STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

Pfizer Stock Price Prediction using LSTM

By

Sushant R Ghorpade

FE 511- Intro to Bloomberg and Thomson Reuters

Instructor: Agathe Sadeghi

Table of Contents

Table of Figures

Introduction

Stock market is characterized as dynamic, unpredictable and non-linear in nature. Predicting stock prices is a challenging task as it depends on various factors including but not limited to political conditions, global economy, company's financial reports and performance etc. Thus, to maximize the profit and minimize the losses, techniques to predict values of the stock in advance by analyzing the trend over the last few years, could prove to be highly useful for making stock market movements [1]. The stock's price only tells you a company's current value or its market value. So, the price represents how much the stock trades at or the price agreed upon by a buyer and a seller. If there are more buyers than sellers, the stock's price will climb. If there are more sellers than buyers, the price will drop [2].

Traditionally, two main approaches have been proposed for predicting the stock price of an organization. Technical analysis method uses historical price of stocks like closing and opening price, volume traded, adjacent close values etc. of the stock for predicting the future price of the stock. The second type of analysis is qualitative, which is performed based on external factors like company profile, market situation, political and economic factors, textual information in the form of financial new articles, social media and even blogs by economic analyst [1].

Why use Machine Learning

Advanced intelligent techniques based on either technical or fundamental analysis are used for predicting stock prices. Particularly, for stock market analysis, the data size is huge and non-linear. To deal with this variety of data efficient model is needed that can identify the hidden patterns and complex relations in this large data set. Machine learning techniques in this area have proved to improve efficiencies by 60-86 percent as compared to the past methods [1]

Using Long short-term memory (LSTM)

LSTMs are very powerful in sequence prediction problems because they can store past information. This is important in our case because the previous price of a stock is crucial in predicting its future price

Long short-term memory (**LSTM**) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (intrusion detection systems) [3].

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.

Data Understanding

For our project we are going to predict stock price for Pfizer last 5-year data which is extracted from Bloomberg terminal using the historical price "**HP**" function and Excel-Add in. The data

consist of three columns i.e., Date, Last price and volume and 1521 observations. The ‘Px_LAST’ which is the close/ last price is the target variable.

Below figures show the procedure of data extraction:

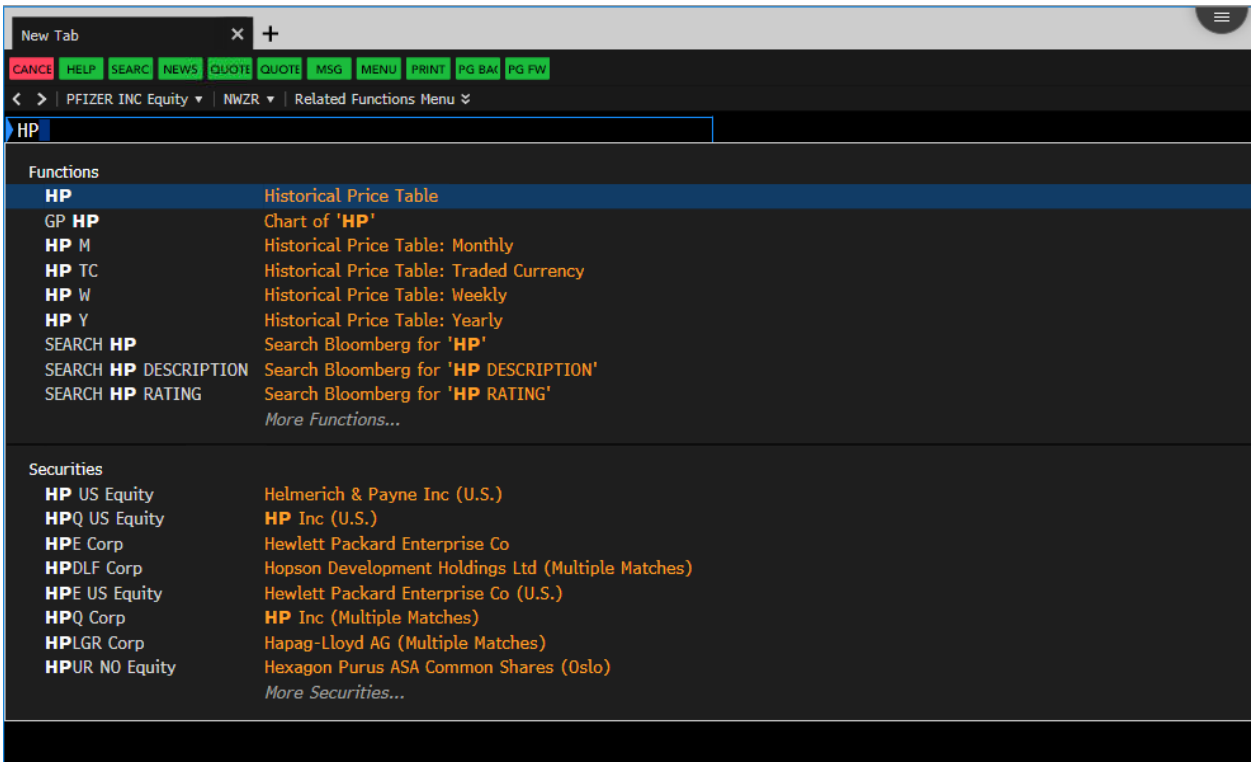
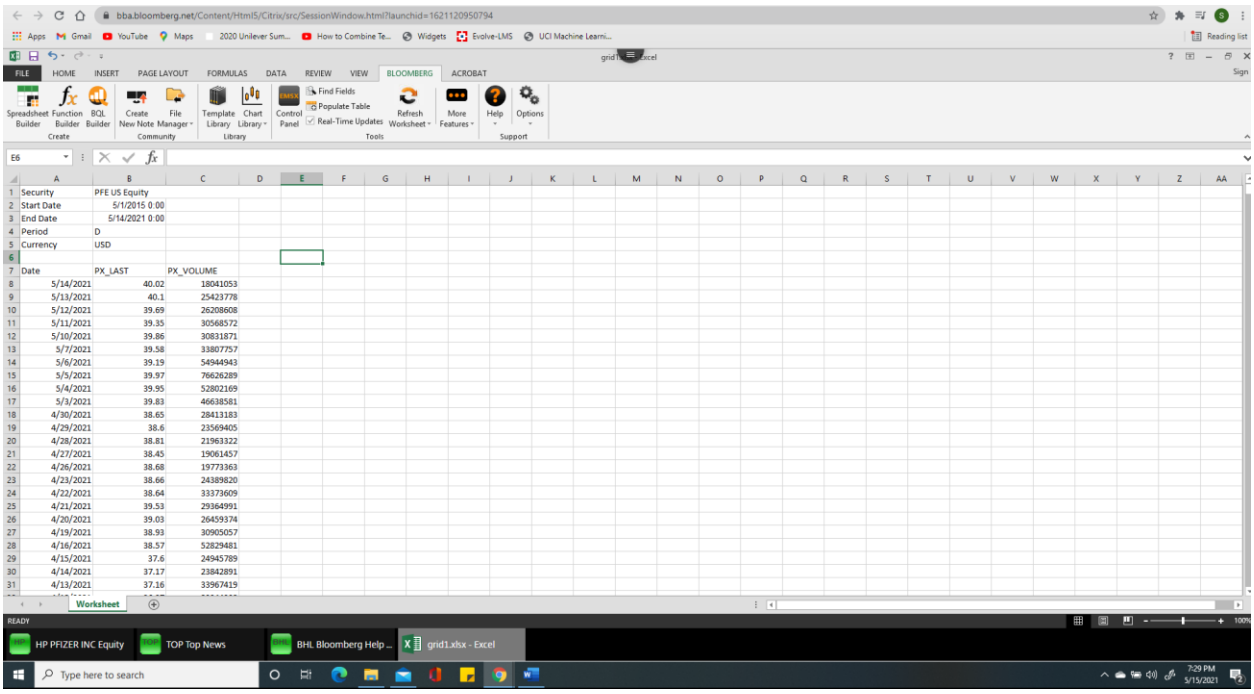


Figure 1.Using Historical Pricing Table

Figure 2.Data Table in Bloomberg interface



Security	PFE US Equity	
Start Date	5/1/2015 0:00	
End Date	5/14/2021 0:00	
Period	D	
Currency	USD	
Date	PX_LAST	PX_VOLUME
5/14/2021	40.02	18041053
5/13/2021	40.1	25423778
5/12/2021	39.69	26208608
5/11/2021	39.35	30568572
5/10/2021	39.86	30831871
5/7/2021	39.58	33807757
5/6/2021	39.19	54944943
5/5/2021	39.97	76626289
5/4/2021	39.55	52802169
5/3/2021	39.83	46638581
4/30/2021	38.65	28413183
4/29/2021	38.6	23569405
4/28/2021	38.81	21963322
4/27/2021	38.45	19061457
4/26/2021	38.68	19773363
4/23/2021	38.66	24389820
4/22/2021	38.64	33373609
4/21/2021	39.53	29364991
4/20/2021	39.03	26459374
4/19/2021	38.93	30905057
4/16/2021	38.57	52829481
4/15/2021	37.6	24945789
4/14/2021	37.17	23842891
4/13/2021	37.16	33967419

Figure 3.In Excel-Add in

To get more insights on our data we will import some libraries like pandas, NumPy, seaborn and matplotlib

Figure below show the trend in our closing values. There can be seen an upward trend increasing and decreasing spikes. To get view, we must look at the statistical summary of it.

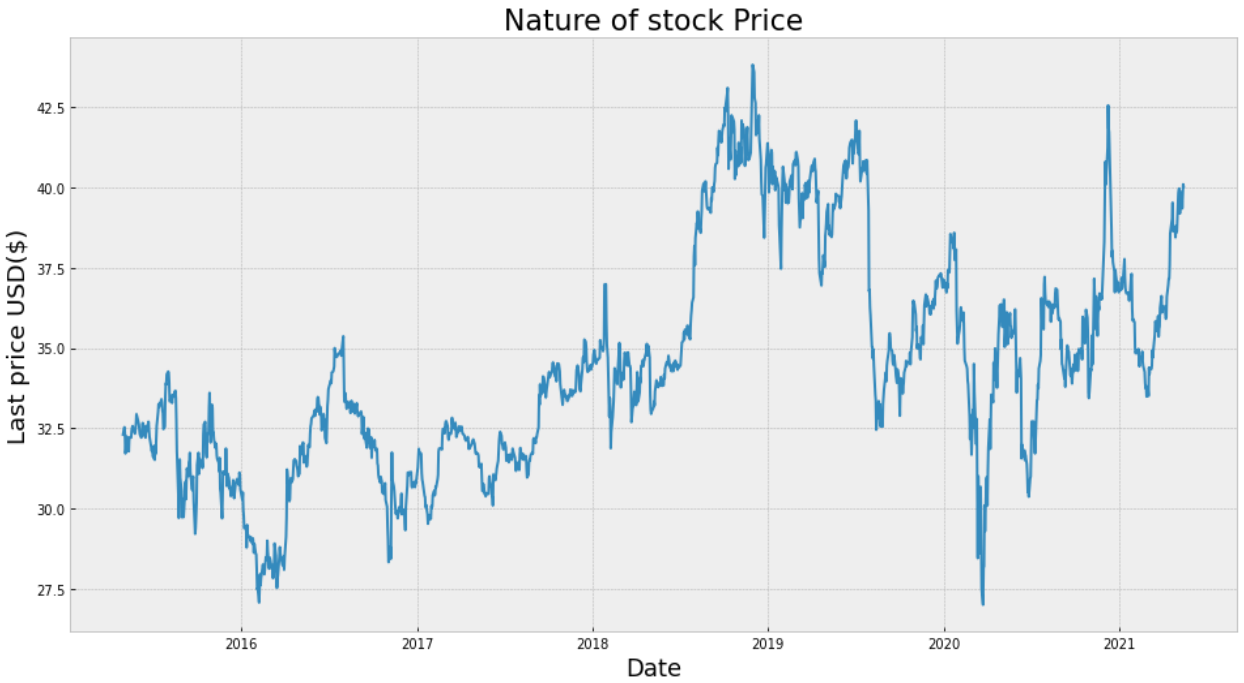


Figure 4.Nature of Stocks

Below are the statistical insights of the target variable:

	Mean	Std	Min	25%	50%	75%	max
Values	34.46	3.48	27	31.87	34.06	36.39	43.82

By looking at the table we can say that the maximum value is beyond the third quartile which indicates the presence of outliers in data. We use a boxplot shown in the figure below to visualize and identify number of outliers.

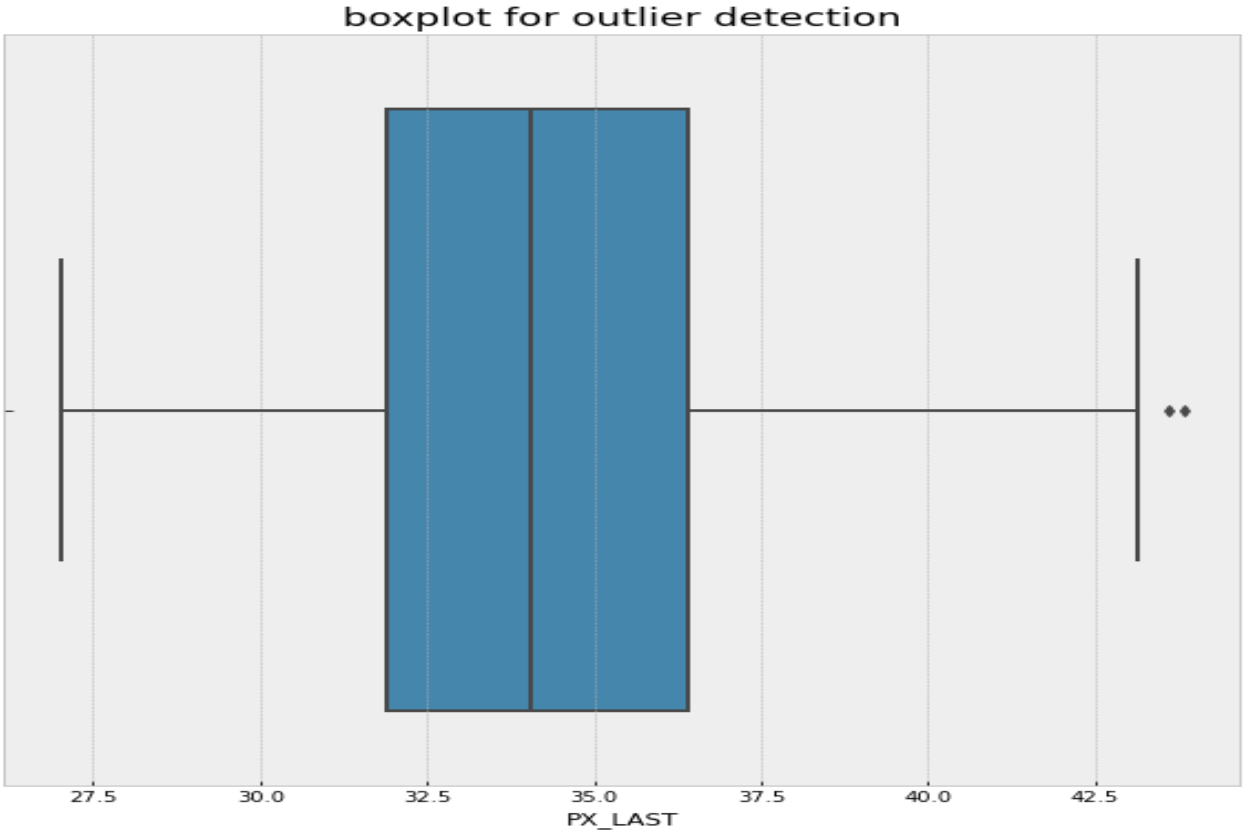


Figure 5.Boxplot for Outlier Detection

The shape of distribution is described by number of peaks and by its possession of symmetry, its tendency to skew, or its uniformity.

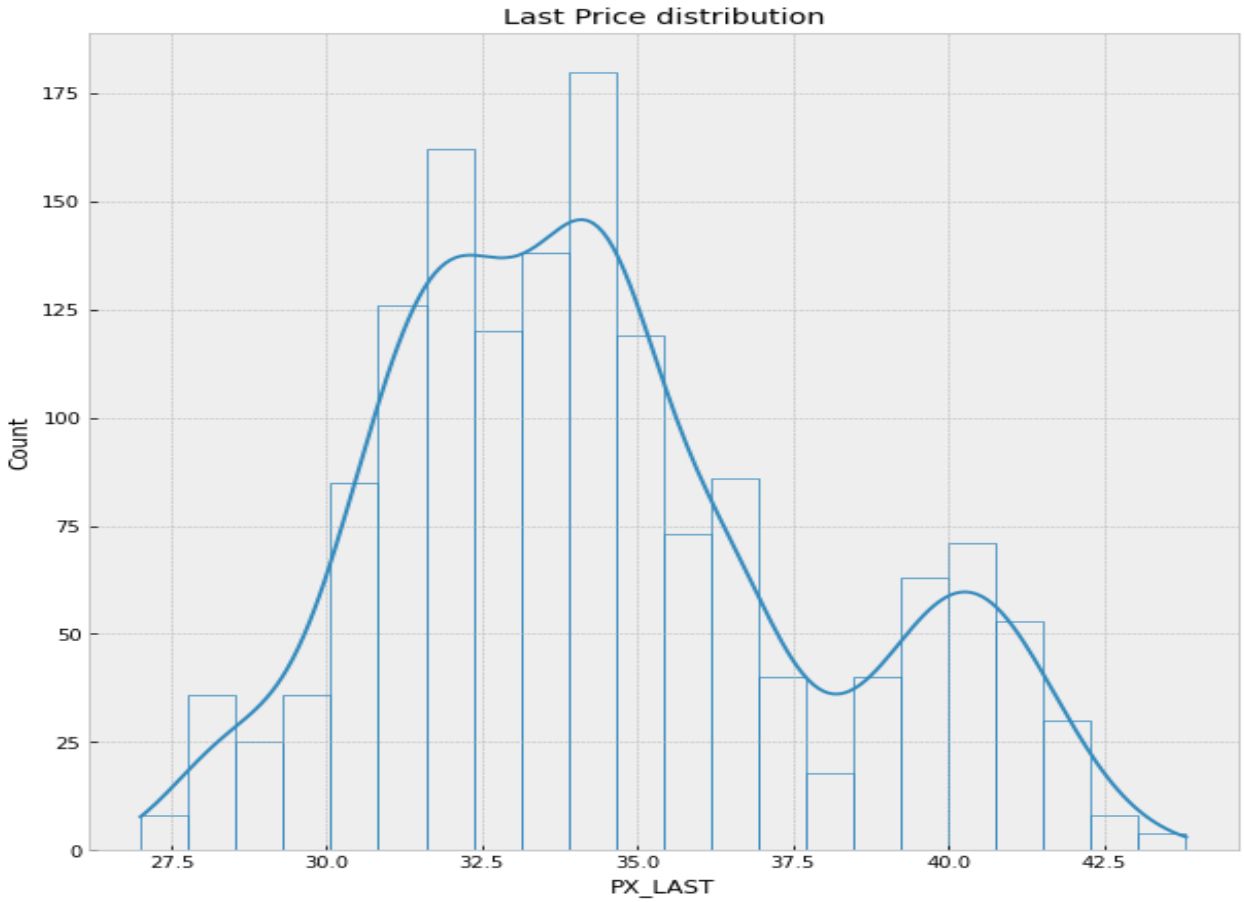


Figure 6.Distribution of Last price values

Data preparation

For forecasting future values, we need to prepare data accordingly. Firstly, we set our dates as index values as future values depend on previous values this would be useful in splitting train and test values. Second is to filter data by only keeping our predicting variables and create an array with the values of last price.

Next step is to normalize the data. the goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. For this we use MinMaxScalar to scale data between 0 to 1.

Once done, its time to split data for training our model. For our model are going to make prediction for next 30 day as financial month consist of 30 days. We will split 80 percent of data for training and remaining 20 percent for testing and validation.

Apart from that it is also important to check whether the time series is stationary or non-stationary. To test it we have used Dickey- Fuller test [4]. For this we will use the adfuller module form statsmodels. The results of test were “Failed to Reject null hypothesis - Time Series in Non-Stationary” i.e., of data is not stationary. If we fail to reject the null hypothesis, we can say that the series is non-stationary. This means that the series can be linear or difference stationary [5] this means value are self-dependent and resemble random walk, which is common with financial data. It can be treated by transforming using differencing.

Modeling

A machine learning model is a file that has been trained to recognize certain types of patterns. You train a model over a set of data, providing it an algorithm that it can use to reason over and learn from those data [6]. Once you have trained the model, you can use it to reason over data that it has not seen before and make predictions about those data [6].

For our predictive model, we are implementing LSTM. LSTM is a type of recurrent neural network capable of learning dependence in sequential predicting problems, which is suitable since data is not stationary and is independent.

For this we have imported Sequential [7], Dense [8], Dropout [9], LSTM from module model and layers in keras, respectively. We build a layer model which reduces in size to one and use dropout reduces chances of over fitting. Once done we need to compile our model. We have implemented **adam** optimizer and '**mean squared error'** as our loss function.

Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models. **Adam** combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems [10].

MSE is **used** to check how close estimates or forecasts are to actual values. Lower the **MSE**, the closer is forecast too actual. This is **used** as a model evaluation measure for regression models and the lower value indicates a better fit [11].

Once all the above steps are done, we are ready to fit our model. We are setting batch size to 50 and epochs to 100. the number of training examples in one forward/backward pass. The higher the batch size, the more memory space you will need. While one epoch indicates one forward pass and one backward pass of all the training examples.

Evaluation

Once we fit the model, we need to evaluate it. For this we have considered two main parameters i.e., Root mean squared error (RMSE), MAE and R Squared value.

Since the errors are squared before they are averaged, the **RMSE** gives a relatively high weight to large errors. This means the **RMSE** is most useful when large errors are particularly undesirable. Both the MAE and **RMSE** can range from 0 to ∞. They are negatively oriented scores: Lower values are better [12].

While R Squared is used to explain the variance in our model. R Squared is secondary stastical measure for our predictive model as values in time series are interdependent on previous values which make the highly correlated.

The below figure show distribution of our predictive value. The spread resembles narrow normal distribution, which indicate that the standard deviation is small, and values are around the mean.

Also, the area under the normal distribution curve represents probability and the total area under the curve sums to one. Most of the continuous data values in a normal distribution tend to cluster around the mean, and the further a value is from the mean, the less likely it is to occur [13].

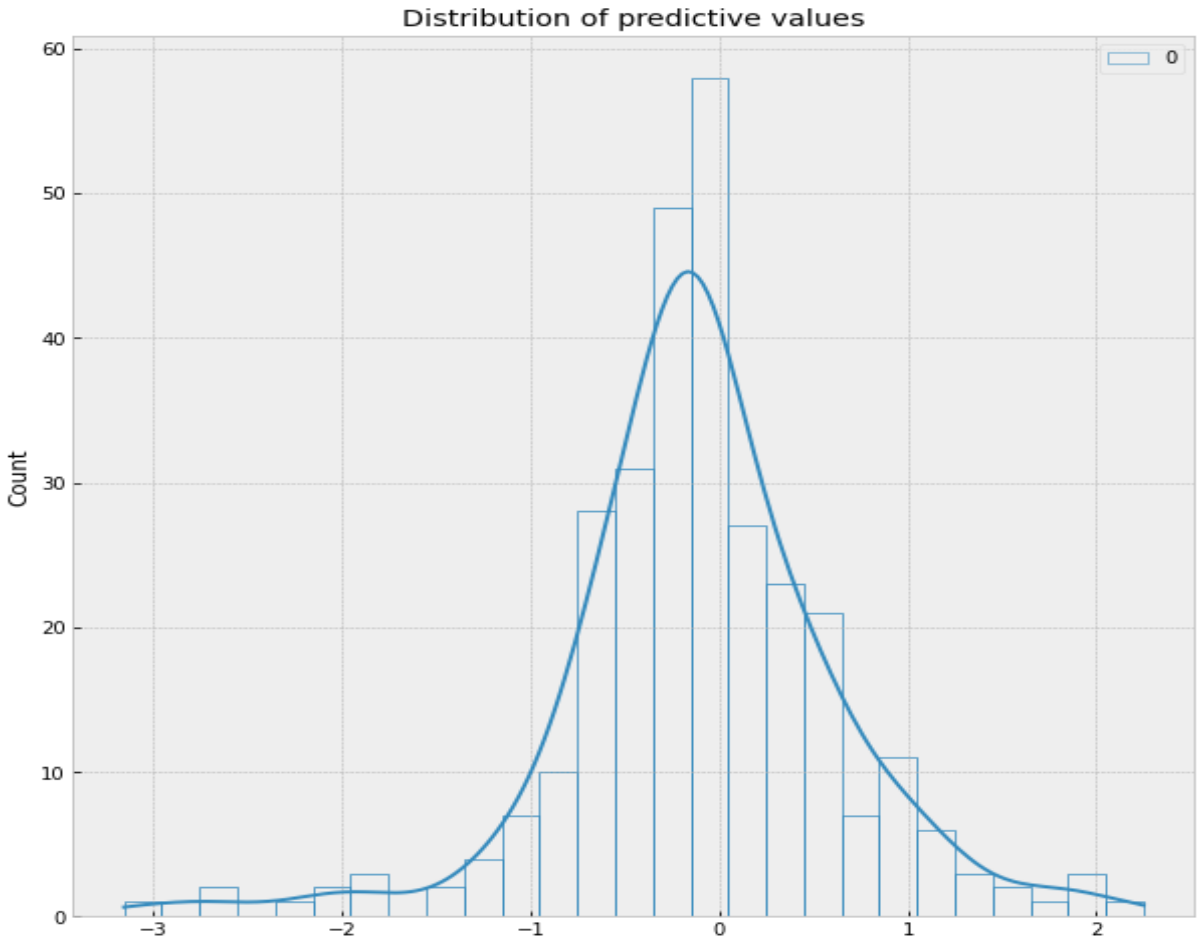


Figure 7.Distribution of predictive values

To get a better perspective we can have look at the below graph. We see that the values are closely clustered around the straight line. The line is sketched with true values which also indicated how closely are our predicted to real ones. The points away from the line are the same values with larger standard deviation in the distribution curve.

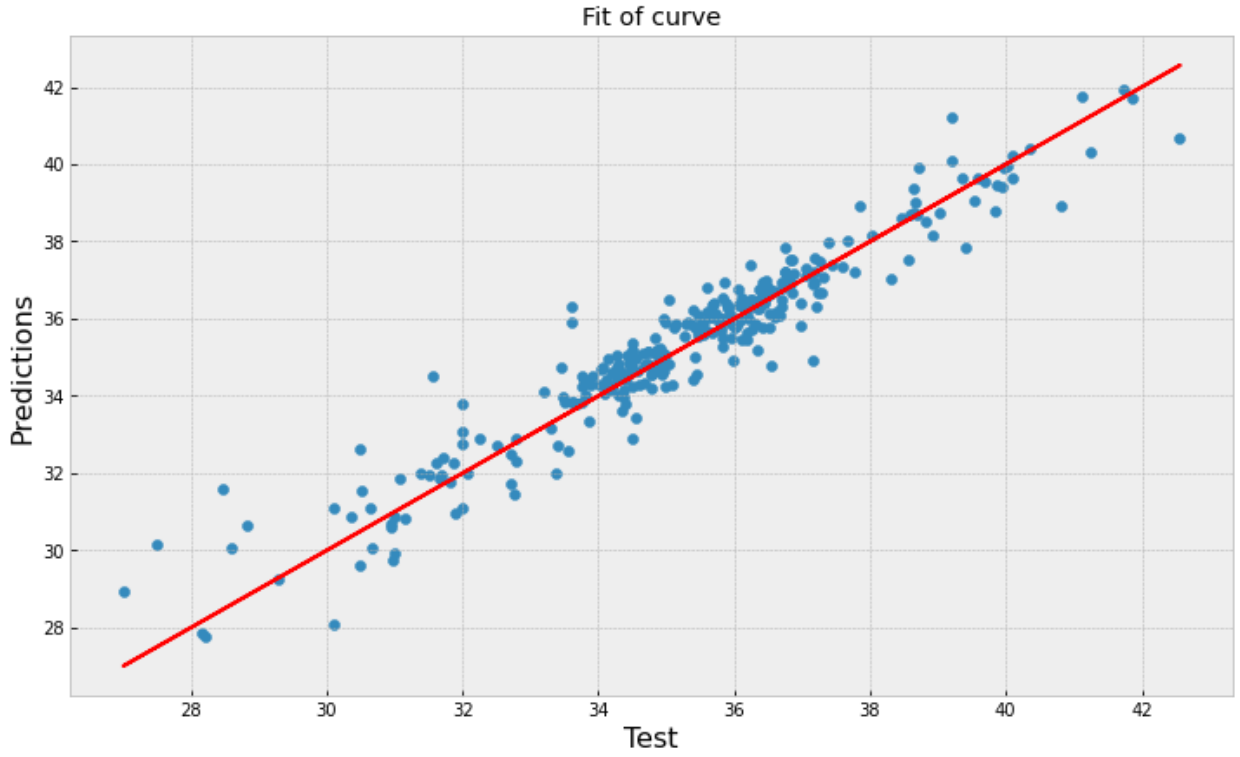


Figure 8.Fit of Curve

To sum up, we have plotted our values with real time data and the output can be seen in below figure.

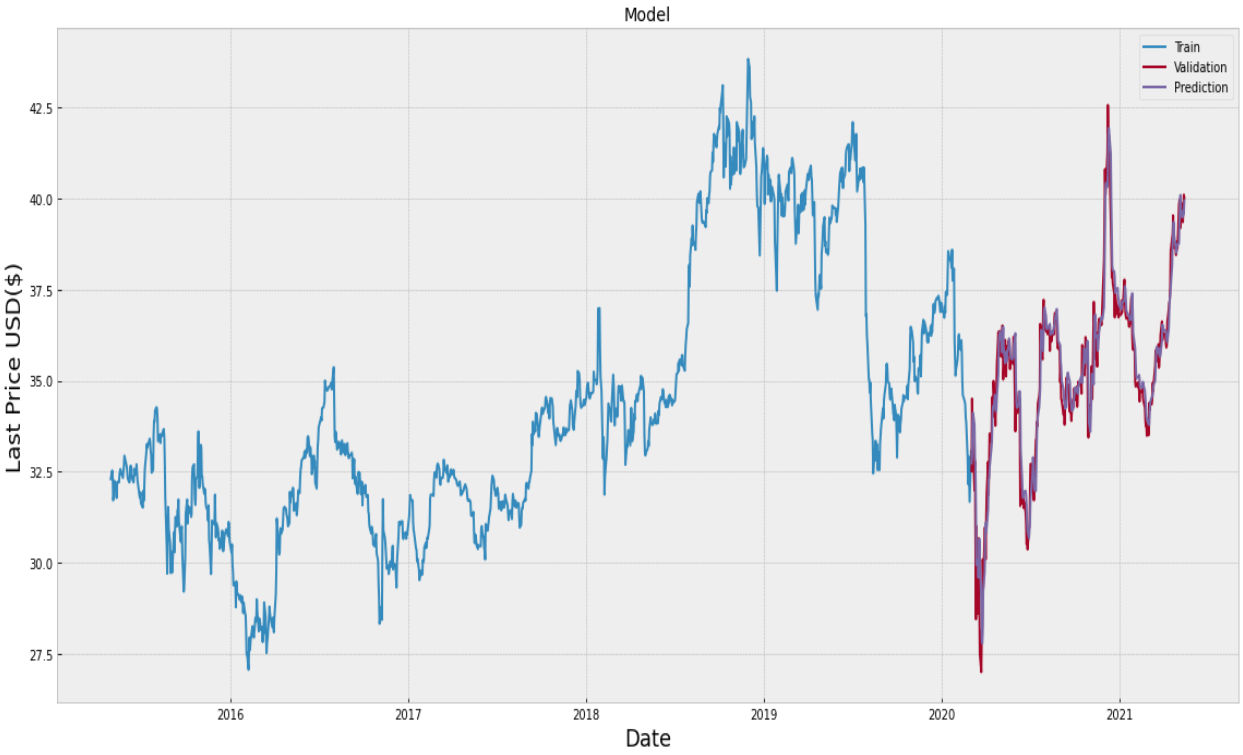


Figure 9.Predictive Model

The model has performed really good and below can be seethe overview of our predictions compared to real values:

Date	PX_LAST	Prediction
2020-03-03	32.5146	32.877579
2020-03-04	34.5053	32.759075
2020-03-05	33.6143	34.217144
2020-03-06	33.1972	33.901833
2020-03-09	32.0027	33.328110
...	...	...
2021-05-10	39.8600	39.558048
2021-05-11	39.3500	39.924683
2021-05-12	39.6900	39.536404
2021-05-13	40.1000	39.718517
2021-05-14	40.0200	40.136860

Summary Statistics:

1. Test for Stationarity(ADF test):
    a. P-value: 0.122890
    b. Critical Values:
        i. 1%: -3.435
        ii. 5%: -2.568
        iii. 10%: -2.568

    Failed to Reject null Hypothesis(Ho) – Time Series in Non-Stationary

2. Root Mean Squared Error (RMSE): 0.06626
3. Mean Absolute Error (MAE): 0.4767
4. R Squared: 0.92

From Statistics we can conclude that 92 percent of variation are explained by our model and small RMSE indicated lower noise and better predictive values.

References

[1] D. C. A. T. K. Mehar Vijha, "Stock Closing Price Prediction using Machine Learning Techniques," *International Conference on Computational Intelligence and Data Science (ICCIDS 2019),* 2019.

[2] "https://www.investopedia.com/," [Online]. Available: https://www.investopedia.com/articles/stocks/08/stock-prices-fool.asp.

[3] L. s.-t. memory, "wikipedia.org," [Online]. Available: https://en.wikipedia.org/wiki/Long_short-term_memory.

[4] Wikipedia, "Dickey–Fuller test," [Online]. Available: https://en.wikipedia.org/wiki/Dickey%E2%80%93Fuller_test.

[5] A. SINGH, "A Gentle Introduction to Handling a Non-Stationary Time Series in Python," [Online]. Available: https://www.analyticsvidhya.com/blog/2018/09/non-stationary-time-series-python/#:~:text=If%20we%20fail%20to%20reject,stationary%20in%20the%20next%20section)..

[6] Microsoft, "What is a machine learning model?," [Online]. Available: https://docs.microsoft.com/en-us/windows/ai/windows-ml/what-is-a-machine-learning-model#:~:text=A%20machine%20learning%20model%20is,and%20learn%20from%20those%20data..

[7] keras.io, "The Sequential class," [Online]. Available: https://keras.io/api/models/sequential/.

[8] keras.io, "Dense layer," [Online]. Available: https://keras.io/api/layers/core_layers/dense/.

[9] keras.io, "Dropout layer," [Online]. Available: https://keras.io/api/layers/regularization_layers/dropout/.

[10] "Machine Learning Mastery," 13 January 2021. [Online]. Available: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

[11] G. L. Team, "Mean Squared Error – Explained | What is Mean Square Error?," 8 August 2020. [Online]. Available: https://www.mygreatlearning.com/blog/mean-square-error-explained/.

[12] "Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE)," [Online]. Available: http://www.eumetrain.org/data/4/451/english/msg/ver_cont_var/uos3/uos3_ko1.htm.

[13] Dr. Saul McLeod, "Introduction to the Normal Distribution (Bell Curve)," 2019. [Online]. Available: https://www.simplypsychology.org/normal-distribution.html.