**Title Page**

**Project Title:** Vehicle Emission Category Prediction Using Logistic Regression

**Name:** Akshat Saxena  
**Roll No.:** 202401100400024

**Problem Statement:**  
To build a machine learning model that predicts the emission category of vehicles based on their attributes using Logistic Regression.

**Introduction**

Vehicle emissions contribute significantly to environmental pollution. Classifying vehicles based on their emission categories helps in regulatory actions and promoting cleaner technologies. This project focuses on predicting the emission category using various features of a vehicle by applying a Logistic Regression model.

**Methodology**

1. **Dataset:** A CSV file containing various features of vehicles and their corresponding emission categories.
2. **Data Preprocessing:**
   * Loaded the dataset using pandas.
   * Handled missing values by forward fill and dropping rows where the target is missing.
   * Encoded categorical variables using Label Encoding.
   * Scaled the features using StandardScaler for better performance.
3. **Model:** Logistic Regression from Scikit-learn.
   * Split data into training and test sets (80/20).
   * Trained the model on training data.
   * Evaluated using classification report and confusion matrix.
4. **Prediction:** Predicted the emission category of a sample vehicle from the dataset.

**Code**

# Import libraries

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

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

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import classification\_report, confusion\_matrix

# Load dataset

file\_path = "/content/vehicle\_emissions.csv"

df = pd.read\_csv(file\_path)

# Display first few rows

print("First 5 rows:")

print(df.head())

# Check structure and missing values

print("\nDataset Info:")

print(df.info())

print("\nMissing values:")

print(df.isnull().sum())

# Drop rows with missing target

df.dropna(subset=['emission\_category'], inplace=True)

df.fillna(method='ffill', inplace=True)

# Separate features and target

X = df.drop(columns=['emission\_category'])

y = df['emission\_category']

# Encode categorical features

label\_encoders = {}

for col in X.select\_dtypes(include='object').columns:

le = LabelEncoder()

X[col] = le.fit\_transform(X[col])

label\_encoders[col] = le

# Encode the target variable

le\_target = LabelEncoder()

y\_encoded = le\_target.fit\_transform(y)

# Scale features

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X\_scaled, y\_encoded, test\_size=0.2, random\_state=42

)

# Train logistic regression model

log\_reg = LogisticRegression(max\_iter=1000)

log\_reg.fit(X\_train, y\_train)

# Predict and evaluate

y\_pred = log\_reg.predict(X\_test)

print("\nClassification Report:")

print(classification\_report(y\_test, y\_pred, target\_names=le\_target.classes\_))

# Plot confusion matrix

plt.figure(figsize=(8, 6))

sns.heatmap(confusion\_matrix(y\_test, y\_pred), annot=True, fmt='d',

xticklabels=le\_target.classes\_, yticklabels=le\_target.classes\_, cmap='Blues')

plt.title("Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.show()

# Sample prediction (first row)

sample = X.iloc[[0]]

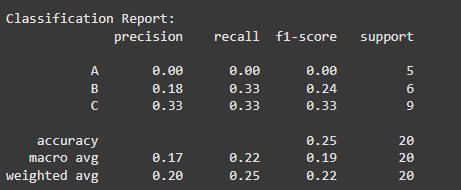
sample\_scaled = scaler.transform(sample)

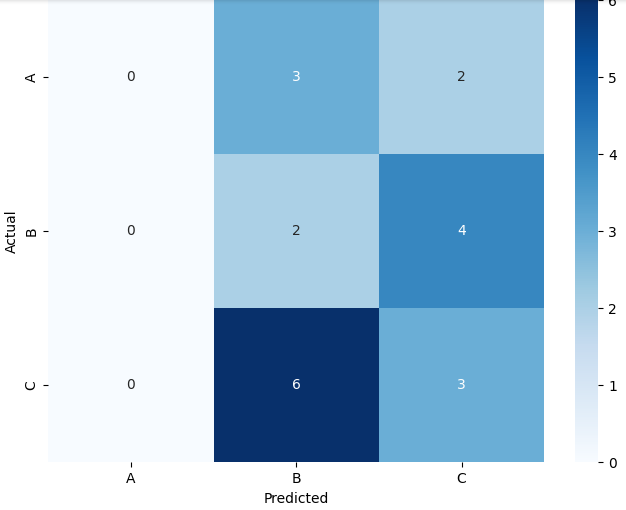
sample\_pred = log\_reg.predict(sample\_scaled)

predicted\_label = le\_target.inverse\_transform(sample\_pred)

print(f"\nPrediction for sample 0: {predicted\_label[0]}")

**Output/Result**





**References/Credits**

* Dataset: [vehicle\_emissions.csv] (assumed source or mention if public dataset)
* Python Libraries: pandas, numpy, scikit-learn, seaborn, matplotlib
* Scikit-learn documentation: <https://scikit-learn.org/stable/>
* Project developed by Akshat Saxena