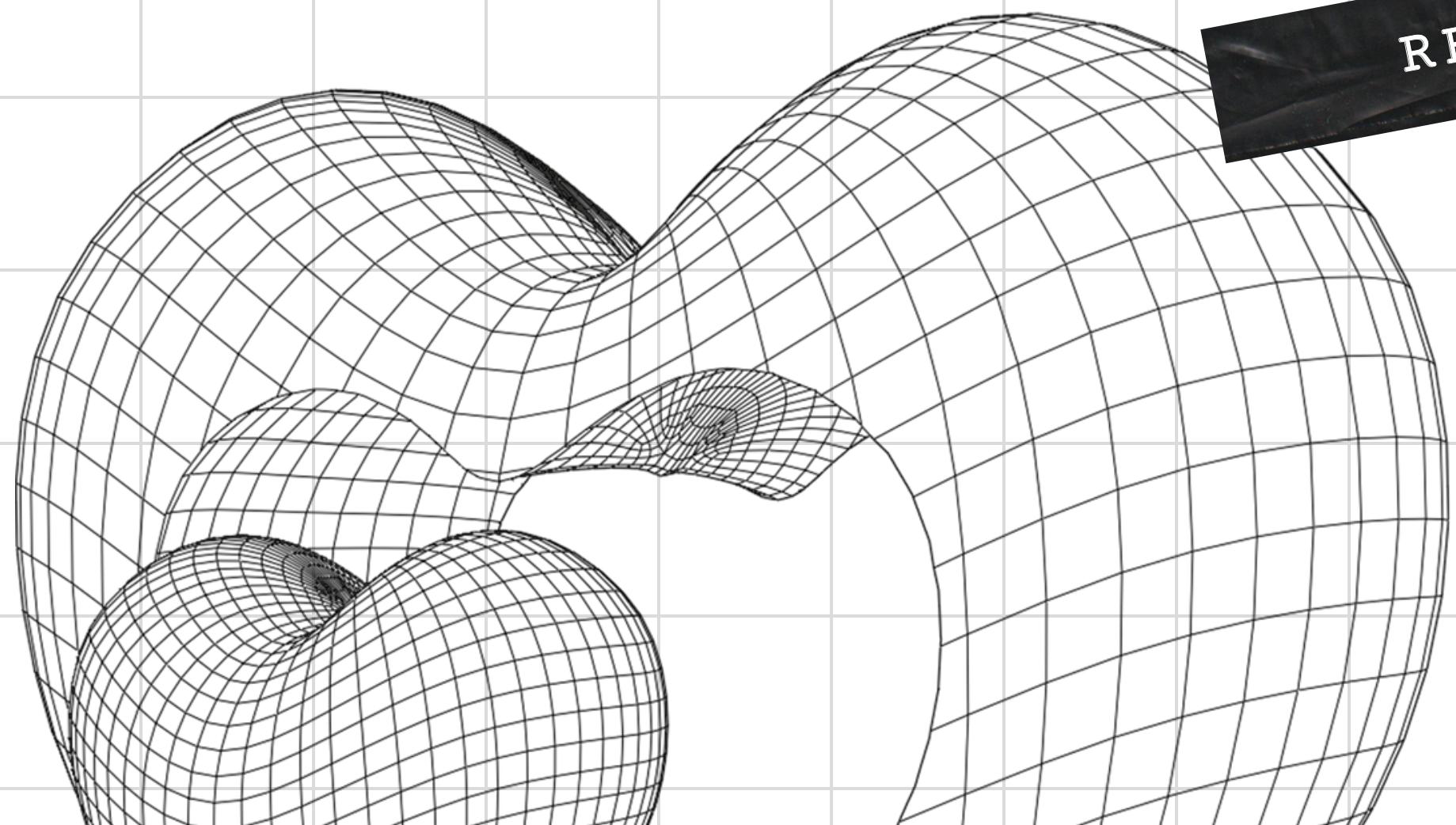


FINAL PROJECT



REPORT



NHẬT MINH

NGỌC ANH

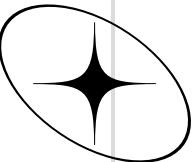
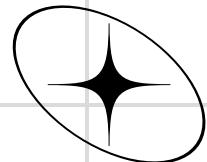
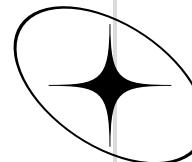


TABLE OF CONTENTS

1	INTRODUCTION	4	MODEL BUILDING
2	DATA PREPROCESSING	5	MODEL COMPARISON
3	EXPLORATORY DATA ANALYSIS - EDA	6	RESULT VISUALIZATION





INTRODUCTION

Big Tech Giants Stock Price Data

1. CONTEXT:

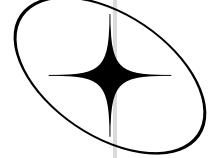
THIS DATASET CONSISTS OF THE DAILY STOCK PRICES AND VOLUME OF 14 DIFFERENT TECH COMPANIES:

- APPLE (AAPL)
- AMAZON (AMZN)
- NVIDIA (NVDA)
- META PLATFORMS (META)
- ADOBE (ADBE)
- INTEL CORPORATION (INTC)
- NETFLIX (NFLX)
- TESLA (TSLA) AND MORE!

2. TIME PERIOD:

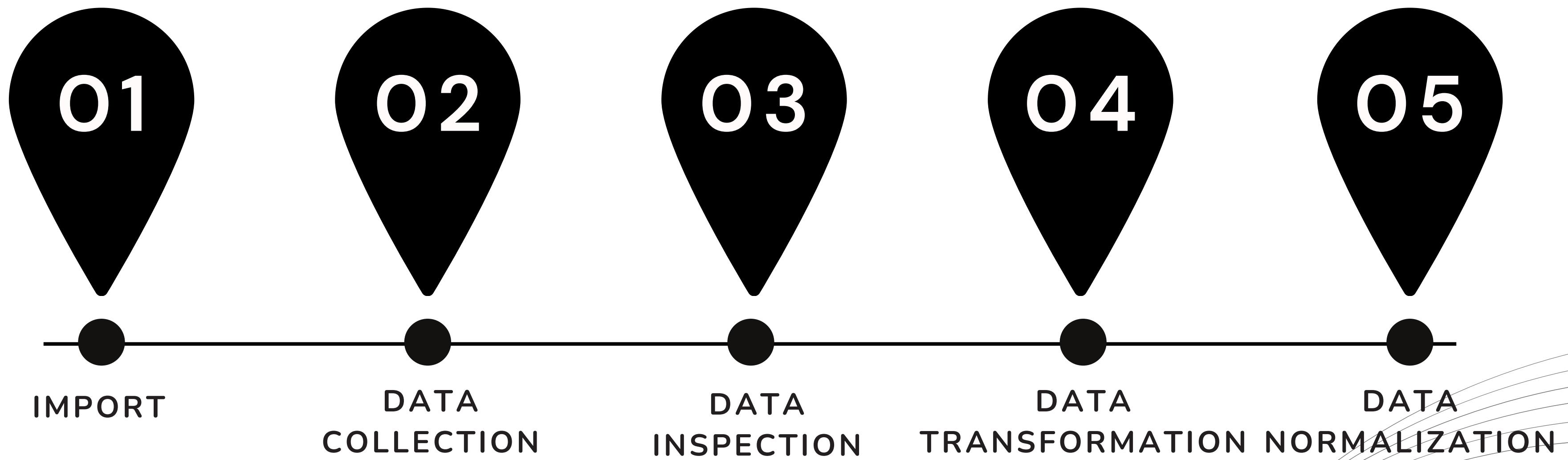
- JAN 2010 - JAN 2023

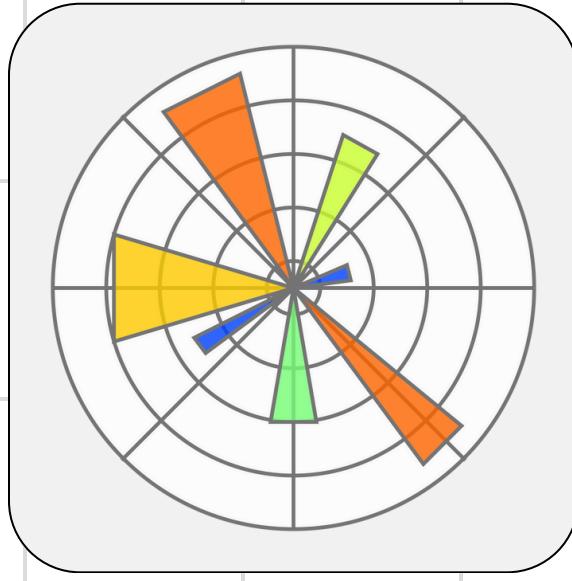
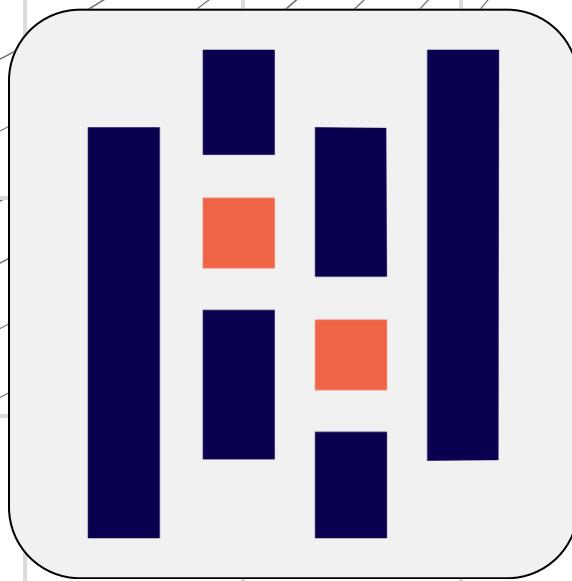
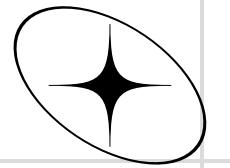
VARIABLES



VARIABLE	DESCRIPTION
stock_symbol	stock_symbol
date	date
open	The price at market open.
high	The highest price for that day.
low	The lowest price for that day.
close	The price at market close, adjusted for splits.
adj_close	The closing price after adjustments for all applicable splits and dividend distributions. Data is adjusted using appropriate split and dividend multipliers, adhering to Center for Research in Security Prices (CRSP) standards.
volume	The number of shares traded on that day.

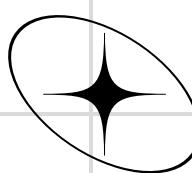
DATA PREPROCESSING

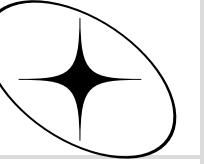




1 IMPORT

```
import pandas as pd  
import numpy as np  
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, r2_score  
from datetime import datetime
```





2

DATA COLLECTION

LOAD THE DATA COMPANIES

```
companies = 'https://drive.google.com/file/d/1WAM3YBCANlxK5ck67YDuAC7wVCo4f9L2/view?usp=drive_link'
```

READ DATA

```
companies_path = 'https://drive.google.com/uc?export=download&id=' + companies.split('/')[-2]
```

```
companies_df = pd.read_csv(companies_path,encoding= 'unicode_escape')
```

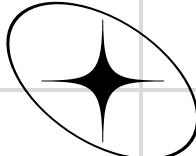
LOAD THE DATA STOCKPRICES

```
stockprices = 'https://drive.google.com/file/d/1MgPyKaob_ihkjxmoHs-NlYxhKVcronO2/view?usp=sharing'
```

READ DATA

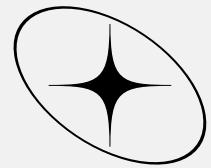
```
stockprices_path = 'https://drive.google.com/uc?export=download&id=' + stockprices.split('/')[-2]
```

```
stock_prices_df = pd.read_csv(stockprices_path,encoding= 'unicode_escape')
```



3

DATA INSPECTION



DISPLAY THE FIRST FEW ROWS

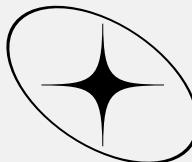
```
print("Big Tech Companies Dataset:")
print(companies_df.head())
print("\nBig Tech Stock Prices Dataset:")
print(stock_prices_df.head())
```



```
Big Tech Companies Dataset:
  stock_symbol          company
0      AAPL        Apple Inc.
1      ADBE      Adobe Inc.
2      AMZN Amazon.com, Inc.
3      CRM  Salesforce, Inc.
4      CSCO Cisco Systems, Inc.
```

```
Big Tech Stock Prices Dataset:
  stock_symbol       date     open     high      low    close  adj_close \
0      AAPL 2010-01-04  7.622500  7.660714  7.585000  7.643214   6.515213
1      AAPL 2010-01-05  7.664286  7.699643  7.616071  7.656429   6.526476
2      AAPL 2010-01-06  7.656429  7.686786  7.526786  7.534643   6.422664
3      AAPL 2010-01-07  7.562500  7.571429  7.466071  7.520714   6.410790
4      AAPL 2010-01-08  7.510714  7.571429  7.466429  7.570714   6.453412
```


	volume
0	493729600
1	601904800
2	552160000
3	477131200
4	447610800



3 DATA INSPECTION

**CHECK THE DATA TYPES
AND MISSING VALUES**

```
print("Big Tech Companies Dataset:")
print(companies_df.info())

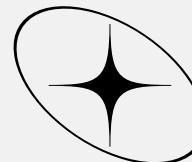
print("\nBig Tech Stock Prices Dataset:")
print(stock_prices_df.info())
```



```
Big Tech Companies Dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   stock_symbol    14 non-null   object  
 1   company        14 non-null   object  
dtypes: object(2)
memory usage: 352.0+ bytes
None
```

```
Big Tech Stock Prices Dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45088 entries, 0 to 45087
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   stock_symbol    45088 non-null  object  
 1   date          45088 non-null  object  
 2   open           45088 non-null  float64 
 3   high           45088 non-null  float64 
 4   low            45088 non-null  float64 
 5   close          45088 non-null  float64 
 6   adj_close       45088 non-null  float64 
 7   volume          45088 non-null  int64  
dtypes: float64(5), int64(1), object(2)
memory usage: 2.8+ MB
None
```

3 DATA INSPECTION

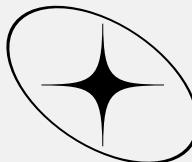


CHECK SUMMARY STATISTICS OF DATASETS

```
print("Big Tech Companies Dataset:")  
print(companies_df.describe())  
  
print("\nBig Tech Stock Prices Dataset:")  
print(stock_prices_df.describe())
```



```
Big Tech Companies Dataset:  
    stock_symbol      company  
count          14          14  
unique          14          14  
top           AAPL  Apple Inc.  
freq             1            1  
  
Big Tech Stock Prices Dataset:  
      open        high       low      close  adj_close \\  
count  45088.000000  45088.000000  45088.000000  45088.000000  45088.000000  
mean   89.266584    90.369825   88.111930   89.271306   85.209631  
std    101.626955   103.001073  100.124399  101.592916  100.995967  
min    1.076000    1.108667   0.998667   1.053333   1.053333  
25%   25.670000    25.930135   25.360001   25.660000   22.076433  
50%   47.930000    48.459999   47.465000   47.970001   45.377333  
75%   128.662502   129.848900  127.253945  128.640609  113.672460  
max   696.280029   700.989990  686.090027  691.690002  691.690002  
  
      volume  
count  4.508800e+04  
mean   5.297813e+07  
std    9.324730e+07  
min    5.892000e+05  
25%   9.629425e+06  
50%   2.646315e+07  
75%   5.839768e+07  
max   1.880998e+09
```

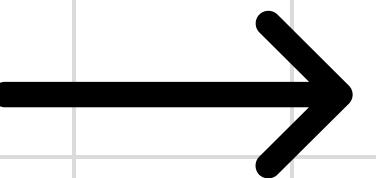


3

DATA INSPECTION

THE NUMBER OF UNIQUE COMPANIES AND SYMBOLS IN THE DATASETS

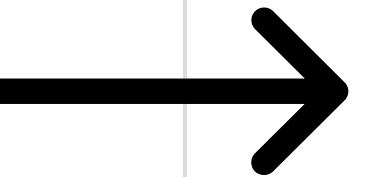
```
print("\nUnique Companies in Big Tech Companies  
Dataset:")  
print(companies_df['company'].nunique())  
  
print("\nUnique Stock Symbols in Big Tech Stock Prices  
Dataset:")  
print(stock_prices_df['stock_symbol'].nunique())
```



```
Unique Companies in Big Tech Companies Dataset:  
14  
  
Unique Stock Symbols in Big Tech Stock Prices Dataset:  
14
```

CHECK FOR MISSING VALUES

```
print("\nMissing Values in Big Tech Companies  
Dataset:")  
print(companies_df.isnull().sum())  
  
print("\nMissing Values in Big Tech Stock Prices  
Dataset:")  
print(stock_prices_df.isnull().sum())
```



```
Missing Values in Big Tech Companies Dataset:  
stock_symbol      0  
company          0  
dtype: int64  
  
Missing Values in Big Tech Stock Prices Dataset  
stock_symbol      0  
date              0  
open              0  
high              0  
low               0  
close             0  
adj_close         0  
volume            0  
dtype: int64
```

3

DATA INSPECTION

THE NUMBER OF STOCK SYMBOLS

```
print("\nStock Symbol Counts in Big  
Tech Stock Prices Dataset:")  
print(stock_prices_df['stock_symbol'].val  
ue_counts())
```



```
Stock Symbol Counts in Big Tech Stock Prices Dataset:  
stock_symbol  
AAPL      3271  
ADBE      3271  
AMZN      3271  
CRM       3271  
CSCO      3271  
GOOGL     3271  
IBM       3271  
INTC      3271  
MSFT      3271  
NFLX      3271  
NVDA      3271  
ORCL      3271  
TSLA      3148  
META      2688  
Name: count, dtype: int64
```

4

DATA TRANSFORMATION

CONVERT THE DATE COLUMN TO DATETIME FORMAT

```
stock_prices_df['date'] = pd.to_datetime(stock_prices_df['date'])
```

5

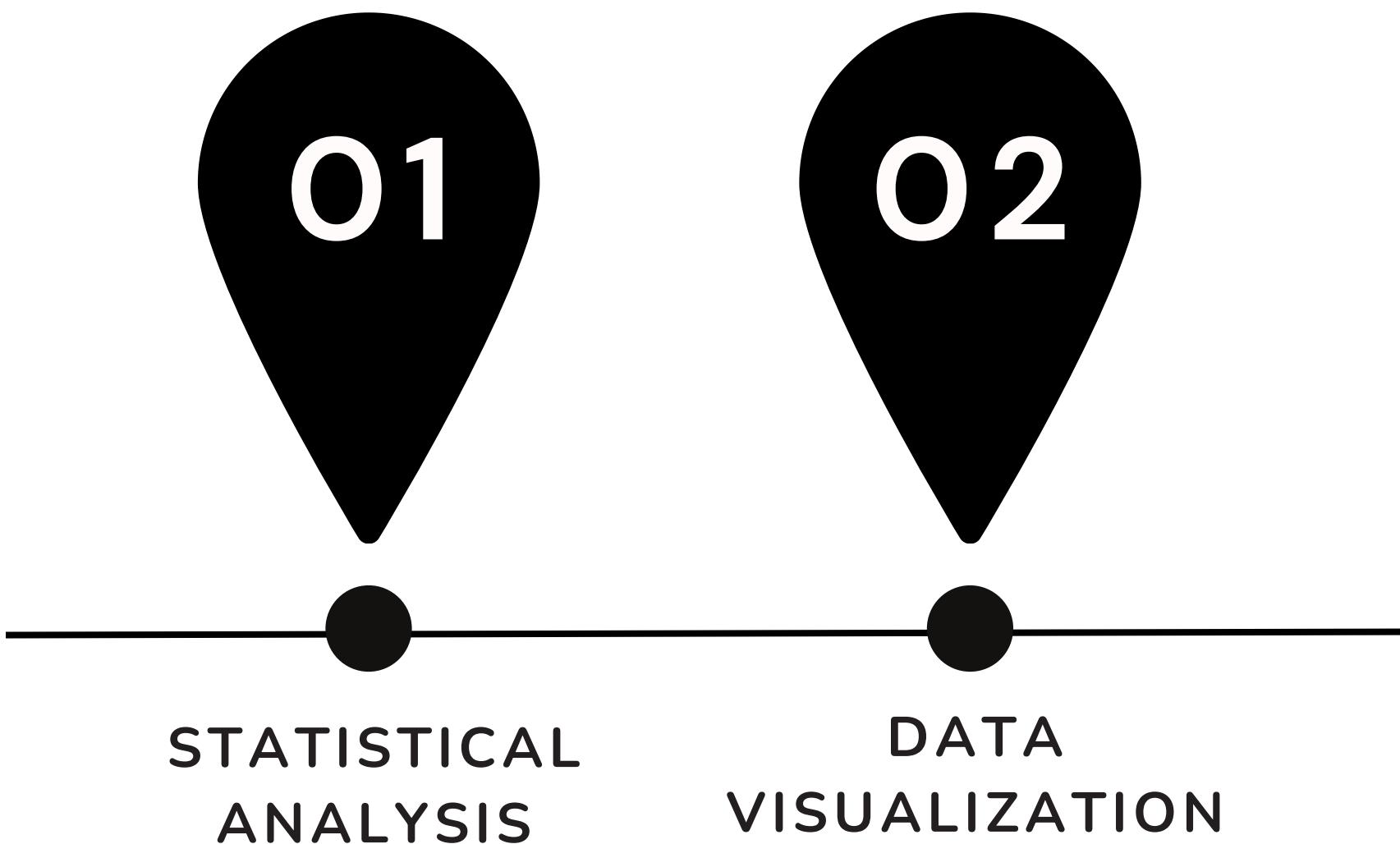
DATA NORMALIZATION

INITIALIZING MINMAXSCALER

```
from sklearn.preprocessing import MinMaxScaler  
scaler = MinMaxScaler()  
columns_to_normalize = ['open', 'high', 'low', 'close', 'adj_close', 'volume']  
stock_prices_df[columns_to_normalize] = scaler.fit_transform(stock_prices_df[columns_to_normalize])
```

EXPLORATORY DATA ANALYSIS

EDA



1

STATISTICAL ANALYSIS

RENAME

```
columns_of_statistical_analysis = ['open', 'high', 'low', 'close']  
print(columns_of_statistical_analysis)
```

MEAN

```
means = stock_prices_df[columns_of_statistical_analysis].mean()
```

MEDIAN

```
medians = stock_prices_df[columns_of_statistical_analysis].median()
```

STANDARD DEVIATION

```
standard_deviations = stock_prices_df[columns_of_statistical_analysis].std()
```



['open', 'high', 'low', 'close']

1

STATISTICAL ANALYSIS

PRINTING

```
print("Means:")
print(means)
print("\nMedians:")
print(medians)
print("\nStandard Deviations:")
print(standard_deviations)
```



Means:

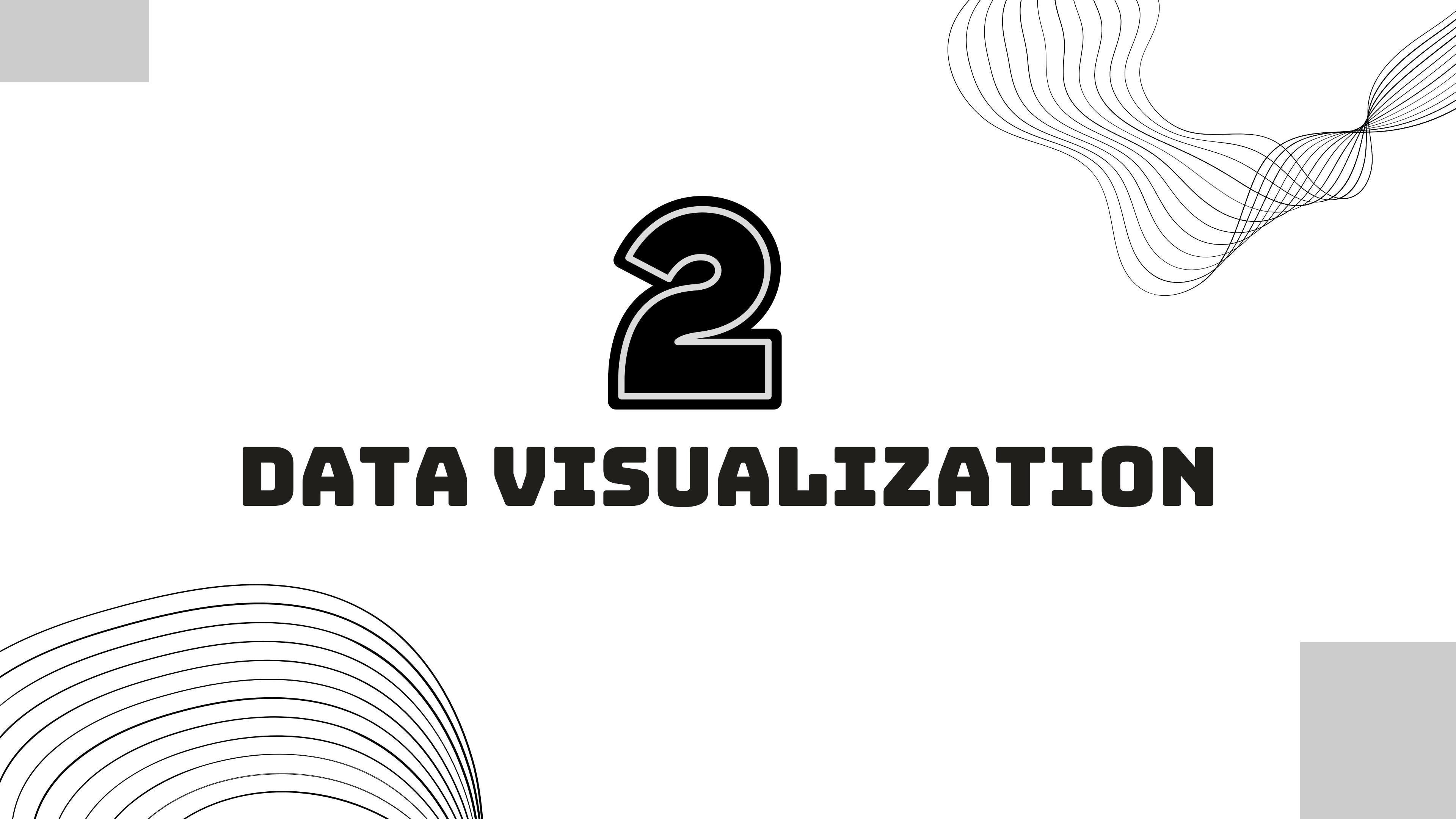
```
open      0.126856
high     0.127538
low      0.127156
close    0.127734
dtype: float64
```

Medians:

```
open      0.067396
high     0.067656
low      0.067825
close    0.067932
dtype: float64
```

Standard Deviations:

```
open      0.146183
high     0.147169
low      0.146148
close    0.147100
dtype: float64
```



2

DATA VISUALIZATION

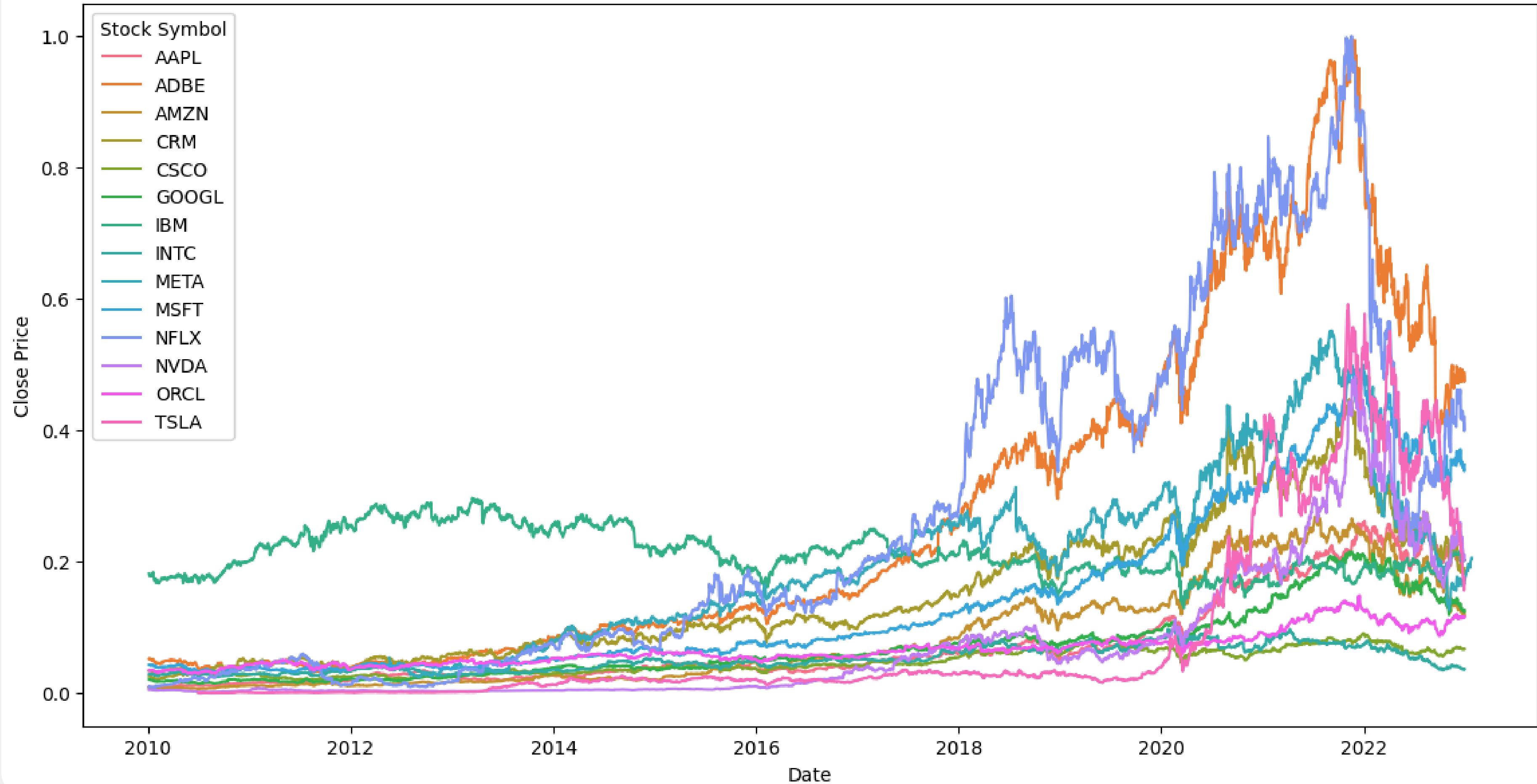
2

DATA VISUALIZATION

STOCK PRICES VISUALIZATION

```
plt.figure(figsize=(14, 7))
sns.lineplot(data=stock_prices_df, x='date', y='close', hue='stock_symbol')
plt.title('Stock Prices Over Time')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.legend(title='Stock Symbol')
plt.show()
```

Stock Prices Over Time

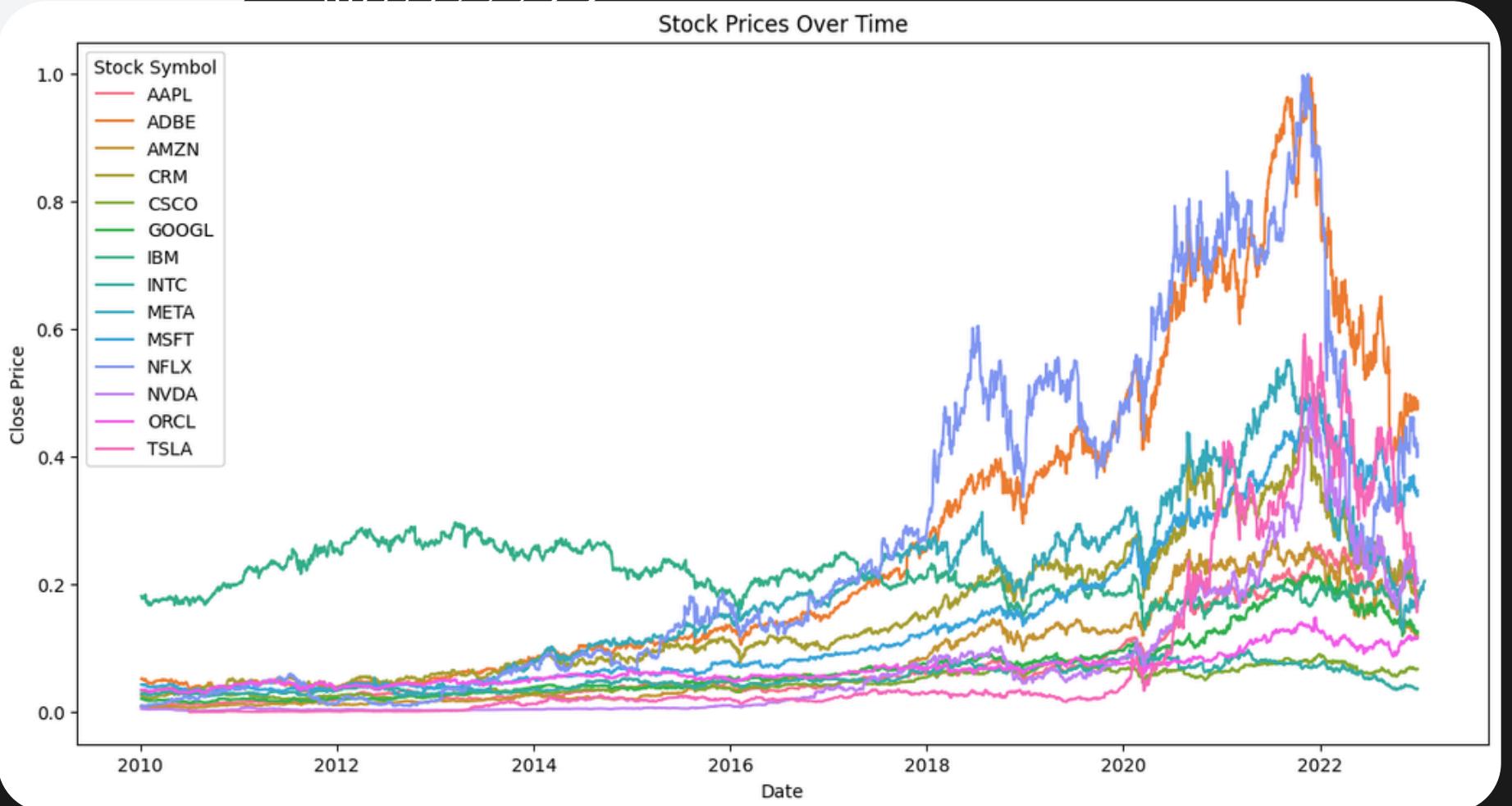


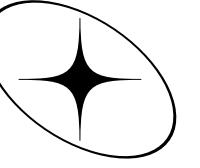
NHẬN XÉT

- CÓ XU HƯỚNG TĂNG TRƯỞNG TRONG GIAI ĐOẠN 2010-2022, VỚI NHIỀU CỔ PHIẾU ĐẠT ĐỈNH VÀO KHOẢNG NĂM 2022.
- CÁC CỔ PHIẾU NHƯ **AMAZON (AMZN)**, **ADOBE (ADBE)**, VÀ **TESLA (TSLA)** CÓ SỰ TĂNG TRƯỞNG ĐÁNG KỂ NHẤT.
- CÓ SỰ BIẾN ĐỘNG MẠNH VÀO KHOẢNG CÁC NĂM 2020-2022.

MỤC ĐÍCH

NHẬN DIỆN XU HƯỚNG DÀI HẠN CỦA TỪNG CỔ PHIẾU (XU HƯỚNG TĂNG, GIẢM, DAO ĐỘNG KHÔNG RÕ RÀNG).



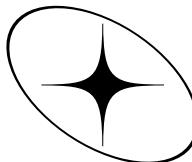


2

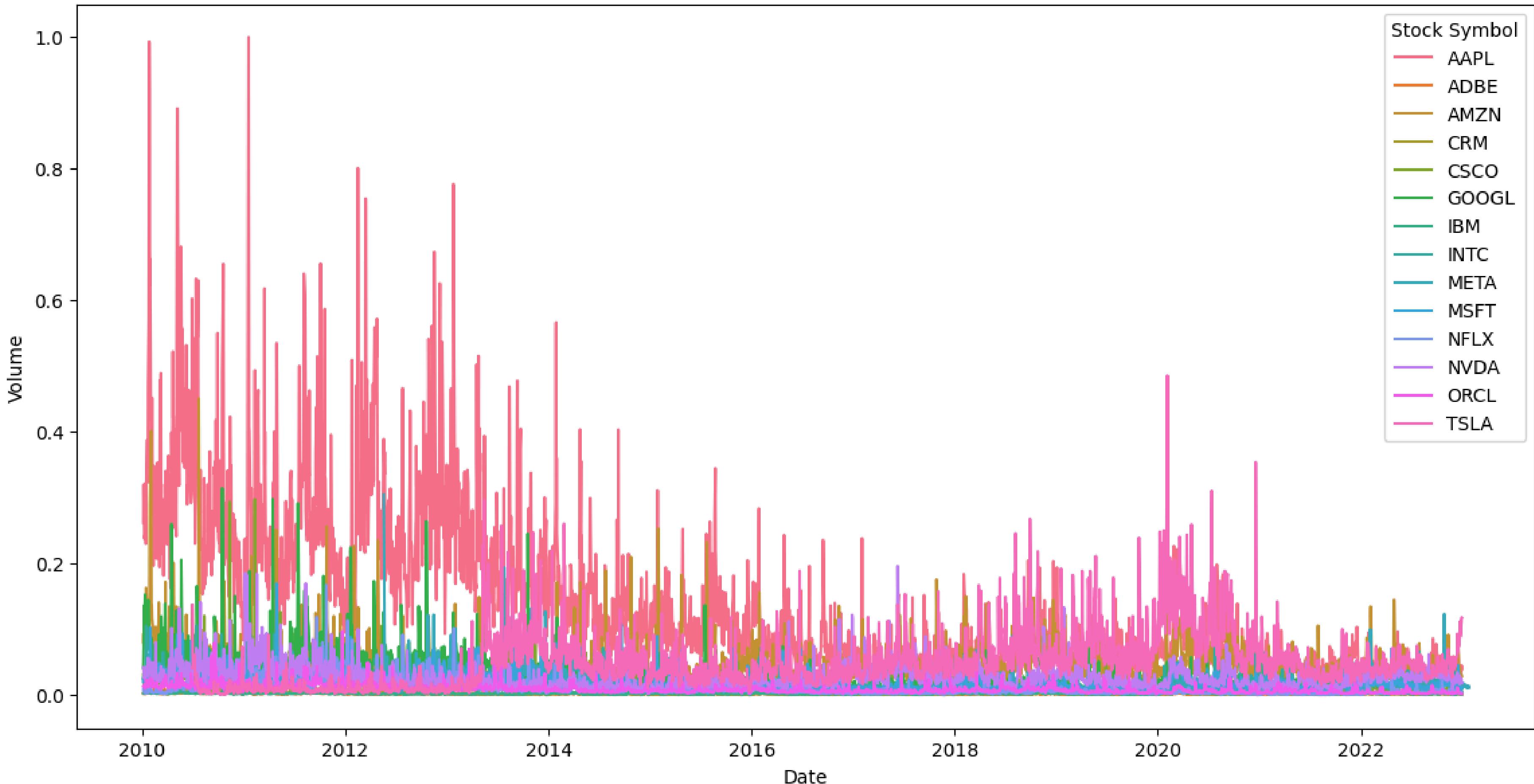
DATA VISUALIZATION

VOLUME DATA VISUALIZATION

```
plt.figure(figsize=(14, 7))
sns.lineplot(data=stock_prices_df, x='date', y='volume', hue='stock_symbol')
plt.title('Trading Volume Over Time')
plt.xlabel('Date')
plt.ylabel('Volume')
plt.legend(title='Stock Symbol')
plt.show()
```

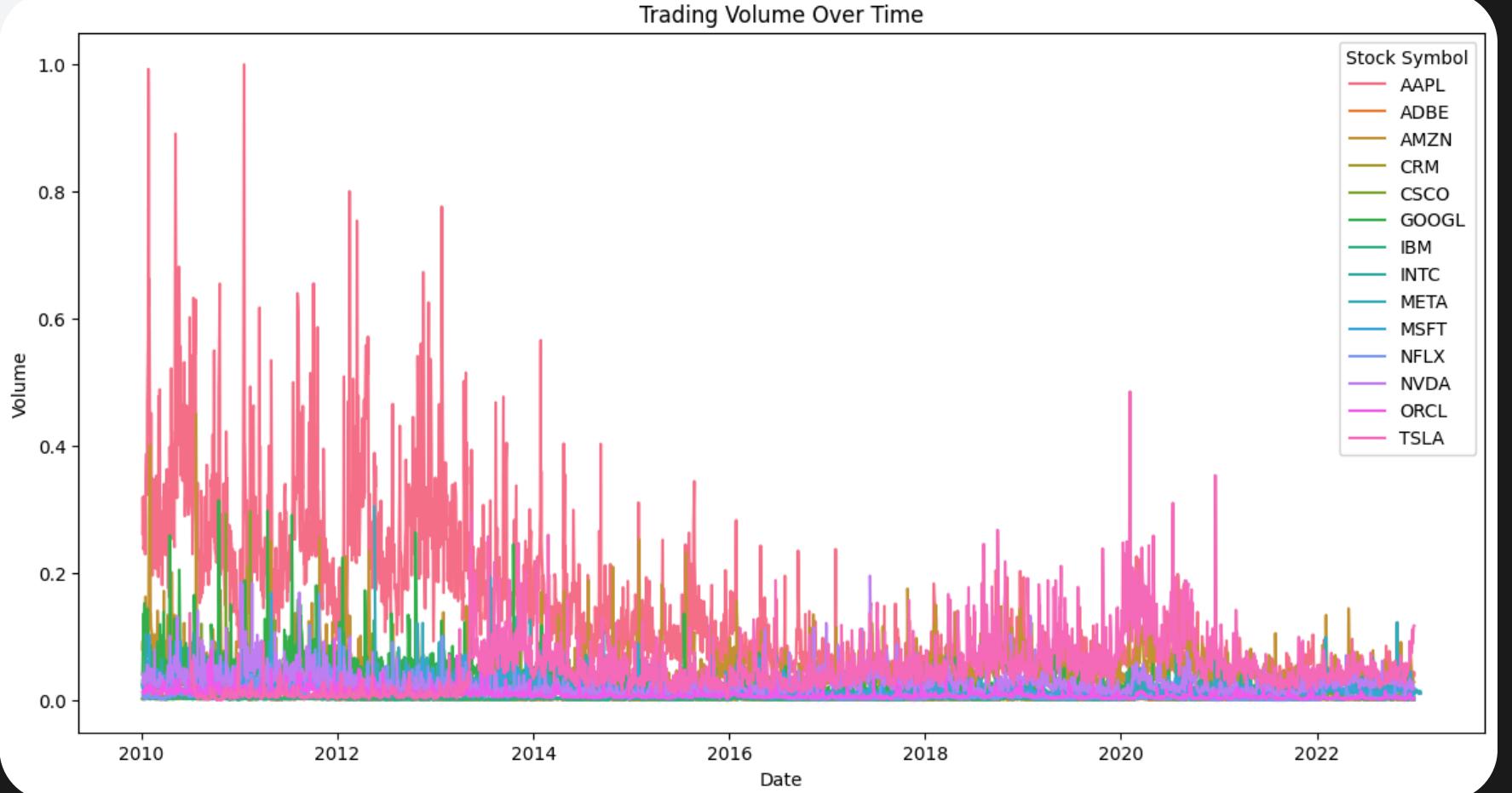


Trading Volume Over Time



MỤC ĐÍCH

- PHÁT HIỆN CÁC GIAI ĐOẠN CÓ KHỐI LƯỢNG GIAO DỊCH CAO HOẶC THẤP.
- XÁC ĐỊNH TÍNH THANH KHOẢN THÔNG QUA KHỐI LƯỢNG GIAO DỊCH.



NHẬN XÉT

- KHỐI LƯỢNG GIAO DỊCH CỦA **TESLA (TSLA)** NỔI BẬT VỚI NHIỀU ĐỈNH CAO.
- CÁC CÔNG TY KHÁC NHƯ **APPLE (AAPL)**, **AMAZON (AMZN)**, VÀ **GOOGLE (GOOGL)** CÓ KHỐI LƯỢNG GIAO DỊCH ỔN ĐỊNH.
- NHÌN CHUNG, KHỐI LƯỢNG GIAO DỊCH CÓ XU HƯỚNG GIẢM DẦN TỪ NĂM 2010 ĐẾN NĂM 2022.

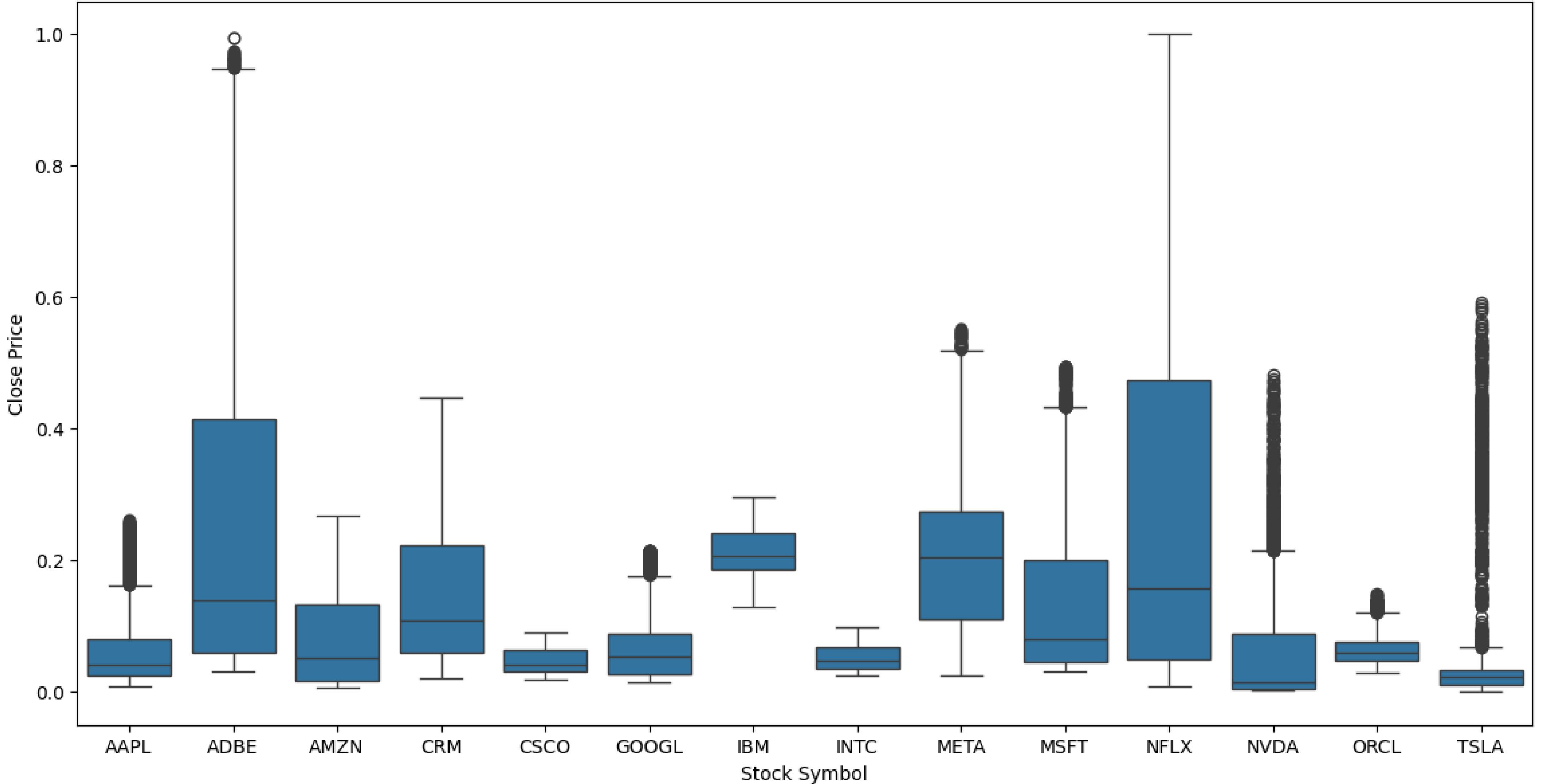
2

DATA VISUALIZATION

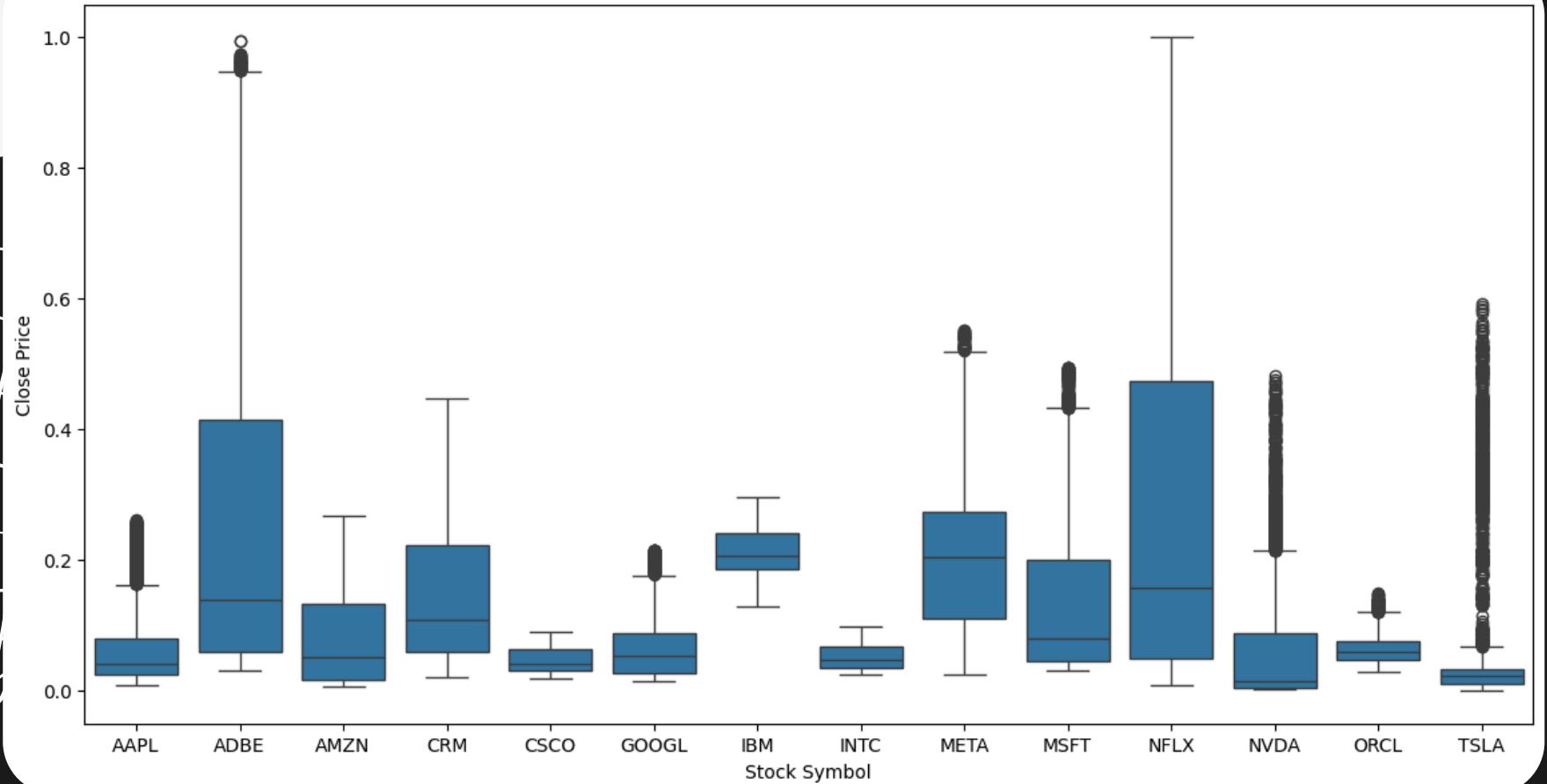
DISTRIBUTION OF CLOSING PRICES OF DIFFERENT COMPANIES

```
plt.figure(figsize=(14, 7))
sns.boxplot(data=stock_prices_df, x='stock_symbol', y='close')
plt.title('Distribution of Closing Prices by Stock Symbol')
plt.xlabel('Stock Symbol')
plt.ylabel('Close Price')
plt.show()
```

Distribution of Closing Prices by Stock Symbol



Distribution of Closing Prices by Stock Symbol



MỤC ĐÍCH

- PHÁT HIỆN CÁC CỔ PHIẾU CÓ GIÁ CAO HOẶC THẤP HƠN SO VỚI CÁC CỔ PHIẾU KHÁC.
- XÁC ĐỊNH CÁC CỔ PHIẾU CÓ GIÁ BIẾN ĐỘNG LỚN HOẶC NHỎ.

NHẬN XÉT

- **ADBE (ADOBE): GIÁ ĐÓNG CỦA CAO NHẤT VÀ BIẾN ĐỘNG LỚN.**
- **META (FACEBOOK): BIẾN ĐỘNG RỘNG VỚI NHIỀU ĐIỂM NGOẠI LỆ.**
- **NFLX (NETFLIX): BIẾN ĐỘNG GIÁ RỘNG.**
- **TSLA (TESLA): NHIỀU BIẾN ĐỘNG VỚI NHIỀU ĐIỂM NGOẠI LỆ.**
- **MSFT (MICROSOFT) VÀ NVDA (NVIDIA): CÓ GIÁ ĐÓNG CỦA TRUNG BÌNH CAO VÀ BIẾN ĐỘNG ĐÁNG CHÚ Ý.**

2 DATA VISUALIZATION

TIME SERIES ANALYSIS OF A SPECIFIC COMPANY (E.G. APPLE)

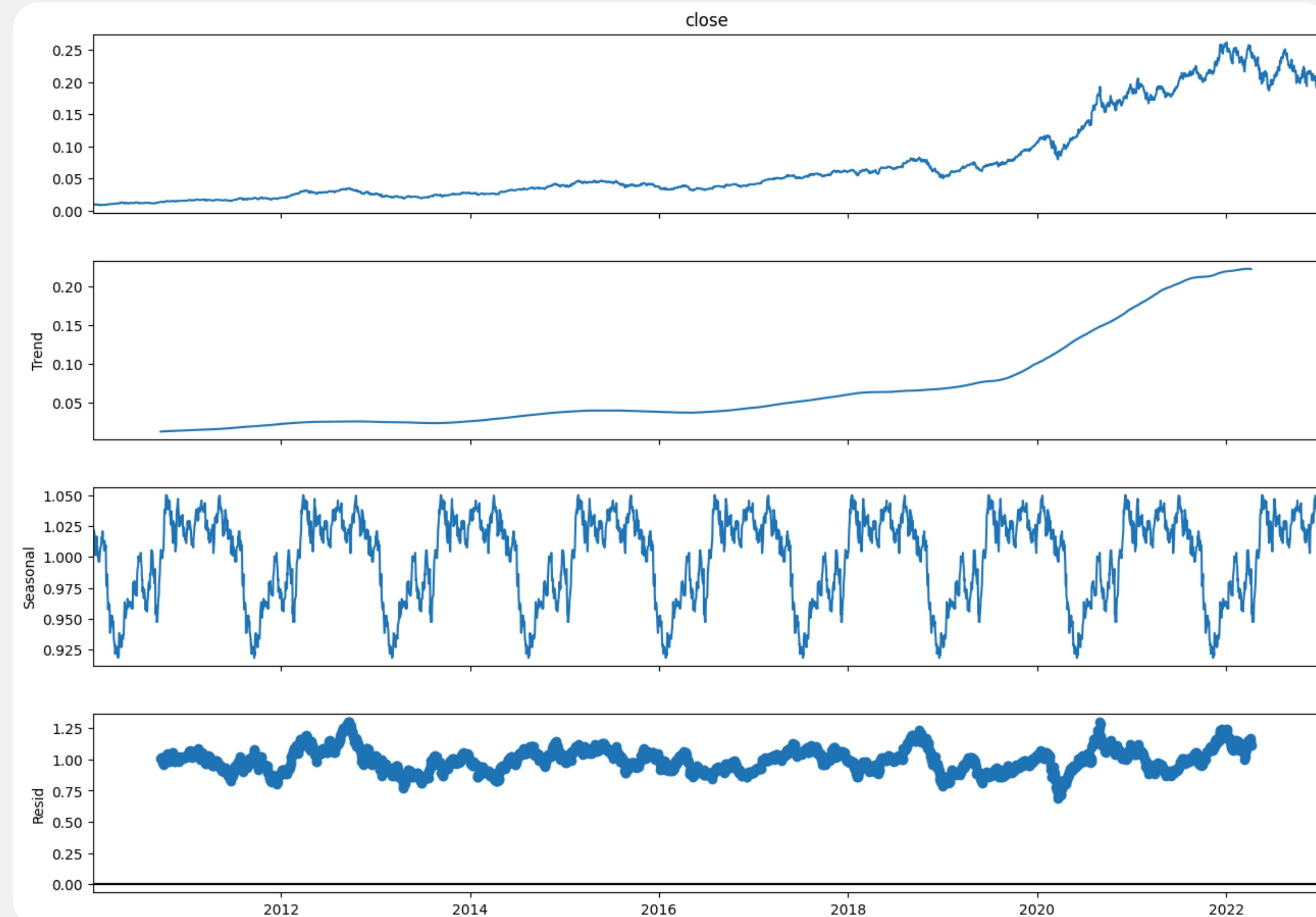
```
apple_stock = stock_prices_df[stock_prices_df['stock_symbol'] == 'AAPL']
apple_stock.set_index('date', inplace=True)
```

FOR TIME SERIES ANALYSIS

```
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.arima.model import ARIMA
import prophet
```

TIME SERIES DECOMPOSITION OF APPLE STOCK PRICES

```
decomposition = seasonal_decompose(apple_stock['close'], model='multiplicative',
period=365)
fig = decomposition.plot()
fig.set_size_inches(14, 10)
plt.show()
```



NHẬN XÉT

1. CLOSE (GIÁ ĐÓNG CỬA)

- Mô tả: Hiển thị dữ liệu gốc về giá đóng cửa của cổ phiếu Apple qua thời gian.
- Nhận xét: Giá cổ phiếu có xu hướng tăng dần từ năm 2010 đến khoảng năm 2022.

2. TREND (XU HƯỚNG)

- Mô tả: Hiển thị thành phần xu hướng của dữ liệu giá cổ phiếu.
- Nhận xét: Xu hướng giá cổ phiếu có sự tăng trưởng ổn định trong suốt giai đoạn từ 2010 đến 2022.

3. SEASONAL (THÀNH PHẦN MÙA VỤ)

- Mô tả: Hiển thị thành phần mùa vụ của dữ liệu giá cổ phiếu, cho thấy các dao động định kỳ hàng năm.
- Nhận xét: Có sự biến động rõ rệt, với các đỉnh và đáy lặp lại đều đặn theo thời gian.

4. RESID (THÀNH PHẦN NGẪU NHIÊN)

- Mô tả: Hiển thị phần dư của dữ liệu giá cổ phiếu sau khi đã loại bỏ xu hướng và thành phần mùa vụ.
- Nhận xét: Thành phần ngẫu nhiên cho thấy sự dao động xung quanh mức giá trung bình.

2

DATA VISUALIZATION

APPLE CLOSING PRICES AND 30-DAY MOVING AVERAGE

ADD THE MOVING AVERAGES OF APPLE STOCK PRICES

```
apple_stock['rolling_mean'] = apple_stock['close'].rolling(window=30).mean()
```

```
plt.figure(figsize=(14, 7))
```

```
apple_stock[['close', 'rolling_mean']].plot()
```

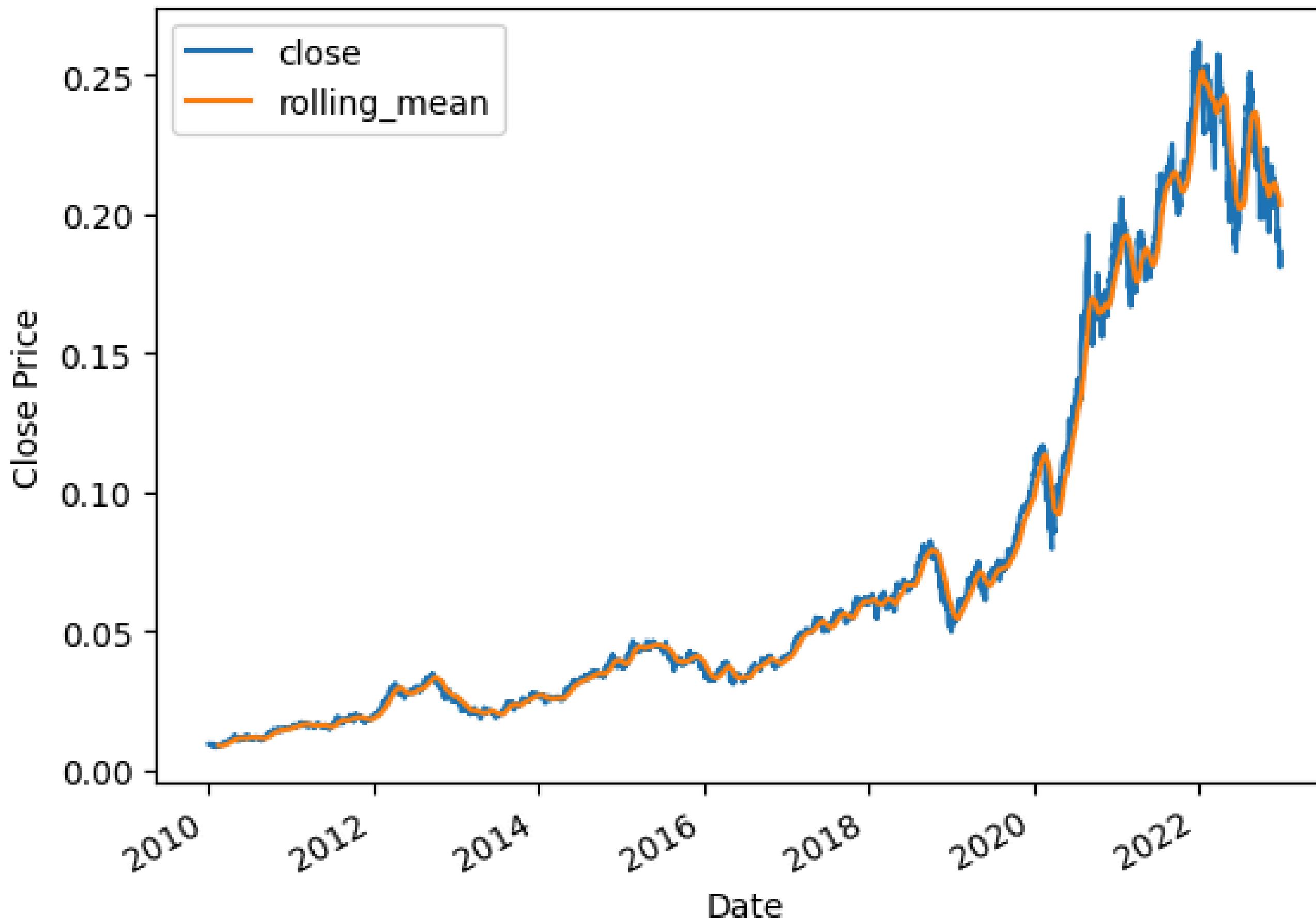
```
plt.title('Apple Closing Prices and 30-Day Moving Average')
```

```
plt.xlabel('Date')
```

```
plt.ylabel('Close Price')
```

```
plt.show()
```

Apple Closing Prices and 30-Day Moving Average



MỤC ĐÍCH

1. DỰ ĐOÁN GIÁ CỔ PHIẾU:

- Nhận diện các xu hướng ngắn hạn và dài hạn rõ ràng hơn.
- Khi giá cổ phiếu cắt lên trên đường trung bình động, đó có thể là tín hiệu mua; ngược lại, khi giá cổ phiếu cắt xuống dưới đường trung bình động, đó có thể là tín hiệu bán.

2. PHÂN TÍCH XU HƯỚNG THỊ TRƯỜNG:

- Nhận diện các xu hướng chính của giá cổ phiếu.
- Khi đường giá cổ phiếu nằm trên đường trung bình động và cả hai đều tăng, đó thường là tín hiệu của xu hướng tăng; ngược lại, khi giá cổ phiếu nằm dưới đường trung bình động và cả hai đều giảm, đó thường là tín hiệu của xu hướng giảm.

2

DATA VISUALIZATION

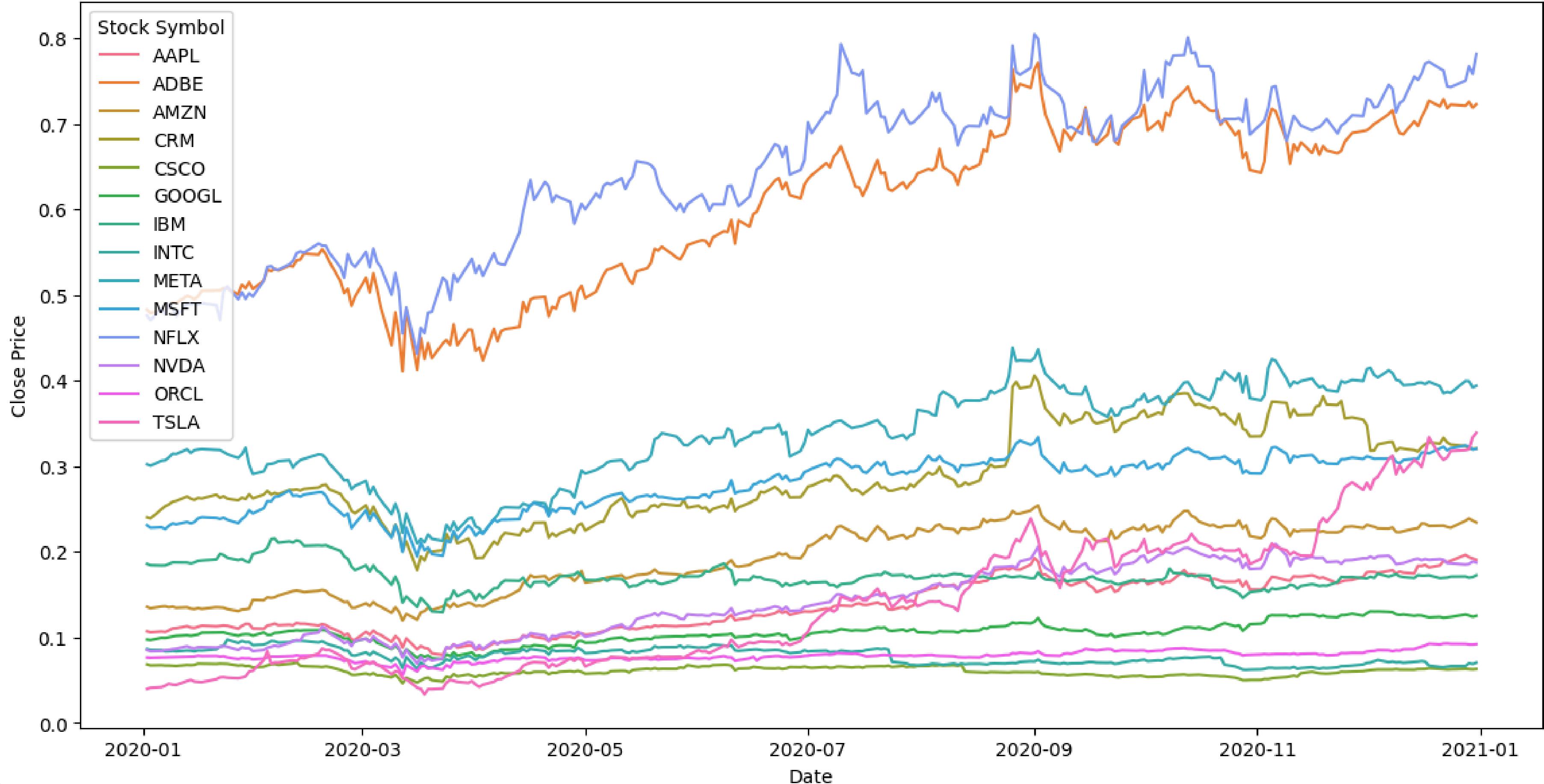
STOCK PRICES DURING 2020

EXAMINING PRICE CHANGES IN A SPECIFIC PERIOD (FOR EXAMPLE, 2020)

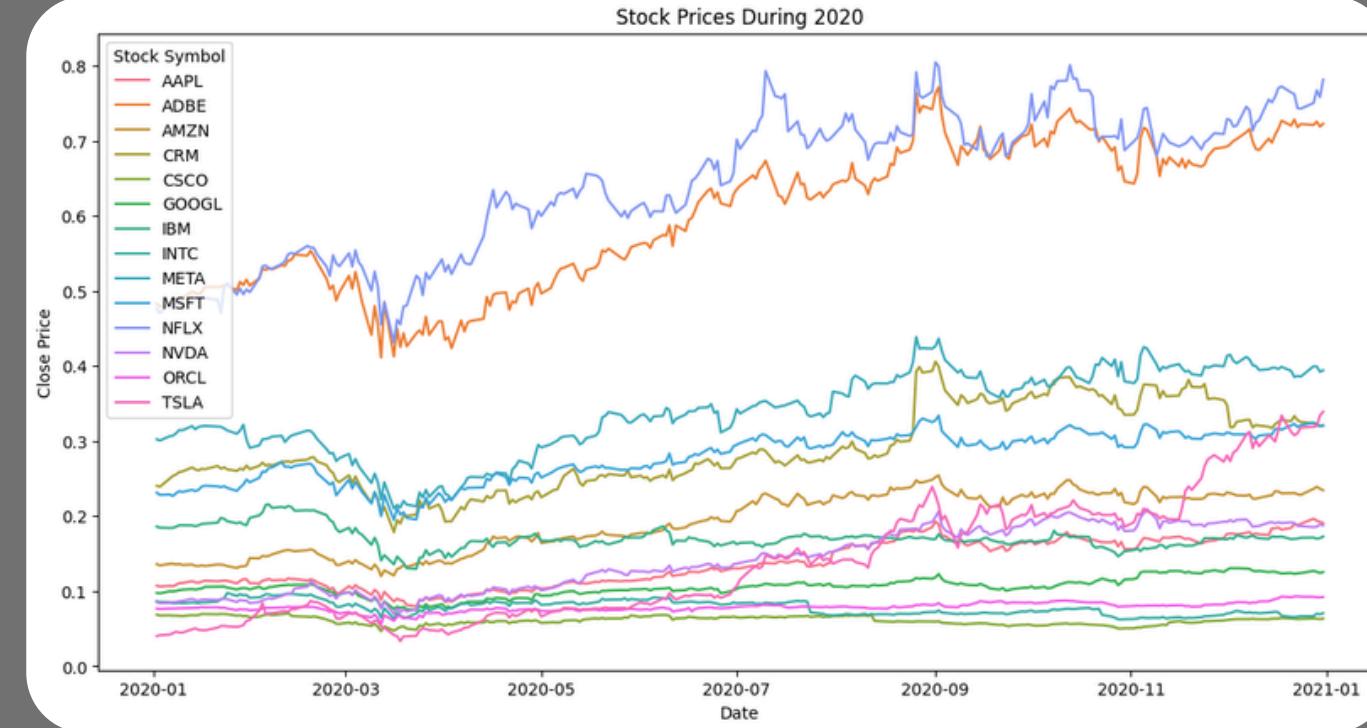
```
big_tech_stock_prices_2020 = stock_prices_df[(stock_prices_df['date'] >= '2020-01-01')  
& (stock_prices_df['date'] <= '2020-12-31')]
```

```
plt.figure(figsize=(14, 7))  
sns.lineplot(data=big_tech_stock_prices_2020, x='date', y='close', hue='stock_symbol')  
plt.title('Stock Prices During 2020')  
plt.xlabel('Date')  
plt.ylabel('Close Price')  
plt.legend(title='Stock Symbol')  
plt.show()
```

Stock Prices During 2020



NHẬN XÉT



- 1. DỰ ĐOÁN XU HƯỚNG TƯƠNG LAI CỦA CÁC CỔ PHIẾU NÀY.**
- 2. PHÂN TÍCH CÁCH MÀ CÁC CỔ PHIẾU CÔNG NGHỆ LỚN PHẢN ỨNG VỚI CÁC SỰ KIỆN THỊ TRƯỜNG TRONG NĂM 2020 (VÍ DỤ: ĐẠI DỊCH COVID-19)**

- Cổ phiếu tăng mạnh trong năm 2020:
 - TSLA (Tesla): Có xu hướng tăng mạnh, đặc biệt là trong nửa cuối năm 2020.
 - AAPL (Apple): Tăng trưởng ổn định trong suốt năm 2020.
- Cổ phiếu khác có hiệu suất đáng chú ý:
 - AMZN (Amazon): Có sự tăng trưởng mạnh mẽ trong năm 2020, phản ánh xu hướng mua sắm trực tuyến tăng cao trong đại dịch.
 - MSFT (Microsoft): Tăng trưởng ổn định trong năm 2020.
 - NFLX (Netflix): Tăng trưởng đáng kể do nhu cầu giải trí trực tuyến tăng cao trong thời gian giãn cách xã hội.

2

DATA VISUALIZATION

YEARLY PERCENTAGE CHANGE IN AVERAGE CLOSING PRICES

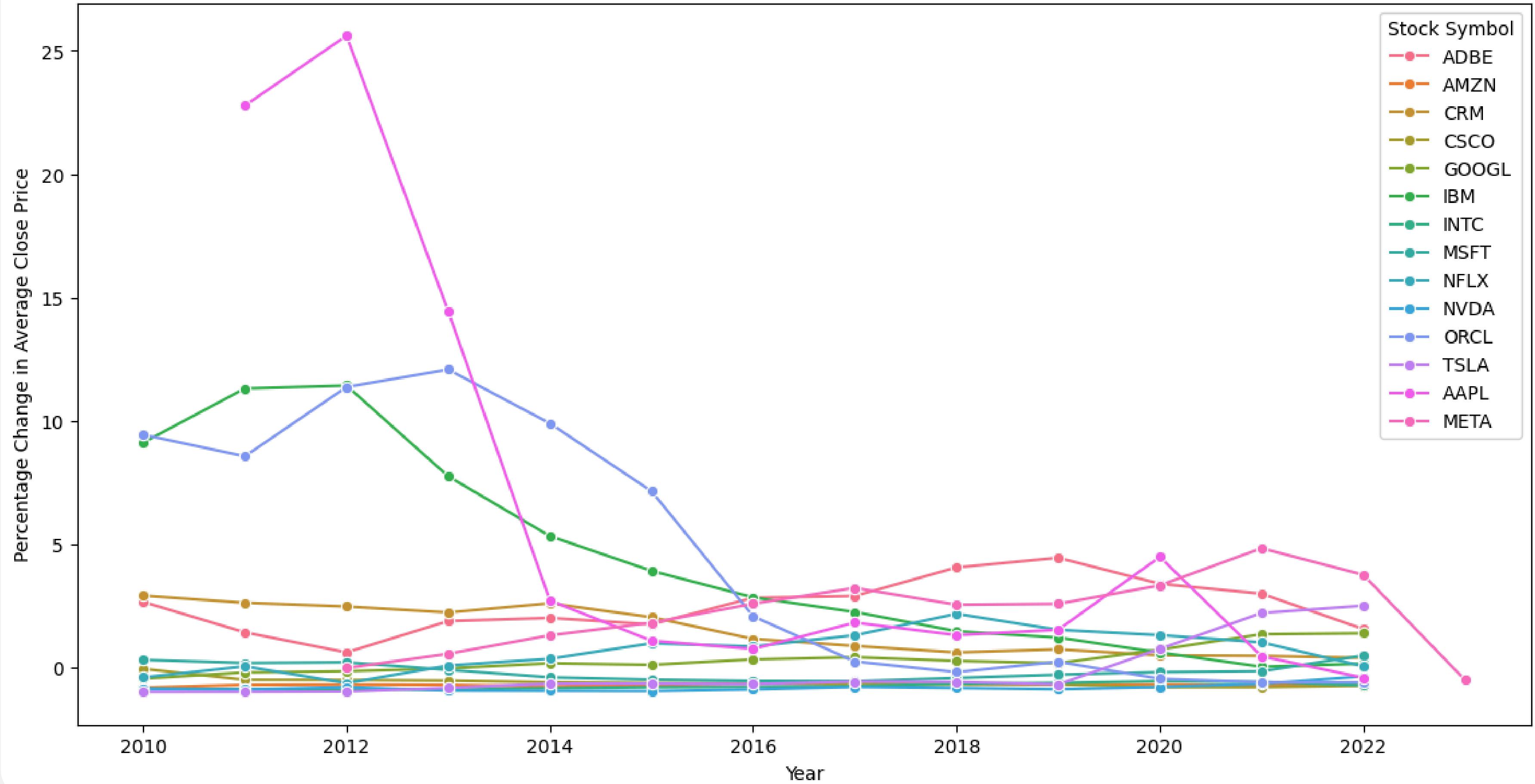
CALCULATE THE PERCENTAGE CHANGES OF ANNUAL CLOSING PRICES

```
yearly_price_change = stock_prices_df.groupby(['year', 'stock_symbol'])  
['close'].mean().pct_change().reset_index()  
yearly_price_change = yearly_price_change.dropna()
```

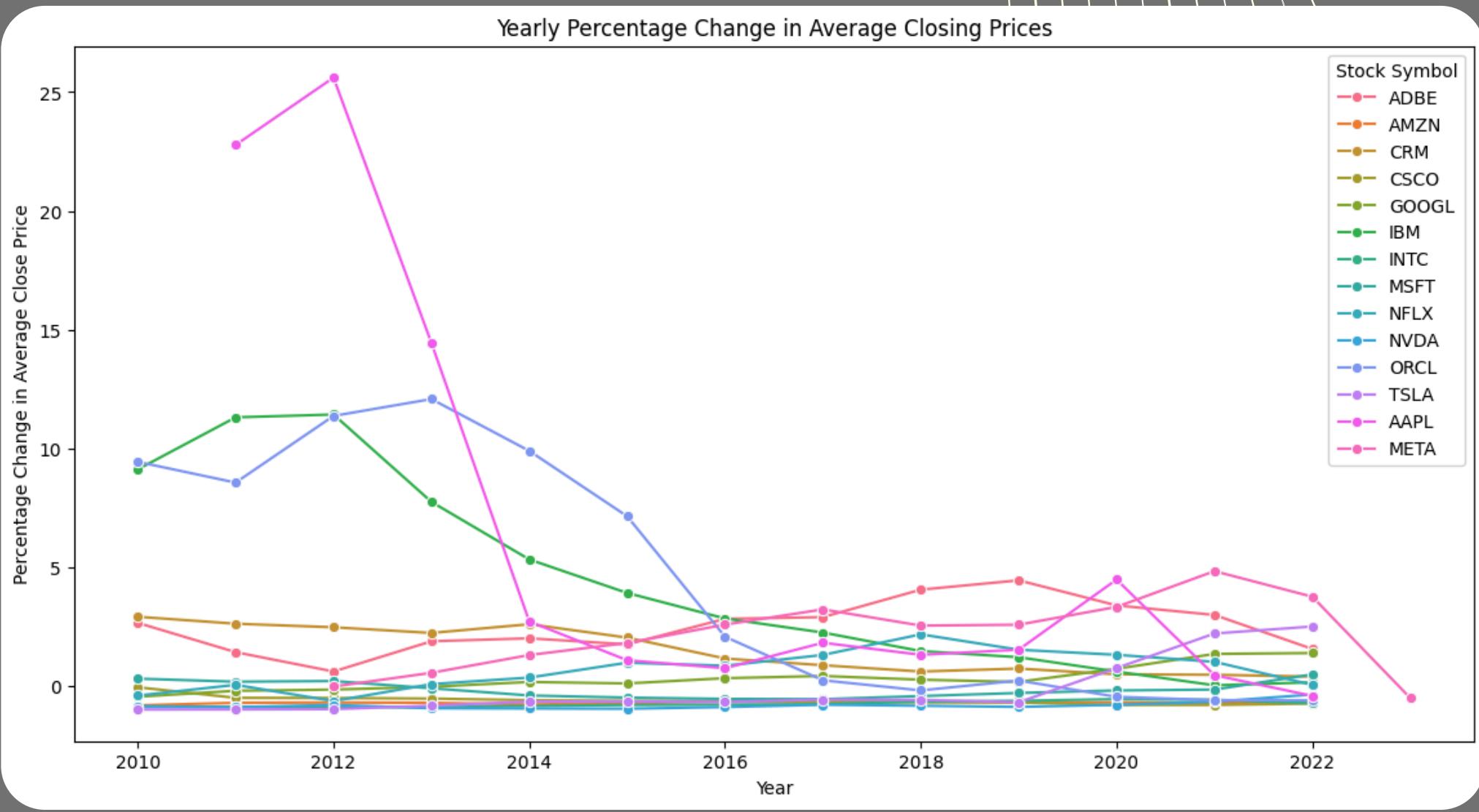
VISUALIZE ANNUAL PERFORMANCE

```
plt.figure(figsize=(14, 7))  
sns.lineplot(data=yearly_price_change, x='year', y='close', hue='stock_symbol', marker='o')  
plt.title('Yearly Percentage Change in Average Closing Prices')  
plt.xlabel('Year')  
plt.ylabel('Percentage Change in Average Close Price')  
plt.legend(title='Stock Symbol')  
plt.show()
```

Yearly Percentage Change in Average Closing Prices



NHẬN XÉT



- TSLA: Thay đổi phần trăm lớn và biến động mạnh mẽ, đặc biệt tăng đột biến vào các năm 2013-2014 và 2019-2021.
- NFLX: Biến động lớn vào các năm 2010-2015, sau đó giảm dần.
- NVDA: Thay đổi phần trăm lớn từ năm 2015 và đạt đỉnh vào khoảng năm 2018-2020.
- META (Facebook): Thay đổi phần trăm khá lớn vào các năm 2012-2013 và 2019-2020.

CONCLUSION

- Hiểu rõ xu hướng và biến động giá của các cổ phiếu công nghệ lớn.
- So sánh hiệu suất giữa các công ty.
- Phân tích tác động của các sự kiện thị trường lớn.
- Đưa ra quyết định đầu tư dựa trên dữ liệu cụ thể và rõ ràng.

MODEL BUILDING



LINEAR
REGRESSION
MODEL

RANDOM
FOREST
ALGORITHM

GRADIENT
BOOSTING
ALGORITHM

LINEAR REGRESSION MODEL

1 IMPORT

```
from sklearn.model_selection import train_test_split  
from sklearn.preprocessing import StandardScaler  
from sklearn.linear_model import LinearRegression  
from sklearn.metrics import mean_squared_error, r2_score,  
mean_absolute_error
```

2 CODE

PREPARE THE DATA FOR MODELING

```
X = stock_prices_df[['open', 'high', 'low', 'volume']]  
y = stock_prices_df['close']
```

SPLIT THE DATA INTO TRAINING AND TESTING SETS

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

INITIALIZE AND TRAIN THE MODEL

```
ln_model = LinearRegression()  
ln_model.fit(X_train, y_train)
```

MAKE PREDICTIONS

```
y_pred = ln_model.predict(X_test)
```

EVALUATE THE MODEL

```
mse = mean_squared_error(y_test, y_pred_LN)  
rmse = np.sqrt(mse)  
mae = mean_absolute_error(y_test, y_pred_LN)  
r2 = r2_score(y_test, y_pred_LN)
```

```
print(f"Mean Squared Error: {mse}")
```

```
print(f"Root Mean Squared Error: {rmse}")
```

```
print(f"Mean Absolute Error: {mae}")
```

```
print(f"R-squared: {r2}")
```



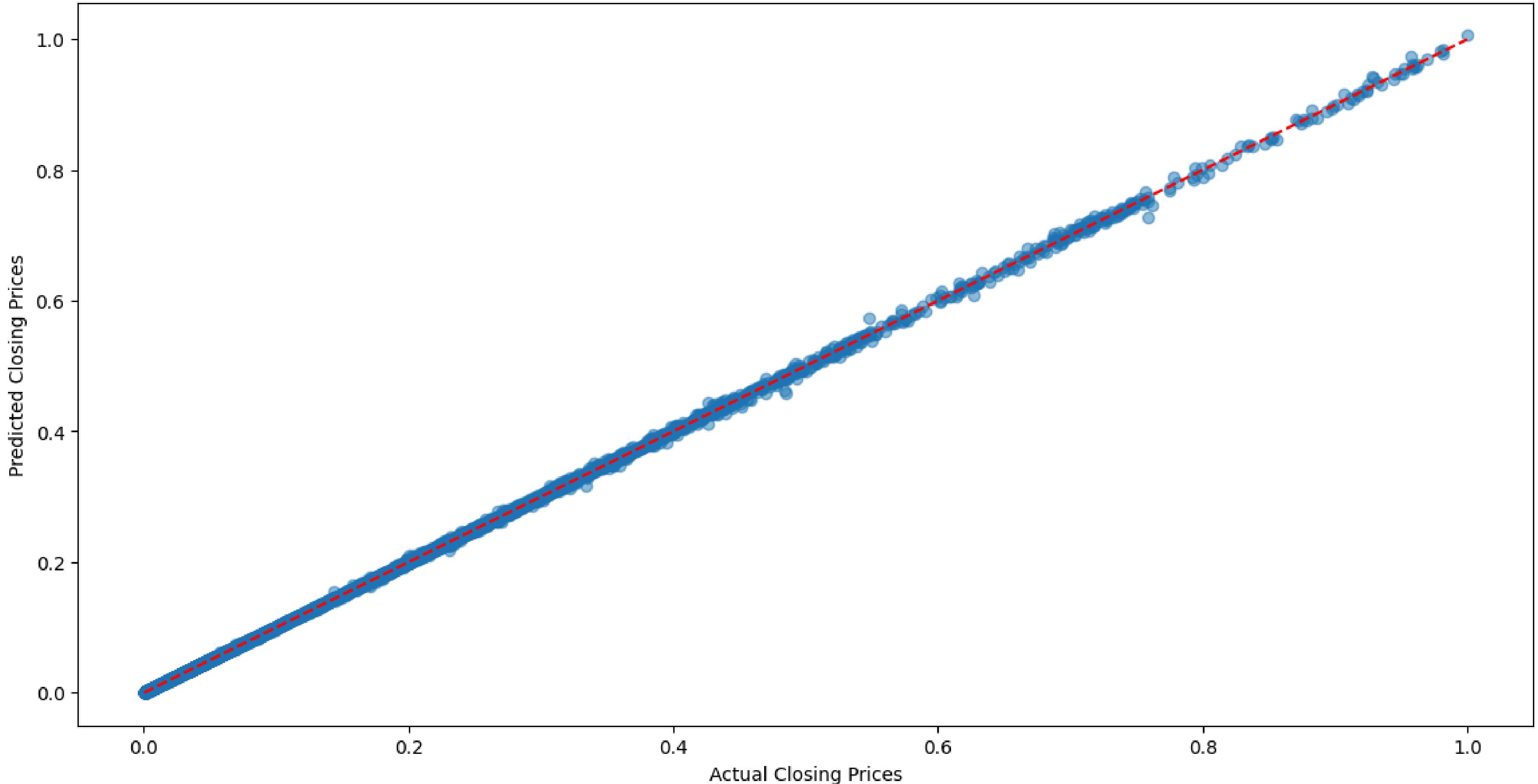
Mean Squared Error: 2.887451110720118e-06
Root Mean Squared Error: 0.0016992501613123708
Mean Absolute Error: 0.0007400411381147923
R-squared: 0.9998722419528762

3

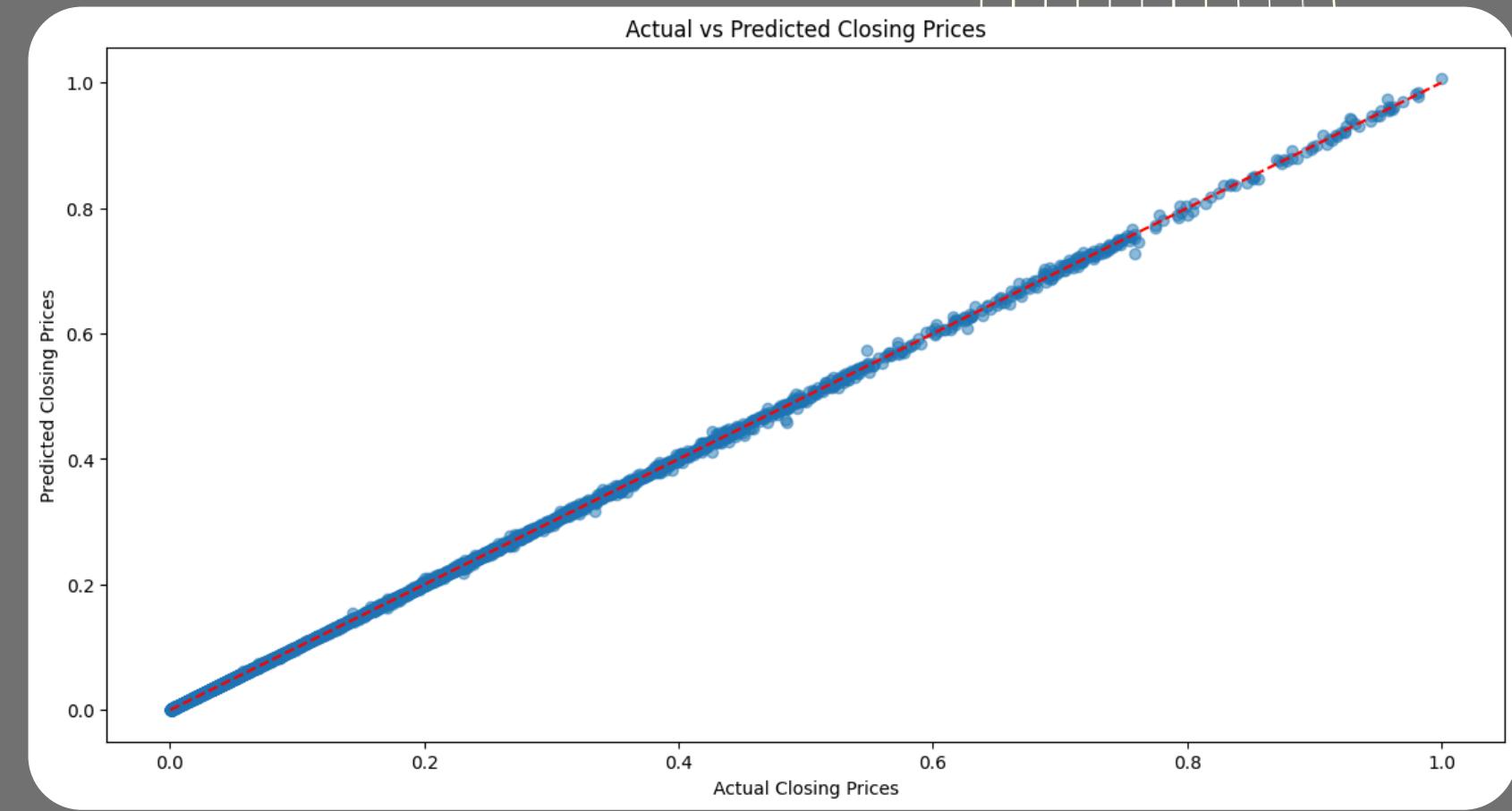
PLOT THE ACTUAL VS PREDICTED VALUES

```
plt.figure(figsize=(14, 7))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.title('Actual vs Predicted Closing Prices')
plt.xlabel('Actual Closing Prices')
plt.ylabel('Predicted Closing Prices')
plt.show()
```

Actual vs Predicted Closing Prices



NHẬN XÉT



1. Hiệu Suất Model:

- Mean Squared Error (MSE): Giá trị MSE rất nhỏ ($2.887451110720118e-06$) cho thấy mô hình dự đoán khá chính xác, với sai số bình phương trung bình rất thấp.
- Root Mean Squared Error (RMSE): Giá trị RMSE là 0.0016992501613123708 cũng rất nhỏ, cho thấy mô hình dự đoán rất chính xác và sai số trung bình là rất nhỏ.
- Mean Absolute Error (MAE): Giá trị MAE là 0.0007400411381147923 cũng rất nhỏ, cho thấy sai số tuyệt đối trung bình giữa dự đoán và thực tế là rất thấp, thể hiện sự chính xác của mô hình.
- R-squared (R^2): Giá trị R^2 rất cao (0.9998722419528762) gần bằng 1, cho thấy mô hình giải thích gần như toàn bộ sự biến thiên của dữ liệu. Đây là một chỉ số rất tốt, thể hiện mô hình có khả năng dự đoán chính xác và hiệu quả.

Tổng kết: Các chỉ số trên cho thấy mô hình hồi quy này có hiệu quả rất cao trong việc dự đoán dữ liệu, với sai số rất nhỏ và khả năng giải thích hầu như toàn bộ sự biến thiên của dữ liệu thực.

2. Diễn Giải Biểu Đồ:

- Biểu đồ phân tán của giá đóng cửa thực tế so với giá trị dự đoán cho thấy các điểm nằm rất gần đường gạch đỏ ($y = x$), cho thấy các dự đoán của mô hình khớp rất tốt với giá trị thực tế.

RANDOM FOREST ALGORITHM

1 IMPORT

```
from sklearn.ensemble import RandomForestRegressor  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.model_selection import train_test_split  
from sklearn.metrics import accuracy_score, classification_report,  
confusion_matrix  
from sklearn.datasets import make_classification
```

2 CODE

INITIALIZE AND TRAIN THE MODEL

```
RF_model = RandomForestClassifier(n_estimators=100, random_state=42)
```

TRAIN THE MODEL

```
RF_model.fit(X_train, y_train)
```

MAKE PREDICTIONS

```
y_pred_RF = RF_model.predict(X_test)
```

EVALUATE THE MODEL

```
print(f"Mean Squared Error: {mse}")
```

```
print(f"Root Mean Squared Error: {rmse}")
```

```
print(f"Mean Absolute Error: {mae}")
```

```
print(f"R-squared: {r2}")
```



Mean Squared Error: 0.055

Root Mean Squared Error: 0.2345207879911715

Mean Absolute Error: 0.055

R-squared: 0.77997799779978

NHẬN XÉT

Hiệu Suất Model:

- Mean Squared Error (MSE): Giá trị MSE là 0.055. Mặc dù giá trị này không phải là quá cao, nó vẫn cho thấy rằng mô hình có một mức độ sai số nhất định trong dự đoán.
- Root Mean Squared Error (RMSE): Giá trị RMSE là 0.2345207879911715, thể hiện sai số trung bình. Giá trị RMSE cho thấy sai số của mô hình không quá lớn nhưng vẫn cần cải thiện.
- Mean Absolute Error (MAE): Giá trị MAE là 0.055, cho thấy sai số tuyệt đối trung bình giữa dự đoán và thực tế là 0.055. Điều này cho thấy mô hình có độ chính xác tương đối, nhưng vẫn có một khoảng cách giữa giá trị dự đoán và giá trị thực.
- Giá trị R^2 là 0.77997799779978, nghĩa là mô hình giải thích được khoảng 78% sự biến thiên của dữ liệu thực, cho thấy còn khoảng 22% sự biến thiên chưa được mô hình giải thích, nghĩa là mô hình vẫn có thể được cải thiện để dự đoán chính xác hơn.

Tổng kết: Các chỉ số trên cho thấy mô hình này có khả năng dự đoán tương đối tốt.

GRADIENT BOOSTING ALGORITHM

1 IMPORT

```
from sklearn.model_selection import train_test_split,  
cross_val_score  
from sklearn.ensemble import GradientBoostingRegressor
```

2 CODE

PREPARE THE DATA FOR MODELING

```
X = stock_prices_df[['open', 'high', 'low', 'volume']]  
y = stock_prices_df['close']
```

SPLIT THE DATA INTO TRAINING AND TESTING SETS

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

INITIALIZE AND TRAIN THE MODEL

```
model = GradientBoostingRegressor(random_state=42)  
gb_model = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42)  
gb_model.fit(X_train, y_train)
```

MAKE PREDICTIONS

```
y_pred = gb_model.predict(X_test)
```

CALCULATE EVALUATION INDEXES

```
mse = mean_squared_error(y_test, y_pred)  
rmse = np.sqrt(mse)  
mae = mean_absolute_error(y_test, y_pred)  
r2 = r2_score(y_test, y_pred)
```

```
print(f"Mean Squared Error: {mse}")
```

```
print(f"Root Mean Squared Error: {rmse}")
```

```
print(f"Mean Absolute Error: {mae}")
```

```
print(f"R-squared: {r2}")
```



Mean Squared Error: 7.31411544253504e-06
Root Mean Squared Error: 0.0027044621355336146
Mean Absolute Error: 0.001422611864772455
R-squared: 0.9996763799387259

NHẬN XÉT

Hiệu Suất Model:

- Mean Squared Error (MSE): Giá trị MSE là $7.31411544253504\text{e-}06$ rất nhỏ. Điều này chỉ ra rằng mô hình dự đoán khá chính xác.
- Root Mean Squared Error (RMSE): Giá trị RMSE là 0.0027044621355336146 cũng rất nhỏ. RMSE cho thấy mô hình có sai số nhỏ và độ chính xác cao.
- Mean Absolute Error (MAE): Giá trị MAE là 0.001422611864772455 cũng rất nhỏ. Điều này phản ánh rằng mô hình có độ chính xác tốt trong việc dự đoán.
- R-squared (R^2): Giá trị R^2 là 0.9996763799387259 rất cao, gần bằng 1. Đây là một chỉ số rất tốt, thể hiện rằng mô hình có khả năng dự đoán chính xác và hiệu quả.

Tổng kết: Các chỉ số trên cho thấy mô hình hồi quy này có hiệu suất rất cao. Điều này chỉ ra rằng mô hình dự đoán rất chính xác, với sai số rất thấp và khả năng giải thích gần như toàn bộ sự biến thiên của dữ liệu thực.

MODEL COMPARISON

LINEAR REGRESSION

- Mean Squared Error (MSE): 2.887451110720118e-06
- Root Mean Squared Error (RMSE): 0.0016992501613123708
- Mean Absolute Error (MAE): 0.0007400411381147923
- R-squared: 0.9998722419528762

RANDOM FOREST

- Mean Squared Error (MSE): 0.055
- Root Mean Squared Error (RMSE): 0.2345207879911715
- Mean Absolute Error (MAE): 0.055
- R-squared: 0.77997799779978

GRADIENT BOOSTING

- Mean Squared Error (MSE): 7.31411544253504e-06
- Root Mean Squared Error (RMSE): 0.0027044621355336146
- Mean Absolute Error (MAE): 0.001422611864772455
- R-squared: 0.9996763799387259



Kết luận:

- Dựa trên các chỉ số MSE, RMSE, MAE và R-squared, mô hình Linear Regression và Gradient Boosting đều thể hiện hiệu suất rất tốt với sai số rất nhỏ và R^2 rất cao, gần bằng 1.
- Tuy nhiên, **Linear Regression có các chỉ số nhỏ hơn** một chút so với Gradient Boosting, đặc biệt là MSE, RMSE và MAE, cho thấy nó có thể là **mô hình tốt nhất** trong việc dự đoán dữ liệu này.

RESULT VISUALIZATION

1 CODING

LỌC DỮ LIỆU QUÝ 1 VÀ QUÝ 2 NĂM 2021 CỦA APPL

```
data_q1_q2_2021 = stock_prices_df[(stock_prices_df['date'] >= '2021-01-01') & (stock_prices_df['date'] <= '2021-06-30') &  
(stock_prices_df['stock_symbol']=='AAPL')]
```

CHUẨN BỊ DỮ LIỆU KIỂM TRA

```
X_test = data_q1_q2_2021[['open', 'high', 'low', 'volume']]  
y_test = data_q1_q2_2021['close']
```

KHỞI TẠO VÀ HUẤN LUYỆN MÔ HÌNH

```
LN_model = LinearRegression()  
LN_model.fit(X_train, y_train)
```

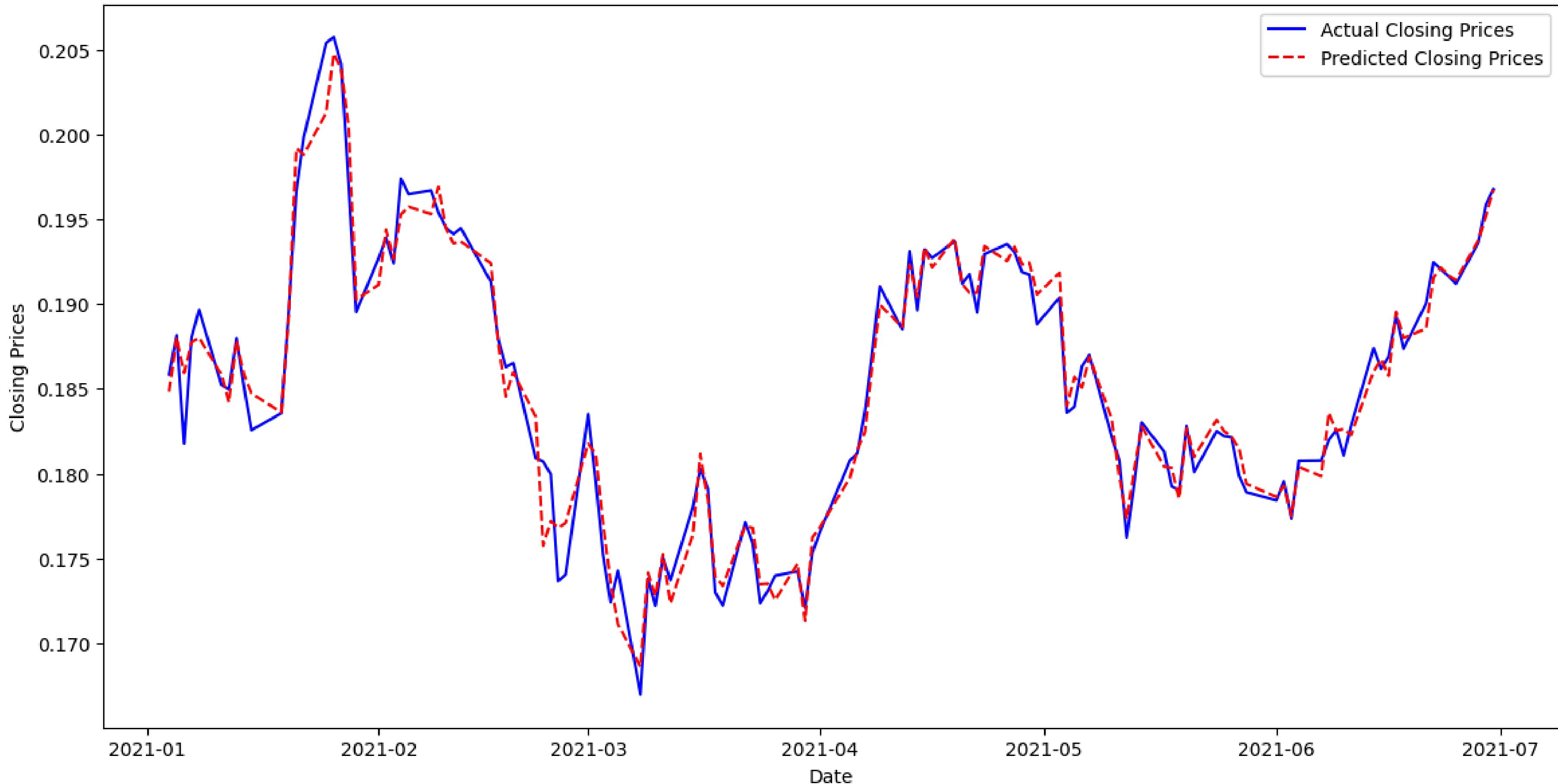
DỰ ĐOÁN

```
y_pred_LN = LN_model.predict(X_test)
```

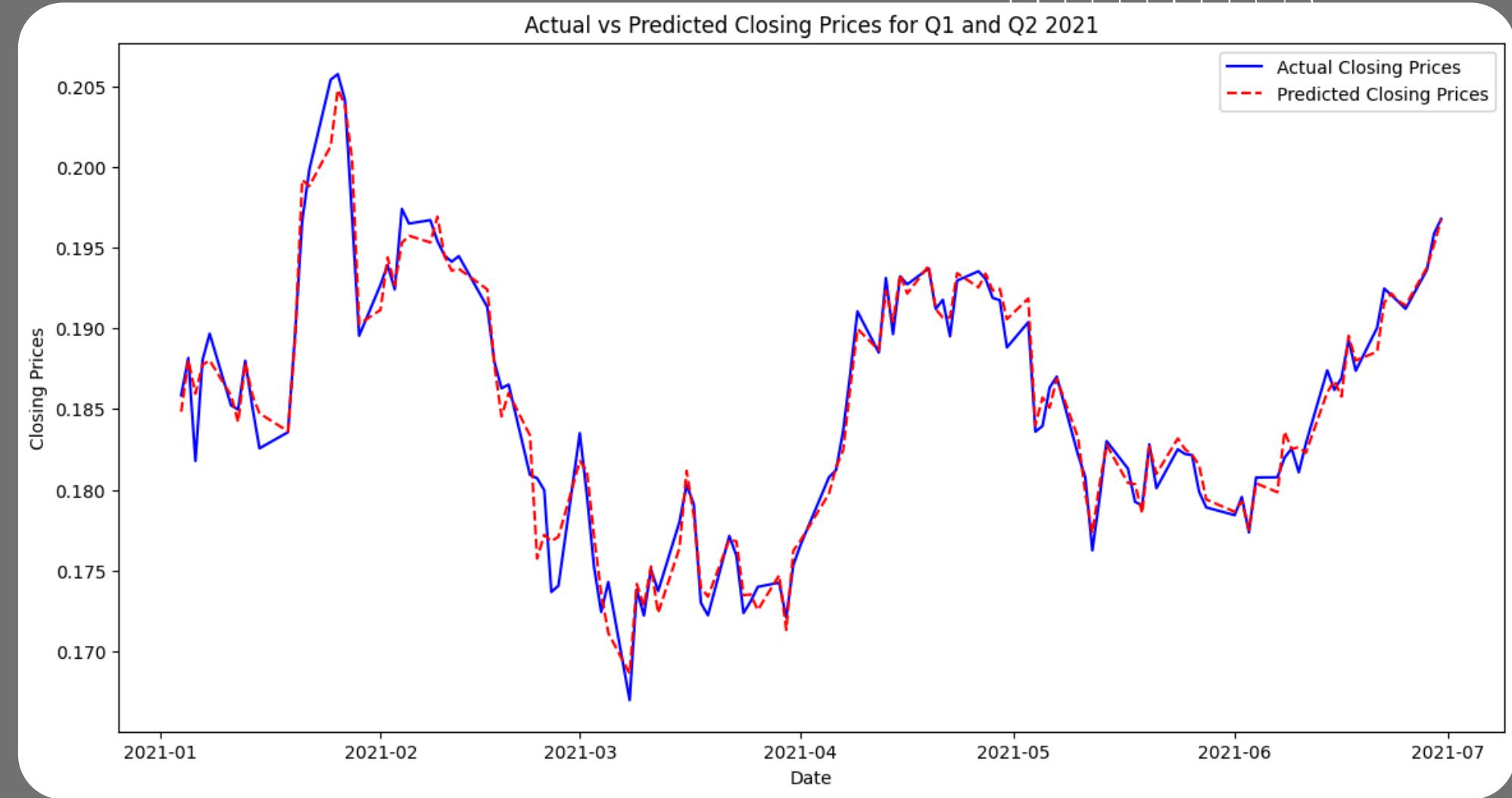
BIỂU ĐỒ SO SÁNH GIÁ TRỊ THỰC VÀ DỰ ĐOÁN

```
plt.figure(figsize=(14, 7))  
plt.plot(data_q1_q2_2021['date'], y_test, label='Actual Closing Prices', color='blue')  
plt.plot(data_q1_q2_2021['date'], y_pred_LN, label='Predicted Closing Prices', color='red', linestyle='--')  
plt.title('Actual vs Predicted Closing Prices for Q3 and Q4 2021')  
plt.xlabel('Date')  
plt.ylabel('Closing Prices')  
plt.legend()  
plt.show()
```

Actual vs Predicted Closing Prices for Q1 and Q2 2021



NHẬN XÉT



Giải thích biểu đồ:

1. Sự khớp nhau giữa giá trị thực tế và giá trị dự đoán:

- Đường màu đỏ (giá dự đoán) bám sát rất gần với đường màu xanh (giá thực tế) trong hầu hết các khoảng thời gian, cho thấy mô hình dự đoán khá chính xác.
- Mỗi đỉnh và đáy trên đường giá thực tế đều có đường dự đoán tương ứng, điều này thể hiện khả năng của mô hình trong việc bắt kịp các biến động ngắn hạn của thị trường.

2. Biến động của giá cổ phiếu:

- Biểu đồ thể hiện rõ các giai đoạn giá tăng và giảm trong quý 1 và quý 2 năm 2021.
- Có nhiều đoạn giá dao động mạnh, đặc biệt là vào cuối tháng 1, đầu tháng 2, và tháng 3. Mô hình vẫn dự đoán tương đối chính xác trong các giai đoạn này.

3. Độ chính xác của mô hình:

- Biểu đồ cho thấy mô hình Linear Regression đã được huấn luyện tốt và có khả năng dự đoán chính xác giá đóng cửa của cổ phiếu trong giai đoạn này.
- Đường dự đoán (màu đỏ) và đường thực tế (màu xanh) gần như trùng khớp nhau ở nhiều đoạn, thể hiện mức độ chính xác cao.

2 CODING

LỌC DỮ LIỆU QUÝ 3 VÀ QUÝ 4 NĂM 2021

```
data_q3_q4_2021 = stock_prices_df[(stock_prices_df['date'] >= '2021-07-01') & (stock_prices_df['date'] <= '2021-12-30') &  
(stock_prices_df['stock_symbol']=='AAPL')]
```

CHUẨN BỊ DỮ LIỆU KIỂM TRA

```
X_test = data_q3_q4_2021[['open', 'high', 'low', 'volume']]  
y_test = data_q3_q4_2021['close']
```

KHỞI TẠO VÀ HUẤN LUYỆN MÔ HÌNH

```
LN_model = LinearRegression()  
LN_model.fit(X_train, y_train)
```

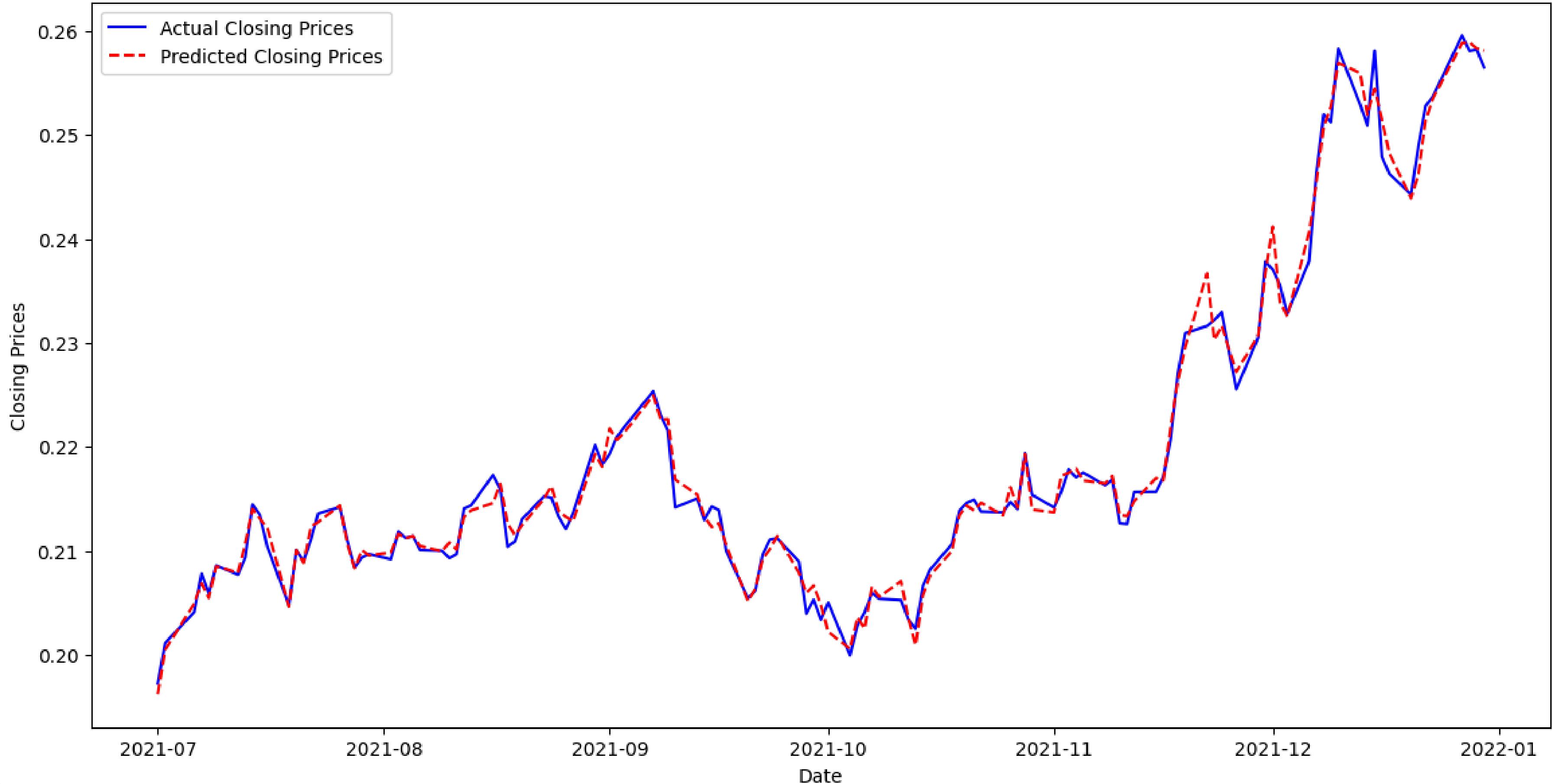
DỰ ĐOÁN

```
y_pred_LN = LN_model.predict(X_test)
```

BIỂU ĐỒ SO SÁNH GIÁ TRỊ THỰC VÀ DỰ ĐOÁN

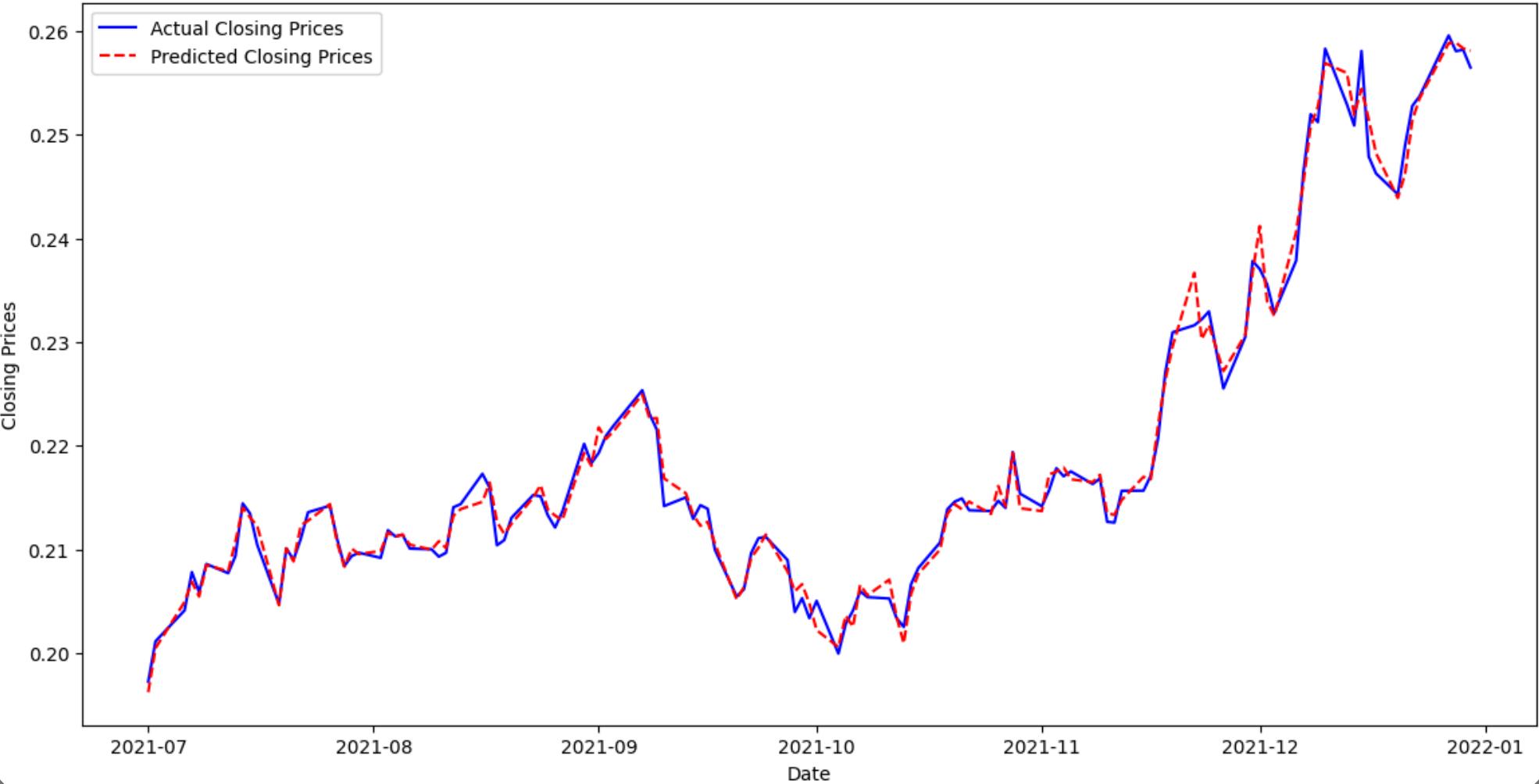
```
plt.figure(figsize=(14, 7))  
plt.plot(data_q3_q4_2021['date'], y_test, label='Actual Closing Prices', color='blue')  
plt.plot(data_q3_q4_2021['date'], y_pred_LN, label='Predicted Closing Prices', color='red', linestyle='--')  
plt.title('Actual vs Predicted Closing Prices for Q3 and Q4 2021')  
plt.xlabel('Date')  
plt.ylabel('Closing Prices')  
plt.legend()  
plt.show()
```

Actual vs Predicted Closing Prices for Q3 and Q4 2021



NHẬN XÉT

Actual vs Predicted Closing Prices for Q3 and Q4 2021



Giải thích Biểu đồ

1. Sự khớp nhau giữa giá trị thực tế và giá trị dự đoán:

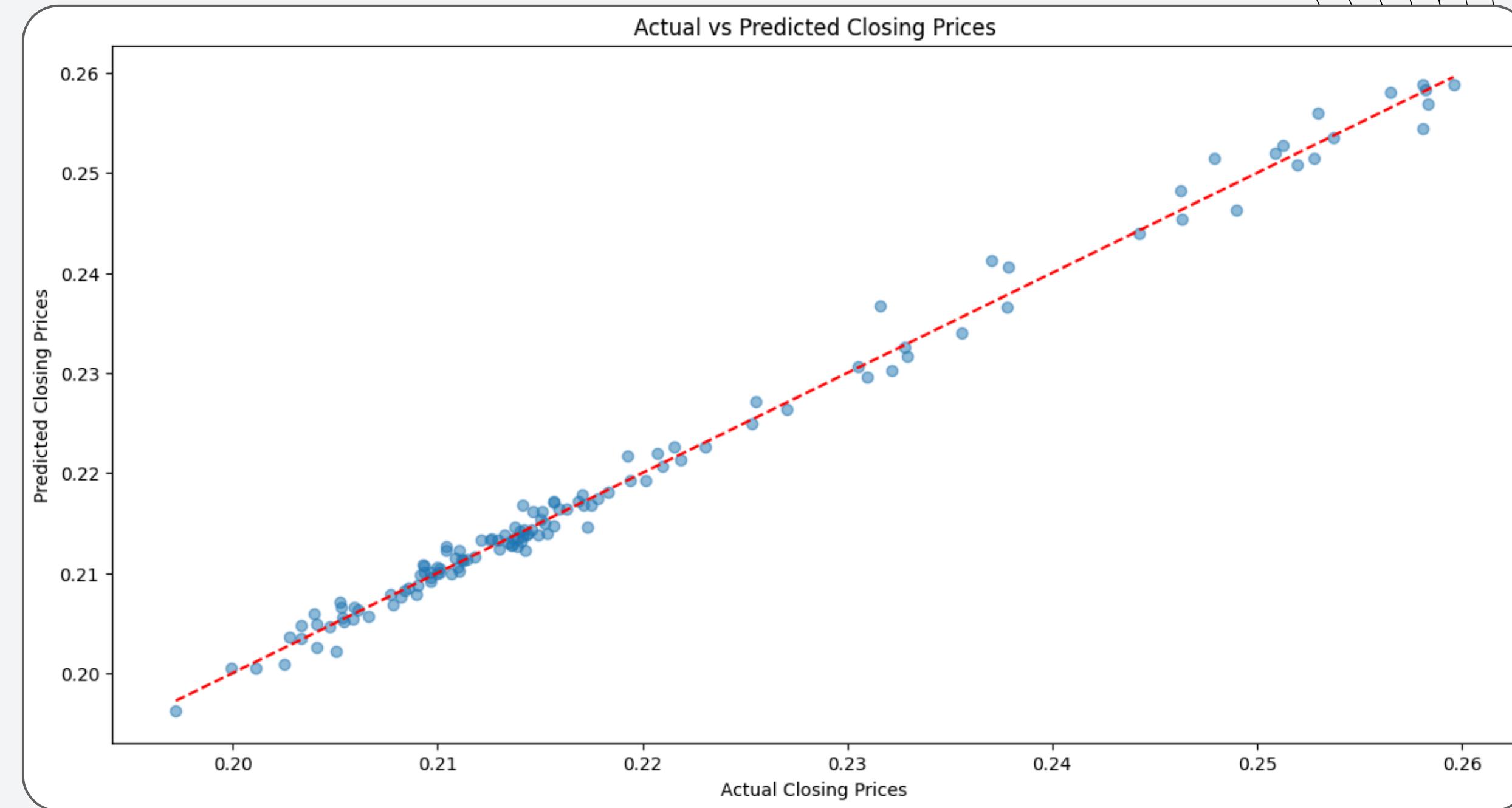
- Đường màu đỏ (giá dự đoán) bám sát rất gần với đường màu xanh (giá thực tế) trong suốt khoảng thời gian, cho thấy mô hình dự đoán khá chính xác.
- Mỗi đỉnh và đáy trên đường giá thực tế đều có đường dự đoán tương ứng, điều này thể hiện khả năng của mô hình trong việc bắt kịp các biến động ngắn hạn của thị trường

2. Biến động của giá cổ phiếu:

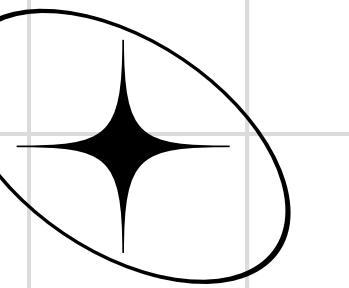
- Biểu đồ thể hiện rõ các giai đoạn giá tăng và giảm trong quý 3 và quý 4 năm 2021.
- Có nhiều đoạn giá dao động mạnh, đặc biệt là vào tháng 9, tháng 10 và tháng 12. Mô hình vẫn dự đoán tương đối chính xác trong các giai đoạn này.

3. Độ chính xác của mô hình:

- Biểu đồ cho thấy mô hình Linear Regression đã được huấn luyện tốt và có khả năng dự đoán chính xác giá đóng cửa của cổ phiếu trong giai đoạn này.
- Đường dự đoán (màu đỏ) và đường thực tế (màu xanh) gần như trùng khớp nhau ở nhiều đoạn, thể hiện mức độ chính xác cao.



- **Hiệu quả của mô hình:** Mô hình Linear Regression đã cho thấy hiệu quả tốt trong việc dự đoán giá đóng cửa của cổ phiếu trong năm 2021. Độ chính xác cao của mô hình này cho phép các nhà đầu tư sử dụng nó để dự đoán giá trong các giai đoạn ngắn hạn.
- **Ứng dụng thực tế:** Mô hình này có thể được sử dụng để hỗ trợ các quyết định đầu tư ngắn hạn, giúp nhà đầu tư xác định các điểm mua/bán tiềm năng dựa trên dự đoán giá tương lai. Tiếp tục theo dõi và điều chỉnh mô hình dựa trên dữ liệu mới để duy trì độ chính xác cao trong dự đoán.



THANK YOU

