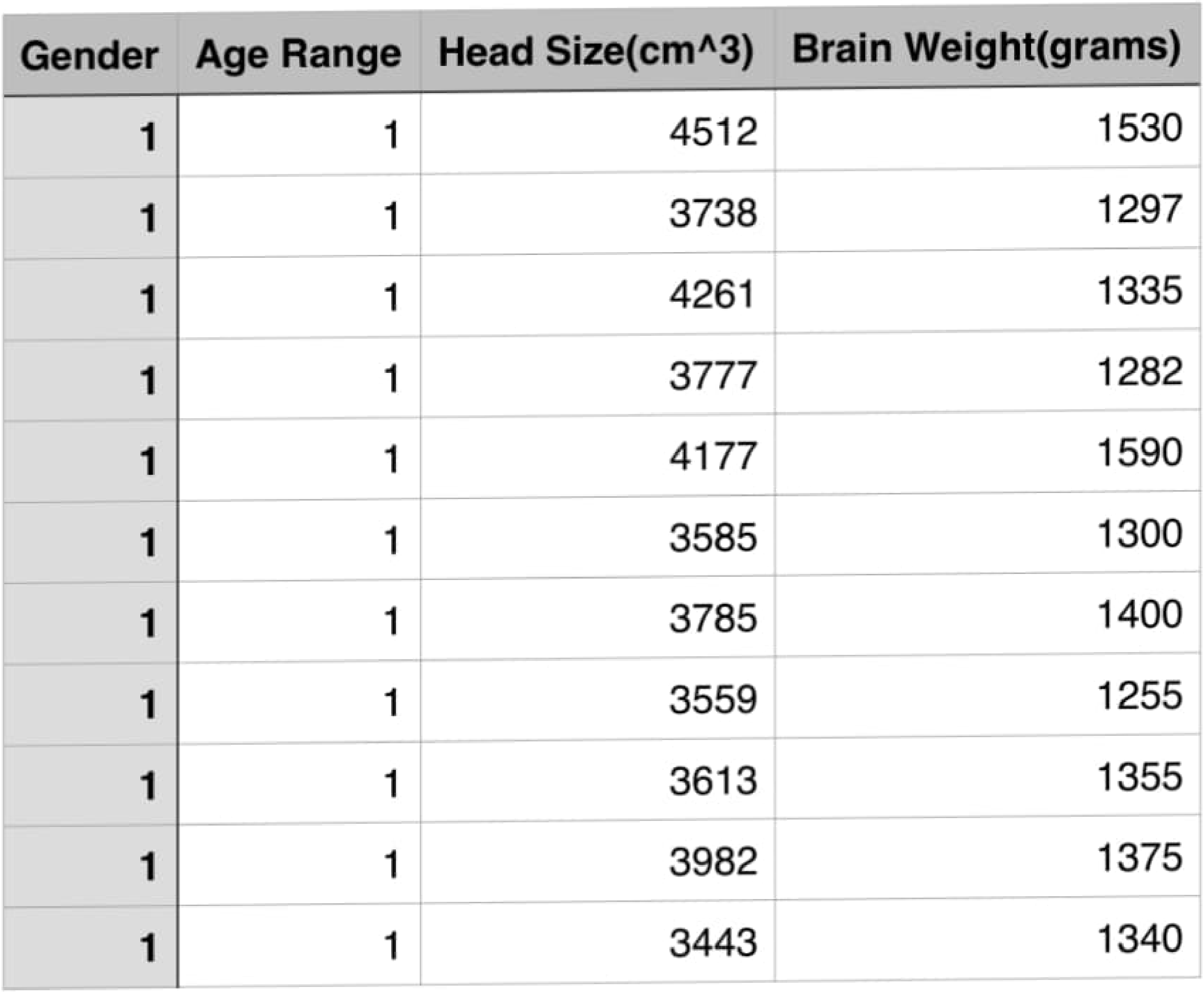
Application 10

Supervised Machine Learning

Linear Regression with User Defined Algorithm

There is one data set which contains information about Head and Brain.



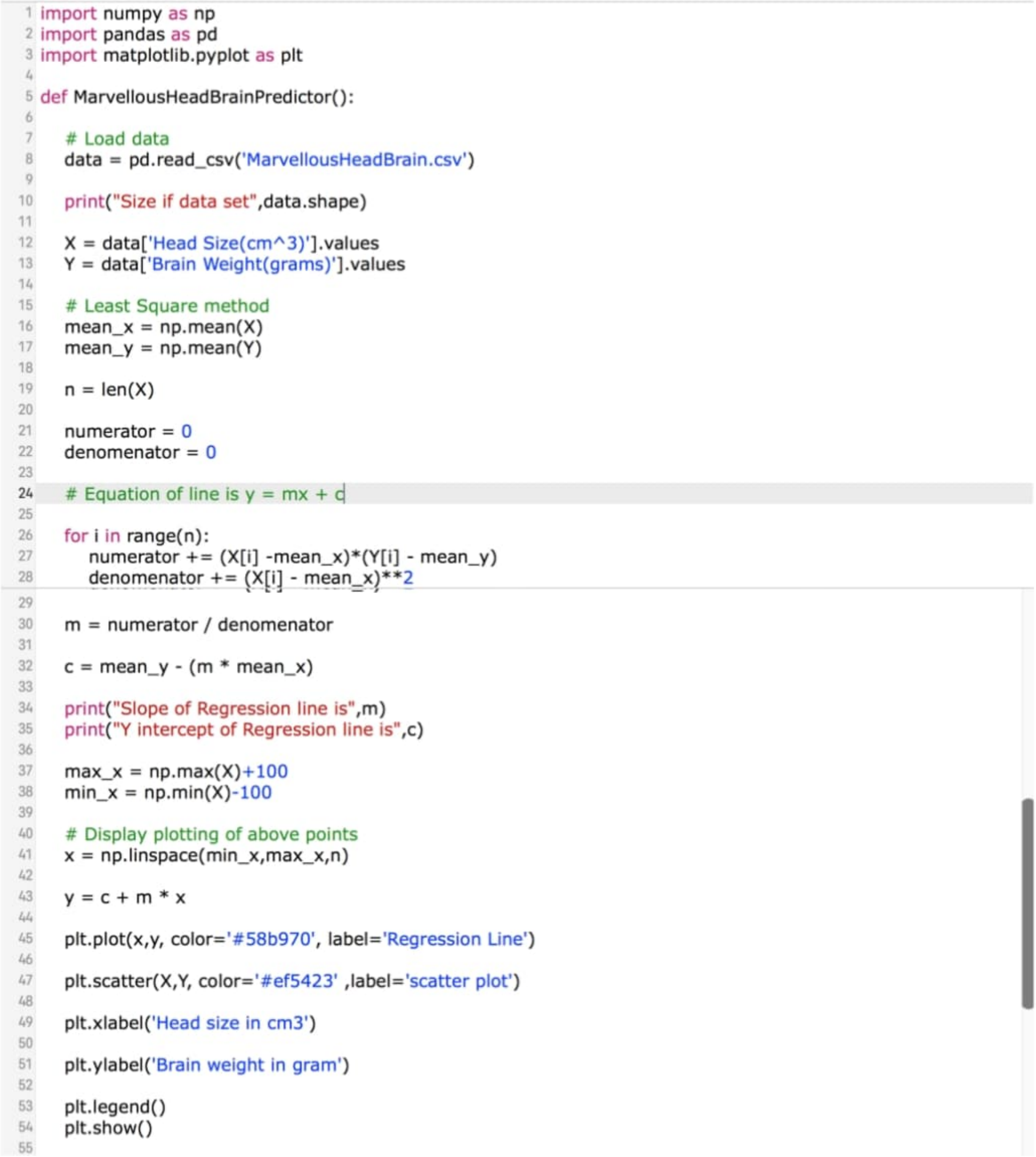
Gender	Age Range	Head Size(cm^3)	Brain Weight(grams)
1	1	4512	1530
1	1	3738	1297
1	1	4261	1335
1	1	3777	1282
1	1	4177	1590
1	1	3585	1300
1	1	3785	1400
1	1	3559	1255
1	1	3613	1355
1	1	3982	1375
1	1	3443	1340

Above data set contains information about Head and brain size depends on gender and age.

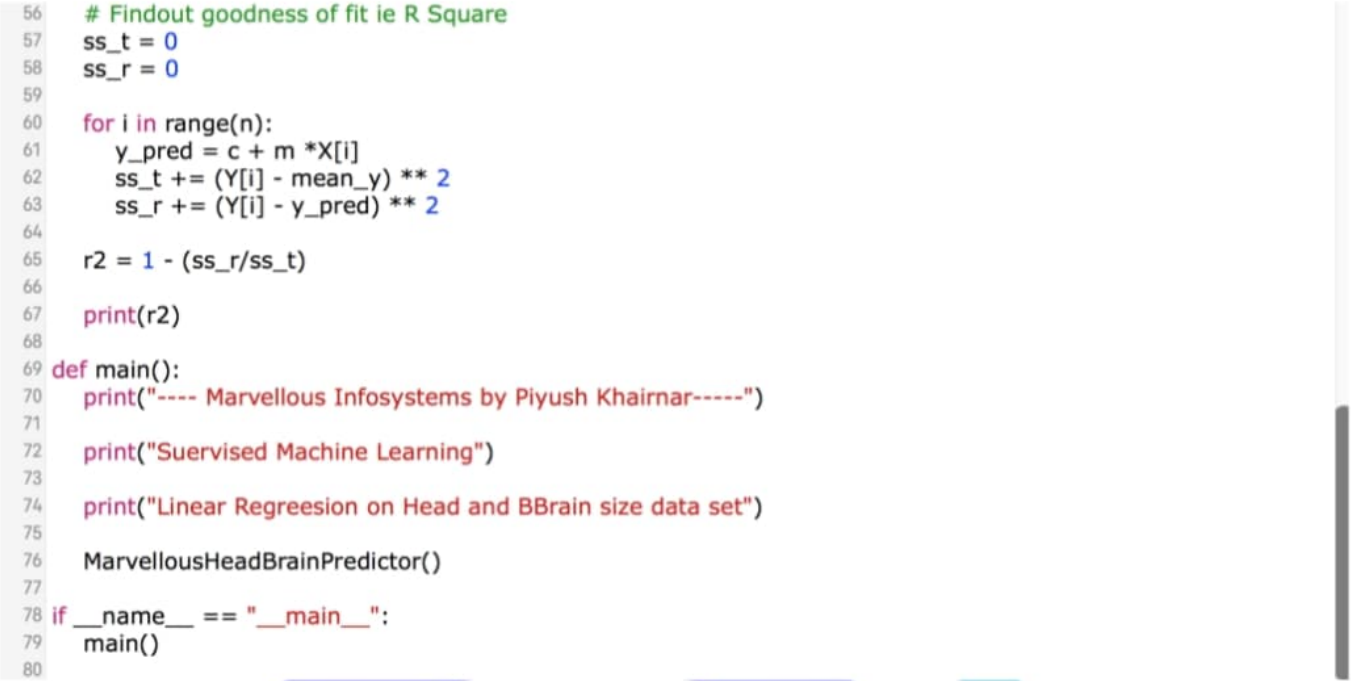
Consider below characteristics of Machine Learning Application :

Classifier :	**Linear Regression**
DataSet :	**Head Brain Dataset**
Features :	**Gender, Age, Head size, Brain weight**
Labels :	**-**
Training Dataset :	**237**

Consider below application which uses User defines Linear Regression algorithm to train above data set.

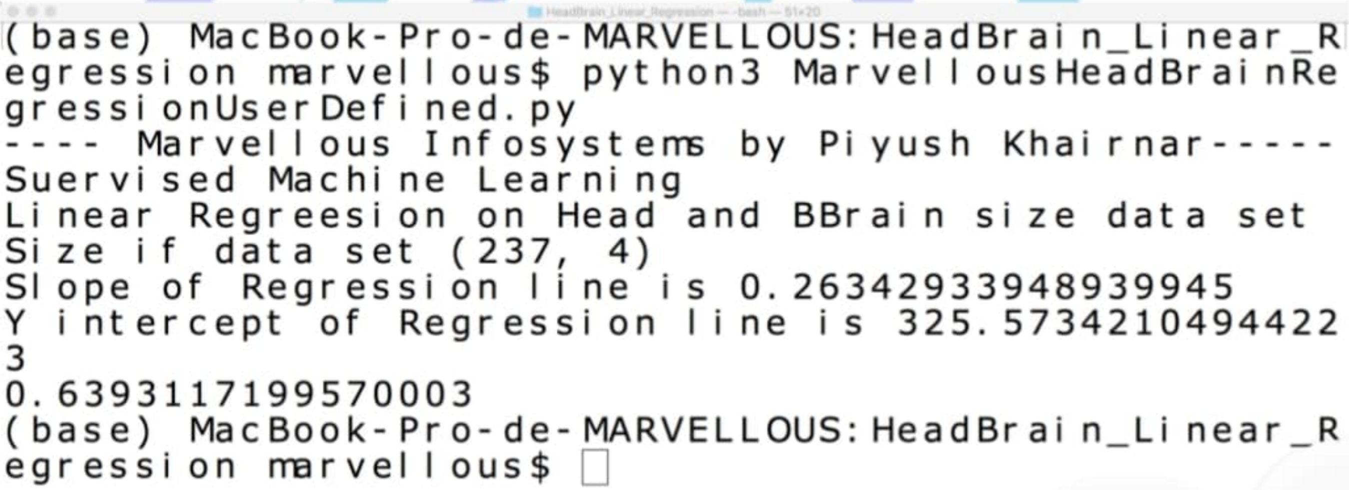


```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def MarvellousHeadBrainPredictor():

    # Load data
    data = pd.read_csv('MarvellousHeadBrain.csv')

    print("Size if data set",data.shape)

    X = data['Head Size(cm^3)'].values
    Y = data['Brain Weight(grams)'].values

    # Least Square method
    mean_x = np.mean(X)
    mean_y = np.mean(Y)

    n = len(X)

    numerator = 0
    denomenator = 0

    # Equation of line is y = mx + c

    for i in range(n):
        numerator += (X[i] -mean_x)*(Y[i] - mean_y)
        denomenator += (X[i] - mean_x)**2

    m = numerator / denomenator

    c = mean_y - (m * mean_x)

    print("Slope of Regression line is",m)
    print("Y intercept of Regression line is",c)

    max_x = np.max(X)+100
    min_x = np.min(X)-100

    # Display plotting of above points
    x = np.linspace(min_x,max_x,n)

    y = c + m * x

    plt.plot(x,y, color='#58b970', label='Regression Line')

    plt.scatter(X,Y, color='#ef5423' ,label='scatter plot')

    plt.xlabel('Head size in cm3')

    plt.ylabel('Brain weight in gram')

    plt.legend()
    plt.show()

```



```python
    # Findout goodness of fit ie R Square
    ss_t = 0
    ss_r = 0

    for i in range(n):
        y_pred = c + m *X[i]
        ss_t += (Y[i] - mean_y) ** 2
        ss_r += (Y[i] - y_pred) ** 2

    r2 = 1 - (ss_r/ss_t)

    print(r2)

def main():
    print("---- Marvellous Infosystems by Piyush Khairnar-----")

    print("Suervised Machine Learning")

    print("Linear Regreesion on Head and BBrain size data set")

    MarvellousHeadBrainPredictor()

if __name__ == "__main__":
    main()
```

Output of above application



```
(base) MacBook-Pro-de-MARVELLOUS:HeadBrain_Linear_R
egression marvellous$ python3 MarvellousHeadBrainRe
gressionUserDefined.py
---- Marvellous Infosystems by Piyush Khairnar-----
Suervised Machine Learning
Linear Regreesion on Head and BBrain size data set
Size if data set (237, 4)
Slope of Regression line is 0.26342933948939945
Y intercept of Regression line is 325.5734210494422
3
0.6393117199570003
(base) MacBook-Pro-de-MARVELLOUS:HeadBrain_Linear_R
egression marvellous$ 
```

Graph of above data set

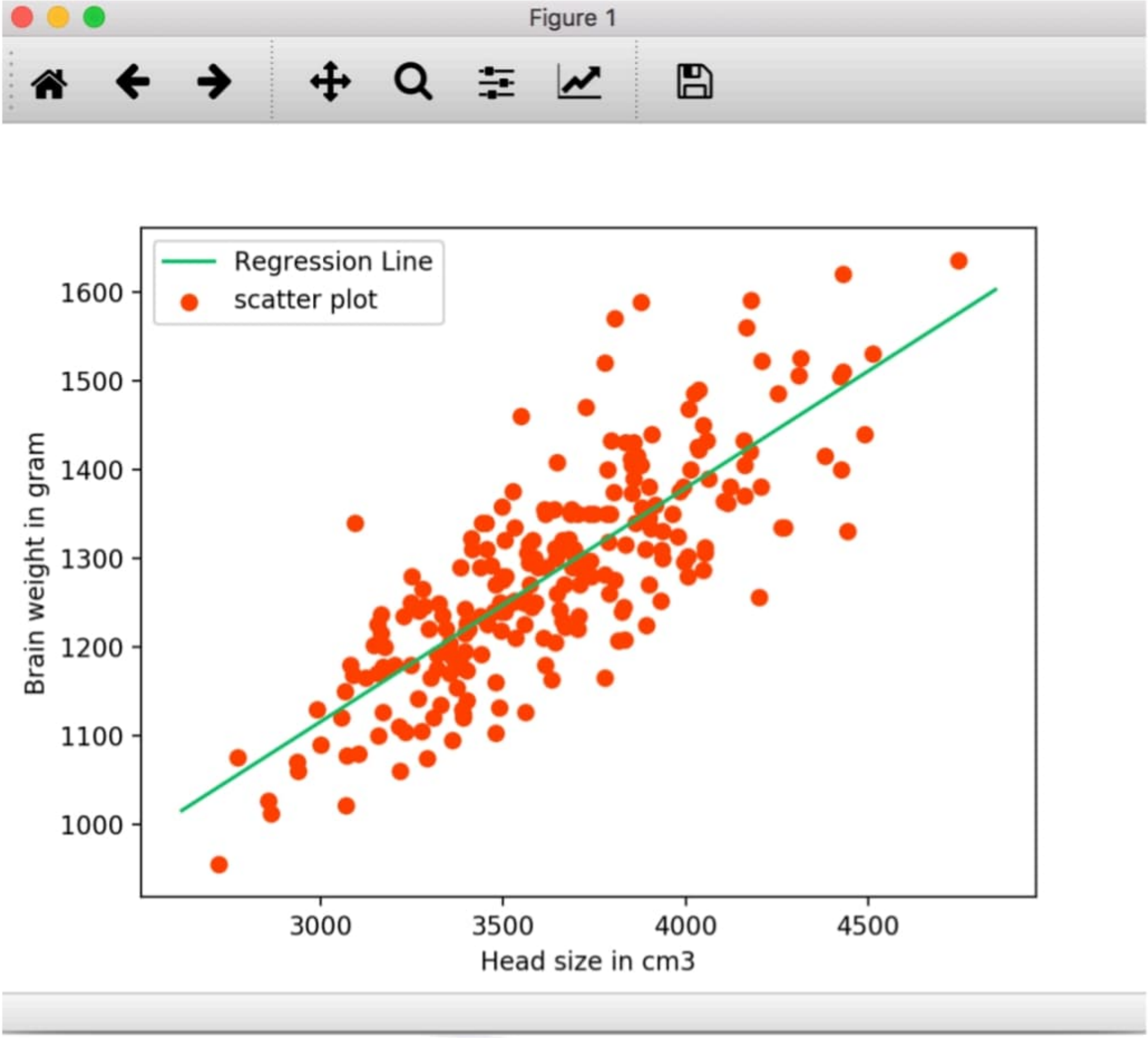


Figure 1
Regression Line
scatter plot
Brain weight in gram
Head size in cm3