Phase 4:
Air Quality Assessment of TamilNadu

Model Building:

Clustering Analysis:

Use unsupervised learning techniques like K-Means clustering or DBSCAN to group your data into clusters based on the available features (SO2, NO2, RSPM/PM10). This can help identify patterns or similarities in air quality data.

Importing Libraries:

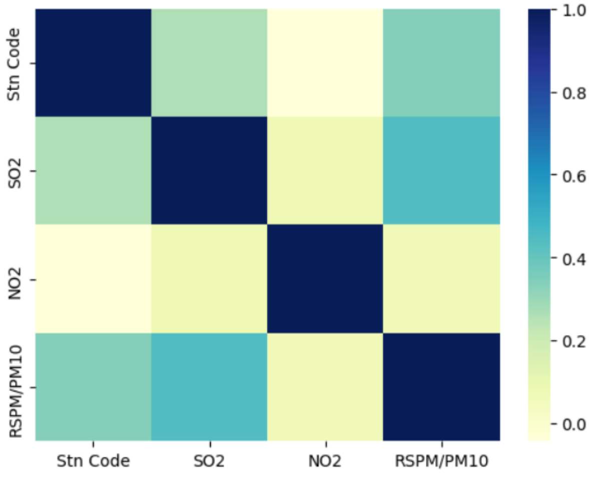
The code begins by importing the necessary Python libraries, including Pandas for data handling, NumPy for numerical operations, Scikit-Learn for machine learning, and Matplotlib for data visualization.

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
```

Feature Selection:

The code selects the features (independent variables) to be used for clustering, which are 'SO2,' 'NO2,' and 'RSPM/PM10.' These features will be used to determine the clusters.

```
import seaborn as sns
sns.heatmap(data.corr(),cmap='YlGnBu')
```



```
X = data[['SO2', 'NO2', 'RSPM/PM10']]
```

Feature Standardization:

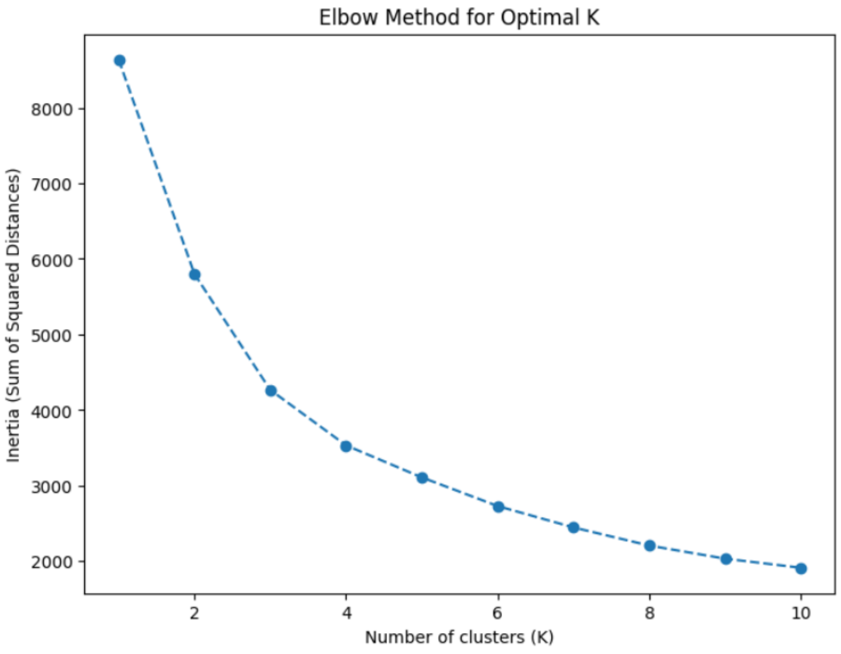
The features are standardized using the StandardScaler from Scikit-Learn. Standardization ensures that all features have a mean of 0 and a standard deviation of 1, which is important for K-Means clustering.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X = scaler.fit_transform(X)
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=0).fit(X)
    inertia.append(kmeans.inertia_)
```

Determine the Optimal Number of Clusters:

The code then uses the Elbow method to find the optimal number of clusters (K). It iterates through different values of K and calculates the inertia, which is the sum of squared distances from data points to their assigned cluster centers. The Elbow method plots these inertias for various K values to help you identify the "elbow point" where increasing K doesn't significantly reduce the inertia.

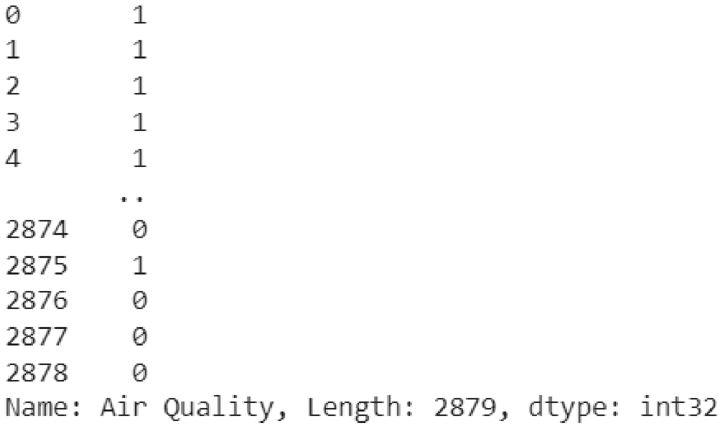
```
plt.figure(figsize=(8, 6))
plt.plot(range(1, 11), inertia, marker='o', linestyle='--')
plt.title('Elbow Method for Optimal K')
plt.xlabel('Number of clusters (K)')
plt.ylabel('Inertia (Sum of Squared Distances)')
plt.show()
```



K-Means Clustering:

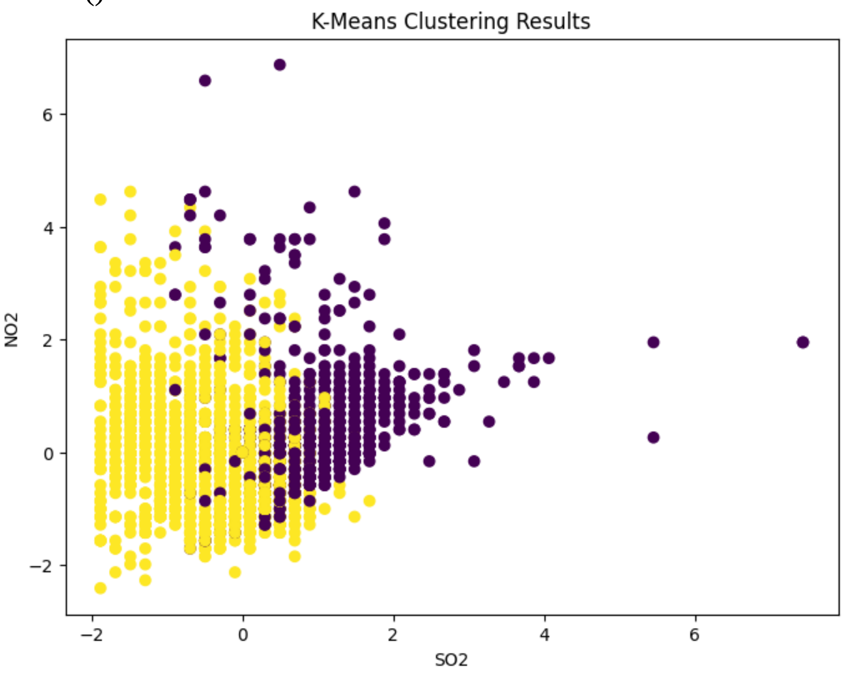
After determining the optimal K (in this case, K = 3), the code performs K-Means clustering using the KMeans algorithm from Scikit-Learn. The clusters are assigned to the 'Cluster' column in the dataset.

kmeans = KMeans(n_clusters=2, random_state=0)
data['Air Quality'] = kmeans.fit_predict(X)



```
0       1
1       1
2       1
3       1
4       1
       ..
2874    0
2875    1
2876    0
2877    0
2878    0
Name: Air Quality, Length: 2879, dtype: int32
```

plt.figure(figsize=(8, 6))
plt.scatter(X[:, 0], X[:, 1], c=data['Air Quality'], cmap='viridis')
plt.title('K-Means Clustering Results')
plt.xlabel('SO2')
plt.ylabel('NO2')
plt.show()

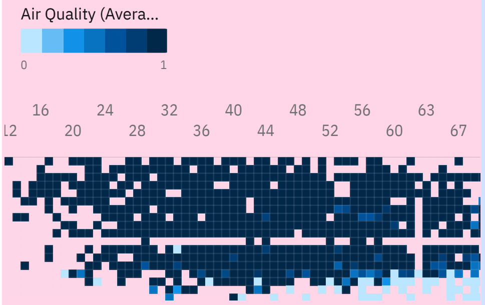


Visualization and Insights:

1. The RSPM/PM10 and Air Quality relationship is weakly influenced by SO2.
2. RSPM/PM10 44 has the highest Air Quality at 16.63, out of which SO2 2 contributed the most at 1.
3. Air Quality is most unusual when City/Town/Village/Area is Trichy, Coimbatore and Mettur.
4. Chennai is the most frequently occurring category of City with a count of 1000 items with Air Quality values (34.7 % of the total).
5. Over all air qualities, the average of NO2 is 22.14, the average of RSPM/PM10 is 62.49,the average of SO2 is 11.5.

Tab 1

Air Quality by SO2 and RSPM/PM10



RSPM/PM10

180K

RSPM/PM10

NO2

63.7K

NO2

SO2

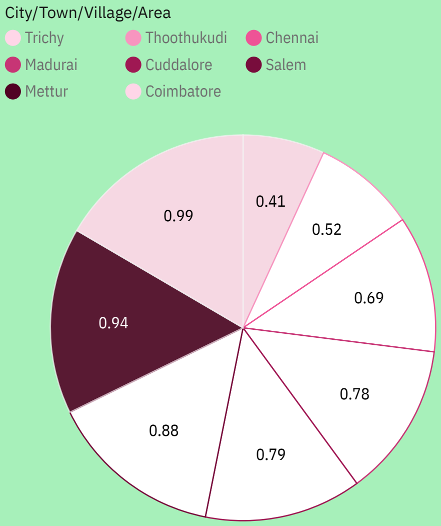
33.1K

SO2

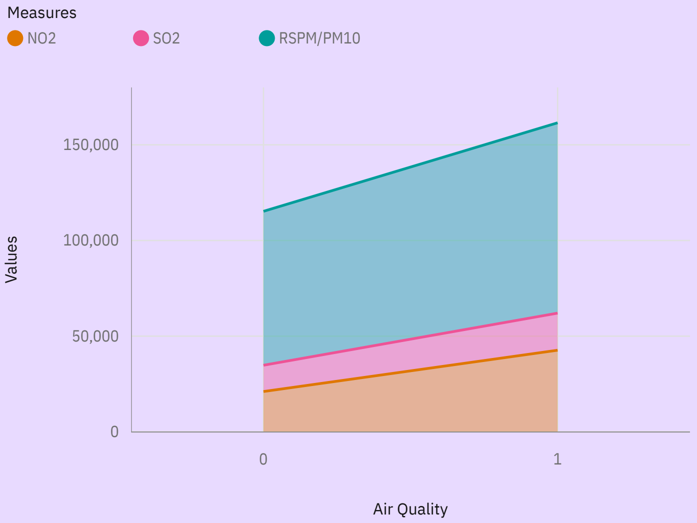
Air Quality

0 1

Air Quality by City/Town/Village/Area



NO2, SO2 and RSPM/PM10 by Air Quality


Measures
NO2
SO2
RSPM/PM10
Values
Air Quality