**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)

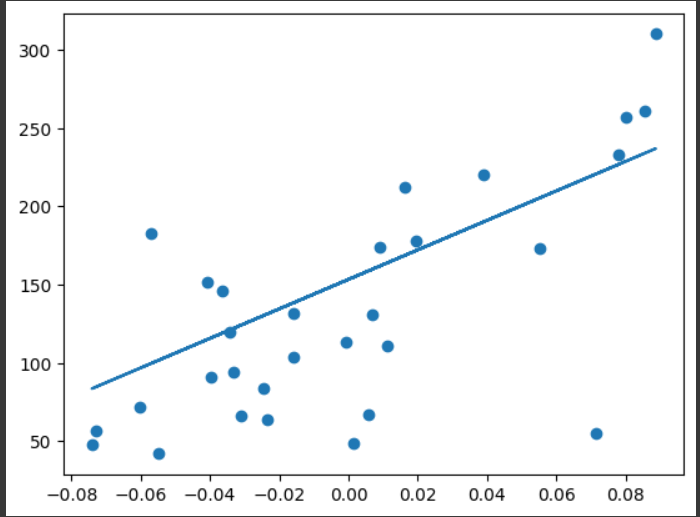
diabetes\_y\_predicted = model.predict(diabetes\_X\_test)

plt.scatter(diabetes\_X\_test, diabetes\_y\_test)

plt.plot(diabetes\_X\_test, diabetes\_y\_predicted)

plt.show()

**Output:**

****

**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()

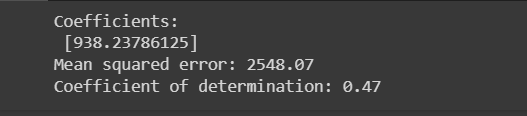
regr.fit(diabetes\_X\_train, diabetes\_y\_train)

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:**