

# **Predicción de valores y tendencias de cierre del IBEX35 mediante machine learning y webscraping**

**Forecasting of IBEX 35 closing values and trends using  
machine learning and web scraping**

**Santiago López Begines, PhD**  
**Diciembre 2025**

**Trabajo Fin de Máster**

**Máster en Big Data y Data Science aplicados a la  
Economía y a la Administración y Dirección de  
Empresas**



# Predicción de valores y tendencias de cierre del IBEX35 mediante *machine learning* y *webscraping*

Trabajo de fin de Máster

Santiago López Begines, PhD

2025-12-05

## Índice

<b>1 Resumen</b>	<b>5</b>
<b>2 Abstract</b>	<b>5</b>
<b>3 Introducción</b>	<b>6</b>
3.1 Ibex35 . . . . .	6
3.2 Empresas que lo componen y su cotización a 31 Octubre de 2025 . . . . .	7
3.3 Información exógena . . . . .	7
3.3.1 Índices externos . . . . .	7
3.3.2 Análisis de sentimientos . . . . .	8
3.4 Selección de Fuentes de Datos de Sentimiento . . . . .	8
3.4.1 Intención Inicial: Análisis de Prensa Nacional e Internacional . . . . .	8
3.4.2 Limitaciones Reconocidas de GDELT . . . . .	10
3.4.3 Justificación del Uso de GDELT . . . . .	10
<b>4 Hipótesis de trabajo y Objetivos principales</b>	<b>11</b>
<b>5 Materiales y Métodos</b>	<b>11</b>
5.1 Software . . . . .	11
5.2 Índices bursátiles . . . . .	12
5.3 Análisis de Sentimientos . . . . .	12
5.4 Modelos aplicados . . . . .	14
5.4.1 Modelos Lineales . . . . .	15
5.4.2 Modelos Basados en Árboles . . . . .	17
5.4.3 Redes Neuronales Clásicas . . . . .	20
5.4.4 Redes Neuronales Recurrentes . . . . .	21
<b>6 Resultados</b>	<b>25</b>
6.1 Fase 1: Descarga y consolidación de datos. . . . .	25
6.1.1 Descarga de datos financieros . . . . .	25
6.1.2 Descarga de datos desde GDELT . . . . .	26
6.2 Fase 2: Análisis exploratorio de datos financieros y datos de sentimientos de GDELT . . . . .	27

6.2.1	Análisis exploratorio de índices bursátiles . . . . .	27
6.2.2	Análisis exploratorio de noticias mediante GDELT. . . . .	38
6.3	Fase 3: Ingeniería de características. . . . .	43
6.3.1	Creación de variables financieras . . . . .	44
6.3.2	Comprobación de variables presentes . . . . .	46
6.3.3	Limpieza de Variables . . . . .	47
6.3.4	Random Forest para ranking de variables . . . . .	48
6.3.5	Limpieza del análisis de sentimientos . . . . .	50
6.4	Fase 4: Normalización y escalado de datos. . . . .	51
6.4.1	Unión de datos financieros con datos de sentimientos . . . . .	51
6.4.2	Normalización de datos . . . . .	52
6.5	Fase 5: Predicción de valores futuros. . . . .	52
6.5.1	Modelos Clásicos . . . . .	53
6.5.2	Modelos de Machine Learning. . . . .	54
6.5.3	Evaluación de predicciones . . . . .	67
6.6	Fase 6: Validación final con datos nuevos . . . . .	71
6.6.1	Descarga de nuevos datos . . . . .	71
6.6.2	Preparación de datos para validación . . . . .	71
6.6.3	Evaluar modelos . . . . .	73
<b>7</b>	<b>Discusión</b>	<b>78</b>
<b>8</b>	<b>Conclusiones</b>	<b>79</b>
<b>Referencias</b>		<b>80</b>

# 1 Resumen

Este Trabajo Fin de Máster desarrolla un sistema predictivo del IBEX35 mediante machine learning, integrando datos financieros (2004-2025) con análisis de sentimiento de GDELT. Se implementaron 348 configuraciones experimentales, entre clasificación y regresión, combinando múltiples arquitecturas (modelos clásicos ARIMA/Prophet, basados en árboles como Random Forest/XGBoost/LightGBM, redes neuronales MLP, y arquitecturas recurrentes GRU/LSTM) evaluadas sobre diferentes datasets (escalados/sin escalar, series largas/cortas, con/sin sentimiento) y horizontes temporales (1, 5, 10, 20 días).

El análisis exploratorio reveló tres períodos de volatilidad extrema (crisis 2008, deuda europea 2011-2012, COVID-19), y correlaciones significativas con índices europeos (DAX 0.82, EuroNext 0.87, FTSE 0.79). El feature engineering generó 97 variables técnicas y exógenas, reducidas mediante correlación, Random Forest y criterios de varianza a 44 predictores finales.

Los resultados muestran precisiones direccionales entre 53-62%, siendo GRU/LSTM los más consistentes (62.22% y 61.97%). Sin embargo, las comparaciones estadísticas mediante tests de McNemar, Diebold-Mariano y bootstrap revelaron que la mayoría de modelos no superan significativamente el baseline. El sentimiento de GDELT aporta mejoras marginales (7% en LightGBM específico) pero generalmente no significativas. Los criterios naïve (Cohen's Kappa  $>0.1$ , p-value binomial  $<0.05$ ) identificaron únicamente 10 modelos con capacidad predictiva real de 352 evaluados. La validación final (octubre-noviembre 2025) mostró accuracy 36-46%, evidenciando sobreajuste y limitada generalización, confirmando la extrema dificultad de capturar patrones predictivos consistentes en series financieras de alta frecuencia con elevada relación señal/ruido.

# 2 Abstract

This Master's thesis develops a predictive system for the IBEX35 using machine learning, integrating financial data (2004-2025) with GDELT sentiment analysis. We implemented 348 experimental configurations combining multiple architectures (classical ARIMA/Prophet, tree-based Random Forest/XGBoost/LightGBM, MLP neural networks, and recurrent GRU/LSTM) evaluated across different datasets (scaled/unscaled, long/short series, with/without sentiment) and temporal horizons (1, 5, 10, 20 days).

Exploratory analysis revealed three extreme volatility periods (2008 crisis, 2011-2012 European debt, COVID-19), and significant correlations with European indices (DAX 0.82, EuroNext 0.87, FTSE 0.79). Feature engineering generated 97 technical and exogenous variables, reduced through correlation analysis, Random Forest importance, and variance criteria to 44 final predictors.

Results show directional accuracies between 53-62%, with GRU/LSTM most consistent (62.22% and 61.97%). However, statistical comparisons using McNemar, Diebold-Mariano tests and bootstrap revealed most models significantly do not outperform baseline. GDELT sentiment provides marginal improvements (7% in specific LightGBM) but generally non-significant. Naïve criteria (Cohen's Kappa  $>0.1$ , binomial p-value  $<0.05$ ) identified only 10 models with genuine predictive capacity from 352 evaluated. Final validation (October-November 2025) showed 36-46% accuracy, evidencing overfitting and limited generalization, confirming the extreme difficulty of capturing consistent predictive patterns in high-frequency financial series with elevated signal-to-noise ratios.

### 3 Introducción

#### 3.1 Ibex35

Los índices bursátiles constituyen una herramienta invaluable para el análisis del mercado financiero. Estos índices nos proporcionan información resumida y cuantificable, facilitando la toma de decisiones. A su vez, ayudan a inversores a comparar el rendimiento de carteras y estrategias de inversión (Yubero and Aguilar 2024).

El primer índice bursátil fue establecido en New York por Charles Dow, Edward D. Jones y asociados en 1884, fundando el *Dow Jones Transportation Average (DJTA)*. Con el tiempo surgieron otros índices para medir diferentes segmentos del mercado. Entre ellos cabe destacar el S&P500, que aglutina las 500 empresas de cotización pública con mayor valoración en el mercado estadounidense, el DAX40, que hace lo propio con las empresas del mercado alemán, el FTSE100 del mercado británico o el IBEX35 en el mercado español.

El IBEX35 fue creado el 1992 como un índice selectivo de los 35 valores más líquidos del SIBE (Sistema de Interconexión Bursátil Español). Al ser un índice selectivo, se calcula a partir de una selección de valores representativos de el mercado, contraste uno que considere todos los valores del mercado, como *all share*.

El Ibex 35 es un índice de precios ponderado por capitalización ajustada por flotante, de manera que las empresas con mayor capitalización son las que tienen mayor peso en el índice y, por tanto, las fluctuaciones de sus precios son las que influyen en mayor medida en el movimiento final del índice. La base del Ibex 35 es de 3.000 puntos y su fecha de publicación por primera vez fue el 14 de enero de 1992. En cada instante  $t$ , el índice se calcula aplicando al índice obtenido en el instante inmediatamente anterior ( $t - 1$ ) la variación de la capitalización de las empresas que lo componen, corregida por un factor de ajuste ( $J$ ) que corrige el posible impacto de determinadas operaciones financieras.(Yubero and Aguilar 2024)

A lo largo de su historia, el IBEX35 ha experimentado crisis financieras, turbulencias políticas y shocks económicos que han generado alta volatilidad. Esta inestabilidad característica de los mercados bursátiles, combinada con su sensibilidad extrema a condiciones iniciales, ha llevado a que sean modelados como sistemas caóticos (Klioutchnikov, Sigova, and Beizerov 2017; Mandelbrot and Hudson 2004).

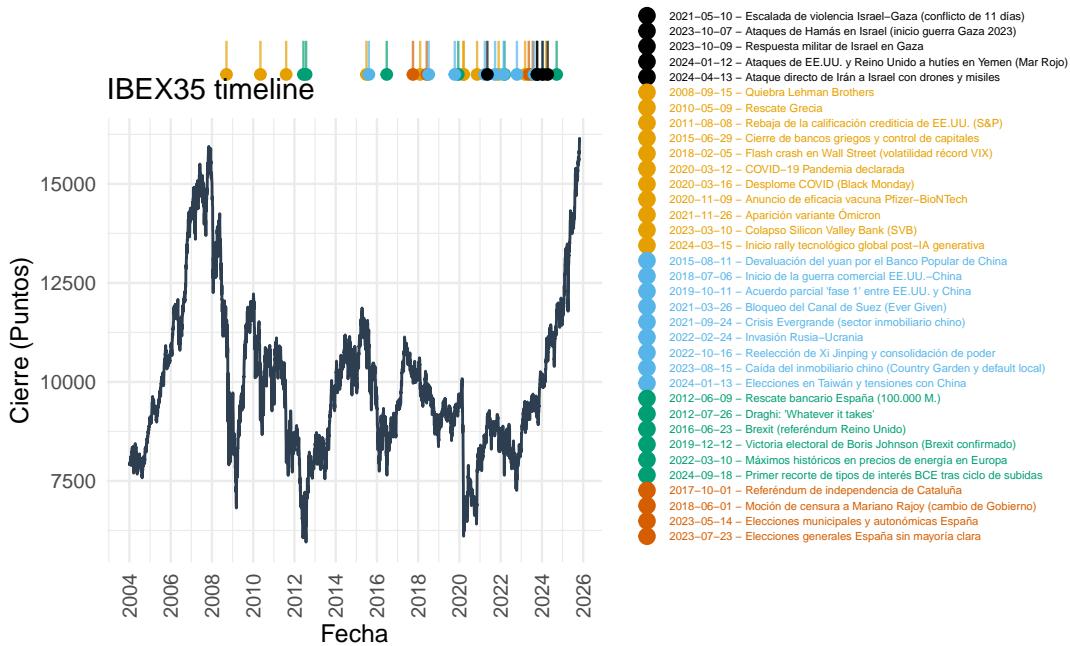


Figure 1: Serie histórica de precios de cierre del IBEX e impactos de eventos históricos

La predicción de estos mercados resulta extraordinariamente difícil, incluso empleando técnicas avanzadas de inteligencia artificial. Fischer y Krauss(Fischer and Krauss 2018) reportan que redes LSTM (Long-Short Term Memory) alcanzan apenas 56% de precisión direccional de retornos en el S&P500, superando el azar por escaso margen. Esta dificultad generalizada en la predicción, presente incluso en mercados altamente líquidos y con abundante información, establece expectativas realistas sobre el desafío que representa modelar el IBEX35.

### 3.2 Empresas que lo componen y su cotización a 31 Octubre de 2025

Las catálogo de empresas o compañías que componen el IBEX35 está controlado por el Comité Asesor Técnico, dependiente de su administrador, la Sociedad de Bolsas S.A., entidad participada por Bolsas y Mercados Españoles, S.A. (BME)(Yubero and Aguilar 2024).

### 3.3 Información exógena

#### 3.3.1 Índices externos

Como se ha expuesto anteriormente, la cotización del IBEX35 puede verse afectada por eventos políticos y/o económicos nacionales e internacionales. Para complementar la predicción del IBEX35, emplearemos además otros índices internacionales como son:

Table 1: Índices y activos financieros utilizados

Ticker	Nombre	Descripción
--------	--------	-------------

$^{\text{IBEX}}$	IBEX 35	Índice bursátil de referencia de España (35 empresas)
$^{\text{GSPC}}$	S&P 500	Índice de las 500 empresas más grandes de EE.UU.
$^{\text{GDAXI}}$	DAX	Índice de las 40 empresas más grandes de Alemania
$^{\text{SBF120}}$	SBF 120	Índice de las 120 empresas más grandes de Francia
$^{\text{N100}}$	Euronext 100	Índice de las 100 empresas más grandes de la Eurozona
$^{\text{N225}}$	Nikkei 225	Índice de las 225 empresas más grandes de Japón
$^{\text{FTSE}}$	FTSE 100	Índice de las 100 empresas más grandes del Reino Unido
$^{\text{VIX}}$	VIX	Índice de volatilidad del mercado de EE.UU. (miedo del mercado)
EURUSD=X	EUR/USD	Tipo de cambio Euro-Dólar estadounidense
GC=F	Gold Futures	Precio del oro (futuros)
CL=F	Crude Oil Futures	Precio del petróleo WTI (futuros)

### 3.3.2 Análisis de sentimientos

Tetlock P.C reportó en 2007 (Tetlock 2007) que el análisis del contenido de la columna “Abreast of the Market” del *Wall Street Journal* puede predecir movimientos del mercado, correlacionando altos niveles de pesimismo en medios retornos negativos al día siguiente. Este efecto se revierte a los 5-10 días, concluyendo que la información se va incorporando poco a poco. En la misma línea, (Da, Engelberg, and Gao 2015) construyeron un índice FEARS (Financial and Economic Attitudes Revealed by Search) usando Google Search Volume Index. Este índice agrupa términos clave como *recesión*, *bancarrota*, *desempleo*, etc. Entre 2004 y 2011, observaron que los FEARS (i) predicen reverisiones de rentabilidad a corto plazo, (ii) predicen aumentos temporales de la volatilidad y (iii) predicen flujos de fondos mutuos desde fondos de renta variable hacia fondos de renta fija. Los resultados fueron coherentes con las teorías sobre el sentimiento de los inversores.

## 3.4 Selección de Fuentes de Datos de Sentimiento

### 3.4.1 Intención Inicial: Análisis de Prensa Nacional e Internacional

El diseño inicial de este estudio contemplaba la integración de fuentes de prensa financiera española, europea y estadounidense mediante técnicas de web scraping y extracción de feeds RSS. La estrategia planteada incluía:

- *Prensa española*: Se priorizaron fuentes con cobertura directa del IBEX35 como Cinco Días, Expansión, El Economista, El Confidencial y La Vanguardia, dada su especialización en mercados nacionales y su impacto potencial en inversores locales.

- *Prensa europea:* Se identificaron fuentes clave como Reuters Europe, Handelsblatt (Alemania) y Les Échos (Francia), junto con comunicados del Banco Central Europeo, considerando la alta integración de mercados europeos y su influencia directa sobre valores del IBEX35.
- *Prensa estadounidense:* Se planificó incorporar CNBC, MarketWatch, comunicados de la Reserva Federal y Reuters US, reconociendo el impacto de los mercados estadounidenses (S&P500) y las decisiones de política monetaria de la Fed sobre los índices europeos.

Esta aproximación hubiera permitido capturar sentimientos específicos del mercado español con granularidad temporal y temática adaptada al problema de investigación.

#### **3.4.1.1 Limitaciones Técnicas y Cambio de Estrategia**

Durante la fase de implementación se identificaron múltiples obstáculos que imposibilitaron la ejecución del plan original:

1. *Disponibilidad de RSS limitada:* Varios medios españoles (El Economista, El Confidencial) restringen o han discontinuado sus feeds RSS completos, ofreciendo únicamente titulares sin contenido textual suficiente para análisis de sentimiento.
2. *Barreras técnicas de acceso:* Implementaciones anti-scraping mediante sistemas de detección de bots, captchas dinámicos y limitación de tasa de peticiones dificultaron la extracción automatizada y sistemática de contenido.
3. *Requisitos de suscripción:* Fuentes de alta calidad como Financial Times, The Wall Street Journal o Bloomberg requieren suscripciones de pago y restringen el acceso programático a su contenido, excediendo los recursos disponibles para este proyecto.
4. *Heterogeneidad de formatos:* La diversidad de estructuras HTML, idiomas y formatos de publicación entre fuentes complicaba la construcción de un pipeline unificado y robusto de procesamiento.
5. *Cobertura temporal inconsistente:* Los feeds RSS disponibles raramente mantienen archivos históricos más allá de 30-90 días, limitando severamente el periodo de análisis retrospectivo necesario para modelado predictivo.

Ante estas limitaciones, se adoptó el proyecto **GDELT (Global Database of Events, Language, and Tone)** como fuente primaria de datos de sentimiento. GDELT es una iniciativa de monitorización global de medios de comunicación que opera desde 2013, procesando noticias en más de 100 idiomas desde miles de fuentes en tiempo real.(Leetaru and Schrot, n.d.)

#### **Características principales de GDELT:**

- **Cobertura global:** Procesa aproximadamente 300,000 artículos diarios de medios impresos, online, televisión y radio de todo el mundo.
- **Acceso abierto:** Todos los datos son de dominio público y accesibles sin restricciones mediante APIs o descargas directas.
- **Granularidad temporal:** Actualizaciones cada 15 minutos con archivos históricos disponibles desde 1979 (eventos) y 2013 (GKG completo).

- **Datos estructurados:** El Global Knowledge Graph (GKG) proporciona extracción automática de entidades, temas, sentimientos y metadatos pre-procesados.

#### **Variables relevantes del GKG 1.0:**

La base de datos GKG incluye variables críticas para análisis de sentimiento:

- **TONE:** Métrica compuesta que cuantifica el sentimiento del texto en una escala de -100 (extremadamente negativo) a +100 (extremadamente positivo), calculada como la diferencia entre proporciones de palabras positivas y negativas.
- **Componentes de TONE:** Positive Score, Negative Score, Polarity, Activity Reference Density y Self/Group Reference Density, permitiendo análisis multidimensional del sentimiento.
- **THEMES:** Taxonomía de temas automáticamente identificados (e.g., ECON\_STOCKMARKET, ECON\_BANKRUPTCY), facilitando filtrado de noticias financieras relevantes.
- **ORGANIZATIONS:** Extracción de nombres de empresas mencionadas, permitiendo vincular noticias con valores específicos del IBEX35.
- **NUMARTS:** Número de artículos que mencionan el evento, funcionando como proxy de importancia mediática.

#### **3.4.2 Limitaciones Reconocidas de GDELT**

Es importante reconocer las limitaciones metodológicas de GDELT en comparación con aproximaciones basadas en modelos de lenguaje modernos:

1. **Análisis léxico tradicional:** GDELT utiliza diccionarios de palabras predefinidas para calcular sentimientos, sin incorporar modelos de lenguaje contextuales (BERT, GPT, RoBERTa) que capturan semántica compleja y matices lingüísticos.
2. **Tokenización básica:** El procesamiento se basa en separación de palabras convencional, sin considerar sub-word tokenization (BPE, WordPiece) que mejora el manejo de términos financieros especializados o neologismos.
3. **Ausencia de fine-tuning:** Los modelos de GDELT no están específicamente entrenados en corpus financieros, potencialmente reduciendo precisión en la detección de sentimientos en contextos bursátiles.
4. **Generalización geográfica:** Aunque filtra por países/empresas, GDELT agrega noticias globales sin especialización en mercados españoles, diluyendo señales locales específicas del IBEX35.

#### **3.4.3 Justificación del Uso de GDELT**

No obstante estas limitaciones, GDELT presenta ventajas prácticas significativas para este estudio:

- **Viabilidad técnica:** Acceso programático estable y documentado, sin restricciones legales o económicas.
- **Cobertura temporal:** Datos históricos consistentes desde 2015, permitiendo análisis de series largas.

- **Escalabilidad:** Procesamiento de volúmenes masivos de noticias sin necesidad de infraestructura propia de scraping.
- **Reproducibilidad:** Base de datos pública que permite replicación y validación independiente de resultados.

En resumen, aunque GDELT no representa la solución óptima desde una perspectiva de estado del arte en NLP, constituye una alternativa pragmática y viable que permite explorar una de las hipótesis centrales de este trabajo: determinar si variables de sentimiento mediático aportan información predictiva adicional a modelos basados exclusivamente en datos financieros técnicos.

## 4 Hipótesis de trabajo y Objetivos principales

La evolución de los mercados bursátiles es el reflejo dinámico de múltiples factores económicos, políticos y sociales. En este trabajo se aborda la predicción del comportamiento del índice IBEX35 a partir de un análisis extensivo de series temporales históricas. Se pretende integrar información financiera con señales exógenas como prensa económica, expectativas de otros índices internacionales y evolución de materias primas, con el objetivo de construir de predicción modelos robustos.

1. Construir una base de datos consolidada del IBEX35 desde 1992 hasta 2025.
2. Realizar un análisis descriptivo y exploratorio completo de precios, retornos, volatilidad y correlaciones.
3. Evaluar la relación entre las series del IBEX35 y variables externas (prensa económica, otros índices bursátiles, divisas, materias primas). Uso de web scrapping y web mining junto con análisis de sentimientos para intentar predecir comportamientos de la bolsa.
4. Entrenar y comparar modelos predictivos tradicionales y avanzados (ARIMA, Prophet, Regresión Logística, Random Forest, XGBoost, Light GBM, Redes Neuronales MLP, GRU y LSTM).
5. Validar la capacidad predictiva y la interpretabilidad de los modelos generados.

## 5 Materiales y Métodos

### 5.1 Software

Para la programación del análisis se usó principalmente R 4.5.2 (2025-10-31) y RStudio 2025.09.2+418. Algunas partes, especialmente lo relacionado con machine learning, fueron programadas en Python 3.13.5. En la programación de algunas partes de R y Python se usó la ayuda del modelo Claude Sonnet 4.5 de Anthropic (Anthropic 2024). Las versiones específicas de librerías usadas pueden consultarse en los documentos anexos relacionados con las 5 fases en que ha sido dividido el análisis.

## 5.2 Índices bursátiles

Para el acceso a los índices bursátiles se hizo uso de la librería 'yfinance' de python. Esta librería descarga los datos de los índices requeridos haciendo uso del *ticker*, identificador único del índice o valor. Hacemos uso de la función `stocks_downloader.download_ibex` almacenada en `code/py/stocks_downloader.py`. Ver Resultados/Fase 1 para mayor detalle.

## 5.3 Análisis de Sentimientos

Para los datos de noticias y análisis básico de sentimientos, se hizo uso del repositorio GDELT <http://data.gdelproject.org/gkg/>.

Para este trabajo se ha hecho uso de la versión GDELT 1.0 Global Knowledge Graph (GKG). Esta versión hace recopilación de artículos desde el 1 de Abril de 2013. La base de datos GKG hace un grafo completo de personas, organizaciones, ubicaciones, emociones, temas... además de incluir un identificador único de evento y vincular la información extraída con eventos de ese mismo artículo. La base de datos GKG se actualiza diariamente a las 6AM EST. El formato de los archivos descargados es un CSV comprimido en ZIP: `YYYYMMDD.gkg.csv.zip`. La versión 2.0 de GKG ofrece una nueva formulación de los atributos, además de actualizar un nuevo archivo cada 15 minutos. Para este trabajo se eligió la versión 1.0 sobre la 2.0 debido a la facilidad en el manejo del número de archivos (1 archivo/día versus 96 archivos/día).

En GDELT 1.0, las variables relevantes para el análisis de sentimientos incluyen:

- DATE: Fecha del evento (YYYYMMDD)
- NUMARTS: Número de artículos que mencionan el evento (proxy de importancia)
- COUNTS: Campo de GKG, no de Event Database
- THEMES: Lista de temas identificados (campo de GKG)
- LOCATIONS: Ubicaciones geográficas extraídas con coordenadas
- PERSONS: Nombres de personas identificadas en el texto
- ORGANIZATIONS: Empresas, ONGs, IGOs mencionadas
- TONE: Métrica de sentimiento (ver detalles abajo)
- CAMEOEVENTIDS: IDs únicos de eventos CAMEO relacionados
- SOURCES: Dominios/fuentes que publicaron el evento
- SOURCEURLS: URLs completas de los artículos fuente

El campo más adecuado para el análisis de sentimientos es **TONE**. TONE es el promedio del tono de todos los documentos que mencionan un evento, rangeando de -100 (extremadamente negativo) a +100 (extremadamente positivo), con valores comunes entre -10 y +10. Se calcula como Positive Score menos Negative Score. Los 6 componentes de TONE son:

- Tone: Score general (-100 a +100) = Positive Score - Negative Score
- Positive Score: % de palabras con connotación positiva (0 a +100)
- Negative Score: % de palabras con connotación negativa (0 a +100)
- Polarity: % de palabras con carga emocional (independiente de si es positiva/negativa)
- Activity Reference Density: % de palabras activas (proxy de "actividad" del texto)
- Self/Group Reference Density: % de pronombres (referencias personales/grupales)

La Interpretación práctica de éstos campos sería:

Table 2: Métricas de sentimiento GDELT

Métrica	Descripción	Rango	Interpretación
<b>mean_tone</b>	Sentimiento direccional (parámetro más importante)	-5 a -2	Día muy negativo para IBEX35
		-2 a 0	Día ligeramente negativo
		0 a 2	Día ligeramente positivo
		2 a 5	Día muy positivo
<b>mean_polarity</b>	Intensidad emocional	< 5	Día tranquilo, poca carga emocional
		5-10	Intensidad normal
		> 10	Día con mucha carga emocional (crisis, euforia)
<b>sd_tone</b>	Volatilidad	Baja (< 3)	Consenso en el sentimiento
		Alta (> 5)	Noticias muy mixtas (positivas y negativas)

Para poder realizar la descarga, filtrado y consolidación de archivos en lote, se ha usado del siguiente esquema:

Table 3: Pipeline de procesamiento GDELT

Fase	Script	Directorio Salida	Tamaño aproximado	Tiempo estimado
Descarga de CSV.ZIP	code/R/11_web_scraping_download_gdelt_parallel.R	lgdelt_raw	200Gb aprox	3-4h
Conversión a parquet	code/sh/step1_zip_to_parquet.sh & convert_zip_to_parquet.R	gdelt_parquet	200Gb aprox	4-6h
Filtrado	code/R/12.filter_parquet_files.R	gdelt_filtered	60Mb aprox	1-2h
Consolidación	code/R/14.consolidate_filtered_batches.R	lgdelt_filtered_consolidated	40Mb aprox	1-2 min aprox

: Esquema de descarga, filtrado y consolidación de archivos desde GDELT

Para el filtrado, se hace uso de la librería `duckdb`, para evitar falsos positivos y reducir el ruido, se establecen tres filtros de los que ha de cumplir al menos 2 para ser considerada noticia válida.

Para garantizar que las noticias procesadas fueran relevantes al mercado español, se implementó un sistema de filtrado, haciendo uso de la libreria `duckdb`, con tres criterios booleanos independientes, requiriendo que cada artículo cumpliera al menos dos de ellos (`num_criterios_ibex35 >= 2`). Este umbral evita tanto falsos positivos por menciones tangenciales como la exclusión de noticias relevantes que no cumplan los tres criterios simultáneamente. El código completo se encuentra en `code/R/12.filter_parquet_files.R` y `code/R/13.filter_script.R`.

El primer criterio evalúa temas económicos mediante expresiones regulares en el campo THEMES de GDELT, identificando menciones de mercados bursátiles

(ECON\_STOCKMARKET, WB\_\*STOCK), contexto geográfico español (SPAIN, SPANISH, IBEX, MADRID.\*EXCHANGE) y la infraestructura del mercado (BME). El segundo criterio busca nombres de empresas del IBEX35 en el campo ORGANIZATIONS, incluyendo las principales compañías del índice (Santander, BBVA, Telefónica, Iberdrola, Inditex, Repsol, entre otras) así como términos institucionales del mercado español (Bolsa de Madrid, BME, Latibex). El tercer criterio filtra por fuentes de información españolas, tanto por dominio (.es en SOURCEURLS) como por medios específicos en SOURCES (El País, El Mundo, Expansión, Cinco Días, Europa Press, EFE, etc.), asegurando cobertura desde perspectiva nacional.

Esta aproximación multicriterio permite capturar tanto noticias corporativas específicas como análisis macroeconómicos del mercado español, maximizando la señal de sentimiento relevante para la predicción del IBEX35.

## 5.4 Modelos aplicados

La predicción de series temporales financieras presenta desafíos únicos que requieren una selección cuidadosa de algoritmos. Los mercados financieros se caracterizan por:

- Alta dimensionalidad: Múltiples indicadores técnicos, fundamentales y de sentimiento
- No estacionariedad: Las relaciones estadísticas cambian con el tiempo
- Baja relación señal/ruido: Las señales predictivas son débiles comparadas con el ruido
- No linealidad: Relaciones complejas entre variables que los modelos lineales no capturan
- Dependencias temporales: El contexto histórico importa para predecir el futuro

Este proyecto implementa una batería de 10 modelos que abarcan desde enfoques lineales simples (baselines) hasta arquitecturas deep learning especializadas en secuencias temporales, permitiendo una comparación sistemática de su capacidad predictiva.

Para la predicción de los valores del IBEX35, establecemos dos problemas diferentes:

Por un lado un problema de clasificación, donde trataremos de predecir la dirección de los retornos futuros `direction_next`, positivo o negativo, a 1, 5, 10 y 20 días.

Por otro lado, un problema de regresión, donde se tratarán de predecir los valores de retornos del IBEX35 `returns_next` a 1, 5, 10 y 20 días.

Table 4: Configuración de modelos por tipo de problema

Problema	Datos	Modelos	N
Regresión	Scaled	Linear, Ridge, Random Forest, XGBoost, LightGBM, Ensemble, Multi-Layer Perceptron, Gated Recurrent Unit, Long Short-Term Memory	9
Regresión	Unscaled	Random Forest, XGBoost, LightGBM, Ensemble, Gated Recurrent Unit, Long Short-Term Memory	6
Clasificación	Scaled	Logistic, Random Forest, XGBoost, LightGBM, Ensemble, Multi-Layer Perceptron, Gated Recurrent Unit, Long Short-Term Memory	8

Clasificación	Unscaled	Random Forest, XGBoost, LightGBM, Ensemble, Gated Recurrent Unit, Long Short-Term Memory	6
---------------	----------	---------------------------------------------------------------------------------------------	---

: Modelos propuestos por tipo de problema y escalado Z-score de datos

### 5.4.1 Modelos Lineales

#### 5.4.1.1 Regresión Lineal

##### 5.4.1.1.1 Descripción

La regresión lineal modela la relación entre features y target como una combinación lineal:

$$\hat{y} = \beta_0 + \sum_{i=1}^p \beta_i x_i$$

donde los coeficientes  $\beta$  se estiman minimizando el error cuadrático medio.

##### 5.4.1.1.2 Justificación en Series Financieras

- **Baseline interpretable:** Proporciona una referencia contra la cual medir modelos más complejos
- **Interpretabilidad:** Los coeficientes indican directamente la contribución marginal de cada variable
- **Velocidad:** Entrenamiento instantáneo.

##### 5.4.1.1.3 Limitaciones

- Asume relaciones lineales (raramente válido en finanzas)
- Sensible a multicolinealidad entre indicadores técnicos
- Sin regularización, tiende a sobreajustar con muchas features
- No captura interacciones entre variables

#### 5.4.1.2 Ridge Regression

##### 5.4.1.2.1 Descripción

Ridge añade regularización L2 a la regresión lineal, penalizando coeficientes grandes:

$$\min_{\beta} \left\{ \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^p \beta_j^2 \right\}$$

El término  $\alpha$  controla la fuerza de regularización: valores altos producen coeficientes más pequeños y un modelo más simple.

#### 5.4.1.2.2 Justificación en Series Financieras

- **Control de multicolinealidad:** Los indicadores técnicos suelen estar altamente correlacionados (RSI, estocástico, momentum). Ridge estabiliza las estimaciones en presencia de multicolinealidad
- **Reducción de sobreajuste:** La penalización previene que el modelo se ajuste excesivamente al ruido histórica
- **Baseline regularizado:** Referencia más robusta que regresión lineal pura

#### 5.4.1.2.3 Comparación con Lasso

Ridge (L2) mantiene todas las features con coeficientes reducidos, mientras que Lasso (L1) produce selección de features (coeficientes exactamente cero). En finanzas, donde muchas features pueden tener señales débiles pero acumulativas, Ridge suele preferirse.

#### 5.4.1.2.4 Configuración Implementada

```
Ridge(alpha=1.0,           # Fuerza de regularización  
      random_state=42)
```

### 5.4.1.3 Regresión Logistica

#### 5.4.1.3.1 Descripción

La regresión logística modela la probabilidad de clase positiva mediante la función sigmoidal:

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \sum \beta_i x_i)}}$$

Se entrena maximizando la log-verosimilitud (equivalente a minimizar binary cross-entropy).

#### 5.4.1.3.2 Justificación en Series Financieras

- **Probabilidades calibradas:** Proporciona probabilidades interpretables, no solo clasificaciones
- **Baseline para clasificación:** Análogo a Linear/Ridge para regresión
- **Robustez:** Menos propensa a sobreajuste que modelos más complejos
- **Interpretabilidad:** Log-odds ratios indican importancia relativa de features

#### 5.4.1.3.3 Threshold y Trading

El umbral estándar de 0.5 puede no ser óptimo para trading. Un análisis de curva ROC permite seleccionar umbrales que maximicen métricas específicas (precision para evitar falsas señales, recall para no perder oportunidades).

#### 5.4.1.3.4 Configuración Implementada

```
LogisticRegression(  
    max_iter=1000,          # Iteraciones suficientes para convergencia  
    random_state=42,  
    n_jobs=-1)
```

### 5.4.2 Modelos Basados en Árboles

#### 5.4.2.1 Random Forest

##### 5.4.2.1.1 Descripción

Random Forest es un ensemble de árboles de decisión entrenados con bagging (bootstrap aggregating) y selección aleatoria de features. Cada árbol: 1. Se entrena en una muestra bootstrap del dataset 2. En cada split, considera solo un subconjunto aleatorio de features 3. Crece hasta profundidad máxima o criterio de parada

La predicción final es el promedio (regresión) o voto mayoritario (clasificación) de todos los árboles.

##### 5.4.2.1.2 Justificación en Series Financieras

- **Captura no linealidades:** Los splits de árboles modelan relaciones no lineales e interacciones automáticamente
- **Robustez al ruido:** El promediado de múltiples árboles reduce varianza
- **Importancia de features:** Proporciona rankings de importancia basados en reducción de varianza
- **Sin supuestos distribucionales:** No asume normalidad ni linealidad
- **Manejo de outliers:** Menos sensible a valores extremos que modelos lineales
- **Invariante a escala:** No requiere normalización de features

##### 5.4.2.1.3 Limitaciones en Finanzas

- Tiende a sobreajustar con series temporales (datos no independientes e idénticamente distribuidos)
- No extrapola bien fuera del rango de entrenamiento
- Las predicciones están acotadas al rango histórico de targets
- Puede memorizar patrones espurios en datos ruidosos

##### 5.4.2.1.4 Configuración Implementada

```
RandomForestRegressor( # o Classifier  
    n_estimators=200,      # Número de árboles  
    max_depth=10,         # Profundidad máxima. Limita complejidad  
    min_samples_split=10,  # Mínimo para split. Previene splits con pocos  
    ↵ datos               # Features por split. Decorrelaciona árboles  
    max_features='sqrt',  # Features por split. Decorrelaciona árboles
```

```
random_state=42,  
n_jobs=-1)
```

#### 5.4.2.2 XGBoost

##### 5.4.2.2.1 Descripción

XGBoost (eXtreme Gradient Boosting) (Chen and Guestrin 2016) construye árboles secuencialmente, donde cada nuevo árbol corrige los errores residuales de los anteriores. Implementa:

- **Gradient boosting:** Optimiza función de pérdida mediante gradiente descendente en espacio funcional
- **Regularización:** Términos L1 y L2 en la función objetivo
- **Pruning:** Poda de árboles basada en ganancia
- **Optimizaciones:** Histogramas aproximados, manejo de valores faltantes, paralelización

##### 5.4.2.2.2 Justificación en Series Financieras

- **Estado del arte en competiciones:** Consistentemente entre los mejores en problemas tabulares
- **Flexibilidad:** Funciones de pérdida personalizables (útil para métricas financieras)
- **Regularización integrada:** Control de sobreajuste sin preprocesamiento adicional
- **Eficiencia:** Más rápido que Random Forest para exactitud comparable
- **Feature importance:** Múltiples métodos (gain, cover, frequency)
- **Manejo de desbalance:** Parámetro scale\_pos\_weight para clases desbalanceadas

##### 5.4.2.2.3 Riesgos de Sobreajuste

El boosting, al ser un método aditivo, puede sobreajustar si no se controla adecuadamente. La configuración debe incluir: - Early stopping basado en validación - Limitación de profundidad y número de estimadores - Subsampling de datos y features

##### 5.4.2.2.4 Configuración Implementada

```
XGBRegressor( # o XGBClassifier  
    n_estimators=200, # Número de rondas de boosting  
    max_depth=6, # Profundidad máxima. Limita complejidad, reduce  
    ↵ sobreajuste. Más conservador que RF (boosting acumula complejidad)  
    learning_rate=0.1, # Shrinkage estándar  
    subsample=0.8, # Stochastic gradient boosting  
    colsample_bytree=0.8, # Decorrelación similar a RF  
    random_state=42,  
    n_jobs=-1)
```

#### 5.4.2.3 LightGBM

#### 5.4.2.4 Descripción

LightGBM (Light Gradient Boosting Machine) (Ke et al. 2017) es una implementación de gradient boosting optimizada para eficiencia:

- **Gradient-based One-Side Sampling (GOSS)**: Mantiene instancias con gradientes grandes, submuestrea las demás
- **Exclusive Feature Bundling (EFB)**: Agrupa features mutuamente excluyentes
- **Histogram-based splitting**: Discretiza valores continuos en bins
- **Leaf-wise growth**: Crece el árbol por hojas (vs. level-wise), más eficiente pero riesgo de sobreajuste

##### 5.4.2.4.1 Justificación en Series Financieras

- **Velocidad**: 2-10x más rápido que XGBoost en datasets grandes
- **Memoria eficiente**: Importante para backtesting con muchos periodos
- **Manejo de categorías**: Soporte nativo para features categóricas (sectores, días de semana)
- **Performance competitiva**: Similar o superior a XGBoost en muchos benchmarks
- **Escalabilidad**: Permite experimentación rápida con muchas configuraciones

##### 5.4.2.4.2 Consideraciones

El crecimiento leaf-wise puede producir árboles desbalanceados que sobreajustan en datasets pequeños. El parámetro `num_leaves` debe calibrarse cuidadosamente.

##### 5.4.2.4.3 Configuración Implementada

```
LGBMRegressor( # o LGBMClassifier
    n_estimators=200, # Número de rondas de boosting
    max_depth=6, # Profundidad máxima. Limita complejidad, reduce
    ↵ sobreajuste. Más conservador que RF (boosting acumula complejidad)
    learning_rate=0.1, # Shrinkage estándar
    num_leaves=31, # Control de complejidad.
    feature_fraction=0.8, # # Decorrelación similar a RF. Equivalente a
    ↵ colsample_bytree
    bagging_fraction=0.8, # #Stochastic gradient boosting. Equivalente a
    ↵ subsample
    bagging_freq=5, # Frecuencia de bagging
    random_state=42
    verbose=-1, # Suprime warnings
    n_jobs=-1)
```

#### 5.4.2.5 Ensemble (Stacking)

##### 5.4.2.5.1 Descripción

Stacking combina múltiples modelos base mediante un meta-modelo que aprende a ponderar sus predicciones:

- Nivel 0 (Base):** Modelos diversos se entrenan en los datos originales
- Nivel 1 (Meta):** Un modelo final se entrena usando las predicciones de nivel 0 como features

Se usa validación cruzada para generar predicciones out-of-fold, evitando data leakage.

#### 5.4.2.5.2 Justificación en Series Financieras

- Diversificación de modelos:** Análogo a diversificación de portafolio
- Reducción de varianza:** Diferentes modelos capturan diferentes patrones
- Robustez:** Menos dependiente de un único algoritmo
- Aprovecha fortalezas complementarias:** RF captura interacciones, XGBoost errores residuales, LightGBM patrones sutiles

#### 5.4.2.5.3 Configuración Implementada

```
# Regresión
StackingRegressor(
    estimators=[
        ('rf', RandomForestRegressor(n_estimators=100, max_depth=10)),
        ('xgb', XGBRegressor(n_estimators=100, max_depth=6)),
        ('lgbm', LGBMRegressor(n_estimators=100, max_depth=6))
    ],
    final_estimator=Ridge(alpha=1.0),
    cv=5,
    n_jobs=-1)
# Clasificación
StackingClassifier(
    estimators=[...], # Mismos base learners
    final_estimator=LogisticRegression(max_iter=1000),
    cv=5,
    n_jobs=-1)
```

### 5.4.3 Redes Neuronales Clásicas

#### 5.4.3.1 MLP (Multi-Layer Perceptron)

##### 5.4.3.1.1 Descripción

El Perceptrón Multicapa es una red neuronal feedforward con capas ocultas completamente conectadas. Cada neurona computa:

$$h_j = \sigma \left( \sum_i w_{ij} x_i + b_j \right)$$

donde  $\sigma$  es una función de activación no lineal (ReLU en este caso).

### 5.4.3.2 Justificación en Series Financieras

- **Aproximador universal:** Puede representar cualquier función continua (teorema de aproximación universal)
- **Captura no linealidades:** Las activaciones permiten modelar relaciones complejas
- **Flexibilidad arquitectónica:** Fácil ajustar capacidad añadiendo capas/neuronas
- **Baseline de deep learning:** Referencia para comparar con arquitecturas más sofisticadas

### 5.4.3.3 Limitaciones para Series Temporales

- **No considera orden temporal:** Trata cada observación independientemente
- **No modela dependencias:** No captura patrones secuenciales (para eso están RNN/LSTM)
- **Requiere features engineered:** Necesita indicadores técnicos precalculados con look-backs

#### 5.4.3.3.1 Configuración Implementada

```
MLPRegressor( # o MLPClassifier  
    hidden_layer_sizes=(100, 50), #Arquitectura piramidal, reducción gradual  
    activation='relu',  
    solver='adam',  
    learning_rate_init=0.001, #Learning rate estándar para Adam  
    max_iter=500,  
    early_stopping=True, #Previene sobreajuste  
    validation_fraction=0.2, #20% para monitorear generalización. Proporcion  
    ↵ de datos usados para validación  
    random_state=42)
```

## 5.4.4 Redes Neuronales Recurrentes

### 5.4.4.1 GRU (Gated Recurrent Unit)

#### 5.4.4.1.1 Descripción

GRU (Cho et al. 2014) es una arquitectura recurrente diseñada para capturar dependencias temporales mediante mecanismos de compuerta:

- **Update gate (z):** Controla cuánta información del estado anterior mantener
- **Reset gate (r):** Determina cuánta información pasada olvidar
- **Candidate hidden state:** Nuevo estado candidato combinando input actual e historia

$$\begin{aligned} z_t &= \sigma(W_z x_t + U_z h_{t-1}) \\ r_t &= \sigma(W_r x_t + U_r h_{t-1}) \\ \tilde{h}_t &= \tanh(W x_t + U(r_t \odot h_{t-1})) \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \end{aligned}$$

#### 5.4.4.1.2 Justificación en Series Financieras

- **Memoria selectiva:** Las compuertas aprenden qué información histórica es relevante
- **Eficiencia:** Menos parámetros que LSTM (2 compuertas vs 3)
- **Gradientes estables:** Arquitectura diseñada para evitar vanishing/exploding gradients
- **Patrones temporales:** Captura dependencias como momentum, mean reversion, estacionalidad

#### 5.4.4.1.3 Comparación con LSTM

Aspecto	GRU	LSTM
Compuertas	2 (update, reset)	3 (input, forget, output)
Parámetros	Menos	Más
Velocidad	Más rápido	Más lento
Performance	Similar en muchos casos	Mejor en dependencias muy largas
Memoria	Menor	Mayor

#### 5.4.4.1.4 Arquitecturas Implementadas

**Small** (pruebas rápidas):

```
GRU(32) → Dropout(0.4) → Dense(16) → Dropout(0.3) → Output(1)  
Epochs: 50 | Batch: 16
```

**Medium** (balance):

```
GRU(50, return_seq) → Dropout(0.3) → GRU(32) → Dropout(0.3) → Dense(16) →  
↪ Output(1)  
Epochs: 100 | Batch: 32
```

**Large** (máxima capacidad):

```
GRU(64, return_seq) → Dropout(0.2) → GRU(64, return_seq) → Dropout(0.2) →  
GRU(32) → Dropout(0.2) → Dense(16) → Output(1)  
Epochs: 150 | Batch: 32
```

#### 5.4.4.1.5 Configuración Implementada

```
# Arquitectura Medium (default)  
Sequential([  
    GRU(50, activation='relu', return_sequences=True,  
        input_shape=(lookback, n_features)),  
    Dropout(0.3),  
    GRU(32, activation='relu'),  
    Dropout(0.3),  
    Dense(16, activation='relu'),
```

```

        Dense(1, activation='linear') # o 'sigmoid' para clasificación
    ])
# Para regresión
model.compile(
    optimizer=Adam(learning_rate=0.001),
    loss='mse',
    metrics=['mae'])
# Para clasificación
model.compile(
    optimizer=Adam(learning_rate=0.001),
    loss='binary_crossentropy',
    metrics=['accuracy'])

```

Table 6: Configuración de hiperparámetros para GRU

Hiperparámetro	Valor	Justificación
lookback	20	~1 mes de trading, captura ciclos cortos
dropout	0.3	Regularización para prevenir sobreajuste
optimizer	Adam	Adaptativo, robusto
learning_rate	0.001	Estándar, con ReduceLROnPlateau

#### 5.4.4.2 LSTM (Long Short-Term Memory)

##### 5.4.4.2.1 Descripción

LSTM (Hochreiter and Schmidhuber 1997) es la arquitectura recurrente más utilizada para series temporales, diseñada explícitamente para aprender dependencias a largo plazo. Introduce una “celda de memoria” con tres compuertas:

- **Forget gate (f):** Decide qué información descartar de la celda
- **Input gate (i):** Decide qué nueva información almacenar
- **Output gate (o):** Decide qué parte del estado de celda usar como salida

$$\begin{aligned}
 f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) \\
 i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\
 \tilde{C}_t &= \tanh(W_C[h_{t-1}, x_t] + b_C) \\
 C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\
 o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\
 h_t &= o_t \odot \tanh(C_t)
 \end{aligned}$$

##### 5.4.4.2.2 Justificación en Series Financieras

- **Dependencias a largo plazo:** Puede recordar información de 100+ timesteps atrás
- **Estado del arte histórico:** Arquitectura dominante para secuencias antes de Transformers
- **Patrones complejos:** Captura regímenes de mercado, ciclos económicos
- **Flexibilidad:** Múltiples variantes (stacked, bidirectional, attention)
- **Robustez:** Bien estudiado, comportamiento predecible

#### 5.4.4.2.3 Arquitecturas Implementadas

**Small** (baseline):

```
LSTM(32, tanh) → Dropout(0.3) → Dense(16, relu) → Dropout(0.2) → Output(1)
Params: ~5K | Epochs: 50 | Batch: 16
```

**Medium** (recomendado):

```
LSTM(50, return_seq) → Dropout(0.3) → LSTM(32) → Dropout(0.3) → Dense(16) →
↪ Output(1)
Params: ~15K | Epochs: 100 | Batch: 32
```

**Large** (alta capacidad):

```
LSTM(64, return_seq) → Dropout(0.2) → LSTM(64, return_seq) → Dropout(0.2) →
LSTM(32) → Dropout(0.2) → Dense(16) → Output(1)
Params: ~50K | Epochs: 150 | Batch: 32
```

**XLarge** (máxima capacidad):

```
LSTM(128, return_seq) → Dropout(0.2) → LSTM(64, return_seq) → Dropout(0.2) →
LSTM(64, return_seq) → Dropout(0.2) → LSTM(32) → Dropout(0.2) → Dense(16) →
↪ Output(1)
Params: ~150K | Epochs: 200 | Batch: 64
```

**Bidirectional** (contexto completo):

```
Bidirectional(LSTM(50, return_seq)) → Dropout(0.3) →
Bidirectional(LSTM(32)) → Dropout(0.3) → Dense(16) → Output(1)
Params: ~40K | Epochs: 100 | Batch: 32
```

#### 5.4.4.2.4 Configuración Implementada

```
# Arquitectura Medium (default)
Sequential([
    LSTM(50, activation='tanh', return_sequences=True,
        input_shape=(lookback, n_features)),
    Dropout(0.3),
```

```

LSTM(32, activation='tanh'),
Dropout(0.3),
Dense(16, activation='relu'),
Dense(1, activation='linear') # o 'sigmoid' para clasificación
])
# Compilación
model.compile(
    optimizer=Adam(learning_rate=0.001),
    loss='mse', # o 'binary_crossentropy' para clasificación
    metrics=['mae'] # o 'accuracy' para clasificación
)
# Callbacks
callbacks = [
    EarlyStopping(monitor='val_loss', patience=20,
    ↵ restore_best_weights=True),
    ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=10,
    ↵ min_lr=1e-6),
    ModelCheckpoint(filepath='best_model.keras', save_best_only=True)
]

```

Table 7: Configuración de hiperparámetros para LSTM ##### Creación de Secuencias

Hiperparámetro	Valor	Justificación
lookback	20	Ventana de ~1 mes de trading
activation (LSTM)	tanh	Estándar para LSTM, rango [-1, 1]
activation (Dense)	relu	Permite representaciones no lineales
dropout	0.2-0.3	Regularización entre capas
optimizer	Adam	Adaptativo con momentum
learning_rate	0.001	Estándar, reducido dinámicamente
patience (early_stop)	20	Permite convergencia antes de parar
patience (reduce_lr)	10	Reduce LR si plateau
factor (reduce_lr)	0.5	Reducción moderada

## 6 Resultados

El proyecto ha sido estructurado en varias fases:

### 6.1 Fase 1: Descarga y consolidación de datos.

#### 6.1.1 Descarga de datos financieros

Hemos hecho uso de la librería yfinance de Python para la descarga los datos. En RStudio, se ha usado reticulate para poder combinar en un mismo documento Quattro tanto código R como Python. Hemos usado los siguientes índices, ya señalados en el capítulo de Introducción:

índices usados, almacenados en un diccionario stocks\_to\_download, guardado en code/py/stocks\_list.py

```
stocks_to_download = [
    {"ticker": "^IBEX", "name": "ibex35"}, 
    {"ticker": "^GSPC", "name": "s_p500"}, 
    {"ticker": "^GDAXI", "name": "dax"}, 
    {"ticker": "^SBF120", "name": "sbfi20"}, 
    {"ticker": "^N100", "name": "euro_next100"}, 
    {"ticker": "^N225", "name": "nikkei225"}, 
    {"ticker": "^FTSE", "name": "ftse100"}, 
    {"ticker": "^VIX", "name": "volatility_index"}, 
    {"ticker": "EURUSD=X", "name": "euro_dollar"}, 
    {"ticker": "GC=F", "name": "gold"}, 
    {"ticker": "CL=F", "name": "oil"}]
```

Se genera la función stocks\_downloader, almacenada en code/py/stocks\_downloader.py

```
# Importar script como módulo
import sys
sys.path.append("code/py")
import stocks_downloader
from stocks_list import stocks_to_download, ibex_selected_companies
# Diccionario para guardar resultados
results_indexes = {}
# Bucle general
for stock in stocks_to_download:
    ticker = stock["ticker"]
    name = stock["name"]
    print(f"Descargando {name} ({ticker})...")
    df = stocks_downloader.download_ibex(ticker, "2004-01-01", "2025-10-16",
    ↵ False)
    # Guardar en el diccionario, clave = nombre del índice
    results_indexes[name] = df
```

Una vez descargados, los datos son transferidos a R haciendo uso del script code/R/03.transfer\_py\_to\_r.R. Este script formatea el data.frame de python a data.frame de R y carga los datos en la memoria de RStudio con reticulate::py\_to\_r(py\$results\_indexes), ajusta los nombres de columnas. Finalmente, guarda en formato \*.RDS, \*.xlsx y \*.parquet.

### 6.1.2 Descarga de datos desde GDELT

Una vez descargados los datos financieros, procedemos a descargar los datos análisis de sentimientos desde GDELT. El script usado es code/R/11.web\_scraping\_download\_gdelt\_parallel.R

En nuestro caso elegimo la fecha inicio:2014-01-01 y la fecha de fin: 2025-10-16, dando lugar a 4307 días. Dado que la actualización de datos en GDELT es diaria, el script genera una lista de fechas diarias entre las fecha de inicio y fin. Luego genera otra lista

con las fechas de archivos válidos descargados. Finalmente, descarga sólo las fechas faltantes. En una versión anteriormente ejecutada, ya se descargaron 2342 archivos por lo que en esta captura se han descargado 1941 archivos. Los archivos disponibles en GDELT para su descarga están en formato \*.CSV.ZIP. Esta descarga da lugar a 4283 archivos con un total de **144.9Gb**. Para un manejo más eficiente de estos datos, se procede a su transformación a formato parquet. En pruebas previas, al intentar ejecutar el script desde RStudio, este mostraba falta de memoria interrumpiendo la ejecución, por lo que debido a estas limitaciones de hardware, se procedió a ejecutar el script R desde la terminal Linux. Para ello se ejecuta `code/sh/step1_zip_to_parquet.sh` desde la terminal con `bash step1_zip_to_parquet.sh`. Esto ejecuta internamente el script `code/R/convert_zip_to_parquet.R` que convertirá los \*.zip a \*.parquet, formato más optimizado para grandes conjuntos de datos. En el archivo `step1_zip_to_parquet.sh`, podemos editar los siguientes parámetros:

```
#Editar las siguientes variables
INPUT_DIR="/mnt/NTFS/gdelt_consolidated/gdelt_raw"
OUTPUT_DIR="/mnt/NTFS/gdelt_consolidated/gdelt_parquet"
LOG_FILE="step1_zip_to_parquet.log"
NUM_CORES=2 # 2 cores para 16GB RAM
```

La conversión a formato parquet da lugar a 4283 archivos con un total de 189.7Gb. Posteriormente a la conversión a parquet, filtramos los datos en cada uno de los 4283 archivos usando los filtros explicados en materiales y métodos y ejecutados en el script `code/R/12.filter_parquet_files.R`. Los archivos filtrados, ocupan un tamaño total de 65.1Mb, una reducción del 99.6%. Una vez filtrados, mediante el script `core/R/14.consolidate_filtered_batches.R` procederemos a la consolidación de los datos filtrados en un único archivo parquet que usaremos durante todo el análisis. Este archivo final ocupa solo 30.8Mb, lo que unido a un formato eficiente hace que el procesamiento posterior sea mucho más rápido.

## 6.2 Fase 2: Análisis exploratorio de datos financieros y datos de sentimientos de GDELT

El análisis exploratorio se centró en la caracterización de las variables base del estudio: series temporales financieras del IBEX35 y sus componentes, índices internacionales de referencia, y datos agregados de noticias del proyecto GDELT (Global Database of Events, Language, and Tone). Este análisis establece el contexto necesario para el posterior desarrollo de características y variables avanzadas.

### 6.2.1 Análisis exploratorio de índices bursátiles

#### 6.2.1.1 Rango temporal de datos

El período de análisis seleccionado abarca desde enero de 2004 hasta octubre de 2025, proporcionando 21.8 años de datos históricos. Esta ventana temporal fue elegida considerando: (1) la disponibilidad y calidad de datos en Yahoo Finance, (2) la cobertura de eventos económicos significativos incluyendo la crisis financiera de 2008, la crisis de deuda europea (2011-2012), y la pandemia COVID-19, y (3) la necesidad de suficientes observaciones para entrenar modelos de deep learning.

Table 8: Rango temporal de datos para los índices bursátiles analizados

Índice	Fecha Inicio	Fecha Fin	Número de observaciones	Días de calendario	Años de datos
euro_dollar	2004-01-01	2025-10-15	5653	7958	21.79
dax	2004-01-02	2025-10-15	5537	7957	21.79
euro_next100	2004-01-02	2025-10-15	5576	7957	21.79
ftse100	2004-01-02	2025-10-15	5504	7957	21.79
ibex35	2004-01-02	2025-10-15	5561	7957	21.79
s_p500	2004-01-02	2025-10-15	5482	7957	21.79
volatility_index	2004-01-02	2025-10-15	5482	7957	21.79
gold	2004-01-05	2025-10-15	5474	7954	21.78
nikkei225	2004-01-05	2025-10-15	5332	7954	21.78
oil	2004-01-05	2025-10-15	5478	7954	21.78

#### 6.2.1.2 Gestión de valores faltantes por días no bursátiles

Se observaron dos gaps significativos en los valores de la cotización del cambio Euro/Dólar en agosto de 2008 de 8 y 18 días, coincidente con volatilidad extrema poco antes del colapso de Lehman Brothers. Además, se observa otro gap importante en la cotización del índice NIKKEI en Mayo de 2019, sociado a la sucesión imperial en Japón y el cierre prolongado de mercados.

Table 9: Detección de gaps en los datos de los índices bursátiles analizados

Índice	Fecha	Número de días
euro_dollar	2008-08-26	18
nikkei225	2019-05-07	11
euro_dollar	2008-08-08	8

#### 6.2.1.3 Detección de valores faltantes en OHLCV

No se detectaron valores faltantes en los valores OHLCV (Open, High, Low, Close, Volume)

==== VALORES FALTANTES EN ÍNDICES ===

No se detectaron valores faltantes (NA)

#### 6.2.1.4 Detección de días de trading esperados

Se detectaron diferencias mínimas (alrededor del 3%) en el número de días de trading con respecto al esperado probablemente atribuidas a festivos locales no reflejados en un calendario estandar de 252 días.

Table 10: Rango temporal de datos para los índices bursátiles analizados

Índice	Fecha Inicio	Fecha Fin	N observado	N esperado	%	Diferencia
nikkei225	2004-01-05	2025-10-15	5332	5487.77	97.16	155.77
gold	2004-01-05	2025-10-15	5474	5487.77	99.75	13.77
oil	2004-01-05	2025-10-15	5478	5487.77	99.82	9.77
s_p500	2004-01-02	2025-10-15	5482	5489.84	99.86	7.84
volatility_index	2004-01-02	2025-10-15	5482	5489.84	99.86	7.84
ftse100	2004-01-02	2025-10-15	5504	5489.84	100.26	-14.16
dax	2004-01-02	2025-10-15	5537	5489.84	100.86	-47.16
ibex35	2004-01-02	2025-10-15	5561	5489.84	101.30	-71.16
euro_next100	2004-01-02	2025-10-15	5576	5489.84	101.57	-86.16
euro_dollar	2004-01-01	2025-10-15	5653	5490.53	102.96	-162.47

### 6.2.1.5 Detección de valores anómalos e inconsistencias lógicas

Se detectan valores anómalos en los datos descargados, tales como precios negativos o inconsistencias lógicas entre los valores de apertura, cierre, máximo y mínimo. Concretamente en los datos de cotización del petróleo (Oil) durante los días 20 y 21 de Abril de 2020. Esto se debió a momentos clave durante el confinamiento provocado por el COVID-19 a nivel mundial. Dicho confinamiento y parada casi total de la economía mundial provocó éstas oscilaciones del precio del petróleo.

Table 11: Detección de valores anómalos en los datos de los índices bursátiles analizados

Índice	Fecha	Open	High	Low	Close	Volume
oil	2020-04-20	17.73	17.85	-40.32	-37.63	247947
oil	2020-04-21	-14.00	13.86	-16.74	10.01	2288230

#### Resumen de la validación de datos

Período de datos: 2004-01-01 a 2025-10-15

Número de índices: 10

Total observaciones índices: 55,079

#### HALLAZGOS PRINCIPALES:

- Completitud promedio índices: 100.34%
- Gaps detectados en índices: 3
- Valores faltantes (NA) en índices: 0
- Valores anómalos (precio  $\leq 0$ ): 2
- Inconsistencias lógicas: 506

## Dashboard de Calidad de Datos

Detección de anomalías y problemas en series temporales

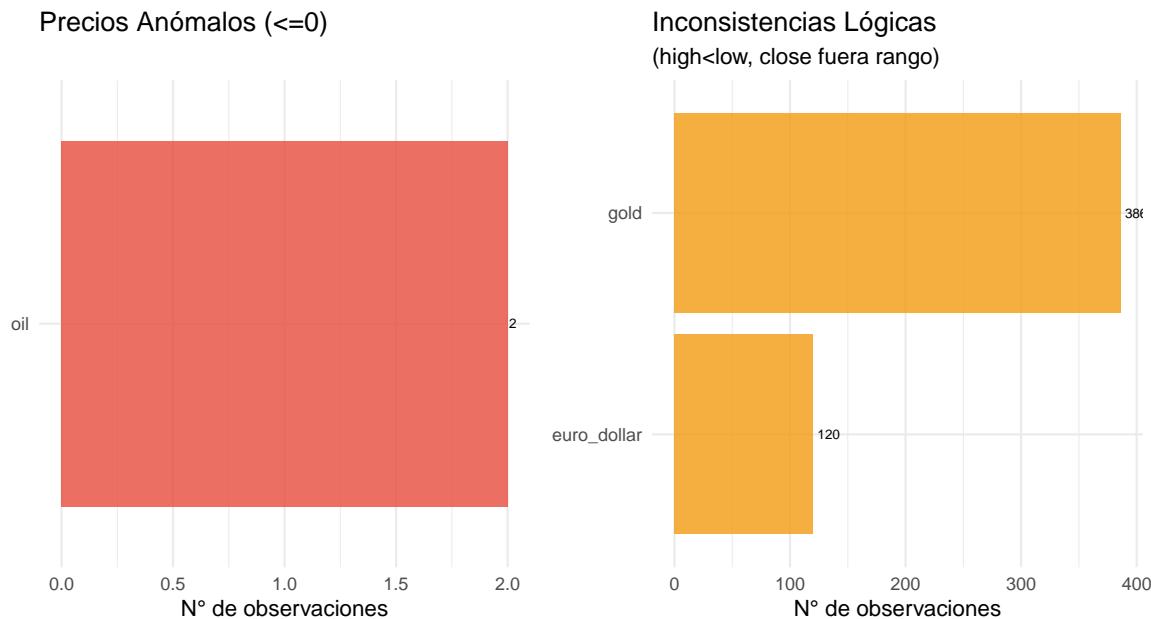


Figure 2: Anomalías de ausencia y lógicas

### 6.2.1.6 Análisis univariante de la serie IBEX35: Características de variables base

#### 6.2.1.6.1 Retornos, diferenciación, asimetría y Curtosis

Calculo de los retornos logarítmicos, primera diferencia (Velocidad) y segunda diferencia (Aceleración)

Primera derivada =  $d(\text{precio})/dt \approx \Delta \text{precio}/\Delta t$

**Por qué es importante:** Precio → Retornos → Velocidad → Aceleración (0<sup>a</sup>) (1<sup>a</sup>) (1<sup>a</sup>) (2<sup>a</sup>)

Table 12: Variables derivadas y sus fórmulas

Derivada	Variable	Formula	Significado
Retorno de precio	log_return_pct	$r_t = \ln(P_t/P_{t-1}) \times 100$	Retorno logarítmico porcentual. Mide el cambio porcentual continuo del precio. Ventajas: aditivos en el tiempo, simétricos (subida/bajada), asume capitalización continua.
Volatilidad_20	volatility_20	$\sigma_t = \sqrt{\frac{1}{n-1} \sum_{i=0}^{19} (r_{t-i} - \bar{r})^2}$	Dispersión/riesgo de retornos. rollapply() aplica la desviación estándar (sd) en ventanas deslizantes de 20 observaciones, alineadas a la derecha. Mayor volatilidad = mayor incertidumbre.
1 <sup>a</sup> de retornos	returns_velocity	$v_t = r_t - r_{t-1}$	Aceleración/desaceleración del momentum. diff(log_return_pct) calcula la diferencia entre retornos consecutivos. Mide si el momentum se acelera ( $v>0$ ) o desacelera ( $v<0$ ). Detecta cambios en la tendencia de los retornos.

Table 12: Variables derivadas y sus fórmulas (*continued*)

Derivada	Variable	Formula	Significado
1 <sup>a</sup> de volatilidad	volatility_velocity	$\Delta\sigma_t = \sigma_t - \sigma_{t-1}$	Tasa de cambio del riesgo. <code>diff(volatility_20)</code> calcula la diferencia entre volatilidades consecutivas. Indica si el mercado se vuelve más volátil ( $\Delta\sigma > 0$ ) o más estable ( $\Delta\sigma < 0$ ). Útil para régimenes de riesgo cambiantes.
1 <sup>a</sup> de volumen	volume_velocity	$\Delta V_t = \frac{V_t - V_{t-1}}{V_{t-1}} \times 100$	Cambio porcentual en actividad. <code>diff(volume) / head(volume, -1)</code> divide la diferencia de volúmenes por el volumen anterior. Expansión ( $>0$ ) o contracción ( $<0$ ) de la actividad de trading. Señala entrada/salida de liquidez.
2 <sup>a</sup> de retornos	returns_acceleration	$a_t = (r_t - r_{t-1}) - (r_{t-1} - r_{t-2})$	Curvatura del momentum. <code>diff(log_return_pct, differences = 2)</code> aplica <code>diff()</code> dos veces consecutivas. Detecta puntos de inflexión. Si $a > 0$ con $v < 0$ : posible suelo. Si $a < 0$ con $v > 0$ : posible techo. Anticipa cambios de régimen.

```

# Calcular retornos logarítmicos
ibex <- all_stocks_df |>
  dplyr::filter(index == "ibex35") |>
  dplyr::arrange(date) |>
  dplyr::mutate(
    # Retornos
    log_return_pct = c(NA, diff(log(close))) * 100,
    # Volatilidad: Riesgo
    volatility_20 = zoo::rollapply(log_return_pct, width = 20, FUN = sd,
                                   fill = NA, align = "right"),
    # 1a derivada de retornos: Momentum
    returns_velocity = c(NA, diff(log_return_pct)),
    # 1a derivada de volatilidad: Cambio en riesgo
    volatility_velocity = c(NA, diff(volatility_20)),
    # 1a derivada de volumen: Cambio en actividad
    volume_velocity = c(NA, diff(volume) / head(volume, -1)) * 100,
    # 2a derivada de retornos: Inflexiones
    returns_acceleration = c(NA, diff(returns_velocity)),
    # Añadir calendario
    weekday = weekdays(date),
    month = lubridate::month(date, label = TRUE, abbr = TRUE),
    quarter = quarter(date),
    weekday = factor(weekday,
                     levels = c("lunes", "martes", "miércoles", "jueves",
                               ↵ "viernes")),
    quarter = factor(quarter))
  
```

La distribución de retornos logarítmicos diarios presentó: - Media: 1.2% (ligeramente positiva) - Desviación estándar: 1.36% - Asimetría: -0.43 (sesgo negativo, colas izquierdas más pesadas) - Curtosis: 13.70 (exceso significativo respecto a distribución normal).

Table 13: Estadísticas de retornos IBEX35 2004-2025

Métrica	Crudo
Media	0.0122507
Mediana	0.0713125
Desv_Std	1.3591409
Asimetría	-0.4325973
Curtosis	13.6880664
Min	-15.1511759
Max	13.4836367

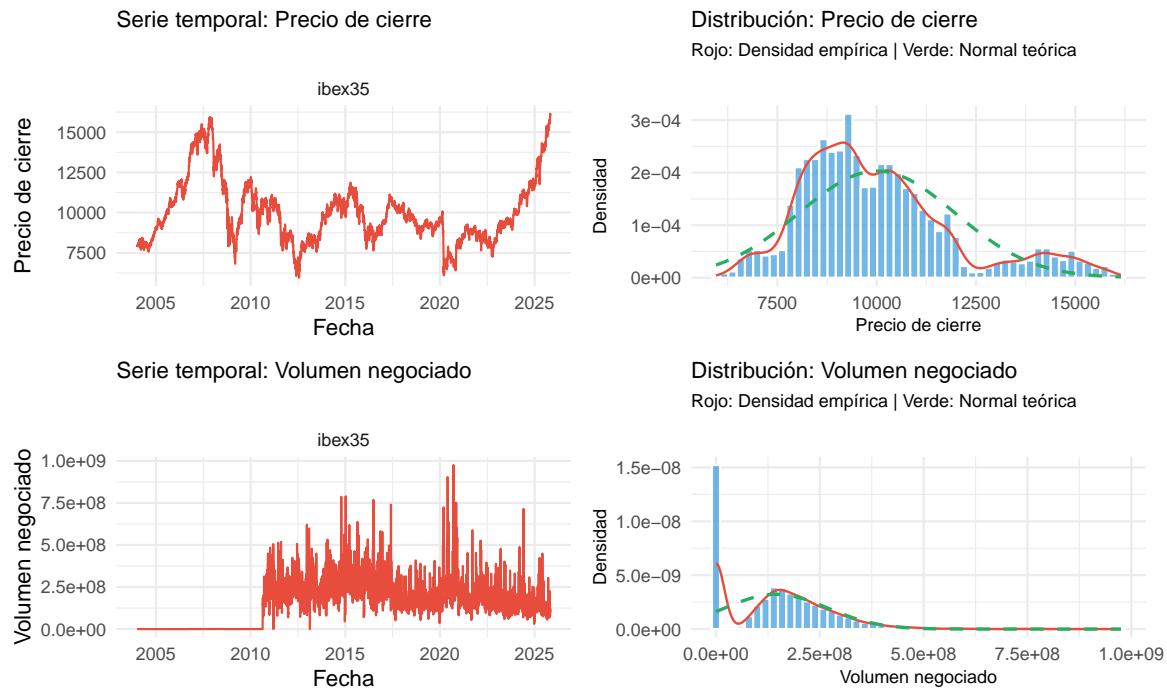


Figure 3: Distribución de precios y volumen negociado

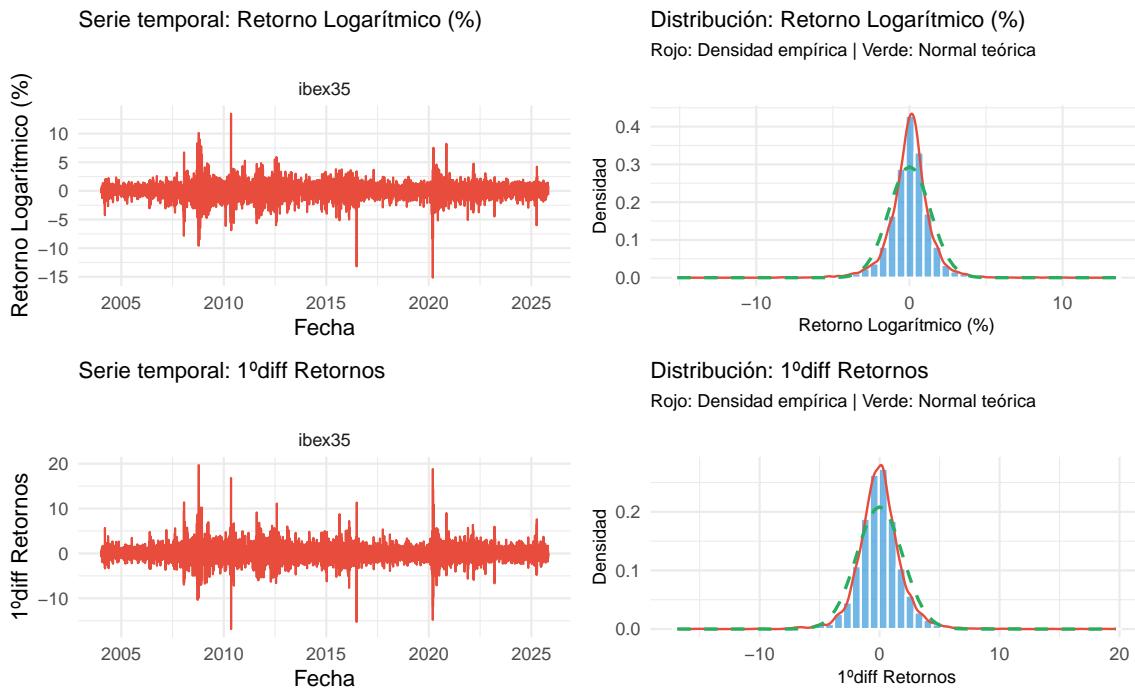


Figure 4: Distribución de retornos logarítmicos

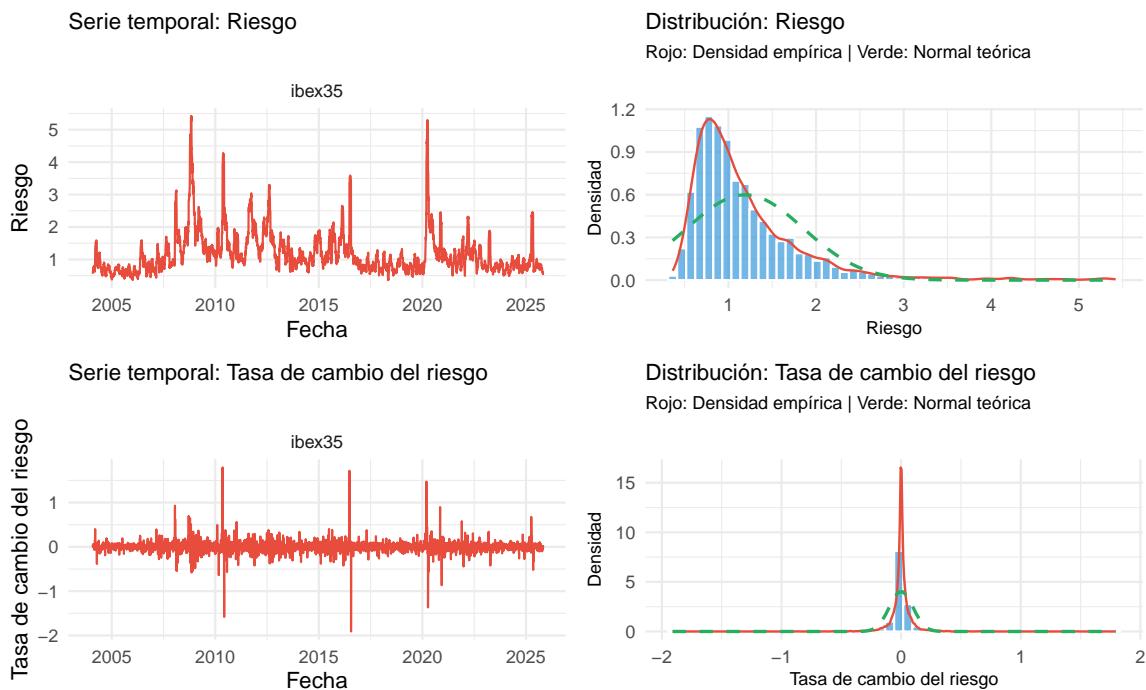


Figure 5: Distribución del riesgo y tasa de cambio del riesgo

#### 6.2.1.6.2 Volatilidad

El análisis de volatilidad rolling (ventanas de 10, 20 y 60 días) reveló tres períodos de volatilidad extrema:

Crisis financiera global (2008-2009): Volatilidad máxima de 5.2% diaria (60-day rolling), 8 veces superior a la media histórica. El periodo septiembre-noviembre 2008 concentró 18 de los 25 días con mayor volatilidad del dataset completo.

Crisis de deuda europea (2011-2012): Picos de volatilidad coincidentes con amenazas de quiebra soberana de Grecia e Italia. La volatilidad promedio en este periodo (2.4%) duplicó la del periodo 2004-2007.

Pandemia COVID-19 (marzo-abril 2020): Caída del 35% en el IBEX35 en 4 semanas, con volatilidad intradiaria alcanzando 6.8% diaria. Notable por su velocidad: la mayor caída porcentual en el menor tiempo registrado desde 1992.

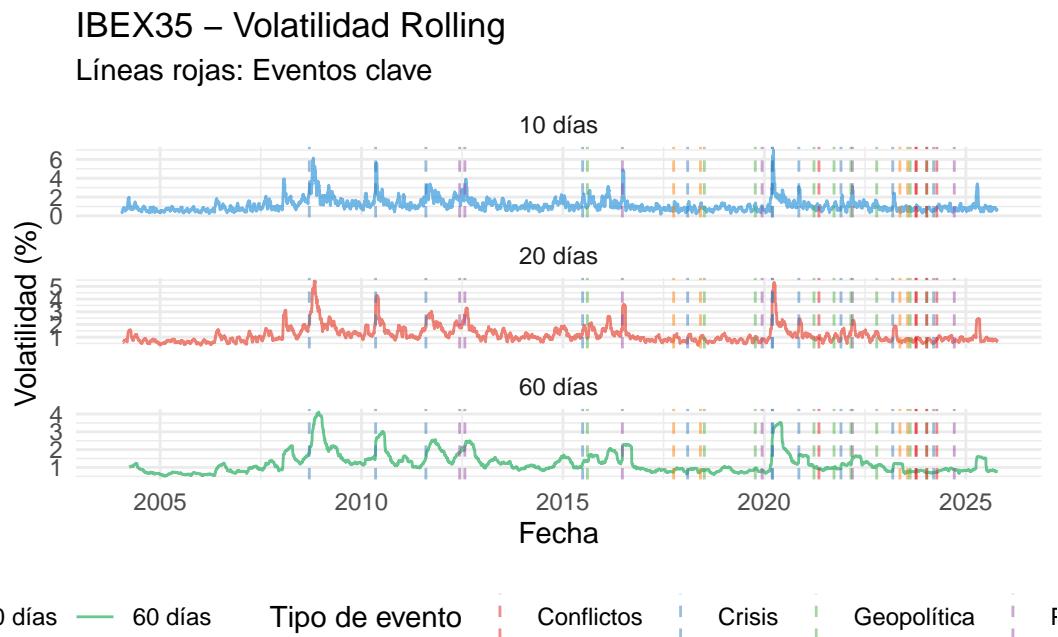


Figure 6: Momentos de máxima volatilidad

#### 6.2.1.6.3 Normalidad

El test de Jarque-Bera y Shapiro-Wilk rechazaron la normalidad ( $p < 0.001$ ), confirmando las colas pesadas características de series financieras. La presencia de curtosis elevada justifica el uso de modelos capaces de capturar eventos extremos más allá de supuestos gaussianos.

Shapiro-Wilk (muestra n=5000):

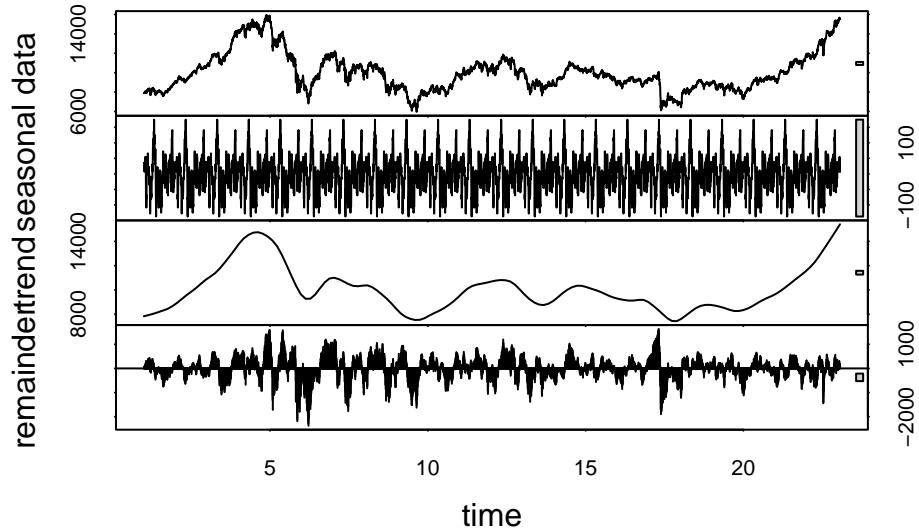
```
Shapiro-Wilk normality test
data: returns_sample
W = 0.91177, p-value < 2.2e-16
Jarque-Bera:
    Jarque Bera Test
data: returns_clean
X-squared = 26773, df = 2, p-value < 2.2e-16
```

Interpretación:

- Shapiro-Wilk: p-valor = 0.0000 → RECHAZA normalidad
- Jarque-Bera: p-valor = 0.0000 → RECHAZA normalidad

#### 6.2.1.6.4 Estacionalidad y patrones temporales

Se observa que no existe un patrón claro estacional y que el ruido es prácticamente aleatorio.



#### 6.2.1.6.5 Efectos de calendario

No se observa un cambio especial en los precios de cierre o retornos medios en relación con el día de la semana, mes o trimestre.

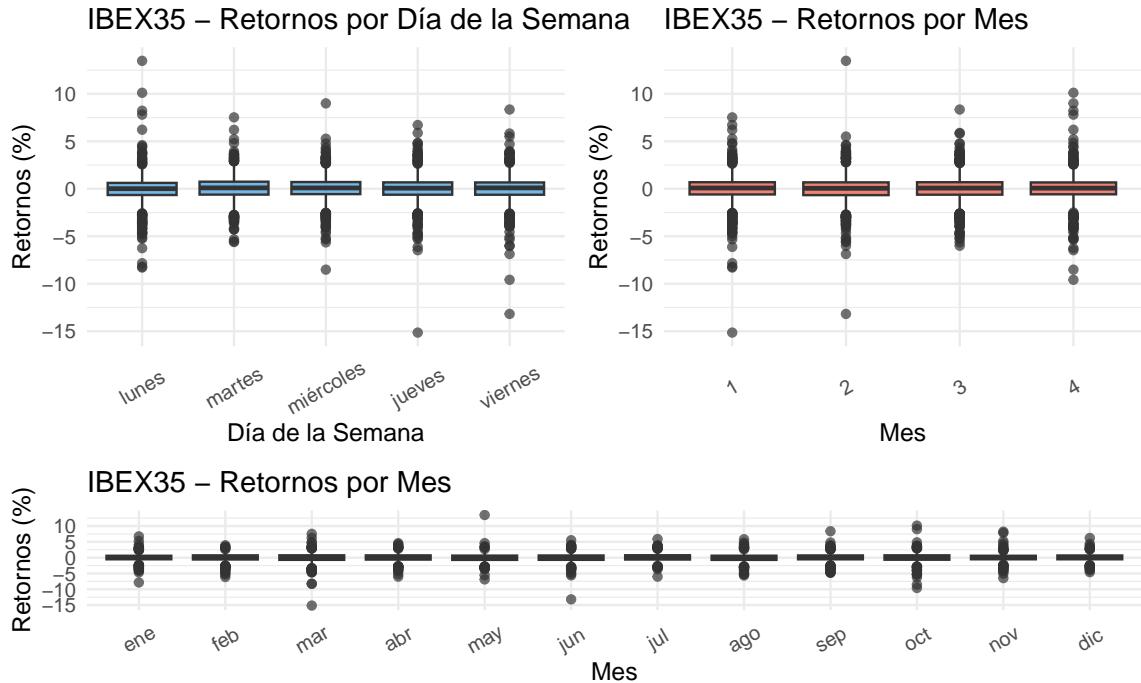


Figure 7: Efectos del calendario en el retorno logarítmico

#### 6.2.1.7 Análisis multivariante

##### 6.2.1.7.1 Correlaciones con otros índices

Se observa una correlación positiva con índices bursátiles europeos, DAX 0.82, Euro Next100 0.87 y FTSE 0.78. A su vez también se observa una correlación importante con el S&P500, 0.55, y una menor correlación con el petróleo (Panagiotidis and Rutledge 2010), 0.25 y la bolsa japonesa 0.31, quizás señal de una menor integración de las bolsas asiáticas con las europeas. Por otro lado, el indicador VIX conocido como “índice del miedo” (Poon and Granger 2003)], mostró correlación negativa con IBEX35 del -0.42.

## Matriz de Correlaciones – Índices Bursátiles

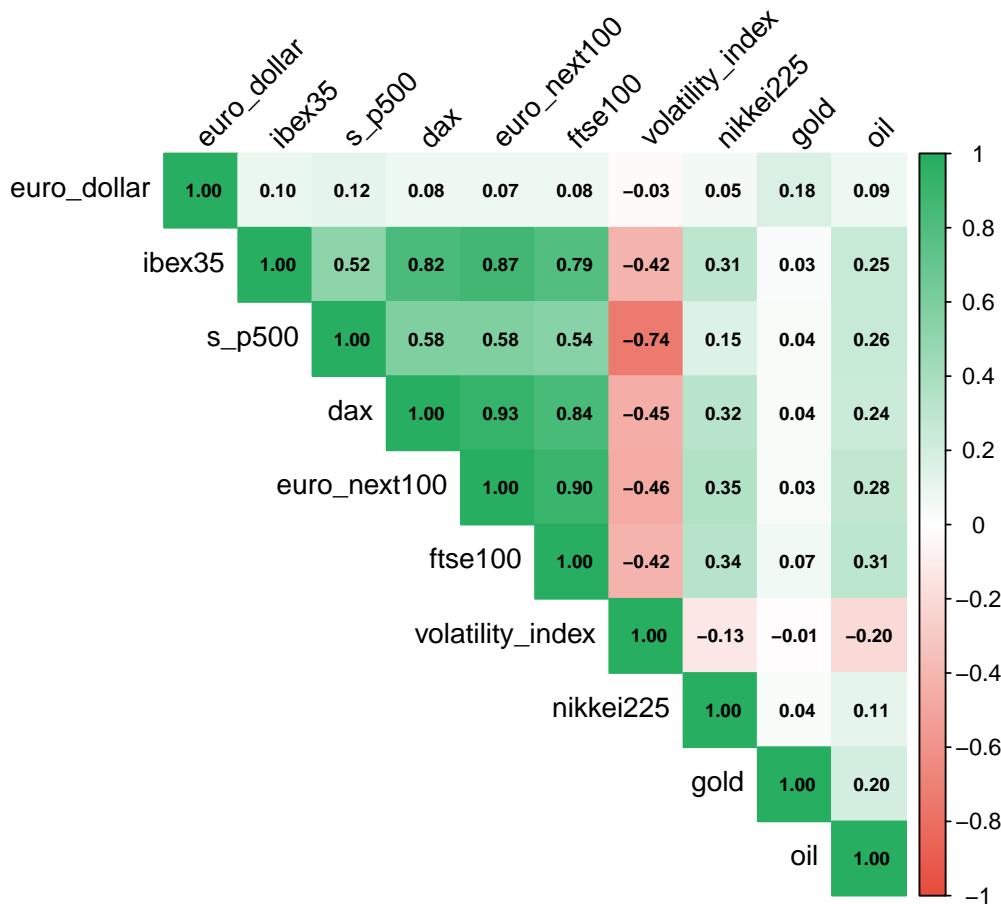


Figure 8: Correlación con otros índices internacionales

### 6.2.1.7.2 Causalidad de Granger: ¿Qué índices predicen al IBEX?

El test de causalidad de Granger (Clive W. J. Granger 1969) evalúa si los valores pasados de una variable mejoran la predicción de otra variable. En este análisis se aplicó con 5 retardos (lags) para determinar qué índices financieros y variables macroeconómicas tienen capacidad predictiva sobre el IBEX35. La hipótesis nula establece que los valores históricos de cada índice no aportan información adicional para predecir el IBEX35. Un p-valor inferior a 0.05 permite rechazar esta hipótesis, indicando que existe una relación temporal significativa.

El análisis identificó cuatro variables con causalidad Granger significativa sobre el IBEX35. El S&P 500 mostró la relación más fuerte ( $p = 2.13 \times 10^{-39}$ ,  $F = 38.98$ ), lo que indica que los movimientos del mercado estadounidense preceden y predicen fuertemente al mercado español. Este resultado es coherente con la integración de los mercados globales y el efecto de arrastre que ejerce Wall Street sobre las bolsas europeas. El índice de

volatilidad VIX también presentó causalidad fuerte ( $p = 3.8 \times 10^{-26}$ ,  $F = 26.14$ ), reflejando que los cambios en la percepción del riesgo de mercado anticipan movimientos del IBEX35.

Con efectos de menor magnitud pero estadísticamente significativos, el tipo de cambio Euro/Dólar ( $p = 9.93 \times 10^{-4}$ ,  $F = 4.11$ ) y el DAX alemán ( $p = 0.038$ ,  $F = 2.35$ ) también mostraron capacidad predictiva. El primero resulta relevante dado que las empresas del IBEX35 tienen importante exposición internacional, mientras que la relación con el DAX refleja la integración económica europea. Por el contrario, EuroNext 100, oro, petróleo, FTSE100 y Nikkei 225 no demostraron relación temporal significativa con el IBEX35 bajo este marco de análisis.

Es fundamental entender que causalidad de Granger no implica causalidad real en sentido estricto, sino únicamente que una variable contiene información útil para predecir otra en el tiempo(C. W. J. Granger 1980). Los resultados dependen del número de retardos seleccionados, en este caso 5lags (Mavrotas and Kelly 2008; Berben and Jansen 2005). Además, la ausencia de causalidad Granger no significa que las variables no estén correlacionadas con el IBEX35, simplemente que sus valores pasados no mejoran la predicción al nivel de lags especificado.

Desde una perspectiva práctica para el modelado predictivo del IBEX35, estos resultados sugieren priorizar la inclusión del S&P 500, VIX y Euro/Dólar como variables exógenas, considerar el DAX como complemento, y asignar menor prioridad a las variables sin causalidad temporal demostrada.

Table 14: Causalidad de Granger

Índice	F stat	p-value	Causalidad
s_p500	38.8212724	0.0000000	Sí
volatility_index	25.9688682	0.0000000	Sí
euro_dollar	4.0810513	0.0010650	Sí
dax	2.3760363	0.0366191	Sí
euro_next100	2.0295365	0.0713535	NO
gold	1.8554920	0.0987088	NO
oil	1.7103215	0.1285627	NO
ftse100	1.6317524	0.1479222	NO
nikkei225	0.3487216	0.8833326	NO

## 6.2.2 Análisis exploratorio de noticias mediante GDELT.

### 6.2.2.1 Scores de sentimiento diario

El script code/R/15.sentiment\_score.R procesa los seis campos del indicador TONE de GDELT para extraer métricas de sentimiento asociadas al IBEX35. El procedimiento se divide en cuatro fases: (1) extracción de estadísticas básicas de los campos TONE (tone score, positive, negative, polarity, activity y self-reference), (2) agregación temporal del sentimiento a nivel diario, mensual y anual, (3) identificación de las principales fuentes de noticias y (4) generación de visualizaciones que incluyen series temporales, distribuciones, correlaciones entre campos y análisis de volatilidad. La importancia de este análisis radica en establecer las características fundamentales del sentimiento de noticias sobre el mercado español, revelando patrones de cobertura mediática, la distribución entre noticias positivas y negativas, y la persistencia temporal de estos indicadores. Los resultados se materializan en archivos

Parquet con agregaciones temporales (diarias, mensuales y anuales) y siete gráficos que documentan la evolución histórica del sentimiento, facilitando la posterior integración con variables financieras en los modelos predictivos. En los análisis posteriores usaremos el archivo `daily_sentiment`, el cual agrega los resultados a nivel diario, misma escala temporal que los datos financieros. En la siguiente tabla puede verse el resumen estadístico del archivo de entrada, archivo filtrado y concatenado que contenía todas las noticias relacionadas con el Ibex35 y los filtros especificados en materiales y métodos.

Table 15: Estadísticas descriptivas del dataset GDELT

Métrica	Valor
Total registros	66377
Días únicos	4257
Total artículos	136745
Fecha mínima	2014-01-01
Fecha máxima	2025-10-16
Tone medio	-1.02
Tone SD	3.19
Tone mínimo	-25
Tone máximo	13.85
Positive medio	2.33
Positive SD	1.48
Negative medio	3.35
Negative SD	2.49
Polarity medio	5.68
Polarity SD	2.57
Activity medio	20.82
Activity SD	3.57
Self-reference medio	0.67
Self-reference SD	1.03

## 6 Campos TONE – Serie Temporal Diaria

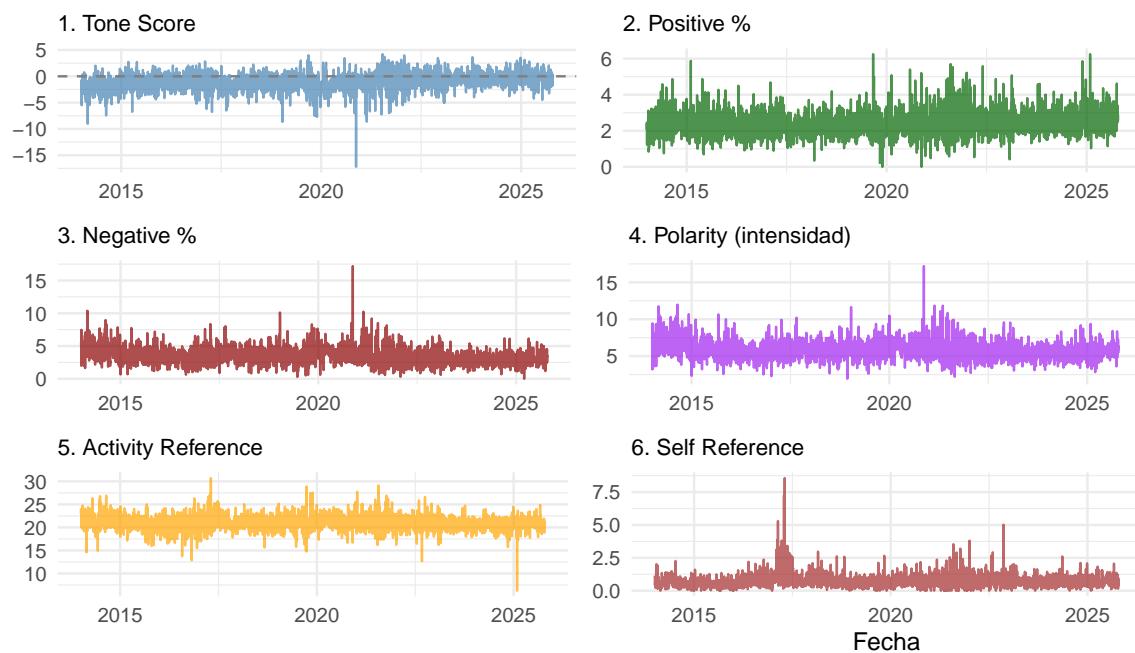


Figure 9: Score de sentimiento diario

## Distribución de 6 Campos TONE (muestra 100k registros)

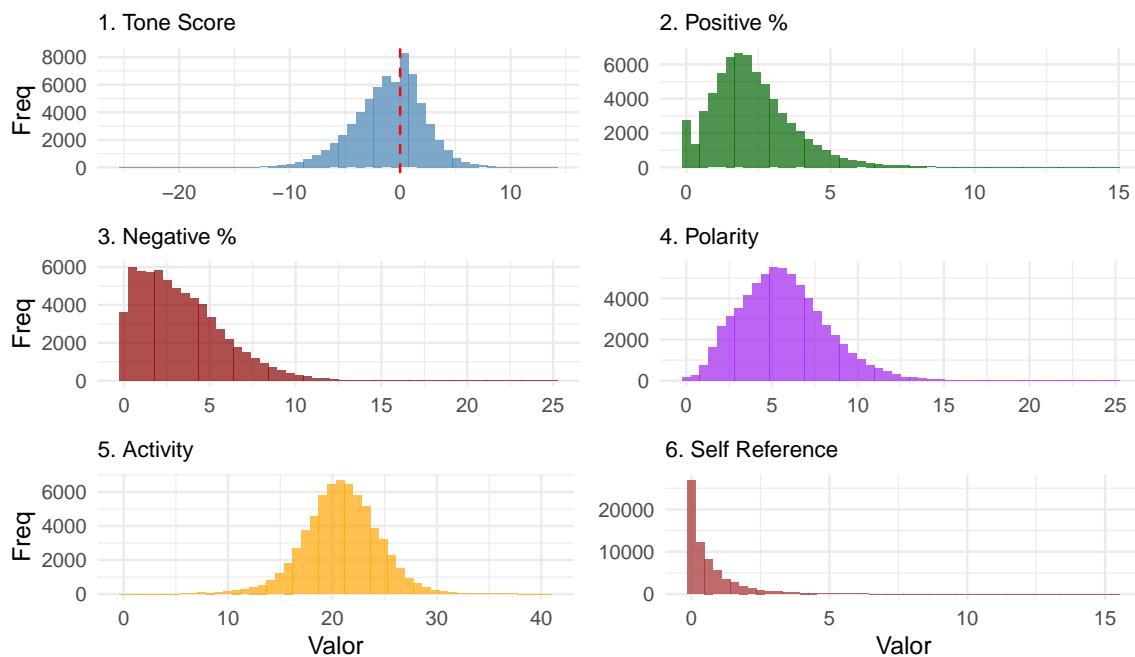


Figure 10: Distribución del Score de sentimiento diario

**Matriz de Correlación – Campos TONE**

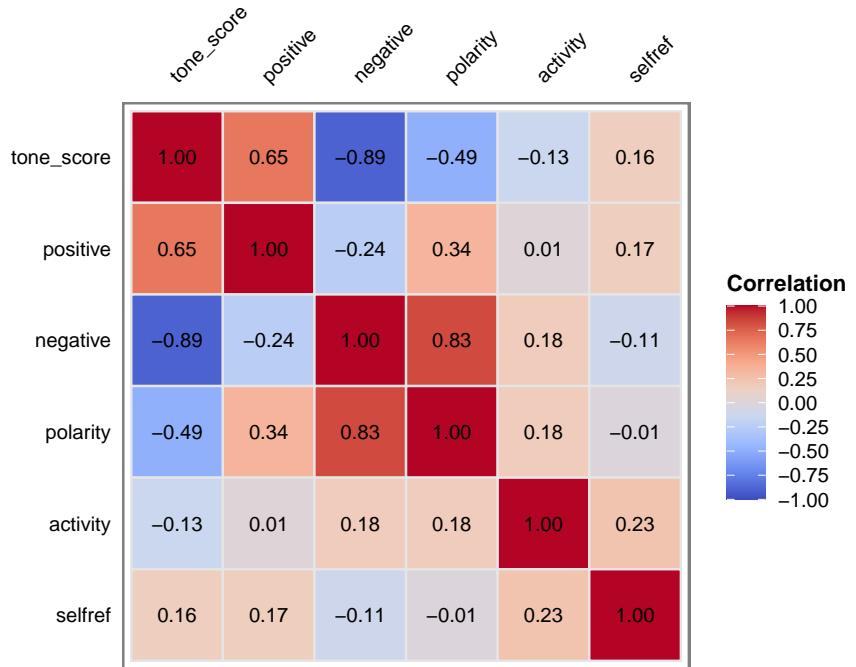


Figure 11: Heatmap de correlación entre campos de Tone

#### 6.2.2.2 Conteo de noticias positivas/negativas

El script 16.analysis\_sentiment\_counts\_ibex35.R cuantifica la distribución de noticias positivas y negativas sobre el IBEX35 mediante clasificación categórica del tone score. El procedimiento implementa (1) clasificación de registros en cuatro categorías de sentimiento (muy negativo, ligeramente negativo, ligeramente positivo, muy positivo), (2) cálculo de conteos globales y proporciones, (3) análisis de evolución temporal diaria y mensual del balance positivo/negativo, y (4) cálculo del ratio positivo/negativo como indicador de sesgo mediático. La relevancia de este análisis reside en cuantificar objetivamente el sesgo informativo del mercado, dado que la literatura sugiere que un exceso de noticias positivas o negativas puede influir en la percepción de los inversores y, consecuentemente, en la dinámica del mercado. Las salidas incluyen archivos Parquet con conteos temporales, un CSV con estadísticas globales, y cuatro visualizaciones que muestran la proporción de sentimientos, su evolución temporal y el ratio positivo/negativo a lo largo del período analizado. La siguiente tabla muestra el conteo global agrupado en 4 categorías:

Categoría	Sentimiento Simple	Número de registros	Número de Artículos	%registros	%artículos
Muy negativo	Negativo	22949	57496	34.57	42.05
Ligeramente negativo	Negativo	15664	32432	23.60	23.72
Ligeramente positivo	Positivo	17775	29882	26.78	21.85
Muy positivo	Positivo	9989	16935	15.05	12.38

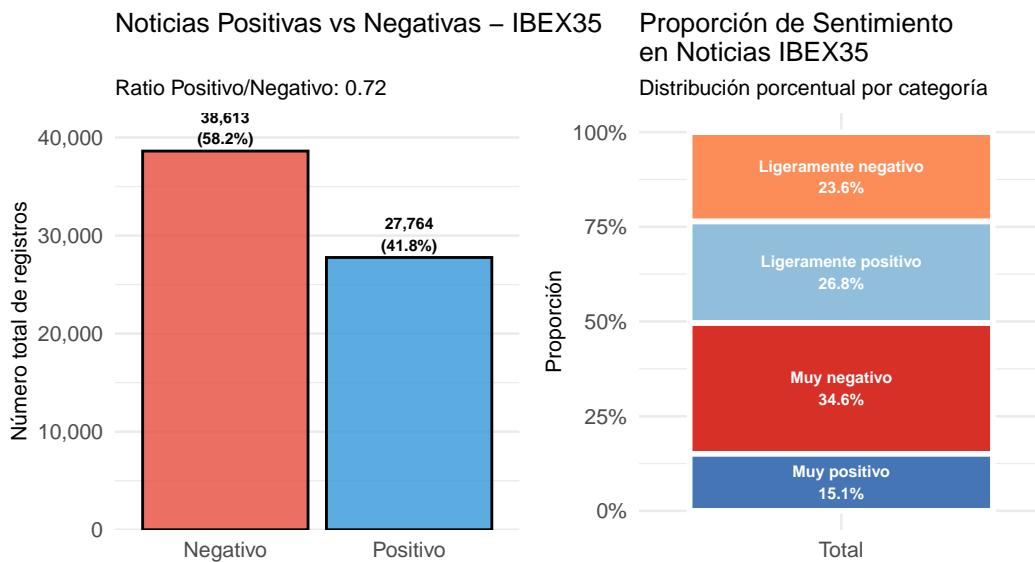


Figure 12: Porcentaje de noticias positivas y negativas

#### 6.2.2.3 Intensidad de cobertura mediática

Por último, el script `17.analysis_sentiment_intensity_ibex35.R` profundiza en la caracterización cualitativa del sentimiento mediante la categorización de tres dimensiones: valencia emocional (tone score), intensidad emocional (polarity) y volatilidad (desviación estándar del tone score). El procedimiento define umbrales cuantitativos para clasificar cada dimensión en tres niveles de intensidad, aplicando esta categorización a agregaciones diarias, mensuales y anuales del sentimiento. La importancia metodológica radica en transformar variables continuas de sentimiento en categorías interpretables que facilitan la identificación de períodos con consenso informativo versus períodos con noticias contradictorias, los cuales podrían asociarse con diferentes dinámicas de mercado. Los resultados se presentan mediante nueve gráficos de frecuencia (tres por cada escala temporal) que documentan la distribución de las categorías de sentimiento, permitiendo identificar si el mercado se caracteriza típicamente por sentimientos extremos o moderados, alta o baja carga emocional, y consenso o disenso mediático.

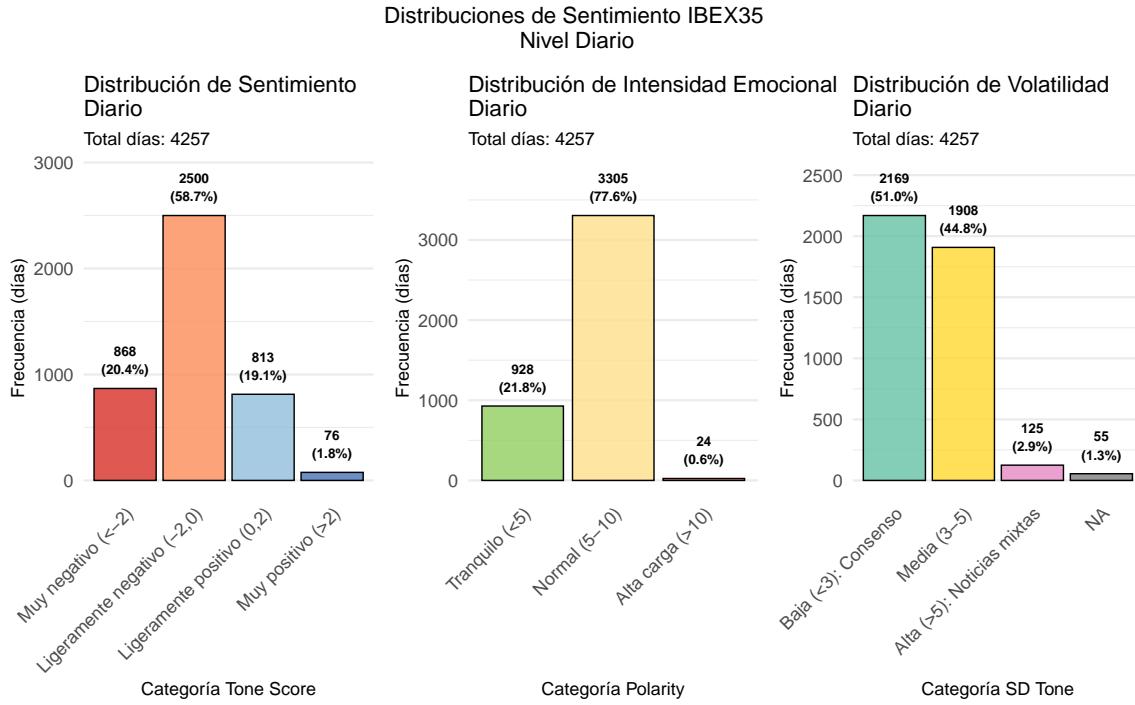


Figure 13: Intensidad de cobertura mediática diaria

### 6.3 Fase 3: Ingeniería de características.

La fase de feature engineering transforma las variables base descritas en la Fase 2 en un conjunto extenso de predictores técnicos, temporales, y de sentimiento. Esta sección describe la creación sistemática de features agrupados por familia, acompañado de análisis de sus características distribucionales y comportamiento temporal.

El pipeline de feature engineering se estructuró en módulos independientes para facilitar mantenimiento y extensibilidad:

- Módulo de features financieros técnicos (`code/R/04.financial_features.R`): Calcula indicadores técnicos derivados de OHLCV
- Módulo de features externos (`code/R/05.external_features.R`): Procesa variables de mercados internacionales y macroeconomía. Todos los módulos implementan split temporal estricto para evitar look-ahead bias. Los parámetros de transformación (medias, desviaciones, lambdas óptimos) se estiman exclusivamente en conjunto de entrenamiento y se aplican a test.

Los indicadores técnicos son transformaciones de las series de precios y volumen diseñadas para capturar momentum, tendencia, volatilidad, y estructura de mercado. Su uso está ampliamente documentado en finanzas cuantitativas Pring (2002) y estudios de machine learning aplicado a mercados Fischer and Krauss (2018).

La descripción de todas las variables técnicas generadas se encuentran descritas en el Anexo 1. “Documentación de Variables Financieras e Índices Externos”

Antes de normalizar y aplicar transformaciones, vamos a separar los datos en train y test. El set completo de datos usados es desde el 01-01-2004 hasta el 15-10-2025. Usaremos

como train los datos hasta el 15-10-2024, y como test los datos del día 16-10-2024 hasta el 16-10-2025. Debido a que los datos usados para el análisis de sentimientos abarcarán sólo desde el 2014-01-01, para facilitar el análisis, los datasets finales de train y test se filtrarán posteriormente para que empiecen desde esa fecha. A su vez, separaremos sólo las variables originales (“date”, “open”, “high”, “low”, “close”, “volume”). El resto serán calculadas mediante una función específica.

### 6.3.1 Creación de variables financieras

```
# Seleccionar datos Ibex
ibex <- all_stocks_df |>
  dplyr::filter(index == "ibex35") |>
  dplyr::select(date, open, high, low, close, volume) |>
  dplyr::arrange(date)

# Ejecutar el script con la función
source("code/R/04.financial_features.R")
train_start <- "2004-01-01"
train_end <- "2024-10-15"
test_start <- "2024-10-16"
test_end <- "2025-10-16"
result <- calculate_financial_features_split(
  df = ibex, # Dataset completo (train + test juntos)
  train_start <- train_start,
  train_end <- train_end,
  test_start <- test_start,
  test_end <- test_end,
  apply_scaling = FALSE,
  apply_yj = FALSE)
```

```
==== FEATURE ENGINEERING FINANCIERO SIN LOOK-AHEAD BIAS ====
Train: 2004-01-01 a 2024-10-15
Test: 2024-10-16 a 2025-10-16
Obs train: 5306
Obs test: 256
--- Procesando TRAIN ---
--- Procesando TEST ---
RESUMEN - FEATURES FINANCIERAS SIN LOOK-AHEAD BIAS
Features creadas: 45
Train obs: 5306
Test obs: 256
Stats guardados: 2
Columnas con NAs train: 39 de 45
Columnas con NAs test: 39 de 45
- Sin look-ahead bias
- Estadísticos de train aplicados a test
== Dataset: Ibex35 ==
```

```

== Train dataset ==
== Starting date: 2004-01-01 == End date: 2024-10-15==
== Test dataset ==
== Starting date: 2024-10-16 == End date: 2025-10-16==
==== DIMENSIONES FINALES ====
Train:5306obs x51vars
Test:256obs x51vars

```

### 6.3.1.1 Incorporación de variables de otros índices

El cálculo y justificación de estas variables se encuentra en el Anexo 1. La programación de la función se encuentra en el anexo Código/R/05.external\_features.R

```

source("code/R/05.external_features.R")
# 2. Llamar a la función
result <- calculate_external_features_split(
  all_stocks_df = all_stocks_df,
  indices = get_recommended_indices("extended"),
  target_index = "ibex35",
  train_start = train_start,
  train_end = train_end,
  test_start = test_start,
  test_end = test_end,
  features_config = get_default_config(),
  missing_method = "forward_fill",
  lag_international = TRUE
)

```

### 6.3.1.2 Incorporación de Euribor

Para ello se hace uso de la librería fedr de la Reserva Federal de EEUU. Mediante API, obtenida previo registro, es posible acceder a diferentes índices. Para el Euribor 3M se hace uso de la clave IR3TIB01EZM156N. Dado que es un dato que se publica cada mes y que se mantiene estable durante ese periodo, se completan los NA el el valor válido justamente anterior.

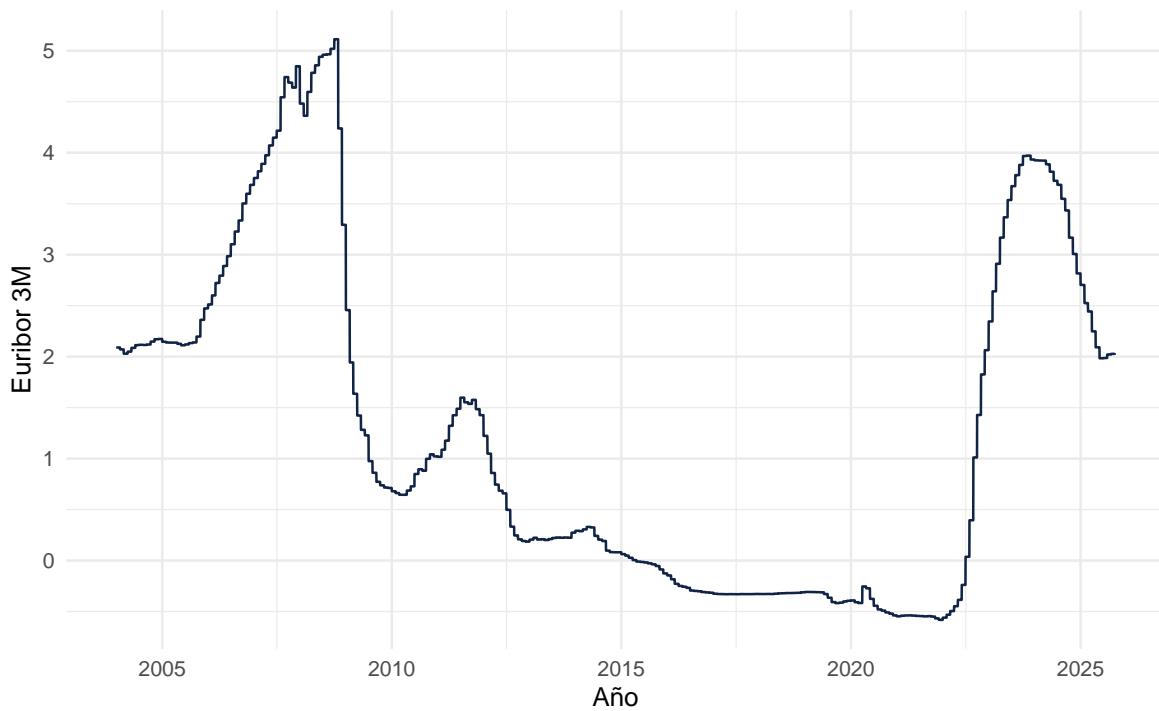


Figure 14: Evolución Euribor3M

### 6.3.2 Comprobación de variables presentes

Table 17: Resumen de datasets

Dataset	Numero.de.Variables
Train	97
Test	97

Table 18: Variables presentes

Variables (1-33)	Variables (34-66)	Variables (67-97)
date	open	high
low	close	volume
weekday	month	quarter
close_smooth	log_return_pct	price_momentum
volatility_20	returns_velocity	volatility_velocity
volume_velocity	returns_acceleration	volatility_acceleration
returns_next	returns_next_5	returns_next_10
returns_next_20	log_close	sma_20
ema_12	dist_sma20	rsi_14
roc_5	obv	volume_roc

volume_sma20	volume_ratio	returns_lag1
returns_lag5	returns_lag10	returns_lag20
volatility_lag1	volatility_lag5	returns_mean_5
returns_mean_20	volatility_5	volatility_10
range_5	range_20	high_max_20
low_min_20	price_position_20	fastK
fastD	slowD	obv_norm
sp500_return	sp500_return_lag1	sp500_momentum
sp500_vol20	dax_return	dax_return_lag1
dax_momentum	dax_vol20	ftse100_return
ftse100_return_lag1	ftse100_momentum	ftse100_vol20
euronext100_return	euronext100_return_lag1	euronext100_momentum
euronext100_vol20	spread_target_sp500	spread_target_dax
spread_eu_us	vix_close	vix_change
vix_return	vix_lag1	vix_zscore
vix_regime	oil_return	oil_return_lag1
oil_momentum	oil_vol20	gold_return
gold_return_lag1	gold_momentum	gold_vol20
oil_gold_ratio	oil_gold_ratio_change	eurodollar_return
eurodollar_level	eurodollar_ma20	eurodollar_deviation
eurodollar_momentum	eurodollar_return_lag1	corr_target_sp500_60
corr_target_dax_60	risk_on_score	vol_spread_target_sp500
euribor_3m		

### 6.3.3 Limpieza de Variables

Se eliminan las variables close, close\_smooth, open, low y high por estar claramente relacionadas con log\_return. Se eliminan también aquellas variables con >10% de NA. En este caso, ninguna variable presentó más del 10% de NA, por lo que sólo se eliminaron las 5 variables indicadas anteriormente. A su vez, se realiza la imputación de NA en el resto de variables mediante Forward fill. Los NA presentes son fundamentalmente provocados durante el cálculo. Por ejemplo, en el caso de SMA20 tendrán los primeros 20 valores como NA.

#### 6.3.3.1 Imputación de NAs restantes. Caso de cotización del petróleo COVID'19

La crisis del COVID'19 afectó a la cotización del petróleo, llegando a cotizar en negativo durante algunos días. Este precio negativo provoca que se obtengan valores NA al calcular logaritmos, especialmente en las variables "oil\_return", "oil\_return\_lag1", "oil\_vol20", "oil\_momentum", "risk\_on\_score". Por ello se recalculará oil\_return mediante interpolación lineal, suponiendo un precio estable y al mínimo durante esos días de cotización negativa. Seguidamente se recalculará el resto de variables. Se hizo uso del script code/R/06.cleaning\_features.R

#### 6.3.3.2 Limpieza por correlación y varianza cercana a cero

No se detectaron variables con varianza cercana a cero. En cuanto a las variables altamente correlacionadas, se establece un umbral de correlación al 85%. La función `caret::findCorrelation()` busca los pares con una correlación superior al 85% y elimina aquella variable que presente mayor correlación media con el resto de variables, manteniendo la variable más “independiente”. Se hizo uso del script `code/R/06.cleaning_features.R`, con la función `reduce_by_correlation_variance()`.

Esta reducción elimina 30 variables numéricas, una reducción del 32.6%, pasando el dataset de 92 variables totales a 62.

Las variables eliminadas son: "vix\_close", "dist\_sma20", "vix\_lag1", "rsi\_14", "euronext100\_momentum", "volatility\_10", "returns\_mean\_20", "euronext100\_vol20", "volatility\_20", "volatility\_lag1", "ftse100\_vol20", "dax\_vol20", "price\_position\_20", "fastK", "fastD", "log\_close", "low\_min\_20", "ema\_12", "sma\_20", "log\_return\_pct", "returns\_velocity", "spread\_target\_sp500", "euronext100\_return", "returns\_lag1", "euronext100\_return\_lag1", "eurodollar\_ma20", "vix\_change", "volume", "obv", "eurodollar\_deviation"

### 6.3.4 Random Forest para ranking de variables

Para llevar a cabo una selección de variables se hace uso de `RandomForest`, estableciendo un umbral del 90% de explicación de la varianza para la selección de variables. Se hace uso de la función personalizada `rank_features_by_importance()`, accesible en el script `code/R/06.cleaning_features.R`. Para el arbol `randomForest` se hace uso de la siguiente programación:

```
rf_model <- randomForest(
  returns_next ~ .,
  data = train_rf,
  ntree = 500,
  importance = TRUE,
  mtry = sqrt(n_vars), #default
  nodesize = 5,
  keep.forest = TRUE
)

source("../code/R/06.cleaning_features.R")

# Ejecutar RF con plots en TIFF y PDF
rf_results <- rank_features_by_importance(
  train_df = train_for_rf,
  test_df = test_for_rf,
  target_var = "returns_next",
  ntree = 500,
  importance_threshold = 90,
  save_plots = TRUE,
  save_files = TRUE,
  output_dir = output_path,
  plot_formats = c("tiff", "pdf"), # Guardar en ambos formatos
```

```

plot_width = 20,
plot_height = 16,
verbose = TRUE,
targets = c("returns_next", "returns_next_5", "returns_next_10",
           "returns_next_20")
)

```

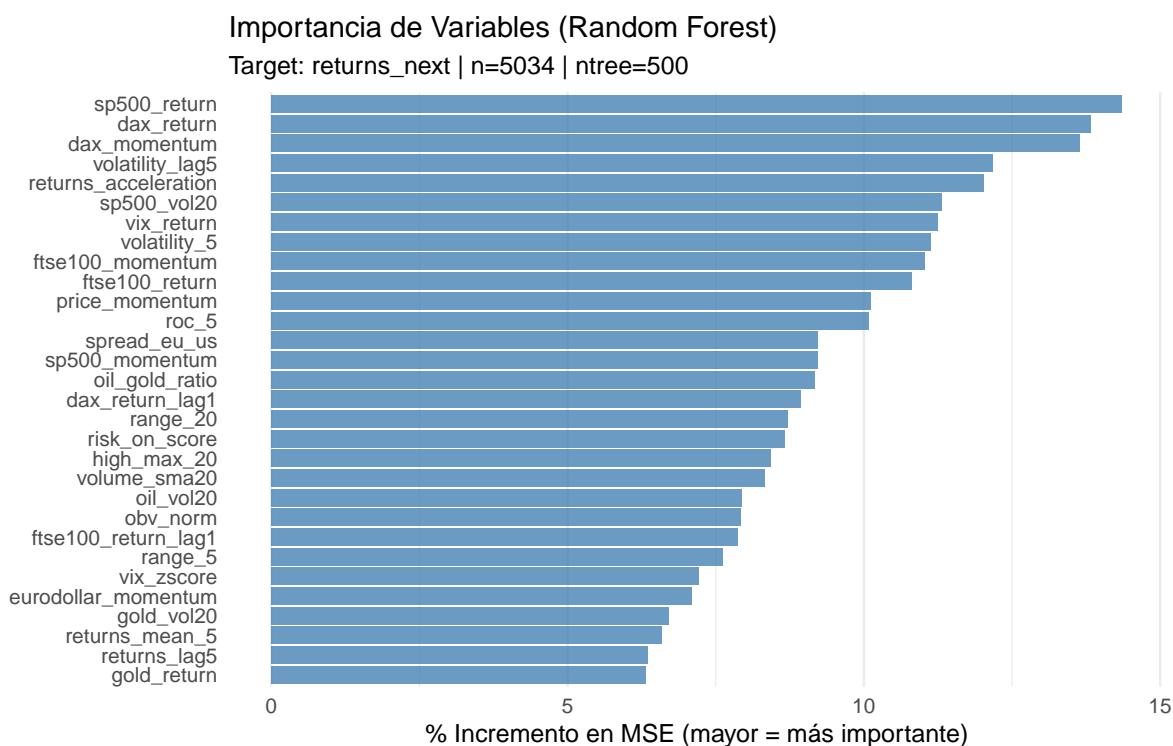


Figure 15: Importancia de variables for RF

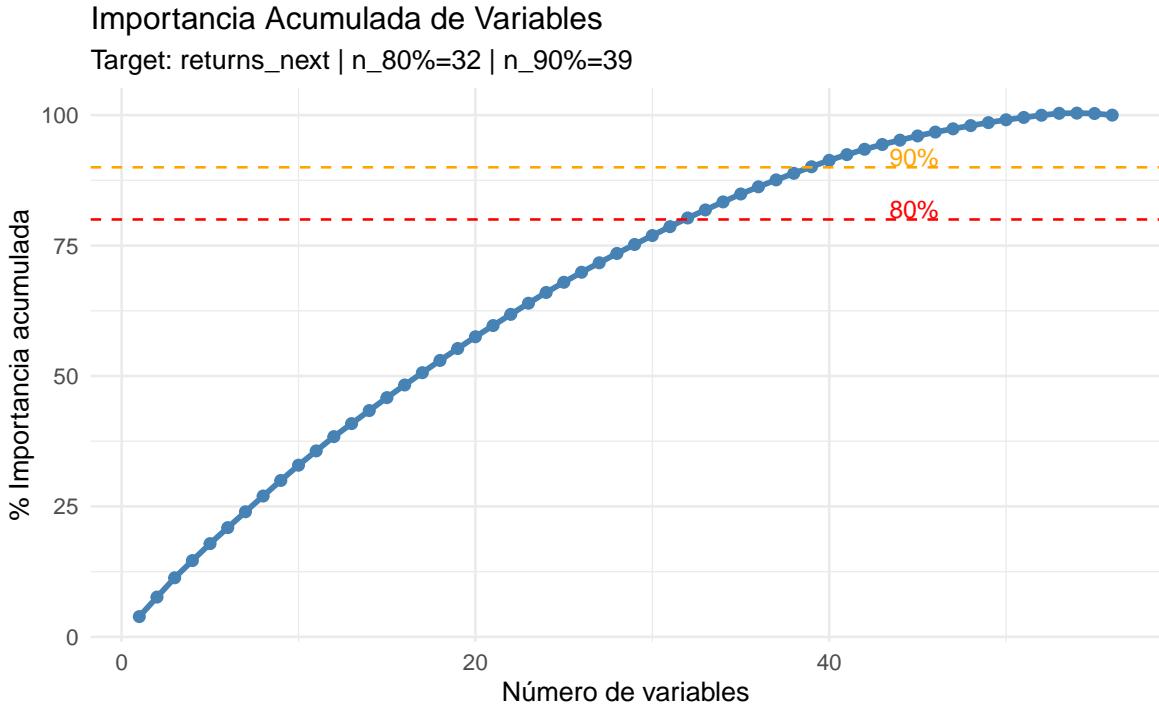


Figure 16: Importancia acumulada por RF

Tras la eliminación de las siguientes variables por RandomForest: "volume\_velocity", "volatility\_acceleration" "volume\_roc", "volume\_ratio", "returns\_lag10", "returns\_lag20", "sp500\_return\_lag1", "oil\_return\_lag1", "gold\_return\_lag1", "gold\_momentum", "oil\_gold\_ratio\_change", "eurodollar\_return", "eurodollar\_return\_lag1", "corr\_target\_sp500\_60", "corr\_target\_dax\_60", "weekday", "month", "quarter"; nos quedamos con 44 variables totales.

Etapa	Número Variable	N cambio Variables	% cambio
Inicial	6	NA	NA
Variables técnicas	51	45	750
Variables exógena	97	46	90.2
Limpieza	92	-5	-5.15
Correlación / Varianza	62	-30	-32.61
RF 90%	44	-18	-29.03

### 6.3.5 Limpieza del análisis de sentimientos

Partimos del dataset obtenido en la Fase 2: Scores de sentimiento diario daily\_sentiment.

Procederemos a su limpieza previa a su integración con los datos financieros. Para la limpieza de NAs por días de ausencia de publicaciones, asumiremos que el sentimiento permanece, por lo que los datos se imputaran mediante fill-forward. En caso que quedasen datos ausentes al inicio del periodo éstos se imputarán con la mediana.

#### 6.3.5.1 Limpieza por correlación y varianza cercana a cero

Se procede a una limpieza de variables por correlación y por varianza cercana a cero. Se elimina la variable `mean_tone_score` por alta correlación con `median_tone_score`, prevaleciendo esta última.

### Matriz de Correlación (cutoff = 0.90)

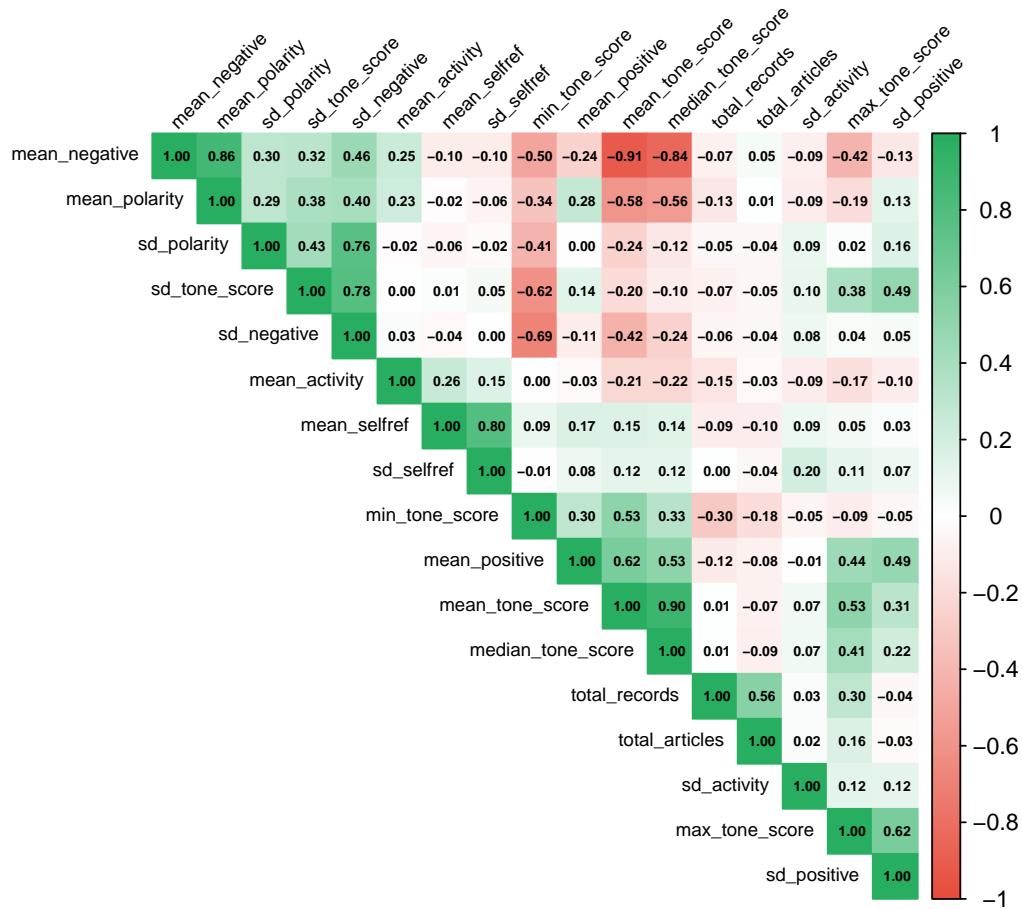


Figure 17: Matriz de correlaciones de variables sentiment

## 6.4 Fase 4: Normalización y escalado de datos.

### 6.4.1 Unión de datos financieros con datos de sentimientos

Dado que los datos del análisis de sentimientos únicamente alcanzan hasta 2014, elaboraremos tres datasets diferentes, (1) Datos financieros de 2004 a 2024, (2) Datos financieros de 2014 a 2024, y (3) Datos financieros con datos de sentimientos de 2014 a 2024. Con el set 2 elaboraremos una línea base y con ello comprobaremos si los datos de sentimientos de GDELT tienen algún aporte al modelo predictivo y si bien el aumento de la serie histórica

mejora el modelo predictivo.

Tras la unión, comprobamos que no existan NA. Encontramos 12 días en los que no hubo noticias (o el archivo GDELT no estaba disponible). Procedemos a imputar los NAs considerando que el sentimiento persiste igual al día anterior (forward fill).

Antes de la normalización de los datos, comprobamos mediante randomForest si podemos eliminar alguna variable. La ejecución de RandomForest se llevó a cabo con los mismos parámetros que en el punto anterior. Esta vez RF nos indica que bastan 39 variables para explicar el 90% de la varianza, exactamente igual al punto anterior. Entre las primeras 30 variables no encontramos ninguna de sentimientos, lo que ya nos indica que el análisis de sentimiento desde GDELT posiblemente aporten poco al modelo predictivo.

Por último antes de normalizar, crearemos las variables target categóricas, `direction_next` con ventanas temporales 1,5,10 y 20. Estas nuevas variables target serán binarias, siendo 0 si el signo de `returns_next` es negativo y siendo 1 si el signo de `returns next` es positivo.

#### 6.4.2 Normalización de datos

La normalización de los datos se llevó a cabo mediante Z-score. La normalización se calcula exclusivamente en el set de entrenamiento, de donde se extraen los parámetros estadísticos y éstos se aplican al set de test. La función puede encontrarse en el script `code/R/20.feature_scaling.R` y a continuación se detalla su uso:

```
source("../code/R/20.feature_scaling.R")
# Para datasets SOLO financieros
scaled_financial <- create_scaled_versions(
  train_df = train_final,
  test_df = test_final,
  scale_method = "center_scale", # Z-score estándar
  save_scaler = TRUE,
  output_dir = "output/RData/ML_financial")
# Para datasets CON sentimiento
scaled_sentiment <- create_scaled_versions(
  train_df = train_final_with_sentiment,
  test_df = test_final_with_sentiment,
  scale_method = "center_scale",
  save_scaler = TRUE,
  output_dir = "output/RData/ML_sentiment")
```

### 6.5 Fase 5: Predicción de valores futuros.

Mediante diferentes algoritmos trataremos de predecir valores futuros. Para realizar esta predicción usaremos modelos clásicos como Arima y Prophet y modelos avanzados de *Machine Learning*.

Mientras que con los modelos clásicos trataremos de predecir los retornos en el periodo de test, con los modelos avanzados podemos establecer diferentes variables objetivo. En este caso, trataremos de dar solución a dos problemas diferentes, uno de regresión y otro de clasificación. Mediante la regresión trataremos de predecir los retornos futuros a 1 día,

5 días (una semana), 10 días (2 semanas) y 20 días (aprox. un mes). Mediante la clasificación, trataremos de predecir el signo de los retornos, es decir la evolución del mercado, prediciendo si el mercado baja (0) o sube (1). Modelos Clásicos: Arima y Prophet

Los modelos clásicos para series temporales son modelos univariados. Por ello, usaremos series temporales con sólo una variable (precio o retorno según el caso). Para los modelos 'ARIMA' y 'ARIMAX', y PROPHET usaremos el retorno, ya que es una transformación estacionaria y sin tendencia (valores distribuidos alrededor del 0). Además, al ser una log-transformada, su distribución se asemeja a la normal. En el caso del modelo ARIMAX, usaremos los retornos de alguno índices internacionales. Toda la programación detallada relacionada a los modelos clásicos de la fase 5 puede encontrarse en el documento Fase5.qmd.

### 6.5.1 Modelos Clásicos

#### 6.5.1.1 Arima

La metodología ARIMA, trata de realizar previsiones acerca de los valores futuros de una variable, utilizando únicamente como información la contenida en los valores pasados de la propia serie temporal. Como requisitos, necesitamos que la serie sea estacionaria.

Comprobamos la estacionalidad de la serie mediante el test "Augmented Dickey-Fuller" y comprobamos que en el caso de la serie original (close), el p-value es mayor que 0.05, por lo que se acepta la hipótesis nula, la serie no es estacionaria. En el caso de la serie diferenciada (log\_return) si que presenta estacionariedad, con p-value=0.01. Para la identificación del modelo arima podemos usar dos funciones `arima`, donde tenemos que indicar los coeficientes (p,d,q) y la estacionalidad; o `auto.arima`, dónde ésta ajusta automáticamente dichos coeficientes para obtener el mejor ajuste (mejor AIC o BIC). Usando la función `forecast::autoarima`, encontramos que el modelo óptimo es el ARIMA(0,0,0) e incluso forzando la inclusión de la media continua con ARIMA(0,0,0). ¿Por qué ARIMA(0,0,0)? El modelo está diciendo: "Los returns del IBEX son ruido blanco puro, no hay patrón autocorrelativo". Forzaremos manualmente el modelo para que incluya las medias para ver si esto mejora el modelo. Obtenemos en este caso una precisión de la dirección del 59.84%. Valor sospecho que trataremos más adelante.

#### 6.5.1.2 Arimax

El modelo ARIMAX, `auto.arima` con variables exógenas, permite la incorporación de variables exógenas para aumentar la precisión de la predicción. Para el modelo ARIMAX usaremos como variables exógenas los siguientes retornos: - S&P500, DAX, VIX, OIL y EuroDollar. Se separará el dataset en train y test y se calcularon los retornos de las variables (ibex\_return, sp500\_return, dax\_return, eurodollar\_return, oil\_return y vix\_return). Acto seguido se calcularon los lag-1 de las variables exógenas. Se construyó una matrix con los 5 índices exógenos. Dicha matriz se incorpora al modelo arimax mediante el parámetro `xreg` de `auto.arima`.

#### 6.5.1.3 Prophet

Prophet es un algoritmo desarrollado por Facebook y consiste en un modelo de regresión aditiva con una tendencia de curva de crecimiento lineal o logística por partes. Incluye un componente estacional anual modelado usando series de Fourier y un componente estacional semanal modelado usando variables ficticias.

Usamos los hiperparámetros por defecto, únicamente indicando que no existe periodicidad anual, ni semanal ni diaria.

#### 6.5.1.4 Comparación de modelos clásicos

Table 20: Resultados de modelos de series temporales

Modelo	RMSE	MAE	MAPE	Direction Accuracy	Samples
ARIMAX	0.980	0.722	125.301	53.247	231
Prophet	1.036	0.721	100.879	59.843	254
ARIMA mean	1.039	0.728	98.921	59.843	254
ARIMA	1.040	0.730	100.000	0.000	254

Resulta sospechoso que los modelos clásicos Arima con medias y Prophet tenga una precisión tan buena en el contexto de las series temporales financieras. Observando con detalle las predicciones se observan que estos dos modelos con mejor precisión siempre pronostican subida. En el periodo de test (2024-10-16 a 2024-10-15), el 59.84% de los días son de subida. Un modelo que siempre prediga subida en ese periodo obtendría una precisión de 59.84%, con lo que podemos concluir que los modelos clásicos no ofrecen una predicción correcta.

#### 6.5.2 Modelos de Machine Learning.

En la siguiente tabla se describen los modelos usados según el problema (clasificación / regresión). A su vez, disponemos de tres datasets diferentes: Datos financieros desde 2004, datos financieros desde 2014 y datos financieros con datos de sentimientos desde 2014. Según el modelo a usar faremos uso de los datos escalados o no-escalados. Finalmente disponemos de 4 variables objetivo por problema. Para regresión tenemos los valores de retornos futuros a 1,5,10 y 20 días (`returns_next`, `returns_next_5`, `returns_next_10` y `returns_next_20`), que en el problema de clasificación se convierten en `direction_next`, `direction_next_5`, `direction_next_10` y `direction_next_20`.

Table 21: Descripción de datasets utilizados

Dataset	Tipo	Descripción
financial_scaled	Scaled	Variables financieras normalizadas
financial_unscaled	Unscaled	Variables financieras sin normalizar
financial_long_scaled	Scaled	Serie larga financiera normalizada
financial_long_unscaled	Unscaled	Serie larga financiera sin normalizar
sentiment_scaled	Scaled	Variables financieras + sentimiento normalizadas

sentiment_unscaled	Unscaled	Variables financieras + sentimiento sin normalizar
--------------------	----------	----------------------------------------------------

Table 22: Configuración experimental completa

Problema	Datos	Datasets	Targets	Modelos	Total
Regresión	Scaled	3	4	9	108
Regresión	Unscaled	3	4	6	72
Clasificación	Scaled	3	4	8	96
Clasificación	Unscaled	3	4	6	72
<b>TOTAL</b>		<b>NA</b>	<b>NA</b>	<b>NA</b>	<b>348</b>

### 6.5.2.1 Comparación de modelos

#### 6.5.2.1.1 Regresión

Tomando todos los modelos de regresión, obtenemos aquellos con mayor precisión, independientemente del target y los representamos en un mapa de calor en función del dataset usado. Parece ser que los modelos LSTM junto con los GRU son los que mejor se comportan de media.

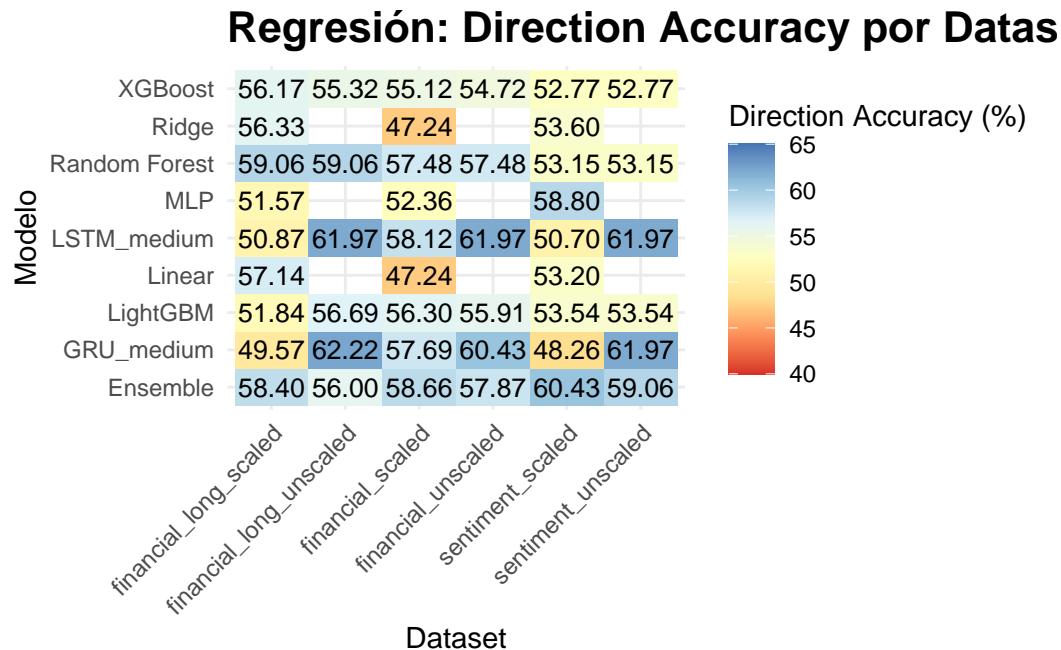


Figure 18: Heatmap de Direction Accuracy por dataset y modelo

Table 23: Precisión máxima por modelo

Dataset	Modelo	Balanced Accuracy
financial_long_unscaled	GRU_medium	62.22
financial_long_unscaled	LSTM_medium	61.97
financial_unscaled	LSTM_medium	61.97
sentiment_unscaled	GRU_medium	61.97
sentiment_unscaled	LSTM_medium	61.97
financial_unscaled	GRU_medium	60.43
sentiment_scaled	Ensemble	60.43
financial_long_scaled	Random Forest	59.06
financial_long_unscaled	Random Forest	59.06
sentiment_unscaled	Ensemble	59.06

Heatmap de Direction Accuracy por dataset y modelo

- Top10 modelos por RMSE

Cuando comparamos los modelos por RMSE, vemos que los modelos LSTM y GRU no aparecen en el top10, aparece RandomForest.

Table 24: Mejor modelo por RMSE

Dataset	Target	Modelo	RMSE	R2	Balanced Accuracy
financial_long_scaled	returns_next	Random Forest	0.97	0.13	59.06
financial_long_unscaled	returns_next	Random Forest	0.97	0.13	59.06
financial_long_scaled	returns_next	Ensemble	0.98	0.10	55.51
sentiment_unscaled	returns_next	Random Forest	0.99	0.09	53.15
sentiment_scaled	returns_next	Random Forest	0.99	0.09	53.15
financial_long_unscaled	returns_next	Ensemble	0.99	0.08	55.91
financial_scaled	returns_next	Random Forest	0.99	0.08	57.48
financial_unscaled	returns_next	Random Forest	0.99	0.08	57.48
financial_long_scaled	returns_next	Ridge	1.00	0.07	50.79
financial_long_scaled	returns_next	Linear	1.00	0.07	50.79

- Mayor exactitud por modelo

De los 9 modelos usados en regresión, extraemos en qué condiciones (dataset-target) obtienen mejor precisión. En caso de empate, escogemos el que tenga menos RMSE. Observamos claramente ahora, que el modelo GRU a pesar de tener una de las mayores precisiones, también tiene un RMSE extremadamente alto.

Table 25: Mayor exactitud en cada modelo

Dataset	Target	Modelo	RMSE	R2	Balancead Accuracy
sentiment_scaled	returns_next_20	Ensemble	1.04	-0.02	60.43
financial_long_unscaled	returns_next_10	GRU_medium	660.02	-395133.68	62.22
financial_long_unscaled	returns_next	LightGBM	1.02	0.02	56.69
financial_long_scaled	returns_next_10	Linear	1.06	-0.02	57.14
financial_unscaled	returns_next	LSTM_medium	1.04	-0.01	61.97
sentiment_scaled	returns_next_5	MLP	1.07	-0.05	58.80
financial_long_scaled	returns_next	Random Forest	0.97	0.13	59.06
financial_long_scaled	returns_next_10	Ridge	1.06	-0.02	56.33
financial_long_scaled	returns_next_20	XGBoost	1.15	-0.23	56.17

Mayor exactitud por modelo

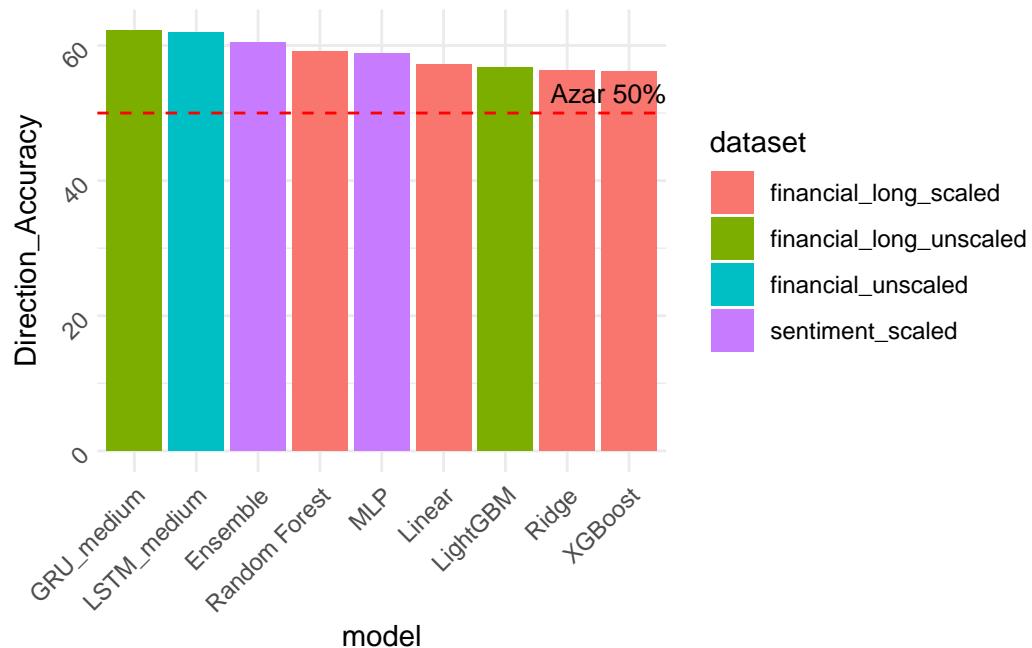


Figure 19: Mayor exactitud por modelo

- Mejor modelo por target

Obtenemos el mejor modelo por target, siendo los modelos basados en árboles los que mejor se comportan, RandomForest y el ensemble de modelos de árboles (RandomForest, XGBoost y LightGBM)

Table 26: Mejor modelo por target

Dataset	Target	Modelo	RMSE	R2	Balancead Accuracy
financial_long_scaled	returns_next	Random Forest	0.97	0.13	59.06

sentiment_unscaled	returns_next_10	LSTM_medium	1.05	-0.01	61.78
sentiment_unscaled	returns_next_20	Ensemble	1.04	-0.01	49.36
financial_long_scaled	returns_next_5	Ensemble	1.04	-0.01	58.40

### 6.5.2.1.2 Clasificación

En el caso de clasificación, obtenemos varios modelos de diversas arquitecturas con precisiones por encima del 60%. Analicemoslas con más detalle.

Table 27: Precisión máxima por modelo

Dataset	Modelo	Accuracy
financial_long_scaled	Random Forest	0.63
financial_long_unscaled	Ensemble	0.63
financial_long_unscaled	Random Forest	0.63
financial_long_unscaled	GRU_medium	0.62
financial_long_unscaled	LightGBM	0.62
financial_long_unscaled	LSTM_medium	0.62
financial_scaled	GRU_medium	0.62
financial_unscaled	GRU_medium	0.62
financial_unscaled	LSTM_medium	0.62
sentiment_scaled	GRU_medium	0.62

Heatmap de Accuracy por dataset y modelo

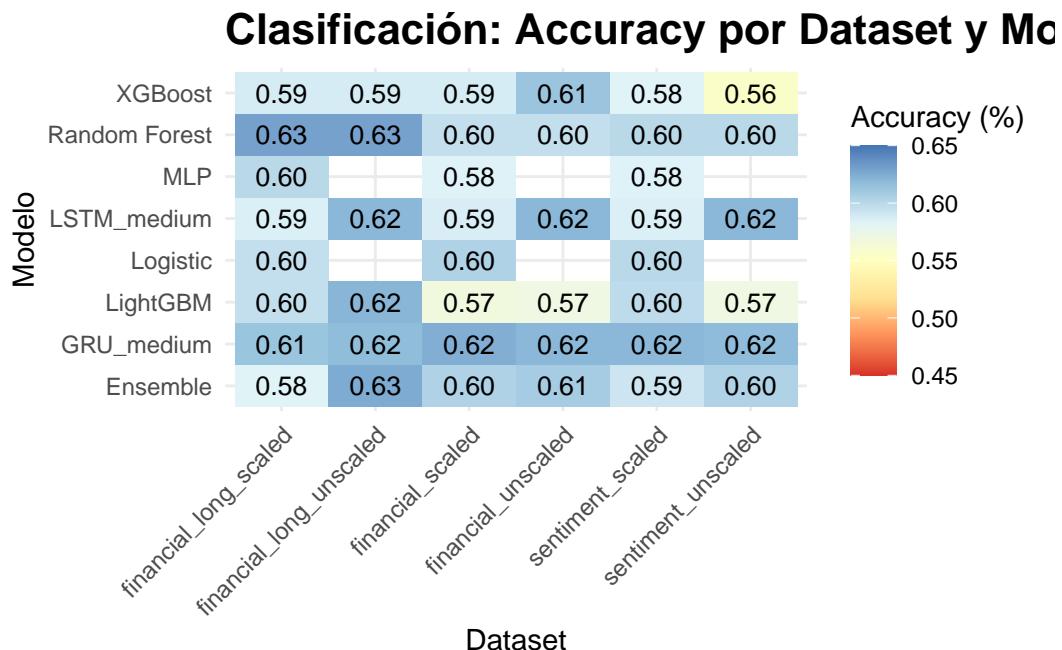


Figure 20: Heatmap de Accuracy por dataset y modelo

- Top10 modelos por F1-score

Parece ser que los modelos de clasificación basados en redes neuronales como GRU y LSTM tienen un desempeño bastante bueno.

Table 28: Mejor modelo por F1-score

Dataset	Target	Modelo	F1	Accuracy	ROC AUC
financial_scaled	direction_next	GRU_medium	0.77	0.62	0.54
financial_unscaled	direction_next	GRU_medium	0.77	0.62	0.50
financial_unscaled	direction_next	LSTM_medium	0.77	0.62	0.49
financial_long_unscaled	direction_next	LSTM_medium	0.77	0.62	0.51
sentiment_unscaled	direction_next	LSTM_medium	0.77	0.62	0.51
financial_unscaled	direction_next_20	LSTM_medium	0.76	0.62	0.50
financial_long_unscaled	direction_next_20	LSTM_medium	0.76	0.62	0.50
sentiment_unscaled	direction_next_20	LSTM_medium	0.76	0.62	0.50
sentiment_scaled	direction_next	GRU_medium	0.76	0.62	0.43
financial_unscaled	direction_next_10	GRU_medium	0.76	0.62	0.48

- Mayor exactitud por modelo

Para clasificación, el modelo con mayor precisión y mayor F1-scores es Ensemble, para el dataset “financial\_long\_unscaled” y el target “direction\_next\_20”

Table 29: Mayor exactitud en cada modelo

Dataset	Target	Modelo	F1	Accuracy
financial_long_unscaled	direction_next_20	Ensemble	0.76	0.63
financial_scaled	direction_next	GRU_medium	0.77	0.62
financial_long_unscaled	direction_next_10	LightGBM	0.72	0.62
financial_scaled	direction_next_5	Logistic	0.74	0.60
financial_unscaled	direction_next	LSTM_medium	0.77	0.62
financial_long_scaled	direction_next_20	MLP	0.72	0.60
financial_long_scaled	direction_next_20	Random Forest	0.76	0.63
financial_unscaled	direction_next_20	XGBoost	0.71	0.61

Mayor exactitud por modelo

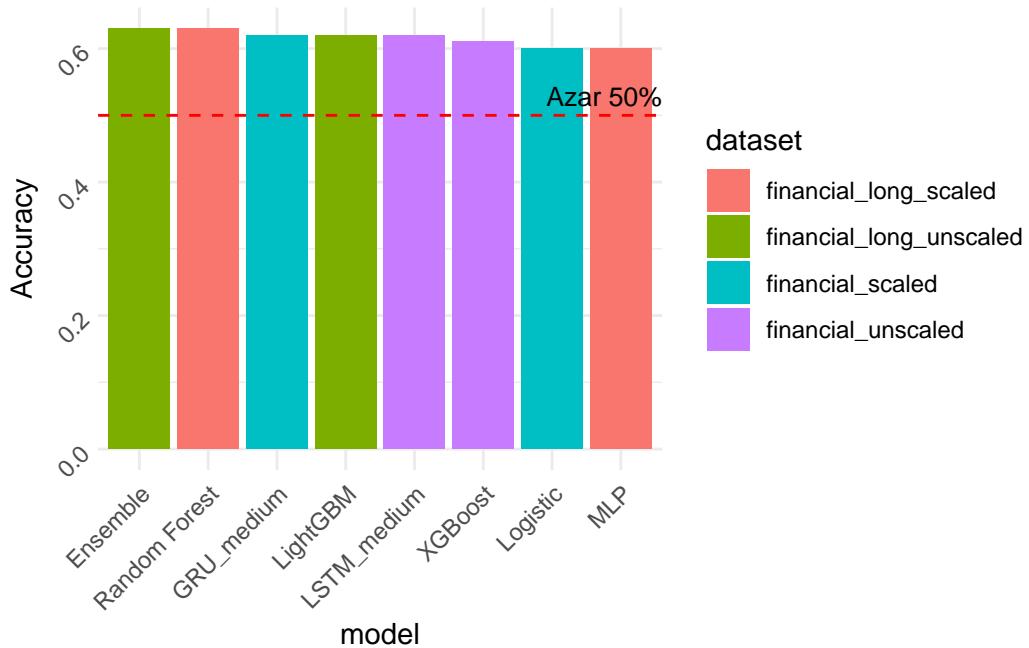


Figure 21: Mayor exactitud por modelo

#### Mejor modelo por target

Finalmente, se muestra en la siguiente tabla el mejor modelo obtenido en cada uno de los 4 targets usados.

Table 30: Mejor modelo por target

Dataset	Target	Modelo	F1	Accuracy
financial_scaled	direction_next	GRU_medium	0.77	0.62
financial_long_scaled	direction_next_10	Random Forest	0.75	0.62
financial_long_scaled	direction_next_20	Random Forest	0.76	0.63
financial_unscaled	direction_next_5	LSTM_medium	0.76	0.62

#### 6.5.2.2 ¿Mejoran las predicciones al incorporar GDELT?

Uno de los objetivos de este estudio era comprobar si la inclusión de análisis de sentimientos desde GDELT mejora o aporta ruido a la predicción de futuros valores del Ibex35.

##### 6.5.2.2.1 Regresión

A priori parece que los datos de sentimientos mejoran considerablemente algunos modelos. Analicemos estadísticamente estas diferencias.

Table 31: Mejor modelo por target

Escala	Modelo	Target	Direccion accuracy financial	Direction Accuracy sentiment	Diferencia

unscaled	GRU_medium	returns_next_20	38.14	60.47	22.33
unscaled	GRU_medium	returns_next	47.01	61.97	14.96
unscaled	GRU_medium	returns_next_10	38.22	52.00	13.78
scaled	Ensemble	returns_next_20	48.51	60.43	11.91
scaled	LSTM_medium	returns_next_20	39.07	50.70	11.63
scaled	MLP	returns_next_10	45.31	56.73	11.43
scaled	MLP	returns_next_5	49.20	58.80	9.60
scaled	Random Forest	returns_next_20	45.11	51.91	6.81
unscaled	Random Forest	returns_next_20	45.11	51.91	6.81
scaled	LightGBM	returns_next_10	45.31	51.84	6.53

#### Comparación estadística entre pares modelo-target entre diferentes datasets

Para el análisis estadístico entre las predicciones de un mismo modelo sobre dos datasets diferentes (sentiment y financial), usaremos el Test de Diebold-Mariano. El test de Diebold-Mariano (Diebold and Mariano 1995) evalúa si las diferencias en precisión predictiva entre dos modelos de pronóstico son estadísticamente significativas, analizando directamente la serie temporal de errores de predicción. El test es robusto a autocorrelación en los errores y no requiere que los modelos estén anidados ni que los errores sigan una distribución normal, lo que lo hace especialmente adecuado para series financieras donde estas condiciones raramente se cumplen. Permite además especificar diferentes funciones de pérdida (MSE, MAE), adaptándose a distintos criterios de evaluación según el contexto aplicado. Para la predicción de rendimientos del IBEX35, este test cuantifica si el valor añadido del análisis de sentimiento de noticias (GDEL) produce mejoras estadísticamente significativas en la precisión de los pronósticos, controlando por la estructura temporal inherente a los datos financieros y los distintos horizontes de predicción evaluados (1, 5, 10 y 20 días).

Para el cálculo del intervalo de confianza en la predicción usaremos el Bootstrap. El bootstrap (Efron and Tibshirani 1994) es un método de remuestreo no paramétrico que permite estimar la distribución muestral de una estadística sin asumir una forma distribucional específica en los datos subyacentes. Mediante la generación de múltiples muestras con reemplazo del conjunto de evaluación, se obtiene una distribución empírica de las métricas de rendimiento que posibilita el cálculo de intervalos de confianza robustos. Esta aproximación es particularmente relevante en series financieras como el IBEX35, donde los supuestos de normalidad y homocedasticidad raramente se cumplen. Mientras que los tests de hipótesis (McNemar, Diebold-Mariano) determinan la significación estadística de las diferencias entre modelos, el bootstrap cuantifica la magnitud e incertidumbre de dichas diferencias, proporcionando una caracterización más completa del valor añadido del análisis de sentimiento. Esta combinación de inferencia frecuentista y estimación por remuestreo sigue las recomendaciones metodológicas actuales para evaluación rigurosa de modelos predictivos en finanzas.

En nuestras comparaciones de dataset1 respecto a dataset2, si el valor de la diferencia es  $>0$ , el modelo mejora con dataset2. Si el intervalo de confianza no cruza el 0, es una mejora significativa; y si además  $p\text{-value} < 0.05$ , la mejora es estadísticamente significativa.

En el caso de regresión, obtenemos que el 13 comparaciones de 60 (21.7%) obtienen una mejora significativa cuando se añaden los datos de sentimiento de GDEL. En el caso de comparar los datasets sólo con variables financieras, series corta (desde 2014) y seria larga (desde 2004), obtenemos una mejora en el 30% de los casos.

=====

RESUMEN DE COMPARACIONES ESTADÍSTICAS

=====

-----

COMPARACIÓN: Financial\_vs\_Sentiment

-----

Total: 60 comparaciones  
 Significativas (DM): 13 (21.7%)  
 Significativas (Bootstrap): 16 (26.7%)  
 Dataset1 mejor RMSE: 38 | Dataset2 mejor RMSE: 22

-----

COMPARACIÓN: Financial\_vs\_Financial\_long

-----

Total: 60 comparaciones  
 Significativas (DM): 18 (30.0%)  
 Significativas (Bootstrap): 16 (26.7%)  
 Dataset1 mejor RMSE: 45 | Dataset2 mejor RMSE: 15

Table 32: Mejor bootstrap por comparación. Datos financieros y de sentimientos

Modelo	Target	Dataset1	Dataset2	boot_mean_diff
LightGBM	returns_next_10	financial_scaled	sentiment_scaled	0.07
Linear	returns_next_10	financial_scaled	sentiment_scaled	0.06
Ridge	returns_next_10	financial_scaled	sentiment_scaled	0.06
Linear	returns_next_5	financial_scaled	sentiment_scaled	0.02
Ridge	returns_next_5	financial_scaled	sentiment_scaled	0.02
LSTM_medium	returns_next_20	financial_scaled	sentiment_scaled	0.02
LightGBM	returns_next	financial_scaled	sentiment_scaled	-0.10
GRU	returns_next_10	financial_unscaled	sentiment_unscaled	-360.31
GRU	returns_next	financial_unscaled	sentiment_unscaled	-936.68
GRU	returns_next_5	financial_unscaled	sentiment_unscaled	-9595.42

Table 33: Mejor bootstrap por comparación. Datos financieros, serie corta y serie larga

Modelo	Target	Dataset1	Dataset2	boot_mean_diff
GRU	returns_next_5	financial_unscaled	financial_long_unscaled	281.29
Linear	returns_next_10	financial_scaled	financial_long_scaled	0.21
Ridge	returns_next_10	financial_scaled	financial_long_scaled	0.21
XGBoost	returns_next_5	financial_scaled	financial_long_scaled	0.14
Linear	returns_next_20	financial_scaled	financial_long_scaled	0.11

Ridge	returns_next_20	financial_scaled	financial_long_scaled	0.11
XGBoost	returns_next_5	financial_unscaled	financial_long_unscaled	0.10
LightGBM	returns_next	financial_unscaled	financial_long_unscaled	0.09
Linear	returns_next	financial_scaled	financial_long_scaled	0.08
Ridge	returns_next	financial_scaled	financial_long_scaled	0.08

GDELT mejora un 7% los resultados de LightGBM, con dataset scaled y target returns\_next\_10, uno de los mejores modelos.

#### 6.5.2.2.2 Clasificación

Table 34: Mejor modelo por target

Escala	Modelo	Target	Accuracy financial	Accuracy sentiment	Diferencia
scaled	LSTM_medium	direction_next_20	0.48	0.59	0.11
unscaled	LightGBM	direction_next	0.48	0.56	0.07
scaled	MLP	direction_next_5	0.51	0.58	0.07
scaled	GRU_medium	direction_next_20	0.54	0.60	0.06
scaled	LSTM_medium	direction_next	0.45	0.51	0.06
scaled	Logistic	direction_next_10	0.49	0.53	0.04
unscaled	LightGBM	direction_next_10	0.50	0.55	0.04
scaled	LightGBM	direction_next	0.56	0.60	0.04
scaled	LightGBM	direction_next_20	0.53	0.57	0.04
scaled	LightGBM	direction_next_10	0.55	0.59	0.04

Comparación estadística entre pares modelo-target entre diferentes datasets

Para comparar las predicciones de clasificación entre dos modelos sobre el mismo conjunto de datos, utilizamos el Test de McNemar. El test de McNemar (Dietterich 1998) es el método estadístico estándar para comparar el rendimiento predictivo de dos clasificadores evaluados sobre el mismo conjunto de datos. A diferencia de comparar simplemente las tasas de acierto, este test analiza específicamente las discordancias entre modelos (casos donde uno acierta y el otro falla), evaluando si estas diferencias son estadísticamente significativas mediante una distribución chi-cuadrado. Su principal ventaja es que no asume independencia entre las predicciones, requisito que se viola cuando ambos modelos se evalúan sobre las mismas observaciones. Dietterich (1998) demostró que el test de McNemar presenta un error tipo I controlado y mayor potencia estadística que alternativas como el t-test pareado o la validación cruzada repetida para comparación de clasificadores. En el contexto de predicción de movimientos del IBEX35, este test permite determinar si la incorporación de variables de sentimiento derivadas de GDELT mejora significativamente la capacidad de clasificar correctamente la dirección del mercado (subida/bajada) respecto a modelos basados únicamente en indicadores técnicos y financieros.

Observando el resumen de las comparaciones estadísticas den el caso de clasificación, observamos que en 3 de las 56 comparaciones (5.4%) se obtiene una mejora significativa al añadir los datos de sentimiento de GDELT. Analizando la comparación dentre series financieras cortas (desde 2014) y largas (desde 2004), detectamos que 10 de 56 comparaciones (17.9%) obtienen una mejora significativa.

=====

RESUMEN DE COMPARACIONES ESTADÍSTICAS

=====

-----

COMPARACIÓN: Financial\_vs\_Sentiment

-----

Total: 56 comparaciones  
Significativas (McNemar): 3 (5.4%)  
Significativas (Bootstrap): 3 (5.4%)  
Dataset1 mejor: 22 | Dataset2 mejor: 25

-----

COMPARACIÓN: Financial\_vs\_Financial\_long

-----

Total: 56 comparaciones  
Significativas (McNemar): 10 (17.9%)  
Significativas (Bootstrap): 12 (21.4%)  
Dataset1 mejor: 24 | Dataset2 mejor: 27

Table 35: Mejor bootstrap por comparación. Datos financieros y de sentimientos

comparison_type	model	target	scaling	diff
Financial_vs_Sentiment	LightGBM	direction_next	unscaled	0.07
Financial_vs_Sentiment	GRU_medium	direction_next_5	scaled	-0.09
Financial_vs_Sentiment	GRU_medium	direction_next	unscaled	-0.14

Table 36: Mejor bootstrap por comparación. Datos financieros, serie corta y serie larga

comparison_type	model	target	scaling	diff
Financial_vs_Financial_long	GRU_medium	direction_next_5	unscaled	0.23
Financial_vs_Financial_long	GRU_medium	direction_next_20	unscaled	0.19
Financial_vs_Financial_long	LSTM_medium	direction_next	scaled	0.13
Financial_vs_Financial_long	LightGBM	direction_next_10	unscaled	0.12
Financial_vs_Financial_long	Logistic	direction_next_10	scaled	0.11
Financial_vs_Financial_long	GRU_medium	direction_next	unscaled	-0.07
Financial_vs_Financial_long	GRU_medium	direction_next_10	scaled	-0.08
Financial_vs_Financial_long	LightGBM	direction_next_5	scaled	-0.09
Financial_vs_Financial_long	LSTM_medium	direction_next_10	scaled	-0.10
Financial_vs_Financial_long	GRU_medium	direction_next_10	unscaled	-0.24

GDELT mejora la predicción de clasificación un 7% en el modelo LightGBM con datos

sin escalar y target dirección\_next. Por otro lado, empeora la predicción, aporta ruido, para GRU, dirección\_next sin escalar y GRU, dirección\_next\_5 escalados.

#### 6.5.2.3 Comparación entre horizontes

En este punto compararemos si es mejor predecir a corto plazo (1 día) o a más largo plazo (20 días). Comparamos los resultados de los modelos en función del horizonte de predicción.

##### 6.5.2.3.1 Regresión

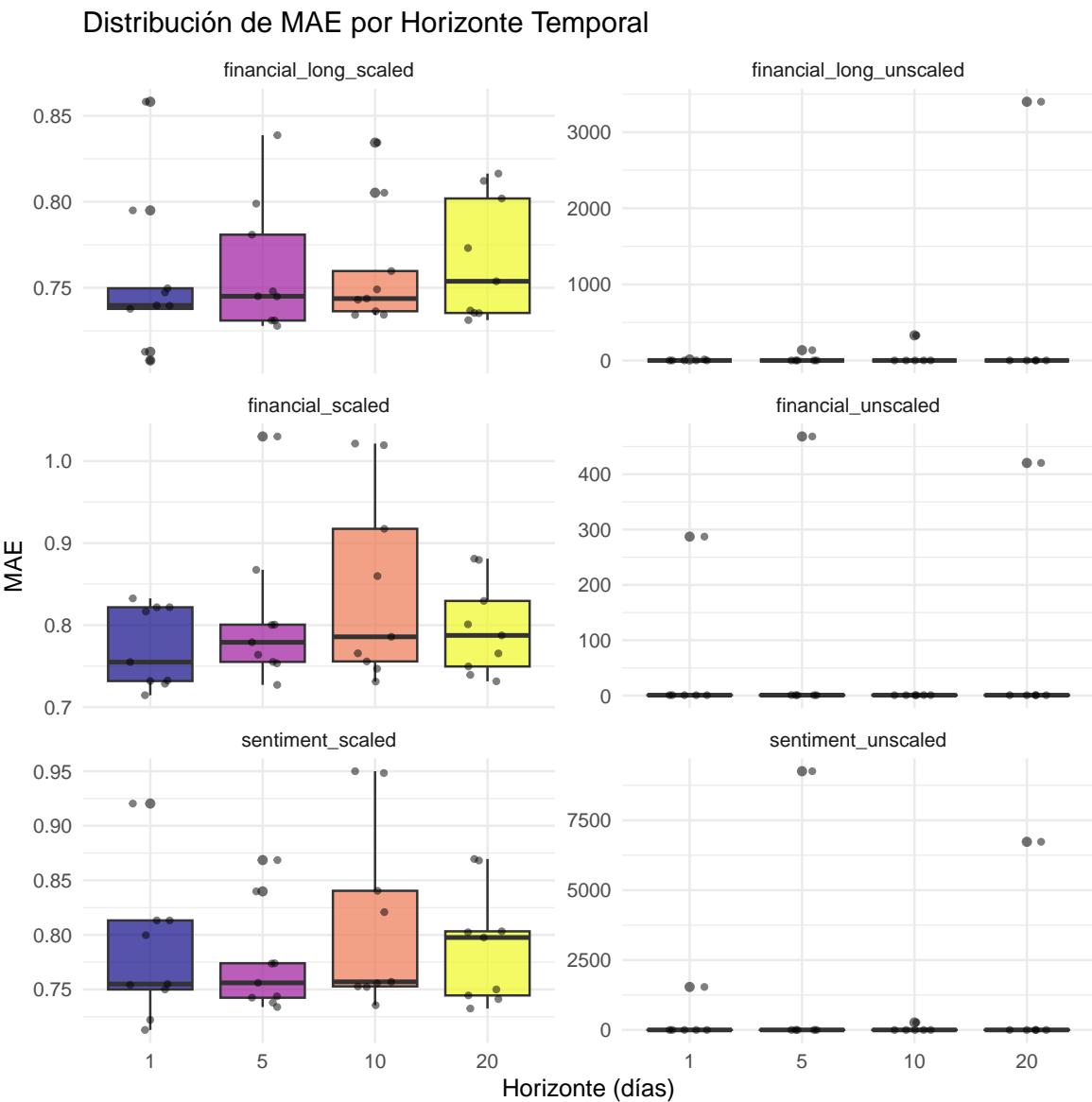


Figure 22: Distribución de los errores en función del horizonte y dataset usado

##### 6.5.2.3.2 Clasificación

### Evolución de Accuracy por Horizonte Temporal

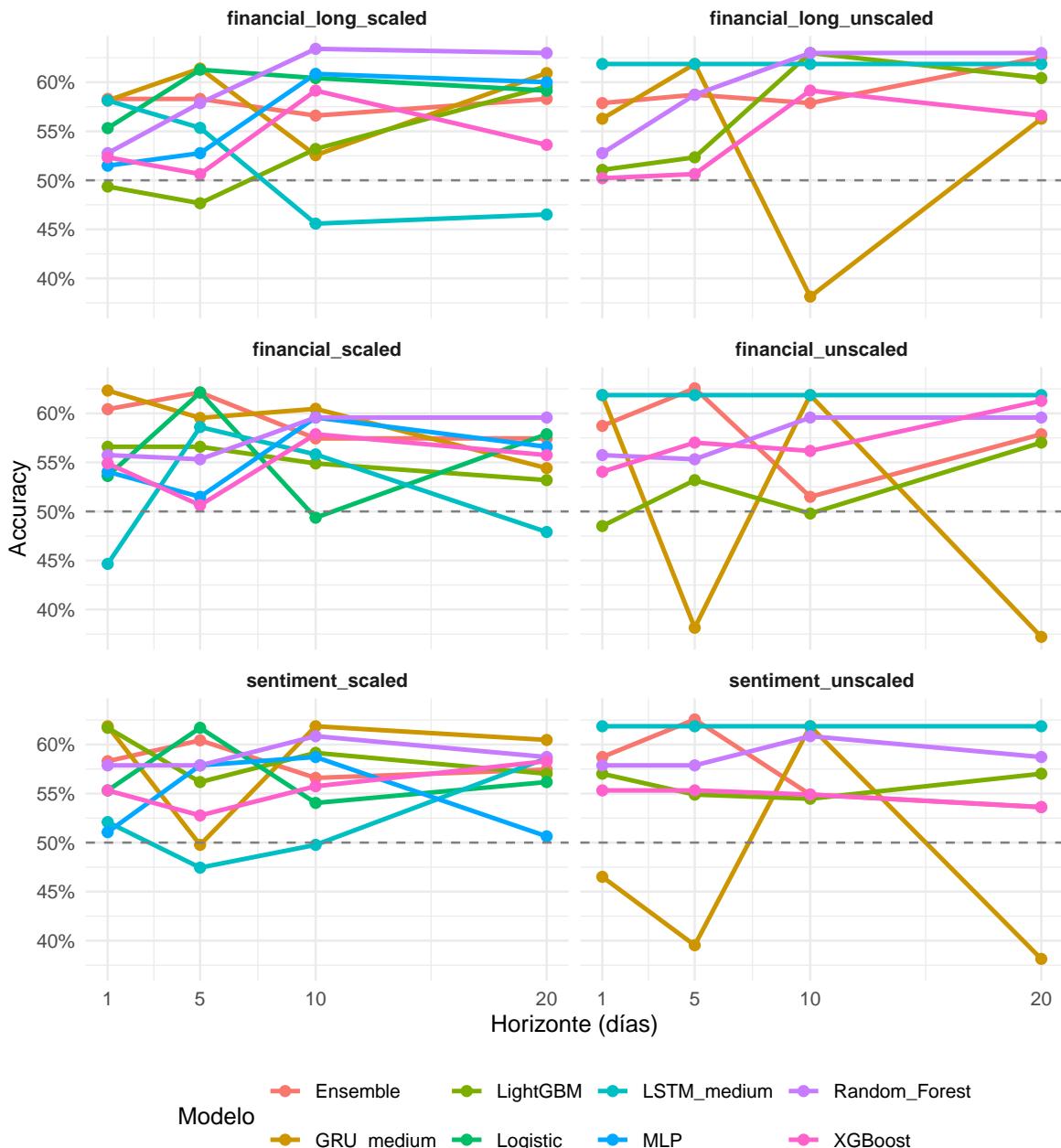


Figure 23: Precisión de horizontes por dataset usado

#### 6.5.2.3.3 Horizonte óptimo

## Resumen Ejecutivo: Efecto del Horizonte Temporal

### Comparaciones Significativas entre Horizontes

Línea roja = 5% esperado por azar

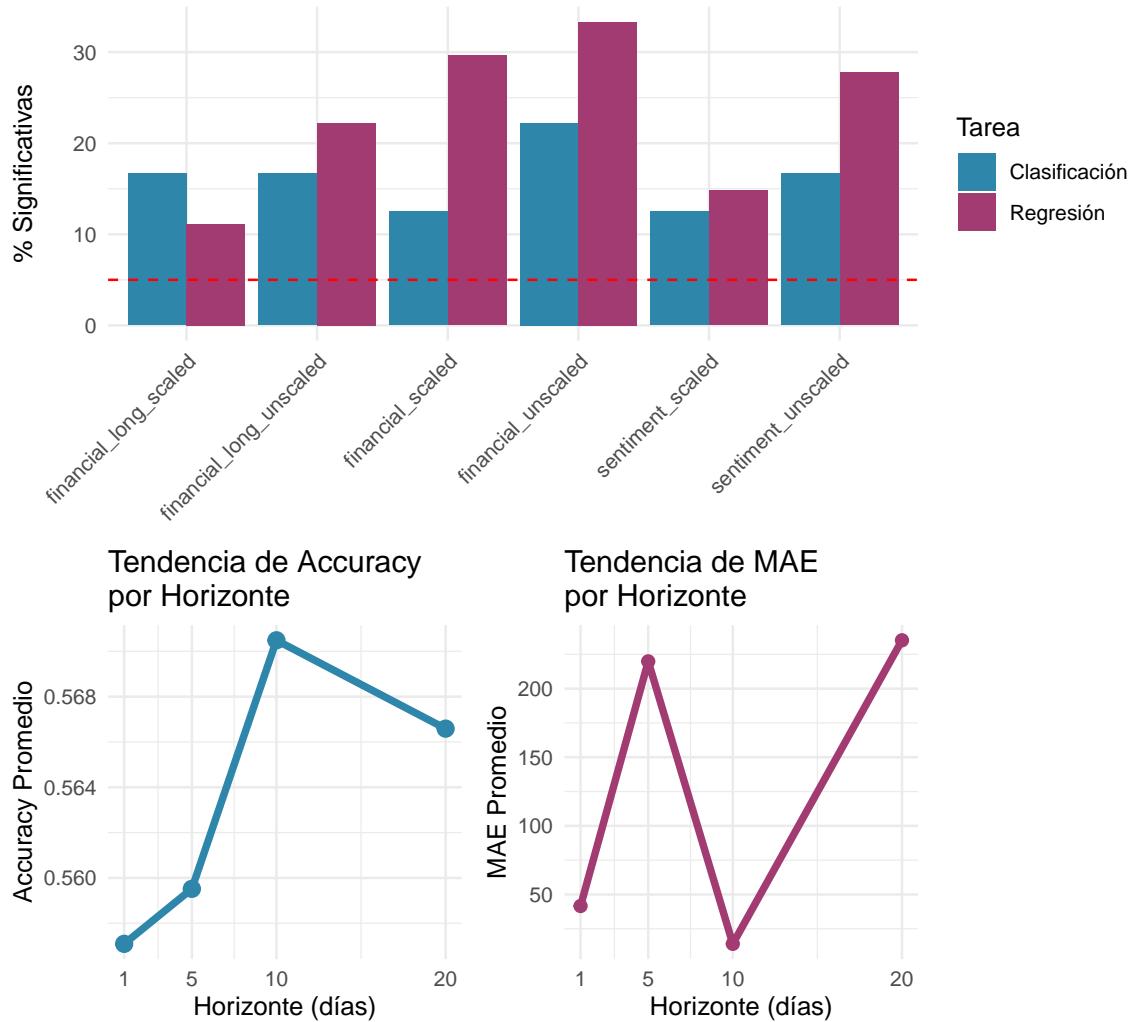


Figure 24: Efecto del horizonte temporal. Parte superior, porcentaje de comparaciones significativas entre horizontes. Parte inferior izquierda, Precisión promedia en cada horizonte temporal. Parte inferior derecho, MAE promedio en cada horizonte temporal

### 6.5.3 Evaluación de predicciones

¿Son reales las predicciones? ¿Cómo se comparan estas predicciones con las de un modelo naïve, aquel que predice siempre la clase mayoritaria? El problema del modelo ingenuo lo observamos en el caso de los modelos clásicos, los cuales pronosticaban con una precisión del ~60%. Estos modelos siempre pronosticaban subida y acertaban el 60% de las veces porque el test está en un ciclo de subida donde el 60% de los días son de subida.

La función `evaluate_all_models()` en el script `code/R/22.evaluate_naive_models.R` evalúa múltiples modelos de clasificación para determinar si sus predicciones son **estadísticamente significativas** o si se comportan como un **clasificador ingenuo** (naive classifier).

Un clasificador ingenuo es aquel que predice siempre la clase mayoritaria o que no aporta información útil más allá del azar. En problemas de predicción financiera con clases desbalanceadas, un modelo puede aparentar buen accuracy simplemente prediciendo siempre “sube” si el mercado sube el 55% de los días. La función realiza los siguientes pasos:

1. **Cálculo de métricas de rendimiento:** Para cada modelo, calcula métricas clave como accuracy, balanced accuracy, F1-score, precision, recall y Cohen's Kappa.
2. **Comparación con modelo ingenuo:** Compara las métricas del modelo con las de un clasificador ingenuo que siempre predice la clase mayoritaria.
3. **Test estadístico:** Realiza un test binomial para comprobar si la accuracy del modelo es significativamente mejor que la del clasificador ingenuo.
4. **Criterios Naïve:** Define criterios para considerar un modelo como naïve:
  - Si la Cohen's Kappa es menor o igual a 0.1, indicando que el modelo no tiene mejor acuerdo que el azar.
  - Si el balanced accuracy es menor o igual a 0.52, indicando que el modelo no es mejor que el azar en clases desbalanceadas.
  - Si el F1-score es menor o igual a 0.1, indicando que el modelo no tiene un buen equilibrio entre precisión y recall. Un clasificador ingenuo (que predice siempre la clase mayoritaria) tendrá  $F1 \approx 0$  en la clase minoritaria porque no la predice nunca.
5. **Veredicto:** Basándose en las métricas y el test estadístico, asigna un veredicto a cada modelo si cumple con dos o más criterios naïve:
  - “REAL”: Si el modelo supera significativamente al clasificador ingenuo.
  - “NAIVE”: Si el modelo no supera significativamente al clasificador ingenuo. El resultado es un data frame que resume el rendimiento y veredicto de cada modelo, permitiendo identificar cuáles modelos aportan valor real en la predicción frente a aquellos que simplemente replican el comportamiento del clasificador ingenuo.
6. **Filtrado:** Finalmente, filtra los modelos para quedarse solo con aquellos que son considerados “REALES”, Cohen's Kappa  $>0.1$  y p-value del test binomial  $<0.05$ , es decir, que superan significativamente al clasificador ingenuo.

### 6.5.3.1 Clasificación

#### 6.5.3.1.1 Evaluar modelos

Aplicando los filtros descritos anteriormente, obtenemos que el único modelo que cumple con todas las características es “LightGBM” para el problema de clasificación, dataset escalado y target “direction\_next\_5”. La contraparte del modelo entrenado con datos de sentimientos o con la serie larga de datos financieros muestran un p-value  $>0.05$ . A su vez, el modelo “LightGBM” que mostró un aumento en la precisión en el target “direccion\_next” al incorporar datos de sentimientos, también mostró un p-value binomial  $<0.05$  y una Cohen's Kappa  $<0.1$ .

Table 37: Modelos reales de clasificación

Variable	Valor
Dataset	financial_scaled
Target	direction_next_5
Model	LightGBM
N_samples	250
Prevalence_Class_0	0.396
Prevalence_Class_1	0.604
Accuracy	0.568
Balanced_Accuracy	0.5624
Cohen_Kappa	0.1214
Sensitivity	0.5354
Specificity	0.5894
F1_Score	0.4953
Precision_Class_0	0.6593
Recall_Class_0	0.5894
F1_Class_0	0.6224
Precision_Class_1	0.4609
Recall_Class_1	0.5354
F1_Class_1	0.4953
Minority_Correct	53
Minority_Total	99
P_Value_Binomial	0.003445
Naive_Criteria_Count	0
Is_Naive	FALSE
Verdict	REAL

### 6.5.3.2 Regresión

En el caso de los modelos de regresión, y para poder compararlos con los de clasificación, convertimos sus predicciones a clasificación, asignando 1, si el retorno predicho es positivo, o asignando 0 si el retorno predicho es negativo.

#### 6.5.3.2.1 Evaluar modelos

Aplicando los filtros descritos anteriormente, obtenemos que los modelos que cumplen con todas las características son 9, destacando “XGBoost” y “LightGBM” para el problema de regresión. Destacando XGBoost por su predicción a más largo plazo. A su vez, el modelo “LightGBM” que mostró un aumento en la precisión en el target “return\_next\_10” al incorporar datos de sentimientos escalados, también mostró un p-value binomial  $<0.05$  y una Cohen’s Kappa  $<0.1$ .

Table 38: Modelos reales de regresión

X1	X2	X3	X4	X5	X6

Dataset	financial long scaled	financial unscaled	financial long unscaled	financial scaled	financial long unscaled	sentiment scaled
Target	returns next 20	returns next 5	returns next 20	returns next	returns next	returns next 10
Model	XGBoost	LightGBM	XGBoost	LightGBM	LightGBM	MLP
N_samples	235	250	235	254	254	245
Prevalence_Class_0	0.3787	0.3960	0.3787	0.3976	0.3976	0.3878
Prevalence_Class_1	0.6213	0.6040	0.6213	0.6024	0.6024	0.6122
Accuracy	0.5617	0.5160	0.5532	0.5630	0.5669	0.5673
Balanced_Accuracy	0.5705	0.5698	0.5636	0.5632	0.5598	0.5598
Cohen_Kappa	0.1315	0.1212	0.1182	0.1217	0.1168	0.1161
Sensitivity	0.6067	0.8283	0.6067	0.5644	0.5248	0.5263
Specificity	0.5342	0.3113	0.5205	0.5621	0.5948	0.5933
F1_Score	0.5118	0.5754	0.5070	0.5067	0.4907	0.4854
Precision_Class_0	0.6903	0.7344	0.6847	0.6615	0.6547	0.6642
Recall_Class_0	0.5342	0.3113	0.5205	0.5621	0.5948	0.5933
F1_Class_0	0.6023	0.4372	0.5914	0.6078	0.6233	0.6268
Precision_Class_1	0.4426	0.4409	0.4355	0.4597	0.4609	0.4505
Recall_Class_1	0.6067	0.8283	0.6067	0.5644	0.5248	0.5263
F1_Class_1	0.5118	0.5754	0.5070	0.5067	0.4907	0.4854
Minority_Correct	54	82	54	57	53	50
Minority_Total	89	99	89	101	101	95
P_Value_Binomial	0.000011	0.000000	0.000011	0.000517	0.006504	0.004222
Naive_Criteria_Count	0	0	0	0	0	0
Is_Naive	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Verdict	REAL	REAL	REAL	REAL	REAL	REAL

### 6.5.3.3 Unir resultados de evaluación

Table 39: Número de modelos por tipo

Tipo de modelo	Veredicto	N
Clasificación	INGENUO	154
Clasificación	REAL	13
Clasificación	NA	1
Clásico	INGENUO	4
Regresión	INGENUO	135
Regresión	REAL	45

Table 40: Modelos REALES tras evaluación naïve

Dataset	Target	Modelo	Balanced Accuracy	Cohen's Kappa	P-Value Binomial
financial_scaled	direction_next_5	LightGBM	0.56	0.12	0.003
financial_long_scaled	returns_next_20	XGBoost	0.57	0.13	0.000
financial_long_unscaled	returns_next	LightGBM	0.56	0.12	0.007
financial_long_unscaled	returns_next_20	XGBoost	0.56	0.12	0.000
financial_scaled	returns_next	LightGBM	0.56	0.12	0.001
financial_scaled	returns_next	XGBoost	0.56	0.11	0.000
financial_unscaled	returns_next	LightGBM	0.55	0.10	0.011
financial_unscaled	returns_next_5	LightGBM	0.57	0.12	0.000
sentiment_scaled	returns_next_10	MLP	0.56	0.12	0.004
sentiment_scaled	returns_next_5	Ridge	0.56	0.10	0.000

## 6.6 Fase 6: Validación final con datos nuevos

Basándonos en los resultados de la evaluación de modelos, seleccionamos los mejores modelos para la validación final con datos nuevos.

### 6.6.1 Descarga de nuevos datos

Para la validación final de los modelos seleccionados, se descargan datos nuevos del Ibex35 desde 2025-10-17 hasta 2025-11-30.

### 6.6.2 Preparación de datos para validación

Se preparan los datos nuevos para la validación final de los modelos seleccionados. Para ello se hace uso del script `code/R/24.scaling_validation_data.R`, el cual usa los parámetros de normalización (mean & sd) usados en el Z-score del conjunto de entrenamiento y los aplica a los nuevos datos OHLCV de todos los índices usados.

```
source("../code/R/24.scaling_validation_data.R")
# Crear archivo de log
log_file <- file.path(paste0("../", "output/log", "fase_6_log_validation_",
  format(Sys.time(), "%Y%m%d_%H%M%S"), ".txt"))

log_cat("== Iniciando archivo Log Fase 6: Validación ==\n")
log_cat(paste0("== Fecha: ", format(Sys.time(), "%Y%m%d_%H%M%S"), " ==\n\n"))
# Cargar parámetros de normalización
scaling_params <- readRDS(file = paste0("../",
  output_path, "/RData/ML_financial/scaler_zscore.rds"))
#scaling_params <- readRDS(file = paste0("../",
  output_path, "/RData/ML_financial_long/scaler_zscore.rds"))

start_date <- min(validation_data$date)
end_date <- max(validation_data$date)
```

```

# Datos Euribor
# Configurar API key (solo primera vez)
fredr_set_key("399e99f309c87a112f4e4bb45e502578")

# Descargar Euribor a 3M
euribor <- fredr(
  series_id = c("IR3TIB01EZM156N"),
  observation_start = as.Date(start_date),
  observation_end = as.Date(end_date)
)

# Renombrar columnas
euribor <- euribor %>%
  dplyr::select(date, value) %>%
  rename(euribor_3m = value)

# Crear secuencia completa de fechas (días hábiles o todos los días)
fechas_completas <- data.frame(
  date = seq(min(euribor$date), max(euribor$date), by = "day")
)

# Unir y llenar NA hacia adelante
euribor_completo <- fechas_completas %>%
  dplyr::left_join(euribor, by = "date") %>%
  fill(euribor_3m, .direction = "down")
euribor_long <- euribor_completo %>%
  mutate(
    index = "euribor_3m",
    close = euribor_3m,
    open = NA,
    high = NA,
    low = NA,
    volume = NA
  ) %>%
  dplyr::select(date, index, open, high, low, close, volume)

validation_data <- bind_rows(validation_data, euribor_long)

validation_data <- validation_data |>
  mutate(
    index = as.factor(index),
    date = as.Date(date)
  )

# 1. Calcular features SIN escalado
validation_unscaled <- calculate_validation_features(
  all_stocks_df = validation_data,

```

```

validation_start = "2025-10-17",
validation_end = "2025-11-30",
target_index = "ibex35",
scaler_path = NULL,
min_history_days = 252,
verbose = TRUE
)

# 2. Calcular features CON escalado
validation_scaled <- calculate_validation_features(
  all_stocks_df = validation_data,
  validation_start = "2025-10-17",
  validation_end = "2025-11-30",
  target_index = "ibex35",
  scaler_path = paste0("../",
    ↵ output_path, "/RData/ML_financial/scaler_zscore.rds"), # ruta al
    ↵ archivo de scaler
  min_history_days = 252,
  verbose = TRUE
)
# Guardar datos preparados
saveRDS(validation_unscaled, file =
  ↵ "../output/RData/validation_unscaled.rds")
saveRDS(validation_scaled, file = "../output/RData/validation_scaled.rds")
#guardar en CSV
write_csv(validation_unscaled, file =
  ↵ "../output/tables/validation_unscaled.csv")
write_csv(validation_scaled, file =
  ↵ "../output/tables/validation_scaled.csv")

```

### 6.6.3 Evaluar modelos

#### 6.6.3.1 LightGBM classification - financial\_scaled - direction\_next\_5

```

import sys
sys.path.append("../code/py")

from validate_model import (
    load_model,
    load_validation_data,
    verify_features,
    prepare_data,
    evaluate_model,
    EXPECTED_FEATURES)

model, is_clf =
  ↵ load_model("../data/py_models_checking/clf_fin_scaled_lightgbm_direction_next_5/LightGBM"

```

```

df = load_validation_data("../output/tables/validation_scaled.csv")
verify_features(df, EXPECTED_FEATURES)
X, y, dates = prepare_data(df, target_col="direction_next_5")
results = evaluate_model(model, X, y, dates)

```

Cargando modelo desde: ../data/py\_models\_checking/clf\_fin\_scaled\_lightgbm\_direction\_next\_5/L  
 Modelo cargado: LGBMClassifier  
 Tipo detectado: Clasificador  
 Cargando datos desde: ../output/tables/validation\_scaled.csv  
 Datos cargados: 31 filas, 51 columnas

==== VERIFICACIÓN DE FEATURES ====  
 Todas las features esperadas están presentes  
 True

==== PREPARACIÓN DE DATOS ====  
 Target: direction\_next\_5  
 Observaciones con target válido: 26 de 31

==== EVALUACIÓN DEL MODELO (CLASIFICACIÓN) ====  
 --- Métricas de Clasificación ---  
 Accuracy: 0.4615 (46.15%)  
 Precision: 0.5882  
 Recall: 0.5882  
 F1-Score: 0.5882  
 AUC-ROC: 0.4444

--- Comparación con Baseline ---  
 Baseline (% clase 1): 0.6538 (65.38%)  
 Mejora sobre random: -3.85 pp  
 Mejora sobre baseline: -19.23 pp

--- Matriz de Confusión ---  

		Predicho
	Neg	Pos
Real Neg	2	7
Real Pos	7	10

--- Distribución de Predicciones ---  
 Predicciones positivas: 17 (65.4%)  
 Predicciones negativas: 9 (34.6%)

--- Análisis Temporal ---  
 Accuracy por semana:  
 aciertos total accuracy

week			
42	1	1	1.0
43	4	5	0.8
44	1	5	0.2
45	4	5	0.8
46	0	5	0.0
47	2	5	0.4

### 6.6.3.2 Evaluar LightGBM regression - financial\_unscaled - return\_next\_5

```
import sys
sys.path.append("../code/py")

from validate_model import (
    load_model,
    load_validation_data,
    verify_features,
    prepare_data,
    evaluate_model,
    EXPECTED_FEATURES)
# Regresor LightGBM
model, is_clf =
    ↳ load_model("../data/py_models_checking/reg_fin_unscaled_lightgbm_returns_next_5/LightGBM")
df = load_validation_data("../output/tables/validation_unscaled.csv")
verify_features(df, EXPECTED_FEATURES)
X, y, dates = prepare_data(df, target_col="returns_next_5")
results = evaluate_model(model, X, y, dates)
```

Cargando modelo desde: ../data/py\_models\_checking/reg\_fin\_unscaled\_lightgbm\_returns\_next\_5/LightGBM  
 Modelo cargado: LGBMRegressor  
 Tipo detectado: Regresor  
 Cargando datos desde: ../output/tables/validation\_unscaled.csv  
 Datos cargados: 31 filas, 51 columnas

==== VERIFICACIÓN DE FEATURES ====  
 Todas las features esperadas están presentes  
 True

==== PREPARACIÓN DE DATOS ====  
 Target: returns\_next\_5  
 Observaciones con target válido: 26 de 31

==== EVALUACIÓN DEL MODELO (REGRESIÓN) ====  
 --- Métricas de Regresión ---  
 RMSE: 1.1517  
 MAE: 0.8747

R<sup>2</sup>: -0.4831

MSE: 1.3263

--- Métricas Direccionales (signo del retorno) ---

Accuracy direccional: 0.4231 (42.31%)

Baseline (% positivos): 0.6538 (65.38%)

Mejora sobre random: -7.69 pp

--- Matriz de Confusión (Dirección) ---

	Predicho	
	Neg	Pos
Real Neg	3	6
Real Pos	9	8

--- Estadísticas de Predicciones ---

Media real: 0.1386

Media predicha: 0.0181

Std real: 0.9644

Std predicha: 0.4389

Correlación: -0.2709

--- Análisis Temporal ---

Métricas por semana:

week	aciertos_dir	total	accuracy_dir	error_medio	error_std
42	0	1	0.0	1.8866	NaN
43	2	5	0.4	0.2004	0.5854
44	2	5	0.4	-0.3263	0.7950
45	3	5	0.6	0.2939	1.5144
46	3	5	0.6	-0.8257	1.2538
47	1	5	0.2	0.9071	0.8481

### 6.6.3.3 Evaluar XGBoost regression - financial\_long\_scaled - return\_next\_20

```
import sys
sys.path.append("../code/py")

from validate_model import (
    load_model,
    load_validation_data,
    verify_features,
    prepare_data,
    evaluate_model,
    EXPECTED_FEATURES
)
```

```

model, is_clf =
    ↵ load_model("../data/py_models_checking/reg_fin_long_scaled_xgboost_returns_next_20/XGBoost")
df = load_validation_data("../output/tables/validation_scaled.csv")
verify_features(df, EXPECTED_FEATURES)
X, y, dates = prepare_data(df, target_col="returns_next_20")
results = evaluate_model(model, X, y, dates)

```

Cargando modelo desde: ../data/py\_models\_checking/reg\_fin\_long\_scaled\_xgboost\_returns\_next\_20  
Modelo cargado: XGBRegressor  
Tipo detectado: Regresor  
Cargando datos desde: ../output/tables/validation\_scaled.csv  
Datos cargados: 31 filas, 51 columnas

==== VERIFICACIÓN DE FEATURES ====  
Todas las features esperadas están presentes  
True

==== PREPARACIÓN DE DATOS ====  
Target: returns\_next\_20  
Observaciones con target válido: 11 de 31

==== EVALUACIÓN DEL MODELO (REGRESIÓN) ====  
--- Métricas de Regresión ---  
RMSE: 1.3145  
MAE: 1.1246  
R<sup>2</sup>: -0.4477  
MSE: 1.7280

--- Métricas Direccionales (signo del retorno) ---  
Accuracy direccional: 0.3636 (36.36%)  
Baseline (% positivos): 0.6364 (63.64%)  
Mejora sobre random: -13.64 pp

--- Matriz de Confusión (Dirección) ---  

	Predicho	
	Neg	Pos
Real Neg	2	2
Real Pos	5	2

--- Estadísticas de Predicciones ---  
Media real: -0.1136  
Media predicha: 0.0023  
Std real: 1.1459  
Std predicha: 0.4707  
Correlación: -0.2911

--- Análisis Temporal ---

Métricas por semana:

week	aciertos_dir	total	accuracy_dir	error_medio	error_std
42	1	1	1.0	-1.3966	NaN
43	2	5	0.4	-0.9104	1.2530
44	1	5	0.2	0.9348	0.7525

## 7 Discusión

El objetivo principal de este proyecto era implementar y evaluar la capacidad predictiva de modelos de *machine learning* para anticipar los movimientos direccionales del IBEX35, uno de los índices más importantes para la economía española, incorporando variables financieras y métricas de análisis de sentimientos extraídas de GDELT. El desarrollo o evolución de los mercados financieros está sujeto a multitud de factores, tanto políticos, sociales o económicos, lo que hace que su predicción de futuros sea una labor difícil, e incluso con cierta naturaleza caótica.

El rendimiento obtenido en los modelos de clasificación direccional oscila entre el 56 y 62% de exactitud, siendo los modelos aquellos con arquitectura LSTM y GRU con rangos de datos cortos (2014-2024) (Tabla 28). Aunque éstas cifras apenas mejoran el azar (50%) están en línea con el rango superior de algunos estudios ya comentados en la introducción (Fischer and Krauss 2018; Da, Engelberg, and Gao 2015). Esto podría atribuirse al extenso trabajo en el *feature engineering* realizado. En el caso de los modelos de regresión, la precisión se sitúa en el 62.22 y 61.97% para los modelos basados en LSTM y GRU respectivamente, con los grupos de datos correspondientes a la serie larga (2004-2024) (Tabla 23).

Analizando la comparación del aporte a los modelos del análisis básico de sentimientos de GDELT, observamos que en un primer análisis podríamos hablar de un incremento en la precisión de hasta 22 puntos en los modelos GRU para clasificación (Tabla 31) y 11 puntos para clasificación en los modelos LSTM. Estos resultados podrían indicar que existe una mejora predictiva con el uso del análisis de sentimientos básico de GDELT. Sin embargo, cuando se comparan las diferencias estadísticas entre valores reales y predichos, mediante los test de Diebold-Mariano para regresión y McNemar para clasificación, combinado con *bootstrap* para el cálculo del intervalo de confianza y su significados, la mayoría de modelos no son estadísticamente significativos. En el caso de los modelos de regresión, sólo 13 modelos (21,7%) mostró una diferencia significativa entre la incorporación o no de datos de sentimientos de GDELT (Tabla 32), siendo el modelo con mayor diferencia LightGBM entrenado con la serie corta escalada con z-score (2014-2024) y a un horizonte de 10 días. En el caso de clasificación, sólo 3 modelos (5,4%) resultaron significativamente diferentes, siendo el único que aporta una mejora en la incorporación de GDELT LightGBM, entrenado con serie financiera corta sin escalar y a un horizonte de 1 día, el cual ofrece una diferencia de un 7%.

Analizando el efecto del horizonte temporal, se observa como el porcentaje de comparaciones significativas es generalmente mayor en los modelos de regresión. A su vez, se observa un incremento en la precisión de los modelos de clasificación a 10 y 20 días, mientras que en los modelos de regresión el cambio es más irregular, presentando mayor error a

5 y 20 días y menor error a uno y 10 días.

Para comprobar si las predicciones de estos modelos eran satisfactorias se procedió a convertir a clasificación los modelos de regresión, en función del signo del retorno predicho. Esto nos permite comparar más correctamente modelos de regresión y clasificación. Se hizo uso de las métricas de rendimiento de la matriz de confusión y el test estadístico binomial. El objetivo era diferenciar aquellos modelos con una capacidad predictiva real de aquellos que sólo pronostican la clase mayoritaria. Esta diferenciación es importante especialmente ya que la serie histórica del IBEX35 (2004-2025) presenta un 53% de días de subida y 47% de días de bajada. En el caso del periodo de test, el 59,84% de los días son de subida, por lo que un modelo que siempre predice la serie mayoritaria tendría un acierto del 59,84%.

Haciendo uso de los criterios naïve (de ingenuidad) ( $F1$ -balanceada > 0,52,  $F1$ -score > 0,1, Cohen's Kappa > 0,1 y significancia en el test binomial) clasificamos a los modelos como reales o ingenuos. De esta forma, obtenemos que el 33% de los modelos de regresión son modelos reales, el 8,38% de los modelos de clasificación son reales y el 100% de los modelos clásicos son modelos ingenuos.

Finalmente, de los modelos reales, los tres modelos que mostraron mejor predicción real son LightGBM para clasificación y LightGBM y XGBoost para regresión. Estos tres modelos se probaron con un nuevo set de datos comprendido entre el 16 de octubre de 2025 al 30 de noviembre de 2025, mostrando una clara incapacidad en la predicción de nuevos valores futuros. Esto pone de manifiesto que aunque son modelos que superan todas las métricas establecidas, su precisión es similar o inferior a un modelo que siempre pronostique subida. Esta caída en la precisión de los modelos en el periodo de evaluación podría indicar que o bien las relaciones aprendidas por los modelos entre 2004-2024, no aplican a 2025, por lo que estaríamos ante un cambio de régimen de mercado. O más plausible, que los modelos se han sobre-ajustado a los valores de entrenamiento y test y no poseen capacidad predictiva real, o que las variables técnicas y financieras no logran capturar todos los comportamientos del mercado.

## 8 Conclusiones

El presente Trabajo Fin de Máster ha desarrollado y evaluado un pipeline completo de machine learning para la predicción de movimientos direccionales del IBEX35, integrando indicadores técnicos, variables de mercados internacionales y análisis de sentimiento de noticias mediante GDELT. Las principales conclusiones derivadas del estudio son las siguientes:

1. Los modelos de machine learning evaluados (Random Forest, XGBoost, LightGBM, MLP, GRU y Ensemble) alcanzan precisiones de clasificación direccional entre el 53% y el 62%, superando marginalmente el azar pero sin diferencias estadísticamente significativas entre arquitecturas. El mejor modelo individual (GRU con 62.39%) se sitúa en el rango superior de los resultados reportados en la literatura para mercados desarrollados.
2. Las variables de sentimiento extraídas de GDELT no proporcionan mejoras estadísticamente significativas sobre los modelos basados exclusivamente en datos financieros. Las diferencias observadas no superan los umbrales de significancia en los tests de McNemar o Diebold-Mariano ni en los intervalos de confianza bootstrap, sugiriendo que el sentimiento mediático público se incorpora rápidamente a los precios del IBEX35.

3. Ni el horizonte de predicción (1, 5, 10 o 20 días) ni la profundidad histórica de entrenamiento (10 vs 20 años) producen diferencias consistentes y estadísticamente significativas en el rendimiento predictivo. Los datos más recientes parecen ser al menos tan informativos como series históricas más largas que atraviesan múltiples regímenes de mercado.
4. Todos los modelos exhiben un sesgo pronunciado hacia la predicción de movimientos alcistas, reflejando la distribución histórica del índice. Este sesgo limita severamente la aplicabilidad práctica de los modelos para estrategias que requieran identificar correctamente ambas direcciones del mercado.

Como líneas de trabajo futuro, se propone explorar técnicas de NLP más sofisticadas (modelos transformer, análisis específico por compañías), incorporar datos alternativos de mayor frecuencia, implementar estrategias de balanceo de clases y evaluar el rendimiento en diferentes regímenes de volatilidad. Asimismo, sería valioso extender el análisis a otros índices bursátiles para evaluar la generalización de los hallazgos. En síntesis, este trabajo demuestra que la predicción de movimientos direccionales del IBEX35 mediante *machine learning* y análisis de sentimiento presenta limitaciones fundamentales en el contexto evaluado.

## Referencias

- Anthropic. 2024. "Claude Sonnet 4.5." <https://www.anthropic.com>.
- Berben, Robert-Paul, and W. Jos Jansen. 2005. "Testing for Granger Causality: A Note on the Lag Length." *Economics Letters* 88 (3): 405411. <https://doi.org/10.1016/j.econlet.2005.04.005>.
- Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–94. ACM. <https://doi.org/10.1145/2939672.2939785>.
- Cho, Kyunghyun, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. "Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation." *arXiv Preprint arXiv:1406.1078*.
- Da, Zhi, Joseph Engelberg, and Pengjie Gao. 2015. "The Sum of All FEARS Investor Sentiment and Asset Prices." *Rev Financ Stud* 28 (1): 1–32. <https://doi.org/10.1093/rfs/hhu072>.
- Diebold, Francis X., and Roberto S. Mariano. 1995. "Comparing Predictive Accuracy." *Journal of Business & Economic Statistics* 13 (3): 253–63. <https://doi.org/10.1080/07350015.1995.10524599>.
- Dietterich, T. G. 1998. "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms." *Neural Computation* 10 (7): 1895–1923. <https://doi.org/10.1162/089976698300017197>.
- Efron, Bradley, and R. J. Tibshirani. 1994. *An Introduction to the Bootstrap*. Chapman; Hall/CRC. <https://doi.org/10.1201/9780429246593>.
- Fischer, Thomas, and Christopher Krauss. 2018. "Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions." *European Journal of Operational Research* 270 (2): 654–69. <https://doi.org/10.1016/j.ejor.2017.11.054>.

- Granger, C. W. J. 1980. "Testing for Causality." *Journal of Economic Dynamics and Control* 2 (January): 329–52. [https://doi.org/10.1016/0165-1889\(80\)90069-X](https://doi.org/10.1016/0165-1889(80)90069-X).
- Granger, Clive W. J. 1969. "Investigating Causal Relations by Econometric Models and Cross-Spectral Methods." *Econometrica : Journal of the Econometric Society* 37 (3): 424438. <https://doi.org/10.2307/1912791>.
- Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9 (8): 1735–80. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." In *Advances in Neural Information Processing Systems*, 30:3146–54.
- Klioutchnikov, Igor, Mariia Sigova, and Nikita Beizerov. 2017. "Chaos Theory in Finance." *Procedia Computer Science* 119 (January): 368–75. <https://doi.org/10.1016/j.procs.2017.11.196>.
- Leetaru, Kalev, and Philip A Schrot. n.d. "GDELT: Global Data on Events, Location and Tone,"
- Mandelbrot, Benoit B., and Richard L. Hudson. 2004. *The (Mis)behavior of Markets: A Fractal View of Risk, Ruin, and Reward*. New York: Published by Basic Books.
- Mavrotas, George, and Robert Kelly. 2008. "Selection of Lag Order in Granger Causality Tests." *Journal of Statistical Computation and Simulation* 78 (6): 564584. <https://doi.org/10.1080/00949650601109908>.
- Murphy, John J. 1999. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. New York: New York Institute of Finance.
- Panagiotidis, Theodore, and Emerson Rutledge. 2010. "Oil and Stock Returns: Evidence from European Industrial Sector Indices in a Time-Varying Environment." *Journal of International Financial Markets, Institutions and Money* 20 (5): 571587. <https://doi.org/10.1016/j.intfin.2010.07.002>.
- Poon, Ser-Huang, and Clive W. J. Granger. 2003. "Forecasting Volatility in Financial Markets: A Review." *Journal of Economic Literature* 41 (2): 478539. <https://doi.org/10.1257/jel.41.2.478>.
- Pring, Martin J. 2002. *Technical Analysis Explained: The Successful Investor's Guide to Spotting Investment Trends and Turning Points*. 4th ed. New York: McGraw-Hill.
- Sezer, Omer Berat, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. 2020. "Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review: 2005–2019." *Applied Soft Computing* 90: 106181. <https://doi.org/10.1016/j.asoc.2020.106181>.
- Tetlock, Paul C. 2007. "Giving Content to Investor Sentiment: The Role of Media in the Stock Market." *The Journal of Finance* 62 (3): 1139–68. <https://doi.org/10.1111/j.1540-6261.2007.01232.x>.
- Yubero, María José Gómez, and Miguel Palomero Aguilar. 2024. "Análisis comparado de la representatividad, metodología de cálculo y transparencia del Ibex 35." *CNMV Boletín* I (May): 123–66.