

BTC_Financial_Analysis

September 14, 2024

```
[2]: pip install pandas matplotlib seaborn scikit-learn fpdf
```

```
Requirement already satisfied: pandas in
/home/zatiel/anaconda3/lib/python3.12/site-packages (2.2.2)
Requirement already satisfied: matplotlib in
/home/zatiel/anaconda3/lib/python3.12/site-packages (3.8.4)
Requirement already satisfied: seaborn in
/home/zatiel/anaconda3/lib/python3.12/site-packages (0.13.2)
Requirement already satisfied: scikit-learn in
/home/zatiel/anaconda3/lib/python3.12/site-packages (1.4.2)
Collecting fpdf
  Downloading fpdf-1.7.2.tar.gz (39 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy>=1.26.0 in
/home/zatiel/anaconda3/lib/python3.12/site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
/home/zatiel/anaconda3/lib/python3.12/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
/home/zatiel/anaconda3/lib/python3.12/site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in
/home/zatiel/anaconda3/lib/python3.12/site-packages (from pandas) (2023.3)
Requirement already satisfied: contourpy>=1.0.1 in
/home/zatiel/anaconda3/lib/python3.12/site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
/home/zatiel/anaconda3/lib/python3.12/site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/home/zatiel/anaconda3/lib/python3.12/site-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/home/zatiel/anaconda3/lib/python3.12/site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in
/home/zatiel/anaconda3/lib/python3.12/site-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=8 in
/home/zatiel/anaconda3/lib/python3.12/site-packages (from matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/home/zatiel/anaconda3/lib/python3.12/site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: scipy>=1.6.0 in
/home/zatiel/anaconda3/lib/python3.12/site-packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in
```

```

/home/zatiel/anaconda3/lib/python3.12/site-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/home/zatiel/anaconda3/lib/python3.12/site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: six>=1.5 in
/home/zatiel/anaconda3/lib/python3.12/site-packages (from python-
dateutil>=2.8.2->pandas) (1.16.0)
Building wheels for collected packages: fpdf
  Building wheel for fpdf (setup.py) ... done
  Created wheel for fpdf: filename=fpdf-1.7.2-py2.py3-none-any.whl
size=40702
sha256=420e4fb470c41aa4965211b3631f19d32724dc4854dbd1c84b564e048535f6a6
  Stored in directory: /home/zatiel/.cache/pip/wheels/6e/62/11/dc73d78e40a218ad5
2e7451f30166e94491be013a7850b5d75
Successfully built fpdf
Installing collected packages: fpdf
Successfully installed fpdf-1.7.2
Note: you may need to restart the kernel to use updated packages.

```

```

[9]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# 1. Cargamos el dataset
url = 'https://www.cryptodatadownload.com/cdd/Binance_BTCUSDT_d.csv'
df = pd.read_csv(url, skiprows=1)

# Imprimimos las primeras filas del DataFrame y las columnas
print(df.head())
print("Número de columnas en el DataFrame:", len(df.columns))
print("Columnas en el DataFrame:", df.columns.tolist())

# 2. Renombramos las columnas para simplificar el manejo de datos
new_column_names = ['unix', 'date', 'symbol', 'open', 'high', 'low', 'close', 'volume_btc', 'volume_usd', 'tradedcount']
df.columns = new_column_names

# Convertimos la columna de fechas a formato datetime
df['date'] = pd.to_datetime(df['date'])

# Filtramos las columnas importantes
try:
    df_clean = df[['date', 'close', 'volume_usd']]
except KeyError as e:
    print(f"Error: No se encontró una de las columnas: {e}")

```

```

# 3. Graficamos la tendencia de precios de cierre de Bitcoin
plt.figure(figsize=(12, 6))
plt.plot(df_clean['date'], df_clean['close'], label='Precio de cierre',
        color='blue')
plt.title('Tendencia del Precio de Cierre de Bitcoin')
plt.xlabel('Fecha')
plt.ylabel('Precio (USD)')
plt.legend()
plt.show()

# Gráfico avanzado: Distribución de precios de cierre
plt.figure(figsize=(10, 5))
sns.histplot(df_clean['close'], bins=50, color='purple', kde=True)
plt.title('Distribución del Precio de Cierre de Bitcoin')
plt.show()

# Gráfico avanzado: Volumen de transacciones (USD) a lo largo del tiempo
plt.figure(figsize=(12, 6))
plt.plot(df_clean['date'], df_clean['volume_usd'], label='Volumen (USD)',
        color='green')
plt.title('Volumen de Transacciones de Bitcoin (USD)')
plt.xlabel('Fecha')
plt.ylabel('Volumen (USD)')
plt.legend()
plt.show()

# 4. Limpieza de datos: Eliminamos duplicados y revisamos valores nulos
df_clean.drop_duplicates(inplace=True)
df_clean = df_clean.dropna()

# 5. Preparamos los datos para el modelo predictivo
X = df_clean[['volume_usd']] # Volumen como predictor
y = df_clean['close'] # Precio de cierre como variable objetivo

# Dividimos los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
        random_state=42)

# Creamos el modelo de regresión lineal
model = LinearRegression()
model.fit(X_train, y_train)

# Hacemos predicciones con el conjunto de prueba
y_pred = model.predict(X_test)

# Evaluamos el modelo

```

```

mse = mean_squared_error(y_test, y_pred)
print(f"Error cuadrático medio (MSE): {mse:.2f}")

# Graficamos las predicciones frente a los valores reales
plt.figure(figsize=(10, 5))
plt.scatter(y_test, y_pred, color='blue', edgecolor='k', alpha=0.7)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()],
         color='red', lw=2)
plt.title('Predicción vs Valor Real (Regresión Lineal)')
plt.xlabel('Valor Real')
plt.ylabel('Predicción')
plt.show()

```

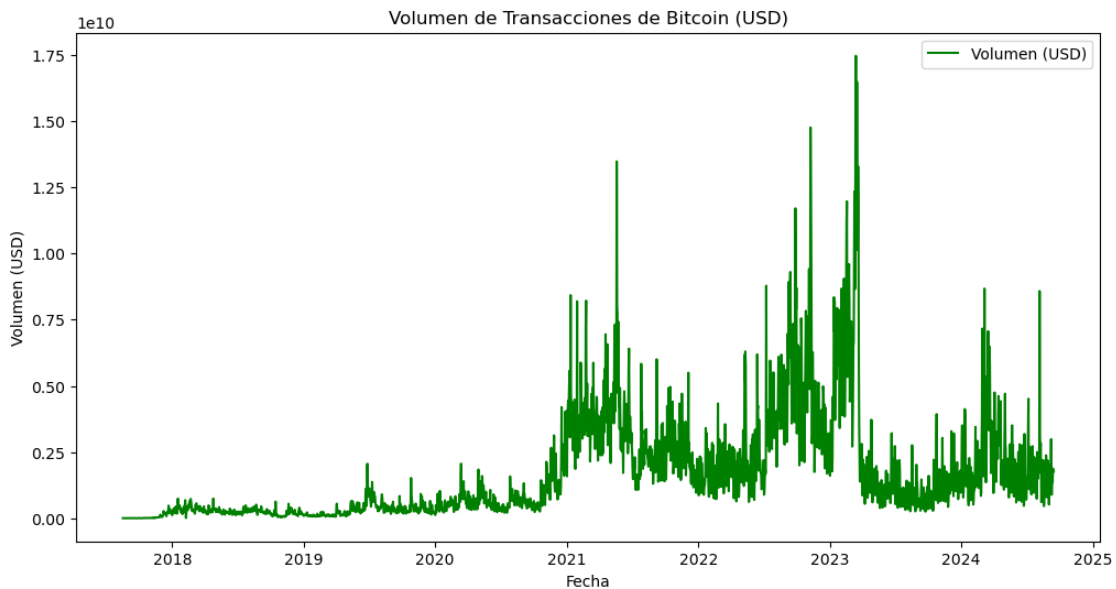
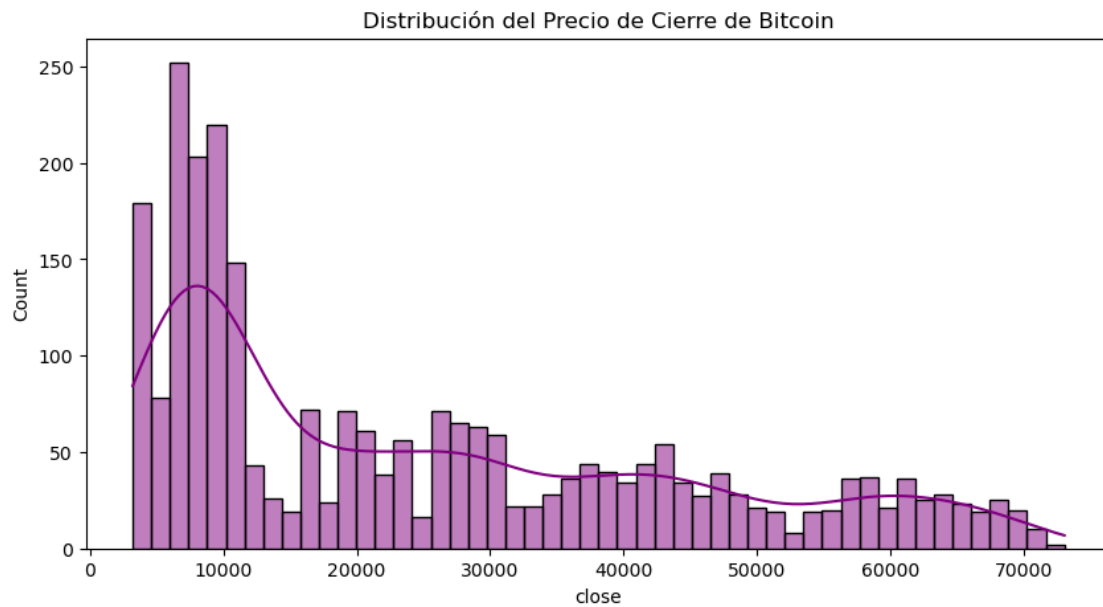
	Unix	Date	Symbol	Open	High	Low	Close \
0	1726185600000	2024-09-13	BTCUSDT	58132.31	60625.00	57632.62	60498.00
1	1726099200000	2024-09-12	BTCUSDT	57338.00	58588.00	57324.00	58132.32
2	1726012800000	2024-09-11	BTCUSDT	57635.99	57981.71	55545.19	57338.00
3	1725926400000	2024-09-10	BTCUSDT	57042.01	58044.36	56386.40	57635.99
4	1725840000000	2024-09-09	BTCUSDT	54869.95	58088.00	54591.96	57042.00

	Volume BTC	Volume USDT	tradedcount
0	29825.23333	1.760672e+09	3378012
1	31074.40631	1.802849e+09	3706764
2	33026.56757	1.875739e+09	4045103
3	23626.78126	1.349365e+09	2843148
4	32384.51737	1.809715e+09	3355912

Número de columnas en el DataFrame: 10

Columnas en el DataFrame: ['Unix', 'Date', 'Symbol', 'Open', 'High', 'Low', 'Close', 'Volume BTC', 'Volume USDT', 'tradedcount']

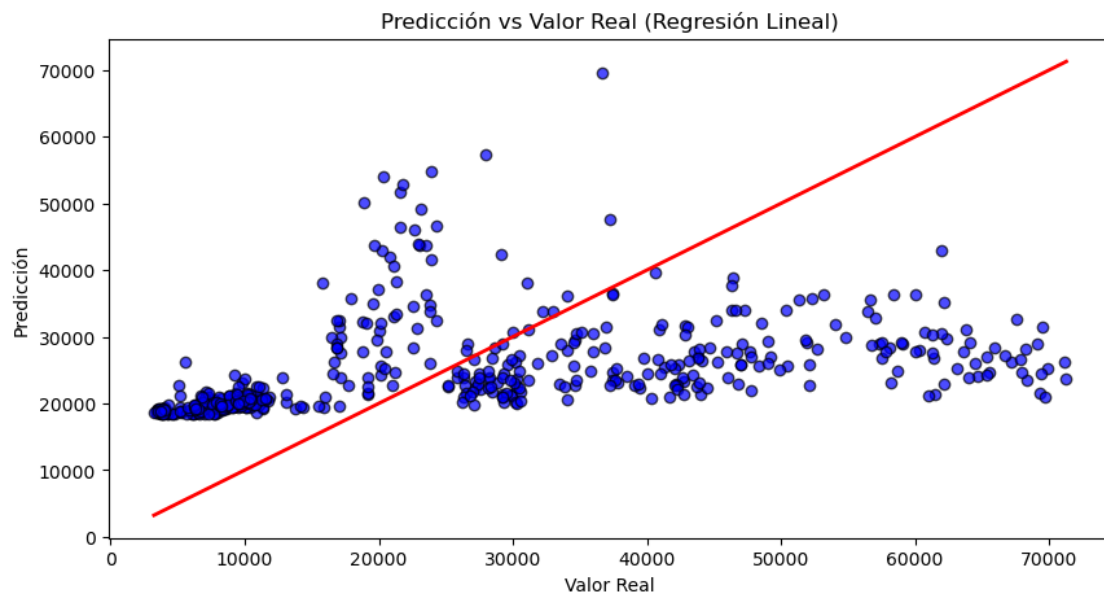




/tmp/ipykernel_11568/2290526499.py:55: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
df_clean.drop_duplicates(inplace=True)

Error cuadrático medio (MSE): 296929021.51



[]: