

BIG DATA ANALYTICS LAB - PROJECT REVIEW

Project Title :

Stock Market Prediction

Team Members :

Naveen Kumar R (17BIT006)

Jegan j (17BIT016)

Boopathy K (17BIT028)

Naveeth Z A (17BIT031)

Project Completion Status :

Project Completed

Project Description :

- ★ Forecasting the stock market is a way to predict future prices of stocks. The Stock prices are dynamic day by day, so it is hard to decide what is the best time to buy and sell stocks.
- ★ It is a long time attractive topic for researchers and investors from its existence. Machine Learning provides a wide range of algorithms, which has been reported to be quite effective in predicting the future stock prices.

- ★ In this project, we explored different data mining algorithms to forecast stock market prices for the NSE stock market. Our goal is to compare various algorithms and evaluate models by comparing prediction accuracy. We examined a few models including Linear regression, Arima, LSTM, Random Forest and Support Vector Regression.
- ★ Based on the accuracy calculated using RMSE of all the models, we predicted prices of different industries. For forecasting, we used historical data of NSE stock market and applied a few preprocessing methods to make prediction more accurate and relevant.

Project modules :

- 5 major modules involved in this project
- Following algorithms are used for time series analysis:
 1. Linear regression
 2. Support Vector Regression (SVR)
 3. Long Short Term Memory (LSTM)
 4. Autoregressive Integrated Moving Average (ARIMA)
 5. Random Forest

Dataset

- Dataset was taken from [here](<https://www.kaggle.com/ramamet4/nse-company-stocks>)
- It includes India stocks and our index covers a diverse set of sectors featuring many indian companies.

	Date	Open	High	Low	Close	Volume
0	2014-12-18	561.963918	562.615636	561.317698	562.008419	148.219931
1	2014-12-19	588.985235	589.995805	588.078859	589.031208	335.969799
2	2014-12-22	603.079123	603.608772	602.417544	603.047193	169.870175
3	2014-12-23	600.358528	600.876254	599.931438	600.357692	97.444816
4	2014-12-24	588.538106	589.038987	588.018722	588.531498	117.449339
...
178	2015-09-24	1425.896581	1426.803205	1425.048077	1425.834615	85.448718
179	2015-09-28	1413.101322	1413.839868	1412.314537	1412.951322	71.374449
180	2015-09-29	1383.679461	1384.679253	1382.544606	1383.513900	82.506224
181	2015-09-30	1400.488693	1400.946985	1399.891960	1400.346734	54.628141
182	2015-10-01	1423.937900	1424.694521	1422.964840	1423.700000	94.360731

183 rows × 6 columns

Team Members Contribution :

1. Jegan J (17BIT031):

My part is to retrieve datasets from the available resources and implementing the linear regression module along with the other team member Naveeth.

I came across various websites and came across some effective datasets from the website called Kaggle. Here is the dataset link

"<https://www.kaggle.com/ramamet4/nse-company-stocks>".

Two datasets namely "groupedddf.csv" and "infotech.csv" were taken from there which was used for our project.

As for the implementing part, I came to an understanding what Linear Regression is and how it works.

Linear Regression is a method of deriving an approximate value between the variation in prices.

It was mainly used in our project to analyse a variation in stock prices for a given time period and that data is used to provide the stock prices for the upcoming dates. This method also showed better accuracy than the other methods.

2. Naveeth Z A (17BIT031)

As we discussed with the team , each teammate took two algorithm to study and analyze the stock market and predict its future trends

So I took the **Random forest** and **Linear Regression algorithm** for Stock market Prediction.

- First I took the Random Forest algorithm and applied to the data set. I was able to predict the future stock values with an RMSE Value of 0.502885160662743
- Next I applied the linear regression algorithm to the data set. As i did for Random forest, i was able to predict the stock values but with an higher precision RMSE value of 0.5048180544142529

Linear Regression was more accurate than the Random Forest method.

3. Boopathy k (17BIT028):

Sir,I spent some days studying two algorithms such as the ARIMA model and SVR model. ARIMA model is nothing but it is basically used for statistical analysis and SVR model is referred to as the Support Vector Machine- a classification algorithm, but it applies to predict real values. I'm working on both algorithms but i can't get a clear result on the SVR algorithm which is especially used to predict the values on stock.

One of my teammates also supported me but the expected result was not possible. So that we decided to work with the ARIMA model, using this my work is done.

4. Naveen Kumar (17BIT006)

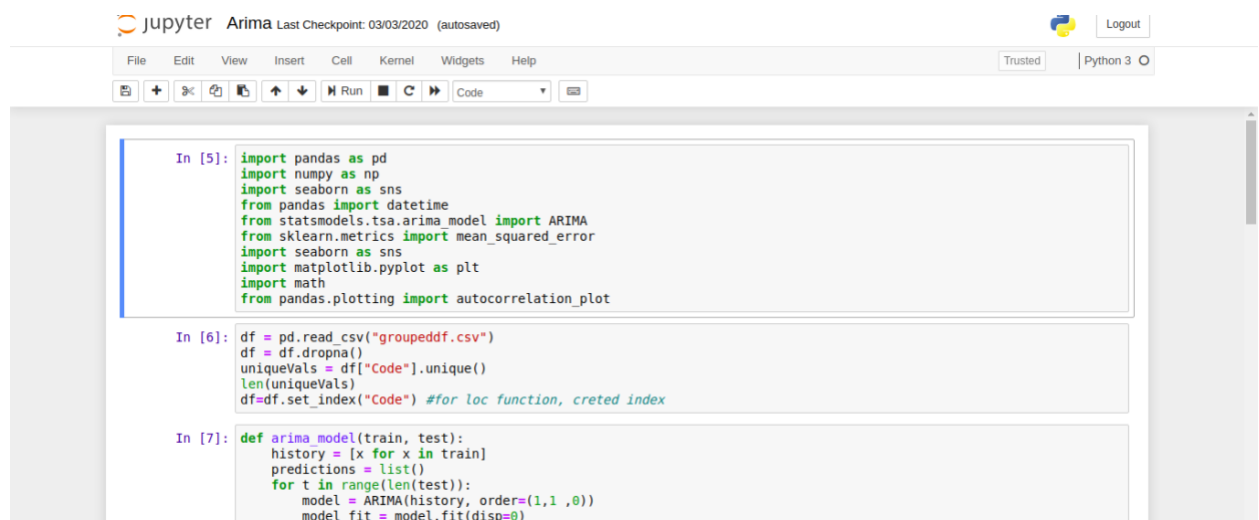
The work given to me is to check which model gives the best results for the taken dataset (ex. 2015). So I took arima and lstm model to check which gives approximately close results to the original current stock data set (ex.2019). So when I use lstm it seems the prices Open, Close, Low, High is vary too much from the current dataset. So I used the arima model which results approximately close to all the prices like Open, Close, Low, High.

So I suggest my team go with an arima model.

Project Screenshots :

1. Arima Model

Module code :



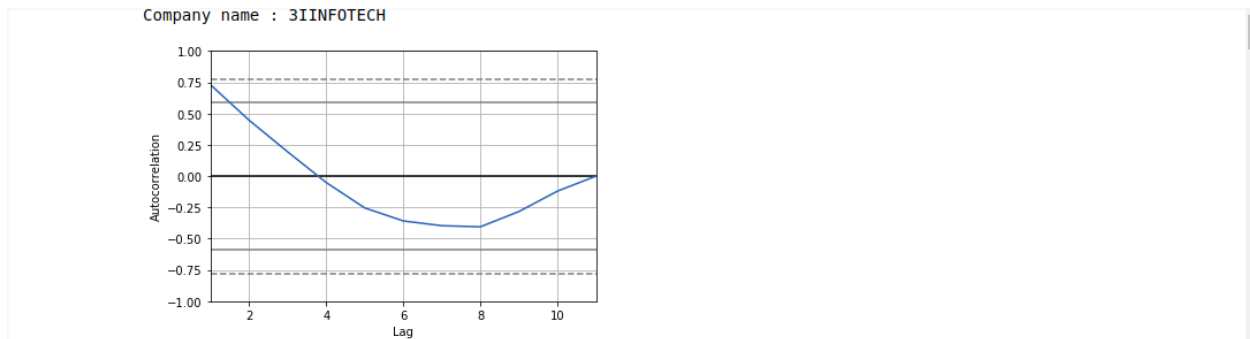
```
In [5]: import pandas as pd
import numpy as np
import seaborn as sns
from pandas import datetime
from statsmodels.tsa.arima_model import ARIMA
from sklearn.metrics import mean_squared_error
import seaborn as sns
import matplotlib.pyplot as plt
import math
from pandas.plotting import autocorrelation_plot

In [6]: df = pd.read_csv("groupeddf.csv")
df = df.dropna()
uniqueVals = df["Code"].unique()
len(uniqueVals)
df=df.set_index("Code") #for loc function, created index

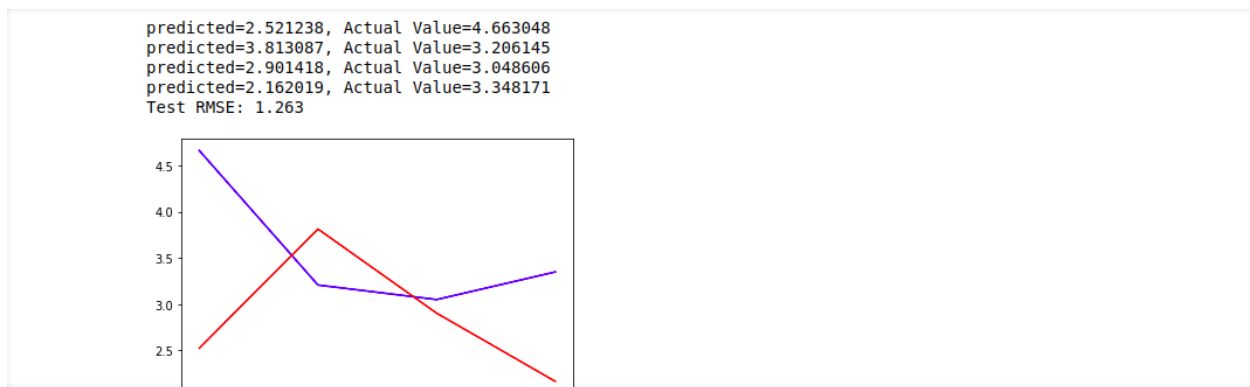
In [7]: def arima_model(train, test):
    history = [x for x in train]
    predictions = list()
    for t in range(len(test)):
        model = ARIMA(history, order=(1,1,0))
        model_fit = model.fit(dispatch=0)
```

Data Analysis of ARIMA MODEL:

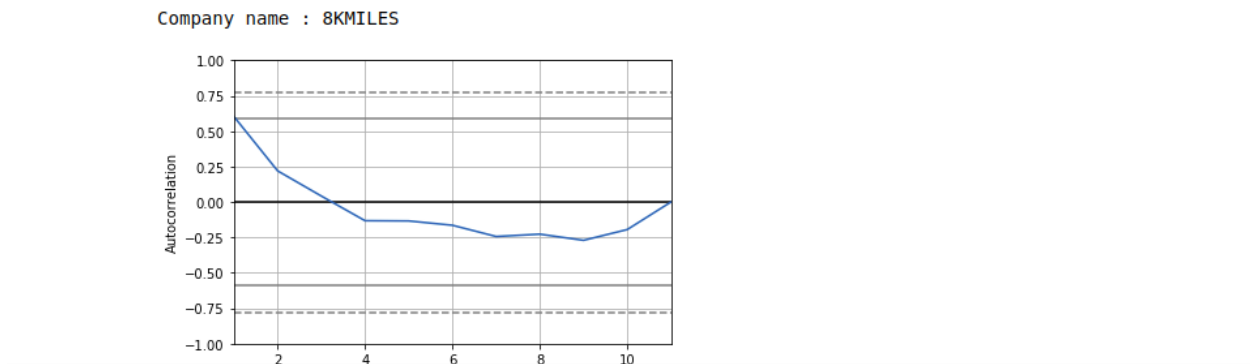
1. Company Name - 3IINFOTECH



Output :

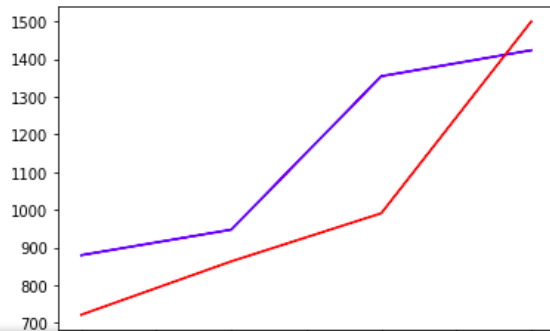


Company Name - 8KMILES



Output -

predicted=721.484495, Actual Value=879.681825
predicted=863.211881, Actual Value=947.393459
predicted=990.871242, Actual Value=1354.689960
predicted=1500.078689, Actual Value=1423.700000
Test RMSE: 206.344



Linear Regression

- Code

```
In [7]: from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score, mean_squared_error
```

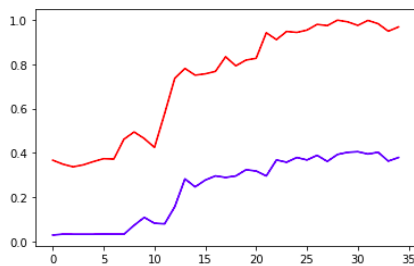
```
In [8]: def create_dataset(dataset, past=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-past-1):
        j = dataset[i:(i+past), 0]
        dataX.append(j)
        dataY.append(dataset[i + past, 0])
    return np.array(dataX), np.array(dataY)
```

```
In [9]: from sklearn.preprocessing import MinMaxScaler
def testandtrain(prices):
    scaler = MinMaxScaler(feature_range=(0, 1))
    prices = scaler.fit_transform(prices)
    trainsize = int(len(prices) * 0.80)
    testsize = len(prices) - trainsize
    train, test = prices[0:trainsize, :], prices[trainsize:len(prices), :]
    print(len(train), len(test))

    x_train, y_train = create_dataset(train, 1)
    x_test, y_test = create_dataset(test, 1)
```

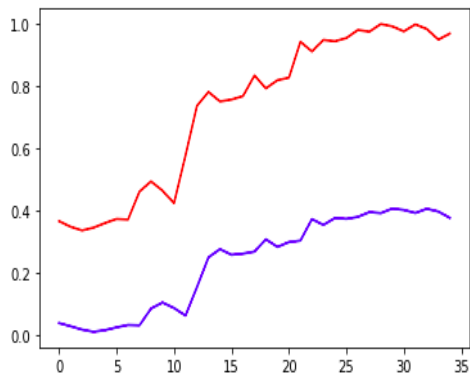
1. Original Data Set Value

```
In [17]: plt.plot(test,color="blue")  
plt.plot(testY,color='red')  
plt.show()
```



Predicted Value using Linear Regression

```
In [20]: plt.plot(test,color="blue")  
plt.plot(testY,color='red')  
plt.show()
```



**Original Value and the predicted Value
were the same**

CONCLUSION

1. Algorithms compared using RMSE calculated for each model.
2. LSTM and Arima model gave lowest RMSE and highest accurate prediction. So these are winning algorithms.
3. SVR gave the highest RMSE.

