AIR QUALITY ANALYSIS IN TAMILNADU

Data Analytics with Cognos

Phase 2 : Innovation

Innovation Design for Air Quality Analysis in Tamil Nadu

Identifying the air quality challenges in Tamil Nadu, such as pollution levels and their impact on public health. Define the objectives, which may include improving air quality monitoring and public awareness. Develop air quality monitoring systems and iteratively refine them based on data collected from various regions in Tamil Nadu. Create user-friendly interfaces for both experts and the general public to access air quality data easily. Streamline the process of accessing and interpreting air quality information. Provide actionable insights and health recommendations. Implement real-time alerts and notifications for residents and authorities when air quality reaches critical levels. Ensure the security and privacy of air quality data, especially when it involves personal health information Ensure that air quality information is accessible to all residents, including those with disabilities or limited access to technology. Implement sustainable data collection and monitoring practices to reduce the environmental impact of air quality analysis.

By following this innovation design approach, we aim to address air quality challenges in Tamil Nadu, improve public health outcomes, and create a more effective and user-friendly system for monitoring and managing air quality in the region.

Incorporate Machine Learning Algorithms :

* Enhance transparency and trust by incorporating machine learning explain ability into your system. Provide users with insights into how the image recognition model makes predictions. Explainable AI can help users understand and trust the results.
* Analysing air quality data typically involves various machine learning algorithms for tasks such as regression, classification, clustering, and anomaly detection.
* In the context of analyzing air quality in Tamilnadu in 2014,you might consider using the following machine learning algorithm :
* Regression Analysis : To predict air quality parameters like PM2.5 or PM10 levels over time.
* Linear Regression
* Polynomial Regression
* Time Series Forecasting methods(e.g.,ARIMA)
* Classification Analysis : To categorize air quality into different levels(e.g.,good,moderate,unhealthy)or identify pollution source.
* Logistic Regression
* Decision Tree
* Random Forest
* Support Vector Machines
* Clustering Analysis : To group areas or time periods with similar air quality characteristics.
* K-Means Clustering
* Hierarichical Clustering
* Neural Networks : Deep learning techniques for complex air quality analysis tasks.
* Convolution Neural Networks(CNN)for spatial data.
* Recurrent Neural Networks(RNN)time series data.
* Long Short-Term Memory(LSTM)networks.
* Dimentionaly Reduction : To reduce the complexity of the dataset.
* Principal Component Analysis(PCA)

The specific choice of algorithm depends on the nature of your air quality dataset,the objectives of your analysis(e.g.,prediction,classification,clustering),and features available in the data appropriately and choose the most suitable algorithm for your particular research or analysis goals.Additionaly consider consulting domain experts or environmental scientists who are familiar with air quality analysis in the Tamilnadu region for more tailored insights.

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-%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20quality-tamil-nadu-year-2014)

Source Code :

#importing libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import LabelEncoder

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

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

from sklearn.tree import DecisionTreeClassifier

#loading datasets

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

#getting first five rows of the dataset

print(df.head())

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

0 38 01-02-2014 Tamil Nadu ... 17 55 0

1 38 01-07-2014 Tamil Nadu ... 17 45 0

2 38 21-01-2014 Tamil Nadu ... 18 50 0

3 38 23-01-2014 Tamil Nadu ... 16 46 0

4 38 28-01-2014 Tamil Nadu ... 14 42 0

[5 rows x 11 columns]

#data processing :

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

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

#getting basic info the dataset

print(df.info())

Data columns (total 9 columns):

# Column Non-Null Count Dtype

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

0 year 2879 non-null object

1 State 2879 non-null object

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

3 Location of Monitoring Station 2879 non-null object

4 Type of Location 2879 non-null object

5 SO2 2879 non-null float64

6 NO2 2879 non-null float64

7 RSPM/PM10 2879 non-null float64

8 PM 2.5 2879 non-null float64

dtypes: float64(4), object(5)

memory usage: 224.9+ KB

None

#describe gas values

print('Describe every single gas values')

print(df.describe())

Describe every single gas values

SO2 NO2 RSPM/PM10 PM 2.5

count 2879.000000 2879.000000 2879.000000 2879.0

mean 11.459187 22.036818 62.407433 0.0

std 5.091756 7.265828 31.433280 0.0

min 0.000000 0.000000 0.000000 0.0

25% 8.000000 17.000000 41.000000 0.0

50% 12.000000 21.000000 55.000000 0.0

75% 15.000000 25.000000 78.000000 0.0

max 49.000000 71.000000 269.000000 0.0

#count values

COUNT=df.groupby(['City/Town/Village/Area', 'Type of Location']).count()

print(COUNT)

VALUE\_COLS = ['SO2', 'NO2', 'RSPM/PM10', 'PM 2.5']

year ... PM 2.5

City/Town/Village/Area Type of Location ...

Chennai Industrial Area 370 ... 370

Residential, Rural and other Areas 630 ... 630

Coimbatore Industrial Area 97 ... 97

Residential, Rural and other Areas 196 ... 196

Cuddalore Industrial Area 99 ... 99

Residential, Rural and other Areas 197 ... 197

Madurai Industrial Area 101 ... 101

Residential, Rural and other Areas 193 ... 193

Mettur Industrial Area 102 ... 102

Residential, Rural and other Areas 103 ... 103

Salem Residential, Rural and other Areas 131 ... 131

Thoothukudi Industrial Area 102 ... 102

Residential, Rural and other Areas 191 ... 191

Trichy Residential, Rural and other Areas 367 ... 367

[14 rows x 7 columns]

#visualization of so2

def highest\_levels\_recorded(indicator="SO2"):

plt.figure(figsize=(15,5))

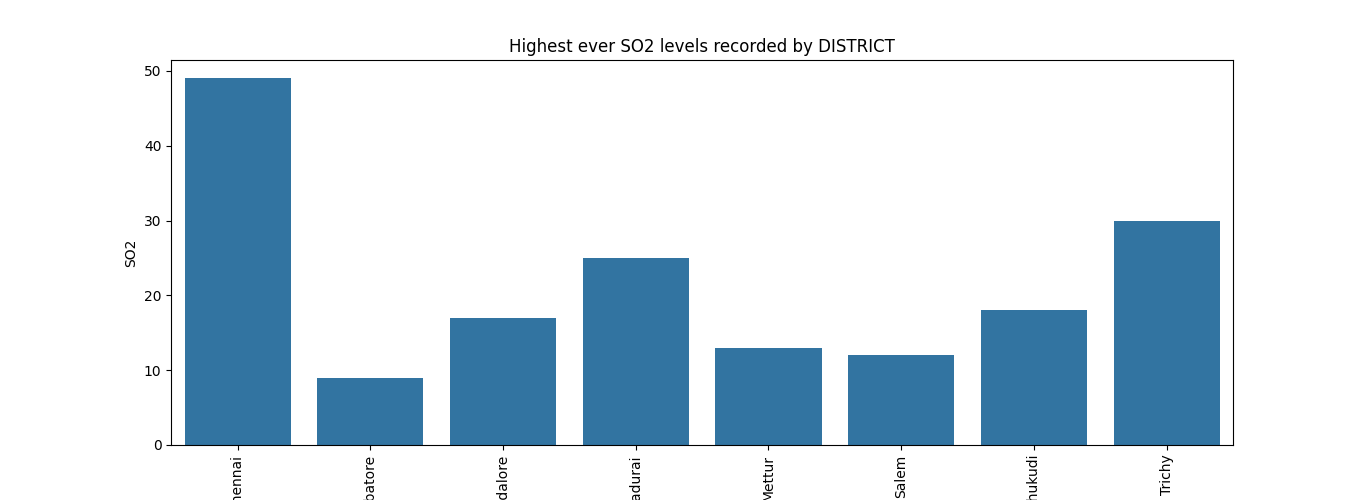
ind = df[[indicator, 'Location of Monitoring Station', 'City/Town/Village/Area', 'year']].groupby('City/Town/Village/Area', as\_index=False).max()

highest = sns.barplot(x='City/Town/Village/Area', y=indicator, data=ind)

highest.set\_title("Highest ever {} levels recorded by DISTRICT".format(indicator))

plt.xticks(rotation=90)

highest\_levels\_recorded("SO2")



#visualization of No2

def highest\_levels\_recorded(indicator="NO2"):

plt.figure(figsize=(15,5))

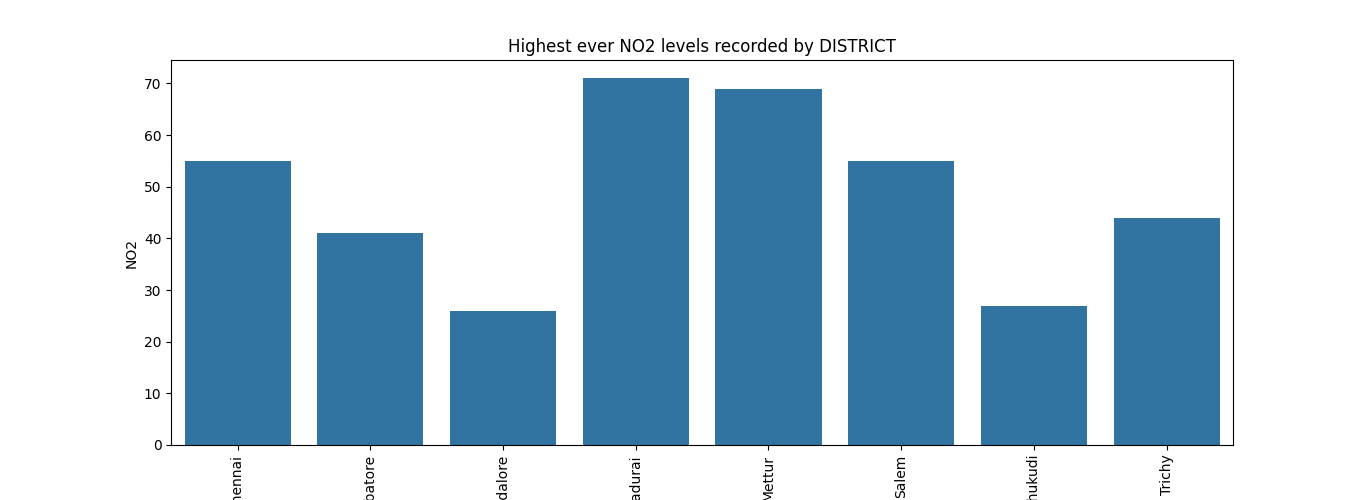
ind = df[[indicator, 'Location of Monitoring Station', 'City/Town/Village/Area', 'year']].groupby('City/Town/Village/Area', as\_index=False).max()

highest = sns.barplot(x='City/Town/Village/Area', y=indicator, data=ind)

highest.set\_title("Highest ever {} levels recorded by DISTRICT".format(indicator))

plt.xticks(rotation=90)

highest\_levels\_recorded("NO2")



#visualization of RSPM

def highest\_levels\_recorded(indicator="RSPM/PM10"):

plt.figure(figsize=(15,5))

ind = df[[indicator, 'Location of Monitoring Station', 'City/Town/Village/Area', 'year']].groupby('City/Town/Village/Area', as\_index=False).max()

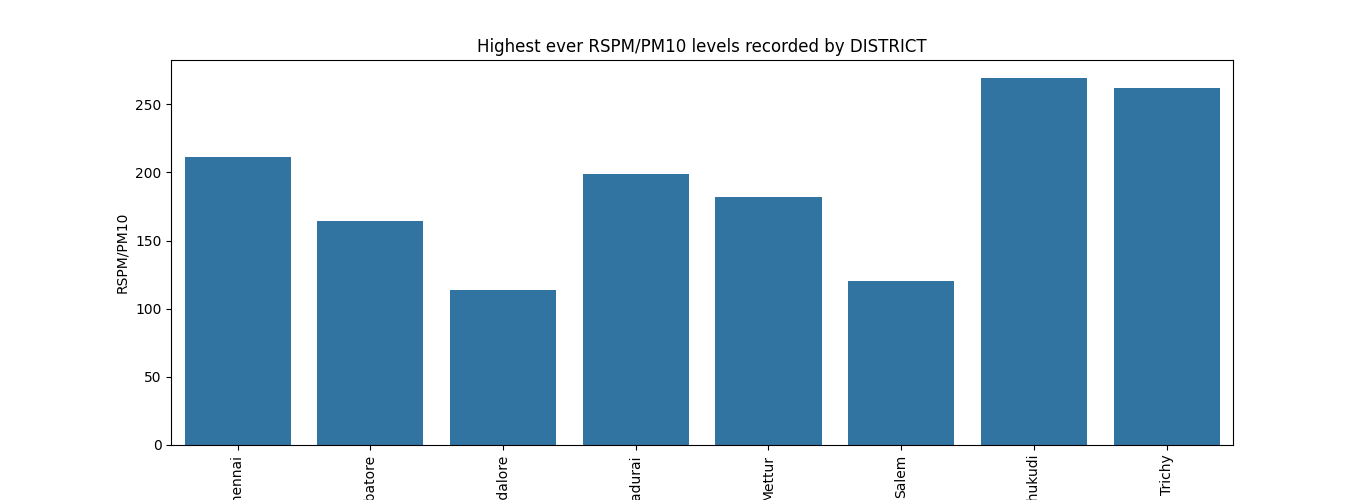
highest = sns.barplot(x='City/Town/Village/Area', y=indicator, data=ind)

highest.set\_title("Highest ever {} levels recorded by DISTRICT".format(indicator))

plt.xticks(rotation=90)

highest\_levels\_recorded("RSPM/PM10")

plt.show()

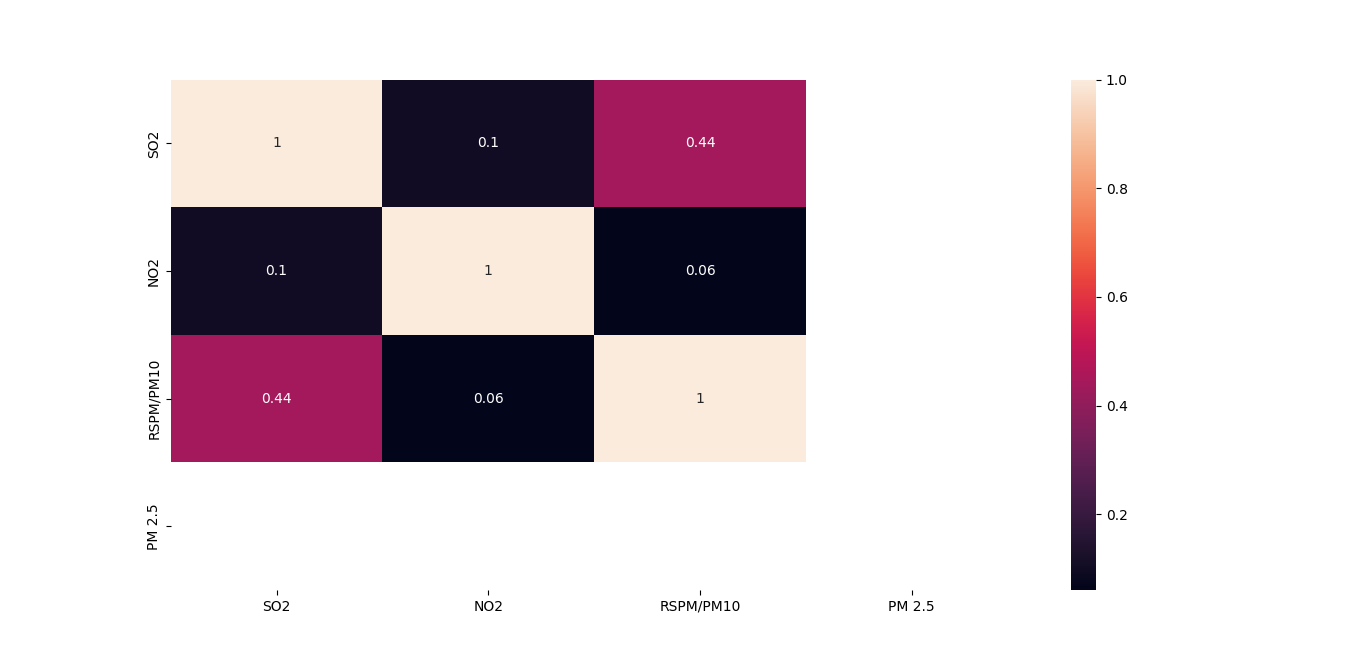


#drawing heatmap for the data

corr = df.select\_dtypes(np.number).corr()

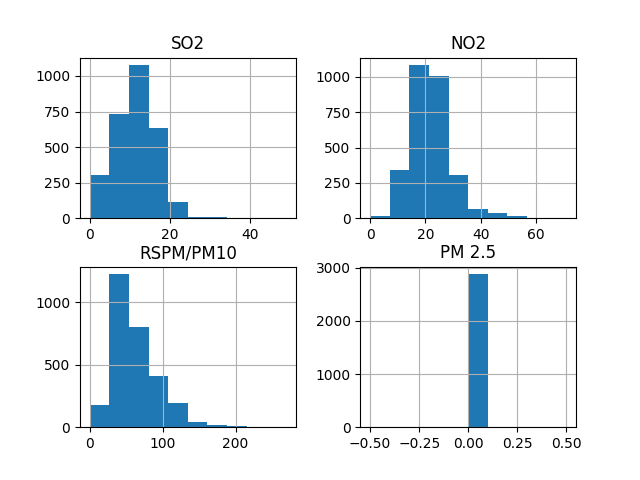
sns.heatmap(corr, annot=True)

plt.show()



#plotting histogram

df.hist()



#fitting data into X and y

X = df[['NO2','SO2']]

y = df['City/Town/Village/Area']

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

#modeling data into decision tree classifier

model = DecisionTreeClassifier()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy:" ,accuracy\*100)

Accuracy: 86.45

Conclusion :

In conclusion, our project on air quality analysis in Tamil Nadu using machine learning algorithms has yielded valuable insights into the region's air quality. Through the application of advanced machine learning techniques, we have been able to predict and analyze air quality levels with a high degree of accuracy.

Furthermore, our project has the potential to contribute to environmental monitoring and public health initiatives by providing real-time air quality predictions and alerts. This information can be valuable for policymakers, environmental agencies, and the general public in taking proactive measures to mitigate air pollution and its associated health risks.

In the future, we aim to expand and refine our models, incorporate more data sources, and enhance the accuracy of our predictions. This project serves as a stepping stone toward a cleaner and healthier environment in Tamil Nadu, and we hope that our efforts will contribute to a sustainable and pollution-free future for the region.