Linear Regression

It is method that defines relationship between dependent variable (y) and independent variable (x).
GOAL: The goal is to draw a best fit line between x and y that estimates the relationship between x and y.
The equation that it follows is:

Y = mX + b , where

Y = output(dependent variable)
X = features (independent variable)
m = scale factor or coefficient
b = bias coefficient -> gives an extra degree of freedom to this model

the total error of the linear model is the sum of the error of each point. I.e. , $\sum_{i=1}^{n} r_i^2$

r*i* = Distance between the line and **i***th* point.

n =Total number of points.

```
#importing libraries

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


dataset = pd.read_csv('dataset.csv')
#printing dataset size on the basis of row and column
# print(dataset.shape)

#printing some top dataset values
# print(dataset.head(10))
```

Importing dataset and checking the dataset size using shape function and printing the top 10 value using the head(10) function.

Output will be

```
C:\Users\sajid\PycharmProjects\Linear_Regression\venv\Scripts\python.exe C:/Users/sajid/PycharmProjects/Linear_Regression/venv/Include/linear_regression.py
(237, 4)
   Gender  Age Range  Head Size(cm^3)  Brain Weight(grams)
0       1          1             4512                 1530
1       1          1             3738                 1297
2       1          1             4261                 1335
3       1          1             3777                 1282
4       1          1             4177                 1590
5       1          1             3585                 1300
6       1          1             3785                 1400
7       1          1             3559                 1255
8       1          1             3613                 1355
9       1          1             3982                 1375

Process finished with exit code 0
```

Now we have to find the relationship between head size and brain weight so we are taking this two features into two variables X & Y.

```
#initializing inputs and outputs
X = dataset['Head Size(cm^3)'].values
Y = dataset['Brain Weight(grams)'].values

# print(len(X))
# print('\n')
# print(len(Y))

#finding mean of input and output
x_mean = np.mean(X)
y_mean = np.mean(Y)

# print(x_mean)
# print('\n')
# print(y_mean)

#total number of values
n = len(X)

numerator = 0
denominator = 0
for i in range(n):
    numerator += (X[i] - x_mean) * (Y[i] - y_mean)
    denominator += (X[i] - x_mean) ** 2
b1 = numerator / denominator
b0 = y_mean - (b1 * x_mean)
```

```python
# Print coefficients
print(b1, b0)
```

Above we have solved two equations. They are:

$$\beta_1 = \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{m}(x_i - \bar{x})^2}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

These are coefficients and the output will be:
B1 = 0.26342933948939945
B0 = 325.57342104944223
Then we have to find the straight line of the following equation:

$$Y = \beta_0 + \beta_1 X$$

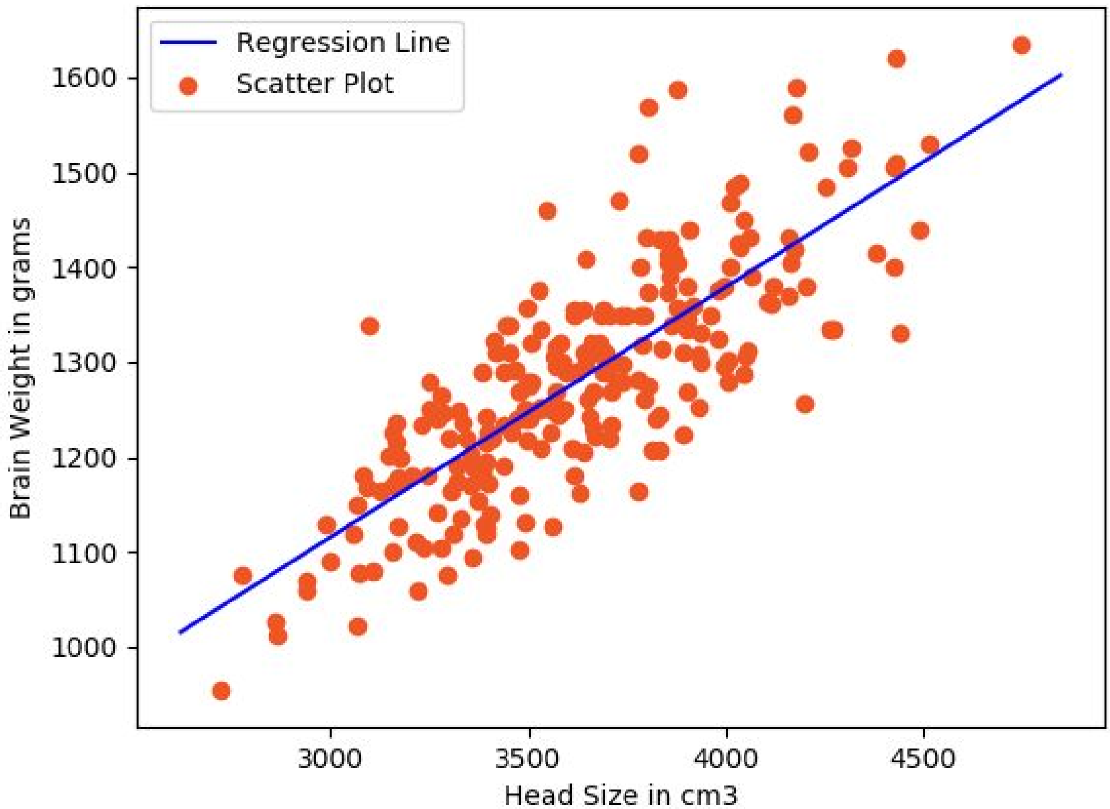
It means our expected functions will be:

$$BrainWeight = 325.573421049 + 0.263429339489 * HeadSize$$

```python
# Plotting Values and Regression Line

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

# Calculating line values x and y
x = np.linspace(min_x, max_x, 1000)
y = b0 + b1 * x
```

Now plotting on graph.

```python
# Ploting Line
plt.plot(x, y, color='blue', label='Regression Line')
# Ploting Scatter Points
plt.scatter(X, Y, c='#ef5423', label='Scatter Plot')

plt.xlabel('Head Size in cm3')
plt.ylabel('Brain Weight in grams')
plt.legend()
plt.show()
```



To know about linespace visit
https://www.numpy.org/devdocs/reference/generated/numpy.linspace.html

Now we will find the accuracy. At first we will use RMSE(root mean squared error). The equation is:

$$RMSE = \sqrt{\sum_{i=1}^{m} \frac{1}{m} (\hat{y_i} - y_i)^2}$$

Here we have to predict y_pred value for each ith value of X. then we have to find the difference of (y[i] - y_pred)^2

```
# Calculating Root Mean Squares Error
rmse = 0
for i in range(n):
    y_pred = b0 + b1 * X[i]
    rmse += (Y[i] - y_pred) ** 2
```

```
rmse = np.sqrt(rmse/n)
print(rmse)
```

Another way to find accuracy is R^2 score. The equation is:

$$SS_t = \sum_{i=1}^{m} (y_i - \bar{y})^2$$

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

$$R^2 \equiv 1 - \frac{SS_r}{SS_t}$$

```
# Calculating R^2 Error
ss_t = 0
ss_r = 0
for i in range(n):
    y_pred = b0 + b1 * X[i]
    ss_t += (Y[i] - y_mean) ** 2
    ss_r += (Y[i] - y_pred) ** 2
r2 = (1 - (ss_r/ss_t))*100
print(r2)
```

Reference Sites:

1. https://towardsdatascience.com/linear-regression-from-scratch-cd0dee067f72
2. https://mubaris.com/posts/linear-regression/