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:

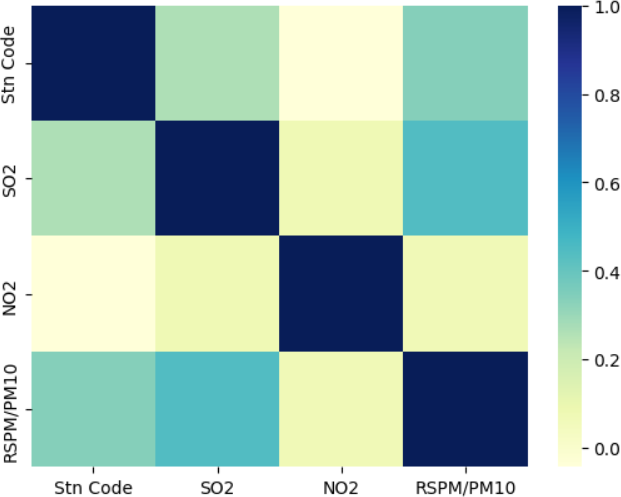
The code begins by importing the necessary Python libraries, including Pandas for data handling, NumPy for numerical operations, Scikit-Learn for machine learning, and Matplotlib for data visualization.

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):

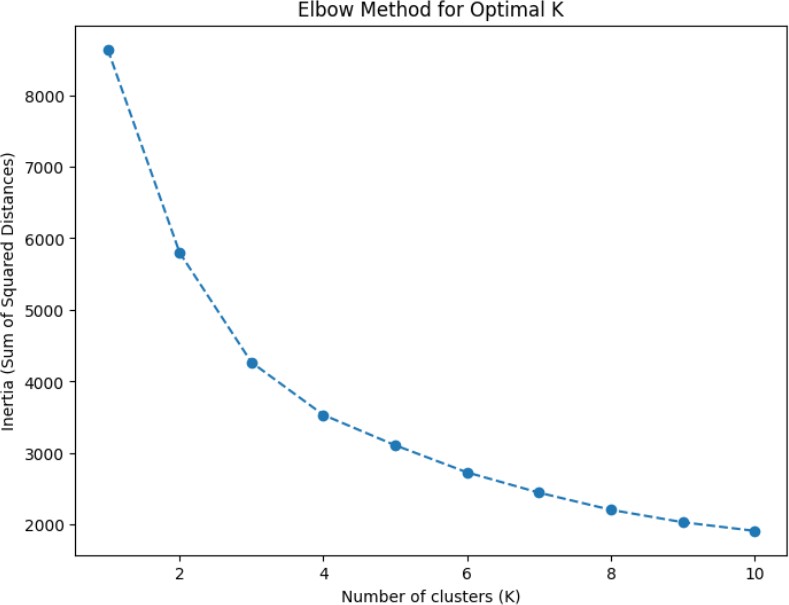
kmeans = KMeans(n\_clusters=k, random\_state=0).fit(X) inertia.append(kmeans.inertia\_)

# 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))

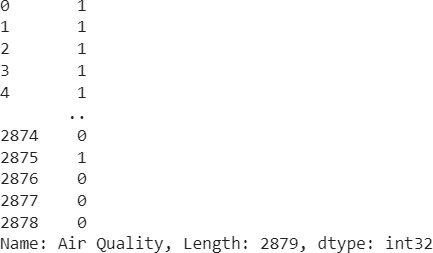
plt.plot(range(1, 11), inertia, marker='o', linestyle='--') plt.title('Elbow Method for Optimal K') plt.xlabel('Number of clusters (K)')

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)



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:

* Chennai has the highest RSPM/PM10 level at 654, with SO2 contributing the most at 59.
* Coimbatore has an RSPM/PM10 level of 61.
* From January 30, 2014, to January 31, 2014, the RSPM/PM10 level in location 10 increased by 300%.
* Chennai has the highest SO2 levels, with Station Code 161 being the primary contributor.
* Chennai is the most frequently occurring City/Town/Village/Area category, accounting for 1000 items with RSPM/PM10 values (34.7% of the total).
* The total number of results for RSPM/PM10 across all City/Town/Village/Area categories is nearly three thousand.

