Stock Market Analysis and Different Methods of Predicting Stock Price

Math 5010 - Introduction to the Mathematics of Finance
Columbia University
May.2.2022

Team Member:
WEILIN ZHOU wz2563
ZHONGCHENG TU zt2286
MINGZHU YU my2691
CHUN-CHIEH TSENG ct3057
WENTAN CHEN wc2768

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Introduction

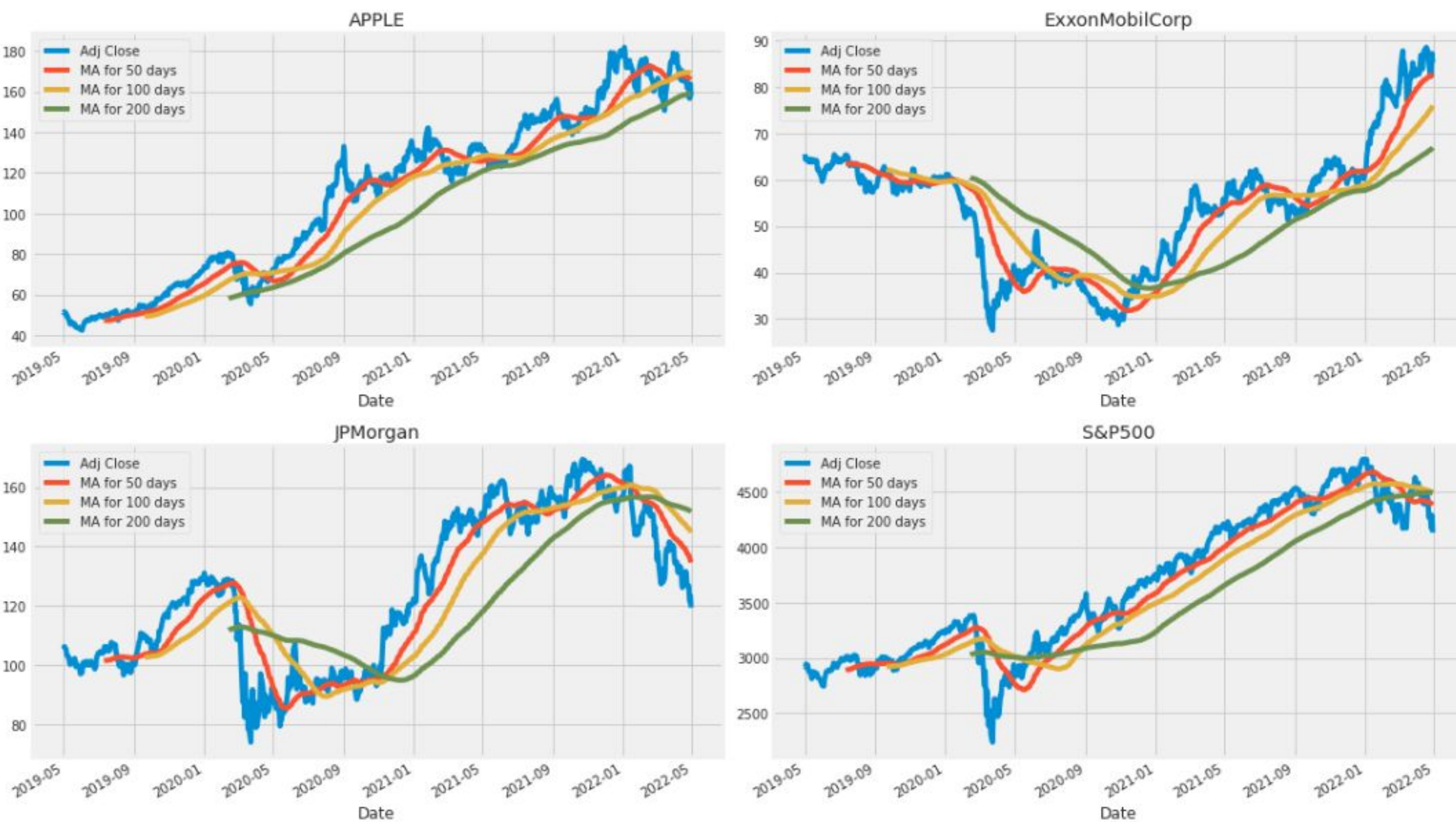
1. Data Visualization and Analysis
2. Linear Regression
3. Times Series ARIMA
4. LSTM
5. GRU
6. Conclusion

Data Description

- **Closing Price**

Data Description

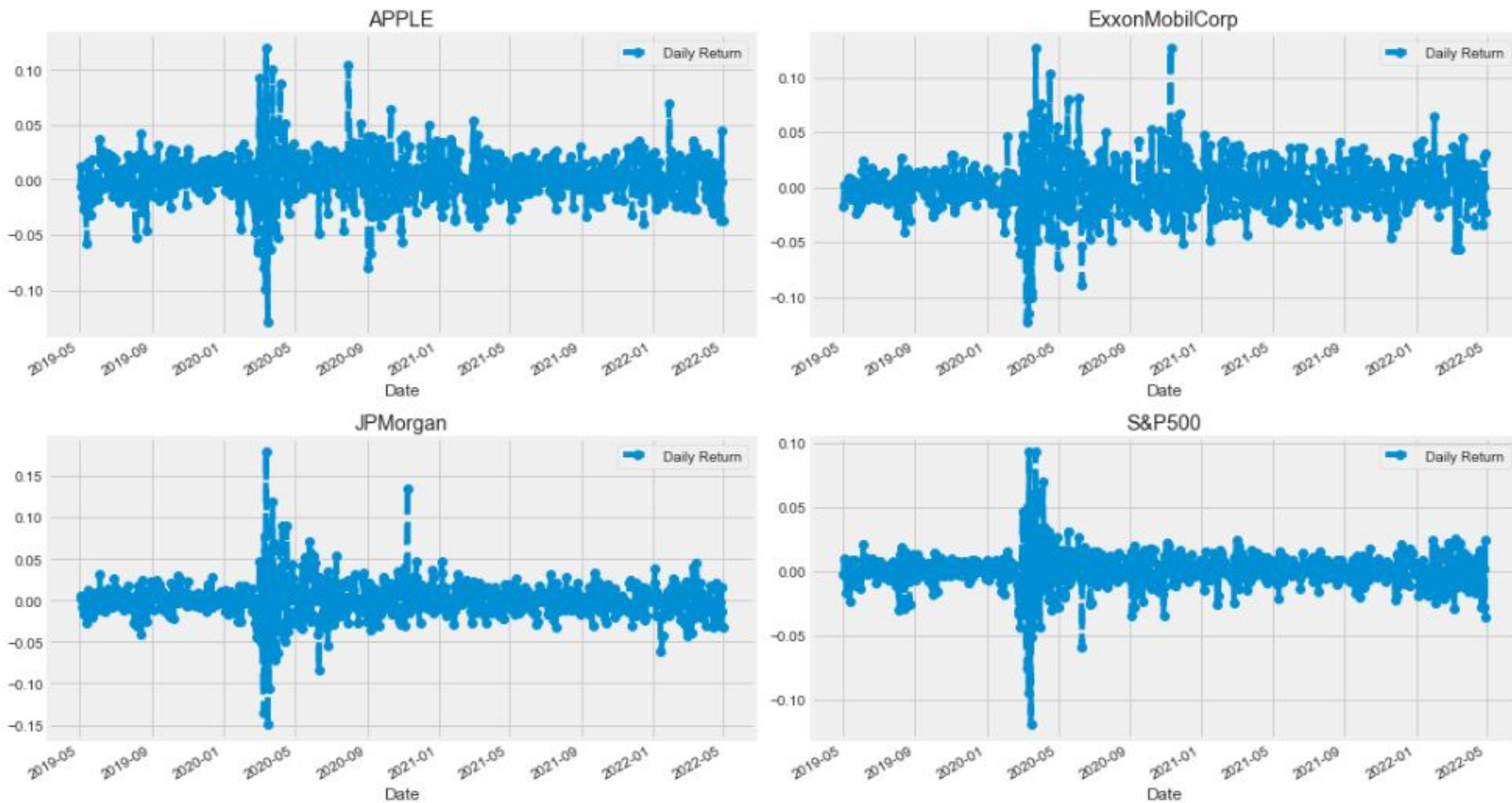
- **Moving Average**



Columbia University
In the City of New York

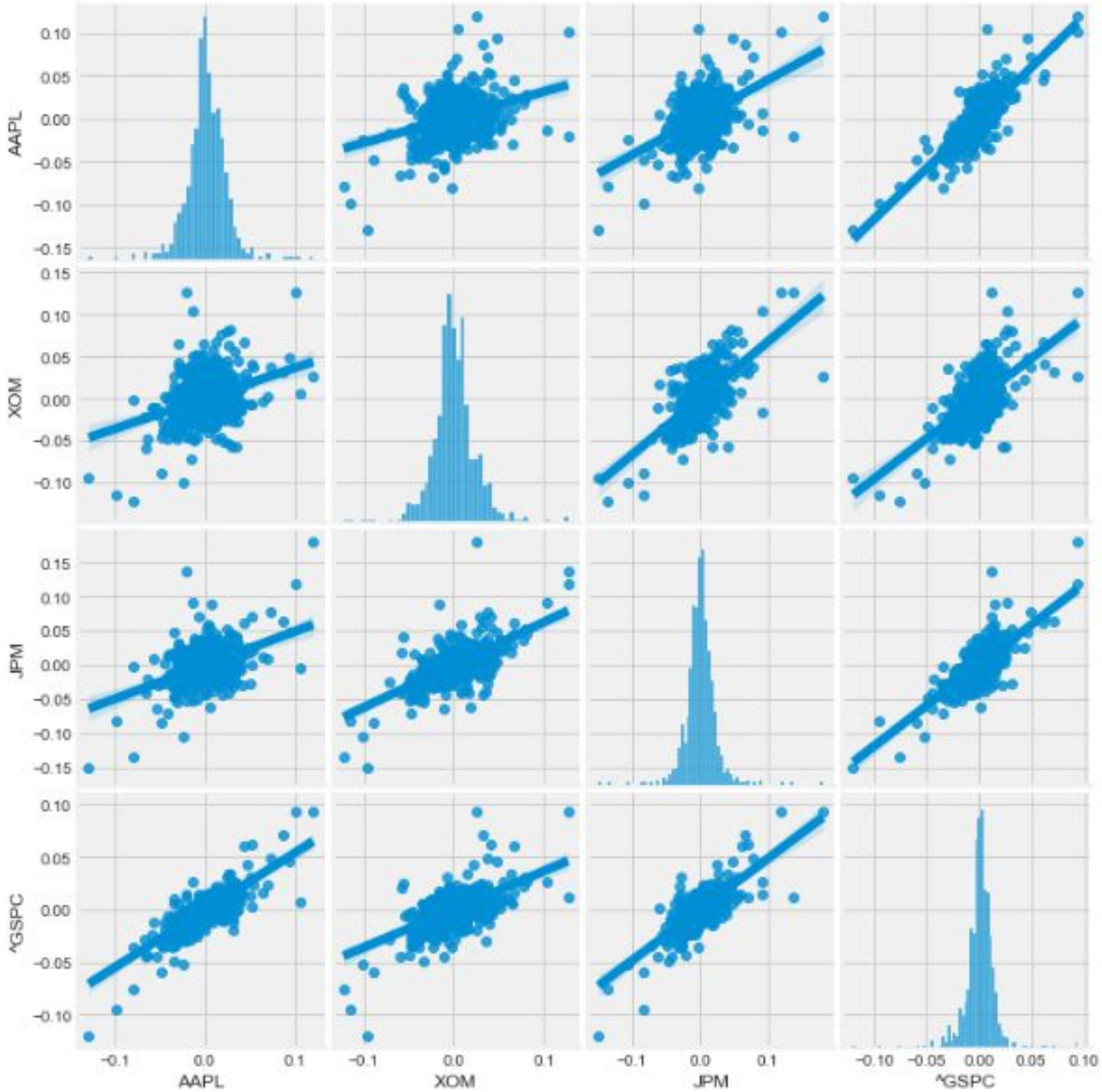
Data Description

- **Daily Return**



COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Data Description



COLUMBIA UNIVE
IN THE CITY OF NEW YORK

Linear Regression Model

It helps identify the relationships between a dependent variable(Y)and one or more independent variables(X). Simple linear regression is defined by using a feature to predict an outcome.

How to perform a simple linear regression

Simple linear regression formula

The formula for a simple linear regression is:

$$y = \beta_0 + \beta_1 X + \varepsilon$$

- **y** is the predicted value of the dependent variable (**y**) for any given value of the independent variable (**x**).
- **B_0** is the **intercept**, the predicted value of **y** when the **x** is 0.
- **B_1** is the regression coefficient – how much we expect **y** to change as **x** increases.
- **x** is the independent variable (the variable we expect is influencing **y**).
- **e** is the **error** of the estimate, or how much variation there is in our estimate of the regression coefficient.

Predict Stock Prices Using Linear Regression

Step 1: Get historical pricing data from Yahoo Finance.

Using Apple stock price and S&P 500 as Examples.

Step 2: Prepare the data

Before we start developing our regression model we are going to trim our data. Only use the “Adj Close” price.

Step 3: Adding Technical Indicators.

Technical indicators are calculated values describing movements in historical pricing data for securities like stocks, bonds, and ETFs.

Commonly used technical indicators include **moving averages** (SMA, EMA, MACD), the **Relative Strength Index** (RSI), **Bollinger Bands** (BBANDS), and several others.

We decide to add an **exponential moving average** (EMA) to our data:

$$EMA = (2 / n+1) \times (Close - Previous\ EMA) + Previous\ EMA$$

Predict Stock Prices Using Linear Regression

we add a new column in our data titled "EMA_20." This is our newly-calculated value representing the exponential moving average calculated over a 20-day period.

The first 19 entries in our data will have a `NaN` value since there weren't proceeding values from which the EMA could be calculated.

we're going to just drop all the rows where we have NaN values and use a slightly smaller dataset by taking the following approach.

APPLE

Date	Adj Close	EMA_20
2018-01-31	39.982349	41.474010
2018-02-01	40.065929	41.339907
2018-02-02	38.327473	41.053009
2018-02-05	37.369884	40.702235
2018-02-06	38.931622	40.533605
...	...	...
2022-04-25	162.880005	167.555304
2022-04-26	156.800003	166.530989
2022-04-27	156.570007	165.582324
2022-04-28	163.639999	165.397341
2022-04-29	157.649994	164.659498

1070 rows × 2 columns

S&P500

Date	Adj Close	EMA_20
2018-01-31	2823.810059	2791.504128
2018-02-01	2821.979980	2794.406590
2018-02-02	2762.129883	2791.332618
2018-02-05	2648.939941	2777.771411
2018-02-06	2695.139893	2769.901742
...	...	...
2022-04-25	4296.120117	4419.894689
2022-04-26	4175.200195	4396.590452
2022-04-27	4183.959961	4376.339929
2022-04-28	4287.500000	4367.878983
2022-04-29	4131.930176	4345.407668

1070 rows × 2 columns

Predict Stock Prices Using Linear Regression

Step 4:Test-Train Data.

Using eighty percent of data for training and the remaining twenty percent for testing is common.

Step 5: Training the Model

We have our data and now we want to see how well it can be fit to a linear model.

Our linear model has now been trained on 856 training samples, and we've generated predicted values based on these training samples.

Let's see how well our model fits our data by examining our model coefficients and **mean absolute error (MAE)** and **coefficient of determination (r2)**.

Formula

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

MAE = mean absolute error

y_i = prediction

x_i = true value

n = total number of data points

Formula

$$R^2 = 1 - \frac{RSS}{TSS}$$

R^2 = coefficient of determination

RSS = sum of squares of residuals

TSS = total sum of squares

Predict stock prices using linear regression

The lower MAE value is better, and the closer our coefficient of the correlation value is to 1.0 the better.

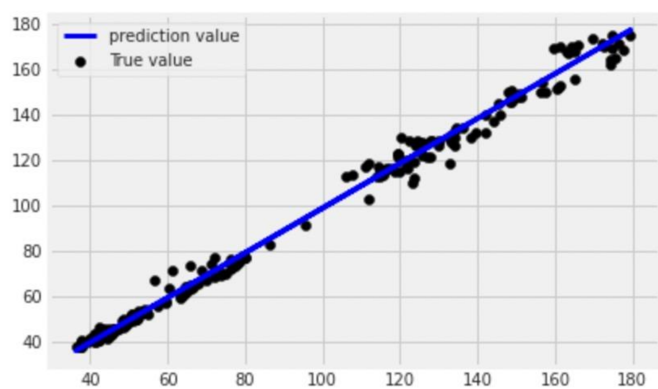
APPLE:

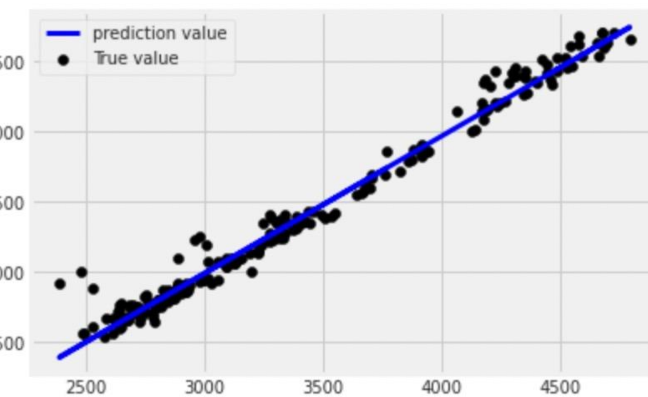
```
Model Coefficients: [[0.98680509]]
Mean Absolute Error: 2.731616416001934
Coefficient of Determination: 0.992364044229694
```

S&P 500:

```
Model Coefficients: [[0.98038546]]
Mean Absolute Error: 63.164174827406214
Coefficient of Determination: 0.9794486290631288
```

Step 6: Plotting



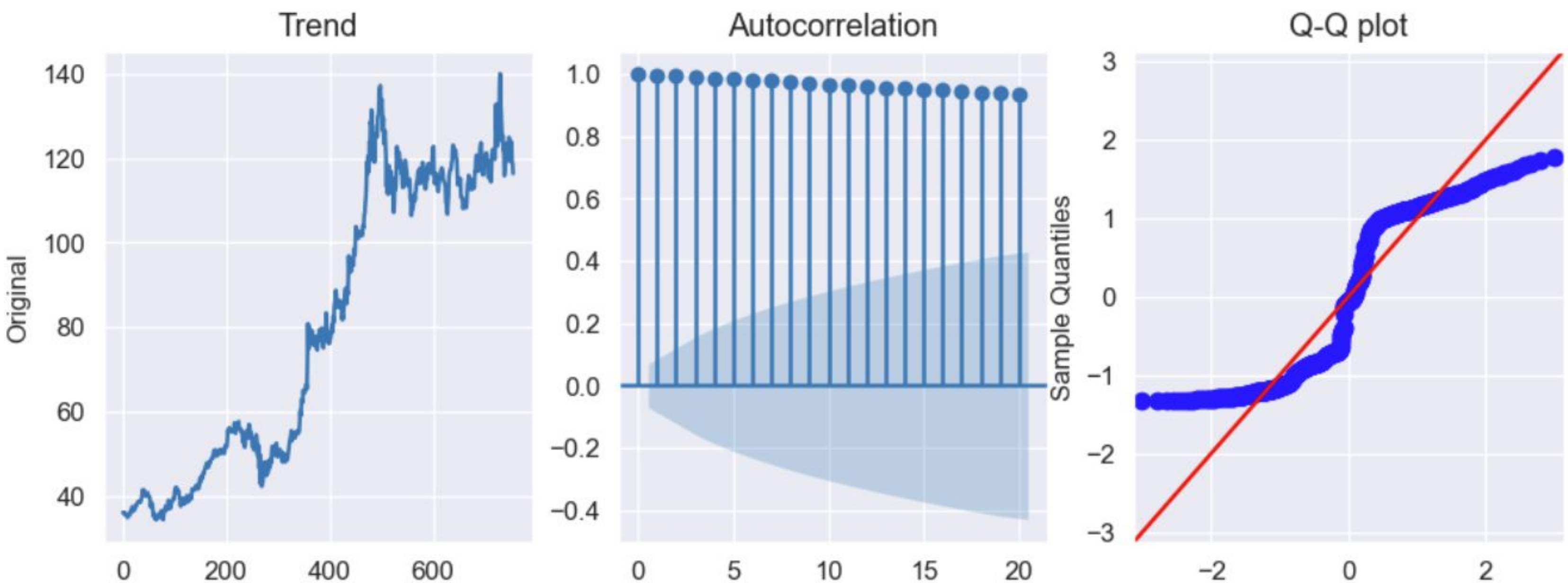


From the above result we can conclude: our model fits our data well, though the MAE is slightly high.Looks like a pretty good fit!(Using EMA_20 as X, and Adj Close Price as Y)

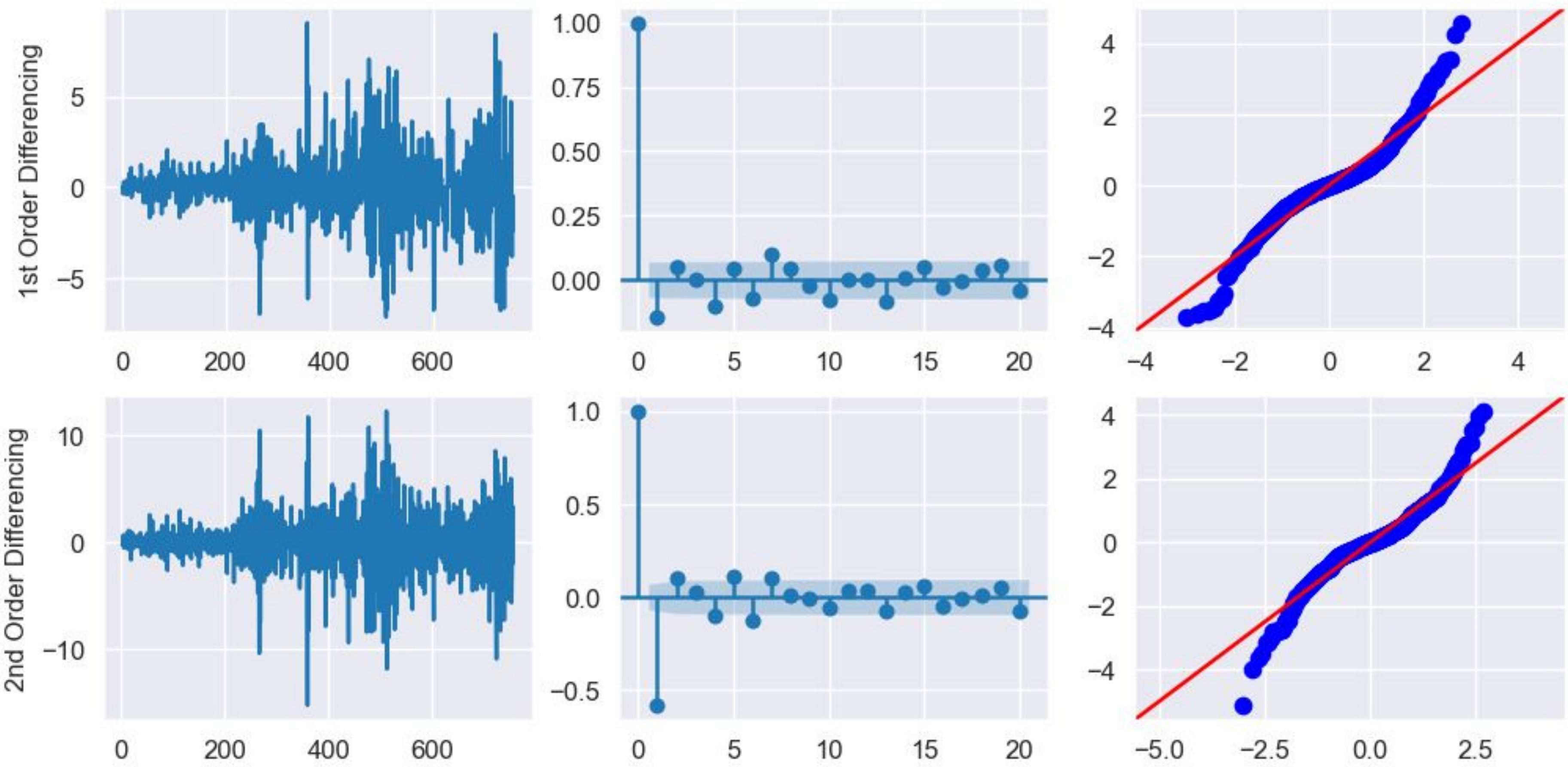
Columbia University
In the City of New York

Predict Future Trend by Times Series

In order to create a model that can better fit the stock trend, we want to transform the data in some way that can let it show a stationary pattern.

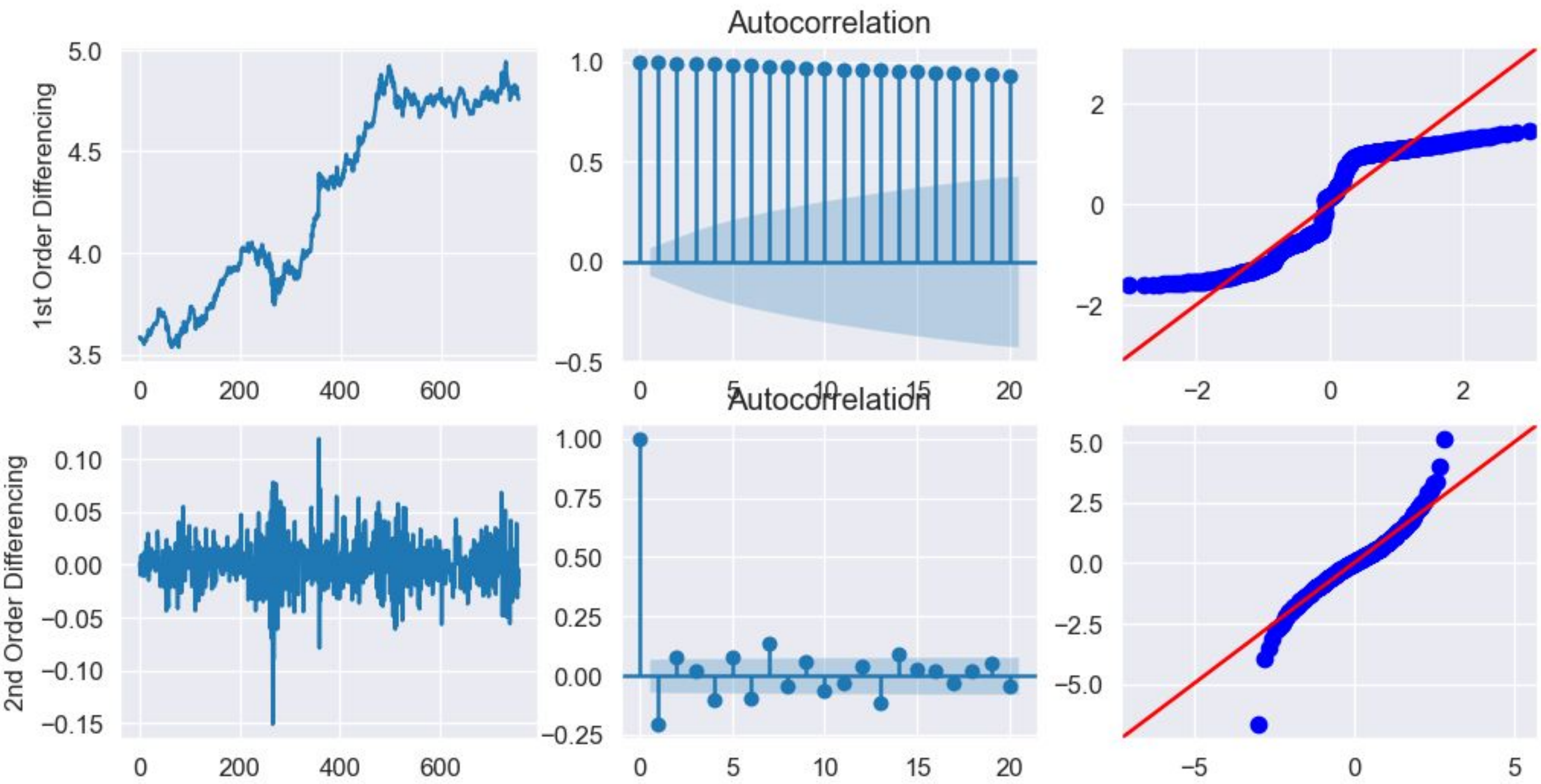


1st and 2nd order differencing data

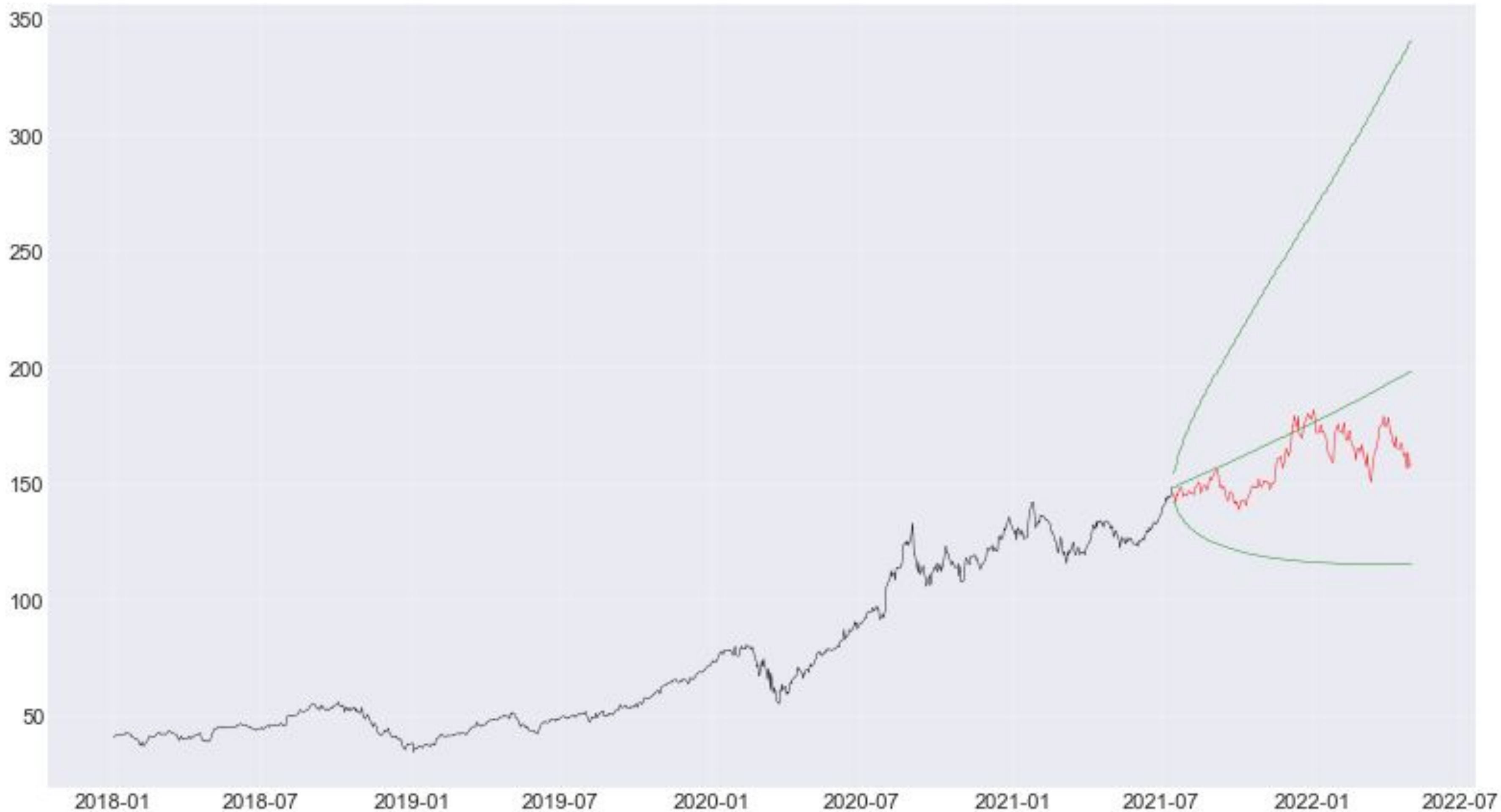


COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Logged data and its transform



Prediction of AAPL this year, train by past 4 years' data

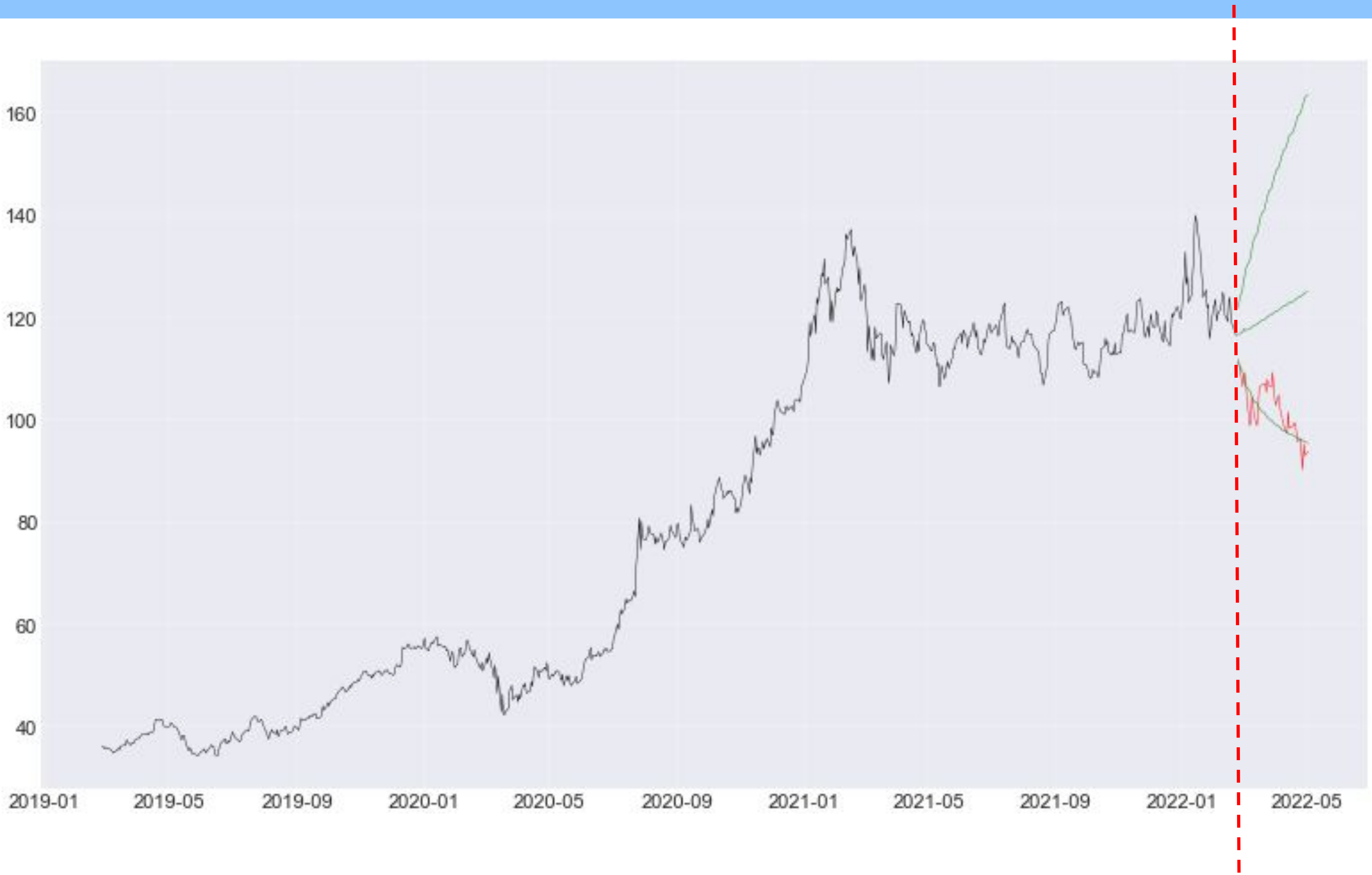


	AAPL	S&P 500
MSE	289.68	34934.59

Disadvantage: External Influence Make Prediction Imprecise

2020-03-05

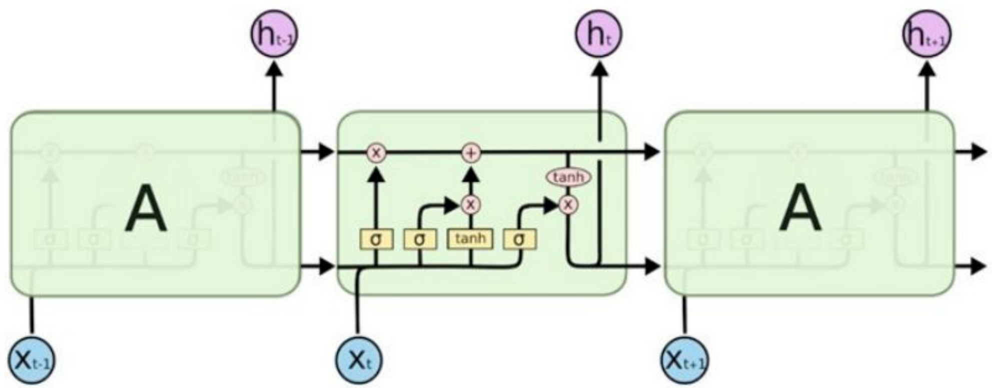
Disadvantage: External Influence Make Prediction Imprecise



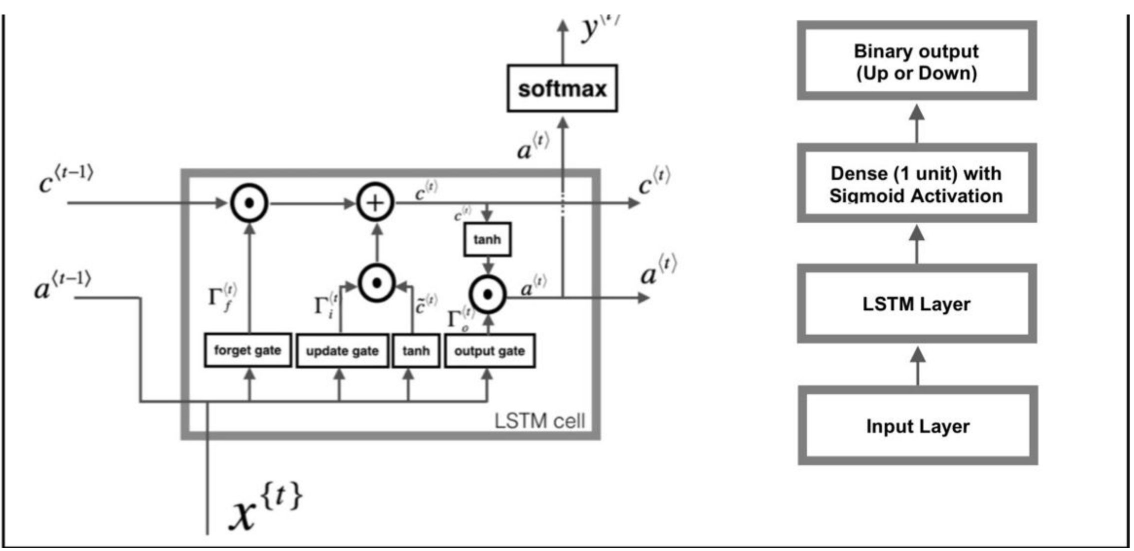
2022-02-24

LSTM (Long Short-Term Memory) (Recurrent Neural Network)

A special type of Recurrent Neural Network….



Designed to mitigate the vanishing and exploding gradient problem apart from the hidden state each LSTM cell maintains a **cell state vector** and at each time step the next LSTM can choose to read from it write to it or reset the cell using explicit gating mechanism.



Three gates of LSTM Cell:
Input gate: Is cell updated?
Forget gate: Is memory set to 0?
Output gate: Is current info visible?

They all have a sigmoid activation, so that they **constitute smooth curves** in the range 0 to 1 and model remains differentiable.

$$\bar{C}^{(t)} = \tanh\left(W^C\left[h^{(t-1)}, x^{(t)}\right] + b^C\right)$$

LSTM Procedure

Step1-2: Same as Above
Step 3:Test-Train Data (80:20)
Step 4: Build the model
Step 5: Change the Parameter(Especially Epochs)
Step 6: Prediction and Forecast

```
# load the dataset
df = DataReader('AAPL', data_source='yahoo', start='2020-01-01', end=datetime.now())
data = df.filter(['Close'])
dataset = data.values
# normalize the dataset
scaler = MinMaxScaler(feature_range=(0, 1))
dataset = scaler.fit_transform(dataset)
# split into train and test sets
train_size = int(len(dataset) * 0.8)
test_size = len(dataset) - train_size
train, test = dataset[0:train_size,:], dataset[train_size:len(dataset),:]
# reshape into X=t and Y=t+1
look_back = 20
trainX, trainY = create_dataset(train, look_back)
testX, testY = create_dataset(test, look_back)
# reshape input to be [samples, time steps, features]
trainX = numpy.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
testX = numpy.reshape(testX, (testX.shape[0], 1, testX.shape[1]))
```

LSTM

Different amount of Dataset used in predicting same stock and same epochs

```python
# create and fit the LSTM network
model = Sequential()
model.add(LSTM(4, input_shape=(1, look_back)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(trainX, trainY, epochs=25, batch_size=1, verbose=2)
# make predictions
trainPredict = model.predict(trainX)
testPredict = model.predict(testX)
```

MSE:

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

MSE = mean squared error

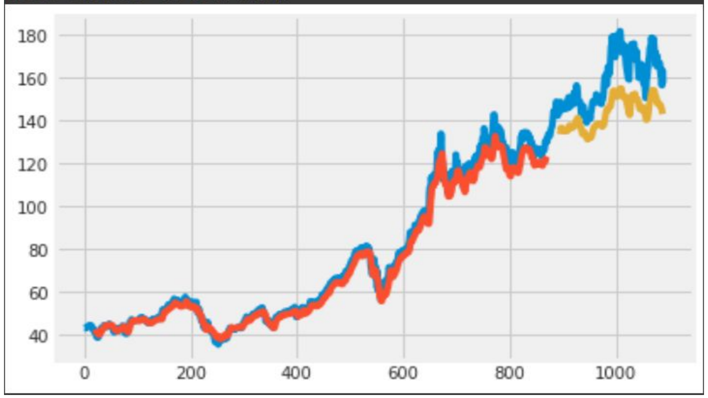
n = number of data points

Y_i = observed values

$\hat{Y}_i$ = predicted values

851 Days APPLE Closing Price

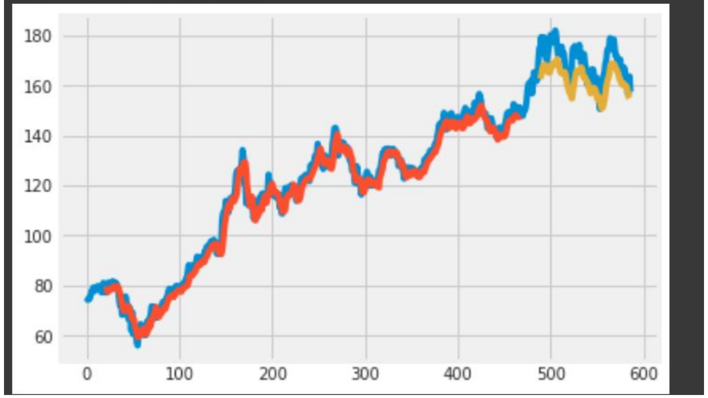
```
851/851 - 1s - loss: 2.5635e-04 - 1s/epoch - 1ms/step
Train Score: 3.73 RMSE
Test Score: 17.02 RMSE
```



df = DataReader('AAPL', data_source='yahoo', start='2018-01-01', end=datetime.now())

448 Days APPLE Closing Price

```
448/448 - 1s - loss: 6.4318e-04 - 635ms/epoch - 1ms/step
Train Score: 3.38 RMSE
Test Score: 8.01 RMSE
```

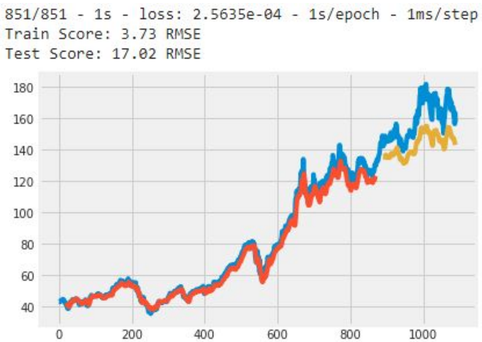


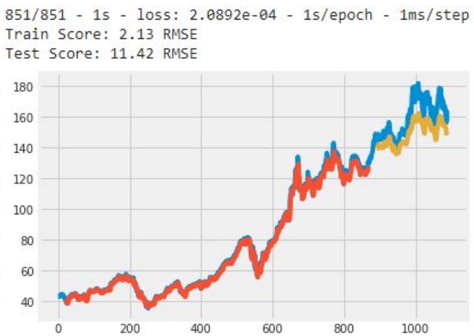
df = DataReader('AAPL', data_source='yahoo', start='2020-01-01', end=datetime.now())

Columbia University
in the City of New York

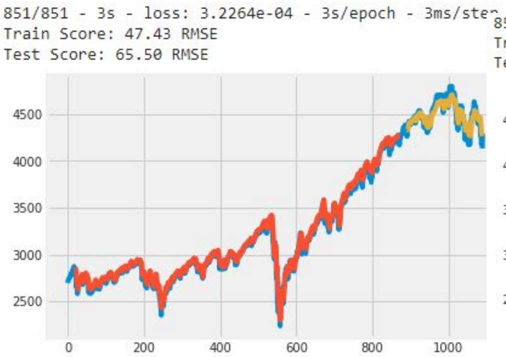
LSTM

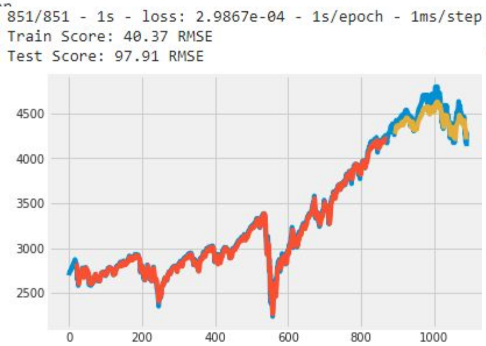
APPLE Starting 2018

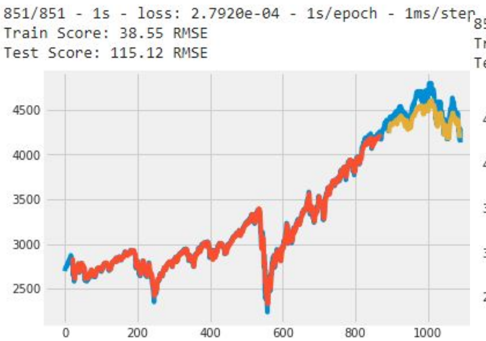


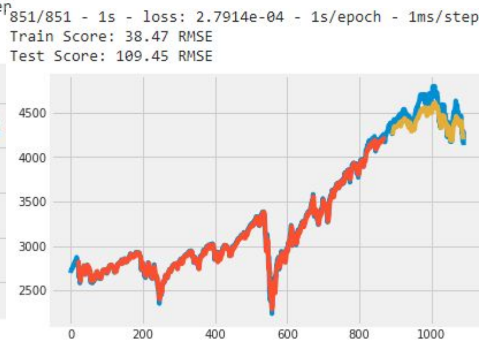


S&P 500 Starting 2018









Conclusion:
Observe that training with less data and more epochs can improve our testing result and at the same time allow us to have better forecasting and prediction values.

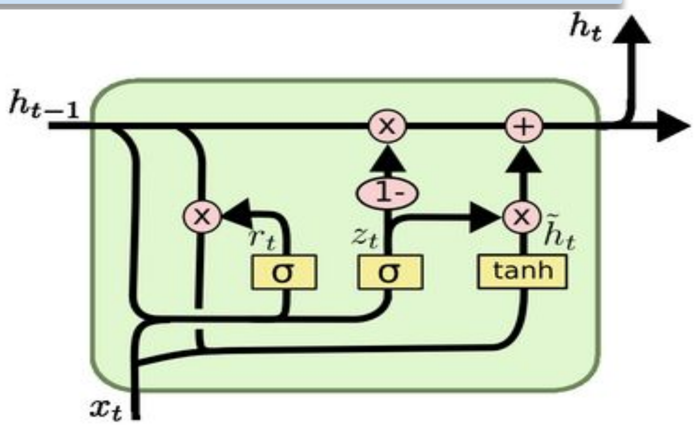
MSE/RMSE	AAPL	S&P 500
Epochs = 25	289.68/17.02	4290.25/65.50
Epochs = 50	210.83/14.52	9586.37/97.91
Epochs = 75	94.09/9.7	13252.61/115.21
Epochs = 100	130.42/11.42	11979.30/109.45

Columbia University
IN THE CITY OF NEW YORK

Gated Recurrent Unit(GRU) in Recurrent Neural Network

- ❑ The GRU is the newer generation of Recurrent Neural networks and is pretty similar to an LSTM. GRU's got rid of the cell state and used the hidden state to transfer information. It only has two gates, a reset gate and update gate.

- ❑ The update gate acts similar to the forget and input gate of an LSTM. It decides what information to throw away and what new information to add. The reset gate is another gate is used to decide how much past information to forget.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

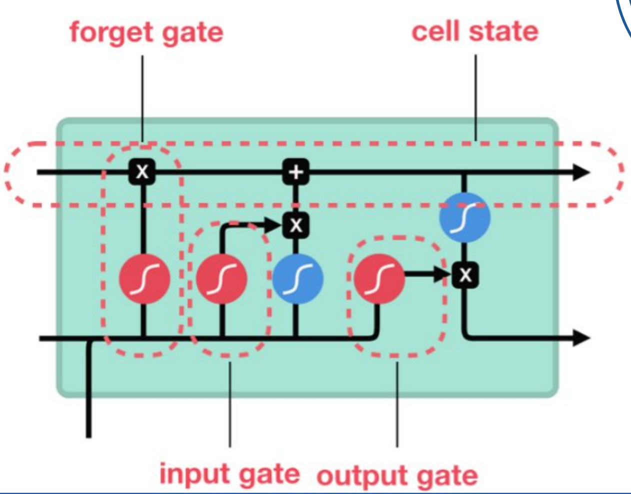
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

GRU vs. LSTM

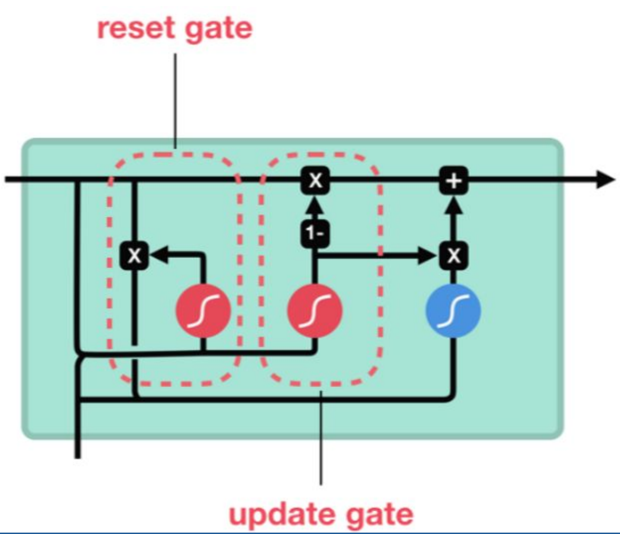
LSTM

- ❑ Three gate
- ❑ LSTM process internal memory
- ❑ In LSTM the input gate and target gate are coupled by an update gate. The responsibility of reset gate is taken by the two gates



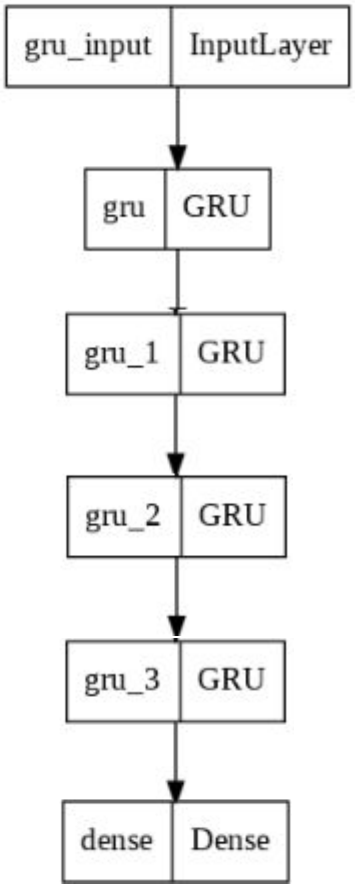
GRU

- ❑ Two gate
- ❑ GRU does not possess any internal memory
- ❑ GRU reset gate is applied directly to the previous hidden state



Fitting GRU Into Our Data

- To set our model, we add 4 hidden layers. For each layer, we drop out 20% nodes to stabilize the GRU Model. We also set 20 days as a window to predict the price in the next window.
- The next step is using the model to fit our data, we also use 80% data to train and 20% data to test and set epochs equal to 100.
- Finally, we can draw the graph of our prediction and calculate the MSE, MAE, and RMSE.



	AAPL	S&P 500
MSE	37.3876	11512.0608
MAE	4.7988	90.5938
RMSE	6.1145	107.2942

Fitting GRU Into Our Data

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Conclusion

Linear Regression Model:

- Limitations: it need **specific assumption.**
- Might not be suitable prediction when suffer **short-term volatility** in stocks.

Time Series Model:

- Hard to give a very precise prediction in a **long term period.**
- Created large error if **external events** happen.

*LSTM Model vs GRU Model (Preferred):

- They both **yield better** than previous two model.
- Their MSE will change due to the model parameter we choose(eg.epchos).