CS60050 Machine Learning Assignment 3

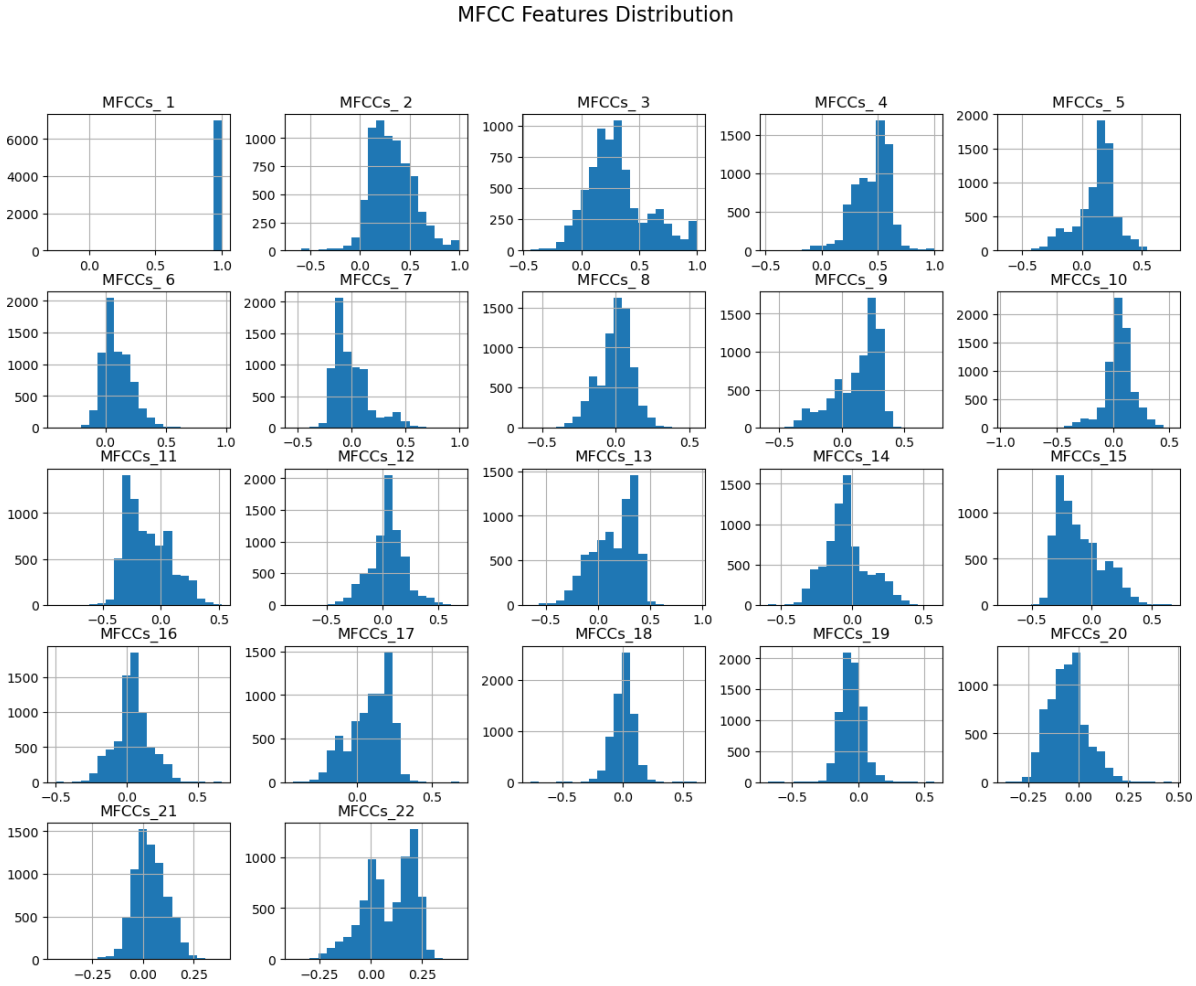
24CS60R77 – Ishan Rai

Part B (K-Means Clustering)

1. Data Preprocessing and Exploration

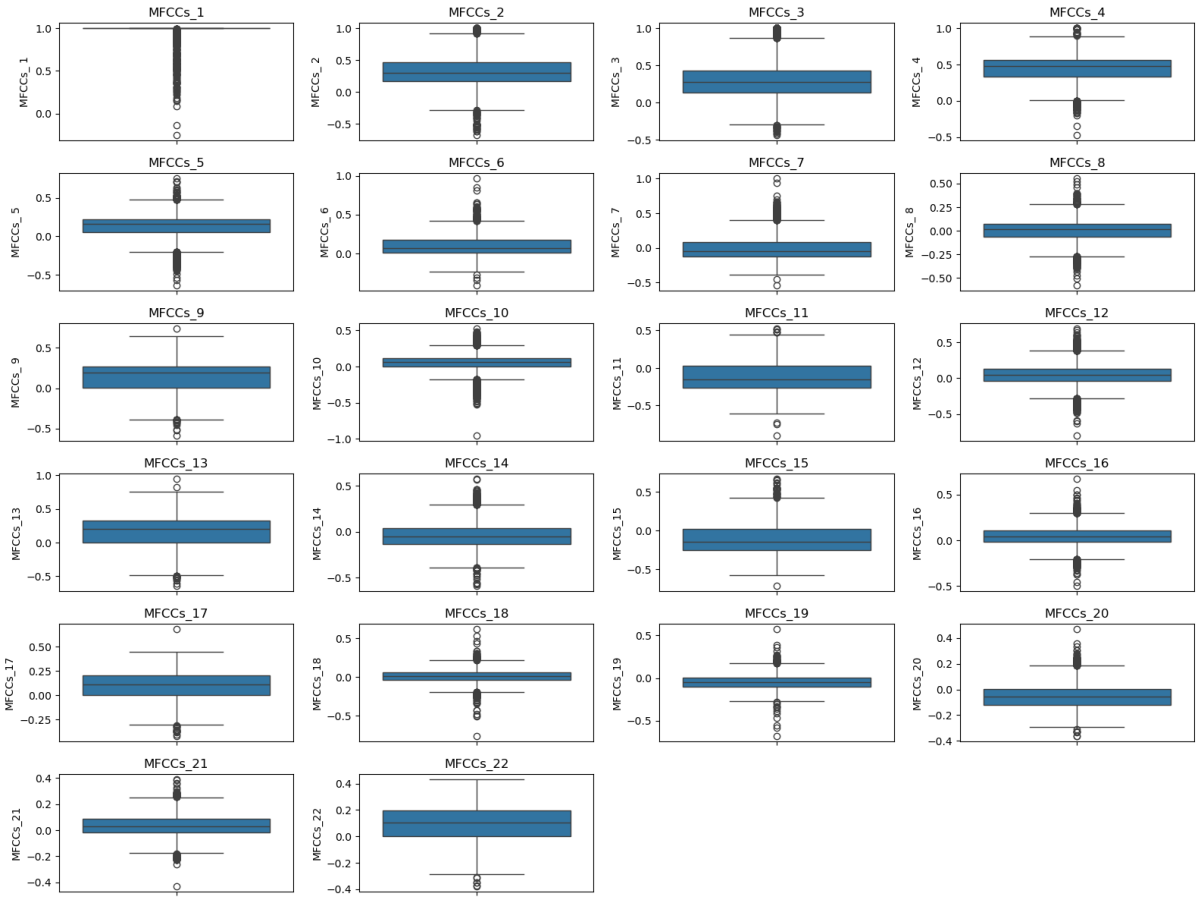
The dataset contains recordings of anuran (frog) calls. Each recording is segmented into syllables, and 22 MFCC features are extracted from each syllable. The dataset includes labels for family, genus, and species of each frog. The data is imbalanced, with some species having significantly more samples than others. The recordings were collected in various locations in Brazil and Argentina under real-world noise conditions.

Histogram:



This histogram displays the distribution of each MFCC feature in the dataset. The distributions appear to be roughly normal, with some features showing slight skewness. This visualization provides insights into the range and variability of the MFCC features, which can be helpful for understanding the data and feature engineering.

Box Plot:



This boxplot visualizes the distribution of each MFCC feature. The box represents the interquartile range (IQR), with the median marked by a line within the box. The whiskers extend to 1.5 times the IQR, and outliers are shown as individual points. This plot helps identify potential outliers, the spread of the data, and the presence of any skewness in the distributions of the MFCC features.

Data Standardization:

The data standardization process involves transforming the MFCC features to have a mean of 0 and a standard deviation of 1. This is achieved using a StandardScaler object from scikit-learn. By standardizing the features, we ensure that they are on a similar scale, making them comparable and improving the performance of machine learning models. This is particularly important for algorithms that are sensitive to feature scales, such as SVM and KNN.

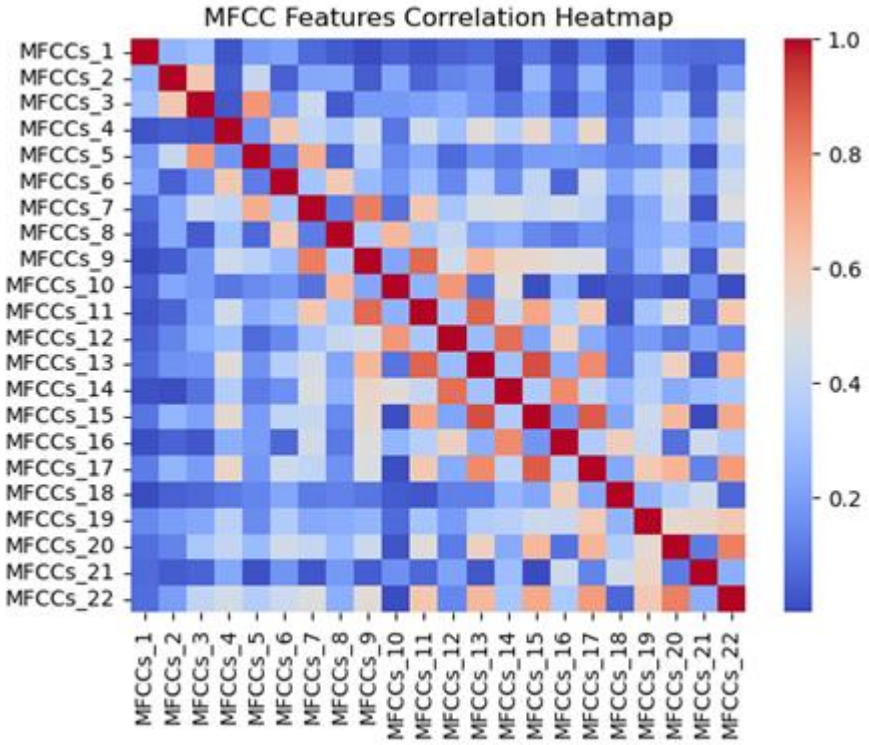
Feature selection:

Feature engineering involves creating new features from existing ones to improve model performance. In this case, polynomial features were generated by multiplying existing MFCC features with each other up to a specified degree (2 in this case). This transformation introduces non-linear relationships between features, potentially capturing complex patterns in the data. The resulting polynomial features are added to the original feature set, expanding the feature space and potentially improving the model's ability to learn complex decision boundaries.

Feature Correlation Analysis:

This heat map visualizes the correlation between different MFCC features. The darker shades of red indicate a stronger positive correlation, while the darker shades of blue indicate a stronger negative correlation. The diagonal line of perfect correlation is expected. Some features show moderate to strong correlations, suggesting potential redundancy. This

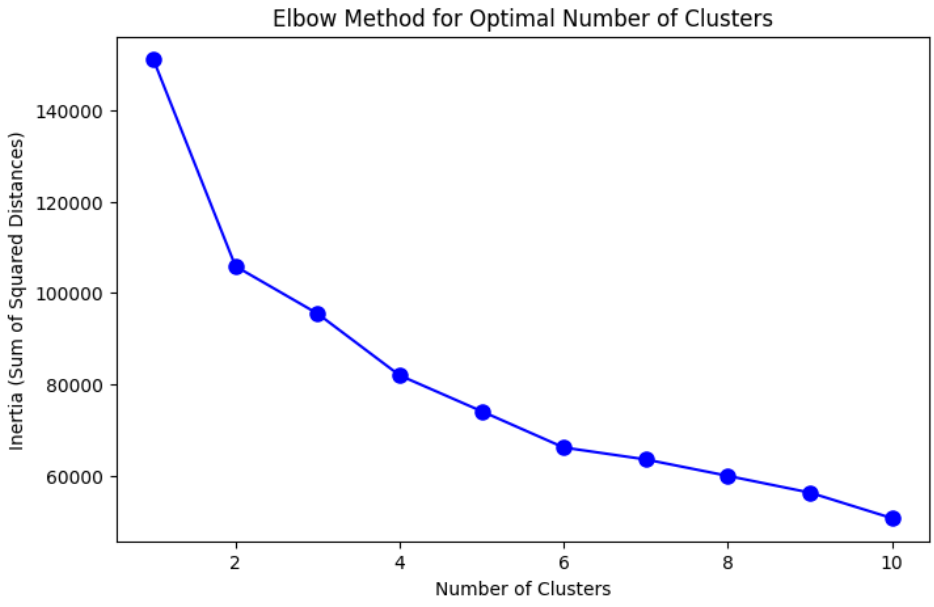
information can be used to select a subset of features or apply dimensionality reduction techniques.



2. K-Means Clustering

Elbow method:

The Elbow Method is a technique used to determine the optimal number of clusters for K-Means clustering. It involves plotting the within-cluster sum-of-squares (inertia) against the number of clusters. As the number of clusters increases, the inertia decreases. However, at some point, the decrease in inertia becomes less significant. The optimal number of clusters is typically identified at the "elbow point" where the rate of decrease in inertia starts to level off. In the given plot, the elbow point appears to be around 2 or 3 clusters.

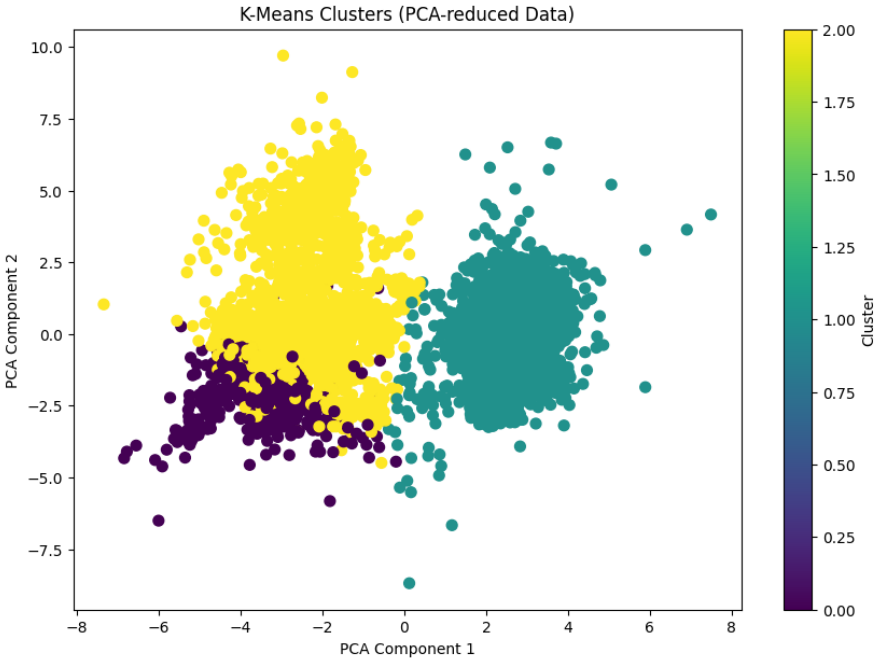


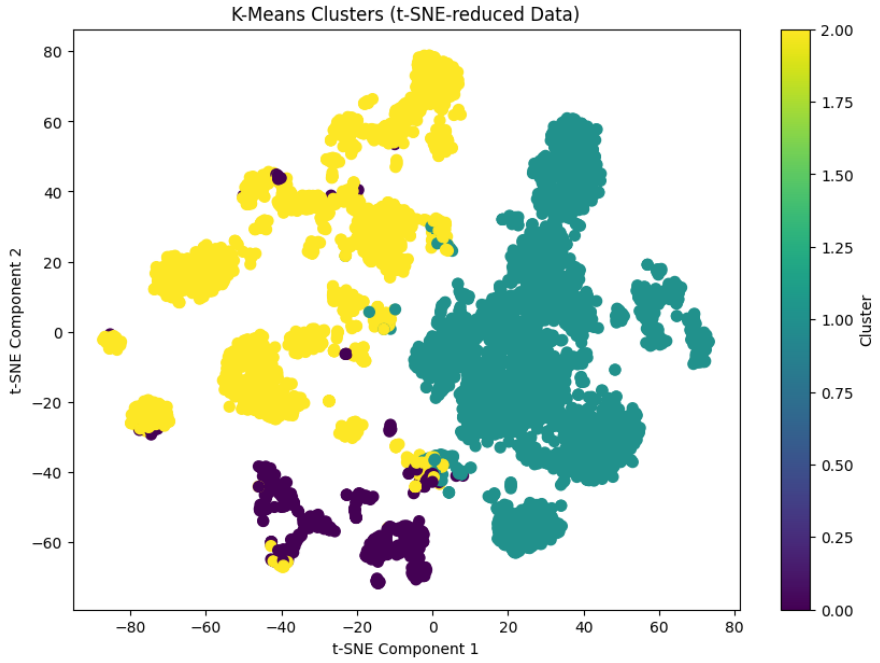
```
Silhouette Score for 3 clusters: 0.3274
```

Cluster Initialization Comparison (Random vs. k-means++):

```
Silhouette Score (Random Initialization): 0.3493
Silhouette Score (k-means++ Initialization): 0.3274
```

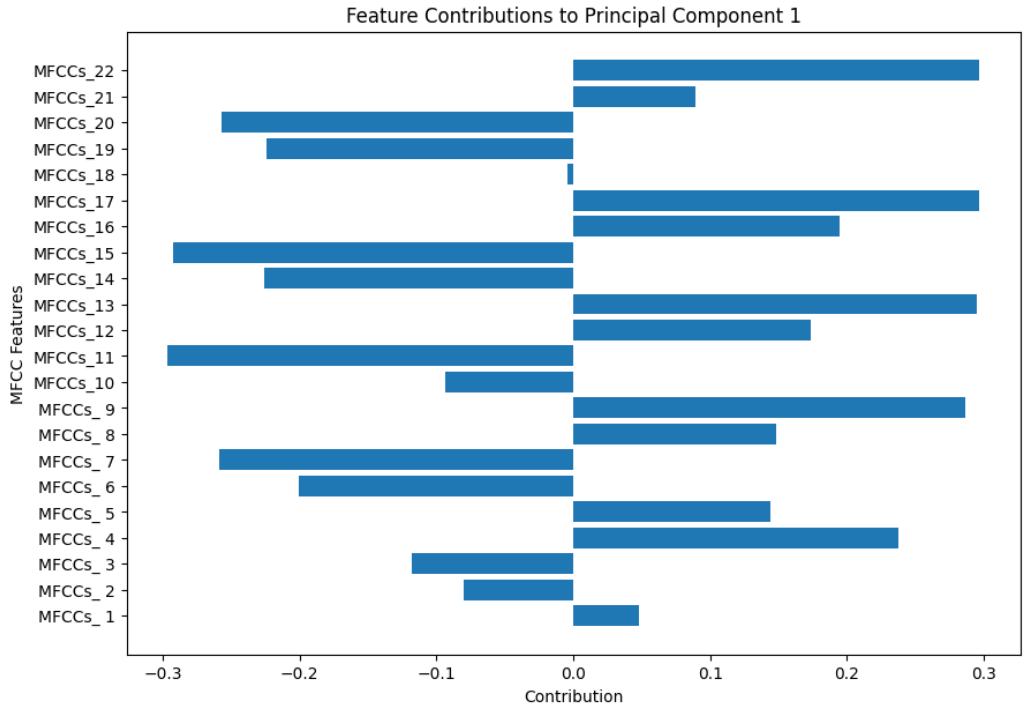
3. Cluster Visualization

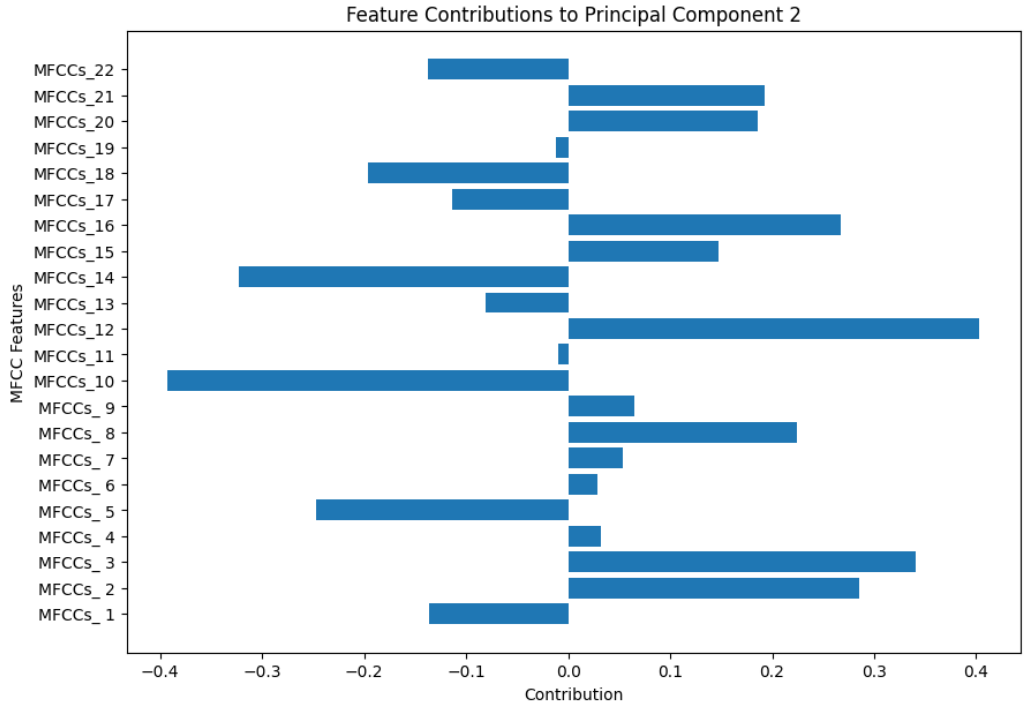




Top contributing features:

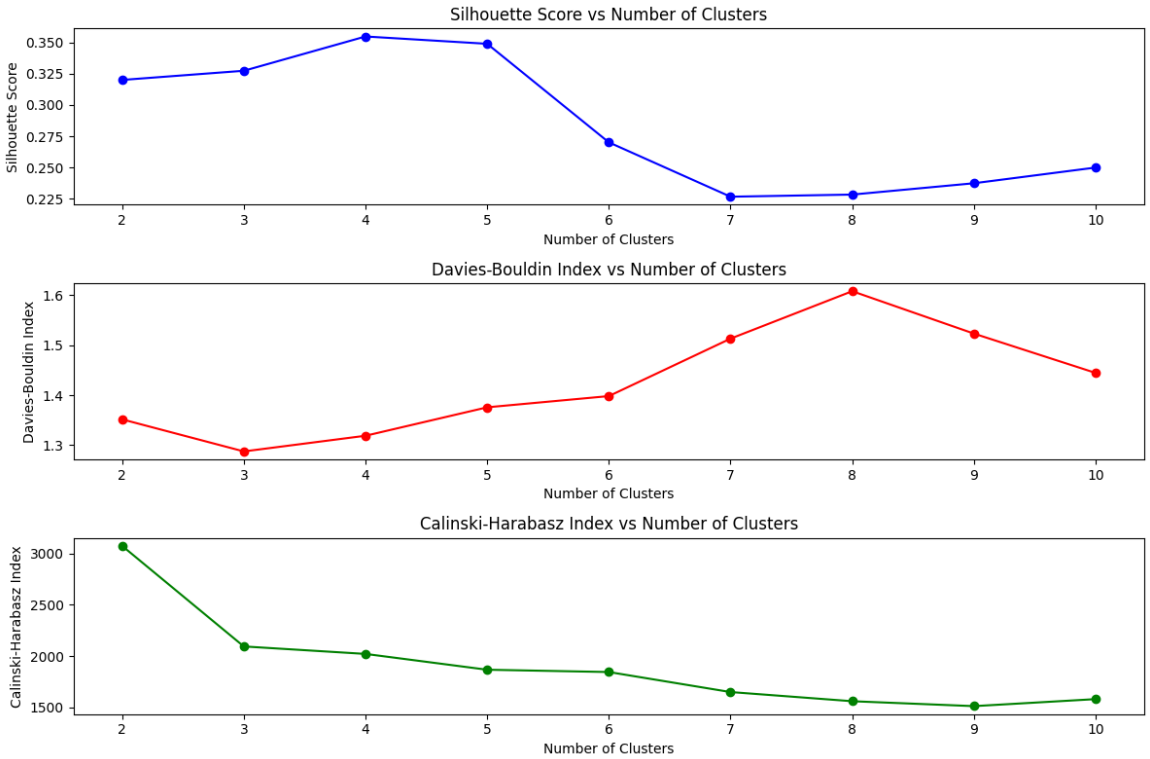
In the first principal component, the top contributing features are MFCCs_22, MFCCs_21, and MFCCs_20, indicating that these acoustic features are highly influential in distinguishing the clusters. For the second principal component, the most important features are MFCCs_12, MFCCs_11, and MFCCs_10. These features likely capture different acoustic characteristics that complement the information provided by the top features in the first principal component.





4. Cluster Evaluation Metrics

```
Davies-Bouldin Index: 1.2876
Calinski-Harabasz Index: 2094.2584
Optimal Number of Clusters: 3
Silhouette Score: 0.3274
```



5. Comparison with Other Clustering Algorithms

1. DBSCAN Clustering

Strengths:

- Density-based: DBSCAN is effective in identifying clusters of varying shapes and sizes, making it suitable for detecting arbitrary clusters and handling noise.

- Noise Handling: Points that don't belong to any cluster (outliers) are categorized as noise, which can be beneficial for datasets with many scattered or irrelevant points.

Weaknesses:

-Parameter Sensitivity: DBSCAN's performance heavily depends on the `eps` (neighborhood radius) and `min_samples` parameters, which may require fine-tuning.

- Cluster Shape Dependency: While DBSCAN handles non-linear clusters well, it struggles with complex patterns if clusters are not well-defined in density.

2. Agglomerative Hierarchical Clustering

Strengths:

- Hierarchical Structure: This method builds a hierarchy, which is helpful for exploring clusters at different levels of granularity.

- Doesn't Require a Predefined Cluster Count: Agglomerative clustering allows for flexible analysis by letting the user set the cutoff distance for clusters at any point.

Weaknesses:

- Computationally Expensive: Hierarchical clustering can be slower on large datasets, as it requires computing distances between all points multiple times.

- Difficulty with Noise: Unlike DBSCAN, it doesn't explicitly classify points as noise, which may lead to clusters being forced even with dissimilar points.

3. K-Means Clustering

Strengths:

- Efficient for Large Datasets: K-Means is relatively fast and scalable, making it suitable for datasets with many points.

- Clear Spherical Clusters: Works well with compact, spherical clusters and is easier to interpret.

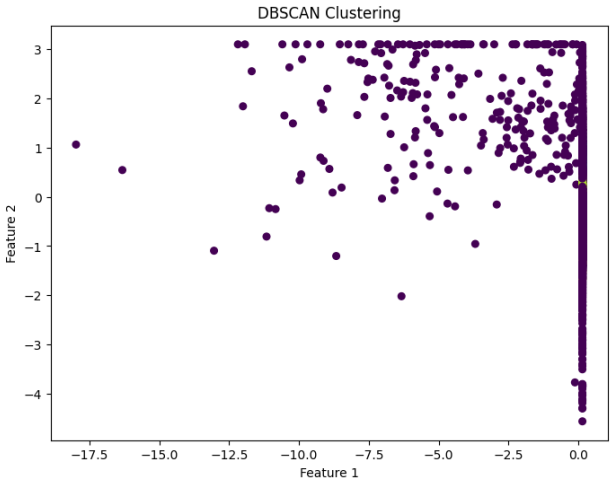
Weaknesses:

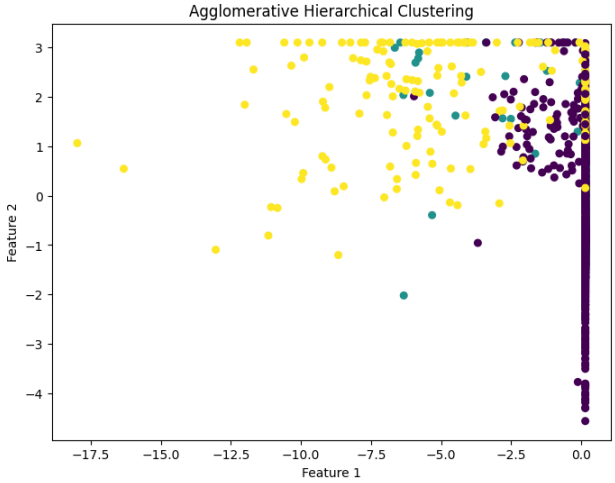
- Sensitivity to Initial Centroids: K-Means' final clustering results depend on the initial centroids, and it may converge to local optima if initialization isn't handled properly.

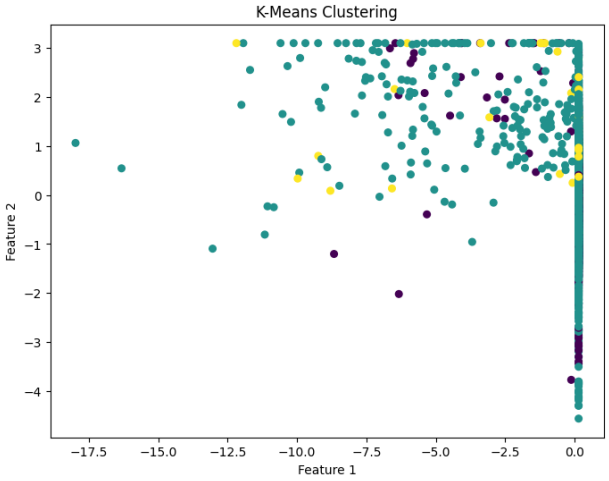
- Requires Predefined Cluster Count: You must specify the number of clusters, which can be challenging if the ideal count isn't known.

- Not Robust to Outliers: K-Means lacks a noise-handling mechanism, making it sensitive to outliers.

For this dataset, DBSCAN** appears to handle noise better and adapt to the irregular cluster density, while Agglomerative Hierarchical Clustering provides flexibility in viewing cluster hierarchies. K-Means, while effective for spherical clusters, struggles with the dataset's irregular density and shape.







6. Analysis and Report

```
K-Means Metrics:
Silhouette Score: 0.3274, Davies-Bouldin Index: 1.2876, Calinski-Harabasz
Index: 2094.2584

Agglomerative Hierarchical Clustering Metrics:
Silhouette Score: 0.3390, Davies-Bouldin Index: 1.4822, Calinski-Harabasz
Index: 2149.8463

DBSCAN Metrics:
Silhouette Score: -0.2293, Davies-Bouldin Index: 1.7034, Calinski-Harabasz
Index: 16.9242
```
