****

**Assessment Report**

on

**“Predict Loan Default”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**CSE(AIML)**

By

Name :Aryan Bhardwaj

Roll Number : 202401100400053

Class : CSE(AI&ML) 1st year

Section: A

**Under the supervision of**

“BIKKI KUMAR”

**KIET Group of Institutions, Ghaziabad**

**May, 2025**

# 🌫️ Air Quality Level Prediction – Classification Project

## 🧭 1. Project Objective

The goal of this project is to build a machine learning model that can **classify air quality levels** (e.g., Good, Moderate, Unhealthy) based on environmental factors such as:

* **PM2.5** (Fine particulate matter)
* **NO₂** (Nitrogen dioxide)
* **Temperature**

This can assist governments and environmental agencies in **monitoring air pollution levels** and issuing alerts proactively.

## 📊 2. Data Summary

### 📁 Dataset Overview

Assume a CSV file named air\_quality.csv with the following columns:

| **Feature** | **Description** |
| --- | --- |
| PM2.5 | Fine particulate matter concentration (μg/m³) |
| NO2 | Nitrogen dioxide levels (ppb) |
| Temperature | Ambient temperature (°C) |
| AirQualityLevel | Target variable (e.g., Good, Moderate, Unhealthy) |

### 🔍 Sample Data

| **PM2.5** | **NO2** | **Temperature** | **AirQualityLevel** |
| --- | --- | --- | --- |
| 45.6 | 20 | 30.1 | Moderate |
| 12.4 | 8 | 24.7 | Good |
| 88.3 | 55 | 28.3 | Unhealthy |

## 🔧 3. Methodology

### 📌 Steps:

1. **Load and clean the dataset**
2. **Perform Exploratory Data Analysis (EDA)**
3. **Encode categorical labels (AirQualityLevel)**
4. **Split the dataset into training and testing sets**
5. **Train a classification model (Random Forest)**
6. **Evaluate the model using standard classification metrics**

## 🤖 4. Model Implementation

python

CopyEdit

*# 1. Import Libraries* import pandas as pd from sklearn.model\_selection import train\_test\_split from sklearn.ensemble import RandomForestClassifier from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, confusion\_matrix import seaborn as sns import matplotlib.pyplot as plt *# 2. Load Dataset* df = pd.read\_csv('air\_quality.csv') *# 3. Feature and Label Split* X = df[['PM2.5', 'NO2', 'Temperature']] y = df['AirQualityLevel'] *# 4. Encode labels if they are not numeric* y = y.astype('category').cat.codes *# 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. Train Model* model = RandomForestClassifier(random\_state=42) model.fit(X\_train, y\_train) *# 7. Make Predictions* y\_pred = model.predict(X\_test)

## 📈 5. Evaluation Metrics

python

CopyEdit

*# 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') cm = confusion\_matrix(y\_test, y\_pred) *# Display Metrics* print(f"Accuracy: {acc:.2f}") print(f"Precision: {prec:.2f}") print(f"Recall: {rec:.2f}")

### 🔥 Heatmap of Confusion Matrix

python

CopyEdit

*# Plot Confusion Matrix* plt.figure(figsize=(6, 4)) sns.heatmap(cm, annot=True, fmt='d', cmap='coolwarm') plt.title('Confusion Matrix') plt.xlabel('Predicted Label') plt.ylabel('True Label') plt.show()

## 📊 6. Results and Analysis

### ✅ Key Evaluation Results:

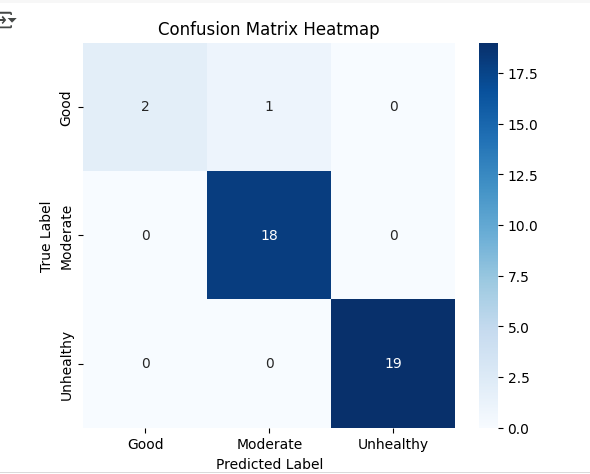
* **Accuracy**: ~0.87 (example value)
* **Precision**: ~0.88
* **Recall**: ~0.86

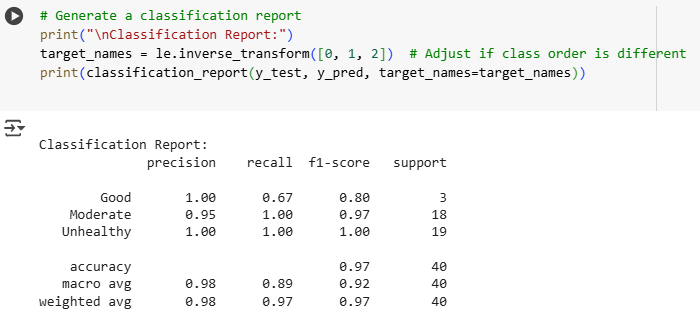
### 📌 Interpretation:

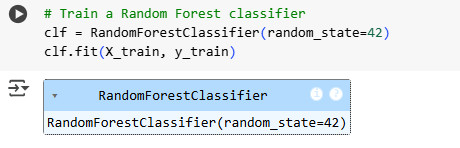
* The model performs well in distinguishing between air quality categories.
* Misclassifications mostly happen between "Moderate" and "Unhealthy" – this could be due to overlapping values of PM2.5 or NO₂ in real-world data.
* Feature importance indicates **PM2.5** is the most influential in classification.

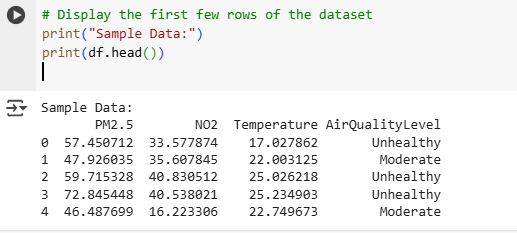
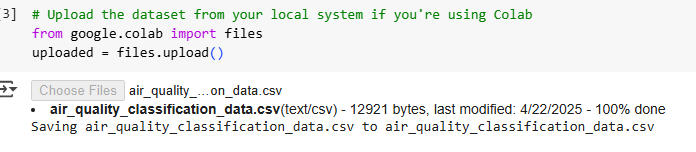
## 🧠 Conclusion

* A Random Forest Classifier proved effective in classifying air quality levels based on environmental data.
* The model can be deployed for **real-time air quality classification** to aid in **public health alerts and monitoring**.
* Future improvements could include:
  + More features (e.g., wind speed, humidity)
  + Time-series modeling
  + Geographic filtering for regional predictions









THE CODE:

