TASK-2

IMPLEMENT A SIMPLE LINEAR

REGRESSION MODEL

## DATA PREPARATION:

In this task the dataset contains a predicator variable(X) and a numerical target variable (y).By using this dataset we can understand the structure and check for any missing value .

We can load this dataset as x and y variables by using numpy module

## DATA SPLITTING:

Here we split the dataset into two sets,one is training set and another is testing set.This splitting an be done in 8:2 ratio.This is done to train a model by using dataset .

By using the libraries scikit-learn we can do splitting.

## MODEL CREATION:

In this we create a linear regression model.Linear regression is used to find the relationship between predicator variable(X) and target variable(y).

## MODEL TRAINING:

We can train this model using the training dataset. Here we are trying to fit the given training data to the model.

## PREDICTION:

In this we are training the model to make the predictions on testing data .Here we predict the targeted variable based on the predictor variable.

## MODEL EVALUATION:

Here we calculate the evaluation metrics to know the model performance .Here we use the metrics like Mean Squared Error(MSE),Root Mean Squared Error(RMSE),and R-squared(R^2).

## VISUALIZATION:

In this we create a plot with the regression line to understand and visualize the how the model fit in the given data.In this we can understand the relationship between the variables.

## CODE:

import numpy as np

import matplotlib.pyplot as plt # Import necessary libraries

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

from sklearn.linear\_model import LinearRegression

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

np.random.seed(0)

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

y = 2 \* X + 1 + 0.1 \* np.random.randn(100, 1) # Target variable

# Split the data into training and testing sets

A1, B1, A2, B2 = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Create a Linear Regressionmodel

model = LinearRegression()

# Train the model on the training data

model.fit(A1, B1)

# Make predictions on the testing data

Y1= model.predict(A2)

# Calculate Mean Squared Error (MSE) and R-squared (R^2) for evaluation

mse = mean\_squared\_error(B2, Y1)

r2 = r2\_score(B2, Y2)

# Print model coefficients and evaluation metrics

print("Coefficients:", model.coef\_)

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

print("Mean Squared Error (MSE):", mse)

print("R-squared (R^2):", r2)

# Visualization

plt.scatter(A2, B2, color='blue', label='Actual')

plt.plot(B2, Y1, color='red', linewidth=2, label='Predicted')

plt.legend()

plt.title('Linear Regression: Actual vs. Predicted')

plt.xlabel('Predictor Variable')

plt.ylabel('Target Variable')

plt.show()

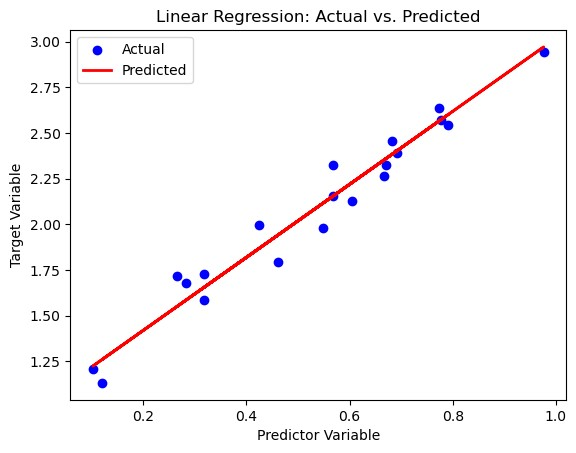
## RESULT:

Coefficients: [[1.99805182]]

Intercept: [1.02063402]

Mean Squared Error (MSE): 0.009177532469714292

R-squared (R^2): 0.9576884341540605



## RESOURCE:

1.Jupyter notebook

2.Module:

1.numpy(dataset)

2.matplotlib

3.sklearn(scikit-learn)

## 