

Exp. No: 01
24/07/2023

A Python program to implement univariate Bivariate and multivariate Regression

Aim:

To implement a Python program using univariate, bivariate and multivariate regression features of a given iris dataset.

Algorithm:

Step 1: Import necessary libraries:

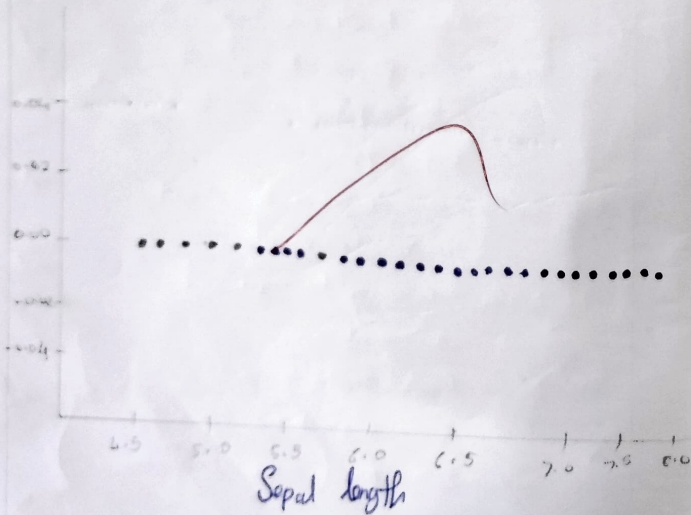
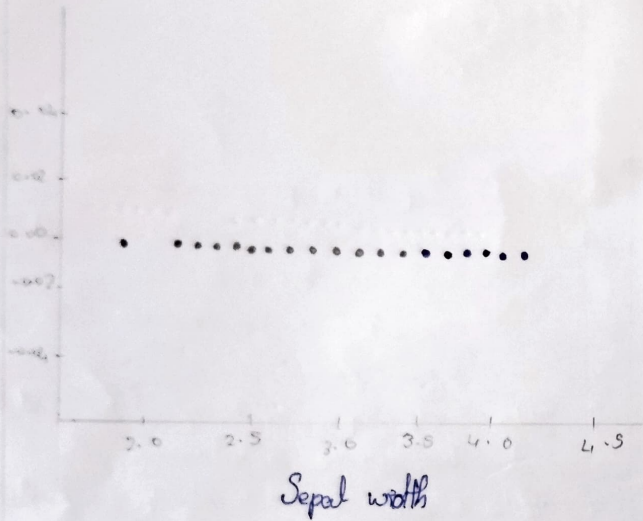
- * pandas for data manipulation, numpy for numerical operations and matplotlib.pyplot for plotting.

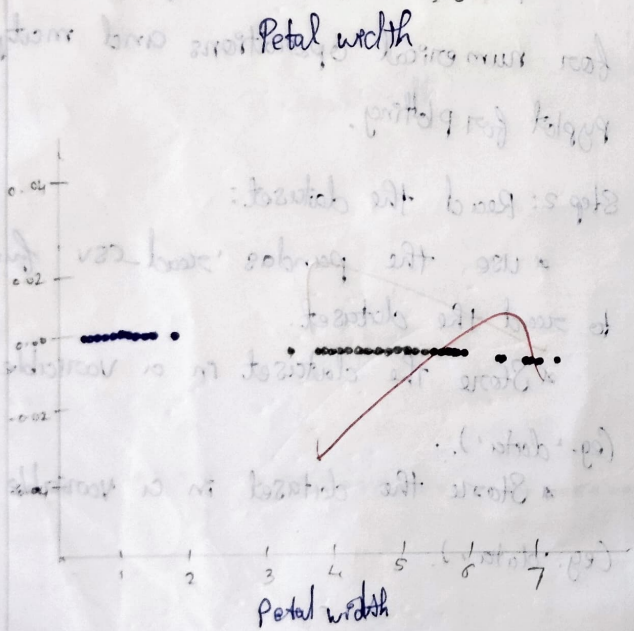
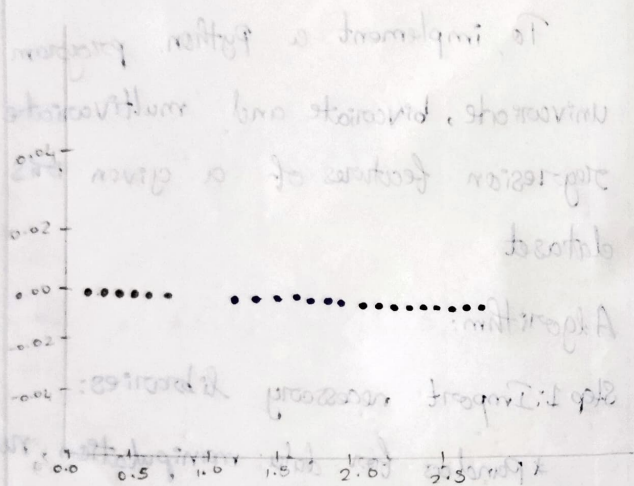
Step 2: Read the dataset:

- * 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').





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, use only one independent variable.

- * Fit a linear regression model to the data using numpy's `polyfit` function or sklearn's `LinearRegression` class.

Step 5: Bivariate Regression:

- * For bivariate regression, use two independent variables.

- * 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

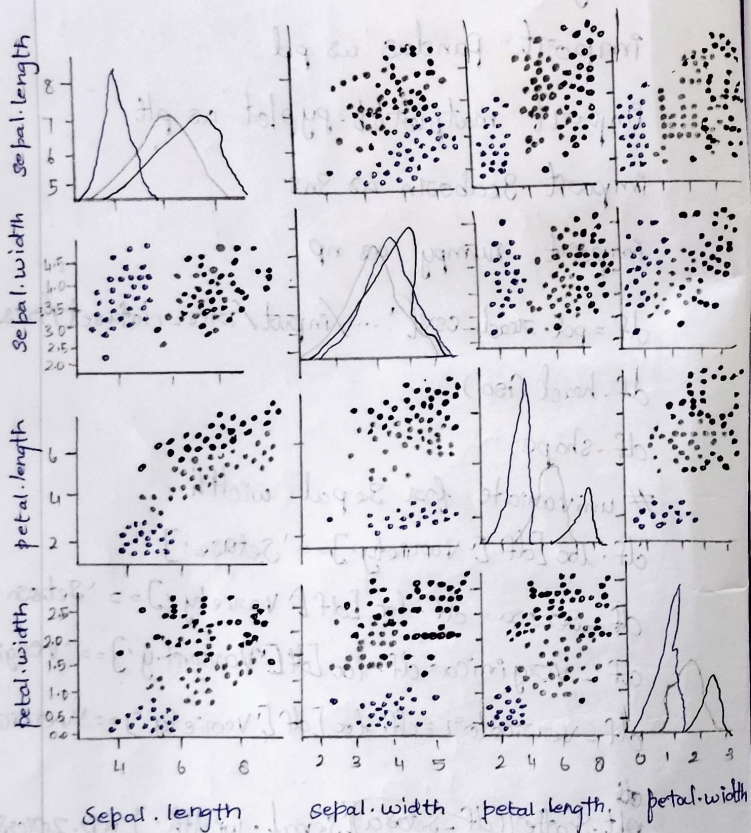
df = pd.read_csv('.../input/iris-dataset/iris.csv')
df.head(10)
df.shape
# univariate for sepal width
df.loc[df['variety'] == 'Setosa']
df - setosa = df.loc[df['variety'] == 'Setosa']
df - virginica = df.loc[df['variety'] == 'Virginica']
df - versicolor = df.loc[df['variety'] == 'Versicolor']

plt.scatter(df - setosa['sepal.width'], np.zeros - like
            (df - setosa['sepal.width']))

plt.scatter(df - versicolor['sepal.width'], np.zeros - like
            (df - versicolor['sepal.width']))

plt.xlabel('sepal.width')
plt.show()
```

Multi variate :-



univariate for sepal length

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

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

df_virginica = df.loc[df['variety'] == 'virginica']

df_versicolour = df.loc[df['variety'] == 'versicolour']

plt.scatter(df_setosa['sepal . length'], np.zeros_like(df_setosa['sepal . length']))

plt.scatter(df_versicolour['sepal . length'], np.zeros_like(df_versicolour['sepal . length']))

plt.xlabel('Sepal - length')

plt.show()

univariate for petal width

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

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

df_virginica = df.loc[df['variety'] == 'virginica']

df_versicolour = df.loc[df['variety'] == 'versicolour']

plt.scatter(df_setosa['sepal . width'], np.zeros_like(df_setosa['sepal . width']))

plt.scatter(df_virginica['sepal . width'], np.zeros_like(df_versicolour['sepal . width']))

plt.scatter(df['versicolor']['sepal.width'], np.zeros
like(df['versicolor']['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~~
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=True
)
plt.show()
```

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

```
plt.show()
```

multivariate all the features

```
sns.pairplot(df, hue='variety', 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.