Assessment Report

on

"Predict The Air Quality Level"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name : Akshat Gupta

Roll Number : 202401100400023

Section: A

Under the supervision of

"BIKKI KUMAR GUPTA"

KIET Group of Institutions, Ghaziabad

Problem Statement:
To classify air pollution levels (e.g., high, low) using environmental features such as PM2.5, NO2, and temperature.

Introduction

Air pollution has become a major concern worldwide due to its harmful effects on health and the environment. This project aims to develop a machine learning model that can predict the level of air quality based on certain environmental indicators such as PM2.5 (particulate matter), NO2 (Nitrogen Dioxide), and temperature. The predicted air quality levels will help in identifying pollution patterns and taking preventive measures. The task is a classification problem, where we classify the air quality as either high, low, or other possible categories.

Methodology

1. **Data Collection:** The dataset contains environmental readings including PM2.5, NO2, and temperature, along with the corresponding air quality level labels.
2. **Data Preprocessing:**
    - Selected relevant features (PM2.5, NO2, temperature).
    - Encoded the target variable (quality_level) using LabelEncoder.
    - Split the dataset into training and testing sets (80/20 split).

3. **Model Selection:** A Random Forest Classifier was used due to its robustness and ability to handle nonlinear relationships and feature importance.
4. **Model Training:** The model was trained on the training dataset.
5. **Model Evaluation:**
    - Predictions were made on the test data.
    - Evaluation metrics such as accuracy, precision, recall, and F1-score were calculated.
    - A confusion matrix was generated and visualized using a heatmap.

Code

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import (
  confusion_matrix, accuracy_score, precision_score,
  recall_score, f1_score
)
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Load dataset
df = pd.read_csv('/content/air_quality.csv')

# Features and Target
X = df[['pm25', 'no2', 'temperature']]
y = df['quality_level']

# Encode categorical labels
y_encoded = LabelEncoder().fit_transform(y)

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded,
test_size=0.2, random_state=42)

# Train the model
clf = RandomForestClassifier(random_state=42)
clf.fit(X_train, y_train)

# Predict
y_pred = clf.predict(X_test)
```

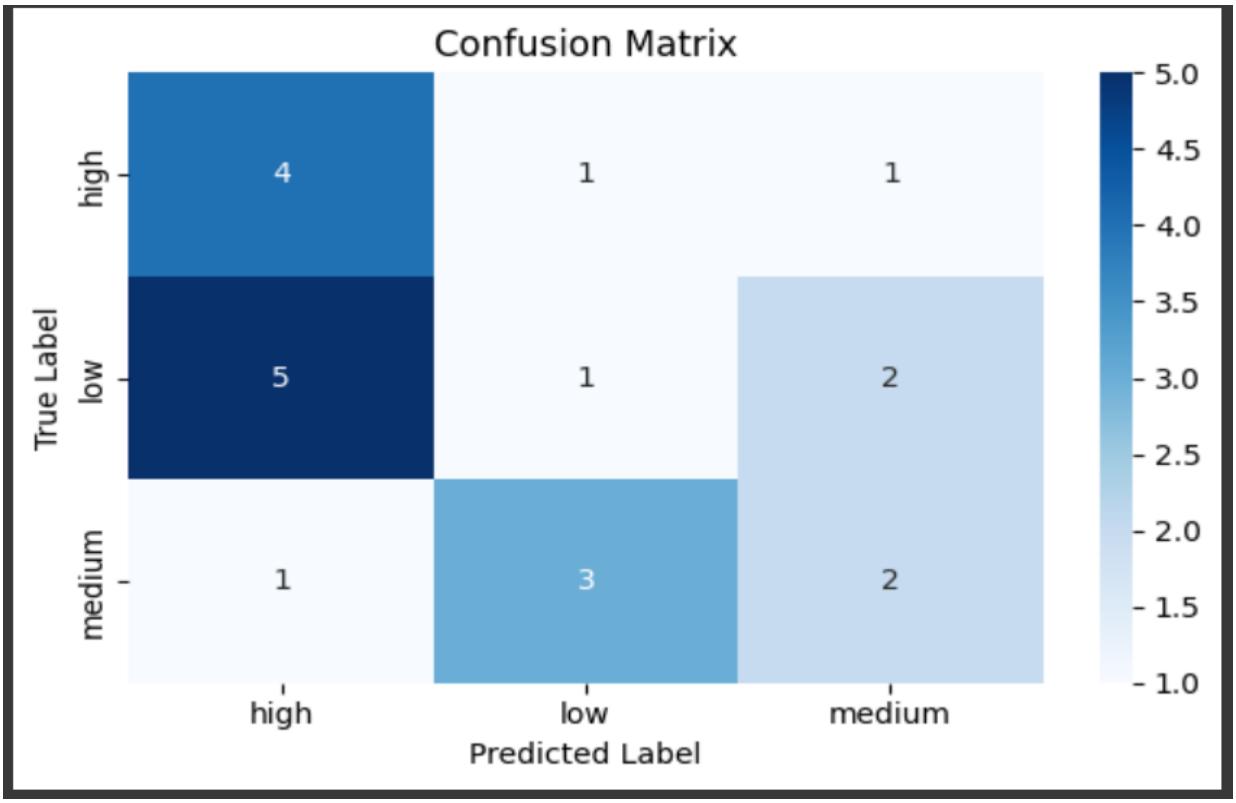
```
# Matrices
print("Accuracy :", accuracy_score(y_test, y_pred))
print("Precision:", precision_score(y_test, y_pred,
average='weighted'))
print("Recall   :", recall_score(y_test, y_pred, average='weighted'))
print("F1 Score :", f1_score(y_test, y_pred, average='weighted'))

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

Output/Result

- **Accuracy:** 0.35
- **Precision:** 0.32
- **Recall:** 0.35
- **F1 Score:** 0.32

Below is the confusion matrix heatmap representing the performance of the model on test data:



References/Credits

- Dataset source: Provided dataset file in CSV format.
- Libraries: pandas, scikit-learn, seaborn, matplotlib
- Confusion Matrix Explanation: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html