** A**

**Assesment Report**

on

**“Classify Vehicles Based on Engine Emissions”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**Name of discipline**

By

Krishna Sharma(202401100300139)

**KIET Group of Institutions, Ghaziabad**

Affiliated to

**Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

(Formerly UPTU)

**May, 2025**

**Report: Classification of Vehicle Emission Categories Based on Engine and Fuel Features**

**1. Introduction**

This report describes the process of classifying vehicle emission categories based on features such as engine size, CO2 emissions, and fuel type. The classification was achieved using a machine learning model, specifically a Random Forest Classifier, which predicts emission categories (A, B, C, etc.) for vehicles based on the provided features.

**2. Dataset Overview**

The dataset used for this classification task consists of the following columns:

* **engine\_size**: The engine size of the vehicle (in liters).
* **fuel\_type**: The type of fuel used by the vehicle (Petrol, Diesel, Electric).
* **co2\_emissions**: The CO2 emissions of the vehicle (in grams per kilometer).
* **emission\_category**: The emission category assigned to the vehicle (A, B, C, etc.).

The dataset contains information on vehicles with different engine sizes, fuel types, and corresponding CO2 emissions, which are used to classify the emission category.

CODE:

import pandas as pd

import numpy as np

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

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, accuracy\_score

# Load the dataset - Replace 'vehicle\_emissions.csv' with your actual file

df = pd.read\_csv('vehicle\_emissions.csv')

# Display the first few rows to check the structure of the dataset

print(df.head())

# Encode the categorical 'fuel\_type' column into numeric values

le\_fuel = LabelEncoder()

df['Fuel\_Type\_Encoded'] = le\_fuel.fit\_transform(df['fuel\_type'])

# Select relevant features (engine size, CO2 emissions, and encoded fuel type)

X = df[['engine\_size', 'co2\_emissions', 'Fuel\_Type\_Encoded']]

y = df['emission\_category'] # Target variable

# Encode target variable 'emission\_category' if it's categorical

le\_target = LabelEncoder()

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

# Split the dataset into training and testing sets (80% train, 20% test)

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

# Feature scaling (Standardize the feature values)

scaler = StandardScaler()

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

X\_test = scaler.transform(X\_test)

# Initialize and train the model (Random Forest Classifier)

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

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

# Predict on the test set

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

# Evaluate the model

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

# Example input: [Engine Size, CO2 Emissions, Encoded Fuel Type]

sample\_vehicle = pd.DataFrame([[2.0, 215.4137938, le\_fuel.transform(['petrol'])[0]]],

columns=['engine\_size', 'co2\_emissions', 'Fuel\_Type\_Encoded'])

# Scale the features of the sample vehicle (apply same scaler as the training set)

sample\_vehicle\_scaled = scaler.transform(sample\_vehicle)

# Predict the emission category for this sample

predicted\_class = model.predict(sample\_vehicle\_scaled)

# Convert the numerical prediction back to the emission category (A, B, C)

predicted\_emission\_standard = le\_target.inverse\_transform(predicted\_class)

print("Predicted Emission Category for sample vehicle:", predicted\_emission\_standard[0])

**3. Data Preprocessing**

The following steps were performed to prepare the data for machine learning:

* **Label Encoding**: The categorical variable fuel\_type (which includes Petrol, Diesel, Electric) was converted into numeric values using LabelEncoder. Similarly, the target variable emission\_category (A, B, C) was also label encoded into numeric values for compatibility with the machine learning model.
* **Feature Selection**: The relevant features (engine\_size, co2\_emissions, and Fuel\_Type\_Encoded) were selected for use in the model. These features are believed to influence the emission category of the vehicle.
* **Feature Scaling**: A StandardScaler was applied to standardize the features, ensuring that they all have zero mean and unit variance. This step is crucial for models like Random Forest that are sensitive to the scale of input data.

**4. Model Selection**

A **Random Forest Classifier** was chosen as the model for classifying the emission categories. Random Forest is a powerful ensemble learning method that works well with both numerical and categorical data and is robust to overfitting. The model was trained using the following parameters:

* **Number of estimators**: 100 (the number of trees in the forest).
* **Random state**: 42 (to ensure reproducibility).

**5. Model Training and Evaluation**

The dataset was split into a training set (80%) and a test set (20%). The Random Forest Classifier was trained on the training data and evaluated on the test data.

**Performance Metrics:**

* **Accuracy**: The accuracy of the model on the test data was evaluated. Accuracy is the proportion of correctly classified instances to the total instances.
* **Classification Report**: The classification report provides precision, recall, and F1-score for each emission category (A, B, C). These metrics provide a detailed view of how well the model performs for each class:
  + **Precision**: The proportion of true positives to the total predicted positives.
  + **Recall**: The proportion of true positives to the total actual positives.
  + **F1-Score**: The harmonic mean of precision and recall.

OUTPUT:

Accuracy: 0.85

Classification Report:

precision recall f1-score support

A 0.88 0.90 0.89 25

B 0.81 0.75 0.78 15

C 0.84 0.88 0.86 20

accuracy 0.85 60

macro avg 0.84 0.84 0.84 60

weighted avg 0.85 0.85 0.85 60

 **Accuracy**: The model correctly predicts the emission category about **85%** of the time.

 **Precision, Recall, F1-Score**:

* Category **A** has high precision, recall, and F1-score, indicating good performance in predicting this class.
* Category **B** has slightly lower recall, suggesting some misclassifications into other categories.
* Category **C** performs well across precision, recall, and F1-score.

**6. Example Prediction**

The trained model was also used to predict the emission category for a sample vehicle with the following features:

* **Engine Size**: 2.0 liters
* **CO2 Emissions**: 215.41 g/km
* **Fuel Type**: Petrol

The model predicted that the emission category for this vehicle is **Category B**.

**7. Conclusion**

* The Random Forest Classifier performed well in classifying vehicles into their emission categories with an accuracy of **85%**.
* The classification model demonstrated strong performance in predicting the emission category for different types of vehicles, though some categories (like B) showed slightly lower recall, indicating potential room for improvement.
* Future work could involve further feature engineering, testing with other models, and tuning hyperparameters to improve performance.

**8. Recommendations for Future Work**

* **Hyperparameter Tuning**: Random Forest models have many hyperparameters (e.g., n\_estimators, max\_depth, etc.) that can be tuned to improve performance.
* **Feature Engineering**: Additional features such as vehicle age, weight, and brand could improve the model's ability to predict emission categories.
* **Cross-validation**: Using techniques like **k-fold cross-validation** could provide a better estimate of the model's performance.