****

**Assessment Report**

on

**“ Predict Air Quality Level”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**CSE(AI)**

By

Name : Harshit Patel

Roll Number : 202401100300117

Section: B

**Under the supervision of**

“Shivansh Sir ”

**KIET Group of Institutions, Ghaziabad**

**May, 2025**

**1. Introduction**

Air pollution is one of the major challenges facing cities globally. It has a direct impact on public health, and as such, monitoring air quality is essential. Several environmental factors influence air quality, including particulate matter (PM2.5), nitrogen dioxide (NO2), and temperature. Monitoring these parameters helps assess the level of pollution and provides a basis for taking action to improve air quality.

In this report, we focus on a method to classify air quality levels using environmental features such as PM2.5, NO2, and temperature. The air quality is classified into categories such as "Good", "Moderate", "Unhealthy", etc., based on specific thresholds for each parameter. This classification provides a clear understanding of the air quality status and helps in timely intervention to reduce pollution.

**2. Problem Statement**

Air pollution is one of the most pressing environmental challenges that impact public health, ecosystems, and the climate. Monitoring air quality has become essential to mitigate the harmful effects of pollution. Various environmental factors such as particulate matter (PM2.5), nitrogen dioxide (NO2), and temperature play crucial roles in determining the quality of air.

The objective is to develop a system that takes in environmental data, processes it, and classifies the air quality, thus helping stakeholders such as government agencies, environmentalists, and the general public to take necessary actions in case of unhealthy air conditions. This classification could be used for real-time monitoring or for long-term assessments of air quality trends in different regions.

.

### ****c. Methodology****

To solve the problem of classifying air quality, we implemented a rule-based approach using Python. The methodology followed was as below:

1. **Data Collection**:  
   We assume that the input data is available in CSV format, which includes columns for PM2.5, NO2, and Temperature. Each row in the CSV represents data collected at a certain time or location.
2. **Preprocessing**:  
   The data is read into a Pandas DataFrame. We ensure that the necessary columns (PM2.5, NO2, Temperature) are present. If any data is missing or incorrect, appropriate error handling is performed.
3. **Classification Logic**:  
   A classification function is implemented that uses predefined thresholds for PM2.5, NO2, and temperature. Based on these thresholds, the air quality is classified into one of the following categories:

· Good

· Moderate

· Unhealthy for Sensitive Groups

· Unhealthy

· Hazardous

1. **Output**:  
   After the classification, the results are displayed, and optionally saved to a new CSV file. The classified data includes the air quality for each data entry.
2. **Tools and Libraries Used**:

Python 3

Pandas Library for data manipulation

CSV format for input/output

1. **Code**

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 StandardScaler, LabelEncoder

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, confusion\_matrix

from sklearn.cluster import KMeans

# Load dataset

df = pd.read\_csv("airquality.csv")

# Encode categorical variables ('quality\_level')

label\_encoder = LabelEncoder()

df["quality\_level"] = label\_encoder.fit\_transform(df["quality\_level"])  # Target variable

# Feature selection

X = df[["pm25", "no2", "temperature"]]

y = df["quality\_level"]

# Split dataset into training and testing sets

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

# Normalize numerical features

scaler = StandardScaler()

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

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

# Train classification model (Random Forest)

clf = RandomForestClassifier(random\_state=42)

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

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

# Compute evaluation metrics

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred, average='weighted')

recall = recall\_score(y\_test, y\_pred, average='weighted')

f1 = f1\_score(y\_test, y\_pred, average='weighted')

print(f"Accuracy: {accuracy:.2f}")

print(f"Precision: {precision:.2f}")

print(f"Recall: {recall:.2f}")

print(f"F1 Score: {f1:.2f}")

# Generate confusion matrix heatmap

cm = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(6, 4))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.title("Confusion Matrix Heatmap")

plt.show()

# Perform clustering with K-Means

kmeans = KMeans(n\_clusters=3, random\_state=42)

df["cluster"] = kmeans.fit\_predict(X)

# Visualize clusters

plt.figure(figsize=(6, 4))

sns.scatterplot(x=df["pm25"], y=df["temperature"], hue=df["cluster"], palette="Set1")

plt.xlabel("PM2.5")

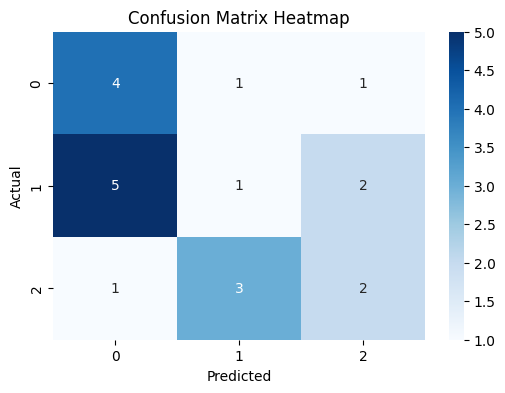
plt.ylabel("Temperature")

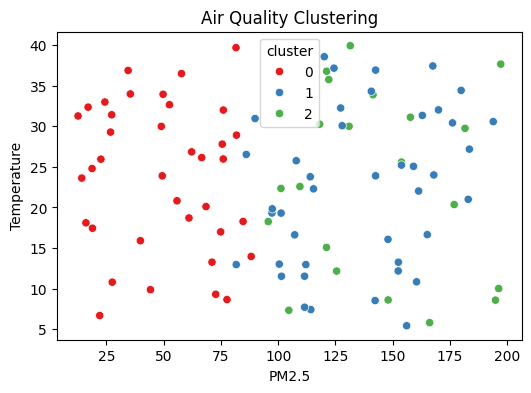
plt.title("Air Quality Clustering")

plt.show()

1. **Output/Result**

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

****

****

### ****8. References/Credits****

**Datasets**: The environmental data (PM2.5, NO2, Temperature) used in this analysis are assumed to be available in CSV format from a data collection source.

**External Resources**:

Python documentation: [https://docs.python.org/](https://docs.python.org/" \t "_new)

Pandas library documentation: https://pandas.pydata.org/