Programming Assignment 3 Part B

LANKA RAMA KRISHNA 22EE30037

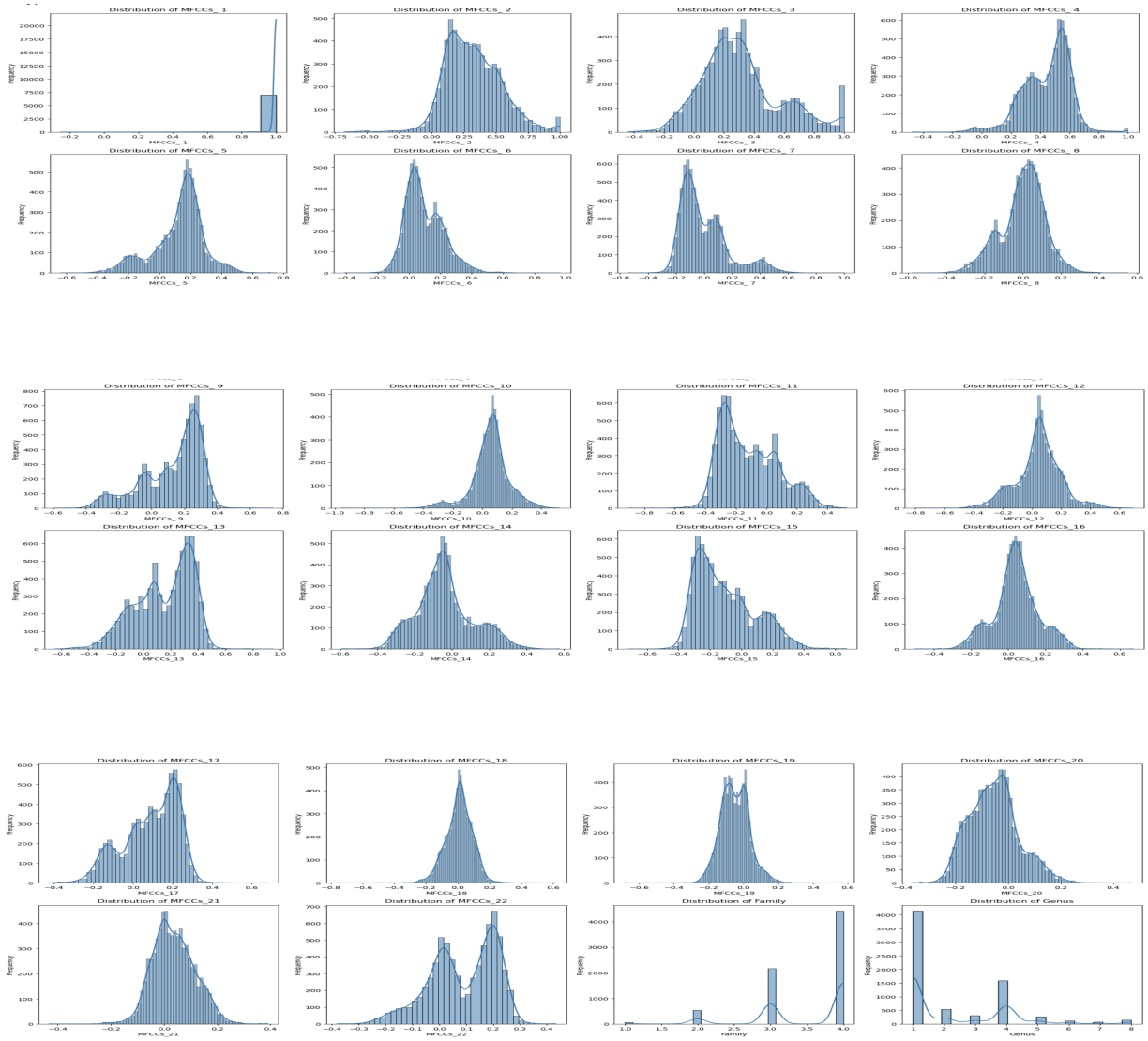
November 5, 2024

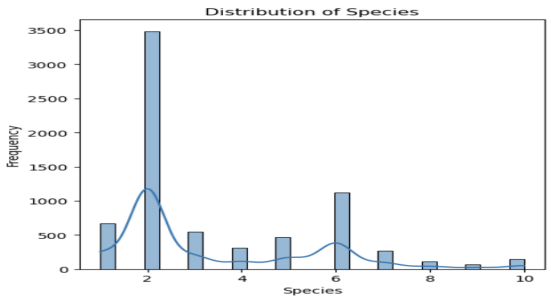
K-Means Clustering - Anuran Calls Dataset (MFCCs)

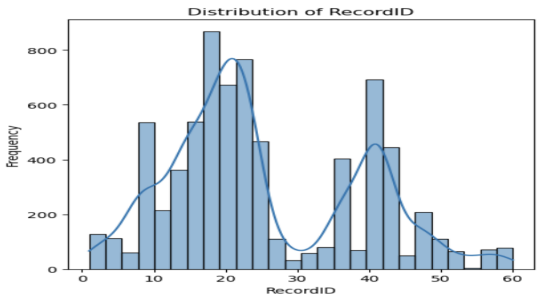
1 Data Preprocessing and Exploration

Exploratory Data Analysis (EDA)

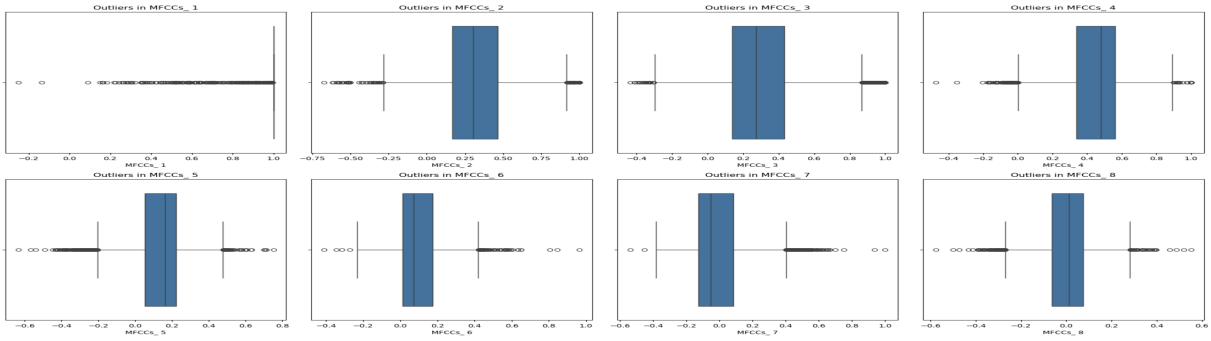
Feature distributions

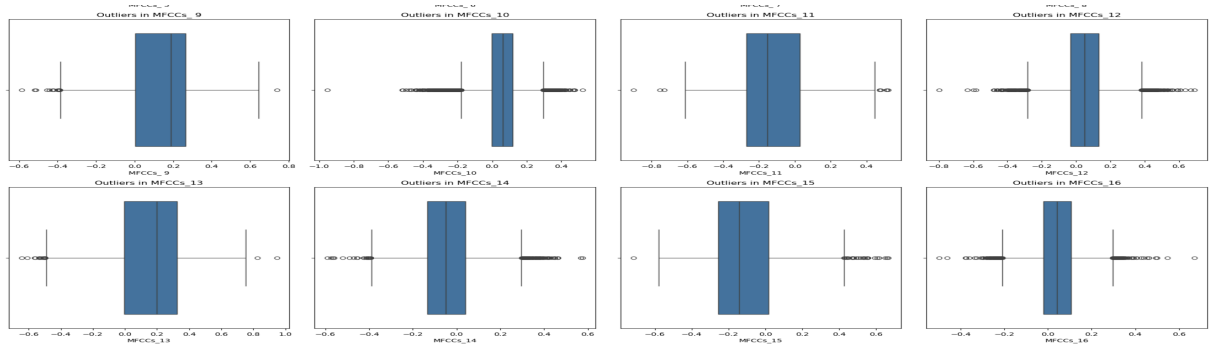


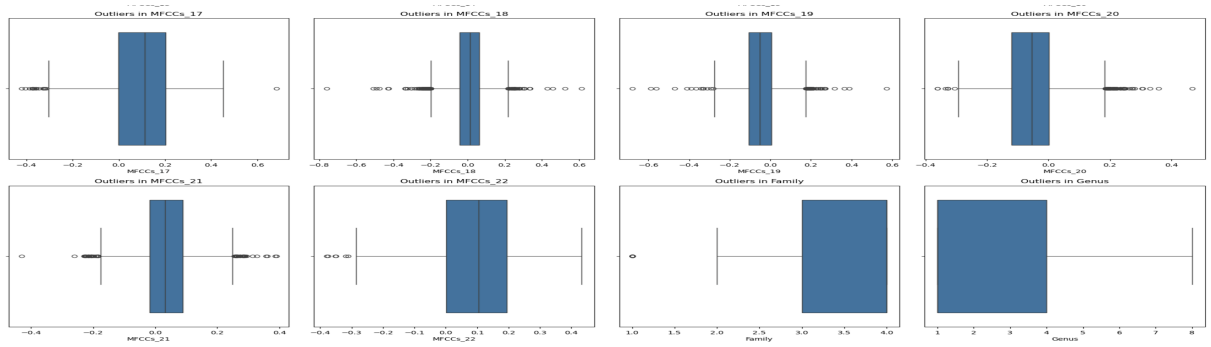


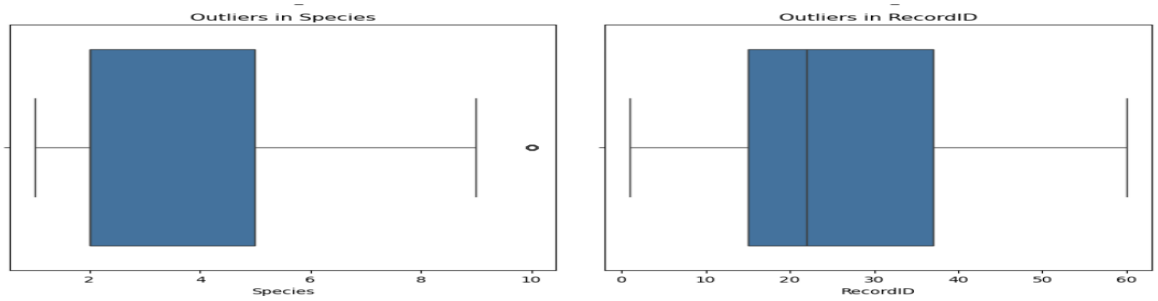


Boxplots for outlier detection









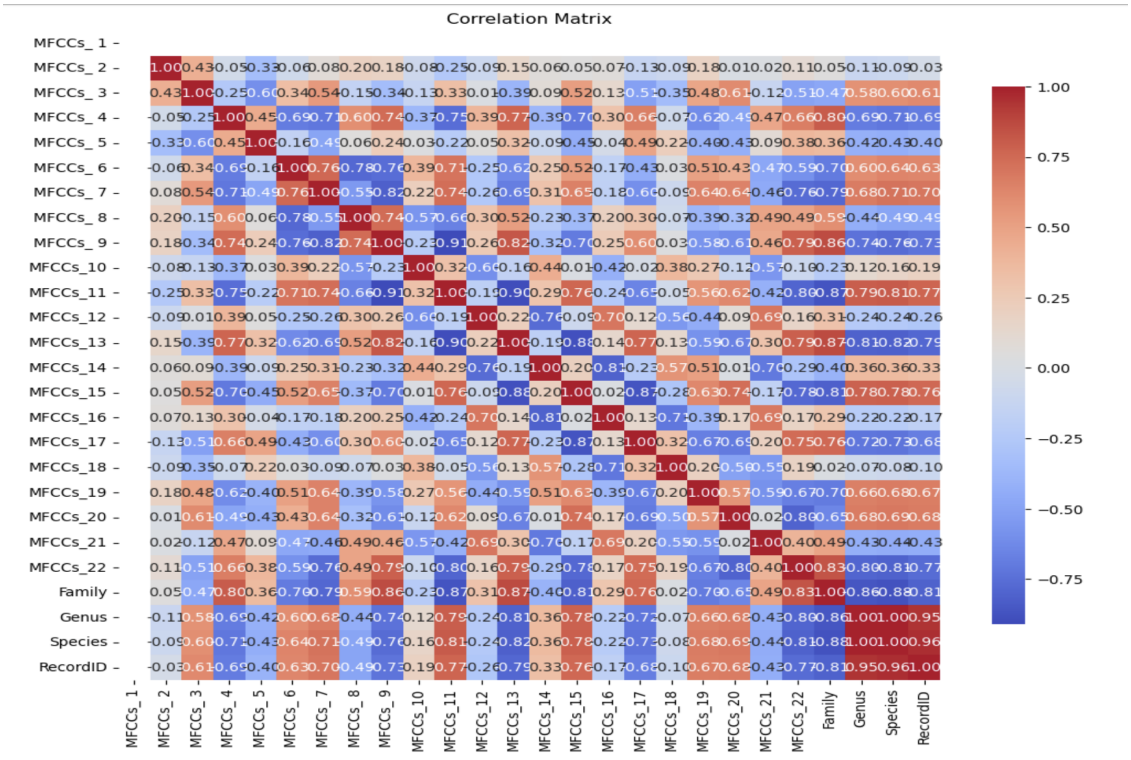
The above feature distributions and box plots were analysed to detect outliers and get an overview of the dataset. and after the data was normalised .

```
Data shape before removing outliers: (7195, 26)
Data shape after removing outliers: (4295, 26)
```

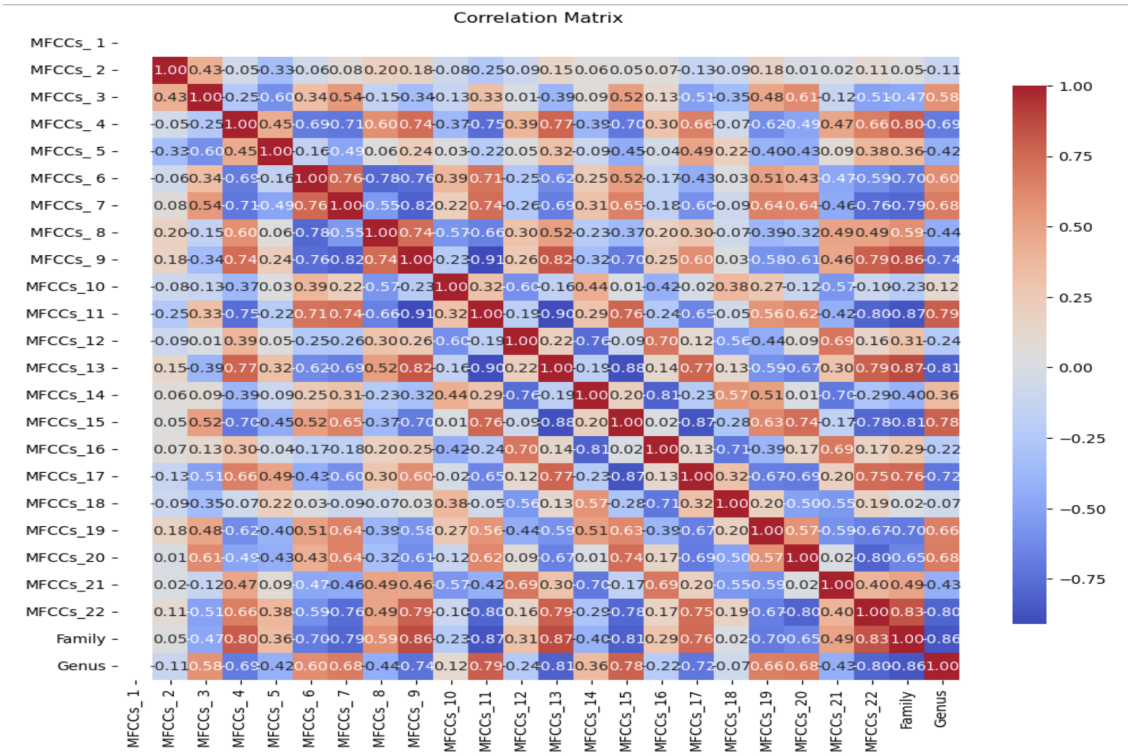
The detected outliers were removed and number of rows decreased.

Feature Correlation Analysis

The correlation matrix for the dataset was plotted.



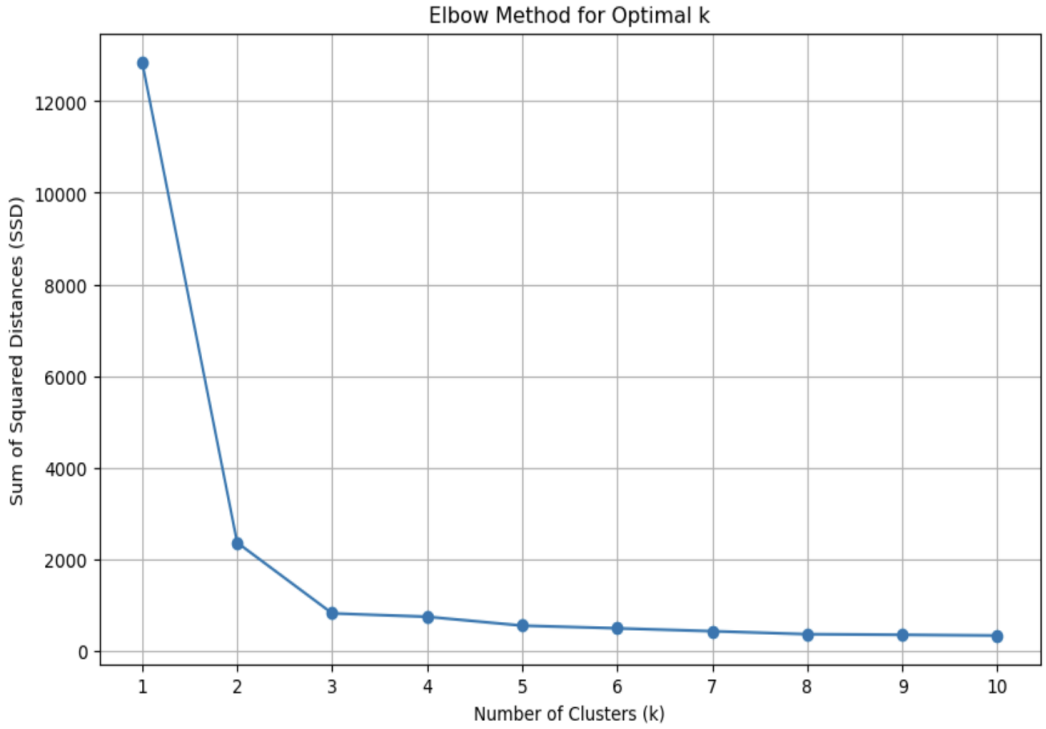
we can see from the plot that Genus Species and RecordID have high correlation to each other so Species and RecordID were removed .



In the feature engineering phase, we generated polynomial features up to degree 2 then selected the top 25 most significant polynomial features to optimize model performance and reduce dimensionality.

2 K-Means Clustering

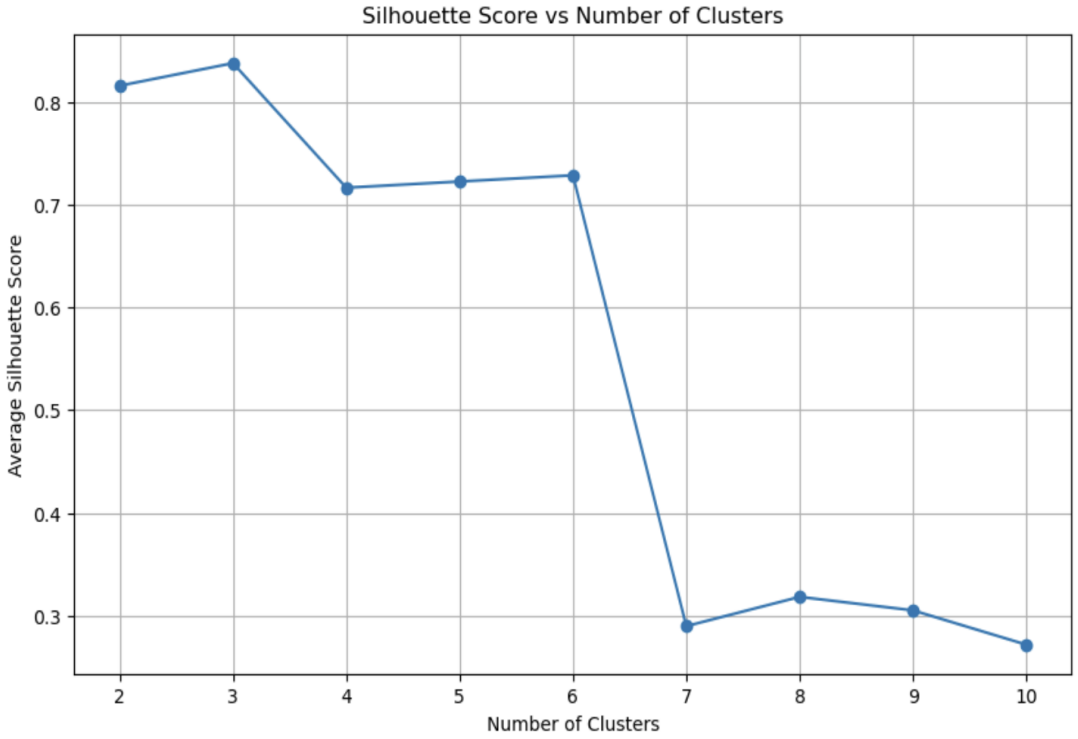
2.1 Elbow Method



The Elbow Method was implemented to determine the optimal number of clusters for the dataset. By running the K-Means clustering algorithm across a range of cluster numbers, the within-cluster sum of squares (WCSS) was plotted against the number of clusters. The resulting graph revealed a distinct "elbow" point, indicating the most suitable number of clusters for further analysis.

From the plotted results, two noticeable decreases in the slope were observed at $K = 2$ and $K = 3$. While $K = 2$ marks an initial elbow point, $K = 3$ was selected as the optimal cluster count, as the WCSS error further decreases and stabilizes, providing improved clustering performance with minimal additional cost. Thus, $K = 3$ is chosen as the most appropriate value for this analysis.

2.2 Silhouette Score Evaluation



A range of cluster numbers from 2 to 10 was evaluated using the silhouette score to assess clustering quality. For each cluster configuration, the K-Means algorithm was applied to the dataset, generating cluster labels. The average silhouette scores were calculated and plotted against the number of clusters. This plot helps identify the optimal number of clusters based on the highest silhouette score, providing insights into the clustering structure of the dataset.

```
Number of clusters: 2, Silhouette Score: 0.8155
Number of clusters: 3, Silhouette Score: 0.8374
Number of clusters: 4, Silhouette Score: 0.7163
Number of clusters: 5, Silhouette Score: 0.7223
Number of clusters: 6, Silhouette Score: 0.7283
Number of clusters: 7, Silhouette Score: 0.2899
Number of clusters: 8, Silhouette Score: 0.3183
Number of clusters: 9, Silhouette Score: 0.3052
Number of clusters: 10, Silhouette Score: 0.2719
Number of clusters for maximum Silhouette Score: 3
```

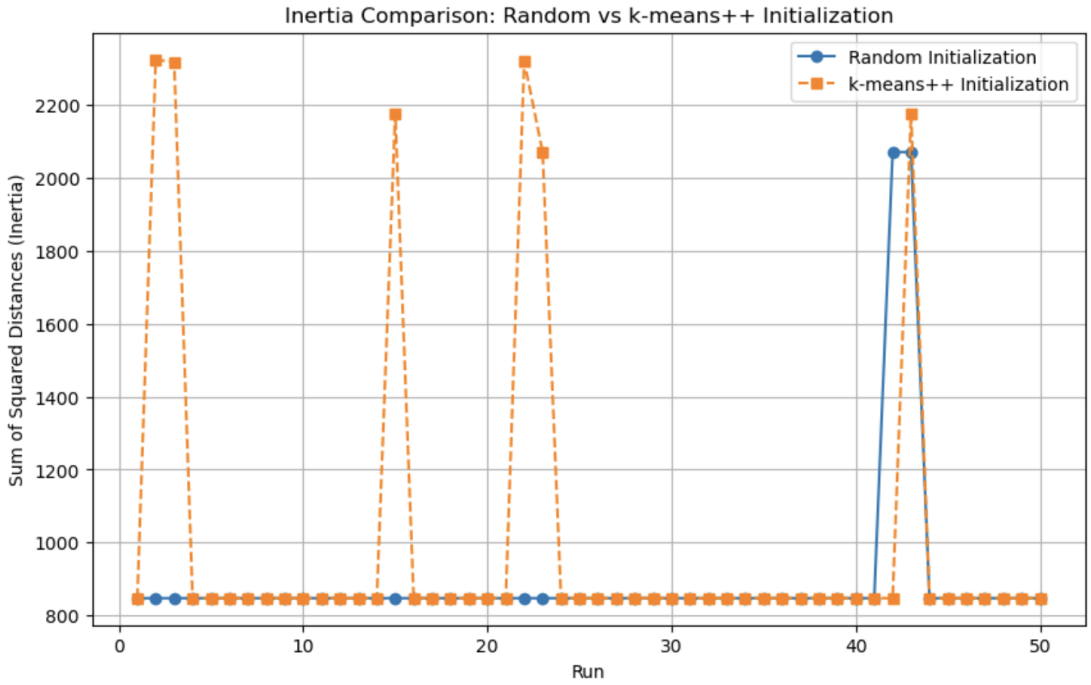
From the analysis of the silhouette score plot, it can be concluded that the optimal number of clusters is 3. This is indicated by the highest average silhouette score observed at this point, suggesting that three clusters provide the best separation and cohesion among the data points. Thus, clustering the data into three distinct groups enhances the overall clustering quality.

2.3 Cluster Implementation

K-Means clustering was then implemented for $K = 2$ and $K = 3$, as these values were identified as optimal based on the silhouette score and the Elbow Method. Although $K = 3$ is the optimal value, $K = 2$ was also included in the analysis to visualize the clustering results and assess the differences in clustering quality and structure between the two configurations.

2.4 Cluster Initialization

we compared the K-Means clustering performance using two initialization methods: random and k-means++. We executed K-Means 50 times for each method, recording the inertia values (sum of squared distances) for each run. We then plotted the inertia values to visualize the differences between the two methods and calculated the average inertia for each, allowing us to evaluate which initialization strategy produced better clustering results.




```
Average inertia with Random Initialization: 896.2567900741735
Average inertia with k-means++ Initialization: 1013.4011516290485
```

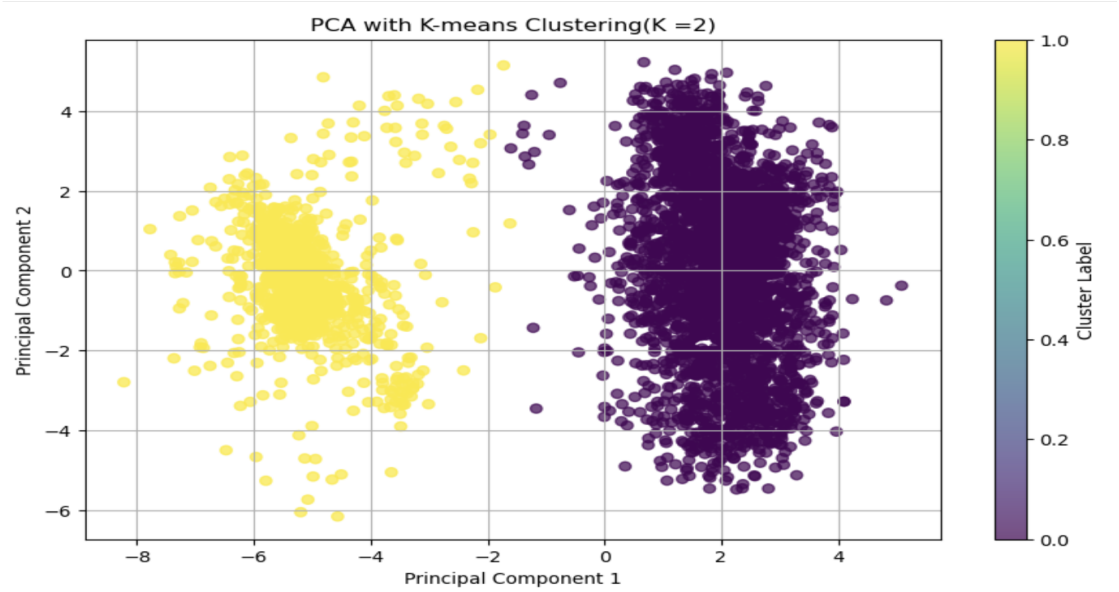
Since k is small (3), the choice of initialization method for K-Means has minimal impact on clustering results. With fewer cluster centers to optimize, both random and k-means++ initializations lead to similar solutions. Additionally, the algorithm converges quickly, and the simplicity of the data structure reduces variability. This makes it less likely to get stuck in poor local minima, resulting in consistent cluster assignments regardless of the initialization method.

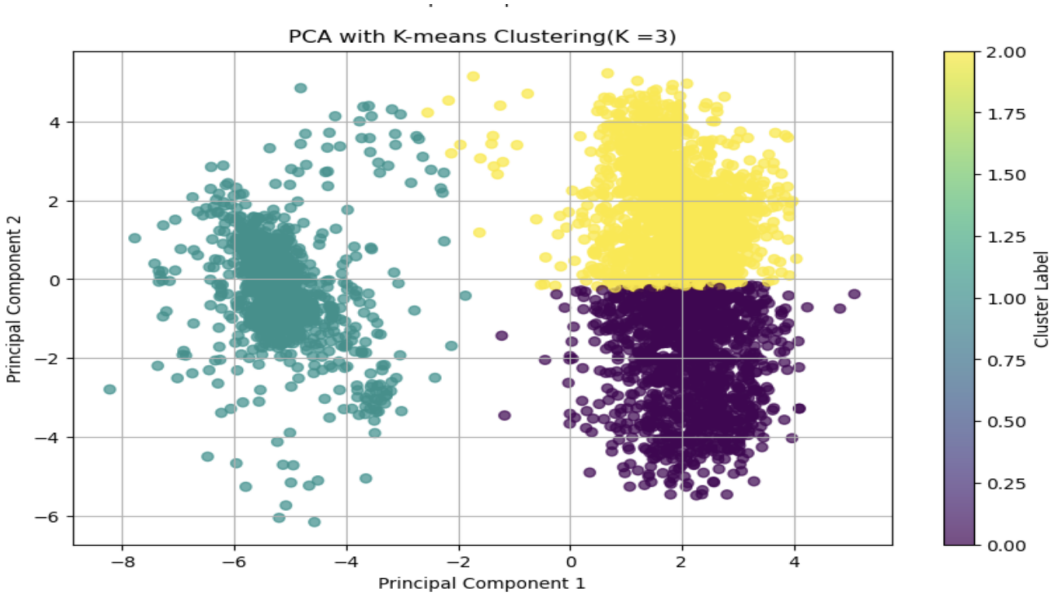
3 Cluster Visualization

We applied PCA to reduce the dataset to two dimensions, followed by K-Means clustering with $K = 2$ and $K = 3$ clusters.

- **Data Standardization**: The dataset is standardized to normalize feature scales.
- **Dimensionality Reduction**: PCA reduces the data to two principal components, capturing the most variance in 2D space.
- **Clustering**: K-Means is applied to the reduced data for $K = 2$ and $K = 3$. The results are visualized with scatter plots showing cluster assignments in the PCA-transformed space.

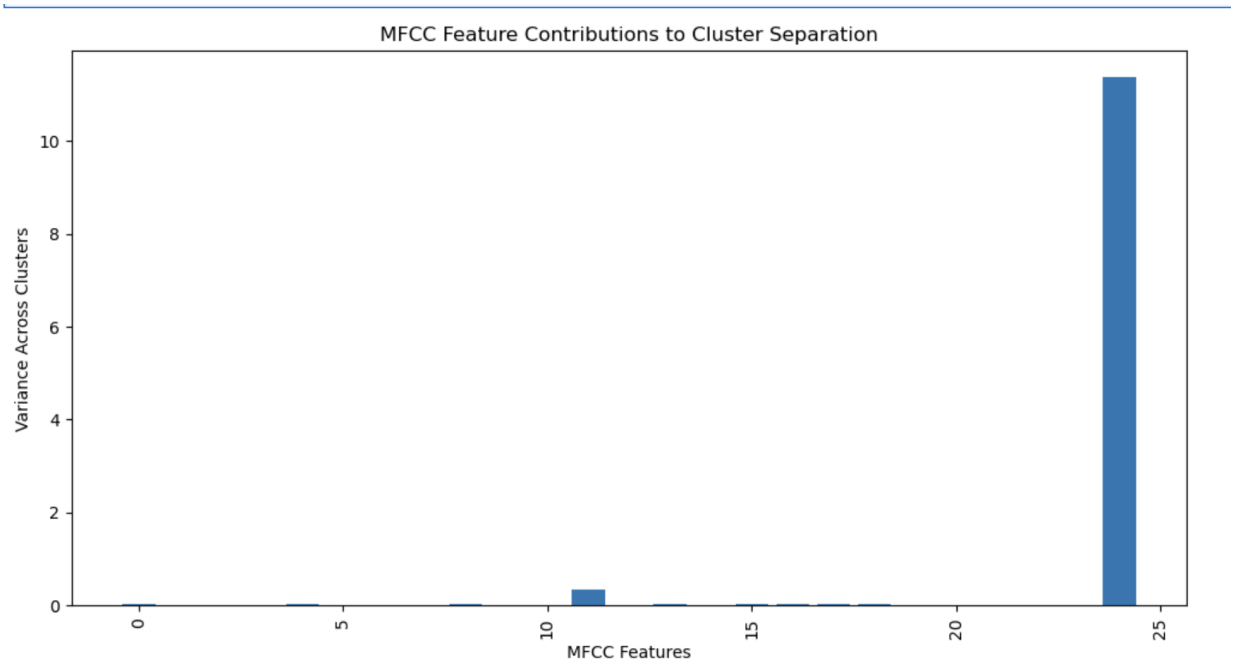
This approach simplifies visualization and reveals cluster structures effectively.





Feature Contribution to Clustering

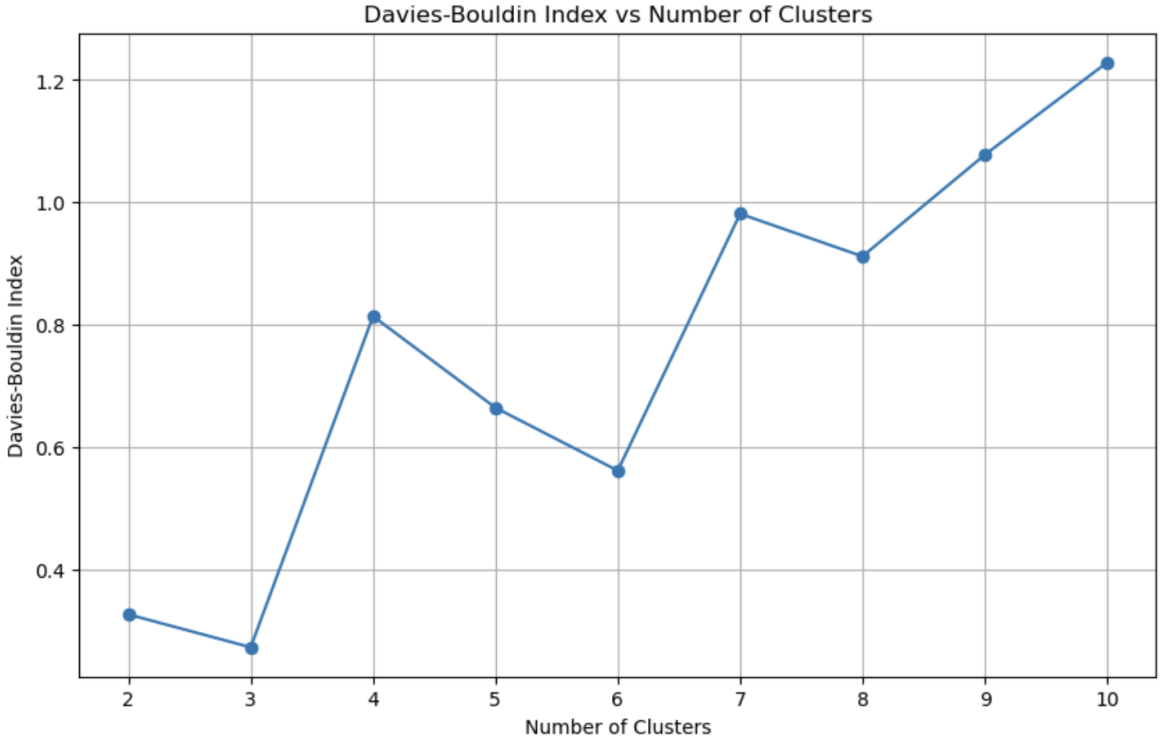
we perform K-Means clustering on a dataset X X containing MFCC features, defining three clusters. After fitting the K-Means model, we calculate the variance of each MFCC feature across the identified clusters to assess their contributions to cluster separation. Finally, we visualize these variances using a bar plot, allowing us to see which MFCC features play a significant role in distinguishing the clusters.



4 Cluster Evaluation Metrics

4.1 Evaluating using Davies-Bouldin Index

In this analysis, we examined the optimal number of clusters for our dataset using K-Means clustering and the Davies-Bouldin Index (DBI). We evaluated cluster counts from 2 to 10, calculating the DBI for each configuration to identify the minimum value, indicating the best clustering quality. To visualize our findings, we plotted the DBI against the number of clusters, helping us interpret the relationship and select the optimal cluster number more effectively.

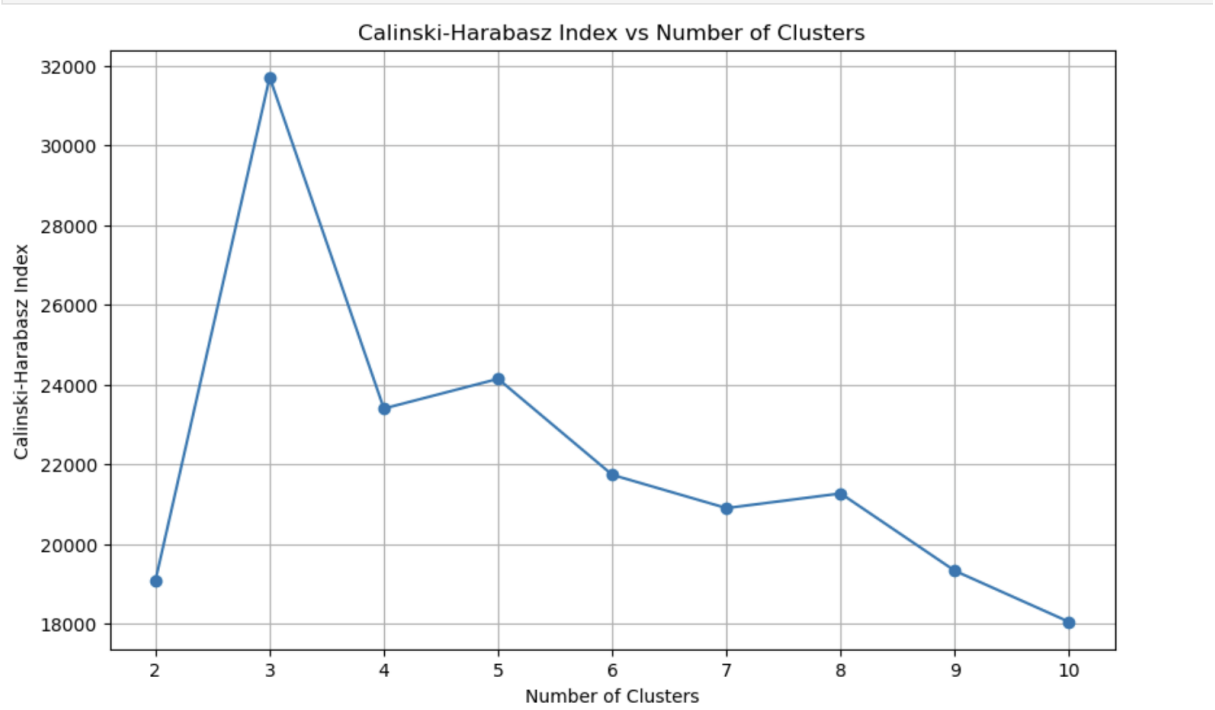



```
Number of clusters: 2, Davies-Bouldin Index: 0.3265
Number of clusters: 3, Davies-Bouldin Index: 0.2727
Number of clusters: 4, Davies-Bouldin Index: 0.8139
Number of clusters: 5, Davies-Bouldin Index: 0.6644
Number of clusters: 6, Davies-Bouldin Index: 0.5615
Number of clusters: 7, Davies-Bouldin Index: 0.9813
Number of clusters: 8, Davies-Bouldin Index: 0.9118
Number of clusters: 9, Davies-Bouldin Index: 1.0779
Number of clusters: 10, Davies-Bouldin Index: 1.2289
Number of clusters for minimum Davies-Bouldin Index: 3
```

From the analysis of the Davies-Bouldin Index (DBI) plotted against the number of clusters, we can determine that the optimal number of clusters is $k = 3$. The graph shows that the DBI decreases as we increase the cluster count from 2 to 3, indicating better clustering quality. However, after $k = 3$, the DBI starts to rise, suggesting that additional clusters do not enhance separation. Thus, the lowest DBI at $k = 3$ indicates well-separated and compact clusters, making it the best choice for our dataset.

4.2 Evaluating using Calinski-Harabasz Index

In this analysis, we explored the optimal number of clusters for our dataset using K-Means clustering and the Calinski-Harabasz Index (CHI). We evaluated cluster counts from 2 to 10, calculating the CHI for each configuration to identify the maximum value, indicating the best clustering quality. To visualize our findings, we plotted the CHI against the number of clusters, which aided in interpreting the relationship and selecting the optimal cluster number effectively.

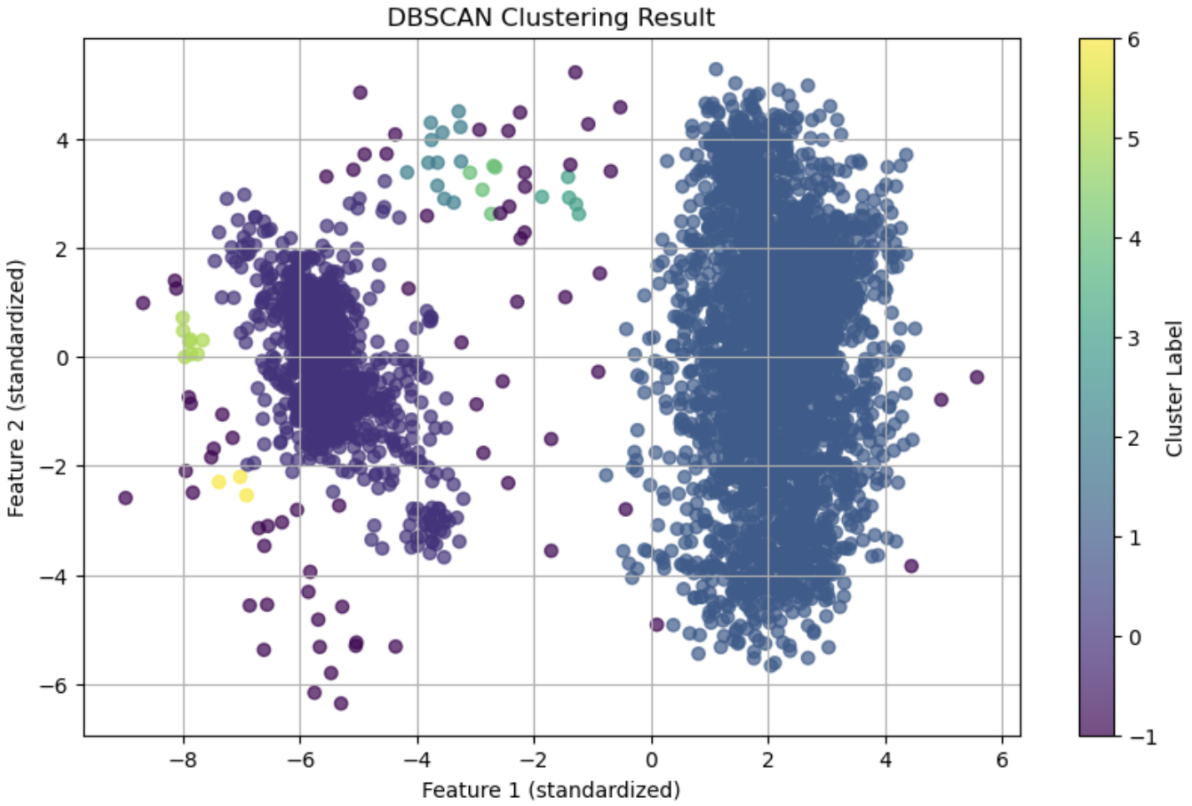



```
Number of clusters: 2, Calinski-Harabasz Index: 19091.8127
Number of clusters: 3, Calinski-Harabasz Index: 31711.0248
Number of clusters: 4, Calinski-Harabasz Index: 23399.0540
Number of clusters: 5, Calinski-Harabasz Index: 24143.9274
Number of clusters: 6, Calinski-Harabasz Index: 21735.9951
Number of clusters: 7, Calinski-Harabasz Index: 20901.1562
Number of clusters: 8, Calinski-Harabasz Index: 21268.7583
Number of clusters: 9, Calinski-Harabasz Index: 19327.9378
Number of clusters: 10, Calinski-Harabasz Index: 18046.5434
Number of clusters for maximum Calinski-Harabasz Index: 3
```

From the analysis of the Calinski-Harabasz Index (CHI) plotted against the number of clusters, we can determine that the optimal number of clusters is also $k = 3$. The graph shows that the CHI increases as we move from 2 to 3 clusters, indicating improved clustering quality. However, beyond $k = 3$, the CHI value starts to decline, suggesting that additional clusters do not significantly enhance the separation of data points. Thus, the maximum CHI at $k = 3$ indicates well-separated clusters with compactness, making it the ideal choice for our dataset.

5 Comparison with Other Clustering Algorithms

1. **DBSCAN Initialization**: We initialized the DBSCAN model with specific `eps` and `min_samples` parameters to define the clustering criteria.
2. **Clustering**: We applied the DBSCAN algorithm to our PCA-reduced dataset X_{pca} to identify cluster labels for each data point.
3. **Visualization**: We created a scatter plot to visualize the clustering results, where we colored the points according to their assigned cluster labels, along with a color bar for reference.
4. **Cluster and Noise Counts**: Finally, we calculated and printed the estimated number of clusters found, excluding noise points, and the total number of noise points identified in the dataset.



Comparison of K-Means and DBSCAN

K-Means Clustering

Strengths:

- **Simplicity and Speed:** K-Means is straightforward to implement and computationally efficient, making it suitable for large datasets.
- **Scalability:** It scales well with the number of samples and can handle high-dimensional data effectively.
- **Easy to Interpret:** The results can be easily interpreted as clusters centered around the mean of the data points.

Weaknesses:

- **Assumption of Spherical Clusters:** K-Means assumes clusters are spherical and equally sized, which may not be true for all datasets.
- **Sensitivity to Initialization:** The final clusters can be highly sensitive to the initial placement of centroids, potentially leading to suboptimal results if not carefully initialized.
- **Fixed Number of Clusters:** The user must specify the number of clusters (k) in advance, which can be challenging without prior knowledge of the data.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Strengths:

- **Ability to Find Arbitrarily Shaped Clusters:** DBSCAN can discover clusters of varying shapes and sizes, making it more flexible than K-Means.
- **Noise Handling:** It identifies and labels noise points, allowing it to separate outliers effectively.
- **No Need to Specify Cluster Count:** The algorithm does not require the number of clusters to be specified in advance, relying instead on density parameters.

Weaknesses:

- **Parameter Sensitivity:** The performance of DBSCAN is sensitive to the choice of parameters ϵ (neighborhood size) and min_samples, which can be difficult to determine.
- **Struggles with Varying Densities:** It may fail to identify clusters if they have significantly different densities, leading to poor clustering results in such scenarios.
- **Memory Usage:** For very large datasets, DBSCAN can be more memory-intensive compared to K-Means due to its reliance on neighborhood searches.

Summary

- **K-Means** is best for large datasets with spherical clusters but requires predefined k and can struggle with outliers.
- **DBSCAN** is more versatile for finding irregular clusters and handling noise but is sensitive to parameter selection and can be inefficient for high-density variations.