```
import  os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

```
dataset_train= pd.read_csv("Google_Stock_Price_Train.csv")
dataset_train.head()
```

	Date	Open	High	Low	Close	Volume
0	1/3/2012	325.25	332.83	324.97	663.59	7,380,500
1	1/4/2012	331.27	333.87	329.08	666.45	5,749,400
2	1/5/2012	329.83	330.75	326.89	657.21	6,590,300
3	1/6/2012	328.34	328.77	323.68	648.24	5,405,900
4	1/9/2012	322.04	322.29	309.46	620.76	11,688,800

```
training_set= dataset_train.iloc[:,1:2].values
print(training_set)
print(training_set.shape)
```

```
[[325.25]
 [331.27]
 [329.83]
 ...
 [793.7 ]
 [783.33]
 [782.75]]
(1258, 1)
```

```
from sklearn.preprocessing import  MinMaxScaler
scaler = MinMaxScaler(feature_range=(0,1))
scaled_training_set = scaler.fit_transform(training_set)
scaled_training_set
```

```
x_train =[]
y_train=[]
for i in range (60,1258):
    x_train.append(scaled_training_set[i-60:i,0])
    y_train.append(scaled_training_set[i,0])

x_train = np.array(x_train)
y_train = np.array(y_train)

print(x_train.shape)
print(y_train.shape)
```

```
(1198, 60)
(1198,)
```

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

```
(1198, 60, 1)
```

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

```
regressor =Sequential()
regressor.add(LSTM(units=50, return_sequences=True,input_shape=(x_train.shape[1],
regressor.add(Dropout(0.2))
regressor.add(LSTM(units=50, return_sequences=True))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units=50, return_sequences=True))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units=50))
regressor.add(Dropout(0.2))

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

```
regressor.compile(optimizer='adam',loss = 'mean_squared_error')
regressor.fit(x_train,y_train,epochs=100,batch_size=32)
```

```
Epoch 1/100
 1/38 [..............................] - ETA: 13:58 - loss: 0.0029
Epoch 2/100
 1/38 [..............................] - ETA: 18s - loss: 0.0026
Epoch 3/100
 1/38 [..............................] - ETA: 32s - loss: 0.0018
Epoch 4/100
 1/38 [..............................] - ETA: 34s - loss: 0.0027
Epoch 5/100
 1/38 [..............................] - ETA: 18s - loss: 0.0016
Epoch 6/100
 1/38 [..............................] - ETA: 30s - loss: 0.0023
Epoch 7/100
 1/38 [..............................] - ETA: 3s - loss: 0.0035
Epoch 8/100
 1/38 [..............................] - ETA: 3s - loss: 0.0032
Epoch 9/100
 1/38 [..............................] - ETA: 3s - loss: 0.0038
Epoch 10/100
 1/38 [                              ] - ETA: 3s - loss: 0.0042
```

```
<keras.callbacks.History at 0x7fef98399910>
```

```
dataset_test =pd.read_csv("Google_Stock_Price_Test.csv")
actual_stock_price = dataset_test.iloc[:,1:2].values
```

```
dataset_total = pd.concat((dataset_train['Open'],dataset_test['Open']), axis =0)
inputs = dataset_total[len(dataset_total)- len(dataset_test)-60:].values

inputs = inputs.reshape(-1,1)
inputs = scaler.transform(inputs)

x_test=[]
for i in range(60,80):
    x_test.append(inputs[i-60:i,0])

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

```
predicted_stock_price= regressor.predict(x_test)
predicted_stock_price = scaler.inverse_transform(predicted_stock_price)
```

```
plt.plot(actual_stock_price, color = 'red', label='Actual Google Stock Price')
plt.plot(predicted_stock_price, color = 'blue', label = 'Predicted Google Stock P
plt.title('Google Stock Price Precdiction')
plt.xlabel('Time')
plt.ylabel('Google Stock Price')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7fef83439c10>
```

Download