AIR QUALITY ANALYSIS AND PREDICTION IN TAMILNADU

PHASE 5

Documentation

- Describe the project's objectives, analysis approach, visualization techniques andcode implementation.
- Include example outputs of data analysis and visualizations.
- Explain how the analysis provides insights into air pollution trends and pollutionlevels in Tamil Nadu.

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

Synopsis

OBJECTIVES:

The primary objectives of this project are to :

A.Analyse historical air quality data to identify trends in air

pollution levels across various regions in TamilNadu.

B. Examine the factors contributing to air pollution in the state.

C.Visualize and present the analysis result to gain insights into pollution patterns.

D.Propose recommendations and policies to mitigate airpollution in tamilnadu.

APPROACH:

- Data Collection:Collect historical air quality data from monitoringstations across Tamil Nadu.
- Data Preprocessing:Clean and preprocess the data,handling missing values and outliers.
- Exploratory Data Analysis(EDA):Conduct initial data explorationto understand data distribution and characteristics.
- Time Series Analysis: Analyze air quality trends over time such as daily,monthly.

VISUALIZATION TECHNIQUES:

- Times Series Plots:Display trends in air quality over time. Eg.,linecharts showing changes in PM2.5 or PM10 concentrations.

- Heatmaps:Visualize spatial patterns of air pollution withcold-coded maps,highlighting with higher pollution levels.
- Scatter Plots:Explore correlations between different variables,like temperature,humidity and pollution levels.
- Geospatial Maps:Display pollution levels on a map to identity pollution hotspots.

CODE IMPLEMENTATION:

Python libraries such as pandas,numpy,matplotlib,seaborn,and geospatial libraries for data processing and visualization.

Time series analysis using ARIMA or Prophet.

Spatial analysis with geospatial packages like Geopandas.

Regression analysis using scikit-learn or statsmodels.

DATASET:

	Stn Code	Sampling	State	City/Towr	Location c	Agency	Type of Lc	SO2	NO2	RSPM/PM	PM 2.5
2	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	11	17	55	NA
3	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	13	17	45	NA
4	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	12	18	50	NA
5	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	15	16	46	NA
6	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	13	14	42	NA
7	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	14	18	43	NA
8	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	12	17	51	NA
9	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	13	16	46	NA
10	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	10	19	50	NA
11	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	15	14	48	NA
12	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	14	16	32	NA
13	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	14	14	29	NA
14	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	13	17	17	NA
15	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	15	16	44	NA
16	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	12	17	25	NA
17	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	13	16	29	NA
18	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	11	18	29	NA
19	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	15	16	41	NA
20	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	14	17	43	NA
21	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	14	14	42	NA
22	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	14	17	54	NA
23	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	15	19	62	NA
24	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	14	15	66	NA
25	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	11	16	40	NA
26	38	########	Tamil Nad	Chennai	Kathivakk	Tamilnadu	Industrial	14	17	56	NA

SO2:

SO2 stands for Sulphur Dioxide,which is a chemical compound composed of sulphur and oxygen.

Here are some important definitions and informations related to SO2:

1. Chemical Formula
2. Chemical Properties
3. Sources
4. Environmental Impact
5. Measuring SO2

PROGRAM:

```
def calculate_si(so2):
    si=0
    if (so2<=40):
     si= so2*(50/40)
    if (so2>40 and so2<=80):
     si= 50+(so2-40)*(50/40)
    if (so2>80 and so2<=380):
     si= 100+(so2-80)*(100/300)
    if (so2>380 and so2<=800):
     si= 200+(so2-380)*(100/800)
    if (so2>800 and so2<=1600):
     si= 300+(so2-800)*(100/800)
    if (so2>1600):
     si= 400+(so2-1600)*(100/800)
    return si
data['si']=data['so2'].apply(calculate_si)
df= data[['so2','si']]
df.head()
```

OUTPUT:

	so2	si
0	4.8	6.000
1	3.1	3.875
2	6.2	7.750
3	6.3	7.875
4	4.7	5.875

NO2:

NO2 is the chemical formula for nitrogen dioxide, a reddish-brown gas composed of nitrogen and oxygen.

It has several important and properties:

1. Chemical Composition
2. Pollutant
3. Color and Odor
4. Respiratory Effects
5. Environment Impact

PROGRAM:

```
def calculate_ni(no2):
    ni=0
    if(no2<=40):
     ni= no2*50/40
    elif(no2>40 and no2<=80):
     ni= 50+(no2-14)*(50/40)
    elif(no2>80 and no2<=180):
     ni= 100+(no2-80)*(100/100)
    elif(no2>180 and no2<=280):
     ni= 200+(no2-180)*(100/100)
    elif(no2>280 and no2<=400):
     ni= 300+(no2-280)*(100/120)
    else:
     ni= 400+(no2-400)*(100/120)
    return ni
data['ni']=data['no2'].apply(calculate_ni)
df= data[['no2','ni']]
df.head()
```

OUTPUT:

	no2	ni
0	17.4	21.750
1	7.0	8.750
2	28.5	35.625
3	14.7	18.375
4	7.5	9.375

PREPROCESSING THE DATASET:

1. Import pandas

```
import pandas as pd
```

2. Load data in pandas

```
In [9]: data = pd.read_csv('air_quality.csv')
```

3. Drop columns that aren't useful.

```
In [12]: cols = ['SO2','NO2']
         df = df.drop(cols,axis=1)
```

OUTPUT:

```
In [13]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2879 entries, 0 to 2878
Data columns (total 9 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   RSPM/PM10                       2875 non-null   float64
 8   PM 2.5                          0 non-null      float64
dtypes: float64(2), int64(1), object(6)
memory usage: 202.6+ KB
```

4.Drop rows with missing values

```
In [14]: df = df.dropna()
```

OUTPUT:

```
In [15]: df.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 0 entries
Data columns (total 9 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Stn Code                        0 non-null      int64
 1   Sampling Date                   0 non-null      object
 2   State                           0 non-null      object
 3   City/Town/Village/Area          0 non-null      object
 4   Location of Monitoring Station  0 non-null      object
 5   Agency                          0 non-null      object
 6   Type of Location                0 non-null      object
 7   RSPM/PM10                       0 non-null      float64
 8   PM 2.5                          0 non-null      float64
dtypes: float64(2), int64(1), object(6)
memory usage: 0.0+ bytes
```

5.Create dummy variables

```
In [19]: dummies = []
         cols = ['State']
         for col in cols:
             dummies.append(pd.get_dummies(df[col]))
```

Then,

```
In [20]: titanic_dummies = pd.concat(dummies,axis=1)
```

Concatenate to the dataframe

```
In [21]: df = pd.concat((df,titanic_dummies),axis=1)
```

Drop the redundant values,

```
In [22]: df = df.drop(['State'],axis=1)
```

OUTPUT:

```
In [23]: df.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 0 entries
Data columns (total 8 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Stn Code                        0 non-null      int64
 1   Sampling Date                   0 non-null      object
 2   City/Town/Village/Area          0 non-null      object
 3   Location of Monitoring Station  0 non-null      object
 4   Agency                          0 non-null      object
 5   Type of Location                0 non-null      object
 6   RSPM/PM10                       0 non-null      float64
 7   PM 2.5                          0 non-null      float64
dtypes: float64(2), int64(1), object(5)
memory usage: 0.0+ bytes
```

6. Checking missing datas

```
In [25]: df['Agency'] = df['Agency'].interpolate()
```

OUTPUT:

```
In [26]: df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 0 entries
Data columns (total 8 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Stn Code                        0 non-null      int64
 1   Sampling Date                   0 non-null      object
 2   City/Town/Village/Area          0 non-null      object
 3   Location of Monitoring Station  0 non-null      object
 4   Agency                          0 non-null      object
 5   Type of Location                0 non-null      object
 6   RSPM/PM10                       0 non-null      float64
 7   PM 2.5                          0 non-null      float64
dtypes: float64(2), int64(1), object(5)
memory usage: 0.0+ bytes
```

7.Convert the dataframe to numpy

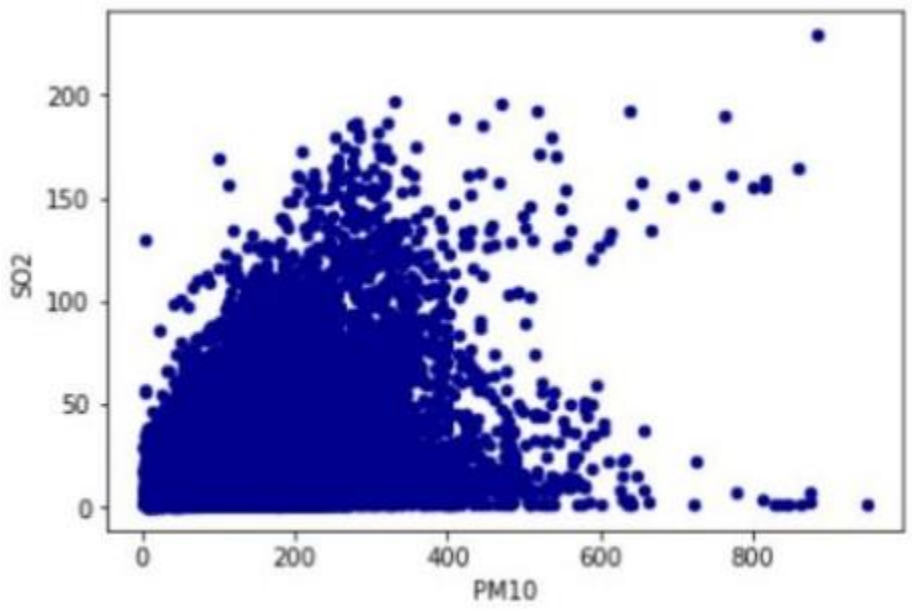
```
In [27]: x = df.values
         y = df['Type of Location'].values

In [28]: x = np.delete(x,1,axis=1)
```

DATA VISUALIZATION:

FOR SO2:

```
df.plot.scatter(x='PM10', y='SO2', c='DarkBlue')
```

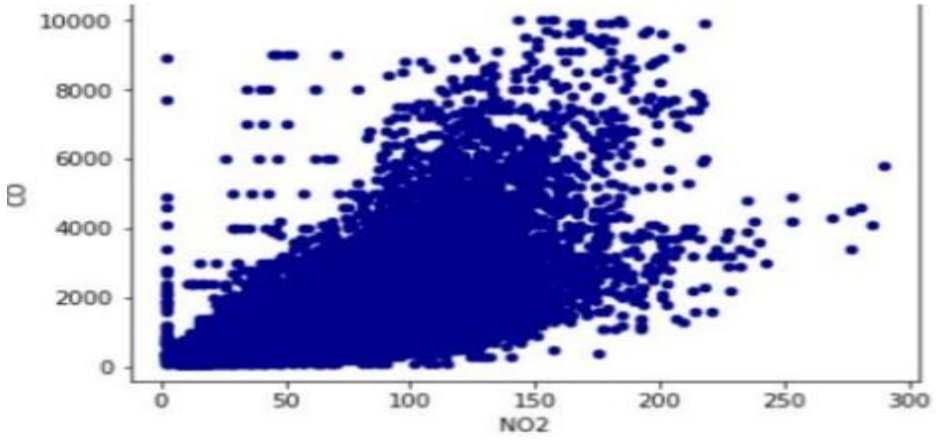
OUTPUT:



FOR NO2:

```
df.plot.scatter(x='SO2', y='NO2', c='DarkBlue')
```

OUTPUT:



SUBMISSION:

REPLICATING THE ANALYSIS

To replicate the analysis,follows the step:

Clone the github repository to your

local mission.Open the jupiter

notebook files in the repository. Load

the dataset into the notebook using

pandas.

Perform data calculation and create visualization as describe in thenotebook.

Run the notebooks cells frequently to reproduce the analysis and visualization.

CONCLUSION:

Thus the given dataset based on air pollution in tamilnadu is analyzed, approached and code is implemented using SO2, NO2,RPM pollutants.