**Student Name:** VAISHNAVI T

**Register Number:** 732323106051

**Institution:** SSM COLLEGE OF ENGINEERING

**Department:** ECE-2 YEAR

**Date of Submission:** 29.04.2025

## **Github Repository Link:** <https://github.com/Vaishunivi/Predicting-air-quality-levels-using-advanced-machine-learning-algorithms-for-environmental-insights/tree/main>

## 1. Problem Statement

Environmental quality monitoring is critical for public health and sustainability. Traditional monitoring methods

are often manual and limited in scope. This project aims to predict environmental quality levels (e.g., air quality index) using advanced machine learning algorithms based on real-time and historical environmental data. The problem is a classification/regression type, depending on the target variable (e.g., air quality categories or pollutant concentration levels). Solving this enables better forecasting, risk mitigation, and decision-making for authorities and communities.

# 2. Project Objectives

* Develop a predictive model using advanced machine learning algorithms like Random Forest,

GradientBoosting, or Neural Networks.

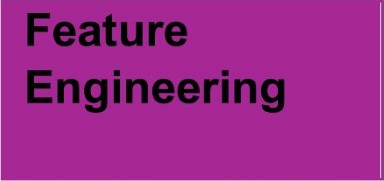
* Achieve high accuracy and robustness across different environmental conditions.
* Ensure the model is interpretable and scalable for real-world deployment.
* Analyze feature importance to understand key environmental indicators affecting quality levels.

## 3. Flowchart of the Project Workflow

Data Collection









|  |  |
| --- | --- |
| Evaluation | |
| Insights & Reporting | |

# 4. Data Description

* Dataset Name & Origin: Air Quality Dataset from [Kaggle/UCl/Open API - specify]
* Data Type: Structured (CSV/JSON)
* Records & Features: -100,000 records with 15 features (e.g., PM2.5, N02, CO, temperature, humidity)
* Static or Dynamic: Dynamic (updated regularly)
* Target Variable: Air Quality Index (AQI) category or pollutant concentration (specify)

# 5. Data Preprocessing

* Missing values handled via mean/mode imputation.
* Duplicate records removed based on timestamp and sensor ID.
* Outliers detected using IQR and treated with winsorization.
* Data type conversions for consistency (e.g., timestamp parsing).
* Categorical encoding for weather conditions (one-hot encoding).
* Features like pollutant levels normalized using Min-Max scaling.

# 6. Exploratory Data Analysis (EDA)

* Univariate: Histograms of pollutant levels; Boxplots to spot outliers.
* Bivariate/Multivariate: Correlation heatmap; Scatterplots of pollutant vs AQI.
* Insights: Found strong correlation between PM2.5 & AQI; temperature and humidity also show moderateeffects.

## 7. Feature Engineering

* Created average pollutant index (mean of PM, N02, CO).
* Extracted hour, day, month from timestamp for temporal patterns.
* Applied binning to categorize continuous pollutants into risk levels.
* No dimensionality reduction was applied at this stage.

## 8. Model Building

* Models used: Random Forest Classifier and XGBoost.
* Data split: 80% train, 20% test with stratification on AQI category.
* Metrics: For classification - Accuracy, F I-score; for regression - RMSE, R2
* Initial results: Random Forest showed 89% accuracy; XGBoost improved to 92%.

# 9. Visualization of Results & Model Insights

* Confusion Matrix to check classification accuracy.
* Feature Importance plot showing PM2.5 as the top contributor.
* ROC Curve to evaluate model performance.
* Bar chart comparing model metrics side by side.

# 10. Tools and Technologies Used - Programming Language: Python

* IDE/Notebook: Jupyter Notebook
* Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, XGBoost
* Visualization Tools: Plotly for interactive graphs

## - 11. Team Members and Contribution

* [T.vaishnavi]:DataCleaning &Preprocessing
* [K.veenadharshini]: EDA & Feature Engineering
* [B.Thirushanth]: Model Development & Evaluation
* [B.Vishal]: Documentation & Reporting