Modelo Predictivo para el Concurso Hull Tactical Market Prediction

Autor: Elias Sebastian Gill Quintana

Tutor: Diego Pedro Pinto Roa

Ingeniería en Informática

Facultad Politécnica
Universidad
Nacional de Asunción

1. Contexto y Motivación

1. Contexto y Motivación

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.

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 %.

1. Contexto y Motivación

Motivación y relevancia

A diferencia de otros ejercicios de predicción bursátil puramente teóricos, el desafío propone un contexto con implicaciones prácticas para estrategias reales de inversión.

La predicción de rendimientos financieros continúa siendo un tema central en el estudio de la eficiencia de los mercados. El avance de las técnicas de aprendizaje automático permite reconsiderar la “Hipótesis del mercado eficiente” en contextos donde los patrones no lineales o las interacciones entre múltiples variables podrían ofrecer señales útiles para la toma de decisiones.

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.

2. Análisis de los Datos

2. Análisis de los Datos

Introducción al Dataset

El dataset contiene información histórica del mercado financiero, cubriendo múltiples décadas.
Incluye datos de comportamiento del S&P 500 y variables económicas relacionadas.
Su estructura está diseñada para modelar **rendimientos y retornos del mercado** frente a tasas libres de riesgo.

Tanto el dataset de entrenamiento como el de testing possen la misma estructura, pero este último añade algunas features nuevas:

- *is_scored*: indica si la fila se evalúa (para la competencia)
- *lagged_forward_returns:* para comparar predicciones
- *lagged_risk_free_rate*
- *lagged_market_forward_excess_returns*

Total de archivos: **13**
Tamaño total: **12.39 MB**
Columnas: **197**

Estructura **train.csv:**

- Contiene datos históricos completos.
- Incluye la variable objetivo *forward_returns* y variables auxiliares.
- Variables principales:
  - ***date_id***: identificador temporal.
  - **M***: dinámica de mercado/técnica.
  - **E***: macroeconómicas.
  - **I***: tasas de interés.
  - **P***: precios/valoración.
  - **V***: volatilidad.
  - **S***: sentimiento.
  - **MOM***: momentum.
  - **D***: variables binarias.
  - *risk_free_rate* y
  - *market_forward_excess_returns* (solo para entrenamiento).

2. Análisis de los Datos

Análisis inicial del Dataset

Un análisis del dataset de entrenamiento realizado por el usuario “SAMOILOV MIKHAIL” en ***Kaggle*** ofrece una visión clara del problema. En primer lugar, SAMOILOV optimizó el dataset mediante un proceso de minimización, casteando los valores de las columnas para reducir su tamaño en memoria, logrando una disminución del 52% en el espacio ocupado.

El dataset contiene 8,990 observaciones (aproximadamente 24 años de datos si suponemos que cada fila corresponde a un día bursátil) y 98 columnas (features).

```
Mem. usage decreased to  3.16 Mb (52.9% reduction)
Dataset shape: (8990, 97)
Time range of data: from 0 to 8989
```

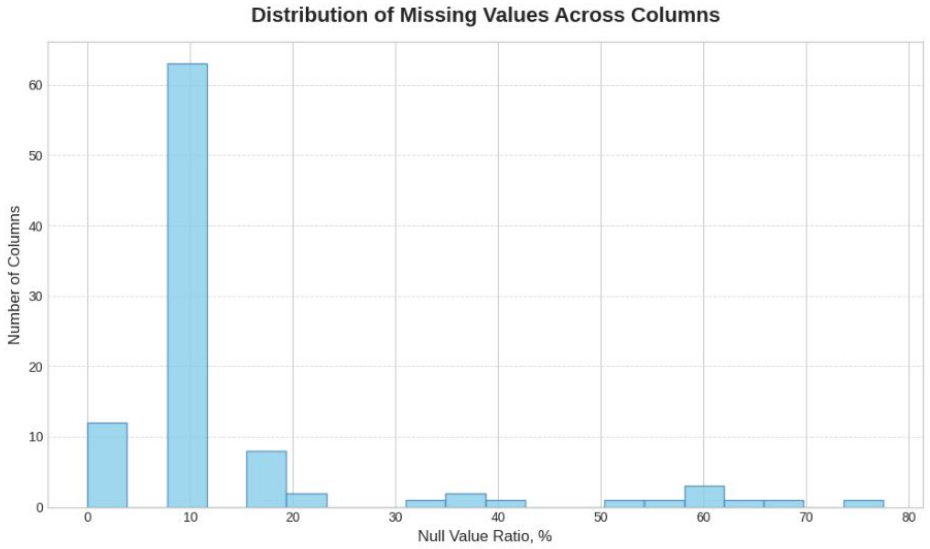
date_id	D1	D2	D3	D4	D5	D6	D7	D8	D9	E1	...	V3	V4	V5	V6	V7	V8	\
											...							
0	0	0	0	1	1	0	0	0	1	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
1	0	0	0	1	1	0	0	0	1	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
2	0	0	0	1	0	0	0	0	1	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
3	0	0	0	1	0	0	0	0	0	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	

2. Análisis de los Datos

Completitud del Dataset

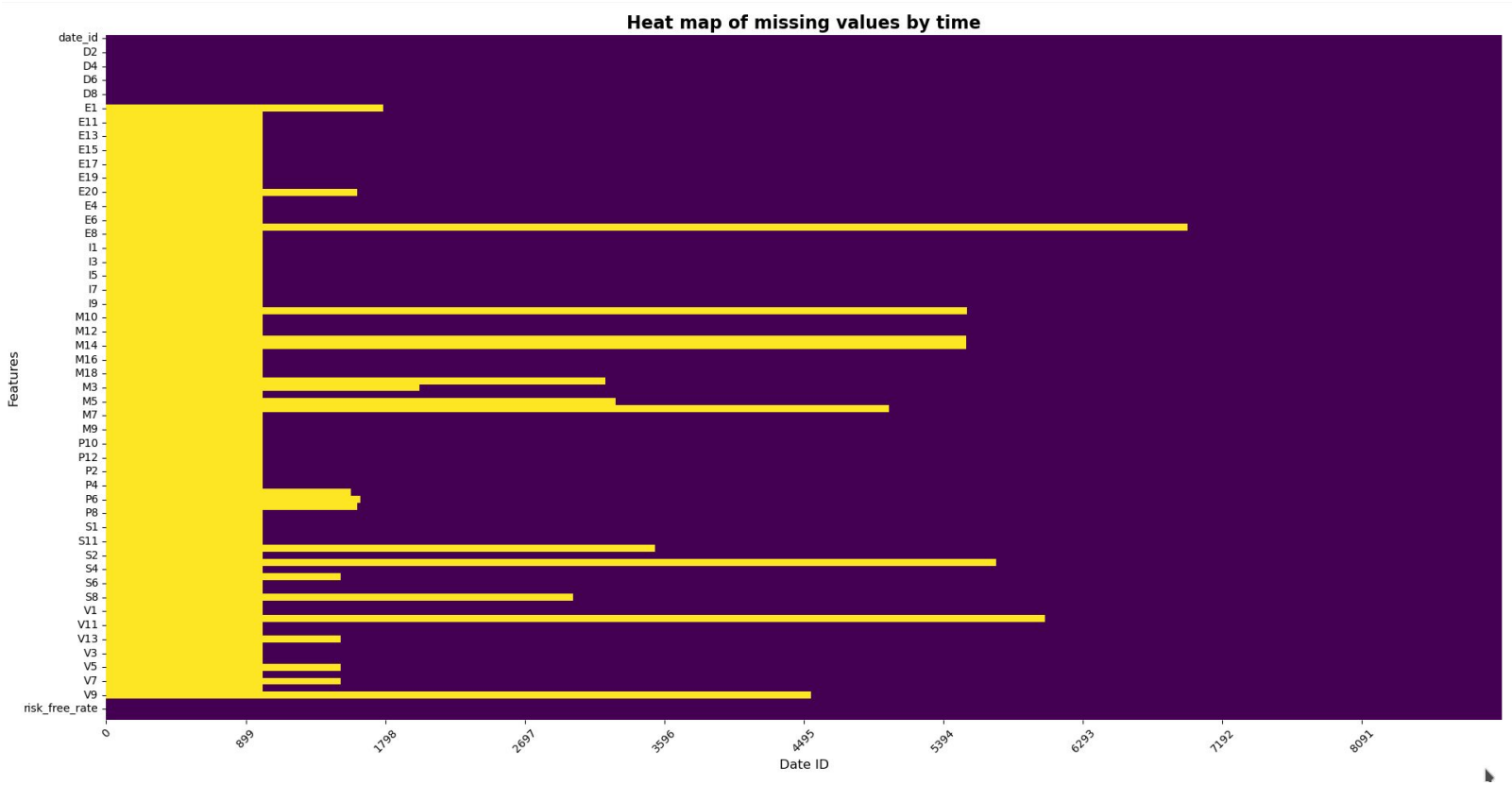
El análisis de las filas mostró una proporción elevada de valores nulos, especialmente en las variables E7, V10, S3, M1 y M14, todas con más del 60% de ausencias.

Del mismo modo, se detectó que a partir del identificador temporal $\text{date\_id} = 5540$ la completitud de datos supera el 95%, por lo que ese punto se considera un umbral útil para trabajar con un subconjunto de datos más limpio (3.450 filas aproximadamente).



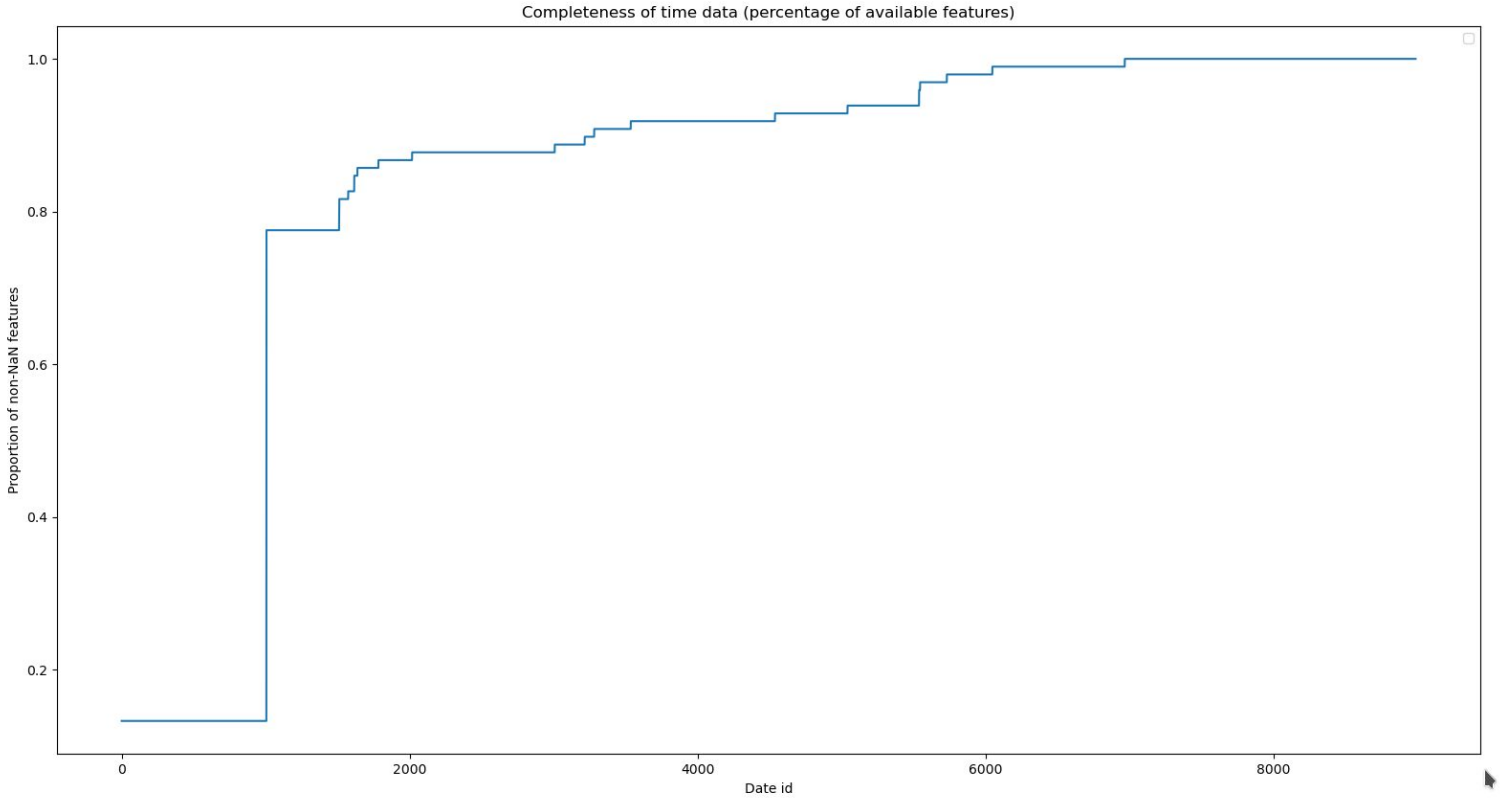
2. Análisis de los Datos

Mapa de calor de valores faltantes



2. Análisis de los Datos

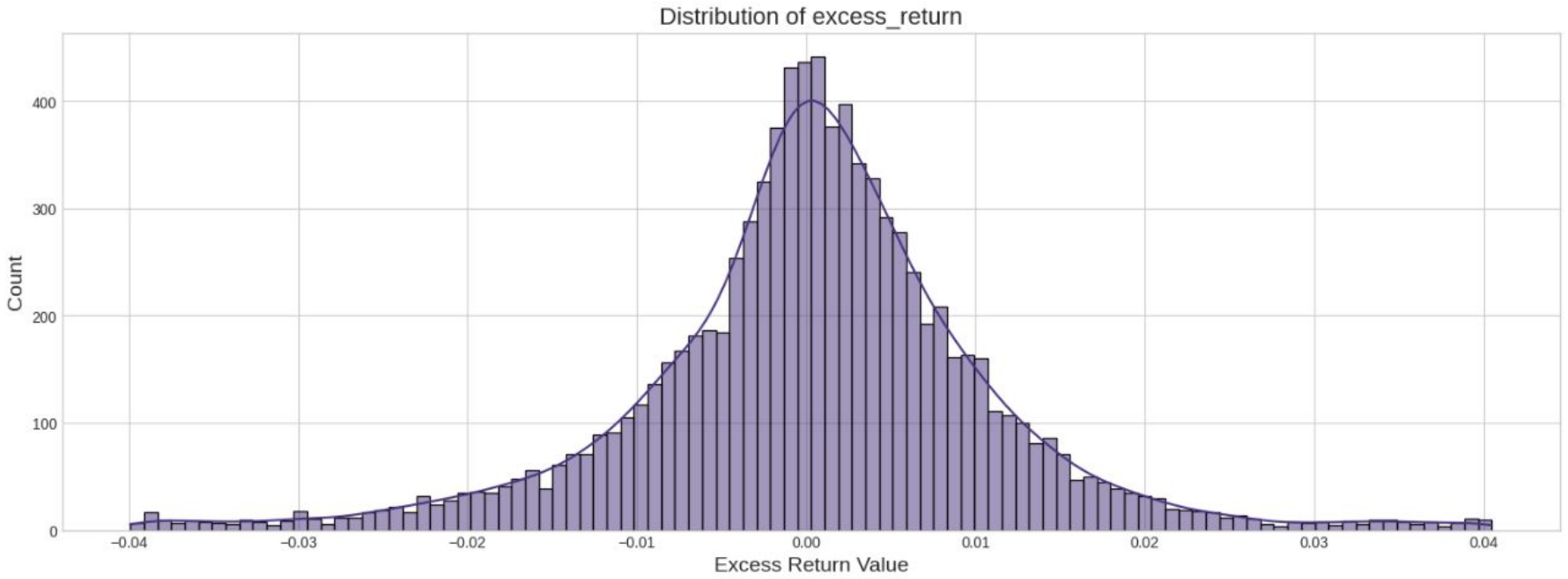
Completitud del dataset



2. Análisis de los Datos

Estudio de la variable objetivo

La variable objetivo según nuestro problema es *excess_return*, definida como la diferencia entre *forward_returns* y *risk_free_rate*. Esta variable no sigue una distribución normal según las pruebas de Jarque-Bera y D'Agostino, presentando colas pesadas y asimetría.

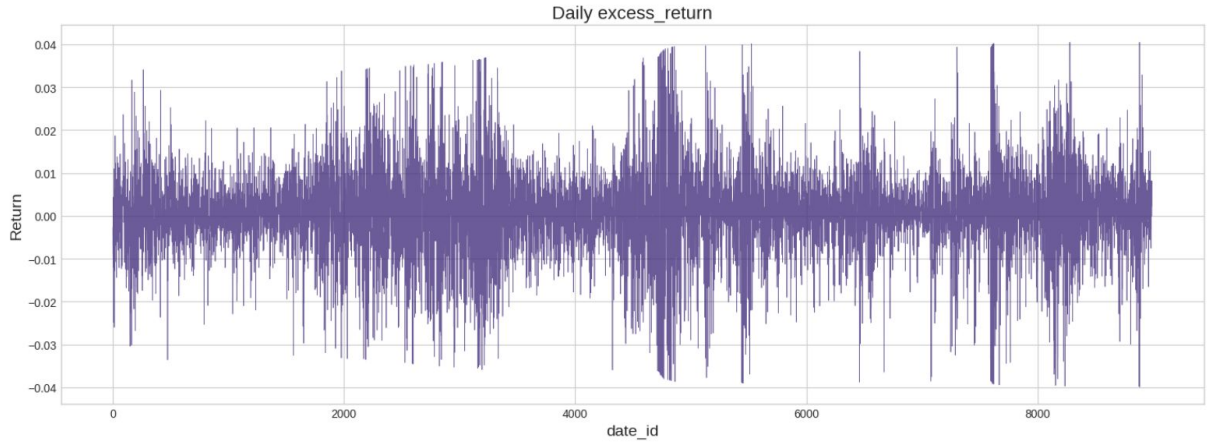


2. Análisis de los Datos

Estudio de la variable objetivo

Se observó además un patrón de volatilidad agrupada, caracterizado por períodos prolongados de alta y baja volatilidad.

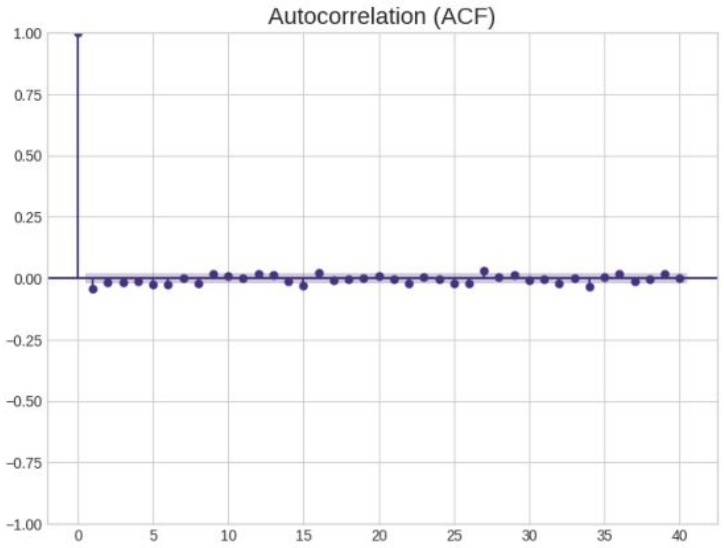
Este comportamiento, común en series financieras, sugiere la presencia de heterocedasticidad condicional.

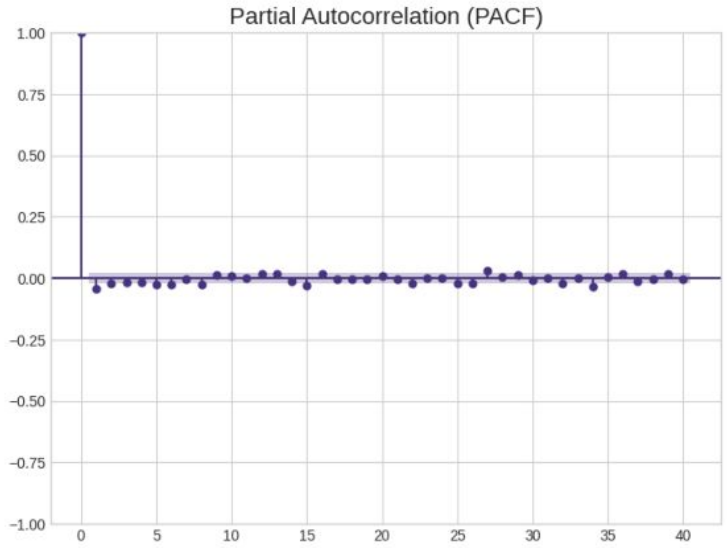


2. Análisis de los Datos

Autocorrelatividad entre variables

La autocorrelación de los retornos a lags cortos es prácticamente nula, lo que sugiere un comportamiento cercano al de un mercado eficiente. En cuanto a correlaciones, las relaciones lineales con el objetivo son muy débiles, lo que indica que la señal predictiva posiblemente reside en interacciones no lineales entre grupos de variables.





2. Análisis de los Datos

Conclusiones

En síntesis, el dataset exhibe **heterocedasticidad** (varianza no constante)**, no estacionariedad** (comportamiento inestable) **y dependencia temporal compleja**, características que invalidan el uso de modelos lineales simples y requieren enfoques más sofisticados.

Adicionalmente, la naturaleza **secuencial y dependiente del tiempo** de los datos financieros impide utilizar métodos de muestreo aleatorio convencionales, ya que estos destruirían la estructura temporal y las relaciones no lineales entre las observaciones.

3. Modelo Propuesto

3. Modelo propuesto

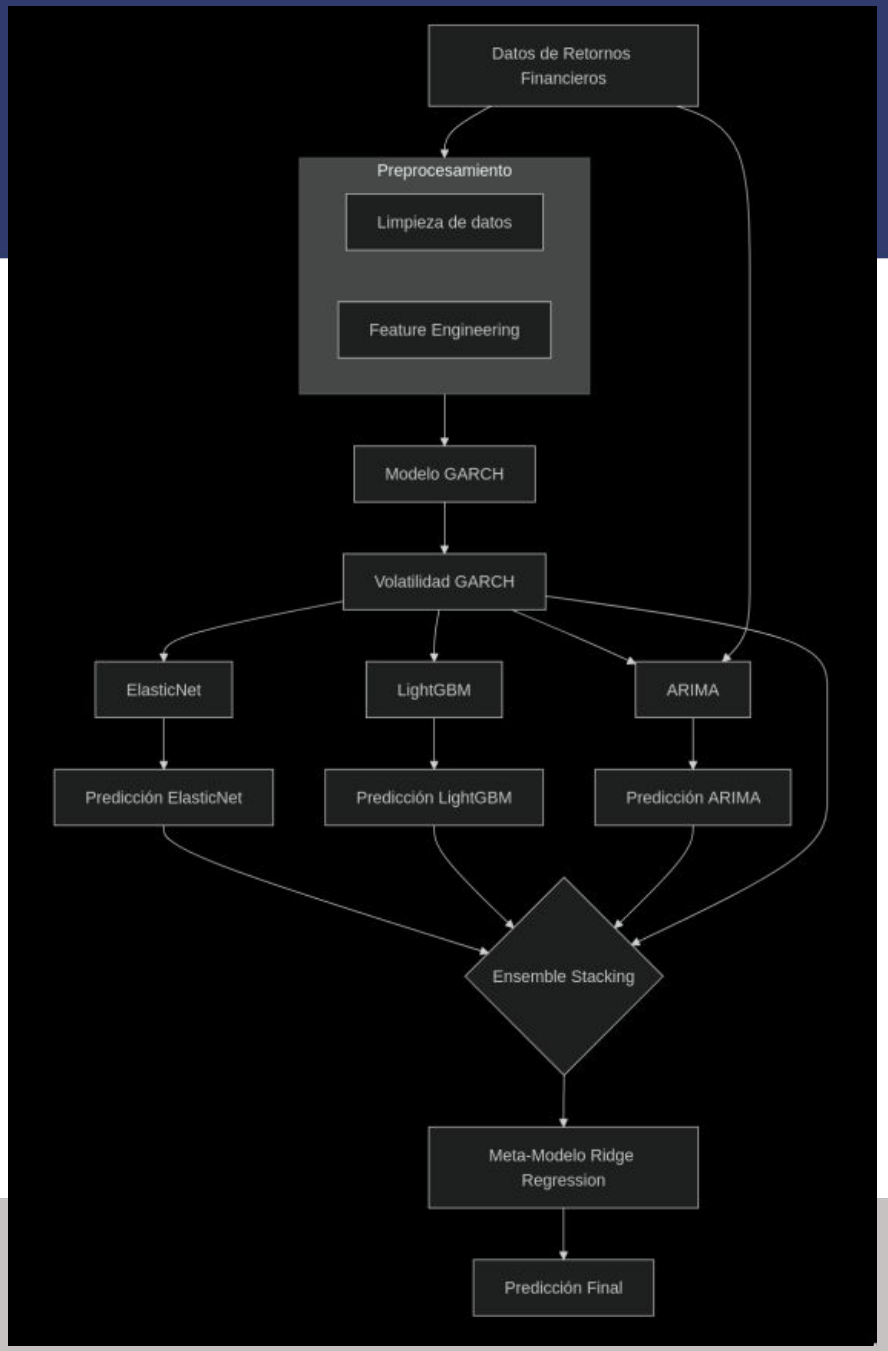
Modelo del stack

El modelo propuesto consiste en un sistema ensamblado basado en Stacking, utilizando los siguientes modelos regresivos:

- **ElasticNet** (penalización L1 + L2)
- **LightGBM** (gradient boost basado en arboles de decision)
- [Opcional] **ARIMA** (modelo estadístico clásico)

Como Meta-learner se utilizará el modelo de regresión lineal **Ridge**.

Todos los modelos recibirán como feature un valor de volatilidad calculado mediante el modelo estadístico **GARCH**.



3. Modelo propuesto

Criterios y riesgos asumidos

Se optó por un modelo de *stacking* debido a su mayor robustez en predicciones en comparación con modelos individuales, a pesar de su explicabilidad moderada-baja.

Se prefirió **LightGBM** sobre **XGBoost** por su mejor rendimiento y potencial de mayor exactitud, aunque con el riesgo de sobreajuste, que deberá monitorearse.

ARIMA se incluye en el *stack* para capturar patrones lineales y de autocorrelación que otros modelos podrían no abordar eficazmente. Sin embargo, su uso es opcional, ya que se requiere evaluar su aporte a la predicción frente al costo computacional, especialmente considerando la no estacionariedad de los datos.

4. Metodología de entrenamiento

4. Metodología de entrenamiento

Preprocesamiento de los datos

Antes del entrenamiento, los datos de train.csv se limpian manejando valores faltantes mediante imputación con **forward-fill** (opcionalmente se pueden hacer pruebas utilizando la mediana).

Para el modelo ARIMA, se realiza el entrenamiento directamente sobre la feature *forward_returns,* aplicado una transformación logarítmica para lograr estacionariedad.

La volatilidad GARCH se calcula a partir de columnas V* y se integra como feature clave en todos los modelos.

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right)$$

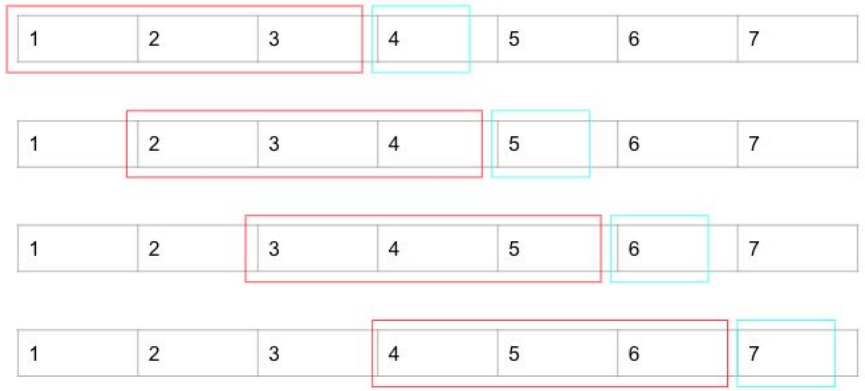
4. Metodología de entrenamiento

Entrenamiento de Modelos Base con Sliding Window

Los modelos base se entrenan usando un esquema de **sliding-window** para recorrer el dataset completo..

La ventana inicial cubre 252 días (un año bursátil aproximado), deslizándose para re-entrenar y predecir un horizonte de 1 día, alineado con *forward_returns*. El re-entrenamiento (volver a calcular los parámetros de los modelos) se realizará cada 7-30 “dias” a modo de optimizar el tiempo computacional del entrenamiento.

GARCH se incorpora como parámetro en cada base learner para modelar volatilidad dinámica.



4. Metodología de entrenamiento

Entrenamiento del Meta-Learner

Una vez generadas las predicciones de los modelos base, el dataset se divide en 80% para la fase de entrenamiento y 20% para testing.

El meta-learner Ridge recibe como features las predicciones de LightGBM, ElasticNet y ARIMA, más la volatilidad GARCH, aprendiendo a ponderarlas.

En total se calcula que se realizarán entre 500 y 1000 ciclos de entrenamiento.

Se implementará un backtesting para evaluar el desempeño diario de la estrategia de inversión, asignando fondos (de 0 a 2) en función de las predicciones generadas usando datos hasta el día previo.

5. Ajuste de Hiperparametros

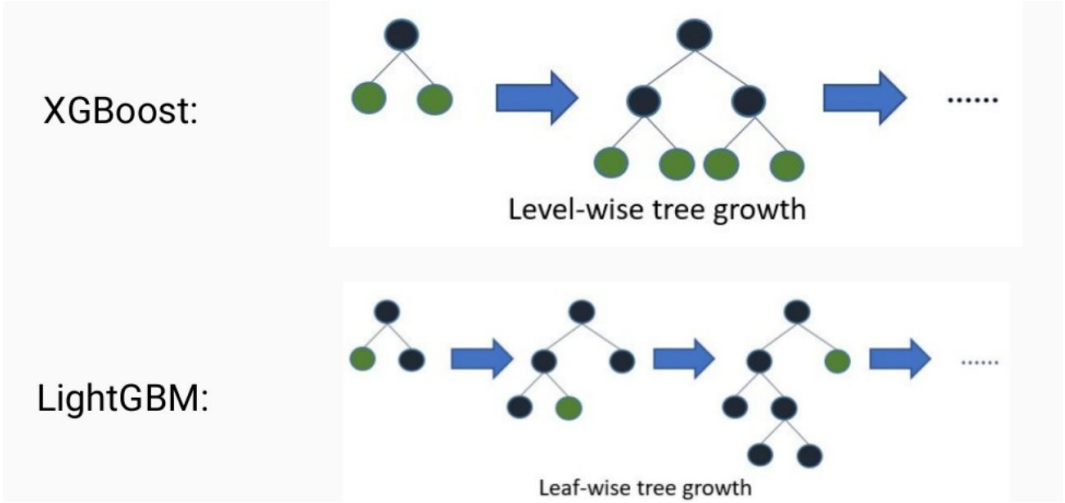
5. Ajuste de hiperparametros

HiperParametrización de LightGBM

Parámetros de entrenamiento:

- Se exploran rangos acotados: número de hojas (31–127),
- profundidad máxima (6–12),
- tasa de aprendizaje (0.01–0.05),
- numero de arboles (500–3000),
- y fracciones de muestreo por fila y columna (0.6–0.9).

Se aplica parada temprana con 50–100 iteraciones de paciencia y validación temporal. El criterio de selección combina el RMSE y la estabilidad.



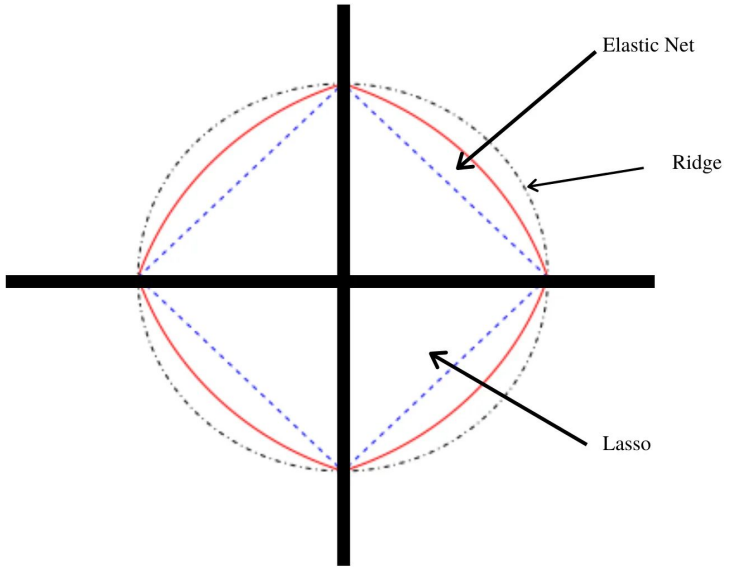
5. Ajuste de hiperparametros

HiperParametrización de ElasticNet

Parámetros de entrenamiento:

- La búsqueda se realiza sobre un espacio logarıtmico de $\alpha \in [10{-}5, 1]$ y proporciones $L1\ ratio \in [0{,}1, 0{,}9]$.
- Se utilizarra $RandomizedSearchCV$ 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.



5. Ajuste de hiperparametros

HiperParametrización de ARIMA

Parámetros de entrenamiento:

- Los parámetros (p, d, q) se estiman mediante búsqueda incremental limitada con *AIC* y *BIC* como criterio.
- Solo se permiten combinaciones pequeñas (p, d, $q \leq 3$). Se entrena sobre cada ventana temporal y se evalúa la persistencia del patrón estimado.

$$\overbrace{y_t^* = \Delta^d y_t}^{\mathbf{I}} \quad \text{serie}$$

$$y_t^* = \mu + \underbrace{\sum_{i=1}^{p} \phi_i y_{t-i}^*}_{\text{AR}} + \underbrace{\sum_{i=1}^{q} \theta_i \epsilon_{t-i}}_{\text{MA}} + \epsilon_t$$

serie diferenciada — constante — error

$$\text{AIC} = 2k - 2\ln(\hat{L})$$

$$\text{BIC} = k\ln(n) - 2\ln(\hat{L}).$$

5. Ajuste de hiperparametros

HiperParametrización de Ridge (meta-modelo)

Parámetros de entrenamiento:

- El meta-modelo Ridge se ajusta sobre las predicciones out-of-fold generadas por los modelos base.
- El parámetro α se selecciona en escala logar´ıtmica [10−4, 101].
- 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.

NOTA: Los parámetros (p, q) del modelo GARCH se determinan de forma empírica (1,1) o (1,2). La distribución de errores utilizada es la distribución normal.

$$RSS_{ridge} = \sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \sum_{j=1}^{p} {\beta_j}^2$$

$$\sigma_t^2 = \omega + \alpha \sum_{p=1}^{p} \varepsilon_{t-p}^2 + \beta \sum_{q=1}^{q} \sigma_{t-q}^2$$

6. Métricas de desempeño y evaluacion

6. Métricas de desempeño y evaluación

Métrica	Fórmula	Explicación
Raíz del Error Cuadrático Medio **(RMSE)**	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$	Mantiene las unidades originales de la variable objetivo, facilitando la interpretación directa del error promedio en la predicción. Penaliza fuertemente los errores grandes.
Error Absoluto Medio **(MAE)**	$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}\lvert y_j - \hat{y}_j \rvert$	Calcula la diferencia absoluta entre cada valor real y su valor predicho, luego promedia todas esas diferencias.
Sharpe Ajustado	$S = \frac{R_p - R_f}{\sigma_p}$	Rendimiento excesivo dividido por volatilidad, penalizando estrategias con >120% volatilidad del mercado (métrica oficial Kaggle).

7. Comparación con otros modelos

7. Comparación con otros modelos

Nuestro modelo adopta un enfoque de *stacking* que integra LightGBM, ElasticNet y ARIMA, con un meta-learner Ridge encargado de combinar las predicciones. Además, incorpora GARCH para modelar la volatilidad dinámica del mercado, con el objetivo de mejorar la estimación del riesgo y optimizar el *Sharpe ratio* ajustado.

A diferencia de los *stacks* más destacados en la comunidad de Kaggle —que suelen combinar XGBoost, LightGBM, ElasticNet y Ridge—, nuestra propuesta integra técnicas estadísticas clásicas de regresión, apostando hacia un enfoque híbrido.

Una vez implementado, el modelo se evaluará frente al *stack* comunitario mediante *backtesting* con validación temporal, simulando asignaciones diarias basadas en información disponible hasta el día previo para preservar la estructura temporal.

Las métricas principales serán el *Sharpe ratio* ajustado (retorno por unidad de riesgo, con límite de volatilidad del 120%) y el RMSE (precisión en las predicciones de retornos). Esta comparación permitirá determinar si la incorporación de ARIMA y GARCH aporta una mejora significativa en el desempeño frente al enfoque de Kaggle, justificando así su mayor costo computacional.

8. Librerias a utilizar

8. Librerias a utilizar

Librería	Propósito
pandas	Manejo y Preprocesamiento de Datos
numpy	Modelado, Ensemble y Métricas
lightgbm	Gradient Boosting
statsmodels	Modelos estadísticos (ARIMA)
pmdarima	Auto-selección ARIMA
arch	Modelado GARCH
optuna	Visualización de resultados
seaborn	Visualización avanzada

9. Trabajos futuros

9. Trabajos futuros

Para **trabajos futuros**, sería interesante explorar las siguientes líneas:

- Implementar ARIMAX incorporando como variables explicativas ciertos parámetros del *dataset*.
- Sustituir o complementar LightGBM con XGBoost dentro del *pipeline* de *stacking*, dado que este último constituye actualmente la solución con mejor desempeño en la comunidad de Kaggle.
- Probar distintas estrategias de partición del conjunto de entrenamiento, priorizando las secciones del *dataset* con mayor limpieza, consistencia y completitud.

10. Conclusión

10. Conclusión

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 de la solución, asegurando que los pasos posteriores se realicen con criterios claros y fundamentados.

Gracias por su atención

11. Referencias

Libros:

- Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
- Geron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- Kubat, M. (2017). An Introduction to Machine Learning (2nd ed.). Springer.
- Ng, A. (n.d.). Machine Learning Yearning. [Manuscript no publicado].

Competencia de Kaggle:

- Blair Hull, Petra Bakosova, Laurent Lanteigne, Aishvi Shah, Euan C Sinclair, Petri Fast, Will Raj, Harold Janecek, Sohier Dane, and Addison Howard. Hull Tactical - Market Prediction. https://kaggle.com/competitions/hull-tactical-market-prediction, 2025. Kaggle.
- Samoilov Mikhail. (n.d.). Hull Tactical Data Analysis. Kaggle. https://www.kaggle.com/code/samoilovmikhail/hull-tactical-data-analysis
- Imaad Mahmood. (n. d). Hull Market Prediction. Kaggle. https://www.kaggle.com/code/imaadmahmood/hull-market-prediction

Artículos en Línea y Blogs:

- Chesa, S. (n.d.). Stacking Ensembles: Combining XGBoost, LightGBM, and CatBoost to Improve Model Performance. Medium. https://medium.com/@stevechesa/stacking-ensembles-combining-xgboost-lightgbm-and-catboost-to-improve-model-performance-d4247d092c2e
- InsightBig. (n.d.). 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