B211038- Srushti Gavale

Google stock

In [2]:

```
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
```

In [3]:

```
dataset_train = pd.read_csv("trainset.csv")
```

In [4]:

```
dataset_train
```

Out[4]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2013-01-02	357.385559	361.151062	355.959839	359.288177	359.288177	5115500
1	2013-01-03	360.122742	363.600128	358.031342	359.496826	359.496826	4666500
2	2013-01-04	362.313507	368.339294	361.488861	366.600616	366.600616	5562800
3	2013-01-07	365.348755	367.301056	362.929504	365.001007	365.001007	3332900
4	2013-01-08	365.393463	365.771027	359.874359	364.280701	364.280701	3373900
...	...	...	...	...	...	...	...
1254	2017-12-22	1061.109985	1064.199951	1059.439941	1060.119995	1060.119995	755100
1255	2017-12-26	1058.069946	1060.119995	1050.199951	1056.739990	1056.739990	760600
1256	2017-12-27	1057.390015	1058.369995	1048.050049	1049.369995	1049.369995	1271900
1257	2017-12-28	1051.599976	1054.750000	1044.770020	1048.140015	1048.140015	837100
1258	2017-12-29	1046.719971	1049.699951	1044.900024	1046.400024	1046.400024	887500

1259 rows × 7 columns

In [5]:

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

In [6]:

```
trainset
```

Out[6]:

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

In [7]:

```
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0,1))
training_scaled = sc.fit_transform(trainset)
```

In [8]:

```
training_scaled
```

Out[8]:

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

In [9]:

```
x_train = []
y_train = []
```

In [10]:

```
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 [11]:

```
x_train.shape
```

Out[11]:

```
(1199, 60)
```

In [12]:

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

In [13]:

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
```

In [14]:

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

In [15]:

```
regressor.add(Dropout(0.2))
```

In [16]:

```
regressor.add(LSTM(units = 50,return_sequences = True))
regressor.add(Dropout(0.2))
```

In [17]:

```
regressor.add(LSTM(units = 50,return_sequences = True))
regressor.add(Dropout(0.2))
```

In [18]:

```
regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))
```

In [19]:

```
regressor.add(Dense(units = 1))
```

In [20]:

```
regressor.compile(optimizer = 'adam',loss = 'mean_squared_error')
```

In [21]:

```
regressor.fit(x_train,y_train,epochs = 100, batch_size = 32)
```

```
Epoch 1/100
38/38 [==============================] - 14s 119ms/step - loss: 0.0297
Epoch 2/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0046
Epoch 3/100
38/38 [==============================] - 6s 158ms/step - loss: 0.0042
Epoch 4/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0040
Epoch 5/100
38/38 [==============================] - 5s 138ms/step - loss: 0.0039
Epoch 6/100
38/38 [==============================] - 5s 134ms/step - loss: 0.0034
Epoch 7/100
38/38 [==============================] - 4s 118ms/step - loss: 0.0037
Epoch 8/100
38/38 [==============================] - 6s 158ms/step - loss: 0.0035
Epoch 9/100
38/38 [==============================] - 4s 117ms/step - loss: 0.0034
Epoch 10/100
```

In [22]:

```
dataset_test =pd.read_csv("testset.csv")
```

In [23]:

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

In [24]:

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

Out[24]:

```
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
```

In [25]:

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

Out[25]:

```
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 [26]:

```
inputs = inputs.reshape(-1,1)
```

In [27]:

```
inputs
```

Out[27]:

```
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.      ],
```

In [28]:

```
inputs = sc.transform(inputs)
inputs.shape
```

Out[28]:

```
(185, 1)
```

In [29]:

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

In [30]:

```
x_test = np.array(x_test)
x_test.shape
```

Out[30]:

```
(125, 60)
```

In [31]:

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

Out[31]:

```
(125, 60, 1)
```

In [32]:

```
predicted_price = regressor.predict(x_test)
```

```
4/4 [==============================] - 2s 33ms/step
```

In [33]:

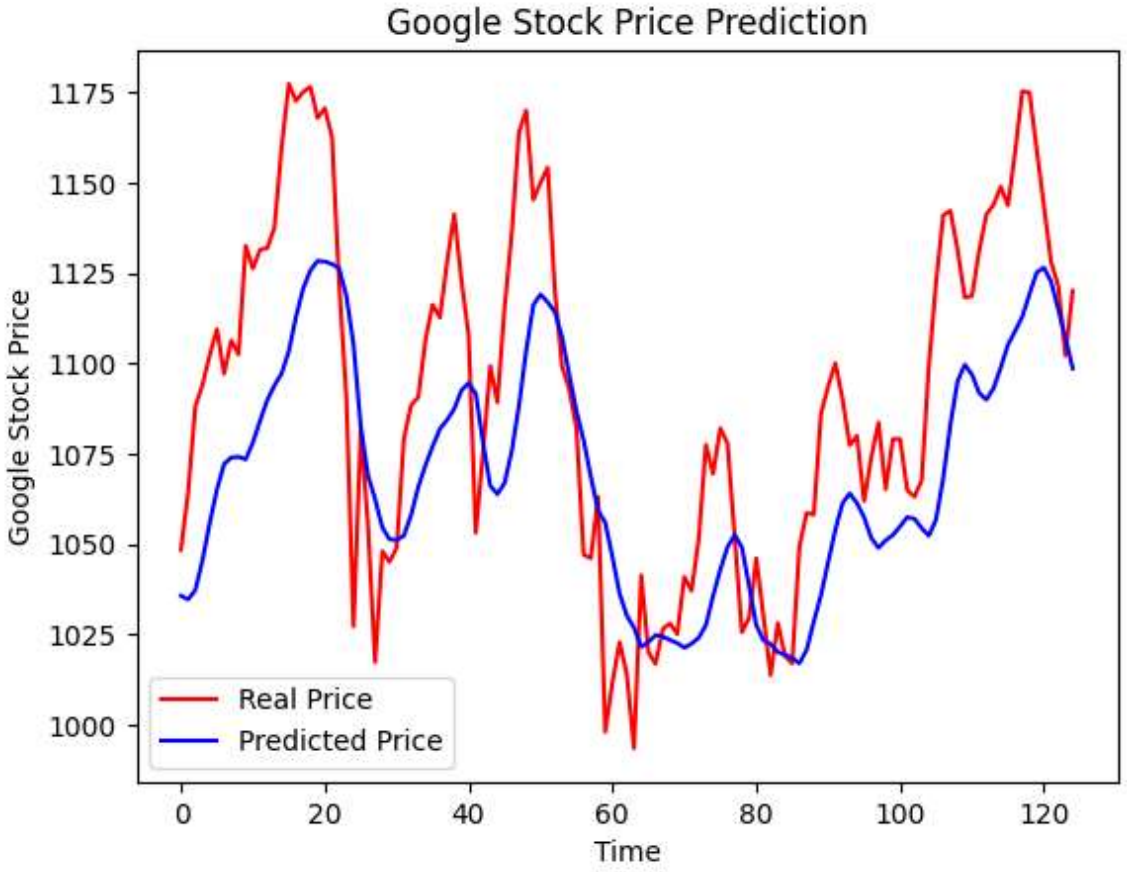
```
predicted_price = sc.inverse_transform(predicted_price)
predicted_price
```

Out[33]:

```
array([[1035.6561 ],
       [1034.6223 ],
       [1037.208  ],
       [1045.4957 ],
       [1055.7665 ],
       [1065.0334 ],
       [1072.0768 ],
       [1073.9147 ],
       [1074.0863 ],
       [1073.3953 ],
       [1078.0466 ],
       [1084.0615 ],
       [1089.7721 ],
       [1093.8557 ],
       [1097.1332 ],
       [1103.3224 ],
       [1112.8597 ],
       [1120.6991 ],
```

In [34]:

```
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()
```



In [ ]: