

LSTM Stock

December 22, 2018

```
In [25]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-p
# For example, here's several helpful packages to load in
```

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
```

```
# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the f
```

```
import os
# print(os.listdir("../input"))
```

```
# Any results you write to the current directory are saved as output.
```

```
In [58]: dataset_train = pd.read_csv("N:/Stock Prediction project/DSE3/trainset.csv")
```

```
In [59]: dataset_train
```

```
Out [59]:
```

	Date	Open	High	Low	Close	\
0	2013-01-02	357.385559	361.151062	355.959839	359.288177	
1	2013-01-03	360.122742	363.600128	358.031342	359.496826	
2	2013-01-04	362.313507	368.339294	361.488861	366.600616	
3	2013-01-07	365.348755	367.301056	362.929504	365.001007	
4	2013-01-08	365.393463	365.771027	359.874359	364.280701	
5	2013-01-09	363.769043	366.789398	361.945892	366.675140	
6	2013-01-10	369.014923	370.092896	364.380066	368.344269	
7	2013-01-11	368.602600	368.816193	365.771027	367.604095	
8	2013-01-14	366.118744	368.701935	358.841095	359.288177	
9	2013-01-15	357.340851	365.125214	353.749207	360.122742	
10	2013-01-16	358.865936	359.829651	354.529144	355.284210	
11	2013-01-17	356.536072	357.494843	353.212677	353.361725	
12	2013-01-18	352.884827	354.082031	348.398987	349.978729	
13	2013-01-22	350.053253	350.391052	345.512787	349.164032	
14	2013-01-23	365.617004	372.079987	365.517670	368.354218	
15	2013-01-24	368.225037	375.969666	367.862396	374.668152	
16	2013-01-25	372.959259	376.789337	372.700928	374.399902	

17	2013-01-28	373.451050	375.358643	371.528564	372.939392
18	2013-01-29	370.962250	376.029297	370.857941	374.404846
19	2013-01-30	374.434662	378.016357	374.022339	374.479370
20	2013-01-31	372.830109	376.362122	372.700928	375.403351
21	2013-02-01	376.650238	385.790802	376.600586	385.294037
22	2013-02-04	381.364594	382.745605	376.685028	377.057617
23	2013-02-05	378.105774	383.063538	377.281158	380.395905
24	2013-02-06	377.082428	383.982574	376.799286	382.596588
25	2013-02-07	382.363098	386.888672	380.276672	384.474365
26	2013-02-08	387.544403	390.793274	387.261230	390.147461
27	2013-02-11	386.684998	388.970123	384.375000	388.682007
28	2013-02-12	388.349152	391.404297	387.166840	387.827545
29	2013-02-13	387.544403	390.137543	387.464905	388.900574
...
1229	2017-11-16	1022.520020	1035.920044	1022.520020	1032.500000
1230	2017-11-17	1034.010010	1034.420044	1017.750000	1019.090027
1231	2017-11-20	1020.260010	1022.609985	1017.500000	1018.380005
1232	2017-11-21	1023.309998	1035.109985	1022.655029	1034.489990
1233	2017-11-22	1035.000000	1039.706055	1031.430054	1035.959961
1234	2017-11-24	1035.869995	1043.177979	1035.000000	1040.609985
1235	2017-11-27	1040.000000	1055.459961	1038.439941	1054.209961
1236	2017-11-28	1055.089966	1062.375000	1040.000000	1047.410034
1237	2017-11-29	1042.680054	1044.079956	1015.650024	1021.659973
1238	2017-11-30	1022.369995	1028.489990	1015.000000	1021.409973
1239	2017-12-01	1015.799988	1022.489990	1002.020020	1010.169983
1240	2017-12-04	1012.659973	1016.099976	995.570007	998.679993
1241	2017-12-05	995.940002	1020.609985	988.280029	1005.150024
1242	2017-12-06	1001.500000	1024.969971	1001.140015	1018.380005
1243	2017-12-07	1020.429993	1034.239990	1018.070984	1030.930054
1244	2017-12-08	1037.489990	1042.050049	1032.521973	1037.050049
1245	2017-12-11	1035.500000	1043.800049	1032.050049	1041.099976
1246	2017-12-12	1039.630005	1050.310059	1033.689941	1040.479980
1247	2017-12-13	1046.119995	1046.665039	1038.380005	1040.609985
1248	2017-12-14	1045.000000	1058.500000	1043.109985	1049.150024
1249	2017-12-15	1054.609985	1067.619995	1049.500000	1064.189941
1250	2017-12-18	1066.079956	1078.489990	1062.000000	1077.140015
1251	2017-12-19	1075.199951	1076.839966	1063.550049	1070.680054
1252	2017-12-20	1071.780029	1073.380005	1061.520020	1064.949951
1253	2017-12-21	1064.949951	1069.329956	1061.793945	1063.630005
1254	2017-12-22	1061.109985	1064.199951	1059.439941	1060.119995
1255	2017-12-26	1058.069946	1060.119995	1050.199951	1056.739990
1256	2017-12-27	1057.390015	1058.369995	1048.050049	1049.369995
1257	2017-12-28	1051.599976	1054.750000	1044.770020	1048.140015
1258	2017-12-29	1046.719971	1049.699951	1044.900024	1046.400024

	Adj Close	Volume
0	359.288177	5115500
1	359.496826	4666500

2	366.600616	5562800
3	365.001007	3332900
4	364.280701	3373900
5	366.675140	4075700
6	368.344269	3695100
7	367.604095	2587000
8	359.288177	5765000
9	360.122742	7906300
10	355.284210	4073100
11	353.361725	4451700
12	349.978729	6495500
13	349.164032	7634000
14	368.354218	11895000
15	374.668152	6809200
16	374.399902	4480700
17	372.939392	3275300
18	374.404846	3516800
19	374.479370	3488500
20	375.403351	3289500
21	385.294037	7540700
22	377.057617	6120500
23	380.395905	3765600
24	382.596588	4183200
25	384.474365	5717300
26	390.147461	6079300
27	388.682007	4363700
28	387.827545	3742100
29	388.900574	2411800
...
1229	1032.500000	1129700
1230	1019.090027	1397100
1231	1018.380005	953500
1232	1034.489990	1097000
1233	1035.959961	746300
1234	1040.609985	537000
1235	1054.209961	1307900
1236	1047.410034	1424400
1237	1021.659973	2459400
1238	1021.409973	1724000
1239	1010.169983	1909600
1240	998.679993	1906400
1241	1005.150024	2067300
1242	1018.380005	1272000
1243	1030.930054	1458200
1244	1037.050049	1290800
1245	1041.099976	1192800
1246	1040.479980	1279500
1247	1040.609985	1282700

1248	1049.150024	1558700
1249	1064.189941	3275900
1250	1077.140015	1554600
1251	1070.680054	1338700
1252	1064.949951	1268600
1253	1063.630005	995700
1254	1060.119995	755100
1255	1056.739990	760600
1256	1049.369995	1271900
1257	1048.140015	837100
1258	1046.400024	887500

[1259 rows x 7 columns]

```
In [60]: trainset = dataset_train.iloc[:,1:2].values
```

```
In [61]: trainset
```

```
Out[61]: array([[ 357.385559],
                 [ 360.122742],
                 [ 362.313507],
                 ...,
                 [1057.390015],
                 [1051.599976],
                 [1046.719971]])
```

```
In [62]: from sklearn.preprocessing import MinMaxScaler
         sc = MinMaxScaler(feature_range = (0,1))
         training_scaled = sc.fit_transform(trainset)
```

```
In [63]: training_scaled
```

```
Out[63]: array([[0.01011148],
                 [0.01388614],
                 [0.01690727],
                 ...,
                 [0.97543954],
                 [0.9674549 ],
                 [0.96072522]])
```

```
In [64]: x_train = []
         y_train = []
```

```
In [65]: for i in range(60,1259):
         x_train.append(training_scaled[i-60:i, 0])
         y_train.append(training_scaled[i,0])
         x_train,y_train = np.array(x_train),np.array(y_train)
```

```
In [66]: x_train.shape
```

```
Out[66]: (1199, 60)
```

```
In [67]: x_train = np.reshape(x_train, (x_train.shape[0],x_train.shape[1],1))
```

```
In [68]: from keras.models import Sequential
         from keras.layers import Dense
         from keras.layers import LSTM
         from keras.layers import Dropout
```

```
In [69]: regressor = Sequential()
         regressor.add(LSTM(units = 50,return_sequences = True,input_shape = (x_train.shape[1]
```

```
In [70]: regressor.add(Dropout(0.2))
```

```
In [71]: regressor.add(LSTM(units = 50,return_sequences = True))
         regressor.add(Dropout(0.2))
```

```
In [72]: regressor.add(LSTM(units = 50,return_sequences = True))
         regressor.add(Dropout(0.2))
```

```
In [73]: regressor.add(LSTM(units = 50))
         regressor.add(Dropout(0.2))
```

```
In [74]: regressor.add(Dense(units = 1))
```

```
In [75]: regressor.compile(optimizer = 'adam',loss = 'mean_squared_error')
```

```
In [76]: regressor.fit(x_train,y_train,epochs = 100, batch_size = 32)
```

```
Epoch 1/100
```

```
1199/1199 [=====] - 7s 6ms/step - loss: 0.0381
```

```
Epoch 2/100
```

```
1199/1199 [=====] - 3s 3ms/step - loss: 0.0042A: 0s - loss: 0.00
```

```
Epoch 3/100
```

```
1199/1199 [=====] - 3s 3ms/step - loss: 0.0035
```

```
Epoch 4/100
```

```
1199/1199 [=====] - 3s 3ms/step - loss: 0.0034A: 0s - lo
```

```
Epoch 5/100
```

```
1199/1199 [=====] - 4s 3ms/step - loss: 0.0034
```

```
Epoch 6/100
```

```
1199/1199 [=====] - 4s 3ms/step - loss: 0.0038
```

```
Epoch 7/100
```

```
1199/1199 [=====] - 4s 3ms/step - loss: 0.0034
```

```
Epoch 8/100
```

```
1199/1199 [=====] - 4s 3ms/step - loss: 0.0033
```

```
Epoch 9/100
```

```
1199/1199 [=====] - 3s 3ms/step - loss: 0.0032
```

```
Epoch 10/100
```

```
1199/1199 [=====] - 4s 3ms/step - loss: 0.0030
```

```
Epoch 11/100
```

```

1199/1199 [=====] - 3s 3ms/step - loss: 0.0032
Epoch 12/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0034
Epoch 13/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0027
Epoch 14/100
1199/1199 [=====] - 4s 3ms/step - loss: 0.0030A: 1s - loss: 0 - ETA: 1s
Epoch 15/100
1199/1199 [=====] - 4s 4ms/step - loss: 0.0027
Epoch 16/100
1199/1199 [=====] - 4s 3ms/step - loss: 0.0026
Epoch 17/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0026
Epoch 18/100
1199/1199 [=====] - 4s 3ms/step - loss: 0.0028
Epoch 19/100
1199/1199 [=====] - 4s 3ms/step - loss: 0.0026
Epoch 20/100
1199/1199 [=====] - 4s 3ms/step - loss: 0.0025
Epoch 21/100
1199/1199 [=====] - 4s 3ms/step - loss: 0.0027
Epoch 22/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0027
Epoch 23/100
1199/1199 [=====] - 4s 3ms/step - loss: 0.0024
Epoch 24/100
1199/1199 [=====] - 4s 3ms/step - loss: 0.0022
Epoch 25/100
1199/1199 [=====] - 4s 3ms/step - loss: 0.0022
Epoch 26/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0024
Epoch 27/100
1199/1199 [=====] - 4s 3ms/step - loss: 0.0021
Epoch 28/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0020
Epoch 29/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0022
Epoch 30/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0019
Epoch 31/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0020
Epoch 32/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0021
Epoch 33/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0022
Epoch 34/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0020
Epoch 35/100

```

```

1199/1199 [=====] - 3s 3ms/step - loss: 0.0022
Epoch 36/100
1199/1199 [=====] - 4s 3ms/step - loss: 0.0018A: 2s - - ETA: 1s -
Epoch 37/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0019
Epoch 38/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0024
Epoch 39/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0021
Epoch 40/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0019
Epoch 41/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0016
Epoch 42/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0017
Epoch 43/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0018
Epoch 44/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0021
Epoch 45/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0020
Epoch 46/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0019
Epoch 47/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0017
Epoch 48/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0016
Epoch 49/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0019
Epoch 50/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0016
Epoch 51/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0017A: 0s - loss: 0.001
Epoch 52/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0018
Epoch 53/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0016
Epoch 54/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0016
Epoch 55/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0018
Epoch 56/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0016
Epoch 57/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0017
Epoch 58/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0016
Epoch 59/100

```

```

1199/1199 [=====] - 3s 3ms/step - loss: 0.0016
Epoch 60/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0015
Epoch 61/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0014
Epoch 62/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0016
Epoch 63/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0015
Epoch 64/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0015
Epoch 65/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0014
Epoch 66/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0014
Epoch 67/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0016
Epoch 68/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0013
Epoch 69/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0013
Epoch 70/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0014
Epoch 71/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0014
Epoch 72/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0013
Epoch 73/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0014
Epoch 74/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0013
Epoch 75/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0014
Epoch 76/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0014
Epoch 77/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0014
Epoch 78/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0012
Epoch 79/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0012
Epoch 80/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0012
Epoch 81/100
1199/1199 [=====] - 4s 3ms/step - loss: 0.0012TA: 2s - loss: 0
Epoch 82/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0011
Epoch 83/100

```



```

1199/1199 [=====] - 3s 3ms/step - loss: 0.0011
Epoch 84/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0012
Epoch 85/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0013
Epoch 86/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0012
Epoch 87/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0011
Epoch 88/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0013
Epoch 89/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0011
Epoch 90/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0012
Epoch 91/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0010
Epoch 92/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0011
Epoch 93/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0011
Epoch 94/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0011
Epoch 95/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0010
Epoch 96/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0011
Epoch 97/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0011
Epoch 98/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0010
Epoch 99/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0010
Epoch 100/100
1199/1199 [=====] - 3s 3ms/step - loss: 0.0010

```

```
Out[76]: <keras.callbacks.History at 0x27e6f7861d0>
```

```
In [77]: dataset_test =pd.read_csv("N:/Stock Prediction project/DSE3/testset.csv")
```

```
In [78]: real_stock_price = dataset_test.iloc[:,1:2].values
```

```
In [79]: dataset_total = pd.concat((dataset_train['Open'],dataset_test['Open']),axis = 0)
dataset_total
```

```
Out[79]: 0      357.385559
         1      360.122742
         2      362.313507
```

3	365.348755
4	365.393463
5	363.769043
6	369.014923
7	368.602600
8	366.118744
9	357.340851
10	358.865936
11	356.536072
12	352.884827
13	350.053253
14	365.617004
15	368.225037
16	372.959259
17	373.451050
18	370.962250
19	374.434662
20	372.830109
21	376.650238
22	381.364594
23	378.105774
24	377.082428
25	382.363098
26	387.544403
27	386.684998
28	388.349152
29	387.544403
	...
95	1061.859985
96	1074.060059
97	1083.560059
98	1065.130005
99	1079.000000
100	1079.020020
101	1064.890015
102	1063.030029
103	1067.560059
104	1099.349976
105	1122.329956
106	1140.989990
107	1142.170044
108	1131.319946
109	1118.180054
110	1118.599976
111	1131.069946
112	1141.119995
113	1143.849976
114	1148.859985

```

115    1143.650024
116    1158.500000
117    1175.310059
118    1174.849976
119    1159.140015
120    1143.599976
121    1128.000000
122    1121.339966
123    1102.089966
124    1120.000000
Name: Open, Length: 1384, dtype: float64

```

```

In [80]: inputs = dataset_total[len(dataset_total) - len(dataset_test)-60:].values
         inputs

```

```

Out[80]: array([ 955.48999 ,  966.700012,  980.        ,  980.        ,  973.719971,
                  987.450012,  992.        ,  992.099976,  990.289978,  991.77002 ,
                  986.        ,  989.440002,  989.52002 ,  970.        ,  968.369995,
                  980.        , 1009.190002, 1014.        , 1015.219971, 1017.210022,
                  1021.76001 , 1022.109985, 1028.98999 , 1027.27002 , 1030.52002 ,
                  1033.98999 , 1026.459961, 1023.419983, 1022.590027, 1019.210022,
                  1022.52002 , 1034.01001 , 1020.26001 , 1023.309998, 1035.        ,
                  1035.869995, 1040.        , 1055.089966, 1042.680054, 1022.369995,
                  1015.799988, 1012.659973,  995.940002, 1001.5        , 1020.429993,
                  1037.48999 , 1035.5        , 1039.630005, 1046.119995, 1045.        ,
                  1054.609985, 1066.079956, 1075.199951, 1071.780029, 1064.949951,
                  1061.109985, 1058.069946, 1057.390015, 1051.599976, 1046.719971,
                  1048.339966, 1064.310059, 1088.        , 1094.        , 1102.22998 ,
                  1109.400024, 1097.099976, 1106.300049, 1102.410034, 1132.51001 ,
                  1126.219971, 1131.410034, 1131.829956, 1137.48999 , 1159.849976,
                  1177.329956, 1172.530029, 1175.079956, 1176.47998 , 1167.829956,
                  1170.569946, 1162.609985, 1122.        , 1090.599976, 1027.180054,
                  1081.540039, 1055.410034, 1017.25        , 1048.        , 1045.        ,
                  1048.949951, 1079.069946, 1088.410034, 1090.569946, 1106.469971,
                  1116.189941, 1112.640015, 1127.800049, 1141.23999 , 1123.030029,
                  1107.869995, 1053.079956, 1075.140015, 1099.219971, 1089.189941,
                  1115.319946, 1136.        , 1163.849976, 1170.        , 1145.209961,
                  1149.959961, 1154.140015, 1120.01001 , 1099.        , 1092.73999 ,
                  1081.880005, 1047.030029, 1046.        , 1063.        ,  998.        ,
                  1011.630005, 1022.820007, 1013.909973,  993.409973, 1041.329956,
                  1020.        , 1016.799988, 1026.439941, 1027.98999 , 1025.040039,
                  1040.880005, 1037.        , 1051.369995, 1077.430054, 1069.400024,
                  1082.        , 1077.859985, 1052.        , 1025.52002 , 1029.51001 ,
                  1046.        , 1030.01001 , 1013.659973, 1028.099976, 1019.        ,
                  1016.900024, 1049.22998 , 1058.540039, 1058.099976, 1086.030029,
                  1093.599976, 1100.        , 1090.        , 1077.310059, 1079.890015,
                  1061.859985, 1074.060059, 1083.560059, 1065.130005, 1079.        ,
                  1079.02002 , 1064.890015, 1063.030029, 1067.560059, 1099.349976,

```

```

1122.329956, 1140.98999 , 1142.170044, 1131.319946, 1118.180054,
1118.599976, 1131.069946, 1141.119995, 1143.849976, 1148.859985,
1143.650024, 1158.5      , 1175.310059, 1174.849976, 1159.140015,
1143.599976, 1128.      , 1121.339966, 1102.089966, 1120.      ])
```

```
In [81]: inputs = inputs.reshape(-1,1)
```

```
In [82]: inputs
```

```
Out[82]: array([[ 955.48999 ],
 [ 966.700012],
 [ 980.      ],
 [ 980.      ],
 [ 973.719971],
 [ 987.450012],
 [ 992.      ],
 [ 992.099976],
 [ 990.289978],
 [ 991.77002 ],
 [ 986.      ],
 [ 989.440002],
 [ 989.52002 ],
 [ 970.      ],
 [ 968.369995],
 [ 980.      ],
 [1009.190002],
 [1014.      ],
 [1015.219971],
 [1017.210022],
 [1021.76001 ],
 [1022.109985],
 [1028.98999 ],
 [1027.27002 ],
 [1030.52002 ],
 [1033.98999 ],
 [1026.459961],
 [1023.419983],
 [1022.590027],
 [1019.210022],
 [1022.52002 ],
 [1034.01001 ],
 [1020.26001 ],
 [1023.309998],
 [1035.      ],
 [1035.869995],
 [1040.      ],
 [1055.089966],
 [1042.680054],
```

[1022.369995],
[1015.799988],
[1012.659973],
[995.940002],
[1001.5],
[1020.429993],
[1037.48999],
[1035.5],
[1039.630005],
[1046.119995],
[1045.],
[1054.609985],
[1066.079956],
[1075.199951],
[1071.780029],
[1064.949951],
[1061.109985],
[1058.069946],
[1057.390015],
[1051.599976],
[1046.719971],
[1048.339966],
[1064.310059],
[1088.],
[1094.],
[1102.22998],
[1109.400024],
[1097.099976],
[1106.300049],
[1102.410034],
[1132.51001],
[1126.219971],
[1131.410034],
[1131.829956],
[1137.48999],
[1159.849976],
[1177.329956],
[1172.530029],
[1175.079956],
[1176.47998],
[1167.829956],
[1170.569946],
[1162.609985],
[1122.],
[1090.599976],
[1027.180054],
[1081.540039],
[1055.410034],

[1017.25],
[1048.],
[1045.],
[1048.949951],
[1079.069946],
[1088.410034],
[1090.569946],
[1106.469971],
[1116.189941],
[1112.640015],
[1127.800049],
[1141.23999],
[1123.030029],
[1107.869995],
[1053.079956],
[1075.140015],
[1099.219971],
[1089.189941],
[1115.319946],
[1136.],
[1163.849976],
[1170.],
[1145.209961],
[1149.959961],
[1154.140015],
[1120.01001],
[1099.],
[1092.73999],
[1081.880005],
[1047.030029],
[1046.],
[1063.],
[998.],
[1011.630005],
[1022.820007],
[1013.909973],
[993.409973],
[1041.329956],
[1020.],
[1016.799988],
[1026.439941],
[1027.98999],
[1025.040039],
[1040.880005],
[1037.],
[1051.369995],
[1077.430054],
[1069.400024],

[1082.],
[1077.859985],
[1052.],
[1025.52002],
[1029.51001],
[1046.],
[1030.01001],
[1013.659973],
[1028.099976],
[1019.],
[1016.900024],
[1049.22998],
[1058.540039],
[1058.099976],
[1086.030029],
[1093.599976],
[1100.],
[1090.],
[1077.310059],
[1079.890015],
[1061.859985],
[1074.060059],
[1083.560059],
[1065.130005],
[1079.],
[1079.02002],
[1064.890015],
[1063.030029],
[1067.560059],
[1099.349976],
[1122.329956],
[1140.98999],
[1142.170044],
[1131.319946],
[1118.180054],
[1118.599976],
[1131.069946],
[1141.119995],
[1143.849976],
[1148.859985],
[1143.650024],
[1158.5],
[1175.310059],
[1174.849976],
[1159.140015],
[1143.599976],
[1128.],
[1121.339966],

```
[1102.089966],  
[1120.      ]])
```

```
In [83]: inputs = sc.transform(inputs)  
inputs.shape
```

```
Out[83]: (185, 1)
```

```
In [84]: x_test = []  
for i in range(60,185):  
    x_test.append(inputs[i-60:i,0])
```

```
In [85]: x_test = np.array(x_test)  
x_test.shape
```

```
Out[85]: (125, 60)
```

```
In [86]: x_test = np.reshape(x_test, (x_test.shape[0],x_test.shape[1],1))  
x_test.shape
```

```
Out[86]: (125, 60, 1)
```

```
In [87]: predicted_price = regressor.predict(x_test)
```

```
In [88]: predicted_price = sc.inverse_transform(predicted_price)  
predicted_price
```

```
Out[88]: array([[1044.8307],  
                [1043.2227],  
                [1045.1395],  
                [1052.9774],  
                [1063.866 ],  
                [1074.5894],  
                [1083.37  ],  
                [1087.0924],  
                [1088.0599],  
                [1087.4794],  
                [1091.1273],  
                [1096.8883],  
                [1102.8933],  
                [1107.642 ],  
                [1111.4912],  
                [1117.7528],  
                [1127.5818],  
                [1136.8397],  
                [1143.4216],  
                [1147.2319],  
                [1147.7607],  
                [1146.9788],
```


[1145.2186],
[1137.653],
[1123.0232],
[1098.3064],
[1079.9979],
[1069.7445],
[1060.2571],
[1055.4653],
[1054.872],
[1056.7211],
[1063.2156],
[1072.4744],
[1080.6366],
[1087.6232],
[1093.7649],
[1097.5038],
[1100.9677],
[1106.1343],
[1108.8862],
[1106.6244],
[1093.6871],
[1079.5643],
[1074.043],
[1075.3395],
[1083.3462],
[1096.4722],
[1113.3584],
[1129.1605],
[1136.1622],
[1136.0964],
[1133.4982],
[1126.2064],
[1114.3472],
[1101.7638],
[1090.9359],
[1078.8405],
[1067.5453],
[1062.1519],
[1052.6641],
[1041.7971],
[1034.7639],
[1030.9658],
[1025.9142],
[1026.8452],
[1029.351],
[1029.9481],
[1029.691],
[1029.3276],

[1028.4214],
[1029.455],
[1031.2512],
[1034.8092],
[1042.8368],
[1051.016],
[1058.3462],
[1062.9633],
[1060.8774],
[1050.7166],
[1038.5541],
[1031.9785],
[1029.3857],
[1026.4559],
[1025.093],
[1024.1652],
[1022.9442],
[1026.4948],
[1034.6141],
[1043.1177],
[1053.1765],
[1063.4604],
[1072.2856],
[1076.5801],
[1074.992],
[1070.7935],
[1064.3425],
[1060.0208],
[1060.5356],
[1061.5637],
[1063.75],
[1066.5348],
[1066.8608],
[1064.7736],
[1062.5754],
[1066.1868],
[1077.148],
[1092.7615],
[1106.9686],
[1114.4674],
[1113.8649],
[1109.0118],
[1105.8606],
[1107.3865],
[1112.3234],
[1118.6447],
[1123.3654],
[1128.0115],

```
[1134.7515],  
[1141.7123],  
[1144.6299],  
[1141.7158],  
[1133.8413],  
[1124.3479],  
[1114.0529]], dtype=float32)
```

```
In [89]: plt.plot(real_stock_price,color = 'red', label = 'Real Price')  
plt.plot(predicted_price, color = 'blue', label = 'Predicted Price')  
plt.title(' Stock Price Prediction')  
plt.xlabel('Time')  
plt.ylabel(' Stock Price')  
plt.legend()  
plt.show()
```

