**Phase-2 Submission Template**

**Student Name:** D. NANDAGOPAL

**Register Number:** 412723205033

**Institution:** TAGORE ENGINEERING COLLEGE

**Department:** B. TECH INFORMATION TECHNOLOGY

**Date of Submission:** 07-05-2025

**GitHub Repository Link:** [Update the project source code to your Github Repository]

### **1. Problem Statement**

### Air pollution has become a significant concern in urban areas, directly impacting public health, quality of life, and environmental sustainability. Predicting the Air Quality Index (AQI) using sensor data can aid in proactive public health responses and policy decisions.

### Problem Type: Regression

### Refinement: Based on dataset analysis, the task was narrowed down to predicting AQI using key pollutant metrics such as CO, NMHC, C6H6, and NOx.

### Relevance: Real-time AQI prediction allows governments and individuals to take preventive actions to reduce exposure to harmful air.

### **2. Project Objectives**

* Build a machine learning model that accurately predicts AQI based on atmospheric pollutant sensor data.
* Develop a user-friendly GUI application for real-time predictions and insights.
* Provide health advice based on predicted AQI levels.
* Updated Goals are improving model interpretability and user interaction via visual tools and health impact categorization and Focus shifted from generic regression to real-time application usability after data exploration.

### **3. Flowchart of the Project Workflow**

### 

### 

### **4. Data Description**

* Dataset name and origin: AirQuality.csv, Kaggle
* Type of data: Time-series, structured sensor data.
* Number of records and features: ~9,358 instances. 15+ features, including pollutant concentration levels.
* Static dataset.
* Target variable: Computed AQI (custom weighted index based on pollutants)

### **5. Data Preprocessing**

### Missing Values: Imputed using Simple Imputer with mean strategy; -200 treated as missing.

### Duplicates: Not explicitly found in the dataset.

### Outliers: Handled indirectly by imputing invalid values (-200).

### Date & Time Parsing: Combined and parsed into Date Time column.

### Standardization: Applied Standard Scaler to normalize features before model training.

### Categorical Encoding: Not required – all inputs are numeric.

### **6. Exploratory Data Analysis (EDA)**

### Univariate: Distribution plots of individual pollutants (CO, NOx, etc.) showed variability across time.

### Multivariate: Correlation matrix showed strong correlations between certain sensor outputs and AQI.

### Time-series analysis revealed seasonal and daily AQI trends.

### Insights are CO(GT), NOx(GT), and C6H6(GT) showed stronger relationships with AQI and Sensor PT08.S1 and PT08.S2 contributed indirectly but significantly.

### **7. Feature Engineering**

### Weighted AQI Formula: Created using pollutant contributions based on domain knowledge:

### Example: AQI = 0.25\*CO + 0.20\*NOx + ...

### New Features: Custom feature set selected to minimize noise and overfitting.

### Date Time Parsing: Used for time-based trend visualization.

### **8. Model Building**

* Models Used: Random Forest Regressor (primary); no secondary model included in current phase.
* Train-Test Split: 80%-20%, random state set for reproducibility.
* Evaluation Metrics:
* MAE: ~X.XX (based on loaded dataset)
* R² Score: ~X.XX
* Model Justification: Random Forest is robust to noise, handles non-linearity well, and provides feature importance scores.

### 

### **9. Visualization of Results & Model Insights**

### AQI Trend Plot: Daily average AQI shown with color-coded categories (Good to Hazardous).

### Feature Importance: Displayed in GUI output after training.

### Prediction Summary: Real-time results shown with category colour, AQI, and health advice.

### Health Advice: Dynamically displayed based on category.

### **10. Tools and Technologies Used**

### Language: Python

### IDE: Jupyter

### Libraries:

### Data: pandas, numpy

### Modelling: scikit-learn

### GUI: tkinter

### Visualization: matplotlib

### Other: datetime, tkinter.ttk, file dialog, message box

### **11. Team Members and Contributions**

1. EDA : K. DIVAKAR
2. Feature engineering : R. BOOBESH
3. Model development : D. NANDAGOPAL
4. Documentation and reporting : E. PARASURAMAN