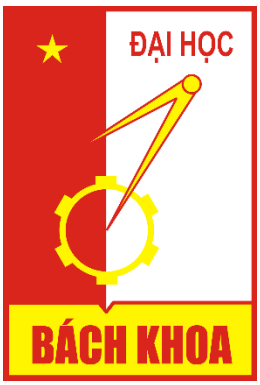
ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO BÀI TẬP LỚN

ĐỀ TÀI: DỰ ĐOÁN GIÁ CHỨNG KHOÁN

Nhóm 8 Mã lớp: 128743

Giáo Viên Hướng Dẫn: PGS. TS. Thân Quang Khoát

Sinh viên thực hiện:

Họ và tên	MSSV
Bùi Đức Chế	20183694
Bùi Văn Sơn	20183820
Nguyễn Hoàng Long	20183791
Phương Trung Kiên	20183776

Hà Nội, tháng 1 năm 2022

PHÂN CÔNG CÔNG VIỆC

Họ và tên	Công việc chính
Bùi Đức Chế	Code Crawl và thiết kế models
Bùi Văn Sơn	Tiền xử lý dữ liệu và xây dựng models
Phương Trung Kiên	Phân tích dữ liệu và làm báo cáo
Nguyễn Hoàng Long	Phân tích dữ liệu và crawl dữ liệu

LỜI MỞ ĐẦU

Ngày nay, cùng với sự phát triển vượt bậc của con người về mọi mặt thì trí tuệ nhân tạo ngày càng được ứng dụng nhiều trong tất cả mọi lĩnh vực đời sống. Nó không chỉ làm cuộc sống trở lên tốt mà còn xử lý được những công việc khó khan tốn nhiều thơi gian. Học máy là một lĩnh vực thuộc trí tuệ nhân tạo. Gần đây, học máy đang dẫn đầu xu thế và tạo nên những thay đổi vượt bậc trong trí tuệ nhân tạo nói chung và công nghệ thông tin nói riêng.

Trong môn học "Nhập môn học máy và khai phá dữ liệu" kỳ này, nhóm 8 thực hiện một project liên quan đến học máy giúp hiểu hơn một bài toán học máy cũng như tạo ra một giải pháp học máy có ứng dụng rộng rãi trong thực tế. Nhóm chúng em đã chọn chủ đề dự đoán giá đóng cửa chứng khoán. Đây là 1 chủ đề khá hay nhất là đang trong thời buổi dịch bênh vẫn đang ảnh hưởng nghiêm trọng đến nền kinh tế nhất là trongg 1 số thị trường như chứng khoán, tiền ảo, bất động sản, …. Do thời gian và khả năng có hạn nên không thể tránh khỏi sai sót. Vì vậy nhóm chúng em thầy có thể nhận xét và đưa ra ý kiến để chúng em hoàn thiện hơn bài báo cáo của nhóm mình. Cuối cùng, nhóm 8 xin chân thành cảm ơn thầy **Thân Quang Khoát** đã hướng dẫn chúng em trong suốt thời gian làm báo cáo này.

Mục Lục

PHẦN I: ĐẶT VẤN ĐỀ

1. Đặt vấn đề

Thị trường chứng khoán được biết đến là dễ biến động, luôn thay đổi khó lường và phi tuyến tính. Dự đoán giá cổ phiếu chính xác là vô cùng khó khăn vì phụ thuộc vào nhiều yếu tố vĩ mô và vi mô, chẳng hạn như yếu tố chính trị, bối cảnh kinh tế trong nước và kinh tế toàn cầu, các cú sốc bất ngờ (Covid-19, thiên tai, mất mùa,...), tình hình hoạt động tài chính của công ty,…

Tuy nhiên, tất cả những điều này cũng có nghĩa là có rất nhiều dữ liệu để tìm ra các pattern của chuỗi chứng khoán. Vì vậy, các chuyên gia phân tích tài chính, các nhà nghiên cứu và các nhà khoa học dữ liệu đang tiếp tục khám phá các kỹ thuật phân tích để phát hiện xu hướng và qui luật vận động của thị trường chứng khoán. Điều này đã làm nảy sinh khái niệm giao dịch theo thuật toán, sử dụng các chiến lược giao dịch tự động, được lập trình sẵn để thực hiện các lệnh.

Để phân tích được dữ liệu chứng khoán chúng ta có thể sử dụng cả phương pháp luận tài chính định lượng truyền thống và thuật toán máy học để dự đoán giá cổ phiếu. Một số phương pháp cụ thể trong việc phân tích dữ liệu chứng khoán chúng ta có thể lựa chọn chẳng hạn:

- Phân tích cổ phiếu theo phương pháp phân tích cơ bản và phương pháp phân tích kỹ thuật
- Dự đoán giá cổ phiếu bằng kỹ thuật Moving Average.
- Áp dụng các thuật toán RNN như LSTM, GRU.
- Sử dụng các lớp mô hình dự báo timeseries truyền thống như AR, MA, ARIMA, SARIMA, ARIMAX, GARCH…

Trong bài tập lớn môn học này chúng em sử dụng thuật toán LSTM để phân tích dữ liệu chứng khoán tại Việt Nam.

2. Mô tả bài toán

Bài toán: Dự đoán giá chứng khoán thông qua dữ liệu từ quá khứ.

- Đầu vào: Các trường dữ liệu ảnh hưởng tới giá của một mã chứng khoán.
- Đầu ra: Giá đóng cửa của một mã chứng khoán trong một ngày cụ thể.
- Tập dữ liệu: tập dữ liệu được sử dụng là dữ liệu lịch sử giao dịch của các mã chứng khoán được thu thập bằng cách crawl dữ liệu trên trang cafef.vn

3. Định hướng công việc

Bước 1: Thu thập dữ liệu.

Bước 2: Phân tích dữ liệu.

Bước 3: Xây dựng mô hình học máy.

PHẦN II: THU THẬP DỮ LIỆU (CRAWL DATA)

Dữ liệu của các mã chứng khoán được thể hiện dưới dạng bảng như hình dưới đây(link: https://s.cafef.vn/Lich-su-giao-dich-VNA-1.chn) và để có thể lấy được chúng chúng ta có thể dùng API để lấy dữ liệu dưới dạng đơn giản hoặc trích xuất HTML để thu được các trường dữ liệu của bảng.

Mã VNA Từ ngày Đến ngày Xem Xem toàn thị trường theo phiên

Ngày	Giá điều chỉnh	Giá đóng cửa	Giá bình quân	Thay đổi (+/-%)	GD khớp lệnh		GD thỏa thuận		Giá tham chiếu	Giá mở cửa	Giá cao nhất	Giá thấp nhất
					KL	GT	KL	GT				
31/12/2021	31.90	31.90	31.90	-0.80 (-2.45 %)	115,301	3,712,672,600	0	0	32.70	32.80	33.30	31.60
30/12/2021	32.70	32.70	32.70	1.50 (4.81 %)	171,500	5,604,460,000	0	0	31.20	31.7	33.50	31.70
29/12/2021	31.70	31.70	31.70	1.10 (3.59 %)	228,000	7,122,110,000	0	0	30.60	30.50	32.00	30.50
27/12/2021	31.10	31.10	31.10	0.40 (1.30 %)	79,005	2,446,594,000	0	0	30.70	31.0	31.60	30.70
24/12/2021	30.90	30.90	30.90	0.00 (0.00 %)	127,500	3,913,800,000	0	0	30.90	30.50	31.50	30.00
23/12/2021	30.50	30.50	30.50	-1.80 (-5.57 %)	415,300	12,835,178,300	0	0	32.30	32.1	32.20	30.00
22/12/2021	32.20	32.20	32.20	-0.40 (-1.23 %)	268,669	8,683,348,900	0	0	32.60	32.60	32.90	32.00
21/12/2021	32.90	32.90	32.90	-0.30 (-0.90 %)	246,631	8,039,036,700	0	0	33.20	33.0	33.50	32.00
20/12/2021	33.00	33.00	33.00	-2.10 (-5.98 %)	400,861	13,326,876,900	0	0	35.10	34.80	34.80	33.00
17/12/2021	34.80	34.80	34.80	-2.10 (-5.69 %)	260,589	9,145,045,000	0	0	36.90	36.4	36.40	34.60
16/12/2021	36.40	36.40	36.40	0.30 (0.83 %)	265,357	9,780,581,900	0	0	36.10	37.00	38.30	36.00
15/12/2021	36.60	36.60	36.60	3.20 (9.58 %)	749,649	27,086,177,300	0	0	33.40	33.8	37.50	33.40
14/12/2021	33.80	33.80	33.80	0.70 (2.11 %)	296,030	9,875,600,000	0	0	33.10	33.00	34.00	32.60
13/12/2021	33.00	33.00	33.00	0.20 (0.61 %)	135,526	4,487,276,500	0	0	32.80	33.0	33.80	32.80
10/12/2021	32.80	32.80	32.80	-0.60 (-1.80 %)	112,640	3,696,274,000	0	0	33.40	33.40	33.40	32.50
09/12/2021	33.40	33.40	33.40	0.70 (2.14 %)	127,191	4,248,383,000	0	0	32.70	32.5	34.00	32.50
08/12/2021	32.70	32.70	32.70	0.20 (0.62 %)	97,900	3,196,500,000	0	0	32.50	33.00	33.10	32.30
07/12/2021	32.80	32.80	32.80	0.10 (0.31 %)	127,404	4,135,342,600	0	0	32.70	32.0	33.00	32.00
06/12/2021	32.20	32.20	32.20	-0.90 (-2.72 %)	174,710	5,710,207,000	0	0	33.10	32.60	33.40	31.50
03/12/2021	32.60	32.60	32.60	-1.20 (-3.55 %)	110,027	3,643,343,000	0	0	33.80	33.6	34.00	32.60

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 >

1. Phân tích, lựa chọn phương pháp crawl.

Để crawl dữ liệu trên chúng ta có thể sử dụng một trong 2 cách là dùng API lấy trực tiếp dữ liệu của 1 mã chứng khoán hoặc thông qua việc trích xuất HTML lấy dữ liệu dạng bảng của mã chứng khoán.Việc crawl thông qua API sẽ được ưu tiên hơn vì dữ liệu thu được tức thời và có thể dùng phân tích được luôn.

Do không tìm được API public của trang này và nhận thấy lượng dữ liệu lấy cũng không quá lớn nên chúng em quyết định crawl thông qua việc trích xuất HTML.

Để crawl xong một mã chứng khoán ta cần lấy được tất cả các bảng dữ liệu của mã chứng khoán đó và để chuyển sang bảng tiếp theo chúng ta cần thực thi javascript của trang.

2. Lựa chọn công cụ crawl.

Qua tìm hiểu nhóm thấy có các công cụ crawl website phổ biến như sau:

- Requests + BeautifulSoup: Sử dụng requests để lấy dữ liệu html sau đó dùng BeautifulSoup để trích xuất dữ liệu HTML ,cách này rất đơn giản và hiệu quả nhưng vấn đề gặp phải là chúng ta không thể khó có thể dùng request để dữ liệu các bảng tiếp theo của mã chứng khoán đó.
- Scrapy: Hoạt động tương tự như BeautifulSoup được tối ưu để có thể crawl được dữ liệu nhanh chóng và ít tiêu tốn tài nguyên hơn nhưng cũng gặp phải vấn đề về việc lấy dữ liệu của các trang tiếp theo.Ở đây ta có thể kết hợp thêm 1 công cụ khác là Splash để thực thi LUA script chuyển qua các bảng dữ liệu tiếp theo.
- Selenium: Khác với 2 công cụ trên selenium được tạo ra với mục đích không phải để crawl website mà để kiểm thử tự động website. Tuy nhiên, với các chức năng có sẵn của selenium chứng ta cũng có thể đồng thời trích xuất dữ liệu HTML cũng như chuyển sang các bảng dữ liệu khác như một người dùng. Do hoạt động bằng cách giả lập người dùng thật nên có thể crawl được các trang web có sử dụng các cơ chế ngăn chặn crawl.Tuy nhiên, selenium có nhược điểm là tốc độ crawl không cao cũng như tiêu tốn rất nhiều tài nguyên của máy tính.

Do lượng dữ liệu crawl cũng không lớn cũng như chưa có kinh nghiệm gì trong việc crawl dữ liệu nên nhóm quyết định sẽ crawl dữ liệu bằng selenium với đặc điểm là tiêu thụ nhiều tài nguyên tốc độ crawl thấp nhưng do là công cụ kiểm thử nên có thể hiển thị trực quan khi dữ liệu đang được crawl.

3. Tiến hành crawl dữ liệu

Sau khi xác định được trang web để crawl, selenium có thể tìm các thành phần của trang web đó và có thể tương tác hoặc lấy dữ liệu của thành phần đó. Để lấy dữ liệu lịch sử của một mã chứng khoán ta sẽ bắt đầu với một đường dẫn có cấu trúc như sau:

```
browser.get("https://s.cafef.vn/Lich-su-giao-dich-"+str(MaCK)+"-1.chn")
```

Dữ liệu được crawl được lấy thông qua XPATH: các hàng dữ liệu đều có id có cấu trúc phần đầu giống nhau nên ta dễ dàng lấy được toàn bộ chúng sau đó lấy dữ liệu của từng thẻ <td> bên trong.

```
stonks =browser.find_elements(By.XPATH, '//tr[starts-with(@id,"ctl00_ContentPlaceHolder1_ctl03_rptData")]')
for stonk in stonks:
    try:
        # print(stonk.text)
        data=stonk.find_elements(By.XPATH,'td')
        datastonk= {'Ngày':            data[0].text,
                    'Giá điều chỉnh':  data[1].text,
                    'Giá đóng cửa':    data[2].text,
                    'Giá bình quân':   data[3].text,
                    'thay đổi':        data[4].text,
                    'KL1':             data[6].text,
                    'GT1':             data[7].text,
                    'KL2':             data[8].text,
                    'GT2':             data[9].text,
                    'Giá tham chiếu':  data[10].text,
                    'Giá mở cửa':      data[11].text,
                    'Giá cao nhất':    data[12].text,
                    'Giá thấp nhất':   data[13].text
                   }
```

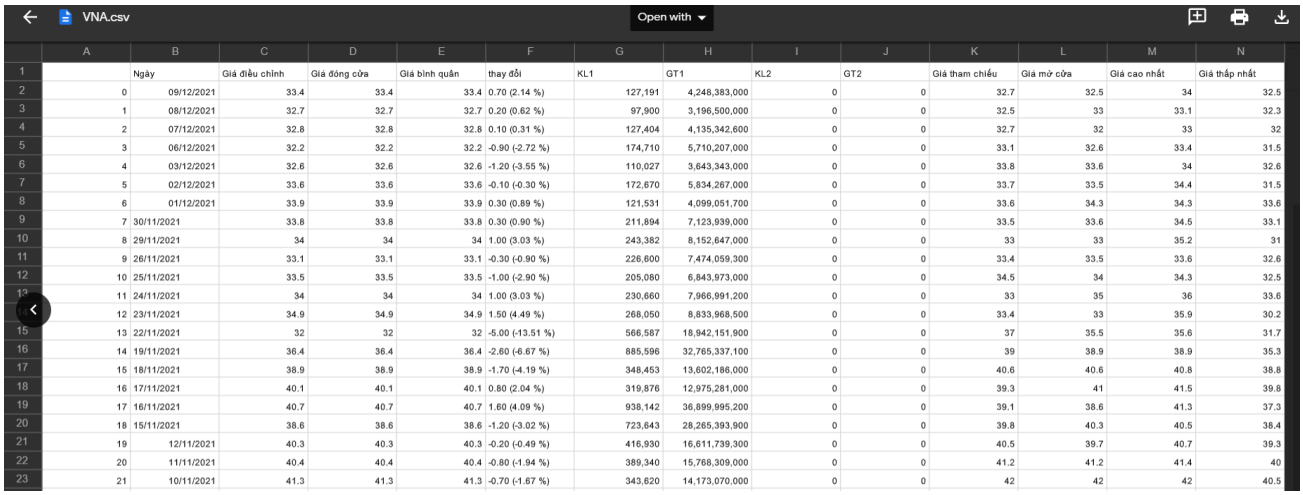
Sau khi lấy dữ liệu xong một trang ta chuyển sang các trang kế tiếp thông qua element chuyển trang:

```
nextpage = browser.find_element(By.XPATH, "(//a[contains(text(),'>')])[2]")  # nut chuyen trang
nextpage.click()  # Trang ke tiep
```

4. **Một số vấn đề gặp phải trong quá trình crawl và cách giải quyết**
   - Trang web không load được hoặc load rất chậm tại một số giai đoạn vào những ngày chứng khoán giao dịch nên việc triển khai crawl chỉ được thực hiện vào những ngày cuối tuần.
   - Các thành phần tương tác chuyển trang không hoạt động. Có thể là cơ chế chống crawl của trang này vì trong quá trình thử tương tác với trang như người dùng bình thường thì càng chuyển về các trang đằng sau thì khi click sang trang khác sẽ phải thực hiện rất nhiều lần mới hoàn thành được.Để giải quyết vấn đề này ta thêm cơ chế kiểm tra trang hiện tại và trang kế tiếp nếu tương tác chuyển sang không thành công hay không hợp lệ thì thực hiện lại sao cho đúng.Tuy nhiên nếu vẫn xảy ra ngoại lệ thì sẽ dừng lại để xử lý bằng tay thông qua giao diện.
   - Một số dữ liệu chứng khoán có cấu trúc bảng dữ liệu khác nên phải sửa lại code phần này.
   - Thỉnh thoảng trang có nhảy ra quảng cáo ngay giữa màn hình gây gián đoạn crawl. Hiện tại thì nhóm xử lý bằng tay thông qua việc giám sát lúc crawl.

5. Kết quả

Mặc dù trong quá trình crawl vẫn phải xử lý bằng tay và cần giám sát nhưng việc thu thập dữ liệu cũng đã được hoàn thành.Dữ liệu được lưu vào file csv ,đầy đủ các trường dữ liệu như trên web và sau khi kiểm tra thì không thấy có dữ liệu NULL nên có thể triển khai các bước tiếp theo luôn.



	A	Ngày	Giá điều chỉnh	Giá đóng cửa	Giá bình quân	thay đổi	KL1	GT1	KL2	GT2	Giá tham chiếu	Giá mở cửa	Giá cao nhất	Giá thấp nhất
2	0	09/12/2021	33.4	33.4	33.4	0.70 (2.14 %)	127,191	4,248,383,000	0	0	32.7	32.5	34	32.5
3	1	08/12/2021	32.7	32.7	32.7	0.20 (0.62 %)	97,900	3,196,500,000	0	0	32.5	33	33.1	32.3
4	2	07/12/2021	32.8	32.8	32.8	0.10 (0.31 %)	127,404	4,135,342,600	0	0	32.7	32	33	32
5	3	06/12/2021	32.2	32.2	32.2	-0.90 (-2.72 %)	174,710	5,710,207,000	0	0	33.1	32.6	33.4	31.5
6	4	03/12/2021	32.6	32.6	32.6	-1.20 (-3.55 %)	110,027	3,643,343,000	0	0	33.8	33.6	34	32.6
7	5	02/12/2021	33.6	33.6	33.6	-0.10 (-0.30 %)	172,670	5,834,267,000	0	0	33.7	33.5	34.4	31.5
8	6	01/12/2021	33.9	33.9	33.9	0.30 (0.89 %)	121,531	4,099,051,700	0	0	33.6	34.3	34.3	33.6
9	7	30/11/2021	33.8	33.8	33.8	0.30 (0.90 %)	211,894	7,123,939,000	0	0	33.5	33.6	34.5	33.1
10	8	29/11/2021	34	34	34	1.00 (3.03 %)	243,382	8,152,647,000	0	0	33	33	35.2	31
11	9	26/11/2021	33.1	33.1	33.1	-0.30 (-0.90 %)	226,600	7,474,059,300	0	0	33.4	33.5	33.6	32.6
12	10	25/11/2021	33.5	33.5	33.5	-1.00 (-2.90 %)	205,080	6,843,973,000	0	0	34.5	34	34.3	32.5
13	11	24/11/2021	34	34	34	1.00 (3.03 %)	230,660	7,966,991,200	0	0	33	35	36	33.6
14	12	23/11/2021	34.9	34.9	34.9	1.50 (4.49 %)	268,050	8,833,968,500	0	0	33.4	33	35.9	30.2
15	13	22/11/2021	32	32	32	-5.00 (-13.51 %)	566,587	18,942,151,900	0	0	37	35.5	35.6	31.7
16	14	19/11/2021	36.4	36.4	36.4	-2.60 (-6.67 %)	885,596	32,765,337,100	0	0	39	38.9	38.9	35.3
17	15	18/11/2021	38.9	38.9	38.9	-1.70 (-4.19 %)	348,453	13,602,186,000	0	0	40.6	40.6	40.8	38.8
18	16	17/11/2021	40.1	40.1	40.1	0.80 (2.04 %)	319,876	12,975,281,000	0	0	39.3	41	41.5	39.8
19	17	16/11/2021	40.7	40.7	40.7	1.60 (4.09 %)	938,142	36,899,995,200	0	0	39.1	38.6	41.3	37.3
20	18	15/11/2021	38.6	38.6	38.6	-1.20 (-3.02 %)	723,643	28,265,393,900	0	0	39.8	40.3	40.5	38.4
21	19	12/11/2021	40.3	40.3	40.3	-0.20 (-0.49 %)	416,930	16,611,739,300	0	0	40.5	39.7	40.7	39.3
22	20	11/11/2021	40.4	40.4	40.4	-0.80 (-1.94 %)	389,340	15,768,309,000	0	0	41.2	41.2	41.4	40
23	21	10/11/2021	41.3	41.3	41.3	-0.70 (-1.67 %)	343,620	14,173,070,000	0	0	42	42	42	40.5

Thông tin cơ bản về các trường thuộc tính

Tên cột	Ý nghĩa
Ngày	Thời gian
Giá điều chỉnh	Điều chỉnh giá của 1 cổ phiếu là sự giảm giá của cổ phiếu đó đi 10% đến 20% so với giá trị ở mức cao gần đây của cổ phiếu đó
Giá đóng cửa	Mức giá của cổ phiếu đó khi kết thúc phiên giao dịch của ngày hôm đó
Giá bình quân	Giá bình quân của cổ phiếu đó
Thay đổi	Giá trị thay đổi so với ngày hôm trước
KL1	Khối lượng cổ phiếu giao dịch khớp lệnh
KL2	Khối lượng cổ phiếu giao dịch thỏa thuận
GT1	Giá trị giao dịch của cổ phiếu khớp lệnh
GT2	Giá trị giao dịch của cổ phiếu thỏa thuận
Giá tham chiếu	Là mức giá đóng cửa tại phiên giao dịch gần nhất trước đó (trừ các trường hợp đặc biệt)
Giá mở cửa	Mức giá của cổ phiếu đó khi mở cửa phiên giao dịch của ngày hôm nay
Giá cao nhất	Biểu thị biên độ giá cao nhất của cổ phiếu trong phiên giao dịch
Giá thấp nhất	Biểu thị biên độ giá thấp nhất của cổ phiếu trong phiên giao dịch

Link tập dữ liệu nhóm đã crawl được:
https://drive.google.com/drive/u/5/folders/1BxK5xO3Lrd47jpxnHQMn5GHBg1PLQi6a?fbclid=IwAR0INuedePxX8DGqQSemAz_Trjnq4auGiIWffvUETZZqXbU9b8SEfr_l8SA

PHẦN III: PHÂN TÍCH DỮ LIỆU

1. Tiền xử lý dữ liệu

Dữ liệu đầu vào:

	Unnamed: 0	Ngày	Giá điều chỉnh	Giá đóng cửa	Giá bình quân	thay đổi	KL1	GT1	KL2	GT2	Giá tham chiếu	Giá mở cửa	Giá cao nhất	Giá thấp nhất
0	0	09/12/2021	33.4	33.4	33.40	0.70 (2.14 %)	127,191	4,248,383,000	0	0	32.70	32.5	34.0	32.5
1	1	08/12/2021	32.7	32.7	32.70	0.20 (0.62 %)	97,900	3,196,500,000	0	0	32.50	33.0	33.1	32.3
2	2	07/12/2021	32.8	32.8	32.80	0.10 (0.31 %)	127,404	4,135,342,600	0	0	32.70	32.0	33.0	32.0
3	3	06/12/2021	32.2	32.2	32.20	-0.90 (-2.72 %)	174,710	5,710,207,000	0	0	33.10	32.6	33.4	31.5
4	4	03/12/2021	32.6	32.6	32.60	-1.20 (-3.55 %)	110,027	3,643,343,000	0	0	33.80	33.6	34.0	32.6
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3278	3278	15/09/2008	38.2	38.2	27.15	11.26 (41.82 %)	484,330	18,711,456,000	0	0	26.94	37.9	39.7	37.9
3279	3279	12/09/2008	37.9	37.9	26.94	9.61 (33.99 %)	2,480	93,992,000	0	0	28.29	37.9	37.9	37.9
3280	3280	11/09/2008	39.8	39.8	28.29	10.09 (33.97 %)	35,120	1,397,776,000	0	0	29.71	39.8	39.8	39.8
3281	3281	10/09/2008	41.8	41.8	29.71	10.53 (33.67 %)	223,650	9,348,730,000	31,000	1,295,800,000	31.27	41.8	41.9	41.8
3282	3282	09/09/2008	44.0	44.0	31.27	-2.00 (-4.35 %)	309,380	14,236,694,000	0	0	46.00	49.0	49.9	43.5

1.1. Tách dữ liệu trường "thay đổi"

```
# process colum 'change'
data[['Change_Value','Change_Percent']] = data['thay đổi'].str.split("(",expand=True)
data['Change_Value'] = data['Change_Value'].str[:-1]
data['Change_Percent'] = (data['Change_Percent'].str)[:-3]
data['Change_Percent'] = data['Change_Percent'].astype(float)
data['Change_Value'] = data['Change_Value'].astype(float)
```

1.2. Thay đổi định dạng và gộp một số trường

```
data['KL1'] = data['KL1'].str.replace(",", '')
data['KL1'] =data['KL1'].astype(float)
data['GT1'] = data['GT1'].str.replace(",", '')
data['GT1'] =data['GT1'].astype(float)

data['KL2'] = data['KL2'].str.replace(",", '')
data['KL2'] =data['KL2'].astype(float)
data['GT2'] = data['GT2'].str.replace(",", '')
data['GT2'] =data['GT2'].astype(float)

data['KL'] = data['KL1'] + data['KL2']
data['GT'] = data['GT1'] + data['GT2']
```

```
# convert object to datetime
data['Ngày'] = pd.to_datetime(data['Ngày'], format='%d/%m/%Y')
```

1.3. Xóa những trường không cần thiết

```
data = data.drop(columns=['Unnamed: 0', 'thay đổi', 'KL1', 'GT1', 'KL2', 'GT2'])
```

1.4. Đảo lại dữ liệu theo thời gian

```
df = data.iloc[::-1]
df.to_csv('VNA_preprocessed.csv',index= False)
```

1.5. Chuẩn hóa dữ liệu

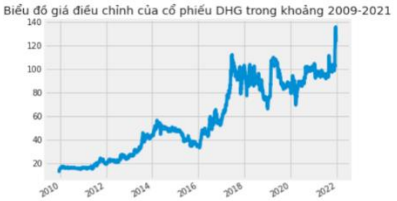
```
from sklearn.preprocessing import MinMaxScaler
def get_normalised_data(data):
    # Initialize a scaler, then apply it to the features
    scaler = MinMaxScaler()
    numerical = ['Giá điều chỉnh', 'Giá đóng cửa', 'Giá bình quân', 'Giá tham chiếu', 'Giá mở cửa', 'Giá cao nhất',
                 'Giá thấp nhất', 'Change_Value', 'Change_Percent', 'KL', 'GT']
    data[numerical] = scaler.fit_transform(data[numerical])
    return data
```

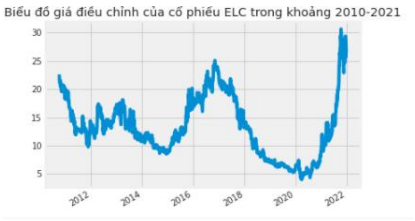
2. Phân tích các dữ liệu thu thập được

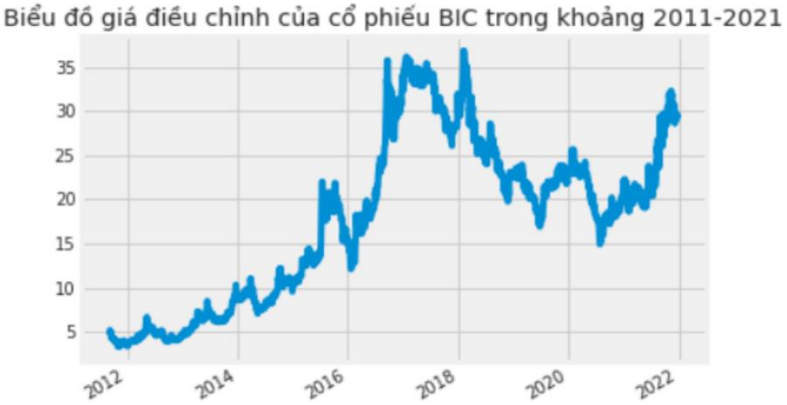
2.1. Biểu đồ giá điều chỉnh

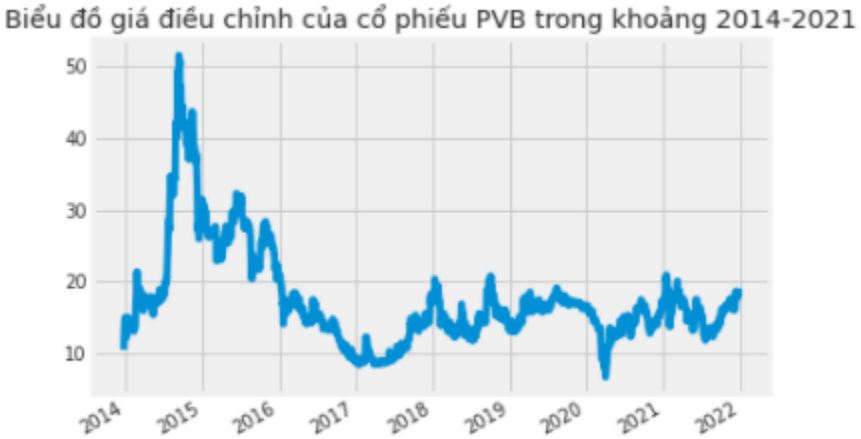
Điều chỉnh giá của 1 cổ phiếu là sự giảm giá của cổ phiếu đó đi từ 10% đến 20% so với giá trị ở mức cao gần đây của cổ phiếu đó. Nó không thể đoán trước được nhưng với số liệu cũng có thể cung cấp cho các nhà đầu tư chứng khoán cơ hội chọn được các những cổ phiếu chất lượng với giá chiết khấu và có thể đưa ra quyết định đầu tư ngắn hạn hay dài hạn.

Các bảng số liệu sau đây về giá điều chỉnh sẽ cho thấy cái nhìn khách quan về vấn đề này







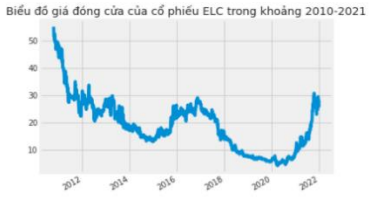


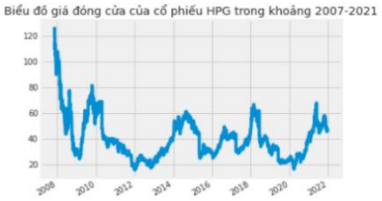
2.2. Biểu đồ giá đóng cửa

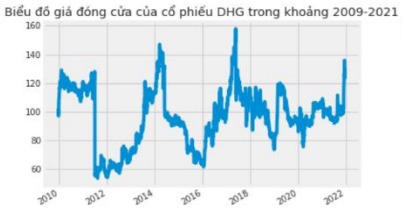
Giá đóng cửa là mức giá cửa cổ phiếu đó khi kết thúc phiên giao dịch của này hôm đó. Giá đóng cửa gồm giá mua và giá bán chứng khoán được xác định theo phương thức đấu giá. Vì cùng một thời điểm có nhiều người mua, nhiều người bán nên phải thực hiện việc đối chiếu giá mua hoặc giá bán. Trong một ngày giao dịch chứng khoán sẽ có mức giá cuối cùng được xác định cho việc mua,

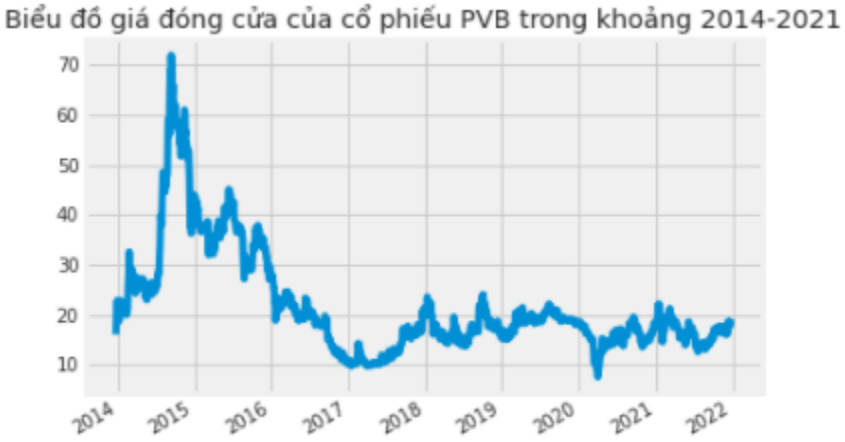
bán từng loại chứng khoán. Giá đóng cửa của ngày hôm nay sẽ là giá tham chiếu của ngày hôm sau làm cơ sở để tính giá trần và giá sàn của ngày đó.

Bảng giá sau đây bao gồm giá đóng cửa của một số mã cổ phiếu





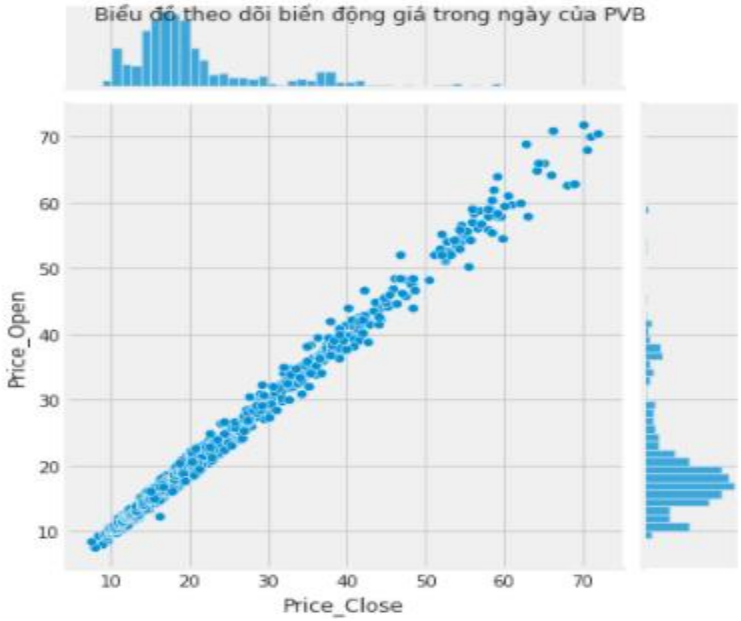


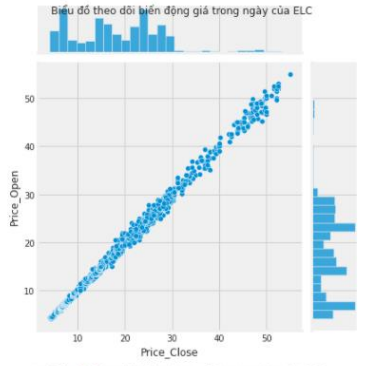


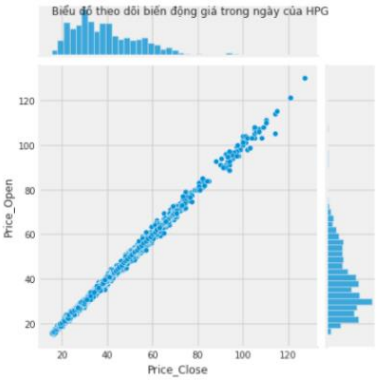
2.3. Biểu đồ theo dõi biến động giá trong ngày

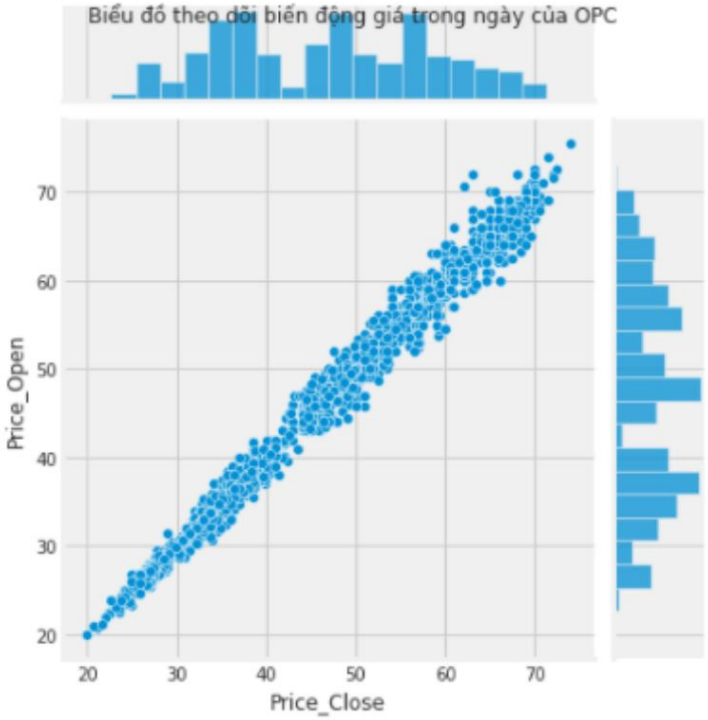
Giá mở cửa là giá thực hiện tại lần khớp lệnh đầu tiên trong ngày giao dịch chứng khoán. Giá mở cửa gồm giá mua và giá bán chứng khoán được xác định theo phương thức đấu giá. Giá đóng cửa là cơ sở để tính giá sàn và giá trần hôm sau nên cũng có là mối quan hệ quan trọng với giá mở cửa.

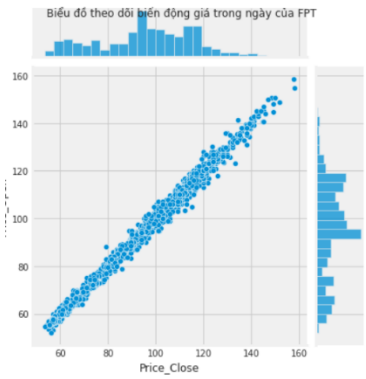
Biểu đồ dưới đây cho cái nhìn tổng quan về mối tương quan này

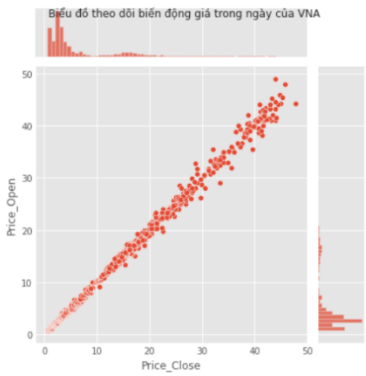


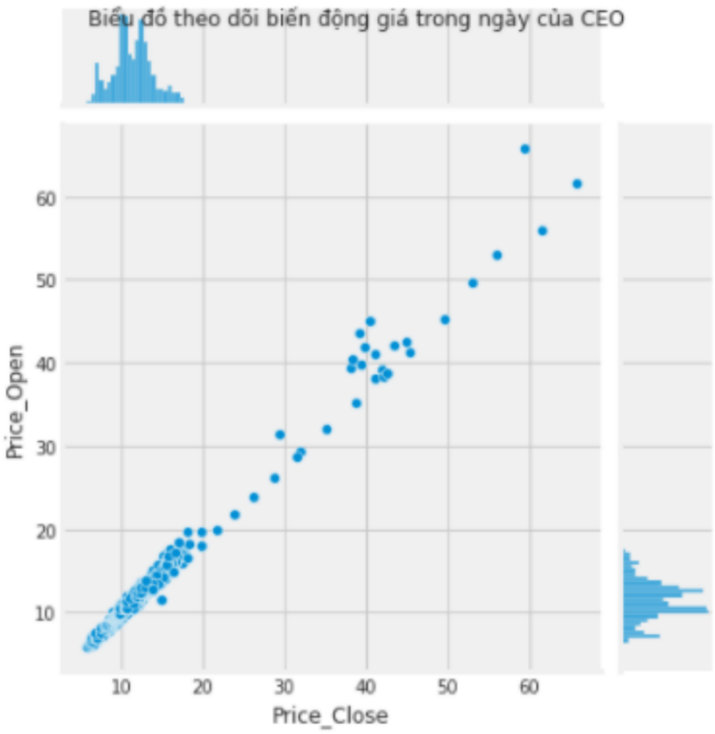

Biểu đồ theo dõi biến động giá trong ngày của ELC
Price_Open
Price_Close

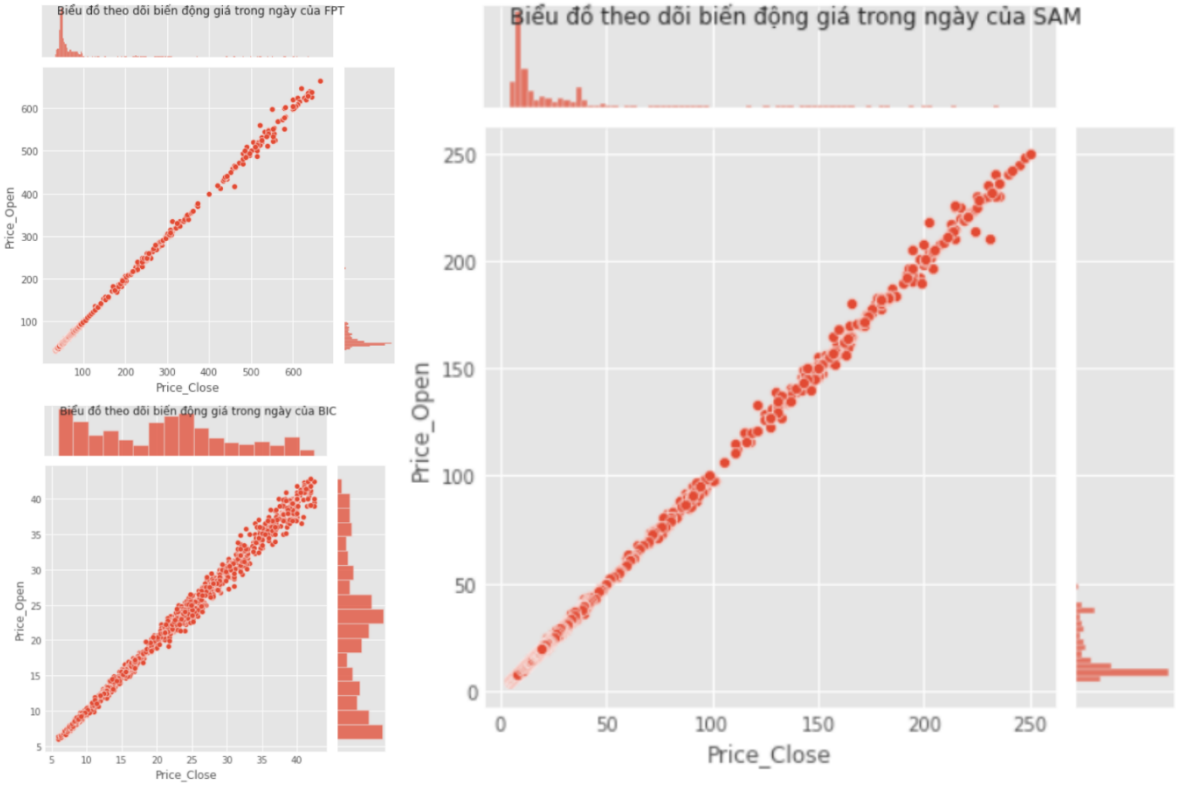

Biểu đồ theo dõi biến động giá trong ngày của HPG
Price_Open
Price_Close


Biểu đồ theo dõi biến động giá trong ngày của OPC
Price_Open
Price_Close


Biểu đồ theo dõi biến động giá trong ngày của FPT
Price_Close


Biểu đồ theo dõi biến động giá trong ngày của VNA
Price_Open
Price_Close

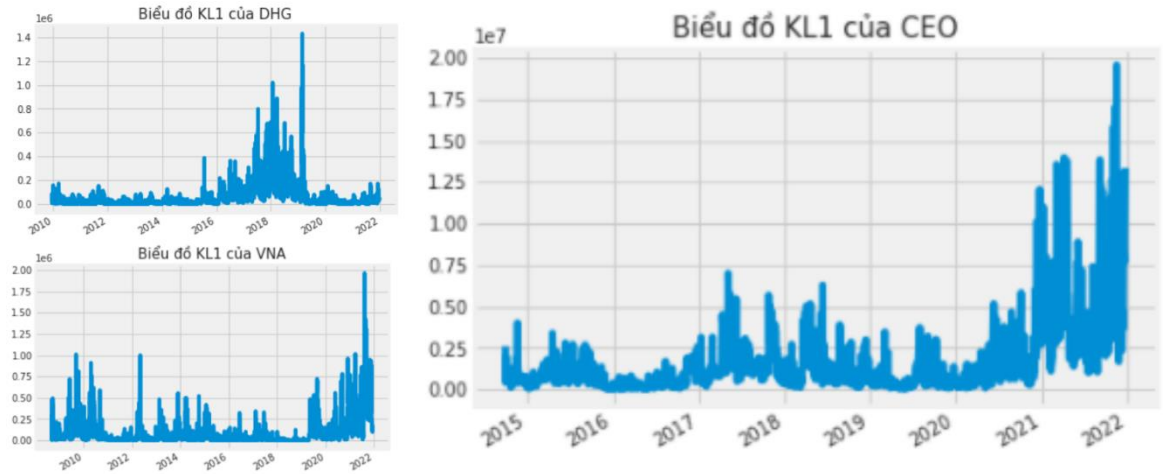

Biểu đồ theo dõi biến động giá trong ngày của CEO
Price_Open
Price_Close

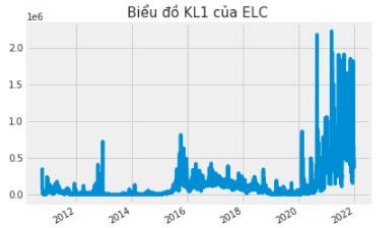


2.4. Biểu đồ khối lượng giao dịch trên sàn chứng khoán

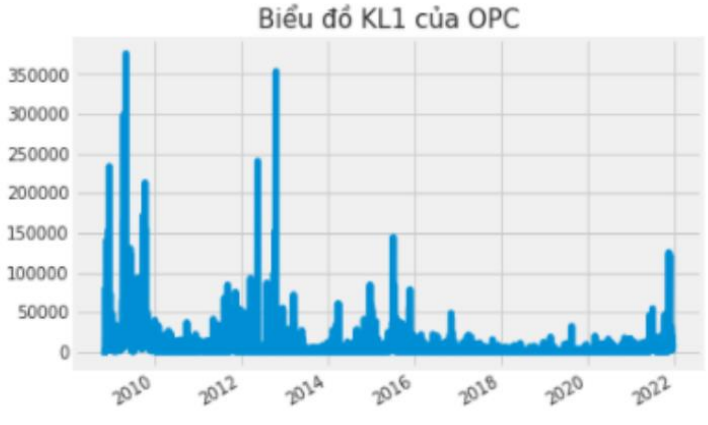
Khối lượng giao dịch của 1 cổ phiếu trên sàn chứng khoán là tổng số cổ phiếu thực sự được giao dịch (mua và bán) trong ngày giao dịch hoặc khoảng thời gian đã định, là một trong những yếu tố giúp cho nhiều nhà đầu tư xác định được xu hướng giá của cổ phiếu trái phiếu. Thông qua khối lượng giao dịch sau một phiên giao dịch ta có thể thấy được nhu cầu của người đầu tư hiện nay như thế nào, đánh giá tiềm năng của cổ phiếu đó.

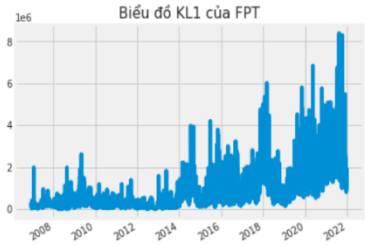
Biểu đồ sau đây sẽ thống kê khối lượng giao dịch trong của các một vài mã chứng khoán

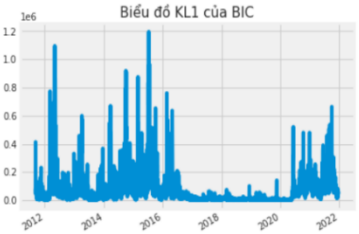


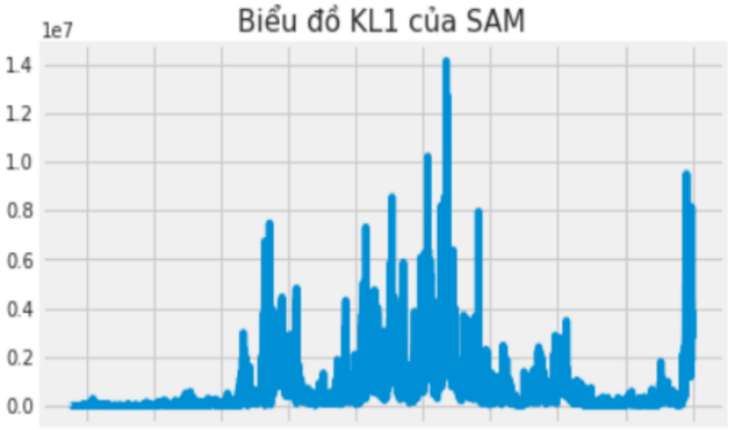

Biểu đồ KL1 của ELC
1e6
2.0
1.5
1.0
0.5
0.0
2012
2014
2016
2018
2020
2022


Biểu đồ KL1 của HPG
1e7
7
6
5
4
3
2
1
0
2008
2010
2012
2014
2016
2018
2020
2022


Biểu đồ KL1 của OPC
350000
300000
250000
200000
150000
100000
50000
0
2010
2012
2014
2016
2018
2020
2022


Biểu đồ KL1 của FPT
1e6
8
6
4
2
0
2008
2010
2012
2014
2016
2018
2020
2022


Biểu đồ KL1 của BIC
1e6
1.2
1.0
0.8
0.6
0.4
0.2
0.0
2012
2014
2016
2018
2020
2022

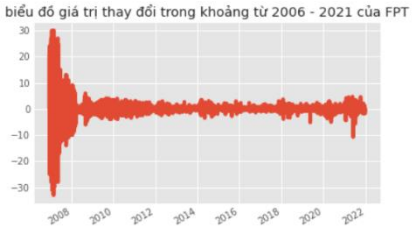

Biểu đồ KL1 của SAM
1e7
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
2004
2006
2008
2010
2012
2014
2016
2018
2020
2022

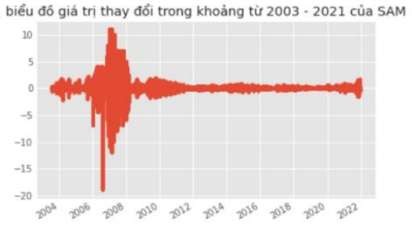

Biểu đồ KL1 của PVB
1e6
4
3
2
1
0
2014
2015
2016
2017
2018
2019
2020
2021
2022

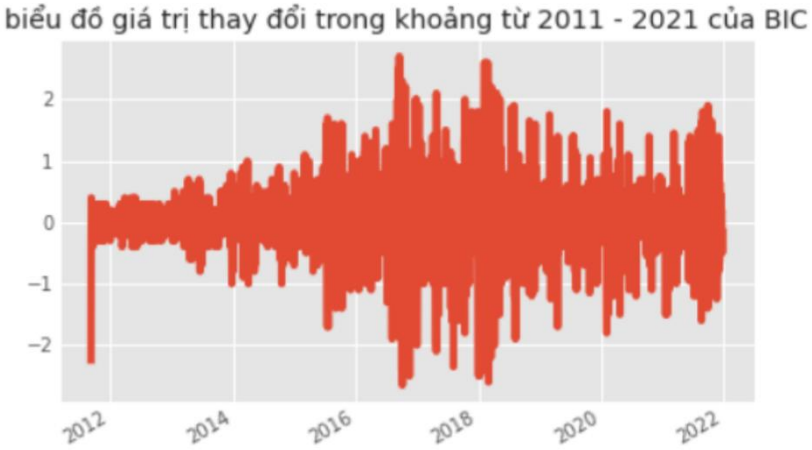
2.5. Biểu đồ giá trị biến động

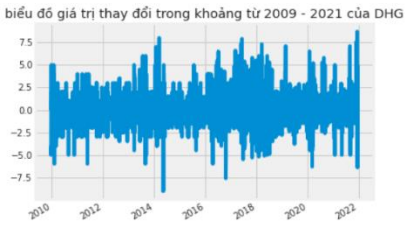
Biến động giá cổ phiếu là một phần không thể thiếu trong thị trường chứng khoán có thể mang lại khi biết những cổ phiếu có tiềm năng biến động mạnh thì ta có thể đưa ra cách giao dịch chúng một cách tối ưu, có thể mang đến những cơ hội thú vị. Một giao dịch bất ổn hơn có khả năng thu được lợi nhuận đáng kể, nhưng cũng có những tổn thất đáng kể.

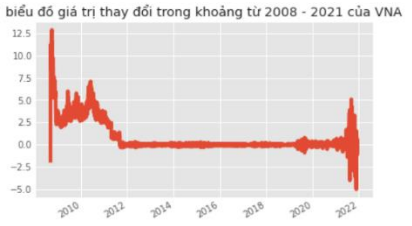
Bảng dữ liệu sau đây đưa ra tổng số biến động của 1 vài mã chứng khoán trên thị trường

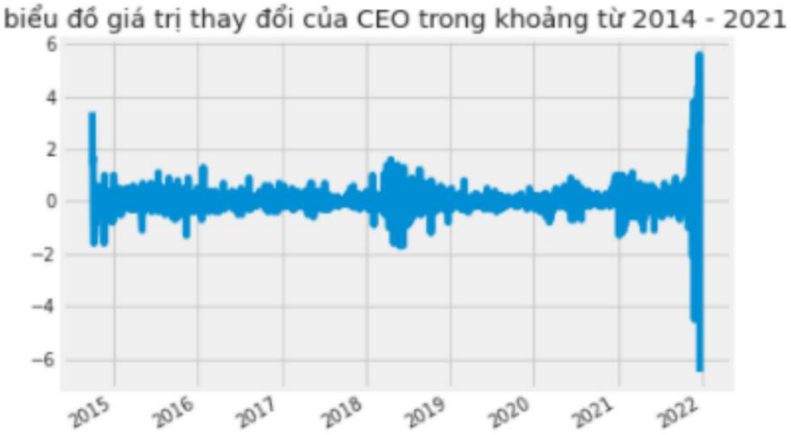


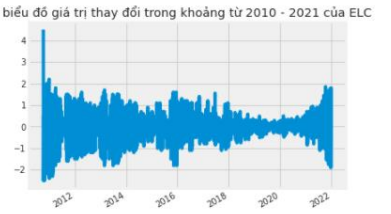


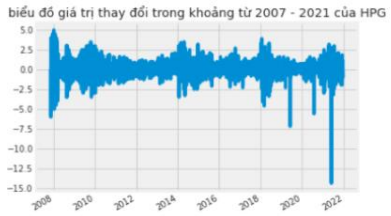


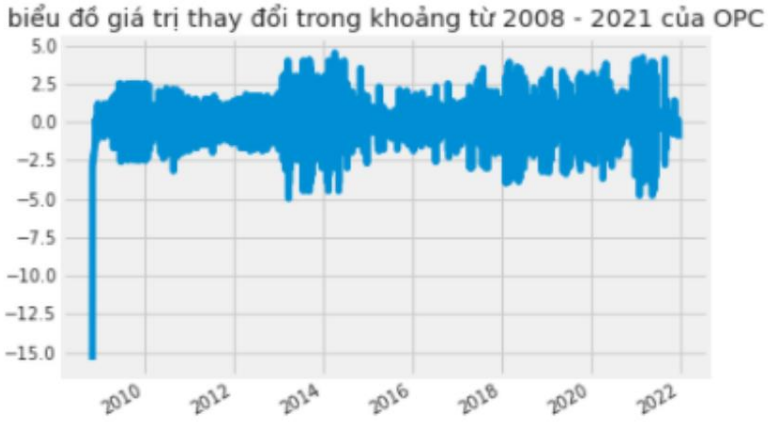


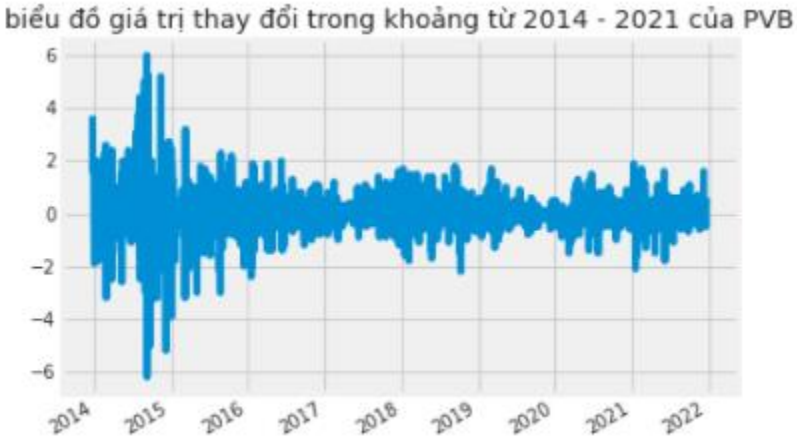












2.6. SMA

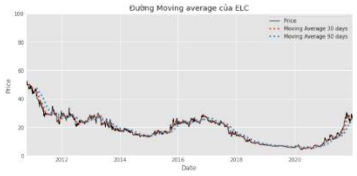
Đường SMA - Trung bình động đơn giản:

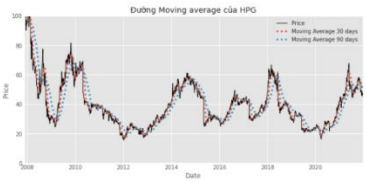
Trung bình động đơn giản (Simple moving Average) là đường trung bình cộng được tính bằng giá đóng trung bình trong một khoảng thời gian xác định. Chỉ số kỹ thuật này được dùng để xác định xem giá tài sản sẽ tiếp tục hay đảo ngược lại với xu hướng. Để tính được mức SMA, ta lấy tổng giá chứng khoán trong 1 khoảng thời gian chia cho khoảng thời gian.

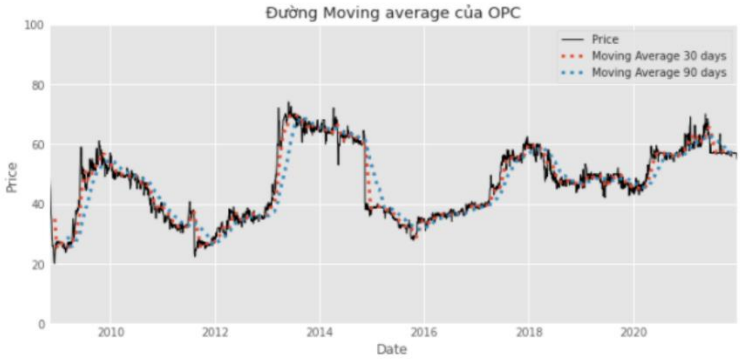
$$SMA= (A_1+ A_2+ A_3+ \ldots. +A_n)/n$$

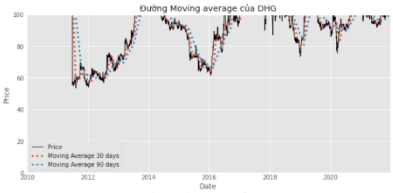
- A_i: Là giá trị trong từng khoản thời gian
- n: Tổng thời gian

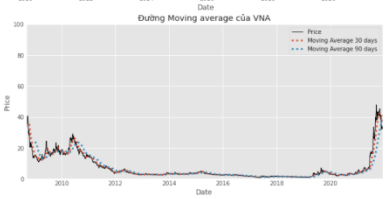
Dưới đây là biểu đồ để thể hiện đường SMA của một vài mã chứng khoán

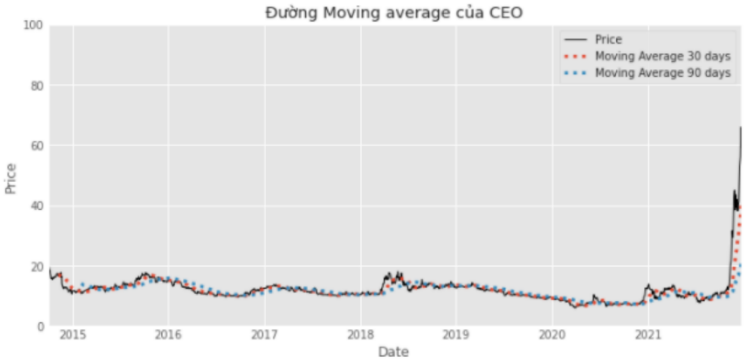

Đường Moving average của ELC
Price
Moving Average 30 days
Moving Average 90 days
Date

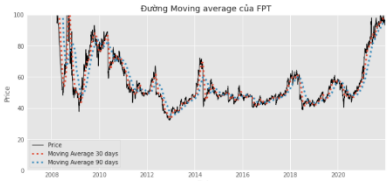

Đường Moving average của HPG
Price
Moving Average 30 days
Moving Average 90 days
Date

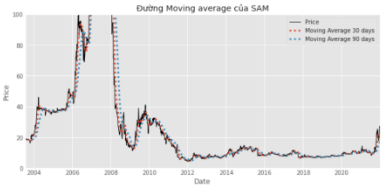

Đường Moving average của OPC
Price
Moving Average 30 days
Moving Average 90 days
Date

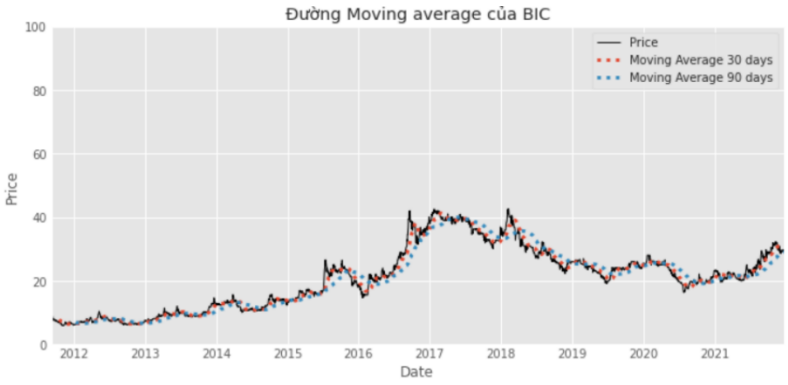

Đường Moving average của DHG
Price
Moving Average 30 days
Moving Average 90 days
Date

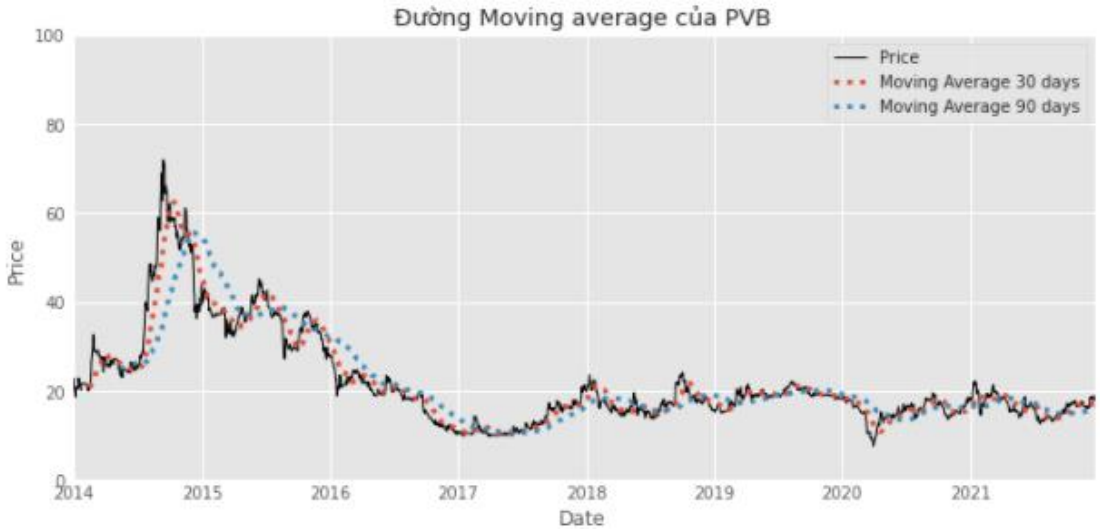

Đường Moving average của VNA
Price
Moving Average 30 days
Moving Average 90 days
Date


Đường Moving average của CEO
Price
Moving Average 30 days
Moving Average 90 days
Date


Đường Moving average của FPT
Price
Moving Average 30 days
Moving Average 90 days
Date


Đường Moving average của SAM
Price
Moving Average 30 days
Moving Average 90 days
Date


Đường Moving average của BIC
Price
Moving Average 30 days
Moving Average 90 days
Date


Đường Moving average của PVB
Price
Moving Average 30 days
Moving Average 90 days
100
80
60
40
20
0
Price
2014
2015
2016
2017
2018
2019
2020
2021
Date

PHẦN IV: XÂY DỰNG MÔ HÌNH HỌC MÁY

1. Phương pháp đánh giá các mô hình

1.1. Kịch bản đánh giá các mô hình

Nhóm chúng em sẽ lấy mã chứng khoán VNA để lấy làm tập dữ liệu và áp dụng các mô hình học máy. Để có thế thực hiện các mô hình thì nhóm chúng em đã chia bộ dữ liệu ra thành 3 phần theo TimeSeriesSplit gồm: training set, validation set, test set với tỉ lệ lần lượt là: 7,5:1,5:1,chia nhỏ dữ liệu để thực hiện xác thực chéo. Phần lớn dữ liệu liệu là traing set là để huấn luyện mô hình, càng nhiều dữ liệu tốt được "feed" vào mô hình thì dự đoán càng chính xác. Test set là phần dữ liệu dùng để đánh giá việc học của mô hình trong đó test set là dữ liệu mới hoàn toàn đối với mô hình để có thể đóng vai trò là dữ liệu thực tế trong bài toán cần giải quyết. Validation set là 1 tập dữ liệu cùng phân phối với test set nhằm cải thiện kết quả đánh giá trên test set vì trong quá trình huấn luyện có thể xảy ra tình trạng overfitting nên tập validation set có tác dụng như là cảnh báo sớm các vấn đề xấu của mô hình.

1.2. Chọn độ đo

Để có thể đánh giá được chất lượng mô hình nhóm chúng em sẽ đánh giá dựa theo SME (Mean Squared Erorr) và RMSE (Root Mean Squared Erorr) để có thể kiểm tra độ tin cậy của mô hình. Để có thể hiểu dễ hơn về RMSE thì ta sẽ đi qua khái niệm MSE trước. MSE đánh giá chất lượng của một ước lượng hay một yếu tố dự báo nào đó. Công thức:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Trong đó: ŷ là giá trị ước lượng, dự báo
y là giá trị quan sát được
n là số lần dự báo ước lượng

Sau khi đã hiểu qua về MSE thì RMSE được coi như là một biện pháp thường sử dụng trong những khác biệt của giá trị được dự đoán bởi mô hình hay là ước lượng các giá trị ta quan sát được. RMSE là thước đo mức độ lan truyền của những phần lỗi dự đoán. Nói cách khác, nó cho bạn biết mức độ tập trung của dữ liệu xung quanh dòng phù hợp nhất. RMSE là thước đo mức độ hiệu quả của mô hình của bạn. Nó thực hiện điều này bằng cách đo sự khác biệt giữa các giá trị dự đoán và giá trị thực tế. RMSE 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. RMSE là thước đo độ chính xác, để so sánh các lỗi dự báo của các mô hình khác nhau cho một tập dữ liệu cụ thể chứ không phải giữa các bộ dữ liệu, vì nó phụ thuộc vào quy mô.

Công thức:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\mathrm{Y}i - \hat{\mathrm{Y}}i)^2}$$

Trong đó: ŷ là giá trị ước lượng, dự báo
y là giá trị quan sát được
n là số lần dự báo ước lượng

2. Mô hình hồi quy tuyến tính

2.1. Lý do chọn mô hình hồi quy tuyến tính

Linear regression là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. Nói cách khác, linear regression là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên biến độc lập (X). Nó có thể được sử dụng cho các trường hợp muốn dự đoán một số lượng liên tục. Dự báo thị trường chứng khoán là một ứng dụng hấp dẫn trong mô hình hồi quy, mô hình giúp cho người sử dụng xác định các điểm giá mở cửa, đóng cửa, …

2.2. Cơ sở lý thuyết

2.2.1. Các khái niệm cơ bản:

- Chuẩn bị: Để bắt đầu với linear regression, cần lướt qua một số khái niệm toán học về thống kê:
  + Tương quan(r) – Giải thích mối quan hệ giữa hai biến, giá trị có thể chạy từ -1 -> +1
  + Phương sai (σ2) – Đánh giá độ phân tán trong dữ liệu.
  + Độ lệch chuẩn(σ) – Đánh giá độ phân tán trong dữ liệu (căn bậc 2 của phương sai)
  + Phân phối chuẩn.
  + Sai số(lỗi)
- Giả định: Không một kích thước nào phù hợp cho tất cả, điều này cũng đúng đối với linear regression, để thỏa mãn linear regression, dữ liệu nên thỏa mãn một vài giả định quan trọng. Nếu dữ liệu không làm theo các giả định, kết quả đạt được sẽ bị sai cũng như gây hiểu nhầm.

- Tuyến tính và thêm vào: Nên có một mối quan hệ tuyến tính giữa biến độc lập và biến không độc lập và ảnh hưởng của sự thay đổi trong giá trị của các biến độc lập nên ảnh hưởng thêm vào các biến phụ thuộc.
- Tính bình thường của phân bổ các lỗi: Sự phân bổ sai khác giữa các giá trị thực và giá trị dự đoán (sai số) nên được phân bổ một cách bình thường.
- Sự tương đồng: Phương sai của các lỗi nên là một giá trị không đổi so với thời gian, dự đoán, giá trị của biến độc lập.
- Sự độc lập về thống kê của các lỗi: Các sai số(dư) không nên có bất kì mối tương quan nào giữa chúng. Ví dụ: Trong trường hợp dữ liệu theo chuỗi thời gian, không nên có sự tương quan giữa các sai số liên tiếp nhau.

2.2.2. Đường hồi quy tuyến tính

Trong khi sử dụng linear regression, mục tiêu là để làm sao một đường thẳng có thể tạo được sự phân bổ gần nhất với hầu hết các điểm. Do đó làm giảm khoảng cách của các điểm dữ liệu cho đến đường đó, thường trong đồ thị các điểm dữ liệu khác nhau và có một đường thẳng đại diện cho một đường gần đúng có thể giải thích mối quan hệ giữa các trục x và y. Thông qua hồi quy tuyến tính, cần phải cố gắng tìm ra một đường như vậy. Ví dụ, nếu có một biến phụ thuộc y và một biến độc lập x – mối quan hệ giữa x và y có thể được biểu diễn dưới dạng phương trình sau:

$$y = b0 + b1*x$$

Trong đó:

+ y là biến phụ thuộc
+ x là biến độc lập
+ b0 là hằng số
+ b1 là hệ số mối quan hệ giữa x và y

- Một số tính chất:

Đường hồi quy luôn luôn đi qua trung bình của biến độc lập x cũng như trung bình của biến phụ thuộc y. Đường hồi qui tối thiểu hóa tổng của "Diện tích các sai số". Đó là lý do tại sao phương pháp hồi quy tuyến tính được gọi là "Ordinary Least Square(OLS)" B1 giải thích sự thay đổi trong y với sự thay đổi x bằng một đơn vị. Nói cách khác, nếu tăng giá trị của x bởi một đơn vị thì nó sẽ là sự thay đổi giá trị của y.

- Tìm đường hồi quy tuyến tính:

Sử dụng công cụ thông kê ví dụ như Excel, R, … sẽ trực tiếp tìm hằng số b0 và b1 như là kết quả của hàm hồi quy tuyến tính. Như lí thuyết ở trên, nó hoạt động trên

khái niệm OLS và cố gắng giảm bớt diện tích sai số, các công cụ này sử dụng các gói phần mềm tính các hằng số.

- Hiệu suất của mô hình:

Một khi xây dựng mô hình, người ta luôn tự hỏi là để biết liệu mô hình có đủ để dự đoán trong tương lai hoặc là mối quan hệ mà đã xây dựng giữa các biến phụ thuộc và độc lập là đủ hay không.

- Công thức tính hiệu suất:

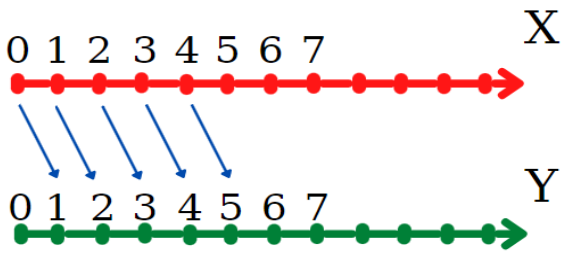
$$R^2 = \frac{TSS - RSS}{TSS}$$

Trong đó:

- Tổng các diện tích(TSS): là một phép đo tổng biến thiên trong tỷ lệ đáp ứng/ biến phụ thuộc y và có thể được coi là số lượng biến thiên vốn có trong đáp ứng trước khi hồi quy được thực hiện.
- Sum of squares(RSS): đo lường lượng biến đổi còn lại không giải thích được sau khi thực hiện hồi quy.
- (TSS - RSS) đo lường mức độ thay đổi trong đáp ứng được giải thích (hoặc loại bỏ) bằng cách thực hiện hồi quy.

2.3. Xây dựng mô hình

Với mô hình tuyến tính thì nhóm em sẽ lấy target trong 1 ngày để làm target cho ngày hôm trước vì vậy nên đối với tập giá trị Y (giá đóng cửa) thì xóa đi dữ liệu ngày đầu tiên còn tập giá trị X (chứa các trường còn lại) thì sẽ xóa đi dữ liệu ngày cuối cùng và các trường không cần thiết như: “Ngày”



Input X: Dữ liệu của ngày hôm nay
Target Y: Là dữ liệu của tương lai

```
from sklearn import linear_model
linear = linear_model.LinearRegression()
input_train = X_train[X_train.columns[1:]].iloc[:-1]
input_val = X_val[X_val.columns[1:]].iloc[:-1]
input_test = X_test[X_test.columns[1:]].iloc[:-1]
target_train = Y_train[1:]
target_val = Y_val[1:]
target_test = Y_test[1:]
```

Sau khi thu được dữ liệu cần thiết ta bắt đầu huấn luyện mô hình với tập Train

```
linear.fit(input_train, target_train)
```

Ta thu được 2 vectơ

```
linear.intercept_
```

```
-0.0007908284496517259
```

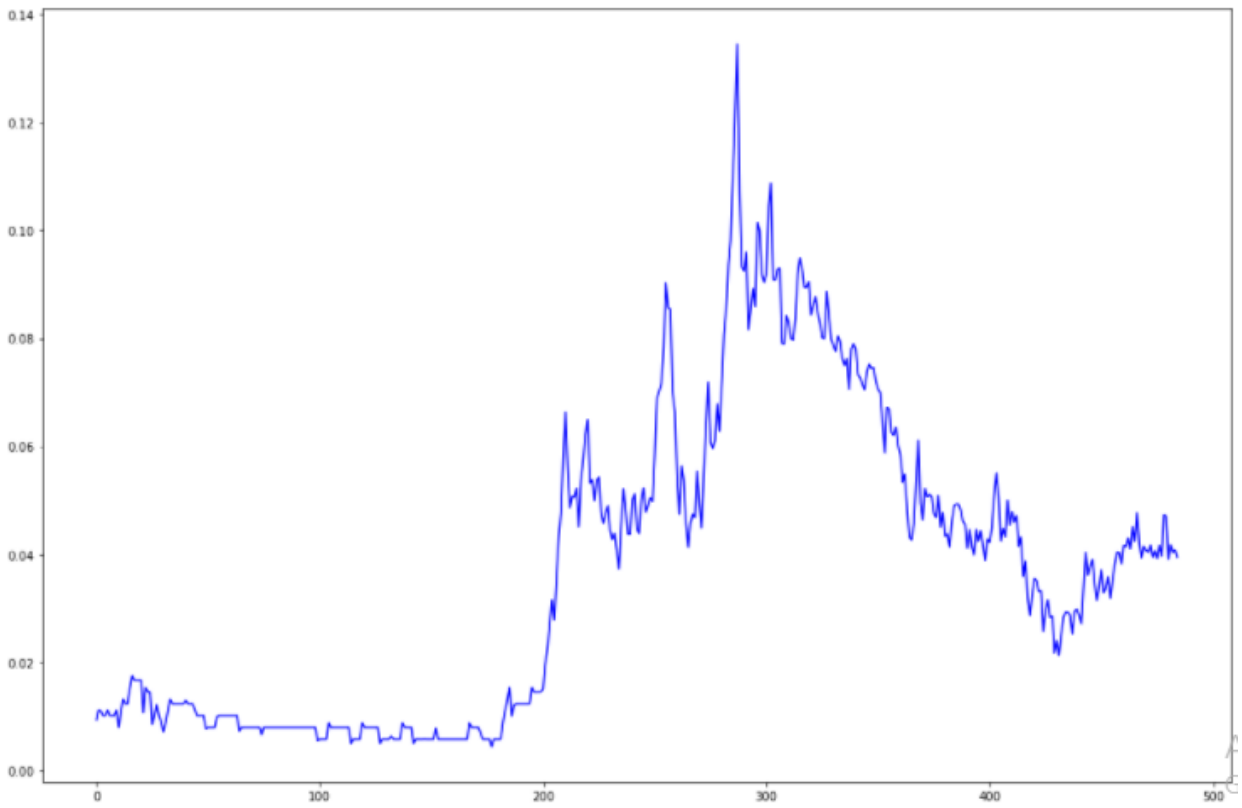
```
linear.coef_
```

```
array([ 0.51435894, -0.38310337, -0.17989476,  0.44172538,  0.61700372,
        0.00174859,  0.01381122])
```

Khi thu được 1 vài kết quả ở tập dữ liệu Train ta sẽ sử dụng mô hình này để dự đoán kết quả trên tập Val

```
predict_val = linear.predict(input_val)
plt.plot(predict_val, '#0000FF', label='Dự đoán tren tập Val')
```

```
[<matplotlib.lines.Line2D at 0x2054e801d90>]
```



2.4. Đánh giá mô hình

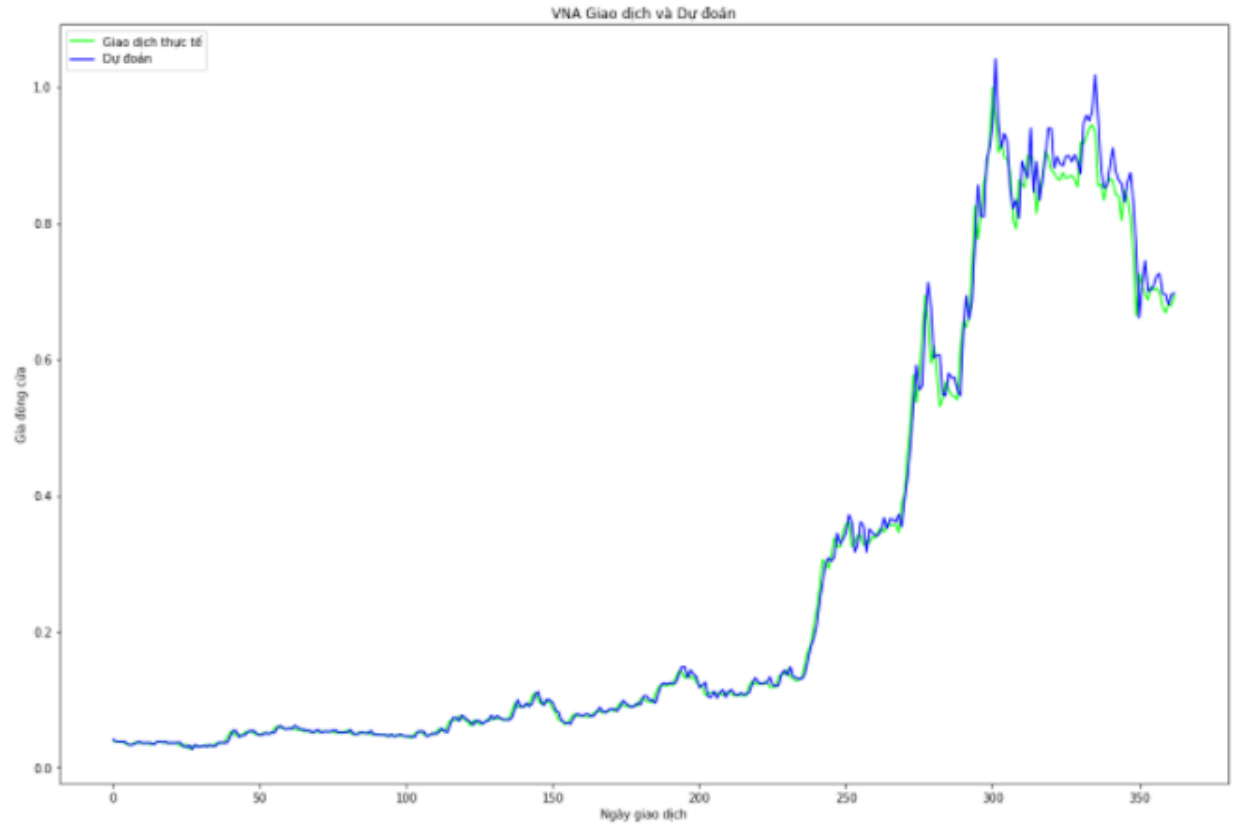
-Để đánh giá được mô hình nhóm chúng em sử dụng MSE và RMSE để có thể đưa ra kết quả nếu giá trị càng bé thì mô hình càng tin cậy:

```
predict_train = linear.predict(input_train)
trainScore = mean_squared_error(predict_train, target_train)
print('Train Score: %.4f MSE (%.4f RMSE)' % (trainScore, math.sqrt(trainScore)))

valScore = mean_squared_error(predict_val, target_val)
print('Val Score: %.8f MSE (%.8f RMSE)' % (valScore, math.sqrt(valScore)))

predict_test = linear.predict(input_test)
testScore = mean_squared_error(predict_test, target_test)
print('Test Score: %.8f MSE (%.8f RMSE)' % (testScore, math.sqrt(testScore)))
```

```
Train Score: 0.0001 MSE (0.0071 RMSE)
Val Score: 0.00001500 MSE (0.00387280 RMSE)
Test Score: 0.00048393 MSE (0.02199836 RMSE)
```

-Kết quả dự đoán của mô hình đối với giá đóng cửa

```
plot_prediction(target_test, predict_test)
```



3. Mô hình LSTM

3.1. Lý do chọn LSTM

Neural Network có 2 mô hình lớn là Convolutional Neural Network (CNN) cho bài toán có input là ảnh và Recurrent neural network (RNN) cho bài toán dữ liệu dạng chuỗi (sequence). Về cơ bản nếu bạn thấy sequence data hay time-series data mà muốn áp dụng Neural Network sẽ nghĩ ngay đến RNN.Tuy nhiên mạng RNN gặp phải một số hạn chế đó là phải thực hiện tuần tự, đạo hàm bị triệt tiêu (Vanishing gradient) , bùng nổ đạo hàm (Exploding gradient).LSTM là một mạng cải tiến của RNN nhằm giải quyết phần nào các vấn đề mà mạng RNN gặp phải. Rất nhiều các bài toán học máy sử dụng LSTM đem lại kết quả rất đáng chú ý so với việc sử dụng các phương pháp khác. Dữ liệu chứng khoán chúng ta thu thập được là một dạng dữ liệu kiểu time-series vì vậy khá phù hợp khi sử dụng LSTM huấn luyện dữ liệu mà nhóm đã thu thập được.

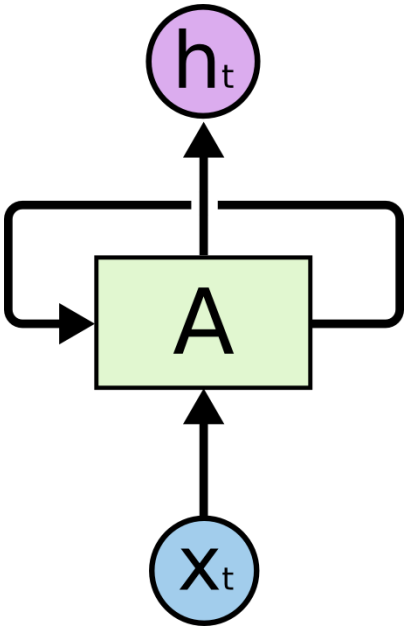
3.2. Cơ sở lý thuyết

3.2.1. Mạng hồi quy RNN

Con người không bắt đầu suy nghĩ của họ từ đầu tại tất cả các thời điểm. Cũng như bạn đang đọc bài viết này, bạn hiểu mỗi chữ ở đây dựa vào từ bạn đã hiểu các chữ trước đó chứ không phải là đọc tới đâu ném hết đi tới đó, rồi lại bắt đầu suy nghĩ lại từ đầu tới chữ bạn đang đọc. Tức là tư duy đã có một bộ nhớ để lưu lại những gì diễn ra trước đó.

Tuy nhiên các mô hình mạng nơ-ron truyền thống thì không thể làm được việc đó, đó có thể coi là một khuyết điểm chính của mạng nơ-ron truyền thống. Ví dụ, bạn muốn phân loại các bối cảnh xảy ra ở tất cả các thời điểm trong một bộ phim, thì đúng là không rõ làm thế nào để có thể hiểu được một tình huống trong phim mà lại phụ thuộc vào các tình huống trước đó nếu sử dụng các mạng nơ-ron truyền thống.

Mạng nơ-ron hồi quy (Recurrent Neural Network) sinh ra để giải quyết vấn đề đó. Mạng này chứa các vòng lặp bên trong cho phép thông tin có thể lưu lại được.

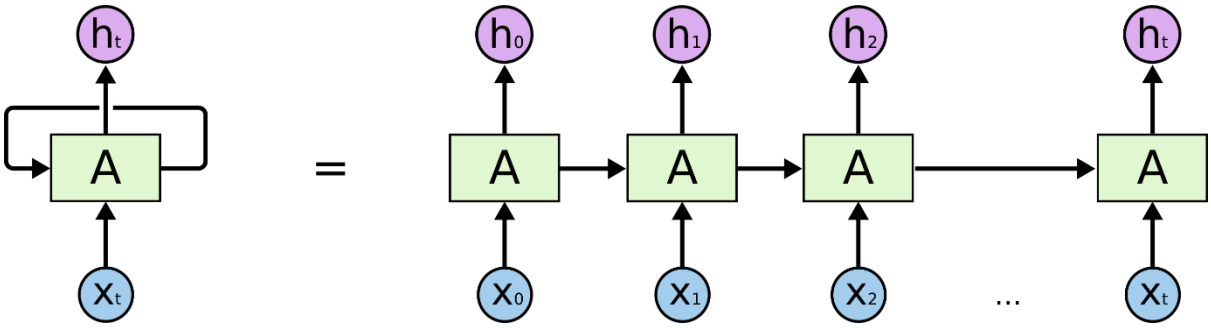


Recurrent Neural Networks have loops.

Hình vẽ trên mô tả một đoạn của mạng nơ-ron hồi quy A*A* với đầu vào là x_t*xt* và đầu ra là h_*tht*. Một vòng lặp cho phép thông tin có thể được truyền từ bước này qua bước này qua bước khác của mạng nơ-ron.

Các vòng lặp này khiến cho mạng nơ-ron hồi quy trông có vẻ khó hiểu. Tuy nhiên, nếu bạn để ý một chút thì nó không khác mấy so với các mạng nơ-ron thuần. Một mạng nơ-ron hồi quy có thể được coi là nhiều bản sao chép của cùng một mạng, trong

đó mỗi đầu ra của mạng này là đầu vào của một mạng sao chép khác. Nói thì hơi khó hiểu, nhưng bạn hãy xem hình mô tả sau:



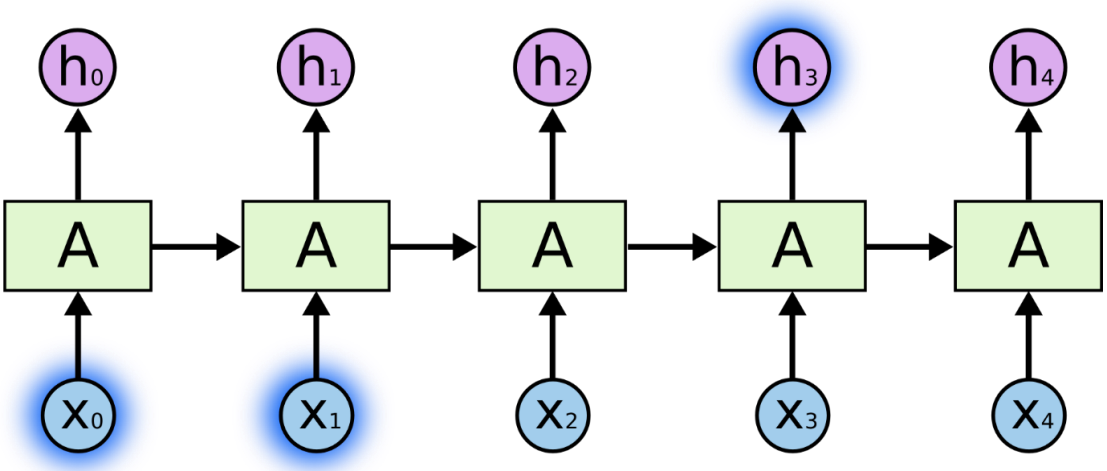
An unrolled recurrent neural network.

Chuỗi lặp lại các mạng này chính là phân giải của mạng nơ-ron hồi quy, các vòng lặp khiến chúng tạo thành một chuỗi danh sách các mạng sao chép nhau. Bạn có thấy nó khác gì một mạng nơ-ron thuần không? Không khác gì phải không? Các nút của mạng vẫn nhận đầu vào và có đầu ra hệt như mạng nơ-ron thuần. Trong vài năm gần đây, việc ứng dụng RNN đã đưa ra được nhiều kết quả không thể tin nổi trong nhiều lĩnh vực: nhận dạng giọng nói, mô hình hóa ngôn ngữ, dịch máy, mô tả ảnh, Danh sách vẫn còn đang được mở rộng tiếp. Anh Andrej Karpathy đã đề cập tới một số kết quả mà RNN mang lại.Đằng sau sự thành công này chính là sự đóng góp của LSTM. LSTM là một dạng đặc biệt của mạng nơ-ron hồi quy, với nhiều bài toán thì nó tốt hơn mạng hồi quy thuần. Hầu hết các kết quả thú vị thu được từ mạng RNN là được sử dụng với LSTM. Trong bài viết này, ta sẽ cùng khám phá xem mạng LSTM là cái gì nhé.

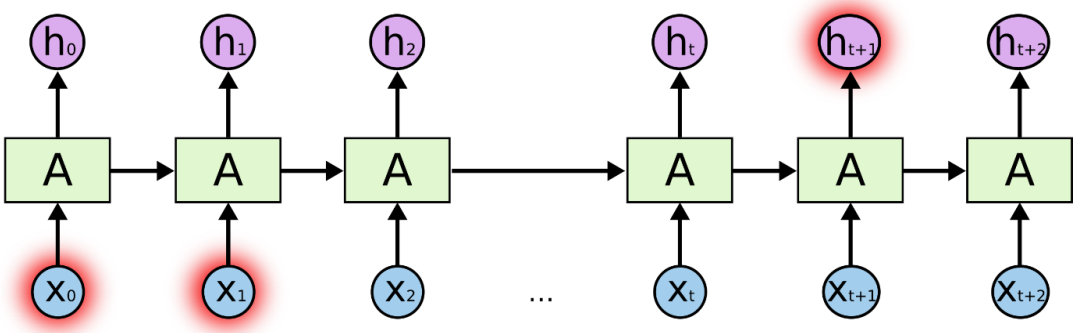
3.2.2. Vấn đề phụ thuộc xa

Một điểm nổi bật của RNN chính là ý tưởng kết nối các thông tin phía trước để dự đoán cho hiện tại. Việc này tương tự như ta sử dụng các cảnh trước của bộ phim để hiểu được cảnh hiện thời. Nếu mà RNN có thể làm được việc đó thì chúng sẽ cực kì hữu dụng, tuy nhiên liệu chúng có thể làm được không? Câu trả lời là *còn tùy*.

Đôi lúc ta chỉ cần xem lại thông tin vừa có thôi là đủ để biết được tình huống hiện tại. Ví dụ, ta có câu: "*các đám may trên bầu trời*" thì ta chỉ cần đọc tới "*các đám may trên bầu*" là đủ biết được chữ tiếp theo là "*trời*" rồi. Trong tình huống này, khoảng cách tới thông tin có được cần để dự đoán là nhỏ, nên RNN hoàn toàn có thể học được.



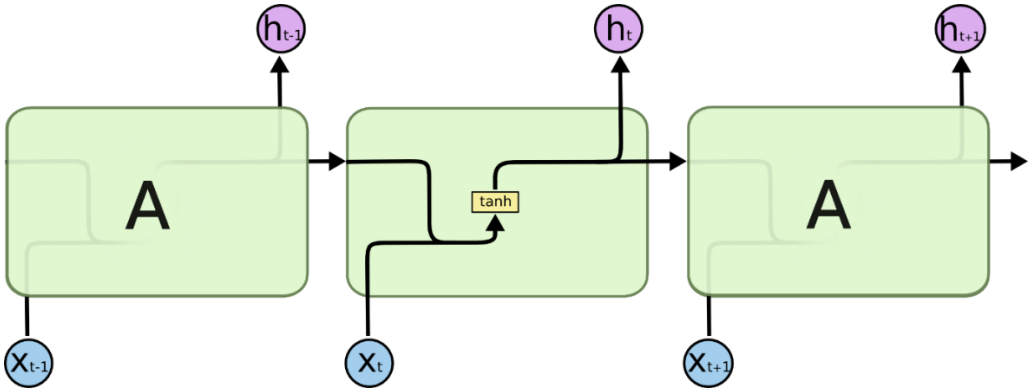
Nhưng trong nhiều tình huống ta buộc phải sử dụng nhiều ngữ cảnh hơn để suy luận. Ví dụ, dự đoán chữ cuối cùng trong đoạn: "*I grew up in France… I speak fluent French.*". Rõ ràng là các thông tin gần ("*I speak fluent*") chỉ có phép ta biết được đằng sau nó sẽ là tên của một ngôn ngữ nào đó, còn không thể nào biết được đó là tiếng gì. Muốn biết là tiếng gì, thì ta cần phải có thêm ngữ cảnh "*I grew up in France*" nữa mới có thể suy luận được. Rõ ràng là khoảng cách thông tin lúc này có thể đã khá xa rồi.Thật không may là với khoảng cách càng lớn dần thì RNN bắt đầu không thể nhớ và học được nữa.



Về mặt lý thuyết, rõ ràng là RNN có khả năng xử lý các phụ thuộc xa (long-term dependencies). Chúng ta có thể xem xét và cài đặt các tham số sao cho khéo là có thể giải quyết được vấn đề này. Tuy nhiên, đáng tiếc trong thực tế RNN có vẻ không thể học được các tham số đó. Vấn đề này đã được khám phá khá sâu bởi Hochreiter (1991) [Đức] và Bengio, et al. (1994), trong các bài báo của mình, họ đã tìm được nhưng lý do căn bản để giải thích tại sao RNN không thể học được.Tuy nhiên, rất cám ơn là LSTM không vấp phải vấn đề đó!

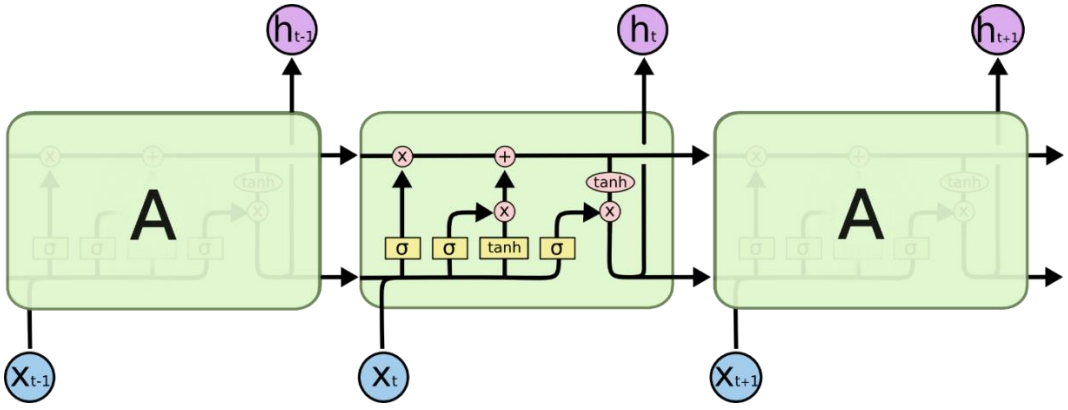
3.2.3. **Mạng LSTM**

Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks) thường được gọi là LSTM - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi Hochreiter & Schmidhuber (1997), và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay. LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào. Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một tầng tanh.



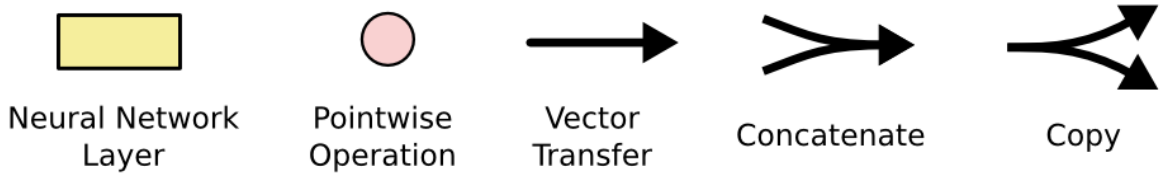
The repeating module in a standard RNN contains a single layer.

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



The repeating module in an LSTM contains four interacting layers.

Giờ thì đừng hoang mang về chi tiết bên trong chúng ngay, chúng ta sẽ khám phá chúng chi tiết chúng ở bước sau. Điều bạn cần làm bây giờ là làm hãy làm quen với các kí hiệu mà ta sẽ sử dụng ở dưới đây:

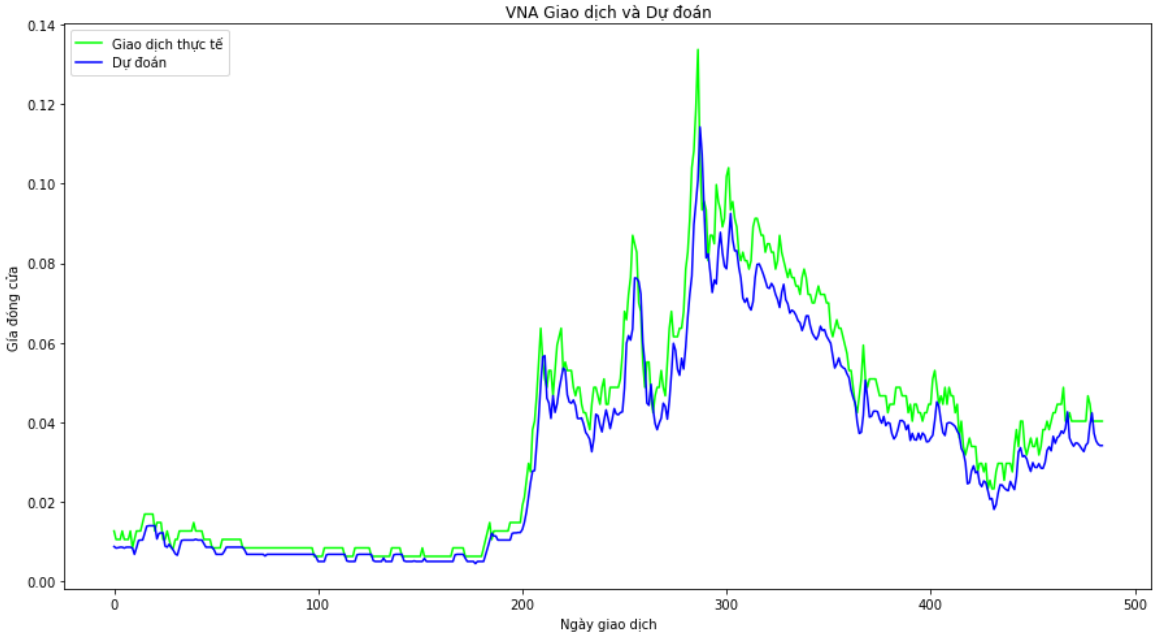


Ở sơ đồ trên, mỗi một đường mang một véc-tơ từ đầu ra của một nút tới đầu vào của một nút khác. Các hình trong màu hồng biểu diễn các phép toán như phép cộng véc-tơ chẳng hạn, còn các ô màu vàng được sử dụng để học trong các từng mạng nơ-ron. Các đường hợp nhau kí hiệu việc kết hợp, còn các đường rẽ nhánh ám chỉ nội dung của nó được sao chép và chuyển tới các nơi khác nhau.

3.3. Xây dựng mô hình

Các tập Train, Val, Test giống với các tập đã dùng trong mô hình Linears Regrestion ở trên.

a) Mô hình LSTM với các tham số mặc định

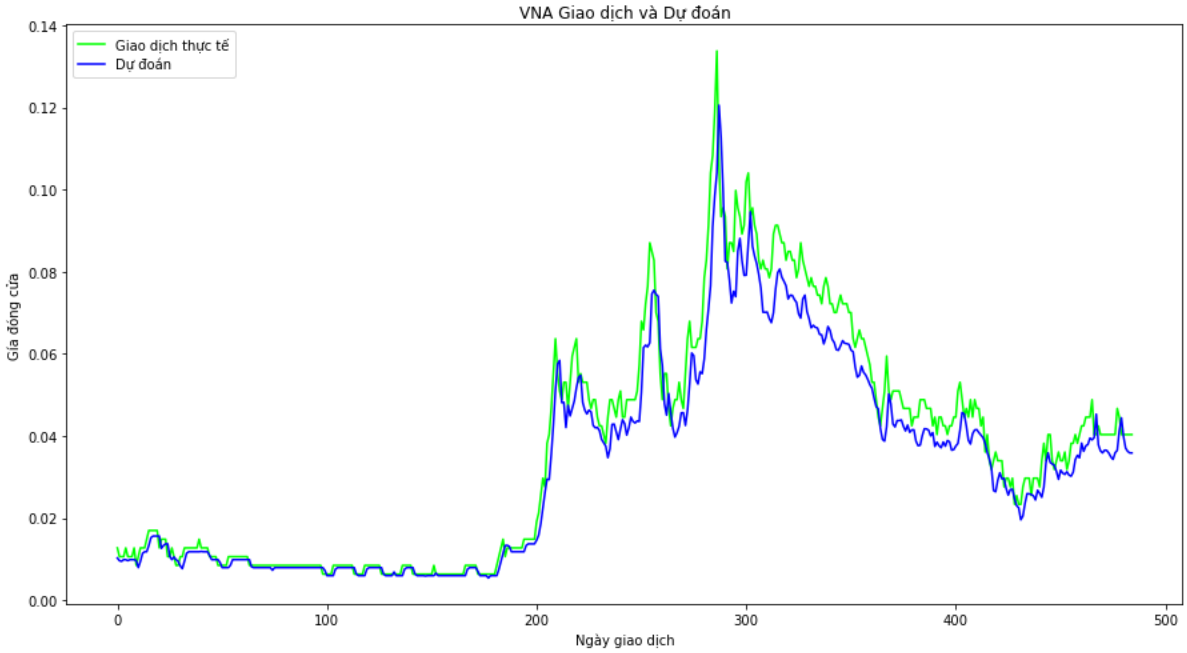


Biểu đồ đường thể hiện giao dịch thực tế và dự đoán trên tập Val

Kết quả:

```
Train Score: 0.0007 MSE (0.0260 RMSE)
Val Score: 0.00005266 MSE (0.00725667 RMSE)
Test Score: 0.00118950 MSE (0.03448910 RMSE)
```

b) Mô hình LSTM với các tham số đã điều chỉnh



Kết quả:

```
LSTM IMPROVED
Train Score: 0.0007 MSE (0.0260 RMSE)
Val Score: 0.00004554 MSE (0.00674866 RMSE)
Test Score: 0.00059399 MSE (0.02437191 RMSE)
```

Nhìn vào kết quả, ta thấy mô hình LSTM cải tiến có kết quả tốt hơn nhiều trên tập Test so với mô hình LSTM cơ bản.

Các tham số đã điều chỉnh:

- Units: số layer ẩn đặc trưng cho độ phức tạp của mạng

	Units	Score Val
0	32	0.000042
1	64	0.000048
2	128	0.000042
3	256	0.000066
4	512	0.000043

Chọn units=32. Vì lượng dữ liệu train không lớn nên không lựa chọn units cao để làm phức tạp mạng hơn.

- Batch_size: số lượng mẫu dữ liệu trong một batch

	Batch Size	Score Val BS
0	16	0.000019
1	32	0.000043
2	64	0.000017
3	128	0.000017
4	256	0.000020

Chọn batch_size=64. Vì dữ liệu không lớn nên muốn lượt duyệt danh sách lâu hơn, thời gian học nhiều hơn.

- Dropout: loại bỏ 1 vài units trong quá trình train mô hình

	Drop Out	Score Val DR
0	0.05	0.000038
1	0.10	0.000050
2	0.15	0.000022
3	0.20	0.000164
4	0.25	0.000082
5	0.30	0.000247
6	0.35	0.000155
7	0.40	0.000086
8	0.45	0.000101
9	0.50	0.000847
10	0.55	0.001190
11	0.60	0.001483
12	0.65	0.001539
13	0.70	0.001919
14	0.75	0.002516
15	0.80	0.002705
16	0.85	0.002392
17	0.90	0.003157
18	0.95	0.004300

Chọn dropout=0.15

3.4. Đánh giá mô hình

a) Kết quả

- **Linear**

```
Train Score: 0.0001 MSE (0.0071 RMSE)
Val Score: 0.00001500 MSE (0.00387280 RMSE)
Test Score: 0.00048393 MSE (0.02199836 RMSE)
```

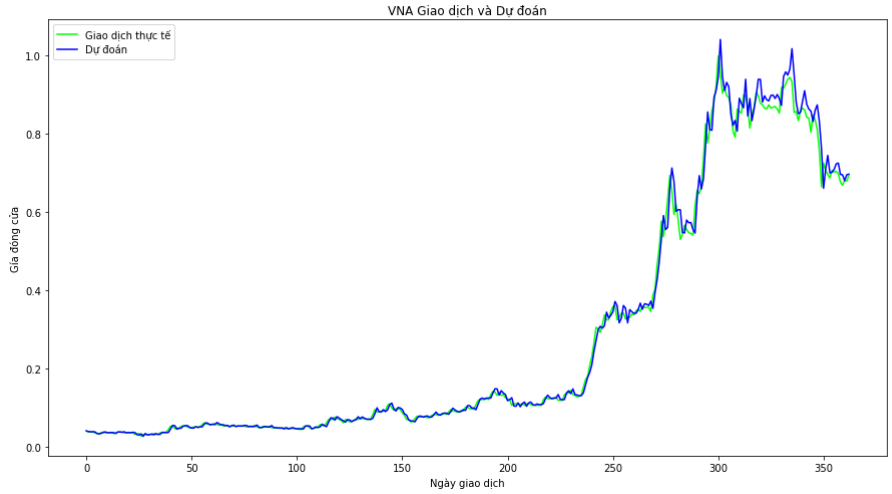
- **LSTM**

```
Train Score: 0.0007 MSE (0.0260 RMSE)
Val Score: 0.00004554 MSE (0.00674866 RMSE)
Test Score: 0.00059399 MSE (0.02437191 RMSE)
```

b) Biểu đồ trên tập Test

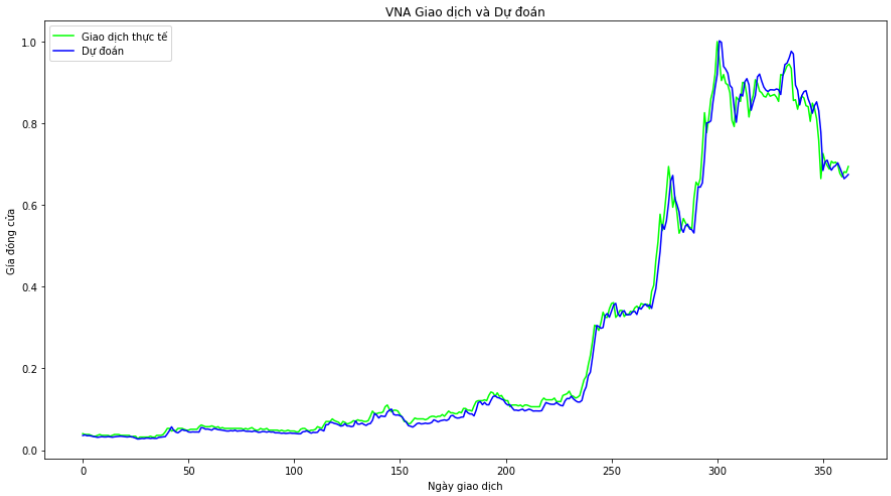
- **Linear**

LINEAR



- **LSTM**

LSTM



c) Nhận xét

Sử dụng xác thực chéo bằng cách đọc ít dữ liệu hơn ta cũng thu được kết quả tương tự khi chạy bộ dữ liệu đầy đủ

Dựa vào độ đo MSE và RMSE của 2 mô hình mà nhóm đã xây dựng, nhóm nhận thấy mô hình Linear cho kết quả tốt hơn so với mô hình LSTM. Mặc dù Linear là một mô hình đơn giản nhưng lại có kết quả khả quan hơn.

Do MSE là kết quả trung bình trên tập dự đoán so với thực tế nên có thể là Linear sẽ cho sai số trung bình trên toàn tập là gần như nhau. Còn mô hình LSTM sẽ cho sai số nhỏ hơn tập trung ở phần đầu và tăng dần sai số về phần sau.

Nhóm đã thử kiểm tra sai số của 10 ngày đầu tiên trên tập Test của 2 mô hình và được kết quả như sau:

	Target	Linear	LSTM	Change Linear	Change LSTM
0	0.040340	0.041116	0.035649	0.000777	-0.004690
1	0.038217	0.039197	0.037202	0.000980	-0.001015
2	0.038217	0.038657	0.035089	0.000441	-0.003127
3	0.038217	0.038738	0.035454	0.000521	-0.002762
4	0.036093	0.038669	0.034546	0.002576	-0.001548
5	0.033970	0.034754	0.033247	0.000784	-0.000723
6	0.033970	0.033141	0.032443	-0.000829	-0.001527
7	0.036093	0.034708	0.031260	-0.001385	-0.004834
8	0.038217	0.036739	0.032024	-0.001478	-0.006193
9	0.036093	0.038390	0.032961	0.002297	-0.003133

Bảng đánh giá dữ liệu dự đoán và thực tế trên 2 mô hình

Nhìn vào bảng kết quả trên, ta thấy Linear vẫn cho kết quả nhưng ngày đầu tốt hơn.

PHẦN V: KẾT LUẬN

Qua việc ứng dụng các mô hình học máy để giải quyết các bài toán dự đoán giá chứng khoán các mô hình học máy đã cho những kết quả dự đoán tốt với độ chính xác cao so với việc dự đoán thủ công thông thường.

Qua đề tải này nhóm em đã phần nào hiểu hơn về các mô hình trong học máy và thấy được ứng dụng của học máy trong cuộc sống đem lại lợi ích nhiều đến như thế nào giúp con người tiết kiệm được rất nhiều thời gian và tiền bạc. Nhóm 8 chúng em xin cảm ơn thầy đã hướng dẫn tận tình trong các bài giảng trên lớp để giúp nhóm chúng em có thể hiểu hơn về các mô hình học máy và có thể hoàn thành project này.

Tài Liệu Tham Khảo

Slide bài giảng môn Học máy và khai phá dữ liệu của thầy Thân Quang Khoát

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

https://nttuan8.com/bai-13-recurrent-neural-network/