

Problem Statement: Air Quality Analysis and Prediction in Tamil Nadu

Problem Definition: The project aims to analyse and visualize air quality data from monitoring stations in Tamil Nadu. The objective is to gain insights into air pollution trends, identify areas with high pollution levels, and develop a predictive model to estimate RSPM/PM10 levels based on SO2 and NO2 levels. This project involves defining objectives, designing the analysis approach, selecting visualization techniques, and creating a predictive model using Python and relevant libraries.

My Approach and methods to analyse the problem is as follows:

1.Project Objectives:

Analyse historical air quality data to identify trends in air pollution levels. Identify areas in Tamil Nadu with consistently high pollution levels. Develop a predictive model to estimate RSPM/PM10 levels based on SO2 and NO2 levels.

2.Data Collection:

Gather air quality data from monitoring stations in Tamil Nadu. You may obtain this data from government agencies, environmental organizations, or academic sources. Ensure that the data includes information on SO2, NO2, RSPM/PM10 levels, and geographic location.

3.Data Preprocessing:

Clean and prepare the data for analysis. This may involve handling missing values, removing duplicates, and converting data types. Additionally, you may need to perform data aggregation if the data is recorded at different time intervals.

4.Exploratory Data Analysis (EDA):

Descriptive statistics to understand data distributions. Time series analysis to identify temporal trends. Geographic mapping to visualize pollution levels across Tamil Nadu.

5.Visualization:

Choose appropriate visualization techniques to communicate your findings effectively. Consider using libraries such as Matplotlib, Seaborn, and Plotly for creating visualizations. Time series plots to show pollution trends over time. Heatmaps to visualize spatial variations in pollution levels. Scatter plots or bar charts to show relationships between variables.

6.Feature Engineering:

Create relevant features for your predictive model. You can calculate lagged values, rolling averages, or other derived features that might capture important patterns in the data.

7.Model Development:

Build a predictive model to estimate RSPM/PM10 levels based on SO2 and NO2 levels. You can use machine learning techniques for regression tasks. Popular Python libraries for this purpose include Scikit-Learn, XG Boost, and TensorFlow. Remember to split your data into training and testing sets to evaluate the model's performance.

8. Model Evaluation:

Assess the performance of your predictive model using appropriate metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), or R-squared. Make sure the model performs well on the testing data to ensure its generalization ability.

9. Interpretability:

Consider using model interpretability techniques to understand how the model makes predictions. This is essential for gaining insights into the relationship between pollutant levels and air quality.

10. Documentation and Reporting:

Document your analysis process, findings, and the steps you took to achieve your objectives. Create a report or presentation summarizing your project's results and insights.

11. Recommendations and Future Work:

Based on your analysis, provide recommendations for policymakers, environmental organizations, or stakeholders to address air pollution issues in Tamil Nadu. You can also suggest avenues for future research or improvements to your predictive model.

12. Deployment:

If applicable, consider deploying your predictive model as a web application or API that can provide real-time air quality predictions based on SO₂ and NO₂ levels.