LSTM Stock

December 23, 2018

In [6]:
```
# 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-py
# 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 fi

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

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

In [7]:
```
dataset_train = pd.read_csv("N:/Stock Prediction project/DSE3/trainset.csv")
```

In [8]:
```
dataset_train
```

Out[8]:

	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 [12]:
```
dataset_train = pd.read_csv("N:/Stock Prediction project/DSE3/msft.csv")
dataset_train = dataset_train.drop(['symbol'], axis=1)
dataset_train = dataset_train.drop(['volume'], axis=1)

dataset_train
```

Out[12]:

	date	open	close	low	high
0	2016-01-05 00:00:00	123.430000	125.839996	122.309998	126.250000
1	2016-01-06 00:00:00	125.239998	119.980003	119.940002	125.540001
2	2016-01-07 00:00:00	116.379997	114.949997	114.930000	119.739998
3	2016-01-08 00:00:00	115.480003	116.620003	113.500000	117.440002
4	2016-01-11 00:00:00	117.010002	114.970001	114.089996	117.330002
5	2016-01-12 00:00:00	115.510002	115.550003	114.500000	116.059998
6	2016-01-13 00:00:00	116.459999	112.849998	112.589996	117.070000
7	2016-01-14 00:00:00	113.510002	114.379997	110.050003	115.029999
8	2016-01-15 00:00:00	113.330002	112.529999	111.919998	114.879997
9	2016-01-19 00:00:00	113.660004	110.379997	109.870003	115.870003
10	2016-01-20 00:00:00	109.059998	109.300003	108.320000	111.599998
11	2016-01-21 00:00:00	109.730003	110.000000	108.320000	110.580002
12	2016-01-22 00:00:00	111.879997	111.949997	110.190002	112.949997
13	2016-01-25 00:00:00	111.320000	110.120003	110.000000	114.629997
14	2016-01-26 00:00:00	110.419998	111.000000	107.300003	111.400002
15	2016-01-27 00:00:00	110.769997	110.709999	109.019997	112.570000
16	2016-01-28 00:00:00	110.900002	112.580002	109.900002	112.970001
17	2016-01-29 00:00:00	113.349998	114.470001	111.669998	114.589996
18	2016-02-01 00:00:00	114.000000	114.500000	112.900002	114.849998
19	2016-02-02 00:00:00	113.250000	110.559998	109.750000	113.860001
20	2016-02-03 00:00:00	113.379997	114.050003	109.639999	114.639999
21	2016-02-04 00:00:00	114.080002	115.709999	114.080002	116.320000
22	2016-02-05 00:00:00	115.120003	114.019997	109.709999	116.489998
23	2016-02-08 00:00:00	113.300003	111.160004	110.459999	113.300003
24	2016-02-09 00:00:00	111.169998	110.650002	109.639999	112.110001
25	2016-02-10 00:00:00	106.730003	107.519997	106.360001	112.110001
26	2016-02-11 00:00:00	105.629997	107.129997	104.110001	109.260002

```
27    2016-02-12 00:00:00  108.559998  107.839996  107.070000  109.430000
28    2016-02-16 00:00:00  109.110001  110.769997  107.010002  111.300003
29    2016-02-17 00:00:00  110.830002  111.239998  107.970001  112.110001
...                   ...         ...         ...         ...         ...
1229           2010-01-06   80.339996   79.830002   79.480003   81.330002
1230           2010-01-06   18.950001   19.150000   18.590000   19.299999
1231           2010-01-06   36.889999   37.009998   36.770000   37.279999
1232           2010-01-06   22.889999   23.000000   22.830000   23.150000
1233           2010-01-06   41.230000   41.490002   41.169998   41.669998
1234           2010-01-06   36.990002   36.849998   36.580002   37.959999
1235           2010-01-06  156.449997  155.240005  154.289993  158.789993
1236           2010-01-06   58.230000   59.779999   57.880001   59.990002
1237           2010-01-06   16.209999   16.389999   16.030001   16.540001
1238           2010-01-06   58.000000   58.089998   57.729999   58.390000
1239           2010-01-06   39.130001   39.230000   38.470001   39.669998
1240           2010-01-06   25.950001   26.580000   25.950001   26.910000
1241           2010-01-06   41.209999   40.889999   40.660000   41.340000
1242           2010-01-06   79.370003   79.010002   78.680000   79.370003
1243           2010-01-06   77.989998   77.660004   77.279999   78.080002
1244           2010-01-06  108.569996  108.920002  108.320000  109.639996
1245           2010-01-06   43.369999   45.840000   43.369999   46.029999
1246           2010-01-06   53.099998   53.430000   52.799999   53.700001
1247           2010-01-06   28.480000   28.160000   28.090000   28.510000
1248           2010-01-06  238.509995  234.669998  234.059998  238.649994
1249           2010-01-06   51.919998   52.000000   51.639999   52.119999
1250           2010-01-06   25.170000   25.219999   25.070000   25.290001
1251           2010-01-06    9.070000    9.160000    8.990000    9.280000
1252           2010-01-06   35.389999   36.689999   35.299999   36.779999
1253           2010-01-06   68.230003   68.440002   68.029999   68.940002
1254           2010-01-06    3.560000    3.640000    3.510000    3.680000
1255           2010-01-06   22.799999   22.760000   22.660000   22.930000
1256           2010-01-06   23.240000   23.129999   23.050001   23.349998
1257           2010-01-06   32.380001   31.709999   31.580000   32.380001
1258           2010-01-06   59.180000   59.430000   59.049999   59.930000

[1259 rows x 5 columns]
```

```
In [13]: trainset = dataset_train.iloc[:,1:2].values

In [14]: trainset

Out[14]: array([[123.43    ],
                [125.239998],
                [116.379997],
                ...,
                [ 23.24    ],
                [ 32.380001],
                [ 59.18    ]])
```

```
In [15]: from sklearn.preprocessing import MinMaxScaler
         sc = MinMaxScaler(feature_range = (0,1))
         training_scaled = sc.fit_transform(trainset)

In [16]: training_scaled

Out[16]: array([[0.19483703],
                [0.19773002],
                [0.18356877],
                ...,
                [0.03469985],
                [0.04930864],
                [0.09214401]])

In [17]: x_train = []
         y_train = []

In [18]: 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 [19]: x_train.shape

Out[19]: (1199, 60)

In [20]: x_train = np.reshape(x_train, (x_train.shape[0],x_train.shape[1],1))

In [21]: from keras.models import Sequential
         from keras.layers import Dense
         from keras.layers import LSTM
         from keras.layers import Dropout

Using TensorFlow backend.


In [22]: regressor = Sequential()
         regressor.add(LSTM(units = 50,return_sequences = True,input_shape = (x_train.shape[1]

In [23]: regressor.add(Dropout(0.2))

In [24]: regressor.add(LSTM(units = 50,return_sequences = True))
         regressor.add(Dropout(0.2))

In [25]: regressor.add(LSTM(units = 50,return_sequences = True))
         regressor.add(Dropout(0.2))

In [26]: regressor.add(LSTM(units = 50))
         regressor.add(Dropout(0.2))
```

```
In [27]: regressor.add(Dense(units = 1))

In [28]: regressor.compile(optimizer = 'adam',loss = 'mean_squared_error')

In [29]: regressor.fit(x_train,y_train,epochs = 100, batch_size = 32)

Epoch 1/100
1199/1199 [==============================] - 6s 5ms/step - loss: 0.0072
Epoch 2/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0062
Epoch 3/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0063
Epoch 4/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0062
Epoch 5/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0062
Epoch 6/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0062
Epoch 7/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0063
Epoch 8/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0061
Epoch 9/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0060
Epoch 10/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0060
Epoch 11/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0060
Epoch 12/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0062
Epoch 13/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0061
Epoch 14/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0059
Epoch 15/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0060
Epoch 16/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0060
Epoch 17/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0059A: 0s - loss: 0.
Epoch 18/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0061
Epoch 19/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0060
Epoch 20/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0060
Epoch 21/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0060
```

```
Epoch 22/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0060
Epoch 23/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0059
Epoch 24/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0059
Epoch 25/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0059
Epoch 26/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0059
Epoch 27/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0058
Epoch 28/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0059
Epoch 29/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0058
Epoch 30/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0059
Epoch 31/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0059
Epoch 32/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0058
Epoch 33/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0058
Epoch 34/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0058
Epoch 35/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 36/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0058
Epoch 37/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0059
Epoch 38/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0058
Epoch 39/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 40/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 41/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0058
Epoch 42/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 43/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0058
Epoch 44/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0057
Epoch 45/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
```

```
Epoch 46/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 47/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 48/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0058
Epoch 49/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 50/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 51/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0058
Epoch 52/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 53/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0058
Epoch 54/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0057
Epoch 55/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0057
Epoch 56/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0058
Epoch 57/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0058
Epoch 58/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0057
Epoch 59/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0057
Epoch 60/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0057
Epoch 61/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 62/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 63/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0056
Epoch 64/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0057
Epoch 65/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0057
Epoch 66/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 67/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 68/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 69/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0056
```

```
Epoch 70/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0057
Epoch 71/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 72/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 73/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0056
Epoch 74/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0056
Epoch 75/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0056
Epoch 76/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 77/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 78/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 79/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0056
Epoch 80/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0057
Epoch 81/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0056
Epoch 82/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0056
Epoch 83/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0056
Epoch 84/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0056
Epoch 85/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 86/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0056
Epoch 87/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0056
Epoch 88/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0056
Epoch 89/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 90/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0057
Epoch 91/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0057
Epoch 92/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0056
Epoch 93/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0056
```

```
Epoch 94/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0062
Epoch 95/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0068
Epoch 96/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0057
Epoch 97/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0057
Epoch 98/100
1199/1199 [==============================] - 3s 3ms/step - loss: 0.0057
Epoch 99/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0056
Epoch 100/100
1199/1199 [==============================] - 3s 2ms/step - loss: 0.0056
```

```
Out[29]: <keras.callbacks.History at 0x20656223940>
```

```
In [36]: dataset_test =pd.read_csv("N:/Stock Prediction project/DSE3/msft_test.csv")
         dataset_test
```

Out[36]:

	date	symbol	open	close	low	high
0	2010-01-06	CHRW	57.820000	57.340000	57.200001	57.990002
1	2010-01-06	CHTR	35.000000	35.000000	35.000000	35.000000
2	2010-01-06	CI	36.939999	36.900002	36.540001	37.560001
3	2010-01-06	CINF	26.500000	26.570000	26.389999	26.580000
4	2010-01-06	CL	83.620003	83.330002	82.400002	83.750000
5	2010-01-06	CLX	61.770000	61.939999	61.500000	62.439999
6	2010-01-06	CMA	30.570000	31.290001	30.430000	31.490000
7	2010-01-06	CMCSA	16.780001	16.620001	16.530001	16.799999
8	2010-01-06	CME	340.509987	339.809990	338.079987	342.000008
9	2010-01-06	CMG	88.589996	87.320000	86.599998	89.599998
10	2010-01-06	CMI	47.849998	48.490002	47.779999	48.619999
11	2010-01-06	CMS	15.740000	15.690000	15.660000	15.840000
12	2010-01-06	CNC	20.950001	21.260000	20.920000	21.280001
13	2010-01-06	CNP	14.450000	14.270000	14.180000	14.450000
14	2010-01-06	COF	40.349998	40.810001	40.200001	40.970001
15	2010-01-06	COG	46.250000	45.970001	45.770000	46.459999
16	2010-01-06	COH	36.650002	37.470001	36.570000	37.570000
17	2010-01-06	COL	56.099998	56.419998	56.099998	56.930000
18	2010-01-06	COO	37.939999	38.169998	37.860001	38.480000
19	2010-01-06	COP	52.669961	53.009959	52.329959	53.069958
20	2010-01-06	COST	59.070000	60.000000	59.000000	60.049999
21	2010-01-06	CPB	33.840000	33.549999	33.360001	33.919998
22	2010-01-06	CRM	74.750000	74.370003	73.980003	75.000000
23	2010-01-06	CSCO	24.540001	24.420000	24.340000	24.740000
24	2010-01-06	CSX	50.230001	50.310001	49.820000	50.710001
25	2010-01-06	CTAS	26.190001	26.420000	26.190001	26.469999

26	2010-01-06	CTL	36.540001	35.279999	35.220001	36.540001
27	2010-01-06	CTSH	47.529999	47.330002	47.060001	47.750000
28	2010-01-06	CTXS	42.730000	42.700001	42.560001	43.330002
29	2010-01-06	CVS	32.450001	32.560001	32.360001	32.980000
..	...	...	...	...	...	...
96	2010-01-06	FRT	67.000000	66.769997	66.120003	67.559998
97	2010-01-06	FSLR	138.429993	140.020004	137.649994	140.250000
98	2010-01-06	FTI	60.080002	60.529999	58.869999	60.610001
99	2010-01-06	FTR	7.990000	7.800000	7.790000	7.990000
100	2010-01-06	GD	69.029999	69.239998	68.750000	69.360001
101	2010-01-06	GE	15.530000	15.450000	15.440000	15.620000
102	2010-01-06	GGP	11.980004	12.039995	11.789995	12.099999
103	2010-01-06	GILD	44.000000	44.759998	43.610001	45.009998
104	2010-01-06	GIS	70.309998	70.660004	69.940002	70.919998
105	2010-01-06	GLW	19.570000	19.379999	19.309999	19.780001
106	2010-01-06	GOOG	625.861078	608.261023	606.361042	625.861078
107	2010-01-06	GOOGL	625.860033	608.260035	606.360021	625.860033
108	2010-01-06	GPC	38.099998	38.020000	37.930000	38.250000
109	2010-01-06	GPN	52.619999	52.369999	52.040001	52.750000
110	2010-01-06	GPS	20.700001	21.040001	20.580000	21.080000
111	2010-01-06	GRMN	32.200001	32.020000	31.850000	32.820000
112	2010-01-06	GS	175.380005	174.259995	173.759995	175.380005
113	2010-01-06	GT	15.760000	15.830000	15.600000	15.970000
114	2010-01-06	GWW	97.269997	97.650002	97.269997	98.250000
115	2010-01-06	HAL	31.740000	32.400002	31.639999	32.590000
116	2010-01-06	HAR	37.220001	37.349998	36.959999	37.450001
117	2010-01-06	HAS	32.220001	32.150002	31.700001	32.220001
118	2010-01-06	HBAN	3.750000	3.860000	3.720000	3.890000
119	2010-01-06	HBI	24.889999	24.889999	24.610001	24.920000
120	2010-01-06	HCN	44.750000	44.439999	44.310001	44.950001
121	2010-01-06	HCP	29.850002	29.810001	29.700002	30.240001
122	2010-01-06	HD	28.879999	28.780001	28.700001	29.000000
123	2010-01-06	HES	63.310001	63.720001	63.310001	64.510002
124	2010-01-06	HIG	25.510000	26.120001	25.090000	26.379999
125	2010-01-06	HOG	25.740000	25.590000	25.450001	25.990000

	volume
0	1685500.0
1	0.0
2	2611900.0
3	1139100.0
4	7195200.0
5	1187100.0
6	1573400.0
7	13870700.0
8	2522500.0
9	282900.0
10	1745900.0

```
11    4671800.0
12     543200.0
13    5716100.0
14    5819000.0
15    4633200.0
16    2968300.0
17     895400.0
18     390700.0
19   11631900.0
20    3722900.0
21    2829500.0
22    5122400.0
23   35715700.0
24    9919200.0
25    1570100.0
26    5333000.0
27    6270400.0
28    3046500.0
29   11319600.0
..          ...
96     578300.0
97    2195800.0
98    2724200.0
99    4075600.0
100   1149700.0
101  55464900.0
102   4165100.0
103  32750400.0
104   4686800.0
105  13448300.0
106   7987100.0
107   7949400.0
108    656600.0
109   1628200.0
110  15155100.0
111   1283100.0
112   7381100.0
113   3111400.0
114    486800.0
115  15720800.0
116    668700.0
117   1307500.0
118  18761000.0
119   1973200.0
120   1061700.0
121   2963700.0
122   8833200.0
123   3545900.0
```

```
124  10690200.0
125   3283200.0

[126 rows x 7 columns]
```

```
In [39]: dataset_test = dataset_test.drop(['symbol'], axis=1)
         dataset_test= dataset_test.drop(['volume'], axis=1)
         real_stock_price = dataset_test.iloc[:,1:2].values
```

```
In [41]: dataset_total = pd.concat((dataset_train['open'],dataset_test['open']),axis = 0)
         dataset_total
```

```
Out[41]: 0      123.430000
         1      125.239998
         2      116.379997
         3      115.480003
         4      117.010002
         5      115.510002
         6      116.459999
         7      113.510002
         8      113.330002
         9      113.660004
         10     109.059998
         11     109.730003
         12     111.879997
         13     111.320000
         14     110.419998
         15     110.769997
         16     110.900002
         17     113.349998
         18     114.000000
         19     113.250000
         20     113.379997
         21     114.080002
         22     115.120003
         23     113.300003
         24     111.169998
         25     106.730003
         26     105.629997
         27     108.559998
         28     109.110001
         29     110.830002
                   ...
         96      67.000000
         97     138.429993
         98      60.080002
         99       7.990000
         100     69.029999
```

```
101     15.530000
102     11.980004
103     44.000000
104     70.309998
105     19.570000
106    625.861078
107    625.860033
108     38.099998
109     52.619999
110     20.700001
111     32.200001
112    175.380005
113     15.760000
114     97.269997
115     31.740000
116     37.220001
117     32.220001
118      3.750000
119     24.889999
120     44.750000
121     29.850002
122     28.879999
123     63.310001
124     25.510000
125     25.740000
Name: open, Length: 1385, dtype: float64
```

In [42]:
```
inputs = dataset_total[len(dataset_total) - len(dataset_test)-60:].values
inputs
```

Out[42]:
```
array([ 27.610001,  34.560001,  13.45    ,  32.5     ,  49.830002,
        39.939999,  29.35    ,  30.92    ,  22.26    ,  26.700001,
        37.959999,  36.25    ,  30.9     ,  47.73    ,  14.23    ,
        38.499999,  69.209999,  56.939999,  41.369999,  44.240002,
       134.600006,  19.57    ,  60.119999,  37.75    , 106.779999,
        66.839996,  81.999999,  45.040001,  16.310013,  11.26    ,
        80.339996,  18.950001,  36.889999,  22.889999,  41.23    ,
        36.990002, 156.449997,  58.23    ,  16.209999,  58.      ,
        39.130001,  25.950001,  41.209999,  79.370003,  77.989998,
       108.569996,  43.369999,  53.099998,  28.48    , 238.509995,
        51.919998,  25.17    ,   9.07    ,  35.389999,  68.230003,
         3.56    ,  22.799999,  23.24    ,  32.380001,  59.18    ,
        57.82    ,  35.      ,  36.939999,  26.5     ,  83.620003,
        61.77    ,  30.57    ,  16.780001, 340.509987,  88.589996,
        47.849998,  15.74    ,  20.950001,  14.45    ,  40.349998,
        46.25    ,  36.650002,  56.099998,  37.939999,  52.669961,
        59.07    ,  33.84    ,  74.75    ,  24.540001,  50.230001,
        26.190001,  36.540001,  47.529999,  42.73    ,  32.450001,
```

```
                 79.440002,  45.779999,  38.5     ,  11.99    ,  33.849998,
                 55.57    ,  14.85    ,  23.6     ,  59.540001,  11.45    ,
                 74.770003,  31.9     ,  32.500001,  28.139999,  50.66    ,
                 48.760002,  82.93    ,  42.15    ,  30.98    ,  28.9     ,
                 34.77    ,  43.049999,  16.659982,  59.490002,  76.580002,
                 18.610001,  23.629998,  44.529999,  44.75    ,  31.48    ,
                 34.490002,  48.619999,  60.91    ,  43.25    ,  20.83    ,
                 98.839996, 108.949997,  33.27    ,  45.      ,  25.77    ,
                 86.519997,  83.230003,   1.79    ,  63.549999,  80.790001,
                 88.290001,  48.07    ,  35.169998,  25.73    ,  11.73    ,
                 11.21    ,  44.189999,  84.860001,  84.330002,  46.299999,
                 52.200001,  24.93    ,  49.419998,  10.3     ,  11.53    ,
                 32.970001,  46.      ,  98.009995,  56.68    ,  16.19    ,
                 13.910004,  67.      , 138.429993,  60.080002,   7.99    ,
                 69.029999,  15.53    ,  11.980004,  44.      ,  70.309998,
                 19.57    , 625.861078, 625.860033,  38.099998,  52.619999,
                 20.700001,  32.200001, 175.380005,  15.76    ,  97.269997,
                 31.74    ,  37.220001,  32.220001,   3.75    ,  24.889999,
                 44.75    ,  29.850002,  28.879999,  63.310001,  25.51    ,
                 25.74    ])

In [43]: inputs = inputs.reshape(-1,1)

In [44]: inputs

Out[44]: array([[ 27.610001],
                [ 34.560001],
                [ 13.45    ],
                [ 32.5     ],
                [ 49.830002],
                [ 39.939999],
                [ 29.35    ],
                [ 30.92    ],
                [ 22.26    ],
                [ 26.700001],
                [ 37.959999],
                [ 36.25    ],
                [ 30.9     ],
                [ 47.73    ],
                [ 14.23    ],
                [ 38.499999],
                [ 69.209999],
                [ 56.939999],
                [ 41.369999],
                [ 44.240002],
                [134.600006],
                [ 19.57    ],
                [ 60.119999],
```

```
[ 37.75    ],
[106.779999],
[ 66.839996],
[ 81.999999],
[ 45.040001],
[ 16.310013],
[ 11.26    ],
[ 80.339996],
[ 18.950001],
[ 36.889999],
[ 22.889999],
[ 41.23    ],
[ 36.990002],
[156.449997],
[ 58.23    ],
[ 16.209999],
[ 58.      ],
[ 39.130001],
[ 25.950001],
[ 41.209999],
[ 79.370003],
[ 77.989998],
[108.569996],
[ 43.369999],
[ 53.099998],
[ 28.48    ],
[238.509995],
[ 51.919998],
[ 25.17    ],
[  9.07    ],
[ 35.389999],
[ 68.230003],
[  3.56    ],
[ 22.799999],
[ 23.24    ],
[ 32.380001],
[ 59.18    ],
[ 57.82    ],
[ 35.      ],
[ 36.939999],
[ 26.5     ],
[ 83.620003],
[ 61.77    ],
[ 30.57    ],
[ 16.780001],
[340.509987],
[ 88.589996],
[ 47.849998],
```

```
[ 15.74    ],
[ 20.950001],
[ 14.45    ],
[ 40.349998],
[ 46.25    ],
[ 36.650002],
[ 56.099998],
[ 37.939999],
[ 52.669961],
[ 59.07    ],
[ 33.84    ],
[ 74.75    ],
[ 24.540001],
[ 50.230001],
[ 26.190001],
[ 36.540001],
[ 47.529999],
[ 42.73    ],
[ 32.450001],
[ 79.440002],
[ 45.779999],
[ 38.5     ],
[ 11.99    ],
[ 33.849998],
[ 55.57    ],
[ 14.85    ],
[ 23.6     ],
[ 59.540001],
[ 11.45    ],
[ 74.770003],
[ 31.9     ],
[ 32.500001],
[ 28.139999],
[ 50.66    ],
[ 48.760002],
[ 82.93    ],
[ 42.15    ],
[ 30.98    ],
[ 28.9     ],
[ 34.77    ],
[ 43.049999],
[ 16.659982],
[ 59.490002],
[ 76.580002],
[ 18.610001],
[ 23.629998],
[ 44.529999],
[ 44.75    ],
```

```
[ 31.48    ],
[ 34.490002],
[ 48.619999],
[ 60.91    ],
[ 43.25    ],
[ 20.83    ],
[ 98.839996],
[108.949997],
[ 33.27    ],
[ 45.      ],
[ 25.77    ],
[ 86.519997],
[ 83.230003],
[  1.79    ],
[ 63.549999],
[ 80.790001],
[ 88.290001],
[ 48.07    ],
[ 35.169998],
[ 25.73    ],
[ 11.73    ],
[ 11.21    ],
[ 44.189999],
[ 84.860001],
[ 84.330002],
[ 46.299999],
[ 52.200001],
[ 24.93    ],
[ 49.419998],
[ 10.3     ],
[ 11.53    ],
[ 32.970001],
[ 46.      ],
[ 98.009995],
[ 56.68    ],
[ 16.19    ],
[ 13.910004],
[ 67.      ],
[138.429993],
[ 60.080002],
[  7.99    ],
[ 69.029999],
[ 15.53    ],
[ 11.980004],
[ 44.      ],
[ 70.309998],
[ 19.57    ],
[625.861078],
```

```
                [625.860033],
                [ 38.099998],
                [ 52.619999],
                [ 20.700001],
                [ 32.200001],
                [175.380005],
                [ 15.76    ],
                [ 97.269997],
                [ 31.74    ],
                [ 37.220001],
                [ 32.220001],
                [  3.75    ],
                [ 24.889999],
                [ 44.75    ],
                [ 29.850002],
                [ 28.879999],
                [ 63.310001],
                [ 25.51    ],
                [ 25.74    ]])
```

In [45]:
```
inputs = sc.transform(inputs)
inputs.shape
```

Out[45]: (186, 1)

In [46]:
```
x_test = []
for i in range(60,185):
    x_test.append(inputs[i-60:i,0])
```

In [47]:
```
x_test = np.array(x_test)
x_test.shape
```

Out[47]: (125, 60)

In [48]:
```
x_test = np.reshape(x_test, (x_test.shape[0],x_test.shape[1],1))
x_test.shape
```

Out[48]: (125, 60, 1)

In [49]:
```
predicted_price = regressor.predict(x_test)
```

In [50]:
```
predicted_price = sc.inverse_transform(predicted_price)
predicted_price
```

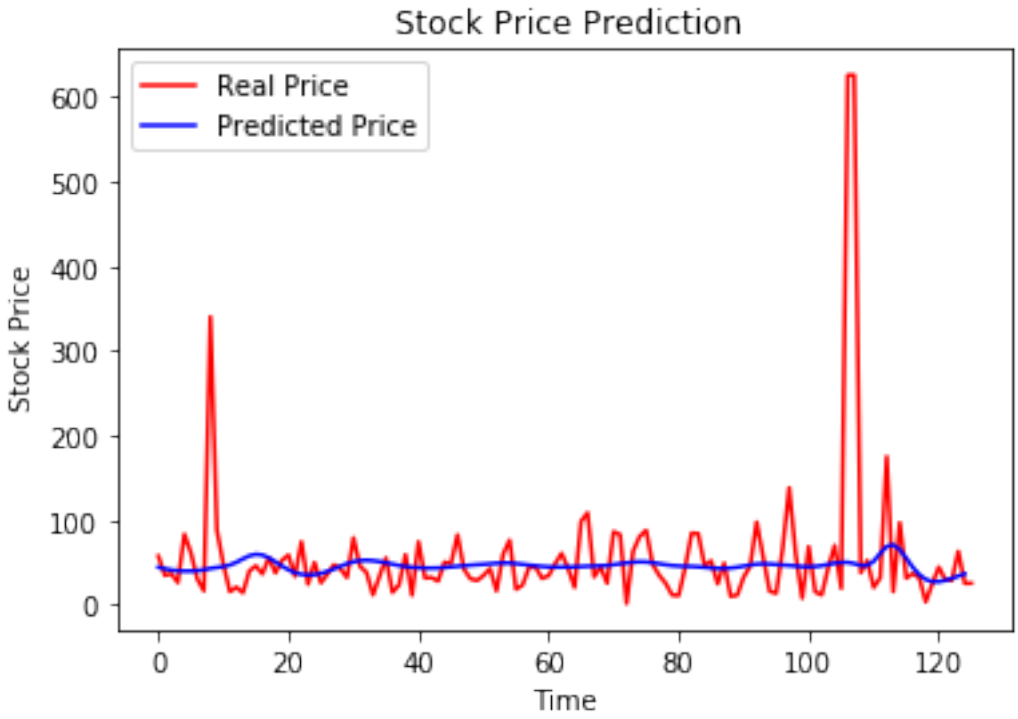
Out[50]:
```
array([[44.51459 ],
       [42.7872  ],
       [41.324886],
       [40.39223 ],
       [40.05125 ],
```

```
[40.11954 ],
[40.592754],
[41.59299 ],
[43.194927],
[44.31469 ],
[45.38308 ],
[47.230633],
[50.411396],
[54.494926],
[58.14856 ],
[59.949615],
[59.035824],
[55.71448 ],
[50.98493 ],
[46.00117 ],
[41.5997  ],
[38.1835  ],
[36.226902],
[35.41745 ],
[35.855362],
[37.280277],
[39.650978],
[42.59512 ],
[45.770737],
[48.750374],
[51.135063],
[52.42761 ],
[52.827335],
[52.32567 ],
[51.285847],
[49.875328],
[48.27705 ],
[46.634937],
[45.446968],
[44.544872],
[43.967827],
[43.62557 ],
[43.553867],
[43.779312],
[44.30041 ],
[45.007553],
[45.741013],
[46.41951 ],
[47.003407],
[47.6426  ],
[48.061123],
[48.72827 ],
[49.29609 ],
```

```
[49.669926],
[49.560265],
[48.865837],
[48.04124 ],
[47.139305],
[46.29994 ],
[45.57469 ],
[45.043278],
[44.783382],
[44.764614],
[44.848503],
[45.06057 ],
[45.418198],
[45.7909  ],
[46.031822],
[46.381927],
[46.655228],
[47.62048 ],
[48.753994],
[49.767193],
[50.606533],
[51.053658],
[50.831696],
[49.97428 ],
[48.795452],
[47.571674],
[46.629658],
[46.025223],
[45.730568],
[45.556583],
[45.09679 ],
[44.45776 ],
[43.732456],
[43.257576],
[43.22302 ],
[43.647724],
[44.602764],
[45.9533  ],
[47.242756],
[48.258854],
[48.599014],
[48.363937],
[47.80205 ],
[47.22476 ],
[46.75012 ],
[46.051697],
[45.32539 ],
[45.1941  ],
```

```
                [45.497196],
                [46.49875 ],
                [47.943554],
                [49.363583],
                [50.20588 ],
                [50.395664],
                [48.888775],
                [46.62298 ],
                [46.66331 ],
                [51.70353 ],
                [60.979603],
                [69.15905 ],
                [71.126724],
                [65.494354],
                [55.24119 ],
                [44.50511 ],
                [35.826893],
                [30.182297],
                [27.628984],
                [27.525534],
                [29.00324 ],
                [31.39471 ],
                [34.169483],
                [36.788795]], dtype=float32)
```

```
In [51]: 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()
```


Stock Price Prediction
Real Price
Predicted Price
Stock Price
Time
0
100
200
300
400
500
600
0
20
40
60
80
100
120