

Stock Data Analysis

Vivek Mondal

Data Used: <https://drive.google.com/file/d/1ZHMASRNKGzfCYd6YMBJLq16Ql26wK1pV/view?usp=sharing>

Tool used: Anaconda Jupyter Notebook.

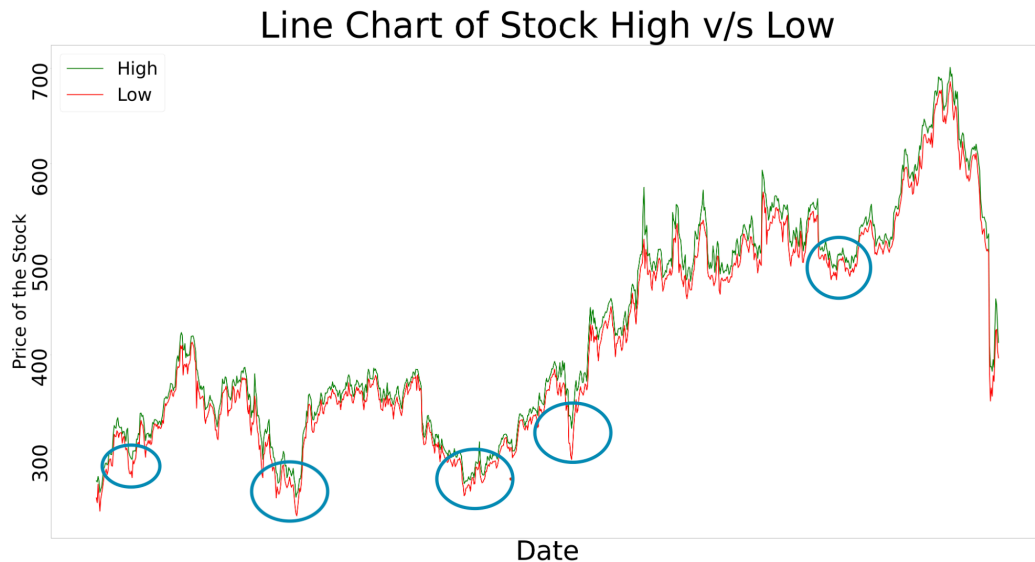
Language used: Python.

Library used: Pandas, Numpy, Matplotlib, Seaborn.

The dataset has been imported using Pandas Dataset, the dataset has a total of 1009 rows and 7 columns and contains date, opening amount of the stock as “Open”, high value of the stock as “High”, low value of the stock as “Low”, closing amount of the stock as “Close”, adjusted closing value as “Adj Close”, and the available stock volume as “Volume”, and the dataset had no Null values . By analyzing the data I have noticed that, the starting stock value was 233.92 and from that, within 2 years and 8 months, it became 692.25, so we can clearly see that this stock gives a good return within 3 years. And in these years the average of the stock is also 419 which is also a good thing.

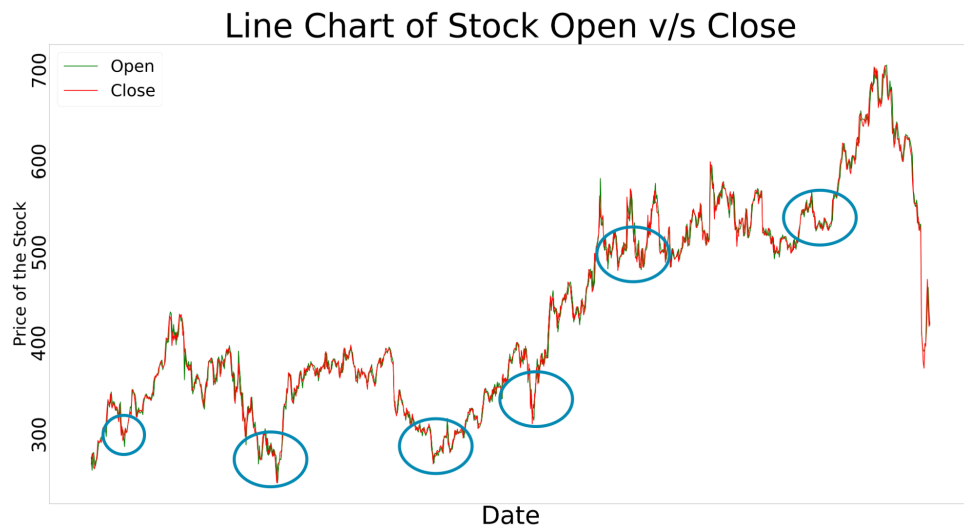
I have added a ‘Profit_Loss_Amount’ and a ‘Profit/Loss’ column in the dataset, to determine the profit and loss.

I have plotted a line chart using the library 'Matplotlib' to analyse the graph between the high and low value of the stock.



By observing the tendency of the graph, I have noticed that after a certain level of decline, the graph started to go upwards, and the difference between the high and value is not too much high. So the customer can have less risk of loss, and both high and low price has increased over the year.

After High v/s Low graph I have plotted a line chart using the library 'Matplotlib' to analyse the graph between the Opening and Closing value of the stock.



Caption

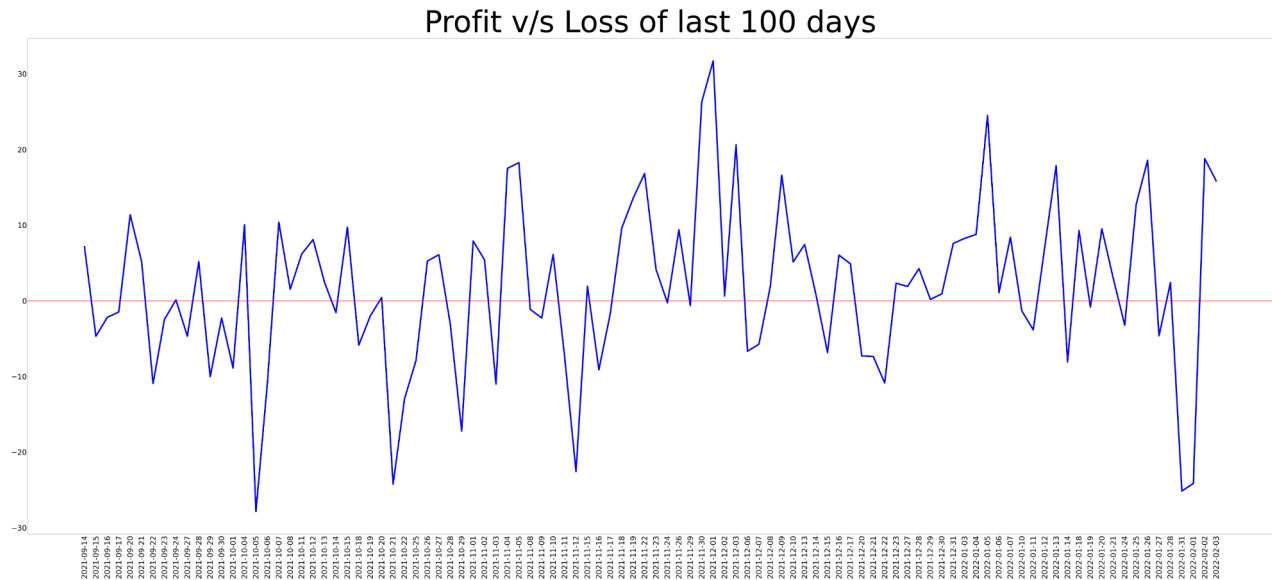
By observing the tendency of the graph, I have noticed that the difference between the Opening and Closing is not too much high.

After Open v/s Close graph I have plotted a line chart using the library 'Matplotlib' to analyse the graph between the Volume and Date of the stock, and Adj Close value v/s Date



From the above charts, I have observed that Day by day the price has increased. I have also noticed that the volume decrease when the stock price grows, and volumes increase when the stock price start decreasing.

I have taken the last 100 data, to observe the trend of the graph, to see if investing now is profitable or not. After plotting that 100 data I have got this graph.



And the tendency of the graph is proof that investing in this stock will be profitable for the customers.

The overall tendency of the graph is increasing, and it has given a very good return to customers within three years. So I will recommend investing in this stock.

Future Trend Prediction of The Stock

To predict the trend of the stock , first I have taken the 'Close' value in a object named y, and rest values like , 'Open,'High', and 'Low' in X.

By using 'StandardScaler' a library of SkLearn ,I have scaled the values of X. Then I have created a function named "split" to split the . Then I have split the data for train and validation.

I have created the LSTM model to train the data set. First I have taken 50 LSTM layers with activation function "relu", And then another 50 layers of LSTM ,with "relu" activation function. Then added a Dense layer. Compiled the model with mean_squared_error loss and using Adam optimizer.

Creating the ML Model

```
In [26]: model=Sequential()
model.add(LSTM(50,input_shape=(X_train.shape[1],X_train.shape[2]),activation='relu',return_sequences=True))
model.add(LSTM(50,activation="relu"))
model.add(Dense(1))
model.compile(loss="mean_squared_error",optimizer="adam")
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
lstm (LSTM)	(None, 10, 50)	10600
lstm_1 (LSTM)	(None, 50)	20200
dense (Dense)	(None, 1)	51

=====
Total params: 30,851
Trainable params: 30,851
Non-trainable params: 0

2023-07-01 15:45:13.863299: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: SSE4.1 SSE4.2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

Fitting the Dataset in the model

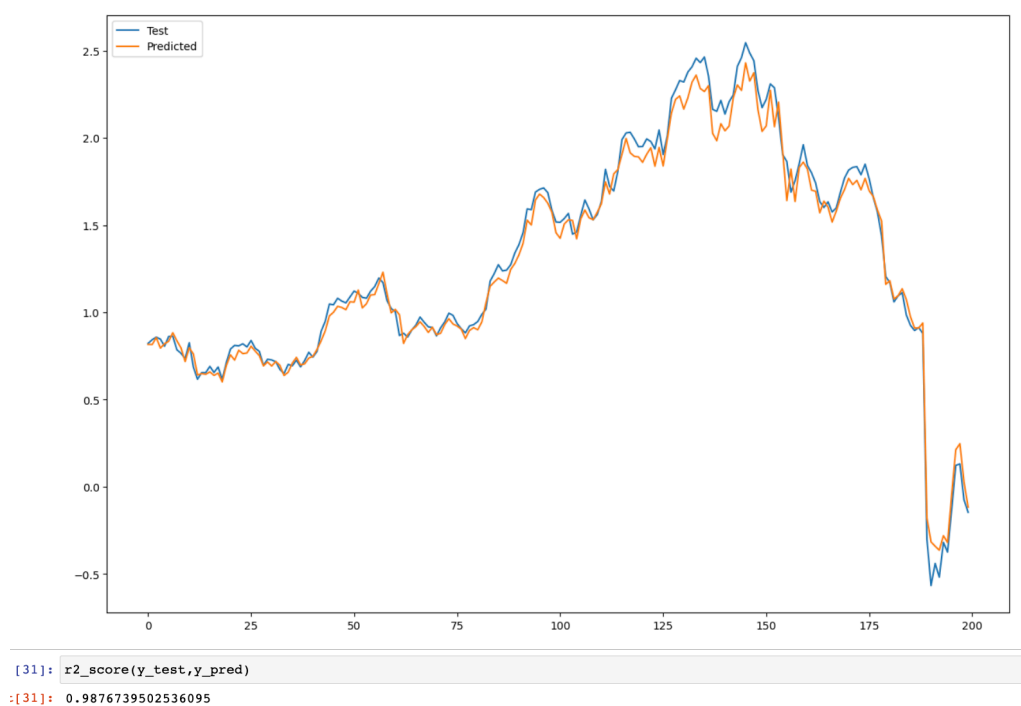
```
In [27]: model.fit(X_train,y_train,epochs=100,verbose=2)
Epoch 52/100
25/25 - 0s - loss: 0.0020 - 81ms/epoch - 3ms/step
Epoch 93/100
25/25 - 0s - loss: 0.0021 - 82ms/epoch - 3ms/step
Epoch 94/100
25/25 - 0s - loss: 0.0021 - 82ms/epoch - 3ms/step
Epoch 95/100
25/25 - 0s - loss: 0.0021 - 82ms/epoch - 3ms/step
Epoch 96/100
25/25 - 0s - loss: 0.0021 - 82ms/epoch - 3ms/step
Epoch 97/100
25/25 - 0s - loss: 0.0023 - 81ms/epoch - 3ms/step
Epoch 98/100
25/25 - 0s - loss: 0.0025 - 82ms/epoch - 3ms/step
Epoch 99/100
25/25 - 0s - loss: 0.0024 - 81ms/epoch - 3ms/step
Epoch 100/100
25/25 - 0s - loss: 0.0022 - 82ms/epoch - 3ms/step
25/25 - 0s - loss: 0.0022 - 82ms/epoch - 3ms/step
```

Out[27]: <keras.callbacks.History at 0x7f7f685a1660>

Caption

Then I have fitted the X_train and y_train data into the model with 100 epochs.

Then by using 'model.predict(validation values)' I gave got the predicted value , by comparing the predicted value and



Caption

the validation value it seems that the model is predicting the values with **98.7 % accuracy**.

End Of The Report.