Phase-2

Student Name: E Thirupathi

Register Number: 620123106119

Institution: AVS engineering college

Department: Electronic and communication engineering

Date of Submission: 08.05.2025

Github Repository Link:

https://github.com/harison26/Harison-naanmudhalvan.git

---

1. Problem Statement

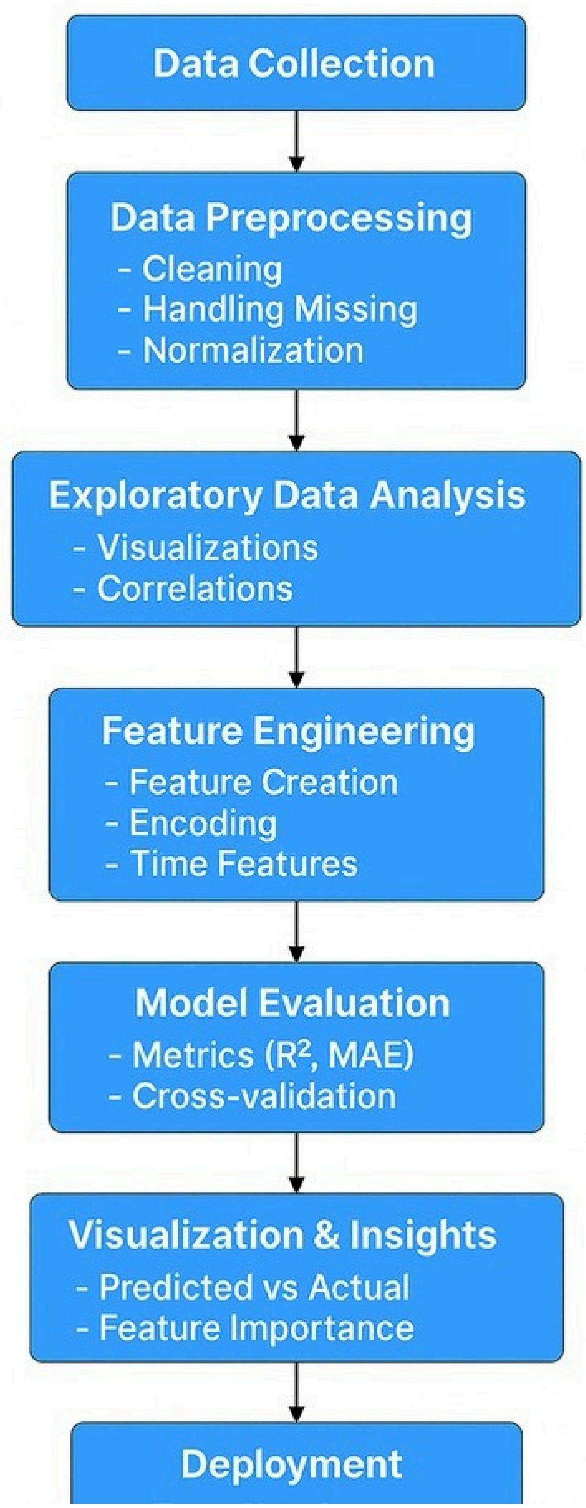
Increasing air pollution levels threaten public health and the environment.

Existing monitoring systems often lack predictive capabilities for proactive measures.
Need for a reliable machine learning model to predict air quality levels using historical sensor data.

2. Project Objectives

- Co llect and preprocess air quality data from relevant sources.
- Analyze and visualize air pollution patterns.
- Develop predictive models using machines learning algorithms.
- Evaluate and compare model performance.

3. Flowchart of the Project Workflow



4. Data Description

- SOURCE:
  OpenAQ, UCI ML Repository, or real-time sensors (DHT11, MQ-135,etc..).

FEATURES:

- Timestamp
- PM2.5,PM10
- CO,NO2,O3,SO2
- Tempreture , Humidity
- AQI (Target variable)

5. Data Preprocessing

- Handling missing/null values (imputation or removal)
- Data type conversion (e.g., data-time parsing)
- Outlier detection and treatment.
- Data normalization or standardization.

6. Exploratory Data Analysis (EDA)

- Distribution of pollutants
- Temporal trends in AQI
- Correlation matrix between pollutants and AQI
- AQI levels by region and time of day

7. Feature Engineering

- Extraction of date-time features (hour, day, month, weekday).
- Creating pollutant interaction terms.
- Encoding categorical features (e.g., location).
- Lag features for time-series modelling.

8. Model Building

- Train-Test split or TimeSeriesSplit

Alogorithms used:

- Linear Regressor
- Gradient Boosting (e.g.,XGBoost, LightGBM)
- LSTM (if time-series)
- Hyperparameter tuning with GridSearchCV or Optuna.

9. Visualization of Results & Model Insights

- Predicted vs Actual AQI plots.
- Residual plots and error distribution.
- Features important graph.
- SHAP values or LIME for model interpretability.

10. Tools and Technologies Used

- Programming Language: Python
- Libraries: Pandas, Numpy, Scikit-learn, Matplotlib, seaborn, XGBoost, LightGBM, SHAP.
- Visualization: Tableau, Power BI, Plotly.
- IDE/Notebook: Jupyter Notebook, VS Code.
- Version Control: GitHub

11. Team Members and Contributions

Saleth harison J - Project Manager

Defined problem scope, coordinated team, monitored progress.

Thirupathi E - Data Scientist

Led data preprocessing, EDA, feature engineering, and model training.

Mourish Kanna V - ML Engineer

Handled model selection, hyperparameter tuning, and optimization.

Sakthivel D - Visualization & Deployment

Designed visuals, created insights, and worked on front-end deployment.