DS 203

Project PresentatioN

TUSHAR SINGHA ROY 23B1233
AYUSH PATEL 23B0037
RISHIK YADAV 23B2103
VEDANT BHARDWAJ 23B0068

Executive Summary

1. Problem Overview
2. Data Overview and Processing
3. EDA on the data
4. Comparing Elbow Plots
5. PCA
6. Clustering
7. Clustering based on MFCC Coefficients
8. Train data preparation for MLP
9. Model Training
10. Predictions
11. Final results

PROBLEM STATEMENT

The main obejective of the problem statement is to utilize the MFCC coefficients which are extracted from audio files to classify 115 songs provided into different categories like songs of specific type or sung by a particular artists

Dataset overview

- Total 115 csv files containing the MFCC data of audio files
- Each csv contains 20 MFCC coefficients per segment, sampled at a rate of 44,100 Hz

Different categories of audio files

- Indian National Anthem renditions
- Marathi Bhavgeet
- Marathi Lavni Songs
- Hindi songs by Asha Bhosle
- Hindi songs by Kishore Kumar
- English songs by Michael Jackson

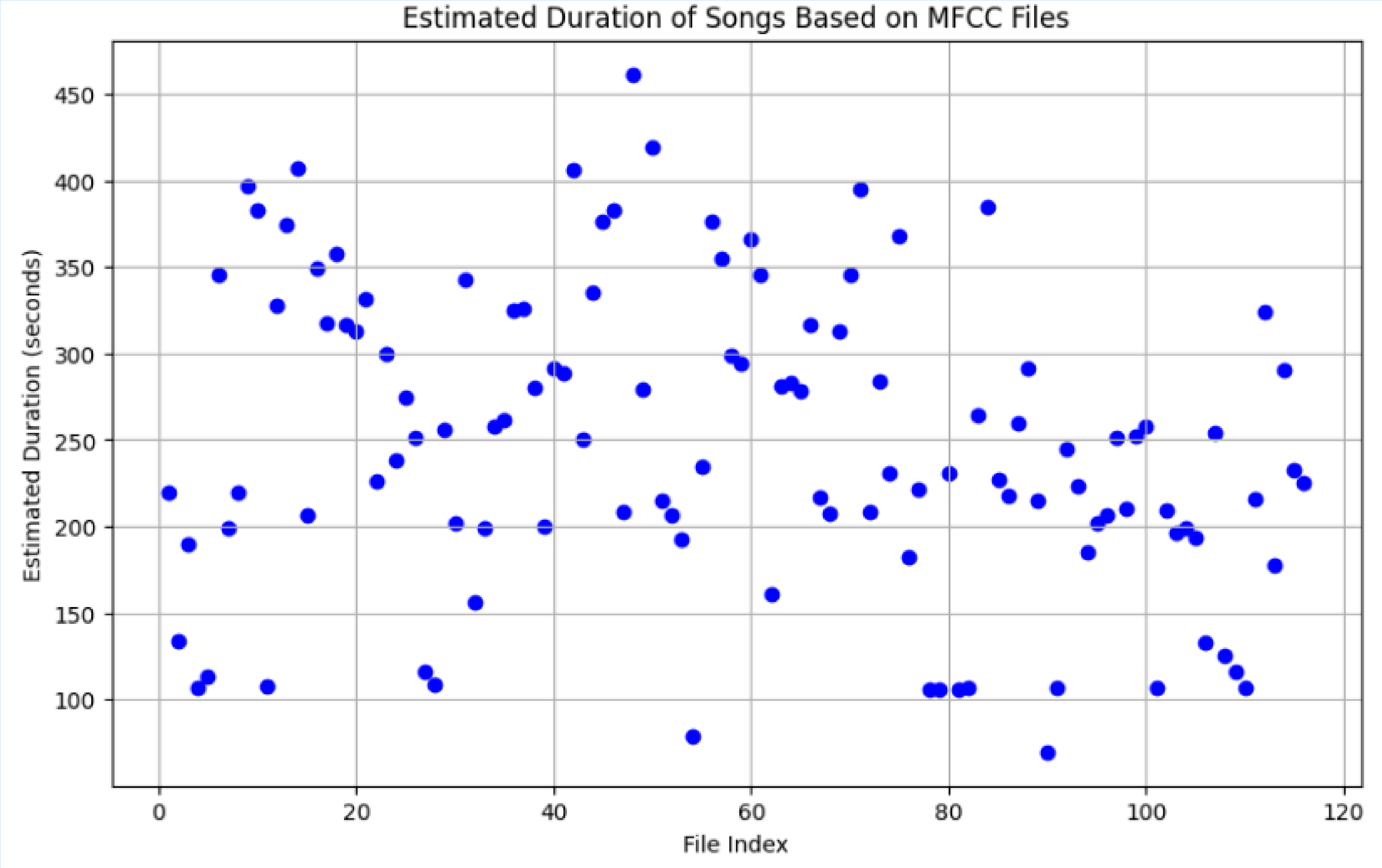
PRIMARY OBJECTIVE

1. Classify the 115 songs into the groups mentioned earlier
2. Identify 3 files that contain the data of
   - The National anthem
   - Songs by Asha Bhosal
   - Songs by Kishore Kumar
   - Songs by Michael Jackson

DATA OVERVIEW AND PROCESSING

The data has 20 rows for 20 different MFCC coefficients and these coefficients are calculated 86 times every seconds

This leads to approximately 10,320 columns for a 2min audio file.
This creates very large dataset showcasing that it is not feasible to work with such a large dataset efficiently and effecitively



The above Scatter plot shows the estimated length of the audio files based on the MFCC files provided

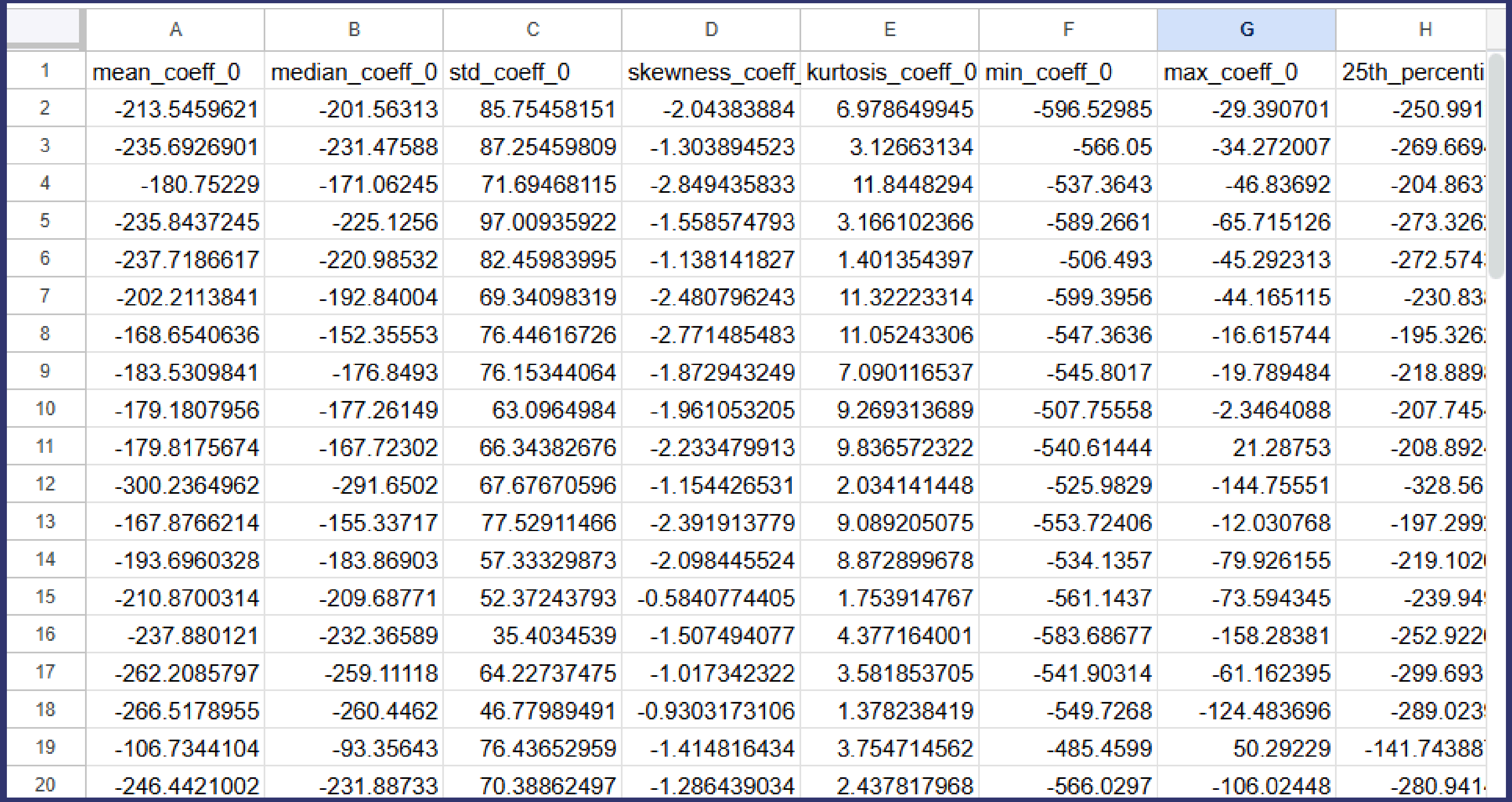
This shows us that out 115 songs song of them extend over even 5 mins

Data Processing

To handle this large dataset efficiently, it's crucial to optimize the processing for both time and resource usage, making the data more manageable and practical

To do this we calculate some key statistics for the different audio files for each of the Coefficients like

- Mean
- Median
- Standard deviation
- Skewness
- Kurtosis
- 25th Percentile
- 75th Percentile



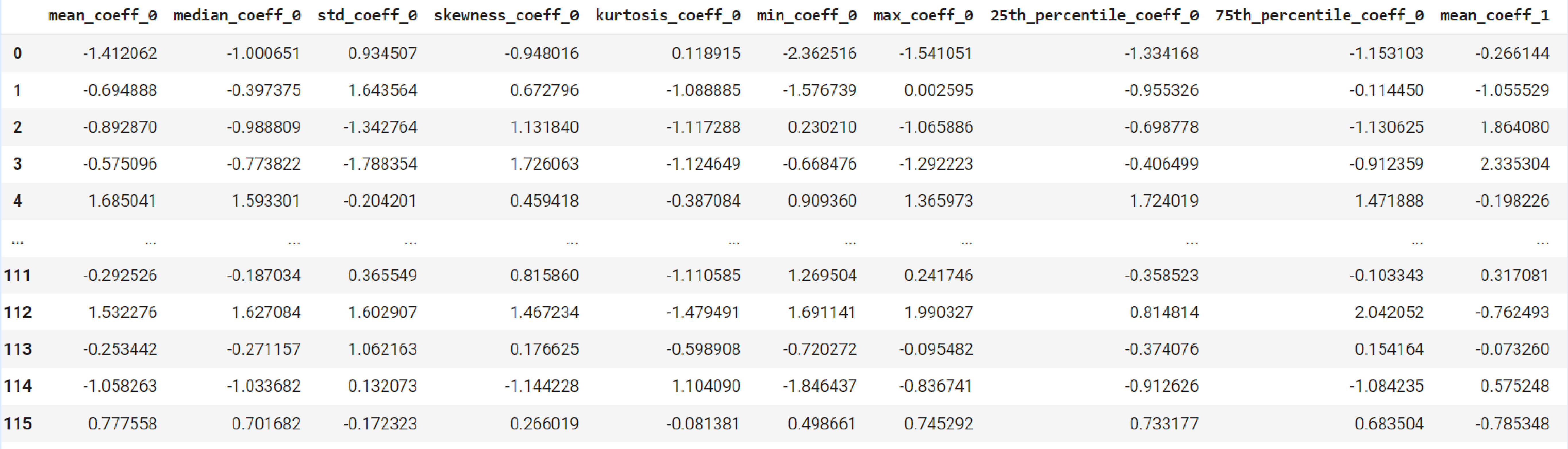
	A	B	C	D	E	F	G	H
1	mean_coeff_0	median_coeff_0	std_coeff_0	skewness_coeff_	kurtosis_coeff_0	min_coeff_0	max_coeff_0	25th_percenti
2	-213.5459621	-201.56313	85.75458151	-2.04383884	6.978649945	-596.52985	-29.390701	-250.991
3	-235.6926901	-231.47588	87.25459809	-1.303894523	3.12663134	-566.05	-34.272007	-269.669
4	-180.75229	-171.06245	71.69468115	-2.849435833	11.8448294	-537.3643	-46.83692	-204.863
5	-235.8437245	-225.1256	97.00935922	-1.558574793	3.166102366	-589.2661	-65.715126	-273.326
6	-237.7186617	-220.98532	82.45983995	-1.138141827	1.401354397	-506.493	-45.292313	-272.574
7	-202.2113841	-192.84004	69.34098319	-2.480796243	11.32223314	-599.3956	-44.165115	-230.83
8	-168.6540636	-152.35553	76.44616726	-2.771485483	11.05243306	-547.3636	-16.615744	-195.326
9	-183.5309841	-176.8493	76.15344064	-1.872943249	7.090116537	-545.8017	-19.789484	-218.889
10	-179.1807956	-177.26149	63.0964984	-1.961053205	9.269313689	-507.75558	-2.3464088	-207.745
11	-179.8175674	-167.72302	66.34382676	-2.233479913	9.836572322	-540.61444	21.28753	-208.892
12	-300.2364962	-291.6502	67.67670596	-1.154426531	2.034141448	-525.9829	-144.75551	-328.56
13	-167.8766214	-155.33717	77.52911466	-2.391913779	9.089205075	-553.72406	-12.030768	-197.299
14	-193.6960328	-183.86903	57.33329873	-2.098445524	8.872899678	-534.1357	-79.926155	-219.102
15	-210.8700314	-209.68771	52.37243793	-0.5840774405	1.753914767	-561.1437	-73.594345	-239.94
16	-237.880121	-232.36589	35.4034539	-1.507494077	4.377164001	-583.68677	-158.28381	-252.922
17	-262.2085797	-259.11118	64.22737475	-1.017342322	3.581853705	-541.90314	-61.162395	-299.693
18	-266.5178955	-260.4462	46.77989491	-0.9303173106	1.378238419	-549.7268	-124.483696	-289.023
19	-106.7344104	-93.35643	76.43652959	-1.414816434	3.754714562	-485.4599	50.29229	-141.74388
20	-246.4421002	-231.88733	70.38862497	-1.286439034	2.437817968	-566.0297	-106.02448	-280.941

The resulted data was a pandas DataFrame which contained the above statistics for all the coefficients which leaded to 180 columns and 115 Rows of data

ITERATION 1 - CLUSTERING

EXPLORATORY DATA ANALYSIS ON MFCC COEFFICIENTS

- After calculating the moments of the distribution across the time series for each coefficient, we do StandardScaler on it, making sure that each moment contributes equally.

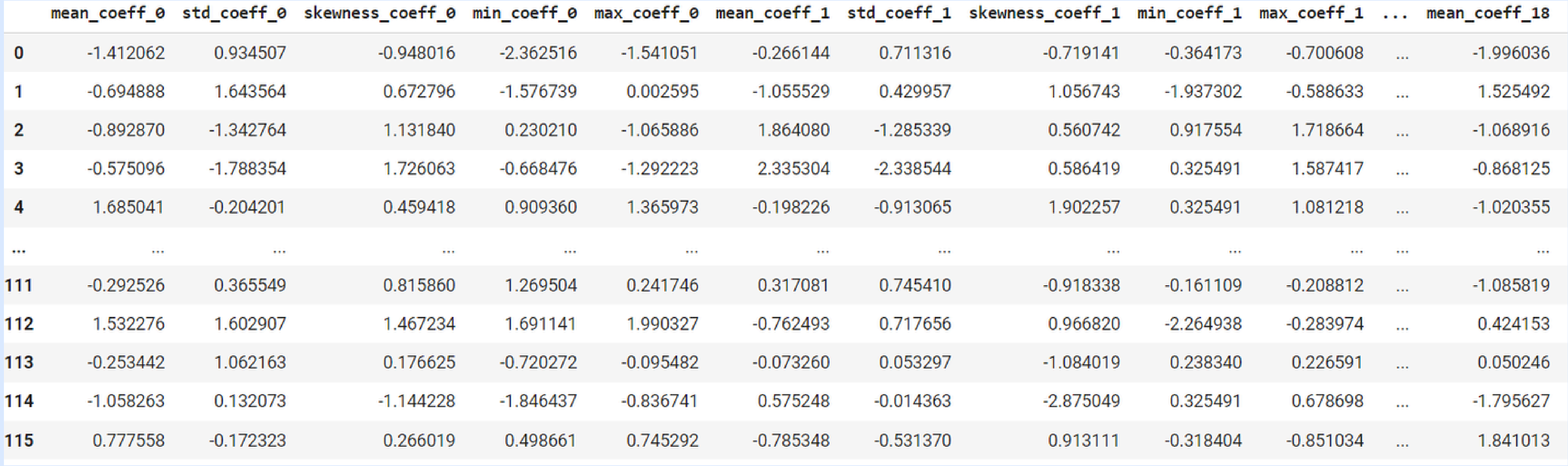


	mean_coeff_0	median_coeff_0	std_coeff_0	skewness_coeff_0	kurtosis_coeff_0	min_coeff_0	max_coeff_0	25th_percentile_coeff_0	75th_percentile_coeff_0	mean_coeff_1
0	-1.412062	-1.000651	0.934507	-0.948016	0.118915	-2.362516	-1.541051	-1.334168	-1.153103	-0.266144
1	-0.694888	-0.397375	1.643564	0.672796	-1.088885	-1.576739	0.002595	-0.955326	-0.114450	-1.055529
2	-0.892870	-0.988809	-1.342764	1.131840	-1.117288	0.230210	-1.065886	-0.698778	-1.130625	1.864080
3	-0.575096	-0.773822	-1.788354	1.726063	-1.124649	-0.668476	-1.292223	-0.406499	-0.912359	2.335304
4	1.685041	1.593301	-0.204201	0.459418	-0.387084	0.909360	1.365973	1.724019	1.471888	-0.198226
...	...	...	...	...	...	...	...	...	...	...
111	-0.292526	-0.187034	0.365549	0.815860	-1.110585	1.269504	0.241746	-0.358523	-0.103343	0.317081
112	1.532276	1.627084	1.602907	1.467234	-1.479491	1.691141	1.990327	0.814814	2.042052	-0.762493
113	-0.253442	-0.271157	1.062163	0.176625	-0.598908	-0.720272	-0.095482	-0.374076	0.154164	-0.073260
114	-1.058263	-1.033682	0.132073	-1.144228	1.104090	-1.846437	-0.836741	-0.912626	-1.084235	0.575248
115	0.777558	0.701682	-0.172323	0.266019	-0.081381	0.498661	0.745292	0.733177	0.683504	-0.785348

The dataframe after **StandardScaler**. There are total of **180 columns**.

- Then we did VIF on the data frame to remove the multicollinear columns. But at the initial iteration **VIF was coming unexpectedly high**. This would mean to remove most of the data.
- So, we first applied correlation matrix, removed the columns with high **Pearson Coefficient value (0.9)** and then applied VIF over it.
- After removing the columns based on Pearson Coefficient, we got a total of **115 columns.**

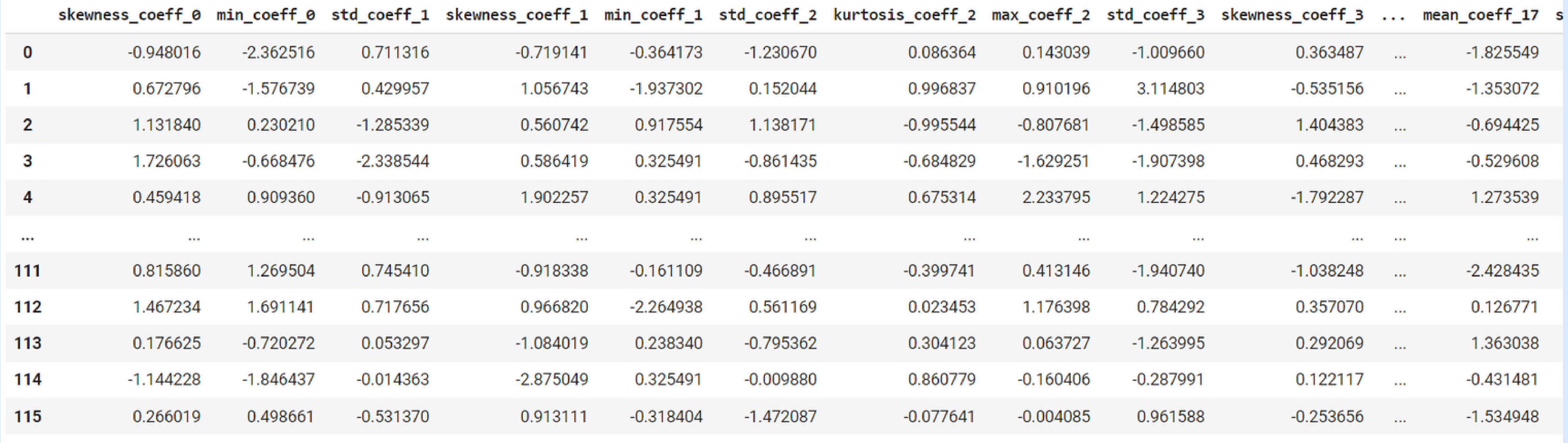
EXPLORATORY DATA ANALYSIS ON MFCC COEFFICIENTS



	mean_coeff_0	std_coeff_0	skewness_coeff_0	min_coeff_0	max_coeff_0	mean_coeff_1	std_coeff_1	skewness_coeff_1	min_coeff_1	max_coeff_1	...	mean_coeff_18
0	-1.412062	0.934507	-0.948016	-2.362516	-1.541051	-0.266144	0.711316	-0.719141	-0.364173	-0.700608	...	-1.996036
1	-0.694888	1.643564	0.672796	-1.576739	0.002595	-1.055529	0.429957	1.056743	-1.937302	-0.588633	...	1.525492
2	-0.892870	-1.342764	1.131840	0.230210	-1.065886	1.864080	-1.285339	0.560742	0.917554	1.718664	...	-1.068916
3	-0.575096	-1.788354	1.726063	-0.668476	-1.292223	2.335304	-2.338544	0.586419	0.325491	1.587417	...	-0.868125
4	1.685041	-0.204201	0.459418	0.909360	1.365973	-0.198226	-0.913065	1.902257	0.325491	1.081218	...	-1.020355
...	...	...	...	...	...	...	...	...	...	...	...	...
111	-0.292526	0.365549	0.815860	1.269504	0.241746	0.317081	0.745410	-0.918338	-0.161109	-0.208812	...	-1.085819
112	1.532276	1.602907	1.467234	1.691141	1.990327	-0.762493	0.717656	0.966820	-2.264938	-0.283974	...	0.424153
113	-0.253442	1.062163	0.176625	-0.720272	-0.095482	-0.073260	0.053297	-1.084019	0.238340	0.226591	...	0.050246
114	-1.058263	0.132073	-1.144228	-1.846437	-0.836741	0.575248	-0.014363	-2.875049	0.325491	0.678698	...	-1.795627
115	0.777558	-0.172323	0.266019	0.498661	0.745292	-0.785348	-0.531370	0.913111	-0.318404	-0.851034	...	1.841013

The data frame after **correlation-based** feature selection. There are a total of **115 columns**.

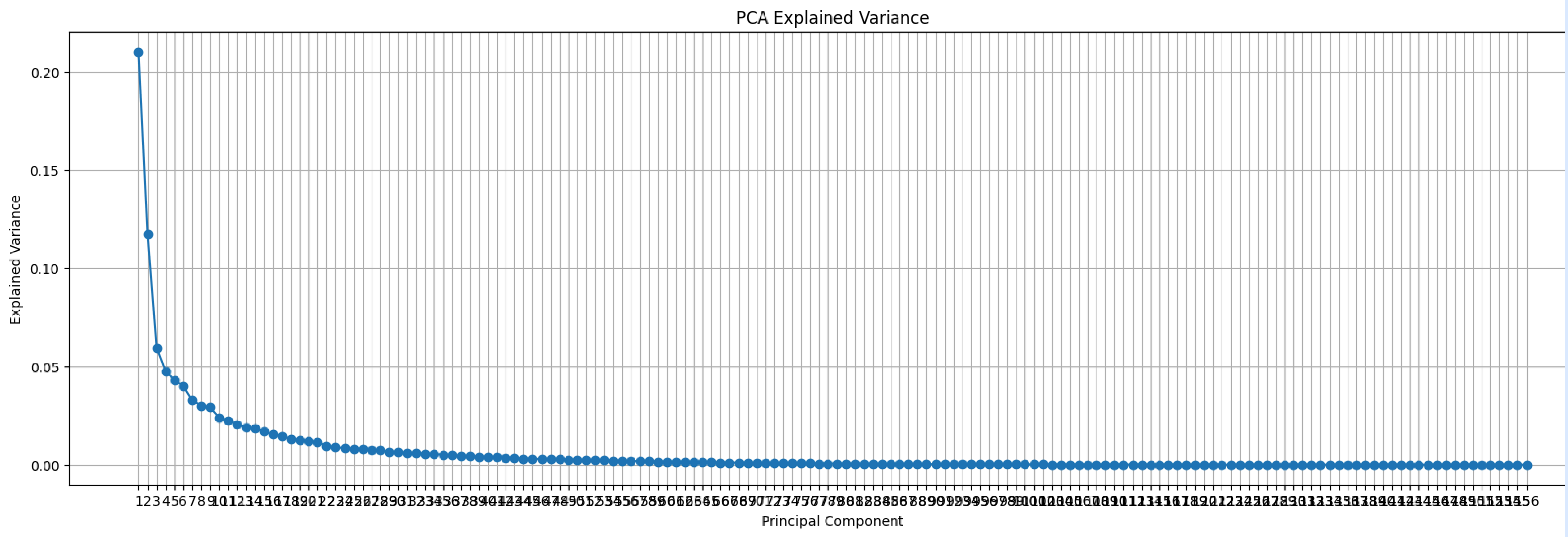
- Now we do VIF with a **VIF threshold of 10** on the new data frame to drop the multi-collinear columns.



	skewness_coeff_0	min_coeff_0	std_coeff_1	skewness_coeff_1	min_coeff_1	std_coeff_2	kurtosis_coeff_2	max_coeff_2	std_coeff_3	skewness_coeff_3	...	mean_coeff_17	s
0	-0.948016	-2.362516	0.711316	-0.719141	-0.364173	-1.230670	0.086364	0.143039	-1.009660	0.363487	...	-1.825549	
1	0.672796	-1.576739	0.429957	1.056743	-1.937302	0.152044	0.996837	0.910196	3.114803	-0.535156	...	-1.353072	
2	1.131840	0.230210	-1.285339	0.560742	0.917554	1.138171	-0.995544	-0.807681	-1.498585	1.404383	...	-0.694425	
3	1.726063	-0.668476	-2.338544	0.586419	0.325491	-0.861435	-0.684829	-1.629251	-1.907398	0.468293	...	-0.529608	
4	0.459418	0.909360	-0.913065	1.902257	0.325491	0.895517	0.675314	2.233795	1.224275	-1.792287	...	1.273539	
...	...	...	...	...	...	...	...	...	...	...	...	...	
111	0.815860	1.269504	0.745410	-0.918338	-0.161109	-0.466891	-0.399741	0.413146	-1.940740	-1.038248	...	-2.428435	
112	1.467234	1.691141	0.717656	0.966820	-2.264938	0.561169	0.023453	1.176398	0.784292	0.357070	...	0.126771	
113	0.176625	-0.720272	0.053297	-1.084019	0.238340	-0.795362	0.304123	0.063727	-1.263995	0.292069	...	1.363038	
114	-1.144228	-1.846437	-0.014363	-2.875049	0.325491	-0.009880	0.860779	-0.160406	-0.287991	0.122117	...	-0.431481	
115	0.266019	0.498661	-0.531370	0.913111	-0.318404	-1.472087	-0.077641	-0.004085	0.961588	-0.253656	...	-1.534948	

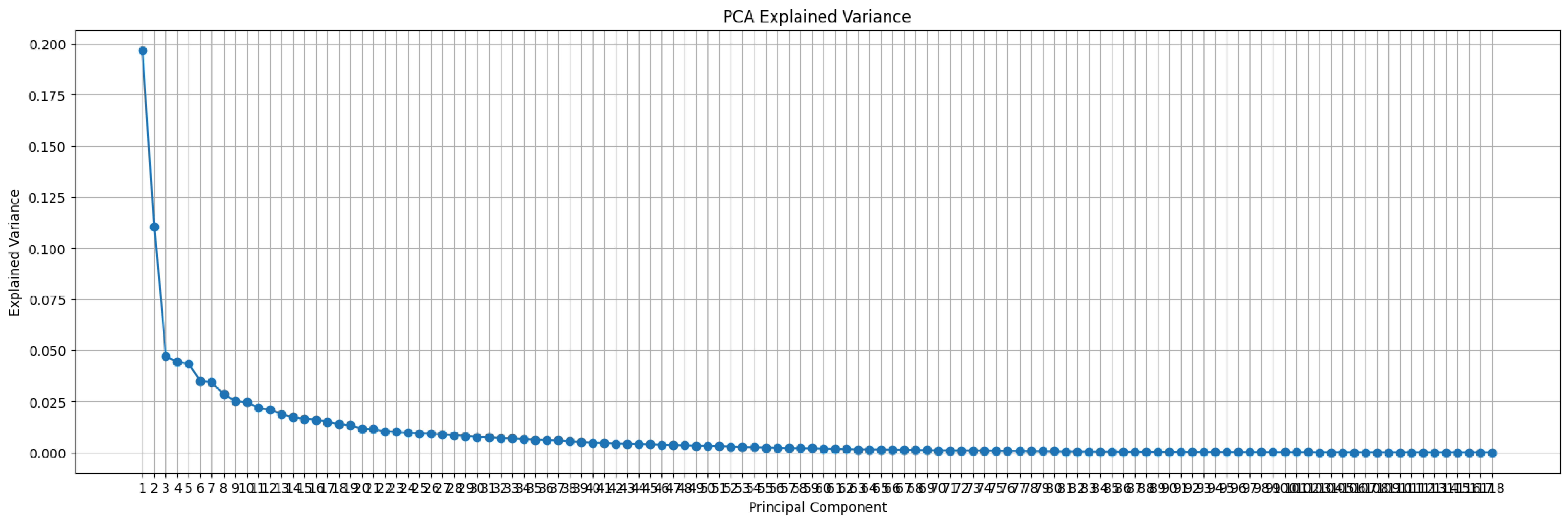
The data frame after **VIF**. There are a total of **23 columns.**

COMPARING ELBOW PLOTS OF 3 TYPES OF DATAFRAME



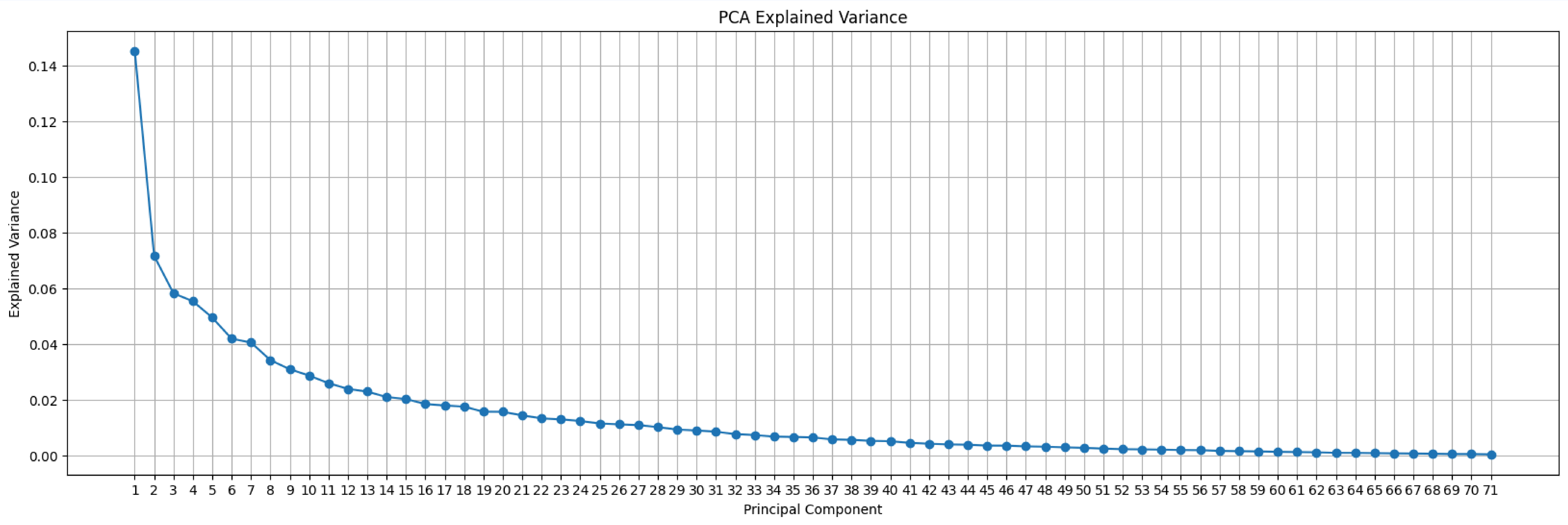
Before EDA data frame

COMPARING ELBOW PLOTS OF 3 TYPES OF DATAFRAME



After Correlation based feature extraction

COMPARING ELBOW PLOTS OF 3 TYPES OF DATAFRAME

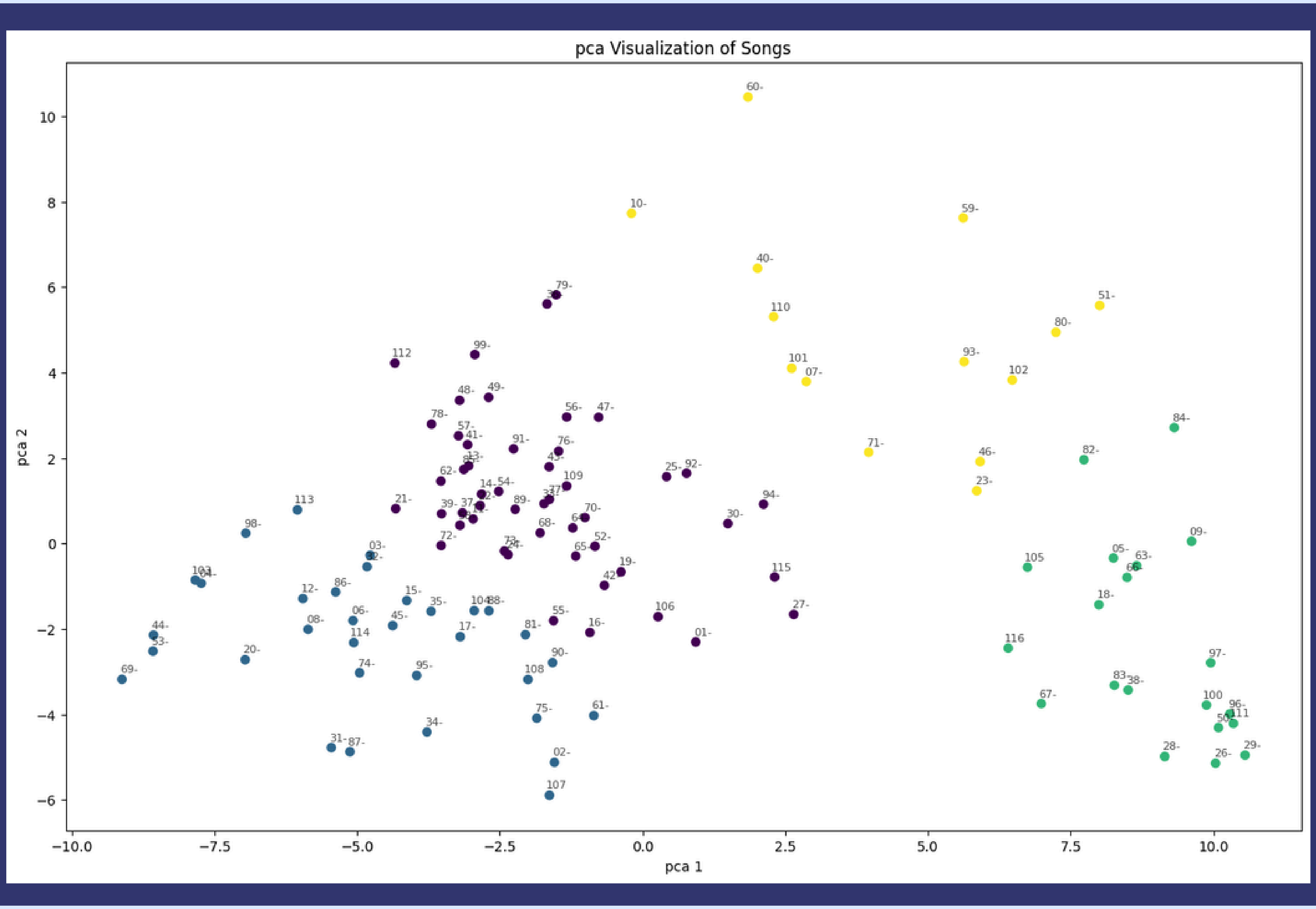


After VIF and correlation-based feature extraction

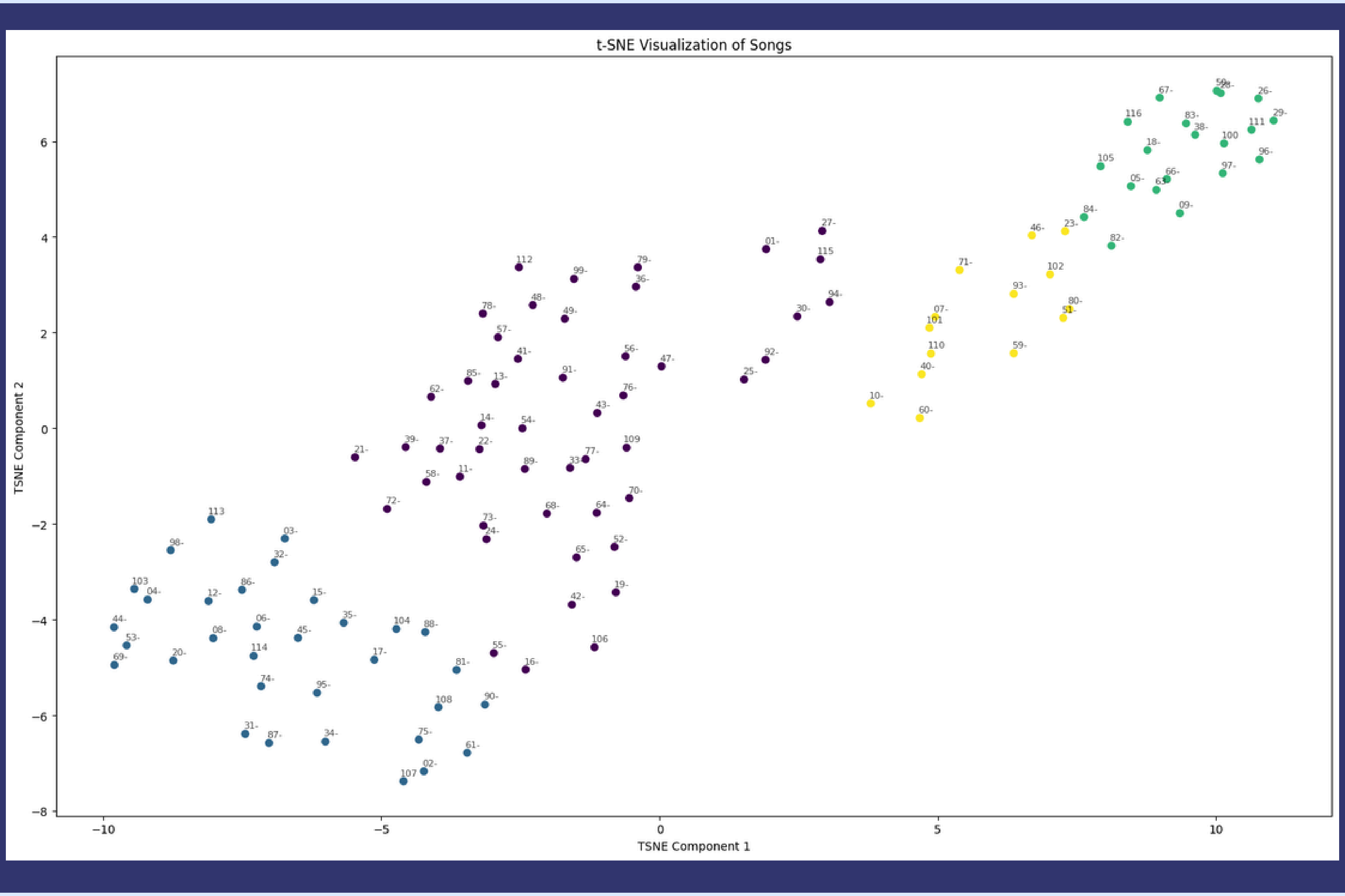
PRINCIPAL COMPONENT ANANLYIS AND CLUSTERING

- We did Principal Component Analysis (for dimensionality reduction) on the data frame with total components to be 2.
- We did t-SNE also on the input data with total components to be 2.

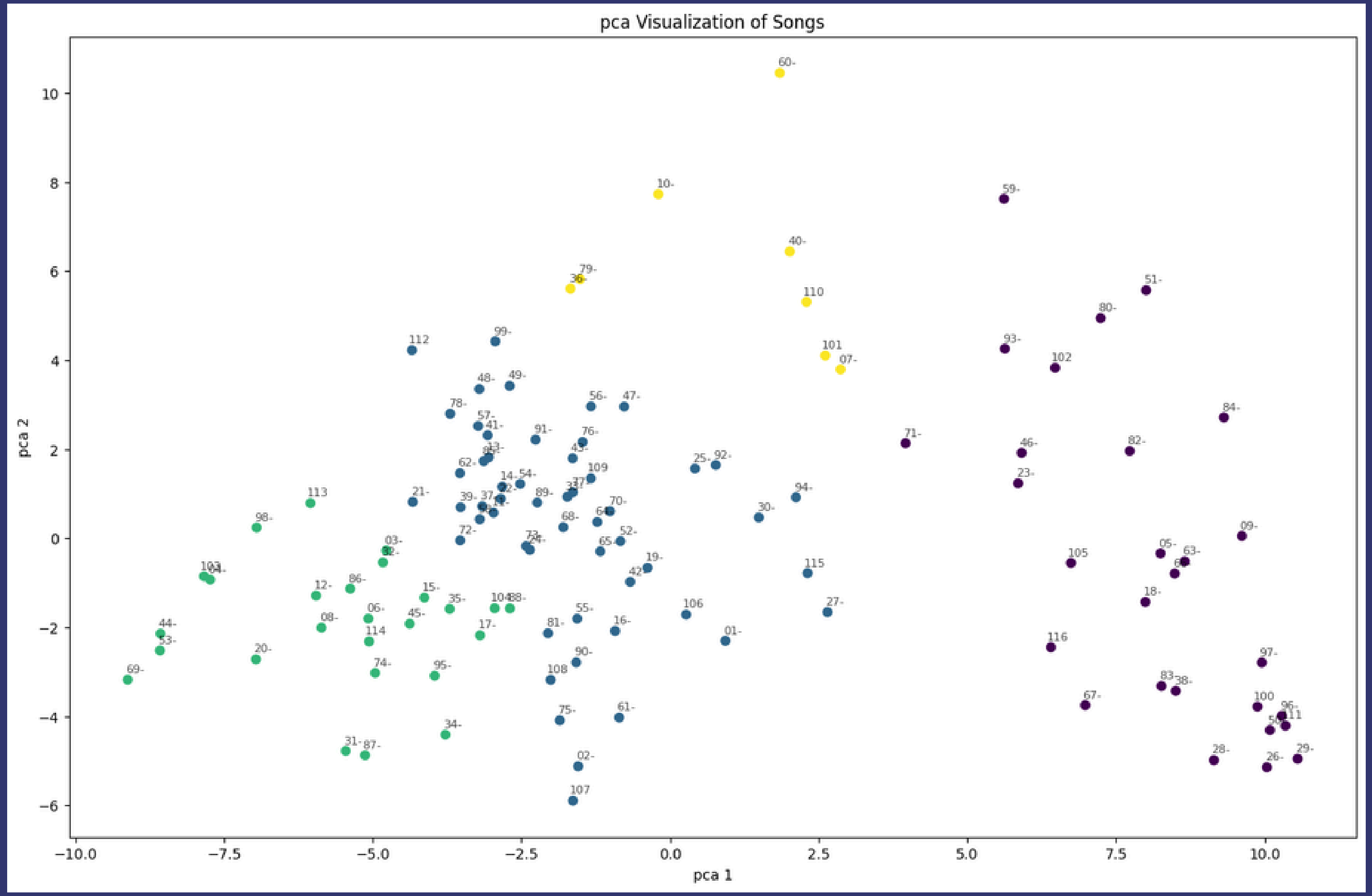
In next iteration we did K-means and Agglomerative Clustering on the PCA data.



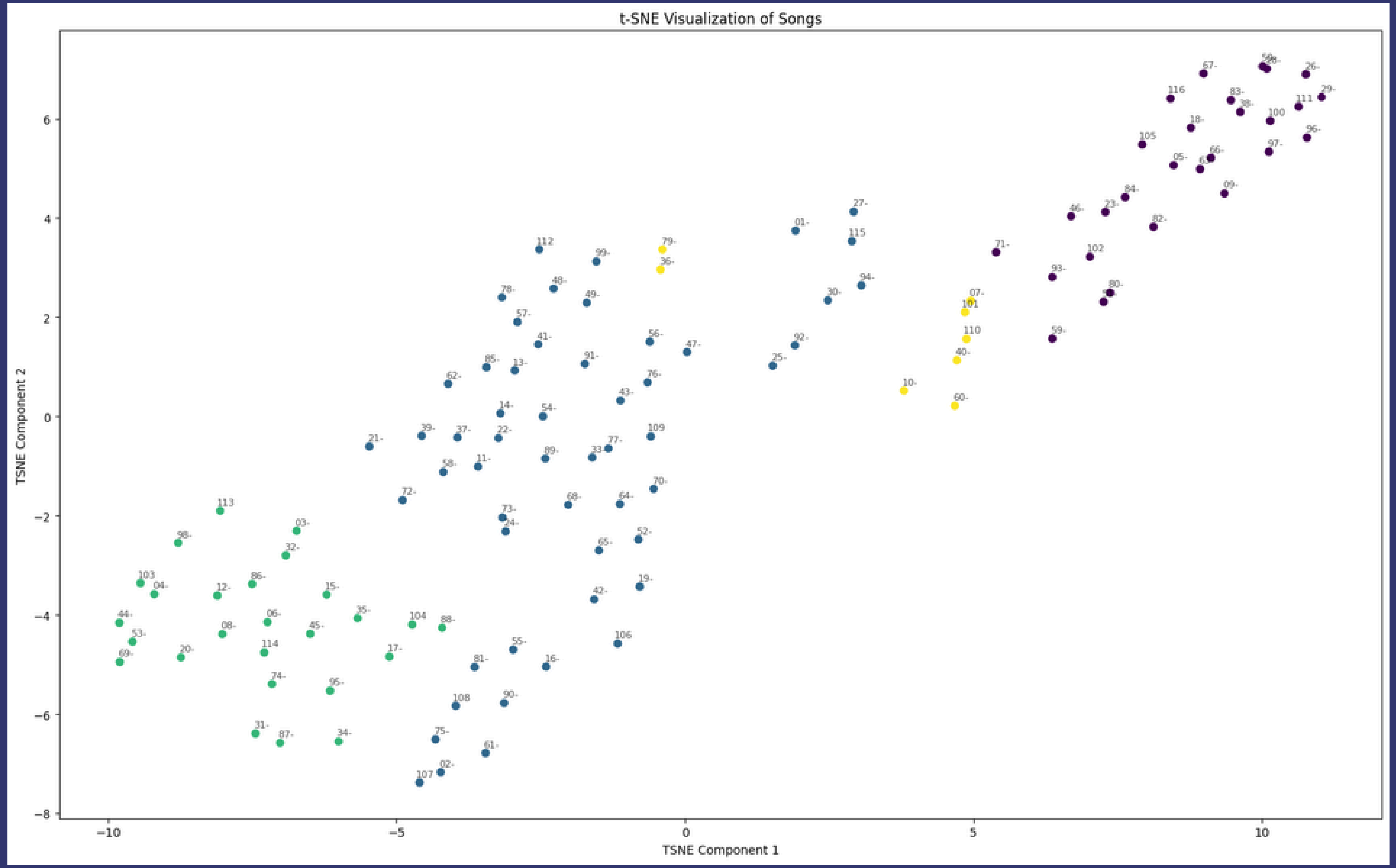
PCA visualization of labels from K-Means



t-SNE visualization of labels from K-Means



PCA visualization of labels from Agglomerative Clustering

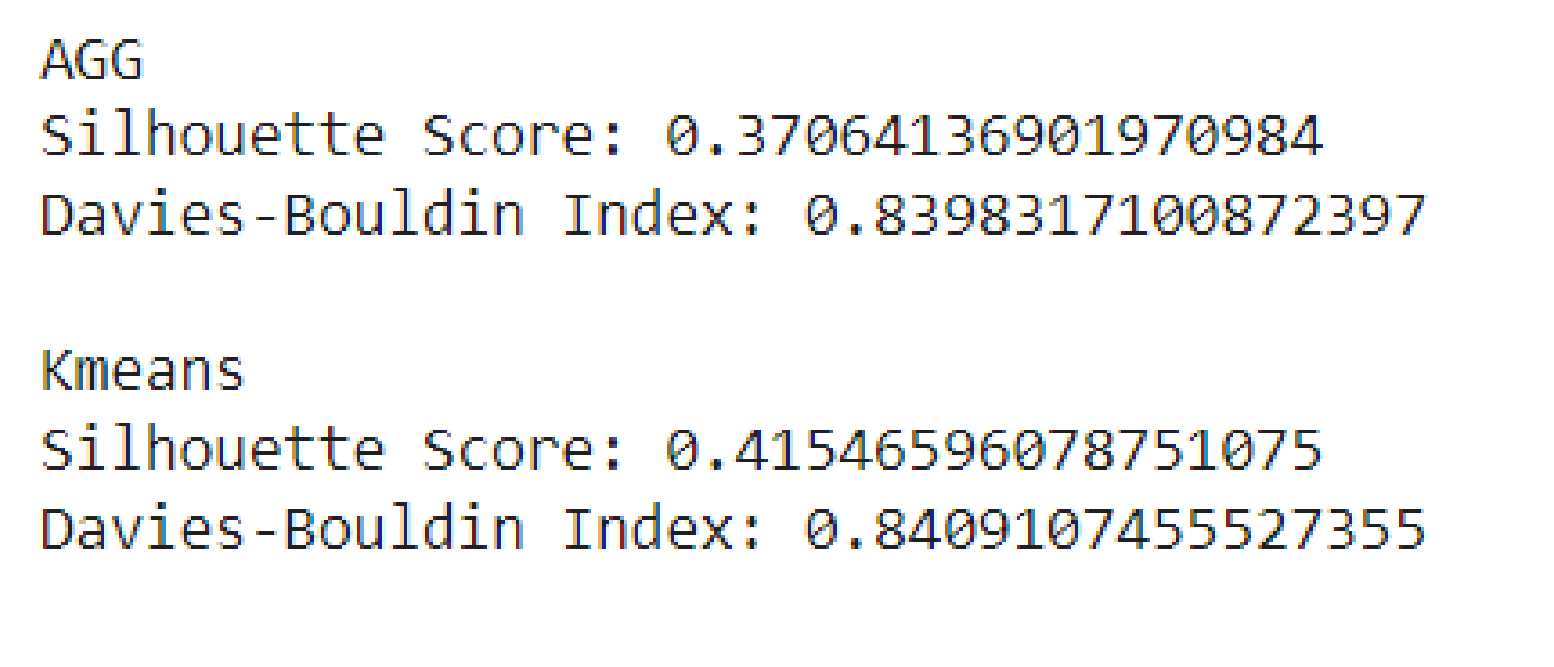


t-SNE visualization of labels from Agglomerative Clustering

Comparison of the 2 clustering algorithms

- **Agglomerative Clustering -** AGG gave Silhouette Score of 0.37 and Davies Bouldin Index of 0.84 which is pretty bad and that’s why we switched to K-Means.

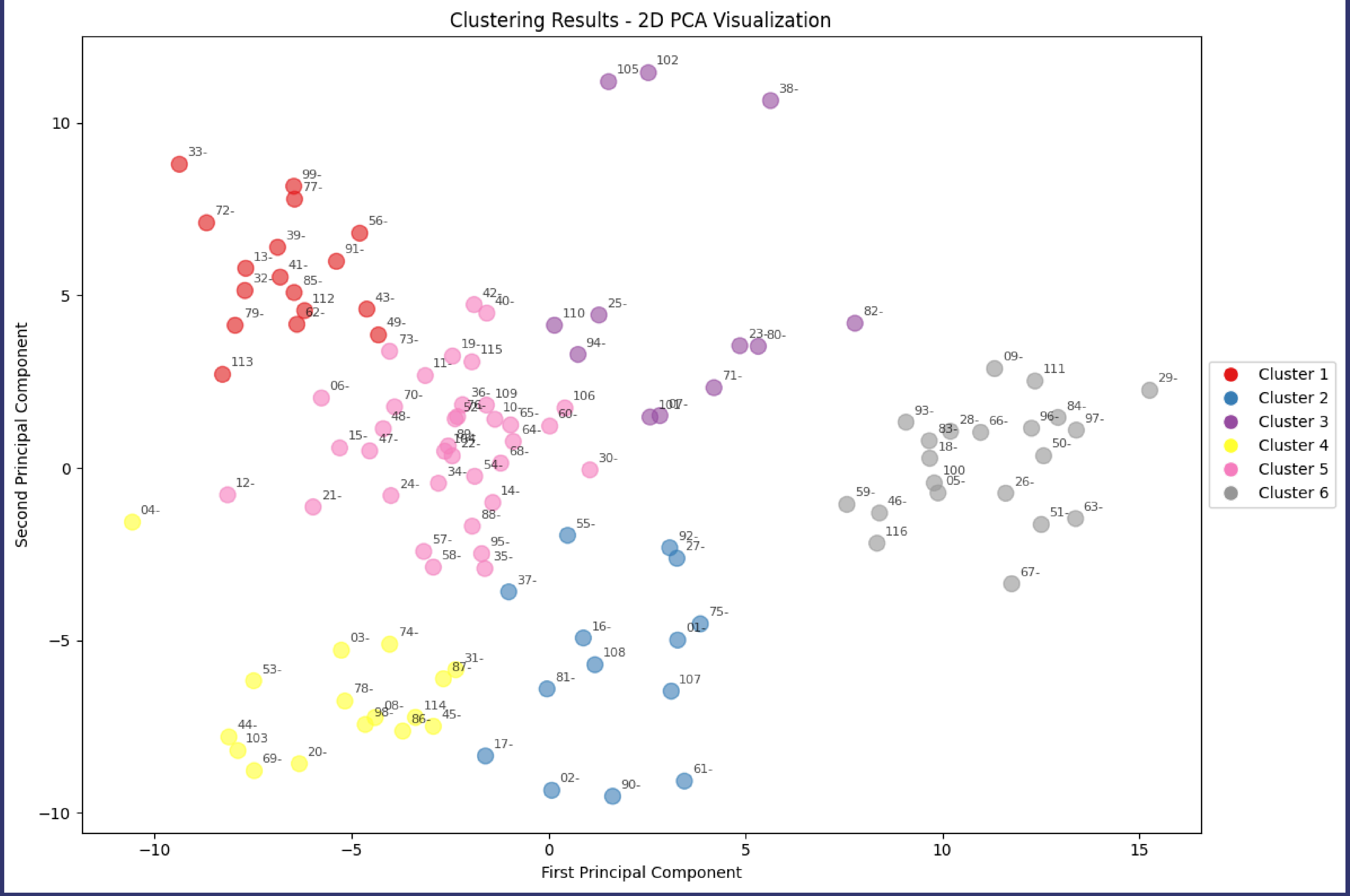
- **K-Means Clustering -** Kmeans gave Silhouette Score of 0.41 and Davies Bouldin Index of 0.84 which not that much of an improvement but still better.



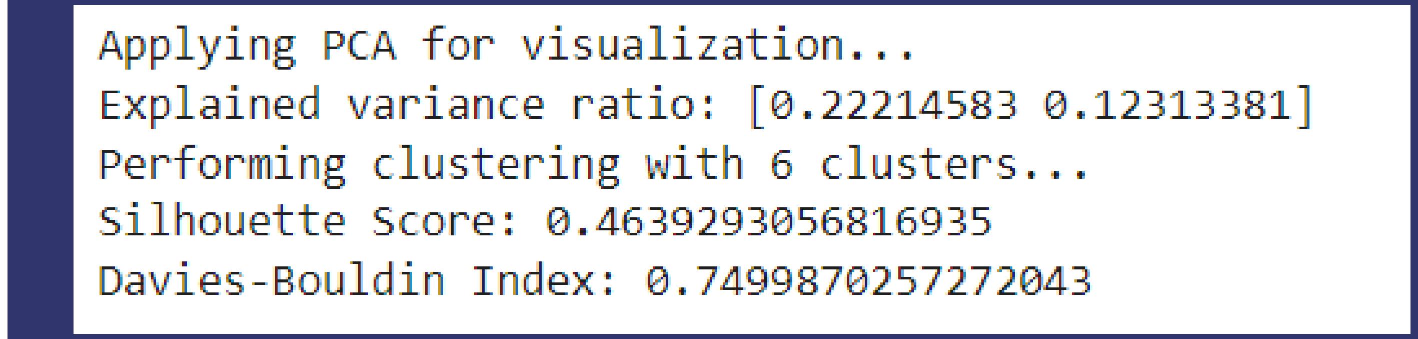
```
AGG
Silhouette Score: 0.37064136901970984
Davies-Bouldin Index: 0.8398317100872397

Kmeans
Silhouette Score: 0.41546596078751075
Davies-Bouldin Index: 0.8409107455527355
```

Iteratively the better clustering



- This was iteratively a better clustering score we got out of all the direct approaches we tried
- This was by K-Means with a Silhouette Score of 0.464 and Davies-Bouldin Index of 0.74



```
Applying PCA for visualization...
Explained variance ratio: [0.22214583 0.12313381]
Performing clustering with 6 clusters...
Silhouette Score: 0.4639293056816935
Davies-Bouldin Index: 0.7499870257272043
```

- Also this was with an average Cohesion Score of 2.66 across all the clusters.
- Cohesion Score is basically mean of norm of distance between cluster points and center.

NOT SO GOOD: Clusters were not that dense, some even had Cohesion score of 4. That’s why we left this approach and focussed more on understanding and optimizing MFCC coefficients

MFCCs and Speech Recognition

- **Zeroth coefficient of MFCC** (in our case the header of the csv files) is representative of **energy of the frame**. It often included as it provides valuable information about emphasis and speech segments

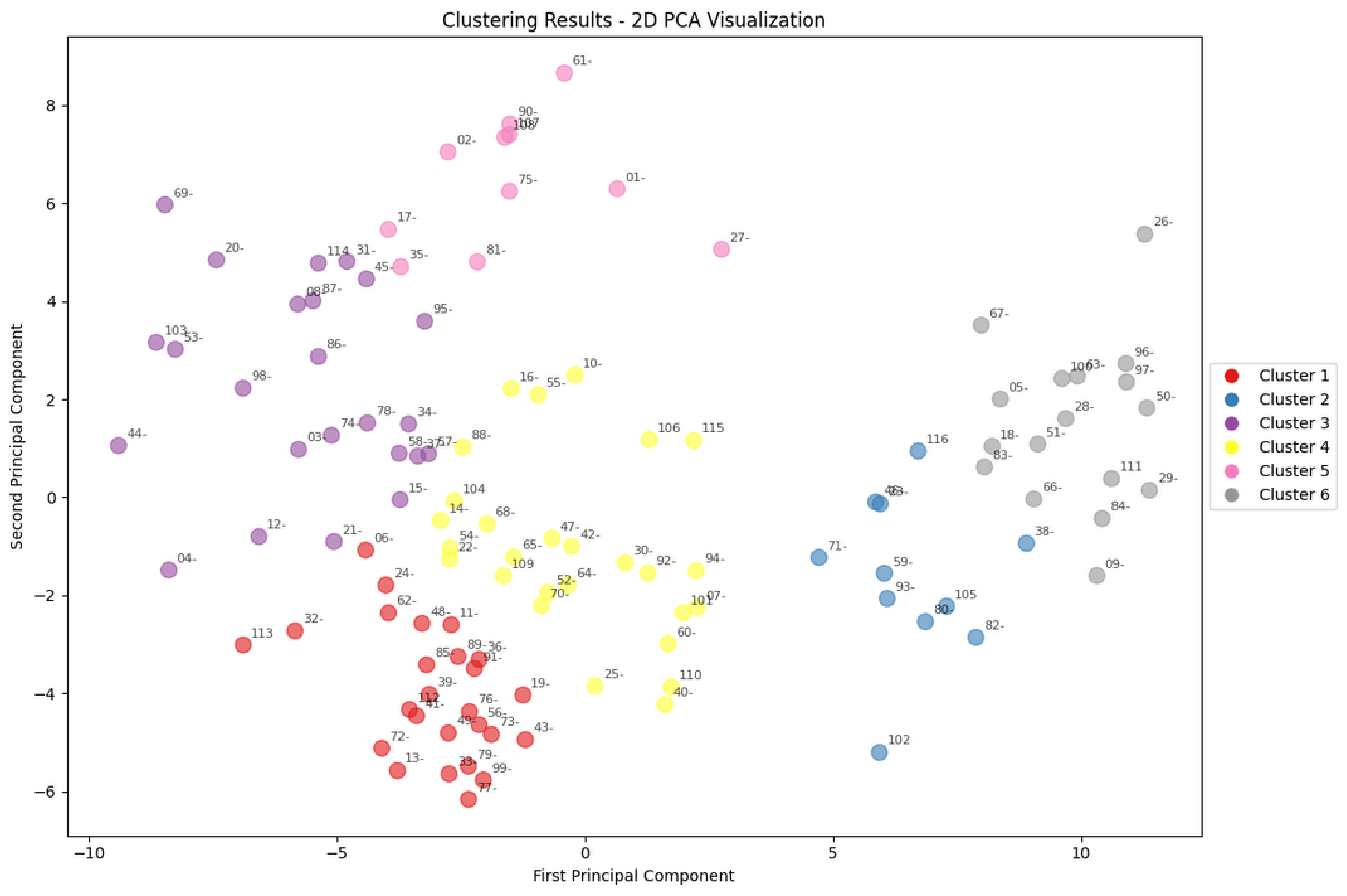
- **Lower MFCCs** (e.g., coefficients 1 to 3): Represent the **general shape of the spectrum**, capturing coarse information like the loudness and fundamental pitch.

- **Middle MFCCs** (e.g., coefficients 4 to 12): Capture the **detailed formant structure**, which contains the most critical information for distinguishing phonemes in speech. This range is generally the most useful for speech recognition tasks.

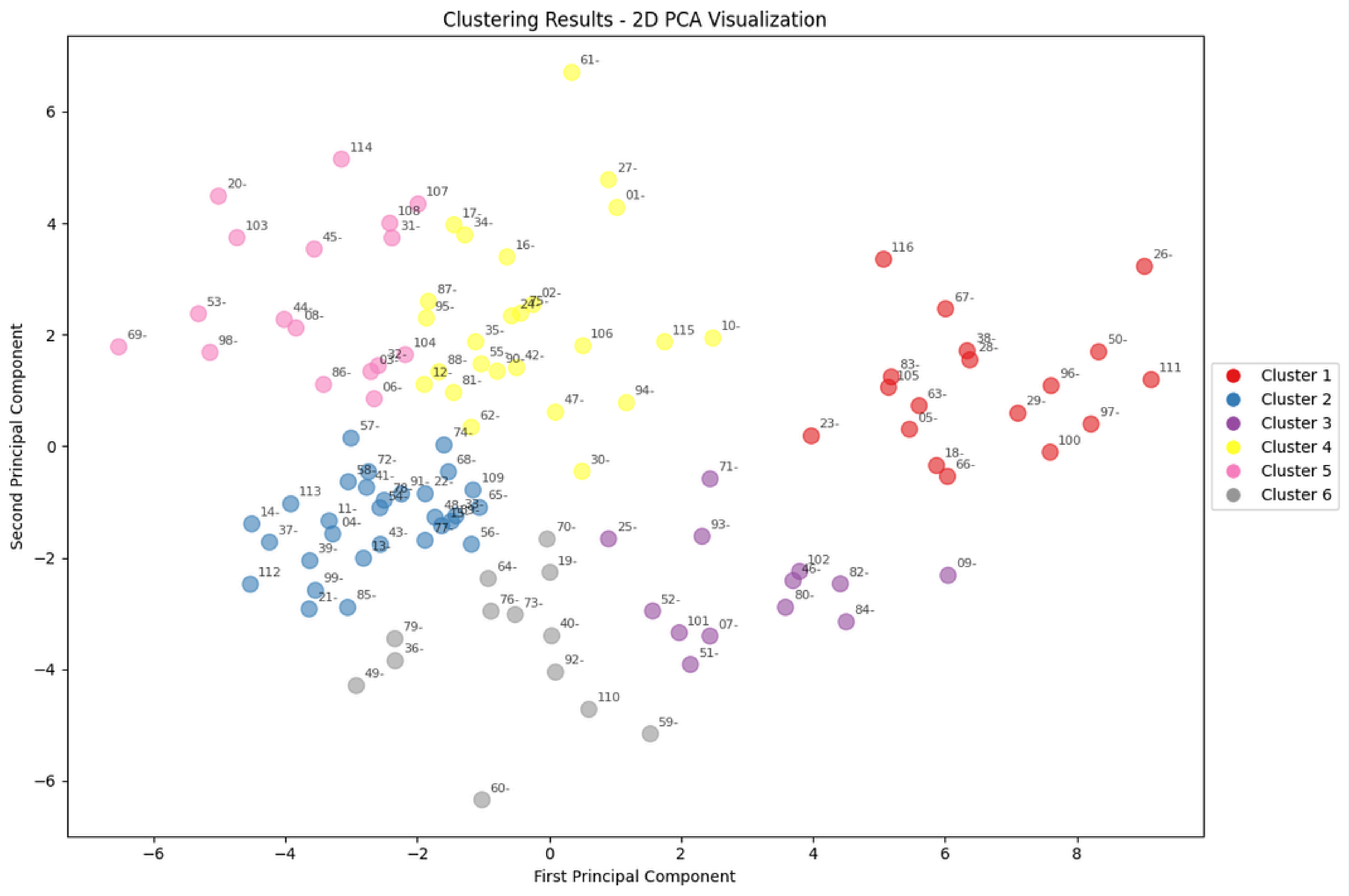
- **Higher MFCCs**: Represent rapid **spectral variations**, often related to noise rather than phonetic information. **These are typically not as useful for speech recognition.**

What we did?

- Initially we tried with taking first 13 coefficients (including 0) as input data and did PCA and clustering over it
- But the scores got even worse. Silhouette Score became 0.38 and Davies-Bouldin index became 0.81

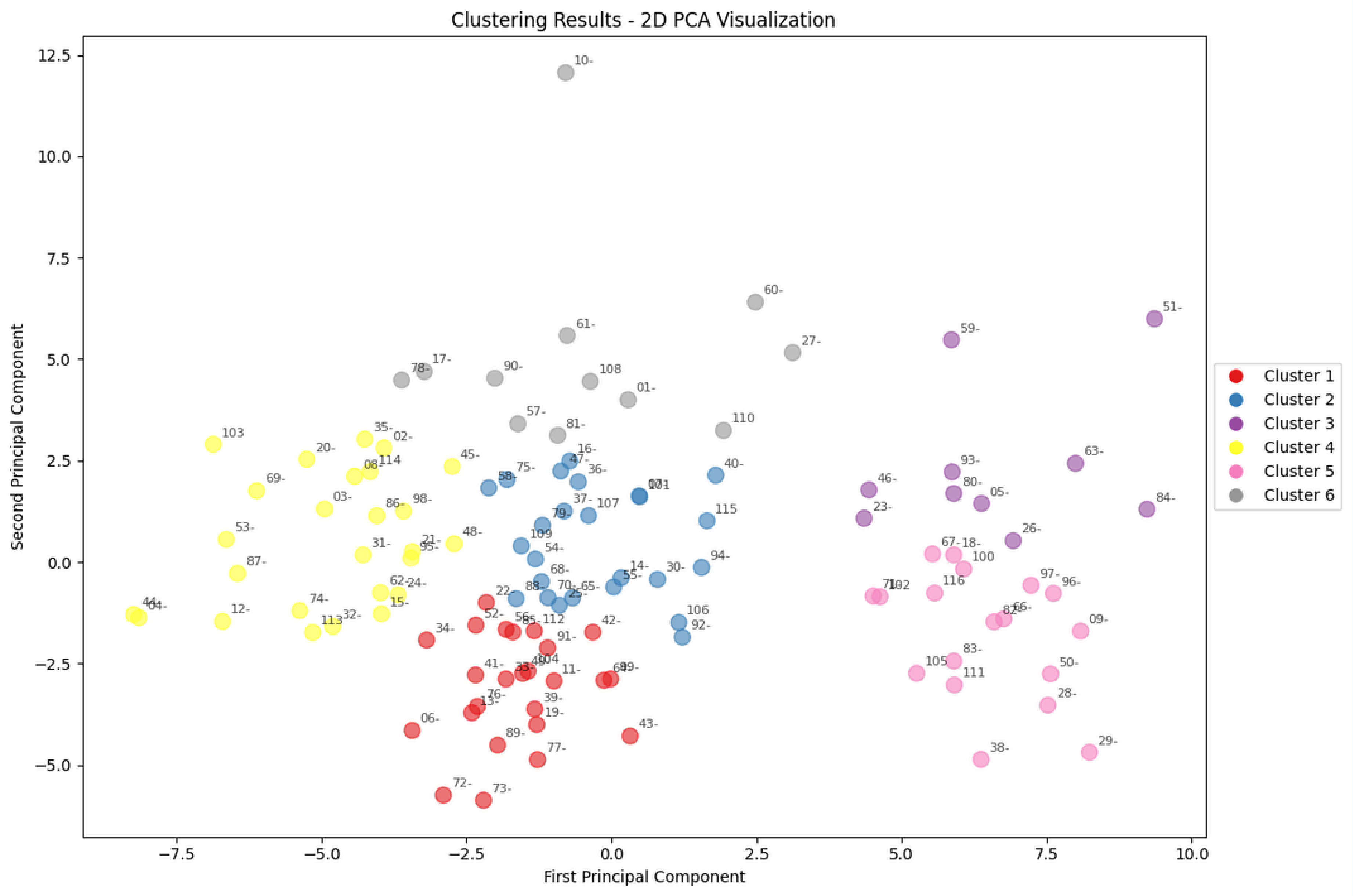


Further Iterations on Clustering



PCA plot of K-Means for first 4 coefficients (0 - 3)

- Silhouette Score of 0.37 and DB index of 0.89



PCA plot of K-Means for 4 - 12 coefficients.

- Silhouette Score of 0.37 and DB index of 0.87

Since Clustering didn’t work, we shifted to training a **neural network** for classification of singers!!

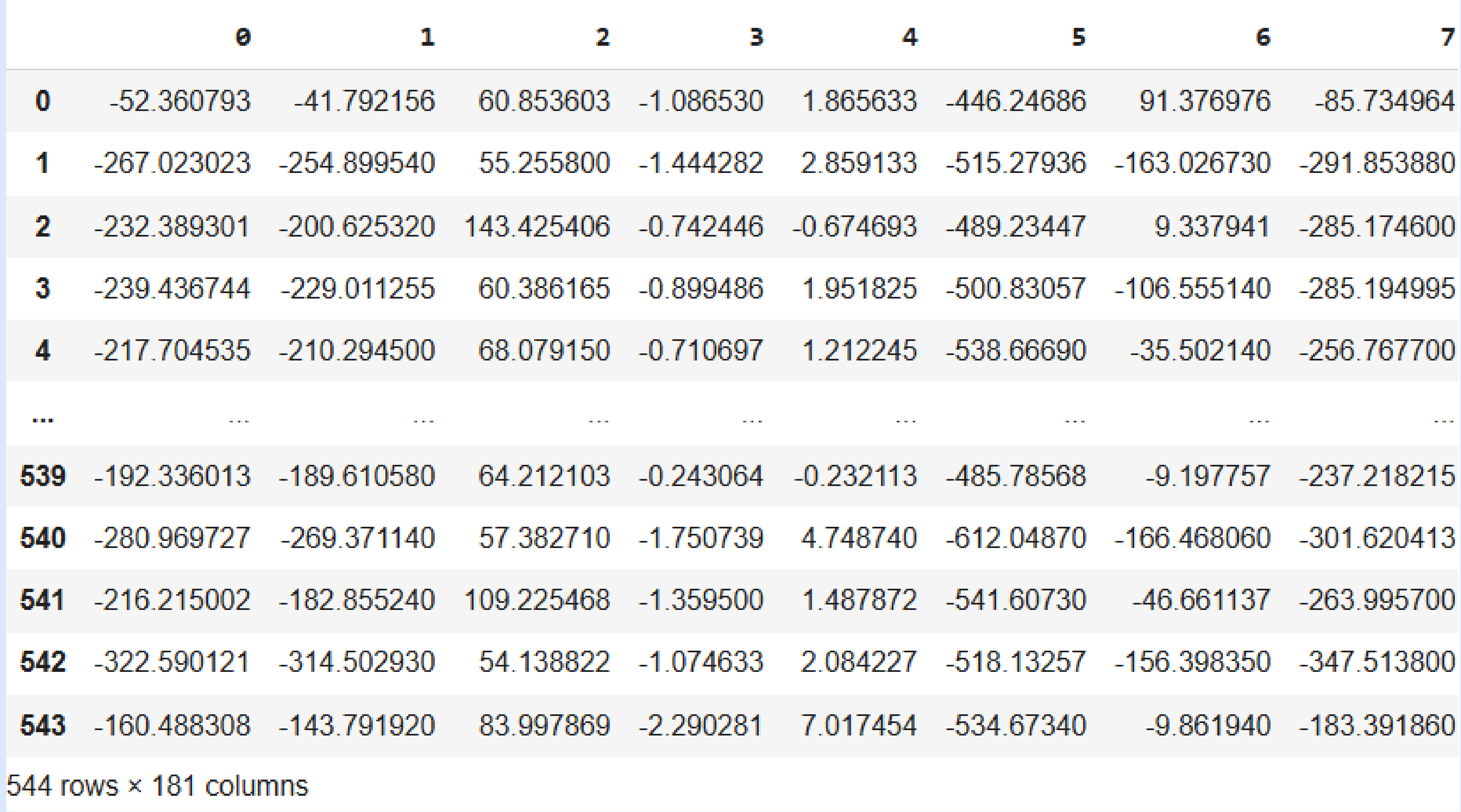
ITERATION 2 - MULTI LAYER PERCEPTRON

TRAIN DATA COLLECTION AND PREPARATION

After the unsatisfied results of clustering aproach, we decided to train a classification model to classify the data into different classes as discussed earlier

First of all we collected data of different categories of songs of the required artists, genres and languages and created labelled datasets of MFCC coefficients similar to that of data set provided.

Following it the data was processed as we talked earlier using different statistics
The Labels of of the data were label encoded



	0	1	2	3	4	5	6	7
0	-52.360793	-41.792156	60.853603	-1.086530	1.865633	-446.24686	91.376976	-85.734964
1	-267.023023	-254.899540	55.255800	-1.444282	2.859133	-515.27936	-163.026730	-291.853880
2	-232.389301	-200.625320	143.425406	-0.742446	-0.674693	-489.23447	9.337941	-285.174600
3	-239.436744	-229.011255	60.386165	-0.899486	1.951825	-500.83057	-106.555140	-285.194995
4	-217.704535	-210.294500	68.079150	-0.710697	1.212245	-538.66690	-35.502140	-256.767700
...	...	...	...	...	...	...	...	...
539	-192.336013	-189.610580	64.212103	-0.243064	-0.232113	-485.78568	-9.197757	-237.218215
540	-280.969727	-269.371140	57.382710	-1.750739	4.748740	-612.04870	-166.468060	-301.620413
541	-216.215002	-182.855240	109.225468	-1.359500	1.487872	-541.60730	-46.661137	-263.995700
542	-322.590121	-314.502930	54.138822	-1.074633	2.084227	-518.13257	-156.398350	-347.513800
543	-160.488308	-143.791920	83.997869	-2.290281	7.017454	-534.67340	-9.861940	-183.391860

544 rows × 181 columns

The resulting dataset we created comprises a total of 544 distinct audio files, with approximately 90 audio samples per class. Consequently, the resulting training data has dimensions of 544 x 181

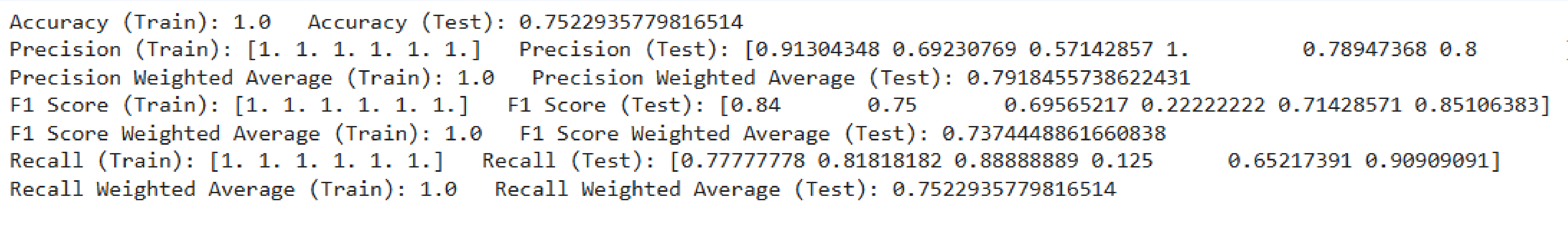
The mapping was done as
0 for Michael Jackson, 1 for N-Anthem, 2 for Asha Bhosle, 3 for Kishore, 4 for Lavani and 5 for Bhavgeet

Model Training

First we created a test train split of 0.2 from the dataset and trained multiple classification models on it like

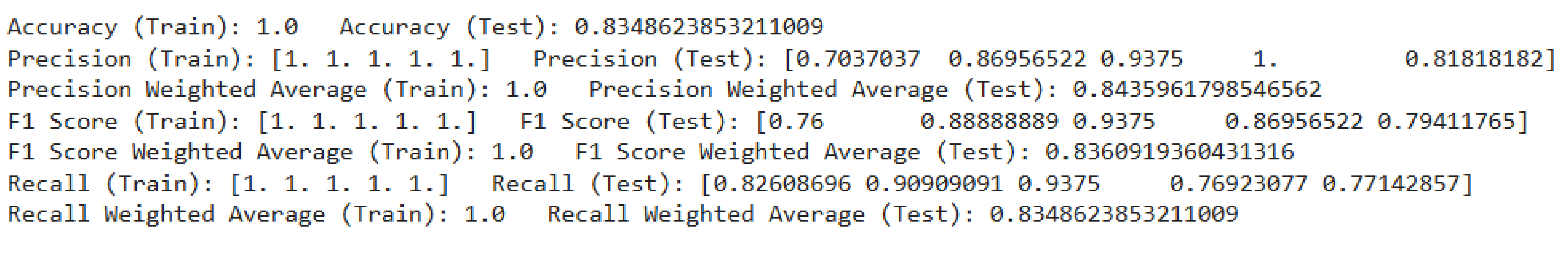
- Random forest classifier
- XGBoost Classifier
- Neural Network

Out the above methods neural networks seemed to outperform the previous two methods



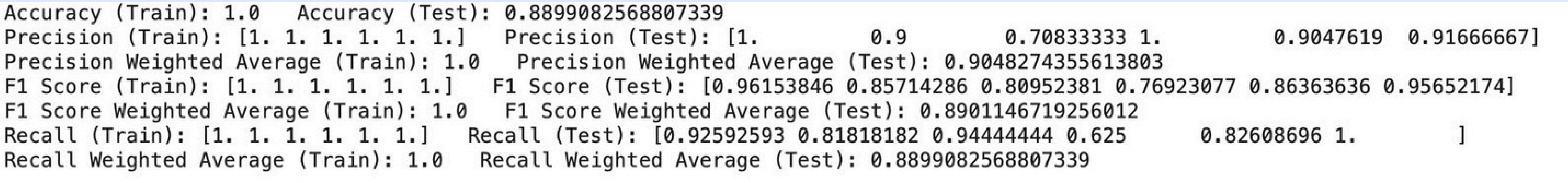
```
Accuracy (Train): 1.0   Accuracy (Test): 0.7522935779816514
Precision (Train): [1. 1. 1. 1. 1. 1.]   Precision (Test): [0.91304348 0.69230769 0.57142857 1.         0.78947368 0.8
Precision Weighted Average (Train): 1.0   Precision Weighted Average (Test): 0.7918455738622431
F1 Score (Train): [1. 1. 1. 1. 1. 1.]   F1 Score (Test): [0.84       0.75       0.69565217 0.22222222 0.71428571 0.85106383]
F1 Score Weighted Average (Train): 1.0   F1 Score Weighted Average (Test): 0.7374448861660838
Recall (Train): [1. 1. 1. 1. 1. 1.]   Recall (Test): [0.77777778 0.81818182 0.88888889 0.125      0.65217391 0.90909091]
Recall Weighted Average (Train): 1.0   Recall Weighted Average (Test): 0.7522935779816514
```

Metrics for Random Forest



```
Accuracy (Train): 1.0   Accuracy (Test): 0.8348623853211009
Precision (Train): [1. 1. 1. 1. 1.]   Precision (Test): [0.7037037  0.86956522 0.9375     1.         0.81818182]
Precision Weighted Average (Train): 1.0   Precision Weighted Average (Test): 0.8435961798546562
F1 Score (Train): [1. 1. 1. 1. 1.]   F1 Score (Test): [0.76       0.88888889 0.9375     0.86956522 0.79411765]
F1 Score Weighted Average (Train): 1.0   F1 Score Weighted Average (Test): 0.8360919360431316
Recall (Train): [1. 1. 1. 1. 1.]   Recall (Test): [0.82608696 0.90909091 0.9375     0.76923077 0.77142857]
Recall Weighted Average (Train): 1.0   Recall Weighted Average (Test): 0.8348623853211009
```

Metrics for XGBoost Classifier



```
Accuracy (Train): 1.0   Accuracy (Test): 0.8899082568807339
Precision (Train): [1. 1. 1. 1. 1. 1.]   Precision (Test): [1.         0.9        0.70833333 1.         0.9047619  0.91666667]
Precision Weighted Average (Train): 1.0   Precision Weighted Average (Test): 0.9048274355613803
F1 Score (Train): [1. 1. 1. 1. 1. 1.]   F1 Score (Test): [0.96153846 0.85714286 0.80952381 0.76923077 0.86363636 0.95652174]
F1 Score Weighted Average (Train): 1.0   F1 Score Weighted Average (Test): 0.8901146719256012
Recall (Train): [1. 1. 1. 1. 1. 1.]   Recall (Test): [0.92592593 0.81818182 0.94444444 0.625      0.82608696 1.        ]
Recall Weighted Average (Train): 1.0   Recall Weighted Average (Test): 0.8899082568807339
```

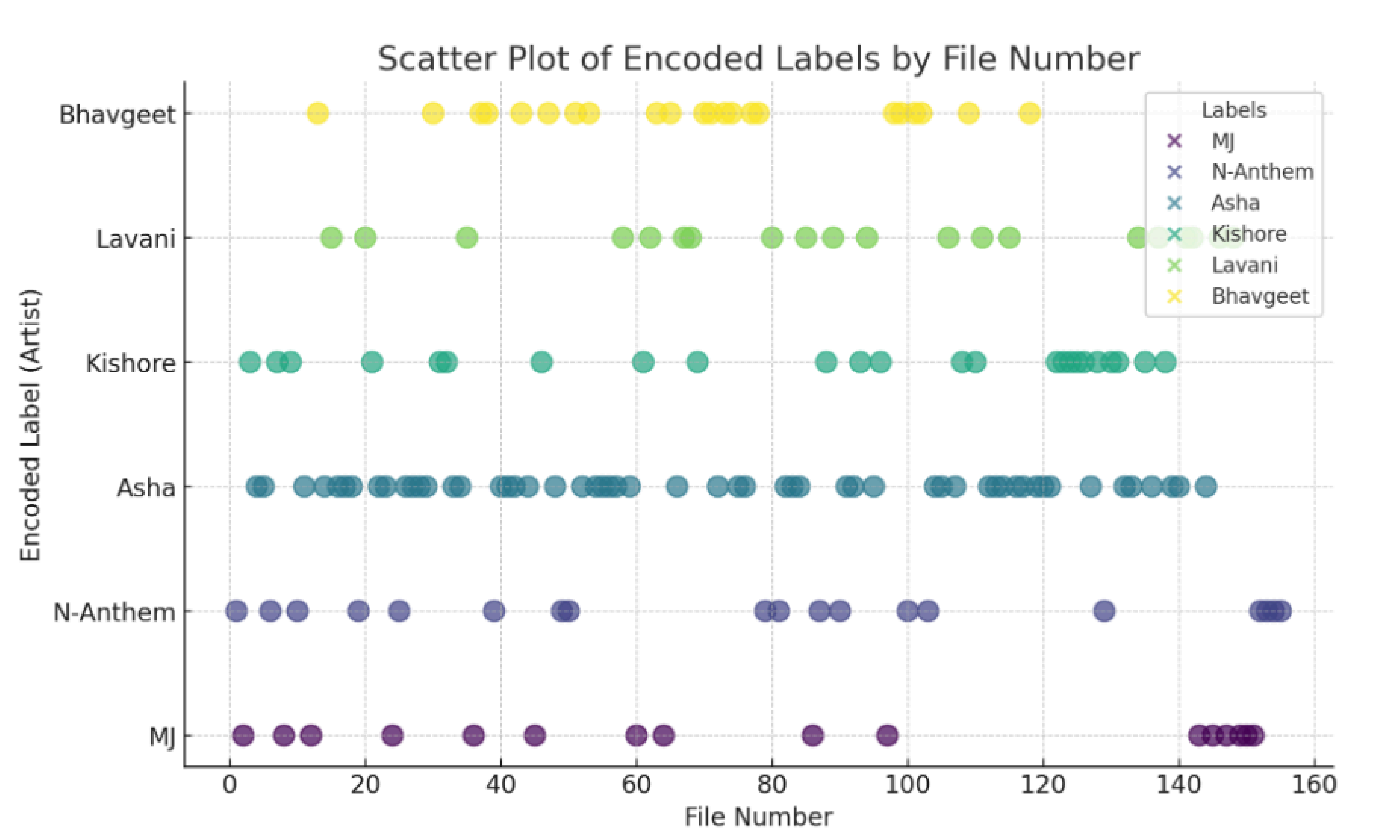
Metrics for Neural Networks

Predictions for the given data

The trained model is then used to predict which class does a given MFCC file belong to.
The results of the prediction are as shown in the Scatter plot

The Initital model trained had a lot of misclassification errors for Bhavgeets because many of the songs in our dataset that we got were sung by Asha Bhosle

So we further added more samples which contained bhavgeet songs which were not sung by Asha Bhosle and tried to improve the accuracy of model for Bhavgeet



Answers to the Problem

All the 115 Files are classified into the categories mentioned in the list as shown in the scatter plot in the previous silde

National Anthem MFCC files	1, 2, 16
Kishore Kumar MFCC files	5, 9 ,18
Asha Bhosle MFCC files	4, 6, 12
Michael Jackson MFCC files	3, 8, 20

Major Learnings

- Learnt about the role of MFCC coefficients in audio processing and speech recognition.
- In-depth analysis of variance inflation factor, principal component analysis, T-SNE while performing clustering of unlabelled dataset.
- In-depth understanding of different statistic metrics while feature engineering to reduce the dataset.
- Learnt and studied about Multi-Layer Perceptor, an artificial neural network, procedure to get more accurate clustering and get better results when compared to clustering in unsupervised learning.

Source Code

CLUSTER

MLP Classifier

Thank You