A PROJECT REPORT

ON

STOCK PREDICTION ANALYSIS MODEL FOR PREDICTING OUTCOME

Submitted in partial fulfillment for the requirement of the award of

TRAINING

IN

Data Analytics, Machine Learning and AI using Python

DiginiqueTechlabs
Ideal for Geeks

Submitted By

Neavil Porus A (Knowledge Institute of Technology, Salem)

Under the guidance of

Mr. Bipul Kumar Shahi

ACKNOWLEDGEMENT

My sincere gratitude and thanks towards my project paper guide **Bipul Shahi**, Corporate Trainer, Developer, Traveller !!! IOT, Artificial Intelligence, Machine Learning, Robotics, Cloud Computing, Android Apps !!!

It was only with his backing and support that I could complete the report. He provided me all sorts of help and corrected me if ever seemed to make mistakes. I have no such words to express my gratitude. I would like to thank my College Management for allowing me to attend these Intern classes with no Interference from my Special Study Schedules. I acknowledge my dearest parents for being such a nice source of encouragement and moral support that helped me tremendously in this aspect. I would also like to thank my Friends who helped in searching various Datasets for my perusal. I also declare to the best of my knowledge and belief that the Project Work has not been submitted anywhere else.

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.

Problem statement

We’ll dive into the implementation part but first it’s important to establish what we’re aiming to solve. 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.

Technology and Concepts

Machine Learning

Learning algorithms are widely used in Prediction models and data analysis. we are going to have a brief look at basics of machine learning.

Machine Learning Conceptual

Machine learning has emerged as a useful tool for modelling problems that are otherwise difficult to formulate exactly. Classical computer programs are explicitly programmed by hand to perform a task. With machine learning, some portion of the human contribution is replaced by a learning algorithm. As availability of computational capacity and data has increased, machine learning has become more and more practical over the years, to the point of being almost ubiquitous.

It can be used in two ways:

- *Supervised Learning*
- *Unsupervised Learning*

Supervised Machine Learning

We are Going to apply SML to predict the required outcome. This can be done only with the help of data analysis via methods such as LSTM, Linear Regression, K-Means, AutoARIMA and RMS Value inference method. We will see them as the flow moves on.

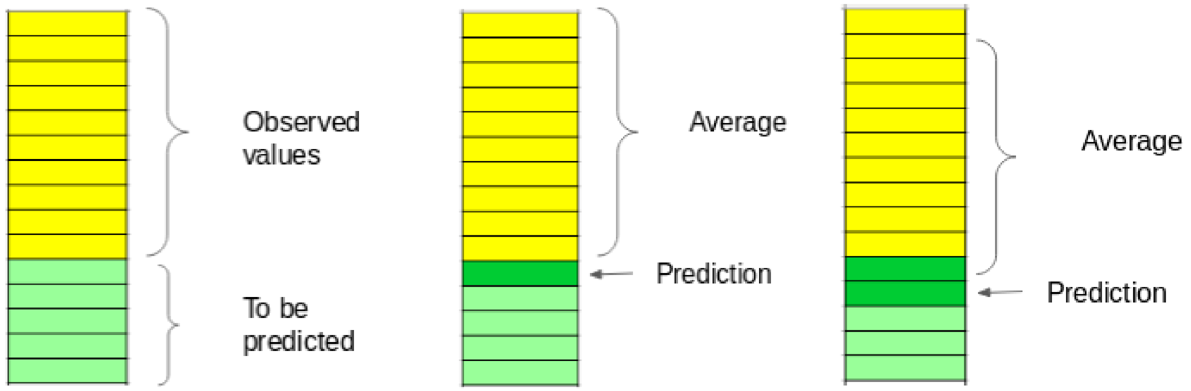
Implementation

Moving Average

‘Average’ is easily one of the most common things we use in our day-to-day lives. For instance, calculating the average marks to determine overall performance, or finding the average temperature of the past few days to get an idea about today’s temperature – these all are routine tasks we do on a regular basis. So this is a good starting point to use on our dataset for making predictions.

The predicted closing price for each day will be the average of a set of previously observed values. Instead of using the simple average, we will be using the moving average technique which uses the latest set of values for each prediction. In other words, for each subsequent step, the predicted values are taken into consideration while removing the oldest observed value from the set. A multi-layer network typically includes three types of layers: an input layer, one or more hidden layers and an output layer. The input layer usually merely passes data along without modifying it. Most of the computation happens in the hidden layers.

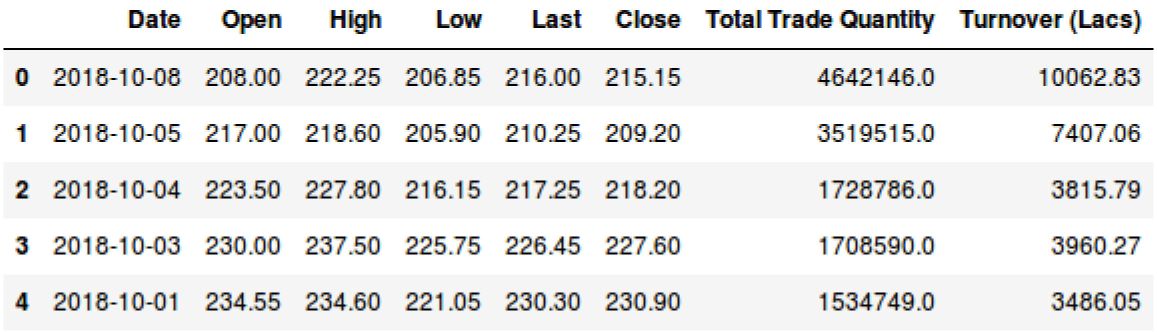
The output layer converts the hidden layer activations to an output, such as a classification. A multilayer feed-forward network with at least one hidden layer can function as a universal approximator.



We will implement this technique on our dataset. The first step is to create a dataframe that contains only the Date and Close price columns, then split it into train and validation sets to verify our predictions.

Prediction Model without any Reference

We will first load the dataset and define the target variable for the problem. This will give us an major understanding over how the datasets are being used to extract features to produced desirable outcome.



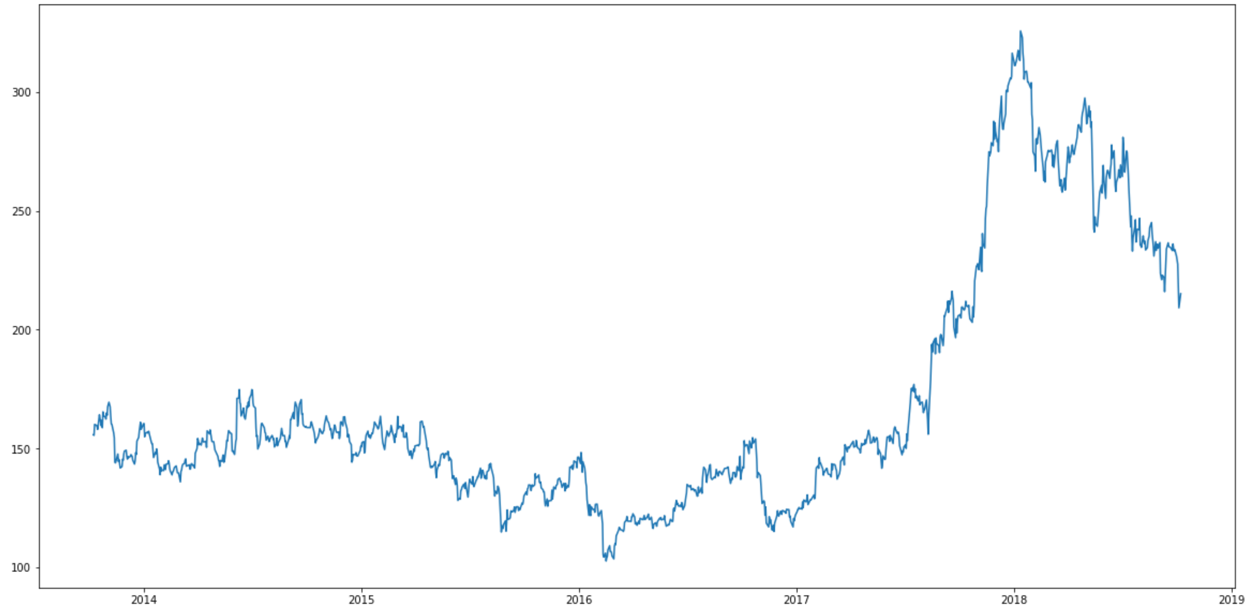
	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-10-08	208.00	222.25	206.85	216.00	215.15	4642146.0	10062.83
1	2018-10-05	217.00	218.60	205.90	210.25	209.20	3519515.0	7407.06
2	2018-10-04	223.50	227.80	216.15	217.25	218.20	1728786.0	3815.79
3	2018-10-03	230.00	237.50	225.75	226.45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

Then we can use these tabulated values to determine over the Predicted Stock Market Graph which can be used to analyze the annual turnover of the company.

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.

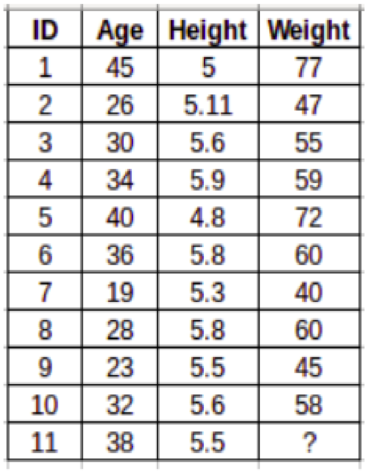
Total Trade Quantity is the number of shares bought or sold in the day and Turnover (Lacs) is the turnover of the particular company on a given date. Another important thing to note is that the market is closed on weekends and public holidays.

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. The Plotting can be done as:

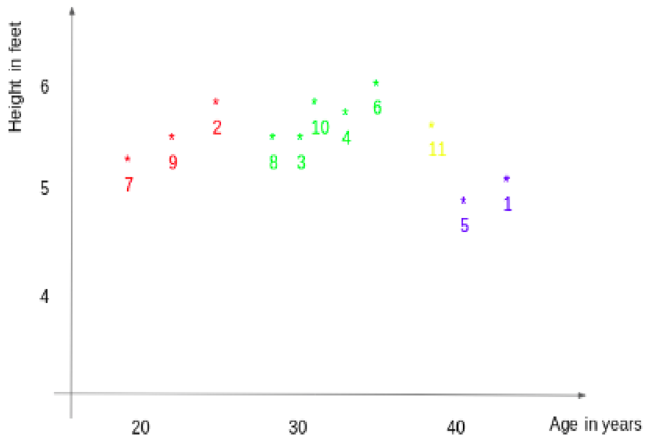


This Model can be taken as a reference for the above datasets.

Implementation



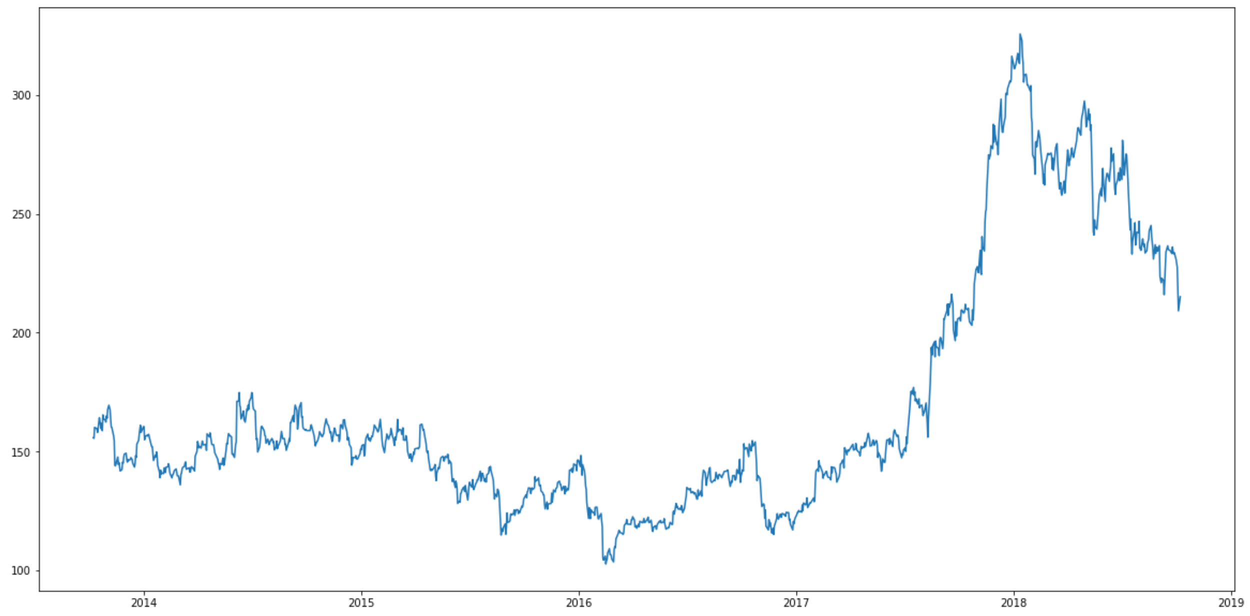
ID	Age	Height	Weight
1	45	5	77
2	26	5.11	47
3	30	5.6	55
4	34	5.9	59
5	40	4.8	72
6	36	5.8	60
7	19	5.3	40
8	28	5.8	60
9	23	5.5	45
10	32	5.6	58
11	38	5.5	?



The Above diagram can be used to create a sample model with the help of RMSE Value. This method can be used to determine the data used in making a prediction.

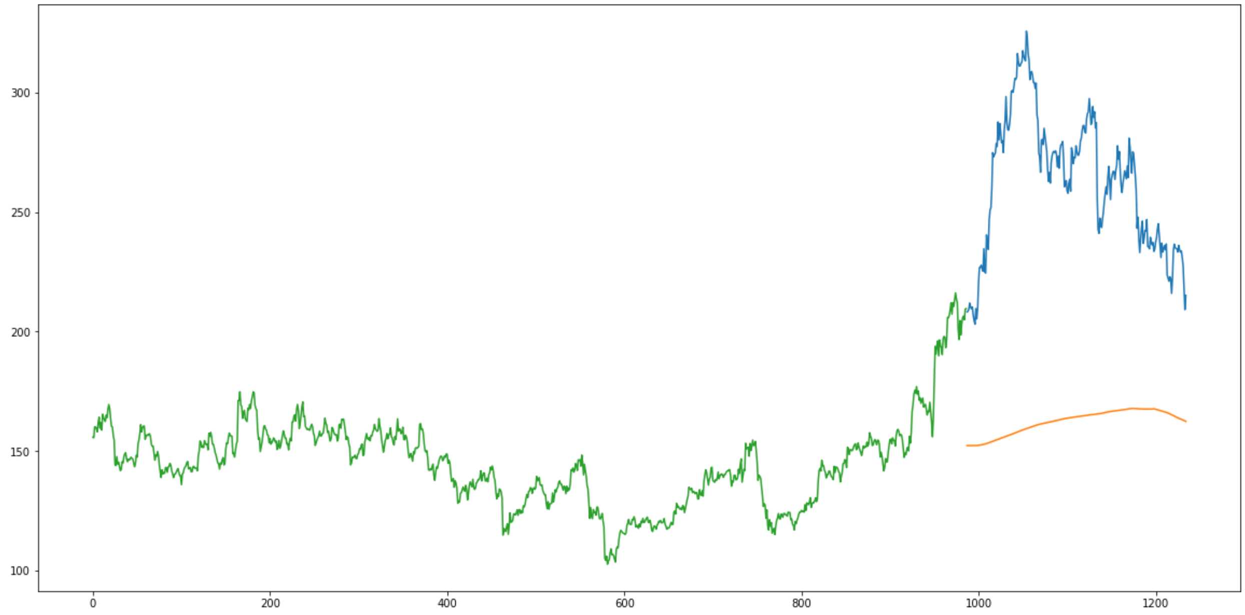
Processing via RMSE Method

This Method Can be used to plot a data graph which can be used to analyze the given data and process an output based on the past data feeding. Just checking the RMSE does not help us in understanding how the model performed. This is the graph before the RMSE Processing.



Untrained data Model

Let’s visualize this to get a more intuitive understanding. So here is a plot of the predicted values along with the actual values. Which would make us understand the wide difference between trained and untrained models. By applying the required data we have produced this outcome.



Thus the RMSE value is predicted by this given input.

Drawbacks

The RMSE value is close to 105 but the results are not very promising (as you can gather from the plot). The predicted values are of the same range as the observed values in the train set (there is an increasing trend initially and then a slow decrease). Thus We can move to other methods like Linear Regression, K-Means and LSTM Method. As we already know that Linear Regression and K-Means can’t be used to handle and produce desired value when we consider such a magnanimous amount of data, we can apply Long Short Term Memory(LSTM) To predict our desired outcome.

LSTM (LONG SHORT TERM MEMORY)

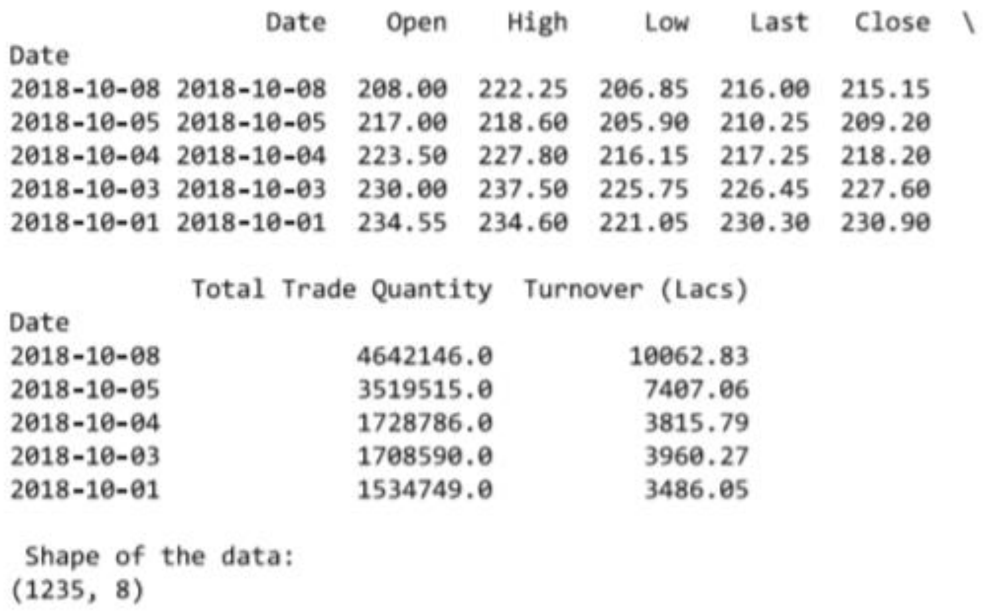
LSTMs are widely used for sequence prediction problems and have proven to be extremely effective. The reason they work so well is because LSTM is able to store past information that is important, and forget the information that is not. LSTM has three gates:

The input gate: The input gate adds information to the cell state.
The forget gate: It removes the information that is no longer required by the model.
The output gate: Output Gate at LSTM selects the information to be shown as output.

EXECUTING LSTM IN MODEL

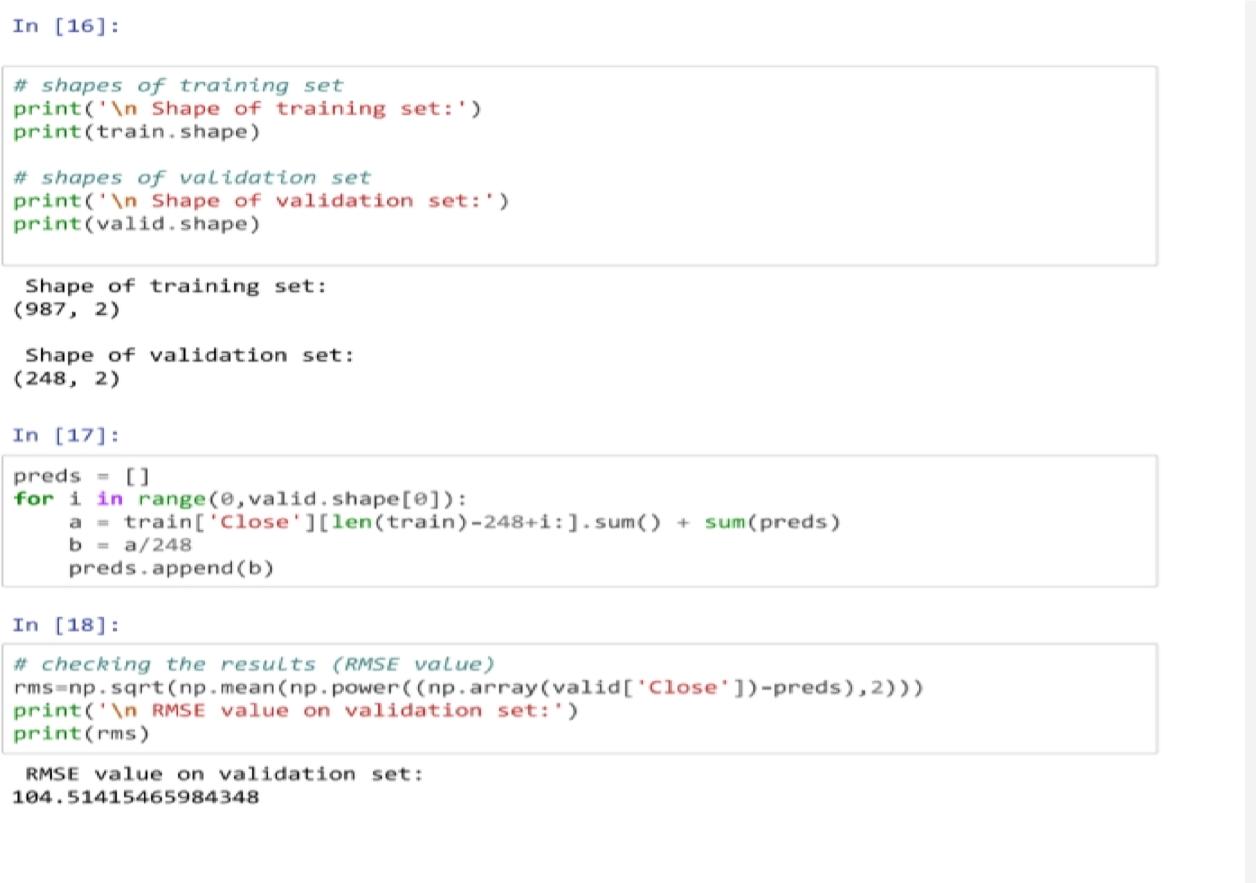
The model can be trained in Keras Framework which can be used to visualize our ideology and produce practical outcome.

It can be pictorialized as:



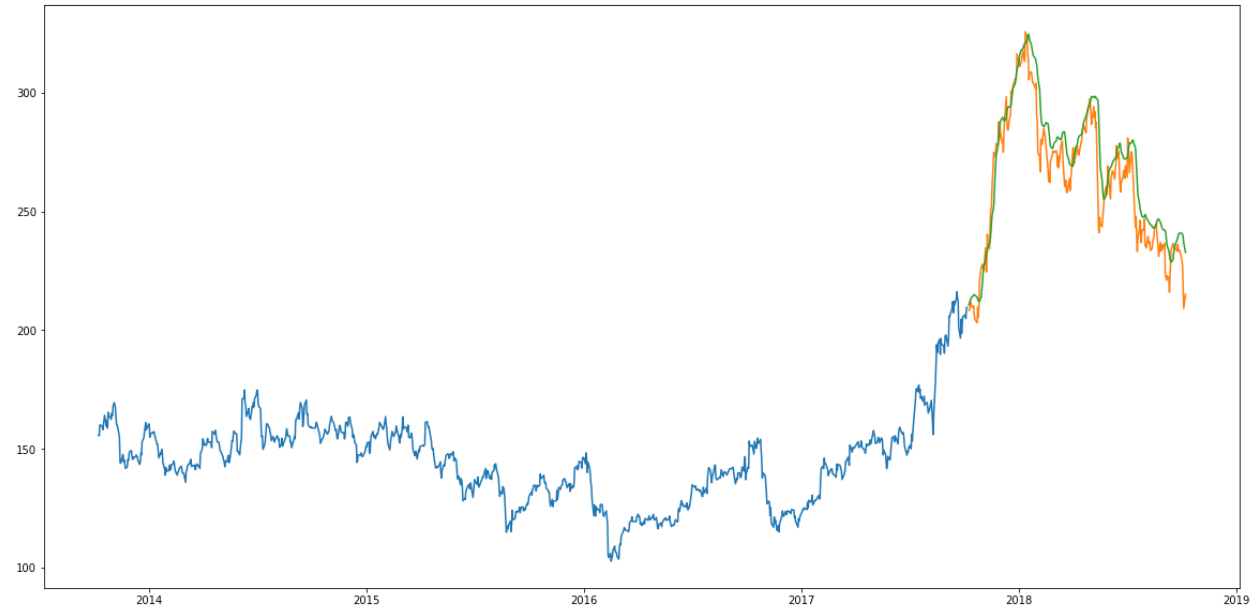
```
                  Date    Open    High     Low    Last   Close  \
Date
2018-10-08 2018-10-08  208.00  222.25  206.85  216.00  215.15
2018-10-05 2018-10-05  217.00  218.60  205.90  210.25  209.20
2018-10-04 2018-10-04  223.50  227.80  216.15  217.25  218.20
2018-10-03 2018-10-03  230.00  237.50  225.75  226.45  227.60
2018-10-01 2018-10-01  234.55  234.60  221.05  230.30  230.90

            Total Trade Quantity  Turnover (Lacs)
Date
2018-10-08             4642146.0         10062.83
2018-10-05             3519515.0          7407.06
2018-10-04             1728786.0          3815.79
2018-10-03             1708590.0          3960.27
2018-10-01             1534749.0          3486.05

 Shape of the data:
(1235, 8)
```

With this, we can use the platform to train the datasets.



```
In [16]:

# shapes of training set
print('\n Shape of training set:')
print(train.shape)

# shapes of validation set
print('\n Shape of validation set:')
print(valid.shape)

 Shape of training set:
(987, 2)

 Shape of validation set:
(248, 2)

In [17]:

preds = []
for i in range(0,valid.shape[0]):
    a = train['Close'][len(train)-248+i:].sum() + sum(preds)
    b = a/248
    preds.append(b)

In [18]:

# checking the results (RMSE value)
rms=np.sqrt(np.mean(np.power((np.array(valid['Close'])-preds),2)))
print('\n RMSE value on validation set:')
print(rms)

 RMSE value on validation set:
104.51415465984348
```

Then after training the pretrained model, we can produce the desired outcome graph.



The Predicted model is near accurate to that of the Correct Values.

Difficulties:

These Models require high framework and configurations to give accesss to these datasets. This can also be a huge drawback if the user doesn't have the required elements. The correct measuring of data and applying the correct method is must to reduce the total time take to produce the model.

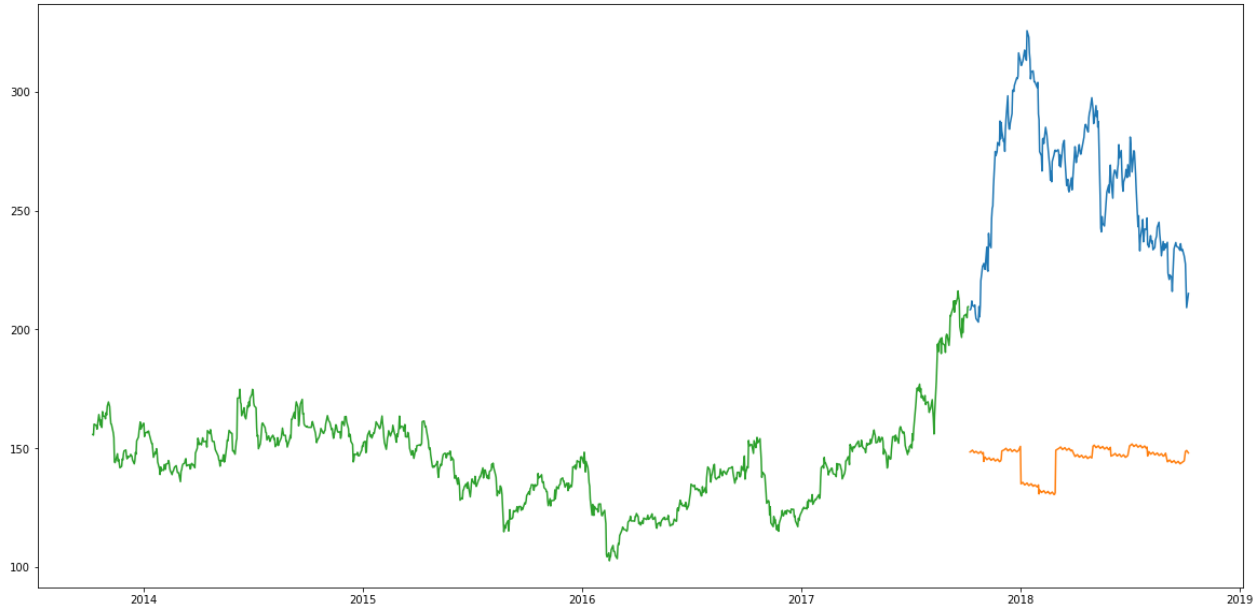
Inference

The LSTM model can be tuned for various parameters such as changing the number of LSTM layers, adding dropout value or increasing the number of epochs. But are the predictions from LSTM enough to identify whether the stock price will increase or decrease? Certainly not!

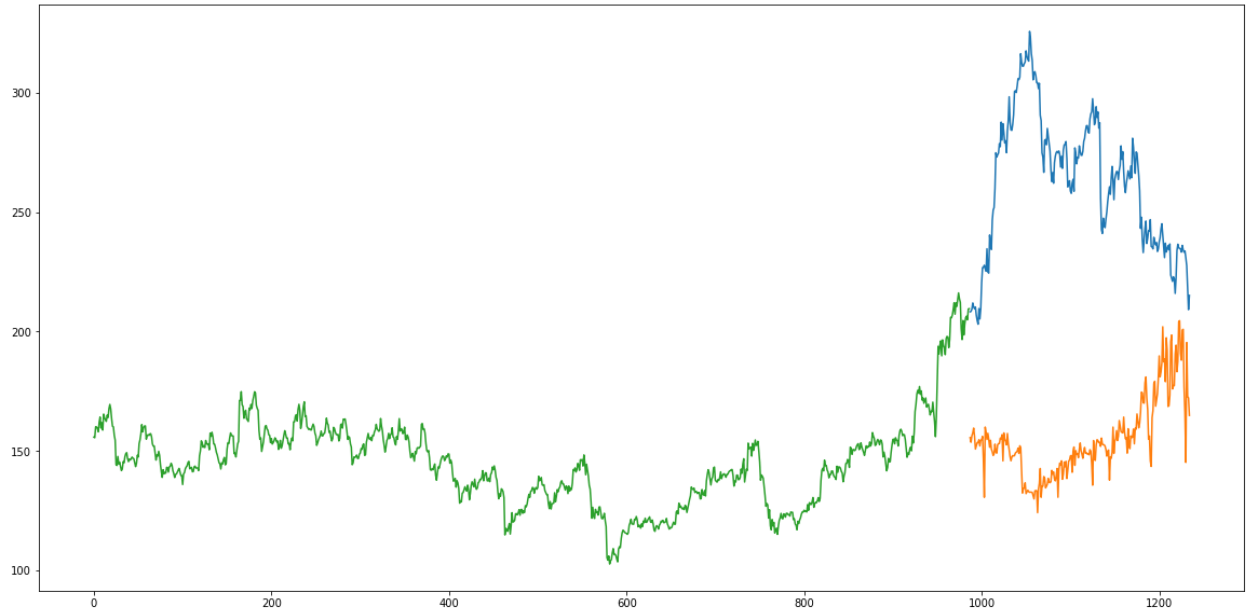
Stock price is affected by the news about the company and other factors like demonetization or merger/demerger of the companies. There are certain intangible factors as well which can often be impossible to predict beforehand.

OTHER TRAINED REFERENCE MODELS:

To Produce the desired outcome, several methods such as Linear Regression, K-Means and other prediction models were also used, but all in vain. They all produced several outcomes but they weren’t even close when compared to my LSTM model. The pictures are furnished for kind reference.



Linear Regression graph for the pretrained model.



K-Means Predicted outcome to my Pretrained model

Since these models couldn't produce the desired outcome, I moved on to LSTM Technique which literally produced the desired outcome.

Conclusions

Time series forecasting is a very intriguing field to work with, as I have realized during my time doing these projects. There is a perception in the community that it's a complex field, and while there is a grain of truth in there, it's not so difficult once you get the hang of the basic techniques. we have worked with historical data about the stock prices of a publicly listed company. We have implemented a mix of machine learning algorithms to predict the future stock price of this company, starting with simple algorithms like averaging and linear regression, and then move on to advanced techniques like LSTM. The project has produced outcome with reference to the above mentioned with accuracy as follows. Linear Regression (0-10%), K-Means(0-10%), RMSE(15-25%) and then the last used LSTM has produced (89.5 – 96.3%) desired outcome.

However, the model has desired outcome with respect to the used datasets, the outcome may vary when new datasets are used to forecast/predict the outcome. The LSTM model can be tuned for various parameters such as changing the number of LSTM layers, adding dropout value or increasing the number of epochs. But are the predictions from LSTM enough to identify whether the stock price will increase or decrease. But that isn't certain as far as the complexity of the algorithm evolves.

Bibliography

- https://www.kaggle.com
- https://www.analyticsvidhya.com
- https://www.medium.com
- https://www.mlindia.com
- https://www.kaggle.com/tatadatasets/newref/#02jwrw0/source.jpg