Dự đoán sự vận động của đồ thị chứng khoán Amazon bằng LSTM

Trường Đại học Khoa học Tự nhiên

Khoa Công nghệ thông tin

Môn Nhập môn Khoa học dữ liệu

Thành viên

- Nguyễn Bảo Long – 18120201
- Mai Ngọc Tú – 18120253

Thu thập dữ liệu

- Sử dụng thư viện **pandas-datareader 0.9.0**

  Cài đặt: **pip install pandas-datareader**

- Sử dụng

  df = DataReader('AMZN', data_source='yahoo', start='2009-01-01', end='2019-12-31')

- Kết quả

Date	High	Low	Open	Close	Volume	Adj Close
2008-12-31	51.689999	49.910000	50.740002	51.279999	7792200	51.279999
2009-01-02	54.529999	51.070000	51.349998	54.360001	7296400	54.360001
2009-01-05	55.740002	53.029999	55.730000	54.060001	9509800	54.060001
2009-01-06	58.220001	53.750000	54.549999	57.360001	11080100	57.360001

Khám phá dữ liệu

- Open Price: Giá mở bán
- Low Price: Giá thấp nhất trong ngày
- High Price: Giá cao nhất trong ngày
- Close Price: Giá cổ phiếu tại phiên giao dịch cuối cùng
- Adj Close Price: Giá điều chỉnh – được coi là giá trị thực sự của cổ phiếu, thường được dùng trong kiểm tra, phân tích chứng khoán
- Volume: Số lượng cổ phiếu giao dịch trong ngày

Câu hỏi

- Thời điểm thích hợp để mua và bán cổ phiếu để thu lời nhiều nhất?
- Hướng giải quyết:

  Mua vào lúc giá thấp

  Bán ra lúc giá cao
- Cơ sở: Đồ thị biểu diễn giá cổ phiếu

Câu hỏi (tt)

- Làm thế nào để biết được lúc nào giá thấp, lúc nào giá cao?
- Dựa vào:

  Cảm tính

  Ứng dụng LSTM trong việc dự đoán sự tăng giảm về giá cổ phiếu
- Ý nghĩa: Thực hiện đầu tư thành công là thực hiện thành công hành vi tạo ra giá trị thặng dư cá nhân. Giá trị thặng dư này mang lại nguồn động lực dồi dào trong việc phát triển bản thân và đất nước

Tiền xử lý dữ liệu

1. Chia tập huấn luyện, tập validation và tập kiểm tra
2. Chỉ sử dụng dữ liệu tại cột Adj Close Price để dự đoán
3. Chuẩn hóa giá trị của cột này bằng **MinMaxScaler** – giúp đẩy nhanh quá trình học và hội tụ của model
4. Chia tập huấn luyện thành tập input và tập output (sẽ nói kỹ hơn ở phần Mô hình hóa)
5. Tạo **preprocess_pipeline** bao gồm các bước 2, 3, 4 để dễ dàng tiền xử lý cho tập validation và tập test

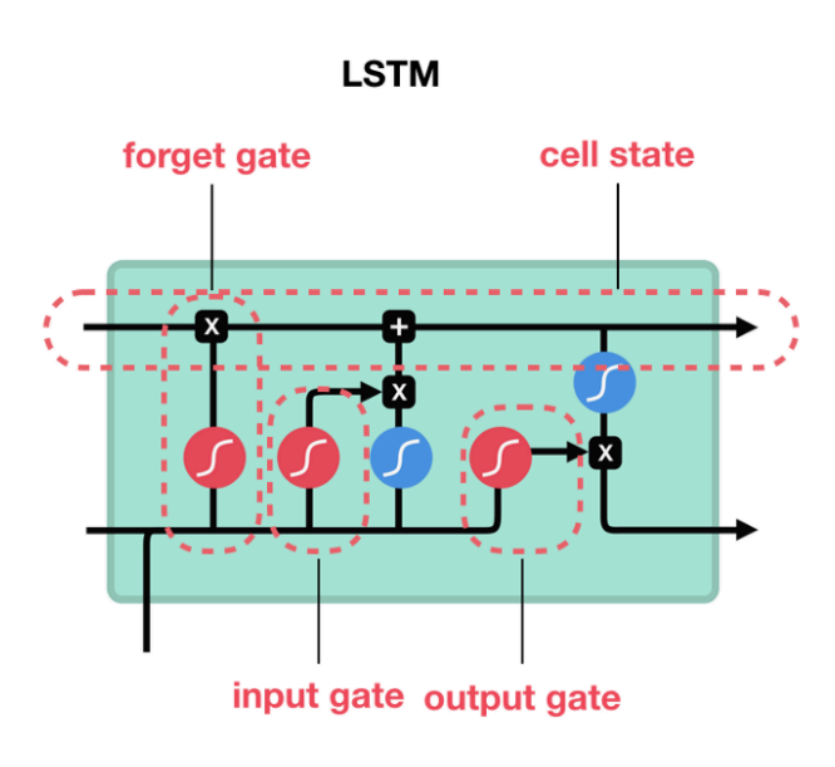
Tại sao dùng LSTM?

- Tính tuần tự của tập dữ liệu: Tập dữ liệu mà trong đó, các mẫu dữ liệu xuất hiện theo một thứ tự (ví dụ như các từ trong 1 văn bản, các nốt nhạc trong 1 bản nhạc,…)
- Mạng NN truyền thống tách dữ liệu ra khỏi ngữ cảnh (các đầu vào độc lập với nhau), không tận dụng tối ưu sự có mặt của tính tuần tự trong tập dữ liệu
- Tính tuần tự mang trong nó 1 thông tin quan trọng ảnh hưởng đến “ngữ nghĩa” của dữ liệu. Cần phải tận dụng thông tin này
- LSTM được sinh ra để xử lý loại dữ liệu này

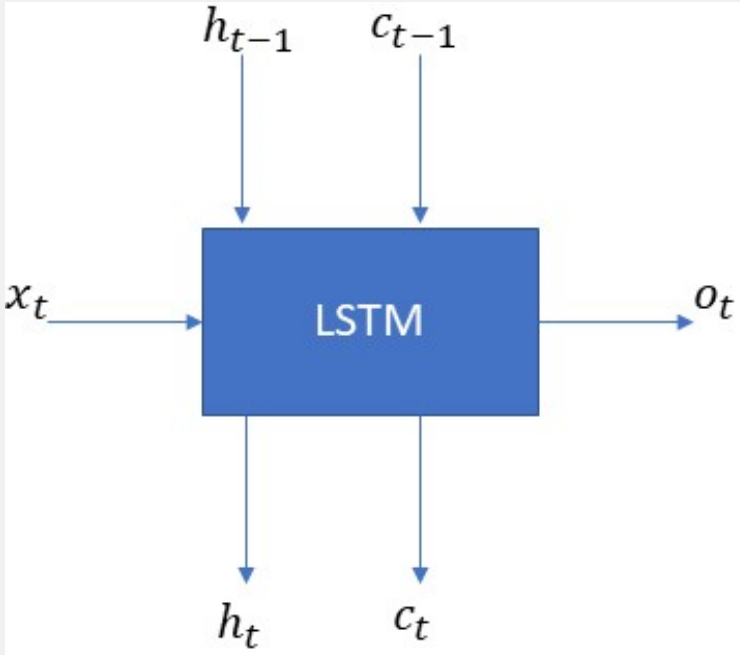
Mô hình hóa – Long Short Term Memory

- Input gate: i_t
- Forget gate: f_t
- Output gate: o_t
- Cell state: c_t
- Ma trận trọng số:

$$W_f, W_i, W_o, W_c$$
$$W_f, W_i, W_o, U_c$$
$$b_f, b_i, b_o, b_c$$



Mô hình hóa – Long Short Term Memory (tt)



σ_g : $sigmoid$

σ_c : tanh

. : element wise multiplication

$$f_t = \sigma_g\,(W_f \times\ x_t + U_f \times h_{t-1} + b_f)$$

$$i_t = \sigma_g\,(W_i \times\ x_t + U_i \times h_{t-1} + b_i)$$

$$o_t = \sigma_g\,(W_o \times\ x_t + U_o \times h_{t-1} + b_o)$$

$$c'_t = \sigma_c\,(W_c \times\ x_t + U_c \times h_{t-1} + b_c)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot c'_t$$

$$h_t = o_t \cdot \sigma_c(c_t)$$

f_t *is the forget gate*

i_t *is the input gate*

o_t *is the output gate*

c_t *is the cell state*

h_t *is the hidden state*

Dữ liệu đầu vào của LSTM

- Dự đoán cho 1 ngày tiếp theo
  - Đầu vào là 60 ngày trước đó
  - Đầu ra là 1 ngày cần dự đoán
- Dự đoán cho 30 ngày tiếp theo
  - Đầu vào là 1500 ngày trước đó
  - Đầu ra là 30 ngày sau đó

Dữ liệu đầu vào của LSTM (tt)

- Dữ liệu train bao gồm tập train_X và train_y
- train_X là 1 mảng numpy 3D có shape là (x, y, z)
- train_y là 1 mảng numpy 2D có shape là (x, t)
- Ý nghĩa của đầu vào

    x: Số mảng có shape là (y, z) được đưa vào

    y: Số dữ liệu đầu vào dùng để dự đoán đầu ra

    z, t: Lượng dữ liệu trên một “dòng” của input và output

Tối ưu trọng số của model

- Giả sử dữ liệu đầu vào có shape (60, 1), đầu ra có shape (30, 1)
- Khi đó

  Các ma trận W phải có shape (x, 60) để có thể nhân được với input

  h và c có shape là (30, 1) để có thể "nhân" với nhau (element wise multiplication)
- Lúc này, f, i, c' sẽ có shape (30, 1)
- Xét cổng forget

  W_f phải có shape (30, 60) vì x có shape (60, 1)

  U_f phải có shape (30, 30) vì h có shape (30, 1)

  b_f phải có shape (30, 1)

Tối ưu trọng số của model (tt)

- Tương tự với những cổng còn lại
- Ma trận trọng số cần phải tối ưu

$$W_f, W_i, W_o, W_c$$

$$W_f, W_i, W_o, U_c$$

$$b_f, b_i, b_o, b_c$$

- Thuật toán tối ưu? **Khó**

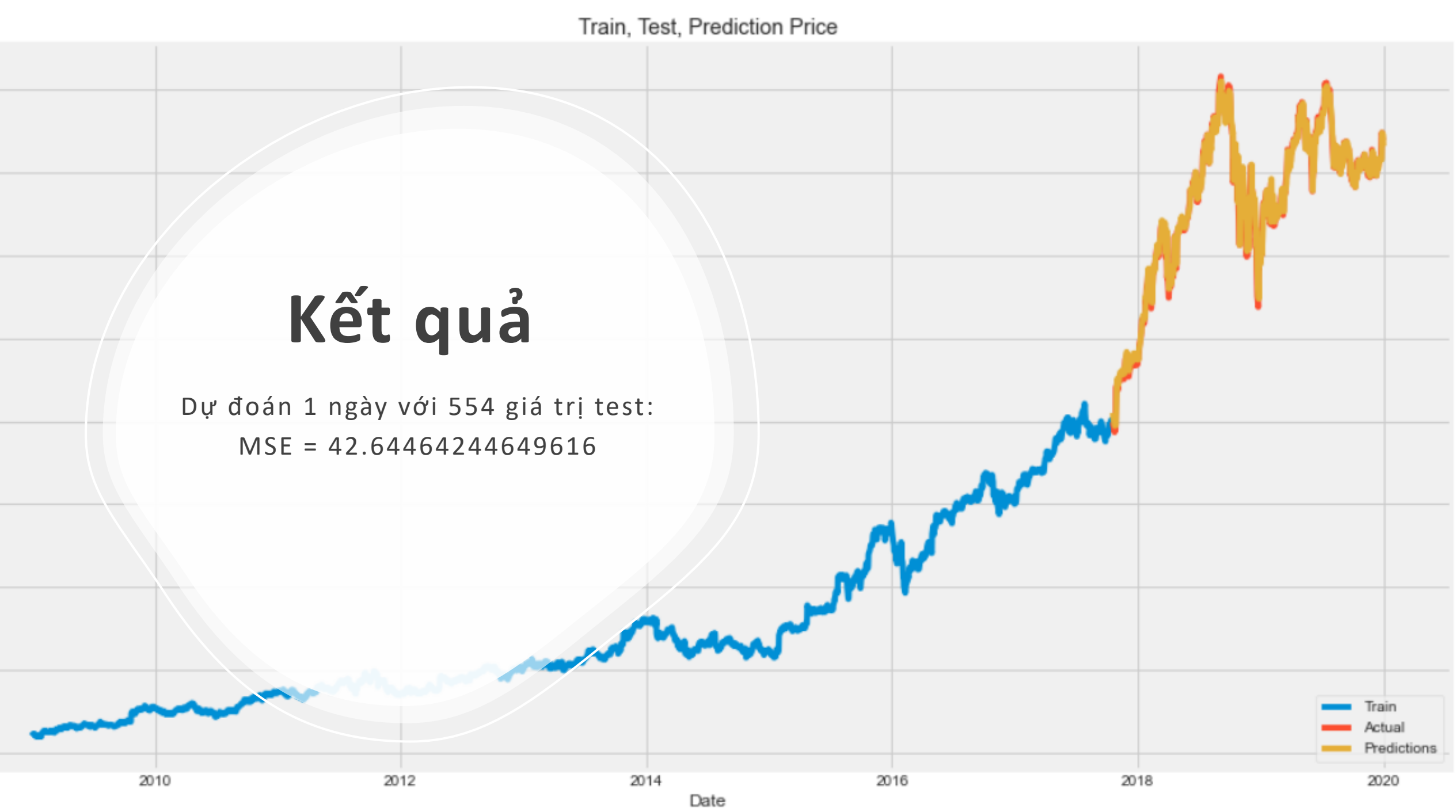
Xây dựng model

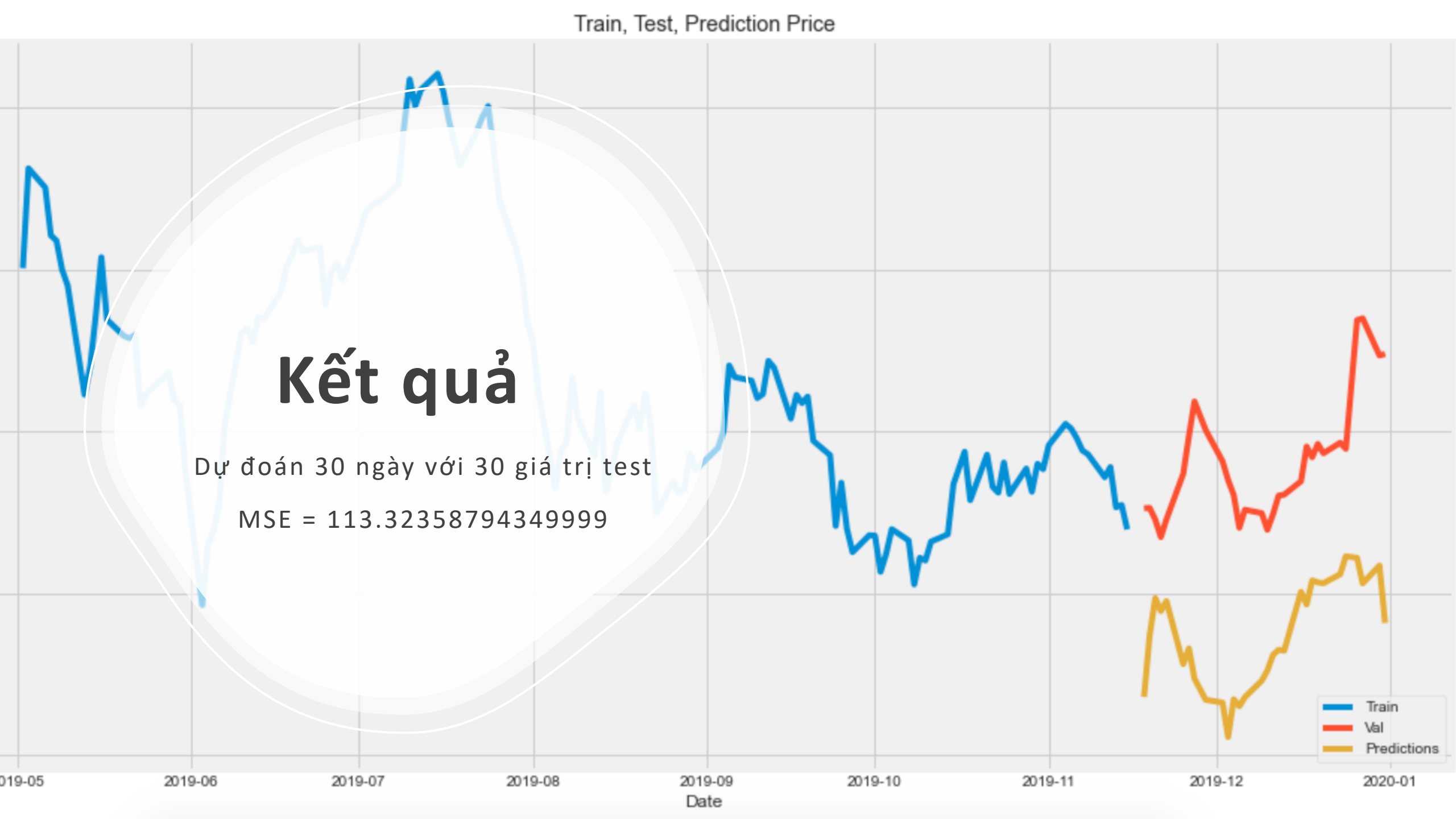
```
model = Sequential()
model.add(LSTM(128, activation='tanh', return_sequences=False,
        input_shape=(train_X.shape[1], 1)))
model.add(Dense(out_length))
model.compile(optimizer='adam', loss='mse')
```

Kết quả

Dự đoán 1 ngày với 554 giá trị test:

MSE = 42.64464244649616




Train, Test, Prediction Price
Kết quả
Dự đoán 30 ngày với 30 giá trị test
MSE = 113.32358794349999
Train
Val
Predictions
019-05
2019-06
2019-07
2019-08
2019-09
2019-10
2019-11
2019-12
2020-01
Date

Khó khăn trong quá trình làm việc

- ❑ Hậu xử lý dữ liệu (phần này đã bị bỏ)
- ❑ Quá trình tìm hiểu cơ sở toán của model (phần này chưa hoàn thiện)
- ❑ Quá trình tìm hiểu cơ sở tối ưu độ lỗi của model (phần này đã bị bỏ)

Tham khảo

- https://towardsdatascience.com/tutorial-on-lstm-a-computational-perspective-f3417442c2cd#:~:text=What%20is%20the%20size%20of,input_dim%2Boutput_dim%2B1)%5D.
- https://dominhhai.github.io/vi/2017/10/what-is-lstm/
- http://cs224d.stanford.edu/lectures/CS224d-Lecture8.pdf
- https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21