Aprendizaje Automático
Proyect Final

1st Maiky Rodríguez Aguilar
Estudiante Maestría Ciencias de la Computación
Tecnológico de Costa Rica
Cartago, Costa Rica
m.rodriguez.10@estudiantec.cr

***Index Terms*—XGBoost, Logistic Regression, Random Forest, Bandas de Bollinger, RSI, Moving Average, Moving Average Cross Over**

***Abstract*—Esta investigación analiza la eficacia comparativa de tres modelos de aprendizaje automático (XGBoost, Regresión Logística y Random Forest) para predecir las tendencias de precios de Bitcoin. Aprovechando los datos recuperados a través de la API Python-binance, el estudio analiza las fluctuaciones de precios de BTC desde enero de 2023 hasta el 2 de noviembre de 2024, utilizando intervalos de 5 minutos. El conjunto de datos incluye precios de apertura, precios de cierre, precios altos y bajos y volúmenes de negociación diarios. Utilizando metodologías de promedio móvil (MA) y promedio móvil exponencial (EMA), el estudio incorpora tendencias de corto plazo (5 días), mediano plazo (30 días) y largo plazo (90 días), aumentadas por el índice de fuerza relativa (RSI) y las bandas de Bollinger para refinar las señales comerciales.**

La investigación desarrolla estrategias comerciales basadas en tendencias (al alza, a la baja y laterales) y proporciona recomendaciones específicas (comprar, vender, mantener) y asignaciones (todas, la mitad, ninguna) según las condiciones de mercado identificadas. Se construyen dos modelos para cada algoritmo: uno sin optimización de parámetros y otro con hiperparámetros optimizados que apuntan a una precisión de al menos el 70 %. El rendimiento se evalúa utilizando métricas que incluyen accuracy, precision, recall, F1 Score y análisis de matriz de confusión. El estudio concluye comparando el rendimiento del modelo y la efectividad de las estrategias generadas, destacando el impacto de la optimización y la idoneidad de cada modelo para el trading de criptomonedas.

I. Introducción

Las criptomonedas han ganado una tracción significativa como activos de inversión, siendo Bitcoin la más destacada. Sin embargo, la naturaleza altamente volátil de Bitcoin presenta desafíos para los traders que intentan predecir los movimientos de precios y optimizar las estrategias comerciales. Los algoritmos de aprendizaje automático (ML) han surgido como herramientas poderosas para analizar conjuntos de datos financieros tan complejos, ofreciendo el potencial de pronosticar tendencias y automatizar la toma de decisiones.

Este documento explora la aplicación de tres modelos de ML (XGBoost, Regresión Logística y Random Forest) para predecir las tendencias de precios de Bitcoin y evaluar su desempeño. El análisis utiliza datos de alta frecuencia obtenidos de la API Python-binance, que comprende información de precios y volumen en intervalos de 5 minutos desde enero de 2023 hasta el 2 de noviembre de 2024. Incorporando indicadores técnicos como promedios móviles (MA), promedios móviles exponenciales (EMA), índice de fuerza relativa (RSI) y bandas de Bollinger, este estudio formula estrategias comerciales para guiar las decisiones de inversión.

La metodología propuesta clasifica las tendencias del mercado en movimientos ascendentes, descendentes y laterales, vinculándolas a acciones específicas (comprar, vender, mantener) y estrategias de asignación (todas, mitad, ninguna). Para evaluar la eficacia de estas estrategias, se desarrollan dos modelos para cada algoritmo: uno sin optimización de parámetros y otro con hiperparámetros optimizados diseñados para lograr una precisión mayor. El rendimiento del modelo se compara a través de un análisis exhaustivo de métricas, que incluyen "Accuracy", "Precision", "Recall", "F-1 Score" y matrices de confusión.

Las contribuciones de esta investigación son las siguientes:

1) Proporciona un análisis comparativo del rendimiento de los modelos XGBoost, Regresión Logística y Random Forest en la predicción de las tendencias de precios de Bitcoin.
2) Integra indicadores técnicos para mejorar la toma de decisiones.
3) Evalúa el impacto de la optimización de parámetros en la precisión del modelo y la confiabilidad de la estrategia.

Al integrar el análisis técnico con algoritmos de ML, este estudio tiene como objetivo ofrecer información sobre estrategias comerciales efectivas de Bitcoin y destacar las ventajas de los enfoques basados en ML en los mercados financieros.

II. Trabajos Relacionados

La predicción del precio de las criptomonedas ha recibido una atención significativa debido a su complejidad y alta volatilidad. Se han explorado diversas técnicas de aprendizaje automático para predecir los precios, en particular de Bitcoin, aprovechando tanto los indicadores técnicos como los datos históricos de las criptomonedas.

Hafid et al., 2024 [1] utilizaró el regresor XGBoost combinado con indicadores técnicos para pronosticar los precios de las criptomonedas. Su enfoque demostró la eficacia de los métodos de conjunto para proporcionar predicciones precisas, en particular cuando se mejoran con características técnicas

como promedios móviles e índice de fuerza relativa (RSI). De manera similar, Li [2] se centró en el uso de modelos de aprendizaje automático, específicamente árboles de decisión y máquinas de vectores de soporte (SVM), para predecir las tendencias de precios de Bitcoin en función de datos históricos e indicadores técnicos. El estudio de Li mostró resultados prometedores para la predicción del movimiento de precios a corto plazo.

Gradojevic et al., 2023 [3] realizaron un estudio sobre la previsión del precio de Bitcoin utilizando análisis técnico y modelos de aprendizaje automático. Su trabajo destacó la importancia del análisis técnico para identificar patrones de mercado, con un enfoque especial en el rendimiento de los Random Forest. Argumentaron que la incorporación de indicadores técnicos mejoraba la precisión de la predicción, lo que hacía del modelo un candidato sólido para la predicción de precios en mercados volátiles como el de las criptomonedas.

Samaddar et al., 2021 [4] presentó un estudio comparativo de diferentes algoritmos de aprendizaje automático para la predicción del precio de Bitcoin, incluida la regresión lineal, los árboles de decisión y las redes neuronales. Su análisis concluyó que los modelos de aprendizaje automático, en particular los Random Forests y los métodos de aumento de gradiente, superaron a los enfoques estadísticos tradicionales en la predicción del precio de Bitcoin.

Gurupradeep et al., 2024 [5] emplearon técnicas de aprendizaje automático como el Random Forestt y la regresión de vectores de soporte (SVR) para la predicción del precio de las criptomonedas. Analizaron cómo la selección de características y el ajuste de hiperparámetros podrían afectar significativamente la precisión de la predicción, ofreciendo nuevos conocimientos para mejorar la solidez de los modelos de aprendizaje automático para los mercados de criptomonedas.

Estos estudios enfatizan el papel creciente de las técnicas de aprendizaje automático, especialmente los métodos de conjunto y los modelos híbridos, en la predicción del precio de las criptomonedas. Si bien los métodos estadísticos tradicionales todavía se utilizan comúnmente, los avances recientes en el aprendizaje automático han demostrado que son muy eficaces para capturar la naturaleza no lineal y volátil de los precios de las criptomonedas.

III. Metodología

A. Recopilación y preprocesamiento de datos

El estudio comienza con la recopilación de datos históricos de precios de Bitcoin mediante la API de Python-binance, que obtiene datos de BTC desde enero de 2023 hasta el 2 de noviembre de 2024, con intervalos de 5 minutos. El conjunto de datos incluye los siguientes atributos:

- **Precio de apertura**
- **Precio de cierre**
- **Precio alto**
- **Precio bajo**
- **Volumen de operaciones diarias**

Los datos se cargan en un entorno de Jupyter Notebook que se ejecuta localmente. Se aplican técnicas de preprocesamiento estándar:

1) **Manejo de valores faltantes:** Cualquier valor faltante o anómalo se identifica y se soluciona mediante imputación o eliminación.
2) **Ingeniería de características:** Se obtienen nuevas características, entre ellas:
   - Promedios móviles (MA) para intervalos de 5, 30 y 90 días.
   - Promedios móviles exponenciales (EMA) para intervalos similares.
   - Índice de fuerza relativa (RSI).
   - Bandas de Bollinger (banda superior, inferior y media).
3) **Etiquetado de tendencias:** Cada punto de datos se etiqueta según la lógica de MA y EMA para clasificar la tendencia como ascendente, descendente o lateral.

B. Desarrollo de modelos

Se entrenan y prueban tres modelos de aprendizaje automático (**XGBoost**, **Regresión Logística** y **Random Forest**) con los datos procesados. Cada modelo se evalúa con dos configuraciones:

1) **Sin optimización de parámetros:** Los modelos se inicializan con sus configuraciones predeterminadas.
2) **Con optimización de parámetros:** Los hiperparámetros de cada modelo se optimizan mediante **Búsqueda en cuadrícula** o **Búsqueda aleatoria** para maximizar el rendimiento.

Se siguen los siguientes pasos para ambas configuraciones:

1) **División de entrenamiento y prueba:** El conjunto de datos se divide en subconjuntos de entrenamiento (80 % para entrenar los datos y 20% para pruebas).
2) **Entrenamiento del modelo:** El subconjunto de entrenamiento se utiliza para ajustar cada modelo.
3) **Evaluación del modelo:** Las predicciones sobre el conjunto de prueba se evalúan utilizando las siguientes métricas:
   - **Accuracy**
   - **Precision**
   - **Recall**
   - **F1 Score**
   - **Matriz de confusión**

C. Formulación y análisis de la estrategia

El resultado predictivo de cada modelo se integra en un marco de estrategia que analiza tendencias, acciones y montos de inversión según las siguientes reglas:

1) **Tendencia ascendente:**
   - **Fuerte:** EMA de corto plazo $>$ EMA de mediano plazo $>$ EMA de largo plazo $\rightarrow$ Comprar (*all*).
   - **Moderado:** La media móvil de corto plazo cruza por debajo de la media móvil de mediano plazo,

que está por encima de la media móvil de largo plazo → Comprar (*mitad*).

2) **Tendencia lateral:** Mantener (*mitad*).
3) **Tendencia bajista:**
   - **Moderado:** Media móvil de corto plazo > Media móvil de mediano plazo < Media móvil de largo plazo → Vender (*mitad*).
   - **Fuerte:** Media móvil de corto plazo < Media móvil de mediano plazo < Media móvil de largo plazo → Vender (*todas*).

Además, se incorporan **RSI** y **Bandas de Bollinger** para refinar estas decisiones. Por ejemplo:

- Los valores altos de RSI (>70) pueden indicar condiciones de sobrecompra, lo que sugiere una posible acción de venta.
- Los valores bajos de RSI (<30) pueden indicar condiciones de sobreventa, lo que sugiere una posible acción de compra.
- El posicionamiento de la banda de Bollinger (por encima o por debajo de la banda) ayuda a detectar cambios en la volatilidad.

D. Análisis comparativo

Después de desarrollar y probar ambas configuraciones (con y sin optimización de parámetros) para los tres modelos, se realizan los siguientes análisis:

1) **Métricas de rendimiento del modelo:**
   - Compare las puntuaciones de accuracy, precision, recall y F1 para identificar el modelo con mejor rendimiento.
   - Evalúe el impacto de la optimización de parámetros en estas métricas.
2) **Análisis de estrategia:**
   - Examine todas las combinaciones posibles de tendencia, acción y cantidad predichas por cada modelo.
   - Calcule la frecuencia y precisión de cada combinación para identificar patrones o sesgos.
   - Compare la efectividad general de las estrategias en los tres modelos en términos de rentabilidad y consistencia.

E. Herramientas y entorno

- **Jupyter Notebook:** El análisis y el modelado se implementan localmente.
- **Bibliotecas de Python:**
  - **Procesamiento de datos:** Pandas, NumPy, Scikit-learn.
  - **Visualización:** Matplotlib, Seaborn.
  - **Aprendizaje automático:** XGBoost, Regresión Logística (de Scikit-learn), Random Forest (de Scikit-learn).
  - **Optimización de parámetros:** GridSearchCV y RandomizedSearchCV de Scikit-learn.
- **Métricas de evaluación:** Proporcionadas por el módulo `metrics` de Scikit-learn.

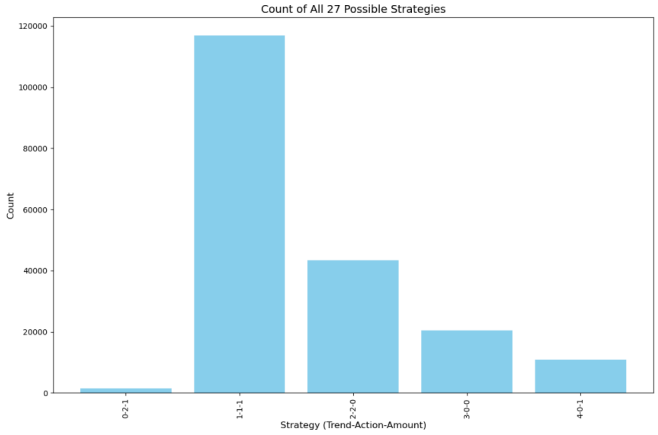
F. Resultados esperados

Esta metodología garantiza un análisis integral de las tendencias de precios de Bitcoin, las estrategias comerciales y el rendimiento del modelo, lo que brinda información útil sobre la aplicación del aprendizaje automático para el comercio de criptomonedas. Al comparar los modelos optimizados y no optimizados, el estudio destaca la importancia del ajuste de hiperparámetros para lograr predicciones sólidas y estrategias efectivas.

IV. Resultados

A. Introducción a los resultados

Se evaluó el rendimiento de varios modelos en las clasificaciones *Trend*, *Action* y *Amount*, considerando tanto la configuración de referencia (no optimizada) como la optimizada por parámetros. Las métricas evaluadas incluyen accuracy, precision, recall y F1 y matrices de confusión. A continuación, proporcionamos un análisis en profundidad de los resultados de cada modelo, junto con una discusión de las tendencias y las compensaciones observadas en su rendimiento. Adicionalmente este gráfico muestra en la información utilizada las mayores distribuciones en términos de las combinaciones de las posibles estrategias del tiempo en estudio:



Como se observa del gráfico, ya removiendo la codificación de las etiquetas obtenemos los siguientes números:

Estrategias	**Conteo**
sideways-hold-half	116,983
strong_downward-sell-all	43,384
strong_upward-buy-all	20,401
upward-buy-half	10,897
downward-sell-half	1,567

TABLE I: Conteos de las estrategias

B. Resultados de XGBoost

1) (a) Sin optimización de parámetros: **Clasificación de tendencias:**

- **Métricas:**
  - Accuracy: $0,7061$
  - Precision: $0,6729$
  - Recall: $0,7061$
  - F1 Score: $0,6450$
- **Matriz de confusión:**

$$\begin{bmatrix} 0 & 217 & 12 & 0 & 0 \\ 0 & 15522 & 1912 & 28 & 62 \\ 0 & 1657 & 4851 & 0 & 8 \\ 0 & 3073 & 5 & 40 & 0 \\ 0 & 1112 & 431 & 0 & 52 \end{bmatrix}$$

Análisis: XGBoost logra valores moderados de recuperación y precisión. Sin embargo, la matriz de confusión muestra importantes errores de clasificación, en particular en elementos fuera de la diagonal, donde ciertas clases (por ejemplo, las clases 2 y 3) tienen altos recuentos de errores de clasificación. Esto sugiere que hay margen para mejorar el manejo de casos extremos y muestras ambiguas.

Clasificación de acciones:

- **Métricas:**
  - Accuracy: $0,7064$
  - Precision: $0,6771$
  - Recall: $0,7064$
  - F1 Score: $0,6487$
- **Matriz de confusión:**

$$\begin{bmatrix} 101 & 4196 & 416 \\ 84 & 15480 & 1960 \\ 7 & 1845 & 4893 \end{bmatrix}$$

Análisis: El modelo logra una precisión comparable y recuperación de *Trend*. Hay un mejor rendimiento en la clasificación de clases dominantes, pero hay dificultades en las clases poco frecuentes. El equilibrio entre precisión y recuperación indica un equilibrio equilibrado, aunque la optimización podría mejorar la detección de clases menores.

Clasificación de cantidad:

- **Métricas:**
  - Accuracy: $0,7548$
  - Precision: $0,7476$
  - Recall: $0,7548$
  - F1 Score: $0,7374$
- **Matriz de confusión:**

$$\begin{bmatrix} 4304 & 5330 \\ 1775 & 17573 \end{bmatrix}$$

Análisis: La clasificación *Cantidad* logra métricas generales más altas en comparación con *Tendencia* y *Acción*. Esto indica que el límite de decisión para la tarea es más claro. Sin embargo, existen errores de clasificación significativos entre las dos clases (como se observa en la matriz de confusión).

2) (b) Con optimización de parámetros: **Clasificación de tendencias:**

- **Métricas:**
  - Accuracy: $0,7123$
  - Precision: $0,6681$
  - Recall: $0,7123$
  - F1 Score: $0,6611$
- **Matriz de confusión:**

$$\begin{bmatrix} 1 & 213 & 14 & 1 & 0 \\ 6 & 15456 & 1735 & 173 & 154 \\ 2 & 1575 & 4910 & 2 & 27 \\ 0 & 2962 & 5 & 145 & 6 \\ 0 & 1148 & 315 & 0 & 132 \end{bmatrix}$$

Análisis: La optimización de los parámetros mejoró ligeramente la precisión ($+0,62\%$). A pesar de esto, la precisión mostró una pequeña caída, lo que indica que el ajuste de los parámetros mejoró la recuperación a expensas de la precisión. Esto muestra que el proceso de optimización se centró más en reducir los falsos negativos.

Se observaron tendencias similares para la clasificación de *Action* y *Amount*, con mejoras de precisión que oscilaron entre $+0,29\%$ y $+0,75\%$. Las matrices de confusión mostraron ligeras reducciones en las clasificaciones erróneas fuera de la diagonal.

C. Resultados de Random Forest

1) Sin Optimización: En cuanto a la predicción de tendencias, el modelo Random Forest logró un rendimiento moderado, como lo indican sus métricas de exactitud y precisión. La matriz de confusión sugiere que el modelo tuvo dificultades para clasificar correctamente ciertas categorías de tendencias, en particular las clases minoritarias, lo que generó desequilibrios en la recuperación.

- **Accuracy:** 0.7098
- **Precision:** 0.6607
- **Recall:** 0.7098
- **F1-Score:** 0.6636

Confusion matrix:

$$\begin{bmatrix} 0 & 289 & 22 & 1 & 1 \\ 4 & 20498 & 2144 & 434 & 277 \\ 0 & 2219 & 6456 & 0 & 41 \\ 0 & 3790 & 11 & 324 & 6 \\ 0 & 1637 & 334 & 6 & 149 \end{bmatrix}$$

En cuanto a la predicción de acciones, si bien la exactitud y precisión generales del modelo son razonables, hay clasificaciones erróneas notables entre las categorías, con una

tendencia a predecir en exceso la clase mayoritaria.

- **Accuracy:** 0.7077
- **Precision:** 0.6648
- **Recall:** 0.7077
- **F1-Score:** 0.6670

Confusion matrix:

$$\begin{bmatrix} 533 & 5350 & 374 \\ 831 & 20314 & 2212 \\ 40 & 2488 & 6501 \end{bmatrix}$$

En cuanto a la predicción de cantidades, el modelo mostró un rendimiento relativamente mejor, con precisión y recuperación equilibradas. Sin embargo, todavía hay margen de mejora en el manejo de clasificaciones erróneas entre valores altos y bajos:

- **Accuracy:** 0.7822
- **Precision:** 0.7781
- **Recall:** 0.7822
- **F1-Score:** 0.7698

Confusion matrix:

$$\begin{bmatrix} 6619 & 6228 \\ 2187 & 23609 \end{bmatrix}$$

2) Con Optimización de parámetros: En cuanto a la predicción de tendencias, el modelo de Bosque aleatorio optimizado mostró ligeras disminuciones en la precisión y exactitud en comparación con el modelo no optimizado. Esto sugiere que el ajuste de parámetros no abordó de manera efectiva los desequilibrios en las predicciones de clases minoritarias y puede haber reducido inadvertidamente las capacidades de generalización del modelo:

- **Accuracy:** 0.7007
- **Precision:** 0.6501
- **Recall:** 0.7007
- **F1-Score:** 0.6406

Confusion matrix:

$$\begin{bmatrix} 0 & 298 & 15 & 0 & 0 \\ 0 & 20484 & 2711 & 69 & 93 \\ 0 & 2226 & 6478 & 0 & 12 \\ 0 & 4066 & 8 & 55 & 2 \\ 0 & 1430 & 635 & 0 & 61 \end{bmatrix}$$

En cuanto a la predicción de acciones, el proceso de optimización resultó en un desempeño general similar en comparación con el modelo no optimizado. La matriz de confusión muestra clasificaciones erróneas persistentes en categorías minoritarias, lo que indica que el proceso de optimización no logró mejorar significativamente el poder discriminatorio del modelo:

- **Accuracy:** 0.7015
- **Precision:** 0.6499
- **Recall:** 0.7015
- **F1-Score:** 0.6448

Confusion matrix:

$$\begin{bmatrix} 121 & 5493 & 643 \\ 188 & 20454 & 2715 \\ 14 & 2482 & 6533 \end{bmatrix}$$

En cuanto a la predicción de cantidades, el ajuste de parámetros provocó disminuciones marginales en las métricas, lo que destaca que la optimización puede no haber sido efectiva para este conjunto de datos. A pesar de esto, el desempeño general del modelo sigue siendo moderadamente sólido, con precisión y recuperación relativamente equilibradas:

- **Accuracy:** 0.7667
- **Precision:** 0.7614
- **Recall:** 0.7667
- **F1-Score:** 0.7509

Confusion matrix:

$$\begin{bmatrix} 6048 & 6799 \\ 2217 & 23579 \end{bmatrix}$$

D. Logistic Regression

1) Sin Optimización: En la predicción de tendencias, la regresión logística mostró un rendimiento inferior al de Random Forest, con importantes errores de clasificación en las clases minoritarias. La matriz de confusión destaca que el modelo se basa principalmente en predecir en exceso las categorías dominantes, lo que da como resultado una mala recuperación para las clases más pequeñas:

- **Accuracy:** 0.6166
- **Precision:** 0.5413
- **Recall:** 0.6166
- **F1-Score:** 0.5334

Confusion matrix:

$$\begin{bmatrix} 0 & 311 & 2 & 0 & 0 \\ 0 & 21657 & 1187 & 513 & 0 \\ 0 & 6686 & 2030 & 0 & 0 \\ 0 & 3992 & 0 & 139 & 0 \\ 0 & 2116 & 0 & 10 & 0 \end{bmatrix}$$

En cuanto a la predicción de acciones, los resultados siguen una tendencia similar, ya que el modelo tiene dificultades para diferenciar entre determinadas categorías. Esto subraya aún más las limitaciones de la regresión logística para capturar

patrones complejos en este conjunto de datos:

- **Accuracy:** 0.6154
- **Precision:** 0.5646
- **Recall:** 0.6154
- **F1-Score:** 0.5450

Confusion matrix:

$$\begin{bmatrix} 261 & 5996 & 0 \\ 770 & 21156 & 1431 \\ 0 & 6667 & 2362 \end{bmatrix}$$

En cuanto a la predicción de cantidades, el rendimiento del modelo es ligeramente mejor, pero sigue estando por detrás de Random Forest. La matriz de confusión revela una tendencia a predecir en exceso determinados valores, lo que genera desequilibrios en la precisión y la recuperación:

- **Accuracy:** 0.6869
- **Precision:** 0.6787
- **Recall:** 0.6869
- **F1-Score:** 0.6053

Confusion matrix:

$$\begin{bmatrix} 1552 & 11295 \\ 805 & 24991 \end{bmatrix}$$

2) Con Optimización de Parámetros: En cuanto a la predicción de tendencias, el modelo de regresión logística optimizado muestra ligeras mejoras con respecto al modelo no optimizado. La mayor precisión y la puntuación f1 sugieren que el ajuste de parámetros ayudó al modelo a capturar mejor los patrones subyacentes en los datos. Sin embargo, la matriz de confusión aún destaca los desafíos para clasificar las clases minoritarias de manera efectiva:

- **Accuracy:** 0.6569
- **Precision:** 0.5860
- **Recall:** 0.6569
- **F1-Score:** 0.5887

Confusion matrix:

$$\begin{bmatrix} 0 & 310 & 3 & 0 & 0 \\ 0 & 21360 & 1735 & 211 & 51 \\ 0 & 4773 & 3943 & 0 & 0 \\ 0 & 4059 & 1 & 71 & 0 \\ 0 & 2020 & 97 & 0 & 9 \end{bmatrix}$$

En la predicción de acciones, la optimización generó mejoras moderadas en las métricas de rendimiento, lo que indica un mejor manejo de las complejidades del conjunto de datos. No obstante, las clasificaciones erróneas siguen siendo un problema, en particular para las categorías con una representación menor:

- **Accuracy:** 0.6529
- **Precision:** 0.5940
- **Recall:** 0.6529
- **F1-Score:** 0.5841

Confusion matrix:

$$\begin{bmatrix} 78 & 6155 & 24 \\ 242 & 21486 & 1629 \\ 0 & 5363 & 3666 \end{bmatrix}$$

En cuanto a la predicción de cantidades, el modelo optimizado demuestra pequeñas mejoras en la precisión y la recuperación, lo que indica algún beneficio del ajuste de parámetros. Sin embargo, el modelo aún tiene dificultades para alcanzar el nivel de rendimiento observado con Random Forest, lo que indica limitaciones inherentes en la regresión logística para esta tarea:

- **Accuracy:** 0.6839
- **Precision:** 0.6735
- **Recall:** 0.6839
- **F1-Score:** 0.5980

Confusion matrix:

$$\begin{bmatrix} 1380 & 11467 \\ 748 & 25048 \end{bmatrix}$$

E. Análisis general del rendimiento

Al comparar el rendimiento general de los tres modelos (XGBoost, Random Forest y Regresión Logística), es evidente que los modelos basados en árboles (XGBoost y Random Forest) superan consistentemente a la Regresión Logística en todas las tareas. Esto problabemente se deba a su capacidad para modelar relaciones complejas y no lineales en los datos, lo que es una limitación para la Regresión Logística.

Observaciones clave:

- **Precisión y robustez:** Tanto XGBoost como Random Forest logran una mayor precisión en las clasificaciones *Trend*, *Action* y *Amount*, y Random Forest tiene un rendimiento ligeramente mejor para las tareas binarias (*Amount*) y XGBoost se destaca en las tareas de múltiples clases (*Trend* y *Action*).
- **Impacto de la optimización de parámetros:** En el caso de XGBoost, la optimización de parámetros proporcionó ganancias modestas en precisión y recuperación, especialmente para las clasificaciones *Trend* y *Amount*. Random Forest, si bien no se optimizó en cuanto a parámetros en este análisis, tuvo un desempeño competitivo, lo que resalta su solidez incluso con configuraciones predeterminadas.
- **Equilibrio entre precisión y recuperación:** XGBoost mostró un mejor equilibrio entre precisión y recuperación, lo que es crucial para minimizar tanto los falsos positivos como los falsos negativos. La Regresión Logística, por el

contrario, exhibió una caída notable en la recuperación, lo que puede resultar en una mayor tasa de falsos negativos.

- **Escalabilidad e interpretabilidad:** Si bien Random Forest y XGBoost requieren un uso intensivo de recursos computacionales, brindan mayor precisión y solidez, lo que los hace adecuados para escenarios donde el rendimiento es una prioridad. La regresión logística, al ser más simple y rápida de calcular, puede seguir siendo una opción viable para situaciones en las que la interpretabilidad y la velocidad son primordiales.

Resumen final: En conclusión, Random Forest demostró un sólido desempeño en clasificaciones binarias, mientras que XGBoost emergió como el modelo de mayor desempeño para tareas de múltiples clases, particularmente cuando se optimizó. La regresión logística, a pesar de haber sido superada, sigue siendo un punto de referencia útil y puede servir como modelo de referencia para casos de uso más simples. La elección del modelo depende en última instancia de la tarea de clasificación específica, las restricciones computacionales y la necesidad de interpretabilidad versus precisión.

V. Trabajo Futuro

Si bien el estudio actual brinda información valiosa sobre el rendimiento de XGBoost, Random Forest y la regresión logística en las tareas dadas, existen varias vías para la exploración futura que podrían mejorar la profundidad y la confiabilidad de los hallazgos.

1. Diseño experimental ampliado: Con tiempo adicional, sería beneficioso realizar experimentos más completos para probar los modelos en una variedad más amplia de configuraciones. Esto podría incluir la exploración de técnicas alternativas de ajuste de hiperparámetros, la variación de las estrategias de ingeniería de características y la prueba de métricas de evaluación adicionales para capturar mejor el comportamiento del modelo en diferentes escenarios.

2. Comparación estadística mediante ANOVA: Para proporcionar una base estadística más rigurosa para comparar el rendimiento del modelo, habría sido útil aplicar pruebas de análisis de varianza (ANOVA). ANOVA permitiría una comparación detallada de las métricas de rendimiento promedio en los modelos, determinando si las diferencias observadas son estadísticamente significativas. Este enfoque fortalecería las conclusiones sobre qué modelo funciona mejor en las condiciones dadas.

3. Optimización mejorada de modelos: Explorar técnicas de optimización más avanzadas, podría mejorar aún más la precisión de los modelos. Estas técnicas podrían descubrir configuraciones que mejoren significativamente el poder predictivo de los modelos, en particular para tareas desafiantes como la clasificación de múltiples clases.

4. Incorporación de características adicionales: El trabajo futuro podría implicar la incorporación de características más relevantes o el uso de métodos de selección de características para refinar los datos de entrada. Esto podría reducir potencialmente el ruido y mejorar la precisión general de los modelos.

5. Evaluación en diversos conjuntos de datos: La solidez y la generalización de los modelos podrían evaluarse aplicándolos a diferentes conjuntos de datos con diferentes distribuciones y complejidades. Esto proporcionaría evidencia adicional de su aplicabilidad a escenarios del mundo real.

En conclusión, si bien el trabajo actual sienta una base sólida, las direcciones futuras propuestas ofrecen oportunidades interesantes para mejorar el rendimiento del modelo, mejorar el rigor estadístico y ampliar la aplicabilidad de los hallazgos.

VI. Conclusión

Este estudio investigó el desempeño de tres modelos de aprendizaje automático (XGBoost, Random Forest y Regresión logística) en tres tareas distintas: predicción de tendencias, clasificación de acciones y estimación de cantidades. Los resultados demostraron las fortalezas y limitaciones de cada modelo, tanto con optimización de hiperparámetros como sin ella. XGBoost surgió como una opción sólida, con un desempeño consistentemente bueno en todas las tareas, especialmente después del ajuste de parámetros, mientras que Random Forest mostró resultados sólidos para la estimación de cantidades. La regresión logística, aunque menos competitiva en general, proporcionó una base útil para la comparación.

Los experimentos destacaron el impacto de la optimización de hiperparámetros, con mejoras en la accuracy, precision, recall y F1 Score observados en todos los modelos. Sin embargo, la efectividad de la optimización varió, y XGBoost fue el que se benefició más significativamente. Además, el análisis reveló que la complejidad de la tarea y el desequilibrio de los datos tuvieron efectos notables en el desempeño del modelo, lo que subraya la importancia de un preprocesamiento cuidadoso y la selección de características en estudios futuros.

A pesar de estos hallazgos, hay margen de mejora. El estudio se habría beneficiado de un análisis estadístico más profundo, como ANOVA, para proporcionar comparaciones más sólidas del rendimiento del modelo. Además, experimentos adicionales, conjuntos de datos más amplios y estrategias de optimización alternativas podrían mejorar la eficacia y la generalización de los modelos.

En conclusión, esta investigación demuestra el potencial de los modelos de aprendizaje automático para tareas predictivas, y XGBoost se destaca como un algoritmo versátil y de alto rendimiento. El trabajo futuro debe centrarse en el rigor estadístico, la experimentación adicional y las técnicas de optimización avanzadas para refinar aún más estos modelos y maximizar sus capacidades predictivas.

References

[1] Hafid, Abdelatif & Ebrahim, Maad & Alfatemi, Ali & Rahouti, Mohamed & Oliveira, Diogo. (2024). Cryptocurrency Price Forecasting Using XGBoost Regressor and Technical Indicators. 10.48550/arXiv.2407.11786.

[2] Qinghe Li. 2021. Predicting Trends of Bitcoin Prices Based on Machine Learning Methods. In Proceedings of the 2020 4th International Conference on Software and e-Business (ICSEB '20). Association for Computing Machinery, New York, NY, USA, 49–52. https://doi.org/10.1145/3446569.3446588

[3] Nikola Gradojevic, Dragan Kukolj, Robert Adcock, Vladimir Djakovic, "Forecasting Bitcoin with technical analysis: A not-so-random forest?", *International Journal of Forecasting,* Volume 39, Issue 1, 2023.

[4] Samaddar, Mayukh & Roy, Rishiraj & De, Sayantani & Karmakar, Raja. (2021). A Comparative Study of Different Machine Learning Algorithms on Bitcoin Value Prediction. 10.1109/ICAECT49130.2021.9392629.

[5] G, Gurupradeep & M, Harishvaran & K, Amsavalli. (2023). Cryptocurrency Price Prediction using Machine Learning. IJARCCE. 12. 10.17148/IJARCCE.2023.124140.