**PROJECT 6:** STOCK PRICE PREDICTION

**PROJECT TITLE:** STOCK PRICE PREDICTION

**PHASE 4:** DEVELOPMENT PART 2

***Given Dataset:***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.No** | **Date** | **Open** | **High** | **Low** | **Close** | **Volume** |
| 1 | 01-03-2012 | 325.25 | 332.83 | 324.97 | 663.59 | 73,80,500 |
| 2 | 01-04-2012 | 331.27 | 333.87 | 329.08 | 666.45 | 57,49,400 |
| 3 | 01-05-2012 | 329.83 | 330.75 | 326.89 | 657.21 | 65,90,300 |
| 4 | 01-06-2012 | 328.34 | 328.77 | 323.68 | 648.24 | 54,05,900 |
| 5 | 01-09-2012 | 322.04 | 322.29 | 309.46 | 620.76 | 1,16,88,800 |
| … | … | … | … | … | … | … |
| 1254 | 12-23-2016 | 790.9 | 792.74 | 787.28 | 789.91 | 6,23,400 |
| 1255 | 12-27-2016 | 790.68 | 797.86 | 787.66 | 791.55 | 7,89,100 |
| 1256 | 12-28-2016 | 793.7 | 794.23 | 783.2 | 785.05 | 11,53,800 |
| 1257 | 12-29-2016 | 783.33 | 785.93 | 778.92 | 782.79 | 7,44,300 |
| 1258 | 12-30-2016 | 782.75 | 782.78 | 770.41 | 771.82 | 17,70,000 |

(1258 rows x 6 columns)

**1.Constructing the RNN**

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import LSTM

from keras.layers import Dropout

**2. Initialising the RNN**

regressor = Sequential()

**3.Adding the first LSTM layer and some Dropout regularisation**

regressor.add(LSTM(units = 50, return\_sequences = True, input\_shape = (X\_train.shape[1], 1)))

regressor.add(Dropout(0.2))

**4.Adding a second LSTM layer and some Dropout regularisation**

regressor.add(LSTM(units = 50, return\_sequences = True))

regressor.add(Dropout(0.2))

**5.Adding a third LSTM layer and some Dropout regularisation**

regressor.add(LSTM(units = 50, return\_sequences = True))

regressor.add(Dropout(0.2))

**6. Adding a fourth LSTM layer and some Dropout regularisation**

regressor.add(LSTM(units = 50))

regressor.add(Dropout(0.2))

**7. Adding the output layer**

regressor.add(Dense(units = 1))

**8. Compiling the RNN**

regressor.compile(optimizer = 'adam', loss = 'mean\_squared\_error')

**9.Fitting the RNN to the Training set**

regressor.fit(X\_train, y\_train, epochs = 100, batch\_size = 32)

**Making the predictions and visualising the results**

**10. Getting the real stock price of 2017**

dataset\_test = pd.read\_csv('Google\_Stock\_Price\_Test.csv')

real\_stock\_price = dataset\_test.iloc[:, 1:2].values

**11.Getting the predicted stock price of 2017**

dataset\_total = pd.concat((dataset\_train['Open'], dataset\_test['Open']), axis = 0)

inputs = dataset\_total[len(dataset\_total) - len(dataset\_test) - 60:].values

inputs = inputs.reshape(-1,1)

inputs = sc.transform(inputs)

X\_test = []

for i in range(60, 80):

X\_test.append(inputs[i-60:i, 0])

X\_test = np.array(X\_test)

X\_test = np.reshape(X\_test, (X\_test.shape[0], X\_test.shape[1], 1))

predicted\_stock\_price = regressor.predict(X\_test)

predicted\_stock\_price = sc.inverse\_transform(predicted\_stock\_price)

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')

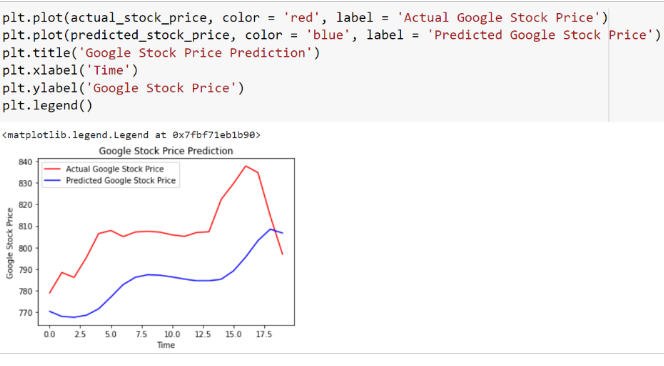
plt.xlabel('Time')

plt.ylabel('Google Stock Price')

plt.legend()

plt.show()

**12. Plotting the Actual and Predicted Prices for Google Stocks.**

****

## **Conclusion**

The stock market plays a remarkable role in our daily lives. It is a significant factor in a country's GDP growth.