TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI: NGHIÊN CỨU VÀ ỨNG DỤNG KỸ THUẬT LOGISTIC REGRESSION KẾT HỢP MÔ HÌNH LSTM VÀO DỰ ĐOÁN THỊ TRƯỜNG CHỨNG KHOÁN

Giảng viên hướng dẫn: ThS. TRẦN PHONG NHÃ
Sinh viên thực hiện: ĐOÀN LÊ MỸ LINH
Lớp: CÔNG NGHỆ THÔNG TIN
Khóa: 59

TP. Hồ Chí Minh, năm 2022

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI: NGHIÊN CỨU VÀ ỨNG DỤNG KỸ THUẬT LOGISTIC REGRESSION KẾT HỢP MÔ HÌNH LSTM VÀO DỰ ĐOÁN THỊ TRƯỜNG CHỨNG KHOÁN

Giảng viên hướng dẫn: ThS. TRẦN PHONG NHÃ
Sinh viên thực hiện: ĐOÀN LÊ MỸ LINH
Lớp: CÔNG NGHỆ THÔNG TIN
Khóa: 59

TP. Hồ Chí Minh, năm 2022

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH

CỘNG HÒA XÃ HỘI CHỦ NGHIÃ VIỆT NAM
Độc lập – Tự do – Hạnh phúc

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP
BỘ MÔN**: CÔNG NGHỆ THÔNG TIN**
-------*-------**

Mã sinh viên: 5951071049 **Họ và tên: Đoàn Lê Mỹ Linh**

Khóa: 59 **Lớp: Công nghệ thông tin**

1. **Tên đề tài:** Nghiên cứu và ứng dụng kỹ thuật Logistic Regression kết hợp mô hình LSTM vào dự đoán thị trường chứng khoán.
2. **Mục tiêu:** Tìm hiểu về ngôn ngữ Python và nghiên cứu một số thuật toán máy học về phân tích và dự đoán kết quả như Logistic Regression, Long – short term memory. Từ đó ứng dụng vào phân tích và đưa ra các dự đoán về giá dựa trên dataset về chứng khoán.
3. **Nội dung thực hiện:**
   - Tìm hiểu ngôn ngữ Python và các thư viện cần sử dụng
   - Tìm hiểu sơ bộ về Machine Learning
   - Nghiên cứu thuật toán máy học: Logistic Regression và mô hình Long – short term memory.
   - Nghiên cứu bài toán phân tích và dự đoán về giá chứng khoán
   - Áp dụng kiến thức: ứng dụng ngôn ngữ Python và 2 thuật toán vào phân tích và đưa ra dự đoán về giá chứng khoán
4. **Công nghệ, công cụ và ngôn ngữ lập trình**
   - Công cụ sử dụng: Visual Studio Code
   - Ngôn ngữ: Python
5. **Các kết quả chính dự kiến**
   - Hiểu và sử dụng được ngôn ngữ lập trình Python
   - Hiểu được các thuật toán máy học cần sử dụng
   - Cài đặt được môi trường sử dụng ngôn ngữ
   - Áp dụng được kiến thức và cho ra kết quả
6. **Kế hoạch đang thực hiện**
   - **Tuần 1-2 và 3:** Tìm và chọn đề tài
   - **Tuần 4:** Đưa ra lựa chọn về đề tài

- **Tuần 5-6:** Tìm hiểu ngôn ngữ Python, thư viện cần sử dụng, đọc sách về ứng dụng AI vào phân tích thị trường chứng khoán.
- **Tuần 7 đến 11:** Nghiên cứu các thuật toán máy học và áp dụng kiến thức vào bài toán.
- **Tuần 12:** Viết báo cáo và làm slide
- **Tuần 13:** Nộp báo cáo và chờ duyệt

7. Giảng viên và cán bộ hướng dẫn

Họ tên: ThS. TRẦN PHONG NHÃ

Đơn vị công tác: Trường Đại học Giao thông Vận tải Phân hiệu tại TP. Hồ Chí Minh

Điện thoại: 0906 761 014 Email: tpnha@utc2.edu.vn

Ngày ..... tháng ..... năm 2022 **Đã giao nhiệm vụ TKTN**

Trưởng BM Công nghệ Thông tin **Giảng viên hướng dẫn**

Trần Phong Nhã

LỜI CẢM ƠN

Trước hết tôi xin gửi lời cảm ơn và bày tỏ lòng biết ơn chân thành đến thầy Trần Phong Nhã, người đã định hướng, cung cấp cho tôi những kiến thức, nguồn tài liệu và tận tình hướng dẫn chỉ bảo tôi trong suốt quá trình thực hiện đồ án tốt nghiệp của mình.

Tôi cũng xin chân thành cảm ơn các thầy, cô giáo của Bộ môn Công Nghệ Thông Tin – Phân hiệu trường Đại học Giao Thông Vận Tải tại TP. Hồ Chí Minh đã dạy bảo, truyền tải kiến thức, tạo điều kiện tốt nhất trong suốt quá trình tôi học tập tại trường.

Tôi cũng xin gửi lời cảm ơn sâu sắc đến gia đình, người thân luôn đồng hành, ủng hộ và động viên con trong học tập và cuộc sống.

Cuối cùng, tôi xin chân thành cảm ơn các bạn sinh viên lớp Công Nghệ Thông Tin K59 đã giúp đỡ, chia sẻ và khuyến khích tôi trong suốt quá trình học tập chung tại trường.

Hồ Chí Minh, ngày 10 tháng 5 năm 2022

Sinh viên

Đoàn Lê Mỹ Linh

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày …… tháng ……. năm …….

Giảng viên hướng dẫn

Trần Phong Nhã

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

STT	Từ viết tắt	Từ đầy đủ
1	AI	Artificial Intelligence
2	ANN	Autoenconder
3	AT	Agorithmic Trading
4	CNN	Convolutional Neuron Network
5	CRM	Customer Relationship Management
6	DBN	Deep Belief Net
7	DNA	Deoxyribonucleic Acid
8	DNN	Deep Neuron Network
9	GDP	Gross Domestic Product
10	GPU	Graphics Processing Unit
11	IoT	Internet of Things
12	LSTM	Long-short Term Memory
13	ML	Machine Learning
14	MLE	Maximum Likelihood Estimation
15	RNN	Recurrent Neural Network

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

MỞ ĐẦU

1. Lý do chọn đề tài

Trong thời đại công nghệ phát triển như ngày nay, các vấn đề tài chính cá nhân dần được nhiều bạn trẻ quan tâm từ sớm. Từ đó dần quan tâm đến đầu tư tài chính nhiều hơn nhằm tăng thêm thu nhập hoặc vốn đầu tư. Và đầu tư chứng khoán là một trong nhiều những hình thức phổ biến đáp ứng được mục đích đầu tư tài chính.

Trên thị trường chứng khoán hiện nay chia thành nhiều loại chứng khoán để các nhà đầu tư lựa chọn. Khi đầu tư, các nhà đầu tư đều hi vọng vốn đầu tư của mình sẽ sinh lời theo thời gian và biết được lúc nào thích hợp để thêm vốn đầu tư hoặc rút vốn. Để biết được điều đó đòi hỏi các nhà đầu tư cần đoán được chính xác về sự biến động trên thị trường chứng khoán. Từ đó quyết định vốn đầu tư của mình sẽ được phân bổ như thế nào, ra sao và vào khi nào thì hợp lý.

Dự báo sự biến động trên thị trường chứng khoán là một chủ đề quan trọng trong lĩnh vực tài chính. Việc dự báo hiệu quả sẽ giúp nhà đầu tư xây dựng được chiến lược đầu tư tối ưu cũng như phòng ngừa rủi ro. Dự báo một số chỉ số tài chính dựa trên một số yếu tố tác động sẽ dễ dàng nhưng kết quả có thể không chính xác, vì trên thực tế các yếu tố ảnh hưởng đến sự biến động của thị trường chứng khoán rất nhiều như tăng trưởng kinh tế, tình hình chính trị, các thông tin truyền thông,… Trong đầu tư chứng khoán, việc đưa ra quyết định đúng đắn trong khoảng thời gian kịp thời là một thách thức lớn đòi hỏi người đầu tư cần có một lượng thông tin đồ sộ để tính toán và dự đoán sự biến động của giá thị trường chứng khoán. Những thông tin này rất quan trọng đối với các nhà đầu tư vì sự biến động của thị trường chứng khoán có thể dẫn đến tổn thất đầu tư đáng kể. Qua đó ta thấy, việc phân tích thông tin lớn này rất hữu ích cho các nhà đầu tư và cũng hữu ích cho việc phân tích xu hướng biến động của các chỉ số thị trường chứng khoán. Rất khó để phân tích tất cả các yếu tố kể trên theo cách thủ công. Vì vậy, cần có một công cụ thông minh để giảm thiểu rủi ro với hy vọng có thể tối đa hóa lợi nhuận. Ngày nay, các mô hình Học máy (Machine Learning) đã trở thành một công cụ phân tích mạnh mẽ được sử dụng để trợ giúp và quản lý đầu tư hiệu quả.

Tuy nhiên, các yếu tố được đưa vào mô hình còn phụ thuộc vào mức độ hiểu biết của người xây dựng mô hình đó về lĩnh vực chứng khoán.

Cụ thể là trong đề tài thực hiện nghiên cứu ứng dụng thuật toán là Logistic Regression và mô hình học sâu LSTM để dự đoán giá của cổ phiếu dựa trên giá đóng cửa của cổ phiếu đó ở các ngày trước.

2. Mục tiêu và nhiệm vụ của đồ án

Tìm hiểu về ngôn ngữ Python và nghiên cứu thuật toán máy học về phân tích và dự đoán kết quả như Logistic Regression, Long – short term memory. Từ đó ứng dụng vào phân tích và đưa ra các dự đoán về giá dựa trên dataset về cổ phiếu được lấy từ trang finance.yahoo.com

3. Bố cục đồ án

Bố cục của đồ án được chia làm 4 phần và bao gồm những nội dung sau:

- Chương 1: Cơ sở lý thuyết: Tìm hiểu kỹ thuật hình học máy Logistic Regression và mô hình mạng LSTM. Các khái niệm liên quan đến đề tài nghiên cứu.
- Chương 2: Phân tích bài toán: Gồm phân tích dữ liệu, đưa ra mô hình phù hợp và phương pháp đánh giá mô hình.
- Chương 3: Thực nghiệm và đánh giá kết quả: Xây dựng cài đặt mô hình, huấn luyện mô hình, thực hiện thử nghiệm dự đoán.
- Kết luận: Tổng kết lại quá trình nghiên cứu và thực nghiệm, những kết quả đạt được.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1 Chứng khoán và thị trường chứng khoán [1]

1.1.1 Chứng khoán

a. Khái niệm

- Chứng khoán là bằng chứng xác nhận quyền và lợi ích hợp pháp của người sở hữu đối với tài sản hoặc phần vốn của tổ chức phát hành. Chứng khoán được thể hiện dưới hình thức chứng chỉ, bút toán ghi sổ hoặc dữ liệu điện tử.

- Chứng khoán bao gồm: Cổ phiếu, trái phiếu, chứng chỉ quỹ đầu tư, chứng khoán phái sinh.

b. Phân loại chứng khoán

Có nhiều cách phân loại chứng khoán dựa trên những tiêu chí khác nhau. Sau đây là một số cách phân loại thường được sử dụng:

- Căn cứ vào chủ thể phát hành: chứng khoán chính phủ và chính quyền địa phương, chứng khoán doanh nghiệp, chứng khoán của ngân hàng thương mại và các tổ chức tài chính tín dụng.

- Căn cứ vào tính chất huy động vốn: chứng khoán vốn (cổ phiếu), chứng khoán nợ (trái phiếu), chứng khoán phái sinh.

- Căn cứ vào lợi tức của chứng khoán: chứng khoán có thu nhập cố định, chứng khoán có thu nhập biến đổi.

- Căn cứ theo hình thức chứng khoán: chứng khoán ghi danh, chứng khoán không ghi danh.

- Căn cứ theo thị trường nơi chứng khoán được giao dịch: chứng khoán niêm yết, chứng khoán không được niêm yết.

1.1.2 Thị trường chứng khoán

a. Khái niệm

- Thị trường chứng khoán là một bộ phận quan trọng của thị trường vốn, hoạt động của nó nhằm huy động những nguồn vốn tiết kiệm nhỏ trong xã hội tập trung thành nguồn vốn lớn tài trợ dài hạn cho các doanh nghiệp, các tổ chức kinh tế và Nhà nước để phát triển sản xuất, tăng trưởng kinh tế hay cho các dự án đầu tư.

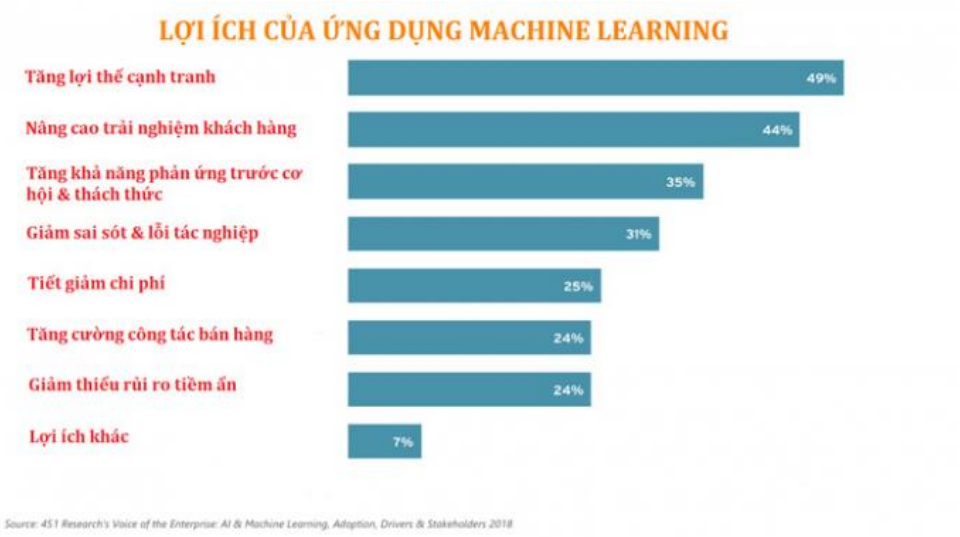
- Thị trường chứng khoán là nơi diễn ra các hoạt động giao dịch mua bán các loại chứng khoán. Việc mua bán này trước tiên được tiến hành ở thị trường sơ cấp khi người mua mua được chứng khoán lần đầu từ những người phát

hành và sau đó ở thị trường thứ cấp khi có sự mua đi bán lại các chứng khoán đã được phát hành ở thị trường sơ cấp. Do vậy, thị trường chứng khoán là nơi các chứng khoán được phát hành và trao đổi.

b. Cơ cấu của thị trường chứng khoán

- Căn cứ vào phương thức giao dịch: có thị trường giao ngay và thị trường tương lai.
- Căn cứ vào tính chất các chứng khoán được giao dịch: được chia thành thị trường cổ phiếu, thị trường trái phiếu và thị trường các chứng khoán phái sinh.
- Căn cứ vào sự luân chuyển các nguồn vốn: có thị trường chứng khoán được chia thành thị trường sơ cấp và thị trường thứ cấp.

1.2 Mối liên hệ giữa Học máy và Thị trường chứng khoán [12]



Hình 1.1 Lợi ích của ứng dụng máy học

- Những biến động trong thị trường chứng khoán luôn được coi là phức tạp và bắt nguồn từ nhiều nguyên nhân khác nhau. Tuy nhiên, điều này không có nghĩa việc dự đoán xu hướng của thị trường này là việc không thể. Trên thực tế, học máy đã làm khá tốt vai trò của một "nhà dự báo" bằng việc phân tích và tận dụng tối đa lượng dữ liệu lịch sử kết hợp cùng với kiến thức của người xây dựng mô hình dự đoán.

- Học máy là mô hình AI được sử dụng rộng rãi nhất trong lĩnh vực tài chính, dựa trên một công trình nghiên cứu hồi năm 1943 của McCullogh và Pitts. Về nguyên tắc, một hệ thống học máy bao gồm: đầu bài, nguồn dữ liệu, mô hình, thuật toán tối ưu, hệ thống đánh giá và kiểm thử.

- Một số điểm nổi bật khi ứng dụng Machine Learning trong ngân hàng vào dự báo thị trường chứng khoán có thể kể đến khả năng phán đoán không giới hạn, trái với những hạn chế trong tư duy con người. Học máy cũng ghi nhận

những sự thay đổi nhỏ nhất về giá, so sánh dữ liệu ở hiện tại với những dữ liệu từ rất lâu trước đây, trợ giúp đắc lực trong việc đưa ra các quyết định đầu tư hiệu quả.

- Trong lĩnh vực tài chính - ngân hàng, Machine Learning, khi được kết hợp với các mô hình phân tích định lượng, phát huy hiệu quả đặc biệt trong việc tìm kiếm các bộ mẫu dữ liệu, đưa ra những dự đoán, hỗ trợ hiệu quả ra quyết định giúp đảm bảo hoạt động kinh doanh liên tục và kiểm soát rủi ro. Trên thế giới, cuộc chạy đua trong ngành Ngân hàng cũng như các thị trường chứng khoán diễn ra đặc biệt sôi động. Từ các công ty công nghệ mới thành lập như Feedzai (trong mảng thanh toán), Shift Technology (trong mảng bảo hiểm), tới các tập đoàn công nghệ khổng lồ như IBM và nhóm dẫn đầu về công nghệ hiện tại như Google, Alibaba và các Fintech, đang dựa vào ưu thế công nghệ để cạnh tranh, lấn sân sang lĩnh vực ngân hàng, tài chính, cổ phiếu thị trường.

- Theo nghiên cứu, vào năm 2017, có tới 78% các ngân hàng được khảo sát bị ảnh hưởng bởi các hành vi gian lận. Chi phí để giải quyết và phục hồi cho vấn đề này cũng đặt gánh nặng lớn lên vai các nhà băng khi họ phải tiêu tốn đến 2,92 USD cho mỗi 1 USD thiệt hại do gian lận.

- Phát hiện gian lận là ứng dụng quan trọng của Machine Learning trong ngân hàng bởi khả năng phân tích nhanh chóng và chính xác hàng triệu điểm dữ liệu từ các giao dịch diễn ra đồng thời. Machine Learning sẽ kiểm tra các thông tin liên quan đến thời gian, hành vi của khách hàng và các thông số khác để xác định đâu là các hành vi gian lận. Sau đó hệ thống sẽ tự động gửi cảnh báo về trung tâm bảo mật, hoặc từ chối giao dịch trong trường hợp gian lận thẻ tín dụng. Từ đó, các nhà băng có thể kịp thời ngăn chặn, tránh những rủi ro không đáng có. Có thể kể đến như: Monzo - một ngân hàng khởi nghiệp tại Anh, đã xây dựng một mô hình phân tích, dự báo đủ nhanh, để kịp thời phát hiện và ngăn chặn những kẻ lừa đảo giả mạo trong quá trình hoàn tất giao dịch, giúp giảm tỷ lệ lừa đảo trên thẻ trả trước từ 0,85% vào tháng 6/2016 xuống dưới 0,1% vào tháng1/2017.

- Phương pháp đánh giá rủi ro truyền thống bộc lộ nhiều điểm yếu do bị hạn chế bởi một vài thông tin thiết yếu như điểm tín dụng. Đó cũng là lý do của Machine Learning trong ngân hàng được sử dụng để giải quyết vấn đề này. Có thể được áp dụng để đánh giá mức độ rủi ro của một khoản đầu tư.

- Giao dịch thuật toán (AT) được định nghĩa là một quy trình thực hiện lệnh trong đó các chỉ thị giao dịch tự động hoặc được lập trình trước sẽ được sử

dụng cho các biến số như giá cả, thời gian và lượng. Các giao dịch này ứng dụng thuật toán của Machine Learning và sự giám sát của con người, từ đó đưa ra những quyết định mua/bán chứng khoán. AT chiếm tới 1/5 tổng số giao dịch trên nền tảng tiền tệ đa đại lý EBS. Đây được coi là một khái niệm mới tại Việt Nam.

- Các tổ chức đầu tư lớn, hoặc các công ty môi giới là những doanh nghiệp sử dụng phần lớn những giao dịch này để tiết kiệm chi phí, đặc biệt với những lệnh có quy mô lớn. Tốc độ thực hiện lệnh của AT cũng rất ấn tượng, thường được người giao dịch thuật toán tận dụng để thực hiện lên tới 10.000 giao dịch mỗi giây. Đặc điểm này cũng hỗ trợ các nhà đầu tư thu lợi nhanh chóng chỉ những những biến động nhỏ trong giá cả.

Hình 1.2 Ứng dụng máy học trong giao dịch thuật toán

1.3 Ngôn ngữ Python

1.3.1 Giới thiệu

- Python là một ngôn ngữ lập trình bậc cao hướng đối tượng được Guido van Rossum cùng các cộng sự tạo ra năm 1991, dành cho mục đích lập trình đa năng. Python được thiết kế với ưu điểm mạnh là câu lệnh ngắn gọn, dễ nhớ, dễ hiểu. Cấu trúc chương trình của Python rõ ràng, dễ đọc và viết hơn rất nhiều so với những ngôn ngữ lập trình khác. Do đó Python được coi là một trong những ngôn ngữ thuận tiện nhất cho người mới học lập trình. Đây là ngôn ngữ lập trình thông dịch, có thể chạy trên nhiều hệ điều hành khác nhau. Python là ngôn ngữ mã nguồn mở và có cộng đồng người dùng lớn. [4]
- Python đã trở thành ngôn ngữ chung cho nhiều ứng dụng khoa học dữ liệu. Nó kết hợp sức mạnh của các ngôn ngữ lập trình có mục đích chung với sự dễ sử dụng của các ngôn ngữ kịch bản miền cụ thể như MATLAB hoặc R. Python có các thư viện để tải dữ liệu, trực quan hóa, thống kê, xử lý ngôn ngữ tự nhiên,

xử lý hình ảnh và hơn thế nữa. Hộp công cụ rộng lớn này cung cấp cho các nhà khoa học dữ liệu một loạt các chức năng cho mục đích chung và mục đích đặc biệt. Một trong những lợi thế chính của việc sử dụng Python là khả năng tương tác trực tiếp với mã, sử dụng thiết bị đầu cuối hoặc các công cụ khác. [3]

- Một số đặc điểm của ngôn ngữ Python: [4]
  - Đơn giản, dễ học
  - Miễn phí, mã nguồn mở
  - Khả chuyển
  - Khả năng mở rộng và khả năng nhúng

1.3.2 Một số thư viện của ngôn ngữ Python

- Numpy: Tên "Numpy" là viết tắt của "Numerical Python". Nó là thư viện thường được sử dụng. Một thư viện học máy phổ biến hỗ trợ các ma trận và dữ liệu đa chiều. Bao gồm các hàm toán học được xây dựng sẵn để dễ dàng tính toán. Ngay cả các thư viện như TensorFlow cũng sử dụng Numpy nội bộ để thực hiện một số hoạt động trên Tensors. Giao diện mảng là một trong những tính năng chính của thư viện này. [5]
- Pandas: Pandas là một thư viện quan trọng cho các nhà khoa học dữ liệu. Đây là một thư viện máy học mã nguồn mở cung cấp các cấu trúc dữ liệu cấp cao linh hoạt và nhiều công cụ phân tích. Nó giúp giảm bớt phân tích dữ liệu, thao tác dữ liệu và làm sạch dữ liệu. Pandas hỗ trợ các hoạt động như sắp xếp, lập chỉ mục lại, lặp lại, kết hợp, chuyển đổi dữ liệu, hình ảnh hóa, tổng hợp, … [5]
- Matplotlib: Thư viện này chịu trách nhiệm vẽ dữ liệu số. Và đó là lý do tại sao nó được sử dụng trong phân tích dữ liệu. Nó cũng là một thư viện mã nguồn mở và vẽ các số liệu được xác định cao như biểu đồ hình tròn, biểu đồ, biểu đồ phân tán, biểu đồ, … [5]
- Sklearn (hay scikit-learn): Nó là một thư viện Python nổi tiếng để làm việc với dữ liệu phức tạp. Scikit-learning là một thư viện mã nguồn mở hỗ trợ học máy. Nó hỗ trợ các thuật toán được giám sát và không được giám sát khác nhau như hồi quy tuyến tính, phân loại, phân cụm, ... Thư viện này hoạt động cùng với Numpy và SciPy. [5]
- Keras: Keras cung cấp thư viện tiện ích numpy, cung cấp các hàm để thực hiện các hành động trên mảng numpy. Sử dụng phương thức to_categorical (), một mảng numpy (hoặc) một vectơ có các số nguyên đại diện cho các danh mục khác nhau, có thể được chuyển đổi thành một mảng numpy (hoặc) một

ma trận có các giá trị nhị phân và có các cột bằng số danh mục trong dữ liệu. [6]

- Seaborn: Seaborn là một thư viện trực quan tuyệt vời để vẽ đồ họa thống kê bằng Python. Nó cung cấp các kiểu và bảng màu mặc định đẹp mắt để làm cho các ô thống kê trở nên hấp dẫn hơn. Nó được xây dựng trên đầu thư viện matplotlib và cũng được tích hợp chặt chẽ với cấu trúc dữ liệu từ gấu trúc. Seaborn nhằm mục đích làm cho trực quan hóa trở thành phần trung tâm của việc khám phá và hiểu dữ liệu. Nó cung cấp các API hướng tập dữ liệu, để chúng ta có thể chuyển đổi giữa các biểu diễn trực quan khác nhau cho các biến giống nhau để hiểu rõ hơn về tập dữ liệu. [7]

1.4 Tổng quan về Machine Learning

1.4.1 Giới thiệu

- Máy học là cách trích xuất kiến thức từ dữ liệu. Nó là một lĩnh vực nghiên cứu ở giao điểm của thống kê, trí tuệ nhân tạo và khoa học máy tính và còn được gọi là phân tích dự đoán hoặc học thống kê. Việc áp dụng các phương pháp học máy trong những năm gần đây đã trở nên phổ biến trong cuộc sống hàng ngày. Từ tự động- gợi ý matic về những bộ phim nên xem, món ăn cần đặt hoặc sản phẩm cần mua, đài phát thanh trực tuyến được cá nhân hóa và nhận ra bạn bè của bạn trong ảnh của bạn, nhiều trang web và thiết bị hiện đại có các thuật toán máy học ở cốt lõi của chúng. Khi bạn nhìn vào một trang web phức tạp như Facebook, Amazon hoặc Netflix, nó rất có thể mọi phần của trang web đều chứa nhiều mô hình học máy. [3]
- Học máy (Machine learning) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Học máy hiện nay được áp dụng rộng rãi bao gồm máy truy tìm dữ liệu, chẩn đoán y khoa, phát hiện thẻ tín dụng giả, phân tích thị trường chứng khoán, phân loại các chuỗi DNA, nhận dạng tiếng nói và chữ viết, dịch tự động, chơi trò chơi và cử động rô-bốt (*robot locomotion*). [13]
- Học máy rất gần với suy diễn thống kê (statistical inference) tuy có khác nhau về thuật ngữ. Một nhánh của học máy là học sâu (Deep Learning) phát triển rất mạnh mẽ gần đây và có những kết quả vượt trội so với các phương pháp học máy khác. Học máy có liên quan lớn đến thống kê, vì cả hai lĩnh vực đều nghiên cứu việc phân tích dữ liệu, nhưng khác với thống kê, học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán. Nhiều bài toán suy luận được xếp vào loại bài toán NP-khó, vì thế một phần của học

máy là nghiên cứu sự phát triển các giải thuật suy luận xấp xỉ mà có thể xử lý được. [13]

1.4.2 Phân loại học máy [13]

- Các thuật toán học máy được phân loại theo kết quả mong muốn của thuật toán. Các loại thuật toán thường dùng bao gồm:

- Học có giám sát—trong đó, thuật toán tạo ra một hàm ánh xạ dữ liệu vào tới kết quả mong muốn. Một phát biểu chuẩn về một việc học có giám sát là bài toán phân loại: chương trình cần học (cách xấp xỉ biểu hiện của) một hàm ánh xạ một vector $[X_1, X_2, \dots, X_N]$ tới một vài lớp bằng cách xem xét một số mẫu dữ liệu - kết quả của hàm đó.
- Học không giám sát—mô hình hóa một tập dữ liệu, không có sẵn các ví dụ đã được gắn nhãn.
- Học nửa giám sát—kết hợp các ví dụ có gắn nhãn và không gắn nhãn để sinh một hàm hoặc một bộ phân loại thích hợp.
- Học tăng cường—trong đó, thuật toán học một chính sách hành động tùy theo các quan sát về thế giới. Mỗi hành động đều có tác động tới môi trường, và môi trường cung cấp thông tin phản hồi để hướng dẫn cho thuật toán của quá trình học.
- Chuyển đổi—tương tự học có giám sát nhưng không xây dựng hàm một cách rõ ràng. Thay vì thế, cố gắng đoán kết quả mới dựa vào các dữ liệu huấn luyện, kết quả huấn luyện, và dữ liệu thử nghiệm có sẵn trong quá trình huấn luyện.
- Học cách học—trong đó thuật toán học thiên kiến quy nạp của chính mình, dựa theo các kinh nghiệm đã gặp.

1.5 Kỹ thuật học máy Logistic Regression

Trong Machine learning thì Logistic Regression là thuộc kỹ thuật học có giám sát.

1.5.1 Giới thiệu Logistic Regression

- Hồi quy logistic là một mô hình thống kê ở dạng cơ bản của nó sử dụng một hàm logistic để mô hình hóa một biến phụ thuộc nhị phân, mặc dù tồn tại nhiều phần mở rộng phức tạp hơn. Trong phân tích hồi quy, hồi quy logistic (hay hồi quy logit) là ước lượng các tham số của mô hình logistic (một dạng của hồi quy nhị phân). Về mặt toán học, mô hình logistic nhị phân có một biến

phụ thuộc với hai giá trị có thể có, chẳng hạn như đạt hoặc không đạt được đại diện bởi một biến chỉ báo, trong đó hai giá trị được gắn nhãn "0" và "1". [2]

- Một số loại mô hình dự đoán sử dụng phân tích logistic:

  - Mô hình tuyến tính tổng quát
  - Sự lựa chọn rời rạc
  - Logit đa thức
  - Đăng nhập hỗn hợp
  - Probit
  - Probit đa thức
  - Đăng nhập có thứ tự

1.5.2 Hàm Logistic Regression [9]

- Loại mô hình thống kê này (còn được gọi là mô hình logit) thường được sử dụng để phân loại và phân tích dự đoán. Hồi quy logistic ước tính xác suất xảy ra sự kiện, chẳng hạn như đã bỏ phiếu hoặc không bỏ phiếu, dựa trên một tập dữ liệu nhất định gồm các biến độc lập. Vì kết quả là một xác suất, biến phụ thuộc bị giới hạn trong khoảng từ 0 đến 1. Trong hồi quy logistic, một phép biến đổi logit được áp dụng trên tỷ lệ cược - nghĩa là xác suất thành công chia cho xác suất thất bại. Đây cũng thường được gọi là tỷ lệ cược log, hoặc logarit tự nhiên của tỷ lệ cược và hàm logistic này được biểu diễn bằng các công thức sau:

$$Logit(pi) = \frac{1}{1+e^{-pi}} \qquad (1.1)$$

$$\ln\left(\frac{pi}{1-pi}\right) = Beta_0 + Beta_1 * X_1 + \cdots + B_k * X_k \qquad (1.2)$$

- Trong phương trình hồi quy logistic này, logit (pi) là biến phụ thuộc hoặc biến phản hồi và x là biến độc lập. Tham số beta hoặc hệ số trong mô hình này thường được ước tính thông qua ước tính khả năng xảy ra tối đa (MLE). Phương pháp này kiểm tra các giá trị beta khác nhau thông qua nhiều lần lặp lại để tối ưu hóa cho phù hợp nhất với tỷ lệ cược nhật ký. Tất cả các lần lặp này tạo ra hàm khả năng log và hồi quy logistic tìm cách tối đa hóa hàm này để tìm ra ước tính tham số tốt nhất. Khi hệ số tối ưu (hoặc các hệ số nếu có nhiều hơn một biến độc lập) được tìm thấy, các xác suất có điều kiện cho mỗi quan sát có thể được tính toán, ghi lại và tổng hợp với nhau để mang lại xác suất dự đoán. Đối với phân loại nhị phân, xác suất nhỏ hơn 0,5 sẽ dự đoán 0 trong khi xác suất lớn hơn 0,5 sẽ dự đoán 1.

1.5.3 Hồi quy Logistic nhị phân [8]

- Hồi quy Logistic nhị phân thường được đề cập liên quan đến các nhiệm vụ phân loại. Mô hình này đơn giản và là một trong những cách bắt đầu dễ dàng để tìm hiểu về cách tạo xác suất, phân loại mẫu và hiểu gradient descent.
- Để thực hiện một dự đoán, sử dụng ký hiệu giống mạng thần kinh: có trọng số (w), đầu vào (x) và bias (b). Ta có thể lặp lại một lỗi và nhiều lỗi lại với nhau và thêm bias vào cuối như thể hiện trong ví dụ sau.

$$z = \left(\sum_{i=1}^{n} w_i x_i\right) + b \quad (1.3)$$

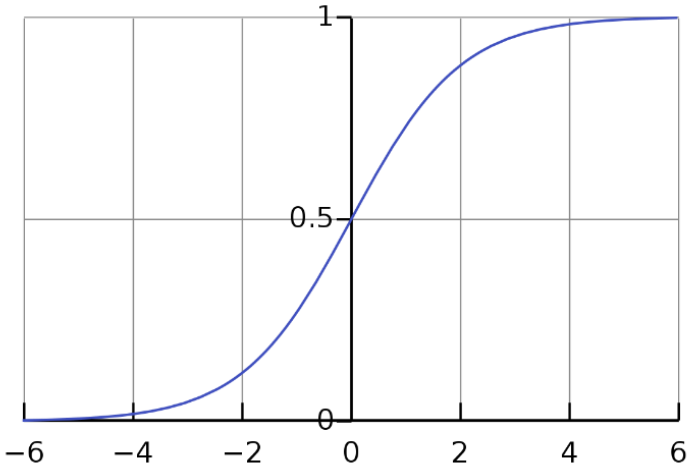
- Tuy nhiên, người ta thường sử dụng ký hiệu vectơ. Điều này có nghĩa là w trở thành một danh sách các giá trị. Ký hiệu vectơ cho phép sử dụng thời gian tính toán nhanh hơn, điều này có thể rất có lợi nếu muốn tạo mẫu nhanh với tập dữ liệu lớn hơn. Ví dụ: vectơ **w** và bạn có thể viết lại phương trình của mình bên trên phương trình sau.

$$z = \boldsymbol{w}.\boldsymbol{x} + b \quad (1.4)$$

- Hàm sigmoid:

$$\hat{y} = \sigma(z) = \begin{cases} \frac{1}{(1+\exp(z))'}, & if\ z \geq 0 \\ \frac{\exp(z)}{(1+\exp(z))'}, & otherwise \end{cases} \quad (1.5)$$

- Mô hình hóa hàm sigmoid:



Hình 1.3 Mô hình hóa hàm Sigmoid

- Hàm Loss:

$$L_{CE}(\hat{y}, y) = -\frac{1}{m}\sum_{i=1}^{m} ylog(\hat{y}) + (1-y)log(1-\hat{y}) \qquad (1.6)$$

- Hàm Loss (còn được gọi là hàm chi phí) là một hàm được sử dụng để đo lường mức độ khác biệt giữa dự đoán với các nhãn.
- Nhìn vào dấu cộng trong phương trình, nếu y = 0, thì vế trái bằng = và nếu y = 1, thì vế phải bằng 0. Một cách hiệu quả, đây là cách ta đo lường mức độ dự đoán của $\hat{y}$ khác với nhãn y, chỉ có thể là 0 hoặc 1 trong thuật toán phân loại nhị phân.
- Bây giờ, để tính toán các gradient nhằm tối ưu hóa trọng số bằng cách sử dụng gradient descent, ta phải tính đạo hàm của hàm Loss. Tóm tắt đạo hàm:

$$\frac{\partial L_{CE}(\hat{y},y)}{\partial b} = \frac{1}{m}(\hat{y} - y) \qquad (1.7)$$

- Khi có các giá trị cuối cùng từ phép tính đạo hàm của mình, ta có thể sử dụng nó trong phương trình gradient descent và cập nhật trọng số và bias.

1.5.4 Ưu – nhược điểm [11]

➢ *Ưu điểm*

- **Hồi quy logistic dễ thực hiện hơn nhiều so với các phương pháp khác, đặc biệt là trong ML:** Mô hình ML có thể được mô tả như một mô tả toán học của một quá trình trong thế giới thực. Quá trình thiết lập mô hình học máy yêu cầu đào tạo và thử nghiệm mô hình. Huấn luyện là quá trình tìm kiếm các mẫu trong dữ liệu đầu vào, để mô hình có thể ánh xạ một đầu vào cụ thể (ví dụ, một hình ảnh) tới một loại đầu ra nào đó, chẳng hạn như một nhãn. Hồi quy logistic dễ đào tạo và triển khai hơn so với các phương pháp khác.
- **Hồi quy logistic hoạt động tốt đối với các trường hợp tập dữ liệu có thể phân tách tuyến tính:** Tập dữ liệu được cho là có thể phân tách tuyến tính nếu có thể vẽ một đường thẳng để tách hai lớp dữ liệu khỏi nhau. Hồi quy logistic được sử dụng khi biến Y chỉ có thể nhận hai giá trị và nếu dữ liệu có thể phân tách tuyến tính, thì việc phân loại nó thành hai lớp riêng biệt sẽ hiệu quả hơn.
- **Hồi quy logistic cung cấp những hiểu biết hữu ích:** Hồi quy logistic không chỉ cho phép đo lường mức độ liên quan của một biến độc lập (tức là (kích thước hệ số), mà còn cho chúng ta biết về hướng của mối quan hệ (tích cực hoặc tiêu cực). Hai biến được cho là có một liên kết tích cực khi sự gia tăng giá trị của một biến số cũng làm tăng giá trị của biến số khác. Ví dụ: càng

dành nhiều giờ tập luyện một môn thể thao thì càng trở nên giỏi hơn trong môn đó.

➢ *Nhược điểm*

- **Hồi quy logistic giả định tính tuyến tính giữa biến dự đoán (phụ thuộc) và biến dự báo (độc lập).** Tại sao đây là một hạn chế? Trong thế giới thực, rất khó có khả năng các quan sát được phân tách tuyến tính. Vì vậy, trong khi dữ liệu có thể phân tách tuyến tính là giả định cho hồi quy logistic, trên thực tế, nó không phải lúc nào cũng thực sự khả thi.

- **Hồi quy logistic có thể không chính xác nếu kích thước mẫu quá nhỏ.** Nếu kích thước mẫu ở mức nhỏ, thì mô hình được tạo ra bằng hồi quy logistic dựa trên số lượng quan sát thực tế nhỏ hơn. Điều này có thể dẫn đến trang bị quá nhiều. Trong thống kê, overfitting là một lỗi mô hình hóa xảy ra khi mô hình quá khớp với một bộ dữ liệu hạn chế vì thiếu dữ liệu đào tạo.

1.6 Học sâu (Deep Learning)

- Học sâu là một nhánh của lĩnh vực học máy liên quan đến các thuật toán bắt chước cách thức hoạt động của bộ não cả về cấu trúc và chức năng. Học sâu chủ yếu được phát triển dựa trên nguyên lý kỹ thuật mạng nơ ron nhân tạo. Hiện nay chưa có sự thống nhất trong định nghĩa về học sâu. [10]

- Theo tác giả Yann LeCun, một trong những cha đẻ của học sâu, thì lĩnh vực này có thể hiểu là lớp các thuật toán học máy cho phép mô hình tính toán tổng hợp nhiều lớp xử lý để khám phá nhiều mức độ trừu tượng khác nhau của dữ liệu (đặc trưng mức cao của dữ liệu) từ tập dữ liệu thô đầu vào.

- Học sâu có thể hiểu là một hệ thống gồm nhiều thành phần mà tất cả chúng đều có thể huấn luyện được. Nó được gọi là "sâu" vì quá trình xử lý có rất nhiều giai đoạn để tri nhận về một đối tượng và tất cả các giai đoạn này đều tham gia vào quá trình học.

- Các mô hình học sâu của Deep Learning:

  - **Mạng neuron tái tạo (RNN)**: mở rộng khả năng của mạng neuron truyền thống và được thiết kế để lập mô hình dữ liệu dạng tuần tự. RNN được sử dụng rộng rãi trong các lĩnh vực khác nhau như xử lý giọng nói, nhận dạng hoạt động của con người, dự đoán chữ viết tay và hiểu ngữ nghĩa.
  - **Autoencoder (ANN)** là mạng nơ ron nhân tạo có khả năng học hiệu quả các biểu diễn của dữ liệu đầu vào mà không cần nhãn. Các biểu diễn này thường có chiều nhỏ hơn nhiều so với đầu vào, do đó autoencoder có thể dùng trong các bài toán giảm chiều dữ liệu. Hơn nữa, autoencoder còn

có thể hoạt động như các bộ phát hiện đặc trưng, để lấy ra các đặc trưng trước khi huấn luyện nhằm thực hiện các bài toán khác.

- **Mạng nơ ron sâu (DNN)** là một dạng cụ thể của lĩnh vực học sâu. Mạng nơ ron sâu là một mạng nơ ron nhân tạo nhưng có kiến trúc phức tạp và "sâu" hơn nhiều so với kiến trúc của mạng nơ ron truyền thống. Nghĩa là nó có số nút trong mỗi lớp và số lớp ẩn lớn hơn rất nhiều và cách thức hoạt động của nó phức tạp hơn so với kiến trúc mạng nơ ron truyền thống.
- **Mạng nơ-ron tích chập (CNN)** là một dạng cụ thể của mạng nơ ron sâu. Mạng nơ ron tích chập có một lớp vào, một lớp ra và nhiều lớp ẩn khác nhau. Các lớp ẩn gồm các loại như: lớp tích chập, lớp giảm kích thước, lớp sửa dữ liệu, lớp chuẩn hóa, lớp kết nối đầy đủ... Trong đó, lớp tích chập được sử dụng nhằm tạo liên kết giữa các lớp liền kề trong phạm vi nhỏ, giới hạn trong một vùng cục bộ.
- **Mạng học sâu niềm tin (DBN)** là một mô hình mạng nơ-ron nhân tạo nhiều lớp. Quá trình huấn luyện mạng DBN gồm hai pha: tiền huấn luyện và hiệu chỉnh trọng số. Trong pha tiền huấn luyện, máy học Boltzmann được sử dụng để khởi tạo trọng số tốt nhất cho mô hình với dữ liệu không cần được gán nhãn. Trong pha hiệu chỉnh trọng số, DBN tiếp tục được huấn luyện bằng phương pháp lan truyền ngược cổ điển với dữ liệu được gán nhãn.

- Là một xu hướng nóng trong công nghệ thông tin, học sâu không những là chủ đề được cộng động nghiên cứu khoa học máy tính quan tâm hàng đầu mà đã vượt ra khuôn khổ của các phòng, dự án nghiên cứu, để trở thành công nghệ được ứng dụng trong thực tiễn.

- Một số ứng dụng của Deep Learning có thể kể đến như: xử lý ngôn ngữ tự nhiên, mô phỏng và nhận diện hình ảnh, trợ lý ảo, ứng dụng xe tự động, trong quản lý quan hệ khách hàng (CRM), dịch thuật, chống gian lận điện tử, thương mại điện tử và cá nhân hóa người dùng,…

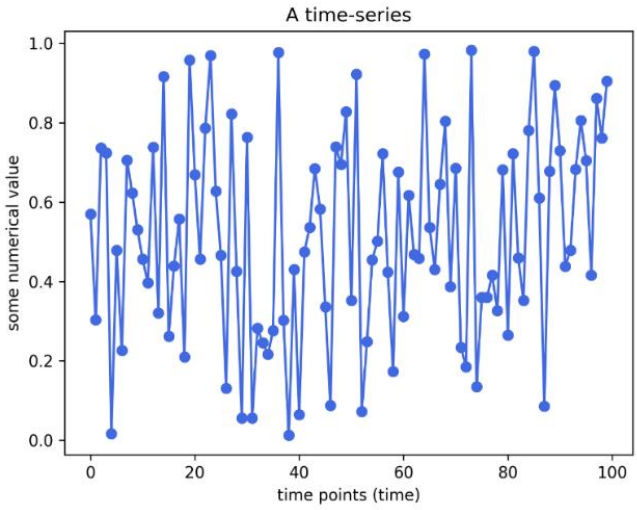
Hình 1.4 Ứng dụng của Deep learning

- Những năm gần đây, kỹ thuật học sâu đang trở thành một trong những lĩnh vực được quan tâm nghiên cứu và ứng dụng đặc biệt trong lĩnh vực khoa học máy tính. Kỹ thuật học sâu đã đạt được những kết quả khả quan với độ chính xác vượt trội so với cách tiếp cận truyền thống, đồng thời thúc đẩy tiến bộ trong đa lĩnh vực như nhận dạng đối tượng, dịch tự động, nhận dạng giọng nói, các trò chơi thông minh và những bài toán khó trong trí tuệ nhân tạo.

- Các chuyên gia trí tuệ nhân tạo và học sâu đều có nhận định rằng để phát triển tốt lĩnh vực này trong cả nghiên cứu lẫn công nghiệp, vấn đề quan trọng là hình thành các cơ sở dữ liệu đủ lớn và đủ tốt dùng trong huấn luyện các mô hình học sâu. Những cơ sở dữ liệu lớn như vậy về ảnh y tế, tiếng nói, tín hiệu điện tim, điện não, ảnh giao thông… đang dần được xây dựng bởi các tập đoàn công nghệ, cộng đồng nghiên cứu trong các trường, viện nghiên cứu dưới sự bảo trợ của Chính phủ.

1.7 Dữ liệu chuỗi thời gian (Time series data)

- Time-series Data: là một chuỗi các điểm dữ liệu, thường bao gồm các phép đo liên tiếp được thực hiện từ cùng một nguồn trong một khoảng thời gian. Phân tích chuỗi thời gian có mục đích nhận đang và tập hợp lại các yếu tố, những biến đổi theo thời gian mà nó có ảnh hưởng đến giá trị của biến quan sát. [14]



Hình 1.5 Ví dụ về Time series

- Trong Time-series Data, có hai loại chính. [14]

  - Chuỗi thời gian thông thường (regular time series), loại thông thường được gọi là số liệu.
  - Chuỗi thời gian bất thường (events) là những sự kiện.

- Ứng dụng: Time-series data được ứng dụng rất rộng rãi trong các lĩnh vực: [14]

- IoT
- DevOps
- Phân tích thời gian thực
- Dự báo kinh tế
- Tính toán doanh số bán hàng
- Phân tích lãi
- Phân tích thị trường
- Kiểm soát quy trình và chất lượng
- Phân tích điều tra

- Ưu điểm của chuỗi thời gian là nó có thể lưu trữ được trạng thái của một trường dữ liệu theo thời gian. Trong khi đó thế giới luôn vận động, các sự vật, hiện tượng hiếm khi dừng lại ở trạng thái tĩnh mà thường thay đổi. Do đó dữ liệu chuỗi thời gian có tính ứng dụng rất cao và được áp dụng trong rất nhiều lĩnh vực khác nhau như: *thống kê, kinh tế lượng, toán tài chính, dự báo thời tiết, dự đoán động đất, điện não đồ, kỹ thuật điều khiển, thiên văn, kỹ thuật truyền thông, xử lý tín hiệu*. Dữ liệu chuỗi thời gian cho phép các quốc qia trên thế giới hàng năm đưa ra dự báo tăng trưởng GDP của mình và các doanh nghiệp dự báo doanh số và triển vọng thị trường. Chính vì thế dữ liệu chuỗi thời gian đóng một vai trò cực kỳ quan trọng đối với sự phát triển của nhân loại. [12]
- Dữ liệu chuỗi thời gian có những tính chất đặc trưng riêng như: [12]

- Tính xu hướng: Tính xu hướng là yếu tố thể hiện xu hướng thay đổi của dữ liệu theo thời gian. Đây là đặc trưng thường thấy của rất nhiều dữ liệu chuỗi thời gian. Đặc biệt là các chuỗi trong kinh tế lượng như: giá cả thị trường chị ảnh hưởng của lạm phát, dân số thế giới tăng qua các năm, nhiệt độ trung bình trái đất tăng theo thời gian do hiệu ứng nhà kính, …. Tính xu hướng cũng ảnh hưởng không nhỏ tới việc đưa ra nhận định về mối quan hệ tương quan giữa các chuỗi số. Tức là về bản chất các chuỗi không tương quan nhưng do chúng cùng có chung xu hướng theo thời gian nên chúng ta nhận định chúng là tương quan. Ví dụ: Số lượng người bị đuối nước hàng năm và sản lượng kem tiêu thụ có mối quan hệ cùng chiều (hay

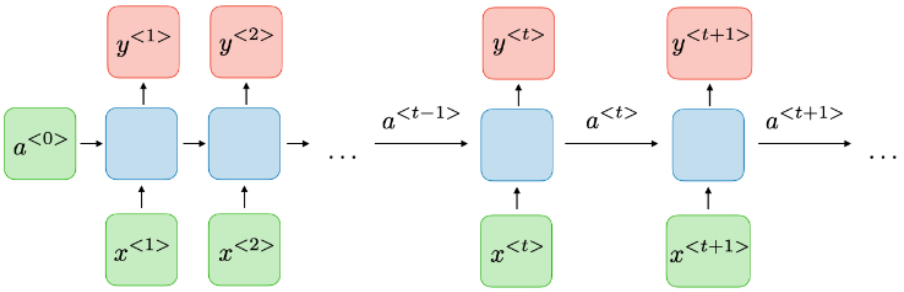
còn gọi là *tương quan tuyến tính dương*). Không khó để chúng ta nhận định được bản chất của sự tương quan này là do chúng có cùng sự tương quan với nhiệt độ. Khi nhiệt độ tăng lên chúng ta đi tắm biển nhiều hơn và dẫn tới số lượng người bị đuối nước cao hơn và đồng thời khi nhiệt độ cao cũng là lúc người ta ăn kem để giải khát nhiều hơn. Tuy nhiên việc ăn kem không phải là nguyên nhân trực tiếp dẫn tới đuối nước. Do đó khi xây dựng các mô hình chuỗi thời gian chúng ta cần loại bỏ yếu tố xu hướng ở những biến input để tìm ra những chuỗi có sự tương quan thực sự.

- Tính chu kỳ: Là quy luật có tính chất lặp lại của dữ liệu theo thời gian. Sự thay đổi thời tiết, sự phát triển của các loài động vật cho tới hành vi mua sắm, tiêu dùng của con người đều bị ảnh hưởng của chu kỳ và lặp lại theo thời gian. Chính vì thế tìm ra được yếu tố chu kỳ sẽ giúp ích cho việc dự báo chính xác hơn. Một ví dụ về tầm quan trọng của chu kỳ đó là các doanh nghiệp sản xuất một mặt hàng cụ thể sẽ biết sản lượng tăng vào thời điểm nào trong năm? Cần phải tuyển thêm bao nhiêu lao động? Mua thêm bao nhiêu nguyên vật liệu để đáp ứng được nhu cầu thị trường. Nếu không hiểu được tính chu kỳ của chuỗi thời gian, doanh nghiệp có thể dự báo sai nhu cầu thị trường và dẫn tới thua lỗ.

1.8 Mạng RNN

1.8.1 Định nghĩa [17]

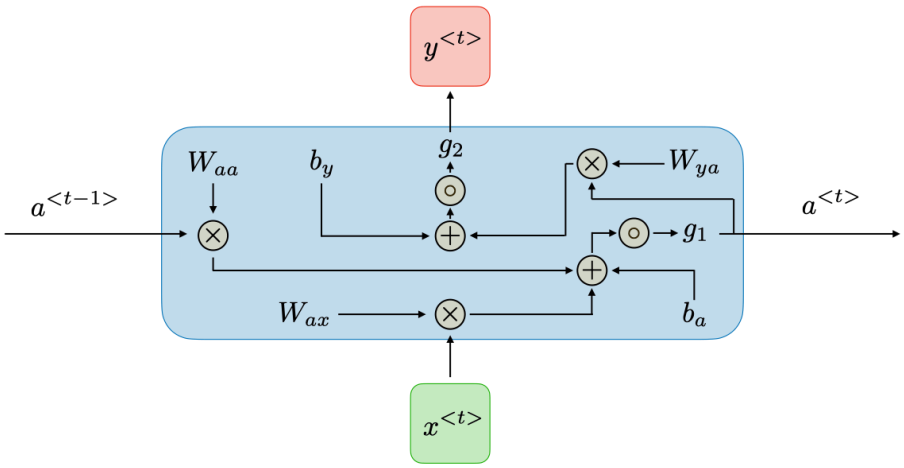
- Mạng nơ-ron tuần hoàn, còn được gọi là RNN, là một lớp mạng nơ-ron cho phép các đầu ra trước đó được sử dụng làm đầu vào trong khi có trạng thái ẩn. Chúng thường như sau:



Hình 1.6 Mạng RNN

1.8.2 Mô hình RNN [17]

- Nó nhận một đầu vào **x**, một vector đại diện đóng vai trò là bộ nhớ để lưu lại các hidden state là **a** rồi tiến hành xử lý và đưa ra 2 đầu ra **y** và **a'.** Điểm đặc biệt của RNN là nó sẽ lưu lại giá trị của **x** để sử dụng cho đầu vào tiếp theo. Cụ thể như sau:



Hình 1.7 Một state của mạng RNN

- Với mỗi step $\mathbf{a}^{<t>}$ là tổng hợp thông tin của state trước và input tại time step **t** là $\mathbf{x}^{<t>}$, activation function $\mathbf{g_1}$ chủ yếu là tanh hoặc ReLu, ta có
  - Hidden state:

$$a^{<t>} = g_1(W_{aa} * a^{<t-1>} + W_{ax} * x^{<t>} + b_a \quad (1.8)$$

  - Output:

$$y^{<t>} = g_2(W_{ya} * a^{<t>} + b_y) \quad (1.9)$$

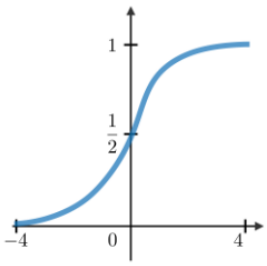
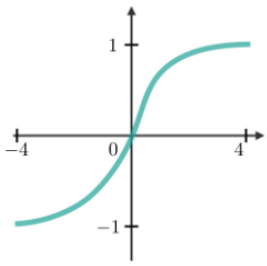
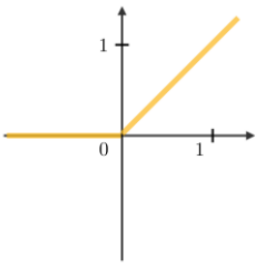
    Với W_{aa}, W_{ax}, b_a, b_y là các hệ số được chia sẻ tạm thời. g_2 là activation function, trong bài này là bài toán phân loại nên sẽ dùng softmax.
- Tóm lại:
  - $x^{<i>}$ là vector có kích thước n*1, $a^{<i>}$ là vector có kích thước m*1, $y^{<i>}$ là vector có kích thước d*1. W_{ax} là ma trận có kích thước m*n, W_{aa} là ma trận m*m, và W_{ya} là ma trận d*m.
  - $a^{<0>}=0$, $a^{<t>}= g_1(W_{ax} * x^{<t>} + W_{aa} * a^{<t-1>})$ với t >= 1 (1.10)
  - $\hat{y} = g(W_{ya} * a^{<t+1>})$ (1.11)
- Loss function:

$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}(\hat{y}^{<t>}, y^{<t>}) \quad (1.12)$$

- Backpropagation through time: được thực hiện tại từng thời điểm, tại step T, đạo hàm của loss function đối với ma trận trọng số W được biểu thị như sau

$$\frac{\partial \mathcal{L}^{(T)}}{\partial W} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}^{(T)}}{\partial W}\Big|_{(t)} \quad (1.13)$$

- Xử lý sự phụ thuộc dài hạn:
  - Các activation functions:

Sigmoid	Tanh	RELU
$g(z) = \frac{1}{1+e^{-z}}$	$g(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$	$g(z) = \max(0, z)$
		

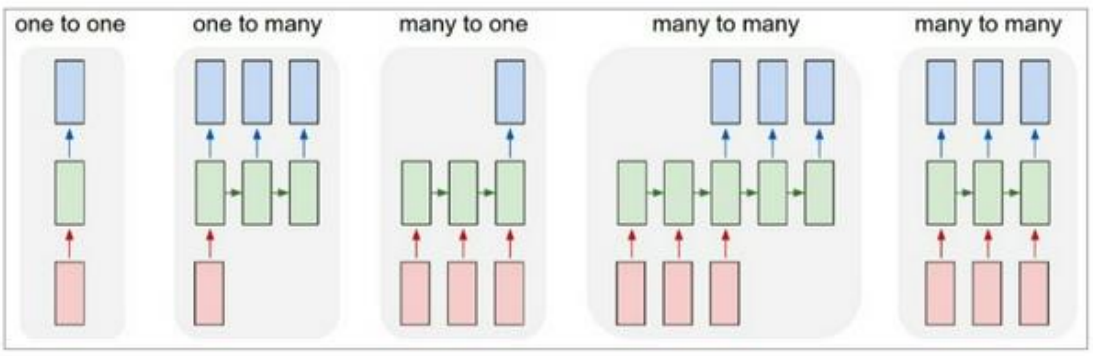
Hình 1.8 Các activation functions

- Để khắc phục vấn đề vanishing gradient, các cổng cụ thể được sử dụng trong một số RNN thường có mục đích rõ ràng. Ký hiệu là T và có công thức chung sau:

$$T = \sigma(W * x^{<t>} + U * a^{<t-1>} + b \qquad (1.14)$$

Trong đó, W, U, b là các hệ số đặc trưng, σ là ký hiệu của hàm sigmoid.

- Chuỗi các đầu vào **x** là những sự kiện xảy ra theo thứ tự thời gian **t**. Những sự kiện này đều có mối liên hệ về thông tin với nhau và thông tin của chúng sẽ được giữ lại để xử lý sự kiện tiếp theo trong mạng neural hồi quy. Vì tính chất này, mạng neural hồi quy phù hợp cho những bài toán với dữ liệu đầu vào dưới dạng chuỗi với các sự kiện trong chuỗi có mối liên hệ với nhau. Vì vậy, mạng neural hồi quy có ứng dụng quan trọng trong các bài toán xử lý ngôn ngữ tự nhiên như: Dịch máy - Neural Machine Translation**,** Phân loại ngữ nghĩa - Semantic classification, Nhận dạng giọng nói: Speech Recognition.
- Một trong những điểm mạnh của mạng neural hồi quy là cho phép tính toán trên một chuỗi các vector. Dưới đây là các kiểu hoạt động của mạng neural hồi quy:



Hình 1.9 Các kiểu mạng RNN

- Mỗi hình chữ nhật là 1 vector và các mũi tên thể hiện các hàm biến đổi. Vector đầu vào có màu đỏ, vector đầu ra có màu xanh biển và vector trạng thái thông tin trao đổi giữa các mạng con có màu xanh lá. Từ trái sang phải ta có:
  - Mạng neural kiểu Vanilla: Đầu vào và đầu ra có kích thước cố định (Bài toán nhận diện ảnh - *Image Classification*)
  - Đầu ra có dạng chuỗi: Đầu vào cố định và đầu ra là một chuỗi các vector (Bài toán tạo tiêu đề cho ảnh - *Image Captioning*)
  - Đầu vào có dạng chuỗi: Đầu vào là một chuỗi vector và đầu ra cố định (Bài toán phân loại ngữ nghĩa - *Sentiment Classification*)
  - Đầu vào và đầu ra có dạng chuỗi: Bài toán Dịch máy - *Neural Machine Translation*
  - Đầu vào và đầu ra có dạng chuỗi đồng bộ: Đầu vào và đầu ra là một chuỗi vector có độ dài bằng nhau (Bài toán phân loại video và gắn nhãn từng frame - *Video Classification*)
- Có thể nhận thấy rằng độ dài các chuỗi đầu vào hay đầu ra tại mỗi trường hợp không bắt buộc phải cố định vì kích thước vector trạng thái thông tin trao đổi trong mạng neural hồi quy là cố định. Giờ chúng ta sẽ đi sâu hơn vào phương thức hoạt động của mạng neural hồi quy.

1.8.3 Ưu điểm và hạn chế của kiến trúc RNN

- Ưu điểm:
  - Khả năng xử lý đầu vào ở bất kỳ độ dài nào
  - Kích thước mô hình không tăng theo kích thước đầu vào
  - Tính toán có tính đến thông tin lịch sử
  - Trọng lượng được chia sẻ theo thời gian
- Hạn chế: [16]
  - Tính toán chậm, thực hiện tuần tự
  - Khó truy cập thông tin đã lâu trước đó ➔ short term memory
  - Không thể xem xét bất kỳ đầu vào nào trong tương lai cho trạng thái hiện tại
  - Vanishing gradient (đạo hàm bị triệt tiêu)
- Để khắc phục những hạn chế của RNN tiêu chuẩn thì LSTM ra đời, nó có khả năng học các phụ thuộc dài hạn.

1.9 Mạng LSTM

1.9.1 Giới thiệu [15]

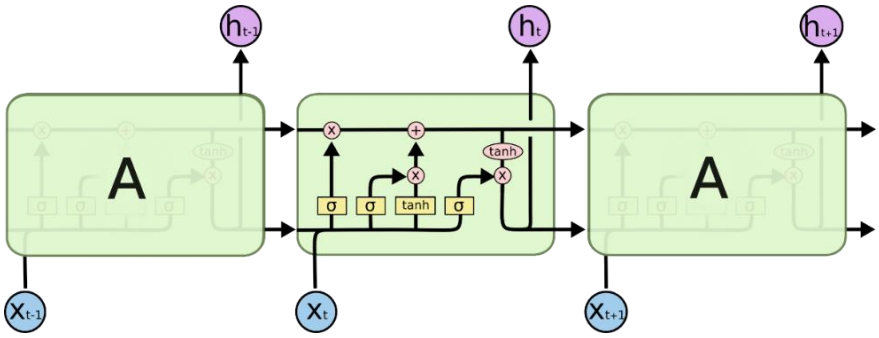
- Bộ nhớ ngắn hạn dài hạn là một loại mạng nơ-ron tuần hoàn. Trong RNN, đầu ra từ bước cuối cùng được cung cấp dưới dạng đầu vào trong bước hiện tại. LSTM được thiết kế bởi Hochreiter & Schmidhuber. Nó giải quyết vấn đề phụ thuộc dài hạn của RNN trong đó RNN không thể dự đoán từ được lưu trữ trong bộ nhớ dài hạn nhưng có thể đưa ra dự đoán chính xác hơn từ thông tin gần đây. Khi độ dài khe hở tăng RNN không cho hiệu suất hiệu quả. Theo mặc định, LSTM có thể giữ lại thông tin trong một khoảng thời gian dài. Nó được sử dụng để xử lý, dự đoán và phân loại trên cơ sở dữ liệu chuỗi thời gian.

- LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

1.9.2 Kiến trúc [15]

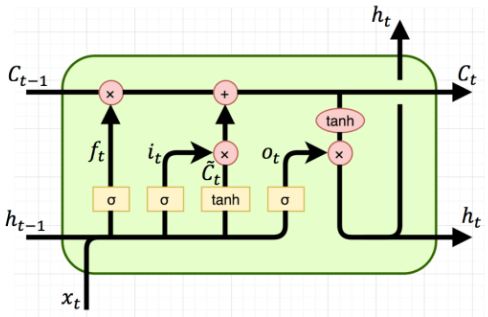
- Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một tầng tanh.

- LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.



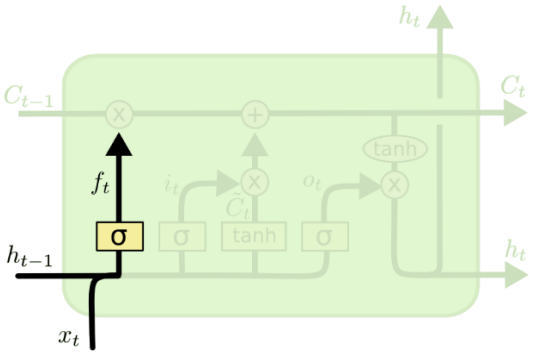
Hình 1.10 Mạng LSTM

- Cụ thể ở mỗi state thứ t của LSTM:



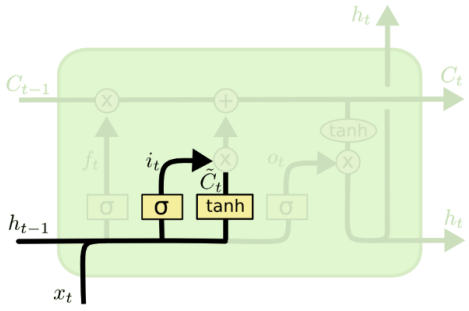
Hình 1.11 Một state của mạng LSTM

- Output: c_t, h_t, ta gọi c là cell state, h là hidden state
- Input: c_{t-1}, h_{t-1}, x_t. Trong đó, x_t đóng vai trò là input ở state thứ t của model. c_{t-1}, h_{t-1} là output của state trước đó. h ở đây có vai trò khá giống với a ở RNN, còn c là điểm mới của LSTM. σ (sigmoid), tanh là các activation funtions. Phép nhân là element-wise multiplication, phép cộng là cộng các ma trận.
- f_t, i_t, o_t ứng với forget gate, input gate và output gate
    - Forget gate: $f_t = \sigma(U_f * x_t + W_t * h_{t-1} + b_f)$ (1.15). Cổng này quyết định lượng thông tin từ state trước bị bỏ đi là bao nhiêu.



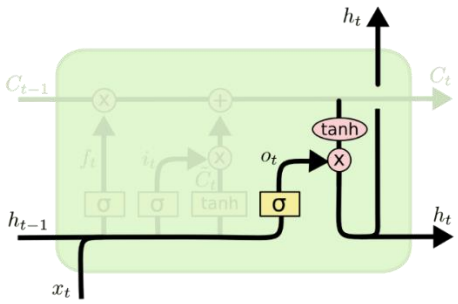
Hình 1.12 Cổng Forget

    - Input gate: $i_t = \sigma(U_i * x_t + W_i * h_{t-1} + b_i)$ (1.16). Cổng này quyết định lượng thông tin đầu vào ảnh hưởng đến state mới là bao nhiêu.



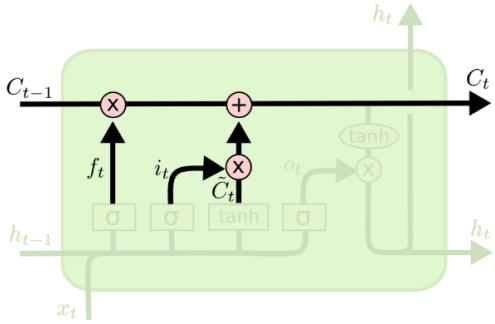
Hình 1.13 Cổng Input

- Output gate: $o_t = \sigma(U_o * x_t + W_o * h_{t-1} + b_o)$ (1.17). Cổng này điều chỉnh lượng thông tin có thể ra ngoài y_t và lượng thông tin tới state tiếp theo.



Hình 1.14 Cổng Output

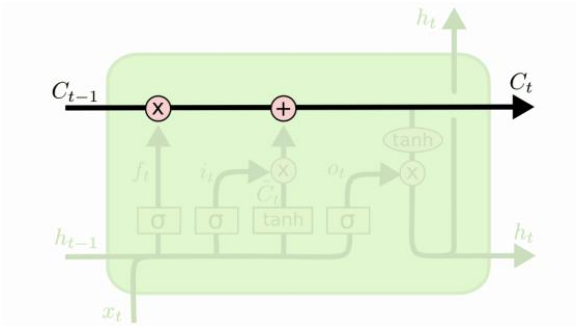
- Nhận xét $0 < f_t, i_t, o_t < 1$ (giá trị của hàm sigmoid nằm trong khoảng [0;1]), b_f, b_i, b_o là hệ số bias, W, U giống với RNN.
- $\tilde{c_t} = \tanh(U_c * x_t + W_c * h_{t-1} + b_c)$ (1.18), giống tính $a^{<t>}$ trong RNN.
- $c_t = f_t * c_{t-1} + i_t * \tilde{c_t}$ (1.19)
- $h_t = o_t + \tanh(c_t)$ (1.20), ngoài ra h_t cũng được dùng để tính ra ouput



Hình 1.15 Giá trị state C

y_t cho state t.

⇨ h_t, $\tilde{c_t}$ khá giống với RNN, nên model có **short term memory**. Trong khi đó c_t giống như một băng chuyền ở mô hình RNN, thông tin nào cần quan trọng và dùng ở sau sẽ đc gửi vào và dùng khi cần ➔ có thể mang đi xa ➔ **long term memory**.



Hình 1.16 Ct của LSTM

- **<u>Tổng kết</u>**:
  - LSTM giải quyết được phần nào vanishing gradient so với RNN.
  - RNN đã chậm thì LSTM còn chậm hơn.
  - Tuy nhiên do được cải tiến hơn RNN, nên LSTM vẫn được sử dụng phổ biến.

CHƯƠNG 2. PHÂN TÍCH BÀI TOÁN

Với sự phát triển của công nghệ thông tin và mạng internet, thì việc tìm hiểu thông tin về chứng khoán không còn là điều khó khăn với mọi người. Nhưng để đưa ra dự đoán về hướng đi lên hay xuống của giá chứng khoán đòi hỏi người đầu tư cần có một lượng kiến thức đủ sâu và rộng. Trải qua nhiều bước tính toán và phân tích phức tạp. Từ đó đưa ra những phân tích chính xác nhất về việc giá chứng khoán đi lên hay đi xuống, vào thời điểm nào thì nên mua, hoặc bán hoặc đầu tư ít rủi ro, thời điểm nào không nên đầu tư,…

Các yếu tố ảnh hưởng đến chiều đi của giá chứng khoán rất nhiều, nên muốn mô hình đưa ra dự đoán chính xác thì người xây dựng mô hình cần hiểu và đưa vào mô hình càng nhiều yếu tố ảnh hưởng càng tốt. Ở bài toán này, vì không có đủ kiến thức về chứng khoán cũng như hiểu biết hết các yếu tổ ảnh hưởng đến giá của chứng khoán nên kết quả không thể áp dụng vào thực tế để đưa ra các quyết định mua bán, hay đầu tư sinh lời.

2.1 Chuẩn bị và phân tích dữ liệu

- Dữ liệu được chọn để sử dụng trong bài toán là lịch sử giá của cổ phiếu công ty Tesla. Dữ liệu được tải về từ trang finance.yahoo.com bao gồm 2014 mẫu, được lấy từ ngày 1/1/2014 đến ngày 30/12/2014, các ngày giao dịch không liên tục do giới hạn giao dịch vào cuối tuần và ngày nghỉ.
- Một vài mẫu trong dữ liệu:

Date	Open	High	Low	Close	Adj Close	Volume
1/2/2014	29.96	30.496	29.31	30.02	30.02	30942000
1/3/2014	30	30.438	29.72	29.912	29.912	23475000
1/6/2014	30	30.08	29.048	29.4	29.4	26805500
1/7/2014	29.524	30.08	29.05	29.872	29.872	25170500
1/8/2014	29.77	30.74	29.752	30.256	30.256	30816000
1/9/2014	30.5	30.686	29.37	29.506	29.506	26910000
1/10/2014	29.692	29.78	28.45	29.144	29.144	37230500
1/13/2014	29.156	29.4	27.564	27.868	27.868	31580500
1/14/2014	28.1	32.4	27.334	32.254	32.254	1.38E+08
1/15/2014	33.69	34.446	32.42	32.826	32.826	1.02E+08
1/16/2014	32.5	34.54	32.48	34.194	34.194	59797000
1/17/2014	34.038	34.64	33.59	34.002	34.002	46031000
1/21/2014	34.248	35.458	34.162	35.336	35.336	48673500
1/22/2014	35.562	36.064	34.952	35.712	35.712	35113000
1/23/2014	35.446	36.476	34.684	36.3	36.3	39337000
1/24/2014	35.57	36.096	34.706	34.92	34.92	38321500
1/27/2014	35.032	35.584	32.942	33.924	33.924	43582000
1/28/2014	34.3	35.796	34.2	35.676	35.676	30467000
1/29/2014	35.06	35.818	34.626	35.046	35.046	29677500
1/30/2014	35.6	36.956	35.402	36.568	36.568	42825000

Hình 2.1 Ví dụ một số mẫu dữ liệu

- Trong đó, các nhãn có nghĩa:
  - Date: là ngày giao dịch
  - Open: Giá mở cửa là giá đóng cửa của phiên giao dịch hôm trước
  - High: là giá cao nhất trong mộ phiên giao dịch hoặc trong một chu kỳ theo dõi biến động giá
  - Low: giá thấp nhất trong một phiên giao dịch hoặc trong một chu kỳ theo dõi biến động giá
  - Close: Giá đóng cửa là giá thực hiện tại lần khớp lệnh cuối cùng trong ngày giao dịch
  - Adj Close: Giá đóng cửa có hiệu chỉnh.
  - Volume: khối lượng giao dịch
- Dữ liệu được sử dụng là dạng dữ liệu time-series, nghĩa là dữ liệu thay đổi theo thời gian. Chu kỳ của dữ liệu là 1 ngày, ta xem mỗi ngày là 1 time-step.
- Cần xử lý để làm sạch dữ liệu nếu dữ liệu có chứa các ký tự đặc biệt, loại bỏ các dữ liệu trống.
- Để tính toán và đưa ra các dự đoán ta dựa vào giá đóng cửa có hiệu chỉnh là cột Adj Close. Trong đó cần tính tỉ suất lợi nhuận giữa các ngày với nhau.
- Công thức tính tỉ suất lợi nhuận theo phần trăm:

$$r_t = \frac{p_t - p_{t-1}}{p_{t-1}} * 100 \qquad (2.1)$$

Tương ứng với (giá đóng cửa hiện tại – giá đóng cửa ngày trước đó)/giá đóng cửa ngày trước đó.

- Ví dụ: trong hình 2.1 ta thấy, ngày 30/1/2014 có giá đóng cửa có hiệu chỉnh là 36.568, ngày 29 có giá đóng cửa có hiệu chỉnh là 35.046. Áp dụng công thức ta tính được tỉ suất lợi nhuận là xấp xỉ 4.34%. Tỷ suất lợi nhuận càng cao thì khả năng sinh lời càng nhiều.
- Tương tự ta tính được tỉ suất lợi nhuận cho hết các mẫu dữ liệu. Sau khi tính xong, tạo một dataframe để lưu trữ các giá trị vừa tính được.
- Tỉ suất lợi nhuận có thể là số âm, trong trường hợp giá đóng cửa có hiệu chỉnh của ngày hiện tại thấp hơn so với ngày trước đó.
- Sau khi có tỉ suất lợi nhuận, tạo một cột Direction trong dataframe lưu trữ trên để gắn nhãn cho dữ liệu đi lên hoặc xuống của các ngày. Nếu tỉ suất là dương thì lưu 1, ngược lại tỉ suất là số âm thì lưu 0. Cột ngày dùng để phục vụ cho việc training model kèm theo cột tỉ suất vừa tính.
- Thực hiện scales cột volume bằng cách lấy giá trị chia cho 100000000.

2.2 Xây dựng mô hình

- Sau khi thu thập và phân tích dữ liệu trước khi tiến hành học máy. Dữ liệu lấy mẫu được chia thành 2 nhóm: tập dữ liệu huấn luyện được sử dụng để thiết lập các mô hình học máy, tập còn lại dùng để test sau khi đã huấn luyện xong.
- Trong bài nghiên cứu này đã sử dụng kỹ thuật Logistic Regression và mô hình LSTM.
- Với thuật toán Logistic Regression:
  - Tính tỉ suất lợi nhuận và gán nhãn cho chúng.
  - Xác định đầu vào là tỉ suất lợi nhuận được tính giữa các ngày, gán nhãn là 0 và 1 cho từng giá trị. Nếu tỉ suất lợi nhuận lớn hơn 0 thì gán là 1, ngược lại gán là 0.
    - Đặt X là biến đầu vào và X là biến đầu ra, ta có: X = (x_1, x_2,…,x_n), Y ϵ {Up, Down}
    - Bài toán sẽ dự báo Y thuộc lớp Up, với Y tương ứng đầu vào x_0, nếu Pr(y = Up | X = x_0) > 0.5; và ngược lại với lớp Down
    - Biến phụ thuộc chỉ có 2 trạng thái tăng/giảm tương ứng 1/0. Muốn đổi ra biến số liên tục ta tính xác suất của 2 trạng thái này. Gọi p là xác suất để biến cố tăng xảy ra, thì 1 – p là xác xuất để biến cố không xảy ra (giảm). Ký hiệu: p(X) = Pr(Y = Up|X). Mô hình hồi quy Logistic có dạng:

      $$Ln\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n \qquad (2.2)$$

      Với $\beta_0, \beta_1, \ldots, \beta_n$ là các hệ số cần ước lượng.
  - Huấn luyện mô hình 2 lần với 2 số lượng biến đầu vào khác nhau
  - Thực hiện dự đoán bằng hàm predict()
  - Đầu ra là chuỗi các số dự đoán
  - Xây dựng ma trận lỗi dựa trên các nhãn và giá trị dự đoán
  - Chia tập dữ liệu train và test
  - Thực hiện dự đoán trên tập test
  - Đánh giá độ chính xác của dự đoán
- Với mô hình LSTM:
  - Vì mỗi ngày là một time step, ta sẽ sử dụng 60 time steps làm input để đưa vào mạng train. Đầu ra là time step tiếp theo. Nghĩa là dùng giá 60 ngày để dự đoán giá của ngày kế tiếp)
  - Scale dữ liệu đầu vào về khoảng [0;1]
  - Xây dựng model với các layer input và output
  - Thực hiện huấn luyện mô hình

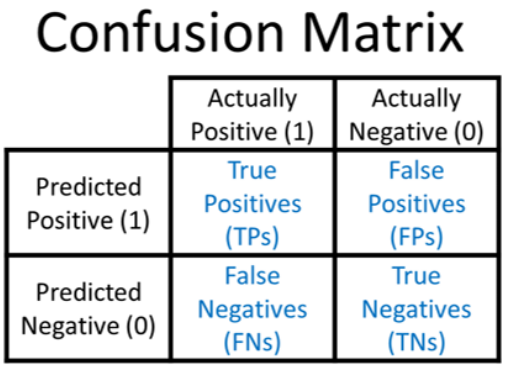
- Thực hiện test mô hình
- Trực quan hóa kết quả test
- Đánh giá mô hình

2.3 Phương pháp đánh giá mô hình

- Khi thực hiện bài toán phân loại, có 4 trường hợp của dự đoán có thể xảy ra: [21]

  - **True Positive (TP):** đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Positive (dự đoán đúng)
  - **True Negative (TN):** đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Negative (dự đoán đúng)
  - **False Positive (FP):** đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Positive (dự đoán sai) – Type I Error
  - **False Negative (FN):** đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Negative (dự đoán sai) – Type II Error

- Bốn trường hợp trên thường được biểu diễn dưới dạng ma trận hỗn loạn (confusion matrix). Chúng ta có thể tạo ra ma trận này sau khi dự đoán xong trên tập dữ liệu thử nghiệm và rồi phân loại các dự đoán vào một trong bốn trường hợp. [21]



Hình 2.2 Confusion matrix

- Trong thực tế có ba độ đo chủ yếu để đánh giá một mô hình phân loại là Accuracy, Precision and Recall: [21]

  - Accuracy được định nghĩa là tỷ lệ phần trăm dự đoán đúng cho dữ liệu thử nghiệm. Nó có thể được tính toán dễ dàng bằng cách chia số lần dự đoán đúng cho tổng số lần dự đoán

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.3)$$

- Precision kiểm tra xem có bao nhiêu kết quả thật là kết quả tích cực trong tổng số các kết quả được dự đoán tích cực

$$Precision = \frac{TP}{TP+FP} \quad (2.4)$$

- Recall: kiểm tra các kết quả dự đoán tích cực chính xác trong số các kết quả tích cực

$$Recall = \frac{TP}{TP+FN} \quad (2.5)$$

- F beta score: là trung bình hài hòa của Accuracy và recall, thể hiện sự đóng góp của cả hai. Sự đóng góp phụ thuộc vào giá trị beta, nếu sự đóng của cả 2 là như nhau thì ta có:

$$F1\ score = 2 * \frac{precision*recall}{precision+recall} \quad (2.6)$$

- Lỗi trung bình bình phương (RMSE) là độ lệch chuẩn của phần dư (lỗi dự đoán). Phần dư là thước đo khoảng cách từ các điểm dữ liệu đường hồi quy; RMSE là thước đo mức độ lan truyền của những phần dư này. Nói cách khác, nó cho bạn biết mức độ tập trung của dữ liệu xung quanh dòng phù hợp nhất. Lỗi bình phương trung bình thường được sử dụng trong khí hậu học, dự báo và phân tích hồi quy để xác minh kết quả thí nghiệm. [20]
- Lỗi trung bình bình phương gốc (RMSE) là thước đo mức độ hiệu quả của mô hình của bạn. Nó thực hiện điều này bằng cách đo sự khác biệt giữa các giá trị dự đoán và giá trị thực tế. R-MSE càng nhỏ tức là sai số càng bé thì mức độ ước lượng cho thấy độ tin cậy của mô hình có thể đạt cao nhất. [20]

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}-y_i)^2}{n}} \quad (2.7)\ [20]$$

CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

3.1 Dữ liệu thực nghiệm

- Phân chia tập dữ liệu: dữ liệu được chia làm hai tập là tập **Train** và tập **Test**.
  - Với thuật toán Logistic Regression
    - Thực nghiệm 1: tập Train được lấy tron giai đoạn năm 2014 đến hết năm 2019 (với 1506 mẫu), tập Test được lấy từ năm 2020 đến hết 2021 (với 506 mẫu)
    - Thực nghiệm 2: tập Train lấy từ năm 2014 đến hết năm 2018 (với 1256 mẫu), còn lại là tập Test từ năm 2019 đến hết 2021 (với 756 mẫu)
  - Với mô hình LSTM: dữ liệu được chia thành 85% Train và 15% Test.

3.2 Môi trường thực nghiệm

- Processor: Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz
- Memory RAM: 16GB
- System type: 64-bit operating system, x64-based processor
- Edition: Windows 10 Home Single Language
- Các thử nghiệm được cài đặt và sử dụng ngôn ngữ Python trên môi trường Visual Code. Với các thư viện của Python như Numpy, Panda, Keras, Matplotlib, Seaborn, Sklearn.

3.3 Xây dựng thực nghiệm

3.3.1 Thực nghiệm với thuật toán Logistic

- Dùng thư viện panda để đọc file dữ liệu TSLA.csv

```
data = pd.read_csv("\Semester 2, 21-22\DATN\Code\TSLA.csv") # doc file
csv
print(data)
```

- Kết quả đọc file csv:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2014-01-02	29.959999	30.496000	29.309999	30.020000	30.020000	30942000
1	2014-01-03	30.000000	30.438000	29.719999	29.912001	29.912001	23475000
2	2014-01-06	30.000000	30.080000	29.048000	29.400000	29.400000	26805500
3	2014-01-07	29.524000	30.080000	29.049999	29.872000	29.872000	25170500
4	2014-01-08	29.770000	30.740000	29.752001	30.256001	30.256001	30816000
...	...	...	...	...	...	...	...
2009	2021-12-23	1006.799988	1072.979980	997.559998	1067.000000	1067.000000	30904400
2010	2021-12-27	1073.670044	1117.000000	1070.719971	1093.939941	1093.939941	23715300
2011	2021-12-28	1109.489990	1119.000000	1078.420044	1088.469971	1088.469971	20108000
2012	2021-12-29	1098.640015	1104.000000	1064.140015	1086.189941	1086.189941	18718000
2013	2021-12-30	1061.329956	1095.550049	1053.150024	1070.339966	1070.339966	15680300

Hình 3.1. Kết quả đọc file csv

➢ **Thực nghiệm 1:**

- Dùng hàm pct_change(). Hàm này so sánh mọi phần tử với phần tử trước của nó và tính tỷ suất lợi nhuận theo công thức 2.1
- Thu được kết quả như sau:

```
0          NaN
1    -0.359757
2    -1.711691
3     1.605442
4     1.285488
        ...
2009  5.761893
2010  2.524830
2011 -0.500025
2012 -0.209471
2013 -1.459227
Name: Adj Close, Length: 2014, dtype: float64
```

Hình 3.2. Tỉ suất lợi nhuận trên cột Adj Close

- Tạo một cột mới tên “Today” để lưu giá trị vừa tính
- Đặt Lag1,…,Lag5 là tỉ suất lợi nhuận của 5 ngày liền trước ngày hiện tại, tức là ngày ở cột “Today”
- Thực hiện tính các Lag, ta thu được kết quả:

	Date	Today	Lag1	Lag2	Lag3	Lag4	Lag5
0	2014-01-02	NaN	NaN	NaN	NaN	NaN	NaN
1	2014-01-03	-0.359757	NaN	NaN	NaN	NaN	NaN
2	2014-01-06	-1.711691	-0.359757	NaN	NaN	NaN	NaN
3	2014-01-07	1.605442	-1.711691	-0.359757	NaN	NaN	NaN
4	2014-01-08	1.285488	1.605442	-1.711691	-0.359757	NaN	NaN
...	...	...	...	...	...	...	...
2009	2021-12-23	5.761893	7.494695	4.288067	-3.498934	0.609548	-5.027716
2010	2021-12-27	2.524830	5.761893	7.494695	4.288067	-3.498934	0.609548
2011	2021-12-28	-0.500025	2.524830	5.761893	7.494695	4.288067	-3.498934
2012	2021-12-29	-0.209471	-0.500025	2.524830	5.761893	7.494695	4.288067
2013	2021-12-30	-1.459227	-0.209471	-0.500025	2.524830	5.761893	7.494695

[2014 rows x 7 columns]

Hình 3.3. Tỉ suất lợi nhuận 5 ngày

- Scale khối lượng cổ phiếu bằng cách lấy "số lượng / 100000000", đặt tên cột là "vol", thu được:

	Date	Today	Vol
0	2014-01-02	NaN	NaN
1	2014-01-03	-0.359757	0.309420
2	2014-01-06	-1.711691	0.234750
3	2014-01-07	1.605442	0.268055
4	2014-01-08	1.285488	0.251705
...	...	...	...
2009	2021-12-23	5.761893	0.312114
2010	2021-12-27	2.524830	0.309044
2011	2021-12-28	-0.500025	0.237153
2012	2021-12-29	-0.209471	0.201080
2013	2021-12-30	-1.459227	0.187180

Hình 3.4. Dữ liệu volume đã scale về khoảng [0;1]

- Sử dụng hàm dropna() để loại bỏ các dòng có dữ liệu trống.
- Đồng thời xác định chiều tăng - giảm "Direction" của giá chứng khoán dựa theo giá trị cột "Today", nếu giá trị lớn 0 thì Direction là 1, ngược lại là 0, thu được:

	Date	Today	Lag1	Lag2	Lag3	Lag4	Lag5	Vol	Direction
6	2014-01-10	-1.226876	-2.478847	1.285488	1.605442	-1.711691	-0.359757	0.269100	0
7	2014-01-13	-4.378256	-1.226876	-2.478847	1.285488	1.605442	-1.711691	0.372305	0
8	2014-01-14	15.738489	-4.378256	-1.226876	-2.478847	1.285488	1.605442	0.315805	1
9	2014-01-15	1.773417	15.738489	-4.378256	-1.226876	-2.478847	1.285488	1.380350	1
10	2014-01-16	4.167428	1.773417	15.738489	-4.378256	-1.226876	-2.478847	1.023280	1
...	...	...	...	...	...	...	...	...	...
2009	2021-12-23	5.761893	7.494695	4.288067	-3.498934	0.609548	-5.027716	0.312114	1
2010	2021-12-27	2.524830	5.761893	7.494695	4.288067	-3.498934	0.609548	0.309044	1
2011	2021-12-28	-0.500025	2.524830	5.761893	7.494695	4.288067	-3.498934	0.237153	0
2012	2021-12-29	-0.209471	-0.500025	2.524830	5.761893	7.494695	4.288067	0.201080	0
2013	2021-12-30	-1.459227	-0.209471	-0.500025	2.524830	5.761893	7.494695	0.187180	0

Hình 3.5. Dữ liệu đã được loại bỏ các giá trị trống

- Sử dụng hàm Logit() với biến đầu ra là "Direction", các biến đầu vào là "Lag" từ 1 đến 5 và "vol". Kết quả trả về:

```
Current function value: 0.691268
Iterations 4
                          Logit Regression Results
==============================================================================
Dep. Variable:              Direction   No. Observations:                 2008
Model:                          Logit   Df Residuals:                     2002
Method:                           MLE   Df Model:                            5
Date:                Sat, 18 Jun 2022   Pseudo R-squ.:                0.001679
Time:                        08:31:52   Log-Likelihood:                -1388.1
converged:                       True   LL-Null:                       -1390.4
Covariance Type:            nonrobust   LLR p-value:                    0.4576
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Lag1          -0.0209      0.013     -1.578      0.115      -0.047       0.005
Lag2           0.0156      0.013      1.177      0.239      -0.010       0.041
Lag3          -0.0047      0.013     -0.354      0.724      -0.030       0.021
Lag4          -0.0007      0.013     -0.051      0.959      -0.026       0.025
Lag5          -0.0108      0.013     -0.820      0.412      -0.037       0.015
Vol            0.1810      0.099      1.827      0.068      -0.013       0.375
==============================================================================
```

Hình 3.6. Bảng giá trị của hàm Logit

- Trong bảng kết quả có hệ số coefficients, sai số chuẩn standard errors, kiểm định z z-statistic, p-values biến có giá trị thống kê.
- Dữ liệu trên bảng cho thấy các biến "Lag" không có giá trị thống kê.
- Thực hiện huấn luyện mô hình dự đoán bằng hàm predict(), các dự đoán thu được là các giá trị từ 0 đến 1, biểu thị cho xác suất tăng của giá cổ phiếu. Nên nếu giá trị lớn hơn 0.5 thì là tăng (Up), còn nhỏ hơn 0.5 là giảm (Down).

```
prediction = result.predict(x)
```

với x là các biến đầu vào "Lag" từ 1 đến 5 và "vol"

- Thiết lập ma trận lỗi (confusion matrix) dựa trên chuyển đổi giá trị dự đoán (Predicted) và giá trị thật (Real) của giá chứng khoán về 2 chỉ số Up - Down để đo lường hiệu suất của dự đoán. Từ đó tính độ chính xác của dự đoán.

```
def confusion_matrix(real, pred):
    predtrans = ['Up' if i > 0.5 else 'Down' for i in pred]
    rl = ['Up' if i > 0 else 'Down' for i in real]
    confusion_matrix = pd.crosstab(pd.Series(rl),
pd.Series(predtrans), rownames=['Real'],colnames=['Predicted'])
    return confusion_matrix
```

- Kết quả dự đoán:

```
6       0.529489
7       0.516467
8       0.530697
9       0.461641
10      0.608498
          ...
2009    0.509177
2010    0.506977
2011    0.519937
2012    0.501988
2013    0.483485
Length: 2008, dtype: float64
```

Hình 3.7. Kết quả dự đoán trên tập huấn luyện

- Kết quả ma trận:

Predicted	Down	Up
Real		
Down	253	713
Up	242	800

Hình 3.8. Ma trận lỗi của tập huấn luyện

  - Độ chính xác của dự đoán Accuracy:

$$accuracy = \frac{253 + 800}{253 + 713 + 242 + 800} = 0.6240039841$$

Độ chính xác dự đoán của huấn luyện này là xấp xỉ 62%

- Các giá trị được tăng thật được dự đoán tăng là

$$precision = \frac{800}{242 + 800} = 0.7677543186$$

Vậy số giá trị tăng thật được dự đoán tăng là xấp xỉ 77%

- Thực hiện chia tập dữ liệu để train và test.
- Tập x_train là các biến đầu vào gồm 5 biến Lag và 1 biến vol. y_train là biến Direction

```
x_train = df[df.year < 2020][name]
y_train = df[df.year < 2020]['Direction']
x_test = df[df.year >= 2020][name]
y_test = df[df.year >= 2020]['Direction']
```

- x_train, y_train lấy các giá trị từ 2014 đến cuối năm 2019
- x_test, y_test lấy giá trị từ năm 2020 đến hết

- Huấn luyện trên tập train như sau:

```
model = sm.Logit(y_train,x_train)
rs = model.fit()
```

- Dự đoán trên tập test:

```
prediction = rs.predict(x_test)
```

- Ma trận lỗi có dạng:

Predicted	Down	Up
Real		
Down	116	106
Up	143	139

Hình 3.9. Ma trận lỗi của tập test (biến là 5 ngày)

- Độ chính xác của dự đoán Accuracy:

$$accuracy = \frac{116 + 139}{116 + 106 + 143 + 139} = 0.505952381$$

Độ chính xác dự đoán của thực nghiệm này là xấp xỉ 50%

- Các giá trị được tăng thật được dự đoán tăng là

$$precision = \frac{139}{139 + 143} = 0.4929078014$$

Vậy số giá trị tăng thật được dự đoán tăng là xấp xỉ 49%.

➢ **Thực nghiệm 2:**

- Tính tỉ suất lợi nhuận của 1 ngày trước ngày dự đoán:

	Date	Today	Lag1
0	2014-01-02	NaN	NaN
1	2014-01-03	-0.359757	NaN
2	2014-01-06	-1.711691	-0.359757
3	2014-01-07	1.605442	-1.711691
4	2014-01-08	1.285488	1.605442
...	...	...	...
2009	2021-12-23	5.761893	7.494695
2010	2021-12-27	2.524830	5.761893
2011	2021-12-28	-0.500025	2.524830
2012	2021-12-29	-0.209471	-0.500025
2013	2021-12-30	-1.459227	-0.209471

[2014 rows x 3 columns]

Hình 3.10 Tỉ suất lợi nhuận của 1 ngày trước ngày dự đoán

- Tương tự ở thực nghiệm 1, scale volume về khoảng [0;1], loại bỏ dữ liệu trống và gắn nhãn, ta được:

	Date	Today	Vol	Lag1	Direction
2	2014-01-06	-1.711691	0.234750	-0.359757	0
3	2014-01-07	1.605442	0.268055	-1.711691	1
4	2014-01-08	1.285488	0.251705	1.605442	1
5	2014-01-09	-2.478847	0.308160	1.285488	0
6	2014-01-10	-1.226876	0.269100	-2.478847	0
...	...	...	...	...	...
2009	2021-12-23	5.761893	0.312114	7.494695	1
2010	2021-12-27	2.524830	0.309044	5.761893	1
2011	2021-12-28	-0.500025	0.237153	2.524830	0
2012	2021-12-29	-0.209471	0.201080	-0.500025	0
2013	2021-12-30	-1.459227	0.187180	-0.209471	0

[2012 rows x 5 columns]

Hình 3.11 Dữ liệu cho thực nghiệm 2

- Kết quả dự đoán trên tập huấn luyện:

```
2       0.512569
3       0.521110
4       0.503120
5       0.507357
6       0.525141
          ...
2009    0.475253
2010    0.484113
2011    0.497673
2012    0.511764
2013    0.509620
Length: 2012, dtype: float64
```

Hình 3.12 Kết quả dự đoán tập huấn luyện (thực nghiệm 2)

- Ma trận lỗi của tập huấn luyện:

Predicted	Down	Up
Real		
Down	149	819
Up	142	902

Hình 3.13 Ma trận lỗi của tập huấn luyện (thực nghiệm 2)

  - Độ chính xác của dự đoán Accuracy:

$$accuracy = \frac{149 + 902}{149 + 819 + 142 + 902} = 0.5223658052$$

Độ chính xác dự đoán của tập test này là xấp xỉ 52%

  - Các giá trị được tăng thật được dự đoán tăng là

$$precision = \frac{902}{142 + 902} = 0.8639846743$$

Vậy số giá trị tăng thật được dự đoán tăng là xấp xỉ 86%

- Thực hiện chia tập dữ liệu để train và test.
- Tập x_train là các biến đầu vào gồm 1 biến Lag và 1 biến vol. y_train là biến tập hợp nhãn Direction

Predicted	Down	Up
Real		
Down	155	187
Up	171	243

Hình 3.14. Ma trận lỗi của tập test (biến là 1 ngày)

  - Độ chính xác của dự đoán Accuracy:

$$accuracy = \frac{155 + 243}{155 + 187 + 171 + 243} = 0.5264550265$$

Độ chính xác dự đoán của tập test này là xấp xỉ 53%

  - Các giá trị được tăng thật được dự đoán tăng là

$$precision = \frac{243}{171 + 243} = 0.5869565217$$

Vậy số giá trị tăng thật được dự đoán tăng là xấp xỉ 59%

- **Nhận xét**: Qua kết quả 2 thực nghiệm trên ta thấy, nếu chỉ lấy biến đầu vào cho thuật toán Logistic Regression là 1 ngày trước ngày được đoán, thì độ chính xác của dự đoán sẽ cao hơn so với khi lấy các biến đầu vào là 5 ngày trước đó. Tuy nhiên, độ chính xác còn khá thấp, chưa có tính tin cậy cao. Qua thực nghiệm với thuật toán Logistic Regression cho thấy sự biến động tăng/giảm của giá chứng khoán phụ thuộc vào giá ngày liền trước đó, không phụ thuộc vào khối lượng giao dịch.

3.3.2 Thực nghiệm với mô hình LSTM

- Đọc data từ trang finance.yahoo.com

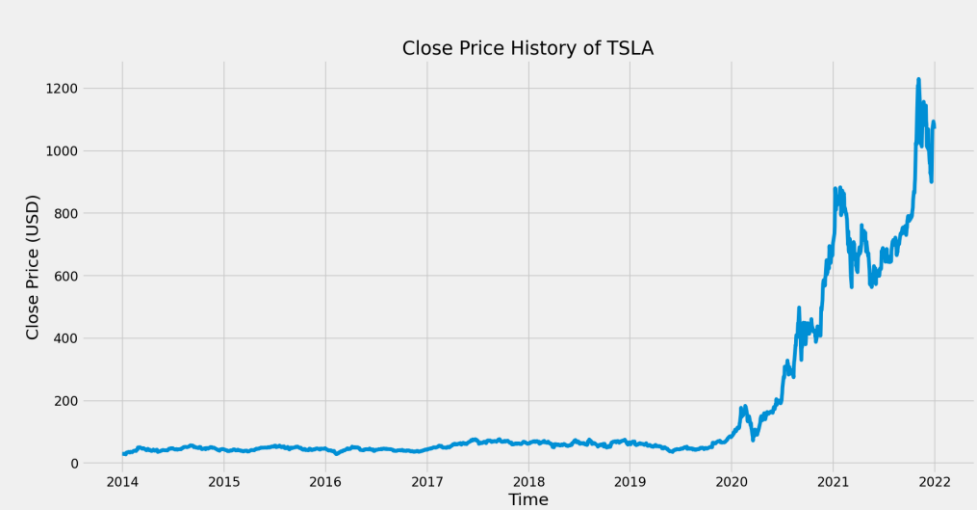
```
df = data.DataReader('TSLA', data_source='yahoo', start ='2014-01-01', end='2021-12-30')
```

Date	High	Low	Open	Close	Volume	Adj Close
2013-12-31	30.639999	29.732000	30.464001	30.086000	21312000.0	30.086000
2014-01-02	30.496000	29.309999	29.959999	30.020000	30942000.0	30.020000
2014-01-03	30.438000	29.719999	30.000000	29.912001	23475000.0	29.912001
2014-01-06	30.080000	29.048000	30.000000	29.400000	26805500.0	29.400000
2014-01-07	30.080000	29.049999	29.524000	29.872000	25170500.0	29.872000
...	...	...	...	...	...	...
2021-12-23	1072.979980	997.559998	1006.799988	1067.000000	30904400.0	1067.000000
2021-12-27	1117.000000	1070.719971	1073.670044	1093.939941	23715300.0	1093.939941
2021-12-28	1119.000000	1078.420044	1109.489990	1088.469971	20108000.0	1088.469971

[2015 rows x 6 columns]

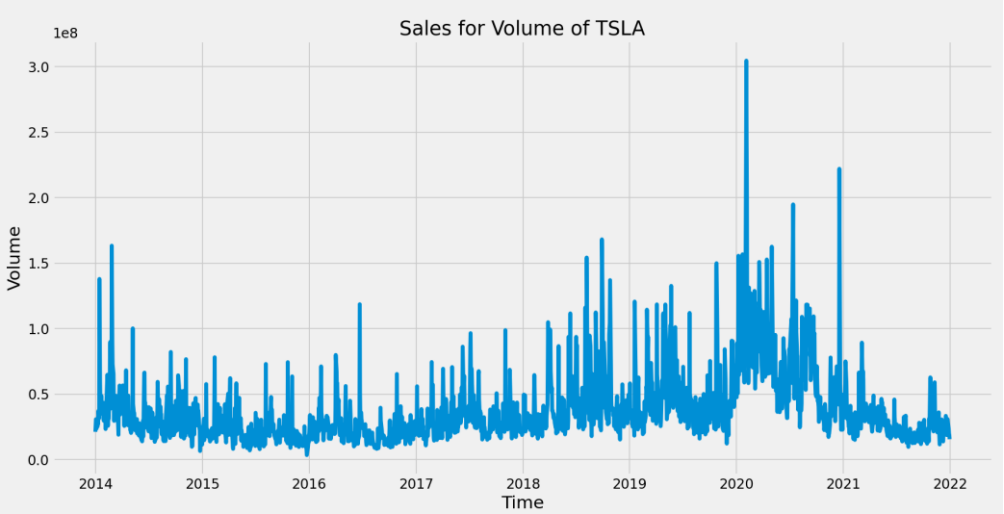
Hình 3.15. Dữ liệu thực nghiệm mô hình LSTM

- Mô hình hóa cột "Close":



Hình 3.16. Biểu đồ cột Close price

- Mô hình hóa cột "Volume"



Hình 3.17. Biểu đồ cột Volume

- Tạo một Dataframe từ cột "Close", convert sang mảng numpy. Lấy 85% của mảng để thực hiện train model, tương đương 1731 mẫu.
- Scale data về khoảng [0;1]. Data sau khi đã scale:

```
[[0.00184519]
 [0.00179029]
 [0.00170044]
 ...
 [0.88233351]
 [0.88043672]
 [0.86725084]]
```

Hình 3.18. Data sau khi đã scale về khoảng [0;1]

- Tạo dataset để train, lấy lượng mẫu từ đầu đến mẫu 1731, và tất cả các cột. Chia làm 2 phần là x_train và y_train datasets. Ta có 2 tập x, y như sau:

```
[array([0.00184519, 0.00179029, 0.00170044, 0.0012745 , 0.00166716,
       0.00198662, 0.00136268, 0.00106153, 0.        , 0.00364879,
       0.00412465, 0.00526271, 0.00510298, 0.00621276, 0.00652556,
       0.00701473, 0.00586668, 0.00503809, 0.00649561, 0.00597151,
       0.00723768, 0.00699976, 0.00628431, 0.00655385, 0.00583673,
       0.00649561, 0.00785164, 0.00952047, 0.00953045, 0.00931415,
       0.01003126, 0.00979833, 0.01070845, 0.00903463, 0.01175167,
       0.01169011, 0.01302949, 0.01807923, 0.01891115, 0.01883462,
       0.01754847, 0.01850518, 0.0192173 , 0.01885458, 0.01890117,
       0.01778141, 0.01655516, 0.01581808, 0.01699608, 0.01638046,
       0.01524572, 0.01574654, 0.01675482, 0.01605601, 0.01590127,
       0.01489965, 0.01344878, 0.01349371, 0.01224915, 0.01131075])]
[0.012150988549463119]
```

Hình 3.19. Tập dataset training

- Xây model LSTM để dự đoán giá chứng khoán:
  - ***Thực nghiệm 1***: trên model gồm 4 lớp, mỗi lớp 50 neurons, và 1 output layer:

```
# Build the LSTM model
model = Sequential()
# 1st layer
model.add(LSTM(50, return_sequences=True,
input_shape=(x_train.shape[1], 1))) # 50 neuron
model.add(Dropout(0.2))
# 2nd layer
model.add(LSTM(50, return_sequences=True))
model.add(Dropout(0.2))
# 3rd layer
model.add(LSTM(50, return_sequences= True))
model.add(Dropout(0.2))
# 4th layer
```

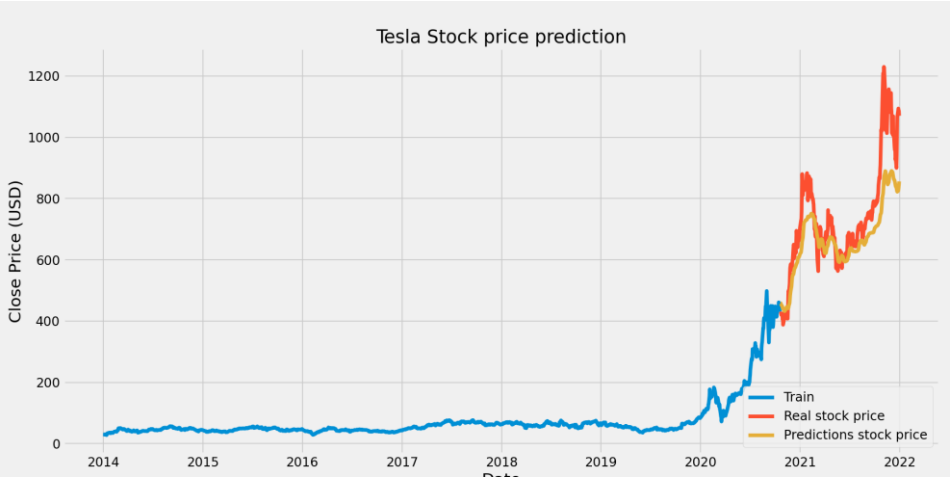
```
model.add(LSTM(50))
model.add(Dropout(0.2))
#output layer
model.add(Dense(1))
```

- Thực hiện biên dịch và train dataset, học 50 lần trên 1 model, với batches là 50 ta thu được kết quả:

```
Epoch 1/50
34/34 [==============================] - 6s 53ms/step - loss: 7.7090e-04
Epoch 2/50
34/34 [==============================] - 2s 50ms/step - loss: 2.2825e-04
Epoch 3/50
34/34 [==============================] - 2s 50ms/step - loss: 3.2075e-04
Epoch 4/50
34/34 [==============================] - 2s 49ms/step - loss: 3.3169e-04
Epoch 5/50
34/34 [==============================] - 2s 49ms/step - loss: 2.0501e-04
Epoch 6/50
34/34 [==============================] - 2s 50ms/step - loss: 2.0108e-04
Epoch 7/50
34/34 [==============================] - 2s 51ms/step - loss: 2.1686e-04
Epoch 8/50
34/34 [==============================] - 2s 52ms/step - loss: 2.3790e-04
Epoch 9/50
34/34 [==============================] - 2s 50ms/step - loss: 1.8137e-04
Epoch 10/50
34/34 [==============================] - 2s 51ms/step - loss: 4.8608e-04
```

Hình 3.20. Quá trình học trên model

- Tạo tập dataset x_test và y_test, định hình chúng và thực hiện dự đoán dựa trên tập test. Train thành công thì tính lỗi trung bình bình phương gốc, để đo mức độ hiệu quả của mô hình ta được: `113.02707599605336` . Nó thực hiện điều này bằng cách đo sự khác biệt giữa các giá trị dự đoán và giá trị thực tế. R-MSE càng nhỏ tức là sai số càng bé thì mức độ ước lượng cho thấy độ tin cậy của mô hình có thể đạt cao nhất.



Hình 3.21. Biểu đồ dự đoán của thực nghiệm 1

- Sự chênh lệch thể hiện qua con số:

Date	Close	Predictions
2020-10-20	421.940002	464.718048
2020-10-21	422.640015	464.486694
2020-10-22	425.790009	462.229797
2020-10-23	420.630005	458.766449
2020-10-26	420.279999	454.764069
...	...	...
2021-12-23	1067.000000	804.648315
2021-12-27	1093.939941	808.739990
2021-12-28	1088.469971	818.819885
2021-12-29	1086.189941	832.223389
2021-12-30	1070.339966	845.883362

Hình 3.22 Sự chênh lệch giữa giá dự đoán và giá thực ở thực nghiệm 1

➔ **Nhận xét**: đường giá dự đoán thấp hơn so với giá thật của cổ phiếu, độ chênh lệch còn nhiều, cho thấy mô hình này chưa có độ tin cậy cao.

- ***Thực nghiệm 2***: model gồm 2 lớp, lớp 1: 128 neurons, lớp 2 64 neurons và 2 ouptut layer lần lượt là 25 neurons và 1 neuron

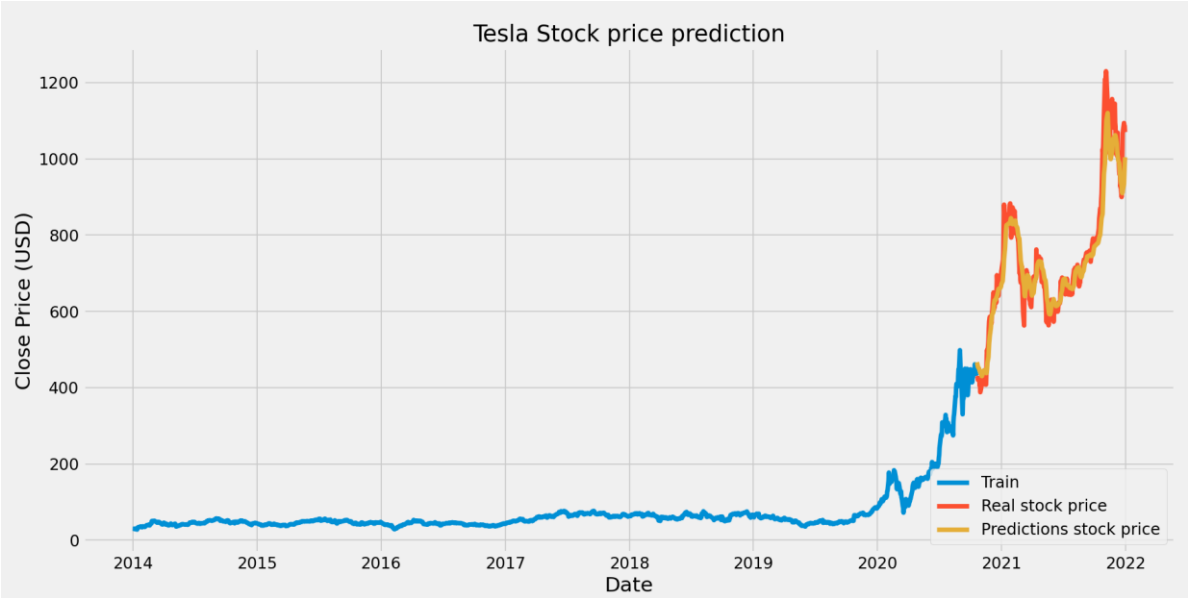
```
    # 1st layer
model.add(LSTM(128, return_sequences=True,
input_shape=(x_train.shape[1], 1))) # 50 neuron
    # 2nd layer
model.add(LSTM(64, return_sequences=False))
   # output layer
model.add(Dense(25)) #25 noron
model.add(Dense(1))
```

  - Thực hiện biên dịch và train dataset, học 2 lần trên 1 model, với batches 1

```
Epoch 1/2
1653/1653 [==============================] - 21s 11ms/step - loss: 8.9441e-04
Epoch 2/2
1653/1653 [==============================] - 19s 11ms/step - loss: 2.2390e-04
```

Hình 3.23 Quá trình học trên model thực nghiệm 2

  - Lỗi trung bình bình phương gốc: 48.77686516414727
  - Biểu đồ dự đoán:



Hình 3.24. Biểu đồ dự đoán của thực nghiệm 2

- Sự chênh lệch về dự đoán được thể hiện dạng số:

Date	Close	Predictions
2020-10-20	421.940002	488.163879
2020-10-21	422.640015	481.290314
2020-10-22	425.790009	475.567230
2020-10-23	420.630005	472.504669
2020-10-26	420.279999	469.827209
...	...	...
2021-12-23	1067.000000	997.286987
2021-12-27	1093.939941	1028.250122
2021-12-28	1088.469971	1061.151855
2021-12-29	1086.189941	1084.703369
2021-12-30	1070.339966	1098.861816

Hình 3.25 Sự chênh lệch giữa giá dự đoán và giá thực ở thực nghiệm 2

➔ **Nhận xét:** đường dự đoán chênh lệch không nhiều so với đường giá thật, cho thấy mức độ tin cậy của mô hình này khá cao. Với việc cho học 2 lần trên một model, nhận thấy sự hiệu quả hơn của mô hình LSTM.

- Như vậy ta thấy, việc xây dựng model với ít layer hơn kết hợp số lần học trên một model ít hơn và khối lượng mẫu mỗi lần học nhiều hơn (ở thực nghiệm 2) cho kết quả tốt hơn so với thực nghiệm 1. Cụ thể là đường giá dự đoán ở thực nghiệm 2 gần đường giá thực tế hơn so với ở thực nghiệm 1.

KẾT LUẬN

1. Kết quả đạt được

- Về bản thân:
  - Nâng cao kỹ năng đọc - hiểu, tìm kiếm tài liệu, đặc biệt là tài liệu bằng tiếng Anh. Từ đó cải thiện vốn tiếng Anh của bản thân.
  - Nâng cao khả năng tự học, nghiên cứu và tìm cách giải quyết vấn đề.
  - Học được thêm nhiều kiến thức mới và các ứng dụng.
  - Biết cách trình bày báo cáo một cách chính xác, rõ ràng, khoa học.
  - Kỹ năng sắp xếp, phân chia thời gian biểu hợp lý.
- Về bài nghiên cứu:
  - Trong bài nghiên cứu này, đã thực nghiệm được 2 mô hình học máy và học sâu, với mỗi mô hình là 2 thực nghiệm. Kết quả cho thấy mô hình LSTM với thực nghiệm 2 cho kết quả tốt nhất trong bộ dữ liệu.
  - Kết quả của thực nghiệm mang tính chính xác tương đối, vì khoảng cách giữa giá trị thật và giá trị dự đoán còn khá xa.
  - Kết quả của các mô hình được đánh giá dựa trên các phương pháp khác nhau đã làm nổi bật lên điểm mạnh của mô hình được đề xuất. Tuy nhiên, các kết quả thực nghiệm chưa thể ứng dụng vào đời sống mà chỉ để phục vụ nghiên cứu. Vì thực nghiệm mang tính chủ quan, trên thực tế thì giá của cổ phiếu còn bị ảnh hưởng bởi nhiều yếu tố khách quan khác. Đòi hỏi người xây dựng mô hình có kiến thức sâu rộng hơn về mảng tài chính để áp dụng công nghệ vào. Từ đó mới đưa ra những dự đoán đáng tin cậy.

2. Hạn chế

- Kiến thức về mảng chứng khoán còn hạn hẹp nên chưa thể xây dựng được mô hình có những yếu tố ảnh hưởng tới kết quả dự đoán.
- Dữ liệu còn hạn chế chuỗi thời gian liên tục và chưa phong phú.
- Phạm vi sử dụng kết quả còn hạn chế, độ chính xác chưa cao
- Chưa thực hiện dự đoán được với mốc thời gian ngắn hơn ví dụ như 12 tiếng, hoặc 6 tiếng.
- Chưa đánh giá toàn diện được các phương pháp thử nghiệm.

3. Hướng phát triển

- Đây là một hướng đi nhiều tiềm năng phát triển trong tương lai. Đòi hỏi người xây dựng mô hình dự đoán không chỉ có kiến thức về công nghệ thông tin, mà còn cần trau dồi thêm kiến thức về tài chính để phát triển và xây dựng mô hình một cách tốt nhất.

- Có thể thử nghiệm trên các mô hình học máy khác và so sánh để cho ra cái nhìn tổng quan hơn về việc dự đoán giá chứng khoán, đồng thời tìm ra mô hình tối ưu cho bài toán dự đoán giá chứng khoán.
- Tìm cách thu thập dữ liệu chi tiết hơn nữa, cụ thể là dữ liệu với chu kỳ thời gian ngắn hơn 24 tiếng, chẳng hạn như theo chu kỳ 12 tiếng, 6 tiếng, hoặc 3 tiếng.
- Thêm các yếu tố có thể ảnh hưởng đến dự đoán vào mô hình để kết quả dự đoán mang tính chính xác cao hơn.

TÀI LIỆU THAM KHẢO

[1] Giáo trình Thị trường chứng khoán, chủ biên: TS. Bạch Đức Hiển, năm 2009, nhà xuất bản Tài Chính, chương 1, 2.

[2] Master ML Algorithms, Jason Brownlee, © Copyright 2016 Jason Brownlee. All Rights Reserved. Chapter III.13.

[3] Introduction to ML with Python: A guide for Data Scientists, Andreas C. Müller & Sarah Guido, Printed in the United States of America, Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. Chapter 1.

[4] Giáo trình Python cơ bản, TS. Nguyễn Văn Hậu – TS. Nguyễn Duy Tân – ThS. Nguyễn Thị Hải Năng – ThS. Nguyễn Hoàng Hiệp, năm 2019, nhà xuất bản Đại học Quốc gia Hà Nội. Bài 1.

[5] Libraries in Python, article contributed by: parthmanchanda81, last update: 18 Oct, 2021. Published in GeeksforGeeks. Link: Libraries in Python - GeeksforGeeks

[6] Python Keras | keras.utils.to_categorical(), article contributed by: manmayi, last update: 23 Jun, 2021. Published in GeeksforGeeks. Link: Python Keras | keras.utils.to_categorical() - GeeksforGeeks

[7] Introduction to Seaborn – Python, article contributed by: 09amit, last update: 03 Jun, 2020. Published in GeeksforGeeks. Link: Introduction to Seaborn - Python - GeeksforGeeks

[8] Article "Implement Logistic Regression from scractch in Python", by Casper Hansen. Published February 14, 2022. Link: Implementing logistic regression from scratch in Python - IBM Developer

[9] What is Logistic Regression?, viết bởi IBM, truy cập lần cuối ngày: 24/6/2022. Link: What is Logistic regression? | IBM

[10] Trang wiki: bài "Học sâu". Truy cập lần cuối: 20/6/2022. Link: Học sâu – Wikipedia tiếng Việt

[11] Bài viết "Mô hình Logit & Probit – Logistic Regression in Stata", Tấn Đăng. Ngày đăng: 04/01/2022. Truy cập lần cuối: ngày 24/6/2022. Link: Mô hình Logit & Probit – Logistic Regression in Stata [2022] (mosl.vn)

[12] Bài viết "7 Trường Hợp Sử Dụng Machine Learning Trong Ngân Hàng", ngày đăng: 5/1/2022. Truy cập lần cuối: ngày 24/6/2022. Link: 7 Trường Hợp Sử Dụng Machine Learning Trong Ngân Hàng (akabot.com)

[12] Bài viết “Dữ liệu chuỗi thời gian”, Phạm Đình Khánh. Truy cập lần cuối: ngày 24/6/2022. Link:

[13] Trang wiki: bài “Học máy”. Truy cập lần cuối: 24/6/2022. Link: Học máy – Wikipedia tiếng Việt

[14] Bài viết “Time-series data”, Hoàng Đức Quân. Truy cập lần cuối: 24/6/2022. Link: Time-Series Data (viblo.asia)

[15] Bài viết “Understanding LSTM Networks”, Posted on August 27, 2015. Truy cập lần cuối: 24/6/2022. Link: Understanding LSTM Networks -- colah's blog

[16] Bài viết “Recurrent Neural Network: Từ RNN đến LSTM”, Nguyễn Thanh Huyền, đăng ngày 24/6//2021. Truy cập lần cuối: 24/6/2022. Link: Recurrent Neural Network: Từ RNN đến LSTM (viblo.asia)

[17] Bài viết “Recurrent Neural Networks cheatsheet”, Afshine and Shervine Amidi. Truy cập lần cuối: 24/6/2022. Link: CS 230 - Recurrent Neural Networks Cheatsheet (stanford.edu)

[18] Bài viết “Stock Market Analysis”, Fares Sayah. Truy cập lần cuối: 24/6/2022. Link: Stock Market Analysis + Prediction using LSTM | Kaggle

[19] Bài viết “Time-series forecasting: Predicting stock prices using an LSTM model”, Serafeim Loukas, đăng ngày: 10/7/2020. Truy cập lần cuối: 24/6/2022. Link: Time-Series Forecasting: Predicting Stock Prices Using An LSTM Model | by Serafeim Loukas | Towards Data Science

[20] Bài viết “RMSE là gì – Mean Squared Error”, đăng ngày 4/2/2021. Truy cập lần cuối: 24/6/2022. Link: Rmse Là Gì - Mean Squared Error - Thienmaonline

[21] Bài viết “Các phương pháp đánh giá mô hình học máy, học sâu”, Rabiloo, đăng ngày: 3/12/2021. Truy cập lần cuối: 24/6/2022. Link: Các phương pháp đánh giá mô hình học máy, học sâu (Machine learning & Deep learning) (rabiloo.com)