**Spotify Project: Application Development**

**Elizabeth Kerrigan**

**Joel Fernandez**

**Wills Mckenna**

**March 2021**

**Overview**

For this project we worked with the Spotify data set-

(<https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks>).

This dataset is over 170,000 songs, with features that capture many aspects of music. We used the dataset to implement several machine learning models with the ultimate goal of creating a “playlist generator”- where based upon user input, the user could be provided with a new collection of 20 songs. Steps in the project included data analysis and exploration, preprocessing of the data, principal component analysis, cluster analysis, and then finally K-nearest-neighbors classification and an app.py that implements the playlist generator task.

**01 - Data Exploration**

**Acousticness:**

A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic. Does not follow a normal distribution. Most of the acousticness appears to occur in the .00 and .95 range.[¶](http://localhost:8888/notebooks/06%20Final/Final%20Project%20-%2001%20-%20Data%20Exploration.ipynb#Does-not-follow-a-normal-distribution.--Most-of-the-acousticness-appears-to-occur-in-the-.00-and-.95-range.)

|  |  |
| --- | --- |
| Datatype: float64  Mean: 0.499  Median: 0.517  Mode: 0.995  Std: 0.380  Min: 0.000  25%: 0.088  50%: 0.517  75%: 0.895  Max: 0.996  Skew: -0.035 |  |

**Top 10 Songs by Acousticness**

### 

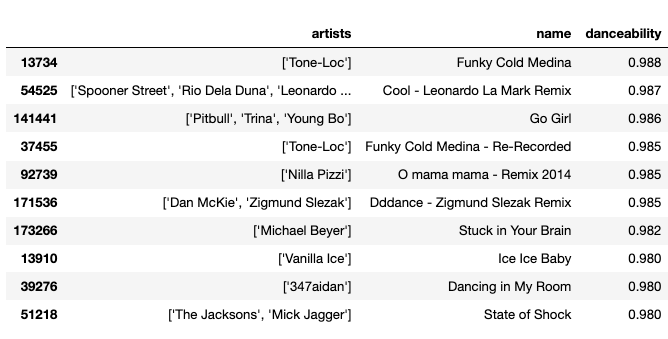
### Danceability:

### Describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

### It's not quite a normal distribution, it appears to skew to the left.[¶](http://localhost:8888/notebooks/06%20Final/Final%20Project%20-%2001%20-%20Data%20Exploration.ipynb#It's-not-quite-a-normal-distribution,-it-appears-to-skew-to-the-left.)

|  |  |
| --- | --- |
| Datatype: float64  Mean: 0.537  Median: 0.548  Mode: 0.565  Std: 0.176  Min: 0.000  25%: 0.414  50%: 0.548  75%: 0.669  Max: 0.988  Skew: -0.236 |  |

### Top 10 Songs by Danceability



### Energy:

### Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy.

### Does not follow a normal distribution. The distribution skews to the right.

|  |  |
| --- | --- |
| Datatype: float64  Mean: 0.483  Median: 0.465  Mode: 0.195  Std: 0.273  Min: 0.000  25%: 0.249  50%: 0.465  75%: 0.711  Max: 1.000  Skew: 0.144 |  |

### Top 10 Songs by Energy

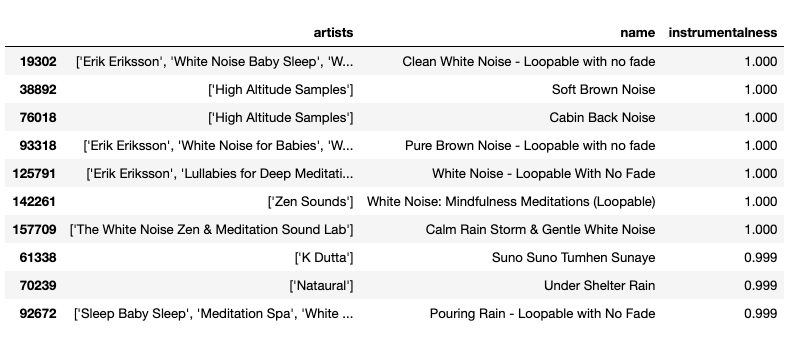


### Instrumentalness:

### Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal.” The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content and is purely instrumental, so for example a classical symphony (with no choral part) would be close to 1.0. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0. Does not follow a normal distribution. Appears to skew to the right. Most of the instrumentalness occurs at the .00 range.

|  |  |
| --- | --- |
| Datatype: float64  Mean: 0.197  Median: 0.001  Mode: 0.0  Std: 0.335  Min: 0.000  25%: 0.000  50%: 0.001  75%: 0.252  Max: 1.000  Skew: 1.364 |  |

### Top 10 Songs by Instrumentalness



### Key:

### The key the track is in. Integers map to the pitches using standard Pitch Class notation.

### E.g. 0 = C, 1 = C♯, D♭, 2 = D, 3 = D♯, E♭, 4 = E, 5 = F, 6 = F♯, G♭, 7 = G, 8 = G♯, A♭, 9 = A, 10 = A♯, B♭, 11 = B

### <https://en.wikipedia.org/wiki/Pitch_class>

### Does not follow a normal distribution. It appears to be sparsely populated, which makes sense as there is no one dominant key in all of western music.

|  |  |
| --- | --- |
| Datatype: int64  Mean: 5.205  Median: 5.000  Mode: 0  Std: 3.518  Min: 0.000  25%: 2.000  50%: 5.000  75%: 8.000  Max: 11.000  Skew: 0.004 |  |

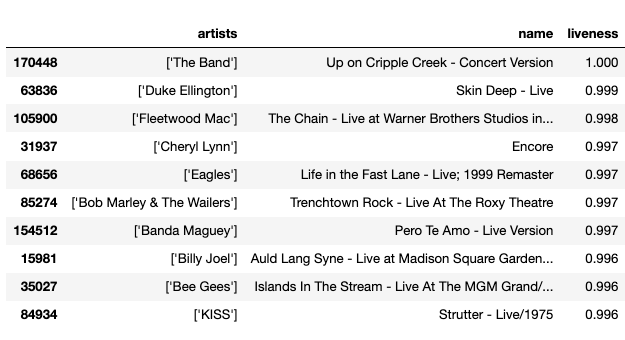
### Liveness:

### Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.

### Does not follow a not follow a normal distribution. Appears to skew to the right. Most of the liveness occurs at the .1 range.

|  |  |
| --- | --- |
| Datatype: float64  Mean: 0.211  Median: 0.138  Mode: 0.111  Std: 0.180  Min: 0.000  25%: 0.099  50%: 0.138  75%: 0.270  Max: 1.000  Skew: 2.078 |  |

### Top 10 Songs by Liveness



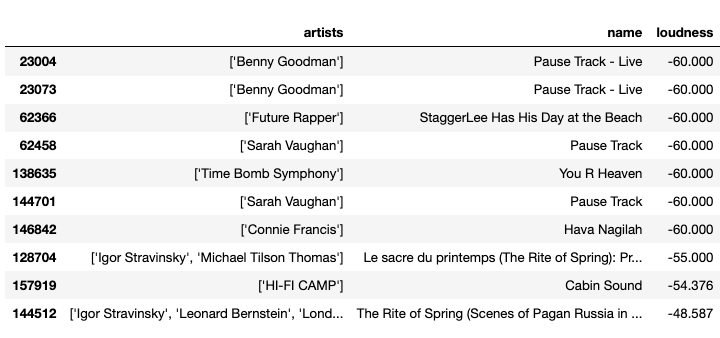
### Loudness:

### The overall loudness of a track in decibels (db). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.

### Does not follow a normal distribution. Appears to skew to the left. Most of the loudness occurs at -10db.[¶](http://localhost:8888/notebooks/06%20Final/Final%20Project%20-%2001%20-%20Data%20Exploration.ipynb#Does-not-follow-a-normal-distribution.--Appears-to-skew-to-the-left.--Most-of-the-loudness-occurs-at--10db.)

|  |  |
| --- | --- |
| Datatype: float64  Mean: -11.751  Median: -10.836  Mode: -7.578  Std: 5.692  Min: -60.000  25%: -14.908  50%: -10.836  75%: -7.499  Max: 3.855  Skew: -0.989 |  |

### Top 10 Songs by Loudness



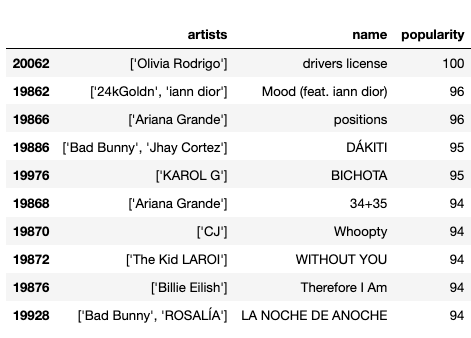
### Popularity:

### The higher the value, the more popular the song is.

### Does not follow a normal distribution. Appears to skew to the right. Can't explain or interpret what going on around the 0 range. Could be due to there being a high number of songs that are not listened to on Spotify.

|  |  |
| --- | --- |
| Datatype: int64  Mean: 25.693  Median: 25.000  Mode: 0  Std: 21.873  Min: 0.000  25%: 1.000  50%: 25.000  75%: 42.000  Max: 100.000  Skew: 0.363 |  |

### Top 10 Songs by Popularity



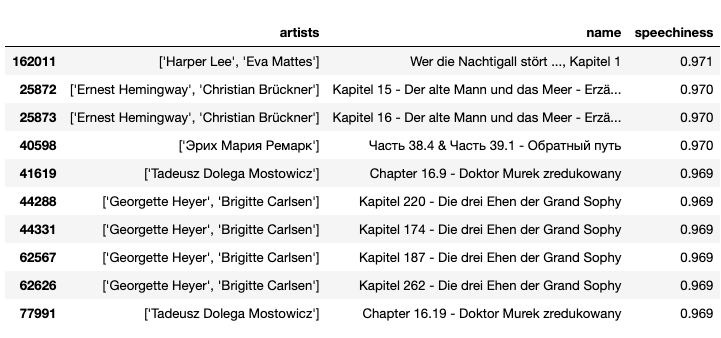
### Speechiness:

### Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words, such as audio-books. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.

### Does not follow a normal distribution. Appears to skew to the right. Most of the speechiness occurs at the 0.0 range.

|  |  |
| --- | --- |
| Datatype: float64  Mean: 0.106  Median: 0.045  Mode: 0.0337  Std: 0.182  Min: 0.000  25%: 0.035  50%: 0.045  75%: 0.076  Max: 0.971  Skew: 3.751 |  |

### Top 10 Songs by Speechiness



### Tempo:

### The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

### Although the distribution almost looks normal. Skews to the right, but not by much. Can't explain why the mode is 0. Requires further investigation.

|  |  |
| --- | --- |
| Datatype: float64  Mean: 117.006  Median: 115.816  Mode: 0.0  Std: 30.254  Min: 0.000  25%: 93.931  50%: 115.816  75%: 135.011  Max: 243.507  Skew: 0.423 |  |

### Top 10 Songs by Tempo



### Valence:

### A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

### Does not follow a normal distribution. It appears to skew to the left.

|  |  |
| --- | --- |
| Datatype: float64  Mean: 0.525  Median: 0.536  Mode: 0.961  Std: 0.264  Min: 0.000  25%: 0.311  50%: 0.536  75%: 0.743  Max: 1.000  Skew: -0.101 |  |

### Top 10 Songs by Valence



### Explicitness:

### Indicates if explicit language was used in the song.

### 93% of the songs were not explicit and 7% of the songs were explicit.[¶](http://localhost:8888/notebooks/06%20Final/Final%20Project%20-%2001%20-%20Data%20Exploration.ipynb#93%-of-the-songs-were-not-explicit-and-7%-of-the-songs-were-explicit.)

|  |  |
| --- | --- |
| Datatype: int64  Not Explicit: 0.932  Explicit: 0.068 |  |

### Mode:

### Mode indicates the modality (major or minor) of a track, the type of scale from which melodic or harmonic content is derived. Major is represented by 1 and minor is 0.

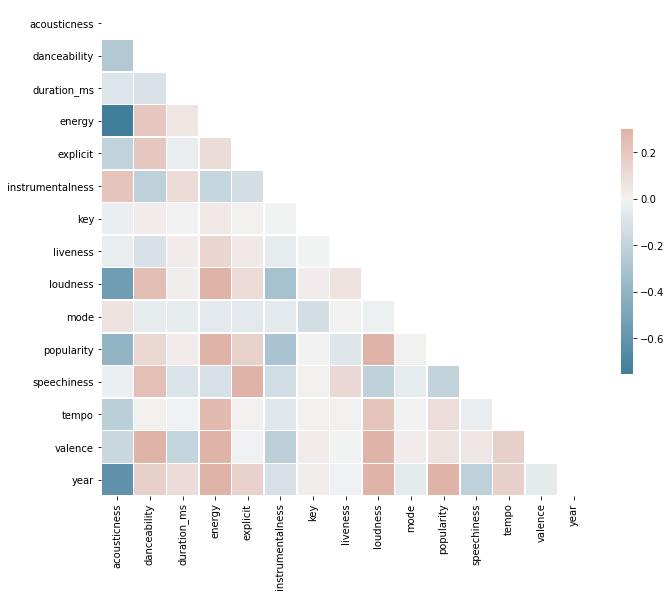
### 70% of the melodic content has a major modality and 30% has a minor modality.

|  |  |
| --- | --- |
| Datatype: int64  Major: 0.702  Minor: 0.298 |  |

# **Correlation Matrix**

### After viewing the correlation matrix, there weren't any high correlation candidates for feature reduction.

### Acousticness and energy appears to be the most significantly correlated, but still not enough for consideration.



### Top Negative Correlations

acousticness loudness -0.546639

loudness acousticness -0.546639

popularity acousticness -0.396744

acousticness popularity -0.396744

instrumentalness loudness -0.317562

loudness instrumentalness -0.317562

popularity instrumentalness -0.300625

instrumentalness popularity -0.300625

danceability acousticness -0.263217

acousticness danceability -0.263217

dtype: float64

### Top Positive Correlations

energy loudness 0.779267

year 0.540850

year energy 0.540850

valence danceability 0.536713

danceability valence 0.536713

popularity year 0.513227

year popularity 0.513227

loudness year 0.465189

year loudness 0.465189

speechiness explicit 0.353872

dtype: float64

**02 – Pre-Processing**

### Performed the following tasks:

### Checked for null values

### 

### Of the 16 numeric attributes, 5 attributes ('key', 'loudness', 'popularity', 'tempo', 'duration\_ms') needed additional normalization

### 

### 2 categorical attributes ('explicit', 'mode') required dummy attribute conversions.

### 

### Binned 'year' and 'release\_date' attribute for every 10 years.

### Binning 'release\_date' would produce the same results as 'year'. Attribute is redundant, so it will be dropped from the dataframe.

**03 – Principle Component Analysis**

### 6 Attributes accounted for 0.88 of the variance.

mode\_0: 0.35

acousticness: 0.20

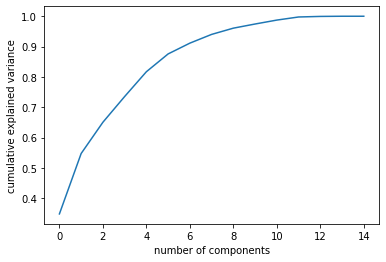
explicit\_0: 0.10

instrumentalness: 0.08

key: 0.08

valence: 0.06

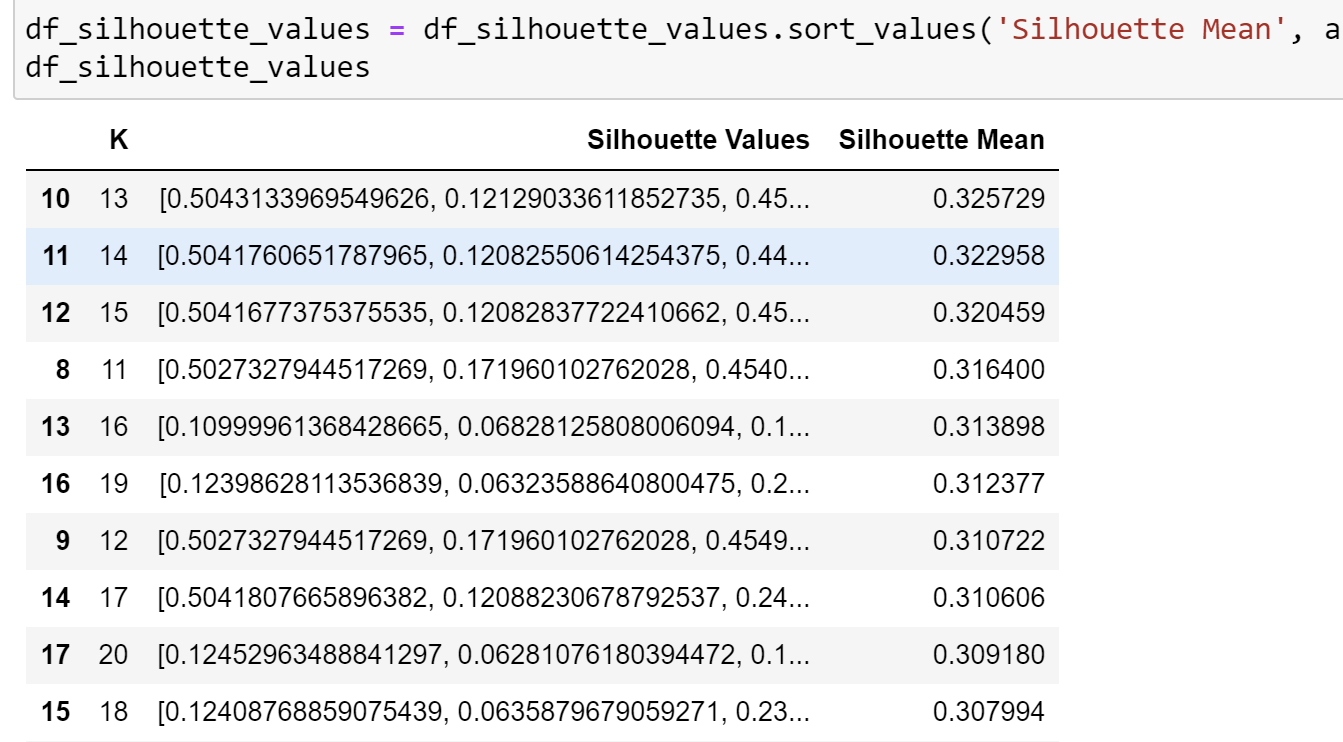
### Scree Plot



**04 - Cluster Analysis**

**Cluster model 1: k-means with our PCA attributes**

The best number of K clusters was 13 after looking at our silhouette values:



**The cluster centroids:**

****

**Size of each cluster:**

Size of Cluster 0 = 7290

Size of Cluster 1 = 23732

Size of Cluster 2 = 14037

Size of Cluster 3 = 20718

Size of Cluster 4 = 23099

Size of Cluster 5 = 16989

Size of Cluster 6 = 4760

Size of Cluster 7 = 23589

Size of Cluster 8 = 10617

Size of Cluster 9 = 8014

Size of Cluster 10 = 7071

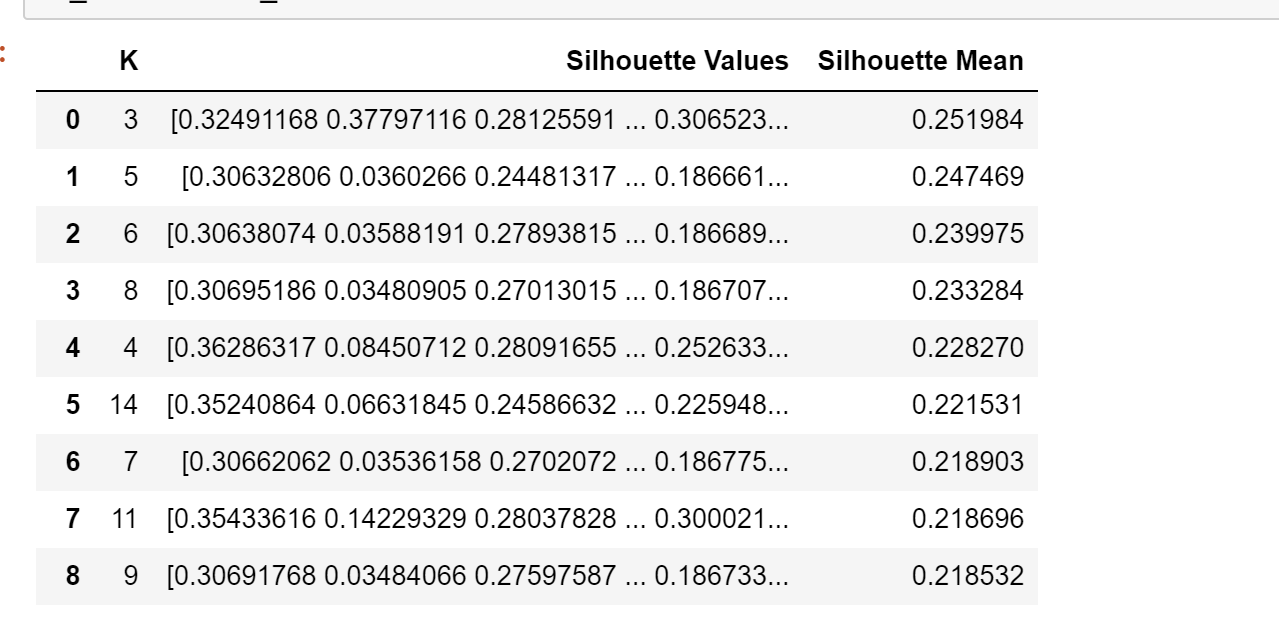
Size of Cluster 11 = 9692

Size of Cluster 12 = 4781

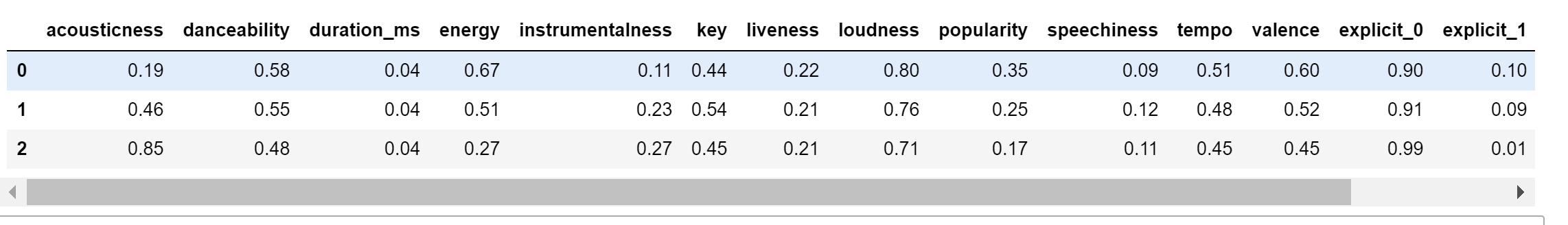
We then added the cluster assignment to each data point, and created a new csv value with the new findings. The cluster assignments later became class labels during KNN.

**Cluster model 2: k-means without PCA.**

The best number of K clusters was 3 after looking at silhouette values:



Centroids (subset):



Size of each cluster:

Size of Cluster 0 = 51901

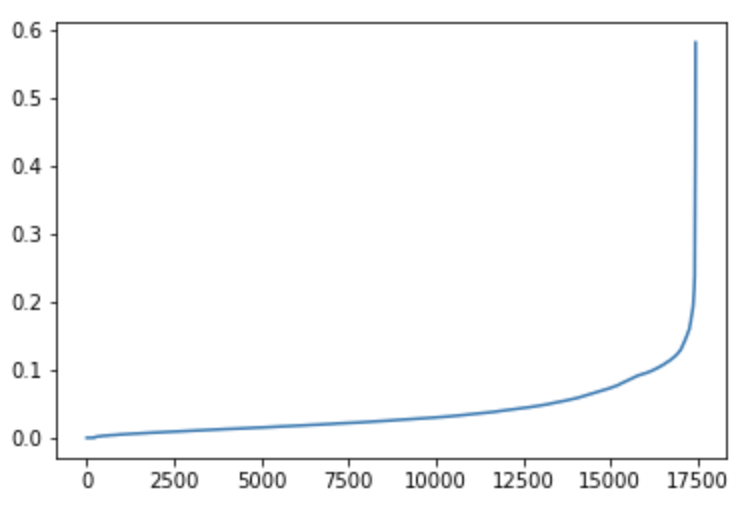
Size of Cluster 1 = 60903

Size of Cluster 2 = 61585

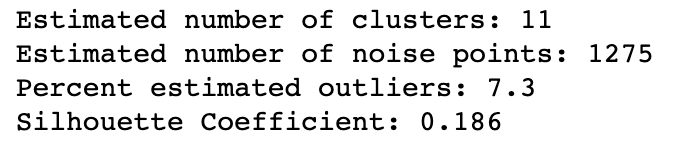
A new csv file was created with these cluster assignments.

**Cluster model 3: DBSCAN with our PCA attributes**

Using the 6 attributes from PCA, a random sample of the data and the knee plot below, we determined the best fit for epsilon to be roughly 0.18, seen below as the point of highest curvature in the plot.

****

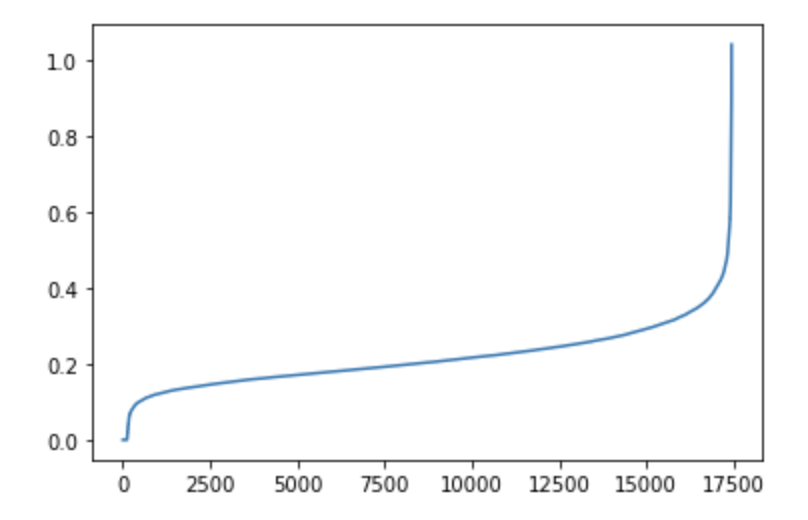
Applying DBSCAN with epsilon = 0.18 and min\_sample = 20, we found an estimated 11 clusters with 7.3% of the sample as outliers. We also found the silhouette value to be 0.186.

****

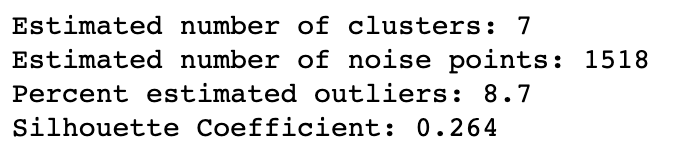
We then applied the cluster assignments to the entire dataset to create labels and saved the new csv file.

**Cluster model 4: DBSCAN without PCA attributes**

Without first applying PCA, we used DBSCAN using each numerical attribute of the Spotify dataset. Using the knee plot on a random sample of 10% of the total data below, we can see that the greatest curvature occurs at about 0.4.

****

Applying DBSCAN with epsilon = 0.4 and min\_sample = 20, we found an estimated 7 clusters with 8.7% of the sample as outliers and a silhouette value of 0.264. This silhouette coefficient is a 40% improvement than DBSCAN with PCA.

****

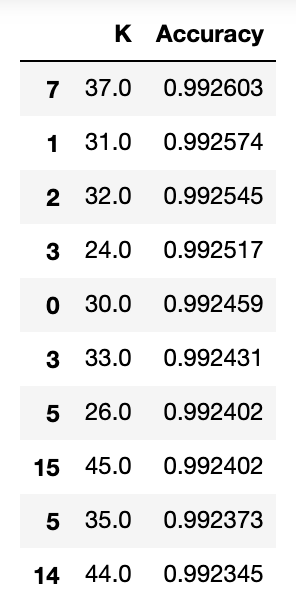
**Using silhouette values to assess these 4 clustering techniques, we determined the best model to be K-Means using PCA with a mean silhouette value of 0.33 for K = 13. That is the model we will employ for our app.**

**05 – Classification**

**KNN using K-Means clusters with and without PCA**

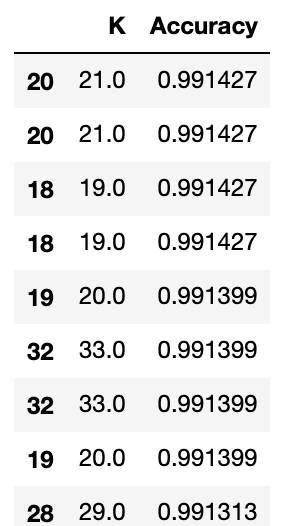
**With PCA**

We found K = 37 to have the highest accuracy with PCA reduction.



**Without PCA**

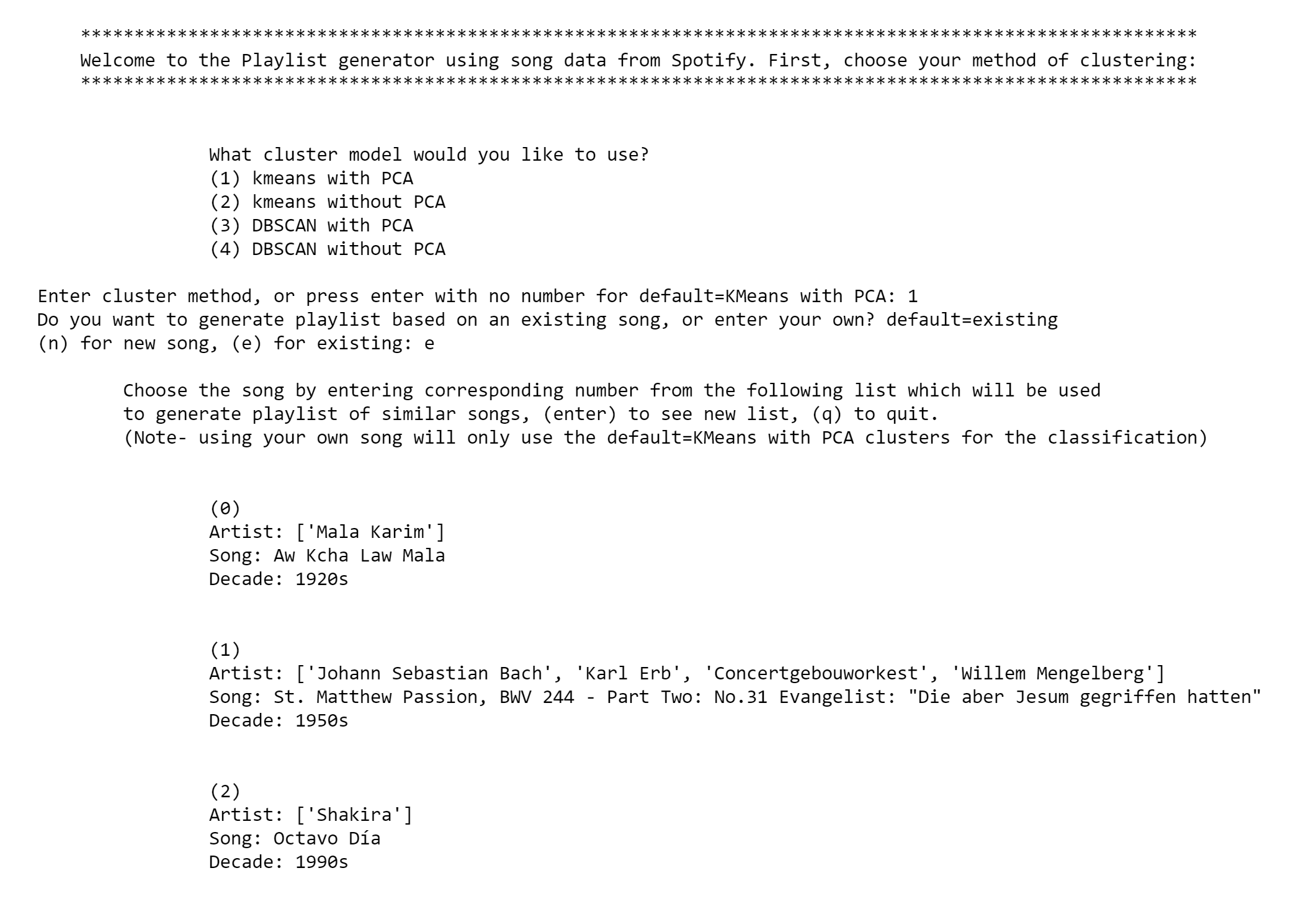
We found K = 21 to have the highest accuracy for K-Means without PCA reduction.



This section of our project has shown that effective KNN classifiers can be developed using our clusters found from previous experiments. The first KNN was using our PCA-Kmeans clusters, and the most effective K was 37. The second KNN classifier was with all features and clusters using Kmeans, and the best K was found to be 21. KNN with our DBSCAN or HAC clusters will not be implemented for this project.

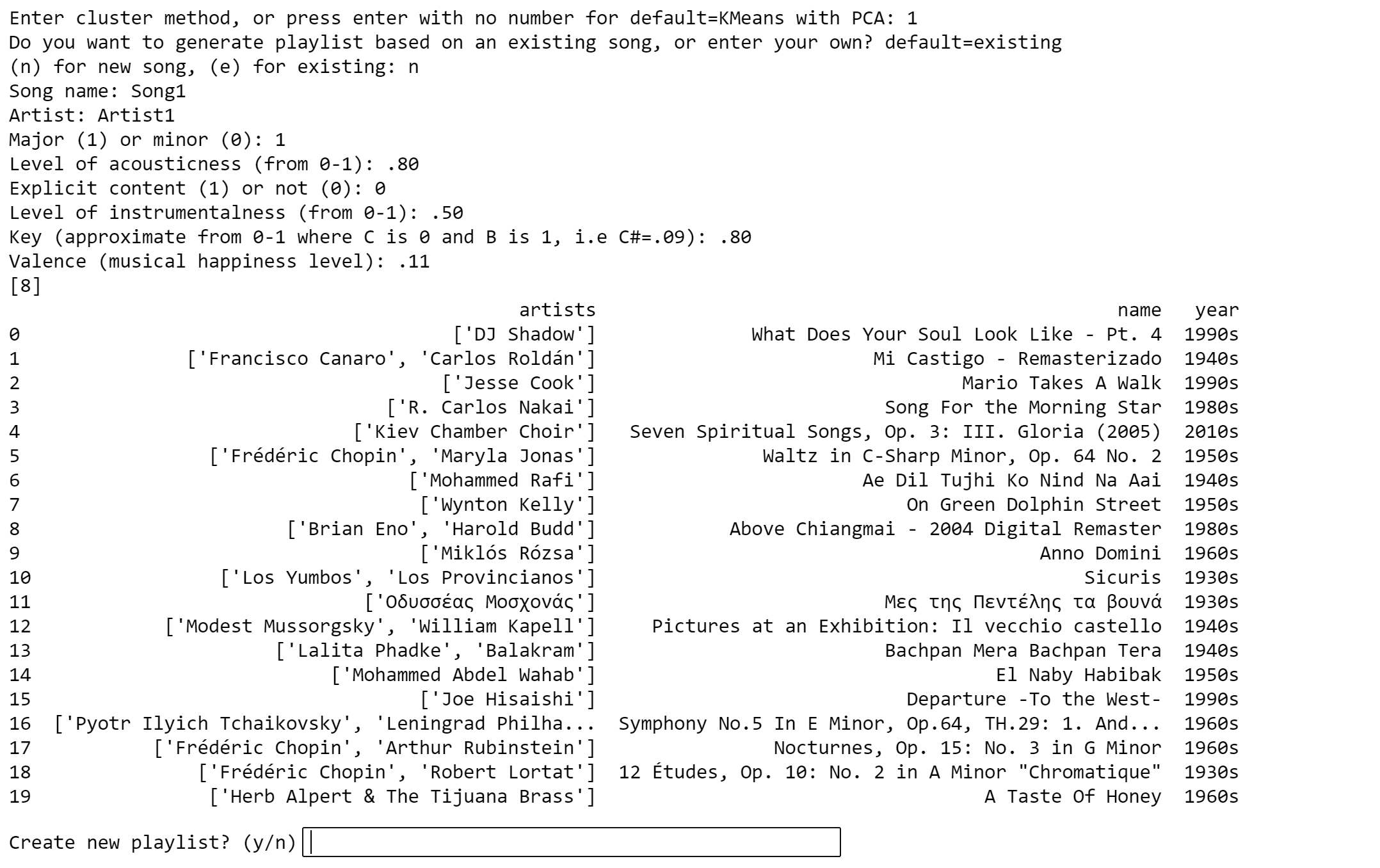
**06 – App Implementation**

The app integrated our cluster data and KNN classifier to create an example of what command line user input could look like. This app addressed the original aim of the project, which was to create a “playlist generator” that could provide the user with a list of similar songs. The app has two main features- one, a new playlist is generated given the user chooses one song out of a list of ten already existing in the dataset; and two, a new playlist is generated given the user inputs a completely new song the app does not know about. The first feature is based solely off of the cluster results from the cluster analysis, while the second feature utilizes the KNN classifier that was developed using the KMeans with PCA cluster data set.

Example screenshot of first feature: 

As can be seen, the user is asked to choose one of the four clustering types that was done in the cluster analysis and the app then uses the appropriate dataset that has the cluster assignments; when the user chooses one of the listed songs, the app simply finds 20 random songs in the same cluster.

Example screenshot of second feature:



Here, the user is asked to input aspects of their new song. The attributes are those that were discovered during our PCA analysis of the dataset. The KNN classifier then assigns it a cluster number, and then 20 songs in the existing data set are chosen based on that cluster assignment.

The video demonstrating the app from our presentation is linked [here](https://vimeo.com/523490396/1f4be6bbce).