Anna University Regional Campus Coimbatore

Anna University: Chennai-600 025

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

IBM Naan Mudhalvan Phase5 Submission

Title: AIR QUALITY ANALYSIS AND PREDICTION IN TAMILNADU

Name	: PRAVEEN S
Register Number:	710021106027
Department	:B.E.ECE
Sem/year	:V/III

Air Quality Analysis and Prediction in Tamil Nadu

Objective:

The objective of this project is to analyze and visualize air quality data from various monitoring stations in Tamil Nadu. The dataset contains measurement of Sulfur Dioxide (SO2), Nitrogen Dioxide (NO2), and Respirable Suspended Particulate Matter 10(RSPM/PM10) levels in different cities, towns, villages and areas. The project aims to gain insights into air pollution trends, identify areas with high pollution trends, identify areas with high pollution levels and create a predictive model to estimate RSPM/PM10 levels based on S02 and NO2 levels.

Abstract:

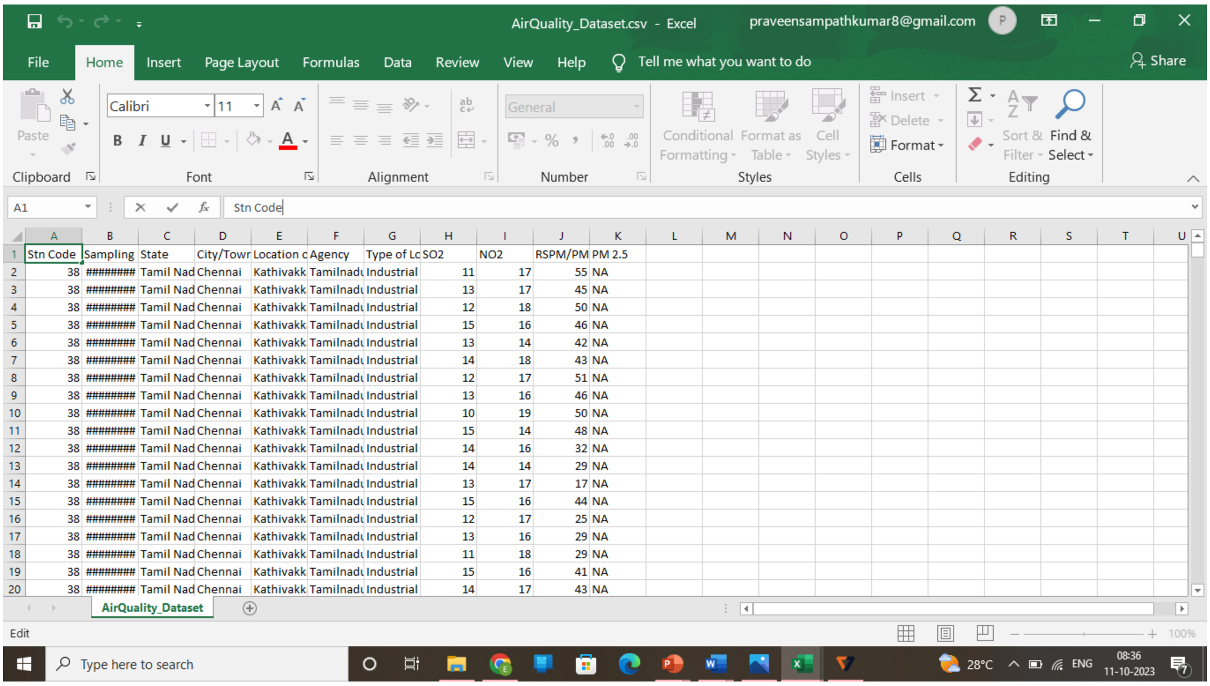
An index which is used to report air quality is called the air quality index (AQI). It measures the impact of air pollution on a person's health over a short period of time. The purpose of the AQI is to educate the public on the negative health effects of local air pollution. Air Pollution implies a great significant on environmental and health challenge and it demands on comprehensive and prediction efforts. This project,**" Air Quality Analysis and Prediction in Tamil Nadu,"** focuses on leveraging data science techniques to access and predict the air quality levels across various regions in Tamil Nadu, India

Data set description and Sample Data:

The link to the dataset for this chosen project is given below:

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

The above dataset contains the combined version of air quality of Tamil Nadu from 2014. This contains some district wise data for the prediction of air quality parameter in the state of Tamil Nadu. This data was released by the Ministry of Environment and Forests and Central Pollution Control Board of India under the National Data Sharing and Accessibility Policy (NDSAP).

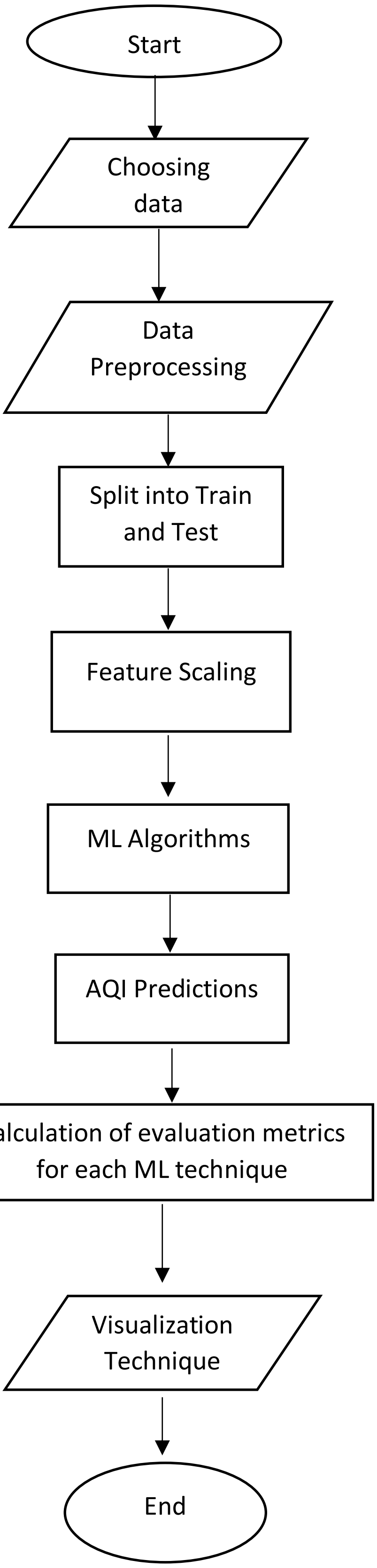


A1 | Stn Code

	A	B	C	D	E	F	G	H	I	J	K
1	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

METHODOLOGY:

Flow Chart for the proposed system



STEP1: Choosing a dataset

Choosing the proper dataset for implementing the project

STEP2: Data Pre-processing

In data pre-processing we have selected data for the analysis of air quality in the various district of Tamil Nadu. Each of the dataset was cleaned by remove null values of the chosen dataset. Microsoft Excel Software was used to remove unnecessary, irrelevant and erroneous data.

STEP3: Splitting of the dataset

The chosen datasets are split into training and test data. These are used to train the model and then test it against the original data. The values predicted by the machine learning algorithm and to predict accuracy of the data.

STEP4: Training the dataset

Empirical studies show the best results which are obtained if 80% of the data is used for training. Random sampling is used as a way to divide the data into train and test sections. It is widely accepted and is very popular.

STEP5: Testing the dataset

Empirical studies show the best results that are obtained if the remaining 20% of the data is used for testing. Random sampling is used as a way to divide the data into train and test sections. It is widely accepted and is very popular.

STEP6: Feature Scaling

The data should be normalized in order to make the dataset more flexible and more consistent. Standard Scalar from Scikit-Learn Library has been used to do so. It normalizes the features by deleting the mean and scaling the unit variance

STEP7: Applying various Machine learning techniques

After the normalization, we need to apply the various machine learning technique for analysing the data. Some of the machine learning technique random forest regression, support vector regression which are used to analysis the air quality index.

STEP8: Applying ML technique-random forest regression

Random forest is a supervised machine learning algorithm that is used for classification and regression problems. It creates decision trees from several samples, using the majority vote for classification and the average in the case of regression. A random forest produces precise predictions that are easy to understand. Effective handling of large datasets is possible.

STEP 9: Calculation of evaluation metric for each ML techniques

The metrics used for the proposed work are R-SQUARE, MSE, RMSE, MAE, and the accuracy of various algorithm.

STEP 10: Determine the efficient Visualization techniques

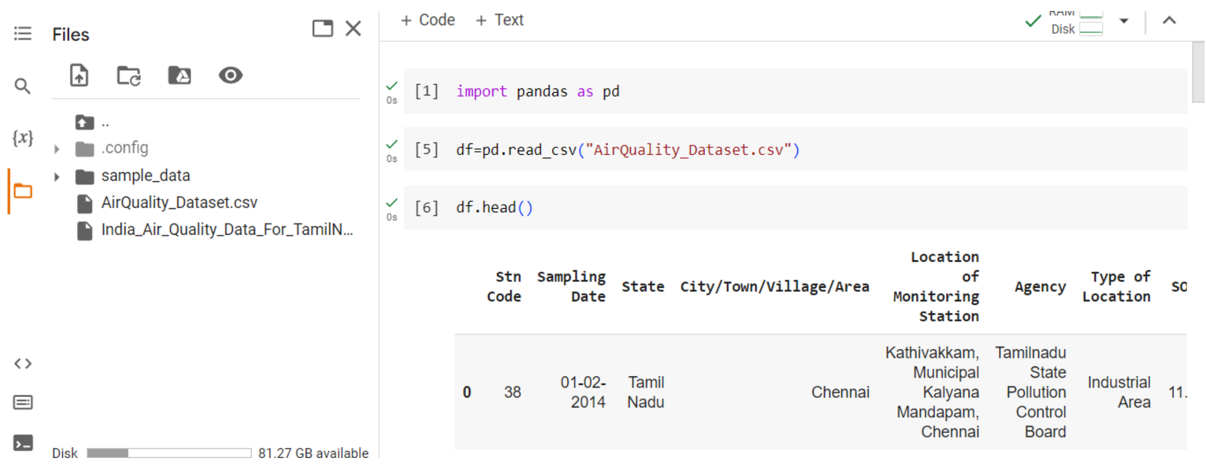
Visualizations play a crucial role in conveying insights from air quality data analysis. Here are some visualization methods and techniques that can be employed in the **"Air Quality Analysis and Prediction in Tamil Nadu"**

- **Time Series Plots**- Plot historical trends of SO2, NO2 and RSPM/PM10 levels over time. Use **line charts** to illustrate daily, monthly or seasonal variations
- **Heatmaps**-Create heatmaps to visualize pollutant concentrations across different monitoring stations and geographical areas.
- **Scatter Plots**-Use scatter plots to explore correlations between air quality parameters.

Explanation:

The given dataset contains the different columns with their specific details. The different columns are **"stn code, sampling date, state, city/town, Location of monitoring stations, Agency, Type of location, SO2, NO2, RSPM/PM, SPM"**

To further proceeding of the project let us drop the unwanted columns that is unnecessary for analysis of the air quality in Tamil Nadu

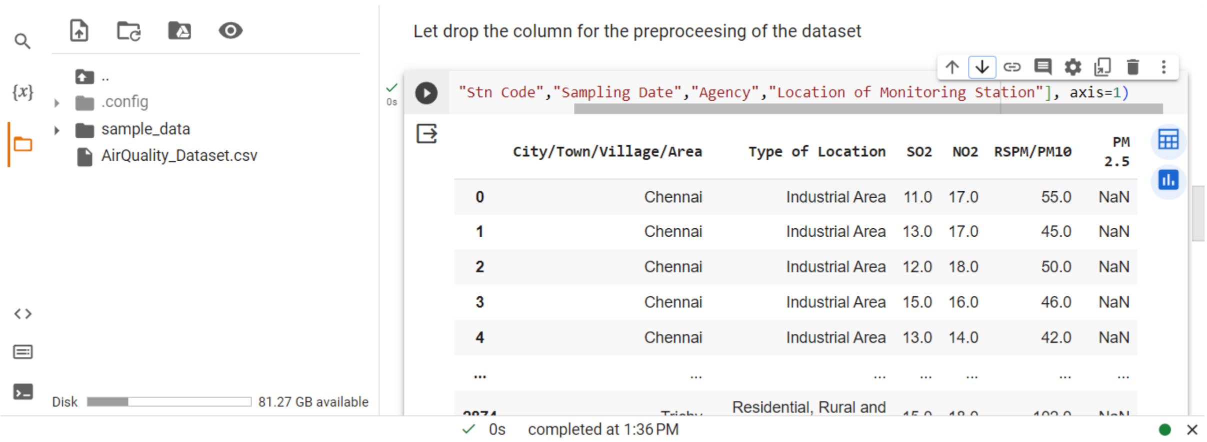


```
[1] import pandas as pd
```

```
[5] df=pd.read_csv("AirQuality_Dataset.csv")
```

```
[6] df.head()
```

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO
0	38	01-02-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.



Let drop the column for the preproceesing of the dataset

```
"Stn Code","Sampling Date","Agency","Location of Monitoring Station"], axis=1)
```

	City/Town/Village/Area	Type of Location	SO2	NO2	RSPM/PM10	PM 2.5
0	Chennai	Industrial Area	11.0	17.0	55.0	NaN
1	Chennai	Industrial Area	13.0	17.0	45.0	NaN
2	Chennai	Industrial Area	12.0	18.0	50.0	NaN
3	Chennai	Industrial Area	15.0	16.0	46.0	NaN
4	Chennai	Industrial Area	13.0	14.0	42.0	NaN
...	...	...	...	...	...	...
2074	Trichy	Residential, Rural and	15.0	18.0	102.0	NaN

✓ 0s completed at 1:36 PM

By this, we are going to use the specific columns for the preprocessing of the data to predict and analyze the air quality in various regions in Tamil Nadu.

1.Data Collection:

Monitoring Stations: Establish a network of air quality monitoring stations across Tamil Nadu. These stations should be strategically located in urban, industrial, and rural areas to capture a representative sample of air quality conditions.

- **Parameters:** Measure various air quality parameters, including particulate matter (PM 2.5 and PM 10), nitrogen dioxide(NO2), sulphur dioxide(SO2), carbon monoxide(CO), ozone(O3) and other volatile organic compounds.
- **Meteorological Data:** Collect meteorological data, such as temperature, humidity, wind speed,
  and wind direction, as these factors can influence air quality.
- **Historical Data:** Gather historical air quality data to establish trends and identify areas with chronic air quality problems.

2. Data Analysis:

Air Quality Index (AQI): Calculate the AQI for different locations in Tamil Nadu to provide a clear and understandable representation of air quality to the public.

- **Identify Hotspots:** Identify areas with consistently poor air quality, such as major cities or industrial zones, and pinpoint the key pollutants responsible.
- **Seasonal Trends:** Analyse seasonal variations in air quality, as well as the factors contributing to these variations, such as agricultural burning, weather conditions, or industrial activity.

3. Pollution Sources:

Industrial Emissions: Emissions from industrial facilities, such as factories and power plants, and assess compliance with emission standards. Examine

- **Vehicle Emissions:** Evaluate the impact of vehicular emissions on air quality, considering the prevalence of different types of vehicles and fuel types.
- **Agricultural Practices:** Investigate the role of agriculture in air quality, including the use of pesticides and burning of crop residues.
- **Waste Management:** Assess waste disposal practices and their impact on air quality, especially in urban areas.

4. Health Impact Assessment:

Collaborate with healthcare institutions to study the health effects of poor air quality on the population of Tamil Nadu. Identify vulnerable groups, such as children, the elderly, and individuals with pre-existing respiratory conditions, and assess their exposure and health outcomes.

5. Policy and Regulation:

Review existing air quality regulations and policies in Tamil Nadu to identify gaps or areas for improvement. Develop or update regulations to control emissions from various sources, and enforce strict compliance measures.

6. Public Awareness:

Launch public awareness campaigns to educate residents about the health risks associated with poor air quality and ways to protect themselves. Provide real-time air quality information through websites, apps, and public displays.

7. Mitigation Strategies:

Implement pollution control technologies in industries and encourage the use of cleaner fuels. Promote sustainable urban planning, public transportation, and green spaces to reduce vehicle emissions and enhance air quality. Encourage agricultural practices that minimize burning and promote sustainable waste management.

8. International Cooperation:

Collaborate with neighbouring states and countries to address transboundary air pollution issues, especially during cross-border events like crop burning. This air quality analysis is the first part of a comprehensive strategy to improve air quality in Tamil Nadu. It is essential to monitor progress over time and adjust strategies as needed to ensure cleaner air for the people and the environment.

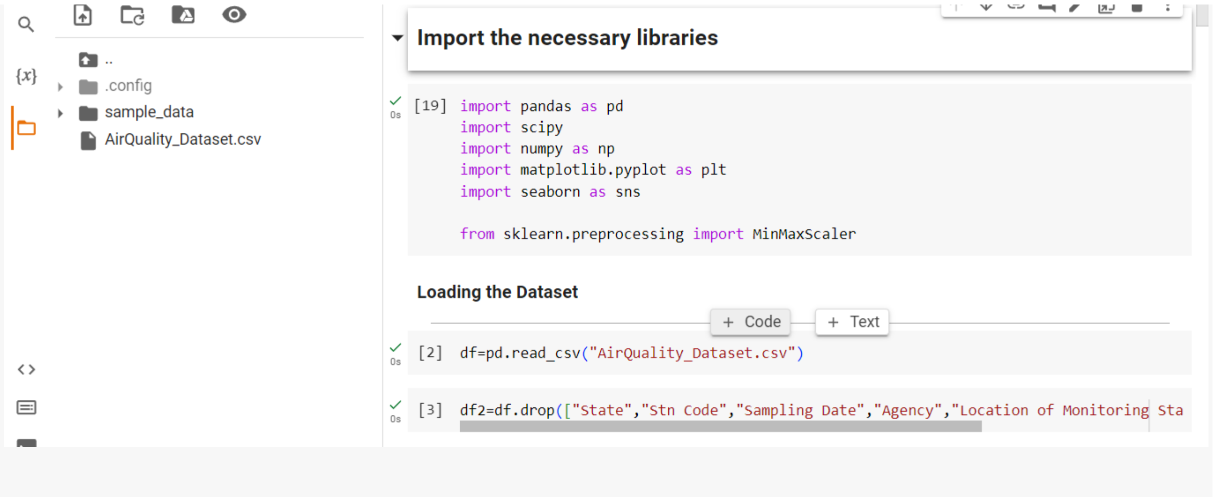
PROGRAM:

Import the necessary libraries:

```
import pandas as pd
import scipy
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
```

Loading the Dataset:

```
df=pd.read_csv("AirQuality_Dataset.csv")

df2=df.drop(["State","Stn Code","Sampling Date","Agency","Location of Monitoring Station","PM 2.5"], axis=1)
```



Import the necessary libraries

```
[19] import pandas as pd
     import scipy
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns

     from sklearn.preprocessing import MinMaxScaler
```

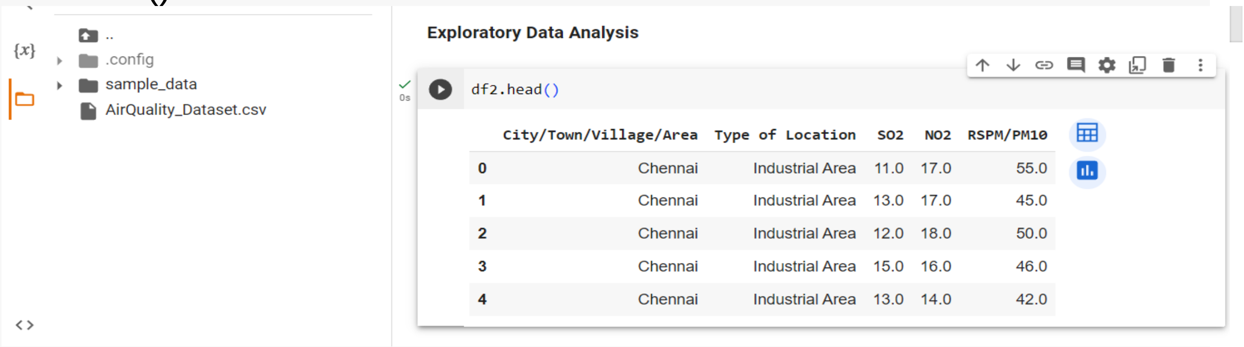
Loading the Dataset

```
[2]  df=pd.read_csv("AirQuality_Dataset.csv")
```

```
[3]  df2=df.drop(["State","Stn Code","Sampling Date","Agency","Location of Monitoring Sta
```

Exploratory Data Analysis:

df2.head()

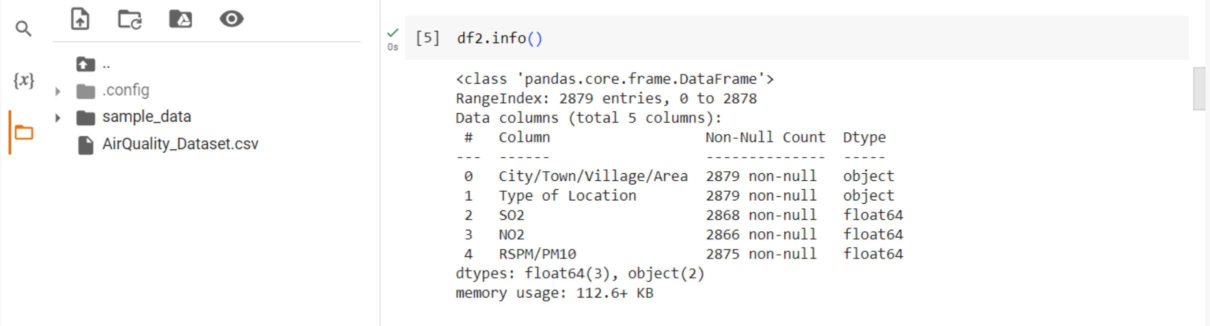


Exploratory Data Analysis

```
df2.head()
```

	City/Town/Village/Area	Type of Location	SO2	NO2	RSPM/PM10
0	Chennai	Industrial Area	11.0	17.0	55.0
1	Chennai	Industrial Area	13.0	17.0	45.0
2	Chennai	Industrial Area	12.0	18.0	50.0
3	Chennai	Industrial Area	15.0	16.0	46.0
4	Chennai	Industrial Area	13.0	14.0	42.0

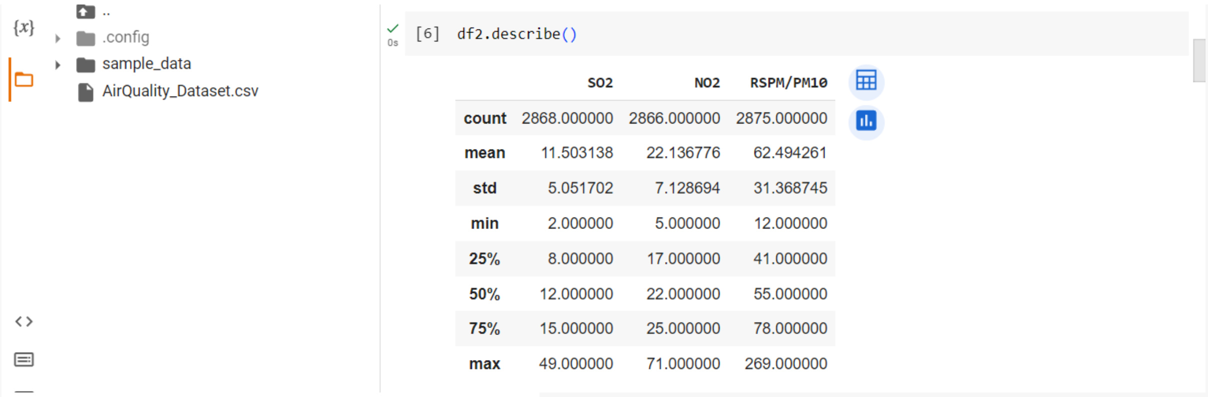
df2.info()



```
[5]  df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2879 entries, 0 to 2878
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   City/Town/Village/Area  2879 non-null   object
 1   Type of Location        2879 non-null   object
 2   SO2                     2868 non-null   float64
 3   NO2                     2866 non-null   float64
 4   RSPM/PM10               2875 non-null   float64
dtypes: float64(3), object(2)
memory usage: 112.6+ KB
```

df2.describe()

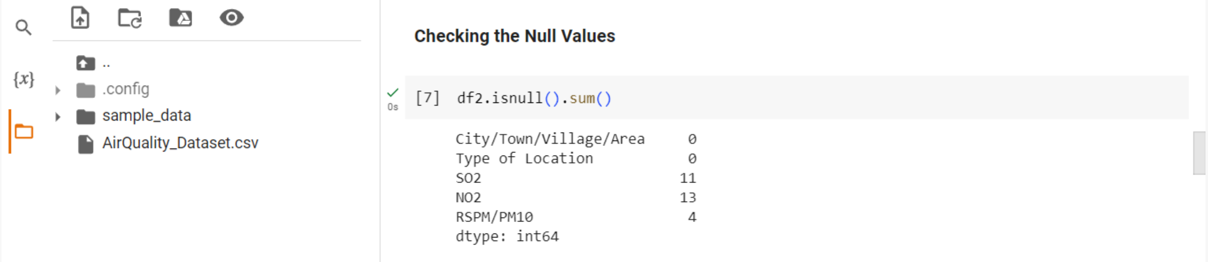


```
[6] df2.describe()
```

	SO2	NO2	RSPM/PM10
count	2868.000000	2866.000000	2875.000000
mean	11.503138	22.136776	62.494261
std	5.051702	7.128694	31.368745
min	2.000000	5.000000	12.000000
25%	8.000000	17.000000	41.000000
50%	12.000000	22.000000	55.000000
75%	15.000000	25.000000	78.000000
max	49.000000	71.000000	269.000000

Checking the Null Values:

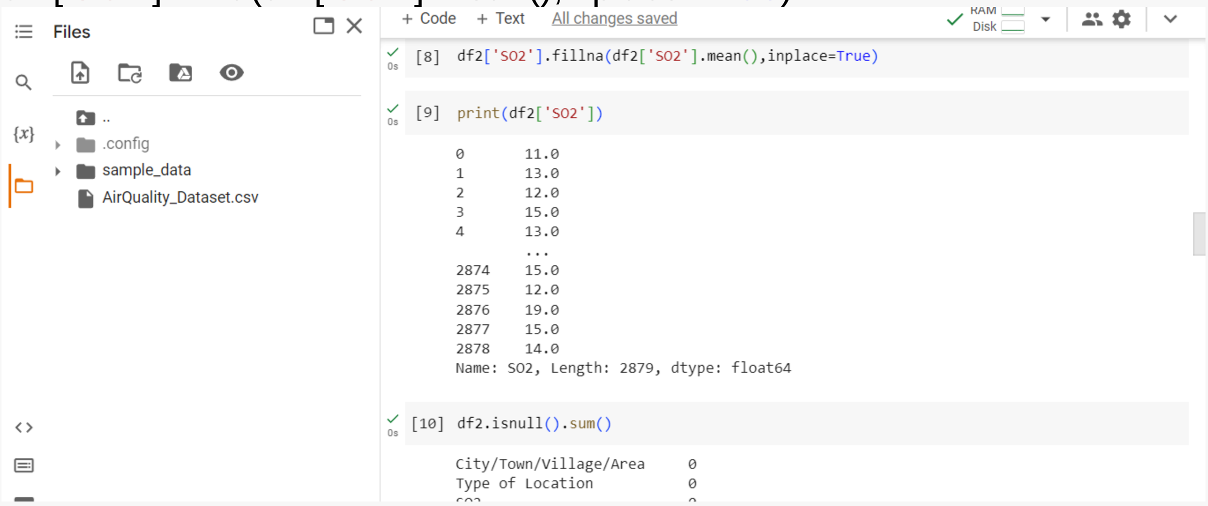
```
df2.isnull().sum()
```



Checking the Null Values

```
[7] df2.isnull().sum()

City/Town/Village/Area     0
Type of Location           0
SO2                       11
NO2                       13
RSPM/PM10                  4
dtype: int64
```

```
df2['SO2'].fillna(df2['SO2'].mean(),inplace=True)
```

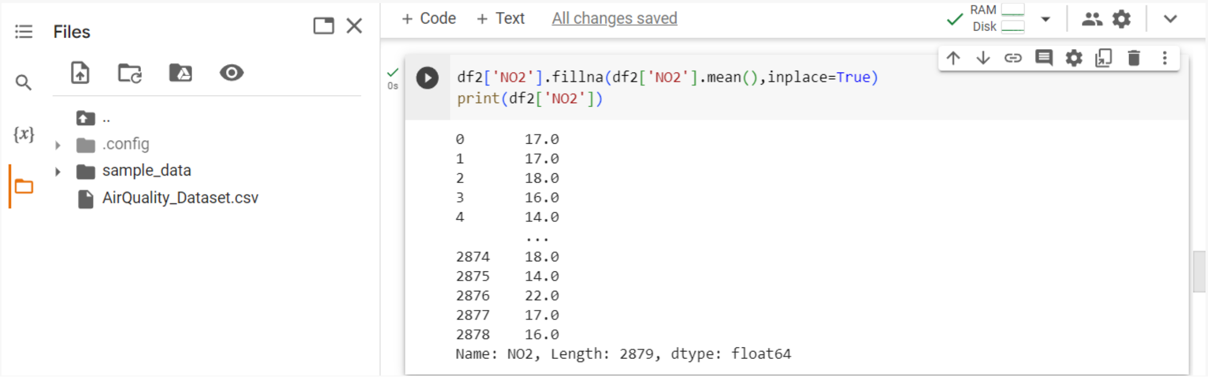


```
[8] df2['SO2'].fillna(df2['SO2'].mean(),inplace=True)
```

```
[9] print(df2['SO2'])

0       11.0
1       13.0
2       12.0
3       15.0
4       13.0
        ...
2874    15.0
2875    12.0
2876    19.0
2877    15.0
2878    14.0
Name: SO2, Length: 2879, dtype: float64
```

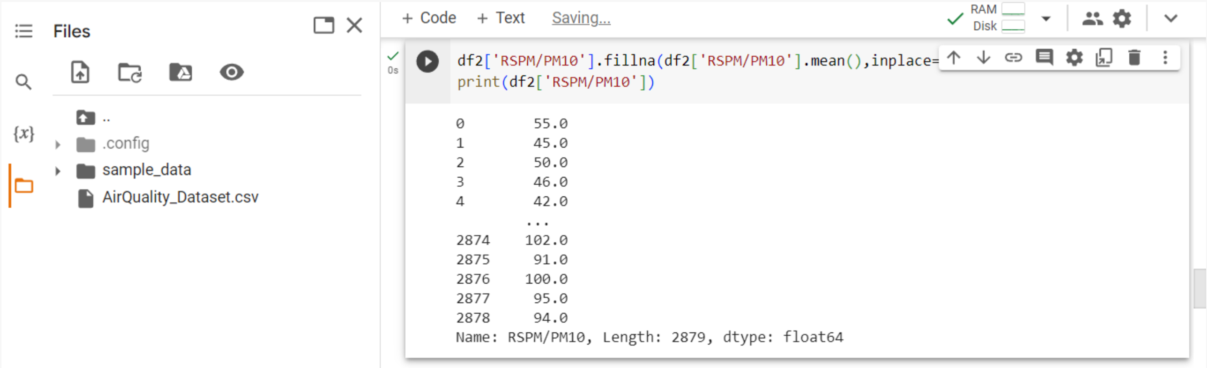
```
[10] df2.isnull().sum()

City/Town/Village/Area     0
Type of Location           0
```

```
df2['NO2'].fillna(df2['NO2'].mean(),inplace=True)
```



```
df2['NO2'].fillna(df2['NO2'].mean(),inplace=True)
print(df2['NO2'])

0       17.0
1       17.0
2       18.0
3       16.0
4       14.0
        ...
2874    18.0
2875    14.0
2876    22.0
2877    17.0
2878    16.0
Name: NO2, Length: 2879, dtype: float64
```

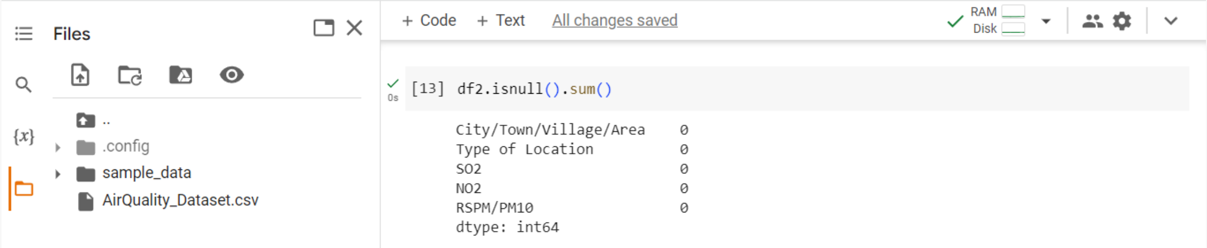
```
df2['RSPM/PM10'].fillna(df2['RSPM/PM10'].mean(),inplace=True)
```



```
df2['RSPM/PM10'].fillna(df2['RSPM/PM10'].mean(),inplace=
print(df2['RSPM/PM10'])
```

```
0        55.0
1        45.0
2        50.0
3        46.0
4        42.0
         ...
2874    102.0
2875     91.0
2876    100.0
2877     95.0
2878     94.0
Name: RSPM/PM10, Length: 2879, dtype: float64
```

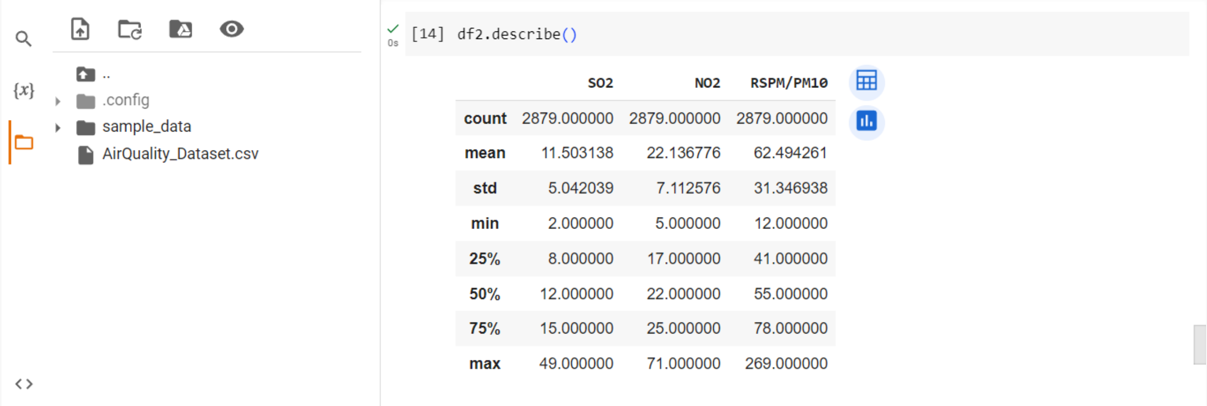
```
df2.isnull().sum()
```



```
[13] df2.isnull().sum()
```

```
City/Town/Village/Area    0
Type of Location          0
SO2                       0
NO2                       0
RSPM/PM10                 0
dtype: int64
```

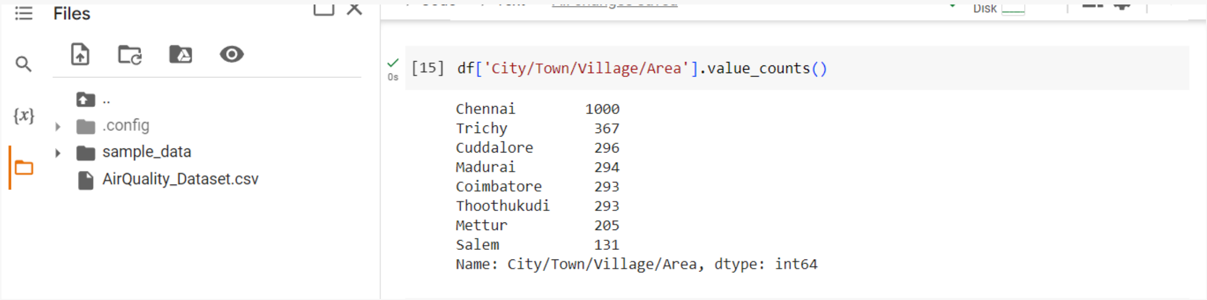
```
df2.describe()
```



```
[14] df2.describe()
```

	SO2	NO2	RSPM/PM10
count	2879.000000	2879.000000	2879.000000
mean	11.503138	22.136776	62.494261
std	5.042039	7.112576	31.346938
min	2.000000	5.000000	12.000000
25%	8.000000	17.000000	41.000000
50%	12.000000	22.000000	55.000000
75%	15.000000	25.000000	78.000000
max	49.000000	71.000000	269.000000

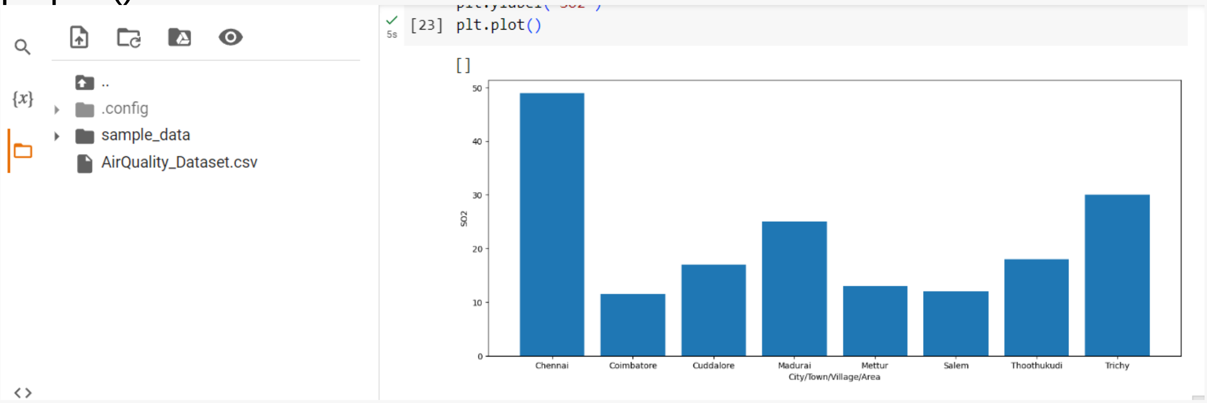
```
df['City/Town/Village/Area'].value_counts()
```



```
[15] df['City/Town/Village/Area'].value_counts()
```

```
Chennai        1000
Trichy          367
Cuddalore       296
Madurai         294
Coimbatore      293
Thoothukudi     293
Mettur          205
Salem           131
Name: City/Town/Village/Area, dtype: int64
```

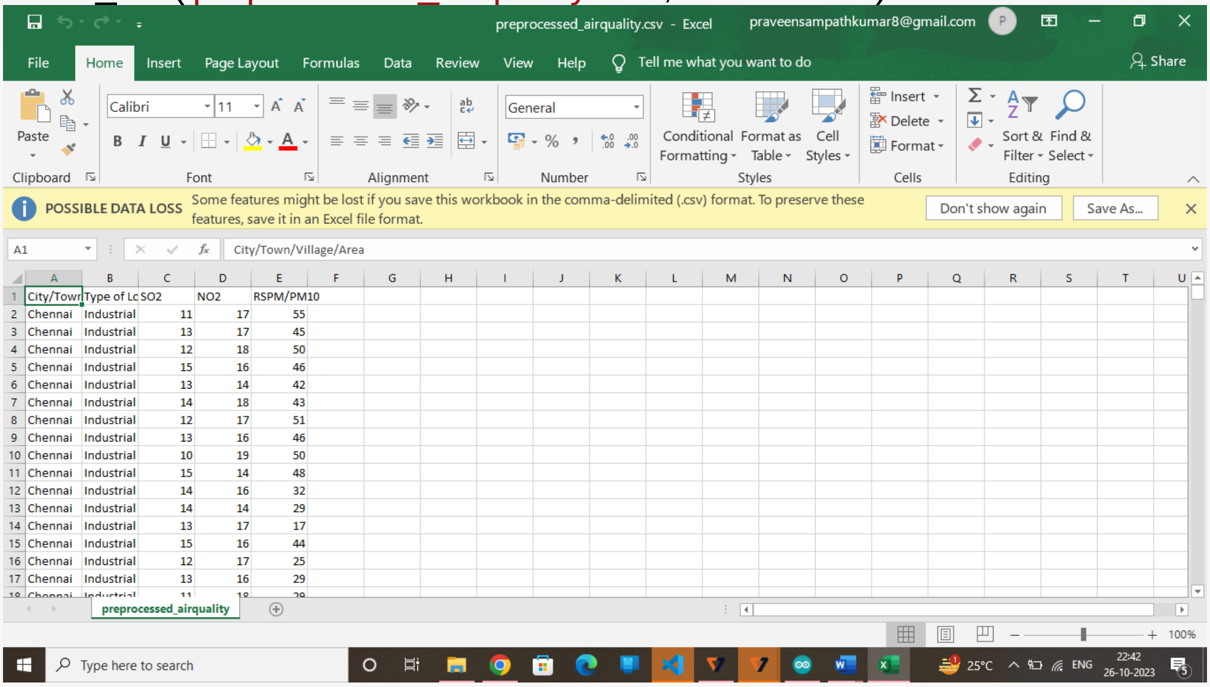
```
plt.figure(figsize=(15,6))
plt.bar(df2['City/Town/Village/Area'],df2['SO2'])
plt.xlabel('City/Town/Village/Area')
plt.ylabel('SO2')
plt.plot()
```



```
[23] plt.plot()

[]
```

Saving the Pre-processed Data:

```
df2.to_csv('preprocessed_airquality.csv',index=False)
```



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

City/Town/Village/Area

	A	B	C	D	E
1	City/Towr	Type of Lc	SO2	NO2	RSPM/PM10
2	Chennai	Industrial	11	17	55
3	Chennai	Industrial	13	17	45
4	Chennai	Industrial	12	18	50
5	Chennai	Industrial	15	16	46
6	Chennai	Industrial	13	14	42
7	Chennai	Industrial	14	18	43
8	Chennai	Industrial	12	17	51
9	Chennai	Industrial	13	16	46
10	Chennai	Industrial	10	19	50
11	Chennai	Industrial	15	14	48
12	Chennai	Industrial	14	16	32
13	Chennai	Industrial	14	14	29
14	Chennai	Industrial	13	17	17
15	Chennai	Industrial	15	16	44
16	Chennai	Industrial	12	17	25
17	Chennai	Industrial	13	16	29

CODE:

Loading the pre-processed dataset:

```
import pandas as pd
df=pd.read_csv('preprocessed_airquality.csv')
```

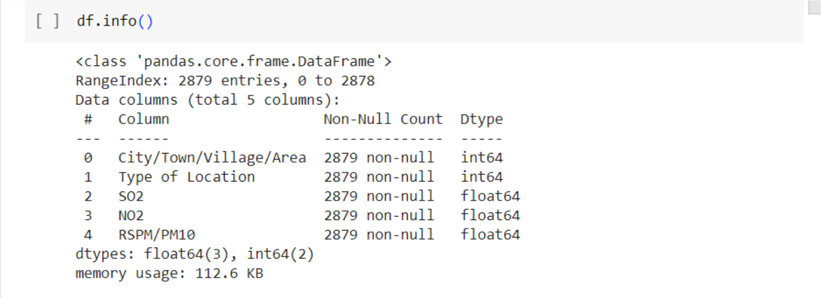
One Hot Encoding:

For categorical variables where no such ordinal relationship exists, the integer encoding is not enough.

In fact, using the one hot encoding technique and allowing the model to assume a natural ordering between categories may result in poor performance or unexpected results. This can be applied to the integer representation.

```
pd.get_dummies(df,columns=['City/Town/Village/Area','Type of Location'])

dist=(df['City/Town/Village/Area'])
distset=set(dist)
dd=list(distset)
dict0fwords={dd[i]:i for i in range(0,len(dd))}
df['City/Town/Village/Area']=df['City/Town/Village/Area'].map(dict0fwords)
```

```
dist=(df['Type of Location'])
distset=set(dist)
dd=list(distset)
dict0fwords={dd[i]:i for i in range(0,len(dd))}
df['Type of Location']=df['Type of Location'].map(dict0fwords)
```



```
[ ]  df.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 2879 entries, 0 to 2878
     Data columns (total 5 columns):
      #   Column                 Non-Null Count  Dtype
     ---  ------                 --------------  -----
      0   City/Town/Village/Area  2879 non-null   int64
      1   Type of Location        2879 non-null   int64
      2   SO2                     2879 non-null   float64
      3   NO2                     2879 non-null   float64
      4   RSPM/PM10               2879 non-null   float64
     dtypes: float64(3), int64(2)
     memory usage: 112.6 KB
```

Model Training:

- Choose a machine learning algorithm
- There are number of different machine learning algorithms that can be used for air quality analysis such as linear regression, KNN, Lasso Regression and Random Forests.

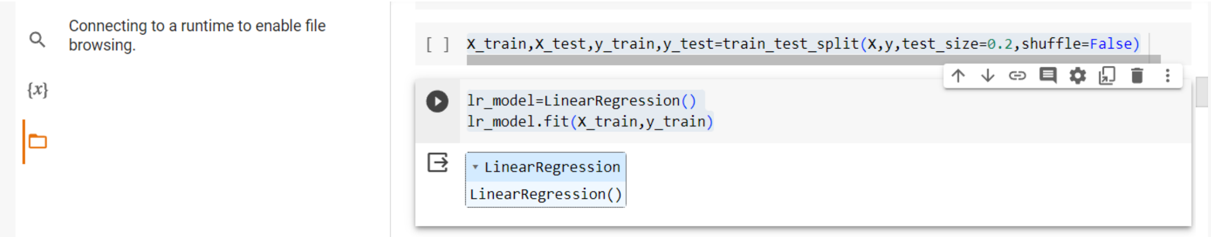
Model Evaluation:

- Model Evaluation is the process of assessing the performance of a machine learning model on unseen data. This is important to ensure that the model will generalize well to the new data.
- There are number of different metrics that can be used to evaluate the performance of air quality analysis and prediction model
- Some of the most common metrics include:

  - **Mean Squared Error (MSE):** This metric measures the average squared difference between the different areas/cities in Tamil Nadu and the various pollutants such as SO2, NO2 and RSPM/PM10.
  - **Root Mean Squared Error (RMSE):** This metric is the square root of the MSE
  - **Mean Absolute Error (MAE):** This metric measures the average absolute difference between the different areas/cities in Tamil Nadu and the various pollutants such as SO2, NO2 and RSPM/PM10.
  - **R-Squared:** This metric measures how well the model explains the variation in the pollutants in the different areas.

Linear Regression:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

X=df[['City/Town/Village/Area']].values
y=df[['SO2','NO2','RSPM/PM10']].values

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,
shuffle=False)

lr_model=LinearRegression()
lr_model.fit(X_train,y_train)
```

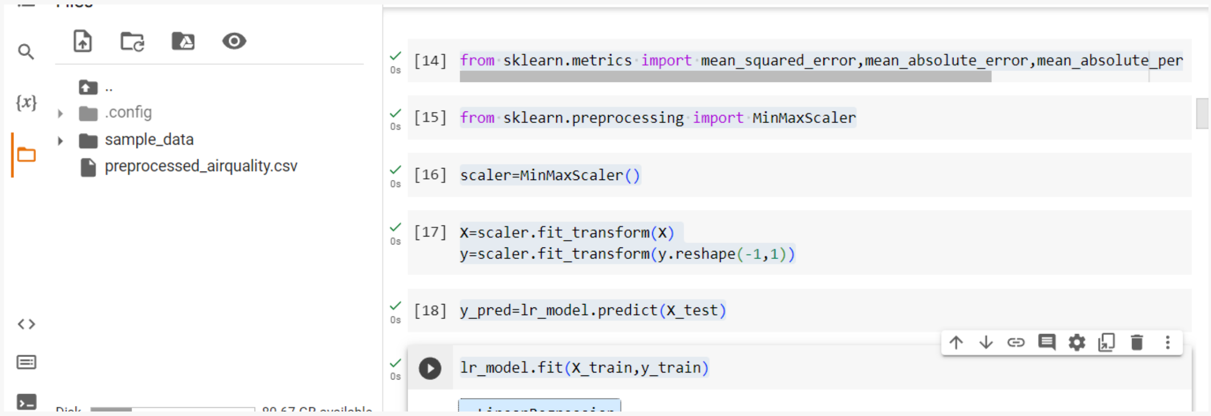


```
[ ] X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,shuffle=False)
```

```
lr_model=LinearRegression()
lr_model.fit(X_train,y_train)
```

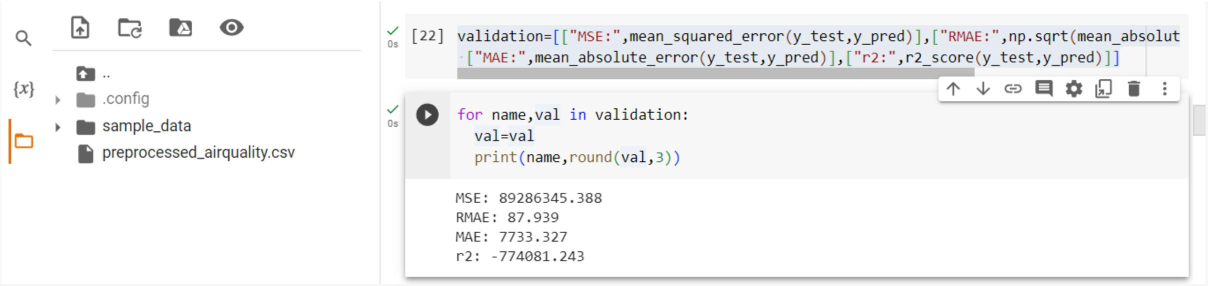
```
▾ LinearRegression
LinearRegression()
```

```
from sklearn.metrics import
mean_squared_error,mean_absolute_error,mean_absolute_percentage_error,r2_score

from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()

X=scaler.fit_transform(X)
y=scaler.fit_transform(y.reshape(-1,1))

y_pred=lr_model.predict(X_test)
lr_model.fit(X_train,y_train)
```



```
[14] from sklearn.metrics import mean_squared_error,mean_absolute_error,mean_absolute_per
[15] from sklearn.preprocessing import MinMaxScaler
[16] scaler=MinMaxScaler()
[17] X=scaler.fit_transform(X)
     y=scaler.fit_transform(y.reshape(-1,1))
[18] y_pred=lr_model.predict(X_test)
     lr_model.fit(X_train,y_train)
```

```
y_pred=scaler.inverse_transform(y_pred)
import numpy as np
validation=[["MSE:",mean_squared_error(y_test,y_pred)],["RMAE:",np.sqrt(mean_absolute_error(y_test,y_pred))],
 ["MAE:",mean_absolute_error(y_test,y_pred)],["r2:",r2_score(y_test,y_pred)]]
for name,val in validation:
  val=val
  print(name,round(val,3))
```



```
[22] validation=[["MSE:",mean_squared_error(y_test,y_pred)],["RMAE:",np.sqrt(mean_absolut
     ["MAE:",mean_absolute_error(y_test,y_pred)],["r2:",r2_score(y_test,y_pred)]]
```

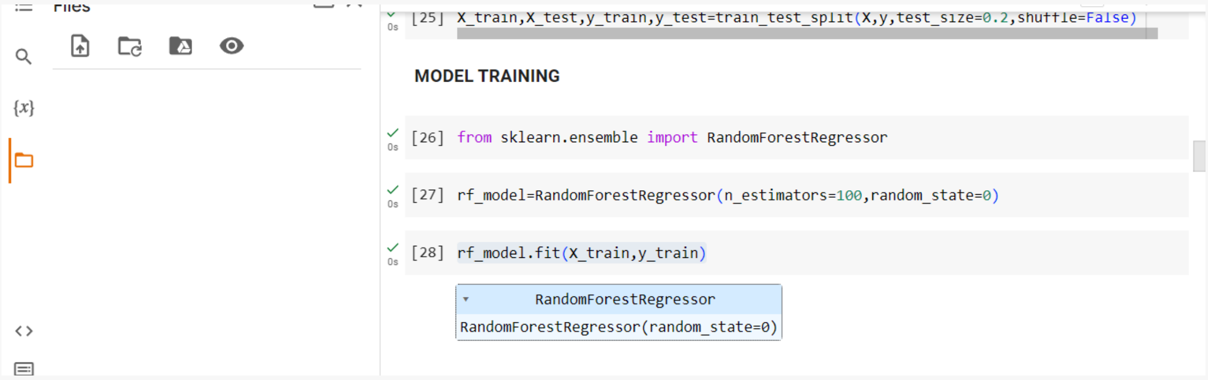
```
for name,val in validation:
  val=val
  print(name,round(val,3))

MSE: 89286345.388
RMAE: 87.939
MAE: 7733.327
r2: -774081.243
```

RANDOM FORESTS:

```
X=df[['City/Town/Village/Area']].values
y=df[['SO2','NO2','RSPM/PM10']].values
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,
shuffle=False)
```

Model Training

```
from sklearn.ensemble import RandomForestRegressor
rf_model=RandomForestRegressor(n_estimators=100,random
_state=0)
rf_model.fit(X_train,y_train)
```



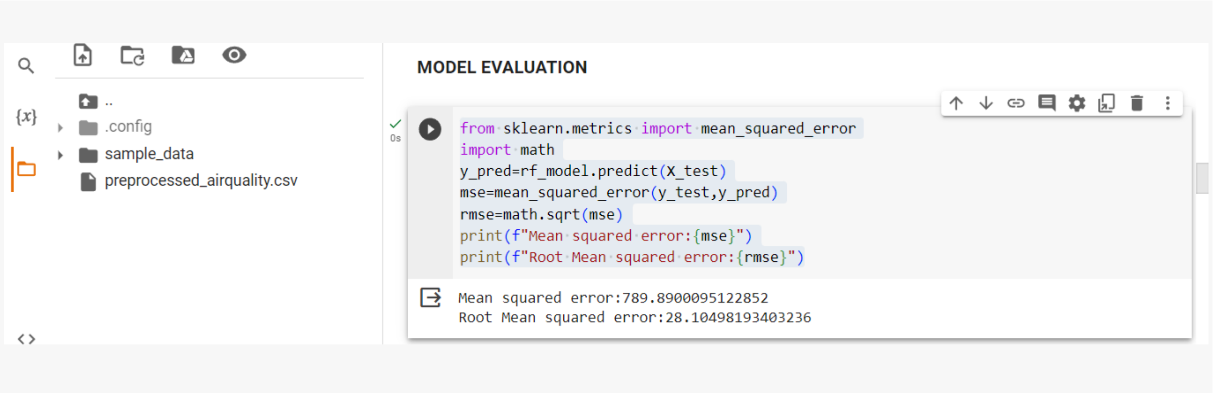
```
[25] X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,shuffle=False)
```

MODEL TRAINING

```
[26] from sklearn.ensemble import RandomForestRegressor
[27] rf_model=RandomForestRegressor(n_estimators=100,random_state=0)
[28] rf_model.fit(X_train,y_train)

     RandomForestRegressor
RandomForestRegressor(random_state=0)
```

Model Evaluation

```
from sklearn.metrics import mean_squared_error
import math
y_pred=rf_model.predict(X_test)
mse=mean_squared_error(y_test,y_pred)
rmse=math.sqrt(mse)
print(f"Mean squared error:{mse}")
print(f"Root Mean squared error:{rmse}")
```

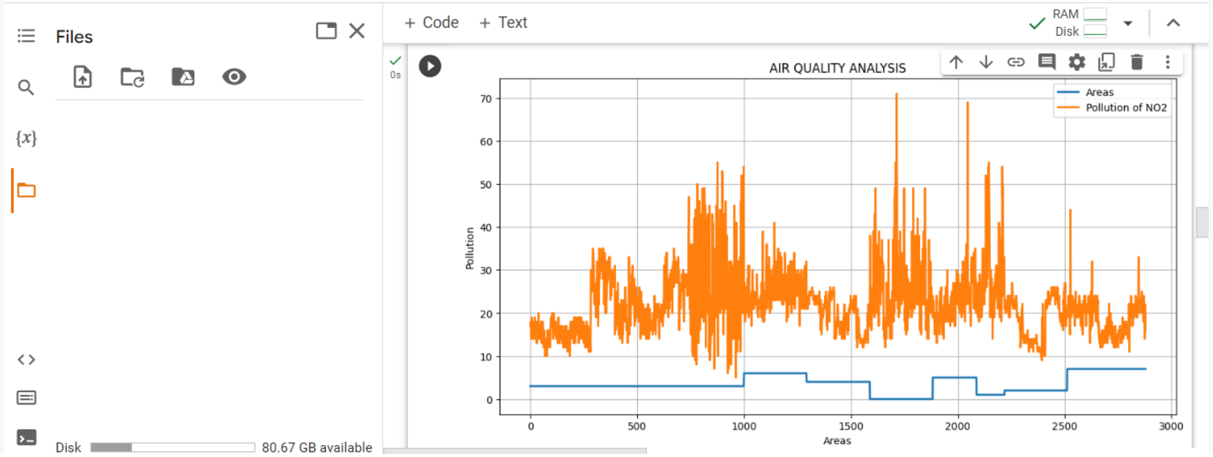


```
from sklearn.metrics import mean_squared_error
import math
y_pred=rf_model.predict(X_test)
mse=mean_squared_error(y_test,y_pred)
rmse=math.sqrt(mse)
print(f"Mean squared error:{mse}")
print(f"Root Mean squared error:{rmse}")
```

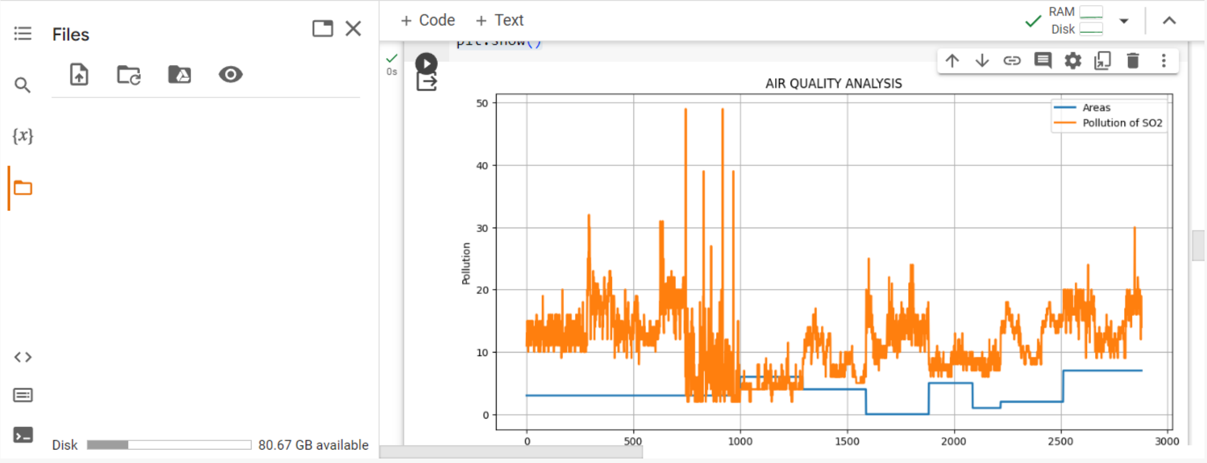
```
Mean squared error:789.8900095122852
Root Mean squared error:28.10498193403236
```

Visualization of Random Forest

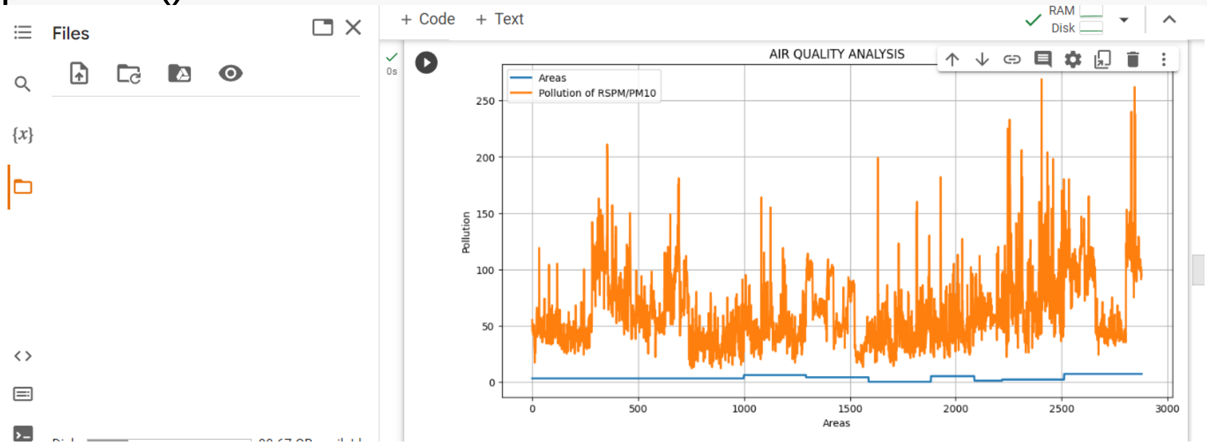
```
import matplotlib.pyplot as plt
data_range=df.index[-len(y_test):]
plt.figure(figsize=(12,6))
plt.plot(df['City/Town/Village/Area'],label='Areas',linewidth=2)
plt.plot(df['NO2'],label='Pollution of NO2',linewidth=2)
plt.title("AIR QUALITY ANALYSIS")
plt.xlabel('Areas')
plt.ylabel('Pollution')
plt.legend()
plt.grid()
plt.show()
```



```
import matplotlib.pyplot as plt
data_range=df.index[-len(y_test):]
plt.figure(figsize=(12,6))
plt.plot(df['City/Town/Village/Area'],label='Areas',linewidth=2)
plt.plot(df['SO2'],label='Pollution of SO2',linewidth=2)
plt.title("AIR QUALITY ANALYSIS")
plt.xlabel('Areas')
plt.ylabel('Pollution')
plt.legend()
plt.grid()
plt.show()
```

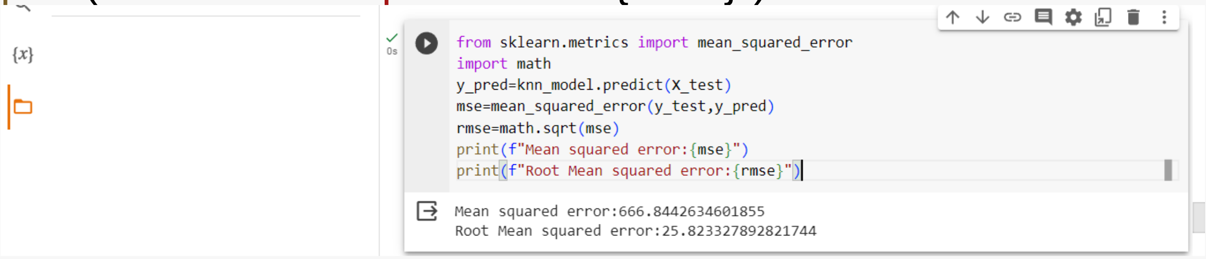


```
import matplotlib.pyplot as plt
data_range=df.index[-len(y_test):]
plt.figure(figsize=(12,6))
plt.plot(df['City/Town/Village/Area'],label='Areas',linewidth=2)
plt.plot(df['RSPM/PM10'],label='Pollution of 
RSPM/PM10',linewidth=2)
plt.title("AIR QUALITY ANALYSIS")
plt.xlabel('Areas')
plt.ylabel('Pollution')
plt.legend()
plt.grid()
plt.show()
```



KNN:

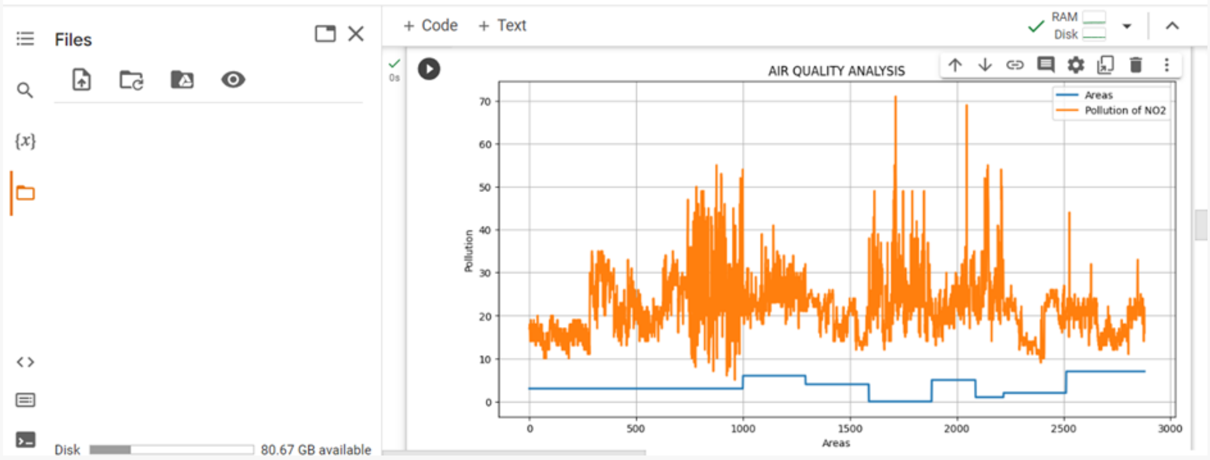
```
X=df[['City/Town/Village/Area']].values
y=df[['SO2','NO2','RSPM/PM10']].values
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,
shuffle=False)
from sklearn.neighbors import KNeighborsRegressor
knn_model=KNeighborsRegressor(n_neighbors=5)
knn_model.fit(X_train,y_train)




from sklearn.metrics import mean_squared_error
import math
y_pred=knn_model.predict(X_test)
mse=mean_squared_error(y_test,y_pred)
rmse=math.sqrt(mse)
print(f"Mean squared error:{mse}")
print(f"Root Mean squared error:{rmse}")
```



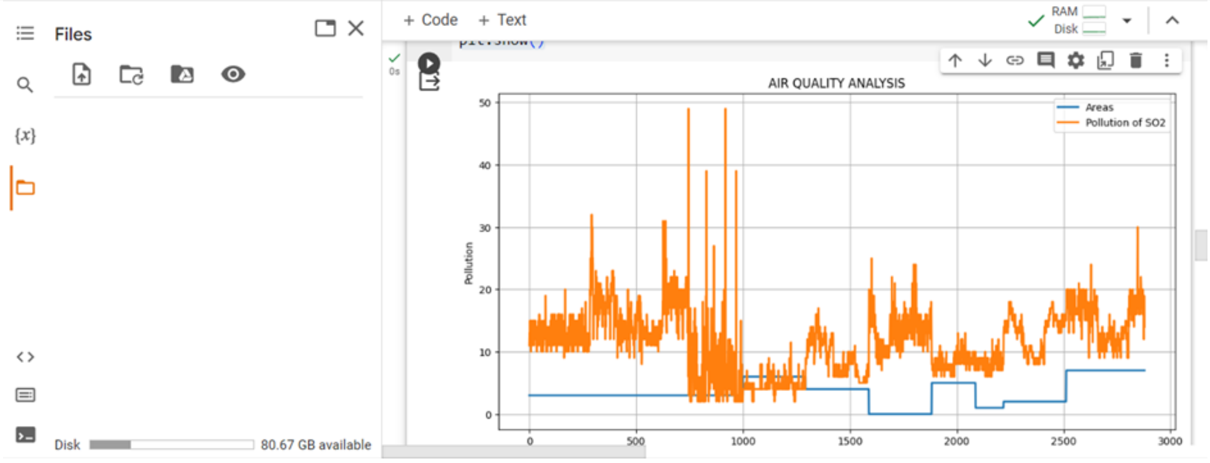
```
from sklearn.metrics import mean_squared_error
import math
y_pred=knn_model.predict(X_test)
mse=mean_squared_error(y_test,y_pred)
rmse=math.sqrt(mse)
print(f"Mean squared error:{mse}")
print(f"Root Mean squared error:{rmse}")
```

```
Mean squared error:666.8442634601855
Root Mean squared error:25.823327892821744
```

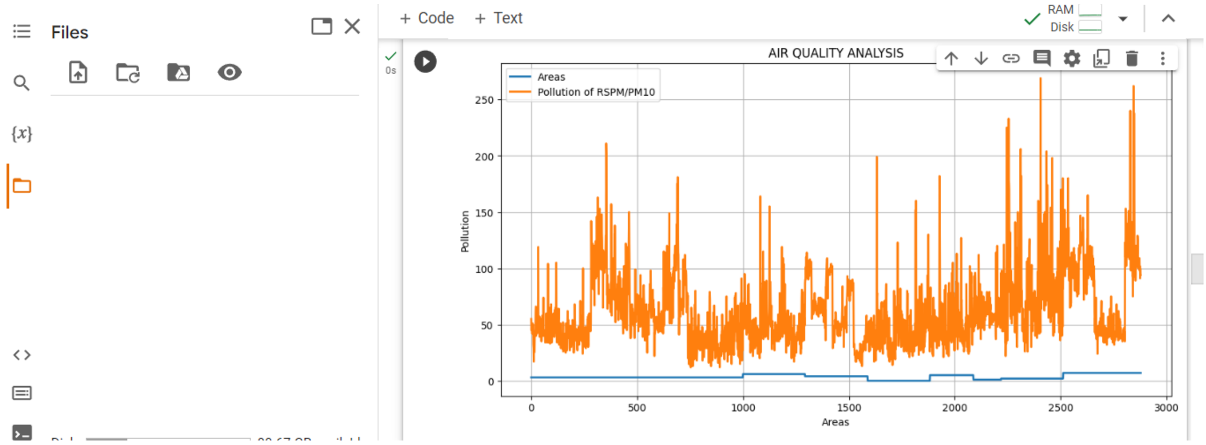
```
import matplotlib.pyplot as plt
data_range=df.index[-len(y_test):]
plt.figure(figsize=(12,6))
plt.plot(df['City/Town/Village/Area'],label='Areas',linewidth=2)
plt.plot(df['NO2'],label='Pollution of NO2',linewidth=2)
plt.title("AIR QUALITY ANALYSIS")
plt.xlabel('Areas')
plt.ylabel('Pollution')
plt.legend()
plt.grid()
plt.show()
```



```
import matplotlib.pyplot as plt
data_range=df.index[-len(y_test):]
plt.figure(figsize=(12,6))
plt.plot(df['City/Town/Village/Area'],label='Areas',linewidth=2)
plt.plot(df['SO2'],label='Pollution of SO2',linewidth=2)
plt.title("AIR QUALITY ANALYSIS")
plt.xlabel('Areas')
plt.ylabel('Pollution')
plt.legend()
plt.grid()
plt.show()
```

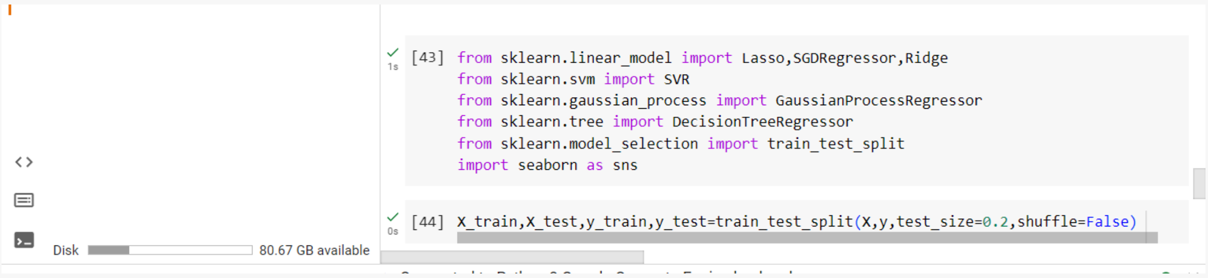


```
import matplotlib.pyplot as plt
data_range=df.index[-len(y_test):]
plt.figure(figsize=(12,6))
plt.plot(df['City/Town/Village/Area'],label='Areas',linewidth=2)
plt.plot(df['RSPM/PM10'],label='Pollution of RSPM/PM10',linewidth=2)
plt.title("AIR QUALITY ANALYSIS")
plt.xlabel('Areas')
plt.ylabel('Pollution')
plt.legend()
plt.grid()
plt.show()
```



Import Libraries:

```
from sklearn.linear_model import Lasso,SGDRegressor,Ridge
from sklearn.svm import SVR
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
import seaborn as sns
```



```
[43] from sklearn.linear_model import Lasso,SGDRegressor,Ridge
     from sklearn.svm import SVR
     from sklearn.gaussian_process import GaussianProcessRegressor
     from sklearn.tree import DecisionTreeRegressor
     from sklearn.model_selection import train_test_split
     import seaborn as sns

[44] X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,shuffle=False)
```

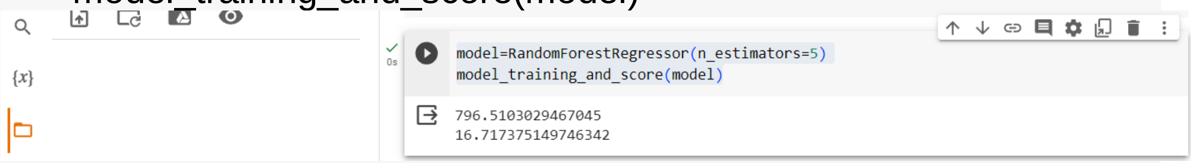
Model Training:

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,
shuffle=False)
ms=[]
ma=[]
mse=mean_squared_error
mae=mean_absolute_error
def model_training_and_score(model):
  model.fit(X_train,y_train)
  y_pred=np.nan_to_num(model.predict(X_test))
  print(mse(y_test,y_pred))
  print(mae(y_test,y_pred))
  ms.append(mse(y_test,y_pred))
  ma.append(mae(y_test,y_pred))
```

Model Evaluation:

1. Random Forest

```
model=RandomForestRegressor(n_estimators=5)
model_training_and_score(model)
```

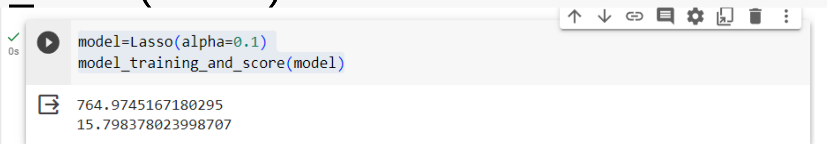


```
model=RandomForestRegressor(n_estimators=5)
model_training_and_score(model)
```

```
796.5103029467045
16.717375149746342
```

2.Lasso Regrssion

```
model=Lasso(alpha=0.1)
model_training_and_score(model)
```

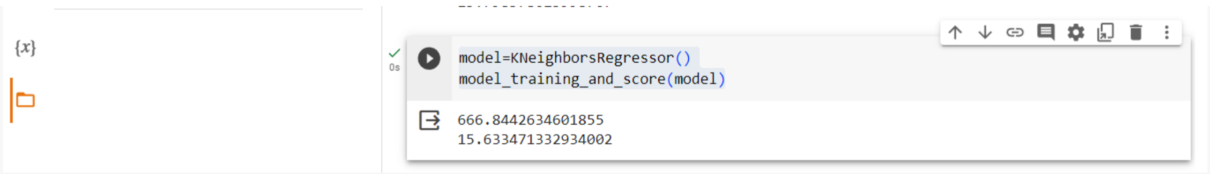


```
model=Lasso(alpha=0.1)
model_training_and_score(model)
```

```
764.9745167180295
15.798378023998707
```

3.K Neighbors Regression

```
model=KNeighborsRegressor()
model_training_and_score(model)
```

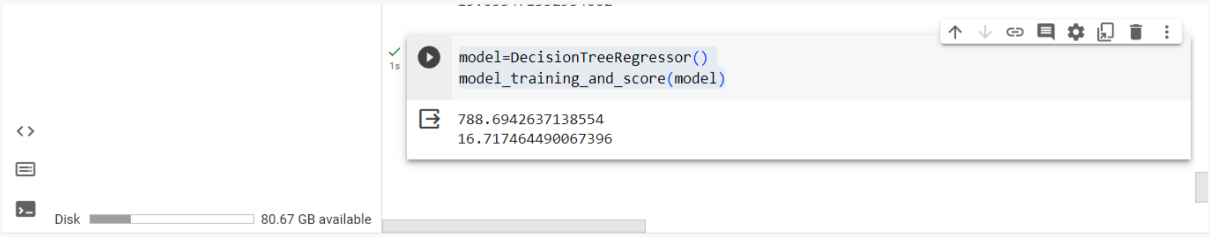


```
model=KNeighborsRegressor()
model_training_and_score(model)
```

```
666.8442634601855
15.633471332934002
```

4. Decision Tree Regression

```
model=DecisionTreeRegressor()
model_training_and_score(model)
```



```
model=DecisionTreeRegressor()
model_training_and_score(model)
```

```
788.6942637138554
16.717464490067396
```

CONCLUSION:

In conclusion, this project focuses on analyzing and predicting air quality in Tamil Nadu has yielded valuable insights and outcomes. Through the collection and analysis of historical air quality data, we are able to identify trends, seasonal variations, and the impact of various factors or air quality. Our predictive models, based on machine learning algorithms, demonstrated reasonable accuracy in forecasting air quality levels.