GOOGLE STOCK PRICE PREDICTION

18CSE398J -Machine Learning – Core Concepts with Applications

(2018 Regulation)
III Year/VI Semester
Academic Year: 2022- 2023
By

MELAM SIVARAM- RA2011003011208

Under the guidance of

Vadivu G
Professor
Department of Data Science And Business Systems

Department of Data Science And Business Systems
Faculty Of Engineering and Technology
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Kancheepuram

MAY 2023

NAME- MELAM SIVA RAM
REG NO- RA2011003011208
Gmail- mr6352@srmist.edu.in

Github link: https://github.com/siva123456789098/ML_project1208

TABLE OF CONTENT:

ABSTRACT

In the world of finance, stock trading is the most essential activity. Predicting the stock market is an act of determining the value of a stock in near future and other financial instruments traded on the financial exchange such as NSE, BSE. The fundamental and technical analysis is being in useby the brokers of stock exchange when stocks are being predicted. Here in this report we proposed the method which is called as Machine learning (ML) which is made available by training the stock data, will then gain intelligence and thus finally uses the acquired knowledge for an appropriate prediction. We used many techniques such as Linear Regression, Support Vector Machine and Decision Tree to predict prices of a stock for small and large capitalizations and in the different markets, employing prices daily with the minute frequencies. Linear Regression is used for when the data is in the form of Linearity, or the data seems to be nearby the line to get fitted. In Support Vector Machine, when the data is spread then the line from where the most of the points pass is drawn and from there the vectors from the points to the line are drawn. Meanwhile, in Decision Tree based on the previous data decisions are made that effect of all the alternatives are checked and the most suitable one is decided for the work to be performed.

PROBLEM STATEMENT

The point of the task is to ascertain or anticipate the future stock costs of organizations utilizingan alternate number of AI and estimating strategies reliant on authentic returns just as numericalnews markers to fabricate an arrangement of numerous or different stocks so as to expand the issue. We do this by putting managed learning techniques for stock value anticipating by understanding the idea of dataset.

OBJECTIVES

To create take a dataset of renowned company.

Feature extraction using fundamental analysis

Applying reduced dataset

Evaluating accuracy

Plotting and analyzing the graph

SCOPE OF THE PROJECT

The project aims to predict the prices of a basket of stocks on the NSE/BSE with an acceptable degree of accuracy. By having an idea about the price of a stock in the market prior to its sale, we will know beforehand which stock to purchase thus making a profit. The successful prediction of a stock's future price could yield significant profit. The efficient-market hypothesis suggests that stock prices reflect all currently available information and any price changes that are not based on newly revealed information thus are inherently unpredictable.

Methodologies

LSTM:

Long Short-Term Memory (LSTM) is a type of recurrent neural network that can be used in stock price prediction. LSTM networks are designed to handle sequence data and are capable of capturing long-term dependencies in the input data.

In stock price prediction, LSTM can be trained on historical stock prices and other relevant information such as economic indicators, news articles, and social media sentiment. The LSTM network can then use this information to make predictions about future stock prices.

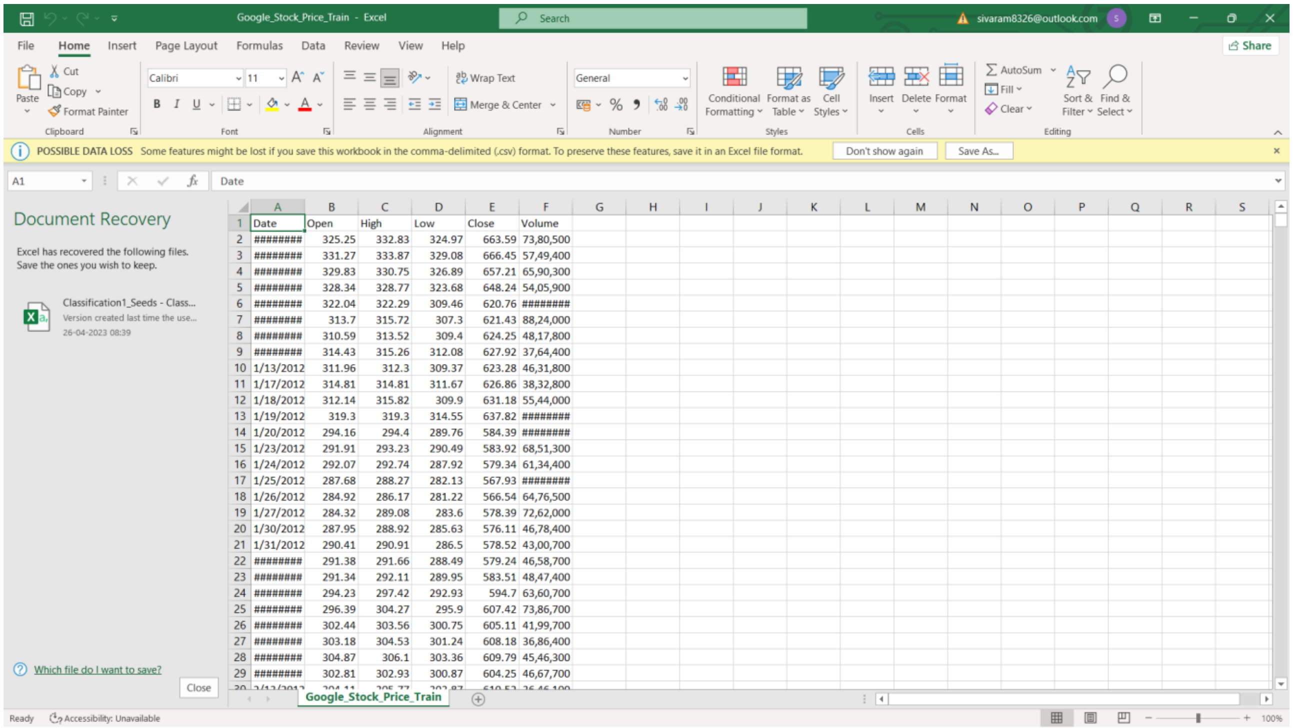
To use LSTM in stock price prediction, you can follow these steps:

1. Collect historical stock price data and other relevant information.
2. Preprocess the data by normalizing the prices and encoding other information such as economic indicators as numerical features.
3. Split the data into training and testing sets.
4. Build an LSTM model using a framework such as TensorFlow or PyTorch.
5. Train the model on the training set.
6. Test the model on the testing set and evaluate its performance using metrics such as Mean Squared Error (MSE) or Root Mean Squared Error (RMSE).

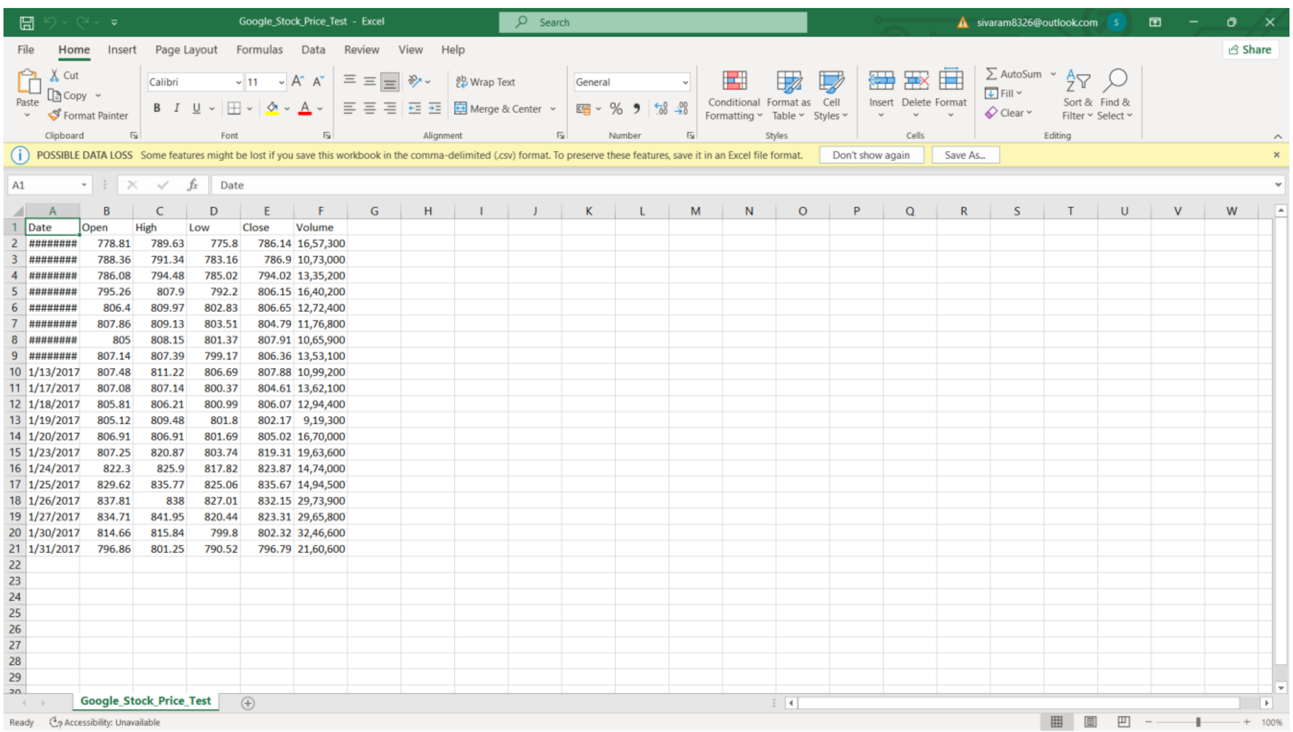
Use the trained LSTM model to make predictions on new data.

It's important to note that stock price prediction is a challenging task and that the accuracy of the predictions can be affected by a range of factors such as market volatility, unexpected events, and changes in economic conditions. Therefore, it's important to use caution when making investment decisions based on predictions made by machine learning models.

DATASETS TRAIN AND TEST

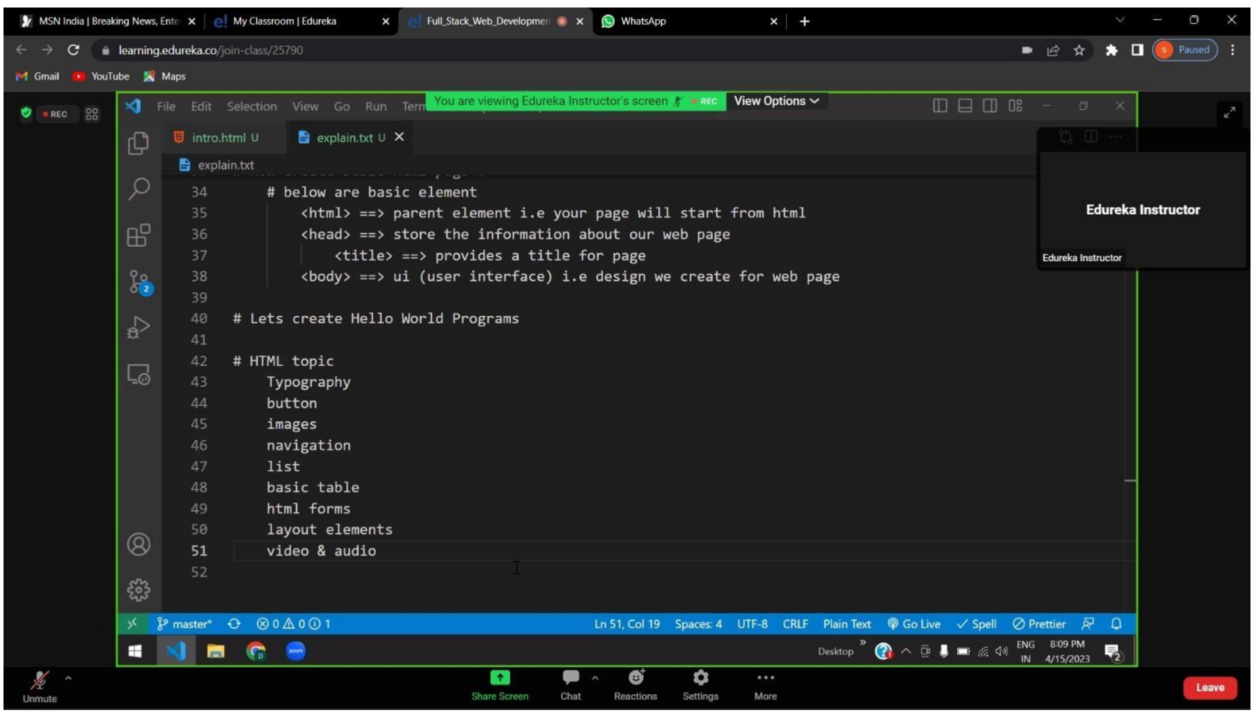


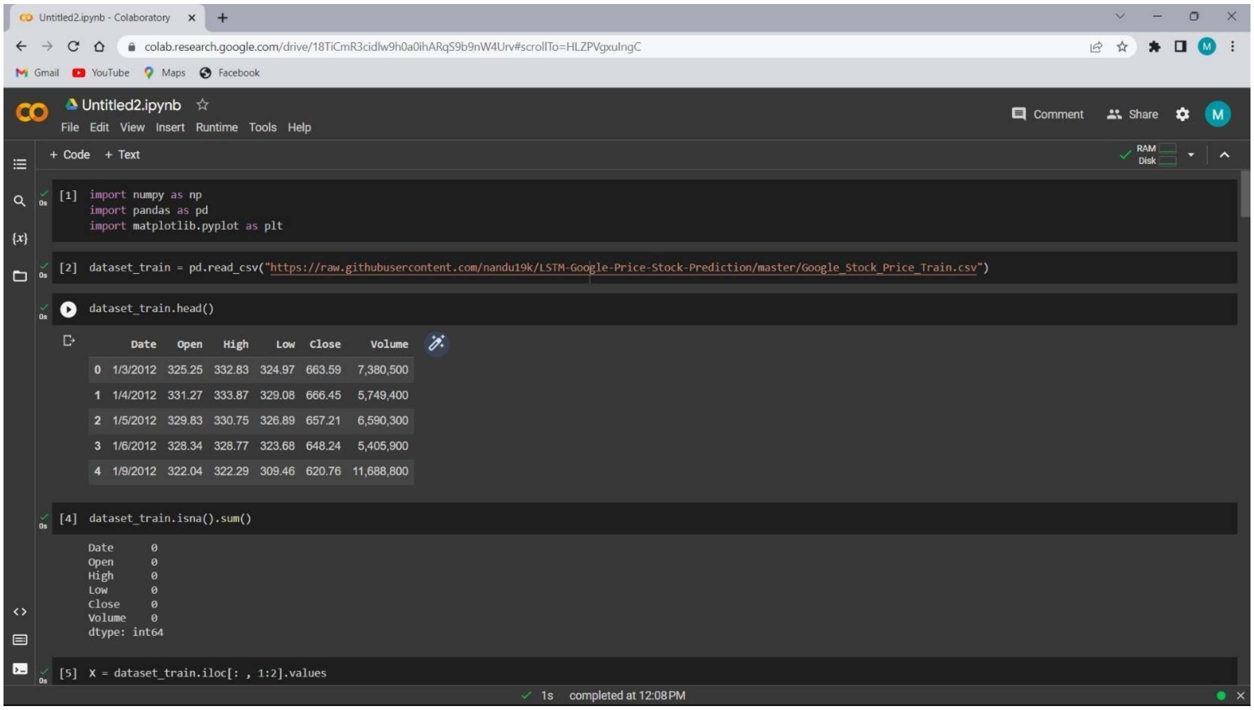
	Date	Open	High	Low	Close	Volume
2	########	325.25	332.83	324.97	663.59	73,80,500
3	########	331.27	333.87	329.08	666.45	57,49,400
4	########	329.83	330.75	326.89	657.21	65,90,300
5	########	328.34	328.77	323.68	648.24	54,05,900
6	########	322.04	322.29	309.46	620.76	########
7	########	313.7	315.72	307.3	621.43	88,24,000
8	########	310.59	313.52	309.4	624.25	48,17,800
9	########	314.43	315.26	312.08	627.92	37,64,400
10	1/13/2012	311.96	312.3	309.37	623.28	46,31,800
11	1/17/2012	314.81	314.81	311.67	626.86	38,32,800
12	1/18/2012	312.14	315.82	309.9	631.18	55,44,000
13	1/19/2012	319.3	319.3	314.55	637.82	########
14	1/20/2012	294.16	294.4	289.76	584.39	########
15	1/23/2012	291.91	293.23	290.49	583.92	68,51,300
16	1/24/2012	292.07	292.74	287.92	579.34	61,34,400
17	1/25/2012	287.68	288.27	282.13	567.93	########
18	1/26/2012	284.92	286.17	281.22	566.54	64,76,500
19	1/27/2012	284.32	289.08	283.6	578.39	72,62,000
20	1/30/2012	287.95	288.92	285.63	576.11	46,78,400
21	1/31/2012	290.41	290.91	286.5	578.52	43,00,700
22	########	291.38	291.66	288.49	579.24	46,58,700
23	########	291.34	292.11	289.95	583.51	48,47,400
24	########	294.23	297.42	292.93	594.7	63,60,700
25	########	296.39	304.27	295.9	607.42	73,86,700
26	########	302.44	303.56	300.75	605.11	41,99,700
27	########	303.18	304.53	301.24	608.18	36,86,400
28	########	304.87	306.1	303.36	609.79	45,46,300
29	########	302.81	302.93	300.87	604.25	46,67,700

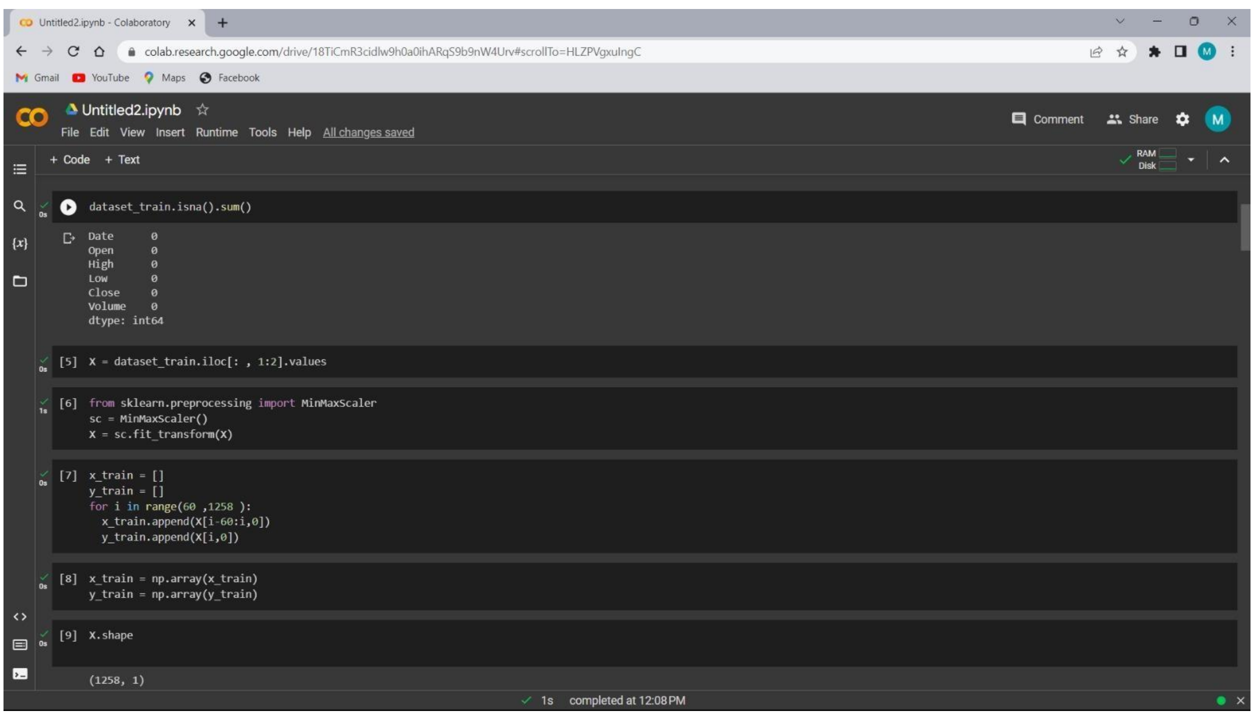


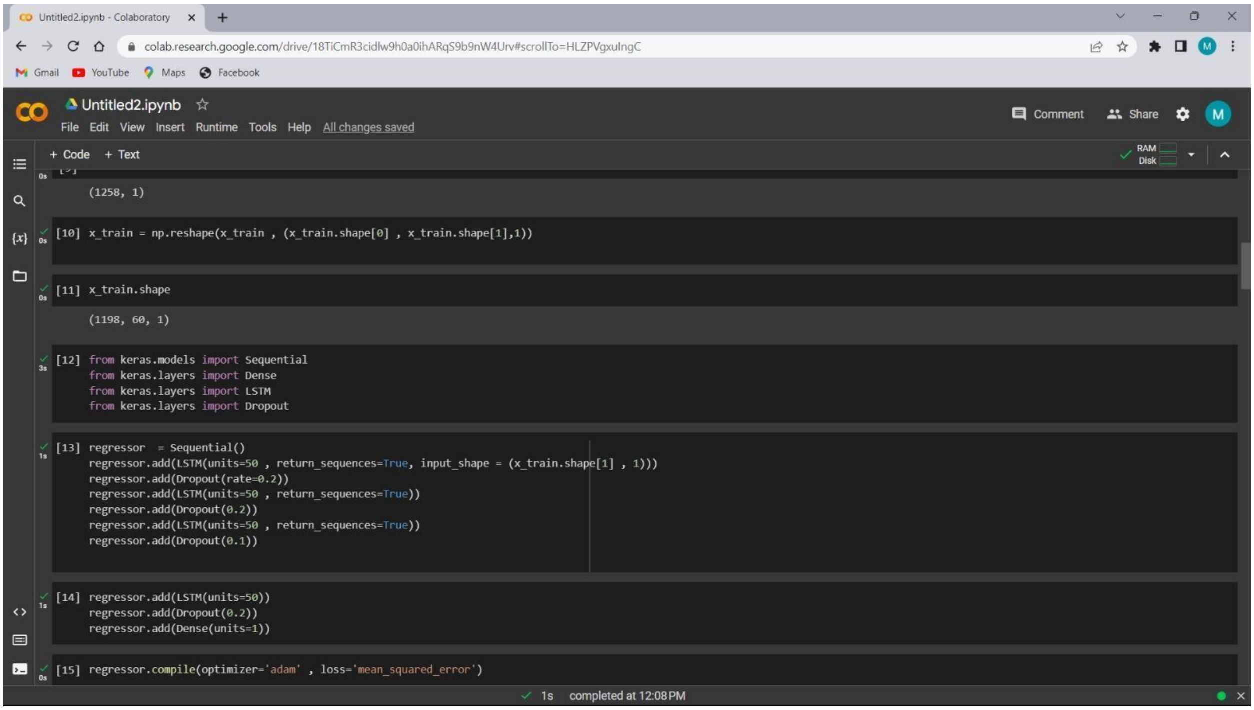
	Date	Open	High	Low	Close	Volume
2	########	778.81	789.63	775.8	786.14	16,57,300
3	########	788.36	791.34	783.16	786.9	10,73,000
4	########	786.08	794.48	785.02	794.02	13,35,200
5	########	795.26	807.9	792.2	806.15	16,40,200
6	########	806.4	809.97	802.83	806.65	12,72,400
7	########	807.86	809.13	803.51	804.79	11,76,800
8	########	805	808.15	801.37	807.91	10,65,900
9	########	807.14	807.39	799.17	806.36	13,53,100
10	1/13/2017	807.48	811.22	806.69	807.88	10,99,200
11	1/17/2017	807.08	807.14	800.37	804.61	13,62,100
12	1/18/2017	805.81	806.21	800.99	806.07	12,94,400
13	1/19/2017	805.12	809.48	801.8	802.17	9,19,300
14	1/20/2017	806.91	806.91	801.69	805.02	16,70,000
15	1/23/2017	807.25	820.87	803.74	819.31	19,63,600
16	1/24/2017	822.3	825.9	817.82	823.87	14,74,000
17	1/25/2017	829.62	835.77	825.06	835.67	14,94,500
18	1/26/2017	837.81	838	827.01	832.15	29,73,900
19	1/27/2017	834.71	841.95	820.44	823.31	29,65,800
20	1/30/2017	814.66	815.84	799.8	802.32	32,46,600
21	1/31/2017	796.86	801.25	790.52	796.79	21,60,600

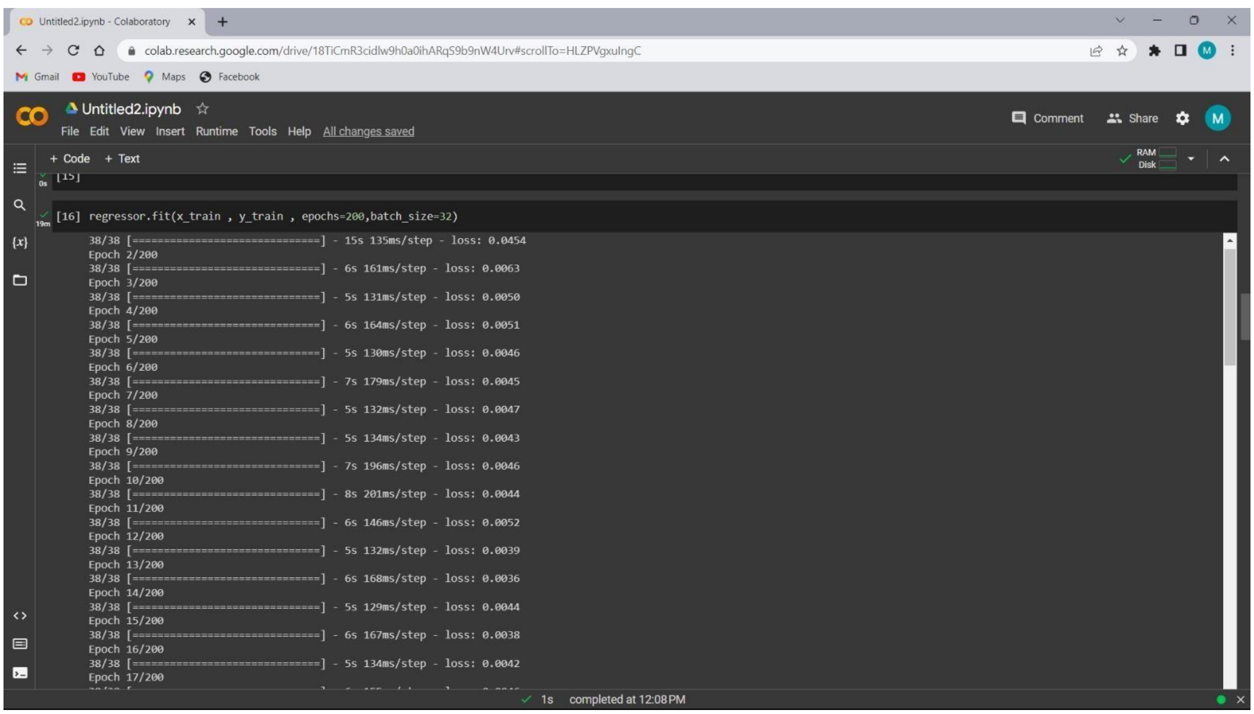
EXECUTION CODE AND SCREEN SHORTS OF PROJECT

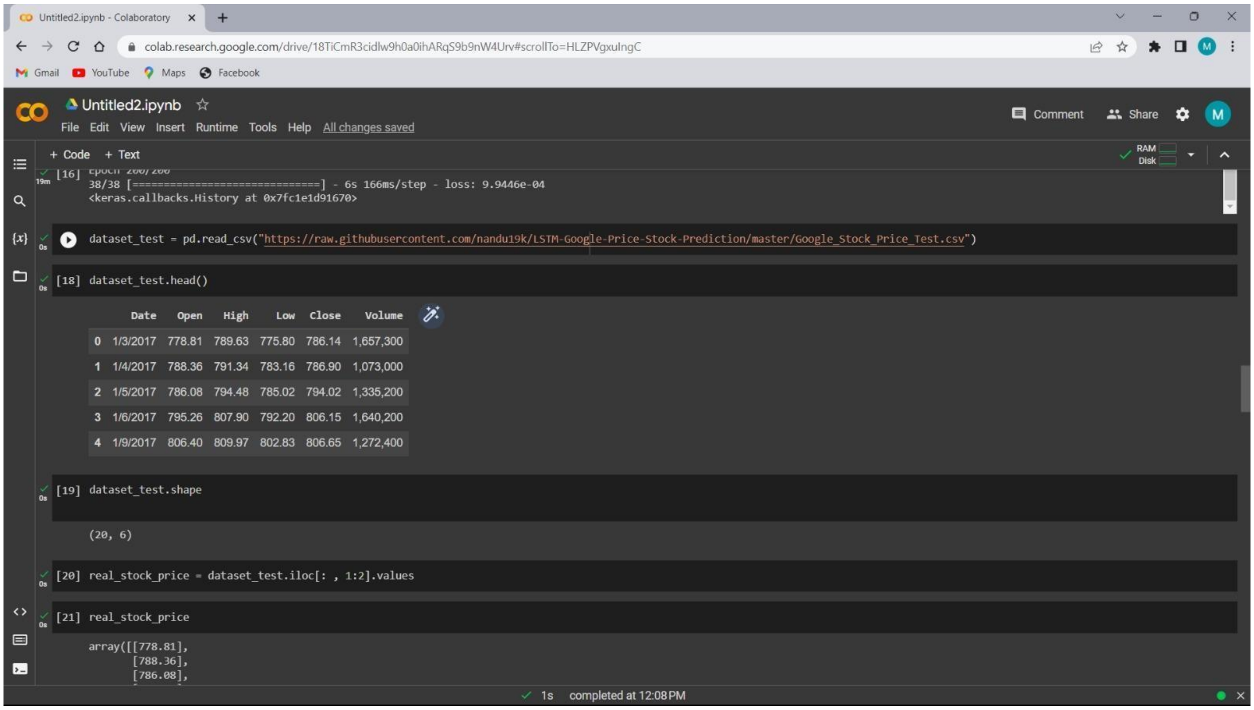


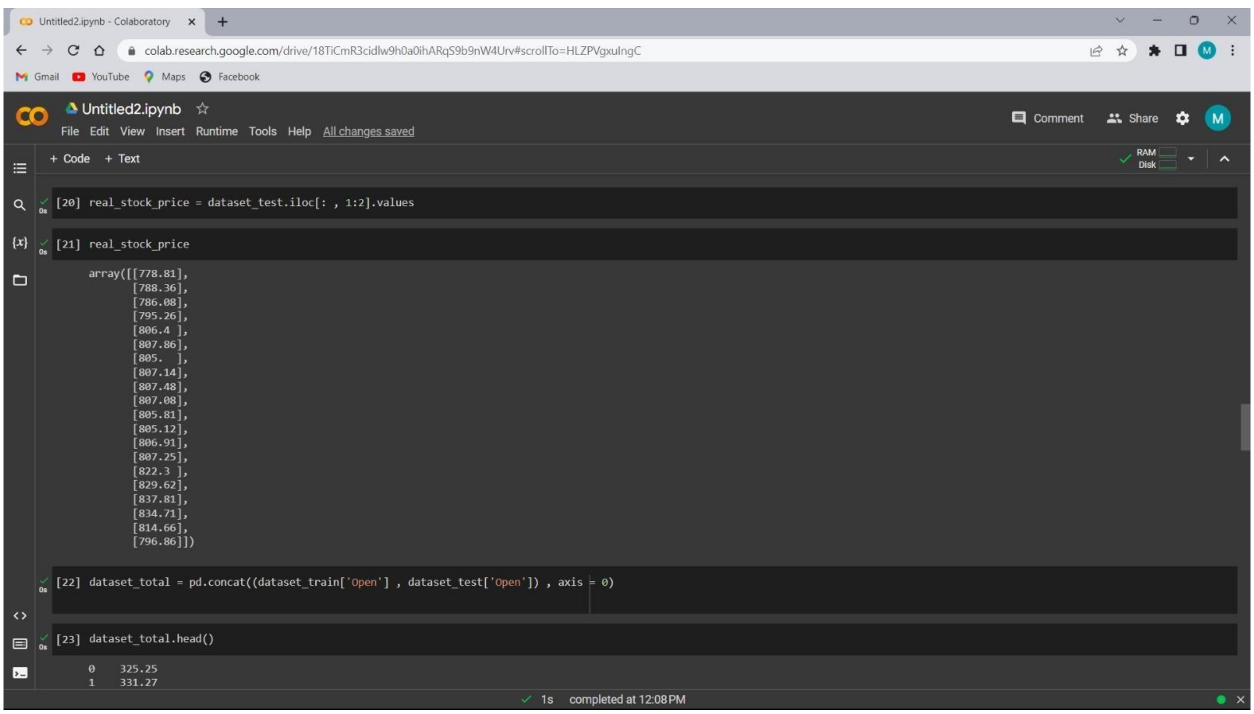

Untitled2.ipynb - Colaboratory
colab.research.google.com/drive/18TiCmR3cidlw9h0a0ihARqS9b9nW4Urv#scrollTo=HLZPVgxulngC
Gmail
YouTube
Maps
Facebook
Untitled2.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
[1] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
[2] dataset_train = pd.read_csv("https://raw.githubusercontent.com/nandu19k/LSTM-Google-Price-Stock-Prediction/master/Google_Stock_Price_Train.csv")
dataset_train.head()
Date Open High Low Close Volume
0 1/3/2012 325.25 332.83 324.97 663.59 7,380,500
1 1/4/2012 331.27 333.87 329.08 666.45 5,749,400
2 1/5/2012 329.83 330.75 326.89 657.21 6,590,300
3 1/6/2012 328.34 328.77 323.68 648.24 5,405,900
4 1/9/2012 322.04 322.29 309.46 620.76 11,688,800
[4] dataset_train.isna().sum()
Date 0
Open 0
High 0
Low 0
Close 0
Volume 0
dtype: int64
[5] X = dataset_train.iloc[: , 1:2].values
1s completed at 12:08 PM

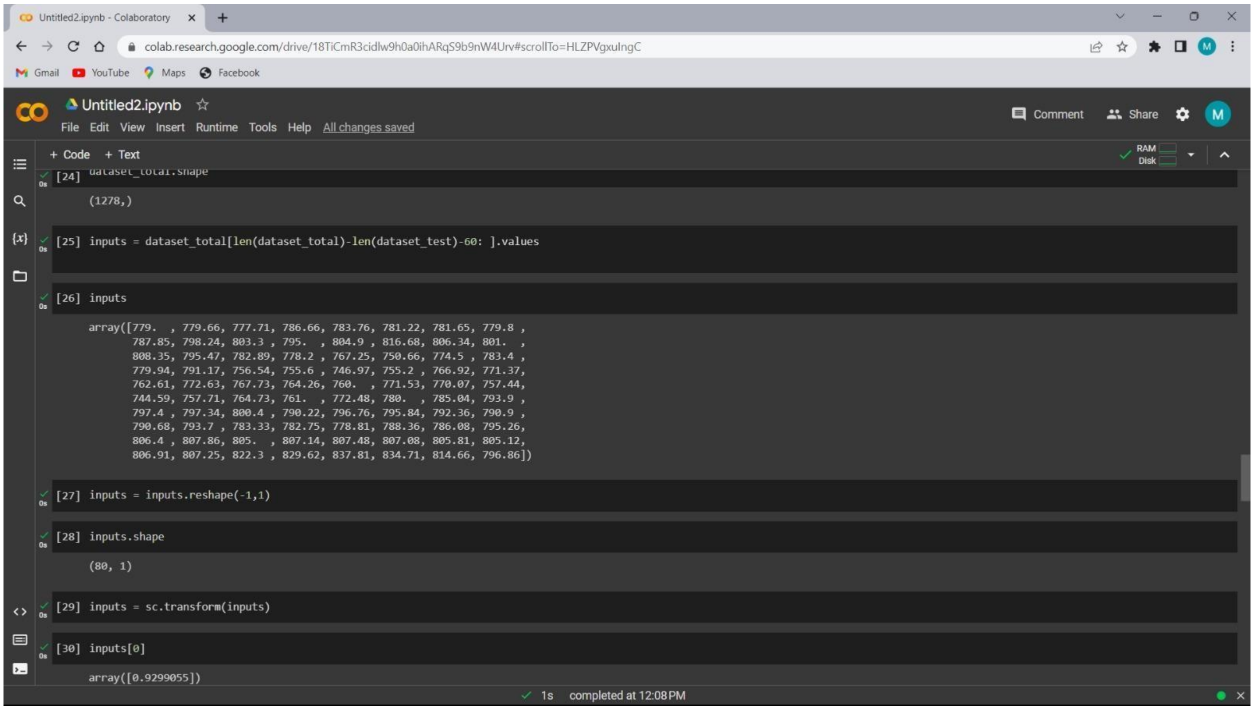

Untitled2.ipynb - Colaboratory
colab.research.google.com/drive/18TiCmR3cidlw9h0a0ihARqS9b9nW4Urv#scrollTo=HLZPVgxulngC
Gmail
YouTube
Maps
Facebook
Untitled2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
dataset_train.isna().sum()
Date 0
Open 0
High 0
Low 0
Close 0
Volume 0
dtype: int64
[5] X = dataset_train.iloc[: , 1:2].values
[6] from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler()
X = sc.fit_transform(X)
[7] x_train = []
y_train = []
for i in range(60 ,1258):
x_train.append(X[i-60:i,0])
y_train.append(X[i,0])
[8] x_train = np.array(x_train)
y_train = np.array(y_train)
[9] X.shape
(1258, 1)
1s completed at 12:08 PM

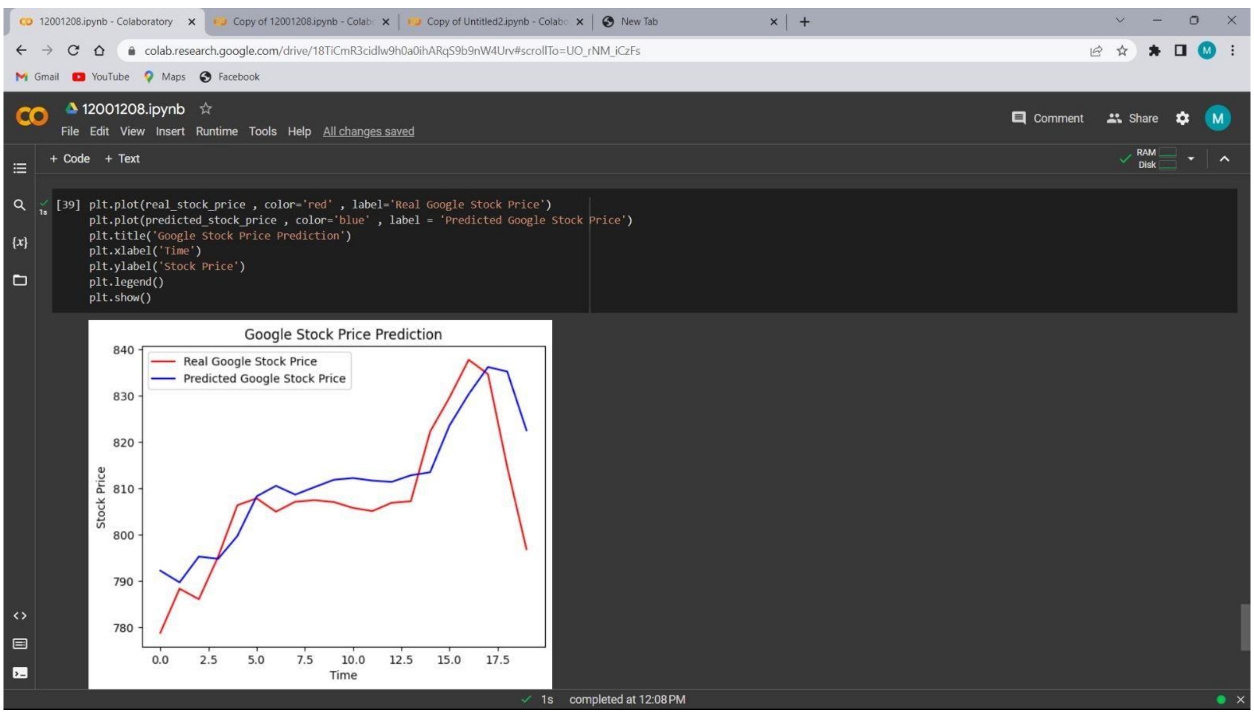

Untitled2.ipynb - Colaboratory
colab.research.google.com/drive/18TiCmR3cidlw9h0a0ihARqS9b9nW4Urv#scrollTo=HLZPVgxulngC
Gmail
YouTube
Maps
Facebook
Untitled2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
(1258, 1)
[10] x_train = np.reshape(x_train , (x_train.shape[0] , x_train.shape[1],1))
[11] x_train.shape
(1198, 60, 1)
[12] from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
[13] regressor = Sequential()
regressor.add(LSTM(units=50 , return_sequences=True, input_shape = (x_train.shape[1] , 1)))
regressor.add(Dropout(rate=0.2))
regressor.add(LSTM(units=50 , return_sequences=True))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units=50 , return_sequences=True))
regressor.add(Dropout(0.1))
[14] regressor.add(LSTM(units=50))
regressor.add(Dropout(0.2))
regressor.add(Dense(units=1))
[15] regressor.compile(optimizer='adam' , loss='mean_squared_error')
1s completed at 12:08 PM


Untitled2.ipynb - Colaboratory
colab.research.google.com/drive/18TiCmR3cidlw9h0a0ihARqS9b9nW4Urv#scrollTo=HLZPVgxulngC
Gmail
YouTube
Maps
Facebook
Untitled2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[16] regressor.fit(x_train , y_train , epochs=200,batch_size=32)
38/38 [==============================] - 15s 135ms/step - loss: 0.0454
Epoch 2/200
38/38 [==============================] - 6s 161ms/step - loss: 0.0063
Epoch 3/200
38/38 [==============================] - 5s 131ms/step - loss: 0.0050
Epoch 4/200
38/38 [==============================] - 6s 164ms/step - loss: 0.0051
Epoch 5/200
38/38 [==============================] - 5s 130ms/step - loss: 0.0046
Epoch 6/200
38/38 [==============================] - 7s 179ms/step - loss: 0.0045
Epoch 7/200
38/38 [==============================] - 5s 132ms/step - loss: 0.0047
Epoch 8/200
38/38 [==============================] - 5s 134ms/step - loss: 0.0043
Epoch 9/200
38/38 [==============================] - 7s 196ms/step - loss: 0.0046
Epoch 10/200
38/38 [==============================] - 8s 201ms/step - loss: 0.0044
Epoch 11/200
38/38 [==============================] - 6s 146ms/step - loss: 0.0052
Epoch 12/200
38/38 [==============================] - 5s 132ms/step - loss: 0.0039
Epoch 13/200
38/38 [==============================] - 6s 168ms/step - loss: 0.0036
Epoch 14/200
38/38 [==============================] - 5s 129ms/step - loss: 0.0044
Epoch 15/200
38/38 [==============================] - 6s 167ms/step - loss: 0.0038
Epoch 16/200
38/38 [==============================] - 5s 134ms/step - loss: 0.0042
Epoch 17/200
1s completed at 12:08 PM


Untitled2.ipynb - Colaboratory
colab.research.google.com/drive/18TiCmR3cidlw9h0a0ihARqS9b9nW4Urv#scrollTo=HLZPVgxulngC
Untitled2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
[16]
38/38 [==============================] - 6s 166ms/step - loss: 9.9446e-04
<keras.callbacks.History at 0x7fc1e1d91670>
dataset_test = pd.read_csv("https://raw.githubusercontent.com/nandu19k/LSTM-Google-Price-Stock-Prediction/master/Google_Stock_Price_Test.csv")
[18] dataset_test.head()
Date Open High Low Close Volume
0 1/3/2017 778.81 789.63 775.80 786.14 1,657,300
1 1/4/2017 788.36 791.34 783.16 786.90 1,073,000
2 1/5/2017 786.08 794.48 785.02 794.02 1,335,200
3 1/6/2017 795.26 807.90 792.20 806.15 1,640,200
4 1/9/2017 806.40 809.97 802.83 806.65 1,272,400
[19] dataset_test.shape
(20, 6)
[20] real_stock_price = dataset_test.iloc[: , 1:2].values
[21] real_stock_price
array([[778.81],
[788.36],
[786.08],
1s completed at 12:08 PM


Untitled2.ipynb - Colaboratory
colab.research.google.com/drive/18TiCmR3cidlw9h0a0ihARqS9b9nW4Urv#scrollTo=HLZPVgxulngC
Untitled2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
[20] real_stock_price = dataset_test.iloc[: , 1:2].values
[21] real_stock_price
array([[778.81],
[788.36],
[786.08],
[795.26],
[806.4],
[807.86],
[805.],
[807.14],
[807.48],
[807.08],
[805.81],
[805.12],
[806.91],
[807.25],
[822.3],
[829.62],
[837.81],
[834.71],
[814.66],
[796.86]])
[22] dataset_total = pd.concat((dataset_train['Open'] , dataset_test['Open']) , axis = 0)
[23] dataset_total.head()
0 325.25
1 331.27
1s completed at 12:08 PM


Untitled2.ipynb - Colaboratory
colab.research.google.com/drive/18TiCmR3cidlw9h0a0ihARqS9b9nW4Urv#scrollTo=HLZPVgxuIngC
Gmail YouTube Maps Facebook
Untitled2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
[24] dataset_total.shape
(1278,)
[25] inputs = dataset_total[len(dataset_total)-len(dataset_test)-60:].values
[26] inputs
array([779. , 779.66, 777.71, 786.66, 783.76, 781.22, 781.65, 779.8 ,
787.85, 798.24, 803.3 , 795. , 804.9 , 816.68, 806.34, 801. ,
808.35, 795.47, 782.89, 778.2 , 767.25, 750.66, 774.5 , 783.4 ,
779.94, 791.17, 756.54, 755.6 , 746.97, 755.2 , 766.92, 771.37,
762.61, 772.63, 767.73, 764.26, 760. , 771.53, 770.07, 757.44,
744.59, 757.71, 764.73, 761. , 772.48, 780. , 785.04, 793.9 ,
797.4 , 797.34, 800.4 , 790.22, 796.76, 795.84, 792.36, 790.9 ,
790.68, 793.7 , 783.33, 782.75, 778.81, 788.36, 786.08, 795.26,
806.4 , 807.86, 805. , 807.14, 807.48, 807.08, 805.81, 805.12,
806.91, 807.25, 822.3 , 829.62, 837.81, 834.71, 814.66, 796.86])
[27] inputs = inputs.reshape(-1,1)
[28] inputs.shape
(80, 1)
[29] inputs = sc.transform(inputs)
[30] inputs[0]
array([0.9299055])
1s completed at 12:08 PM


12001208.ipynb - Colaboratory
Copy of 12001208.ipynb - Colab
Copy of Untitled2.ipynb - Colab
New Tab
colab.research.google.com/drive/18TiCmR3cidlw9h0a0ihARqS9b9nW4Urv#scrollTo=UO_rNM_iCzFs
Gmail YouTube Maps Facebook
12001208.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
[39] plt.plot(real_stock_price , color='red' , label='Real Google Stock Price')
plt.plot(predicted_stock_price , color='blue' , label = 'Predicted Google Stock Price')
plt.title('Google Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('Stock Price')
plt.legend()
plt.show()
Google Stock Price Prediction
Real Google Stock Price
Predicted Google Stock Price
840
830
820
810
800
790
780
Stock Price
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Time
1s completed at 12:08 PM

FUTURE SCOPE OF THE PROJECT:

Our dataset and analysis method can improve potentially.

2. If more accurate algorithm and refined data with precise research is taken then future scope can be done with possible improvement.

3. Introduction of twitter feeds.

4. Advanced predictions form news feed and different websites can be taken for better results.

5. Refining key phase extraction and doing more work will definitely produce better results.

RESULT AND DISCUSSION

DISCUSSION:

- Dilemma b/w overfitting and actual prediction.

  When predictions are made in fields like the stock market where the data is dynamic and will never be the same it is difficult to believe or depend on the predictions alone as there is so much uncertainty.

- Ability to predict the general trend of a given stock.

  With the help of stock market predictions, we can now understand the working of the market better and use it to our advantage to make smart and profitable choices.

- EMA predicts next step with negligible error.

  EMA provides the best in class prediction for single step and therefore cannot be used for long term trading strategies.

- Cannot predict uncertainties.

  There are many factors that affect the stock market as a whole, and there is a significant amount of uncertain and unpredictable factors as well. Unfortunately, we are unable to make predictions on the various uncertain factors that cause fluctuations in the stock market.

- Not investment viable.

  Due to many factors which are uncertain and those which are certain as well, it is not viable to invest in stock markets just based on predictions made basedon historical and other data.

- Good learning experience.

  In order to be able to invest in the stock market or to be able to write code for stock market prediction, the amount of knowledge required is immense and vast, hence during that process so much learning happens.

RESULT:

After all these steps, we can use matplotlib to visualize the result of our predicted stock price and the actual stock price.

REFERENCES:

[1] S. M. Idrees, M. A. Alam and P. Agarwal, "A Prediction Approach for Stock Market Volatility Based on Time Series Data," in IEEE Access, vol. 7, pp. 17287-17298, 2019. doi: 10.1109/ACCESS.2019.2895252

[2] E. W. Saad, D. V. Prokhorov and D. C. Wunsch, "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks," in IEEE Transactions on Neural Networks, vol. 9, no. 6, pp. 1456-1470, Nov. 1998. doi: 10.1109/72.728395

[3] J. Chou and T. Nguyen, "Forward Forecast of Stock Price Using Sliding-Window Metaheuristic-Optimized Machine-Learning Regression," in IEEE Transactions on Industrial Informatics, vol. 14, no. 7, pp. 3132-3142, July 2018. doi: 10.1109/TII.2018.2794389

[4] P. Chang, C. Fan and C. Liu, "Integrating a Piecewise Linear Representation Method and a Neural Network Model for Stock Trading Points Prediction," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 39, no. 1, pp. 80-92, Jan. 2009. doi: 10.1109/TSMCC.2008.2007255

[5] L. Zhang, N. Liu and P. Yu, "A Novel Instantaneous Frequency Algorithm and Its Application in Stock Index Movement Prediction," in IEEE Journal of Selected Topics in Signal Processing, vol. 6, no. 4, pp. 311-318, Aug. 2012. doi: 10.1109/JSTSP.2012.2199079

[6] S. D. Bekiros, "Sign Prediction and Volatility Dynamics With Hybrid Neurofuzzy Approaches," in IEEE Transactions on Neural Networks, vol. 22, no. 12, pp. 2353-2362, Dec. 2011.doi: 10.1109/TNN.2011.216949

www.ir.juit.ac.in

www.kaggle.com

Github link:

https://github.com/siva1234
56789098/ML_project1208