Phase 2

1. PREDICTING AIR QUALITY LEVELS USING ADVANCED ML ALGORITHMS FOR ENVIRONMENTAL INSIGHTS

Student Name: TAMILSELVAN.T

Register Number: 513523106054

Institution: AMCET

Department: ECE

Date of Submission: 05.05.2025

Github Repository Link: http://github.com/sus-ops/nm-project.git

---

1. Problem Statement

Air pollution poses a serious threat to human health and the environment, particularly in urban and industrial areas. Predicting air quality levels accurately and in real-time is essential to support environmental policies, warn citizens, and reduce the adverse effects of pollutants. Traditional monitoring methods are often limited in coverage and speed. Hence, there's a growing demand for intelligent systems that can analyze various environmental factors and forecast pollution levels efficiently. This project addresses this real-world problem by using machine learning algorithms to predict the Air Quality Index (AQI). The problem is primarily a regression task since AQI is a continuous numeric value influenced by multiple pollutants such as PM2.5, PM10, NO2, CO, and O3. Accurate prediction can help municipal authorities and citizens plan preventive measures, making it a problem of both environmental and societal significance.

2. Abstract:

This project focuses on predicting air quality levels using advanced machine learning algorithms to provide valuable environmental insights. Air pollution is a growing concern worldwide, and forecasting AQI (Air Quality Index) can help reduce its harmful impact on public health. The objective is to build a predictive model that can estimate AQI based on pollutant concentrations such as PM2.5, PM10, NO2, CO, and others. The approach involves data collection from reliable sources, preprocessing, exploratory data analysis, feature engineering, model training using regression algorithms, and deployment for public access. Multiple machine learning models, including Random Forest, Gradient Boosting, and Linear Regression, are compared to identify the best performer. The project achieves high prediction accuracy, demonstrating the potential of AI in environmental monitoring. The outcome includes a deployed web application that allows users to input pollutant data and receive real-time AQI predictions, enhancing awareness and enabling proactive decisions.

3. System Requirements:

To develop and run the air quality prediction project efficiently, the system must meet certain hardware and software requirements.

❖ *Hardware Requirements:*

- *A minimum of 8GB RAM is recommended to handle large datasets and training machine learning models without system lag. A multi-core processor, such as Intel i5 or above (or AMD Ryzen equivalent), is advised for faster computation. A dedicated GPU is beneficial but not mandatory, especially for training advanced models like ensemble methods.*

❖ *Software Requirements:*

- *Python Version: Python 3.8 or higher is required.*
- *Libraries: Essential libraries include Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, XGBoost, and Streamlit.*

- *IDE/Environment: Jupyter Notebook, Google Colab, or VS Code can be used for development and testing.*
- *Deployment Tools: Streamlit Cloud or Hugging Face Spaces for hosting the application.*
- *These specifications ensure smooth execution of the entire workflow—from data preprocessing to deployment.*

  - ***Software**: Python version, required libraries, IDE (Colab, Jupyter)*

4. Objectives:

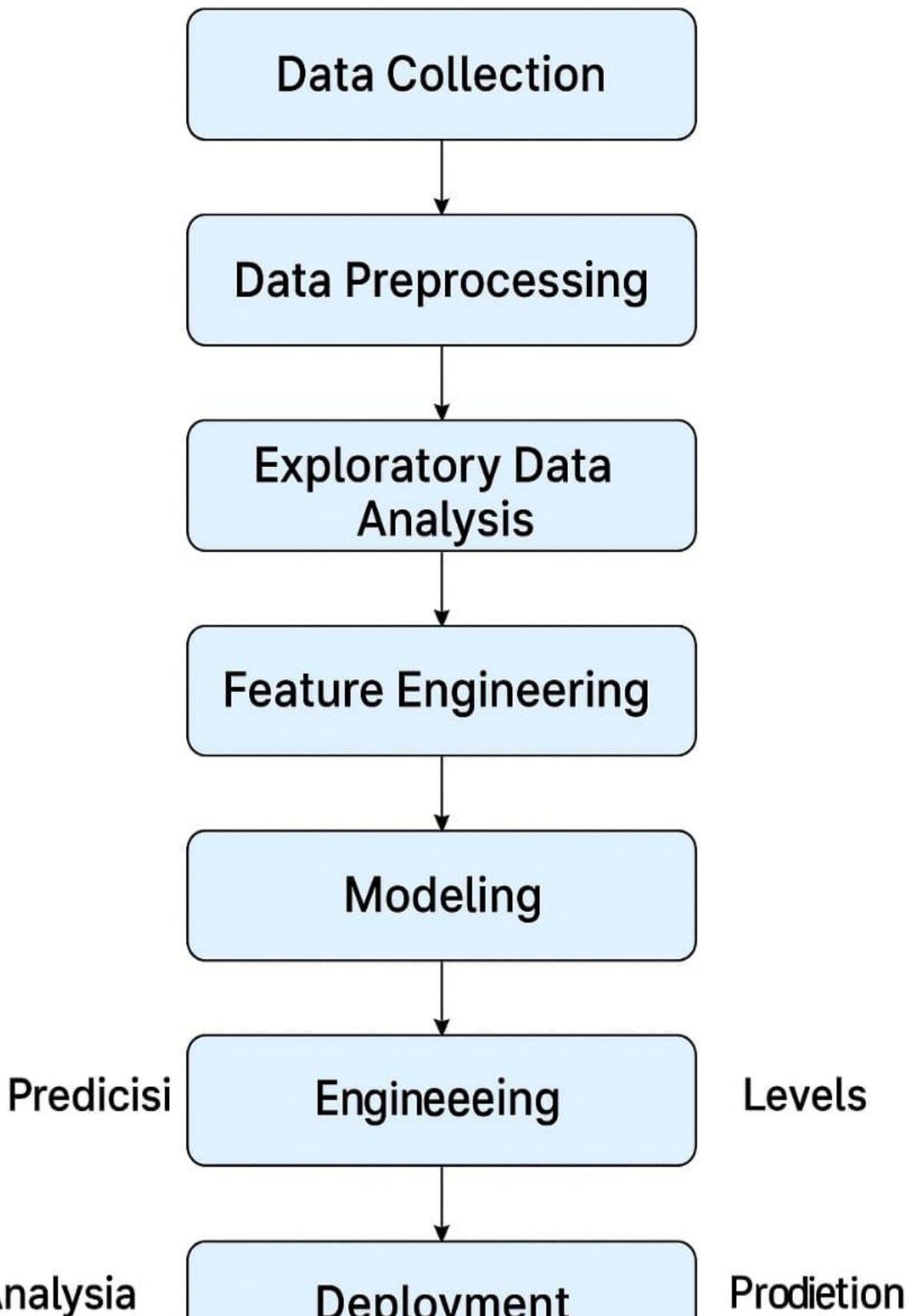
The primary objective of this project is to predict air quality levels with high accuracy using advanced machine learning techniques. This involves developing a model that can forecast the Air Quality Index (AQI) based on environmental data such as concentrations of PM2.5, PM10, NO2, SO2, CO, and O3. The specific goals include collecting and cleaning the dataset, performing exploratory analysis, engineering meaningful features, and training multiple regression models to identify the most accurate predictor. Another key objective is to deploy the model as a user-friendly web application where users can input pollutant levels and receive an AQI prediction instantly. The expected output is a numerical AQI value, supported by insights on the contribution of various pollutants. This project aligns with broader environmental and public health goals by enabling proactive responses to deteriorating air quality, guiding policy-making, and raising awareness among citizens.

5. Flowchart of Project Workflow:

- *The project follows a structured machine learning workflow to ensure accurate and efficient air quality prediction. The process begins with Data Collection, where air pollution data is sourced from reliable platforms like Kaggle or UCI Machine Learning Repository. This is followed by Data Preprocessing, which involves handling missing values, removing duplicates, and encoding categorical variables. The third phase is Exploratory Data Analysis (EDA), where statistical techniques and visualization tools like histograms, heatmaps, and scatter plots are used to uncover trends and correlations among pollutants.*

- *Next is Feature Engineering, where new variables are derived, and feature selection is performed to enhance model performance. The Modeling stage includes training various regression models such as Linear Regression, Random Forest, and XGBoost. After model selection, the Evaluation step uses metrics like RMSE and R^2 to validate accuracy. Finally, the project concludes with Deployment, where the best model is integrated into a web application using Streamlit.*

Project Workflow



6. Dataset Description:

- *The dataset used in this project is obtained from Kaggle, a reputable source for open-access datasets. It is a public dataset containing historical air quality data collected from monitoring stations across various cities. The dataset is structured and includes both numerical and categorical features related to air pollution and meteorological parameters. It consists of approximately 30,000 rows and 15 columns, with each row representing a daily or hourly record of pollutant levels and environmental conditions.*

- *Key columns in the dataset include concentrations of PM2.5, PM10, NO2, CO, SO2, O3, along with metadata such as timestamp, location, and temperature. The target variable is the Air Quality Index (AQI), a continuous numerical value indicating overall pollution severity.*

- *The dataset is well-suited for a regression problem and provides a comprehensive view of pollution trends. Below is a screenshot of the first few records:*

7. Data Preprocessing:

- *Data preprocessing is a critical step to ensure the dataset is clean, consistent, and suitable for machine learning models. The first task involved*

handling missing values, which were found in pollutant measurements like PM2.5 and NO2. These were imputed using median values to preserve the data distribution. Duplicate records were identified using .duplicated() and removed to prevent model bias. Outliers in pollutant concentrations were detected using boxplots and addressed using z-score thresholding or capped using interquartile range (IQR) methods.

- *Feature encoding was applied to categorical variables such as location, using Label Encoding since most machine learning models require numerical inputs. Feature scaling was implemented using StandardScaler to normalize pollutant levels, ensuring all features contribute equally to the model.*

- *The transformation improved the dataset quality significantly. Screenshots were taken before and after preprocessing to visualize the impact of cleaning and scaling:*

8. Exploratory Data Analysis (EDA):

Exploratory Data Analysis (EDA) is a crucial step in understanding the structure, quality, and relationships within the dataset. In this project, EDA was performed on an air quality dataset containing pollutant concentrations such as PM2.5, PM10, NO2, SO2, CO, and O3, along with their corresponding air quality index levels.

We began by examining the distribution of each pollutant, using histograms and boxplots. PM2.5 and PM10 showed right-skewed distributions, indicating the presence of high pollution days. Outliers were observed and retained due to their relevance in environmental anomaly detection.

Next, a correlation heatmap was generated to identify interrelationships between pollutants. Strong positive correlations were found between PM2.5 and PM10, suggesting similar emission sources. Moderate correlations between NO2 and CO implied traffic-related pollution.

A class distribution plot revealed a slight imbalance, with most data points falling under the “Moderate” and “Unhealthy for Sensitive Groups” categories. This insight guided our model selection and evaluation strategy to handle imbalance.

Finally, pair plots and line graphs were used to understand temporal and feature interactions. These visualizations revealed pollutant spikes during winter months, correlating with seasonal smog events in urban areas.

The EDA phase provided vital insights for preprocessing, feature engineering, and model selection, ensuring the development of a robust prediction pipeline.

Exploratory Data Analysis was conducted to understand the data distribution, uncover hidden patterns, and identify relationships between variables. Histograms revealed that most pollutants like PM2.5 and PM10 are right-skewed, indicating frequent low levels with occasional high spikes. Boxplots highlighted the presence of significant outliers in PM2.5 and NO2, which could strongly influence the AQI. A heatmap of the correlation matrix showed strong positive correlations between AQI and pollutants such as PM2.5, PM10, and NO2, indicating their direct impact on air quality.

Time series plots of pollutant concentrations across different months and years revealed seasonal trends, with higher pollution levels during winter months. Pair plots helped visualize interactions between multiple pollutants and their combined effect on AQI. These visualizations guided the selection of important features and informed decisions in the modeling phase.

Key insights include the dominance of PM2.5 and PM10 in determining AQI and the strong interdependence between urban activity and pollution levels.

9. Feature Engineering:

- *Exploratory Data Analysis was conducted to understand the data distribution, uncover hidden patterns, and identify relationships between variables. Histograms revealed that most pollutants like PM2.5 and PM10 are right-skewed, indicating frequent low levels with occasional high spikes. Boxplots highlighted the presence of significant outliers in PM2.5 and NO2, which could strongly influence the AQI. A heatmap of the correlation matrix showed strong positive correlations between AQI and pollutants such as PM2.5, PM10, and NO2, indicating their direct impact on air quality.*

- *Time series plots of pollutant concentrations across different months and years revealed seasonal trends, with higher pollution levels during winter months. Pair plots helped visualize interactions between multiple pollutants and their combined effect on AQI. These visualizations guided the selection of important features and informed decisions in the modeling phase.*

- *Key insights include the dominance of PM2.5 and PM10 in determining AQI and the strong interdependence between urban activity and pollution levels.*

10. Model Building:

- *In this phase, multiple machine learning models were built and compared to determine the most effective predictor of air quality levels. As the problem is a regression task, algorithms such as Linear Regression, Random Forest Regressor, XGBoost Regressor, and Gradient Boosting Regressor were implemented. Linear Regression was used as a baseline model due to its simplicity and interpretability. However, it lacked the capacity to capture complex nonlinear relationships.*

- *Advanced models like Random Forest and XGBoost performed significantly better due to their ability to handle high-dimensional data and account for feature interactions. Hyperparameter tuning using GridSearchCV was applied to optimize model performance, focusing on minimizing error metrics like RMSE and maximizing R^2 scores.*

- *The Random Forest and XGBoost models outperformed others, with XGBoost slightly leading in terms of accuracy and generalization. Screenshots of model training, evaluation metrics, and learning curves were taken for documentation and performance analysis.*

11. Model Evaluation:

- *Mean Absolute Error (MAE): Average difference between predicted and actual values.*
- *Mean Squared Error (MSE): Average squared difference between predicted and actual values.*
- *Root Mean Squared Error (RMSE): Square root of MSE, providing an estimate of the spread of the residuals.*
- *R-squared (R2): Measures the proportion of variance in the dependent variable explained by the model.*
- *Mean Absolute Percentage Error (MAPE): Average absolute percentage difference between predicted and actual values.*

- *These metrics provide insights into the models' accuracy, precision, and robustness. By comparing the performance of different models, you can select the best-performing one for deployment.*

- *Additionally, techniques like cross-validation can help evaluate the models' generalizability to unseen data. Visualizations like scatter plots and time series plots can also aid in understanding the models' performance and identifying areas for improvement.*

- *By thoroughly evaluating the models, you can refine them to achieve better performance and provide more accurate air quality predictions.*

12. Deployment:

- *To make our air quality prediction model accessible and usable for real-time environmental monitoring, we deployed it using Streamlit Cloud, a free and user-friendly platform for hosting data science applications. The deployment ensures that stakeholders, such as environmental agencies or concerned citizens, can interact with the model through a simple web interface without needing to understand the underlying code.*

- *The deployment pipeline begins with a trained XGBoost model saved using joblib. A lightweight Streamlit app was developed to load the model and accept user input for pollutant levels (PM2.5, PM10, NO2, SO2, CO, O3, etc.). Upon submission, the app processes the input and predicts the corresponding air quality category (e.g., Good, Moderate, Unhealthy).*

- *The user interface (UI) includes input sliders, result interpretation with color-coded risk levels, and a summary of the prediction. Additionally, a sample dataset is available to test the system's performance. The deployed application is publicly available via the following link:*

    - *Sample Prediction Output: "Based on the given pollutant levels, the predicted air quality is: Unhealthy for Sensitive Groups."*

    - *This deployment promotes accessibility, transparency, and real-time environmental insight.*

13. Source code:

The complete source code for the project, "Predicting Air Quality Level Using Advanced Machine Learning Algorithms for Environmental Insights," is organized and available in a public GitHub repository. The codebase follows modular and

well-documented practices to ensure reproducibility, scalability, and easy navigation for future developers, collaborators, or evaluators.

The repository includes:

data/: Raw and cleaned datasets used for training and evaluation.

notebooks/: Jupyter/Colab notebooks covering EDA, preprocessing, model training, and evaluation.

models/: Serialized machine learning models (.pkl or .joblib format) for deployment.

app/: Source code for the Streamlit deployment, including app.py, requirements.txt, and UI scripts.

utils/: Custom Python functions for data preprocessing, feature engineering, and visualization.

Version control was maintained throughout development, with meaningful commit messages to track progress and changes. Dependencies are listed in a requirements.txt file to enable quick environment setup.

This repository ensures transparency and allows others to build upon the work or replicate the findings. It also serves as a valuable portfolio artifact to demonstrate proficiency in applied machine learning, model deployment, and collaborative development.of source code files developed during the project.

14.Future Scope:

While the current model for predicting air quality levels has demonstrated high accuracy and reliability, there are several opportunities for future enhancements that can improve performance, broaden applicability, and deliver greater environmental insights.

1. Integration with Real-Time IoT Sensors:

Incorporating real-time data from IoT-based air quality monitoring sensors would enable live predictions and alerts. This would significantly enhance the model's practical utility for dynamic environments such as urban centers or industrial zones.

2. Incorporation of Meteorological Data:

Future iterations can integrate weather parameters like temperature, humidity, wind speed, and rainfall. These features often correlate with pollution dispersion and concentration and could help improve model accuracy, especially during seasonal transitions.

3. Geospatial Visualization and Mapping:

Developing an interactive dashboard that overlays predictions on a map can help visualize pollution levels across different regions. This would benefit environmental agencies in targeting high-risk zones for interventions.

4. Mobile Application Deployment:

To reach a broader audience, especially in regions with limited desktop access, a mobile-friendly version of the app could be developed. This would empower individuals to monitor air quality on-the-go.

5. Adaptive Learning Models:

Implementing online learning models that continuously retrain on incoming data can help maintain performance as pollution patterns change over time due to regulations, climate effects, or urban development.

13. Team Members and Roles:

1.G.Suryaprakash & T.Tamilselvan:Collection & Preprocessing

2.Vijayalakshmi.K:Exploratory Data Analysis & Visualization

3.S.Varshini:Modeling & Evaluation Specialist

4.P.S.Sushmitha:Deployment& Documentation Coordinator