MACHINE LEARNING (CSI0702)

PRACTICAL-3
(LINEAR REGRESSION)

Submitted By:

Harsh Saradva (20BCM024)

Submitted To:

Meenaxi Tank

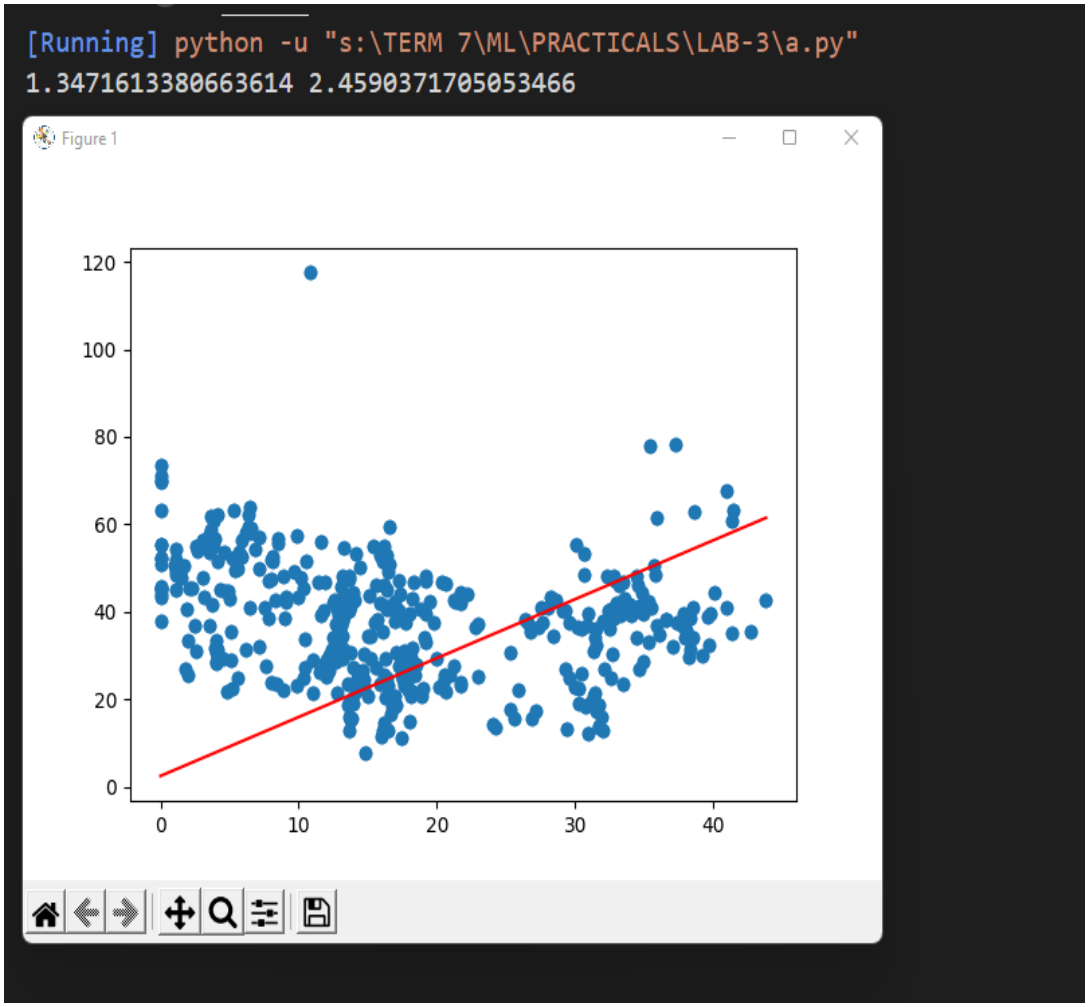
1.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df =pd.read_csv('S:\TERM 7\ML\PRACTICALS\LAB-3\Real_Estate_Price.csv')
plt.plot(df['X2 house age'],df['Y house price of unit area'],'o')
x=(df['X2 house age'])
y=df['Y house price of unit area']
m=0
c=0
l=0.0001
epochs=1000
n=float(len(x))

for i in range(epochs):
    y_pred=m*x+c
    d_m=(-2/n)*sum(x*(y-y_pred))
    d_c=(-2/n)*sum(y-y_pred)
    m=m-l*d_m
    c=c-l*d_c

print(m,c)
y_pred=m*x+c
plt.scatter(x,y)
plt.plot([min(x),max(x)],[min(y_pred),max(y_pred)],color='red')
plt.show()
```

output:


[Running] python -u "s:\TERM 7\ML\PRACTICALS\LAB-3\a.py"
1.3471613380663614 2.4590371705053466
Figure 1
120
100
80
60
40
20
0
0
10
20
30
40

2.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from numpy import linalg as la

def normal_eqn(x, y):
    rs1 = x.T @ x
    inverse = la.inv(rs1)
    rs2 = inverse @ x.T
    newf = rs2 @ y
    return np.array(newf)

df = pd.read_csv('S:\TERM 7\ML\PRACTICALS\LAB-3\data.csv')

x_data = x = df.iloc[:,[0]]
print(x_data)
y_data = df.iloc[:,[1]]
print(y_data)
n = len(x)
print('Rows : ', n)

ones = np.ones([x_data.shape[0], 1])
x_data = np.concatenate((ones, x_data), axis=1)

rec = normal_eqn(x_data,y_data)
print('Intercept : ', rec[0])
print('Slope : ', rec[1])

plt.scatter(x, y_data, color='lightblue')
plt.plot(x, rec[1]*x + rec[0], color='red')
plt.show()
```

output:

```
[Running] python -u "s:\TERM 7\ML\PRACTICALS\LAB-3\b.py"
    32.502345269453031
0            53.426804
1            61.530358
2            47.475640
3            59.813208
4            55.142188
..                 ...
94           50.030174
95           49.239765
96           50.039576
97           48.149859
98           25.128485

[99 rows x 1 columns]
    31.70700584656992
0           68.777596
1           62.562382
2           71.546632
3           87.230925
4           78.211518
..                ...
94          81.536991
95          72.111832
96          85.232007
97          66.224958
98          53.454394

[99 rows x 1 columns]
Rows :  99
Intercept :  [9.90860619]
Slope :  [1.28735737]
```