STA 142B Final Project
Group 7
Jiaxu He
David Mao
Ryan Buchner

Introduction

Nowadays the development of digital content has increased the demand for digital music. Thus, how to classify the genre of these massive amounts of digital music by their similarity in an efficient way has become very important. In this project, we will extract the features from each music file and build the unsupervised machine learning model that could classify these music by their features. The dimensional reduction and clustering methods we use in this project include: Principal Component Analysis(PCA), Kernel PCA, Locally Linear Embedding(LLE), Isomap, K Means Clustering, and Spectral Clustering. The following outlines our process for clustering the music dataset, which consists of 90 songs from several different genres.

Methodology

We utilized the function MidTermFeatures.directory_feature_extraction() from the package pyAudioAnalysis to extract 138 different features from the song data. We experimented with selecting only certain features but found that the best clustering occurred when using all of them and letting the model decide on importance. The features present from this extraction included zero crossing rate, which measures the number of times the signal changes from positive to negative or vice versa; it's roughly a measure of signal noisiness[2]. Other features included the Mel-frequency cepstral coefficients (MFCC), which help capture the shape of the audio waves over the short term windows. In addition, we collected the chroma features which describe the information pertaining to a single pitch into single coefficients (12 coefficients, one for each octave). The chroma features and the MFCCs were collected across the song pieces, so we had access to how the different features varied across the dataset.

We found that the features needed to be scaled; otherwise, a few top features would dominate the analysis, namely beats per minute. Interestingly, clustering without scaling would lead to three very distinct clusters, because the clusters were driven by the BPM statistic, which consisted of 3 main BPMs. However, these groupings were not indicative of any larger pattern; we hypothesize that the clustering is a result of the fact that music is largely produced at certain distinct tempos and so BPM is actually a discrete variable more so than a continuous one.

The data pipeline for the best results obtained will follow what is suggested by Barreira et al. in their article “Unsupervised Music Genre Classification with a Model-Based Approach”, with some minor changes[1]. We first use the features as discussed previously, which is an expansion of the features used in the article. Next, like in the article, we will standardize each of our obtained features, then construct a similarity matrix using the standardized features. Our next steps now differ from the method used in the article. Since we have not discussed Model-Based Clustering Analysis (MBCA) in class, we will not be using the same method used in the article,

which was Principal Component Analysis and MBCA. We will use Isomap with two components to project our data onto a two dimension plane, then use K-means Clustering to attempt to cluster the resulting manifold. Finally, we will calculate the silhouette scores for our clusters, then plot a silhouette plot and a scatterplot visualizing our clusters. We will then use this to choose the optimal Isomap embedding, as well as the optimal number of clusters.

Analysis

Using Isomap on the similarity matrix obtained from the scaled features, we find that our data takes a circular shape, as shown in fig. 1. This poses some problems, namely the clusters found in this embedding are likely to not be convex shaped, meaning that K-means might not give the best clusters and the silhouette score may not be the best indicator of good clusters. However, using K-means still gave better clusters than the other methods. The manifold does not exhibit characteristics that would indicate a density based clustering function would work well. The data does suggest that a kernel based method would work well. However, our attempts at doing spectral clustering and Kernel PCA with a RBF kernel yielded worse results than K-means, due to the difficulty of choosing a proper gamma for the RBF kernel. In order to cluster using k-means, we have from our previous results some intuition that the number of clusters in our data is three, which we can use to fix k=3 in our K-Means algorithm. We choose n_neighbors=4 for Isomap at this value, the circular manifold shape emerges, and clustering this embedding gives a better silhouette score than with a large k. The resulting clusters when applying K-means with k=3 are shown in fig. 2, and the corresponding silhouette plot is shown in fig. 3. From the silhouette plot and visually, we can see that each of the clusters are good quality clusters. We are able to obtain an ARI score of 0.5798 with the cluster labels we found using this method.

Conclusion

Our clustering methodology described above provided a strong clustering according to the “true” labels. Although the ARI score is not particularly high, the data provided was noisy and not conducive to clustering. For instance, listening to a few songs in the dataset revealed that even though some of the songs seemed similar, most seemed to blend together. Our dimension reductions of the features we extracted confirmed the same pattern, as even when distinct clusters presented themselves, they were surrounded by noisy areas between dense clusters. A larger data set may have been more conducive to better clustering as it would have allowed for better deployment of clustering algorithms such as DBscan which require sufficiently dense datasets.

APPENDIX
Figures

Figure 1:

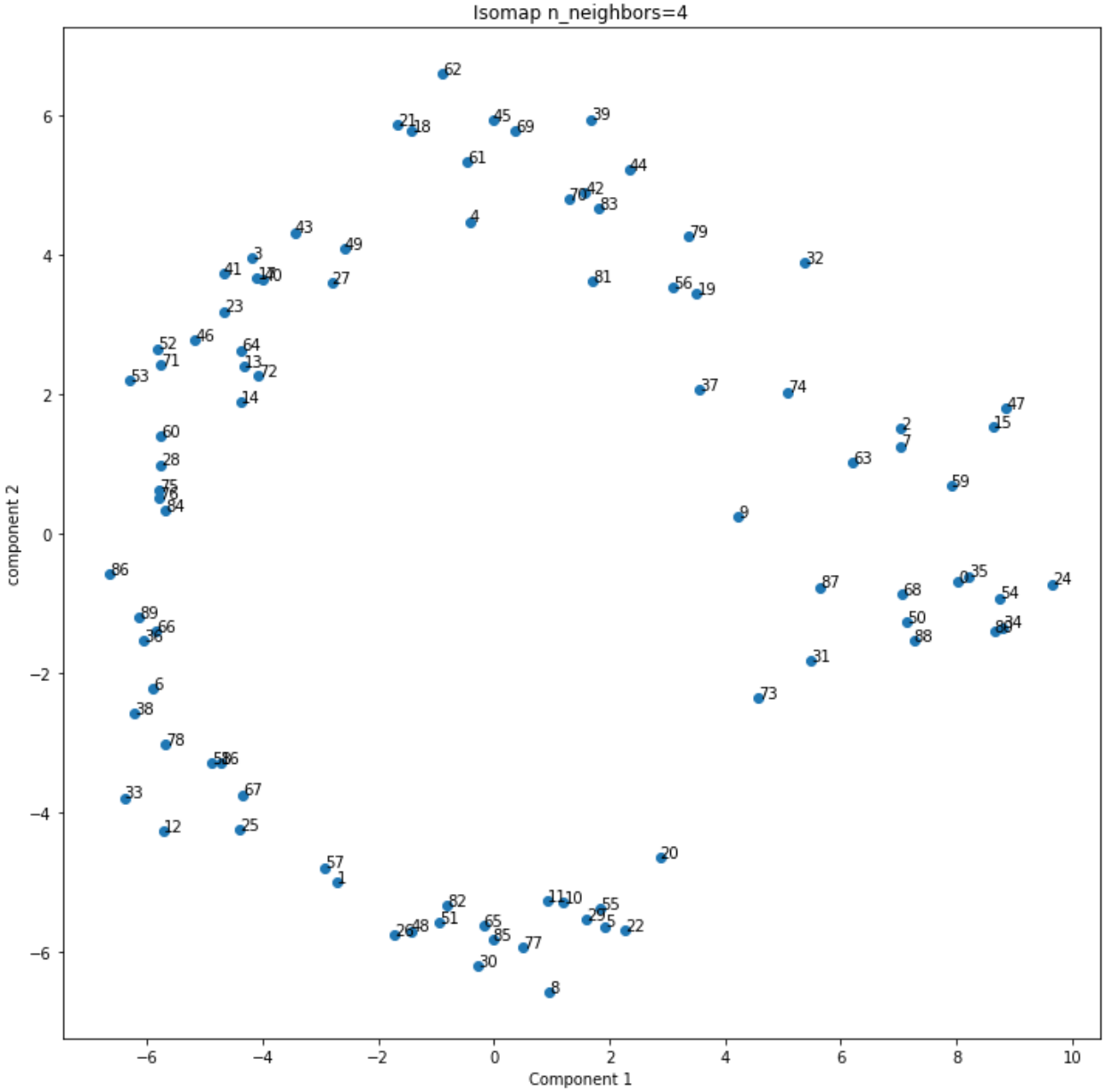


Figure 2:

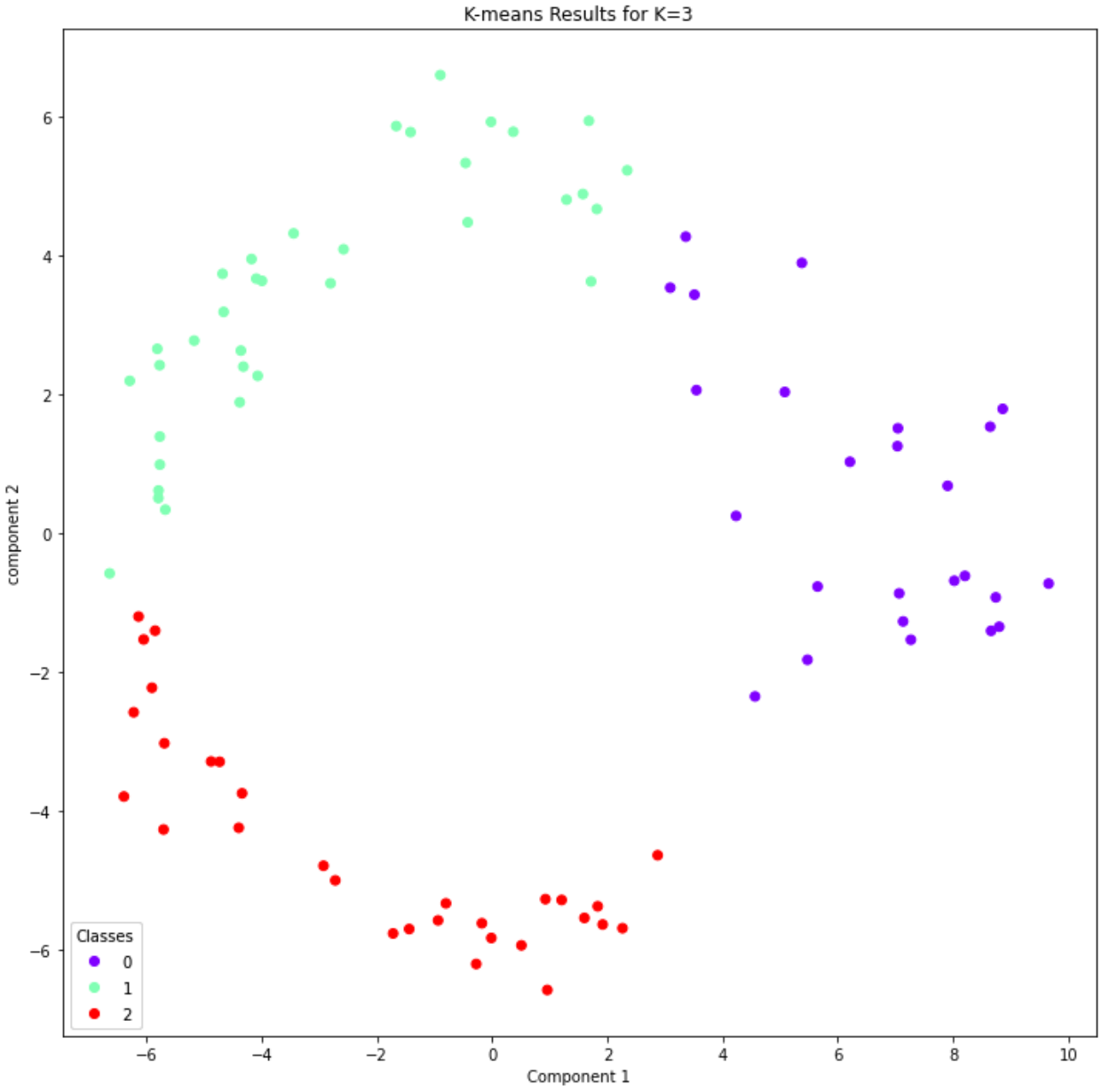
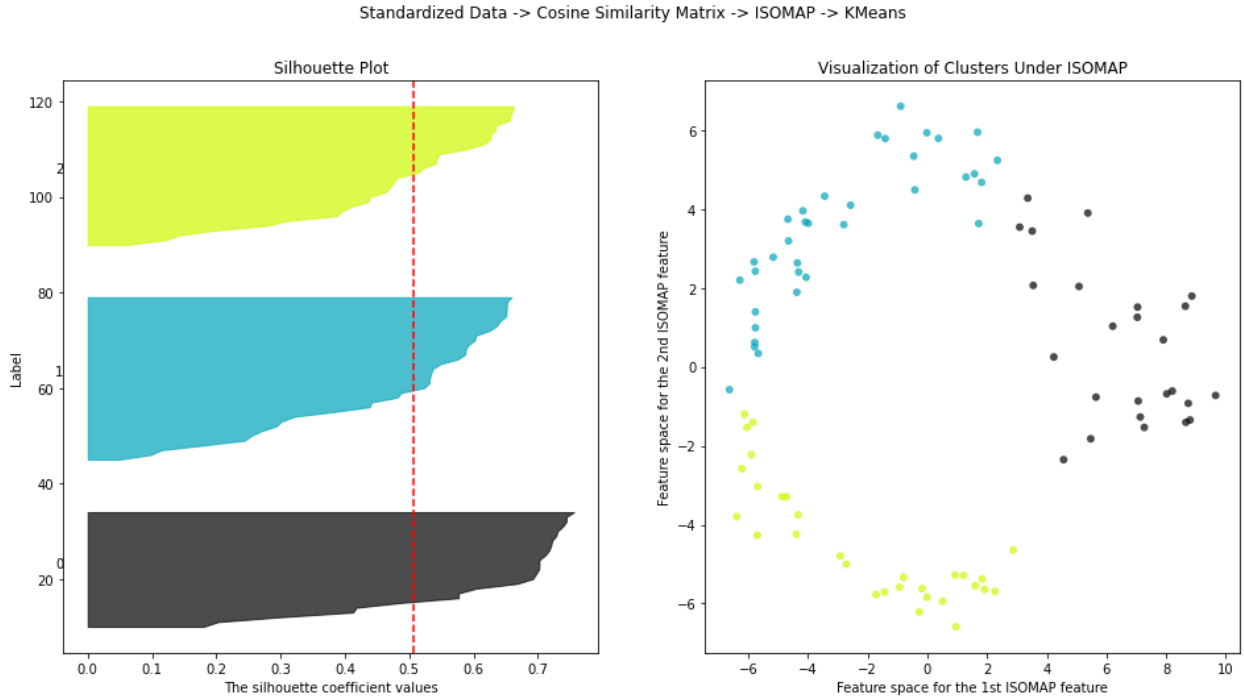
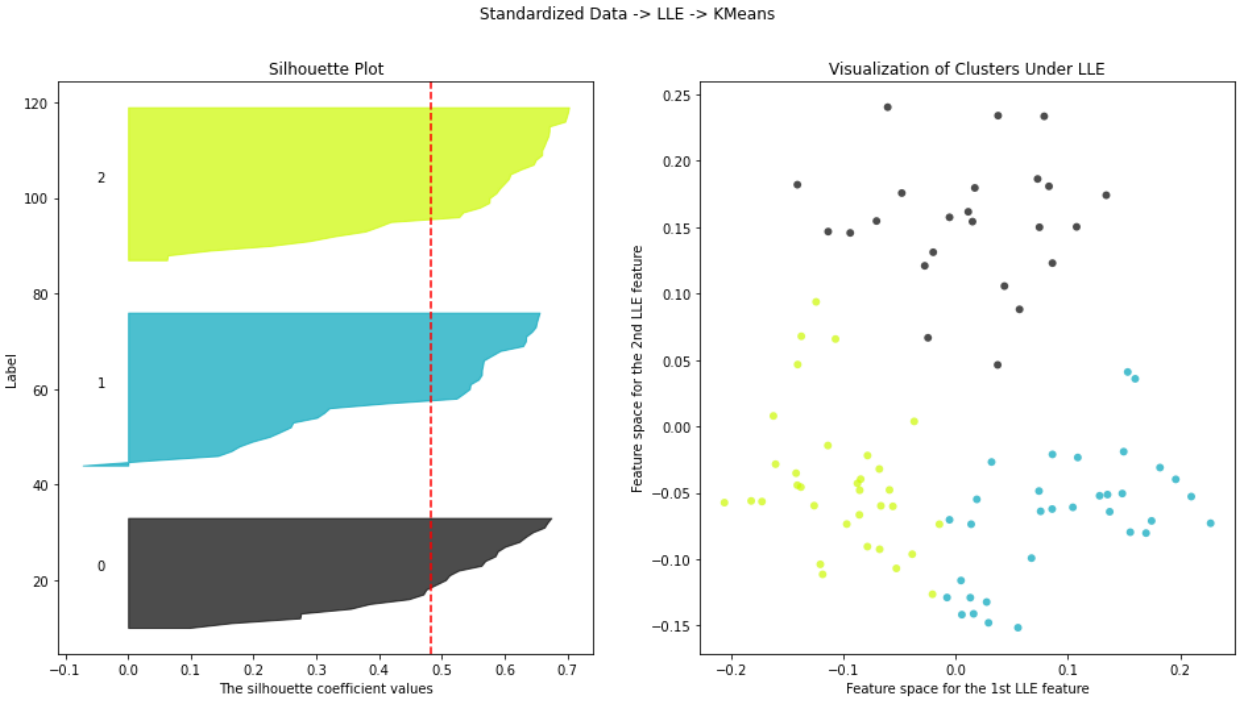

K-means Results for K=3
component 2
Component 1
Classes
0
1
2
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6
8
10

Figure 3:

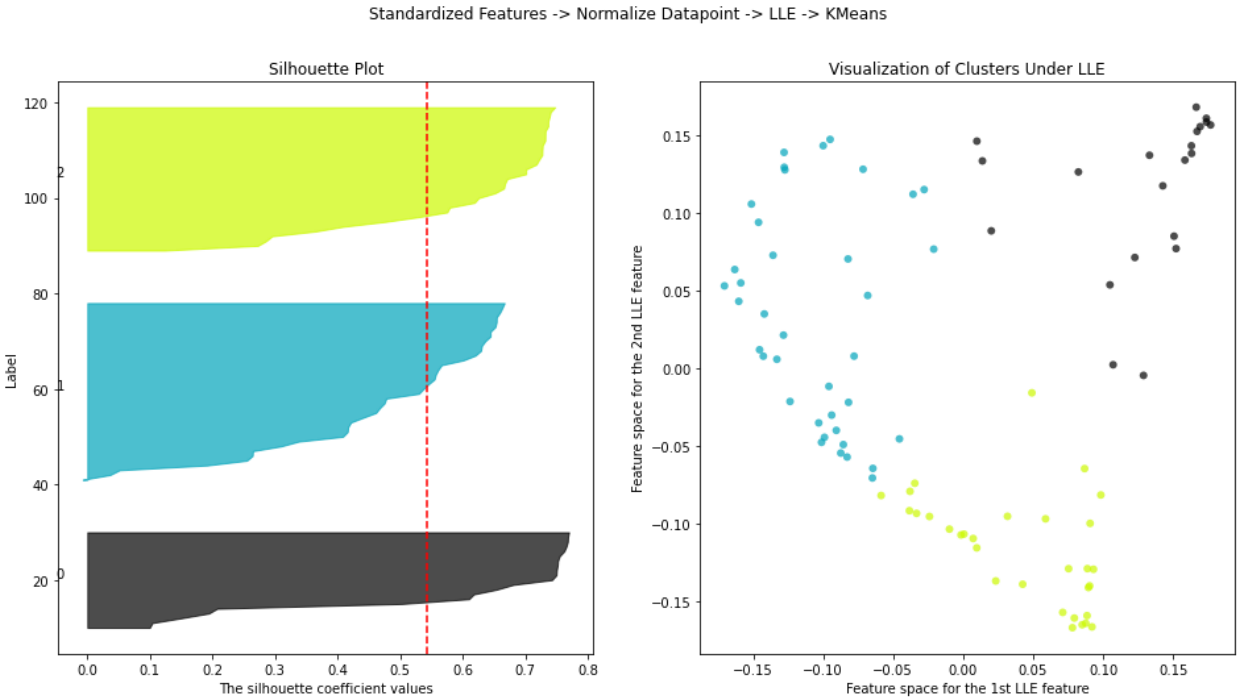

Standardized Data -> Cosine Similarity Matrix -> ISOMAP -> KMeans
Silhouette Plot
Label
The silhouette coefficient values
Visualization of Clusters Under ISOMAP
Feature space for the 2nd ISOMAP feature
Feature space for the 1st ISOMAP feature

Change Log (what changes we made to our process after each submission)
Our initial process only involved scaling the variables, however, after our initial submissions, we decided to normalize the data as well after scaling.

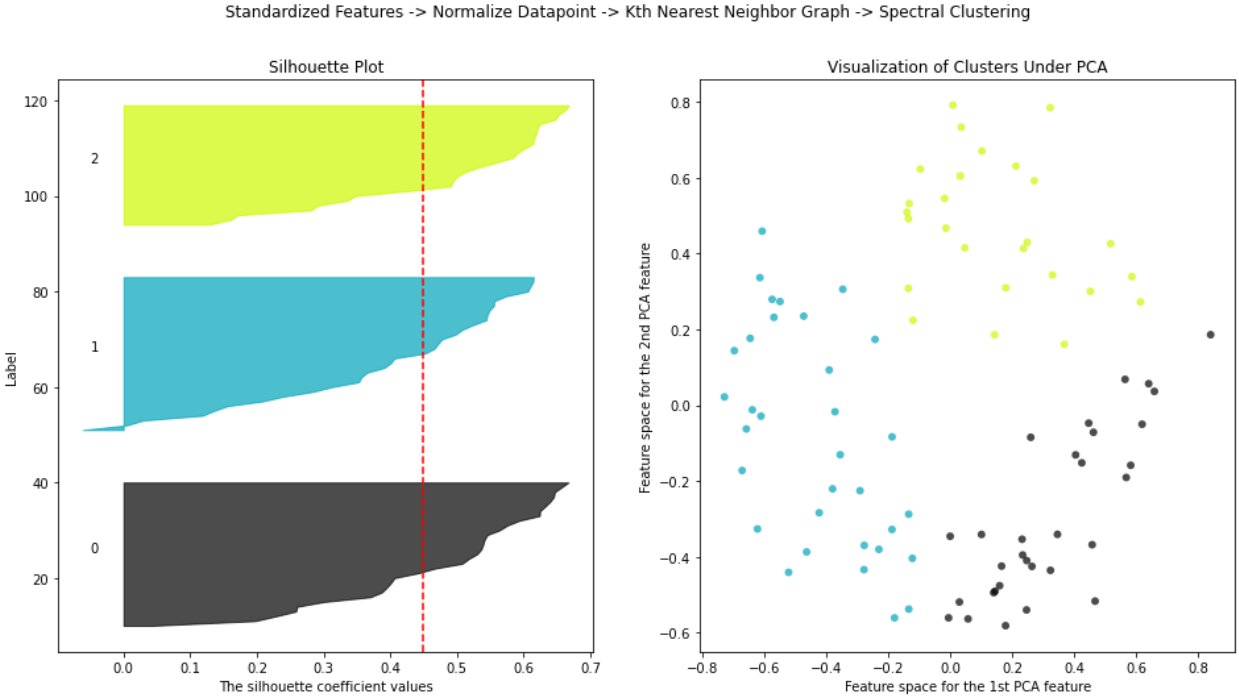
Attempt #1: ARI 40-50. For our first attempt, we look to see how a basic model using the features generated by PyAudioAnalysis's directory_feature_extraction function would do. We generate features using directory_feature_extraction, then standardize the features. We then use Local Linear Embeddings to visualize our data in two dimensions, with n_nieghbors=17 seemingly giving the best embedding. Finally, we use K-means with k=2, 3, 4, but found k=3 to have the best Silhouette score overall (0.483), high Silhouette scores within each cluster, and the best visually looking clusters. We now plan on using the features obtained from directory_feature_extraction with other clustering techniques to see if we can obtain better clusters.



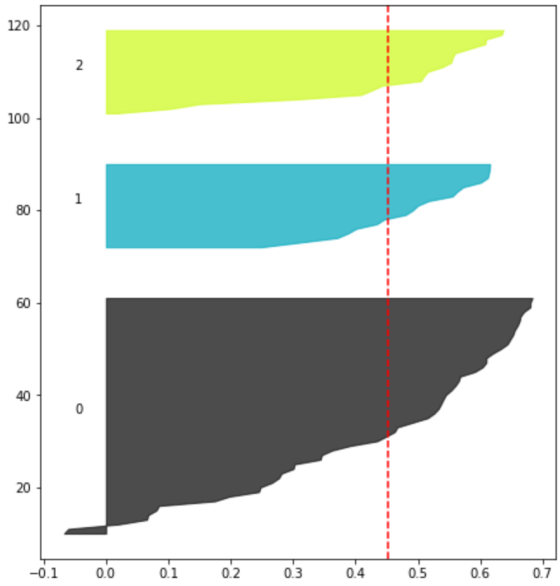
Attempt #2: ARI 40-50. For the next attempt, we see if adding the additional step of normalizing each datapoint after standardizing each feature will give better clusters. The pipeline will be the same as Attempt #1, except we will add a normalization step after standardizing the features. We find that LLE with n_neighbor=5 gives a clear embedding, and again K-means with k=3 gives the best overall Silhouette score (0.5417). Although each individual clusters are more imbalance, the overall score is higher than Attempt #1. However, the submission ARI fell into the same range. We next look into if other clustering techniques would work better.

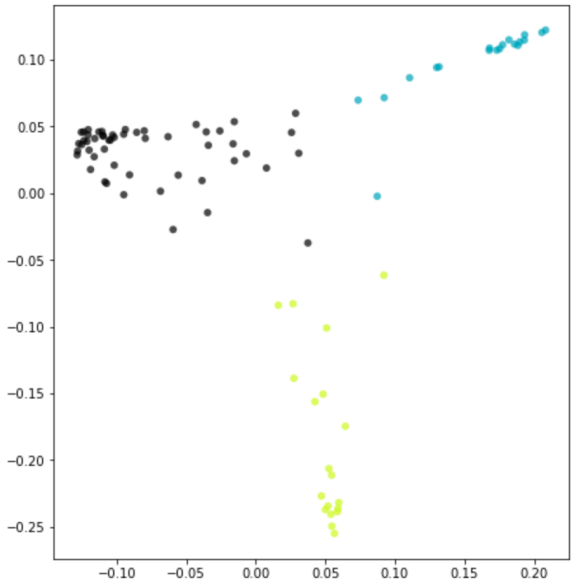


Attempt #3: ARI 40-50 For this attempt, we attempt to see if Spectral Clustering will give us a better result. We use the same standardized and normalized features and construct a Kth Nearest Neighbor Graph and perform spectral clustering. We visualize the results using PCA. We find that using n_neighbors=20 gives the best results. However, the results of using Spectral Clustering also gave an ARI score of between 40-50. With all our results falling in the same range, we will now look for other methods of extracting or processing our features that may give a better result.

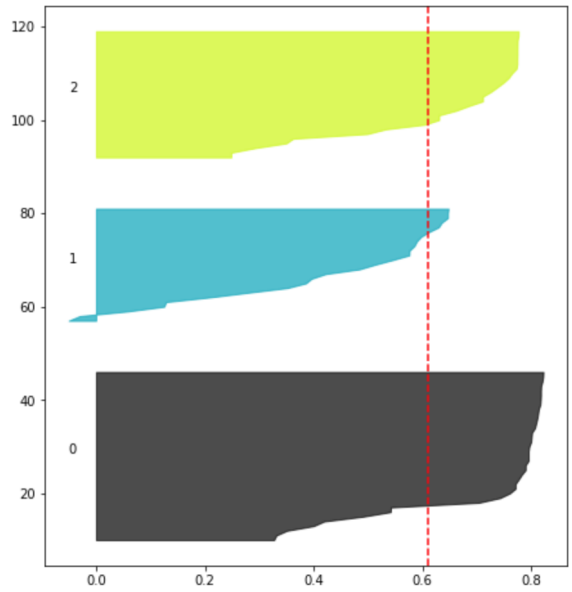


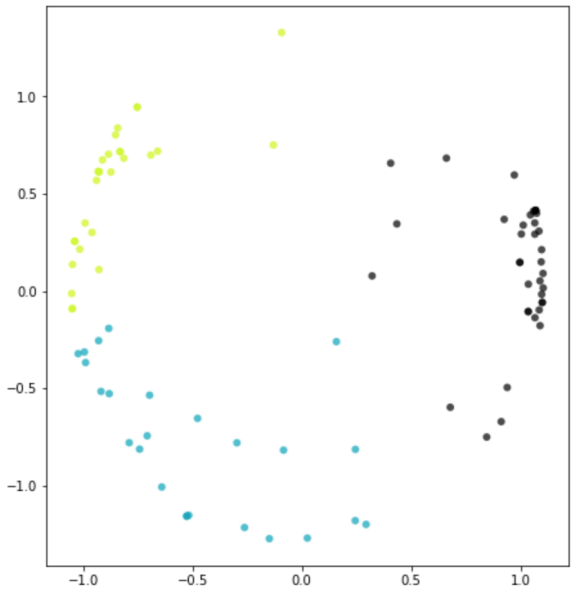
Attempt #4: ARI 30-40. For this attempt, we tried to see if clustering the correlation matrix of the Features matrix will give us a higher ARI score. We used the same standardized features and then calculated the correlation coefficient matrix of the standardized features matrix. After that, we performed the Locally Linear Embedding method and constructed a Kth Nearest Neighbor Graph and performed KMeans clustering. We found that when n_neighbors = 6 gives us the best results and the Silhouette score is high (0.712). However, the results of using KMeans clustering only gave an ARI score of between 30-40.



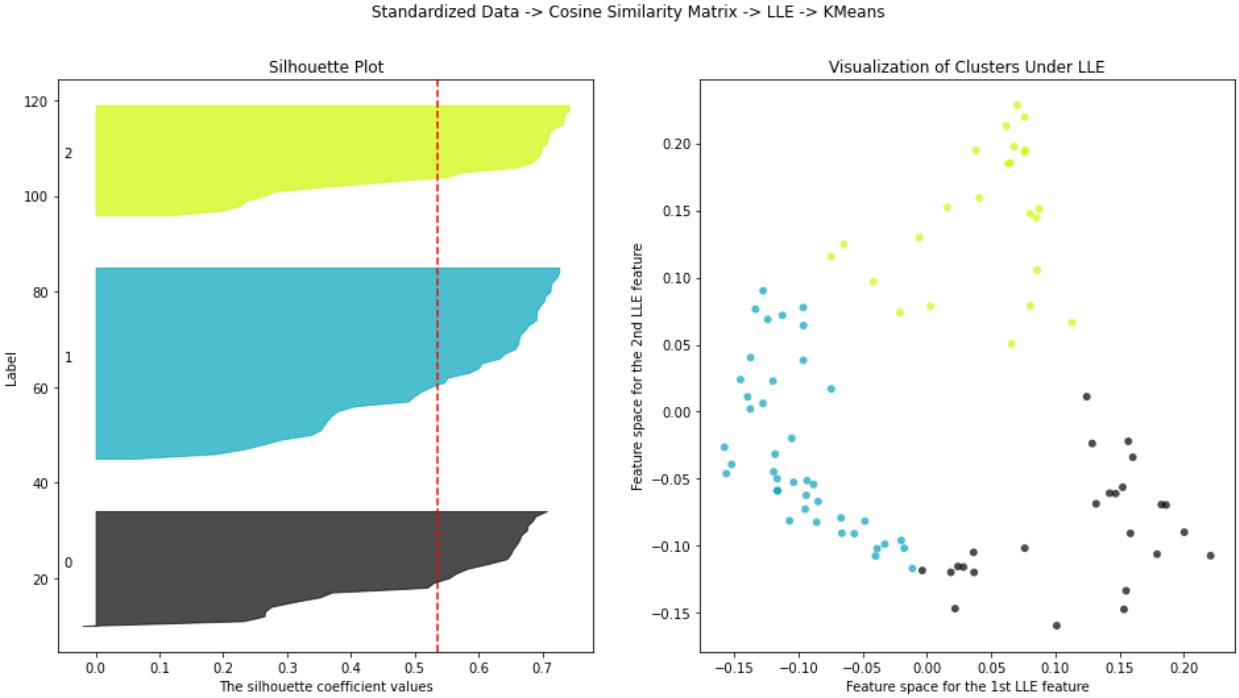


Attempt #5: ARI (20-30) For this attempt, we tried using only a subset of the features. That is, we only used the MFCC means and chroma means. Kernel PCA was applied twice to the scaled and normalized subset of the dataset, with a sigmoid kernel gamma of 5000. Kmeans is then used for the clustering of the data. The average silhouette score is .60.

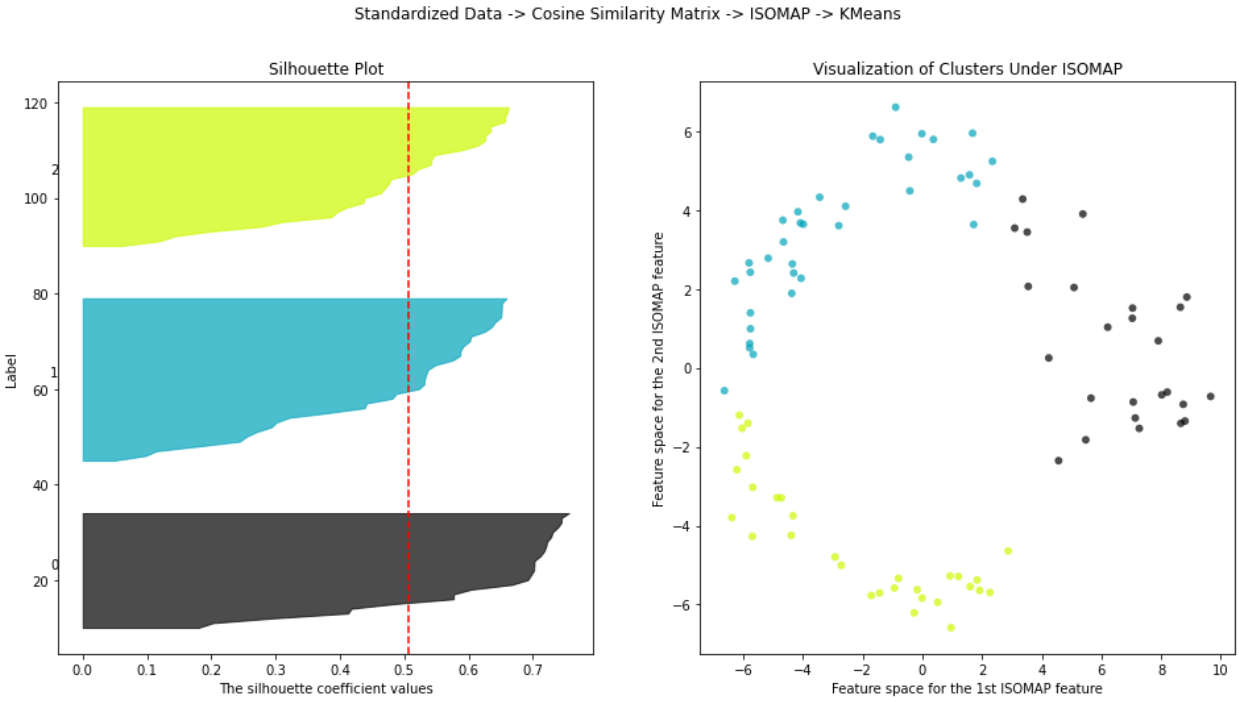




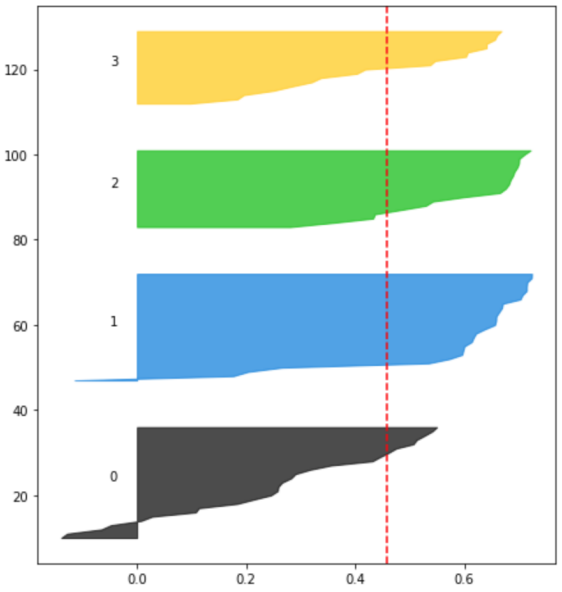
Attempt #6 - Ari (40-50) - Barreira et al. suggests in “Unsupervised Music Genre Classification with a Model-Based Approach” to create a pipeline that takes features from the music set, standardize the features, create a similarity matrix on the standardized features, then cluster the similarity matrix. We will take inspiration from this article and cluster on the similarity matrix. The first attempt we use K-means and LLE. Using this method gives us an Ari score of between 40 and 50. We will attempt to build on this result using other clustering techniques to see if we are able to obtain better results.

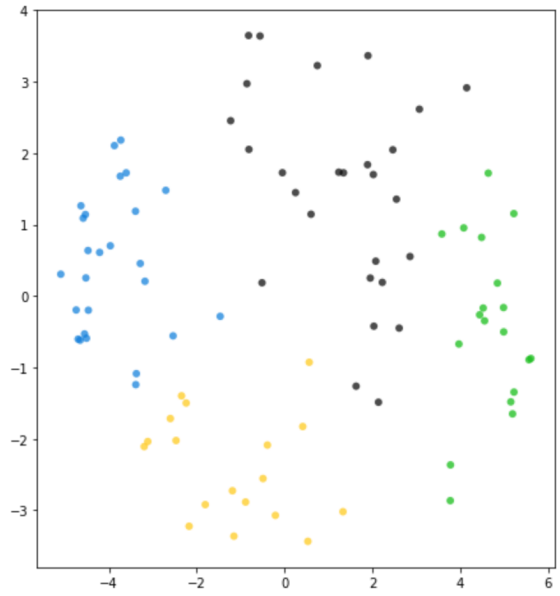


Attempt #7 - ARI 50-60 - We try the same approach as Attempt #6, except this time we use Isomap to attempt to obtain better clusters. With the assumption that we have three clusters, using Isomap with n_neighbors=4 gives both the best visually looking clusters and the best ARI score when using K-means. This method gives us the best submission ARI score so far. We look to further improve our score using similar methods.

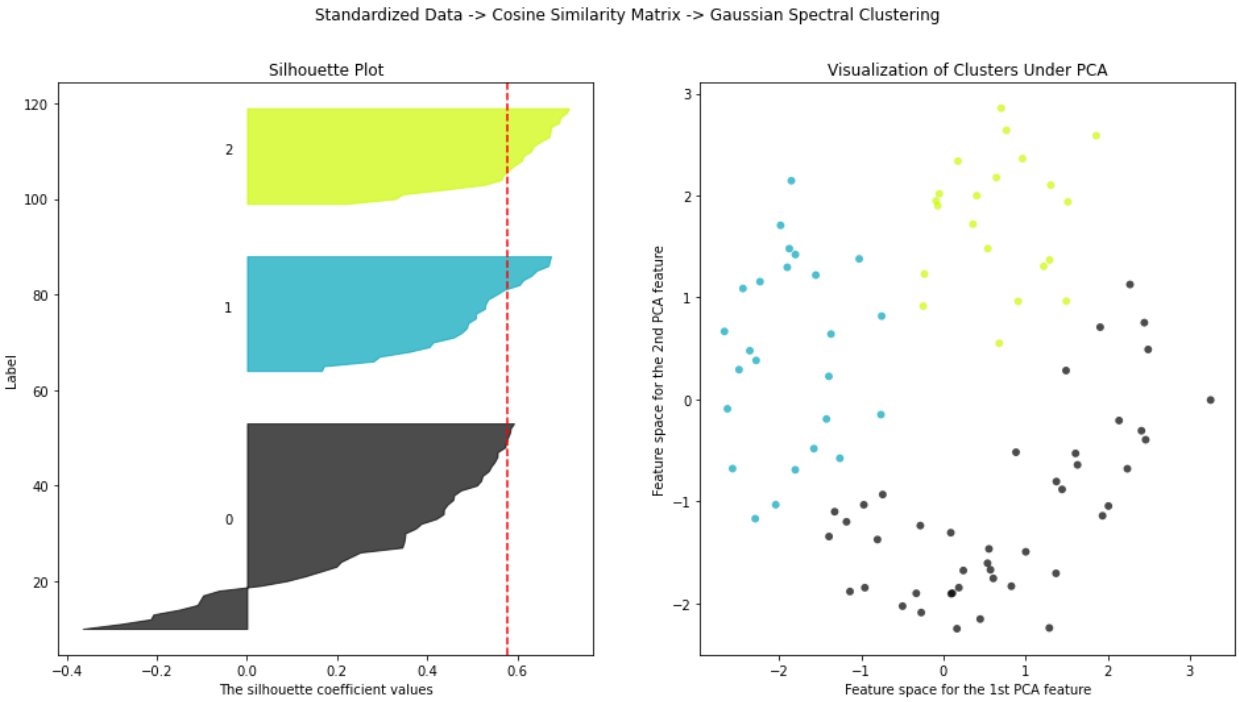


Attempt #8: ARI 20-30: Reading some more online articles about clustering, we found some sources suggesting that a certain subset of features including the mfcc means, spectral centroid and spectral rolloffs varied with music genre. Preceded with a similar approach of correlations and PCA before spectral clustering.

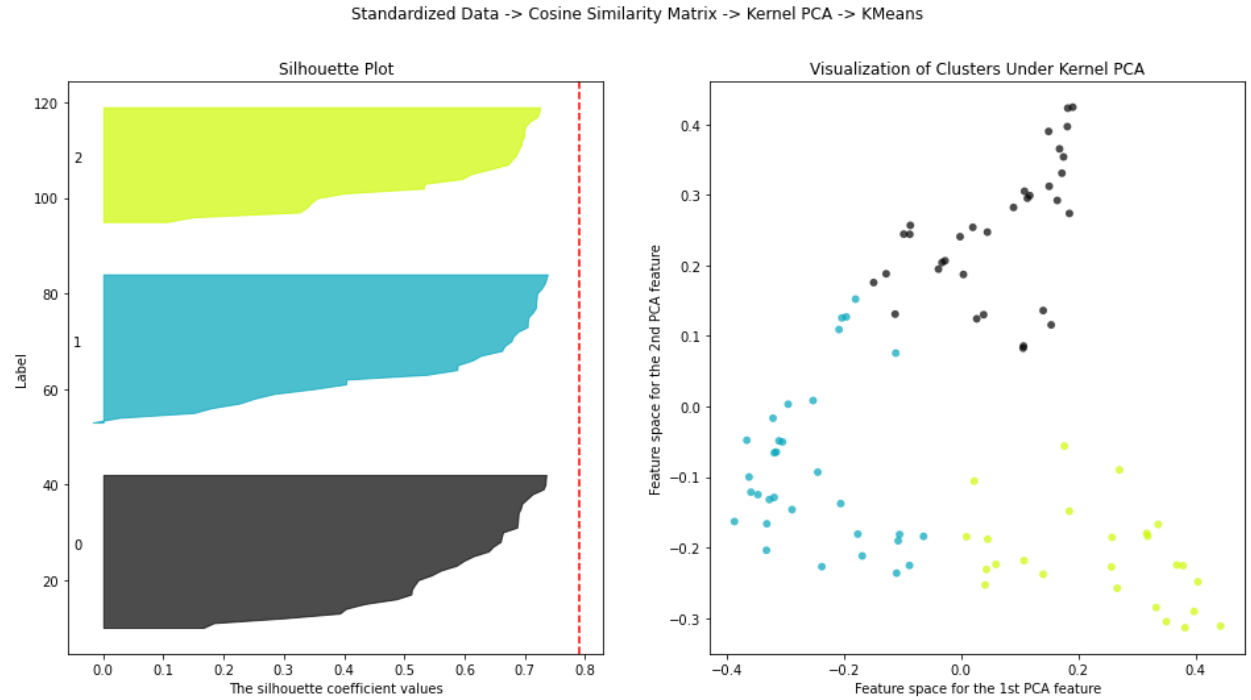




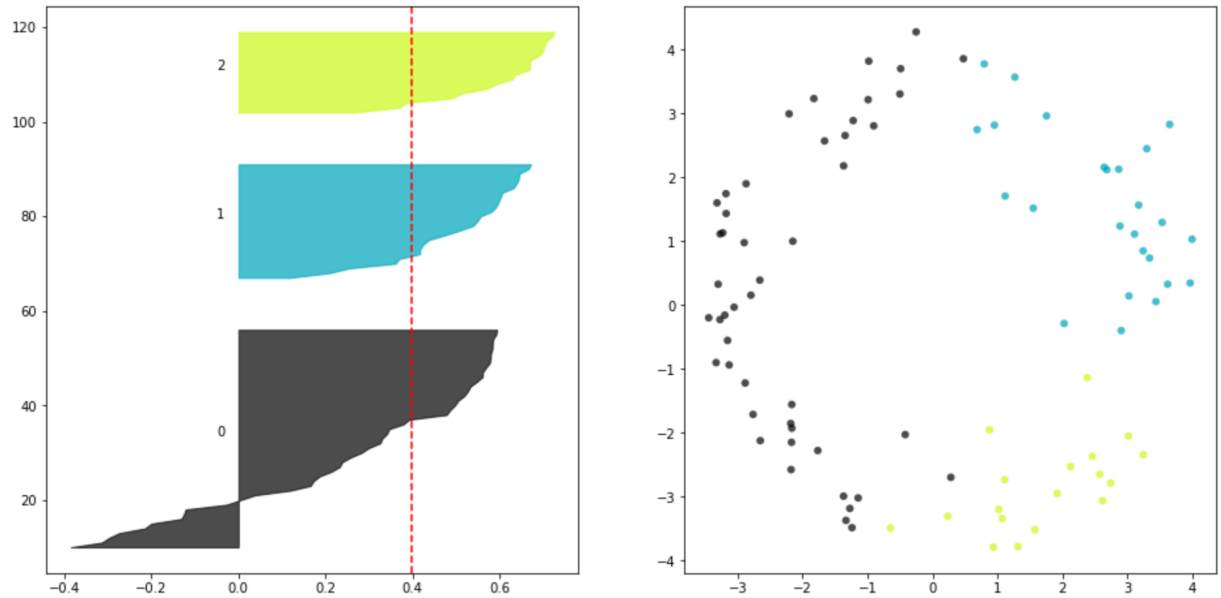
Attempt #9: ARI 30-40: Observing our results from attempt #7, the manifold obtained appears to be curved, indicating that using spectral clustering might help us obtain better results. However, we do not have a great way of determining what gamma to use for our clusters, so we choose to use the default one, which visually seemed to give decent clustering results. The results for doing so are not great however.



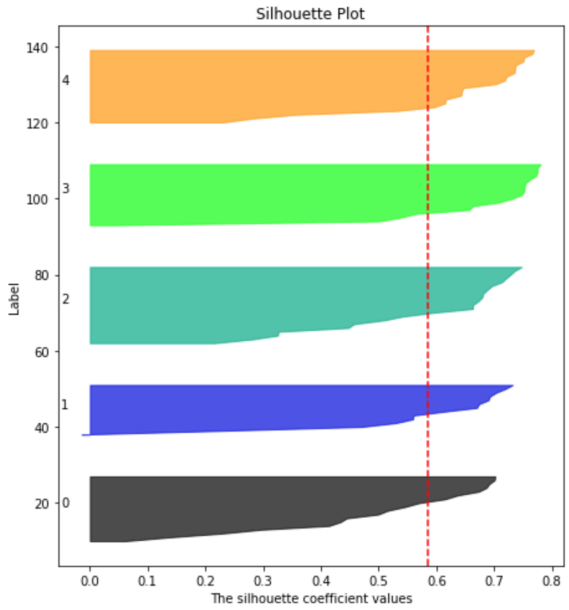
Attempt #10: ARI 30-40: This time, instead of trying spectral clustering, where it was hard to determine what RBF gamma to use, we will instead use kernel PCA to attempt to visualize the application of the RBF kernel, then use k-means clustering. With this method, we were able to obtain a very high silhouette score (0.78), each cluster was well balanced, and visually the clusters looked good. However, the ARI score range was not good.

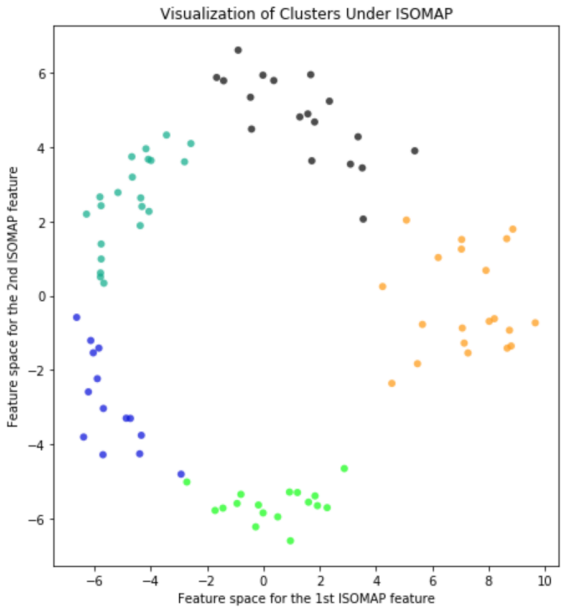


Attempt #11: ARI 30-40: Same as attempt #7, except we normalized the data and set the n_neighbors to 20. But the result is not as good as attempt #7.



Attempt #12: Same as attempt #7, except we used 5 clusters upon observing a higher silhouette score for 5 clusters rather than 3.





Citations

1. Barreira, Luís, Sofia Cavaco, and Joaquim Ferreira da Silva. "Unsupervised music genre classification with a model-based approach." Portuguese Conference on Artificial Intelligence. Springer, Berlin, Heidelberg, 2011.
2. Giannakopoulos, Theodoros, and Aggelos Pikrakis. Introduction to Audio Analysis: a MATLAB® approach. Academic Press, 2014.