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 import time

# Load the data from the CSV file file\_path = ' / content/ test . '

try :

data = pd . read\_csv(file\_path) print("Data Loaded Successfully: " ) print(data) except FileNotFoundError:

print(f"Error: The file {file\_path} was not found. ") exit ( )

 Prepare the data

X = 'Vehicle\_Count' ]] # Features y = data[ ' Light\_Duration ' ] # Target variable

 Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_sp1it(X, y, test\_size=e.2, random\_state=42)

 Create a linear regression model model = LinearRegression()

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

# Make predictions predictions = model . predict (X\_test)

# Evaluate the model mse = mean\_squared\_error(y\_test, predictions) r2\_score(y\_test, predictions)

print("\nMode1 Evaluation: ") print(f"Mean Squared Error: {mse: .2f}")

# Score: {r2: .2f}")

# Visualize the results plt.scatter(X\_test, y\_test, color= 'blue' , label='Actua1 Data' ) plt. scatter (X\_test, predictions, color= ' red' , label= ' Predictions plt. plot (X\_test, predictions, color= ' green ' linewidth=2, label= 'Regression Line' ) plt.title( 'Vehicle Count vs Light Duration' ) plt.xlabel( 'Vehicle Count' ) plt.ylabel( ' Light Duration' ) plt. legend() PIt. show()

# Function to predict light duration based on vehicle count def predict\_light\_duration (vehicle\_count) : if vehicle count < e:

raise ValueError("Vehic1e count cannot be negative. return model.  [vehicle\_count] ] ) ) [ e ]

# Function to simulate traffic light control def traffic\_light\_control (vehicle\_count) :

predicted\_duration = predict\_light\_duration (vehicle\_count) print(f"\nPredicted Light Duration for {vehicle\_count} vehicles: {predicted\_duration: .2f} seconds")

# Simulate the traffic light timing print("Traffic Light will be GREEN for { : .2f} seconds. " . format(predicted\_duration)) time. sleep(predicted\_duration) # Simulate the duration of the green light print("Traffic Light will turn RED now. ")

# Example usage try :

vehicle count\_input - 30 # Example vehicle count traffic\_light\_control (vehicle\_count\_input) except ValueError as e: print(f"Error:

Data Loaded Successfully:

Vehicle\_Count Light\_Duration

 e  le

25 3e

30 35

35 40

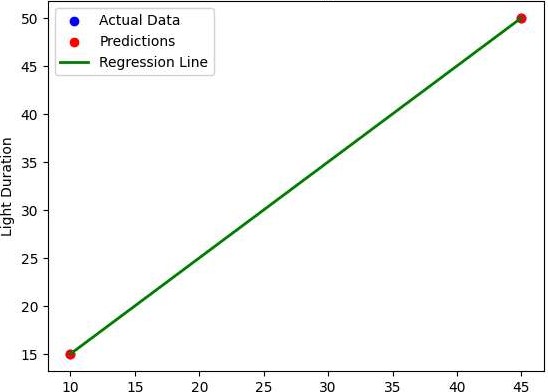
40

50 55

Model Evaluation:

mean Squared Error: e.ee RA2 Score: I.ee

Vehicle Count vs Light Duration



10

15

20

25

30

35

Vehicle Count

/usr/local/lib/ python3.11 'dist-packages/sklearn/utils/val idation. py: 2739 : userWarning: X does not have valid feature names, but LinearRE warnings. warn (

Predicted Light Duration for 3B vehicles: 35 .ee seconds Traffic Light will be GREEN for 35 .ee seconds. Traffic Light will turn RED now.