PREDICTING STOCK PRICE USING DEEP LEARNING

BATCH MEMBER

621721243052 : THIRUMALAI G

Project Title: Stock Price Prediction

Phase 4: Development Part

Topic: Start building the house price prediction model by

loading and pre-processing the dataset



Phase 3 submission document

Stock Price Prediction

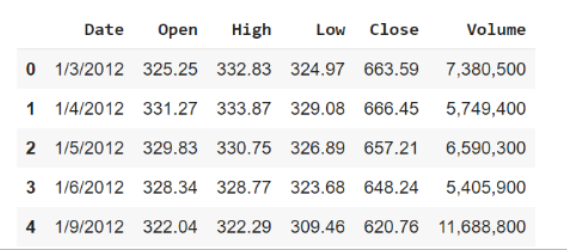
**Introduction:**

Stock Price Prediction using machine learning helps you discover the future value of company stock and other financial assets traded on an exchange. The entire idea of predicting stock prices is to gain significant profits.

**LSTMs** work in a three-step process:

* The first step in **LSTM** is to decide which information to be omitted from the cell in that particular time step. It is decided with the help of a sigmoid function. It looks at the previous state (ht-1) and the current input xt and computes the function.
* There are two functions in the second layer. The first is the sigmoid function, and the second is the tanh function. The sigmoid function decides which values to let through (0 or 1). The tanh function gives the weightage to the values passed, deciding their level of importance from -1 to 1.
* The third step is to decide what will be the final output. First, you need to run a sigmoid layer which determines what parts of the cell state make it to the output. Then, you must put the cell state through the tanh function to push the values between -1 and 1 and multiply it by the output of the sigmoid gate.

**Given data set:**



**Google Stock Price Prediction Using LSTM**

### **Import the Libraries.**

#import libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

### **Load the Training Dataset.**

There are five columns. The Open column tells the price at which a stock started trading when the market opened on a particular day. The Close column refers to the price of an individual stock when the stock exchange closed the market for the day. The High column depicts the highest price at which a stock traded during a period. The Low column tells the lowest price of the period. Volume is the total amount of trading activity during a period of time.

dataset\_train = pd.read\_csv(“Google\_Stock\_Price\_Train.csv”)

dataset\_train.head()

### **Use the Open Stock Price Column to Train Your Model.**

training\_set=dataset\_train.iloc[:,1:2].values

print(training\_set)

print(training\_set.shape)

[[325.25]

[331.27]

[329.83]

……

[793.7]

[783.33]

[782.75]]

(1258,1)

### **4. Normalizing the Dataset.**

From sklearn.preprocesssing import minmaxscalar

Scalar=minmaxscalar(feature\_range=(0,1))

Scaled\_training\_set

array([[0.0.8581368],

[0.09701243],

[0.09433366],

…..,

[0.95725128],

[0.93576041],

[0.93688146]])

### **5. Creating X\_train and y\_train Data Structures**

x\_train=[]

y\_train=[]

for i in range(60,1258):

x\_train.append(scaled\_trained\_set[i-60:i,0])

y\_train.append(scaled\_trained\_set[i,0])

x\_train=np.array(x\_train)

y\_train=np.array(y\_train)

print(x\_train.shape)

print(y\_train.shape)

(1198,60)  
 (1198,)

### **6. Reshape the Data**

x\_train=np.reshape(x\_train,(x\_train.shape[0],x\_train.shape[1],1))

x\_train.shape

(1198,60,1)

### **7. Building the Model by Importing the Crucial Libraries and Adding Different Layers to LSTM.**

from keras.models import sequential

from keras.layers import LSTM|

from keras.layers import dense

from keras.layers import dropout

regressor =Sequential( )

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

regressor. add(Dropout(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. 2) )

regressor.add(LSTM(units = 50))

regressor. add(Dropout(0 . 2) )

regressor.add(Dense(units=1))

### **8. Fitting the Model**

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

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

epoch1/100

38/38 [====================] – 11s 114s/step – loss:0.1011

epoch2/100

38/38 [====================] – 4s 117s/step – loss:0.0061

epoch3/100

38/38 [====================] – 4s 118s/step – loss:0.0063

### **9. Extracting the Actual Stock Prices of Jan-2017**

dataset\_test=pd.read\_csv(“google\_stock\_price\_test.csv”)

actual\_stock price = dataset\_test.iloc[:,1:2].values

### **10. Preparing the Input for the Model**

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=scalar.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))

### **11. Predicting the Values for Jan 2017 Stock Prices.**

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

predicted\_stock\_price=scalar.inverse\_transform(ptredicted\_stock\_price)

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

plt.plot(actual\_stock\_price,color=’red’,label=’actual google stock price’)

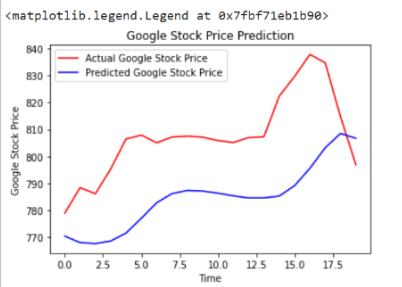
plt.plot(predicted\_stock\_price,color=’blue’,label=’predicted google stock price’)

plt.title(‘google stock price prediction’)

plt.xlabel(‘time’)

plt.ylabel(‘google stock price’)

plt.legend()



**Conclusion:**

The stock market plays a remarkable role in our daily lives. It is a significant factor in a country's GDP growth. In this tutorial, you learned the basics of the stock market and how to perform stock price prediction using machine learning.