

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO BẢO VỆ PROJECT

Môn học: Nhập môn Khoa học dữ liệu

Đề tài: HỆ THỐNG CRAWL DỮ LIỆU TÀI CHÍNH TÍCH HỢP PREDICTION & FORECAST

Giảng viên hướng dẫn: PGS. TS. Phạm Văn Hải

Sinh viên thực hiện:	Nguyễn Đình Thành An	20204936
	Đặng Tiến Dũng	20215011
	Trần An Khang	20210463
	Huỳnh Khắc Anh Khoa	20198183

Hà Nội, 6/2025

MỤC LỤC

LỜI NÓI ĐẦU.....	5
CHƯƠNG 1. GIỚI THIỆU BÀI TOÁN.....	7
1.1 Mô tả bài toán.....	7
1.2 Tìm hiểu về chứng khoán	7
1.2.1 Thị trường chứng khoán	7
1.2.2 Các Yếu Tố Ảnh Hưởng Đến Giá Chứng Khoán	7
1.2.3 Vai Trò Của Công Nghệ Trong Đầu Tư Chứng Khoán	8
CHƯƠNG 2. THU THẬP DỮ LIỆU	9
2.1 Mô tả dữ liệu.....	9
2.1.1 Dữ liệu công ty niêm yết	9
2.1.2 Dữ liệu lịch sử giao dịch chứng khoán	10
2.1.3 Bản tin công ty.....	11
2.2 Phương Pháp Thu Thập	12
2.2.1 Dữ liệu công ty và dữ liệu lịch sử giao dịch chứng khoán	12
2.2.2 Dữ liệu bản tin của công ty:.....	12
CHƯƠNG 3. TIỀN XỬ LÝ DỮ LIỆU	14
3.1 Làm sạch dữ liệu.....	14
3.1.1 Xử lý dữ liệu thiếu.....	14
3.1.2 Loại bỏ nhiễu	14
3.2 Biến đổi dữ liệu	14
3.2.1 Tính toán % thay đổi	14
3.2.2 Tính toán chỉ số Bollinger Bands	15
3.3 Trực quan hóa dữ liệu.....	15
1. Close – open	15
2. BB.....	16
3. MACD (Moving Average Convergence Divergence)	16
4. RSI (Relative Strength Indicator):	16
5. SMA (Simple Moving Average):	17

6. EMA (Exponential Moving Average).....	17
CHƯƠNG 4. XÂY DỰNG MÔ HÌNH DỰ ĐOÁN.....	18
4.1 Linear Regression	18
4.2 K-Neighbors	18
4.3 Random Forest.....	19
4.4 LSTM	20
4.4.1 Vanishing gradient	20
4.4.2 Dữ liệu chuỗi thời gian (Time series)	20
4.4.3 Cấu trúc LSTM.....	21
4.5 Prophet.....	22
CHƯƠNG 5. ĐÁNH GIÁ MÔ HÌNH.....	24
5.1 R^2 score.....	24
5.2 MSE.....	25
5.3 MAE	26
CHƯƠNG 6. LỰA CHỌN MÔ HÌNH ĐỀ XUẤT	27
6.1 Chuẩn bị dữ liệu	27
6.2 Chuẩn hóa dữ liệu.....	28
6.3 Chia tập huấn luyện và kiểm tra	28
6.4 Tạo tập dữ liệu	28
6.5 Định nghĩa mô hình	29
6.6 Huấn luyện mô hình	30
6.7 Dự đoán và đánh giá mô hình.....	31
6.8 Giá trị trả về.....	31
6.9 Các case study nổi bật khi nhóm rút ra trong quá trình triển khai mô hình LSTM	32
6.9.1 Ảnh hưởng của tỷ lệ chia tập huấn luyện và kiểm tra	32
6.9.2 Ảnh hưởng của chỉ số Look-Back	33
6.9.3 Ảnh hưởng của số epoch huấn luyện tối đa (num_epochs) và epoch kiên nhẫn (patience) đối với hiệu suất của mô hình LSTM.	34
CHƯƠNG 7. KẾT QUẢ MÔ HÌNH	35
CHƯƠNG 8. TRIỂN KHAI ỨNG DỤNG.....	37

7.1 Công nghệ sử dụng	37
7.2 Chức năng.....	37
7.2.1 Lựa chọn tra cứu dữ liệu theo ngày và mã cổ phiếu công ty.....	37
7.2.2 Trực quan hóa dữ liệu	38
7.2.3 Hiện thị dữ liệu danh sách công ty và lịch sử cổ phiếu	40
7.2.4 Chức năng dự báo giá cổ phiếu	41
7.2.5 Extension hiện thị tin tức về mã cổ phiếu đã thu thập được.....	42
CHƯƠNG 9. KẾT LUẬN	43
8.1 Kết luận	43
8.2 Hướng phát triển trong tương lai	43

LỜI NÓI ĐẦU

Trong bối cảnh nền kinh tế toàn cầu ngày càng biến động và phức tạp, việc thu thập, xử lý và dự báo dữ liệu tài chính trở thành yếu tố then chốt đối với các nhà đầu tư, doanh nghiệp và tổ chức tài chính. Hệ thống crawl dữ liệu tài chính tích hợp dự đoán và dự báo giá chứng khoán không chỉ hỗ trợ đưa ra các quyết định đầu tư chính xác mà còn giúp giảm thiểu rủi ro và tối ưu hóa lợi nhuận thông qua việc cung cấp thông tin kịp thời và dự báo đáng tin cậy.

Dự đoán giá chứng khoán là một bài toán đầy thách thức do chịu ảnh hưởng từ nhiều yếu tố như kinh tế, chính trị, xã hội và tâm lý thị trường. Các phương pháp truyền thống như phân tích kỹ thuật và cơ bản, dù mang lại một số kết quả tích cực, vẫn còn nhiều hạn chế trong việc xử lý khối lượng dữ liệu lớn và các biến động phức tạp. Sự phát triển của công nghệ thông tin, trí tuệ nhân tạo và các kỹ thuật học máy đã mở ra cơ hội xây dựng các hệ thống tiên tiến, kết hợp thu thập dữ liệu thời gian thực với dự đoán chính xác hơn.

Báo cáo này trình bày một hệ thống crawl dữ liệu tài chính tích hợp khả năng dự đoán và dự báo giá chứng khoán, sử dụng các phương pháp học máy và trí tuệ nhân tạo. Hệ thống tập trung vào việc thu thập dữ liệu từ các nguồn đáng tin cậy, xử lý dữ liệu thô và áp dụng các mô hình như hồi quy tuyến tính, mạng nơ-ron sâu (LSTM), và Prophet để dự báo xu hướng giá. Chúng tôi sẽ giới thiệu quy trình thu thập dữ liệu tự động, các thuật toán dự đoán chính, và các kỹ thuật đánh giá hiệu quả mô hình như R^2 , MAE và MSE.

Mục tiêu của bài tập lớn này là phát triển một hệ thống toàn diện, kết hợp thu thập dữ liệu tài chính với dự đoán và dự báo giá chứng khoán, đạt độ chính xác cao và khả năng ứng dụng thực tiễn. Kết quả nghiên cứu không chỉ mang ý nghĩa lý thuyết, làm rõ tiềm năng của trí tuệ nhân tạo trong tài chính, mà còn cung cấp công cụ hỗ trợ thiết thực cho các nhà đầu tư và các bên liên quan trong việc ra quyết định chiến lược.

PHÂN CÔNG VÀ ĐÁNH GIÁ

MSSV	Họ và tên	Nhiệm vụ	Phần trăm đánh giá
20204936	Nguyễn Đình Thành An	Xây dựng các mô hình dự đoán.	25%
20210463	Trần An Khang	Xây dựng các mô hình dự đoán.	25%
20215011	Đặng Tiến Dũng	Thu thập và tiền xử lý, trực quan hóa dữ liệu.	25%
20198183	Huỳnh Khắc Anh Khoa	Xây dựng giao diện streamlit.	25%

CHƯƠNG 1. GIỚI THIỆU BÀI TOÁN

1.1 Mô tả bài toán

Xây dựng một mô hình tự động crawl dữ liệu tài chính tích hợp forecast & prediction, nhằm cung cấp những dự đoán chính xác và kịp thời, kết hợp với các bản tin tài chính hỗ trợ người dùng trong việc ra quyết định đầu tư.

1.2 Tìm hiểu về chứng khoán

Dữ liệu tài chính là chỉ số có thể đại diện cho giao dịch trên thị trường tài chính. Chứng khoán bao gồm các loại tài sản tài chính như cổ phiếu, trái phiếu, chứng chỉ quỹ và các công cụ phái sinh. Dữ liệu tài chính đóng vai trò quan trọng trong nền kinh tế, giúp huy động vốn cho doanh nghiệp, tạo cơ hội đầu tư cho các nhà đầu tư, và góp phần vào sự phát triển của thị trường tài chính.

1.2.1 Thị trường chứng khoán

Thị trường chứng khoán là nơi diễn ra các hoạt động mua bán chứng khoán giữa các nhà đầu tư. Thị trường chứng khoán có thể được phân thành thị trường sơ cấp và thị trường thứ cấp.

- Thị trường sơ cấp là nơi các chứng khoán mới được phát hành và bán lần đầu tiên cho nhà đầu tư. Đây là kênh huy động vốn trực tiếp cho các tổ chức phát hành như công ty cổ phần, chính phủ. Trong thị trường sơ cấp, quá trình phát hành chứng khoán có thể thông qua đấu giá hoặc bảo lãnh phát hành.
- Thị trường thứ cấp là nơi các chứng khoán đã phát hành được mua bán lại giữa các nhà đầu tư. Thị trường thứ cấp giúp tăng tính thanh khoản cho chứng khoán và tạo cơ hội cho nhà đầu tư mua bán chứng khoán theo nhu cầu. Các giao dịch trên thị trường thứ cấp thường diễn ra trên các sàn giao dịch chứng khoán hoặc thị trường phi tập trung.

1.2.2 Các Yếu Tố Ảnh Hưởng Đến Giá Chứng Khoán

Giá chứng khoán chịu ảnh hưởng của nhiều yếu tố kinh tế, tài chính, và tâm lý thị trường. Các yếu tố chính bao gồm:

1.2.2.1. Yếu Tố Kinh Tế

- Tăng trưởng kinh tế: Sự phát triển của nền kinh tế thường kéo theo sự tăng trưởng của các công ty, làm tăng giá trị cổ phiếu.
- Lãi suất: Lãi suất ảnh hưởng trực tiếp đến chi phí vay vốn và lợi suất kỳ vọng của nhà đầu tư. Lãi suất thấp thường làm tăng giá cổ phiếu và ngược lại.
- Lạm phát: Lạm phát cao làm giảm giá trị tiền tệ, ảnh hưởng đến lợi nhuận thực tế của doanh nghiệp và giá cổ phiếu.

1.2.2.2. Yếu Tố Tài Chính

- Hiệu quả kinh doanh: Kết quả kinh doanh của doanh nghiệp, bao gồm doanh thu, lợi nhuận và cổ tức, ảnh hưởng trực tiếp đến giá cổ phiếu.
- Chính sách tài chính: Các quyết định tài chính như phát hành thêm cổ phiếu, chia cổ tức, mua lại cổ phiếu cũng ảnh hưởng đến dữ liệu tài chính.

1.2.2.3. Yếu Tố Tâm Lý Thị Trường

- Tin tức và sự kiện: Các tin tức kinh tế, chính trị, xã hội và các sự kiện bất ngờ có thể tạo ra biến động lớn trên thị trường chứng khoán.
- Tâm lý đám đông: Tâm lý và hành vi của các nhà đầu tư, bao gồm sự lạc quan hoặc bi quan quá mức, có thể dẫn đến những biến động không dự đoán được của giá chứng khoán.

1.2.3 Vai Trò Của Công Nghệ Trong Đầu Tư Chứng Khoán

Công nghệ đã và đang thay đổi cách thức hoạt động của thị trường chứng khoán, mang lại nhiều lợi ích cho các nhà đầu tư và doanh nghiệp.

1.2.3.4. Giao Dịch Trực Tuyến

Giao dịch chứng khoán trực tuyến giúp nhà đầu tư dễ dàng thực hiện các giao dịch mua bán chứng khoán một cách nhanh chóng và tiện lợi. Các nền tảng giao dịch trực tuyến cung cấp nhiều công cụ hỗ trợ như biểu đồ, phân tích kỹ thuật, và thông tin thị trường.

1.2.3.5. Phân Tích Dữ Liệu

Công nghệ phân tích dữ liệu lớn (Big Data) và trí tuệ nhân tạo (AI) giúp các nhà đầu tư phân tích khối lượng lớn thông tin tài chính, nhận diện các xu hướng và mô hình trong dữ liệu, từ đó đưa ra các quyết định đầu tư chính xác hơn.

1.2.3.6. Dự Đoán Giá Chứng Khoán

Các mô hình học máy (Machine Learning) và học sâu (Deep Learning) được áp dụng để dự đoán giá chứng khoán. Các thuật toán này có khả năng học từ dữ liệu lịch sử và đưa ra dự đoán về giá cổ phiếu trong tương lai, hỗ trợ nhà đầu tư trong việc ra quyết định.

CHƯƠNG 2. THU THẬP DỮ LIỆU

2.1 Mô tả dữ liệu

Dữ liệu thu thập được gồm danh sách công ty, lịch sử giao dịch chứng khoán, và các bản tin tài chính của công ty.

2.1.1 Dữ liệu công ty niêm yết

	ticker	organName	organTypeCode	comGroupCode
0	A32	CTCP 32	1	UPCOM
1	AAA	CTCP Nhựa An Phát Xanh	1	HOSE
2	AAH	CTCP Hợp Nhất	1	UPCOM
3	AAM	CTCP Thủy sản MeKong	1	HOSE
4	AAS	CTCP Chứng khoán SmartInvest	4	UPCOM
5	AAT	CTCP Tập Đoàn Tiên Sơn Thanh Hóa	1	HOSE
6	AAV	CTCP AAV Group	1	HNX
7	ABB	Ngân hàng TMCP An Bình	2	UPCOM
8	ABC	CTCP Truyền thông VMG	1	UPCOM
9	ABI	CTCP Bảo hiểm Ngân hàng Nông nghiệp Việt Nam	3	UPCOM

Tên trường	Ý nghĩa
Ticker	Mã chứng khoán của công ty
organName	Tên đầy đủ của công ty
organTypeCode	Loại công ty
comGroupCode	Tên sàn giao dịch

2.1.2 Dữ liệu lịch sử giao dịch chứng khoán

Time	Open	High	Low	Close	Volume	Ticker
2016-05-20	4,840	5,070	4,840	5,070	2,000	BPC
2016-05-23	5,140	5,170	4,770	5,110	5,400	BPC
2016-05-24	5,070	5,140	5,070	5,140	1,100	BPC
2016-05-25	5,140	5,650	5,140	5,650	40,500	BPC
2016-05-26	5,790	6,200	5,790	6,030	70,400	BPC
2016-05-27	6,270	6,270	6,200	6,270	6,600	BPC
2016-05-30	5,930	5,930	5,650	5,650	20,110	BPC
2016-05-31	5,170	6,060	5,110	5,960	2,700	BPC
2016-06-01	5,760	5,930	5,380	5,930	22,300	BPC
2016-06-02	5,550	5,930	5,550	5,930	4,900	BPC

Tên trường	Ý nghĩa
Time	Ngày phiên giao dịch
Open	Giá mở
Close	Giá đóng
High	Giá cao nhất trong phiên giao dịch ngày
Low	Giá thấp nhất trong phiên giao dịch ngày
Volume	Khối lượng chứng khoán giao dịch trong ngày
Ticker	Mã chứng khoán của công ty

2.1.3 Bản tin công ty

SSN

Tên công ty: Seaprodex Saigon
Ngành: Food & Beverage
Năm thành lập: 1992
Số nhân viên: 12
<http://seaprodexsg.com>

SSN: Thông báo ngày đăng ký cuối cùng dự kiến để thực hiện quyền tham dự Đại hội đồng cổ đông thường niên 2024

Nguồn: HNX
Ngày đăng: 16 tháng 5, 2024
Giá: 1400 VND
Thay đổi: 0 (0.00%)

SSN: Nghị quyết Hội đồng quản trị

Nguồn: HNX
Ngày đăng: 16 tháng 5, 2024
Giá: 1400 VND
Thay đổi: 0 (0.00%)

SSN: Nghị quyết Hội đồng quản trị

Nguồn: HNX
Ngày đăng: 3 tháng 4, 2024
Giá: 1300 VND
Thay đổi: 0 (0.00%)

Với mỗi mã cổ phiếu công ty thu được các bộ dữ liệu sau: -
Thông tin công ty:

- Tên công ty
- Ngành kinh doanh
- Năm thành lập
- Số nhân viên • Website công ty - Bản tin công ty:
- Tiêu đề bản tin
- Nguồn
- Ngày đăng
- Giá cổ phiếu công ty ngày hôm đó
- % thay đổi giá cổ phiếu trong ngày bản tin được đăng tải

2.2 Phương Pháp Thu Thập

2.2.1 Dữ liệu công ty và dữ liệu lịch sử giao dịch chứng khoán

Sử dụng API được cung cấp từ thư viện vnstock để thu thập danh sách công ty niêm yết trên các sàn giao dịch.

Việc thu thập dữ liệu được tiến hành qua các bước sau:

- Kết Nối API: Sử dụng API key để truy cập vào dữ liệu từ thư viện vnstock. API này cung cấp thông tin về các công ty niêm yết và lịch sử giao dịch chứng khoán trên các sàn giao dịch tại Việt Nam.
- Truy Xuất Danh Sách Công Ty: API cho phép truy xuất danh sách các công ty niêm yết, bao gồm các thông tin cơ bản như mã chứng khoán, tên công ty, và ngày niêm yết. Dữ liệu này giúp xây dựng cơ sở dữ liệu ban đầu về các công ty đang giao dịch trên thị trường.
- Thu Thập Dữ Liệu Giao Dịch: Sử dụng API để thu thập lịch sử giao dịch chứng khoán của các công ty. Dữ liệu này bao gồm giá mở cửa, giá đóng cửa, giá cao nhất, giá thấp nhất, và khối lượng giao dịch hàng ngày. Việc thu thập được thực hiện định kỳ để đảm bảo dữ liệu luôn được cập nhật mới nhất.

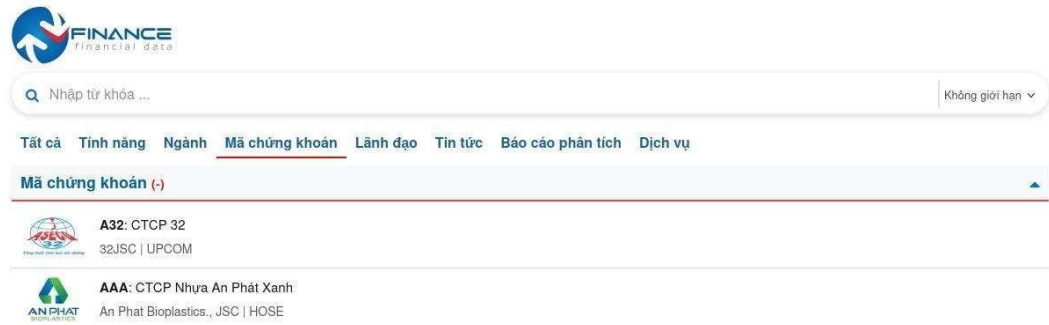
2.2.2 Dữ liệu bản tin của công ty:

Nguồn dữ liệu: vietstock.vn

Cách thu thập dữ liệu từng mã chứng khoán:

Bước 1: Thu thập dữ liệu URL

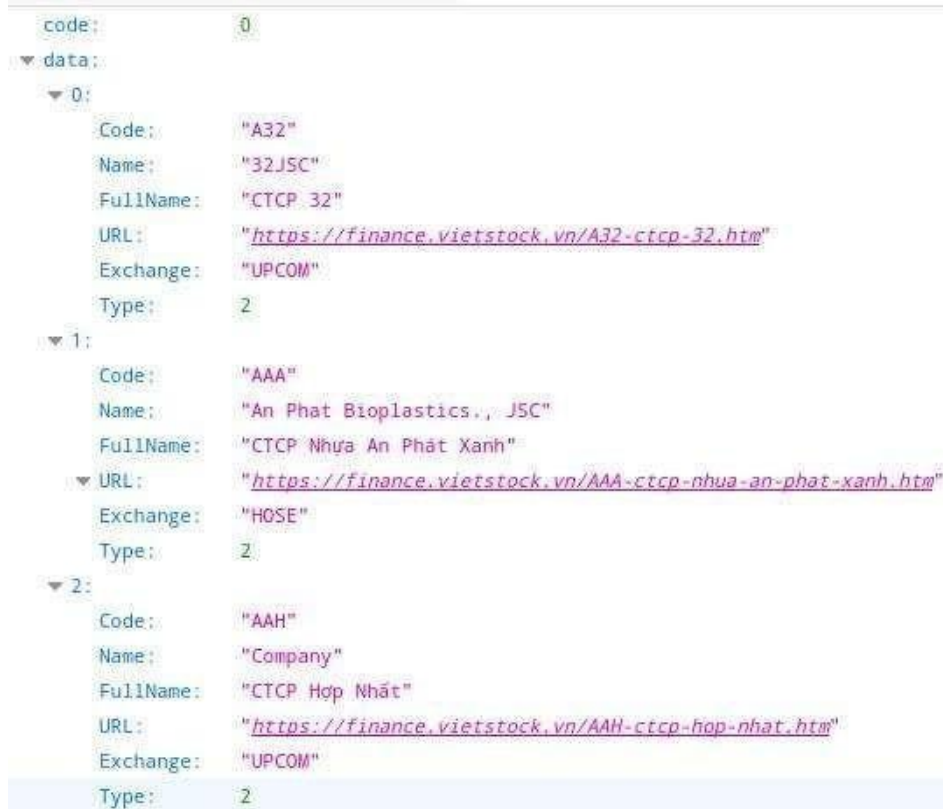
Khi vào trang tìm kiếm mã chứng khoán trên vietstock



Mỗi lần kéo xuống, trong tab network của browser sẽ xuất hiện 1 lần gọi đến API có dạng:

<https://finance.vietstock.vn/searchstock?query=&page=34&pageSize=10&languageId=1>

Sử dụng GET fetch API trên có được 10 công ty dưới dạng file json



Thay đổi API thành <https://finance.vietstock.vn/searchstock?query=&pageSize=2000&languageId=1>

Và sử dụng GET, lấy được danh sách công ty (1608 mã)

Bước 2: Lấy dữ liệu công ty và thông tin mới liên quan

Sau khi lấy được danh sách mã cổ phiếu, gọi API

“[https://apipubaws.tcbs.com.vn/tcanalysis/v1/ticker/\\${company}/overview](https://apipubaws.tcbs.com.vn/tcanalysis/v1/ticker/${company}/overview)” để xem thông tin công ty.

Để lấy danh sách tin tức mới nhất của công ty, từ các url thu thập được phía trên, sử dụng GET để lấy trang HTML. Sau đó, sử dụng DOMParser để tách tin tức ra khỏi trang html.

Từ các thông tin thu được, sử dụng HTML, CSS để biểu diễn lại dưới dạng extension

CHƯƠNG 3. TIỀN XỬ LÝ DỮ LIỆU

Tiền xử lý dữ liệu là một bước quan trọng trong quá trình xây dựng hệ thống dự đoán giá chứng khoán. Dữ liệu thô thường chứa nhiều nhiễu và không đồng nhất, cần được làm sạch và chuẩn hóa để đảm bảo chất lượng và tính nhất quán. Chương này sẽ trình bày các bước tiền xử lý dữ liệu bao gồm làm sạch dữ liệu, chuẩn hóa dữ liệu và tính toán thêm các trường dữ liệu như % thay đổi và chỉ số Bollinger Bands.

3.1 Làm sạch dữ liệu

3.1.1 Xử lý dữ liệu thiếu

Có nhiều phương pháp để xử lý dữ liệu thiếu:

- Loại bỏ các hàng hoặc cột có dữ liệu thiếu: Đây là phương pháp đơn giản nhưng có thể dẫn đến mất mát thông tin đáng kể.
- Điền giá trị trung bình hoặc trung vị: Điền giá trị trung bình hoặc trung vị của cột dữ liệu có thể giúp giữ lại nhiều thông tin hơn.
- Sử dụng các phương pháp dự đoán: Sử dụng các mô hình dự đoán để điền giá trị thiếu dựa trên các dữ liệu hiện có.

Dữ liệu về lịch sử giá cổ phiếu được tính toán thêm các trường như % thay đổi, chỉ số Bollinger Bands để tiện tính toán

3.1.2 Loại bỏ nhiễu

Dữ liệu nhiễu là những dữ liệu không phù hợp hoặc không chính xác, gây ảnh hưởng đến chất lượng của mô hình dự đoán. Các phương pháp phổ biến để loại bỏ nhiễu bao gồm:

- Phát hiện và loại bỏ các giá trị ngoại lệ (outliers): Sử dụng các phương pháp thống kê để phát hiện và loại bỏ các giá trị ngoại lệ.
- Sử dụng các kỹ thuật lọc: Áp dụng các kỹ thuật lọc như lọc trung bình động (moving average) để làm mịn dữ liệu.

3.2 Biến đổi dữ liệu

Biến đổi dữ liệu là quá trình tạo ra các đặc trưng mới hoặc biến đổi các đặc trưng hiện có để tăng cường khả năng học của mô hình.

3.2.1 Tính toán % thay đổi

% thay đổi là tỉ lệ giữa chênh lệch giá đóng cửa của hai ngày liên tiếp và giá đóng cửa của ngày hôm trước. Công thức tính:

$$\% \text{ Thay đổi} = \frac{\text{Giá đóng cửa hôm nay} - \text{Giá đóng cửa hôm qua}}{\text{Giá đóng cửa hôm qua}} \times 100$$

3.2.2 Tính toán chỉ số Bollinger Bands

Chỉ số Bollinger Bands bao gồm ba dải: dải giữa là đường trung bình động chu kỳ 20 ngày (SMA20), dải trên và dải dưới được tính bằng SMA20 cộng/trừ 2 lần độ lệch chuẩn 20 ngày.

- SMA20: Đường trung bình động chu kỳ 20 ngày
- Dải trên: $SMA20 + 2 * \text{Độ lệch chuẩn 20 ngày}$
- Dải dưới: $SMA20 - 2 * \text{Độ lệch chuẩn 20 ngày}$

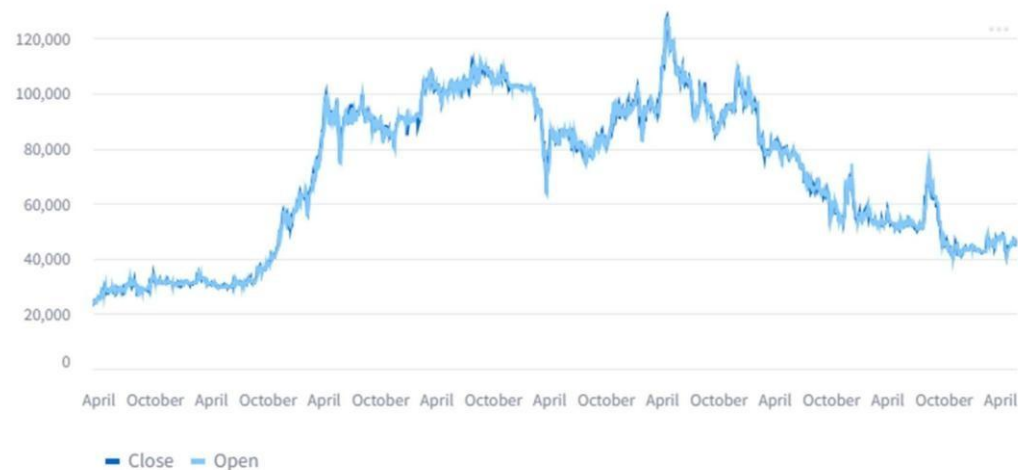
Time	Open	High	Low	Close	Volume	Ticker	Change	upper_band	lower_band
2016-04-11	4,080	4,090	4,090	4,090	2,900	BPC	0	4,198.7569	4,062.2431
2016-04-12	4,080	4,090	4,090	4,090	1,750	BPC	0	4,187.45	4,062.55
2016-04-13	4,080	4,090	4,090	4,090	3,200	BPC	0	4,173.3424	4,065.6576
2016-04-14	4,090	4,090	4,090	4,090	0	BPC	0	4,154.694	4,073.306
2016-04-15	4,230	4,300	4,200	4,240	16,200	BPC	0.0367	4,188.4105	4,051.5895
2016-04-19	4,230	4,240	4,240	4,240	600	BPC	0	4,209.9342	4,039.0658
2016-04-20	4,240	4,240	4,240	4,240	0	BPC	0	4,227.7725	4,030.2275
2016-04-21	4,230	4,270	4,240	4,240	14,500	BPC	0	4,243.277	4,023.723
2016-04-22	4,230	4,490	4,240	4,430	12,200	BPC	0.0448	4,318.2217	3,979.7783
2016-04-25	4,520	4,530	4,460	4,490	10,500	BPC	0.0135	4,391.8992	3,943.1008

3.3 Trực quan hóa dữ liệu

Trực quan hóa dữ liệu sử dụng 6 chế độ đồ thị như sau:

1. Close – open

Close - Open Price



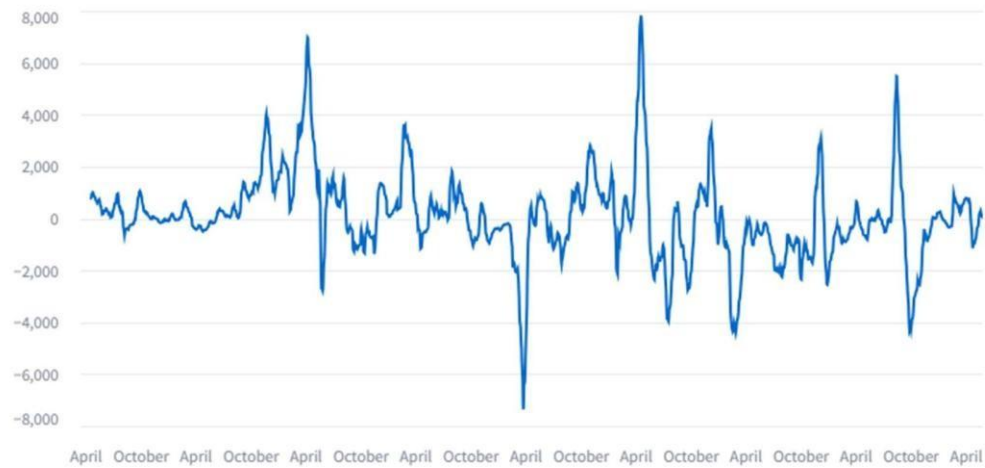
2. BB

BollingerBands



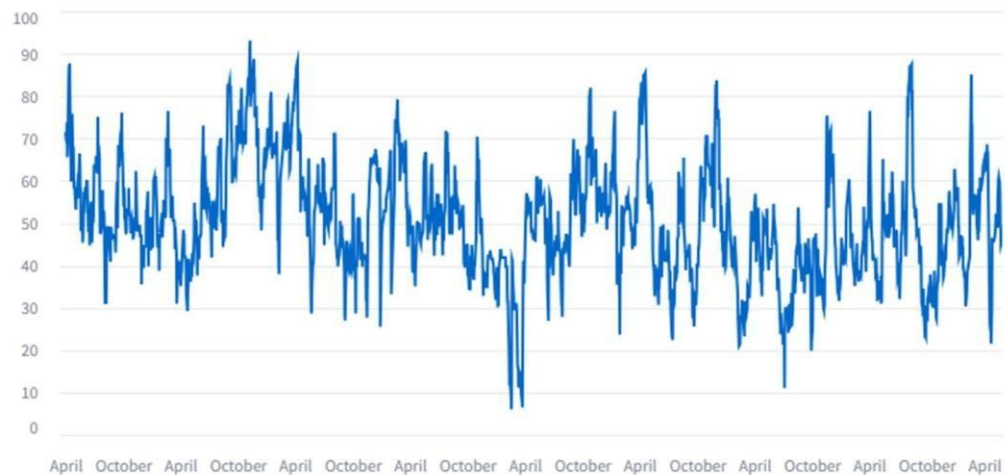
3. MACD (Moving Average Convergence Divergence)

Moving Average Convergence Divergence



4. RSI (Relative Strength Indicator):

Relative Strength Indicator



5. SMA (Simple Moving Average):

Simple Moving Average



6. EMA (Exponential Moving Average)

Exponential Moving Average



CHƯƠNG 4. XÂY DỰNG MÔ HÌNH DỰ ĐOÁN

Chúng ta sẽ sử dụng năm mô hình khác nhau: Linear Regression, Random Forest Regressor, K-Neighbors Regressor, Prophet và Long Short-Term Memory (LSTM).

4.1 Linear Regression

Hồi quy tuyến tính là một phương pháp thống kê được sử dụng để mô hình hóa mối quan hệ giữa một biến phụ thuộc (thường được ký hiệu là yy) và một hoặc nhiều biến độc lập (thường được ký hiệu là xx). Mục tiêu của hồi quy tuyến tính là tìm ra một đường thẳng tốt nhất (hoặc một mặt phẳng trong trường hợp nhiều biến độc lập) để dự đoán giá trị của biến phụ thuộc dựa trên giá trị của các biến độc lập.

Mô hình hồi quy tuyến tính đơn giản:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

Tuy nhiên, bài toán của chúng ta không đơn thuần chỉ có một biến, mà có nhiều biến số khác nhau.

Hồi quy tuyến tính đa biến:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \varepsilon$$

Với:

- y là biến phụ thuộc.
- x_1, x_2, \dots, x_n là các biến độc lập.
- β_0 là hằng số chặn.
- $\beta_1, \beta_2, \dots, \beta_n$ là các hệ số hồi quy tương ứng với các biến độc lập.
- ε là sai số.

4.2 K-Neighbors

K-Nearest Neighbors Regression (KNN Regression) là một thuật toán học máy không tuyến tính được sử dụng để dự đoán giá trị của một biến liên tục. KNN Regression dựa trên ý tưởng rằng những điểm dữ liệu gần nhau có xu hướng có giá trị tương tự nhau.

Ý tưởng chính:

Để dự đoán giá trị của một điểm mới, KNN Regression tìm k điểm dữ liệu gần nhất trong không gian đầu vào và sử dụng giá trị trung bình của chúng để đưa ra dự đoán.

Các bước thực hiện:

- Chọn số lượng hàng xóm (k): Số lượng k được chọn bởi người dùng. Số lượng này có thể ảnh hưởng lớn đến hiệu suất của mô hình.
- Tính khoảng cách: Sử dụng một số biện pháp tính khoảng cách như khoảng cách Euclid để xác định các điểm dữ liệu gần nhất.
- Lấy giá trị trung bình: Tính giá trị trung bình của k điểm dữ liệu gần nhất để dự đoán.

Ưu điểm:

- Đơn giản và dễ hiểu.
- Không cần giả định trước về phân phối của dữ liệu.

Nhược điểm:

- Tính toán chậm với dữ liệu lớn.
- Nhạy cảm với giá trị ngoại lai.
- Hiệu suất phụ thuộc vào cách chọn k và khoảng cách.

4.3 Random Forest

Random Forest là một phương pháp ensemble learning sử dụng nhiều cây quyết định và kết hợp kết quả của chúng để đưa ra dự đoán cuối cùng. Trong trường hợp hồi quy, kết quả của mỗi cây quyết định là một giá trị số, và kết quả cuối cùng là trung bình của tất cả các giá trị dự đoán từ các cây.

Tạo ra Random Forest:

1. Bootstrap Aggregation (Bagging): Tạo ra nhiều mẫu dữ liệu bằng cách lấy mẫu ngẫu nhiên từ tập dữ liệu gốc với thay thế (có thể một điểm dữ liệu xuất hiện nhiều lần trong một mẫu).
2. Xây dựng các cây quyết định: Tạo ra các cây quyết định từ các mẫu bootstrap, mỗi cây được xây dựng với một tập con ngẫu nhiên của các đặc trưng.
3. Dự đoán: Đối với một mẫu mới, mỗi cây quyết định đưa ra một dự đoán và kết quả cuối cùng là trung bình các dự đoán đó.

Quá trình tạo ra 1 cây con:

1. Cấu trúc của một cây quyết định:
 - Nút gốc (Root Node): Nút đầu tiên của cây, đại diện cho toàn bộ tập dữ liệu.
 - Nút quyết định (Decision Nodes): Các nút trung gian chia dữ liệu dựa trên một điều kiện.
 - Nút lá (Leaf Nodes): Nút cuối cùng của cây, đại diện cho kết quả dự đoán. Đối với nhiệm vụ phân loại, các nút lá chứa các nhãn lớp. Đối với nhiệm vụ hồi quy, các nút lá chứa giá trị liên tục.
2. Quá trình xây dựng cây:
 - Chọn đặc trưng để chia (Feature Selection): Quyết định đặc trưng nào sẽ được sử dụng để chia dữ liệu tại mỗi nút quyết định. Với bài toán hồi quy, ta sử dụng MSE đo lường độ lệch bình phương trung bình giữa giá trị dự đoán và giá trị thực tế.
 - Chia dữ liệu: Tạo các nhánh mới dựa trên đặc trưng được chọn.

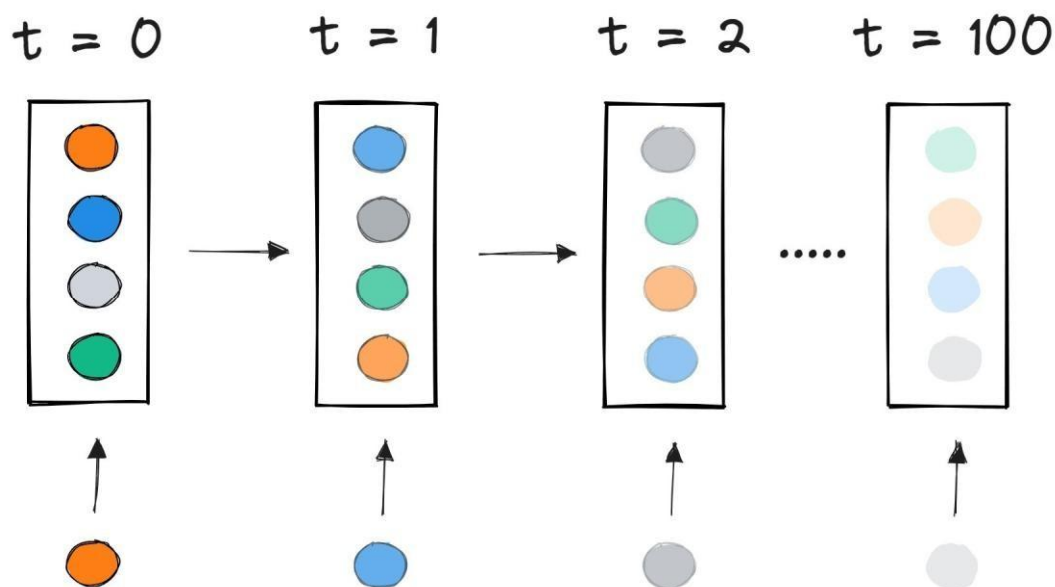
- Lặp lại: Quy trình này lặp lại đệ quy cho đến khi một điều kiện dừng được thỏa mãn (chẳng hạn như đạt đến độ sâu tối đa của cây hoặc không còn đặc trưng nào để chia).

4.4 LSTM

LSTM (Long Short-Term Memory) là một loại mạng nơ-ron hồi quy (Recurrent Neural Network - RNN) được thiết kế để xử lý và dự đoán dữ liệu tuần tự và chuỗi thời gian (time series). LSTM có khả năng ghi nhớ thông tin trong khoảng thời gian dài và giải quyết được vấn đề gradient biến mất (vanishing gradient problem) mà các RNN truyền thống gặp phải.

4.4.1 Vanishing gradient

Vanishing Gradient là một vấn đề phổ biến trong việc huấn luyện các mạng nơ-ron sâu, đặc biệt là trong các mạng hồi quy như RNNs và LSTMs. Vấn đề này xảy ra khi các gradient của hàm mất mát trở nên cực kỳ nhỏ trong quá trình truyền ngược, khiến cho việc cập nhật trọng số trong các lớp đầu tiên trở nên không đáng kể. Điều này làm cho mô hình học rất chậm hoặc thậm chí không học được gì.



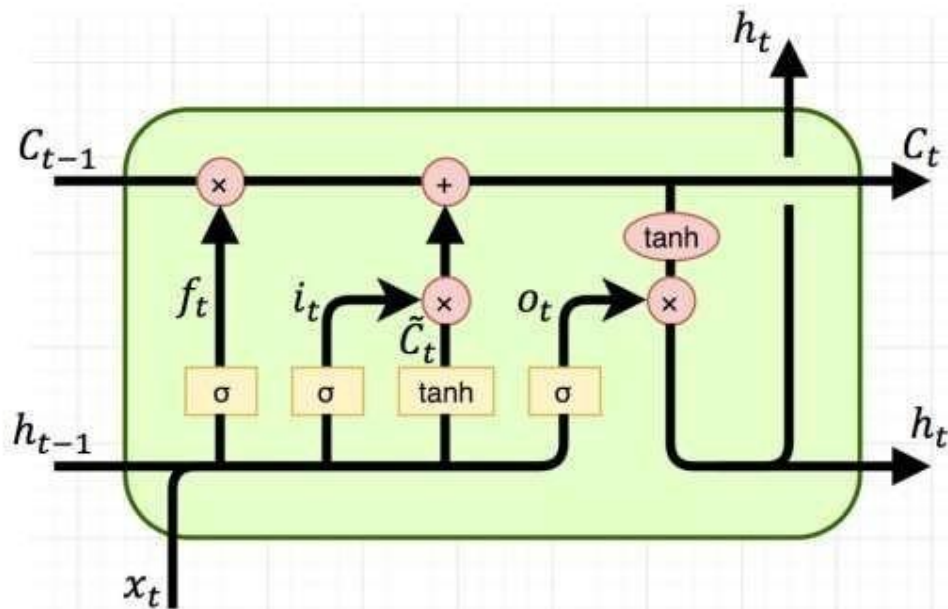
4.4.2 Dữ liệu chuỗi thời gian (Time series)

Là một loại dữ liệu quan trọng trong nhiều lĩnh vực như tài chính, khí tượng học, y tế, và nhiều lĩnh vực khác. Dữ liệu chuỗi thời gian là một tập hợp các điểm dữ liệu được thu thập hoặc ghi nhận tại các khoảng thời gian liên tục.

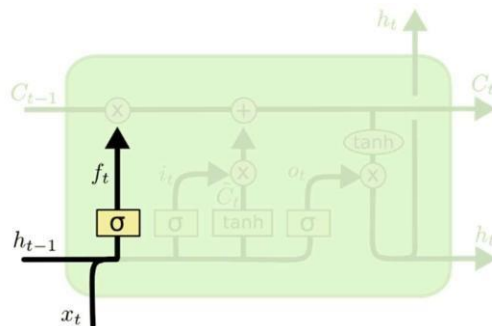
Đặc điểm của Dữ liệu Chuỗi Thời gian: - Thời gian: Mỗi điểm dữ liệu có một nhãn thời gian tương ứng.

- Phụ thuộc: Giá trị của một điểm dữ liệu thường phụ thuộc vào các giá trị trước đó trong chuỗi.
- Xu hướng và mùa vụ: Nhiều chuỗi thời gian có thể có xu hướng dài hạn và các mô hình theo mùa lặp lại.

4.4.3 Cấu trúc LSTM

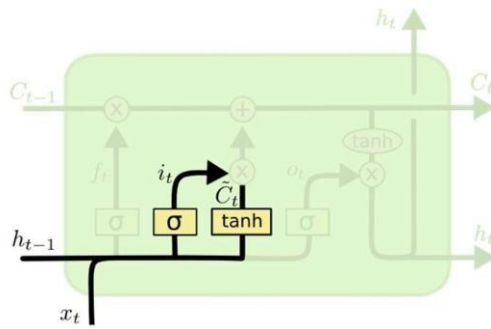


Bước đầu tiên trong LSTM sẽ quyết định xem thông tin nào chúng ta sẽ cho phép đi qua ô trạng thái (cell state)



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

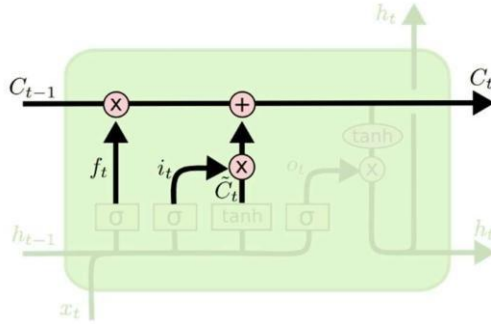
Bước tiếp theo chúng ta sẽ quyết định loại thông tin nào sẽ được lưu trữ trong ô trạng thái. Bước này bao gồm 2 phần. Phần đầu tiên là một tầng ẩn của hàm sigmoid được gọi là tầng cổng vào (input gate layer) quyết định giá trị bao nhiêu sẽ được cập nhật. Tiếp theo, tầng ẩn hàm tanh sẽ tạo ra một véc tơ của một giá trị trạng thái mới mà có thể được thêm vào trạng thái. Tiếp theo kết hợp kết quả của 2 tầng này để tạo thành một cập nhật cho trạng thái.



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

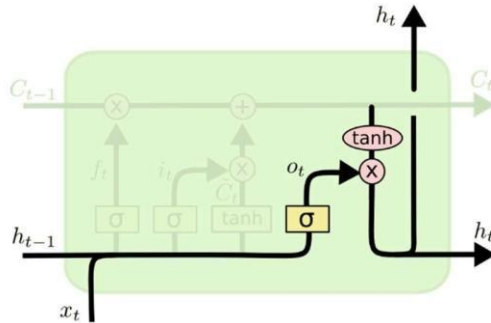
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Cập nhật giá trị cho ô trạng thái bằng cách kết hợp 2 kết quả từ tầng cộng vào và tầng ản hàm tanh:



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

Cuối cùng cần quyết định xem đầu ra sẽ trả về bao nhiêu. Kết quả ở đầu ra sẽ dựa trên ô trạng thái, nhưng sẽ là một phiên bản được lọc. Đầu tiên, chúng ta chạy qua một tầng sigmoid nơi quyết định phần nào của ô trạng thái sẽ ở đầu ra. Sau đó, ô trạng thái được đưa qua hàm tanh (để chuyển giá trị về khoảng -1 và 1) và nhân nó với đầu ra của một cổng sigmoid, do đó chỉ trả ra phần mà chúng ta quyết định.



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

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

4.5 Prophet

Prophet là một thư viện mã nguồn mở được phát triển bởi Facebook (nay là Meta) nhằm giải quyết bài toán dự báo chuỗi thời gian. Thư viện này đặc biệt phù hợp cho những dữ liệu có tính chất thời vụ và xu hướng, chẳng hạn như dữ liệu doanh thu, lưu lượng truy cập website, hay nhu cầu hàng hóa theo thời gian. Prophet được thiết kế với ưu điểm dễ sử dụng và cung cấp kết quả đáng tin cậy ngay cả khi người dùng không có nhiều kinh nghiệm về chuỗi thời gian.

Các đặc điểm nổi bật của Prophet bao gồm:

- Đơn giản hóa quy trình dự báo: Prophet yêu cầu rất ít tinh chỉnh tham số, giúp người dùng nhanh chóng triển khai mô hình dự báo.
- Hỗ trợ tính thời vụ: Prophet tự động nhận diện và xử lý các yếu tố thời vụ trong dữ liệu, chẳng hạn như theo ngày, tuần, tháng hoặc năm.
- Khả năng xử lý dữ liệu bị thiếu hoặc dữ liệu không đều: Thư viện này có thể làm việc với dữ liệu có khoảng cách thời gian không đồng nhất.

- Tùy chỉnh linh hoạt: Người dùng có thể dễ dàng điều chỉnh các yếu tố xu hướng, thời vụ và ngày lễ để cải thiện độ chính xác.
- Tích hợp đa ngôn ngữ: Prophet có sẵn trong cả Python và R, giúp tiếp cận được đa dạng cộng đồng lập trình.

Cấu trúc của Prophet:

Prophet hoạt động dựa trên mô hình cộng tuyến tính:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

Với:

- $g(t)$: Xu hướng (trend).
- $s(t)$: Thời vụ (seasonality).
- $h(t)$: Các ngày đặc biệt (holidays).
- ε_t : Nhiễu (noise)

Lợi ích khi sử dụng Prophet

- Tiết kiệm thời gian nhờ quy trình dự báo tự động và trực quan.
- Cung cấp mô hình có khả năng giải thích rõ ràng, phù hợp cho cả người dùng không chuyên và chuyên gia.

CHƯƠNG 5. ĐÁNH GIÁ MÔ HÌNH

Để đánh giá các mô hình, chúng ta sẽ đánh giá sử dụng 3 metrics, bao gồm R^2 score, MAE và MSE

5.1 R^2 score

Hệ số xác định (coefficient of determination) là một đại lượng trong thống kê được sử dụng để đánh giá mức độ phù hợp của một mô hình hồi quy tuyến tính với dữ liệu. Hệ số này thường được ký hiệu là R^2 .

Hệ số xác định cho biết tỉ lệ phương sai của biến mục tiêu (outcome variable) được giải thích bởi các biến độc lập (independent variables) trong mô hình hồi quy tuyến tính. Nó thường được tính bằng cách so sánh phương sai giữa mô hình hồi quy và phương sai của giá trị trung bình của biến mục tiêu. Công thức toán học:

$$R = 1 - \frac{SS_{res}}{SS_{tot}}$$

Trong đó:

- SS_{res} (Sum of Squares of Residuals): tổng các độ lệch bình phương của phần dư.
- SS_{tot} (Total sum of the errors) tổng độ lệch bình phương của toàn bộ các nhân tố nghiên cứu, được tính bằng công thức:

$$SS_{tot} = \sum_{i=1}^n (y_i - \bar{y})^2$$

- y_i là giá trị thực tế của biến phụ thuộc.
- \bar{y} là giá trị dự đoán của biến phụ thuộc dựa trên các giá trị độc lập được sử dụng trong mô hình.

Ý nghĩa của R^2 Score:

- $R^2 = 1$: Mô hình dự đoán hoàn hảo, mọi điểm dữ liệu đều nằm trên đường hồi quy.
- $R^2 = 0$: Mô hình không giải thích được sự biến thiên nào của dữ liệu; giá trị dự đoán bằng với giá trị trung bình của biến phụ thuộc.
- $0 < R^2 < 1$: Mức độ mà mô hình giải thích được sự biến thiên của dữ liệu, giá trị càng cao thì mô hình càng tốt.
- $R^2 < 0$: Điều này có thể xảy ra nếu mô hình dự đoán kém hơn so với việc chỉ dự đoán bằng giá trị trung bình. Trong trường hợp này, mô hình không phù hợp với dữ liệu.

5.2 MSE

Mean Squared Error (MSE) có lẽ là số liệu phổ biến nhất được sử dụng cho các bài toán hồi quy. Về cơ bản, nó tìm thấy sai số bình phương trung bình giữa các giá trị được dự đoán và thực tế. MSE là thước đo chất lượng của một công cụ ước tính - nó luôn không âm và các giá trị càng gần 0 càng tốt.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y}_i)^2$$

Trong đó:

- n là số điểm dữ liệu
- y_i là giá trị quan sát
- \bar{y}_i là giá trị dự đoán.

Trong phân tích hồi quy, vẽ biểu đồ là một cách tự nhiên hơn để xem xu hướng chung của toàn bộ dữ liệu. Đơn giản MSE cho bạn biết mức độ gần của đường hồi quy với một tập hợp các điểm. Nó thực hiện điều này bằng cách lấy khoảng cách từ các điểm đến đường hồi quy (những khoảng cách này là “sai số”) và bình phương chúng. Bình phương là rất quan trọng để giảm độ phức tạp với các dấu hiệu tiêu cực. Nó cũng tạo ra nhiều trọng lượng hơn cho sự khác biệt lớn hơn.

Để giảm thiểu MSE, mô hình có thể chính xác hơn, có nghĩa là mô hình gần với dữ liệu thực tế hơn. Một ví dụ về hồi quy tuyến tính sử dụng phương pháp này là - phương pháp bình phương nhỏ nhất đánh giá sự phù hợp của mô hình hồi quy tuyến tính với tập dữ liệu hai biến, nhưng giới hạn của nó liên quan đến phân phối dữ liệu đã biết.

MSE càng thấp thì dự báo càng tốt.

5.3 MAE

Mean Absolute Error (MAE) đo độ lớn trung bình của các lỗi trong một tập hợp các dự đoán mà không cần xem xét hướng của chúng. Đó là giá trị trung bình trên mẫu thử nghiệm về sự khác biệt tuyệt đối giữa dự đoán và quan sát thực tế, trong đó tất cả các khác biệt riêng lẻ có trọng số bằng nhau.

$$MSE = \frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}_i|$$

Trong đó:

- n là số điểm dữ liệu
- \bar{y}_i là giá trị thực
- y_i là giá trị dự đoán.

Có thể diễn đạt MAE là tổng hòa của hai thành phần: Bất đồng về số lượng và Bất đồng về phân bố. MAE được biết đến là mạnh mẽ hơn đối với các yếu tố ngoại lai so với MSE. Lý do chính là trong MSE bằng cách bình phương các sai số, các giá trị ngoại lai (thường có sai số cao hơn các mẫu khác) được chú ý nhiều hơn và chiếm ưu thế trong sai số cuối cùng và tác động đến các tham số của mô hình.

CHƯƠNG 6. LỰA CHỌN MÔ HÌNH ĐỀ XUẤT

Nhóm đề xuất mô hình chính cần lưu ý đó là LSTM. Việc lựa chọn mô hình LSTM cho mô hình đề xuất được dựa trên những ưu điểm nổi bật trong việc dự báo dữ liệu chuỗi thời gian.

LSTM, vượt trội trong việc nắm bắt các phụ thuộc dài hạn và các mẫu phức tạp trong dữ liệu, phù hợp khi xu hướng quá khứ ảnh hưởng mạnh đến kết quả tương lai.

Bước	Mô tả
Chuẩn bị dữ liệu	Loại bỏ cột, hoán đổi, chuẩn hóa dữ liệu
Chuẩn hóa dữ liệu	Đưa dữ liệu về thang [0, 1]
Chia tập huấn luyện và kiểm tra	98% huấn luyện, 2% kiểm tra
Tạo tập dữ liệu	Tạo chuỗi cho LSTM
Định nghĩa mô hình	Định nghĩa LSTM với 64 đơn vị, lớp kết nối đầy đủ
Huấn luyện mô hình	Huấn luyện với dừng sớm, lưu mô hình tốt nhất
Dự đoán và đánh giá mô hình	Đánh giá, dự đoán tương lai bằng đầu vào lặp
Giá trị trả về	Trả về thực tế, dự đoán, mô hình, dự đoán tương lai

6.1 Chuẩn bị dữ liệu

```
def LSTM1(data,num):  
    dataset = data  
    dataset = dataset.drop(dataset.columns[5], axis=1)  
    # Drop the 6th column (index 5)  
    dataset = dataset.drop(dataset.columns[5], axis=1)  
  
    #swap column 3 with column 4 with names  
    cols = list(dataset.columns.values)  
    cols[3], cols[4] = cols[4], cols[3]  
    dataset = dataset[cols]  
  
    # Show the head of the normalized dataset  
    print(dataset.head())
```

- Loại bỏ cột thứ 6 (chỉ số 5) hai lần, đảm bảo việc xóa.

- Hoán đổi cột 3 và 4 để sắp xếp lại các đặc trưng cho đầu vào mô hình tốt hơn.
- In đầu tập dữ liệu để kiểm tra các thay đổi.

6.2 Chuẩn hóa dữ liệu

```
# Normalize the dataset
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(dataset)
```

- Sử dụng MinMaxScaler để chuẩn hóa tất cả các đặc trưng về khoảng [0, 1], giúp ổn định quá trình huấn luyện và cải thiện sự hội tụ cho mạng nơ-ron.

6.3 Chia tập huấn luyện và kiểm tra

```
# Split into training and test sets
train_size = int(len(scaled_data) * 0.8)
train_data = scaled_data[:train_size]
test_data = scaled_data[train_size:]
```

- Chia dữ liệu thành 80% tập huấn luyện và 20% tập kiểm tra tối ưu cho LSTM trong dự báo chuỗi thời gian

6.4 Tạo tập dữ liệu

```
# Convert the data to a format suitable for LSTM
def create_dataset(data, look_back=5):
    X, Y = [], []
    for i in range(len(data) - look_back):
        X.append(data[i:(i + look_back), :])
        Y.append(data[i + look_back, :]) # Predicting the entire feature set
    return np.array(X), np.array(Y)

look_back = 1 # You can change this value based on the desired look back period
X_train, y_train = create_dataset(train_data, look_back)
X_test, y_test = create_dataset(test_data, look_back)
```

- Tạo các chuỗi nơi mỗi đầu vào (X) là look_back bước thời gian, và đầu ra (Y) là bước thời gian tiếp theo. Ở đây, look_back là 1, nghĩa là mỗi đầu vào là một bước thời gian duy nhất.
- Chuyển đổi thành tensor PyTorch cho đầu vào mô hình.

6.5 Định nghĩa mô hình

```
class NeuralNetwork(nn.Module):
    def __init__(self, num_feature):
        super(NeuralNetwork, self).__init__()
        self.lstm = nn.LSTM(num_feature, 64, batch_first=True)
        self.fc = nn.Linear(64, num_feature)

    def forward(self, x):
        output, (hidden, cell) = self.lstm(x)
        x = self.fc(hidden[-1])
        return x
```

- Định nghĩa một mô hình LSTM với 64 đơn vị ẩn và một lớp kết nối đầy đủ ánh xạ đến chiều đặc trưng đầu vào.
- Phương thức forward xử lý đầu vào qua LSTM rồi qua lớp kết nối đầy đủ.

6.6 Huấn luyện mô hình

```
model = NeuralNetwork(num_feature=X_train.shape[2])

# Define loss function and optimizer
criterion = nn.MSELoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)

# Training loop with early stopping and saving the best model
num_epochs = 50
best_val_loss = float('inf')
best_model_state = None
patience = 10
for epoch in range(num_epochs):
    model.train()
    train_loss = 0.0
    for batch_x, batch_y in train_loader:
        optimizer.zero_grad()
        output = model(batch_x)
        loss = criterion(output, batch_y)
        loss.backward()
        optimizer.step()
        train_loss += loss.item()

    # Validate the model on the test set
    model.eval()
    with torch.no_grad():
        test_output = model(X_test)
        val_loss = criterion(test_output, y_test)

    # Print training and validation loss
    print(f'Epoch [{epoch+1}/{num_epochs}], Train Loss: {train_loss/len(train_loader):.4f}, Val Loss: {val_loss.item():.4f}')

    # Check for early stopping and save the best model
    if val_loss < best_val_loss:
        best_val_loss = val_loss
        best_model_state = model.state_dict().copy()
        counter = 0
    else:
        counter += 1
        if counter >= patience:
            print(f'Validation loss hasn't improved for {patience} epochs. Early stopping...')
            break

# Save the best model
if best_model_state is not None:
    torch.save(best_model_state, 'LSTM.pth')
    print('Best model saved.')
else:
    print('No improvements in validation loss. Best model not saved.')
```

- Huấn luyện tối đa 50 epoch với hàm mất mát MSE và tối ưu hóa Adam.
- Sử dụng dừng sớm với kiên nhẫn 10 epoch, lưu mô hình tốt nhất dựa trên mất mát xác thực.

6.7 Dự đoán và đánh giá mô hình

```
# Evaluate the model on the test set
model.eval()
with torch.no_grad():
    test_output = model(X_test)
    test_loss = criterion(test_output, y_test)
    print(f'Test Loss: {test_loss.item():.4f}')

# Inverse transform the predictions and actual values for plotting
test_output_np = test_output.numpy()
y_test_np = y_test.numpy()

test_output_scaled = scaler.inverse_transform(test_output_np)
y_test_scaled = scaler.inverse_transform(y_test_np)

num_days = num

# Last day of the test data
last_day = X_test[-1].unsqueeze(0) # Reshape to match model input

# Create a copy of the last day to use for predictions
next_day = last_day.clone()

# Store the predicted values
predicted_prices = []

# Predict the next day's price and use it for the next prediction
with torch.no_grad():
    for _ in range(num_days):
        # Predict the next day's price
        prediction = model(next_day)

        # Append the prediction to the list of predicted prices
        predicted_prices.append(prediction.squeeze().numpy())

        # Update the next_day tensor with the new prediction
        next_day = torch.cat((next_day[:, 1:, :], prediction.unsqueeze(0)), dim=1)

# Convert the list of predicted prices to a numpy array
predicted_prices = np.array(predicted_prices)

# Inverse transform the predicted prices to get them back to the original scale
predicted_prices_scaled = scaler.inverse_transform(predicted_prices)
```

- Đánh giá trên tập kiểm tra, sau đó dự đoán num_days trong tương lai bằng cách sử dụng dự đoán làm đầu vào lặp đi lặp lại.
- Chuyển ngược dự đoán về thang đo ban đầu.

6.8 Giá trị trả về

```
return y_test_scaled[:, -1], test_output_scaled[:, -1], model, predicted_prices_scaled[:, -1]
```

- Trả về giá trị thực tế của tập kiểm tra, giá trị dự đoán của tập kiểm tra, mô hình và dự đoán trong tương lai.


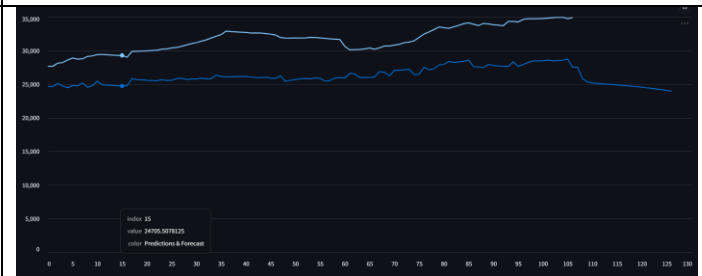
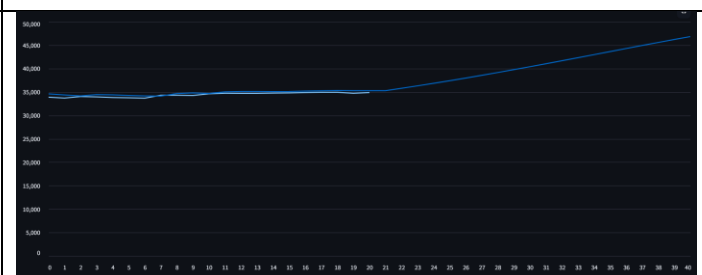
6.9 Các case study nổi bật khi nhóm rút ra trong quá trình triển khai mô hình LSTM

Forecasting khung thời gian 20 ngày cho ngày cho mã cổ phiếu A32

- Duration : 3000
- Start Date – End Date : 6/4/2017 – 23/6/2025

Thu được các case-study chú ý như sau :

6.9.1 Ảnh hưởng của tỷ lệ chia tập huấn luyện và kiểm tra

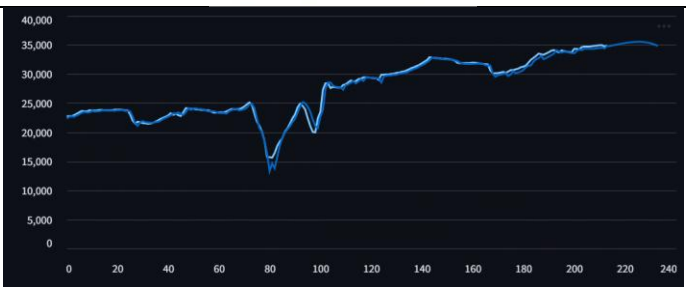
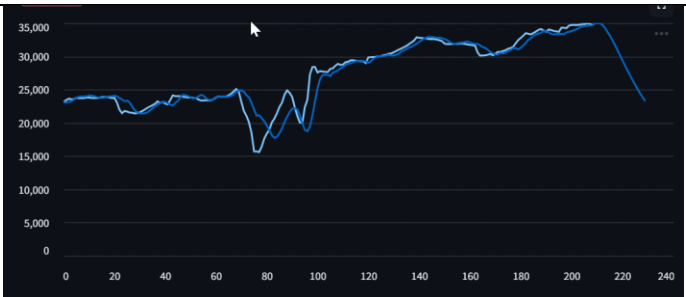
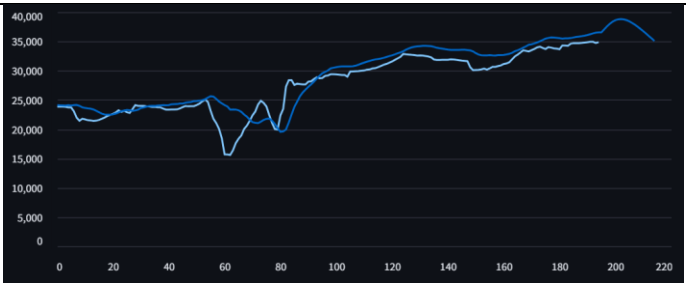
Tỉ lệ	R ² score	MSE	MAE	
80/20	0.979220807 5523376	386.9797668 457031	496702.5	
90/10	-6.94850206 3751221	5313.912597 65625	29369762.0	
98/2	0.078190147 8767395	398.0189819 3359375	182432.703 125	

Kết luận :

- **80/20:** Cung cấp cách tiếp cận cân bằng, với 80% dữ liệu cho huấn luyện và 20% cho kiểm tra. **Đây là lựa chọn tối ưu nhất** dựa trên các chỉ số được cung cấp, với khả năng dự đoán tốt và độ sai lệch thấp.
- **98/2:** R² thấp (0.0782), MSE ở mức trung bình (398.02), và MAE lớn (182432.70), cho thấy mô hình không hiệu quả do tập kiểm tra quá nhỏ dẫn đến đánh giá không chính xác. Nên tăng kích thước tập kiểm tra để có đánh giá đáng tin cậy hơn.

- **90/10**: có hiệu suất tệ nhất với R^2 âm (-6.9485), MSE cực cao (3751221), và MAE ở mức trung bình (5313.91). Do dữ liệu không đủ đại diện, hoặc cấu hình mô hình không phù hợp, chỉ phù hợp cho LSTM nhiều dữ liệu

6.9.2 Ảnh hưởng của chỉ số Look-Back

Look back	R^2 score	MSE	MAE	
1	0.984340846 5385437	383.08917 23632812 5	374314.437 5	
5	0.894657909 8701477	882.75128 17382812	2524364.0	
20	0.803252577 7816772	1600.3283 69140625	4849972.5	

So sánh và nhận xét

- Khi **look back** tăng từ 1 lên 5 và 20:
 - **R^2 giảm dần** (0.9843 → 0.8947 → 0.8033): Hiệu suất mô hình giảm khi sử dụng nhiều dữ liệu lịch sử hơn.
 - **MSE tăng dần** (383.09 → 882.75 → 1600.33): Sai số bình phương trung bình tăng, cho thấy dự đoán kém chính xác hơn.
 - **MAE tăng mạnh** (374314.44 → 2524364.0 → 4849972.5): Sai số tuyệt đối trung bình tăng đáng kể, đặc biệt ở look back = 20.

Điều này cho thấy rằng với tập dữ liệu và cấu hình mô hình hiện tại, **look back = 1** mang lại hiệu suất tốt nhất trên cả ba chỉ số.

Giải thích tại sao look back = 1 lại tối ưu

- **Dữ liệu có thể phụ thuộc mạnh vào ngẫu nhiên:** Nếu dữ liệu (ví dụ: giá cổ phiếu) có tính biến động cao hoặc tuân theo mô hình "random walk", thì giá trị gần nhất (look back = 1) có thể là yếu tố dự đoán tốt nhất, trong khi dữ liệu lịch sử dài hơn (look back = 5 hoặc 20) có thể gây nhiễu.
- **Mô hình chưa đủ phức tạp:** Mô hình LSTM với số lượng đơn vị ẩn cố định (ví dụ: 64) có thể không đủ khả năng xử lý các chuỗi dài hơn (look back lớn), dẫn đến việc học các mẫu phức tạp kém hiệu quả.
- **Số lượng mẫu huấn luyện giảm:** Khi look back tăng, số lượng mẫu huấn luyện khả dụng giảm (vì mỗi mẫu cần nhiều điểm dữ liệu hơn), có thể ảnh hưởng đến khả năng học của mô hình.

6.9.3 Ảnh hưởng của số epoch huấn luyện tối đa (num_epochs) và epoch kiên nhẫn (patience) đối với hiệu suất của mô hình LSTM.

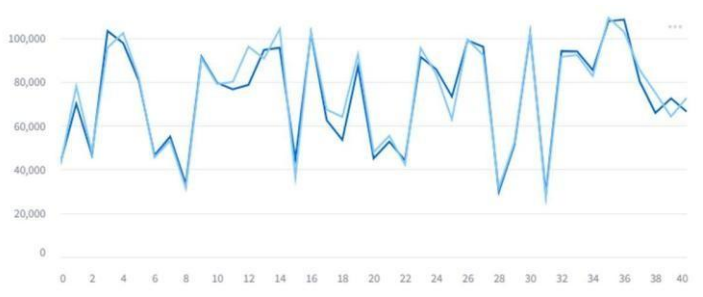
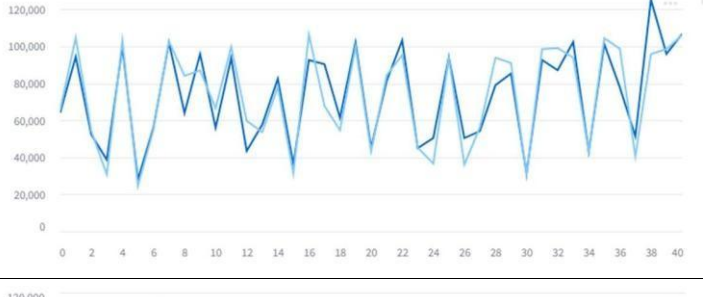
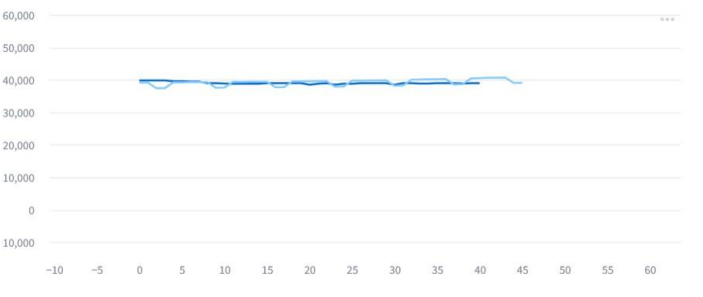
Num_epochs	patience	Epoch thực tế	R ² score	MSE	MAE
50	5	50	0.9255989193916321	764.05725097656245	1778472.25
50	10	50	0.9585139751434326	632.8076171875	991676.5
50	20	50	0.9685295224189758	542.3862915039062	752265.5
100	5	73	0.9809367060661316	457.1025695800781	455686.8125
100	10	100	0.9782314896583557	395.4391784667969	520351.21875
100	20	89	0.9610971212387085	703.7343139648438	929929.4375
200	5	54	0.9810549020767212	408.28082275390625	452861.0
200	10	69	0.9866098761558533	376.36572265625	320074.90625
200	20	110	0.972932755947113	485.8993835449219	647011.5

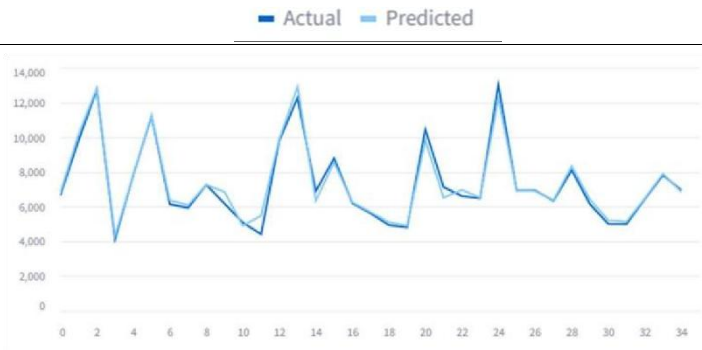
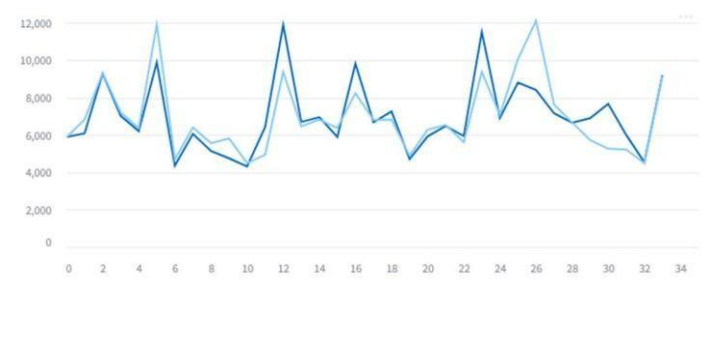
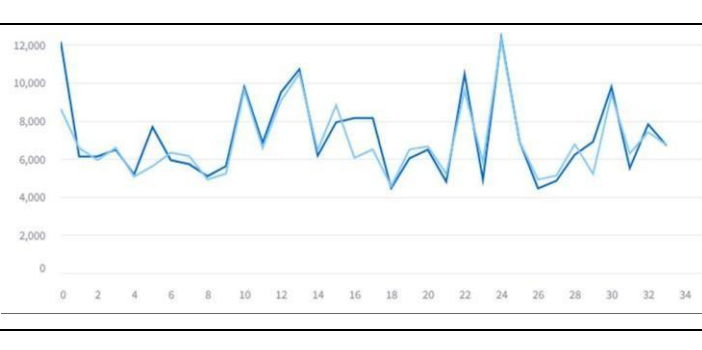
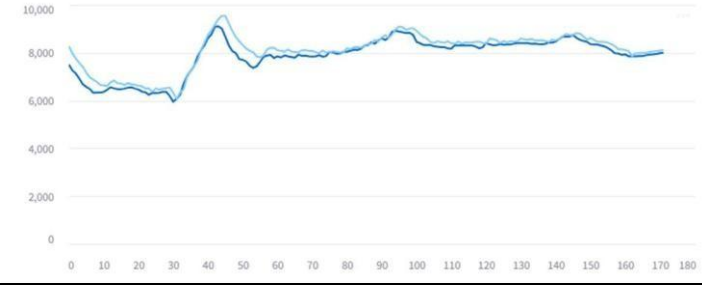
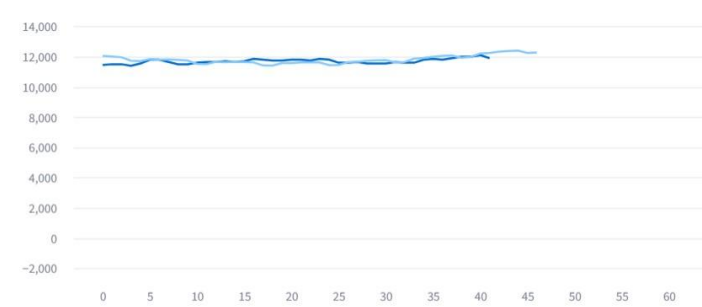
So sánh và nhận xét

- Sự kết hợp num_epochs = 200 và patience = 10 là tối ưu nhất dựa trên dữ liệu này, mang lại R² cao nhất, MSE và MAE thấp nhất.
- Nên chọn num_epochs lớn (như 200) để mô hình có đủ thời gian học.
- Patience nhỏ (5) giúp dừng sớm, tiết kiệm thời gian nhưng có thể dừng trước khi mô hình đạt hiệu suất tối ưu. Patience lớn (20) cho phép huấn luyện lâu hơn, nhưng có thể dẫn đến overfitting, như trong trường hợp num_epochs = 200, patience = 20 (R² giảm, MSE và MAE tăng). Chọn patience vừa phải (như 10) để cân bằng giữa dừng sớm và đảm bảo mô hình học đầy đủ, tránh overfitting.

CHƯƠNG 7. KẾT QUẢ MÔ HÌNH

Chạy thử nghiệm mô hình để so sánh trên mã chứng khoán VIC với chế độ dự đoán 30 ngày kế tiếp

Model	R ² score	MSE	MAE	<div> <div>Actual</div> <div>Predicted</div> </div> 
LinearRegression	0.98048	14439614.24159	2636.04994	
RandomForestRegressor	0.68406	187283758.53022	11997.45696	
KneighborsRegressor	0.91489	54118607.93268	5140.57073	
LSTM	0.47343	1438611.125	913.5336	
Prophet	-7.690690818398311	1005245.6411239543	867.972000488132	

Model	R ² score	MSE	MAE	
LinearRegression	0.97636	126815.27789	247.15244	
RandomForestRegressor	0.61572	1347996.38702	751.45784	
KneighborsRegressor	0.78795	948911.62588	633.11764	
LSTM	0.86266	81314.10156	215.91583	
Prophet	-1.202269698575193	185.3687651912898	54660.26634086556	

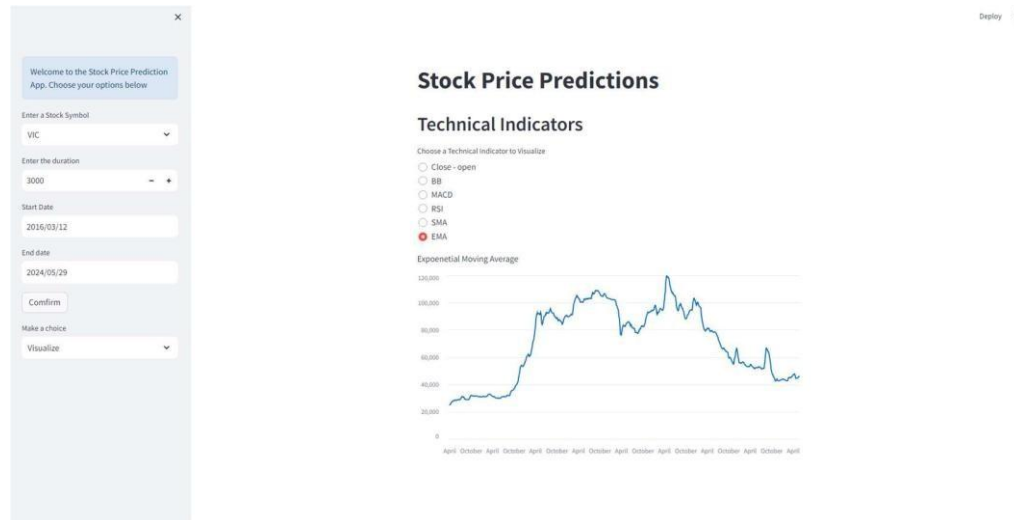
Chạy thử nghiệm mô hình để so sánh trên mã chứng khoán BPC với chế độ dự đoán 30 ngày kế tiếp

CHƯƠNG 8. TRIỂN KHAI ỨNG DỤNG

7.1 Công nghệ sử dụng

Sử dụng thư viện streamlit của python để xây dựng web app.

Streamlit là một thư viện mã nguồn mở trong Python được sử dụng để tạo giao diện người dùng web cho các ứng dụng dữ liệu và machine learning một cách nhanh chóng và dễ dàng.



7.2 Chức năng

7.2.1 Lựa chọn tra cứu dữ liệu theo ngày và mã cổ phiếu công ty

Từ hơn 1000 mã có sẵn trong database, chọn khoảng thời gian muốn tra cứu lịch sử chứng khoán (ngày bắt đầu, ngày kết thúc, khoảng thời gian)

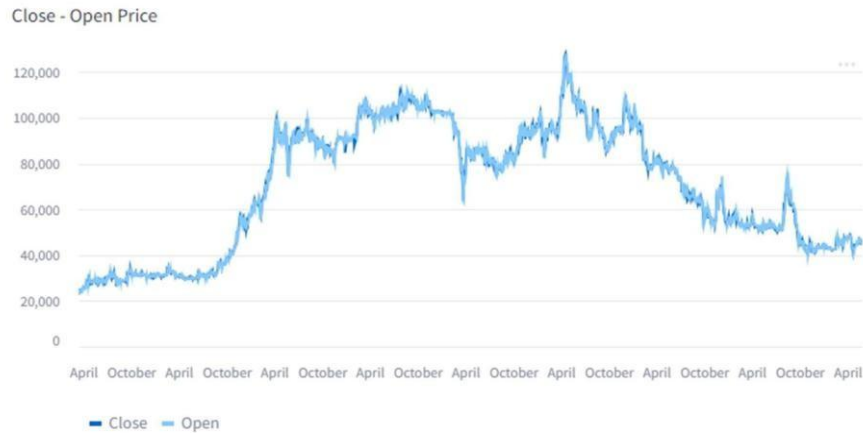
Chọn chế độ của app (Visualize, Recent data, Predict)

Người dùng có thể lựa chọn mã chứng khoán

7.2.2 Trực quan hóa dữ liệu

Chức năng này cung cấp 6 chế độ đồ thị để trực quan hóa dữ liệu giá cổ phiếu

1. **Close – Open** : hiển thị giá mở - đóng trong ngày



2. **BB**: hiển thị dải boilinger bands



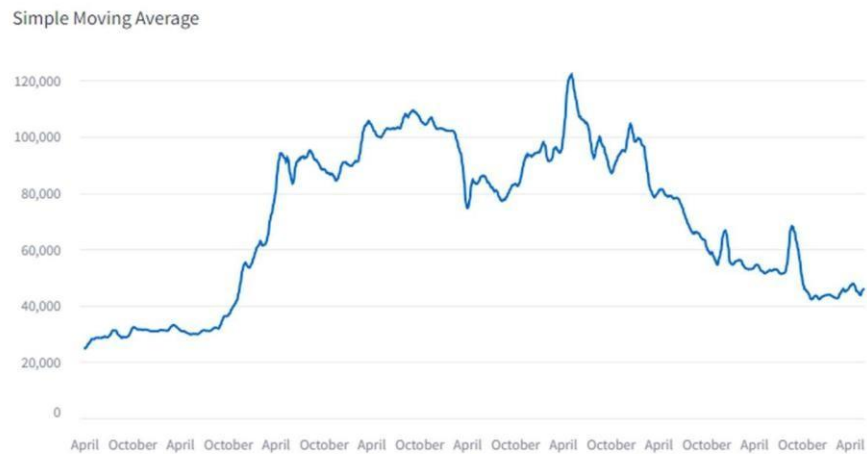
3. **MACD (Moving Average Convergence Divergence)**: hiển thị chỉ báo MACD (chỉ số xác định sự thay đổi trong sức, hướng và thời gian của xu hướng giá cổ phiếu. Khi đường MACD cắt lên trên đường tín hiệu là tín hiệu mua; ngược lại, khi cắt xuống dưới, là tín hiệu bán.



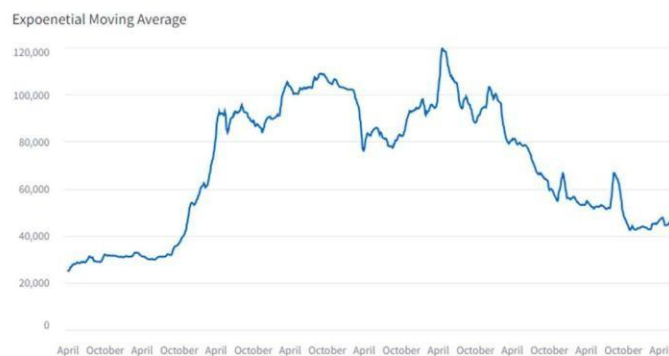
4. **RSI (Relative Strength Indicator):** Báo động lượng đo lường tốc độ và sự thay đổi của chuyển động giá, RSI trên 70 là quá mua (overbought), và dưới 30 là quá bán (oversold).



5. **SMA (Simple Moving Average):** chỉ báo kỹ thuật tính toán giá trung bình của một cổ phiếu trong một khoảng thời gian nhất định. Nó giúp làm mượt các dao động giá và xác định xu hướng dài hạn.



6. **EMA (Exponential Moving Average):** một loại trung bình động tương tự như SMA, nhưng EMA phản ứng nhanh hơn với các thay đổi giá gần đây.



7.2.3 Hiển thị dữ liệu danh sách công ty và lịch sử cổ phiếu

×

Welcome to the Stock Price Prediction App. Choose your options below

Enter a Stock Symbol

BPC

Enter the duration

3000

Start Date

2016/03/12

End date

2024/05/29

Confirm

Make a choice

Recent Data

Company Data

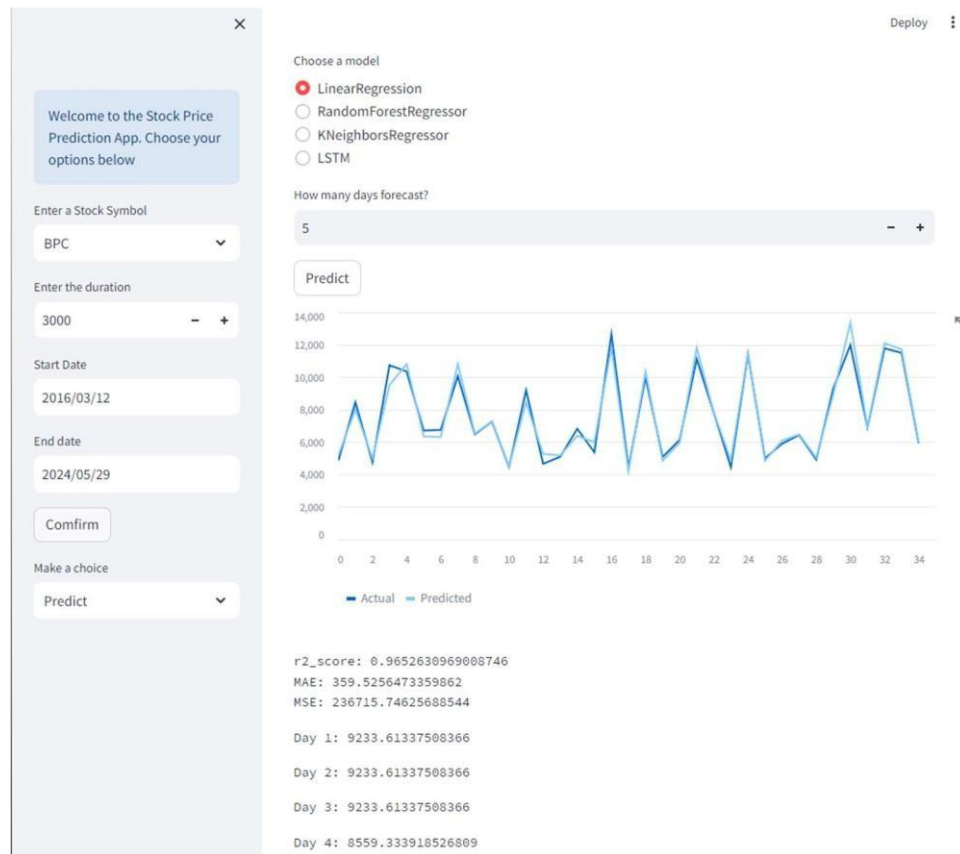
	ticker	organName	organTypeCode	comGroupCode
0	A32	CTCP 32	1	UPCOM
1	AAA	CTCP Nhựa An Phát Xanh	1	HOSE
2	AAH	CTCP Hợp Nhất	1	UPCOM
3	AAM	CTCP Thủy sản MeKong	1	HOSE
4	AAS	CTCP Chứng khoán SmartInvest	4	UPCOM
5	AAT	CTCP Tập Đoàn Tiên Sơn Thanh Hóa	1	HOSE
6	AAV	CTCP AAV Group	1	HNX
7	ABB	Ngân hàng TMCP An Bình	2	UPCOM
8	ABC	CTCP Truyền thông VMG	1	UPCOM
9	ABI	CTCP Bảo hiểm Ngân hàng Nông nghiệp Việt Nam	3	UPCOM

Stock Data

Time	Open	High	Low	Close	Volume	Ticker	Change	upper_band	lower_band
2016-04-25	4,520	4,530	4,460	4,490	10,500	BPC	0.0135	4,391.8992	3,943.1008
2016-04-26	4,390	4,460	4,400	4,400	7,930	BPC	-0.02	4,426.3081	3,936.6919
2016-04-27	4,420	4,800	4,400	4,800	18,800	BPC	0.0909	4,577.5207	3,853.4793
2016-04-28	4,800	4,840	4,710	4,800	12,250	BPC	0	4,688.7482	3,810.2518
2016-04-29	4,800	4,900	4,740	4,800	19,000	BPC	0	4,779.0512	3,787.9488
2016-05-04	4,930	5,240	4,840	5,120	17,800	BPC	0.0667	4,941.9168	3,725.0832
2016-05-05	5,170	5,240	4,930	5,080	6,100	BPC	-0.0078	5,062.1548	3,700.8452
2016-05-06	4,800	5,080	4,800	5,080	30,500	BPC	0	5,162.9705	3,696.0295
2016-05-09	4,800	4,840	4,770	4,840	27,700	BPC	0.0473	5,202.8881	3,730.1119

Chức năng này hiển thị danh sách các công ty và lịch sử giá cổ phiếu của công ty đã chọn.

7.2.4 Chức năng dự báo giá cổ phiếu



Chức năng này cho phép lựa chọn các mô hình đã triển khai (Linear Regression, Random Forest Regressor, K-Neighbors Regressor, LSTM) để dự báo giá cổ phiếu trong một số ngày tới và hiển thị kết quả đánh giá mô hình.

7.2.5 Extension hiển thị tin tức về mã cổ phiếu đã thu thập được

Chức năng này hiển thị các tin tức liên quan đến mã cổ phiếu đã được thu thập từ các nguồn tin tức tài chính uy tín. Chúng ta sẽ sử dụng extension để hiển thị tin tức.

TLT

Tên công ty: Thăng Long Viglacera
Ngành: Construction & Materials
Năm thành lập: 2003
Số nhân viên: 347
<http://www.viglacera-thanglong.com.vn>

TLT: Thay đổi nhân sự
Nguồn: HNX
Ngày đăng: 1 tháng 4, 2024
Giá: 13900 VND
Thay đổi: 0 (0.00%)

TLT: Nghị quyết Đại hội đồng cổ đông thường niên năm 2024
Nguồn: HNX
Ngày đăng: 1 tháng 4, 2024
Giá: 13900 VND
Thay đổi: 0 (0.00%)

TLT: Tài liệu họp Đại hội đồng cổ đông
Nguồn: HNX
Ngày đăng: 8 tháng 3, 2024
Giá: 12600 VND
Thay đổi: 500 (4.10%)

TLT: Ngày đăng ký cuối cùng Đại hội đồng cổ đông

Giá: 13600 VND
Thay đổi: 0 (0.00%)

TLT: Thông báo ngày đăng ký cuối cùng để thực hiện quyền tham dự Đại hội đồng cổ đông thường niên năm 2024
Nguồn: HNX
Ngày đăng: 7 tháng 2, 2024
Giá: 13600 VND
Thay đổi: 0 (0.00%)

TLT: Báo cáo tài chính năm 2023
Nguồn: HNX
Ngày đăng: 31 tháng 1, 2024
Giá: 13600 VND
Thay đổi: 0 (0.00%)

TLT: Báo cáo quản trị công ty năm 2023
Nguồn: HNX
Ngày đăng: 31 tháng 1, 2024
Giá: 13600 VND
Thay đổi: 0 (0.00%)

TLT: Ngày đăng ký cuối cùng trả cổ tức bằng tiền mặt
Nguồn: HNX
Ngày đăng: 28 tháng 7, 2023
Giá: 14700 VND
Thay đổi: 0 (0.00%)

CHƯƠNG 9. KẾT LUẬN

8.1 Kết luận

Dự án phát triển hệ thống crawl dữ liệu tài chính tích hợp dự đoán và dự báo giá cổ phiếu là một ứng dụng quan trọng, mang tính thách thức cao trong lĩnh vực tài chính và học máy. Trong dự án này, nhóm đã xây dựng một ứng dụng cho phép người dùng tiếp cận dữ liệu tài chính một cách trực quan thông qua các biểu đồ đa dạng, tham khảo tin tức liên quan đến mã cổ phiếu quan tâm, và đặc biệt là áp dụng các mô hình học máy và học sâu như hồi quy tuyến tính, LSTM, và Prophet để dự đoán giá cổ phiếu. Hệ thống cung cấp khả năng so sánh hiệu quả giữa các mô hình, giúp người dùng đánh giá ưu và nhược điểm của từng mô hình tùy theo mục đích sử dụng. Bằng cách kết hợp khéo léo các mô hình dự đoán với các tính năng thu thập và trực quan hóa dữ liệu, ứng dụng trở thành công cụ hữu ích cho người dùng có chuyên môn.

Tuy nhiên, dự đoán giá chứng khoán vẫn là một bài toán phức tạp do thị trường tài chính chịu ảnh hưởng từ nhiều yếu tố khó lường như kinh tế, chính trị, và tâm lý thị trường. Do hạn chế về thời gian và mức độ hiểu biết chưa toàn diện về bài toán, nhóm chưa thể tích hợp thêm nhiều mô hình đa dạng hơn hoặc tối ưu hóa hoàn toàn hệ thống. Một số sai sót trong quá trình phát triển cũng có thể xảy ra. Dù vậy, hệ thống này đặt nền tảng cho việc ứng dụng trí tuệ nhân tạo trong tài chính, mở ra tiềm năng cải tiến và phát triển trong tương lai.

8.2 Hướng phát triển trong tương lai

Trong tương lai, chúng em sẽ áp dụng và lựa chọn thêm những thuật toán học máy, học sâu phù hợp hơn, sử dụng thêm nhiều kỹ thuật phân tích, thêm nhiều trường dữ liệu hơn để cải tiến. Đồng thời, kết hợp phân loại tiêu đề của bài báo liên quan tới mã chứng khoán để dự đoán giá một cách tốt hơn.

Những hướng phát triển này không chỉ giúp cải thiện độ chính xác và hiệu quả của dự án mà còn mở rộng phạm vi ứng dụng, từ đó đem lại những giá trị thực tiễn cao hơn trong việc dự đoán và phân tích thị trường chứng khoán. Chúng em hy vọng rằng các cải tiến này sẽ đóng góp tích cực vào lĩnh vực tài chính và đầu tư.