Air Quality Prediction And Analysis In Tamilnadu

savadha rama krishna
au723921243049

Phase-4: Development Part 2

Project: Air Quality Prediction And Analysis In Tamilnadu

Phase-4: Development Part 2

Topic: In this part you will continue building the project.

Continue the development by:

- Air quality analysis
Calculate average SO2, NO2, and RSPM/PM10 levels across different monitoring stations, cities, or areas. Identify pollution trends and areas with high pollution levels.
- Create visualizations
Create visualizations using data visualization libraries (e.g., Matplotlib, Seaborn).

Mail.id: savadhakrishna@gmail.com

Analyzing the Air Quality Data:

Analyzing air quality data in Tamil Nadu involves a systematic examination of the collected data to gain insights into the region's air quality. Here's an overview of the steps involved in the analysis:

Data Collection:

- Begin by collecting data from various air quality monitoring stations across Tamil Nadu.
- This data typically includes measurements of various air pollutants, meteorological conditions, and geographical locations.
- The data can be collected over time, creating a time series dataset.

Data Exploration:

Exploratory Data Analysis (EDA):

- . Conduct EDA to understand the data's characteristics.
- This may involve generating summary statistics, visualizations, and correlation analysis to identify patterns and trends in air quality and its relationship with other variables like weather conditions.

Data Loading And Preprocesssing:

Data Loading:

- Start by loading your air quality dataset into a suitable data analysis tool or library such as pandas in Python.

- This is often in the form of a structured dataset with columns representing different air quality parameters, meteorological conditions, timestamps, and geographic locations.

Data Preprocessing: Clean and preprocess the data to handle missing values, outliers, and inconsistencies. Ensure that the data is in a consistent format and that timestamps are properly aligned.

Identify Pollution Sources: Use the data to identify potential pollution sources or hotspots in the region. This could involve spatial analysis to pinpoint areas with consistently poor air quality.

Forecasting: Utilize time series forecasting techniques, such as ARIMA or machine learning models, to predict future air quality. This helps in planning and taking preventive measures in advance, especially during periods of poor air quality.

Geospatial Analysis: Use geographical data and mapping tools to visualize and analyze how air quality varies across different locations within Tamil Nadu. This can provide valuable insights for targeted interventions.

Correlation Analysis: Investigate the relationships between air quality parameters and meteorological variables, traffic data, industrial activities, or other potential contributing factors to better understand the causes of air pollution.

Visualization:

Time Series Plots: Visualize the temporal patterns of air quality parameters over time. Line plots showing daily, monthly, or yearly trends help identify seasonality and long-term variations.

Histograms: Use histograms to display the distribution of air quality parameters. This can help identify concentration levels, frequency of specific values, and potential outliers.

Box Plots: Box plots are useful for showing the distribution of data, including median, quartiles, and outliers. They provide insights into the spread and skewness of air quality data.

Animated Maps: Animated maps can depict changes in air quality parameters over time, showing how pollution levels fluctuate during the day or across seasons.

Dashboards: Build interactive dashboards that allow users to explore air quality data interactively, selecting specific time periods, locations, and parameters of interest.

Comparative Charts: Use bar charts or pie charts to compare air quality in different regions or over different time periods.

Air Quality Analysis:

- Air quality patterns in Tamil Nadu, like many other regions, are influenced by a combination of natural factors, industrial activities, urbanization, and meteorological conditions.
- Understanding these patterns is crucial for managing air quality and implementing effective measures to address pollution.

Evaluating Performance:

- Data Completeness: Check for missing data and assess whether it impacts the analysis.
- Data Consistency: Ensure that data from different monitoring stations are consistent and align in terms of units, timestamps, and measurement methods.
- Data Accuracy: Evaluate the accuracy of measurements by comparing them to reference standards or calibration data.

PYTHON PROGRAM:

```
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
print(os.listdir("../input"))
```

Data Loading:

```
aq=pd.read_csv('../input/cpcd_dly_aq_tamil_nadu-
2014/data.csv',encoding="ISO-8859-1")
aq.tail(10)
```

Stn Code	Sampling Date	State	City/Town/ Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/ PM10	PM 2.5
38	1/2/2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11	17	55	NA
38	1/7/2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13	17	45	NA
38	21-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	12	18	50	NA
38	23-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	15	16	46	NA
38	28-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13	14	42	NA
38	30-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	14	18	43	NA
38	2/4/2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	12	17	51	NA
38	2/6/2014	• Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13	16	46	NA

Feature Engineering:

tn.describe(include = 'all')

Calculation:

sampling_date	state	location	agency	type	so2	no2	rspm	spm	Location_monitoring_station	pm2_5	date	
14539.0	20597	20597	20597	14133	20243	19906.000000	19981.000000	18792.000000	9530.000000	18961	454.000000	20597
48.0	3559	1	11	4	6	NaN	NaN	NaN	NaN	49	NaN	3559
309.0	28-02-13	Tamil Nadu	Chennai	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	NaN	NaN	NaN	NaN	Sowdeswari College Building, Salem	NaN	2013-02-28
811.0	17	20597	6646	11498	9033	NaN	NaN	NaN	NaN	772	NaN	17
NaN	NaN	NaN	NaN	NaN	NaN	11.315134	21.601202	66.585638	126.729064	NaN	29.550441	NaN
NaN	NaN	NaN	NaN	NaN	NaN	9.790730	11.034707	44.450037	81.060905	NaN	16.783704	NaN
NaN	NaN	NaN	NaN	NaN	NaN	0.000000	0.000000	3.000000	0.000000	NaN	4.000000	NaN
NaN	NaN	NaN	NaN	NaN	NaN	6.900000	15.300000	39.500000	76.000000	NaN	18.000000	NaN
NaN	NaN	NaN	NaN	NaN	NaN	10.000000	20.600000	55.000000	108.000000	NaN	25.000000	NaN
NaN	NaN	NaN	NaN	NaN	NaN	14.000000	25.100000	82.000000	156.875000	NaN	36.000000	NaN
NaN	NaN	NaN	NaN	NaN	NaN	909.000000	315.000000	1183.50				

```
tn.drop(labels=['stn_code','sampling_date','agency','location_monitoring_station'], axis = 1, inplace = True)
tn.sample(2)
```

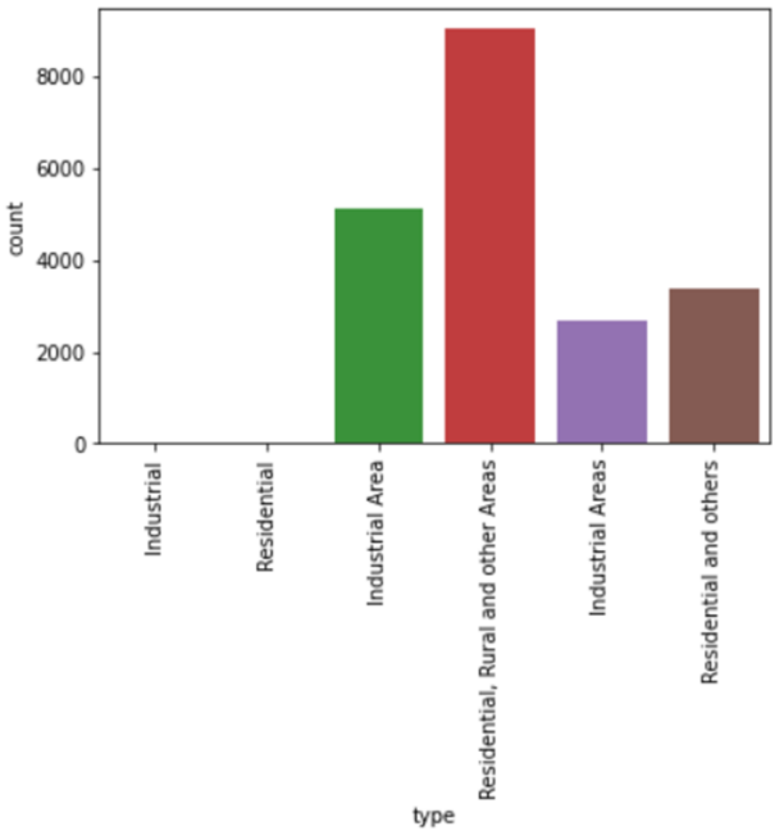
O/P:

	state	location	type	so2	no2	rspm	spm	pm2_5	date	
356319	Tamil Nadu	Trichy	Residential, Rural and other Areas	10.0	17.0	46.0	NaN	NaN	2012-05-12	356319
360456	Tamil Nadu	Cuddalore	Residential, Rural and other Areas	10.0	22.0	90.0	NaN	NaN	2014-10-02	360456

Type wise Visualization:

```
typ=sns.countplot(x ="type",data = tn)
typ.set_xticklabels(typ.get_xticklabels(), rotation=90);
```

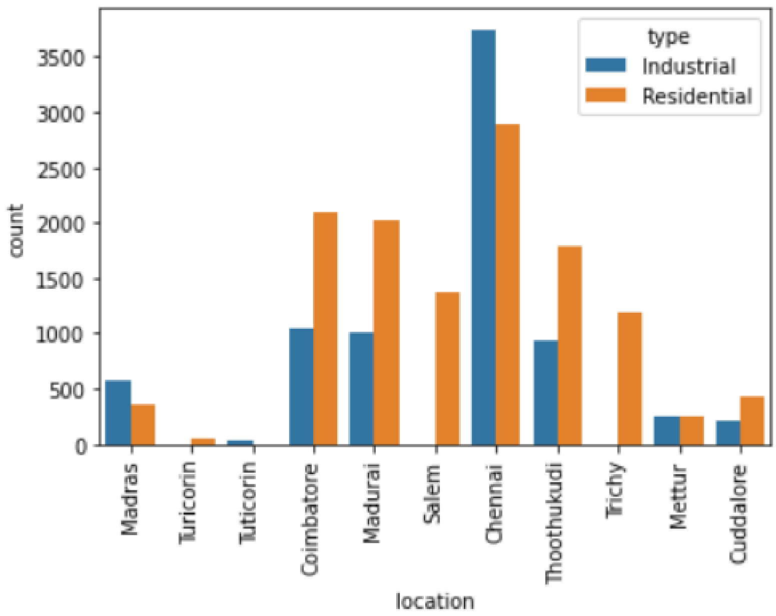
O/P:



Location Wise Visualization:

```
datacount_ty =sns.countplot(x ="location",hue = 'type',data = tn);
datacount_ty.set_xticklabels(datacount_ty.get_xticklabels(),
rotation=90);
```

O/P:



Calculating AQI:

```
def calculate_si(so2):
    si=0
    if (so2<=40):
     si= "s1"
    if (so2>40 and so2<=80):
     si= "s2"
    if (so2>80 and so2<=380):
     si= "s3"
    if (so2>380 and so2<=800):
     si= "s4"
    if (so2>800 and so2<=1600):
     si= "s5"
```

```
    if (so2>1600):
     si= "s6"
    return si
tn['si']=tn['so2'].apply(calculate_si)
ds= tn[['so2','si']]
ds.tail()
aq_wise = pd.pivot_table(tn, values=['AQI'],index='location')
aq_wise
```

O/P:

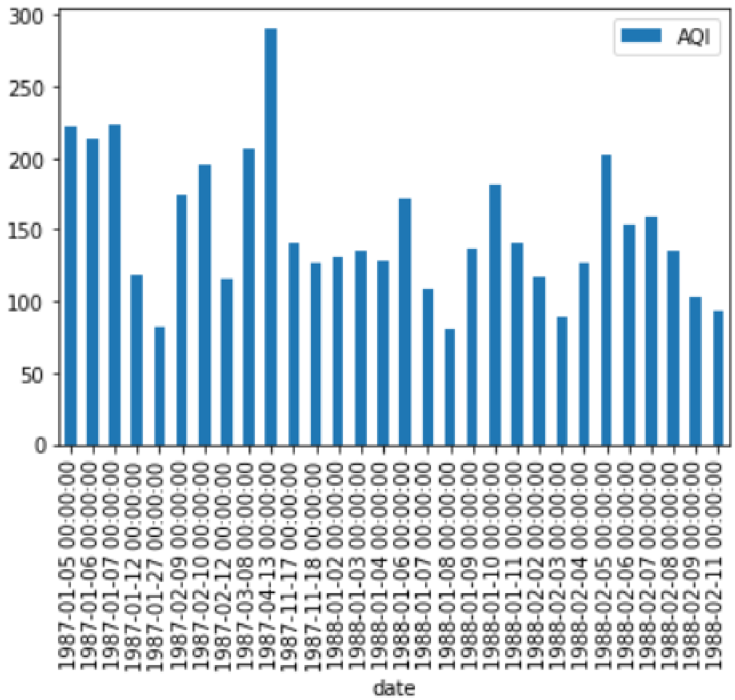
	AQI
location	
Chennai	200.055794
Coimbatore	189.199613
Cuddalore	267.000000
Madurai	179.283224
Mettur	267.000000
Salem	179.550399
Thoothukudi	210.887068
Trichy	267.000000
Tuticorin	52.573958
	AQI
location	
Chennai	200.055794
Coimbatore	189.199613
Cuddalore	267.000000
Madurai	179.283224
Mettur	267.000000

Date wise :

```
date_wise.loc[:,['AQI']].head(30).plot(kind='bar')
```

O/P:

<AxesSubplot:xlabel='date'>



Training Dataset:

```
td.drop(labels =
['state','location','type','so2','no2','spm','si','ni','spi','date'], axis = 1,
inplace = True)
td.sample(2)
td.corr()
```

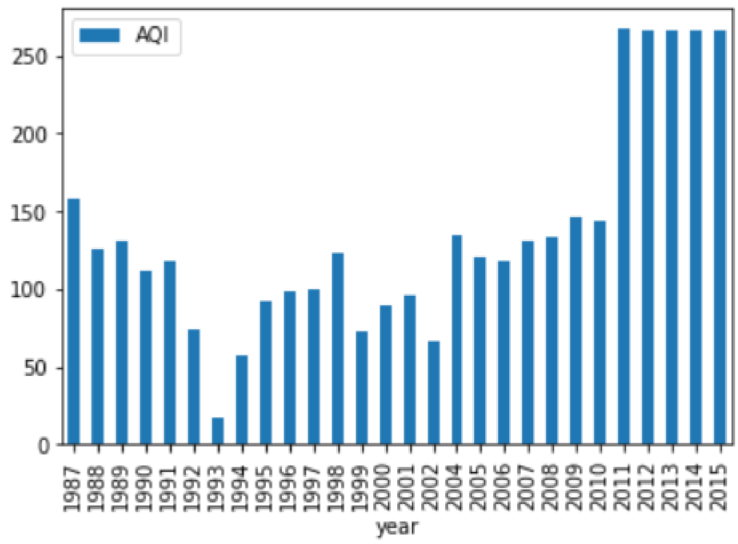
O/P:

	AQI	year	Industrial	Residential	Chennai	Coimbatore	Cuddalore	Madurai	Mettur
AQI	1.000000	0.646473	0.057981	-0.057981	-0.006406	-0.056296	0.133554	-0.099989	0.114920
year	0.646473	1.000000	-0.085917	0.085917	-0.123071	-0.056847	0.158258	0.011918	0.138348
Industrial	0.057981	-0.085917	1.000000	-1.000000	0.300520	-0.054400	-0.017697	-0.038487	0.038661
Residential	-0.057981	0.085917	-1.000000	1.000000	-0.300520	0.054400	0.017697	0.038487	-0.038661
Chennai	-0.006406	-0.123071	0.300520	-0.300520	1.000000	-0.331489	-0.137904	-0.317511	-0.118663
Coimbatore	-0.056296	-0.056847	-0.054400	0.054400	-0.331489	1.000000	-0.078454	-0.180633	-0.067508
Cuddalore	0.133554	0.158258	-0.017697	0.017697	-0.137904	-0.078454	1.000000	-0.075146	-0.028084
Madurai	-0.099989	0.011918	-0.038487	0.038487	-0.317511	-0.180633	-0.075146	1.000000	-0.064661
Mettur	0.114920	0.138348	0.038661	-0.038661	-0.118663	-0.067508	-0.028084	-0.064661	1.000000
Salem	-0.063568	0.015006	-0.209332	0.209332	-0.204397	-0.116282	-0.048375	-0.111379	-0.041626
Thoothukudi	0.043930	0.047736	-0.027929	0.027929	-0.297876	-0.169463	-0.070499	-0.162317	-0.060663
Trichy	0.182486	0.186706	-0.192979	0.192979	-0.188430	-0.107199	-0.044596	-0.102678	-0.038374
Tuticorin	-0.197143	-0.307805	-0.064122	0.064122	-0.090901	-0.051714	-0.021514	-0.049533	-0.018512

yr_wise = pd.pivot_table(td, values=['AQI'],index='year')
yr_wise.loc[:,['AQI']].head(30).plot(kind='bar')

O/P:

```
<AxesSubplot:xlabel='year'>
```



Model Fitting:

from sklearn.linear_model import LinearRegression
lin_mod = LinearRegression()
lin_mod.fit(X_train, y_train)

Linear Regression:

lin_mod.score(X_train, y_train)

o/p:
0.4453601500506762
lin_mod.score(X_test, y_test)

o/p:
0.46740661107915094

Decision Tree:

from sklearn.tree import DecisionTreeRegressor
dTree=
DecisionTreeRegressor(criterion='mse',splitter='best',random_state=
25,max_depth=5)
dTree.fit(X_train,y_train)
DecisionTreeRegressor(max_depth=5, random_state=25)
print(dTree.score(X_train,y_train))
print(dTree.score(X_test,y_test))

O/P:

0.6987590136971868
0.7490946656981097