

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC XÃ HỘI VÀ NHÂN VĂN
KHOA: THƯ VIỆN – THÔNG TIN HỌC



ĐỒ ÁN
KẾT THÚC MÔN HỌC, MÃ ĐỀ 2

Môn học:	Phân tích dữ liệu cho quản lý
Giảng viên hướng dẫn:	ThS. Trần Đình Anh Huy
Lớp:	Lớp B – QLTT K22
Sinh viên thực hiện:	Nghiêm Thị Ngọc Thảo – 2256210057 Ngô Nguyễn Ánh Trang – 2256210064

Tháng 12 năm 2024, Thành phố Hồ Chí

MỤC LỤC

DANH MỤC HÌNH.....	3
Chương 1: Tổng quan.....	5
1.1. Đặt vấn đề.....	5
1.2. Quy trình thực hiện	5
1.3. Công cụ sử dụng	6
1.4. Thông tin dự án và thành viên nhóm	7
Chương 2: Tiền xử lý dữ liệu	9
2.1. Thu thập dữ liệu	9
2.2. Làm sạch dữ liệu.....	10
Chương 3: Thống kê mô tả	12
3.1. Câu hỏi thống kê.....	12
3.2. Thống kê mô tả	12
3.2.1. Giá đóng	12
3.2.2. Lượng cổ phiếu giao dịch.....	16
3.2.3. Tỷ suất lợi nhuận.....	18
3.2.4. Nhận xét	19
3.3. So sánh các công ty	21
Chương 4: Yếu tố thời gian và mô hình dự báo.....	25
4.1. Yếu tố thời gian.....	25
4.2. Mô hình dự báo Moving Average	26
4.2.1. Mô hình dự báo Navie.....	26
4.2.2. Mô hình Moving Average 3-step	27
4.2.3. Mô hình Moving Average 6-step	27
4.3. Mô hình Exponential Smoothing	28
4.3.1. Hệ số $\alpha = 0.1$	28
4.3.2. Hệ số α tối ưu	29
4.4. Mô hình Holt	31
4.4.1. Mô hình Holt hệ số tiêu chuẩn	31
4.4.2. Mô hình Holt hệ số tối ưu	32
4.5. Mô hình Holt-winter.....	33
4.5.1. Mô hình Holt-winter với hệ số tiêu chuẩn	33
4.5.2. Mô hình Holt-winter với hệ tối ưu	33
4.6. Thống kê các chỉ số đo độ lệch	35

Chương 5: Thiết kế giao diện.....	36
5.1. Mô tả tính năng.....	36
5.2. Giao diện.....	36
KẾT LUẬN VÀ ĐỀ XUẤT	39
TÀI LIỆU THAM KHẢO	40
PHỤ LỤC.....	41

DANH MỤC HÌNH

Hình 2. 1. Cài đặt các thư viện cần thiết.....	9
Hình 2. 2. Tải dữ liệu của 5 công ty.....	10
Hình 2. 3. Tải dữ liệu của Microsoft.....	10
Hình 2. 4. Kiểm tra dữ liệu	11
Hình 2. 5. In ra các dòng dữ liệu đầu.....	11
Hình 3. 1. Mô tả các tham số thống kê	12
Hình 3. 2. Vẽ biểu đồ giá đóng (Close) theo ngày của Microsoft	12
Hình 3. 3. Biểu đồ giá đóng (Close) của Microsoft theo ngày	13
Hình 3. 4. Thống kê trung bình giá đóng (Close) theo năm của công ty Microsoft.....	13
Hình 3. 5. Biểu đồ trung bình giá đóng (Close) của Microsoft trong 5 năm (2020-11/2024)	14
Hình 3. 6. Thống kê trung bình giá đóng theo quý của các năm	14
Hình 3. 7. Biểu đồ trung bình giá đóng theo quý từ 2020 đến 2024 của công ty Microsoft	15
Hình 3. 8. Kết quả thống kê trung bình giá đóng theo quý của các năm.....	15
Hình 3. 9. Biểu đồ thống kê lượng cổ phiếu (Volume) giao dịch theo ngày từ năm 2020 đến 11/2024.....	16
Hình 3. 10. Biểu đồ tổng lượng cổ phiếu giao dịch trong các năm từ 2020 đến 11/2024 .	17
Hình 3. 11. Thống kê tổng lượng cổ phiếu giao dịch theo quý của các năm.....	17
Hình 3. 12. Biểu đồ cột tổng lượng giao dịch cổ phiếu theo quý từ 2020 đến 2024.....	18
Hình 3. 13. Biểu đồ tỷ suất lợi nhuận hàng ngày trong năm 2024 của Microsoft.....	19
Hình 3. 14. Biểu đồ tương quan giữa các biến	20
Hình 3. 16. Biểu đồ giá đóng của 5 công ty từ 2020 đến 11/2024	21
Hình 3. 17. Thống kê trung bình giá đóng theo năm của 5 công ty.....	21
Hình 3. 18. Biểu đồ trung bình giá đóng theo năm của 5 công ty	22
Hình 3. 19. Biểu đồ nhiệt tương quan giá đóng của 5 công ty	23
Hình 3. 20. Biểu đồ tỷ suất lợi nhuận theo ngày của Microsoft trong 5 năm (2020-2024)	24
Hình 4. 1. Biểu đồ phân rã chuỗi thời gian Microsoft	25
Hình 4. 2. Biểu đồ mô hình dự báo Moving average (Navie)	26
Hình 4. 3. Biểu đồ mô hình dự báo Moving average (3-step)	27
Hình 4. 4. Biểu đồ mô hình Moving average (6-step)	28
Hình 4. 5. Biểu đồ mô hình dự báo Simple Exponential Smoothing ($\alpha = 0.1$).....	29
Hình 4. 6. Biểu đồ mô hình dự báo Simple exponential smoothing với hệ số tối ưu.....	30
Hình 4. 7. Biểu đồ mô hình dự báo Holt với hệ số tiêu chuẩn	31
Hình 4. 8. Biểu đồ mô hình dự báo Holt với hệ số tối ưu.....	32
Hình 4. 9. Biểu đồ mô hình dự báo Holt-winter với hệ số chuẩn.....	33
Hình 4. 10. Biểu đồ mô hình dự báo Holt-winter với hệ số tối ưu	35
Hình 4. 11. Thống kê các chỉ số đo độ lệch.....	35

Hình 5. 1. Giao diện trang chủ.....	36
Hình 5. 2. Giao diện thống kê mô tả.....	36
Hình 5. 3. Giao diện chức năng phân rã chuỗi thời gian	37
Hình 5. 4. Giao diện phân tích chuỗi thời gian.....	37
Hình 5. 5. Giao diện chức năng tương quan của phần dư.....	37
Hình 5. 6. Giao diện chức năng mô hình dự báo	38
Hình 5. 7. Giao diện khi dự báo theo mô hình mà người dùng lựa chọn	38

Chương 1: Tổng quan

1.1. Đặt vấn đề

Trước bối cảnh công nghệ phát triển vượt bậc, thị trường chứng khoán trong lĩnh vực này cũng có nhiều biến động lớn, chịu tác động từ các yếu tố có thể kể đến như chuyển đổi số toàn cầu, xu hướng phát triển AI, Big Data, Clouding. Nhằm thực hiện phân tích đánh giá chứng khoán trong lĩnh vực công nghệ, năm công ty được lựa chọn gồm IBM, Oracle, Microsoft, Salesforce và SAP — những tên tuổi hàng đầu trong lĩnh vực phần mềm và giải pháp công nghệ doanh nghiệp. Việc phân tích và dự báo cổ phiếu của các công ty này sẽ giúp cung cấp cái nhìn tổng quan về thị trường công nghệ, xu hướng tăng trưởng, cũng như cơ hội đầu tư triển vọng.

1.2. Quy trình thực hiện

Quá trình phân tích và dự báo chứng khoán của 5 công ty trải qua 4 giai đoạn. Giai đoạn đầu tiên là tiền xử lý dữ liệu, dữ liệu được thu thập từ Yahoo Finance và được làm sạch để loại bỏ các giá trị bị thiếu, giá trị ngoại lai hoặc các vấn đề không đồng nhất. Sau đó, dữ liệu được chuyển đổi để phù hợp với các phân tích tiếp theo, bao gồm xử lý dạng ngày tháng và tính toán thêm các chỉ số kỹ thuật (nếu cần).

Tiếp theo là giai đoạn phân tích thống kê mô tả, tập trung vào trả lời các câu hỏi thống kê thông qua việc sử dụng các tham số như giá trị lớn nhất, giá trị nhỏ nhất, giá trị trung bình, độ lệch chuẩn và mức độ biến động của giá cổ phiếu qua thời gian. Các kết quả này được trình bày bằng các biểu đồ trực quan như biểu đồ hộp, biểu đồ đường, hoặc biểu đồ phân định phối, giúp làm rõ xu hướng và đặc điểm của dữ liệu.

Trong giai đoạn phân tích chuỗi thời gian và dự báo, dữ liệu được kiểm tra tính ổn định và tính theo mùa, sau đó áp dụng các mô hình dự báo. Các mô hình được sử dụng bao gồm Moving Average (mô hình trung bình trượt đơn), Simple Exponential Smoothing (mô hình làm mịn theo hàm mũ), Holt và Holt-winter.

Cuối cùng, giai đoạn thiết kế giao diện dự báo tạo ra một giao diện trực quan để người dùng dễ dàng theo dõi kết quả dự báo.

1.3. Công cụ sử dụng

Python được lựa chọn làm công cụ phân tích vì tính linh hoạt, cùng với đó là các thư viện hỗ trợ việc xử lý dữ liệu và phân tích. Dưới đây là bảng trình bày các thư viện được sử dụng phục vụ cho quá trình phân tích và dự báo (Bảng 1.1).

	Thư viện	Chức năng
Tính toán, thống kê	Numpy	Là thư viện cung cấp mảng đa chiều, các phép toán nhanh trên mảng bao gồm toán học, logic, sắp xếp, thống kê cơ bản và nhiều tính năng khác (What Is NumPy? — NumPy v2.1 Manual, n.d.).
	Pandas	Thư viện làm việc với dữ liệu dạng bảng (DataFrame). Thư viện hỗ trợ nhập và xuất dữ liệu từ nhiều định dạng (CSV, Excel, SQL, JSON, v.v.), tính toán thống kê, làm sạch, lọc, nhóm và biến đổi dữ liệu (Pandas - Python Data Analysis Library, n.d.).
	Math	Thư viện cung cấp các hàm toán học cơ bản
	Datetime	Thư viện cung cấp các lớp (classes) để xử lý ngày tháng và thời gian như tính toán khoảng thời gian, định dạng và chuyển đổi ngày giờ và nhiều tính năng khác.
	Yfinance	Thư viện truy cập và lấy dữ liệu chứng khoán từ Yahoo Finance.
Trực quan hóa dữ liệu	Matplotlib	Matplotlib là một thư viện đồ họa mạnh mẽ và linh hoạt trong Python, thường được sử dụng để tạo các đồ thị và hình ảnh, thích hợp cho việc tạo các biểu đồ 2D.

	Seaborn	Seaborn là một thư viện đồ họa dựa trên Matplotlib và cung cấp một giao diện cao cấp hơn để tạo các biểu đồ thống kê.
	Mplfinance	Thư viện được phát triển từ Matplotlib thường được sử dụng cho các biểu đồ tài chính như biểu đồ hình nến (candlestick chart).
Mô hình dự báo	Optuna	Optuna là một công cụ mã nguồn mở nhằm tối ưu hóa siêu tham số để tự động hóa tìm kiếm siêu tham số. Thường được sử dụng cho Deep Learning hoặc Machine Learning.
	Statsmodel.tsa.seasonal	Chứa các lớp mô hình và hàm hữu ích cho phân tích chuỗi thời gian. (Time Series Analysis Tsa — Statsmodels 0.9.0 Documentation, n.d.)
	Statsmodels.tsa.holtwinters	Chứa các lớp mô hình và hàm hữu ích cho làm mịn theo hàm mũ.
	Sklearn.metrics	Thư viện cung cấp các chỉ số đánh giá độ sai lệch của các mô hình dự báo bao gồm sai số bình phương trung bình (MSE) và sai số trung bình tuyệt đối (MAE).

1.4. Thông tin dự án và thành viên nhóm

Link Github dự án: <https://github.com/trangtrang2508/PHANTICHDULIEUCOPHIEU>

Link website dự án: <https://phantichdulieucophieu.streamlit.app/>

Dự án của nhóm được đăng tải trên Github. Cùng với đó một trang web được xây dựng nhằm thử nghiệm thống kê và dự báo chứng khoán. Dưới đây là bảng phân chia công việc 2 thành viên nhóm.

Họ và tên	Mã số sinh viên	Công việc
Nghiêm Thị Ngọc Thảo	2256210057	<ul style="list-style-type: none">• Phân tích cổ phiếu• Báo cáo
Ngô Nguyễn Ánh Trang	2256210064	<ul style="list-style-type: none">• Phân tích cổ phiếu• Xây dựng trang web

Chương 2: Tiền xử lý dữ liệu

2.1. Thu thập dữ liệu

Cài đặt các thư viện cần thiết:

```
import numpy as np
import pandas as pd
import math as mt
import datetime as dt
import optuna as opt
import yfinance as yf

import matplotlib.pyplot as plt
import seaborn as sns
import mplfinance as mpf

from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_acf
from scipy.stats import pearsonr
from statsmodels.tsa.holtwinters import SimpleExpSmoothing
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from statsmodels.tsa.holtwinters import Holt
from sklearn.metrics import mean_squared_error, mean_absolute_error
```

Hình 2. 1. Cài đặt các thư viện cần thiết

Dữ liệu được thu thập từ Yahoo Finance từ 5 mã cổ phiếu tương ứng với 5 công ty trong khoảng thời gian từ 01/01/2020 đến 30/11/2024. Trong đó, Microsoft có mã cổ phiếu là MSFT, Oracle là ORCL, IBM là IBM, Salesforce là CRM và SAP là SAP. Với 6 biến thu thập, cụ thể:

- Open: giá mở cửa, mức giá giao dịch bắt đầu khi thị trường mở cửa
- High: giá cổ phiếu đạt cao nhất trong ngày.
- Low: giá cổ phiếu thấp nhất trong ngày.
- Close: Giá đóng cửa, giá giao dịch cuối của phiếu trong ngày.
- Adj Close: giá đóng cửa điều chỉnh.
- Volume: Khối lượng giao dịch trong ngày, tức là tổng số lượng cổ phiếu hoặc tài sản được mua và bán trong ngày.

```
#Gán mã cổ phiếu
tickers = ['MSFT', 'ORCL', 'IBM', 'CRM', 'SAP']
#Tải dữ liệu cổ phiếu của 5 công ty trong 5 năm
data = yf.download(tickers, start='2020-01-01', end='2024-11-30', group_by='ticker')
#In thông tin dữ liệu
data.info()
```

```
[*****100%*****] 5 of 5 completed<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1237 entries, 2020-01-02 00:00:00+00:00 to 2024-11-29 00:00:00+00:00
Data columns (total 30 columns):
#   Column                Non-Null Count  Dtype
---  -
0   (MSFT, Open)           1237 non-null   float64
1   (MSFT, High)           1237 non-null   float64
2   (MSFT, Low)            1237 non-null   float64
3   (MSFT, Close)          1237 non-null   float64
4   (MSFT, Adj Close)      1237 non-null   float64
5   (MSFT, Volume)         1237 non-null   int64
6   (SAP, Open)            1237 non-null   float64
7   (SAP, High)            1237 non-null   float64
8   (SAP, Low)             1237 non-null   float64
9   (SAP, Close)           1237 non-null   float64
10  (SAP, Adj Close)       1237 non-null   float64
11  (SAP, Volume)          1237 non-null   int64
12  (IBM, Open)            1237 non-null   float64
13  (IBM, High)            1237 non-null   float64
14  (IBM, Low)             1237 non-null   float64
15  (IBM, Close)           1237 non-null   float64
16  (IBM, Adj Close)       1237 non-null   float64
17  (IBM, Volume)          1237 non-null   int64
18  (CRM, Open)            1237 non-null   float64
19  (CRM, High)            1237 non-null   float64
20  (CRM, Low)             1237 non-null   float64
21  (CRM, Close)           1237 non-null   float64
22  (CRM, Adj Close)       1237 non-null   float64
23  (CRM, Volume)          1237 non-null   int64
24  (ORCL, Open)           1237 non-null   float64
25  (ORCL, High)           1237 non-null   float64
26  (ORCL, Low)            1237 non-null   float64
27  (ORCL, Close)          1237 non-null   float64
28  (ORCL, Adj Close)      1237 non-null   float64
29  (ORCL, Volume)         1237 non-null   int64
dtypes: float64(25), int64(5)
memory usage: 299.6 KB
```

Hình 2. 2. Tải dữ liệu của 5 công ty

Trong đó, Microsoft là công ty được chọn để thực hiện phân tích thống kê mô tả.

```
#Tải dữ liệu cổ phiếu của Microsoft trong 5 năm
microsoft_data = yf.download('MSFT', start='2020-01-01', end='2024-11-30')
#In thông tin dữ liệu
microsoft_data.info()
```

```
[*****100%*****] 1 of 1 completed<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1237 entries, 2020-01-02 00:00:00+00:00 to 2024-11-29 00:00:00+00:00
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  -
0   (Adj Close, MSFT)      1237 non-null   float64
1   (Close, MSFT)          1237 non-null   float64
2   (High, MSFT)           1237 non-null   float64
3   (Low, MSFT)            1237 non-null   float64
4   (Open, MSFT)           1237 non-null   float64
5   (Volume, MSFT)         1237 non-null   int64
dtypes: float64(5), int64(1)
memory usage: 67.6 KB
```

Hình 2. 3. Tải dữ liệu của Microsoft

2.2. Làm sạch dữ liệu

Để thực hiện việc làm sạch dữ liệu cần thống kê các dữ liệu thiếu, dữ liệu trống và trùng lặp.

```
#Kiểm tra dữ liệu thiếu, dữ liệu trống, dữ liệu trùng lặp
print("Dữ liệu thiếu: ", microsoft_data.isnull().values.sum())
print("Dữ liệu trống: ", microsoft_data.isna().values.any())
print("Dữ liệu trùng lặp: ", microsoft_data.duplicated().values.sum())
```

Dữ liệu thiếu: 0
Dữ liệu trống: False
Dữ liệu trùng lặp: 0

Hình 2. 4. Kiểm tra dữ liệu

Kết quả kiểm tra cho thấy không có dữ liệu bị thiếu, trống hay trùng lặp. Kết quả trả về khi mô tả dữ liệu thông qua câu lệnh `.info()` hiển thị dữ liệu Microsoft được thu thập 1237 dòng tính từ ngày 02/01/2020 đến 29/11/2024 với 6 cột(hình 2.3).

```
#In các dòng dữ liệu
microsoft_data.head()
```

Price	Adj Close	Close	High	Low	Open	Volume
Ticker	MSFT	MSFT	MSFT	MSFT	MSFT	MSFT
Date						
2020-01-02 00:00:00+00:00	153.630661	160.619995	160.729996	158.330002	158.779999	22622100
2020-01-03 00:00:00+00:00	151.717743	158.619995	159.949997	158.059998	158.320007	21116200
2020-01-06 00:00:00+00:00	152.109863	159.029999	159.100006	156.509995	157.080002	20813700
2020-01-07 00:00:00+00:00	150.722977	157.580002	159.669998	157.320007	159.320007	21634100
2020-01-08 00:00:00+00:00	153.123749	160.089996	160.800003	157.949997	158.929993	27746500

Hình 2. 5. In ra các dòng dữ liệu đầu

Chương 3: Thống kê mô tả

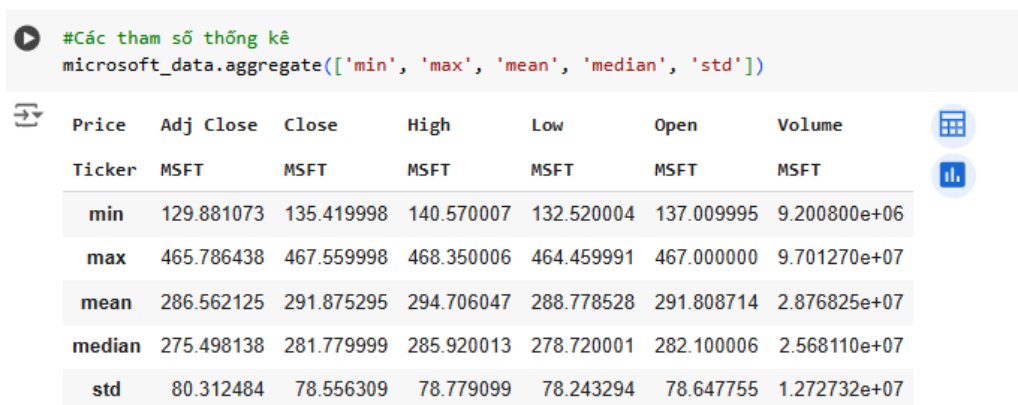
3.1. Câu hỏi thống kê

Với dữ liệu thu thập được, giá đóng (Close) và số lượng cổ phiếu bán ra (Volume) được sử dụng để phân tích mô tả chủ yếu. Các câu hỏi thống kê được đặt ra như sau:

1. Giá đóng cao/thấp nhất là bao nhiêu?
2. Giá đóng trung bình mỗi năm là bao nhiêu?
3. Trung bình giá đóng của các quý năm 2024 so với năm 2022, 2023 như thế nào?
4. Lượng cổ phiếu giao dịch mỗi năm là bao nhiêu?
5. Tỷ suất lợi nhuận theo ngày của công ty trong năm 2024 giao động trong khoảng nào?

3.2. Thống kê mô tả

Hàm aggregate được sử dụng mô tả các tham số thống kê.

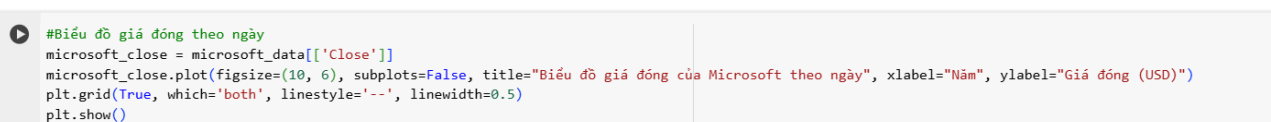


Hình 3. 1. Mô tả các tham số thống kê

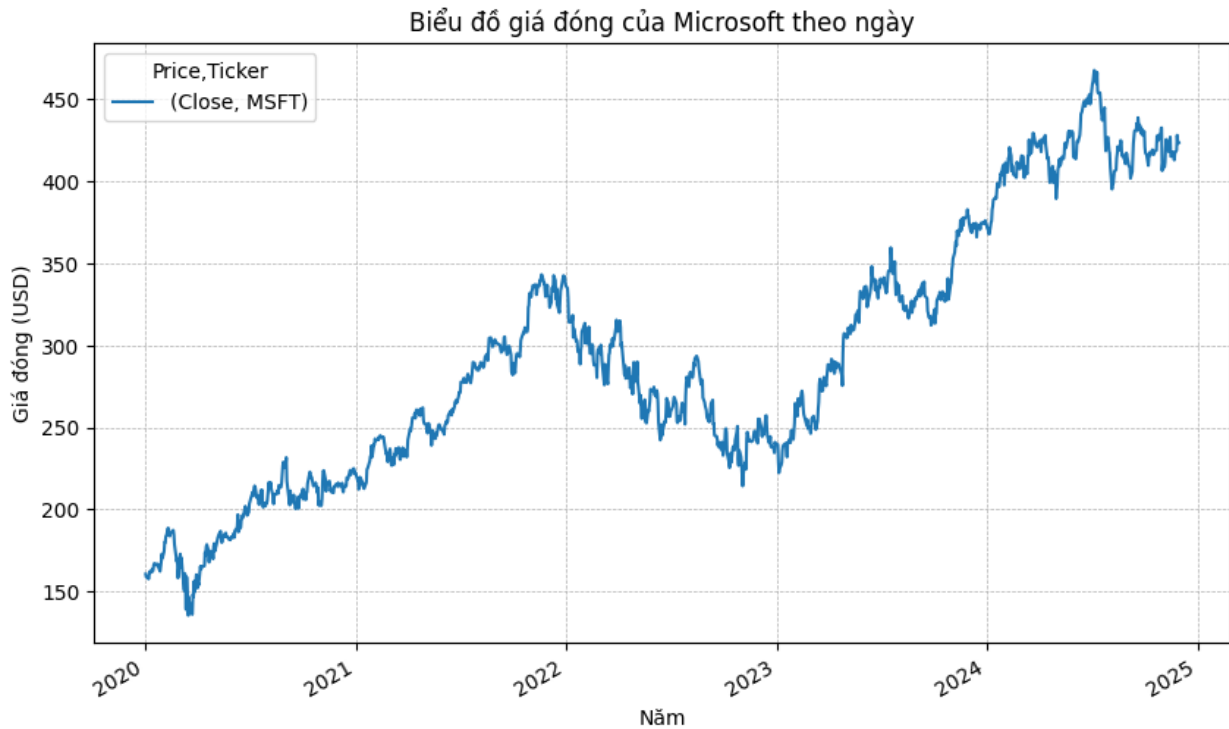
3.2.1. Giá đóng

Dùng hàm `microsoft_data[['Close']]`: lọc biến giá đóng từ dataframe `microsoft_data`.

`microsoft_close.plot()`: vẽ biểu đồ đường giá đóng của Microsoft.



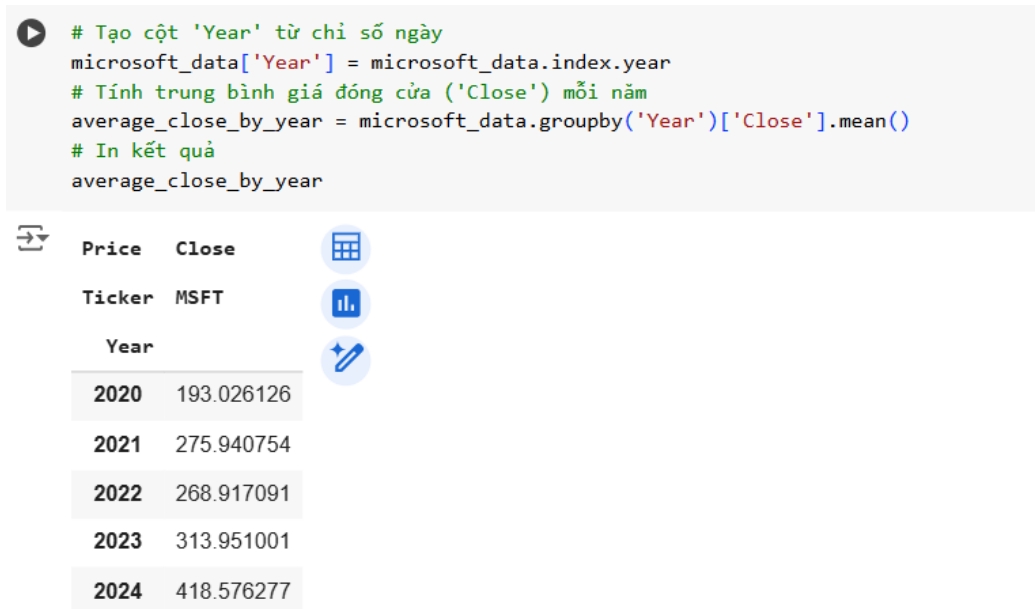
Hình 3. 2. Vẽ biểu đồ giá đóng (Close) theo ngày của Microsoft



Hình 3. 3. Biểu đồ giá đóng (Close) của Microsoft theo ngày

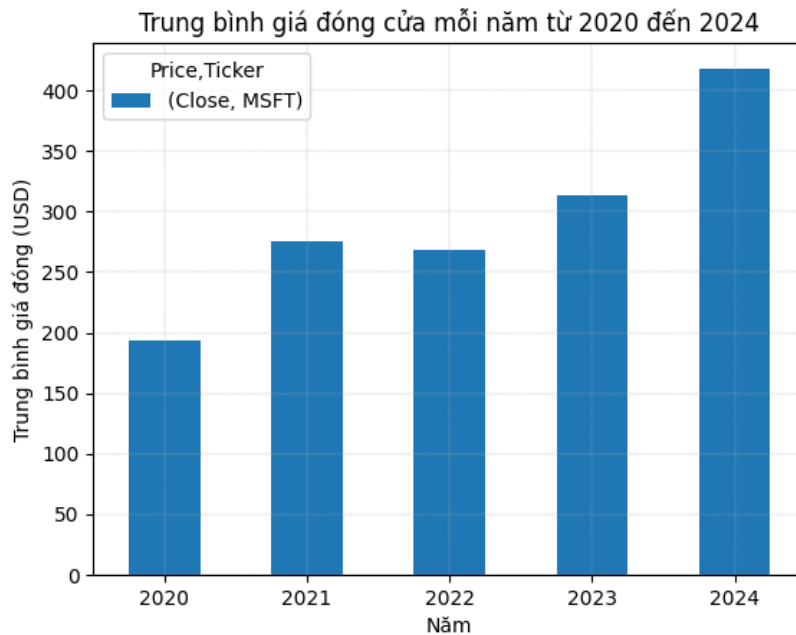
Biểu đồ trên có thể thấy giá đóng theo ngày trong các năm có xu hướng tăng, ngoại trừ năm 2022 có xu hướng giá đóng giảm.

Thống kê trung bình giá đóng theo năm của Microsoft. Kết quả cho thấy từ năm 2023 đến 2024 trung bình giá đóng tăng mạnh so với các năm trước.



Hình 3. 4. Thống kê trung bình giá đóng (Close) theo năm của công ty Microsoft

```
# Vẽ biểu đồ cột
plt.figure(figsize=(10, 8))
average_close_by_year.plot(kind='bar')
plt.title('Trung bình giá đóng cửa mỗi năm từ 2020 đến 2024')
plt.xlabel('Năm')
plt.ylabel('Trung bình giá đóng (USD)')
plt.xticks(rotation=0)
plt.yticks()
plt.grid(True, which='both', linestyle='--', linewidth=0.3)
plt.show()
```



Hình 3. 5. Biểu đồ trung bình giá đóng (Close) của Microsoft trong 5 năm (2020-11/2024)

Từ biểu đồ cột trung bình giá đóng từ năm 2020 đến 2024 có thể thấy, giá đóng tăng dần từ năm 2022 và 2024 có mức giá đóng vượt trội hơn so với các năm trước, đạt trên 400 USD, cao nhất trong 5 năm. Bên cạnh đó, trung bình giá đóng tăng mạnh từ năm 2020 qua năm 2021, và từ năm 2023 qua năm 2024.

Để phân tích biến động giá đóng, trung bình giá đóng các quý trong năm được phân tích dưới đây. Đáng chú ý, năm 2022 trung bình giá đóng giảm qua các quý.

```
microsoft_close = microsoft_data[['Close']].copy()
microsoft_close['Date'] = microsoft_close.index
microsoft_close['Date'] = pd.to_datetime(microsoft_close['Date'])

# Thêm cột 'Year' và 'Quarter' từ cột 'Date'
microsoft_close['Year'] = microsoft_close['Date'].dt.year
microsoft_close['Quarter'] = microsoft_close['Date'].dt.quarter
microsoft_close = microsoft_close[~((microsoft_close['Year'] == 2024) & (microsoft_close['Quarter'] == 4))]

# Tính trung bình giá đóng cửa theo từng năm và từng quý
avg_close_by_quarter = microsoft_close.groupby(['Year', 'Quarter'])['Close'].mean()

# Chuyển đổi dữ liệu sang dạng bảng với hàng là quý và cột là năm
avg_close_pivot = avg_close_by_quarter.unstack(level=0)
avg_close_pivot.index.name = 'Quarter'
avg_close_pivot.columns.name = 'Year'
avg_close_pivot.index = [f"Q{q}" for q in avg_close_pivot.index]

# In kết quả
print("Trung bình giá đóng cửa theo từng quý (2020-2024):")
avg_close_pivot
```

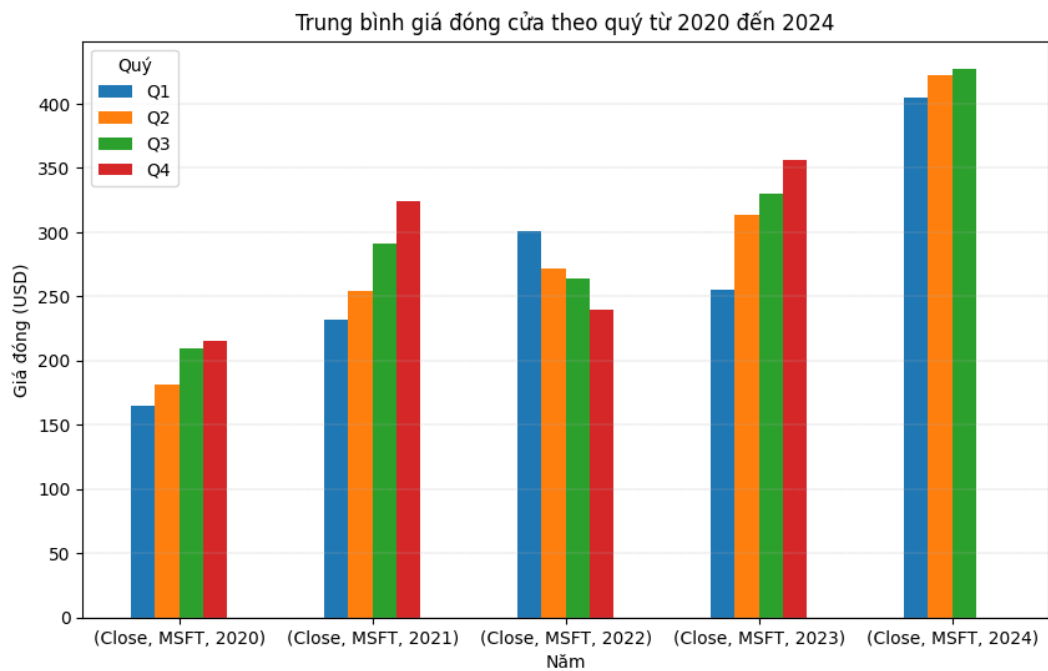
Hình 3. 6. Thống kê trung bình giá đóng theo quý của các năm

Trung bình giá đóng cửa theo từng quý (2020-2024):

Price	Close				
Ticker	MSFT				
Year	2020	2021	2022	2023	2024
Q1	164.449032	232.021475	300.853871	255.079840	404.821639
Q2	181.524921	254.182379	271.403225	313.408064	422.319524
Q3	210.033437	290.950313	264.049374	330.442857	427.465623
Q4	215.024375	324.210157	239.985555	355.930161	NaN

Hình 3. 8. Kết quả thống kê trung bình giá đóng theo quý của các năm

```
# Chuyển đổi dữ liệu để hàng là năm và cột là quý
avg_close_pivot_transposed = avg_close_pivot.T
# Vẽ biểu đồ cột
avg_close_pivot_transposed.plot(kind='bar', figsize=(10, 6))
plt.title("Trung bình giá đóng cửa theo quý từ 2020 đến 2024")
plt.xlabel("Năm")
plt.ylabel("Giá đóng (USD)")
plt.xticks(rotation=0)
plt.yticks()
plt.legend(title="Quý", fontsize=10)
plt.grid(axis='y', linestyle='--', linewidth=0.3)
plt.show()
```

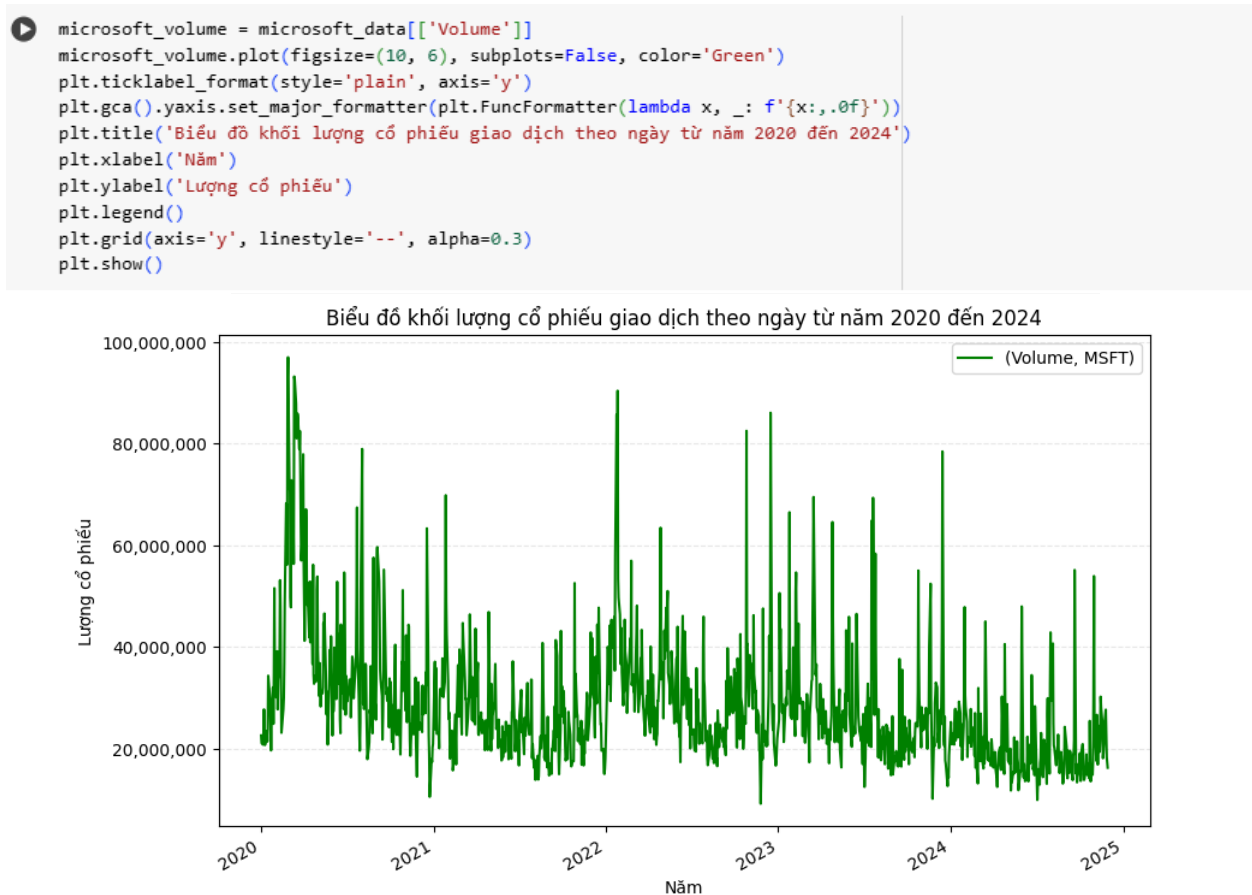


Hình 3. 7. Biểu đồ trung bình giá đóng theo quý từ 2020 đến 2024 của công ty Microsoft

Nhìn chung, trung bình giá đóng của các quý tăng các năm ngoại trừ năm 2022. Quý 4 (màu đỏ) có xu hướng đạt giá trung bình cao nhất trong 4 quý. Năm 2024 sự chênh lệch 3 quý đầu không quá lớn so với các năm trước cho thấy sự ổn định giá đóng.

3.2.2. Lượng cổ phiếu giao dịch

Lượng cổ phiếu giao dịch theo ngày được mô tả trong biểu đồ dưới đây (xem hình 3.9). Ngày giao dịch với lượng cổ phiếu đạt cao nhất nằm trong năm 2020 đạt gần 100 triệu cổ phiếu/ngày. Năm 2020 cũng là năm có nhiều phiên giao dịch với khối lượng lớn cổ phiếu nhất so với các năm về sau.



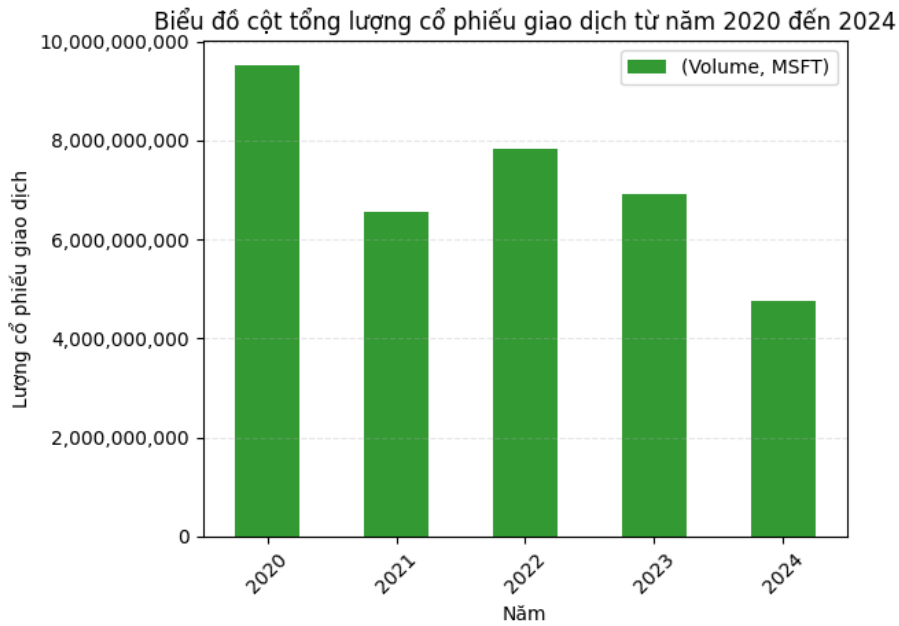
Hình 3. 9. Biểu đồ thống kê lượng cổ phiếu (Volume) giao dịch theo ngày từ năm 2020 đến 11/2024

Qua thống kê tổng lượng cổ phiếu giao dịch của mỗi năm có thể thấy được lượng cổ phiếu giao dịch đang giảm dần từ năm 2022. Năm 2020 là năm có lượng cổ phiếu giao dịch lớn nhất trong 5 năm.

```

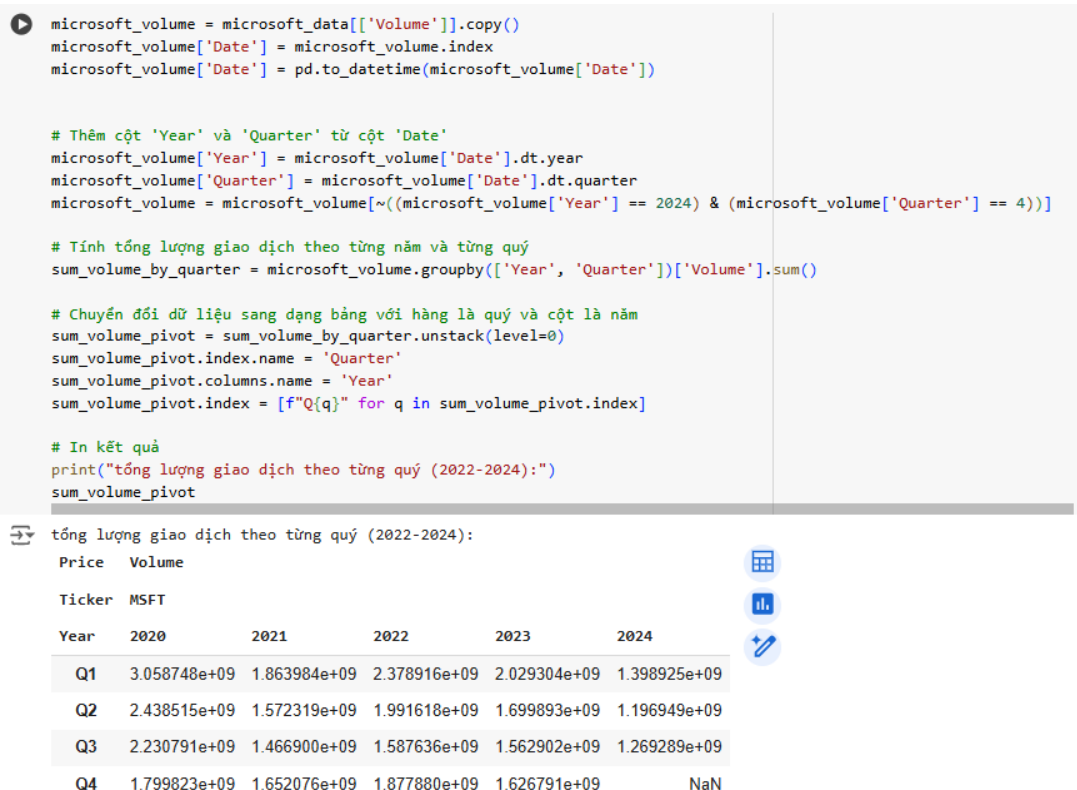
plt.figure(figsize=(16, 8))
volume_by_year.plot(kind='bar', alpha=0.8, color='Green')
plt.title('Biểu đồ cột tổng lượng cổ phiếu giao dịch từ năm 2020 đến 2024')
plt.ylabel('Lượng cổ phiếu giao dịch')
plt.xlabel('Năm')
plt.ticklabel_format(style='plain', axis='y')
plt.gca().yaxis.set_major_formatter(plt.FuncFormatter(lambda x, _: f'{x:,.0f}'))
plt.xticks(rotation=45)
plt.legend()
plt.grid(axis='y', linestyle='--', alpha=0.3)
plt.tight_layout()
plt.show()

```



Hình 3. 10. Biểu đồ tổng lượng cổ phiếu giao dịch trong các năm từ 2020 đến 11/2024

Tương tự với giá đóng, tổng lượng giao dịch cổ phiếu được thống kê theo các quý mỗi năm. Từ năm 2020 đến 2024, lượng giao dịch cổ phiếu của Microsoft có xu hướng giảm dần qua các năm. Năm 2020 có lượng giao dịch cao nhất, đặc biệt trong quý 1, trong khi năm 2024 ghi nhận mức thấp nhất. Trong mỗi năm, lượng giao dịch thường cao nhất vào quý 1 và giảm dần qua các quý tiếp theo. Trong 3 năm từ 2021 đến 2023 có sự tương đồng khi quý 1 là quý có lượng cổ phiếu giao dịch cao nhất giảm dần 2 quý sau và quý 4 tăng trở lại.



Hình 3. 11. Thống kê tổng lượng cổ phiếu giao dịch theo quý của các năm



Hình 3. 12. Biểu đồ cột tổng lượng giao dịch cổ phiếu theo quý từ 2020 đến 2024

3.2.3. Tỷ suất lợi nhuận

Tỷ suất lợi nhuận là tỷ lệ phần trăm lợi nhuận nhận được từ số tiền ban đầu. Với công thức tính:

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}} \quad (\text{Returns and Log Returns, n.d.})$$

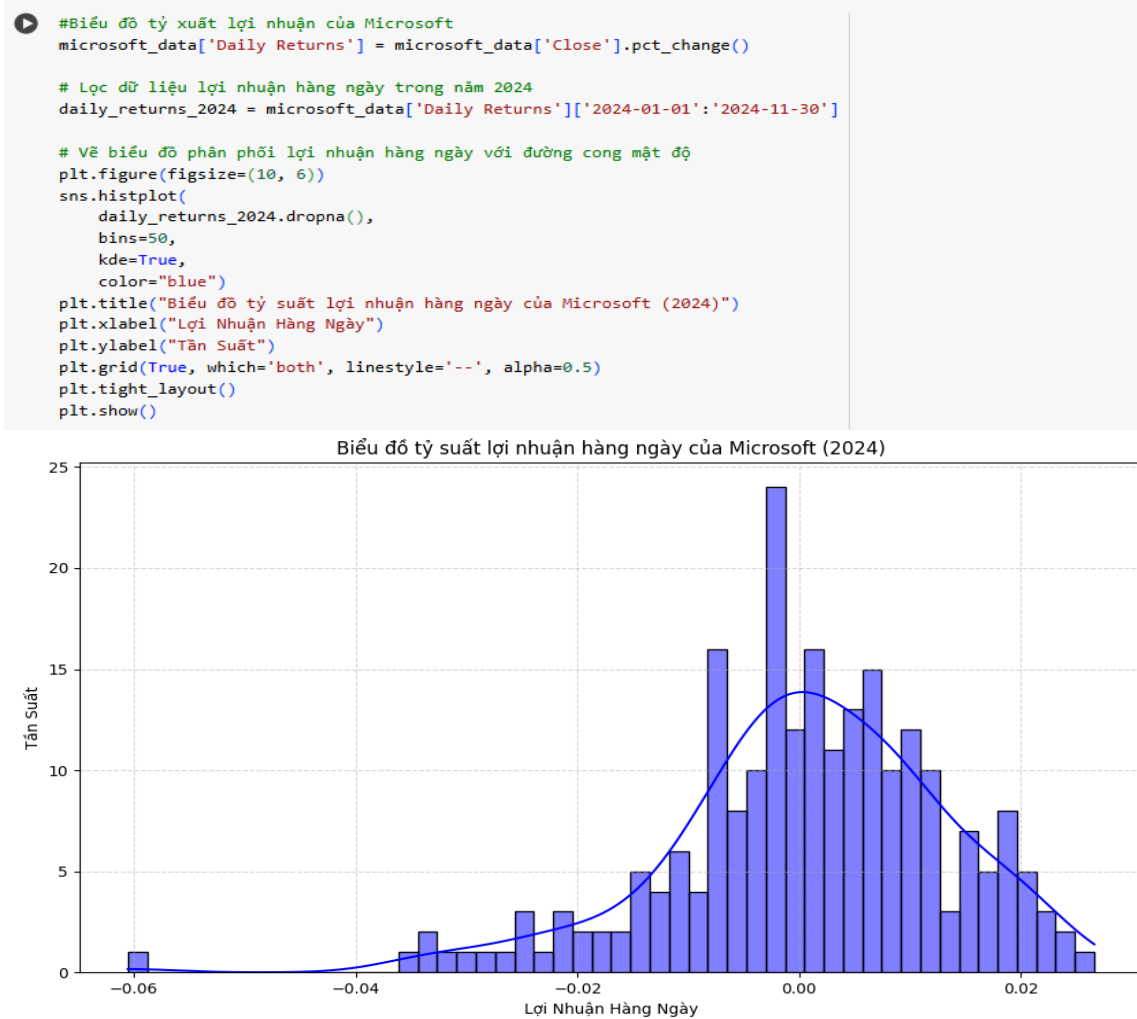
Trong đó: r_t : tỷ suất lợi nhuận.

P_t : giá cổ phiếu tại thời điểm t .

P_{t-1} : giá cổ phiếu tại thời điểm trước đó ($t-1$).

Nếu $r_t > 0$ thì sinh lời, ngược lại $r_t < 0$ lợi nhuận âm. Trong Python hàm `pct_change()` được sử dụng để tính tỷ suất lợi nhuận cổ phiếu. Dưới đây là biểu đồ tỷ suất lợi nhuận hàng ngày của Microsoft trong năm 2024.

Nhìn chung, tỷ suất lợi nhuận tập trung chủ yếu quanh giá trị 0, điều này cho thấy đa số các ngày giao dịch tỷ suất lợi nhuận của Microsoft biến động không lớn. Một số ngày có tỷ suất lợi nhuận âm thấp nhất là -0.06 và dương lớn khoảng 0.02, nhưng tần suất xảy ra thấp, thể hiện các biến động bất thường là hiếm.



3.2.4. Nhận xét

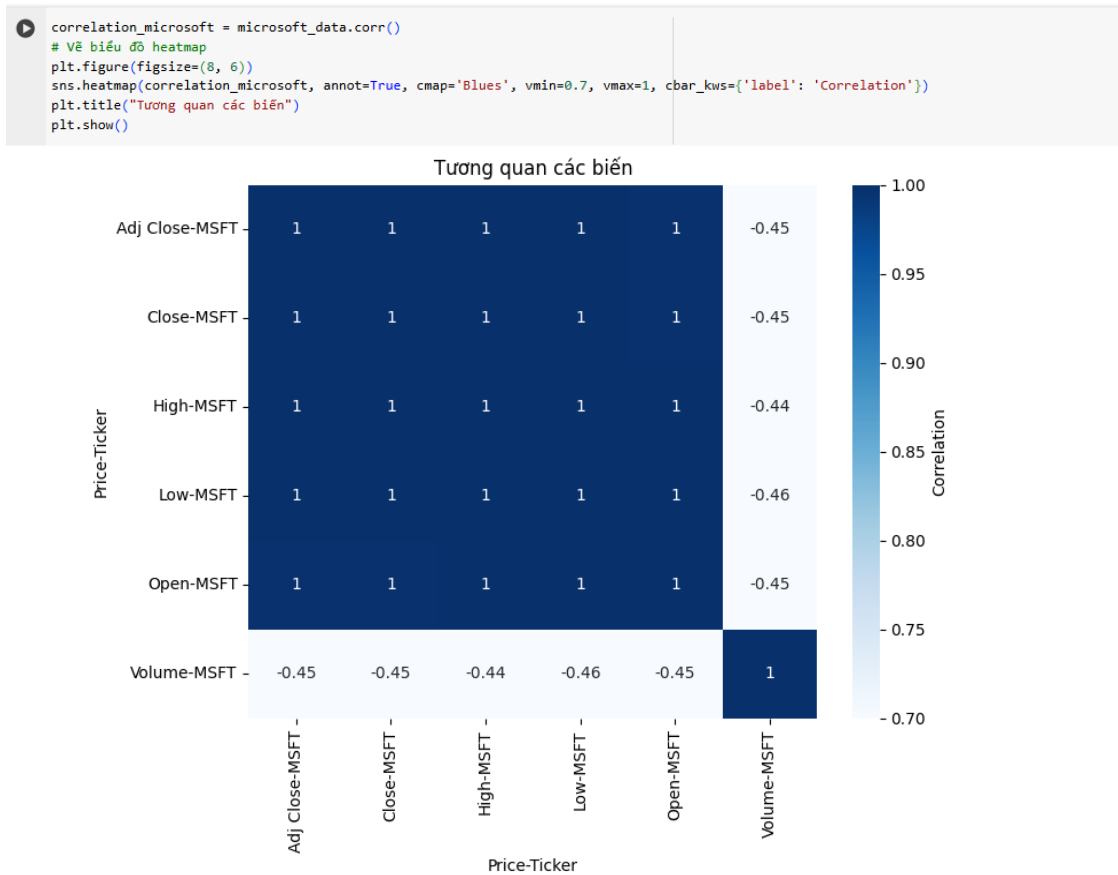
Nhìn chung, giá đóng cổ phiếu Microsoft thể hiện xu hướng tăng ổn định trong những năm gần đây. Ngược lại, lượng giao dịch lại giảm. Khi phân tích tương quan giữa

Hình 3. 13. Biểu đồ tỷ suất lợi nhuận hàng ngày trong năm 2024 của Microsoft

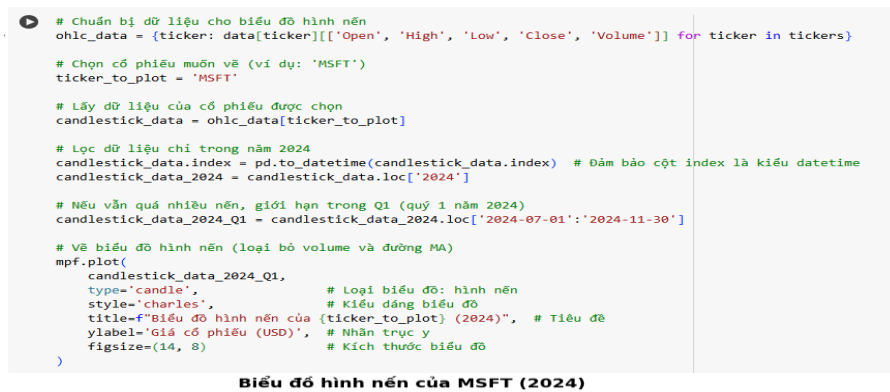
các biến, chỉ số tương quan giữa giá đóng (Close) và lượng giao dịch (Volume) là -0.45, cho thấy mối quan hệ nghịch chiều yếu giữa hai yếu tố này (xem hình 3.14). Có nghĩa rằng khi giá đóng tăng, lượng giao dịch có xu hướng giảm, nhưng mức độ ảnh hưởng không mạnh.

Về tỷ suất lợi nhuận trong năm 2024 dao động từ -6% đến 2%, tập trung quay mức 0% cho thấy sự ổn định trong biến động giá cổ phiếu.

Hình 3.15 là biểu đồ hình nến trong 5 tháng (từ tháng 7 đến tháng 11 năm 2024) của Microsoft. Qua biểu đồ ta có thể thấy từ tháng 7 đến giữa tháng 8 cho thấy biến động lớn với các cây nến dài và các cây nến đỏ chiếm ưu thế. Giai đoạn từ cuối tháng 8 đến tháng 11, biên độ dao động thu hẹp dần, cho thấy giá cổ phiếu ít biến động hơn.



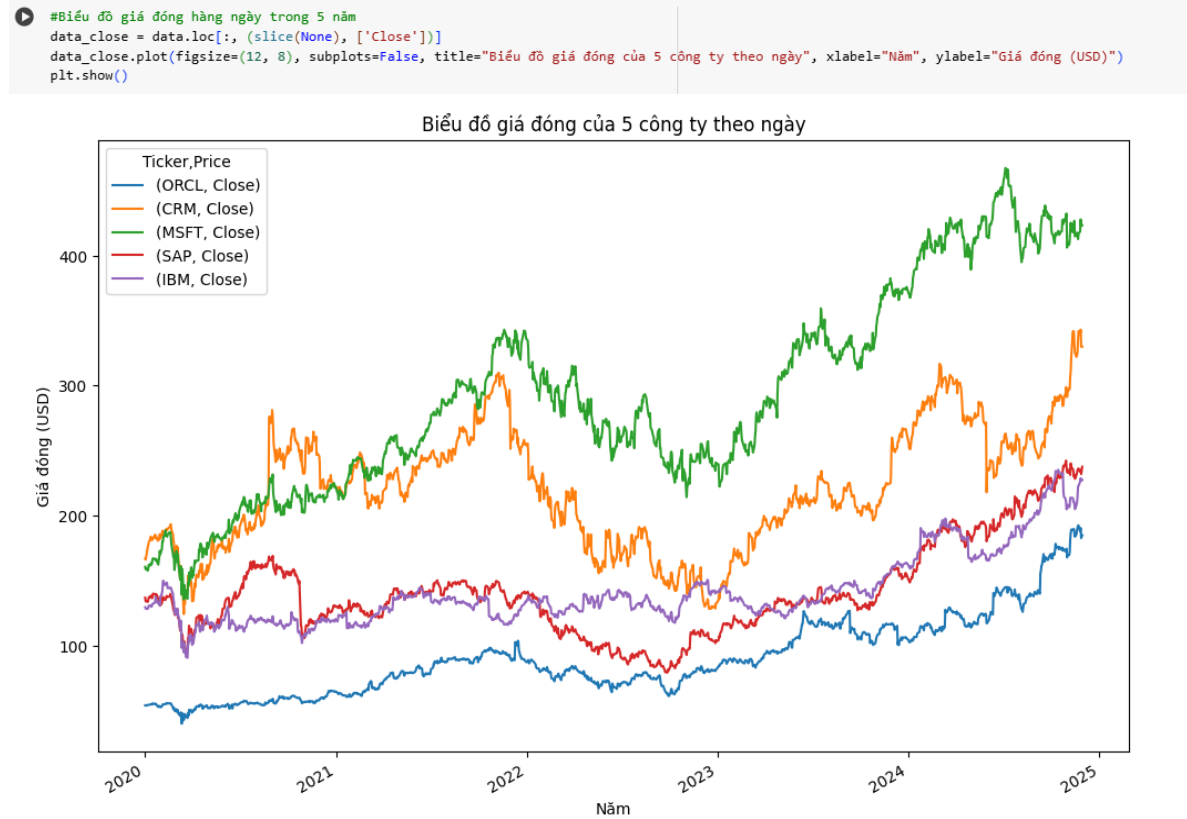
Hình 3. 144. Biểu đồ tương quan giữa các biến



Hình 3. 15. Biểu đồ hình nến về chứng khoán của Microsoft trong 5 tháng gần đây

3.3. So sánh các công ty

Về giá đóng, khi nhìn vào biểu đồ giá đóng của 5 công ty thì Microsoft là công ty có giá đóng cao nhất, thấp nhất là Oracle.



Hình 3. 15. Biểu đồ giá đóng của 5 công ty từ 2020 đến 11/2024

Thống kê trung bình giá đóng trong 5 năm của 5 công ty có thể thấy được năm 2020 Salesforce là công ty có giá đóng trung bình cao nhất, tuy nhiên trong 4 năm trở lại đây Microsoft lại dẫn đầu. IBM và SAP là hai công ty có giá gần nhau, và Oracle là công ty có trung bình giá đóng thấp nhất trong 5 công ty.



Hình 3. 16. Thống kê trung bình giá đóng theo năm của 5 công ty

```

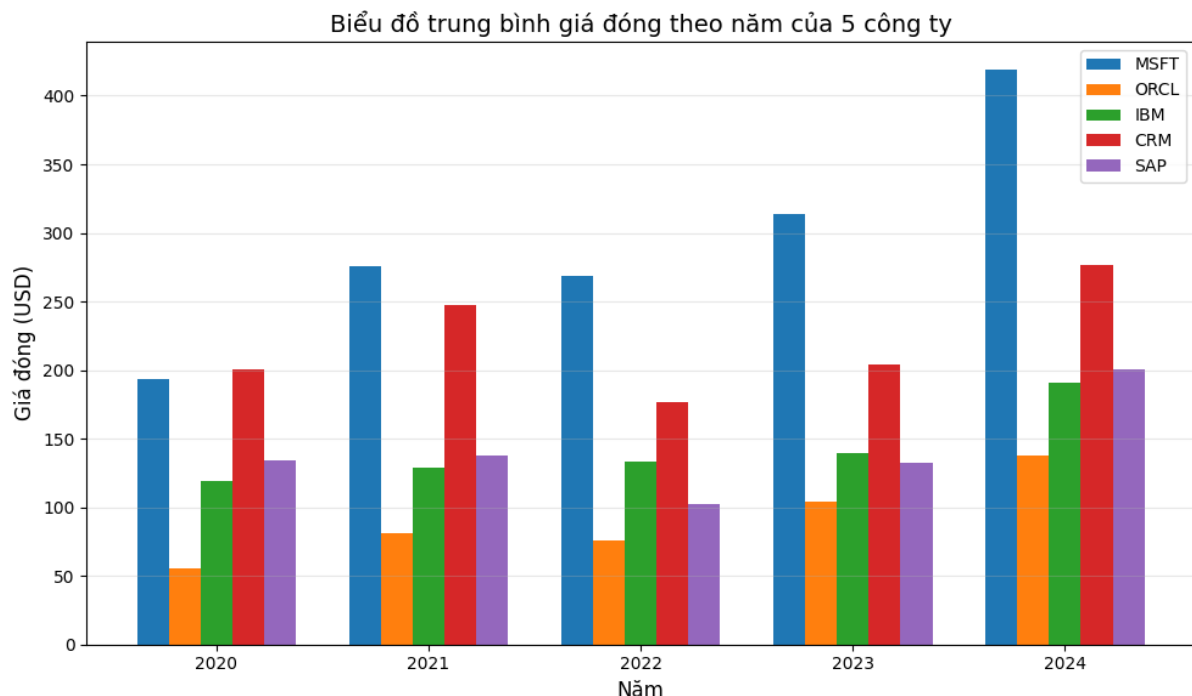
# Vẽ biểu đồ cột nhóm
plt.figure(figsize=(10, 6))

# Xác định số lượng năm
years = average_closer.index
bar_width = 0.15
x = range(len(years))

# Vẽ từng nhóm cột
for i, company in enumerate(average_closer.columns):
    plt.bar([pos + i * bar_width for pos in x], average_closer[company],
            width=bar_width, label=company)

# Thêm nhãn và tiêu đề
plt.xlabel('Năm', fontsize=12)
plt.ylabel('Giá đóng (USD)', fontsize=12)
plt.title('Biểu đồ trung bình giá đóng theo năm của 5 công ty', fontsize=14)
plt.xticks([pos + 2 * bar_width for pos in x], years)
plt.legend()
plt.tight_layout()
plt.grid(axis='y', alpha=0.3)
plt.show()

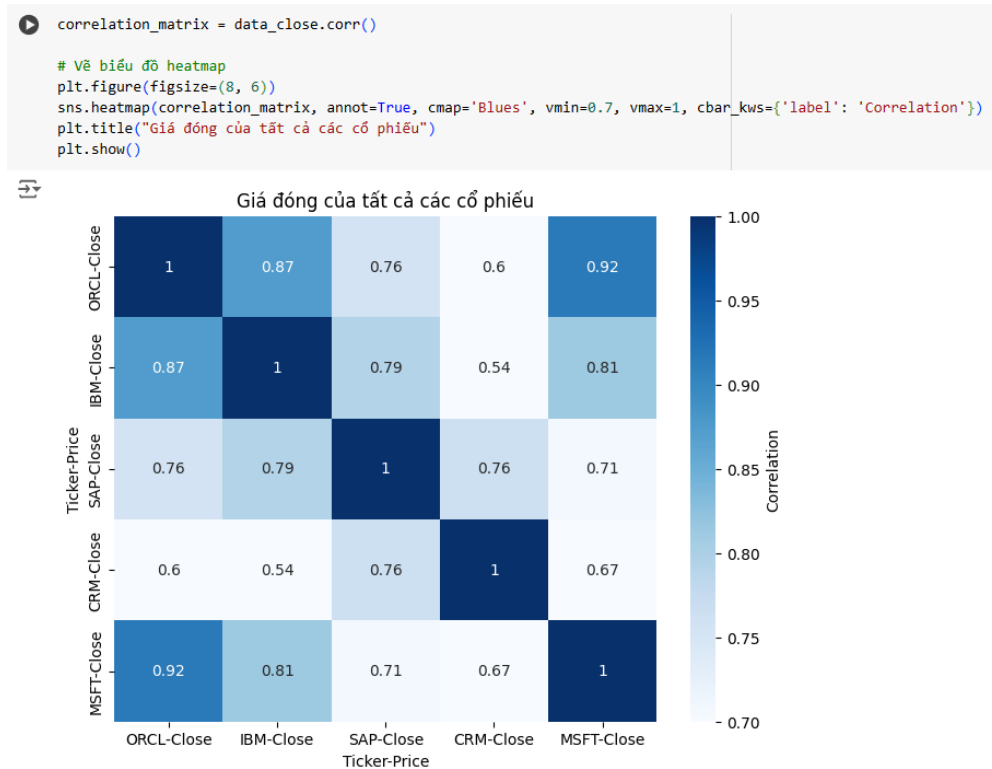
```



Hình 3. 17. Biểu đồ trung bình giá đóng theo năm của 5 công ty

Qua biểu đồ hình 3.17 ta có thể thấy được Microsoft là công ty dẫn đầu về giá cổ phiếu với mức tăng trưởng nhanh so với 4 công ty còn lại từ khoảng 200 USD đạt hơn 400 USD trong 5 năm. Trái lại, Oracle là công ty có sự tăng trưởng ổn định trong 5 năm qua. Đáng chú ý, năm 2022 ngoại trừ IBM có giá tăng nhẹ so với năm 2021 thì 4 công ty còn lại đều có giá trung bình giảm.

Phân tích tương quan giá đóng giữa các công ty, Microsoft và Oracle có tương quan cao nhất 0.92, cho thấy giá cổ phiếu của hai công ty này biến động cùng chiều với mức độ mạnh mẽ. Oracle và IBM cũng có chỉ số tương quan cao là 0.87, IBM và Microsoft có chỉ số tương quan đáng quan tâm là 0.81. Có thể thấy, ba công ty Microsoft, Oracle và IBM là 3 công ty có sự tương quan cao trong biến động giá cổ phiếu. SAP và Salesforce có tương quan với các công ty khác thấp hơn dao động từ 0.54 đến 0.79 thể hiện sự biến động cùng chiều nhưng tác động không quá mạnh (xem hình 3.18).

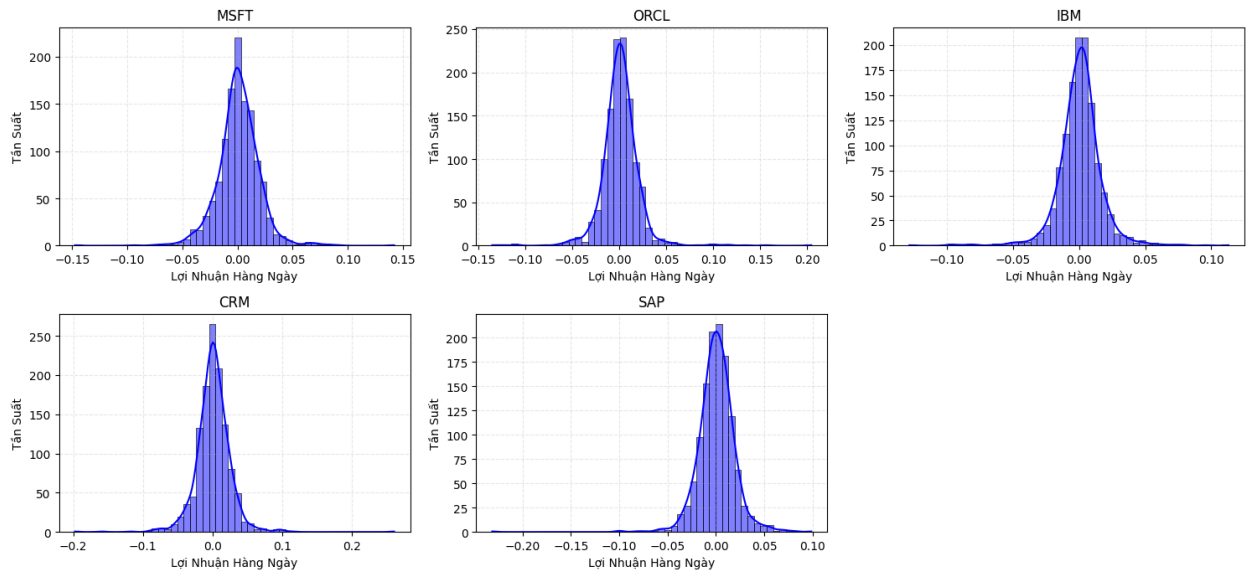


Hình 3. 18. Biểu đồ nhiệt tương quan giá đóng của 5 công ty

Về tỷ suất lợi nhuận theo ngày trong 5 năm qua, cả 5 công ty đều có phân phối lợi nhuận với dạng phân phối chuẩn đỉnh tập trung quay giá trị trung bình (xấp xỉ là 0%). Điều này cho thấy lợi nhuận hàng ngày của các công ty thường ổn định và không có sự biến động lớn. Cụ thể, Salesforce (CRM) có độ biến động cao nhất khi phạm vi lợi nhuận nằm trong khoảng từ -0.2 đến hơn 0.2.



Phân Phối Lợi Nhuận Hàng Ngày (2020-2024)

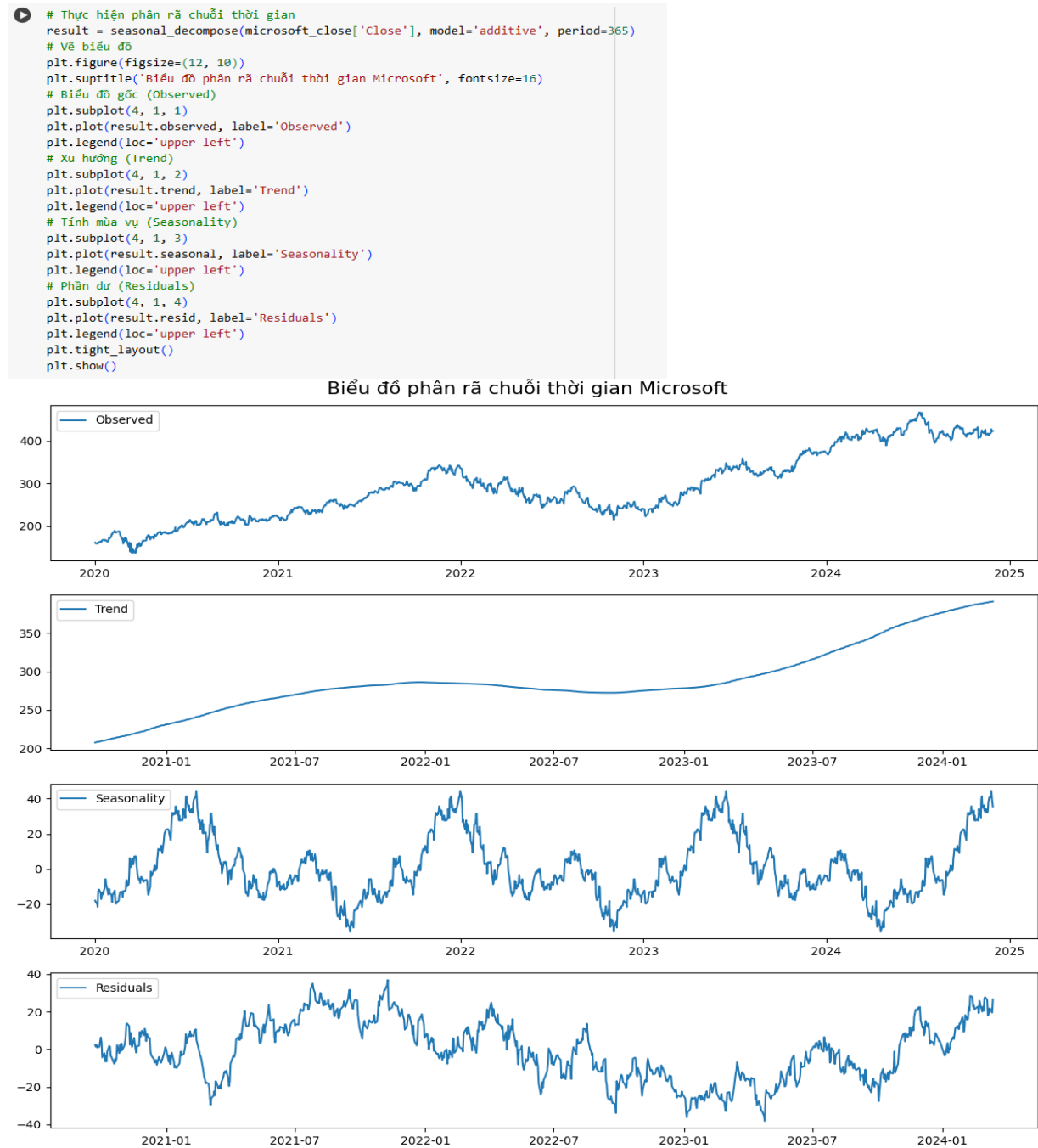


Hình 3. 19. Biểu đồ tỷ suất lợi nhuận theo ngày của Microsoft trong 5 năm (2020-2024)

Chương 4: Yếu tố thời gian và mô hình dự báo

4.1. Yếu tố thời gian

Trong phân tích dữ liệu chuỗi thời gian, xu hướng, mùa vụ và chu kỳ là 3 thành phần quan trọng giúp hiểu rõ cấu trúc của dữ liệu. Tính xu hướng thể hiện sự thay đổi dài hạn, có thể tăng, giảm hoặc giữ nguyên. Tính mùa vụ là các biến động định kỳ lặp lại trong khoảng thời gian cố định (ngày, tuần, tháng, quý, năm). Trong khi đó, chu kỳ là các dao động mang tính lặp lại nhưng không cố định về thời gian, bị ảnh hưởng bởi những yếu tố vĩ mô hơn như chu kỳ kinh tế, xu hướng thị trường và khó dự đoán hơn tính mùa vụ.



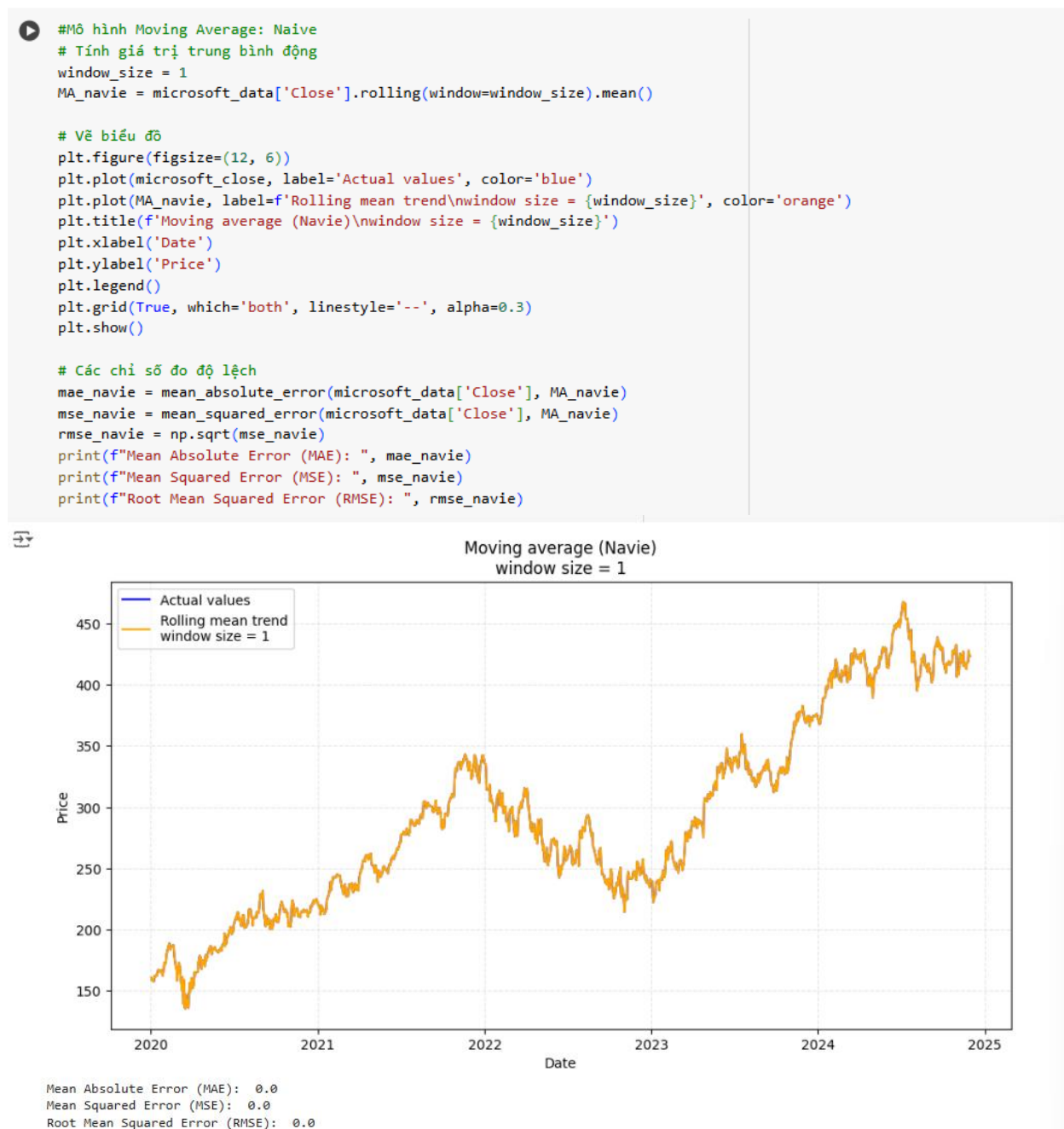
Hình 4. 1. Biểu đồ phân rã chuỗi thời gian Microsoft

Biểu đồ xu hướng (Trend) cho thấy xu hướng tăng trưởng của giá đóng, trong khi đó biểu đồ mùa vụ (Seasonality) cho thấy giá cổ phiếu thường tăng vào giữa năm và giảm vào cuối năm.

4.2. Mô hình dự báo Moving Average

4.2.1. Mô hình dự báo Navie

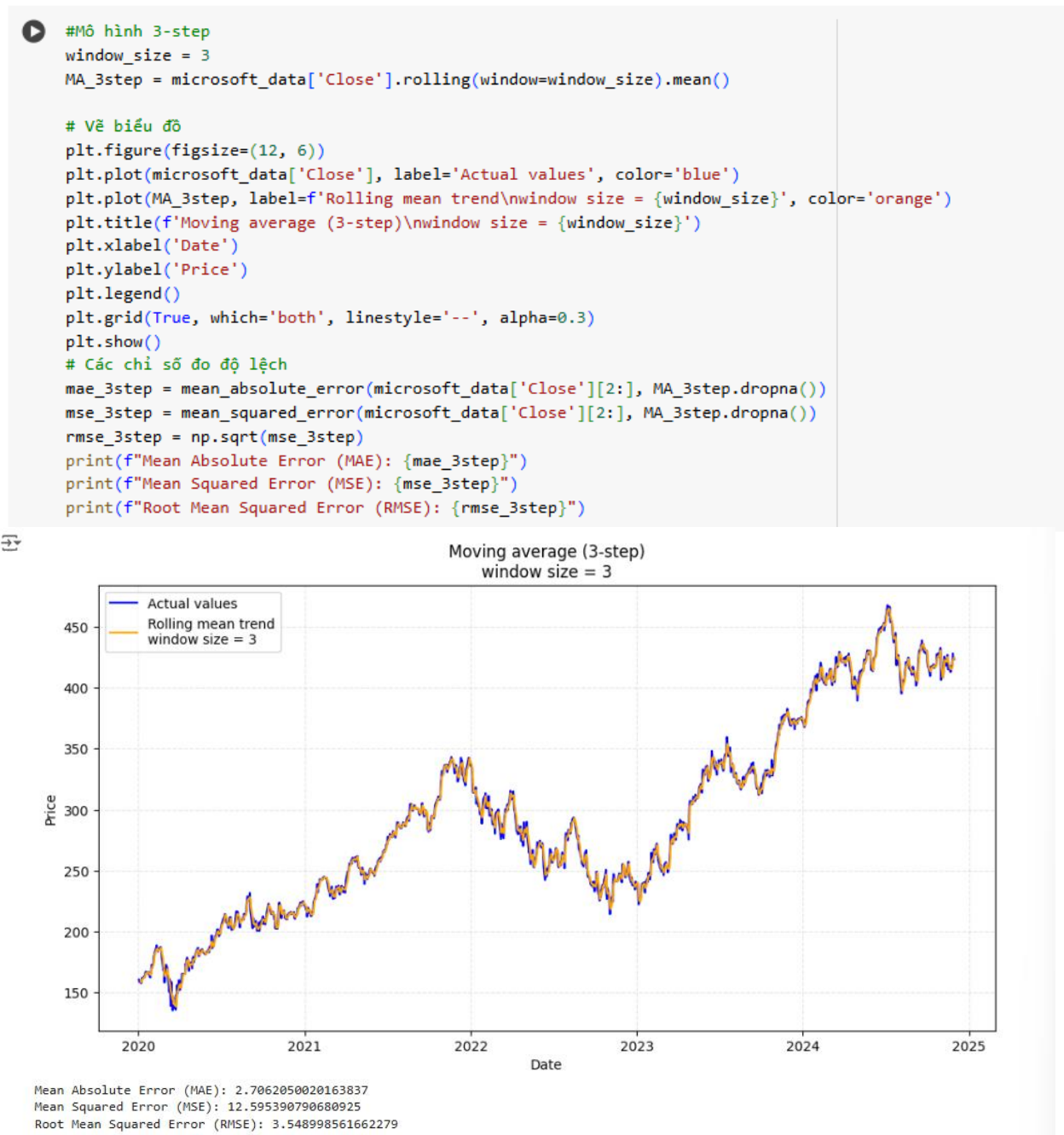
Với khoảng trượt (window size) là 1, mỗi điểm giá trị trung bình chính là giá trị gốc, do đó không có bất kỳ làm mịn nào xảy ra và đường dự đoán khớp hoàn toàn với dữ liệu thực. Mô hình Moving Average Naive sử dụng để tham khảo và so sánh với các mô hình khác nếu mô hình dự báo không tốt hơn Moving Average Naive thì mô hình đó chưa thực sự hiệu quả.



Hình 4. 2. Biểu đồ mô hình dự báo Moving average (Navie)

4.2.2. Mô hình Moving Average 3-step

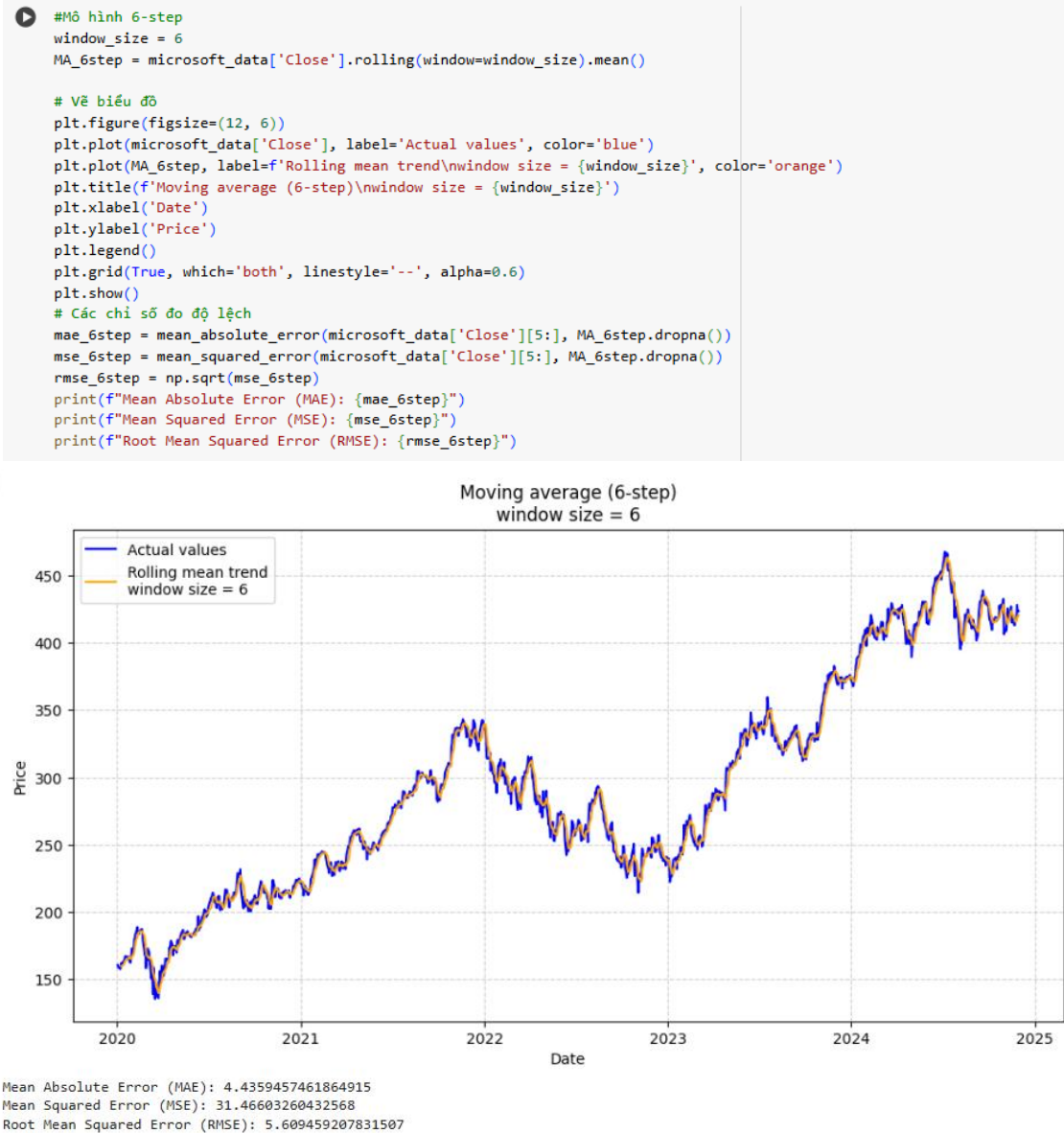
Với window size bằng 3, mô hình chỉ tập trung vào dữ liệu gần nhất (3 phiên), giúp giảm nhiễu ngắn hạn nhưng vẫn phản ứng nhanh với thay đổi mới.



Hình 4. 3. Biểu đồ mô hình dự báo Moving average (3-step)

4.2.3. Mô hình Moving Average 6-step

Với window size bằng 6, mô hình mịn hơn so với giá thực tế, giúp phát hiện xu hướng ngắn hạn rõ ràng hơn và dễ sử dụng trong thực tế.



Hình 4. 4. Biểu đồ mô hình Moving average (6-step)

4.3. Mô hình Exponential Smoothing

Mô hình Exponential Smoothing giúp làm mịn dữ liệu bằng cách giảm trọng số của các quan sát cũ và tập trung vào những quan sát gần đây. Trọng số được xác định bởi tham số α , α càng nhỏ đường càng mượt, α càng lớn đường càng nhạy cảm với thay đổi.

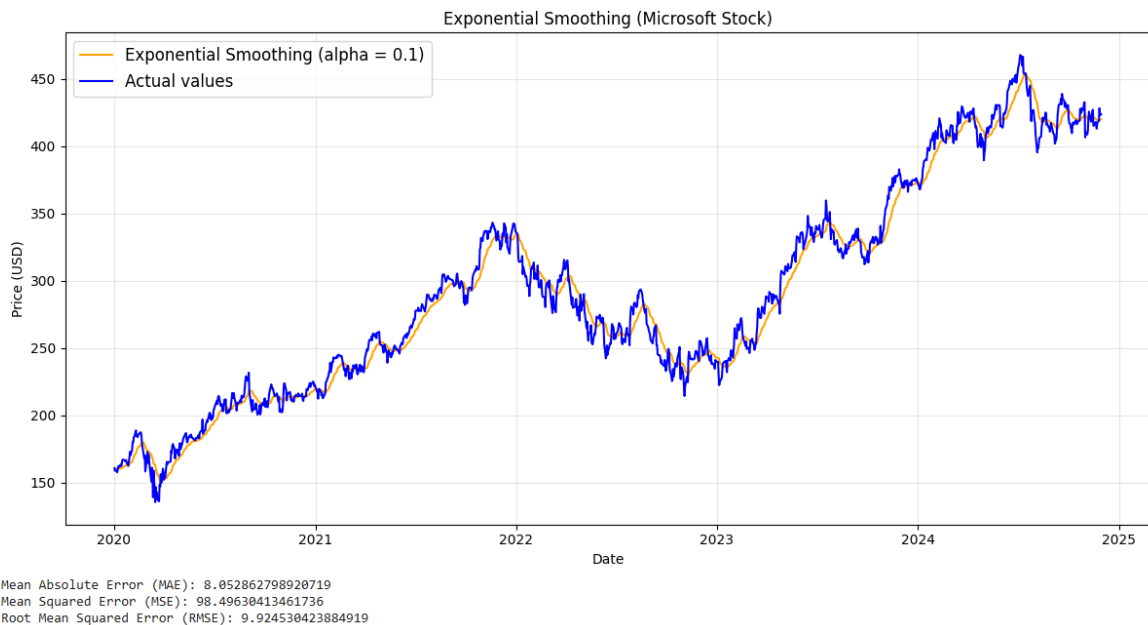
4.3.1. Hệ số $\alpha = 0.1$

Mô hình Exponential Smoothing hệ số $\alpha = 0.1$ làm mịn tốt và đường dự báo loại bỏ được phần lớn nhiễu ngắn hạn. Với α nhỏ, mô hình phản ứng chậm với biến động lớn hoặc thay đổi bất thường trong giá cổ phiếu.

```

#Mô hình Exponential Smoothing: hệ số alpha = 0.1
alpha = 0.1
model = SimpleExpSmoothing(microsoft_data['Close']).fit(smoothing_level=alpha, optimized=False)
SES = model.fittedvalues
# Vẽ biểu đồ
plt.figure(figsize=(12, 6))
plt.plot(SES, label=f'Exponential Smoothing (alpha = {alpha})', color='orange')
plt.plot(microsoft_data['Close'], label='Actual values', color='blue')
plt.title('Exponential Smoothing (Microsoft Stock)')
plt.xlabel('Date')
plt.ylabel('Price (USD)')
plt.legend(fontsize=12)
plt.grid(alpha=0.3)
plt.tight_layout()
plt.show()
# Các chỉ số đo độ lệch
mae_alpha01 = mean_absolute_error(microsoft_data['Close'], SES)
mse_alpha01 = mean_squared_error(microsoft_data['Close'], SES)
rmse_alpha01 = np.sqrt(mse_alpha01)
print(f"Mean Absolute Error (MAE): {mae_alpha01}")
print(f"Mean Squared Error (MSE): {mse_alpha01}")
print(f"Root Mean Squared Error (RMSE): {rmse_alpha01}")

```



Hình 4. 5. Biểu đồ mô hình dự báo Simple Exponential Smoothing ($\alpha = 0.1$)

4.3.2. Hệ số alpha tối ưu

Mô hình Exponential Smoothing với alpha tối ưu tối ưu hóa độ mịn, phản ứng hiệu quả với xu hướng của chuỗi thời gian nhưng vẫn duy trì độ ổn định, loại bỏ nhiễu ngẫu nhiên.

```

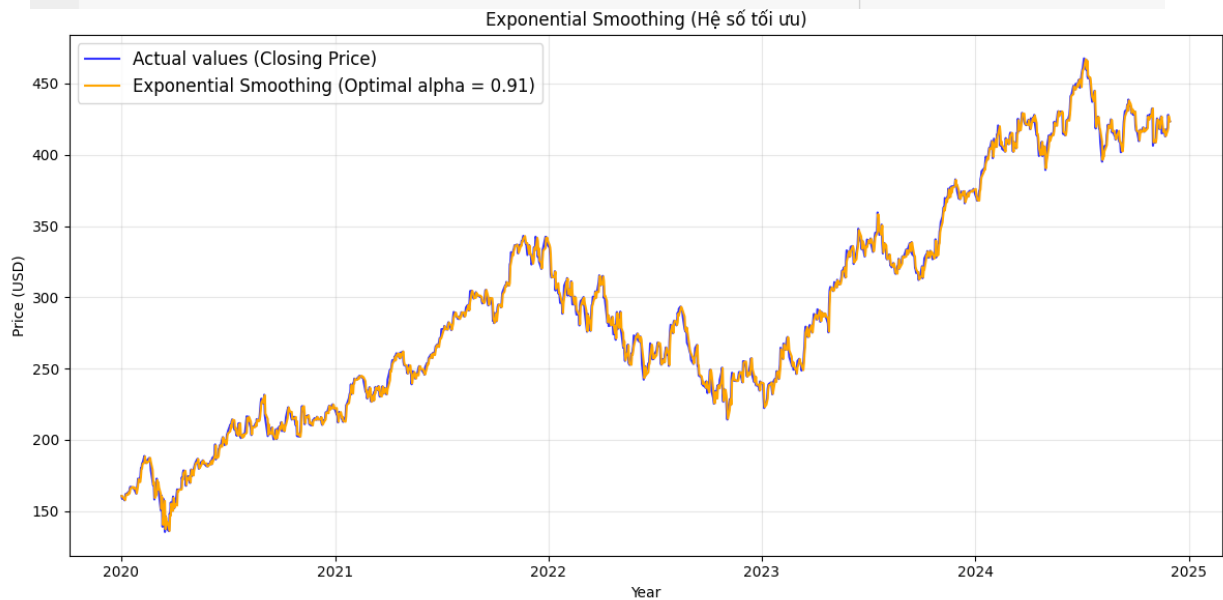
# Hàm tìm alpha tối ưu
def optimize_alpha(data, alpha_values):
    best_alpha = None
    best_mse = float('inf')
    for alpha in alpha_values:
        model = SimpleExpSmoothing(data).fit(smoothing_level=alpha, optimized=False)
        fitted_values = model.fittedvalues
        mse = mean_squared_error(data, fitted_values)
        if mse < best_mse:
            best_mse = mse
            best_alpha = alpha
    return best_alpha, best_mse

# Tập giá trị alpha để thử nghiệm
alpha_range = np.linspace(0.01, 1.0, 100)
optimal_alpha, optimal_mse = optimize_alpha(microsoft_data['Close'], alpha_range)
# Áp dụng mô hình với alpha tối ưu
model = SimpleExpSmoothing(microsoft_data['Close']).fit(smoothing_level=optimal_alpha, optimized=False)
SES_opt = model.fittedvalues

# Vẽ biểu đồ
plt.figure(figsize=(12, 6))
plt.plot(microsoft_data['Close'], label='Actual values (Closing Price)', color='blue', alpha=0.8)
plt.plot(SES_opt, label=f'Exponential Smoothing (Optimal alpha = {optimal_alpha:.2f})', color='orange')
plt.title('Exponential Smoothing (Hệ số tối ưu)')
plt.xlabel('Year')
plt.ylabel('Price (USD)')
plt.legend(fontsize=12)
plt.grid(alpha=0.3)
plt.tight_layout()
plt.show()

# Các chỉ số đo độ lệch
mae_SES_opt = mean_absolute_error(microsoft_data['Close'], SES_opt)
mse_SES_opt = mean_squared_error(microsoft_data['Close'], SES_opt)
rmse_SES_opt = np.sqrt(mse_SES_opt)
print(f"Mean Absolute Error (MAE): {mae_SES_opt}")
print(f"Mean Squared Error (MSE): {mse_SES_opt}")
print(f"Root Mean Squared Error (RMSE): {rmse_SES_opt}")

```



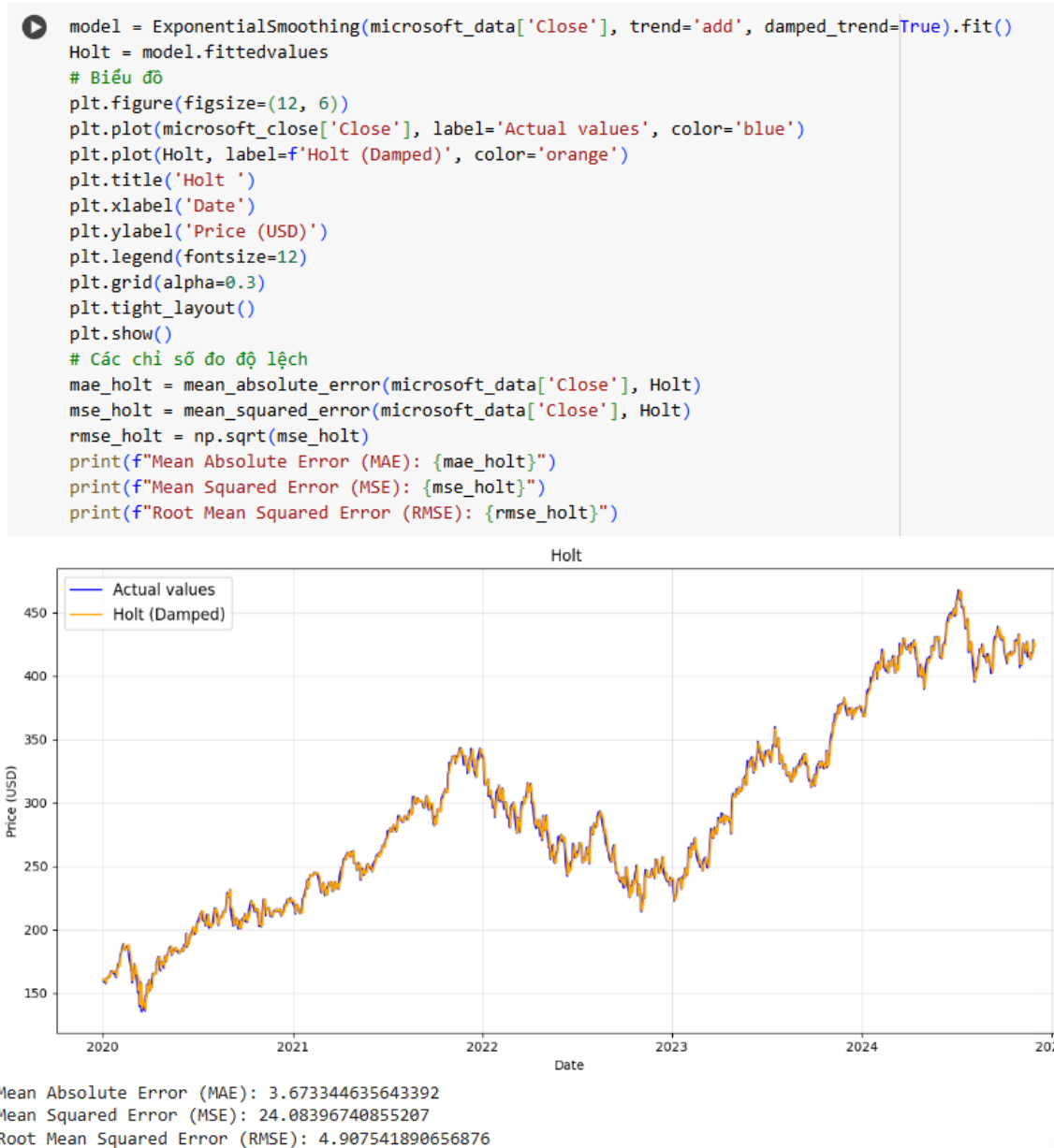
Mean Absolute Error (MAE): 3.6776228688399097
Mean Squared Error (MSE): 24.108419180366784
Root Mean Squared Error (RMSE): 4.91003250298476

Hình 4. 6. Biểu đồ mô hình dự báo Simple exponential smoothing với hệ số tối ưu.

4.4. Mô hình Holt

4.4.1. Mô hình Holt hệ số tiêu chuẩn

Mô hình Holt phù hợp để dự báo chuỗi thời gian có cả mức giá trị và xu hướng tuyến tính. Mô hình sử dụng hai hệ số làm trơn là alpha và beta, tương ứng với mức độ giá trị hiện tại và xu hướng.



Hình 4. 7. Biểu đồ mô hình dự báo Holt với hệ số tiêu chuẩn

4.4.2. Mô hình Holt hệ số tối ưu

Mô hình Holt với hệ số tối ưu là một phương pháp mở rộng của Mô hình Holt được cải tiến bằng cách tối ưu hóa các hệ số α và β để đạt được kết quả dự báo tốt nhất. Mô hình này đặc biệt hữu ích khi bạn muốn tự động điều chỉnh các tham số của mô hình để cải thiện độ chính xác của dự báo.

```
# Hàm tìm alpha và beta tối ưu
def optimize_holt(data, alpha_values, beta_values):
    best_alpha = None
    best_beta = None
    best_mse = float('inf')

    for alpha in alpha_values:
        for beta in beta_values:
            model = ExponentialSmoothing(data, trend='add', damped_trend=True).fit(
                smoothing_level=alpha, smoothing_trend=beta)
            fitted_values = model.fittedvalues
            mse = mean_squared_error(data, fitted_values)
            if mse < best_mse:
                best_mse = mse
                best_alpha = alpha
                best_beta = beta
    return best_alpha, best_beta, best_mse

# Tập giá trị alpha và beta để thử nghiệm
alpha_range = np.linspace(0.01, 1.0, 100)
beta_range = np.linspace(0.01, 1.0, 100)

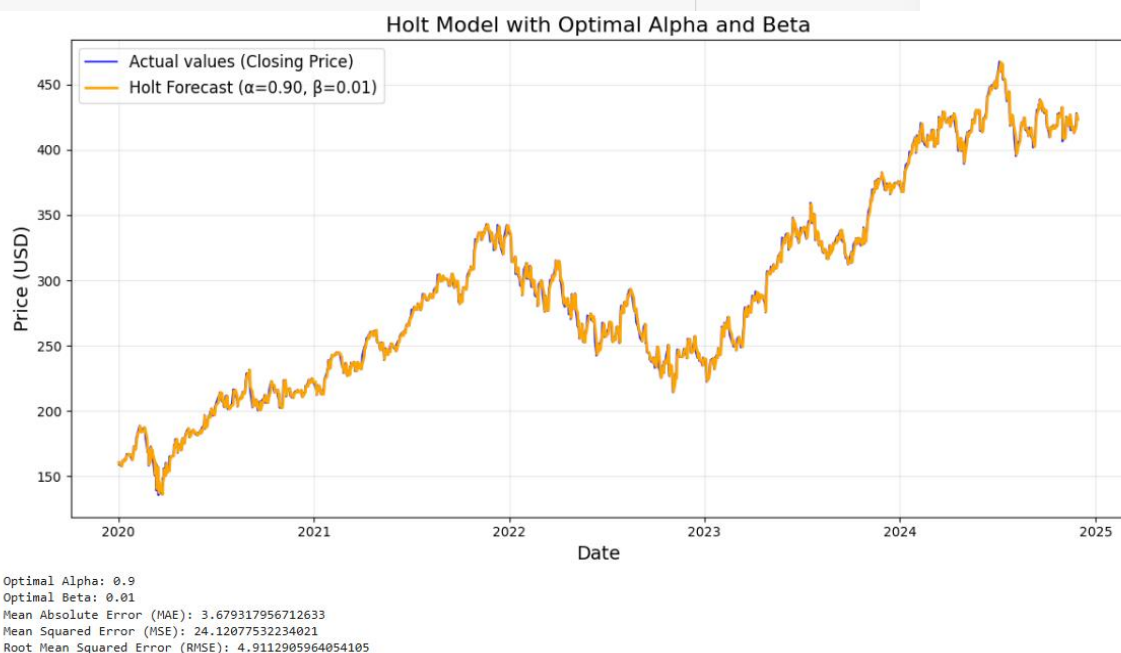
# Tìm alpha và beta tối ưu
optimal_alpha, optimal_beta, optimal_mse = optimize_holt(microsoft_data['Close'], alpha_range, beta_range)

# Áp dụng mô hình với alpha và beta tối ưu
final_model = ExponentialSmoothing(microsoft_data['Close'], trend='add', damped_trend=True).fit(
    smoothing_level=optimal_alpha, smoothing_trend=optimal_beta)
Holt_opt = final_model.fittedvalues

# Vẽ biểu đồ
plt.figure(figsize=(12, 6))
plt.plot(microsoft_data['Close'], label='Actual values (Closing Price)', color='blue', alpha=0.8)
plt.plot(Holt_opt, label=f'Holt Forecast (α={optimal_alpha:.2f}, β={optimal_beta:.2f})', color='orange')
plt.title('Holt Model with Optimal Alpha and Beta', fontsize=16)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Price (USD)', fontsize=14)
plt.legend(fontsize=12)
plt.grid(alpha=0.3)
plt.tight_layout()
plt.show()

# Tính các chỉ số đo lường lỗi
mae_holt_opt = mean_absolute_error(microsoft_data['Close'], Holt_opt)
mse_holt_opt = mean_squared_error(microsoft_data['Close'], Holt_opt)
rmse_holt_opt = np.sqrt(mse_holt_opt)

print(f"Optimal Alpha: {optimal_alpha}")
print(f"Optimal Beta: {optimal_beta}")
print(f"Mean Absolute Error (MAE): {mae_holt_opt}")
print(f"Mean Squared Error (MSE): {mse_holt_opt}")
print(f"Root Mean Squared Error (RMSE): {rmse_holt_opt}")
```

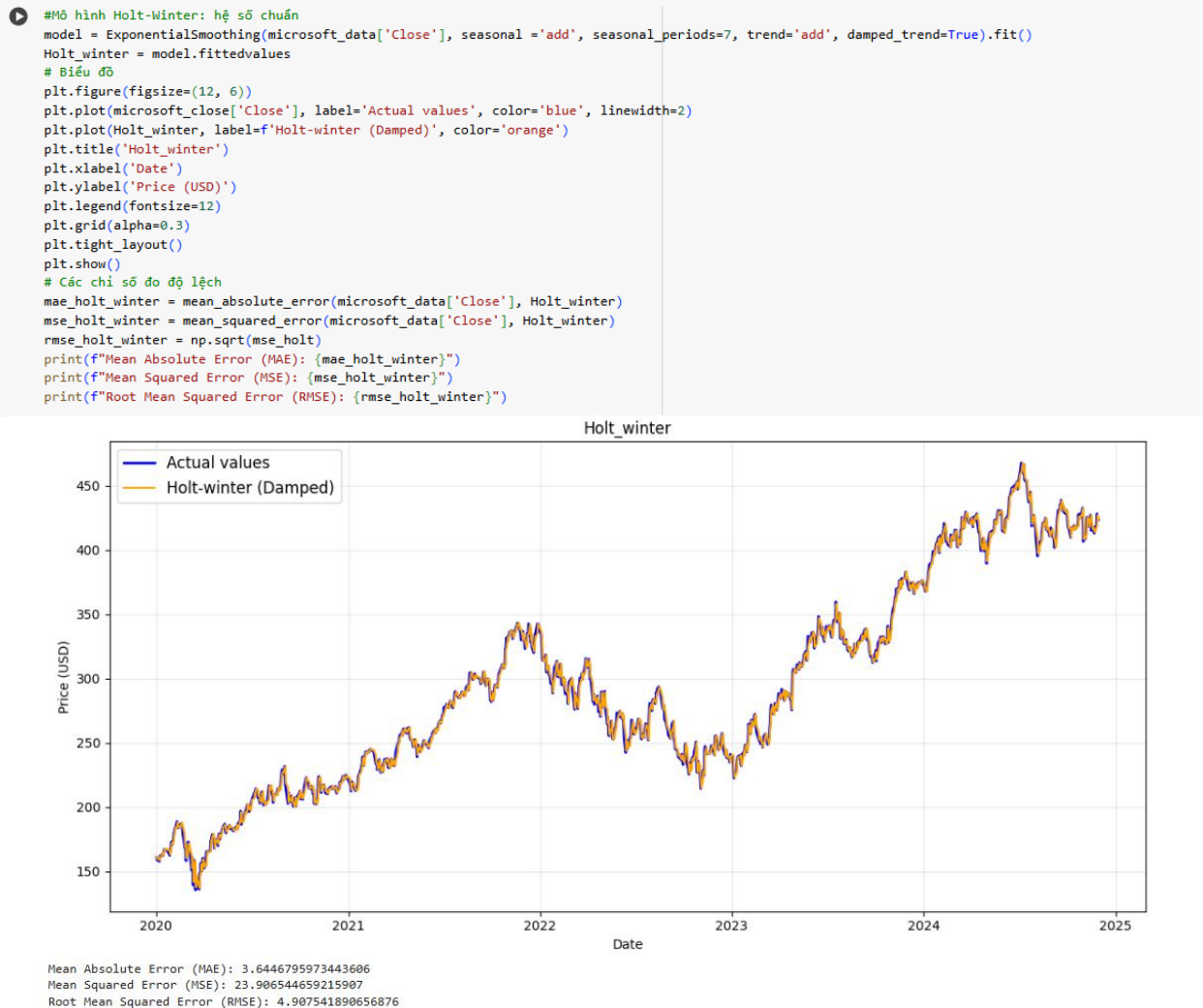


Hình 4. 8. Biểu đồ mô hình dự báo Holt với hệ số tối ưu

4.5. Mô hình Holt-winter

4.5.1. Mô hình Holt-winter với hệ số tiêu chuẩn

Mô hình Holt-Winter là phương pháp dự báo chuỗi thời gian khi dữ liệu có xu hướng (trend) và tính chất mùa vụ (seasonality). Mô hình Holt-Winter hệ số tiêu chuẩn sử dụng các giá trị mặc định hoặc gần đúng để áp dụng nhanh, không tối ưu hóa tham số dựa trên dữ liệu cụ thể.



Hình 4. 9. Biểu đồ mô hình dự báo Holt-winter với hệ số chuẩn

4.5.2. Mô hình Holt-winter với hệ tối ưu

Với phiên bản hệ số tối ưu, các tham số của mô hình được điều chỉnh để phù hợp nhất với dữ liệu thực tế, từ đó tăng độ chính xác của dự báo.

```

# Mô hình Holt-Winter: hệ số tối ưu
# Hàm tìm alpha, beta, gamma tối ưu
def optimize_holt_winters(data, alpha_values, beta_values, gamma_values, seasonal_periods):
    best_alpha = None
    best_beta = None
    best_gamma = None
    best_mse = float('inf')

    for alpha in alpha_values:
        for beta in beta_values:
            for gamma in gamma_values:
                try:
                    model = ExponentialSmoothing(
                        data,
                        trend='add',
                        seasonal='add',
                        seasonal_periods=seasonal_periods
                    ).fit(
                        smoothing_level=alpha,
                        smoothing_trend=beta,
                        smoothing_seasonal=gamma
                    )
                    fitted_values = model.fittedvalues
                    mse = mean_squared_error(data, fitted_values)
                    if mse < best_mse:
                        best_mse = mse
                        best_alpha = alpha
                        best_beta = beta
                        best_gamma = gamma
                except Exception as e:
                    # Bỏ qua các trường hợp không hợp lệ
                    pass

    return best_alpha, best_beta, best_gamma, best_mse

# Tập giá trị alpha, beta, gamma để thử nghiệm
alpha_range = np.linspace(0.01, 1.0, 10)
beta_range = np.linspace(0.01, 1.0, 10)
gamma_range = np.linspace(0.01, 1.0, 10)

# Tìm alpha, beta, gamma tối ưu
optimal_alpha, optimal_beta, optimal_gamma, optimal_mse = optimize_holt_winters(
    microsoft_close['Close'],
    alpha_range,
    beta_range,
    gamma_range,
    seasonal_periods=7
)

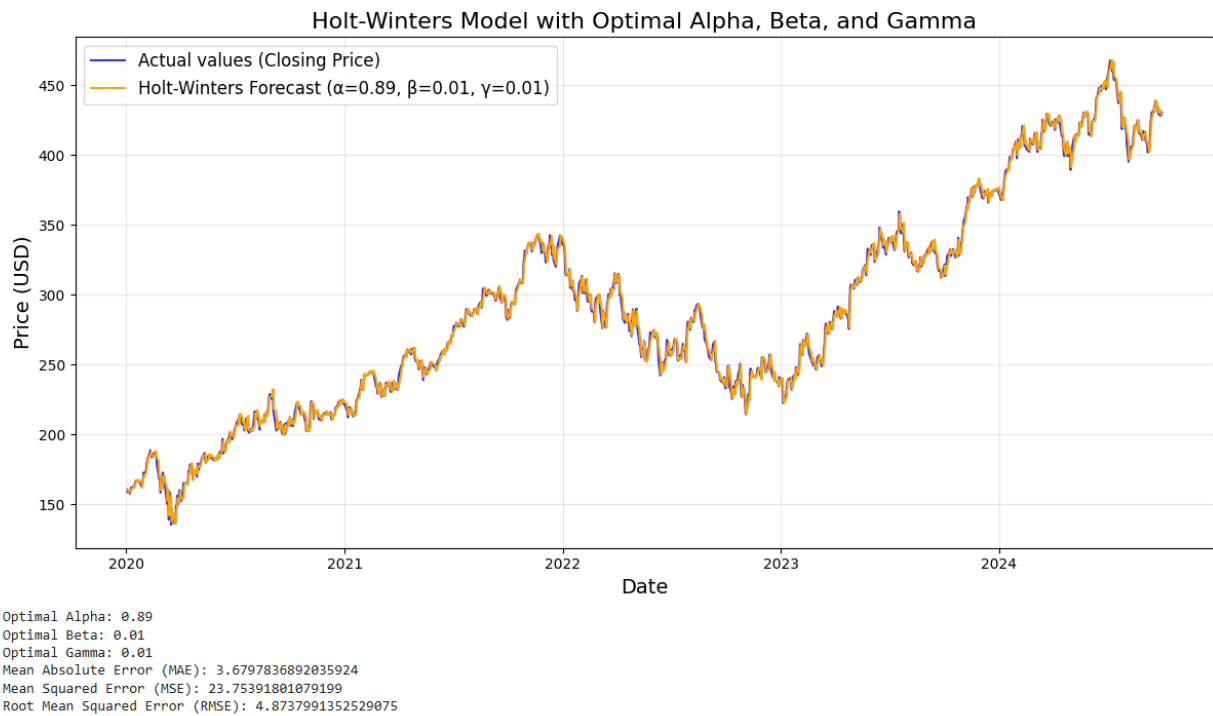
# Áp dụng mô hình với alpha, beta, gamma tối ưu
final_model = ExponentialSmoothing(
    microsoft_close['Close'],
    trend='add',
    seasonal='add',
    seasonal_periods=7
).fit(
    smoothing_level=optimal_alpha,
    smoothing_trend=optimal_beta,
    smoothing_seasonal=optimal_gamma
)
Holt_Winters_opt = final_model.fittedvalues

# Vẽ biểu đồ
plt.figure(figsize=(12, 6))
plt.plot(microsoft_close['Close'], label='Actual values (Closing Price)', color='blue', alpha=0.8)
plt.plot(Holt_Winters_opt, label=f'Holt-Winters Forecast (α={optimal_alpha:.2f}, β={optimal_beta:.2f}, γ={optimal_gamma:.2f})', color='orange')
plt.title('Holt-Winters Model with Optimal Alpha, Beta, and Gamma', fontsize=16)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Price (USD)', fontsize=14)
plt.legend(fontsize=12)
plt.grid(alpha=0.3)
plt.tight_layout()
plt.show()

# Tính các chỉ số đo lường lỗi
mae_holt_winter_opt = mean_absolute_error(microsoft_close['Close'], Holt_Winters_opt)
mse_holt_winter_opt = mean_squared_error(microsoft_close['Close'], Holt_Winters_opt)
rmse_holt_winter_opt = np.sqrt(mse_holt_winter_opt)

print(f"Optimal Alpha: {optimal_alpha}")
print(f"Optimal Beta: {optimal_beta}")
print(f"Optimal Gamma: {optimal_gamma}")
print(f"Mean Absolute Error (MAE): {mae_holt_winter_opt}")
print(f"Mean Squared Error (MSE): {mse_holt_winter_opt}")
print(f"Root Mean Squared Error (RMSE): {rmse_holt_winter_opt}")

```



Hình 4. 10. Biểu đồ mô hình dự báo Holt-winter với hệ số tối ưu

4.6. Thống kê các chỉ số đo độ lệch

Từ bảng thống kê đo lường các chỉ số lệch ta có thể rút ra được một số nhận xét như sau:

- Mô hình Moving Average: dự báo tốt trong thời gian ngắn.
- Mô hình Simple Exponential Smoothing: mô hình dự báo cho thời gian ngắn hạn tuy nhiên không tốt bằng mô hình Moving Average vì có chỉ số MSE cao hơn nhiều.
- Mô hình Holt và Holt-winter: Hai mô hình này phù hợp cho các dự báo dài hạn. Trong đó, Holt-winter dự báo tốt hơn mô hình Holt khi bổ sung các yếu tố mùa vụ, xu hướng.

```
models = ['MA_navie', 'MA_3step', 'MA_6step', 'SES', 'SES_opt', 'Holt', 'Holt_opt', 'Holt_winter', 'Holt_Winters_opt']
mae_scores = [mae_navie, mae_3step, mae_6step, mae_alpha01, mae_SES_opt, mae_holt, mae_holt_opt, mae_holt_winter, mae_holt_winter_opt]
mse_scores = [mse_navie, mse_3step, mse_6step, mse_alpha01, mse_SES_opt, mse_holt, mse_holt_opt, mse_holt_winter, mse_holt_winter_opt]
mape_scores = [rmse_navie, rmse_3step, rmse_6step, rmse_alpha01, rmse_SES_opt, rmse_holt, rmse_holt_opt, rmse_holt_winter, rmse_holt_winter_opt]

# Tạo bảng thống kê các chỉ số
evaluation = pd.DataFrame({'Model': models, 'MSE': mse_scores, 'MAE': mae_scores, 'MAPE': mape_scores})
evaluation.set_index('Model', inplace=True)
evaluation
```

Model	MSE	MAE	MAPE
MA_navie	0.000000	0.000000	0.000000
MA_3step	12.595391	2.706205	3.548999
MA_6step	31.466033	4.435946	5.609459
SES	98.496304	8.052863	9.924530
SES_opt	24.108419	3.677623	4.910033
Holt	24.083967	3.673345	4.907542
Holt_opt	24.120775	3.679318	4.911291
Holt_winter	23.906545	3.644680	4.889432
Holt Winters opt	23.753918	3.679784	4.873799

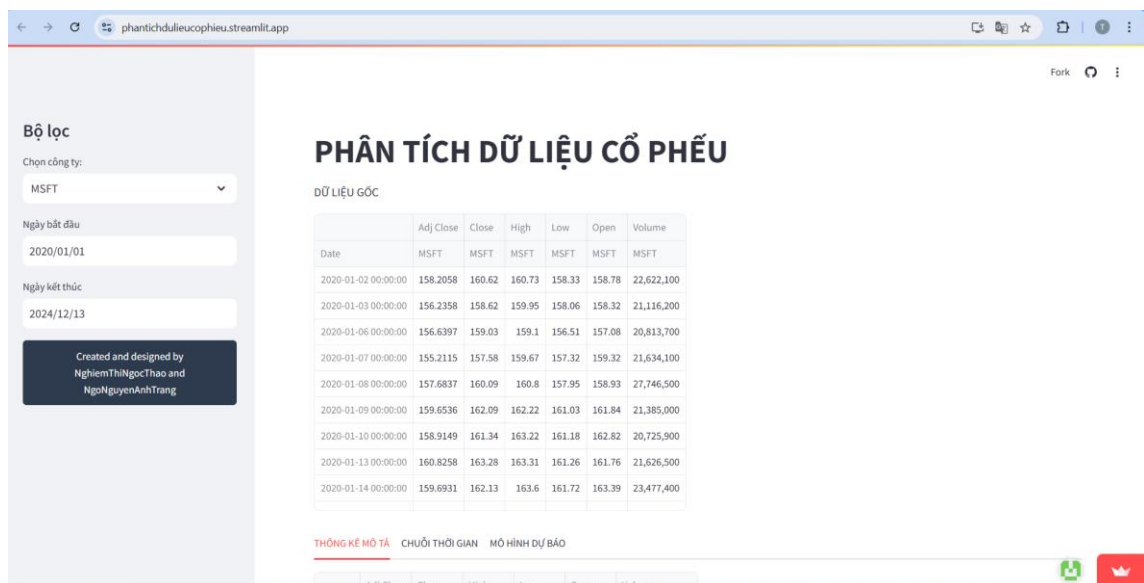
Hình 4. 11. Thống kê các chỉ số đo độ lệch

Chương 5: Thiết kế giao diện

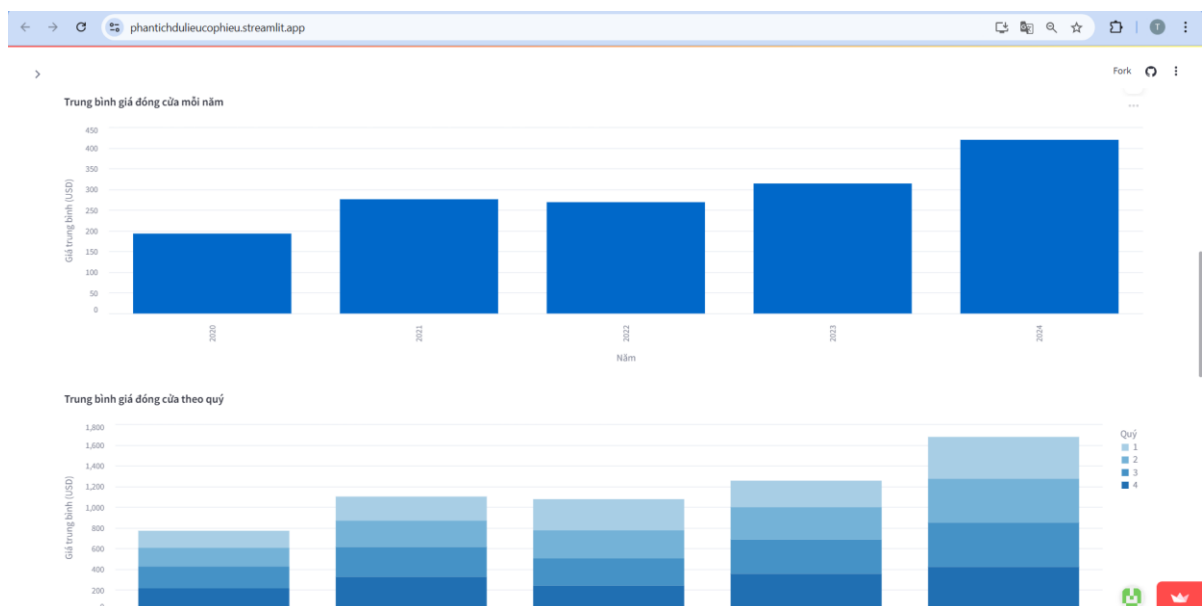
5.1. Mô tả tính năng

Trang web được xây dựng trên nền tảng Streamlit, cung cấp cho người dùng ba tính năng chính để hỗ trợ phân tích chứng khoán là thống kê mô tả, chuỗi thời gian và mô hình dự báo. Người dùng có thể dễ dàng chọn công ty trong danh sách gồm 5 công ty, sau đó lựa chọn khoảng thời gian cần phân tích. Bên cạnh đó, người dùng có thể tương tác với số liệu trên biểu đồ. Giao diện trực quan giúp người dùng tương tác linh hoạt với dữ liệu, từ việc xem các thống kê cơ bản đến việc dự báo xu hướng chứng khoán.

5.2. Giao diện



Hình 5. 1. Giao diện trang chủ



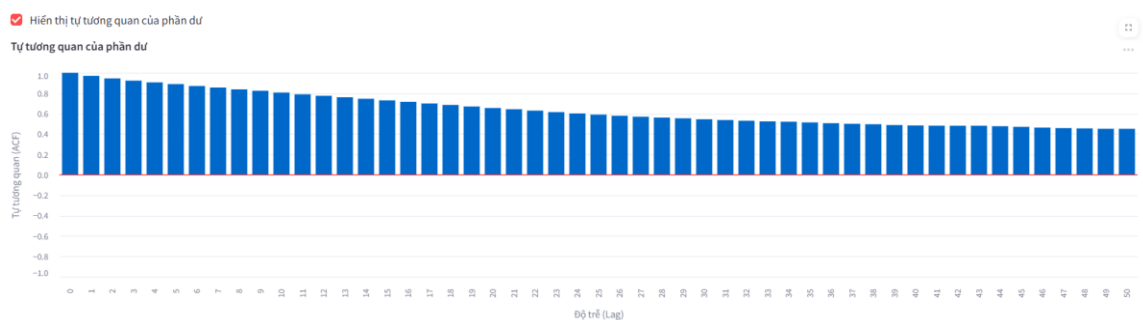
Hình 5. 2. Giao diện thống kê mô tả



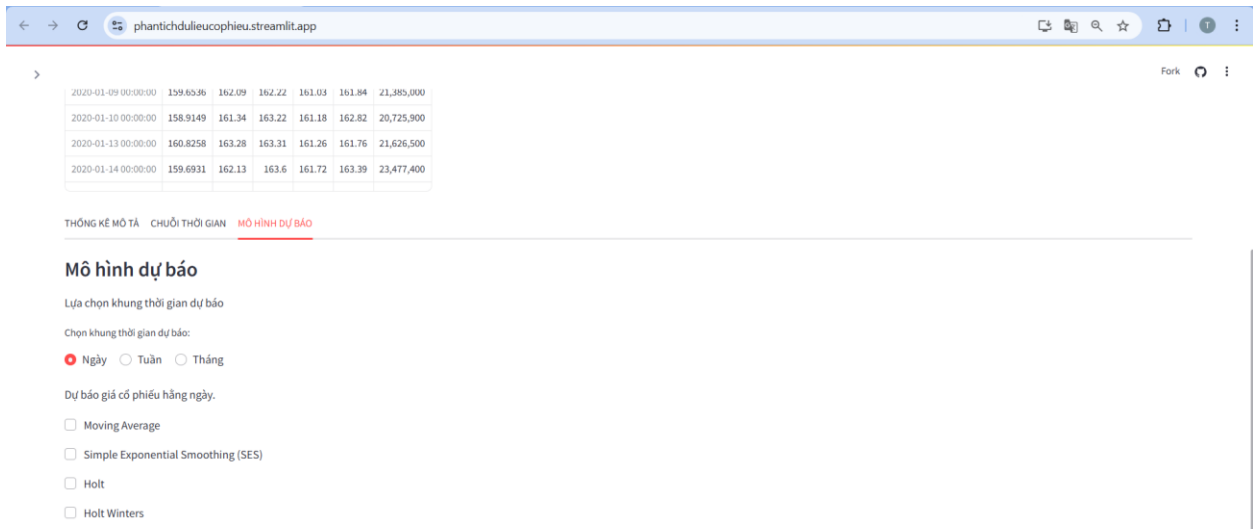
Hình 5. 4. Giao diện phân tích chuỗi thời gian



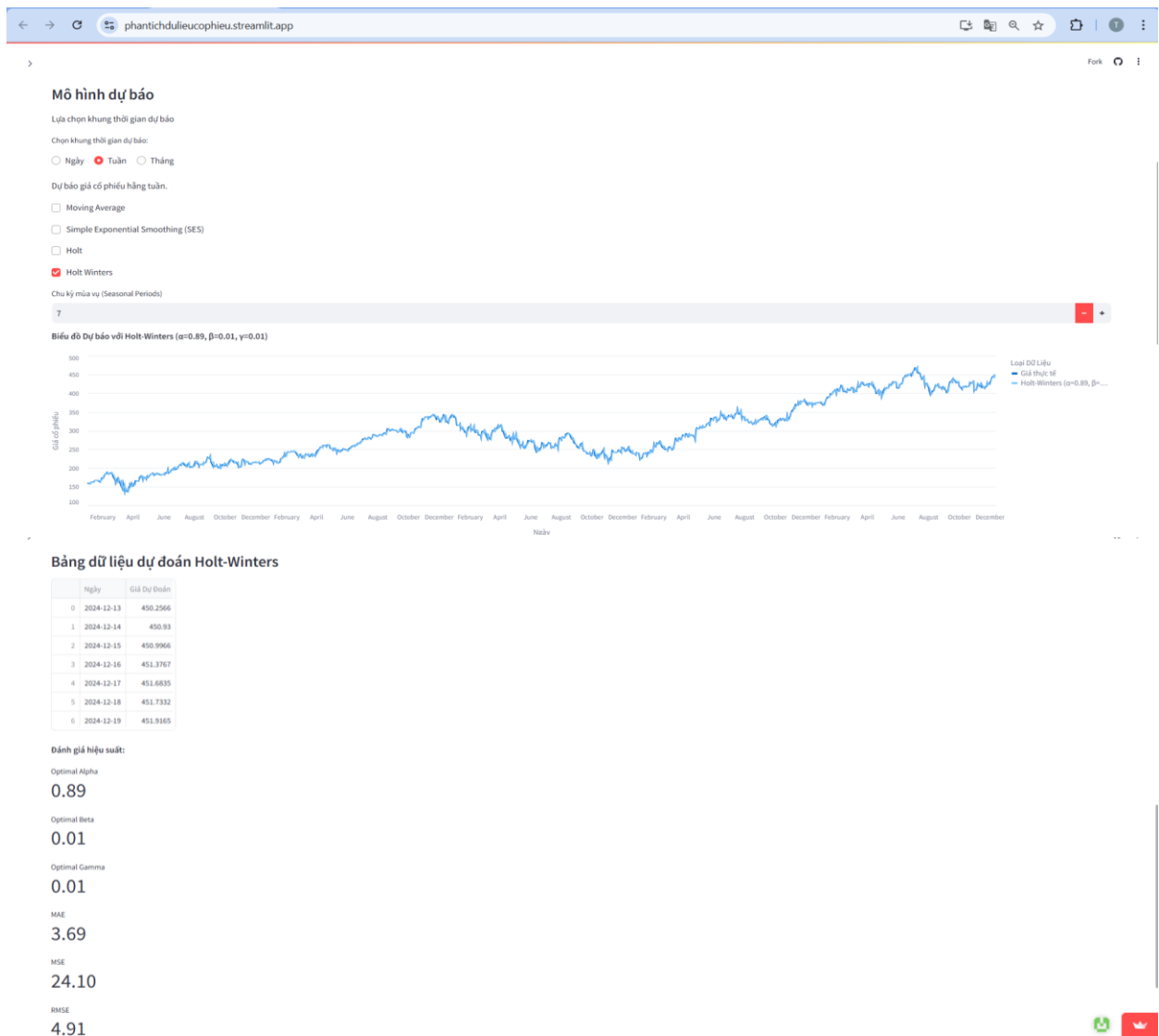
Hình 5. 3. Giao diện chức năng phân rã chuỗi thời gian



Hình 5. 5. Giao diện chức năng tương quan của phần dư



Hình 5. 6. Giao diện chức năng mô hình dự báo



Hình 5. 7. Giao diện khi dự báo theo mô hình mà người dùng lựa chọn

KẾT LUẬN VÀ ĐỀ XUẤT

Dự án đã thực hiện các nhiệm vụ chính, bao gồm thu thập và phân tích thống kê mô tả dữ liệu cổ phiếu của 5 công ty công nghệ lớn, trong đó tập trung sâu vào cổ phiếu của Microsoft. Các chỉ số như giá đóng, khối lượng giao dịch và tỷ suất lợi nhuận đã được phân tích một cách chi tiết. Ngoài ra, dự án đã triển khai các mô hình dự báo như Moving Average, Exponential Smoothing, Holt và Holt-Winter, đồng thời tối ưu hóa tham số để cải thiện độ chính xác.

Tuy nhiên, dự án vẫn tồn tại một số hạn chế. Việc đánh giá hiệu quả của các mô hình cũng chưa được so sánh chi tiết. Dữ liệu phân tích chủ yếu tập trung vào một số chỉ số cơ bản và thời điểm cụ thể, chưa mở rộng đến các yếu tố vĩ mô hoặc dữ liệu đa chiều.

Dựa vào những mô hình mà nhóm đã sử dụng để dự báo, đề xuất phát triển thêm các mô hình dự báo như ARIMA (AutoRegressive Integrated Moving Average) hay LSTM (Long short-term memory). Ngoài ra, giao diện website nên được phát triển thêm các tính năng như so sánh hiệu suất mô hình và cho phép nhập dữ liệu bên ngoài, giúp ứng dụng thực tiễn hơn và hỗ trợ người dùng hiệu quả hơn.

TÀI LIỆU THAM KHẢO

pandas - Python Data Analysis Library. (n.d.). Retrieved December 14, 2024, from <https://pandas.pydata.org/>

Returns and Log Returns. (n.d.). Retrieved December 14, 2024, from <https://gregorygundersen.com/blog/2022/02/06/log-returns/>

Time Series analysis tsa — statsmodels 0.9.0 documentation. (n.d.). Retrieved December 14, 2024, from <https://www.statsmodels.org/0.9.0/tsa.html>

What is NumPy? — NumPy v2.1 Manual. (n.d.). Retrieved December 14, 2024, from <https://numpy.org/doc/stable/user/whatisnumpy.html>

PHỤ LỤC

Link Github dự án: <https://github.com/trangtrang2508/PHANTICHDULIEUCOPHIEU>

Link website dự án: <https://phantichdulieucophieu.streamlit.app/>