

PHASE 2:

AIR QUALITY ANALYSIS

Problem Statement:

The project aims to analyze 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.

Approach for making design:

Data Mining:

Data mining or Knowledge Discovery (KD) is used to read and analyze large datasets and then finding/extracting patterns from the data. It is used for predicting the future trends or forecast patterns over a period. Data mining algorithms are usually based on well known mathematical algorithms and techniques. There are two types of data

mining learning algorithms: 1) Supervised algorithms and 2) Unsupervised algorithms.

We are going to make optimal use of these to train our machine learning model for better prediction. The dataset is provided in the Government website.

Unsupervised learning algorithm:

The Unsupervised algorithm is the process in which the training dataset contains only the input set and not the corresponding target vectors. The main criterion is to find groups or patterns of similar examples within the dataset, called as clustering.

STEPS INVOLVED:

1. Data collection: Collect a dataset of air quality measurements, as described above.
2. Data preparation: Clean and prepare the data for machine learning.
3. Model selection: Choose an unsupervised learning algorithm to train your model. Some popular algorithms for air quality analysis include clustering and anomaly detection.
4. Model training: Train the model on the prepared data.
5. Model evaluation: Evaluate the performance of the trained model on a held-out test set. This will give you an idea of how well the model is able to identify patterns in the data.

6. Model interpretation: Interpret the results of the model to gain insights into the air quality data. For example, you could use clustering to identify groups of air quality monitoring stations with similar air quality patterns.

Supervised learning algorithm:

The Supervised algorithm is the process in which the training data comprises of both the training and the corresponding output target vectors. In this project, a supervised learning algorithm called Artificial Neural Network (ANN) has been used for training, validation and testing the dataset. In addition, to the ANN, a Multiple Linear Regression (MLR) model has been used for comparing the performance against the ANN. The below section introduces the processes of Artificial Neural Network (ANN) and Multiple Linear Regression (MLR).

STEPS INVOLVED:

1. Data collection: Collect a dataset of air quality measurements, including the pollutants of interest (e.g., PM2.5, PM10, ozone, nitrogen dioxide, sulfur dioxide), as well as other factors that may affect air quality (e.g., weather conditions, traffic data, industrial emissions).

2. Data preparation: Clean and prepare the data for machine learning. This may involve removing outliers, filling in missing values, and scaling the data.
3. Feature engineering: Create new features from the existing data that may be more informative for predicting air quality. For example, you could create a feature that represents the average wind speed over the past 24 hours.
4. Model selection: Choose a supervised learning algorithm to train your model. Some popular algorithms for air quality prediction include linear regression, random forest, and support vector machines.
5. Model training: Train the model on the prepared data. This involves feeding the model the input features and the corresponding output labels (air quality measurements).
6. Model evaluation: Evaluate the performance of the trained model on a held-out test set. This will give you an idea of how well the model will generalize to new data.
7. Model deployment: Once you are satisfied with the performance of the model, you can deploy it to production. This may involve saving the model to a file, or integrating it into a web service or mobile app.

Conclusion: Thus the design making phase has been implemented successfully.