Student Name: RAKESH S

Register Number: 410723104061

Institution: DHANALAKSHMI COLEGE OF ENGINEERING

Department: COMPUTER SCIENCE ENGINEERING

Date of Submission: 07/05/2025

Github Repository Link: **https://github.com/Rakesh-1746/NM_rakesh**

Project Title: Predicting air quality levels using advance machine learning algorithm for environment insights

---

1. Problem Statement

- *Air pollution poses serious risks to health and the environment, but traditional monitoring systems are limited in coverage and responsiveness. This project aims to use advanced machine learning algorithms to predict air quality levels in real time, using environmental and meteorological data, to provide accurate and timely insights for better public health and environmental decisions.*

2. Project Objectives

1. ***Develop machine learning models** to predict key air quality indicators such as PM2.5, PM10, NO_2, and O_3.*

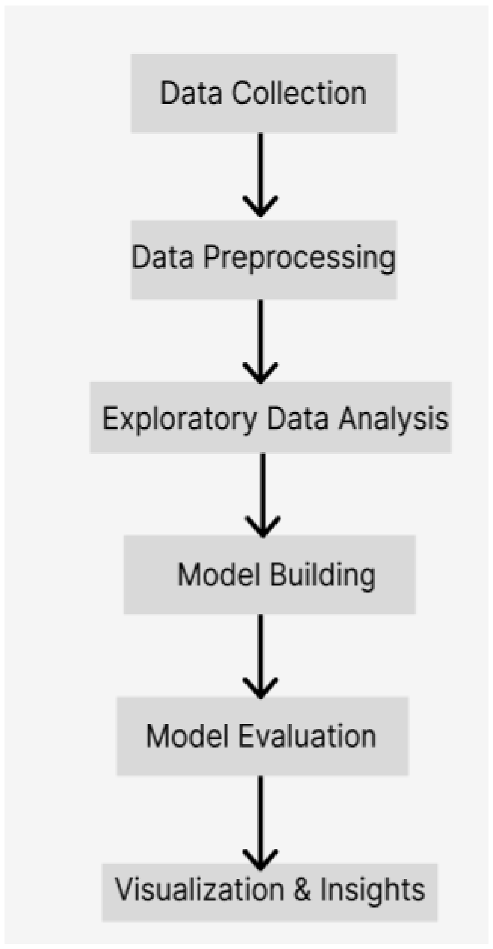
2. ***Integrate meteorological and environmental data*** *(e.g., temperature, humidity, wind speed) to enhance prediction accuracy.*

3. ***Analyze spatial and temporal patterns*** *of air pollution across different regions and time frames.*

4. ***Create a visualization dashboard*** *for real-time air quality insights and forecasts.*

5. ***Provide early warning alerts*** *for hazardous air quality levels to support public health and policy measures.*

3. Flowchart of the Project Workflow



4. Data Description

Air Quality Data
Pollutants: *PM2.5, PM10, NO_2, SO_2, CO, O_3*
Source: *OpenAQ, UCI, government APIs*
Format: *Time series (hourly/daily) with location and timestamp*
Meteorological Data
Features: *Temperature, humidity, wind speed, pressure, rainfall*
Source: *OpenWeatherMap, weather stations*
Synced by location and time
Geographic & Temporal Data
Includes: *Latitude, longitude, altitude, date, time, weekday/weekend*
Used for spatial-temporal analysis
(Optional) Traffic/Industrial Data
For enhanced prediction in urban areas
All datasets are cleaned, merged, and used to train ML models for air quality prediction.

5.Data Preprocessing

1.Data Cleaning

1. Handle missing values using interpolation, forward/backward fill, or mean imputation.
2. Remove or flag outliers using statistical methods (e.g., z-score, IQR).

2.Data Integration

1.Merge air quality, weather, and location data by timestamp and geographic coordinates.

2.Align data frequency (e.g., resample hourly data to daily if needed).

3.Feature Engineering

1.Add lag features and rolling averages for time series trends.

2.Normalize or scale numerical data (e.g., Min-Max Scaling, Standardization).

4.Encoding Categorical Variables

1. Convert time-based categories (e.g., day of week) using one-hot encoding or label encoding.

5.Train-Test Split

1.the dataset chronologically (not randomly) for time series modeling (e.g., 80% train, 20% test)

6.**Data exploratory analysis (EDA**)

1. ***Descriptive Statistics***
   - *Analyzed mean, median, min, max, and standard deviation for pollutants (PM2.5, PM10, NO_2, etc.) and weather features (temperature, humidity, wind speed).*
   - *Identified extreme pollution levels and seasonal trends.*
2. ***Time Series Visualization***
   - *Line plots showed daily and monthly trends in pollutant levels.*
   - *Detected peaks in winter and during rush hours (in urban datasets).*
3. ***Correlation Analysis***
   - *Heatmaps revealed strong correlations between weather conditions (e.g., low wind speed, high humidity) and higher pollutant concentrations.*
   - *Some pollutants (PM2.5 and PM10) were highly correlated.*
4. ***Distribution Analysis***
   - *Histograms and boxplots indicated that PM2.5 and NO_2 had right-skewed distributions.*
   - *Applied log transformation to normalize heavily skewed variables.*
5. ***Pollution Patterns by Time***
   - *Grouped data by hour, day, and month to find pollution patterns.*
   - *Noticed higher pollution during early mornings and in colder months.*
6. ***Missing Data Analysis***
   - *Identified missing values in some sensors and weather readings.*

- *Used interpolation and time-based imputation to fill gaps.*

8.Feature engineering

1. ***Time-Based Features***
   - ***Hour, Day, Month, Weekday/Weekend****: Captures temporal trends (e.g., higher pollution during peak hours or weekdays).*
   - ***Season****: Categorical variable for seasonal variations in air quality.*
2. ***Lag Features***
   - ***Lagged pollutant values*** *(e.g., PM2.5 lagged by 1–3 hours/days): Helps the model capture temporal dependencies.*
   - ***Rolling averages*** *(e.g., 3-day or 7-day rolling mean): Smooths out fluctuations and highlights trends.*
3. ***Meteorological Interactions***
   - *Combine features like* ***Temperature*** × ***Humidity*** *or* ***Wind Speed*** × ***Pressure*** *to capture complex weather effects.*
4. ***Air Quality Index (AQI) Category***
   - *Convert numerical AQI into categorical labels (e.g., Good, Moderate, Unhealthy) for classification tasks.*
5. ***Location-Based Features***
   - *If data from multiple areas: include* ***latitude****,* ***longitude****,* ***altitude****, or even* ***urban/rural tags****.*

9.Model Building

1. Problem Type

- **Regression** task to predict continuous values like PM2.5, PM10, or AQI.
- Can also be framed as **classification** (e.g., predicting AQI categories like

2. Model Selection

Use and compare different machine learning models:

- **Baseline Model**:
  - **Linear Regression** – for simple benchmarking.
- **Tree-Based Models**:
  - **Random Forest Regressor** – handles non-linear relationships well.
  - **Gradient Boosting** (e.g., XGBoost, LightGBM) – highly accurate for structured data.

- **Support Vector Machine**:
  - **SVR (Support Vector Regression)** – works well with high-dimensional data.
- **Neural Networks**:
  - **MLP (Multilayer Perceptron)** – for complex non-linear mapping.
  - **LSTM** – if you're working with time series data for sequential forecasting.

9. Visualization of Results & Model Insights

1. Predicted vs Actual Values

- Line Plot or Scatter Plot to compare predicted vs actual pollutant values (e.g., PM2.5).
- Helps assess how well the model tracks real-world values.

2. Residual Analysis

- Residual plots to visualize prediction errors.
- Should show random distribution; patterns may indicate underfitting or overfitting.

3. Feature Importance

- For tree-based models (like Random Forest or XGBoost), show feature importance bar chart.
- Highlights which variables (e.g., temperature, time of day) most influence pollution predictions**.**

4. Correlation Heatmap

- Displays how pollutants and weather features relate.
- Useful for identifying strong predictors during analysis**.**

5. Time Series Forecast Plot

- Overlay actual and predicted pollution values over time.
- Ideal for LSTM or any time series model to show trend alignment.

10. Tools and Technologies Used

1. Programming Language

- **Python** – Main language for data handling, modeling, and visualization

2. Libraries & Frameworks

- **Data Processing**:
  - Pandas, NumPy – for data manipulation and analysis

- **Visualization**:
    - Matplotlib, Seaborn, Plotly – for charts and dashboards
- **Machine Learning**:
    - Scikit-learn, XGBoost, LightGBM, TensorFlow / Keras – for building and training models
- **Time Series Modeling** (if used):
    - statsmodels, Prophet, LSTM (Keras)

3. Data Sources & APIs

- **OpenAQ**, **UCI Repository**, **OpenWeatherMap API** – for air quality and meteorological data

4. Development Environment

- **Jupyter Notebook** or **Google Colab** – for interactive coding and experimentation
- **VS Code** – for development and script management

5. Model Deployment (Optional)

- **Flask / FastAPI** – to build a simple API for model predictions
- **Streamlit / Dash** – to create an interactive web app
- **Heroku / AWS / Azure** – for deploying the app or API online

11. Team Members and Contributions

Team leader: Nithin chander.R

*****_data collection &data cleaning_

Team member:Vishwanathan.P

*****_Exploratory data analysis_

Team member: Santhosh.S

*****_tools&technologies_

Team member:Rakesh.S

*****Visualization and insights

Team member: Sajit kumar E

*****model evalution and optimization