Phase-2 Submission

Student Name: HemaKanitha S

Register Number:410723104025

Institution: Dhanalakshmi College of Engineering

Department: Computer Science and Engineering

Date of Submission: 08-05-2025

Github Repository Link:https://github.com/Hemakanitha1036/Nm_Hemakanitha

Predicting air quality levels using advanced Machine Learning algorithms for Environmental Insights.

1. Problem Statement

Develop a machine learning model to accurately forecast Air Quality Index (AQI) and pollutant concentrations, enabling early warnings, informed decision-making, and improved public health. The model will utilize historical data, environmental factors, and real-time inputs to predict air quality levels, supporting data-driven policy development and citizen awareness. Develop a robust and accurate air quality prediction model using advanced machine learning techniques to forecast the Air Quality Index (AQI) and concentrations of key pollutants (e.g., PM2.5, PM10, NO2, O3) in a given region.

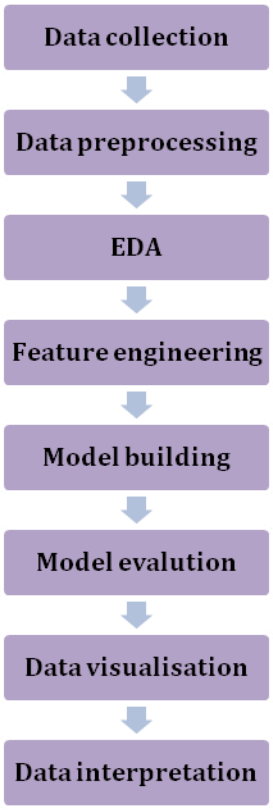
2. Project Objectives

The objective of this project is to design and train a machine learning model that can accurately predict air quality levels, enabling:

- **Early warnings**: Providing timely alerts to citizens and policymakers about potential air quality hazards.

- **Informed decision-making:** Supporting data-driven decision-making for air quality management and policy development.
- **Improved public health:** Reducing the adverse health effects of air pollution by enabling proactive measures to mitigate exposure.
- **Improve prediction accuracy:** Develop a model that accurately forecasts AQI and pollutant concentrations, reducing errors and uncertainties.
- **Identify key factors:** Determine the most influential environmental and anthropogenic factors affecting air quality, enabling targeted interventions.
- **Provide real-time alerts:** Design a system that sends timely notifications to citizens and policymakers about potential air quality hazards, enabling proactive measures.
- **Support policy development:** Offer data-driven insights to inform air quality management policies, regulations, and initiatives.
- **Enhance public awareness:** Develop a user-friendly platform to disseminate air quality information, promoting citizen awareness and engagement.
- **Evaluate model performance:** Continuously assess and refine the model's performance using various metrics, ensuring its reliability and effectiveness.

3. Flowchart of the Project Workflow



4. Data Description

- **Dataset Name**: Air quality prediction
- **Source**:Kaggle
- **Type of Data**: Structured tabular data
- **Number of records**:4000
- **Number of features**:23
- **Target Variable**: Date and Time
- **Static or Dynamic**: Static dataset
- **Dataset Link**: https://www.kaggle.com/datasets/khushikyad001/air-quality-data/data

5. Data Preprocessing

- Verified dataset integrity: no missing or null values.
- Removed irrelevant features with very low variance (e.g., school if only one value).

- Checked and confirmed absence of duplicate rows.
- Categorical features were one-hot encoded for machine learning.
- Applied **StandardScaler** to numerical columns to normalize them.
- Detected outliers using boxplots and z-scores; extreme outliers were investigated.

- Code for Missing Values: There is no missing values.

```
for col in df.columns:
    if df[col].dtype == 'object':
        df[col].fillna(df[col].mode()[0], inplace=True)
    else:
     df[col].fillna(df[col].mean(), inplace=True)
```

6. Exploratory Data Analysis (EDA)

Data Inspection

- Understand the structure, completeness, and types of data.
- Missing Value Heatmap (e.g., using Seaborn'sheatmap)
- Highlights missing data across features.
- Useful for detecting sensors that frequently fail.
- Data Type Count Plot
- Bar chart showing count of numeric vs categorical features.
- Helps assess data quality.
- Identifies features that need imputation or removal.

Time Series Plot

- Analyze trends and seasonality over time.
- Line plots of pollutant levels (e.g., PM2.5, CO) over days/months/years.
- Detect daily/weekly/seasonal trends.

Distribution Analysis

- Understand the distribution (spread and shape) of each feature.
- Histograms or KDE plots (Kernel Density Estimation)
- Show skewness (e.g., PM2.5 may be right-skewed).
- Help determine if transformations (e.g., log scale) are needed.
- Helps select appropriate models (some require normally distributed inputs).
- Detects outliers and anomalies.

Correlation Matrix

- Measure how variables relate to one another.
- Heatmap of Pearson Correlation Coefficients
- High correlation between pollutants (e.g., PM10 and PM2.5) suggests possible multicollinearity.
- Helps identify redundant features.
- Useful for feature selection or dimensionality reduction.

7. Feature Engineering

- Create new features to improve model accuracy.
- Bar plots or box plots of newly created features (e.g., pollutant ratios like PM2.5/PM10).
- plot pollution level by hour of day, day of week.
- Helps see if certain times are more polluted.
- Combined weighted score of multiple pollutants.
- e.g., Temperature × Humidity.
- Yesterday's PM2.5 to predict today's.
- Enhances the model's ability to detect patterns.

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

df['category_encoded'] = le.fit_transform(df['category'])
```

8. Model Building

Air quality prediction is a critical task that involves forecasting the concentration of pollutants in the air, such as particulate matter (PM), nitrogen dioxide(NO2),ozone (O3), and others. Select and implement at least 2 Machine learning. E.g., Logistic Regression, Decision Tree, Random Forest, KNN, etc.

- Justify why these models were selected (based on problem type and data).
- Split data into training and testing sets (with stratification if needed).
- Train models and evaluate initial performance using appropriate metrics.
    - For classification: accuracy, precision, recall, F1-score.
    - For regression: MAE, RMSE, R^2 score.
- **Data Split:**

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, stratify=y, random_state=42)
```

9. Visualization of Results & Model Insight

1. Histogram:

- Shows the distribution of a single variable .
- Histogram of PM2.5 concentration.
- The x-axis shows ranges of PM2.5 levels (e.g., 0–10, 10–20).

- The y-axis shows how many data points fall into each range.

2. Box Plot:

- Summarizes the spread and outliers in a variable.
- Box plot of NO2 levels across different cities or months.
- The central box shows the interquartile range (IQR) (middle 50% of the data).
- The line inside the box shows the median.

3. Pie Chart:

- Shows proportional data or category distribution.
- Pie chart of air quality categories .

4. Scatter Plot:

- Shows the relationship between two continuous variables.
- Each point represents a data observation.

10. Tools and Technologies Used

- **Programming Language**: Python 3
- **Notebook Environment**: Jupyter notebook
- **Key Libraries**: pandas, numpy for data handling.
  matplotlib, seaborn, plotly for visualizations.
  scikit-learn for preprocessing and modelling.

11. Team Members and Contributions

NAMES	ROLES	RESPONSIBILTY
Narmadha D	Member	Data collection and Prepocessing
Mohanapriya K	Member	EDA and Feature Engineering
Abinaya B	Leader	Model building and Evalution
Hemakanitha S	Member	Data Visualisation and Interpretation