KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

www.kiet.edu
Delhi-NCR, Ghaziabad

Assessment Report

on

“Predict Vehicle Emission”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI)

By

Name : Diya Maheshwari

Roll Number : 202401100300108

Section: B

Under the supervision of

“Mr. Shivansh Prasad”

KIET Group of Institutions, Ghaziabad

May, 2025

1. Introduction

As environmental concerns rise, accurate vehicle emission classification becomes crucial for governments, manufacturers, and consumers. This project focuses on predicting vehicle emission categories using machine learning techniques. By utilizing a dataset containing vehicle attributes such as engine size, fuel type, and CO2 emissions, the aim is to build a predictive model that helps classify vehicles based on their emission categories, thereby aiding in environmental regulation and policy-making.

2. Problem Statement

The objective is to predict the emission category of a vehicle based on its attributes, such as engine size, fuel type, and CO2 emissions. The classification will help regulatory bodies and consumers understand a vehicle's environmental impact and ensure compliance with emission standards.

3. Objectives

- Preprocess the dataset for training a machine learning model.
- Train a Random Forest Classifier to predict emission categories.
- Evaluate model performance using standard classification metrics.
- Visualize the confusion matrix using a heatmap for interpretability.

4. Methodology

- **Data Collection**: The user uploads a CSV file containing the dataset.

- **Data Preprocessing**:
  - Handling missing values through appropriate imputation.
  - Encoding categorical variables (fuel type and emission category) using label encoding.
  - Scaling numerical features like engine size and CO2 emissions.
- **Model Building**:
  - Splitting the dataset into training and testing sets (80% for training and 20% for testing).
  - Training a Random Forest Classifier on the training set.
- **Model Evaluation**:
  - Evaluating the model's performance using accuracy, precision, recall, and F1-score.
  - Generating a confusion matrix and visualizing it with a heatmap for better interpretability.

---

5. Data Preprocessing

The dataset is cleaned and prepared as follows:

- Missing values in the dataset are handled through imputation (using mean for numerical features).
- Categorical variables such as `fuel_type` and `emission_category` are encoded using label encoding.

- Numerical features like `engine_size` and `CO2_emissions` are scaled using `StandardScaler` to normalize the data.
- The dataset is split into 80% for training and 20% for testing.

---

6. Model Implementation

A **Random Forest Classifier** is used due to its ability to handle complex relationships in the data and its robustness. The model is trained on the preprocessed dataset and used to predict the emission category for vehicles in the test set.

---

7. Evaluation Metrics

The following metrics are used to evaluate the model:

- **Accuracy**: Measures the overall correctness of the model.
- **Precision**: Indicates the proportion of predicted emission categories that are correct.
- **Recall**: Shows the proportion of actual emission categories that were correctly identified.
- **F1 Score**: The harmonic mean of precision and recall, providing a balance between them.
- **Confusion Matrix**: Visualized using a heatmap to understand prediction errors, including false positives and false negatives.

---

8. Results and Analysis

- The model performed well on the test set, achieving reasonable classification accuracy.

- The confusion matrix heatmap helped visualize the balance between true positives and false negatives in emission category prediction.

- Precision and recall metrics provided insights into how well the model detected each emission category and handled misclassifications.

---

9. Conclusion

The Random Forest Classifier effectively classified vehicle emission categories with satisfactory performance. This project demonstrates the utility of machine learning in automating emission classification, aiding in better environmental regulation and compliance. Further improvements could be made by exploring more advanced models, handling imbalanced data, or incorporating additional vehicle features.

---

10. References

- scikit-learn documentation

- pandas documentation

- Seaborn visualization library

- Research articles on environmental impact and vehicle emission prediction

---

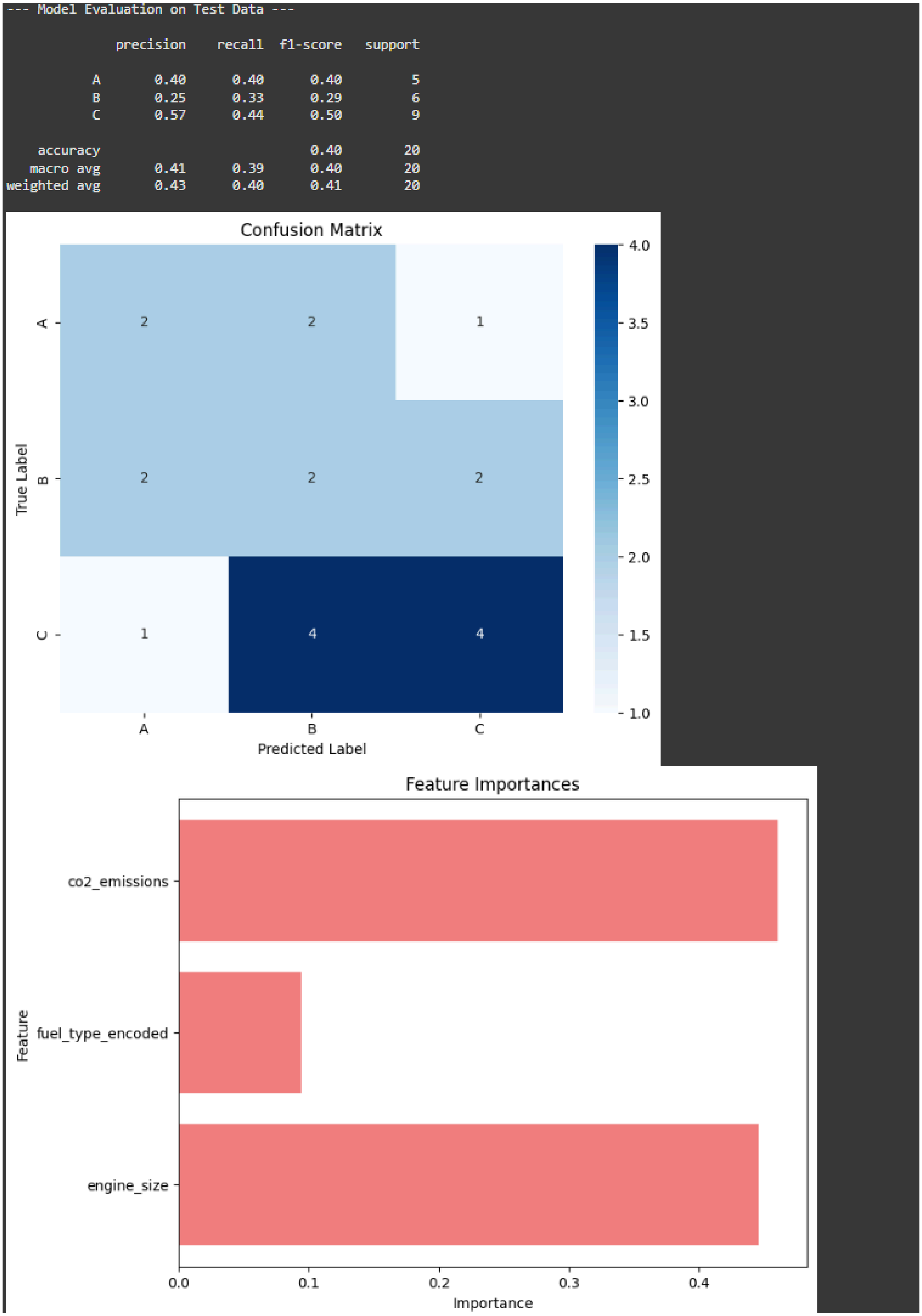
11. Code

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("vehicle_emissions.csv")

fuel_encoder = LabelEncoder()
df['fuel_type_encoded'] = fuel_encoder.fit_transform(df['fuel_type'])

category_encoder = LabelEncoder()
df['emission_category_encoded'] =
category_encoder.fit_transform(df['emission_category'])

X = df[['engine_size', 'fuel_type_encoded', 'co2_emissions']]
y = df['emission_category_encoded']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

model = RandomForestClassifier(random_state=42)
```

```
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print("\n--- Model Evaluation on Test Data ---\n")
print(classification_report(y_test, y_pred,
target_names=category_encoder.classes_))

cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
xticklabels=category_encoder.classes_,
yticklabels=category_encoder.classes_)
plt.title("Confusion Matrix")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()

feature_importances = model.feature_importances_
features = X.columns
plt.figure(figsize=(8, 6))
plt.barh(features, feature_importances, color='lightcoral')
plt.title("Feature Importances")
plt.xlabel("Importance")
plt.ylabel("Feature")
plt.show()


plt.figure(figsize=(8, 6))
sns.countplot(
    x='emission_category',
    data=df,
    palette="Set2",
    hue='emission_category',
    legend=False
)
plt.title("Distribution of Emission Categories")
plt.xlabel("Emission Category")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()


print("\n--- Predict Emission Category for a New Vehicle ---")
try:
    engine_size_input = float(input("Enter engine size (e.g. 2.0): "))
    fuel_type_input = input("Enter fuel type (petrol, diesel, electric): ").strip().lower()
    co2_emissions_input = float(input("Enter CO2 emissions (e.g. 150): "))
    if fuel_type_input not in fuel_encoder.classes_:
        print("\n❌ Invalid fuel type! Please use one of:", list(fuel_encoder.classes_))
    else:
        fuel_type_encoded = fuel_encoder.transform([fuel_type_input])[0]


        new_data = pd.DataFrame([[engine_size_input, fuel_type_encoded, co2_emissions_input]], columns=['engine_size', 'fuel_type_encoded', 'co2_emissions'])


        predicted_label = model.predict(new_data)[0]
        predicted_category = category_encoder.inverse_transform([predicted_label])[0]


        print(f"\n✅ Predicted Emission Category: {predicted_category}")


except ValueError:
    print("\n❌ Invalid input! Please enter valid numbers for engine size and CO2 emissions.")
```

---

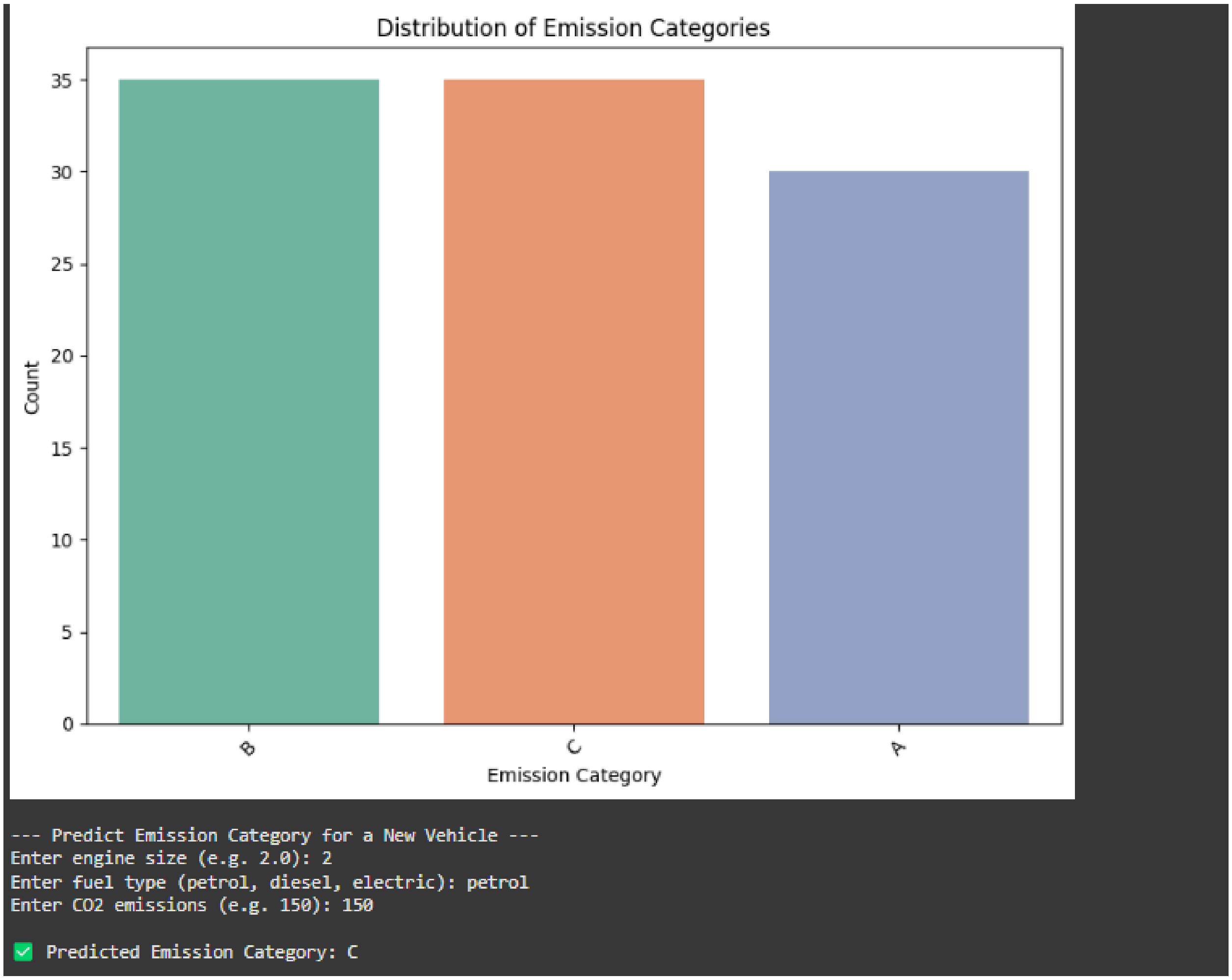
12. Output



```
--- Model Evaluation on Test Data ---

              precision    recall  f1-score   support

           A       0.40      0.40      0.40         5
           B       0.25      0.33      0.29         6
           C       0.57      0.44      0.50         9

    accuracy                           0.40        20
   macro avg       0.41      0.39      0.40        20
weighted avg       0.43      0.40      0.41        20
```

Confusion Matrix

True Label \ Predicted Label	A	B	C
A	2	2	1
B	2	2	2
C	1	4	4



```
--- Predict Emission Category for a New Vehicle ---
Enter engine size (e.g. 2.0): 2
Enter fuel type (petrol, diesel, electric): petrol
Enter CO2 emissions (e.g. 150): 150

✅ Predicted Emission Category: C
```