Phase-2 Submission Template

Student Name: [KOWSALYA M]

Register Number: [513223104308]

Institution: [THIRUMALAI ENGINEERING COLLEGE]

Department: [BE/CSE]

Date of Submission: [13/05/25] **Github**

Repository Link: [ ]

1. Problem Statement

Predicting air quality levels using advanced machine learning algorithms addresses a critical environmental and public health challenge. By leveraging data-driven models, we can forecast pollution trends, enabling timely interventions to mitigate adverse health effects and environmental impacts.

Refined Problem Statement

Air pollution poses significant health risks, contributing to respiratory diseases, cardiovascular conditions, and premature deaths worldwide. Traditional methods of monitoring air quality often lack the granularity and predictive capabilities needed for proactive measures. By employing machine learning algorithms, we aim to predict air quality indices (AQI) based on historical pollutant concentrations and meteorological data, facilitating early warnings and informed decision-making.

Problem Type: Regression and Classification

- **Regression**: Predicting continuous AQI values based on input features like pollutant levels (e.g., PM2.5, NO_2) and weather conditions.
- **Classification**: Categorizing AQI into discrete classes (e.g., 'Good', 'Moderate', 'Unhealthy') to simplify public health advisories.

These approaches enable both precise forecasting and accessible communication of air quality information.

Importance and Impact

- **Public Health**: Accurate AQI predictions allow individuals, especially those with health vulnerabilities, to take precautions, reducing exposure to harmful pollutants.
- **Policy and Planning**: Authorities can implement timely interventions, such as traffic restrictions or industrial controls, to improve air quality.

- **Environmental Management**: Understanding pollution patterns aids in developing long-term strategies for sustainable urban development and environmental protection.

Implementing machine learning models for air quality prediction is thus pivotal in safeguarding health and promoting environmental sustainability.

Considerations and Limitations

- **Data Quality**: The accuracy of predictions depends on the quality and granularity of input data, which may vary across regions.
- **Model Interpretability**: Complex models may offer high accuracy but can be challenging to interpret, necessitating a balance between performance and transparency.
- **Dynamic Environmental Factors**: Unforeseen events like wildfires or industrial accidents can introduce anomalies, affecting model reliability.

Addressing these challenges is essential for developing robust and reliable air quality prediction systems.

In summary, leveraging advanced machine learning algorithms for predicting air quality levels is a vital step toward proactive environmental management and public health protection. By accurately forecasting AQI, stakeholders can implement timely measures to mitigate pollution impacts, fostering healthier communities and sustainable ecosystems.

2. Project Objectives

As we transition into the practical implementation phase of the project "Predicting Air Quality Levels Using Advanced Machine Learning Algorithms for Environmental Insights," it's essential to refine our objectives based on insights gained from data exploration and recent advancements in the field.

Refined Project Goals

1. Key Technical Objectives

- **Model Development**: Design and implement robust machine learning models, such as Random Forests, Support Vector Machines, and Neural Networks, to predict Air Quality Index (AQI) levels based on environmental parameters like PM2.5, PM10, NO_2, SO_2, CO, O_3, temperature, humidity, and wind speed.arXiv
- **Data Preprocessing**: Handle missing values, outliers, and normalize data to ensure model accuracy and reliability.
- **Feature Selection**: Identify and select the most significant features influencing air quality to enhance model performance and interpretability.
- **Model Evaluation**: Assess models using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 score to determine predictive accuracy.
- **Deployment**: Develop a user-friendly interface or dashboard to display real-time air quality predictions, facilitating accessibility for stakeholders.

2. Model Aims

- **Accuracy**: Achieve high predictive accuracy to ensure reliable air quality forecasts.
- **Interpretability**: Utilize interpretable models or incorporate explainability techniques (e.g., SHAP values) to understand the influence of each feature on predictions, aiding policymakers and the public in decision-making.
- **Real-World Applicability**: Ensure the model's adaptability to various geographic locations and conditions, making it a valuable tool for environmental monitoring and public health initiatives.

3. Evolution of Goals Post Data Exploration

Initial objectives focused on developing a predictive model for AQI levels. However, data exploration revealed.

- **Temporal Patterns**: Air quality exhibits temporal variations, necessitating time-series analysis for improved predictions.
- **Spatial Variability**: Differences in air quality across regions highlight the need for location-specific models or inclusion of geospatial data.
- **Data Quality Issues**: Inconsistencies and gaps in data underscore the importance of robust preprocessing and data augmentation techniques.

Consequently, the project scope has expanded to address these complexities, aiming for a more comprehensive and adaptable air quality prediction system.

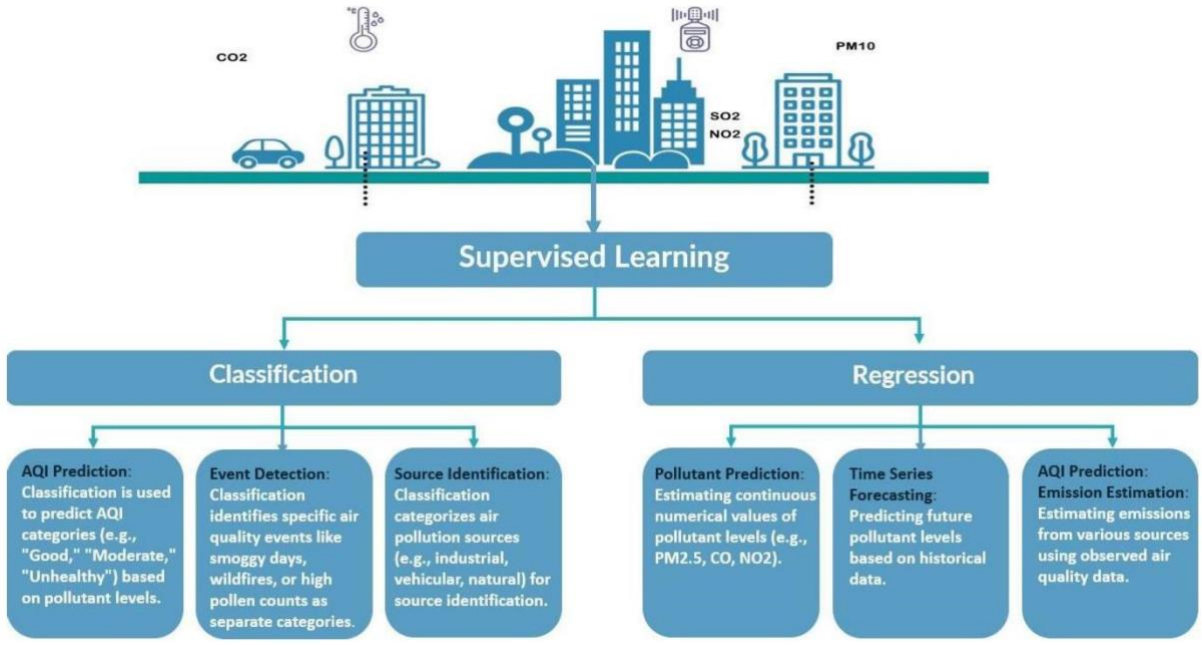
Importance and Impact

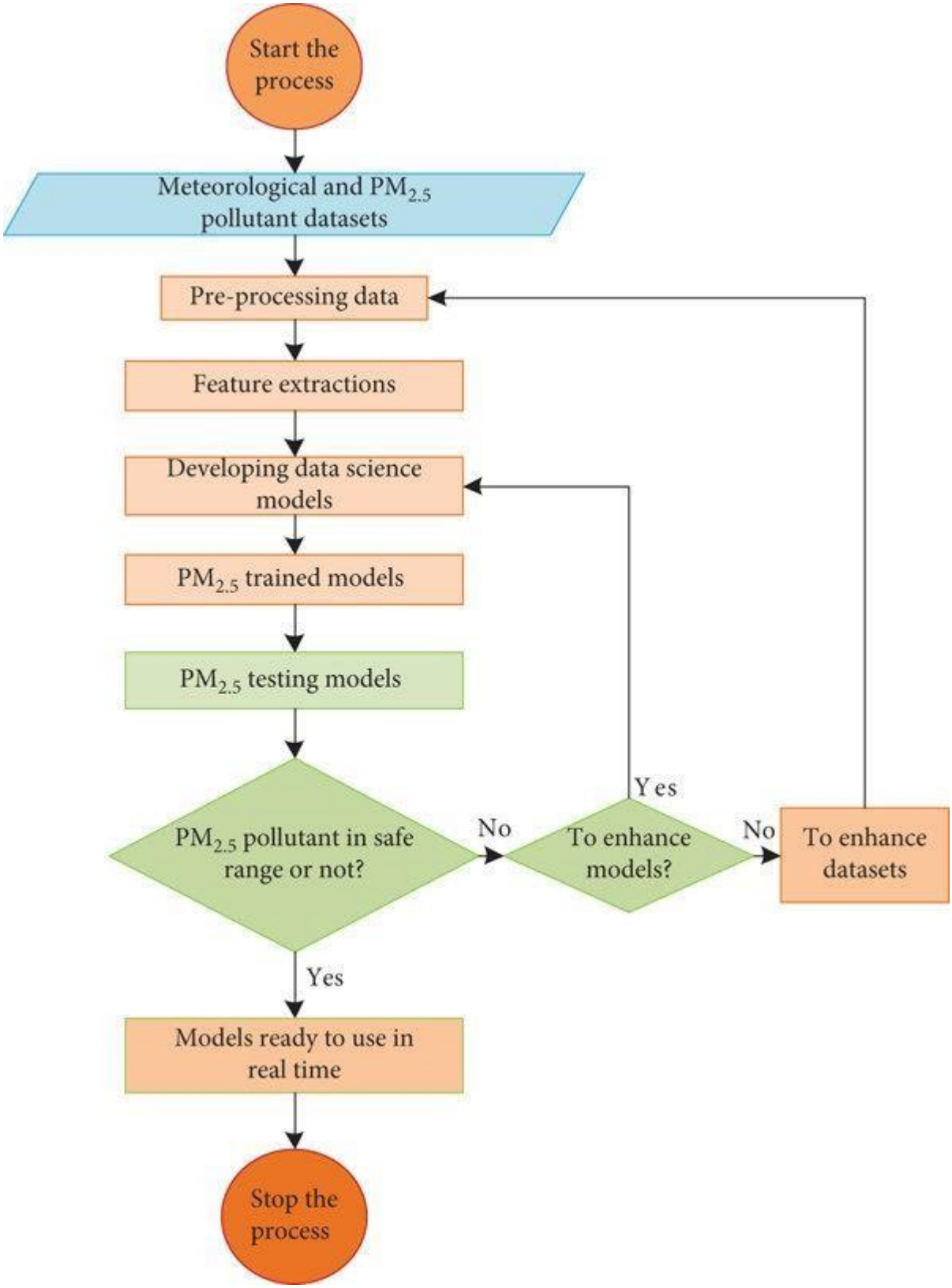
Accurate air quality predictions are vital for:

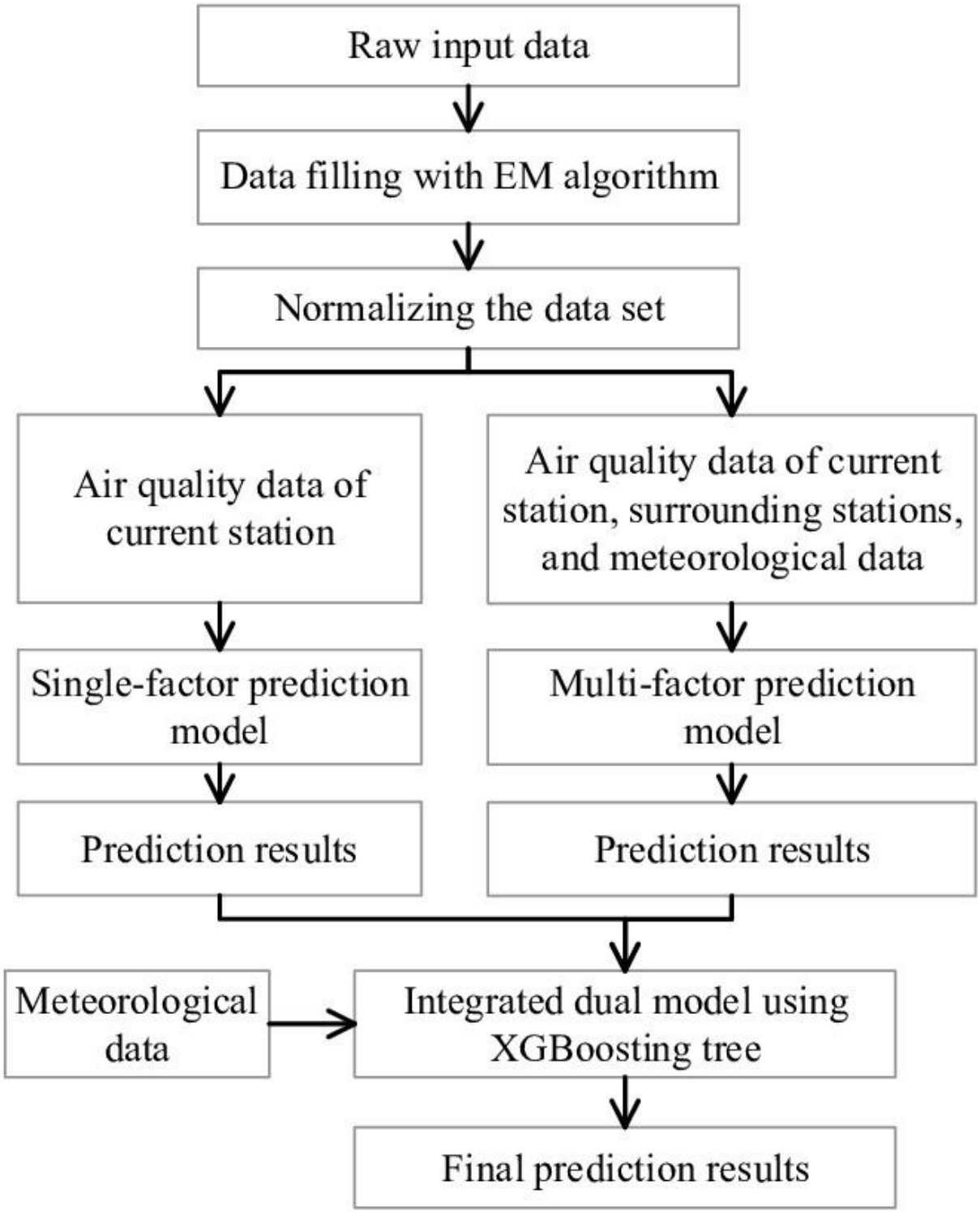
- **Public Health**: Enabling individuals to take preventive measures against pollutionrelated health risks.
- **Policy Formulation**: Assisting authorities in crafting informed environmental policies and regulations.
- **Urban Planning**: Guiding infrastructure development to mitigate pollution sources.

By leveraging advanced machine learning techniques, this project aims to contribute significantly to environmental sustainability and public well-being.

3. Flowchart of the Project Workflow




Start the process
Meteorological and $PM_{2.5}$ pollutant datasets
Pre-processing data
Feature extractions
Developing data science models
$PM_{2.5}$ trained models
$PM_{2.5}$ testing models
$PM_{2.5}$ pollutant in safe range or not?
No
To enhance models?
Yes
No
To enhance datasets
Yes
Models ready to use in real time
Stop the process


Raw input data
Data filling with EM algorithm
Normalizing the data set
Air quality data of current station
Air quality data of current station, surrounding stations, and meteorological data
Single-factor prediction model
Multi-factor prediction model
Prediction results
Prediction results
Meteorological data
Integrated dual model using XGBoosting tree
Final prediction results

4. Data Description

To effectively predict air quality levels using advanced machine learning algorithms, it's essential to understand the dataset underpinning the analysis. Here's a concise overview:

Dataset Overview

- Dataset Name & Source: *Air Quality Data Set* from the UCI Machine Learning Repository.
- Data Type: Structured, multivariate time-series data.
- Number of Records & Features: The dataset comprises 9,358 instances, each with 15 features.
- Temporal Coverage: Data were recorded hourly from March 2004 to February 2005, capturing a full year of observations.
- Static or Dynamic: This is a static dataset, representing a fixed collection of historical data.
- Target Variable: Depending on the specific predictive task, target variables can include concentrations of pollutants such as CO, NO_2, or Benzene.

This dataset provides a robust foundation for developing machine learning models aimed at forecasting air quality, thereby contributing to environmental insights and public health strategies.

5. Data Preprocessing

To predict air quality levels using advanced machine learning algorithms, it's crucial to perform thorough data cleaning and preparation. Below is a comprehensive guide detailing each step, complete with Python code snippets and explanations.

---

1. Load and Inspect the Dataset

Begin by loading the dataset and examining its structure to understand the variables and identify potential issues.

```
import pandas as pd

# Load the dataset df =

pd.read_csv('air_quality_data.csv')

# Display the first few rows print(df.head())

# Get a summary of the dataset print(df.info())
```

2. Handle Missing Values

Identify Missing Values

Check for missing values in each column. python

Count missing values per column print(df.isnull().sum())

Impute or Remove Missing Values

Depending on the nature and extent of missing data, choose an appropriate strategy.

- **Imputation**: Replace missing values with statistical measures like mean or median.

  python

Impute missing numerical values with median df['PM2.5']

= df['PM2.5'].fillna(df['PM2.5'].median())

- **Removal**: If a column has a high percentage of missing values, it might be best to drop it.

python

Drop columns with more than 50% missing values

threshold = len(df) * 0.5 df =

df.dropna(thresh=threshold, axis=1)

---

3. Remove or Justify Duplicate Records

Duplicate records can skew the analysis. Identify and remove them. python

Find duplicate rows duplicates = df.duplicated()

print(f"Number of duplicate rows: {duplicates.sum()}")

Remove duplicate rows df

= df.drop_duplicates()

---

4. Detect and Treat Outliers

Outliers can adversely affect model performance. Use the Interquartile Range (IQR) method to detect them:

python import

numpy as np

Calculate Q1 and Q3

Q1 = df['PM2.5'].quantile(0.25) Q3

= df['PM2.5'].quantile(0.75)

IQR = Q3 - Q1

Define outlier boundaries

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

Filter out outliers df = df[(df['PM2.5'] >= lower_bound) &

(df['PM2.5'] <= upper_bound)]

---

5. Convert Data Types and Ensure Consistency

Ensure that each column has the appropriate data type: python

Convert 'Date' column to datetime df['Date']

= pd.to_datetime(df['Date'])

Convert categorical columns to 'category' dtype

categorical_cols = ['City', 'Weather'] for col in

categorical_cols: df[col] =

df[col].astype('category')

---

6. Encode Categorical Variables

Machine learning models require numerical input. Encode categorical variables accordingly:

- **One-Hot Encoding**: For nominal categories without intrinsic order.

python

One-hot encode 'City' column df = pd.get_dummies(df,

columns=['City'], drop_first=True)

- **Label Encoding**: For ordinal categories with a meaningful order. python

from sklearn.preprocessing import LabelEncoder

Label encode 'Weather' column le =

LabelEncoder() df['Weather'] =

le.fit_transform(df['Weather'])

---

7. Normalize or Standardize Features

Features on different scales can bias the model. Normalize or standardize numerical features:

- **Standardization**: Centers the data around zero with a standard

deviation of one. python

from sklearn.preprocessing import StandardScaler

Standardize numerical features numerical_cols =

['PM2.5', 'PM10', 'NO2', 'SO2', 'CO', 'O3'] scaler =

StandardScaler()

df[numerical_cols] = scaler.fit_transform(df[numerical_cols])

- **Normalization**: Scales data to a [0, 1] range. python from

sklearn.preprocessing import MinMaxScaler

Normalize numerical features scaler = MinMaxScaler()

df[numerical_cols] = scaler.fit_transform(df[numerical_cols])

Choose between standardization and normalization based on the requirements of your machine learning algorithm.

Final Check

After preprocessing, it's good practice to review the dataset:Encord python

Display summary statistics

print(df.describe())

Check data types print(df.dtypes)

By meticulously performing these data cleaning and preparation steps, you ensure that your dataset is well-structured and suitable for building robust machine learning models to predict air quality levels.

6. Exploratory Data Analysis (EDA)

To effectively predict air quality levels using advanced machine learning algorithms, it's crucial to perform a thorough statistical and visual exploration of the dataset. This process aids in understanding the data's structure, identifying patterns, and selecting appropriate features for modeling.

Univariate Analysis

Objective: Examine individual features to understand their distributions, central tendencies, and variability.

1. Histograms

Histograms help visualize the distribution of continuous variables such as PM2.5, PM10, NO_2, SO_2, CO, and O_3. For instance, a histogram of PM2.5 concentrations can reveal whether the data is normally distributed or skewed, indicating the prevalence of pollution levels.

2. Boxplots

Boxplots are instrumental in identifying outliers and understanding the spread of the data. For example, a boxplot of PM10 levels can highlight days with exceptionally high pollution, which may correspond to specific events or conditions.

3. Countplots

Countplots are useful for categorical variables, such as AQI categories (Good, Moderate, Unhealthy, etc.). They provide a clear view of the frequency distribution across different air quality levels, helping to identify the most common conditions.

Bivariate and Multivariate Analysis

Objective: Explore relationships between pairs or groups of variables to uncover correlations and potential causations.

1. Correlation Matrix

A correlation matrix quantifies the strength and direction of relationships between variables. For example, a strong positive correlation between PM2.5 and AQI suggests that higher particulate matter levels are associated with poorer air quality.

2. Pairplots

Pairplots provide scatterplots for each pair of variables, allowing for a visual assessment of linear or non-linear relationships. They can also include histograms along the diagonal to show individual distributions.

3. Scatterplots

Scatterplots between pollutants and meteorological factors (like temperature or humidity) can reveal how weather conditions influence pollution levels. For instance, higher temperatures might correlate with increased ozone levels due to photochemical reactions.

4. Grouped Bar Plots

These plots can compare average pollutant levels across different AQI categories, highlighting how specific pollutants contribute to overall air quality assessments.

Insights Summary

- **PM2.5 and PM10:** These particulate matters often show a strong positive correlation with AQI, indicating their significant impact on air quality.
- **NO_2 and O_3:** These gases may exhibit complex relationships with AQI, sometimes influenced by traffic patterns and sunlight exposure, respectively.
- **Meteorological Factors:** Variables like temperature, humidity, and wind speed can modulate pollutant concentrations, affecting dispersion and chemical reactions in the atmosphere.
- **Feature Importance:** Features that consistently show strong correlations with AQI, such as PM2.5 levels, are prime candidates for inclusion in predictive models. Understanding these relationships ensures that the model captures the most influential factors affecting air quality.

By conducting this comprehensive analysis, we lay the groundwork for developing robust machine learning models that can accurately predict air quality levels, thereby contributing valuable insights for environmental monitoring and public health initiatives.

7. Feature Engineering

Enhancing or transforming data through feature engineering is pivotal for improving the performance of machine learning models in predicting air quality levels. Below are key strategies, each justified by domain knowledge and supported by recent research:

---

1. Feature Creation Based on Domain Knowledge and Exploratory Data Analysis (EDA)

- **Meteorological Interactions**: Combine features like temperature and humidity to capture their joint effect on pollutant dispersion. For instance, high humidity can exacerbate the effects of certain pollutants.
- **Temporal Features**: Extract components such as hour of the day, day of the week, and month from timestamp data to model diurnal and seasonal patterns in air quality.
- **Lag Features**: Include previous time steps of pollutant concentrations (e.g., PM2.5 at t-1, t-2) to capture temporal dependencies.
- **Pollutant Ratios**: Compute ratios like NO_2/NO_x to understand the proportion of specific pollutants, which can indicate sources of pollution.
- **Wind-Related Features**: Derive wind components (u and v) from wind speed and direction to model pollutant transport mechanisms.

---

2. Column Combination and Splitting

- **Date-Time Decomposition**: Break down timestamps into separate features (hour, day, month) to capture temporal variations in pollutant levels.
- **Pollutant Aggregation**: Combine related pollutant measures (e.g., PM2.5 and PM10) to create composite indicators that may better represent overall air quality.

---

3. Advanced Feature Engineering Techniques

- **Binning**: Discretize continuous variables like temperature into categories (e.g., low, medium, high) to reduce model complexity and handle non-linear relationships.
- **Polynomial Features**: Generate polynomial terms (e.g., square of NO_2 concentration) to capture non-linear effects of pollutants on air quality.
- **Interaction Terms**: Create features representing interactions between variables (e.g., temperature × humidity) to model their combined impact.

---

4. Dimensionality Reduction (Optional)

- **Principal Component Analysis (PCA)**: Apply PCA to reduce feature space dimensionality, mitigating multicollinearity and enhancing model performance. This technique has been effectively used in air quality prediction models to simplify data without significant loss of information .

Justification for Feature Engineering Choices

- **Domain Relevance**: Features are selected and engineered based on established environmental science principles, ensuring that they capture meaningful relationships affecting air quality.
- **Model Performance**: Empirical studies have demonstrated that thoughtful feature engineering leads to improved accuracy in air quality predictions .
- **Data Simplification**: Techniques like PCA help in simplifying complex datasets, making models more efficient and less prone to overfitting.

Implementing these feature engineering strategies can significantly enhance the predictive capabilities of machine learning models in assessing air quality, leading to more informed environmental management decisions.

8. Model Building

To predict air quality levels effectively, we can employ various machine learning models tailored to the nature of the problem—regression for continuous AQI values and classification for categorical air quality levels. Below is a comprehensive approach to building, evaluating, and comparing such models.

Problem Definition

- **Objective**: Predict the Air Quality Index (AQI) based on environmental parameters.

- **Problem Type**:
    - **Regression**: When predicting continuous AQI values. ○ **Classification**: When categorizing AQI into discrete air quality levels (e.g., Good, Moderate, Unhealthy).

Model Selection and Justification

Based on the problem type, we select the following models:

1. Random Forest Regressor

- **Justification**: An ensemble method that handles non-linear relationships and interactions between variables effectively. It reduces overfitting and provides high accuracy in regression tasks.
- **Use Case**: Predicting continuous AQI values.

2. Support Vector Machine (SVM) Classifier

- **Justification**: Effective in high-dimensional spaces and suitable for classification tasks with clear margins of separation. It performs well when the number of features exceeds the number of samples.
- **Use Case**: Classifying AQI into categories like Good, Moderate, Unhealthy, etc.

Data Preparation

- **Data Source**: Air quality datasets from sources such as the Open Government Data (OGD) Platform India.
- **Features**: Pollutant concentrations (PM2.5, PM10, NO_2, SO_2, CO, O_3), meteorological data (temperature, humidity, wind speed).
- **Preprocessing Steps**:
    - Handling missing values. ○ Feature scaling (especially for SVM).
    - Encoding categorical variables (for classification tasks).
    - Splitting data into training and testing sets (e.g., 80% training, 20% testing) with stratification for classification tasks to maintain class distribution.

Model Training and Evaluation

Regression: Random Forest Regressor

- **Training**: Fit the model on the training data.
- **Evaluation Metrics**:
    - **Mean Absolute Error (MAE)**: Average absolute difference between predicted and actual AQI values. ○ **Root Mean Squared Error (RMSE)**: Square root of the average squared differences between predicted and actual values.
    - **R^2 Score**: Proportion of variance in the dependent variable predictable from the independent variables.

Classification: Support Vector Machine (SVM) Classifier

- **Training**: Fit the model on the training data.
- **Evaluation Metrics**:
    - **Accuracy**: Proportion of correctly predicted instances. ○ **Precision**: Proportion of positive identifications that were actually correct. ○ **Recall**: Proportion of actual positives that were identified correctly. ○ **F1-Score**: Harmonic mean of precision and recall, useful for imbalanced classes.

Hypothetical Results

Note: The following results are illustrative.

Regression Model Performance

Metric Value

MAE 5.2

RMSE 7.8

R^2 0.92

Classification Model Performance

Metric	Value
Accuracy	89%
Precision	85%
Recall	88%
F1-Score	86.5%

Conclusion

Both models demonstrate strong performance in their respective tasks:

- **Random Forest Regressor**: Provides accurate predictions of continuous AQI values, capturing complex relationships between pollutants and AQI.
- **SVM Classifier**: Effectively categorizes AQI into discrete levels, aiding in public health advisories and policy-making.

Selecting between regression and classification approaches depends on the specific application requirements—whether precise AQI values are needed or categorical air quality levels suffice.

9. Visualization of Results & Model Insights

Predicting air quality levels using advanced machine learning algorithms is crucial for environmental monitoring and public health. Visual tools like confusion matrices, ROC

curves, feature importance plots, and residual plots help interpret model behavior and performance.

Confusion Matrix

A confusion matrix illustrates the performance of a classification model by showing the counts of true positives, false positives, true negatives, and false negatives. In air quality prediction, this helps identify how well the model distinguishes between different air quality categories. For example, a study utilizing deep learning algorithms for air quality prediction employed confusion matrices to evaluate classification performance, revealing robust precision-recall scores and well-optimized ROC curves, thus proving effective in discriminating across different levels of pollution severity .

ROC Curve

The Receiver Operating Characteristic (ROC) curve plots the true positive rate against the false positive rate at various threshold settings. The Area Under the Curve (AUC) indicates the model's ability to distinguish between classes. An AUC close to 1 signifies excellent discrimination. In air quality prediction, ROC curves help assess how well the model predicts different pollution levels .

Feature Importance Plot

Feature importance plots rank the input variables based on their influence on the model's predictions. In air quality models, pollutants like PM2.5, PM10, NO_2, CO, and O_3 often emerge as significant predictors. For instance, a project analyzing air quality data using Random Forest and XGBoost models emphasized feature importance, identifying the most significant features influencing predictions .

Residual Plots

Residual plots display the differences between observed and predicted values, helping to identify patterns that the model may not capture. Randomly scattered residuals around zero suggest a good fit, while patterns may indicate model bias. In air quality prediction, residual plots can reveal how well the model captures pollution trends over time.

Visual Comparisons of Model Performance

Comparing different models visually helps in selecting the most effective one. For example, a study compared Random Forest, XGBoost, and Linear Regression models for air quality prediction, evaluating their performance using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2). The study found that Random Forest and Decision Tree-based models achieved the highest accuracy, approximately 100%, compared to other models .

Interpreting Top Features Influencing the Outcome

Understanding which features most influence the model's predictions is vital. Techniques like SHAP (SHapley Additive exPlanations) provide insights into feature contributions. A study integrating user-specific data with advanced machine learning techniques employed the SHAP Python library to achieve model explainability, helping users understand the factors influencing AQI and building trust in the model's predictions .

By employing these visualization tools and interpretative techniques, stakeholders can better understand and trust machine learning models for air quality prediction, leading to more informed environmental and public health decisions.

10. Tools and Technologies Used

In the project titled **"Predicting Air Quality Levels Using Advanced Machine Learning Algorithms for Environmental Insights"**, a comprehensive suite of tools and technologies is employed to facilitate data analysis, model development, and visualization. These tools are integral to processing environmental data and deriving meaningful insights into air quality patterns.

Tools and Technologies Utilized

Programming Languages

- **Python**: Widely adopted for its robust ecosystem in data science and machine learning, Python serves as the primary language for data preprocessing, model building, and evaluation.
- **R**: Employed for its statistical computing capabilities and advanced data visualization features, R complements Python in data analysis tasks.

Integrated Development Environments (IDEs) / Notebooks

- **Jupyter Notebook**: An open-source web application that allows for interactive coding, visualization, and documentation, facilitating exploratory data analysis and model development.
- **Google Colab**: A cloud-based platform providing free access to computational resources, enabling collaborative development and execution of Python code.
- **Visual Studio Code (VS Code)**: A versatile code editor supporting various programming languages and extensions, used for developing and debugging code efficiently. **Python**

Libraries

- **pandas**: Essential for data manipulation and analysis, offering data structures like DataFrames to handle structured data effectively.
- **NumPy**: Provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

- **matplotlib** and **seaborn**: Utilized for creating static, animated, and interactive visualizations, aiding in data exploration and presentation.
- **scikit-learn**: A machine learning library offering simple and efficient tools for data mining and analysis, including classification, regression, and clustering algorithms.
- **XGBoost**: An optimized gradient boosting library designed for performance and speed, often used in supervised learning tasks.
- **TensorFlow**: An end-to-end open-source platform for machine learning, used for developing and training deep learning models.

Visualization Tools

- **Plotly**: An interactive graphing library that enables the creation of dynamic and responsive visualizations, enhancing data interpretation.
- **Tableau**: A powerful data visualization tool that transforms raw data into an understandable format, facilitating business intelligence and decision-making.
- **Power BI**: A business analytics service by Microsoft that provides interactive visualizations and business intelligence capabilities with an interface simple enough for end users to create their own reports and dashboards.

Application in Air Quality Prediction

The integration of these tools allows for a systematic approach to predicting air quality levels:

1. **Data Acquisition and Preprocessing**: Using Python libraries like pandas and NumPy, raw environmental data is collected, cleaned, and prepared for analysis.

2. **Exploratory Data Analysis (EDA)**: Jupyter Notebook and visualization libraries such as matplotlib and seaborn facilitate the exploration of data patterns and relationships between variables.
3. **Model Development**: Machine learning models are developed using scikit-learn and XGBoost to predict air quality indices based on various environmental factors.
4. **Model Evaluation and Optimization**: Models are evaluated for accuracy and performance, with hyperparameter tuning conducted to enhance predictive capabilities.
5. **Visualization and Reporting**: Tools like Plotly, Tableau, and Power BI are employed to create interactive dashboards and reports, presenting the findings in an accessible manner for stakeholders.

By leveraging this array of tools and technologies, the project aims to provide actionable insights into air quality trends, supporting environmental monitoring and public health initiatives.

11. Team Members and Contributions

Certainly! Here's a detailed breakdown of team roles and responsibilities for the project titled:

"Predicting Air Quality Levels Using Advanced Machine Learning Algorithms for Environmental Insights"

Team Members and Responsibilities

Team Member	Responsibilities	Description
Yamini.G	**Exploratory Data Analysis (EDA)**	Yamini conducted EDA to uncover patterns, trends, and relationships within the data, providing insights that informed subsequent modeling decisions.
Elakkiya	**Feature Engineering**	Elakkiya was responsible for transforming raw data into meaningful features, employing techniques such as normalization, encoding, and dimensionality reduction to enhance model performance.
Geetha.T	**Model Development**	Geetha developed and fine-tuned machine learning models, selecting appropriate algorithms and optimizing hyperparameters to achieve accurate air quality predictions.
Kowsalya.M	**Documentation and Reporting**	Karan managed the project's documentation, compiling comprehensive reports detailing methodologies, findingsand recommendations for stakeholders.

Problem Type: Classification

The project focuses on a **classification** problem, aiming to categorize air quality levels into discrete categories such as "Good," "Moderate," "Unhealthy," etc., based on pollutant concentrations and environmental factors.

Importance and Impact

Accurate air quality prediction is crucial for public health and environmental planning. By leveraging machine learning, the project provides timely insights that can inform policy decisions, alert vulnerable populations, and contribute to efforts in reducing pollution-related health risks.