

TRƯỜNG ĐẠI HỌC KINH TẾ
KHOA THỐNG KÊ – TIN HỌC



BÁO CÁO THỰC TẬP NGHỀ NGHIỆP
NGÀNH HỆ THỐNG THÔNG TIN QUẢN LÝ
CHUYÊN NGÀNH QUẢN TRỊ HỆ THỐNG THÔNG TIN

PHÂN TÍCH DỮ LIỆU CHỨNG KHOÁN

Sinh viên thực hiện : **Trần Văn Lợi**

Lớp : **47K21.2**

Đơn vị thực tập : **TMA SOLUTION BÌNH ĐỊNH**

Cán bộ hướng dẫn : **Nguyễn Thị Thùy Dương**

Giảng viên hướng dẫn : **ThS. Nguyễn Văn Chúc**

Đà Nẵng, 8/2023

NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP

Họ và tên sinh viên: Lớp:

Khoa Thống kê – Tin học, Trường Đại học Kinh tế, Đại học Đà Nẵng

Thực tập từ ngày:...../...../2023 đến ngày:/...../2023

Tên đơn vị thực tập:.....

Địa chỉ:.....

Số điện thoại liên hệ:

Họ tên cán bộ hướng dẫn:

Sau quá trình thực tập của sinh viên tại đơn vị, chúng tôi có một số đánh giá như sau:

STT	Nội dung đánh giá	Rất không tốt	Không tốt	Bình thường	Tốt	Rất tốt
1	Về thái độ, ý thức, đạo đức và việc tuân thủ các quy định và văn hóa đơn vị thực tập	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Kiến thức chuyên môn	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Khả năng hòa nhập, thích nghi và tác phong nghề nghiệp	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	Trách nhiệm, sáng tạo trong công việc	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

(Anh/chị vui lòng đánh dấu X vào ô tương ứng với năng lực của sinh viên)

Ý kiến nhận xét và đề xuất (Nhằm nâng cao chất lượng đào tạo, Nhà trường rất mong muốn nhận thêm những ý kiến khác từ quý doanh nghiệp):

.....
.....
.....
.....

....., ngàythángnăm 2023

Xác nhận của đơn vị thực tập

LỜI CẢM ƠN

Em xin phép được gửi sự tri ân sâu sắc và lời cảm ơn chân thành nhất đối với các thầy cô giáo Khoa Thống kê- Tin học trường Đại Học Kinh Tế Đà Nẵng đã tạo điều kiện để em có điều kiện thực tập. Đặc biệt, em xin trân trọng cảm ơn thầy ThS.Nguyễn Văn Chúc đã nhiệt tình hướng dẫn để em có thể hoàn thành tốt kì thực tập này.

Đặc biệt, em xin chân thành gửi lời cảm ơn đến Mentor Nguyễn Thị Thùy Dương đã hướng dẫn trực tiếp, chỉ đạo và tạo mọi điều kiện giúp đỡ em trong suốt quá trình học cũng như thực tập tại đây.

Em cũng xin trân trọng gửi lời cảm ơn đến ban giám đốc và các anh chị nhân viên công ty TMA Solution Bình Định đã tận tình chỉ dẫn và giúp đỡ em trong suốt thời gian thực tập. Nhờ vậy mà em đã học thêm được nhiều kiến thức mới và có cái nhìn tường tận hơn về lý thuyết chuyên ngành cũng như thực tế áp dụng.

Trong suốt quá trình thực tập cũng như quá trình tìm hiểu, sẽ không thể tránh khỏi những sự thiếu sót và hạn chế. Em rất mong nhận được những ý kiến đóng góp và phản hồi từ quý thầy cô để em có thể khắc phục được những sai sót cũng như rút ra được những bài học cho mình và trau dồi thêm những kiến thức mới. Em xin chân thành cảm ơn!

LỜI CAM ĐOAN

Em xin cam đoan đề tài “Phân tích dữ liệu chứng khoán” là kết quả nghiên cứu độc lập dưới sự hướng dẫn của ThS Nguyễn Văn Chức và mentor Nguyễn Thị Thùy Dương không có sự sao chép từ bất kỳ nguồn nào khác. Ngoài ra, trong bài báo cáo có sử dụng một số nguồn tài liệu tham khảo đã được trích dẫn nguồn và chú thích rõ ràng. Em xin hoàn toàn chịu trách nhiệm trước bộ môn, khoa và nhà trường về sự cam đoan này..

Đà Nẵng, ngày 2 tháng 7 năm 2024

MỤC LỤC

LỜI CẢM ƠN	iii
LỜI CAM ĐOAN.....	iv
DANH MỤC HÌNH ẢNH.....	vii
DANH MỤC CÁC TỪ VIẾT TẮT	ix
LỜI MỞ ĐẦU	1
Chương I: TỔNG QUAN VỀ TMA.....	2
1.1. Quá trình hình thành và phát triển của công ty	2
1.2. Dịch vụ	2
1.3. Tầm nhìn sứ mệnh.....	4
1.4. Lĩnh vực hoạt động	4
1.5. Các giải pháp	5
1.6. Vị trí thực tập và cơ hội nghề nghiệp.....	7
1.6.1. Vị trí thực tập.....	7
1.6.2. Cơ hội nghề nghiệp.....	8
CHƯƠNG II. CƠ SỞ LÝ LUẬN VỀ PHÂN TÍCH DỮ LIỆU, MÔ HÌNH ...	9
2.1. Các khái niệm cơ bản về Data Analyst.....	9
2.1.1. Data Analyst là gì?	9
2.1.2. Các kỹ năng cần có khi phân tích dữ liệu.....	9
2.1.3. Những kỹ thuật phân tích	11
2.1.4. Các phương pháp phân tích dữ liệu cơ bản	12
2.1.5. Tầm quan trọng của phân tích dữ liệu.....	13
2.2. Phân loại dữ liệu	14
2.2.1. Dữ liệu có cấu trúc	14
2.2.1.1. Dữ liệu phân loại.....	14

2.2.1.2. Dữ liệu định lượng	14
2.2.2. Dữ liệu bán và phi cấu trúc.....	14
2.2.2.1. Dạng chữ.....	14
2.2.2.2. Dạng Multimedia	15
2.2.2.3. XML/JSON	15
2.3. Tổng quan về PowerBI, Python	15
2.3.1. PowerBI	15
2.3.2. Python.....	16
2.4. Khái niệm về dãy số thời gian.....	16
2.5. Giới thiệu về Long Short Term Memory (LSTM)	17
2.5.1 Cấu trúc của LSTM.....	17
2.5.2. Cách hoạt động của LSTM.....	18
2.5.3. Ưu điểm của LSTM.....	18
CHƯƠNG III. TRIỂN KHAI.....	19
3.1. Trục quan hóa dữ liệu chuỗi thời gian và dự báo.....	19
3.1.1. Thu Thập Dữ liệu	19
3.1.2. Tiền Xử Lý Dữ Liệu.....	19
3.1.3. Triển khai LSTM.....	21
3.2. Tối Ưu Hóa Danh Mục Đầu Tư	29
3.2.1. Thu Thập Dữ Liệu.....	29
3.2.2. Tiền Xử Lý Dữ Liệu Trên PYTHON.....	30
3.2.3. Lợi Nhuận và Rủi Ro Của Danh Mục.....	32
3.2.4. Tỷ Lệ Sharpe.....	36
3.2.5. Trục Quan Hóa.....	39
TÀI LIỆU THAM KHẢO	45

DANH MỤC HÌNH ẢNH

Hình 1: Logo công ty TMA	2
Hình 2: Dữ liệu công ty AMZN	19
Hình 3: Import thư viện	Error! Bookmark not defined.
Hình 4: Xóa cột và thống nhất cách biểu diễn dữ liệu	20
Hình 5: Chuyển cột ‘Date’ thành Index.....	20
Hình 6: Kiểm tra giá trị null.....	21
Hình 7: Vẽ biểu đồ đường về dữ liệu giá đóng cửa trong quá khứ.....	21
Hình 8: Chia dữ liệu train và test	22
Hình 9: Chuẩn hóa minmax	22
Hình 10: Lấy giá trị huấn luyện và giá trị dự báo	22
Hình 11: Dữ liệu giá trị huấn luyện.....	Error! Bookmark not defined.
Hình 12: Dữ liệu giá trị dự báo	Error! Bookmark not defined.
Hình 13: Chuyển giá trị thành mảng 3 chiều cho huấn luyện và mảng 2 chiều cho dự báo	23
Hình 14: Số lượng lớp	24
Hình 15: Huấn luyện mô hình.....	24
Hình 16: Dự đoán trên dữ liệu train	25
Hình 17: Xử lý và dự đoán trên dữ liệu test	25
Hình 18: Biểu đồ đường về giá trị thực tế và giá dự đoán.....	26
Hình 19: Chỉ số đánh giá của dữ liệu train	27
Hình 20: Chỉ số đánh giá về dữ liệu test.....	27
Hình 21: Biểu đồ đường về dự đoán giá ngày tiếp theo.....	29
Hình 22: Dữ liệu chứng khoán công ty Amazon	30
Hình 23: Xử lý dữ liệu và chuyển cột ‘Date’ thành Index và giữ lại cột cần thiết.....	31

Hình 24: Kiểm tra dữ liệu null	32
Hình 25: Tính toán lợi nhuận mỗi ngày của các cổ phiếu	32
Hình 26: Tính toán lợi nhuận tích lũy và vẽ biểu đồ đường cho các cổ phiếu	33
Hình 27: Tính toán lợi nhuận theo năm, ma trận phương sai, giá trị độ lệch chuẩn.....	34
Hình 28: Tạo ra Weights, tính toán lợi nhuận và giá trị volatility của mỗi portfolio	35
Hình 29: Tính Sharpe Ratio của mỗi portfolio	37
Hình 30: Biểu đồ Scatter thể hiện portfolio tối ưu nhất	39
Hình 31: Dashboard Portfolio Optimization.....	40
Hình 32: Biểu đồ về lợi nhuận tích lũy qua các năm	40
Hình 33: Biểu đồ tròn về tỉ lệ phân bổ cổ phiếu và các chỉ số	42
Hình 34: Các chỉ số của mỗi công ty.....	43

DANH MỤC CÁC TỪ VIẾT TẮT

DA	: Data Analyst
LSTM	: Long Short Term Memory
AMZN	: Amazon
APPL	: Apple
BRK-A	: BERKSHIRE HATHAWAY
GOOG	: GOOGLE
NOK	: NOKIA
NFST	: CHIPBOND TECHNOLOGY
NVDA	: NVIDIA
TSLA	: TESLA

LỜI MỞ ĐẦU

1. Mục tiêu của đề tài

- **Thực Quan Hóa và Dự Báo Chuỗi Thời Gian**
 - Dự báo giá cổ phiếu tương lai bằng mô hình Long Short Term Memory
- **Tối Ưu Hóa Danh Mục Đầu Tư**
 - Tối đa hóa lợi nhuận cho mức độ rủi ro nhất định bằng cách sử dụng tỷ Lệ Sharpe.
 - Tạo bảng điều khiển tương tác trong Power BI để phân tích và tối ưu hóa hiệu suất của danh mục đầu tư cổ phiếu.

2. Đối tượng và phạm vi nghiên cứu

- Dữ liệu mô phỏng giá chứng khoán của các công ty trong sàn chứng khoán Mỹ

3. Kết cấu của đề tài

Đề tài được tổ chức gồm phần mở đầu, 3 chương nội dung và phần kết luận:

- Mở đầu
- **Chương 1:** Tổng quan về công ty TMA
- **Chương 2:** Cơ sở lý luận về phân tích dữ liệu, mô hình machine learning
- **Chương 3:** Triển khai
- Kết luận và hướng phát triển

CHƯƠNG I: TỔNG QUAN VỀ TMA

1.1. Quá trình hình thành và phát triển của công ty

Được thành lập năm 1997, TMA là tập đoàn công nghệ hàng đầu Việt Nam với 3500 kỹ sư và khách hàng là những tập đoàn công nghệ cao hàng đầu thế giới từ 30 quốc gia. TMA hiện có 7 chi nhánh tại Việt Nam (6 tại Tp.HCM và 1 ở Tp. Quy Nhơn) cùng 6 chi nhánh ở nước ngoài (Mỹ, Úc, Canada, Đức, Nhật, Singapore).



Hình 1: Logo công ty TMA

1.2. Dịch vụ

Với kinh nghiệm phong phú, TMA mang đến cho khách hàng các giải pháp toàn diện trong nhiều lĩnh vực với việc đồng hành cùng khách hàng từ giai đoạn tư vấn, ý tưởng, phân tích yêu cầu, tới các giai đoạn thiết kế, kiểm thử cho đến khi hoàn thiện sản phẩm và vận hành hệ thống 24/24.

- Project Planning
 - Lập kế hoạch dự án theo các mục tiêu và chiến lược của khách hàng với các giai đoạn triển khai theo thứ tự ưu tiên.
 - Lập chương trình chuyển đổi số, chuyển qua mô hình online, chuyển qua cloud...
- Solution Consulting
 - Phân tích hệ thống hiện tại và đề xuất các giải pháp giúp các công ty áp dụng các công nghệ mới để nâng cao năng lực cạnh tranh và hiệu quả hoạt động.
- R&D

- Đánh giá và thử nghiệm các công nghệ và phương pháp khác nhau để đề xuất giải pháp khả thi và phù hợp nhất.
- **Prototype PoC**
 - Đánh giá tính khả thi của ý tưởng.
 - Phát triển sản phẩm mẫu trong thời gian ngắn nhất.
- **Software Development**
 - Phát triển phần mềm với các công nghệ khác nhau.
 - Nâng cấp phần mềm sẵn có.
- **UX/UI Design**
 - Thiết kế giao diện Web, smartphone, máy tính bảng.
 - Cải tiến giao diện để tăng tính tương tác và trải nghiệm người dùng.
- **Software Testing**
 - Đánh giá chất lượng phần mềm với nhiều tiêu chí khác nhau.
 - Kiểm thử tự động.
- **Porting & Migration**
 - Chuyển đổi hệ thống cũ sang các công nghệ mới để tăng hiệu suất.
 - Chuyển đổi dữ liệu.
- **IT Managed Services**
 - Tư vấn, thiết kế hệ thống mạng, máy tính, đường truyền, bảo mật.
 - Quản trị về máy chủ và điện toán đám mây, hỗ trợ và bảo trì các ứng dụng.
 - Giải pháp điều hành và giám sát hệ thống mạng (NOC - Network Operation Center).
 - Giám sát hệ thống 24/24.
 - Sửa chữa và khắc phục sự cố.
- **Security Services**
 - Kiểm tra, đánh giá và cải tiến hệ thống an toàn thông tin.
 - Xây dựng hệ thống giám sát an ninh mạng (SOC - Security Operation Center).
 - Nâng cấp và chuẩn hóa hệ thống an toàn thông tin.

1.3. Tầm nhìn sứ mệnh

- Sứ mệnh:
 - BGD tập đoàn tin rằng, với sứ mệnh “Technology for People & Business” cùng khẩu hiệu “Yes! We Can”, Tập đoàn Công nghệ TMA sẽ tiếp tục đổi mới, sáng tạo, phát triển vững mạnh và đem lại những giá trị đích thực, vững bền cho khách hàng, đội ngũ nhân viên và cộng đồng.
- Nhiệm vụ:
 - Giúp khách hàng thành công bằng cách cung cấp các giải pháp phần mềm hiệu quả và sáng tạo
- Tầm nhìn:
 - Trở thành đối tác phần mềm đáng tin cậy và sáng tạo cho mọi khách hàng
- Giá trị cốt lõi:
 - Respect:
 - Đối xử với người khác theo cách bạn muốn được đối xử
 - Honesty
 - Thành thật với người khác và chính mình
 - Commitment
 - Chúng tôi biến lời hứa của mình thành hiện thực

1.4. Lĩnh vực hoạt động

- Trí tuệ nhân tạo/máy học

Đội ngũ kỹ sư đã ứng dụng Trí tuệ nhân tạo vào lĩnh vực tự động hóa máy móc, giáo dục, dược phẩm, xe hơi, quản trị nguồn nhân lực, nông nghiệp...với các công nghệ và giải pháp:
- Phân tích ngôn ngữ tự nhiên (NLP)
 - Nhận dạng hình ảnh và video (Object Detection)
 - Nhận dạng tài liệu (Document Parser)
 - Nhận dạng quảng cáo (Brand Detection)
 - Phân tích năng lực và hành vi học sinh (Student Analytics)
 - Tối ưu hoạt động máy móc (Machine Optimization)
- Dữ liệu lớn / phân tích dữ liệu

- Thiết kế hệ thống dữ liệu doanh nghiệp
- Thu thập và phân tích dữ liệu trong thời gian thực
- Tích hợp và tổng hợp dữ liệu
- Chuyển đổi dữ liệu
- Dự báo
- Iot và thiết bị thông minh

Có hơn 150 kỹ sư đang làm việc trong các dự án về IoT và Thiết bị thông minh áp dụng trong nhiều lĩnh vực:

- Công nghiệp
- Điện tử
- Xe hơi
- Viễn thông
- Y tế
- Giao thông
- Quản lý học sinh
- Quản lý tài sản

1.5. Các giải pháp

- Go innovative

Ứng dụng các công nghệ và phương pháp mới để tạo ra các sản phẩm và giải pháp sáng tạo - đột phá có tính cạnh tranh cao, hoặc tạo ra các giải pháp mới cho các bài toán cũ.

- Go digital

Tư vấn và triển khai giải pháp chuyển đổi số phù hợp với nhu cầu từng doanh nghiệp - tổ chức. Chia ra nhiều giai đoạn để tối ưu vốn đầu tư và mang lại hiệu quả thiết thực cho doanh nghiệp sau mỗi giai đoạn. Áp dụng các giải pháp sẵn có của TMA để giảm thời gian và chi phí chuyển đổi số.

- Go online

Giúp các doanh nghiệp chuyển đổi mô hình kinh doanh qua online:

- Phát triển ứng dụng web, mobile để giao dịch và tương tác với khách hàng.
- Thiết kế hệ thống hỗ trợ (back-office).

- Go mobile

Phát triển các ứng dụng di động và máy tính bảng để tương tác với khách hàng – đối tác hoặc tăng hiệu quả hoạt động của nhân viên.

- Go cloud

Thiết kế và triển khai các giải pháp điện toán đám mây:

- Phân tích hệ thống để xác định các ứng dụng và dữ liệu nên chuyển lên cloud.
- Chọn lựa cloud phù hợp cho từng doanh nghiệp.
- Chuyển đổi dữ liệu và ứng dụng lên cloud.
- Go automation

Ứng dụng các công nghệ trí tuệ nhân tạo, phân tích ngôn ngữ tự nhiên, nhận dạng ảnh, video để tạo ra các giải pháp tự động hóa quy trình hoạt động trong doanh nghiệp nhằm tăng năng suất, tăng sự chính xác và tiết kiệm chi phí nhân công.

- Go integrated

Giải pháp tích hợp các ứng dụng riêng lẻ vào chung một hệ thống, thiết kế dữ liệu thống nhất, báo cáo và phân tích chung giữa các phòng ban, các chi nhánh và công ty thành viên.

- Go smart

- Chuyển đổi các thiết bị thành smart device
- Điều khiển từ xa
- Phân tích dữ liệu và tự động xử lý
- Phát triển phần mềm cho smart camera, robot, drone...áp dụng 5g và các công nghệ nhận dạng hình ảnh và video

- Go interactive

Ứng dụng các phương pháp mới nhất để tạo ra các thiết kế giao diện web, mobile, pc trực quan và có tính tương tác cao.

- Go secure

Tư vấn và triển khai các giải pháp bảo mật trong doanh nghiệp, đánh giá an toàn web site, xây dựng các trung tâm điều hành mạng và bảo mật.

1.6. Vị trí thực tập và cơ hội nghề nghiệp

1.6.1. Vị trí thực tập

- Mô tả công việc

Business Intelligence (BI) là quá trình sử dụng công nghệ, công cụ và quy trình để thu thập, tích hợp, phân tích và trình bày thông tin kinh doanh. Mục tiêu của BI là hỗ trợ việc ra quyết định kinh doanh thông qua việc cung cấp các dữ liệu chính xác, đáng tin cậy và có giá trị

- Các nhiệm vụ chính:
 - Thu thập Dữ liệu
 - Xác định Nguồn Dữ liệu: Tìm hiểu và xác định các nguồn dữ liệu cần thiết, bao gồm dữ liệu nội bộ từ hệ thống ERP, CRM, và dữ liệu bên ngoài từ thị trường.
 - Truy xuất Dữ liệu: Sử dụng các công cụ ETL (Extract, Transform, Load) để truy xuất và tích hợp dữ liệu từ các nguồn khác nhau.
 - Làm sạch và Chuẩn hóa Dữ liệu
 - Làm sạch Dữ liệu: Xử lý các dữ liệu bị lỗi, thiếu sót, hoặc không nhất quán để đảm bảo tính chính xác.
 - Chuẩn hóa Dữ liệu: Đảm bảo dữ liệu được định dạng và lưu trữ theo một cấu trúc nhất định để dễ dàng phân tích.
 - Lưu trữ và Quản lý Dữ liệu
 - Thiết kế Kho Dữ liệu: Thiết kế và xây dựng các kho dữ liệu (data warehouses) để lưu trữ dữ liệu một cách hiệu quả.
 - Quản lý Cơ sở Dữ liệu: Đảm bảo rằng dữ liệu được lưu trữ an toàn và có thể truy xuất nhanh chóng khi cần thiết.
 - Phân tích Dữ liệu
 - Phân tích Mô tả (Descriptive Analytics): Phân tích dữ liệu lịch sử để hiểu rõ các xu hướng và mô hình đã xảy ra.

- Phân tích Chẩn đoán (Diagnostic Analytics): Tìm hiểu nguyên nhân của các xu hướng và sự kiện trong dữ liệu.
- Phân tích Dự đoán (Predictive Analytics): Sử dụng các mô hình dự báo để dự đoán các xu hướng và sự kiện trong tương lai.
- Phân tích Đề Xuất (Prescriptive Analytics): Đưa ra các khuyến nghị cụ thể dựa trên kết quả phân tích để cải thiện quyết định kinh doanh.
- Trực quan hóa Dữ liệu
 - Tạo Báo cáo và Bảng điều khiển: Sử dụng các công cụ như Power BI, Tableau để tạo ra các báo cáo và bảng điều khiển trực quan.
 - Trình bày Kết quả: Trình bày các kết quả phân tích dưới dạng biểu đồ, bảng biểu để dễ dàng hiểu và ra quyết định.
- Đưa ra Khuyến nghị và Hỗ trợ Quyết định
 - Cung cấp Thông tin và Khuyến nghị: Dựa trên kết quả phân tích, đưa ra các khuyến nghị để cải thiện hoạt động kinh doanh.
 - Hỗ trợ Quyết định: Hỗ trợ các nhà quản lý và lãnh đạo trong việc ra quyết định dựa trên dữ liệu.
- Giám sát và Đánh giá Hiệu suất
 - Theo dõi Hiệu suất: Sử dụng các chỉ số KPI để theo dõi và đánh giá hiệu suất của các hoạt động kinh doanh.
 - Phản hồi và Điều chỉnh: Dựa trên kết quả giám sát, đưa ra các điều chỉnh cần thiết để cải thiện hiệu suất.

1.6.2. Cơ hội nghề nghiệp

- Mức lương fresher: Tùy vào khả năng mà mức lương khởi điểm từ 6 triệu-2000\$.

CHƯƠNG II. CƠ SỞ LÝ LUẬN VỀ PHÂN TÍCH DỮ LIỆU, MÔ HÌNH

2.1. Các khái niệm cơ bản về Data Analyst

2.1.1. Data Analyst là gì?

Data Analyst (DA) là một tập hợp các quy trình, kiến thức, công cụ, và công nghệ được sử dụng để thu thập, lưu trữ, phân tích và trình bày dữ liệu doanh nghiệp nhằm hỗ trợ quá trình ra quyết định kinh doanh. BI giúp các tổ chức biến dữ liệu thô thành thông tin có giá trị và có thể hành động, từ đó cải thiện hiệu quả và hiệu suất kinh doanh.

2.1.2. Các kỹ năng cần có khi phân tích dữ liệu

Bất kể đang làm việc trong ngành nghề nào, nếu không hiểu những gì cần phải phân tích thì công việc sẽ gặp rất nhiều khó khăn, thậm chí là đi sai hướng. Đó là lý do tại sao kiến thức chuyên môn lại là một trong những yêu cầu chính của nhà tuyển dụng khi xét duyệt ứng viên vào cho vị trí này.

- Kỹ năng đặt câu hỏi

Giải quyết vấn đề là một trong những kỹ năng quan trọng nhất mà các chuyên viên phân tích dữ liệu cần có. Khoảng 90% công việc của chuyên viên phân tích dữ liệu đòi hỏi khả năng tư duy phản biện và sự khéo léo trong cách đặt câu hỏi. Chuyên viên phân tích dữ liệu cần phải đưa ra được các câu hỏi xác đáng cũng như vận dụng khả năng tư duy logic của mình để đọc hiểu dữ liệu và đưa ra giải pháp khắc phục điểm yếu và phát huy điểm mạnh. Chuyên viên phân tích càng nhạy bén với các con số thì càng nhanh chóng tìm được các giải pháp phù hợp.

- Các kỹ năng toán học

Việc trau dồi kỹ năng toán học là cực kỳ quan trọng. Chuyên viên phân tích dữ liệu phải am hiểu và vận dụng hiệu quả những kiến thức về toán thống kê, đại số tuyến tính, toán học ứng dụng,... Cần phải biết cách trích xuất thông tin từ các tập dữ liệu lớn sử dụng các công thức toán học và phần mềm máy tính.

- Thành thạo Excel và ngôn ngữ truy vấn cơ sở dữ liệu

Sử dụng thành thạo Microsoft Excel là một trong những kỹ năng cần thiết để có thể phân tích dữ liệu một cách hiệu quả. Đối với các chuyên viên phân tích dữ liệu thì đây là công cụ không thể thiếu trong quá trình làm việc. Các chuyên viên phân tích dữ liệu cần phải thành thạo ít nhất một ngôn ngữ truy vấn. Những ngôn ngữ này được sử dụng để hướng dẫn máy tính thực hiện các nhiệm vụ cụ thể liên quan đến phân tích dữ liệu. Một trong số ngôn ngữ truy vấn phổ biến nhất là SQL. Ngoài ra, còn có một số ngôn ngữ khác được thiết kế để thực hiện những chức năng cụ thể trong một số lĩnh vực chuyên biệt.

- **Khả năng trực quan hóa dữ liệu**

Công việc của chuyên viên phân tích dữ liệu là nghiên cứu, tìm hiểu về các chủ đề phức tạp và trình bày chúng một cách đơn giản nhất để đối tượng khách hàng có thể hiểu được. Để truyền tải được thông tin qua các con số, cần sử dụng các công cụ hỗ trợ trực quan như đồ thị, biểu đồ. Chúng là các phương tiện phổ biến và cực kỳ hiệu quả để minh họa những gì muốn diễn đạt.

- **Kỹ năng giao tiếp**

Các chuyên viên phân tích dữ liệu cần có kỹ năng giao tiếp, truyền đạt thông tin một cách hiệu quả vì công việc này đòi hỏi phải phối hợp với các bên liên quan, đồng nghiệp và cả các nhà cung cấp dữ liệu. Chuyên viên phân tích dữ liệu sẽ thường phải trình bày về những phân tích, kết luận của mình và đôi khi người nghe lại là những người chưa từng biết tới các phương pháp và quy trình phân tích trên. Khi đó, công việc của chuyên viên phân tích dữ liệu là biến các thuật ngữ phức tạp thành các khái niệm đơn giản và truyền đạt một cách dễ hiểu cho đồng nghiệp, khách hàng.

- **Hiểu biết về Machine learning**

Machine learning là một ứng dụng của trí tuệ nhân tạo (Artificial Intelligence - AI), có chức năng giúp các hệ thống tự học hỏi và nâng cấp mà không cần phải thông qua lập trình. Machine learning chủ yếu tập trung vào việc phát triển các chương trình máy tính thông qua nguồn dữ liệu thu thập được.

Trí tuệ nhân tạo (AI) và phân tích dự đoán (predictive analytics) đang trở thành hai trong số những chủ đề hot nhất của lĩnh vực khoa học dữ liệu, bước đệm rẽ hướng sang Data Scientist. Am hiểu về Machine Learning không phải là điều kiện bắt buộc nhưng để có thể dẫn đầu trong lĩnh vực phân tích dữ liệu, cần phải nắm được các công cụ và các khái niệm có liên quan đến công nghệ này.

2.1.3. Những kỹ thuật phân tích

- **Phân Tích Mô Tả (Descriptive Analytics)**
 - Mục đích: Tóm tắt và mô tả các đặc điểm chính của dữ liệu hiện tại hoặc dữ liệu lịch sử.
 - Phương pháp: Sử dụng các số liệu thống kê cơ bản như trung bình, trung vị, độ lệch chuẩn và các biểu đồ như biểu đồ cột, biểu đồ đường, biểu đồ tròn.
 - Ứng dụng: Giúp hiểu rõ về các xu hướng, mô hình và hiệu suất trong quá khứ.
- **Phân Tích Chẩn Đoán (Diagnostic Analytics)**
 - Mục đích: Tìm hiểu nguyên nhân của các hiện tượng hoặc xu hướng đã được xác định trong phân tích mô tả.
 - Phương pháp: Sử dụng phân tích tương quan, phân tích nguyên nhân-gốc, và phân tích chuỗi thời gian.
 - Ứng dụng: Giúp giải thích tại sao một sự kiện xảy ra và xác định các yếu tố ảnh hưởng.
- **Phân Tích Dự Đoán (Predictive Analytics)**
 - Mục đích: Dự đoán các xu hướng và kết quả trong tương lai dựa trên dữ liệu lịch sử và các mô hình thống kê.
 - Phương pháp: Sử dụng các mô hình hồi quy, phân tích chuỗi thời gian, mạng nơ-ron, và các thuật toán học máy (machine learning).
 - Ứng dụng: Dự đoán doanh số, xu hướng thị trường, hành vi khách hàng, và các rủi ro tiềm ẩn.
- **Phân Tích Đề Xuất (Prescriptive Analytics)**
 - Mục đích: Đưa ra các khuyến nghị cụ thể để tối ưu hóa kết quả và hỗ trợ quyết định kinh doanh.

- Phương pháp: Sử dụng các mô hình tối ưu hóa, mô phỏng, và các thuật toán học tăng cường (reinforcement learning).
- Ứng dụng: Đề xuất chiến lược marketing, quản lý chuỗi cung ứng, và phân bổ nguồn lực hiệu quả.

2.1.4. Các phương pháp phân tích dữ liệu cơ bản

- **Phân tích cụm**

Hành động nhóm một tập hợp các phần tử dữ liệu theo cách cho biết các phần tử giống nhau hơn (theo một nghĩa cụ thể) với nhau hơn là các phần tử trong các nhóm khác. Vì không có biến đích khi phân nhóm, phương pháp này thường được sử dụng để tìm các mẫu ẩn trong dữ liệu. Phương pháp này cũng được sử dụng để cung cấp ngữ cảnh bổ sung cho một xu hướng hoặc tập dữ liệu.

- **Phân tích theo nhóm**

Loại phương pháp phân tích dữ liệu này sử dụng dữ liệu lịch sử để kiểm tra và so sánh một phân đoạn xác định về hành vi của người dùng, sau đó có thể được nhóm lại với những phân đoạn khác có đặc điểm tương tự. Bằng cách sử dụng phương pháp phân tích dữ liệu này, bạn có thể có được nhiều hiểu biết sâu sắc về nhu cầu của người tiêu dùng hoặc hiểu biết chắc chắn về một nhóm mục tiêu rộng lớn hơn.

- **Phân tích hồi quy**

Phân tích hồi quy sử dụng dữ liệu lịch sử để hiểu giá trị của biến phụ thuộc bị ảnh hưởng như thế nào khi một (hồi quy tuyến tính) hoặc nhiều biến độc lập (hồi quy bội) thay đổi hoặc giữ nguyên. Bằng cách hiểu mối quan hệ của từng biến và cách chúng phát triển trong quá khứ, bạn có thể dự đoán các kết quả có thể xảy ra và đưa ra quyết định kinh doanh tốt hơn trong tương lai.

- **Mạng nơ-ron**

Mạng nơ-ron tạo nền tảng cho các thuật toán thông minh của học máy. Nó là một dạng phân tích theo hướng dữ liệu cố gắng, với sự can thiệp tối thiểu, để hiểu cách bộ não con người xử lý thông tin chi tiết và dự đoán các giá trị. Mạng nơ-ron học hỏi từ mỗi và mọi giao dịch dữ liệu, nghĩa là chúng phát triển và tiến bộ theo

thời gian. Một lĩnh vực ứng dụng điển hình của mạng nơ-ron là phân tích dữ liệu dự đoán. Có các công cụ báo cáo BI có tính năng này được triển khai bên trong chúng

- **Phân tích nhân tố**

Phân tích nhân tố, còn được gọi là "giảm thứ nguyên", là một loại phân tích dữ liệu được sử dụng để mô tả sự thay đổi giữa các biến quan sát, tương quan về số lượng các biến không được quan sát có khả năng thấp hơn được gọi là nhân tố. Mục đích ở đây là phát hiện ra các biến tiềm ẩn độc lập, một phương pháp phân tích lý tưởng để hợp lý hóa các phân đoạn dữ liệu cụ thể.

- **Khai thác dữ liệu**

Một phương pháp phân tích là thuật ngữ chung cho các chỉ số kỹ thuật và thông tin chi tiết để có thêm giá trị, hướng và ngữ cảnh. Bằng cách sử dụng đánh giá thống kê khám phá, khai thác dữ liệu nhằm xác định các yếu tố phụ thuộc, quan hệ, mẫu dữ liệu và xu hướng để tạo ra và nâng cao kiến thức. Khi xem xét cách phân tích dữ liệu, áp dụng tư duy khai thác dữ liệu là điều cần thiết để thành công như vậy, đó là một lĩnh vực đáng để khám phá chi tiết hơn.

- **Phân tích văn bản**

Phân tích văn bản, còn được gọi trong ngành là khai thác văn bản, là quá trình lấy một bộ dữ liệu văn bản lớn và sắp xếp nó theo cách giúp dễ quản lý hơn. Bằng cách thực hiện quá trình làm sạch này một cách chi tiết nghiêm ngặt, bạn sẽ có thể trích xuất dữ liệu thực sự có liên quan đến doanh nghiệp của mình và sử dụng dữ liệu đó để phát triển những thông tin chi tiết hữu ích.

Các công cụ và kỹ thuật phân tích dữ liệu hiện đại đẩy nhanh quá trình phân tích văn bản. Nhờ sự kết hợp của máy học và các thuật toán thông minh, có thể thực hiện các quy trình phân tích nâng cao như phân tích cảm tính. Kỹ thuật này cho phép hiểu ý định và cảm xúc của văn bản, chẳng hạn như văn bản tích cực, tiêu cực hoặc trung tính, sau đó cho điểm tùy thuộc vào các yếu tố và danh mục nhất định.

2.1.5. Tầm quan trọng của phân tích dữ liệu

Phân tích dữ liệu rất quan trọng vì nó cung cấp nền tảng cho việc ra quyết định dựa trên thông tin chính xác và kịp thời, giúp doanh nghiệp giảm thiểu rủi ro

và tối ưu hóa hiệu suất. Bằng cách khai thác và phân tích dữ liệu, các tổ chức có thể hiểu rõ hơn về xu hướng, hành vi khách hàng và hiệu suất kinh doanh, từ đó đưa ra các quyết định chiến lược thông minh, cải thiện hoạt động và thúc đẩy sự phát triển bền vững.

2.2. Phân loại dữ liệu

2.2.1. Dữ liệu có cấu trúc

Dữ liệu có cấu trúc là dữ liệu được tổ chức theo một định dạng xác định, dễ dàng lưu trữ và truy xuất trong các cơ sở dữ liệu quan hệ như SQL. Loại dữ liệu này thường bao gồm bảng với các hàng và cột rõ ràng, ví dụ như dữ liệu tài chính, thông tin khách hàng, và các giao dịch thương mại.

2.2.1.1. Dữ liệu phân loại

Dữ liệu phân loại, còn gọi là dữ liệu định tính, là dữ liệu được phân thành các nhóm hoặc loại dựa trên các đặc tính hoặc thuộc tính không có thứ tự tự nhiên. Ví dụ bao gồm màu sắc, loại sản phẩm, và trạng thái đơn hàng (như "chờ xử lý", "đã giao", "hủy bỏ").

2.2.1.2. Dữ liệu định lượng

Dữ liệu định lượng là dữ liệu thể hiện các giá trị số lượng và có thể được đo lường. Loại dữ liệu này bao gồm dữ liệu liên tục (như chiều cao, cân nặng) và dữ liệu rời rạc (như số lượng sản phẩm bán ra, số lượng nhân viên).

2.2.2. Dữ liệu bán cấu trúc và phi cấu trúc

Dữ liệu bán cấu trúc là dữ liệu không tuân theo một cấu trúc nghiêm ngặt như dữ liệu có cấu trúc, nhưng vẫn chứa các nhãn hoặc đánh dấu để phân loại các phần tử. Ví dụ bao gồm email, tệp XML và JSON. Dữ liệu phi cấu trúc là dữ liệu không có cấu trúc cụ thể hoặc không theo một định dạng xác định, chẳng hạn như văn bản tự do, video, và hình ảnh.

2.2.2.1. Dạng chữ

Dữ liệu dạng chữ là bất kỳ thông tin nào được biểu thị dưới dạng văn bản, bao gồm tài liệu, bài viết, blog, và tin nhắn. Loại dữ liệu này có thể là

cả có cấu trúc (như trong cơ sở dữ liệu văn bản) và phi cấu trúc (như email và tài liệu văn bản tự do).

2.2.2.2. Dạng Multimedia

Dữ liệu dạng multimedia bao gồm hình ảnh, âm thanh, video, và các tệp đa phương tiện khác. Loại dữ liệu này thường là phi cấu trúc và yêu cầu các kỹ thuật đặc biệt để lưu trữ, truy xuất và phân tích.

2.2.2.3. XML/JSON

XML (eXtensible Markup Language) và JSON (JavaScript Object Notation) là hai định dạng phổ biến để lưu trữ và trao đổi dữ liệu bán cấu trúc. XML sử dụng các thẻ để xác định các phần tử dữ liệu và thuộc tính, trong khi JSON sử dụng cặp khóa-giá trị để tổ chức dữ liệu. Cả hai đều được sử dụng rộng rãi trong các ứng dụng web và dịch vụ API để trao đổi dữ liệu giữa các hệ thống.

2.3. Tổng quan về PowerBI, Python

2.3.1. PowerBI

Power BI là một công cụ Business Intelligence (BI) mạnh mẽ được phát triển bởi Microsoft. Power BI cung cấp khả năng kết nối với nhiều nguồn dữ liệu, từ đó tạo ra các báo cáo và bảng điều khiển trực quan, giúp người dùng phân tích và hiểu rõ hơn về dữ liệu của mình. Các tính năng nổi bật của Power BI bao gồm:

- **Trực quan hóa Dữ liệu:** Tạo ra các biểu đồ, đồ thị và báo cáo tương tác.
- **Kết nối Dữ liệu Đa Dạng:** Hỗ trợ kết nối với nhiều nguồn dữ liệu khác nhau như SQL Server, Excel, SharePoint, và các dịch vụ đám mây như Azure, Google Analytics.
- **Tích hợp DAX:** Sử dụng ngôn ngữ DAX (Data Analysis Expressions) để thực hiện các phép tính và phân tích dữ liệu phức tạp.
- **Chia sẻ và Hợp tác:** Dễ dàng chia sẻ báo cáo và bảng điều khiển với các thành viên trong nhóm hoặc toàn bộ tổ chức.
- **Cập nhật Thời Gian Thực:** Cập nhật dữ liệu và báo cáo liên tục để cung cấp thông tin mới nhất.

2.3.2. Python

Python là một ngôn ngữ lập trình mạnh mẽ và linh hoạt, nổi tiếng với cú pháp đơn giản và dễ hiểu, giúp các lập trình viên nhanh chóng phát triển các ứng dụng. Python được sử dụng rộng rãi trong nhiều lĩnh vực, đặc biệt là phân tích dữ liệu, khoa học dữ liệu và học máy. Các điểm mạnh của Python bao gồm:

- Thư viện Phong Phú: Có rất nhiều thư viện mạnh mẽ dành cho phân tích dữ liệu như NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, TensorFlow.
- Cộng đồng Mạnh Mẽ: Cộng đồng lập trình viên Python rất lớn, với nhiều tài liệu, diễn đàn và hỗ trợ từ cộng đồng.
- Đa Năng và Linh Hoạt: Python có thể được sử dụng cho phát triển web, tự động hóa, phân tích dữ liệu, trí tuệ nhân tạo, và nhiều ứng dụng khác.
- Khả Năng Mở Rộng: Dễ dàng tích hợp với các ngôn ngữ và công nghệ khác.

2.4. Khái niệm về dãy số thời gian

Dãy số thời gian trong tiếng Anh là Time Series. *Dãy số thời gian* còn được gọi là chuỗi thời gian. Dãy số thời gian là dãy các trị số của chỉ tiêu thống kê được sắp xếp theo thứ tự thời gian, ví dụ như hàng ngày, hàng giờ, hàng tháng hoặc theo các đơn vị thời gian khác nhau tùy thuộc vào bối cảnh ứng dụng.

Phân tích dãy số thời gian thường được sử dụng để hiểu và dự đoán xu hướng và mô hình hóa trong dữ liệu định giá rủi ro, quản lý tồn kho, dự báo kỳ vọng tương lai và nhiều ứng dụng khác. Các phương pháp phân tích dãy số thời gian có thể bao gồm việc áp dụng các mô hình thống kê, mô hình hồi quy và các kỹ thuật khác như smoothing, decomposition và forecasting.

Các thành phần của dãy số thời gian:

- Trend - Xu hướng: Xu hướng hiển thị hướng chung của dữ liệu dãy số thời gian trong một khoảng thời gian dài. Xu hướng có thể tăng (tăng), giảm (giảm) hoặc ngang (tĩnh).
- Cyclical Component - Thành phần mang tính chu kỳ: Đây là những xu hướng không có sự lặp lại cố định trong một khoảng thời gian cụ thể. Chu kỳ đề cập đến khoảng

thời gian thăng trầm, bùng nổ và suy thoái của một dãy số thời gian, chủ yếu được quan sát thấy trong các chu kỳ kinh doanh.

- Irregular Variation - Biến động không đều: Đây là những biến động trong dữ liệu dãy số thời gian trở nên rõ ràng khi loại bỏ các biến thể theo xu hướng và chu kỳ. Những biến thể này không thể đoán trước, thất thường và có thể ngẫu nhiên hoặc không.
- Stationary – Tính dừng: Là khi trung bình và phương sai của Y_t không thay đổi theo thời gian và giá trị của đồng phương sai giữa hai thời đoạn chỉ phụ thuộc vào khoảng cách hay độ trễ về thời gian giữa hai thời đoạn này chứ không phụ thuộc vào thời điểm thực tế mà đồng phương sai được tính.
- Seasonality - Tính mùa vụ: Thành phần mùa vụ thể hiện xu hướng lặp lại về thời gian, phương hướng và cường độ.

2.5. Giới thiệu về Long Short Term Memory (LSTM)

Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks), thường được gọi là LSTM là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi Hochreiter & Schmidhuber, và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

2.5.1 Cấu trúc của LSTM

- LSTM khắc phục các vấn đề của RNN thông qua một cấu trúc đặc biệt với các thành phần sau:
 - Cell State (Trạng thái ô nhớ): Đây là phần trung tâm của LSTM, chịu trách nhiệm truyền thông tin qua nhiều bước thời gian. Thông tin có thể dễ dàng chảy qua trạng thái ô nhớ với các thay đổi nhỏ, giúp duy trì thông tin lâu dài.
 - Forget Gate (Cửa quên): Quyết định thông tin nào cần quên từ trạng thái ô nhớ. Đầu vào cho cửa quên là trạng thái ẩn từ bước thời gian trước và đầu vào hiện tại, qua một hàm sigmoid để đưa ra giá trị từ 0 đến 1.
 - Input Gate (Cửa đầu vào): Quyết định thông tin nào từ đầu vào hiện tại cần được thêm vào trạng thái ô nhớ. Cửa này cũng sử dụng hàm sigmoid và tanh để quyết định mức độ cập nhật.

- Output Gate (Cửa đầu ra): Quyết định thông tin nào từ trạng thái ô nhớ sẽ được xuất ra ngoài. Nó sử dụng một hàm sigmoid để xác định phần nào của trạng thái ô nhớ sẽ là đầu ra.

2.5.2. Cách hoạt động của LSTM

Mỗi bước thời gian trong LSTM bao gồm các bước sau:

- Quyết định thông tin cần quên:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

- Quyết định thông tin cần cập nhật:

- Input Gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

- Tạo ra thông tin ứng viên mới:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

- Cập nhật trạng thái ô nhớ:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

- Quyết định thông tin đầu ra:

- Output Gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

- Tạo trạng thái ẩn mới:

$$h_t = o_t * \tanh(C_t)$$

2.5.3. Ưu điểm của LSTM

- Khả năng ghi nhớ dài hạn: LSTM có khả năng ghi nhớ thông tin qua nhiều bước thời gian mà không bị mất mát hoặc vanishing gradient.
- Linh hoạt: Có thể áp dụng cho nhiều loại dữ liệu chuỗi thời gian, bao gồm ngôn ngữ tự nhiên, chuỗi tín hiệu, và nhiều loại dữ liệu khác.

CHƯƠNG III. TRIỂN KHAI

3.1. Trực quan hóa dữ liệu chuỗi thời gian và dự báo

3.1.1. Thu Thập Dữ liệu

- Dữ liệu được thu thập từ kaggle về giá chứng khoán của công ty AMZN

Date	Low	Open	Volume	High	Close	Adjusted Close
15-05-1997	0.096354	0.121875003	1443120000	0.125	0.097916998	0.097916998
16-05-1997	0.085417002	0.098438002	294000000	0.098958001	0.086457998	0.086457998
19-05-1997	0.081249997	0.088021003	122136000	0.088541999	0.085417002	0.085417002
20-05-1997	0.081771001	0.086457998	109344000	0.087499999	0.081771001	0.081771001
21-05-1997	0.068750001	0.081771001	377064000	0.082291998	0.071354002	0.071354002
22-05-1997	0.065624997	0.071874999	235536000	0.072396003	0.069792002	0.069792002
23-05-1997	0.066666998	0.070312999	318744000	0.076041996	0.075000003	0.075000003
27-05-1997	0.072916999	0.075521	173952000	0.082291998	0.079167001	0.079167001
28-05-1997	0.076563001	0.081249997	91488000	0.081771001	0.076563001	0.076563001
29-05-1997	0.073958002	0.077082999	69456000	0.077082999	0.075259998	0.075259998
30-05-1997	0.073958002	0.075000003	51888000	0.075521	0.075000003	0.075000003
2/6/1997	0.075000003	0.075521	11832000	0.076563001	0.075521	0.075521
3/6/1997	0.073958002	0.076563001	23664000	0.076563001	0.073958002	0.073958002
4/6/1997	0.069792002	0.073958002	61608000	0.074478999	0.070832998	0.070832998
5/6/1997	0.068750001	0.070832998	113448000	0.077082999	0.077082999	0.077082999
6/6/1997	0.075521	0.075781003	156144000	0.085417002	0.082813002	0.082813002
9/6/1997	0.082813002	0.082813002	47040000	0.085417002	0.084375001	0.084375001
10/6/1997	0.076563001	0.085417002	109176000	0.085417002	0.079167001	0.079167001
11/6/1997	0.076563001	0.079687998	23760000	0.080208004	0.077082999	0.077082999
12/6/1997	0.077604003	0.079167001	32640000	0.082291998	0.080208004	0.080208004
13-06-1997	0.079167001	0.081249997	13872000	0.081249997	0.079167001	0.079167001
16-06-1997	0.078125	0.080208004	18264000	0.080208004	0.078645997	0.078645997
17-06-1997	0.07474	0.079948001	94128000	0.079948001	0.075259998	0.075259998
18-06-1997	0.075000003	0.076041996	49296000	0.076823004	0.075521	0.075521
19-06-1997	0.075000003	0.075521	20064000	0.076563001	0.075521	0.075521
20-06-1997	0.075000003	0.076563001	67752000	0.077604003	0.076301999	0.076301999

Hình 2: Dữ liệu công ty AMZN

- Về dữ liệu ta có:
 - Date: Ngày giao dịch chứng khoán
 - Low: Giá thấp nhất trong ngày mà cổ phiếu có thể đạt được (Giá sàn)
 - Open: Giá mở cửa trong ngày giao dịch
 - Volume: Khối lượng giao dịch của cổ phiếu trong ngày
 - High: Mức giá cao nhất trong ngày mà cổ phiếu có thể đạt được (Giá trần)
 - Close: Giá đóng cửa của ngày giao dịch
 - Adjusted Close: Giá điều chỉnh sau khi loại bỏ các tác động về cổ tức,...

3.1.2. Tiền Xử Lý Dữ Liệu

```
# đọc dữ liệu từ file csv
df = pd.read_csv('AMZN.csv')

# Xóa hai dòng "KL" và "Thay đổi %" từ DataFrame dataSet
df = df.drop(columns=["Open", "Low", "High", "Volume", "Adjusted Close"])
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True, format='mixed').dt.strftime('%d-%m-%Y')
print(df)
```

	Date	Close
0	15-05-1997	0.097917
1	16-05-1997	0.086458
2	19-05-1997	0.085417
3	20-05-1997	0.081771
4	21-05-1997	0.071354
...
6433	06-12-2022	88.250000
6434	07-12-2022	88.459999
6435	08-12-2022	90.349998
6436	09-12-2022	89.089996
6437	12-12-2022	88.504997

Hình 3: Xóa cột và thống nhất cách biểu diễn dữ liệu

Đọc dữ liệu từ file csv và tiến hành thay đổi định dạng của cột date về cùng một định dạng

```
[10] df['Date'] = pd.to_datetime(df['Date'], format='%d-%m-%Y') # Changed format string
df.sort_values(by=['Date'], inplace=True, ascending=True)
df.set_index('Date', inplace=True)

# Hiển thị lại DataFrame sau khi xóa
print(df)
```

Date	Close
1997-05-15	0.097917
1997-05-16	0.086458
1997-05-19	0.085417
1997-05-20	0.081771
1997-05-21	0.071354
...	...
2022-12-06	88.250000
2022-12-07	88.459999
2022-12-08	90.349998
2022-12-09	89.089996
2022-12-12	88.504997

[6438 rows x 1 columns]

Hình 4: Chuyển cột 'Date' thành Index

Đưa cột Date trở thành Index và tiến hành xóa các cột không cần thiết trong quá trình dự đoán

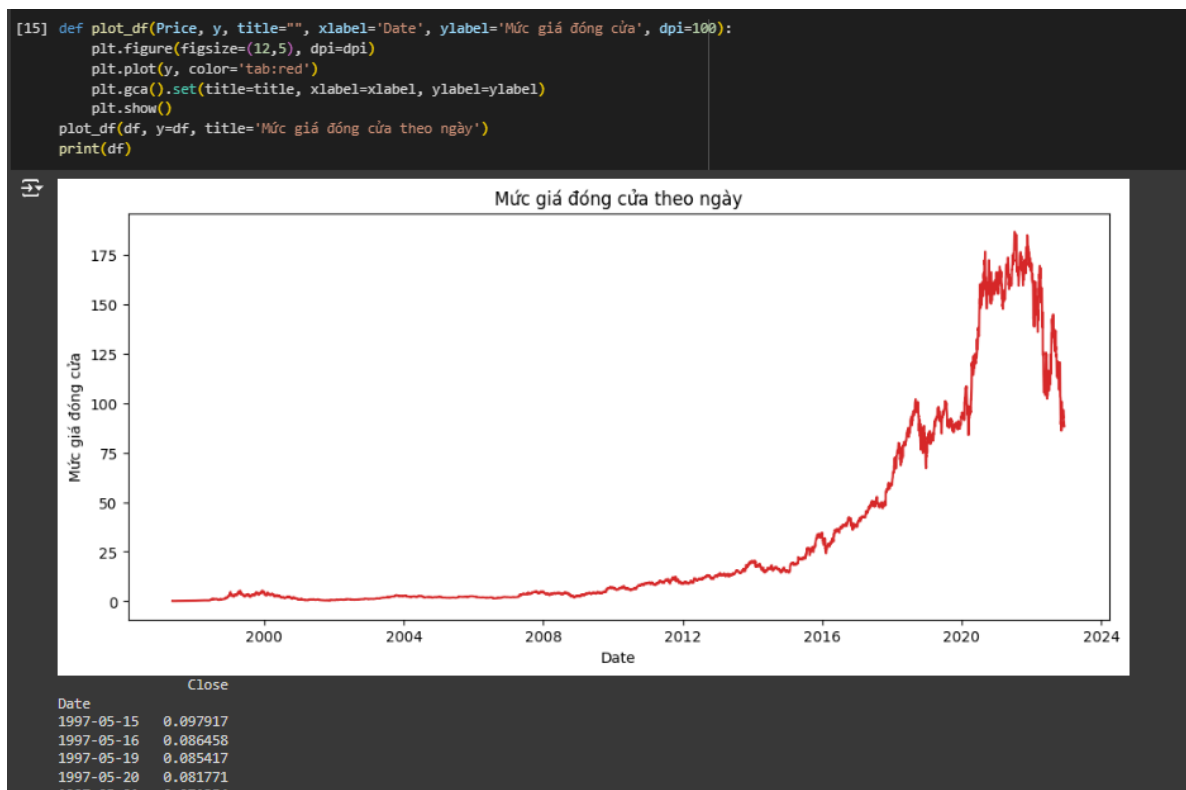
```
#dữ liệu 5 dòng đầu
print(df.isnull().sum())

Close      0
dtype: int64
```

Hình 5: Kiểm tra giá trị null

Kiểm tra xem giá trị có bị null

3.1.3. Triển khai LSTM



Hình 6: Vẽ biểu đồ đường về dữ liệu giá đóng cửa trong quá khứ

Vẽ biểu đồ giá trong lịch sử của cổ phiếu Amazon thì nhìn chung dữ liệu có xu hướng tăng qua các năm và độ biến động trong vòng 25 năm qua là khá lớn.

```
[16] #chia tập dữ liệu
      data = df.values
      train_data = data[:4507]
      test_data = data[4507:]
```

Hình 7: Chia dữ liệu train và test

Tiến hành chia tập dữ liệu ra thành dữ liệu train và dữ liệu test với dữ liệu train là 70% dùng để huấn luyện mô hình và dữ liệu test là 30% dùng để mô hình dự đoán

```
[18] #chuẩn hóa dữ liệu
      sc = MinMaxScaler(feature_range=(0,1))
      sc_train = sc.fit_transform(data)
```

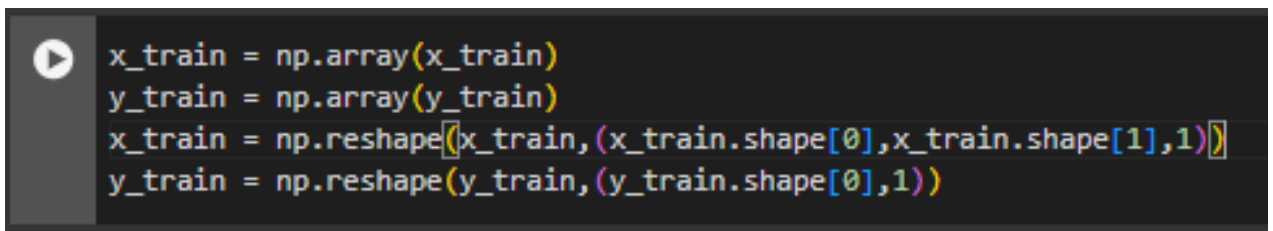
Hình 8: Chuẩn hóa minmax

- Dữ liệu gốc data sẽ được chuẩn hóa bằng cách áp dụng fit_transform. Phương thức này sẽ thực hiện hai việc:
 - fit: Tính toán các tham số cần thiết cho việc chuẩn hóa (cụ thể là giá trị nhỏ nhất và lớn nhất trong dữ liệu gốc).
 - transform: Áp dụng quá trình chuẩn hóa lên dữ liệu, chuyển đổi các giá trị của data để nằm trong khoảng từ 0 đến 1
 - Tác dụng của việc này là vì LSTM sử dụng các hàm kích hoạt như sigmoid và tanh, các hàm này hoạt động tốt nhất khi các giá trị đầu vào nằm trong khoảng nhất định (ví dụ: [-1, 1] cho tanh và [0, 1] cho sigmoid). Nếu dữ liệu không được chuẩn hóa, các giá trị lớn có thể khiến gradient bị bão hòa và làm chậm quá trình học hoặc thậm chí gây ra lỗi gradient biến mất cũng như là giảm thiểu tác động của các giá trị outlier và giá trị nhiễu

```
[19] #tạo vòng lặp các giá trị
      x_train,y_train=[],[]
      for i in range(50,len(train_data)):
          x_train.append(sc_train[i-50:i,0]) #lấy 50 giá đóng cửa liên tục
          y_train.append(sc_train[i,0]) #lấy ra giá đóng cửa ngày hôm sau
```

Hình 9: Lấy giá trị huấn luyện và giá trị dự báo

- Tạo ra 2 danh sách trống `x_train` và `y_train`
- Với mảng `x_train` thì chúng ta sẽ lưu trữ giá trị của 50 ngày trong dữ liệu ví dụ mảng đầu tiên chứa giá trị từ 1 đến 50 thì mảng thứ hai sẽ chứa 2 đến 51 lần lượt như vậy cho đến hết dữ liệu của tệp `train_data`. Đối với mảng `y_train` thì mảng này sẽ lưu giá trị của ngày thứ `i` ví dụ giá trị đầu tiên của tệp `x_train` là 1-50 thì giá trị sẽ được đưa vào `y_train` là 51.
- Mục đích của tệp `x_train` sẽ làm đầu vào cho mô hình LSTM, còn mục đích của tệp `y_train` là làm mục tiêu cho dự báo.
- Bằng cách lấy 50 giá trị đóng cửa làm đầu vào và giá trị đóng cửa của ngày tiếp theo làm mục tiêu làm để huấn luyện mô hình sẽ học cách dự báo dựa trên các mẫu thời gian trước đó.



```

x_train = np.array(x_train)
y_train = np.array(y_train)
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
y_train = np.reshape(y_train, (y_train.shape[0], 1))

```

Hình 10: Chuyển giá trị thành mảng 3 chiều cho huấn luyện và mảng 2 chiều cho dự báo

- Tiến hành chuyển đổi `x_train` và `y_train` thành mảng Numpy
- Sau đó sử dụng `reshape` để thay đổi `x_train` thành mảng 3 chiều và `y_train` thành mảng 2 chiều
- Mục đích của việc này là do mô hình LSTM trong Keras yêu cầu dữ liệu đầu vào có dạng (`batch_size`, `timesteps`, `features`), trong đó:
 - `batch_size`: Số lượng mẫu trong mỗi batch (có thể bỏ qua nếu không sử dụng `batch_size`).
 - `timesteps`: Số lượng bước thời gian (ở đây là 50).
 - `features`: Số lượng tính năng tại mỗi bước thời gian (ở đây là 1).
- Việc chuyển đổi `y_train` thành mảng 2 chiều với kích thước (số mẫu, 1) đảm bảo rằng các giá trị mục tiêu cũng có định dạng nhất quán, giúp mô hình dễ dàng học cách dự báo giá trị mục tiêu từ dữ liệu đầu vào.


```

[23] #xây dựng mô hình
model = Sequential() #tạo lớp mạng cho dữ liệu đầu vào
#2 lớp LSTM
model.add(LSTM(units=128,input_shape=(x_train.shape[1],1),return_sequences=True))
model.add(LSTM(units=64))
model.add(Dropout(0.5)) #loại bỏ 1 số đơn vị tránh học tủ (overfitting)
model.add(Dense(1)) #output đầu ra 1 chiều
#đo sai số tuyệt đối trung bình có sử dụng trình tối ưu hóa adam
model.compile(loss='mean_absolute_error',optimizer='adam')

```

Hình 11: Số lượng lớp

- Sequential(): Tạo một mô hình tuần tự, trong đó các lớp sẽ được xếp chồng lên nhau theo thứ tự.
- LSTM(units=128, input_shape=(x_train.shape[1], 1), return_sequences=True): Thêm lớp LSTM đầu tiên với 128 đơn vị (neurons).
- LSTM(units=64): Thêm lớp LSTM thứ hai với 64 đơn vị (neurons)
- Dropout(0.5): Thêm lớp Dropout với tỷ lệ dropout là 0.5.
 - Dropout giúp giảm overfitting bằng cách ngẫu nhiên loại bỏ một tỷ lệ phần trăm (ở đây là 50%) các đơn vị (neurons) trong quá trình huấn luyện.
- Dense(1): Thêm lớp Dense với 1 đơn vị đầu ra (để dự báo một giá trị duy nhất).
- loss='mean_absolute_error': Sử dụng hàm mất mát là sai số tuyệt đối trung bình (MAE). MAE đo lường độ lệch tuyệt đối trung bình giữa các giá trị dự báo và giá trị thực tế.
- optimizer='adam': Adam là một thuật toán tối ưu hóa dựa trên gradient descent, phổ biến và hiệu quả trong việc huấn luyện các mô hình học sâu.

```

[ ] #huấn luyện mô hình
save_model = "save_model.h5"
best_model = ModelCheckpoint(save_model,monitor='loss',verbose=2,save_best_only=True,mode='auto')
model.fit(x_train,y_train,epochs=150,batch_size=50,verbose=2,callbacks=[best_model])

```

Epoch 137/150

Epoch 137: loss did not improve from 0.00221
90/90 - 12s - loss: 0.0023 - 12s/epoch - 135ms/step
Epoch 138/150

Epoch 138: loss did not improve from 0.00221
90/90 - 12s - loss: 0.0023 - 12s/epoch - 137ms/step
Epoch 139/150

Epoch 139: loss did not improve from 0.00221
90/90 - 12s - loss: 0.0023 - 12s/epoch - 128ms/step
Epoch 140/150

Hình 12: Huấn luyện mô hình

- Tiến hành lưu trữ mô hình tốt nhất ra một file
- Tiếp theo là theo dõi giá trị 'loss' và chỉ lưu mô hình nếu giá trị loss cải thiện so với các epoch trước đó
- Sau đó tiến hành huấn luyện mô hình bằng giá trị đầu vào và giá trị mục tiêu

```
#dữ liệu train
y_train = sc.inverse_transform(y_train) #giá thực
final_model = load_model("save_model.h5")
y_train_predict = final_model.predict(x_train) #dự đoán giá đóng cửa trên tập dữ liệu train
y_train_predict = sc.inverse_transform(y_train_predict) #giá dự đoán
```

140/140 [=====] - 7s 42ms/step

Hình 13: Dự đoán trên dữ liệu train

- Tiến hành dự đoán trên tập train

```
#xử lý dữ liệu test
test = df[len(train_data)-50:].values
test = test.reshape((-1,1))
sc_test = sc.transform(test)

x_test = []
for i in range(50,test.shape[0]):
    x_test.append(sc_test[i-50:i,0])
x_test = np.array(x_test)
x_test = np.reshape(x_test,(x_test.shape[0],x_test.shape[1],1))

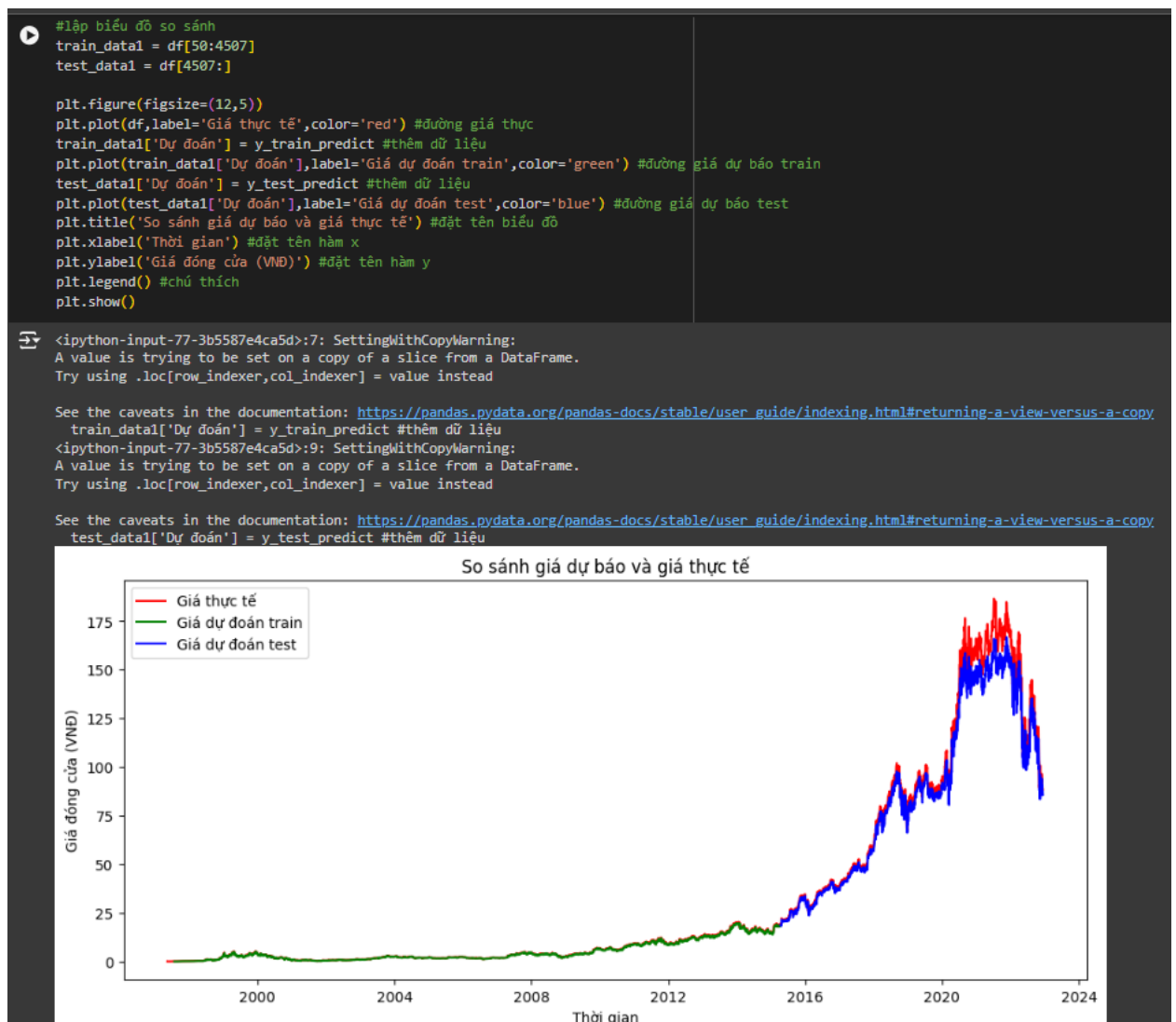
#dữ liệu test
y_test = data[4507:] #giá thực
y_test_predict = final_model.predict(x_test)
y_test_predict = sc.inverse_transform(y_test_predict) #giá dự đoán
```

61/61 [=====] - 2s 37ms/step

Hình 14: Xử lý và dự đoán trên dữ liệu test

- `df[len(train_data)-50:]` lấy 50 ngày dữ liệu cuối cùng từ `df`, nơi `df` là DataFrame chứa toàn bộ dữ liệu giá cổ phiếu.
- `test.reshape(-1,1)` thay đổi hình dạng của dữ liệu từ (50, 1) để chuẩn bị cho bước chuẩn hóa dữ liệu. Đây là bước chuyển dữ liệu thành một mảng hai chiều, với 50 hàng và 1 cột.
- Sau đó tiến hành chuyển hóa dữ liệu sang chuẩn hóa minmax
- Vòng lặp `for` tạo ra các chuỗi dữ liệu để đưa vào mô hình LSTM. Mỗi chuỗi có 50 bước (timesteps), được sử dụng để dự đoán giá cho bước tiếp theo.

- `sc_test[i-50:i,0]` lấy 50 bước dữ liệu liên tiếp để tạo thành một mẫu đầu vào cho mô hình.
- `np.array(x_test)` chuyển danh sách `x_test` thành mảng NumPy.
- `x_test` được reshape thành (số mẫu, 50, 1) để phù hợp với định dạng đầu vào của mô hình LSTM, với 50 là số bước thời gian và 1 là số đặc trưng.
- `y_test_predict = final_model.predict(x_test)` sử dụng mô hình LSTM đã huấn luyện (`final_model`) để dự đoán giá đóng cửa cho các chuỗi dữ liệu đầu vào `x_test`.
- `sc.inverse_transform(y_test_predict)` chuyển đổi dự đoán từ không gian chuẩn hóa trở về giá thực tế. `sc.inverse_transform` áp dụng phép biến đổi ngược lại của `MinMaxScaler` hoặc bất kỳ scaler nào bạn đã dùng.



Hình 15: Biểu đồ đường về giá trị thực tế và giá dự đoán

- Hiện thị giá trị thực tế và giá trị dự đoán của tập train và tập test

```
#r2
print('Độ phù hợp tập train:',r2_score(y_train,y_train_predict))
#mae
print('Sai số tuyệt đối trung bình trên tập train:',mean_absolute_error(y_train,y_train_predict))
#mae
print('Phần trăm sai số tuyệt đối trung bình tập train:',mean_absolute_percentage_error(y_train,y_train_predict))
```

Độ phù hợp tập train: 0.9970145015299093
Sai số tuyệt đối trung bình trên tập train: 0.20370921317995466
Phần trăm sai số tuyệt đối trung bình tập train: 0.06829966597313117

Hình 16: Chỉ số đánh giá của dữ liệu train

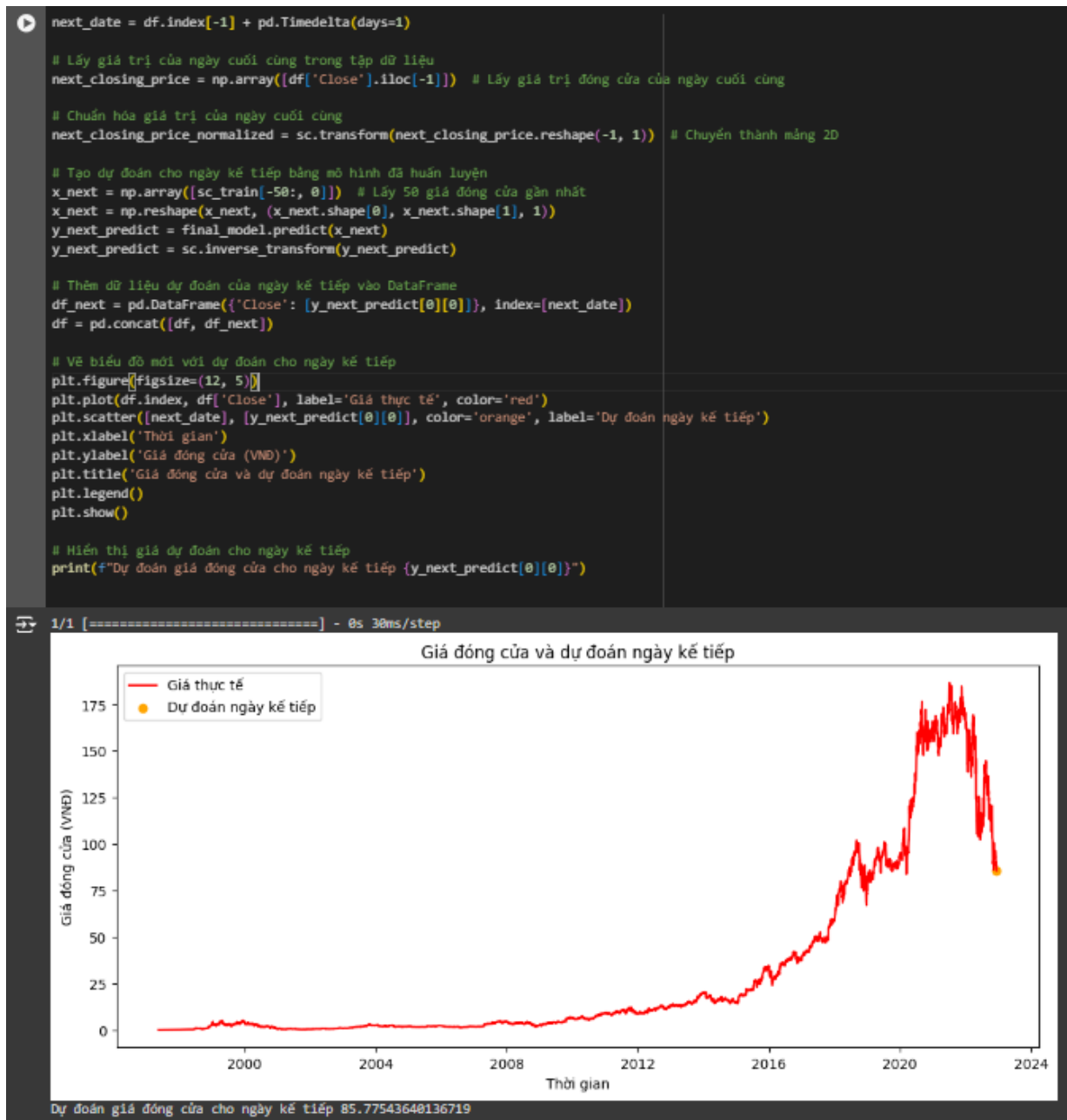
- Chỉ số đánh giá cho dữ liệu train
 - $R^2 = 0.997$ mô hình LSTM của bạn giải thích được khoảng 99.7% biến động của giá trị thực trên tập huấn luyện. Đây là một chỉ số rất cao, cho thấy mô hình có độ chính xác rất tốt trên tập huấn luyện.
 - Với $MAE = 0.20$, sai số trung bình giữa giá trị dự đoán và giá trị thực trên tập huấn luyện là khoảng 0.20. Đây là một giá trị khá nhỏ, cho thấy mô hình dự đoán giá trị rất gần với giá trị thực.
 - Với $MAPE = 0.07$ (7%), sai số trung bình dưới dạng phần trăm là khoảng 5.42%. Điều này cho thấy rằng dự đoán của mô hình chỉ sai lệch khoảng 5.42% so với giá trị thực, là một mức sai số thấp và chấp nhận được trong nhiều trường hợp.
- Kết luận
 - Mô hình LSTM của bạn có hiệu suất rất cao trên tập huấn luyện, với các chỉ số R^2 , MAE, và MAPE đều cho thấy sự phù hợp và độ chính xác tốt.
 - Tuy nhiên, cần kiểm tra hiệu suất của mô hình trên tập dữ liệu kiểm tra (test data) để đảm bảo rằng mô hình không bị overfitting (quá khớp) và có thể tổng quát hóa tốt trên các dữ liệu mới.

```
[ ] #r2
print('Độ phù hợp tập test:',r2_score(y_test,y_test_predict))
#mae
print('Sai số tuyệt đối trung bình trên tập test (VNĐ):',mean_absolute_error(y_test,y_test_predict))
#mae
print('Phần trăm sai số tuyệt đối trung bình tập test:',mean_absolute_percentage_error(y_test,y_test_predict))
```

Độ phù hợp tập test: 0.9757878365614301
Sai số tuyệt đối trung bình trên tập test (VNĐ): 5.305837470181268
Phần trăm sai số tuyệt đối trung bình tập test: 0.04714974448724023

Hình 17: Chỉ số đánh giá về dữ liệu test

- Chỉ số đánh giá cho dữ liệu test:
 - Giá trị R^2 gần 1 (0.975) cho thấy mô hình LSTM giải thích được khoảng 97.5% biến động của giá trị thực trên tập kiểm tra. Đây là một chỉ số rất cao, cho thấy mô hình có hiệu suất tốt trên tập dữ liệu kiểm tra.
 - MAE cho biết sai số trung bình giữa giá trị dự đoán và giá trị thực là khoảng 5.30. Đây là một giá trị tương đối nhỏ, cho thấy dự đoán của mô hình khá chính xác.
 - MAPE cho biết sai số trung bình dưới dạng phần trăm là khoảng 4,7%. Điều này cho thấy rằng dự đoán của mô hình chỉ sai lệch khoảng 4.7% so với giá trị thực, là một mức sai số chấp nhận được.
- Kết luận:
 - Hiệu suất mô hình LSTM trên tập kiểm tra cũng rất cao, với R^2 cao (0.975) và các chỉ số MAE, MAPE đều cho thấy sự phù hợp và độ chính xác tốt.
 - Mô hình không chỉ hoạt động tốt trên tập huấn luyện mà còn có khả năng tổng quát tốt trên dữ liệu mới (tập kiểm tra), cho thấy mô hình không bị overfitting.



Hình 18: Biểu đồ đường về dự đoán giá ngày tiếp theo

Dự đoán cho ngày tiếp theo.

3.2. Tối Ưu Hóa Danh Mục Đầu Tư

3.2.1. Thu Thập Dữ Liệu

- Dữ liệu của bài toán này được thu thập nhiều công ty chứng khoán để tạo thành một danh mục bao gồm các công ty APPLE(AAPL), AMAZON(AMZN), BERKSHIRE HATHAWAY(BRK-A), GOOGLE(GOOG), NOKIA(NOK), CHIPBOND TECHNOLOGY(NFST), NVIDIA(NVDA), TESLA(TSLA)

Date	Low	Open	Volume	High	Close	Adjusted Close
15-05-1997	0.096354	0.121875003	1443120000	0.125	0.097916998	0.097916998
16-05-1997	0.085417002	0.098438002	294000000	0.098958001	0.086457998	0.086457998
19-05-1997	0.081249997	0.088021003	122136000	0.088541999	0.085417002	0.085417002
20-05-1997	0.081771001	0.086457998	109344000	0.087499999	0.081771001	0.081771001
21-05-1997	0.068750001	0.081771001	377064000	0.082291998	0.071354002	0.071354002
22-05-1997	0.065624997	0.071874999	235536000	0.072396003	0.069792002	0.069792002
23-05-1997	0.066666998	0.070312999	318744000	0.076041996	0.075000003	0.075000003
27-05-1997	0.072916999	0.075521	173952000	0.082291998	0.079167001	0.079167001
28-05-1997	0.076563001	0.081249997	91488000	0.081771001	0.076563001	0.076563001
29-05-1997	0.073958002	0.077082999	69456000	0.077082999	0.075259998	0.075259998
30-05-1997	0.073958002	0.075000003	51888000	0.075521	0.075000003	0.075000003
2/6/1997	0.075000003	0.075521	11832000	0.076563001	0.075521	0.075521
3/6/1997	0.073958002	0.076563001	23664000	0.076563001	0.073958002	0.073958002
4/6/1997	0.069792002	0.073958002	61608000	0.074478999	0.070832998	0.070832998
5/6/1997	0.068750001	0.070832998	113448000	0.077082999	0.077082999	0.077082999
6/6/1997	0.075521	0.075781003	156144000	0.085417002	0.082813002	0.082813002
9/6/1997	0.082813002	0.082813002	47040000	0.085417002	0.084375001	0.084375001
10/6/1997	0.076563001	0.085417002	109176000	0.085417002	0.079167001	0.079167001
11/6/1997	0.076563001	0.079687998	23760000	0.080208004	0.077082999	0.077082999
12/6/1997	0.077604003	0.079167001	32640000	0.082291998	0.080208004	0.080208004
13-06-1997	0.079167001	0.081249997	13872000	0.081249997	0.079167001	0.079167001
16-06-1997	0.078125	0.080208004	18264000	0.080208004	0.078645997	0.078645997
17-06-1997	0.07474	0.079948001	94128000	0.079948001	0.075259998	0.075259998
18-06-1997	0.075000003	0.076041996	49296000	0.076823004	0.075521	0.075521
19-06-1997	0.075000003	0.075521	20064000	0.076563001	0.075521	0.075521
20-06-1997	0.075000003	0.076563001	67752000	0.077604003	0.076301999	0.076301999

Hình 19: Dữ liệu chứng khoán công ty Amazon

- Đây là hình ảnh dữ liệu của công ty AMZN và các công ty khác tương tự.
- Về dữ liệu ta có:
 - Date: Ngày giao dịch chứng khoán
 - Low: Giá thấp nhất trong ngày mà cổ phiếu có thể đạt được (Giá sàn)
 - Open: Giá mở cửa trong ngày giao dịch
 - Volume: Khối lượng giao dịch của cổ phiếu trong ngày
 - High: Mức giá cao nhất trong ngày mà cổ phiếu có thể đạt được (Giá trần)
 - Close: Giá đóng cửa của ngày giao dịch
 - Adjusted Close: Giá điều chỉnh sau khi loại bỏ các tác động về cổ tức,...

3.2.2. *Tiền Xử Lý Dữ Liệu Trên PYTHON*

```

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
AMZN=pd.read_csv('AMZN.csv')
BRKA=pd.read_csv('BRK-A.csv')
GOOG=pd.read_csv('GOOG.csv')
AAPL=pd.read_csv('AAPL.csv')
TSLA=pd.read_csv('TSLA.csv')
NSFT=pd.read_csv('MSFT.csv')
NOK=pd.read_csv('NOK.csv')
NVDA=pd.read_csv('NVDA.csv')
stock_dataframes = {
    'AMZN': AMZN,
    'BRK-A': BRKA,
    'GOOG': GOOG,
    'AAPL': AAPL,
    'TSLA': TSLA,
    'NSFT': NSFT,
    'NOK': NOK,
    'NVDA': NVDA
}
# Khoảng thời gian cần lọc
start_date = '2010-06-29'
end_date = '2022-09-30'

# Xử lý dữ liệu cho từng cổ phiếu
for ticker, df in stock_dataframes.items():
    # Explicitly specify the date format
    df['Date'] = pd.to_datetime(df['Date'], format='%d-%m-%Y')
    df.set_index('Date', inplace=True)
    df.sort_index(inplace=True)
    # Lọc dữ liệu theo khoảng thời gian
    df = df.loc[start_date:end_date]

    # Chỉ giữ lại các cột cần thiết
    df = df[['Adjusted Close']] # Giữ lại cột 'Adj Close' để tính toán lợi nhuận
    # Cập nhật lại DataFrame
    stock_dataframes[ticker] = df
stock_dataframes

```

{'AMZN':		Adjusted Close
Date		
2010-06-29		5.430500
2010-06-30		5.463000
2010-07-01		5.548000
2010-07-02		5.457000
2010-07-06		5.503000
...		...
2022-09-26		115.150002
2022-09-27		114.410004
2022-09-28		118.010002
2022-09-29		114.800003
2022-09-30		113.000000

[3087 rows x 1 columns], 'BRK-A':		Adjusted Close
Date		
2010-06-29		120199.0
2010-06-30		120000.0
2010-07-01		118095.0
2010-07-02		115500.0
2010-07-06		116505.0
...		...
2022-09-26		399128.0
2022-09-27		401490.0
2022-09-28		410705.0
2022-09-29		406700.0
2022-09-30		406470.0

Hình 20: Xử lý dữ liệu và chuyển cột 'Date' thành Index và giữ lại cột cần thiết

- Ta sẽ đọc dữ liệu của 8 dataset của mỗi công ty
- Đưa dữ liệu của các dataframe vào một dictionary để dễ dàng quản lí
- Xác định khoảng thời gian cần lọc
- Xử lý dữ liệu cho từng cổ phiếu
 - Chuyển đổi cột 'Date' sang định dạng datetime.
 - Đặt cột 'Date' làm chỉ mục.
 - Sắp xếp lại DataFrame theo chỉ mục 'Date'.
 - Lọc dữ liệu theo khoảng thời gian đã xác định.
 - Giữ lại cột 'Adjusted Close' để tính toán lợi nhuận.


```

print(AMZN.isnull().sum())
print(BRKA.isnull().sum())
print(GOOG.isnull().sum())
print(AAPL.isnull().sum())
print(TSLA.isnull().sum())
print(NSFT.isnull().sum())
print(NOK.isnull().sum())
print(NVDA.isnull().sum())

```

```

Low      0
Open      0
Volume    0
High      0
Close     0
Adjusted Close  0
dtype: int64
Low      0
Open      0
Volume    0
High      0
Close     0
Adjusted Close  0

```

Hình 21: Kiểm tra dữ liệu null

- Kiểm tra dữ liệu null cho mỗi dataframe

3.2.3. Lợi Nhuận và Rủi Ro Của Danh Mục

```

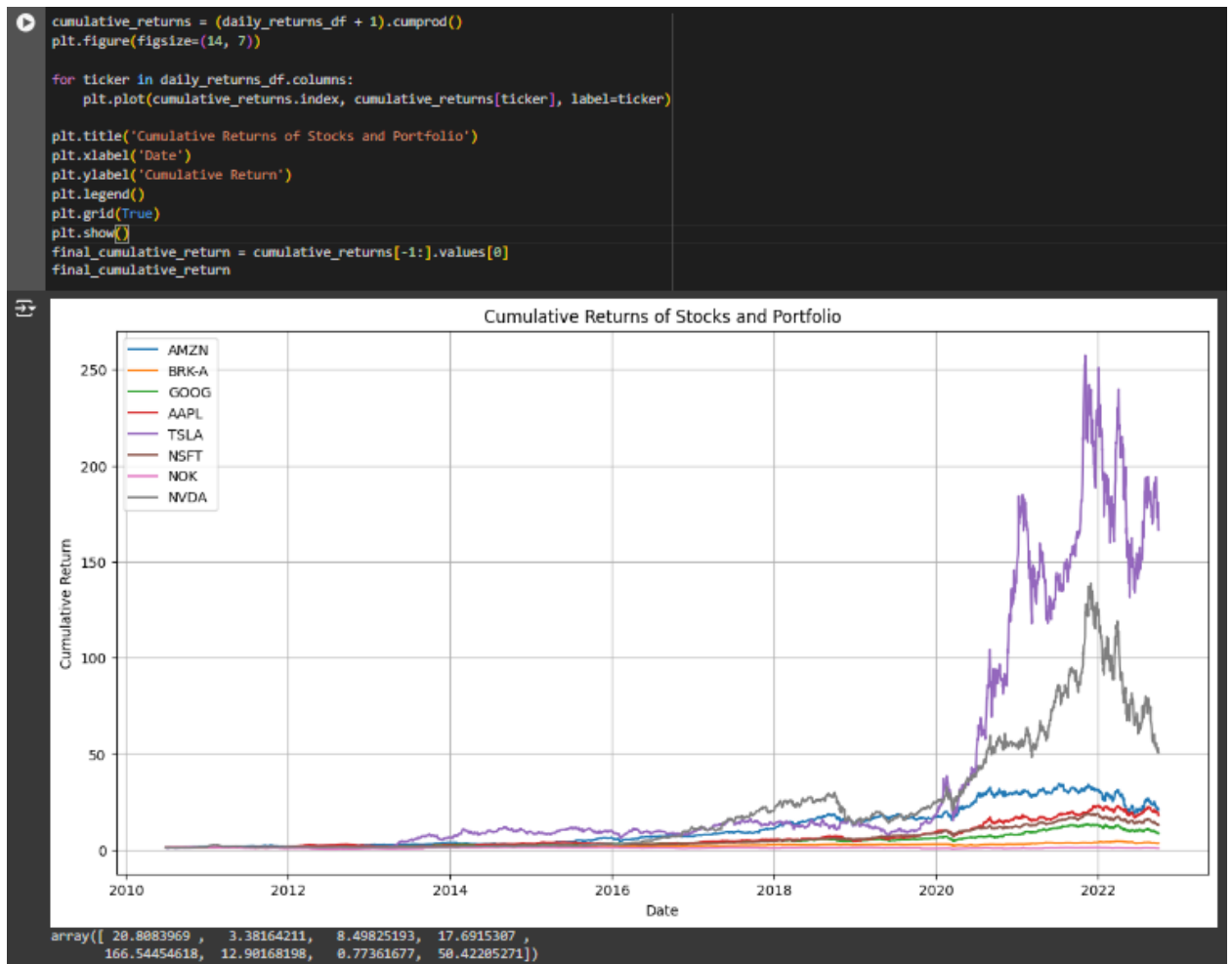
daily_returns = {ticker: df['Adjusted Close'].pct_change().dropna() for ticker, df in stock_dataframes.items()}
# Chuyển đổi dictionary thành DataFrame
daily_returns_df = pd.DataFrame(daily_returns)
daily_returns_df.head(5)

```

	AMZN	BRK-A	GOOG	AAPL	TSLA	NSFT	NOK	NVDA
Date								
2010-06-30	0.005985	-0.001656	-0.020495	-0.018113	-0.002511	-0.012870	0.016209	-0.025763
2010-07-01	0.015559	-0.015875	-0.012271	-0.012126	-0.078473	0.006519	0.025767	0.016650
2010-07-02	-0.016402	-0.021974	-0.006690	-0.006197	-0.125683	0.004750	0.008373	-0.012524
2010-07-06	0.008430	0.008701	-0.001100	0.006844	-0.160937	0.023636	-0.005931	-0.010732
2010-07-07	0.030620	0.029046	0.032403	0.040381	-0.019243	0.020151	0.042959	0.048323

Hình 22: Tính toán lợi nhuận mỗi ngày của các cổ phiếu

- Sử dụng phương thức `.pct_change().dropna()`: Phương thức này tính toán lợi nhuận hàng ngày dưới dạng phần trăm thay đổi của giá đóng cửa điều chỉnh. Cụ thể, nó tính toán tỉ lệ thay đổi giữa giá trị hiện tại và giá trị trước đó. Sau đó tiến hành xóa bỏ các giá bị rỗng
- Tiến hành chuyển đổi từ Dictionary sang DataFrame



Hình 23: Tính toán lợi nhuận tích lũy và vẽ biểu đồ đường cho các cổ phiếu

- Sử dụng phương thức cumprod(): Phương thức này tính toán tích lũy lợi nhuận của tất cả các giá trị trong chuỗi.
- Sử dụng giá trị của được lưu vào trong biến để vẽ biểu đồ lợi nhuận tích lũy của mỗi cổ phiếu

```

mean_returns = daily_returns_df.mean() * 252

# Calculate covariance matrix
cov_matrix = daily_returns_df.cov() * 252

# Calculate volatility (annualized standard deviation)
volatility = daily_returns_df.std() * np.sqrt(252)

# Print the results
print("Mean Returns:\n", mean_returns)
print("\nCovariance Matrix:\n", cov_matrix)
print("\nVolatility:\n", volatility)

```

Mean Returns:

AMZN	0.300618
BRK-A	0.117160
GOOG	0.210051
AAPL	0.274834
TSLA	0.578378
NSFT	0.241898
NOK	0.077491
NVDA	0.416937

dtype: float64

Covariance Matrix:

	AMZN	BRK-A	GOOG	AAPL	TSLA	NSFT	NOK	NVDA
AMZN	0.105618	0.022149	0.051522	0.044315	0.064220	0.046324	0.036408	0.066137
BRK-A	0.022149	0.035327	0.024126	0.024074	0.027875	0.025021	0.028110	0.032966
GOOG	0.051522	0.024126	0.070916	0.041570	0.050600	0.043842	0.034117	0.059882
AAPL	0.044315	0.024074	0.041570	0.080118	0.056087	0.042526	0.037942	0.061861
TSLA	0.064220	0.027875	0.050600	0.056087	0.322667	0.051705	0.048777	0.090823
NSFT	0.046324	0.025021	0.043842	0.042526	0.051705	0.065961	0.035519	0.063834
NOK	0.036408	0.028110	0.034117	0.037942	0.048777	0.035519	0.197035	0.052368
NVDA	0.066137	0.032966	0.059882	0.061861	0.090823	0.063834	0.052368	0.194588

Volatility:

AMZN	0.324990
BRK-A	0.187955
GOOG	0.266301
AAPL	0.283050
TSLA	0.568038
NSFT	0.256829
NOK	0.443886
NVDA	0.441121

Hình 24: Tính toán lợi nhuận theo năm, ma trận phương sai, giá trị độ lệch chuẩn

- Tính lợi nhuận trung bình của mỗi cổ phiếu theo năm bằng cách tính trung bình của lợi nhuận theo ngày và nhân với 255 ngày giao dịch
- Tính toán ma trận phương sai của các cổ phiếu theo năm dựa trên lợi nhuận theo ngày
- Tính độ lệch chuẩn của các cổ phiếu

```
[13] num_portfolios = 100

# Khởi tạo mảng để lưu trữ kết quả
# Store weights for each portfolio in a separate list
results = np.zeros((2, num_portfolios))
portfolio_weights = []

# Vòng lặp để tạo ngẫu nhiên trọng số danh mục đầu tư và tính toán lợi nhuận và độ biến động
for i in range(num_portfolios):
    weights = np.random.random(len(daily_returns_df.columns))
    weights /= np.sum(weights)

    # Tính toán lợi nhuận và độ biến động của danh mục đầu tư
    portfolio_return = np.sum(mean_returns * weights) # Annualized return
    portfolio_volatility = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights)))
    # Lưu trữ kết quả
    results[0, i] = portfolio_return
    results[1, i] = portfolio_volatility
    portfolio_weights.append(weights) # Append weights to the list

# Chuyển đổi kết quả thành DataFrame để dễ dàng hiển thị
results_df = pd.DataFrame(results.T, columns=['Return', 'Volatility'])
# Add weights to the DataFrame
results_df['Weights'] = portfolio_weights

# Hiển thị một vài kết quả đầu tiên
print(results_df.head())
print(results_df.tail())
```

	Return	Volatility	Weights
0	0.199429	0.205534	[0.09836100267773282, 0.2716680235066498, 0.11...
1	0.318277	0.261589	[0.11036242274706975, 0.12744095321408608, 0.0...
2	0.306752	0.252070	[0.07397416430178182, 0.04228663136405609, 0.1...
3	0.316857	0.253649	[0.09210565717535105, 0.15921837384997847, 0.0...
4	0.213446	0.228418	[0.041732099447586886, 0.11360360011548627, 0....
	Return	Volatility	Weights
95	0.263768	0.240987	[0.127884779190633, 0.07342888149397682, 0.255...
96	0.236986	0.250988	[0.1911912989648646, 0.02314611636345554, 0.22...
97	0.293312	0.257484	[0.21132779673295093, 0.04951809942926381, 0.1...
98	0.294517	0.260037	[0.051860160126006055, 0.12334637660911668, 0....
99	0.273443	0.228540	[0.0630502601924782, 0.19799315477805893, 0.00...

Hình 25: Tạo ra Weights, tính toán lợi nhuận và giá trị volatility của mỗi portfolio

- Khởi tạo các biến để lưu trữ kết quả
 - Tạo biến để chỉ định số lượng portfolio muốn tạo ra
 - Tạo mảng 2D để lưu trữ lợi nhuận và độ biến động của mỗi danh mục đầu tư. Kích thước của mảng là 2 hàng và num_portfolios cột.
 - Tạo mảng rỗng để lưu trữ các trọng số của danh mục đầu tư
- Tạo ngẫu nhiên trọng số danh mục đầu tư
 - `weights = np.random.random(len(daily_returns_df.columns))`: Tạo một mảng trọng số ngẫu nhiên cho các cổ phiếu trong danh mục đầu tư
 - `weights /= np.sum(weights)`: Chuẩn hóa trọng số để tổng trọng số bằng 1
- Tính toán lợi nhuận và độ biến động

- `portfolio_return = np.sum(mean_returns * weights)`: Tính toán lợi nhuận hàng năm của danh mục đầu tư bằng cách nhân lợi nhuận trung bình hàng năm của từng cổ phiếu với trọng số tương ứng và tổng hợp lại.
- `portfolio_volatility=np.sqrt(np.dot(weights.T,np.dot(cov_matrix, weights)))`: Tính toán độ biến động của danh mục đầu tư bằng công thức độ lệch chuẩn có trọng số (công thức này sử dụng ma trận hiệp phương sai và trọng số của từng cổ phiếu).
- Lưu lợi nhuận theo năm và độ biến động vào Dataframe

3.2.4. Tỷ Lệ Sharpe

- *Tỷ lệ Sharpe* là một thước đo xem lợi nhuận thu được là bao nhiêu trên một đơn vị rủi ro khi đầu tư vào một tài sản hay đầu tư theo một chiến lược kinh doanh.
- Công thức tính

$$\text{Tỷ lệ Sharpe} = (R_p - R_f) / \sigma_p$$

- Trong đó:
 - R_p là tỷ suất lợi nhuận của danh mục đầu tư
 - R_f là tỷ suất lợi nhuận phi rủi ro
 - σ_p là độ lệch chuẩn của tỷ suất lợi nhuận vượt quá của danh mục

```

risk_free_rate=0.0231
sharpe_ratio = (results_df['Return'] - risk_free_rate) / results_df['Volatility']
results_df['Sharpe Ratio'] = sharpe_ratio
max_sharpe_idx = results_df['Sharpe Ratio'].idxmax()
max_sharpe_portfolio = results_df.iloc[max_sharpe_idx]
print(results_df)
print("Portfolio with the highest Sharpe Ratio:\n")
print(f"Return: {max_sharpe_portfolio['Return']:.5f}")
print(f"Volatility: {max_sharpe_portfolio['Volatility']:.5f}")
print(f"Sharpe Ratio: {max_sharpe_portfolio['Sharpe Ratio']:.5f}")
print(f"Weights:\n{dict(zip(daily_returns_df.columns, max_sharpe_portfolio['Weights']))}")

```

```

Return Volatility Weights \
0 0.199429 0.205534 [0.0983610026773282, 0.2716688235066498, 0.11...
1 0.318277 0.261589 [0.1103624224706975, 0.2744095321486686, 0.1...
2 0.300752 0.252070 [0.07397416430178182, 0.04228663136405609, 0.1...
3 0.316857 0.253649 [0.09218565717535185, 0.15921837384997847, 0.0...
4 0.213446 0.228410 [0.041732099447586886, 0.11368360011548627, 0.0...
...
95 0.263768 0.240987 [0.127884779190633, 0.07342888149397682, 0.255...
96 0.236886 0.250988 [0.1911912989648646, 0.0214611636345354, 0.22...
97 0.293312 0.257404 [0.2112779673295905, 0.04091809942926381, 0.1...
98 0.294517 0.260037 [0.051860160126080855, 0.12334637660911668, 0.0...
99 0.273443 0.228540 [0.0638562661924782, 0.19799315477805893, 0.00...

Sharpe Ratio
0 0.857989
1 1.128398
2 1.125292
3 1.158123
4 0.833326
...
95 0.998075
96 0.852175
97 1.049432
98 1.043764
99 1.095481

[100 rows x 4 columns]
Portfolio with the highest Sharpe Ratio:
Return: 0.34019
Volatility: 0.26374
Sharpe Ratio: 1.20228
Weights:
{'ARKH': 0.10056109396086743, 'BRK-A': 0.06133255207282404, 'GOOG': 0.017866962710368504, 'AAPL': 0.24152739861443573, 'TSLA': 0.20983657413904241, 'MSFT': 0.21184238511570733, 'NOK': 0.01603921310462872, 'WDA': 0.140093828282125}

```

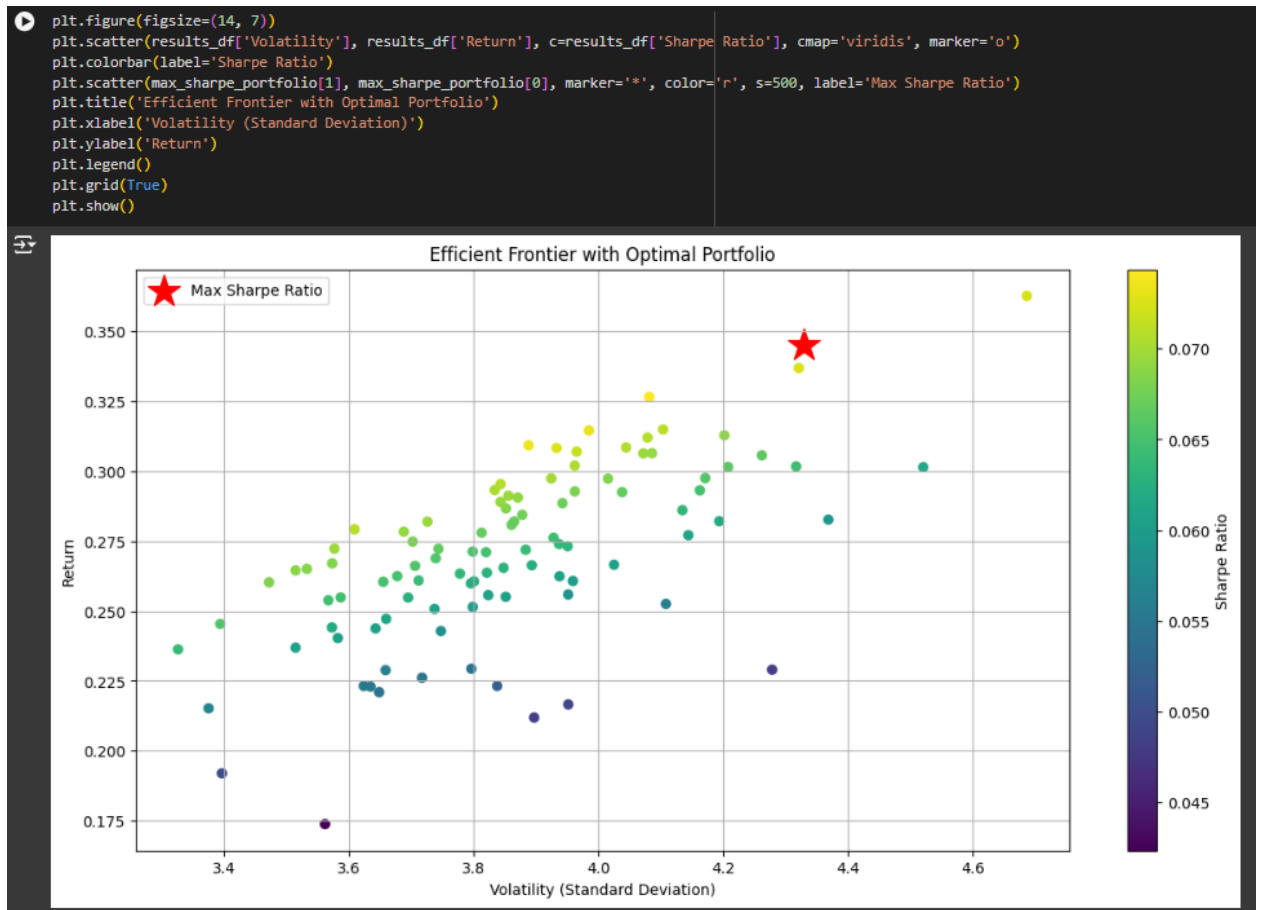
Hình 26: Tính Sharpe Ratio của mỗi portfolio

- Tính toán Sharpe Ratio cho mỗi danh mục trong đó tỉ lệ phi rủi ro sẽ bằng tỉ lệ trái phiếu của chính phủ trong 10 năm
- Tìm ra giá trị Sharpe Ratio lớn nhất trong tất cả portfolio
- Trả về giá trị của các chỉ số và portfolio tối ưu
- Dựa trên kết quả mới, chúng ta có thể phân tích các chỉ số như sau:
 - Return (Lợi nhuận): 0.34019
 - Ý nghĩa: Lợi nhuận hàng năm của danh mục đầu tư là 34.019%. Đây là mức lợi nhuận khá cao và hấp dẫn đối với nhà đầu tư.
 - Volatility (Độ biến động): 0.26374
 - Ý nghĩa: Độ biến động hàng năm của danh mục đầu tư là 26.374%. Mức độ biến động này tương đối thấp so với kết quả trước đó (432.999%), cho thấy danh mục đầu tư này có mức độ rủi ro thấp hơn.
 - Sharpe Ratio: 1.20228
 - Ý nghĩa: Sharpe Ratio là 1.20228. Sharpe Ratio đo lường hiệu suất điều chỉnh rủi ro của danh mục đầu tư. Một Sharpe Ratio lớn hơn 1

cho thấy rằng danh mục đầu tư này có hiệu suất điều chỉnh rủi ro tốt, tức là lợi nhuận của nó đủ để bù đắp rủi ro.

- Weights (Trọng số):
 - AMZN: 0.10056
 - Trọng số của Amazon là khoảng 10.06%.
 - BRK-A: 0.0613
 - Trọng số của Berkshire Hathaway là khoảng 6.13%.
 - GOOG: 0.01786
 - Trọng số của Google là khoảng 1.79%.
 - AAPL: 0.2415
 - Trọng số của Apple là khoảng 24.15%.
 - TSLA: 0.2098
 - Trọng số của Tesla là khoảng 20.98%.
 - NSFT: 0.2118
 - Trọng số của Microsoft là khoảng 21.18%.
 - NOK: 0.0160
 - Trọng số của Nokia là khoảng 1.60%.
 - NVDA: 0.1409
 - Trọng số của Nvidia là khoảng 14.10%.
- Đánh giá tổng quan
 - Lợi nhuận cao: Danh mục đầu tư này có lợi nhuận hàng năm cao (34.019%), điều này hấp dẫn đối với nhà đầu tư.
 - Rủi ro thấp: Độ biến động hàng năm tương đối thấp (26.374%), cho thấy danh mục đầu tư có mức độ rủi ro thấp hơn so với kết quả trước đó.
 - Hiệu suất điều chỉnh rủi ro tốt: Sharpe Ratio cao hơn 1 (1.20228) cho thấy rằng lợi nhuận của danh mục đầu tư đủ để bù đắp rủi ro, điều này là tín hiệu tốt cho nhà đầu tư.
 - Phân bổ tài sản: Danh mục đầu tư có sự phân bổ khá cân bằng giữa các cổ phiếu lớn như Apple (24.15%), Tesla (20.98%), và Microsoft (21.18%). Các cổ phiếu khác như Amazon, Nvidia cũng chiếm tỷ lệ đáng kể, trong khi Google và Nokia chiếm tỷ lệ nhỏ hơn.

- Kết luận:
 - Danh mục đầu tư này có một sự kết hợp tốt giữa lợi nhuận cao và rủi ro thấp, dẫn đến Sharpe Ratio cao, làm cho nó trở thành một lựa chọn hấp dẫn đối với các nhà đầu tư tìm kiếm lợi nhuận điều chỉnh rủi ro tốt.

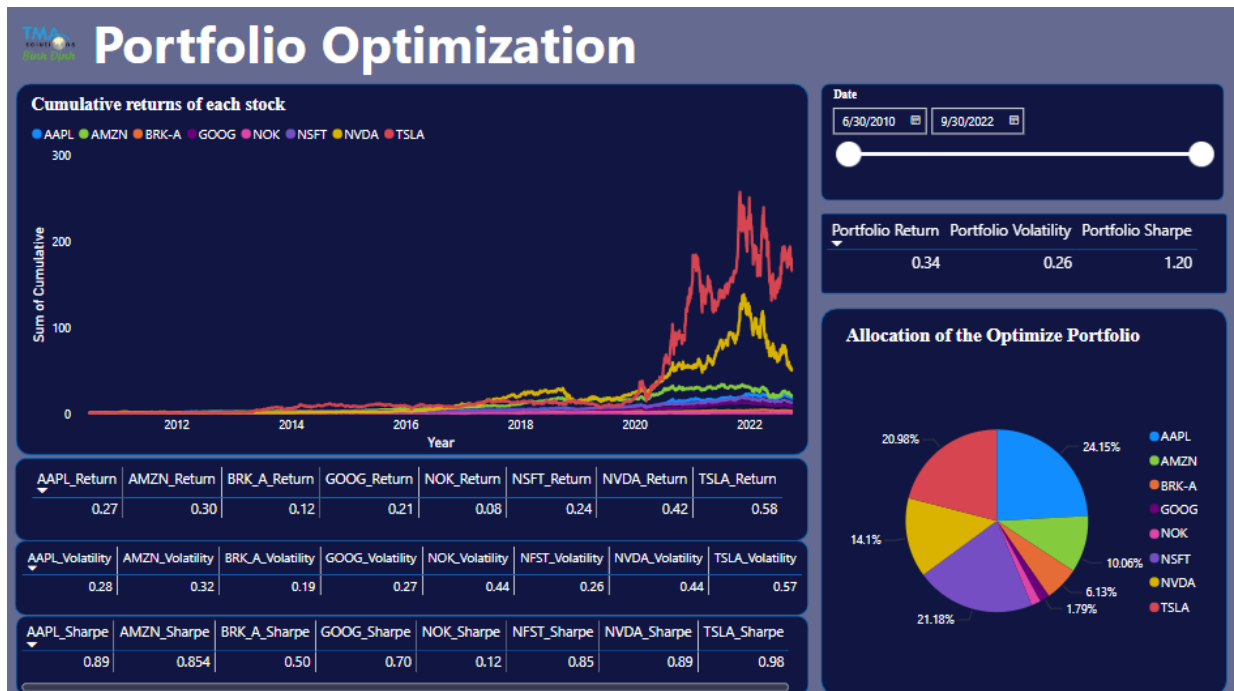


Hình 27: Biểu đồ Scatter thể hiện portfolio tối ưu nhất

- Vẽ biểu đồ Scatter để kiểm chứng lại xem portfolio đã được tìm thấy trên kết quả là tối ưu nhất

3.2.5. *Trực Quan Hóa*

- Dashboard:



Hình 28: Dashboard Portfolio Optimization

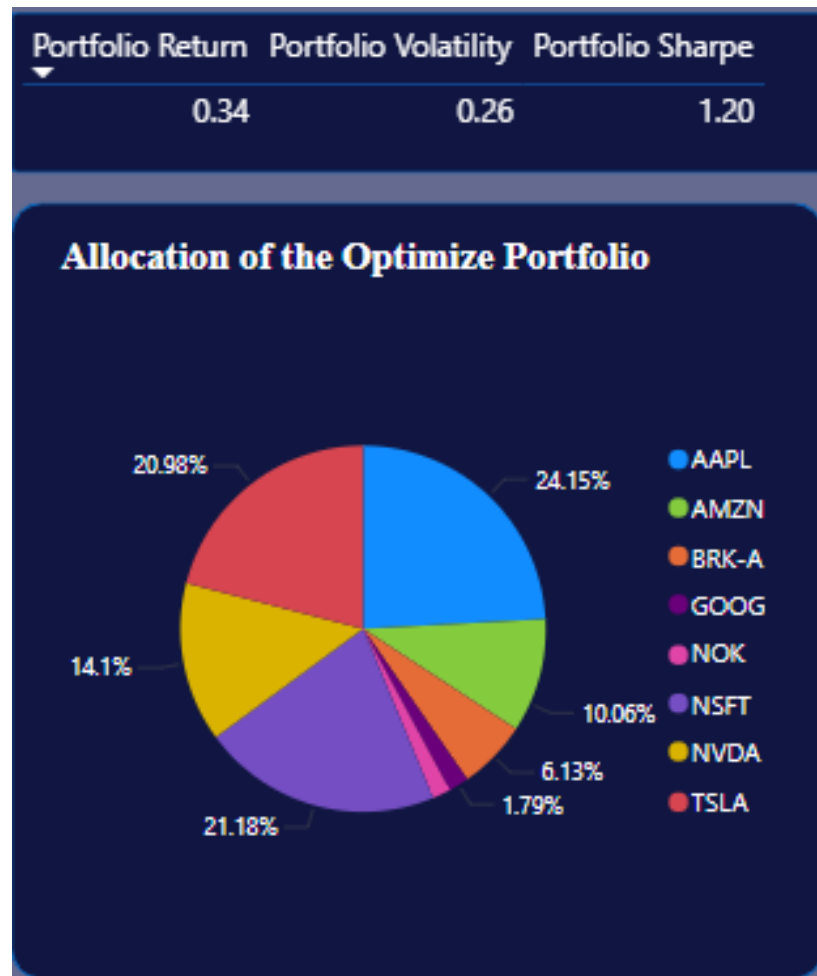
- Phân tích:



Hình 29: Biểu đồ về lợi nhuận tích lũy qua các năm

- Đối với biểu đồ đầu tiên trong Dashboard thì đây là biểu đồ lợi nhuận tích lũy qua các năm. Biểu đồ này cho chúng ta thấy là đối với 1\$ đầu tư vào năm 2010 thì đến năm 2022 bạn đã thu về được bao nhiêu tiền.
- Thì chúng ta có thể thấy cổ phiếu có lợi nhuận tích lũy cao nhất là công ty Tesla trong khi đó công ty mang về lợi nhuận thấp nhất là Nokia. Ngoài ra thì các công ty công nghệ khác cũng mang về lợi nhuận khá ấn tượng như là Nvidia, Amazon, Apple

- Nguyên nhân:
 - Có thể nói là dưới thời lãnh đạo của ông Elon Musk công ty Tesla đã trở thành công ty hàng đầu trong lĩnh vực xe điện. Lợi nhuận tích lũy của Tesla thể hiện sự tăng trưởng mạnh mẽ, đặc biệt là từ năm 2019 trở đi. Sự thành công của các mẫu xe như Model S, Model 3, Model X, và Model Y cùng với công nghệ tiên tiến về pin và hệ thống tự lái đã góp phần đáng kể vào sự tăng trưởng này. Ngoài ra, Tesla cũng mở rộng hoạt động sản xuất với các Gigafactory tại Mỹ, Trung Quốc, và Đức, giúp tăng năng lực sản xuất và phân phối.
 - Còn đối với Nokia, công ty này đã dần mất thị phần và sức cạnh tranh đối với các công ty sản xuất điện thoại khác như là Apple và Samsung nên cổ phiếu này không được kì vọng quá nhiều
 - Các cổ phiếu khác vẫn giữ được một mức tăng trưởng ổn định trong thị trường
- Qua đó cho chúng ta thấy rằng là nên tập trung phân bổ nguồn tiền vào các cổ phiếu công nghệ như là Tesla, Amazon, NVIDIA... Nhưng thị trường thì luôn hoạt động xoay quanh cổ phiếu dẫn đầu nếu như cổ phiếu dẫn đầu của nhóm ngành công nghệ có sự kiện gì bất lợi dẫn đến việc giảm giá thì sẽ kéo theo cả nhóm ngành cổ phiếu công nghệ sẽ giảm theo và chúng ta sẽ dễ bị thua lỗ. Bởi vì vậy chúng ta mới có câu nói “ Không nên bỏ hết trứng vào cùng một giỏ” câu nói này ngụ ý rằng không nên mua cùng một loại cổ phiếu vì vậy chúng ta cần phân bổ cho các nhóm ngành khác như cổ phiếu BRK-A và NSFT
- Tiếp theo chúng ta sẽ tiến hành đến với việc phân tích portfolio tối ưu nhất để xem đã có sự phân bổ hợp lí dựa trên lợi nhuận tích lũy của từng cổ phiếu.



Hình 30: Biểu đồ tròn về tỉ lệ phân bổ cổ phiếu và các chỉ số

- Dựa trên chỉ số của portfolio được phân bổ tối ưu nhất trong tổng số 100 portfolio phân bổ thì ta có thể thấy danh mục này mang về cho chúng ta một khoảng lợi nhuận cũng khá cao với 34% và cùng với đó là mức độ rủi ro thấp hơn khá nhiều so với lợi nhuận thu về được là 26% rõ ràng nhìn vào đây ta có thể mạnh dạn để phân bổ cổ phiếu theo như phân tích. Đối với chỉ số Sharpe thì chỉ số này lớn hơn 1 nên đây chính là giá trị tối ưu cho một chiến lược hiệu quả hoặc hiệu suất của danh mục đầu tư. Qua đó có thể cho ta thấy đây hoàn toàn là một danh mục tốt dựa trên số liệu nhưng trên thực tế thì còn phụ thuộc nhiều vào các yếu tố vi mô và vĩ mô.
- Đối với biểu đồ tròn bên dưới đây là biểu đồ phân bổ số lượng tiền của chúng ta cho mỗi cổ phiếu ở trong danh mục thì chúng ta có thể thấy các công ty công nghệ đều được phân bổ một lượng tiền rất lớn trong danh mục trong đó cao nhất là cổ phiếu của công ty Apple với 24.15% tiếp đến là các công ty như Tesla, Amazon, NVIDIA đều được phân bổ khá lớn, tất cả những sự phân bổ này đều khá tốt. Nhưng vấn đề

là nằm ở các công ty như Nokia , Google và cả BRK-A liệu đây có phải là cổ phiếu đáng kì vọng mà chúng ta nên đầu tư. Liệu khi chúng ta bỏ các cổ phiếu này ra và khi thay các cổ phiếu khác vào thì danh mục của chúng ta có tối ưu hơn không?

- Thì để tìm hiểu sâu hơn vào việc đó thì chúng ta sẽ đến với việc so sánh hiệu suất giữa từng cổ phiếu với nhau để tiến hành cơ cấu danh mục hợp lí.

AAPL_Return	AMZN_Return	BRK_A_Return	GOOG_Return	NOK_Return	NSFT_Return	NVDA_Return	TSLA_Return
0.27	0.30	0.12	0.21	0.08	0.24	0.42	0.58
AAPL_Volatility	AMZN_Volatility	BRK_A_Volatility	GOOG_Volatility	NOK_Volatility	NSFT_Volatility	NVDA_Volatility	TSLA_Volatility
0.28	0.32	0.19	0.27	0.44	0.26	0.44	0.57
AAPL_Sharpe	AMZN_Sharpe	BRK_A_Sharpe	GOOG_Sharpe	NOK_Sharpe	NSFT_Sharpe	NVDA_Sharpe	TSLA_Sharpe
0.89	0.854	0.50	0.70	0.12	0.85	0.89	0.98

Hình 31: Các chỉ số của mỗi công ty

- Đối với các số liệu này thì các cổ phiếu như Tesla, NVIDIA, Apple, AMZN và NSFT vẫn có một tỉ lệ Sharpe khá cao tất cả đều trên 0.85 điều này cho thấy các cổ phiếu có khả năng điều chỉnh về rủi ro tốt còn về lợi nhuận thì đều là các con số hết sức ấn tượng vì đối với một doanh nghiệp tốt phần trăm tăng trưởng trong quá khứ cũng đã phản ánh được việc đó rồi có những công ty rất lớn quy mô rất lớn với mức tăng trưởng 20% một năm thì rất tuyệt vời. Mặc dù có phần nhỏ hơn so với mức độ rủi ro đối với đa số cổ phiếu nhưng với tỉ lệ sharpe và lợi nhuận như vậy thì các công ty này vẫn có sức hút để cho chúng ta đầu tư.
- Nhưng đối với Nokia thì hoàn toàn ngược lại đây là một công ty tăng trưởng rất chậm nhưng mà mức độ rủi ro thì lại rất cao và chỉ số sharpe ratio thì lại cực kì thấp thì điều này cho thấy công ty này có rất nhiều vấn đề và để tăng chỉ số sharpe cho portfolio thì chúng ta nên loại bỏ cổ phiếu này và thêm mã khác tốt hơn.
- Còn đối với Google và BRK-A thì chỉ số này cũng không đến nỗi nào có thể sử dụng được nếu như chưa tìm ra được cổ phiếu nào tốt hơn hoặc có thể thay thế khi có cổ phiếu tốt.
- Nhìn chung đối với những gì được thể hiện trên biểu đồ và các con số thì chúng ta hoàn toàn có thể tin tưởng vào danh mục này sẽ mang lại lợi nhuận lớn cho chúng ta trong tương lai.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

- Đạt được:
 - Nắm bắt được tác phong làm việc của việc đi làm ở doanh nghiệp
 - Bổ sung và nâng cao kiến thức về ngôn ngữ python và ứng dụng Power BI.
 - Nâng cao kỹ năng phân tích dữ liệu và trực quan hóa dữ liệu một cách dễ hiểu.
 - Nắm rõ quy trình làm việc của một DA.
 - Được trang bị các kiến thức về DA, vận dụng những gì được học tại công ty để áp dụng vào bài báo cáo.
- Hạn chế:
 - Vì kiến thức mới rất nhiều nên em chưa thể hiểu sâu trong thời gian thực tập ngắn nên có thể sẽ xảy ra một số sai sót
- Hướng phát triển:
 - Nghiên cứu chuyên sâu để hiểu biết thêm về nghề DA
 - Học nâng cao hiểu biết sâu hơn về ngôn ngữ python và ứng dụng PowerBI
 - Học các kiến thức liên quan đến machine learning model
 - Nâng cao khả năng phân tích và kỹ năng story telling

TÀI LIỆU THAM KHẢO

1. niithanoi. (n.d.). Retrieved from <https://niithanoi.edu.vn/data-analysis-la-gi-cac-loai-phuong-phap-phan-tich-du-lieu-co-ban-va-lam-the-nao-de-phan-tich-du-lieu.html>.
2. Nttuan8. (n.d.). Retrieved from <https://nttuan8.com/bai-14-long-short-term-memory-lstm/>.
3. Wikipedia. (n.d.). Retrieved from https://vi.wikipedia.org/wiki/B%E1%BB%99_nh%E1%BB%9B_d%C3%A0i-ng%E1%BA%AFn_h%E1%BA%A1n.