TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HỒ CHÍ MINH
VIỆN CÔNG NGHỆ THÔNG TIN VÀ ĐIỆN, ĐIỆN TỬ

-----□&□-----

UTH
UNIVERSITY
OF TRANSPORT
HOCHIMINH CITY

BÁO CÁO MÔN HỌC

NGÔN NGỮ LẬP TRÌNH PYTHON

TÊN ĐỀ TÀI

Ứng dụng LSTM trong dự báo giá và xu hướng biến động của Bitcoin

Lớp : **7480201190360**

Người hướng dẫn : **ThS. Huỳnh Thanh Việt**

Sinh viên thực hiện: : **Lê Hoàng Tuấn**

Mã sinh viên : **079204034414**

Thành phố Hồ Chí Minh, năm 2025

LỜI CẢM ƠN

Nhóm chúng em xin gửi lời cảm ơn chân thành đến Thầy Huỳnh Thanh Việt, giảng viên học phần Ngôn ngữ lập trình, người đã tận tình giảng dạy và truyền đạt cho em những kiến thức quý báu trong suốt quá trình học tập.

Những kiến thức mới mẻ và bổ ích từ Thầy đã giúp em hiểu sâu hơn về xử lý dữ liệu cũng như cách vận dụng trong thực tiễn để giải quyết các mô hình bài toán. Tuy nhiên, do kiến thức và kinh nghiệm thực hành còn hạn chế, việc áp dụng của em vẫn chưa thật sự tốt, và em sẽ tiếp tục nỗ lực để hoàn thiện hơn.

Em xin trân trọng cảm ơn!

Source Code: https://github.com/bush-le/btc-price-prediction-lstm

A. LỜI MỞ ĐẦU

Thị trường tiền điện tử, đặc biệt là Bitcoin (BTC), đã trở thành một lĩnh vực thu hút sự quan tâm lớn từ các nhà đầu tư, tổ chức tài chính và cộng đồng nghiên cứu. Với đặc tính biến động mạnh và khó dự đoán, giá Bitcoin chịu ảnh hưởng từ nhiều yếu tố như cung cầu thị trường, tin tức kinh tế, chính sách pháp lý và tâm lý nhà đầu tư. Do đó, việc xây dựng các mô hình dự báo không chỉ giúp nhà đầu tư đưa ra quyết định giao dịch hiệu quả, mà còn hỗ trợ quản lý rủi ro và phát triển chiến lược đầu tư bền vững.

Báo cáo này tập trung vào việc ứng dụng mô hình LSTM, một loại mạng nơ-ron tái hồi có khả năng xử lý và ghi nhớ thông tin trong chuỗi thời gian, để giải quyết hai nhiệm vụ quan trọng: (1) dự báo giá Bitcoin trong tương lai và (2) dự báo xu hướng biến động giá - tăng hoặc giảm dựa trên dữ liệu lịch sử được thu thập từ Binance. Đề tài sử dụng dữ liệu đa biến, kết hợp các đặc trưng giá, khối lượng giao dịch và chỉ báo kỹ thuật, nhằm khai thác tốt hơn các mẫu ẩn trong dữ liệu thị trường.

Báo cáo sẽ trình bày quy trình thu thập, tiền xử lý và phân tích dữ liệu, xây dựng và huấn luyện các mô hình LSTM cho cả hai bài toán, đồng thời so sánh, đánh giá kết quả để rút ra nhận xét cũng như đề xuất hướng tối ưu hóa nhằm nâng cao độ chính xác và khả năng ứng dụng thực tiễn.

MỤC LỤC

B. NỘI DUNG

1. TỔNG QUAN ĐỀ TÀI

1.1. Lý do chọn đề tài

Dự báo giá Bitcoin là một bài toán thách thức và có ý nghĩa lớn trong lĩnh vực tài chính số. Sự biến động mạnh của giá Bitcoin, cùng với các mẫu phi tuyến và phụ thuộc thời gian phức tạp, đòi hỏi các mô hình dự báo tiên tiến hơn so với các phương pháp truyền thống. Mô hình LSTM, với khả năng ghi nhớ các mẫu dài hạn trong chuỗi thời gian, là một giải pháp phù hợp để giải quyết bài toán dự báo giá Bitcoin.

Việc nghiên cứu và áp dụng LSTM không chỉ phục vụ mục tiêu dự báo giá trị tương lai của Bitcoin mà còn hỗ trợ dự báo xu hướng tăng hoặc giảm, từ đó mang lại giá trị thực tiễn trong việc hỗ trợ các nhà đầu tư, tổ chức tài chính và nhà hoạch định chiến lược đưa ra quyết định hiệu quả. Ngoài ra, đề tài còn góp phần khẳng định tiềm năng của mô hình LSTM trong phân tích và dự báo chuỗi thời gian tài chính, đặc biệt trong bối cảnh thị trường tiền điện tử luôn biến động mạnh. Nhận thấy tính cấp thiết và giá trị ứng dụng cao, nhóm quyết định lựa chọn và phát triển đề tài này.

1.2. Cơ sở lý thuyết LSTM

Long Short-Term Memory (LSTM) là một loại mạng nơ-ron hồi quy (RNN) được thiết kế để xử lý và dự đoán dữ liệu chuỗi thời gian, đặc biệt hiệu quả với các chuỗi có phụ thuộc dài hạn. LSTM được giới thiệu bởi Hochreiter và Schmidhuber (1997) nhằm khắc phục vấn đề tiêu biến gradient trong RNN truyền thống thông qua cơ chế cổng (gates).

Mỗi đơn vị LSTM bao gồm một ô nhớ (cell state) và ba cổng chính: cổng quên (loại bỏ thông tin không cần thiết), cổng vào (thêm thông tin mới), và cổng ra (quyết định đầu ra). Ô nhớ được cập nhật liên tục để ghi nhớ các thông tin quan trọng qua nhiều bước thời gian.

LSTM có các ưu điểm nổi bật: ghi nhớ phụ thuộc dài hạn, lọc bỏ nhiễu, và linh hoạt cho cả dự báo giá (hồi quy) và dự đoán xu hướng (phân loại).

Nhờ khả năng xử lý dữ liệu phi tuyến, giảm nhiễu và ghi nhớ dài hạn, LSTM là lựa chọn phù hợp để dự báo giá và xu hướng Bitcoin, giúp mô hình học được các mẫu giá và xu hướng phức tạp từ dữ liệu nến 15 phút.

1.3. Mục tiêu

- **Thu thập và làm sạch dữ liệu:** Sử dụng Python để xử lý dữ liệu, đảm bảo chất lượng dữ liệu cho các bước phân tích và dự báo.
- **Trực quan hóa dữ liệu:** Biểu diễn dữ liệu dưới dạng hình ảnh và biểu đồ để nhận biết xu hướng, mẫu hình và mối quan hệ trong dữ liệu.
- **Chuẩn hóa và mã hóa dữ liệu:** Chuyển đổi dữ liệu về dạng phù hợp với các thuật toán học máy.

- **Xây dựng mô hình học sâu:** Phát triển mô hình dự báo dựa trên dữ liệu đã xử lý.
- **Đánh giá và tối ưu hóa mô hình:** Sử dụng các chỉ số đánh giá và đề xuất các giải pháp cải thiện hiệu quả dự báo.

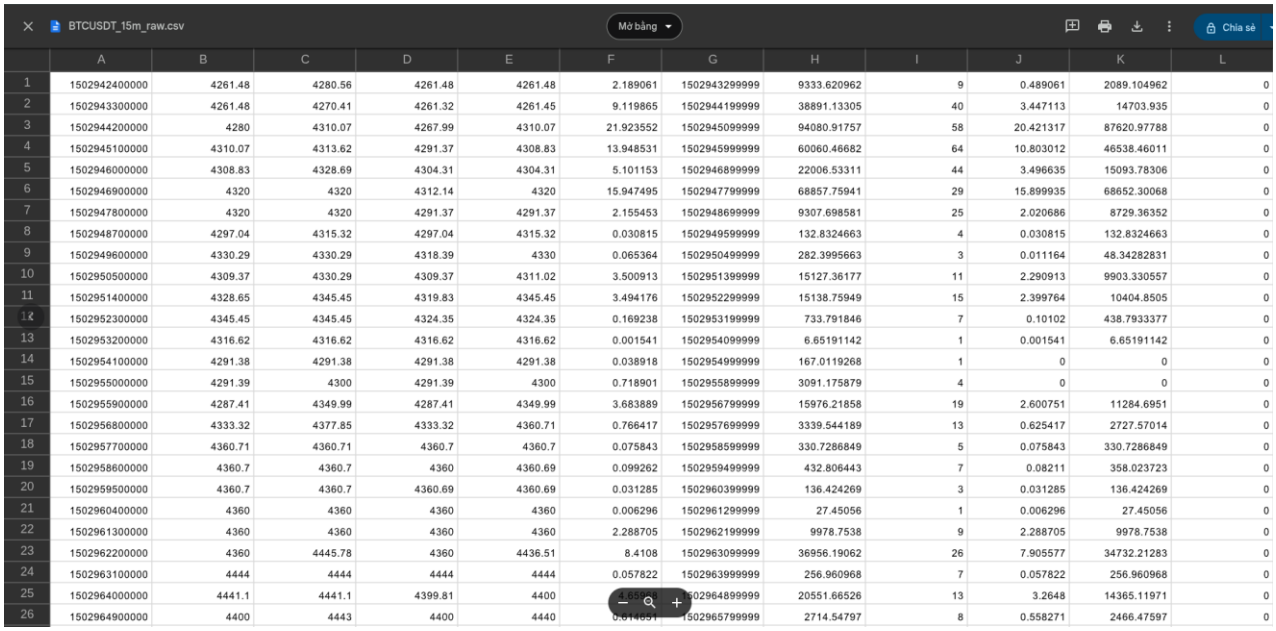
1.4. Tổng quan về dữ liệu

Nguồn dữ liệu: Dữ liệu giá Bitcoin (BTC/USDT) được thu thập từ sàn giao dịch Binance thông qua Binance API, sử dụng thư viện python-binance với khung thời gian 15 phút, giai đoạn từ 2017 đến Tháng 8 năm 2025.

Thu thập dữ liệu:

```python
#Lấy dữ liệu
from binance.client import Client
from google.colab import drive

# Mount Google Drive
drive.mount('/content/drive')

# Kết nối Binance
api_key = userdata.get('APIKeyBinnace')
secret_key = userdata.get('SecretKeyBinnace')
client = Client(api_key, secret_key)

# Tham số
symbol = "BTCUSDT"
interval = Client.KLINE_INTERVAL_15MINUTE
start_date = "1 Jan 2017"

# Lấy dữ liệu
klines = client.get_historical_klines(symbol, interval, start_date)

# Xuất ra CSV vào Drive
csv_path = "/content/drive/MyDrive/BTCUSDT_15m_raw.csv"
pd.DataFrame(klines).to_csv(csv_path, index=False, header=False)

# In ra thử 1 cây nến đầu tiên
print(klines[0])
print(f"File đã lưu tại: {csv_path}")
```

File CSV chứa dữ liệu lấy được:



BTCUSDT_15m_raw.csv — Mở bằng — Chia sẻ

	A	B	C	D	E	F	G	H	I	J	K	L
1	1502942400000	4261.48	4280.56	4261.48	4261.48	2.189061	1502943299999	9333.620962	9	0.489061	2089.104962	0
2	1502943300000	4261.48	4270.41	4261.32	4261.45	9.119865	1502944199999	38891.13305	40	3.447113	14703.935	0
3	1502944200000	4280	4310.07	4267.99	4310.07	21.923552	1502945099999	94080.91757	58	20.421317	87620.97788	0
4	1502945100000	4310.07	4313.62	4291.37	4308.83	13.948531	1502945999999	60060.46682	64	10.803012	46538.46011	0
5	1502946000000	4308.83	4328.69	4304.31	4304.31	5.101153	1502946899999	22006.53311	44	3.496635	15093.78306	0
6	1502946900000	4320	4320	4312.14	4320	15.947495	1502947799999	68857.75941	29	15.899935	68652.30068	0
7	1502947800000	4320	4320	4291.37	4291.37	2.155453	1502948699999	9307.698581	25	2.020686	8729.36352	0
8	1502948700000	4297.04	4315.32	4297.04	4315.32	0.030815	1502949599999	132.8324663	4	0.030815	132.8324663	0
9	1502949600000	4330.29	4330.29	4318.39	4330	0.065364	1502950499999	282.3995663	3	0.011164	48.34282831	0
10	1502950500000	4309.37	4330.29	4309.37	4311.02	3.500913	1502951399999	15127.36177	11	2.290913	9903.330557	0
11	1502951400000	4328.65	4345.45	4319.83	4345.45	3.494176	1502952299999	15138.75949	15	2.399764	10404.8505	0
12	1502952300000	4345.45	4345.45	4324.35	4324.35	0.169238	1502953199999	733.791846	7	0.10102	438.7933377	0
13	1502953200000	4316.62	4316.62	4316.62	4316.62	0.001541	1502954099999	6.65191142	1	0.001541	6.65191142	0
14	1502954100000	4291.38	4291.38	4291.38	4291.38	0.038918	1502954999999	167.0119268	1	0	0	0
15	1502955000000	4291.39	4300	4291.39	4300	0.718901	1502955899999	3091.175879	4	0	0	0
16	1502955900000	4287.41	4349.99	4287.41	4349.99	3.683889	1502956799999	15976.21858	19	2.600751	11284.6951	0
17	1502956800000	4333.32	4377.85	4333.32	4360.71	0.766417	1502957699999	3339.544189	13	0.625417	2727.57014	0
18	1502957700000	4360.71	4360.71	4360.7	4360.7	0.075843	1502958599999	330.7286849	5	0.075843	330.7286849	0
19	1502958600000	4360.7	4360.7	4360	4360.69	0.099262	1502959499999	432.806443	7	0.08211	358.023723	0
20	1502959500000	4360.7	4360.7	4360.69	4360.69	0.031285	1502960399999	136.424269	3	0.031285	136.424269	0
21	1502960400000	4360	4360	4360	4360	0.006296	1502961299999	27.45056	1	0.006296	27.45056	0
22	1502961300000	4360	4360	4360	4360	2.288705	1502962199999	9978.7538	9	2.288705	9978.7538	0
23	1502962200000	4360	4445.78	4360	4436.51	8.4108	1502963099999	36956.19062	26	7.905577	34732.21283	0
24	1502963100000	4444	4444	4444	4444	0.057822	1502963999999	256.960968	7	0.057822	256.960968	0
25	1502964000000	4441.1	4441.1	4399.81	4400	[illegible]	1502964899999	20551.66526	13	3.2648	14365.11971	0
26	1502964900000	4400	4443	4400	4440	0.614851	1502965799999	2714.54797	8	0.558271	2466.47597	0

Mô tả bộ dữ liệu:

- Số dòng: 280 253
- Số cột: 12 (Dữ liệu các cột do Binance cung cấp theo thứ tự)

STT	Tên cột chuẩn	Kiểu dữ liệu	Mô tả
1	open_time	int64	Thời gian mở nến dưới dạng timestamp (ms)
2	open	float64	Giá mở nến
3	high	float64	Giá cao nhất trong nến
4	low	float64	Giá thấp nhất trong nến
5	close	float64	Giá đóng nến
6	volume	float64	Khối lượng coin giao dịch trong nến
7	close_time	int64	Thời gian đóng nến dưới dạng timestamp (ms)
8	quote_asset_volume	float64	Khối lượng tiền quote (USDT) giao dịch trong nến
9	number_of_trades	int64	Số lượng giao dịch trong nến
10	taker_buy_base_asset_volume	float64	Khối lượng coin mua bởi bên taker
11	taker_buy_quote_asset_volume	float64	Khối lượng tiền quote (USDT) mua bởi bên taker
12	ignore	int64	Cột không sử dụng

1.5. Kỹ thuật sử dụng

- Môi trường lập trình: Google Colab, Jupyter Notebook, Anaconda
- Ngôn ngữ lập trình: Python (phiên bản 3.12.11)
- Thư viện:
  - Xử lý dữ liệu: pandas, numpy
  - Trực quan hóa: matplotlib, seaborn
  - Deep learning: sklearn, tensorflow/keras
  - Lấy dữ liệu thị trường: python-binance
- Mô hình dự báo: LSTM (Long Short-Term Memory)
- GPU: NVIDIA GeForce RTX 3050 (CUDA Toolkit, cuDNN)
- Chỉ số đánh giá:
  - Dự báo giá: MSE, RMSE, MAE, R^2
  - Dự báo xu hướng: Accuracy, Precision, Recall, F1-score

2. Xử lý và trực quan hóa dữ liệu

2.1. Khám phá dữ liệu ban đầu

```
[11]  1 print(df.head()) #5 dòng đầu
```

```
              0       1       2       3       4         5   \
0  1502942400000  4261.48  4280.56  4261.48  4261.48  2.189061
1  1502943300000  4261.48  4270.41  4261.32  4261.45  9.119865
2  1502944200000  4280.00  4310.07  4267.99  4310.07 21.923552
3  1502945100000  4310.07  4313.62  4291.37  4308.83 13.948531
4  1502946000000  4308.83  4328.69  4304.31  4304.31  5.101153

              6            7   8         9            10  11
0  1502943299999   9333.620962   9  0.489061   2089.104962   0
1  1502944199999  38891.133046  40  3.447113  14703.934995   0
2  1502945099999  94080.917568  58 20.421317  87620.977876   0
3  1502945999999  60060.466816  64 10.803012  46538.460109   0
4  1502946899999  22006.533111  44  3.496635  15093.783057   0
```

```
[12]  1 print(df.tail()) #5 dòng cuối
```

```
                  0         1         2         3         4         5   \
280248  1755674100000  113722.84  113837.93  113720.86  113720.87   91.20673
280249  1755675000000  113720.87  113720.87  113588.00  113612.01  127.34973
280250  1755675900000  113612.01  113672.74  113432.00  113490.13  124.31949
280251  1755676800000  113490.14  113609.94  113401.00  113609.94  119.43721
280252  1755677700000  113609.94  113854.34  113609.93  113854.33  186.23173

                  6            7     8         9            10  11
280248  1755674999999  1.037709e+07  16790  56.19940  6.394022e+06   0
280249  1755675899999  1.447374e+07  21108  65.78950  7.476427e+06   0
280250  1755676799999  1.411649e+07  17054  62.15294  7.058497e+06   0
280251  1755677699999  1.355756e+07  25094  73.33694  8.325082e+06   0
280252  1755678599999  2.118173e+07  21800  94.40609  1.073844e+07   0
```

```
[13]  1 print(df.info()) #Thông tin tổng quan
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 280253 entries, 0 to 280252
Data columns (total 12 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   0       280253 non-null  int64
 1   1       280253 non-null  float64
 2   2       280253 non-null  float64
 3   3       280253 non-null  float64
 4   4       280253 non-null  float64
 5   5       280253 non-null  float64
 6   6       280253 non-null  int64
 7   7       280253 non-null  float64
 8   8       280253 non-null  int64
 9   9       280253 non-null  float64
 10  10      280253 non-null  float64
 11  11      280253 non-null  int64
dtypes: float64(8), int64(4)
memory usage: 25.7 MB
None
```

```
1 print(df.describe()) #Kiểm tra thống kê cơ bản
```

```
                 0              1              2              3   \
count  2.802530e+05  280253.000000  280253.000000  280253.000000
mean   1.629462e+12   32839.944117   32908.548056   32769.098271
std    7.293337e+10   29149.178672   29195.083913   29102.217734
min    1.502942e+12    2830.000000    2880.010000    2817.000000
25%    1.566355e+12    8911.070000    8935.000000    8884.000000
50%    1.629553e+12   23359.320000   23403.940000   23312.570000
75%    1.692621e+12   48265.890000   48397.030000   48132.050000
max    1.755678e+12  124243.310000  124474.000000  123666.010000

                 4              5              6              7   \
count  280253.000000  280253.000000  2.802530e+05  2.802530e+05
mean    32840.323336     673.062480  1.629463e+12  1.833338e+07
std     29149.538499    1084.658613  7.293337e+10  2.865500e+07
min      2820.000000       0.000000  1.502943e+12  0.000000e+00
25%      8911.010000     183.691948  1.566355e+12  2.948082e+06
50%     23359.040000     345.343772  1.629553e+12  8.763874e+06
75%     48265.890000     696.766332  1.692622e+12  2.246675e+07
max    124243.320000   40371.405060  1.755679e+12  8.961201e+08

                 8              9              10        11
count  280253.000000  280253.000000  2.802530e+05  280253.0
mean    18457.962363     334.427246  9.064991e+06       0.0
std     27392.818673     541.186124  1.438883e+07       0.0
min         0.000000       0.000000  0.000000e+00       0.0
25%      3561.000000      89.796380  1.453104e+06       0.0
50%      9008.000000     172.179050  4.180236e+06       0.0
75%     20981.000000     346.882421  1.107299e+07       0.0
max    606234.000000   19925.616600  5.619115e+08       0.0
```

2.2. Tiền xử lý dữ liệu

2.2.1. Thêm tiêu đề cho các cột

```
1 #Thêm tiêu đề cho các cột
2 # Gán tên cột theo thứ tự Binance cung cấp
3 df.columns = ['open_time','Open','High','Low','Close','Volume','close_time',
4               'quote_asset_volume','number_of_trades','taker_buy_base_asset_volume',
5               'taker_buy_quote_asset_volume','ignore']
6 df.columns
```

```
Index(['open_time', 'Open', 'High', 'Low', 'Close', 'Volume', 'close_time',
       'quote_asset_volume', 'number_of_trades', 'taker_buy_base_asset_volume',
       'taker_buy_quote_asset_volume', 'ignore'],
      dtype='object')
```

2.2.2. Loại bỏ các cột không cần thiết

Trong bài toán dự báo giá Bitcoin (hoặc dự đoán tăng/giảm), các cột quan trọng nhất là:

- open_time → làm trục thời gian cho chuỗi dữ liệu.
- Open, High, Low, Close → giá mở, cao, thấp, đóng; đặc trưng cơ bản để mô hình học chuỗi dự đoán giá.

- Volume → khối lượng giao dịch, cung cấp thông tin về thanh khoản và sức mạnh xu hướng.

Các cột khác (như quote_asset_volume, number_of_trades, taker_buy_base_asset_volume,…) ít tác động trực tiếp đến dự báo giá ngắn hạn nên có thể loại bỏ để giảm nhiễu và tối ưu hóa dữ liệu cho mô hình.

```
#Loại bỏ cột không cần thiết
df = df[['open_time','Open','High','Low','Close','Volume']]
df
```

	open_time	Open	High	Low	Close	Volume
0	1502942400000	4261.48	4280.56	4261.48	4261.48	2.189061
1	1502943300000	4261.48	4270.41	4261.32	4261.45	9.119865
2	1502944200000	4280.00	4310.07	4267.99	4310.07	21.923552
3	1502945100000	4310.07	4313.62	4291.37	4308.83	13.948531
4	1502946000000	4308.83	4328.69	4304.31	4304.31	5.101153
...	...	...	...	...	...	...
280248	1755674100000	113722.84	113837.93	113720.86	113720.87	91.206730
280249	1755675000000	113720.87	113720.87	113588.00	113612.01	127.349730
280250	1755675900000	113612.01	113672.74	113432.00	113490.13	124.319490
280251	1755676800000	113490.14	113609.94	113401.00	113609.94	119.437210
280252	1755677700000	113609.94	113854.34	113609.93	113854.33	186.231730

280253 rows × 6 columns

2.2.3. Kiểm tra giá trị thiếu, không hợp lệ hoặc trùng

```
print(df.isna().sum()) #Kiểm tra giá trị trống
print(df.duplicated().sum()) #Kiểm tra trùng nhau
```

```
open_time    0
Open         0
High         0
Low          0
Close        0
Volume       0
dtype: int64
0
```

2.2.4. Chuyển kiểu dữ liệu

Hiện tại open_time là int64 (timestamp) cần chuyển sang datetime và set index để dễ trực quan hóa

```
[29]  1 df['open_time'] = pd.to_datetime(df['open_time'], unit='ms') #Chuyển kiểu open_time sang datetime
      2 df.set_index('open_time', inplace=True) # Set open_time làm index
```

2.2.5. Bổ sung chỉ số kỹ thuật

Thêm MA20 (đường trung bình động 20 nến) và RSI14 (chỉ số sức mạnh tương đối, chu kỳ 14) để tăng cường thông tin cho mô hình.

- MA20: Làm mượt dữ liệu giá, giúp nhận diện xu hướng dài hạn và giảm nhiễu trong khung nến 15 phút.
- RSI14: Đo lường tâm lý thị trường (quá mua >70, quá bán <30), hỗ trợ dự báo xu hướng tăng/giảm.
- Cách tính: Tính MA20 bằng hàm rolling của pandas. Tính RSI14 dựa trên công thức chuẩn, sử dụng giá thay đổi để tính gain/loss.

```python
# Tính MA20
df['MA20'] = df['Close'].rolling(window=20).mean()

# Tính RSI14
def calculate_rsi(data, periods=14):
    delta = data.diff()
    gain = np.where(delta > 0, delta, 0)
    loss = np.where(delta < 0, -delta, 0)
    gain = pd.Series(gain, index=data.index)
    loss = pd.Series(loss, index=data.index)
    avg_gain = gain.rolling(window=periods).mean()
    avg_loss = loss.rolling(window=periods).mean()
    rs = avg_gain / avg_loss
    rs = rs.replace([np.inf, -np.inf], np.nan)  # Xử lý chia cho 0
    rsi = 100 - (100 / (1 + rs))
    return rsi

df['RSI14'] = calculate_rsi(df['Close'], periods=14)

# Xử lý giá trị thiếu do tính toán
df = df.dropna()
print(f"Dữ liệu sau bổ sung MA20, RSI14: {df.shape}")
```

```
Dữ liệu sau bổ sung MA20, RSI14: (280233, 8)
```

2.2.6. Kiểm tra dữ liệu

```
#Kiểm tra lại dữ liệu
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 280233 entries, 2017-08-17 08:45:00 to 2025-08-20 08:15:00
Data columns (total 8 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   Open    280233 non-null  float64
 1   High    280233 non-null  float64
 2   Low     280233 non-null  float64
 3   Close   280233 non-null  float64
 4   Volume  280233 non-null  float64
 5   Year    280233 non-null  int32  
 6   MA20    280233 non-null  float64
 7   RSI14   280233 non-null  float64
dtypes: float64(7), int32(1)
memory usage: 18.2 MB
None
```

```
print(df.describe())
```

```
                Open           High            Low          Close  \
count  280233.000000  280233.000000  280233.000000  280233.000000   
mean    32841.855988   32910.463483   32771.005906   32842.234816   
std     29149.269830   29195.174494   29102.310023   29149.630034   
min      2830.000000    2880.010000    2817.000000    2820.000000   
25%      8912.100000    8936.880000    8885.000000    8912.430000   
50%     23362.780000   23405.000000   23314.900000   23362.970000   
75%     48267.590000   48399.000000   48133.330000   48267.950000   
max    124243.310000  124474.000000  123666.010000  124243.320000   

              Volume           Year           MA20          RSI14  
count  280233.000000  280233.000000  280233.000000  280233.000000  
mean      673.105907    2021.134849   32838.532085      50.612246  
std      1084.682869       2.343292   29145.361384      15.221196  
min         0.000000    2017.000000    2995.682000       0.020264  
25%       183.717630    2019.000000    8907.464500      40.117786  
50%       345.368684    2021.000000   23362.457000      50.574376  
75%       696.797755    2023.000000   48230.842000      61.040681  
max     40371.405060    2025.000000  123555.717000      99.877560  
```

2.2.7. Lưu lại dữ liệu đã xử lý

```
#Xuất file CSV dữ liệu đã xử lý
# Đường dẫn lưu file xử lý xong
output_path = "/content/drive/MyDrive/BTCUSDT_15m_processed.csv"

# Xuất DataFrame ra CSV
df.to_csv(output_path, index=False)  # index=False để không xuất cột chỉ số

print(f"Đã lưu file xử lý xong tại: {output_path}")
```

```
Đã lưu file xử lý xong tại: /content/drive/MyDrive/BTCUSDT_15m_processed.csv
```

BTCUSDT_15m_processed.csv

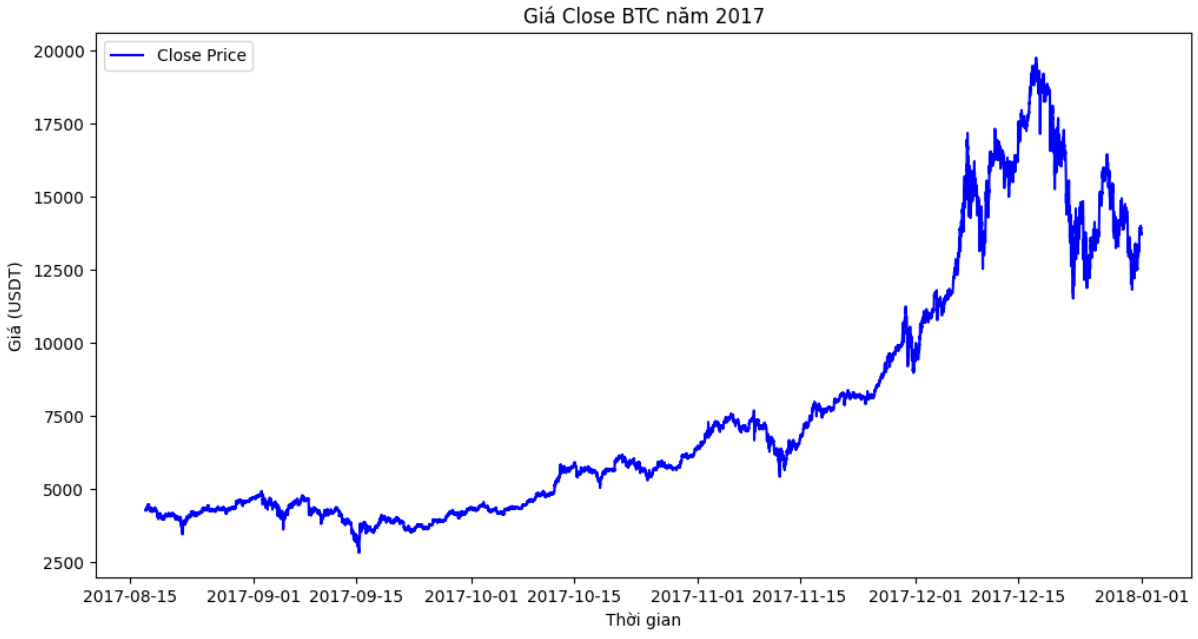
A	B	C	D	E	F	G	H
Open	High	Low	Close	Volume	Year	MA20	RSI14
4360.7	4360.7	4360.69	4360.69	0.031285	2017	4319.2215	58.33504035
4360	4360	4360	4360	0.006296	2017	4324.1475	65.87554939
4360	4360	4360	4360	2.288705	2017	4329.075	61.62330905
4360	4445.78	4360	4436.51	8.4108	2017	4335.397	70.96405936
4444	4444	4444	4444	0.057822	2017	4342.1555	77.4140348
4441.1	4441.1	4399.81	4400	4.65968	2017	4346.94	60.81869025
4400	4443	4400	4440	0.614651	2017	4352.94	71.33685104
4400	4419	4400	4415	2.078235	2017	4359.1215	67.06327182
4419	4470	4419	4460	17.512833	2017	4366.3555	77.3698221
4460	4474.8	4454.72	4474.8	16.084544	2017	4373.5955	77.81490675
4474.8	4485.39	4465.06	4485.39	8.844785	2017	4382.314	74.63430609
4485.39	4485.39	4459.96	4469.93	0.583082	2017	4388.538	69.53426814
4469.93	4469.93	4427.3	4427.3	1.505642	2017	4393.6855	60.33583711
4436.06	4449.56	4436.06	4441.87	11.174158	2017	4399.948	62.05381006

2.3. Phân tích dữ liệu qua các năm

2.3.1. Năm 2017

```
[59]  1 btc_2017 = df[df.index.year == 2017]
      2 print("Số lượng bản ghi 2017:", len(btc_2017))
      3 print(btc_2017['Close'].describe())
```

```
Số lượng bản ghi 2017: 13103
count    13103.000000
mean      7571.629454
std       4261.428809
min       2820.000000
25%       4308.030000
50%       5779.690000
75%       9501.090000
max      19756.020000
Name: Close, dtype: float64
```



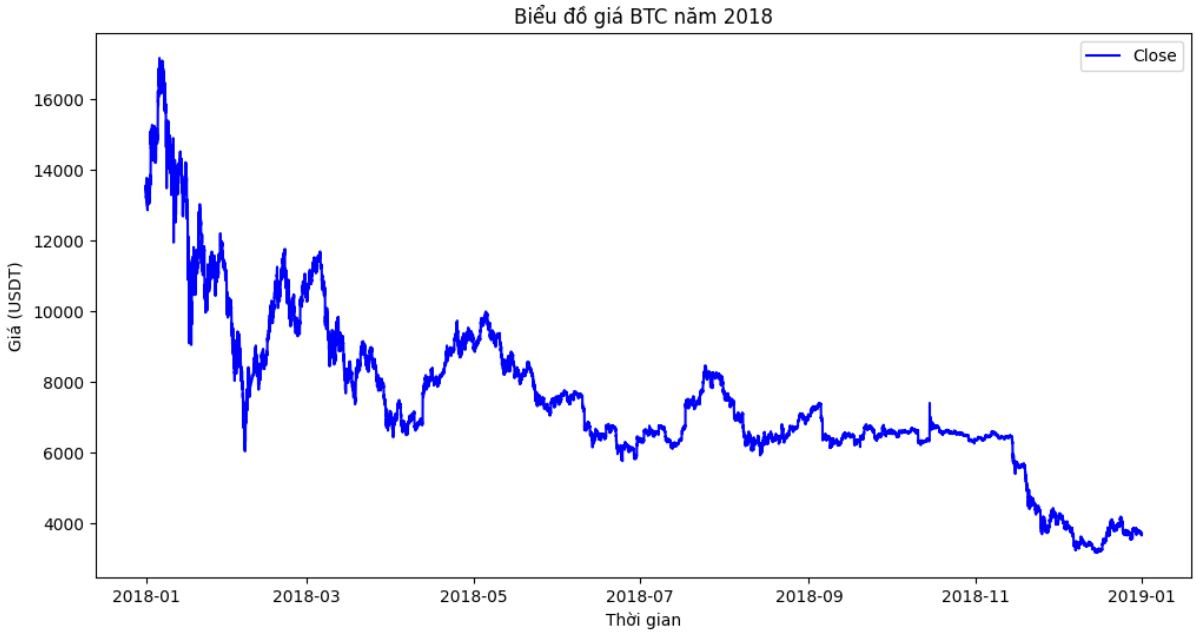
Trong giai đoạn từ tháng 8 đến tháng 12 năm 2017, dữ liệu giá đóng cửa của Bitcoin cho thấy mức trung bình khoảng 7.572 USD với độ lệch chuẩn hơn 4.261 USD, phản ánh sự biến động rất lớn quanh mức giá trung bình. Giá thấp nhất trong giai đoạn này ghi nhận khoảng 2.820 USD, trong khi giá cao nhất đạt gần 19.756 USD, cho thấy biên độ dao động rộng. Các phân vị cũng thể hiện rõ quá trình tăng giá: 25% dữ liệu nằm dưới mức 4.308 USD, 50% dưới mức 5.780 USD, trong khi 75% nằm dưới mức 9.501 USD - điều này phản ánh sự tăng trưởng mạnh mẽ về giá trong những tháng cuối năm.

Quan sát thống kê và diễn biến thị trường cho thấy Bitcoin đã trải qua một giai đoạn bùng nổ điển hình. Giá trị tăng mạnh và lập đỉnh lịch sử gần 20.000 USD vào cuối năm 2017, biến năm này trở thành cột mốc thu hút sự quan tâm toàn cầu đối với tiền điện tử.

2.3.2. Năm 2018

```
#Lọc dữ liệu năm 2018
btc_2018 = df.loc['2018-01-01':'2018-12-31']
print("Số lượng bản ghi 2018:", len(btc_2018))
print(btc_2018['Close'].describe())
```

```
Số lượng bản ghi 2018: 34778
count    34778.000000
mean      7539.923261
std       2387.267314
min       3167.070000
25%       6408.027500
50%       6907.160000
75%       8593.702500
max      17173.970000
Name: Close, dtype: float64
```



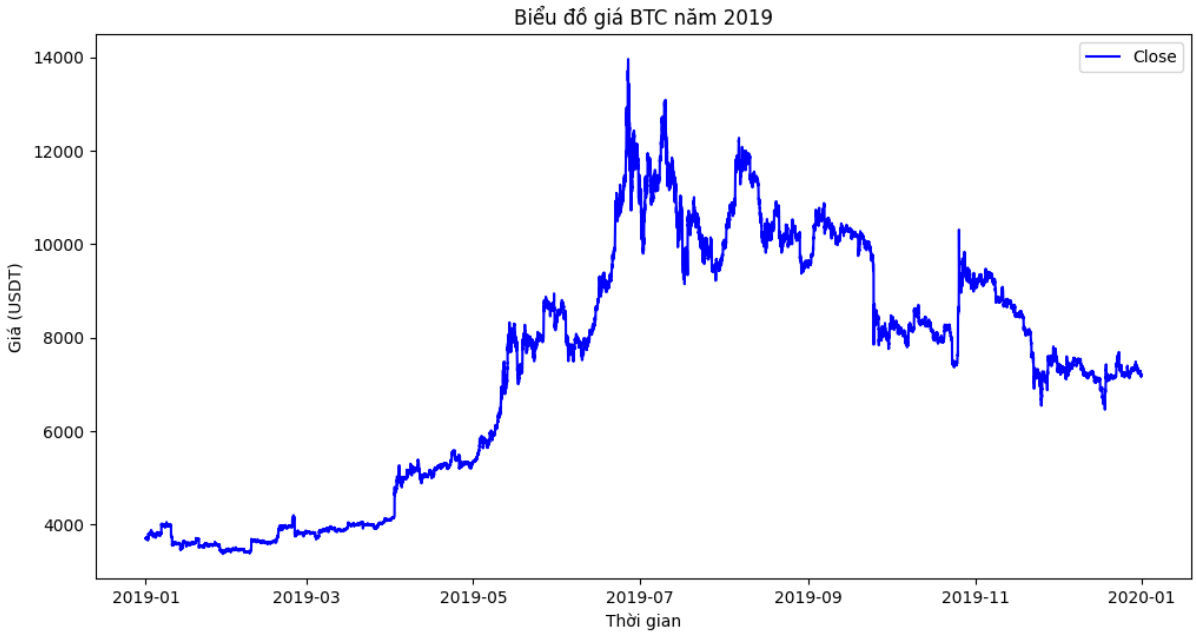
Trong năm 2018, dữ liệu giá đóng cửa của Bitcoin cho thấy mức trung bình khoảng 7.540 USD, với độ lệch chuẩn hơn 2.387 USD, phản ánh sự biến động đáng kể quanh mức giá trung bình. Giá thấp nhất trong năm được ghi nhận ở mức khoảng 3.167 USD, trong khi mức cao nhất đạt gần 17.174 USD. Các phân vị thể hiện rõ xu hướng giảm dần: 25% dữ liệu nằm dưới 6.408 USD, 50% dưới 6.907 USD, và 75% dưới 8.594 USD, cho thấy phần lớn thời gian giao dịch của năm 2018 diễn ra ở mức giá thấp hơn so với giai đoạn đầu năm.

Quan sát này phù hợp với thực tế thị trường khi Bitcoin bước vào giai đoạn suy giảm mạnh sau đỉnh cao cuối năm 2017. Sự sụt giảm kéo dài trong năm 2018 thường được gọi là "mùa đông tiền số", phản ánh sự thoái trào sau bong bóng ICO. Đây là giai đoạn thị trường thể hiện rõ tính chu kỳ, với mức điều chỉnh trên 70% so với đỉnh cũ.

2.3.3. Năm 2019

```
#Lọc dữ liệu năm 2019
btc_2019 = df.loc['2019-01-01':'2019-12-31']
print("Số lượng bản ghi 2019:", len(btc_2019))
print(btc_2019['Close'].describe())
```

```
Số lượng bản ghi 2019: 34923
count    34923.000000
mean      7353.050820
std       2643.552613
min       3366.410000
25%       4445.730000
50%       7800.870000
75%       9566.135000
max      13960.760000
Name: Close, dtype: float64
```



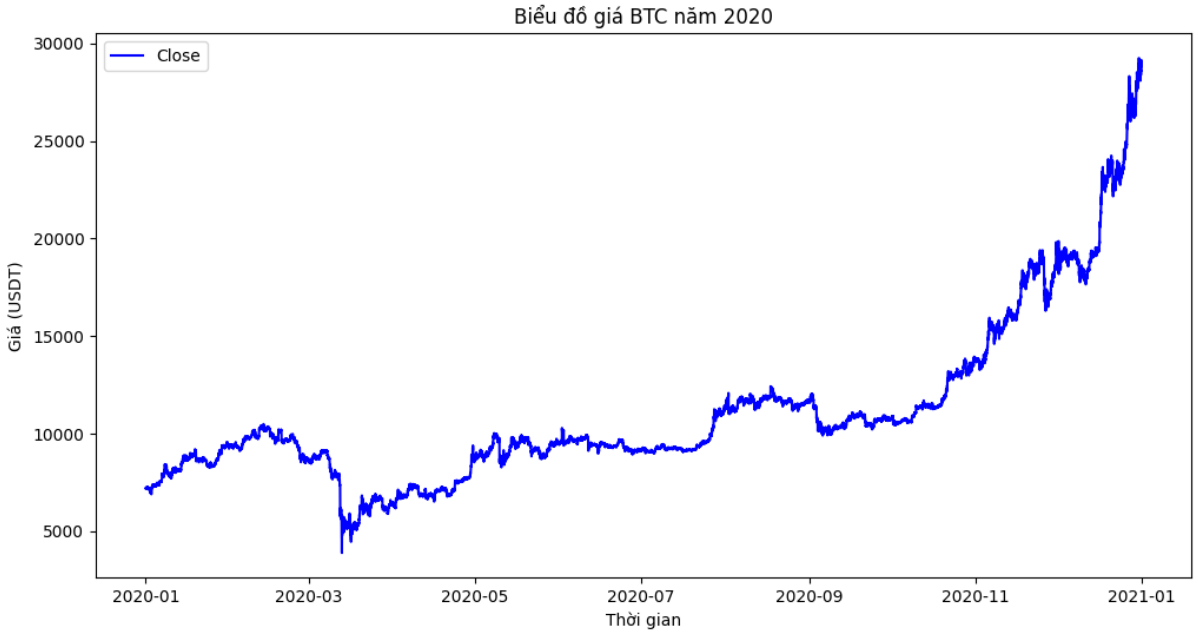
Trong năm 2019, dữ liệu giá đóng cửa của Bitcoin cho thấy mức trung bình khoảng 7.353 USD, với độ lệch chuẩn hơn 2.644 USD, phản ánh sự biến động mạnh quanh mức giá trung bình. Giá thấp nhất trong năm rơi vào khoảng 3.366 USD, trong khi mức cao nhất đạt gần 13.961 USD. Phân vị thống kê thể hiện xu hướng dao động rộng: 25% dữ liệu nằm dưới 4.446 USD, 50% dưới 7.801 USD, và 75% dưới 9.566 USD, cho thấy thị trường đã có giai đoạn hồi phục rõ rệt so với năm 2018.

Quan sát diễn biến cho thấy năm 2019 đánh dấu sự trở lại của dòng vốn vào Bitcoin, đặc biệt trong nửa đầu năm khi giá tăng nhanh và đạt đỉnh trung gian vào tháng 6. Tuy nhiên, nửa cuối năm thị trường điều chỉnh và đi ngang, phản ánh trạng thái chờ đợi trước khi bước vào một chu kỳ tăng trưởng mới. Đây là năm phục hồi quan trọng sau "mùa đông tiền số" 2018, dù xu hướng chưa đủ mạnh để tạo lập đỉnh lịch sử mới.

2.3.4. Năm 2020

```
#Lọc dữ liệu năm 2020
btc_2020 = df.loc['2020-01-01':'2020-12-31']
print("Số lượng bản ghi 2020:", len(btc_2020))
print(btc_2020['Close'].describe())
```

```
Số lượng bản ghi 2020: 35053
count    35053.000000
mean     11066.416934
std       4240.579281
min       3882.220000
25%       8855.900000
50%       9692.040000
75%      11627.400000
max      29248.310000
Name: Close, dtype: float64
```



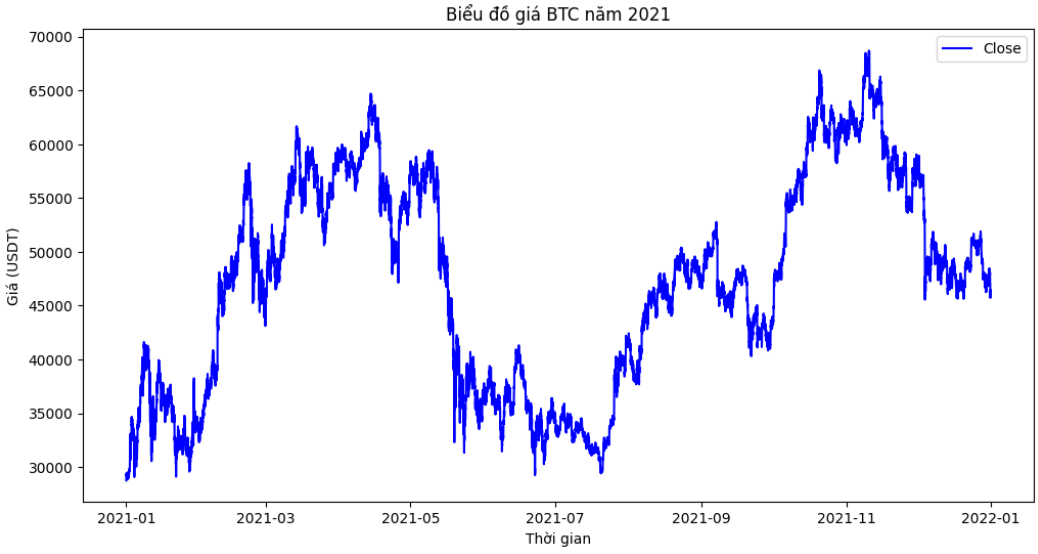
Trong năm 2020, dữ liệu giá đóng cửa của Bitcoin cho thấy mức trung bình khoảng 11.066 USD, với độ lệch chuẩn hơn 4.241 USD, phản ánh sự biến động lớn quanh mức giá trung bình. Giá thấp nhất trong năm ghi nhận ở mức khoảng 3.882 USD, trong khi mức cao nhất đạt gần 29.248 USD. Các phân vị thể hiện xu hướng tăng rõ rệt: 25% dữ liệu nằm dưới 8.856 USD, 50% dưới 9.692 USD, và 75% dưới 11.627 USD, cho thấy phần lớn thời gian của năm giá Bitcoin duy trì dưới 12.000 USD trước khi bứt phá mạnh vào cuối năm.

Quan sát diễn biến cho thấy thị trường Bitcoin trải qua hai giai đoạn nổi bật. Đầu năm, giá sụt giảm mạnh vào tháng 3/2020 khi đại dịch COVID-19 gây ra cú sốc thanh khoản toàn cầu, đưa Bitcoin về dưới 4.000 USD. Tuy nhiên, từ quý II trở đi, thị trường nhanh chóng phục hồi nhờ các gói nới lỏng tiền tệ và dòng vốn chảy vào tài sản rủi ro. Nửa cuối năm chứng kiến đà tăng liên tục, với giá vượt 20.000 USD lần đầu tiên kể từ 2017 và đạt gần 29.000 USD vào cuối năm. Năm 2020 vì vậy được xem là bước ngoặt, đặt nền móng cho chu kỳ tăng trưởng bùng nổ tiếp theo.

2.3.5. Năm 2021

```
#Lọc dữ liệu năm 2021
btc_2021 = df.loc['2021-01-01':'2021-12-31']
print("Số lượng bản ghi 2021:", len(btc_2021))
print(btc_2021['Close'].describe())
```

```
Số lượng bản ghi 2021: 34975
count    34975.000000
mean     47357.659716
std       9821.773703
min      28752.800000
25%      38061.210000
50%      47898.090000
75%      56203.555000
max      68718.900000
Name: Close, dtype: float64
```



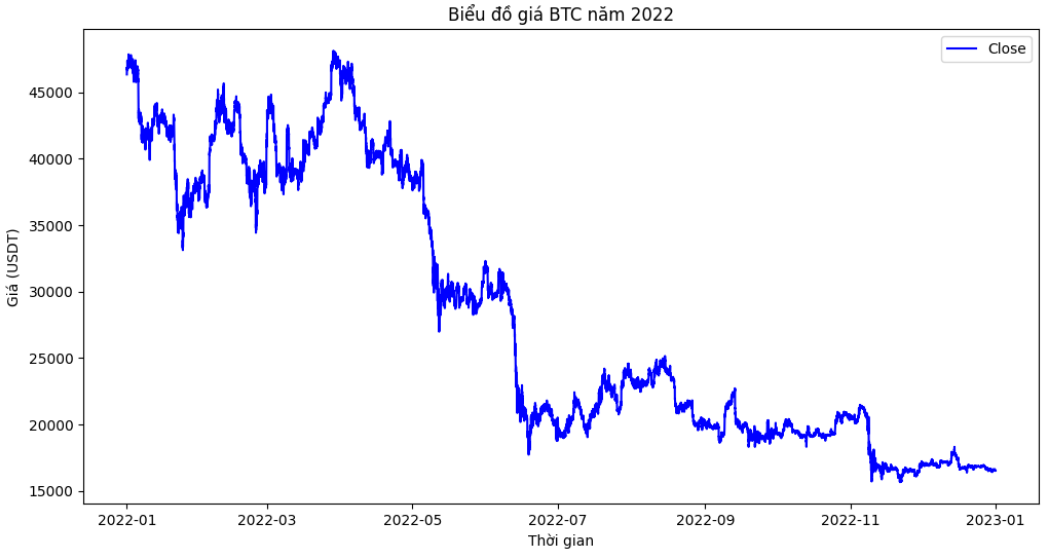
Trong năm 2021, dữ liệu giá đóng cửa của Bitcoin cho thấy mức trung bình đạt khoảng 47.358 USD, với độ lệch chuẩn gần 9.822 USD, phản ánh sự biến động rất cao quanh mức giá trung bình. Giá thấp nhất trong năm được ghi nhận ở mức 28.753 USD, trong khi giá cao nhất đạt gần 68.719 USD, lập đỉnh lịch sử mới vào tháng 11. Các phân vị cho thấy cấu trúc giá trải rộng: 25% dữ liệu nằm dưới 38.061 USD, 50% dưới 47.898 USD, và 75% dưới 56.204 USD, cho thấy giá thường xuyên duy trì ở mức cao hơn giai đoạn trước đó.

Diễn biến thực tế cho thấy năm 2021 là một trong những giai đoạn sôi động nhất của thị trường tiền điện tử. Trong nửa đầu năm, Bitcoin tăng mạnh và đạt đỉnh khoảng 64.000 USD vào tháng 4, nhưng sau đó giảm sâu về quanh 30.000 USD vào tháng 7 do các chính sách siết chặt và lo ngại về môi trường khai thác tại Trung Quốc. Nửa cuối năm, Bitcoin tiếp tục hồi phục và lập đỉnh mới gần 69.000 USD vào tháng 11 trước khi điều chỉnh. Đây là năm minh họa rõ rệt tính chất rủi ro cao: lợi nhuận tiềm năng lớn song đi kèm những đợt điều chỉnh sâu, thể hiện đặc trưng biến động mạnh vốn có của Bitcoin.

2.3.6. Năm 2022

```python
#Lọc dữ liệu năm 2022
btc_2022 = df.loc['2022-01-01':'2022-12-31']
print("Số lượng bản ghi 2022:", len(btc_2022))
print(btc_2022['Close'].describe())
```

```
Số lượng bản ghi 2022: 35040
count    35040.000000
mean     28226.341193
std      10191.525718
min      15649.520000
25%      19520.727500
50%      23149.875000
75%      39014.985000
max      48118.480000
Name: Close, dtype: float64
```



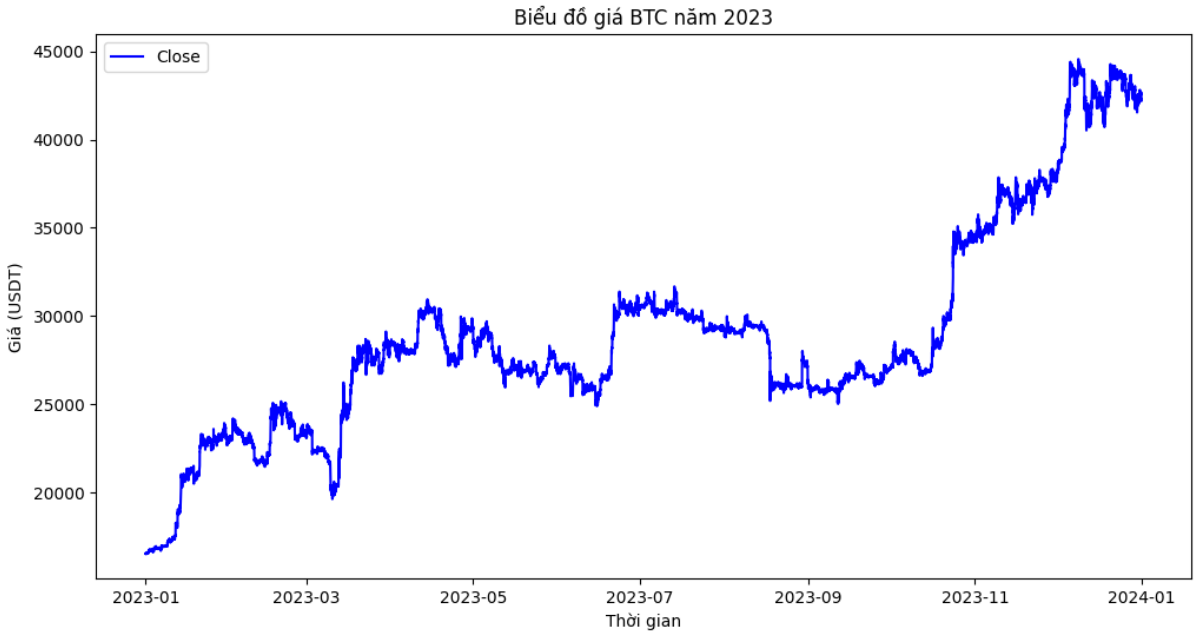
Trong năm 2022, dữ liệu giá đóng cửa của Bitcoin cho thấy mức trung bình khoảng 28.226 USD, với độ lệch chuẩn hơn 10.192 USD, phản ánh sự biến động rất mạnh so với mức trung bình. Giá thấp nhất trong năm ghi nhận ở mức 15.650 USD, trong khi mức cao nhất đạt gần 48.118 USD. Các phân vị thể hiện sự phân hóa rõ rệt: 25% dữ liệu nằm dưới 19.521 USD, 50% dưới 23.150 USD, và 75% dưới 39.015 USD, cho thấy phân nửa thời gian trong năm, giá Bitcoin giao dịch dưới 23.000 USD, thấp hơn nhiều so với năm 2021.

Diễn biến thực tế cho thấy 2022 là một năm suy giảm mạnh của thị trường. Sau khi mở đầu năm ở vùng trên 45.000 USD, Bitcoin dần trượt dốc, đặc biệt là hai giai đoạn sụt giảm nghiêm trọng: tháng 5/2022 khi hệ sinh thái Terra/LUNA sụp đổ, và tháng 11/2022 khi sàn giao dịch FTX phá sản, kéo theo niềm tin toàn thị trường suy yếu. Cuối năm, giá chỉ còn quanh 16.500 USD. Đây là một trong những giai đoạn giảm sâu nhất trong lịch sử Bitcoin, minh chứng cho tính rủi ro đặc hữu của tài sản này.

2.3.7. Năm 2023

```python
#Lọc dữ liệu năm 2023
btc_2023 = df.loc['2023-01-01':'2023-12-31']
print("Số lượng bản ghi 2023:", len(btc_2023))
print(btc_2023['Close'].describe())
```

```
Số lượng bản ghi 2023: 35035
count    35035.00000
mean     28804.37489
std       5882.40774
min      16510.68000
25%      25922.28000
50%      27716.32000
75%      30286.63000
max      44563.95000
Name: Close, dtype: float64
```



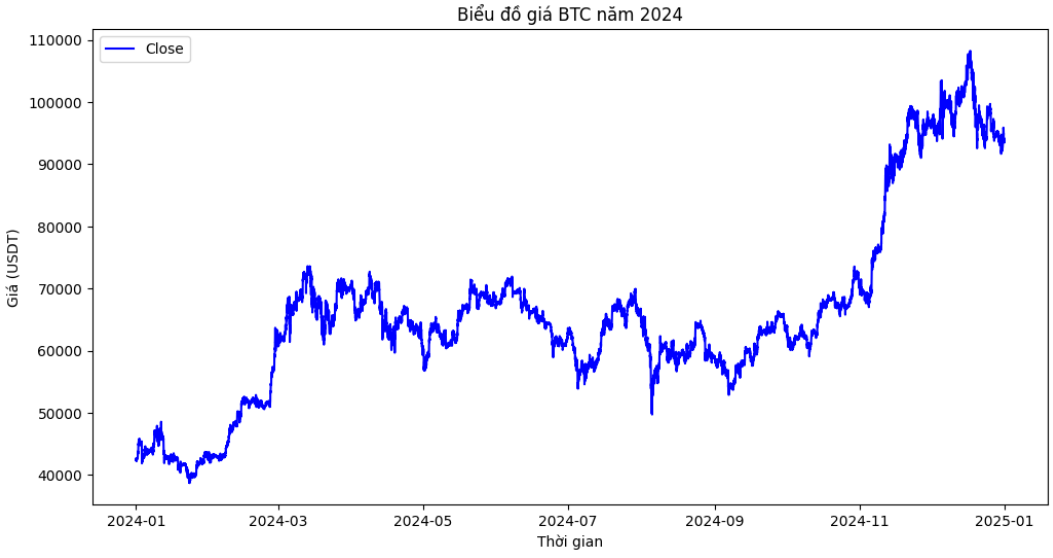
Trong năm 2023, dữ liệu giá đóng cửa của Bitcoin cho thấy mức trung bình khoảng 28.804 USD, với độ lệch chuẩn hơn 5.882 USD, phản ánh sự dao động lớn nhưng thấp hơn so với giai đoạn khủng hoảng 2022. Giá thấp nhất trong năm ghi nhận ở mức 16.511 USD, trong khi giá cao nhất đạt gần 44.564 USD. Các phân vị thể hiện sự tập trung của giá quanh vùng trung bình: 25% dữ liệu nằm dưới 25.922 USD, 50% dưới 27.716 USD, và 75% dưới 30.287 USD, cho thấy phần lớn thời gian trong năm, Bitcoin dao động trong vùng 25.000–30.000 USD trước khi bứt phá vào cuối năm.

Diễn biến thực tế cho thấy 2023 là một năm phục hồi quan trọng của Bitcoin sau giai đoạn suy giảm nghiêm trọng 2022. Giá mở đầu năm quanh 16.500 USD, dần phục hồi trong các quý và tăng tốc từ quý IV, đạt đỉnh gần 44.600 USD vào tháng 12. Động lực tăng trưởng đến từ sự kỳ vọng về việc Ủy ban Chứng khoán Mỹ (SEC) phê duyệt các quỹ ETF Bitcoin giao ngay, cùng với việc môi trường vĩ mô ổn định hơn khi lạm phát toàn cầu hạ nhiệt. Đây là năm củng cố lại niềm tin thị trường, với xu hướng tăng rõ rệt và biến động tương đối lành mạnh so với các năm trước.

2.3.8. Năm 2024

```python
#Lọc dữ liệu năm 2024
btc_2024 = df.loc['2024-01-01':'2024-12-31']
print("Số lượng bản ghi 2024:", len(btc_2024))
print(btc_2024['Close'].describe())
```

```
Số lượng bản ghi 2024: 35136
count     35136.000000
mean      65897.872800
std       14680.590731
min       38705.290000
25%       59012.755000
50%       64207.685000
75%       69067.312500
max      108258.390000
Name: Close, dtype: float64
```



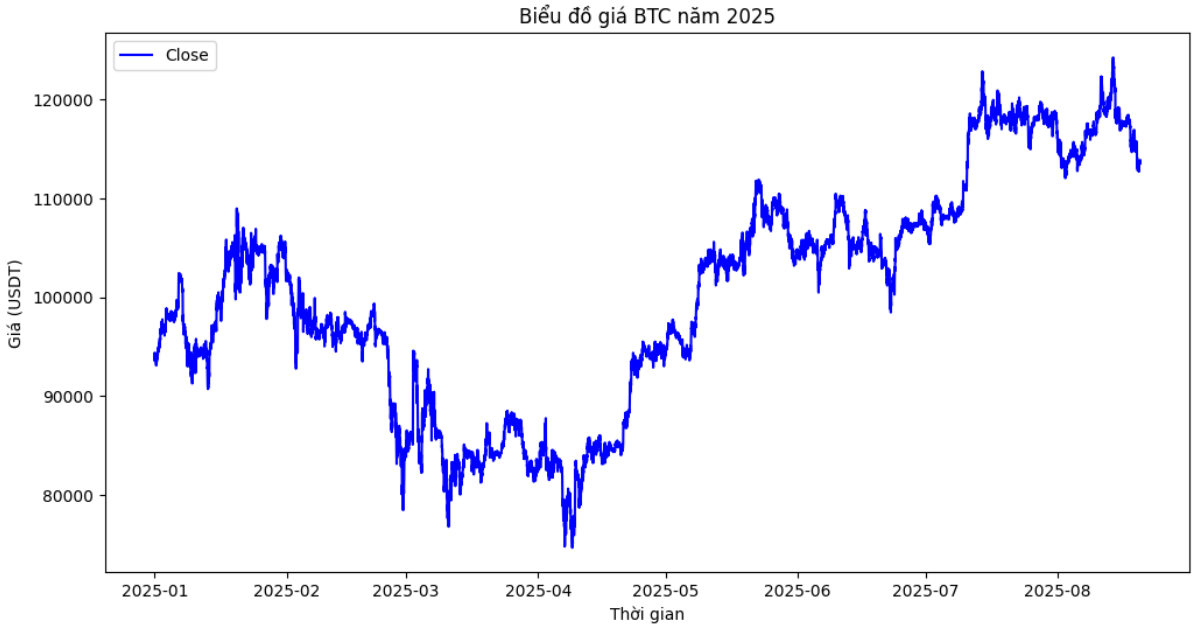
Trong năm 2024, dữ liệu giá đóng cửa của Bitcoin cho thấy mức trung bình khoảng 65.898 USD, với độ lệch chuẩn hơn 14.681 USD, phản ánh sự biến động rất mạnh quanh mức giá trung bình. Giá thấp nhất trong năm ghi nhận ở mức 38.705 USD, trong khi mức cao nhất đạt trên 108.258 USD, thiết lập kỷ lục mới trong lịch sử giao dịch Bitcoin. Các phân vị thể hiện xu hướng tăng rõ rệt: 25% dữ liệu nằm dưới 59.013 USD, 50% dưới 64.208 USD, và 75% dưới 69.067 USD, cho thấy phần lớn thời gian trong năm Bitcoin được giao dịch quanh vùng 60.000-70.000 USD trước khi bứt phá lên trên 100.000 USD.

Diễn biến thực tế cho thấy 2024 là một năm bùng nổ của Bitcoin, được thúc đẩy bởi hai sự kiện nền tảng quan trọng: việc SEC phê duyệt các quỹ ETF Bitcoin giao ngay vào đầu năm, giúp dòng vốn tổ chức chính thức tham gia thị trường, và sự kiện halving diễn ra vào tháng 4/2024, làm giảm một nửa phần thưởng khối và thắt chặt nguồn cung. Những yếu tố này kết hợp đã đẩy giá Bitcoin vượt qua đỉnh cũ, khẳng định vị thế của tài sản kỹ thuật số này trong hệ thống tài chính toàn cầu.

2.3.9. Năm 2025

```
#Lọc dữ liệu năm 2025
btc_2025 = df.loc['2025-01-01':'2025-12-31']
print("Số lượng bản ghi 2025:", len(btc_2025))
print(btc_2025['Close'].describe())
```

```
Số lượng bản ghi 2025: 22210
count     22210.000000
mean     100293.959503
std       11521.194940
min       74690.570000
25%       92870.172500
50%      101900.390000
75%      107956.010000
max      124243.320000
Name: Close, dtype: float64
```



Trong năm 2025, dữ liệu giá đóng cửa của Bitcoin cho thấy mức trung bình đạt khoảng 100.294 USD, với độ lệch chuẩn hơn 11.521 USD, phản ánh sự biến động lớn ngay cả khi giá đã đạt vùng sáu chữ số. Giá thấp nhất từ đầu năm ghi nhận ở mức 74.691 USD, trong khi mức cao nhất đạt hơn 124.243 USD, tiếp tục lập đỉnh lịch sử mới. Các phân vị cho thấy cấu trúc giá cao: 25% dữ liệu nằm dưới 92.870 USD, 50% dưới 101.900 USD, và 75% dưới 107.956 USD, nghĩa là phần lớn thời gian trong năm Bitcoin giao dịch quanh vùng trên 90.000 USD.

Diễn biến thực tế cho thấy 2025 là giai đoạn hậu-halving, khi Bitcoin tiếp tục hưởng lợi từ chu kỳ giảm cung và dòng vốn tổ chức. Việc duy trì giá ở vùng cao, vượt xa các mức kỷ lục trước đó, cho thấy sự trưởng thành của thị trường tiền điện tử và sự chấp nhận ngày càng rộng rãi của Bitcoin như một loại tài sản toàn cầu. Dù vậy, mức biến động vẫn ở mức cao, với nhiều nhịp điều chỉnh ngắn hạn trong phạm vi 10-15%, phản ánh đặc trưng rủi ro vốn có của thị trường.

2.3.10.Kết luận

Trong giai đoạn 2017-2025, thị trường Bitcoin thể hiện rõ đặc trưng biến động mạnh và mang tính chu kỳ. Bắt đầu từ mức giá chỉ khoảng 1.000 USD vào đầu năm 2017, Bitcoin đã nhiều lần ghi nhận những đợt tăng trưởng bùng nổ, lập các mốc đỉnh mới, sau đó suy giảm sâu rồi phục hồi. Các dấu mốc quan trọng có thể kể đến như gần 20.000 USD cuối năm 2017, giảm về khoảng 3.000 USD cuối năm 2018, vượt 60.000 USD trong giai đoạn 2020-2021, rơi xuống dưới 20.000 USD trong năm 2022 và tiếp tục lập kỷ lục mới trên 120.000 USD vào năm 2025.

Kết quả thống kê mô tả cho từng năm cho thấy Bitcoin có độ lệch chuẩn lớn, biên độ dao động rộng và mức sụt giảm đáng kể trong các giai đoạn điều chỉnh, cao hơn nhiều so với phần lớn kênh đầu tư truyền thống. Điều này vừa cho thấy tiềm năng sinh lợi vượt trội, vừa nhấn mạnh mức độ rủi ro mà nhà đầu tư phải đối diện khi tham gia thị trường này.

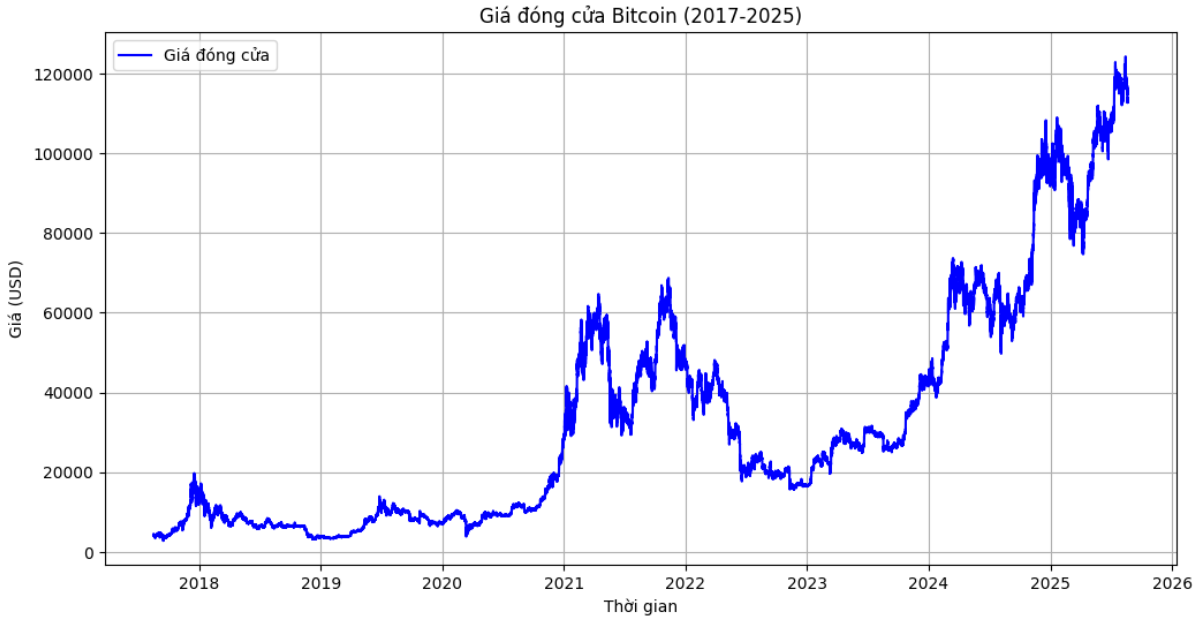
Trong xử lý dữ liệu, các biến động đột ngột quan sát được trong giai đoạn nghiên cứu không thể coi là ngoại lệ, mà là đặc điểm vốn có của Bitcoin. Vì vậy, khi xây dựng mô hình dự báo, đặc biệt với các mô hình học sâu như LSTM, những biến động này cần được giữ nguyên để phản ánh đúng bản chất chuỗi thời gian. Quá trình tiền xử lý dữ liệu chỉ nên tập trung vào việc loại bỏ các giá trị bất hợp lý do lỗi thu thập, đồng thời áp dụng các phương pháp chuẩn hóa như MinMaxScaler hoặc StandardScaler nhằm hỗ trợ mô hình hóa hiệu quả.

Tổng thể, phân tích giai đoạn 2017 - 2025 khẳng định rằng Bitcoin là một loại tài sản có tiềm năng sinh lợi vượt trội nhưng cũng đi kèm rủi ro cực kỳ cao. Chính đặc trưng biến động mạnh và mang tính chu kỳ này cần

được mô hình hóa trong các nghiên cứu, không chỉ nhằm dự báo giá mà còn để dự báo xu hướng tăng - giảm. Đây là cơ sở quan trọng để ứng dụng các mô hình học sâu như LSTM trong việc hỗ trợ ra quyết định đầu tư và quản lý rủi ro.

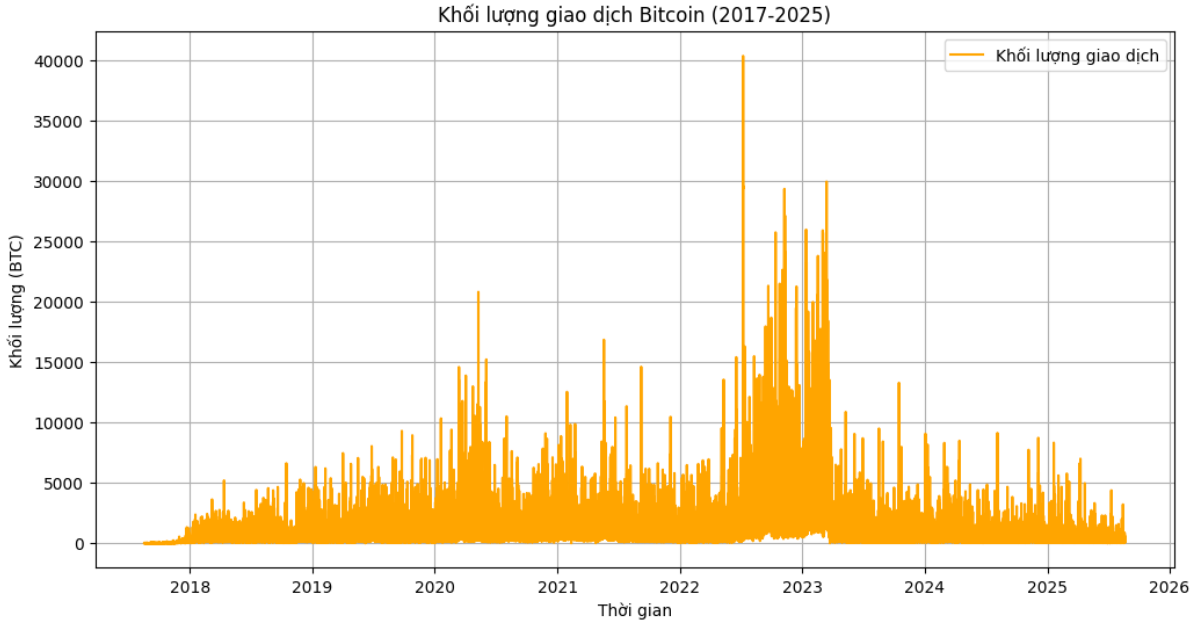
2.4. Trực quan hóa dữ liệu

2.4.1. Biểu đồ đường giá đóng cửa



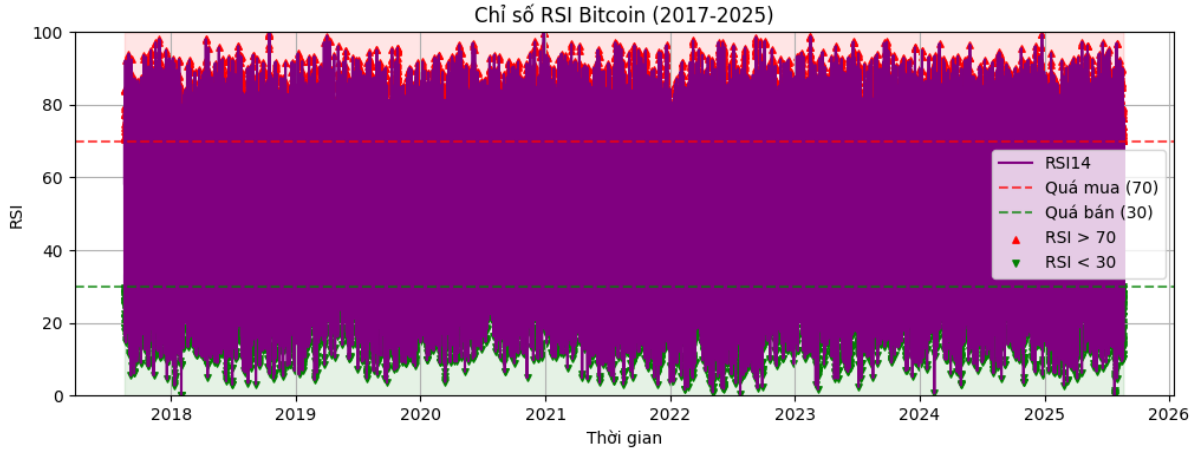
- **Nhận xét:** Biểu đồ giá đóng cửa thể hiện rõ các chu kỳ tăng giảm của Bitcoin giai đoạn 2017–2025. Các pha tăng mạnh nổi bật gồm năm 2021 với hai đỉnh gần 69,000 USD, và chu kỳ 2024–2025 khi giá vượt 120,000 USD nhờ tác động của ETF và halving. Ngược lại, các pha giảm sâu xảy ra vào 2018 (~3,167 USD) và 2022 (~16,548 USD), xen kẽ một số nhịp phục hồi ngắn hạn như năm 2023. Biểu đồ này không chỉ phản ánh xu hướng dài hạn và các mốc biến động quan trọng, mà còn cung cấp dữ liệu đầu vào hữu ích để mô hình LSTM học được các mẫu giá đặc trưng, hỗ trợ dự báo xu hướng chính xác hơn.

2.4.2. Biểu đồ khối lượng giao dịch



- Nhận xét: Biểu đồ khối lượng giao dịch cho thấy sự gia tăng đột biến trong các giai đoạn biến động lớn, đặc biệt là năm 2021 (đỉnh giá) và 2022–2023 (biến động mạnh). Mối liên hệ chặt chẽ giữa khối lượng cao và biến động giá khẳng định vai trò của khối lượng trong việc báo hiệu sức mạnh xu hướng. Đây là một đặc trưng quan trọng giúp mô hình LSTM dự báo xu hướng tăng/giảm chính xác hơn.

2.4.3. Biểu đồ RSI



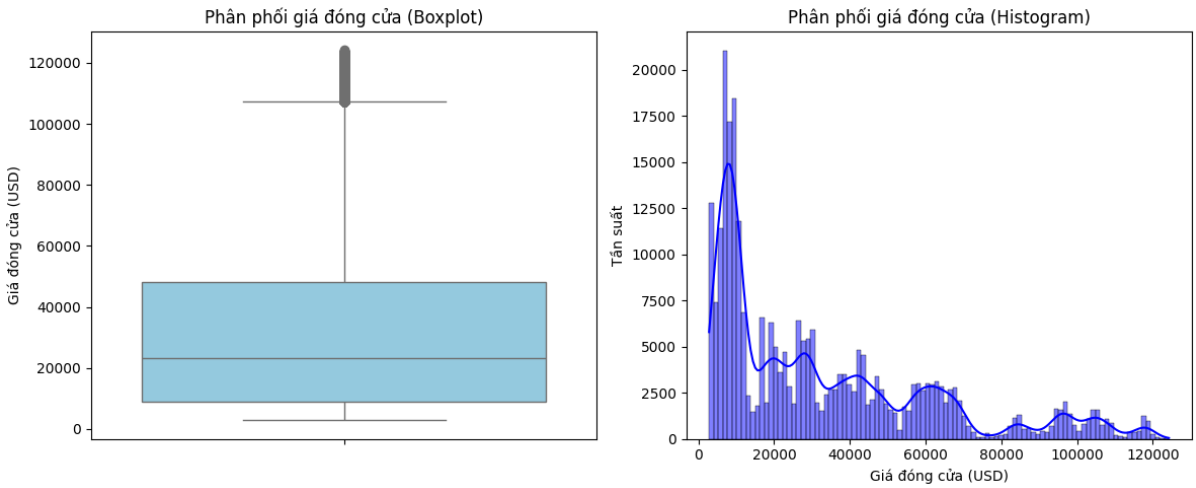
- Nhận xét: Biểu đồ RSI (chu kỳ 14) cho thấy nhiều lần thị trường rơi vào trạng thái quá mua (RSI > 70), đặc biệt trong các chu kỳ tăng giá mạnh năm 2021 và 2024–2025. Ngược lại, các tín hiệu quá bán (RSI < 30) xuất hiện rõ vào giai đoạn suy giảm sâu năm 2018 và 2022. Trong năm 2023, RSI dao động chủ yếu quanh vùng trung tính (40–60), phản ánh trạng thái tích lũy. Những biến động này trùng khớp với các đỉnh và đáy giá, cung

cấp tín hiệu quan trọng về tâm lý thị trường và khả năng đảo chiều. Do đó, RSI là một đặc trưng thiết yếu, hỗ trợ mô hình LSTM nắm bắt xu hướng tăng/giảm một cách hiệu quả hơn.

2.4.4. Biểu đồ đường trung bình động (MA)

- Nhận xét: Biểu đồ MA20 (đường trung bình động 20 nến) làm mượt dữ liệu giá, giúp nhận diện xu hướng ngắn–trung hạn rõ ràng hơn. Các điểm giao cắt giữa giá đóng cửa và MA20 (như năm 2021 và 2024) thường báo hiệu sự thay đổi xu hướng, hỗ trợ xác định các pha tăng hoặc giảm. MA20 là một đặc trưng quan trọng, cung cấp tín hiệu tin cậy để mô hình LSTM học và dự báo xu hướng giá hiệu quả hơn.

2.4.5. Boxplot + Histplot giá đóng cửa



- Biểu đồ Boxplot cho thấy phân phối giá đóng cửa Bitcoin lệch phải, với median quanh 20,000–25,000 USD và phần lớn dữ liệu tập trung dưới

50,000 USD. Các mức giá cực trị trên 100,000 USD (xuất hiện chủ yếu từ 2024–2025) phản ánh xu hướng tăng trưởng mạnh nhờ những sự kiện quan trọng như ETF và halving. Đây là giá trị hợp lệ, không nên loại bỏ, vì cung cấp thông tin cần thiết để mô hình LSTM học các mẫu giá trong giai đoạn tăng trưởng.

- Biểu đồ Histogram bổ sung góc nhìn chi tiết hơn, cho thấy các cụm phân phối giá rõ rệt: mật độ dày đặc dưới 20,000 USD (2017–2020), đỉnh phụ quanh 40,000–60,000 USD (2021–2022) và một cụm giá mới trên 100,000 USD (2024–2025). Các cụm này trùng khớp với các chu kỳ thị trường, minh chứng rằng phân phối giá không ngẫu nhiên mà gắn liền với các pha biến động. Điều này giúp mô hình LSTM dễ dàng nhận diện đặc trưng của từng giai đoạn để cải thiện độ chính xác dự báo xu hướng giá.

2.4.6. Kết Luận

Kết quả phân tích cho thấy Bitcoin trong giai đoạn 2017–2025 mang đặc trưng rõ nét của một thị trường chu kỳ, nơi các pha tăng trưởng bùng nổ và điều chỉnh sâu thường gắn liền với những sự kiện có tính bước ngoặt như halving và sự chấp thuận ETF Bitcoin. Giá, khối lượng giao dịch và các chỉ báo kỹ thuật (RSI, MA20) đồng thuận phản ánh tâm lý thị trường, từ trạng thái hưng phấn quá mức đến giai đoạn suy giảm và tích lũy.

Các công cụ thống kê (Boxplot, Histogram) cho thấy phân phối giá có dạng lệch phải, với những cụm giá đặc trưng tương ứng từng chu kỳ, đặc biệt là sự hình thành mặt bằng giá mới trên 100,000 USD trong giai đoạn 2024-2025. Những mức giá cực trị này không phải nhiễu mà là minh chứng cho sự dịch chuyển cấu trúc thị trường trong dài hạn.

Tổng thể, tập hợp các phân tích trên khẳng định rằng Bitcoin không vận động ngẫu nhiên mà tuân theo những mô hình hành vi lặp lại. Đây chính là nền tảng quan trọng để mô hình LSTM khai thác, học được đặc trưng của từng chu kỳ và từ đó nâng cao khả năng dự báo xu hướng giá với độ tin cậy cao hơn.

3. PHÂN TÍCH DỮ LIỆU ĐƯA VÀO MÔ HÌNH LSTM

3.1. Chọn dữ liệu

3.1.1. Kiểm tra Outlier

Sử dụng phương pháp IQR để kiểm tra outlier trong giá đóng cửa (Close) của Bitcoin (nến 15 phút, 280,253 dòng, 2017-2025):

```python
# Kiểm tra outlier bằng IQR
Q1 = df['Close'].quantile(0.25)
Q3 = df['Close'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
outliers = df[(df['Close'] < lower_bound) | (df['Close'] > upper_bound)]['Close']
print(f"Số lượng outlier (dựa trên IQR): {len(outliers)}")
print(f"Outlier min: {outliers.min()}, max: {outliers.max()}")
```

```
Số lượng outlier (dựa trên IQR): 6163
Outlier min: 107298.42, max: 124243.32
```

[12]
```python
# Kiểm tra phân bố outlier theo năm
df['Year'] = df.index.year
outliers = df[(df['Close'] >= 107298.42) & (df['Close'] <= 124243.32)]
print(outliers.groupby('Year').size())
```

```
Year
2024      17
2025    6146
dtype: int64
```

Phân tích outlier bằng phương pháp IQR cho thấy khoảng giá 107,298.42 - 124,243.32 USD có 6,163 điểm, trong đó 6,146 điểm thuộc năm 2025 và chỉ 17 điểm xuất hiện ở cuối năm 2024. Điều này chứng tỏ các giá trị này không phải nhiễu mà phản ánh xu hướng tăng giá mạnh (bull run) của Bitcoin trong giai đoạn 2025. Vì vậy, việc loại bỏ chúng sẽ làm mất thông tin xu hướng quan trọng, đặc biệt khi phân phối giá lệch phải khiến IQR không còn phù hợp. Bởi vậy, dữ liệu năm 2025 được lựa chọn để huấn luyện mô hình LSTM nhằm nắm bắt chính xác đặc tính mới của thị trường.

```python
df = df[df['Year'] >= 2025]
print(df.shape)
print(df.tail())
```

```
(22210, 9)
                  open_time       Open       High        Low      Close  \
280228  2025-08-20 07:15:00  113722.84  113837.93  113720.86  113720.87
280229  2025-08-20 07:30:00  113720.87  113720.87  113588.00  113612.01
280230  2025-08-20 07:45:00  113612.01  113672.74  113432.00  113490.13
280231  2025-08-20 08:00:00  113490.14  113609.94  113401.00  113609.94
280232  2025-08-20 08:15:00  113609.94  113854.34  113609.93  113854.33

           Volume  Year         MA20      RSI14
280228   91.20673  2025  113581.3400  59.885407
280229  127.34973  2025  113593.8795  50.157108
280230  124.31949  2025  113596.4715  44.318596
280231  119.43721  2025  113607.8475  52.546550
280232  186.23173  2025  113620.0785  58.255708
```

3.2. Đánh nhãn dữ liệu

Để xây dựng tập huấn luyện cho hai bài toán, dữ liệu được gán nhãn như sau:

- Bài toán hồi quy (dự báo giá):
  Mục tiêu là giá đóng cửa ở bước kế tiếp:

$$\mathbf{y_t = Close_{t+1}}$$

  Dữ liệu đầu vào X_t là các đặc trưng tại thời điểm t.

```
features = ['Open', 'High', 'Low', 'Close', 'Volume', 'MA20', 'RSI14']

## hồi quy
y_price = df['Close'].shift(-1)
y_price = y_price.dropna()
X_price  = df[features].loc[y_price.index]
```

- Bài toán phân loại (dự đoán xu hướng):
  Mục tiêu là nhãn xu hướng (1 = tăng, 0 = giảm)
  Dữ liệu được gán nhãn xu hướng bằng cách so sánh trung bình động 12 phiên (MA12) hiện tại với giá trị sau 12 phiên (~3 tiếng).
  - Nếu MA12 trong tương lai lớn hơn MA12 hiện tại → nhãn 1 (xu hướng tăng).
  - Ngược lại → nhãn 0 (xu hướng giảm/không tăng).

```
# phân loại
df['Close_MA12'] = df['Close'].rolling(window=12).mean()
df['Trend'] = (df['Close_MA12'].shift(-12) > df['Close_MA12']).astype(int)
y_trend = df['Trend'].dropna()
X_trend = df[features].loc[y_trend.index]
```

3.3. Kiểm định các đặc trưng đưa vào mô hình

Phần này kiểm định và lựa chọn các đặc trưng từ dữ liệu Bitcoin (BTC/USDT, nến 15 phút, 2017–2025) để đưa vào mô hình LSTM cho hai bài toán: dự báo giá (hồi quy) và dự đoán xu hướng tăng/giảm (phân loại). Các đặc trưng được kiểm định bằng phân tích tương quan, Mutual Information (MI), và phân tích đặc điểm dữ liệu, đảm bảo tối ưu hóa hiệu suất mô hình.

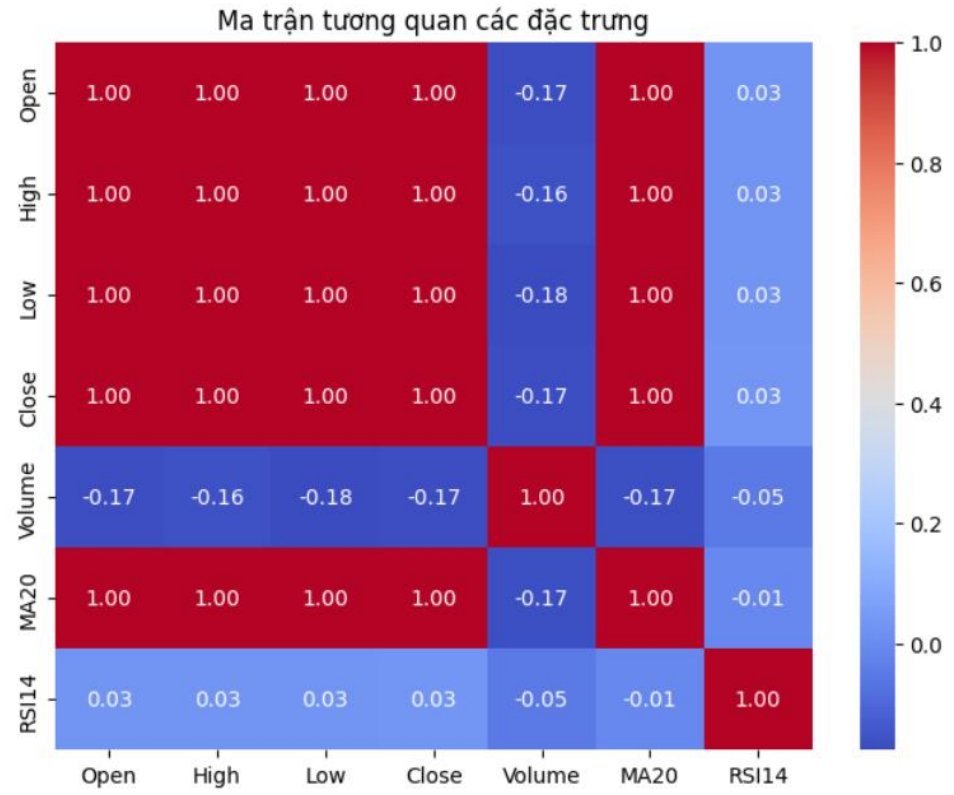
3.3.1. Cơ sở lý thuyết

- Tương quan (Correlation): đo mối quan hệ tuyến tính giữa các biến. Hệ số gần 1 hoặc -1 cho thấy liên hệ mạnh, gần 0 là yếu. Đơn giản nhưng chỉ bắt được quan hệ tuyến tính.

- Mutual Information (MI): đo mức độ đặc trưng giảm bất định cho mục tiêu, phát hiện cả quan hệ tuyến tính và phi tuyến. MI càng cao → đặc trưng càng quan trọng.

3.3.2. Phân tích tương quan

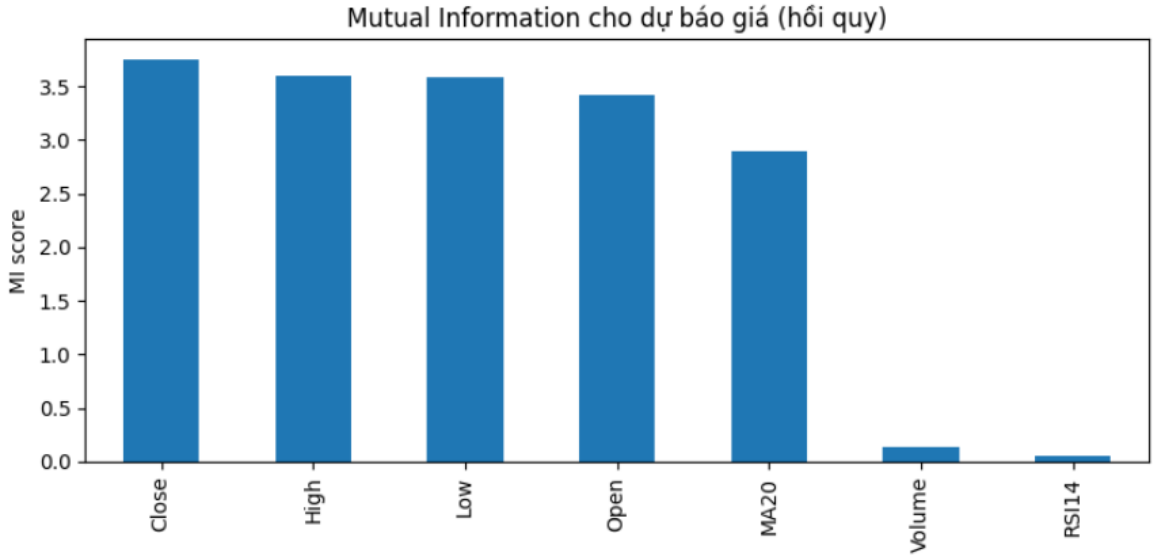
```
#Phân tích tương quan
corr = df[features].corr()

plt.figure(figsize=(8,6))
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Ma trận tương quan các đặc trưng")
plt.show()

print("=== Ma trận tương quan ===")
print(corr)
```

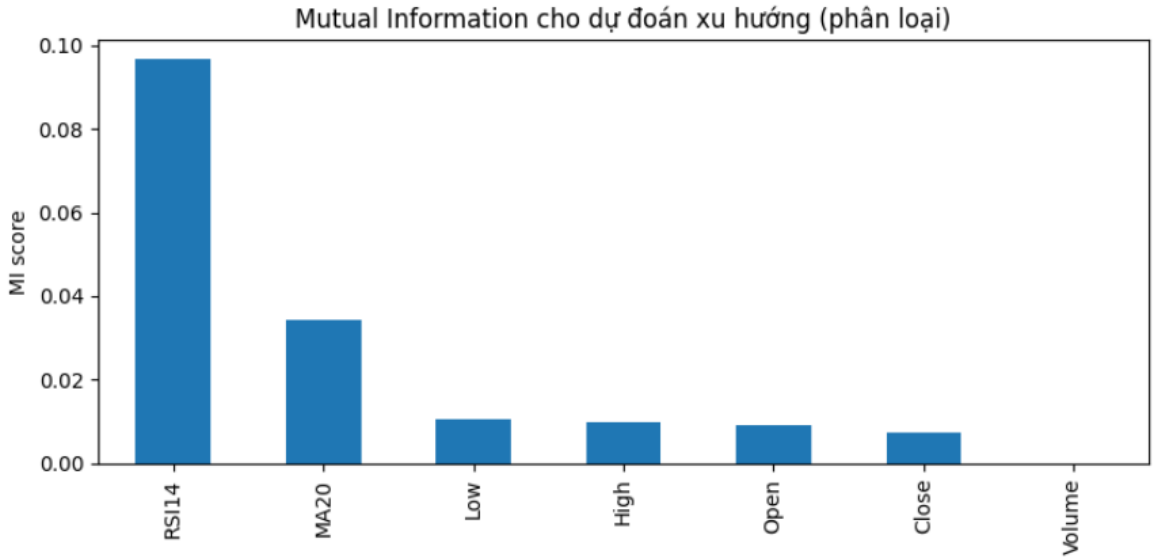



```
=== Ma trận tương quan ===
              Open      High       Low     Close    Volume      MA20     RSI14
Open      1.000000  0.999875  0.999868  0.999794 -0.168192  0.998960  0.025425
High      0.999875  1.000000  0.999773  0.999885 -0.159677  0.998880  0.027582
Low       0.999868  0.999773  1.000000  0.999883 -0.177093  0.998784  0.029063
Close     0.999794  0.999885  0.999883  1.000000 -0.168398  0.998783  0.030444
Volume -0.168192 -0.159677 -0.177093 -0.168398  1.000000 -0.165295 -0.050885
MA20      0.998960  0.998880  0.998784  0.998783 -0.165295  1.000000 -0.008916
RSI14     0.025425  0.027582  0.029063  0.030444 -0.050885 -0.008916  1.000000
```

3.3.3. Phân tích Mutual Information (MI)

```python
from sklearn.feature_selection import mutual_info_regression, mutual_info_classif
# Mutual Information (MI)

# MI cho hồi quy
mi_price = mutual_info_regression(X_price.values, y_price.values, random_state=42)
mi_price_series = pd.Series(mi_price, index=features).sort_values(ascending=False)

# MI cho phân loại
y_trend = y_trend.astype(int)# chuyển lại int 0/1
mi_trend = mutual_info_classif(X_trend.values, y_trend.values, random_state=42)
mi_trend_series = pd.Series(mi_trend, index=features).sort_values(ascending=False)

# Vẽ biểu đồ MI
plt.figure(figsize=(8,4))
mi_price_series.plot(kind='bar')
plt.title("Mutual Information cho dự báo giá (hồi quy)")
plt.ylabel("MI score")
plt.tight_layout()
plt.show()

plt.figure(figsize=(8,4))
mi_trend_series.plot(kind='bar')
plt.title("Mutual Information cho dự đoán xu hướng (phân loại)")
plt.ylabel("MI score")
plt.tight_layout()
plt.show()

print("\n=== Shapes ===")
print("X_price:", X_price.shape, " y_price:", y_price.shape)
print("X_trend:", X_trend.shape, " y_trend:", y_trend.shape)

print("\n=== MI cho dự báo giá (Close[t+1]) ===")
print(mi_price_series)

print("\n=== MI cho dự đoán xu hướng (Trend) ===")
print(mi_trend_series)
```






```
=== Shapes ===
X_price: (22209, 7)  y_price: (22209,)
X_trend: (22210, 7)  y_trend: (22210,)

=== MI cho dự báo giá (Close[t+1]) ===
Close     3.749718
High      3.593554
Low       3.576742
Open      3.422731
MA20      2.895316
Volume    0.135215
RSI14     0.054958
dtype: float64

=== MI cho dự đoán xu hướng (Trend) ===
RSI14     0.096576
MA20      0.034294
Low       0.010724
High      0.009784
Open      0.009240
Close     0.007316
Volume    0.000000
dtype: float64
```

3.3.4. Nhận xét và lựa chọn đặc trưng cho từng bài toán

- **Bài toán dự báo giá (hồi quy)**
  - Tương quan (corr):
    - Open, High, Low, Close, MA20 đều có tương quan rất cao (>0.95) -> cung cấp thông tin gần giống nhau nhưng vẫn bổ sung chi tiết về biến động giá.
    - Volume và RSI14 có tương quan thấp (~0.1 và ~0.03) -> gần như độc lập với giá.

- Mutual Information (MI):
  - Close (3.75), High (3.59), Low (3.57), Open (3.42), MA20 (2.89) có MI rất cao -> là các đặc trưng quan trọng.
  - Volume (0.13) đóng góp nhỏ nhưng vẫn thêm thông tin về sức mạnh thị trường.
  - RSI14 (0.05) gần như không đóng góp.

  Chọn đặc trưng cho hồi quy: Open, High, Low, Close, MA20, Volume.

- **Bài toán dự đoán xu hướng (phân loại):**
  - Tương quan (corr):
    - Open, High, Low, Close, MA20 gần như trùng lặp nhau (>0.95). RSI14 ít tương quan với giá -> thông tin độc lập, hữu ích cho phân loại.
    - Volume không tương quan -> ít giá trị.

  - Mutual Information (MI):
    RSI14 (0.097) và MA20 (0.034) có MI cao nhất -> đặc trưng quan trọng nhất cho phân loại.
    Nhóm giá (Low 0.011, High 0.0098, Open 0.0092, Close 0.0073) có MI thấp -> chỉ mang tính bổ sung.

    Chọn đặc trưng cho phân loại: RSI14, MA20, Close.

3.4. Chia tập Train/Test

3.4.1. Đánh lại nhãn

Đánh lại nhãn với các đặc trưng mới sau khi đã chọn lọc qua:

```
#Đánh lại nhãn

# Các đặc trưng đã chọn
features_reg = ['Open', 'High', 'Low', 'Close','Volume','MA20']  # cho hồi quy
features_cls = ['RSI14', 'MA20', 'Close']  # cho phân loại

#Nhãn cho hồi quy (dự báo Close[t+1])
y_price = df['Close'].shift(-1).dropna()
X_price = df[features_reg].loc[y_price.index]

#Nhãn cho phân loại (xu hướng tăng/giảm)
df['Close_MA12'] = df['Close'].rolling(window=12).mean()
df['Trend'] = (df['Close_MA12'].shift(-12) > df['Close_MA12']).astype(int)
y_trend = df['Trend'].dropna()
X_trend = df[features_cls].loc[y_trend.index]
```

3.4.2. Chia tập theo tỷ lệ 80/20

```python
# Chia tập theo tỷ lệ 80/20
split_ratio = 0.8
split_idx_price = int(len(X_price) * split_ratio)
split_idx_trend = int(len(X_trend) * split_ratio)

# Hồi quy
X_train_reg, X_test_reg = X_price[:split_idx_price], X_price[split_idx_price:]
y_train_reg, y_test_reg = y_price[:split_idx_price], y_price[split_idx_price:]

# Phân loại
X_train_cls, X_test_cls = X_trend[:split_idx_trend], X_trend[split_idx_trend:]
y_train_cls, y_test_cls = y_trend[:split_idx_trend], y_trend[split_idx_trend:]

print("Bộ Train/Test hồi quy:")
print(X_train_reg.shape, y_train_reg.shape, X_test_reg.shape, y_test_reg.shape)

print("\nBộ Train/Test phân loại:")
print(X_train_cls.shape, y_train_cls.shape, X_test_cls.shape, y_test_cls.shape)
```

```
Bộ Train/Test hồi quy:
(224185, 6) (224185,) (56047, 6) (56047,)

Bộ Train/Test phân loại:
(224186, 3) (224186,) (56047, 3) (56047,)
```

3.4.3. Chuẩn hóa dữ liệu

Trước khi đưa dữ liệu vào LSTM, cần chuẩn hóa vì các đặc trưng có thang đo rất khác nhau (giá BTC hàng chục nghìn USD, RSI 0–100, Volume dao động lớn). Nếu không chuẩn hóa, mô hình sẽ thiên lệch về đặc trưng có giá trị lớn và dễ gặp hiện tượng gradient vanishing/exploding.

Trong báo cáo, dữ liệu được chuẩn hoá bằng hai phương pháp: MinMaxScaler (đưa dữ liệu về khoảng [0,1]) và StandardScaler (chuẩn hoá dữ liệu theo phân phối chuẩn với trung bình bằng 0 và độ lệch chuẩn bằng 1):

```python
#Chuẩn hóa dữ liệu
from sklearn.preprocessing import MinMaxScaler, StandardScaler, RobustScaler

# scaler cho hồi quy
scaler_reg = StandardScaler()
X_train_reg_scaled = scaler_reg.fit_transform(X_train_reg)
X_test_reg_scaled  = scaler_reg.transform(X_test_reg)

scaler_y = MinMaxScaler(feature_range=(0,1))
y_train_reg_scaled = scaler_y.fit_transform(y_train_reg.values.reshape(-1,1)).ravel()
y_test_reg_scaled  = scaler_y.transform(y_test_reg.values.reshape(-1,1)).ravel()

#scaler cho phân loại
scaler_cls = StandardScaler()
X_train_cls_scaled = scaler_cls.fit_transform(X_train_cls)
X_test_cls_scaled  = scaler_cls.transform(X_test_cls)
```

3.5. Xây dựng mô hình LSTM

3.5.1. Cơ sở lý thuyết

- LSTM (Long Short-Term Memory) là mạng nơ-ron hồi quy (RNN) được thiết kế để học các chuỗi dữ liệu có quan hệ theo thời gian.
- Khác với các mô hình truyền thống chỉ học từng điểm dữ liệu độc lập, LSTM có thể ghi nhớ thông tin từ nhiều bước thời gian trước đó để dự đoán giá trị ở bước tiếp theo.
- Do đó, dữ liệu đầu vào cho LSTM phải có dạng 3 chiều:

  [samples, timesteps, features]

  - samples: số lượng mẫu (số lần huấn luyện).
  - timesteps: số bước thời gian quan sát (window size).
  - features: số đặc trưng tại mỗi thời điểm.

3.5.2. Sliding window (Cửa sổ trượt)

Trong bài toán tài chính, giá hiện tại phụ thuộc vào nhiều giá trị trong quá khứ, không chỉ giá ngay trước đó.

Kỹ thuật sliding window chia dữ liệu thành các chuỗi con liên tiếp, giúp mô hình học được:

- Xu hướng (trend) ngắn hạn và dài hạn.
- Mẫu lặp (patterns) của thị trường, ví dụ khi RSI cao + Volume tăng → dễ đảo chiều.

```python
# Tạo bộ dữ liệu dạng sliding window
def create_sequences_reg(X, y, timesteps=32):
    Xs, ys = [], []
    for i in range(len(X) - timesteps):
        Xs.append(X[i:i+timesteps])
        ys.append(y[i+timesteps-1])
    return np.array(Xs), np.array(ys)

def create_sequences_cls(X, y, timesteps=96):
    Xs, ys = [], []
    for i in range(len(X) - timesteps):
        Xs.append(X[i:i+timesteps])
        ys.append(y[i+timesteps-1])
    return np.array(Xs), np.array(ys)
```

```python
# Hồi quy
X_train_reg_seq, y_train_reg_seq = create_sequences_reg(X_train_reg_scaled, y_train_reg_scaled)
X_test_reg_seq, y_test_reg_seq = create_sequences_reg(X_test_reg_scaled, y_test_reg_scaled)

# Phân loại
X_train_cls_seq, y_train_cls_seq = create_sequences_cls(X_train_cls_scaled, y_train_cls.values)
X_test_cls_seq, y_test_cls_seq = create_sequences_cls(X_test_cls_scaled, y_test_cls.values)

print("Bộ dữ liệu hồi quy:", X_train_reg_seq.shape, y_train_reg_seq.shape, X_test_reg_seq.shape, y_test_reg_seq.shape)
print("Bộ dữ liệu phân loại:", X_train_cls_seq.shape, y_train_cls_seq.shape, X_test_cls_seq.shape, y_test_cls_seq.shape)
```

```
Bộ dữ liệu hồi quy: (17735, 32, 6) (17735,) (4410, 32, 6) (4410,)
Bộ dữ liệu phân loại: (17672, 96, 3) (17672,) (4346, 96, 3) (4346,)
```

3.5.3. Mô hình LSTM dự báo giá (Hồi quy)

- Callback trong huấn luyện mô hình

Trong quá trình huấn luyện, mô hình sử dụng EarlyStopping để dừng sớm khi val_loss không còn cải thiện (10 epoch) và khôi phục trọng số tốt nhất, đồng thời dùng ReduceLROnPlateau để giảm learning rate (0.2 lần, min=1e-5) khi mô hình chững lại (5 epoch), giúp tránh overfitting và tối ưu hiệu quả hơn.

```
# Callback EarlyStopping
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
early_stop = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=0.00001)
callbacks = [early_stop, reduce_lr]
```

- Xây dựng mô hình
  - LSTM(128, return_sequences=True): lớp LSTM đầu tiên học đặc trưng chuỗi, trả về toàn bộ chuỗi ẩn để truyền tiếp.
  - Dropout(0.3): giảm overfitting bằng cách ngẫu nhiên bỏ 30% neuron trong quá trình huấn luyện.
  - LSTM(64, return_sequences=False): lớp LSTM thứ hai học sâu hơn nhưng chỉ trả về trạng thái cuối cùng.
  - Dense(32, relu): lớp fully-connected để trích chọn đặc trưng phi tuyến.
  - Dense(1): lớp đầu ra, trả về một giá trị liên tục (giá Close dự báo).

- Mô hình sử dụng Adam (lr=0.0001) để tối ưu, hàm mất mát MSE (Mean Squared Error) và đánh giá bằng MAE (Mean Absolute Error).

```
# Xây dựng mô hình LSTM cho hồi quy
tf.keras.backend.clear_session()
from tensorflow.keras.optimizers import Adam

model_reg = Sequential([
    LSTM(128, return_sequences=True, input_shape=(X_train_reg_seq.shape[1], X_train_reg_seq.shape[2])),
    Dropout(0.3),
    LSTM(64, return_sequences=False),
    Dense(32, activation='relu'),
    Dense(1)
])

model_reg.compile(optimizer=Adam(learning_rate=0.0001), loss='mean_squared_error', metrics=['mae'])
model_reg.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 32, 128)	69,120
dropout (Dropout)	(None, 32, 128)	0
lstm_1 (LSTM)	(None, 64)	49,408
dense (Dense)	(None, 32)	2,080
dense_1 (Dense)	(None, 1)	33

```
Total params: 120,641 (471.25 KB)

Trainable params: 120,641 (471.25 KB)

Non-trainable params: 0 (0.00 B)
```

- Huấn luyện mô hình

```
# Huấn luyện mô hình dự báo giá
tf.keras.backend.clear_session()
start_time = time.time()

history_reg = model_reg.fit(
    X_train_reg_seq, y_train_reg_seq,
    validation_data=(X_test_reg_seq, y_test_reg_seq),
    epochs=200, batch_size=64,
    shuffle=False,
    callbacks=callbacks,
    verbose=1
)

end_time = time.time()
print(f"Thời gian huấn luyện: {end_time - start_time:.2f} giây")
```

Mô hình LSTM được huấn luyện với 200 epoch, batch size = 64, và shuffle=False để giữ nguyên thứ tự dữ liệu chuỗi thời gian. Tập validation giúp theo dõi hiệu năng, EarlyStopping dừng sớm khi val_loss không cải thiện, ReduceLROnPlateau tự động giảm learning rate khi mô hình chững lại. Thời gian huấn luyện cũng được ghi nhận để đánh giá chi phí tính toán.

- Thời gian huấn luyện: 706.60 giây

3.5.4. Mô hình LSTM dự báo tăng/giảm (Phân loại)

- Xây dựng mô hình

  Mô hình LSTM cho phân loại xu hướng giá gồm 2 tầng LSTM (128 và 64 nút) để học đặc trưng chuỗi, kết hợp Dropout 0.3 nhằm giảm overfitting. Sau đó là tầng Dense 32 nút với hàm kích hoạt ReLU để trích xuất đặc trưng phi tuyến. Tầng đầu ra dùng sigmoid cho xác suất xu hướng (0 hoặc 1). Mô hình được biên dịch với Adam (lr=0.0001), hàm mất mát binary_crossentropy và đánh giá bằng accuracy.

```
# Xây dựng mô hình LSTM cho phân loại
from tensorflow.keras.optimizers import Adam


model_cls = Sequential([
    LSTM(128, return_sequences=True, input_shape=(X_train_cls_seq.shape[1], X_train_cls_seq.shape[2])),
    Dropout(0.3),
    LSTM(64),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid')  # đầu ra xác suất [0,1]
])

model_cls.compile(optimizer=Adam(learning_rate=0.0001), loss='binary_crossentropy', metrics=['accuracy'])
model_cls.summary()
```

```
/home/bush/.conda/envs/tf-gpu/lib/python3.12/site-packages/keras/src/layers/rnn/rnn.py:199: UserWarning: Do not pass an `input_shape`/`input_
dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(**kwargs)
```

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 96, 128)	67,584
dropout (Dropout)	(None, 96, 128)	0
lstm_1 (LSTM)	(None, 64)	49,408
dense (Dense)	(None, 32)	2,080
dense_1 (Dense)	(None, 1)	33

Total params: 119,105 (465.25 KB)

- Huấn luyện mô hình

  Mô hình LSTM phân loại được huấn luyện với 150 epoch và batch size = 64. Tham số shuffle=False đảm bảo dữ liệu chuỗi thời gian giữ nguyên thứ tự. Tập validation được dùng để theo dõi hiệu năng trong quá trình huấn luyện. Các callback EarlyStopping và ReduceLROnPlateau giúp dừng sớm khi mô hình không cải thiện và tự động giảm learning rate khi cần thiết, nhằm tránh overfitting và tối ưu tốc độ hội tụ. Đồng thời, thời gian huấn luyện được ghi nhận để đánh giá chi phí tính toán của mô hình.

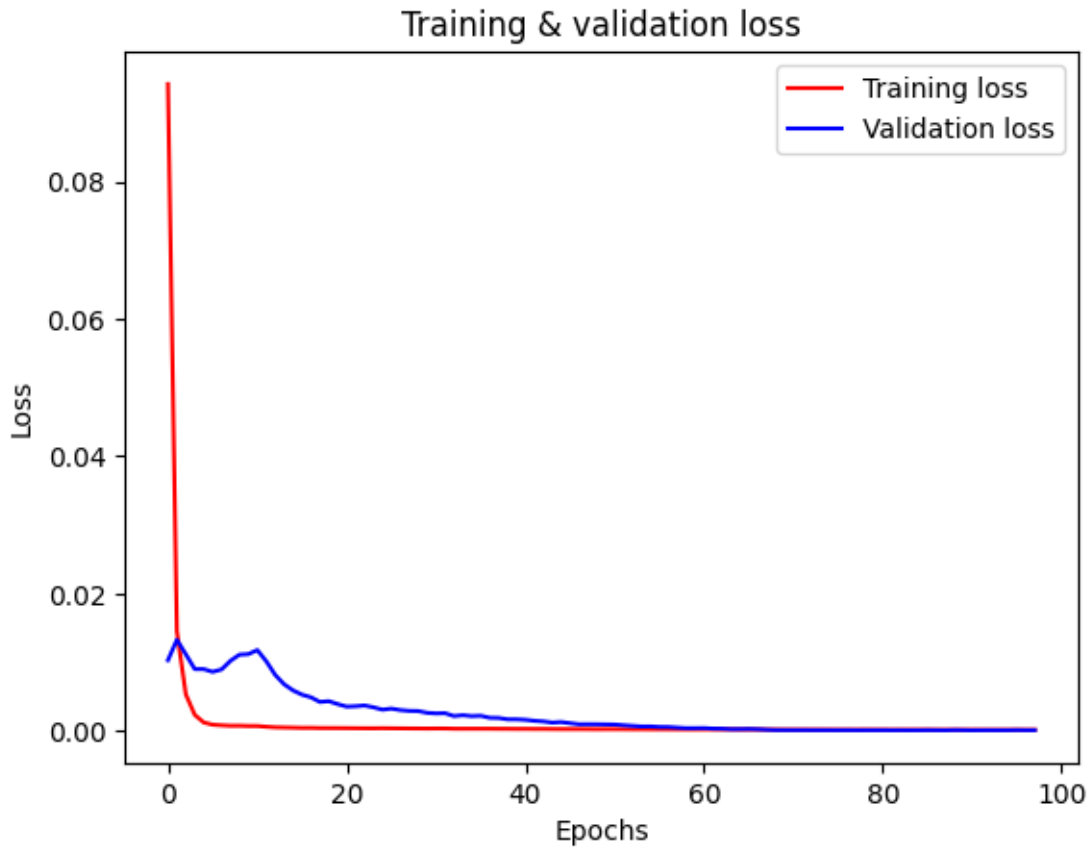
```python
# Huấn luyện mô hình phân loại
tf.keras.backend.clear_session()
start_time = time.time()

history_cls = model_cls.fit(
    X_train_cls_seq, y_train_cls_seq,
    validation_data=(X_test_cls_seq, y_test_cls_seq),
    epochs=150, batch_size=64,
    shuffle=False,
    callbacks=callbacks,
    verbose=1
)

end_time = time.time()
print(f"Thời gian huấn luyện: {end_time - start_time:.2f} giây")
```

- Thời gian huấn luyện: 214.76 giây

4. ĐÁNH GIÁ HIỆU QUẢ MÔ HÌNH LSTM

4.1. Dự Báo Giá

4.1.1. Biểu đồ Loss qua các Epoch



Biểu đồ training loss và validation loss cho thấy mô hình hội tụ rất tốt.

- Ở các epoch đầu (1–10), loss giảm mạnh từ 0.094 xuống < 0.01, chứng tỏ LSTM học nhanh từ dữ liệu chuỗi thời gian.
- Từ epoch 20 trở đi, validation loss tiếp tục giảm và đạt mức cực nhỏ (~1e-4), gần như song song với training loss.
- Nhờ EarlyStopping và ReduceLROnPlateau, quá trình huấn luyện được kiểm soát, tránh overfitting hoặc dao động quá mức.
- Tổng thời gian huấn luyện ~706 giây với 98 epoch, dừng sớm khi mô hình không còn cải thiện đáng kể.

Nhận xét: Loss giảm đều, hội tụ tốt, mô hình hồi quy LSTM học được quy luật giá từ dữ liệu.

4.1.2. Tính các chỉ số

```
# inverse_transform dự báo để so sánh trên thang đo gốc:
y_pred_scaled = model_reg.predict(X_test_reg_seq)
y_pred = scaler_y.inverse_transform(y_pred_scaled)
y_true = scaler_y.inverse_transform(y_test_reg_seq.reshape(-1,1))
```

```
138/138 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step
```

```
# Tính các metric
mse = mean_squared_error(y_true, y_pred)
rmse = np.sqrt(mse)
mae = mean_absolute_error(y_true, y_pred)
r2 = r2_score(y_true, y_pred)

# In kết quả
print(f"MSE: {mse:.2f}")
print(f"RMSE: {rmse:.2f}")
print(f"MAE: {mae:.2f}")
print(f"R²: {r2:.2f}")
```

```
MSE: 167996.99
RMSE: 409.87
MAE: 294.18
R²: 0.98
```

Sau khi huấn luyện, mô hình được đánh giá bằng các chỉ số hồi quy:

- MSE (Mean Squared Error): 167,996.99 → Là sai số bình phương trung bình giữa giá trị dự đoán và giá trị thực tế. Chỉ số này càng nhỏ thì mô hình càng chính xác. Tuy nhiên, vì bình phương sai số nên MSE nhạy với các outlier (sai số lớn sẽ bị phóng đại).
- RMSE (Root Mean Squared Error): 409.87 USD → Là căn bậc hai của MSE, có cùng đơn vị với dữ liệu gốc (USD). RMSE phản ánh

sai số trung bình giữa giá dự báo và giá thực tế. Với giá BTC ~113,000 USD, RMSE ~410 USD nghĩa là mô hình chỉ sai lệch khoảng 0.3–0.4%, được xem là rất nhỏ trong dự báo tài chính.

- MAE (Mean Absolute Error): 294.18 USD → Là sai số tuyệt đối trung bình giữa giá dự đoán và giá thực tế. Khác với MSE, MAE không bình phương sai số nên ít bị ảnh hưởng bởi outlier, phản ánh trực tiếp sai số trung bình mỗi lần dự báo.
- R^2 Score (Coefficient of Determination): 0.98 → Đo mức độ mô hình giải thích được biến thiên của dữ liệu thực tế. $R^2 = 0.98$ nghĩa là mô hình giải thích được 98% biến động giá, chỉ còn 2% là do yếu tố ngẫu nhiên hoặc sai số. Giá trị R^2 càng gần 1 càng tốt.
  Nhận xét: Các chỉ số cho thấy mô hình LSTM hồi quy có độ chính xác rất cao. Sai số dự báo nhỏ, ổn định, và mô hình giải thích hầu hết biến động giá BTC trong giai đoạn test.

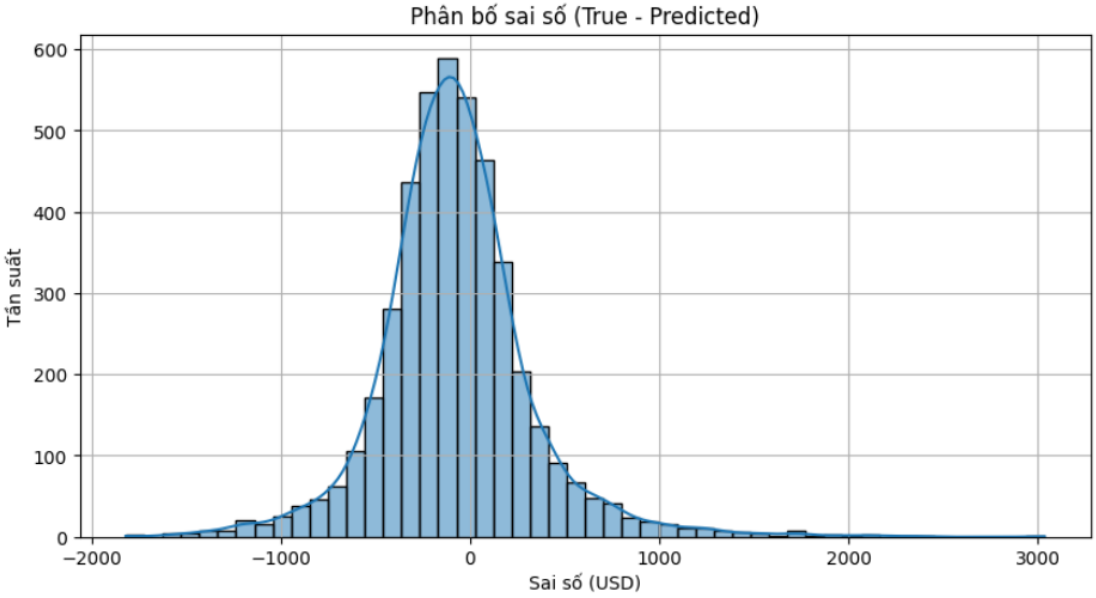
4.1.3. Dự đoán giá Bitcoin so với Giá thực tế

- Biểu đồ cho thấy đường dự đoán (màu cam) gần như trùng với đường giá thực tế (màu xanh).
- Mô hình bám sát xu hướng giá trong giai đoạn test (tháng 8/2025).
- Hạn chế:
  - Ở đỉnh giá (local maxima), mô hình dự đoán thấp hơn thực tế.
  - Ở đáy giá (local minima), mô hình dự đoán cao hơn thực tế.
  - Đây là đặc điểm thường gặp của LSTM: phản ứng chậm với biến động đột ngột.

Nhận xét: Mô hình mạnh trong dự đoán xu hướng chung.

4.1.4. Phân bố sai số

```
# Phân bố sai số
errors = y_true.flatten() - y_pred.flatten()
plt.figure(figsize=(10, 5))
sns.histplot(errors, bins=50, kde=True)
plt.title('Phân bố sai số (True - Predicted)')
plt.xlabel('Sai số (USD)')
plt.ylabel('Tần suất')
plt.grid(True)
plt.show()
```



- Biểu đồ sai số (True - Predicted) phân bố quanh 0, gần với chuẩn Gaussian.
- Sai số phổ biến trong khoảng ±300 - 400 USD.
- Một số outliers có sai số >1000 USD, xảy ra khi thị trường biến động mạnh.

Nhận xét: Sai số nhỏ, ổn định, mô hình đáng tin cậy trong điều kiện bình thường.

4.2. Dự Báo Tăng/Giảm

4.2.1. Biểu đồ Loss qua các Epoch

- Training loss giảm từ 0.68 xuống ~0.61.
- Validation loss ban đầu giảm nhẹ, sau đó dao động quanh 0.65–0.70.
- Mô hình hội tụ nhưng validation loss không giảm sâu, khó phân biệt xu hướng tăng/giảm.
- Thời gian huấn luyện ~215 giây với 10 epoch (EarlyStopping dừng sớm).

Nhận xét: Mô hình phân loại có dấu hiệu underfitting hoặc dữ liệu chưa đủ đặc trưng.

4.2.2. Các chỉ số

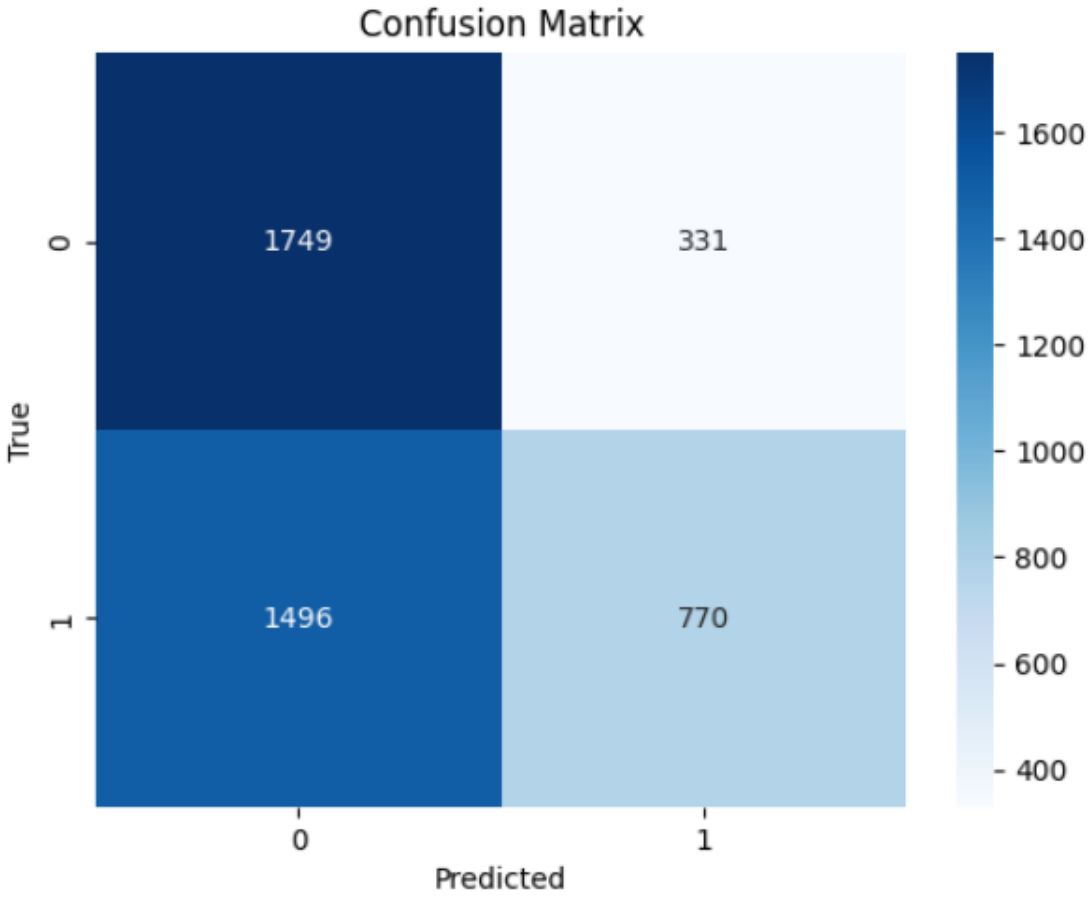
```
136/136 ━━━━━━━━━━━━━━━━━━━━ 2s 16ms/step
              precision    recall  f1-score   support

           0       0.54      0.84      0.66      2080
           1       0.70      0.34      0.46      2266

    accuracy                           0.58      4346
   macro avg       0.62      0.59      0.56      4346
weighted avg       0.62      0.58      0.55      4346
```

Kết quả classification_report:

- Accuracy: 0.58 → cao hơn baseline ngẫu nhiên (0.5) một chút.
- Precision (class 1 – tăng): 0.70 → khi dự đoán tăng, 70% là đúng.
- Recall (class 1 – tăng): 0.34 → chỉ phát hiện 34% trường hợp tăng thực sự.
- F1-score (class 1): 0.46 → thấp.
- Class 0 (giảm): precision 0.54, recall 0.84 → dự đoán giảm tốt hơn tăng.

Nhận xét: Mô hình thiên về dự đoán giảm, chưa đủ tin cậy để dùng cho trading.

4.2.3. Kiểm tra phân bố dự đoán



- True giảm (0): 2080 mẫu → 1749 đúng, 331 sai.
- True tăng (1): 2266 mẫu → 770 đúng, 1496 sai.
- Accuracy tổng thể = 58%.

Nhận xét: Mô hình có xu hướng dự đoán "giảm" nhiều hơn, gây mất cân bằng precision-recall.

5. KẾT LUẬN

5.1. Dự Báo Giá

Mô hình LSTM hồi quy cho kết quả rất ấn tượng:

- $R^2 = 0.98$: giải thích được tới 98% biến thiên giá, chứng tỏ mô hình đã học tốt các quy luật ẩn trong dữ liệu.
- MAE = 294 USD, RMSE = 410 USD: sai số tuyệt đối nhỏ so với mức giá trung bình trên 100,000 USD, tương đương sai số chỉ khoảng 0.3-0.4%.
- Biểu đồ so sánh giá dự báo và giá thực tế cho thấy mô hình bám sát xu hướng thị trường, dự đoán chính xác cả các giai đoạn tăng và giảm.

Ý nghĩa: Mô hình hồi quy hoàn toàn có thể ứng dụng như một công cụ hỗ trợ dự báo ngắn hạn trong giao dịch, giúp nhà đầu tư ước lượng giá tương lai gần và từ đó đưa ra quyết định hợp lý hơn.

Hạn chế: Tuy nhiên, ở những pha biến động cực mạnh, mô hình dự đoán chậm và thường ước lượng thấp hơn đỉnh hoặc cao hơn đáy. Đây là nhược điểm phổ biến của LSTM khi xử lý dữ liệu tài chính giàu nhiễu và biến động đột ngột.

5.2. Dự Báo Tăng/Giảm

Kết quả của mô hình phân loại xu hướng kém khả quan hơn:

- Accuracy = 58%: chỉ nhỉnh hơn so với dự đoán ngẫu nhiên (50%), chưa đạt mức ứng dụng thực tiễn.
- Precision (class tăng) = 0.70: khi mô hình báo tín hiệu tăng thì khá tin cậy.
- Recall (class tăng) = 0.34: bỏ sót tới 2/3 số phiên tăng, dẫn đến hiệu quả thấp trong việc phát hiện cơ hội giao dịch.
- Ma trận nhầm lẫn cho thấy mô hình có xu hướng nghiêng về dự đoán giảm, an toàn nhưng làm giảm tính hữu ích trong trading.

Ý nghĩa: Mô hình phân loại hiện tại chỉ mang tính tham khảo, có thể dùng như một chỉ báo phụ để xác nhận xu hướng giảm, nhưng chưa đủ mạnh để đưa tín hiệu mua/bán độc lập.

Hạn chế: Việc bỏ sót nhiều tín hiệu tăng khiến mô hình không phù hợp trong môi trường giao dịch thực tế, nơi quyết định thường phụ thuộc vào việc phát hiện sớm các cơ hội tăng giá.

5.3. Tổng hợp và Định hướng cải thiện

Kết quả nghiên cứu cho thấy:

- Mô hình hồi quy đã thành công, có độ chính xác cao và tiềm năng ứng dụng thực tiễn trong phân tích kỹ thuật và quản lý rủi ro.
- Mô hình phân loại mới chỉ ở mức thử nghiệm, cần nhiều cải tiến để đạt độ tin cậy.

Định hướng cải thiện: Mở rộng đặc trưng: thêm các chỉ báo kỹ thuật khác (MACD, Bollinger Bands, Stochastic, Momentum), cũng như dữ liệu ngoài chuỗi (tin tức, tâm lý thị trường).

5.4. Tổng Kết

Đề tài đã chứng minh tính khả thi của việc áp dụng LSTM trong dự báo chuỗi thời gian tài chính, cụ thể là giá và xu hướng Bitcoin.

- Hồi quy: mang lại kết quả xuất sắc, có thể sử dụng ngay để hỗ trợ quyết định đầu tư ngắn hạn.
- Phân loại xu hướng: cần cải tiến thêm để trở thành công cụ đáng tin cậy trong trading.

Tổng thể, nghiên cứu đã mở ra hướng ứng dụng thực tiễn cho các mô hình học sâu trong phân tích thị trường tiền điện tử, đồng thời cung cấp nền tảng để phát triển các mô hình mạnh hơn, kết hợp nhiều nguồn dữ liệu hơn trong tương lai.

C. TÀI LIỆU THAM KHẢO

[1]. Binance Spot API Documentation, https://developers.binance.com/docs/binance-spot-api-docs

[2]. Kaggle Notebook: Bitcoin Price Prediction using LSTM, https://www.kaggle.com/code/meetnagadia/bitcoin-price-prediction-using-lstm