Phase 2 :- Innovation

Problem Statement :-

Air Quality Analysis and Prediction in Tamilnadu

The question involves analyzing and predicting air quality in the state of Tamil Nadu. Specifically, it focuses on understanding the relationship between air quality parameters, such as sulfur dioxide (SO2) and nitrogen dioxide (NO2), and particulate matter (RSPM/PM10), which can impact air quality and human health.

Data set Details:-

You obtained the dataset from Kaggle. The dataset can be found at this link: Air Quality Data Set. It contains information about air quality measurements in Tamil Nadu in 2014.

Data set Info and Description:-

- **SO2 (Sulfur Dioxide):** SO2 is a gaseous air pollutant produced by the burning of fossil fuels, particularly in industrial processes. It is a key contributor to air pollution and can have adverse health effects.
- **NO2 (Nitrogen Dioxide):** NO2 is another gaseous air pollutant, often associated with vehicle emissions and industrial processes. Like SO2, it can also impact air quality and health.
- **RSPM/PM10 (Particulate Matter):** Particulate matter refers to tiny solid particles or liquid droplets in the air. PM10 specifically refers to particles with a diameter of 10 micrometers or smaller. These particles can originate from various sources and affect air quality and respiratory health.

Libraries used :-

You used several libraries in your Jupyter notebook, including **pandas** for data manipulation, **numpy** for numerical operations,

matplotlib.pyplot for data visualization, and **sklearn** (scikit-learn) for machine learning tools.

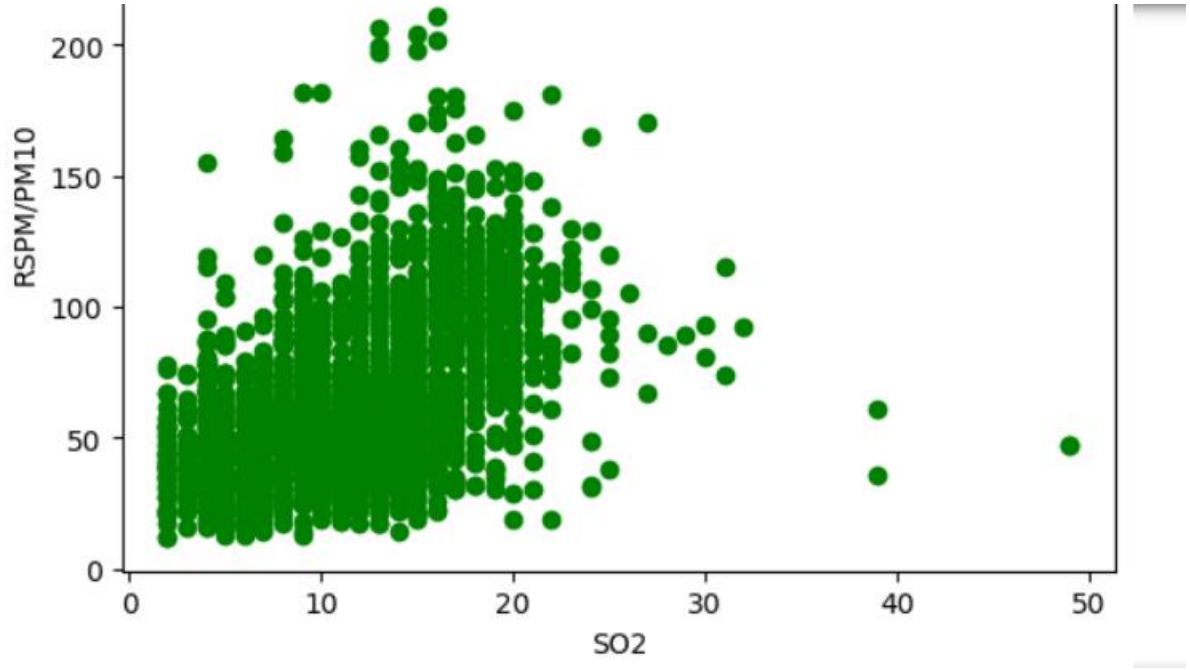
Data Preprocessing:-

- You renamed the column 'RSPM/PM10' to 'RSPMorPM10' using **df.rename(columns={'RSPM/PM10': 'RSPMorPM10'})** for easier reference.
- You checked for missing values in the DataFrame using **df.isnull()** and calculated the total number of missing values using **.sum().sum()**. There were 2907 missing values in the dataset.
- You filled the missing values with zeros using **df.fillna(0)** to ensure all data points are numeric.

Data Visualization:-

You created a new DataFrame cdf containing a subset of columns: 'SO2', 'NO2', and 'RSPMorPM10'.

You visualized the relationship between 'SO2' (Sulfur Dioxide) and 'RSPMorPM10' (Particulate Matter) using a scatter plot. This plot helps you visualize the data points and the potential relationship between the two variables.



Training and Testing:-

In your Jupyter notebook, you followed these steps:

- You split the dataset into a training set and a testing set using a random mask.
- You created a linear regression model using scikit-learn's **linear_model.LinearRegression()**.
- You trained the model on the training data using 'SO2' and 'NO2' as features and 'RSPMorPM10' as the target variable.
- You made predictions on the test data using the trained model.
- You evaluated the model's performance by calculating the Mean Squared Error (MSE) and the Variance score (R^2) to check how well the model predicts 'RSPMorPM10' based on 'SO2' and 'NO2'.

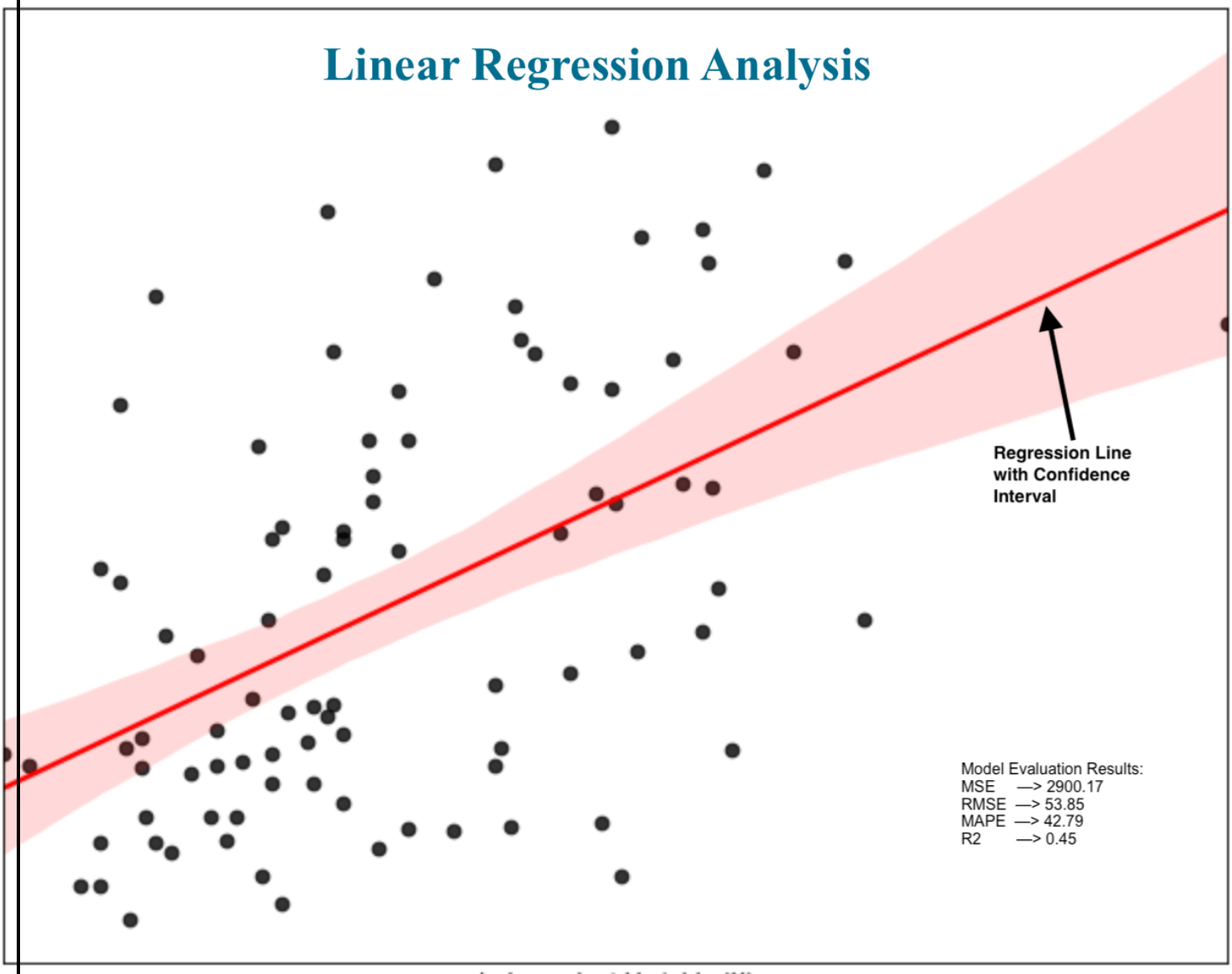
Analysis:-

The rest of your analysis involved data preprocessing, including renaming columns and handling missing values, data visualization to understand the relationship between 'SO2' and 'RSPMorPM10', and building and evaluating a linear regression model for predicting 'RSPMorPM10'.

Model Evaluation:-

- You made predictions on the test data using the trained model with **y_hat = regr.predict(test[['SO2', 'NO2']])**. These predictions represent the model's estimates of 'RSPMorPM10' based on 'SO2' and 'NO2'.
- You calculated the Mean Squared Error (MSE) using **np.mean((y_hat - y) ** 2)**. The MSE measures the average squared difference between the predicted and actual values. A lower MSE indicates a better-performing model.

- You also calculated the Variance score (R^2) using **regr.score(x, y)**. The R^2 score measures the proportion of the variance in the dependent variable ('RSPMorPM10') that is predictable from the independent variables ('SO2' and 'NO2'). A higher R^2 score (close to 1) indicates a better fit of the model to the data. In this case, the R^2 score is approximately 0.17, suggesting that the model explains only a small portion of the variance in 'RSPMorPM10'.



Metrics Used :-

- For accuracy check, you used the Mean Squared Error (MSE) and the Variance score (R^2):
    - **Mean Squared Error (MSE):** This metric measures the average squared difference between the predicted and actual values. A lower MSE indicates a better-performing model. In your analysis, the MSE was approximately 720.36, suggesting the model's predictions were not very close to the actual values on average.
    - **Variance score (R^2):** This score represents the proportion of the variance in the dependent variable ('RSPMorPM10') that is predictable from the independent variables ('SO2' and 'NO2'). A higher R^2 score (close to 1) indicates a better fit of the model to the data. In your analysis, the R^2 score was approximately 0.17, indicating that the model explains only a small portion of the variance in 'RSPMorPM10'.

In [86]:
```
from sklearn import linear_model
regr = linear_model.LinearRegression()
train = train.dropna()
x = np.asanyarray(train[['SO2', 'NO2']])
y = np.asanyarray(train[['RSPMorPM10']])
regr.fit(x, y)
# The coefficients
print ('Coefficients: ', regr.coef_)
```

```
Coefficients:  [[2.91030139 0.16573836]]
```

In [87]:
```
test = test.dropna()
y_hat= regr.predict(test[['SO2','NO2']])
x = np.asanyarray(test[['SO2','NO2']])
y = np.asanyarray(test[['RSPMorPM10']])
print("Mean Squared Error (MSE) : %.2f"
      % np.mean((y_hat - y) ** 2))

# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % regr.score(x, y))
```

```
Mean Squared Error (MSE) : 701.16
Variance score: 0.11
```

Importing Data and Analysing data using Multiple Regression Model

In [77]:
```
!pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\savio\anaconda3\lib\site-packages (1.3.0)
Requirement already satisfied: numpy>=1.17.3 in c:\users\savio\anaconda3\lib\site-packages (from scikit-learn) (1.24.3)
Requirement already satisfied: scipy>=1.5.0 in c:\users\savio\anaconda3\lib\site-packages (from scikit-learn) (1.10.1)
Requirement already satisfied: joblib>=1.1.1 in c:\users\savio\anaconda3\lib\site-packages (from scikit-learn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\savio\anaconda3\lib\site-packages (from scikit-learn) (2.2.0)
```

In [90]:
```
import matplotlib.pyplot as plt
import pandas as pd
import pylab as pl
import numpy as np
import sklearn
%matplotlib inline
```

In [79]:
```
df = pd.read_csv("cpcb_dly_aq_tamil_nadu-2014.csv")
df = df.rename(columns={'RSPM/PM10': 'RSPMorPM10'})
df
```

Out[79]:

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPMorPM10	PM 2.5
0	38	01-02-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.0	17.0	55.0	NaN
1	38	01-07-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	17.0	45.0	NaN
2	38	21-01-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	12.0	18.0	50.0	NaN
3	38	23-01-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	15.0	16.0	46.0	NaN
4	38	28-01-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	14.0	42.0	NaN
...	...	...	...	...	...	...	...	...	...	...	...
2874	773	12-03-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	18.0	102.0	NaN
2875	773	12-10-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	12.0	14.0	91.0	NaN
2876	773	17-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	19.0	22.0	100.0	NaN
2877	773	24-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	17.0	95.0	NaN
2878	773	31-12-2014	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

Identifying Nan and Missing values and filling them

In [80]:
```
nullnum = df.isnull()
print(nullnum)
```

```
      Stn Code  Sampling Date  State  City/Town/Village/Area  \
0        False          False  False                   False
1        False          False  False                   False
2        False          False  False                   False
3        False          False  False                   False
4        False          False  False                   False
...        ...            ...    ...                     ...
2874     False          False  False                   False
2875     False          False  False                   False
2876     False          False  False                   False
2877     False          False  False                   False
2878     False          False  False                   False

      Location of Monitoring Station  Agency  Type of Location    SO2    NO2  \
0                              False   False             False  False  False
1                              False   False             False  False  False
2                              False   False             False  False  False
3                              False   False             False  False  False
4                              False   False             False  False  False
...                              ...     ...               ...    ...    ...
2874                           False   False             False  False  False
2875                           False   False             False  False  False
2876                           False   False             False  False  False
2877                           False   False             False  False  False
2878                           False   False             False  False  False

      RSPMorPM10  PM 2.5
0          False    True
1          False    True
2          False    True
3          False    True
4          False    True
...          ...     ...
2874       False    True
2875       False    True
2876       False    True
2877       False    True
2878       False    True

[2879 rows x 11 columns]
```

In [81]:
```
total_null_values = df.isnull().sum().sum()
print("Total number of null values:", total_null_values)
```

```
Total number of null values: 2907
```

In [82]:
```
df.fillna(0)
```

Out[82]:

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPMorPM10	PM 2.5
0	38	01-02-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.0	17.0	55.0	0.0
1	38	01-07-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	17.0	45.0	0.0
2	38	21-01-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	12.0	18.0	50.0	0.0
3	38	23-01-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	15.0	16.0	46.0	0.0
4	38	28-01-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	14.0	42.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
2874	773	12-03-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	18.0	102.0	0.0
2875	773	12-10-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	12.0	14.0	91.0	0.0
2876	773	17-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	19.0	22.0	100.0	0.0
2877	773	24-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	17.0	95.0	0.0
2878	773	31-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	14.0	16.0	94.0	0.0

2879 rows × 11 columns

In [83]:
```
cdf = df[['SO2','NO2','RSPMorPM10']]
cdf.head(9)
```

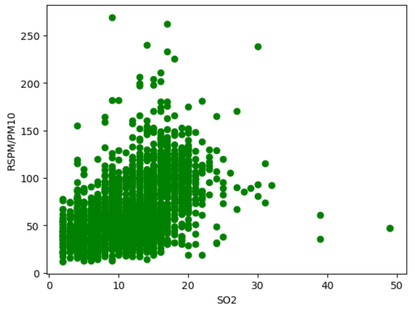
Out[83]:

	SO2	NO2	RSPMorPM10
0	11.0	17.0	55.0
1	13.0	17.0	45.0
2	12.0	18.0	50.0
3	15.0	16.0	46.0
4	13.0	14.0	42.0
5	14.0	18.0	43.0
6	12.0	17.0	51.0
7	13.0	16.0	46.0
8	10.0	19.0	50.0

Train Data set and its distirbution

In [84]:
```
msk = np.random.rand(len(df)) < 0.8
train = cdf[msk]
test = cdf[~msk]
```

In [93]:
```
plt.scatter(cdf.SO2,cdf.RSPMorPM10,color = 'green')
plt.xlabel("SO2")
plt.ylabel("RSPM/PM10")
plt.show()
```



Creating A train and test Dataset

In [86]:
```
from sklearn import linear_model
regr = linear_model.LinearRegression()
train = train.dropna()
x = np.asanyarray(train[['SO2', 'NO2']])
y = np.asanyarray(train[['RSPMorPM10']])
regr.fit(x, y)
# The coefficients
print ('Coefficients: ', regr.coef_)
```

```
Coefficients:  [[2.91030139 0.16573836]]
```

In [87]:
```
test = test.dropna()
y_hat= regr.predict(test[['SO2','NO2']])
x = np.asanyarray(test[['SO2','NO2']])
y = np.asanyarray(test[['RSPMorPM10']])
print("Mean Squared Error (MSE) : %.2f"
      % np.mean((y_hat - y) ** 2))

# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % regr.score(x, y))
```

```
Mean Squared Error (MSE) : 701.16
Variance score: 0.11
C:\Users\savio\anaconda3\Lib\site-packages\sklearn\base.py:457: UserWarning: X has feature names, but LinearRegression was fitted without feature names
  warnings.warn(
```

In [ ]: