Air Q Assessment TN

Phase 3 Submission Document.

Project Title: Air Q Assessment TN

Phase 3: Development Part 1

Topic:

Begin building the project by loading and pre-processing the dataset. Begin the analysis by loading and pre-processing the air quality dataset. Load the dataset using Python and data manipulation libraries (e.g., pandas)

Project 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.

1.Project Objective:

- Analyse air quality data from monitoring stations in Tamil Nadu.
- Gain insights into air pollution trends.
- Identify areas with high pollution levels.
- Develop a predictive model to estimate RSPM/PM10 levels based on SO2 and NO2 levels.

2.Analysis Approach:

- Data Collection: Specify how you will gather air quality data, including sources, frequency, and data formats.

- Data Pre-processing: Outline steps for cleaning, handling missing data, and ensuring data quality.
- Exploratory Data Analysis (EDA): Describe techniques for exploring data to uncover trends and patterns.
- Feature Engineering: Detail how you'll create relevant features for modelling.
- Model Selection: Decide on machine learning algorithms for building the predictive model
- Evaluation Metrics: Define how you'll measure the model's performance.
- Cross-validation: Plan for validating the model's performance.
- Hyper parameter Tuning: Discuss methods for optimizing model parameters.

3.Visualization Techniques:

- Select appropriate visualization techniques based on the nature of the data:
- Line charts for time series analysis of air quality trends.
- Heat maps or spatial maps for pinpointing pollution hotspots.
- Scatter plots or correlation matrices to understand relationships between variables.
- Ensure that the visualizations are user-friendly and insightful for stakeholders.

4.Python and Libraries:

- Specify the Python libraries you plan to use for data analysis and modelling (e.g., pandas, NumPy, scikit-learn).
- Mention any specific data visualization libraries (e.g., Matplotlib, Seaborn) that will be employed.

- Include any additional libraries required for geospatial analysis if applicable.

Contents for Project Phase 3:

To incorporate machine learning algorithms to improve the accuracy of a predictive model, you can follow these steps:

Data Pre-processing:

- Collect and clean your data, handling missing values and outliers.
- Encode categorical variables and scale numerical features if necessary.

Split the Data:

- Divide your dataset into training, validation, and test sets.
- Select machine learning algorithms suitable for your problem (e.g., regression, classification, clustering).

Data source:

Dataset Link:

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

Necessary steps to follow:

- Import Libraries
- Load the Dataset
- Exploratory Data Analysis (EDA)
- Feature Engineering
- Split the Data
- Feature Scaling

1.Import Libraries:

Start by importing the necessary libraries for pre-processing and loading the data set of Air Q Assessment in Tamil Nadu in 2014

Program:

```
Import pandas as pd

Import numpy as np

From sklearn.model_selection import train_test_split

From sklearn.preprocessing import StandardScaler
```

2.Load the Dataset:

Load your dataset into a Pandas Data Frame. You can typically find house price datasets in CSV format, but you can adapt this code to other formats as needed.

Program:

```
df = pd.read_csv(‘ E:\cpcb_dly_aq_tamil_nadu_

 2014.csv ‘)

Pd.read()
```

3.Exploratory Data Analysis (EDA):

Perform EDA to understand your data better. This includes checking for missing values, exploring the data’s statistics, and visualizing it to identify patterns.

Program:

```
# Check for missing values

 Print(df.isnull().sum())

 # Explore statistics
 Print(df.describe())
# Visualize the data (e.g., histograms, scatter plots, etc.)
```

4.Feature Engineering:

Depending on your dataset, you may need to create new features or transform existing ones. This can involve one-hot encoding categorical variables, handling date/time data, or scaling numerical features.

Program:

```
 # Example: One-hot encoding for categorical variables
 Df = pd.get_dummies(df, columns=[‘ Avg. Area Income ‘,
 ‘ Avg. AreaHouse Age ‘])
```

5.Split the Data:

Split your dataset into training and testing sets. This helps you evaluate your model’s performance later.

Program:

```
 X = df.drop(‘price’, axis=1) # Features
 Y = df[‘price’] # Target variable
 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

6.Feature Scaling:

Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean=0 and std=1) is a common choice.

Program:

```
Scaler = StandardScaler() X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Importance of loading and processing dataset:

Loading and pre-processing the dataset is an important first step in building any machine learning model. However, it is especially important for air quality assessment models.

By loading and pre-processing the dataset, we can ensure that the machine learning algorithm is able to learn from the data effectively and accurately.

To load and pre-process the air quality dataset for Tamil Nadu in 2014, we can follow these steps using Python and the Pandas library.

Import the necessary libraries:

```
import pandas as pd
```

Load the dataset into a Pandas DataFrame:

```
file_path = ‘cpcb_dly_aq_tamil_nadu_-2014.csv’

df = pd.read_csv(file_path)
```

Explore the dataset to get an understanding of its structure and contents:

```
# Display the first few rows of the dataset

Print(df.head())

# Get basic statistics of the dataset

Print(df.describe())
```

Data Pre-processing:

Depending on the quality of our dataset and our analysis goals, we may need to perform various pre-processing steps, such as handling missing values, renaming columns, and converting data types. Here are some common pre-processing tasks:

Handling missing values: we can use methods like fillna() or dropna() to handle missing data.

Renaming columns: we can use rename() to give more meaningful names to columns.

Converting data types: Ensure that numerical columns are of the correct data type (e.g., float or int).

For example, to handle missing values by filling them with the mean value for each column:

```
# Fill missing values with the mean of each column

df = df.fillna(df.mean())
```

Filter the dataset for the year 2014:

```
# dataset has a ‘date’ column

df[‘date’] = pd.to_datetime(df[‘date’])

 # Convert the ‘date’ column to datetime

df_2014 = df[df[‘date’].dt.year == 2014]
```

Now, we have a Pandas Data Frame df_2014 containing the air quality data for Tamil Nadu in 2014.

Remember to adjust these steps according to the actual structure and quality of our dataset. Data pre-processing often depends on the specific characteristics of our data and the objectives of our analysis.

How to overcome the challenges of loading and pre-processing a air quality analysis in Tamil Nadu in 2014 dataset:

There are a number of things that can be done to overcome the challenges of loading and pre-processing a air quality analysis in Tamil Nadu in 2014 .

- **Use a data pre-processing library:**

There are a number of libraries available that can help with data pre-processing tasks, such as handling missing values, encoding categorical variables, and scaling the features.

- **Carefully consider the specific needs of your model:**

The best way to pre-process the data will depend on the specific machine learning algorithm that you are using. It is important to carefully consider the requirements of the algorithm and to pre-processes the data in a way that is compatible with the algorithm.

- **Validate the pre-processed data:**

It is important to validate the pre-processed data to ensure that it is in a format that can be used by the machine learning algorithm and that it is of high quality. This can be done by inspecting the data visually or by using statistical methods.

1.Loading the dataset:

- Loading the dataset using machine learning is the process of bringing the data into the machine learning environment so that it can be used to train and evaluate a model.
- The specific steps involved in loading the dataset will vary depending on the machine learning library or framework that is being used. However, there are some general steps that are common to most machine learning frameworks:

a. Identify the dataset:

The first step is to identify the dataset that you want to load. This dataset may be stored in a local file, in a database, or in a cloud storage service.

b. Load the dataset:

Once you have identified the dataset, you need to load it into the machine learning environment. This may involve using a built-in function in the machine learning library, or it may involve writing your own code.

c. Pre-process the dataset:

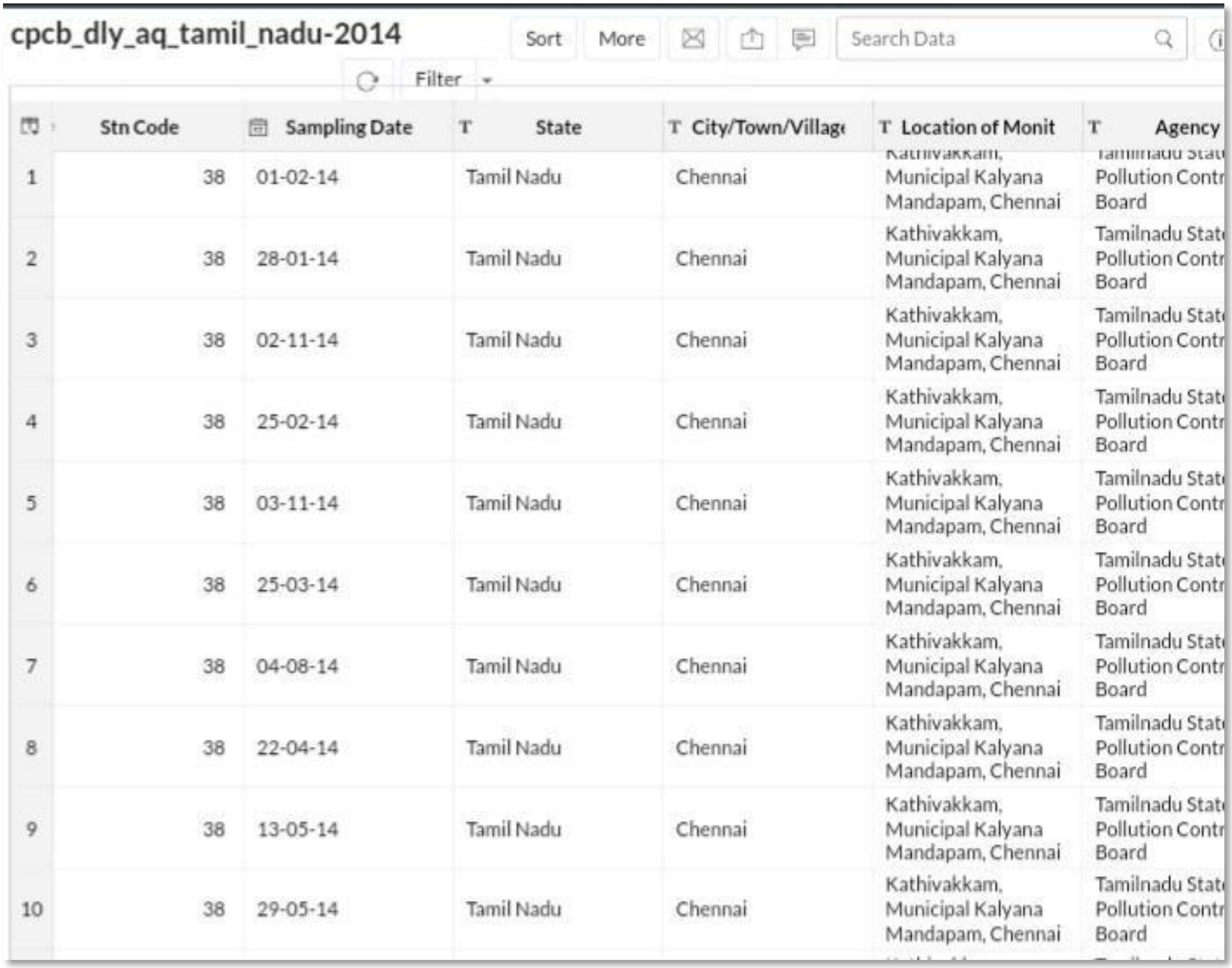
Once the dataset is loaded into the machine learning environment ,you may need to pre process it before you can start training and evaluating your model. This may involve cleaning the data, transforming the data into a suitable format, and splitting the data into training and test sets.

Program:

```
import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import os

for dirname, _, filenames in os.walk(‘/input/cpcb_dly_aq_tamil_nadu_2014.csc’):

   for filename in filenames:

print(os.path.join(dirname, filename))

df=pd.read_csv(‘../input/input/cpcb_dly_aq_tamil_nadu_2014.csc’,encoding=’cp1252’)
```

```
df.head()
```

Out:



cpcb_dly_aq_tamil_nadu-2014

	Stn Code	Sampling Date	State	City/Town/Villag	Location of Monit	Agency
1	38	01-02-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu Stat Pollution Contr Board
2	38	28-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu Stat Pollution Contr Board
3	38	02-11-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu Stat Pollution Contr Board
4	38	25-02-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu Stat Pollution Contr Board
5	38	03-11-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu Stat Pollution Contr Board
6	38	25-03-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu Stat Pollution Contr Board
7	38	04-08-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu Stat Pollution Contr Board
8	38	22-04-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu Stat Pollution Contr Board
9	38	13-05-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu Stat Pollution Contr Board
10	38	29-05-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu Stat Pollution Contr Board

```
df.shape
```

Out:

```
(2879,  11)
```

```
df[df[‘state’]==’Tamil Nadu’][‘location’].unique()
```

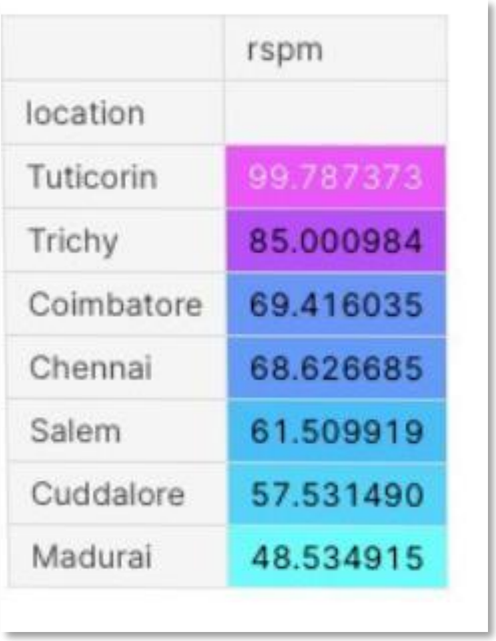
Out:

```
array([‘Madras’, ‘Turicorin’, ‘Tuticorin’, ‘Coimbatore’, ‘Madurai’,
       ‘Salem’, ‘Chennai’, ‘Thoothukudi’, ‘Trichy’, ‘Mettur’, ‘Cuddalore’],
      Dtype=object)
```

```
df[‘location’]=df[‘location’].replace((‘Madras’, ‘Turicorin’, ‘Thoothukudi’,
‘Mettur’),(‘Chennai’, ‘Tuticorin’, ‘Tuticorin’, ‘Salem’))
```

```
df[df[‘state’]==’Tamil Nadu’][[‘location’,
‘rspm’]].groupby([‘location’]).agg(‘mean’).sort_values(‘rspm’,

ascending=False).style.background_gradient(cmap=’cool’
```

Out:

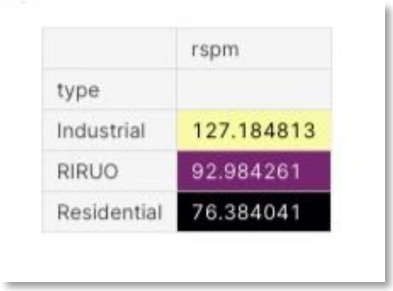


	rspm
location	
Tuticorin	99.787373
Trichy	85.000984
Coimbatore	69.416035
Chennai	68.626685
Salem	61.509919
Cuddalore	57.531490
Madurai	48.534915

```
#Let’s check which type of area causing more pollution in tuticorin
```

```
df[df[‘location’]==’Tuticorin’][[‘type’,
‘rspm’]].groupby([‘type’]).agg(‘mean’).sort_values(‘rspm’,
ascending=False).style.background_gradient(cmap=’inferno ’)
```

Out:

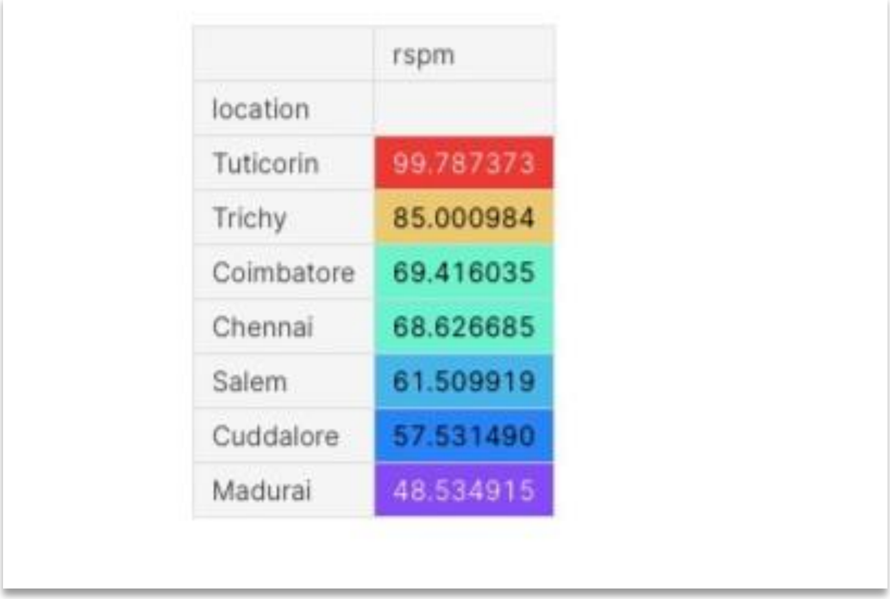


	rspm
type	
Industrial	127.184813
RIRUO	92.984261
Residential	76.384041

```
#As we know Industrial areas causing more pollution let’s check which industrial
areas causing more pollution in Tamil Nadu
```

```
df[(df[‘type’]==’industrial’)|(df[‘state’]==’Tamil Nadu’)][[‘location’,

‘rspm’]].groupby([‘location’]).agg(‘mean’).sort_values(‘rspm’,

ascending=False).style.background_gradient(cmap=’rainbow’)
```

Out:



	rspm
location	
Tuticorin	99.787373
Trichy	85.000984
Coimbatore	69.416035
Chennai	68.626685
Salem	61.509919
Cuddalore	57.531490
Madurai	48.534915

Data visualization :

Data visualization is a crucial aspect of data analysis and communication. It involves creating graphical representations of data to help people understand patterns, trends, and insights in the data.

Types of Visualizations:

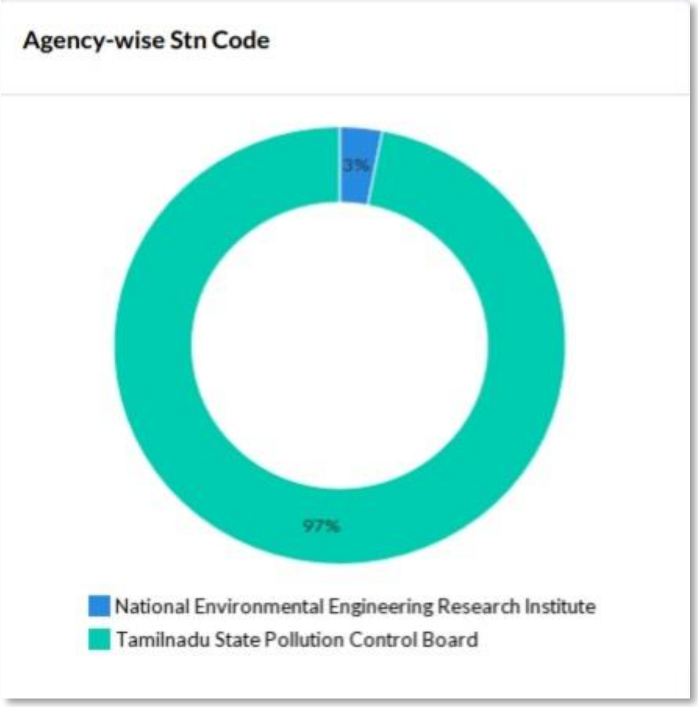
There are various types of data visualizations, including bar charts, line graphs, scatter plots, heatmaps, pie charts, histograms, and more. The choice of visualization depends on the data and the insights you want to convey.

Air Q Assessment TN- Data Visualization:

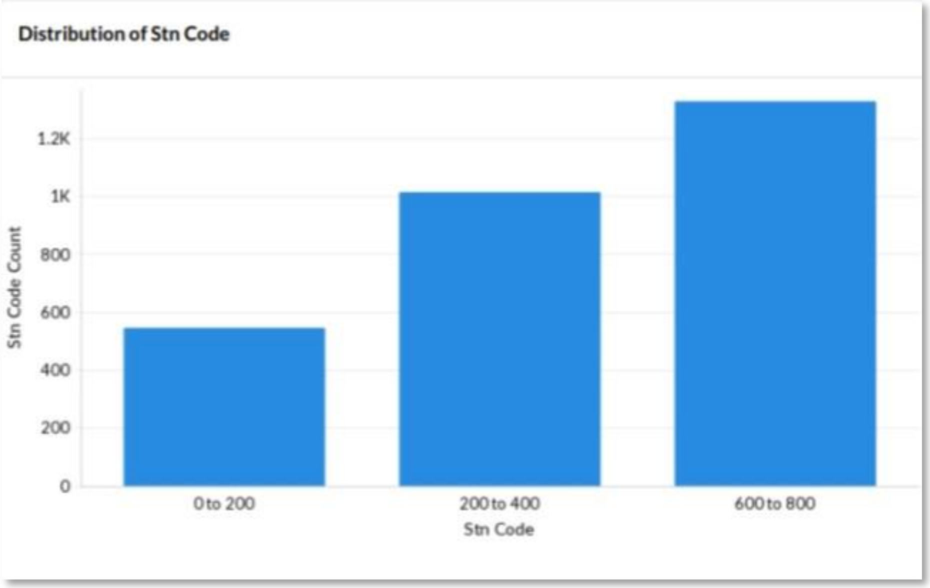
Air quality data can be visualized by combining real-time monitoring data with Python programming. Interactive graphs can be created.

Air quality monitors are equipped with sensors that detect specific pollutants. Some monitors use lasers to scan particulate matter density in a cubic meter of air. Others use satellite imaging to measure energy reflected or emitted by the Earth.

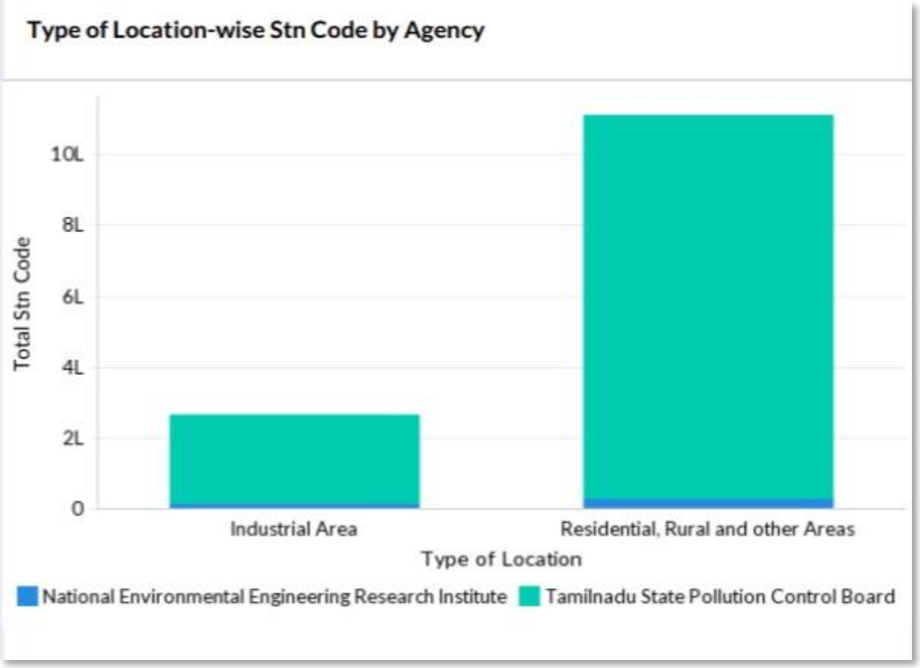
Agency wise Stn-Code(Station-Code):



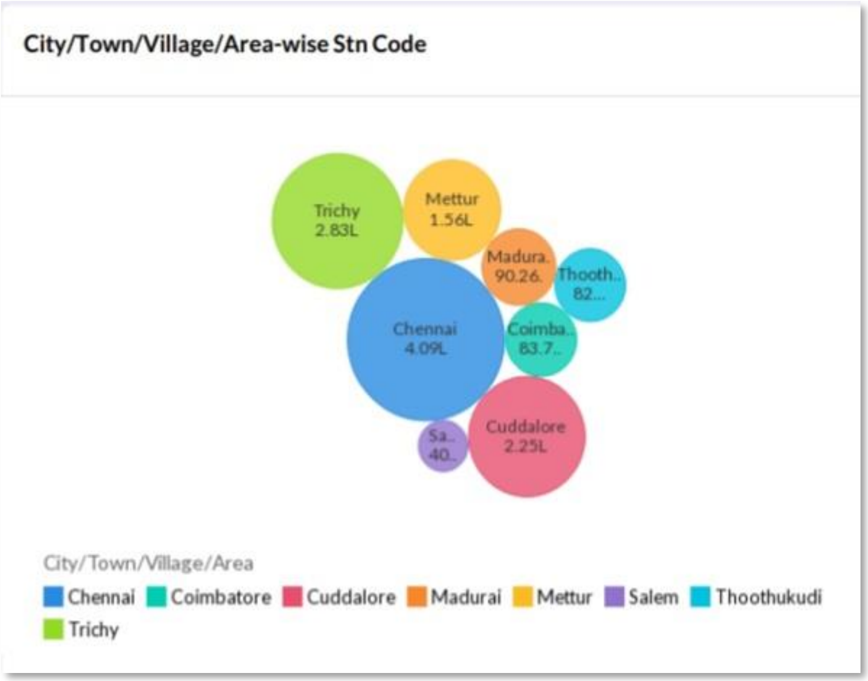
Distribution of Stn-Code:



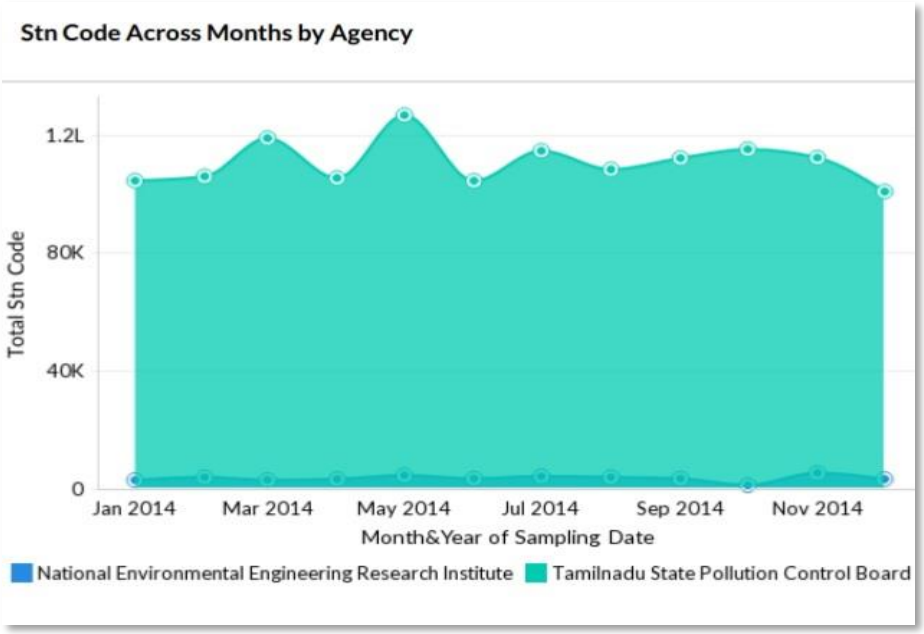
Type of location wise Stn-Code by Agency:



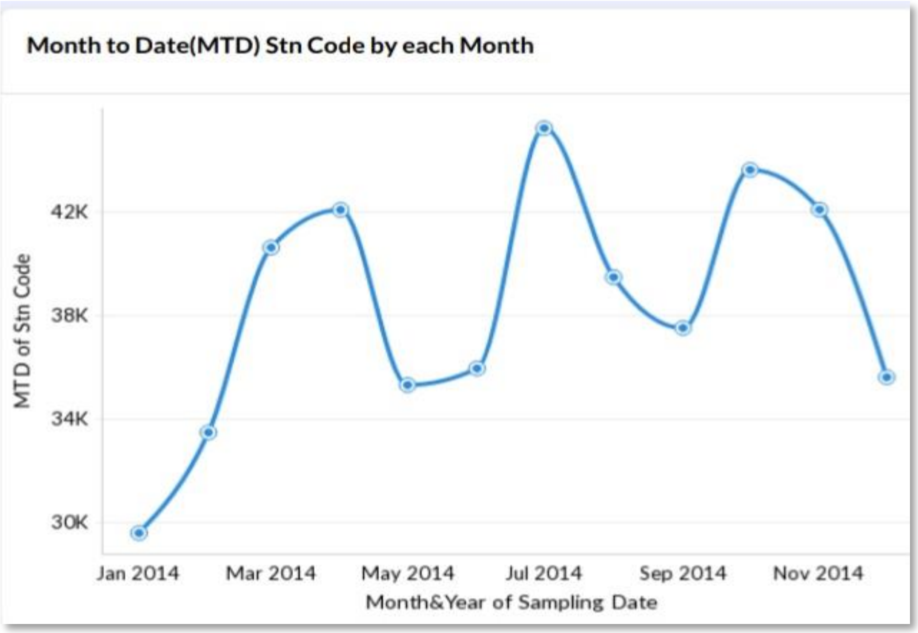
City/Town/Village/Area-wise Stn-Code:



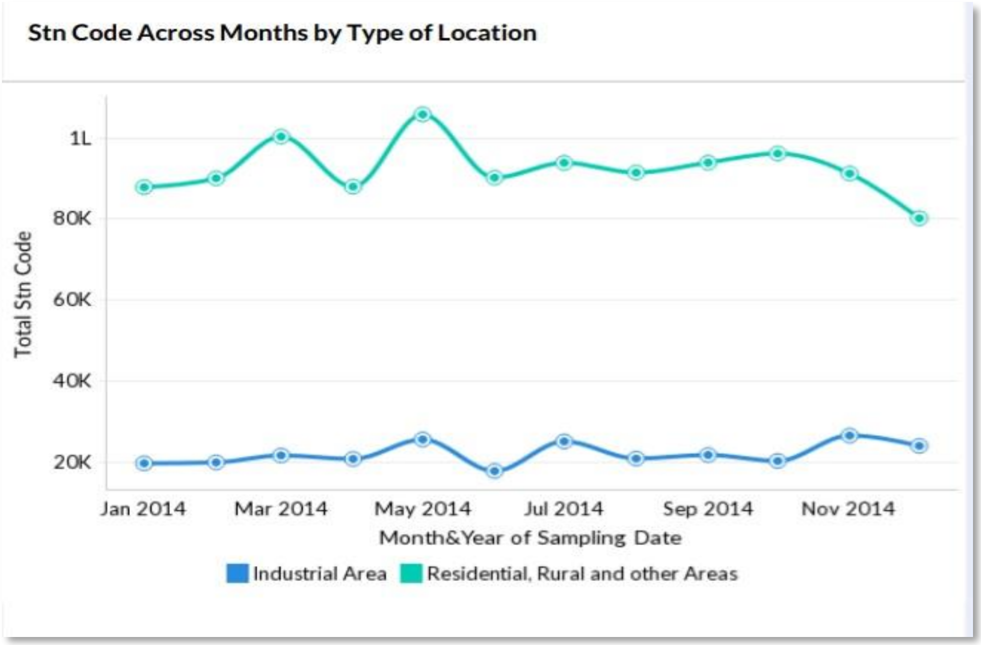
Stn-Code Across Month by Agency:



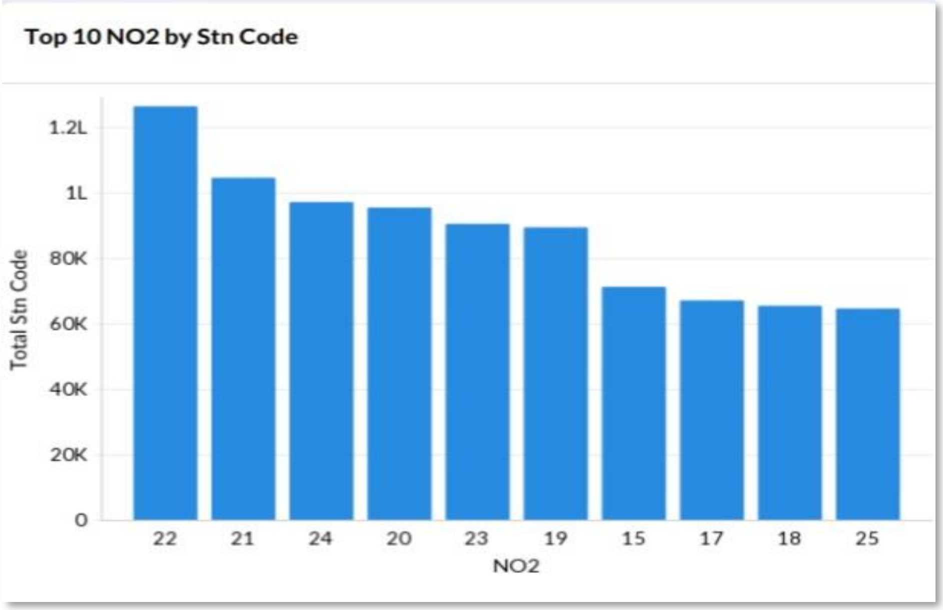
Month to Date(MTD)Stn-Code by Each Month:



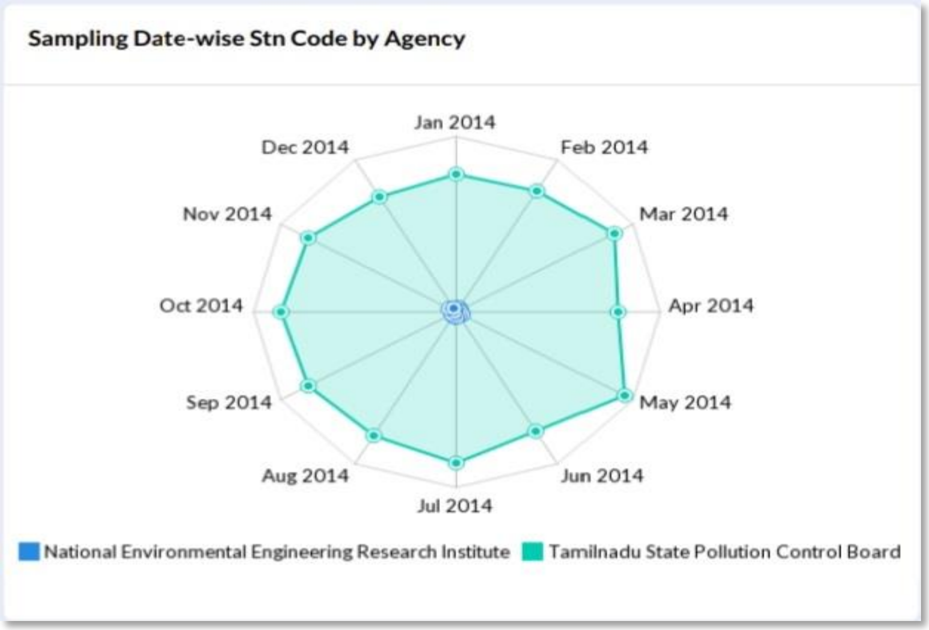
Stn-Code Across Months by Type of Location:



Top 10 NO2 by Stn-Code:



Sampling Data wise Stn-Code by Agency:

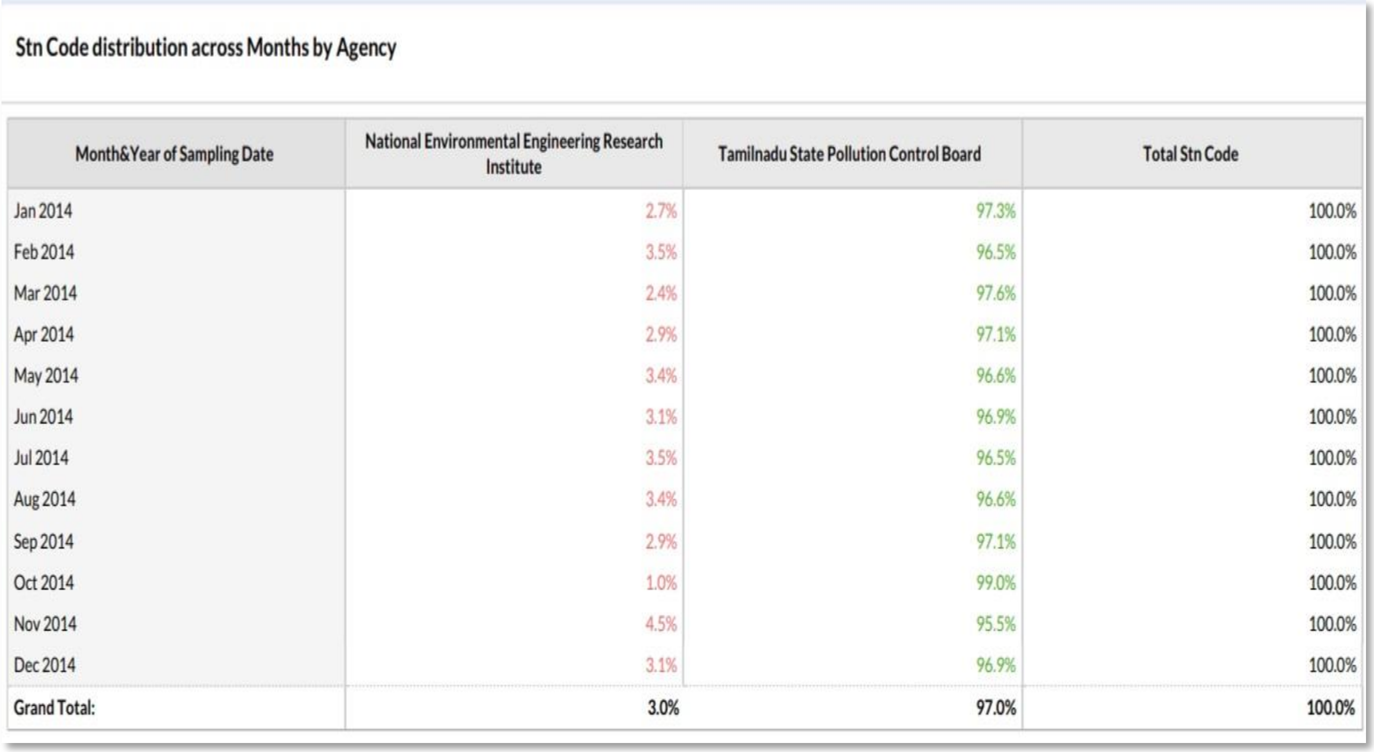


Stn-Code Distribution Across Months by Agency:

The Tamil Nadu Pollution Control Board (TNPCB) operates air quality monitoring stations in 20 districts across the state. The TNPCB has eight stations in Chennai, three in Thoothukudi, three in Coimbatore, and one in Salem.

The following agencies monitor air quality twice a week:

- Pollution Control Committees
- Central Pollution Control Board
- National Environmental Engineering Research Institute
- State Pollution Control Boards

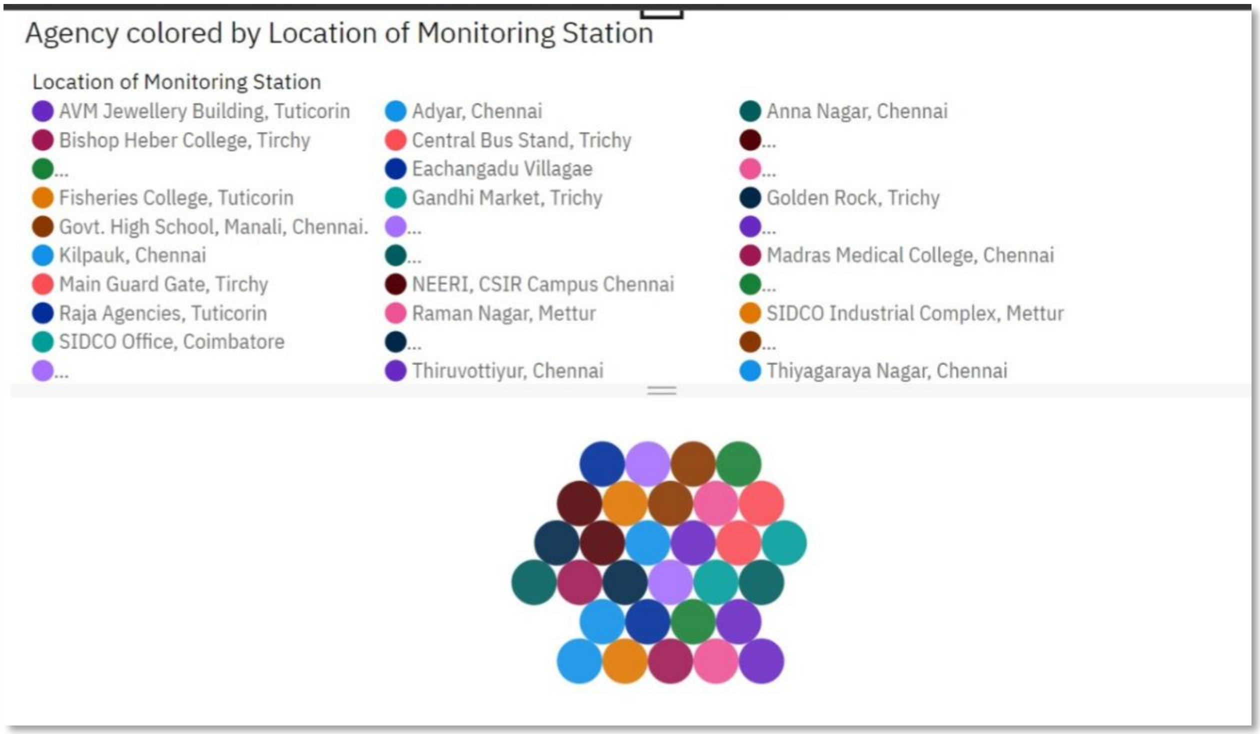


Stn Code distribution across Months by Agency

Month&Year of Sampling Date	National Environmental Engineering Research Institute	Tamilnadu State Pollution Control Board	Total Stn Code
Jan 2014	2.7%	97.3%	100.0%
Feb 2014	3.5%	96.5%	100.0%
Mar 2014	2.4%	97.6%	100.0%
Apr 2014	2.9%	97.1%	100.0%
May 2014	3.4%	96.6%	100.0%
Jun 2014	3.1%	96.9%	100.0%
Jul 2014	3.5%	96.5%	100.0%
Aug 2014	3.4%	96.6%	100.0%
Sep 2014	2.9%	97.1%	100.0%
Oct 2014	1.0%	99.0%	100.0%
Nov 2014	4.5%	95.5%	100.0%
Dec 2014	3.1%	96.9%	100.0%
Grand Total:	3.0%	97.0%	100.0%

<u>Agency coloured by location of monitoring station:</u>

Tamil Nadu Pollution Control Board is operating eight ambient air quality monitoring stations in Chennai, Three ambient air quality monitoring stations in Thoothukudi, Three ambient air quality monitoring stations in Coimbatore ,One ambient air quality monitoring stations in Salem, Three ambient air quality monitoring stations in Madurai, Five ambient air quality monitoring stations in Trichy, Three ambient air quality monitoring stations in Cuddalore and Two ambient air quality monitoring stations in Mettur under National Air Quality Monitoring Programme (NAMP) funded by Central Pollution Control Board.



Conclusion of Phase-3 Project:

In the Phase 3 conclusion, we will summarize the data analysis and data visualization of **Air Q Assessment of Tamil Nadu** by using the given dataset.

From this development part 1,we built the project by loading and pre-processing the dataset. Load the dataset using Python and data manipulation libraries (e.g., pandas). In future we will develop part 2 by building the project by performing different activities like feature engineering ,model training, evaluation.