**Architecture**

**AIR QUALITY**

**INDEX PREDICTION**

|  |  |
| --- | --- |
| **Written by** | **Gunaseelan V** |
| **Documented Version** | **1.0** |
| **Last Revised Date** | **18 December 2024** |

**Table of Contents**

1. **Architecture**
2. **Architecture Description**  
   2.1. **Data Description**  
   2.2. **Data Collection**  
   2.3. **Data Transformation**  
   2.4. **Data Pre-processing**  
   2.5. **Model Building**  
   2.6. **Data from User**  
   2.7. **Data Validation**  
   2.8. **Model Call for Prediction**  
   2.9. **Deployment**

### ****1. Architecture****

The architecture of the AQI prediction system is designed to efficiently process and predict the AQI values based on input pollutant data. It utilizes machine learning models such as Random Forest and Gradient Boosting to make predictions. The system incorporates data preprocessing, validation, and deployment via Streamlit for interactive user experience. The goal is to predict AQI based on input features like pollutant levels (Ozone, NO, PM10, CO).

### ****2. Architecture Description****

The system architecture consists of several key components working together:

**Data Input Layer**:

* + Users provide input data through the interface, including pollutant levels such as **Ozone (O3)**, **NO**, **PM10**, and **CO**.
  + The user inputs pollutant values and triggers the prediction by clicking the **“Predict”** button.

**Data Preprocessing Layer**:

* + **Validation**: Input data is validated to ensure correctness and consistency before being fed into the prediction models. This step includes checks for missing or out-of-range values.
  + **Transformation**: The input data is transformed (e.g., normalization or scaling) if necessary to match the format required by the prediction models.

**Modeling Layer**:

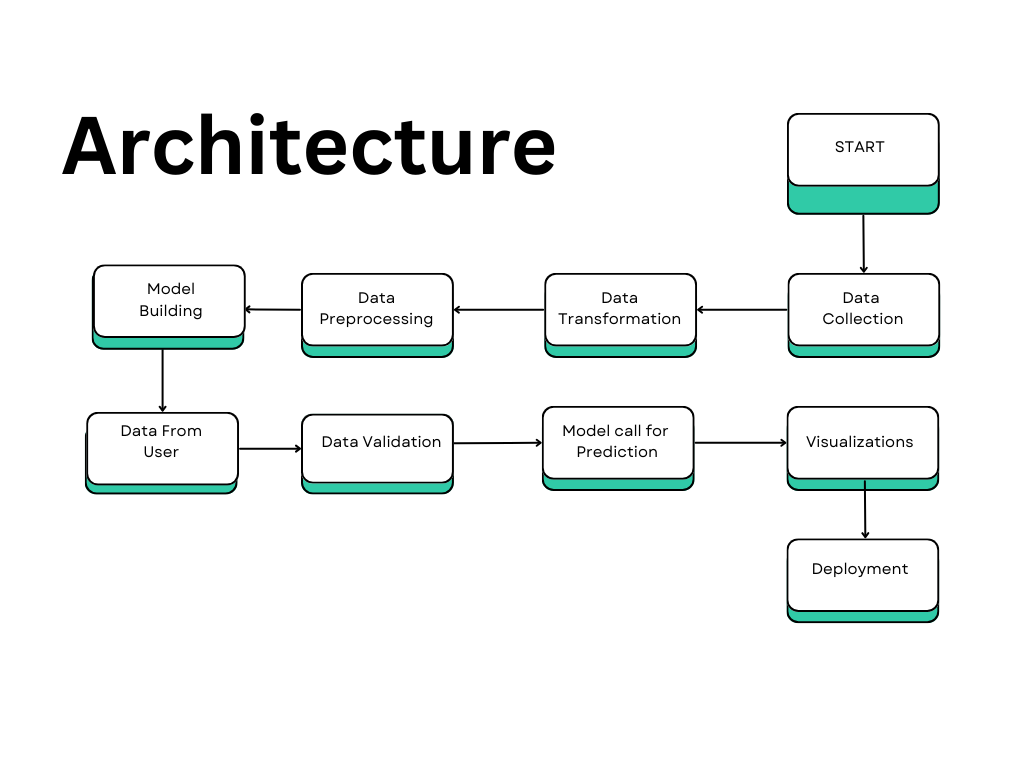
* + This layer includes trained machine learning models (**Random Forest** and **Gradient Boosting**) to predict AQI based on the input data. The models are trained on historical data and updated periodically to ensure accurate predictions.
  + The models process the pollutant levels and generate an AQI value.

**Output Layer**:

* + The predictions are displayed to the user, along with visualizations for further analysis. These visualizations include:
    - **Pollutant Levels vs AQI**: Graph showing the relationship between pollutant levels (Ozone, NO, PM10, CO) and the predicted AQI.
    - **Pollutant Contribution vs AQI Value**: Graph showing how each pollutant contributes to the final AQI value.

**Deployment Layer**:

* + The system is deployed on a cloud platform via **Streamlit**, making it accessible to users.
  + Streamlit provides a user-friendly interface where users can input data, trigger predictions, and view the results in real-time.



### ****2.1 Data Description****

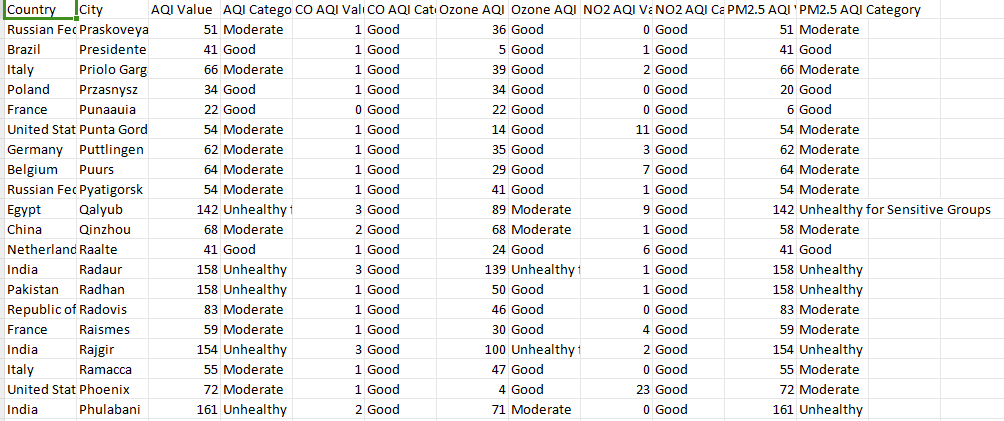
The dataset used for training the AQI prediction model includes historical pollutant data along with environmental information. Key features in the dataset include:

* **Ozone (O3)**: Concentration of Ozone in the air.
* **NO (Nitrogen Oxides)**: Concentration of Nitrogen Oxides in the air.
* **PM10**: Concentration of particulate matter with a diameter of less than 10 micrometers.
* **CO (Carbon Monoxide)**: Concentration of Carbon Monoxide in the air.
* **AQI Value**: The dependent variable representing the predicted Air Quality Index (AQI) based on the input pollutant levels.

### ****2.2 Data Collection****

The data is collected from multiple sources, including:

DATASET:



* **Historical Pollutant Data**: Includes past measurements of pollutants like Ozone, NO, PM10, and CO from various monitoring stations, collected in CSV or other suitable formats.
* **Environmental Data**: External factors such as weather conditions (temperature, humidity) or other environmental parameters affecting air quality.
* **Geographical Information**: Data related to the location of monitoring stations, such as city, area, and geographical characteristics, which can influence AQI levels.

### ****2.3 Data Transformation****

Data transformation is performed to prepare the dataset for modeling:

* **Feature Encoding**: Categorical features like location type or pollution sources may be one-hot encoded to allow them to be used by machine learning models.
* **Log Transformation**: Continuous features such as pollutant concentrations (e.g., PM10, CO) may be transformed using logarithms to handle skewness and improve model performance.
* **Normalization**: Numerical features (such as pollutant levels) may be normalized or scaled to ensure consistency in the data and improve model efficiency.

### ****2.4 Data Pre-processing****

Before training the model, pre-processing steps are conducted:

* **Missing Data Handling**: Missing values are imputed based on the type of variable (mean for continuous variables, mode for categorical variables).
* **Outlier Detection**: Outliers in the pollutant data are identified and handled to avoid distorting the model’s predictions.
* **Data Splitting**: The dataset is split into training and testing sets to evaluate the model’s performance effectively.

### ****2.5 Model Building****

The prediction models are built using:

* **Random Forest**: An ensemble learning method that uses multiple decision trees to improve predictive accuracy, suitable for large datasets with multiple input features.
* **Gradient Boosting**: A boosting technique that builds trees sequentially, where each tree corrects the errors made by the previous one. These models are trained using the processed training data to predict AQI values based on the pollutant concentrations.

### ****2.6 Data from User****

The user interacts with the system by providing the following data:

* **Ozone (O3)**: Concentration of Ozone in the air.
* **NO**: Concentration of Nitrogen Oxides in the air.
* **PM10**: Concentration of particulate matter.
* **CO**: Concentration of Carbon Monoxide in the air.
* Ozone (O3) Category : Good, poor ,unhealthy
* This input is entered into the web interface developed using **Streamlit**.

### ****2.7 Data Validation****

Data validation ensures the integrity and consistency of the user input:

* **Format Checking**: Ensures all inputs are in the correct format (e.g., numerical inputs for pollutant levels, text inputs for location type).
* **Range Checking**: Validates that input values lie within expected ranges (e.g., pollutants should be non-negative, AQI values should be within standard ranges).
* **Missing Data**: Any missing or incomplete input is flagged, and the user is prompted to fill in all required fields.

### ****2.8 Model Call for Prediction****

Once the user input is validated, the pre-trained models (Random Forest or Gradient Boosting) are called to make predictions:

* The system loads the appropriate model based on the user’s location or other input variables.
* The model processes the input and predicts the AQI value based on the pollutant concentrations.

### ****2.9 Deployment****

The system is deployed on a cloud platform using **Streamlit**, making it accessible via a web interface. The application can be deployed on platforms such as **AWS** or **Heroku** to ensure scalability and availability for real-time predictions.

### ****2.10 Workflow Description****

1. **User Input**: The user interacts with the **Streamlit** app, providing the necessary inputs such as Ozone (O3) concentration, NO, PM10, CO levels, and location type through an input form.
2. **Data Transfer to Pickle File**: Once the user submits the input, the data is sent to a **Pickle** file. This file contains the pre-trained machine learning model, which was built using historical pollutant data and saved for deployment.
3. **Model Loading and Prediction Process**:
   * The **Pickle** file loads the trained model (Random Forest or Gradient Boosting).
   * The model processes the input values, applying necessary transformations (like log transformation or normalization) to handle skewness in the data.
   * Based on the input values, the model predicts the AQI value.
4. **Output Display**:
   * After the model generates the prediction, the result (predicted AQI value) is sent back to the **Streamlit** app.
   * The predicted AQI value is displayed to the user in a clear, user-friendly format.
5. **Visualization**:
   * Along with the prediction, the **Streamlit** app generates real-time visualizations, such as:
     + **Pollutant Levels vs AQI** graph
     + **Pollutant Contribution vs AQI Value** graph
   * These visualizations help users understand the relationship between pollutant concentrations and the predicted AQI value.
6. **Final Display**: The user can view both the prediction and the visualizations, assisting them in making data-driven decisions based on the predicted AQI values.