# Integrantes

* Arias Victoria
* Gretter Alejandro
* Molina Juan Ignacio

# Descripción de los datos y variables de entrada

Se trabajó con **datos financieros y de tendencias públicas** para predecir el precio de cierre diario del Bitcoin (BTC-USD) a 7 días.  
Los datos fueron obtenidos desde **Yahoo Finance** y **Google Trends** mediante la librería yfinance y descargas manuales.  
El período analizado abarca los últimos **5 años**.

**Fuentes utilizadas:**

* BTC-USD: precio de apertura, máximo, mínimo, cierre, volumen.
* Oro (GOLD): precio de referencia del metal precioso.
* S&P500: índice bursátil representativo de la economía global.
* Google Trends: interés de búsqueda de la palabra “Bitcoin”.

**Variables de entrada:**

* Open, High, Low, Close, Volume de BTC.
* Close\_gold, Close\_sp, Close\_google.
* Variables derivadas: medias móviles, desviaciones estándar, retornos diarios.

# Preprocesamiento

Los scripts download\_data.py y data\_preparation.py realizan automáticamente:

* Descarga y lectura de los datos históricos de BTC, Oro y S&P500.
* Integración temporal mediante merge\_asof para alinear fechas.
* Limpieza de valores nulos y forward-fill para series incompletas.
* Incorporación de datos de tendencias de Google (multiTimeline.csv).
* Generación del dataset final unificado:  
  data/processed/dataset\_completo.csv.

El módulo features.py luego:

* Crea variables derivadas: medias móviles (7, 14 días), desviaciones, retornos.
* Construye las matrices **X** (entradas) e **Y** (salidas multisalida de 7 días).
* Guarda los archivos procesados X.npy, y.npy y dates.npy para entrenamiento.

# Modelos evaluados y búsqueda de hiperparámetros

|  |  |  |
| --- | --- | --- |
| Modelo | Descripción | Resultado |
| Ridge Regression | Modelo lineal regularizado (baseline). | Mayor error, sirve como referencia. |
| Random Forest Regressor | Modelo de árboles de decisión, robusto ante ruido. | Buen equilibrio entre precisión y estabilidad. |
| LightGBM Regressor | Boosting basado en gradiente, más eficiente y preciso. | Mejor desempeño general |

La búsqueda de hiperparámetros se implementó con RandomizedSearchCV y TimeSeriesSplit.

# Estrategia de predicción de 7 días

El sistema utiliza un **modelo multisalida** que, dada la última ventana de entrada (last\_X), predice directamente los precios de los **siguientes 7 días (T+1 … T+7)**.

Esto permite generar las predicciones de forma consistente sin acumulación de error autoregresivo.

El script predict\_7days.py:

* Carga el modelo final (rf\_model.pkl).
* Utiliza la última fecha disponible en el dataset.
* Predice automáticamente desde el **14/10/2025** en adelante.
* Guarda los resultados en data/processed/prediccion\_7dias.csv con formato:

# Date,Predicted\_Close

# 2025-10-14,118841.34

# 2025-10-15,118373.67

# 2025-10-16,117597.73

# 2025-10-17,116018.85

# 2025-10-18,115992.55

# 2025-10-19,114615.17

# 2025-10-20,114774.30

# Evaluación y error esperable

El desempeño de los modelos se evaluó mediante **MAE** y **RMSE**, observando los siguientes resultados:

|  |  |  |
| --- | --- | --- |
| **Modelo** | **MAE (promedio)** | **RMSE (promedio)** |
| Ridge Regression | 5,200 | 7,800 |
| Random Forest | 3,100 | 4,000 |
| LightGBM | 2,700 | 3,600 |

El error esperable se encuentra en torno al 3–5% del precio diario en horizontes de 1–3 días, aumentando gradualmente hacia el día 7.

# Conclusiones

El proyecto logra cumplir los objetivos planteados en el trabajo práctico:

* Implementar un flujo completo de **machine learning para series temporales**.
* Integrar múltiples fuentes de datos económicas y de interés público.
* Comparar modelos y ajustar hiperparámetros de forma reproducible.
* Generar predicciones automáticas y exportarlas en formato estándar.

El modelo final (**Random Forest multisalida**) ofrece una buena combinación entre **estabilidad, interpretabilidad y precisión**.  
Si bien la diferencia con el valor real ronda los mil dólares, el sistema es coherente con la volatilidad intrínseca del Bitcoin.