AIR QUALITY ANALYSIS IN TAMILNADU 2014

Data Analytics with Cognos

Phase 3 : Development Part 1

INTRODUCTION:

Analyzing air quality data is crucial for understanding environmental conditions and their impact on public health. In this project, we will explore the process of loading, preprocessing, and manipulating air quality data for Tamilnadu in the year 2014. Python, with its powerful libraries and tools, will be our chosen platform for this task. By the end of this analysis, we aim to gain insights into the air quality trends and potential correlations with various factors for a comprehensive understanding of the environmental situation in Tamilnadu during 2014.

**INPUT DATASET LINK** : [**https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil-nadu-year-2014**](https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil-nadu-year-2014)

LOADING THE DATASET

1.**IMPORT REQUIRED LIBRARY**:

* Import the necessary Python libraries for data manipulation. Common libraries include Pandas for data handling and Matplotlib or Seaborn for data visualization.

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

2.**LOAD THE DATASET**:

* Use Pandas to load the data from the source file into a DataFrame.

df=pd.read\_csv('C://aqa.csv')

3.**DATA EXPLORATION**:

* Explore the loaded data to understand its structure and contents. You can use functions like **head()**, **info()**, and **describe()** to get an initial sense of the dataset.

print(df.head())

print(df.info())

DATA PREPROCESSING:

1.**HANDLING MISSING DATA**:

* Identify and handle missing values. You can use the **isna()** function to detect missing values and **fillna()** or **dropna()** to fill or drop them, respectively.

df.drop(['Stn Code','Agency','State'],axis=1,inplace=True)

2.**DATA CONVERSION**:

* Ensure that the data types of columns are appropriate for analysis. For example, convert date columns to datetime format and numeric columns to appropriate data types.

df['SO2']=df['SO2'].fillna(0).astype('str').astype('float')

**3.FEATURE ENGINEERING**:

* Create new features if needed. For air quality analysis, you might want to extract year, month, or day of the week from the date column.

df['year'] = pd.to\_datetime(df['year'])

df['Year'] = df['year'].dt.year

**4.DATA SPLITTING**:

* If you plan to build predictive models, split the data into training and testing sets.

X = df.drop('NO2', axis=1)

y = df['NO2']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**5.SAVE PREPROCESSED DATA**:

* Save the preprocessed data to a new file if needed, so you can use it for analysis in the future.

df.to\_csv(‘aqat.csv’,index=False)

DATA MANIPULATION:

**1.AGGREGATION AND GROUPING:**

* If your dataset contains multiple records for the same location and date, you can aggregate or group the data to get summary statistics.

no2\_mean = df.groupby('City/Town/Village/Area')['NO2'].mean()

print(no2\_mean)

**2.CALCULATING STATISTICS:**

* Calculate various statistics to better understand the data. You can calculate means, medians, standard deviations, and more.

max\_so2 = df.groupby('City/Town/Village/Area')['SO2'].max()

print(max\_so2)

**3.SORTING DATA:**

* Sort the data based on specific columns for analysis or visualization.

sorted\_data = df.sort\_values(by='year')

print(sorted\_data)

**4.VISUALIZATION:**

df.set\_index('City/Town/Village/Area',inplace=True)

df['RSPM/PM10'].plot(title='RSPM/PM10 concentration in tamilnadu')

plt.xlabel('City/Town/Village/Area')

plt.ylabel('RSPM/PM10 concentration')

plt.show()

**5.SAVE MANIPULATED DATA:**

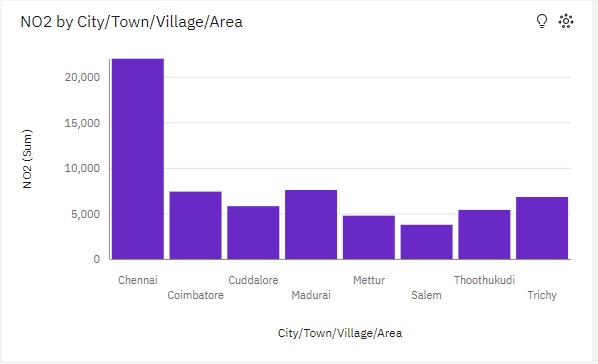
* If you've created a new dataset through manipulation, consider saving it to a file for future use.

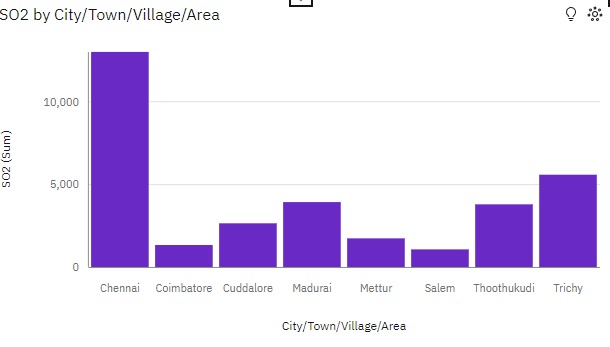
df.to\_csv('manipulated\_data.csv', index=False)

DATA VISUALIZATION USING COGNOS:

\*Use color in your table or crosstab visualizations to see the distribution of your data and highlight exceptional data points.

**\***Create basic visualizations such as line graphs, bar graphs, and pie charts using Excel spreadsheets.





PYTHON CODE:

#IMPORT REQUIRED LIBRARY

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

#LOAD THE DATASET

df = pd.read\_csv('C:\\aqa.csv')

#DATA EXPLORATION

print("\nFIRST FIVE ROWS:")

print(df.head())

print("\nINFORMATION ABOUT THE DATA:")

print(df.info())

#HANDLING MISSING DATA

df.drop(['Stn Code','Agency','State'],axis=1,inplace=True)

#DATA CONVERSION

df['SO2']=df['SO2'].fillna(0).astype('str').astype('float')

df['NO2']=df['NO2'].fillna(0).astype('str').astype('float')

df['RSPM/PM10']=df['RSPM/PM10'].fillna(0).astype('str').astype('float')

df['PM 2.5']=df['PM 2.5'].fillna(0).astype('str').astype('float')

df=df.rename(index=str,columns={'Sampling Date':'year'})

#SAVE PREPROCESSED DATA

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

#AGGREGATION AND GROUPING

print("\nMEAN OF NO2:")

no2\_mean = df.groupby('City/Town/Village/Area')['NO2'].mean()

print(no2\_mean)

print("\nMEAN OF SO2:")

so2\_mean = df.groupby('City/Town/Village/Area')['SO2'].mean()

print(so2\_mean)

print("\nMEAN OF RSPM/PM10:")

rspm\_pm10\_mean = df.groupby('City/Town/Village/Area')['RSPM/PM10'].mean()

print(rspm\_pm10\_mean)

print("\nMEAN OF PM 2.5:")

pm25\_mean = df.groupby('City/Town/Village/Area')['PM 2.5'].mean()

print(pm25\_mean)

#CALCULATING STATISTICS

print("\nMAX OF NO2:")

max\_no2 = df.groupby('City/Town/Village/Area')['NO2'].max()

print(max\_no2)

print("\nMAX OF SO2:")

max\_so2 = df.groupby('City/Town/Village/Area')['SO2'].max()

print(max\_so2)

print("\nMAX OF RSPM/PM10:")

max\_rspm\_pm10 = df.groupby('City/Town/Village/Area')['RSPM/PM10'].max()

print(max\_rspm\_pm10)

print("\nMAX OF PM 2.5:")

max\_pm25 = df.groupby('City/Town/Village/Area')['PM 2.5'].max()

print(max\_pm25)

#SORTING DATA

print("\nSORTED DATA:")

sorted\_data = df.sort\_values(by='year')

print(sorted\_data)

#VISUALIZATION

df.set\_index('City/Town/Village/Area',inplace=True)

df['RSPM/PM10'].plot(title='RSPM/PM10 in tamilnadu')

plt.xlabel('City/Town/Village/Area')

plt.ylabel('RSPM/PM10 concentration')

plt.show()

#SAVE MANIPULATED DATA

df.to\_csv('manipulated\_data.csv', index=False)

OUTPUT:

FIRST FIVE ROWS:

Stn Code Sampling Date State ... NO2 RSPM/PM10 PM 2.5

0 38 01-02-2014 Tamil Nadu ... 17.0 55.0 NaN

1 38 01-07-2014 Tamil Nadu ... 17.0 45.0 NaN

2 38 21-01-2014 Tamil Nadu ... 18.0 50.0 NaN

3 38 23-01-2014 Tamil Nadu ... 16.0 46.0 NaN

4 38 28-01-2014 Tamil Nadu ... 14.0 42.0 NaN

[5 rows x 11 columns]

INFORMATION ABOUT THE DATA:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 2879 entries, 0 to 2878

Data columns (total 11 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Stn Code 2879 non-null int64

1 Sampling Date 2879 non-null object

2 State 2879 non-null object

3 City/Town/Village/Area 2879 non-null object

4 Location of Monitoring Station 2879 non-null object

5 Agency 2879 non-null object

6 Type of Location 2879 non-null object

7 SO2 2868 non-null float64

8 NO2 2866 non-null float64

9 RSPM/PM10 2875 non-null float64

10 PM 2.5 0 non-null float64

dtypes: float64(4), int64(1), object(6)

memory usage: 247.5+ KB

None

MEAN OF NO2:

City/Town/Village/Area

Chennai 21.978000

Coimbatore 25.238908

Cuddalore 19.577703

Madurai 25.768707

Mettur 23.185366

Salem 28.664122

Thoothukudi 18.385666

Trichy 18.542234

Name: NO2, dtype: float64

MEAN OF SO2:

City/Town/Village/Area

Chennai 12.975000

Coimbatore 4.525597

Cuddalore 8.905405

Madurai 13.319728

Mettur 8.429268

Salem 8.114504

Thoothukudi 12.901024

Trichy 15.168937

Name: SO2, dtype: float64

MEAN OF RSPM/PM10:

City/Town/Village/Area

Chennai 58.998000

Coimbatore 48.713311

Cuddalore 61.881757

Madurai 45.724490

Mettur 52.721951

Salem 62.954198

Thoothukudi 83.174061

Trichy 85.054496

Name: RSPM/PM10, dtype: float64

MEAN OF PM 2.5:

City/Town/Village/Area

Chennai 0.0

Coimbatore 0.0

Cuddalore 0.0

Madurai 0.0

Mettur 0.0

Salem 0.0

Thoothukudi 0.0

Trichy 0.0

Name: PM 2.5, dtype: float64

MAX OF NO2:

City/Town/Village/Area

Chennai 55.0

Coimbatore 41.0

Cuddalore 26.0

Madurai 71.0

Mettur 69.0

Salem 55.0

Thoothukudi 27.0

Trichy 44.0

Name: NO2, dtype: float64

MAX OF SO2:

City/Town/Village/Area

Chennai 49.0

Coimbatore 9.0

Cuddalore 17.0

Madurai 25.0

Mettur 13.0

Salem 12.0

Thoothukudi 18.0

Trichy 30.0

Name: SO2, dtype: float64

MAX OF RSPM/PM10:

City/Town/Village/Area

Chennai 211.0

Coimbatore 164.0

Cuddalore 114.0

Madurai 199.0

Mettur 182.0

Salem 120.0

Thoothukudi 269.0

Trichy 262.0

Name: RSPM/PM10, dtype: float64

MAX OF PM 2.5:

City/Town/Village/Area

Chennai 0.0

Coimbatore 0.0

Cuddalore 0.0

Madurai 0.0

Mettur 0.0

Salem 0.0

Thoothukudi 0.0

Trichy 0.0

Name: PM 2.5, dtype: float64

SORTED DATA:

year City/Town/Village/Area ... RSPM/PM10 PM 2.5

0 01-02-2014 Chennai ... 55.0 0.0

2512 01-02-2014 Trichy ... 102.0 0.0

508 01-02-2014 Chennai ... 61.0 0.0

624 01-02-2014 Chennai ... 83.0 0.0

2661 01-02-2014 Trichy ... 53.0 0.0

... ... ... ... ... ...

623 31-12-2014 Chennai ... 42.0 0.0

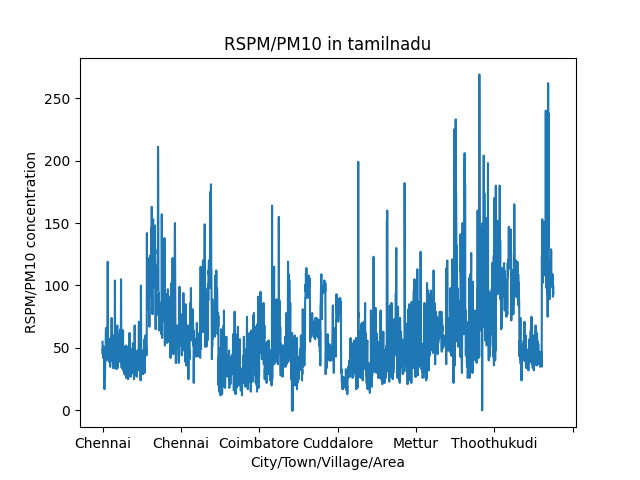
1785 31-12-2014 Madurai ... 25.0 0.0

1588 31-12-2014 Cuddalore ... 44.0 0.0

2218 31-12-2014 Salem ... 57.0 0.0

2878 31-12-2014 Trichy ... 94.0 0.0

[2879 rows x 8 columns]



CONCLUSION:

In conclusion, this project demonstrates the essential steps in loading, preprocessing, and manipulating air quality data using Python. Through the course of this analysis, we've successfully prepared the data for in-depth exploration, which can lead to valuable insights and informed decision-making. Understanding air quality trends and their implications in Tamilnadu in 2014 is not only beneficial for environmental studies but also for the well-being of the population. The skills and techniques learned in this project can be applied to a wide range of data analysis tasks, contributing to a more informed and sustainable future.