**NAME : MANGESH A. GHADWAJE**

**ROLL NO:24**

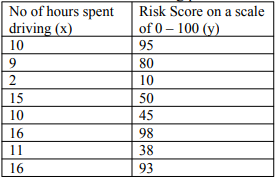
**BATCH : B2**

**COURSE: ML PRACTICAL**

**Assginment No. 3**

**Problem Statement :**

**Implement Linear Regression: The following table shows the results of a recently conducted study on the correlation of the number of hours spent driving with the risk of developing acute backache. Find the equation of the best fit line for this data. Predict risk score for 20 hours using predictor model.**



**Code :**

***import pandas as pd***

***import numpy as np***

***import matplotlib.pyplot as plt***

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

***from sklearn.linear\_model import LinearRegression***

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

***# Step 1: Read the dataset***

***data = pd.read\_csv('driving\_dataset.csv')***

***# Print the original dataset***

***print("Original Dataset:")***

***print(data)***

***# Step 2: Extract independent and dependent variables***

***x = data.iloc[:, 0].values.reshape(-1, 1) # Independent variable ('hours')***

***y = data.iloc[:, 1].values # Dependent variable ('risk\_score')***

***# Print extracted variables***

***print("\nIndependent variable (x):")***

***print(x)***

***print("\nDependent variable (y):")***

***print(y)***

***# Step 3: Train-Test Split***

***x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.3, random\_state=42)***

***# Print training and test data***

***print("\nTraining Data:")***

***print(pd.DataFrame({'Hours': x\_train.flatten(), 'Risk Score': y\_train}))***

***print("\nTest Data:")***

***print(pd.DataFrame({'Hours': x\_test.flatten(), 'Risk Score': y\_test}))***

***# Step 4: Create and fit the linear regression model***

***regressor = LinearRegression()***

***regressor.fit(x\_train, y\_train)***

***# Step 5: Predict on the test set***

***y\_pred = regressor.predict(x\_test)***

***# Print regression coefficients***

***print(f"\nRegression Coefficient (Slope): {regressor.coef\_[0]:.3f}")***

***print(f"Intercept: {regressor.intercept\_:.3f}")***

***# Print predictions***

***print("\nPredictions on Test Data:")***

***predictions\_df = pd.DataFrame({'Hours': x\_test.flatten(), 'Actual Risk Score': y\_test, 'Predicted Risk Score': y\_pred})***

***print(predictions\_df)***

***# Step 7: Visualizing the Training and Test Set Results***

***plt.figure(figsize=(12, 6))***

***# Scatter plot of the training set results***

***plt.scatter(x\_train, y\_train, color='purple', label='Training data')***

***# Plot the regression line for the training set***

***plt.plot(x\_train, regressor.predict(x\_train), color='cyan', label='Regression Line (Training Set)')***

***# Scatter plot of the test set results***

***plt.scatter(x\_test, y\_test, color='orange', label='Test data')***

***# Plot the regression line for the test set***

***plt.plot(x\_test, y\_pred, color='magenta', linestyle='--', label='Regression Line (Test Set)')***

***plt.title('Training and Test Set Results')***

***plt.xlabel('Hours Spent Driving')***

***plt.ylabel('Risk Score')***

***plt.legend()***

***plt.grid(True)***

***plt.show()***

**Output :**

