

Aim:

To implement a Python program using  
Univariate, bivariate and multivariate  
Regression features of a given Iris  
dataset.

Algorithm:

Step 1: Import necessary libraries:

\* pandas for data manipulation, numpy  
for numerical operations and matplotlib,  
pyplot for plotting.

Step 2: Read the dataset:

use the pandas 'read\_csv' function  
to read the dataset.

Store the dataset in a variable

(eg. 'data').

Store the dataset in a variable

(eg. 'data').

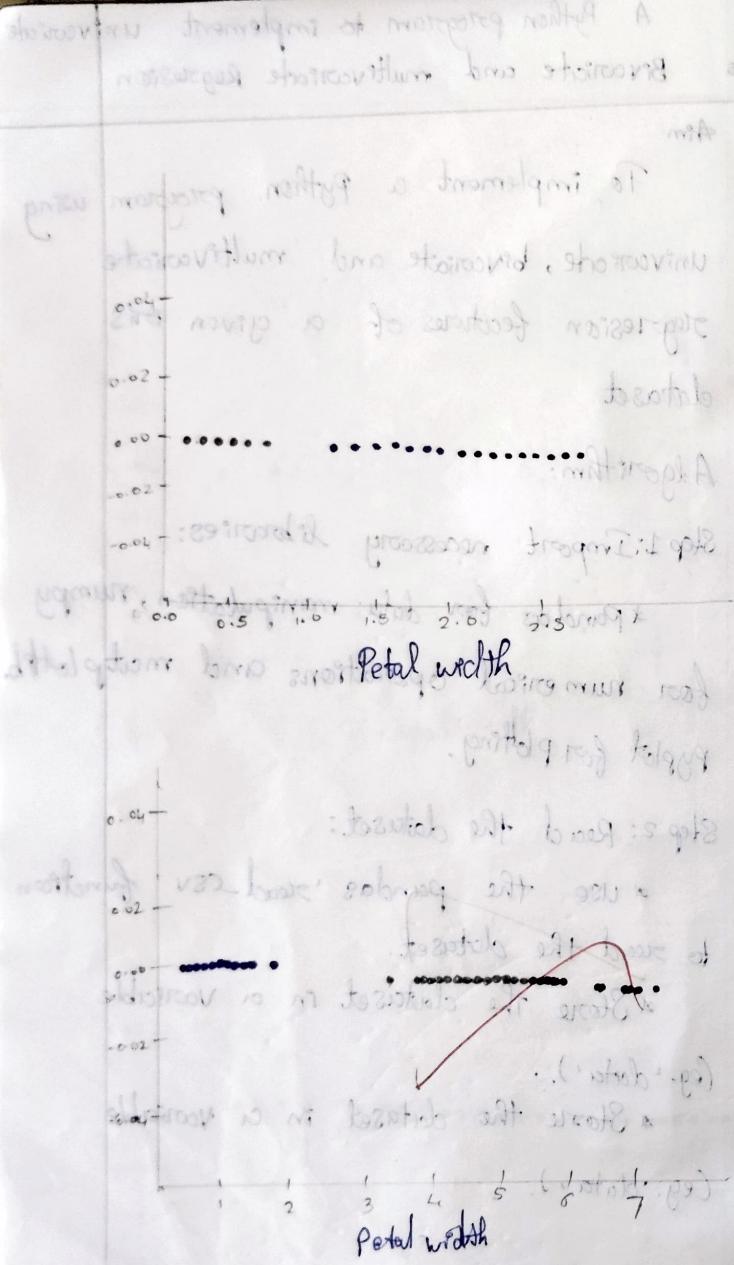
using



Sepal width



Sepal length



Step 3: Prepare the data:

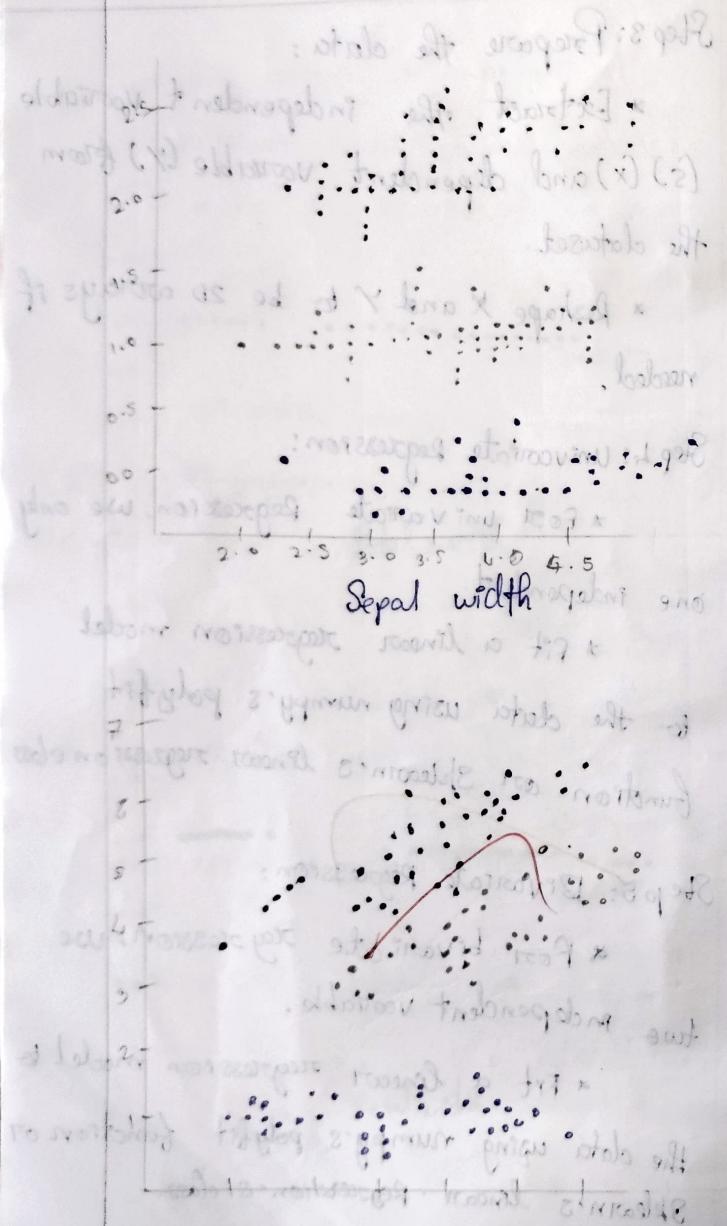
- \* Extract the independent variable ( $s$ ) ( $x$ ) and dependent variable ( $y$ ) from the dataset.
  - \* Reshape  $X$  and  $Y$  to be 2D arrays if needed.

## Step 4: Univariate Regression

- \* For univariate regression, we only have one independent variable
  - \* fit a linear regression model to the data using numpy's polyfit function or Scikit-learn's linear regression class.

## Step 5: Bivariate Regression:

- \* For bivariate regression, we have two independent variables.
    - a Fit a linear regression model to the data using numpy's polyfit function or sklearn's LinearRegression class.



Step 6: plot the results:

- \* For univariate regression, plot the original data points  $(x, y)$  as a scatter plot and the regression line as a line plot.
- \* For multivariate regression, plot the predicted values against the actual values.

Step 7: Display the result:

- \* Print the Co-efficients (slope) and Intercept for each regression model.
- \* Print the R-Squared value for each regression model.

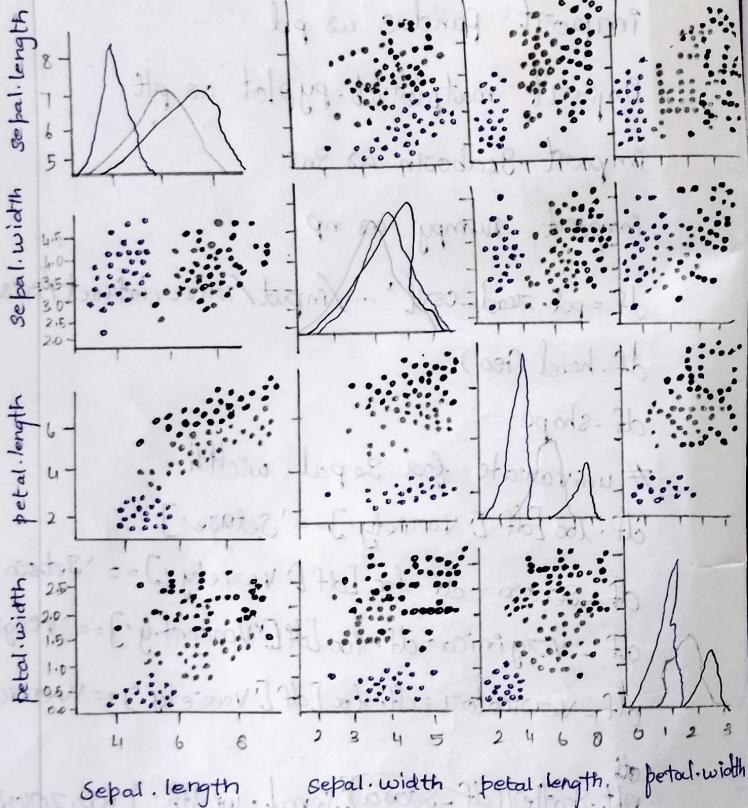
Step 8: Complete the program:

- \* Combine all the steps into a python program.
- \* Run the program to perform univariate, bivariate and multivariate regression on the data set.

## Program :

```
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
import numpy as np  
df = pd.read_csv('.../input/iris-dataset/iris.csv')  
df.head(50)  
df.shape  
# univariate for Sepal width  
clf.loc[df['Variety'] == 'Setosa']  
df_setosa = clf.loc[df['Variety'] == 'Setosa']  
df_virginica = clf.loc[df['Variety'] == 'Virginica']  
df_versicolor = clf.loc[df['Variety'] == 'Versicolor']  
#  
plt.scatter(df_setosa['Sepal.Width'], np.zeros(len(df_setosa['Sepal.Width'])))  
plt.scatter(df_versicolor['Sepal.Width'], np.zeros(len(df_versicolor['Sepal.Width'])))  
plt.xlabel('Sepal.Width')  
plt.show()
```

Multivariate :-



# univariate for sepal length

clf.loc [df['variety'] == 'Setosa']  
df\_Setosa = df.loc [df['variety'] == 'Setosa']

df\_Versicolor = df.loc [df['variety'] == 'Versicolor']

df\_Virginica = df.loc [df['variety'] == 'Virginica']

df\_Versicolor = df.loc [df['variety'] == 'Versicolor']

plt.scatter(df\_Setosa['Sepal.length'], np.zeros\_like(df\_Setosa['Sepal.length']))

plt.scatter(df\_Versicolor['Sepal.length'], np.zeros\_like(df\_Versicolor['Sepal.length']))

plt.xlabel('Sepal.length')

plt.show()

# univariate for petal width

clf.loc [df['variety'] == 'Setosa']

df\_Setosa = df.loc [df['variety'] == 'Setosa']

df\_Virginica = df.loc [df['variety'] == 'Virginica']

df\_Versicolor = df.loc [df['variety'] == 'Versicolor']

plt.scatter(df\_Setosa['Sepal.width'], np.zeros\_like(df\_Setosa['Sepal.width']))

plt.scatter(df\_Virginica['Sepal.width'], np.zeros\_like(df\_Virginica['Sepal.width']))

plt.scatter(df\_Versicolor['Sepal.width'], np.zeros\_like(df\_Versicolor['Sepal.width']))

- plt.Scatter (df - versicolor['Sepal. width'], np.zeros like (df - virginica['Sepal. width']))
- plt.xlabel ('Sepal width')
- plt.show()
- # univariate for petal length
- df. loc [df ['variety'] == 'setosa']
- df - setosa = df. loc [df ['variety'] == 'setosa']
- df - virginica = df. loc [df ['variety'] == 'virginica']
- df - versicolor = df. loc [df ['variety'] == 'versicolor']
- plt.Scatter (df - virginica['petal.length'], np.zeros like (df - versicolor['petal.length']))
- plt.scatter (df - versicolor['petal.length'], np.zeros like (df - versicolor['petal.length']))

~~plt. xlaide (petal length, )~~

```
plt.show()
```

# bivariate sepal width vs petal width

sns.FacetGrid (df, hue = 'varietyp', size=5) <sup>et</sup> map  
(plt.Scatter, "sepal.width", "petal.width")  
add legend()

plt.show()

# bivariate Sepal.length vs petal.length

sns.FacetGrid (df, hue = 'varietyp', size=5) <sup>et</sup> map(plt.Scatter,  
"sepal.length", "petal.length")  
add legend()

plt.show()

# multivariate all the features

sns.pairplot (df, hue = 'varietyp', size=2)

Result:

Thus the python program to implement univariate, bivariate and multivariate regression ~~features~~ for the given iris dataset is analyzed and the features are plotted using scatter plot.