# Air Quality Analysis In TamilNadu

In this part to begin building our project by loading and preprocessing the dataset. Begin the analysis by loading and preprocessing the air quality dataset.

To begin our analysis and development project by loading and preprocessing an air quality dataset, I can follow these steps using Python and the Pandas library. In this example, I'll assume the given CSV file containing the air quality data. I can adapt the code to your specific dataset format.

#### LOADING AND PREPROCESSING METHODS:

- IMPORT LIBRARIES
- LOAD THE DATASET
- EXPLORE THE DATASET
- HANDLING MISSING DATA
- DATA CLEANING
- DATA TRANSFORMATION
- FEATURE ENGINEERING
- EXPLORATORY DATA ANALYSIS (EDA)
- SAVE PREPROCESSED DATASET

#### FROM THE GIVEN DATASET:

In this section I Use The Csv File Air Quality Analysis In Tamil Nadu Dataset From IBM Naan Mudhalvan

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

#### **IMPORT LIBRARIES:**

Import the necessary libraries, primarily Pandas for data manipulation and NumPy for numerical operations. You may also need other libraries based on the specifics of your dataset.

- IMPORT PANDAS AS PD
- O IMPORT NUMPY AS NP

#### LOAD THE DATASET:

Load your dataset into a Pandas DataFrame. Replace 'your\_dataset.csv' with the actual file path or URL of your dataset.

data = pd.read\_csv('Example.csv')
If you have a different format (e.g., Excel, JSON), you can use appropriate
Pandas functions like pd.read\_excel() or pd.read\_json().

```
In [2]: import pandas as pd
          import plotly.express as px
          import plotly.io as pio
          import plotly.graph_objects as go
pio.templates.default = "plotly_white"
          data = pd.read_csv("D:\cpcb_dly_aq_tamil_nadu-2014.csv")
          print(data.head())
             Stn Code Sampling Date
                                                  State City/Town/Village/Area \
                 rn Code Sampling Date State City/Town/Vil

38 01-02-14 Tamil Nadu

38 01-07-14 Tamil Nadu

38 21-01-14 Tamil Nadu

38 23-01-14 Tamil Nadu

38 28-01-14 Tamil Nadu
                                     Location of Monitoring Station \
          0 Kathivakkam, Municipal Kalyana Mandapam, Chennai
          1 Kathivakkam, Municipal Kalyana Mandapam, Chennai
          2 Kathivakkam, Municipal Kalyana Mandapam, Chennai
3 Kathivakkam, Municipal Kalyana Mandapam, Chennai
           4 Kathivakkam, Municipal Kalyana Mandapam, Chennai
                                                        Agency Type of Location SO2 NO2 \
          0 Tamilnadu State Pollution Control Board Industrial Area 11.0 17.0
1 Tamilnadu State Pollution Control Board Industrial Area 13.0 17.0
             Tamilnadu State Pollution Control Board Industrial Area 12.0 18.0
             Tamilnadu State Pollution Control Board Industrial Area 15.0 16.0
             Tamilnadu State Pollution Control Board Industrial Area 13.0 14.0
              RSPM/PM10 PM 2.5
                   55.0
                    46.0
                               NaN
                    42.0
```

## **Explore the Dataset:**

Begin by getting an overview of your dataset. Check the first few rows, column names, and data types.

#### Ex:

print(df.head()) # Display the first few rows

```
In [3]: print(data.describe())
               Stn Code
                              502
                                               RSPM/PM10 PM 2.5
       count 2879.000000 2879.000000 2879.000000 2879.000000
       mean 475.750261 11.515109
                                    22.136158 62.511289
                                                           NaN
       std 277.675577 5.071178 7.123029 31.393031
                                                           NaN
       min
             38.000000 2.000000
                                    5.000000 12.000000
                                                          NaN
       25% 238.000000 8.000000 17.000000 41.000000
                                                           NaN
       50%
            366.000000 12.000000 22.000000 55.000000
                                                           NaN
       75%
            764.000000 15.000000
                                    25.000000 78.000000
                                                           NaN
             773.000000
                        49.000000
       max
                                    71.000000 269.000000
                                                           NaN
```

print(df.columns) # List of column names

print(df.dtypes) # Data types of each column

```
print(data.dtypes)
Stn Code
                                int64
Sampling Date
                                object
City/Town/Village/Area
                             object
Location of Monitoring Station object
                               object
Type of Location
                               object
S02
                              float64
NO2
                              float64
RSPM/PM10
                              float64
PM 2.5
                               float64
dtype: object
```

## HANDLING MISSING DATA:

Identify and handle missing data, which could involve removing rows with missing values or imputing missing values.

```
# Check for missing values
print(df.isnull().sum())
# Handle missing values (example: impute with mean)
df['column_name'].fillna(df['column_name'].mean(), inplace=True)
```

```
# Check for missing values
print(data.isnull().sum())
# Handle missing values (example: impute with mean)
data['PM 2.5'].fillna(data['PM 2.5'].mean(), inplace=True)
Stn Code
                                      Θ
Sampling Date
                                      0
City/Town/Village/Area
Location of Monitoring Station
Type of Location
                                      0
                                      0
S02
NO<sub>2</sub>
                                      0
RSPM/PM10
                                      0
PM 2.5
                                   2879
dtype: int64
```

## Data Cleaning:

Clean the data by addressing any data anomalies, inconsistencies, or outliers.

## **Data Transformation:**

Depending on your project's requirements, you may need to transform the data. This could include converting date columns to datetime objects, encoding categorical variables, or scaling numerical features.

import matplotlib.pyplot as plt
from pandas.api.types import is\_string\_dtype, is\_numeric\_dtype

df = pd.read\_csv("../input/marketing-data/marketing\_data.csv")
df.head()

<pre>import matplotlib.pyplot as plt from pandas.api.types import is_string_dtype, is_numeric_dtype  df = pd.read_csv("D:\cpcb_dly_aq_tamil_nadu-2014.csv")  df.head()</pre>											
	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10	PM 2.5
0	38	01-02-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.0	17.0	55.0	NaN
1	38	01-07-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	17.0	45.0	NaN
2	38	21-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	12.0	18.0	50.0	NaN
3	38	23-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	15.0	16.0	46.0	NaN
4	38	28-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	14.0	42.0	NaN

## Feature Engineering:

Create new features or modify existing ones to improve your dataset's quality.

## Exploratory Data Analysis (EDA):

Perform exploratory data analysis using visualizations (e.g., Matplotlib or Seaborn) to gain insights into your data.

# Save Preprocessed Dataset:

Once you've completed preprocessing, save the cleaned and transformed dataset to a new file for future use.

df.to\_csv('preprocessed\_dataset.csv', index=False)

These steps provide a general guideline for loading and preprocessing a dataset. The specifics may vary depending on your dataset, project goals, and data quality