STOCK MARKET PREDICTION

Report submitted in partial fulfillment of the requirement for
the certificate of Project Completion

Under the Supervision of

Bipul Sahi

By

Shashank Shukla (8279726437)
Yash Agarwal (8979339036)

Diginique TechLabs in assosciation with Cognizance, IIT Roorkee

Session :- 2019-20

Link of project on github- https://github.com/shanky1947/Stock-Market-Prediction/blob/master/stock_newnew.ipynb

DECLARATION

This is to certify that Report entitled “STOCK MARKET PREDICTION” which is submitted in partial fulfillment of the requirement for the certificate of project completion comprises only original work and studies carried out by students himself. The matter embodied in this report has not been submitted for the award of any other degree.

Date: 30-07-19

Approved by:

Signature
(Bipul Sahi)
Project guide
Mentor

(Sudhir Chaudary)
Director
Diginique TechLabs

TABLE OF CONTENT

ABSTRACT

This is a project report on STOCK MARKET PREDICTION. During the making or developing of this project we explored new ideas and functionality behind the working of a various types of neural networks. This project is the output of our planning, schedule, programming skill and the hard work, and this report reflects our steps taken at various levels of programming skill, planning and schedule. We have learnt a lot during this project and liked the improvement in our testing skills and deep concept related to these kinds of projects. Our project is STOCK MARKET PREDICTION. We have used special types of neural networks so called LSTM (Large Short Term Memory). It is called so because LSTM are used when data which was used previous before is also required for training and predicting output. In case of stock market, data of past days are also required to predict stock market of next day accurately.

We are happy with the accuracy it showed but really prediction stock market is a great challenge as stock changes with the events happening in real world and are not fully depended on data of past. Although we tried out best to predict stocks of next day with some good accuracy. Fist we thought to use RNN (Recurrent Neural Network) but due to low accuracy we moved on to LSTM.

CHAPTER - 1

INTRODUCTION

Predicting how the stock market will perform is one of the most difficult things to do. There are so many factors involved in the prediction – physical factors vs. physiological, rational and irrational behavior, etc. All these aspects combine to make share prices volatile and very difficult to predict with a high degree of accuracy.

First, we thought can we use machine learning as a game changer in this domain? Using features like the latest announcements about an organization, their quarterly revenue results, etc., machine learning techniques have the potential to unearth patterns and insights we didn't see before, and these can be used to make unerringly accurate predictions.

In this project, we will work with historical data about the stock prices of a publicly listed company. We will implement a mix of machine learning algorithms to predict the future stock price of this company, starting with LSTM.

Broadly, stock market analysis is divided into two parts – Fundamental Analysis and Technical Analysis.

- Fundamental Analysis involves analyzing the company's future profitability on the basis of its current business environment and financial performance.

- Technical Analysis, on the other hand, includes reading the charts and using statistical figures to identify the trends in the stock market.

In this project, our focus will be on the technical analysis part. We used dataset from Quandl and for this particular project, we have used the data for 'Tata Global Beverages'.

CHAPTER – 2

DATA MODIFICATION

There are multiple variables in the dataset – date, open, high, low, last, close, total_trade_quantity, and turnover.

- The columns *Open* and *Close* represent the starting and final price at which the stock is traded on a particular day.

- *High*, *Low* and *Last* represent the maximum, minimum, and last price of the share for the day.

- *Total Trade Quantity* is the number of shares bought or sold in the day and *Turnover (Lacs)* is the turnover of the particular company on a given date.

Another important thing to note is that the market is closed on weekends and public holidays. Notice the above table again, some date values are missing – 2/10/2018, 6/10/2018, 7/10/2018. Of these dates, 2nd is a national holiday while 6th and 7th fall on a weekend.

Out[2]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-10-08	208.00	222.25	206.85	216.00	215.15	4642146.0	10062.83
1	2018-10-05	217.00	218.60	205.90	210.25	209.20	3519515.0	7407.06
2	2018-10-04	223.50	227.80	216.15	217.25	218.20	1728786.0	3815.79
3	2018-10-03	230.00	237.50	225.75	226.45	227.60	1708590.0	3960.27
4	2018-10-01	234.55	234.60	221.05	230.30	230.90	1534749.0	3486.05

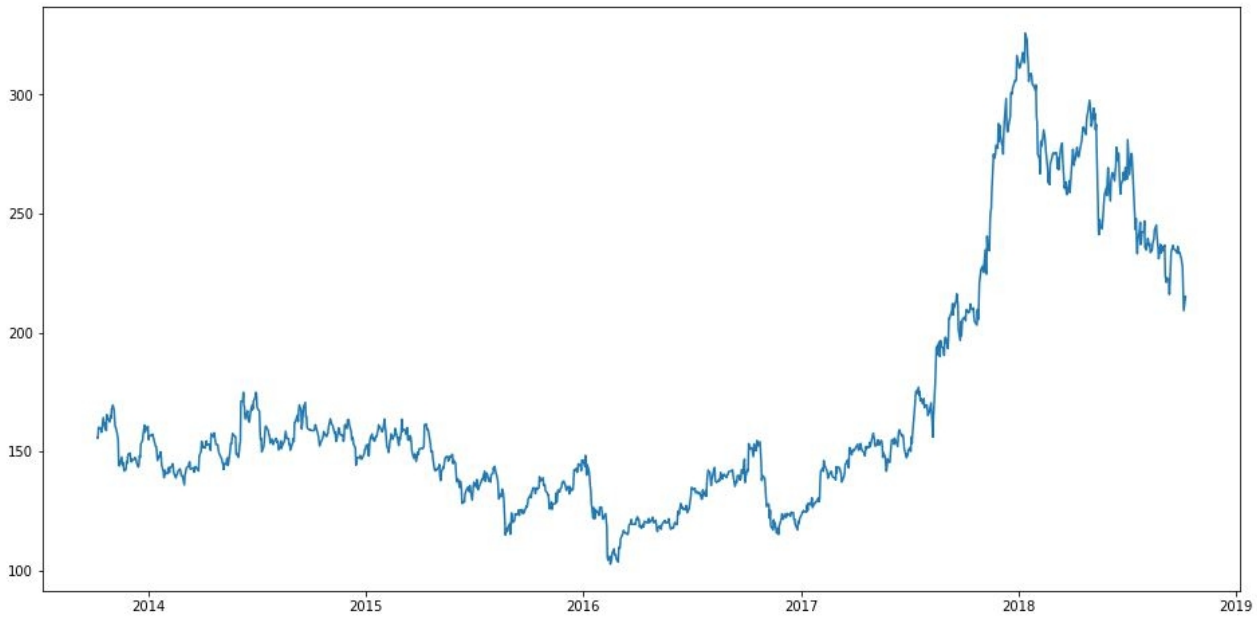
The profit or loss calculation is usually determined by the closing price of a stock for the day, Hence we will consider the closing price as the target variable. Let's plot the target variable to understand how it's shaping up in our data.

In [170]:
```
#setting index as date
df['Date'] = pd.to_datetime(df.Date,format='%Y-%m-%d')
df.index = df['Date']
df.head()
```

Out[170]:

Date	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
2018-10-08	2018-10-08	208.00	222.25	206.85	216.00	215.15	4642146.0	10062.83
2018-10-05	2018-10-05	217.00	218.60	205.90	210.25	209.20	3519515.0	7407.06
2018-10-04	2018-10-04	223.50	227.80	216.15	217.25	218.20	1728786.0	3815.79
2018-10-03	2018-10-03	230.00	237.50	225.75	226.45	227.60	1708590.0	3960.27
2018-10-01	2018-10-01	234.55	234.60	221.05	230.30	230.90	1534749.0	3486.05

Out[3]: [<matplotlib.lines.Line2D at 0x2454d15eba8>]



CHAPTER-3

TECHNOLOGY PLATFORM

We used Jupyter Notebook on Anaconda Navigator. We took help of python libraries like tensorflow, sklearn, keras, pandas, numpy, matplotlib, etc.

CHAPTER-4

TESTING

```
In [101]: model.compile(loss='mean_squared_error', optimizer='adam')
          model.fit(x_train, y_train, epochs=5, batch_size=1, verbose=2)

          WARNING:tensorflow:From C:\ProgramData\Anaconda3\envs\myenv\lib\site-packages\tensorflow\python\keras\utils\losses_utils.py:17
          0: to_float (from tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
          Instructions for updating:
          Use tf.cast instead.
          WARNING:tensorflow:From C:\ProgramData\Anaconda3\envs\myenv\lib\site-packages\tensorflow\python\ops\math_ops.py:3066: to_int32
          (from tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
          Instructions for updating:
          Use tf.cast instead.
          Epoch 1/5
           - 209s - loss: 0.0045
          Epoch 2/5
           - 188s - loss: 0.0013
          Epoch 3/5
           - 193s - loss: 0.0014
          Epoch 4/5
           - 180s - loss: 0.0010
          Epoch 5/5
           - 174s - loss: 8.2781e-04

Out[101]: <tensorflow.python.keras.callbacks.History at 0x280e86daeb8>
```

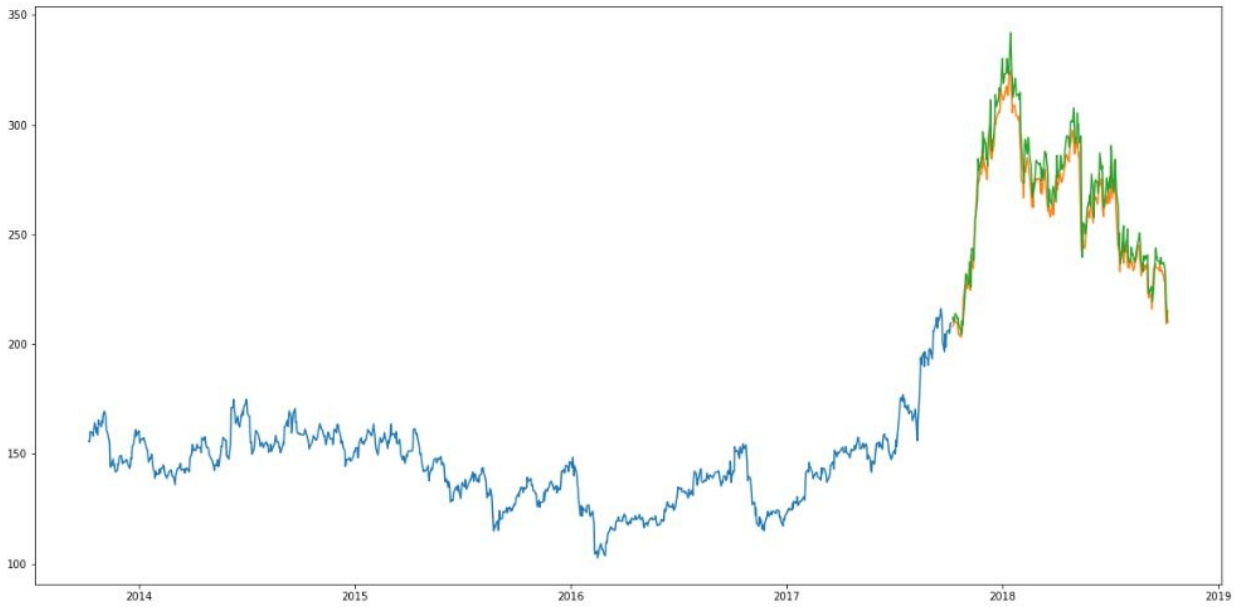
```
In [201]: closing_price = scaler.inverse_transform(closing_price)
          closing_price

Out[201]: array([[212.16121],
                 [210.594  ],
                 [210.08562],
                 [211.08871],
                 [214.03137],
                 [212.88107],
                 [211.49576],
                 [211.98015],
                 [208.85457],
                 [205.94345],
                 [204.29813],
                 [210.84212],
                 [208.6133 ],
                 [211.53862],
                 [223.19499],
                 [232.01456],
                 [231.95163],
                 [230.8679 ],
                 [228.68845],

In [202]: #tells how much our predicted value is away from line of best fit
          rms=np.sqrt(np.mean(np.power((test-closing_price),2)))
          rms

Out[202]: 9.123841630467266
```

Out[203]: [<matplotlib.lines.Line2D at 0x280f8735748>,
<matplotlib.lines.Line2D at 0x280f8735a20>]



CHAPTER-5

CODE

```
#import packages

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

df = pd.read_csv('C://Users/shash/Downloads/NSE-TATAGLOBAL11.csv')

df

#for normalizing data

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature_range=(0, 1))

#setting figure size

from matplotlib.pylab import rcParams

rcParams['figure.figsize'] = 20,10

#plot

plt.figure(figsize=(16,8))

plt.plot(df['Close'], label='Close Price history')

#setting index as date

df['Date'] = pd.to_datetime(df.Date,format='%Y-%m-%d')

df.index = df['Date']

df.head()

#plot

plt.figure(figsize=(16,8))

plt.plot(df['Close'], label='Close Price history')

import tensorflow as tf

import sklearn
```

```
from sklearn.preprocessing import MinMaxScaler
#for normalizing data
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 1))
Sequential=tf.keras.models.Sequential()
Dense=tf.keras.layers.Dense
Dropout=tf.keras.layers.Dropout
LSTM=tf.keras.layers.LSTM
#arranging in ascending order of index value
data = df.sort_index(ascending=True, axis=0)
#making a data frame with only date and close as columns
new_data = pd.DataFrame(index=range(0,len(df)),columns=['Date', 'Close'])
#entering data to our new empty data frame
for i in range(0,len(data)):
    new_data['Date'][i] = data['Date'][i]
    new_data['Close'][i] = data['Close'][i]
#setting date as index
new_data.index = new_data.Date
new_data.drop('Date', axis=1, inplace=True)
#creating training and testing sets
dataset = new_data.values


train = dataset[0:987,:]
test = dataset[987:,:]
print(train.shape,test.shape)
test
#converting dataset into a given range
scaler = MinMaxScaler(feature_range=(0, 1))
```

```
scaled_data = scaler.fit_transform(dataset)


print(scaled_data.shape,scaled_data)

scaled_data[60,0]

#converting dataset into x_train and y_train

x_train, y_train = [], []

for i in range(60,len(train)):

    x_train.append(scaled_data[i-60:i,0])

    y_train.append(scaled_data[i,0])

x_train

x_train, y_train = np.array(x_train), np.array(y_train)

x_train = np.reshape(x_train, (x_train.shape[0],x_train.shape[1],1))

# create and fit the LSTM network

model = Sequential

model.add(LSTM(units=50, return_sequences=True, input_shape=(x_train.shape[1],1)))

model.add(LSTM(units=50))

model.add(Dense(1))

model.compile(loss='mean_squared_error', optimizer='adam')

model.fit(x_train, y_train, epochs=5, batch_size=1, verbose=2)

#predicting 246 values, using past 60 from the train data

inputs = new_data[len(new_data) - len(test) - 60:].values

inputs

inputs  = scaler.transform(inputs)

inputs

X_test = []

for i in range(60,inputs.shape[0]):

    X_test.append(inputs[i-60:i,0])

X_test = np.array(X_test)
```

```
X_test

X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))

X_test

closing_price = model.predict(X_test)

closing_price

closing_price = scaler.inverse_transform(closing_price)

closing_price

#tells how much our predicted value is away from line of best fit

rms=np.sqrt(np.mean(np.power((test-closing_price),2)))

rms

#for plotting

trainp = new_data[:987]

testp = new_data[987:]

testp['Predictions'] = closing_price

plt.plot(trainp['Close'])

plt.plot(testp[['Close','Predictions']])
```

Link of project on github- https://github.com/shanky1947/Stock-Market-Prediction/blob/master/stock_newnew.ipynb

Link of project on Google drive (jupyter notebook file)-

https://drive.google.com/open?id=1a7ho75F0R9sSYn_sF3wz41R-KHuRVLiC

CHAPTER-6

CONCLUSION

The prediction appears to be quite good after using LSTM type of neural network. Rms error came to around 9.12 which is descent in case of prediction of a stock market. As stock market varies with other many factors such as political issues, phycological issues, etc. All these aspects combine to make share prices volatile and very difficult to predict with a high degree of accuracy. So, we tried our best to calculate trend of stock market accurately. But further improvement is possible in this domain. Very high accuracy could be achieved by modifying our neural network structure and increasing hidden layers in the code. Person who will be able to predict stock market with high accuracy will be the king of market and he will be the next billionaire. Machine Learning and Artificial Intelligence are taking over the world and are currently the hot topic of discussions. We will be able to change the world once we harnessed full capabilities of machine learning, deep learning, artificial intelligence, etc.

CHAPTER -7

BIBLIOGRAPHY

BOOKS:

1) The Hundred Page Machine Learning Book by Andriy Burkov.

2) Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville 2 – 5th Edition.

WEBSITES:

1) www.google.com
2) www.analyticsvidhya.com
3) www.kaggle.com
4) www.keras.com