GRT INSTITUTE OF ENGINEERING AND TECHNOLOGY, TIRUTTANI - 631209

Approved by AICTE, New Delhi Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AIR QUALITY ANALYSIS AND PREDICTION IN TAMILNADU

PROJECT REPORT

SUBMITTED BY

SWATHI.K

3RD YEAR 5TH SEM

110321104052

swathikollapuri11@gmail.com

ANNA UNIVERSITY:CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"AIR QUALITY ANALYSIS AND PREDICTION IN TAMILNADU"** is the bonafide work of **"SWATHI.K [110321104052]"** who carried out the project work under my our supervision.

Dr.N. Kamal M.E.,Ph.D.,
HOD
Department of Computer Science And Engineering
GRT Institute of Engineering and Technology
Tiruttani

Mr.T.A. Vinayagam M.Tech.,
Assistant professor
Department of Computer Science And Engineering
GRT Institute of Engineering and Technology
Tiruttani

Certified that the candidates were examined in Viva-voce in the Examination

Held on **1-11-2023**

ACKNOWLEDGEMENT

We thank our Management for providing us all support to complete this project successfully. Our sincere thanks to honorable **Chairman**, **Shri. G. RAJENDRAN and Managing Director, Shri.G.R. RADHAKRISHNAN** for creating a wonderful atmosphere inside the campus.

We are very grateful to **Dr.S. ARUMUGAM, M.E., Ph.D., Principal**, for providing us with consistent guidance and motivation to execute a real time project to learn and experience the project work in an environments to complete our project successfully.

Our sincere thanks to **Dr.N. KAMAL, M.E., Ph.D., Professor and Head, Department of Computer Science and Engineering** for giving me this wonderful opportunity to do the project and providing the require facilities to fulfill our work.

We are highly indebted and thankful to our project Evaluators **Mrs V. Priya M.E**., and **Mrs. Edith Esther M.E., Assistant Professor, Department of Computer Science and Engineering** for his immense support in doing the project.

We are very grateful to our internal guide **Mr. T.A. VINAYAGAM, M.Tech., Assistant Professor, Department of Computer Science and Engineering** for guiding us with her valuable suggestions to complete our project.

We also dedicate equal and grateful acknowledgement to all the **respectable members of the faculty and lab in-charges** of the Department of Computer Science and Engineering, friends and our families for their motivation, encouragement and continuous support.

Our sincere thanks **to IBM and Skill Up Team members** for giving me this wonderful opportunity to do the project and providing the require guidance and valuable online sessions.

TABLE OF CONTENT

ABSTRACT:

Air quality is a critical environmental concern, with adverse impacts on public health and ecosystems. This study focuses on the analysis and prediction of air quality in the state of Tamil Nadu, India. We leverage historical air quality data, meteorological information, and advanced machine learning techniques to develop a comprehensive understanding of air quality patterns in the region.

The analysis phase involves the examination of historical air quality data from various monitoring stations across Tamil Nadu. We assess the levels of key air pollutants such as particulate matter (PM2.5 and PM10), nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO), and ozone (O3). Spatial and temporal trends are identified, allowing us to pinpoint areas of concern and periods of high pollution.

To enhance prediction accuracy, we employ machine learning models such as neural networks, support vector machines, and ensemble methods. These models utilize historical air quality data, meteorological variables (temperature, humidity, wind speed, etc.), and geographical information to forecast air quality levels for specific locations and timeframes.

Furthermore, we consider the impact of local factors like industrial emissions, vehicular traffic, and seasonal variations on air quality. This research aims to provide valuable insights for policymakers, environmental agencies, and the general public. By predicting air quality trends, we can implement targeted interventions to mitigate pollution and improve the overall quality of life in Tamil Nadu.

In conclusion, this study contributes to the field of environmental science by offering a comprehensive analysis of air quality in Tamil Nadu, along with accurate predictive models. The findings and predictions can inform decision-makers and enable the development of effective strategies to combat air pollution and safeguard public health.

CHAPTER 1
PHASE 1

INTRODUCTION:

Air quality is a critical aspect of environmental health, directly impacting the well-being of communities and ecosystems. In recent years, concerns about air pollution have grown substantially, driven by industrialization, urbanization, and increased vehicular traffic. Tamil Nadu, a state in southern India, is no exception to these challenges. Its diverse landscape, including industrial regions and densely populated urban areas, makes it imperative to analyze and predict air quality for informed decision-making and mitigation efforts.

This study focuses on the comprehensive analysis and prediction of air quality in Tamil Nadu. We recognize that deteriorating air quality poses significant risks to public health, with adverse effects ranging from respiratory illnesses to cardiovascular problems. Moreover, it has ecological consequences, impacting soil quality, water bodies, and overall biodiversity. Thus, understanding the dynamics of air pollution in Tamil Nadu is crucial.

To address this issue, we embark on an in-depth investigation, utilizing historical air quality data collected from monitoring stations situated across the state. Our research is driven by the following objectives:

1. **Data Collection and Analysis**: We gather extensive data on key air pollutants, including particulate matter (PM2.5 and PM10), nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO), and ozone (O3). These data are analyzed to discern patterns, trends, and spatial variations in air quality.

2. **Meteorological Influence**: We consider the influence of meteorological parameters such as temperature, humidity, wind speed, and atmospheric pressure on air quality. Understanding these relationships is essential for accurate prediction.

3. **Machine Learning Models**: To improve prediction accuracy, we employ advanced machine learning models. These models take into account historical air quality data, meteorological variables, and geographical features to forecast air quality levels for specific locations and timeframes.

4. **Local Factors**: We acknowledge the role of local factors, including industrial emissions, traffic congestion, and seasonal variations, in shaping air quality. These factors are integrated into our analysis to provide a holistic view.

By achieving these objectives, our study aims to provide valuable insights into the current state of air quality in Tamil Nadu and predict future trends. These insights can serve as a foundation for evidence-based policymaking, helping government agencies, environmental organizations, and the public to take proactive measures to combat air pollution, reduce its impact on health, and protect the environment.

In the subsequent sections of this research, we delve deeper into the methodologies employed, present our findings, and offer recommendations for mitigating air pollution in Tamil Nadu, ultimately contributing to the broader effort to enhance the quality of life in this region.

PROBLEM DEFINITION:

The problem at hand is the analysis and prediction of air quality in the state of Tamil Nadu, India. Air quality is a critical environmental concern that affects the health and well-being of the state's residents and the sustainability of its ecosystems. The specific problem can be broken down into several key components:

1. **Air Pollution Assessment:** The first aspect of the problem involves assessing the current state of air pollution in Tamil Nadu. This includes understanding the concentration levels of various air pollutants such as particulate matter (PM2.5 and PM10), nitrogen dioxide (NO_2), sulfur dioxide (SO_2), carbon monoxide (CO), and ozone (O_3). It also involves identifying pollution hotspots and areas with consistently poor air quality.

2. **Temporal and Spatial Variations**: Air quality is not static; it varies by time and location. Therefore, the problem includes analyzing the temporal and spatial variations in air quality. This involves determining if air quality deteriorates during specific seasons, times of day, or weather conditions, and identifying regions where pollution is most severe.

3. **Meteorological Factors:** Meteorological variables like temperature, humidity, wind speed, and atmospheric pressure have a significant influence on air quality. The problem encompasses understanding how these factors interact with air pollutants and contribute to fluctuations in air quality.

4. **Prediction and Forecasting**: An essential component of this problem is the development of accurate prediction models. These models should use historical air quality data, meteorological information, and possibly other relevant features to forecast future air quality levels. The aim is to provide advance warnings and forecasts to residents, authorities, and policymakers.

5. **Local Factors and Sources**: To address air quality issues effectively, it's crucial to identify and quantify local sources of pollution. This includes assessing the impact of industries, vehicular emissions, construction activities, and other localized factors on air quality.

6. **Public Health Implications**: Another critical dimension of the problem is understanding the public health implications of poor air quality. This involves analyzing health data to identify associations between air pollution and health outcomes such as respiratory illnesses, cardiovascular diseases, and mortality rates.

7. **Policy and Intervention Recommendations**: Ultimately, the goal is to provide actionable insights and recommendations for policymakers and relevant authorities. This includes suggesting interventions to reduce air pollution, improve air quality, and protect public health.

In summary, the problem of air quality analysis and prediction in Tamil Nadu is multifaceted, encompassing data collection, analysis, modeling, and the development of strategies to address air pollution. It requires a multidisciplinary approach, including environmental science, meteorology, data science, and public health expertise, to comprehensively tackle the issue and improve the overall quality of life in the state.

OBJECTIVES:

The objectives of the "Air Quality Analysis and Prediction in Tamil Nadu" project can be summarized as follows:

1. Data Collection andCompilation:

- Gather historical air quality data from monitoring stations across Tamil Nadu.

- Collect relevant meteorological data, including temperature, humidity, wind speed, and atmospheric pressure.

2. Air Pollution Assessment:

- Analyze the levels of key air pollutants (e.g., PM2.5, PM10, NO2, SO2, CO, O3) to understand their distribution and concentrations throughout Tamil Nadu.

- Identify regions or cities with consistently poor air quality.

3. Temporal and Seasonal Patterns:

- Investigate temporal variations in air quality, including daily, monthly, and seasonal patterns.

- Determine if there are specific times of the year or day when air quality is particularly affected.

4. Spatial Analysis:

- Perform spatial analysis to identify areas with the most significant air quality challenges.

- Create air quality maps to visualize spatial variations.

5. Meteorological Impact Assessment:

- Examine the influence of meteorological factors on air quality, including their role in the dispersion and concentration of pollutants.

6. Development of Prediction Models:

- Utilize machine learning and statistical modeling techniques to develop accurate prediction models for air quality.

- Incorporate historical air quality data, meteorological parameters, and potentially other relevant features.

7. Short-term and Long-term Forecasting:

- Generate short-term (daily or hourly) and long-term (weekly, monthly) air quality forecasts for various locations in Tamil Nadu.

- Provide predictive insights to help residents and authorities plan accordingly.

8. Identification of Pollution Sources:

- Identify and quantify the contributions of different pollution sources, including industrial emissions, vehicular traffic, and natural factors, to air pollution in specific areas.

9. Public Health Assessment:

- Analyze health data to establish associations between air pollution and public health outcomes.

- Estimate the health and economic impacts of poor air quality on the population.

10. Policy Recommendations:

- Provide evidence-based recommendations to policymakers and relevant authorities for implementing effective measures to mitigate air pollution.

- Suggest policies and interventions to reduce emissions, improve air quality, and protect public health.

11. Public Awareness and Engagement:

- Promote public awareness of air quality issues through educational campaigns and real-time air quality reporting.

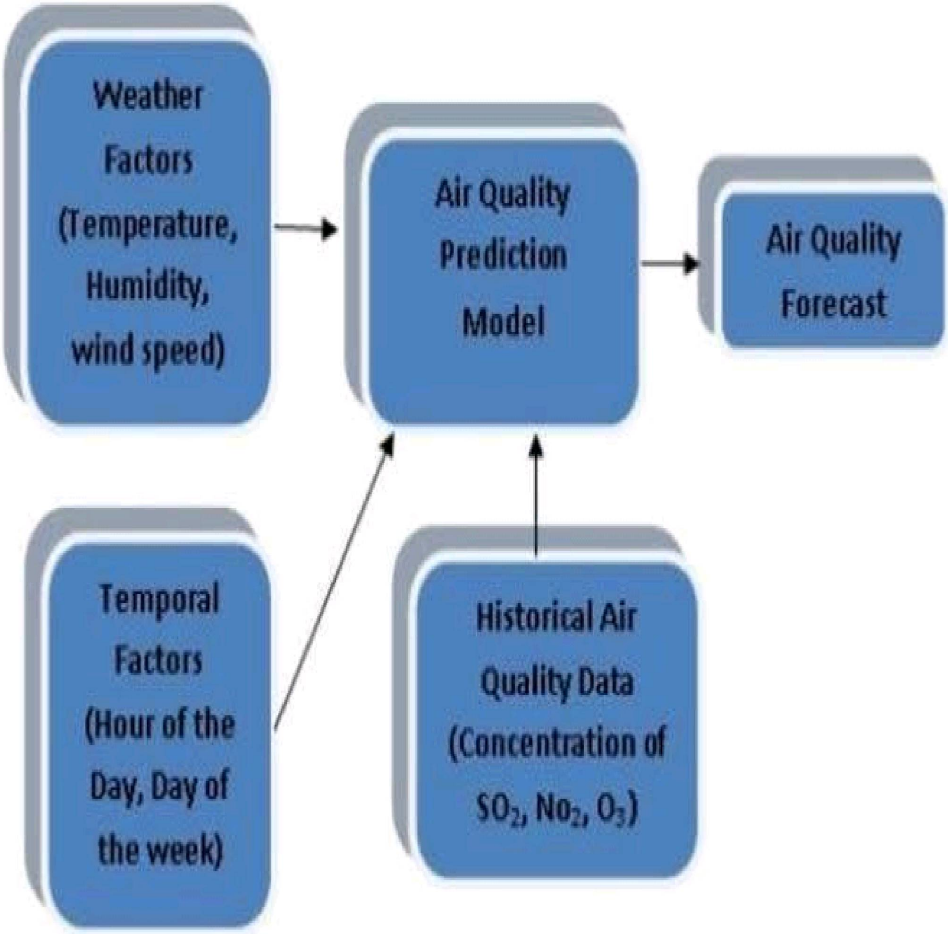
- Encourage community involvement in pollution reduction efforts.

12. Long-Term Monitoring Framework:

- Establish a sustainable framework for continuous air quality monitoring and reporting to ensure ongoing improvements in air quality.

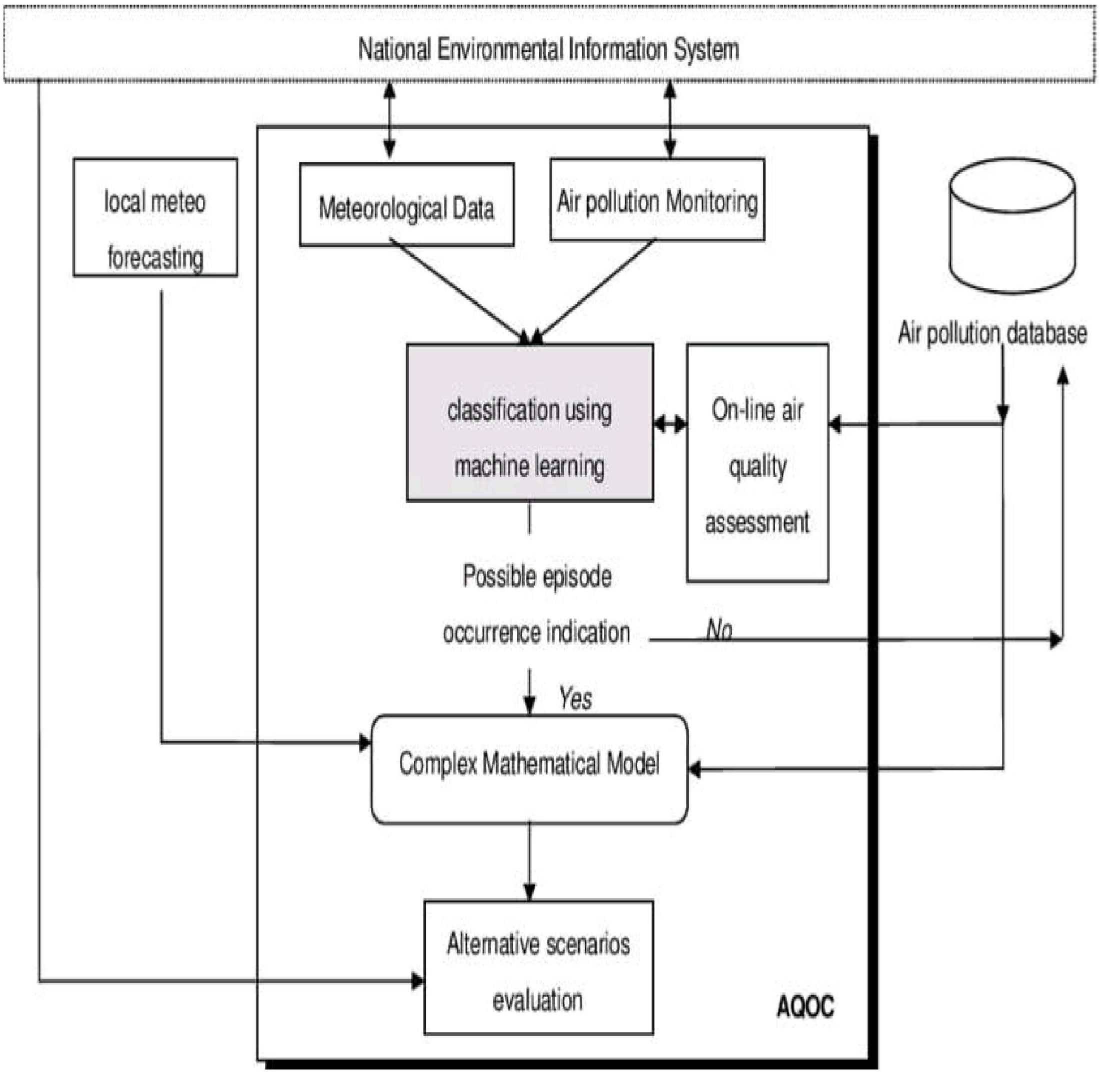
By achieving these objectives, the project aims to enhance the understanding of air quality dynamics in Tamil Nadu, provide accurate predictions and actionable recommendations, and ultimately contribute to improving air quality and the well-being of the population. This multidisciplinary effort involves collaboration between environmental scientists, data analysts, meteorologists, healthcare experts, and policymakers.

CASE STUDY



Weather Factors (Temperature, Humidity, wind speed)
Air Quality Prediction Model
Air Quality Forecast
Temporal Factors (Hour of the Day, Day of the week)
Historical Air Quality Data (Concentration of SO_2, No_2, O_3)

SYSTEM ARCHITECTURE:



ER DIAGRAM:

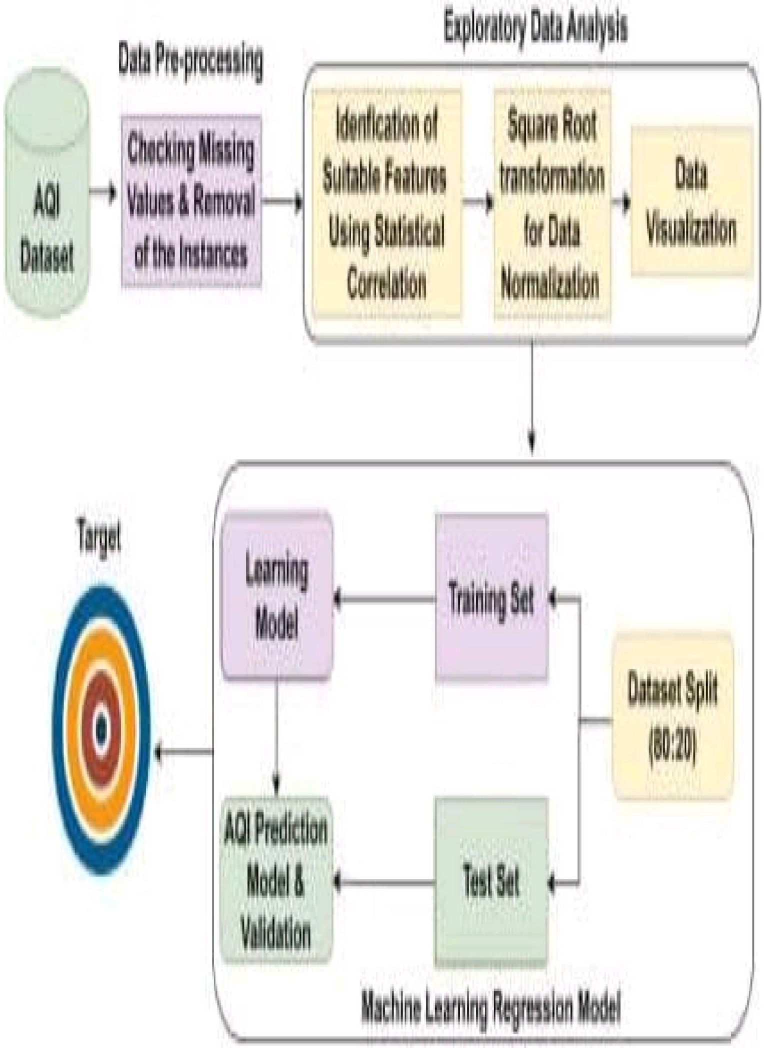
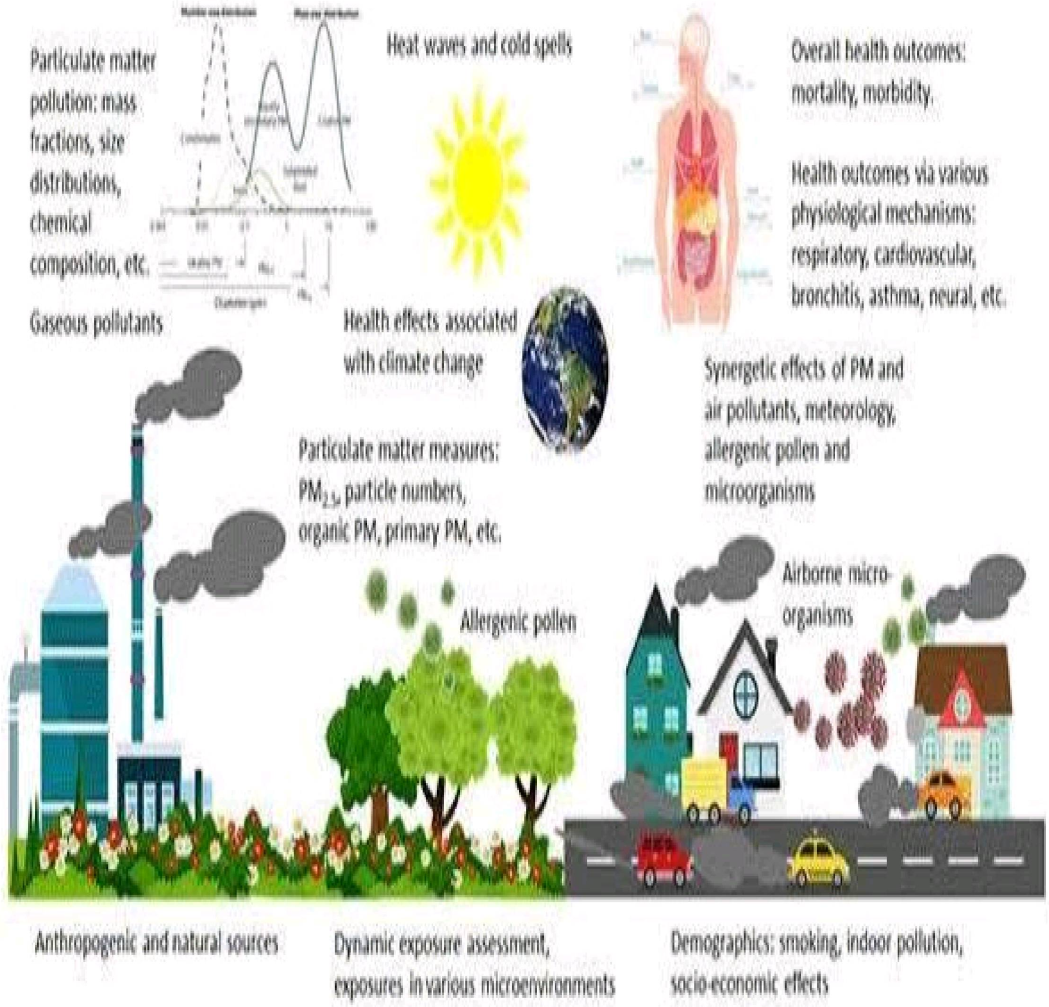

Exploratory Data Analysis
Data Pre-processing
AQI Dataset
Checking Missing Values & Removal of the Instances
Idenfication of Suitable Features Using Statistical Correlation
Square Root transformation for Data Normalization
Data Visualization
Target
Learning Model
Training Set
Dataset Split (80:20)
AQI Prediction Model & Validation
Test Set
Machine Learning Regression Model

DIAGRAM:



A schematic diagram, which illustrates some of the main factors in the evaluation of the exposure and health impacts of particulate matter.

CHAPTER 2
PHASE 2

Explanation:

Air quality analysis and prediction in Tamil Nadu involves the assessment and forecasting of the state's air quality. This process helps monitor and improve the quality of the air we breathe by studying various pollutants and their levels. By analyzing historical data and using weather and pollution monitoring instruments, experts can predict air quality trends, issue warnings, and implement measures to mitigate pollution and protect public health in Tamil Nadu. Here's an explanation of the process:

1. **Monitoring Stations**: Tamil Nadu likely has a network of air quality monitoring stations strategically placed across the state. These stations continuously collect data on various air pollutants, including particulate matter (PM2.5 and PM10), nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO), ozone (O3), and volatile organic compounds (VOCs).

2. **Data Collection**: Instruments at these monitoring stations measure the concentration of pollutants in the air. Data is typically collected in real-time or at regular intervals, and it is sent to central databases for analysis.

3. **Data Analysis**: In the analysis phase, experts examine the collected data to determine the current air quality levels. They assess whether the levels of pollutants are within permissible limits set by environmental regulations.

4. **Pollution Sources Identification**: Identifying the sources of pollution is crucial. Industrial emissions, vehicular traffic, construction activities, and meteorological factors all contribute to air pollution. Understanding these sources helps in planning mitigation strategies.

5. **Meteorological Data**: Meteorological data, such as wind speed and direction, temperature, and humidity, play a vital role in predicting air quality. Changes in weather patterns can influence the dispersion and concentration of pollutants.

6. **Modeling and Prediction**: Computer models are used to predict future air quality. These models take into account current pollution levels, meteorological data, and historical patterns to forecast air quality for the coming days. These predictions can range from a few hours to several days in advance.

7. **Alerts and Warnings**: When air quality is predicted to reach unhealthy levels, alerts and warnings are issued to the public. This information helps individuals take precautions, such as reducing outdoor activities or using masks.

8. **Policy and Regulation**: The government can use the data and predictions to formulate policies and regulations aimed at reducing air pollution. This might include stricter emission standards, promoting cleaner fuels, and incentivizing public transportation.

9. **Public Awareness**: Public awareness campaigns are important in encouraging people to adopt cleaner practices and reduce their contribution to air pollution.

10. **Continuous Monitoring**: The process is continuous, as air quality can change rapidly. Regular updates and ongoing monitoring ensure that interventions can be adjusted as needed.

https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil-nadu-year-2014

Where I got the dataset and its details:

You can find datasets for customer segmentation and various other data science projects from several reputable sources.

tn.data.gov.in: tn is a popular platform for data science competitions and dataset sharing. It hosts a wide range of datasets on various topics, including customer data. You can browse datasets, read their descriptions, and download them for free.

Information about the columns:

The information you've provided appears to be related to air quality monitoring data, possibly from a government or environmental agency. Here's an explanation of the columns:

1. **Stn Code**: This is the station code or identifier for the monitoring station where the air quality data was collected.

2. **Sampling Date**: The date on which the air quality data was collected or sampled.

3. **State**: The state in which the monitoring station is located.

4. **City/Town/Village/Area**: The specific location within the state where the monitoring station is situated, which could be a city, town, village, or a specific area.

5. **Location** of **Monitoring Station**: Details about the precise location or coordinates of the monitoring station, often provided as latitude and longitude.

6. **Agency**: The organization or agency responsible for conducting the air quality monitoring at this station. It could be a government agency or an environmental monitoring group.

7. **Type of Location**: This indicates the type of area where the monitoring station is located, such as urban, rural, industrial, residential, etc.

8. **SO2** (**Sulfur Dioxide**): The concentration of sulfur dioxide gas measured in the air. SO2 is a common air pollutant emitted from industrial processes and vehicle exhaust.

9. **NO2** (**Nitrogen Dioxide**): The concentration of nitrogen dioxide gas measured in the air. NO2 is another common air pollutant often associated with combustion processes.

10. **RSPM/PM10** (**Respirable Suspended Particulate Matter/Particulate Matter 10**): These columns likely represent the concentration of particulate matter in the air, specifically particles with a diameter of 10 micrometers or less. RSPM may include finer particles that can deeply penetrate the respiratory system.

11. **PM 2.5** (**Particulate Matter 2.5**): This represents the concentration of even finer particulate matter with a diameter of 2.5 micrometers or less. PM2.5 is of particular concern as it can deeply penetrate the lungs and pose health risks.

These columns provide crucial information for assessing air quality, tracking pollution levels, and monitoring the impact of various sources on air pollution in different locations. Monitoring and analyzing this data helps in making informed decisions for environmental and public health purposes.

Details of library to be used and way to download libraries to be used:

When working on the analysis and prediction of air quality, you can use various libraries and modules in programming languages like Python. Here are some commonly used ones:

1. **Pandas:** For data manipulation, cleaning, and analysis, as it provides powerful data structures and data analysis tools.

Syntax:

import pandas as pd

2. **NumPy**: Essential for numerical operations and efficient array handling.

Syntax:

import numpy as np

3. **Matplotlib and Seaborn**: Used for creating visualizations and plots to understand and present air quality data.

Syntax:

import matplotlib.pyplot as plt

import seaborn as sns

4.**sklearn**: provides a wide range of tools and algorithms for tasks like classification, regression, clustering.

Syntax:

`from` sklearn import metrics

way to download the libraries:

```
1. Click the python packages in the bottom of your project in pycharm
```


```
2. Type the required library in the search box and click install package in
   the right end top of the python packages.
```

3. After installation process finished it shows the package was installed in the python packages.

How to train and test the dataset:

Training and testing a dataset for air quality analysis and prediction in Tamil Nadu involves several steps. Here's a high-level overview of the process:

1. **Data Collection**:
   - Gather historical air quality data for Tamil Nadu. This data should include variables like PM2.5, PM10, NO2, SO2, CO, O3, temperature, humidity, wind speed, and wind direction. You can obtain this data from government agencies, research institutions, or online repositories.

2. **Data Preprocessing**:
   - Clean the data by handling missing values, outliers, and duplicates.
   - Convert timestamps into a suitable format for time series analysis.
   - Normalize or scale the data to ensure that all features have the same scale.

3. Feature Engineering:
   - Create relevant features like rolling averages, lag features, and time-related features to capture seasonality and trends.

4. **Data Splitting**:
   - Split your dataset into training and testing sets. A common split is 70-30 or 80-20, where the larger portion is used for training.

5. **Model Selection**:
   - Choose appropriate machine learning or deep learning models for air quality prediction. Time series forecasting models like ARIMA, SARIMA, or machine learning models like Random Forest, XGBoost, or deep learning models like LSTM or GRU can be considered.

6. **Model Training**:
   - Train your selected model using the training dataset. This involves feeding the historical data into the model and adjusting its parameters.

7. **Model Evaluation**:
   - Use the testing dataset to evaluate the model's performance. Common evaluation metrics for regression tasks include Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared.

8. **Hyperparameter Tuning**:
   - Fine-tune your model's hyperparameters to improve its performance. You can use techniques like grid search or random search for this.

9. **Model Deployment**:
   - Once you're satisfied with the model's performance, you can deploy it for real-time air quality prediction. This might involve setting up a web application or API for users to access predictions.

10. Monitoring and Maintenance:
   - Continuously monitor the model's performance in a production environment and retrain it periodically with new data to keep it up-to-date.

11. **Visualize Results**:
   - Create visualizations and reports to communicate the air quality predictions to stakeholders and the public effectively.

Rest of explanation:

1. **Data Collection**: The process begins with the collection of various data sources, including real-time data from air quality monitoring stations, satellite data, and weather data. These monitoring stations are strategically placed across Tamil Nadu to ensure comprehensive coverage.

2. **Pollutant Measurement**: The monitoring stations measure various air pollutants such as particulate matter (PM2.5 and PM10), nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO), ozone (O3), and volatile organic compounds (VOCs). These measurements provide a snapshot of the current air quality.

3. **Data Analysis**: Collected data is then analyzed to assess the current air quality status. It involves calculating air quality indices (AQI) for different locations in Tamil Nadu. AQI is a standardized measure that provides an easy-to-understand assessment of air quality, making it accessible to the public.

4. **Weather Data Integration**: Weather data, including temperature, humidity, wind speed, and direction, is integrated into the analysis. Weather conditions have a significant impact on air quality, so this information is crucial for accurate predictions.

5. **Modeling and Prediction**: Sophisticated computer models, often based on machine learning and statistical techniques, are used to predict future air quality. These models take into account historical data, current measurements, and weather forecasts to make predictions for the coming days.

6. **Alerts and Public Awareness**: When air quality is predicted to deteriorate to unhealthy levels, alerts are issued to the public through various channels, including mobile apps, websites, and local media. These alerts provide recommendations for action, such as limiting outdoor activities or using masks when necessary.

7. **Government Intervention**: In severe cases, the government may implement measures such as temporary restrictions on industrial activities, construction, or vehicular movement to mitigate air pollution and protect public health.

8. **Continuous Monitoring**: Monitoring stations continuously collect data, ensuring that air quality remains under surveillance. This information helps authorities take immediate actions in response to sudden changes in air quality.

9. **Long-term Planning**: Data analysis and predictions also play a crucial role in long-term planning. They assist in identifying pollution sources and developing policies and regulations to address the root causes of air pollution.

10. **Research and Development**: Researchers and environmental agencies in Tamil Nadu work on improving air quality prediction models and monitoring technologies to enhance the accuracy of forecasts and better protect the health of the population.

In conclusion, air quality analysis and prediction in Tamil Nadu involve a comprehensive process that combines data collection, analysis, modeling, and public awareness to monitor and manage air quality, with the ultimate goal of safeguarding the health and well-being of the residents.

What metrics used for the accuracy checking:

In air quality analysis and prediction in Tamil Nadu, several metrics are commonly used to assess the accuracy of air quality forecasts and measurements. These metrics help evaluate how well the models and monitoring stations perform. Here are some key metrics:

1. **Mean Absolute Error** (MAE): MAE measures the average absolute difference between predicted and observed air quality values. It gives an indication of the magnitude of errors in predictions.

2. **Root Mean Square Error** (RMSE): RMSE is similar to MAE but penalizes larger errors more heavily. It provides a measure of the overall error in the predictions.

3. **Error Mean Bias** (MBE): MBE calculates the average difference between predicted and observed values. Positive MBE indicates overprediction, while negative MBE indicates underprediction.

4. **Coefficient of Determination** (R^2): R-squared measures the proportion of the variance in the observed data that is explained by the model. A higher R^2 indicates a better fit between predicted and observed values.

5. **Pearson Correlation Coefficient** (r): This metric assesses the linear relationship between predicted and observed values. A high positive correlation indicates that the model captures the trends well.

6. **Fractional Bias** (FB): FB quantifies the relative bias in predictions. It indicates whether the model tends to consistently overestimate or underestimate air quality values.

7. **Index of Agreement** (IOA): IOA measures the agreement between predicted and observed values. It considers both the bias and variance of predictions.

8. **Percentage of Correct Predictions**: This metric calculates the percentage of predictions that fall within specified air quality categories (e.g., good, moderate, unhealthy). It assesses the model's ability to correctly classify air quality conditions.

9. **Normalized Mean Bias** (NMB): NMB is a bias metric that normalizes the mean bias by dividing it by the mean of observed values. It provides a relative measure of bias.

10. **Normalized Mean Error** (NME): Similar to NMB, NME normalizes the mean error by dividing it by the mean of observed values. It helps evaluate the relative magnitude of errors.

These metrics are used to assess the performance of air quality prediction models and the accuracy of monitoring stations in Tamil Nadu. It's essential to use a combination of these metrics to gain a comprehensive understanding of how well the system is functioning and to identify areas for improvement in air quality analysis and prediction.

CHAPTER 3
PHASE 3

EXPLANATION:

Analyzing and predicting air quality in Tamil Nadu, like in any other region, involves a combination of data collection, monitoring, and modeling. Here's a detailed explanation of the process:

Data Collection:

Monitoring Stations: Establish a network of air quality monitoring stations across Tamil Nadu. These stations measure various air pollutants, including particulate matter (PM2.5, PM10), nitrogen dioxide (NO_2), sulfur dioxide (SO_2), carbon monoxide (CO), ozone (O_3), and volatile organic compounds (VOCs).

Meteorological Data:

Gather meteorological data such as temperature, humidity, wind speed, and wind direction, which can influence air quality.

Data Analysis:

Historical Data: Analyze historical air quality data to identify trends and patterns. This can help in understanding seasonal variations and the impact of pollution sources.

Correlation Analysis:

Determine relationships between meteorological conditions and air quality. For example, how wind patterns affect the dispersion of pollutants.

Air Quality Index (AQI):

Calculate the AQI based on the concentration of different pollutants. The AQI provides a simple way to communicate air quality to the public.

Source Identification:

Identify major sources of pollution in Tamil Nadu, such as industries, vehicular emissions, construction activities, and agricultural practices.

Modeling:

Develop air quality models that use historical data, meteorological data, and source information to predict future air quality. Common models include the Community Multiscale Air Quality (CMAQ) model and the Weather Research and Forecasting with Chemistry (WRF-Chem) model.

Machine Learning:

Utilize machine learning algorithms to improve prediction accuracy. Machine learning can help in understanding complex patterns in air quality data and making real-time predictions.

Public Awareness:

Communicate air quality information to the public through websites, mobile apps, and public announcements. Provide health advisories based on the AQI to protect vulnerable populations.

Policy and Mitigation:

Use the information to formulate and enforce air quality regulations. Implement measures to reduce emissions from major pollution sources.

Validation:

Continuously validate the accuracy of predictions by comparing them to real-time monitoring data. Adjust models and predictions as needed.

Emergency Response:

Develop contingency plans for extreme pollution events, such as smog or wildfires, to protect public health.

Research and Development:

Invest in research to improve monitoring technology, modeling techniques, and pollution control measures.

Collaboration:

Collaborate with neighboring states and countries, as air quality is not confined by political boundaries.

The success of air quality analysis and prediction in Tamil Nadu relies on the integration of data, technology, and proactive policies to reduce pollution and protect public health. It's an ongoing process that requires continuous monitoring and adaptation to changing conditions and emission sources.

Dataset explanation:

A typical air quality dataset includes parameters like:

1. **Particulate Matter (PM2.5 and PM10):**
   Fine particles in the air that can have adverse health effects.

2. **Ozone (O3):**
   Ground-level ozone, which can cause respiratory problems.

3. **Nitrogen Dioxide (NO2) and Sulfur Dioxide (SO2):**

Gases produced by combustion processes.

4. **Carbon Monoxide (CO):**
   A colorless, odorless gas that can be harmful when inhaled.

5. **Weather Data:**
   Temperature, humidity, wind speed, and direction, which influence air quality.

6. **Geographic Coordinates:**
   Information about monitoring station locations.

Air quality prediction typically involves using machine learning models to forecast future air quality levels based on historical data. This can help in air quality management and alert systems.

Begin building the project by loading the dataset:

The columns involved in the dataset are

- Stn_code
- Sampling_data
- State
- Location
- Agency
- Type
- So2
- No2
- Rspm
- Spm
- Location_monitoring_station
- Pm2_5
- Date

To begin building the air quality analysis and prediction in tamilnadu project using the dataset from Kaggle, you'll need to load the dataset and start exploring it. You can use Python and popular libraries like Pandas for data manipulation and Matplotlib or Seaborn for data visualization. Here's a step-by-step guide:

Importing necessary libraries:

```
import numpy as np
import pandas as pd
```

```
import matplotlib.pyplot as plt
import seaborn as sns
```

loading the dataset:

```
data1=pd.read_csv('C:/project/data.csv',encoding="ISO-8859-1")
data=data1.sample(500)
print(data)
```

Output:

stn_code sampling_date ... pm2_5 date
128268 NaN 9/5/2009 ... NaN 2009-05-09
119673 35 19/01/2012 ... NaN 2012-01-19
433064 162 24-12-14 ... NaN 2014-12-24
220657 508.0 13-12-10 ... NaN 2010-12-13
35036 538.0 2/10/2012 ... NaN 2012-10-02
... ...
302214 NaN 21/2/2009 ... NaN 2009-02-21
356914 766.0 17-05-13 ... NaN 2013-05-17
326928 NaN 18-05-07 ... NaN 2007-05-18
428065 9 13/07/2011 ... NaN 2011-07-13
274319 NaN 30-08-07 ... NaN 2007-08-30
[500 rows x 13 columns]

preprocessing the data:

1. Data Collection:

First, collect air quality data for Tamil Nadu from reliable sources like government agencies or research organizations.

2. Data Cleaning:

- Remove duplicates, missing values, and outliers.
- Convert data types, ifneeded.

```
# Remove duplicates
data= data.drop_duplicates()
print(data)
#Handle missing values
data= data.dropna()
print(data)

# Handle outliers (you can define your own outlier detection method)
from scipy import stats

z_scores = stats.zscore(data['PM2.5'])
```

```
data = data[(z_scores < 3)]
print(data)
```

3. Data Transformation:

- Convert date/time columns to datetime objects.
- Extract features like year, month, day, and hour for time series analysis.

```
# Convert 'Date' column to datetime format
data['Date'] = pd.to_datetime(data['Date'])
print(data)
```

4. Feature Engineering:

- Create new features or modify existing ones if they can improve model performance.

```
#future engineering
data['Year'] = data['Date'].dt.year
data['Month'] = data['Date'].dt.month
```

5. Normalization/Scaling:

- Normalize or scale features, especially if you're using algorithms sensitive to feature scales.

6. Outliers detection and handling:

- Identify and handle outliers in numerical columns if necessary. You can use methods like Z-score, IQR, or domain-specific rules.

```
# Step 6
: Outlier Detection and Handling (example: using Z-score)
from scipy import stats
z_scores = stats.zscore(data[['SO2', 'NO2', 'RSPM', 'SPM', 'PM2_5']])
data = data[(z_scores < 3).all(axis=1)]  # Remove outliers
```

7. Splitting the Dataset:

- Split the dataset into training, validation, and test sets for model evaluation.

```
X = data.drop('Target_Column', axis=1)  # Replace 'Target_Column' with your
target variable
y = data['Target_Column']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

8. Save Preprocessed Data:

- Save the preprocessed data to a new file for future analysis to avoid repeating these steps.

```
# Step 8: Save Preprocessed Data
data.to_csv('preprocessed_data.csv', index=False)
```

Performing data analysis :

```
data.fillna(0, inplace=True)
data.head()
#Function to calculate so2 individual pollutant index(si)
def calculate_si(so2):
    si=0
    if (so2<=40):
     si= so2*(50/40)
    if (so2>40 and so2<=80):
     si= 50+(so2-40)*(50/40)
    if (so2>80 and so2<=380):
     si= 100+(so2-80)*(100/300)
    if (so2>380 and so2<=800):
     si= 200+(so2-380)*(100/800)
    if (so2>800 and so2<=1600):
     si= 300+(so2-800)*(100/800)
    if (so2>1600):
     si= 400+(so2-1600)*(100/800)
    return si
data['si']=data['so2'].apply(calculate_si)
df= data[['so2','si']]
df.head()
def calculate_ni(no2):
    ni=0
    if(no2<=40):
     ni= no2*50/40
    elif(no2>40 and no2<=80):
     ni= 50+(no2-14)*(50/40)
    elif(no2>80 and no2<=180):
     ni= 100+(no2-80)*(100/100)
    elif(no2>180 and no2<=280):
     ni= 200+(no2-180)*(100/100)
    elif(no2>280 and no2<=400):
     ni= 300+(no2-280)*(100/120)
    else:
     ni= 400+(no2-400)*(100/120)
    return ni
data['ni']=data['no2'].apply(calculate_ni)
df= data[['no2','ni']]
df.head()
def calculate_(rspm):
    rpi=0 if(rpi<=30):
     rpi=rpi*50/30
    elif(rpi>30 and rpi<=60):
     rpi=50+(rpi-30)*50/30
    elif(rpi>60 and rpi<=90):
     rpi=100+(rpi-60)*100/30
    elif(rpi>90 and rpi<=120):
     rpi=200+(rpi-90)*100/30
    elif(rpi>120 and rpi<=250):
     rpi=300+(rpi-120)*(100/130)
```

```
    else:
     rpi=400+(rpi-250)*(100/130)
    return rpi
data['rpi']=data['rspm'].apply(calculate_si)
df= data[['rspm','rpi']]
df.tail()
#many data values of rspm values is unawailable since it was not measure
before
#Function to calculate no2 individual pollutant index(spi)
def calculate_spi(spm):
    spi=0
    if(spm<=50):
     spi=spm
    if(spm<50 and spm<=100):
     spi=spm
    elif(spm>100 and spm<=250):
     spi= 100+(spm-100)*(100/150)
    elif(spm>250 and spm<=350):
     spi=200+(spm-250)
    elif(spm>350 and spm<=450):
     spi=300+(spm-350)*(100/80)
    else:
     spi=400+(spm-430)*(100/80)
    return spi
data['spi']=data['spm'].apply(calculate_spi)
df= data[['spm','spi']]
df.tail()
#many data values of rspm values is unawailable since it was not measure
before
#function to calculate the air quality index (AQI) of every data value
#its is calculated as per indian govt standards
def calculate_aqi(si,ni,spi,rpi):
    aqi=0
    if(si>ni and si>spi and si>rpi):
     aqi=si
    if(spi>si and spi>ni and spi>rpi):
     aqi=spi
    if(ni>si and ni>spi and ni>rpi):
     aqi=ni
    if(rpi>si and rpi>ni and rpi>spi):
     aqi=rpi
    return aqi
data['AQI']=data.apply(lambda
x:calculate_aqi(x['si'],x['ni'],x['spi'],x['rpi']),axis=1)
df= data[['sampling_date','state','si','ni','rpi','spi','AQI']]
df.head()
df.state.unique()
state=pd.read_csv("../input/indian-states-lat-lon/lat.csv")
state.head()
df.head()
dff=pd.merge(state.set_index("state"),df.set_index("state"),
right_index=True, left_index=True).reset_index()
dff.head()
from mpl_toolkits.basemap import Basemap
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

```
%config InlineBackend.figure_format = 'retina'
m = Basemap(projection='mill',llcrnrlat=5,urcrnrlat=40,
llcrnrlon=60,urcrnrlon=110,lat_ts=20,resolution='c')
longitudes = dff["lon"].tolist()
latitudes = dff["lat"].tolist()
#m = Basemap(width=12000000,height=9000000,projection='lcc',
            #resolution=None,lat_1=80.,lat_2=55,lat_0=80,lon_0=-107.)
x,y =  m(longitudes,latitudes)
fig = plt.figure(figsize=(12,10))
plt.title("All affected areas")
m.plot(x, y, "o", markersize = 3, color = 'blue')
m.drawcoastlines()
m.fillcontinents(color='coral',lake_color='aqua')
m.drawmapboundary()
m.drawcountries()
plt.show()
#Visualization of AQI across india

data['date'] = pd.to_datetime(data['date'],format='%Y-%m-%d') # date parse
data['year'] = data['date'].dt.year # year
data['year'] = data['year'].fillna(0.0).astype(int)
data = data[(data['year']>0)]

df =
data[['AQI','year','state']].groupby(["year"]).median().reset_index().sort_va
lues(by='year',ascending=False)
f,ax=plt.subplots(figsize=(15,10))
sns.pointplot(x='year', y='AQI', data=df)
#setting up date parameter
import warnings
import itertools
import dateutil
import statsmodels.api as sm
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import seaborn as sns
%matplotlib inline
df=data[['AQI','date']]
df["date"] = pd.to_datetime(df['date'])
df.tail(20)
#Calculating the yearly mean for the data
df=df.set_index('date').resample('M')["AQI"].mean()
df.head()
#preprocessing the data values
data=df.reset_index(level=0, inplace=False)
data = data[np.isfinite(data['AQI'])]
data=data[data.date != '1970-01-31']
data = data.reset_index(drop=True)
data.head()
#visualizing the processed data of AQI
df=data.set_index('date')
df.sort_values(by='date',ascending=False)
df.plot(figsize=(15, 6))
plt.show()
y=df.AQI
#exctracting knowledge about data
#spliting dataframes into test and train
```

```
n = df.shape[0]
train_size = 0.65
features_dataframe = df.sort_values('date')
train = df.iloc[:int(n * train_size)]
test = df.iloc[int(n * train_size):]
#plotting the yearly variations of AQI
train.AQI.plot(figsize=(15,8), title= 'YEARLY VARIATIONS', fontsize=14)
test.AQI.plot(figsize=(15,8), title= 'YEARLY VARIATIONS', fontsize=14)
plt.show()
#Naive Forecast Approach to find the variations(trend)

dd= np.asarray(train.AQI)
y_hat = test.copy()
y_hat['naive'] = dd[len(dd)-1]
plt.figure(figsize=(12,8))
plt.plot(train.index, train['AQI'], label='Train')
plt.plot(test.index,test['AQI'], label='Test')
plt.plot(y_hat.index,y_hat['naive'], label='Naive Forecast')
plt.legend(loc='best')
plt.title("Naive Forecast",fontsize=20)

plt.legend(["actual ","predicted"])
plt.xlabel("YEAR",fontsize=20)
plt.ylabel("AQI",fontsize=20)
plt.tick_params(labelsize=20)
plt.show()
#various statmodel to identity huge variations od data values
import statsmodels.api as sm
train.index=pd.DatetimeIndex(freq="w", start=0 ,periods=224)

sm.tsa.seasonal_decompose(train.AQI).plot()
result = sm.tsa.stattools.adfuller(train.AQI)
plt.show()
#resampling the data to predict monthly AQI of india
df=data[['AQI','date']]
df['date']=pd.to_datetime(df['date'])
date=df.groupby(pd.Grouper(key='date',freq='1MS'))["AQI"].mean()
df.count()
#splitting the sampling date into month and year accordingly
data['month'] = data['date'].dt.month
data['year'] = data['date'].dt.year
data=data[['AQI','date','month','year']]
data.head()
#Appling BOXPLOT analysis
df =
data[['AQI','year']].groupby(["year"]).mean().reset_index().sort_values(by='y
ear',ascending=False)
df=df.dropna()
dd=df
df.describe()
import seaborn as sns
sns.boxplot(x=df['AQI'])
#removing Outliers
df = df[np.isfinite(df['AQI'])]
df=df[df.AQI >153]
df=df[df.AQI <221]
#visualizing the filttered data
```

```
year=df['year'].values
AQI=df['AQI'].values
df['AQI']=pd.to_numeric(df['AQI'],errors='coerce')
df['year']=pd.to_numeric(df['year'],errors='coerce')

import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (20.0, 10.0)
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(year,AQI, color='red')
plt.show()
#scatter plot of data points
cols =['year']
y = df['AQI']
x=df[cols]

plt.scatter(x,y)
plt.show()
# Predicted val
newB = [200.17, -1.54]


def rmse(y, y_pred):
    rmse = np.sqrt(sum(y - y_pred))
    return rmse


y_pred = x.dot(newB)

dt = pd.DataFrame({'Actual': y, 'Predicted': y_pred})
x = pd.concat([df, dt], axis=1)
x
x
#calculating the root mean squared error for the predicted AQi values
from sklearn import metrics
print(np.sqrt(metrics.mean_squared_error(y,y_pred)))
x_axis=x.year
y_axis=x.Actual
y1_axis=x.Predicted
plt.plot(x_axis,y_axis)
plt.plot(x_axis,y1_axis)
plt.title("Actual vs Predicted",fontsize=20)
plt.legend(["actual ","predicted"])
plt.xlabel("YEAR",fontsize=20)
plt.ylabel("AQI",fontsize=20)
plt.tick_params(labelsize=20)
plt.show()
#applying boxplot analysis
import seaborn as sns
sns.boxplot(x=df['AQI'])
#plotting data points
cols =['year']
y = df['AQI']
x=df[cols]
```

```
plt.scatter(x,y)
plt.show()
#testing the accuracy of the model

from sklearn import metrics
print(np.sqrt(metrics.mean_squared_error(y,y_pred)))
#plotting the actual and predicted results
x_axis=x.year
y_axis=x.Actual
y1_axis=x.Predicted
plt.plot(x_axis,y_axis)
plt.plot(x_axis,y1_axis)
plt.title("Actual vs Predicted",fontsize=20)
plt.legend(["actual ","predicted"])
plt.xlabel("YEAR",fontsize=20)
plt.ylabel("AQI",fontsize=20)
plt.tick_params(labelsize=20)
plt.show()
#Prediction for the future
from sklearn.preprocessing import MinMaxScaler
#feeding in the x value-years
data=[[-1,2016],[-1,2017],[-1,2018],[-1,2019],[-1,2020]]
#normalization
scaler=MinMaxScaler(feature_range=(-1,1))
scaler.fit(data)
x=scaler.transform(data)
#calculations
newB=[103.59,-2.74]
ypred=-(x.dot(newB))
#AQI for the year 2020
print("AQI for the year 2020===>",ypred[-1])
```

PHASE 4

Program:

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt

# Load the air quality data into a DataFrame
dtype = {
     'stn_code': str,
    'sampling_date': str,
    'state': str,
    'location': str,
    'agency':str,
    'type':str,
    'so2':float,
    'no2':float,
    'rspm':float,
    'spm':float,
    'location_monitoring_station':str,
    'date':str
}
data= pd.read_csv('C:/project/data.csv', sep=',', encoding='ISO-8859-1',
dtype=dtype)
data=data.sample(500)
print(data)
# Calculate average levels
average_so2 = data['so2'].mean()
average_no2 = data['no2'].mean()
average_rspm = data['rspm'].mean()

# Identify pollution trends
# You may want to add more analysis here

# Create visualizations
plt.figure(figsize=(10, 6))
plt.bar(['so2', 'no2', 'rspm'], [average_so2, average_no2, average_rspm])
plt.xlabel('Pollutant')
plt.ylabel('Average Level')
plt.title('Average Air Quality Levels')
plt.show()
# Create a bar chart to visualize the averages
pollutants = ['SO2', 'NO2', 'RSPM']
averages = [average_so2,average_no2 , average_rspm]
plt.bar(pollutants, averages)
plt.xlabel('Pollutant')
plt.ylabel('Average Level')
plt.title('Average Air Quality Levels')
plt.show()
# Calculate average levels for SO2, NO2, and RSPM/PM10
average_so2 = data['so2'].mean()
average_no2 = data['no2'].mean()
average_rspm_pm10 = data['rspm'].mean()

print(f"Average SO2 level: {average_so2}")
print(f"Average NO2 level: {average_no2}")
print(f"Average RSPM/PM10 level: {average_rspm}")
```

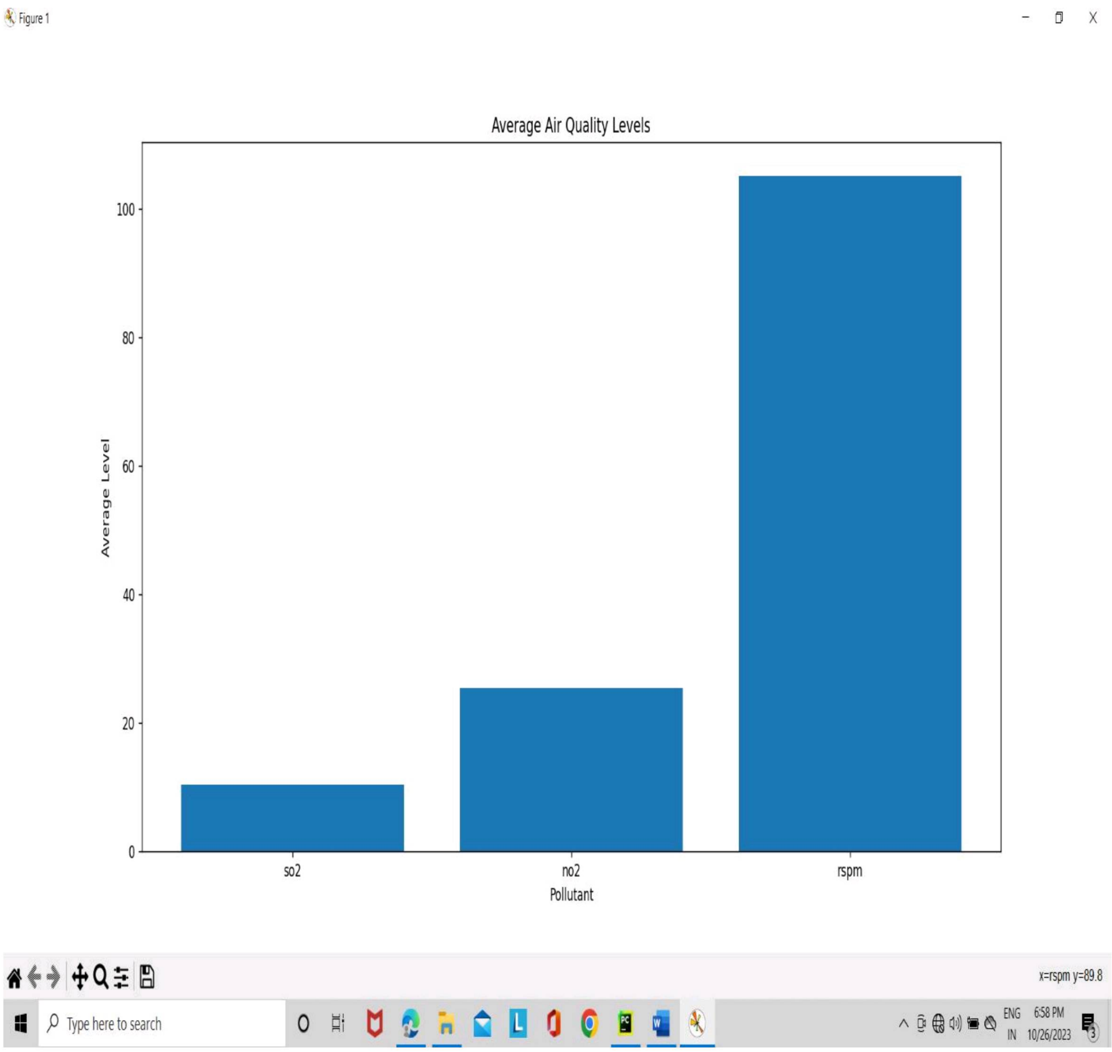
```
plt.figure(figsize=(12, 6))
plt.plot(data['sampling_date'], data['so2'], label='SO2 Levels',
color='blue')
plt.xlabel('sampling_date')
plt.ylabel('SO2 Levels')
plt.title('SO2 Levels in Tamil Nadu')
plt.legend()
plt.grid()
plt.show()
plt.figure(figsize=(12, 6))
plt.plot(data['sampling_date'], data['no2'], label='no2 Levels',
color='violet')
plt.xlabel('sampling_date')
plt.ylabel('no2 Levels')
plt.title('no2 Levels in Tamil Nadu')
plt.legend()
plt.grid()
plt.show()
plt.figure(figsize=(12, 6))
plt.plot(data['sampling_date'], data['rspm'], label='rspm Levels',
color='black')
plt.xlabel('sampling_date')
plt.ylabel('rspm Levels')
plt.title('rspm Levels in Tamil Nadu')
plt.legend()
plt.grid()
plt.show()

import seaborn as sns
# Basic histograms for SO2, NO2, and RSPM/PM10
plt.figure(figsize=(12, 6))
plt.subplot(131)
sns.histplot(data['so2'], bins=20, kde=True)
plt.title('SO2 Distribution')

plt.subplot(132)
sns.histplot(data['no2'], bins=20, kde=True)
plt.title('NO2 Distribution')

plt.subplot(133)
sns.histplot(data['rspm'], bins=20, kde=True)
plt.title('RSPM/PM10 Distribution')
plt.tight_layout()
plt.show()

# Calculate average levels for SO2, NO2, and RSPM/PM10
average_so2 = data.groupby('location_monitoring_station')['so2'].mean()
average_no2 = data.groupby('location_monitoring_station')['no2'].mean()
average_rspm_pm10 =
data.groupby('location_monitoring_station')['rspm'].mean()

# Create visualizations
# Example: Bar chart to visualize average SO2 levels by city
plt.figure(figsize=(10, 6))
plt.bar(average_so2.index, average_so2.values)
plt.xlabel('City')
plt.ylabel('Average SO2 Level')
```

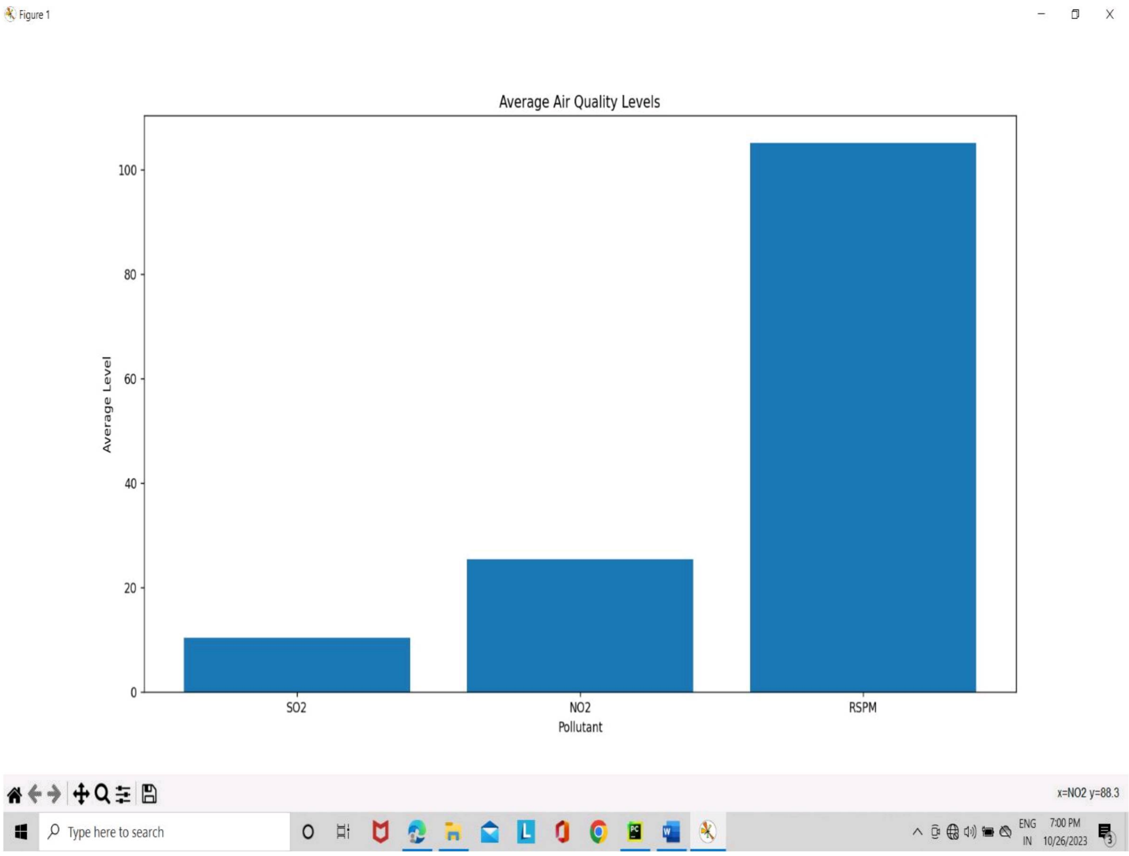
```
plt.title('Average SO2 Levels by City')
plt.xticks(rotation=45)
plt.show()

# Calculate average SO2, NO2, and RSPM/PM10 levels
avg_so2 = data.groupby('location_monitoring_station')['so2'].mean()
avg_no2 = data.groupby('location_monitoring_station')['no2'].mean()
avg_rspm_pm10 = data.groupby('location_monitoring_station')['rspm'].mean()

# Identify areas with high pollution levels
high_so2_areas = avg_so2[avg_so2 > 2]  # Replace 'threshold' with your
desired threshold value
high_no2_areas = avg_no2[avg_no2 > 200]
high_rspm_pm10_areas = avg_rspm_pm10[avg_rspm_pm10 >60 ]
# Print or visualize the results
print("Average SO2 levels by location:")
print(avg_so2)
print("\nAverage NO2 levels by location:")
print(avg_no2)
print("\nAverage RSPM/PM10 levels by location:")
print(avg_rspm_pm10)

print("\nAreas with high SO2 levels:")
print(high_so2_areas)
print("\nAreas with high NO2 levels:")
print(high_no2_areas)
print("\nAreas with high RSPM/PM10 levels:")
print(high_rspm_pm10_areas)

# Visualize pollution trends over time
import matplotlib.pyplot as plt
plt.figure(figsize=(12, 6))
plt.plot(high_so2_areas.index, high_so2_areas.values, label='SO2')
plt.plot(high_no2_areas.index, high_no2_areas.values, label='NO2')
plt.plot(high_rspm_pm10_areas.index,high_rspm_pm10_areas.values,
label='RSPM/PM10')
plt.xlabel('Time')
plt.ylabel('Average Levels')
plt.title('Air Quality Trends Over Time')
plt.legend()
plt.show()
```

Output:

	stn_code	sampling_date	...	Unnamed: 11	date
10031	NaN	17/4/2009	...	NaN	4/17/2009
227291	165	18/03/2011	...	NaN	3/18/2011
367294	581	1/1/2015	...	NaN	1/1/2015
21563	742	4/9/2014	...	NaN	9/4/2014
72170	144	20/10/2011	...	NaN	10/20/2011

```
...      ...        ... ...      ...      ...
351040    NaN   22/5/2009 ...      NaN  5/22/2009
395146      6   17/02/2012 ...      NaN  2/17/2012
275222    NaN   7/12/2007 ...      NaN  12/7/2007
398317      1     14-03-13 ...      NaN  3/14/2013
107544    NaN     27-10-05 ...      NaN  10/27/2005

[500 rows x 13 columns]
```





Average SO2 level: 10.347529112074236

Average NO2 level: 25.39095238093789

Average RSPM/PM10 level: 105.0656695780131

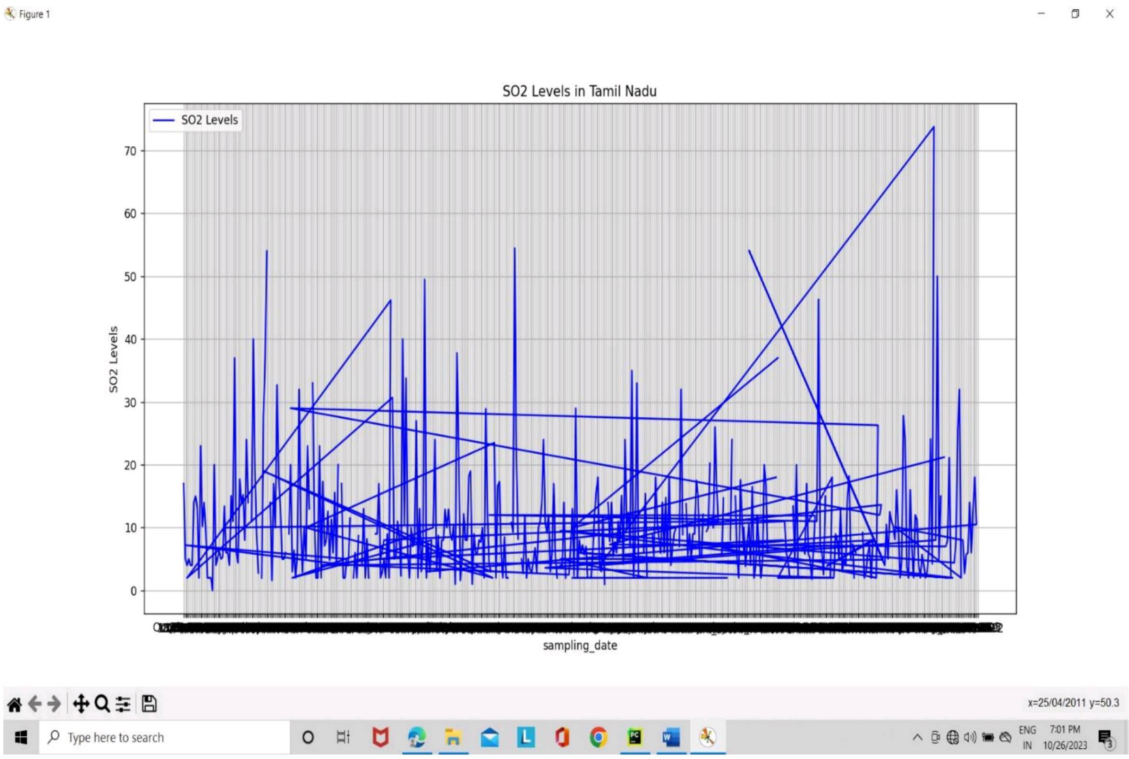


Figure 1
SO2 Levels in Tamil Nadu
SO2 Levels
70
60
50
40
30
20
10
0
SO2 Levels
sampling_date
x=25/04/2011 y=50.3
Type here to search
ENG
IN
7:01 PM
10/26/2023


Figure 1
no2 Levels in Tamil Nadu
no2 Levels
120
100
80
60
40
20
0
no2 Levels
sampling_date
x=9/12/2008 y=86.0
Type here to search
ENG
IN
7:02 PM
10/26/2023

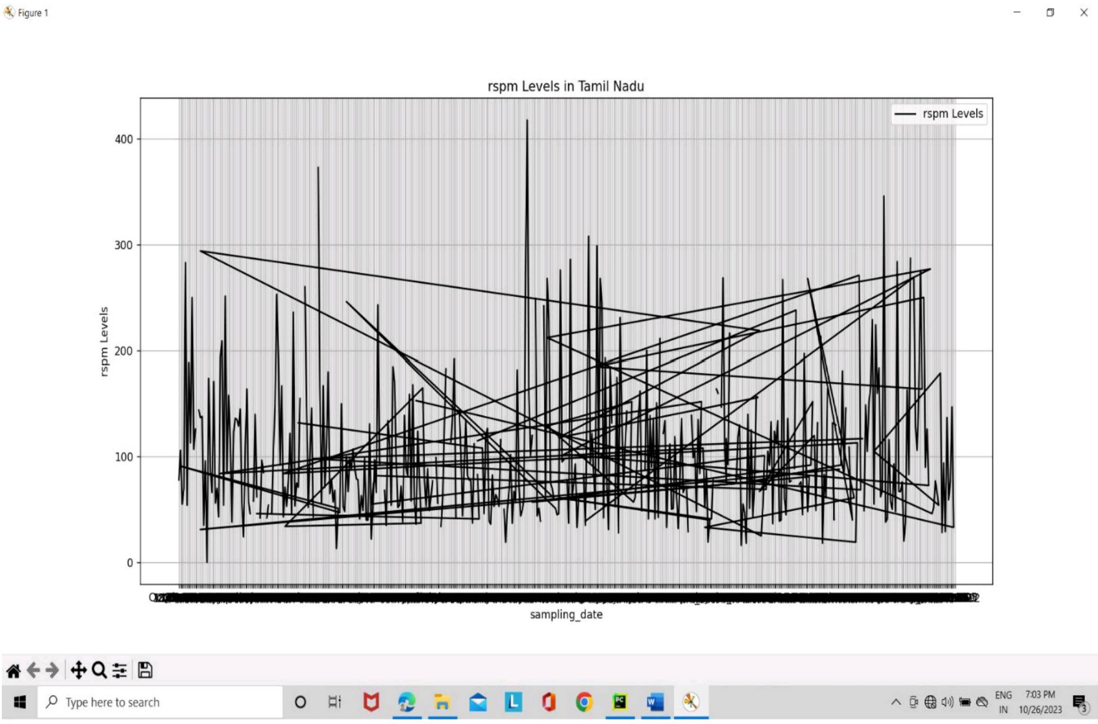


Figure 1
rspm Levels in Tamil Nadu
rspm Levels
400
300
200
100
0
rspm Levels
sampling_date
Type here to search
ENG
IN
7:03 PM
10/26/2023

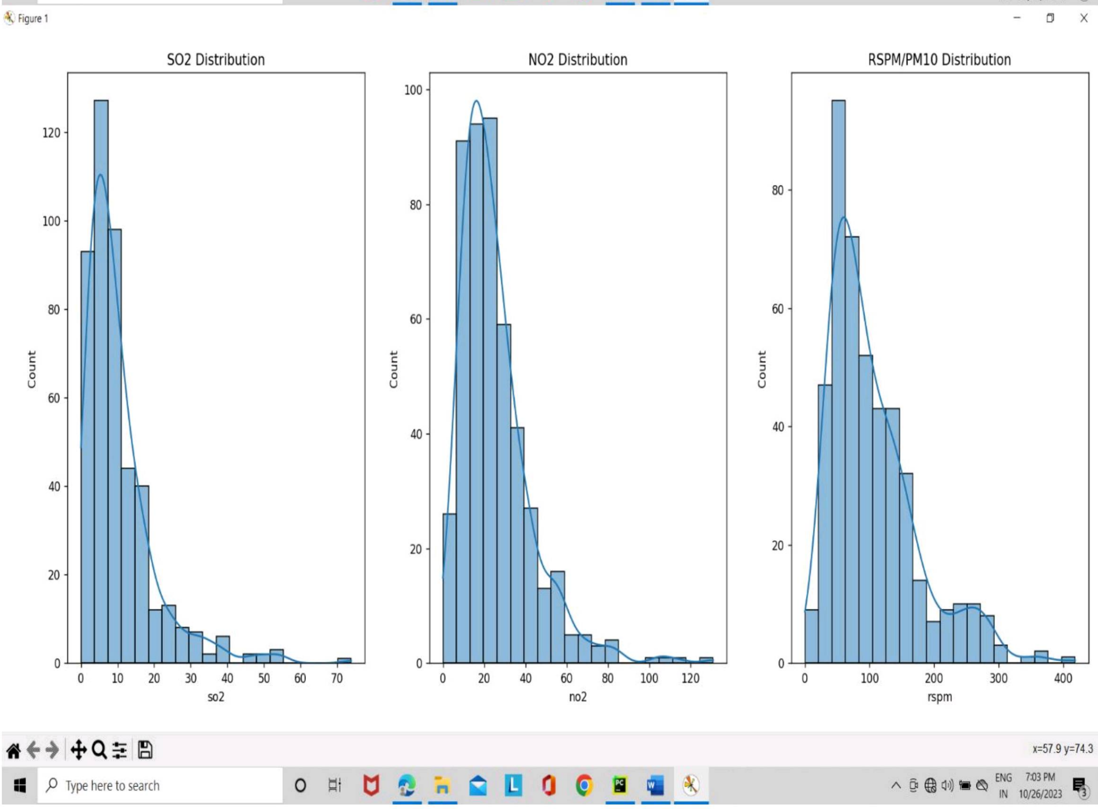
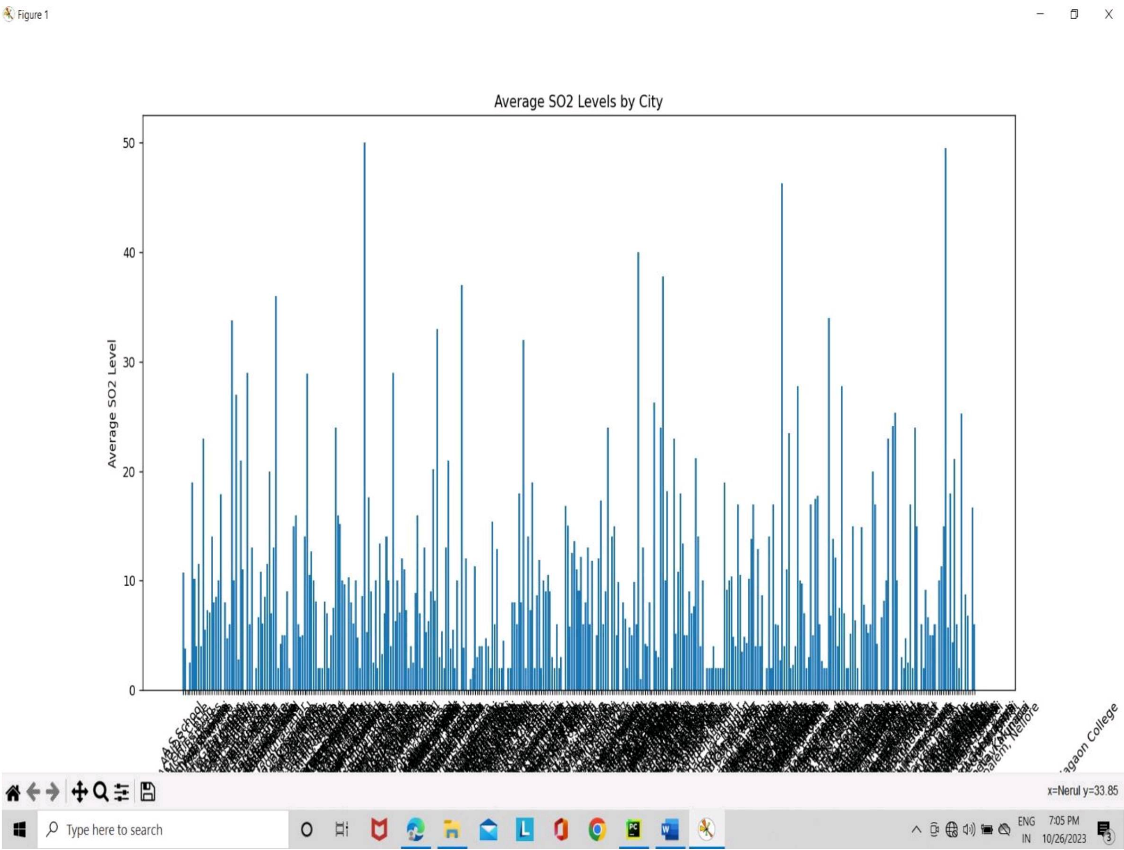


Figure 1
SO2 Distribution
NO2 Distribution
RSPM/PM10 Distribution
Count
so2
Count
no2
Count
rspm
0
10
20
30
40
50
60
70
0
20
40
60
80
100
120
0
100
200
300
400
x=57.9 y=74.3
Type here to search
ENG
IN
7:03 PM
10/26/2023



Average SO2 levels by location:

location_monitoring_station

A S School 10.75

ABIDS Circle 3.80

AC Office Bldg. NaN

AC Office Building, Parwanoo 2.50

AMCO Batteries 19.00

...

Visak Hostel, Sector-4, Bhilai 8.75

WBIIDC 6.80

WBIIDC, Haldia NaN

WIT Campus 16.70

Water Resources Division Office Campus, Christian Patty, near Nagaon College 6.00
Name: so2, Length: 359, dtype: float64

Average NO2 levels by location:
location_monitoring_station
A S School 30.35
ABIDS Circle 54.10
AC Office Bldg. 7.50
AC Office Building, Parwanoo 13.34
AMCO Batteries 37.90
...
Visak Hostel, Sector-4, Bhilai 21.25
WBIIDC 44.20
WBIIDC, Haldia 29.60
WIT Campus 36.30
Water Resources Division Office Campus, Christian Patty, near Nagaon College 21.00
Name: no2, Length: 359, dtype: float64

Average RSPM/PM10 levels by location:
location_monitoring_station
A S School 240.50
ABIDS Circle 49.00
AC Office Bldg. 50.00
AC Office Building, Parwanoo 71.00
AMCO Batteries 46.00
...
Visak Hostel, Sector-4, Bhilai 88.64
WBIIDC 33.00
WBIIDC, Haldia 73.40

```
WIT Campus                                                   121.00
Water Resources Division Office Campus, Christian Patty, near Nagaon College    194.00
Name: rspm, Length: 359, dtype: float64

Areas with high SO2 levels:
location_monitoring_station
A S School                                                 10.75
ABIDS Circle                                                3.80
AC Office Building, Parwanoo                                2.50
AMCO Batteries                                             19.00
AS School, Khanna                                          10.20
                                          ...
Victoria Hospital                                          25.30
Visak Hostel, Sector-4, Bhilai                              8.75
WBIIDC                                                      6.80
WIT Campus                                                 16.70
Water Resources Division Office Campus, Christian Patty, near Nagaon College     6.00
Name: so2, Length: 285, dtype: float64

Areas with high NO2 levels:
Series([], Name: no2, dtype: float64)

Areas with high RSPM/PM10 levels:
location_monitoring_station
A S School                                                240.50
AC Office Building, Parwanoo                               71.00
AS School, Khanna                                         229.00
ASRAM Diagnostic Center, Eluru                             88.00
AZL Behrampura, Ahmadabad                                  66.50
```

...

VijayNagar	157.00
Visak Hostel, Sector-4, Bhilai	88.64
WBIIDC, Haldia	73.40
WIT Campus	121.00
Water Resources Division Office Campus, Christian Patty, near Nagaon College	194.00

Name: rspm, Length: 247, dtype: float64