Phase 3: Development Part 1

In this part you will begin building your project by loading and preprocessing the dataset.

Begin the analysis by loading and preprocessing the air quality dataset

Load the dataset using Python and data manipulation libraries (e.g., pandas).

Air Quality Analysis and Prediction in Tamil Nadu

1. Import the necessary libraries:

First, import the libraries you'll need for data manipulation and analysis. You'll primarily use `pandas` for data handling.

```
import pandas as pd
import matplotlib.pyplot as plt
```

2. Load the dataset:

You'll need to load the air quality dataset into a pandas DataFrame. You can use various methods to load data depending on the file format. For example, if you have a CSV file, you can use `pd.read_csv()`.

```
# Assuming your data is in a CSV file
data = pd.read_csv('/content/cpcb_dly_aq_tamil_nadu-2014 (1).csv')
```

3. Data Exploration:

Once the data is loaded, you can start exploring it. Here are some common operations to get an initial understanding of the dataset:

```
# Display basic statistics
print(data.describe())

# Check for missing values
print(data.isnull().sum())

# Check unique values in categorical columns
print(data['State'].unique())
print(data['City/Town/Village/Area'].unique())
# ... Repeat for other categorical columns
```

```
          Stn Code           SO2           NO2    RSPM/PM10  PM 2.5
count  2879.000000  2868.000000  2866.000000  2875.000000     0.0
mean    475.750261    11.503138    22.136776    62.494261     NaN
std     277.675577     5.051702     7.128694    31.368745     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
max     773.000000    49.000000    71.000000   269.000000     NaN
Stn Code                            0
Sampling Date                       0
State                               0
City/Town/Village/Area              0
Location of Monitoring Station      0
Agency                              0
Type of Location                    0
SO2                                11
NO2                                13
RSPM/PM10                           4
PM 2.5                           2879
dtype: int64
['Tamil Nadu']
['Chennai' 'Coimbatore' 'Cuddalore' 'Madurai' 'Mettur' 'Salem'
 'Thoothukudi' 'Trichy']
```

4. Data Preprocessing:

Based on the initial exploration, you might need to perform data preprocessing. This can include handling missing values, renaming columns, converting data types, and more.

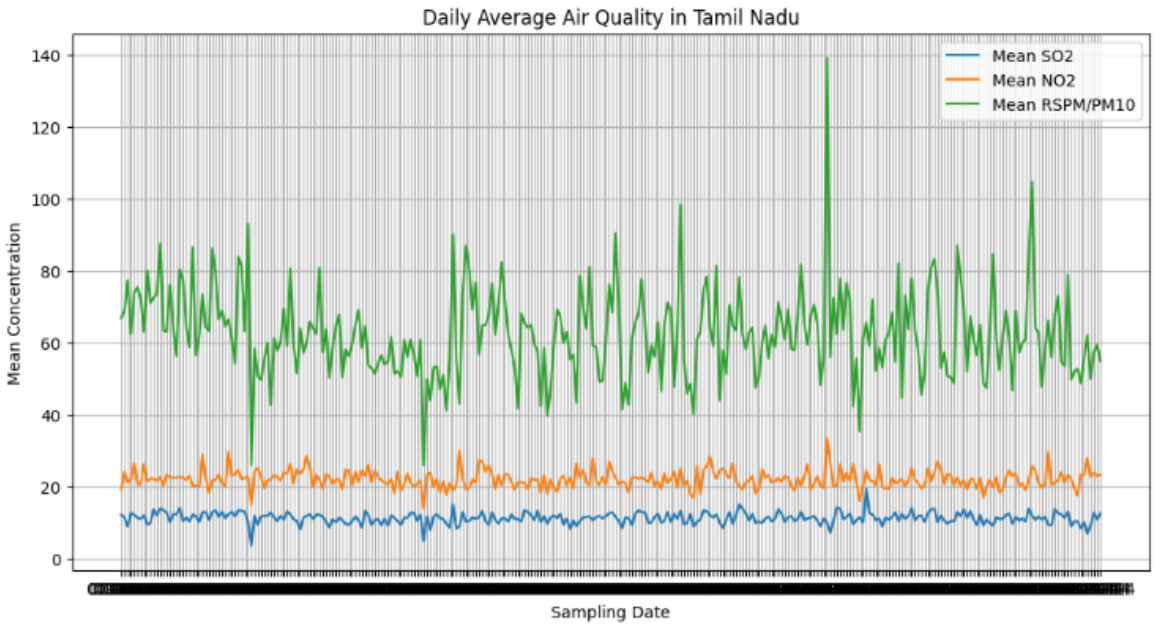
5. Data Visualization:

You can create visualizations to better understand the data. Matplotlib or Seaborn can be used for this purpose. you may need to perform various preprocessing tasks like data cleaning, data transformation, feature engineering, and data normalization.

Line Chart

```
# Group data by date and calculate mean values
daily_mean = data.groupby('Sampling Date').mean()

# Plot daily average air quality
plt.figure(figsize=(12, 6))
plt.plot(daily_mean.index, daily_mean['SO2'], label='Mean SO2')
plt.plot(daily_mean.index, daily_mean['NO2'], label='Mean NO2')
plt.plot(daily_mean.index, daily_mean['RSPM/PM10'], label='Mean RSPM/PM10')
plt.xlabel('Sampling Date')
plt.ylabel('Mean Concentration')
plt.title('Daily Average Air Quality in Tamil Nadu')
plt.legend()
plt.grid(True)
plt.show()
```



In the data visualization step, a heatmap was created using Seaborn and Matplotlib to provide a visual representation of air quality levels in different monitoring locations over time. This heatmap helps to reveal patterns and variations in RSPM/PM10 pollutant levels across different locations, enhancing our understanding of air quality trends.

Heatmap

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

# Load your air quality dataset
# Replace 'your_dataset.csv' with the actual file path
df = pd.read_csv('/content/cpcb_dly_aq_tamil_nadu-2014 (1).csv')

# Select the relevant columns for the heatmap (e.g., pollutant levels by location and time)
# Replace 'Pollutant', 'Location', and 'Time' with your column names
data = df.pivot_table(index='Location of Monitoring Station', values='RSPM/PM10')

# Create a heatmap
plt.figure(figsize=(12, 8))  # Adjust the figure size as needed
sns.heatmap(data, cmap='YlGnBu', annot=True, fmt=".1f")

# Customize the heatmap labels and title
plt.xlabel('Time')
plt.ylabel('Location')
plt.title('Air Quality Heatmap')

# Display the heatmap
plt.show()
```

7. Feature selection:

Feature selection is important to choose the most relevant variables for your analysis. You can use techniques like feature importance scores, correlation analysis, or domain knowledge to select features.

Popular feature selection methods include Recursive Feature Elimination (RFE), SelectKBest, or using machine learning models that provide feature importance scores.

Be sure to document the features you select and the rationale behind the selection process.

8. Save the preprocessed dataset:

To save your preprocessed dataset, you can use pandas to save it as a CSV, Excel, or any other format that suits your needs. For instance, you can use `df.to_csv()` to save it as a CSV file.

It's a good practice to save the preprocessed dataset to a new file or object to ensure that you can work with a clean and consistent dataset in subsequent steps of your analysis.

Consider using meaningful names for the saved file to distinguish it from the original dataset.