Task 7 - Stock Market Prediction using Numerical and Tecxtual Analysis

TSF-GRIP Internship

Data Science & Business Analytics Tasks¶

Level : Advanced

Submitted by : Mohd Tarique Khan

Date : 07/08/2021

Objective: Create a hybrid model for stock price/performance prediction using numerical analysis of historical stock prices, and sentimental analysis of news headlines

Stock to analyze and predict - SENSEX (S&P BSE SENSEX)

Data used: for historical stock prices from finance.yahoo.com

Data used: for textual (news) data from https://bit.ly/36fFPI6

In [1]:
```
# Import Liabraries

# To ignore warnings during the session
import warnings
warnings.filterwarnings('ignore')

# Importing essential libraries
import pandas as pd
import numpy as np

# Importing Data Visualization Libraries
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

Downloading Historical Stock Prices from Yahoo Finance

In [2]:
```
# Importing required libraries
import time
import datetime
```

In [3]:
```
# Creating Variables, we can create data file for many companies but here i will take Microsoft only for asses
# tickers = ['MSFT', 'GOOG', 'AAPL', 'RELIANCE.NS'] # For downloading Multiple Companies data from Yahoo Financ
tickers = ['RELIANCE.NS']
interval = '1d'
period1 = int(time.mktime(datetime.datetime(2016,8,10,23,59).timetuple()))
period2 = int(time.mktime(datetime.datetime(2020,12,31,23,59).timetuple()))
print(period1)
print(period2)
```

```
1470862740
1609448340
```

In [4]:
```
xlwriter = pd.ExcelWriter('Historical_Stock_Prices.xlsx', engine='openpyxl')

for ticker in tickers:
    query_string =f'https://query1.finance.yahoo.com/v7/finance/download/{ticker}?period1={period1}&period2={pe
#                  https://query1.finance.yahoo.com/v7/finance/download/{ticker}?period1={period1}&period2={pe
    df = pd.read_csv(query_string)
    df.to_excel(xlwriter, sheet_name=ticker, index=False)

# xlwriter.save() # For saving an excel file
df.to_csv('Historical_Stock_Prices_csv.csv', index=False) # For saving a csv file
```

In [5]:
```
# reading the datasets into pandas
stock_price = pd.read_csv("C:/Users/IT/TSF GRIP/TSF-GRIP-DSBA-Internship/Historical_Stock_Prices_csv.csv')
```

In [6]:
```
stock_price.head()
```

Out[6]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2016-08-11	496.270844	506.622711	495.106873	502.288818	489.428925	6539201.0
1	2016-08-12	503.725189	516.553589	503.725189	513.036926	499.901855	7552641.0
2	2016-08-16	513.259827	517.494690	502.982239	507.638092	494.641266	6169134.0
3	2016-08-17	507.241852	509.569794	498.871185	501.892548	489.042816	7147050.0
4	2016-08-18	503.229889	507.018982	503.229889	504.245270	491.335297	3576481.0

Loading already downloaded Times of India News Headlines from Harvard Dataverse

In [7]:
```
news_headlines = pd.read_csv('india-news-headlines.csv')
```

In [8]:
```
news_headlines.head()
```

Out[8]:

	publish_date	headline_category	headline_text
0	20010102	unknown	Status quo will not be disturbed at Ayodhya; s...
1	20010102	unknown	Fissures in Hurriyat over Pak visit
2	20010102	unknown	America's unwanted heading for India?
3	20010102	unknown	For bigwigs; it is destination Goa
4	20010102	unknown	Extra buses to clear tourist traffic

In [9]:
```
news_headlines.tail()
```

Out[9]:

	publish_date	headline_category	headline_text
3424062	20201231	city.jodhpur	Covid-19: Despite dip in cases; Rajasthan amon...
3424063	20201231	city.udaipur	Covid-19: Despite dip in cases; Rajasthan amon...
3424064	20201231	city.ajmer	Covid-19: Despite dip in cases; Rajasthan amon...
3424065	20201231	removed	Govt extends deadline for use of FASTag till F...
3424066	20201231	entertainment.bengali.movies.news	Celebs plan to party safely and responsibly on...

Data Visualization & Cleaning

Stock Price Data

In [10]:
```
# checking for null values in both the datasets
stock_price.isna().any()
```

Out[10]:
```
Date         False
Open          True
High          True
Low           True
Close         True
Adj Close     True
Volume        True
dtype: bool
```

In [11]:
```
news_headlines.isna().any()
```

Out[11]:
```
publish_date         False
headline_category    False
headline_text        False
dtype: bool
```

In [12]:
```
stock_price.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1083 entries, 0 to 1082
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       1083 non-null   object
 1   Open       1081 non-null   float64
 2   High       1081 non-null   float64
 3   Low        1081 non-null   float64
 4   Close      1081 non-null   float64
 5   Adj Close  1081 non-null   float64
 6   Volume     1081 non-null   float64
dtypes: float64(6), object(1)
memory usage: 59.4+ KB
```

In [13]:
```
news_headlines.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3424067 entries, 0 to 3424066
Data columns (total 3 columns):
 #   Column             Dtype
---  ------             -----
 0   publish_date       int64
 1   headline_category  object
 2   headline_text      object
dtypes: int64(1), object(2)
memory usage: 78.4+ MB
```

In [14]:
```
# converting the "Date" column datatype from 'object' to 'datetime'
stock_price['Date'] = pd.to_datetime(stock_price['Date']).dt.normalize()

# filtering the important columns required
stock_price = stock_price.filter(['Date', 'Close', 'Open', 'High', 'Low', 'Volume'])

# setting column 'Date' as the index column
stock_price.set_index('Date', inplace= True)

# sorting the data according to the index i.e 'Date'
stock_price = stock_price.sort_index(ascending=True, axis=0)
stock_price
```

Out[14]:

Date	Close	Open	High	Low	Volume
2016-08-11	502.288818	496.270844	506.622711	495.106873	6539201.0
2016-08-12	513.036926	503.725189	516.553589	503.725189	7552641.0
2016-08-16	507.638092	513.259827	517.494690	502.982239	6169134.0
2016-08-17	501.892548	507.241852	509.569794	498.871185	7147050.0
2016-08-18	504.245270	503.229889	507.018982	503.229889	3576481.0
...	...	...	...	...	...
2020-12-24	1994.150024	1947.000000	1998.800049	1945.900024	9588577.0
2020-12-28	2003.300049	2006.699951	2018.000000	1995.300049	7947719.0
2020-12-29	1990.050049	2009.000000	2012.300049	1982.550049	8589407.0
2020-12-30	1995.500000	1995.250000	2007.199951	1975.550049	10173132.0
2020-12-31	1985.300049	1993.500000	2011.900024	1978.599976	8667516.0

1083 rows × 5 columns

In [15]:
```
df1= stock_price[['Close']]
```

In [16]:
```
df1.head()
```

Out[16]:

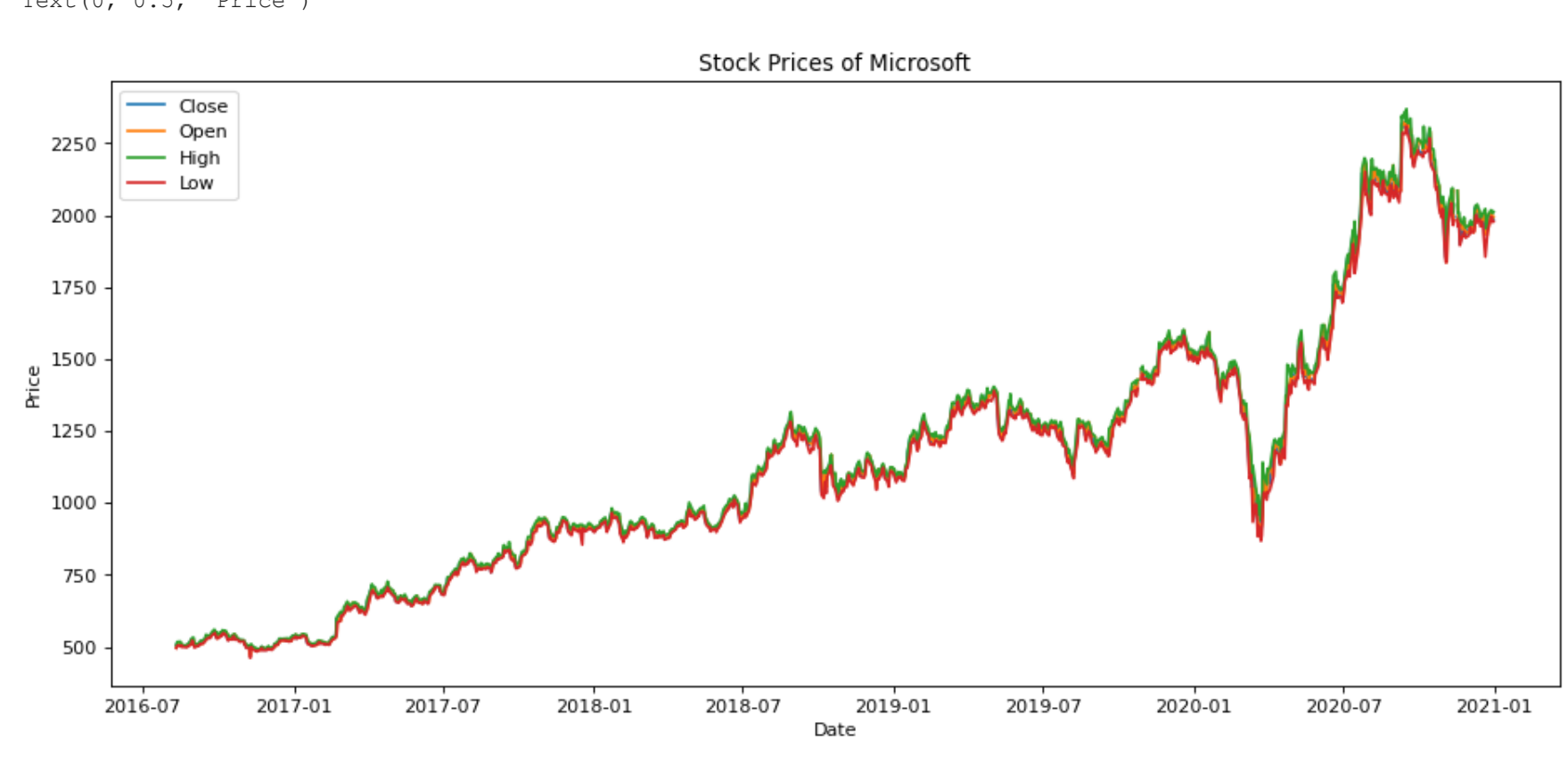
Date	Close
2016-08-11	502.288818
2016-08-12	513.036926
2016-08-16	507.638092
2016-08-17	501.892548
2016-08-18	504.245270

In [17]:
```
from matplotlib.pyplot import figure

figure(figsize=(14, 6), dpi=80)
plt.plot(stock_price["Close"], label="Close")
plt.plot(stock_price["Open"], label="Open")
plt.plot(stock_price["High"], label="High")
plt.plot(stock_price["Low"], label="Low")

plt.legend()
plt.title("Stock Prices of Microsoft")
plt.xlabel("Date")
plt.ylabel("Price")
```

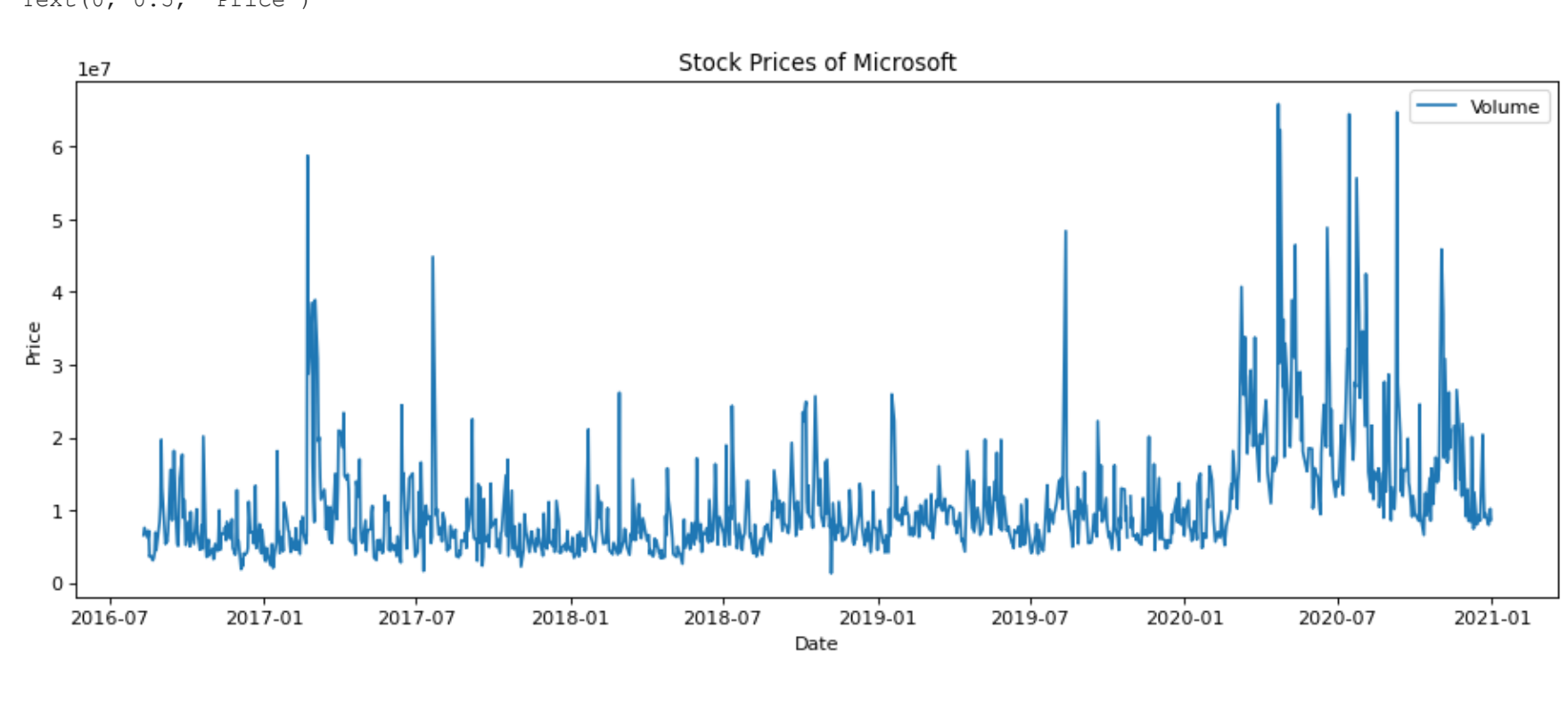
Out[17]: Text(0, 0.5, 'Price')



In [18]:
```
figure(figsize=(14, 5), dpi=80)
plt.plot(stock_price["Volume"], label="Volume")

plt.legend()
plt.title("Stock Prices of Microsoft")
plt.xlabel("Date")
plt.ylabel("Price")
```

Out[18]: Text(0, 0.5, 'Price')



News Data

In [19]:
```
# Coverting the "Date" column datatype from 'object' to 'datetime'
news_headlines["publish_date"] = news_headlines["publish_date"].astype(str)
news_headlines["publish_date"] = news_headlines["publish_date"].apply(lambda x: x[0:4]+'-'+x[4:6]+'-'+x[6:8])
news_headlines["publish_date"] = pd.to_datetime(news_headlines["publish_date"]).dt.normalize()

# filtering the important columns required
news_headlines = news_headlines.filter(["publish_date", "headline_text"])

# grouping the news headlines according to 'Date'
news_headlines = news_headlines.groupby(['publish_date'])['headline_text'].apply(lambda x: ','.join(x)).reset_i

# setting column 'Date' as the index column
news_headlines.set_index('publish_date', inplace= True)

# sorting the data according to the index i.e 'Date'
news_headlines = news_headlines.sort_index(ascending=True, axis=0)
news_headlines
```

Out[19]:

publish_date	headline_text
2001-01-02	Status quo will not be disturbed at Ayodhya; s...
2001-01-03	Powerless north India gropes in the dark,Think...
2001-01-04	The string that pulled Stephen Hawking to Indi...
2001-01-05	Light combat craft takes India into club class...
2001-01-06	Light combat craft takes India into club class...
...	...
2020-12-27	#BigInterview! Dhritiman Chatterjee: Nobody da...
2020-12-28	Horoscope Today; 28 December 2020: Check astro...
2020-12-29	Man recovers charred remains of 'thief' from h...
2020-12-30	Numerology Readings 30 December 2020: Predicti...
2020-12-31	Horoscope Today; 31 December 2020: Check astro...

7262 rows × 1 columns

Concatenating(combining) both Stock Price and News data

In [20]:
```
# concatenating the datasets stock_price and news_headlines
concat_data = pd.concat([stock_price, news_headlines], axis=1)

# dropping the null values if any
concat_data.dropna(axis=0, inplace=True)

# displaying the combined stock_data
concat_data
```

Out[20]:

	Close	Open	High	Low	Volume	headline_text
2016-08-11	502.288818	496.270844	506.622711	495.106873	6539201.0	Watch: Pooja Hegde recreates Hrithik Roshan's ...
2016-08-12	513.036926	503.725189	516.553589	503.725189	7552641.0	Bomb scare at west Delhi Metro station,Mobile ...
2016-08-16	507.638092	513.259827	517.494690	502.982239	6169134.0	Gnocchi: The delightful dumplings,5 delicious ...
2016-08-17	501.892548	507.241852	509.569794	498.871185	7147050.0	Vicky Kaushal: Lesser known facts,Withold 20% ...
2016-08-18	504.245270	503.229889	507.018982	503.229889	3576481.0	Rakshabandhan Special: Bollywood brother-siste...
...	...	...	...	...	...	...
2020-12-24	1994.150024	1947.000000	1998.800049	1945.900024	9588577.0	How to set the mood for sex during cold winter...
2020-12-28	2003.300049	2006.699951	2018.000000	1995.300049	7947719.0	Horoscope Today; 28 December 2020: Check astro...
2020-12-29	1990.050049	2009.000000	2012.300049	1982.550049	8589407.0	Man recovers charred remains of 'thief' from h...
2020-12-30	1995.500000	1995.250000	2007.199951	1975.550049	10173132.0	Numerology Readings 30 December 2020: Predicti...
2020-12-31	1985.300049	1993.500000	2011.900024	1978.599976	8667516.0	Horoscope Today; 31 December 2020: Check astro...

1081 rows × 6 columns

Adding Columns for Sentiment scoring

In [21]:
```
# adding columns to concat_data
concat_data['Compound'] = ''
concat_data['Negative'] = ''
concat_data['Neutral'] = ''
concat_data['Positive'] = ''
concat_data.head()
```

Out[21]:

	Close	Open	High	Low	Volume	headline_text	Compound	Negative	Neutral	Positive
2016-08-11	502.288818	496.270844	506.622711	495.106873	6539201.0	Watch: Pooja Hegde recreates Hrithik Roshan's ...				
2016-08-12	513.036926	503.725189	516.553589	503.725189	7552641.0	Bomb scare at west Delhi Metro station,Mobile ...				
2016-08-16	507.638092	513.259827	517.494690	502.982239	6169134.0	Gnocchi: The delightful dumplings,5 delicious ...				
2016-08-17	501.892548	507.241852	509.569794	498.871185	7147050.0	Vicky Kaushal: Lesser known facts,Withold 20% ...				
2016-08-18	504.245270	503.229889	507.018982	503.229889	3576481.0	Rakshabandhan Special: Bollywood brother-siste...				

In [22]:
```
# Importing Libraries for developing Sentiment Analysis
import nltk
from nltk.classify import NaiveBayesClassifier
from nltk.corpus import subjectivity
from nltk.sentiment import SentimentAnalyzer
from nltk.sentiment.util import *
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import unicodedata
# nltk.download('vader_lexicon')
```

In [23]:
```
# instantiating the Sentiment Analyzer
s_analyzer = SentimentIntensityAnalyzer()

# calculating sentiment scores
concat_data['Compound'] = concat_data['headline_text'].apply(lambda x: s_analyzer.polarity_scores(x)['compound'
concat_data['Negative'] = concat_data['headline_text'].apply(lambda x: s_analyzer.polarity_scores(x)['neg'])
concat_data['Neutral'] = concat_data['headline_text'].apply(lambda x: s_analyzer.polarity_scores(x)['neu'])
concat_data['Positive'] = concat_data['headline_text'].apply(lambda x: s_analyzer.polarity_scores(x)['pos'])

# displaying the stock data
concat_data.head()
```

Out[23]:

	Close	Open	High	Low	Volume	headline_text	Compound	Negative	Neutral	Positive
2016-08-11	502.288818	496.270844	506.622711	495.106873	6539201.0	Watch: Pooja Hegde recreates Hrithik Roshan's ...	-0.9999	0.159	0.750	0.091
2016-08-12	513.036926	503.725189	516.553589	503.725189	7552641.0	Bomb scare at west Delhi Metro station,Mobile ...	-0.9999	0.138	0.779	0.083
2016-08-16	507.638092	513.259827	517.494690	502.982239	6169134.0	Gnocchi: The delightful dumplings,5 delicious ...	-0.9511	0.117	0.764	0.119
2016-08-17	501.892548	507.241852	509.569794	498.871185	7147050.0	Vicky Kaushal: Lesser known facts,Withold 20% ...	-0.9999	0.147	0.759	0.094
2016-08-18	504.245270	503.229889	507.018982	503.229889	3576481.0	Rakshabandhan Special: Bollywood brother-siste...	-0.9990	0.142	0.734	0.125

In [24]:
```
concat_data.shape
```

Out[24]: (1081, 10)

Preparing concatenated data for analysis and saving it

In [25]:
```
# dropping the 'headline_text' which is now not required
concat_data.drop(['headline_text'], inplace=True, axis=1)

# rearranging the columns of the whole concat_data
concat_data = concat_data[['Close', 'Compound', 'Negative', 'Neutral', 'Positive', 'Open', 'High', 'Low', 'Volu

concat_data.head()
```

Out[25]:

	Close	Compound	Negative	Neutral	Positive	Open	High	Low	Volume
2016-08-11	502.288818	-0.9999	0.159	0.750	0.091	496.270844	506.622711	495.106873	6539201.0
2016-08-12	513.036926	-0.9999	0.138	0.779	0.083	503.725189	516.553589	503.725189	7552641.0
2016-08-16	507.638092	-0.9511	0.117	0.764	0.119	513.259827	517.494690	502.982239	6169134.0
2016-08-17	501.892548	-0.9999	0.147	0.759	0.094	507.241852	509.569794	498.871185	7147050.0
2016-08-18	504.245270	-0.9990	0.142	0.734	0.125	503.229889	507.018982	503.229889	3576481.0

In [26]:
```
# Saving concat_data as stockdata_final csv file
concat_data.to_csv('Final_stock_data.csv')
```

Opening the "Final_stock_data.csv" as DataFrame

In [27]:
```
fsdata = pd.read_csv('Final_stock_data.csv')
fsdata.head()
```

Out[27]:

	Unnamed: 0	Close	Compound	Negative	Neutral	Positive	Open	High	Low	Volume
0	2016-08-11	502.288818	-0.9999	0.159	0.750	0.091	496.270844	506.622711	495.106873	6539201.0
1	2016-08-12	513.036926	-0.9999	0.138	0.779	0.083	503.725189	516.553589	503.725189	7552641.0
2	2016-08-16	507.638092	-0.9511	0.117	0.764	0.119	513.259827	517.494690	502.982239	6169134.0
3	2016-08-17	501.892548	-0.9999	0.147	0.759	0.094	507.241852	509.569794	498.871185	7147050.0
4	2016-08-18	504.245270	-0.9990	0.142	0.734	0.125	503.229889	507.018982	503.229889	3576481.0

In [28]:
```
# renaming the index column
fsdata.rename(columns={'Unnamed: 0':'Date'}, inplace = True)

# setting the column 'Date' as the index column
fsdata.set_index('Date', inplace = True)
```

In [29]:
```
fsdata.head()
```

Out[29]:

Date	Close	Compound	Negative	Neutral	Positive	Open	High	Low	Volume
2016-08-11	502.288818	-0.9999	0.159	0.750	0.091	496.270844	506.622711	495.106873	6539201.0
2016-08-12	513.036926	-0.9999	0.138	0.779	0.083	503.725189	516.553589	503.725189	7552641.0
2016-08-16	507.638092	-0.9511	0.117	0.764	0.119	513.259827	517.494690	502.982239	6169134.0
2016-08-17	501.892548	-0.9999	0.147	0.759	0.094	507.241852	509.569794	498.871185	7147050.0
2016-08-18	504.245270	-0.9990	0.142	0.734	0.125	503.229889	507.018982	503.229889	3576481.0

In [30]:
```
fsdata.shape
```

Out[30]: (1081, 9)

In [31]:
```
fsdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1081 entries, 2016-08-11 to 2020-12-31
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Close     1081 non-null   float64
 1   Compound  1081 non-null   float64
 2   Negative  1081 non-null   float64
 3   Neutral   1081 non-null   float64
 4   Positive  1081 non-null   float64
 5   Open      1081 non-null   float64
 6   High      1081 non-null   float64
 7   Low       1081 non-null   float64
 8   Volume    1081 non-null   float64
dtypes: float64(9)
memory usage: 84.5+ KB
```

In [32]:
```
fsdata.describe()
```

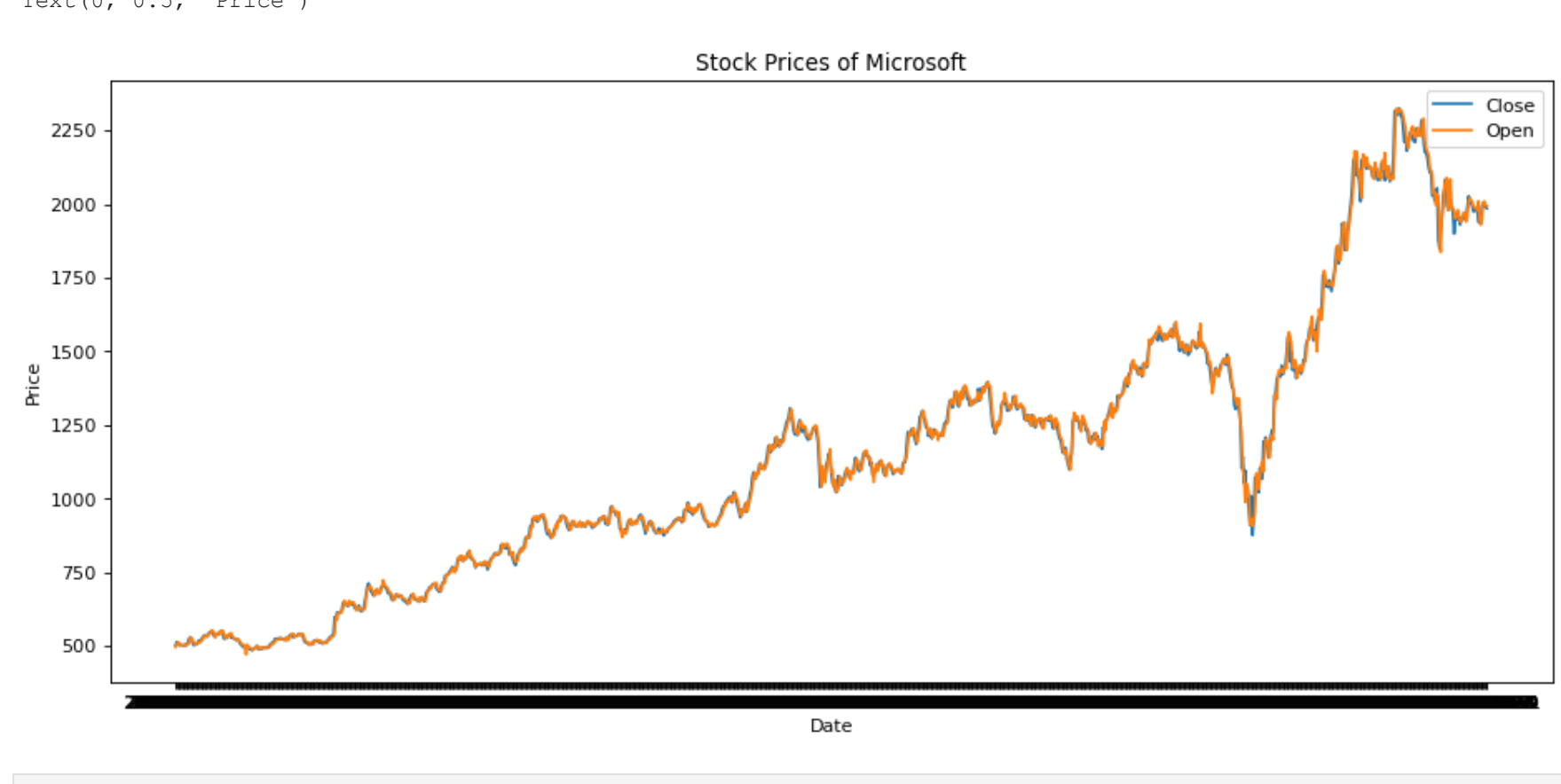
Out[32]:

	Close	Compound	Negative	Neutral	Positive	Open	High	Low	Volume
count	1081.000000	1081.000000	1081.000000	1081.000000	1081.000000	1081.000000	1081.000000	1081.000000	1.081000e+03
mean	1143.425064	-0.979221	0.133335	0.781732	0.084934	1144.553183	1158.242249	1129.895129	1.059065e+07
std	454.456311	0.194421	0.017276	0.022553	0.013185	456.220901	461.834555	448.404941	7.916529e+06
min	484.284454	-1.000000	0.072000	0.728000	0.046000	470.539764	491.689270	460.633667	1.296168e+06
25%	811.161011	-0.999900	0.123000	0.765000	0.077000	813.984253	819.283997	804.474365	5.854289e+06
50%	1100.518188	-0.999800	0.134000	0.779000	0.084000	1099.676147	1118.398682	1090.661621	8.186126e+06
75%	1373.233154	-0.999700	0.144000	0.794000	0.092000	1366.051147	1385.863403	1353.420898	1.255695e+07
max	2324.550049	0.999600	0.199000	0.867000	0.128000	2325.000000	2369.350098	2310.550049	6.584835e+07

In [33]:
```
figure(figsize=(14, 6), dpi=80)
plt.plot(fsdata["Close"], label="Close")
plt.plot(fsdata["Open"], label="Open")

plt.legend()
plt.title("Stock Prices of Microsoft")
plt.xlabel("Date")
plt.ylabel("Price")
```

Out[33]: Text(0, 0.5, 'Price')



In [34]:
```
# Calculating the 150 days rolling mean or moving average
# (D1+D2+D3....D150/150=RM1, D2+D3+D4.....D151/150 =RM2......)
fsdata.rolling(150).mean().head(10)
```

Out[34]:

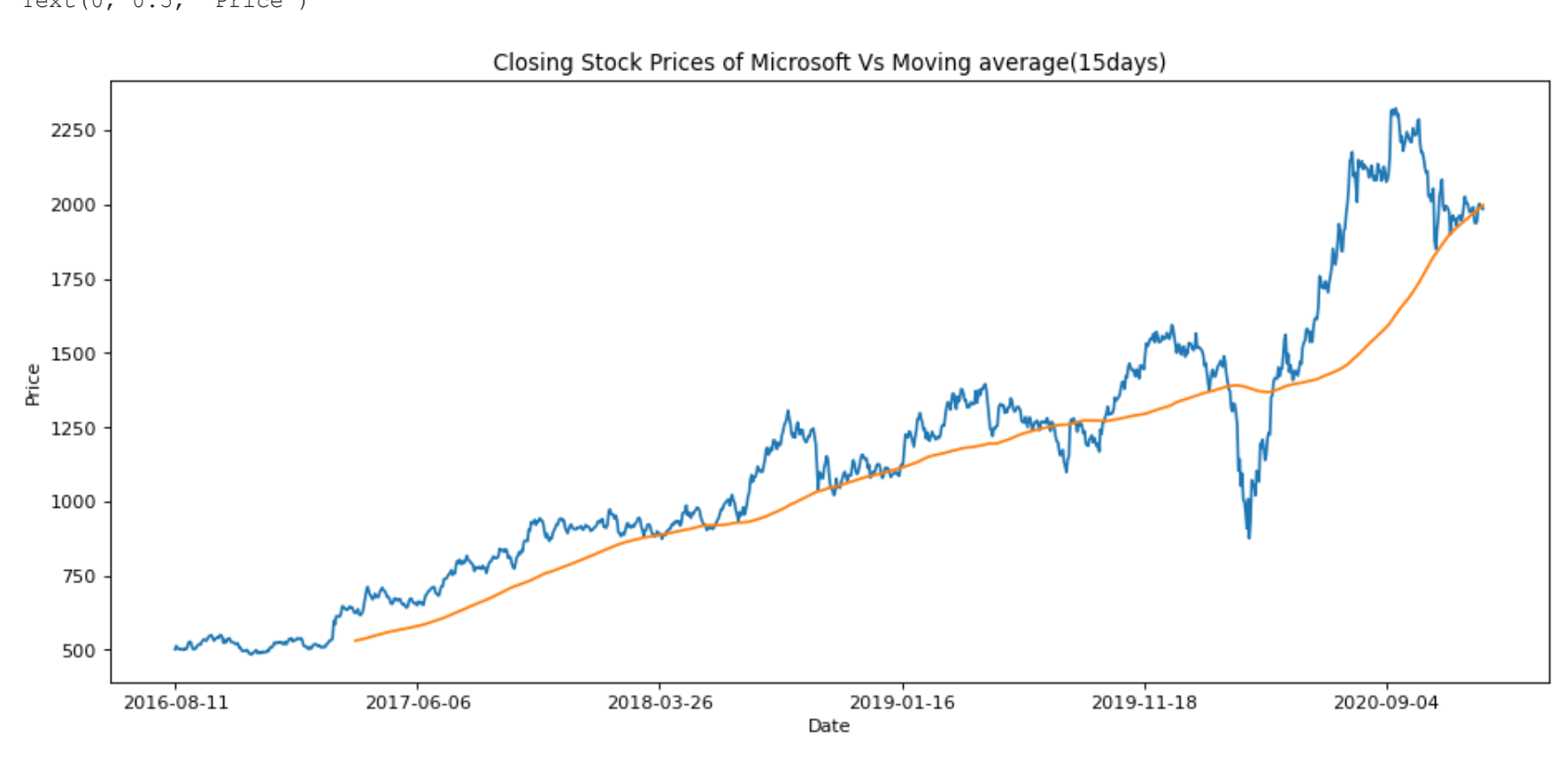
Date	Close	Compound	Negative	Neutral	Positive	Open	High	Low	Volume
2016-08-11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2016-08-12	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2016-08-16	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2016-08-17	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2016-08-18	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2016-08-19	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2016-08-22	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2016-08-23	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2016-08-24	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2016-08-25	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [35]:
```
# Creating a visualization on 15 Days moving average
figure(figsize=(14, 6), dpi=80)

fsdata['Close'].plot()
fsdata.rolling(window = 150).mean()['Close'].plot()

# plt.legend()
plt.title("Closing Stock Prices of Microsoft Vs Moving average(15days)")
plt.xlabel("Date")
plt.ylabel("Price")
```

Out[35]: Text(0, 0.5, 'Price')



Data Preparation for Modelling

In [36]:
```
# dropping nulls
fsdata = fsdata.dropna()
```

In [37]:
```
fsdata.columns
```

Out[37]:
```
Index(['Close', 'Compound', 'Negative', 'Neutral', 'Positive', 'Open', 'High',
       'Low', 'Volume'],
      dtype='object')
```

In [38]:
```
cols = ['Compound', 'Negative', 'Neutral', 'Positive', 'Open', 'High','Low', 'Volume']
len(cols)
```

Out[38]: 8

In [39]:
```
features= fsdata.iloc[:,1:]
features
```

Out[39]:

Date	Compound	Negative	Neutral	Positive	Open	High	Low	Volume
2016-08-11	-0.9999	0.159	0.750	0.091	496.270844	506.622711	495.106873	6539201.0
2016-08-12	-0.9999	0.138	0.779	0.083	503.725189	516.553589	503.725189	7552641.0
2016-08-16	-0.9511	0.117	0.764	0.119	513.259827	517.494690	502.982239	6169134.0
2016-08-17	-0.9999	0.147	0.759	0.094	507.241852	509.569794	498.871185	7147050.0
2016-08-18	-0.9990	0.142	0.734	0.125	503.229889	507.018982	503.229889	3576481.0
...	...	...	...	...	...	...	...	...
2020-12-24	-0.9971	0.082	0.854	0.064	1947.000000	1998.800049	1945.900024	9588577.0
2020-12-28	-0.9997	0.126	0.790	0.084	2006.699951	2018.000000	1995.300049	7947719.0
2020-12-29	-0.9998	0.136	0.774	0.090	2009.000000	2012.300049	1982.550049	8589407.0
2020-12-30	-0.9998	0.126	0.797	0.077	1995.250000	2007.199951	1975.550049	10173132.0
2020-12-31	-0.9996	0.124	0.783	0.094	1993.500000	2011.900024	1978.599976	8667516.0

1081 rows × 8 columns

In [40]:
```
target = fsdata.iloc[:,:1]
target
```

Out[40]:

Date	Close
2016-08-11	502.288818
2016-08-12	513.036926
2016-08-16	507.638092
2016-08-17	501.892548
2016-08-18	504.245270
...	...
2020-12-24	1994.150024
2020-12-28	2003.300049
2020-12-29	1990.050049
2020-12-30	1995.500000
2020-12-31	1985.300049

1081 rows × 1 columns

Scaling the data to manage the sensitivity of LSTM Model.

Applying the MinMaxScaler

In [41]:
```
# Importing Libraries for Data Preprocessing
from sklearn import preprocessing
from sklearn import metrics
from sklearn.preprocessing import MinMaxScaler
```

In [42]:
```
# scaling the feature dataset
scaler_features = preprocessing.MinMaxScaler (feature_range=(-1, 1))
x = np.array(features).reshape((len(features) ,len(cols)))
x = scaler_features.fit_transform(x)

# scaling the target variable
scaler_target = preprocessing.MinMaxScaler (feature_range=(-1, 1))
y = np.array (target).reshape ((len( target), 1))
y = scaler_target.fit_transform (y)

# displaying the scaled feature dataset and the target variable
x, y
```

Out[42]:
```
(array([[-0.99989998,  0.37007874, -0.68345324, ..., -0.98409357,
         -0.96272998, -0.83755675],
        [-0.99989998,  0.03937008, -0.26618705, ..., -0.97351564,
         -0.95341247, -0.80615766],
        [-0.95109022, -0.29133858, -0.48201439, ..., -0.97251323,
         -0.95421569, -0.84902242],
        ...,
        [-0.99979996,  0.00787402, -0.3381295 , ...,  0.61968632,
          0.64538938, -0.77403586],
        [-0.99979996, -0.1496063 , -0.00719424, ...,  0.61425393,
          0.63782147, -0.72496781],
        [-0.99959992, -0.18110236, -0.20863309, ...,  0.61926023,
          0.64111884, -0.77161584]]),
 array([[-0.98043286],
        [-0.96875182],
        [-0.97461927],
        ...,
        [ 0.63646552],
        [ 0.64238852],
        [ 0.63130322]]))
```

Splitting dataset into train and test split, in time-series analysis it cannot be separated by train-test split method as it is date dependent.

In [43]:
```
training_size=int(len(x)*0.70)
test_size=len(x)-training_size
```

In [44]:
```
training_size,test_size
```

Out[44]: (756, 325)

Separating Train and test data features and target.

In [45]:
```
X_train = x[0:training_size,:]
X_test = x[training_size:len(x),:]
y_train = y[0:training_size,:]
y_test = y[training_size:len(x),:]
```

In [46]:
```
X_train.shape,y_train.shape,X_test.shape,y_test.shape
```

```
Out[50]: array([[-2.10506289e-01],
       [-1.97371827e-01],
       [-2.23263847e-01],
       [-2.06845769e-01],
       [-2.22832213e-01],
       [-2.37151984e-01],
       [-2.28269998e-01],
       [-2.56961269e-01],
       [-1.75893737e-01],
       [-1.92204089e-01],
       [-1.49678476e-01],
       [-1.48763546e-01],
       [-1.30192299e-01],
       [-1.17003975e-01],
       [-9.20269528e-02],
       [-1.21471899e-01],
       [-1.14850806e-01],
       [-1.18026829e-01],
       [-1.15073528e-01],
       [-1.00101534e-01],
       [-5.91909305e-02],
       [-7.01183608e-02],
       [-6.43049728e-02],
       [-5.76835796e-02],
       [-4.88555876e-02],
       [-2.28558427e-02],
       [-1.48534484e-03],
       [-3.85396272e-03],
       [-2.72690056e-02],
       [ 2.01542002e-02],
       [ 1.45019660e-02],
       [ 5.30979475e-02],
       [ 6.60709548e-02],
       [ 5.01911024e-02],
       [ 4.21705167e-02],
       [ 4.29779208e-02],
       [ 3.18357138e-02],
       [ 1.84853953e-02],
       [ 4.40007764e-02],
       [ 2.98973220e-02],
       [ 1.08415791e-02],
       [ 5.87500485e-02],
       [ 4.84680364e-02],
       [ 5.71889673e-02],
       [ 4.46667247e-02],
       [ 9.90685635e-02],
       [ 1.39870443e-01],
       [ 1.29051737e-01],
       [ 1.38633405e-01],
       [ 1.54836165e-01],
       [ 1.53436539e-01],
       [ 1.63772882e-01],
       [ 1.75022222e-01],
       [ 1.83639555e-01],
       [ 1.81697177e-01],
       [ 1.73515033e-01],
       [ 1.45308227e-01],
       [ 1.43316514e-01],
       [ 1.47676846e-01],
       [ 1.66732455e-01],
       [ 1.55266798e-01],
       [ 1.55702295e-01],
       [ 1.61995484e-01],
       [ 1.77821473e-01],
       [ 1.60272949e-01],
       [ 1.56074204e-01],
       [ 1.70231387e-01],
       [ 2.06943379e-01],
       [ 1.95262273e-01],
       [ 1.85440554e-01],
       [ 1.38579542e-01],
       [ 1.05151299e-01],
       [ 1.34165480e-01],
       [ 1.36157194e-01],
       [ 1.03697943e-01],
       [ 8.89071097e-02],
       [ 1.26575527e-01],
       [ 1.28567241e-01],
       [ 9.01865771e-02],
       [ 1.15056008e-01],
       [ 1.02728950e-01],
       [ 1.40248214e-01],
       [ 1.39871443e-01],
       [ 1.35618968e-01],
       [ 1.20223612e-01],
       [ 1.14248470e-01],
       [ 1.29374644e-01],
       [ 1.75775897e-01],
       [ 1.23399636e-01],
       [ 1.25068308e-01],
       [ 1.24476220e-01],
       [ 1.17478355e-01],
       [ 1.13772392e-01],
       [ 9.56234934e-02],
       [ 9.81579601e-02],
       [ 6.68783600e-02],
       [ 2.80133320e-02],
       [-6.54535739e-03],
       [-3.46984048e-02],
       [ 9.33436187e-03],
       [ 3.31270753e-02],
       [ 4.31395083e-02],
       [ 1.71396309e-02],
       [ 1.92390702e-02],
       [ 3.77025921e-02],
       [ 5.65988789e-02],
       [ 6.07417621e-02],
       [ 7.52219883e-02],
       [ 6.51558250e-02],
       [ 5.34747187e-02],
       [ 9.26627871e-02],
       [ 7.34455930e-02],
       [ 2.93052335e-02],
       [-1.43161455e-03],
       [-2.77005591e-02],
       [-3.38908662e-02],
       [-5.59027890e-02],
       [-1.09360160e-01],
       [-8.06151576e-02],
       [-8.40063659e-02],
       [-1.14763214e-01],
       [-1.57968441e-01],
       [-3.26832325e-01],
       [-2.84414502e-01],
       [-3.81900112e-01],
       [-3.36360129e-01],
       [-4.32823017e-01],
       [-4.61112784e-01],
       [-4.83638330e-01],
       [-5.38329280e-01],
       [-4.30400676e-01],
       [-5.74556710e-01],
       [-5.10660797e-01],
       [-3.61175698e-01],
       [-3.78455043e-01],
       [-3.79100993e-01],
       [-4.16949232e-01],
       [-3.27262958e-01],
       [-3.63113550e-01],
       [-3.66343302e-01],
       [-2.27839366e-01],
       [-2.42857948e-01],
       [-2.12928637e-01],
       [-2.46087701e-01],
       [-2.88397931e-01],
       [-2.68803830e-01],
       [-2.08568437e-01],
       [-1.87251801e-01],
       [-1.94195846e-01],
       [-5.82758007e-02],
       [-5.04166677e-02],
       [-7.85666311e-02],
       [ 1.29410184e-01],
       [ 1.12183508e-01],
       [ 9.26644923e-03],
       [ 5.19675004e-02],
       [ 1.88084367e-02],
       [ 4.62076742e-02],
       [ 4.63153986e-02],
       [ 9.60541280e-02],
       [ 1.55105346e-01],
       [ 1.71254242e-01],
       [ 1.62324087e-01],
       [ 1.00021105e-01],
       [ 3.42697267e-02],
       [ 3.97552578e-02],
       [ 3.94846183e-02],
       [ 4.87187557e-03],
       [ 3.18264275e-02],
       [ 4.00298181e-02],
       [ 2.94879148e-02],
       [ 2.13369174e-02],
       [ 4.47033198e-02],
       [ 7.37206072e-02],
       [ 6.31892560e-02],
       [ 1.25995644e-01],
       [ 1.42677992e-01],
       [ 1.49144529e-01],
       [ 1.90605963e-01],
       [ 1.92670775e-01],
       [ 1.79411873e-01],
       [ 1.84253930e-01],
       [ 1.82291918e-01],
       [ 1.84851591e-01],
       [ 2.00387159e-01],
       [ 2.28372250e-01],
       [ 2.31795563e-01],
       [ 2.29241611e-01],
       [ 2.73311389e-01],
       [ 3.85795152e-01],
       [ 3.71395057e-01],
       [ 3.43953364e-01],
       [ 3.51506571e-01],
       [ 3.40692967e-01],
       [ 3.66504458e-01],
       [ 3.46398665e-01],
       [ 3.25695076e-01],
       [ 3.62102868e-01],
       [ 3.86827560e-01],
       [ 4.16768942e-01],
       [ 4.86215467e-01],
       [ 4.55404590e-01],
       [ 4.27745592e-01],
       [ 4.56274083e-01],
       [ 5.14743957e-01],
       [ 5.76637144e-01],
       [ 5.57074750e-01],
       [ 4.77738376e-01],
       [ 4.77086322e-01],
       [ 5.51314659e-01],
       [ 5.60226495e-01],
       [ 6.16359724e-01],
       [ 6.51626320e-01],
       [ 7.10096194e-01],
       [ 8.06114783e-01],
       [ 8.17037173e-01],
       [ 8.40403365e-01],
       [ 7.52318201e-01],
       [ 7.65577369e-01],
       [ 7.20203484e-01],
       [ 6.57060318e-01],
       [ 8.10951255e-01],
       [ 7.84704883e-01],
       [ 7.93019060e-01],
       [ 8.06468076e-01],
       [ 7.75532165e-01],
       [ 7.82692967e-01],
       [ 7.85954843e-01],
       [ 7.79923071e-01],
       [ 7.70956974e-01],
       [ 7.66558379e-01],
       [ 7.76119273e-01],
       [ 7.80247668e-01],
       [ 7.52753080e-01],
       [ 7.36233779e-01],
       [ 7.51340187e-01],
       [ 7.36505479e-01],
       [ 7.96496766e-01],
       [ 7.67479269e-01],
       [ 7.73510794e-01],
       [ 7.36988799e-01],
       [ 7.82102390e-01],
       [ 7.86606783e-01],
       [ 7.69109468e-01],
       [ 7.31234394e-01],
       [ 7.37103006e-01],
       [ 7.63675470e-01],
       [ 8.22634351e-01],
       [ 9.88534210e-01],
       [ 9.94783508e-01],
       [ 9.76090408e-01],
       [ 9.93805295e-01],
       [ 1.00000000e+00],
       [ 9.71960516e-01],
       [ 9.79513720e-01],
       [ 9.25336917e-01],
       [ 8.74756763e-01],
       [ 8.98112533e-01],
       [ 8.84207164e-01],
       [ 8.66486557e-01],
       [ 8.02299545e-01],
       [ 9.13599428e-01],
       [ 9.01970725e-01],
       [ 8.92080742e-01],
       [ 8.77897953e-01],
       [ 8.75887534e-01],
       [ 9.27130036e-01],
       [ 9.07295937e-01],
       [ 9.00992446e-01],
       [ 9.04905031e-01],
       [ 9.52343729e-01],
       [ 9.59734020e-01],
       [ 8.71703248e-01],
       [ 8.38338552e-01],
       [ 8.38773166e-01],
       [ 8.16711080e-01],
       [ 7.82694464e-01],
       [ 7.63512290e-01],
       [ 7.70141875e-01],
       [ 6.78904965e-01],
       [ 6.84773709e-01],
       [ 6.59722924e-01],
       [ 6.76514058e-01],
       [ 7.06509702e-01],
       [ 7.14091771e-01],
       [ 6.84693920e-01],
       [ 6.52948358e-01],
       [ 5.98373137e-01],
       [ 6.78959357e-01],
       [ 7.02379810e-01],
       [ 7.39168085e-01],
       [ 6.44236029e-01],
       [ 6.25543128e-01],
       [ 6.43366668e-01],
       [ 6.39943224e-01],
       [ 6.33368032e-01],
       [ 6.18098577e-01],
       [ 5.38055757e-01],
       [ 5.93699865e-01],
       [ 6.08208727e-01],
       [ 5.90548233e-01],
       [ 5.95764792e-01],
       [ 5.70985839e-01],
       [ 5.98264883e-01],
       [ 6.01796582e-01],
       [ 6.08208727e-01],
       [ 5.89407040e-01],
       [ 6.01850842e-01],
       [ 6.40486623e-01],
       [ 6.76568318e-01],
       [ 6.54888719e-01],
       [ 6.53582612e-01],
       [ 6.37824017e-01],
       [ 6.19402684e-01],
       [ 6.21793723e-01],
       [ 6.31628180e-01],
       [ 6.38095717e-01],
       [ 5.81765049e-01],
       [ 5.78484450e-01],
       [ 5.86255259e-01],
       [ 6.40921369e-01],
       [ 6.50865613e-01],
       [ 6.36466551e-01],
       [ 6.42388523e-01],
       [ 6.31303220e-01]])
```

In [51]:
```python
# printing the shape of the training and the test datasets
print('Number of rows and columns in the Training set X:', X_train.shape, 'and y:', y_train.shape)
print('Number of rows and columns in the Test set X:', X_test.shape, 'and y:', y_test.shape)
```

```
Number of rows and columns in the Training set X: (756, 8) and y: (756, 1)
Number of rows and columns in the Test set X: (325, 8) and y: (325, 1)
```

Reshape input to be [samples, time steps, features] which is required for LSTM

In [52]:
```python
X_train = X_train.reshape (X_train.shape + (1,))
X_test = X_test.reshape(X_test.shape + (1,))

# printing the re-shaped feature dataset
print('Shape of Training set X:', X_train.shape)
print('Shape of Test set X:', X_test.shape)
print('Shape of Training set y:', y_train.shape)
print('Shape of Test set y:', y_test.shape)
```

```
Shape of Training set X: (756, 8, 1)
Shape of Test set X: (325, 8, 1)
Shape of Training set y: (756, 1)
Shape of Test set y: (325, 1)
```

In [53]:
```python
# Importing Libraries for Model Building
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Dropout, Activation
```

In [54]:
```python
# setting the seed to achieve consistent and less random predictions at each execution
np.random.seed(2000)
```

In [55]:
```python
# Setting model architecture and compiling it
model=Sequential()
model.add(LSTM(150,return_sequences=True,input_shape=(len(cols),1)))
model.add(Dropout(0.2))
model.add(LSTM(150,return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(150))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer="adam")
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
lstm (LSTM)                  (None, 8, 150)            91200     
_________________________________________________________________
dropout (Dropout)            (None, 8, 150)            0         
_________________________________________________________________
lstm_1 (LSTM)                (None, 8, 150)            180600    
_________________________________________________________________
dropout_1 (Dropout)          (None, 8, 150)            0         
_________________________________________________________________
lstm_2 (LSTM)                (None, 150)               180600    
_________________________________________________________________
dropout_2 (Dropout)          (None, 150)               0         
_________________________________________________________________
dense (Dense)                (None, 1)                 151       
=================================================================
Total params: 452,551
Trainable params: 452,551
Non-trainable params: 0
_________________________________________________________________
```

In [56]:
```python
# Fitting the model using the training dataset
model.fit(X_train, y_train, validation_data=(X_test,y_test), epochs=100, batch_size=64, verbose=1)
```

```
Epoch 1/100
12/12 [==============================] - 3s 234ms/step - loss: 0.1471 - val_loss: 0.3683
Epoch 2/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0694 - val_loss: 0.5459
Epoch 3/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0490 - val_loss: 0.3981
Epoch 4/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0350 - val_loss: 0.2337
Epoch 5/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0174 - val_loss: 0.1120
Epoch 6/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0106 - val_loss: 0.0503
Epoch 7/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0103 - val_loss: 0.0772
Epoch 8/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0073 - val_loss: 0.0735
Epoch 9/100
12/12 [==============================] - 1s 60ms/step - loss: 0.0066 - val_loss: 0.0594
Epoch 10/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0068 - val_loss: 0.0595
Epoch 11/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0057 - val_loss: 0.0586
Epoch 12/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0062 - val_loss: 0.0528
Epoch 13/100
12/12 [==============================] - 1s 60ms/step - loss: 0.0056 - val_loss: 0.0471
Epoch 14/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0058 - val_loss: 0.0442
Epoch 15/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0051 - val_loss: 0.0423
Epoch 16/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0049 - val_loss: 0.0414
Epoch 17/100
12/12 [==============================] - 1s 60ms/step - loss: 0.0051 - val_loss: 0.0375
Epoch 18/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0046 - val_loss: 0.0309
Epoch 19/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0046 - val_loss: 0.0297
Epoch 20/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0041 - val_loss: 0.0254
Epoch 21/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0042 - val_loss: 0.0235
Epoch 22/100
12/12 [==============================] - 1s 63ms/step - loss: 0.0037 - val_loss: 0.0205
Epoch 23/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0038 - val_loss: 0.0209
Epoch 24/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0036 - val_loss: 0.0198
Epoch 25/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0038 - val_loss: 0.0187
Epoch 26/100
12/12 [==============================] - 1s 65ms/step - loss: 0.0041 - val_loss: 0.0190
Epoch 27/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0045 - val_loss: 0.0146
Epoch 28/100
12/12 [==============================] - 1s 67ms/step - loss: 0.0038 - val_loss: 0.0140
Epoch 29/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0035 - val_loss: 0.0113
Epoch 30/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0030 - val_loss: 0.0116
Epoch 31/100
12/12 [==============================] - 1s 66ms/step - loss: 0.0032 - val_loss: 0.0100
Epoch 32/100
12/12 [==============================] - 1s 65ms/step - loss: 0.0030 - val_loss: 0.0084
Epoch 33/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0028 - val_loss: 0.0087
Epoch 34/100
12/12 [==============================] - 1s 63ms/step - loss: 0.0026 - val_loss: 0.0068
Epoch 35/100
12/12 [==============================] - 1s 65ms/step - loss: 0.0026 - val_loss: 0.0075
Epoch 36/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0024 - val_loss: 0.0062
Epoch 37/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0024 - val_loss: 0.0070
Epoch 38/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0024 - val_loss: 0.0064
Epoch 39/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0023 - val_loss: 0.0056
Epoch 40/100
12/12 [==============================] - 1s 63ms/step - loss: 0.0023 - val_loss: 0.0059
Epoch 41/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0022 - val_loss: 0.0057
Epoch 42/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0020 - val_loss: 0.0055
Epoch 43/100
12/12 [==============================] - 1s 65ms/step - loss: 0.0021 - val_loss: 0.0057
Epoch 44/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0020 - val_loss: 0.0056
Epoch 45/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0020 - val_loss: 0.0056
Epoch 46/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0018 - val_loss: 0.0056
Epoch 47/100
12/12 [==============================] - 1s 65ms/step - loss: 0.0021 - val_loss: 0.0056
Epoch 48/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0021 - val_loss: 0.0055
Epoch 49/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0022 - val_loss: 0.0056
Epoch 50/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0020 - val_loss: 0.0057
Epoch 51/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0018 - val_loss: 0.0054
Epoch 52/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0019 - val_loss: 0.0057
Epoch 53/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0018 - val_loss: 0.0055
Epoch 54/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0019 - val_loss: 0.0055
Epoch 55/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0018 - val_loss: 0.0055
Epoch 56/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0018 - val_loss: 0.0067
Epoch 57/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0018 - val_loss: 0.0057
Epoch 58/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0018 - val_loss: 0.0065
Epoch 59/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0019 - val_loss: 0.0059
Epoch 60/100
12/12 [==============================] - 1s 66ms/step - loss: 0.0017 - val_loss: 0.0065
Epoch 61/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0017 - val_loss: 0.0067
Epoch 62/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0016 - val_loss: 0.0056
Epoch 63/100
12/12 [==============================] - 1s 65ms/step - loss: 0.0016 - val_loss: 0.0054
Epoch 64/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0019 - val_loss: 0.0065
Epoch 65/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0017 - val_loss: 0.0058
Epoch 66/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0019 - val_loss: 0.0064
Epoch 67/100
12/12 [==============================] - 1s 63ms/step - loss: 0.0017 - val_loss: 0.0061
Epoch 68/100
12/12 [==============================] - 1s 63ms/step - loss: 0.0014 - val_loss: 0.0079
Epoch 69/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0016 - val_loss: 0.0064
Epoch 70/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0015 - val_loss: 0.0073
Epoch 71/100
12/12 [==============================] - 1s 63ms/step - loss: 0.0016 - val_loss: 0.0061
Epoch 72/100
12/12 [==============================] - 1s 63ms/step - loss: 0.0018 - val_loss: 0.0078
Epoch 73/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0017 - val_loss: 0.0060
Epoch 74/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0014 - val_loss: 0.0070
Epoch 75/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0015 - val_loss: 0.0066
Epoch 76/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0017 - val_loss: 0.0060
Epoch 77/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0014 - val_loss: 0.0082
Epoch 78/100
12/12 [==============================] - 1s 66ms/step - loss: 0.0014 - val_loss: 0.0073
Epoch 79/100
12/12 [==============================] - 1s 66ms/step - loss: 0.0013 - val_loss: 0.0082
Epoch 80/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0014 - val_loss: 0.0086
Epoch 81/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0016 - val_loss: 0.0084
Epoch 82/100
12/12 [==============================] - 1s 65ms/step - loss: 0.0015 - val_loss: 0.0082
Epoch 83/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0014 - val_loss: 0.0093
Epoch 84/100
12/12 [==============================] - 1s 63ms/step - loss: 0.0017 - val_loss: 0.0068
Epoch 85/100
12/12 [==============================] - 1s 63ms/step - loss: 0.0017 - val_loss: 0.0131
Epoch 86/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0014 - val_loss: 0.0072
Epoch 87/100
12/12 [==============================] - 1s 60ms/step - loss: 0.0014 - val_loss: 0.0074
Epoch 88/100
12/12 [==============================] - 1s 60ms/step - loss: 0.0014 - val_loss: 0.0076
Epoch 89/100
12/12 [==============================] - 1s 60ms/step - loss: 0.0013 - val_loss: 0.0090
Epoch 90/100
12/12 [==============================] - 1s 60ms/step - loss: 0.0012 - val_loss: 0.0069
Epoch 91/100
12/12 [==============================] - 1s 60ms/step - loss: 0.0012 - val_loss: 0.0080
Epoch 92/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0013 - val_loss: 0.0088
Epoch 93/100
12/12 [==============================] - 1s 61ms/step - loss: 0.0012 - val_loss: 0.0095
Epoch 94/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0012 - val_loss: 0.0099
Epoch 95/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0013 - val_loss: 0.0089
Epoch 96/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0013 - val_loss: 0.0140
Epoch 97/100
12/12 [==============================] - 1s 62ms/step - loss: 0.0013 - val_loss: 0.0088
Epoch 98/100
12/12 [==============================] - 1s 65ms/step - loss: 0.0015 - val_loss: 0.0114
Epoch 99/100
12/12 [==============================] - 1s 64ms/step - loss: 0.0014 - val_loss: 0.0093
Epoch 100/100
12/12 [==============================] - 1s 65ms/step - loss: 0.0013 - val_loss: 0.0083
```

```
Out[56]: <tensorflow.python.keras.callbacks.History at 0x266c62fab50>
```

In [57]:
```python
import tensorflow as tf
tf.__version__
```

```
Out[57]: '2.3.0'
```

In [58]:
```python
### Lets Do the prediction and check performance metrics
train_predict=model.predict(X_train)
test_predict=model.predict(X_test)
```

In [59]:
```python
train_predict
```

Out[59]:

```
array([[-0.97483623],
       [-0.9882636 ],
       [-0.9865346 ],
       [-0.9761518 ],
       [-0.9751721 ],
       [-0.9908042 ],
       [-0.9906024 ],
       [-1.0066769 ],
       [-0.998734  ],
       [-0.9795304 ],
       [-0.9802616 ],
       [-0.96256274],
       [-0.9685448 ],
       [-0.9600761 ],
       [-0.9529413 ],
       [-0.9752836 ],
       [-0.9831726 ],
       [-0.9875447 ],
       [-0.9847991 ],
       [-0.97656494],
       [-0.9643179 ],
       [-0.9611386 ],
       [-0.9557785 ],
       [-0.9348234 ],
       [-0.9559502 ],
       [-0.9775354 ],
       [-0.9571408 ],
       [-0.940342  ],
       [-0.94421965],
       [-0.9175503 ],
       [-0.93618214],
       [-0.94511116],
       [-0.94711095],
       [-0.9551183 ],
       [-0.9473042 ],
       [-0.9375523 ],
       [-0.9485467 ],
       [-0.9324448 ],
       [-0.94103456],
       [-0.938506  ],
       [-0.93119   ],
       [-0.9653443 ],
       [-0.9579471 ],
       [-0.96491635],
       [-0.93943197],
       [-0.9525579 ],
       [-0.9374565 ],
       [-0.9622635 ],
       [-0.9613672 ],
       [-0.9710115 ],
       [-0.9550902 ],
       [-0.97291946],
       [-0.9648526 ],
       [-0.9654022 ],
       [-1.001542  ],
       [-0.96982664],
       [-0.9690185 ],
       [-0.9928331 ],
       [-1.0303155 ],
       [-0.98124504],
       [-0.99978966],
       [-0.9986796 ],
       [-1.0197377 ],
       [-1.0101289 ],
       [-0.99671894],
       [-0.98325044],
       [-0.99881333],
       [-0.9945849 ],
       [-0.99548316],
       [-1.0291368 ],
       [-0.9976259 ],
       [-0.99472183],
       [-0.99954   ],
       [-0.99743617],
       [-0.9883601 ],
       [-1.0003879 ],
       [-1.0042125 ],
       [-1.002946  ],
       [-0.98879725],
       [-0.98842704],
       [-0.9785792 ],
       [-0.9827107 ],
       [-0.9766821 ],
       [-0.96894693],
       [-0.969471  ],
```

```
[-0.962396  ],
[-0.97066814],
[-0.9642936 ],
[-0.96500856],
[-0.97176677],
[-0.95734614],
[-0.9692872 ],
[-0.97233194],
[-0.9654644 ],
[-0.9596343 ],
[-0.9405297 ],
[-0.9569927 ],
[-0.96026486],
[-0.96320903],
[-0.959239  ],
[-0.9631989 ],
[-0.95060104],
[-0.94903934],
[-0.9589004 ],
[-0.9554166 ],
[-0.94821095],
[-0.95683646],
[-0.9764323 ],
[-0.97492665],
[-0.9707357 ],
[-0.9854936 ],
[-0.9824869 ],
[-0.9796099 ],
[-0.9553481 ],
[-0.97493005],
[-0.9727612 ],
[-0.95787966],
[-0.99476576],
[-0.9755545 ],
[-0.9694839 ],
[-0.98506784],
[-0.9726924 ],
[-0.98852557],
[-0.98402405],
[-0.9740134 ],
[-0.95922077],
[-0.9664389 ],
[-0.9558403 ],
[-0.96176237],
[-0.9606533 ],
[-0.976011  ],
[-0.8513112 ],
[-0.8797508 ],
[-0.81830716],
[-0.86632943],
[-0.87929475],
[-0.87259996],
[-0.81797975],
[-0.8072946 ],
[-0.81237423],
[-0.8182931 ],
[-0.8336141 ],
[-0.84164095],
[-0.8334028 ],
[-0.8329018 ],
[-0.83828735],
[-0.83317316],
[-0.8462059 ],
[-0.8505891 ],
[-0.866487  ],
[-0.8748145 ],
[-0.8499688 ],
[-0.8542701 ],
[-0.83995074],
[-0.8655406 ],
[-0.85179955],
[-0.8148283 ],
[-0.7908667 ],
[-0.7653742 ],
[-0.7473538 ],
[-0.7813522 ],
[-0.7774311 ],
[-0.77240455],
[-0.7769635 ],
[-0.8126859 ],
[-0.8013108 ],
[-0.78854173],
[-0.80530846],
[-0.80549115],
[-0.80288726],
```

```
[-0.7719874 ],
[-0.7411901 ],
[-0.75633055],
[-0.77714235],
[-0.7846554 ],
[-0.7882281 ],
[-0.8035762 ],
[-0.7904671 ],
[-0.8032425 ],
[-0.8316145 ],
[-0.8237153 ],
[-0.8116639 ],
[-0.8036542 ],
[-0.81849813],
[-0.8209202 ],
[-0.8102964 ],
[-0.83688307],
[-0.82441604],
[-0.8291678 ],
[-0.8330789 ],
[-0.83462393],
[-0.8388937 ],
[-0.8301788 ],
[-0.82281023],
[-0.80801916],
[-0.8141436 ],
[-0.80675197],
[-0.8259072 ],
[-0.8226724 ],
[-0.8142252 ],
[-0.82968247],
[-0.83922243],
[-0.832006  ],
[-0.8217199 ],
[-0.82663417],
[-0.85781515],
[-0.7973672 ],
[-0.79541755],
[-0.77564055],
[-0.7883239 ],
[-0.79167485],
[-0.7680311 ],
[-0.7610854 ],
[-0.74699897],
[-0.7548191 ],
[-0.7677236 ],
[-0.7873442 ],
[-0.79646635],
[-0.8028762 ],
[-0.7852101 ],
[-0.7611485 ],
[-0.7632643 ],
[-0.7361439 ],
[-0.7332952 ],
[-0.73264474],
[-0.71403414],
[-0.72009814],
[-0.71882725],
[-0.7063728 ],
[-0.7059453 ],
[-0.7205002 ],
[-0.70923185],
[-0.6112351 ],
[-0.6462869 ],
[-0.64869875],
[-0.6645507 ],
[-0.66064066],
[-0.69289863],
[-0.63507324],
[-0.67592883],
[-0.6518399 ],
[-0.65200347],
[-0.64905983],
[-0.6784737 ],
[-0.6825065 ],
[-0.67920786],
[-0.68012244],
[-0.6843075 ],
[-0.69863963],
[-0.6587251 ],
[-0.6783585 ],
[-0.702377  ],
[-0.6975234 ],
[-0.69940114],
[-0.7051281 ],
```

```
[-0.69204855],
[-0.69468063],
[-0.7040108 ],
[-0.70029324],
[-0.6762236 ],
[-0.67354584],
[-0.6714532 ],
[-0.6524686 ],
[-0.6244449 ],
[-0.64062613],
[-0.6561361 ],
[-0.66904855],
[-0.6646229 ],
[-0.6214977 ],
[-0.62314916],
[-0.6207006 ],
[-0.6375311 ],
[-0.63412327],
[-0.60301137],
[-0.6274422 ],
[-0.6418949 ],
[-0.67418396],
[-0.6676584 ],
[-0.6599449 ],
[-0.668608  ],
[-0.6940747 ],
[-0.67284566],
[-0.66405416],
[-0.6604335 ],
[-0.64640266],
[-0.64171183],
[-0.6388399 ],
[-0.6384369 ],
[-0.6140304 ],
[-0.55282927],
[-0.5741489 ],
[-0.5962799 ],
[-0.55930537],
[-0.57412815],
[-0.5321686 ],
[-0.52593136],
[-0.5347238 ],
[-0.5252262 ],
[-0.53240436],
[-0.53192365],
[-0.518446  ],
[-0.5051586 ],
[-0.51924706],
[-0.5204858 ],
[-0.53486913],
[-0.5359856 ],
[-0.5541513 ],
[-0.56338483],
[-0.57097405],
[-0.58744264],
[-0.58217543],
[-0.5760503 ],
[-0.56401825],
[-0.53782076],
[-0.55415297],
[-0.53116876],
[-0.5211266 ],
[-0.52368385],
[-0.5018891 ],
[-0.5148868 ],
[-0.52266735],
[-0.52544457],
[-0.5184839 ],
[-0.5586611 ],
[-0.56051147],
[-0.54654866],
[-0.5614615 ],
[-0.5424786 ],
[-0.5357208 ],
[-0.5320006 ],
[-0.5474894 ],
[-0.55439115],
[-0.5540915 ],
[-0.5211621 ],
[-0.56494373],
[-0.56535304],
[-0.5404661 ],
[-0.5574237 ],
[-0.54325795],
[-0.54085356],
```

```
[-0.5399728 ],
[-0.52811337],
[-0.52748215],
[-0.57466143],
[-0.55818   ],
[-0.5504036 ],
[-0.54869807],
[-0.55093116],
[-0.5419785 ],
[-0.5282836 ],
[-0.53499466],
[-0.52799976],
[-0.5138695 ],
[-0.51623327],
[-0.5321636 ],
[-0.55336773],
[-0.5407203 ],
[-0.5256083 ],
[-0.53463197],
[-0.46901432],
[-0.4820295 ],
[-0.49813077],
[-0.5080253 ],
[-0.50525856],
[-0.5051073 ],
[-0.48656425],
[-0.52324617],
[-0.56960434],
[-0.5586763 ],
[-0.55078894],
[-0.58719087],
[-0.57423055],
[-0.5512936 ],
[-0.5197971 ],
[-0.5189943 ],
[-0.5286435 ],
[-0.52966744],
[-0.55589783],
[-0.5271504 ],
[-0.5460542 ],
[-0.5315146 ],
[-0.5252626 ],
[-0.5238112 ],
[-0.46252185],
[-0.51794136],
[-0.5250676 ],
[-0.54026425],
[-0.56549674],
[-0.574048  ],
[-0.55581534],
[-0.5327493 ],
[-0.5346788 ],
[-0.5394249 ],
[-0.5464695 ],
[-0.5479533 ],
[-0.57113045],
[-0.59918606],
[-0.5716066 ],
[-0.55581874],
[-0.5624861 ],
[-0.5825455 ],
[-0.5754334 ],
[-0.5697622 ],
[-0.57367074],
[-0.5960345 ],
[-0.5482305 ],
[-0.571695  ],
[-0.56247896],
[-0.54804623],
[-0.52666885],
[-0.536079  ],
[-0.5696296 ],
[-0.5321605 ],
[-0.52810556],
[-0.50489783],
[-0.57049036],
[-0.56164134],
[-0.51880443],
[-0.5406936 ],
[-0.5157849 ],
[-0.49232188],
[-0.44999388],
[-0.44482276],
[-0.4586564 ],
[-0.49341887],
```

```
[-0.47130257],
[-0.49270284],
[-0.5031774 ],
[-0.48984155],
[-0.4920556 ],
[-0.5229641 ],
[-0.4837668 ],
[-0.48767844],
[-0.47224426],
[-0.49182016],
[-0.5209859 ],
[-0.5219484 ],
[-0.5166368 ],
[-0.52035904],
[-0.549005  ],
[-0.556964  ],
[-0.5531799 ],
[-0.5482079 ],
[-0.536079  ],
[-0.5401209 ],
[-0.514856  ],
[-0.51577413],
[-0.5274228 ],
[-0.51260865],
[-0.5170591 ],
[-0.49846333],
[-0.4876702 ],
[-0.47477826],
[-0.4715337 ],
[-0.47344404],
[-0.44908357],
[-0.4457356 ],
[-0.43458453],
[-0.448326  ],
[-0.44266665],
[-0.40198016],
[-0.37856257],
[-0.45861268],
[-0.4503604 ],
[-0.46983498],
[-0.49971423],
[-0.49818859],
[-0.49087802],
[-0.4975734 ],
[-0.487434  ],
[-0.43108216],
[-0.48314592],
[-0.47316995],
[-0.43302295],
[-0.42000115],
[-0.34581915],
[-0.32164744],
[-0.35088745],
[-0.3831534 ],
[-0.36310676],
[-0.37083244],
[-0.32082015],
[-0.3275003 ],
[-0.33576915],
[-0.36759087],
[-0.33505672],
[-0.3332684 ],
[-0.2785238 ],
[-0.24902597],
[-0.24368809],
[-0.26559812],
[-0.26643926],
[-0.2610236 ],
[-0.2626575 ],
[-0.24024892],
[-0.21248378],
[-0.23740909],
[-0.26411888],
[-0.2638298 ],
[-0.22219676],
[-0.23947278],
[-0.2023977 ],
[-0.19093977],
[-0.17556636],
[-0.15980746],
[-0.16253377],
[-0.13799967],
[-0.10787818],
[-0.14145887],
[-0.13881099],
```

```
[-0.17993641],
[-0.19405244],
[-0.19158913],
[-0.17495707],
[-0.14002515],
[-0.1715281 ],
[-0.16901325],
[-0.18054454],
[-0.186404  ],
[-0.20076329],
[-0.2094518 ],
[-0.23094293],
[-0.17993617],
[-0.21063848],
[-0.19980012],
[-0.19436169],
[-0.16958228],
[-0.17599308],
[-0.20649566],
[-0.20915249],
[-0.22489846],
[-0.30122426],
[-0.3252773 ],
[-0.32687464],
[-0.33570206],
[-0.34157166],
[-0.32885122],
[-0.31158525],
[-0.29058492],
[-0.24091628],
[-0.303593  ],
[-0.33528703],
[-0.39655492],
[-0.37427282],
[-0.4011227 ],
[-0.4118161 ],
[-0.37347284],
[-0.3723808 ],
[-0.35935447],
[-0.38297868],
[-0.34899005],
[-0.3841649 ],
[-0.35413906],
[-0.370216  ],
[-0.33687708],
[-0.3667166 ],
[-0.35729358],
[-0.34581017],
[-0.35072267],
[-0.32593483],
[-0.30080432],
[-0.30599922],
[-0.31240067],
[-0.3181612 ],
[-0.34050703],
[-0.33364055],
[-0.29281932],
[-0.2572409 ],
[-0.25276205],
[-0.28944585],
[-0.29952782],
[-0.31703135],
[-0.31193775],
[-0.33048338],
[-0.33674636],
[-0.35185185],
[-0.33343035],
[-0.3218986 ],
[-0.35074762],
[-0.34003717],
[-0.3255577 ],
[-0.2991932 ],
[-0.3331024 ],
[-0.33163932],
[-0.36534536],
[-0.36374402],
[-0.3182337 ],
[-0.31800154],
[-0.31257513],
[-0.31851977],
[-0.3346822 ],
[-0.36124012],
[-0.35480925],
[-0.34319016],
[-0.35026163],
```

```
[-0.34816262],
[-0.3490038 ],
[-0.33871546],
[-0.363733  ],
[-0.31863302],
[-0.30832347],
[-0.2893688 ],
[-0.22535203],
[-0.17656776],
[-0.17510962],
[-0.20081761],
[-0.17127998],
[-0.18727046],
[-0.18254994],
[-0.2079727 ],
[-0.24767242],
[-0.21348023],
[-0.1962116 ],
[-0.14815158],
[-0.15167479],
[-0.12260136],
[-0.1226622 ],
[-0.1542359 ],
[-0.16217943],
[-0.17351255],
[-0.19158424],
[-0.19650686],
[-0.18520465],
[-0.2242416 ],
[-0.20682591],
[-0.18548852],
[-0.18938446],
[-0.19108051],
[-0.2139413 ],
[-0.20761314],
[-0.18903668],
[-0.20625801],
[-0.21131103],
[-0.15441962],
[-0.16687976],
[-0.17482074],
[-0.14359385],
[-0.1006078 ],
[-0.08403498],
[-0.07317685],
[-0.06137813],
[-0.09249358],
[-0.06300573],
[-0.04765236],
[-0.05881524],
[-0.09487528],
[-0.08237198],
[-0.0522115 ],
[-0.05933266],
[-0.04571945],
[-0.06460261],
[-0.03138626],
[-0.06465159],
[-0.06486168],
[-0.08433124],
[-0.09788489],
[-0.09744839],
[-0.10410368],
[-0.0990607 ],
[-0.10882033],
[-0.08582556],
[-0.00537946],
[-0.05687936],
[-0.07063954],
[-0.05550996],
[-0.0334436 ],
[-0.06301397],
[-0.03985966],
[-0.02145409],
[-0.03152151],
[-0.03657   ],
[-0.05625435],
[-0.08469273],
[-0.11377499],
[-0.15173298],
[-0.17993005],
[-0.15947104],
[-0.14862217],
[-0.17493431],
[-0.15690859],
```

```
       [-0.09819542],
       [-0.06052746],
       [-0.07402602],
       [-0.08650881],
       [-0.09297885],
       [-0.10303281],
       [-0.07622878],
       [-0.10329957],
       [-0.09814303],
       [-0.08127877],
       [-0.09589313],
       [-0.0740165 ],
       [-0.08406149],
       [-0.11581768],
       [-0.12329687],
       [-0.18079755],
       [-0.10848285],
       [-0.1104458 ],
       [-0.12299681],
       [-0.1327952 ],
       [-0.15284291],
       [-0.14740877],
       [-0.1615561 ],
       [-0.13659951],
       [-0.1634362 ],
       [-0.18196048],
       [-0.14620087],
       [-0.1335533 ],
       [-0.16082412],
       [-0.18644604],
       [-0.17531335],
       [-0.16803381],
       [-0.16314703],
       [-0.16649526],
       [-0.1777025 ],
       [-0.16948503],
       [-0.1654028 ],
       [-0.17176192],
       [-0.15356806],
       [-0.15688415],
       [-0.16167954],
       [-0.15315829],
       [-0.15678295],
       [-0.15728515],
       [-0.15420076],
       [-0.13893908],
       [-0.17269208],
       [-0.17205605],
       [-0.20149481],
       [-0.2216219 ],
       [-0.23351303],
       [-0.26551732],
       [-0.2602864 ],
       [-0.24465646],
       [-0.26767194],
       [-0.27237314],
       [-0.3130468 ],
       [-0.29650956],
       [-0.26947606],
       [-0.01421402],
       [-0.11647566],
       [-0.13390312],
       [-0.14380977],
       [-0.15508236],
       [-0.17655966],
       [-0.18617947],
       [-0.16456422],
       [-0.15270375],
       [-0.13874783],
       [-0.17558582],
       [-0.19280401],
       [-0.18699968],
       [-0.21671426],
       [-0.21542048],
       [-0.22743428],
       [-0.21667923]], dtype=float32)
```

In [60]:

```
test_predict
```

```
Out[60]: array([[-2.36171201e-01],
       [-2.09867405e-01],
       [-2.24732265e-01],
       [-2.21147945e-01],
       [-2.21012726e-01],
       [-2.36283556e-01],
       [-2.37104729e-01],
       [-2.49181047e-01],
       [-1.58354357e-01],
       [-1.68651313e-01],
       [-1.38292536e-01],
       [-1.51685774e-01],
       [-1.29515380e-01],
       [-1.31048045e-01],
       [-1.00626901e-01],
       [-1.09073654e-01],
       [-1.52935907e-01],
       [-1.16713330e-01],
       [-1.36697471e-01],
       [-1.13968924e-01],
       [-5.28925657e-02],
       [-8.07356536e-02],
       [-8.90637189e-02],
       [-7.99197406e-02],
       [-6.35090770e-02],
       [-5.35912033e-02],
       [ 2.57566571e-04],
       [ 8.75729695e-03],
       [ 2.48129629e-02],
       [-1.32116508e-02],
       [-3.43226381e-02],
       [ 2.52818130e-02],
       [ 2.68405192e-02],
       [ 4.67278846e-02],
       [ 3.94826755e-02],
       [ 2.50465345e-02],
       [ 1.28868669e-02],
       [-4.18348238e-03],
       [ 4.86811623e-03],
       [ 9.64852422e-03],
       [-1.92091244e-02],
       [ 3.81959938e-02],
       [ 2.99125202e-02],
       [ 3.51733603e-02],
       [ 4.16714288e-02],
       [ 8.10274026e-02],
       [ 1.82417989e-01],
       [ 1.00789934e-01],
       [ 1.34776473e-01],
       [ 1.12837642e-01],
       [ 1.70248806e-01],
       [ 1.09475545e-01],
       [ 1.24283537e-01],
       [ 1.36507139e-01],
       [ 1.87906101e-01],
       [ 1.36715189e-01],
       [ 1.20226651e-01],
       [ 1.38316914e-01],
       [ 9.83441174e-02],
       [ 8.29329938e-02],
       [ 1.11468479e-01],
       [ 9.88829062e-02],
       [ 1.17136377e-01],
       [ 1.30531430e-01],
       [ 1.15614846e-01],
       [ 1.24755099e-01],
       [ 1.22124895e-01],
       [ 1.42931521e-01],
       [ 1.75437883e-01],
       [ 1.46040648e-01],
       [ 1.26311004e-01],
       [ 1.28874049e-01],
       [ 8.28236192e-02],
       [ 1.00698113e-01],
       [ 1.10429883e-01],
       [ 7.27534294e-02],
       [ 8.86896998e-02],
       [ 9.21179950e-02],
       [ 8.36370736e-02],
       [ 4.97493707e-02],
       [ 7.06553608e-02],
       [ 9.71463174e-02],
       [ 9.34812874e-02],
       [ 1.16423205e-01],
       [ 8.97788107e-02],
       [ 9.34617370e-02],
       [ 8.35818350e-02],
       [ 1.39426410e-01],
       [ 1.67988052e-01],
       [ 9.69386234e-02],
       [ 7.14453608e-02],
       [ 4.37125725e-02],
       [ 8.36229026e-02],
       [ 5.07295467e-02],
       [ 7.83000290e-02],
       [ 6.55528307e-02],
       [ 2.42632249e-02],
       [ 2.64461823e-02],
       [-4.82963249e-02],
       [ 1.03033371e-02],
       [ 1.22037716e-02],
       [ 2.37392969e-02],
       [-4.41355258e-03],
       [ 2.03976035e-02],
       [ 1.27046593e-02],
       [ 2.39249728e-02],
       [ 4.28546544e-02],
       [ 5.70873655e-02],
       [ 4.32609282e-02],
       [ 1.57654509e-02],
       [ 4.91692030e-02],
       [ 4.46600728e-02],
       [ 2.41684802e-02],
       [ 2.58886926e-02],
       [ 5.59911132e-03],
       [-3.83003689e-02],
       [-2.59079598e-02],
       [-6.46162587e-02],
       [-9.67393517e-02],
       [-1.05582537e-01],
       [-7.31647611e-02],
       [-1.44359112e-01],
       [-1.17964536e-01],
       [-2.36715093e-01],
       [-3.18044275e-01],
       [-3.17088813e-01],
       [-4.93011802e-01],
       [-4.18937385e-01],
       [-4.89501834e-01],
       [-5.36433220e-01],
       [-4.99850124e-01],
       [-5.16771853e-01],
       [-5.02981305e-01],
       [-3.29910189e-01],
       [-3.13317567e-01],
       [-3.30542445e-01],
       [-3.48723895e-01],
       [-3.18185947e-01],
       [-3.15849036e-01],
       [-2.96376199e-01],
       [-2.33964562e-01],
       [-1.84731871e-01],
       [-1.76307455e-01],
       [-2.31575117e-01],
       [-1.94536328e-01],
       [-1.47523278e-01],
       [-1.97700679e-01],
       [-1.79186508e-01],
       [-1.60155568e-01],
       [ 1.52651861e-01],
       [ 1.13400887e-02],
       [ 2.35295712e-02],
       [ 9.96716470e-02],
       [ 1.15173385e-01],
       [ 4.84539755e-02],
       [ 1.44598842e-01],
       [ 8.94428790e-02],
       [ 9.17349160e-02],
       [ 8.67832899e-02],
       [ 1.20777661e-01],
       [ 2.63546050e-01],
       [ 2.48818170e-01],
       [ 2.70062625e-01],
       [ 1.90016180e-01],
       [ 1.01684481e-01],
       [ 1.15358800e-01],
       [ 1.42268583e-01],
       [ 1.26407468e-01],
       [ 8.27005059e-02],
       [ 6.98762387e-02],
       [ 6.90156072e-02],
       [ 5.13250344e-02],
       [ 5.08379377e-02],
       [ 9.28604305e-02],
       [ 9.32220370e-02],
       [ 1.36715859e-01],
       [ 1.17226139e-01],
       [ 1.38762221e-01],
       [ 1.77346766e-01],
       [ 2.06868261e-01],
       [ 1.98793650e-01],
       [ 1.53131276e-01],
       [ 1.41711965e-01],
       [ 1.18610293e-01],
       [ 1.67735457e-01],
       [ 2.51111418e-01],
       [ 2.75979698e-01],
       [ 2.40440726e-01],
       [ 2.60131508e-01],
       [ 5.21208227e-01],
       [ 4.39152092e-01],
       [ 3.55009884e-01],
       [ 3.90150368e-01],
       [ 3.18224519e-01],
       [ 3.33751321e-01],
       [ 3.08426589e-01],
       [ 3.09667915e-01],
       [ 3.14946085e-01],
       [ 3.31287533e-01],
       [ 3.64578098e-01],
       [ 4.46245193e-01],
       [ 4.21099246e-01],
       [ 3.80236745e-01],
       [ 3.92875612e-01],
       [ 4.53388363e-01],
       [ 6.03656769e-01],
       [ 5.76214671e-01],
       [ 7.54649937e-01],
       [ 5.05316913e-01],
       [ 4.97715950e-01],
       [ 4.98327017e-01],
       [ 5.36381662e-01],
       [ 6.32744014e-01],
       [ 6.50027514e-01],
       [ 9.25978184e-01],
       [ 8.28288555e-01],
       [ 7.49760568e-01],
       [ 7.81223953e-01],
       [ 7.55837798e-01],
       [ 7.59898126e-01],
       [ 6.09435916e-01],
       [ 7.93733716e-01],
       [ 8.07988048e-01],
       [ 6.98937237e-01],
       [ 6.62637413e-01],
       [ 6.39321089e-01],
       [ 7.00601995e-01],
       [ 6.39744520e-01],
       [ 6.26884460e-01],
       [ 6.27070248e-01],
       [ 6.08886659e-01],
       [ 6.33118391e-01],
       [ 6.63577318e-01],
       [ 5.94434559e-01],
       [ 6.03191376e-01],
       [ 6.19978189e-01],
       [ 5.75321496e-01],
       [ 6.97942615e-01],
       [ 6.30499840e-01],
       [ 6.22973919e-01],
       [ 7.42805779e-01],
       [ 6.46105647e-01],
       [ 6.13827586e-01],
       [ 5.97314060e-01],
       [ 6.09605908e-01],
       [ 5.76658010e-01],
       [ 6.15890563e-01],
       [ 7.20901668e-01],
       [ 1.11850572e+00],
       [ 8.71453226e-01],
       [ 8.26952159e-01],
       [ 7.61572480e-01],
       [ 8.05222988e-01],
       [ 7.53844619e-01],
       [ 7.35996783e-01],
       [ 7.62250662e-01],
       [ 7.36607909e-01],
       [ 7.72218466e-01],
       [ 6.74262941e-01],
       [ 6.85286522e-01],
       [ 6.79337025e-01],
       [ 7.01950908e-01],
       [ 7.10757434e-01],
       [ 6.93357348e-01],
       [ 6.50812030e-01],
       [ 6.58876657e-01],
       [ 8.20794582e-01],
       [ 6.92217946e-01],
       [ 6.83590055e-01],
       [ 6.63556814e-01],
       [ 7.01429427e-01],
       [ 7.46577322e-01],
       [ 6.89119279e-01],
       [ 6.66614592e-01],
       [ 6.90593064e-01],
       [ 6.36952400e-01],
       [ 6.67381406e-01],
       [ 6.38756156e-01],
       [ 6.10711694e-01],
       [ 6.04871571e-01],
       [ 5.95226407e-01],
       [ 5.74582994e-01],
       [ 5.61894119e-01],
       [ 5.96165836e-01],
       [ 7.04879522e-01],
       [ 6.18520558e-01],
       [ 5.95441043e-01],
       [ 5.30092776e-01],
       [ 6.57877266e-01],
       [ 6.07044339e-01],
       [ 6.19753897e-01],
       [ 6.27431810e-01],
       [ 5.56619406e-01],
       [ 5.96695006e-01],
       [ 6.25520170e-01],
       [ 5.25344133e-01],
       [ 5.25459051e-01],
       [ 5.81226110e-01],
       [ 5.40816844e-01],
       [ 5.06338418e-01],
       [ 5.19658387e-01],
       [ 4.94562626e-01],
       [ 5.51949143e-01],
       [ 4.75617588e-01],
       [ 4.74748075e-01],
       [ 5.10047913e-01],
       [ 4.81806457e-01],
       [ 4.68505681e-01],
       [ 5.73947847e-01],
       [ 5.59931874e-01],
       [ 5.13377190e-01],
       [ 5.43975055e-01],
       [ 5.03012955e-01],
       [ 4.92819786e-01],
       [ 4.90046620e-01],
       [ 5.08861601e-01],
       [ 4.97311831e-01],
       [ 5.28764665e-01],
       [ 4.83484149e-01],
       [ 4.61871445e-01],
       [ 4.72857773e-01],
       [ 5.11073232e-01],
       [ 5.09303451e-01],
       [ 5.15689433e-01],
       [ 5.00236452e-01]], dtype=float32)
```

```
In [61]: ##Transformback to original form
         train_predict=scaler_target.inverse_transform(np.array(train_predict).reshape((len(train_predict),1)))
         test_predict=scaler_target.inverse_transform(np.array(test_predict).reshape((len(test_predict),1)))
```

```
In [62]: # printing the predictions
         print('Train Predictions:')
         train_predict[0:5]
```

```
Train Predictions:
Out[62]: array([[507.43845],
       [495.08347],
       [496.6744 ],
       [506.22794],
       [507.1294 ]], dtype=float32)
```

```
In [63]: print('Test Predictions:')
         test_predict[0:5]
```

```
Test Predictions:
Out[63]: array([[1187.1084],
       [1211.2194],
       [1197.6338],
       [1200.9318],
       [1201.0562]], dtype=float32)
```

```
In [64]: # calculating the training mean-squared-error
         train_loss = model.evaluate(X_train, y_train, batch_size = 1)

         # calculating the test mean-squared-error
         test_loss = model.evaluate(X_test, y_test, batch_size = 1)

         # printing the training and the test mean-squared-errors
         print('Train Loss =', round(train_loss,4))
         print('Test Loss =', round(test_loss,4))
```

```
756/756 [==============================] - 3s 3ms/step - loss: 3.4740e-04
325/325 [==============================] - 1s 3ms/step - loss: 0.0083
Train Loss = 0.0003
Test Loss = 0.0083
```

```
In [65]: import math
         from sklearn.metrics import mean_squared_error
```

```
In [66]: # calculating root mean squared error
         train_root_mean_square_error = math.sqrt(mean_squared_error(y_train,train_predict))
         print('Training Root Mean Square Error =',train_root_mean_square_error)
```

```
Training Root Mean Square Error = 954.4861867164946
```

```
In [67]: # calculating root mean squared error
         test_root_mean_square_error = math.sqrt(mean_squared_error(y_test, test_predict))
         print('Testing Root Mean Square Error =',test_root_mean_square_error)
```

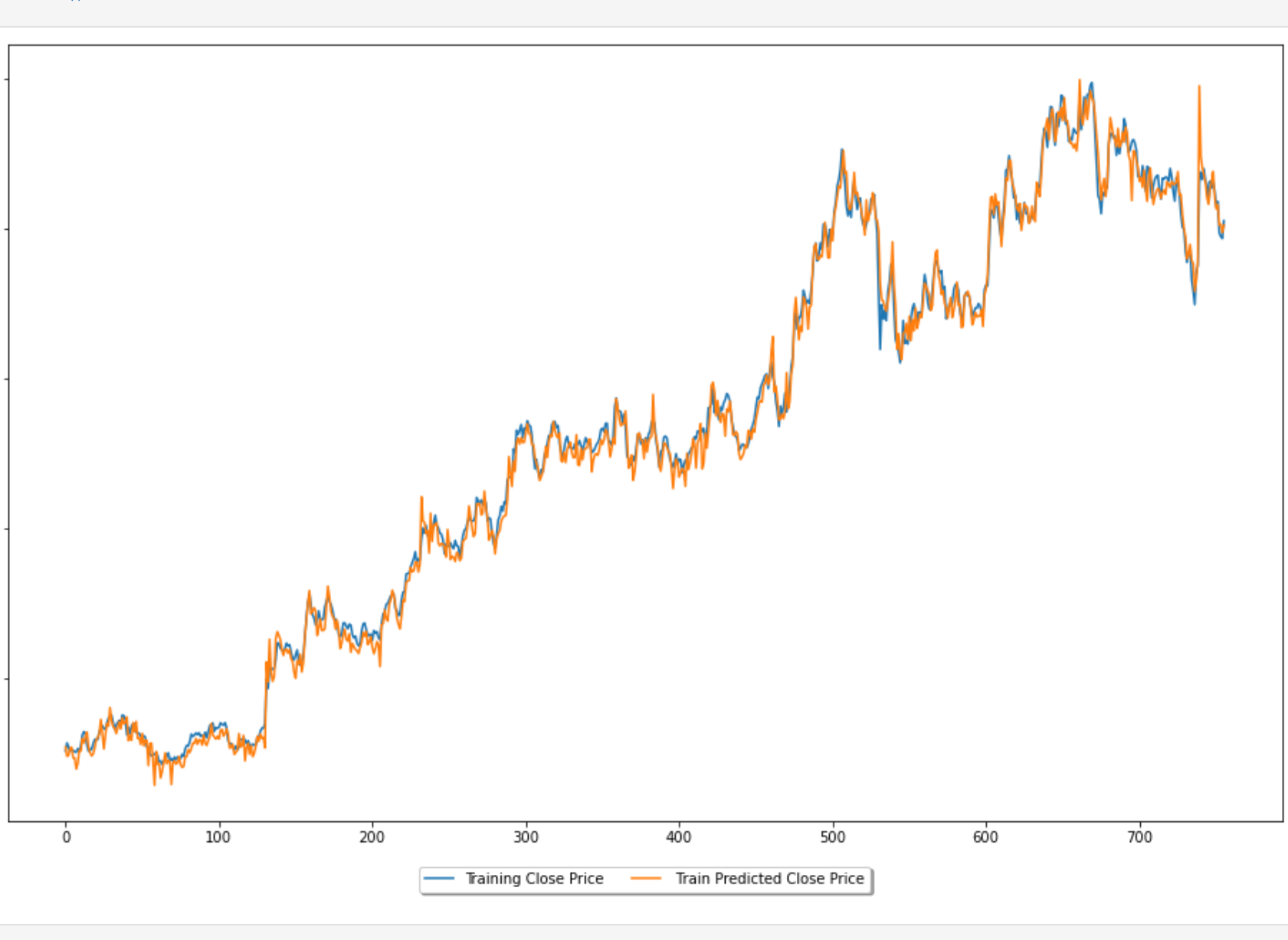
```
Testing Root Mean Square Error = 1662.2179856324092
```

```
In [68]: # unscaling the test feature dataset, x_test
         X_test = scaler_features.inverse_transform(np.array(X_test).reshape((len(X_test), len(cols))))

         # unscaling the test y dataset, y_test
         y_train = scaler_target.inverse_transform(np.array(y_train).reshape((len(y_train), 1)))
         y_test = scaler_target.inverse_transform(np.array(y_test).reshape((len(y_test), 1)))
```

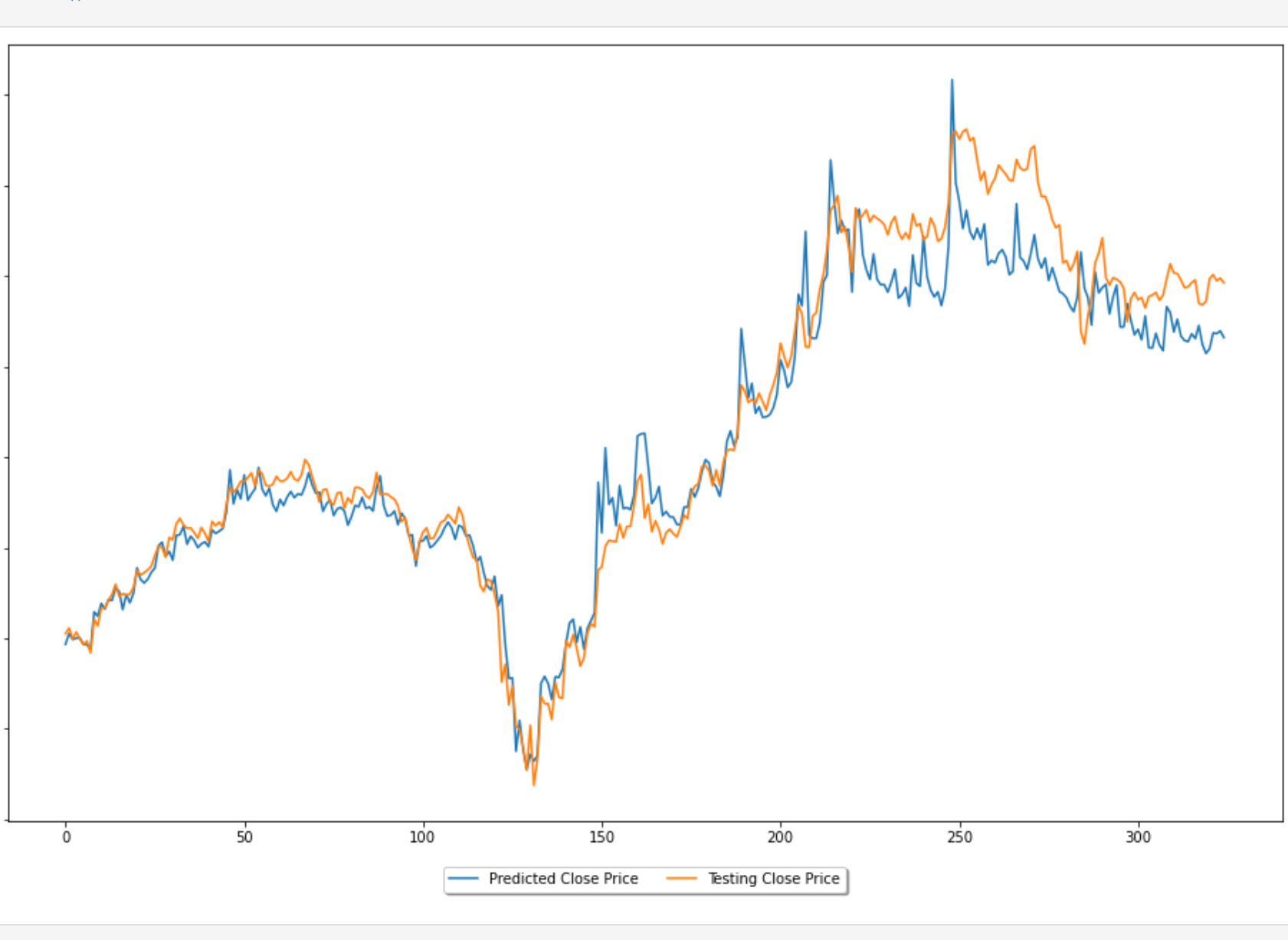
```
In [69]: # plotting
         plt.figure(figsize=(16,10))

         plt.plot([row[0] for row in y_train], label="Training Close Price")
         plt.plot(train_predict, label=" Train Predicted Close Price")
         plt.legend(loc='upper center', bbox_to_anchor=(0.5, -0.05), fancybox=True, shadow=True, ncol=2)
         plt.show()
```




```
In [70]: # plotting
         plt.figure(figsize=(16,10))

         # plt.plot([row[0] for row in y_train], label="Training Close Price")
         plt.plot(test_predict, label="Predicted Close Price")
         plt.plot([row[0] for row in y_test], label="Testing Close Price")
         plt.legend(loc='upper center', bbox_to_anchor=(0.5, -0.05), fancybox=True, shadow=True, ncol=2)
         plt.show()
```




```
In [ ]:
```