

Exp.no:2

A Python program to implement Simple linear regression using least square method

Aim:-

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 (Eg: headbrain.csv).
- * Store dataset in a variable (Eg: 'data').

Step 3: Prepare the data:

- * Extract the independent variable (X) and dependent variable (Y) from dataset.
- * Reshape X and Y to be 2D array if needed.

Step 4: Calculate the mean:

- * Calculate the mean of X and Y.

Step 5: Calculate the co-efficients:

* 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 the formula: $b = \bar{y} - m\bar{x}$.

Step 6: Make predictions:

* use the calculated slope and intercept to make predictions for each x value:

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

Step 7: Plot regression lines:

* plot the original data points (x, y) as

Scatter plot.

* plot the regression line ($x, \text{predicted}$)

as a line plot.

Step 9: Display the result:

* Print the slope intercept and R-Squared value.

Step 10:- Complete the program:

* Combine all the steps into a ^{python.} program.
* Run the program to perform simple linear regression on the dataset.

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 (cm3)']),
                 np.array(list(data['brain weight (grams)'])))
```

Print(X[:5], Y[:5])

[4512 3798 4261 3777 4177] [1530 1297
1335 1282 1590]

```
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("Slope(m):", m, "Intercept(c):", c)
```

```
    return lambda x: m * x + c
```

```
# Get regression function
```

```
lin = get_line(x, y)
```

```
# plotting
```

```
x = np.linspace(np.min(x) - 100, np.max(x) + 100, 1000)
```

```
y = np.array([lin(val) for val in x])
```

```
plt.plot(x, y, color='red', label='Regression line')
```

```
plt.scatter(x, y, color='green', label='Data points')
```

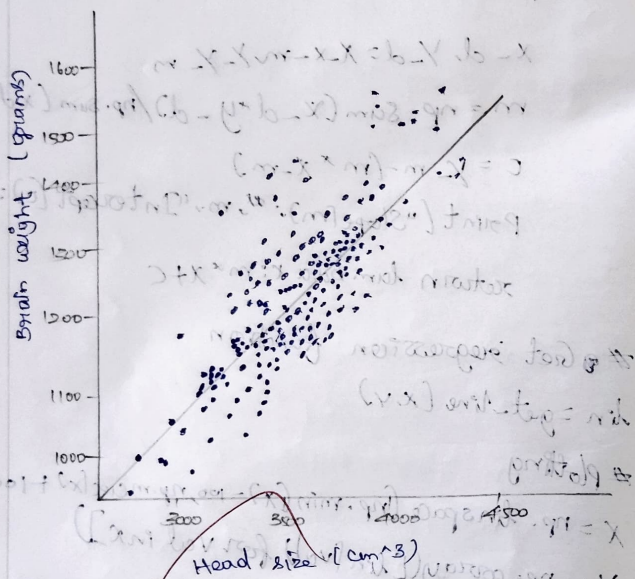
```
plt.xlabel('Head size (cm³)')
```

```
plt.ylabel('Brain Weight (grams)')
```

```
plt.legend()
```

```
plt.show()
```

$$f(x) \text{ recomb. } q(x, y) \text{ recomb. } q(x, y) = \alpha x + (1-\alpha)y$$

$$(1-\alpha) \cdot f(x) + \alpha \cdot f(y)$$


Regression line

- Scatter plot

```
def get_error(line_func, x, y):
```

$$-Y_{-m} = n p \cdot \text{mean}(Y)$$
$$Y_{pred} = np \cdot \text{array}([line - func(-) \text{ for } -in x])$$
$$SS_t = np \cdot \text{sum}((y - \bar{y})^2)$$
$$SS_{\pi} = np \cdot \text{sum}[(y - \hat{y}_{\text{pred}})^2 \cdot z]$$
$$x_2 = 1 - (33 - \pi(55 - t))$$

27. 12

Print R^2 from manual method

```
Print("manual R^2 score:", get_score((i, k, y))
```

~~# using sklearn linear regression~~

~~# using sklearn~~ ~~linear regression~~
from sklearn.linear_model import LinearRegression

$-x - \text{reshape} = x \cdot \text{reshape}(\text{den}(x), 1) \# \text{reshape to } 20 \text{ for Sklearn.}$

$$\pi_{\text{reg}} = L_{\text{reg}} \text{ and } \text{Reg} \pi_{\text{reg}} = \pi(1)$$
$$\text{reg. fit}(X_{\text{reshaped}}, y)$$

```
print("Skewness R^2 Score:" + negr.score(x_test, y))
```

function to compute R^2 score manually

(V * X - E) * (V * X - E)

(V * X - E) * (V * X - E)

(V * X - E) * (V * X - E)

(V * X - E) * (V * X - E)

(V * X - E) * (V * X - E)

(V * X - E) * (V * X - E)

(V * X - E) * (V * X - E)

function to compute R^2 score manually

(V * X - E) * (V * X - E)

(V * X - E) * (V * X - E)

(V * X - E) * (V * X - E)

(V * X - E) * (V * X - E)

(V * X - E) * (V * X - E)

(V * X - E) * (V * X - E)

(V * X - E) * (V * X - E)

(V * X - E) * (V * X - E)

(V * X - E) * (V * X - E)

To implement python program for the logistic using svm as default

Algorithm

Step 1: Import Necessary libraries.

• Pandas for data manipulation

• sklearn model selection for train-test split

• sklearn.preprocessing for data preprocessing

• matplotlib.pyplot for plotting

Step 2: Read the Dataset:

• Use pandas to read the svm.csv dataset into a Dataframe.

Step 3: Preprocess the data:

• Split the dataset into training and testing sets.

Result:

Thus, the python program to implement simple linear regression using least square method for the given head brain dataset is analyzed and linear regression line is constructed successfully.