INFORME TRABAJO PRÁCTICO

INTELIGENCIA ARTIFICIAL: MACHINE LEARNING

Ingeniería Informática - 2025

Alumnos:

- Clementz Nicolas
- Lerda Mateo
- Canello Lucas
- Segura Mijael

Profesores:

- Fisanotti, Juan Pedro
- Rossanigo, Ariel
- Ferrero, Mariano

Descripción de los datos y variables de entrada

1. OHLC BTC: Datos diarios obtenidos desde la fecha más antigua disponible mediante la API de Binance. Se usaron los precios de Apertura, Máximo y Mínimo como variables, siendo el Cierre nuestro target.

2. OHLC WTI y OHLC BRENT: Precios del petróleo producido en EE. UU. y el Mar del Norte. Su inclusión se basa en la relación entre el costo energético y la minería de BTC, además de su influencia en la inflación global.

3. OHLC SPX (Ticker del S&P 500): Indicador del sentimiento de riesgo de los inversores. Históricamente mostró una correlación positiva moderada con el BTC, ya que ambos responden a la confianza o incertidumbre del mercado.

4. OHLC XAU/USD (Precio del oro en USD): Activo considerado reserva de valor. Si bien la correlación con BTC es baja, sus movimientos reflejan el entorno macroeconómico global.

5. CPIAUCSL: Índice de precios al consumidor de EE. UU. Incluido por su relación con la inflación: un IPC alto puede incentivar la compra de BTC como resguardo de valor.

6. VIX: Índice de volatilidad del mercado. A menudo mantiene una correlación negativa con el BTC, reflejando que el aumento de incertidumbre en los mercados puede afectar su cotización.

Preprocesamiento sobre nuestros datos

Hicimos un preprocesamiento similar sobre todos nuestros datos, para convertir las fechas al mismo tipo de dato, además de contar con una nomenclatura de features consistente que nos permita manejarlas mejor. También hubo que realizar un preprocesado para detectar columnas sin valor en alguno de sus registros, así como convertir cadenas de texto a números (por ejemplo el volumen, que nos venía en formatos como "209,16K"). De esto fuimos perdiendo varios datos, ya que muchas de las variables seleccionadas no eran operables los fines de semana.

Una vez realizado todo esto, y teniendo nuestros datos en distintos datasets ya de una forma mucho más manejable y con features consistentes, decidimos integrar todo en un único dataset para entrenar a nuestros modelos, uniendolos a todos por su columna "Date".

Luego, empezamos a entrenar nuestro primer modelo, que fue un Random Forest, para el cual nos dimos cuenta que no contábamos con muchas features, entonces las predicciones eran bastante malas. Lo que hicimos al ver esto, fue empezar a calcular features polinómicas (usando **PolynomialFeatures**) a partir de todas las columnas que teníamos, usando un grado máximo de 2 para tampoco excedernos, ya que desde la fecha más antigua del BTC los registros eran aproximadamente 2000, con lo cual en un grado mayor íbamos a contar con pocos datos para tantas features.

Una vez hecho esto, volvimos a entrenar a nuestro modelo, y nos dio resultados muchísimo más alentadores. También nos dimos cuenta que la fecha no estaba aportando ningún valor al mismo, con lo cual la primera decisión fue tratar de extraer más datos de ella en varias columnas (año, mes, día de la semana), pero tampoco hubo un cambio significativo.

Una vez teníamos un modelo que predecía un poco mejor, nos dispusimos a mejorarlo, entonces investigando un poco, pudimos encontrar que las **medias móviles** son útiles en modelos que trabajan con datos que cambian en el tiempo (como nuestros precios diarios), ya que lo ayudan a entender mejor si el precio viene bajando o subiendo sostenidamente, en lugar de ver valores diarios aislados. Las medias calculadas son:

BTC_Close_MA7: promedio móvil de 7 días del precio de cierre. Suaviza fluctuaciones diarias y refleja tendencia a corto plazo.

BTC_Close_MA30: promedio móvil de 30 días del precio de cierre. Muestra la tendencia más general o de largo plazo.

BTC_Close_diff1: diferencia entre el precio de cierre actual y el del día anterior. Indica el cambio diario del precio, útil para medir la variación o volatilidad.

Finalmente, previo al entrenamiento de cada uno de nuestros modelos se creó un pipeline que incluía un **scaler** para las features numéricas, buscando manejarnos dentro de rangos consistentes para poder ayudar a modelos que sean sensibles a la escala de los datos.
Usamos **StandardScaler** para transformarlas en datos con media 0 y desviación estándar 1

Modelos evaluados

KNN Regressor:

Usado como modelo inicial. No logró buenos resultados al no capturar tendencias temporales ni relaciones complejas.

Random Forest Regressor:

Modelo basado en múltiples árboles de decisión. Tras ajustar parámetros (n_estimators=200, max_depth=None), mostró mejoras significativas gracias a su capacidad de modelar relaciones no lineales y reducir el sobreajuste.

MLP Regressor:

Red neuronal multicapa con capas de 128 y 64 neuronas. Permite representar relaciones complejas, pero requiere más datos y tiempo de entrenamiento. Su desempeño fue inferior al de Random Forest (mayor MAE), aunque con mejor suavizado en fluctuaciones.

XGBoost Regressor:

Modelo de boosting que combina árboles secuencialmente. Configurado con n_estimators=2000, learning_rate=0.03, max_depth=7, subsample=0.9 y colsample_bytree=0.9, presentó los mejores resultados (menor MAE y mayor R^2), detectando patrones temporales y relaciones sutiles entre los indicadores y el precio del BTC.

Estrategia de predicción de los 7 días:

Para generar la predicción de los próximos 7 días, entrenamos siete modelos independientes, uno para cada horizonte temporal (BTC-Close-t+1, BTC-Close-t+2, ... BTC-Close-t+7).
Cada modelo fue entrenado utilizando las mismas variables de entrada, pero con el target desplazado según el día a predecir.
Esta estrategia nos permitió ajustar individualmente los parámetros y errores esperables por día, evitando la acumulación de error típica de los enfoques secuenciales (donde una predicción alimenta a la siguiente).
A diferencia de lo esperado, el mejor desempeño no se observó en los primeros días, sino en los últimos horizontes, donde los errores fueron relativamente menores. Esto podría deberse a que, al usar medias móviles y diferencias temporales como features, los modelos lograron captar mejor las tendencias generales del mercado que las fluctuaciones diarias más bruscas.

Conclusión:

El proyecto integró múltiples fuentes financiera y macroeconómica de información para modelar el precio de cierre diario del Bitcoin. A través de un preprocesamiento y unificación de datos, se logró construir un dataset coherente. El modelo Random Forest sirvió como línea base, revelando la necesidad de ampliar el espacio de características mediante la generación de features polinómicas y la incorporación de indicadores temporales derivados, luego se probaron distintos algoritmos, donde XGBoost Regressor mostró el mejor desempeño. Finalmente, la estrategia de entrenar modelos independientes para cada horizonte de predicción, permitió minimizar la acumulación de error y mejorar la estabilidad de los resultados, con un rendimiento superior en los horizontes más largos.

En conjunto, el trabajo demuestra que la combinación de fuentes macroeconómicas relevantes, ingeniería de características temporales y modelos de ensamble puede ofrecer predicciones robustas del precio del BTC.