vez se ha construido la red neuronal recurrente con tres capas LSTM y se ha entrenado con los precios históricos de la acción elegida, se procede al testeo del modelo para analizar exhaustivamente su comportamiento respecto a un conjunto de datos nuevo.

Para este nuevo proceso, se cargan las cotizaciones correspondientes a los primeros meses del año 2023. Lo que se quiere comprobar, es que el modelo funciona correctamente para un conjunto de datos que no ha visto en el entrenamiento.

Ilustración 13. Intervalo de tiempo escogido para el testeo del modelo

```
# Cargar datos de test
test_start = dt.datetime(2023, 1, 1)
test_end = dt.datetime(2023, 5, 19)
```

Elaboración propia

Después, se crea un conjunto de datos que combina el *dataset* de entrenamiento con el *dataset* de test, que se llamará 'total_dataset'. Luego, se selecciona una porción específica de los datos totales, se remodelan los datos para que sean compatibles con el modelo y se escalan para prepararlos adecuadamente como inputs en el modelo de predicción.

Ilustración 14. Preparación de los datos para la prueba del modelo

```
total_dataset = pd.concat((data['Adj Close'], test_data['Adj Close']), axis=0)
model_inputs = total_dataset[len(total_dataset) - len(test_data) - prediction_days:].values
model_inputs = model_inputs.reshape(-1, 1)
model_inputs = scaler.transform(model_inputs)
```

Elaboración propia

Por último, se preparan los datos de prueba y se almacenan en la variable 'x_test' dividiendo los datos en ventanas consecutivas de longitud 'prediction_days' (60 días) y realizando las predicciones correspondientes utilizando el modelo entrenado. Finalmente, las predicciones se transforman inversamente al escalado y se agregan a una lista para su posterior análisis y evaluación.

Ilustración 15. Realización de predicciones para la prueba del modelo

```
x_test = []
for x in range(prediction_days, len(model_inputs)):
    x_test.append(model_inputs[x - prediction_days:x, 0])
x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
predicted_prices = model.predict(x_test)
predicted_prices = scaler.inverse_transform(predicted_prices)
predictions_test.append(predicted_prices)
```

Elaboración propia

6.6. Predicción de las cotizaciones futuras

En la predicción de las cotizaciones en un futuro cercano es dónde surgen los problemas. A medida que aumenta la distancia en el tiempo de las predicciones, la precisión del modelo tiende a disminuir. Como el modelo está hecho para predecir un día, observando los 60 días anteriores, se usa ese valor del día siguiente para generar la predicción total.

Con la predicción a un día, el modelo desplaza los 60 días incluyendo la nueva predicción y pronostica el precio de los días siguientes realizando este paso en cada iteración. Como el objetivo del modelo es la predicción a 30 días, el bucle 'for' generará las predicciones día a día realizando los pasos anteriormente descritos. Para un mejor desempeño del algoritmo, se realizará varias veces este proceso y se devolverá la media de las predicciones obtenidas.

Ilustración 16. Realización de las predicciones futuras

```
real_data = [model_inputs[len(model_inputs) - prediction_days:len(model_inputs), 0]]
real_data = np.array(real_data)
real_data = np.reshape(real_data, (real_data.shape[0], real_data.shape[1], 1))

predictions_future = []
days_to_predict = 30

for _ in range(days_to_predict):
    prediction = model.predict(real_data)
    predictions_future.append(scaler.inverse_transform(prediction)[0])
    real_data = np.concatenate((real_data[:, 1:, :], prediction.reshape((1, 1, 1))), axis=1)
```

Elaboración propia

6.7. Métricas de rendimiento

Anteriormente se habló sobre la eficiencia del algoritmo para predecir en base a unos datos de prueba que no habían sido datos de entrada en el entrenamiento del modelo. Para realizar unas métricas sobre la precisión que tiene el modelo creado se usa el cálculo del error cuadrático medio (MSE) y su raíz, el RMSE.

Ilustración 17. Cálculo de las métricas de rendimiento

```
mse = mean_squared_error(actual_prices, predicted_prices)
rmse = np.sqrt(mean_squared_error(actual_prices, predicted_prices))
```

Elaboración propia

El MSE calcula el promedio de la diferencia cuadrada entre las predicciones y el objetivo. Este valor es siempre positivo y se aproxima a 0 cuánto mejor es la predicción del modelo. Se calcula mediante la fórmula:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

, donde N es el número total de instancias del conjunto de datos a medir, y_i son los valores reales y \hat{y}_i son los valores predichos.

Sin embargo, el MSE no se utiliza directamente ya que tiene la desventaja de que sus unidades de error se encuentran al cuadrado, dificultando la interpretación del resultado. Ante esta limitación, se usa generalmente el RMSE.

El RMSE es una de las medidas típicas de rendimiento para problemas de regresión. Representa los errores que comete el modelo enfatizando aquellos errores más grandes. Se calcula realizando la raíz cuadrada de la métrica MSE y sirve para escalar los errores para poder compararlos con los valores objetivo (Géron, 2019).

$$RMSE = \sqrt{MSE}$$

Además de estas métricas existen otras comunes como el error absoluto medio (MAE), que calcula el promedio del valor absoluto entre la observación y la predicción o el R-cuadrado (R²), que evalúa la relación lineal entre la variable objetivo y la variable predicha.

7. Resultados del modelo

La ejecución del programa en Python para predecir las cotizaciones futuras de una acción en bolsa consta de las partes mencionadas en el capítulo anterior: la descarga y preparación de los datos, la construcción y entrenamiento del modelo, la prueba del modelo y las predicciones futuras.

A continuación, se muestran los resultados de la ejecución del programa en sus distintas secciones para las acciones de las empresas seleccionadas.

7.1. Descarga y visualización de los precios históricos

El método 'head' permite la visualización rápida de las primeras y últimas líneas de los datos descargados para corroborar la correcta descarga de los precios históricos de las empresas seleccionadas. Después, se representa el conjunto entero de datos históricos para la visualización simple de la evolución de su cotización.

7.1.1. Datos históricos de Inditex

Inditex (ITX.MC en Yahoo Finance) ha experimentado un importante crecimiento en el precio de cierre ajustado entre el periodo comprendido entre 2013 y 2023. Como se puede observar en la ilustración 18, los valores de los demás precios como el de apertura o el de cierre son más cercanos a los del final del 2022. Los repartos de dividendos pueden ser un motivo que explique el fuerte ajuste del cierre en los valores del inicio de 2013.

Ilustración 18. Cabecera de los datos históricos de Inditex

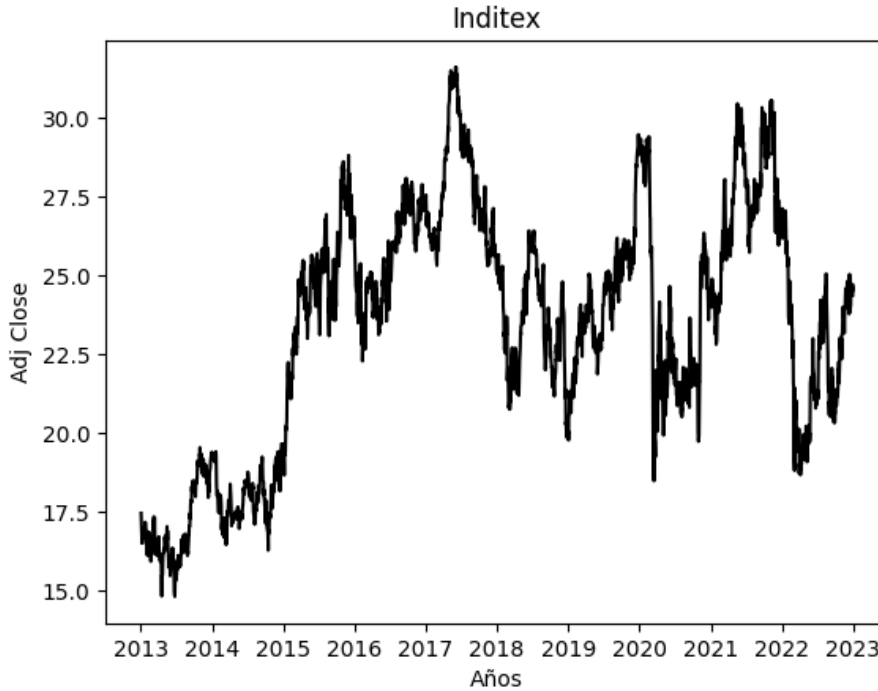
```
In [7]: print(yf.download('ITX.MC', start, end).head)
[*****100%*****] 1 of 1 completed
<bound method NDFrame.head of
Date
2013-01-02 21.959999 22.139999 21.629999 21.860001 17.464930 6402505
2013-01-03 21.900000 21.900000 21.420000 21.500000 17.177307 4055610
2013-01-04 21.520000 21.520000 21.320000 21.320000 17.033497 8884885
2013-01-07 21.299999 21.400000 20.629999 20.660000 16.506193 6516385
2013-01-08 20.670000 21.150000 20.600000 21.120001 16.873711 5971315
...
...
...
2022-12-23 25.059999 25.240000 24.850000 24.940001 24.465254 1405048
2022-12-27 25.010000 25.290001 25.010000 25.110001 24.632019 1028113
2022-12-28 25.080000 25.280001 24.959999 24.959999 24.484873 1108435
2022-12-29 24.830000 25.240000 24.750000 25.200001 24.720306 968642
2022-12-30 24.969999 25.129999 24.709999 24.850000 24.376968 1471839

[2558 rows x 6 columns]>
```

Elaboración propia

La representación gráfica de la evolución de la cotización de Inditex se puede observar en la gráfica 4:

Gráfico 4. Evolución del precio de cierre ajustado de las acciones de Inditex (2013-2023)



Elaboración propia

7.1.2. Datos históricos de Iberdrola

Iberdrola (IBE.MC en Yahoo Finance) ha multiplicado el valor de su acción por más de cuatro veces en la última década, como se puede observar en la ilustración 19.

Ilustración 19. Cabecera de los datos históricos de Iberdrola

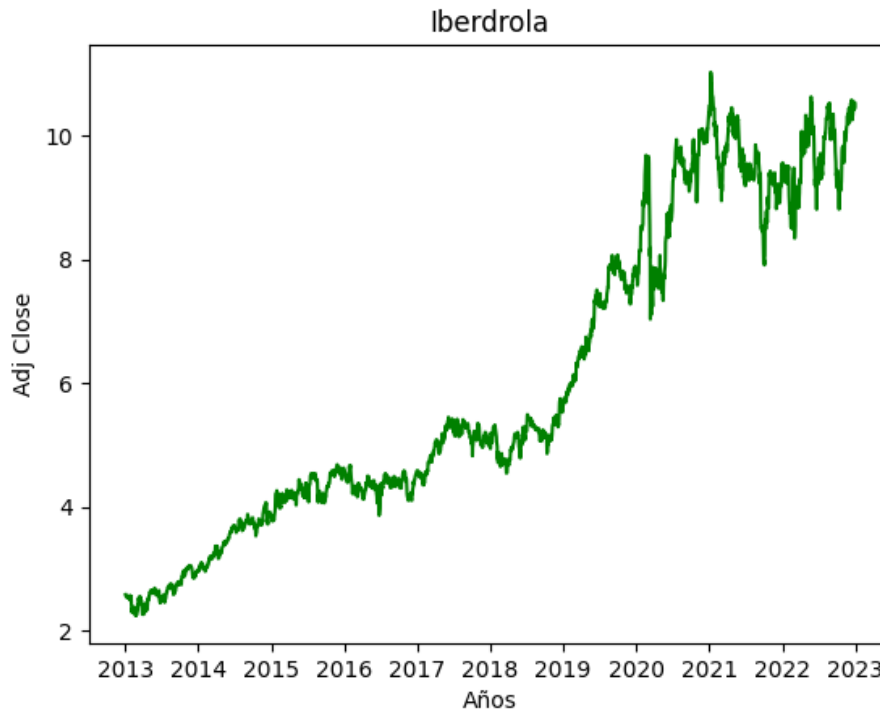
```
In [24]: print(yf.download('IBE.MC', start, end).head)
[*****100%*****] 1 of 1 completed
<bound method NDFrame.head of
Date      Open      High      Low      Close      Adj Close      Volume
2013-01-02  4.270    4.291    4.204    4.291    2.583587    62406514
2013-01-03  4.120    4.152    4.056    4.115    2.563034    40042894
2013-01-04  4.100    4.126    4.081    4.110    2.559919    104449702
2013-01-07  4.091    4.128    4.073    4.094    2.549953    33644522
2013-01-08  4.076    4.105    4.060    4.074    2.537497    38010423
...
2022-12-23  10.960    10.960    10.815    10.880    10.408959    5950541
2022-12-27  10.955    10.995    10.900    10.900    10.428093    7927977
2022-12-28  10.940    11.035    10.910    10.925    10.452012    7805532
2022-12-29  10.960    11.065    10.895    11.015    10.538115    8776526
2022-12-30  11.000    11.015    10.930    10.930    10.456796    11481718

[2558 rows x 6 columns]>
```

Elaboración propia

La representación gráfica de la evolución de la cotización de Iberdrola se puede observar en la gráfica 5. En ella se aprecia fácilmente el fuerte crecimiento cosechado por la empresa en la última década.

Gráfico 5. Evolución del precio de cierre ajustado de las acciones de Iberdrola (2013-2023)



Elaboración propia

7.1.3. Datos históricos del Banco Santander

El Banco Santander (SAN.MC en Yahoo Finance) ha perdido notablemente valor en los últimos años. Como se observa en la ilustración 20, sus valores de cierre entre los años 2023 y 2013 difieren en casi 3 euros, suponiendo una caída de más del 50%, si atendemos exclusivamente a este precio. Pero, como sucedía con el caso de Inditex, los valores en el precio ajustado también experimentaron una notable caída, aunque mucho menor en comparación con el valor de cierre (*Close*).

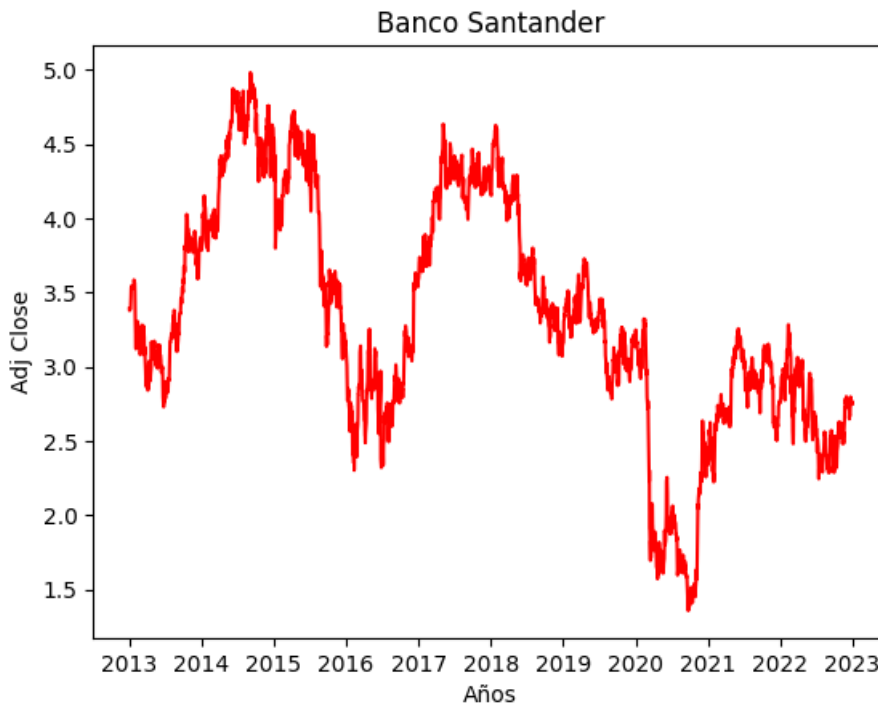
Ilustración 20. Cabecera de los datos históricos de Santander

```
In [6]: print(yf.download('SAN.MC', start, end).head)
[*****100%*****] 1 of 1 completed
<bound method NDFrame.head of
Date
2013-01-02  5.843962  5.965555  5.787408  5.965555  3.389719  84000802
2013-01-03  5.909000  5.966497  5.896747  5.944818  3.377936  31346019
2013-01-04  5.947646  6.003258  5.925966  5.966497  3.390253  83958757
2013-01-07  5.993832  6.041903  5.948588  5.958957  3.385969  99865587
2013-01-08  5.942933  6.019281  5.906172  5.975923  3.395610  204031624
...
2022-12-23  2.817500  2.835500  2.782500  2.795000  2.744514  21811089
2022-12-27  2.810000  2.836000  2.802000  2.815000  2.764152  21064298
2022-12-28  2.806000  2.819000  2.792000  2.798000  2.747459  18502658
2022-12-29  2.781000  2.822500  2.769000  2.818000  2.767098  25760397
2022-12-30  2.801000  2.816500  2.774000  2.802500  2.751878  40266237
[2558 rows x 6 columns]>
```

Elaboración propia

La representación gráfica de la evolución de la cotización del Banco Santander se puede observar en la gráfica 6.

Gráfico 6. Evolución del precio de cierre ajustado de las acciones de Santander (2013-2023)



Elaboración propia

7.1.4. Datos históricos del BBVA

El Banco Bilbao Vizcaya Argentaria (BBVA.MC en Yahoo Finance) ha crecido ligeramente en el periodo estudiado como se ve en la cabecera de los datos de la

ilustración 21. A pesar de una subida mayor de un 20% (aunque en 10 años), su cotización se ha mostrado notoriamente volátil si atendemos a la observación de su cotización por periodos trimestrales o anuales, como se aprecia en el gráfico 7. Con constantes subidas y bajadas el valor ha conseguido crecer entre el 2013 y el 2023.

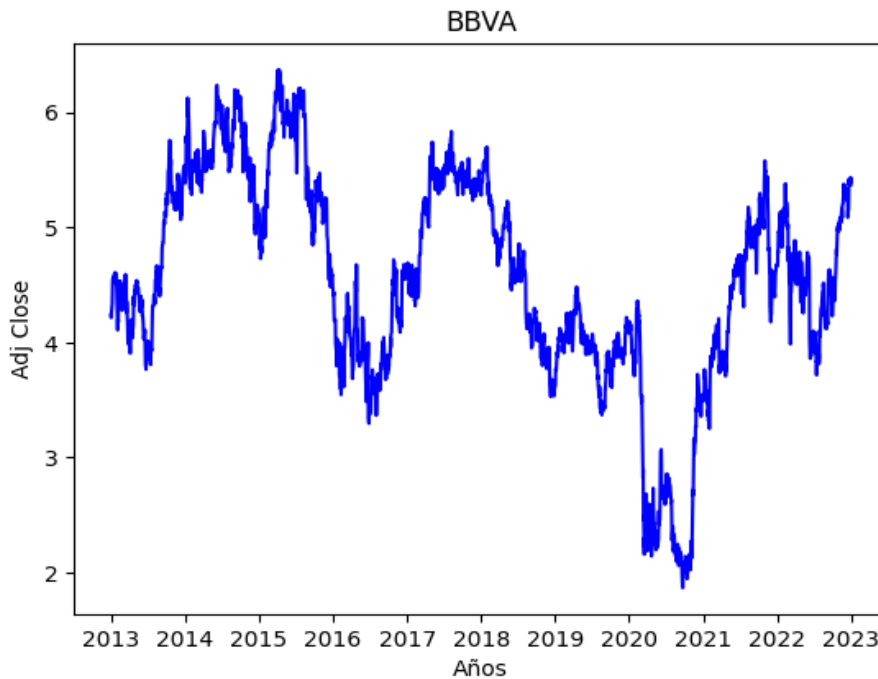
Ilustración 21. Cabecera de los datos históricos de BBVA

```
In [36]: print(yf.download('BBVA.MC', start, end).head)
[*****100%*****] 1 of 1 completed
<bound method NDFrame.head of
Date
2013-01-02  7.096  7.287  7.045  7.287  4.238847  193615334
2013-01-03  7.215  7.290  7.171  7.245  4.214416  404850032
2013-01-04  7.240  7.319  7.210  7.318  4.256879  205290519
2013-01-07  7.333  7.419  7.333  7.406  4.308069  67371537
2013-01-08  7.363  7.497  7.330  7.451  4.334245  56140482
...
...
...
...
2022-12-23  5.650  5.693  5.622  5.666  5.398695  6924974
2022-12-27  5.667  5.690  5.629  5.649  5.382497  6957865
2022-12-28  5.656  5.687  5.636  5.655  5.388214  5817544
2022-12-29  5.628  5.720  5.610  5.702  5.432997  8439852
2022-12-30  5.683  5.688  5.630  5.634  5.368205  7801055
[2558 rows x 6 columns]>
```

Elaboración propia

La representación gráfica de la evolución de la cotización del Banco BBVA se puede observar en la siguiente gráfica:

Gráfico 7. Evolución del precio de cierre ajustado de las acciones de BBVA (2013-2023)



Elaboración propia

7.1.5. Datos históricos de Amadeus

Amadeus IT Group (AMS.MC en Yahoo Finance) ha conseguido multiplicar su valor por 3 en la década estudiada. Su valor creció fuertemente desde el 2013 hasta casi la crisis de la covid-19 y, tras superarla, ha conseguido recuperar la senda del crecimiento, aunque lejos de los máximos de 2019.

Ilustración 22. Cabecera de los datos históricos de Amadeus

```
In [50]: print(yf.download('AMS.MC', start, end).head)

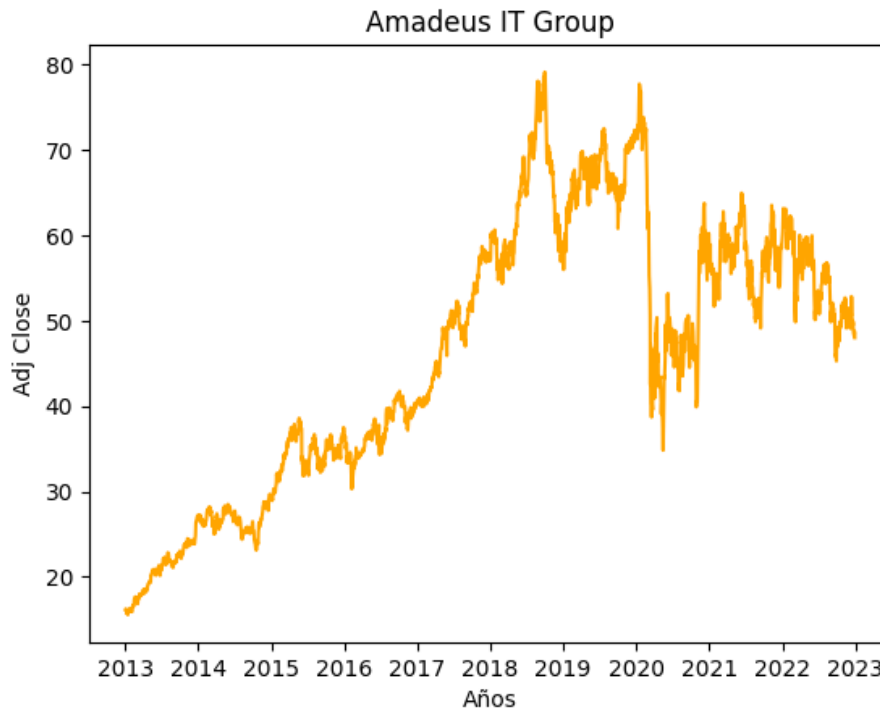
[*****100%*****] 1 of 1 completed
<bound method NDFrame.head of
Date
2013-01-02 19.250000 19.250000 18.754999 18.875000 16.145170 1790073
2013-01-03 18.780001 19.000000 18.660000 19.000000 16.252090 9703172
2013-01-04 18.980000 19.000000 18.799999 19.000000 16.252090 1113283
2013-01-07 18.875000 18.930000 18.504999 18.735001 16.025419 7799090
2013-01-08 18.549999 18.695000 18.344999 18.370001 15.713207 1668781
...
...
2022-12-23 49.639999 49.650002 48.990002 49.220001 48.669140 428692
2022-12-27 49.369999 49.540001 48.360001 48.750000 48.204399 391855
2022-12-28 48.950001 49.459999 48.720001 49.259998 48.708691 424718
2022-12-29 49.209999 49.360001 48.660000 49.299999 48.748245 317133
2022-12-30 49.099998 49.099998 48.439999 48.549999 48.006638 936749

[2558 rows x 6 columns]>
```

Elaboración propia

La representación gráfica de la evolución de la cotización de Amadeus se puede observar en el gráfico 8. En la representación se observa de manera clara cómo los años de la crisis de la covid-19 frenaron su notable crecimiento hasta la fecha.

Gráfico 8. Evolución del precio de cierre ajustado de las acciones de Amadeus (2013-2023)



Elaboración propia

7.2. Construcción y entrenamiento del modelo

El modelo 'Sequential' se inicializa y comienza a construirse contando con 50.851 parámetros en el conjunto de las 3 capas añadidas más la capa 'Dense'.

Gráfico 9. Visualización de la construcción del modelo en tiempo de ejecución

```

Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
lstm (LSTM)                  (None, 60, 50)           10400
dropout (Dropout)           (None, 60, 50)           0
lstm_1 (LSTM)                (None, 60, 50)           20200
dropout_1 (Dropout)         (None, 60, 50)           0
lstm_2 (LSTM)                (None, 50)                20200
dropout_2 (Dropout)         (None, 50)                0
dense (Dense)                (None, 1)                 51
-----
Total params: 50851 (198.64 KB)
Trainable params: 50851 (198.64 KB)
Non-trainable params: 0 (0.00 Byte)
    
```

Elaboración propia

El modelo se construye con las capas determinadas anteriormente. En la ejecución, se muestra el progreso del entrenamiento como en la siguiente figura.

Gráfico 10. Visualización del entrenamiento del modelo creado, en tiempo de ejecución

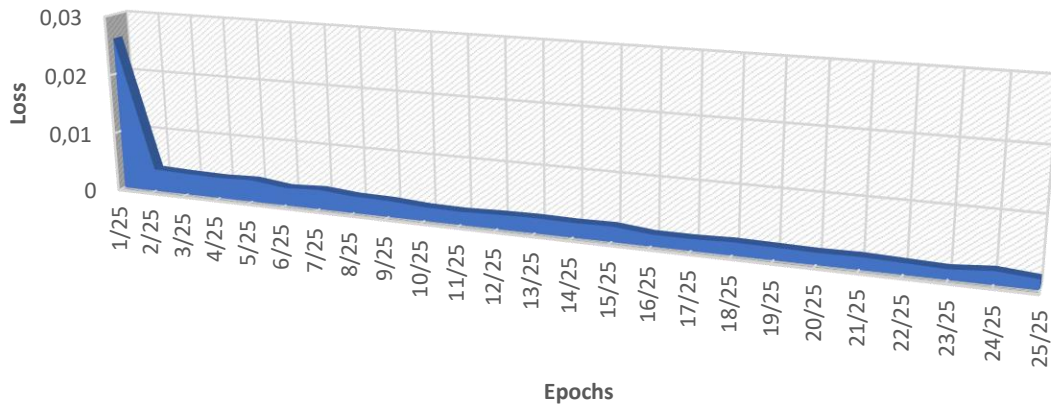
```
Epoch 1/25
79/79 [=====] - 9s 49ms/step - loss: 0.0155
Epoch 2/25
79/79 [=====] - 4s 48ms/step - loss: 0.0035
Epoch 3/25
79/79 [=====] - 4s 50ms/step - loss: 0.0031
Epoch 4/25
79/79 [=====] - 4s 48ms/step - loss: 0.0037
Epoch 5/25
79/79 [=====] - 4s 49ms/step - loss: 0.0034
```

Elaboración propia

La información relevante que puede observarse mientras dura el entrenamiento son:

- Las épocas ('epochs'): Se establece un contador del 1 al número máximo elegido (25) dónde va recorriendo el conjunto de datos en cada época.
- El tamaño de las muestras: Se puede visualizar como aparece el número 79/79 que representa aproximadamente la división entre el conjunto de datos introducidos (2558 filas) y el tamaño del lote seleccionado (32). En cada época, los datos de entrenamiento se dividen en lotes y se procesan en paralelo.
- El tiempo de recorrido. Muestra el tiempo promedio que tarda en completar el lote actual en cada época.
- La pérdida. Muestra el valor de pérdida en cada época. Con ello se visualiza la capacidad de aprendizaje del modelo según avanza el entrenamiento. A medida que avanza el entrenamiento, en cada época, la pérdida debe decrecer como consecuencia del ajuste.

Gráfico 11. Evolución de la pérdida a medida que avanza el entrenamiento



Elaboración propia

Este entrenamiento se realiza el número de veces definido por la variable incorporada *iterations*, que en este caso es 10.

Gráfico 12. Iteraciones del entrenamiento

```
iterations = 10

for i in range(iterations):
    model.fit(x_train, y_train, epochs=25, batch_size=32)
```

Elaboración propia

Mediante la realización de diversas pruebas y con las métricas de MSE y RMSE que se mostrarán posteriormente, se determinó que el modelo disminuye el error a medida que se reentrena, hasta volverse prácticamente constante. Se ha observado que, en la primera y la segunda iteración, los errores son muy superiores a las siguientes iteraciones y, por ello, se decide entrenar 10 veces el modelo para obtener mejores resultados.

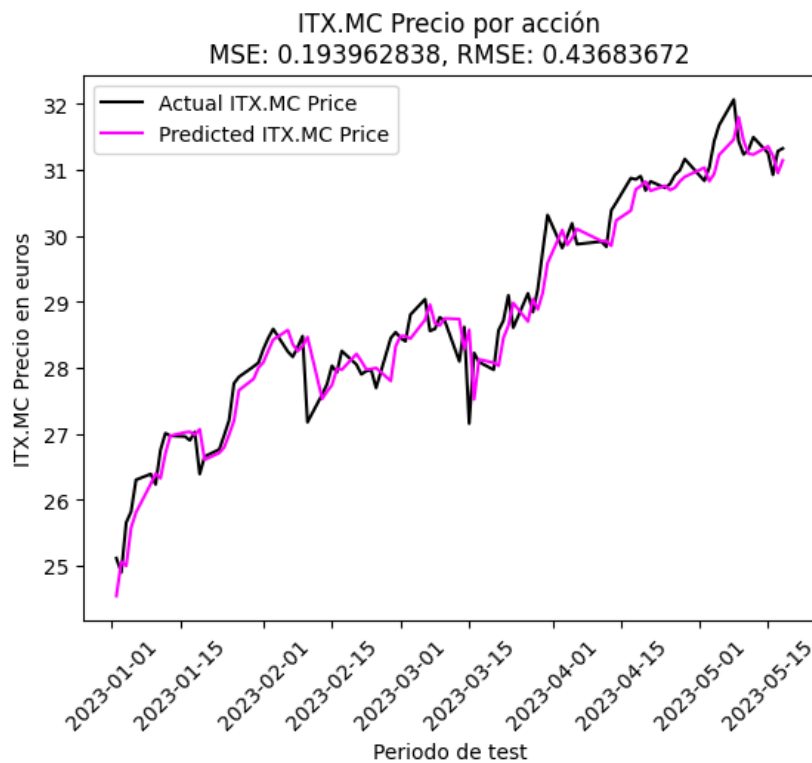
7.3. Test y evaluación del rendimiento del modelo

A continuación, el modelo ya entrenado carga los datos de test y trata de predecir a partir de los nuevos datos no conocidos. Como la prueba se encuentra dentro del mismo bucle 'for' que el ajuste del modelo, la prueba y medida del error también se realiza 10 veces. Lo que muestran las siguientes gráficas de color magenta son los promedios de las predicciones para el conjunto de datos de prueba. Además, se muestra el promedio del MSE y el RMSE para cada test.

7.3.1. Test y predicción sobre Inditex

En la gráfica 13 se puede observar como la curva construida a partir de los valores predichos se asemeja a los precios reales de cotización. Por ello, se puede afirmar que, para este valor en este periodo, el modelo funciona de forma óptima. Además, el valor RMSE en el promedio de las 10 iteraciones del test se sitúa en torno a los 0,43€. Este valor significa que la desviación promedio entre los precios predichos por el modelo y los precios reales es de apenas 43 céntimos de euro. Teniendo en cuenta que las acciones oscilan entre los 25 y los 32 euros, se puede concluir que el error es mínimo.

Gráfico 13. Test del modelo con precios históricos de Inditex

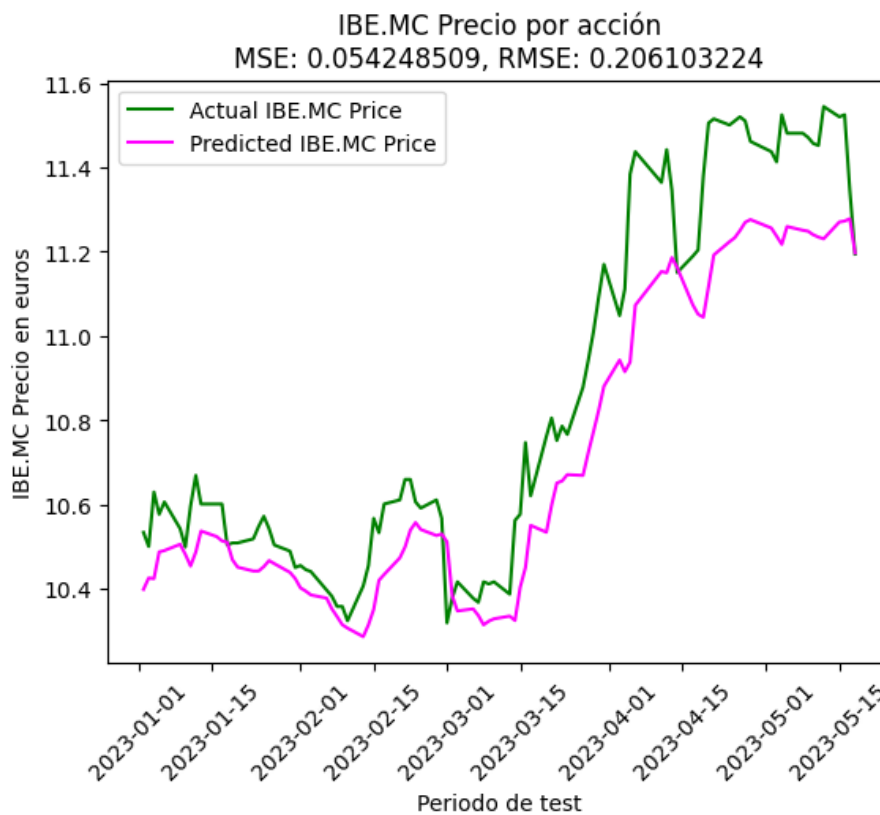


Elaboración propia

7.3.2. Test y predicción sobre Iberdrola

En la gráfica 14 se puede observar cómo los valores consiguen captar la tendencia de los datos del test, aunque funcionan mejor en la primera mitad del periodo de prueba y, en la segunda mitad aumenta su desviación respecto a los datos reales. En esta gráfica también se puede observar un valor RMSE pequeño, en comparación con el precio en euros que se mueve alrededor de los 11€ por acción. La desviación (positiva o negativa) en promedio es de unos 20 céntimos.

Gráfico 14. Test del modelo con precios históricos de Iberdrola

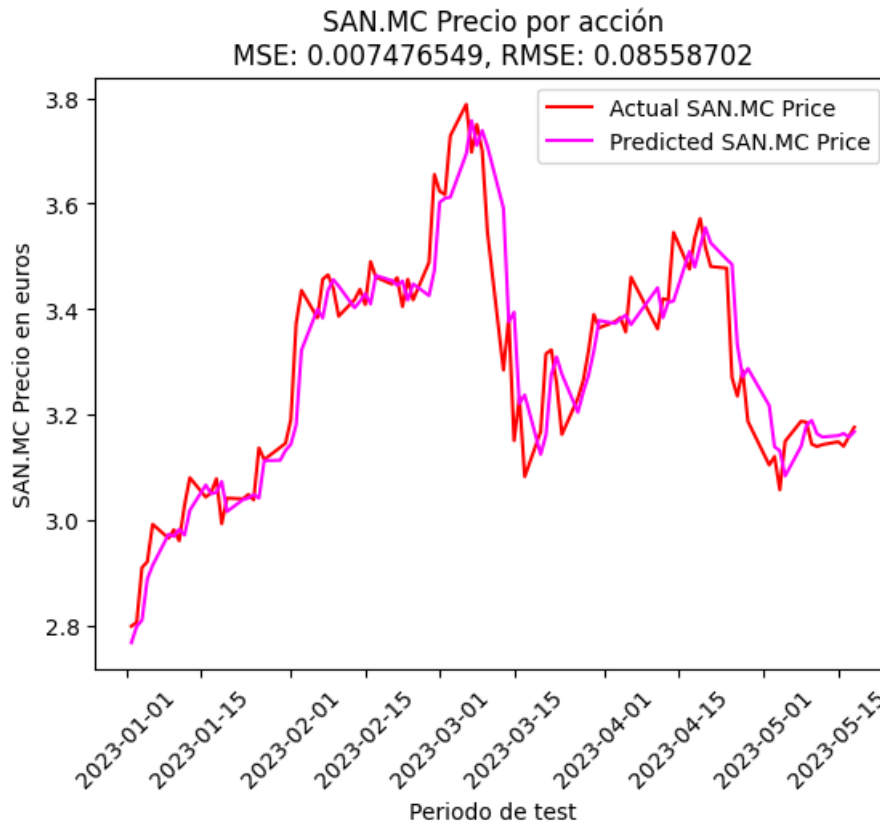


Elaboración propia

7.3.3. Test y predicción sobre Santander

En la gráfica 15 se visualiza rápidamente como las curvas entre los precios predichos y los precios reales para el periodo son realmente parecidas, e, incluso se solapan en ciertas fechas. La comprobación de que el modelo es óptimo para el Banco Santander en este periodo lo proporciona el valor de RMSE. Este valor de 0,086 indica que la desviación promedio durante la prueba es aproximadamente de unos 8 o 9 céntimos de euro. Las acciones del banco oscilan entre los 2,8 y 3,8 euros por lo que el error es relativamente mínimo.

Gráfico 15. Test del modelo con precios históricos de Santander

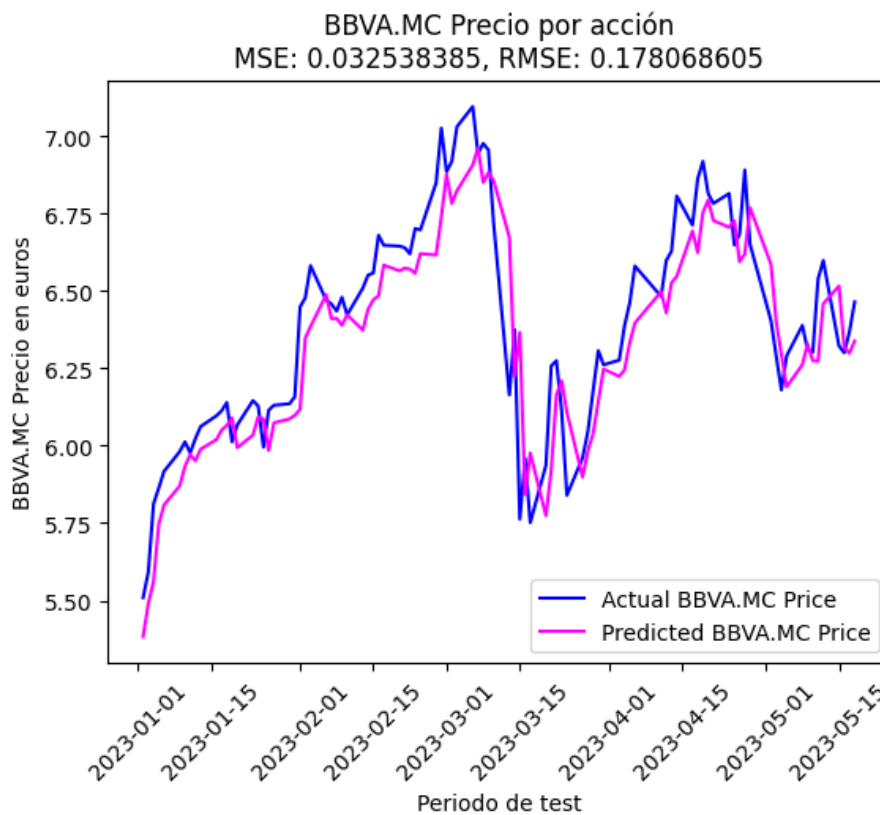


Elaboración propia

7.3.4. Test y predicción sobre BBVA

En la gráfica 16 se observa la similitud entre los precios predichos por el modelo y los precios reales de cotización para el banco. El modelo logra capturar correctamente las tendencias y se demuestra al ver que el RMSE del test, es de 0,178 en promedio. Las acciones del BBVA cotizan entre los 5,5 y 7 euros durante el periodo de la prueba y la desviación promedio durante esta es menor de 18 céntimos, siendo esta, un gran resultado.

Gráfico 16. Test del modelo con precios históricos de BBVA

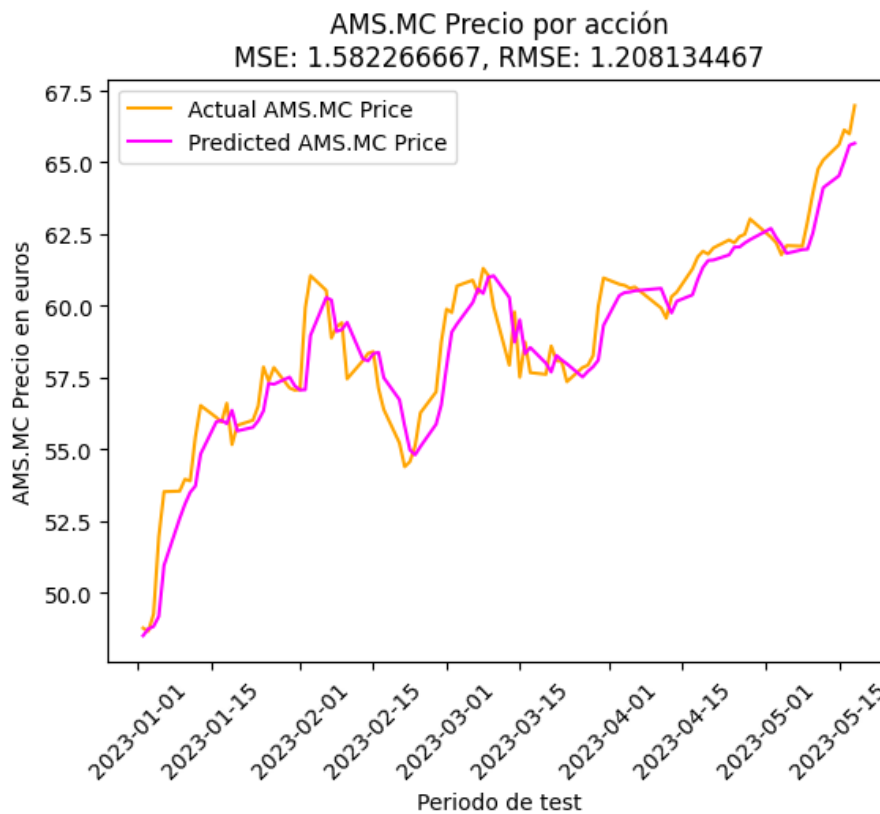


Elaboración propia

7.3.5. Test y predicción sobre Amadeus

En la gráfica se puede visualizar como ambas curvas tienen una gran semejanza a lo largo del periodo. De nuevo, el RMSE en promedio para las 10 iteraciones es de 1,21. Este valor es sumamente bajo teniendo en cuenta que las acciones cotizan entre los 50 y los 67 euros y su desviación positiva o negativa apenas alcanza los 1,21 euros.

Gráfico 17. Test del modelo con precios históricos de Amadeus



Elaboración propia

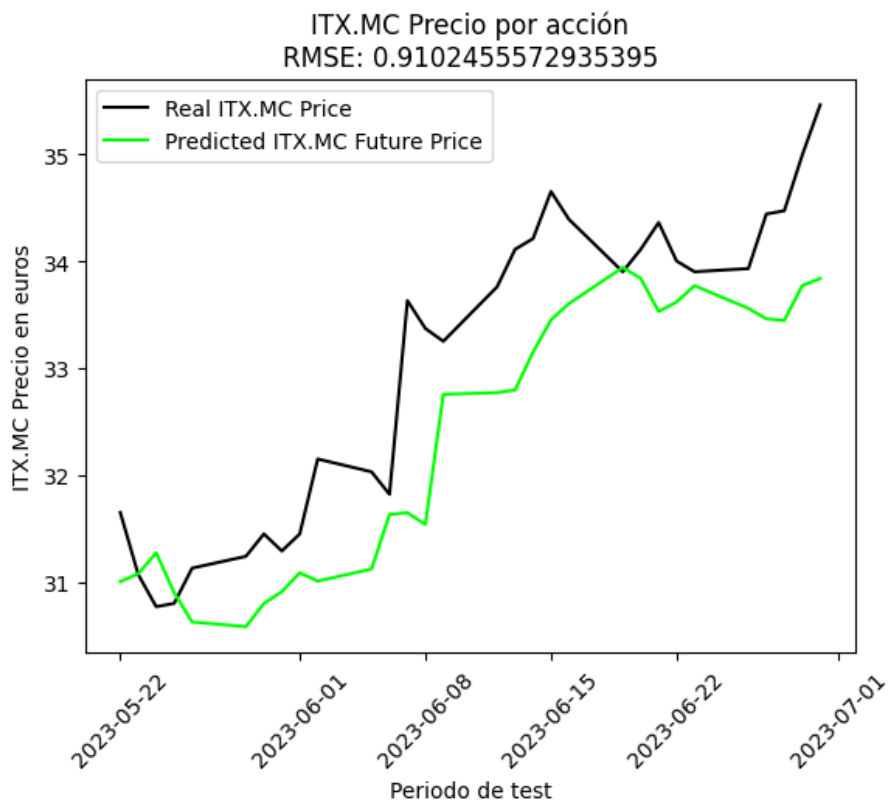
7.4. Predicciones futuras

En esta ocasión, el modelo no recibirá ningún dato e intentará predecir los valores futuros de cada acción para el día siguiente con un horizonte temporal de 30 días. De nuevo, se debe recordar que los 30 días no son días naturales, si no sesiones de cotización con el mercado abierto.

7.4.1. Predicción futura de Inditex

En la gráfica 18 se puede observar como el modelo capta la tendencia alcista de las acciones de Inditex. Pese a que el precio pronosticado se encuentra unos céntimos de euro por debajo en gran parte de la predicción, la dirección marcada por las predicciones puede servir para indicar hacia donde evolucionará la cotización. Por último, el RMSE para este periodo es bastante pequeño, sobre los 0,91 euros.

Gráfico 18. Predicciones futuras de los precios de las acciones de Inditex

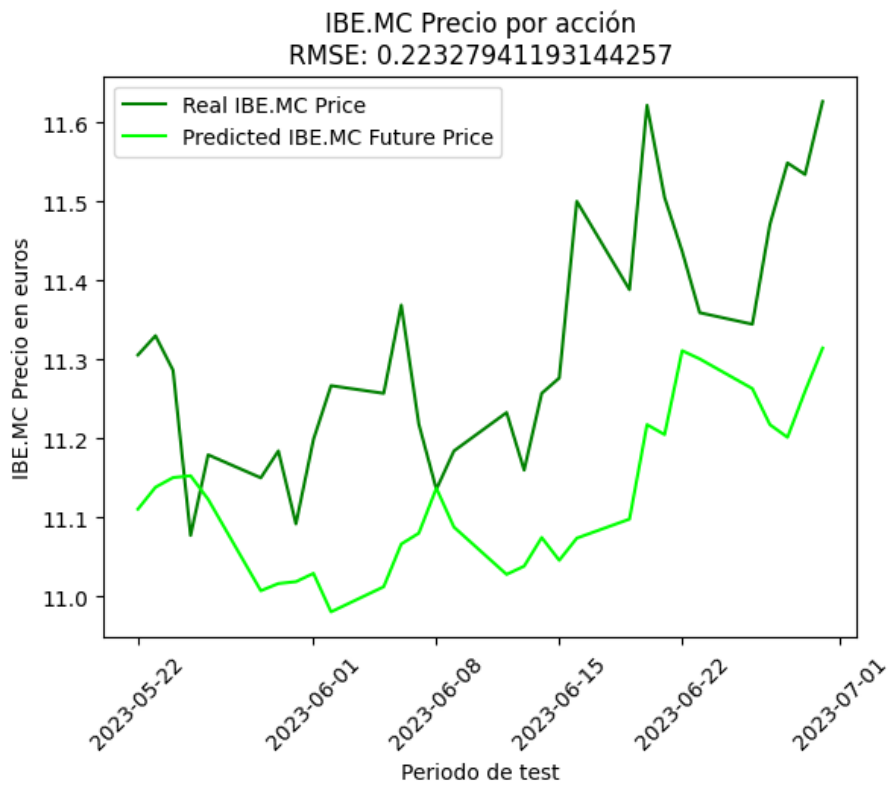


Elaboración propia

7.4.2. Predicción futura de Iberdrola

En la gráfica 19 se visualiza una curva de predicción futura que, si bien guarda relación con los precios reales que se han producido, a medida que aumenta el tiempo, la precisión disminuye y sus precios al final de los 30 días se encuentran a una distancia mayor de los 30 céntimos por acción. En cuanto al RMSE producido, se sitúa por encima de los 22 céntimos de euro.

Gráfico 19. Predicciones futuras de los precios de las acciones de Iberdrola

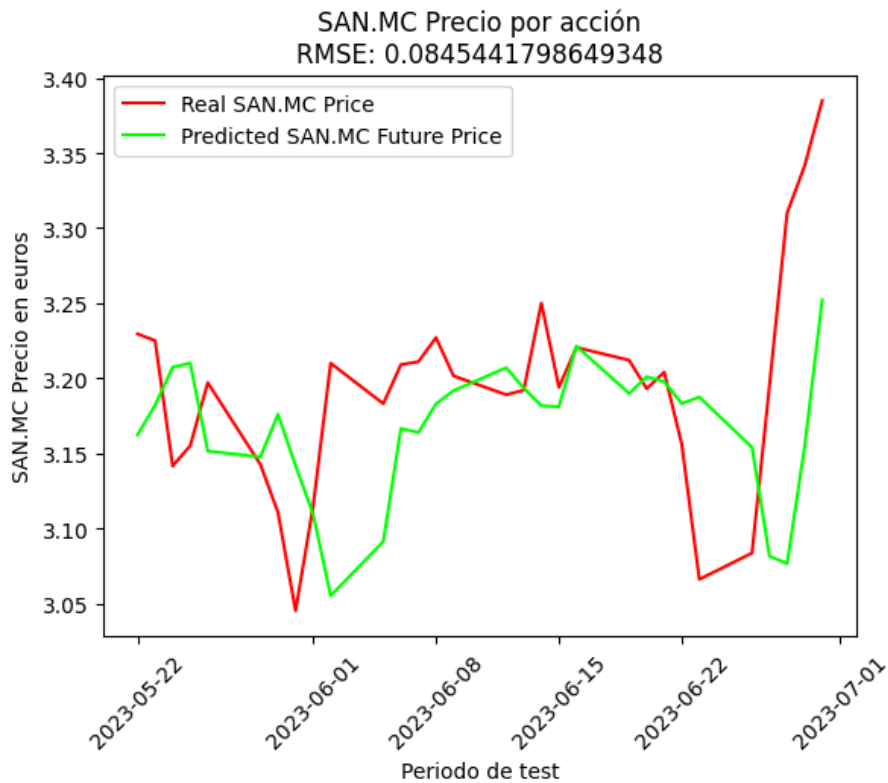


Elaboración propia

7.4.3. Predicción futura de Santander

En la gráfica 20 se observa que las diferencias entre los precios reales y los predichos en ese intervalo de tiempo son notables. En algunos días logra captar las tendencias, pero se sitúa lejos de los valores máximos o mínimos alcanzados en esas sesiones. Como punto positivo, el promedio del precio de los valores futuros predichos es de 3,167 euros y el promedio de los precios reales en ese periodo es de 3,193 euros, siendo muy similares. Por otra parte, el RMSE es cercano a los 8 céntimos de desviación promedio.

Gráfico 20. Predicciones futuras de los precios de las acciones de Santander

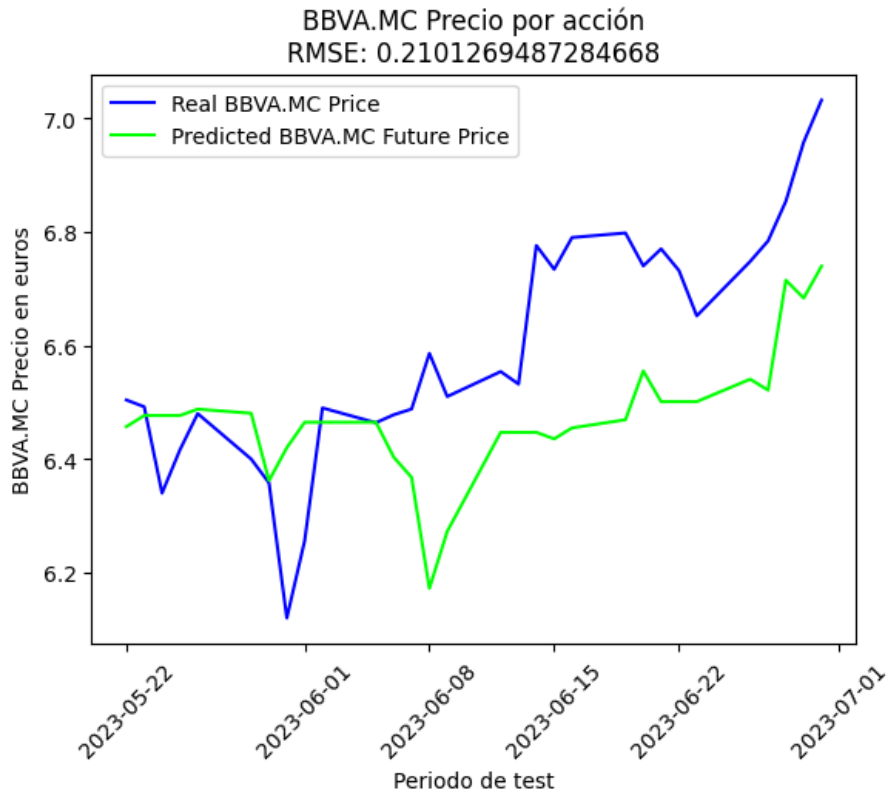


Elaboración propia

7.4.4. Predicción futura de BBVA

En la gráfica 21 se puede observar que las curvas, si bien son muy similares, los precios predichos están claramente desplazados hacia la derecha. Esto quiere decir, que la predicción futura funciona con retraso respecto al mercado abierto y, por ello, se concluye que la predicción es realmente mala en la primera mitad de la curva. En la segunda mitad de la curva, la predicción mejora ligeramente, aunque los precios predichos continúan por debajo de los reales. El RMSE se sitúa en torno a los 21 céntimos por acción.

Gráfico 21. Predicciones futuras de los precios de las acciones de BBVA

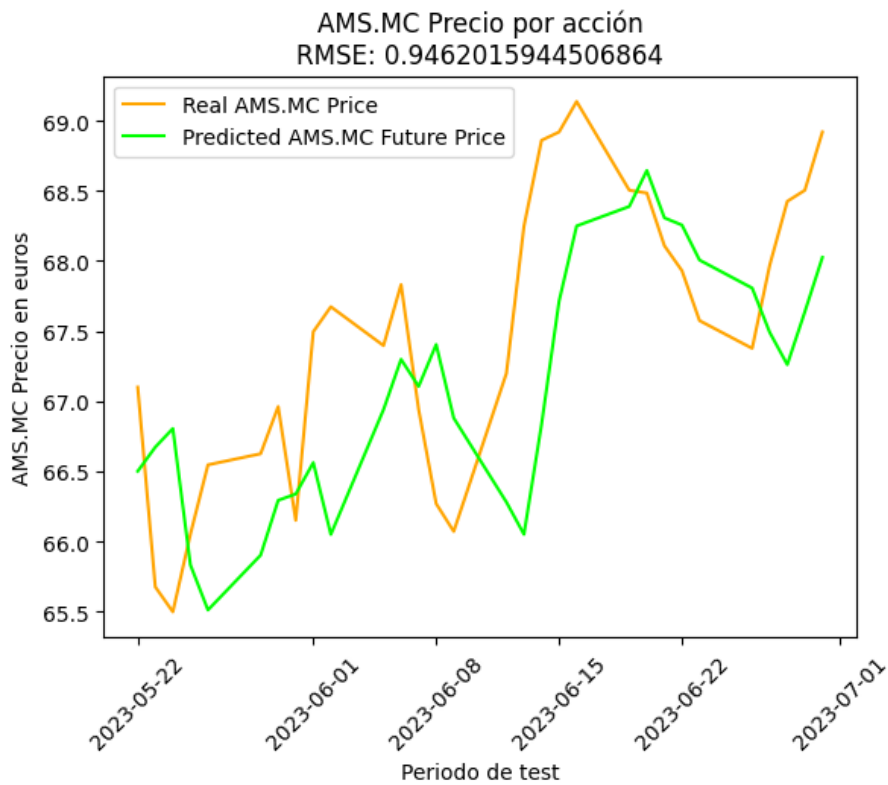


Elaboración propia

7.4.5. Predicción futura de Amadeus

En la gráfica 22 se visualiza la diferencia entre ambas curvas durante todo el periodo analizado. De forma similar a lo que sucedía con las predicciones de BBVA, para Amadeus también el modelo está prediciendo los cambios en la cotización con retraso. Finalmente, el RMSE es bastante pequeño, de 0,94 euros, teniendo en cuenta que las acciones se mueven entre los 65 y los 70 euros, siendo un error aceptable.

Gráfico 22. Predicciones futuras de los precios de las acciones de Amadeus IT Group



Elaboración propia

8. Estrategia de inversión

El objetivo de la creación del modelo es averiguar si los datos históricos de una acción pueden ayudar a predecir el comportamiento futuro de esta. Sin embargo, aunque los errores sean relativamente pequeños, el programa no es lo suficientemente completo como para ser un indicador de dónde invertir. Para ello, habría que incorporar múltiples variables y realizar pruebas y métricas suficientes para comprobar su utilidad.

A pesar de estas limitaciones, se estudian las posibilidades de inversión con los valores predichos para analizar su comportamiento en la Bolsa de España usando este modelo, en el periodo posterior a los datos de prueba, es decir, a finales de mayo de 2023.

8.1. Análisis del mercado

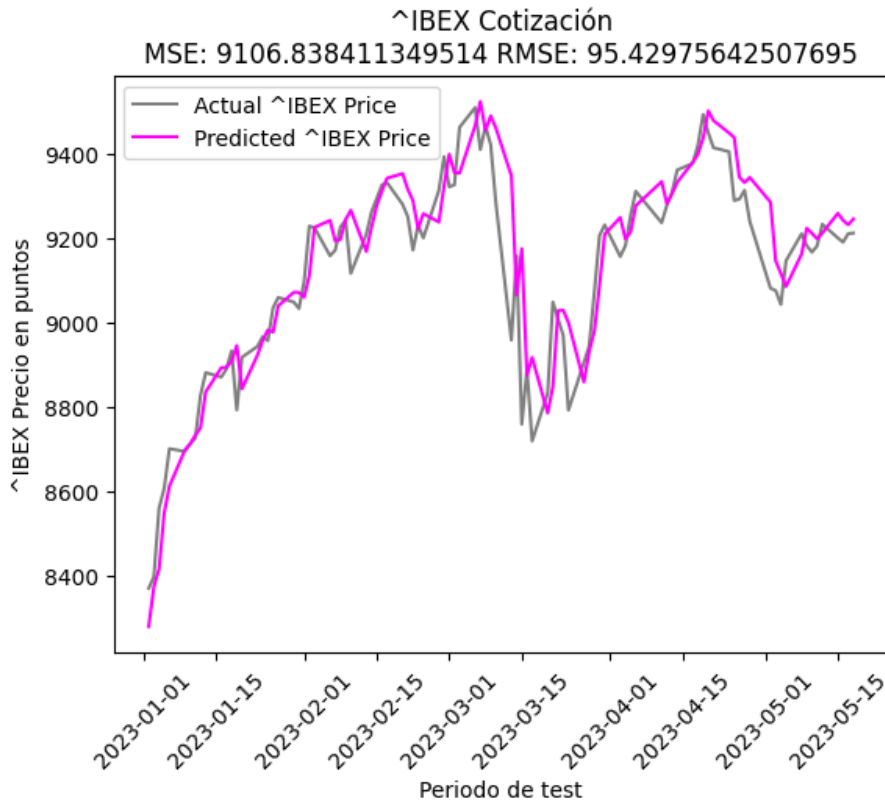
Los valores analizados en el anterior capítulo corresponden a acciones de empresas españolas cotizadas en el índice IBEX 35, índice referencia de la bolsa de España formado por las 35 empresas con mayor capitalización, liquidez y volumen de negociación.

Debido a la naturaleza de los mercados financieros, no se puede realizar un estudio de la acción de una empresa sin tener en cuenta el mercado o mercados donde cotiza. Por ello, también se ha decidido incluir el estudio del IBEX 35 en el modelo para su prueba y predicción futura. A diferencia de las acciones, el valor del IBEX se encuentra en puntos calculados en base a la capitalización de las empresas que lo componen y ajustado por capital flotante.

Por la naturaleza extraordinaria del valor y por su difícil comparación en valor respecto a las acciones, se ha decidido incluir en este apartado, pues su estudio es de gran valor para la decisión sobre una inversión en una de las empresas estudiadas.

En la gráfica se puede observar como la curva construida a partir de los valores predichos se asemeja a los precios reales de cotización del índice. Debido a la alta semejanza entre los valores reales y las predicciones, se puede afirmar que, para este valor en este periodo, el modelo funciona de forma óptima. También es importante destacar que el valor RMSE en el promedio de las 10 iteraciones del test se sitúa en torno a los 95 puntos. Teniendo en cuenta que el índice se mueve alrededor de los 9.000 puntos, se puede afirmar que es un error aceptable para la predicción.

Gráfico 23. Test del modelo con precios históricos del Ibex 35

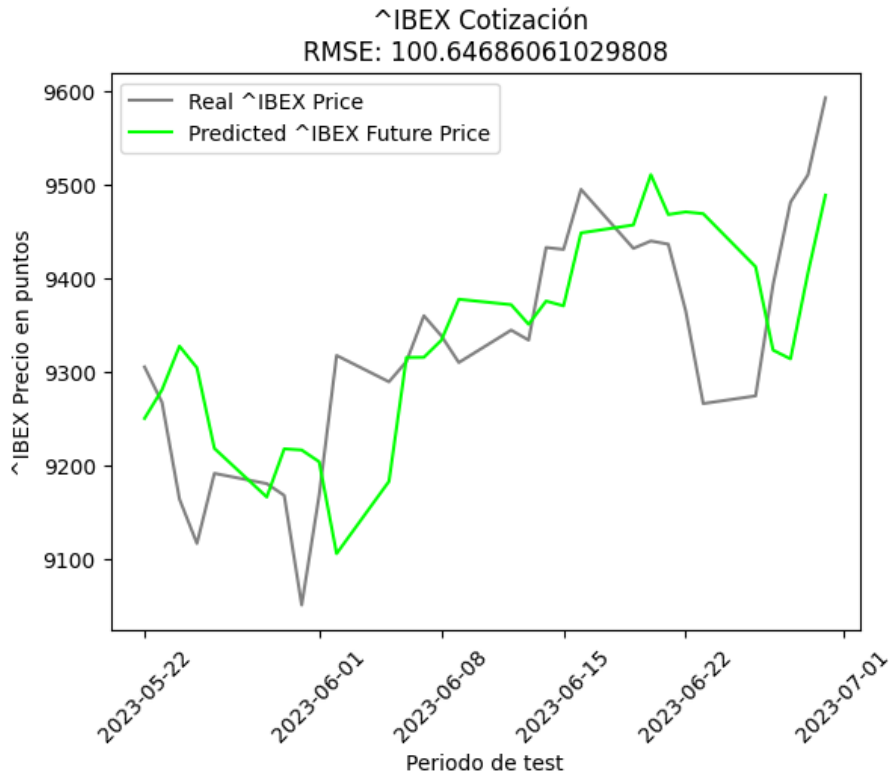


Elaboración propia

Una vez entrenado el modelo para el IBEX, se procederá, análogamente al proceso efectuado con las acciones, es decir, a la predicción de los valores futuros para cada sesión durante un periodo corto de tiempo.

En la gráfica 24 se puede visualizar la amplia diferencia en las predicciones durante la primera mitad de esta. Aunque en la segunda mitad, mejoran ligeramente, el desplazamiento producido hacia la derecha indica el retraso en la predicción del valor. Además, los cálculos devuelven un RMSE superior a los 100 puntos.

Gráfico 24. Predicciones futuras de los precios del índice Ibex 35



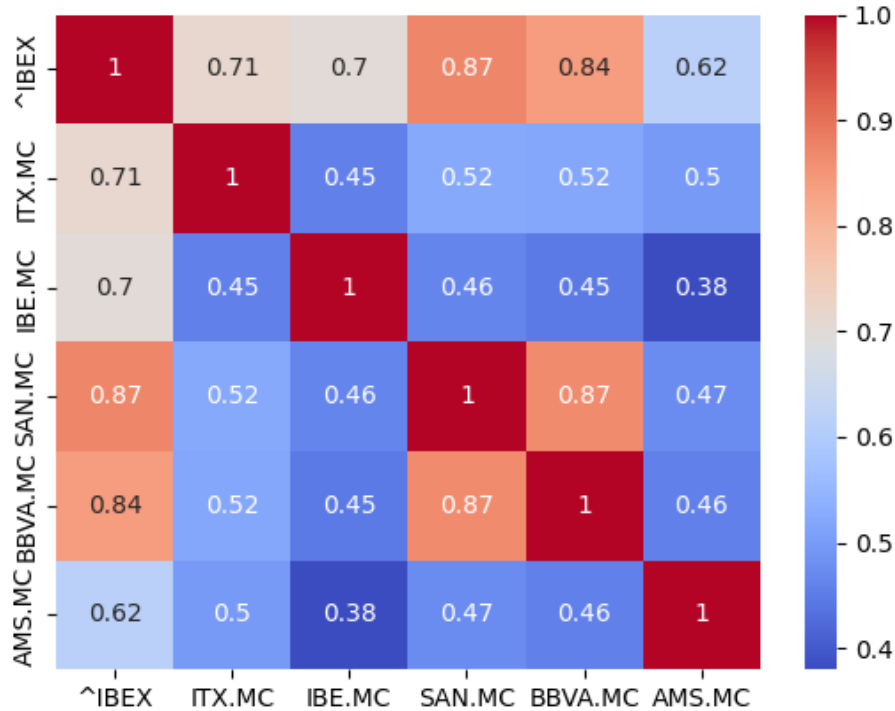
Elaboración propia

8.2. Correlación con el mercado

Previo al estudio de las estrategias de inversión, lo primero que se debe analizar, es el mercado en su conjunto y la correlación de los activos financieros estudiados con el mercado. La correlación permite saber la dependencia lineal entre cada acción y el índice donde cotiza. Es importante para el inversor ya que ayuda en la diversificación de la cartera y reduce los riesgos para el capital.

En la gráfica 25 se muestra la correlación de Pearson, que es la más utilizada y es fácilmente interpretable. Normalmente, la correlación se sitúa en valores contenidos en el intervalo $[-1, 1]$ pero, en esta representación escalada, la mayor correlación se produce con el valor 1 y la menor con el 0. Cuanto más cercano esté el valor de la empresa con respecto al valor del índice IBEX, la empresa se comportará de forma más parecida al mercado. En caso contrario, el movimiento de la acción no guarda relación con el IBEX o se comporta de forma inversa.

Gráfico 25. Correlación entre IBEX35, Inditex, Iberdrola, Santander, BBVA y Amadeus (2013-2023)



Elaboración propia

En la representación mediante el mapa de colores se pueden obtener múltiples interpretaciones necesarias para elaborar una estrategia de inversión. En primer lugar, se analiza la relación entre las empresas y el índice:

- Inditex respecto al Ibex 35: El valor de 0,71 indica que la empresa textil no guarda una estrecha relación por el valor, a pesar de ser el mayor valor del índice.
- Iberdrola respecto al Ibex 35: De manera similar a Inditex, las subidas o bajadas en el índice no son seguidas estrechamente por la empresa energética.
- Santander respecto al Ibex 35: Con un valor de 0.87, el mayor banco de España tiene a evolucionar de manera similar a como lo hace el Ibex, y es la acción que guarda mayor correlación de las cinco.
- BBVA respecto al Ibex 35: El segundo banco español por excelencia, también guarda una fuerte correlación con el Ibex, situándose en 0,84 puntos.

- Amadeus respecto al Ibex 35: La empresa tecnológica posee el menor valor de correlación respecto al índice de todas las estudiadas. Con esto se infiere que la dirección que toma el índice no es seguida por la cotización de Amadeus.

Además de estas importantes correlaciones, se puede destacar el valor de 0,87 entre el Banco Santander y el BBVA, un valor obvio teniendo en cuenta la pertenencia al mismo sector y subsector (banca minorista) y la magnitud internacional de ambos. Finalmente, se debe destacar la baja correlación entre Amadeus y el resto de las empresas, debido a la naturaleza específica del negocio de la empresa del sector tecnológico y turístico.

8.3. Método de inversión

Los resultados diarios de cada predicción futura se comparan respecto al precio de cierre ajustado real que marcó el mercado. En las tablas incluidas en el anexo se puede observar claramente que el modelo no acierta en gran número de ocasiones para los valores estudiados.

En la elaboración de dichas tablas, se ha obtenido el precio de cierre ajustado del día anterior y se ha comparado con la dirección tomada al día siguiente por el precio real y por el precio predicho. El resultado de la comparación puede ser un error cuando difieren ambos precios, un acierto con la flecha hacia arriba indicando un aumento del valor de la acción o un acierto con la flecha hacia abajo indicando que el valor se ha depreciado ese día. Al final de la tabla, se han contado tanto errores como aciertos para calcular el porcentaje de error.

Si atendemos al porcentaje de aciertos de todas las tablas, ninguna supera el 53% de máximo que marca la empresa Iberdrola. En el mínimo de acierto se encuentra Amadeus con un escaso 33%. Por otro lado, el porcentaje máximo de errores lo marca, indubitadamente, Amadeus con un 66%, seguido de Inditex con un 63%. A la luz de estos resultados, una estrategia de compra de acciones no parece la más adecuada.

A pesar de los malos resultados, existen alternativas para intentar aprovechar las predicciones del modelo. Debido a que comprar acciones no sería una buena inversión atendiendo únicamente al modelo predictivo, la compra de un derivado cuyo subyacente sea una de estas 5 acciones, podría ser un posible enfoque. La CNMV lo define de la siguiente manera: “un producto derivado es un instrumento financiero cuyo valor deriva

de la evolución de los precios de otro activo, denominado activo subyacente (CNMV, 2014)”.

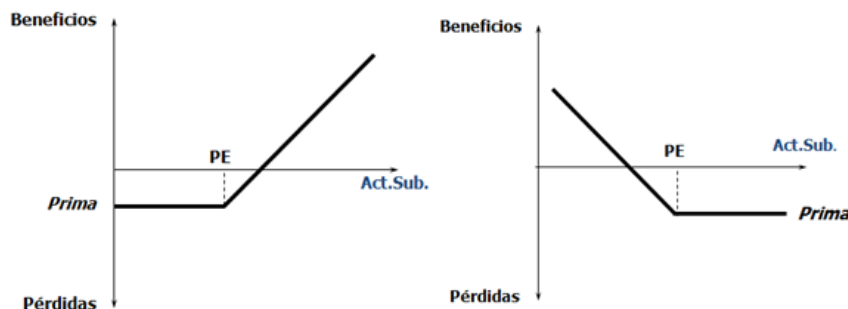
Lo primero que se debe plantear, es el uso que se quiere hacer del derivado. Se puede usar como cobertura de una posición, como especulación, como método de arbitraje o como parte de una estrategia de diversificación de la cartera. En este supuesto, el planteamiento supone que su uso será especulativo.

Una vez elegido el instrumento financiero, existen múltiples tipos de derivados: futuros, opciones, *warrants*, certificados, CFD, etc. Entre estos tipos, la selección como método de inversión corresponderá al uso de opciones.

La CNMV define una opción como: “un contrato que implica un derecho para el comprador y una obligación para el vendedor, a comprar (o vender) una determinada cuantía del activo subyacente en un plazo estipulado a un precio convenido de antemano (precio de ejercicio) (CNMV, 2013)”. Para obtener los derechos mencionados se debe pagar un precio, conocido como la prima. Según el derecho que otorgan las opciones, se distingue la opción de compra o *call* y la opción de venta o *put*. En ambos el comprador decide, ya que tiene el derecho (pero no la obligación) para adquirir o vender el subyacente (acción) al precio ya determinado en la fecha marcada.

Una vez dentro de las opciones, el inversor se situaría siempre en la posición larga o compradora, teniendo en sus manos el derecho a comprar o el derecho a vender. La compra de la opción sería a fecha de un día, para ajustarse al horizonte temporal de cada predicción. Si cree que la acción subirá porque se lo indica la predicción del modelo, compraría una *call*, y si cree que la acción bajará porque así lo marca la predicción, compraría una *put*. De esta forma, las pérdidas estarían limitadas a la prima pagada y la ganancia sería ilimitada.

Ilustración 23. Beneficios y pérdidas posibles con la compra de una *call* y una *put*



Fuente: Tutoriales sobre opciones financieras, Instituto BME (BME, s. f.).

En la ilustración anterior, se puede apreciar como en la compra de una *call* (figura izquierda) las pérdidas cuando desciende del valor de la acción por debajo del precio de ejercicio están limitadas a la prima. En cambio, cuando el precio del activo subyacente aumenta, los beneficios son ilimitados. En sentido contrario, con la compra de una *put* cuando el precio del activo subyacente disminuye, los beneficios son ilimitados, mientras que cuando aumenta, las pérdidas están acotadas al precio de la prima pagada.

Se debe poner en conocimiento que los mercados organizados pueden no disponer de opciones para todas las acciones y sobre cualquier vencimiento, en cuyo caso se buscarían alternativas en mercados no organizados.

9. Conclusiones

En el presente trabajo se ha explorado la teoría detrás de la inteligencia artificial, centrándose en su rama de *machine learning* y, más en profundidad, en el *deep learning* y las redes neuronales recurrentes (RNN). Este punto de vista teórico ha sido el pilar para la construcción de un modelo de predicción para activos financieros.

El sector financiero está avanzando rápidamente en la era digital, sobre todo, en las empresas que combinan las finanzas con la tecnología, las *fintech*. Por ello, el desarrollo de modelos de predicción, para su uso en el sector, ofrece un amplio abanico de posibilidades. A través de diferentes técnicas como redes neuronales, regresiones, o árboles de decisión, entre otras, se pueden encontrar nuevos casos de uso y aplicaciones con los datos financieros.

El análisis de series temporales de tipo *stock*, como son las acciones, resulta un problema complicado, especialmente para arquitecturas más simples, como las regresiones o los árboles. A través del *deep learning* y las redes neuronales recurrentes u otros algoritmos más avanzados, se pueden obtener resultados prometedores en la captura de patrones y dependencias a largo plazo.

A través de la implementación práctica del modelo, mediante la configuración de redes neuronales de tipo *Long short-term memory* (LSTM), se ha pretendido demostrar la viabilidad de utilizar este tipo de redes en la predicción futura de las cotizaciones de una acción en la Bolsa. En la implementación se optó por el uso de la librería 'Keras' como *backend* de 'Tensorflow', además de otras librerías para el manejo de los datos y su visualización. Para la creación del modelo se configuraron los parámetros de tal forma que existiera un equilibrio entre el tiempo de ejecución y el rendimiento de la red neuronal. Es posible obtener un modelo más optimizado, pero requiere mayor cantidad de datos de entrada y la realización de un gran número de pruebas, además de contar con un *hardware* más potente. Un punto interesante para un trabajo futuro puede ser modificar cada hiperparámetro, como las unidades, la pérdida, los *epochs* o el *batch_size*, y realizar métricas de tiempos y capacidades de predicción del modelo.

El modelo creado se ha probado con los precios de cierre ajustados de las empresas con mayor capitalización del índice Ibex 35. En las 5 empresas, más en el propio índice, el modelo se ha testeado comparando datos no conocidos en el entrenamiento del modelo con las predicciones realizadas por este. Los resultados del

test han sido realmente favorables, obteniendo unas métricas de error en MSE y RMSE, relativamente bajas.

Para averiguar el comportamiento del modelo en un caso real, se ha realizado la predicción diaria para cada valor durante 30 sesiones. Para analizar la precisión del modelo se han predicho los precios para las acciones en el mes pasado y en el anterior y se han comparado con los precios efectivamente alcanzados en el mercado bursátil. A diferencia de los resultados del test, las predicciones futuras no han sido certeras. Pese a que el RMSE, en general, es bajo para todas las empresas, en la mayoría de los casos se produce un desplazamiento a la derecha que se liga a un retraso en la predicción para el día siguiente. El retardo provoca que se capten tarde las tendencias, indicando con uno o dos días de retraso un punto de inflexión, que puede ser crítico en un caso real. Si el algoritmo predice una tendencia alcista después de que se haya producido el máximo en el precio y se compra la acción en ese punto, la tendencia nueva bajista, captada tardíamente, podría producir una cuantiosa pérdida de la inversión.

Los resultados diarios para cada valor se han comparado con su precio real y su precio real del día anterior. De esta manera, se ha establecido un mecanismo para contar los aciertos y errores de las predicciones futuras. A la luz de los datos cosechados, se infiere que el modelo no funciona de forma óptima, pues, en general, los aciertos se equiparan a los fallos.

Se ha demostrado que un modelo basado en los datos históricos mediante una arquitectura de tres capas LSTM no arroja conclusiones esperanzadoras en las condiciones anteriormente descritas. Pese a la ingente cantidad de precios de cierre ajustados introducidos en el modelo (más de 10 años de cotizaciones), y, pese a unos buenos resultados en las pruebas con nuevos datos del año vigente, la salida del modelo es incorrecta, al menos, en la mitad de las ocasiones. A pesar de estos puntos negativos, no quiere decir que el modelo sea inservible, ni mucho menos. Este modelo puede ser el cimiento para la creación de uno con mayor complejidad que sea capaz de capturar patrones más rápidamente.

El objetivo de los modelos predictivos no es otro que el principal objetivo de cualquier inversor, batir a los mercados financieros. ¿Para qué querría un inversor perder tiempo y dinero en estudiar los mercados pudiendo indexarse a un índice como el S&P500 que ha obtenido un rendimiento anual medio cercano al 10% en sus casi 100 años de historia? Según Francisco Nogales (2023) es posible batir a los mercados aunque la tarea resulta muy difícil debido a la pequeña relación entre los datos históricos de una acción y sus precios futuros (Nogales, 2023, p. 120). A pesar de la dificultad del

problema, existen evidencias empíricas que muestran la relación positiva entre los fondos que cuentan con herramientas de *machine learning* para la selección de los activos de su cartera y la rentabilidad de la misma (DeMiguel et al., 2021).

Como propuestas de mejora del proyecto en el futuro, se puede estudiar la incorporación de otras variables de entrada como son otros precios históricos, el volumen, eventos y datos financieros de la empresa y del sector, indicadores macroeconómicos o análisis de noticias y sentimientos.

Finalmente, la arquitectura LSTM empleada parece funcionar de forma adecuada, por lo menos en la creación, entrenamiento y prueba de los modelos. De todas formas, existen nuevos tipos de redes como las *Transformer*, o arquitecturas como GRU, que pueden captar patrones más complejos y tener mayor memoria a largo plazo, pudiendo mejorar el funcionamiento del modelo.

En última instancia, el éxito en la aplicación de modelos predictivos para activos financieros en la vida real requiere la prueba de múltiples arquitecturas además de un amplio conocimiento en el análisis de datos, la programación de modelos de aprendizaje automático y sobre el sector financiero.

Anexo

Tabla 1. Comparación del precio del día anterior con el real y su predicción para el Ibex 35

IBEX 35			
Fecha	Precio Real	Predicción	Resultado
19/05/2023	9.251,50		
22/05/2023	9.305,00	9.249,96	ERROR
23/05/2023	9.267,00	9.280,60	ACIERTO ↓
24/05/2023	9.163,50	9.327,21	ERROR
25/05/2023	9.116,10	9.304,05	ERROR
26/05/2023	9.191,10	9.217,79	ACIERTO ↑
29/05/2023	9.180,10	9.165,78	ACIERTO ↓
30/05/2023	9.167,50	9.217,34	ERROR
31/05/2023	9.050,20	9.216,07	ERROR
01/06/2023	9.167,50	9.203,54	ACIERTO ↑
02/06/2023	9.317,30	9.105,19	ERROR
05/06/2023	9.289,10	9.182,52	ACIERTO ↓
06/06/2023	9.310,80	9.315,09	ACIERTO ↑
07/06/2023	9.359,80	9.315,31	ACIERTO ↑
08/06/2023	9.338,30	9.333,57	ACIERTO ↓
09/06/2023	9.309,70	9.377,43	ERROR
12/06/2023	9.344,40	9.371,61	ACIERTO ↑
13/06/2023	9.333,70	9.350,61	ERROR
14/06/2023	9.432,80	9.375,47	ACIERTO ↑
15/06/2023	9.430,80	9.370,38	ACIERTO ↓
16/06/2023	9.495,00	9.448,43	ACIERTO ↑
19/06/2023	9.431,90	9.456,94	ACIERTO ↓
20/06/2023	9.439,80	9.510,77	ACIERTO ↑
21/06/2023	9.436,40	9.468,09	ERROR
22/06/2023	9.364,70	9.470,94	ERROR
23/06/2023	9.265,80	9.468,98	ERROR
26/06/2023	9.274,00	9.412,24	ACIERTO ↑
27/06/2023	9.393,00	9.323,00	ACIERTO ↑
28/06/2023	9.481,30	9.313,62	ERROR
29/06/2023	9.510,60	9.405,64	ERROR
30/06/2023	9.593,00	9.488,76	ERROR
		Total errores	14
		% errores	47%
		% aciertos	53%

Elaboración propia

Tabla 2. Comparación del precio del día anterior con el real y su predicción para Inditex

INDITEX			
Fecha	Precio Real	Predicción	Resultado
19/05/2023	31,39		
22/05/2023	31,65	31,00	ERROR
23/05/2023	31,07	31,08	ACIERTO ↓
24/05/2023	30,77	31,27	ERROR
25/05/2023	30,80	30,90	ACIERTO ↑
26/05/2023	31,13	30,63	ERROR
29/05/2023	31,24	30,58	ERROR
30/05/2023	31,45	30,80	ERROR
31/05/2023	31,29	30,91	ACIERTO ↓
01/06/2023	31,45	31,09	ERROR
02/06/2023	32,15	31,01	ERROR
05/06/2023	32,03	31,12	ACIERTO ↓
06/06/2023	31,82	31,63	ACIERTO ↓
07/06/2023	33,63	31,65	ERROR
08/06/2023	33,37	31,54	ACIERTO ↓
09/06/2023	33,25	32,75	ACIERTO ↓
12/06/2023	33,76	32,77	ERROR
13/06/2023	34,11	32,80	ERROR
14/06/2023	34,21	33,15	ERROR
15/06/2023	34,65	33,45	ERROR
16/06/2023	34,39	33,60	ACIERTO ↓
19/06/2023	33,90	33,94	ACIERTO ↓
20/06/2023	34,11	33,84	ERROR
21/06/2023	34,36	33,53	ERROR
22/06/2023	34,00	33,62	ACIERTO ↓
23/06/2023	33,90	33,77	ACIERTO ↓
26/06/2023	33,93	33,56	ERROR
27/06/2023	34,44	33,46	ERROR
28/06/2023	34,47	33,44	ERROR
29/06/2023	34,99	33,77	ERROR
30/06/2023	35,46	33,84	ERROR

Total errores **19**
 % errores **63%**
 % aciertos **37%**

Elaboración propia

Tabla 3. Comparación del precio del día anterior con el real y su predicción para Iberdrola

IBERDROLA			
Fecha	Precio Real	Predicción	Resultado
19/05/2023	11,32		
22/05/2023	11,31	11,11	ACIERTO ↓
23/05/2023	11,33	11,14	ERROR
24/05/2023	11,29	11,15	ACIERTO ↓
25/05/2023	11,08	11,15	ACIERTO ↓
26/05/2023	11,18	11,12	ACIERTO ↑
29/05/2023	11,15	11,01	ACIERTO ↓
30/05/2023	11,18	11,02	ERROR
31/05/2023	11,09	11,02	ACIERTO ↓
01/06/2023	11,20	11,03	ERROR
02/06/2023	11,27	10,98	ERROR
05/06/2023	11,26	11,01	ACIERTO ↓
06/06/2023	11,37	11,07	ERROR
07/06/2023	11,22	11,08	ACIERTO ↓
08/06/2023	11,14	11,14	ACIERTO ↓
09/06/2023	11,18	11,09	ERROR
12/06/2023	11,23	11,03	ERROR
13/06/2023	11,16	11,04	ACIERTO ↓
14/06/2023	11,26	11,07	ERROR
15/06/2023	11,28	11,05	ERROR
16/06/2023	11,50	11,07	ERROR
19/06/2023	11,39	11,10	ACIERTO ↓
20/06/2023	11,62	11,22	ERROR
21/06/2023	11,51	11,20	ACIERTO ↓
22/06/2023	11,44	11,31	ACIERTO ↓
23/06/2023	11,36	11,30	ACIERTO ↓
26/06/2023	11,34	11,26	ACIERTO ↓
27/06/2023	11,47	11,22	ERROR
28/06/2023	11,55	11,20	ERROR
29/06/2023	11,53	11,26	ACIERTO ↓
30/06/2023	11,63	11,31	ERROR
		Total errores	14
		% errores	47%
		% aciertos	53%

Elaboración propia

Tabla 4. Comparación del precio del día anterior con el real y su predicción para Santander

BANCO SANTANDER			
Fecha	Precio Real	Predicción	Resultado
19/05/2023	3,20		
22/05/2023	3,23	3,16	ERROR
23/05/2023	3,22	3,18	ACIERTO ↓
24/05/2023	3,14	3,21	ACIERTO ↓
25/05/2023	3,15	3,21	ACIERTO ↑
26/05/2023	3,20	3,15	ERROR
29/05/2023	3,14	3,15	ACIERTO ↓
30/05/2023	3,11	3,18	ERROR
31/05/2023	3,05	3,14	ERROR
01/06/2023	3,11	3,11	ACIERTO ↑
02/06/2023	3,21	3,06	ERROR
05/06/2023	3,18	3,09	ACIERTO ↓
06/06/2023	3,21	3,17	ERROR
07/06/2023	3,21	3,16	ERROR
08/06/2023	3,23	3,18	ERROR
09/06/2023	3,20	3,19	ACIERTO ↓
12/06/2023	3,19	3,21	ERROR
13/06/2023	3,19	3,19	ACIERTO ↑
14/06/2023	3,25	3,18	ERROR
15/06/2023	3,19	3,18	ACIERTO ↓
16/06/2023	3,22	3,22	ACIERTO ↑
19/06/2023	3,21	3,19	ACIERTO ↓
20/06/2023	3,19	3,20	ACIERTO ↓
21/06/2023	3,20	3,20	ACIERTO ↑
22/06/2023	3,16	3,18	ACIERTO ↓
23/06/2023	3,07	3,19	ERROR
26/06/2023	3,08	3,15	ACIERTO ↑
27/06/2023	3,20	3,08	ERROR
28/06/2023	3,31	3,08	ERROR
29/06/2023	3,34	3,15	ERROR
30/06/2023	3,38	3,25	ERROR
		<i>Total errores</i>	15
		<i>% errores</i>	50%
		<i>% aciertos</i>	50%

Elaboración propia

Tabla 5. Comparación del precio del día anterior con el real y su predicción para BBVA

BBVA			
Fecha	Precio Real	Predicción	Resultado
19/05/2023	6,49		
22/05/2023	6,50	6,46	ERROR
23/05/2023	6,49	6,48	ACIERTO ↓
24/05/2023	6,34	6,48	ACIERTO ↓
25/05/2023	6,42	6,48	ACIERTO ↑
26/05/2023	6,48	6,49	ACIERTO ↑
29/05/2023	6,40	6,48	ERROR
30/05/2023	6,36	6,36	ACIERTO ↓
31/05/2023	6,12	6,42	ERROR
01/06/2023	6,26	6,46	ACIERTO ↑
02/06/2023	6,49	6,46	ACIERTO ↑
05/06/2023	6,46	6,46	ACIERTO ↓
06/06/2023	6,48	6,40	ERROR
07/06/2023	6,49	6,37	ERROR
08/06/2023	6,59	6,17	ERROR
09/06/2023	6,51	6,27	ACIERTO ↓
12/06/2023	6,55	6,45	ERROR
13/06/2023	6,53	6,45	ACIERTO ↓
14/06/2023	6,78	6,45	ERROR
15/06/2023	6,73	6,44	ACIERTO ↓
16/06/2023	6,79	6,45	ERROR
19/06/2023	6,80	6,47	ERROR
20/06/2023	6,74	6,56	ACIERTO ↓
21/06/2023	6,77	6,50	ERROR
22/06/2023	6,73	6,50	ACIERTO ↓
23/06/2023	6,65	6,50	ACIERTO ↓
26/06/2023	6,75	6,54	ERROR
27/06/2023	6,78	6,52	ERROR
28/06/2023	6,85	6,71	ERROR
29/06/2023	6,96	6,68	ERROR
30/06/2023	7,03	6,74	ERROR
		<i>Total errores</i>	16
		<i>% errores</i>	53%
		<i>% aciertos</i>	47%

Elaboración propia

Tabla 6. Comparación del precio del día anterior con el real y su predicción para Amadeus

AMADEUS IT GROUP			
Fecha	Precio Real	Predicción	Resultado
19/05/2023	66,94		
22/05/2023	67,10	66,50	ERROR
23/05/2023	65,68	66,67	ACIERTO ↓
24/05/2023	65,50	66,81	ERROR
25/05/2023	66,05	65,83	ACIERTO ↑
26/05/2023	66,55	65,51	ERROR
29/05/2023	66,63	65,90	ERROR
30/05/2023	66,96	66,29	ERROR
31/05/2023	66,15	66,34	ACIERTO ↓
01/06/2023	67,50	66,56	ACIERTO ↑
02/06/2023	67,67	66,05	ERROR
05/06/2023	67,40	66,94	ACIERTO ↓
06/06/2023	67,83	67,30	ERROR
07/06/2023	66,94	67,11	ACIERTO ↓
08/06/2023	66,27	67,40	ERROR
09/06/2023	66,07	66,88	ERROR
12/06/2023	67,20	66,28	ACIERTO ↑
13/06/2023	68,25	66,05	ERROR
14/06/2023	68,86	66,83	ERROR
15/06/2023	68,92	67,71	ERROR
16/06/2023	69,14	68,25	ERROR
19/06/2023	68,50	68,39	ACIERTO ↓
20/06/2023	68,48	68,65	ERROR
21/06/2023	68,11	68,31	ACIERTO ↓
22/06/2023	67,93	68,26	ERROR
23/06/2023	67,58	68,01	ERROR
26/06/2023	67,38	67,81	ERROR
27/06/2023	67,97	67,49	ACIERTO ↑
28/06/2023	68,43	67,26	ERROR
29/06/2023	68,50	67,64	ERROR
30/06/2023	68,92	68,03	ERROR
		<i>Total errores</i>	20
		<i>% errores</i>	67%
		<i>% aciertos</i>	33%

Elaboración propia

Bibliografía

- Alarcón, N. (2020, julio 7). *OpenAI Presents GPT-3, a 175 Billion Parameters Language Model*. NVIDIA Technical Blog. <https://developer.nvidia.com/blog/openai-presents-gpt-3-a-175-billion-parameters-language-model/>
- Alonso-Robisco, A., & Carbo, J. M. (2023). Aprendizaje automático en modelos de concesión de crédito: Oportunidades y riesgos. *Funcas*.
- Alternative.me. (s. f.). *Crypto Fear & Greed Index—Bitcoin Sentiment*. Recuperado 14 de julio de 2023, de <https://alternative.me/crypto/fear-and-greed-index/>
- Alubankudi, B., & Tapia, M. (2014). *¿Qué sabemos de la negociación de alta frecuencia?* Bolsas y Mercados. https://www.bolsasymercados.es/esp/Estudios-Publicaciones/Documento/5118__Qu%C3%A9_sabemos_de_la_negociaci%C3%B3n_de_alta_frecuencia_
- AWS Amazon. (2023). *¿Qué es el Procesamiento del lenguaje natural? - NLP - AWS*. Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/nlp/>
- BBVA. (2023, mayo 11). *El científico de datos, un profesional clave para los clientes del sector bancario*. BBVA NOTICIAS. <https://www.bbva.com/es/innovacion/el-cientifico-de-datos-un-profesional-clave-para-el-sector-bancario-y-para-sus-clientes/>
- BME. (s. f.). *Tutoriales sobre opciones financieras*. BME. Recuperado 14 de julio de 2023, de <https://www.institutobme.es/es/home/finanzas-basicas/tutoriales/opciones-financieras.html>
- Bobadilla, J. (2021). *Machine Learning y Deep Learning: Usando Python, Scikit y Keras*. Ediciones de la U.

ChatGPT. (2023). *ChatGPT*. <https://chat.openai.com>

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. <https://doi.org/10.48550/ARXIV.1406.1078>

CNMV. (2013, diciembre 10). *CNMV - Opciones*. <https://www.cnmv.es/Portal/inversor/Opciones.aspx>

CNMV. (2014, enero 17). *CNMV - Productos derivados*. <https://www.cnmv.es/Portal/inversor/Derivados.aspx>

CNMV. (2023, julio 6). *CNMV - Glosario Financiero*. <https://www.cnmv.es/Portal/Inversor/Glosario.aspx?id=0&term=Activo&idlang=1>

DeMiguel, V., Gil-Bazo, J., Nogales, F. J., & A. P. Santos, A. (2021). Can Machine Learning Help to Select Portfolios of Mutual Funds? *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3768753>

elEconomista.es. (2010, octubre 1). *Las causas del flash crash al descubierto: La SEC publica su informe oficial*. elEconomista.es. <https://www.eleconomista.es/mercados-cotizaciones/noticias/2492339/10/10/Las-causas-del-flash-crash-al-descubierto-la-SEC-publica-su-informe-oficial.html>

FundéuRAE. (2022, diciembre 16). *Palabra del año 2022 | FundéuRAE*. <https://www.fundeu.es/palabra-del-ano-2022/>

Garate, G. (2021, julio 6). *¿Qué entendemos por algoritmo?* [Universidad de la Empresa (UDE)]. <https://ude.edu.uy/que-son-algoritmos/>

- Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (Second edition). O'Reilly Media, Inc.
- Hasbrouck, J., & Saar, G. (2011). Low-Latency Trading. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.1695460>
- IBM. (s. f.). *¿Qué es machine learning? | IBM*. Recuperado 4 de julio de 2023, de <https://www.ibm.com/es-es/topics/machine-learning>
- Jiménez, A. (2018, junio 17). La estructura de los datos. *Medium*. https://medium.com/@ana_datagirl/la-estructura-de-los-datos-9c0acaf00afd
- Latza, T., Marsh, I. W., & Payne, R. (2014). Fast Aggressive Trading. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.2542184>
- Maté, A. I., García, I. M., & Suárez, G. R. (2021). Educación financiera y decisiones de ahorro e inversión: Un análisis de la Encuesta de Competencias Financieras (ECF). *CNMV*, 75.
- MathWorks. (s. f.). *Redes generativas antagónicas*. Recuperado 5 de julio de 2023, de <https://es.mathworks.com/discovery/generative-adversarial-networks.html>
- Merino, M. (2023, febrero 24). *Google llega tarde a competir con ChatGPT... Pero sin una legendaria innovación suya de 2017 su rival ni existiría*. Genbeta. <https://www.genbeta.com/a-fondo/google-llega-tarde-a-competir-chatgpt-legendaria-innovacion-suya-2017-su-rival-existiria>
- Microsoft Azure. (2023). *¿Qué es el aprendizaje automático? | Microsoft Azure*. <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-machine-learning-platform>
- National Geographic. (2019, agosto 5). *Breve historia visual de la inteligencia artificial*. www.nationalgeographic.com.es.

https://www.nationalgeographic.com.es/ciencia/breve-historia-visual-inteligencia-artificial_14419

Nogales, F. J. (2023, mayo). *¿Es posible batir a los mercados financieros usando herramientas de big data y de machine learning?* Funcas. <https://www.funcas.es/articulos/es-posible-batir-a-los-mercados-financieros-usando-herramientas-de-big-data-y-de-machine-learning/>

NVIDIA. (s. f.). *¿GPU vs. CPU? ¿Qué es la computación por GPU?* | NVIDIA. Recuperado 5 de julio de 2023, de <https://www.nvidia.com/es-la/drivers/what-is-gpu-computing/>

Parra, F. (2019). *Estadística y Machine Learning con R*. <https://bookdown.org/content/2274/portada.html>

Prieto Meléndez, R., Herrera, A., Pérez, J., & Padrón-Godínez, A. (2000). *EL MODELO NEURONAL DE McCULLOCH Y PITTS Interpretación Comparativa del Modelo*.

Ramírez, F. (2018, julio 19). *Historia de la IA: Frank Rosenblatt y el Mark I Perceptrón, el primer ordenador fabricado específicamente para crear redes neuronales en 1957*. Think Big. <https://telefonicatech.com/blog/historia-de-la-ia-frank-rosenblatt-y-e>

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). *DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter*. <https://doi.org/10.48550/ARXIV.1910.01108>

The Economist. (2016, junio 23). From not working to neural networking. *The Economist*. <https://www.economist.com/special-report/2016/06/23/from-not-working-to-neural-networking>

Thompson, E. (2019, septiembre 6). 10 ejemplos de que ya dependes de la IA en tu vida diaria. *OpenMind*. <https://www.bbvaopenmind.com/tecnologia/inteligencia-artificial/10-ejemplos-de-que-ya-dependes-de-la-ia-en-tu-vida-diaria/>

TIOBE Index. (2023, junio). TIOBE. <https://www.tiobe.com/tiobe-index/>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. <https://doi.org/10.48550/ARXIV.1706.03762>

Yahoo Finance. (s. f.). *¿Qué es el cierre ajustado? | Finance Ayuda - SLN28256*. Recuperado 14 de julio de 2023, de <http://es.ayuda.yahoo.com/kb/SLN28256.html>

Ye, M., Yao, C., & Gai, J. (2012). The Externality of High Frequency Trading. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.2066839>

