

EXP. NO: 8

A. PYTHON PROGRAM TO IMPLEMENT

Date: 9/8/23. SIMPLE LINEAR REGRESSION, USING LEAST SQUARE METHOD

Aim:

$$(8 \times (m + b - y)) \text{ max } = 1 - 23$$

$$(8 \times (b + y - m)) \text{ max } = 1 - 22$$

TO implement a python program for constructing a simple linear regression using Least square method.

Algorithm:

Step 1: Import necessary libraries:

* Pandas for data manipulation and matplotlib.pyplot for plotting

Step 2: Read the dataset:

* Use the pandas 'read_csv' function to read the dataset.

Step 3: Prepare the data:

* Extract the independent variable (x) and dependent variable from the dataset.

* Reshape x and y to be 2D arrays if needed.

Step 4: Calculate the mean:

* Calculate the mean of x and y .

Step 5: Calculate the coefficient.

* Calculate the slope (m) using the formula:

$$m = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

* Calculate the intercept (b) using formula: $b = \bar{y} - m\bar{x}$.

Step 6: Make Predictions:

* Use the calculated slope and intercept to make prediction for each x value.

$$\hat{y} = mx + b.$$

Program:

```
import pandas as pd  
import matplotlib.pyplot as plt  
import numpy as np  
data = pd.read_csv('headbrain.csv')  
X, Y = np.array(list(data['Head size(cm^3)'])),  
    np.array(list(data['Brain weight(grams)']))  
print(X[:5], Y[:5])  
[4512 3738 4261 3777 4177] [1530 1297 1335 1282 159]  
def get_line(x, y):  
    x_m, y_m = np.mean(x), np.mean(y)  
    print(x_m, y_m)  
    x_d, y_d = x - x_m, y - y_m  
    m = np.sum(x_d * y_d) / np.sum(x_d ** 2)  
    c = y_m - (m * x_m)  
    print(m, c)  
    return lambda n: m * n + c  
lin = get_line(X, Y)  
x = np.linspace(np.min(X) - 100, np.max(X) + 100, 1000)  
y = np.array([lin(n) for n in x])  
plt.plot(x, y, color='red', label='Regression Line'),  
plt.scatter(X, Y, color='green', label='Scatter Plot'),  
plt.xlabel('Head size(cm^3)'),  
plt.ylabel('Brain weight(grams)'),  
plt.legend()  
plt.show()
```

Step 7: Plot the regression line:

* Plot the original data points (x_i, y_i) as a scatter plot.

* Plot the regression line $(x_i, \text{predicted } y)$ as line plot.

Step 8: Calculate the R-squared value:

* Calculate the total sum of squares $TSS = \sum_{i=1}^n (y_i - \bar{y})^2$.

* Calculate the residual sum of squares (RSS) using the formula

$$RSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

* R-squared value $R^2 = 1 - \frac{RSS}{TSS}$, where n is the number of data points.

Step 9: Display the results.

* Print the slope, intercept, & R-squared value.

Step 10: Complete the program:

* Combine all the steps into a Python program.

```
(work - N) * work / N = 117.02 = work - Y
```

```
(work - Y) * work / N = 787.42 = work - X
```

```
[(E, 0) for E in range(0, 100)]
```

Dataset:

```
(os.path.join("C:\\"), "data\heart.csv") = df
```

```
{work - Y, work - X} if i in df
```

```
(os.path.join("C:\\"), "data\heart.csv") = df
```

```
(work - X) * work - Y = 1078.42
```

```
(work - Y) * work
```

```
(work - Y) * work - heart['age'] = M3
```

Result:

```
(M3, "P1", "Predicted heart@")
```

The Python program to implement "Pima" with least square method for the given head brain dataset is analyzed and the linear regression line is successfully implemented.

```
(0.02 * os.getcwd(), M3)
```

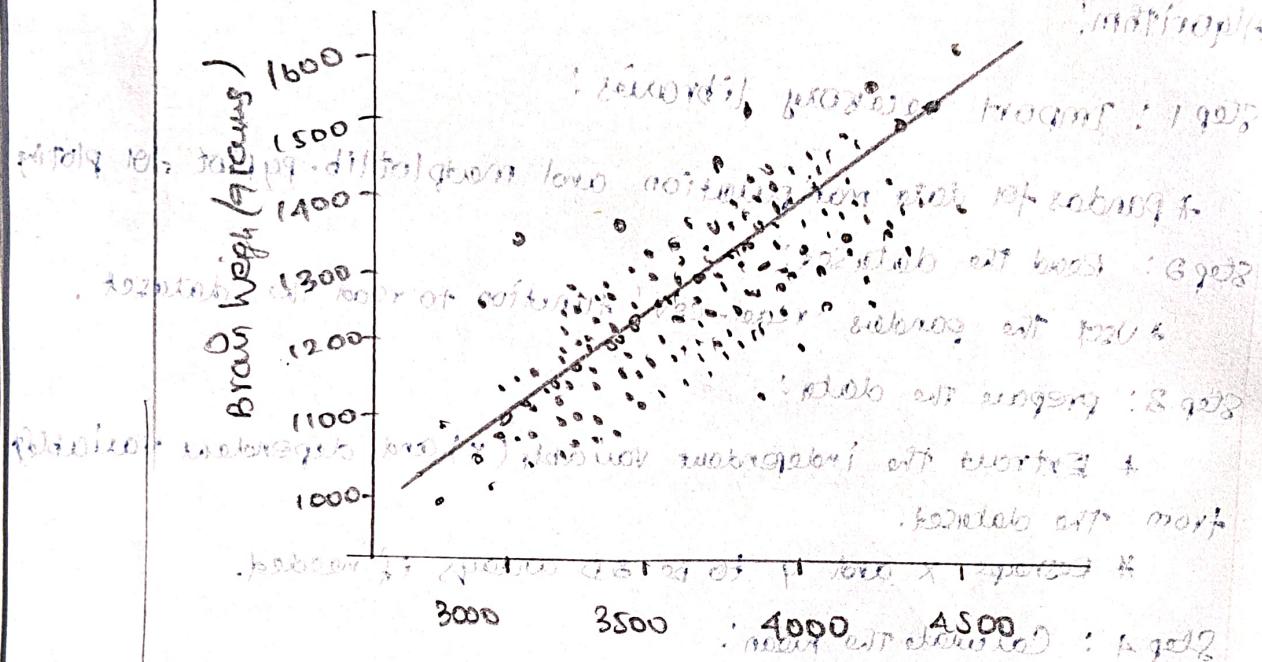
def get_slope(linreg_func, y):
 m = np.mean(y) + linreg_func(0, m)
 y_pred = np.array([linreg_func(x, m) for x in x])

$$SS_t = \sum ((y - \bar{y})^2)$$

$$SS_r = \sum ((y - \hat{y})^2)$$

return 1 - (SS_r / SS_t)

get_rvalue(linreg_func) using R-squared for
other regression models



Y. bird head size (cm) \hat{y} = b₀ + b₁x

parameters w.r.t. variables : 2 sets

sum of all points (x) against all variables *

$$\frac{(x - \bar{x})(\bar{x} - \bar{x})}{\sqrt{\sum (x - \bar{x})^2}} = n$$

$$\frac{(x - \bar{x})(\bar{x} - \bar{x})}{\sqrt{\sum (x - \bar{x})^2}}$$

sum of all points (y) against all variables *

sum of all points (y) against all variables *

start at first point and calculate w.r.t. all x

values X and ref variables :

$$d + xm = D$$