**Practical No 6 : Regression Analysis: Fit a linear regression model to a dataset and interpret the coefficients.**

Fit a linear regression model to a dataset and interpret the coefficients.

**Steps to Implement:**

**Step 1: Install Required Libraries**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.linear\_model import LinearRegression

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

**Step 2: Load Dataset**

data = pd.read\_csv("data.csv") # Replace with actual dataset path

data.head()

**Step 3: Define Variables**

X = data[["Feature\_Column"]] # Independent variable

y = data["Target\_Column"] # Dependent variable

**Step 4: Split Data**

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

**Step 5: Fit Linear Regression Model**

model = LinearRegression()

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

**Step 6: Interpret Coefficients**

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

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

**Step 7: Make Predictions**

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

**Step 8: Evaluate Model**

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print("Mean Squared Error:", mse)

print("R-squared Score:", r2)

**Step 9: Plot Results**

plt.scatter(X\_test, y\_test, color='blue', label='Actual')

plt.plot(X\_test, y\_pred, color='red', linewidth=2, label='Predicted')

plt.xlabel("Feature")

plt.ylabel("Target")

plt.legend()

plt.show()

**Conclusion:**

* **Intercept:** The value of the dependent variable when the independent variable is 0.
* **Coefficient (Slope):** Shows the effect of the independent variable on the dependent variable.
* **Evaluation Metrics:** MSE indicates error, and R² score measures model performance.
* **Visualization:** Shows the fitted regression line against actual data points.