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	API	Application Programming Interface
3	CRM	Customer Relationship Management
4	DNA	Deoxyribonucleic Acid
5	GDP	Gross Domestic Product
6	GPU	Graphics Processing Unit
7	IoT	Internet of Things
8	LSTM	Long-short Term Memory
9	ML	Machine Learning
10	MLE	Maximum Likelihood Estimation
11	RNN	Recurrent Neuron Network
12	RMSE	Root Mean Squared Error

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 thuật toán 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 thuật toán 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

- 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.

1.1.2 Thị trường chứng khoán

- 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.

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

- Những biến động trong thị trường chứng khoán luôn có nhiều nguyên nhân phức tạp và 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 được ứng dụng vào dự đoán bằng việc phân tích và tận dụng tối đa lượng dữ liệu lịch sử kết hợp cùng với kiến thức về tài chính 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: 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 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 của con người. Học máy có khả năng 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, điều mà con người khó có thể thực hiện được vì trí nhớ có hạn của mình, là trợ thủ đắc lực giúp nhà đầu tư đưa ra quyết định.

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. 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. 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 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. Sklearn 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 Pandas. 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 Sơ lược về Machine Learning

1.4.1 Giới thiệu

- Học máy là một lĩnh vực của trí tuệ nhân tạo. 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. [3]
- Việc nghiên cứu 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 về những bộ phim nên xem, quảng cáo sản phẩm người dùng muốn mua, đề xuất các sản phẩm liên quan đến sản phẩm người dùng đã mua, hoặc muốn mua để người dùng đưa ra những so sánh rồi lựa chọn, 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. [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.
- 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.

1.5 Thuật toán học máy Logistic Regression

Trong Machine learning thì Logistic Regression là thuộc thuật toán 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. [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

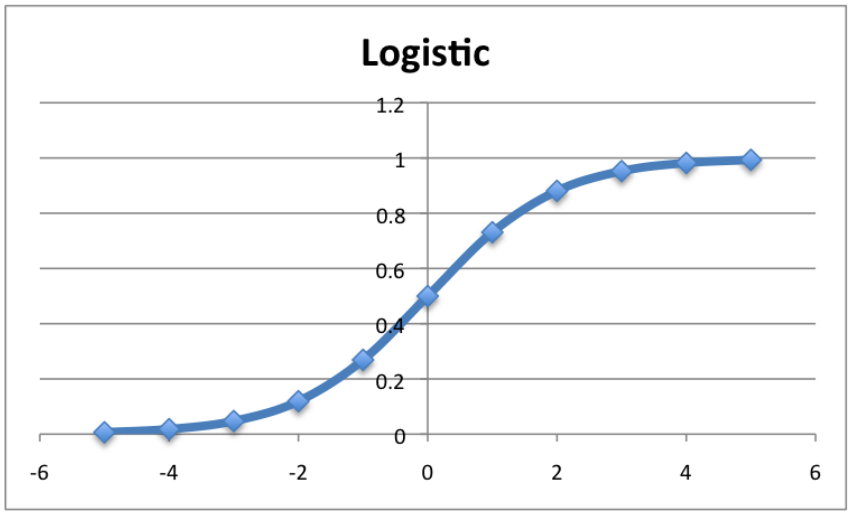
- Đăng nhập hỗn hợp
- Probit
- Probit đa thức
- Đăng nhập có thứ tự

1.5.2 Hàm Logistic [2]

- Hồi quy Logistic được đặt tên cho hàm được sử dụng cốt lõi của phương pháp, hàm logistic. Hay còn được gọi là hàm Sigmoid hoặc đường cong Sigmoid. Đó là một đường cong hình chữ S có thể lấy bất kỳ số nào có giá trị thực và ánh xạ nó thành một giá trị từ 0 đến 1, nhưng không bao giờ chính xác ở giới hạn 0 và 1.
- Hàm có công thức như sau:

$$f(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} \qquad (1.1)$$

- Trong đó: e là cơ số logarit tự nhiên, x là giá trị thực tế mà ta muốn ánh xạ.
- Ví dụ các số từ -5 đến 5 được chuyển thành phạm vi 0 và 1 bằng cách sử dụng hàm Logistic:



Hình 1.1 Minh họa hàm Logistic

- Hồi quy Logistic sử dụng một phương trình làm đại diện. Giá trị đầu vào x được kết hợp tuyến tính bằng cách sử dụng trọng số hoặc giá trị hệ số để dự đoán đầu ra y. Giá trị đầu ra được mô hình hóa là giá trị nhị phân (0 hoặc 1) chứ không phải là các giá trị số.
- Ví dụ:

$$y = \frac{e^{B0+B1*x}}{e^{B0+B1*x}+1} \qquad (1.2)$$

- Trong đó: y là đầu ra được dự đoán, B0 là số hạng thiên vị hoặc giới hạn chặn và B1 là hệ số cho giá trị đầu vào x. Một cột đầu vào có một hệ số B được liên kết (giá trị thực không đổi) phải được học từ dữ liệu huấn luyện. Biểu diễn thực tế của mô hình được lưu trữ trong bộ nhớ hoặc tệp là các hệ số trong phương trình.

1.5.3 Xác suất dự đoán của hồi quy Logistic [2]

- Mô hình xác suất hồi quy Logistic của lớp mặc định (ví dụ: lớp đầu tiên). Ví dụ dự đoán giới tính của mọi người thông qua chiều cao của họ, thì lớp đầu tiên có thể là nam và mô hình hồi quy Logistic có thể được viết dưới dạng xác suất của nam so với chiều cao của một người hoặc nhiều hơn.

$$P(\text{gender} = \text{male} \mid \text{height}) \qquad (1.3)$$

- Nếu đặt gender là Y, height là X. X thuộc lớp mặc định Y = 1, ta có mô hình xác suất hồi quy được viết lại như sau:

$$P(X) = P(Y = 1 \mid X) \qquad (1.4)$$

- Các dự đoán được biến đổi bằng cách sử dụng hàm Logistic. Hệ quả của việc dự đoán bằng hàm Logistic là các dự đoán không là sự kết hợp tuyến tính của các đầu vào, ví dụ tiếp tục ở trên, mô hình có thể được biểu diễn như sau:

$$p(X) = \frac{e^{B0+B1*X}}{e^{B0+B1*X}+1} \qquad (1.5)$$

- Biến đổi phương trình (1.5), ta được:

$$\frac{p(X)}{1-p(X)} = e^{B0+B1*X} \qquad (1.6)$$

- Lấy logarit tự nhiên cả hai vế, ta được:

$$\ln\left(\frac{p(X)}{1-p(X)}\right) = B0 + B1 * X \qquad (1.7)$$

- Ta nhận thấy ở vế trái phương trình (1.7), $\frac{p(X)}{1-p(X)}$ là tỉ lệ giữa xác xuất xảy ra chia cho xác suất không xảy ra. Tỉ lệ đó được gọi là odds. Logarit tự nhiên của odds là một hàm tuyến tính của các tham số B và các biến X. Ta có thể viết lại phương trình trên như sau:

$$\ln(\text{odds}) = B0 + B1*X \qquad (1.8)$$

- Vì odds được biến đổi theo logarit, ta có thể dùng chức năng liên kết logit để di chuyển số mũ e về bên phải và viết thành:

$$\text{odds} = e^{B0+B1*X} \qquad (1.9)$$

- Sau các bước biến đổi trên ta nhận thấy mô hình xác suất hồi quy Logistic vẫn là một tổ hợp tuyến tính của các đầu vào, nhưng sự kết hợp tuyến tính này liên quan đến log-odds của lớp mặc định.

1.5.4 Model của hồi quy Logistic [2]

- Các hệ số của thuật toán hồi quy Logistic phải được ước tính từ dữ liệu training. 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.
- Các hệ số tốt nhất sẽ dẫn đến một mô hình dự đoán giá trị rất gần với 1 (tiếp tục ví dụ ở 1.5.3 là: nam) cho lớp mặc định và giá trị rất gần với 0 (nữ) cho lớp khác.

1.5.5 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.
- 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 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) thông qua log-odds. Trong thế giới thực, rất khó có khả năng các quan sát được phân tách tuyến tính. Vì vậy, khi dữ liệu có thể phân tách tuyến tính là giả định cho hồi quy logistic
- 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 Sơ lược về 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.

- Là một xu hướng có tiềm năng phát triển mạnh 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 và còn có những ứng dụng thực tiễn vào trong đời sống.

- 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,…

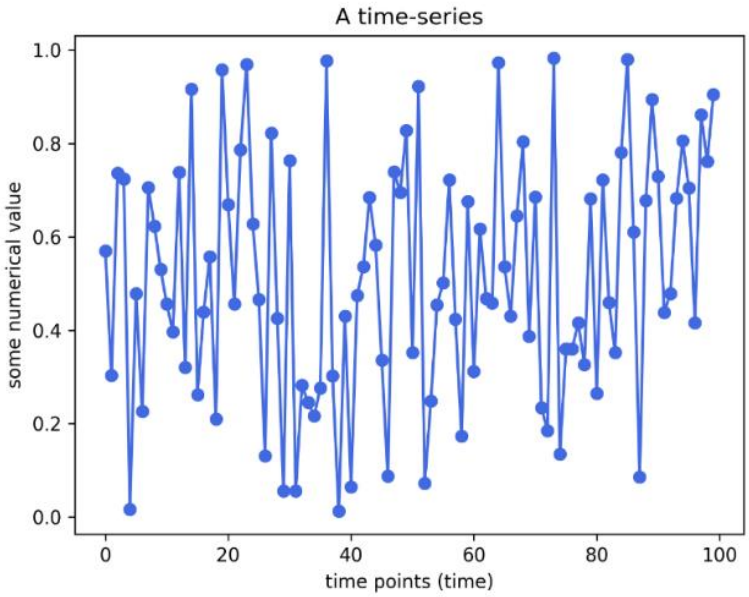
Hình 1.2 Ví dụ ứng dụng của Deep learning

- 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 vưc 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.

1.7 Dữ liệu chuỗi thời gian (Time series data)

- Dữ liệu về chứng khoán được ghi nhận theo ngày và kéo dài liên tục cho tới khi chúng bị loại bỏ khỏi sàn giao dịch. Do vậy mà chúng được xếp vào dạng dữ liệu là 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 dạng 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 1.3 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. 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. [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. 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. Một ví dụ:

1.8 Mạng LSTM

1.8.1 Giới thiệu [15]

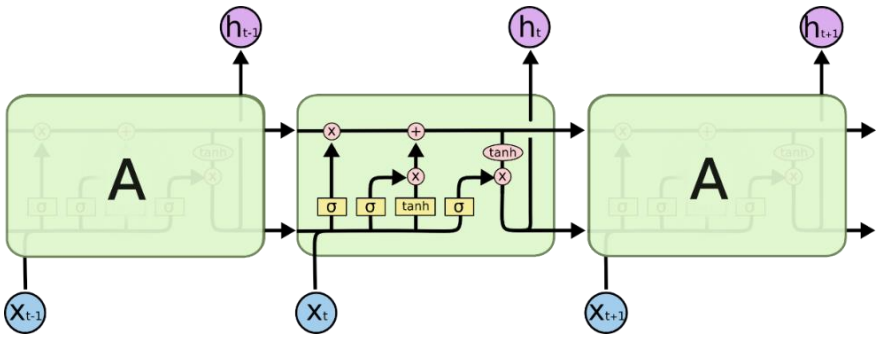
- Mạng LSTM là một loại mạng nơ-ron tuần hoàn. Trong RNN, đầu ra từ bước cuối cùng được cung cấp dưới dạng đầu vào trong bước hiện tại. LSTM được thiết kế bởi Hochreiter & Schmidhuber. Nó giải quyết vấn đề phụ thuộc dài hạn của RNN. Theo mặc định, LSTM có thể giữ lại thông tin trong một khoảng thời gian 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.

1.8.2 Kiến trúc [15]

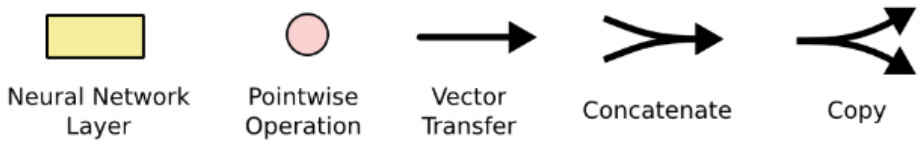
- Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một lớp tanh.

- 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 lớp mạng nơ-ron, chúng có tới 4 lớp tương tác với nhau một cách rất đặc biệt.



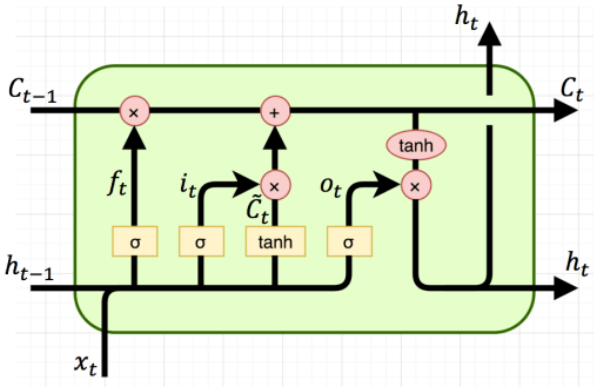
Hình 1.4 Mạng LSTM

- Các ký hiệu ở hình trên được giải thích như sau:



Hình 1.5 Giải thích các ký hiệu ở hình 1.4

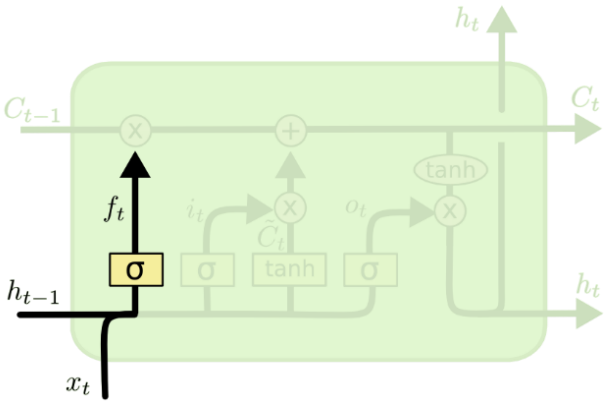
- Cụ thể ở mỗi state thứ t của LSTM:



Hình 1.6 Một state của mạng LSTM

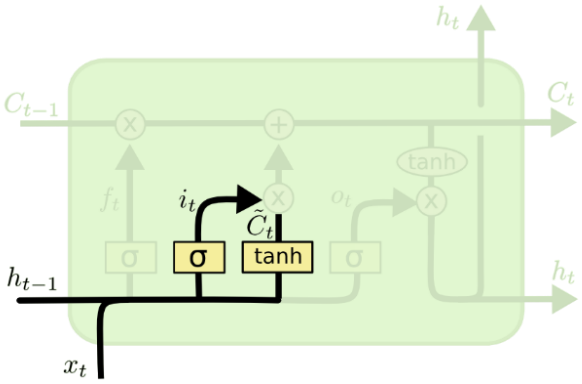
- Output: c_t, h_t, ta gọi c là cell state, h là hidden state

- 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. Tanh có dạng: tanh x = $\frac{e^x - e^{-x}}{e^x + e^{-x}}$ (1.10). 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: $f_t = \sigma(U_f * x_t + W_t * h_{t-1} + b_f)$ (1.11). Cổng này quyết định lượng thông tin từ state trước bị bỏ đi là bao nhiêu.



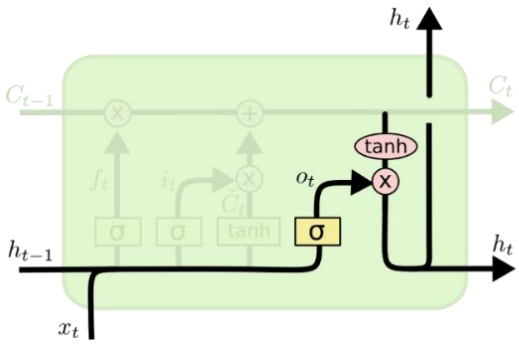
Hình 1.7 Cổng Forget

    - Input gate: $i_t = \sigma(U_i * x_t + W_i * h_{t-1} + b_i)$ (1.12). Cổng này quyết định lượng thông tin đầu vào ảnh hưởng đến state mới là bao nhiêu.



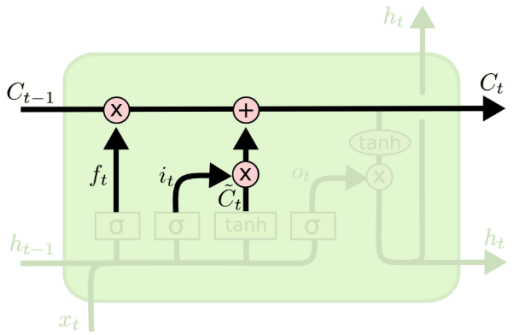
Hình 1.8 Cổng Input

    - Output gate: $o_t = \sigma(U_o * x_t + W_o * h_{t-1} + b_o)$ (1.13). Cổng này điều chỉnh lượng thông tin có thể ra ngoài y_t và lượng thông tin tới state tiếp theo.



Hình 1.9 Cổng Output

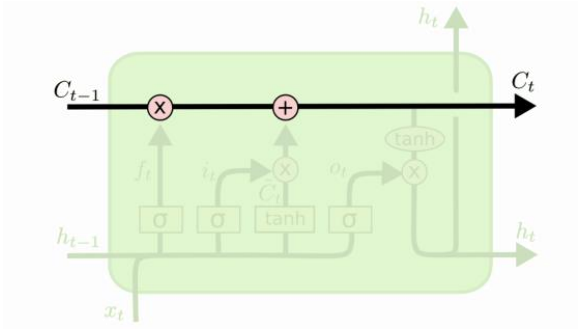
- Nhận xét $0 < f_t, i_t, o_t < 1$ (giá trị của hàm sigmoid nằm trong khoảng [0;1]), b_f, b_i, b_o là hệ số bias, W, U giống với RNN.
- $\tilde{c_t} = \tanh(U_c * x_t + W_c * h_{t-1} + b_c)$ (1.14), giống tính $a^{<t>}$ trong RNN.
- $c_t = f_t * c_{t-1} + i_t * \tilde{c_t}$ (1.15)
- $h_t = o_t + \tanh(c_t)$ (1.16), ngoài ra h_t cũng được dùng để tính ra ouput



Hình 1.10 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 1.11 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.

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.
- Để dự đoán dựa vào tỉ suất lợi nhuận. Ta dựa vào giá đóng cửa của các ngày phía trước ở cột Close tính tỉ suất lợi nhuận giữa các ngày với nhau, rồi dùng làm input cho bài toán.
- Công thức tính tỉ suất lợi nhuận dựa trên giá đóng cửa:

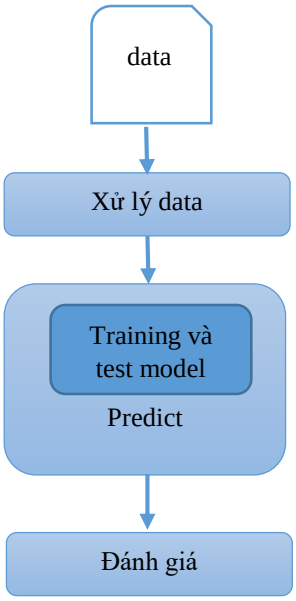
$$r_t = \frac{p_t - p_{t-1}}{p_{t-1}} \qquad (2.1)$$

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 là 36.568, ngày 29 có giá đóng cửa là 35.046. Áp dụng công thức ta tính được tỉ suất lợi nhuận ta thu được 0.0434286366. 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 mảng 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 mảng tên Direction để gắn nhãn cho dữ liệu là tăng hoặc giảm dựa vào tỉ suất lợi nhuận vừa tính được ở trên. Nếu tỉ suất là dương thì gắn là 1, ngược lại tỉ suất là số âm thì gắn 0. Mục đính là để sử dụng cho training và testing model.

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.
- Mô hình tổng quát được đề xuất để dự đoán:



Hình 2.2 Mô hình đề xuất

- Với thuật toán Logistic Regression:
  - Xác định đầu vào là tỉ suất lợi nhuận được tính giữa các ngày. Đặt X là biến đầu vào (tỉ suất lợi nhuận), và Y là biến đầu ra (chiều tăng giảm). Ta có: X = (x_1, x_2,…,x_n), Y ϵ {Up, Down}.
  - Bài toán sẽ dự báo Y thuộc lớp Up, với đầu vào x_0, nếu Pr(y = Up | X = x_0) > 0.5; và ngược lại với lớp Down
  - Biến phụ thuộc Y 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) = P(Y = Up|X).
  - Mô hình hồi quy Logistic có dạng:
    - $Ln\left(\frac{p(X)}{1-p(X)}\right) = B0 + B1X_1 + \cdots + BnX_n$ (2.2)
    - Với B0, B1,…,Bn là các hệ số cần ước lượng, ở bài toán này là các giá trị được dự đoán bằng hàm predict().
  - Chia tập dữ liệu thành training và testing
  - Huấn luyện mô hình 2 lần với 2 số lượng biến đầu vào khác nhau
  - Thực hiện dự đoán trên tập testing
  - Đánh giá độ chính xác của dự đoán
- Với mạng 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]

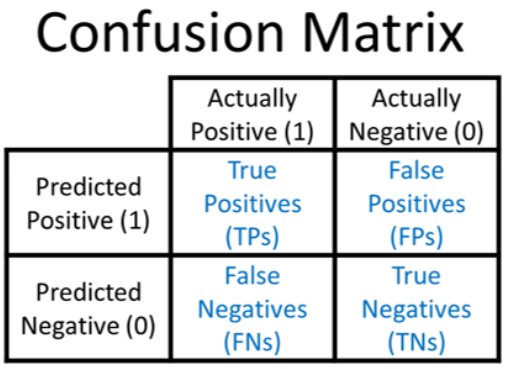
- Xây dựng model với các layer input và output
- Thực hiện huấn luyện và test mô hình
- Trực quan hóa kết quả test
- Đánh giá mô hình

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 2.3 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

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

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

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
```

- 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

- Tính logarit tự nhiên của các giá trị vừa tính trên để ánh xạ các giá trị từ 0 đến 1, kết quả:

```
0          NaN
1    -0.003604
2    -0.017265
3     0.015927
4     0.012773
        ...
2009  0.056020
2010  0.024935
2011 -0.005013
2012 -0.002097
2013 -0.014700
```

Hình 3.3 Kết quả khi thực hiện ánh xạ các giá trị

- Tạo một cột để lưu trữ giá trị vừa tính là cột "Return". Thực hiện gán nhã cho cột Return và lưu vào cột "Direction"
- Đặt "Lag" là tên cột lưu trữ. Xây dựng hàm lagit để dịch chuyển chỉ mục ở cột "Return" và lưu vào cột "Lag" bằng hàm shift(). Dịch chuyển tương tự với cột "Direction" và lưu vào cột "Dr Lag". Ở thực nghiệm 1 dùng 5 biến Lag để training và testing.
- Sử dụng hàm dropna() để loại bỏ các dòng có dữ liệu trống.
- Kết quả:

Return	...	Lag 2	Dr Lag 2	Lag 3	Dr Lag 3	Lag 4	Dr Lag 4	Lag 5	Dr Lag 5
-0.012345	...	0.012773	1	0.015927	1	-0.017265	0	-0.003604	0
-0.044770	...	-0.025101	0	0.012773	1	0.015927	1	-0.017265	0
0.146163	...	-0.012345	0	-0.025101	0	0.012773	1	0.015927	1
0.017579	...	-0.044770	0	-0.012345	0	-0.025101	0	0.012773	1
0.040829	...	0.146163	1	-0.044770	0	-0.012345	0	-0.025101	0
...	...	...	...	...	...	...	...	...	...
0.056020	...	0.041987	1	-0.035616	0	0.006077	1	-0.051585	0
0.024935	...	0.072271	1	0.041987	1	-0.035616	0	0.006077	1
-0.005013	...	0.056020	1	0.072271	1	0.041987	1	-0.035616	0
-0.002097	...	0.024935	1	0.056020	1	0.072271	1	0.041987	1
-0.014700	...	-0.005013	0	0.024935	1	0.056020	1	0.072271	1

Hình 3.4. Tỉ suất lợi nhuận được dịch chuyển chỉ mục 5 lần

- Phân chia tập dữ liệu để tiến hành training và testing:

```
from sklearn.model_selection import train_test_split
train, test = train_test_split(data, shuffle=False,
test_size=0.37, random_state=0)
train = train.copy()
test = test.copy()
```

- Training và test model:

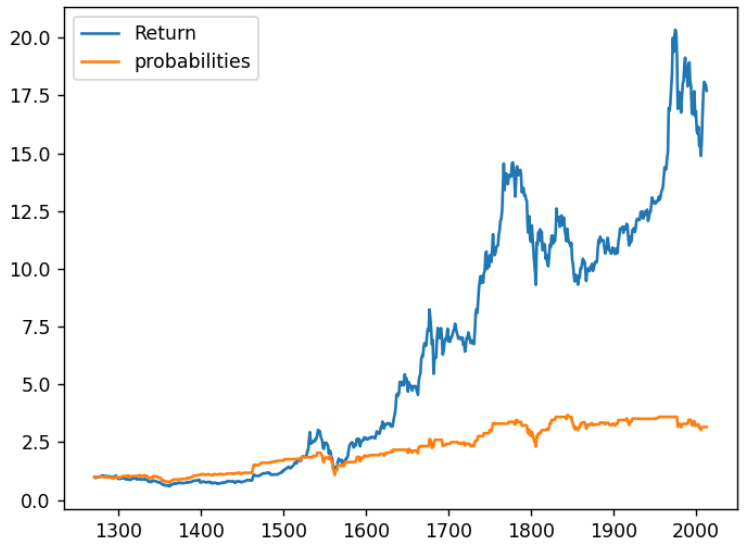
```
model = LogisticRegression()
model.fit(train[dirname], train['Direction'])
test['prediction_Logit'] = model.predict(test[dirname])
test['probabilities'] = test['prediction_Logit'] * test['Return']
print(np.exp(test[['Return','probabilities']].sum()))
```

- Kết quả:

```
predict testing:
  1271    1
1272    0
1273    1
1274    0
1275    1
       ..
2009    0
2010    0
2011    0
2012    0
2013    0
Name: prediction_Logit, Length: 743, dtype: int64
Return           17.705617
probabilities     3.154726
```

Hình 3.5 Kết quả dự đoán thực nghiệm 1

- Mô hình hóa kết quả ta được:



Hình 3.6 Mô hình hóa kết quả thực nghiệm 2

- Đánh giá thực nghiệm dựa trên ma trận hỗn loạn:

```
Confusion matrix:
 [[189 149]
 [227 178]]
Ket qua danh gia:
              precision    recall  f1-score   support

           0       0.45      0.56      0.50       338
           1       0.54      0.44      0.49       405

    accuracy                           0.49       743
```

Hình 3.7 Kết quả đánh giá thực nghiệm 1

Nhìn vào hình 3.7 ta thấy giá trị Accuracy là 0.49, một kết quả khá thấp. Cho thấy độ chính xác của dự đoán trong thực nghiệm này khá thấp.

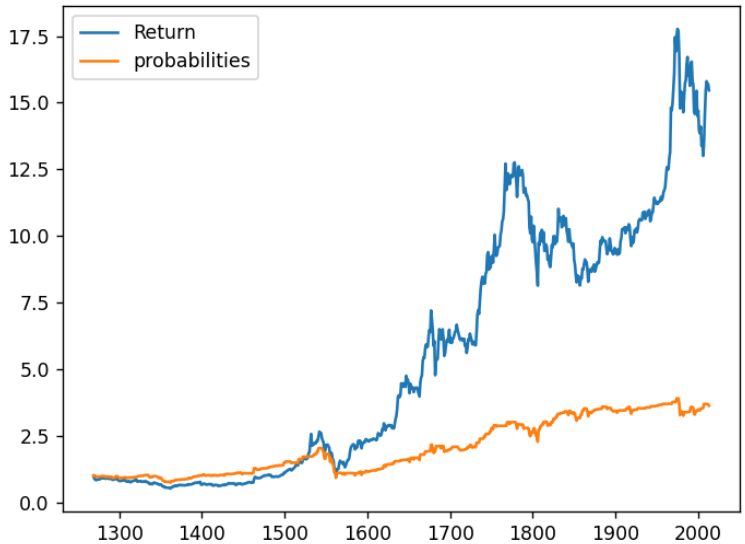
➢ **Thực nghiệm 2:**

- Ở thực nghiệm này ta dùng 1 biến đầu vào là Lag 1, tiến hành training và testing ta được:

```
predict testing:
  1269    0
1270    0
1271    1
1272    1
1273    1
       ..
2009    0
2010    0
2011    0
2012    1
2013    1
Name: prediction_Logit, Length: 745, dtype: int64
Return           15.465106
probabilities     3.626742
```

Hình 3.8 Kết quả dự đoán thực nghiệm 2

- Mô hình hóa kết quả thực nghiệm:



Hình 3.9 Mô hình hóa thực nghiệm 2

- Đánh giá kết quả thực nghiệm:

```
Confusion matrix:
 [[194 145]
 [213 193]]
Ket qua danh gia:
              precision    recall  f1-score   support

           0       0.48      0.57      0.52       339
           1       0.57      0.48      0.52       406

    accuracy                           0.52       745
```

Hình 3.10 Kết quả đánh giá thực nghiệm 2

Nhìn hình 3.10 ta nhận thấy giá trị Accuracy là 0.52, cho kết quả cao hơn thực nghiệm 1. Độ tin cậy theo đó cũng được nâng lên. Nhưng trên thực tế vẫn là một kết quả khá thấp.

- **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

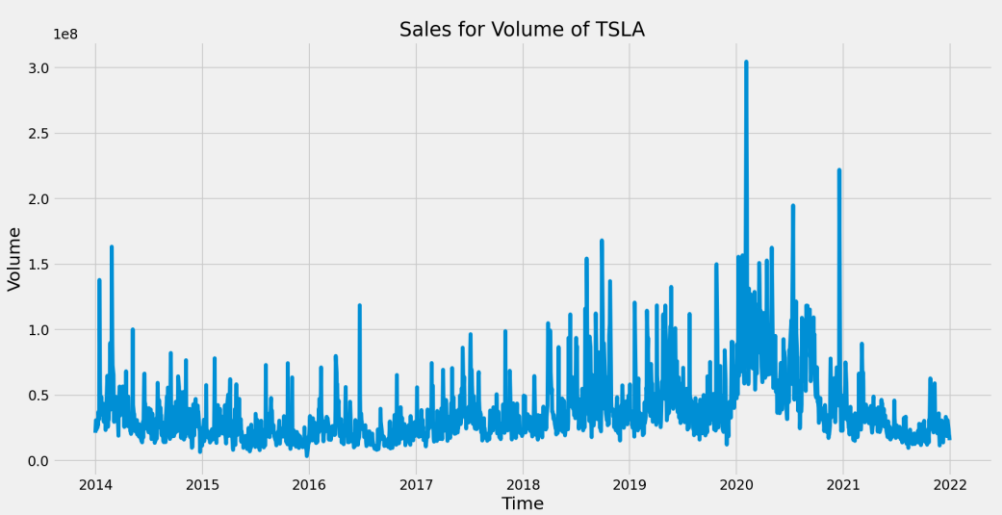
[2015 rows x 6 columns]

Hình 3.11. Dữ liệu thực nghiệm mô hình LSTM

- Mô hình hóa cột “Close”:

Hình 3.12. Biểu đồ cột Close price

- Mô hình hóa cột "Volume"



Hình 3.13. 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 3.14. 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 3.15. 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:

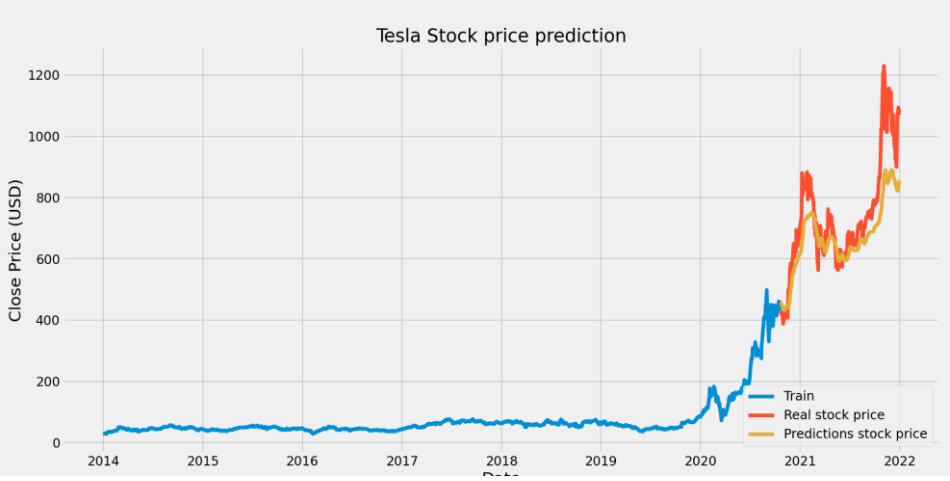
```
# Build the LSTM model
model = Sequential()
# 1st layer
model.add(LSTM(50, return_sequences=True,
input_shape=(x_train.shape[1], 1))) # 50 neuron
model.add(Dropout(0.2))
# 2nd layer
model.add(LSTM(50, return_sequences=True))
model.add(Dropout(0.2))
# 3rd layer
model.add(LSTM(50, return_sequences= 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 3.16. 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 3.17. 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 3.18 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

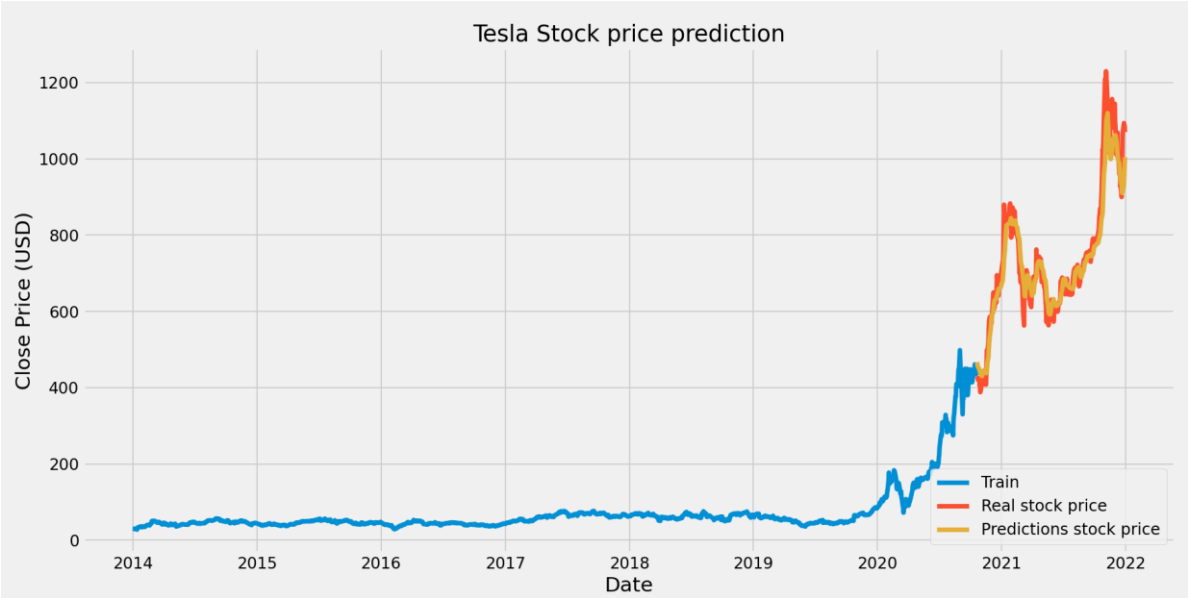
```
    # 1st layer
model.add(LSTM(128, return_sequences=True,
input_shape=(x_train.shape[1], 1))) # 50 neuron
    # 2nd layer
model.add(LSTM(64, return_sequences=False))
   # output layer
model.add(Dense(25)) #25 noron
model.add(Dense(1))
```

  - 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 3.19 Quá trình học trên model thực nghiệm 2

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



Hình 3.20. 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 3.21 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] Trang wiki: bài "Học sâu". Truy cập lần cuối: 20/6/2022. Link: Học sâu – Wikipedia tiếng Việt

[10] 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)

[11] 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 "Stock Market Analysis", Fares Sayah. Truy cập lần cuối: 24/6/2022. Link: Stock Market Analysis + Prediction using LSTM | Kaggle

[17] 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

[18] 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

[19] 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)