# AIR QUALITY ANALYSIS IN TAMIL NADU

| | |
|---|---|
| Date | 28-10-2023 |
| Team ID | 719 |
| Project Name | Air Quality Analysis in Tamil Nadu |

**Table of Contents:**



**1. Introduction:**

Air pollution is a growing concern worldwide, and its adverse effects on human health and the environment are well-documented. In the Indian state of Tamil Nadu, rapid urbanization and industrialization have led to an increase in air pollution levels, raising serious public health and environmental issues. To address this problem, a comprehensive project has been initiated to analyze and visualize air quality data from monitoring stations located throughout Tamil Nadu. The project's primary goal is to gain insights into air pollution trends, identify areas with high pollution levels, and develop a predictive model to estimate Respirable Suspended Particulate Matter (RSPM) and Particulate Matter (PM10) levels based on the concentrations of Sulphur dioxide (SO2) and Nitrogen dioxide (NO2).

**2. Problem Statement:**

The project involves establishing air quality sensors in cities, identifying pollution sources, developing an accessible app, and optimizing transportation routes to enhance air quality, public health, and environmental sustainability.

**3. Tools and Libraries:**

**Python:**

- ✓ Python is a versatile, high-level programming language known for its simplicity and readability. It's widely used in data analysis, machine learning, and scientific computing due to its extensive libraries and frameworks.

**Pandas:**

- ✓ Pandas is a Python library for data manipulation and analysis. It provides data structures like DataFrames and Series, making it easy to work with structured data. Pandas is essential for data loading, cleaning, and transformation.

**NumPy:**

- ✓ NumPy is another Python library that focuses on numerical computing. It provides support for large, multi-dimensional arrays and matrices, as well as a variety of mathematical functions to operate on these arrays. It's fundamental for numerical data processing.

**Matplotlib:**

- ✓ Matplotlib is a popular data visualization library in Python. It allows you to create static, animated, or interactive visualizations in a wide range of formats, including line plots, bar charts, scatter plots, and more. It's excellent for visualizing data and trends.

**Seaborn:**

- ✓ Seaborn is a Python data visualization library built on top of Matplotlib. It provides a high-level interface for creating attractive and informative statistical graphics. Seaborn simplifies the creation of complex visualizations and is often used for creating aesthetically pleasing charts.

**Scikit-Learn:**

- ✓ Scikit-Learn, also known as sklearn, is a powerful machine learning library in Python. It offers a wide range of tools for data preprocessing, model selection, training, and evaluation. It's especially useful for building predictive models and performing machine learning tasks.

In the context of our project, these tools and libraries play the following roles:

- Python serves as the programming language for your project, providing a flexible and accessible environment for data analysis and modeling.
- Pandas is used for data manipulation and analysis, including loading, cleaning, and organizing the air quality data.
- NumPy complements Pandas by providing fundamental support for numerical operations and handling multi-dimensional arrays, which are often used in data analysis.
- Matplotlib is employed for creating visualizations that help in understanding pollution trends and conveying insights effectively.
- Seaborn enhances the visualization process by providing a higher-level interface to create aesthetically pleasing and informative statistical graphics.
- Scikit-Learn plays a crucial role in building the predictive model that estimates RSPM/PM10 levels based on SO2 and NO2 data.

**4. Model Selection and Training:**

- Choose Support Vector Machine (SVM) for regression and classification tasks, handling complex data relationships.
- Train the model using training data and target variables.
- Monitor performance in production to adapt to changing air quality conditions and plan for retraining or updates as new data or pollution patterns change.
- We may consider using an alternative algorithm for this model in order to enhance our ability to predict values. The choice of the best algorithm depends on the specific characteristics of the dataset and the problem we are trying to address.

**Support Vector Machine (SVM):**

- ✓ Support Vector Machine (SVM) is a machine learning algorithm used to develop a predictive model for estimating RSPM/PM10 levels based on the levels of Sulphur dioxide (SO2) and Nitrogen dioxide (NO2).
- ✓ SVM works by finding the best hyperplane that separates data points with different levels of RSPM/PM10 based on the levels of SO2 and NO2.
- ✓ This hyperplane allows the model to make predictions about air quality, specifically RSPM/PM10 levels, by analyzing the relationships between the pollutants and creating a decision boundary in a high-dimensional feature space.
- ✓ SVM is a valuable tool for understanding and forecasting air quality trends.

**Decision Tree:**

- ✓ A Decision Tree is a predictive model that makes estimates of RSPM/PM10 levels based on the levels of two key pollutants, Sulphur dioxide (SO2) and Nitrogen dioxide (NO2).
- ✓ It does so by creating a tree-like structure of decisions and tests, where each branch and leaf node represents a different decision based on the levels of these pollutants.
- ✓ The final leaf nodes provide predictions for RSPM/PM10 levels based on the pollutant levels and decisions made along the way.
- ✓ This Decision Tree helps in understanding and predicting air quality by analyzing the relationships between SO2, NO2, and RSPM/PM10 levels.

**5. Pollution Trends and Areas:**

Calculate average SO2, NO2, and RSPM/PM10 levels across different monitoring stations, cities, or areas.

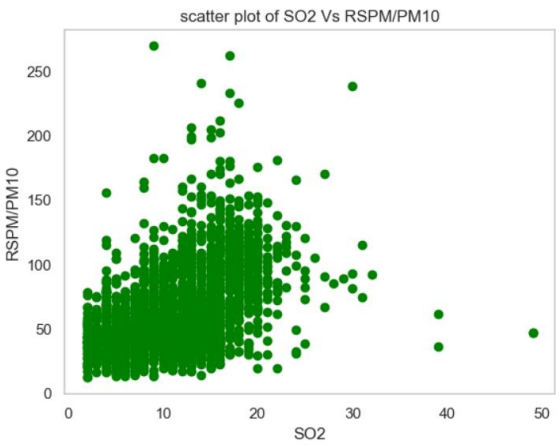
```
#mean SO2,NO2,RSPM/PM10
mean_SO2 = new_data['SO2'].mean()
mean_NO2 = new_data['NO2'].mean()
mean_RSPM_PM10 = new_data['RSPM/PM10'].mean()
```

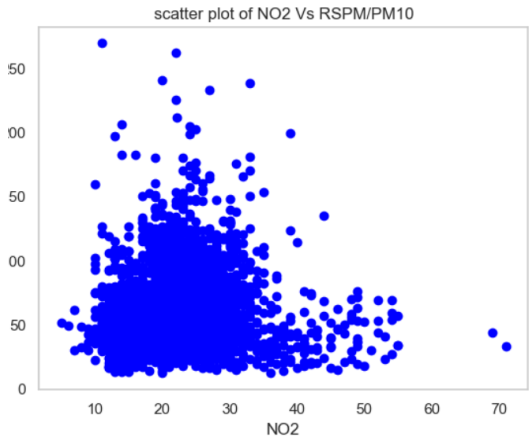
```
# Assuming your DataFrame is named 'air_quality_data'
average=new_data.groupby(['Location of Monitoring Station','City/Town/Village/Area', 'Type of Location'])[['SO2', 'NO2', 'RSPM/PM10']].mean()
```

```
average.mean()
```

**6. Visualization:**

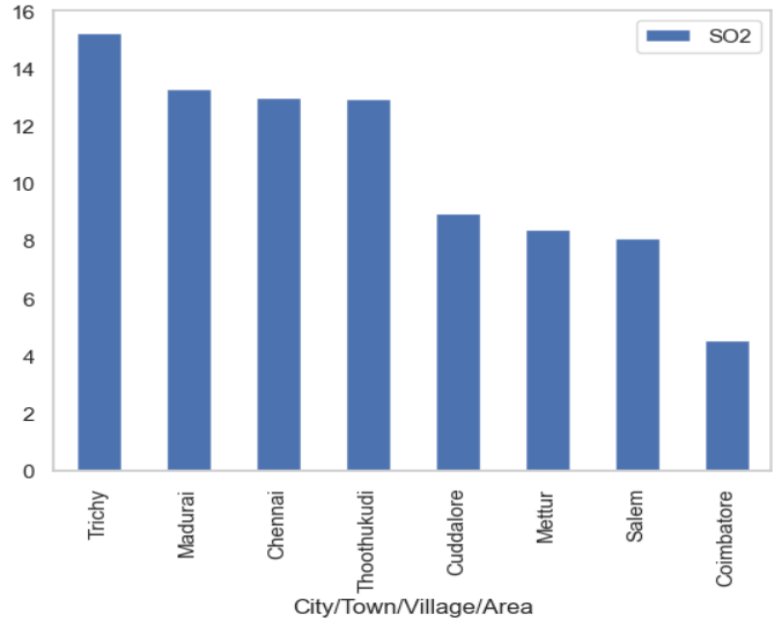
```
fig=plt.figure()
plt.scatter(new_data['SO2'],new_data['RSPM/PM10'], color ='green')
plt.xlabel("SO2")
plt.ylabel("RSPM/PM10")
plt.title("scatter plot of SO2 Vs RSPM/PM10")
plt.grid(False)
plt.show()

fig=plt.figure()
plt.scatter(new_data['NO2'],new_data['RSPM/PM10'], color ='blue')
plt.xlabel("NO2")
plt.ylabel("RSPM/PM10")
plt.title("scatter plot of NO2 Vs RSPM/PM10")
plt.grid(False)
plt.show()
```



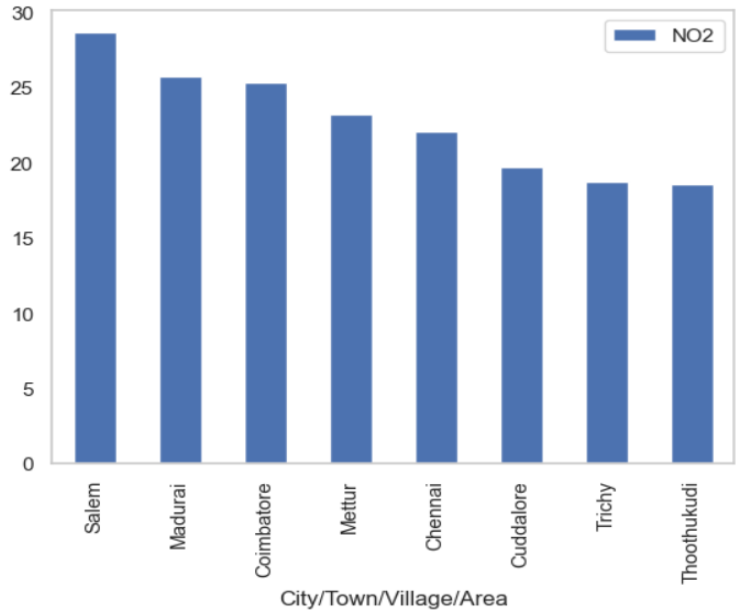



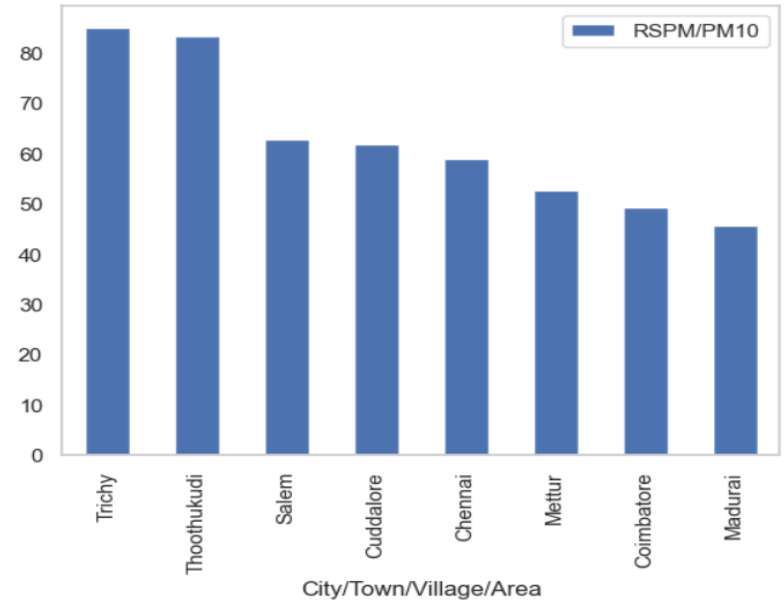

```
plt.figure(figsize=(1,1))
maxSO2 = loc.sort_values(by='SO2',ascending=False)
maxSO2.loc[:,['SO2']].head(10).plot(kind='bar');
plt.grid(False)
plt.show()

plt.figure(figsize=(1,1))
maxNO2 = loc.sort_values(by='NO2',ascending=False)
maxNO2.loc[:,['NO2']].head(10).plot(kind='bar');
plt.grid(False)
plt.show()

maxRSPM_PM10 = loc.sort_values(by='RSPM/PM10',ascending=False);
maxRSPM_PM10.loc[:,['RSPM/PM10']].head(10).plot(kind='bar');
plt.grid(False)
plt.show()
```

```
<Figure size 100x100 with 0 Axes>
```

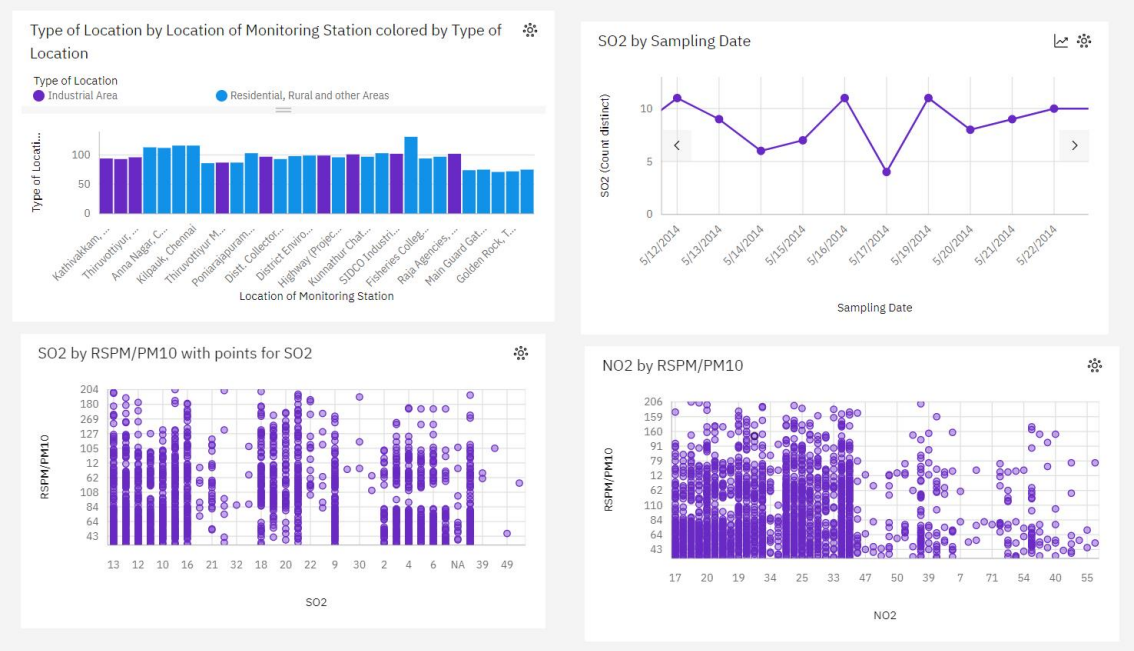


<Figure size 100x100 with 0 Axes>



## 6.1 Data Visualization with IBM Cognos:




## Conclusion:

The SVM model outperforms the Decision Tree model in terms of accuracy, with significantly lower mean absolute error and mean squared error. It also has a strong fit to the data, explaining a significant portion of its variance. The Decision Tree model, on the other hand, shows a negative R-squared, indicating a poor fit. Therefore, the SVM model is the better choice for this task, offering superior predictive accuracy and a more robust overall fit to the data.