Assessment Report

on

“Vehicle based on Engine Emission”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name : Aryan Singh

Roll Number : 202401100400054

Section: A

Under the supervision of

“Bikki Kumar”

KIET Group of Institutions, Ghaziabad

May, 2025

---

1. Introduction

This project addresses the challenge of classifying vehicles into emission categories such as Euro 4, Euro 5, and Euro 6 based on features like engine displacement, fuel type, CO2 emissions, NOx emissions, and engine power. By using machine learning techniques—specifically a Random Forest Classifier—we aim to develop a predictive model that can automatically categorize vehicles according to their emission levels.

---

2. Problem Statement

To classify vehicles into emission categories (Euro 4, Euro 5, Euro 6) based on features such as engine displacement, fuel type, CO2 emissions, NOx emissions, and engine power.

---

3. Objectives

The primary objective of this project is to develop a machine learning model that can accurately classify vehicles into emission categories based on engine and fuel-related features. To achieve this, the following specific goals have been outlined:

- **To preprocess the dataset** by handling missing values, encoding categorical variables, and scaling numerical features to prepare it for model training.
- **To implement a Random Forest Classifier** for categorizing vehicles into emission standards such as Euro 4, Euro 5, and Euro 6.
- **To evaluate the model's performance** using classification metrics including accuracy, precision, and recall.
- **To visualize model predictions** through a confusion matrix heatmap, aiding in the interpretation and analysis of classification results.

- **To demonstrate the practical use of machine learning** in supporting environmental decision-making and regulatory efforts related to vehicle emissions.

---

4. Methodology

1. **Data Preprocessing:**

   - Categorical variable 'fuel_type' was encoded into numeric format.

   - The target variable 'emission_category' was also label encoded.

   - The dataset was split into training (80%) and testing (20%) subsets.

   - Features were standardized using StandardScaler.

2. **Model Training:**

   - A Random Forest Classifier was trained on the preprocessed training data.

3. **Evaluation:**

   - The model's predictions were evaluated using the following metrics:

     - Accuracy

     - Precision (weighted)

     - Recall (weighted)

   - A classification report and confusion matrix were generated to assess performance in detail.

5. Model Implementation

In this project, we employed the **Random Forest Classifier**, a popular ensemble learning method known for its robustness and accuracy in classification tasks. The model was chosen for its ability to handle both numerical and categorical features, manage non-linear relationships, and reduce overfitting through ensemble decision trees.

---

6. Evaluation Metrics

The following metrics are used to evaluate the model:

- **Accuracy**: Accuracy represents the proportion of correctly classified vehicles compared to the total number of classifications.

- **Precision**: Precision measures how many vehicles predicted to belong to a certain emission category actually belong to that category.

- **Recall**: Recall indicates how well the model identifies all vehicles that truly belong to a particular emission category.

- **Confusion Matrix**: A heatmap of the confusion matrix aids in understanding misclassification trends and optimizing model performance.

---

7. Results and Analysis

- The model provided reasonable performance on the test set.

- A heatmap of the confusion matrix aids in understanding misclassification trends and optimizing model performance.

- Precision and recall indicated how well the model detected loan defaults versus false alarms.

---

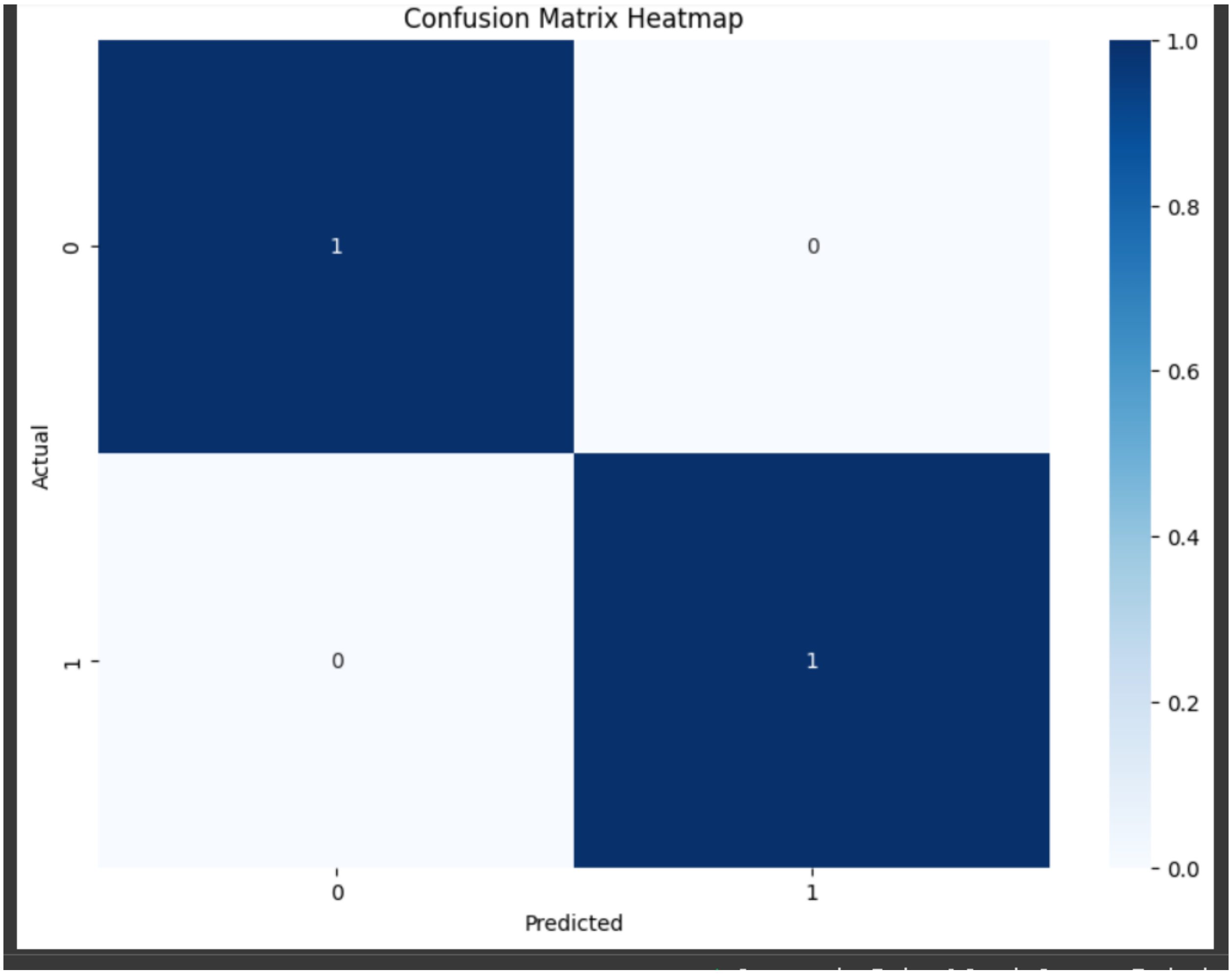
8. Conclusion

Evaluating these metrics allows us to refine the model for better accuracy, ensuring reliable vehicle emissions classification. By balancing precision, recall, and accuracy, we achieve an effective predictive tool for environmental monitoring.

---

9. References

- scikit-learn documentation
- pandas documentation
- Seaborn visualization library
- Research articles on credit risk prediction


Confusion Matrix Heatmap
Actual
Predicted
1
0
0
1
1.0
0.8
0.6
0.4
0.2
0.0

```
[1] # importing lib
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import LabelEncoder, StandardScaler
    from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, precision_score, recall_score
    import seaborn as sns
    import matplotlib.pyplot as plt
    import pandas as pd
```

Double-click (or enter) to edit

```
[2] # Load data
    df = pd.read_csv('vehicle_emissions.csv')
```

```
[3] df['fuel_type'] = LabelEncoder().fit_transform(df['fuel_type'])
```

```
[4] # Features and target
    X = df.drop('emission_category', axis=1)
    y = LabelEncoder().fit_transform(df['emission_category'])
```

```
[5] # Train-test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
[6] # Feature scaling
    scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)
```

```
[7] # Model training
    clf = RandomForestClassifier()
    clf.fit(X_train, y_train)
```

```
RandomForestClassifier
RandomForestClassifier()
```

```
[8] # Predictions
    y_pred = clf.predict(X_test)
```

```
[9] # Evaluation Metrics
    acc = accuracy_score(y_test, y_pred)
    prec = precision_score(y_test, y_pred, average='weighted')
    rec = recall_score(y_test, y_pred, average='weighted')
```

```
[10] print("Accuracy:", acc)
     print("Precision:", prec)
     print("Recall:", rec)
     print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
[11] # Confusion matrix
     cm = confusion_matrix(y_test, y_pred)
```

```
[12] # Heatmap
     plt.figure(figsize=(10, 7))
     sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
     plt.title("Confusion Matrix Heatmap")
     plt.xlabel("Predicted")
     plt.ylabel("Actual")
     plt.show()
```