Frog Call Acoustic Analysis

A Clustering Approach

Introduction

- **Background**: Anurans, commonly known as frogs and toads, are known for their distinct vocalizations, often used for communication and mating. These vocalizations, or calls, have specific acoustic features that can potentially distinguish between different families.
- **Importance**: Understanding these calls can not only aid in the identification of different anuran species but could also contribute to the studies of behavioral patterns, mating rituals, and potentially, the impact of environmental changes on these species.

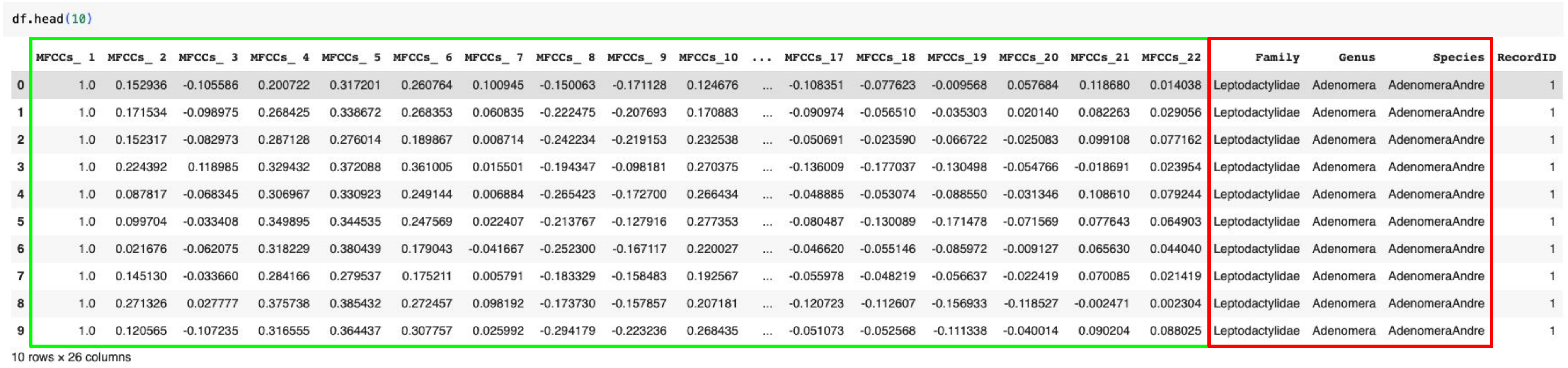
Problem Statement

- **Objective**: Our goal is to leverage the power of machine learning, specifically unsupervised algorithms, to analyze and classify these acoustic features, thereby distinguishing different anuran families based on their calls.
- **Approach**: I intend to use different clustering algorithms, including K-Means, DBSCAN, Affinity Propagation, OPTICS, Mean-shift, and Agglomerative Clustering. These algorithms will enable us to analyze the anuran call data and uncover any intrinsic groupings.
- **Evaluation**: In order to compare the performance of the different clustering algorithms, we will use the Adjusted Rand Index (ARI). The ARI is a measure that quantifies the similarity between the true labels and the labels predicted by the clustering algorithm.
- **Why ARI?**: Unlike other metrics, the ARI accounts for the fact that random label assignments will result in some amount of "agreement" and corrects for this, providing a more accurate measure of how well the clustering algorithm performs relative to a random assignment.

Dataset

- Available on the UC Irvine Machine Learning Repository
- Name: Anuran Calls (MFCCs)
- Creator: Colonna et al.
- DOI: 10.24432/C5CC9H
- Link: https://archive.ics.uci.edu/dataset/406/anuran+calls+mfccs

Data Overview



```
df.head(10)
```

	MFCCs_ 1	MFCCs_ 2	MFCCs_ 3	MFCCs_ 4	MFCCs_ 5	MFCCs_ 6	MFCCs_ 7	MFCCs_ 8	MFCCs_ 9	MFCCs_10	...	MFCCs_17	MFCCs_18	MFCCs_19	MFCCs_20	MFCCs_21	MFCCs_22	Family	Genus	Species	RecordID
0	1.0	0.152936	-0.105586	0.200722	0.317201	0.260764	0.100945	-0.150063	-0.171128	0.124676	...	-0.108351	-0.077623	-0.009568	0.057684	0.118680	0.014038	Leptodactylidae	Adenomera	AdenomeraAndre	1
1	1.0	0.171534	-0.098975	0.268425	0.338672	0.268353	0.060835	-0.222475	-0.207693	0.170883	...	-0.090974	-0.056510	-0.035303	0.020140	0.082263	0.029056	Leptodactylidae	Adenomera	AdenomeraAndre	1
2	1.0	0.152317	-0.082973	0.287128	0.276014	0.189867	0.008714	-0.242234	-0.219153	0.232538	...	-0.050691	-0.023590	-0.066722	-0.025083	0.099108	0.077162	Leptodactylidae	Adenomera	AdenomeraAndre	1
3	1.0	0.224392	0.118985	0.329432	0.372088	0.361005	0.015501	-0.194347	-0.098181	0.270375	...	-0.136009	-0.177037	-0.130498	-0.054766	-0.018691	0.023954	Leptodactylidae	Adenomera	AdenomeraAndre	1
4	1.0	0.087817	-0.068345	0.306967	0.330923	0.249144	0.006884	-0.265423	-0.172700	0.266434	...	-0.048885	-0.053074	-0.088550	-0.031346	0.108610	0.079244	Leptodactylidae	Adenomera	AdenomeraAndre	1
5	1.0	0.099704	-0.033408	0.349895	0.344535	0.247569	0.022407	-0.213767	-0.127916	0.277353	...	-0.080487	-0.130089	-0.171478	-0.071569	0.077643	0.064903	Leptodactylidae	Adenomera	AdenomeraAndre	1
6	1.0	0.021676	-0.062075	0.318229	0.380439	0.179043	-0.041667	-0.252300	-0.167117	0.220027	...	-0.046620	-0.055146	-0.085972	-0.009127	0.065630	0.044040	Leptodactylidae	Adenomera	AdenomeraAndre	1
7	1.0	0.145130	-0.033660	0.284166	0.279537	0.175211	0.005791	-0.183329	-0.158483	0.192567	...	-0.055978	-0.048219	-0.056637	-0.022419	0.070085	0.021419	Leptodactylidae	Adenomera	AdenomeraAndre	1
8	1.0	0.271326	0.027777	0.375738	0.385432	0.272457	0.098192	-0.173730	-0.157857	0.207181	...	-0.120723	-0.112607	-0.156933	-0.118527	-0.002471	0.002304	Leptodactylidae	Adenomera	AdenomeraAndre	1
9	1.0	0.120565	-0.107235	0.316555	0.364437	0.307757	0.025992	-0.294179	-0.223236	0.268435	...	-0.051073	-0.052568	-0.111338	-0.040014	0.090204	0.088025	Leptodactylidae	Adenomera	AdenomeraAndre	1

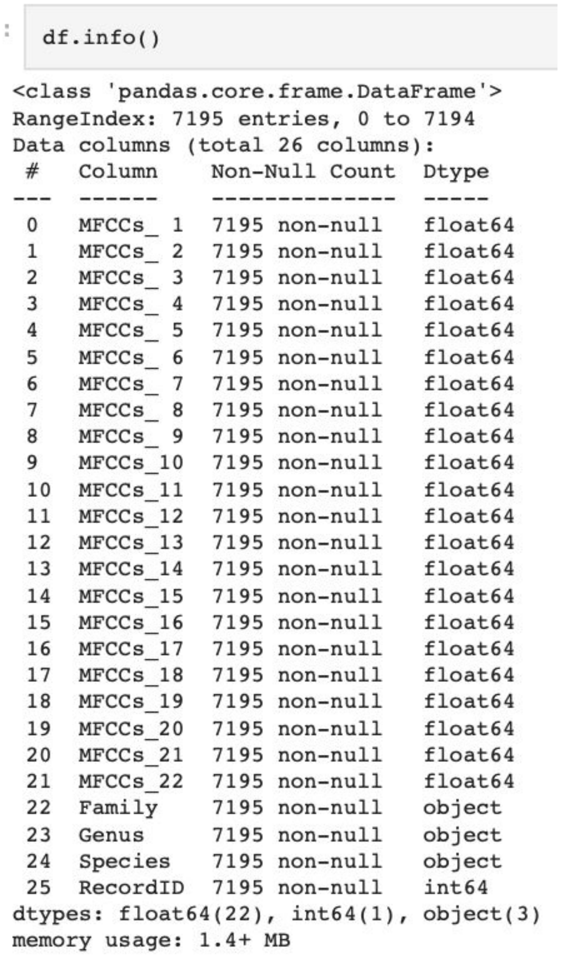
10 rows × 26 columns

Features

Labels

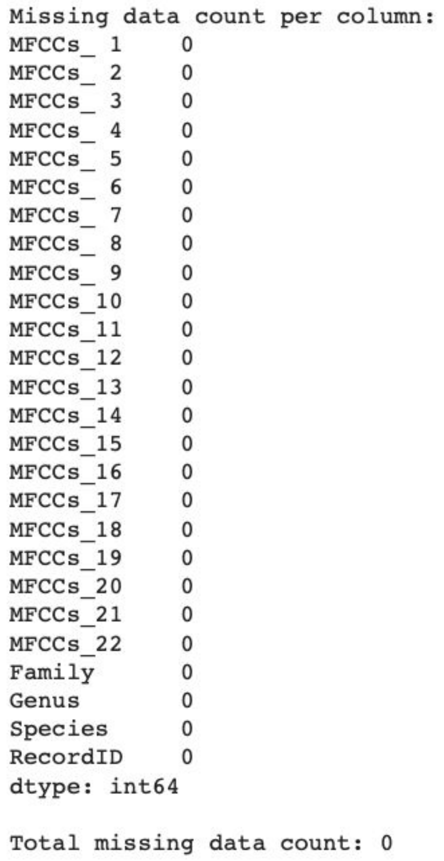
Data Overview

# of instances: 7195
# of attributes: 22



```
df.info()
```

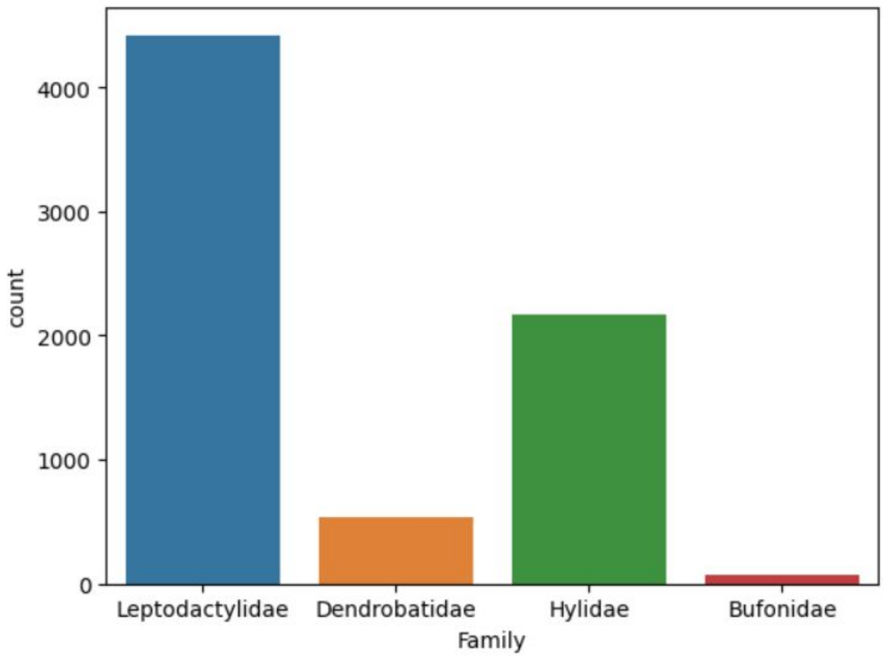
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7195 entries, 0 to 7194
Data columns (total 26 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   MFCCs_ 1   7195 non-null   float64
 1   MFCCs_ 2   7195 non-null   float64
 2   MFCCs_ 3   7195 non-null   float64
 3   MFCCs_ 4   7195 non-null   float64
 4   MFCCs_ 5   7195 non-null   float64
 5   MFCCs_ 6   7195 non-null   float64
 6   MFCCs_ 7   7195 non-null   float64
 7   MFCCs_ 8   7195 non-null   float64
 8   MFCCs_ 9   7195 non-null   float64
 9   MFCCs_10   7195 non-null   float64
 10  MFCCs_11   7195 non-null   float64
 11  MFCCs_12   7195 non-null   float64
 12  MFCCs_13   7195 non-null   float64
 13  MFCCs_14   7195 non-null   float64
 14  MFCCs_15   7195 non-null   float64
 15  MFCCs_16   7195 non-null   float64
 16  MFCCs_17   7195 non-null   float64
 17  MFCCs_18   7195 non-null   float64
 18  MFCCs_19   7195 non-null   float64
 19  MFCCs_20   7195 non-null   float64
 20  MFCCs_21   7195 non-null   float64
 21  MFCCs_22   7195 non-null   float64
 22  Family     7195 non-null   object
 23  Genus      7195 non-null   object
 24  Species    7195 non-null   object
 25  RecordID   7195 non-null   int64
dtypes: float64(22), int64(1), object(3)
memory usage: 1.4+ MB
```



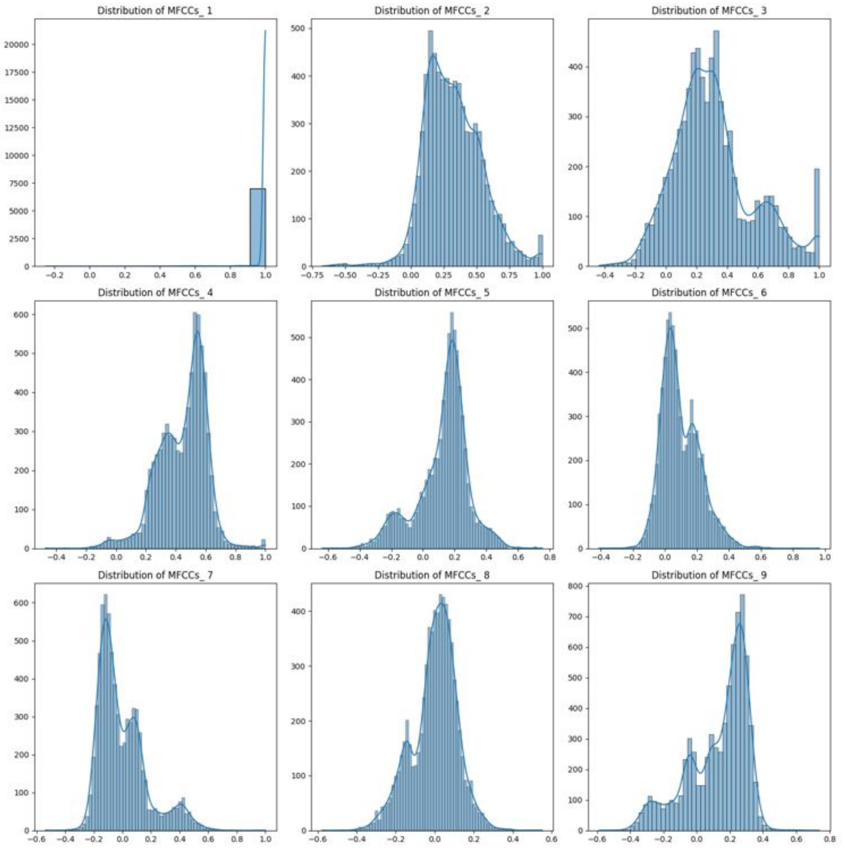
```
Missing data count per column:
MFCCs_ 1     0
MFCCs_ 2     0
MFCCs_ 3     0
MFCCs_ 4     0
MFCCs_ 5     0
MFCCs_ 6     0
MFCCs_ 7     0
MFCCs_ 8     0
MFCCs_ 9     0
MFCCs_10     0
MFCCs_11     0
MFCCs_12     0
MFCCs_13     0
MFCCs_14     0
MFCCs_15     0
MFCCs_16     0
MFCCs_17     0
MFCCs_18     0
MFCCs_19     0
MFCCs_20     0
MFCCs_21     0
MFCCs_22     0
Family       0
Genus        0
Species      0
RecordID     0
dtype: int64

Total missing data count: 0
```

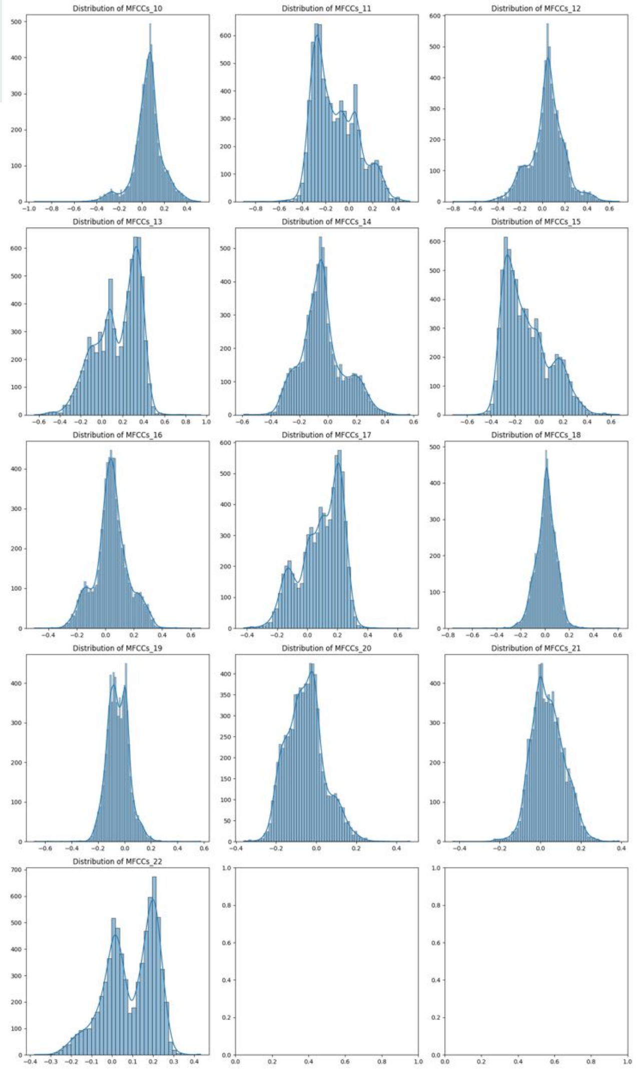
Data Visualization

```
sns.countplot(data=df, x="Family")
plt.show()
```

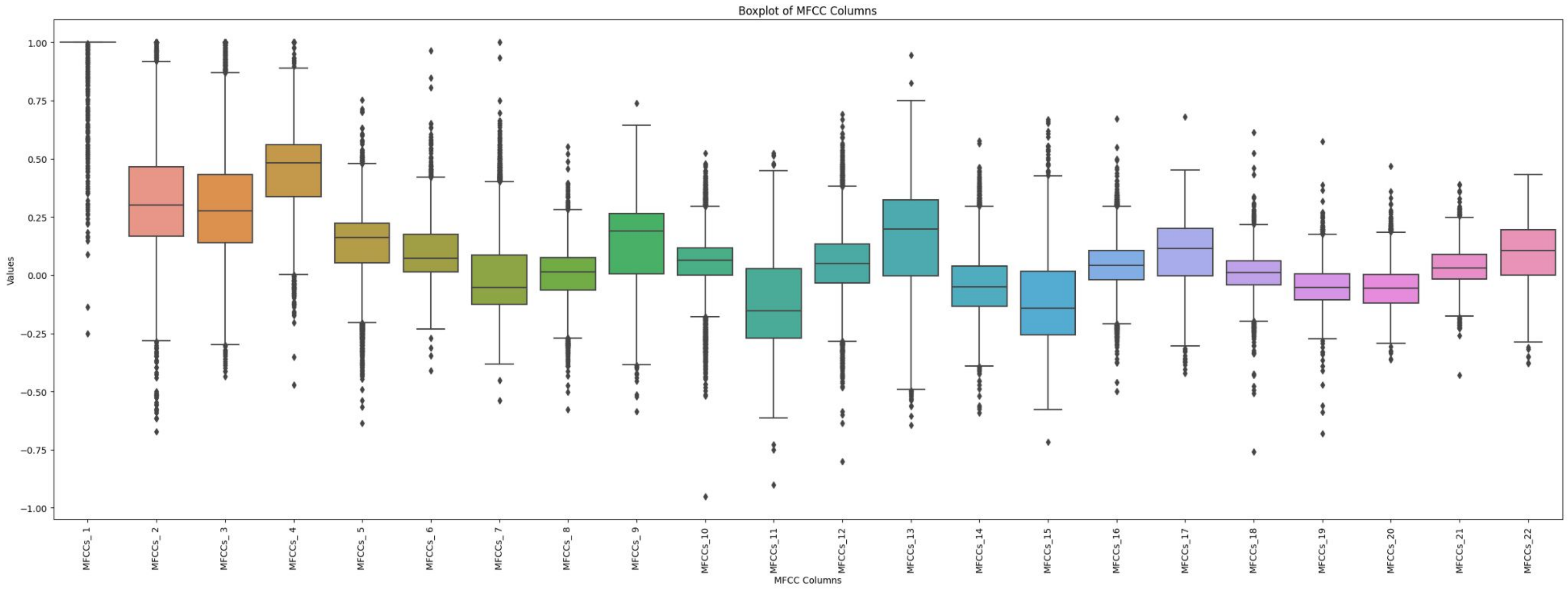


Data Visualization

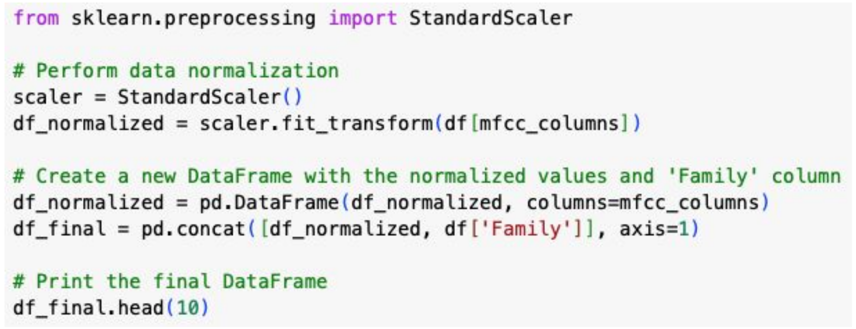




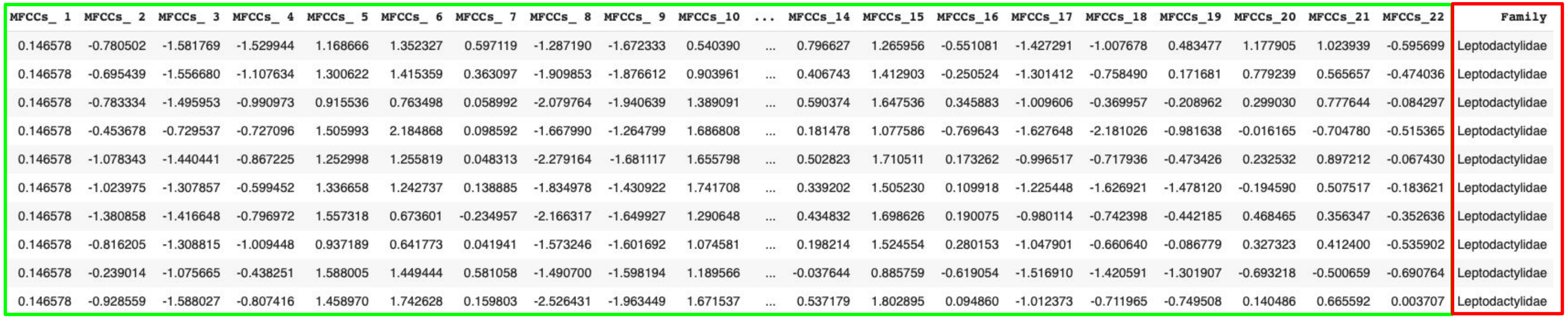
Data Visualization



Data Transformation



```python
from sklearn.preprocessing import StandardScaler

# Perform data normalization
scaler = StandardScaler()
df_normalized = scaler.fit_transform(df[mfcc_columns])

# Create a new DataFrame with the normalized values and 'Family' column
df_normalized = pd.DataFrame(df_normalized, columns=mfcc_columns)
df_final = pd.concat([df_normalized, df['Family']], axis=1)

# Print the final DataFrame
df_final.head(10)
```



MFCCs_ 1	MFCCs_ 2	MFCCs_ 3	MFCCs_ 4	MFCCs_ 5	MFCCs_ 6	MFCCs_ 7	MFCCs_ 8	MFCCs_ 9	MFCCs_10	...	MFCCs_14	MFCCs_15	MFCCs_16	MFCCs_17	MFCCs_18	MFCCs_19	MFCCs_20	MFCCs_21	MFCCs_22	Family
0.146578	-0.780502	-1.581769	-1.529944	1.168666	1.352327	0.597119	-1.287190	-1.672333	0.540390	...	0.796627	1.265956	-0.551081	-1.427291	-1.007678	0.483477	1.177905	1.023939	-0.595699	Leptodactylidae
0.146578	-0.695439	-1.556680	-1.107634	1.300622	1.415359	0.363097	-1.909853	-1.876612	0.903961	...	0.406743	1.412903	-0.250524	-1.301412	-0.758490	0.171681	0.779239	0.565657	-0.474036	Leptodactylidae
0.146578	-0.783334	-1.495953	-0.990973	0.915536	0.763498	0.058992	-2.079764	-1.940639	1.389091	...	0.590374	1.647536	0.345883	-1.009606	-0.369957	-0.208962	0.299030	0.777644	-0.084297	Leptodactylidae
0.146578	-0.453678	-0.729537	-0.727096	1.505993	2.184868	0.098592	-1.667990	-1.264799	1.686808	...	0.181478	1.077586	-0.769643	-1.627648	-2.181026	-0.981638	-0.016165	-0.704780	-0.515365	Leptodactylidae
0.146578	-1.078343	-1.440441	-0.867225	1.252998	1.255819	0.048313	-2.279164	-1.681117	1.655798	...	0.502823	1.710511	0.173262	-0.996517	-0.717936	-0.473426	0.232532	0.897212	-0.067430	Leptodactylidae
0.146578	-1.023975	-1.307857	-0.599452	1.336658	1.242737	0.138885	-1.834978	-1.430922	1.741708	...	0.339202	1.505230	0.109918	-1.225448	-1.626921	-1.478120	-0.194590	0.507517	-0.183621	Leptodactylidae
0.146578	-1.380858	-1.416648	-0.796972	1.557318	0.673601	-0.234957	-2.166317	-1.649927	1.290648	...	0.434832	1.698626	0.190075	-0.980114	-0.742398	-0.442185	0.468465	0.356347	-0.352636	Leptodactylidae
0.146578	-0.816205	-1.308815	-1.009448	0.937189	0.641773	0.041941	-1.573246	-1.601692	1.074581	...	0.198214	1.524554	0.280153	-1.047901	-0.660640	-0.086779	0.327323	0.412400	-0.535902	Leptodactylidae
0.146578	-0.239014	-1.075665	-0.438251	1.588005	1.449444	0.581058	-1.490700	-1.598194	1.189566	...	-0.037644	0.885759	-0.619054	-1.516910	-1.420591	-1.301907	-0.693218	-0.500659	-0.690764	Leptodactylidae
0.146578	-0.928559	-1.588027	-0.807416	1.458970	1.742628	0.159803	-2.526431	-1.963449	1.671537	...	0.537179	1.802895	0.094860	-1.012373	-0.711965	-0.749508	0.140486	0.665592	0.003707	Leptodactylidae

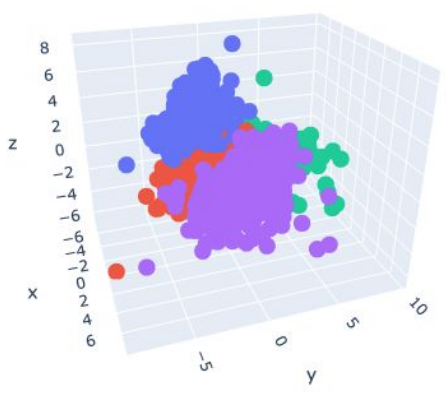
Label

Features

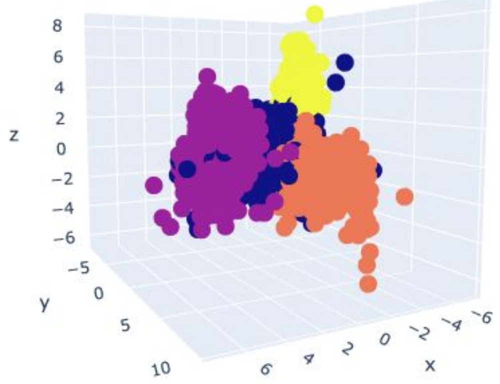
Proposed Procedures

1. GridSearchCV for hyperparameters tuning (using the best Silhouette Score)
2. Perform the proposed clustering model
3. Perform PCA for dimensionality reduction to n_components = 3 and visualize the clusters.
4. Calculate the Adjusted Rand Index based on the assigned clusters and the ground truth classes “Family”
5. Models comparison using the Adjusted Rand Index

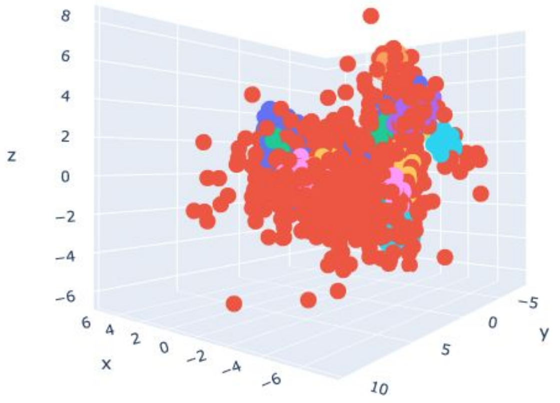
Some Cluster Visualizations



K-Means

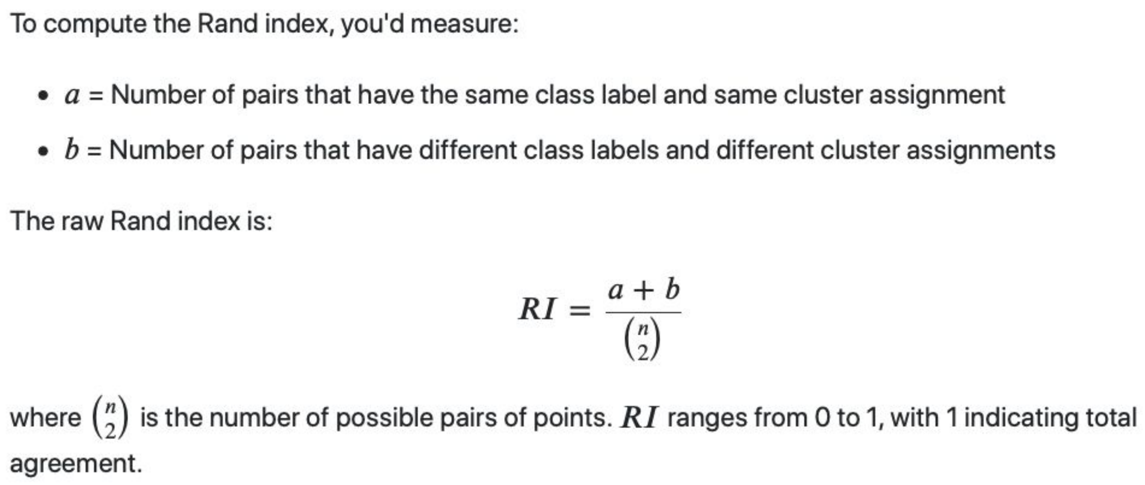


Agglomerative Clustering



DBSCAN

Adjusted Rand Index



To compute the Rand index, you'd measure:

- a = Number of pairs that have the same class label and same cluster assignment
- b = Number of pairs that have different class labels and different cluster assignments

The raw Rand index is:

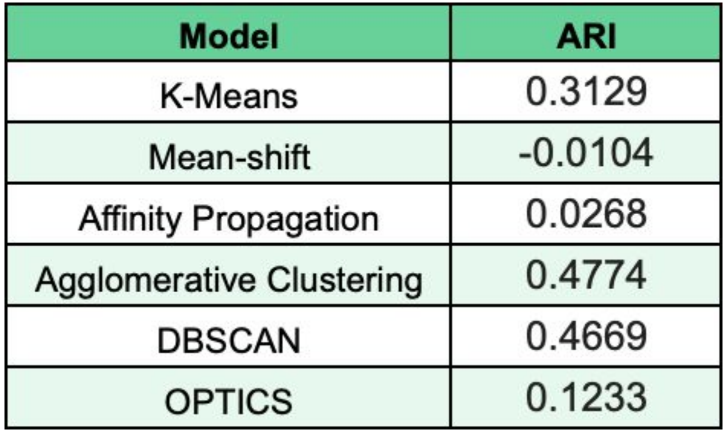
$$RI = \frac{a + b}{\binom{n}{2}}$$

where $\binom{n}{2}$ is the number of possible pairs of points. RI ranges from 0 to 1, with 1 indicating total agreement.

- **The Adjusted Rand Index** also accounts for chance, thus ensured to have a value close to 0.0 for random labeling independently of the number of clusters and samples and exactly 1.0 when the clusterings are identical.
- The adjusted Rand index is bounded below by -0.5 for especially discordant clusterings.

Reference: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html

Results



Model	ARI
K-Means	0.3129
Mean-shift	-0.0104
Affinity Propagation	0.0268
Agglomerative Clustering	0.4774
DBSCAN	0.4669
OPTICS	0.1233

- All results are below 0.50 which shows poor coordination between the assigned clusters and the ground truth labels.
- Two models show potential for further improvement: Agglomerative Clustering (0.4774) and DBSCAN (0.4669)
- Consider supervised algorithms for this project