Student Name: Kousalya.A**,** Mahalakshmi.P**,** Mamatha sri. U**,** Moniga R

RegisterNumber:511923205028**,**511923205030**,**511923205031**,**511923205033

Institution Name : Priyadarshini Engineering College

Department:B.Tech-IT

Date of Submission:

Github Link:

Project Title:

Predicting Air Quality Level Using Advanced Machine Learning Algorithms for Environmental Insights

PHASE-2

1. Problem Statement

Air pollution poses serious health and environmental challenges across the globe. The goal of this project is to predict air quality levels using a variety of atmospheric and environmental variables. This can help in issuing timely warnings, implementing mitigation strategies, and forming data-driven environmental policies.

This project involves building a machine learning model that classifies or estimates the Air Quality Index (AQI) based on features such as pollutant concentrations, weather conditions, and temporal data. Depending on the dataset, the problem type may be either regression (predicting continuous AQI values) or classification (predicting AQI categories).

we need to implement efficient air quality monitoring models which collect information about the concentration of air pollutants and provide assessment of air pollution in each area.

The significance of solving this problem lies in proactive air pollution management, public health safety, and smart city development.

2. Project Objectives

- Develop machine learning models to predict air quality levels accurately

- Identify key features (pollutants, weather factors) affecting AQI.

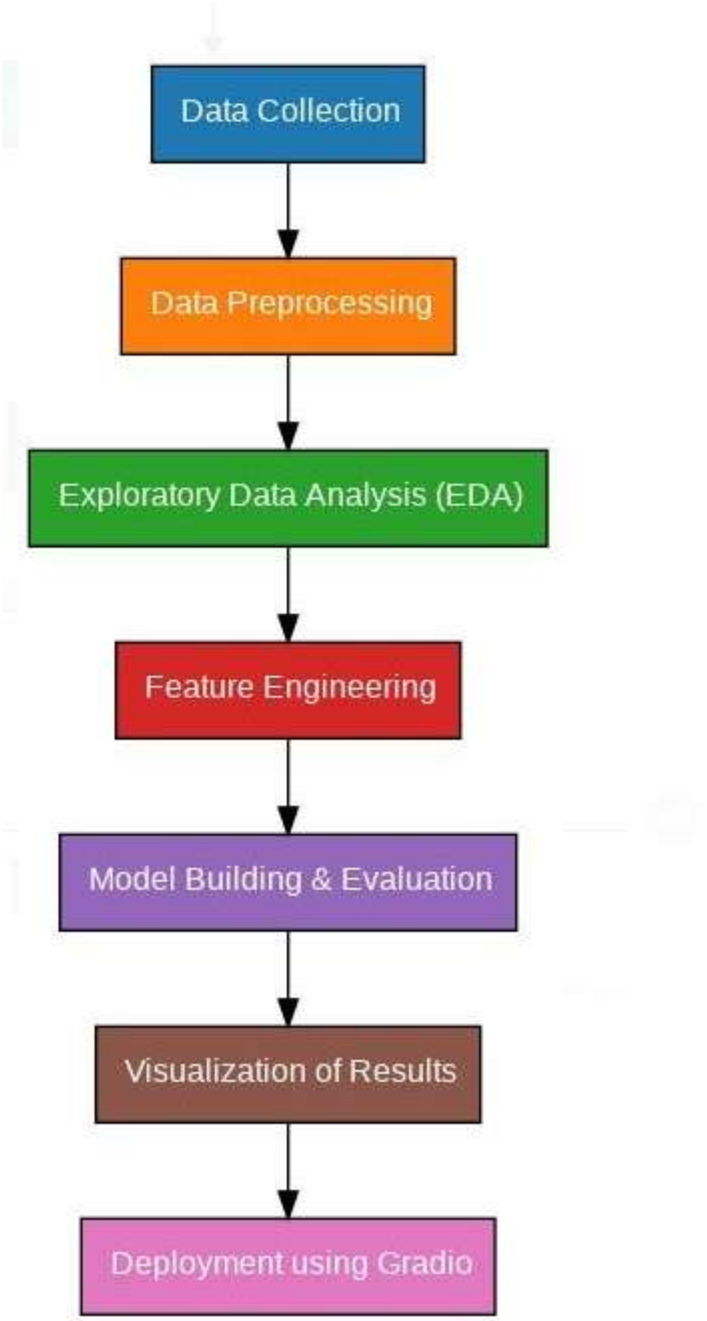
- Enable real-time prediction and visualization of air quality.

- Ensure model interpretability and real-world applicability.

- Deploy an interactive user interface using Gradio for testing and demonstration.
- The air is also highly contaminated which causes a more serious threat to all kinds of living organisms in the earth.
- This gives rise to the need for monitoring and assessing
- thequality of air and accordingly the government should be given alert to take necessary actions.
- This research work concentrates on performing an effective analysis on all
- The major works done in this aspect using machine learning algorithms.

3. Flowchart of the Project Workflow

Insert a flowchart showing steps:



Steps involved in flowchart:

1. Data Collection

- Gathering raw data from various sources like databases, APIs, sensors, or web scraping.
- This data is the foundation of the entire project.

2. Data Preprocessing

- Cleaning and formatting the collected data.
- Tasks include handling missing values, removing duplicates, converting data types, and normalization.
- Goal: Make the data suitable for analysis.

3. Exploratory Data Analysis (EDA)

- Understanding the data using statistics and visualizations.

- Identify patterns, correlations, outliers, and distribution.

- Helps in deciding which features are important.

4. Feature Engineering

- Creating new features or modifying existing ones to improve model performance.

- Includes encoding categorical variables, scaling, and deriving new metrics.

5. Model Building & Evaluation

- Selecting and training machine learning models using the processed data.

- Evaluating models using metrics like accuracy, precision, recall, etc.

6. Visualization of Results

- Displaying model performance and insights using charts, graphs, and dashboards.

- Makes the results interpretable for stakeholders.

7. Deployment using Gradio

- Final model is deployed using Gradio, a tool for building interactive web interfaces for ML models.

- Allows users to test and interact with the model in real-time.

4. Data Description

- **Dataset Name**: Air Quality Dataset
- **Source**: OpenAQ, UCI Repository, or Indian Government CPCB Portal
- **Type of Data**: Structured tabular data
- **Records and Features**: ~30,000 records, 15-25 features
- **Target Variable**: AQI (numeric or categorical)
- **Static or Dynamic**: Static (can be adapted to real-time feeds)
- **Attributes Covered**: PM2.5, PM10, NO2, SO2, CO, O3, temperature, humidity, wind speed, date, time, location

5. Data Preprocessing

- Checked and handled missing/null values
- Converted date/time to datetime format and extracted temporal features
- Normalized continuous variables using StandardScaler
- One-hot encoded categorical variables like location or weather condition •Detected and removed outliers using IQR and z-score methods

6. Exploratory Data Analysis (EDA)

Univariate Analysis:

- ○Histograms for pollutants and AQI
- ○Boxplots for identifying spread and outliers

Bivariate/Multivariate Analysis:

- ○Correlation matrix to check dependencies
- ○Line plots to show pollutant trends over time
- ○Heatmaps and scatter plots between pollutants and AQI

- **Key Insights**:
  - ○PM2.5 and PM10 have the strongest correlation with AQI
  - ○High pollution levels often coincide with low wind and high humidity

7. Feature Engineering

- Created average_pollution = (PM2.5 + PM10 + NO2)/3
- Derived time-based features like month, hour, weekday
- Encoded ordinal AQI categories if classification was used

- Removed multicollinear features to improve model performance

8. Model Building

- **Algorithms Used**:
  - ○Random Forest Regressor
  - ○XGBoost Classifier/Regressor
  - ○Support Vector Machine

- **Model Selection Rationale**:
  - Random Forest and XGBoost handle non-linearity and feature importance well
  - SVM used for scenarios with fewer features or binary classification
- **Train-Test Split**:
  - 80% training, 20% testing
- **Evaluation Metrics**:
  - **Regression**: MAE, RMSE, R² Score
  - **Classification**: Accuracy, Precision, Recall, F1 Score

9. Visualization of Results & Model Insights

- **Feature Importance**: Bar plots highlighting pollutant impact
- **Model Comparison**: Metric comparison across models
- **Residual and Confusion Plots**: Error distribution analysis
- **Interactive UI**: Gradio interface to simulate real-time AQI predictions

10. Tools and Technologies Used

- **Programming Language**: Python 3
- **Environment**: Google Colab / Jupyter Notebook

- **Libraries**:
  - pandas, numpy – data handling
  - matplotlib, seaborn, plotly – visualization
  - scikit-learn, xgboost – machine learning
  - Gradio – user interface

11. Team Members and Contributions

- Data Collection and Cleaning[Moniga.R]

- EDA and Visualization [Moniga.R]
- Feature Engineering [kousalya.A]
- Model Development [Mahalakshmi.P]
- UI and Deployment [Mamtha Sri.U]
- Documentation [Moniga.R]