**AML Algorithm #3: Predict diabetes using Simple Linear Regression**

# Python code on sklearn linear regression example

# Importing required libraries

import numpy as np

import matplotlib.pyplot as plt

from sklearn.datasets import load\_diabetes

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

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

# Loading the sklearn diabetes dataset

X, Y = load\_diabetes(return\_X\_y=True)

# Taking only one feature to perform simple linear regression

X = X[:,8].reshape(-1,1)

# Splitting the dependent and independent features of the dataset into training and testing dataset

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split( X, Y, test\_size = 0.4, random\_state = 10 )

# Creating an instance for the linear regression model of sklearn

lr = LinearRegression()

# Training the model by passing the dependent and independent features of the training dataset

lr.fit( X\_train, Y\_train )

# Creating an array of predictions made by the model for the unseen or test dataset

Y\_pred = lr.predict( X\_test )

# The value of the coefficients for the independent feature through the multiple regression model

print("Value of the oefficients: \n", lr.coef\_)

# The value of the mean squared error

print(f"Mean square error: {mean\_squared\_error( Y\_test, Y\_pred)}")

# The value of the coefficient of determination, i.e., R-square score of the model

print(f"Coefficient of determination: {r2\_score( Y\_test, Y\_pred )}")

# Plotting the output

plt.scatter(X\_test, Y\_test, color = "black", label = "original data")

plt.plot(X\_test, Y\_pred, color = "blue", linewidth=3, label = "regression line")

plt.xlabel("Independent Feature")

plt.ylabel("Target Values")

plt.title("Simple Linear Regression(diabetes)")

plt.show()

**OUTPUT:**



