Practical No.:-01

**Aim:-**Implement Linear Regression (Diabetes Dataset).

# Theory

Linear regression is just the process of estimating an unknown quantity based on some known ones (this is the regression part) with the condition that the unknown quantity can be obtained from the known ones by using only 2 operations: scalar multiplication and addition (this is the linear part).

# Material

* matplotlib.pyplot
* numpy
* seaborn

# Program

# Import Dependencies import numpy as np

import matplotlib.pyplot as plt

from sklearn import datasets,linear\_model,metrics from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_squared\_error, r2\_score import seaborn as sns

# Load the diabetes dataset

diabetes=datasets.load\_diabetes()

# X - feature vectors # y - Target values X=diabetes.data y=diabetes.target

# splitting X and y into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.4, random\_state=1)

el.LinearRegre# Create linear regression objest lin\_reg=linear\_modssion()

# Train the model using trai and test data lin\_reg.fit(X\_train,y\_train)

# Predict values for X\_test data predicted = lin\_reg.predict(X\_test)

# Regression coefficients

print('\n Coefficients are:\n',lin\_reg.coef\_)

# Intecept

print('\nIntercept : ',lin\_reg.intercept\_)

# variance score: 1 means perfect prediction print('Variance score: ',lin\_reg.score(X\_test, y\_test))

# Mean Squared Erroe print("Mean squared error: %.2f\n"

% mean\_squared\_error(y\_test, predicted)) # Original data of X\_test

expected = y\_test

# Plot a graph for expected and predicted values plt.title('Linear Regression ( DIABETS Dataset)') plt.scatter(expected,predicted,c='b',marker='.',s=36)

plt.plot(np.linspace(0, 330, 100),np.linspace(0, 330, 100), '--r', linewidth=2) plt.show()

# Output





