Dự đoán giá chứng khoáng của công ty Apple bằng LSTM

Data Science

Vietnam Japan University (VJU)

Bui The Trung - 21110108

Thu nhập dữ liệu (2)

Sử dụng thư viện pandas-datareader 1.16.0

- pip install pandas-datareader

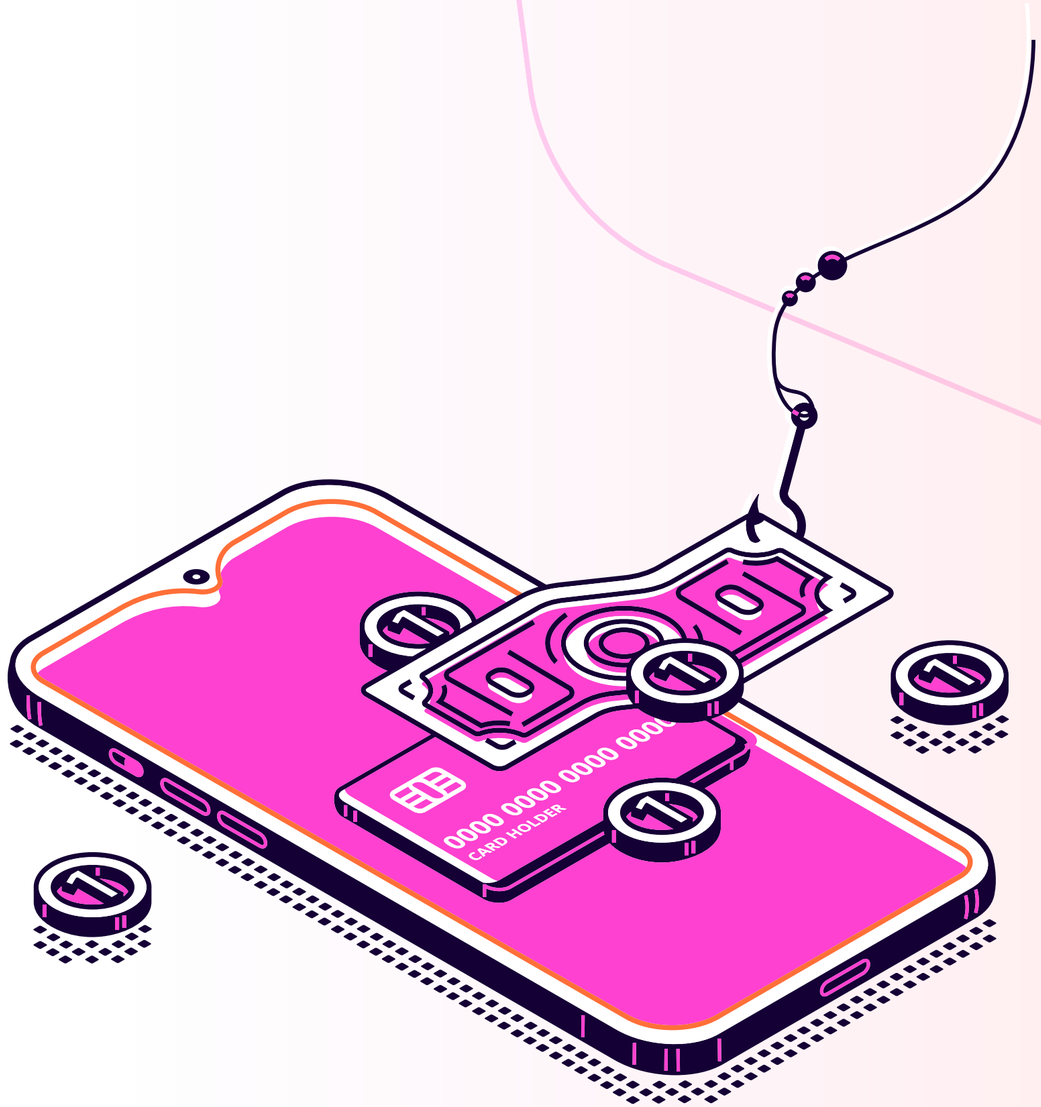
Đọc dữ liệu +

- df = pd.read_csv('Apple_Stock_Prices.csv')

Result:

Date	Open	High	Low	Close	Adj Close	Volume
2005-02-01	1.375893	1.388750	1.367500	1.384464	1.180142	678395200
2005-02-02	1.391964	1.426964	1.387321	1.421964	1.212108	1020062400
2005-02-03	1.412500	1.418393	1.380893	1.389464	1.184404	731651200
2005-02-04	1.390536	1.409464	1.384464	1.407857	1.200083	563556000
2005-02-07	1.409464	1.416964	1.383929	1.409643	1.201605	524456800
...	...	...	...	...	...	...
2023-01-23	138.119995	143.320007	137.899994	141.110001	141.110001	81760300
2023-01-24	140.309998	143.160004	140.300003	142.529999	142.529999	66435100
2023-01-25	140.889999	142.429993	138.809998	141.860001	141.860001	65799300
2023-01-26	143.169998	144.250000	141.899994	143.960007	143.960007	54105100
2023-01-27	143.160004	147.229996	143.080002	145.929993	145.929993	70492800

4529 rows × 6 columns



Các dữ liệu (3)

55%

Open/close price

- **Open price:** Giá mở bán
- **Close price:** Giá cổ phiếu tại phiên giao dịch cuối cùng

High/low price

- **High price:** Giá cao nhất trong ngày
- **Low price:** Giá thấp nhất trong ngày

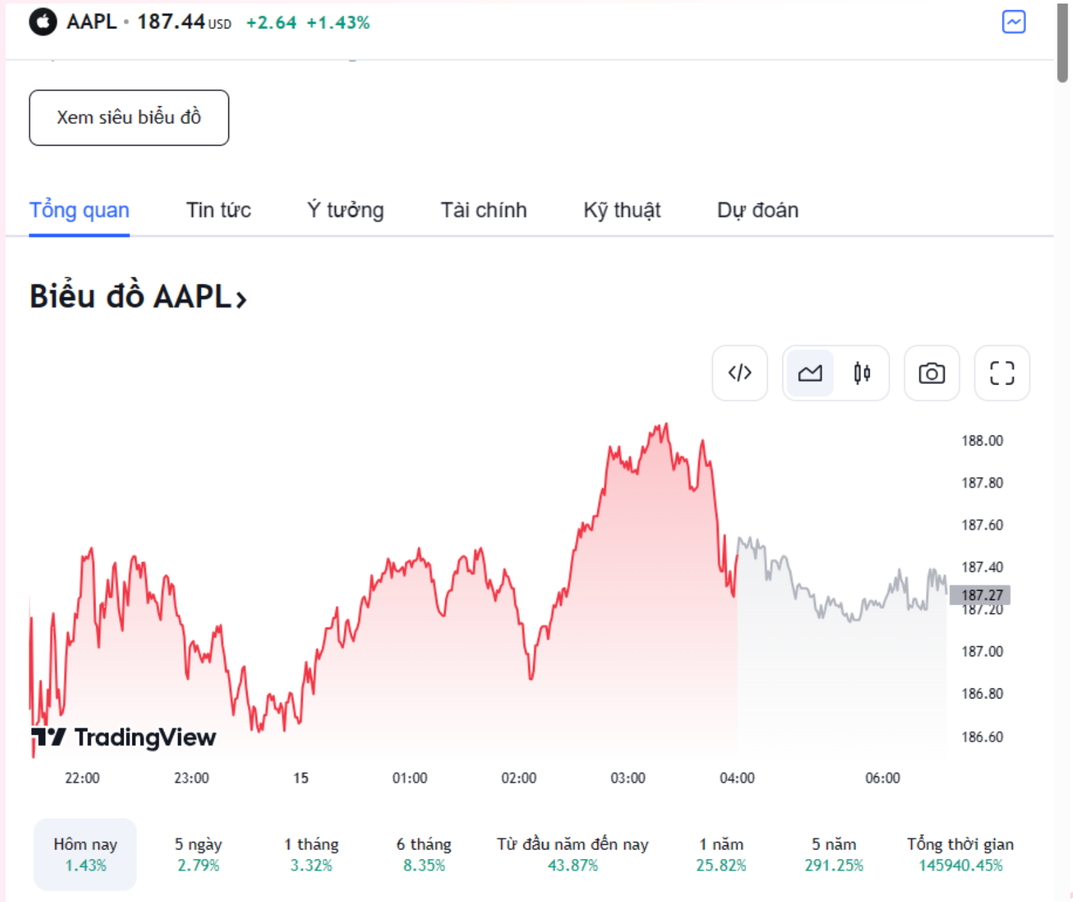
Adj Close price

Giá điều chỉnh - được coi là giá thực sự của cổ phiếu, thường được dùng trong kiểm tra, phân tích chứng khoáng

Volume

Số lượng cổ phiếu dùng trong ngày

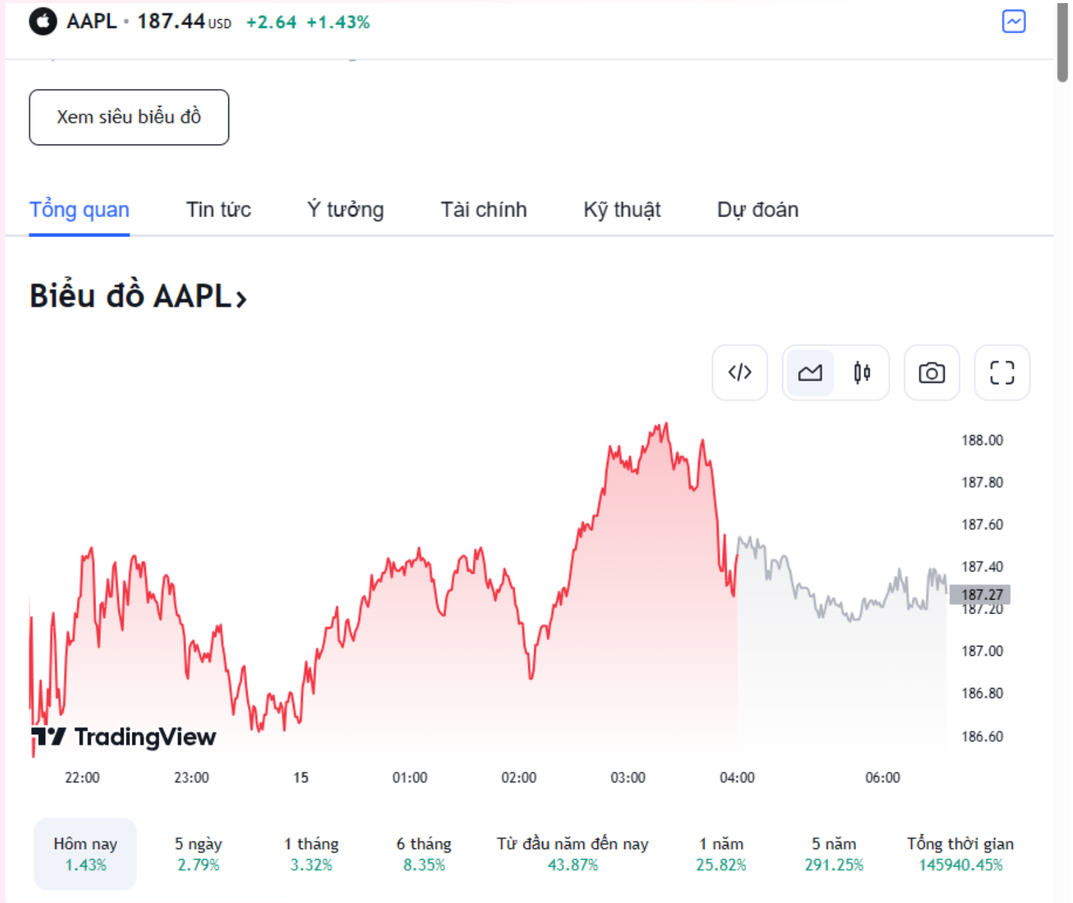
Why choose? (4)



Câu hỏi

- **Thời điểm thích hợp để mua, bán cổ phiếu để thu lời về nhiều nhất?**
- **Solution:**
  - **Mua vào lúc giá thấp**
  - **Bán ra lúc giá cao**
- **Cơ sở: Đồ thị biểu diễn giá cổ phiếu**

Why choose? (5)

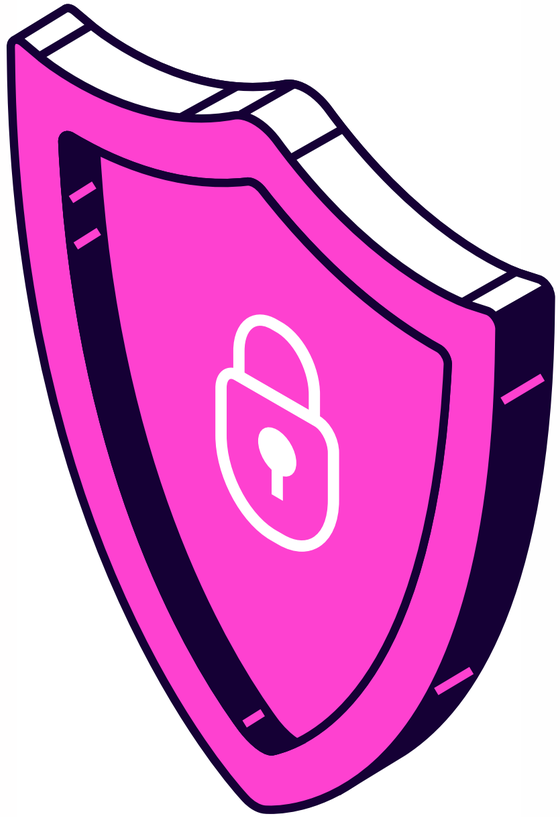


Câu hỏi

- **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 (6)



01.

Phân loại

Chia tập huấn luyện, tập validation và tập kiểm tra

02.

Data

Chỉ sử dụng dữ liệu tại cột Adj Close Price để dự đoán

03.

Chuẩn hóa

Chuẩn hóa giá trị của cột này bằng MinMaxScaler – giúp đẩy nhanh quá trình học và hội tụ của model

04.

Input-Output

Chia tập huấn luyện thành tập input và tập output (sẽ nói kỹ hơn ở phần Mô hình hóa)

05.

Preprocess_pipeline

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? (7)

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,...)

NEURAL NETWORK

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.

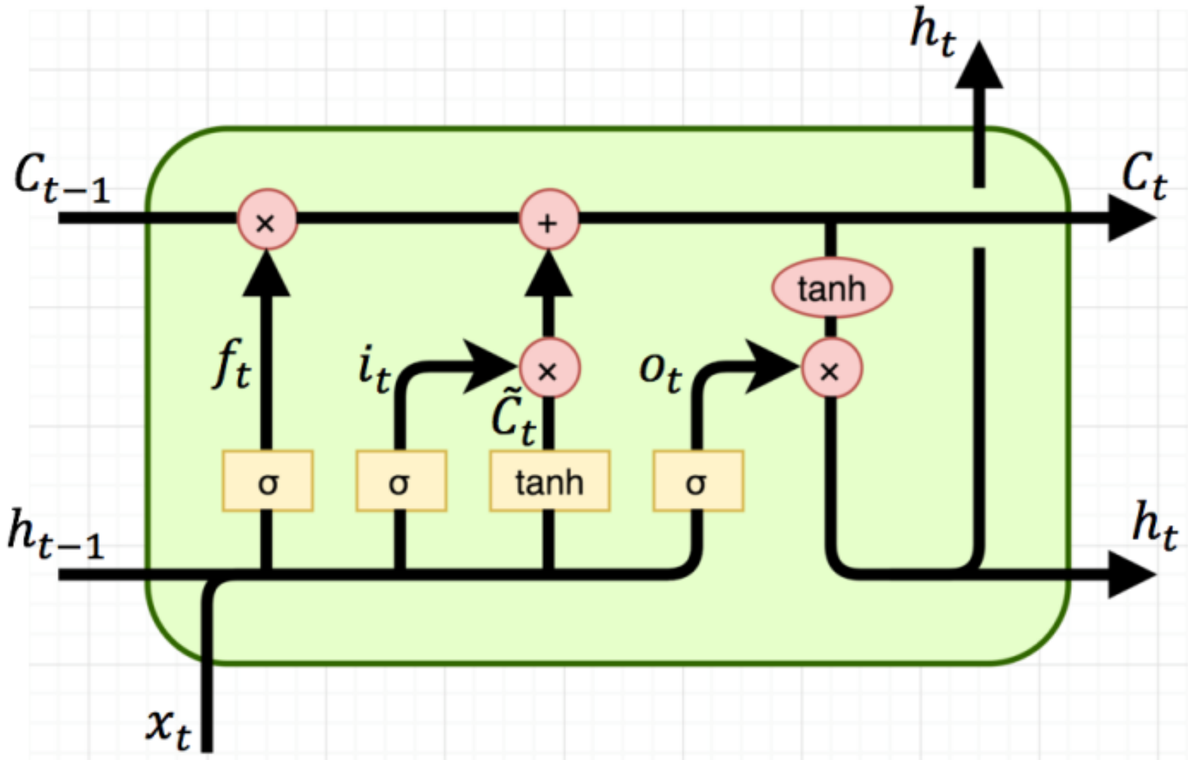
NGỮ NGHĨA

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 (8)

- Output: ct,ht, ta gọi c là cell state, h là hidden state.
- Input: ct−1,ht−1,xt. Trong đó xt là input ở state thứ t của model. ct−1,ht−1 là output của layer trước. h đóng vai trò khá giống như s ở RNN, trong khi c là điểm mới của LSTM.
- Forget gate: ft=σ(Uf∗xt+Wf∗ht−1+bf)
- Input gate: it=σ(Ui∗xt+Wi∗ht−1+bi)
- Output gate: ot=σ(Uo∗xt+Wo∗ht−1+bo)
- Ma trận trọng số: Wf, Wi, Wo, Wc
  Wf, Wi, Wo, Wc
  bf, bi, bo, bc



Mô hình LSTM

Dữ liệu đầu vào của LSTM (9)

Input: 60 ngày trước đó

Dự đoán 1 ngày tiếp theo

Output: 1 ngày cần dự đoán

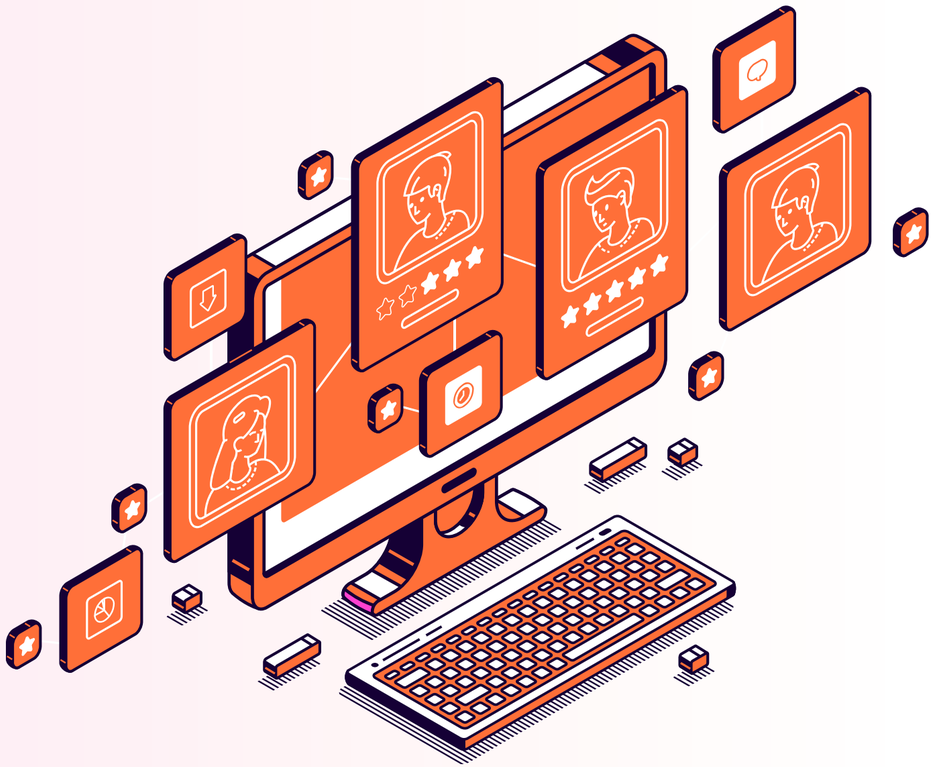
Input: 1500 ngày trước đó

Dự đoán cho 30 ngày tiếp theo

Output: 30 ngày sau đó

- 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 (10)



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

Khi đó

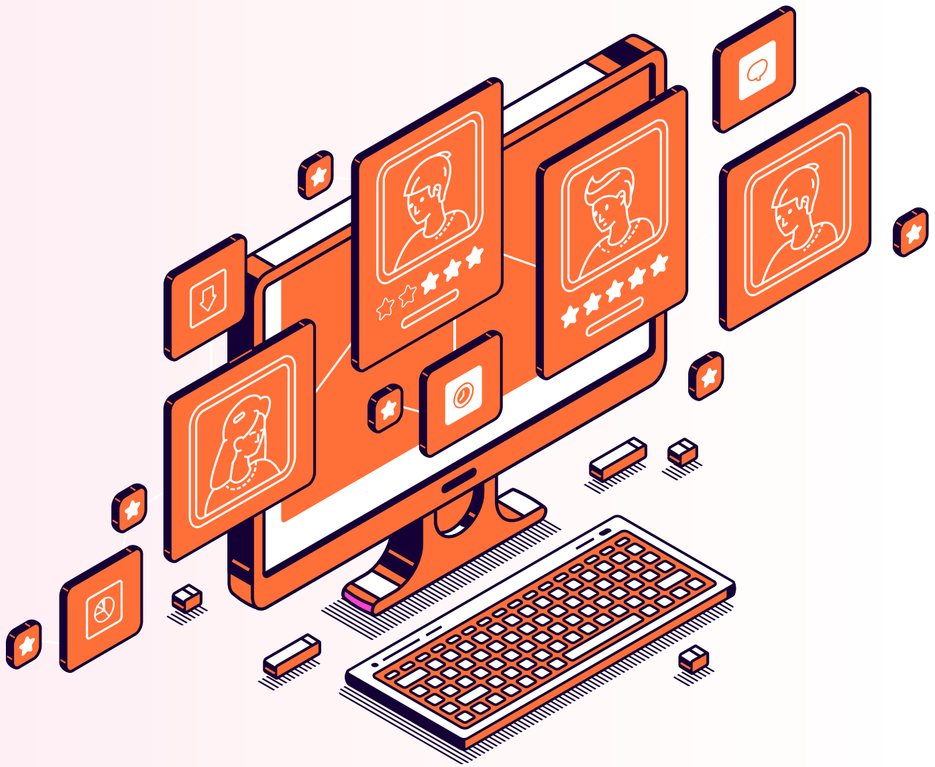
- Các ma trận W phải có shape (x, 60) để có thể nhân được với input
- h và c có shape là (30, 1) để có thể "nhân" với nhau (element wise multiplication)

Lúc này, f, i, c' sẽ có shape (30, 1)

Xét cổng forget:

- Wf phải có shape (30, 60) vì x có shape (60, 1)
- Uf phải có shape (30, 30) vì h có shape (30, 1)
- bf phải có shape (30, 1)

Tối ưu trọng số của model (11)



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

- **Tương tự với những cổng còn lại**
- **Ma trận trọng số cần phải tối ưu:**
  Wf, Wi, Wo, Wc
  Wf, Wi, Wo, Wc
  bf, bi, bo, bc
- **Thuật toán tối ưu?** Khó

Xây dựng model (12)

```python
# Tạo model
def createModel(train_X, time_steps=60, out_length=1):
    model = Sequential()
    model.add(LSTM(60, activation='tanh', return_sequences=False, input_shape=(train_X.shape[1], 1)))
    model.add(Dense(out_length))
    model.compile(optimizer='adam', loss='mse')

    return model
```

Xây dựng model (13)

```
# tạo dữ liệu train để fit (bao gồm train_df và valid_df)
train_inputs = df[:len(df) - len(test_df) - 60]
train_X, train_y = preprocess_pipeline.fit_transform(train_inputs)

# build model
model = createModel(train_X)
model.fit(train_X, train_y, batch_size=1, epochs=best_epoch)
```

Xây dựng model (14)

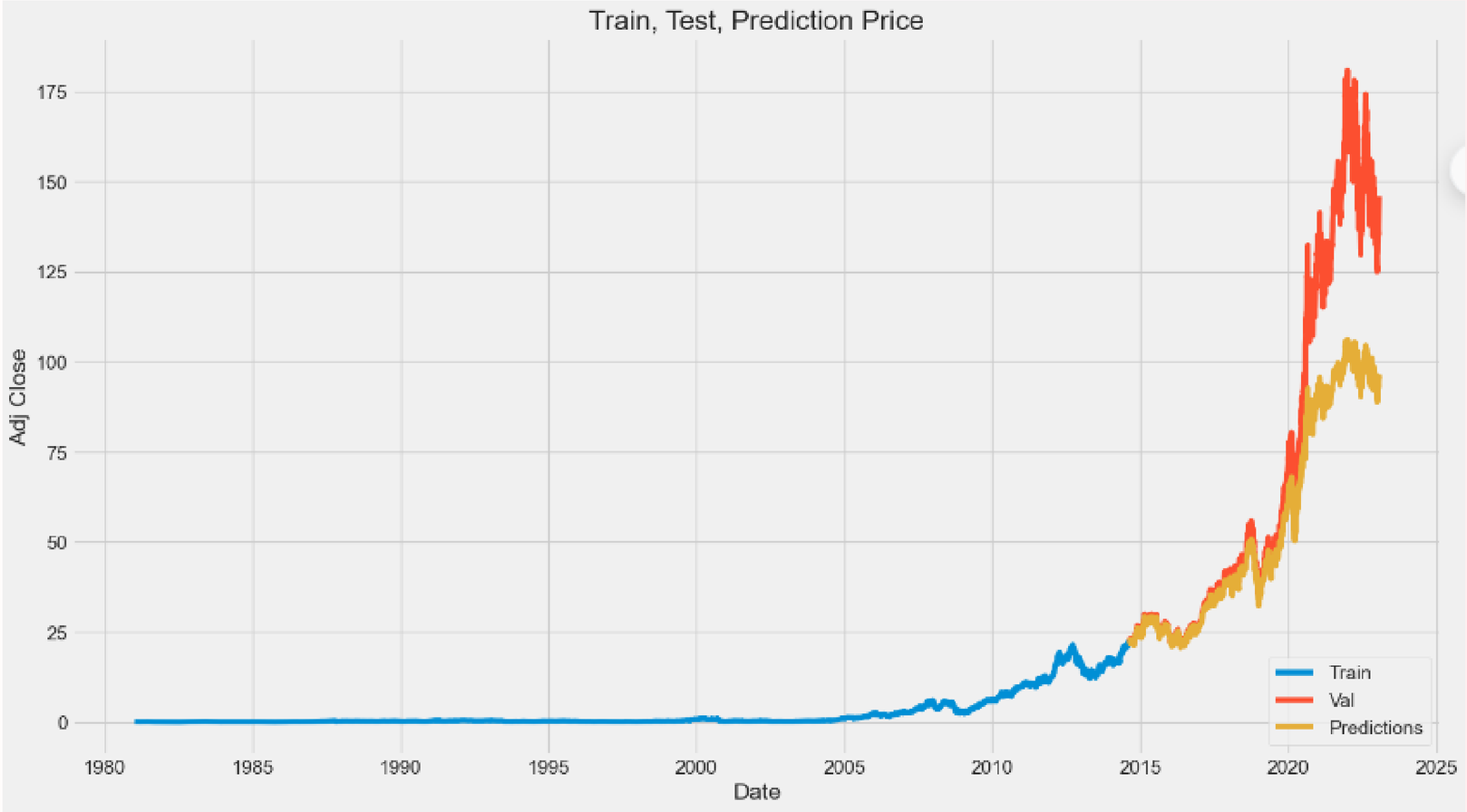
```
# tạo dữ liệu test
test_inputs = df[len(df) - len(test_df) - 60:]
test_X, test_y = preprocess_pipeline.transform(test_inputs)

# dự đoán tập inputs và chuẩn hóa ngược lại về dạng giá ban đầu
pred_test = model.predict(test_X)
pred_test = preprocess_pipeline.inverse_transform(pred_test)
```

Kết quả (15)

- **Train**
- **Val**
- **Predictions**

Dự đoán 1 ngày với 2122 giá trị test

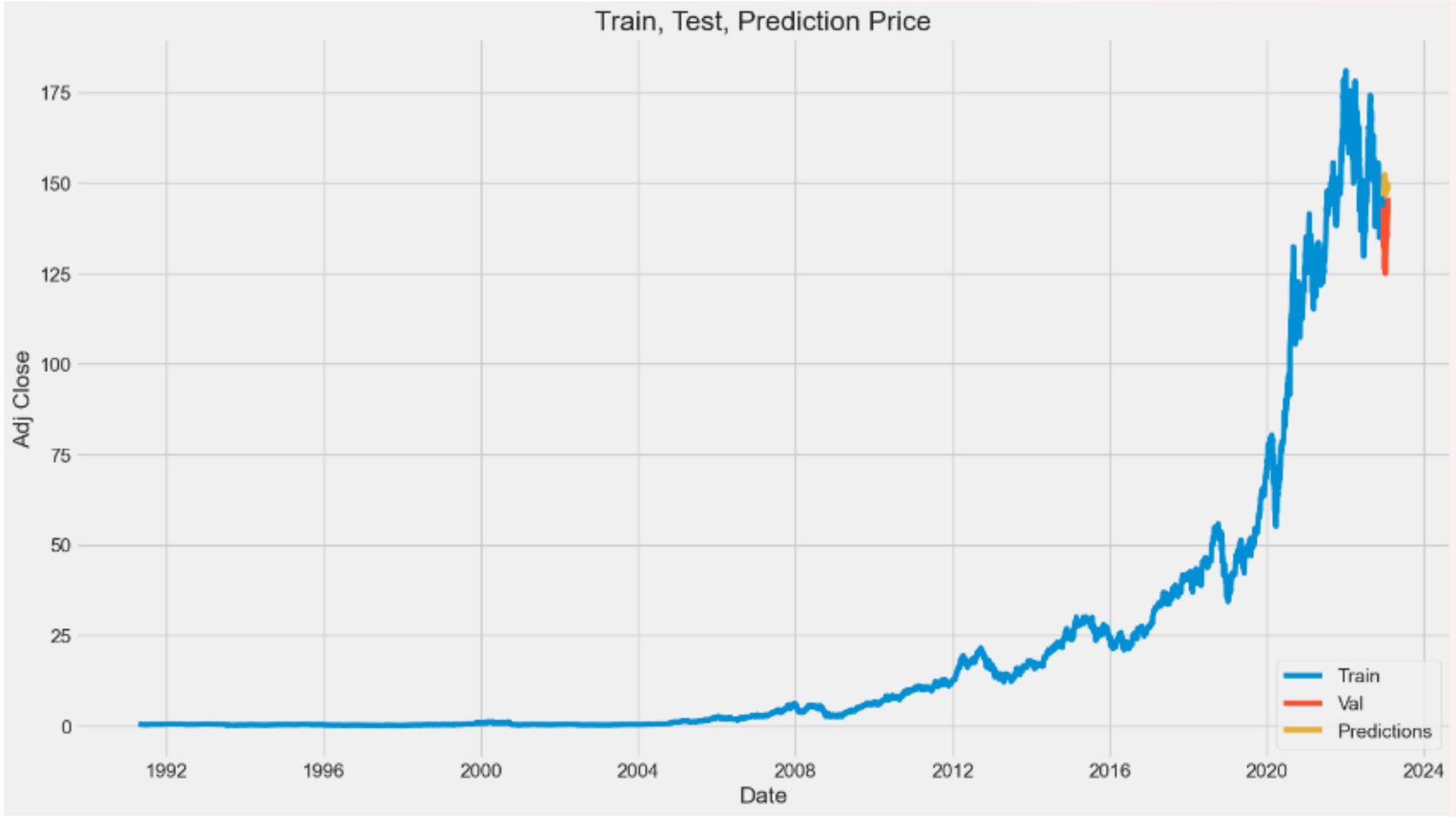
MSE = 26.652150430735873



Kết quả (16)

- **Train**
- **Val**
- **Predictions**

Dự đoán 30 ngày với 30 giá trị test
MSE = 16.084394047657305



Khó khăn trong quá trình làm việc (17)

- Hậu xử lý dữ liệu
- Quá trình tìm hiểu cơ sở toán của model
- Quá trình tìm hiểu cơ sở tối ưu độ lỗi của model

Tham khảo (18)

- dominhhai.github.io
- CS224d-Lecture8
- balolongnguyenmac.github.io
- Data Apple Stock Price Kaggle

Get protected today!

Contact info

84+ 373 104 304

21110108@st.vju.ac.vn