Ex. No: 1

Date: 26/7/25.

A PYTHON PROGRAM TO IMPLEMENT UNIVARIATE, BIVARIATE AND MULTIVARIATE REGRESION.

Aim:

To implement a python program using univariate, bivariate and multivariate regression features for 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 pandaset 'read_csv' function to read the dataset.

* Store the dataset in a variable.

Step 3: Prepare the data:

* Extract the independant variables (X) and dependant variable (Y) from the dataset.

* Reshape X and Y to be 2D arrays if needed.

Step 4: Univariate Regression:

* For univariate regression, use only one independent variables.

* Make predictions using the model.

* Calculate the R-squared value to evaluate the model's performance.

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(150)
df.shape
#univariate for sepal width.
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_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.xlabel('sepal.width')
plt.show().

#univariate for sepal 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_Setosa['sepal.length'], np.zeros_like(df_Setosa['sepal.length']))
plt.scatter(df_Virginica['sepal.length'], np.zeros_like(df_Virginica['sepal.length']))
plt.scatter(df_Versicolor['sepal.length'], np.zeros_like(df_versicolor['sepal.length']))
plt.xlabel('sepal.length')
plt.show()
```

Step 5 : Bivariate Regression :

* For bivariate regression use two independant variables
* Make predictions using the model.
* Calculated the R-squared values to evaluate the model's performance.

Step 6 : Multivariate Regression :

* For multivariate regression, use more than two independant variables.
* Make predictions using the model.
* Calculate the R-squared value to evaluate the model's performance.

Step 7 : Plot the resuts :

* For univariate regression, plot the original the data points (x, y) as a scatter plot and the regression line as a line plot.
* For bivariate regression, plot the original data points (x_1, x_2) as a 3D scatter plot and the regression plane.

Step 8 : Display the results :

* Print the coefficients (slope) and intercept for each regression model.
* Print the R-squared value for each regression model

Step 9 : Complete the program :

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

univariate for petal width

```
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_Setosa['petal.width'], np.zeros_like(df_Setosa['petal.width
Plt.scatter(df_Virginica['petal.width'], np.zeros_like(df_Virginica['petal.width
Plt.scatter(df_Versicolor['petal.width'], np.zeros_like(df_Versicolor['petal.width
Plt.xlabel('petal.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_Setosa['petal.length'], np.zeros_like(df_Setosa['petal.width']))
Plt.Scatter(df_virginica['petal.length'], np.zeros_like(df_virginica['petal.length']))
Plt.Scatter(df_versicolor['petal.length'], np.zeros_like(df_versicolor['petal.length']))
Plt.xlabel('petal.length')
Plt.show()
```

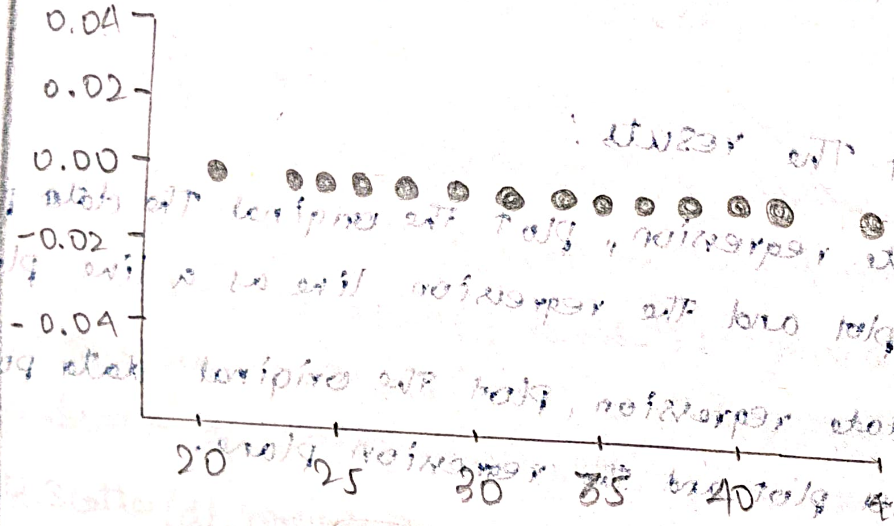
bivariate sepal.width vs petal.width.

```
sns.FacetGrid(df, hue='variety', size=5).map(plt.scatter, "sepal.width",
"petal.width").add_legend();
Plt.show()
```

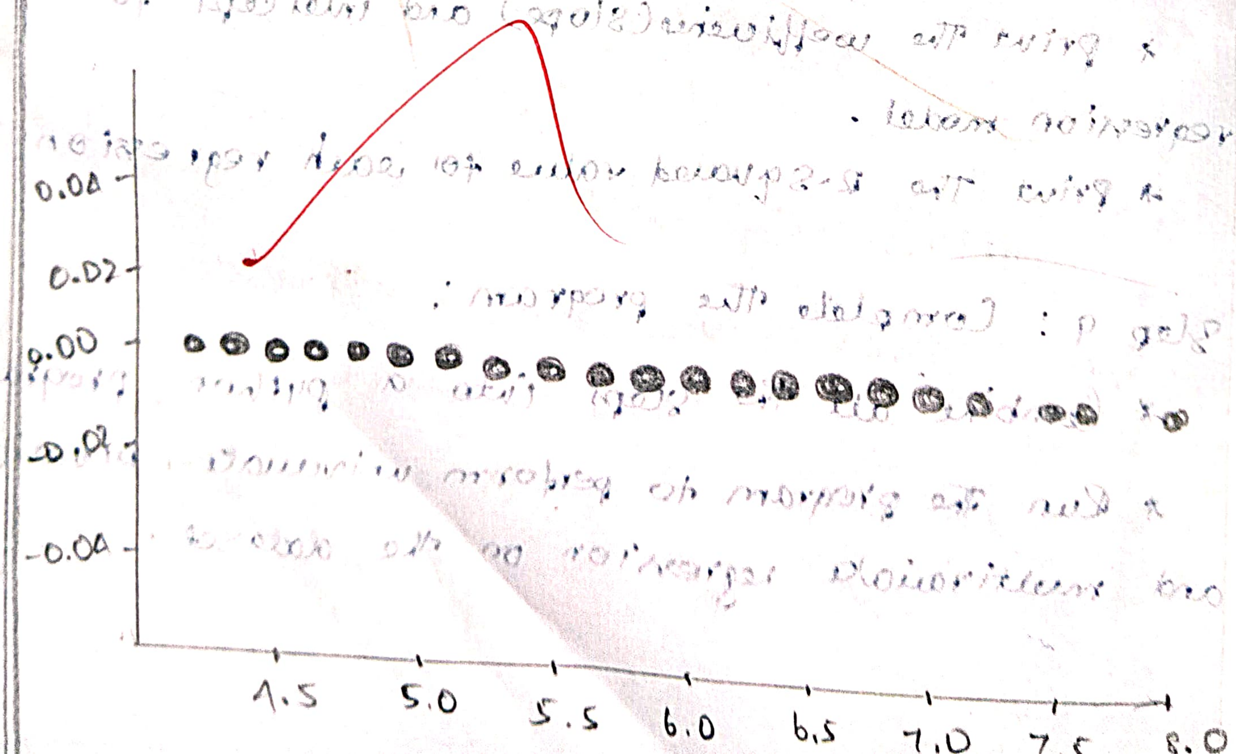
```
#bivariate sepal.length vs. petal.length:
sns.FacetGrid(df, hue='variety', size=5).map(plt.scatter,
"sepal.length", "petal_length").add_legend();
plt.show()
```

```
#multivariate all the features
sns.pairplot(df, hue='variety", size=2).
```

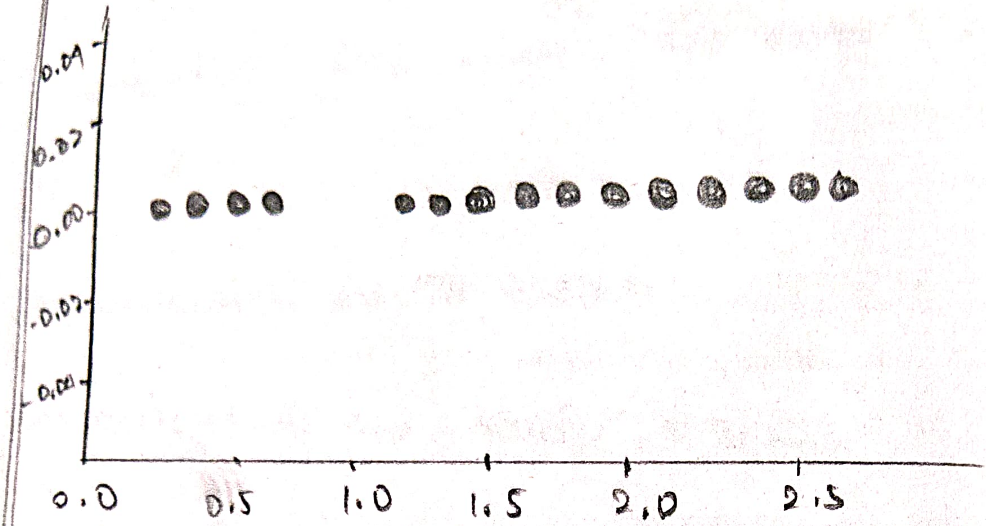
Sepal width:



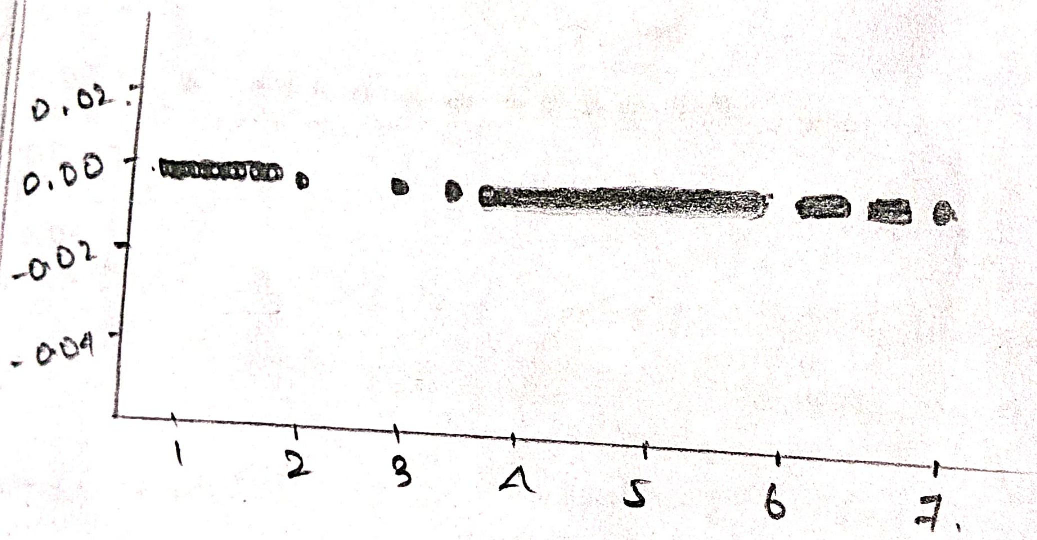
Sepal length:



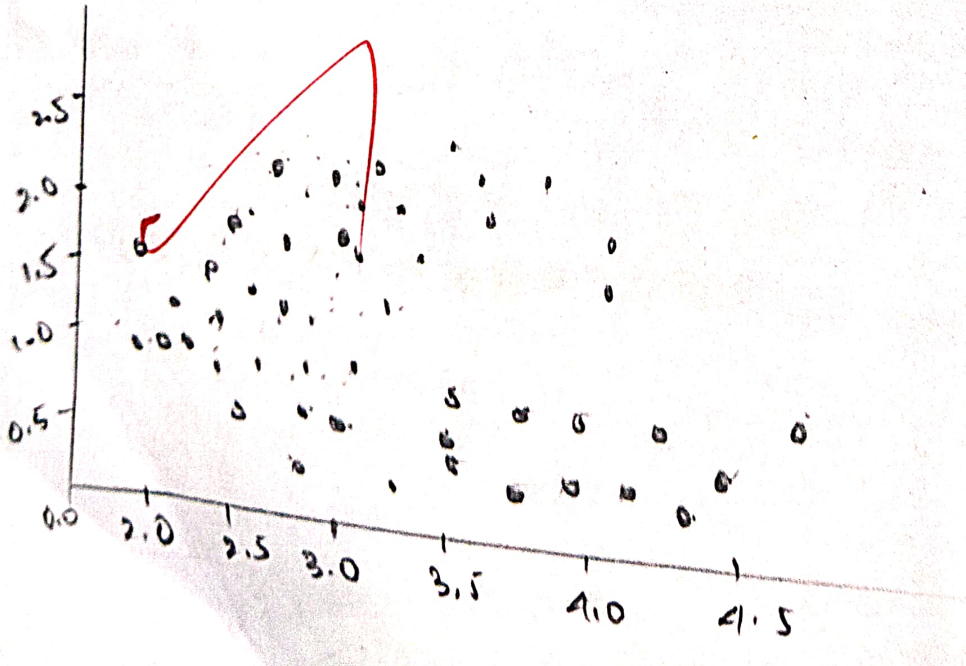
petal. width:

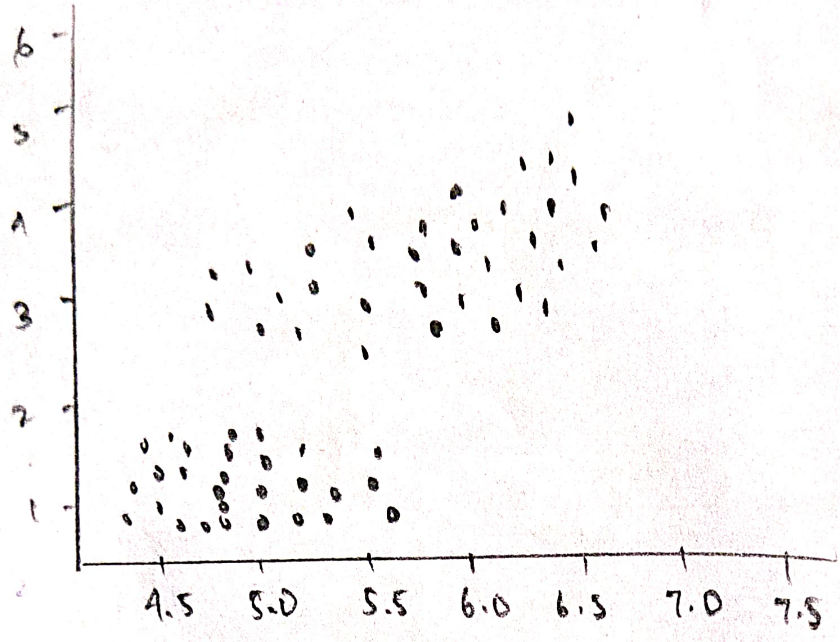


petal. length:

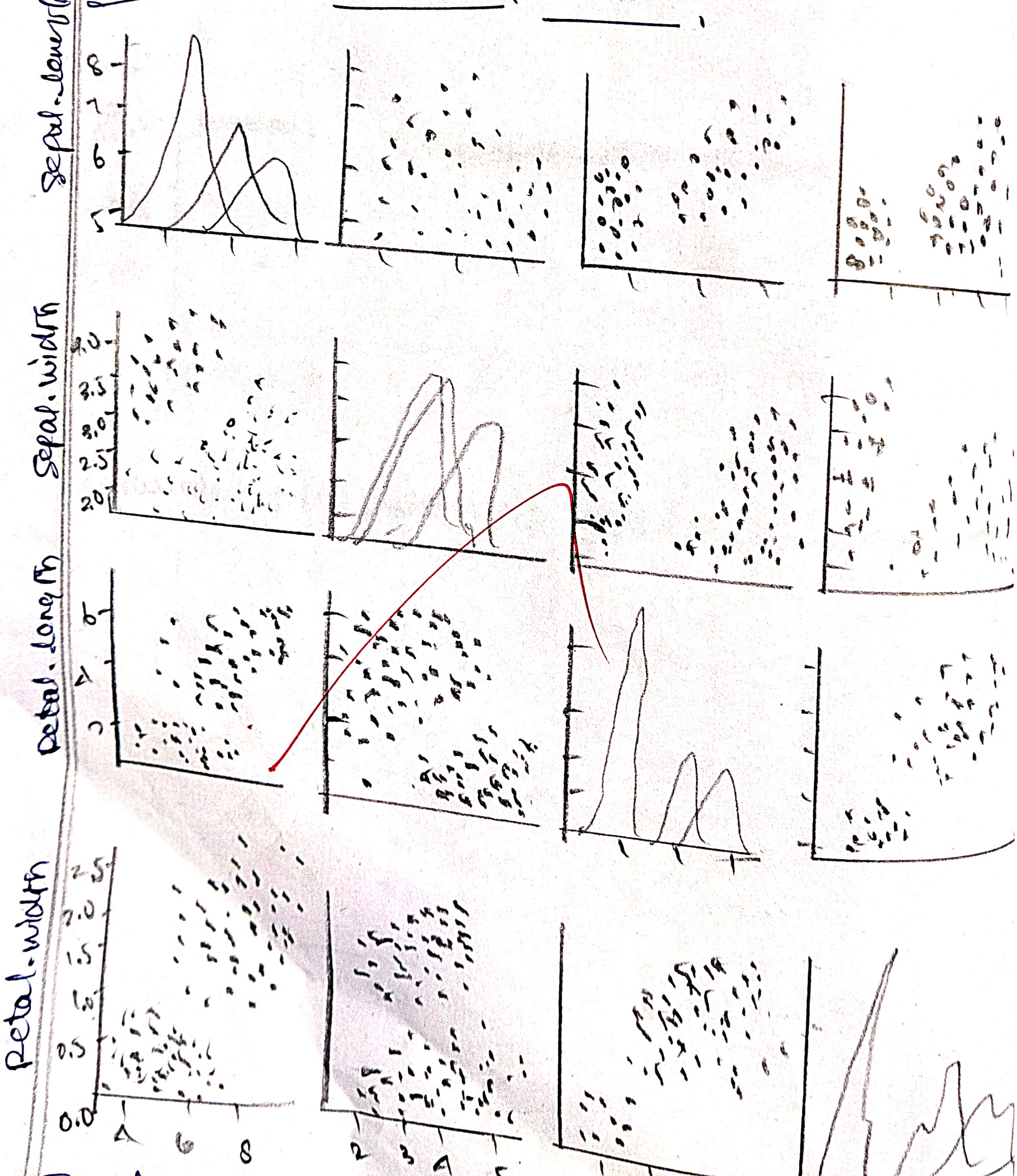


bivariate sepal, width vs petal width.





multivariate all The features :



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.

A 2/8/25.