Subject: Data Mining and Warehousing **SAP ID: 60004220253 – Devansh Mehta**

Experiment 04

Aim: Implementation of Linear Regression for Single Variate and Multi-variate

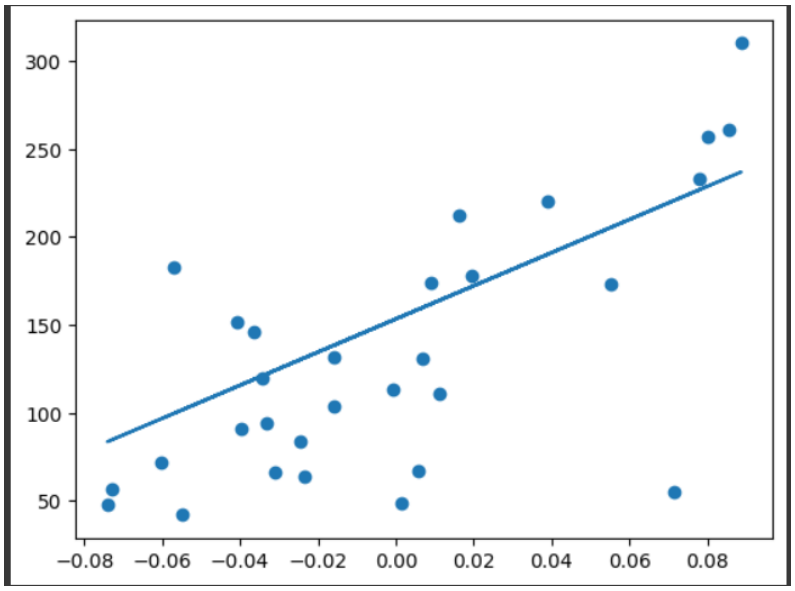
Part A: Program Single variate using inbuilt functions.

Code:

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error

diabetes = datasets.load_diabetes()
diabetes_X = diabetes.data[:,np.newaxis,2]
diabetes_X_train = diabetes_X[:-30]
diabetes_X_test = diabetes_X[-30:]
diabetes_y_train = diabetes.target[:-30]
diabetes_y_test = diabetes.target[-30:]
model = linear_model.LinearRegression()
model.fit(diabetes_X_train, diabetes_y_train)
diabetes_y_predicted = model.predict(diabetes_X_test)
plt.scatter(diabetes_X_test, diabetes_y_test)
plt.plot(diabetes_X_test, diabetes_y_predicted)
plt.show()
```

Output:



Subject: Data Mining and Warehousing **SAP ID: 60004220253 – Devansh Mehta**

Part B: Program Multi variate using inbuilt functions.

Code:

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)
diabetes_X = diabetes_X[:, np.newaxis, 2
diabetes_X_train = diabetes_X[:-20]
diabetes_X_test = diabetes_X[-20:]
diabetes_y_train = diabetes_y[:-20]
diabetes_y_test = diabetes_y[-20:]
regr = linear_model.LinearRegression()
regr.fit(diabetes_X_train, diabetes_y_train)
diabetes_y_pred = regr.predict(diabetes_X_test)
print("Coefficients: \n", regr.coef_)
print("Mean squared error: %.2f" % mean_squared_error(diabetes_y_test, diabetes_y_pred))
print("Coefficient of determination: %.2f" % r2_score(diabetes_y_test, diabetes_y_pred))
```

Output:

```
Coefficients:
 [938.23786125]
Mean squared error: 2548.07
Coefficient of determination: 0.47
```