Phase 3:Development part 1

Title:Being the analysis by loading and preprocessing the air quality dataset

Introduction:

To analyze an air quality dataset in Python, you'll typically need to load the for data manipulation and Matplotlib for data visualization. You may need to install these libraries if you haven't already: dataset, preprocess it, and then perform various analyses based on your specific goals. Here's a general outline of how you can do this using the popular Pandas

Make sure to replace `'your_dataset.csv'` with the actual file path to your air quality dataset. You should customize the analysis and preprocessing steps according to the characteristics of your dataset and the specific questions you want to answer. Common preprocessing steps include handling missing data, converting data types, and filtering out irrelevant columns.

For more advanced analysis, you might want to use machine learning or time series analysis techniques, depending on the nature of your dataset and your objectives

Dataset:

Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station
38	1/2/2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chenn
38	1/7/2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chenn
71	3/3/2014	Tamil Nadu	Chennai	Govt. High School, Manali, Chennai.
71	3/6/2014	Tamil Nadu	Chennai	Govt. High School, Manali, Chennai.
72	3/5/2014	Tamil Nadu	Chennai	Thiruvottiyur, Chennai
72	3/7/2014	Tamil Nadu	Chennai	Thiruvottiyur, Chennai

766	5/7/2014	Tamil Nadu	Chennai	Thiyagaraya Nagar, Chennai
766	5/9/2014	Tamil Nadu	Chennai	Thiyagaraya Nagar, Chennai
765	3/3/2014	Tamil Nadu	Chennai	Anna Nagar, Chennai
765	3/5/2014	Tamil Nadu	Chennai	Anna Nagar, Chennai
765	3/7/2014	Tamil Nadu	Chennai	Anna Nagar, Chennai

Program:

Import Necessary Libraries:

```
import pandas as pd

import numpy as np
```

Load the Dataset:

Load your air quality dataset using Pandas. You can typically load data from various file formats like CSV, Excel, or databases. Replace **'your_dataset.csv'** with the actual file path or URL of your dataset.

```
data = pd.read_csv('your_dataset.csv')
```

Explore the Data:

It's a good practice to get a sense of your dataset by checking the first few rows and basic statistics**.**

```
print(data.head())

print(data.describe())
```

Handle Missing Values:

Depending on the dataset, you may have missing values that need to be addressed. You can either remove rows or fill in missing data. Common methods include:

- Removing rows with missing values:

```
data.dropna(inplace=True)
```

- Filling missing values with a specific value (e.g., mean, median):

data.fillna(data.mean(), inplace=True)

Data Preprocessing:

Depending on the nature of your dataset, you may need to perform additional preprocessing steps like encoding categorical variables, scaling numerical features, and creating new features.

- Encoding categorical variables (if any):

data = pd.get_dummies(data, columns=['categorical_column'])

- Scaling numerical features (e.g., using Min-Max scaling or Standardization):

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

data[['numerical_feature1', 'numerical_feature2']]=scaler.fit_transform(data[['numerical_feature1', 'numerical_feature2']])

Split Data into Features and Target:

Identify your target variable (the variable you want to predict) and separate it from the features.

X = data.drop(columns=['target_variable']) y = data['target_variable']

Train-Test Split:

Split your data into training and testing sets to evaluate your machine learning model. This step is crucial for model validation.

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

Data is Ready for Modeling:

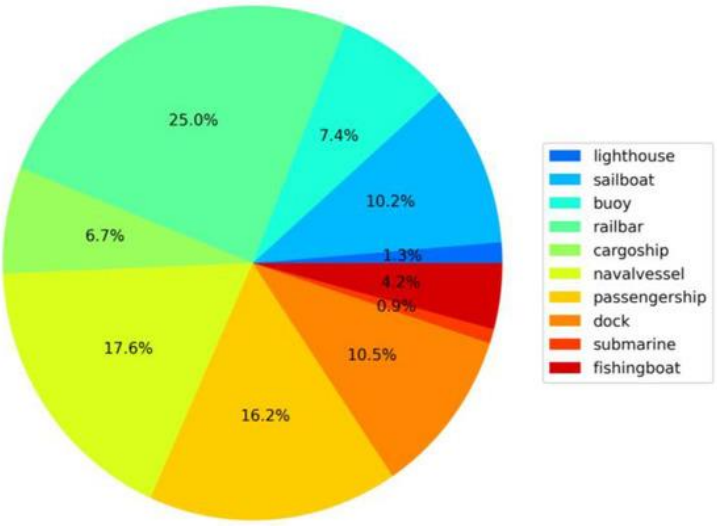
At this point, your data is preprocessed and ready for use in machine learning models. You can now proceed with model selection, training, and evaluation.

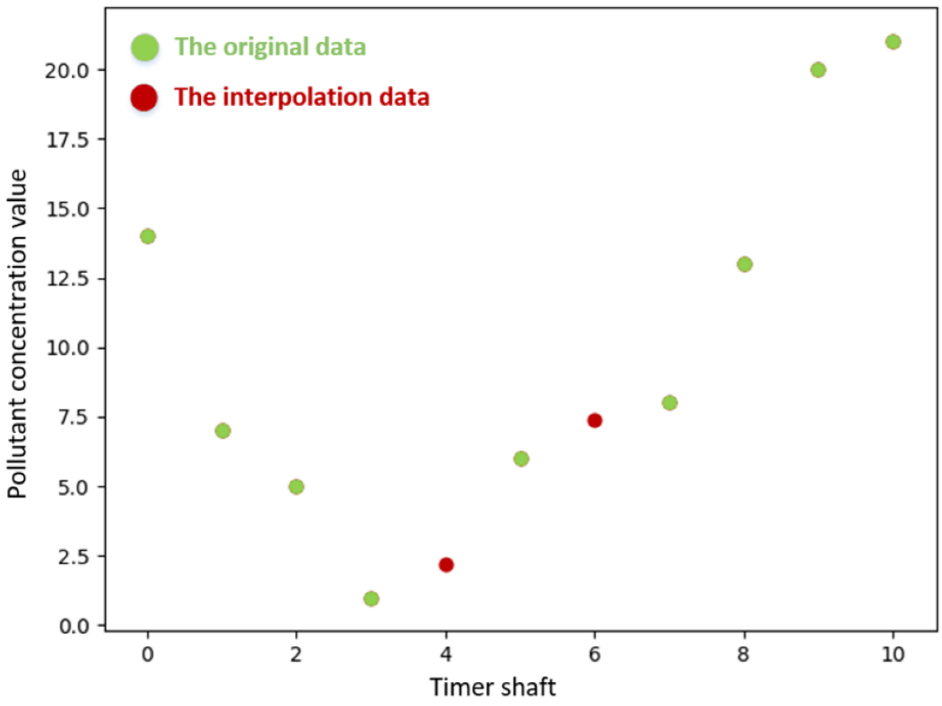
Remember that the specific steps may vary depending on your dataset and the problem you are trying to solve. Additionally, you may need to use different techniques or libraries for advanced data preprocessing or domain-specific requirements.

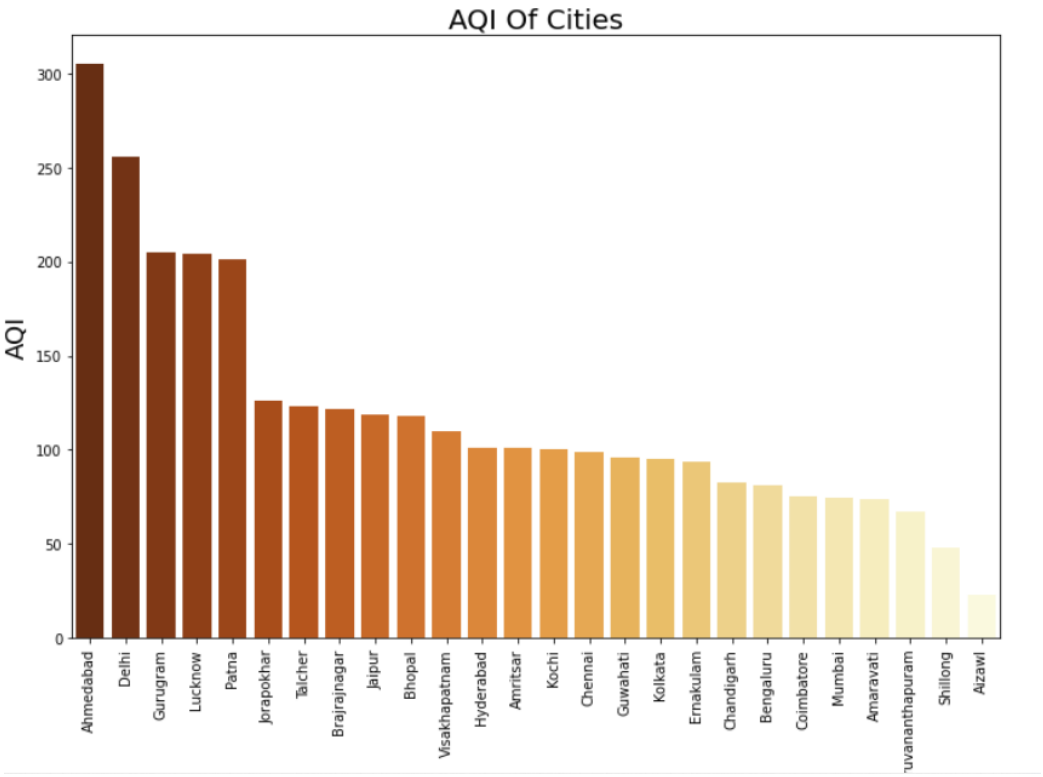
Program:

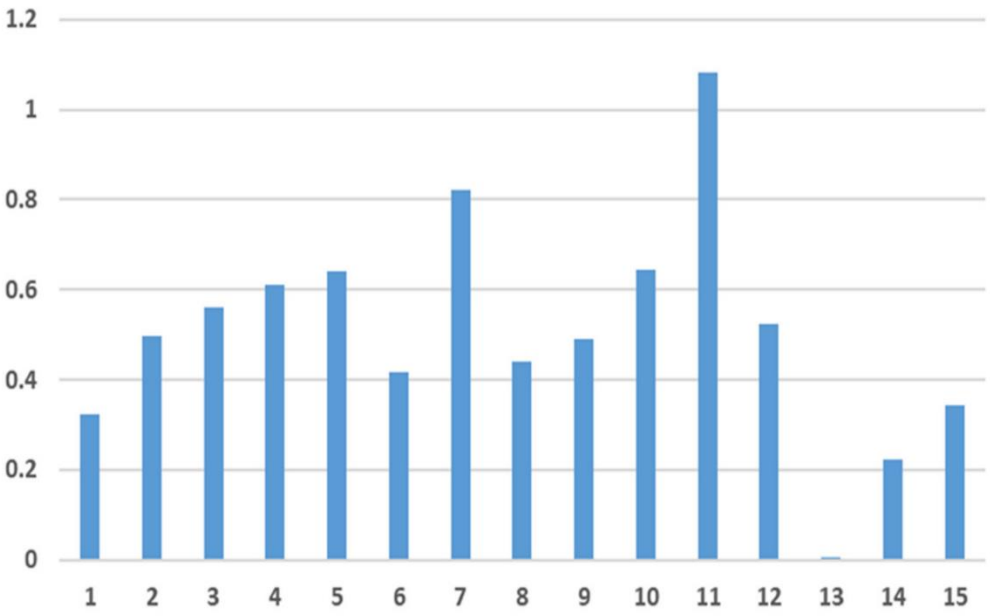
```
import pandas as pd

import numpy as np

data = pd.read_csv('your_air_quality_dataset.csv')

print(data.head())

print(data.describe())

data.fillna(data.mean(), inplace=True)

data = pd.get_dummies(data, columns=['categorical_column'])

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

data[['numerical_feature1', 'numerical_feature2']] =
scaler.fit_transform(data[['numerical_feature1', 'numerical_feature2']])

X = data.drop(columns=['target_variable'])

y = data['target_variable']

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Output:


25.0%
7.4%
10.2%
6.7%
1.3%
4.2%
0.9%
17.6%
10.5%
16.2%
lighthouse
sailboat
buoy
railbar
cargoship
navalvessel
passengership
dock
submarine
fishingboat


The original data
The interpolation data
Pollutant concentration value
Timer shaft
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
2
4
6
8
10





Conclusion:

1. **Loading the Dataset:** Use libraries like Pandas to load your air quality dataset from a file (e.g., CSV) or another data source.

2. **Exploratory Data Analysis (EDA)**: Conduct basic exploratory data analysis to understand the structure and characteristics of your data, including checking the first few rows and obtaining summary statistics.

3. **Handling Missing Values**: Identify and handle missing values in the dataset. You can choose to remove rows with missing values or impute missing values using appropriate strategies like mean, median, or custom methods.

4. **Data Preprocessing**: Depending on the nature of your data, perform preprocessing tasks such as encoding categorical variables, scaling numerical features, and creating new features. This step can be tailored to the specific requirements of your dataset and the machine learning model you intend to use.

5. **Splitting Data**: Split your data into features (X) and the target variable (y). This separation is essential for supervised machine learning tasks.

6. **Train-Test Split**: Further split your data into training and testing sets, allowing you to evaluate the performance of machine learning models accurately.

After completing these steps, your air quality dataset should be in a suitable format for use in machine learning models or for further data analysis. Remember that the specific preprocessing steps and techniques may vary depending on the dataset's characteristics and the goals of your analysis or modeling project.