

PHÒNG THÍ NGHIỆM TÍNH TOÁN HIỆU NĂNG CAO CÂU LẠC BỘ BIG DATA



**HIGH
PERFORMANCE
COMPUTING**
Laboratory



BIG DATA CLUB

BÁO CÁO BÀI TẬP LỚN SỐ 3

DỰ ĐOÁN GIÁ MỘT SỐ MÃ CHỨNG KHOÁN TẠI VIỆT NAM BẰNG CÁC MÔ HÌNH TIME-SERIES FORECASTING

Nhóm: 8

Sinh viên thực hiện: Huỳnh Tấn Lộc - K20
Phan Phước Minh - K20
Nguyễn Tuấn Minh - K21
Phạm Phú Khang - K21

Thành phố Hồ Chí Minh, Tháng 8/2022

Mục lục

1	Giới thiệu về Time-Series Forecasting	2
1.1	Time-Series	2
1.2	Mô tả và dự đoán	2
1.2.1	Time-Series Analysis	2
1.2.2	Time-Series Forecasting	2
1.3	Các thành phần của Time-Series	3
1.4	Những điều cần quan tâm về việc dự đoán	3
2	Time-Series Forecasting trong dự đoán chứng khoán	4
3	Mô tả Dataset	5
3.1	Dataset HNX	5
3.2	Dataset HNX30	5
3.3	Dataset VNIndex	6
3.4	Dataset VN100	6
3.5	Dataset VN30	7
4	Các thuật toán sử dụng	8
4.1	LSTM	8
4.1.1	Khái niệm và ý tưởng của LSTM	8
4.1.2	Mô hình LSTM	8
4.1.3	Ứng dụng của LSTM	9
4.2	ARIMA	9
4.2.1	Khái niệm và ý tưởng của mô hình ARIMA	9
4.2.2	Mô hình ARIMA	10
5	Kết quả mô hình	11
6	Về web sản phẩm	12
6.1	Sơ lược về website	12
6.2	Hướng dẫn sử dụng	12
6.3	Phản hồi người dùng:	13
7	Hướng phát triển	14

1 Giới thiệu về Time-Series Forecasting

Time-series Forecasting (Dự đoán chuỗi thời gian) là một lĩnh vực quan trọng của Machine Learning và có rất nhiều bài toán dự đoán liên quan đến các thành phần của thời gian. Tuy nhiên Time-Series Forecasting thường bị bỏ qua bởi vì chính các thành phần của thời gian làm cho bài toán về chuỗi thời gian trở nên khó khăn hơn để có thể xử lý được.

1.1 Time-Series

Một dataset (tập dữ liệu) thông thường trong Machine Learning là tập hợp của các observation (quan sát).

Observation #1
Observation #2
Observation #3

Tuy nhiên một time-series dataset (tập dữ liệu chuỗi thời gian) thì lại khác. Time-series thêm vào một sự ràng buộc thứ tự giữa các observation với nhau: một chiều không gian. Chiều không gian này vừa là một hạn chế vừa là một cấu trúc để cung cấp nguồn thông tin bổ sung.

Hay nói đơn giản, một time-series (chuỗi thời gian) là một chuỗi các observation được sắp xếp theo thứ tự thời gian một cách liên tục. Time-series bao gồm 2 thành phần: Thời gian và observation của mốc thời gian đó.

Time #1, Observation
Time #2, Observation
Time #3, Observation

1.2 Mô tả và dự đoán

Trước khi đi vào time-series forecasting (dự đoán chuỗi thời gian), chúng ta có thể thực hiện mô tả dataset tùy vào mục đích cuối cùng của chúng ta. Hiểu một dataset, hay còn gọi là time-series analysis (phân tích chuỗi thời gian) có thể giúp đưa ra một dự đoán tốt hơn, tuy nhiên đó là một điều không bắt buộc và đôi khi có thể dẫn đến lãng phí thời gian không cần thiết.

1.2.1 Time-Series Analysis

Trong Time-Series analysis, một Time-Series sẽ được mô hình hóa để xác định các thành phần của nó như xu hướng, quan hệ với các yếu tố bên ngoài,... Time-Series analysis liên quan đến việc phát triển các mô hình nắm bắt hoặc mô tả một Time-Series.

Time-Series analysis trả lời cho câu hỏi “vì sao” của một Time-Series dataset, nó thường liên quan đến việc đưa ra các giả định về hình thức của dữ liệu và phân tích Time-Series thành các thành phần con.

Mục tiêu chính của Time-Series analysis là phát triển các mô hình toán học cung cấp các mô tả hợp lý từ dữ liệu mẫu.

1.2.2 Time-Series Forecasting

Time-Series forecasting sử dụng các thông tin từ Time-Series (có thể có các thông tin thêm khác) để dự đoán giá trị tương lai của Time-Series đó.

Đưa ra dự đoán về tương lai được gọi là ngoại suy trong xử lý thống kê một Time-Series data. Các lĩnh vực hiện đại tập trung hơn và chủ đề và coi nó là Time-Series forecasting. Dự đoán liên quan đến việc lấy các mô hình phù hợp với observation trong quá khứ và dùng chúng để đưa ra các observation trong tương lai.

Các mô hình mô tả có thể vay mượn từ tương lai (tức là để làm mịn hoặc loại bỏ nhiễu), chúng chỉ tìm cách mô tả tốt nhất dữ liệu. Một điểm khác biệt quan trọng trong dự báo là tương lai là hoàn toàn không có sẵn và chỉ được ước tính từ những gì đã xảy ra.

Kỹ năng của mô hình dự báo chuỗi thời gian được xác định bởi hiệu suất của nó khi dự đoán tương lai. Điều này thường được thể hiện bằng việc có thể giải thích tại sao một dự đoán cụ thể lại thực hiện, khoảng tin cậy và thậm chí hiểu rõ hơn những nguyên nhân cơ bản đằng sau vấn đề.

1.3 Các thành phần của Time-Series

Time-Series analysis có thể cung cấp cho ta một tập hợp các kỹ thuật để hiểu tốt hơn về dataset. Hữu ích nhất trong chúng là việc phân chia Time-Series thành 4 thành phần:

- Level (mức độ): giá trị đường cơ sở của chuỗi nếu nó là đường thẳng.
- Trend (tính xu hướng): xu hướng của dữ liệu (tăng hoặc giảm).
- Seasonality (tính mùa vụ): sự lặp lại một hành vi theo chu kỳ của chuỗi theo thời gian.
- Noise (nhiều): sự thay đổi ngẫu nhiên của các observation không thể giải thích bằng mô hình, chỉ ra sự bất thường của dữ liệu.

Tất cả Time-Series đều có level, đa số có noise, trend và seasonality là tùy chọn.

1.4 Những điều cần quan tâm về việc dự đoán

Trong lúc dự đoán, hiểu được mục đích cuối cùng là rất quan trọng. Sử dụng Socratic method và hỏi nhiều câu hỏi có thể giúp làm rõ các chi tiết cụ thể của bài toán. Có thể ví dụ như:

- Có bao nhiêu dữ liệu chúng ta đã có sẵn và liệu ta có thể tập hợp chúng lại?
- Liệu việc dự đoán có thể được cập nhật thường xuyên hay chỉ thực hiện một lần?
- Việc dự báo được yêu cầu với tần suất tạm thời như thế nào?

Dữ liệu về Time-Series thường yêu cầu: cleaning, scaling, và transformation. Ví dụ như:

- Frequency (tần suất): Có thể dữ liệu được cung cấp ở một tần suất quá cao hoặc có khoảng cách không đều theo thời gian lấy mẫu trong một số mô hình.
- Outliers (ngoại lệ): có thể có các giá trị bị hỏng hoặc ngoại lệ quá lớn cần được xác định và xử lý
- Missing (mất mát): có thể có những khoảng trống hoặc dữ liệu bị mất cần được lấp đầy.

2 Time-Series Forecasting trong dự đoán chứng khoán

Dự báo thị trường chứng khoán là một hành vi nhằm xác định giá trị tương lai của cổ phiếu doanh nghiệp hoặc các công cụ tài chính khác được giao dịch trên các sàn giao dịch. Dự báo thành công về giá cổ phiếu trong tương lai có thể tạo ra lợi nhuận đáng kể. Theo EMH (efficiency market hypothesis), giá cổ phiếu phản ánh tất cả thông tin hiện có, do đó không thể dự báo bất kỳ sự thay đổi giá trị nào không dựa trên thông tin mới được công bố. Mặc dù những người khác không đồng ý với giả thuyết này, nhưng một số người ủng hộ quan điểm này nắm giữ vô số phương pháp và kỹ thuật được cho là cho phép họ truy cập vào thông tin giá trị cổ phiếu trong tương lai.



Dự báo thị trường chứng khoán đặc biệt khó khăn, do tính chất phi tuyến, biến động và phức tạp của thị trường. Trước khi xuất hiện công nghệ machine learning, các dự báo về thị trường chứng khoán thường được thực hiện thông qua phân tích cơ bản và kỹ thuật. Với công nghệ máy tính, chẳng hạn như machine learning, xuất hiện và phát triển trong kinh doanh, deep learning, đặc biệt là mô hình neural network, đã trở thành điểm nóng hiện nay của mô hình dự đoán chứng khoán. Trong khi đó, dự báo thị trường chứng khoán thuận tiện và hiệu quả hơn nhờ các công nghệ này. Hiện tại, các mô hình dự báo chứng khoán thường rơi vào các mô hình tuyến tính truyền thống và các mô hình được đại diện bởi deep learning. Tuy nhiên, vì time-series data có cả phần tuyến tính và phần phi tuyến, nên các kết quả dự báo đơn lẻ thông qua các mô hình dự báo thường không đáng tin cậy như vậy. Do đó, nhiều chuyên gia và học giả kết hợp nhiều mô hình đơn lẻ khác nhau để cải thiện đáng kể độ chính xác và ổn định của các kết quả dự báo.

3 Mô tả Dataset

Dữ liệu về chứng khoán của nhóm được lấy từ trang <https://www.investing.com/indices/vn> với 5 bộ Data như sau

3.1 Dataset HNX



Dữ liệu sẽ là chỉ số giá HNX được lấy trong 13 năm, từ 02/01/2009 đến 29/07/2022. Sau khi tải dữ liệu từ file csv và tiền xử lý trong Dataframe, dữ liệu sẽ trông như sau:

Tập dữ liệu dùng để huấn luyện sẽ là 90% của tập dữ liệu gốc và được scale về khoảng (0, 1) trước khi bắt đầu. Tập huấn luyện sẽ gồm y_{train} gồm các giá trị cần dự đoán và tương ứng là x_{train} với mỗi điểm dữ liệu sẽ là 60 giá trị trước giá trị cần dự đoán.

3.2 Dataset HNX30



Dữ liệu sẽ là chỉ số giá HNX30 được lấy trong 8 năm, từ 04/11/2014 đến 29/07/2022. Sau khi tải dữ liệu từ file csv và tiền xử lý trong Dataframe, dữ liệu sẽ trông như sau:

Tập dữ liệu dùng để huấn luyện sẽ là 90% của tập dữ liệu gốc và được scale về khoảng (0, 1) trước khi bắt đầu. Tập huấn luyện sẽ gồm y_{train} gồm các giá trị cần dự đoán và tương ứng là x_{train} với mỗi điểm dữ liệu sẽ là 60 giá trị trước giá trị cần dự đoán.

3.3 Dataset VNIndex



Dữ liệu sẽ là chỉ số giá HNX30 được lấy trong 13 năm, từ 02/01/2009 đến 29/07/2022. Sau khi tải dữ liệu từ file csv và tiền xử lý trong Dataframe, dữ liệu sẽ trông như sau:

Tập dữ liệu dùng để huấn luyện sẽ là 90% của tập dữ liệu gốc và được scale về khoảng (0, 1) trước khi bắt đầu. Tập huấn luyện sẽ gồm y_{train} gồm các giá trị cần dự đoán và tương ứng là x_{train} với mỗi điểm dữ liệu sẽ là 60 giá trị trước giá trị cần dự đoán.

3.4 Dataset VN100



Dữ liệu sẽ là chỉ số giá VN100 được lấy trong 8 năm, từ 05/11/2014 đến 29/07/2022. Sau khi tải dữ liệu từ file csv và tiền xử lý trong Dataframe, dữ liệu sẽ trông như sau:

Tập dữ liệu dùng để huấn luyện sẽ là 90% của tập dữ liệu gốc và được scale về khoảng (0, 1) trước khi bắt đầu. Tập huấn luyện sẽ gồm y_{train} gồm các giá trị cần dự đoán và tương ứng là x_{train} với

mỗi điểm dữ liệu sẽ là 60 giá trị trước giá trị cần dự đoán.

3.5 Dataset VN30



Dữ liệu sẽ là chỉ số giá VN Index được lấy trong 3 năm, từ 02/01/2019 đến 29/07/2022. Sau khi tải dữ liệu từ file csv và tiền xử lí trong Dataframe, dữ liệu sẽ trông như sau:

Tập dữ liệu dùng để huấn luyện sẽ là 90% của tập dữ liệu gốc và được scale về khoảng (0, 1) trước khi bắt đầu. Tập huấn luyện sẽ gồm y_{train} gồm các giá trị cần dự đoán và tương ứng là x_{train} với mỗi điểm dữ liệu sẽ là 60 giá trị trước giá trị cần dự đoán.

4 Các thuật toán sử dụng

4.1 LSTM

4.1.1 Khái niệm và ý tưởng của LSTM

Long Short-Term Memory (LSTM) là một mạng nơ-ron nhân tạo dùng trong lĩnh vực trí tuệ nhân tạo (AI) và học sâu (Deep learning). Không như các mạng nơ-ron truyền thẳng tiêu chuẩn (feedforward neural network), LSTM mang bản chất là một mạng nơ-ron hồi quy (Recurrent Neural Network - RNN). Đồng nghĩa với việc, LSTM có các kết nối phản hồi.

Các RNN nói chung và LSTM nói riêng được dùng để xử lý thông tin dạng chuỗi (sequence/time-series). Một RNN như vậy có thể xử lý không chỉ một điểm dữ liệu (như là bức ảnh) mà toàn bộ chuỗi dữ liệu (chẳng hạn video - một chuỗi các bức ảnh).

Theo lý thuyết, một RNN cổ điển có thể theo dõi sự phụ thuộc dài hạn trong chuỗi đầu vào. Tuy nhiên vẫn đề với nó lại nằm ở bản chất tính toán: khi huấn luyện một RNN cổ điển dùng thuật toán lan truyền ngược (back-propagation), Gradient dài hạn được tính toán bằng lan truyền ngược có thể "biến mất" (tức dần tiến về 0) hoặc "phát nổ" (tiến tới vô cùng). Nguyên do là bởi các tính toán liên quan trong quá trình mà dùng các số được ước lượng thiếu chính xác.

Như vậy về lý thuyết là RNN có thể mang thông tin từ các lớp trước đến các lớp sau, nhưng thực tế là thông tin chỉ mang được qua một số lượng trạng thái nhất định, sau đó thì sẽ bị "vanishing gradient", hay nói cách khác là model RNN cổ điển chỉ học được từ các trạng thái gần nó (short-term memory). Đó là lý do Long short-term Memory (LSTM) ra đời.

4.1.2 Mô hình LSTM

Một đơn vị LSTM thông dụng bao gồm các thành phần: cell, input gate, output gate và forget gate. Cell được dùng để ghi nhớ các giá trị xuyên suốt các khoảng thời gian và ba cổng (gate) sẽ điều phối dòng chảy của thông tin vào và ra các cell.

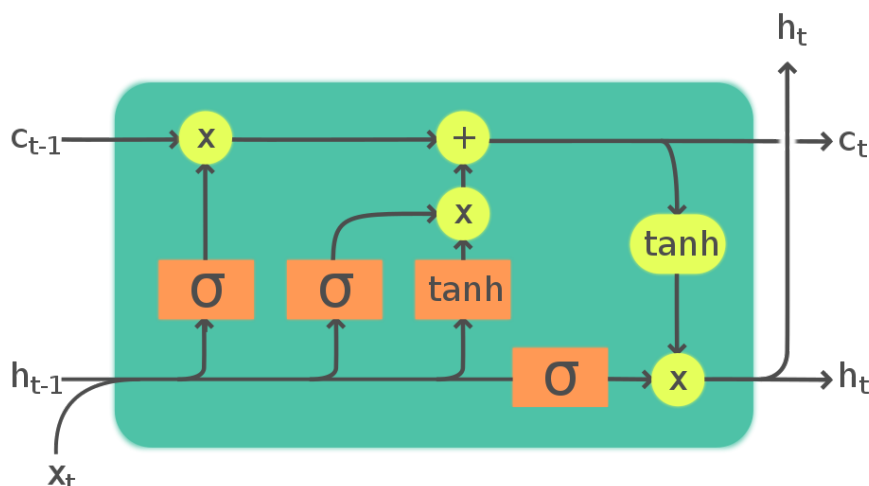
Dạng chuẩn của các phương trình khi truyền vào của đơn vị LSTM được cho như sau:

- $f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$
- $i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$
- $o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$
- $\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$
- $c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$
- $h_t = o_t \circ \sigma_h(c_t)$





Trong đó:

- t : bước thời gian
- Toán tử \circ : tích Hadamard (tích các phần tử với nhau)
- $x_t \in \mathbb{R}^d$: vector đầu vào của đơn vị LSTM
- $f_t \in (0, 1)^h$: vector kích hoạt của forget gate
- $i_t \in (0, 1)^h$: vector kích hoạt của input gate
- $o_t \in (0, 1)^h$: vector kích hoạt của output gate
- $h_t \in (-1, 1)^h$: vector của hidden state, là vector đầu ra của 1 đơn vị LSTM
- $\tilde{c}_t \in (-1, 1)^h$: vector kích hoạt của đầu vào một cell
- $c_t \in \mathbb{R}^h$: vector trạng thái một cell
- $W \in \mathbb{R}^{h \times d}, U \in \mathbb{R}^{h \times h}, b \in \mathbb{R}^h$: ma trận trọng số và các tham số vector bias cần được học lúc huấn luyện

- d, h : số lượng input features và số lượng hidden units
- σ_g : hàm sigmoid
- σ_c : hàm tanh
- σ_h : hàm tanh hoặc hàm $\sigma_h(x) = x$



Legend:

Layer	Componentwise	Copy	Concatenate
			

Hình 1: Mô hình LSTM

4.1.3 Ứng dụng của LSTM

Về cơ bản, khi áp dụng thuật toán back propagation cho LSTM tương tự như RNN, thì LSTM vẫn bị vanishing gradient nhưng bị ít hơn nhiều so với RNN. Do đó LSTM được dùng phổ biến hơn RNN cho các bài toán thông tin dạng chuỗi. Một mạng LSTM rất thích hợp để dùng trong các tác vụ phân loại, xử lý và dự đoán dựa trên dữ liệu time-series (như dữ liệu chứng khoán mà nhóm sẽ trình bày trong phần sau).

Khả năng ít nhạy cảm với độ dài khoảng trống giúp LSTM có lợi thế hơn so với RNN, các mô hình hidden Markov và các phương pháp học tuần tự trong nhiều ứng dụng. Có thể kể đến một số ứng dụng phổ biến của LSTM như: dự đoán chứng khoán, điều khiển robot, nhận dạng hành động, nhận dạng giọng nói và chăm sóc sức khỏe.

4.2 ARIMA

4.2.1 Khái niệm và ý tưởng của mô hình ARIMA

Chúng ta biết rằng hầu hết các chuỗi thời gian đều có sự tương quan giữa giá trị trong quá khứ đến giá trị hiện tại. Mức độ tương quan càng lớn khi chuỗi càng gần thời điểm hiện tại. Chính vì thế mô hình ARIMA sẽ tìm cách đưa vào các biến trễ nhằm tạo ra một mô hình dự báo fitting tốt hơn giá trị của chuỗi. ARIMA model là viết tắt của cụm từ Autoregressive Integrated Moving Average. Mô hình sẽ biểu diễn phương trình hồi qui tuyến tính đa biến (multiple linear regression) của các biến đầu vào (còn gọi là biến phụ thuộc trong thống kê).

4.2.2 Mô hình ARIMA

Mô hình ARIMA gồm có 3 thành phần chính:

- **Auto regression:** Kí hiệu là AR. Đây là thành phần tự hồi qui bao gồm tập hợp các độ trễ của biến hiện tại. Độ trễ bậc p chính là giá trị lùi về quá khứ p bước thời gian của chuỗi. Độ trễ dài hoặc ngắn trong quá trình AR phụ thuộc vào tham số trễ p . Cụ thể, quá trình $AR(p)$ của chuỗi x_t được biểu diễn như sau:

$$AR(p) = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p}$$

- **Moving average:** Quá trình trung bình trượt được hiểu là quá trình dịch chuyển hoặc thay đổi giá trị trung bình của chuỗi theo thời gian. Do chuỗi của chúng ta được giả định là dừng nên quá trình thay đổi trung bình dường như là một chuỗi nhiễu trắng. Quá trình moving average sẽ tìm mối liên hệ về mặt tuyến tính giữa các phần tử ngẫu nhiên ε_t (stochastic term). Chuỗi này phải là một chuỗi nhiễu trắng thỏa mãn các tính chất:

$$\begin{cases} E(\varepsilon_t) = 0 & (1) \\ \sigma(\varepsilon_t) = \alpha & (2) \\ \rho(\varepsilon_t, \varepsilon_{t-s}) = 0, \forall s \leq t & (3) \end{cases}$$

Về (1) có nghĩa rằng kì vọng của chuỗi bằng 0 để đảm bảo chuỗi dừng không có sự thay đổi về trung bình theo thời gian. Về (2) là phương sai của chuỗi không đổi. Do kì vọng và phương sai không đổi nên chúng ta gọi phân phối của nhiễu trắng là phân phối xác định (identical distribution) và được kí hiệu là:

$$\varepsilon_t \sim WN(0, \sigma^2)$$

Nhiều trắng là một thành phần ngẫu nhiên thể hiện cho yếu tố không thể dự báo của model và không có tính qui luật. Quá trình trung bình trượt được biểu diễn theo nhiễu trắng như sau:

$$MA(q) = \mu + \sum_{i=1}^q \theta_i \varepsilon_{t-i}$$

Quá trình này có thể được biểu diễn theo dịch chuyển trễ - backshift operator B như sau:

$$MA(q) = \mu + (1 + \theta_1 B + \dots + \theta_q B^q) \varepsilon_t$$

- **Intergrated:** Là quá trình đồng tích hợp hoặc lấy sai phân. Yêu cầu chung của các thuật toán trong time series là chuỗi phải đảm bảo tính dừng. Hầu hết các chuỗi đều tăng hoặc giảm theo thời gian. Do đó yếu tố tương quan giữa chúng chưa chắc là thực sự mà là do chúng cùng tương quan theo thời gian. Khi biến đổi sang chuỗi dừng, các nhân tố ảnh hưởng thời gian được loại bỏ và chuỗi sẽ dễ dự báo hơn. Để tạo thành chuỗi dừng, một phương pháp đơn giản nhất là chúng ta sẽ lấy sai phân. Một số chuỗi tài chính còn qui đổi sang logarit hoặc lợi suất. Bậc của sai phân để tạo thành chuỗi dừng còn gọi là bậc của quá trình đồng tích hợp (order of intergration). Quá trình sai phân bậc d của chuỗi được thực hiện như sau:

- Sai phân bậc 1: $I(1) = \Delta(x_t) = x_t - x_{t-1}$.
- Sai phân bậc d : $I(d) = \Delta^d(x_t) = \Delta(\Delta(\dots\Delta(x_t)))$.
t times

Thông thường chuỗi sẽ dừng sau quá trình đồng tích hợp $I(0)$ hoặc $I(1)$. Rất ít chuỗi chúng ta phải lấy tới sai phân bậc 2. Một số trường hợp chúng ta sẽ cần biến đổi logarit hoặc căn bậc 2 để tạo thành chuỗi dừng. Phương trình hồi quy $ARIMA(p, d, q)$ có thể được biểu diễn dưới dạng:

$$\Delta x_t = \phi_1 \Delta x_{t-1} + \phi_2 \Delta x_{t-2} + \dots + \phi_p \Delta x_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q}$$

Trong đó Δx_t là giá trị sai phân bậc d và ε_t là các chuỗi nhiễu trắng.

Như vậy về tổng quát thì ARIMA là mô hình kết hợp của 2 quá trình tự hồi qui và trung bình trượt. Dữ liệu trong quá khứ sẽ được sử dụng để dự báo dữ liệu trong tương lai. Trước khi huấn luyện mô hình, cần chuyển hóa chuỗi sang chuỗi dừng bằng cách lấy sai phân bậc 1 hoặc logarit. Ngoài ra mô hình cũng cần tuân thủ điều kiện ngặt về sai số không có hiện tượng tự tương quan và phần dư là nhiễu trắng.

5 Kết quả mô hình

Trong Bài tập lớn này, nhóm lựa chọn dùng RMSE (Root Mean Square Error) làm tiêu chuẩn để đánh giá. RMSE được tính bằng phương pháp sau:

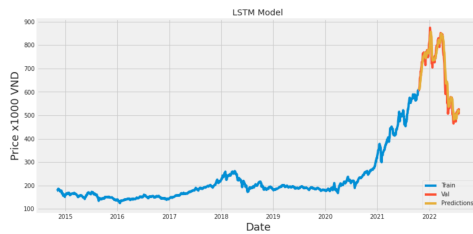
$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}}$$

Trong đó:

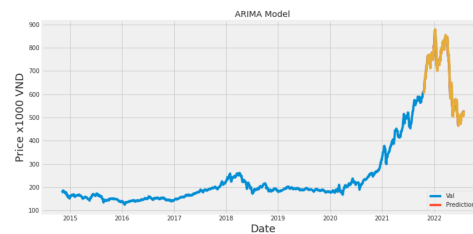
- \hat{y}_i là giá trị ước lượng.
- y_i là giá trị của biến thực tế.
- n là kích thước mẫu.

Tập dữ liệu đại diện nhóm sử dụng là HNX30 đã được đề cập ở trên.

So sánh kết quả của 2 mô hình LSTM và ARIMA như sau:



(a) Mô hình LSTM



(b) Mô hình ARIMA

Bằng mắt thường thì ta có thể thấy gần như tương tự nhau.

Dùng tiêu chuẩn RMSE thì ta có được kết quả sau:

```
[951] rmse = sqrt(mean_squared_error(y_test, predictions))
      print('Test RMSE - LSTM: %.3f' % rmse)

Test RMSE - LSTM: 26.699
```

(a) Mô hình LSTM

```
[958] ## ARIMA MODEL
      rmse = sqrt(mean_squared_error(test, predictions))
      print('Test RMSE - ARIMA: %.3f' % rmse)

Test RMSE - ARIMA: 15.700
```

(b) Mô hình ARIMA

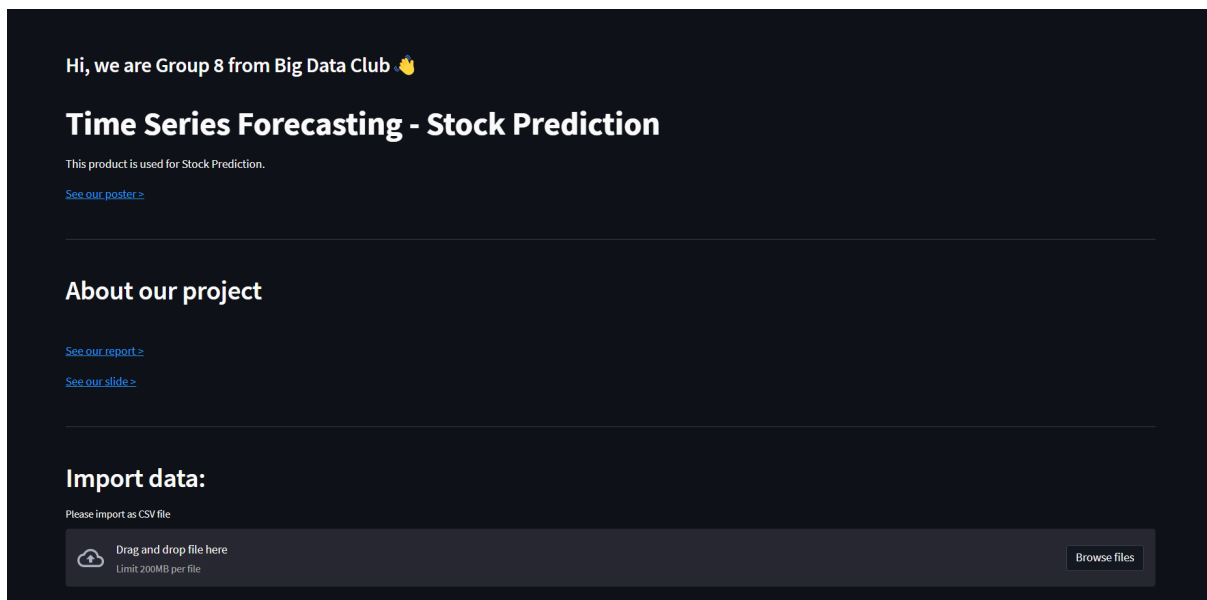
Ta có thể nhận thấy giữa 2 model không có sự chênh lệch quá lớn về kết quả, tuy nhiên ARIMA model có sự chính xác cao hơn, và điều này cũng tương tự với các tập dữ liệu khác. Dù vậy, ARIMA model có thời gian train lâu hơn so với LSTM với layer ít node nhưng điều này không là điểm quan trọng do trong khuôn khổ Bài tập lớn này không quá cần thiết phải train quá nhanh mà cần đảm bảo độ chính xác.

Khi đưa vào ứng dụng, nhóm sẽ thực hiện kiểm thử về độ chính xác của nhiều mô hình khác nhau, mô hình nào đạt hiệu quả cao hơn sẽ được sử dụng để dự đoán giá trị.

6 Về web sản phẩm

6.1 Sơ lược về website

Hiện tại, nhóm đã triển khai sản phẩm dưới dạng một trang web. Vì hạn chế về tài nguyên, web đang được deploy ở local của thành viên trong nhóm. Giao diện web như sau:

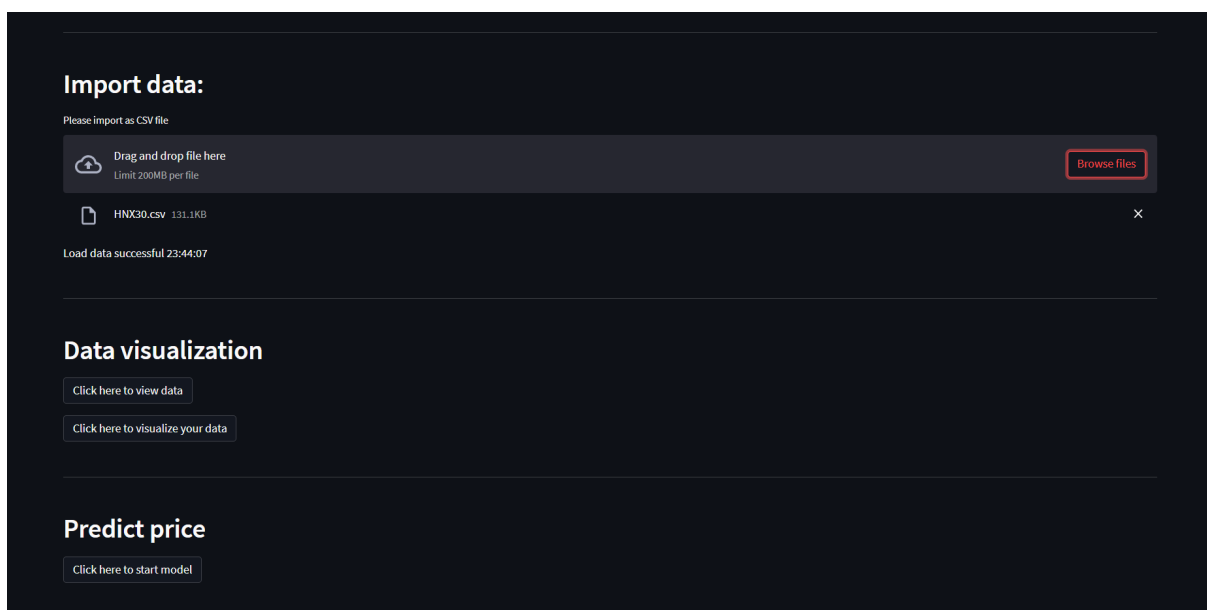


Hình 4: Giao diện website

6.2 Hướng dẫn sử dụng

Để sử dụng website, đầu tiên người dùng cần tải lên file có định dạng `".csv"` bằng cách nhấn vào Browse file ở phần **Import data**.

Sau khi tải lên bộ dữ liệu, giao diện sẽ như sau:



Hình 5: Giao diện sau khi đã tải data lên

Có các lựa chọn cho người dùng như sau:

- View data: Data sau khi được tiền xử lý chỉ còn gồm ngày và giá, người dùng click vào button "*Click here to view data*" để xem về lịch sử giá chứng khoán.
- Visualize data: Bộ dữ liệu sẽ được trực quan hóa thành biểu đồ, người dùng click vào button "*Click here to visualize your data*" để sử dụng.
- Predict model: Đây là phần chính của sản phẩm. Người dùng click vào button "*Build model*" để khởi động model. Hệ thống sẽ tiến hành dự đoán kết quả thông qua hai mô hình là LSTM và ARIMA sau đó trả về biểu đồ dự đoán cho người dùng.

6.3 Phản hồi người dùng:

Người dùng có thể liên lạc với nhóm thông qua phần Feedback theo form:



The image shows a feedback form on a dark background. The title 'Give us your feedback!' is in white. Below it are three white input fields: 'Your name', 'Your email', and 'Your message here'. At the bottom left is an orange 'Send' button.

Hình 6: *Feedback form*

7 Hướng phát triển

Đây chỉ là hai mô hình đơn giản, trong tương lai, nếu có thời gian thì nhóm sẽ hiện thực các mô hình phức tạp hơn.

Nhóm cũng sẽ xem xét một số phương pháp kết hợp mô hình LSTM và ARIMA lại với nhau để tận dụng tính ưu việt và ưu điểm của nhau để tăng độ chính xác cho kết quả dự đoán.

Nhóm sẽ có thể triển khai trang web online và ứng dụng cho người dùng thực tế để đánh giá được mức độ hoàn thiện của đề tài để đưa ra những giải pháp tốt hơn.



Hình 7: Giá cổ phiếu và chứng khoán

Về hạn chế, mô hình chỉ dựa trên những dữ liệu số đã có sẵn, tuy nhiên, sự thay đổi giá chứng khoán phụ thuộc vào rất nhiều yếu tố tác động bên ngoài. Do đó, mô hình không thể dự đoán được kết quả với độ chính xác quá cao khi có những sự kiện đặc biệt diễn ra. Trong tương lai, có thể nhóm sẽ nghiên cứu thêm về các mô hình có thể cho người dùng cập nhật các yếu tố tác động đến giá, hay thiết kế thêm các hàm nội suy để có thể tự suy luận ra những yếu tố sẽ tác động đến giá cả trên thị trường.

Ngoài ra, vì hạn chế tiếp cận, mô hình vẫn đang dừng lại ở mức dự đoán giá sau khi nhận được một dataset tĩnh. Trong tương lai, nhóm sẽ cố gắng triển khai để cập nhật dữ liệu theo thời gian thực (từng ngày) để đưa ra hướng xử lý tốt nhất.

Tài liệu

- [1] Jake VanderPlas - Python Data Science Handbook_ Essential Tools for Working with Data (2016, O'Reilly Media)
- [2] Jason-Brownlee-Introduction-to-Time-Series-Forecasting-with-Python-How-to-Prepare-Data-and-Develop-Models-to-Predict-the-Future-v1.9-2020
- [3] Mô hình ARIMA trong time series: https://phamdinhkhanh.github.io/2019/12/12/ARIMAModel.html?fbclid=IwAR3_tB7XxIKiGFqJuFE7SEvh6Yob1MdAvvrUzgpL0bxq3gd1zZA_lh-QTmo
- [4] Streamlit documentation: <https://docs.streamlit.io/library/get-started>
- [5] Stock Market Predictions with LSTM in Python: <https://www.datacamp.com/tutorial/lstm-python-stock-market>
- [6] Stock Price Prediction and Forecasting using Stacked LSTM <https://www.analyticsvidhya.com/blog/2021/05/stock-price-prediction-and-forecasting-using-stacked-lstm/#:~:text=LSTMs%20are%20widely%20used%20for,the%20information%20that%20is%20not>.

Link github dự án: https://github.com/Prosecutor22/TimeSeriesForecasting_StockPrice.git