STA142B Final Project

James Jiang: Contributed to feature extraction, functions to generate design matrix, PCA, scree and cost, K-means for PCA and MDS, DBSCAN parameter tuning. Contributed to the write-up.

Jeff Anderson: Methodology. Dimension Reduction. Function wrapping. Cluster and MDS assessments. Clustering methods.Sang along to mp3 files with his voice of an angle.

Kevin Xu: Contributed to selecting feature groups and features for clustering. Constructed dendrograms to tune k, the number of clusters (for K-Means). Contributed to writing a function that reduces dimensionality, clusters the data, extracts the cluster indices, and returns a silhouette plot. Contributed to the write-up.

Introduction

Music information retrieval is an emerging sub-topic of Audio clustering, a discipline that has grown in popularity in recent years. The purpose of this project is to build a robust unsupervised model that clusters music audio mp3 files by genre.

Our approach is to first extract and generate a selection of audio features based on spectral and rhythmic characteristics. To do this, we investigated various functions from the `librosa` package, and selected feature groups and associated features that we felt would be important for distinguishing genres. Next, we applied various combinations of linear and nonlinear dimension reduction techniques to our data. Finally, we utilized various clustering techniques to cluster our data. To tune our dimension-reduction and clustering hyperparameters, we created a grid-search using the silhouette score, observed the associated scatterplot color-coded by clustering index, and listened to how the audio files sound.

We find that the PCA dimension reduction technique with 3 PCA components, with K-Means clustering and k=3 clusters, achieved the highest performing clustering results.

Feature Extraction and Generation

We distinguish the audio files based on what boils down to seven feature groups obtained via Librosa packages. Many of these feature groups focus on analyzing timbre, which deal with the type of instrument being used. Timbre can be analyzed using the following features. Mel-Frequency Cepstrum Coefficients (MFCC) are the coefficients that make up the Mel-frequency cepstrum, which visualizes a transformation of frequency and loudness. Spectral Centroids relate to the center of mass of the spectrum, or how high the frequency is on average. Zero crossing rate is an indication of when the signal changes signs, and can be useful for recognizing speech. Spectral Rolloff is another power spectral function that is useful for detecting speech. Speech shows higher concentration of energy in lower frequency bands, while no speech shows the opposite. Spectral contrast is a measure of the distribution between the spectral peaks and troughs.

Some other features that were used: Chroma Energy Normalized (CENS) analyzes the 12 pitches along with their octave shifts of the song. Tempo is the beats per minute of the song. After compiling a dataframe of these 74 features with all 90 audio files, the data was standardized by subtracting each feature with its mean, and dividing by its standard deviation.

https://github.com/subho406/Audio-Feature-Extraction-using-Librosa/blob/master/Song%20Analysis.ipynb

Dimension Reduction

The primary dimension reduction techniques used in the course of this project were: principal component analysis (PCA), multidimensional scaling (MDS), and locally linear embedding (LLE).

Dimension reduction was used to visualize our data as well as to analyze both the performance of individual clustering methods and to tune the hyperparameters of each clustering method. While the dimension can be reduced to any arbitrary size smaller than the initial dimension of the features space, for visualization purposes it is useful to select 2 or 3 dimensions in order to construct a scatter plot visualization of the data and cluster assignments. We decided to mainly explore various dimension reduction methods to avoid the curse of dimensionality due to the high-dimensional feature space that we had obtained: we have 90 observations and 74 features.

PCA:

To tune the number of PCA components we use a scree-plot that identifies the amount of variance that every additional principal component contributes to our data.
Based on the elbow of the scree plot, we identified 5 principal components to sufficiently capture the 'essence' of our data. I.e. every additional principal component after 5 does not explain a significant amount of variance. *See Figure 1.1*

MDS:

To tune the number of MDS components, we created stress plots as well as a quasi-scree plot that calculates the stress value over a range of MDS number of component values. Similar to the scree plot to determine the optimal number of principal components, the number of dimensions to select for MDS is determined by an elbow in the stress plot.
Based on this criteria, *we identified 5 MDS components as the optimal number of components*. See Figure 1.2

Locally Linear Embedding:

Description: LLE is a manifold learning method that approximates the manifold structure of our data by first approximating the local geometry of our data to then be used to approximate the global embedding of the data.

Tuning:

To tune the number of LLE components, we chose to plot the scatter plot of LLE reduced data for LLE_components = [2,20], a low number of neighbors to approximate the local geometric structure of the data. Out of those values, LLE_components: 2,8,12 appeared to perform the best: creating the most separated and tight clusters out of the range of values.
Based on this criteria, *we identified 12 components as the optimal number of components*. See Figure 1.3

Clustering

K-Means:

Description: The K-means algorithm initializes k different cluster centroids and assigns every datapoint to the closest centroid. The centroid is then updated to the euclidean mean of all data points assigned to its associated cluster. Then, again, the data points are assigned to the closest centroid. This process iterates until the cluster centroid locations converge.

Tuning: To tune k, we created dendrograms (hierarchical clustering) using two methods: (i) Ward's method and (ii) Complete linkage on our dimension-reduced data, and identified suitable distances to 'cut' our dendrograms.

For PCA data (with 5 PCA components), we identified k= 3 or 4 emergent clusters from dendrograms and cost plot: See Figure 2.1

For MDS data (5 components), we identified k = 4 clusters. See Figure 2.2

For LLE data (with 12 LLE neighbors and 2 LLE components), we identified k= 3 emergent clusters. See Figure 2.3

Density-based Spatial Clustering Algorithm with Noise (DBSCAN):
Description: DBSCAN is a popular clustering algorithm that can be used to detect clusters of arbitrary shape based on the regional density of datapoints. Regions of a high density (within a given neighborhood of datapoints), are clustered together.

Tuning: We assume that the Min-points hyperparameter (associated with density requirement) is 2 times the dimensions in our reduced data (E.g. For PCA reduced data, with 5 principal components, we have Min-Points=10.). The epsilon value is the max distance allowed between two points, and is approximated by visualization. We used knn to create a plot of sorted observations with their distances, and found epsilon based on reasonable closeness (elbow).
See Figure 2.4 for PCA, MDS, LLE DBSCAN tuning

Evaluation and Assessment

To evaluate the different clustering methods, we obtained the silhouette scores.
PCA (5 components, k = 3 | k-means): 0.3203
PCA (5 components, k = 4 | k-means): 0.2856
MDS (5 components, k = 3| k-means): 0.2373
MDS (5 components, k = 4| k-means): 0.2252
LLE (12 components, k = 3| k-means): 0.5322
PCA (5 components, eps = 6.5 | DBSCAN): 0.3198
MDS (5 components, eps = 7 | DBSCAN): 0.2301
LLE (12 neighbors, 2 components, eps = 0.05 | DBSCAN): 0.2830
See Figure 3

Conclusions

MDS and PCA dimension reduction with 5 components gave very similar groupings when put into 3 clusters. With the exception of track 23 and 58, the clustering for MDS and PCA are identical. This would suggest that MDS and PCA give similar reduced matrices. But we decided to proceed with MDS clustering at 4 groups, based on the dendrograms.

We obtained the top five loadings of the PCA reductions (See Figure 4). We see that most of the important features gravitated towards groups that decided timbre, which distinguishes the type of instrument, and the presence of voice.

One aspect to consider in the future is selectivity of features. The features we used were all statistics of the features from librosa. We could try removing features to see if we get better clusters. We could also use different interpretations of features; using maximum and minimum values of a feature instead of just the average and standard deviation. This might explain why our DBSCAN did not work so well with our data. We suspect that it did not work because our feature dissimilarities were not great enough. Since most of our features dealt with averages, we could have a lower dissimilarity than if we used maximums and minimums.

Appendix

Figure 1.1: Scree plot PCA components determination

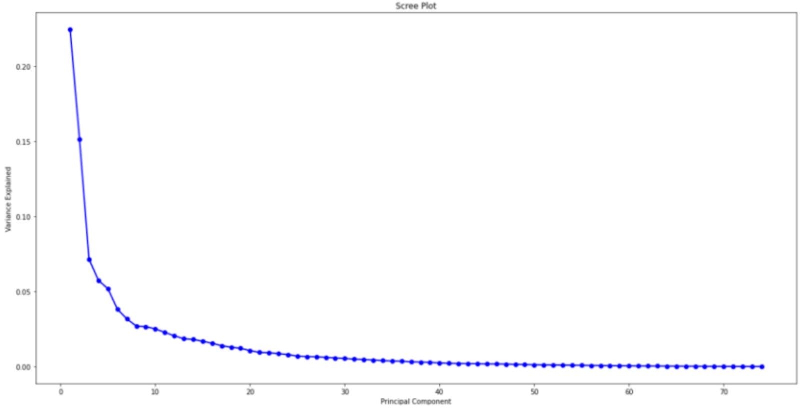


Figure 1.2: Stress plot MDS components determination

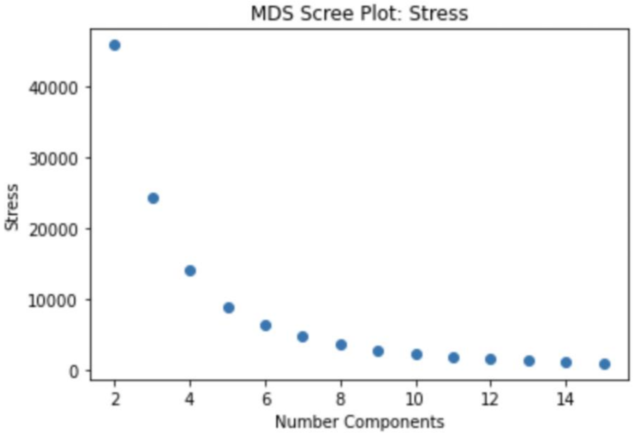


Figure 1.3: LLE

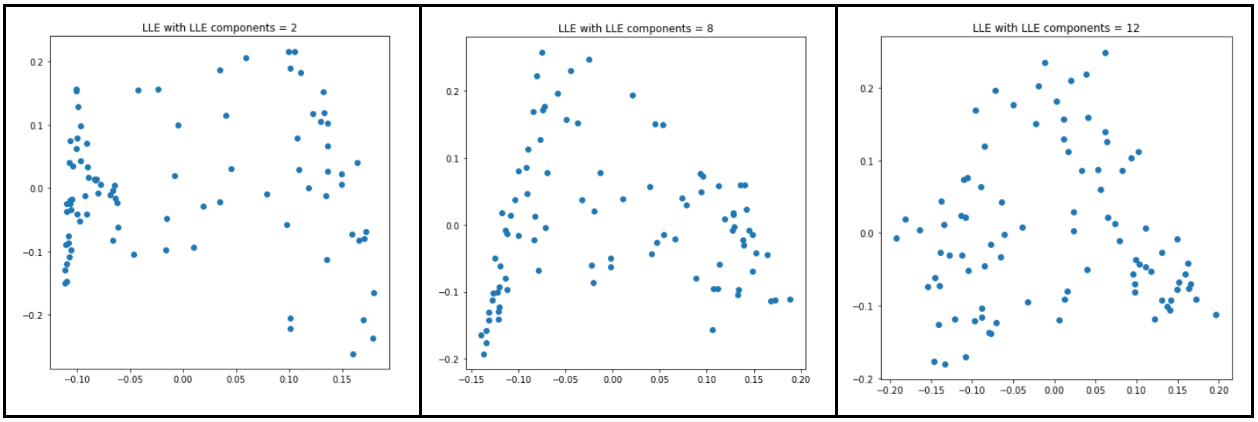


Figure 2.1 PCA cluster approximation

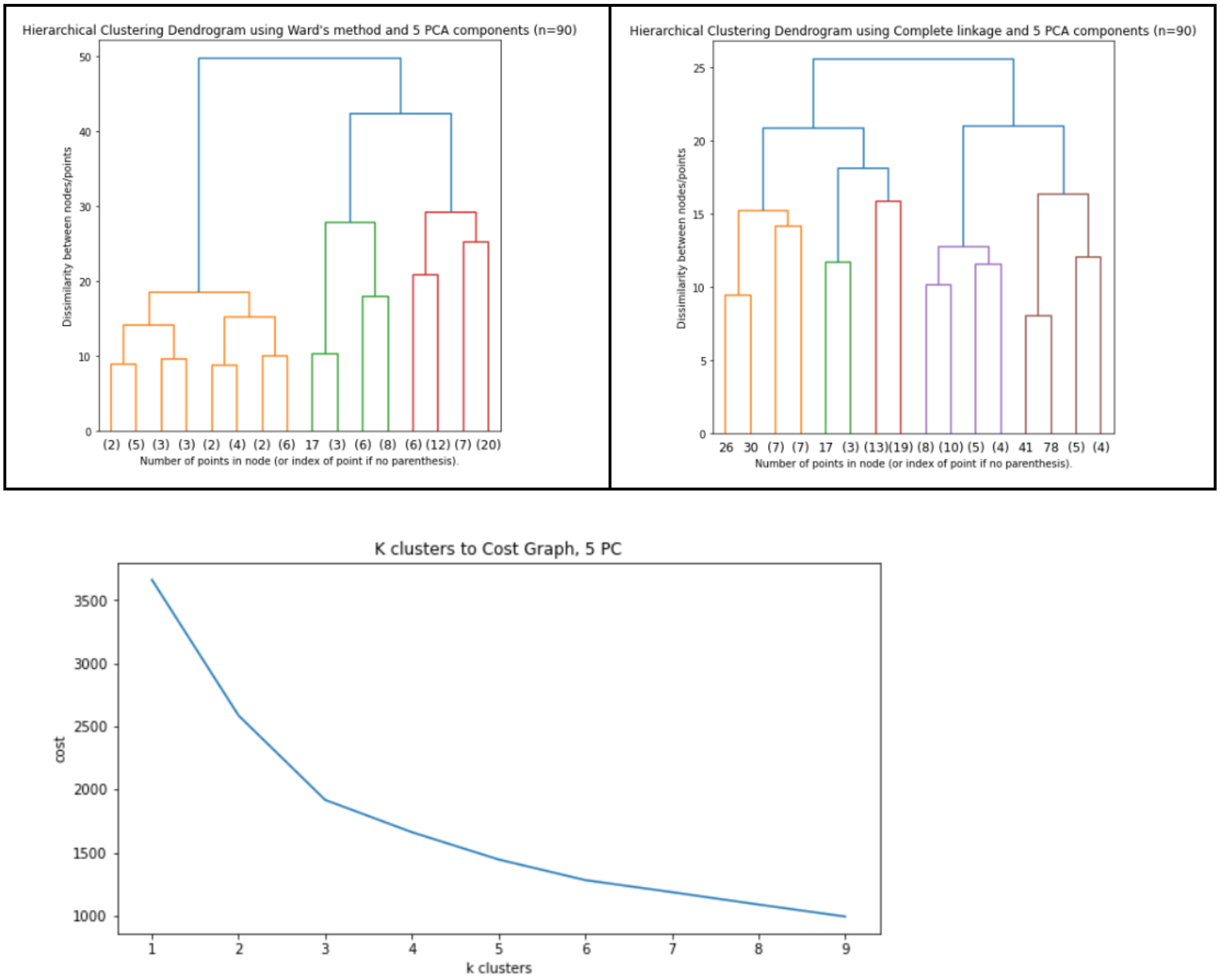
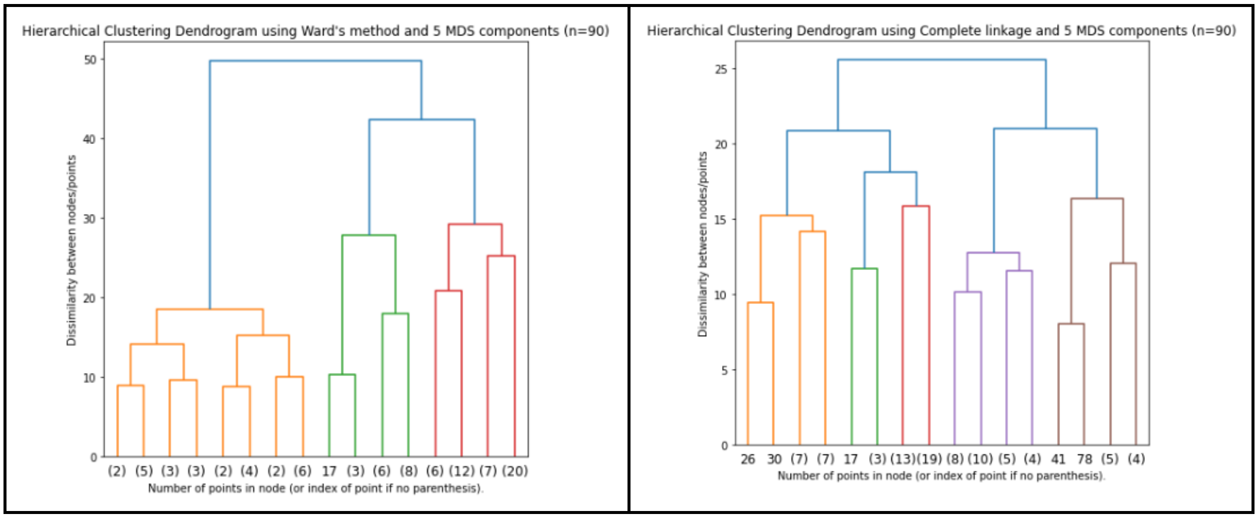


Figure 2.2: MDS Cluster approximation



For MDS data (with 5 MDS components), we identified k= 3 or 4 emergent clusters:

Figure 2.3: LLE cluster approximation

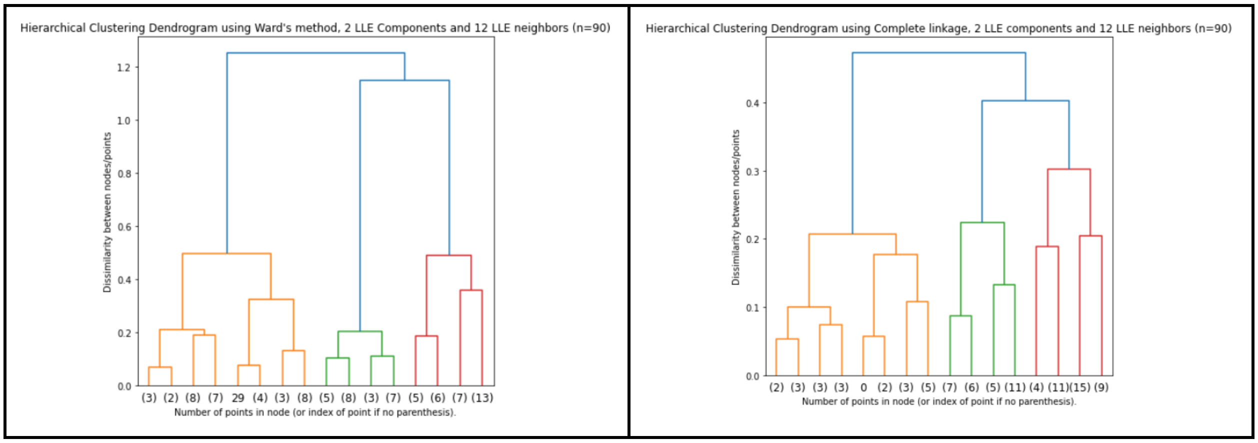
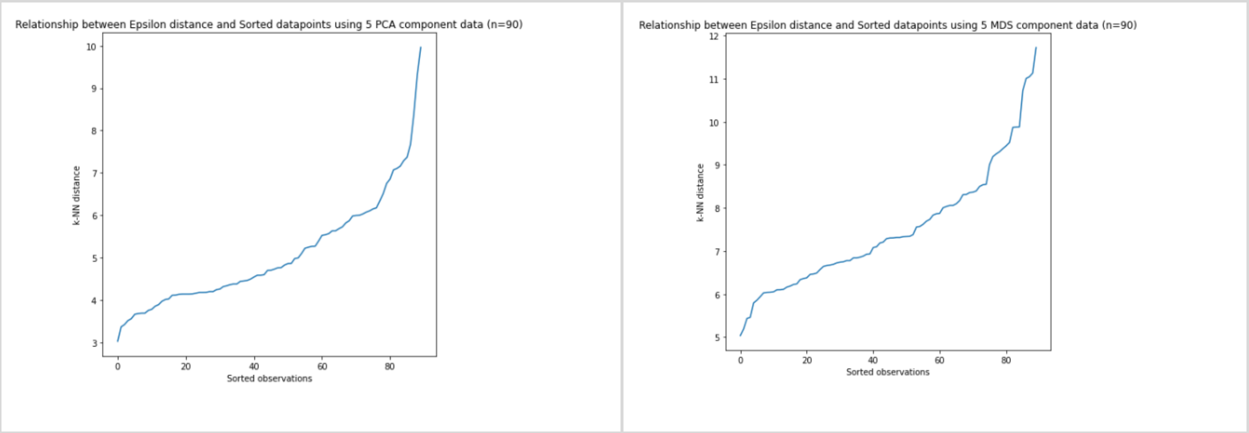


Figure 2.4 Tuning DBSCAN parameters



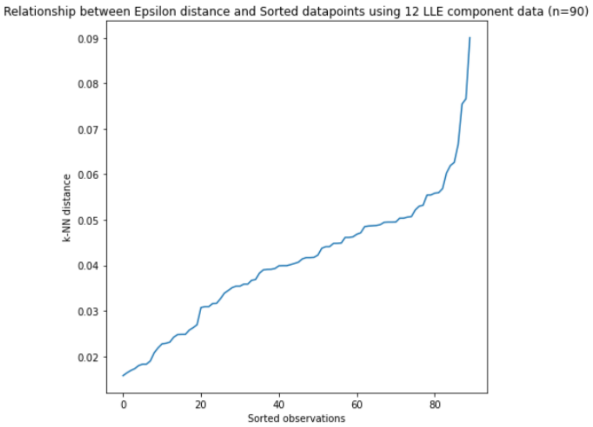
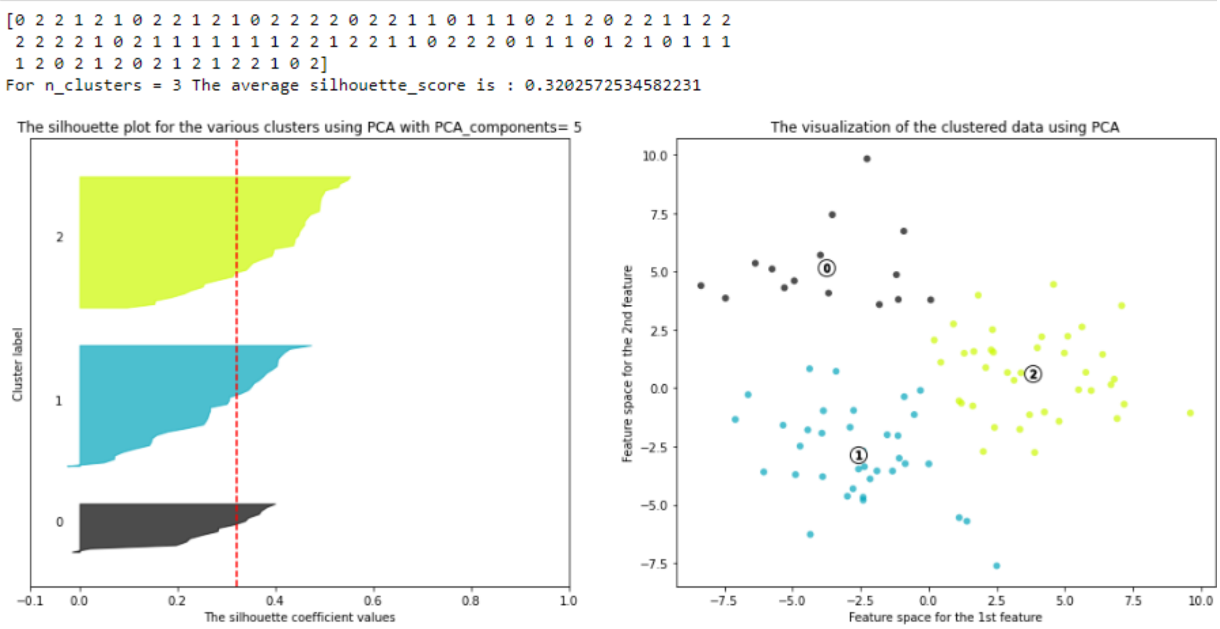
For PCA data (with 5 PCA components), we identified $\varepsilon = [6,6.5]$	For MDS data (with 5 MDS components), we identified $\varepsilon = [7,8.5]$
	
For LLE data (with 12 LLE neighbors and 2 LLE components), we identified $\varepsilon = [0.05,0.06]$	

Figure 3: Silhouette scores for all clustering methods done

PCA components=5, k=3 | k-means

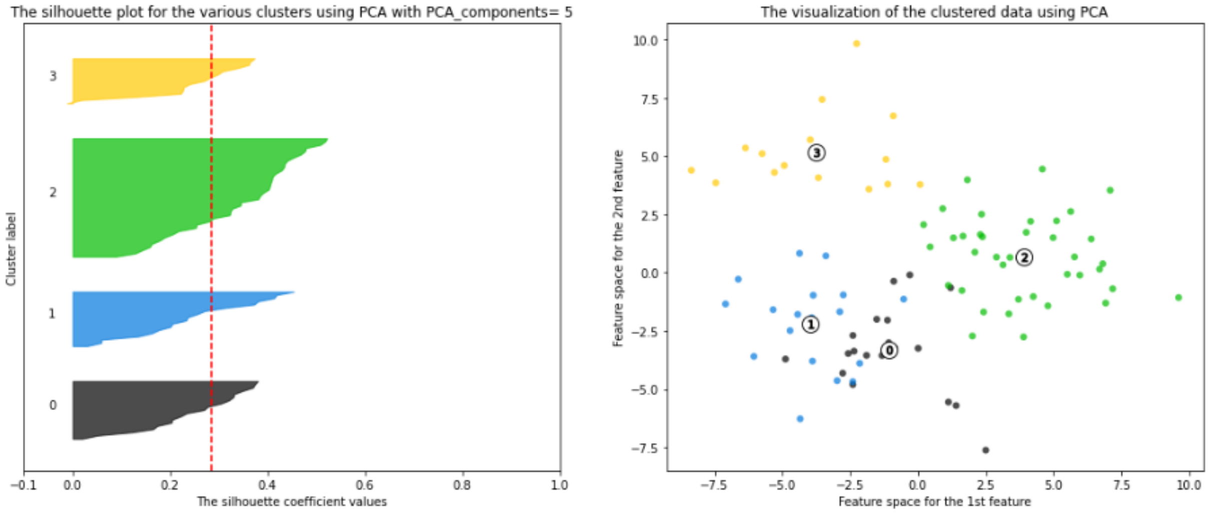


```
[0 2 2 1 2 1 0 2 2 1 2 1 0 2 2 2 2 0 2 2 1 1 0 1 1 1 0 2 1 2 0 2 2 1 1 2 2
 2 2 2 2 1 0 2 1 1 1 1 1 1 1 2 2 1 2 2 1 1 0 2 2 2 0 1 1 1 0 1 2 1 0 1 1 1
 1 2 0 2 1 2 0 2 1 2 1 2 2 1 0 2]
For n_clusters = 3 The average silhouette_score is : 0.3202572534582231
```

The silhouette plot for the various clusters using PCA with PCA_components= 5
Cluster label
2
1
0
−0.1
0.0
0.2
0.4
0.6
0.8
1.0
The silhouette coefficient values
The visualization of the clustered data using PCA
Feature space for the 2nd feature
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
0
1
2
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Feature space for the 1st feature

PCA components=5, k=4 | k-means

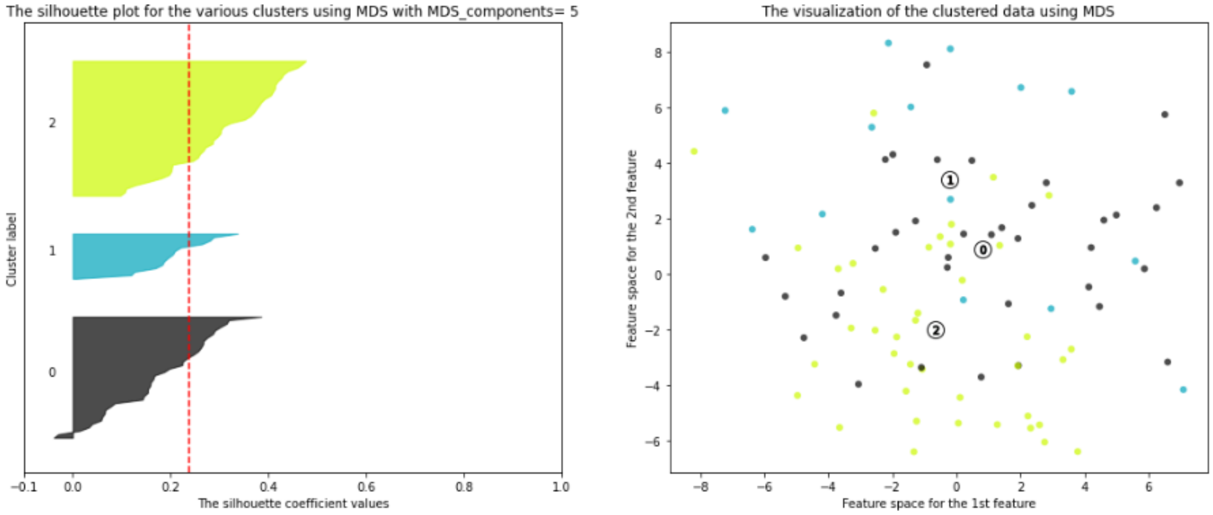
```
[3 2 2 0 2 1 3 2 2 0 2 1 3 2 2 2 2 3 2 2 1 0 3 0 1 1 3 2 1 2 3 2 2 0 0 2 2
 2 2 2 2 0 3 0 1 1 0 1 0 1 0 2 2 0 2 2 0 0 3 2 2 2 3 0 1 1 3 0 2 0 3 1 1 0
 1 2 3 2 0 2 3 2 1 2 1 2 2 1 3 2]
For n_clusters = 4 The average silhouette_score is : 0.2855904206117958
```


The silhouette plot for the various clusters using PCA with PCA_components= 5

The visualization of the clustered data using PCA

MDS components=5, k=3 | k-means

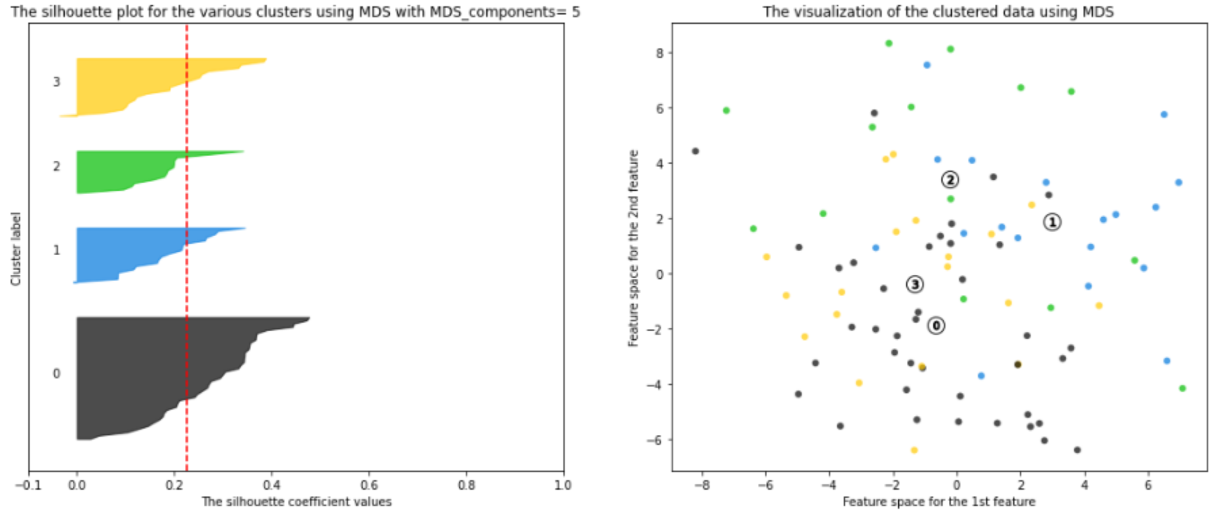
```
[1 2 2 0 2 0 1 2 2 0 2 0 1 2 2 2 2 1 2 2 0 0 1 0 0 0 1 2 0 2 1 2 2 0 0 2 2
 2 2 2 2 0 1 2 0 0 0 0 0 0 0 2 2 0 2 2 0 0 2 2 2 2 1 0 0 0 1 0 2 0 1 0 0 0
 0 2 1 2 0 2 1 2 0 2 0 2 2 0 1 2]
For n_clusters = 3 The average silhouette_score is : 0.23729356036691351
```


The silhouette plot for the various clusters using MDS with MDS_components= 5

The visualization of the clustered data using MDS

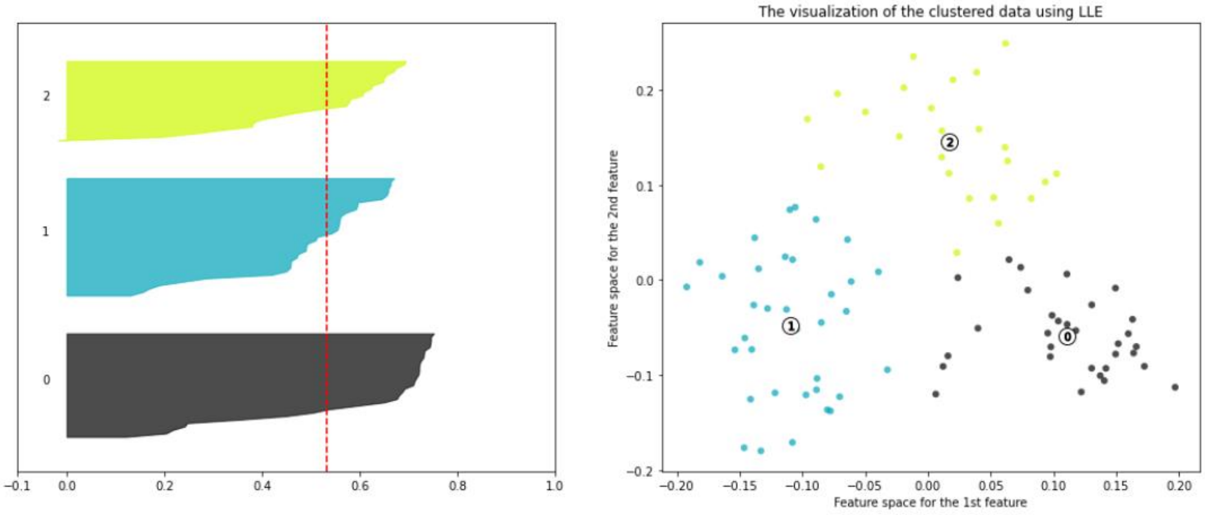
MDS components=5, k=4 | k-means

```
[2 0 0 1 0 3 2 0 0 1 0 3 2 0 0 0 0 2 0 0 3 1 2 1 3 3 2 0 3 0 2 0 0 1 1 0 0
 0 0 0 0 1 2 0 3 3 1 3 1 3 1 0 0 1 0 0 1 1 0 0 0 0 2 1 3 3 2 1 0 1 2 3 3 1
 3 0 2 0 1 0 2 0 3 0 3 3 0 3 2 0]
For n_clusters = 4 The average silhouette_score is : 0.22522593350705114
```



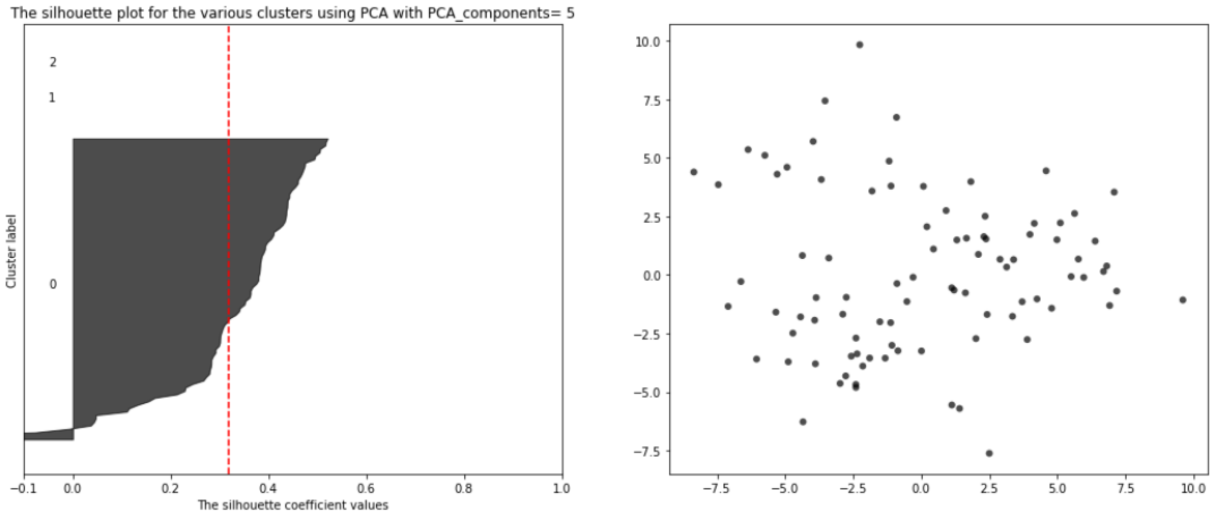
LLE components = 12, k=3 | k-means

```
[2 0 0 1 2 1 2 0 0 1 0 1 2 0 0 2 0 2 0 0 1 1 2 1 1 1 2 0 1 0 2 0 0 1 1 0 0
 0 2 2 2 1 2 0 1 1 1 1 1 2 1 2 0 1 0 0 1 1 2 0 0 0 2 1 1 1 2 1 0 1 2 1 1 1
 1 0 2 0 1 0 2 2 1 0 1 0 0 1 2 2]
For n_clusters = 3 The average silhouette_score is : 0.5322204589761099
```



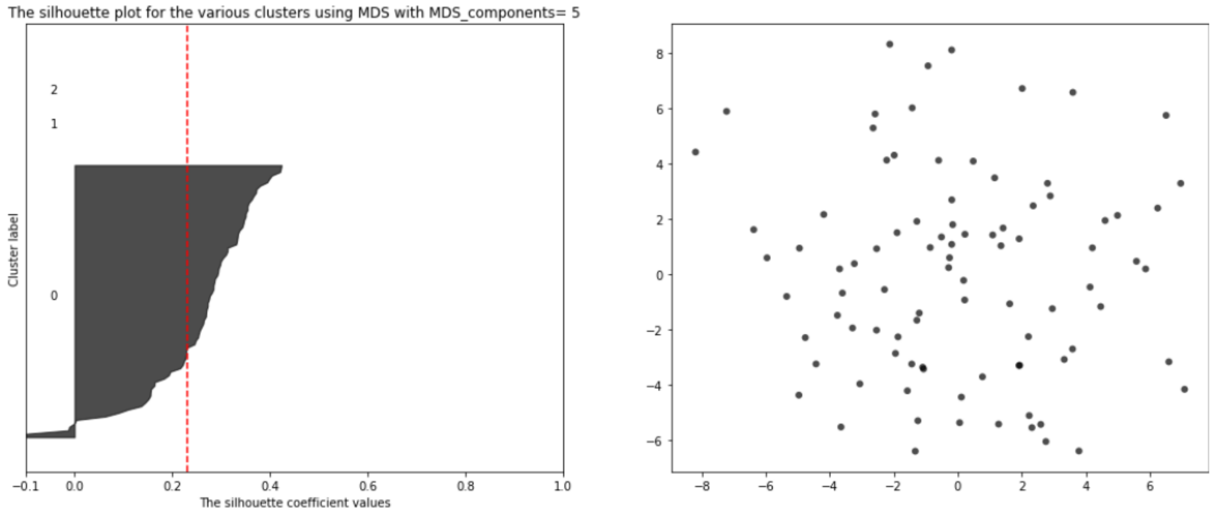
PCA components = 5, DBSCAN_eps = 6.5

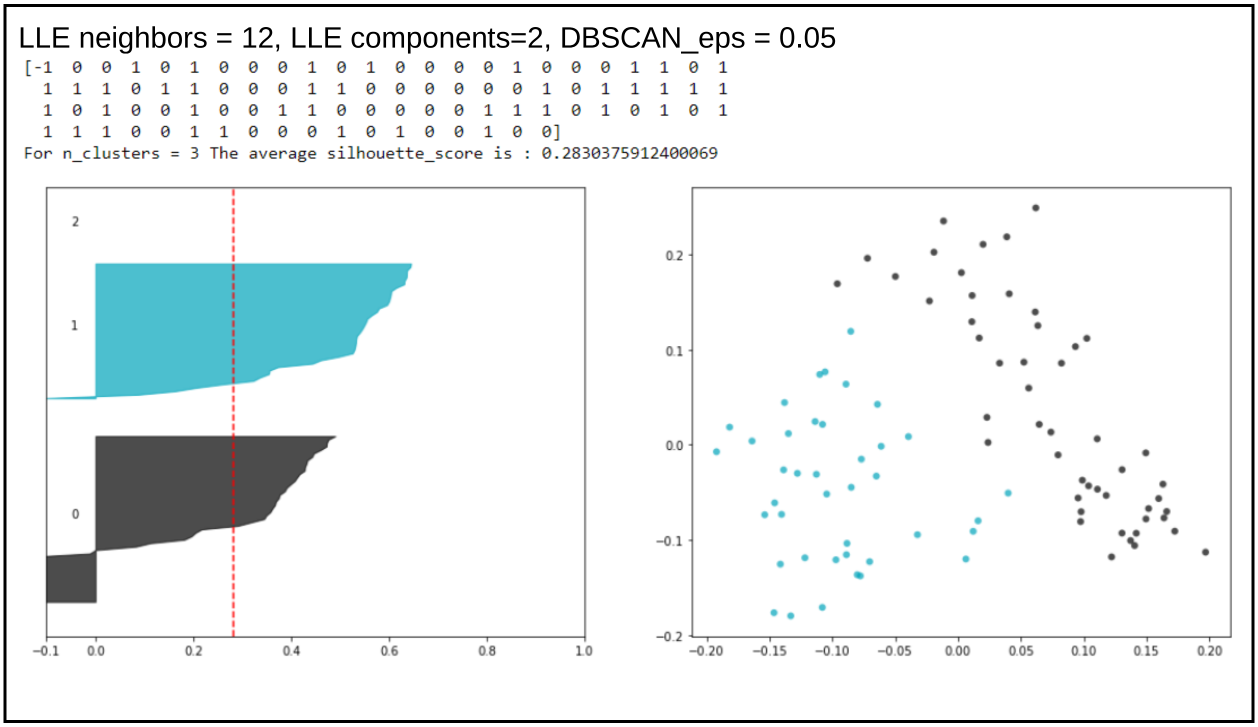
```
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
For n_clusters = 3 The average silhouette_score is : 0.3197660058818096
```



MDS components= 5, DBSCAN_eps = 7

```
[ 0  0  0  0  0  0 -1  0  0  0  0  0 -1  0  0  0  0 -1  0  0  0  0  0  0
  0  0 -1  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1  0
  0  0  0  0 -1  0 -1  0 -1  0  0  0  0  0  0  0  0  0]
For n_clusters = 3 The average silhouette_score is : 0.23014755334758594
```

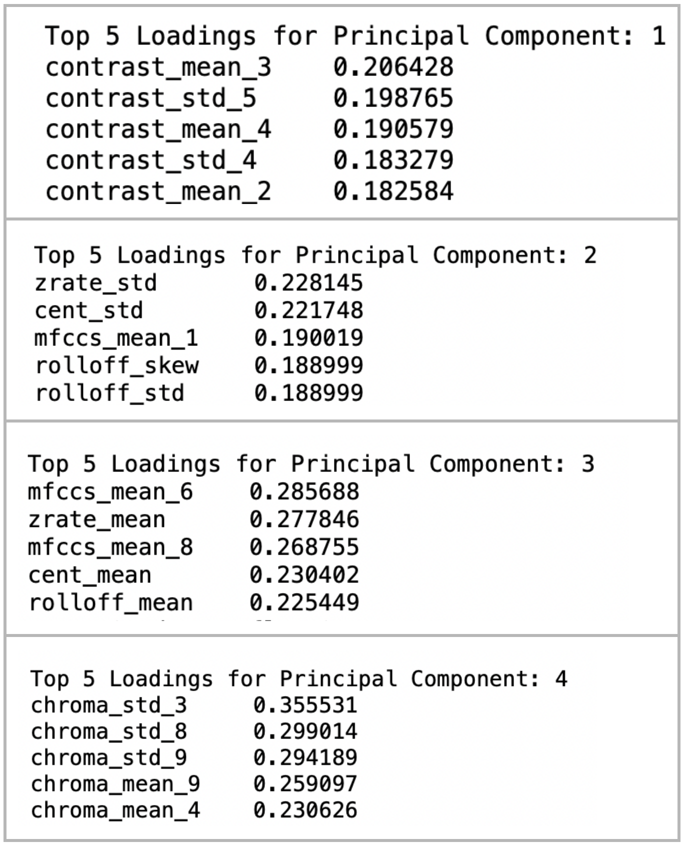




LLE neighbors = 12, LLE components=2, DBSCAN_eps = 0.05

```
[-1  0  0  1  0  1  0  0  0  1  0  1  0  0  0  0  1  0  0  0  1  1  0  1
  1  1  1  0  1  1  0  0  0  1  1  0  0  0  0  0  0  1  0  1  1  1  1  1
  1  0  1  0  0  1  0  0  1  1  0  0  0  0  0  1  1  1  0  1  0  1  0  1
  1  1  1  0  0  1  1  0  0  0  1  0  1  0  0  1  0  0]
For n_clusters = 3 The average silhouette_score is : 0.2830375912400069
```

Figure 4: Factor loadings for PCA

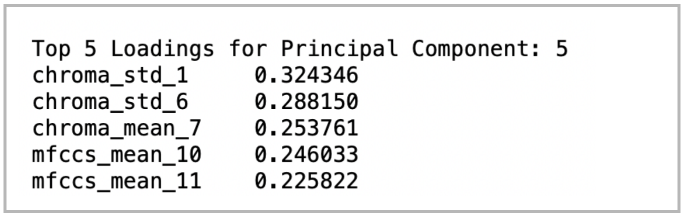


```
Top 5 Loadings for Principal Component: 1
contrast_mean_3    0.206428
contrast_std_5     0.198765
contrast_mean_4    0.190579
contrast_std_4     0.183279
contrast_mean_2    0.182584
```

```
Top 5 Loadings for Principal Component: 2
zrate_std        0.228145
cent_std         0.221748
mfccs_mean_1     0.190019
rolloff_skew     0.188999
rolloff_std      0.188999
```

```
Top 5 Loadings for Principal Component: 3
mfccs_mean_6    0.285688
zrate_mean      0.277846
mfccs_mean_8    0.268755
cent_mean       0.230402
rolloff_mean    0.225449
```

```
Top 5 Loadings for Principal Component: 4
chroma_std_3     0.355531
chroma_std_8     0.299014
chroma_std_9     0.294189
chroma_mean_9    0.259097
chroma_mean_4    0.230626
```



```
Top 5 Loadings for Principal Component: 5
chroma_std_1     0.324346
chroma_std_6     0.288150
chroma_mean_7    0.253761
mfccs_mean_10    0.246033
mfccs_mean_11    0.225822
```