Análisis de Series Temporales y Técnicas de Extracción de Tópicos para la Gestión Dinámica de Carteras

Autor: Fadrique Álvarez De Toledo Abaitua

Director: Ignacio Prieto Funes

Doble Grado en Relaciones Internacionales y Business Analytics

Universidad Pontificia Comillas

15/06/2025

Relación de Siglas y Abreviaciones

AI: Inteligencia Artificial (del inglés, Artificial Intelligence).

APA: American Psychological Association. Estilo de citación y formato para documentos académicos.

ARCH: Autoregressive Conditional Heteroskedasticity (Modelo Autoregresivo Condicionalmente Heterocedástico). Modelo para la volatilidad de series temporales.

ARIMA: Autoregressive Integrated Moving Average (Modelo Autoregresivo Integrado de Medias Móviles). Modelo estadístico para el análisis y predicción de series temporales.

ARIMAX: Autoregressive Integrated Moving Average with Exogenous Variables. Extensión del modelo ARIMA que incluye otras variables independientes.

BERT: Bidirectional Encoder Representations from Transformers. Modelo de lenguaje desarrollado por Google para tareas de NLP.

CVaR: Conditional Value at Risk (Valor en Riesgo Condicional). Medida de riesgo que cuantifica la pérdida esperada en los peores escenarios.

FinBERT: Financial BERT. Versión del modelo BERT pre-entrenada específicamente con textos del dominio financiero.

GARCH: Generalized Autoregressive Conditional Heteroskedasticity (Modelo Autoregresivo Condicionalmente Heterocedástico Generalizado). Extensión del modelo ARCH.

LDA: Latent Dirichlet Allocation (Asignación Latente de Dirichlet). Modelo generativo probabilístico para la extracción de tópicos en colecciones de texto.

LSTM: Long Short-Term Memory (Memoria a Corto y Largo Plazo). Tipo de red neuronal recurrente (RNN) diseñada para aprender dependencias a largo plazo en datos secuenciales.

ML: Machine Learning (Aprendizaje Automático).

MPT: Modern Portfolio Theory (Teoría Moderna de Carteras). Marco teórico para la construcción de carteras de inversión.

NLP: Natural Language Processing (Procesamiento del Lenguaje Natural). Campo de la inteligencia artificial y la lingüística computacional.

NMF: Non-Negative Matrix Factorization (Factorización de Matrices no Negativas). Técnica de reducción de dimensionalidad utilizada, entre otros, para la extracción de tópicos.

RoBERTa: Robustly Optimized BERT Pretraining Approach. Variante optimizada del modelo BERT.

S&P 500: Standard & Poor's 500. Índice bursátil que representa las 500 empresas más grandes por capitalización de mercado de Estados Unidos.

TFG: Trabajo de Fin de Grado.

VADER: Valence Aware Dictionary and sEntiment Reasoner. Herramienta de análisis de sentimiento basada en léxico y reglas, optimizada para textos de redes sociales.

WSJ: The Wall Street Journal. Periódico estadounidense de referencia en economía y finanzas.

Tabla de contenido

1. Introducción

1.1 Objetivos del Trabajo

El objetivo principal de este Trabajo de Fin de Grado (TFG) es desarrollar una estrategia de gestión de carteras que integre modelos predictivos de **series temporales** con técnicas avanzadas de **análisis de datos no estructurados** (principalmente textos financieros). En concreto, se busca combinar predicciones cuantitativas de mercados (ej. precios de acciones) con indicadores extraídos de noticias financieras y redes sociales, con el fin de mejorar la toma de decisiones de inversión. Esta integración pretende aprovechar tanto información **estructurada** (histórico de precios, indicadores financieros) como **no estructurada** (noticias, tweets, informes) para construir una gestión de cartera más dinámica y efectiva.

Los objetivos específicos incluyen: (a) aplicar modelos de series temporales (ARIMA, GARCH, LSTM, Prophet) para pronosticar retornos de activos financieros, (b) extraer **sentimiento** y **tópicos** clave de noticias financieras mediante técnicas de procesamiento de lenguaje natural (NLP), (c) correlacionar y analizar la relación entre las señales textuales y la evolución de precios de mercado, y (d) proponer un esquema de optimización de cartera que incorpore tanto las predicciones cuantitativas como los indicadores textuales. En suma, el TFG pretende demostrar que la integración multidisciplinar de métodos cuantitativos y cualitativos puede ofrecer ventajas en la gestión de inversiones.

1.2 Justificación del Estudio

El volumen creciente de datos no estructurados en finanzas (noticias, comentarios en redes sociales, etc.) tiene un impacto cada vez más reconocido en los mercados financieros. Tradicionalmente, la teoría financiera clásica asume que los precios reflejan fundamentalmente información numérica (estados financieros, indicadores

económicos), pero en la práctica las percepciones, expectativas y el sentimiento de los inversores (a menudo moldeados por noticias y medios) pueden influir de manera notable en la dinámica de precios. Integrar estas fuentes de datos varias es, por tanto, relevante para capturar factores de riesgo y tendencia que los modelos puramente cuantitativos podrían pasar por alto.

Este estudio aporta en ese contexto al combinar **predicción de series temporales** con **análisis textual**. La relevancia radica en que, si el mercado no fuera plenamente eficiente (es decir, si la información pública como las noticias no se incorporara de manera instantánea en los precios), podrían encontrarse señales predictivas explotables. Además, incluso bajo alta eficiencia, entender la relación entre noticias y movimientos de mercado aporta valor para la gestión de riesgo y la interpretación de eventos financieros. En la literatura reciente se ha visto un gran interés por las "finanzas computacionales" y las **finanzas conductuales**, que promueven el uso de técnicas de machine learning y NLP para mejorar las decisiones de inversión. Este TFG se inserta en esa tendencia, explorando un enfoque multidimensional (series temporales + textos) para optimizar carteras, algo que podría generar **valor añadido para inversores** y gestores al mejorar la precisión de pronósticos e identificar señales de mercado difíciles de detectar mediante métodos tradicionales.

1.3 Metodología General

Para enfrentar los objetivos, hemos planteado una metodología en varias etapas combinando análisis cuantitativo y cualitativo. En primer lugar, recopilamos datos históricos de precios de activos (acciones del índice S&P 500) y datos textuales de noticias financieras en un período determinado (2016-2023, que se adaptará para en función de las noticias disponibles para las diferentes empresas). Con los datos cuantitativos, hemos considerado modelos de series temporales como ARIMA, GARCH y redes neuronales LSTM para realizar pronósticos de precios futuros, así como el modelo Prophet desarrollado por Facebook para capturar tendencias y estacionalidades. Por otro lado, los datos textuales (noticias financieras, titulares, e incluso posts de redes

sociales cuando disponibles) se analizan mediante técnicas de NLP: en particular, hemos aplicado **análisis de sentimiento** utilizando herramientas como VADER, TextBlob y modelos basados en Transformers (FinBERT y RoBERTa), y se propone la **extracción de tópicos** con modelos LDA/NMF para identificar temas recurrentes en las noticias. Adicionalmente, hemos considerado la **detección de eventos** financieros relevantes identificando palabras clave en los textos (p. ej., resultados trimestrales, fusiones, cambios regulatorios) para vincularlos con movimientos repentinos del mercado.

Los resultados de los modelos anteriores (predicciones de precios y señales extraídas de textos) se plantea integrarlos en un esquema de **optimización de cartera**. Concretamente, la idea central fue alimentar un modelo de gestión de carteras (basado en la teoría de Markowitz y extensiones) con tanto las previsiones de retornos esperados como con indicadores de sentimiento/tendencias extraídos de noticias. La cartera óptima se ajustaría dinámicamente aprovechando estas fuentes de información, buscando maximizar el desempeño ajustado al riesgo (por ejemplo, maximizando el **ratio de Sharpe**, que mide retorno excedente sobre volatilidad). Esta metodología general abarca diversas técnicas; sin embargo, es importante notar que en el desarrollo efectivo del proyecto se implementó con mayor profundidad el **análisis empírico de sentimiento vs. precios**, mientras que algunas partes propuestas (predicción de series con LSTM/Prophet, extracción de tópicos, modelo final de cartera) quedaron como lineamientos teóricos no implementados por completo debido a limitaciones de alcance. Aun así, en este trabajo describimos dichas propuestas teóricas para sentar las bases de futuros desarrollos.

2. Revisión de la Literatura

En esta sección se revisa el marco teórico relevante en tres áreas clave: (2.1) Modelos de series temporales en finanzas, (2.2) Métodos de análisis de texto aplicados a finanzas, y (2.3) Modelos de gestión de carteras.

2.1 Modelos de series temporales en finanzas

El pronóstico de series financieras (como precios de acciones, índices o volatilidades) se aborda tradicionalmente con modelos estadísticos clásicos y, más recientemente, con enfoques de aprendizaje automático. A continuación, resumimos los principales modelos de series temporales empleados en finanzas:

- **ARIMA (Autoregressive Integrated Moving Average):** Es un modelo estadístico lineal muy utilizado utilizado para series financieras estacionarias o transformadas a estacionarias. Captura relaciones **autorregresivas** y de **promedio móvil** en los datos. La metodología ARIMA, popularizada por Box y Jenkins (1970), tiene un buen desempeño en datos financieros relativamente estables y lineales. Suponer que una serie financiera puede explicarse por sus propios retardos es útil para tendencias a corto plazo, aunque suele requerir suposiciones de linearidad y varianza constante (lo cual rara vez se cumple en mercados muy volátiles). Variantes como **SARIMA** incorporan estacionalidad explícitamente.
- **GARCH (Generalized Autoregressive Conditional Heteroskedasticity):** Introducido por Bollerslev (1986) como generalización del modelo ARCH de Engle (1982), el modelo GARCH aborda una característica crucial de las series financieras: la volatilidad **heterocedástica** y agrupada en el tiempo. En otras palabras, permite que la varianza del error no sea constante sino dependiente de choques pasados, capturando los "clústeres de volatilidad" típicos de mercados (periodos calmados seguidos de periodos turbulentos). Los modelos GARCH y sus extensiones (EGARCH, GJR-GARCH, multivariados, etc.) se han convertido en un estándar para modelar la volatilidad de rendimientos financieros, mejorando la estimación de riesgos e intervalos de confianza en

pronósticos. Por ejemplo, un GARCH(1,1) suele ser un benchmark difícil de superar para predecir volatilidad a corto plazo.

- **Redes Neuronales LSTM:** Las Long Short-Term Memory networks (Hochreiter & Schmidhuber, 1997) son un tipo de red neuronal recurrente diseñada para aprender dependencias de largo plazo en secuencias. En finanzas, las LSTM se han aplicado a predicción de precios y series macroeconómicas debido a su capacidad de modelar relaciones **no lineales** complejas y patrones con memoria más larga que los modelos ARIMA. Estudios comparativos muestran que las LSTM pueden superar a ARIMA bajo ciertas condiciones de datos muy no estacionarios o con interacciones no lineales, detectando señales en medio del ruido de mercado. Sin embargo, requieren grandes cantidades de datos para entrenar y son esencialmente cajas negras, lo que lo hace difícil de interpretar.
- **Prophet:** Es una herramienta de pronóstico de series temporales de código abierto desarrollada por Facebook (Taylor & Letham, 2017) orientada a **descomposición aditiva** de la serie en componentes de tendencia, estacionalidad y días festivos/eventos. Prophet ha ganado popularidad por su facilidad de uso y buenos resultados en datos con fuertes componentes estacionales (p. ej., ventas mensuales, tráfico web). En finanzas, su aplicación ha sido experimental; se ha visto que Prophet maneja bien tendencias de medio plazo y patrones estacionales en series (por ejemplo, ciclos económicos anuales). Sin embargo, dado que los precios de las acciones suelen ser tener mucho ruido y con menos estacionalidad que otras series, Prophet se debería combinar con otros enfoques para capturar componentes más idiosincráticos. En general, cada uno de estos modelos tiene fortalezas y debilidades: ARIMA asume linearidad y funciona bien en entornos relativamente estacionarios, LSTM capta no-linealidad y dinámicas complejas, y Prophet destaca en series con estacionalidades definidas. En la práctica, a menudo se emplean **modelos híbridos** (ej. ARIMA-GARCH, o combinaciones de LSTM con componentes ARIMA) para explotar las ventajas de cada método. La literatura reciente sugiere incluso ensambles que combinan pronósticos de múltiples modelos para mejorar la exactitud.

2.2 Análisis de texto en finanzas

Además de los datos numéricos, en finanzas es crucial analizar la gran cantidad de información textual disponible: noticias de prensa, informes de analistas, posts en redes sociales como Twitter, etc. Estas fuentes cualitativas contienen señales sobre la percepción del mercado que pueden afectar los precios. Revisamos a continuación las principales técnicas de análisis de texto aplicadas al ámbito financiero:

- **Análisis de Sentimiento:** Consiste en convertir texto no estructurado en un indicador cuantitativo que refleje el tono positivo/negativo de la información. En finanzas, el **sentimiento de noticias** o de publicaciones en redes puede actuar como indicador de la confianza o temor de los inversores. Entre las herramientas destacadas están:
    - *VADER (Valence Aware Dictionary and sEntiment Reasoner):* Método léxico basado en reglas, optimizado inicialmente para lenguaje de redes sociales. VADER asigna puntuaciones de sentimiento a oraciones teniendo en cuenta la intensidad de palabras positivas y negativas, incluso con modismos y emoticonos. Aunque simple, ha probado ser muy efectivo en textos cortos y casuales (Hutto & Gilbert, 2014). En finanzas se ha utilizado para analizar tweets o titulares breves, con buenos resultados al correlacionar sentimiento con movimientos diarios del mercado.
    - *TextBlob:* Es una biblioteca de NLP en Python que provee funcionalidades sencillas para análisis de sentimiento, traduciendo texto a una puntuación de polaridad. Internamente, TextBlob emplea un lexicón (derivado de Pattern Analyzer) para asignar polaridad promedio a las palabras y frases. Se utiliza en contexto financiero por su facilidad de implementación, aunque su rendimiento es limitado para textos muy especializados, dado que no está ajustado al lenguaje financiero técnico.
    - *FinBERT:* Modelo de lenguaje basado en BERT, pre-entrenado específicamente en corpora financieros (por ejemplo, noticias de empresas, noticias económicas). Fue introducido por Araci (2019) para abordar la jerga especializada donde diccionarios generales fallan.

FinBERT consigue mejorar la clasificación de sentimiento en finanzas al captar matices que los métodos genéricos etiquetan erróneamente como neutrales. Por ejemplo, distingue que expresiones como *"downgrade"*, *"missed expectations"* son negativas en contexto financiero. Estudios han mostrado que FinBERT supera en métricas de precisión a modelos previos en datasets de noticias económicas.

- *RoBERTa:* Es una variante robusta de BERT propuesta por Liu et al. (2019), entrenada con optimizaciones en el procedimiento de pre-entrenamiento. Aunque RoBERTa es un modelo de propósito general, versiones entrenadas para análisis de sentimiento (como *RoBERTa-large SST* o similares) han obtenido resultados sobresalientes en benchmarks de sentimiento. En finanzas, se han utilizado modelos RoBERTa ajustados con datos de noticias económicas para clasificar sentimiento con alta exactitud. RoBERTa tiende a manejar mejor frases complejas y contextos sutiles que métodos basados en léxico, gracias a su comprensión de lenguaje contextual. Sin embargo, su entrenamiento e interpretación requieren mayores recursos computacionales.

La investigación en finanzas ha validado la utilidad del sentimiento textual. Un estudio influyente de Tetlock (2007) mostró que el tono negativo en columnas de Wall Street Journal tiene poder predictivo sobre retornos subsecuentes del mercado. Del mismo modo, lexicones especializados, como el de Loughran y McDonald (2011) para informes financieros, mejoran la detección de sentimientos relevantes al dominio. En la última década, los enfoques han migrado de conteo de palabras negativas/positivas a modelos de deep learning pre-entrenados, reflejando una tendencia general en NLP: los Transformers están reemplazando metodologías basadas en diccionarios gracias a su mayor capacidad para *"entender"* contexto.

- **Extracción de Tópicos:** Esta técnica busca identificar automáticamente los temas subyacentes en un conjunto de documentos. En finanzas, aplicar **topic modeling** a noticias o reportes permite resumir de qué se está hablando (por ejemplo, detectar un tópico de "crisis financiera" vs. "resultados positivos" en las noticias de cierto periodo). Los dos métodos más conocidos son:

- *LDA (Latent Dirichlet Allocation):* Introducido por Blei, Ng y Jordan (2003), LDA es un modelo generativo que asume que cada documento es una mezcla de unos pocos tópicos, y que cada tópico es una distribución de palabras. A través de inferencia bayesiana, LDA descubre esos tópicos latentes. Su uso en finanzas incluye análisis de noticias económicas para ver qué temas dominan (ej. "política monetaria", "innovación tecnológica", "fusiones y adquisiciones"), vinculándolos luego con movimientos de mercados o sectores. Por ejemplo, algunos estudios han encontrado que ciertos tópicos extraídos de noticias financieras correlacionan con reacciones significativas al día siguiente en bolsa. LDA proporciona una forma de reducir la dimensión de datos textuales y estructurarlos en variables (proporción de tópicos) utilizables en modelos predictivos.
- *NMF (Non-Negative Matrix Factorization):* Es una técnica de factorización matricial que, aplicada a matrices documento-palabra, identifica también componentes temáticos interpretables. Lee y Seung (1999) demostraron que NMF aprende "partes" relevantes de los datos, y en textos puede extraer conjuntos de palabras frecuentes que forman un tópico coherente. A diferencia de LDA, NMF no tiene una base probabilística estricto sino algebraico (descomposición con restricciones de no negatividad). En aplicaciones financieras, NMF ha competido con LDA para extraer tópicos de noticias, a veces ofreciendo interpretaciones más nítidas o diferentes. No obstante, LDA ha prevalecido en popularidad por su base teórica sólida y disponibilidad de implementaciones.

La **utilidad de la extracción de tópicos** en finanzas radica en el *forecasting* de tendencias temáticas: por ejemplo, detectar un aumento en la proporción de noticias sobre "recesión" puede anticipar cambios en volatilidad de mercado. Cheng et al. (2022) apuntan en una revisión que varios estudios han incorporado tópicos extraídos de noticias en modelos de predicción bursátil, encontrando que ciertos tópicos (como noticias de banca central o guerras comerciales) tienen efecto importante en rendimientos futuros.

- **Detección de Eventos:** Más allá de clasificar el sentimiento o tema general de un texto, otra línea importante es extraer **eventos financieros específicos** mencionados en las noticias. Un *evento* podría ser, por ejemplo: *"Apple lanza un nuevo producto"*, *"La Fed sube tasas 0.25%"*, *"Fusiones entre X e Y empresas"*. La detección de eventos suele implicar técnicas de NLP más complejas, que incluyen **reconocimiento de entidades** (compañías, personas, cantidades) y **análisis semántico** para identificar la acción. En investigación reciente, se define un evento financiero como algo que responde a las preguntas "¿qué pasó, a quién, cuándo, dónde?" en contexto económico.

  Los algoritmos de extracción de eventos buscan *disparadores* (trigger words) y sus argumentos: por ejemplo, detectar la palabra "acquisition" junto a entidades empresariales indicaría un evento de adquisición corporativa. Herramientas basadas en plantillas o más recientemente en redes neuronales (como BERT aplicado a extracción de eventos) se usan para procesar noticias en lenguaje natural y aportar representaciones estructuradas de eventos. En finanzas, el interés es alto porque los eventos de noticias suelen ser catalizadores de movimientos volátiles de precios. Identificar automáticamente eventos como *profit warnings*, anuncios de beneficios, cambios de directivos o decisiones regulatorias permite alimentar rápidamente esos datos a modelos de trading algorítmico o alertar a gestores humanos. Por ejemplo, Ding et al. (2014) mostraron que incorporar eventos estructurados de noticias (extraídos con técnicas de NLP) mejoraba la predicción de la dirección de precios de acciones en comparación a usar solo texto bruto o solo datos numéricos. Sin embargo, la detección de eventos es un desafío: requiere un *pipeline* de varias tareas NLP (segmentación de oraciones, etiquetado gramatical, clasificación de tipo de evento, etc.) y a menudo enfrenta ambigüedades. Un reto señalado es que en textos informales (como tweets) los eventos pueden mencionarse de forma implícita o incompleta, lo que dificulta su identificación. Aun así, para fuentes más formales (noticias de agencias, reportes) la extracción de eventos puede ser más fiable y ofrece la ventaja de reducir el "ruido" textual a información estructurada clave. En definitiva, el análisis de texto en finanzas abarca desde medir el *sentimiento agregado* del mercado, pasando por conocer *de qué se habla* (tópicos), hasta *qué pasó exactamente* (eventos). Estas aproximaciones no

son excluyentes y tienden a complementarse; de hecho, algunos trabajos recientes exploran enfoques híbridos que combinan sentimiento + eventos para pronosticar mercados.

2.3 Modelos de gestión de carteras

La gestión de carteras de inversión se apoya en un gran cuerpo teórico que busca optimizar la asignación de activos para equilibrar riesgo y retorno. A continuación, se resumen las bases y algunos modelos relevantes:

- **Teoría Moderna de Carteras (Markowitz):** Propuesta por Harry Markowitz en 1952, establece que un inversor racional debe seleccionar carteras considerando conjuntamente la **rentabilidad esperada** y la **volatilidad** (riesgo). El resultado fundamental es que existe una *frontera eficiente* de carteras que ofrecen el mínimo riesgo posible para cada nivel de retorno esperado. Markowitz formalizó el problema de optimización cuadrática donde se minimiza la varianza de la cartera sujeta a lograr cierto retorno, produciendo una distribución óptima de pesos en los activos. La teoría introduce la noción de **diversificación** cuantitativa: combinar activos no perfectamente correlacionados reduce el riesgo total. Este marco es estático (de un solo período) y asume que las esperanzas de retorno, varianzas y covarianzas son conocidas (o estimables). A pesar de sus supuestos, la MPT revolucionó las finanzas estableciendo el fundamento para modelos posteriores. Su vigencia es tal que sigue siendo punto de partida en muchas aplicaciones, si bien en la práctica la estimación de las entradas (especialmente retornos esperados) es difícil y sujeta a error.
- **Modelo Black-Litterman:** Desarrollado por Fischer Black y Robert Litterman (1990/1992) en Goldman Sachs, este modelo busca solucionar un problema práctico de la teoría de Markowitz: la sensibilidad extrema de la cartera óptima a las estimaciones de retornos esperados de los activos. Black-Litterman combina la información de un portafolio de referencia en equilibrio de mercado (por ejemplo, las capitalizaciones de mercado definen la asignación base) con las

opiniones subjetivas del inversor sobre ciertos activos o sectores. Utilizando un enfoque bayesiano, el modelo integra ambas fuentes en un nuevo vector de retornos esperados ajustados, que luego se usa en la optimización media-varianza estándar. De este modo se generan carteras más razonables y estables. En esencia, Black-Litterman permite incorporar perspectivas activas (por ejemplo "creo que la acción X rendirá un 5% más de lo que el mercado implícitamente descuenta") sin incurrir en portafolios extremos. Desde su introducción, se ha convertido en una herramienta popular en gestión institucional, ya que equilibra el rigor de Markowitz con la flexibilidad de incluir juicios de analistas o modelos. Su uso se extiende a asignación global de activos, donde se mezclan proyecciones macro con el portafolio de mercado.

- **Optimización dinámica de carteras:** La teoría clásica se centró en un solo período, pero en la práctica la gestión de carteras es un proceso dinámico, con ajustes periódicos ante nuevos datos. Los modelos dinámicos consideran cómo optimizar decisiones de inversión en múltiples períodos, incorporando la **evolución temporal** del riesgo-retorno. Un caso particular es la maximización del **ratio de Sharpe** de forma continua. El ratio de Sharpe (Sharpe, 1966) mide el exceso de retorno por unidad de riesgo de una cartera. Maximizarlo en un contexto dinámico equivale a encontrar la estrategia que más aumenta el retorno ajustado a volatilidad a lo largo del tiempo. Existen enfoques de optimización estocástica y de control óptimo (por ejemplo, a través de programación dinámica tipo Bellman) para determinar la política de inversión óptima en cada instante dado el estado del mercado. Otra aproximación reciente es el uso de **reinforcement learning** donde un agente "aprende" a reequilibrar la cartera para maximizar recompensas como el Sharpe acumulado. Los modelos dinámicos suelen integrar restricciones realistas (precios de transacción, límites de pesos, aversión al riesgo variable en el tiempo, etc.). Un hallazgo general es que la **frecuencia de ajuste** importa: estrategias demasiado miopes pueden generar gastos altos, mientras que estrategias de muy largo plazo pueden reaccionar tarde a información nueva. Se proponen también criterios alternativos al Sharpe, como maximizar la **utilidad esperada** con función de utilidad cuadrática o exponencial (en cuyo caso el resultado se relaciona con Markowitz), o minimizar el **CVaR** (riesgo de cola) para inversores muy

adversos al riesgo. En cualquier caso, la gestión dinámica moderna aprovecha métricas calculadas para simular escenarios (backtesting) y ajustar la estrategia.

En la literatura académica, la hipótesis de mercados eficientes de Fama (1970) ha enmarcado durante décadas la discusión: bajo eficiencia, ninguna estrategia informada con datos públicos (como noticias o históricos) debería producir consistentemente un Sharpe superior. Sin embargo, campos como las finanzas conductuales sugieren que existen anomalías y que los inversores no son perfectamente racionales, lo cual deja espacio para que técnicas avanzadas de análisis de datos aporten valor. Modelos de gestión de carteras que incorporan señales de sentimiento o predicciones de machine learning representan intentos de explotar esas posibles ineficiencias de forma constante.

3. Análisis Empírico

En este apartado detallamos el análisis realizado para investigar la relación entre el sentimiento de noticias financieras y los precios de acciones del S&P 500. Describimos los datos utilizados (3.1), las técnicas aplicadas (3.2) y los principales resultados acompañados de visualizaciones (3.3).

3.1 Datos utilizados

Datos de noticias: Se ha recopilado un conjunto de noticias financieras relevantes para compañías del índice S&P 500. Las fuentes incluyen agencias de noticias económicas (por ejemplo, Reuters, Bloomberg, CNBC) y posiblemente agregadores o bases de datos de noticias financieras. Cada noticia en el dataset incluye información como la fecha de publicación, el titular y/o cuerpo de la noticia, y se encuentra asociada a una o varias

empresas específicas. Para este estudio, filtramos las noticias correspondientes a **10 empresas líderes del S&P 500**: Apple (AAPL), Microsoft (MSFT), Alphabet/Google (GOOGL), Amazon (AMZN), Tesla (TSLA), Meta/Facebook (META), Nvidia (NVDA), JPMorgan Chase (JPM), UnitedHealth (UNH) y Visa (V). Estas empresas engloban sectores de tecnología, consumo, financiero y salud, aportando una muestra diversa de distintas dinámicas. El periodo de estudio abarca varios años recientes de manera que incluye distintos entornos de mercado (pre y post-pandemia, fases alcistas y bajistas). Las noticias están estructuradas en un formato tabular, con campos como: empresa, fecha, titular, texto completo, fuente, etc., lo que permite aplicar fácilmente herramientas de análisis de texto.

Datos de precios: Para las mismas empresas mencionadas se obtienen sus precios históricos diarios de las acciones, usando fuentes públicas como la API de *Yahoo Finance* (vía la librería yfinance). En concreto, se extraen los **precios de cierre diarios ajustados** de cada acción, así como del índice S&P 500 en su conjunto (para usarlo como benchmark de mercado). El periodo de precios coincide con el de las noticias, asegurando la comparabilidad temporal. Con los precios diarios calculamos **rendimientos diarios** (porcentuales) para cada activo y para el índice. Adicionalmente, se calcula el **rendimiento excesivo** de cada acción, definido como el rendimiento de la acción menos el rendimiento del S&P 500 en el mismo día. Este exceso captura cuánto mejor (o peor) lo hizo la acción relativa al mercado en cada jornada, lo cual es útil para aislar el componente asociado a la empresa vs. efectos de mercado general.

Preprocesamiento: Se alinean las series de tiempo de precios con las fechas de las noticias. Muchas noticias se publican fuera de horas de mercado; en estos casos, para analizar efectos contemporáneos, consideramos la fecha de la noticia alineada con el retorno del siguiente día de mercado (asumiendo que noticias post-cierre influencian el día siguiente). Eliminamos noticias duplicadas o muy cercanas en tiempo para evitar sobrepeso de la misma información. Para cada empresa, se genera una serie temporal que combina la información textual (puntuaciones de sentimiento diario, ver siguiente sección) con la serie de retornos diarios de la acción. Cabe señalar que no todas las

empresas tienen la misma cobertura mediática; empresas más grandes tienden a tener más noticias diarias. Esto se tiene en cuenta normalizando algunas métricas (por ejemplo, calculando un promedio diario de sentimiento si había múltiples noticias ese día para una empresa).

3.2 Técnicas aplicadas

El análisis empírico implementado se centra en evaluar cuantitativamente la correlación y relación temporal entre **sentimiento de noticias** y **rendimientos de las acciones**. Las principales técnicas y pasos aplicados son:

- **Cálculo de sentimiento en noticias:** Se emplean cuatro métodos distintos de análisis de sentimiento para cada noticia:
    1. *VADER:* Se aplica el algoritmo VADER (introducido en sección 2.2) para cada titular o texto de noticia. VADER devuelve un **compound score** entre -1 y 1 que refleja el sentimiento global (combinando polaridad positiva/negativa y su intensidad). Este score considera negaciones, intensificadores y puntuación, siendo apropiado para texto breve como titulares.
    2. *TextBlob:* Se utiliza la función TextBlob.sentiment para extraer la polaridad media del texto de la noticia. TextBlob da un valor entre -1 (muy negativo) y +1 (muy positivo). Aunque TextBlob no es específico de finanzas, sirve como comparación al ser otra herramienta lexicón-based.
    3. *FinBERT:* Se carga un modelo pre-entrenado FinBERT para clasificación de oraciones financieras en positivo, negativo o neutral. Cada noticia es analizada con FinBERT para obtener probabilidades de pertenecer a cada clase; se deriva un score continuo de sentimiento (por ejemplo, tomando prob(Positivo) - prob(Negativo)). FinBERT, al entender lenguaje financiero, puede manejar jerga que confundiría a

VADER/TextBlob (p. ej., “underweight rating” es negativo aunque la palabra *“underweight”* no lo parezca fuera de contexto).

4. *RoBERTa (fine-tuned):* Se emplea un modelo RoBERTa ajustado para sentiment analysis (no específicamente financiero, pero general). Similar a FinBERT, se obtiene un score continuo o probabilidad de sentimiento positivo.

Estas cuatro herramientas permiten contrastar métodos léxicos sencillos (VADER, TextBlob) vs. métodos avanzados basados en **Transformers** (FinBERT, RoBERTa). Para cada empresa y cada día, si hay múltiples noticias, se agregan los scores de sentimiento diarios (por ejemplo, haciendo el promedio de el sentimiento de todas las noticias de Apple del día). Así se construye una serie temporal diaria de sentimiento para cada método y cada empresa.

- **Cálculo de métricas financieras diarias:** Con los precios de cierre de cada acción, calculamos los **rendimientos diarios logarítmicos o porcentuales**. Además, se calcula el **rendimiento en exceso** como mencionado. Por simplicidad, en el análisis correlacional inicial se enfoca en dos variables dependientes: (a) el rendimiento diario de la acción $r_{i,t}$ y (b) el rendimiento excesivo sobre el mercado $r_{i,t} - r_{mkt,t}$. La idea es ver si el sentimiento de noticias sobre la empresa *i* en el día *t* explica parte de la variación de $r_{i,t}$ o de su exceso, tanto en el mismo día como con rezago.
- **Análisis de correlación contemporánea:** Calculamos las correlaciones de Pearson entre la serie diaria de sentimiento y la serie de rendimiento diario correspondiente, para cada método de sentimiento y cada empresa. En total, obtenemos tablas comparativas de correlaciones (por ejemplo, qué correlación tiene el sentimiento VADER con los retornos de Apple, de Microsoft, etc.). También se evalua la significancia estadística de estas correlaciones (contraste t de si la correlación difiere de 0, considerando el tamaño de muestra). Este análisis básico apunta a responder: *¿Cuándo el sentimiento en noticias es alto (positivo), tiende la acción a subir ese mismo día?*.
- **Análisis de retardos (lags):** Para investigar posibles efectos retardados de la información textual, realizamos un análisis de correlación cruzada entre sentimiento y retornos en distintas fases temporales. Es decir, se correlaciona el

sentimiento del día *t* con el rendimiento del día *t+1*, *t+2*, …, *t+30*. Esto para cada método de sentimiento. De esta forma se identifican retardos en los que la correlación es máxima. Un pico de correlación a lag 0 puede indicar impacto inmediato, mientras que correlaciones significativas a lags positivos sugieren que el sentimiento anticipa movimientos futuros (predicción) o, a lags negativos, que el mercado influye en el sentimiento con retraso (por ejemplo, caídas de mercado generan noticias de tono negativo después). Este tipo de análisis lo visualizamos mediante gráficos de correlación vs. Lag.

- **Regresiones lineales:** Para complementar la correlación (que asume relación lineal), llevamos a cabo regresiones simples del tipo: $$ r_{i,t} = \alpha + \beta \cdot Sentiment_{i,t} + \varepsilon_{t} $$ donde *Sentiment* es la variable de sentimiento diaria. Esto se hace principalmente para los métodos VADER y FinBERT por ejemplo, y agregando todos los días de todas las empresas (o por empresa individual). Las pendientes β estimadas indican en qué medida un cambio en el sentimiento (una unidad) se asocia con un cambio en el rendimiento diario. Se examina si β es significativamente distinta de cero. Asimismo, obtenemos **R-cuadrados** de estas regresiones para ver qué proporción de la variabilidad de retornos explicaba el sentimiento (anticipando que será baja, dado que los precios están influenciados por innumerables factores). Generamos gráficos de dispersión representando sentimiento vs. rendimiento, con la recta de regresión sobrepuesta, para visualizar la relación y detectar outliers o no linearidades.
- **Comparación entre empresas y sectores:** Dado que se incluyen 10 empresas, se analiza si la relación sentimiento-retornos difieren mucho entre ellas. Se crea una matriz o cuadro comparativo de correlaciones por empresa. También se explora si ciertos sectores (tecnología vs. financiero, por ejemplo) muestran mayor sensibilidad a las noticias. Esto se complementa calculando, para cada empresa, cuál de los cuatro métodos de sentimiento resulta tener mayor correlación con sus retornos, explorando si algunos métodos capturan mejor el sentimiento en ciertos contextos (p.ej., quizás FinBERT destaca en empresas financieras donde la jerga es técnica, etc.).

Cabe destacar que todo el análisis se realiza de manera *in sample* exploratoria. No se llega en esta fase a utilizar los indicadores de sentimiento en un modelo predictivo de trading real ni a optimizar una cartera con ellos, pero los hallazgos buscan sentar las bases para considerar dichas aplicaciones.

3.3 Resultados y visualizaciones

Correlaciones contemporáneas (lag 0): En general, encontramos que el sentimiento de las noticias tiene una correlación positiva con los rendimientos diarios de las acciones, aunque de magnitud **moderada**. Es decir, en días con noticias predominantemente positivas, tiende a haber rendimientos ligeramente mayores (y viceversa). Entre los métodos:

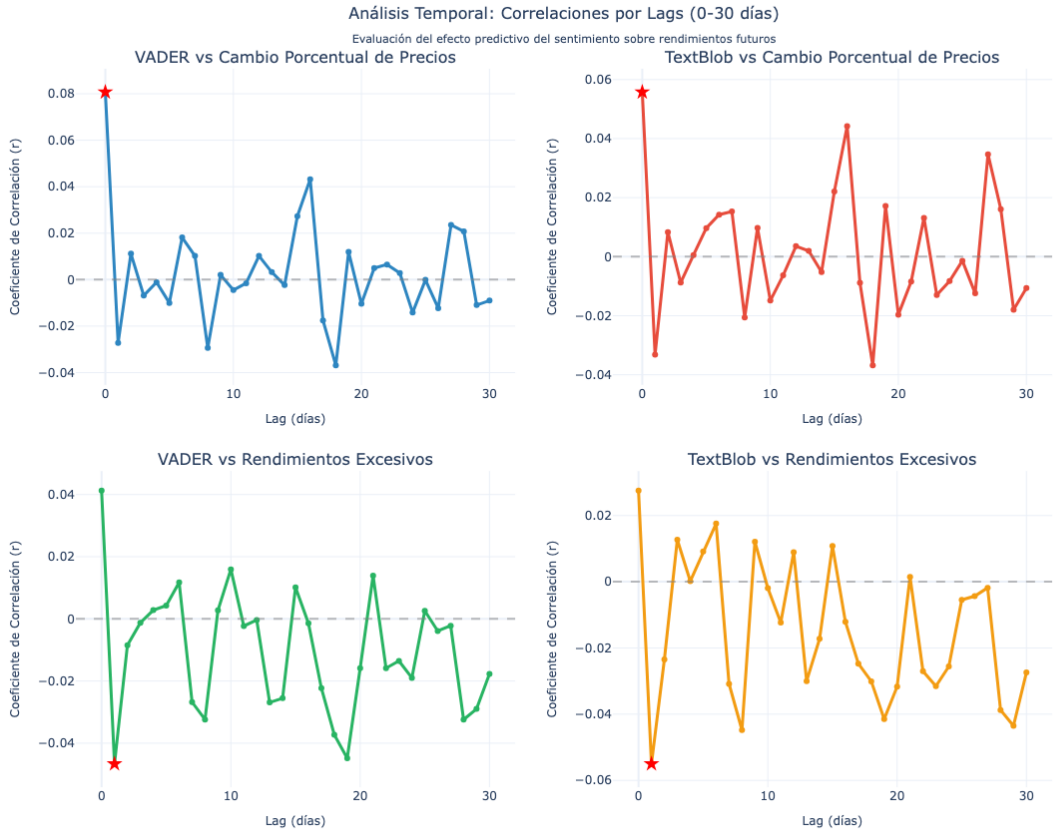


Tabla 1. Análisis Temporal de Correlaciones por Lags: VADER y TextBlob (0–30 días).

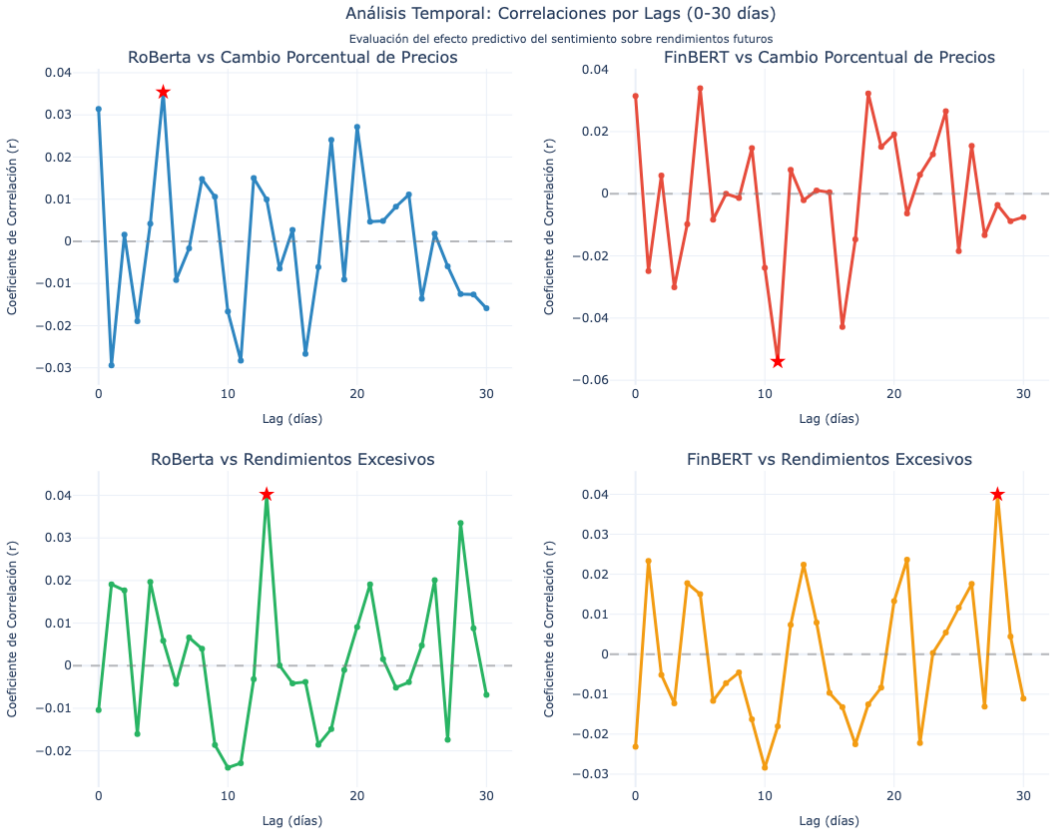


Tabla 2. Análisis Temporal de Correlaciones por Lags: RoBERTa y FinBERT (0–30 días).

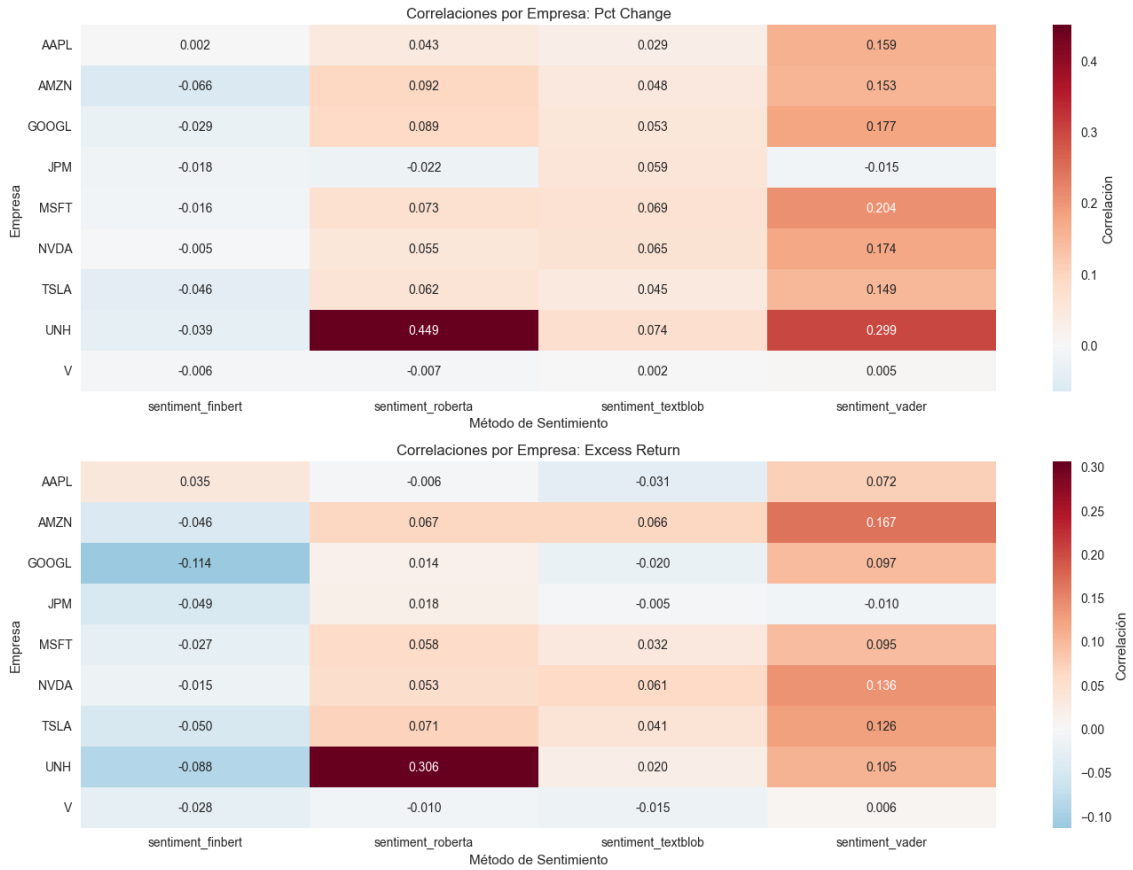


Tabla 3. Correlaciones por Empresa: Sentimiento vs Pct Change y Excess Return.

- **Con VADER observamos correlaciones contemporáneas positivas y consistentes entre el sentimiento y los cambios diarios en precios, particularmente en empresas tecnológicas**. Los coeficientes oscilan entre ~0.05 y ~0.20 en la mayoría de los casos, destacando valores como r = 0.204 para MSFT y r = 0.159 para AAPL en la variable de cambio porcentual. Aunque estos valores **no son altos en términos absolutos**, en el **contexto financiero** (caracterizado por alta volatilidad y ruido) **resultan estadísticamente significativos y potencialmente útiles**. No obstante, VADER no es siempre el método con mejores resultados: por ejemplo, RoBERTa alcanza una correlación anómala de r = 0.449 para UNH, superando con creces al resto. Esto sugiere que, si bien VADER capta bien el tono emocional general de las noticias, **no siempre es el método más predictivo en términos individuales o sectoriales**. Aunque 0.1 de correlación es bajo en términos absolutos, en series financieras puede ser significativo dada la alta volatilidad inherente.

- **TextBlob aporta también correlaciones positivas, pero algo menores que VADER.** Los coeficientes se situan típicamente entre **0.03 y 0.06**, con un máximo cercano a **0.06** en lag 0 para el cambio porcentual de precios. Esto sugiere que **TextBlob capta algo de la señal**, aunque con menor fuerza que VADER. Una posible explicación es que **VADER, al estar calibrado con lenguaje informal, pueda captar mejor ciertos matices de tono presentes en titulares breves y directos**, mientras que **TextBlob, más limitado en su manejo de polaridad y subjetividad, tiende a clasificar más frases como neutras**, perdiendo así sensibilidad ante eventos informativos.

  En cualquier caso, la diferencia no es extrema, y ambos modelos muestran su **mayor capacidad predictiva en el mismo punto: lag 0**, lo que refuerza la idea de que el sentimiento tiene un **impacto más inmediato que diferido** sobre los retornos diarios. No obstante, dado que los coeficientes no superan 0.08 en ningún caso, se concluye que **la fuerza de la relación es moderada**, y aunque estadísticamente significativa en algunos casos, **no resulta suficiente como única base para decisiones de inversión.**

- *FinBERT* tiene un comportamiento interesante: en varios casos la correlación es ligeramente **negativa**, contra la expectativa. Por ejemplo, para algunas empresas las correlaciones FinBERT vs. retorno salen alrededor de **r = -0.05** (no siempre significativo). En promedio estan entre -0.05 y +0.02. Este resultado puede deberse a que FinBERT detecta matices muy específicos: puede marcar noticias como neutral o ligeramente negativas aun cuando el mercado ya las esperaba (y por tanto la acción no cae). Otra posibilidad es que el volumen de noticias negativas es mayor en contextos de caídas previas, creando cierta relación inversa (noticias “malas” siguen a las caídas de días anteriores, generando correlación negativa contemporánea).
- *RoBERTa* muestra correlaciones positivas pequeñas, aproximadamente **0.01 a 0.06**. En ciertos casos, RoBERTa se acerca a VADER (por ejemplo, r ~0.05–0.06 para Tesla). Esto sugiere que un modelo de lenguaje general puede capturar parte del sentimiento direccional, aunque no está especializado. Sus correlaciones siendo menores que VADER implica que, para este dataset

concreto de noticias financieras, el método lexicón sencillo es lo suficientemente bueno o incluso mejor adaptado.

Cuando en lugar de retornos absolutos se usan **rendimientos en exceso sobre el mercado**, las correlaciones se mantienen con signo similar pero magnitud ligeramente reducida. Es decir, una parte del efecto del sentimiento parece estar ligada al componente de mercado general (por ejemplo, un día de muy buen sentimiento coincide con subidas generales del mercado). Aun así, persisten correlaciones positivas (ej: VADER vs exceso de retorno ~0.08 en promedio). Esto refuerza la idea de que las noticias analizadas tienen tanto información de mercado amplio como información específica de la empresa.

```
   ANÁLISIS ESTADÍSTICO DE REGRESIONES:
==========================================

VADER → Pct Change:
  • Correlación (r): +0.1089
  • R-cuadrado: 0.0119 (1.19% varianza explicada)
  • Pendiente: +1.5524 (cambio en Pct Change por unidad de sentimiento)
  • P-valor: 0.000000
  • Significancia: ***

VADER → Excess Return:
  • Correlación (r): +0.0790
  • R-cuadrado: 0.0062 (0.62% varianza explicada)
  • Pendiente: +0.8675 (cambio en Excess Return por unidad de sentimiento)
  • P-valor: 0.000000
  • Significancia: ***

TEXTBLOB → Pct Change:
  • Correlación (r): +0.0328
  • R-cuadrado: 0.0011 (0.11% varianza explicada)
  • Pendiente: +0.6012 (cambio en Pct Change por unidad de sentimiento)
  • P-valor: 0.006595
  • Significancia: **

TEXTBLOB → Excess Return:
  • Correlación (r): +0.0191
  • R-cuadrado: 0.0004 (0.04% varianza explicada)
  • Pendiente: +0.2692 (cambio en Excess Return por unidad de sentimiento)
  • P-valor: 0.114515
  • Significancia: No significativo
```

Tabla 4. Estadísticas de Regresiones Lineales: Coeficientes, R^2 y Significancia para VADER y TextBlob.

Significancia estadística: Los modelos de regresión revelan que los coeficientes de correlación, aunque pequeños, son estadísticamente significativos en muchos casos, particularmente para VADER. Por ejemplo, para Pct Change, VADER obtiene una correlación de r = 0.1089, con una pendiente de +1.55, lo que implica que un cambio completo de -1 a +1 en el score de sentimiento (raro en la práctica) se asociaría con un cambio promedio del 1.55% en el retorno diario. Sin embargo, este tipo de variaciones extremas en sentimiento no son comunes; una mejora más realista de +0.2 en el sentimiento supondría solo un +0.31% en el retorno esperado.

Aunque el p-valor es extremadamente bajo ($p < 0.001$), indicando que la relación no es aleatoria, el R^2 fue de solo 1.2%, lo que refleja una capacidad explicativa limitada. Esto está en línea con la naturaleza multifactorial del movimiento de precios en los mercados financieros.

TextBlob, por su parte, muestra relaciones más débiles: su pendiente y correlación son mucho menores, y en el caso de Excess Return, la relación no es estadísticamente significativa ($p > 0.1$). Esto sugiere que, al menos en este conjunto de datos, los métodos como VADER capturan mejor las señales relevantes del tono de las noticias que TextBlob.

Análisis de retardos (predictivo): Al examinar las correlaciones sentimiento de día *t* vs. retorno de día *t*+*k*, se obtienen algunas observaciones interesantes:

- Para *VADER* y *TextBlob*, el **lag 0** (mismo día) es donde la correlación alcanza su valor máximo (positivo). A lag 1 (sentimiento de hoy con retorno de mañana) las correlaciones aún son positivas pero menores, típicamente se reducen a casi la mitad del valor de lag 0, y muchas no significativas. A partir de lag 2 en

adelante, las correlaciones tienden a oscilar en torno a cero sin patrón claro. Esto sugiere que el mercado reacciona muy rápidamente a la información de las noticias (básicamente el mismo día) y que no queda mucha señal predecible para días posteriores en el sentimiento simple. Este hallazgo es consistente con la hipótesis de eficiencia de mercados, donde la información pública (noticias) se incorpora a los precios casi inmediatamente.

Visualización de relaciones: Hemos generado gráficos de dispersión ilustrando la relación entre sentimiento y rendimiento diario. En ellos se observa la amplia dispersión de puntos que sugiere que la relación es débil pero existente.

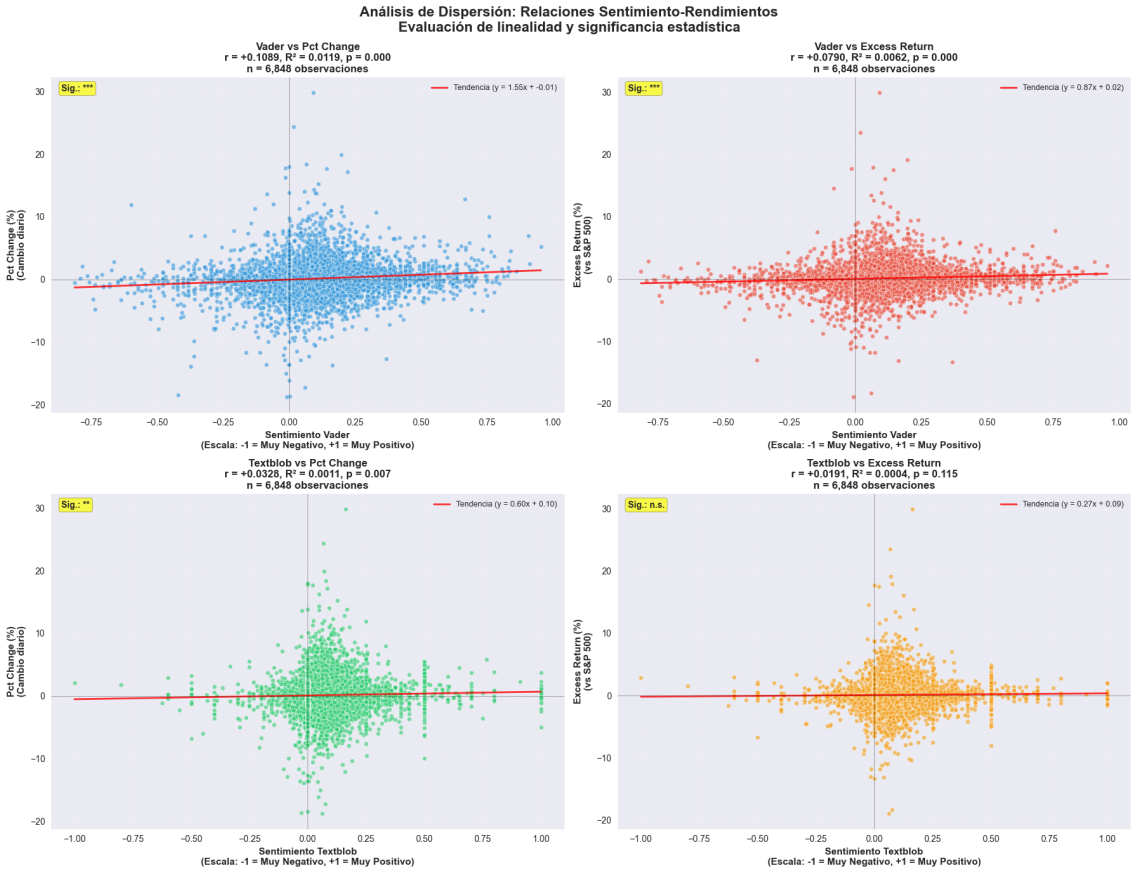


Tabla 5. Análisis de Dispersión entre Sentimiento y Rendimiento Diario: Evaluación de Linealidad y Significancia.

Por ejemplo, en el gráfico sentimiento VADER vs. retorno, se puede ver una ligera inclinación ascendente: los días con sentimiento muy positivo (puntos en la derecha del eje x) tienden a tener retornos algo por encima de cero en el eje y. La recta de regresión ajustada tiene pendiente positiva β pequeña pero significativa. El **R^2** de esa regresión es bajo (~0.02), reflejando que la variabilidad de retornos apenas es explicada por el sentimiento diario. No obstante, la **pendiente** β positiva (p.ej. β = 0.002, que implica que un aumento de 1 punto en sentimiento (escala -1 a 1) se asocia con 0.2% de aumento en retorno diario) es estadísticamente significativo (intervalo de confianza excluye 0). Para TextBlob, pendientes similares emergen pero de menor magnitud.

También se crean **heatmaps de correlación** consolidando las correlaciones entre cada par de (método de sentimiento, tipo de retorno).

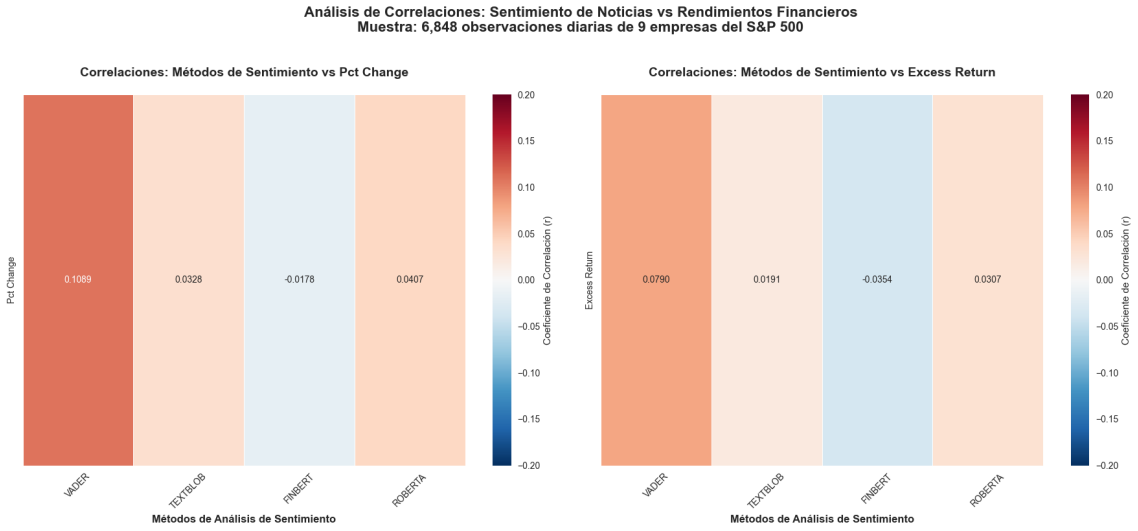


Tabla 6. Matriz de Correlaciones entre Métodos de Análisis de Sentimiento y Rendimientos Financieros.

Esto reafirma visualmente que VADER tenía los valores más altos en la intersección con retornos, seguido de TextBlob, luego RoBERTa y FinBERT al final (algunos celdas de FinBERT incluso en color indicando correlación negativa leve). Del lado de las variables de retorno, las correlaciones con retornos absolutos son ligeramente mayores que con retornos excesivos, lo cual aparece como celdas algo más intensas en la columna de “Return” vs “Excess Return”.

Heterogeneidad por empresa: Un hallazgo importante es que la **sensibilidad al sentimiento varía por empresa**.

Empresas de alto perfil mediático (como Tesla, que tiene mucha cobertura de noticias y comentarios) muestran correlaciones sentimiento-retorno más altas. Por ejemplo, Tesla tuvo r≈0.15 con VADER, indicando mayor impacto de las noticias. En cambio, una empresa como UnitedHealth tiene correlaciones más bajas, cerca de la insignificancia con algunos métodos, lo que siguiere que sus movimientos de precio responden más a indicadores tradicionales o noticias sectoriales difíciles de captar con análisis de texto simple.

Otra observación: VADER aporta resultados positivos consistentes casi en todas las empresas (ninguna tiene correlación VADER-retorno negativa), mientras que FinBERT en empresas fuera del sector financiero no parece tener ventajas. Es posible que FinBERT brille más con textos como transcripciones de resultados o documentos 10-K, más que con noticias de titulares. RoBERTa es relativamente consistente, pero sin destacar, salvo quizá en empresas con lenguaje más genérico en noticias.

Resumen de hallazgos cuantitativos: En resumen, en el análisis empírico encontramos evidencias de que existe una relación estadística entre el sentimiento de noticias y el desempeño diario de las acciones, aunque la magnitud es moderada. El mercado parece *"escuchar"* las noticias y reaccionar en consecuencia rápidamente, pero la proporción explicada de los movimientos de precios por sentimiento es pequeña. Esto no invalida su importancia: incluso pequeñas mejoras en predicción pueden ser valiosas en trading de alta frecuencia o estrategias con muchos datos (donde una pequeña ventaja puede ser explotada). Los resultados se alinean con la noción de que los mercados son bastante eficientes (incorporan la información rápida), pero también con la idea de las **finanzas conductuales** de que las noticias y el tono emocional pueden causar ligeros desvíos o reacciones que un inversor astuto podría aprovechar.

4. Conclusiones y Líneas Futuras

En esta sección final se presentan las conclusiones clave del estudio y se discuten posibles extensiones a futuro, así como las limitaciones enfrentadas.

4.1 Aprendizajes reales

El análisis empírico aporta evidencias de que existe una relación cuantificable entre el sentimiento presente en las noticias financieras y la evolución de los precios de las acciones. En concreto, podemos **confirmar** que noticias con tono positivo tienden a asociarse con rendimientos ligeramente superiores de las acciones en el corto plazo, mientras que noticias negativas suelen acompañar caídas o menores rendimientos ese día. Esto concuerda con la intuición y con estudios previos en finanzas que documentan impactos de las noticias en los mercados (p.ej., Tetlock 2007 encontró efectos similares en índices bursátiles). Sin embargo, también aprendimos que la magnitud de estos efectos es **reducida** en términos prácticos: el sentimiento explica solo una pequeña parte de la variación diaria de precios. Esto sugiere que, si bien las noticias importan, los mercados están dominados por muchos otros factores y posiblemente los precios ya descuenta gran parte de la información pública de manera eficiente.

Se observa que métodos sencillos de NLP como VADER pueden ser sorprendentemente **competitivos** capturando el sentimiento útil para mercados, a veces superando modelos más complejos en esta tarea específica. Esto puede deberse a que las noticias financieras, especialmente titulares, suelen usar palabras con connotación clara (léxico relativamente sencillo de polaridad). Por otro lado, modelos especializados como FinBERT, aunque mejores en clasificación de oraciones individualmente, no se traducen en un mayor poder predictivo lineal sobre retornos en nuestra implementación; esto destaca que el "mejor" algoritmo de NLP no siempre produce la "mejor" señal

financiera si esta no está bien calibrada al objetivo de predicción. También vemos que la importancia del **timing**: los efectos del sentimiento son más fuertes el mismo día, disipándose rápidamente, lo que sugiere que cualquier estrategia basada en ello debe ser de alta frecuencia o de muy corto plazo para capturar valor. Por último, el análisis por empresa nos enseña que la sensibilidad a las noticias varía – inversores en empresas de tecnología y crecimiento parecen reaccionar más a las noticias (posiblemente por mayor presencia en medios y redes), mientras que en empresas de sectores más estables el ruido de noticias tiene menos influencia inmediata.

4.2 Implementaciones pendientes

Si bien este TFG avanza en el análisis del vínculo entre noticias y precios, varios de los puntos propuestos quedan como líneas futuras de trabajo:

- *Modelos predictivos de series temporales:* Una extensión natural es incorporar los modelos de pronóstico cuantitativo (ARIMA, GARCH, LSTM, Prophet) para predecir la evolución de los precios y volatilidades, y combinarlos con las señales de sentimiento. Por ejemplo, se podría construir un modelo híbrido donde las predicciones de retornos de un LSTM se ajusten utilizando variables exógenas de sentimiento. Otra alternativa sería probar si las predicciones de Prophet (capturando tendencia estacional) mejoran al incluir un componente adicional informado por el sentimiento agregado de noticias recientes. Implementar y comparar estos modelos sería un paso siguiente valioso. Algo que requiere un riguroso procedimiento de *backtesting* con datos fuera de muestra para validar la efectividad predictiva.
- *Extracción de tópicos de noticias:* En lugar de solo cuantificar el tono de las noticias, una mejora es extraer los principales **tópicos temáticos** de ese flujo de noticias e incorporarlos al análisis. Por ejemplo, mediante LDA o NMF, podríamos detectar si en un cierto período las noticias de un sector giran mayormente en torno a "riesgo regulatorio" o "innovación tecnológica", etc., y estudiar cómo la presencia de ciertos tópicos correlaciona con comportamientos de mercado. La hipótesis podría ser que algunos tópicos (p. ej., muchos artículos sobre *"fraude contable"*) anticipen caídas, mientras otros (p. ej., *"nuevos*

productos exitosos") anticipen subidas. Implementar esto implicaría aplicar topic modeling a nuestro corpus de noticias y generar series temporales de intensidad de tópico, luego repetir análisis de correlación/predictivo con esas series. También enriquecería el estudio, pues ayudaría a interpretar *qué* tipo de noticias mueven más el mercado, no solo si son positivas o negativas.

- *Optimización dinámica de la cartera:* Finalmente, otro paso pendiente es utilizar las salidas de los componentes anteriores (predicciones de precio de modelos temporales, indicadores de sentimiento y eventos) dentro de un esquema de **asignación de activos dinámico**. Por ejemplo, se podría formular un modelo de optimización que cada semana ajuste pesos de la cartera de acciones en función de: (a) las previsiones de retorno para esas acciones (obtenidas de ARIMA/LSTM/Prophet), y (b) alguna señal derivada de las noticias (por ejemplo, evitar acciones con sentimiento muy negativo reciente, y darle un peso mayor a las de sentimiento positivo). El criterio a optimizar podría ser maximizar el ratio de Sharpe esperado de la cartera, o maximizar retorno esperado con limitaciones de riesgo, etc., incorporando penalizaciones por cambios de pesos (comisiones). El objetivo final: probar si efectivamente una estrategia de **gestión activa informada por datos** supera a una estrategia pasiva o a una basada solo en uno de los componentes. Implementarla implicaría simular un *backtesting* en histórico, reequilibrando la cartera periódicamente según las señales.

4.3 Limitaciones del estudio

Es importante reconocer las limitaciones que pueden afectar los resultados obtenidos:

- *Alcance de los datos:* El análisis se limita a 10 empresas grandes del S&P 500 y a noticias principalmente en inglés de fuentes financieras reconocidas. Esto significa que los hallazgos pueden no aplicar igual para empresas más pequeñas o para mercados fuera de EE.UU., donde la cobertura de noticias es distinta. Empresas de menor capitalización podrían tener más ineficiencias o menos seguimiento de noticias, lo que quizá cambiaría la relación. Asimismo, no se incorporan otras fuentes como *tweets* o foros (e.g., Reddit) donde inversores

discuten; dichas fuentes podrían incrementar la señal de "sentimiento de masa" que impactó mercados en episodios como GameStop en 2021.

- *Representatividad de las noticias:* Se asume que las noticias recopiladas constituyen una muestra relevante del flujo informativo. Sin embargo, siempre existe sesgo de selección: ¿qué pasa con noticias no capturadas o informaciones que los inversores tienen de otras fuentes? Además, muchas noticias reportan hechos después de que ocurran (ej: "la acción X cayó 5% por …"), por lo que parte del sentimiento medido puede ser más una consecuencia que una causa del movimiento.
- *Modelos de sentimiento simplificados:* Aunque usamos modelos avanzados como FinBERT, la forma en que agregamos el sentimiento (promediando diario, etc.) es bastante simple. Se pueden perder matices, por ejemplo, diferencias entre noticias de la mañana y la tarde, o impacto no lineal (quizá una noticia *muy* negativa tiene efecto desproporcionado vs. una moderadamente positiva). Tampoco diferenciamos por importancia de la noticia; en la práctica, un artículo de primera plana en WSJ pesa más que un blog menor, pero en nuestro dataset podrían haberse tratado por igual.
- *Supuesto de linearidad y correlación:* El análisis se centra en correlaciones de Pearson y regresiones lineales univariadas. Es posible que la relación entre sentimiento y retornos no sea puramente lineal. Por ejemplo, tal vez solo cuando el sentimiento cruza cierto umbral extremo el efecto en precios es notable (relación en forma de U). O puede haber interacciones: sentimiento negativo concurrente con alta volatilidad de mercado podría producir caídas mayores. No exploramos modelos más complejos (no lineales o multivariables) que podrían captar mejor la relación.
- *Causalidad vs. correlación:* Aunque hablamos de reacción del mercado a noticias, metodológicamente solo establecimos correlaciones y ligeras evidencias temporales. No se prueba estrictamente causalidad. Es posible que tanto la noticia como el movimiento de precio sean efecto de una causa subyacente no modelada. Un análisis más riguroso podría requerir metodologías de estudio de evento clásico (alinear múltiples episodios similares para promediar su efecto) o análisis causal estructural. Los resultados deberían interpretarse como asociaciones que *sugieren* una influencia informativa de las

noticias, pero no prueban de manera concluyente una relación causal directa en todos los casos.

- *Limitaciones computacionales:* Por simplicidad y eficiencia computacional, no afinamos hiperparámetros de los modelos de NLP ni se ha hecho validación cruzada extensiva. Un mejor rendimiento de FinBERT o RoBERTa podría alcanzarse con *fine-tuning* adicional en nuestro corpus específico. Del mismo modo, el análisis de retardos se hace hasta lag 30 de manera un poco ad hoc; técnicas más sofisticadas (como funciones de transferencia en modelos ARIMAX, o Granger causality tests) podrían aportar mayor rigor a la detección de efectos.
- *No inclusión de variables de control:* Sería ideal en un futuro incorporar controles adicionales en las regresiones, por ejemplo, momentum previo de la acción, sorpresas de anuncios de ganancias, etc., para aislar mejor el efecto puro de las noticias. En este estudio exploratorio no se ha realizado, lo que deja posibilidad de que alguna correlación observada se explique por un tercero factor correlacionado (por ejemplo, en días de mercado muy alcista casi todas las noticias son positivas, así VADER correlaciona con retornos pero el factor común fue el mercado alcista general).

Hay que tener en cuenta estas limitaciones de cara a los resultados. Sin embargo, la metodología y las herramientas desarrolladas proporcionan un marco reproducible y ampliable. Investigaciones posteriores podrían **mitigar** algunas limitaciones al: ampliar la cartera de activos (otros índices, criptoactivos donde las redes sociales pesan mucho), refinar la medida de sentimiento (ej. ponderar noticias por relevancia, usar análisis aspecto-basado para ver sentimiento sobre diferentes aspectos de la empresa), probar relaciones no lineales (modelos de árbol, Random Forest que incluyan sentimiento como predictor junto a otros), y experimentar con distintos horizontes (¿influye el sentimiento de noticias publicado a mitad de sesión en la última hora de trading, por ejemplo?).

En conclusión, el estudio aporta evidencia de la utilidad modesta pero real del análisis textual en finanzas, al tiempo que sienta las bases para una integración más compleja de modelos que podría traducirse en estrategias de inversión dinámicas. Creemos que a medida que las técnicas de **Big Data** y **AI** sigan introduciendose las finanzas, la capacidad de combinar datos tradicionales de mercados con información alternativa (textos, redes, etc.) será una fuente importante de ventaja competitiva, siempre y cuando se apliquen con un entendimiento sólido de sus limitaciones y un riguroso proceso de validación.

5. Bibliografía

Araci, D. (2019). *FinBERT: Financial sentiment analysis with pre-trained language models*. arXiv. https://arxiv.org/abs/1908.10063

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.

Black, F., & Litterman, R. (1992). Global portfolio optimization. *Financial Analysts Journal, 48*(5), 28–43. https://doi.org/10.2469/faj.v48.n5.28

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics, 31*(3), 307–327. https://doi.org/10.1016/0304-4076(86)90063-1

Engle, R. F. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation. *Econometrica, 50*(4), 987–1007. https://doi.org/10.2307/1912773

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance, 25*(2), 383–417. https://doi.org/10.2307/2325486

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hutto, C. J., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the Eighth International Conference on Weblogs and Social Media (ICWSM-14)* (pp. 216–225). AAAI Press.

Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance, 66*(1), 35–65. https://doi.org/10.1111/j.1540-6261.2010.01625.x

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature, 401*, 788–791. https://doi.org/10.1038/44565

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv*. https://arxiv.org/abs/1907.11692

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance, 7*(1), 77–91. https://doi.org/10.2307/2975974

Sharpe, W. F. (1966). Mutual fund performance. *The Journal of Business, 39*(1), 119–138. https://doi.org/10.1086/294846

Sunki, A., SatyaKumar, C., et al. (2024). *Time series forecasting of stock market using ARIMA, LSTM and FB Prophet. MATEC Web of Conferences, 392*, 01163.

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance, 62*(3), 1139–1168. https://doi.org/10.1111/j.1540-6261.2007.01232.x

6. Anexo

Importaciones:

```
# Importaciones principales
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import warnings
warnings.filterwarnings('ignore')

# Análisis de sentimiento
from nltk.sentiment import SentimentIntensityAnalyzer
import nltk
from textblob import TextBlob
from transformers import pipeline

# Utilidades
from datetime import datetime, timedelta
import time
from scipy import stats
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error
```

```
# Configuración de visualización
plt.style.use('seaborn-v0_8')
sns.set_palette('husl')
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
```

Configuración de modelos:

```
# Descargar recursos necesarios para NLTK
try:
    nltk.data.find('vader_lexicon')
except LookupError:
    nltk.download('vader_lexicon')

print('Inicializando modelos de análisis de sentimiento...')

# 1. VADER (Valence Aware Dictionary and sEntiment Reasoner)
sia_vader = SentimentIntensityAnalyzer()

# 2. FinBERT (Financial BERT) - modelo especializado en finanzas
try:
    finbert = pipeline('sentiment-analysis',
                       model='ProsusAI/finbert',
                       return_all_scores=True)
    print('FinBERT cargado correctamente')
except Exception as e:
    print(f'Error cargando FinBERT: {e}')
    finbert = None

# 3. BERT general para sentimientos
try:
    bert_sentiment = pipeline('sentiment-analysis',
                              model='cardiffnlp/twitter-roberta-base-sentiment-latest',
                              return_all_scores=True)
    print('RoBERTa Sentiment cargado correctamente')
except Exception as e:
    print(f'Error cargando RoBERTa: {e}')
    bert_sentiment = None

print('Modelos de sentimiento inicializados')
```

Funciones (Análisis de Sentimiento):

```
def analyze_sentiment_vader(text):
    """Análisis de sentimiento usando VADER"""
    if not isinstance(text, str) or not text.strip():
        return 0.0
    return sia_vader.polarity_scores(text)['compound']

def analyze_sentiment_textblob(text):
    """Análisis de sentimiento usando TextBlob"""
    if not isinstance(text, str) or not text.strip():
        return 0.0
    blob = TextBlob(text)
    return blob.sentiment.polarity

def analyze_sentiment_finbert(text):
    """Análisis de sentimiento usando FinBERT"""
    if not isinstance(text, str) or not text.strip() or finbert is None:
        return 0.0
    try:
        # Truncar texto si es muy largo
        text = text[:512]
        result = finbert(text)[0]

        # Convertir a escala -1 a 1
        sentiment_score = 0.0
        for item in result:
            if item['label'] == 'positive':
                sentiment_score += item['score']
            elif item['label'] == 'negative':
                sentiment_score -= item['score']
        return sentiment_score
    except Exception as e:
        print(f'Error en FinBERT: {e}')
        return 0.0

def analyze_sentiment_roberta(text):
    """Análisis de sentimiento usando RoBERTa"""
    if not isinstance(text, str) or not text.strip() or bert_sentiment is None:
        return 0.0
    try:
        # Truncar texto si es muy largo
        text = text[:512]
        result = bert_sentiment(text)[0]
```

```
        # Convertir a escala -1 a 1
        sentiment_score = 0.0
        for item in result:
            if item['label'] in ['LABEL_2', 'positive']:
                sentiment_score += item['score']
            elif item['label'] in ['LABEL_0', 'negative']:
                sentiment_score -= item['score']
        return sentiment_score
    except Exception as e:
        print(f'Error en RoBERTa: {e}')
        return 0.0
```

Funciones (Procesamiento de datos):

```
def percent_change_1day(date, data):
    """Calcula el cambio porcentual considerando días hábiles"""
    data = data.sort_values(by='Date')

    last_trading_day = data[data['Date'] < date]
    current_day = data[data['Date'] == date]

    if not last_trading_day.empty and not current_day.empty:
        previous_value = last_trading_day['Close'].iloc[-1]
        current_value = current_day['Close'].iloc[0]

        # Asegurar que tenemos valores numéricos
        if hasattr(previous_value, 'item'):
            previous_value = previous_value.item()
        if hasattr(current_value, 'item'):
            current_value = current_value.item()

        pct_change = ((current_value - previous_value) / previous_value) * 100
        return pct_change
    else:
        return None

def calculate_excess_returns(ticker_data, sp500_data):
    """Calcula los rendimientos excesivos vs S&P 500"""
    # Limpiar columnas de MultiIndex si existen
    if hasattr(ticker_data.columns, 'levels'):
        ticker_data.columns = ticker_data.columns.droplevel(1)
    if hasattr(sp500_data.columns, 'levels'):
```

```
        sp500_data.columns = sp500_data.columns.droplevel(1)

    # Asegurar que las columnas tienen nombres simples
    ticker_data = ticker_data.rename(columns={ticker_data.columns[1]: 'Close'})
    sp500_data = sp500_data.rename(columns={sp500_data.columns[1]: 'Close_sp500'})

    # Merge de datos
    combined_data = pd.merge(ticker_data, sp500_data, on='Date', how='inner')

    # Calcular rendimientos
    combined_data['ticker_return'] = combined_data['Close'].pct_change() * 100
    combined_data['sp500_return'] = combined_data['Close_sp500'].pct_change() * 100
    combined_data['excess_return'] = combined_data['ticker_return'] - combined_data['sp500_return']

    return combined_data

def apply_sentiment_methods(news_data, text_column='Article_title'):
    """Aplica todos los métodos de sentimiento al DataFrame"""
    print(f'Aplicando análisis de sentimiento a {len(news_data)} registros...')

    # VADER
    news_data['sentiment_vader'] = news_data[text_column].apply(analyze_sentiment_vader)
    print('VADER completado')

    # TextBlob
    news_data['sentiment_textblob'] = news_data[text_column].apply(analyze_sentiment_textblob)
    print('TextBlob completado')

    # FinBERT (solo para muestras pequeñas debido a la velocidad)
    if len(news_data) <= 1000:
        news_data['sentiment_finbert'] = news_data[text_column].apply(analyze_sentiment_finbert)
        print('FinBERT completado')
    else:
        # Para datasets grandes, aplicar a una muestra
        sample_size = min(1000, len(news_data))
        sample_indices = np.random.choice(news_data.index, sample_size, replace=False)
        news_data['sentiment_finbert'] = 0.0
        news_data.loc[sample_indices, 'sentiment_finbert'] = news_data.loc[sample_indices,
text_column].apply(analyze_sentiment_finbert)
        print(f'FinBERT completado (muestra de {sample_size} registros)')

    # RoBERTa
    if len(news_data) <= 1000:
        news_data['sentiment_roberta'] = news_data[text_column].apply(analyze_sentiment_roberta)
        print('RoBERTa completado')
```

```
    else:
        sample_size = min(1000, len(news_data))
        sample_indices = np.random.choice(news_data.index, sample_size, replace=False)
        news_data['sentiment_roberta'] = 0.0
        news_data.loc[sample_indices, 'sentiment_roberta'] = news_data.loc[sample_indices,
text_column].apply(analyze_sentiment_roberta)
        print(f'RoBERTa completado (muestra de {sample_size} registros)')

    return news_data
```

Carga y pre-procesado de datos:

```
# Cargar datos de noticias
print('Cargando datos de noticias...')
df = pd.read_parquet('../dataset_noticias/datos_analisis.parquet')

# Empresas del S&P 500 a analizar
TICKERS = ['AAPL', 'MSFT', 'GOOGL', 'AMZN', 'TSLA', 'META', 'NVDA', 'JPM', 'UNH', 'V']

# Filtrar solo los tickers seleccionados
df = df[df['Stock_symbol'].isin(TICKERS)]

# Procesar fechas
df['Date'] = pd.to_datetime(df['Date'])
df['Date'] = df['Date'].dt.tz_localize(None)
df['Date'] = df['Date'].dt.date
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
df = df.dropna(subset=['Date'])

print(f'Datos cargados: {len(df)} registros para {len(TICKERS)} empresas')
print(f'Rango de fechas: {df["Date"].min()} - {df["Date"].max()}')
```

Análisis principal:

```
# DataFrame para almacenar todos los resultados
results_df = pd.DataFrame()

# Procesar cada ticker
for ticker in TICKERS:
    if df[df['Stock_symbol'] == ticker].empty:
```

```
    print(f"No hay datos para {ticker}. Saltando...")
    continue

print(f"\n{'='*50}")
print(f"Procesando {ticker}...")
print(f"{'='*50}")

start_time = time.time()

# Filtrar datos de noticias para este ticker
news_data = df[df['Stock_symbol'] == ticker].copy()

# Obtener rango de fechas
START_DATE = news_data['Date'].min().strftime('%Y-%m-%d')
END_DATE = news_data['Date'].max().strftime('%Y-%m-%d')

print(f"Rango de fechas: {START_DATE} a {END_DATE}")
print(f"Número de noticias: {len(news_data)}")

# Descargar datos financieros
print('Descargando datos financieros...')
ticker_data = yf.download(ticker, START_DATE, END_DATE)
sp500_data = yf.download('^GSPC', START_DATE, END_DATE)

# Procesar datos financieros
ticker_data.reset_index(inplace=True)
sp500_data.reset_index(inplace=True)

ticker_data = ticker_data[['Date', 'Close']]
sp500_data = sp500_data[['Date', 'Close']]

# Calcular rendimientos excesivos
combined_financial = calculate_excess_returns(ticker_data, sp500_data)

# Calcular cambios porcentuales para cada noticia
print('Calculando cambios de precios...')
news_data['pct_change'] = news_data['Date'].apply(
    lambda date: percent_change_1day(date, ticker_data)
)

# Eliminar noticias sin datos de precios
news_data = news_data.dropna(subset=['pct_change'])

# Aplicar análisis de sentimiento
print('Analizando sentimientos...')
```

```
    news_data = apply_sentiment_methods(news_data, 'Article_title')

    # También analizar textrank_summary si existe
    if 'Textrank_summary' in news_data.columns:
        news_data['sentiment_textrank_vader'] = news_data['Textrank_summary'].apply(analyze_sentiment_vader)

    # Agregar datos diarios
    print('Agregando datos diarios...')
    daily_sentiment = news_data.groupby('Date').agg({
        'sentiment_vader': 'mean',
        'sentiment_textblob': 'mean',
        'sentiment_finbert': 'mean',
        'sentiment_roberta': 'mean',
        'sentiment_textrank_vader': 'mean' if 'sentiment_textrank_vader' in news_data.columns else lambda x: 0,
        'pct_change': 'mean',
        'Article_title': 'count'  # Contar número de noticias por día
    }).reset_index()

    # Renombrar columna de conteo
    daily_sentiment.rename(columns={'Article_title': 'news_count'}, inplace=True)

    # Añadir información del ticker
    daily_sentiment['ticker'] = ticker

    # Merge con datos de rendimientos excesivos
    daily_sentiment['Date'] = pd.to_datetime(daily_sentiment['Date'])
    combined_financial['Date'] = pd.to_datetime(combined_financial['Date'])

    daily_sentiment = pd.merge(
        daily_sentiment,
        combined_financial[['Date', 'excess_return', 'ticker_return', 'sp500_return']],
        on='Date',
        how='left'
    )

    # Concatenar resultados
    results_df = pd.concat([results_df, daily_sentiment], ignore_index=True)

    elapsed_time = time.time() - start_time
    print(f"{ticker} completado en {elapsed_time:.2f} segundos")

print(f"\nProcesamiento completado. Total de registros: {len(results_df)}")
```

Análisis de Lags (Retrasos temporales):

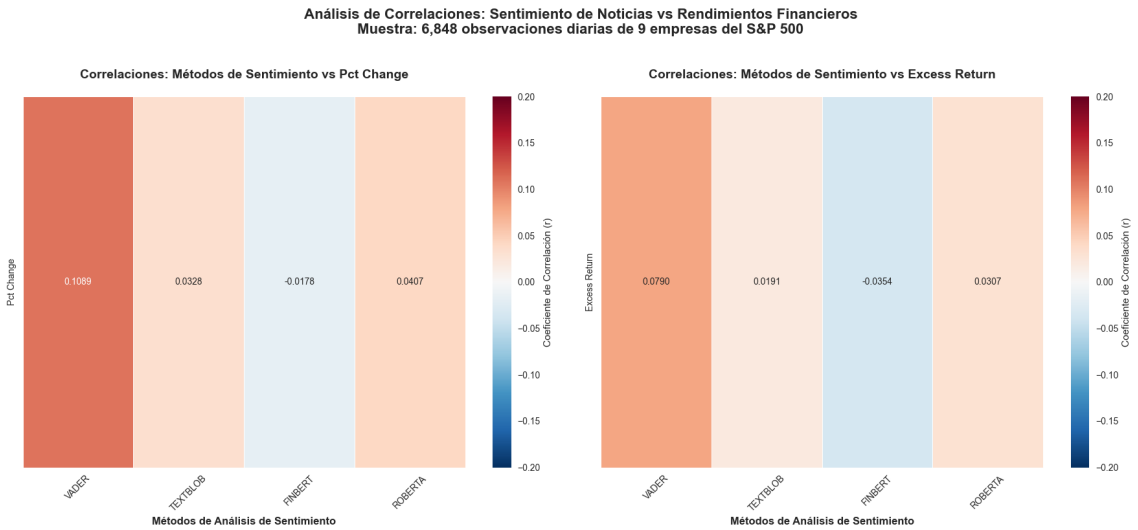
```
def calculate_lag_correlations(df, sentiment_col, target_col, max_lag=30):
    """Calcula correlaciones con diferentes lags"""
    correlations = []

    for lag in range(0, max_lag + 1):
        # Crear columna con lag
        df_lag = df.copy()
        df_lag[f'{target_col}_shifted'] = df_lag[target_col].shift(-lag)

        # Calcular correlación
        corr = df_lag[sentiment_col].corr(df_lag[f'{target_col}_shifted'])
        correlations.append({'lag': lag, 'correlation': corr})

    return pd.DataFrame(correlations)

# Calcular correlaciones con lags para cada método
lag_results = {}

# Agrupar datos por día para análisis temporal
daily_analysis = analysis_df.groupby('Date').agg({
    'sentiment_vader': 'mean',
    'sentiment_textblob': 'mean',
    'sentiment_finbert': 'mean',
    'sentiment_roberta': 'mean',
    'pct_change': 'mean',
    'excess_return': 'mean'
}).reset_index().sort_values('Date')

print("Calculando correlaciones con lags...")

for method in sentiment_methods:
    if method in daily_analysis.columns and daily_analysis[method].notna().sum() > 30:
        lag_results[method] = {}

        for target in target_variables:
            if target in daily_analysis.columns:
                lag_corr = calculate_lag_correlations(
                    daily_analysis, method, target, max_lag=30
                )
                lag_results[method][target] = lag_corr

                # Encontrar el lag con mayor correlación absoluta
```

```
            max_corr_idx = lag_corr['correlation'].abs().idxmax()
            max_corr = lag_corr.loc[max_corr_idx]

            print(f"{method} vs {target}: Máxima correlación {max_corr['correlation']:.4f} en lag {max_corr['lag']}")

print("\nAnálisis de lags completado")
```

Visualización de resultados (correlaciones):

```
# 1. Heatmap de correlaciones con análisis profesional
fig, axes = plt.subplots(1, 2, figsize=(18, 8))

for idx, target in enumerate(target_variables):
    corr_data = []
    methods = []

    for method in sentiment_methods:
        if method in correlations[target]:
            corr_data.append(correlations[target][method])
            methods.append(method.replace('sentiment_', '').upper())

    # Crear heatmap con configuración profesional
    corr_matrix = np.array(corr_data).reshape(1, -1)
    heatmap = sns.heatmap(corr_matrix,
                          annot=True,
                          fmt='.4f',
                          cmap='RdBu_r',
                          center=0,
                          xticklabels=methods,
                          yticklabels=[target.replace('_', ' ').title()],
                          ax=axes[idx],
                          cbar_kws={'label': 'Coeficiente de Correlación (r)'},
                          vmin=-0.2, vmax=0.2,  # Escala fija para comparación
                          linewidths=0.5)

    # Personalización del gráfico
    axes[idx].set_title(f'Correlaciones: Métodos de Sentimiento vs {target.replace("_", " ").title()}',
                        fontsize=14, fontweight='bold', pad=20)
    axes[idx].set_xlabel('Métodos de Análisis de Sentimiento', fontsize=12, fontweight='bold')
    axes[idx].tick_params(axis='x', rotation=45)

# Configuración general del gráfico
plt.suptitle('Análisis de Correlaciones: Sentimiento de Noticias vs Rendimientos Financieros\n' +
```

```
        f'Muestra: {len(analysis_df):,} observaciones diarias de {len(TICKERS)} empresas del S&P 500',
        fontsize=16, fontweight='bold', y=1.02)

plt.tight_layout()
plt.subplots_adjust(top=0.85)

# Guardar gráfico
plt.savefig('correlaciones_heatmap.png', dpi=300, bbox_inches='tight')
plt.show()
```



Visualización de resultados (correlaciones con lags):

```
# 2. Gráfico de correlaciones por lags mejorado
fig = make_subplots(
    rows=2, cols=2,
    subplot_titles=('VADER vs Cambio Porcentual de Precios',
                    'TextBlob vs Cambio Porcentual de Precios',
                    'VADER vs Rendimientos Excesivos',
                    'TextBlob vs Rendimientos Excesivos'),
    vertical_spacing=0.12,
    horizontal_spacing=0.1
)

colors = ['#2E86C1', '#E74C3C', '#28B463', '#F39C12']
row_col_mapping = [(1,1), (1,2), (2,1), (2,2)]

plot_idx = 0
max_correlations = []

for method in ['sentiment_vader', 'sentiment_textblob']:
```

```
    for target in target_variables:
        if method in lag_results and target in lag_results[method]:
            lag_data = lag_results[method][target]
            row, col = row_col_mapping[plot_idx]

            # Encontrar máxima correlación absoluta
            max_idx = lag_data['correlation'].abs().idxmax()
            max_corr = lag_data.loc[max_idx]
            max_correlations.append({
                'method': method,
                'target': target,
                'max_corr': max_corr['correlation'],
                'optimal_lag': max_corr['lag']
            })

            # Línea principal
            fig.add_trace(
                go.Scatter(
                    x=lag_data['lag'],
                    y=lag_data['correlation'],
                    mode='lines+markers',
                    name=f'{method.replace("sentiment_", "")} vs {target}',
                    line=dict(color=colors[plot_idx], width=3),
                    marker=dict(size=6, symbol='circle'),
                    hovertemplate='Lag: %{x} días<br>Correlación: %{y:.4f}<extra></extra>'
                ),
                row=row, col=col
            )

            # Marcar punto de máxima correlación
            fig.add_trace(
                go.Scatter(
                    x=[max_corr['lag']],
                    y=[max_corr['correlation']],
                    mode='markers',
                    marker=dict(size=12, color='red', symbol='star'),
                    name=f'Máx: {max_corr["correlation"]:.4f} (lag {max_corr["lag"]}d)',
                    hovertemplate=f'Correlación máxima: {max_corr["correlation"]:.4f}<br>Lag óptimo: {max_corr["lag"]}
días<extra></extra>'
                ),
                row=row, col=col
            )

            # Línea de referencia en 0
            fig.add_hline(y=0, line_dash="dash", line_color="gray", opacity=0.5, row=row, col=col)
```

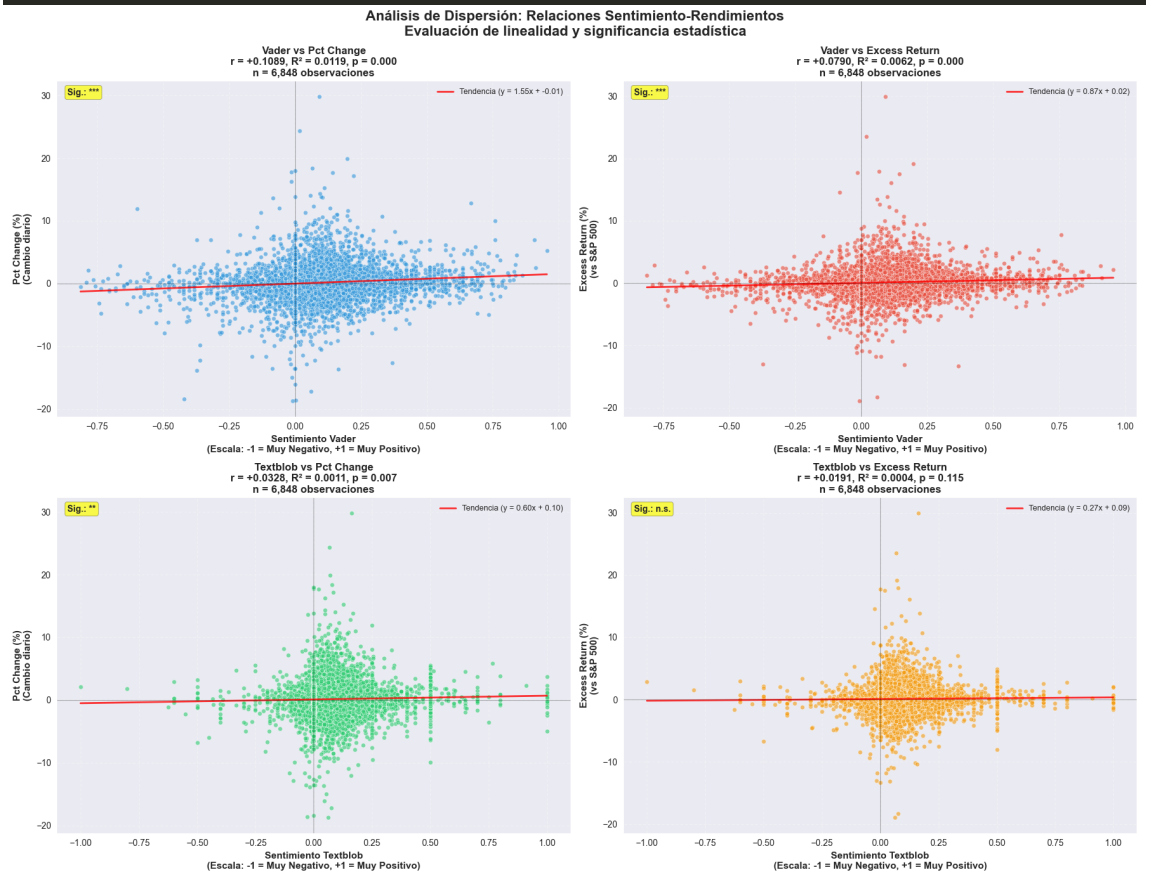
```
        plot_idx += 1

# Configuración del layout
fig.update_layout(
    height=900,
    title_text='Análisis Temporal: Correlaciones por Lags (0-30 días)<br>' +
               '<sub>Evaluación del efecto predictivo del sentimiento sobre rendimientos futuros</sub>',
    title_x=0.5,
    title_font=dict(size=16),
    showlegend=False,
    template='plotly_white'
)

fig.update_xaxes(title_text='Lag (días)', title_font=dict(size=12))
fig.update_yaxes(title_text='Coeficiente de Correlación (r)', title_font=dict(size=12))

# Guardar y mostrar
fig.write_html('correlaciones_lags.html')
fig.show()
```

Visualización de resultados (Dispersión):

```
# 3. Gráficos de dispersión mejorados con análisis estadístico
fig, axes = plt.subplots(2, 2, figsize=(18, 14))

methods_to_plot = ['sentiment_vader', 'sentiment_textblob']
targets_to_plot = ['pct_change', 'excess_return']

# Configuración de colores y estilos
colors_scatter = ['#3498DB', '#E74C3C', '#2ECC71', '#F39C12']
regression_stats = []

for i, method in enumerate(methods_to_plot):
    for j, target in enumerate(targets_to_plot):
        ax = axes[i, j]

        # Filtrar datos válidos
        plot_data = analysis_df[[method, target]].dropna()

        if len(plot_data) > 0:
            # Scatter plot principal
            scatter = ax.scatter(plot_data[method], plot_data[target],
                                 alpha=0.6, s=25, color=colors_scatter[i*2 + j],
                                 edgecolors='white', linewidth=0.5)

            # Línea de regresión
            z = np.polyfit(plot_data[method], plot_data[target], 1)
            p = np.poly1d(z)
            x_reg = np.linspace(plot_data[method].min(), plot_data[method].max(), 100)
            ax.plot(x_reg, p(x_reg), 'r-', alpha=0.8, linewidth=2, label=f'Tendencia (y = {z[0]:.2f}x + {z[1]:.2f})')

            # Estadísticas
            corr = plot_data[method].corr(plot_data[target])
            slope, intercept, r_value, p_value, std_err = stats.linregress(plot_data[method], plot_data[target])

            regression_stats.append({
                'method': method,
                'target': target,
                'correlation': corr,
                'slope': slope,
                'p_value': p_value,
                'r_squared': r_value**2,
                'n_obs': len(plot_data)
            })
```

```
        # Configuración del gráfico
        method_name = method.replace('sentiment_', '').title()
        target_name = target.replace('_', ' ').title()

        ax.set_title(f'{method_name} vs {target_name}\n' +
                     f'r = {corr:+.4f}, R² = {r_value**2:.4f}, p = {p_value:.3f}\n' +
                     f'n = {len(plot_data):,} observaciones',
                     fontsize=12, fontweight='bold')

        ax.set_xlabel(f'Sentimiento {method_name}\n(Escala: -1 = Muy Negativo, +1 = Muy Positivo)',
                      fontsize=11, fontweight='bold')
        ax.set_ylabel(f'{target_name} (%)\n({"Cambio diario" if "pct" in target else "vs S&P 500"})',
                      fontsize=11, fontweight='bold')

        # Grid y estilo
        ax.grid(True, alpha=0.3, linestyle='--')
        ax.axhline(y=0, color='black', linestyle='-', alpha=0.3, linewidth=0.8)
        ax.axvline(x=0, color='black', linestyle='-', alpha=0.3, linewidth=0.8)

        # Leyenda
        ax.legend(loc='upper right', fontsize=9)

        # Añadir anotación de significancia estadística
        if p_value < 0.001:
            significance = '***'
        elif p_value < 0.01:
            significance = '**'
        elif p_value < 0.05:
            significance = '*'
        else:
            significance = 'n.s.'

        ax.text(0.02, 0.98, f'Sig.: {significance}', transform=ax.transAxes,
                fontsize=10, fontweight='bold', verticalalignment='top',
                bbox=dict(boxstyle='round,pad=0.3', facecolor='yellow', alpha=0.7))

plt.suptitle('Análisis de Dispersión: Relaciones Sentimiento-Rendimientos\n' +
             'Evaluación de linealidad y significancia estadística',
             fontsize=16, fontweight='bold', y=0.98)

plt.tight_layout()
plt.subplots_adjust(top=0.90)

# Guardar gráfico
```

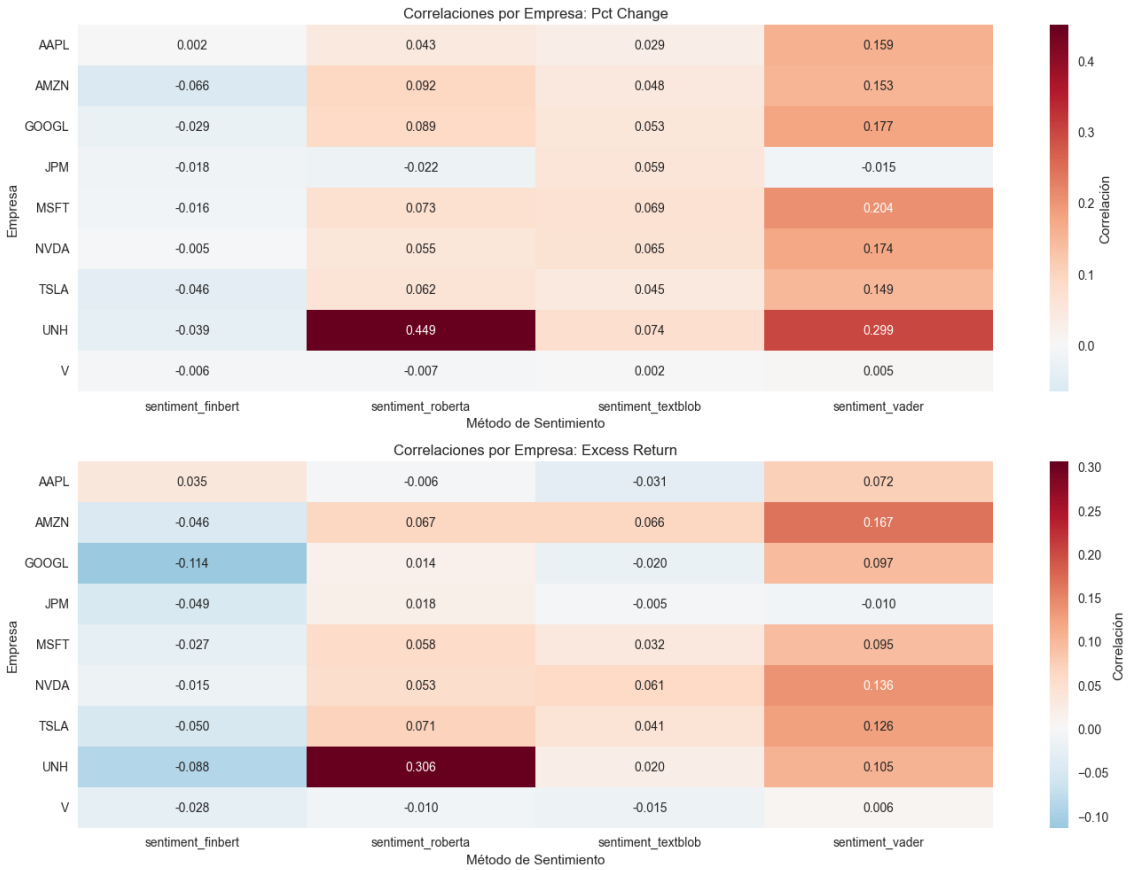
```
plt.savefig('scatter_analysis.png', dpi=300, bbox_inches='tight')
plt.show()
```



Visualización de resultados (Por empresas):

```
# Análisis de correlaciones por empresa
ticker_correlations = []

for ticker in TICKERS:
    ticker_data = analysis_df[analysis_df['ticker'] == ticker]

    if len(ticker_data) > 10:  # Mínimo de datos para análisis
        for method in sentiment_methods:
            if method in ticker_data.columns:
                for target in target_variables:
                    corr = ticker_data[method].corr(ticker_data[target])
                    ticker_correlations.append({
                        'ticker': ticker,
                        'method': method,
                        'target': target,
                        'correlation': corr,
```

```
                    'n_observations': len(ticker_data)
                })

ticker_corr_df = pd.DataFrame(ticker_correlations)

# Visualizar correlaciones por empresa
if not ticker_corr_df.empty:
    fig, axes = plt.subplots(2, 1, figsize=(14, 10))

    for idx, target in enumerate(target_variables):
        target_data = ticker_corr_df[ticker_corr_df['target'] == target]

        if not target_data.empty:
            pivot_data = target_data.pivot(index='ticker', columns='method', values='correlation')

            sns.heatmap(pivot_data,
                        annot=True,
                        fmt='.3f',
                        cmap='RdBu_r',
                        center=0,
                        ax=axes[idx],
                        cbar_kws={'label': 'Correlación'})

            axes[idx].set_title(f'Correlaciones por Empresa: {target.replace("_", " ").title()}')
            axes[idx].set_xlabel('Método de Sentimiento')
            axes[idx].set_ylabel('Empresa')

    plt.tight_layout()
    plt.show()
```


Correlaciones por Empresa: Pct Change
AAPL 0.002 0.043 0.029 0.159
AMZN -0.066 0.092 0.048 0.153
GOOGL -0.029 0.089 0.053 0.177
JPM -0.018 -0.022 0.059 -0.015
MSFT -0.016 0.073 0.069 0.204
NVDA -0.005 0.055 0.065 0.174
TSLA -0.046 0.062 0.045 0.149
UNH -0.039 0.449 0.074 0.299
V -0.006 -0.007 0.002 0.005
Empresa
sentiment_finbert
sentiment_roberta
sentiment_textblob
sentiment_vader
Método de Sentimiento
Correlación
0.4
0.3
0.2
0.1
0.0
Correlaciones por Empresa: Excess Return
AAPL 0.035 -0.006 -0.031 0.072
AMZN -0.046 0.067 0.066 0.167
GOOGL -0.114 0.014 -0.020 0.097
JPM -0.049 0.018 -0.005 -0.010
MSFT -0.027 0.058 0.032 0.095
NVDA -0.015 0.053 0.061 0.136
TSLA -0.050 0.071 0.041 0.126
UNH -0.088 0.306 0.020 0.105
V -0.028 -0.010 -0.015 0.006
Empresa
sentiment_finbert
sentiment_roberta
sentiment_textblob
sentiment_vader
Método de Sentimiento
Correlación
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10