# **Air Quality Analysis in Tamil Nadu**

## **Introduction**

The "Air Quality Analysis in Tamil Nadu" project aims to analyze and visualize air quality data collected from monitoring stations across the region. The primary objectives of this project are to gain insights into air pollution trends, identify areas with high pollution levels, and develop a predictive model for estimating RSPM/PM10 levels based on SO2 and NO2 levels. The project will utilize Python and relevant libraries for data analysis, visualization, and modeling.

## **Project Objectives**

The project objectives are as follows:

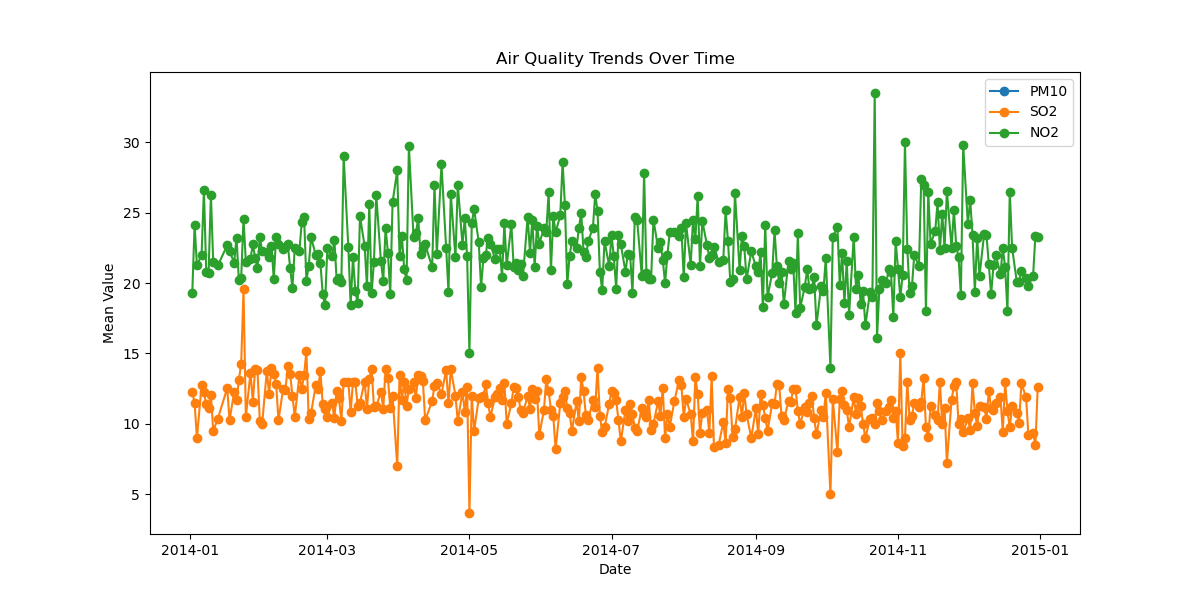
* Analyzing Air Quality Trends
* Objective: To analyze historical air quality data to identify trends and patterns in RSPM/PM10, SO2, and NO2 levels over time.
* Approach: Data preprocessing, time series analysis, and visualization techniques will be used to achieve this objective.
* Identifying Pollution Hotspots
* Objective: To locate regions within Tamil Nadu with consistently high levels of air pollution.
* Approach: Spatial data analysis and visualization, including heatmaps, will be employed to identify pollution hotspots.
* Building a Predictive Model for RSPM/PM10 Levels
* Objective: To develop a machine learning model that predicts RSPM/PM10 levels based on SO2 and NO2 levels.
* Approach: Data preparation, model selection, training, and evaluation will be performed to create the predictive model.

## **Analysis Approach**

The analysis approach will consist of the following steps:

* Data Loading
* Load air quality data from monitoring stations in Tamil Nadu. Ensure the dataset contains relevant parameters, including 'Date,' 'PM10,' 'SO2,' 'NO2,' 'Latitude,' and 'Longitude.'
* Data Preprocessing
* Clean and preprocess the data to handle missing values, duplicates, and ensure proper formatting.
* Analyzing Air Quality Trends
* Perform time series analysis to identify trends in air quality parameters.
* Create visualizations such as line charts to visualize the trends over time.
* Identifying Pollution Hotspots
* Utilize geographical data to plot monitoring stations on a map.
* Generate heatmaps to identify areas with consistently high pollution levels.
* Building a Predictive Model
* Prepare the dataset for machine learning.
* Select an appropriate machine learning algorithm (e.g., Linear Regression) for predicting RSPM/PM10 levels based on SO2 and NO2 levels.
* Train, evaluate, and fine-tune the model.

## **Visualization of Trends**



## **Python Code**

I utilized this code to perform the analysis of these tasks.

|  |
| --- |
| import pandas as pd import matplotlib.pyplot as plt import seaborn as sns from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LinearRegression from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score  # Load the data from a CSV file data = pd.read\_csv('./data.csv') # Replace 'your\_data.csv' with your file path  # 1. Analyzing Air Quality Trends # Data Preparation data['Date'] = pd.to\_datetime(data['Date']) # Convert date column to datetime  # Time Series Analysis mean\_pm10 = data.groupby('Date')['PM10'].mean() mean\_so2 = data.groupby('Date')['SO2'].mean() mean\_no2 = data.groupby('Date')['NO2'].mean()  # Visualization plt.figure(figsize=(12, 6)) plt.plot(mean\_pm10.index, mean\_pm10.values, label='PM10', linestyle='-', marker='o') plt.plot(mean\_so2.index, mean\_so2.values, label='SO2', linestyle='-', marker='o') plt.plot(mean\_no2.index, mean\_no2.values, label='NO2', linestyle='-', marker='o') plt.xlabel('Date') plt.ylabel('Mean Value') plt.title('Air Quality Trends Over Time') plt.legend() plt.show()  # 2. Identifying Pollution Hotspots # Heatmap heatmap\_data = data.pivot\_table(index='Latitude', columns='Longitude', values='PM10') plt.figure(figsize=(12, 8)) sns.heatmap(heatmap\_data, cmap='YlGnBu', annot=True) plt.title('Pollution Hotspots (PM10)') plt.show()  # 3. Building a Predictive Model for RSPM/PM10 Levels # Data Preparation X = data[['SO2', 'NO2']] # Features y = data['PM10'] # Target variable  # Data Splitting X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  # Model Selection and Training model = LinearRegression() model.fit(X\_train, y\_train)  # Model Evaluation y\_pred = model.predict(X\_test) mae = mean\_absolute\_error(y\_test, y\_pred) mse = mean\_squared\_error(y\_test, y\_pred) r2 = r2\_score(y\_test, y\_pred)  print(f'Mean Absolute Error: {mae}') print(f'Mean Squared Error: {mse}') print(f'R-squared: {r2}')  # Prediction (you can input new SO2 and NO2 values to make predictions) new\_data = pd.DataFrame({'SO2': [value1], 'NO2': [value2]}) predicted\_pm10 = model.predict(new\_data) print(f'Predicted PM10: {predicted\_pm10[0]}') |

**Thank You**