Software Requirement Specification (SRS)

Project: Air Quality Index (AQI) Predictor
Author: Tanmay Agarwal
Date: 30 August 2025

1. Introduction

1.1 Purpose

The **Air Quality Index (AQI) Predictor** is being developed to help predict pollutant levels (CO, NO_2, NMHC, etc.) based on weather conditions and sensor data. This will allow the system to **forecast AQI trends**, making it easier for government agencies, researchers, and the public to take **preventive measures accordingly**.

1.2 Scope

- Uses machine learning (Linear Regression, KNN, and other models) to predict pollutant concentrations.
- Provides insights into air quality variations influenced by weather and sensor data.
- Supports data-driven awareness for the general public and government bodies.
- Long-term goal: Improve air pollution monitoring and preventive health measures.

1.3 Benefits & Goals

- Early warnings about rising pollution.
- Assists in policy-making for environmental protection.
- Acts as an educational tool for students and researchers.

2. Overall Description

2.1 Boundaries & Objectives

- Focuses only on **pollutant prediction** (not real-time monitoring).
- Data source: **UCI Air Quality Dataset**.

- Predicts pollutants using **historical weather and sensor readings**.

2.2 Glossary

- **AQI**: Air Quality Index
- **CO**: Carbon Monoxide
- **NO_2**: Nitrogen Dioxide
- **NMHC**: Non-Methane Hydrocarbons
- **Regression**: A machine learning method for predicting continuous values
- **KNN**: K-Nearest Neighbors (regression technique)
- **SVR**: Support Vector Regression (ML algorithm for complex prediction)
- **Sensors**: Devices that measure pollutant-related gases
- **PT08.S1(CO)**: Tin Oxide (CO-sensitive)
- **PT08.S2(NMHC)**: Titanium Oxide (Non-Methane Hydrocarbons)
- **PT08.S3(NOx)**: Tungsten Oxide (Nitrogen Oxides)
- **PT08.S4(NO_2)**: Tungsten Oxide (Nitrogen Dioxide)
- **PT08.S5(O_3)**: Indium Oxide (Ozone-sensitive)

2.3 References

- UCI Machine Learning Repository: Air Quality Dataset
- Research papers on ML-based air pollution prediction
- WHO Air Quality Standards

2.4 Relation to Existing Systems

- Existing AQI dashboards (government platforms) only **display real-time values**.
- This system provides **prediction capabilities**, which most dashboards lack.

3. System Features

3.1 Key Features

- Accepts **sensor + weather data** as input.
- Predicts **multiple pollutants (CO, NO_2, NMHC, etc.)**.
- Supports **multiple ML models**: Linear Regression, KNN.
- Provides **accuracy metrics** (RMSE, R^2 score).
- Displays **visualizations** (predicted vs. actual, residual plots).

3.2 Target Users

- **Students & Researchers**: Learn and test ML techniques on environmental data.
- **Environmental Agencies**: Use predictions for policy-making.
- **General Public**: Understand AQI levels in a simple form.

4. System Constraints

4.1 Limitations

- Prediction accuracy depends on dataset quality.
- Requires Python with ML libraries installed (Scikit-learn, Pandas, Matplotlib).

4.2 Assumptions

- Input dataset is preprocessed and valid.
- Sensor readings are calibrated and reliable.

5. Functional Requirements

5.1 Input Module

- Accepts dataset with **weather and sensor readings**.

5.2 Data Preprocessing Module

- Cleans **invalid values**.
- Handles **missing data**.
- Normalizes features if required.

5.3 Prediction Module

- **Linear Regression**: Basic pollutant prediction.
- **KNN Regression**: Predicts pollutant levels using nearest neighbors.

5.4 Evaluation Module

- Calculates model performance using:
  - **RMSE (Root Mean Square Error)**
  - **R^2 Score (Coefficient of Determination)**

5.5 Output & Visualization Module

- Displays predicted pollutant levels (CO, NO_2, NMHC).
- Shows summary of system performance metrics.
- Provides **visualizations**:
  - Predicted vs. Actual plots.
  - Residual error plots.
  - Feature vs. pollutant trend graphs.
- kit-learn, Pandas, Matplotlib).

4.2 Assumptions

- Input dataset is preprocessed and valid.
- Sensor readings are calibrated and reliable.

5. Functional Requirements

5.1 Input Module

- Accepts dataset with **weather and sensor readings**.

5.2 Data Preprocessing Module

- Cleans **invalid values**.
- Handles **missing data**.
- Normalizes features if required.

5.3 Prediction Module

- **Linear Regression**: Basic pollutant prediction.
- **KNN Regression**: Predicts pollutant levels using nearest neighbors.

5.4 Evaluation Module

- Calculates model performance using:
  - **RMSE (Root Mean Square Error)**
  - **R^2 Score (Coefficient of Determination)**

5.5 Output & Visualization Module

- Displays predicted pollutant levels (CO, NO_2, NMHC).
- Shows summary of system performance metrics.
- Provides **visualizations**:
  - Predicted vs. Actual plots.
  - Residual error plots.
  - Feature vs. pollutant trend graphs.
- **Predicts pollutant levels using nearest neighbors.**

5.4 Evaluation Module

- Calculates model performance using:
  - **RMSE (Root Mean Square Error)**
  - **R^2 Score (Coefficient of Determination)**

5.5 Output & Visualization Module

- Displays predicted pollutant levels (CO, NO_2, NMHC).
- Shows summary of system performance metrics.
- Provides **visualizations**:
  - Predicted vs. Actual plots.
  - Residual error plots.
  - Feature vs. pollutant trend graphs.
  - kit-learn, Pandas, Matplotlib).

4.2 Assumptions

- Input dataset is preprocessed and valid.
- Sensor readings are calibrated and reliable.

5. Functional Requirements

5.1 Input Module

- Accepts dataset with **weather and sensor readings**.

5.2 Data Preprocessing Module

- Cleans **invalid values**.
- Handles **missing data**.
- Normalizes features if required.

5.3 Prediction Module

- **Linear Regression**: Basic pollutant prediction.
- **KNN Regression**: Predicts pollutant levels using nearest neighbors.

5.4 Evaluation Module

- Calculates model performance using:
    - **RMSE (Root Mean Square Error)**
    - **R^2 Score (Coefficient of Determination)**

5.5 Output & Visualization Module

- Displays predicted pollutant levels (CO, NO_2, NMHC).
- Shows summary of system performance metrics.
- Provides **visualizations**:
    - Predicted vs. Actual plots.
    - Residual error plots.
    - Feature vs. pollutant trend graphs.