PHASE 2-INNOVATION

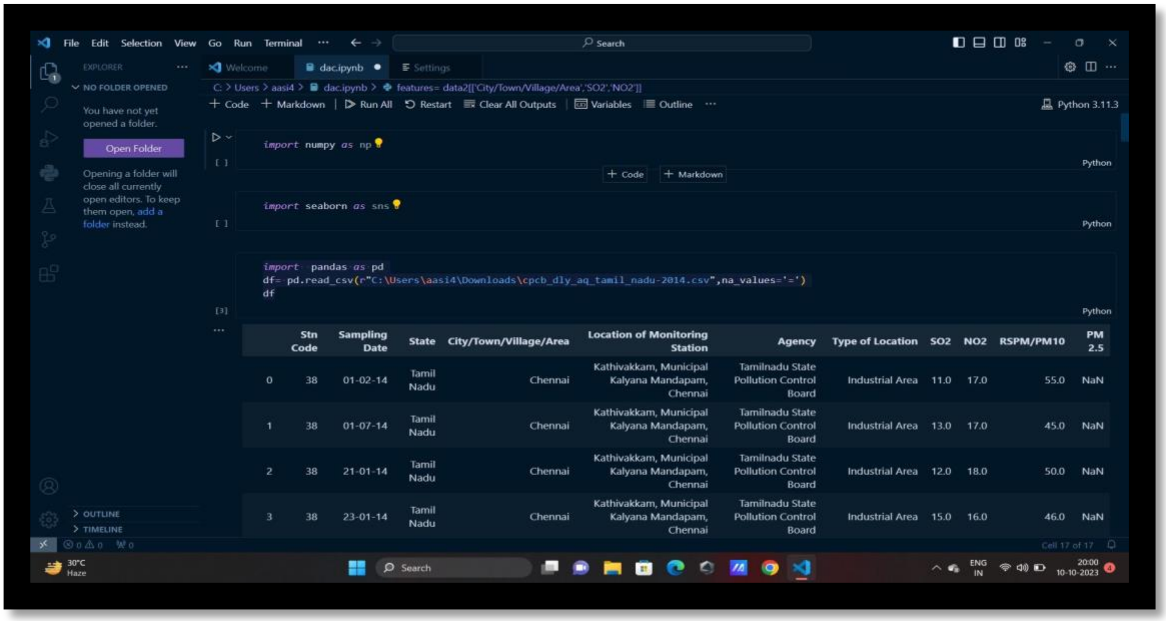
AIR QUALITY ANALYSIS IN TAMILNADU

- **Data Collection and Preparation:**

- Collect historical air quality data from monitoring stations in Tamil Nadu. Sources might include government agencies, environmental organizations, or publicly available datasets.

- Preprocess the data to handle missing values, outliers, and inconsistencies.

- Ensure that the data includes relevant features such as RSPM/PM10, SO2, and NO2 levels, along with location and timestamp information.



```
import numpy as np
```

```
import seaborn as sns
```

```
import  pandas as pd
df= pd.read_csv(r"C:\Users\aasi4\Downloads\cpcb_dly_aq_tamil_nadu-2014.csv",na_values='=')
df
```

[3]

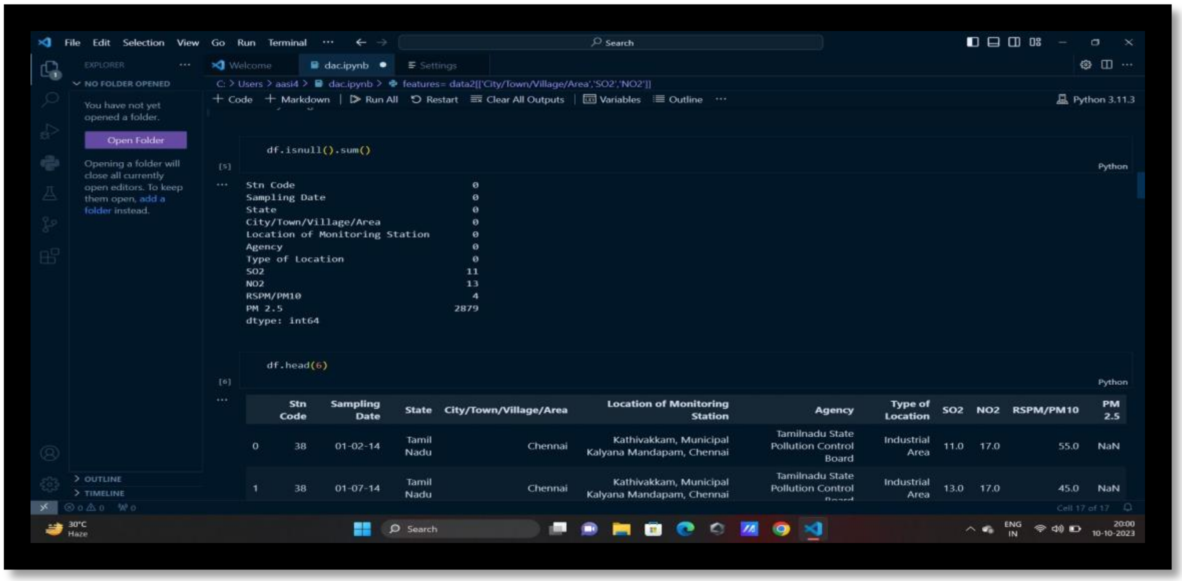
	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10	PM 2.5
0	38	01-02-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.0	17.0	55.0	NaN
1	38	01-07-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	17.0	45.0	NaN
2	38	21-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	12.0	18.0	50.0	NaN
3	38	23-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	15.0	16.0	46.0	NaN

- **Exploratory Data Analysis (EDA):**

- Perform EDA to gain initial insights into the data.

- Visualize air pollution trends over time, identifying patterns and seasonal variations.

- Use statistical measures and visualizations (e.g., line plots, heatmaps) to understand correlations and relationships between variables.

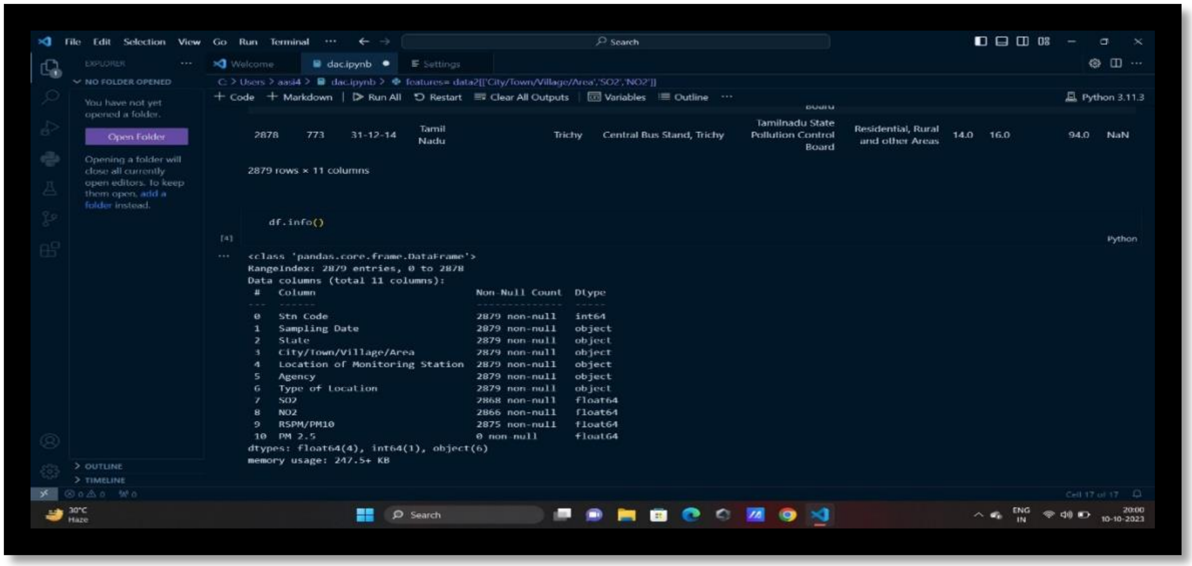

File Edit Selection View Go Run Terminal
Search
dac.ipynb
C: > Users > aasi4 > dac.ipynb > features= data2[['City/Town/Village/Area','SO2','NO2']]
+ Code + Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.11.3
Open Folder
Cell 17 of 17

```
df.isnull().sum()
```

```
Stn Code                            0
Sampling Date                       0
State                               0
City/Town/Village/Area              0
Location of Monitoring Station      0
Agency                              0
Type of Location                    0
SO2                                11
NO2                                13
RSPM/PM10                           4
PM 2.5                           2879
dtype: int64
```

```
df.head(6)
```

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10	PM 2.5
0	38	01-02-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.0	17.0	55.0	NaN
1	38	01-07-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control	Industrial Area	13.0	17.0	45.0	NaN


File Edit Selection View Go Run Terminal
Search
dac.ipynb
C: > Users > aasi4 > dac.ipynb > features= data2[['City/Town/Village/Area','SO2','NO2']]
+ Code + Markdown Run All Restart Clear All Outputs Variables Outline
Python 3.11.3
Open Folder
Cell 17 of 17

2878	773	31-12-14	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	14.0	16.0	94.0	NaN

2879 rows × 11 columns

```
df.info()
```

```
<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
```

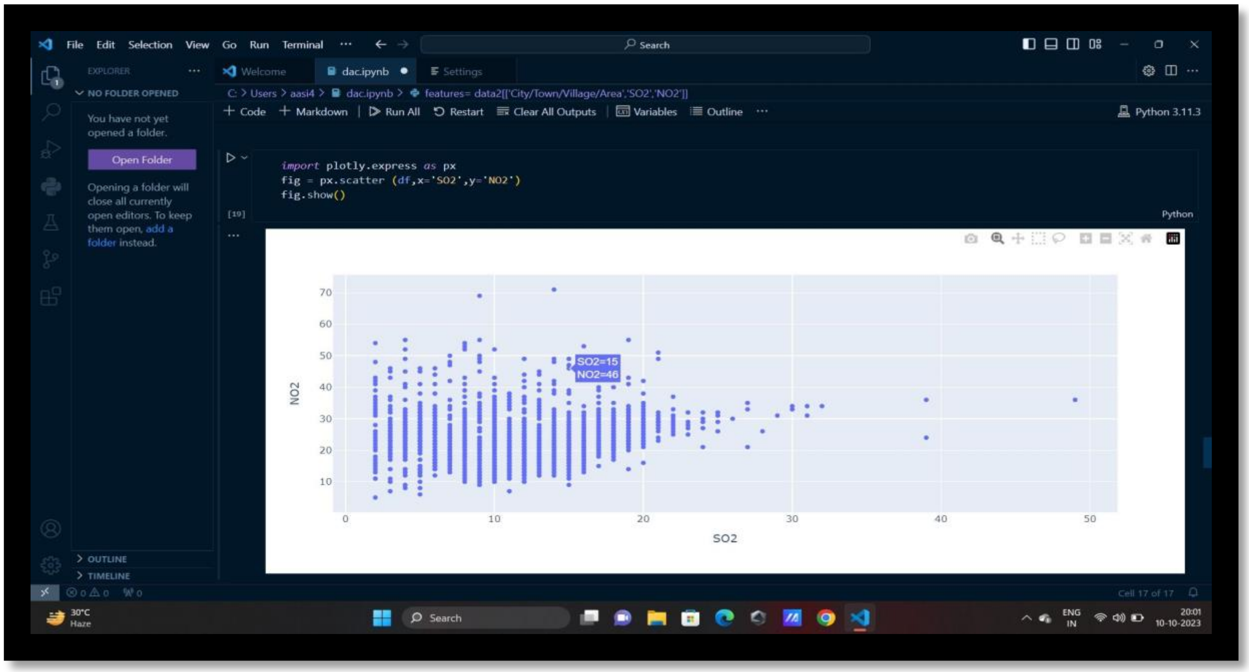
- **Data Visualization:**

- Select appropriate visualization techniques to effectively communicate your findings.

- Create interactive maps to visualize air quality across different regions in Tamil Nadu.

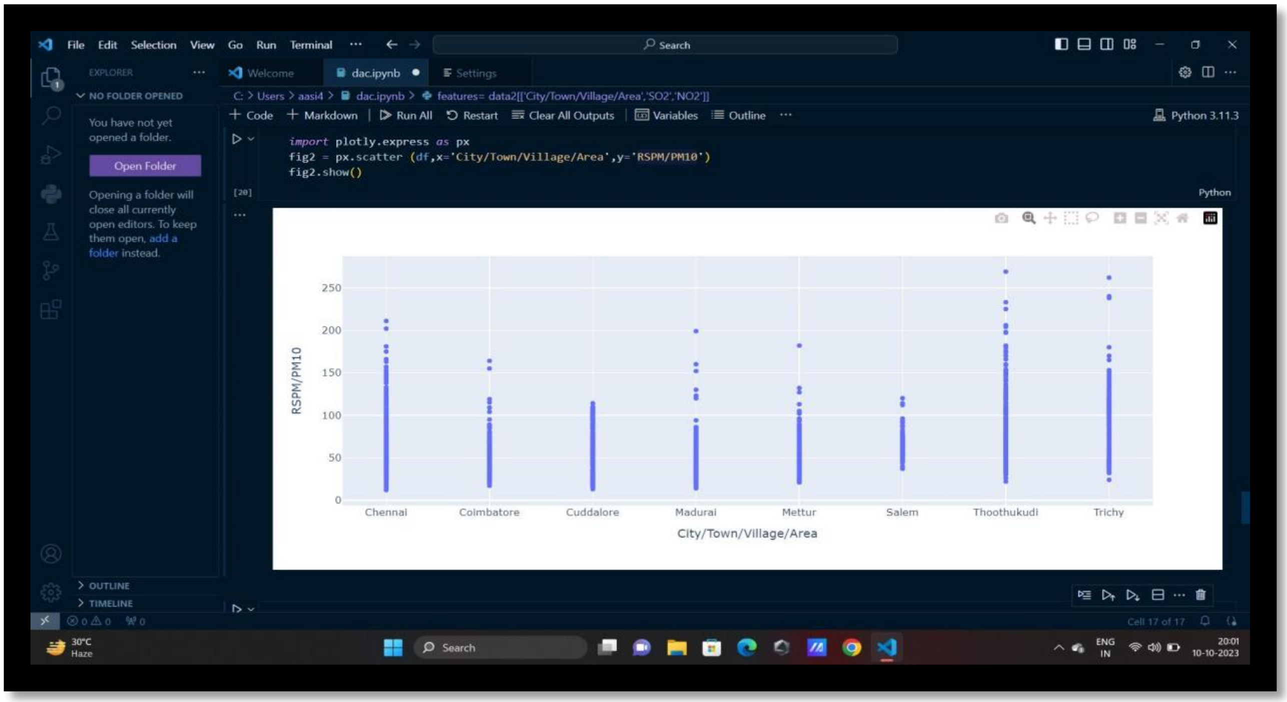
- Generate time series plots to highlight pollution trends and fluctuations.

- Utilize seaborn, matplotlib, or specialized libraries like Plotly for data visualization.



```
import plotly.express as px
fig = px.scatter (df,x='SO2',y='NO2')
fig.show()
```

File Edit Selection View Go Run Terminal
Search
EXPLORER
Welcome
dac.ipynb
Settings
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
C: > Users > aasi4 > dac.ipynb > features= data2[['City/Town/Village/Area','SO2','NO2']]
+ Code + Markdown | Run All Restart Clear All Outputs Variables Outline
Python 3.11.3
[19]
Python
SO2=15
NO2=46
70
60
50
40
30
20
10
NO2
0
10
20
30
40
50
SO2
OUTLINE
TIMELINE
Cell 17 of 17
30°C
Haze
Search
ENG
IN
20:01
10-10-2023



```
import plotly.express as px
fig2 = px.scatter (df,x='City/Town/Village/Area',y='RSPM/PM10')
fig2.show()
```

File Edit Selection View Go Run Terminal
Search
EXPLORER
Welcome
dac.ipynb
Settings
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
C: > Users > aasi4 > dac.ipynb > features= data2[['City/Town/Village/Area','SO2','NO2']]
+ Code + Markdown | Run All Restart Clear All Outputs Variables Outline
Python 3.11.3
[20]
Python
250
200
150
100
50
0
RSPM/PM10
Chennai
Coimbatore
Cuddalore
Madurai
Mettur
Salem
Thoothukudi
Trichy
City/Town/Village/Area
OUTLINE
TIMELINE
Cell 17 of 17
30°C
Haze
Search
ENG
IN
20:01
10-10-2023

- **Feature Engineering:**

  - Create additional features if needed, such as lagged values of pollution levels to capture time dependencies.

  - Calculate rolling statistics or moving averages to smooth data for modeling.

  - Normalize or scale features as necessary for modeling.

- **Model Development:**

  - Split the data into training and testing sets, considering time-based splitting if applicable.

  - Choose an appropriate machine learning algorithm for regression, as your goal is to predict RSPM/PM10 levels based on SO2 and NO2 levels.

  - Train and fine-tune the predictive model using libraries like scikit-learn or XGBoost.

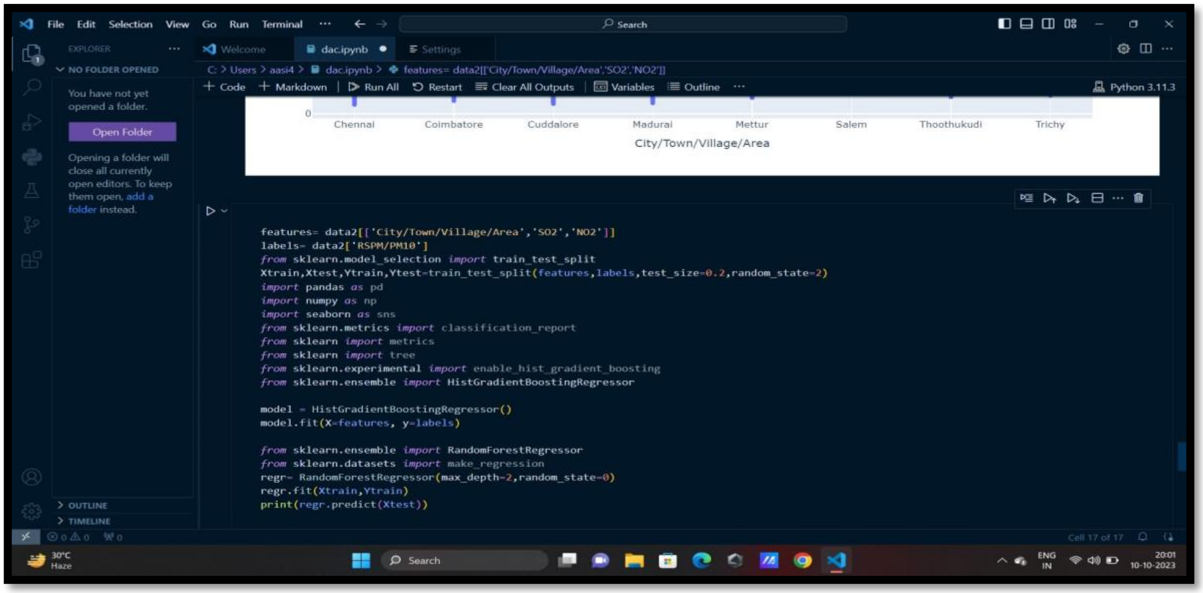
  - Evaluate the model's performance using relevant metrics like Mean Absolute Error (MAE) or Root Mean Square Error (RMSE).

- **Model Interpretation:**

  - Explain how the model makes predictions, considering feature importance analysis.

  - Visualize the model's predictions and compare them to actual values.

  - Assess the model's accuracy in estimating RSPM/PM10 levels based on SO2 and NO2 levels.



```
features= data2[['City/Town/Village/Area','SO2','NO2']]
labels= data2['RSPM/PM10']
from sklearn.model_selection import train_test_split
Xtrain,Xtest,Ytrain,Ytest=train_test_split(features,labels,test_size=0.2,random_state=2)
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.metrics import classification_report
from sklearn import metrics
from sklearn import tree
from sklearn.experimental import enable_hist_gradient_boosting
from sklearn.ensemble import HistGradientBoostingRegressor

model = HistGradientBoostingRegressor()
model.fit(X=features, y=labels)

from sklearn.ensemble import RandomForestRegressor
from sklearn.datasets import make_regression
regr= RandomForestRegressor(max_depth=2,random_state=0)
regr.fit(Xtrain,Ytrain)
print(regr.predict(Xtest))
```