ĐỒ ÁN CUỐI KÌ

KHOA HỌC DỮ LIỆU

THÀNH VIÊN:

1712834 – TRẦN MINH TRÍ

1712852 – NGUYỄN NHẬT TRƯỜNG

ĐỀ TÀI

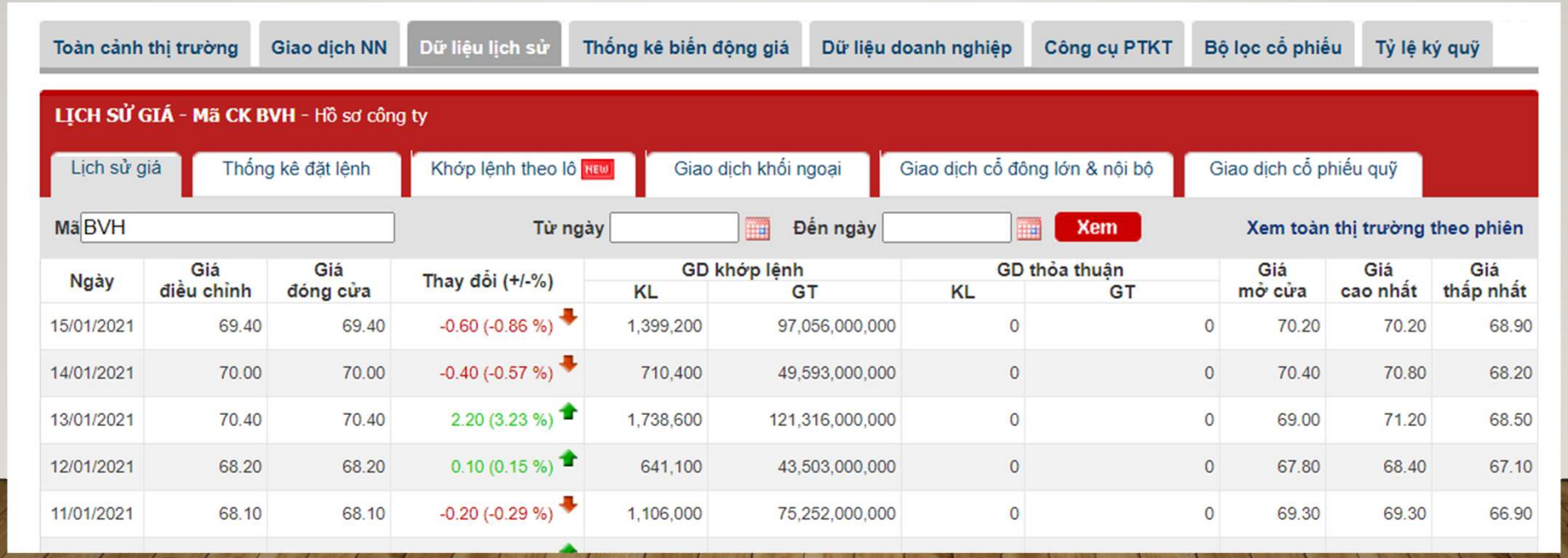
Dự đoán giá cổ phiếu

- Câu hỏi: Liệu có thể dự đoán giá cổ phiếu trong ngày tiếp theo dựa vào giá cổ phiếu đó trong quá khứ?
- Mục tiêu: Dự đoán cổ phiếu bằng mô hình máy học.
- Lợi ích: Kết quả dự đoán giúp quyết định mua hay bán cổ phiếu.

THU THẬP DỮ LIỆU

Nguồn dữ liệu

- Dữ liệu thu thập từ website cafef.vn, cổ phiếu được chọn trong đồ án này có mã chứng khoán: BVH



Toàn cảnh thị trường | Giao dịch NN | Dữ liệu lịch sử | Thống kê biến động giá | Dữ liệu doanh nghiệp | Công cụ PTKT | Bộ lọc cổ phiếu | Tỷ lệ ký quỹ

LỊCH SỬ GIÁ - Mã CK BVH - Hồ sơ công ty

Lịch sử giá | Thống kê đặt lệnh | Khớp lệnh theo lô NEW | Giao dịch khối ngoại | Giao dịch cổ đông lớn & nội bộ | Giao dịch cổ phiếu quỹ

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

Ngày	Giá điều chỉnh	Giá đóng cửa	Thay đổi (+/-%)	GD khớp lệnh KL	GD khớp lệnh GT	GD thỏa thuận KL	GD thỏa thuận GT	Giá mở cửa	Giá cao nhất	Giá thấp nhất
15/01/2021	69.40	69.40	-0.60 (-0.86 %)	1,399,200	97,056,000,000	0	0	70.20	70.20	68.90
14/01/2021	70.00	70.00	-0.40 (-0.57 %)	710,400	49,593,000,000	0	0	70.40	70.80	68.20
13/01/2021	70.40	70.40	2.20 (3.23 %)	1,738,600	121,316,000,000	0	0	69.00	71.20	68.50
12/01/2021	68.20	68.20	0.10 (0.15 %)	641,100	43,503,000,000	0	0	67.80	68.40	67.10
11/01/2021	68.10	68.10	-0.20 (-0.29 %)	1,106,000	75,252,000,000	0	0	69.30	69.30	66.90

THU THẬP DỮ LIỆU

Nguồn dữ liệu

- Nhóm đã kiểm tra có được phép thu thập dữ liệu

```
rp = urllib.robotparser.RobotFileParser()
rp.set_url('https://s.cafef.vn/robots.txt')
rp.read()
rp.can_fetch('*','https://s.cafef.vn/Lich-su-giao-dich-BVH-1.chn')
```

```
True
```

Disclaimer:

"Dữ liệu được tổng hợp từ các nguồn đáng tin cậy, có giá trị tham khảo với các nhà đầu tư.
Tuy nhiên, chúng tôi không chịu trách nhiệm trước mọi rủi ro nào do sử dụng các dữ liệu này"

Theo Trí thức trẻ

THU THẬP DỮ LIỆU

Cách thu thập dữ liệu: Webcrawler

- Nhóm sử dụng selenium để parse HTML và thực thi script dịch chuyển qua các trang trên bảng dữ liệu

```
def get_stock_data(stock_symbol, output_file):
    url = 'https://s.cafef.vn/Lich-su-giao-dich-' + stock_symbol + '-1.chn'

    file = open(output_file, 'w', encoding='utf-8')
    file.write(f'Date,Open,High,Low,Close\n')

    driver = webdriver.Chrome(executable_path='./chromedriver.exe')
    driver.get(url)
    html = HTML(html=driver.page_source)

    while True:
        for i in row_id:
            html = HTML(html=driver.page_source)
            row = html.find('tr#ctl00_ContentPlaceHolder1_ctl03_rptData2_ctl' + i + 'itemTR', first=True)
            if row:
                date = row.find('td.Item_DateItem', first=True).text
                date = pd.to_datetime(date, format='%d/%m/%Y').strftime('%Y-%m-%d')

                prices = row.find('td.Item_Price10')
                op, hi, lo, cl = prices[5].text, prices[6].text, prices[7].text, prices[1].text
                file.write(f'{date},{op},{hi},{lo},{cl}\n')

        button = driver.find_elements(By.LINK_TEXT, '>')
        if len(button) > 0:
            button[0].click()
            time.sleep(1)
        else:
            break

    file.close()
```

KHÁM PHÁ VÀ PHÂN TÍCH

Một số đặc trưng của dữ liệu

- Dữ liệu gồm 5 cột, 2868 dòng
- Dữ liệu thu được có khoảng 25/6/2009 đến 16/12/2020
- Ý nghĩa các cột dữ liệu:
  - Datetime: Ngày
  - Open: Giá mở cửa
  - High: Giá cao nhất
  - Low: Giá thấp nhất
  - Close: Giá đóng cửa

```
BVH = pd.read_csv('csv/BVH.csv', parse_dates={'Datetime':['Date']}).iloc[::-1]
BVH.set_index(['Datetime'], inplace = True)
BVH
```

Datetime	Open	High	Low	Close
2009-06-25	46.2	46.2	45.0	46.2
2009-06-26	48.5	48.5	48.0	48.5
2009-06-29	50.5	50.5	50.5	50.5
2009-06-30	53.0	53.0	53.0	53.0
2009-07-01	50.5	51.5	50.5	50.5
...	...	...	...	...
2020-12-10	57.9	58.0	56.5	56.7
2020-12-11	56.7	57.5	56.0	57.5
2020-12-14	58.0	59.4	57.3	58.6
2020-12-15	58.6	59.4	58.0	58.2
2020-12-16	58.4	58.6	57.8	58.6

2868 rows × 4 columns

KHÁM PHÁ VÀ PHÂN TÍCH

Một số đặc trưng của dữ liệu

- Kiểu dữ liệu của các cột:

```
BVH.dtypes
```

```
Open      float64
High      float64
Low       float64
Close     float64
dtype: object
```

KHÁM PHÁ VÀ PHÂN TÍCH

Một số đặc trưng của dữ liệu

- Ở đây, nhóm tập trung mô hình hóa và dự đoán cột Close, giá đóng cửa

KHÁM PHÁ VÀ PHÂN TÍCH

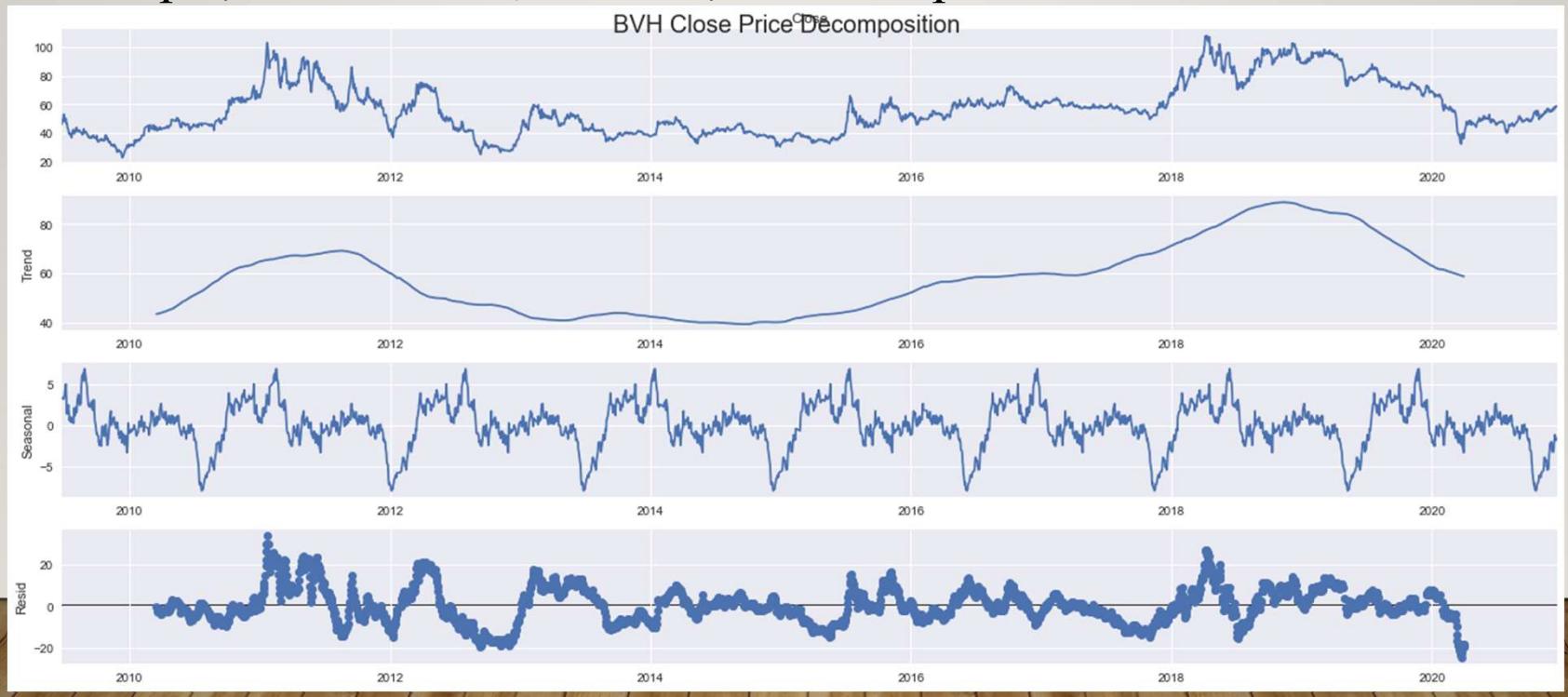
Phân tích dữ liệu

- Nhóm thực hiện phân tích thành phần chuỗi thời gian (Time-series decomposition) cho dãy giá đóng.
- Quá trình này cho phép chuỗi thời gian được thể hiện qua 3 đặc trưng chính là Trend, Seasonality và Noise.
- Tuy nhiên, do dữ liệu được sử dụng bị thiếu ở một số ngày (vấn đề này sẽ được xử lý ở dưới), khiến chuỗi thời gian được sử dụng không có tần số (frequency) cụ thể, dẫn đến quá trình decompose không thể diễn ra.

KHÁM PHÁ VÀ PHÂN TÍCH

Phân tích dữ liệu

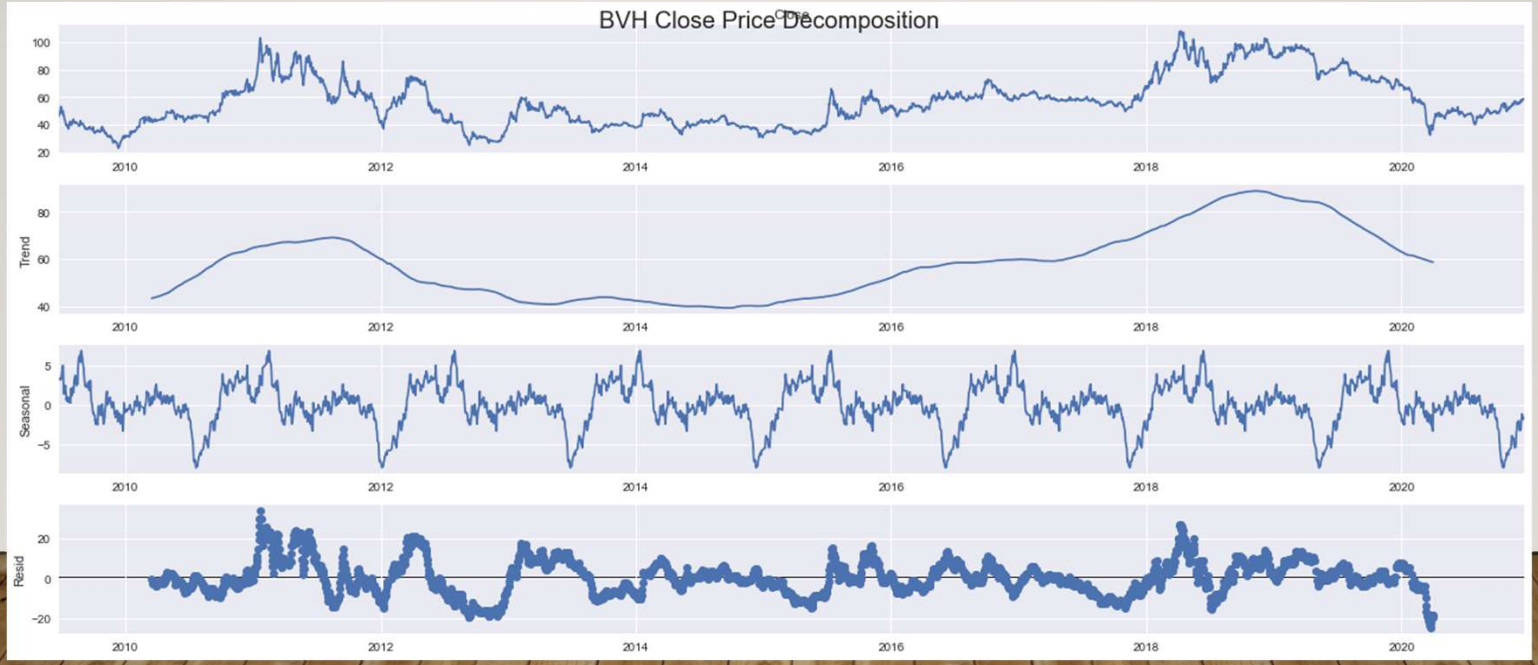
- Để khắc phục vấn đề trên, nhóm đặt tham số period=365



KHÁM PHÁ VÀ PHÂN TÍCH

Phân tích dữ liệu

- Nhận xét: chuỗi thời gian không có xu hướng - **trend** nào rõ rệt nhưng tính **seasonality** cho biết dữ liệu có diễn biến lặp lại mỗi ~18 tháng. Tuy nhiên, phần **noise** (residue) cho thấy dữ liệu vẫn mang tính ngẫu nhiên cao



TIỀN XỬ LÍ

Thêm những điểm dữ liệu bị thiếu

- Như đã nhắc đến, dữ liệu thu thập được bị đứt khoảng ở một số ngày. Để khắc phục, nhóm tự viết hàm để thêm những vị trí thiếu bằng khoảng giữa 2 đầu dữ liệu quan sát được gần nhất
- Sau khi xử lí, dữ liệu có 4193 dòng

```
BVH_filled
```

	Open	High	Low	Close
2009-06-25	46.2	46.2	45.0	46.2
2009-06-26	48.5	48.5	48.0	48.5
2009-06-27	49.2	49.2	48.8	49.2
2009-06-28	49.9	49.9	49.6	49.9
2009-06-29	50.5	50.5	50.5	50.5
...	...	...	...	...
2020-12-12	57.1	58.1	56.4	57.9
2020-12-13	57.5	58.7	56.8	58.3
2020-12-14	58.0	59.4	57.3	58.6
2020-12-15	58.6	59.4	58.0	58.2
2020-12-16	58.4	58.6	57.8	58.6

4193 rows × 4 columns

TIỀN XỬ LÍ

Tách dữ liệu cho huấn luyện

- Dữ liệu sẽ được tách thành 3 tập: train, validation, test với tỉ lệ (gần đúng) 70%-15%-15%

```
def split_data(dataset):
    train_data, test_data = train_test_split(dataset, shuffle=False, test_size=0.15)
    train_data, validation_data = train_test_split(train_data, shuffle=False, test_size=0.177)

    return train_data, validation_data, test_data
```

TIỀN XỬ LÍ

Thêm những điểm dữ liệu bị thiếu

- Biểu đồ thể hiện những điểm đã được điền:

TIỀN XỬ LÍ

Thêm những điểm dữ liệu bị thiếu

- Biểu đồ thể hiện những điểm đã được điền, phóng to trong 365 ngày cuối:

TIỀN XỬ LÍ

Thêm những điểm dữ liệu bị thiếu

- Biểu đồ thể hiện những điểm đã được điền, phóng to trong 365 ngày cuối:

TIỀN XỬ LÍ

Chuẩn hóa dữ liệu và chuyển về dạng timestep

- Nhóm sử dụng MinMaxScaler để chuẩn hóa dữ liệu về range(0, 1)
- Nhóm viết một transformer để chuyển dữ liệu ban đầu thành tập X, Y để có thể huấn luyện.
  - Tập X chứa dữ liệu n ngày trước đó
  - Tập Y chứa dữ liệu m ngày cần dự đoán

```
class Timestep_Converter(BaseEstimator, TransformerMixin):
    def __init__(self, steps=50, lag=0, y_len=1):
        self.steps = steps
        self.lag = lag
        self.y_len = 1
    def fit(self, X_df, y=None):
        return self
    def transform(self, data, y=None):
            X = []
            Y = []

            for i in range(len(data)):
                end_ixd = i + self.steps

                if end_ixd > len(data)-self.y_len:
                    break

                seq_x, seq_y = data[i:end_ixd-self.lag], data[end_ixd:end_ixd + self.y_len]
                X.append(np.array(seq_x))
                Y.append(np.array(seq_y))

            X = np.array(X).reshape(len(X), self.steps-self.lag, 1)
            Y = np.array(Y)
            Y = Y.reshape(Y.shape[0], Y.shape[1])

            return X,Y
```

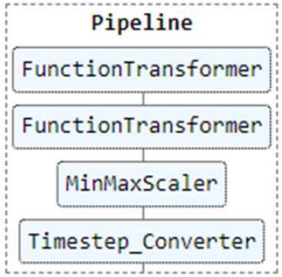
MÔ HÌNH HÓA

Pipeline tiền xử lí

- Pipeline tiền xử lí tổng hợp các bước tiền xử lí có dạng sau:

```
preprocess_pipeline = make_pipeline(FunctionTransformer(fill_time_point),
                                    FunctionTransformer(get_close_price),
                                    MinMaxScaler(feature_range=(0,1)),
                                    Timestep_Converter())

preprocess_pipeline
```



MÔ HÌNH HÓA

Xây dựng mô hình

- Nhóm sử dụng mô hình LSTM của thư viện keras như sau:
  - 2 LSTM Layer
  - 1 Dropout Layer
  - 2 Dense Layer

```
def create_LSTM(input_shape, lr, output_shape=1):
    model = Sequential()
    model.add(LSTM(units=50, return_sequences=True,input_shape=input_shape))
    model.add(LSTM(units=50, return_sequences=False))
    model.add(Dropout(0.25))
    model.add(Dense(units=50))
    model.add(Dense(units=output_shape))
    model.compile(optimizer=Adam(lr=lr), loss='mean_squared_error')
    return model
```

- Ngoài ra, nhóm có sử dụng early stopping trong keras.callbacks để dừng sớm khi độ lỗi trên tập validation không giảm với patience = 15 epochs.
- Huấn luyện tối đa 100 epochs

MÔ HÌNH HÓA

Tìm mô hình tốt nhất

- Nhóm chạy thử nghiệm 3 siêu tham số steps, learning rate và batch size.
- Nhóm thực hiện thử nghiệm với 2 mô hình:

Mô hình dự đoán 1 ngày tiếp theo

	16	32	64	128
0.001	2.638	2.650	2.598	2.723
0.010	2.786	2.581	2.606	2.636
0.100	481.168	81.047	23.452	286.521
1.000	271.395	284.319	510.052	417.536

Steps = 30

	16	32	64	128
0.001	2.760	2.691	2.779	3.021
0.010	2.626	2.770	2.622	2.655
0.100	248.205	5.164	322.614	405.425
1.000	279.331	260.462	312.084	274.143

Steps = 40

	16	32	64	128
0.001	2.860	2.712	2.724	2.951
0.010	2.741	2.739	2.765	2.714
0.100	5.727	59.331	10.624	3.276
1.000	505.191	254.910	260.014	283.737

Steps = 50

	16	32	64	128
0.001	2.806	2.760	2.777	3.376
0.010	2.985	2.826	2.903	2.656
0.100	5.492	60.638	13.197	71.923
1.000	249.886	250.580	240.906	530.073

Steps = 60

Mô hình dự đoán 7 ngày tiếp theo

	16	32	64	128
0.001	9.972	10.307	10.029	10.190
0.010	10.231	10.339	10.271	10.740
0.100	18.771	604.939	14.134	15.452

Steps = 30

	16	32	64	128
0.001	10.436	9.970	10.183	10.580
0.010	10.449	11.209	10.198	10.502
0.100	713.639	583.609	24.357	440.249

Steps = 40

	16	32	64	128
0.001	10.992	10.383	10.366	10.470
0.010	10.914	10.510	10.479	11.512
0.100	15.247	315.538	767.123	170.647

Steps = 50

	16	32	64	128
0.001	10.853	11.112	10.565	11.017
0.010	10.955	10.557	11.027	12.234
0.100	1084.098	13.505	16.381	260.448

Steps = 60

MÔ HÌNH HÓA

Mô hình tốt nhất tìm được

- Mô hình dự đoán 1 ngày tiếp theo tốt nhất:

Model tốt nhất

```
best_val_err, best_steps, best_lr, best_batch_size
```

```
(2.580698711379751, 30, 0.01, 32)
```

- Mô hình dự đoán 7 ngày tiếp theo tốt nhất:

Model tốt nhất

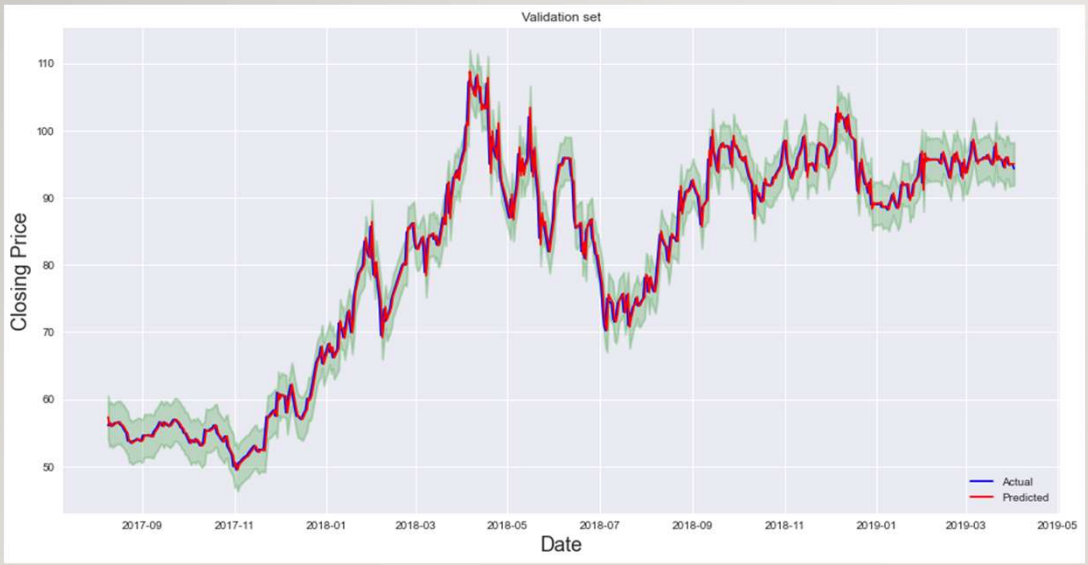
```
best_val_err_7, best_steps_7, best_lr_7, best_batch_size_7
```

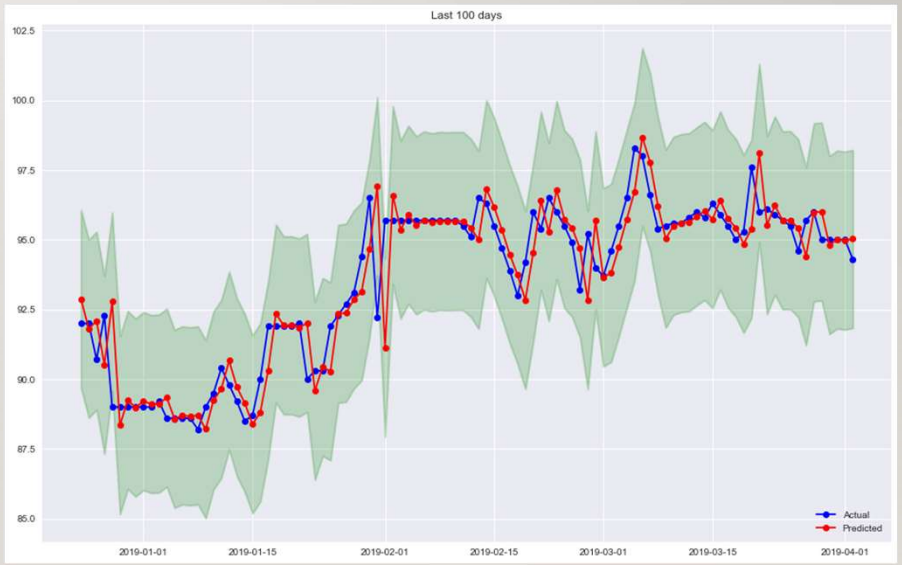
```
(9.97022470918136, 40, 0.001, 32)
```

MÔ HÌNH HÓA

Kết quả mô hình tốt nhất trên tập validation

- Mô hình dự đoán 1 ngày tiếp theo:

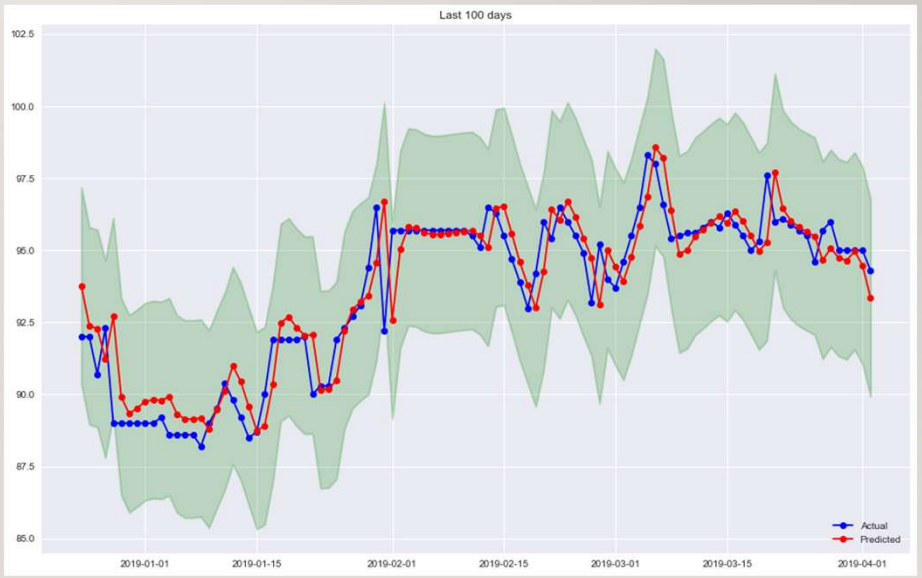




MÔ HÌNH HÓA

Kết quả mô hình tốt nhất trên tập validation

- Mô hình dự đoán 7 ngày tiếp theo:

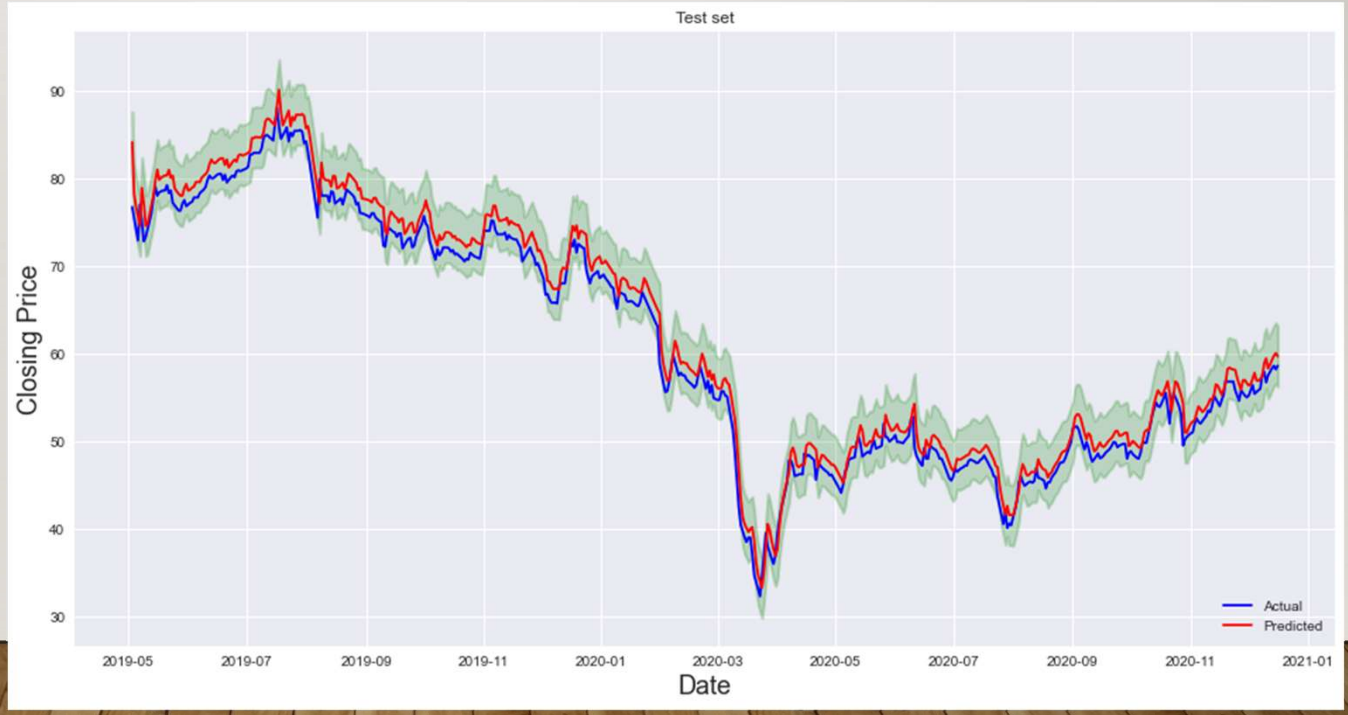


MÔ HÌNH HÓA

Độ lỗi trên tập test

- Mô hình dự đoán 1 ngày tiếp theo:

```
MSE: 3.2234174747835063
count    594.000000
mean      -1.487683
std        1.005942
min       -7.413678
25%       -1.945801
50%       -1.525513
75%       -0.994465
max        2.801329
Name: Error, dtype: float64
```

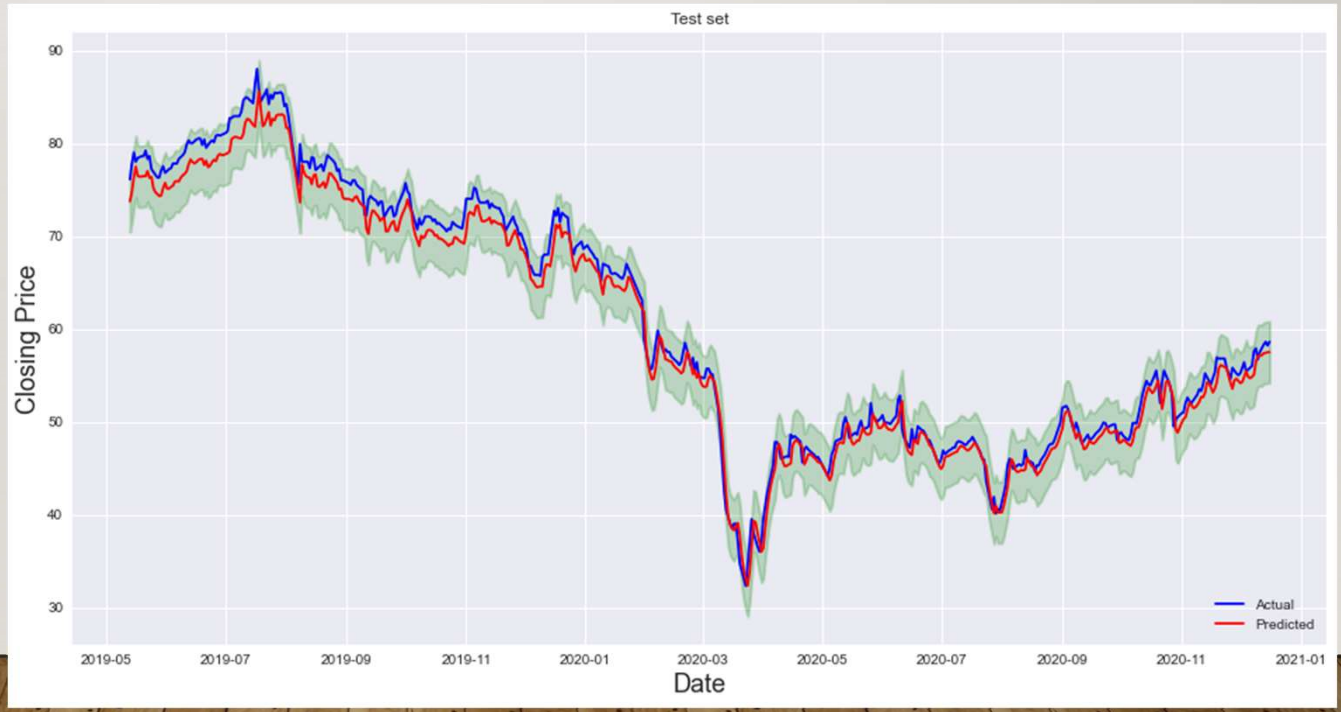


MÔ HÌNH HÓA

Độ lỗi trên tập test

- Mô hình dự đoán 7 ngày tiếp theo:

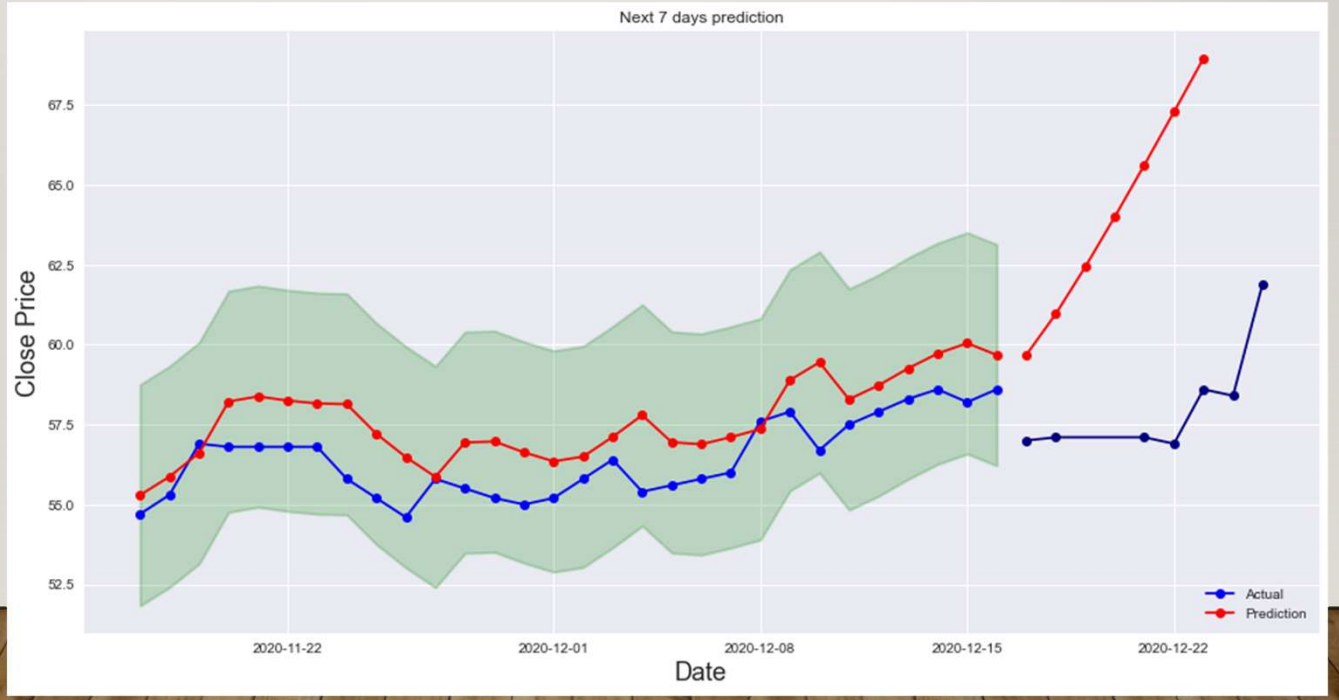
```
MSE on 7 days: 5.5960501980831845
MSE: 2.5084656422429434
count    584.000000
mean       1.130687
std        1.110010
min       -3.121815
25%        0.489900
50%        1.101181
75%        1.879736
max        6.294173
Name: Error, dtype: float64
```



DỰ ĐOÁN

Mô hình dự đoán 1 ngày tiếp theo

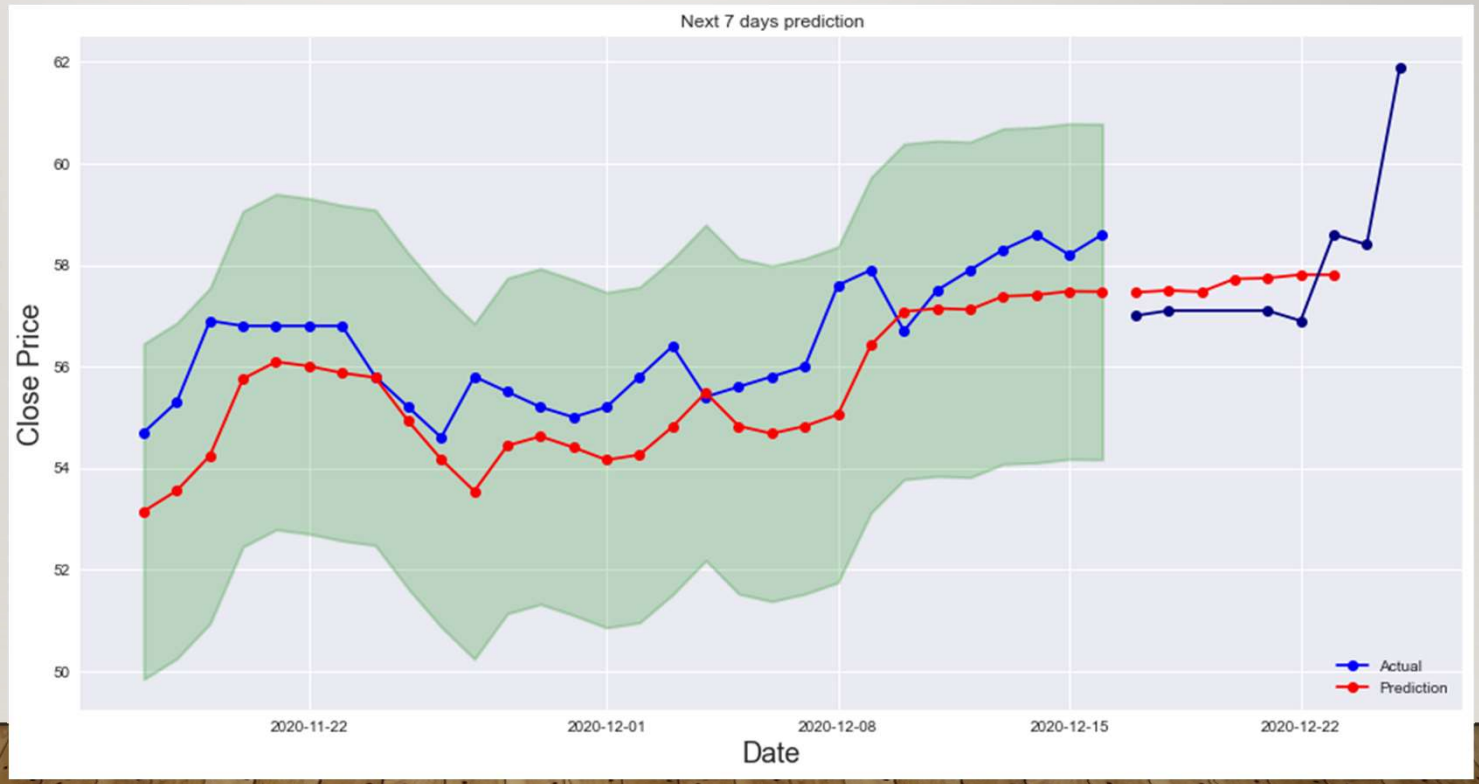
- Mô hình chỉ dự đoán được 1 ngày tiếp theo, nên nhóm thử dùng phương pháp dự đoán liên tiếp, với kết quả dự đoán của ngày trước.
- Kết quả dự đoán:



DỰ ĐOÁN

Mô hình dự đoán 7 ngày tiếp theo

- Kết quả dự đoán:



ĐÁNH GIÁ ĐỒ ÁN

Kinh nghiệm tích lũy

- Qua đồ án này, nhóm học được nhiều về mô hình hóa dự đoán cho kiểu dữ liệu chuỗi thời gian, kiểu dữ liệu chưa được demo trực tiếp trong khóa học
- Học và làm quen với sử dụng pipeline cho việc huấn luyện được gọn gàng hơn

ĐÁNH GIÁ ĐỒ ÁN

Khó khăn mắc phải

- Lần đầu tiếp xúc với việc dự đoán trên kiểu timeseries nên nhóm phải tốn nhiều thời gian nghiên cứu, nhưng thấy được đó vẫn chưa đủ.
- Tiền xử lý, mô hình hóa còn nhiều chỗ có thể cải tiến để thu được mô hình tốt hơn

ĐÁNH GIÁ ĐỒ ÁN

Hướng phát triển nếu có thêm thời gian

- Tìm hiểu thêm các phương pháp tiền xử lí để cải tiến mô hình
- Tìm hiểu thêm về cách dự đoán tương lai gần khi đã có mô hình máy học
- Tìm hiểu thêm và chạy thử các siêu tham số trên cái layer


Thank you for attending