# 

PREDICTING AIR QUALITY LEVEL USING MACHINE LEARNING ALGORITHMS FOR ENVIRONMENTAL INSIGHTS

STUDENT NAME: SHAIK.KARIMULLA

REGISTRATION NUMBER :410623104103

INSTITUATION :DHAANISH AHMED COLLEGE OF ENGINEERING

DEPARTMENT :COMPUTER SCIENCE AND ENGINEERING

DATA OF SUBMISSION : 30-04-2025

1. Problem Statement

Air quality has a profound impact on public health and environmental sustainability more during the upcoming days. Existing methods for air quality assessment primarily focus on real-time monitoring rather than prediction. There is a pressing need for models that can forecast pollution levels to enable preventive action. Machine learning provides sophisticated approaches to analyse complex environmental datasets. This project seeks to build a predictive model to estimate air quality based on atmospheric and emission data. The outcome will enhance environmental monitoring systems and inform mitigation strategies.

2. Objectives of the project

* To collect and preprocess real-world air quality datasets from reliable environmental sources.
* To analyze key environmental factors (such as PM2.5, PM10, NO2, CO, temperature, humidity) affecting air quality.
* To apply suitable machine learning algorithms (like Random Forest, Support Vector Machine, or Neural Networks) for predicting air quality levels.
* To evaluate the performance of the models using appropriate metrics like accuracy, RMSE, and R² score.
* To develop a predictive system that can provide early warnings about poor air quality conditions.
* To generate actionable insights that can aid policymakers, environmental agencies, and the public in managing air pollution effectively.

3. Scope of the Project

**3.1. Data Collection and Analysis**

* Gather air quality datasets from open sources (e.g., AQI datasets, meteorological data).
* Analyse important features such as PM2.5, PM10, NO2, CO, O3, temperature, humidity, wind speed, etc.
* Understand patterns and trends influencing air pollution levels.

**3.2. Machine Learning Model Development**

* Select appropriate machine learning algorithms (Random Forest, SVM, Decision Trees, etc.).
* Train models to predict Air Quality Index (AQI) categories (e.g., Good, Moderate, Unhealthy).
* Fine-tune model parameters for better accuracy and performance.

**3.3. Model Evaluation**

* Evaluate models using metrics like Accuracy, Precision, Recall, RMSE (Root Mean Square Error), and R² Score.
* Perform cross-validation to ensure the model is robust and generalized.

**3.4. Prediction and Forecasting**

* Develop a system capable of making short-term air quality predictions (hourly/daily forecasts).
* Provide real-time insights to warn about deteriorating air quality conditions.

**3.5. Environmental Insights and Decision Support**

* Derive actionable insights for urban planners, environmental agencies, and the general public.
* Help in strategic decision-making for pollution control and preventive health measures.

**3.6. Scope Limitations**

* The project will focus only on software-based prediction, not on creating any hardware devices.
* Prediction will be based on available datasets; real-time sensor integration is outside the current project scope.
* Global predictions are not included; the model will be region-specific depending on the dataset used.

4. Data Sources

Why These Sources Were Chosen:

The selected datasets are trusted, publicly available, and widely used in environmental research and predictive modeling.

CPCB data ensures accuracy for Indian cities, making the model relevant to local conditions.

WAQI and OpenAI provide real-time updates, allowing the system to be expanded for live predictions.

UCI and Kaggle datasets add historical depth, helping the machine learning model learn from long-term pollution patterns.

These sources ensure a rich combination of real-world, real-time, and historical information, leading to a robust and reliable prediction system.

5. High level Methodology

**5.1. Data Collection**

* Gather air quality and meteorological data from multiple sources such as CPCB, WAQI, OpenAQ, and UCI repositories.
* Ensure the data includes key parameters like PM2.5, PM10, CO, NO2, O3, temperature, humidity, and wind speed.

**5.2. Data Preprocessing**

* Handle missing values, incorrect readings, and outliers using appropriate cleaning techniques.
* Normalize or scale the data if necessary to improve model performance.
* Perform feature engineering to create new relevant features if needed.

**5.3. Exploratory Data Analysis (EDA)**

* Analyze relationships between different pollutants and air quality indices.
* Visualize data patterns and trends using graphs, heatmaps, and correlation matrices.

**5.4. Model Selection and Training**

* Choose suitable machine learning algorithms (such as Random Forest, Support Vector Machine, Decision Tree, or Gradient Boosting).
* Split the data into training and testing sets.
* Train models using the training set and optimize hyperparameters to improve accuracy.

**5.5. Model Evaluation**

* Evaluate model performance using metrics such as Accuracy, RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and R² Score.
* Compare multiple models to identify the best-performing one.

**5.6. Prediction and Validation**

* Use the trained model to predict air quality levels on unseen test data.
* Validate predictions through comparison with actual air quality index values.

**5.7. Deployment and Insights Generation**

* Present the prediction results through simple visualization dashboards or reports.
* Generate actionable insights to help environmental agencies and the public make informed decisions.

6. Tools and Technologies

1. Programming Languages

Python: For data preprocessing, model building, training, and evaluation.

2. Development Environment

Jupyter Notebook: For writing and testing Python code interactively.

Google Colab: For cloud-based development with GPU support (optional).

VS Code: For coding and script development.

3. Libraries and Frameworks

Pandas: For data loading, cleaning, and manipulation.

NumPy: For numerical operations and array handling.

Scikit-learn: For implementing machine learning algorithms and model evaluation.

Matplotlib and Seaborn: For data visualization and exploratory data analysis (EDA).

4. Machine Learning Algorithms

Random Forest Classifier/Regressor

Support Vector Machine (SVM)

Decision Tree

Gradient Boosting Machines (GBM)

(Other models like Linear Regression or XGBoost can also be tested optionally.)

5. Data Sources and APIs

CPCB Datasets (Government source in CSV/Excel format)

WAQI API (Real-time air quality data access)

OpenAQ API and datasets

UCI Machine Learning Repository

6. Version Control

3

Git and GitHub: For version management and collaboration (optional but professional if needed).

7 . TEAM MEMBERS

1: SYED.MUAZ: Project Manager / Team Leader

**Responsibilities:**

* Oversees the entire project timeline, scope, and task distribution.
* Coordinates meetings, deadlines, and ensures collaboration.
* Acts as the point of contact between team and supervisor/professor.
* Ensures documentation and final presentation are aligned.

2. A PUGHLAN: Data Engineer

**Responsibilities:**

* Collects and preprocesses air quality data (from sensors, APIs, CSV files, etc.).
* Cleans, transforms, and structures the data for modeling.
* Handles missing data, normalization, encoding, and time-series formatting if needed.

3. MD.FARAAZ AHMED : **Machine Learning Engineer**

* **Responsibilities:**
  + Selects and implements suitable ML algorithms (e.g., Random Forest, XGBoost, LSTM).
  + Tunes hyperparameters and trains models.
  + Validates and evaluates model performance using metrics like RMSE, MAE, R².

4. SK.KARIMULLA: **Domain Analyst / Environmental Expert**

* **Responsibilities:**
  + Researches air quality indicators (e.g., PM2.5, PM10, CO, NO₂).
  + Interprets how model results relate to environmental standards (e.g., AQI categories).
  + Ensures that predictions make environmental sense and are interpretable.

5. P.RATHINA SABAPATHY: **Visualization & Report Specialist**

* **Responsibilities:**
  + Creates dashboards, charts, and visual summaries using tools like Matplotlib, Seaborn, or Power BI/Tableau.
  + Designs the project report, slides, and visual presentation.
  + Helps communicate findings to a non-technical audience.