

The logo consists of the lowercase letters "viu" in white, centered within a solid orange circle.

**Universidad
Internacional
de Valencia**

Análisis de Sentimientos en la Formación de una Cartera de Inversión

Titulación:
Máster universitario en
Big Data y Ciencia de
datos

Curso Académico
2024-2025

Alumno/a: Sellés Pérez,
Alejandro
DNI: 74380598R

Director/a del TFM: Edith
Cecilia Macedo Ruiz

Convocatoria:
Septiembre 2025

Primera

Índice general

Índice de figuras	2
Índice de cuadros	2
1 Introducción	8
2 Marco teórico	10
2.1 Redes neuronales	10
2.1.1 Neuronas artificiales	10
2.1.2 Redes neuronales artificiales	11
2.1.3 Entrenamiento de una red neuronal	12
2.2 Modelado de secuencias	13
2.2.1 Redes neuronales recurrentes	14
2.2.2 Arquitectura LSTM	17
2.2.3 Arquitectura GRU	18
2.3 Técnicas de Procesamiento del Lenguaje Natural para Análisis de Sentimientos en Finanzas	19
2.3.1 Del PLN clásico a los modelos de lenguaje modernos	20
2.3.2 Limitaciones de RNN, LSTM y GRU en PLN	20
2.3.3 Arquitectura Transformer y mecanismo de atención	21
2.3.4 Modelos preentrenados: BERT y FinBERT	24
2.4 Estado del arte: Modelos de Redes Neuronales para predecir la bolsa	26
3 Experimentación	28
3.1 Objetivos	28
3.2 Entorno de trabajo	29
3.3 Presentación del dataset	29
3.4 Metodología	31
3.4.1 Preprocesamiento de datos	31
3.4.2 Entrenamiento de modelos	32
3.4.3 Evaluación de modelos	34
3.4.4 Combinación de modelos	35

3.5 Resultados	37
3.5.1 Gilead Sciences (GILD)	37
3.5.2 Coca-Cola (KO)	41
3.5.3 Wells Fargo (WFC)	44
3.5.4 Merck & Co. (MRK)	47
3.5.5 Invesco QQQ Trust (QQQ)	50
3.5.6 NVIDIA (NVDA)	53
4 Conclusiones	56
5 Siguientes pasos	58
Referencias	59

Índice de figuras

2.1	<u>Comparación de neuronas</u>	11
2.2	<u>Esquema de una red neuronal</u>	12
2.3	<u>RNN vs NN</u>	14
2.4	<u>Esquema de una RNN</u>	15
2.5	<u>Arquitectura de una celda de RNN</u>	16
2.6	<u>Arquitectura de una celda LSTM</u>	18
2.7	<u>Arquitectura de una celda GRU</u>	19
2.8	<u>Comparación entre embeddings estáticos y contextuales)</u>	21
2.9	<u>Arquitectura Transformer (Vaswani et al., 2017)</u>	22
2.10	<u>Diagrama del Pre-training y Fine-Tuning de BERT (Devlin et al., 2019)</u>	25
2.11	<u>Diagrama del input de BERT (Devlin et al., 2019)</u>	25
3.1	<u>Histórico del precio de cierre - GILD</u>	37
3.2	<u>Subidas y bajadas 2023 - GILD</u>	38
3.3	<u>Balance de clases - GILD</u>	38
3.4	<u>Histórico del precio de cierre - KO</u>	41
3.5	<u>Subidas y bajadas 2023 - KO</u>	42
3.6	<u>Balance de clases - KO</u>	42
3.7	<u>Histórico del precio de cierre - WFC</u>	44
3.8	<u>Subidas y bajadas 2023 - WFC</u>	45
3.9	<u>Balance de clases - WFC</u>	45
3.10	<u>Histórico del precio de cierre - MRK</u>	47
3.11	<u>Subidas y bajadas 2023 - MRK</u>	47
3.12	<u>Balance de clases - MRK</u>	48
3.13	<u>Histórico del precio de cierre - QQQ</u>	50
3.14	<u>Subidas y bajadas 2023 - QQQ</u>	51
3.15	<u>Balance de clases - QQQ</u>	51
3.16	<u>Histórico del precio de cierre - NVDA</u>	53

3.17 <u>Subidas y bajadas 2023 - NVDA</u>	54
3.18 <u>Balance de clases - NVDA</u>	54

Índice de cuadros

3.1	Resultados de Clasificación Binaria - GILD	39
3.2	Resultados de Regresión Logarítmica - GILD	39
3.3	Resultados del Ensemble por Accuracy - GILD	40
3.4	Resultados del Ensemble por Precisión - GILD	40
3.5	Resultados de Clasificación Binaria - KO	43
3.6	Resultados de Regresión Logarítmica - KO	43
3.7	Resultados del Ensemble por Accuracy - KO	43
3.8	Resultados del Ensemble por Precisión - KO	43
3.9	Resultados de Clasificación Binaria - WFC	46
3.10	Resultados de Regresión Logarítmica - WFC	46
3.11	Resultados del Ensemble por Accuracy - WFC	46
3.12	Resultados del Ensemble por Precisión - WFC	46
3.13	Resultados de Clasificación Binaria - MRK	48
3.14	Resultados de Regresión Logarítmica - MRK	49
3.15	Resultados del Ensemble por Accuracy - MRK	49
3.16	Resultados del Ensemble por Precisión - MRK	49
3.17	Resultados de Clasificación Binaria - QQQ	52
3.18	Resultados de Regresión Logarítmica - QQQ	52
3.19	Resultados del Ensemble por Accuracy - QQQ	52
3.20	Resultados del Ensemble por Precisión - QQQ	52
3.21	Resultados de Clasificación Binaria - NVDA	55
3.22	Resultados de Regresión Logarítmica - NVDA	55
3.23	Resultados del Ensemble por Accuracy - NVDA	55
3.24	Resultados del Ensemble por Precisión - NVDA	55

Resumen

A lo largo de este trabajo se evalúa la capacidad de modelos de *Machine Learning* para anticipar, a muy corto plazo, el signo del movimiento en el precio de una acción, con el fin de diseñar una estrategia o bot de *trading* automatizado que incorpore dichos impulsos en una cartera de inversión. Además, con el objetivo de aportar una mayor riqueza a los modelos, no solo se consideran variables técnicas derivadas de las series temporales de los precios, sino que también se incorporan indicadores de análisis de sentimiento extraídos de noticias corporativas mediante modelos basados en *BERT*.

Se exploran dos enfoques principales para el problema: la clasificación binaria y la regresión de retornos logarítmicos. Los modelos considerados incluyen redes neuronales recurrentes (*LSTM* y *GRU*) y *Random Forest*. Además, para mejorar la robustez de las predicciones se diseñan varios *ensamblados* de votación ponderada, con pesos calculados mediante una función *softmax*, siguiendo dos criterios: uno orientado a los mejores modelos según el *Accuracy*, y otro enfocado en la *Precisión* de la clase positiva (*Comprar*).

Los resultados obtenidos se ajustan a lo esperado: la predicción a muy corto plazo presenta limitaciones inherentes y constituye un problema altamente complejo. En consecuencia, los modelos desarrollados no son lo suficientemente fiables como para operar de forma autónoma en un bot de *trading*. Sin embargo, algunos sí muestran potencial como herramientas de apoyo en el análisis técnico. En particular, los *ensamblados* basados en el *Accuracy* resultan útiles en general para identificar con fiabilidad los escenarios en los que conviene *no comprar*, mientras que los esquemas orientados a la *Precisión* de la clase positiva ofrecen un mejor desempeño al anticipar oportunidades de *compra*.

Palabras clave: trading algorítmico, análisis de sentimiento financiero, BERT, LSTM, GRU, series temporales financieras.

Abstract

This work evaluates the ability of *Machine Learning* models to anticipate, in the very short term, the direction of stock price movements, with the aim of designing an automated *trading* strategy or bot that incorporates these signals into an investment portfolio. To enrich the models, not only are technical variables derived from price time series considered, but also sentiment indicators extracted from corporate news using *BERT*-based models.

Two main approaches to the problem are explored: binary classification and regression of logarithmic returns. The models considered include recurrent neural networks (*LSTM* and *GRU*) and *Random Forest*. To enhance predictive robustness, several weighted-voting *ensembles* are designed, with weights computed through a *softmax* function, following two criteria: one oriented towards the best models according to *Accuracy*, and another focused on the *Precision* of the positive class (*Buy*).

The results are consistent with expectations: very short-term prediction presents inherent limitations and remains a highly complex challenge. Consequently, the developed models are not reliable enough to operate autonomously in a *trading* bot. Nevertheless, some show potential as decision-support tools within technical analysis. In particular, *Accuracy*-based ensembles prove useful for reliably identifying scenarios in which it is advisable *not to buy*, while those oriented towards the *Precision* of the positive class perform better at anticipating *buy* opportunities.

Keywords: algorithmic trading, financial sentiment analysis, BERT, LSTM, GRU, financial time series.

1. Introducción

La bolsa de valores refleja, en gran medida, la complejidad y la incertidumbre propias del comportamiento humano. Como señaló John Maynard Keynes: “*los mercados pueden permanecer irracionales más tiempo del que usted puede permanecer solvente*”. Esta afirmación sintetiza la dificultad de anticipar la dinámica de un sistema en el que confluyen decisiones individuales, acontecimientos globales y factores económicos. Esta combinación de racionalidad e imprevisibilidad convierte la predicción bursátil en un desafío en el que la matemática se entrelaza con la psicología y la sociología.

Durante décadas, los métodos tradicionales para modelar los mercados financieros se apoyaron en herramientas matemáticas y estadísticas, como los modelos ARIMA o GARCH, ampliamente utilizados para capturar patrones temporales y volatilidad. Asimismo, las ecuaciones diferenciales estocásticas y el modelo de Black–Scholes desempeñaron un papel central en la valoración de derivados y activos financieros, sentando las bases de la teoría financiera moderna. No obstante, estos enfoques presentan limitaciones a la hora de representar la naturaleza no lineal, dinámica y multifactorial de los precios, especialmente en horizontes temporales cortos. Además, la influencia de factores externos, tales como noticias, acontecimientos sociales o cambios en el ánimo del mercado, resulta difícil de modelar cuantitativamente mediante estas técnicas clásicas.

En los últimos años, el *aprendizaje automático* ha abierto nuevas vías para la predicción bursátil. Las redes neuronales recurrentes, en particular las arquitecturas *LSTM*, han mostrado capacidad para modelar dependencias en series temporales y resultan adecuadas para datos financieros, aunque su uso sigue enfrentando problemas de sobreajuste y falta de interpretabilidad.

Al mismo tiempo, sabemos que el precio de las acciones no depende únicamente de su historial, sino también de factores externos como noticias y opinión pública. Esto ha impulsado el empleo de técnicas de procesamiento del lenguaje natural y análisis de sentimientos para complementar los modelos tradicionales.

En este contexto, este trabajo plantea un enfoque híbrido que combina *RNN* con señales de sentimiento obtenidas de noticias financieras, con la idea de integrar patrones históricos y componentes de percepción del mercado. Esta línea se apoya en investigaciones recientes, como *Ensemble time series models for stock price prediction and portfolio optimization with sentiment analysis* (Narayana et al., 2025), que destacan la utilidad de fusionar datos heterogéneos para capturar la complejidad de los mercados a corto plazo.

Finalmente, cabe destacar que el objetivo último del proyecto fue aportar una visión práctica y actualizada sobre el empleo de *Machine Learning* en la predicción financiera, con especial atención a horizontes muy cortos, como las variaciones de

los precios de cierre de un día para otro. Desde el inicio se asumió que identificar patrones en escalas tan reducidas es un reto altamente complejo y conlleva una elevada probabilidad de obtener malos resultados y modelos poco fiables. Aun así, dado que este enfoque ha sido menos estudiado en comparación a otro tipo de predicciones bursátiles en ventanas más amplias, este trabajo pretende servir como prueba inicial y punto de partida para futuras investigaciones.

2. Marco teórico

2.1. Redes neuronales

A lo largo de la historia, numerosos avances en ingeniería se inspiraron en fenómenos presentes en la naturaleza, y la inteligencia artificial no fue una excepción. Muchos de sus modelos partieron de principios biológicos para construir sistemas computacionales complejos. Uno de los ejemplos más representativos fueron las *Redes Neuronales Artificiales (RNA)*, cuyo diseño se basó, aunque de forma simplificada, en el funcionamiento del cerebro humano. En él, miles de millones de neuronas se conectan a través de sinapsis, conformando una red dinámica capaz de aprender, adaptarse y almacenar información. Esta idea sirvió de base para el desarrollo de algoritmos de *Machine Learning* que buscaron emular dichas capacidades mediante estructuras matemáticas y computacionales.

Neuronas artificiales

El concepto de *neurona artificial* nació en 1943, cuando McCulloch y Pitts (1943) presentaron un modelo lógico simplificado del funcionamiento neuronal. En su propuesta, el comportamiento de una neurona se representaba mediante una combinación de operaciones booleanas, lo que dio lugar a un sistema capaz de simular procesos de decisión básicos. Este trabajo sentó las bases teóricas para el desarrollo posterior de las *Redes Neuronales Artificiales*.

Un avance decisivo en esta línea fue realizado por Rosenblatt (1958), al introducir el *perceptrón*, un modelo más sofisticado que incorporaba elementos esenciales del *Machine Learning*: ponderación de entradas, función de activación y ajuste de parámetros. Esta arquitectura, aunque todavía limitada, permitió por primera vez entrenar una red para resolver problemas simples de clasificación.

Desde un punto de vista matemático, el *perceptrón* puede entenderse como una generalización del modelo de regresión lineal. En este contexto, se define una variable de salida Y como combinación lineal de un conjunto de variables de entrada x_1, x_2, \dots, x_n , con un término de error aleatorio ε :

$$Y = f(x_1, x_2, \dots, x_n) + \varepsilon. \quad (2.1)$$

Cuando f es lineal, esta relación se expresa como:

$$Y = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n + \varepsilon. \quad (2.2)$$

En el modelo neuronal, los coeficientes w_1, \dots, w_n pasan a interpretarse como

pesos sinápticos, que indican la influencia de cada entrada sobre la salida, mientras que w_0 se conoce como *sesgo*. La salida del sistema se obtiene al aplicar una función de activación ϕ a la suma ponderada de las entradas, permitiendo así introducir no linealidad en el modelo:

$$\hat{Y} = \phi \left(w_0 + \sum_{i=1}^n w_i x_i \right). \quad (2.3)$$

Este paralelismo con la regresión lineal facilita la interpretación de las *neuronas artificiales* y permite establecer una comparación directa con una neurona biológica. En este marco, los *pesos sinápticos* se interpretan como la fuerza de las conexiones entre neuronas, mientras que el *sesgo* funciona como un umbral que determina la activación de la neurona. Las entradas x_i se asocian con las *dendritas*, encargadas de recibir señales de otras neuronas. La suma ponderada de estas entradas y la función de activación ϕ resultan análogas al *soma*, el cuerpo celular donde se integra la información y se decide si se genera un impulso eléctrico. Finalmente, la salida \hat{Y} se compara con el *axón*, prolongación responsable de transmitir la señal a otras neuronas para continuar el proceso de comunicación. De este modo, la función de activación simula el disparo neuronal no lineal, transformando la combinación de señales entrantes en una respuesta adecuada para el procesamiento de información.

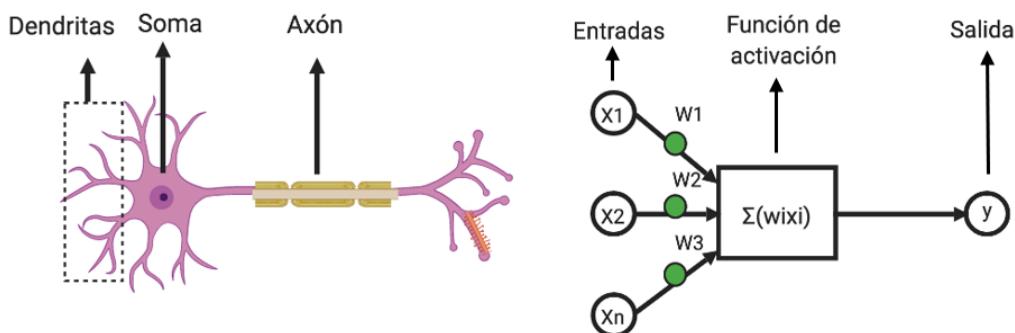


Figura 2.1: Comparación de neuronas

Redes neuronales artificiales

Una *red neuronal artificial* se construye conectando múltiples *neuronas artificiales* de manera organizada en capas. Cada neurona recibe señales de la capa anterior y transmite su salida a la siguiente. Las neuronas que reciben directamente las entradas externas x_1, \dots, x_n conforman la *capa de entrada*. Las que generan el resultado final de la red constituyen la *capa de salida*. Entre ambas se sitúan las

denominadas *capas ocultas*, encargadas de procesar y transformar la información a través de sus conexiones.

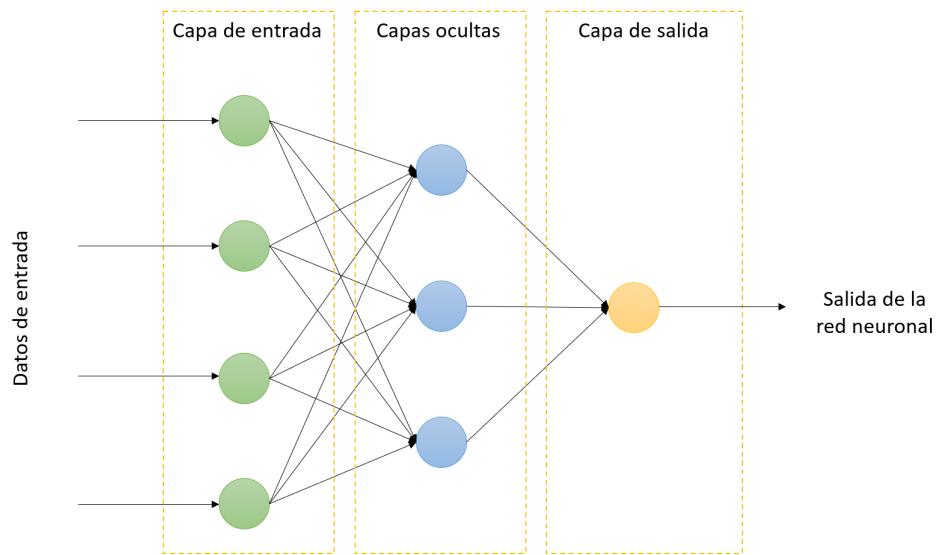


Figura 2.2: Esquema de una red neuronal

Si solo se aplicaran transformaciones lineales en cada neurona, la combinación de todas estas capas sería equivalente a una sola operación lineal, lo que limitaría el poder expresivo de la red. Para evitar esta restricción, cada neurona aplica una función no lineal conocida como *función de activación* a la suma ponderada de sus entradas. Estas funciones permiten a la red aprender representaciones complejas y resolver problemas no lineales.

Entre las funciones de activación más utilizadas en redes neuronales se encuentran:

- **ReLU:** $\phi(x) = \max(0, x)$.
- **Sigmoide:** $\sigma(x) = \frac{1}{1+e^{-x}}$.
- **Tangente hiperbólica:** $\phi(x) = \tanh(x)$.
- **Softmax:** $\phi(x_i) = \frac{e^{x_i}}{\sum_{k=1}^n e^{x_k}}$ para $i = 1, \dots, n$.

Entrenamiento de una red neuronal

El entrenamiento de una *red neuronal* consiste en ajustar los *pesos sinápticos* de cada conexión con el objetivo de minimizar el error entre las salidas del modelo y los valores esperados. Para ello se emplea un conjunto de datos conocido, que suele dividirse en dos subconjuntos: uno de *entrenamiento* (aproximadamente el 80 %) y

otro de *test* (el 20 % restante), lo que permite evaluar la capacidad de generalización del modelo.

El algoritmo más utilizado para ajustar los parámetros de una *red neuronal* es la *retropropagación del error* (*backpropagation*), propuesta por Rumelhart, Hinton y Williams (1986). Este método se fundamenta en el *descenso del gradiente* para minimizar una función de coste C , que cuantifica la diferencia entre las predicciones de la red y los valores esperados. La elección de esta función depende del tipo de problema: en tareas de regresión suele emplearse el error cuadrático medio, mientras que en problemas de clasificación es habitual utilizar la entropía cruzada. A partir de esta función, el algoritmo determina cómo deben ajustarse los pesos de cada neurona propagando el error desde la salida hacia las capas anteriores, optimizando de este modo el rendimiento del modelo.

La *retropropagación* se basa en aplicar la regla de la cadena para calcular los gradientes de la función de coste con respecto a los pesos y sesgos de cada capa. Dados los pesos W^l , sesgos b^l y activaciones a^l , la propagación hacia delante y el cálculo de los gradientes se resumen en las siguientes expresiones:

$$z^l = a^{l-1} \cdot W^l + b^l, \quad a^l = f(z^l), \quad (2.4)$$

$$\frac{\partial C}{\partial W^l} = a^{l-1} \delta^l, \quad \frac{\partial C}{\partial b^l} = \delta^l, \quad \delta^l = \frac{\partial C}{\partial z^l}. \quad (2.5)$$

Este proceso se repite durante múltiples iteraciones (*epochs*), subdividiendo el conjunto de entrenamiento en pequeños lotes denominados *batches*, y actualizando los pesos de manera progresiva tras cada uno. La combinación de estas técnicas permite entrenar redes profundas de forma eficiente y estable.

2.2. Modelado de secuencias

Existen numerosas tareas en las que los datos presentan una estructura secuencial, es decir, donde el orden de los elementos aporta información fundamental para su análisis. Este es el caso de las series temporales financieras, como la evolución del precio de una acción a lo largo del tiempo; o, del lenguaje natural, donde el significado de una palabra depende del contexto en el que aparece. En este trabajo, el objetivo consiste en predecir el comportamiento de las acciones del mercado de valores combinando información histórica del precio con análisis de sentimientos extraído de noticias o redes sociales.

Para abordar este tipo de problemas se emplean modelos capaces de capturar las dependencias temporales y contextuales de los datos. Las *Redes Neuronales Recurrentes* (*RNN*) y sus variantes, como las *Long Short-Term Memory* (*LSTM*) y las *Gated Recurrent Unit* (*GRU*), han mostrado una notable eficacia en este ámbito, por lo que en los apartados siguientes se describirá su funcionamiento y su utilidad para el modelado de secuencias.

Redes neuronales recurrentes

Las *Redes Neuronales Recurrentes (RNN)* constituyen una clase de redes neuronales diseñadas específicamente para procesar datos secuenciales. A diferencia de las redes *feedforward*, en las que cada entrada se procesa de forma independiente, las *RNN* introducen conexiones recurrentes que permiten mantener información del pasado mediante un estado interno. De este modo, la salida en un instante dado depende no solo de la entrada actual, sino también del historial de entradas anteriores, almacenado en el estado oculto.

Esta propiedad las hace especialmente adecuadas para tareas en las que el orden y la dependencia temporal resultan fundamentales, como el modelado del lenguaje, el reconocimiento de voz o la predicción de series temporales. El concepto fue introducido por Rumelhart, Hinton y Williams (1986), sobre la base de contribuciones previas como las de Hopfield (1982).

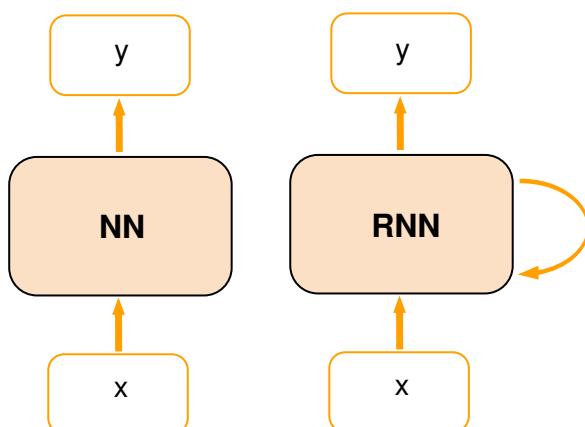


Figura 2.3: [RNN vs NN](#)

La principal diferencia radica en que una *RNN* incorpora una noción de *memoria temporal*. Esta memoria se representa mediante un vector de estado oculto, el cual se actualiza en cada paso temporal. En otras palabras, la red mantiene una representación interna de lo que ha procesado hasta ese momento, lo que le permite tomar decisiones más informadas sobre la entrada actual.

Formalmente, el funcionamiento de una *RNN* se entiende de forma más clara al visualizar su estructura desenrollada en el tiempo. Aunque en su forma compacta aparece como una sola celda con bucles, al extenderla secuencialmente se observa cómo la información fluye desde un paso temporal al siguiente. En cada instante t , la red recibe una entrada x_t y el estado oculto anterior h_{t-1} , y produce un nuevo estado oculto h_t y, en muchos casos, una salida y_t .

La actualización del estado oculto suele expresarse mediante las siguientes ecuaciones:

$$h_t = \phi(Ux_t + Wh_{t-1} + b_h) , \quad (2.6)$$

$$y_t = \psi(Vh_t + b_y) , \quad (2.7)$$

donde:

- U , W y V son matrices de pesos entrenables que transforman la entrada, el estado oculto y la salida respectivamente.
- b_h y b_y son vectores de sesgo.
- ϕ y ψ son funciones de activación.

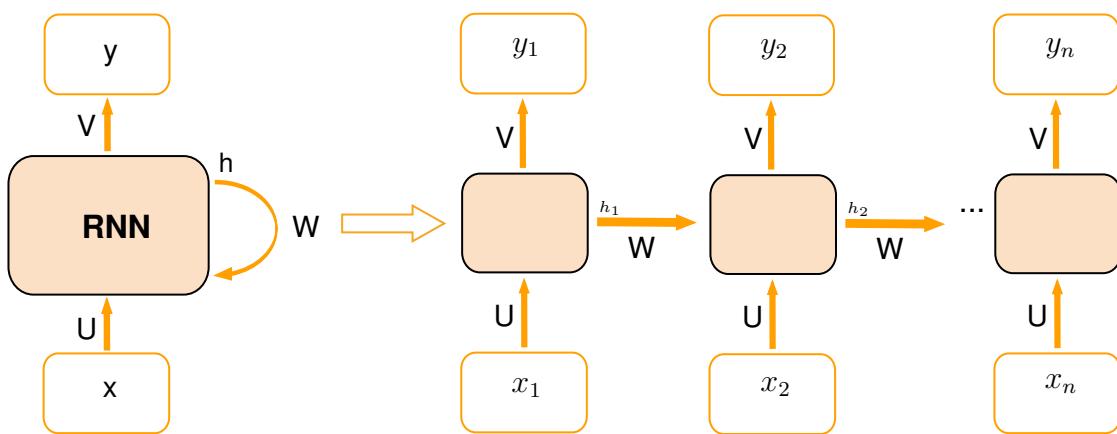


Figura 2.4: Esquema de una RNN

Para comprender mejor esta operación resulta útil centrarse en una única celda de una *RNN*. En cada paso temporal, la celda recibe dos entradas: la entrada externa x_t y el estado oculto anterior h_{t-1} , que actúa como memoria de lo ocurrido en pasos previos. Ambas fuentes de información se combinan linealmente mediante las matrices de pesos U y W , tras lo cual se aplica una función de activación no lineal para generar el nuevo estado oculto h_t . Este estado se propaga a la siguiente celda y, en caso necesario, también puede emplearse para producir una salida y_t .

Este estado oculto actúa como una memoria dinámica que transporta información de manera recurrente. Sin embargo, como se señaló en (Bengio, Simard, & Frasconi, 1994), estas redes presentan una limitación importante: su capacidad para recordar información disminuye con el tiempo, especialmente en secuencias largas. La causa principal es el problema del *desvanecimiento* o *explosión* del gradiente: a medida que los errores se retropropagan a lo largo de muchos pasos temporales, los gradientes pueden volverse extremadamente pequeños o grandes, lo que impide un aprendizaje efectivo. En consecuencia, la red tiende a “olvidar” información relevante del pasado distante, limitando su rendimiento en tareas que requieren memoria.

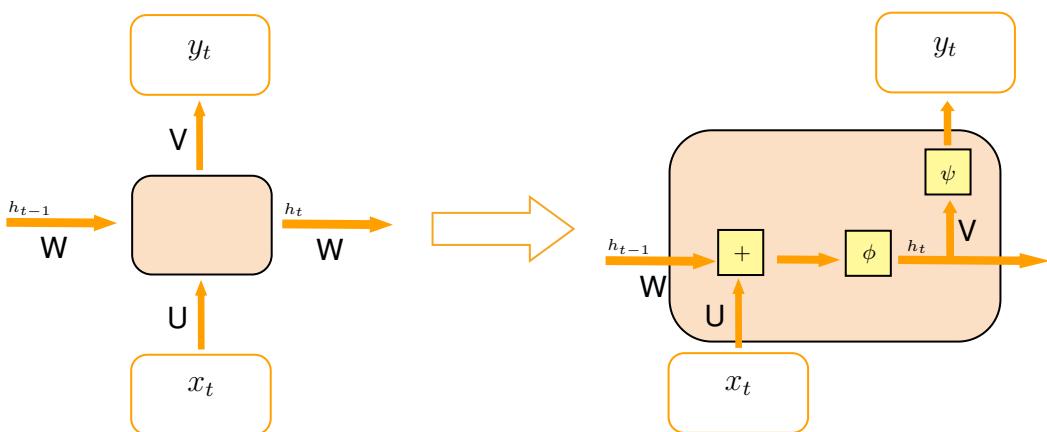


Figura 2.5: Arquitectura de una celda de RNN

prolongada. Este obstáculo motivó el desarrollo de arquitecturas más sofisticadas, como *LSTM* y *GRU*, diseñadas específicamente para mitigar este problema.

Por otro lado, las *Redes Neuronales Recurrentes* ofrecen una gran flexibilidad para procesar secuencias, ya que pueden configurarse en función de cómo se presenten los datos de entrada y de las salidas que se desean obtener. Tal como se señala en (Goodfellow, Bengio, & Courville, 2016), esta versatilidad da lugar a diferentes arquitecturas que se adaptan a la naturaleza específica de cada problema secuencial:

- *Uno a uno*: una única entrada produce una única salida. Es equivalente a una red tradicional sin carácter secuencial.
- *Uno a muchos*: a partir de una única entrada se genera una secuencia de salidas. Un ejemplo típico es la generación de texto a partir de una etiqueta inicial.
- *Muchos a uno*: se procesa una secuencia de entradas para producir una única salida. Este es el caso más común en tareas de clasificación o en la predicción de series temporales, como la que se aborda en este trabajo.
- *Muchos a muchos*: tanto la entrada como la salida son secuencias. Puede aplicarse de forma directa (alineación uno a uno) o mediante un enfoque *encoder-decoder* (secuencia completa a secuencia completa), como en los sistemas de traducción automática.

Estas variantes permiten adaptar la arquitectura de una *RNN* a distintos tipos de problema. En el caso concreto de este trabajo, se emplea una arquitectura del tipo *muchos a uno*, ya que se toma una ventana temporal con valores históricos de precios de acciones y noticias empresariales como entrada, con el objetivo de predecir la evolución futura de un activo financiero.

Una consideración importante en el entrenamiento de estas redes es que no se utiliza la retropropagación estándar, sino una variante adaptada denominada *retropropagación a través del tiempo* (*Backpropagation Through Time*, *BPTT*) (Goodfellow, Bengio, & Courville, 2016). Esta técnica consiste en desenrollar la red en el tiempo durante la fase de entrenamiento y aplicar retropropagación acumulando gradientes paso a paso, como si se tratara de una red profunda con tantas capas como pasos temporales.

Arquitectura LSTM

Como se señaló anteriormente, las *RNN* tradicionales presentan dificultades para aprender dependencias a largo plazo. Para resolver este inconveniente, Hochreiter y Schmidhuber (1997) introdujeron la arquitectura *Long Short-Term Memory* (*LSTM*), que incorpora una célula de memoria capaz de conservar información relevante durante intervalos de tiempo arbitrariamente largos.

En cada instante t , la célula *LSTM* mantiene dos vectores de estado:

- c_t : estado de celda, que almacena la memoria a largo plazo.
- h_t : estado oculto, que actúa como memoria a corto plazo y salida de la célula.

El flujo de información se controla mediante tres compuertas, cada una implementada como una transformación afín seguida de una función de activación:

- Puerta de olvido: $f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$, con σ la sigmoide.
- Puerta de entrada: $i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$.
- Propuesta de nuevo contenido: $\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$.
- Puerta de salida: $o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$.

A continuación, tomando \odot como el producto elemento a elemento, el estado de la celda y el estado oculto se actualizan mediante:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t , \quad (2.8)$$

$$h_t = o_t \odot \tanh(c_t) . \quad (2.9)$$

De este modo, las compuertas $\{f_t, i_t, o_t\}$ controlan, respectivamente, qué información se descarta, cuál se incorpora al estado y qué parte del estado de la celda se expone en la salida. Esta estructura permite que la célula *LSTM* mantenga gradientes estables y aprenda dependencias a largo plazo de forma eficaz.

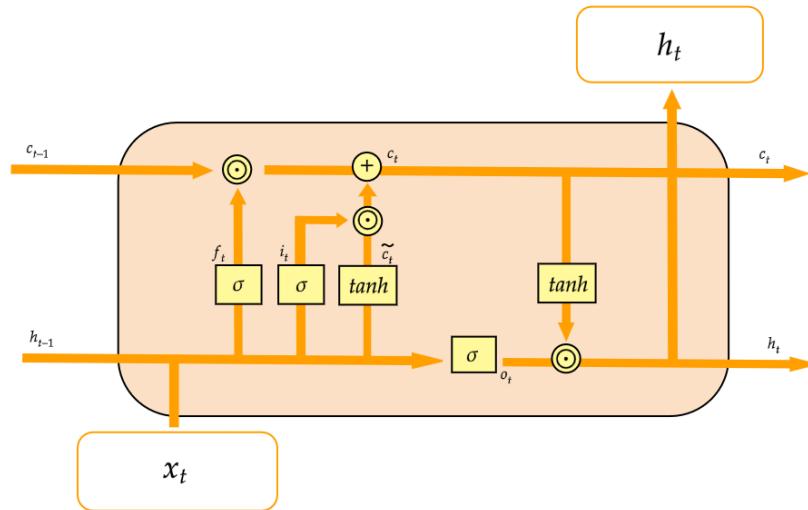


Figura 2.6: Arquitectura de una celda LSTM

Arquitectura GRU

Como alternativa a la arquitectura *LSTM*, Cho (2014) propuso las *Gated Recurrent Units (GRU)*, con el objetivo de simplificar el diseño de las redes recurrentes con compuertas, manteniendo la capacidad de aprender dependencias a largo plazo pero con menos parámetros y mayor eficiencia computacional.

El motivo principal para introducir las *GRU* fue reducir la complejidad de las *LSTM*, eliminando el estado de celda separado c_t y fusionando algunas compuertas, de modo que la red resultara más sencilla y rápida de entrenar sin perder poder expresivo.

En cada instante t , la célula *GRU* mantiene un único vector de estado oculto h_t , que cumple tanto el papel de memoria a corto plazo como a largo plazo.

El flujo de información se controla mediante dos compuertas, definidas como transformaciones afines seguidas de funciones de activación:

- Puerta de actualización:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) . \quad (2.10)$$

- Puerta de reinicio:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) . \quad (2.11)$$

A partir de estas compuertas se calcula un estado oculto candidato, que constituye la propuesta de actualización del estado en el instante t :

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) , \quad (2.12)$$

donde \odot es el producto elemento a elemento.

Finalmente, el estado oculto se actualiza combinando el estado anterior con el candidato a través de la compuerta de actualización:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t . \quad (2.13)$$

De este modo, la compuerta de reinicio r_t regula qué parte del estado anterior se emplea para generar el nuevo contenido, mientras que la compuerta de actualización z_t determina en qué medida se conserva el estado previo y cuánto se sustituye por la nueva información.

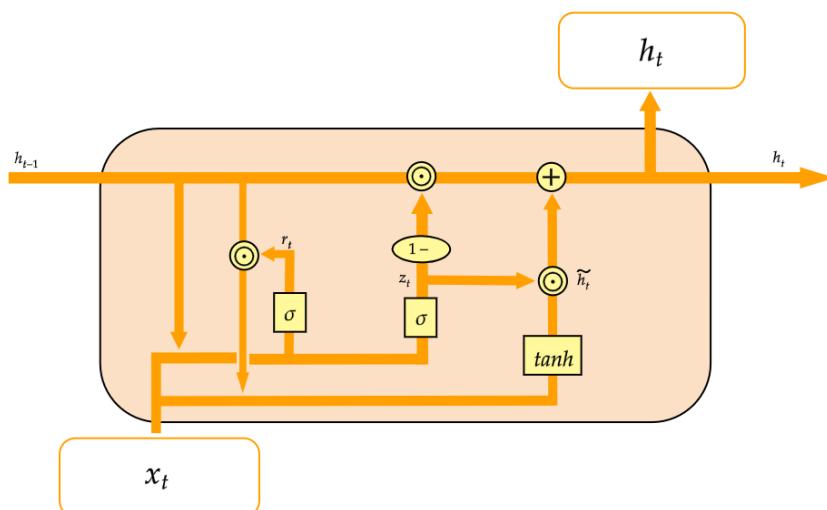


Figura 2.7: Arquitectura de una celda GRU

Esta arquitectura más simple permite que las *GRU* aprendan dependencias a largo plazo de manera eficaz, utilizando menos parámetros y, en general, con un mejor desempeño computacional, lo que las convierte en una alternativa popular a las *LSTM*.

2.3. Técnicas de Procesamiento del Lenguaje Natural para Análisis de Sentimientos en Finanzas

En esta sección se presentan de forma breve los métodos clásicos de representación del lenguaje, con el fin de situar el contexto en el que surgen los modelos

modernos. A continuación se explica el funcionamiento de la arquitectura *Transformer* y se introduce *BERT*, junto con su adaptación al dominio financiero (*FinBERT*), que constituyen la base del análisis de sentimientos aplicado en este trabajo.

Del PLN clásico a los modelos de lenguaje modernos

Los primeros enfoques de *Procesamiento del Lenguaje Natural* se basaron en representaciones por conteo, como *Bag of Words (BoW)* y *TF-IDF* (Salton & Buckley, 1988). En estos métodos, cada documento se representa mediante un vector que refleja la frecuencia de aparición de las palabras, sin tener en cuenta el orden ni las relaciones semánticas entre ellas. Aunque estas técnicas resultan sencillas y eficaces en tareas básicas de clasificación de texto, generan vectores de alta dimensionalidad y carecen de capacidad para capturar el significado del lenguaje.

El desarrollo posterior de *embeddings* como *Word2Vec* (Mikolov, Chen, Corrado, & Dean, 2013) y *GloVe* (Pennington, Socher, & Manning, 2014) supuso un avance significativo. Estos modelos representan las palabras en un espacio continuo de baja dimensión, de modo que la proximidad entre vectores refleja relaciones semánticas y sintácticas. Matemáticamente, un *embedding* estático puede entenderse como una función

$$f : \mathcal{V} \rightarrow \mathbb{R}^d , \quad (2.14)$$

que asigna a cada palabra w de un vocabulario \mathcal{V} un vector fijo $f(w)$ en un espacio de dimensión d . Este vector permanece invariable independientemente de la frase en la que aparezca la palabra. Por ejemplo, el término *banco* tendría un único vector, pese a que su significado difiera entre *banco financiero* y *banco de un parque*.

Para superar esta limitación surgieron los *embeddings contextuales*, empleados en los modelos de lenguaje modernos. En este caso, la representación de una palabra depende del contexto en el que aparece. Formalmente, cada palabra w_t en una secuencia (w_1, w_2, \dots, w_T) se asocia a un vector h_t que es función no solo de w_t , sino también del resto de palabras de la secuencia:

$$h_t = f(w_t | w_1, w_2, \dots, w_T) . \quad (2.15)$$

De este modo, un mismo término puede adquirir representaciones distintas según su uso, lo que permite capturar ambigüedades semánticas y dependencias de largo alcance. Este cambio de paradigma resultó decisivo para el éxito de modelos como *ELMo* (Peters et al., 2018) y, posteriormente, *BERT* (Devlin et al., 2019), en los que las representaciones contextuales constituyen el núcleo de su funcionamiento.

Limitaciones de RNN, LSTM y GRU en PLN

Las arquitecturas recurrentes, incluidas variantes más avanzadas como *LSTM* y *GRU*, han sido ampliamente utilizadas para el modelado de secuencias en *Pro-*

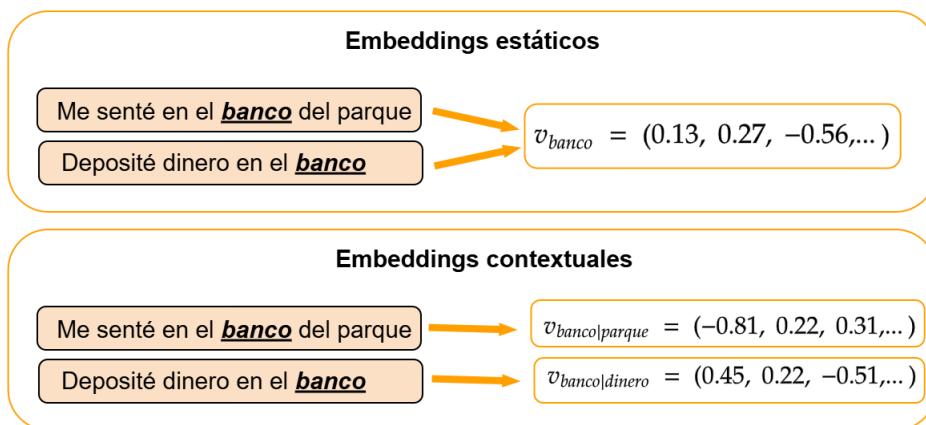


Figura 2.8: Comparación entre embeddings estáticos y contextuales)

cesamiento del Lenguaje Natural. Sin embargo, presentan limitaciones importantes cuando se aplican a textos extensos:

1. *Procesamiento secuencial*: cada estado oculto h_t depende del anterior h_{t-1} , lo que impide la parallelización y aumenta el tiempo de entrenamiento en comparación con arquitecturas que procesan todas las posiciones de la secuencia en paralelo.
2. *Problemas de gradiente*: aunque las variantes con compuertas mitigan parcialmente el problema, persisten los fenómenos de *desvanecimiento* y *explosión* del gradiente, lo que dificulta el aprendizaje de dependencias de largo alcance.
3. *Memoria limitada*: la información contextual se condensa en un vector de estado que se actualiza en cada paso, lo que provoca pérdida de detalles relevantes en secuencias largas.

Estas limitaciones, tanto en eficiencia como en capacidad de modelado, motivaron la búsqueda de alternativas más adecuadas. Los mecanismos de atención y, en particular, la arquitectura *Transformer*, surgieron como respuesta a estas dificultades al permitir un modelado más directo de dependencias de largo alcance y un entrenamiento altamente paralelo.

Arquitectura Transformer y mecanismo de atención

Los *Transformers* fueron introducidos por Vaswani et al. (2017) en el artículo *Attention is All You Need*. Esta propuesta marcó una ruptura con los enfoques basados

en recurrencia, al sustituirla completamente por mecanismos de atención. La arquitectura original sigue un esquema *encoder–decoder*: el *encoder* procesa la secuencia de entrada para obtener representaciones intermedias, mientras que el *decoder* genera la salida paso a paso a partir de dichas representaciones y de su propio historial. Esta estructura resultó especialmente eficaz en tareas de traducción automática, aunque el bloque de *encoder* por sí solo se emplea como base en modelos de lenguaje preentrenados como *BERT*.

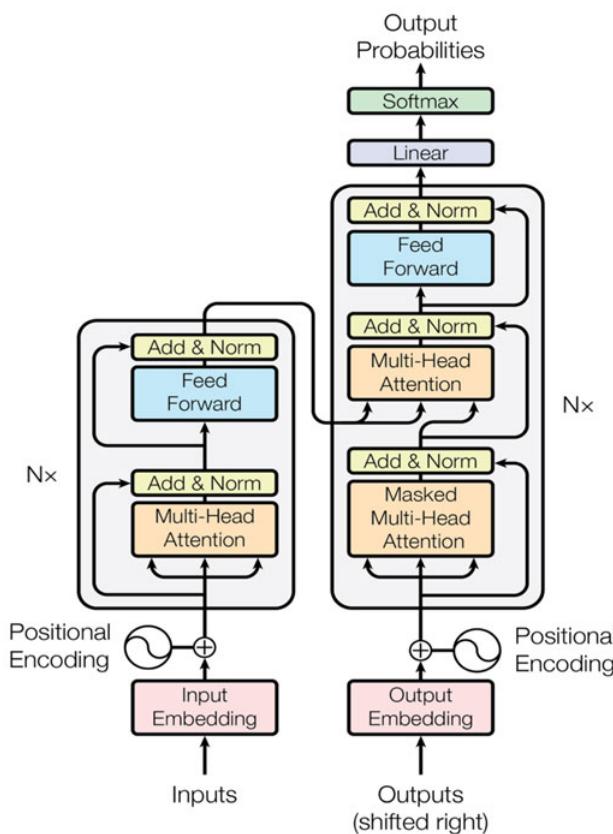


Figura 2.9: [Arquitectura Transformer \(Vaswani et al., 2017\)](#)

Codificación posicional

A diferencia de las *RNN*, los *Transformers* no procesan las secuencias de manera ordenada paso a paso, por lo que resulta necesario incorporar información posicional explícita a los *embeddings* de entrada. Para ello se utilizan los *positional encodings*, que añaden un patrón determinista de senos y cosenos de distinta frecuencia en cada dimensión del *embedding*:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right), \quad (2.16)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right), \quad (2.17)$$

donde *pos* representa la posición en la secuencia y *d* la dimensión del *embedding*. De este modo, cada palabra conserva su posición relativa y el modelo puede diferenciar, por ejemplo, entre «el banco está cerrado» y «cerrado está el banco».

Atención y autoatención

A diferencia de los modelos basados en recurrencia, como *LSTM* o *GRU*, los *Transformers* emplean un mecanismo de atención para modelar las relaciones entre los elementos de una secuencia (Vaswani et al., 2017). Una función de atención puede entenderse como un mapeo entre una *consulta* (*query*) y un conjunto de pares *clave–valor*, cuya salida es una combinación ponderada de los valores. Los pesos de esta combinación se determinan mediante una función de compatibilidad que evalúa el grado de relación entre cada consulta y cada clave.

En el caso concreto de los *Transformers*, se utiliza la *atención por producto punto escalado* (*scaled dot-product attention*). Dadas las matrices de consultas *Q*, claves *K* y valores *V*, el mecanismo calcula la matriz de atención como:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \quad (2.18)$$

donde *d_k* es la dimensión de las claves. El producto punto *QK^T* mide la similitud entre cada consulta y cada clave; la normalización por $\sqrt{d_k}$ evita valores excesivamente grandes; y la función *softmax* transforma estas similitudes en distribuciones de probabilidad que actúan como pesos sobre los valores.

Este mecanismo se aplica en varios puntos de la arquitectura mostrada en la Figura 2.9. En el *encoder* se emplea *autoatención*, ya que *Q*, *K* y *V* provienen de la misma secuencia de entrada: cada palabra ajusta su representación en función de todas las demás. En el *decoder* también se aplica *autoatención*, pero con enmascaramiento para que cada posición solo pueda atender a las anteriores, evitando así el uso de información futura durante la generación. Finalmente, en el bloque que conecta *encoder* y *decoder*, las consultas provienen del *decoder* mientras que las claves y valores proceden del *encoder*, lo que permite que la salida dependa explícitamente de la representación de la entrada.

Atención multi-cabeza

Para capturar distintos tipos de relaciones de forma simultánea, los *Transformers* emplean varias *cabezas de atención* en paralelo. Cada cabeza opera sobre proyecciones diferentes de *Q*, *K* y *V*:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V). \quad (2.19)$$

Posteriormente, las salidas de todas las cabezas se concatenan y se proyectan mediante una matriz W^O :

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O . \quad (2.20)$$

Este mecanismo permite que algunas cabezas se especialicen en dependencias locales, asociadas a palabras cercanas, mientras que otras capturan dependencias de mayor alcance, vinculadas a palabras más distantes, lo que enriquece la representación final.

Ventajas principales

El diseño del *Transformer* introduce mejoras clave frente a las arquitecturas recurrentes, lo que explica por qué se ha convertido en la base de los modelos de lenguaje modernos:

- Permite procesar todas las posiciones de la secuencia en paralelo, reduciendo de forma significativa el tiempo de entrenamiento.
- Facilita la captura de dependencias de largo alcance de manera directa y estable.
- Integra múltiples perspectivas mediante el mecanismo de atención multi-cabeza.

Modelos preentrenados: BERT y FinBERT

BERT (*Bidirectional Encoder Representations from Transformers*) fue presentado por Devlin et al. (2019) como un modelo de lenguaje basado únicamente en la pila de *encoders* de la arquitectura *Transformer*. Su principal contribución fue un preentrenamiento profundo y bidireccional, que le permite generar representaciones contextuales más ricas que las de modelos anteriores.

El preentrenamiento de *BERT* se apoya en dos tareas generales:

- *Masked Language Modeling (MLM)*: se enmascaran aleatoriamente algunas palabras de la entrada y el modelo debe predecirlas en función del contexto. Formalmente, dado un conjunto de tokens $\{w_1, \dots, w_T\}$, se reemplaza un subconjunto M por el token especial [MASK] y el modelo estima la probabilidad $P(w_m | w_{\setminus M})$ para cada $m \in M$.
- *Next Sentence Prediction (NSP)*: el modelo recibe pares de oraciones y debe predecir si la segunda sigue realmente a la primera en el texto original, lo que refuerza su capacidad para capturar relaciones de coherencia entre frases.

Estas tareas no están orientadas de manera directa al análisis de sentimientos, sino a dotar al modelo de un conocimiento general del lenguaje. Para aplicar *BERT* a una tarea específica se recurre al *fine-tuning*, es decir, un ajuste adicional con ejemplos de la tarea objetivo.

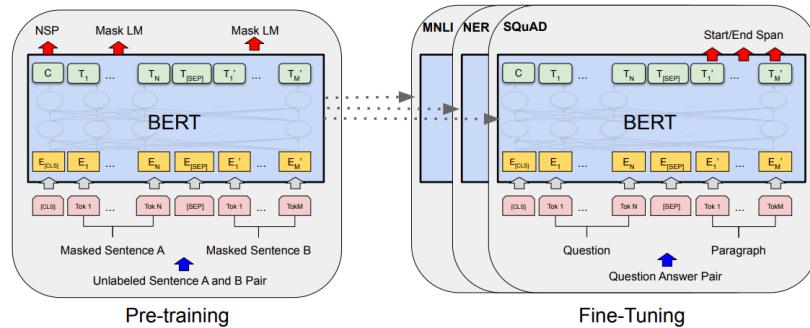


Figura 2.10: Diagrama del Pre-training y Fine-Tuning de BERT (Devlin et al., 2019)

En el caso de la clasificación de sentimiento, la secuencia de entrada se construye añadiendo el token especial [CLS] al inicio del texto (en este trabajo, el *titular* y el *resumen* concatenados de cada noticia). Tras procesar la secuencia en las capas del *Transformer*, la representación final del token [CLS] actúa como un *embedding* global del texto. Este vector se conecta a una capa clasificadora con función *softmax*:

$$y = \text{softmax}(W h_{[CLS]} + b), \quad (2.21)$$

donde $h_{[CLS]}$ es la representación final del token [CLS]. De este modo, el modelo aprende a asignar probabilidades a las clases de sentimiento (*positivo*, *neutral*, *negativo*) a partir de ejemplos anotados.

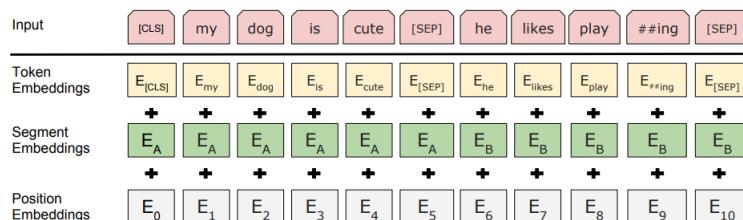


Figura 2.11: Diagrama del input de BERT (Devlin et al., 2019)

Dado que el lenguaje financiero presenta características propias, en 2019 se introdujo *FinBERT* (Araci, 2019), una variante de *BERT* adaptada específicamente a este dominio. El modelo fue entrenado sobre extensos corpus de informes financieros y noticias de mercado, lo que le permite captar matrices que el *BERT* genérico no refleja con la misma precisión. Por este motivo, y dado que el objetivo de este trabajo es analizar el sentimiento en noticias financieras, se emplea *FinBERT* como herramienta principal para obtener las clasificaciones de sentimiento.

2.4. Estado del arte: Modelos de Redes Neuronales para predecir la bolsa

El uso de técnicas de *aprendizaje automático* en los mercados financieros creció de manera notable en los últimos años, en paralelo al auge del *trading* algorítmico y de los denominados *trading bots*. Estos sistemas, ampliamente extendidos en la operativa de alta frecuencia, se apoyaron en técnicas de *Machine Learning* para detectar patrones y ejecutar órdenes en cuestión de milisegundos (Chan, 2013). Su popularización impulsó también la investigación académica en modelos predictivos para series temporales financieras y análisis de sentimiento.

Tradicionalmente, en el ámbito profesional de las grandes gestoras y fondos de inversión, las estrategias cuantitativas estuvieron dominadas por métodos matemáticos y estadísticos, como los modelos estocásticos de precios (e.g., procesos de Itô, modelos de difusión) o las finanzas cuantitativas clásicas (Shreve, 2004). Estos enfoques permitieron modelar dinámicas de precios bajo supuestos probabilísticos y constituyeron la base de herramientas ampliamente utilizadas en la valoración de derivados o en estrategias de arbitraje. Sin embargo, su capacidad para incorporar información no estructurada, como noticias o datos de redes sociales, resultaba limitada.

En el ámbito de las redes neuronales, las primeras aproximaciones aplicaron arquitecturas *feed-forward* a series de precios, combinadas con indicadores técnicos (White, 1988). Aunque ofrecieron resultados iniciales prometedores, estas redes presentaban dificultades para generalizar y eran muy sensibles al ruido característico de los mercados financieros. Posteriormente, se exploraron otros enfoques como las máquinas de soporte vectorial, que también demostraron utilidad en la predicción de series temporales financieras (Kim, 2003).

Con la llegada de las redes recurrentes, especialmente *LSTM* y *GRU*, se produjo un avance importante en la capacidad de modelar dependencias temporales. Estos modelos resultaron más adecuados para capturar patrones secuenciales en series financieras y fueron aplicados con éxito a la predicción de precios y tendencias (Fischer & Krauss, 2018). Trabajos posteriores confirmaron la utilidad de las *LSTM* en mercados emergentes y en diferentes contextos financieros (Chen, Zhou, & Dai, 2015). No obstante, las redes recurrentes mantenían limitaciones derivadas de su naturaleza secuencial, que encarecía su entrenamiento y dificultaba la integración de información heterogénea.

El desarrollo de los *Transformers* abrió una nueva línea de investigación, al eliminar la dependencia estrictamente secuencial y permitir entrenamientos más paralelizables. Modelos como *BERT* y su adaptación *FinBERT* demostraron gran capacidad para procesar información textual procedente de noticias, informes o redes sociales, captando matices lingüísticos relevantes para la dinámica de los mercados. En muchos casos, estos enfoques superaron a los modelos recurrentes tradicionales,

sobre todo cuando se combinaron con variables de mercado en esquemas multimodales (Araci, 2019). Investigaciones recientes ampliaron esta línea con propuestas como *FinGPT*, un modelo basado en *Transformers* entrenado específicamente para tareas financieras, que demostró un rendimiento competitivo en predicción de sentimiento y generación de señales de mercado (Zhang, Han, Xu, & Li, 2023).

En la actualidad, la investigación más avanzada, del mismo modo que el presente trabajo, apunta hacia modelos híbridos que integran series temporales de precios con información textual y de sentimiento, lo que abre la puerta a predicciones más robustas y ajustadas al contexto real de los mercados.

3. Experimentación

El objetivo de esta parte práctica consiste en diseñar y evaluar modelos predictivos de precios bursátiles que integren tanto información histórica de las series temporales como variables derivadas del análisis de sentimiento extraído de noticias financieras mediante técnicas de *Procesamiento del Lenguaje Natural*.

De este modo, se aprovechan no solo los patrones temporales presentes en los datos de precios, sino también la influencia del ánimo del mercado capturada a través del lenguaje. La combinación de ambos enfoques busca mejorar la capacidad predictiva y la robustez de los modelos frente a la complejidad inherente al comportamiento financiero.

A continuación, se presentan los detalles relativos a los datos empleados, los objetivos concretos del proyecto y la metodología aplicada para la construcción, integración y evaluación de los modelos.

3.1. Objetivos

El objetivo principal de este trabajo consiste en desarrollar un sistema predictivo capaz de anticipar, a partir de información financiera y de noticias, si el precio de una acción subirá o bajará al día siguiente. Con estas predicciones, se busca diseñar una estrategia de *trading* automatizada (bot de *trading algorítmico*) que aproveche dichas señales para tomar decisiones de compra o venta de manera sistemática.

Los objetivos específicos de este trabajo son los siguientes:

- Probar y comparar la eficacia de distintos algoritmos de *Machine Learning* y redes neuronales en la predicción de la dirección del precio de las acciones en el corto plazo.. En particular, la comparación entre modelos recurrentes como *LSTM* y *GRU* frente a modelos basados en árboles de decisión.
- Incorporar variables derivadas del análisis de sentimientos de noticias financieras y evaluar su impacto en el rendimiento de los modelos y en la calidad de las predicciones.
- Elaborar modelos de *ensamblado* capaces de integrar la información de los mejores modelos individuales, con el fin de mejorar la capacidad predictiva respecto a su desempeño por separado.
- Analizar la viabilidad del sistema desarrollado como herramienta práctica para la elaboración de un bot de *trading* automatizado, o bien como apoyo en la toma de decisiones de inversión, evaluando en última instancia si los modelos presentan utilidad real en estos contextos.

3.2. Entorno de trabajo

Para el desarrollo de este trabajo se empleó la plataforma *Google Colab Pro*, en su modalidad de suscripción mensual de 10,95 €. En particular, para la ejecución de la mayoría de los *notebooks* se utilizó una GPU NVIDIA T4, que facilitó el entrenamiento y evaluación eficiente de los modelos de aprendizaje profundo.

El flujo de trabajo se apoyó en las principales librerías del ecosistema científico de *Python*, incluyendo herramientas para el procesamiento y manipulación de datos, como NumPy y pandas; para el modelado estadístico y de *Machine Learning*, como scikit-learn, XGBoost o statsmodels; así como librerías especializadas en redes neuronales, entre ellas TensorFlow y Keras.

En el caso de los modelos de lenguaje natural, como se señaló en el apartado anterior, se empleó *FinBERT*, una variante de *BERT* preentrenada en textos financieros. Este modelo, disponible públicamente a través de la plataforma Hugging Face, se integró en *Python* mediante la librería transformers, que permitió la descarga automática y el uso del modelo junto con su *tokenizer* correspondiente.

Adicionalmente, se contó con el apoyo de técnicas de *IA generativa*, en particular *ChatGPT* (OpenAI, 2023), que se empleó como herramienta de apoyo para la generación y depuración de código, la revisión de erratas y la búsqueda de referencias bibliográficas. Con el fin de mantener la trazabilidad del proceso, las interacciones realizadas mediante *prompts* se documentaron junto con el resto del material del proyecto.

Finalmente, se utilizó la plataforma *GitHub* como repositorio de control de versiones y para la documentación del trabajo. Todos los *notebooks* desarrollados, las pruebas realizadas, los modelos entrenados y las conversaciones con *ChatGPT* se encuentran disponibles en el siguiente enlace:

<https://github.com/data4selles/MSc-Big-Data-and-Data-Science-Projects>

3.3. Presentación del dataset

Para la construcción y evaluación de los modelos predictivos se empleó el conjunto de datos *FNSPID: Financial News Dataset for Stock Price Movement Prediction* (Zhong, 2020), que integra información de mercado junto con un amplio número de noticias financieras. Los precios históricos de las acciones (apertura, cierre, máximos, mínimos y volumen) se obtuvieron mediante la librería yfinance. A partir de esta información se generaron 92 variables técnicas con la librería ta, que incluyeron indicadores ampliamente utilizados en *trading* como medias móviles, RSI, MACD, bandas de Bollinger y distintas medidas de volatilidad.

El periodo de análisis abarcó desde 2012 hasta 2023. Para el diseño experimental, los datos correspondientes al intervalo 2012–2021 se destinaron al entre-

namiento, el año 2022 se empleó como conjunto de validación para la selección de modelos y el año 2023 se reservó para la evaluación final. Esta división respondió a criterios de coherencia temporal: un histórico suficientemente amplio se utilizó para el entrenamiento; un año intermedio se destinó al ajuste de hiperparámetros sin comprometer información futura; y el último año disponible se reservó como muestra independiente con el fin de estimar el rendimiento real de los modelos en condiciones fuera de muestra.

La selección de compañías respondió a su diversidad sectorial y a la heterogeneidad de sus comportamientos en el mercado. Concretamente, se trabajó con:

- **Gilead Sciences (GILD)**: empresa del sector farmacéutico cuyo comportamiento en 2020 destacó por un crecimiento al alza motivado por la pandemia, en contraste con la evolución negativa del mercado en general.
- **Coca-Cola (KO)**: multinacional de bienes de consumo representativa de un sector defensivo, caracterizado por estabilidad en sus precios y menor exposición a coyunturas macroeconómicas adversas.
- **Merck & Co. (MRK)**: compañía farmacéutica marcada por la innovación que mostró patrones diferentes respecto a *Gilead Sciences* durante la pandemia.
- **Invesco QQQ Trust (QQQ)**: fondo cotizado (*ETF*) que replica el índice Nasdaq-100 y refleja la evolución agregada de las principales empresas tecnológicas, aportando una perspectiva diversificada frente al resto de acciones individuales.
- **Wells Fargo (WFC)**: entidad financiera de gran relevancia en el sector bancario estadounidense, cuya cotización estuvo especialmente condicionada por problemas regulatorios y por la sensibilidad a los ciclos económicos.
- **NVIDIA (NVDA)**: compañía tecnológica que en los últimos años experimentó un crecimiento constante impulsado por la demanda de procesadores gráficos, con episodios de fuertes revalorizaciones asociados al auge de la inteligencia artificial y la computación de alto rendimiento.

Antes de presentar los resultados, en el apartado 3.5 se llevó a cabo un análisis exploratorio de los datos de cada compañía, con el objetivo de contextualizar el rendimiento de los modelos. Este análisis se apoyó tanto en la evolución histórica de las cotizaciones como en la revisión de noticias particulares del propio conjunto de datos, lo que permitió vincular determinados eventos con los movimientos observados en el mercado.

3.4. Metodología

La metodología adoptada en este trabajo se inspira en el planteamiento de Narayana et al. (2025), con las adaptaciones necesarias para ajustarse al alcance y los objetivos de este TFM. Se estructura en cuatro fases principales que permiten organizar el desarrollo del estudio de manera coherente:

- *Preprocesamiento de datos*: se preparan y transforman tanto las series temporales de precios como las variables derivadas del análisis de sentimientos, con el fin de generar un conjunto de datos consistente y adecuado para el modelado.
- *Entrenamiento de modelos*: se ajustan distintos algoritmos de *Machine Learning* y arquitecturas neuronales, siguiendo configuraciones comparables.
- *Evaluación*: centrada en medir el rendimiento de los modelos con métricas estándar de clasificación, a fin de estimar su capacidad de generalización.
- *Combinación de resultados*: se construyen esquemas de *ensamblado* para integrar predicciones y potenciar el rendimiento conjunto frente a los modelos individuales.

A continuación, se detallan de manera sistemática estas fases metodológicas que conformaron el desarrollo del estudio.

Preprocesamiento de datos

El preprocesamiento constituyó una etapa fundamental para garantizar que la información de entrada resultara adecuada para el entrenamiento de los modelos. Esta fase abarcó tanto la selección de características de mercado como la incorporación de variables derivadas del análisis de sentimientos aplicado a las noticias financieras.

En primer lugar, dado el elevado número de variables obtenidas a partir de las series históricas de precios descargadas con *yfinance* y de los indicadores técnicos de *trading* generados, se aplicó para cada compañía un *Random Forest* inicial con el objetivo de estimar la importancia de las variables (*feature importance*) y reducir la dimensionalidad del conjunto de datos. A partir de este procedimiento se seleccionaron las 20 características técnicas más relevantes. Este enfoque resulta habitual en la literatura, ya que proporciona un criterio práctico para priorizar variables explicativas; sin embargo, presenta limitaciones conocidas. En particular, la importancia de variables en los modelos de *Random Forest* puede verse sesgada hacia predictores altamente correlacionados, lo que implica que parte de la información seleccionada podría ser redundante y no aportar valor adicional al modelo, afectando potencialmente a la calidad de las predicciones.

En segundo lugar, se abordó el análisis de sentimientos a partir de las noticias del *dataset*. Cada una incluía varias columnas de resúmenes generados automáticamente mediante diferentes algoritmos (LSA, LexRank, TextRank y Luhn), además del título original del artículo. Dado que algunos resúmenes aparecían vacíos o contenían mensajes irrelevantes (por ejemplo, comenzaban con *Click to get this free report*), se estableció un criterio de prioridad para seleccionar el texto más adecuado. En concreto, para cada noticia, se tomó en primer lugar el resumen generado con LSA; y, en caso de no estar disponible o ser inválido, se recurrió sucesivamente a LexRank, TextRank; y, por último, a Luhn. El resumen seleccionado se concatenó con el título de la noticia, de manera que cada registro dispusiera de un texto completo y consistente para el análisis.

Sobre este texto obtenido se aplicó el modelo *FinBERT* mencionado en la sección anterior. Este modelo no solo clasifica cada noticia en categorías positiva, negativa o neutra, sino que también devuelve probabilidades asociadas a cada clase y una puntuación continua de sentimiento.

Finalmente, con el objetivo de generar variables comparables con las series temporales de precios, la información de sentimiento de noticia se agrupó por fechas, obteniendo un resumen de las variables con una frecuencia diaria. Para ello, las predicciones de *FinBERT* se agruparon por fecha y se calcularon estadísticas agregadas como la media, la desviación estándar, los valores máximo y mínimo, así como los recuentos de noticias positivas y negativas. Estas variables resumen permitieron capturar de manera sintética el tono general de la información disponible en el mercado en cada jornada, facilitando su integración en los modelos predictivos junto con las características de mercado.

Entrenamiento de modelos

Para el desarrollo de los modelos se contemplaron dos configuraciones de variables de entrada: únicamente variables procedentes del análisis de sentimientos; y una combinación de las veinte variables técnicas seleccionadas a partir de la importancia de variables del *Random Forest* inicial, junto con las variables de análisis de sentimientos.

A partir de estas configuraciones se desarrollaron dos enfoques predictivos diferenciados:

1. *Clasificación binaria.*

El objetivo fue predecir si el precio de cierre del día siguiente aumentaría o disminuiría. El *target* se definió como:

$$y_t = \begin{cases} 1 & \text{si } P_{t+1} > P_t, \\ 0 & \text{en caso contrario.} \end{cases} \quad (3.1)$$

Los modelos recurrentes se entrenaron con función de pérdida de entropía cruzada binaria y se aplicó un umbral de 0.5 sobre la salida sigmoide para determinar la clase predicha. Los modelos de *Random Forest* generaron directamente probabilidades asociadas a cada clase.

2. **Regresión sobre rentabilidad logarítmica.**

En este enfoque se modeló la magnitud del cambio de precio mediante la rentabilidad logarítmica:

$$r_t = \ln \left(\frac{P_{t+1}}{P_t} \right) . \quad (3.2)$$

Los mismos modelos se entrenaron ahora como regresores con salida lineal. La predicción continua de r_t se transformó posteriormente en una señal de compra o venta aplicando la regla de signo:

$$\hat{y}_t = \begin{cases} 1 & \text{si } \hat{r}_t > 0, \\ 0 & \text{si } \hat{r}_t \leq 0. \end{cases} \quad (3.3)$$

De este modo, los modelos de regresión se evaluaron también en términos de su capacidad para inducir una clasificación binaria coherente con la dirección del movimiento del precio.

En cuanto a los modelos considerados, se incluyeron las siguientes aproximaciones:

- **Redes Neuronales Recurrentes (LSTM y GRU).**

La longitud de la ventana temporal se fijó en 60 días, tomando como referencia trabajos recientes en predicción financiera que mostraron que este horizonte permitía capturar dependencias a corto y medio plazo sin introducir un exceso de ruido ni incrementar de forma desproporcionada la complejidad del modelo (Narayana et al., 2025). De esta manera, se garantizó un equilibrio entre la disponibilidad de información histórica y la capacidad de generalización del modelo.

La arquitectura se diseñó con dos capas recurrentes de 128 y 64 unidades respectivamente, siguiendo la práctica habitual en la literatura para capturar patrones secuenciales de distinta granularidad. Este número de unidades se consideró suficiente para modelar relaciones no lineales en los datos sin generar un sobreajuste excesivo. Tras cada capa recurrente se aplicó un *dropout* del 30 %, con el fin de reducir la varianza y mejorar la robustez frente al ruido característico de las series financieras. Posteriormente, se añadió una capa

densa de 32 neuronas con activación ReLU, destinada a combinar las representaciones aprendidas y a facilitar la extracción de características de mayor nivel.

En el caso de la clasificación binaria, la capa de salida consistió en una única neurona con activación sigmoide, optimizada mediante entropía cruzada binaria. Para la regresión, la capa de salida fue una neurona con activación lineal, optimizada con error cuadrático medio (*MSE*). Como métricas complementarias se emplearon *MAE* y *RMSE*.

En ambos casos se utilizó el optimizador *Adam* con una tasa de aprendizaje de 0.001, valor ampliamente empleado en la literatura por ofrecer un buen equilibrio entre velocidad de convergencia y estabilidad en redes neuronales profundas. Finalmente, se aplicó *early stopping* con un máximo de 20 épocas, con el propósito de prevenir el sobreajuste y detener el entrenamiento cuando no se observaban mejoras significativas en el conjunto de validación.

- ***Random Forest.***

La estrategia de optimización incluyó una primera búsqueda amplia de hiperparámetros mediante *Random Search*, seguida de un *Grid Search* restringido a la vecindad de los parámetros más prometedores, con el fin de refinar el ajuste del modelo.

- ***Modelos de boosting.***

Se realizaron pruebas preliminares con algoritmos como *XGBoost* y *AdaBoost*. Los resultados obtenidos mostraron un comportamiento muy similar al de *Random Forest*, sin mejoras significativas en las métricas de evaluación. Por otro lado, el entrenamiento de estos modelos resultó considerablemente más costoso en términos de tiempo de ejecución, lo que incrementaba de manera notable la duración de los *notebooks*. Además, el objetivo principal de este trabajo fue evaluar en profundidad el desempeño de redes recurrentes como *LSTM* y *GRU*, comparándolas con modelos basados en árboles. En este contexto, la inclusión de *Random Forest* como referencia no secuencial se consideró suficiente. Por tanto, se optó por descartar los modelos de *boosting* en el análisis final.

En consecuencia, al considerar todas las combinaciones posibles de configuraciones, se entrenaron un total 12 modelos para cada una de las compañías.

Evaluación de modelos

Para la validación se utilizó una partición temporal: los datos anteriores a 2022 se destinaron al entrenamiento, el año 2022 a la validación y el año 2023 al conjunto de prueba. Por tanto, la evaluación final de los modelos se llevó a cabo sobre el conjunto de prueba correspondiente a 2023 completo.

En la evaluación se consideraron varias métricas de clasificación. En primer lugar, se empleó la *Accuracy*, que mide el porcentaje de predicciones correctas sobre el total de casos:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (3.4)$$

donde TP representa los verdaderos positivos (subidas correctamente predichas), TN los verdaderos negativos (bajadas correctamente predichas), FP los falsos positivos (predicciones de subida que en realidad fueron bajada) y FN los falsos negativos (predicciones de bajada que en realidad fueron subida).

De manera complementaria, se utilizó la *Precisión*, definida como:

$$\text{Precisión} = \frac{TP}{TP + FP}. \quad (3.5)$$

Esta métrica indica la proporción de predicciones de subida que realmente resultaron correctas. En el contexto del *trading*, la *Precisión* es relevante porque una señal positiva suele traducirse en una operación de compra. Adicionalmente, en la parte de los *ensamblados* se analizó también de forma separada para cada clase, con el fin de evaluar el comportamiento del modelo tanto en predicciones de subida como de bajada.

Finalmente, la última métrica en la que se puso foco a lo largo de este trabajo fue la *Exhaustividad (Recall)*, calculada como:

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (3.6)$$

Esto refleja la capacidad del modelo para identificar todas las subidas reales. En términos de inversión, un alto *Recall* significa que el sistema detecta la mayoría de las oportunidades de compra presentes en el mercado, lo que resulta fundamental para no omitir movimientos relevantes del precio.

Combinación de modelos

Con el objetivo de mejorar la robustez de las predicciones, se exploraron distintas estrategias de combinación de modelos (*ensamblados*). Inicialmente se consideraron enfoques de *hard voting* (votación mayoritaria no ponderada), esquemas de *soft voting* basados en el promedio de probabilidades y aproximaciones inspiradas en *stacking*. Sin embargo, en la práctica estos métodos presentaron limitaciones: los sistemas de votación directa resultaban demasiado simples y tendían a favorecer modelos con alta *Accuracy* pero muy baja *Precisión*, lo que reducía el rendimiento en la detección de oportunidades de compra, aspecto clave en este trabajo.

Para superar este problema, se decidió utilizar directamente las predicciones binarias $\{0, 1\}$ de cada modelo en lugar de sus probabilidades. La motivación fue que las probabilidades estimadas no eran comparables entre clasificadores: en algunos

casos se concentraban en torno a 0,5, mientras que en otros adoptaban distribuciones mucho más extremas. Esta falta de calibración provocaba que el *soft voting* degradara de forma notable la precisión global. Al restringirse a la predicción final (comprar/no comprar), se homogeneizó el espacio de decisión y se incrementó la estabilidad del *ensamblado*.

Además, se introdujo un esquema de votación ponderada en el que los pesos de cada modelo no se fijaron de manera arbitraria, sino que se derivaron automáticamente a partir de su desempeño en el conjunto de validación. Para ello, se definió una métrica compuesta que combina *Accuracy* y *Precisión*, con el fin de favorecer explícitamente a los modelos con mejor capacidad de predicción positiva:

$$s_i = \beta_{\text{acc}} \cdot \text{Accuracy}_i + \beta_{\text{prec}} \cdot \text{Precisión}_i , \quad (3.7)$$

donde s_i es la puntuación bruta del modelo i , y los coeficientes β_{acc} y β_{prec} permiten ajustar la importancia relativa de ambas métricas. En particular, para el *ensamblado* priorizando la precisión se tomó $\beta_{\text{prec}} = 0,6$ y $\beta_{\text{acc}} = 0,4$, mientras que para el orientado a la exactitud global se fijó $\beta_{\text{acc}} = 0,9$ y $\beta_{\text{prec}} = 0,1$.

A continuación, estas puntuaciones se transformaron en pesos normalizados mediante una función *softmax*, lo que asegura que los modelos con mejor desempeño reciban mayor influencia, sin excluir por completo al resto:

$$w_i = \frac{\exp(\alpha \cdot (s_i - \bar{s}))}{\sum_j \exp(\alpha \cdot (s_j - \bar{s}))} , \quad (3.8)$$

donde α controla la “temperatura” del *softmax* (a mayor valor, más concentrados quedan los pesos en los mejores modelos) y \bar{s} es la media de los s_i .

Finalmente, la predicción del *ensamblado* en cada instante t se obtuvo como una media ponderada de las predicciones binarias de los modelos considerados:

$$\hat{y}_{\text{ensemble},t} = I\left(\sum_{i \in M_t} w_i \cdot \hat{y}_{i,t} \geq 0,5\right) , \quad (3.9)$$

donde M_t representa el conjunto de modelos con predicción disponible en t .

3.5. Resultados

En esta sección se presentan los resultados obtenidos a partir de los distintos modelos aplicados a cada una de las compañías seleccionadas. Además, antes de analizar las tablas con las métricas finales, se llevó a cabo un análisis exploratorio del histórico de los precios de cierre de cada empresa con el propósito de contextualizar los conjuntos de entrenamiento y prueba. Este análisis permitió disponer de una visión preliminar del comportamiento de cada compañía, lo que facilitó la interpretación posterior de los modelos y la identificación de patrones coherentes con los movimientos observados en el mercado.

Gilead Sciences (GILD)

La Figura 3.1 recoge la evolución histórica del precio de cierre de *GILD*. Entre 2012 y 2015, la acción experimentó un notable crecimiento, impulsado por el éxito comercial de los tratamientos contra la hepatitis C. Tras ese repunte, la cotización se estabilizó y mostró oscilaciones más moderadas en los años siguientes. Un episodio destacado se produjo en 2020: a diferencia de muchas compañías que sufrieron el impacto de la crisis del COVID-19, *Gilead Sciences* registró un repunte gracias al antiviral *Veklury (remdesivir)*, empleado en el tratamiento de la enfermedad, lo que reforzó sus ingresos en plena pandemia.

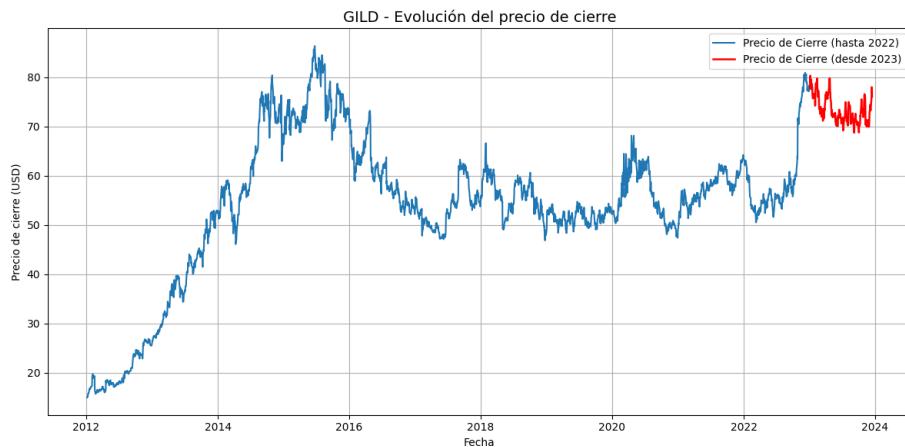


Figura 3.1: Histórico del precio de cierre - GILD

En relación con el año 2023 (Figura 3.2), el precio de la acción no mostró una tendencia clara, sino que alternó subidas y bajadas frecuentes, reflejando la incertidumbre del mercado respecto a la compañía. Esta evolución se relacionó con varios factores: la reducción de los ingresos procedentes de *Veklury* tras el retroceso de la pandemia, el crecimiento en áreas estratégicas como las terapias oncológicas y los

tratamientos celulares, y la publicación de avances en ensayos clínicos, como los resultados positivos de un fármaco preventivo contra el VIH, que generaron episodios puntuales de optimismo en la cotización.

Por otro lado, el balance de clases del conjunto de datos (Figura 3.3) mostró una distribución equilibrada entre jornadas de subida y de bajada. Este aspecto resulta relevante; ya que, a pesar de la tendencia alcista observada en una parte del histórico, la proporción entre clases se mantuvo equilibrada.

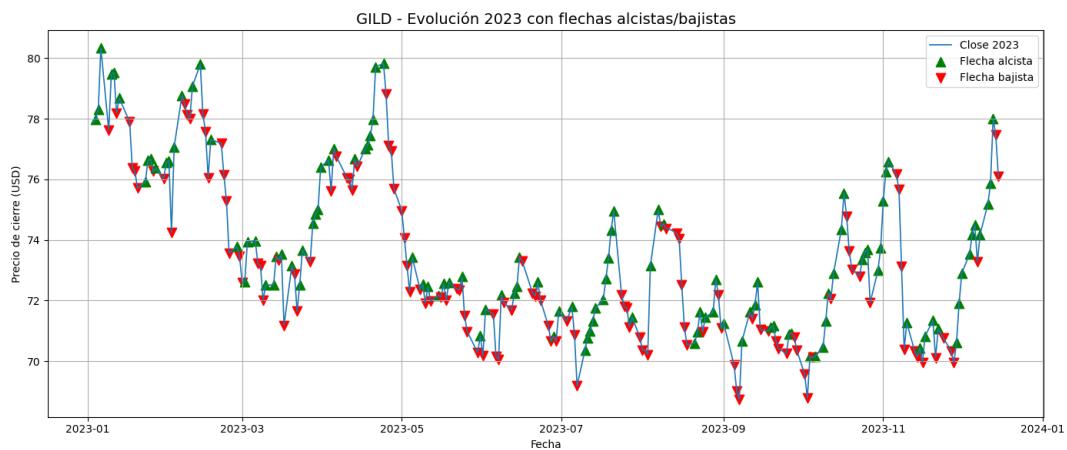


Figura 3.2: Subidas y bajadas 2023 - GILD

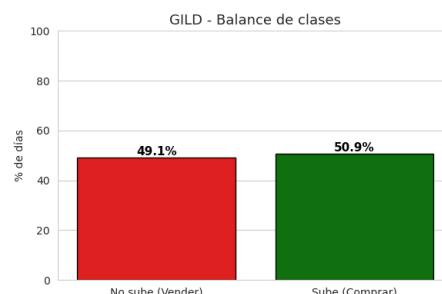


Figura 3.3: Balance de clases - GILD

Una vez contextualizada la evolución de la compañía y analizada la distribución de clases, se presentan los resultados de los modelos predictivos, tanto en el enfoque de clasificación binaria como en el de regresión logarítmica, junto con los ensamblados derivados.

En el caso de *Gilead Sciences*, los resultados ponen de manifiesto varios patrones de interés. En primer lugar, los modelos de regresión logarítmica (Tabla 3.2) alcanzaron valores de *Accuracy* aceptables, en ocasiones superiores a los de la clasificación binaria (Tabla 3.1), aunque con niveles de *Precisión* muy bajos (inferiores

al 30 %). Esto indica que, si bien identificaron correctamente muchos días de *No comprar*, fallaron con frecuencia al señalar oportunidades de compra, lo que limita su aplicabilidad práctica.

Al incorporar estos modelos en *ensamblados* basados en la *Accuracy*, se observó un efecto claro: la predicción de la clase *No comprar* mejoró de manera significativa, alcanzando una precisión superior al 75 % (Tabla 3.3). Sin embargo, esta mejora se produjo a costa de un descenso en el rendimiento de la clase *Comprar*, de modo que el sistema resultó más adecuado como herramienta de apoyo para descartar operaciones que como generador de señales de entrada.

En cambio, los *ensamblados* ponderados por *Precisión* (Tabla 3.4) ofrecieron un comportamiento distinto. En este caso, la clase *Comprar* alcanzó una precisión cercana al 59 % y un *Recall* en torno al 74 %, lo que supuso un mejor equilibrio entre ambas clases. No obstante, esta mejora en la detección de oportunidades de compra se acompañó de una pérdida de fiabilidad en la clase *No comprar*. En consecuencia, este enfoque puede resultar útil en escenarios donde se priorice la identificación de señales de entrada, aunque ello implique asumir un mayor riesgo de falsos positivos.

En conjunto, los resultados asociados a *Gilead Sciences* reflejaron la dificultad de obtener un modelo que funcionara de manera equilibrada en ambas clases. No obstante, también mostraron que los *ensamblados* pueden adaptarse a distintos objetivos según la naturaleza de los modelos con los que trabajan: por un lado, pueden premiar estrategias más conservadoras, evitando compras poco fiables; y, por otro lado, pueden tratar de acertar un mayor número de operaciones de compra a cambio de una menor exactitud global.

Cuadro 3.1: Resultados de Clasificación Binaria - GILD

Modelo	Accuracy	Precisión	Recall	F1-Score	Peso Acc.	Peso Prec.
Combined LSTM	0.6094	0.6897	0.5556	0.6154	0.120	0.293
Combined GRU	0.6064	0.6667	0.4583	0.5432	0.116	0.252
Sentiment RF	0.5403	0.5556	0.6154	0.5839	-	0.099
Sentiment GRU	0.5323	0.5285	1.0000	0.6915	-	0.082
Sentiment LSTM	0.5161	0.5214	0.9385	0.6703	-	0.073
Combined RF	0.4597	0.4773	0.3231	0.3853	-	-

Cuadro 3.2: Resultados de Regresión Logarítmica - GILD

Modelo	Accuracy	Precisión	Recall	F1-Score	Peso Acc.	Peso Prec.
Reg Sentiment RF	0.6276	0.2581	0.2712	0.2645	0.144	-
Reg Combined LSTM	0.6237	0.3778	0.7083	0.4928	0.138	-
Reg Combined GRU	0.5873	0.3200	0.4706	0.3810	0.096	-
Reg Combined RF	0.5858	0.2500	0.3390	0.2878	-	-
Reg Sentiment GRU	0.5714	0.3077	0.4706	0.3721	-	-
Reg Sentiment LSTM	0.4286	0.2564	0.5882	0.3571	-	-

Cuadro 3.3: Resultados del Ensemble por Accuracy - GILD

Clase	Precisión	Recall	F1-Score
No Comprar	0.7740	0.7611	0.7675
Comprar	0.3065	0.3220	0.3140
Accuracy			0.6527

Cuadro 3.4: Resultados del Ensemble por Precisión - GILD

Clase	Precisión	Recall	F1-Score
No Comprar	0.6047	0.4407	0.5098
Comprar	0.5926	0.7385	0.6575
Accuracy			0.5968

Coca-Cola (KO)

La evolución histórica del precio de cierre de Coca-Cola desde 2012 muestra una trayectoria marcadamente alcista (Figura 3.4). Durante más de una década, la acción mantuvo un crecimiento sostenido, con la excepción de 2020, cuando sufrió una caída significativa a raíz de la crisis del COVID-19. La recuperación, sin embargo, fue rápida, y a partir de 2021 el valor volvió a situarse en niveles elevados. Desde 2022, la cotización permaneció en la zona alta, aunque se produjeron dos descensos relevantes: el primero a mediados de 2022, vinculado a la presión inflacionaria y al encarecimiento de costes en el sector de consumo, y el segundo en octubre de 2023, asociado a previsiones menos optimistas y a una actitud más cautelosa por parte de los inversores. Esta última caída se refleja también en la Figura 3.5, donde no se aprecia una tendencia definida a lo largo del año.

En cuanto a la distribución de clases (Figura 3.6), del mismo modo que en la compañía anterior, el conjunto de datos presenta un equilibrio razonable entre días al alza y a la baja. A pesar de la tendencia general positiva en el largo plazo, no se detecta un sesgo marcado.

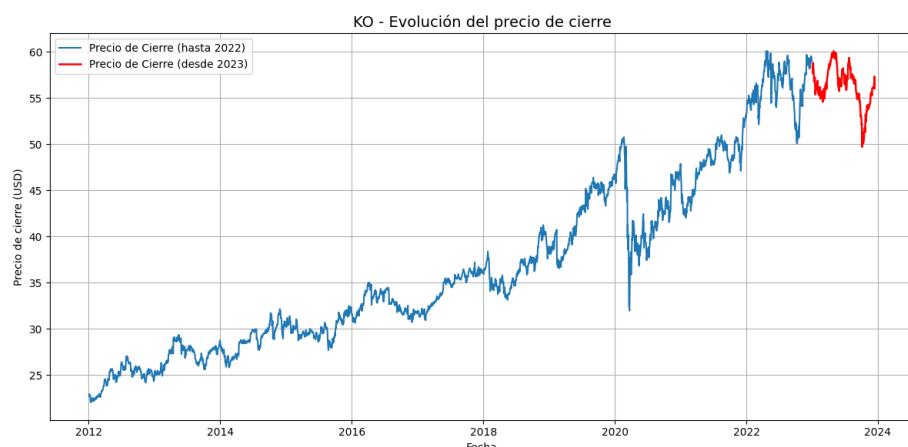


Figura 3.4: Histórico del precio de cierre - KO

Al pasar al análisis predictivo, los modelos de clasificación binaria (Tabla 3.5) ofrecieron rendimientos modestos: los valores de *Accuracy* se situaron en torno al 50 % y no se observaron diferencias relevantes entre enfoques tradicionales como *Random Forest* y otros más actuales como *LSTM* o *GRU*.

Los resultados de la regresión logarítmica (Tabla 3.6) siguieron un patrón similar al observado en *Gilead Sciences*: la *Precisión* en la clase *Comprar* resultó muy baja (por debajo del 35 %), aunque en algunos casos se alcanzaron niveles de *Accuracy* superiores al 60 %. En consecuencia, los modelos se mostraron más fiables a la hora de anticipar los días en que no convenía comprar, pero tuvieron dificultades para identificar oportunidades de entrada.

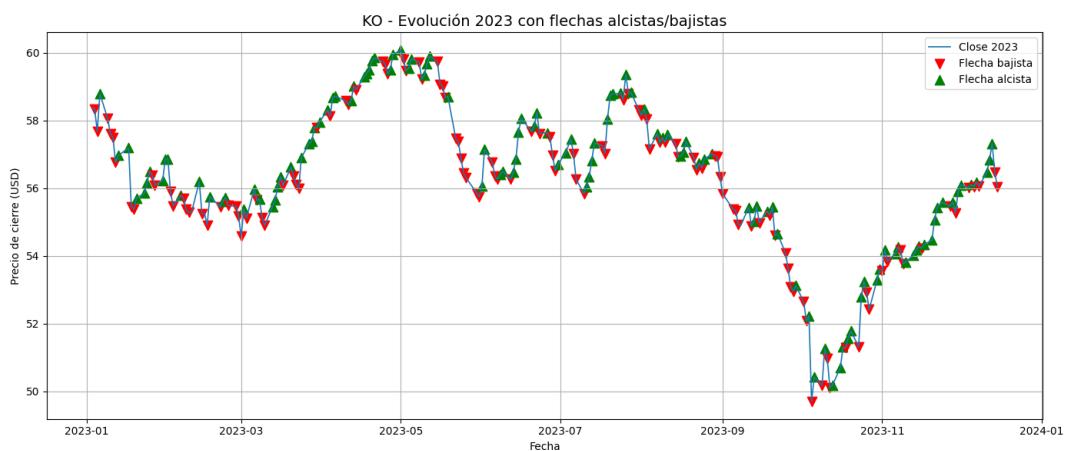


Figura 3.5: Subidas y bajadas 2023 - KO

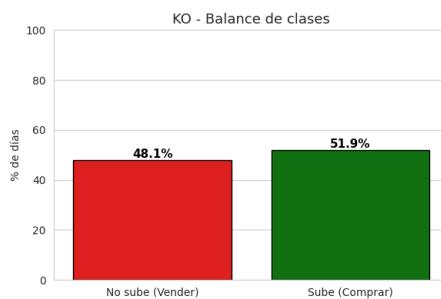


Figura 3.6: Balance de clases - KO

Los *ensamblados* confirmaron esta tendencia, aunque con matices. El ponderado por *Accuracy* alcanzó una precisión elevada en la clase *No comprar* (76 %), pero sacrificó notablemente la clase *Comprar*, que se redujo a valores inferiores al 30 % (Tabla 3.7). En cambio, el *ensamblado* basado en *Precisión* ofreció un reparto algo más equilibrado entre ambas clases, aunque el rendimiento global fue bajo: la *Accuracy* total apenas llegó al 45 % (Tabla 3.8). Este último resultado resulta particularmente revelador, ya que sugiere que, en este caso, los modelos no lograron capturar patrones útiles en la dinámica de la acción y que, de hecho, una estrategia opuesta a la señal emitida habría obtenido mejores resultados, lo cual carece de valor práctico.

Cuadro 3.5: **Resultados de Clasificación Binaria - KO**

Modelo	Accuracy	Precisión	Recall	F1-Score	Peso Acc.	Peso Prec.
Combined RF	0.5160	0.5280	0.5841	0.5546	0.091	0.160
Sentiment GRU	0.5068	0.5170	0.6726	0.5846	0.083	0.144
Sentiment LSTM	0.4932	0.5119	0.3805	0.4365	-	0.133
Combined GRU	0.4709	0.5043	0.5743	0.5370	-	0.116
Sentiment RF	0.4658	0.4821	0.4779	0.4800	-	0.099
Combined LSTM	0.4591	0.4835	0.5301	0.5057	-	-

Cuadro 3.6: **Resultados de Regresión Logarítmica - KO**

Modelo	Accuracy	Precisión	Recall	F1-Score	Peso Acc.	Peso Prec.
Reg Sentiment RF	0.6067	0.2571	0.3000	0.2769	0.158	-
Reg Sentiment GRU	0.5669	0.2963	0.3478	0.3200	0.115	-
Reg Sentiment LSTM	0.5605	0.3506	0.5870	0.4390	0.114	-
Reg Combined LSTM	0.5241	0.2532	0.4000	0.3101	-	-
Reg Combined GRU	0.5159	0.2727	0.3913	0.3214	-	-
Reg Combined RF	0.4895	0.2281	0.4333	0.2989	-	-

Cuadro 3.7: **Resultados del Ensemble por Accuracy - KO**

Clase	Precisión	Recall	F1-Score
No Comprar	0.7634	0.5525	0.6410
Comprar	0.2636	0.4833	0.3412
Accuracy			0.5353

Cuadro 3.8: **Resultados del Ensemble por Precisión - KO**

Clase	Precisión	Recall	F1-Score
No Comprar	0.4222	0.3585	0.3878
Comprar	0.4729	0.5398	0.5041
Accuracy			0.4521

Wells Fargo (WFC)

El comportamiento histórico de los precios de cierre de *Wells Fargo* (Figura 3.7) refleja una evolución menos lineal que la de otras compañías analizadas. Tras una tendencia alcista inicial a partir de 2012, la acción se estabilizó alrededor de 2015 y, desde entonces, presentó oscilaciones marcadas con episodios de caídas y posteriores recuperaciones. La caída más severa se produjo en 2020, en plena crisis del COVID-19, que impactó de manera directa en el sector bancario. A diferencia de otras entidades que lograron recuperar sus niveles previos con relativa rapidez, *Wells Fargo* tardó más en hacerlo, afectada además por problemas de reputación y sanciones regulatorias que arrastraba desde años anteriores. En 2022, el título alcanzó un máximo a comienzos de año, pero perdió fuerza posteriormente en un contexto de subidas de tipos de interés e incertidumbre macroeconómica. Finalmente, en 2023, la cotización se mantuvo en un rango estable cercano a los 40 dólares, sin mostrar una tendencia clara (Figura 3.8).

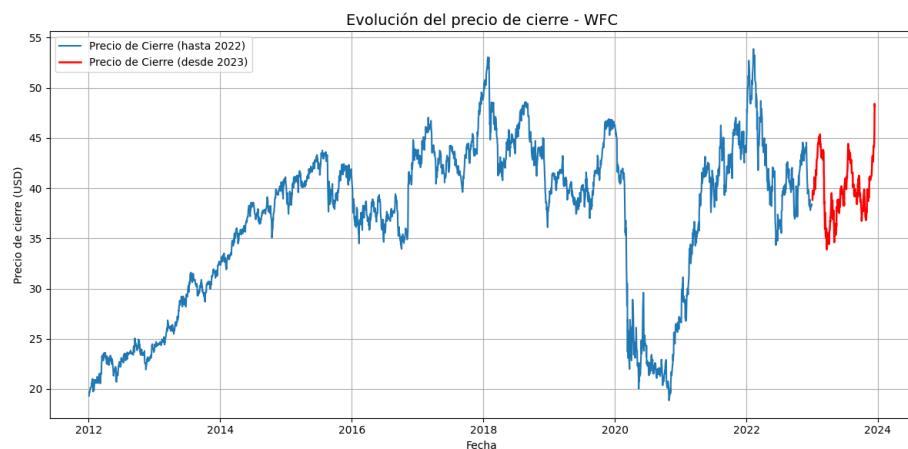


Figura 3.7: Histórico del precio de cierre - WFC

En lo referente al análisis predictivo, los resultados de la clasificación binaria (Tabla 3.9) mostraron valores de *Accuracy* cercanos al 50 %, lo que limita su fiabilidad como métrica principal. Sin embargo, al atender a la *Precisión* y al *Recall*, se apreciaron comportamientos más matizados. Destacó especialmente el modelo *Sentiment GRU*, que alcanzó una precisión de 0.5515 y un *Recall* de 0.8198, mostrando una buena capacidad para identificar días de subida a costa de un número reducido de falsos positivos. Debido al bajo rendimiento en términos de *Accuracy*, únicamente el modelo *Combined RF* fue incluido en el ensemble asociado a esta métrica, lo que derivó en un fuerte sesgo hacia la clase *No comprar* y un rendimiento limitado para la clase *Comprar* (Tabla 3.11).

Los modelos de regresión logarítmica (Tabla 3.10) mostraron un patrón similar al observado en *Coca-Cola*: el *Accuracy* alcanzó valores razonables (en torno al 60 % en el mejor caso), pero la precisión en la clase *Comprar* resultó insuficiente. El

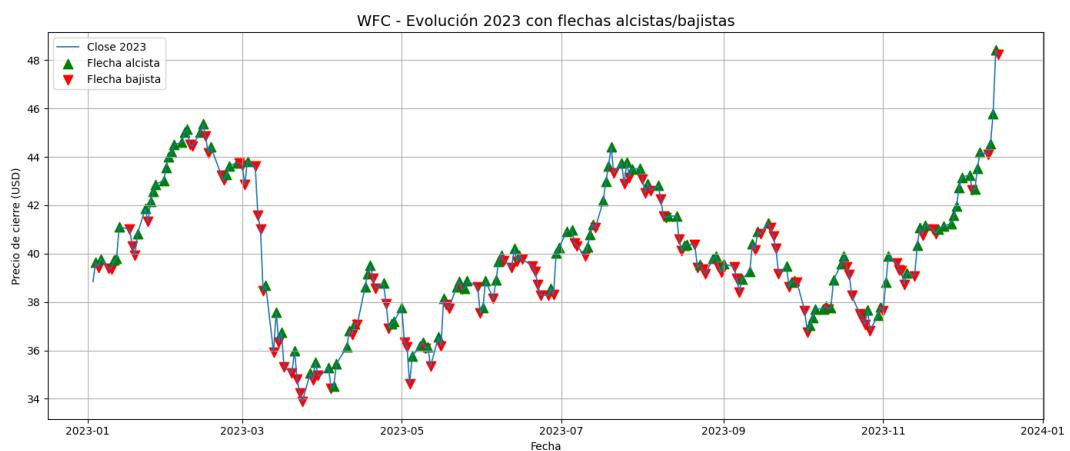


Figura 3.8: Subidas y bajadas 2023 - WFC

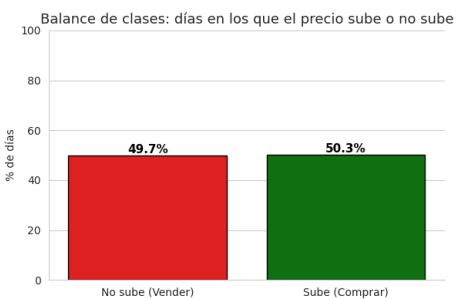


Figura 3.9: Balance de clases - WFC

ensamblado ponderado por *Precisión* (Tabla 3.12) mejoró ligeramente el equilibrio entre clases, con una precisión de 0.5625 y un *Recall* de 0.5676 en la clase *Comprar*, aunque el rendimiento general continuó siendo modesto. En este escenario concreto, el ensemble por precisión aportó poco valor práctico, y el modelo individual *Sentiment GRU* se presentó como la opción más prometedora para anticipar oportunidades de compra.

Cuadro 3.9: Resultados de Clasificación Binaria - WFC

Modelo	Accuracy	Precisión	Recall	F1-Score	Peso Acc.	Peso Prec.
Combined RF	0.5459	0.6222	0.5045	0.5572	0.116	0.260
Sentiment GRU	0.5204	0.5515	0.8198	0.6594	-	0.153
Sentiment RF	0.4847	0.5510	0.4865	0.5167	-	0.133
Sentiment LSTM	0.4745	0.5290	0.6577	0.5863	-	0.111
Combined GRU	0.4458	0.5051	0.5376	0.5208	-	0.086
Combined LSTM	0.4338	0.5065	0.5000	0.5032	-	-

Cuadro 3.10: Resultados de Regresión Logarítmica - WFC

Modelo	Accuracy	Precisión	Recall	F1-Score	Peso Acc.	Peso Prec.
Reg Sentiment RF	0.6151	0.2090	0.2642	0.2333	0.142	-
Reg Combined LSTM	0.5854	0.2609	0.5143	0.3462	0.115	-
Reg Sentiment GRU	0.5821	0.2692	0.4375	0.3333	0.112	-
Reg Sentiment LSTM	0.5672	0.2833	0.5312	0.3696	0.100	-
Reg Combined GRU	0.5373	0.2000	0.3125	0.2439	-	-
Reg Combined RF	0.5272	0.1875	0.3396	0.2416	-	-

Cuadro 3.11: Resultados del Ensemble por Accuracy - WFC

Clase	Precisión	Recall	F1-Score
No Comprar	0.7818	0.6898	0.7330
Comprar	0.2368	0.3333	0.2769
Accuracy			0.6100

Cuadro 3.12: Resultados del Ensemble por Precisión - WFC

Clase	Precisión	Recall	F1-Score
No Comprar	0.4286	0.4235	0.4260
Comprar	0.5625	0.5676	0.5650
Accuracy			0.5051

Merck & Co. (MRK)

La evolución del precio de cierre de *Merck & Co.* (Figura 3.10) mostró un crecimiento sostenido durante la última década, con incrementos prácticamente continuos año tras año. El impacto de la pandemia en 2020 fue más moderado que en otras compañías del sector, como *Gilead Sciences*, ya que el foco de *Merck* en innovación oncológica redujo su exposición directa a tratamientos vinculados al COVID-19. Tras esta breve corrección, la acción retomó su trayectoria alcista, alcanzando máximos históricos a finales de 2022. Durante 2023, sin embargo, la cotización adoptó un comportamiento más lateral, caracterizado por oscilaciones sin una dirección clara (Figura 3.11).

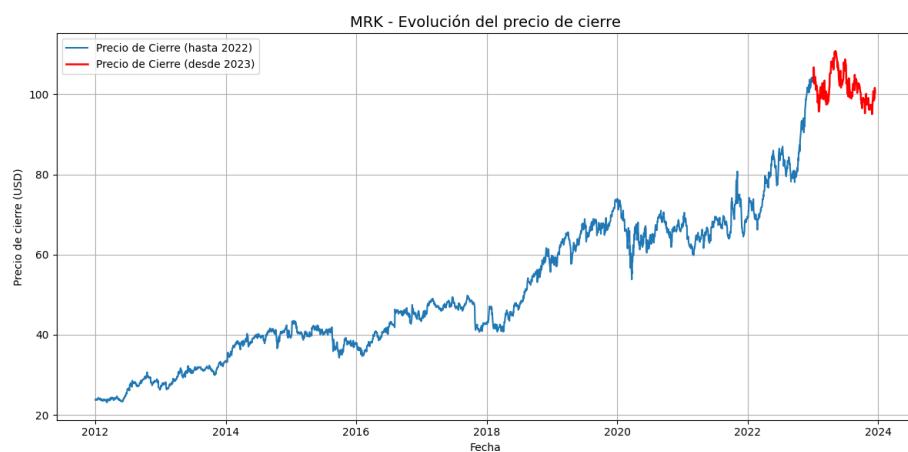


Figura 3.10: Histórico del precio de cierre - MRK

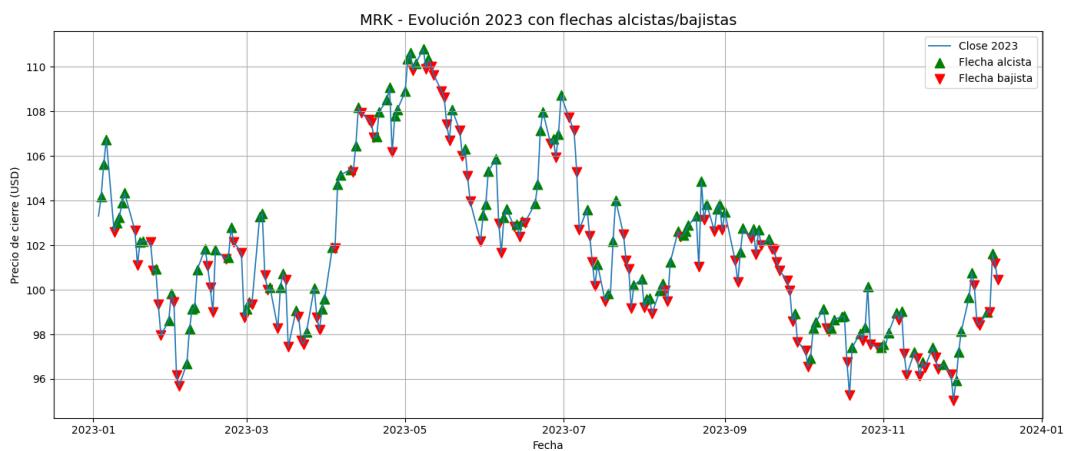
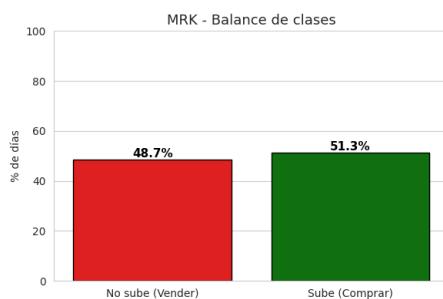


Figura 3.11: Subidas y bajadas 2023 - MRK

Figura 3.12: Balance de clases - MRK

En el plano predictivo, los resultados fueron poco alentadores. Los modelos de clasificación binaria (Tabla 3.13) apenas superaron el 50 % de *Accuracy* en los mejores casos, e incluso algunos descendieron por debajo de ese umbral. El modelo *Reg Sentiment LSTM*, con un *Accuracy* del 41 %, constituye un ejemplo extremo de esta falta de fiabilidad, hasta el punto de que habría sido más efectivo seguir la estrategia contraria a sus predicciones.

Los modelos de regresión logarítmica (Tabla 3.14) lograron valores de *Accuracy* ligeramente superiores (alrededor del 59 % en el mejor escenario), pero acompañados de niveles muy bajos de *Precisión* en la clase *Comprar*. Esto limita su aplicación práctica, pues predicen razonablemente bien cuándo no conviene entrar al mercado, pero fallan de forma reiterada en la detección de oportunidades de compra.

Los *ensamblados* repitieron un patrón ya observado en otras compañías. Por un lado, el ponderado por *Accuracy* (Tabla 3.15) alcanzó buenos resultados en la clase *No comprar*, con una precisión cercana al 78 %, pero sacrificó casi por completo la clase *Comprar*, cuyos valores se situaron entre el 20 y el 30 %. Por otra parte, el basado en *Precisión* (Tabla 3.16) logró un mayor equilibrio entre clases, aunque su rendimiento global se mantuvo alrededor del 50 %. Este resultado sugiere que, en el caso de *Merck*, los modelos no fueron capaces de identificar patrones de utilidad real y que las predicciones se aproximaron a un comportamiento aleatorio.

Cuadro 3.13: Resultados de Clasificación Binaria - MRK

Modelo	Accuracy	Precisión	Recall	F1-Score	Peso Acc.	Peso Prec.
Sentiment GRU	0.5075	0.5378	0.5981	0.5664	0.072	0.149
Combined GRU	0.5030	0.5532	0.2921	0.3824	-	0.160
Sentiment RF	0.5025	0.5444	0.4579	0.4975	-	0.152
Sentiment LSTM	0.4925	0.5234	0.6262	0.5702	-	0.128
Combined LSTM	0.4892	0.5455	0.1644	0.2526	-	0.145
Combined RF	0.4724	0.5088	0.5421	0.5249	-	-

Cuadro 3.14: Resultados de Regresión Logarítmica - MRK

Modelo	Accuracy	Precisión	Recall	F1-Score	Peso Acc.	Peso Prec.
Reg Combined LSTM	0.5952	0.2676	0.5429	0.3585	0.173	-
Reg Combined GRU	0.5652	0.2000	0.3333	0.2500	0.128	-
Reg Sentiment RF	0.5607	0.2360	0.3621	0.2857	0.122	-
Reg Sentiment GRU	0.5362	0.2258	0.4667	0.3043	0.096	-
Reg Combined RF	0.5063	0.2368	0.4655	0.3140	-	-
Reg Sentiment LSTM	0.4130	0.1852	0.5000	0.2703	-	-

Cuadro 3.15: Resultados del Ensemble por Accuracy - MRK

Clase	Precisión	Recall	F1-Score
No Comprar	0.7815	0.6310	0.6982
Comprar	0.2247	0.3774	0.2817
Accuracy			0.5750

Cuadro 3.16: Resultados del Ensemble por Precisión - MRK

Clase	Precisión	Recall	F1-Score
No Comprar	0.4561	0.5652	0.5049
Comprar	0.5294	0.4206	0.4688
Accuracy			0.4874

Invesco QQQ Trust (QQQ)

El precio de cierre de *Invesco QQQ Trust* (Figura 3.13) reflejó durante la última década una tendencia marcadamente alcista, sustentada en el buen desempeño de las principales compañías tecnológicas que integran el índice. Este comportamiento se interrumpió en 2022 con una corrección significativa, en línea con la caída general del sector tecnológico y el impacto de las subidas de tipos de interés. A lo largo de 2023, el valor alcanzó un mínimo y desde entonces inició una recuperación sostenida, configurando de nuevo una trayectoria ascendente (Figura 3.14).

En cuanto al balance de clases de esta compañía (Figura 3.15), se observó un ligero sesgo positivo en el conjunto completo de datos, con un 55,4 % de días al alza frente a un 44,6 % a la baja. Aunque esta diferencia no supone una descompensación extrema entre ambas categorías, sí implica que un modelo trivial que predijese siempre *Compra* podría alcanzar por sí solo una *Accuracy* entorno al 55,4 %. Este valor puede resultar engañoso si no se contrasta con otras métricas adicionales que reflejen de forma más precisa la capacidad del modelo para discriminar entre clases.

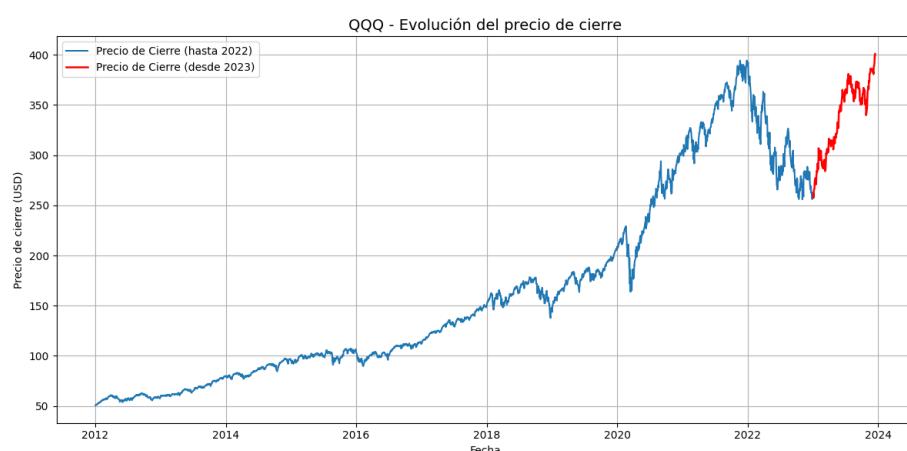


Figura 3.13: Histórico del precio de cierre - QQQ

El análisis de los modelos predictivos dejó a la vista varias limitaciones. Por un lado, en clasificación binaria (Tabla 3.17), algunos algoritmos alcanzaron un *Recall* del 100 %, pero lo hicieron prediciendo casi todos los días como *Compra*. Este comportamiento generó un sesgo claro hacia la clase positiva, acompañado de valores de *Precisión* inferiores al 60 %, lo que evidencia un sobreajuste hacia una sola clase y reduce la utilidad práctica del enfoque. Por otro lado, los modelos de regresión logarítmica (Tabla 3.18) reprodujeron un patrón ya observado en otras compañías: ofrecieron *Accuracy* cercanos al 60 %, pero con precisiones muy bajas en la clase *Comprar*, lo que limitó su capacidad para identificar oportunidades reales de entrada.

En cuanto a los *ensamblados*, el esquema ponderado por *Accuracy* (Tabla 3.19) mostró un rendimiento intermedio: alcanzó una precisión del 76 % en la clase *No*



Figura 3.14: Subidas y bajadas 2023 - QQQ

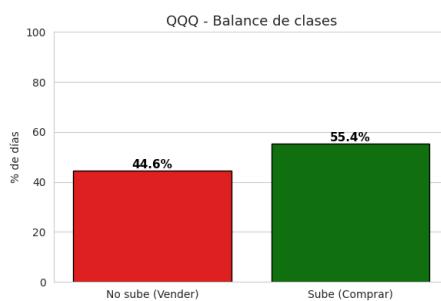


Figura 3.15: Balance de clases - QQQ

comprar, aunque penalizó de manera considerable la clase *Comprar*. El *ensamblado* por *Precisión* (Tabla 3.20), en cambio, elevó el *Recall* de la clase *Comprar* hasta el 91 %, pero lo consiguió a costa de sobrerepresentar esta clase, reduciendo con ello la fiabilidad global del sistema. En conjunto, aunque la tendencia de 2023 fue claramente alcista, los modelos no lograron capturar un equilibrio adecuado entre señales positivas y negativas, y en particular los modelos *ensamblados* basados en *Precisión* no parecieron muy apropiados para apoyar decisiones de compra en un escenario real.

Cuadro 3.17: Resultados de Clasificación Binaria - QQQ

Modelo	Accuracy	Precisión	Recall	F1-Score	Peso Acc.	Peso Prec.
Combined LSTM	0.5636	0.5636	1.0000	0.7209	0.092	0.146
Combined GRU	0.5590	0.5590	1.0000	0.7171	-	0.139
Sentiment LSTM	0.5556	0.5556	1.0000	0.7143	-	0.135
Sentiment RF	0.5378	0.5840	0.5840	0.5840	-	0.149
Sentiment GRU	0.5156	0.5385	0.8960	0.6727	-	-
Combined RF	0.4711	0.6500	0.1040	0.1793	-	0.169

Cuadro 3.18: Resultados de Regresión Logarítmica - QQQ

Modelo	Accuracy	Precisión	Recall	F1-Score	Peso Acc.	Peso Prec.
Reg Combined RF	0.6067	0.3077	0.3750	0.3380	0.141	-
Reg Combined GRU	0.5828	0.3030	0.4762	0.3704	0.111	-
Reg Sentiment LSTM	0.5644	0.2899	0.4762	0.3604	0.092	-
Reg Sentiment RF	0.5607	0.1846	0.1875	0.1860	0.089	-
Reg Sentiment GRU	0.5521	0.2687	0.4286	0.3303	-	-
Reg Combined LSTM	0.5181	0.2644	0.4423	0.3309	-	-

Cuadro 3.19: Resultados del Ensemble por Accuracy - QQQ

Clase	Precisión	Recall	F1-Score
No Comprar	0.7639	0.6250	0.6875
Comprar	0.3125	0.4688	0.3750
Accuracy			0.5833

Cuadro 3.20: Resultados del Ensemble por Precisión - QQQ

Clase	Precisión	Recall	F1-Score
No Comprar	0.4762	0.1000	0.1653
Comprar	0.5588	0.9120	0.6930
Accuracy			0.5511

NVIDIA (NVDA)

En la evolución histórica del precio de cierre (Figura 3.16) se observó un crecimiento casi ininterrumpido hasta comienzos de 2022, impulsado por la expansión del sector tecnológico, la elevada demanda de GPUs en videojuegos, minería de criptomonedas y aplicaciones de inteligencia artificial. Sin embargo, durante ese mismo año la cotización sufrió una corrección significativa, en consonancia con la caída generalizada del sector, asociada a la subida de tipos de interés, la ralentización del consumo de hardware y las restricciones a la exportación de chips hacia China. A lo largo de 2023, el valor retomó con fuerza la tendencia alcista, beneficiado por el auge de la inteligencia artificial generativa y la creciente demanda de GPUs en este ámbito, situando a *NVIDIA* entre las compañías con mayor revalorización bursátil del periodo (Figura 3.17).

El balance de clases del conjunto de datos (Figura 3.18) mostró una distribución relativamente equilibrada, con una ligera prevalencia de días al alza. Esta configuración, del mismo modo que en la mayoría de las otras compañías, planteó un escenario de entrenamiento sin un sesgo fuerte hacia ninguna de las dos clases.

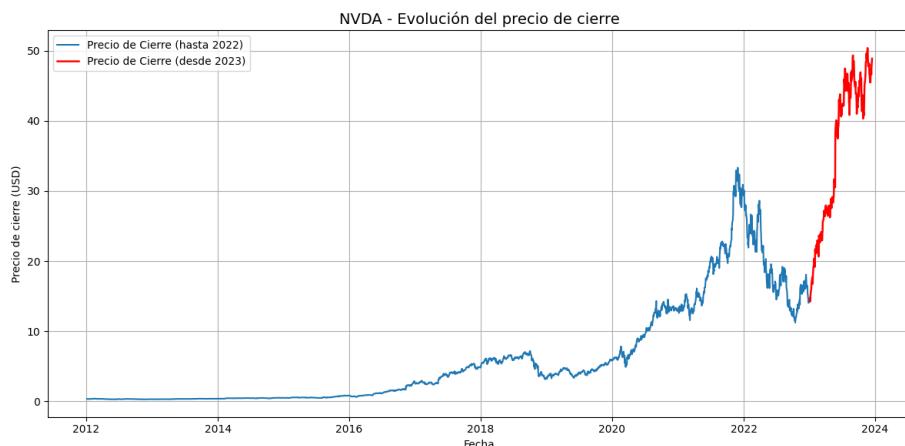


Figura 3.16: Histórico del precio de cierre - NVDA

En lo referente a los resultados, los modelos de regresión logarítmica (Tabla 3.22) ofrecieron valores de *Accuracy* aceptables, cercanos al 60 %, aunque con niveles de *Precisión* insuficientes en la clase *Comprar*, lo que limitó su valor práctico. En el caso de la clasificación binaria (Tabla 3.21), los valores de *Accuracy* fueron en general reducidos, con numerosos modelos por debajo del 50 %. No obstante, algunos alcanzaron métricas de *Precisión* destacables pese a su bajo rendimiento global. El modelo *Sentiment LSTM* constituyó un ejemplo llamativo: con un *Accuracy* de apenas un 49 %, alcanzó un 70 % de *Precisión* en la clase *Comprar*, aunque con un *Recall* muy bajo. En la práctica, este comportamiento implicó que el modelo predijo pocas veces la clase positiva, pero cuando lo hizo, la señal resultó fiable, lo que lo



Figura 3.17: Subidas y bajadas 2023 - NVDA

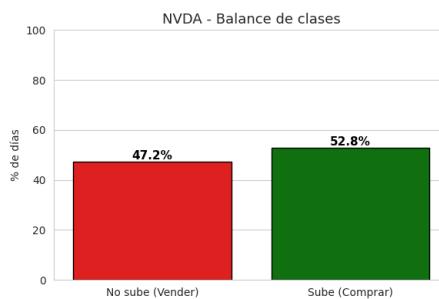


Figura 3.18: Balance de clases - NVDA

convirtió en un candidato interesante como apoyo puntual en la toma de decisiones de compra.

Los resultados de los *ensamblados* ofrecieron una visión complementaria. El esquema ponderado por *Precisión* (Tabla 3.24) mostró un claro sesgo hacia la clase *No comprar*, reduciendo de manera significativa la detección de señales positivas. Sin embargo, cuando identificó oportunidades de entrada, lo hizo con un 72 % de *Precisión*, lo que lo convierte en una herramienta útil para confirmar hipótesis de compra, aunque implique dejar escapar muchas oportunidades. Por su parte, el *ensamblado* basado en *Accuracy* (Tabla 3.23) ofreció un comportamiento opuesto: alcanzó un 76 % de *Precisión* en la clase *No comprar*, lo que lo hace más adecuado como filtro para descartar operaciones que como generador de señales de entrada.

Cuadro 3.21: Resultados de Clasificación Binaria - NVDA

Modelo	Accuracy	Precisión	Recall	F1-Score	Peso Acc.	Peso Prec.
Combined GRU	0.5450	0.5826	0.5826	0.5826	0.092	0.132
Combined RF	0.5062	0.5789	0.4818	0.5259	-	0.110
Sentiment LSTM	0.4979	0.7000	0.2044	0.3164	-	0.220
Sentiment GRU	0.4855	0.6857	0.1752	0.2791	-	0.192
Sentiment RF	0.4813	0.5612	0.4015	0.4681	-	-
Combined LSTM	0.4807	0.5833	0.1429	0.2295	-	0.102

Cuadro 3.22: Resultados de Regresión Logarítmica - NVDA

Modelo	Accuracy	Precisión	Recall	F1-Score	Peso Acc.	Peso Prec.
Reg Combined GRU	0.6089	0.3382	0.4792	0.3966	0.175	-
Reg Combined RF	0.5649	0.2625	0.3182	0.2877	0.113	-
Reg Combined LSTM	0.5502	0.3107	0.5818	0.4051	0.097	-
Reg Sentiment LSTM	0.5475	0.3187	0.6042	0.4173	0.095	-
Reg Sentiment RF	0.5356	0.2632	0.3788	0.3106	-	-
Reg Sentiment GRU	0.5251	0.2921	0.5417	0.3796	-	-

Cuadro 3.23: Resultados del Ensemble por Accuracy - NVDA

Clase	Precisión	Recall	F1-Score
No Comprar	0.7566	0.6609	0.7055
Comprar	0.3295	0.4394	0.3766
Accuracy			0.6000

Cuadro 3.24: Resultados del Ensemble por Precisión - NVDA

Clase	Precisión	Recall	F1-Score
No Comprar	0.4604	0.8942	0.6078
Comprar	0.7179	0.2044	0.3182
Accuracy			0.5021

4. Conclusiones

Análisis de resultados

Los resultados obtenidos muestran que los modelos de regresión logarítmica tuvieron un rendimiento limitado, con valores de *Precisión* generalmente por debajo del 30 %, lo que impidió que fueran incluidos en los *ensamblados* construidos a partir de esta métrica. No obstante, algunos alcanzaron valores aceptables de *Accuracy*, lo que hizo que en determinados casos fueran considerados dentro de los *ensamblados* basados en dicha métrica. Su inclusión, sin embargo, redujo la *Precisión* en la clase positiva, limitando su utilidad para la predicción de señales de compra. Aun así, esta característica permitió que los *ensamblados* por *Accuracy* se mostrasen eficaces a la hora de descartar compras, llegando en algunos casos a superar el 75 % de *Precisión* en la clase negativa.

En cambio, los *ensamblados* orientados a la *Precisión* presentaron un comportamiento más heterogéneo. En compañías como *Coca-Cola*, el rendimiento apenas superó el azar, mientras que en otras, como *Gilead Sciences* o *NVIDIA*, se observaron resultados más prometedores, con valores en torno al 60 % de *Precisión* y un 74 % de *recall* en el primer caso, o bien una *Precisión* elevada pero con bajo *recall* en el segundo. Estos hallazgos sugieren que, en contextos específicos, los modelos pueden aportar información útil como herramienta de apoyo más que como sistemas autónomos de predicción.

Por otra parte, al comparar las arquitecturas de redes recurrentes (*LSTM* y *GRU*) con modelos clásicos basados en árboles de decisión como *Random Forest*, no se observó una ventaja clara a favor de las arquitecturas más complejas. Del mismo modo, los experimentos mostraron que los modelos entrenados con datasets completos (variables técnicas + sentimientos) no superaron de forma consistente a los que utilizaron únicamente las variables de sentimiento. En general, ambos enfoques ofrecieron resultados muy similares, lo que refuerza la idea de que la capacidad predictiva de los modelos aplicados fue modesta.

Limitaciones del estudio

Los resultados deben interpretarse con cautela. En primer lugar, la predicción diaria de movimientos bursátiles está condicionada por un alto nivel de ruido y aleatoriedad, lo que dificulta que los modelos aprendan patrones estables. Además, el periodo de prueba se limitó al año 2023, un ejercicio especialmente marcado por la volatilidad derivada de la inflación, las políticas monetarias restrictivas de los bancos centrales y el auge del sector tecnológico impulsado por la inteligencia artificial. Este contexto particular puede haber influido en los resultados y restringe la generaliza-

ción a otros períodos históricos. La selección de compañías, aunque sectorialmente diversa, fue reducida, y no puede considerarse representativa de todo el mercado.

Otro aspecto a tener en cuenta es la propia construcción del dataset, ya que la calidad y relevancia de las noticias financieras son heterogéneas y pueden introducir sesgos difíciles de controlar. Del mismo modo, el procedimiento de selección de variables técnicas basado en *Random Forest* puede favorecer predictores correlacionados, reduciendo la efectividad de la reducción de dimensionalidad.

Aplicabilidad y utilidad práctica

A pesar de estas limitaciones, el trabajo aporta algunas conclusiones relevantes. En particular, los *ensamblados* por *Accuracy* mostraron utilidad como herramienta de apoyo para identificar escenarios en los que no conviene realizar operaciones de compra, lo que podría servir como filtro adicional dentro de un proceso de análisis técnico. Más allá de su aplicación directa, los resultados refuerzan la idea, ya señalada en estudios previos, de que la predicción a muy corto plazo en los mercados financieros resulta especialmente compleja y está dominada por el ruido.

En este sentido, el presente trabajo puede considerarse una prueba de concepto que evidencia tanto las dificultades como las posibilidades de aplicar técnicas de *Machine Learning* y análisis de sentimientos a la predicción bursátil. Aunque los modelos desarrollados no constituyen una estrategia de *trading* lista para ser implementada, sí ofrecen un punto de partida para investigaciones futuras orientadas a horizontes temporales más amplios o a la integración de información adicional. De este modo, la contribución principal reside en haber explorado de manera sistemática la interacción entre variables técnicas y de sentimiento en el contexto de predicciones de muy corto plazo, aportando evidencias empíricas sobre sus limitaciones y posibles usos como sistemas de soporte en la toma de decisiones.

5. Siguientes pasos

Una posible línea de extensión parte del uso de *BERT* en este trabajo para incorporar información procedente de noticias financieras. En lugar de limitarse a representaciones estáticas del texto, resulta interesante considerar modelos de lenguaje de gran escala (*LLMs*), capaces de capturar relaciones semánticas más complejas y de mostrar un mejor rendimiento en tareas de comprensión contextual. Combina-dos con técnicas de *Retrieval-Augmented Generation (RAG)*, estos modelos podrían integrar de forma dinámica fuentes externas, tales como resultados trimestrales de compañías, informes de mercado o noticias financieras recientes, ampliando de ma-nera sustancial la información accesible al sistema.

Otra vía de desarrollo consiste en la aplicación de redes neuronales a la resolu-ción de ecuaciones en derivadas parciales (EDPs), empleando enfoques como los *Physics-Informed Neural Networks (PINNs)*. Del mismo modo que en este trabajo se recurrió a redes para modelar series temporales, estas arquitecturas se utilizarían para aproximar soluciones a EDPs relevantes en la valoración de opciones, como los modelos de Black-Scholes o Heston (Dhiman & Hu, 2023). Aunque el propósito no sería generar señales directas de compra o venta, la propuesta mantendría la idea de aplicar redes neuronales a problemas propios de las finanzas cuantitativas, reforzando el vínculo entre técnicas de aprendizaje profundo y teoría financiera.

Finalmente, otra dirección prometedora es abordar el problema desde la pers-pectiva del *Reinforcement Learning*. Agentes entrenados en entornos de simulación pueden aprender estrategias de trading optimizando recompensas ligadas al benefi-cio acumulado. Frameworks como *FinRL* ofrecen entornos completos para entrenar agentes de aprendizaje por refuerzo profundo (*DRL*) en condiciones cercanas a los mercados reales, considerando factores como costos de transacción, liquidez o aver-sión al riesgo (Liu, Yang, Gao, & Wang, 2021). Asimismo, se han propuesto enfoques orientados al aprovechamiento de oportunidades intradía, apoyados en arquitectu-ras de tipo *dueling Q-network* y funciones de recompensa diseñadas para equilibrar riesgo y rentabilidad (Kabbani & Duman, 2022).

Bibliografía

- [1] Araci, D. (2019). FinBERT: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*. Recuperado de <https://arxiv.org/abs/1908.10063>
- [2] Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166. <https://doi.org/10.1109/72.279181>
- [3] Chan, E. (2013). *Algorithmic trading: Winning strategies and their rationale*. Wiley Finance.
- [4] OpenAI. (2023). ChatGPT [Large language model]. Recuperado de <https://chat.openai.com/>
- [5] Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. In *Proceedings of the IEEE International Conference on Big Data* (pp. 2823–2824). IEEE. <https://doi.org/10.1109/BigData.2015.7364089>
- [6] 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. *arXiv preprint arXiv:1406.1078*. <https://arxiv.org/abs/1406.1078>
- [7] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. <https://arxiv.org/abs/1810.04805>
- [8] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)* (pp. 4171–4186). <https://doi.org/10.18653/v1/N19-1423>
- [9] Dhiman, A., & Hu, Y. (2023). Physics informed neural network for option pricing. *arXiv preprint arXiv:2312.06711*. <https://arxiv.org/abs/2312.06711>
- [10] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654–669. <https://doi.org/10.1016/j.ejor.2017.11.054>
- [11] Goldberg, Y. (2017). *Neural network methods for natural language processing*. Synthesis Lectures on Human Language Technologies, 10(1), 1–309. <https://doi.org/10.2200/S00762ED1V01Y201703HLT037>

- [12] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- [13] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222–2232. <https://doi.org/10.1109/TNNLS.2016.2582924>
- [14] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [15] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8), 2554–2558. <https://doi.org/10.1073/pnas.79.8.2554>
- [16] Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 328–339). <https://doi.org/10.18653/v1/P18-1031>
- [17] Kabbani, T., & Duman, E. (2022). Deep reinforcement learning approach for trading automation in the stock market. *arXiv preprint arXiv:2208.07165*. <https://arxiv.org/abs/2208.07165>
- [18] Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1–2), 307–319. [https://doi.org/10.1016/S0925-2312\(03\)00372-2](https://doi.org/10.1016/S0925-2312(03)00372-2)
- [19] Liu, B. (2012). *Sentiment analysis and opinion mining*. Synthesis Lectures on Human Language Technologies, 5(1), 1–167. <https://doi.org/10.2200/S00416ED1V01Y201204HLT016>
- [20] Liu, X.-Y., Yang, H., Gao, J., & Wang, C. D. (2021). FinRL: Deep reinforcement learning framework to automate trading in quantitative finance. In *Proceedings of the 2nd ACM International Conference on AI in Finance (ICAIF 21)* (pp. 1–9). ACM. <https://doi.org/10.1145/3490354.3494381>
- [21] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
- [22] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115–133. <https://doi.org/10.1007/BF02478259>
- [23] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. <https://arxiv.org/abs/1301.3781>

- [24] Narayana, M. L., Kartha, A. J., Mandal, A. K., Roshini, P., Suresh, A., & Jose, A. C. (2025). Ensemble time series models for stock price prediction and portfolio optimization with sentiment analysis. *Journal of Intelligent Information Systems*, 65(1), 1–25. <https://doi.org/10.1007/s10844-024-00892-7>
- [25] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). <https://doi.org/10.3115/v1/D14-1162>
- [26] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of NAACL-HLT* (pp. 2227–2237). <https://doi.org/10.18653/v1/N18-1202>
- [27] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408. <https://doi.org/10.1037/h0042519>
- [28] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536. <https://doi.org/10.1038/323533a0>
- [29] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513–523. [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0)
- [30] Shreve, S. E. (2004). *Stochastic calculus for finance II: Continuous-time models*. Springer. <https://doi.org/10.1007/b97490>
- [31] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2), 267–307. https://doi.org/10.1162/COLI_a_00049
- [32] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, ., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS 2017)* (pp. 5998–6008). <https://arxiv.org/abs/1706.03762>
- [33] White, H. (1988). Economic prediction using neural networks: The case of IBM daily stock returns. In *Proceedings of the IEEE International Conference on Neural Networks* (pp. 451–458). IEEE. <https://doi.org/10.1109/ICNN.1988.23864>
- [34] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Dean, J. (2016). Googles neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*. <https://arxiv.org/abs/1609.08144>

- [35] Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4), e1253. <https://doi.org/10.1002/widm.1253>
- [36] Zhang, Y., Han, X., Xu, W., & Li, Z. (2023). FinGPT: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*. <https://arxiv.org/abs/2306.06031>
- [37] Zhong, D. (2020). FNSPID: A financial news dataset for stock price movement prediction. *GitHub repository*. Recuperado de https://github.com/Zdong104/FNSPID_Financial_News_Dataset