CP 318 Data Science for Smart City Applications
Project 2: Exploratory Data Analysis
Team: X-MEN | MD FARHAD HUSSAIN (23113) | ASHHAR ZAMAN (22881)

1 Motivation & Problem Statement

In the context of Pune, a Smart City, understanding and predicting air quality across different areas is crucial for effective city planning and management. The objective of this project is to employ clustering techniques on the Smart City Testbed data, specifically the Air Quality Index (AQI) parameters, to unveil patterns and group similar areas within Pune. By leveraging machine learning clustering algorithms, the aim is to categorize locations with similar air quality characteristics, providing valuable insights into the spatial distribution of air pollution across the city.

2 Dataset description

This data is a subset of the Smart City Testbed collected by Pune Smart City Development Corporation Limited (PSCDCL) and IISC, Bangalore in 2019 trying to solve simple to complex use cases using smart city testbed. Data Frequency: From 01-Apr-2019 To 31-Jul-2019 every 15 minutes interval Date format : DD/MM/YYYY HH:MM:SS (24 Hrs format) Y or Dependent Variable <- Air quality Index (AQI) X or Independent Variables <- Smart city Indexing Parameters. The dataset has been sourced from the Kaggle repository accessible at
Pune Smart City Dataset . Notably, no prior work has been undertaken on this dataset.

3 Data Processing

3.1 Handling Null Values

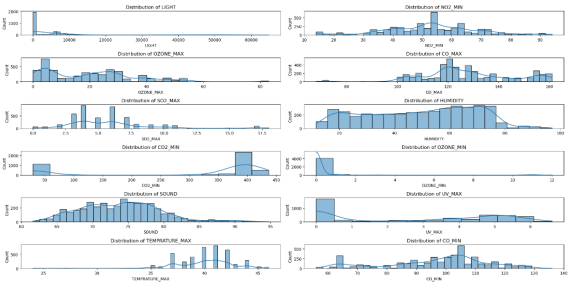
Certain columns, such as NO_MAX and NO_MIN, containing null values, have been excluded from the dataset as they do not contribute to subsequent analyses. The percentage of missing values was observed to be minimal, prompting the decision to impute the missing values instead of removing entire rows. Subsequently, a histogram was generated for a specific location to assess data distribution. The analysis revealed a non-Gaussian distribution, leading to the choice of filling missing data using the median of the corresponding location, considering the potential variation in median values across different locations.

3.2 Resampling

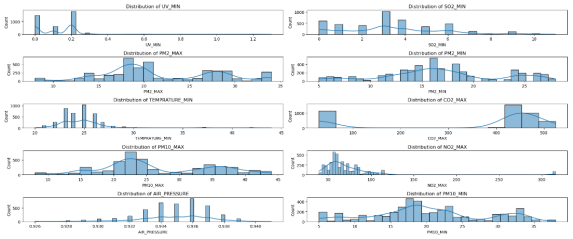
Given the time series nature of the data, previously mentioned to be sampled at 15-minute intervals, a decision was made to address the high volume of instances. The dataset, initially comprising approximately 100,000 instances, underwent resampling at a daily frequency. This strategic resampling reduced the number of instances to around 1,200. Apart from achieving a more manageable dataset size, this resampling also functioned as a means of noise filtering within the data. While resampling the data the resampling command was putting null values in the rows of those time intervals which were absent from the original data resulting increased number of instances, those rows were removed by dropping the null rows. The resampling at an hourly frequency was also done but the MDS of hourly data was taking more computing power so the mean of the data is taken for resampling.

3.3 Robust scaling and outlier detection

The distribution of our data deviates from normality, making robust scaling an appropriate choice, especially given its resilience to the influence of outliers figure 1. Additionally, considering that environmental data from each location is collected using different sensors, scaling the data on a per-location basis is deemed necessary for accurate scaling figure 2.

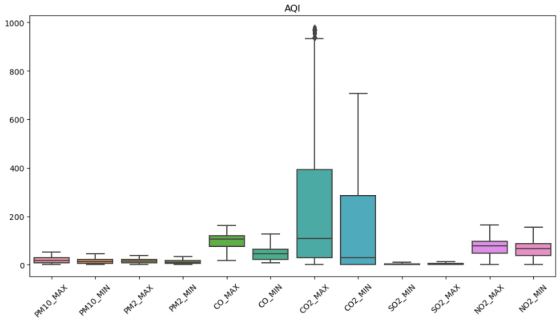


(a)

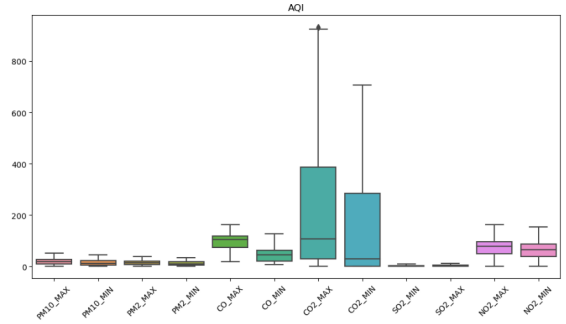


(b)

Figure 1: Data distribution of Bopadi_Square_65



(a) Before Outlier removal

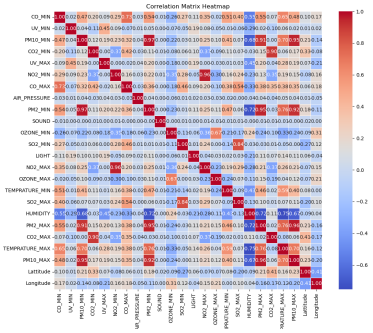


(b) After Outlier removal

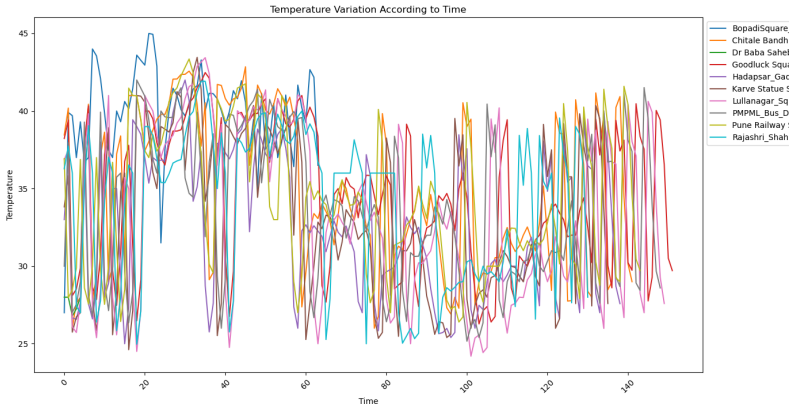
Figure 2: Box plots of the data before and after outlier removal

4 Descriptive Analysis

The correlation matrix is a fundamental component of the descriptive analysis, offering insights into the relationships between different variables within the dataset. Following the creation and analysis of the correlation matrix, notable features have been identified and considered for further analysis. It is found that the temperature



(a) Correlation Matrix



(b) Time vs Temperature graph

Figure 3

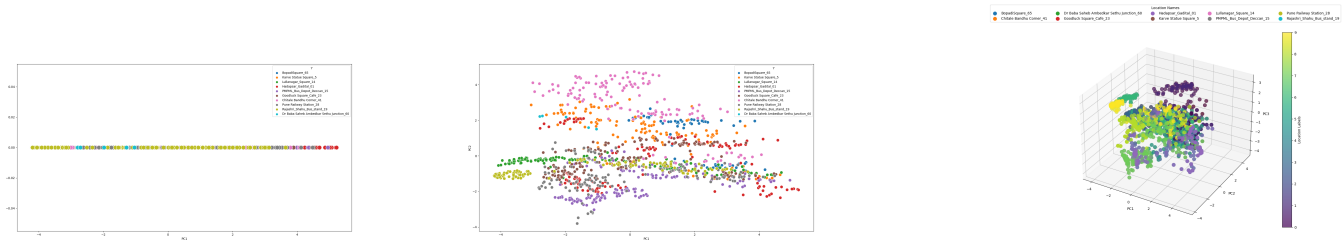
is more correlated to the AQI parameters figure 3a. It can be seen that some location are following similar patterns so we can say that there is a possibility of some clustering 3b.

5 Dimension Reduction

The primary goal is to simplify the dataset while retaining its essential characteristics, patterns, and information.

5.1 Principal Component Analysis

In this section, Principal Component Analysis (PCA) was conducted on a subset of features related to the Air Quality Index (AQI). The selected features were standardized, and the resulting data was subjected to PCA to identify key patterns and reduce dimensionality. The top three principal components were chosen, and the standardized data was projected onto these components. The results were visualized through scatter plots.



(a) PCA in 1D (b) PCA in 2D (c) PCA in 3D

Figure 4: Principal Component Analysis

5.2 Multi-Dimensional Scaling (MDS)

In this section, Multi-Dimensional Scaling (MDS) was employed to reduce the dimensionality of the dataset to two dimensions figure 7a.

5.3 Location Clustering Visualization

After determining the optimal number of clusters (k=3) using the elbow method, the k-means clustering and spectral algorithm was applied to the latitude and longitude features of the dataset. The resulting clusters were then visualized on both a 2D scatter plot and an interactive Folium map figure 5. The Folium map visualizes the location clusters in an interactive manner. Each circle marker on the map represents a location, with color indicating its assigned cluster. This dynamic visualization allows for a geographic understanding of the clustering results.

(a) Clustering (b) Folium Pune map

Figure 5: Location Clustering

6 Clustering Analysis

In this section, we aimed to identify optimal clusters for different locations in Pune using the K-means algorithm based on the features of AQI. The Elbow Method was employed to determine the most suitable number of clusters (k). AQI features taken are PM10_MAX, PM10_MIN, PM2_MAX, PM2_MIN, CO_MAX, CO_MIN, CO2_MAX, CO2_MIN, SO2_MIN, SO2_MAX, NO2_MAX, NO2_MIN.

6.1 Elbow Method for Optimal k

The Elbow Method involves plotting the sum of squared distances (inertia) for different values of k and identifying the "elbow" point, where the rate of decrease in inertia slows down. This point signifies a balance between

maximizing within-cluster similarity and minimizing the number of clusters. After applying the elbow method and manually analyzing its corresponding plot, we determined that the optimal number of clusters for the subsequent clustering analysis is $k = 3$.

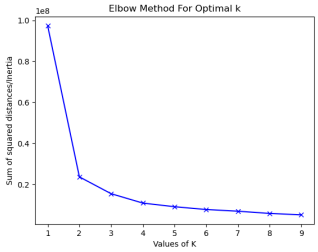
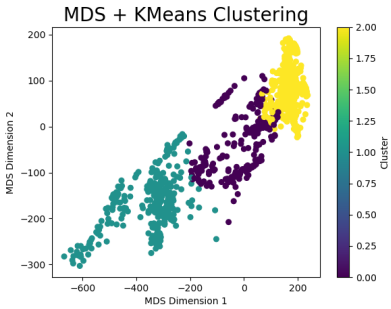


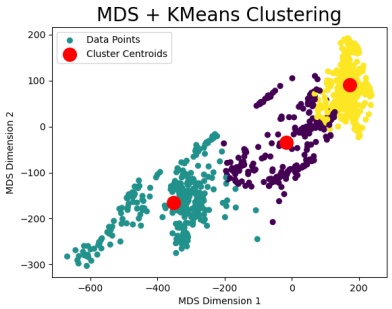
Figure 6: Caption

6.2 K-Means with MDS

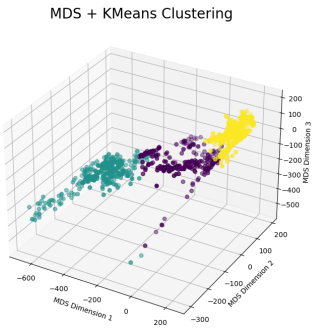
In this phase of the analysis, we applied KMeans clustering to the features reduced using Multi-Dimensional Scaling (MDS). The MDS algorithm was employed to reduce the dimensionality of the dataset to two and three dimensions. The number of clusters (k) was predefined as 3 for this analysis figure 7b and figure 7c



(a) MDS visualization in 2D



(b) Kmeans with MDS in 2D

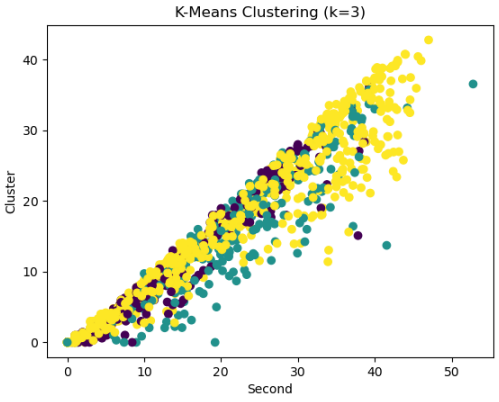


(c) Kmeans with MDS in 3D

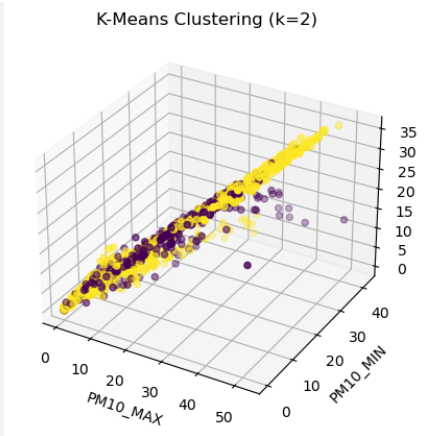
Figure 7: Kmeans with MDS

6.3 K-means without MDS

In this section, we applied the K-Means clustering algorithm to explore patterns and groupings within the dataset based on Air Quality Index (AQI) parameters and can see that it is not making clusters that is properly visible figure 8.



(a) K means with k=3 without MDS

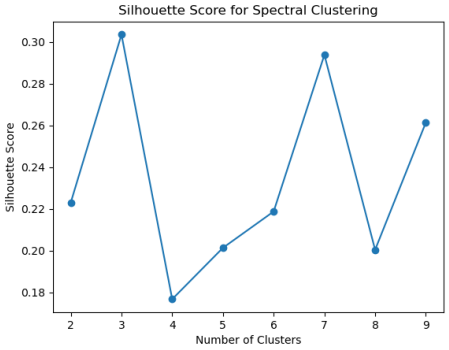


(b) K means with k=2 without MDS

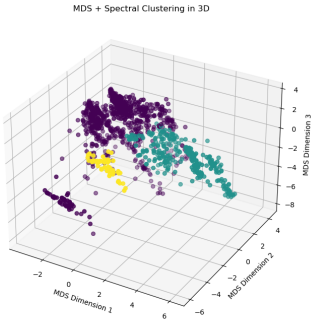
Figure 8: K means without MDS

6.4 Spectral Clustering

The silhouette score analysis helps determine the optimal number of clusters for Spectral Clustering. Aiding in the selection of an appropriate number of clusters figure 9a. The 3D visualization demonstrates the results of Spectral Clustering applied to the log-transformed and standardized AQI parameters. Each data point is color-coded according to its assigned spectral cluster, offering insights into the spatial distribution of clusters in the reduced 3D feature space figure 9b.



(a) Silhoutte Score



(b) Spectral Clustering

Figure 9: Spectral Clustering

7 Conclusion and Insights

In comparing PCA and MDS for our non-normally distributed data, MDS demonstrated better cluster separability despite requiring more computing power and time. PCA, while faster, captured less variance (82%) with three principal components. Given the non-normally distributed nature of our data, MDS appears to be a more suitable choice for revealing distinct clusters. The clustering analysis identified three distinct clusters of locations in Pune based on air quality characteristics. The spatial distribution of clusters was visualized through scatter plots and an interactive Folium map, providing a geographic understanding of air quality patterns of Pune which can be further used for making policies regarding urban planning, public health, optimize resource allocation, implement targeted interventions, and enhance smart city development etc. The location with more AQI are usually found on the highways and urban area and vice versa according to the analysis.

8 References

- "Pune Smart City Dataset." Https://Www.Kaggle.Com/Datasets/Akshman/Pune-smartcity-test-dataset.
- jssuriyakumar. "How to Resample Time Series Data in Python?" Geeksforgeeks. December 19, 2021. https://www.geeksforgeeks.org/how-to-resample-time-series-data-in-python/.
- Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
- J. D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.
- Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.
- McKinney, W., & others. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).
- python-visualization. (2020). Folium. Retrieved from https://python-visualization.github.io/folium/
- da Costa-Luis, (2019). tqdm: A Fast, Extensible Progress Meter for Python and CLI. Journal of Open Source Software, 4(37), 1277, https://doi.org/10.21105/joss.01277