### Code Documentation for Project Report

#### **1. Data Collection and Preprocessing**

* **Purpose**: This step ensures the raw data is cleaned and prepared for further analysis. It involves handling missing values, standardizing data through normalization, and making the dataset suitable for input into machine learning models.
* **Implementation**:
  + Missing values are handled using statistical methods, such as filling them with the median or mean.
  + Normalization ensures that all features are on the same scale, reducing potential bias in the models.

#### **2. Feature Engineering**

* **Purpose**: Feature engineering enhances the dataset by adding new, meaningful features that improve model performance. It involves creating features like rolling averages or lagged values to capture trends and temporal dependencies in the data.
* **Implementation**:
  + Rolling averages smooth out data fluctuations to highlight trends over time.
  + Lagged features include information from previous time steps, helping the model predict future values based on historical trends.

#### **3. Model Training**

* **Purpose**: This phase involves building machine learning models such as Random Forest, SVM, and LSTM to predict air quality levels. Each model is trained on the preprocessed dataset to learn patterns and make predictions.
* **Implementation**:
  + Random Forest and SVM are trained using tabular data, leveraging libraries that support these algorithms.
  + LSTM is employed for time-series prediction, as it can capture sequential dependencies in the data.

#### **4. Model Evaluation**

* **Purpose**: The evaluation phase assesses the accuracy and reliability of the models. Metrics like Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R² are used to measure performance.
* **Implementation**:
  + RMSE evaluates the average magnitude of prediction errors.
  + R² assesses how well the model explains the variability in the data.
  + MAE provides a simple average of absolute errors, giving insight into the prediction accuracy.

#### **5. Visualization**

* **Purpose**: Visualizations are used to interpret data and communicate findings effectively. They showcase trends in the dataset and compare actual versus predicted values for model validation.
* **Implementation**:
  + Dataset summaries are visualized through time-series plots, showing pollutant trends.
  + Predicted versus actual comparisons validate model accuracy.

### How to Present This to Your Supervisor

1. **Highlight Contributions**:
   * Clearly specify your contributions, such as data preprocessing scripts, feature engineering, and model training procedures.
   * Mention the use of external libraries (e.g., Scikit-learn for machine learning or TensorFlow for deep learning).
2. **Add Detailed Comments**:
   * Explain the purpose of each section in the code. For instance, describe preprocessing as essential for data quality, or explain feature engineering as improving prediction accuracy.
   * Annotate specific functions, such as those for model training or evaluation, to clarify their role in the workflow.
3. **Organize Your Repository**:
   * Structure your repository logically with directories like src/ for scripts, data/ for datasets, and notebooks/ for exploratory work.
   * Ensure all scripts are well-commented, clean, and easy to follow for supervisors reviewing your work.

## **Methodology**

### ****1. Data Collection****

* **Source**: Data is collected from the London Air Quality Network (LAQN) and OpenAQ platforms, covering pollutant concentrations (e.g., PM2.5, PM10, and NO₂) and meteorological data (e.g., temperature, humidity, and wind speed).
* **Timeframe**: The dataset spans 2018–2023, capturing seasonal and temporal variations.

### ****2. Data Preprocessing****

* **Steps**:
  + Handle missing values to ensure data completeness.
  + Normalize features for consistency and better model performance.
  + Create new features, such as rolling averages and lagged variables, to capture temporal patterns and enhance predictions.

### ****3. Model Development****

* **Algorithms**:
  + Random Forest and Support Vector Machines (SVM) for baseline model performance.
  + Long Short-Term Memory (LSTM) for capturing sequential dependencies in time-series data.
* **Tools**:
  + Python libraries: Scikit-learn, TensorFlow/Keras.

### ****4. Model Evaluation****

* **Metrics**:
  + **Root Mean Square Error (RMSE)**: Measures the average magnitude of prediction errors.
  + **Mean Absolute Error (MAE)**: Quantifies the average absolute difference between predictions and actual values.
  + **R²**: Explains how much variance in the data is captured by the model.

### ****5. Visualization****

* **Purpose**: Visualizations are used to validate models and communicate insights.
* **Examples**:
  + Scatter plots for predicted vs. actual values.
  + Time-series graphs for pollutant trends.

### ****Repository Structure****

This repository includes scripts for data preprocessing, model training, evaluation, and visualizations, organized into appropriate directories for easy navigation.