Machine Learning Engineer Nanodegree

Capstone Report: Chaitanya Potnis

Stock Price Trend Prediction Using LSTM. December 8 2018

Domain Background:

Predicting stock prices is always being research topic in the market and there are not even more than 100 research paper on stock predictions. As we all know Stock prices of a company depends on various features. If we are able get some features and develop a algorithm or model to predict stock prices, it won't be accurate as the real stock prices. Therefore with some limited feature we can predict the stock price trend (is it an upwards trend or an downwards trend). As we know stocks price are time series data and there are very few algorithms or model that handle the time series data well, therefore I selected a state of the art model that is an LSTM(Long Short Term Memory) which is an Recurrent Neural Network and it performs well on time series data. Presently with all the computation power and data we can build an powerful LSTM model to predict the stock price trend.

Stocks of a company are fuel that runs the whole organization. As we know that stocks are highly volatile in nature and first rule of finance is based on the Brownian motion which states that stock prices have random behavior and are continuously changing over the time and are driven by Brownian motion process. There are many factors affecting the stock of a particular company, like Company A's stock price depend on their rival Company B. If Company B launches a Product in market and it has a huge success so, it will affect the stock prices of their rival Company A and visa versa. In this scenario it's just one company but in reality a company's stock may be affected by it's own decision and products as well as the products and decision of their subsidiary, dependent companies and rivals. Therefore it makes prediction of future stock prices more difficult.
As most of the research work in stock price prediction is done using financial concepts like Monte Carlo technique and by using technical indicators like Rate of change, momentum, relative index strength, average direction index etc. But in this project I am using LSTM model which is a recurrent neural network to predict the future stock prices trend.

Problem Statement :

As discussed earlier there are various ways to find future stock prices by using financial concepts like Monte Carlo technique and by using technical indicators like Rate of change, momentum, relative index strength, average direction index etc. But in this project I will be using the power of Deep learning to find the future stock prices. First we will train the LSTM model with prices (consider Open price) which is a time series data. And try to predict the next day price. As mentioned earlier it not possible to predict the accurate price, we will convert the problem from regression to classification, to predict the trend that is whether the stock price will increase (upward trend) or it will decrease (downward trend) which makes this problem a binary classification problem. This information itself can help many organizations and individuals in decision making.

Dataset and Inputs:

In this project is use Google Stock Data to predict the trend in the company's stock prices. The stock data can be retrieved from pandas datareader package source being yahoo finance.
Features in data are : Open, High, Low, Close, Adj Close, Vloume.

	Open	High	Low	Close	Adj Close	Volume
Date						
2017-01-03	778.809998	789.630005	775.799988	786.140015	786.140015	1657300
2017-01-04	788.359985	791.340027	783.159973	786.900024	786.900024	1073000
2017-01-05	786.080017	794.479980	785.020020	794.020020	794.020020	1335200
2017-01-06	795.260010	807.900024	792.203979	806.150024	806.150024	1640200
2017-01-09	806.400024	809.966003	802.830017	806.650024	806.650024	1272400

First 5 Rows of Google Dataset.

Solution Statement:

Data analysis is a key step in problem solving and deep learning. First, we need to perform Exploratory dataset analysis to find patterns ,correlations, calculating Moving averages, Daily Returns in data[1]. Next step is problem solving here the problem is to predict the future stock prices trend.

As the stock data is random and volatile, it will be difficult to predict the "Exact stock Price " but by using LSTM and feeding it previous 60 time steps, the predicted value obtained can much closer to the real stock price and can capture the maximum trend[3]
LSTM is a version of Recurrent Neural Network which doesn't face Vanishing Gradient Problem and can handle the long time series data very well as compared to Traditional RNN.
LSTM also can learn patterns in stock prices which will be help for Stock Trend Prediction.
Therefore this will be a perfect model for the our Time Series Data(Stock Prices).

Evaluation Metrics:

Using the Actual Prices we can find the Actual Trend and in similar way the predict trend can be calculated by Predicted prices.

Algorithm used for calculating the trend:

if (Next Day Open Price > Previous Day open Price):
Trend = +1
else:
Trend = -1

+1 Denotes Upwards Trend
-1 Denotes Downwards Trend

Then the we can calculate the accuracy from predicted trend and actual trend.

Project Design:

- Programming Language : Python
- Libraries : Numpy, Pandas, Matplotlib, Seaborn, Keras
- Workflow
    - Data Gathering:
      Using Pandas Datareader and yahoo finance data will be gathered.

    - Exploratory Data Analysis:
      EDA gives better understanding of the data and problem.
      Plot the stocks price
      Calculating moving average and daily return and plotting them

    - Data Preprocessing:
      There is lot of pre-processing required for the data and we need previous 60 days stock data(that is 3 financial months) to predict the next day stock price. Pre-processing includes changing the dimensions, creating special data-frames which include **T-60** (previous 60 time steps) features for **T'**th time step and this row will be used to predict the **T+1** Time step data

    - Benchmark Model:
      A simple Naïve baseline model which will help to evaluate the final LSTM model. Baseline model will be a random choice of upwards trend or a downwards trend (with Probability of 50 -50%)

    - Designing and Developing the LSTM model:
      Designing the LSTM model with Kears (Tensorflow as Backend)

    - Training the LSTM Model:
      Train the model on a n number of epochs where n is number of epoch eher model doesn't overfit or underfit.

- Testing the LSTM model:
  Test the model a test set which is kept aside during training

- Evaluating the Model:
  Calculate the accuracy from the Actual Trend and Predicted Trend

Data Set Overview:

For this project I am using dataset of Stock Prices of Apple, Google, Microsoft, Amazon.

Data Set Overview:

	Open	High	Low	Close	Adj Close	Volume
Date						
2017-01-03	778.809998	789.630005	775.799988	786.140015	786.140015	1657300
2017-01-04	788.359985	791.340027	783.159973	786.900024	786.900024	1073000
2017-01-05	786.080017	794.479980	785.020020	794.020020	794.020020	1335200
2017-01-06	795.260010	807.900024	792.203979	806.150024	806.150024	1640200
2017-01-09	806.400024	809.966003	802.830017	806.650024	806.650024	1272400

First 5 Rows of Google Dataset.

	Open	High	Low	Close	Adj Close	Volume
Date						
2017-01-03	115.800003	116.330002	114.760002	116.150002	113.847588	28781900
2017-01-04	115.849998	116.510002	115.750000	116.019997	113.720169	21118100
2017-01-05	115.919998	116.860001	115.809998	116.610001	114.298470	22193600
2017-01-06	116.779999	118.160004	116.470001	117.910004	115.572701	31751900
2017-01-09	117.949997	119.430000	117.940002	118.989998	116.631287	33561900

First 5 Rows of Apple Dataset.

	Open	High	Low	Close	Adj Close	Volume
Date						
2017-01-03	62.790001	62.840000	62.130001	62.580002	60.926991	20694100
2017-01-04	62.480000	62.750000	62.119999	62.299999	60.654381	21340000
2017-01-05	62.189999	62.660000	62.029999	62.299999	60.654381	24876000
2017-01-06	62.299999	63.150002	62.040001	62.840000	61.180119	19922900
2017-01-09	62.759998	63.080002	62.540001	62.639999	60.985405	20256600

First 5 Rows of Microsoft Dataset.

	Open	High	Low	Close	Adj Close	Volume
Date						
2017-01-03	757.919983	758.760010	747.700012	753.669983	753.669983	3521100
2017-01-04	758.390015	759.679993	754.200012	757.179993	757.179993	2510500
2017-01-05	761.549988	782.400024	760.260010	780.450012	780.450012	5830100
2017-01-06	782.359985	799.440002	778.479980	795.989990	795.989990	5986200
2017-01-09	798.000000	801.770020	791.770020	796.919983	796.919983	3440100

First 5 Rows of Amazon Dataset.

Data Preprocessing

There is lot of pre-processing required for the data and we need previous 60 days stock data(that is 3 financial months) to predict the next day stock price. Pre-processing includes changing the dimensions, creating special data-frames which include **T-60** (previous 60 time steps) features for **T'**th time step and this row will be used to predict the **T+1** Time step data.

Training Data:

```
[[0.08545249 0.09660926 0.09393314 ... 0.07812537 0.07999865 0.08461274]
 [0.09660926 0.09393314 0.09118315 ... 0.07999865 0.08461274 0.08591389]
 [0.09393314 0.09118315 0.07950958 ... 0.08461274 0.08591389 0.08435434]
 ...
 [0.92118079 0.92448743 0.93058039 ... 0.95482243 0.95211037 0.95170163]
 [0.92448743 0.93058039 0.93000453 ... 0.95211037 0.95170163 0.95731171]
 [0.93058039 0.93000453 0.93123052 ... 0.95170163 0.95731171 0.9380481 ]]
```

Special Data frame to train the LSTM Model Contain 60 timesteps (X_train).

List of Stock Price to be Predicted:

```
[0.08591389 0.08435434 0.07422185 ... 0.95731171 0.9380481  0.93697064]
```

Each element in list is a stock price which is Label (y_train) for our Model to be trained.

Model Input:

[0.08545249 0.09660926 0.09393314 ... 0.07812537 0.07999865 0.08461274] -> **0.08591389**

X_train **y_train**

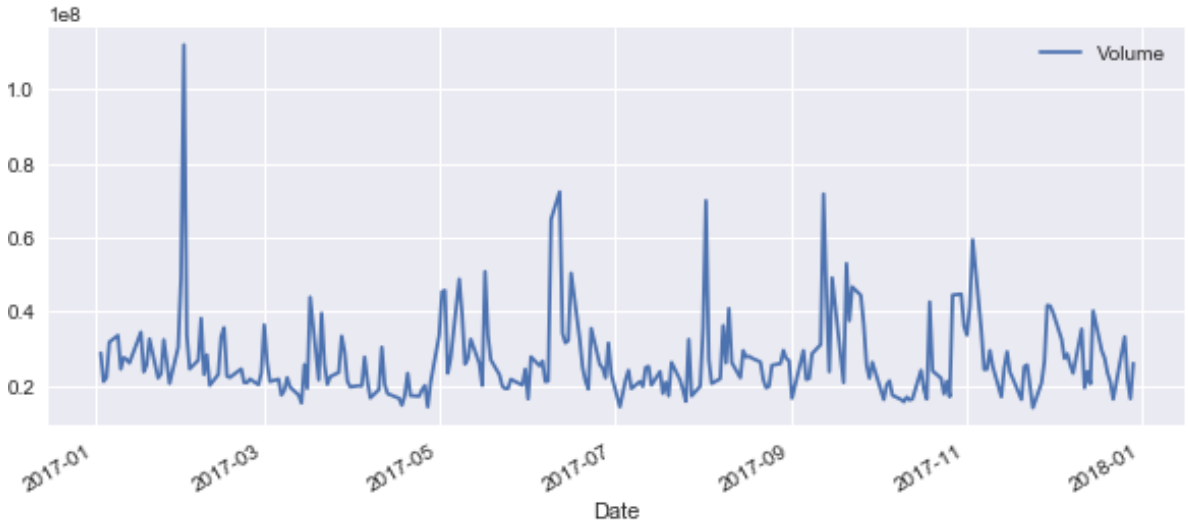
Exploratory Data Analysis

	Open	High	Low	Close	Adj Close	Volume
count	251.000000	251.000000	251.000000	251.000000	251.000000	2.510000e+02
mean	150.444900	151.406056	149.487650	150.551075	148.903352	2.713162e+07
std	14.744788	14.822607	14.479401	14.621191	15.111322	1.128930e+07
min	115.800003	116.330002	114.760002	116.019997	113.720169	1.402670e+07
25%	141.895005	142.739998	141.029999	141.815002	139.606492	2.041325e+07
50%	152.449997	153.860001	151.130005	152.740005	150.784286	2.440950e+07
75%	159.714996	160.710007	158.540001	159.855004	158.595932	2.994720e+07
max	175.110001	177.199997	174.860001	176.419998	175.703629	1.119850e+08

In the above Picture we can analyze different statistics about the Apple Stock Data. We can see that highest volume that was traded on a particular day was 1.119850e+08, Highest stock value was 177.199997 on a particular day between 2017 – 2018. We can Analyze different Quantile range of the Features [1].

Adj Close Price from 2017 to 2018

The above Graph shows the change in Adj close from January 2017 to January 2018[1].



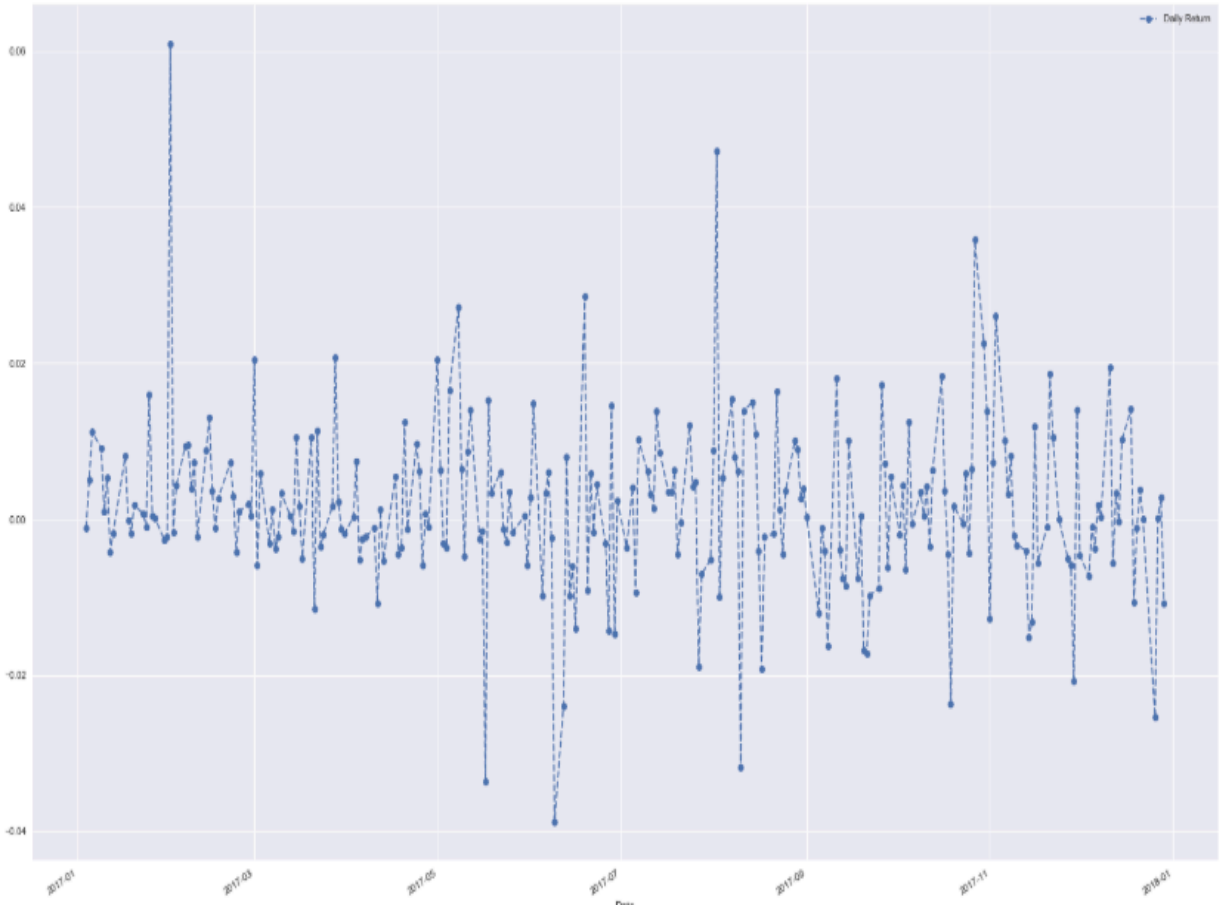
Apple stock' Volume traded from 2017 to 2018

The above Graph shows the Volume Traded from January 2017 to January 2018 for Apple Stock[1].

Adj Close, Moving Average for 10 days, Moving Average for 20 Days, Moving Average for 50 Days for Apple Stocks.

Moving Average : The Moving Average calculates the series of averages in different sets of Time window . Above I have calculated Moving Average for 10 days, Moving Average for 20 days, Moving Average for 50 days. As the time window increases the graph get more smoother [4].

Daily Returns



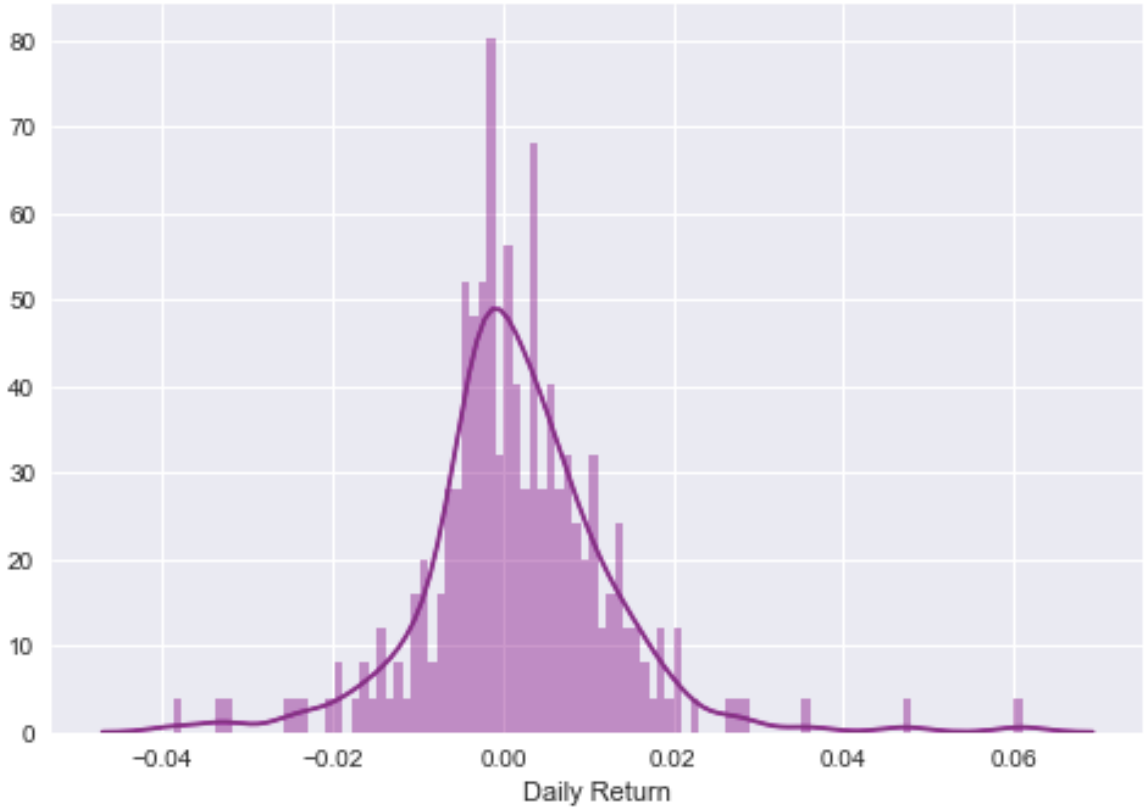
Daily return for Apple Stocks.

Daily returns :

(Close Price Today – Close Price Yesterday) / (Close Price Yesterday)

Daily returns give a number which represent that the given stock on a particular day made a profit or loss.

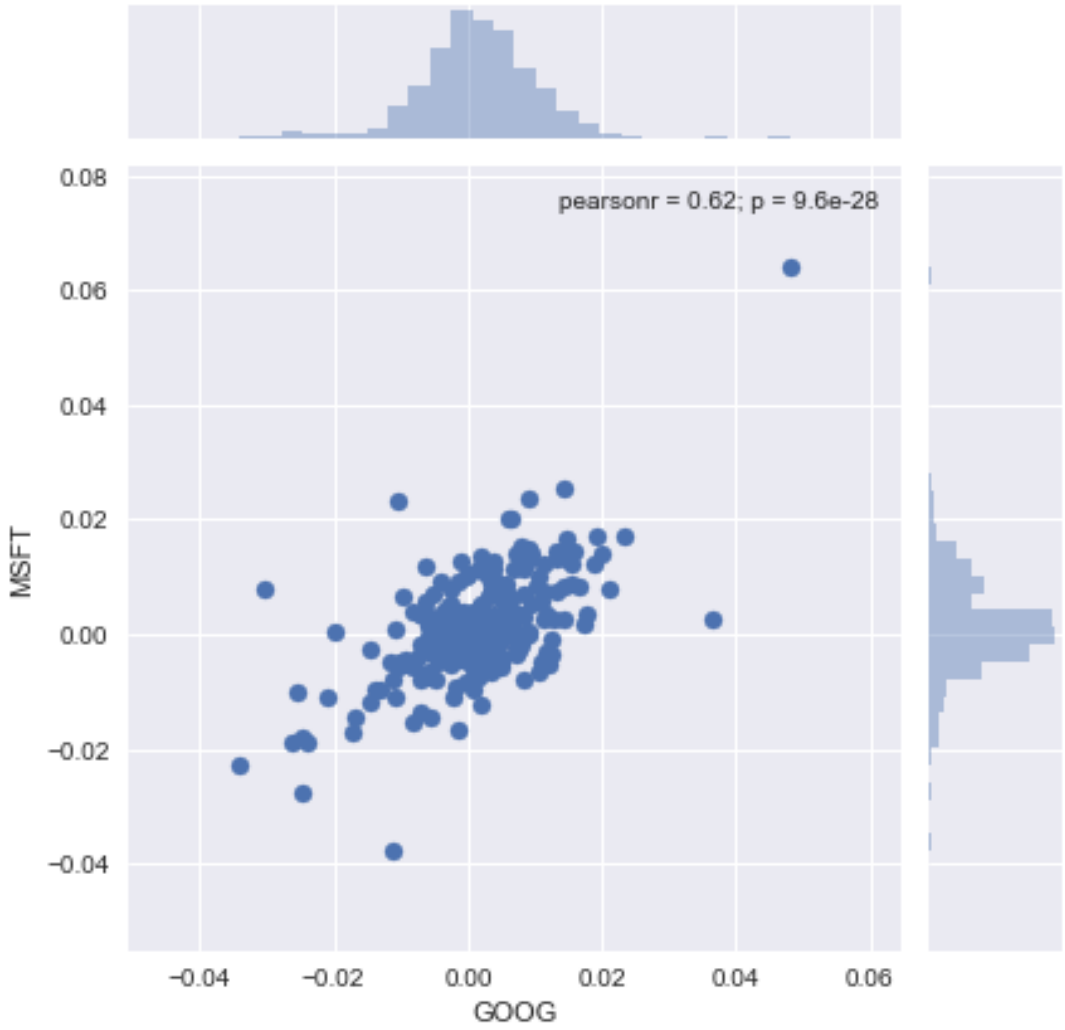
Above Graph shows the Daily Return of Apple Stock from 2017 to 2018 [5].



Average Daily return for Apple Stocks.

Above Graph show that the daily returns of the Apple Stocks from 2017 to 2018 were on positive side of the distribution [5].

Correlation between Stock's of Different Companies.

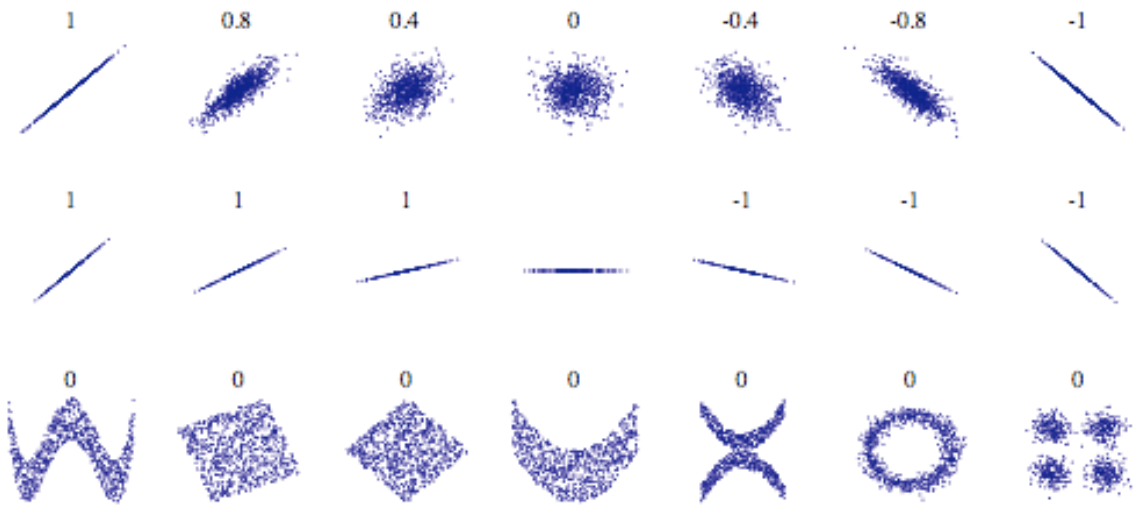


Correlation plot between Apple and Microsoft Stocks

Above Graph shows the correlation between Microsoft and Apple Stock feature used to plot is daily returns of the both companies from 2017 to 2018.

We can see that the Pearson product-moment correlation coefficient is 0.62
Which shows some light correlation between the two companies stock's.

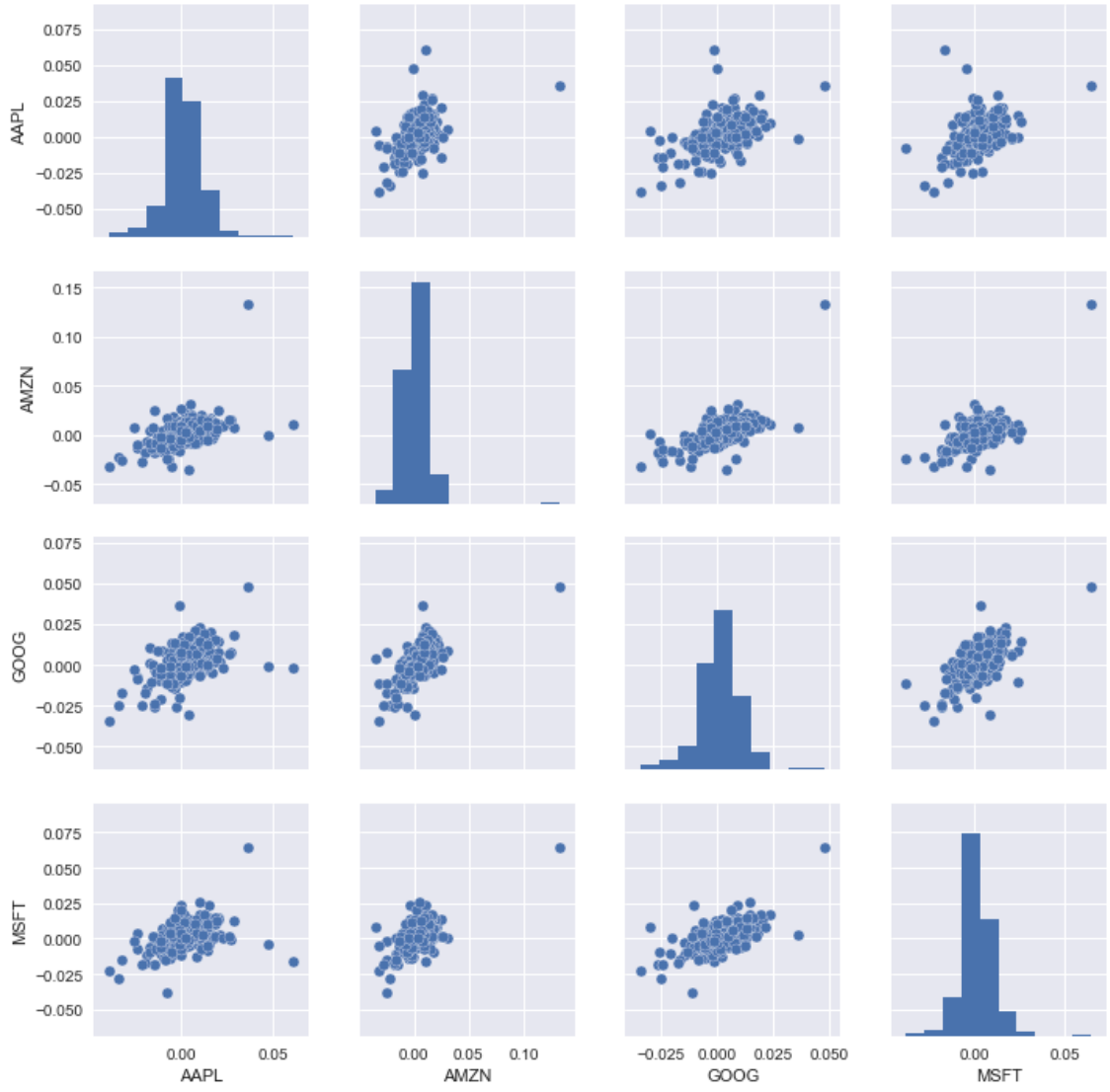
Pearson Product-moment Correlation Coefficient



Above Figure shows different graphs and their corresponding Pearson product-moment correlation coefficient.

Pearson product-moment correlation coefficient:
How correlated are the daily percentage returns are. It is also defined as a measure of the strength of a linear association between two variables

Correlation Between Apple Google, Microsoft and Amazon.



Correlation plot between Apple, Microsoft, Amazon, Google Stocks

Above Figure shows correlation between Apple, Google, Microsoft, Amazon.
We can see that Google and Microsoft are the most Correlated of the all companies.

Benchmark Model:

As this a binary classification problem that is I am trying to predict whether next day stock prices goes up(upward trend) or it goes down(downward trend). We can Randomly predict the stock price trend for a particular day as it will have 50-50 % probability.

Results:

	Predicted Benchmark	Real
0	1	1
1	1	-1
2	1	1
3	-1	1
4	-1	1
5	-1	-1
6	-1	1
7	-1	1
8	-1	-1
9	-1	-1
10	-1	-1
11	1	1
12	1	1
13	-1	1
14	-1	1
15	1	1
16	1	-1
17	1	-1

Accuracy = 0.50 %

Prediction Approaches

There are various way to predict Stock Prices
The most common ways using Machine learning and Deep learning are:

1) Financial Concepts such as Technical Indicators and using it as feature for the machine learning model.
2) Deep Learning with Time Series Data(Stock's).

In the first method we calculate the following technical indicators:

1)Trend indicators
- a) Moving Averages levels.
- b) Parabolic Stop and Reverse (Parabolic SAR
- c) Moving Average Convergence Divergence (MACD)

2) Momentum indicators
- a) Stochastic Oscillator: price range.
- b)Commodity Channel Index (CCI):
- c)Relative Strength Index (RSI)

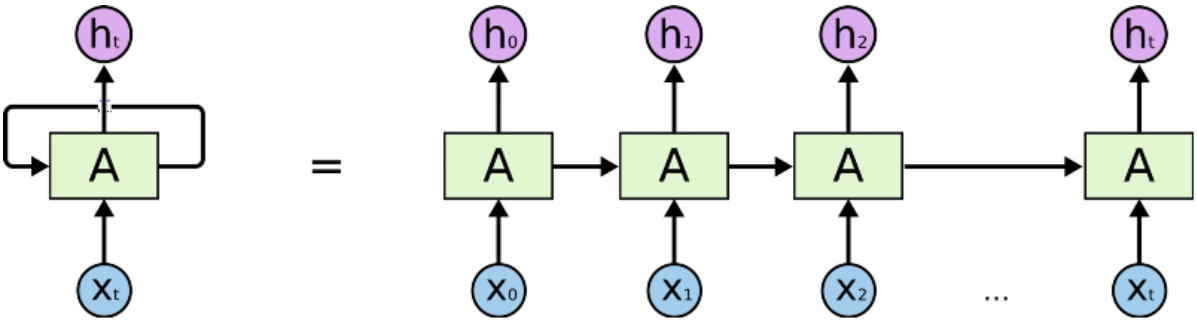
3) Volatility Indicators
- a) Bollinger bands
- b) Average True Range
- c) Standard Deviation

4) Volume Indicators
- a) Chaikin Oscillator
- b) On-Balance Volume (OBV)
- c) Volume Rate of Change:

After calculating these Indicators, these indicators are used as features and feed into a Machine Learning Classification or Regression Model to Predict the Stock trend or Stock Price Respectively.[2]

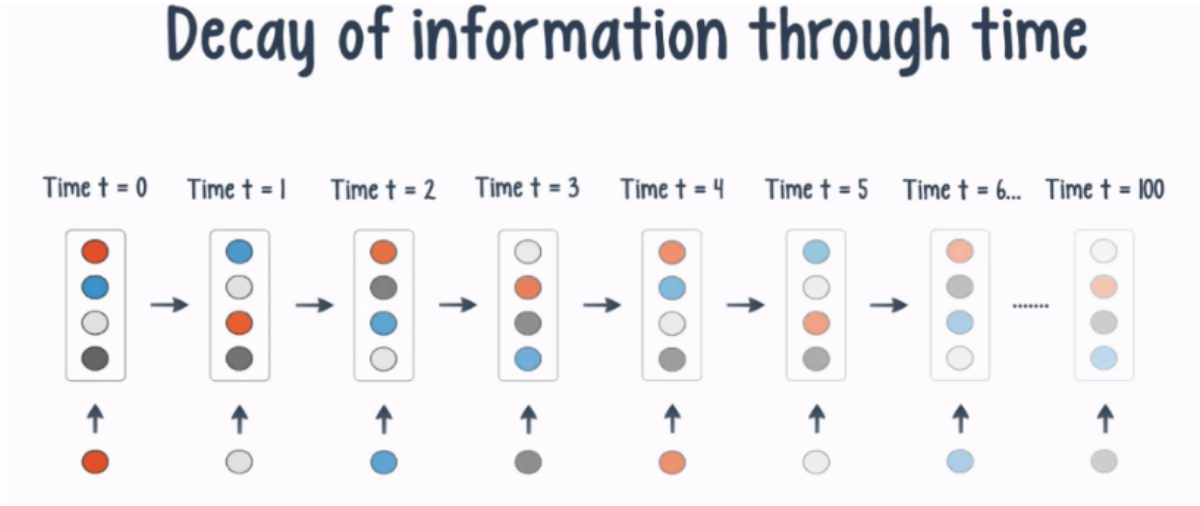
This Approach gives a accuracy between 50% to 65%.

Second Approach is by using the power of Deep Learning.



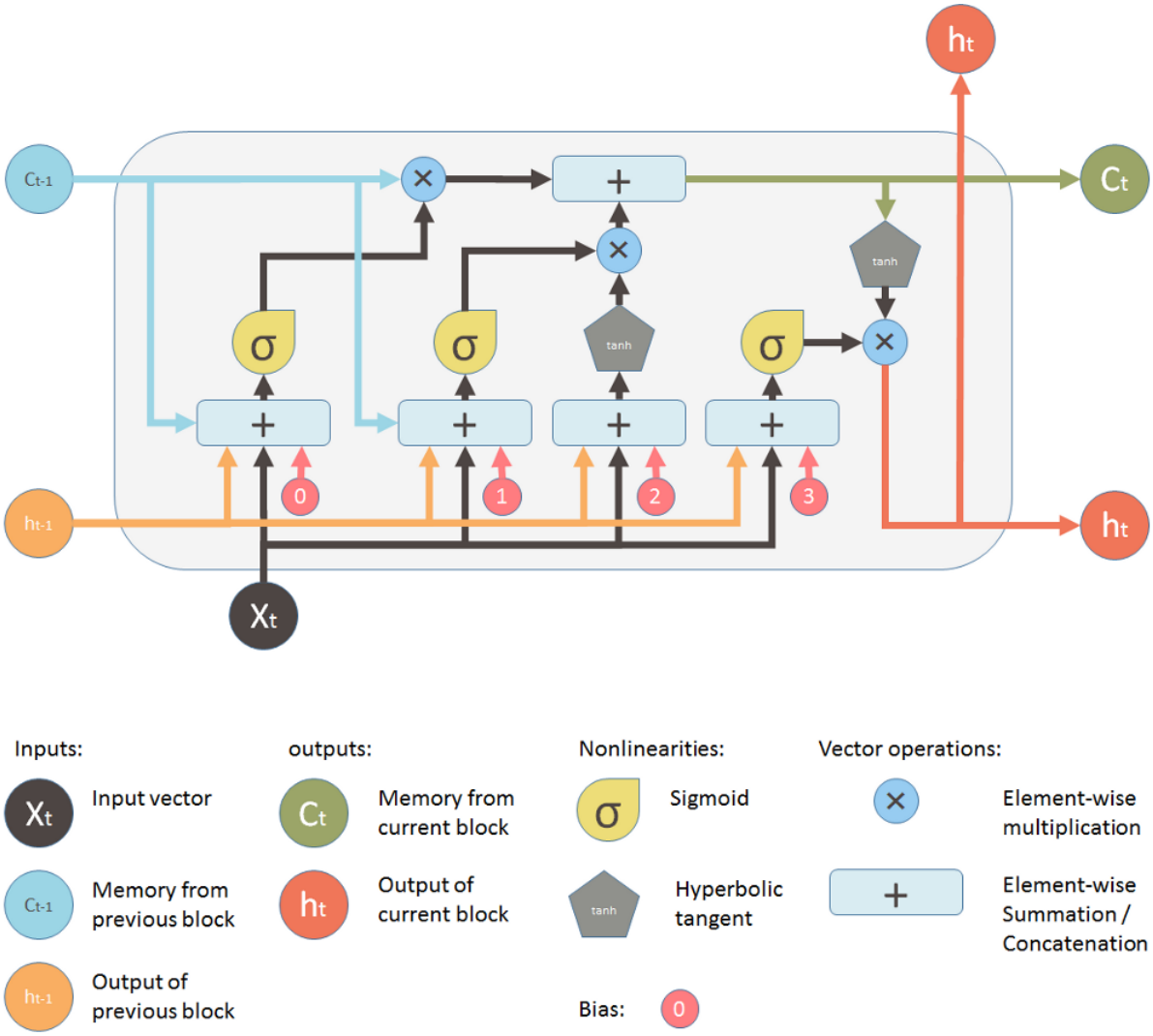
Recurrent Neural Network Handle the time series Data very well. We can use RNN (Recurrent Neural Network) and predict the next day stock price as well as the stock trend(Upwards or Downwards)

But for these problem RNN can handle the data as explained earlier 'to predict the next day stock price I will feed the stock prices of last 60 days(that is 3 financial months)'.
Recurrent Neural Networks face a major problem while Back-Propagation through time, this problem is know as Vanishing Gradient.



This problem was solved by Sepp Hochreiter and Jugen Schmidhuber in 1997, they came up with a solution known as LSTM (Long Short Term Memory) [6].

Long Short Term Memory:



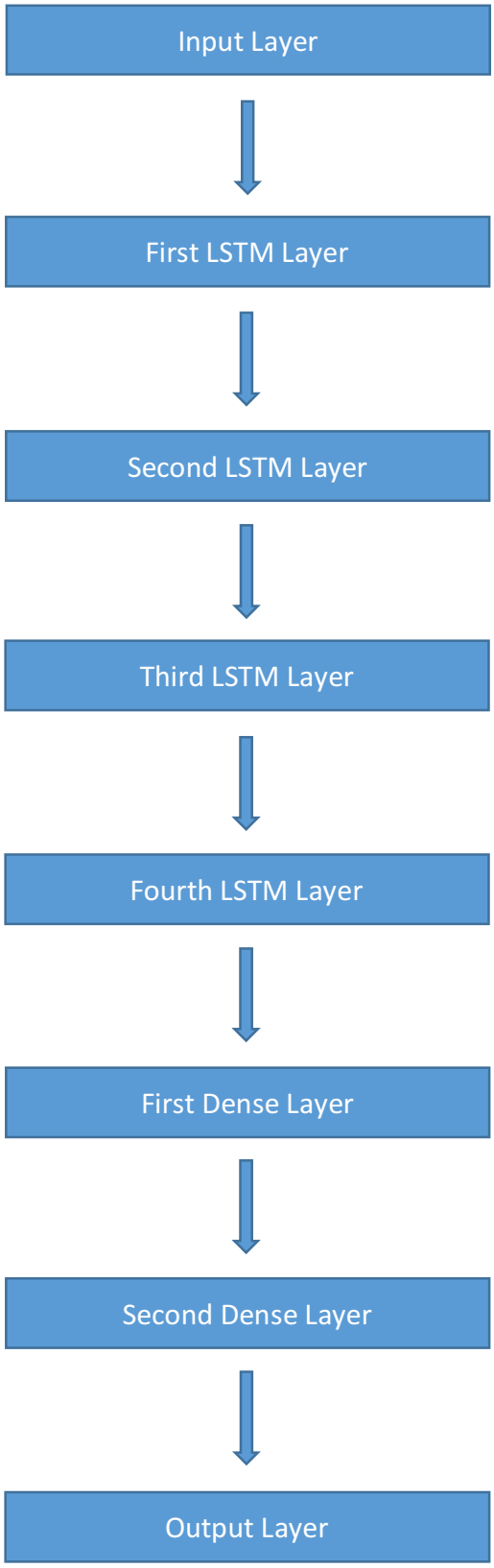
LSTM(Long Short Term Memory) Cell.

LSTM is a version of Recurrent Neural Network which doesn't face Vanishing Gradient Problem and can handle the long time series data very well as compared to Traditional RNN.
LSTM also can learn patterns in stock prices which will be help for Stock Trend Prediction [2].

Therefore this will be a perfect model for the our Time Series Data(Stock Prices).

Long Short Term Memory Architecture:

LSTM used in this project has the following architecture:



Training the LSTM Model

Optimizer used for these model was Adam which is Adaptive Moment Estimation and combines the power of AdaGrad and RMSProp.

Loss Function or Error Function used for these model was :Mean Squared Error.

This model was trained of 100 epochs and with batch size of 32.

Following was the loss at the a particular epochs:

Epoch	Loss
1	0.0607
10	0.0050
20	0.0034
30	0.0028
40	0.0026
50	0.0024
60	0.0022
70	0.0021
80	0.0021
90	0.0017
100	0.0017

We can see that First epoch had a loss of 0.475 and gradually it started to decrease and the final epoch that is 100'th epoch had a loss of 0.0014 which shows that the model had trained very well on the data.

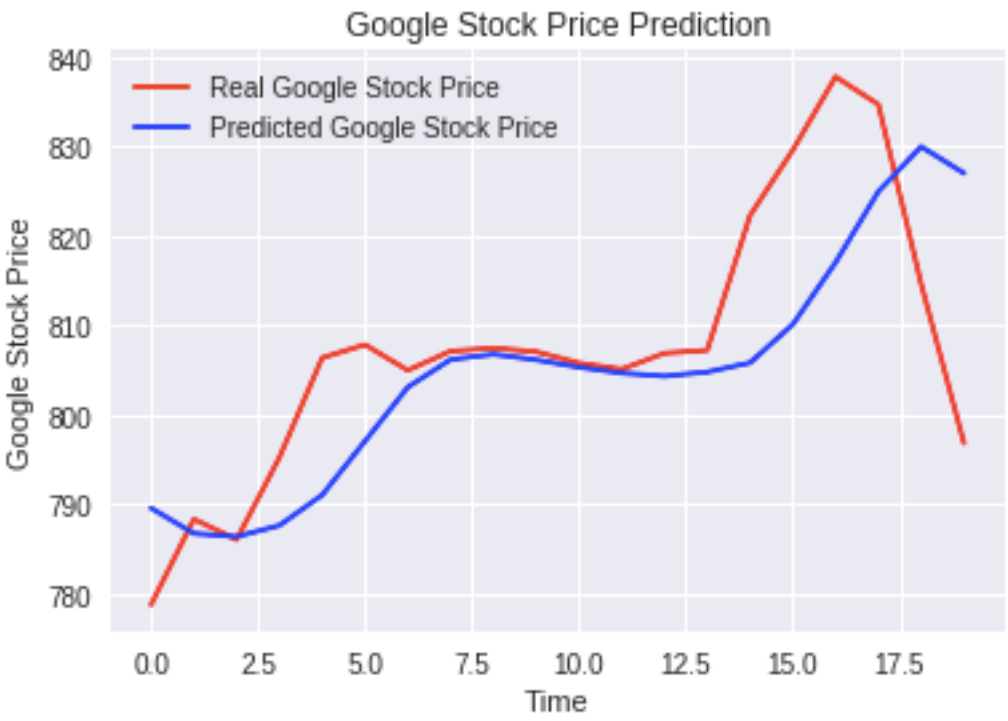
Training Time : 3402 Seconds (56.6 Minutes.)

Training Data : Google Stock's Open price from 2012 to 2017 (5 Years of data)

Results

After Training it on 5 Years of Stock price it could learn the Pattern very well .

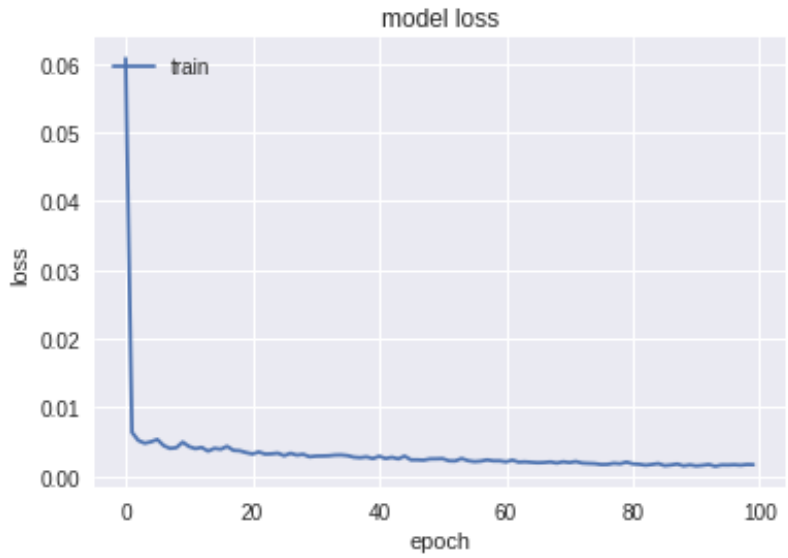
Testing Data : 1 Jan to 31 Jan 2018.



Comparing the Predicted and Actual Prices(Google Stock's Open Price).

We can see that the model could predict the trend (Upwards or Downwards Trend) very well.

The Difference in the prices are because as explained earlier the Stock Prices are very Volatile and follows the Brownian Motion Principle and there many factor affecting the stock prices. And it doesn't only depend on previous Stock Prices.



Classification Accuracy

Using the Actual Prices we can find the Actual Trend:

Algorithm used for calculating the trend:

if (Next Day Open Price > Previous Day open Price):
Trend = +1
else:
Trend = -1

+1 Denotes Upwards Trend
-1 Denotes Downwards Trend

	Predicted Stock Price Trend	Real Stock Price
0	-1	1
1	-1	-1
2	1	1
3	1	1
4	1	1
5	1	-1
6	1	1
7	1	1
8	-1	-1
9	-1	-1
10	-1	-1
11	-1	1
12	1	1
13	1	1
14	1	1
15	1	1
16	1	-1
17	1	-1
18	-1	-1

Calculating the Accuracy Score between Actual Trend and Predicted Trend:

Accuracy Score : 0.7368421
F1 Score : 0.73

Conclusion

By using the power of LSTM network, we can handle the time series data very well and can capture the maximum trend. With the help of visualizations technique we can verify the real stock data and predict stock data trend.

And the accuracy of 73.68421% (0.7368421) is also high and therefore we can conclude that the model could predict the Trend with Probability of 73.68% in the given Month of January 2018.

Reference:

[1] Andrew Gelman, *Exploratory Data Analysis for Complex Models*, Columbia University

[2] Xinjie Di, *Stock Trend Prediction with Technical Indicators using SVM,* SCPD student from Apple Inc

[3] Sepp Hochreiter, J□urgen Schmidhuber, *LONG SHORT-TERM MEMORY*

[4] Rob J Hyndman, *Moving averages*

[5] Jerold B. Warner and Stephen J. Brown, *USING DAILY STOCK RETURNS: THE CASE OF EVENT STUDIES,* University of Rochester

[6] Sepp Hochreiter, *THE VANISHING GRADIENT PROBLEM DURING LEARNING RECURRENT NEURAL NETS AND PROBLEM SOLUTIONS,* Institut fur Informatik, Technische Universitat Munchen