# PROGRAM 4: Write a Python program to implement Simple Linear Regression

# **Simple Linear Regression in Python**

Simple Linear Regression is a linear regression model that uses a single feature to make predictions. In this example, we will implement a Simple Linear Regression model using Python and the scikit-learn library.

Theory:

Simple Linear Regression is a linear model that predicts a continuous output variable based on a single input feature. The model assumes a linear relationship between the input feature and the output variable.

The equation for Simple Linear Regression is:

y = β0 + β1x + ε

where:

* y is the output variable
* x is the input feature
* β0 is the intercept or bias term
* β1 is the slope coefficient
* ε is the error term

# Code:

# import numpy as np

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

# from sklearn.linear\_model import LinearRegression

# from sklearn import metrics

# import matplotlib.pyplot as plt

# np.random.seed(0)

# X = np.random.rand(100, 1)

# y = 3 + 2 \* X + np.random.randn(100, 1)

# X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# model = LinearRegression()

# model.fit(X\_train, y\_train)

# y\_pred = model.predict(X\_test)

# print("Intercept: ", model.intercept\_)

# print("Slope: ", model.coef\_)

# print("Mean Absolute Error: ", metrics.mean\_absolute\_error(y\_test, y\_pred))

# print("Mean Squared Error: ", metrics.mean\_squared\_error(y\_test, y\_pred))

# print("Root Mean Squared Error: ", np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

# plt.scatter(X\_test, y\_test, label="Data")

# plt.plot(X\_test, y\_pred, color="red", label="Regression Line")

# plt.legend()

# plt.show()

# Output:

Intercept: [3.20634019]

Slope: [[1.9805182]]

Mean Absolute Error: 0.8014554834253331

Mean Squared Error: 0.9177532469714291

Root Mean Squared Error: 0.9579943877557056

# linear