AIR QUALITY ANALYSIS IN TAMILNADU

Phase 3: Development Part 1

TOPIC: **Begin the analysis by loading and preprocessing the air quality dataset.**

INTRODUCTION

- Clean air is the foremost requirement to sustain healthy lives of humankind and those of the supporting ecosystems which in return affect the human wellbeing. Due to the rapid growth of economy and fossil fuel consumption and lack of emission controls, we have experienced substantially elevated concentrations of air pollutants, which not only degrade regional air quality, but also exert significant impacts on public health and global climate.

- In an era where urbanization and industrialization are on the rise, monitoring and improving air quality has become a paramount concern for environmentalists, policymakers, and the general public. Poor air quality can have severe health, environmental, and economic consequences. To address this critical issue, we embark on an Air Quality Analysis Project utilizing the power of Python and various data manipulation libraries.

- This project aims to leverage the capabilities of Python, a versatile and widely-used programming language, along with popular libraries like NumPy, Pandas, Matplotlib, and Seaborn, to gather, analyze, and visualize air quality data. By harnessing the potential of these tools, we can better understand air quality patterns, identify sources of pollution, and contribute to data-driven decision-making for improving air quality standards.

Given Dataset:

STN CODE	SAMPLING DATE	STATE	CITY	MONITORING STATION	AGENCY	TYPE OF LOCATION	SO2	NO2	RSPM/PM10	PM2.5
38	02-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	11	17	55	NA
38	07-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	13	17	45	NA
38	21-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	12	18	50	NA
38	23-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	15	16	46	NA
38	28-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	13	14	42	NA
....	....	....	....	....	....	....	....	....	....	....
773	03-12-2014	TAMILNADU	TRICHY	CENTRAL BUS STAND	TAMILNADU STATE POLLUTION CONTROL BOARD	RESIDENTIAL ,RURAL AND OTHER AREAS	15	18	102	NA
773	10-12-2014	TAMILNADU	TRICHY	CENTRAL BUS STAND	TAMILNADU STATE POLLUTION CONTROL BOARD	RESIDENTIAL ,RURAL AND OTHER AREAS	12	14	91	NA
773	17-12-2014	TAMILNADU	TRICHY	CENTRAL BUS STAND	TAMILNADU STATE POLLUTION CONTROL BOARD	RESIDENTIAL ,RURAL AND OTHER AREAS	19	22	100	NA
773	24-12-2014	TAMILNADU	TRICHY	CENTRAL BUS STAND	TAMILNADU STATE POLLUTION CONTROL BOARD	RESIDENTIAL ,RURAL AND OTHER AREAS	15	17	95	NA
773	31-12-2014	TAMILNADU	TRICHY	CENTRAL BUS STAND	TAMILNADU STATE POLLUTION CONTROL BOARD	RESIDENTIAL ,RURAL AND OTHER AREAS	14	16	94	NA

The above is the given dataset of Air Quality Analysis in Tamil Nadu in 2014. This dataset consists of 11 columns and 2880 rows.

NECESSARY STEPS TO FOLLOW

1.Import Libraries:

Start by importing the neccessary libraries. Load your dataset into the pandas dataframe. And then display the output.

Program:

```
# importing pandas module for data frame
import pandas as pd

# loading dataset and storing in train variable
train=pd.read_csv('cpcb_dly_aq_tamil_nadu-2014.csv')

# display top 5 data
train.head()
```

Output:

STN CODE	SAMPLING DATE	STATE	CITY	MONITORING STATION	AGENCY	TYPE OF LOCATION	SO2	NO2	RSPM/PM10	PM2.5
38	02-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	11	17	55	NA
38	07-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	13	17	45	NA
38	21-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	12	18	50	NA
38	23-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	15	16	46	NA
38	28-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	13	14	42	NA

2.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.)
```

3.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:

```
# Extract date and time components
data['hour'] = data['timestamp'].dt.hour
data['day_of_week'] = data['timestamp'].dt.dayofweek

 # Lag features
data['pm25_lag_1'] = data['pm25'].shift(1)
 # Rolling statistics
```

```
data['pm25_rolling_mean'] = data['pm25'].rolling(window=7).mean()


# Interaction features
data['pm25_temperature_interaction'] = data['pm25'] *
data['temperature']
```

4.Split the Data:

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

Program:

```
# Define your features (X) and target (y)
 X = data[['Year', 'Month', 'Daily_PM2.5_Avg']]
y = data['AirQualityCategory']


 # Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

5.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)
```

6.Data Transformation:

Scaling, normalization, and categorical variable encoding are common preprocessing techniques. These transformations ensure that data is in a suitable format for machine learning algorithms, guaranteeing consistent scales across features.

Program:

```
#Transform the test data using the same scaler
X_test_scaled = scaler.transform(X_test)
```

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.Preprocess the dataset:

Once the dataset is loaded into the machine learning environment, you may need to preprocess 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 datavinto training and test sets.

PROGRAM:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score,
mean_absolute_error,mean_squared_error
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
import xgboost as xg
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
/opt/conda/lib/python3.10/site-packages/scipy/_init_.py:146:
UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for
this version of SciPy (detected version 1.23.5
warnings.warn(f"A NumPy version >={np_minversion} and
<{np_maxversion} "
```

Loading dataset:

```
dataset = pd.read_csv('cpcb_dly_aq_tamil_nadu-2014.csv')
```

OUTPUT:

STN CODE	SAMPLING DATE	STATE	CITY	MONITORING STATION	AGENCY	TYPE OF LOCATION	SO2	NO2	RSPM/PM10	PM2.5
38	02-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	11	17	55	NA
38	07-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	13	17	45	NA
38	21-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	12	18	50	NA
38	23-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	15	16	46	NA
38	28-01-2014	TAMILNADU	CHENNAI	KATHIVAKKAM	TAMILNADU STATE POLLUTION CONTROL BOARD	INDUSTRIAL AREA	13	14	42	NA
....	....	....	....	....	....	....	....	....	....	....
773	03-12-2014	TAMILNADU	TRICHY	CENTRAL BUS STAND	TAMILNADU STATE POLLUTION CONTROL BOARD	RESIDENTIAL ,RURAL AND OTHER AREAS	15	18	102	NA
773	10-12-2014	TAMILNADU	TRICHY	CENTRAL BUS STAND	TAMILNADU STATE POLLUTION CONTROL BOARD	RESIDENTIAL ,RURAL AND OTHER AREAS	12	14	91	NA
773	17-12-2014	TAMILNADU	TRICHY	CENTRAL BUS STAND	TAMILNADU STATE POLLUTION CONTROL BOARD	RESIDENTIAL ,RURAL AND OTHER AREAS	19	22	100	NA
773	24-12-2014	TAMILNADU	TRICHY	CENTRAL BUS STAND	TAMILNADU STATE POLLUTION CONTROL BOARD	RESIDENTIAL ,RURAL AND OTHER AREAS	15	17	95	NA
773	31-12-2014	TAMILNADU	TRICHY	CENTRAL BUS STAND	TAMILNADU STATE POLLUTION CONTROL BOARD	RESIDENTIAL ,RURAL AND OTHER AREAS	14	16	94	NA

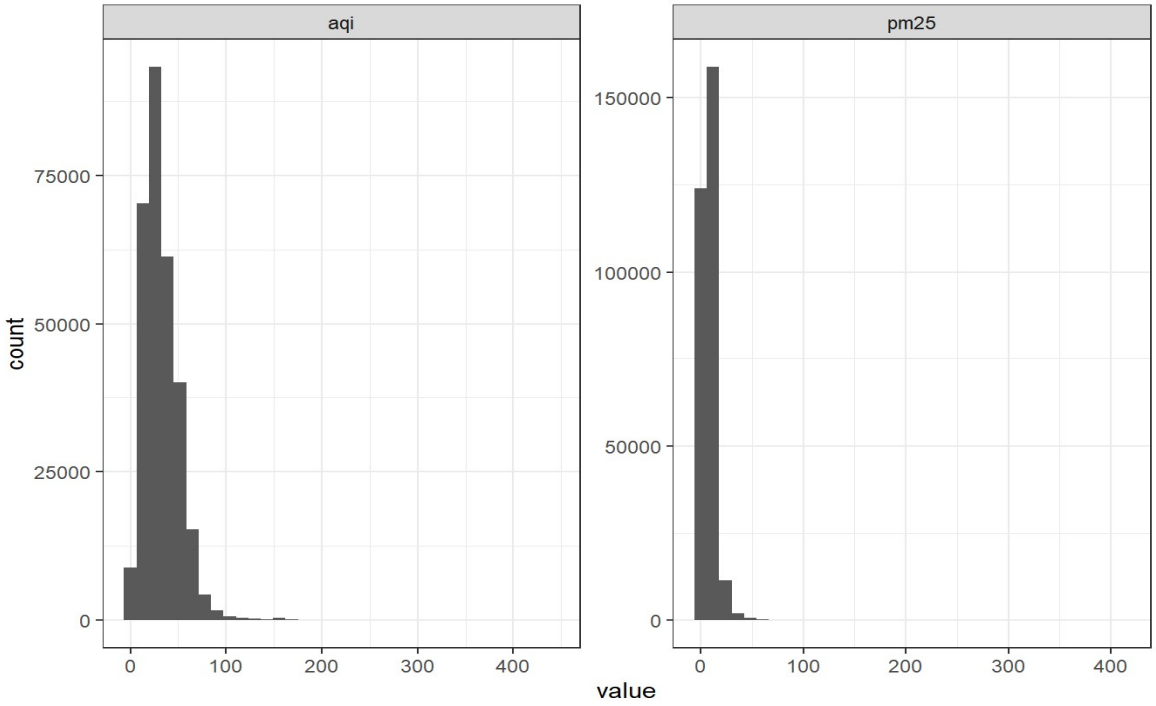
PREPROCESSING THE DATASET

- Data preprocessing is the process of cleaning, transforming, and integrating data in order to make it ready for analysis.
- Preprocessing often involves feature selection, creation, and transformation. This crucial step allows us to identify the most relevant variables, reduce dimensionality, and adapt the data to better suit our modeling goals.

Program 1:

```
## histogram for aqi and pm2.5
vis <-
  pmaqi %>%
  select(pm25, aqi) %>%
  gather(key = class, value = value)
  ggplot(data = vis) +
  geom_histogram(aes(x = value), bins = 35) +
  facet_wrap(~ class, scales = "free") +
  theme_bw()
```

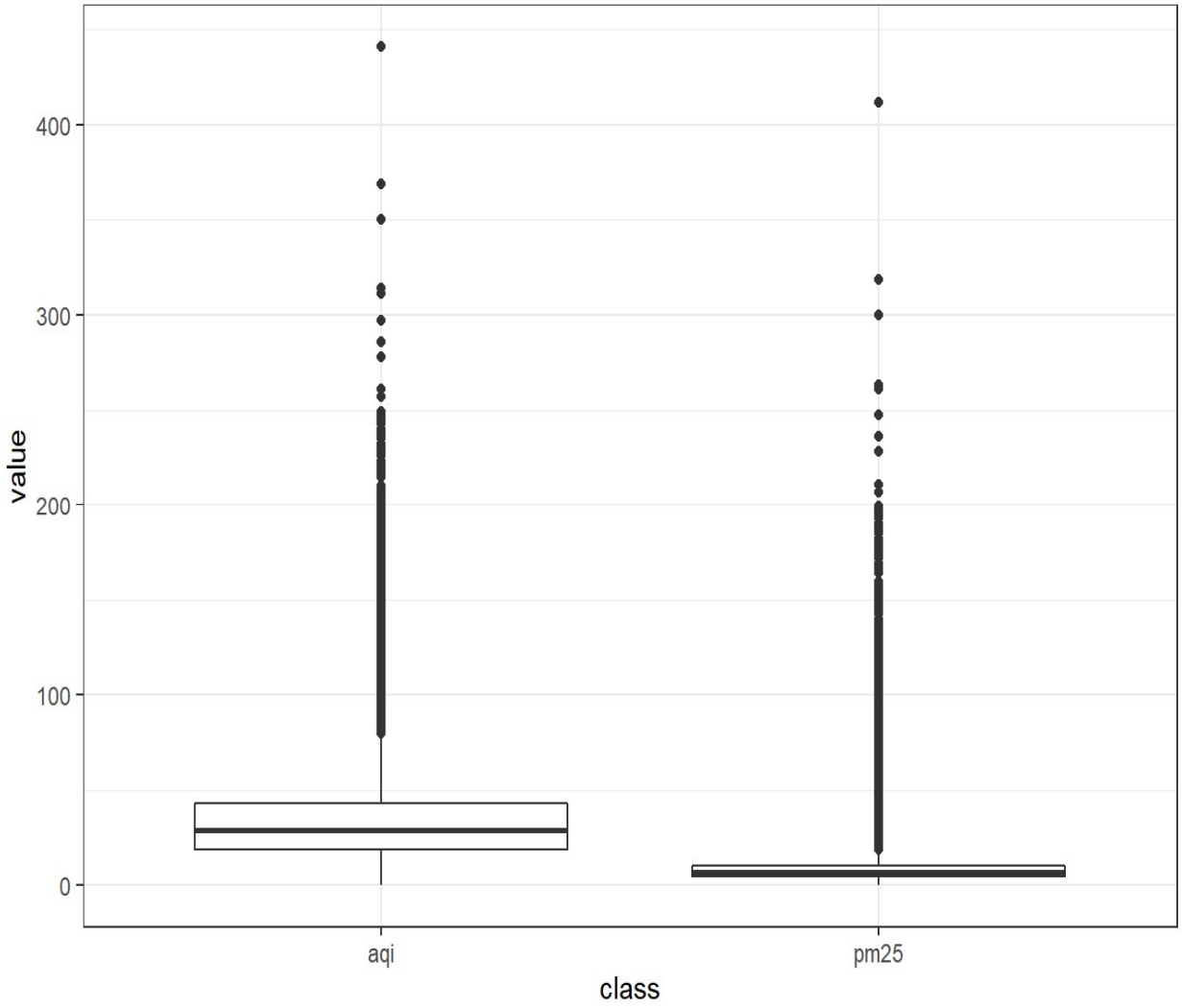
Output:



Program 2:

```
ggplot(data = vis) +
  geom_boxplot(aes(x =class,  y = value)) +
  theme_bw()
```

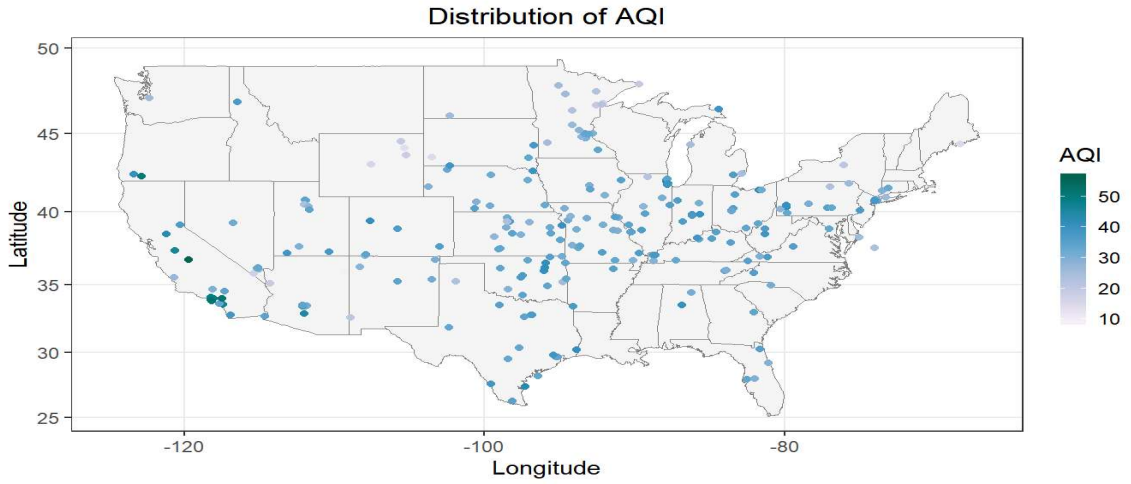
Output:



Program 3:

```
#for aqi index
vis2 <- vis[vis$class == "aqi", ]
ggplot() +
 geom_polygon(aes(x = long, y = lat, group = group),
           fill = "whitesmoke", colour = "gray50", size = 0.1,
           data = states) +
 geom_point(aes(x = long, y = lat, color = value), data = vis2) +
 scale_color_distiller("AQI", palette = "PuBuGn", direction = 0) +
 xlab("Longitude") + ylab("Latitude") +
 ggtitle("Distribution of AQI") +
 coord_map() +
 theme_bw() +
 theme(plot.title = element_text(hjust = 0.5),
     legend.key.width = unit(4, "mm"))
```

Output:



CONCLUSION

- Loading and preprocessing the dataset are pivotal phases in any data analysis project, and they hold equal importance in the context of an air quality analysis project. These initial steps are instrumental in shaping the trajectory of the project and the quality of the insights that can be extracted.

- By loading the dataset, we establish the foundation for data analysis and machine learning. It allows us to confirm that the data is accessible and formatted correctly, laying the groundwork for reliable analysis. Preprocessing encompasses data cleaning, where missing values, duplicate records, and outliers are addressed. Clean data is the cornerstone of accurate analysis and modeling.

- In conclusion, dataset loading and preprocessing are not merely preliminary chores but the cornerstones of successful air quality analysis. They set the stage for rigorous, reliable, and informative analysis, offering the potential to yield actionable insights that can contribute to the betterment of air quality, public health, and environmental well-being. The care and precision applied to these phases directly influence the strength and significance of the project's outcomes.