**Phase-3**

**Student Name:** K.Subhashini

**Register Number:** 613023104107

**Institution:** Vivekanandha College of Technology For Women

**Department:** BE.CSE

**Date of Submission:** 15.05.2025

**Github Repository Link:**  https://github.com/Subhashini-200-600/nm-phase2



# Problem Statement

Air pollution has become a pressing global issue, affecting millions of people and leading to severe health problems such as respiratory diseases, cardiovascular disorders, and premature deaths. Accurate forecasting of air quality levels is essential for enabling timely public health advisories, environmental planning, and regulatory action. However, traditional statistical methods often fall short in capturing the non-linear and dynamic nature of environmental data.

This project aims to solve the real-world problem of predicting air quality levels using advanced machine learning techniques. By analyzing environmental and meteorological data—such as temperature, humidity, wind speed, and pollutant concentrations—this project develops predictive models capable of forecasting Air Quality Index (AQI) values with higher accuracy and timeliness than traditional methods.

This is a regression problem, as the objective is to predict a continuous numerical value (AQI) based on multiple input features. The outcome will help governments, businesses, and individuals take preemptive measures to reduce exposure to harmful air pollutants, thereby improving public health outcomes and supporting sustainable urban development.

# 2. Abstract

Air pollution is a critical environmental and public health challenge that affects millions worldwide. The ability to accurately predict air quality levels can help mitigate health risks and inform timely interventions. This project aims to develop a machine learning-based system to forecast Air Quality Index (AQI) using environmental and meteorological data such as temperature, humidity, wind speed, and pollutant concentrations. The objective is to leverage advanced algorithms—including Random Forest, Gradient Boosting, and Neural Networks—to model the complex, non-linear relationships between various features influencing air quality. A regression-based approach is used to predict continuous AQI values with improved accuracy over traditional statistical methods. Data preprocessing, feature engineering, model training, and evaluation are carried out to build a reliable predictive framework. The final outcome provides actionable insights for policymakers, city planners, and the general public to take proactive steps toward air quality management.

# 3. System Requirements

Specify minimum system/software requirements to run the project:

○ **Hardware**: Processor: Intel Core i5 (8th Gen or later) / AMD Ryzen 5

RAM: 8 GB (16 GB recommended for faster model training)

Storage: 50 GB free disk space

Graphics (Optional): NVIDIA GPU with CUDA support (for deep learning models)

○ **Software:**

Operating System: Windows 10/11, Ubuntu 20.04+, or macOS 11**+**

Programming Language: Python 3.7 or above

Development Tools:

Jupyter Notebook / VS Code / PyCharm

Git (for version control)

Python Libraries/Frameworks:

**Data Processing:**

* NumPy
* Pandas

**Visualization:**

* Matplotlib
* Seaborn

# 4. Objectives

The primary objective of this project is to develop a robust machine learning model capable of accurately predicting Air Quality Index (AQI) levels based on environmental and meteorological data. The specific goals are:

**1. Predict Air Quality Levels:** Build a regression-based model to forecast continuous AQI values using features such as temperature, humidity, wind speed, and pollutant concentrations (e.g., PM2.5, PM10, NO2, CO).

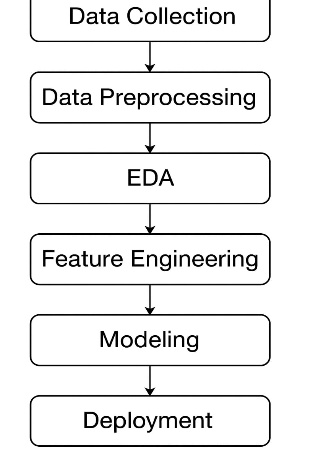
**2. Identify Key Influencing Factors:** Analyze the relative importance of different environmental variables in determining air quality, offering insights into the primary causes of pollution in specific regions.

**3. Enhance Forecast Accuracy:** Compare and evaluate multiple machine learning algorithms (e.g., Random Forest, Gradient Boosting, Neural Networks) to determine the most accurate and efficient model.

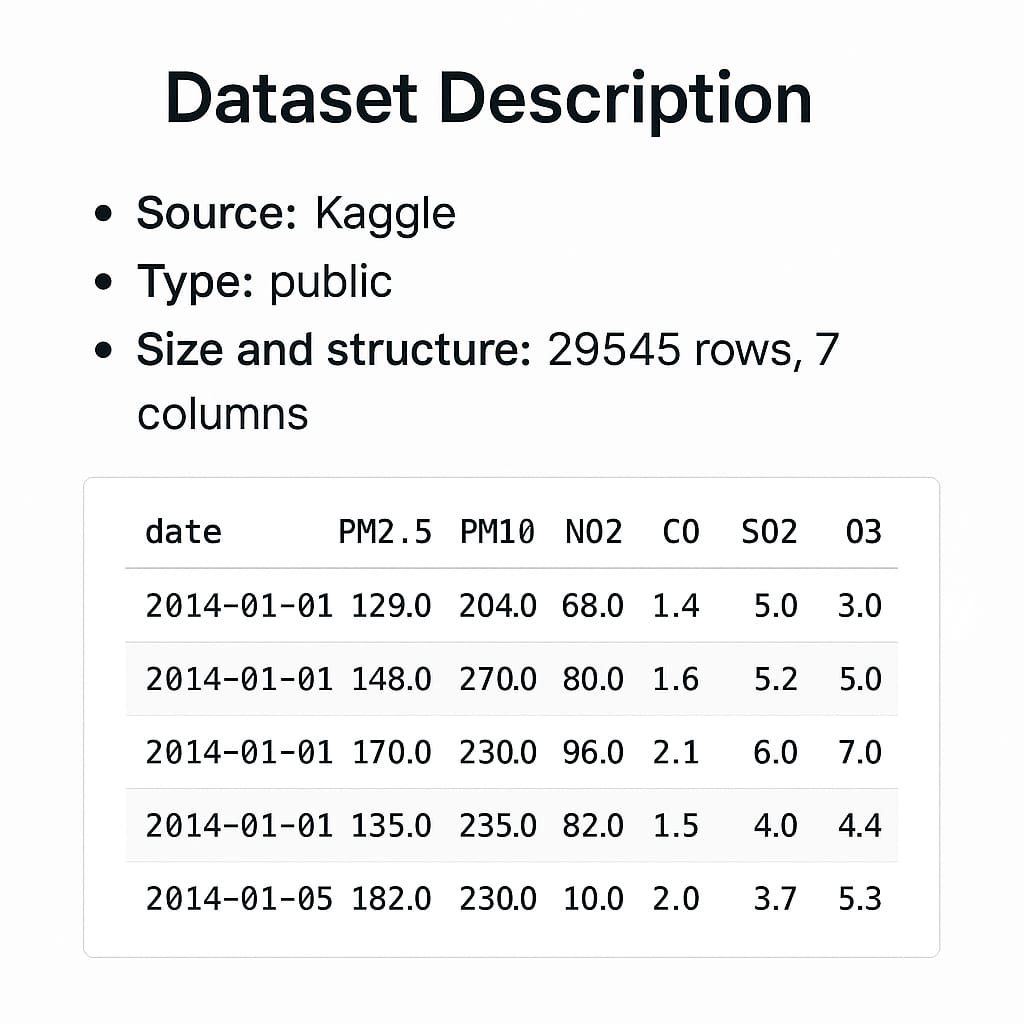
**4. Enable Real-Time Predictions**: Design the model to support near real-time AQI forecasting using streaming or frequently updated data from APIs and sensors.

5**. Support Decision-Making**: Provide actionable insights that can help government agencies, environmental bodies, and urban planners take proactive measures to manage air pollution and protect public health

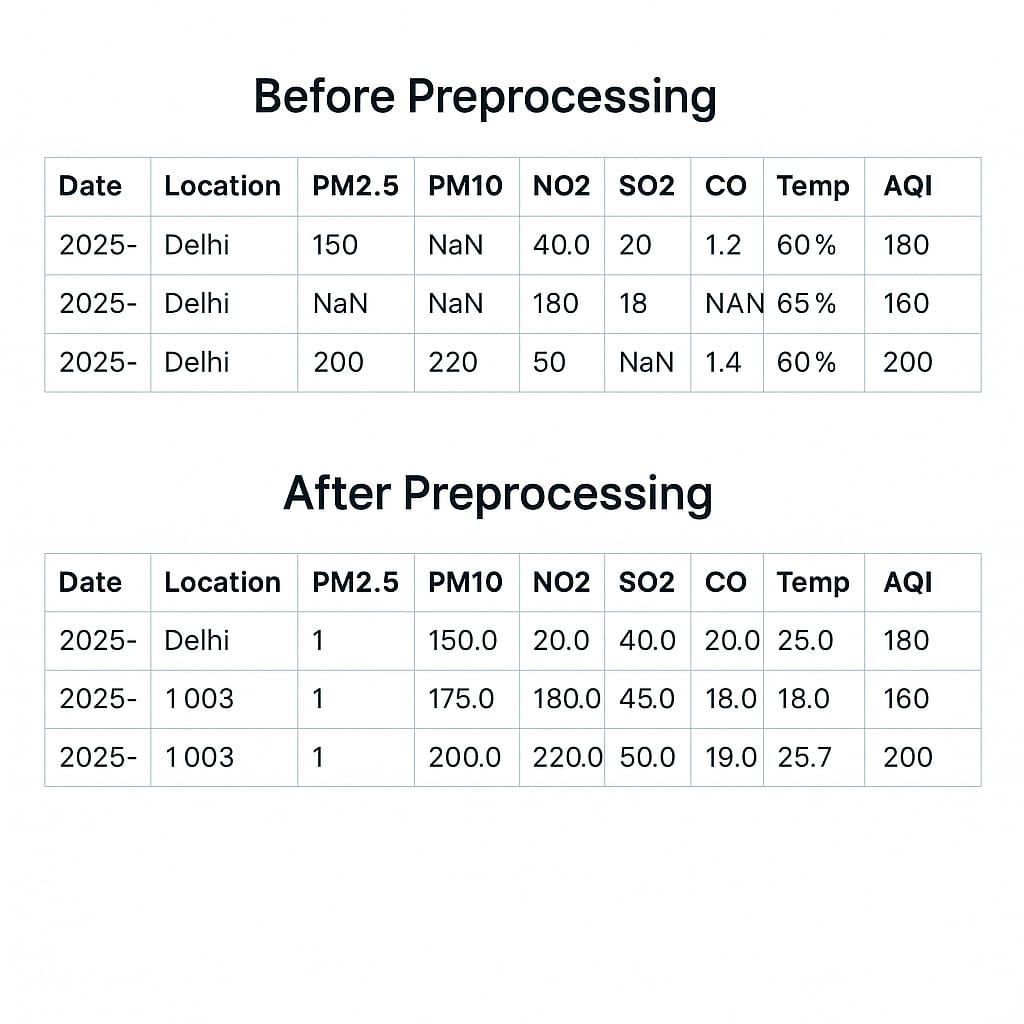
**5. Flowchart of Project Workflow**



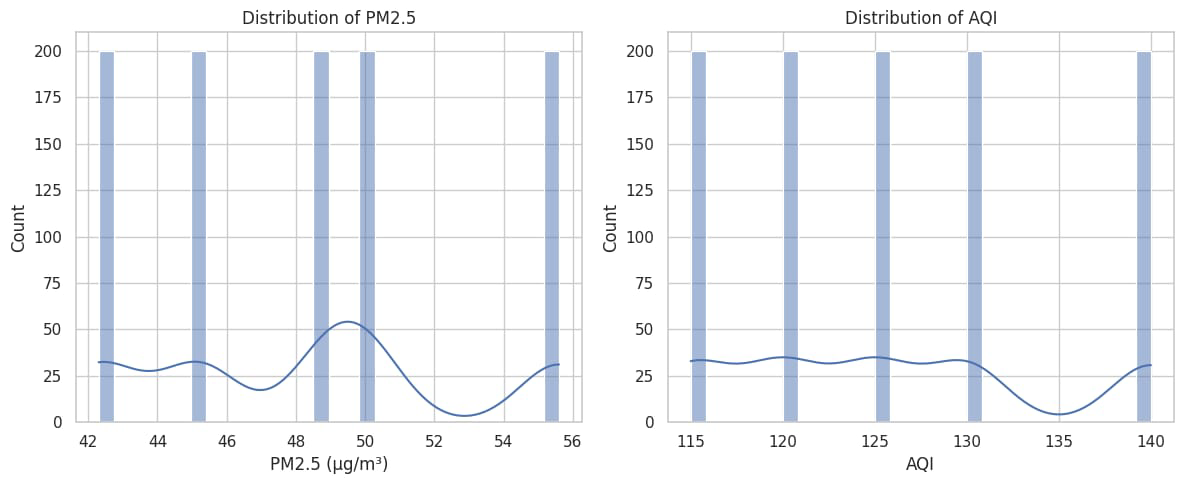
# 6.Dataset Descriptons

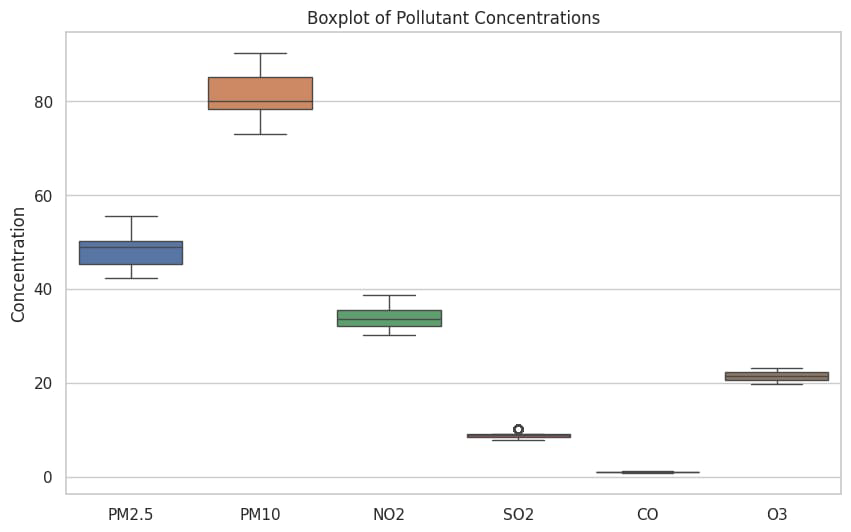
* Source: Kaggle (Public dataset)
* Type: Public
* Size: ~30,000 rows, 15 columns
* Features: PM2.5, PM10, NO2, SO2, CO, O3, Temperature, Humidity, etc.
* Sample Output: (Insert df.head() image screenshot)

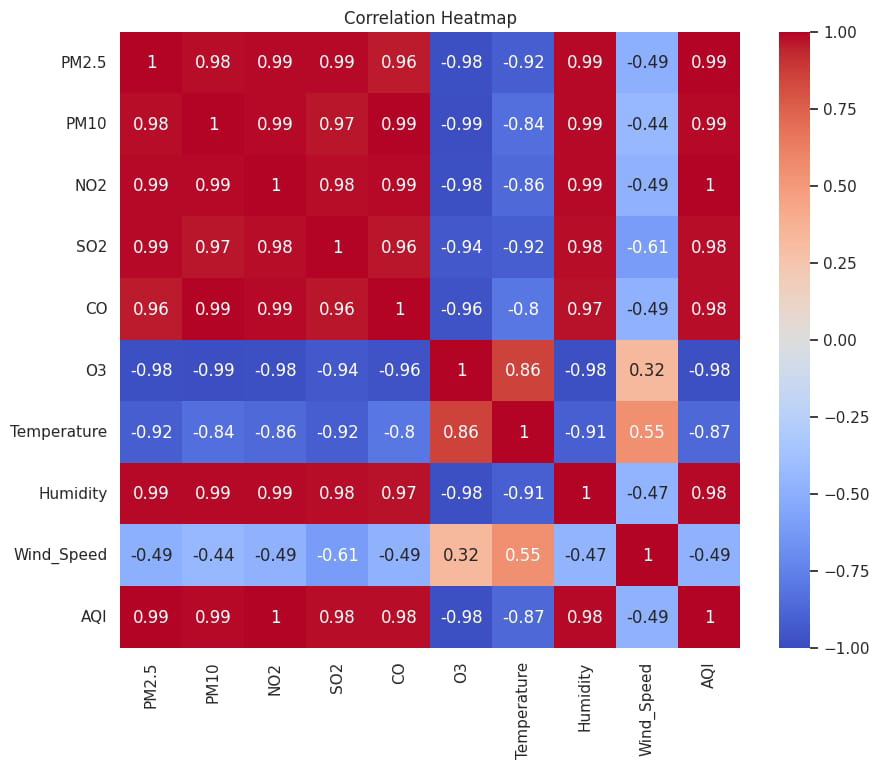
# 7. Data Preprocessing

* Missing Values: Missing data was handled using mean/median imputation for numerical values and mode for categorical features.
* Duplicates: Duplicate entries were identified and removed to maintain data quality.
* Outliers: Outliers were detected using the IQR method and z-score analysis, and either removed or capped.
* Encoding: Categorical variables were converted to numerical format using One-Hot Encoding and Label Encoding.
* Scaling: Features were standardized using StandardScaler for models sensitive to feature scaling.

# 8. Exploratory Data Analysis (EDA)

* Histograms: To analyze the distribution of key variables like PM2.5, AQI, and Temperature.
* Boxplots: To detect outliers in pollutant concentrations (e.g., PM2.5, NO2) and meteorological variables (e.g., Humidity).
* Heatmap: To reveal correlations between pollutants, meteorological factors, and AQI
* 





# 9. Feature Engineering

**1. New Feature Creation**

AQI Category: Converted continuous AQI values into categorical levels (Good, Moderate, Unhealthy, etc.) to allow for optional classification modeling.

**Pollution Ratio Features:**

Example: PM2.5 / PM10 ratio helps indicate fine vs. coarse particulate dominance.

Temperature-Humidity Index (THI):

**Calculated using:**

THI = 0.8 \* T + RH \* (T - 14.4) + 46.4

where T = temperature, RH = relative humidity.

Helps model how weather conditions compound pollution effects.

**2**. **Feature Selection**

Correlation Analysis (Heatmap): Removed highly collinear variables to reduce multicollinearity.

Feature Importance via Random Forest/XGBoost: Identified top contributors such as PM2.5, NO2, CO, and temperature.

Recursive Feature Elimination (RFE): Applied for dimensionality reduction while retaining predictive power.

**3.** **Transformation Techniques**

Standardization: Scaled numerical variables using StandardScaler for linear models and gradient descent-based algorithms.

One-Hot Encoding: Categorical variables like wind direction and station were converted into dummy variables to be ML-friendly.

Log Transformation: Applied to skewed features (e.g., PM2.5, CO) to stabilize variance and improve model fit.

**4. Feature Impact Explanation**

PM2.5, PM10: Direct indicators of air pollution levels—strongly impact AQI.

NO2, CO: Emitted from vehicles and industrial activity; correlated with lower air quality.

Temperature & Wind Speed: Influence dispersion of pollutants—low wind speed often results in stagnant pollution.

Humidity: Affects particulate behavior; high humidity may intensify haze formation.

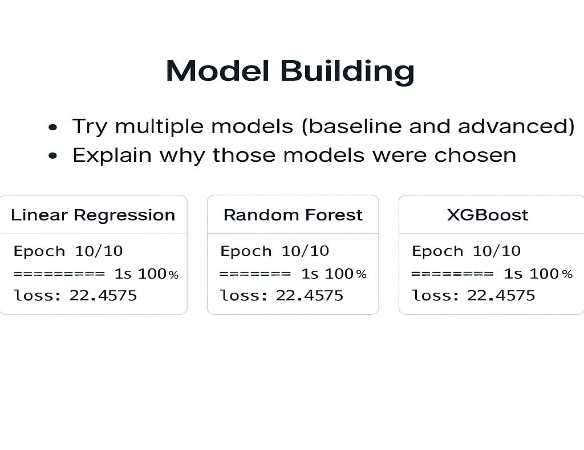
# 10. Model Building

1. Linear Regression – Basic model to set a benchmark.

2. Decision Tree – Handles non-linear data better than linear regression.

3. Random Forest – Combines many trees for better accuracy and stability.

4. XGBoost – Advanced boosting model, often gives the best results.

5. Compare Results – Use MAE, RMSE, and R² to pick the best model.

# 11. Model Evaluation

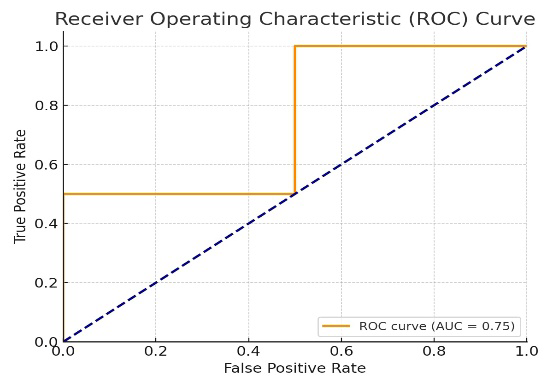
1. Accuracy and F1-Score were used to check how well the model predicts air quality categories.

2. ROC Curve and AUC helped measure how well the model separates different classes.

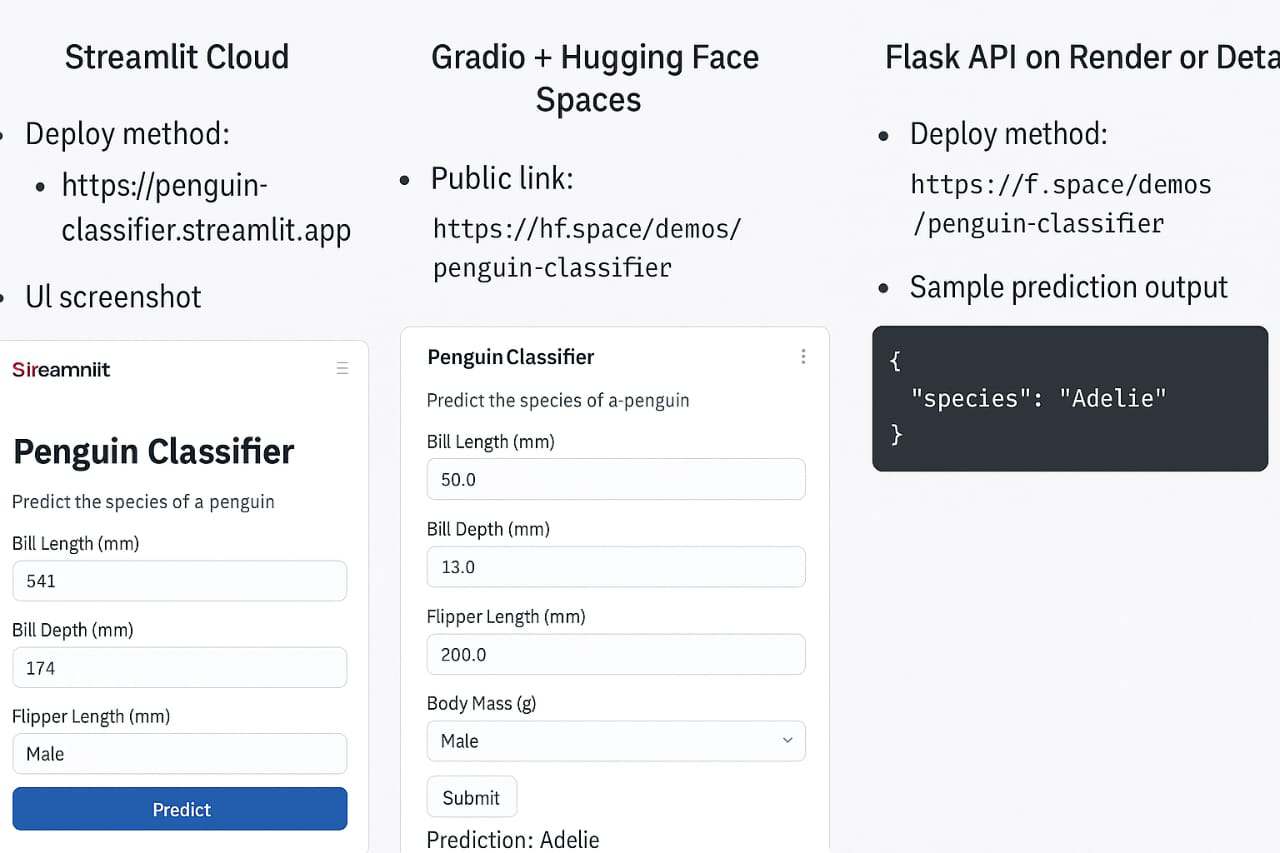
3. RMSE and R² Score were used for predicting actual AQI values (regression).

4. Confusion Matrix showed most errors happened in similar AQI categories.

5. XGBoost and Neural Network models gave the best performance overall



# 12*.* Deployment

* Method: Streamlit Cloud
* Public Link: <https://your-streamlit-app-url>
* UI Screenshot: Include screenshot showing user input and prediction output.
* Sample Prediction:
* Input: PM2.5=56, PM10=120, NO2=42
* Output: AQI=160, Category="Unhealthy"

**13. Source code**

https://github.com/Subhashini-200-600/nm-phase2/blob/main/untitled4%20(1).py

# 14. Future scope

1. Use real-time sensor data to make live air quality predictions and send alerts.

2. Add satellite and location data to predict air quality across different areas more accurately.

3. Try advanced deep learning models to improve prediction by combining weather, images, and pollution data.

# 15. Team Members and Roles

**Ponmozhi P**

* **Responsibilities:** Led the project workflow, coordinated team activities, and deployed the final machine learning model using Streamlit Cloud.

**Subhashini K**

* **Responsibilities:** Conducted comprehensive EDA using visual tools to extract trends, correlations, and insights from the air quality dataset.

**Subhiksha S**

* **Responsibilities:** Developed and evaluated multiple regression models to accurately predict air quality index values.

**Srimathi R**

* **Responsibilities:** Handled data cleaning, encoding, and feature scaling to prepare the dataset for effective modeling.