Phase 3: Development Part 1

In this part, we will begin building our project by loading and preprocessing the dataset.

We will load the dataset using Python and data manipulation libraries (e.g., pandas).

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

To load and preprocess the air quality dataset from the provided link, we can use the `pandas` library in Python. Here's a step-by-step guide to help us get started:

1. Install and Import Libraries:
   First, we need to make sure we have the `pandas` library installed. If we haven't installed it, we can do so using pip:

   ```
   pip install pandas
   ```

   Then, we import the necessary libraries:

   ```
   import pandas as pd
   ```

2. Download the Dataset:
   We can download the dataset from the provided link manually and save it to our local directory. Alternatively, we can automate this process using Python libraries like `requests`.

   - Visit the provided link: https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil-nadu-year-2014
   - Look for a download link on the webpage and download the dataset in a format like CSV.

   If we want to automate the download process, we can use Python's `requests` library to fetch the dataset from the URL and save it to our local directory.

   ```
   import requests
   ```

```
url = "<https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil
-nadu-year-2014>"
response = requests.get(url)

with open("air_quality_data.csv", "wb") as file:
    file.write(response.content)
```

This code will download the dataset as a CSV file named "air_quality_data.csv."

3. Load the Dataset:
   Now that we have the dataset locally, we can load it into a pandas DataFrame:

```
# Load the dataset into a DataFrame
df = pd.read_csv("air_quality_data.csv")

# Display the first few rows of the DataFrame to inspect the data
print(df.head())
```

   This code loads the dataset into a pandas DataFrame named `df`. We can use the `head()` method to display the first few rows of the dataset to get an overview of the data.

4. Preprocess the Data:
   Depending on the structure and quality of the dataset, we may need to perform various preprocessing steps, such as handling missing values, renaming columns, and converting data types. We can use pandas functions for these tasks.

   For example, if we have missing values in the dataset, we can use the `fillna()` method to fill them with appropriate values, or we can drop rows with missing data using the `dropna()` method.

   Additionally, we can rename columns using the `rename()` method and perform other data cleaning operations as needed.

With these steps, we can load and preprocess the air quality dataset for our analysis. We should make sure to inspect the data and perform any necessary data cleaning and preprocessing steps to prepare it for our analysis."

To load and preprocess the air quality dataset from the provided link, we will be using the `pandas` library in Python.

1. Install and Import Libraries:
   First, make sure you have the `pandas` library installed. If you haven't installed it, you can do so using pip:

   ```
   pip install pandas
   ```

   Then, import the necessary libraries:

   ```
   import pandas as pd
   ```

2. Download the Dataset:
   You can download the dataset from the provided link manually and save it to your local directory. Alternatively, you can automate this process using Python libraries like `requests`. Here's how you can do it manually:

   - Visit the provided link: https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil-nadu-year-2014
   - Look for a download link on the webpage and download the dataset in a format like CSV.

   If you want to automate the download process, you can use Python's `requests` library to fetch the dataset from the URL and save it to your local directory.

   ```
   import requests

   url = "<https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil
   -nadu-year-2014>"
   response = requests.get(url)

   with open("air_quality_data.csv", "wb") as file:
       file.write(response.content)
   ```

   This code will download the dataset as a CSV file named "air_quality_data.csv."

3. Load the Dataset:
   Now that you have the dataset locally, you can load it into a pandas DataFrame:

```
# Load the dataset into a DataFrame
df = pd.read_csv("air_quality_data.csv")

# Display the first few rows of the DataFrame to inspect the data
print(df.head())
```

This code loads the dataset into a pandas DataFrame named `df`. You can use the `head()` method to display the first few rows of the dataset to get an overview of the data.

4. Preprocess the Data:
   Depending on the structure and quality of the dataset, you may need to perform various preprocessing steps, such as handling missing values, renaming columns, and converting data types. You can use pandas functions for these tasks.

   For example, if you have missing values in the dataset, you can use the `fillna()` method to fill them with appropriate values, or you can drop rows with missing data using the `dropna()` method.

   Additionally, you can rename columns using the `rename()` method and perform other data cleaning operations as needed.

With these steps, you can load and preprocess the air quality dataset for your analysis. Make sure to inspect the data and perform any necessary data cleaning and preprocessing steps to prepare it for your analysis.