

Diseño de un Modelo Predictivo para el Concurso Hull Tactical Market Prediction (Kaggle)

Elias Sebastian Gill Quintana

Octubre 2025

1. Contexto del Problema

El presente trabajo tiene como objetivo analizar y diseñar una propuesta de solución al desafío *Hull Tactical Market Prediction*, organizado en la plataforma Kaggle. Este problema pertenece al ámbito de la predicción financiera y plantea el reto de anticipar los rendimientos diarios del índice S&P 500 bajo una restricción de volatilidad. A diferencia de otros ejercicios de predicción bursátil puramente teóricos, el desafío propone un contexto con implicaciones prácticas para estrategias reales de inversión.

El participante debe construir un modelo capaz de estimar el exceso de rendimiento del mercado, comparado con la tasa libre de riesgo, optimizando una métrica basada en una variante del ratio de Sharpe. La tarea incluye la generación de una estrategia de asignación diaria de capital en un rango permitido entre 0 y 2, que represente la proporción de fondos invertidos en el S&P 500. Este rango permite reflejar tanto posiciones conservadoras como estrategias apalancadas, siempre dentro de la restricción de volatilidad del 120 %.

El conjunto de datos suministrado está compuesto por información histórica de mercado que incluye múltiples categorías de variables: dinámicas de mercado, macroeconómicas, de tasas de interés, valoración, volatilidad, sentimiento, momentum y variables binarias. Los archivos de entrenamiento contienen además los rendimientos *forward* del mercado, la tasa libre de riesgo y el rendimiento excedente relativo al mercado, que constituyen las variables objetivo o de referencia para el entrenamiento del modelo.

2. Clasificación del Problema

El problema corresponde a una tarea de *predicción supervisada de series temporales* con salida continua. Desde la perspectiva de aprendizaje automático, puede clasificarse como un problema de regresión, ya que el objetivo consiste en predecir valores numéricos continuos que representen rendimientos futuros del mercado. No obstante, presenta características particulares del ámbito financiero que exigen respetar el orden temporal de los datos y evitar la fuga de información entre conjuntos de entrenamiento y validación.

La naturaleza temporal del conjunto implica que los métodos tradicionales de validación aleatoria no son apropiados, por lo que se emplearán técnicas de validación cronológica o *walk-forward*. Además, debido a la existencia de múltiples grupos de variables correlacionadas, el problema puede considerarse de alta dimensionalidad y con potencial presencia de multicolinealidad. Esto motiva la selección de modelos capaces de manejar correlaciones complejas, ruido y valores faltantes.

3. Motivación y Relevancia

La predicción de rendimientos financieros continúa siendo un tema central en el estudio de la eficiencia de los mercados. La hipótesis del mercado eficiente sostiene que los precios reflejan toda la información disponible, lo cual implicaría que no es posible obtener ganancias sistemáticas mediante modelos predictivos. Sin embargo, el avance de las técnicas de aprendizaje automático permite reconsiderar esa hipótesis en contextos donde los patrones no lineales o las interacciones entre múltiples variables podrían ofrecer señales útiles para la toma de decisiones.

Desarrollar un modelo capaz de mejorar el desempeño del índice S&P 500 bajo restricciones de volatilidad no solo representa un desafío técnico, sino también una oportunidad de explorar cómo los modelos de inteligencia artificial pueden contribuir a la gestión cuantitativa de portafolios. La relevancia del problema radica en la posibilidad de construir estrategias de inversión más eficientes en riesgo y rendimiento, aplicando técnicas de predicción avanzadas sobre datos públicos y privados del mercado.

4. Limitaciones Actuales en Modelos

Los modelos tradicionales de predicción de series financieras presentan varias limitaciones. En primer lugar, las técnicas lineales como la regresión múltiple o los modelos ARIMA tienden a suponer relaciones estáticas y lineales entre las variables, lo que no refleja la naturaleza dinámica y no lineal de los mercados financieros. En segundo lugar, muchos modelos ignoran la heterocedasticidad y los cambios estructurales que caracterizan a los datos financieros, lo que reduce su capacidad de generalización.

En el contexto de las competencias de Kaggle, el uso de arquitecturas más modernas, como redes neuronales recurrentes, transformadores o modelos híbridos que integren componentes de series temporales con aprendizaje profundo, se ha mostrado más prometedor. No obstante, estos modelos suelen requerir una cantidad considerable de datos y una cuidadosa selección de hiperparámetros para evitar sobreajuste. Además, la interpretabilidad de sus predicciones continúa siendo un desafío, lo cual es especialmente importante en entornos financieros donde las decisiones deben ser justificables.

Por otro lado, las restricciones impuestas por la competencia —como la ausencia de conexión a Internet y el límite de tiempo de ejecución— también condicionan la elección de modelos y técnicas de entrenamiento. Esto implica priorizar soluciones eficientes que logren un equilibrio entre precisión, estabilidad y tiempo de ejecución.

5. Análisis del Dataset

El análisis exploratorio del conjunto de datos reveló una estructura compleja y con desafíos típicos de series financieras. El dataset consta de 8,990 observaciones y 98 columnas, con una proporción significativa de valores ausentes en determinados grupos de variables. Tras una optimización de tipos, se logró una reducción de memoria de aproximadamente 52.9 %, manteniendo la integridad de la información. La variable objetivo principal, *excess_return*, se define como la diferencia entre los retornos futuros del mercado y la tasa libre de riesgo (*forward_returns - risk_free_rate*).

Los análisis estadísticos iniciales muestran que la distribución de esta variable no es normal. Pruebas como Jarque-Bera y D'Agostino arrojan p-valores muy inferiores a 0.05, confirmando la presencia de colas pesadas y asimetría. Este hallazgo invalida el uso de modelos lineales basados en supuestos gaussianos y orienta la selección hacia métodos robustos o transformaciones del objetivo. Además, se identificó un fenómeno de *volatility clustering*, caracterizado por períodos

alternados de alta y baja varianza. La autocorrelación de los retornos diarios es baja a lags cortos, lo que coincide con la hipótesis de eficiencia del mercado en horizontes de un día.

Respecto a los valores ausentes, las columnas con mayor proporción de nulos son E7 (77.52 %), V10 (67.29 %), S3 (63.77 %), M1 (61.70 %) y M14 (61.62 %). Se identificó un punto temporal a partir del cual más del 95 % de las variables están completas ($date_id = 5540$), lo que sugiere dos enfoques posibles: (a) emplear todo el historial aplicando imputaciones causales, o (b) restringir el análisis al periodo posterior para maximizar la calidad de datos, a costa de perder observaciones.

El estudio de correlaciones con el objetivo revela relaciones lineales débiles. Las mayores correlaciones absolutas no superan 0.07, lo cual sugiere que las señales predictivas podrían encontrarse en interacciones no lineales o combinaciones temporales de variables. Los mapas de correlación por grupos (prefijos D, E, I, M, P, S, V y MOM) muestran redundancia interna significativa, lo que justifica aplicar técnicas de reducción dimensional o selección de variables basadas en importancia de modelo.

Desde el punto de vista operativo, los resultados del análisis conducen a las siguientes conclusiones: los modelos lineales simples son inadecuados; la ingeniería de características debe incorporar estadísticas móviles, volatilidad histórica e interacciones entre grupos; y las estrategias de imputación deben ser temporales y sin fuga de información. Se recomienda emplear validación temporal estricta mediante *walk-forward validation* para asegurar la robustez fuera de muestra.

En resumen, el conjunto de datos presenta alta dimensionalidad, heterogeneidad temporal y baja correlación directa con el objetivo. La señal predictiva, aunque débil, puede ser aprovechada a través de modelos no lineales con manejo explícito de volatilidad y regularización adecuada. Estos hallazgos orientan la arquitectura del modelo, la estrategia de entrenamiento y las métricas de evaluación a emplear en etapas posteriores.

6. Modelo Propuesto

El modelo propuesto combina enfoques estadísticos y de aprendizaje automático mediante una arquitectura de *stacking*. Se busca capturar tanto las dependencias lineales y parciales de las variables predictoras como las no linealidades y efectos dinámicos presentes en series financieras.

La estructura general está compuesta por tres modelos base y un meta-aprendiz:

1. **ElasticNet**: utilizado para la selección de variables y la obtención de relaciones lineales parciales entre los predictores y los rendimientos diarios del S&P 500. Combina las penalizaciones L1 y L2 para controlar la complejidad y la multicolinealidad del conjunto de datos.

2. **LightGBM**: modelo basado en árboles de decisión que captura relaciones no lineales y permite una representación más flexible de los patrones en los datos. Su eficiencia computacional y capacidad para manejar interacciones complejas lo convierten en un componente esencial del ensamblado.

3. **ARIMA**: incorporado como modelo adicional para evaluar la capacidad de los métodos clásicos de series temporales en la predicción de rendimientos. Se integra en paralelo a los otros modelos dentro del esquema de stacking, aportando estructura temporal explícita.

4. **GARCH**: empleado como módulo de estimación de volatilidad condicional. Este componente no participa directamente en la predicción del retorno, sino que genera una proyección de la varianza esperada, utilizada para ajustar la asignación de capital y controlar el riesgo.

Las predicciones de los modelos base se combinan mediante un **meta-modelo Ridge**, que actúa como nivel superior de agregación. El Ridge se entrena sobre las salidas *out-of-fold* de los modelos base, lo que reduce el riesgo de sobreajuste y permite obtener una combinación óptima de las señales generadas.

La incorporación de GARCH permite adaptar el tamaño de posición o exposición del modelo según la volatilidad prevista, mejorando la estabilidad del rendimiento acumulado y reduciendo drawdowns. Este módulo se actualiza de forma periódica con ventana móvil para reflejar cambios recientes en la dinámica del mercado.

En síntesis, el modelo propuesto busca un equilibrio entre interpretabilidad, capacidad predictiva y control de riesgo. ElasticNet aporta transparencia, LightGBM flexibilidad, ARIMA estructura temporal y GARCH control de volatilidad. El meta-aprendiz Ridge asegura una integración coherente de todas las fuentes de señal bajo un marco robusto de validación temporal.

7. Metodología del Entrenamiento

El entrenamiento del sistema propuesto se estructura bajo un enfoque cronológico estricto, evitando toda fuga de información y priorizando reproducibilidad. Se busca un balance entre rendimiento, interpretabilidad y estabilidad temporal.

7.1. Estructura temporal

El entrenamiento se realiza mediante un esquema de validación *walk-forward*. En cada iteración se entrena el modelo con los datos previos y se valida sobre el siguiente bloque temporal. Este método asegura que toda decisión se base únicamente en información disponible en el pasado. Se utilizarán dos configuraciones: una con el conjunto completo de datos imputados causalmente y otra con los períodos más recientes para evaluar la robustez del modelo frente a diferentes horizontes históricos.

7.2. Preprocesamiento

El flujo de preprocesamiento incluye la detección y tratamiento de valores faltantes, imputación causal limitada por ventana temporal, estandarización de variables y generación de características derivadas. Ninguna operación puede usar datos posteriores al punto de predicción. Para los modelos lineales se aplicará escalado estándar, mientras que LightGBM y ARIMA podrán operar sobre variables sin escalar. Las transformaciones se ajustarán de forma independiente en cada bloque de entrenamiento y se serializarán para garantizar consistencia en producción.

7.3. Entrenamiento de modelos base

ElasticNet se empleará como modelo lineal regularizado para selección de variables y captura de relaciones proporcionales. LightGBM se entrenará en paralelo para modelar interacciones no lineales y estructuras complejas del mercado. Ambos modelos generarán predicciones *out-of-fold* que luego alimentarán al meta-modelo Ridge. En paralelo, se podrá ajustar un modelo ARIMA de horizonte un día, cuya salida servirá como predicción complementaria o como entrada adicional al meta-modelo.

7.4. Estimación de volatilidad

Sobre los retornos o residuos del meta-modelo se ajustará un modelo GARCH o EGARCH que proporcione una estimación de volatilidad condicional. Este componente permitirá adaptar la asignación de capital y mantener la volatilidad total dentro de los límites definidos por la competencia. La actualización del modelo de volatilidad se realizará de forma periódica con ventanas móviles para evitar recalibraciones diarias innecesarias.

7.5. Stacking y meta-aprendizaje

Las predicciones de ElasticNet, LightGBM y ARIMA constituirán las variables de entrada para un meta-modelo Ridge, encargado de combinar los resultados de cada modelo base. Este meta-modelo se entrenará únicamente con predicciones *out-of-fold* para preservar la independencia entre entrenamiento y validación. Se evaluará la estabilidad temporal de sus coeficientes para garantizar que el ensamble no se sobreajuste a períodos específicos.

7.6. Optimización e hiperparámetros

La búsqueda de hiperparámetros se limitará a combinaciones eficientes de regularización y tasa de aprendizaje, utilizando búsqueda aleatoria o bayesiana con número de iteraciones restringido. LightGBM se ajustará con parada temprana basada en rendimiento en el bloque de validación. ElasticNet explorará un rango logarítmico de penalizaciones y proporciones L1/L2. El objetivo no es alcanzar el mínimo error puntual, sino maximizar la estabilidad del modelo en distintos intervalos.

7.7. Backtesting y evaluación

El backtesting simulará la operación diaria del sistema. Para cada día se generarán las características con información disponible hasta el día anterior, se realizarán las predicciones base y meta, y se calculará la asignación de capital ajustada por volatilidad. Se registrarán métricas de retorno acumulado, volatilidad, ratio de Sharpe modificado y drawdown máximo. También se incluirán estimaciones de error predictivo (MAE y RMSE) para controlar la coherencia entre desempeño financiero y precisión estadística.

7.8. Prevención de sobreajuste

Todas las transformaciones se entranan únicamente sobre datos previos a la fecha de validación. No se permite ninguna forma de mezclado entre períodos. Se controlará la estabilidad de los coeficientes de ElasticNet y la importancia de variables en LightGBM para detectar desviaciones o dependencia excesiva de características específicas. La regularización del meta-modelo Ridge actúa como capa adicional de control de sobreajuste.

7.9. Reproducibilidad y control computacional

Cada etapa del proceso se ejecutará con semillas fijas, registro de parámetros y serialización de modelos y transformadores. Se asegurará que todo el pipeline se ejecute dentro del límite computacional establecido por Kaggle. Los resultados de cada bloque de validación se almacenarán para análisis posterior y comparación entre configuraciones.

7.10. Criterio final

La configuración seleccionada será aquella que maximice el ratio de Sharpe ajustado por volatilidad en la validación temporal manteniendo un comportamiento estable y coherente entre períodos. El entrenamiento final se realizará con la configuración óptima sobre todo el rango histórico disponible antes de generar las predicciones del conjunto de prueba.

8. Ajuste de Hiperparámetros

El ajuste de hiperparámetros se plantea como un proceso controlado y reproducible enfocado en estabilidad temporal y eficiencia computacional. Dado el volumen de datos y la naturaleza secuencial de la validación, se prioriza la optimización robusta sobre la búsqueda exhaustiva.

8.1. Enfoque general

Se empleará validación temporal *walk-forward* en lugar de particiones aleatorias. Cada bloque de entrenamiento genera un conjunto de métricas que reflejan el desempeño fuera de muestra. El objetivo del ajuste no es únicamente minimizar el error medio, sino asegurar consistencia y bajo drift en los coeficientes y predicciones a lo largo del tiempo.

8.2. ElasticNet

La búsqueda se realiza sobre un espacio logarítmico de $\alpha \in [10^{-5}, 1]$ y proporciones $l1_ratio \in [0,1, 0,9]$. Se utilizará *RandomizedSearchCV* con validación temporal y número limitado de iteraciones (10–20) por ventana. Los parámetros finales se eligen en función de estabilidad y magnitud de coeficientes, no únicamente por métrica de error. La regularización combinada L1–L2 controla tanto la complejidad como la multicolinealidad del modelo.

8.3. LightGBM

El ajuste del modelo no lineal se orienta a encontrar un balance entre sesgo, varianza y coste computacional. Se exploran rangos acotados: número de hojas (31–127), profundidad máxima (6–12), tasa de aprendizaje (0.01–0.05), número de árboles (500–3000), y fracciones de muestreo por fila y columna (0.6–0.9). Se aplica *early stopping* con 50–100 iteraciones de paciencia y validación temporal. El criterio de selección combina el RMSE en validación con la estabilidad del *feature importance*.

8.4. ARIMA

Los parámetros (p, d, q) se estiman mediante búsqueda incremental limitada con AIC y BIC como criterio. Solo se permiten combinaciones pequeñas $(p,q \leq 3)$. Se entrena sobre cada ventana temporal y se evalúa la persistencia del patrón estimado. Este modelo se integra como fuente secundaria de predicción o como entrada para el meta-modelo.

8.5. Ridge (meta-modelo)

El meta-modelo Ridge se ajusta sobre las predicciones *out-of-fold* generadas por los modelos base. El parámetro α se selecciona en escala logarítmica $[10^{-4}, 10^1]$. El criterio de selección se basa en la estabilidad del peso asignado a cada modelo base y en la reducción del error promedio combinado. Se penalizan configuraciones con pesos excesivamente concentrados en un solo modelo.

8.6. GARCH

Los parámetros (p, q) se determinan de forma empírica, típicamente (1,1) o (1,2), priorizando convergencia y estabilidad. La distribución de errores (normal o Student-t) se selecciona en función del ajuste a colas pesadas. El tuning busca minimizar el AIC sin generar inestabilidades

numéricas. El modelo se recalibra cada cierto número de observaciones para mantener capacidad adaptativa sin sobreajuste.

8.7. Procedimiento de búsqueda

El pipeline completo se evalúa mediante búsqueda jerárquica: primero se ajustan los modelos base por separado, luego se calibran los hiperparámetros del meta-modelo con sus predicciones *out-of-fold*, y finalmente se reentrena el sistema completo con los valores óptimos. Todas las búsquedas utilizan semillas fijas y límites estrictos de iteraciones para garantizar reproducibilidad y compatibilidad con entornos de ejecución limitados como Kaggle.

8.8. Criterios de selección

Los hiperparámetros óptimos se eligen con base en: 1. Métricas de validación temporal: RMSE, MAE y ratio de Sharpe simulado. 2. Estabilidad de coeficientes y *feature importance*. 3. Robustez frente a cambios de ventana y a períodos de alta volatilidad. 4. Coste computacional razonable para reentrenamientos periódicos.

La configuración final se define como aquella que maximiza el rendimiento ajustado por riesgo y demuestra coherencia estructural entre ventanas de validación consecutivas.

9. Evaluación y Métricas de Desempeño

La evaluación del modelo se fundamentará en un conjunto de métricas cuantitativas que permitan medir tanto la precisión predictiva como la estabilidad temporal del sistema una vez implementado. El objetivo es establecer un marco de referencia reproducible que sirva para comparar distintas configuraciones de modelos y validar la efectividad del enfoque propuesto en contextos reales.

9.1. Objetivos de la evaluación

El proceso de evaluación busca determinar tres aspectos principales: la exactitud de las predicciones sobre los valores futuros, la capacidad del modelo para mantener consistencia a lo largo del tiempo y la robustez frente a períodos de alta variabilidad. No se persigue únicamente minimizar el error promedio, sino también identificar la confiabilidad del modelo en escenarios de mercado cambiantes.

9.2. Métricas de error

Las métricas de error se utilizarán para medir la diferencia entre los valores reales y las predicciones del modelo. Las principales métricas a emplear serán:

Error cuadrático medio (RMSE): penaliza errores grandes y proporciona una visión general del desempeño global. **Error absoluto medio (MAE)**: mide la desviación promedio en unidades originales y es menos sensible a valores atípicos. **Error porcentual absoluto medio (MAPE)**: permite evaluar el rendimiento en términos relativos, expresando la magnitud del error en porcentaje. **Coeficiente de determinación (R^2)**: cuantifica el grado de explicación de la varianza de la variable objetivo por el modelo.

9.3. Métricas de estabilidad

La estabilidad se analizará a través de indicadores que midan la variabilidad de las métricas a lo largo del tiempo. Esto permitirá identificar si el modelo mantiene su desempeño de forma sostenida o presenta degradación en distintos períodos.

Se calculará la desviación estándar de las métricas de error a lo largo de las ventanas temporales y el coeficiente de variación del RMSE, como indicador del grado de estabilidad estructural del modelo. La baja dispersión se interpretará como un signo de robustez temporal.

9.4. Métricas financieras complementarias

Dado que el contexto de aplicación implica variables económicas o de mercado, se incorporarán métricas que evalúen el desempeño ajustado por riesgo:

Ratio de Sharpe simulado: mide la relación entre retorno medio y volatilidad de las predicciones, útil para interpretar la eficiencia del modelo bajo incertidumbre. **Maximum Drawdown:** identifica la pérdida máxima en un periodo de tiempo simulado, sirviendo como indicador de riesgo extremo en las predicciones.

9.5. Criterios de interpretación

Los resultados de las métricas se interpretarán considerando tanto el valor absoluto como la consistencia entre períodos. Un modelo será considerado adecuado si presenta un RMSE bajo, un MAE estable y un ratio de Sharpe positivo sostenido en múltiples ventanas de validación. Además, se valorará la coherencia entre distintas métricas, evitando configuraciones que optimicen un indicador a costa del deterioro de otros.

9.6. Validación cruzada temporal

El método de validación adoptado será la validación cruzada temporal (*Time Series Cross-Validation*), que consiste en entrenar y evaluar el modelo sobre ventanas progresivas de datos. Este enfoque permite medir el desempeño en condiciones más realistas, preservando la estructura secuencial y evitando fuga de información entre períodos.

9.7. Relevancia de las métricas

Las métricas definidas servirán no solo para evaluar el rendimiento inicial del sistema, sino también para monitorear su desempeño durante la fase operativa. Su función será doble: proporcionar una medida objetiva de éxito en la implementación y actuar como mecanismo de control para la detección temprana de degradación en el modelo con el paso del tiempo.

10. Interpretación de Resultados

La interpretación de los resultados tendrá como propósito traducir los valores obtenidos de las métricas en conclusiones técnicas y operativas sobre el desempeño del modelo. No se trata de analizar resultados existentes, sino de establecer un marco metodológico que permita, en el futuro, entender y justificar el comportamiento del sistema tras su entrenamiento y validación.

10.1. Propósito de la interpretación

El análisis de resultados buscará determinar si el modelo cumple con los objetivos de predicción planteados, evaluando su capacidad de generalización, estabilidad y coherencia con

los supuestos teóricos. La interpretación no se limitará a valores numéricos, sino que también considerará su relevancia práctica dentro del contexto financiero del problema.

10.2. Dimensiones de análisis

La interpretación se centrará en tres dimensiones principales: precisión, estabilidad y relevancia económica. La precisión reflejará el grado de acierto en las predicciones; la estabilidad medirá la consistencia de dicho desempeño a lo largo del tiempo; y la relevancia económica permitirá juzgar si las predicciones tienen sentido en términos de comportamiento real del mercado.

10.3. Análisis comparativo

Una parte importante de la interpretación consistirá en comparar los resultados obtenidos por distintas configuraciones del modelo, analizando cómo el ajuste de hiperparámetros, la selección de características o los métodos de regularización afectan las métricas de desempeño. Este análisis permitirá identificar patrones de mejora o degradación en el modelo.

10.4. Consideraciones sobre el contexto financiero

Dado que el dominio del problema es el mercado bursátil, la interpretación de las métricas deberá relacionarse con las condiciones de riesgo y volatilidad. Por ejemplo, un modelo con bajo error pero alta variabilidad podría considerarse poco confiable desde una perspectiva de inversión. Por el contrario, un modelo con rendimiento moderado pero estable podría tener mayor valor práctico.

10.5. Implicancias y validación conceptual

Finalmente, la interpretación de los resultados permitirá verificar si los hallazgos son coherentes con los principios de las finanzas y la teoría del mercado eficiente. Se espera que el análisis facilite la comprensión de cómo los patrones aprendidos por el modelo se relacionan con fenómenos económicos reales, y si la estrategia propuesta tiene potencial de aplicación más allá del entorno experimental.

11. Librerías y Entorno de Implementación

La implementación del modelo propuesto se realizará principalmente en el entorno de Python, debido a su amplia disponibilidad de librerías científicas, facilidad para manipular datos y compatibilidad con frameworks de aprendizaje automático y análisis estadístico. A continuación se describen las librerías que conformarán el entorno de trabajo base y sus respectivas funciones dentro del proyecto.

11.1. Librerías principales de aprendizaje automático

scikit-learn: será la librería central para la implementación de modelos lineales como ElasticNet y Ridge, además de proveer herramientas para estandarización, particionado temporal, validación y búsqueda de hiperparámetros. **LightGBM**: se utilizará para el modelo de gradient boosting, aprovechando su eficiencia en datos tabulares y su capacidad de manejo de relaciones no lineales. **statsmodels**: permitirá implementar modelos estadísticos clásicos como ARIMA y GARCH, esenciales para el componente temporal y de estimación de volatilidad.

11.2. Librerías para análisis de datos y manipulación

pandas: se usará como base para la manipulación estructurada de datos, limpieza, imputación y construcción de ventanas temporales. **numpy**: se empleará para operaciones numéricas vectorizadas y generación de estructuras de datos optimizadas. **scipy**: servirá como apoyo en funciones estadísticas, optimización y generación de distribuciones.

11.3. Visualización y análisis exploratorio

matplotlib y **seaborn**: se utilizarán para visualizaciones exploratorias, gráficos de correlación, tendencias temporales y curvas de validación. **plotly**: opcionalmente se considerará para visualizaciones interactivas de resultados y comportamiento temporal del modelo.

11.4. Optimización, reproducibilidad y control

joblib: para la serialización de modelos, scalers y metadatos. **optuna** o **scikit-optimize**: como herramientas opcionales para la optimización de hiperparámetros mediante búsqueda bayesiana. **random** y **numpy.random**: para el control de semillas de aleatoriedad y garantizar reproducibilidad experimental.

11.5. Entorno de ejecución

El desarrollo se realizará en un entorno Python 3.10 o superior, con dependencias gestionadas mediante **conda** o **pip**. La ejecución de experimentos, backtesting y generación de gráficos se llevará a cabo en notebooks Jupyter, lo que permitirá documentación simultánea y análisis visual directo.

12. Conclusión

El presente informe ha establecido un marco conceptual y metodológico para abordar el problema de predicción de rendimientos del S&P 500 utilizando un enfoque mixto que combina modelos lineales, no lineales y estadísticos. Se ha detallado la estructura del dataset, los desafíos asociados a valores faltantes, colas pesadas, autocorrelación débil y volatilidad agrupada, así como las estrategias de preprocesamiento y selección de características que se emplearán para mitigar dichos problemas.

Se ha propuesto un pipeline de modelado que integra ElasticNet para interpretabilidad y selección inicial de variables, LightGBM para capturar relaciones no lineales complejas, un meta-modelo Ridge para combinación de predicciones y modelos de volatilidad GARCH y ARIMA para estimar riesgos y patrones temporales. La metodología de entrenamiento y ajuste de hiperparámetros se fundamenta en validación temporal y técnicas robustas de backtesting para evitar sobreajuste y garantizar consistencia.

Asimismo, se han definido las métricas de evaluación y los criterios de interpretación que permitirán, una vez implementado el sistema, medir su desempeño en términos de precisión, estabilidad temporal y relevancia financiera. Se ha descrito el entorno de librerías y herramientas necesarias, asegurando reproducibilidad y trazabilidad de los experimentos.

En síntesis, este trabajo proporciona una hoja de ruta detallada para la implementación de un sistema de predicción robusto y controlado, integrando consideraciones estadísticas, financieras y computacionales. La planificación aquí presentada servirá como base para la ejecución efectiva del proyecto, asegurando que los pasos posteriores se realicen con criterios claros y fundamentados.

13. Bibliografía y Fuentes

A continuación, se presenta una recopilación de fuentes bibliográficas y recursos en línea que han sido fundamentales para el desarrollo y la implementación de la solución propuesta en este trabajo.

- Andrew Ng, *Machine Learning Yearning*.
- Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*.
- Christopher M. Bishop, *Pattern Recognition and Machine Learning*.
- Ian Goodfellow, Yoshua Bengio y Aaron Courville, *Deep Learning*.
- Kaggle, Hull Tactical Market Prediction: <https://www.kaggle.com/competitions/hull-tactical-market-prediction>
- InsightBig, GARCH vs ML Models vs ANNs: Which One for Volatility Prediction: <https://www.insightbig.com/post/garch-vs-ml-models-vs-anns-which-one-for-volatility-prediction>
- Steve Chesa, Stacking Ensembles: Combining XGBoost, LightGBM, and CatBoost to Improve Model Performance: <https://medium.com/@stevechesa/stacking-ensembles-combining-xgboost-lightgbm-and-catboost-to-improve-model-performance-d4247d092c2e>