

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

-----<>-----



ĐỒ ÁN CUỐI KỲ

PHÂN TÍCH VÀ DỰ ĐOÁN GIÁ CỎ PHIẾU BẰNG MÔ HÌNH DỰ BÁO

Môn: Phân tích dữ liệu cho quản lý

Giảng viên : Trần Đình Anh Huy

Nhóm 1 – Lớp Quản lý thông tin B

STT	Họ và tên	Mã số sinh viên
1	Hoàng Xuân Quốc	2156210125
2	Đặng Hoàng Chiến	2156210095
3	Nguyễn Viết Đức	2156210100



MỤC LỤC

DANH MỤC BẢNG	4
DANH MỤC HÌNH	5
I. TỔNG QUAN.....	7
1. Giới thiệu	7
2. Sơ đồ phân rã chức năng	7
3. Công cụ thực hiện	9
3.1 Python và thư viện sử dụng	9
3.1.1 Giới thiệu sơ lược về python.....	9
3.1.2 Những thư viện được sử dụng.....	9
3.2 Những mô hình sử dụng để dự báo	12
3.3 Giới thiệu về dữ liệu	17
4. Thông tin nhóm và phân chia nhiệm vụ.....	17
II. QUÁ TRÌNH THỰC HIỆN VÀ KẾT QUẢ	18
1. Tổng quan dữ liệu.....	18
1.1 Import thư viện	18
1.2 Tải dataset.....	19
1.3 Mô tả dữ liệu.....	19
2. Tiền xử lý dữ liệu	20
3. Thống kê mô tả	21
3.1 Các vấn đề thống kê.....	21
3.2 Kết luận các vấn đề thống kê.....	32
3.2.1 Về các thông tin cơ bản của dữ liệu	32
3.2.2 Về sự tương quan, khác biệt giữa các biến.....	32
3.2.3 Về cổ phiếu năm gần nhất (2023)	33
3.2.4 Về phân phối dữ liệu.....	33
4. Phân tích chuỗi thời gian	33
4.1 Xác định tính mùa vụ, xu hướng và chu kỳ	33
4.2 So sánh các mã cổ phiếu	41
5. Mô hình.....	46



5.1	Moving average naive	46
5.2	Moving average khoảng trượt 3	47
5.3	Moving average khoảng trượt 6	48
5.4	Simple Exponential Smoothing với alpha=0.1	49
5.5	Simple Exponential Smoothing với alpha tối ưu.....	50
5.6	Holt với hệ số chuẩn	51
5.7	Holt với hệ số tối ưu	53
5.8	Holt winter với hệ số chuẩn.....	55
5.9	Holt winter với hệ số tối ưu	56
6.	Đánh giá và so sánh mô hình	58
III.	THIẾT KẾ GIAO DIỆN DỰ BÁO	59
1.	Giới thiệu công cụ và trang web dự báo.....	59
2.	Sơ đồ lớp (class diagram).....	60
3.	Kết quả.....	62
3.1	Trang chính – trực quan.....	62
3.2	Trang thống kê mô tả	63
3.3	Trang phân tách time series	64
3.4	Trang predict.....	64
IV.	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	65
1.	Kết luận	65
2.	Hướng phát triển	65
V.	TÀI LIỆU THAM KHẢO.....	66
VI.	PHỤ LỤC – CÁC LIÊN KẾT DỰ ÁN GITHUB	66
	Liên kết đến dự án github	66
	Liên kết đến trang web dự án.....	66



DANH MỤC BẢNG

Bảng 1. 1 Thống kê các thư viện sử dụng	12
Bảng 1. 2 Phân công nhiệm vụ	18
Bảng 3. 1 Các class và chức năng xây dựng trang web	60
Bảng 3. 2 Chi tiết main_class	61
Bảng 3. 3 Chi tiết descriptive_statistics	61
Bảng 3. 4 Chi tiết time_series.....	62
Bảng 3. 5 Chi tiết train_models	62



DANH MỤC HÌNH

Hình 1. 1 Sơ đồ phân rã chức năng	8
Hình 1. 2 Cấu trúc thư mục dự án Github	18
Hình 2. 1 Import thư viện	19
Hình 2. 2 Tải dataset.....	19
Hình 2. 3 Mô tả kiểu dữ liệu từng cột	19
Hình 2. 4 Xem 10 dòng đầu tiên của dữ liệu.....	20
Hình 2. 5 Thay đổi tên các cột.....	20
Hình 2. 6 Kiểm tra dữ liệu trống, dữ liệu thiếu	20
Hình 2. 7 Kiểm tra dữ liệu trùng lặp	20
Hình 2. 8 Chuyển dữ liệu cho cột date	21
Hình 2. 9 In ra dữ liệu hiện có.....	21
Hình 2. 10 Thông tin ngày bắt đầu,kết thúc,tổng số ngày	21
Hình 2. 11 Một số thông tin cơ bản về dữ liệu	22
Hình 2. 12 Nhóm dữ liệu theo tháng	22
Hình 2. 13 Biểu đồ trung bình giá giao dịch theo tháng	23
Hình 2. 14 Biểu đồ trung bình giá giao dịch theo quý	24
Hình 2. 15 Biểu đồ trung bình giá giao dịch theo năm	25
Hình 2. 16 Biểu đồ Heatmap	25
Hình 2. 17 Biểu đồ Pairplot mối tương quan các biến	26
Hình 2. 18 Tạo closedf	27
Hình 2. 19 Histogram	27
Hình 2. 20 Tạo close_stock_2023	28
Hình 2. 21 Trung bình close price theo tháng 2023	29
Hình 2. 22 Trung bình close price theo quý	30
Hình 2. 23 Lấy closedf mới	30
Hình 2. 24 Trung bình giá đóng cửa theo tháng các năm 2021,2022,2023	31
Hình 2. 25 Trung bình close price theo quý các năm 2021,2022,2023	31
Hình 2. 26 Biểu đồ so sánh tỷ suất lợi nhuận 2021,2022,2023	32
Hình 2. 27 Dataframe closedfcopy	33
Hình 2. 28 Đặt cột date làm index	34
Hình 2. 29 Seasonal decompose 2023	35
Hình 2. 30 Biểu đồ giá đóng cửa theo năm	36
Hình 2. 31 Seasonal Decompose các năm	37
Hình 2. 32 Seasonal Decompose có yếu tố mùa vụ các năm	39
Hình 2. 33 Autocorrelation	40
Hình 2. 34 Tải dữ liệu 5 mã cổ phiếu	41
Hình 2. 35 Kiểm tra dữ liệu từ 5 mã cổ phiếu	41



Hình 2. 36 Biểu đồ đường giá cổ phiếu 5 hãng ô tô theo năm	42
Hình 2. 37 Biểu đồ số lượng cổ phiếu được giao dịch theo từng ngày	43
Hình 2. 38 Biểu đồ số tiền giao dịch theo từng ngày của 5 mã cổ phiếu	44
Hình 2. 39 Biểu đồ lợi nhuận tích lũy của 5 mã cổ phiếu	45
Hình 2. 40 Tạo dataframe mới là closedf	46
Hình 2. 41 Moving average naive	46
Hình 2. 42 Moving average khoảng trượt 3	47
Hình 2. 43 Moving average khoảng trượt 6	48
Hình 2. 44 Simple Exponential alpha 0.1	49
Hình 2. 45 Hàm tối ưu SES alpha tối ưu	50
Hình 2. 46 Chạy mô hình SES alpha tối ưu	50
Hình 2. 47 Biểu đồ SES alpha tối ưu	51
Hình 2. 48 Biểu đồ Holt hệ số chuẩn	52
Hình 2. 49 Hàm tối ưu Holt	53
Hình 2. 50 Chạy mô hình Holt hệ số tối ưu	53
Hình 2. 51 Biểu đồ Holt hệ số tối ưu	54
Hình 2. 52 Xây dựng mô hình holt winter hệ số chuẩn	55
Hình 2. 53 Biểu đồ Holt winter hệ số chuẩn	56
Hình 2. 54 Xây dựng hàm tối ưu holt winter	56
Hình 2. 55 Chạy mô hình holt winter	57
Hình 2. 56 Biểu đồ holt winter hệ số tối ưu	58
Hình 2. 57 Dataframe kết quả từ các mô hình	59
Hình 3. 1 Sơ đồ lớp giao diện trang web	60
Hình 3. 2 Trang chính – trực quan	63
Hình 3. 3 Trang thống kê mô tả	63
Hình 3. 4 Trang phân tách time series	64
Hình 3. 5 Trang predict	64



I. TỔNG QUAN

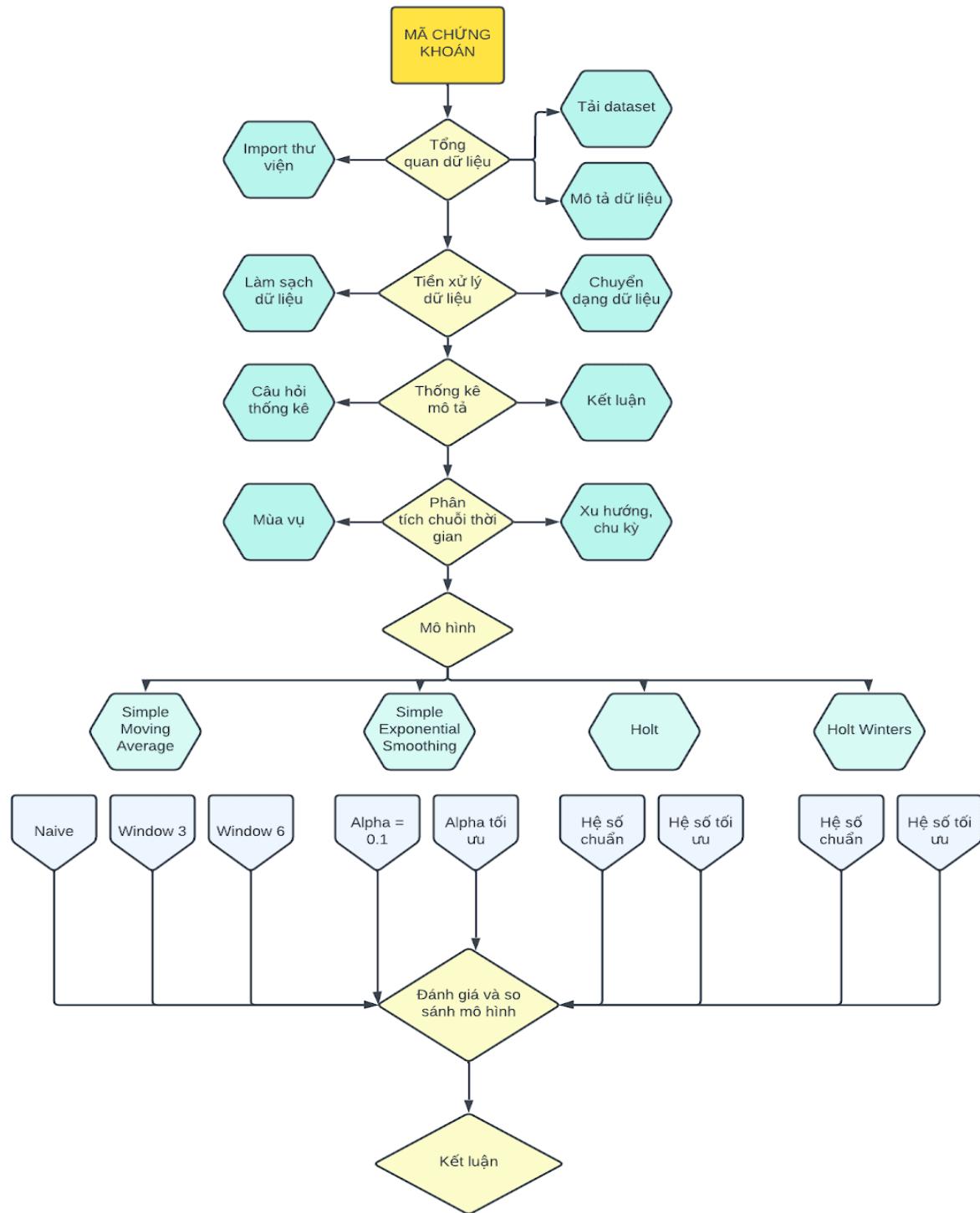
1. Giới thiệu

Trong những năm gần đây, thị trường chứng khoán đã trở nên đầy biến động do tác động của các yếu tố đa dạng như biến đổi chính sách, sự biến đổi kinh tế toàn cầu và các yếu tố nguy cơ như sau đại dịch, các cuộc chiến địa chính trị hay những xu hướng phát triển mới. Những thay đổi này đã tạo ra những thách thức cho việc phân tích và dự đoán trong ngành chứng khoán vốn đã nhiều thử thách. Đặc biệt, ngành công nghiệp ô tô đang trải qua những biến đổi sâu sắc từ sự cạnh tranh cao độ, công nghệ tiên tiến đến sự thay đổi trong yêu cầu về môi trường. Việc phân tích dữ liệu chứng khoán có thể giúp phát hiện xu hướng đầu tư, tiềm năng tăng trưởng và rủi ro trong ngành. Từ những yếu tố trên, nhóm quyết định lựa chọn đề tài “Phân tích dữ liệu chứng khoán của Tesla, Ford, Volkswagen, Toyota, BMW”, đây là 5 doanh nghiệp rất lớn và đi đầu trong phát triển ngành ô tô hiện nay. Nhóm hy vọng sẽ có cơ hội áp dụng kiến thức học được vào thực tế, hiểu sâu hơn về thị trường và có cái nhìn toàn diện hơn về ngành ô tô, đồng thời cung cấp cơ sở cho việc ra quyết định đầu tư thông minh và hiệu quả.

2. Sơ đồ phân rã chức năng

Dự án thể hiện được những chức năng chính sau khi áp dụng vào việc phân tích dữ liệu và dự báo:

- Nêu tổng quan về dữ liệu.
- Tiền xử lý dữ liệu.
- Thực hiện phân tích và thống kê mô tả.
- Phân tích chuỗi thời gian.
- Mô hình dự báo và kết quả.
- Đánh giá và so sánh mô hình thông qua bảng summary.
- Tiến hành tạo lập giao diện là trang web để phân tích và dự báo giá chứng khoán.
- Kết luận.



Hình 1. 1 Sơ đồ phân rã chức năng



3. Công cụ thực hiện

3.1 Python và thư viện sử dụng

3.1.1 Giới thiệu sơ lược về python

Python là một ngôn ngữ lập trình thông dịch, hướng đối tượng và là một ngôn ngữ bậc cao ngữ nghĩa động. Python hỗ trợ các module và gói, khuyến khích chương trình module hóa và tái sử dụng mã. Python được sử dụng rộng rãi trong các lĩnh vực như khoa học dữ liệu, trí tuệ nhân tạo, phát triển web, phát triển ứng dụng di động, và nhiều lĩnh vực khác. Python có cú pháp đơn giản, dễ đọc và dễ hiểu, giúp cho việc học và sử dụng Python trở nên dễ dàng hơn. Ngoài ra, Python còn có một cộng đồng lớn và nhiều tài liệu hướng dẫn, hỗ trợ rất tốt trong việc học và tìm hiểu.

3.1.2 Những thư viện được sử dụng

STT	Tên thư viện	Chức năng chính
1	Pandas	Pandas được thiết kế chủ yếu để xử lý và phân tích dữ liệu. Chức năng chính của Pandas là cung cấp cấu trúc dữ liệu linh hoạt như DataFrame và Series để dễ dàng thao tác và phân tích dữ liệu tabular: DataFrame và Series, Trục quan hóa dữ liệu, Thao tác dữ liệu với thời gian, Đánh chỉ số và multilndexing, Xử lý dữ liệu thiếu, Kết hợp và gộp dữ liệu,...
2	Numpy	NumPy tập trung vào cung cấp đối tượng mảng nhiều chiều và hỗ trợ các phép toán học hiệu quả trên các mảng này. Chức năng chính của NumPy là cung cấp một nền tảng mạnh mẽ cho tính toán khoa học và số học: Mảng, Thao tác mảng, Phép toán toán học, Truy cập và thay đổi phần tử mảng, Đại số tuyến tính, Tạo mảng, Thao tác dữ liệu thời gian, Hỗ trợ C-API,...
3	Math	Thư viện math trong Python cung cấp các hàm toán học cơ bản. Chức năng chính là hỗ trợ các phép toán và tính toán toán học trong các chương trình Python: Hàm số cơ học, Hàm số làm tròn và số nguyên, Hàm Trigonometry, Hàm số ngẫu nhiên, Hàm số tự nhiên,...
4	Datetime	Thư viện này cung cấp các lớp và hàm để làm việc với ngày tháng và thời gian trong Python.



		Lấy thông tin thời gian, Định dạng chuỗi thời gian, Các phép toán thời gian, lớp ‘datetime’ và ‘date’,...
5	Pyplot	Vẽ đồ thị cơ bản, Định dạng biểu đồ, Định dạng dữ liệu, Biểu đồ phân phôi, Lưu và hiển thị biểu đồ,...
6	statsmodels.api	statsmodels.api.summary: Hiển thị tóm tắt kết quả của mô hình. statsmodels.api.graphics: Hỗ trợ vẽ đồ thị và biểu đồ cho phân tích thống kê. statsmodels.api.GLM: Phân tích mô hình tuyến tính tổng quát. statsmodels.api.tsa: Mô hình và phân tích dữ liệu chuỗi thời gian....
7	Seaborn	cung cấp một giao diện cao cấp để vẽ các biểu đồ thống kê hấp dẫn và dễ đọc: Biểu đồ phân phôi, Biểu đồ tương quan, Biểu đồ phân loại, Thiết lập giao diện,...
8	Optuna	được thiết kế để tìm kiếm không gian siêu tham số để tối ưu hóa mục tiêu cụ thể, thường là một hàm mất mát trong quá trình huấn luyện mô hình máy học: Tìm kiếm tối ưu siêu tham số, Quản lý thông tin và kết quả thử nghiệm, Kiểm tra kết quả tìm kiếm,....
9	sklearn.metrics	Cung cấp các hàm và phương thức để đo lường và đánh giá hiệu suất của mô hình học máy <ul style="list-style-type: none">• mean absolute error: Đo lường độ chênh lệch trung bình giữa giá trị thực tế và giá trị dự đoán.• mean_squared_error: Đo lường độ chênh lệch bình phương trung bình giữa giá trị thực tế và giá trị dự đoán.• mean_squared_log_error: Đo lường độ chênh lệch bình phương logarithmic trung bình giữa giá trị thực tế và giá trị dự đoán.• r2_score: Đo lường độ giải thích của mô hình trên dữ liệu....



10	sklearn.preprocessing	<p>Được thiết kế để hỗ trợ quá trình tiền xử lý dữ liệu trước khi áp dụng các mô hình học máy</p> <ul style="list-style-type: none">StandardScaler: Chuẩn hóa dữ liệu bằng cách loại bỏ trung bình và chia tỷ lệ độ lệch chuẩn của mỗi đặc trưngMinmaxScaler: Chuyển đổi dữ liệu về một khoảng giá trị đã chọn thông qua việc chia tỷ lệ và dịch chúng.
11	statsmodels.tsa.seasonal	Chứa các công cụ và mô hình liên quan đến phân tích chuỗi thời gian với yếu tố mùa vụ (seasonal): Seasonal Decompose: Chia một chuỗi thời gian thành các thành phần trend, seasonal, và residual. Cung cấp các công cụ để phân tích và trực quan hóa các thành phần này,...
12	statsmodels.tsa.holtwinters	Chứa các công cụ và mô hình liên quan đến phân tích và dự đoán chuỗi thời gian sử dụng mô hình Holt-Winters. Mô hình Holt-Winters là một mô hình dự báo chuỗi thời gian, bao gồm các thành phần trend, seasonal và một thành phần error: Exponential Smoothing : Là lớp cung cấp mô hình Holt-Winters. Nó hỗ trợ các phiên bản khác nhau của mô hình, bao gồm mô hình có trend, mô hình có seasonal, và mô hình có cả trend và seasonal,..
13	statsmodels.graphics.tsaplots	Chứa các công cụ và hàm để vẽ các biểu đồ liên quan đến phân tích và kiểm định chuỗi thời gian: plot acf Vẽ hàm tương quan tự quyết định (ACF - Autocorrelation Function) của một chuỗi thời gian. ACF đo lường mức độ tương quan giữa các giá trị của chuỗi thời gian và các giá trị trước đó cùng một số lượng bước thời gian,...
14	statsmodels.tsa.stattools	Chứa các công cụ và hàm liên quan đến phân tích thống kê và kiểm định trong lĩnh vực chuỗi thời gian: Tính hàm tương quan tự quyết định (ACF) của chuỗi thời gian.
15	Plotly	Plotly là thư viện Python mạnh mẽ cho việc tạo đồ thị và biểu đồ tương tác. Nó hỗ trợ nhiều loại biểu đồ, tích hợp dễ dàng với các framework web, và có khả năng tương tác trực tiếp trên nền web.



		Plotly cũng được sử dụng rộng rãi cho việc thực hiện trực tiếp trong Jupyter Notebooks và hỗ trợ nhiều ngôn ngữ lập trình khác nhau.
16	Yfinance	yfinance là thư viện Python tiện ích cho việc truy xuất dữ liệu tài chính từ Yahoo Finance. Với API đơn giản, nó cho phép người dùng thuận lợi lấy thông tin như giá cổ phiếu, thay đổi giá, và khối lượng giao dịch. Sự dễ sử dụng và tích hợp linh hoạt làm cho yfinance trở thành một công cụ phổ biến trong việc phân tích dữ liệu tài chính và thị trường chứng khoán.
17	Streamlit	Streamlit là một framework mã nguồn mở được sử dụng để xây dựng ứng dụng web với Python một cách nhanh chóng và dễ dàng. Đây là một công cụ phổ biến cho việc tạo ra các ứng dụng dựa trên dữ liệu hoặc máy học mà không cần kiến thức sâu về front-end. Bằng cách sử dụng các thư viện như Pandas, Matplotlib và Plotly, nó có thể hiển thị và tương tác với dữ liệu một cách linh hoạt. Điểm mạnh của Streamlit là khả năng tập trung vào logic của ứng dụng.

Bảng 1.1 Thống kê các thư viện sử dụng

3.2 Những mô hình sử dụng để dự báo

a) Mô hình Moving Average

Mô hình moving average là một công cụ trong phân tích chuỗi dữ liệu thời gian để dự đoán xu hướng hoặc loại bỏ các dao động ngắn hạn. Mô hình moving average được sử dụng rộng rãi trong việc dự đoán xu hướng hoặc làm mịn dữ liệu bằng cách loại bỏ các dao động ngắn hạn, giúp nhìn nhận rõ hơn về xu hướng dài hạn của chuỗi dữ liệu thời gian.

- Mô hình Moving Average Naive
- Mô hình Moving Average 3 - step
- Mô hình Moving Average 6 - step

Công thức của mô hình moving average được tính bằng cách lấy trung bình cộng của các giá trị trong một cửa sổ thời gian di chuyển qua chuỗi dữ liệu. Được biểu diễn cụ thể như sau:

$$(X_{t-1} + X_{t-2} + \dots + X_{t-n})/n$$

Trong đó:



- $(X_{t-1} + X_{t-2} + \dots + X_{t-n})$ là các giá trị của chuỗi dữ liệu tại các thời điểm trước đó (ví dụ: giá cổ phiếu trong các ngày trước đó).
- n là số lượng giá trị được sử dụng để tính trung bình di chuyển (còn được gọi là độ dài cửa sổ).

Mục tiêu:

- Dự đoán xu hướng hoặc mức trung bình của chuỗi thời gian.
- Loại bỏ nhiễu và làm mịn dữ liệu.

Ưu điểm:

- Dễ triển khai và hiểu.
- Được sử dụng để xác định xu hướng trong dữ liệu thời gian.

Nhược điểm:

- Dễ triển khai và hiểu.
- Được sử dụng để xác định xu hướng trong dữ liệu thời gian.
b) Simple Exponential Smoothing

Mô hình Simple Exponential Smoothing là một phương pháp trong dự báo chuỗi dữ liệu thời gian, thường được sử dụng để dự đoán xu hướng hoặc mức độ biến động của chuỗi thời gian. Mô hình Simple Exponential Smoothing tập trung vào việc dự đoán giá trị tiếp theo dựa trên trọng số mà mỗi quan sát được gán và giá trị dự đoán trước đó. Mô hình này thường sử dụng giá trị dự đoán trước đó để cập nhật và tạo ra dự đoán mới.

- Simple Exponential Smoothing với $\alpha = 0,1$
- Simple Exponential Smoothing với $\alpha = 0,9$

Công thức của mô hình Simple Exponential Smoothing được biểu diễn như sau:

$$y^{t+1} = \alpha * y^t + (1 - \alpha) * y^t$$

Trong đó:

- y^{t+1} là giá trị dự đoán tiếp theo trong chuỗi thời gian.
- y^t là giá trị quan sát tại thời điểm t trong chuỗi.
- y^t là giá trị dự đoán tại thời điểm t trước đó.
- α là hằng số smoothing (smoothing parameter) có giá trị trong khoảng từ 0 đến 1, quyết định mức độ ảnh hưởng của giá trị mới vào việc dự đoán.

Giá trị của α quyết định mức độ quan trọng giữa giá trị mới và dự đoán trước đó. Nếu α gần với 0, mô hình sẽ đặc biệt nhấn mạnh vào giá trị dự đoán trước đó hơn. Ngược lại, nếu α



gần với 1, mô hình sẽ đặc biệt nhấn mạnh vào giá trị mới nhất hơn trong việc tạo ra dự đoán.

Mục tiêu:

- Dự đoán xu hướng tương lai của chuỗi thời gian dựa trên thông tin quan sát trước đó.
- Loại bỏ nhiễu và làm mịn dữ liệu.

Ưu điểm:

- Dễ triển khai và hiểu.
- Phản ánh sự quan trọng của các quan sát gần đây hơn so với quan sát xa hơn.

Nhược điểm:

- Cần phải chọn giá trị α phù hợp, điều này có thể yêu cầu thử nghiệm hoặc tối ưu hóa.
- Không xử lý tốt với các chuỗi thời gian có xu hướng hoặc mô hình phức tạp hơn.
c) Mô hình Holt

Mô hình Holt là một mô hình mở rộng của mô hình Simple Exponential Smoothing, được sử dụng để dự báo chuỗi dữ liệu thời gian với xu hướng tăng hoặc giảm. Bao gồm hai thành phần chính: thành phần mức (level) và thành phần xu hướng (trend). Nó cho phép tính toán một cách linh hoạt với cả thành phần mức và thành phần xu hướng, giúp dự báo được cả xu hướng tăng hoặc giảm trong chuỗi dữ liệu thời gian.

- Holt hệ số chuẩn.
- Holt hệ số tối ưu.

Công thức của mô hình Holt được mô tả như sau:

Công thức cập nhật cho thành phần mức (l_t):

$$l_t = \alpha \cdot y_t + (1 - \alpha) \cdot (l_{t-1} + b_{t-1})$$

Công thức cập nhật cho thành phần xu hướng (b_t):

$$b_t = \beta \cdot (l_t - l_{t-1}) + (1 - \beta) \cdot b_{t-1}$$

Công thức dự đoán giá trị tiếp theo (y^{t+1}):

$$y^{t+1} = l_t + b_t$$

Trong đó:



- lt là thành phần mức tại thời điểm t.
- bt là thành phần xu hướng tại thời điểm t.
- yt là giá trị quan sát tại thời điểm t trong chuỗi.
- α và β là các hằng số smoothing (trong khoảng từ 0 đến 1) quyết định mức độ ảnh hưởng của các thành phần trước đó đối với dự đoán.
- y^{t+1} là giá trị dự đoán tiếp theo.

Ưu điểm:

- Phù hợp với xu hướng biến đổi dài hạn: Mô hình Holt cho phép dự đoán và mô hình hóa các xu hướng tăng hoặc giảm trong chuỗi dữ liệu thời gian.
- Tính linh hoạt: Có thể áp dụng cả cho dữ liệu có thành phần mùa vụ hoặc không có thành phần mùa vụ.
- Dễ dàng thực hiện và hiểu: Mô hình này có cấu trúc đơn giản, dễ áp dụng và hiểu.

Nhược điểm:

- Không phản ánh được sự biến đổi của dữ liệu không đều: Trong trường hợp dữ liệu có sự biến đổi không đều qua thời gian, mô hình Holt có thể không hiệu quả.
- Cần cấu hình tham số tốt: Đôi khi việc chọn và điều chỉnh các tham số như hệ số smoothing (alpha và beta) có thể gây khó khăn và ảnh hưởng đến chất lượng dự đoán của mô hình.
- Giả định tĩnh về xu hướng và mùa vụ: Mô hình Holt giả định rằng xu hướng và thành phần mùa vụ không thay đổi qua thời gian, điều này có thể không phù hợp với dữ liệu có sự thay đổi đột ngột hoặc không ổn định.

d) Mô hình Holt Winter

Mô hình Holt-Winters là một phương pháp dự báo chuỗi thời gian trong phân tích dữ liệu. Phương pháp này giúp dự đoán xu hướng và mô hình hóa thành phần mùa vụ trong chuỗi dữ liệu thời gian. Hệ số tối ưu trong Holt-Winter bao gồm việc tối ưu hóa các tham số α , β và γ để tăng độ chính xác của dự đoán trong mô hình.

- Holt Winter hệ số chuẩn.
- Holt Winter hệ số tối ưu.

Mô hình Holt-Winters sử dụng ba thành phần chính để mô tả dữ liệu:

- Thành phần mức (Level): Đây là giá trị trung bình của chuỗi dữ liệu tại mỗi điểm thời gian.
- Thành phần xu hướng (Trend): Là sự thay đổi dài hạn trong dữ liệu, biểu thị xu hướng tăng hoặc giảm qua các quan sát.



- Thành phần mùa vụ (Seasonal): Là biến đổi theo chu kỳ trong dữ liệu, thường là các biến đổi lặp lại theo mùa vụ hoặc chu kỳ cố định.

Công thức mô hình:

Công thức cập nhật cho thành phần mức (lt):

$$lt = \alpha * st - myt + (1 - \alpha) * (lt - 1 + bt - 1)$$

Công thức cập nhật cho thành phần xu hướng (bt):

$$bt = \beta * (lt - lt - 1) + (1 - \beta) * bt - 1$$

Công thức cập nhật cho thành phần mùa vụ (st):

$$st = \gamma * ltyt + (1 - \gamma) * st - m$$

Công thức dự đoán giá trị tiếp theo (y^{t+1}):

$$y^{t+1} = (lt + bt) * st - m + 1$$

Trong đó:

- lt là thành phần mức tại thời điểm t.
- bt là thành phần xu hướng tại thời điểm t.
- st là thành phần mùa vụ tại thời điểm t.
- yt là giá trị quan sát tại thời điểm t trong chuỗi.
- α, β và γ là các hằng số smoothing (trong khoảng từ 0 đến 1) quyết định mức độ ảnh hưởng của các thành phần trước đó đối với dự đoán.
- m là độ dài của chu kỳ mùa vụ.

Ưu điểm:

- Phù hợp với xu hướng biến đổi dài hạn: Mô hình Holt cho phép dự đoán và mô hình hóa các xu hướng tăng hoặc giảm trong chuỗi dữ liệu thời gian.
- Tính linh hoạt: Có thể áp dụng cả cho dữ liệu có thành phần mùa vụ hoặc không có thành phần mùa vụ.
- Dễ dàng thực hiện và hiểu: Mô hình này có cấu trúc đơn giản, dễ áp dụng và hiểu.

Nhược điểm:

- Không phản ánh được sự biến đổi của dữ liệu không đều: Trong trường hợp dữ liệu có sự biến đổi không đều qua thời gian, mô hình Holt có thể không hiệu quả.
- Cần cấu hình tham số tốt: Đôi khi việc chọn và điều chỉnh các tham số như hệ số smoothing (alpha và beta) có thể gây khó khăn và ảnh hưởng đến chất lượng dự đoán của mô hình.



- Giả định tĩnh về xu hướng và mùa vụ: Mô hình Holt giả định rằng xu hướng và thành phần mùa vụ không thay đổi qua thời gian, điều này có thể không phù hợp với dữ liệu có sự thay đổi đột ngột hoặc không ổn định.

3.3 Giới thiệu về dữ liệu

Dữ liệu được lấy từ trang Yahoo Finance: Yahoo Finance là một trang web và ứng dụng cung cấp thông tin tài chính, tin tức và dữ liệu thị trường cho người đầu tư, nhà giao dịch và những người quan tâm đến thị trường tài chính. Dịch vụ này là một trong những nguồn thông tin tài chính hàng đầu trên thế giới và cung cấp một loạt các công cụ hữu ích để theo dõi và nghiên cứu thị trường.

Là dữ liệu giá cổ phiếu trong 5 năm gần đây từ năm 2018-2023

4. Thông tin nhóm và phân chia nhiệm vụ

Nhóm thực hiện đề tài trong vòng 2 tuần, với 3 thành viên chính, mỗi thành viên đảm nhiệm những nhiệm vụ khác nhau như bảng 1.2. Nhóm đã thực hiện tải dữ liệu và tiến hành phân tích dự báo, tất cả công cụ, thư viện và những yếu tố phần mềm cần thiết để demo dữ liệu nhóm đã tiến hành đóng gói lại vào dự án cộng đồng trên nền tảng GitHub.

a) Giới thiệu về nền tảng Github và dự án của nhóm

GitHub là một nền tảng lưu trữ mã nguồn và quản lý dự án phổ biến dành cho các nhà phát triển phần mềm. Nó cho phép người dùng lưu trữ mã nguồn của dự án, theo dõi sự thay đổi và hợp nhất các thay đổi. Các tính năng chính của GitHub bao gồm:

- Lưu trữ mã nguồn: Người dùng có thể tạo và quản lý các kho lưu trữ (repositories) để lưu trữ mã nguồn của dự án.
- Quản lý phiên bản: GitHub ghi lại lịch sử thay đổi của mã nguồn, cho phép người dùng theo dõi các sửa đổi, trở về phiên bản cũ, và hợp nhất các thay đổi.
- Hợp tác và Phản hồi: Nền tảng này cho phép các nhà phát triển cùng làm việc trên dự án, đưa ra đề xuất (pull request), xem xét và thảo luận về mã nguồn.
- Công cụ Quản lý Dự án: GitHub cung cấp các công cụ quản lý dự án như quản lý nhiệm vụ (issues), wiki, và dự án kanban board để theo dõi tiến độ công việc.

GitHub đã trở thành một trong những công cụ quan trọng cho cộng đồng phát triển phần mềm, cung cấp nền tảng cho sự hợp tác, kiểm soát phiên bản, và lưu trữ mã nguồn dự án một cách dễ dàng và hiệu quả. Hợp tác với đồng nghiệp và kiểm soát phiên bản.

Dự án của nhóm trên nền tảng GitHub:

Link truy cập dự án: [GITHUB-STOCK-PRICE-PREDICTION-PYTHON](https://github.com/yourusername/GITHUB-STOCK-PRICE-PREDICTION-PYTHON)

- Thông tin cách cài đặt và sử dụng: README.md



- Thông tin các thư viện cần thiết: requirements.txt

- Cấu trúc thư mục dự án github:

Cấu Trúc Thư Mục (Directory Structure)

- `dataset/` : Chứa dữ liệu lịch sử về giá cổ phiếu (Contains historical stock price data).
- `train_folder/` : Chứa các notebook Jupyter cho phân tích và dự đoán (Contains Jupyter notebooks for analysis and prediction).
- `asset/image/` : Lưu trữ ảnh của dự án (Stores project images)
- `info_stock/` : Lưu trữ thông tin cổ phiếu (Stores stock information).
- `introduction` : Bản thảo word của dự án (Word draft of the project).
- `app_test.py` : Tập chứa giao diện demo thuật toán (The file contains the algorithm demo interface)

Hình 1. 2 Cấu trúc thư mục dự án Github

b) Phân chia nhiệm vụ:

STT	Họ và tên	MSSV	Nhiệm vụ
1	Hoàng Xuân Quốc	2156210125	Phân tích cổ phiếu TSLA, đánh giá kết quả thực hiện, giao diện phân tích dự báo
2	Nguyễn Viết Đức	2156210100	Phân tích cổ phiếu Ford và Vosgogen, làm báo cáo
3	Đặng Hoàng Chiến	2156210095	Phân tích cổ phiếu Toyota và BMW, làm báo cáo

Bảng 1. 2 Phân công nhiệm vụ

II. QUÁ TRÌNH THỰC HIỆN VÀ KẾT QUẢ

1. Tổng quan dữ liệu

1.1 Import thư viện

- Import những thư viện cần thiết để sử dụng trong bài làm(các thư viện đã được mô tả rõ ở trên)



```
import os
import pandas as pd
import numpy as np
import math
import datetime as dt
from matplotlib import pyplot as plt
import statsmodels.api as sm
import seaborn as sns
import optuna
import yfinance as yf

from sklearn.metrics import mean_squared_error, mean_absolute_error, explained_variance_score, explained_variance_score, r2_score, mean_absolute_percentage_error
from sklearn.preprocessing import MinMaxScaler
from pickle import TRUE
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.holtwinters import SimpleExpSmoothing
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.tsa.stattools import acf
```

Hình 2. 1 Import thư viện

1.2 Tải dataset

- Tải dataset cho file làm.

```
[ ] df = pd.read_csv('../dataset/TESSA.csv')
```

Hình 2. 2 Tải dataset

1.3 Mô tả dữ liệu

- Mô tả kiểu dữ liệu của từng cột trong dataset.

```
# Mô tả kiểu dữ liệu của từng cột trong data set
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1259 entries, 0 to 1258
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Date        1259 non-null   object  
 1   Open         1259 non-null   float64 
 2   High         1259 non-null   float64 
 3   Low          1259 non-null   float64 
 4   Close        1259 non-null   float64 
 5   Adj Close    1259 non-null   float64 
 6   Volume       1259 non-null   int64  
dtypes: float64(5), int64(1), object(1)
memory usage: 69.0+ KB
```

Hình 2. 3 Mô tả kiểu dữ liệu từng cột

- Xem 10 dòng đầu tiên của dữ liệu.

```
# Xem 10 dòng đầu tiên của dataset
df.head(10)
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2018-12-24	20.900000	20.966667	19.680000	19.692667	19.692667	83398500
1	2018-12-26	20.000000	21.798000	19.606001	21.739332	21.739332	122446500
2	2018-12-27	21.322666	21.478001	20.100000	21.075333	21.075333	128626500
3	2018-12-28	21.540001	22.416000	21.227333	22.257999	22.257999	149085000
4	2018-12-31	22.519333	22.614000	21.684000	22.186666	22.186666	94534500
5	2019-01-02	20.406668	21.008667	19.920000	20.674667	20.674667	174879000
6	2019-01-03	20.466667	20.626667	19.825333	20.024000	20.024000	104478000
7	2019-01-04	20.400000	21.200001	20.181999	21.179333	21.179333	110911500
8	2019-01-07	21.448000	22.449333	21.183332	22.330667	22.330667	113268000
9	2019-01-08	22.797333	22.934000	21.801332	22.356667	22.356667	105127500

Hình 2. 4 Xem 10 dòng đầu tiên của dữ liệu

2. Tiết xu lý dữ liệu

2.1 Làm sạch dữ liệu

- Thay đổi tên các cột trong dataset và xem lại dataset

```
# Thay đổi tên cột để dễ thao tác
df= df.rename(columns={'Date': 'date','Open': 'open','Hight': 'high','Low': 'low','Close': 'close','Adj Close': 'adj_close','Volume': 'volume'})
df.head(10)
```

	date	open	high	low	close	adj_close	volume
0	2018-12-24	20.900000	20.966667	19.680000	19.692667	19.692667	83398500
1	2018-12-26	20.000000	21.798000	19.606001	21.739332	21.739332	122446500
2	2018-12-27	21.322666	21.478001	20.100000	21.075333	21.075333	128626500
3	2018-12-28	21.540001	22.416000	21.227333	22.257999	22.257999	149085000
4	2018-12-31	22.519333	22.614000	21.684000	22.186666	22.186666	94534500

Hình 2. 5 Thay đổi tên các cột

- Kiểm tra dữ liệu thiếu, dữ liệu trống

```
# Kiểm tra dữ liệu thiếu
print("Null values", df.isnull().values.sum())
print("NA values:", df.isna().values.any())

Null values 0
NA values: False
```

Hình 2. 6 Kiểm tra dữ liệu trống, dữ liệu thiếu

- Kiểm tra dữ liệu trùng lặp

```
# Kiểm tra dữ liệu lặp
df.duplicated().sum()

0
```

Hình 2. 7 Kiểm tra dữ liệu trùng lặp



2.2 Chuyển dạng dữ liệu

- Chuyển đổi cột date thành dạng dữ liệu thời gian datetime.

```
# Chuyển cột date sang datetime
df['date'] = pd.to_datetime(df.date)
df.head(5)
```

	date	open	High	low	close	adj_close	volume
0	2018-12-24	20.900000	20.966667	19.680000	19.692667	19.692667	83398500
1	2018-12-26	20.000000	21.798000	19.606001	21.739332	21.739332	122446500
2	2018-12-27	21.322666	21.478001	20.100000	21.075333	21.075333	128626500
3	2018-12-28	21.540001	22.416000	21.227333	22.257999	22.257999	149085000
4	2018-12-31	22.519333	22.614000	21.684000	22.186666	22.186666	94534500

Hình 2. 8 Chuyển dữ liệu cho cột date

- In tổng số hàng và tổng số cột có trong dữ liệu

```
print("Total number of day: ", df.shape[0])
print("Total number of fields: ", df.shape[1])
```

Total number of day: 1259
Total number of fields: 7

Hình 2. 9 In ra dữ liệu hiện có

3. Thống kê mô tả

3.1 Các vấn đề thống kê

- a) Mô tả cơ bản về dữ liệu
- In ra ngày bắt đầu, ngày kết thúc và tổng số ngày trong dataframe

```
print("Starting date:", df.loc[0]['date'])
print("Ending date:", df.loc[df.index[-1], 'date'])
print("Total date", df.loc[df.index[-1], 'date'] - df.loc[0, 'date'])
```

Starting date: 2018-12-24 00:00:00
Ending date: 2023-12-22 00:00:00
Total date 1824 days 00:00:00

Hình 2. 10 Thông tin ngày bắt đầu, kết thúc, tổng số ngày

- b) Các thông tin: giá trị cao nhất, thấp nhất, giá trị trung bình
- Tóm tắt thống kê của các giá trị trong DataFrame, bao gồm các thống kê như số lượng dòng, trung bình, độ lệch chuẩn, giá trị tối thiểu, các phần vị, và giá trị tối đa của các cột có thể tính toán.



df.describe()							
	date	open	High	low	close	adj_close	volume
count	1259	1259.000000	1259.000000	1259.000000	1259.000000	1259.000000	1.259000e+03
mean	2021-06-24 06:04:51.660047616	170.068495	173.920073	165.922177	170.026750	170.026750	1.340823e+08
min	2018-12-24 00:00:00	12.073333	12.445333	11.799333	11.931333	11.931333	2.940180e+07
25%	2020-03-25 12:00:00	48.586668	50.715000	46.710333	49.277334	49.277334	8.010540e+07
50%	2021-06-24 00:00:00	199.300003	203.333328	194.070007	199.316666	199.316666	1.095203e+08
75%	2022-09-22 12:00:00	250.906662	255.473327	244.934998	251.470001	251.470001	1.583427e+08
max	2023-12-22 00:00:00	411.470001	414.496674	405.666656	409.970001	409.970001	9.140820e+08
std		Nan	108.708024	111.052511	106.066435	108.563841	8.542378e+07

Hình 2. 11 Một số thông tin cơ bản về dữ liệu

- c) Kiểm tra tương quan giữa các biến, sự khác biệt
- Nhóm các dữ liệu từ cột date, tính giá trị trung bình của cột open và close sau đó sắp xếp lại theo tự tự tên các tháng

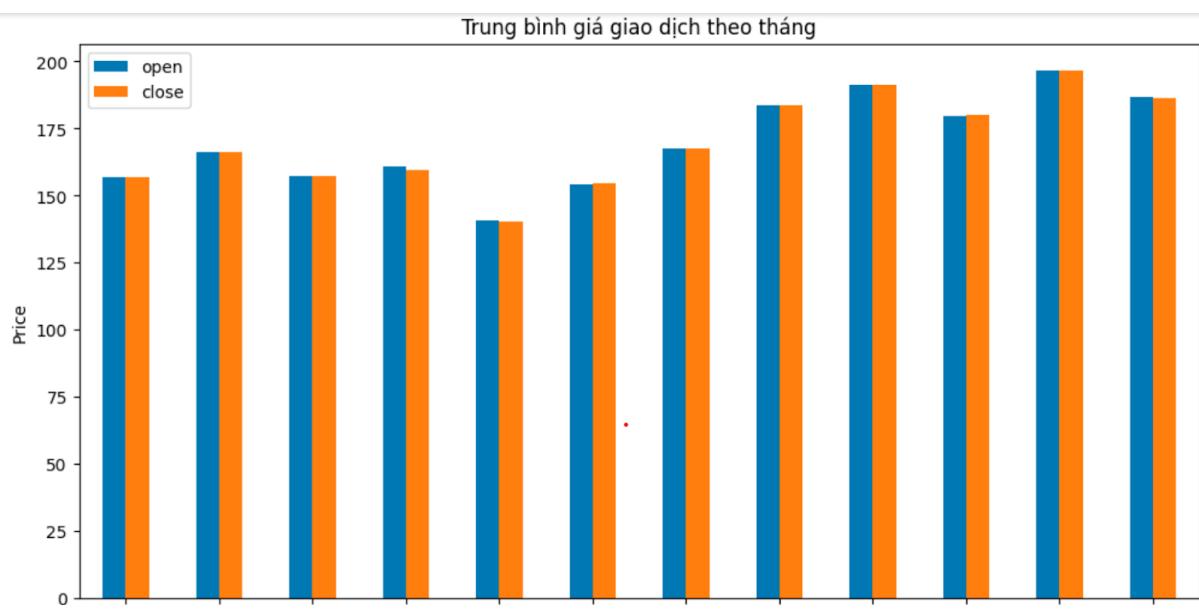
```
# Nhóm dữ liệu theo tháng của các cột open,close,high và low
monthvise = df.groupby(df['date'].dt.strftime('%B'))[['open','close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']
monthvise = monthvise.reindex(new_order, axis=0)
monthvise.head()
```

	open	close
date		
January	156.991492	156.963478
February	166.057768	166.234336
March	157.045744	157.059154
April	160.669125	159.618458
May	140.527238	140.258159

Hình 2. 12 Nhóm dữ liệu theo tháng

- Trung bình giá giao dịch theo tháng. Vẽ biểu đồ cột “kind = bar”, dựa liệu phía trên “monthvise”

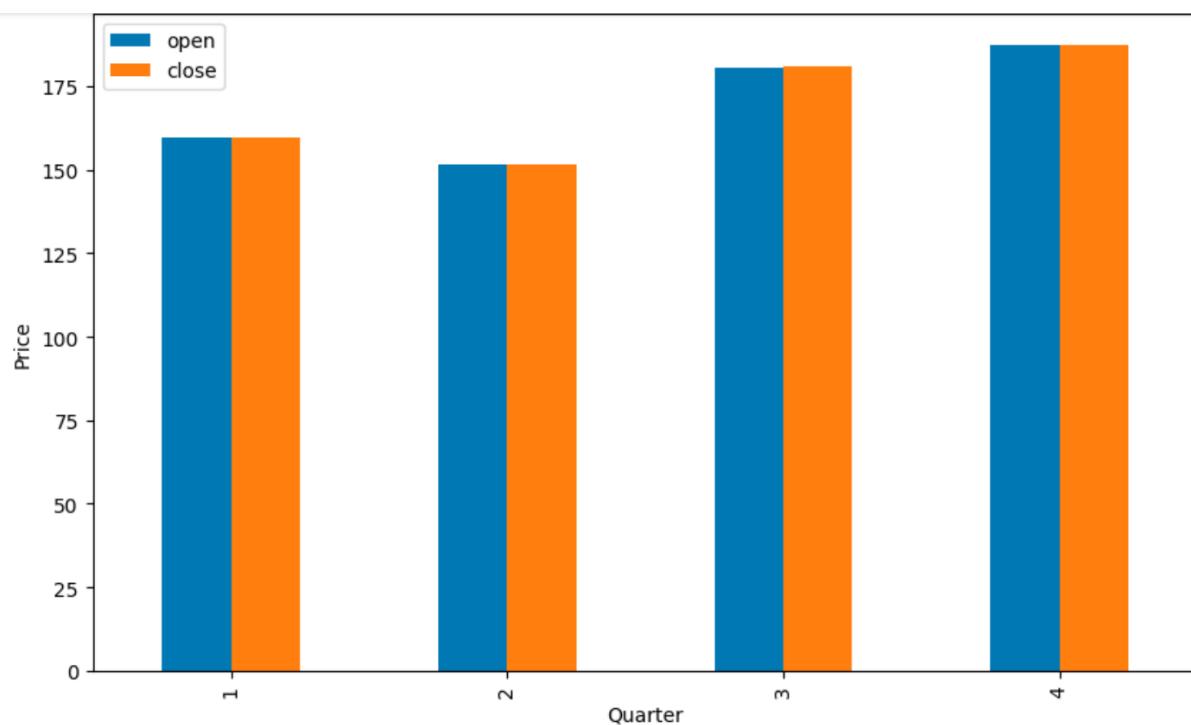
```
# Vẽ bar-plot mô tả
monthvise.plot(kind = 'bar', figsize=(12,6))
plt.xlabel('Month')
plt.ylabel('Price')
plt.title('Trung bình giá giao dịch theo tháng')
plt.show()
```



Hình 2. 13 Biểu đồ trung bình giá giao dịch theo tháng

- Trung bình giá giao dịch theo quý, vẽ biểu đồ cột “kind = bar” với nhóm dữ liệu theo quý

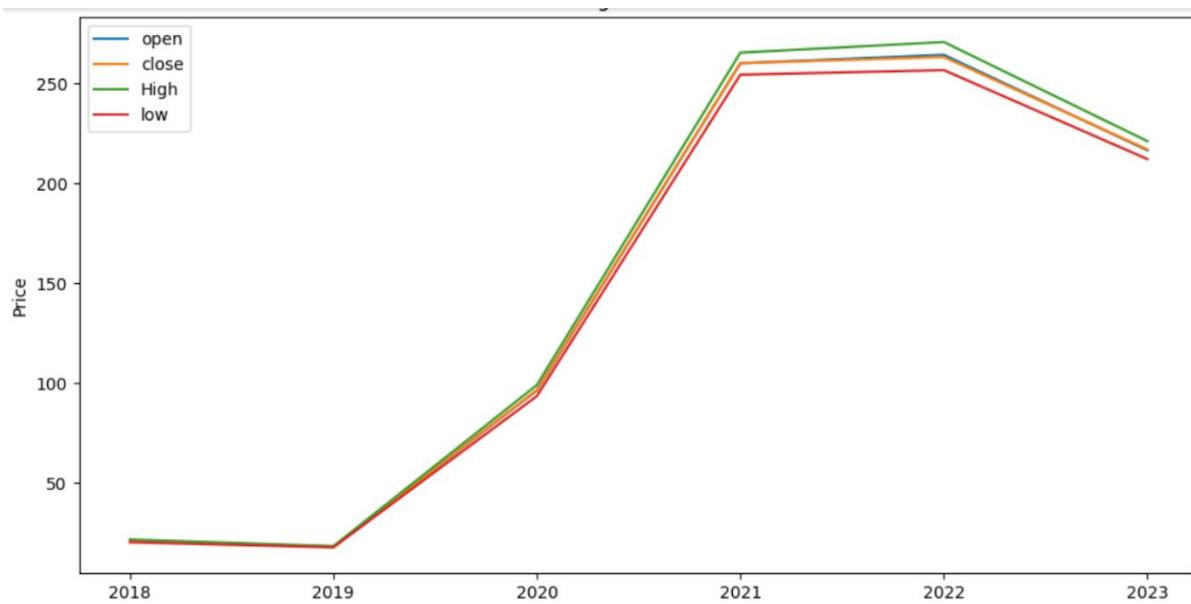
```
df.groupby(df['date'].dt.quarter)[['open','close']].mean().plot(kind='bar',figsize=(10,6))
plt.xlabel('Quarter')
plt.ylabel('Price')
plt.title('Giá trung bình theo quý')
plt.show()
```



Hình 2. 14 Biểu đồ trung bình giá giao dịch theo quý

- Biểu đồ giá trung bình theo năm. Nhóm dữ liệu theo năm từ cột 'date' và tính giá trung bình của các cột 'open', 'close', 'High', và 'low'. Vẽ biểu đồ cột “kind = bar” dựa trên dữ liệu được nhóm.

```
df.groupby(df['date'].dt.year)[['open','close','High','low']].mean().plot(kind='bar',figsize=(12,6))
plt.xlabel('Year')
plt.ylabel('Price')
plt.title('Giá trung bình theo năm')
plt.show()
```



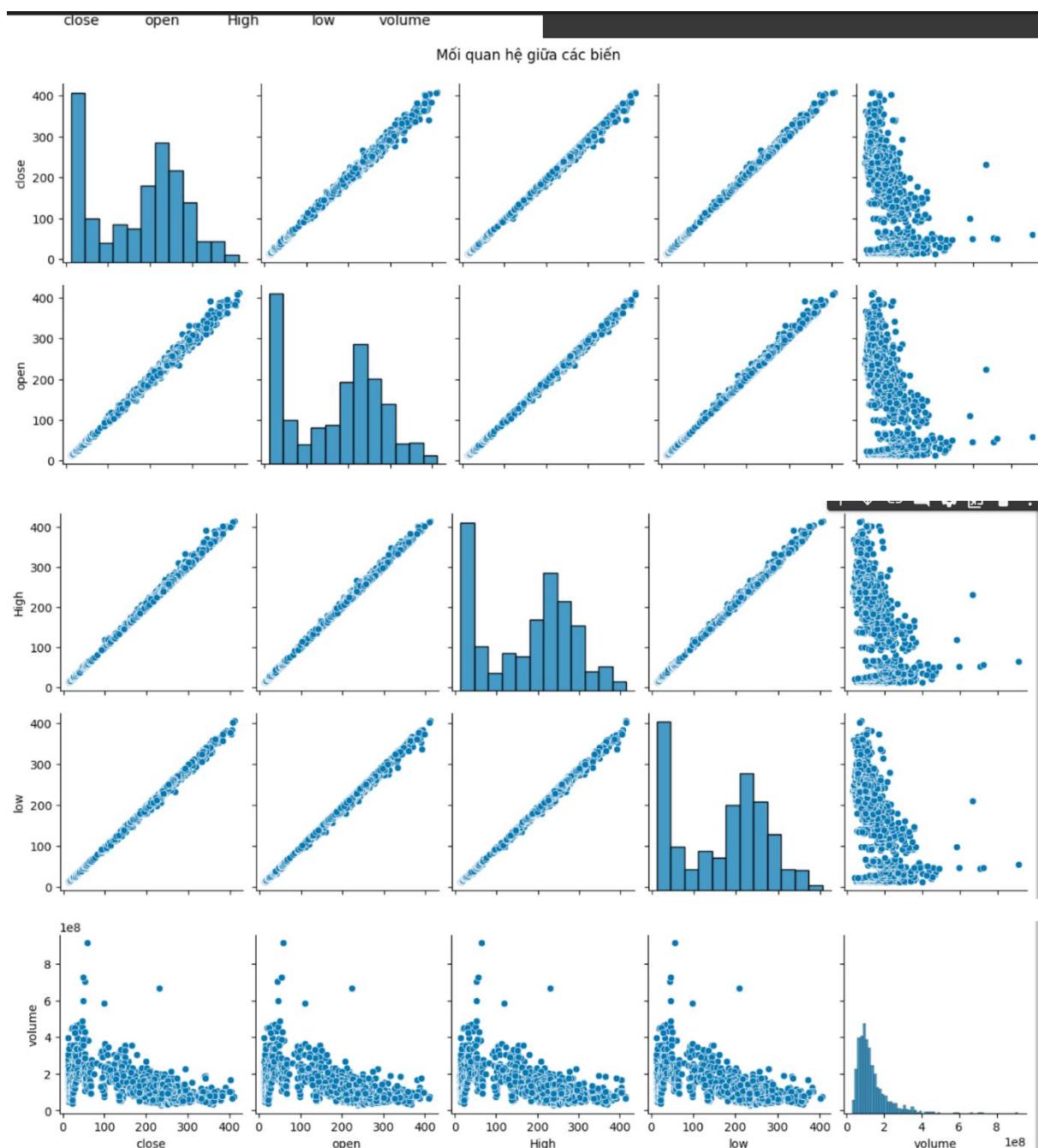
Hình 2. 15 Biểu đồ trung bình giá giao dịch theo năm

- Vẽ biểu đồ heatmap (sns.heatmap) để hiển thị độ tương quan giữa các biến từ DataFrame (df) dựa trên ma trận tương quan corr. Màu sắc trên biểu đồ sẽ thể hiện mức độ tương quan, với annot=True để hiển thị giá trị tương quan trên từng ô của heatmap. Vẽ biểu đồ Pairplot (sns.pairplot) để hiển thị mối quan hệ giữa các cặp biến từ danh sách columns_to_corr. Đây là một loạt các biểu đồ phân tán cho từng cặp biến, và plt.suptitle được sử dụng để đặt tiêu đề chung cho các biểu đồ này.

```
# Vẽ heatmap thể hiện độ tương quan giữa các biến
columns_to_corr = ['close', 'open', 'High', 'low', 'volume']
corr = df[columns_to_corr].corr()
sns.heatmap(corr, cmap='coolwarm', annot=True)
plt.title('Tương quan giữa các biến ')
plt.show()
# Vẽ Pairplot
sns.pairplot(data=df[columns_to_corr])
plt.suptitle('Mối quan hệ giữa các biến ', y=1.02)
plt.show()
```



Hình 2. 16 Biểu đồ Heatmap



Hình 2. 17 Biểu đồ Pairplot mối tương quan các biến

d) Kiểm tra phân phối dữ liệu

- Tạo dataframe mới bằng hai cột close và date.

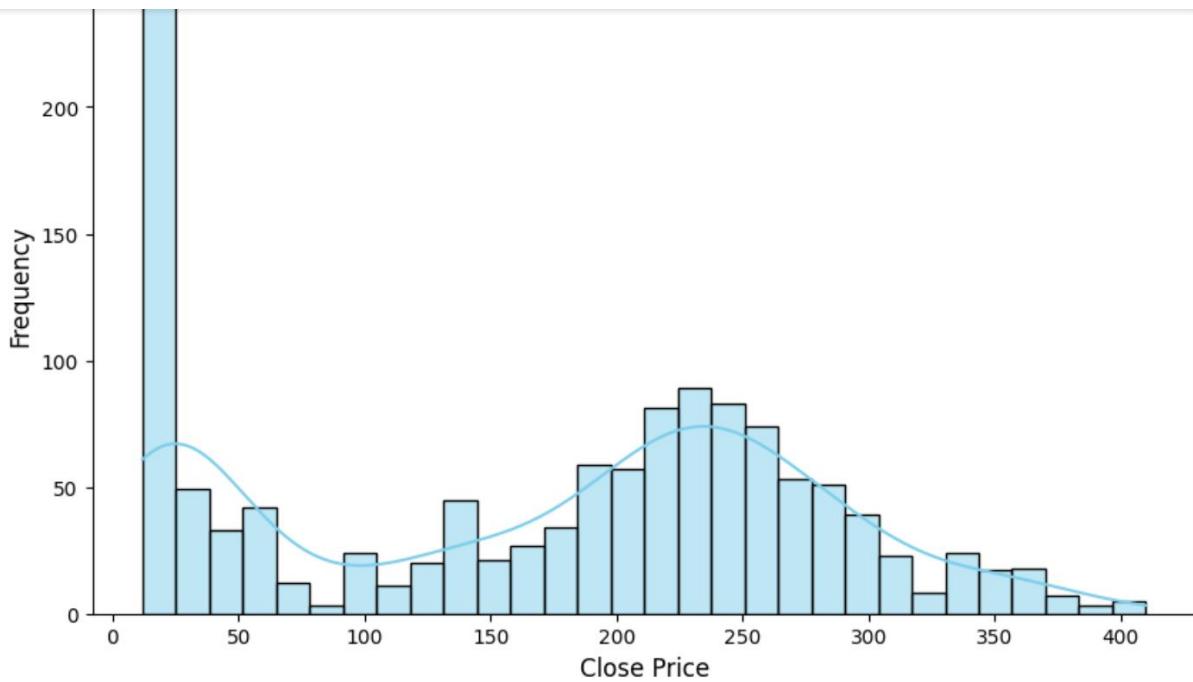


```
# Tạo dataframe mới bằng cột close và date  
closedf = df[['date','close']]
```

Hình 2. 18 Tạo closedf

- Vẽ biểu đồ histogram (`sns.histplot`) cho cột 'close' từ DataFrame (`df`). Sử dụng `bins=30` để chia dữ liệu thành 30 khoảng, `kde=True` để hiển thị đường cong ước lượng mật độ xác suất. Tính toán và in ra độ xiên của dữ liệu cột 'close' thông qua hàm `skew()` để đo độ méo của phân phối dữ liệu.

```
# Vẽ histogram cho cột 'close'  
plt.figure(figsize=(10, 6))  
sns.histplot(closedf['close'], bins=30, kde=True, color='skyblue')  
plt.title('Phân phối close Prices ', fontsize=14)  
plt.xlabel('Close Price', fontsize=12)  
plt.ylabel('Frequency', fontsize=12)  
  
# Hiển thị đồ thị histogram  
plt.show()  
  
# Đo độ xiên của dữ liệu  
skewness = df['close'].skew()  
print(f"Độ xiên của dữ liệu: {skewness}")
```



Độ xiên của dữ liệu: -0.14202842019334957

Hình 2. 19 Histogram

- e) Thẻ hiện giá trung bình theo tháng và theo quý năm 2023



- Tạo một datadrame mới **close_stock_2023**, chứa dữ liệu từ cột 'date' sau ngày '2023-01-01' trong dataframe gốc. In ra tổng số ngày trong tập dữ liệu mới. In ra toàn bộ **close_stock_2023** để xem dữ liệu trong phạm vi thời gian này.

```
# Tạo biến mới chứa dữ liệu 2023
close_stock_2023 = closedf[closedf['date'] > '2023-01-01'].copy()
print("Total date for prediction: ", close_stock_2023.shape[0])
print(close_stock_2023)

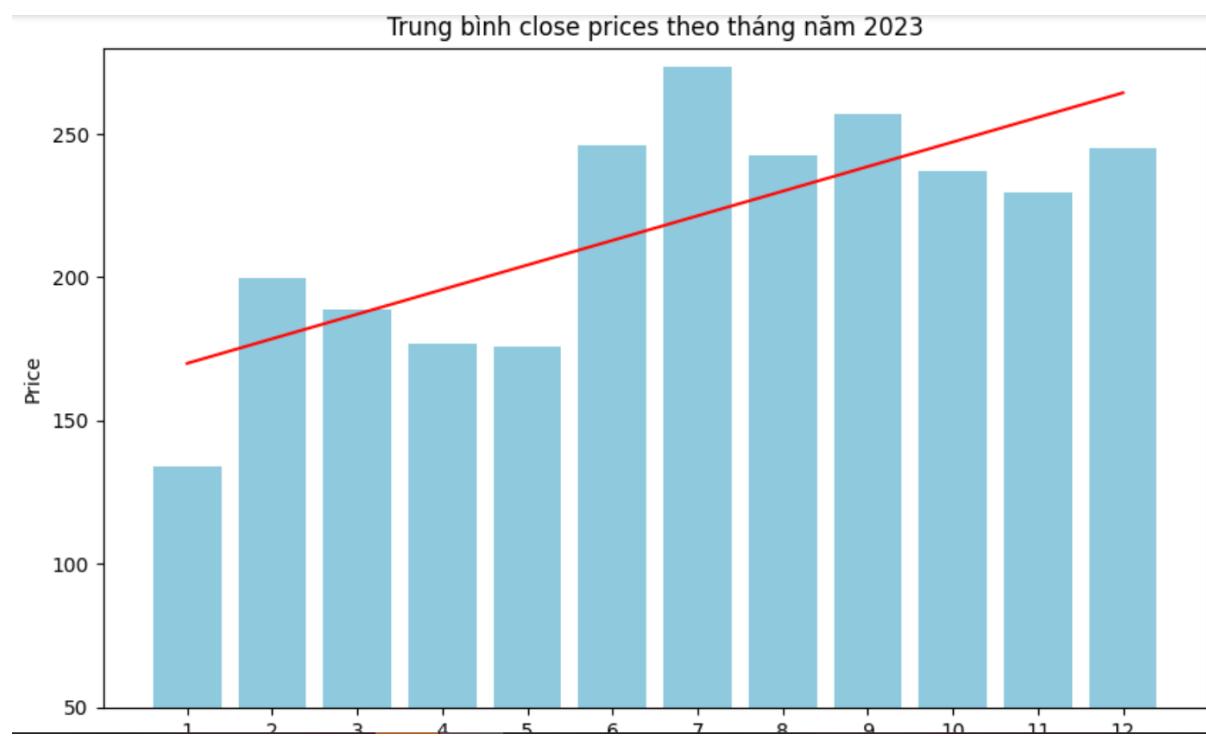
Total date for prediction: 246
      date       close
1013 2023-01-03  108.099998
1014 2023-01-04  113.639999
1015 2023-01-05  110.339996
1016 2023-01-06  113.059998
1017 2023-01-09  119.769997
...
1254 2023-12-18  252.080002
1255 2023-12-19  257.220001
1256 2023-12-20  247.139999
1257 2023-12-21  254.500000
1258 2023-12-22  252.539993

[246 rows x 2 columns]
```

Hình 2. 20 Tạo close_stock_2023

- Vẽ biểu đồ dạng thanh (**sns.barplot**) hiển thị giá trị trung bình của cột 'close' theo từng tháng trong năm 2023 từ **close_stock_2023**. Sau đó, sử dụng **np.polyfit** để tính toán và vẽ đường thẳng tuyến tính (**plt.plot**) dựa trên giá trị trung bình của cột 'close' theo tháng.

```
plt.figure(figsize=(10, 6))
sns.barplot(close_stock_2023.groupby(close_stock_2023['date'].dt.month)[['close']].mean(), color="skyblue")
x_values = np.unique(close_stock_2023['date'].dt.month - 1) # Lấy giá trị duy nhất của tháng
y_values = close_stock_2023.groupby(close_stock_2023['date'].dt.month)[['close']].mean()
slope, intercept = np.polyfit(x_values, y_values, 1)
plt.plot(x_values, slope * x_values + intercept, color='red', linestyle='solid')
plt.xlabel('Month')
plt.ylabel('Price')
plt.ylim(50,280)
plt.title('Trung bình close prices theo tháng năm 2023')
plt.show()
```



Hình 2. 21 Trung bình close price theo tháng 2023

- Vẽ biểu đồ dạng thanh (`sns.barplot`) hiển thị giá trị trung bình của cột 'close' theo từng quý trong năm 2023 từ `close_stock_2023`.

```
plt.figure(figsize=(10, 6))
sns.barplot(close_stock_2023.groupby(close_stock_2023['date'].dt.quarter)['close'].mean(), color="skyblue")
plt.xlabel('Quarter')
plt.ylabel('Price')
plt.title('Trung bình close price theo quý năm 2023')
plt.show()
```



Hình 2. 22 Trung bình close price theo quý

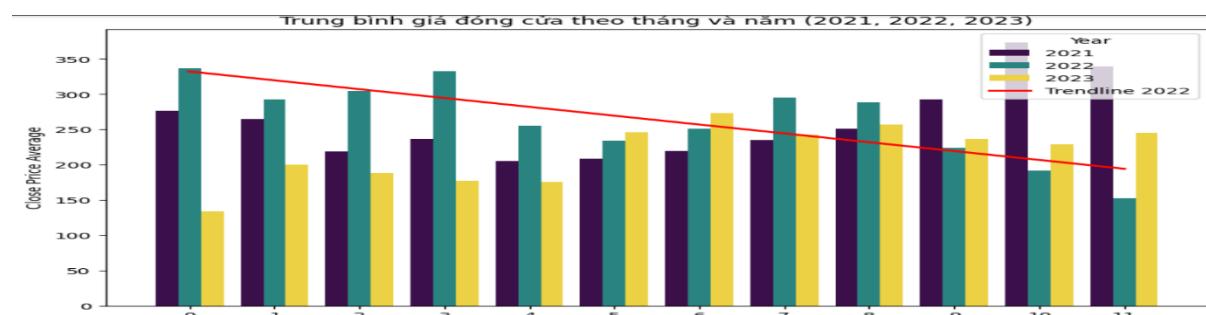
- f) So sánh giá cổ phiếu trung bình các năm 2021,2022,2023
- Tạo 1 dataframe mới có tên là **closedf** chỉ với hai cột date và close

```
closedf = df[['date', 'close']].copy()
```

Hình 2. 23 Lấy closedf mới

- Tạo biểu đồ dạng thanh (**sns.barplot**) để so sánh giá trung bình đóng cửa theo từng tháng và từng năm trong khoảng thời gian 2021, 2022 và 2023 từ **closedf**.

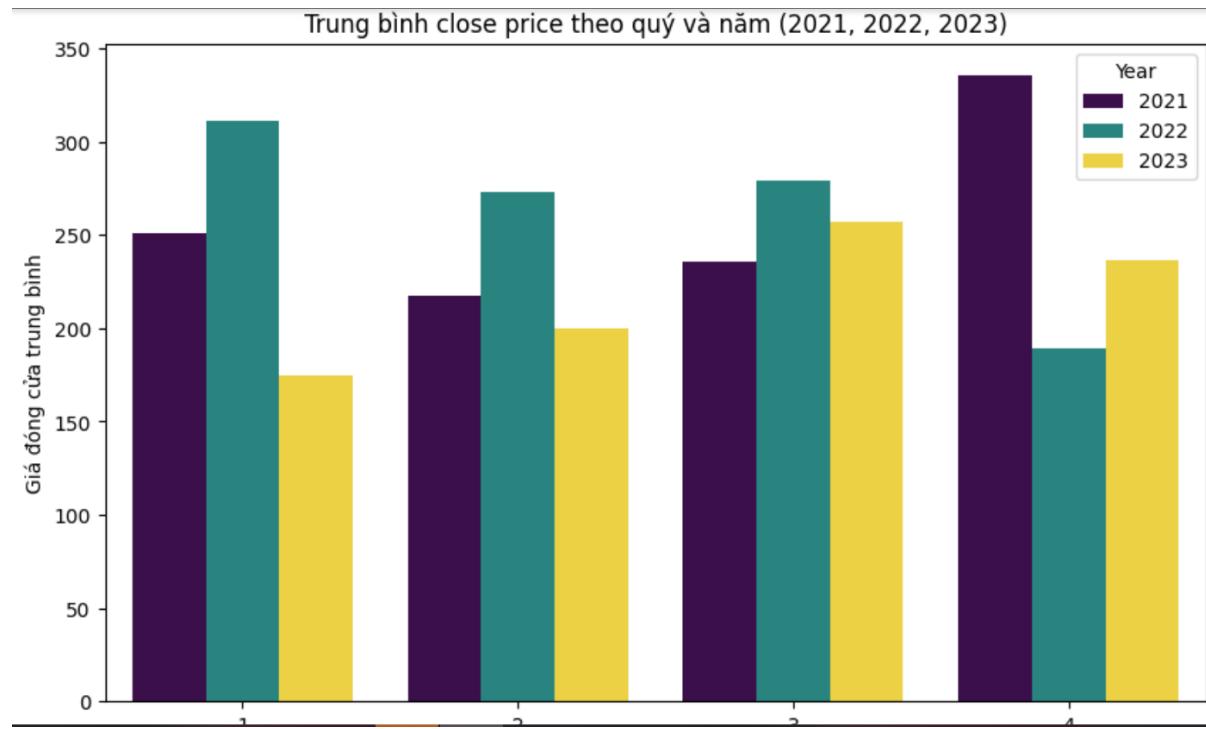
```
# Tạo biểu đồ so sánh giá đóng cửa trung bình theo tháng giữa các năm
closedf['Year'] = closedf['date'].dt.year
closedf['Month'] = closedf['date'].dt.month - 1
years_of_interest = [2021, 2022, 2023]
df_filtered = closedf[closedf['Year'].isin(years_of_interest)]
plt.figure(figsize=(10, 6))
ax = sns.barplot(x='Month', y='close', hue='Year', data=df_filtered.groupby(['Month', 'Year'])['close'].mean().reset_index(), palette=sns.color_palette('Set2'))
sns.regplot(x='Month', y='close', data=df_filtered[df_filtered['Year'] == 2022].groupby(['Month'])['close'].mean().reset_index(),
            ci=None, scatter=False, ax=ax, line_kws={'linestyle': '--', 'color': 'red'}, label='Trendline 2022')
plt.xlabel('Month')
plt.ylabel('Close Price Average')
plt.title('Trung bình giá đóng cửa theo tháng và năm (2021, 2022, 2023)')
plt.legend(title='Year', loc='upper right')
plt.show()
```



Hình 2. 24 Trung bình giá đóng cửa theo tháng các năm 2021,2022,2023

- Tạo biểu đồ dạng thanh (`sns.barplot`) để so sánh giá đóng cửa trung bình theo từng quý và từng năm trong khoảng thời gian 2021, 2022 và 2023 từ dataframe `closeddf`.

```
# Tạo biểu đồ so sánh giá đóng cửa trung bình theo quý giữa các năm
closeddf['Quarter'] = closeddf['date'].dt.quarter
years_of_interest = [2021, 2022, 2023]
df_filtered = closeddf[closeddf['Year'].isin(years_of_interest)]
plt.figure(figsize=(10, 6))
sns.barplot(x='Quarter', y='close', hue='Year', data=df_filtered.groupby(['Quarter', 'Year'])['close'].mean().reset_index(), palette='Set1')
plt.xlabel('Quý')
plt.ylabel('Giá đóng cửa trung bình')
plt.title('Trung bình close price theo quý và năm (2021, 2022, 2023)')
plt.show()
```



Hình 2. 25 Trung bình close price theo quý các năm 2021,2022,2023

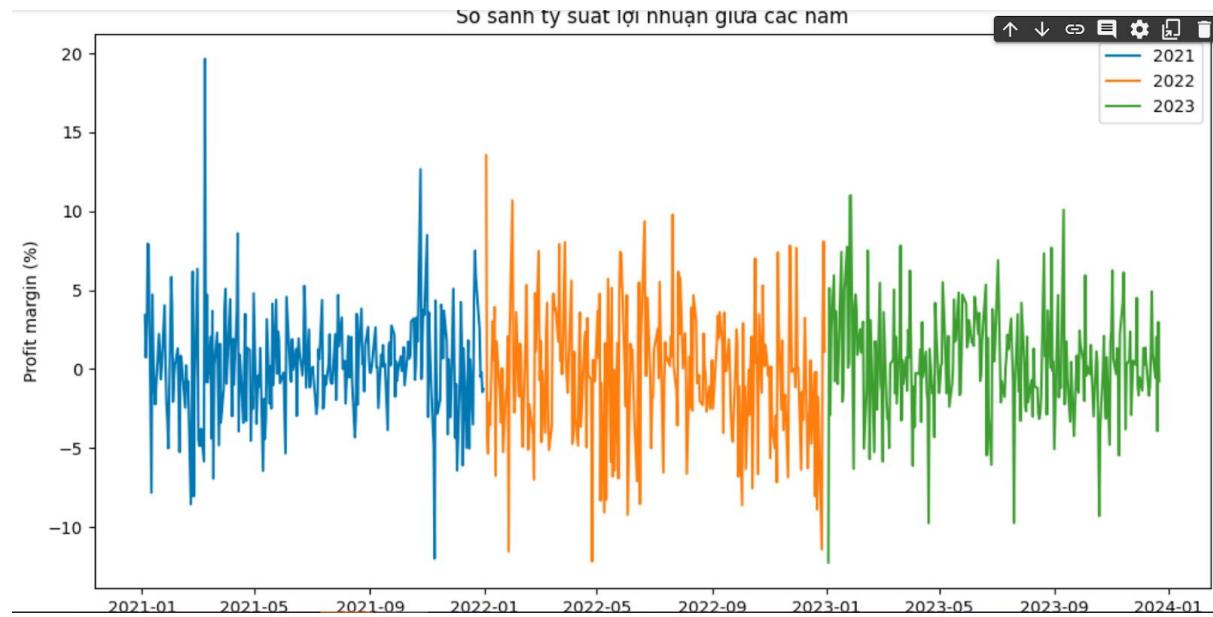
- Đoạn code này tạo biểu đồ đường so sánh tỷ suất lợi nhuận (%) theo ngày giữa các năm 2021, 2022 và 2023 từ DataFrame `closeddf`. Với vòng lặp `for` được sử dụng để vẽ từng đường dữ liệu tỷ suất lợi nhuận (%) cho mỗi năm trên cùng một biểu đồ.



```
# Tính tỷ suất lợi nhuận dựa trên giá đóng cửa
closedf['Return'] = closedf['close'].pct_change() * 100

plt.figure(figsize=(12, 6))
for year in [2021, 2022, 2023]:
    data_year = closedf[closedf['date'].dt.year == year]
    plt.plot(data_year['date'], data_year['Return'], label=str(year))

plt.title('So sánh tỷ suất lợi nhuận giữa các năm')
plt.xlabel('Date')
plt.ylabel('Profit margin (%)')
plt.legend()
plt.show()
```



Hình 2. 26 Biểu đồ so sánh tỷ suất lợi nhuận 2021,2022,2023

3.2 Kết luận các vấn đề thống kê

3.2.1 Về các thông tin cơ bản của dữ liệu

- Một số quan sát rút ra :
 - Tổng số quan sát là 1259.
 - Giá trị cao nhất, thấp nhất của các cột dữ liệu khác nhau đáng kể cho thấy phạm vi giá trị lớn của các biến.
 - Độ lệch chuẩn của từng biến khá lớn, cho thấy mức độ biến động cao của dữ liệu.
 - Giá trị giao dịch lớn, trung bình khoảng 1.3 tỉ.
- Barplot giá trung bình theo tháng:
 - Hiển thị sự phân bố close price và open price theo tháng.
 - Giá trị giữa 2 biến không có sự chênh lệch đáng kể.
- Barplot giá trung bình theo quý:
 - Hiển thị sự phân bố close price và open price theo quý.
 - Giá trị giữa 2 biến cũng không có sự chênh lệch đáng kể.



- Line Plot giá trung bình thay đổi theo năm:
 - Hiển thị 4 biến close, open, high, low theo năm.
 - Có sự tương quan thuận cao giữa các biến close, open, high và low, nhưng có sự tương quan nghịch yếu với volume.
 - Heatmap và pairplot:
 - Cho thấy mối quan hệ và tính tương quan giữa 5 biến.
 - Các biến close, open, high và low có mối quan hệ tuyến tính với nhau.
 - Volume không có sự tương quan chặt chẽ với bất kỳ biến nào.
- 3.2.3 Về cổ phiếu năm gần nhất (2023)
- Giá cổ phiếu năm 2023 có xu hướng tăng trưởng đi lên từ đầu năm đến cuối năm.
 - Thông qua việc xem xét barplot theo quý, quý 3 chứng kiến sự tăng trưởng mạnh nhất của cổ phiếu.
- 3.2.4 Về phân phối dữ liệu
- Giá cổ phiếu chủ yếu nằm trong khoảng từ 150-300.
 - Thông qua Skewness và histogram có thể thấy giá cổ phiếu lệch phải nhẹ nhưng không đáng kể.
- 3.2.5 Về việc so sánh các chỉ số các năm 2021, 2022, 2023
- Với barplot : Các năm 2021 và 2023 đều có xu hướng tăng trưởng vào cuối năm. Năm 2022 lại có sự sụt giảm giá cổ phiếu vào cuối năm.
 - Với Profit Plot : Năm 2022 có biến động lợi nhuận cao so với các năm khác.

4. Phân tích chuỗi thời gian

4.1 Xác định tính mùa vụ, xu hướng và chu kỳ

- Đoạn mã trên tạo ra một DataFrame mới có tên là closedfcopy, chỉ chứa các cột 'date' và 'close' từ DataFrame gốc df. Điều này là bước chuẩn bị dữ liệu cho việc thực hiện các phân tích, trực quan hóa, mô hình hóa dữ liệu liên quan đến giá đóng cửa trong thời gian cụ thể.

```
[ ] #tạo ra một DataFrame mới có tên là closedfcopy  
closedfcopy = df[['date','close']].copy()
```

Hình 2. 27 Dataframe closedfcopy

Kết quả trả về của dòng mã này là một DataFrame mới (closedfcopy) chứa các cột 'date' và 'close' từ DataFrame gốc (df), và nó được sử dụng để tiếp tục xử lý dữ liệu mà không làm thay đổi dữ liệu trong DataFrame gốc.



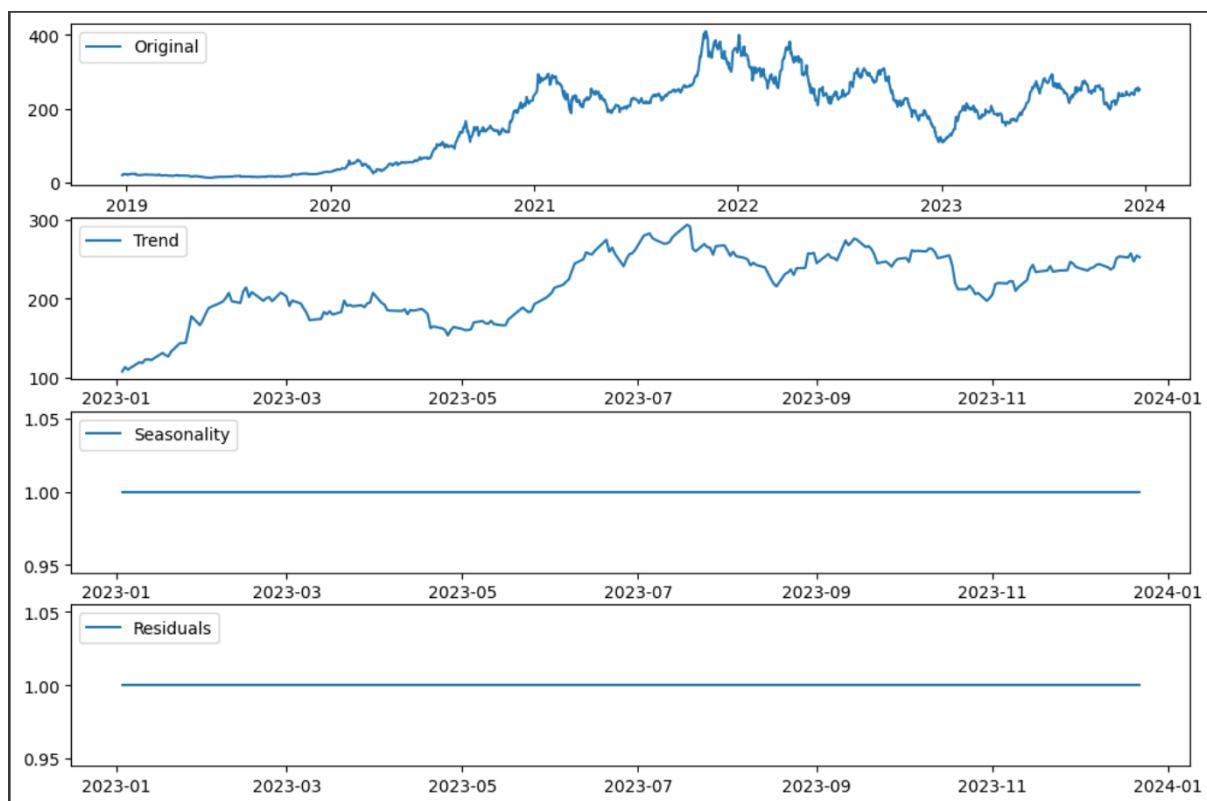
- Được sử dụng để thiết lập cột 'date' làm chỉ số (index) cho DataFrame closedfcopy trong thư viện pandas của Python: Dễ dàng Truy cập Dữ liệu theo Chỉ số Thời gian, Tính năng Resampling và Thống kê thời gian, Hiển thị và Trực quan hóa dữ liệu.

```
[ ] # Đặt cột date làm index  
closedfcopy = closedfcopy.set_index('date')
```

Hình 2. 28 Đặt cột date làm index

- Vẽ biểu đồ các yếu tố thời gian năm 2023:

```
# Vẽ biểu đồ các yếu tố xu hướng năm 2023  
result = seasonal_decompose(closedfcopy['close'][closedfcopy.index > '2023-01-01'], model='multiplicative', period=1)  
  
plt.figure(figsize=(12, 8))  
  
plt.subplot(4, 1, 1)  
plt.plot(closedfcopy['close'], label='Original')  
plt.legend(loc='upper left')  
  
plt.subplot(4, 1, 2)  
plt.plot(result.trend, label='Trend')  
plt.legend(loc='upper left')  
  
plt.subplot(4, 1, 3)  
plt.plot(result.seasonal, label='Seasonality')  
plt.legend(loc='upper left')  
  
plt.subplot(4, 1, 4)  
plt.plot(result.resid, label='Residuals')  
plt.legend(loc='upper left')  
plt.show()
```



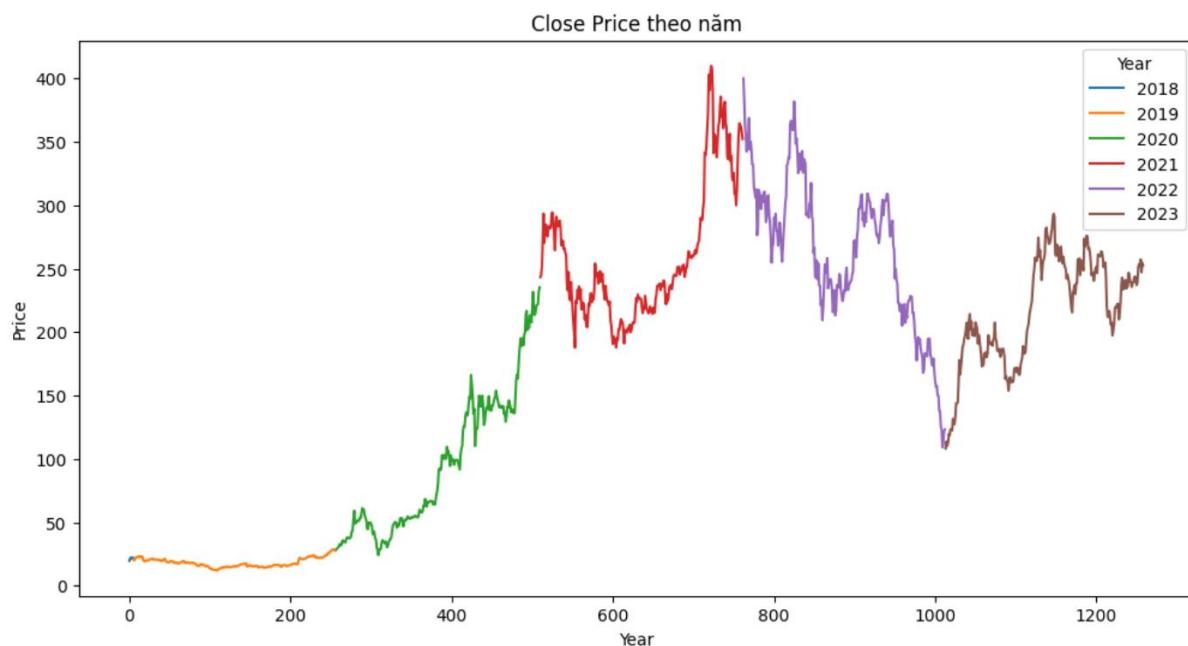
Hình 2. 29 Seasonal decompose 2023

Đoạn mã này thực hiện phân tích thành phần chuỗi thời gian (original, trend, seasonality, residuals) cho dữ liệu giá đóng cửa ('close') từ một ngày cụ thể trở đi và hiển thị kết quả bằng các đồ thị đồng thời.

- Vẽ biểu đồ giá đóng cửa theo năm:



```
#Vẽ biểu đồ giá close theo năm  
df.groupby(df['date'].dt.year)['close'].plot(figsize=(12,6))  
plt.xlabel('Year')  
plt.ylabel('Price')  
plt.title('Close Price theo năm')  
plt.legend(title='Year')  
plt.show()
```



Hình 2. 30 Biểu đồ giá đóng cửa theo năm

Đoạn mã này được sử dụng để tạo một biểu đồ thể hiện giá đóng cửa theo năm, giúp quan sát xu hướng và biến động giá của tài sản trong mỗi năm.

Kết quả trả về của đoạn mã này là một biểu đồ đường thể hiện giá đóng cửa theo năm, giúp phân tích xu hướng và biến động giá trong từng năm của dữ liệu tài sản.

- Vẽ biểu đồ các yếu tố thời gian qua các năm(dài hạn):



```
# Vẽ biểu đồ các yếu tố xu hướng qua các năm
result = seasonal_decompose(closedfcopy['close'], model='multiplicative', period=1)

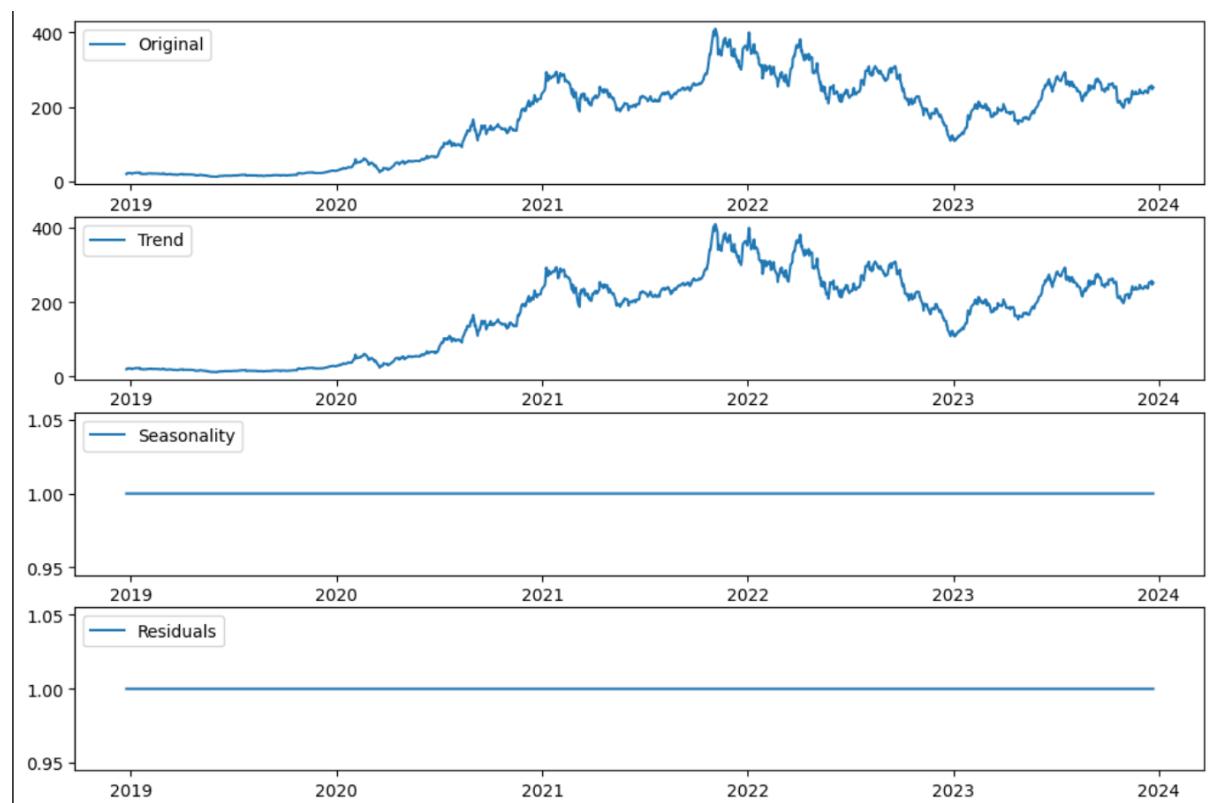
plt.figure(figsize=(12, 8))

plt.subplot(4, 1, 1)
plt.plot(closedfcopy['close'], label='Original')
plt.legend(loc='upper left')

plt.subplot(4, 1, 2)
plt.plot(result.trend, label='Trend')
plt.legend(loc='upper left')

plt.subplot(4, 1, 3)
plt.plot(result.seasonal, label='Seasonality')
plt.legend(loc='upper left')

plt.subplot(4, 1, 4)
plt.plot(result.resid, label='Residuals')
plt.legend(loc='upper left')
```



Hình 2. 31 Seasonal Decompose các năm

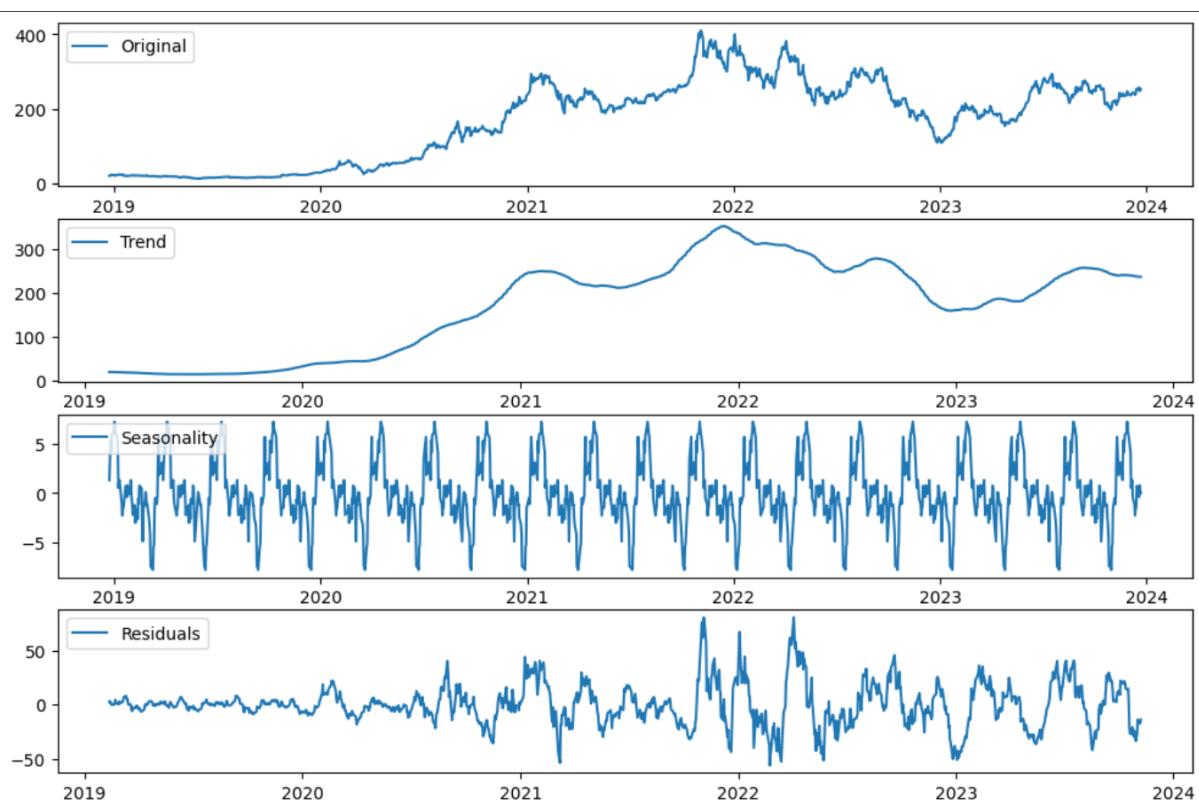


Đoạn mã này thực hiện phân tích thành phần chuỗi thời gian của giá đóng cửa qua các năm và vẽ biểu đồ cho các thành phần xu hướng, mùa vụ và phần dư, giúp hiểu rõ hơn về cấu trúc thời gian của dữ liệu.

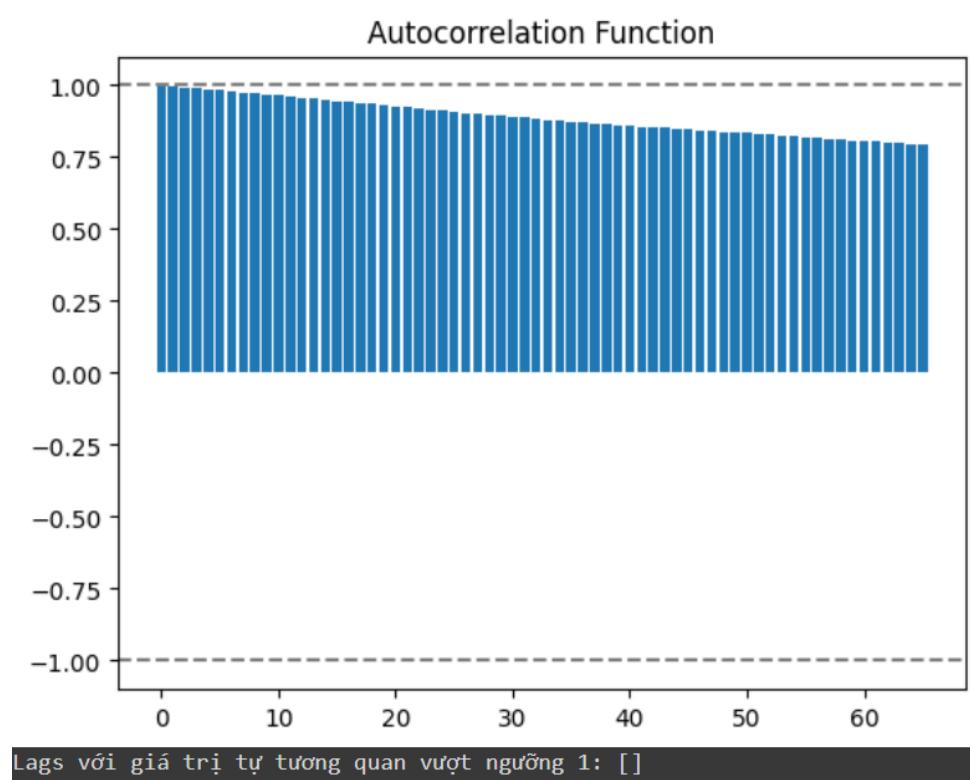
Đoạn mã này sử dụng các phương thức và module từ thư viện statsmodels và matplotlib để thực hiện phân tích thành phần chuỗi thời gian và vẽ biểu đồ cho các thành phần xu hướng, mùa vụ, và phần dư của dữ liệu giá đóng cửa.

- Vẽ biểu đồ các yếu tố mùa vụ qua các năm:

```
[ ] # Vẽ biểu đồ các yếu tố mùa vụ theo quý qua các năm
result = seasonal_decompose(closedfcopy['close'], model='additive', period=65)
plt.figure(figsize=(12, 8))
plt.subplot(4, 1, 1)
plt.plot(closedfcopy['close'], label='Original')
plt.legend(loc='upper left')
plt.subplot(4, 1, 2)
plt.plot(result.trend, label='Trend')
plt.legend(loc='upper left')
plt.subplot(4, 1, 3)
plt.plot(result.seasonal, label='Seasonality')
plt.legend(loc='upper left')
plt.subplot(4, 1, 4)
plt.plot(result.resid, label='Residuals')
plt.legend(loc='upper left')
plt.show()
# Tính giá trị tự tương quan
acf_values, conf_int = acf(closedfcopy['close'], nlags=65, alpha=0.05)
plt.bar(range(len(acf_values)), acf_values)
plt.axhline(y=1, linestyle='--', color='gray') # Vẽ đường ngưỡng 1
plt.axhline(y=-1, linestyle='--', color='gray') # Vẽ đường ngưỡng -1
plt.title('Autocorrelation Function')
plt.show()
# Tìm các lag khi giá trị tự tương quan vượt ngưỡng 1
lags_exceeding_threshold = np.where(np.abs(acf_values) > 1)[0]
print("Lags với giá trị tự tương quan vượt ngưỡng 1:", lags_exceeding_threshold)
```



Hình 2. 32 Seasonal Decompose có yếu tố mùa vụ các năm



Hình 2. 33 Autocorrelation

Đoạn mã này thực hiện một loạt các phân tích thống kê và vẽ biểu đồ để hiểu rõ hơn về thành phần mùa vụ và tự tương quan trong dữ liệu giá đóng cửa ('close'). Thực hiện phân tích thành phần chuỗi thời gian và kiểm tra tự tương quan để hiểu rõ hơn về mô hình chuỗi thời gian và các yếu tố ảnh hưởng.

Đoạn mã này sử dụng các phương thức và module từ thư viện statsmodels để thực hiện phân tích thành phần chuỗi thời gian và kiểm tra tự tương quan, giúp hiểu rõ hơn về cấu trúc và đặc điểm của chuỗi thời gian.

Nhận xét về trung hạn và dài hạn:

- Về trung hạn (2023): Cho thấy có yếu tố xu hướng trong close price năm 2023 ==> Xu hướng tăng nhưng không rõ ràng.
- Về dài hạn: Yếu tố mùa vụ không rõ ràng. Close Price có xu hướng tăng mạnh 2 lần vào cuối năm 2020 và cuối 2021. Ngược lại, Close Price cũng có đợt giảm mạnh là đầu 2021 đến cuối 2021, đầu 2022 đến đầu 2023.



4.2 So sánh các mã cổ phiếu

- Tải dữ liệu 5 mã cổ phiếu bằng yfinance

```
[ ] # Tên 5 mã cổ phiếu
tickers = ['F', 'VOW.DE', 'TM', 'TSLA', 'BMW.DE']
# Lấy dữ liệu từ Yahoo Finance
data = yf.download(tickers, start='2018-12-27', end='2023-12-27')

[          0%] [*****100%*****] 5 of 5 completed
```

Hình 2. 34 Tải dữ liệu 5 mã cổ phiếu

Đoạn mã này được sử dụng để lấy dữ liệu giá cổ phiếu từ Yahoo Finance cho 5 mã cổ phiếu khác nhau trong khoảng thời gian từ '2018-12-27' đến '2023-12-27'.

- Kiểm tra dữ liệu:

```
[ ] # Kiểm tra dữ liệu thiếu
print("Null values", data.isnull().values.sum())
print("NA values:", data.isna().values.any())
```

Hình 2. 35 Kiểm tra dữ liệu từ 5 mã cổ phiếu

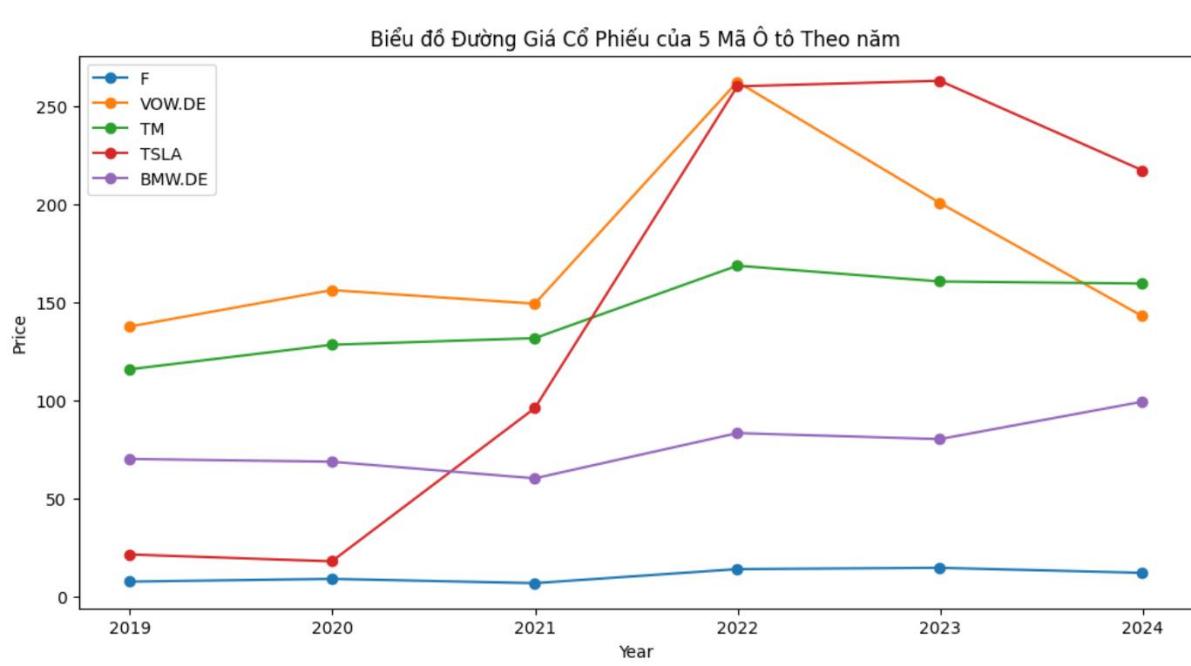
Đoạn mã này được sử dụng để kiểm tra và báo cáo về sự xuất hiện của giá trị thiêu (null hoặc NaN) trong dữ liệu giá cổ phiếu.

a) Lập biểu đồ đường để so sánh dữ liệu 5 mã:

```
[ ] #Gộp dữ liệu theo tháng và vẽ biểu đồ đường cho giá cổ phiếu của 5 mã ô tô theo tháng
plt.figure(figsize=(12, 6))

for ticker in tickers:
    data_monthly = data['Close'][ticker].resample('Y').mean()
    plt.plot(data_monthly, label=ticker, marker='o')

plt.title('Biểu Đồ Đường Giá Cổ Phiếu của 5 Mã Ô tô Theo năm')
plt.xlabel('Year')
plt.ylabel('Price')
plt.legend()
plt.show()
```



Hình 2. 36 Biểu đồ đường giá cổ phiếu 5 hãng ô tô theo năm

Đoạn mã này gộp dữ liệu theo tháng và vẽ biểu đồ đường cho giá cổ phiếu của 5 mã ô tô theo từng năm để so sánh mã cổ phiếu trung bình của 5 năm 2019 - 2023.

- Nhân xét:

- Giống nhau:

- Từ đầu năm 2020 đến cuối năm 2021 các mã cổ phiếu đều có xu hướng tăng trưởng, tăng nhanh nhất là Tesla và WOW
- Từ cuối năm 2021 đến 2023, các mã cổ phiếu ô tô có xu hướng giảm so với đỉnh năm 2021.

- Khác nhau:

- Đối với Tesla, năm 2022 chưa chứng kiến xu hướng giảm, còn đối với các mã khác đã có sự sụt giảm so với đỉnh là cuối 2021.
- BMW và TOYOTA có sự tăng trưởng trở lại vào đầu năm 2023, các mã khác vẫn tiếp tục giảm.

- b) Biểu đồ số lượng cổ phiếu giao dịch theo từng ngày:



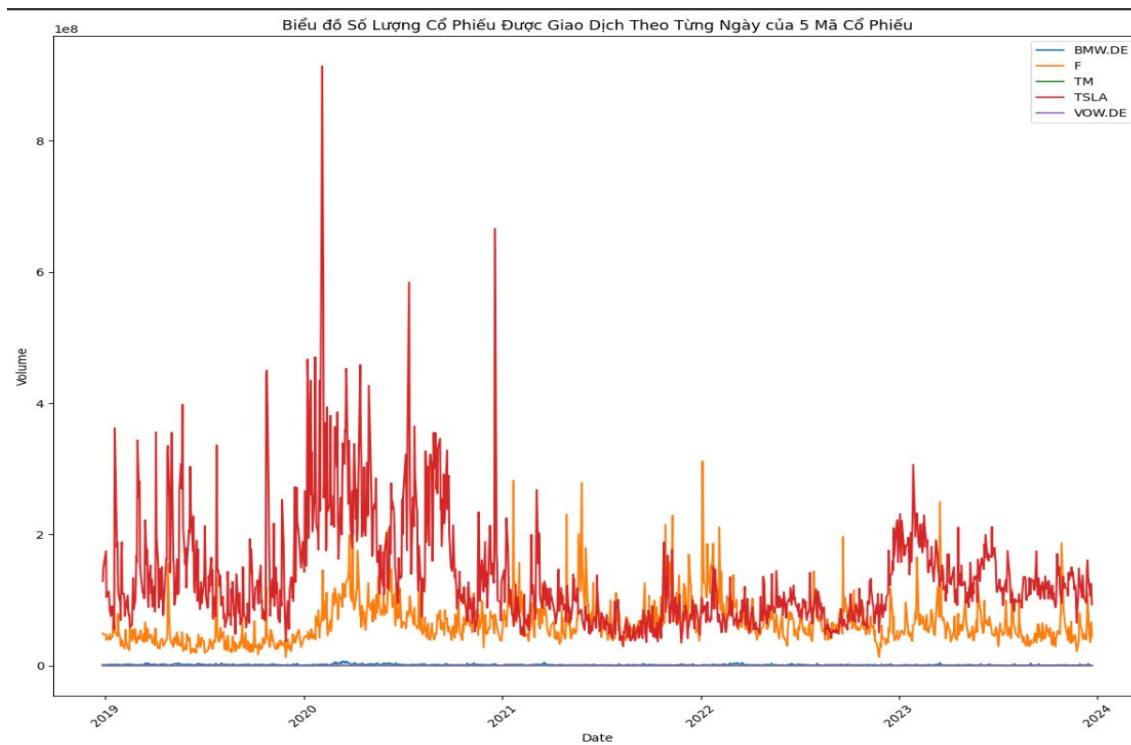
[]

```
# Lấy dữ liệu số lượng cổ phiếu được giao dịch của từng mã
volume_traded_tickers = data['Volume']

# Vẽ biểu đồ đường cho từng mã cổ phiếu
plt.figure(figsize=(12, 10))

for ticker in volume_traded_tickers.columns:
    plt.plot(data.index, data['Volume'][ticker], label=ticker)

plt.title('Biểu đồ Số Lượng Cổ Phiếu Được Giao Dịch Theo Từng Ngày của 5 Mã Cổ Phiếu')
plt.xlabel('Date')
plt.ylabel('Volume')
plt.legend()
# Xoay nhãn trục x để dễ đọc
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



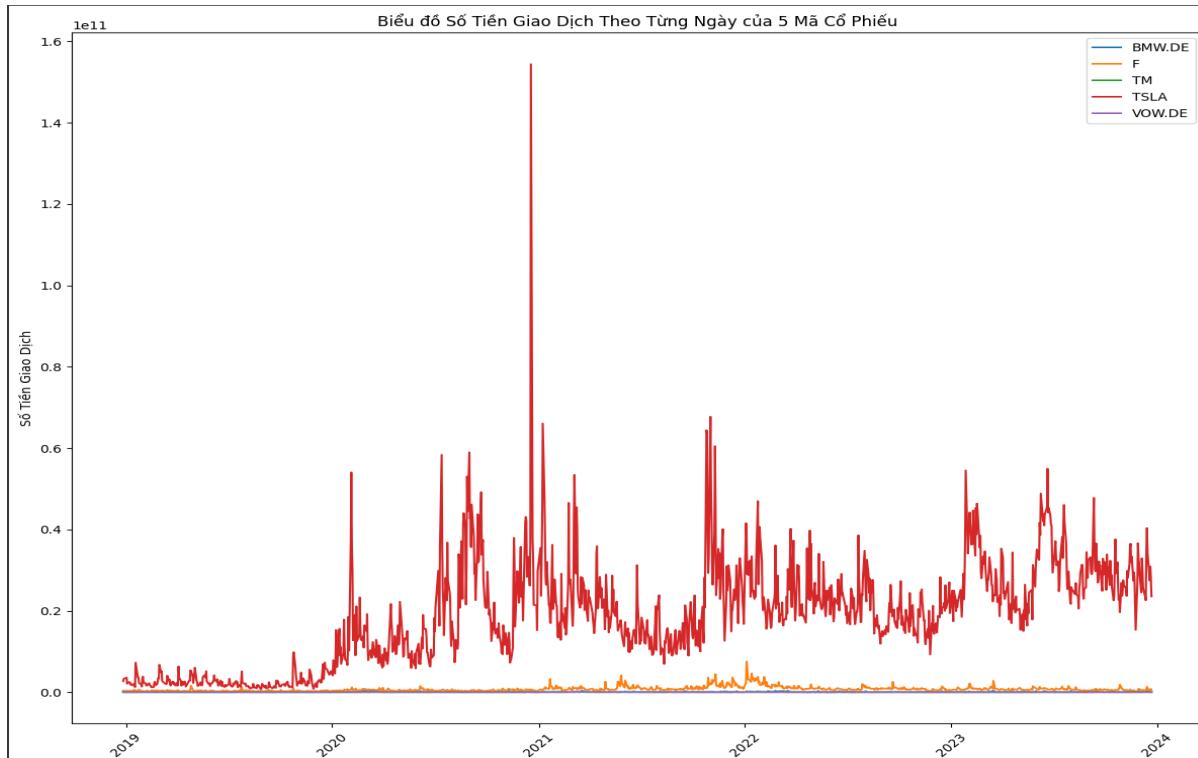
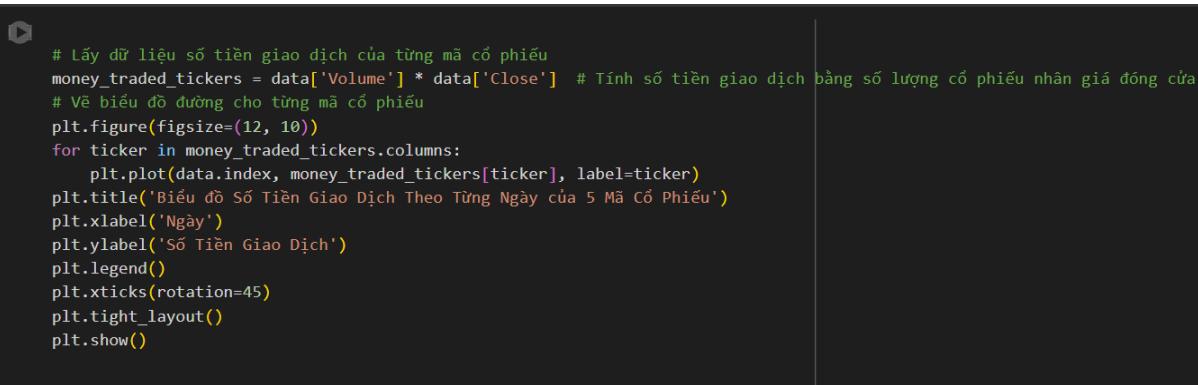
Hình 2. 37 Biểu đồ số lượng cổ phiếu được giao dịch theo từng ngày

Đoạn mã này lấy dữ liệu về số lượng cổ phiếu được giao dịch của từng mã cổ phiếu và vẽ biểu đồ đường cho số lượng cổ phiếu được giao dịch theo từng ngày để so sánh lượng giao dịch của 5 mã.

- Nhận xét:



- Tesla và VOW có mức độ biến động giao dịch cao, đặc biệt là trong 2 năm 2020, 2021. Các mã còn lại có mức biến động thấp.
 - Từ cuối năm 2021, mức độ biến động giao dịch bắt đầu giảm xuống.
- c) *Biểu đồ số tiền giao dịch theo từng ngày*



Hình 2. 38 Biểu đồ số tiền giao dịch theo từng ngày của 5 mã cổ phiếu

Đoạn mã trên được sử dụng để lấy dữ liệu về số tiền giao dịch (số tiền được trao đổi khi mua bán cổ phiếu) của từng mã cổ phiếu và sau đó vẽ biểu đồ đường cho số tiền giao dịch theo từng ngày để so sánh số tiền giao dịch.



- Nhận xét:

- Về tổng giá trị giao dịch, Tesla có tổng giá trị cao nhất, xếp dưới là VOW, sau đó là các mã còn lại.
- Tổng giá trị giao dịch đang còn biến động trong vòng 5 năm qua.

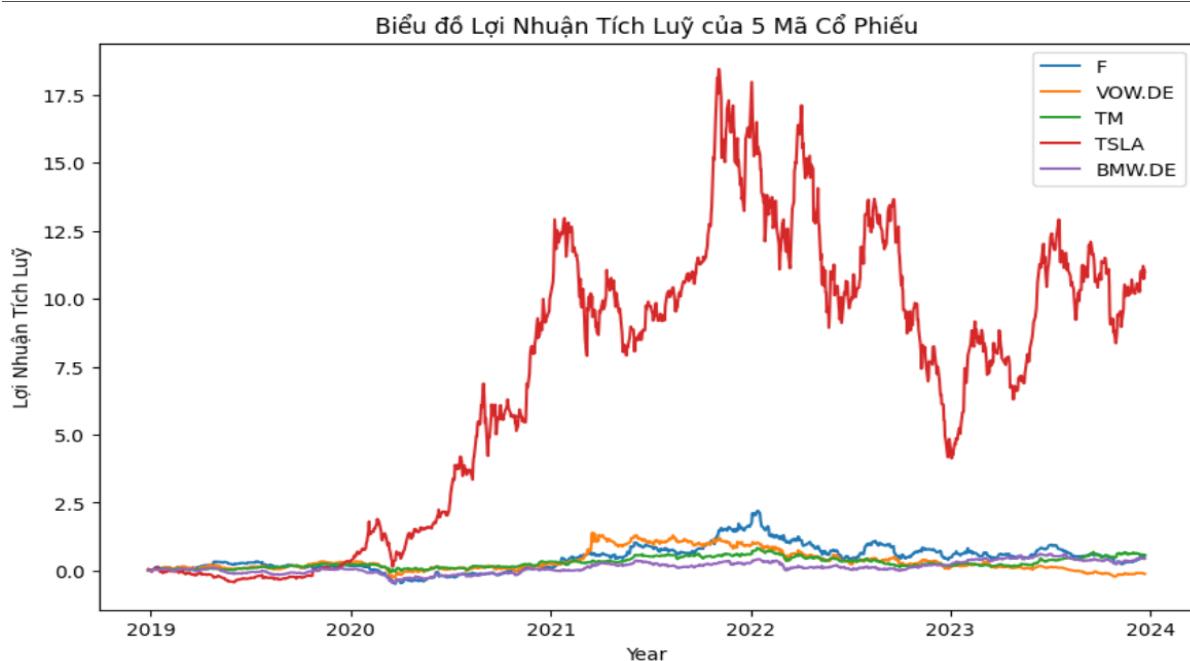
d) *Biểu đồ lợi nhuận tích lũy của 5 mã cổ phiếu*

```
[ ] # Tính toán lợi nhuận từ dữ liệu giá cổ phiếu
returns = data['Close'].pct_change() # Tính toán tỷ lệ biến đổi
cumulative_returns = (1 + returns).cumprod() - 1 # Tính toán lợi nhuận tích luỹ

# Vẽ biểu đồ lợi nhuận tích luỹ của 5 mã cổ phiếu
plt.figure(figsize=(10, 6))

for ticker in tickers:
    plt.plot(cumulative_returns[ticker], label=ticker)

plt.title('Biểu đồ Lợi Nhuận Tích Luỹ của 5 Mã Cổ Phiếu')
plt.xlabel('Year')
plt.ylabel('Lợi Nhuận Tích Luỹ')
plt.legend()
plt.show()
```



Hình 2. 39 Biểu đồ lợi nhuận tích lũy của 5 mã cổ phiếu

Đoạn mã này tính toán và vẽ biểu đồ lợi nhuận tích luỹ của 5 mã cổ phiếu theo thời gian. Lợi nhuận tích luỹ thường được sử dụng để theo dõi sự biến động của giá cổ phiếu theo thời gian để so sánh lợi nhuận tích lũy của các năm.



- Nhận xét:

- Về lợi nhuận tích lũy, Tesla có sự biến động cao qua các năm.
- Xếp dưới là VOW, tuy nhiên từ cuối năm 2022, Ford có sự vươn lên thay thế vị trí của VOW.

5. Mô hình

5.1 Moving average naive

- Tạo 1 dataframe mới có tên là **closedf** với hai cột là date và close.

```
[ ] closedf = df[['date', 'close']].copy()
```

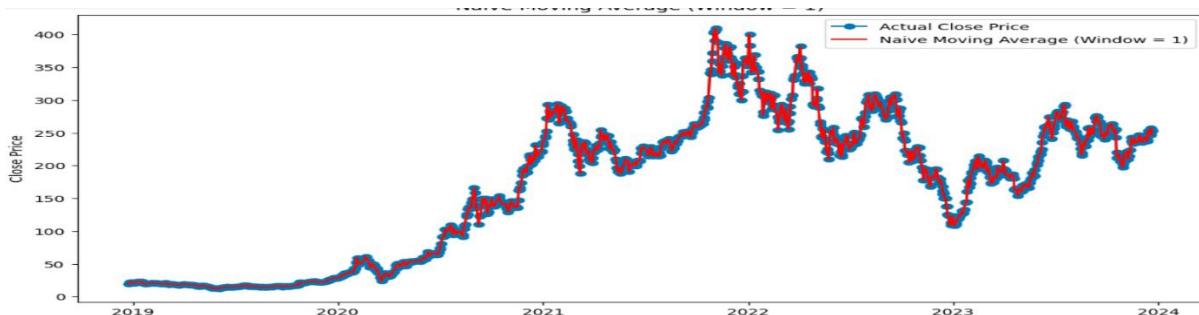
Hình 2. 40 Tạo dataframe mới là closedf

- Vẽ biểu đồ cho Naive Moving Average Với cửa sổ (window_size = 1) từ dữ liệu trong **closedf** đã lưu trước đó. Hiển thị giá đóng cửa thực tế và giá dự đoán với step = 1 . Tính toán và in ra các chỉ số cần như sai số trung bình tuyệt đối (MAE), tỷ lệ sai số trung bình tuyệt đối (MAPE), sai số bình phương trung bình (MSE), và điểm R2 được tính toán và hiển thị ra màn hình.

```
window_size = 1

# Tính trung bình động naive
closedf['MA_Naive'] = closedf['close'].rolling(window=window_size).mean()
plt.figure(figsize=(12, 6))
plt.plot(closedf['date'], closedf['close'], marker='o', linestyle='-' )
plt.plot(closedf['date'], closedf['MA_Naive'], label=f'Naive Moving Average (Window = {window_size})', linestyle='--', color='red')
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.title(f'Naive Moving Average (Window = {window_size})')
plt.legend()
plt.show()

# Các chỉ số đánh giá:
mae_naive = mean_absolute_error(closedf['close'],closedf['MA_Naive'])
mape_naive = mean_absolute_percentage_error(closedf['close'],closedf['MA_Naive'])
mse_naive = mean_squared_error(closedf['close'],closedf['MA_Naive'])
r2_naive = r2_score(closedf['close'],closedf['MA_Naive'])
print("mean_absolute_error:",mae_naive)
print("mean_absolute_percentage_error",mape_naive)
print("mean_squared_error:",mse_naive)
print("r2_score:",r2_naive)
```



Hình 2. 41 Moving average naive

- Các chỉ số đánh giá

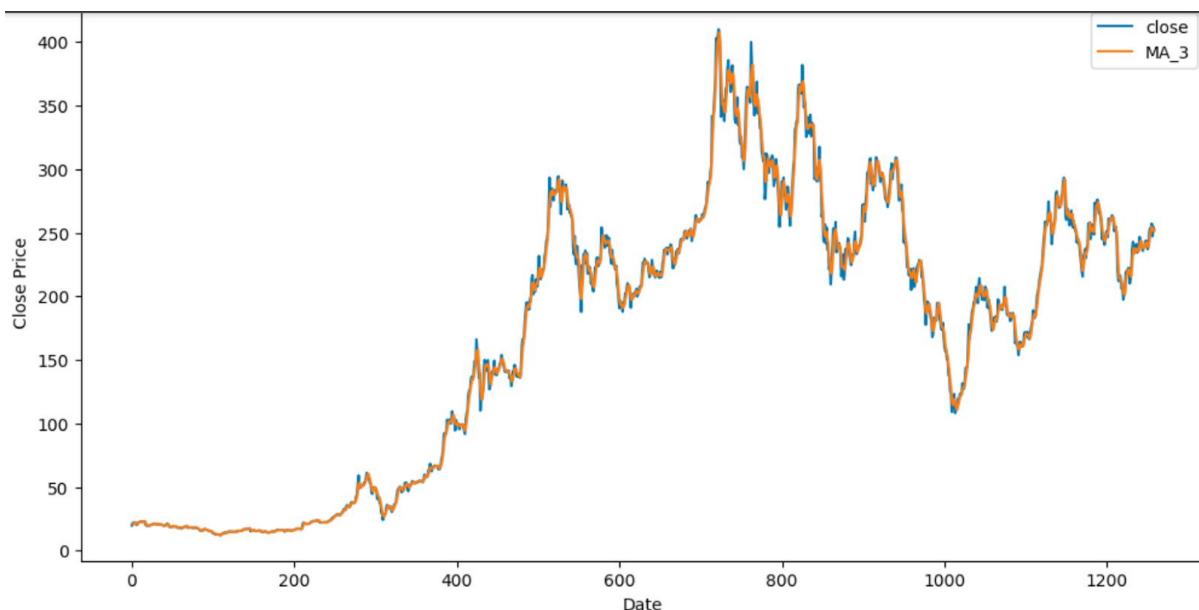


```
mean_absolute_error: 0.0
mean_absolute_percentage_error 0.0
mean_squared_error: 0.0
r2_score: 1.0
```

5.2 Moving average khoảng trượt 3

- Vẽ biểu đồ cho Naive Moving Average Với cửa sổ (window_size = 3) từ dữ liệu trong **closedf** đã lưu trước đó. Hiển thị giá đóng cửa thực tế và giá dự đoán với step = 3. Tính toán và in ra các chỉ số cần như sai số trung bình tuyệt đối (MAE), tỷ lệ sai số trung bình tuyệt đối (MAPE), sai số bình phương trung bình (MSE), và điểm R2 được tính toán và hiển thị ra màn hình.

```
window_size = 3
closedf['MA_3'] = closedf['close'].rolling(window=window_size).mean()
closedf['close'].plot(figsize=(12,6),legend = TRUE)
closedf['MA_3'].plot(figsize=(12,6),legend = TRUE)
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.title(f'3-Steps Moving Average Model (Window = {window_size})')
plt.legend()
plt.show()
# Chỉ số đánh giá
mae_3mva = mean_absolute_error(closedf['close'][2:],closedf['MA_3'].dropna())
mape_3mva = mean_absolute_percentage_error(closedf['close'][2:],closedf['MA_3'].dropna())
mse_3mva = mean_squared_error(closedf['close'][2:],closedf['MA_3'].dropna())
r2_3mva = r2_score(closedf['close'][2:],closedf['MA_3'].dropna())
print("mean_absolute_error:",mae_3mva)
print("mean_absolute_percentage_error",mape_3mva)
print("mean_square_error:",mse_3mva)
print("r2_score:",r2_3mva)
```



Hình 2. 42 Moving average khoảng trượt 3

- Các chỉ số

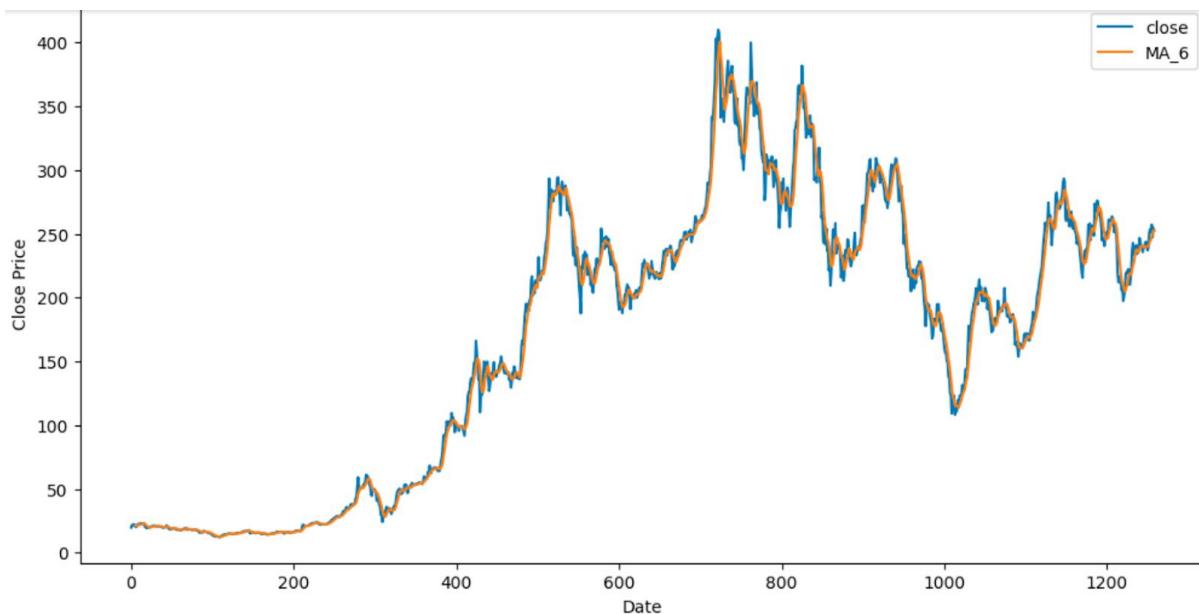
```
mean_absolute_error: 3.6621371797931586
mean_absolute_percentage_error 0.02198101895217465
mean_squared_error: 31.664890763045435
r2_score: 0.9973073968630725
```



5.3 Moving average khoảng trượt 6

- Vẽ biểu đồ cho Naive Moving Average Với cửa sổ (window_size = 6) từ dữ liệu trong **closedf** đã lưu trước đó. Hiển thị giá đóng cửa thực tế và giá dự đoán với step = 6. Tính toán và in ra các chỉ số cần như sai số trung bình tuyệt đối (MAE), tỷ lệ sai số trung bình tuyệt đối (MAPE), sai số bình phương trung bình (MSE), và điểm R2 được tính toán và hiển thị ra màn hình.

```
window_size = 6
closedf['MA_6'] = closedf['close'].rolling(window=window_size).mean()
closedf['close'].plot(figsize=(12,6),legend = TRUE)
closedf['MA_6'].plot(figsize=(12,6),legend = TRUE)
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.title(f'6-Steps Moving Average Model (Window = {window_size})')
plt.legend()
plt.show()
# Chỉ số đánh giá
mae_6mva = mean_absolute_error(closedf['close'][5:],closedf['MA_6'].dropna())
mape_6mva = mean_absolute_percentage_error(closedf['close'][5:],closedf['MA_6'].dropna())
mse_6mva = mean_squared_error(closedf['close'][5:],closedf['MA_6'].dropna())
r2_6mva = r2_score(closedf['close'][5:],closedf['MA_6'].dropna())
print("mean_absolute_error:",mae_6mva)
print("mean_absolute_percentage_error:",mape_6mva)
print("mean_square_error:",mse_6mva)
print("r2_score:",r2_6mva)
```



Hình 2. 43 Moving average khoảng trượt 6

- Các chỉ số:

```
mean_absolute_error: 6.24197776076555
mean_absolute_percentage_error 0.037729482152751896
mean_square_error: 87.38094104623815
r2_score: 0.9925539826800457
```

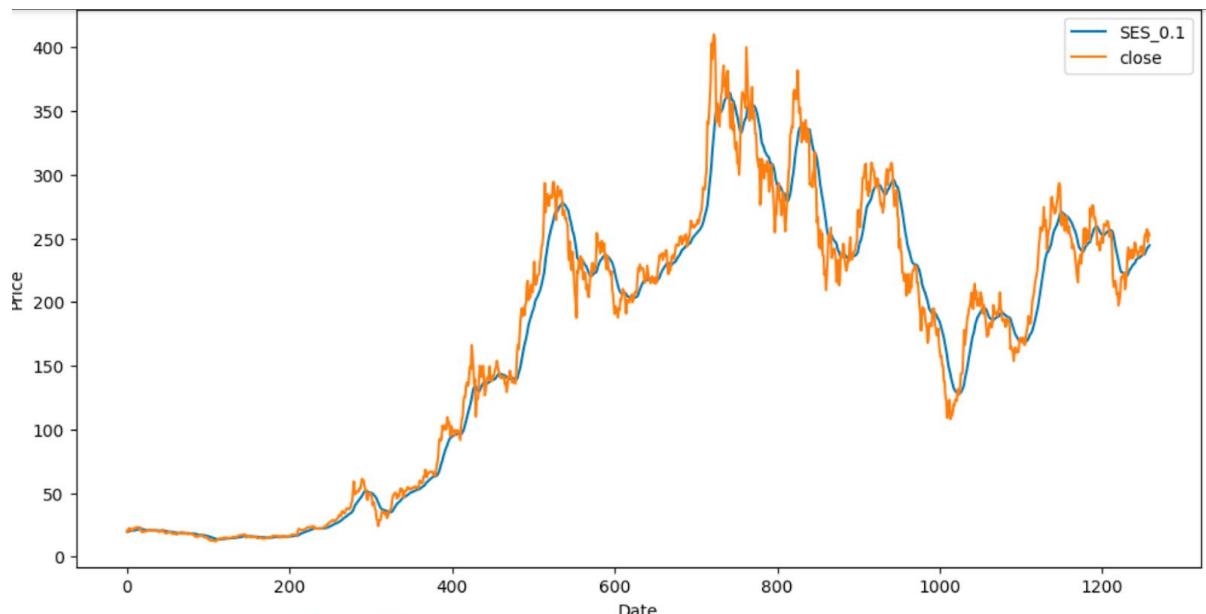


5.4 Simple Exponential Smoothing với alpha=0.1

- Đoạn mã trước tiên sử dụng Mô hình **Simple Exponential Smoothing** để dự đoán giá đóng cửa trong DataFrame closedf. Được sử dụng để mô hình hóa xu hướng của dữ liệu thời gian với mức độ trọng số alpha = 0.1. Vẽ biểu đồ so sánh giữa giá đóng cửa thực tế và giá dự đoán từ mô hình SES. Tính toán và hiển thị các chỉ số đánh giá như sai số trung bình tuyệt đối (MAE), tỷ lệ sai số trung bình tuyệt đối (MAPE), sai số bình phương trung bình (MSE) và điểm R2.

```
model = SimpleExpSmoothing(closedf['close']).fit(smoothing_level=0.1)
closedf['SES_0.1'] = model.fittedvalues
closedf['SES_0.1'].plot(figsize=(12,6),legend = TRUE)
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Simple Exponential Smoothing (Alpha = 0.1)')
closedf['close'].plot(figsize=(12,6),legend = TRUE)
plt.show()

# Chỉ số đánh giá
mae_alpha01 = mean_absolute_error(closedf['close'],closedf['SES_0.1'])
mape_alpha01 = mean_absolute_percentage_error(closedf['close'],closedf['SES_0.1'])
mse_alpha01 = mean_squared_error(closedf['close'],closedf['SES_0.1'])
r2_alpha01 = r2_score(closedf['close'],closedf['SES_0.1'])
print("mean_absolute_error:",mae_alpha01)
print("mean_absolute_percentage_error",mape_alpha01)
print("mean_square_error:",mse_alpha01)
print("r2_score:",r2_alpha01)
```



Hình 2. 44 Simple Exponential alpha 0.1

- Các chỉ số

```
mean_absolute_error: 12.078178514499019
mean_absolute_percentage_error 0.07567013596587005
mean_square_error: 326.2479724949026
r2_score: 0.9722972739782327
```



5.5 Simple Exponential Smoothing với alpha tối ưu

- Sử dụng hàm được đặt tên là **objective** dùng để tạo các thử nghiệm trong việc tối ưu hóa tham số alpha cho mô hình simple Exponential Smoothing. Hàm này nhận một đối tượng **trial** từ thư viện **Optuna** để thử nghiệm các giá trị alpha từ 0.01 đến 0.99. Trong mỗi lần thử nghiệm, nó sẽ sử dụng giá trị alpha được đề xuất để tạo và đánh giá mô hình SES với dữ liệu từ cột 'close' trong df **closedf**.

```
# Tạo hàm generate
def objective(trial):
    alpha = trial.suggest_float('alpha', 0.01, 0.99)
    model = SimpleExpSmoothing(closedf['close'].fit(smoothing_level=alpha))
    predictions = model.fittedvalues
    MAE = mean_absolute_error(closedf['close'], predictions.dropna())
    return MAE
```

Hình 2. 45 Hàm tối ưu SES alpha tối ưu

- Sử dụng thư viện Optuna để tối ưu hóa tham số alpha cho Mô hình Bình phương Mượt Đơn giản (SES). Nó sẽ thực hiện 100 lượt thử nghiệm để tìm giá trị alpha tối ưu, dựa trên việc tối thiểu hóa sai số trung bình tuyệt đối (MAE). Kết quả là giá trị alpha tối ưu và dự đoán 30 giá trị tiếp theo từ mô hình SES được hiển thị và lưu vào dataframe **closedf**. Kết quả là giá trị alpha tối ưu và dự đoán 30 giá trị tiếp theo từ mô hình SES được hiển thị và lưu vào dataframe **closedf**.

```
# Tìm alpha tối ưu bằng trình tối ưu optuna
study = optuna.create_study(direction='minimize')
study.optimize(objective, n_trials=100)
best_alpha_optuna = study.best_params['alpha']
mae_best_ses = study.best_value
optimal_model_optuna = SimpleExpSmoothing(closedf['close'].fit(smoothing_level=best_alpha_optuna))
forecast_values = optimal_model_optuna.forecast(steps=30)
print(forecast_values)
closedf['SES_Optimal_Optuna'] = optimal_model_optuna.fittedvalues
```

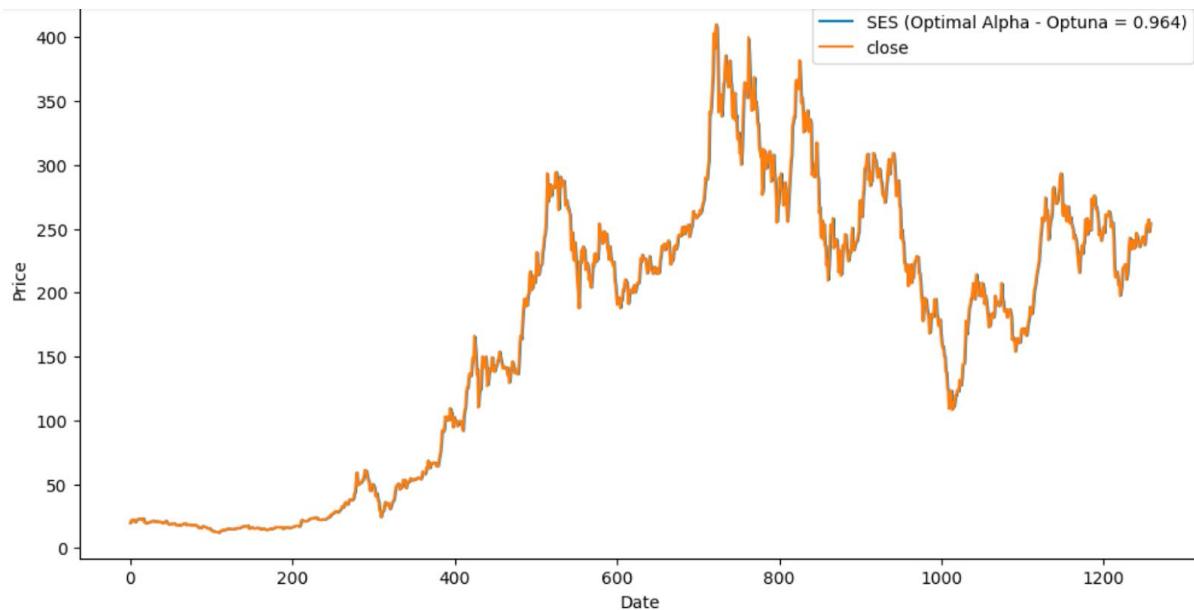
Hình 2. 46 Chạy mô hình SES alpha tối ưu

- Vẽ biểu đồ so sánh giá dự đoán từ mô hình SES với alpha tối ưu (được tìm bằng Optuna) và giá thực tế từ cột 'close' trong df **closedf**. In ra giá trị alpha tối ưu và sai số trung bình tuyệt đối (MAE) tương ứng. Tính toán và hiển thị các chỉ số đánh giá như sai số trung bình tuyệt đối (MAE), tỷ lệ sai số trung bình tuyệt đối (MAPE), sai số bình phương trung bình (MSE) và điểm R2.



```
# Xây dựng biểu đồ
closedf[['SES_Optimal_Optuna']].plot(figsize=(12,6),legend = TRUE, label=f'SES (Optimal Alpha - Optuna = {best_alpha_optuna:.3f})')
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Simple Exponential Smoothing (Optimal Alpha = {best_alpha_optuna:.3f})')
closedf[['close']].plot(figsize=(12,6),legend = TRUE)
plt.show()
# In giá trị alpha tối ưu và MAE tương ứng
print("Optimal Alpha:", best_alpha_optuna)
print("Best MAE - Optuna:", mae_best_ses)

mape_sesop = mean_absolute_percentage_error(closedf['close'],closedf[['SES_Optimal_Optuna']])
mse_sesop = mean_squared_error(closedf['close'],closedf[['SES_Optimal_Optuna']])
r2_sesop = r2_score(closedf['close'],closedf[['SES_Optimal_Optuna']])
print("mean_absolute_error:",mae_best_ses)
print("mean_absolute_percentage_error",mape_sesop)
print("mean_square_error:",mse_sesop)
print("r2_score:",r2_sesop)
```



Hình 2. 47 Biểu đồ SES alpha tối ưu

- Các chỉ số

```
Optimal Alpha: 0.963960776369011
Best MAE - Optuna: 4.825555973851038
mean_absolute_error: 4.825555973851038
mean_absolute_percentage_error 0.028892156423852847
mean_square_error: 58.27273347608036
r2_score: 0.9950518816785826
```

5.6 Holt với hệ số chuẩn

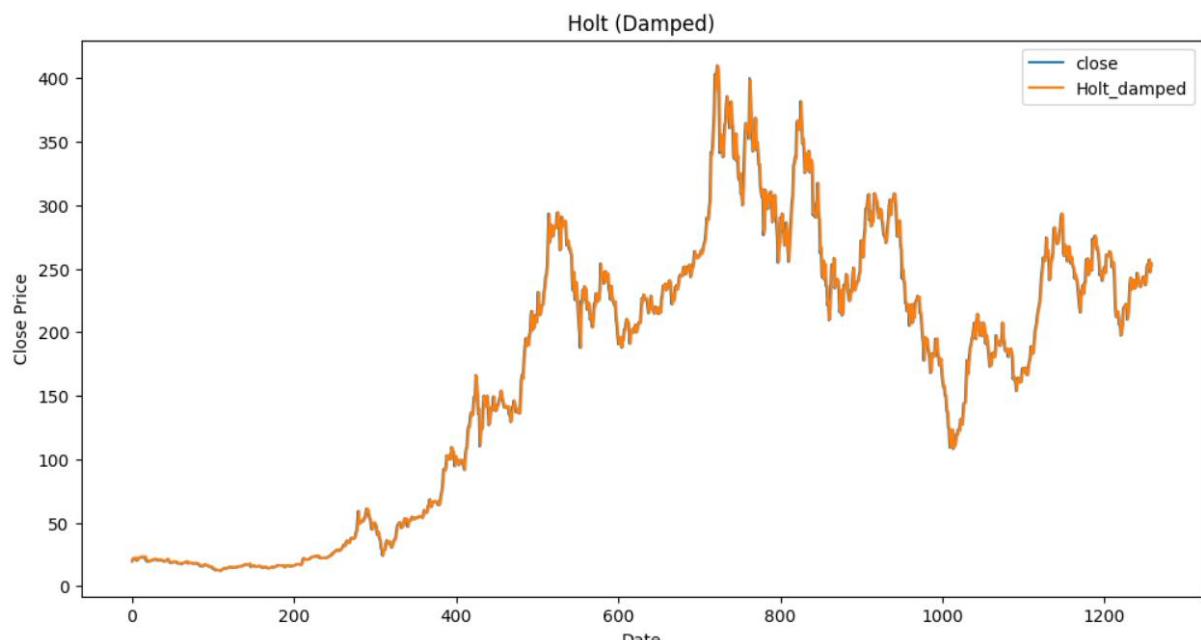
- Đoạn mã này thực hiện việc xây dựng một mô hình Holt-Winters với hệ số alpha, beta chuẩn và damped, sau đó lưu trữ giá trị dự đoán vào một cột mới trong DataFrame.



```
[ ] # Biểu đồ
closedf['close'].plot(figsize=(12,6),legend = TRUE)
closedf['Holt_damped'] .plot(figsize=(12,6),legend = TRUE)
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.title('Holt (Damped)')
plt.legend()
plt.show()

# Tính chỉ số đánh giá
mae_holt = mean_absolute_error(closedf['close'], closedf['Holt_damped'].dropna())
print("Mean Absolute Error (MAE):", mae_holt)

mape_holt = mean_absolute_percentage_error(closedf['close'],closedf['Holt_damped'])
mse_holt = mean_squared_error(closedf['close'],closedf['Holt_damped'])
r2_holt = r2_score(closedf['close'],closedf['Holt_damped'])
print("mean_absolute_percentage_error",mape_holt)
print("mean_square_error:",mse_holt)
print("r2_score:",r2_holt)
```



```
Mean Absolute Error (MAE): 4.82759035516011
mean_absolute_percentage_error 0.029105364691746974
mean_square_error: 58.26523336768943
r2_score: 0.9950525185360205
```

Hình 2. 48 Biểu đồ Holt hé số chuẩn

Vẽ biểu đồ so sánh giữa giá thực tế và giá dự đoán từ mô hình Holt-Winters với hé số alpha và beta chuẩn và damped



Tính chỉ số đánh giá: Tính toán các chỉ số đánh giá như Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), và R-squared (R2) cho mô hình Holt-Winters với hệ số alpha và beta chuẩn và damped.

5.7 Holt với hệ số tối ưu

- Xây dựng hàm tối ưu:

```
[ ] # Xây dựng mô hình Holt với alpha và beta
def objective_holt(trial):
    alpha = trial.suggest_float('alpha', 0.01, 0.99)
    beta = trial.suggest_float('beta', 0.01, 0.99)
    model = ExponentialSmoothing(df['close'], trend='add', damped=True).fit(smoothing_level=alpha, smoothing_slope=beta)
    predictions = model.fittedvalues
    MAE = mean_absolute_error(df['close'], predictions.dropna())
    return MAE
```

Hình 2. 49 Hàm tối ưu Holt

- Định nghĩa một hàm mục tiêu để được tối ưu hóa bởi optuna. Hàm này sẽ chấm điểm các tham số alpha và beta trong quá trình tinh chỉnh của mô hình Holt-Winters.
 - Xây dựng và huấn luyện mô hình Holt-Winters với các tham số alpha và beta đã được đề xuất.
 - Tính toán giá trị dự đoán từ mô hình và sau đó tính toán MAE giữa giá thực tế và giá dự đoán.
 - Kết quả trả về: Trả về giá trị MAE để optuna có thể sử dụng để tối ưu hóa tham số.
- Chạy mô hình:

```
▶ # Tìm giá trị alpha và beta tối ưu
study = optuna.create_study(direction='minimize')
study.optimize(objective_holt, n_trials=200)
best_alpha_optuna = study.best_params['alpha']
best_beta_optuna = study.best_params['beta']
mae_holt_op = study.best_value

# Xây dựng mô hình Holt với alpha và beta tối ưu từ Optuna
optimal_model_optuna = ExponentialSmoothing(closedf['close'], trend='add', damped=True)
.fit(smoothing_level=best_alpha_optuna, smoothing_slope=best_beta_optuna)
forecast_values = optimal_model_optuna.forecast(steps=7)
print(forecast_values)
closedf['Holt_Optimal_Optuna'] = optimal_model_optuna.fittedvalues
```

Hình 2. 50 Chạy mô hình Holt hệ số tối ưu

- Đoạn mã trên sử dụng thư viện optuna để tối ưu hóa giá trị của tham số alpha và beta cho mô hình Holt-Winters
- Tạo ra một đối tượng study để chứa thông tin về các thử nghiệm và kết quả tối ưu hóa.
- Thực hiện tối ưu hóa tham số alpha và beta bằng cách chạy hàm mục tiêu

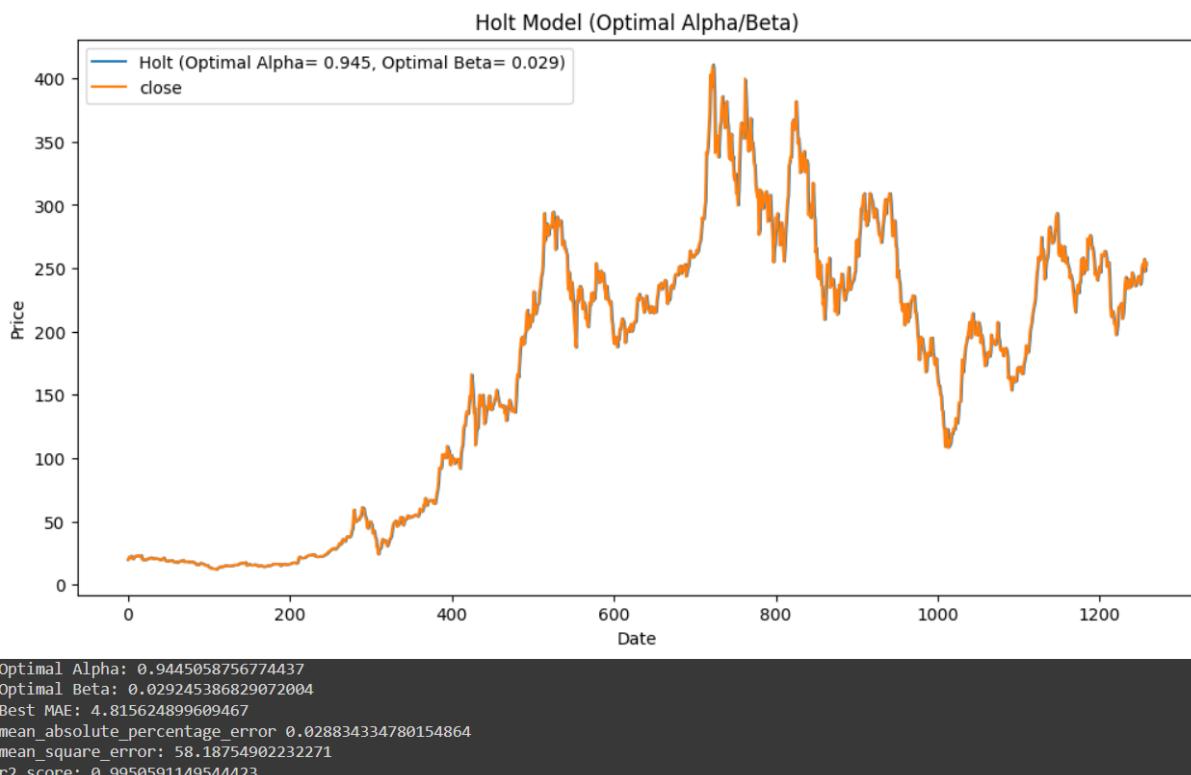


- Lấy giá trị tối ưu của alpha và beta từ kết quả tối ưu hóa của optuna.
- Lấy giá trị MAE tối ưu từ kết quả tối ưu hóa của optuna.

- Xây dựng biểu đồ:

```
[ ] # Xây dựng biểu đồ
closedf[['Holt_Optimal_Optuna']].plot(figsize=(12,6),legend = TRUE, label=f'Holt (Optimal Alpha= {best_alpha_optuna:.3f}, Optimal Beta= {best_beta_optuna:.3f})')
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Holt Model (Optimal Alpha/Beta)')
closedf[['close']].plot(figsize=(12,6),legend = TRUE)
plt.show()
# In giá trị alpha và beta tối ưu và MAE tương ứng
print("Optimal Alpha:", best_alpha_optuna)
print("Optimal Beta:", best_beta_optuna)
print("Best MAE:", mae_holt_op)

mape_holt_op = mean_absolute_percentage_error(closedf[['close']],closedf[['Holt_Optimal_Optuna']])
mse_holt_op = mean_squared_error(closedf[['close']],closedf[['Holt_Optimal_Optuna']])
r2_holt_op = r2_score(closedf[['close']],closedf[['Holt_Optimal_Optuna']])
print("mean absolute percentage error:",mape_holt_op)
print("mean square error:",mse_holt_op)
print("r2 score:",r2_holt_op)
```



Hình 2. 51 Biểu đồ Holt hé số tối ưu

- Vẽ biểu đồ so sánh giá thực tế (closedf[['close']]) và giá dự đoán từ mô hình Holt-Winters với alpha và beta tối ưu từ optuna.
- In ra giá trị tối ưu của alpha và beta, cũng như giá trị tối ưu của MAE.
- Tính toán và in ra các chỉ số đánh giá như MAPE, MSE và R-squared cho mô hình Holt-Winters tối ưu.



5.8 Holt winter với hệ số chuẩn

- Xây dựng mô hình:

```
[ ] # Xây dựng mô hình Holt-Winters với hệ số chuẩn (damping) và mùa vụ (seasonal)
model = ExponentialSmoothing(df['close'], trend='add', seasonal='add', damped=True, seasonal_periods=7).fit()
closedf['Holt_wt'] = model.fittedvalues
```

Hình 2. 52 Xây dựng mô hình holt winter hệ số chuẩn

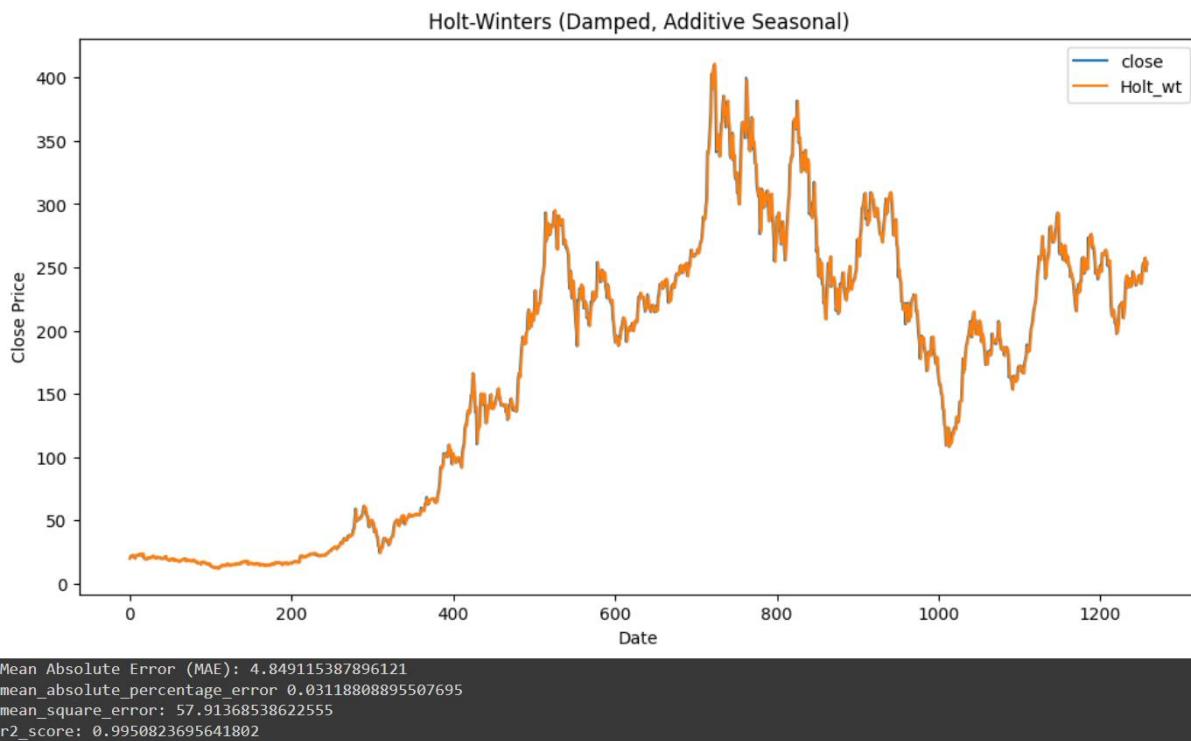
Đoạn mã này sử dụng mô hình Holt-Winters để dự đoán giá cổ phiếu từ dữ liệu đầu vào và lưu giữ giá trị dự đoán vào một cột mới trong DataFrame. Mô hình này có các thành phần xu hướng, mùa vụ, và sử dụng hệ số chuẩn để kiểm soát sự biến đổi của xu hướng theo thời gian.

- Vẽ biểu đồ:

```
[ ] # Vẽ biểu đồ
closedf['close'].plot(figsize=(12,6), legend = True)
closedf['Holt_wt'].plot(figsize=(12,6), legend = True)
plt.xlabel('Date')
plt.ylabel('Close Price')
plt.title(f'Holt-Winters (Damped, Additive Seasonal)')
plt.legend()
plt.show()

# Tính chỉ số đánh giá
mae_holtwinter = mean_absolute_error(df['close'], closedf['Holt_wt'].dropna())
print("Mean Absolute Error (MAE):", mae_holtwinter)

mape_holtwinter = mean_absolute_percentage_error(closedf['close'],closedf['Holt_wt'] )
mse_holtwinter = mean_squared_error(closedf['close'],closedf['Holt_wt'] )
r2_holtwinter = r2_score(closedf['close'],closedf['Holt_wt'] )
print("mean_absolute_percentage_error",mape_holtwinter)
print("mean_square_error:",mse_holtwinter)
print("r2_score:",r2_holtwinter)
```



Hình 2. 53 Biểu đồ Holt winter hệ số chuẩn

- Vẽ biểu đồ so sánh giữa giá đóng cửa thực tế và giá dự đoán từ mô hình Holt-Winters
- Tính các chỉ số đánh giá (evaluation metrics) của mô hình Holt-Winters: Tính toán các chỉ số đánh giá như Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), và R-squared (R2) để đánh giá hiệu suất của mô hình Holt-Winters.

5.9 Holt winter với hệ số tối ưu

- Xây dựng hàm tối ưu:

```
[ ] # Xây dựng hàm tối ưu
seasonal_periods = 60
def objective_holtwinter(trial):
    alpha = trial.suggest_float('alpha', 0.01, 0.99)
    beta = trial.suggest_float('beta', 0.01, 0.99)
    gamma = trial.suggest_float('gamma', 0, 0.99)
    seasonal = trial.suggest_categorical('seasonal', ['add', 'multiplicative'])

    model = ExponentialSmoothing(df['close'], trend='add', seasonal=seasonal, seasonal_periods=seasonal_periods, damped=True)
    .fit(smoothing_level=alpha, smoothing_slope=beta, smoothing_seasonal=gamma)
    predictions = model.fittedvalues
    MAE = mean_absolute_error(df['close'], predictions.dropna())
    return MAE
```

Hình 2. 54 Xây dựng hàm tối ưu holt winter

Đoạn mã này được sử dụng để định nghĩa hàm mục tiêu cho quá trình tinh chỉnh tham số (hyperparameter tuning) của mô hình Holt-Winters bằng cách sử dụng thư viện



optuna. Nhiệm vụ của quá trình này là tìm ra các giá trị tham số tối ưu để giảm thiểu Mean Absolute Error (MAE) giữa giá.

- Chạy mô hình:

```
[ ] # Tìm giá trị tối ưu cho alpha, beta, gamma, và seasonal
study = optuna.create_study(direction='minimize')
study.optimize(objective_holtwinter, n_trials=200)

# Lấy giá trị alpha, beta, gamma, và seasonal tối ưu
best_alpha_optuna = study.best_params['alpha']
best_beta_optuna = study.best_params['beta']
best_gamma_optuna = study.best_params['gamma']
best_seasonal_optuna = study.best_params['seasonal']

# Kiểm tra nếu MAE thấp nhất
mae_holtwinter_op = study.best_value

# Xây dựng mô hình Holt-Winters với hệ số tối ưu từ Optuna
optimal_model_optuna = ExponentialSmoothing(closeddf['close'], trend='add',
                                              seasonal=best_seasonal_optuna, seasonal_periods=seasonal_periods, damped=True)
.optifit(smoothing_level=best_alpha_optuna, smoothing_slope=best_beta_optuna, smoothing_seasonal=best_gamma_optuna)
forecast_values = optimal_model_optuna.forecast(steps=7)
print(forecast_values)
closeddf['Holt_Winters_Optimal'] = optimal_model_optuna.fittedvalues
```

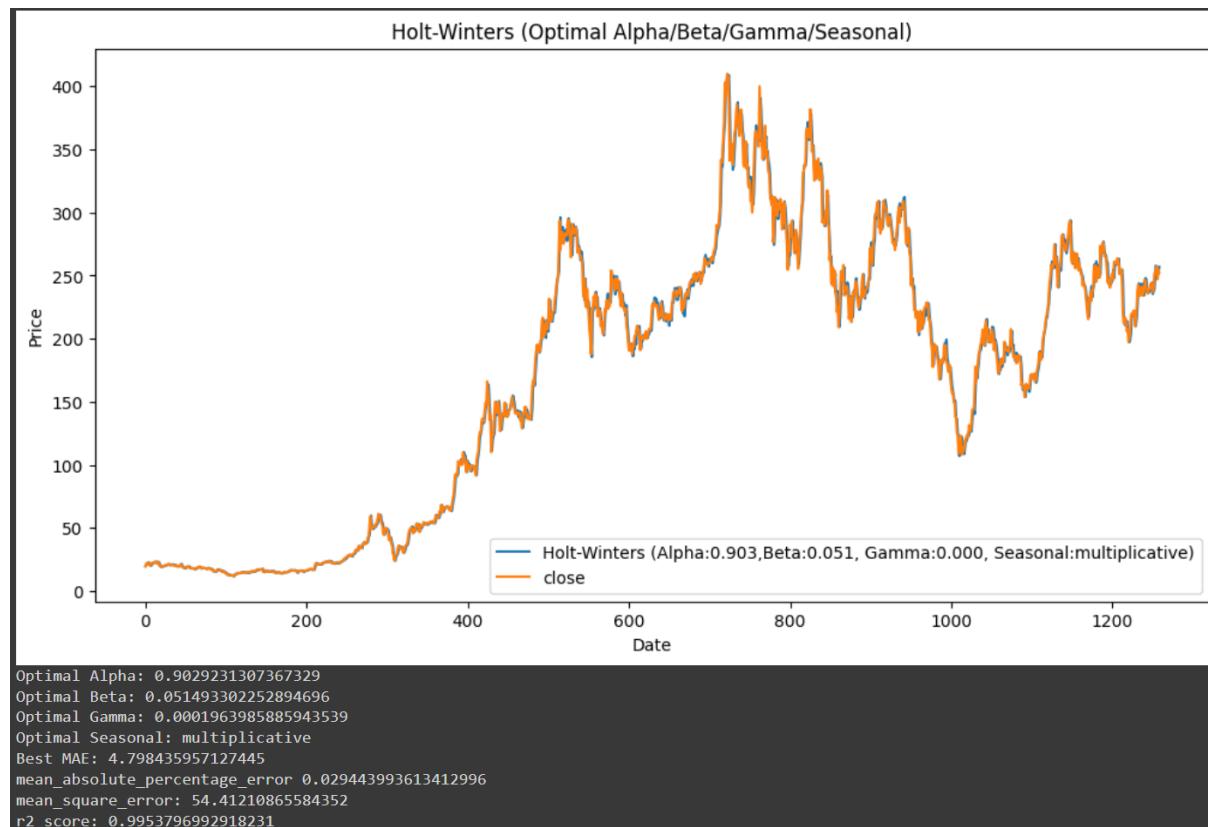
Hình 2. 55 Chạy mô hình holt winter

Đoạn mã này thực hiện quá trình tinh chỉnh tham số cho mô hình Holt-Winters bằng cách sử dụng thư viện optuna, sau đó xây dựng một mô hình Holt-Winters với các tham số tối ưu từ kết quả tinh chỉnh. Nó cũng dự đoán giá trị cho 7 bước tiếp theo và lưu trữ kết quả dự đoán vào DataFrame.

- Vẽ biểu đồ:

```
[ ] # Biểu đồ
closeddf['Holt_Winters_Optimal'].plot(figsize=(12,6), legend = TRUE, label=f'Holt-Winters (Alpha:{best_alpha_optuna:.3f}\
,Beta:{best_beta_optuna:.3f}, Gamma:{best_gamma_optuna:.3f}, Seasonal:{best_seasonal_optuna})')
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Holt-Winters (Optimal Alpha/Beta/Gamma/Seasonal)')
closeddf['close'].plot(figsize=(12,6), legend = TRUE)
plt.show()

# In giá trị alpha, beta, gamma, và seasonal tối ưu và MAE tương ứng
print("Optimal Alpha:", best_alpha_optuna)
print("Optimal Beta:", best_beta_optuna)
print("Optimal Gamma:", best_gamma_optuna)
print("Optimal Seasonal:", best_seasonal_optuna)
print("Best MAE:", mae_holtwinter_op)
mape_holtwinter_op = mean_absolute_percentage_error(closeddf['close'],closeddf['Holt_Winters_Optimal'])
mse_holtwinter_op = mean_squared_error(closeddf['close'],closeddf['Holt_Winters_Optimal'])
r2_holtwinter_op = r2_score(closeddf['close'],closeddf['Holt_Winters_Optimal'])
print("mean_absolute_percentage_error",mape_holtwinter_op)
print("mean_square_error:",mse_holtwinter_op)
print("r2_score:",r2_holtwinter_op)
```



Hình 2. 56 Biểu đồ holt winter hệ số tối ưu

- Vẽ biểu đồ so sánh giữa giá thực tế (closeddf['close']) và giá dự đoán từ mô hình Holt-Winters với các tham số tối ưu.
- Vẽ biểu đồ cho giá thực tế để so sánh với dự đoán từ mô hình Holt-Winters.
- In ra giá trị tối ưu của alpha, beta, gamma, seasonal và MAE.
- Tính toán và in ra các chỉ số đánh giá như MAPE, MSE và R-squared cho mô hình Holt-Winters tối ưu.

6. Đánh giá và so sánh mô hình

- Tạo dataframe chứa kết quả từ các mô hình

```
models = ['Simple SMA_naive', 'Simple SMA_3', 'Simple SMA_6', 'Simple SMA_alpha01', 'Simple SMA_alpha_op', 'Holt', 'Holt_op', 'HoltWinter', 'HoltWinter_op']
mae_scores = [mae_naive, mae_3mva, mae_6mva, mae_alpha01, mae_best_ses, mae_holt, mae_holt_op, mae_holtwinter, mae_holtwinter_op]
mse_scores = [mse_naive, mse_3mva, mse_6mva, mse_alpha01, mse_sesop, mse_holt, mse_holt_op, mse_holtwinter, mse_holtwinter_op]
mape_scores = [mape_naive, mape_3mva, mape_6mva, mape_alpha01, mape_sesop, mape_holt, mape_holt_op, mape_holtwinter, mape_holtwinter_op]
r2_scores = [r2_naive, r2_3mva, r2_6mva, r2_alpha01, r2_sesop, r2_holt, r2_holt_op, r2_holtwinter, r2_holtwinter_op]

# Tạo bảng đánh giá
evaluation_df = pd.DataFrame({'Model': models, 'MSE': mse_scores, 'MAE': mae_scores, 'MAPE': mape_scores, 'R2 Score': r2_scores})
evaluation_df.set_index('Model', inplace=True)
print(evaluation_df)
```



Model	MSE	MAE	MAPE	R2 Score
Simple SMA_naive	0.000000	0.000000	0.000000	1.000000
Simple SMA_3	31.664891	3.662137	0.021981	0.997307
Simple SMA_6	87.380941	6.241978	0.037729	0.992554
Simple SMA_alpha01	326.247972	12.078179	0.075670	0.972297
Simple SMA_alpha_op	58.272733	4.825556	0.028892	0.995052
Holt	58.265233	4.827590	0.029105	0.995053
Holt_op	58.187549	4.815625	0.028834	0.995059
Holtwinter	57.913685	4.849115	0.031188	0.995082
HoltWinter_op	54.412109	4.798436	0.029444	0.995380

Hình 2. 57 Dataframe kết quả từ các mô hình

- Moving Average: Mô hình có khả năng dự đoán tốt đối với những dự báo ngắn ngày, mang lại hiệu suất cao với MSE thấp.
- Simple Exponential Smoothing: Mô hình có khả năng dự đoán cho những dự báo ngắn ngày, nhưng có hiệu suất thấp hơn đối với moving average.
- Holt: Mô hình mang lại hiệu quả cao đối với các dự đoán dài ngày hơn, MSE thấp hơn Moving Average và SEM.
- Holt Winter: Thêm yếu tố xu hướng và mùa vụ, các dự đoán xa có MSE thấp, MAE được cải thiện so với các mô hình trước.

III. THIẾT KẾ GIAO DIỆN DỰ BÁO

1. Giới thiệu công cụ và trang web dự báo

Những công cụ chính mà nhóm sử dụng để thiết lập trang web phân tích và dự báo chứng khoán bao gồm:

- Streamlit: Thiết kế khung trang web, là framework chính để thiết các phần tử của trang như sidebar, các trường input, option, tiêu đề, đồng thời chuyển kết quả lên server lưu trữ.
- Plotly: giống như matplotlib, thư viện này dùng để tạo ra các biểu đồ thể hiện dữ liệu, tuy nhiên thư viện này cho phép tạo ra các biểu đồ có tính tương tác cao.
- Yfinance: đóng vai trò như API lấy dữ liệu từ nền tảng Yahoo Finance
- Os: thư viện dùng để đọc được đường dẫn tuyệt đối của các file trong trang web.
- Các thư viện khác sử dụng để tính toán, mô hình hóa và thực hiện đối với dữ liệu khác đều đã được nêu ở phần trên : Pandas, Numpy, Sklearn, Optuna, Statsmodel,...

Nhóm đã tiến hành đưa thiết kế giao diện lên nền tảng cloud Streamlit (nền tảng hỗ trợ public trang web có sử dụng streamlit). Việc đưa phần thiết kế được thực hiện thông qua những bước sau:

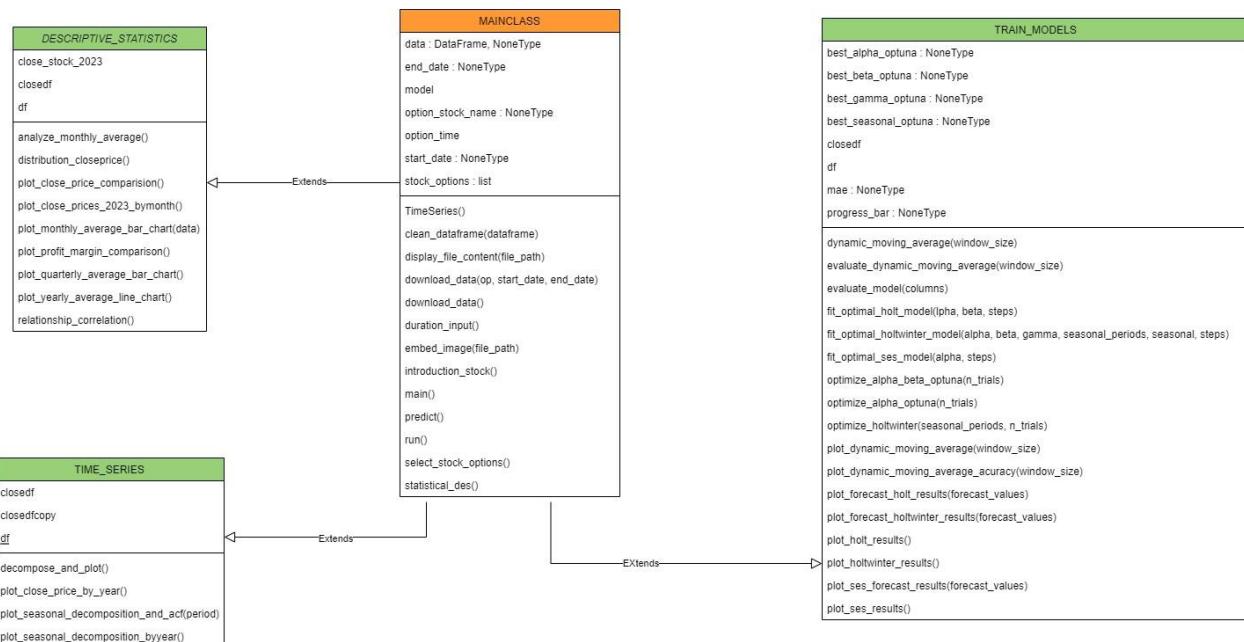
- Thiết kế giao diện local, chỉnh sửa, chạy thử các model và các biểu đồ.



- Đăng kí nền tảng Streamlit bằng tài khoảng Github.
- Sử dụng kho của Github làm nơi lưu trữ data và các file sử dụng cho trang web.
- Tiến hành xây dựng file requirements.txt để khai báo cho Streamlit server biết các thư viện với các phiên bản tương ứng của python để thiết lập môi trường chính của trang web.
- Reset lại cloud server của Streamlit để tải giao diện từ kho trên Github.

Kết quả đã được thể hiện ở trang web sau của nhóm : [STOCK-PRICE-PREDICTION](#)

2. Sơ đồ lớp (class diagram)



Hình 3. 1 Sơ đồ lớp giao diện trang web

STT	Tên class	Chức năng chính
1	MAINCLASS	Thực hiện các lệnh bên trong các tab. Ở mỗi tab, class này sẽ kế thừa các phương thức từ các class tương ứng để thực hiện.
2	DESCRIPTIVE_STATISTICS	Bao gồm nhiều hàm được xây dựng để tạo ra các biểu đồ ở tab thống kê mô tả của trang web
3	TIME_SERIES	Bao gồm các hàm xây dựng tạo ra các biểu đồ và dữ liệu ở tab phân tích time series của trang
4	TRAIN_MODELS	Bao gồm các hàm tối ưu mô hình, hàm lựa chọn và hàm xây dựng biểu đồ.

Bảng 3. 1 Các class và chức năng xây dựng trang web



MAIN_CLASS		
STT	Tên phương thức	Chức năng chính
1	TimeSeries	Hàm sử dụng kế thừa từ class TIME_SERIES để chạy dữ liệu cho tab phân tách time series
2	clean_dataframe	Hàm làm sạch dữ liệu
3	display_file_content	Hàm đọc dữ liệu file txt
4	download_data	Hàm download data từ yfinance
5	duration_input	Hàm điều khiển nhập dữ liệu trường input
6	embed_image	Đọc file hình ảnh
7	introduction_stock	Điều khiển thực hiện tab tổng quan
8	main	Hàm thực hiện các option giữa các tab. Thực hiện khi có option từ người dùng
9	predict	Hàm điều khiển thực hiện các phương thức từ class TRAIN_MODEL cho tab Predict
10	run	Tải giao diện
11	select_stock_options	Điều khiển option lựa chọn cổ phiếu
12	statistical_des	Điều khiển thực hiện class DESCRIPTIVE_STATISTIC cho tab thống kê mô tả.

Bảng 3. 2 Chi tiết main_class

DESCRIPTIVE_STATISTICS		
STT	Tên phương thức	Chức năng chính
1	analyze_monthly_average	Nhóm dữ liệu theo tháng
2	distribution_closeprice	Thực hiện phân phối dữ liệu
3	plot_close_price_comparision	Trực quan biểu đồ đường
4	plot_close_prices_2023_bymonth	Trực quan biểu đồ close price theo tháng 2023
5	plot_monthly_average_bar_chart	Trực quan biểu đồ cột theo tháng
6	plot_profit_margin_comparison	Biểu đồ tỷ suất lợi nhuận
7	plot_quarterly_average_bar_chart	Biểu đồ close price theo quý
8	plot_yearly_average_line_chart	Biểu đồ close price theo năm
9	relationship_correlation	Biểu đồ heatmap và pairplot

Bảng 3. 3 Chi tiết descriptive_statistics



TIME SERIES		
STT	Tên phương thức	Chức năng chính
1	decompose_and_plot	Vẽ biểu đồ trong nhóm seasonal decompose
2	plot close price by year	Vẽ biểu đồ close price theo năm
3	plot seasonal decomposition and acf	Vẽ biểu đồ seasonal decompose và ACF
4	plot_seasonal_decomposition_byyear	Vẽ biểu đồ seasonal decompose theo năm

Bảng 3. 4 Chi tiết time_series

TRAIN_MODELS		
STT	Tên phương thức	Chức năng chính
1	dynamic_moving_average	Nhận tham số và tính toán mô hình moving average
2	evaluate_dynamic_moving_average	Hàm đánh giá chỉ số cho moving average
3	evaluate_model	Hàm đánh giá chỉ số cho các mô hình
4	fit_optimal_holt_model	Train model holt
5	fit_optimal_holtwinter_model	Train model holt winter
6	fit_optimal_ses_model	Train model SES
7	optimize_alpha_beta_optuna	Hàm tối ưu Holt
8	optimize_alpha_optuna	Hàm tối ưu SES
9	optimize_holtwinter	Hàm tối ưu Holtwinter
10	plot_dynamic_moving_average	Vẽ biểu đồ moving average
11	plot_dynamic_moving_average_accuracy	
12	plot_forecast_holt_results	Vẽ biểu đồ holt
13	plot_holt_results	
14	plot_forecast_holtwinter_results	Vẽ biểu đồ Holt winter
15	plot_holtwinter_results	
16	plot_ses_forecast_results	Vẽ biểu đồ SES
17	plot_ses_results	

Bảng 3. 5 Chi tiết train_models

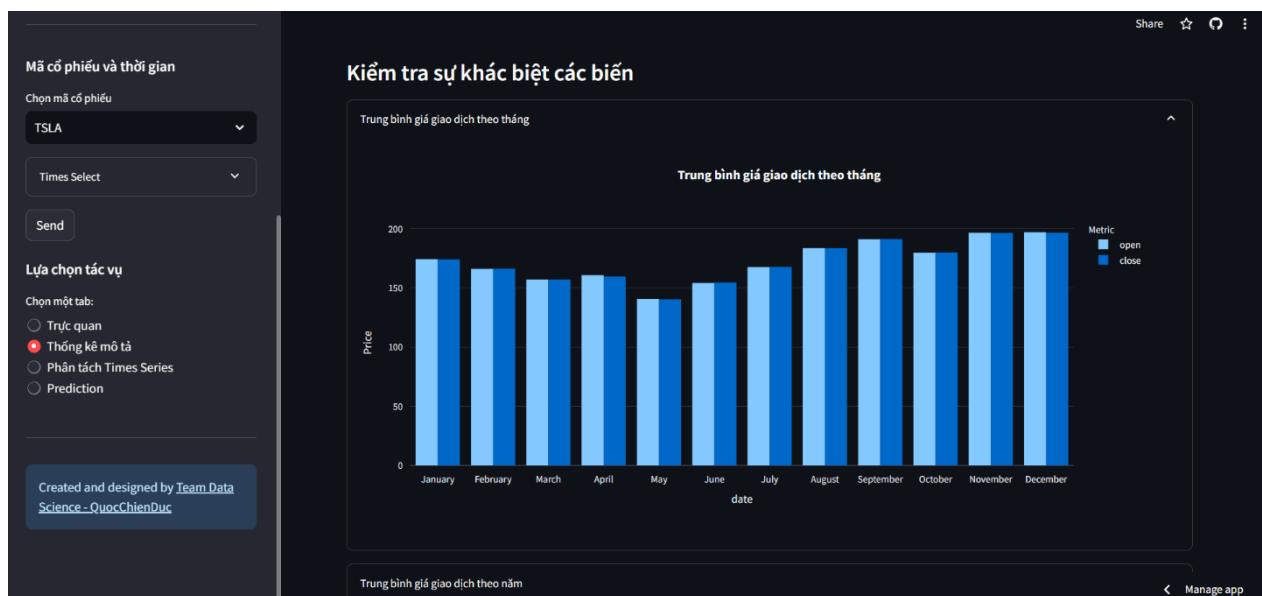
3. Kết quả

3.1 Trang chính – trực quan



Hình 3. 2 Trang chính – trực quan

3.2 Trang thống kê mô tả



Hình 3. 3 Trang thống kê mô tả

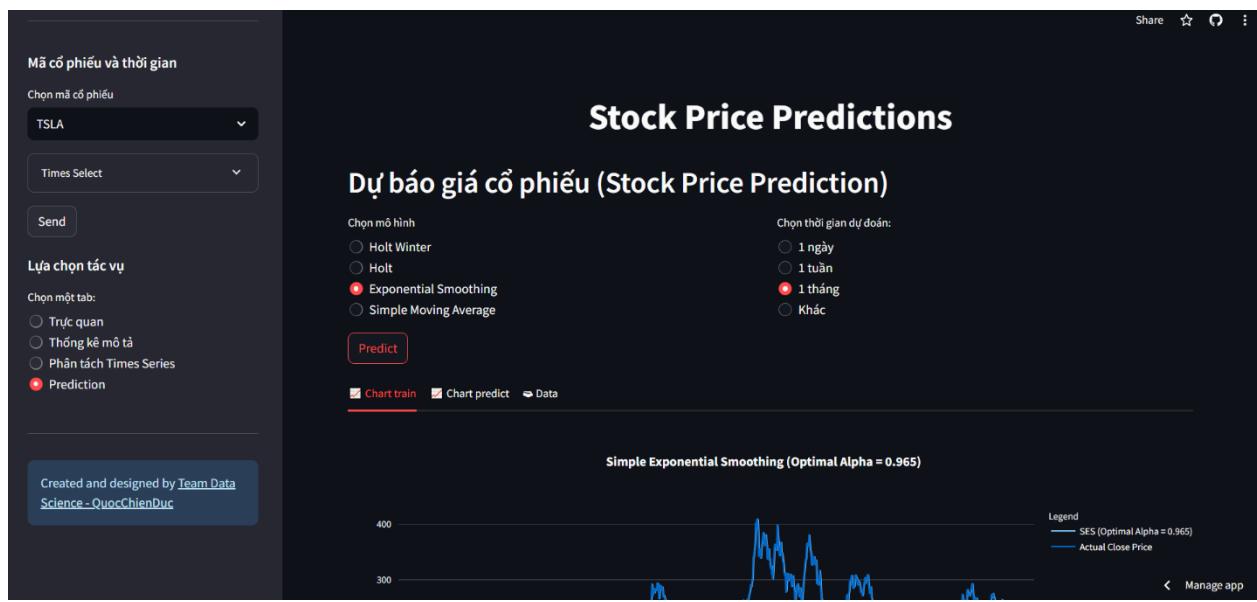


3.3 Trang phân tách time series



Hình 3. 4 Trang phân tách time series

3.4 Trang predict



Hình 3. 5 Trang predict



IV. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Qua việc phân tích các chỉ số từ thống kê mô tả, các kết quả đã phần nào phản ánh được những biến động liên quan đến các loại cổ phiếu ô tô trong những năm gần đây. Qua đó, nhóm đã đánh giá được những sự thay đổi, so sánh được thị trường ô tô và phát hiện được những trong dữ liệu cổ phiếu.

Việc phát triển những mô hình dự báo là nhiệm vụ và là hướng đi không còn mới, tuy nhiên ở Việt Nam những năm gần đây mới bắt đầu phát triển. Nhóm thực hiện mong muốn được phát triển nhiều mô hình hơn, ngày càng tối ưu và hướng dẫn cho những người đi sau có thể tiếp nhận những kinh nghiệm.

Qua việc thực hiện đề tài, nhóm cũng đã rút ra được những yếu điểm cũng như là ưu điểm của dự án và các mô hình sử dụng trong đó như sau:

Về ưu điểm:

- Phân tích được những chỉ số thống kê liên quan đến cổ phiếu và giá cổ phiếu.
- Trình bày được những yếu tố thời gian, phân tích các mã cổ phiếu khác nhau và đưa ra sự so sánh.
- Sử dụng các mô hình thời gian để dự báo, trong đó có việc xây dựng được các hàm tối ưu, sử dụng được thư viện optuna nhằm tối ưu thời gian chạy của mô hình.
- Trực quan được các mô hình và các kết quả phân tích được lên website, xây dựng được ở nền tảng website các biểu đồ động có tính tương tác và tính thực tế cao khi sử dụng.

Về nhược điểm:

- Những mô hình nhóm sử dụng trong quá trình dự báo có tính dự đoán tương đối, tức là việc dự báo còn phụ thuộc vào nhiều yếu tố mà trong khuôn khổ mô hình không giải quyết được.
- Đối với việc dự báo các kết quả dài ngày, các mô hình của nhóm sẽ có sai số ngày càng cao, nếu sử dụng dữ liệu nhiều có thể dẫn đến tình trạng overfitting khiến dự báo sai lệch.

2. Hướng phát triển

Dựa trên những phân tích của nhóm, các yếu tố đánh giá ở trên, nhóm đưa ra những hướng phát triển khả dụng sau cho đề tài:



- Sử dụng mô hình ARIMA (AutoRegressive Integrated Moving Average): Mô hình ARIMA là một lựa chọn phổ biến trong dự báo chuỗi thời gian, kết hợp giữa yếu tố autoregressive (AR) và moving average (MA).
- Mô hình neural networks, đặc biệt là mạng nơ-ron hồi quy (RNN) hoặc mạng nơ-ron hồi quy dài hạn (LSTM), có thể được sử dụng để dự báo chuỗi thời gian phức tạp.
- Prophet là một mô hình dự báo do Facebook phát triển, được thiết kế đặc biệt cho việc dự báo chuỗi thời gian có tính chất ngày, có thể xử lý các yếu tố như ngày nghỉ, sự biến động mùa vụ, và các sự kiện lớn.
- Mô hình gradient boosting như XGBoost và LightGBM không chỉ dành cho dự báo dạng phân loại mà còn có thể được sử dụng để dự báo chuỗi thời gian.
- GARCH (Generalized Autoregressive Conditional Heteroskedasticity): GARCH là một lựa chọn cho việc mô hình hóa biến động trong phân phối của chuỗi thời gian và có thể được sử dụng để dự báo biến động giá cổ phiếu.

V. TÀI LIỆU THAM KHẢO

IvT, P., & IvT. (2023). *Moving Average là gì? Tìm hiểu về đường trung bình động*. Retrieved from <https://hocpriceaction.com/moving-average-la-gi-ve-duong-trung-binh-dong/>

Tanna, V. (2022). *Time series Forecasting - Holt's method*. Retrieved from <https://www.datascienceprophet.com/time-series-forecasting-holts-method/>

SolarWinds. (2023). *Holt-Winters Forecasting and Exponential Smoothing Simplified*. Retrieved from <https://orangematter.solarwinds.com/2019/12/15/holt-winters-forecasting-simplified/>

Streamlit: Công cụ cho demo code python. (2019). Retrieved from <https://fullstackstation.com/gioi-thieu-streamlit-la-gi/>

VI. PHỤ LỤC – CÁC LIÊN KẾT DỰ ÁN GITHUB

Liên kết đến dự án github

[GITHUB-STOCK-PRICE-PREDICTION-PYTHON](https://github.com/.../GITHUB-STOCK-PRICE-PREDICTION-PYTHON)

Liên kết đến trang web dự án

[STOCK-PRICE-PREDICTION](https://github.com/.../STOCK-PRICE-PREDICTION)