import numpy as np   
import pandas as pd   
import matplotlib.pyplot as plt  
from sklearn.preprocessing import MinMaxScaler  
from keras.models import Sequential  
from keras.layers import Dense  
from keras.layers import LSTM  
from keras.layers import Dropout

dataset\_train = pd.read\_csv("trainset.csv")

dataset\_train.head(10)

Date Open High Low Close Adj Close \  
0 2013-01-02 357.385559 361.151062 355.959839 359.288177 359.288177   
1 2013-01-03 360.122742 363.600128 358.031342 359.496826 359.496826   
2 2013-01-04 362.313507 368.339294 361.488861 366.600616 366.600616   
3 2013-01-07 365.348755 367.301056 362.929504 365.001007 365.001007   
4 2013-01-08 365.393463 365.771027 359.874359 364.280701 364.280701   
5 2013-01-09 363.769043 366.789398 361.945892 366.675140 366.675140   
6 2013-01-10 369.014923 370.092896 364.380066 368.344269 368.344269   
7 2013-01-11 368.602600 368.816193 365.771027 367.604095 367.604095   
8 2013-01-14 366.118744 368.701935 358.841095 359.288177 359.288177   
9 2013-01-15 357.340851 365.125214 353.749207 360.122742 360.122742   
  
 Volume   
0 5115500   
1 4666500   
2 5562800   
3 3332900   
4 3373900   
5 4075700   
6 3695100   
7 2587000   
8 5765000   
9 7906300

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

trainset

array([[ 357.385559],  
 [ 360.122742],  
 [ 362.313507],  
 ...,  
 [1057.390015],  
 [1051.599976],  
 [1046.719971]])

sc = MinMaxScaler(feature\_range = (0,1))  
training\_scaled = sc.fit\_transform(trainset)

training\_scaled

array([[0.01011148],  
 [0.01388614],  
 [0.01690727],  
 ...,  
 [0.97543954],  
 [0.9674549 ],  
 [0.96072522]])

x\_train = []  
y\_train = []

for i in range(60,1259):  
 x\_train.append(training\_scaled[i-60:i, 0])  
 y\_train.append(training\_scaled[i,0])

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

x\_train.shape

(1199, 60)

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

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

model.add(Dropout(0.2))

model.add(LSTM(units = 50,return\_sequences = True))  
model.add(Dropout(0.2))

model.add(LSTM(units = 50,return\_sequences = True))  
model.add(Dropout(0.2))

model.add(LSTM(units = 50))  
model.add(Dropout(0.2))

model.add(Dense(units = 1))

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

model.fit(x\_train,y\_train,epochs = 30, batch\_size = 32)

<keras.callbacks.History at 0x7fbc12483fd0>

dataset\_test =pd.read\_csv("testset.csv")

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

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

0 357.385559  
1 360.122742  
2 362.313507  
3 365.348755  
4 365.393463  
 ...   
120 1143.599976  
121 1128.000000  
122 1121.339966  
123 1102.089966  
124 1120.000000  
Name: Open, Length: 1384, dtype: float64

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

inputs = inputs.reshape(-1,1)

inputs = sc.transform(inputs)  
inputs.shape

(185, 1)

x\_test = []  
for i in range(60,185):  
 x\_test.append(inputs[i-60:i,0])

x\_test = np.array(x\_test)  
x\_test.shape

(125, 60)

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

(125, 60, 1)

predicted\_price = model.predict(x\_test)

4/4 [==============================] - 2s 8ms/step

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

plt.plot(real\_stock\_price,color = 'red', label = 'Real Price')  
plt.plot(predicted\_price, color = 'blue', label = 'Predicted Price')  
plt.title('Google Stock Price Prediction')  
plt.xlabel('Time')  
plt.ylabel('Google Stock Price')  
plt.legend()  
plt.show()

