Experiment-3

Student Name: YANA SRIVASTAVA **UID: 20BCS2279**
Branch: CSE **Section/Group: 906/B**
Semester: 5th **Date of Performance: 01/09/2022**
Subject Name: ML Lab **Subject Code: 20CSP-317**

1.Aim/Overview of the practical: Implement Linear Regression on any data set.

2. Task to be done:

- **Pick a dataset.**
- **Load it into the program with appropriate path of dataset.**
- **Use numpy loadtxt function as it will load it as numpy N-dimensional array.**
- **Pick columns for both input & output.**
- **Split these columns into Train & Test using sklearn package, within that model_selection's train_test_split & pass input & output column, by default it splits in 75:25 train test split but we can alter this by explicitly passing as test_size in range [0,1].**
- **Create linear regression algorithm object by importing it from sklearn.linear_model as LinearRegression.**
- **As linear regression is all about curve fitting we will fit our input , output in algorithm using fit function.**

- **Store slope (m) , y intercept (c) & use dataset for creating a graph wherein we will draw data-point (of testing data) as well as line given by LinearRegression algorithm which best fits with the dataset we had picked so that we will get some visual clues about how well our algorithm is performing & various inputs.**

3. Algorithm/Flowchart (For programming based labs):

Linear Regression is the most basic algorithm in Machine Learning. It is a regression algorithm which means that it is useful when we are required to predict continuous values, that is, the output variable 'y' is continuous in nature.

4. CODE:

```
#CHIRAG BITHER
#20BCS1838
from sklearn import model_selection
from sklearn.linear_model import LinearRegression
import numpy as np
import matplotlib.pyplot as plt

# Loading dataset in program :
data = np.loadtxt("C:\\Users\\satya\\Downloads\\data.csv",delimiter=",")
# Picking column of input & output :
# Reshaping input column as fit function requires multi-dimensional array ( atleast 2D)
#
x = data[:,0].reshape(-1,1)
y = data[:,1]

# print(x.shape)
# Splitting Dataset for train & testing : ( Tuple unpacking )
X_train , X_test , Y_train , Y_test = model_selection.train_test_split(x,y)
```

```
# Now lets get an algorithm : ( Create object of Linear Regression )
algo1 = LinearRegression()
#Curve fitting :
algo1.fit(X_train,Y_train)

print("Slope : ",algo1.coef_[0])
print("Y intercept : ",algo1.intercept_)

# Code for creating graph :
# Stored coefficient values :
slope = algo1.coef_[0]
y_intercept = algo1.intercept_
# Creating line datapoint from given values of slope & intercept :
x_line = np.arange(10,70,0.1)
y_line = slope*(x_line) + y_intercept
# Draing in graph & labelling graph for better representation :
plt.xlabel("Input")
plt.ylabel("Output")
plt.title("Linear regression's given line v/s Testing data ")
plt.plot(x_line,y_line,"r",linewidth = 4)
plt.grid()
# Spreading Testing data all over the graph :
plt.scatter(X_test,Y_test)
plt.show()
```

5. OUTPUT:

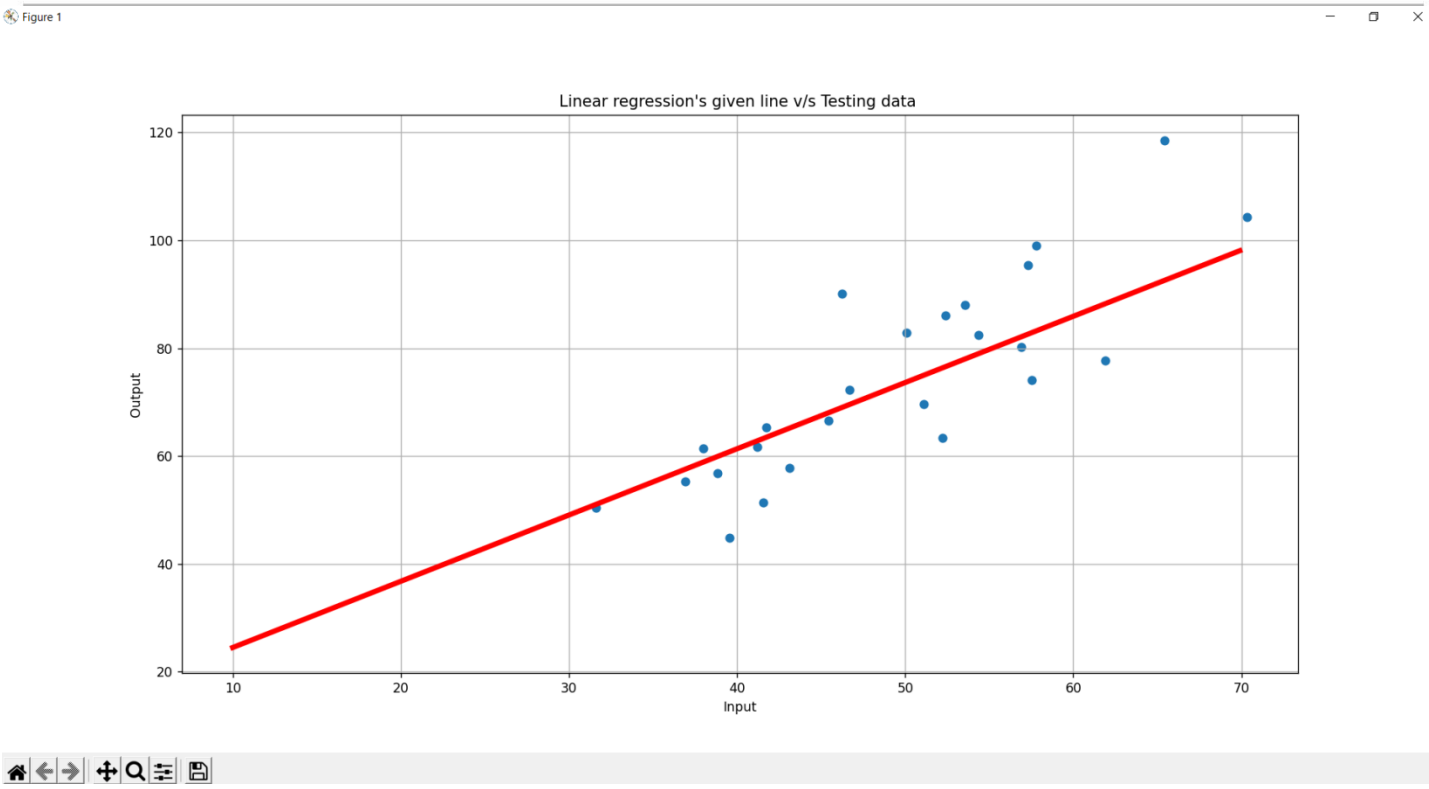
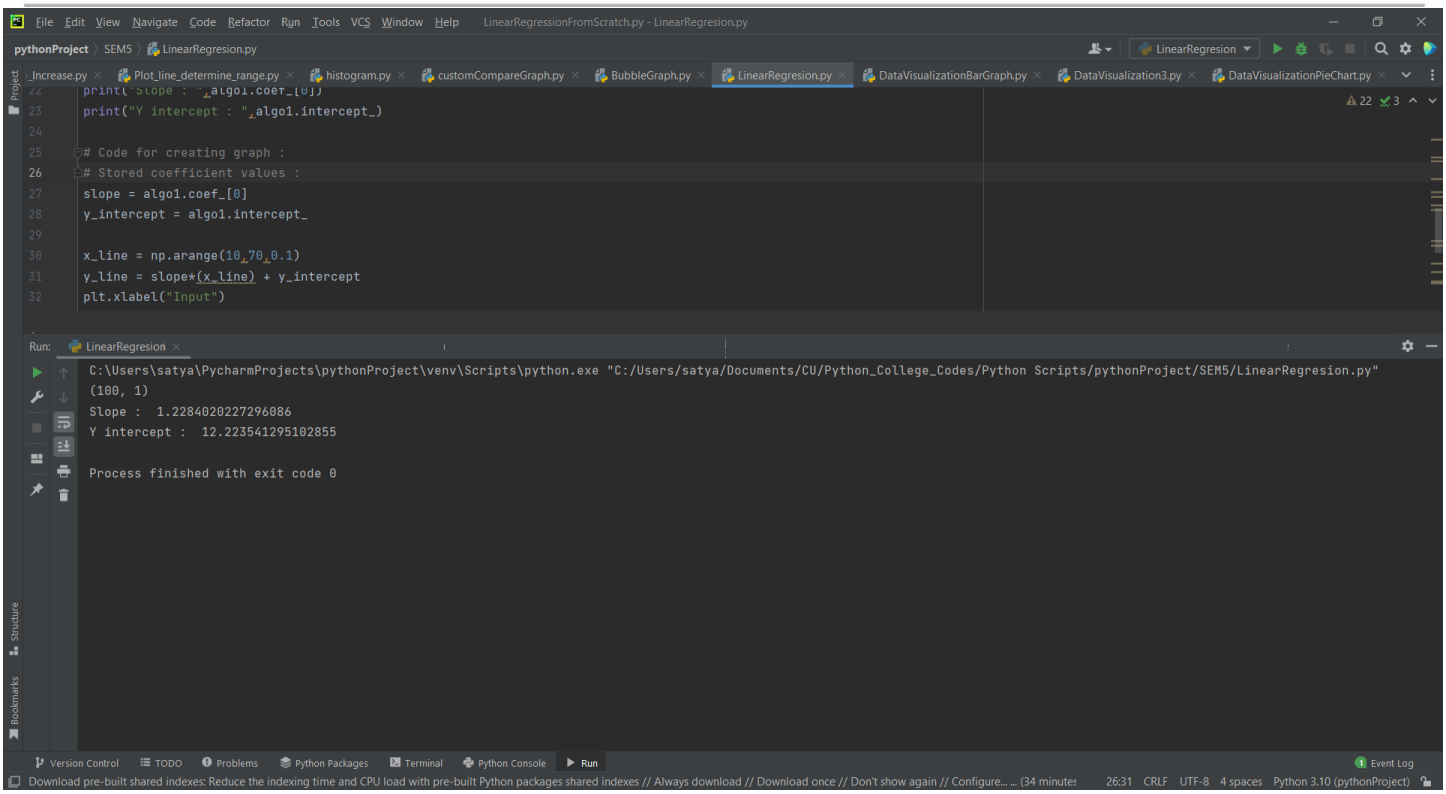


Figure 1
Linear regression's given line v/s Testing data
Output
Input
120
100
80
60
40
20
10
20
30
40
50
60
70



Learning outcomes (What I have learnt):

- **Used OOPs concept as well as numpy array's application in graph & in data loading.**
- **Get to know about how curve fitting works & how it predicts the output using Linear Regression algorithm.**
- **Learned how to create the graph between testing data points & Line given by algorithm & how to read that graph.**

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks

1.			
2.			
3.			