Unsupervised Learning Techniques for Music Collection Segmentation

Frank Johnson

UCF MSDA Program
Orlando, United States
fr005010@ucf.edu

Abstract— As curators of music, DJs often amass vast libraries of songs and sounds over the course of their careers. Over time, library management becomes crucial to ensure that a DJ can access their tracks quickly as well as find other related tracks that will mix well with the original one.

As a DJ, I regularly collect music across many genres to use in music mixes, livestreams, and in-person events. As I continually search for new methods to group tracks together into playlists I try to incorporate techniques of data science to optimize my analysis process. The goal of this project was to employ statistical analysis and unsupervised learning techniques to uncover insights about the tracks in my library, discover connections between tracks, and ultimately grouping them into meaningful categories that will support my longer-term focus.

The tracks used were sourced from my own Rekordbox DJ software library in the form of an XML file that I parsed to collect relevant metadata for each track. Additional metadata for these tracks is then retrieved from Spotify's API.

Data files: (*collection_7.3.24.xml), (df_spotified_v1.csv)*

Jupyter Notebook: *DJ Library Clustering Analysis*

Keywords— *Metadata, Correlation, Principal Component Analysis, K-Means, K-Medoids*

I. Music Import and Exploratory Data Analysis

The relevant metadata from Rekordbox included: Track ID, Track Name, Artist, BPM (Beats Per Minute), Key, Runtime (in seconds), Genre, and Year (denoting year released). Further details of these attributes are covered in the Jupyter Notebook.

To get the raw music metadata, I first imported the XML file of music library metadata stored in the Rekordbox DJ software. I then parsed this XML file and stored relevant data as a dataframe to support further data cleaning and analysis. This included cleaning the dataframe to eliminate many duplicate files inadvertently created over the years as well as standardizing column names and data entries via methods such as value imputation or row removal. The resulting dataframe contained 2,717 tracks. The result is shown in *Table 1.* After importing and cleaning the initial data, I conducted EDA to reveal any interesting insights which could be leveraged for further analysis.

I used a correlation heatmap (*Figure 2)* to examine the relationship between the relevant attributes. This provided some intuition about the features' relationship with each other. There was a mildly negative relationship between Year and Runtime. This meant that more modern tracks tended to have a shorter runtime than older ones (*Figure 3).* There were also many tracks that had no year assigned for which the average run time was nearly 8 minutes.

Overall, this analysis gave me some intuition of the types of tracks in my library and can enable me to pursue DJ interests according to the bulk of my library makeup. For instance, I may opt to focus more on creating mixes and pursuing gigs for events where the audience prefers modern music and some throwbacks from the 90s / early 2000s. However, I may shy away from events for Gen X / Baby Boomers who may want to hear mostly pre-90s music.

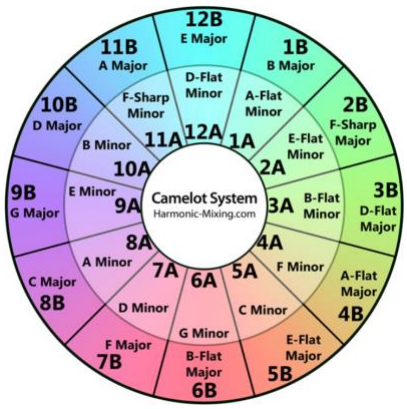

Fig. 1. *Camelot Wheel Key Mapping*

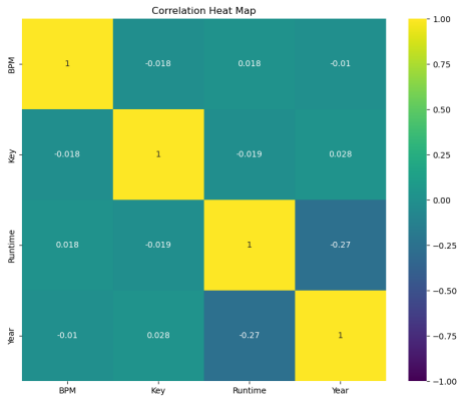

Fig. 2. *Rekordbox Attributes Correlation Heatmap*

TABLE I. CLEANED DATAFRAME HEAD

	TrackID	Name	Artist	BPM	Key	Runtime	Genre	Year
1105	261731391	Ice Me Out (Dirty-Cyberkid Intro/Outro)	Kash Doll	75.00	9A	193	Trap	2018
1106	143132557	Where I'm From	Digable Planets	98.25	4B	268	Hip Hop	1993
1107	57065216	Ms. New Booty feat. Ying Yang Twins (DJ Organi...	Bubba Sparxxx	97.00	12A	270	hip hop x r&b	2006
1108	90098842	Salt Shaker (CLEAN Intro)	Ying Yang Twins	102.06	2A	267	hip hop x r&b	2002
1109	10239911	Get Your Freak On (Dirty Intro)	Missy Elliott	88.00	4A	185	Hip Hop	2011
1110	165170375	Back That Ass Up feat. Mannie Fresh & Lil' Way...	Juvenile	95.80	6A	280	hip hop x r&b	1998
1111	169581582	Izzo (H.O.V.A.) (Dirty)	Jay-Z	87.00	2B	238	Hip Hop	2001
1112	64047844	Blame It (Instrumental)	Jamie Foxx ft T-Pain	88.00	8B	280	R&B	2008
1113	223640683	100 Bands (Dirty)	Mustard ft Quavo, 21 Savage, YG & Meek Mill	93.00	12A	179	Hip Hop	2019
1114	159971611	After Hours (Clean)	The Weeknd	109.00	5A	362	Pop	2020

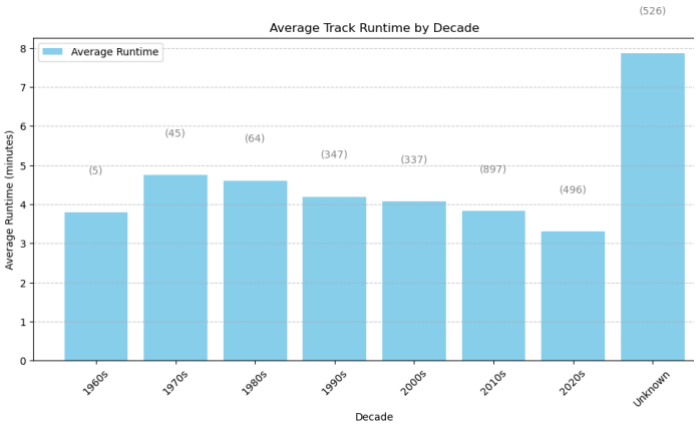

Fig. 3. *Average Track Runtime by Decade*

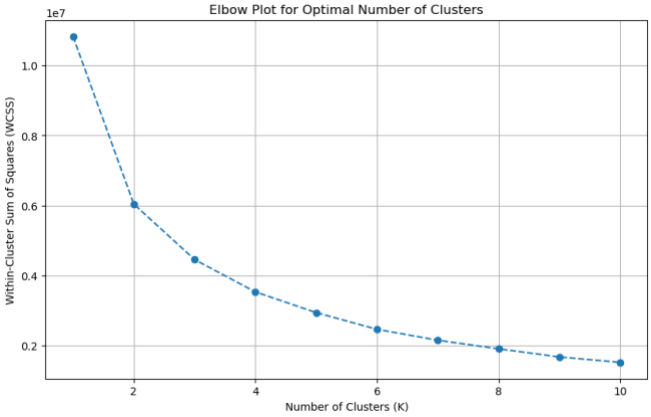

Fig. 4. *Elbow Method for Initial K-Means Clustering*

II. INITIAL K-MEANS CLUSTERING

Next, I conducted K-Means clustering to group tracks into clusters according to relevant attributes. This was a useful exercise to determine if there were other methods to group tracks together aside from by genre, which is the default method of grouping many songs in most music platforms. The The I then clustered the data using the K-Means algorithm to group tracks into globular clusters. I clustered the data using the BPM, Key, Runtime, and Decade attributes (a calculated feature/column based on the Year of a track). To determine the

optimal number of clusters for grouping the tracks, I used the elbow method, which plots the within-cluster sum of squares (WCSS) against the number of clusters. WCSS measures total variance within each cluster, with lower values indicating that clusters are more compact. As K grows, the clusters become more compact and WCSS decreases. When the rate of this decrease in WCSS begins to level out, the algorithm has ideally found the optimal balance between the number of clusters and cluster compactness.

Figure 4 shows the rate of decrease began to flatten at about k=2 clusters; however, this would not make for a very interesting track grouping, and so I used k=4 clusters instead, which was still a fair choice given that the slope begins to decrease faster for K > 4.

Most of the tracks from Cluster 0 appeared to be from the 1990s, while many tracks from Cluster 2 were multi-track mixes that I previously recorded myself (as evidenced by the runtimes, which are much higher than the standard runtime of any one track). Cluster 2 tracks also seemed to consist of those tracks that did not have a specific Year of release designated in the metadata. In contrast, data points in Cluster 4 seemed to consist of those tracks from the 2020s decade.

III. Fetching additional spotify metadata

To enhance clustering and analysis, I then decided to access Spotify's API to retrieve additional track metadata for review. Details of accessing the API via relevant functions is detailed in the accompanying Jupyter Notebook. Although standardizing the Rekordbox track names did result in retrieving Spotify data for 1,000 more tracks than would have been found otherwise, not all tracks were found via the Spotify API. In the end, 1,543 of the original 2,717 tracks were located. The following attributes were retrieved for the subset of Rekordbox library songs that were found in Spotify's database: Danceability, Energy, Loudness, Speechiness, Acousticness, Instrumentalness, Liveness, and Valence (full details of these attributes available in Jupyter Notebook).

IV. Additional Exploratory Data Analysis

After including the Spotify metadata in the Rekordbox library data, I again measured the correlation between all variables and found a correlation between Loudness and Energy. Given that Loudness on its own is not an especially interesting characteristic, I removed this feature to mitigate any impact to clustering and kept the Energy attribute instead.

I also plotted the distribution of each feature to obtain a better understanding of the general data (*Figure 5*). A large portion of the tracks hovered around the 100 BPM range, and Keys were somewhat uniformly distributed across all tracks. Track runtime followed a normal distribution with an average value of around 4 minutes and standard deviation of over 2 minutes. My own pre-recorded DJ mixes included in the library certainly skewed this number, as they can run over 1 hour depending on the mix.

Additionally, according to the corresponding distributions, tracks tended to be more danceable, medium energy, low in acoustic instrumentation, and generally not performed in front of a live audience. Speechiness was also relatively low across most tracks; however, virtually all tracks had low Instrumentalness. These two features seemed to directly contradict each other, as one would expect low Speechiness tracks to have high Instrumentalness and vice versa. The fact that both attributes were low seemed to point to a potential inaccuracy in the way Spotify measured and/or defined these metrics.

V. K-Medoids Clustering with Principal Components

Finally, I decided to reduce the data dimensionality of the 11 total features used as much as possible by applying Principal Component Analysis (PCA) to retain relevant information while enabling new insights. I also opted to use K-Medoids clustering to designate a specific track in each cluster as the centroid. Plotting the variance explained by PCA components (*Figure 6*) revealed that 8 components were still needed to capture 80% of the data's variance. While this did reduce dimensionality, the magnitude of reduction was not very large. The resulting clusters did not appear to group tracks by any discernable pattern; however, I was able to identify the *Centroid Tracks* which essentially represented an average of the relevant information of all tracks in their respective clusters (*Table II*). These tracks could be designated as the best example of the types of tracks found in each cluster.

VI. K-Medoids Clustering with 2 Principal Components

As a last step, I also clustered tracks using only the first 2 principal components. I plotted the tracks on a scatter plot of Component 1 vs. Component 2 and highlighted the *Centroid Tracks* as well as 1 additional randomly selected track for each cluster (Figure 7). While these 2 components only captured approximately 30% of the data's variance, they allowed me to visualize the K-Medoids clustering in 2D space. Additionally, because this was an unsupervised learning problem with no target clustering assignments, there was no incorrect clustering methodology and grouping the data in different ways allowed me to see other categorizations of tracks that I would not have before.

VII. Conclusion

DJing, like most artistic forms, is a difficult subject to objectively analyze. Waveforms, tempo, and track length are just a few of the attributes we can review for tracks to gain further insights; however, often metadata alone cannot fully describe a musical composition. This is evidenced by some of the analysis of this project. While unsupervised learning

afforded the ability to examine tracks without needing to label findings as right or wrong, it sometimes resulted in ambiguity that is uninterpretable.

This was also seen in some of the data insights derived from Spotify when the characteristics of a song that are described by specific attributes contradict the values of those very attributes. For instance, many tracks in the library, while obviously joyous in tone when listened to, have very low valence values that imply these same tracks are not cheerful at all. Conversely, many tracks that exude negative emotions such as fear or anger have high valence values that denote positive moods. Many other examples exist for other features. That said, these attributes do accurately describe other tracks and provide new dimensions from which to view the data.

Overall, the statistical and machine learning techniques used for analyzing this music library help to shape my understanding of the data at some level, albeit in an abstract manner. Additional data cleaning, feature engineering, and creative modeling solutions would surely yield even more interesting results.

TABLE II. K-MEDOID CENTROID TRACKS USING 8 PCA COMPONENT

	Name	Artist	Cluster_KMed
824	Broken Clocks	SZA	0
823	Coffee Bean	Travis Scott	1
1198	Heartless	Kanye West	2
415	Lavender	BadBadNotGood feat Kaytranada	3

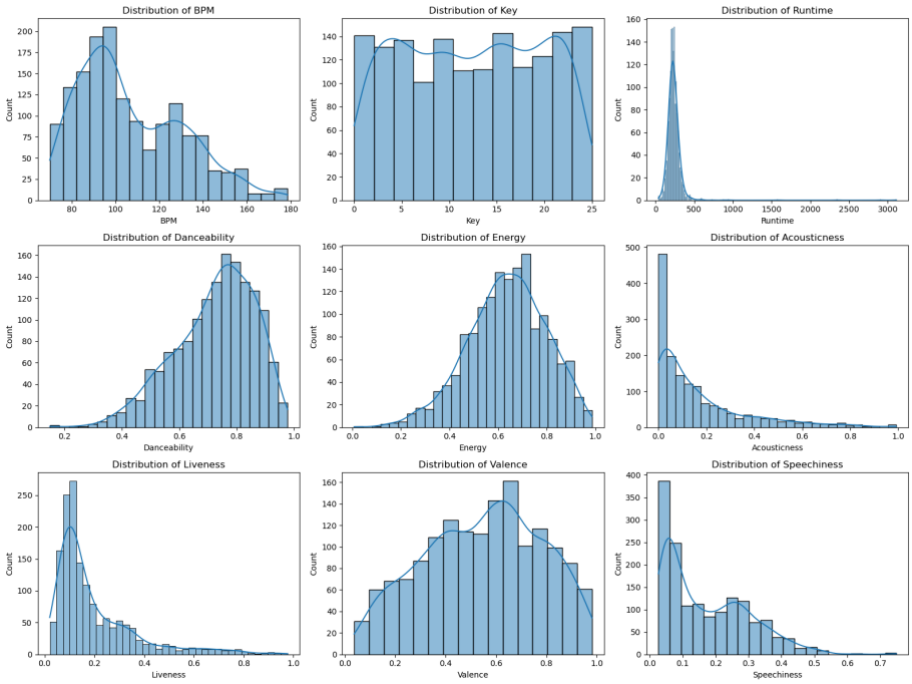

Fig. 5. *Attribute Data Distributions*

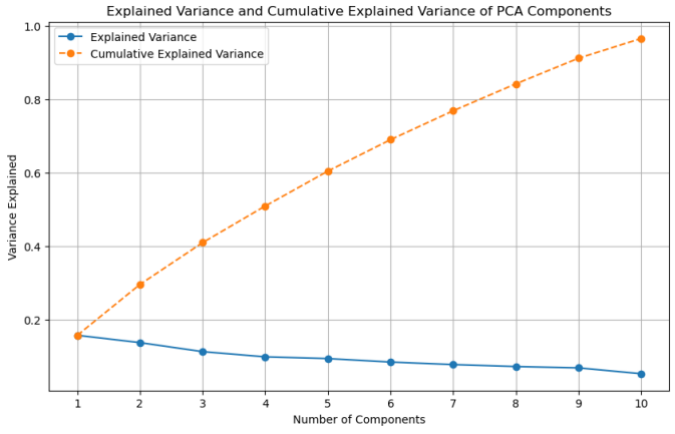

Fig. 6. *Principal Components Explained Variance*

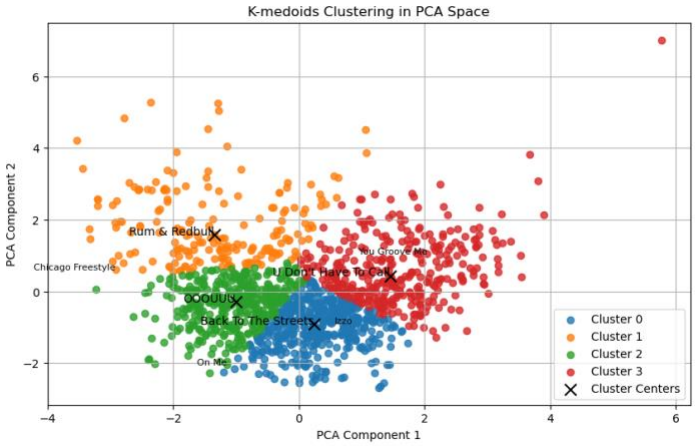

Fig. 7. K-Medoids Clusterings with 2 PCA Components

REFERENCES

[1] J. Camagong, "Data Science for DJing," The Startup, Sep. 19, 2020. [Online]. Available: https://medium.com/swlh/data-science-for-djing-128f1b347a9

[2] "Harmonic Mixing Guide," Mixed In Key. [Online]. Available: https://mixedinkey.com/harmonic-mixing-guide/

[3] Akamai Developer, "How to Use Spotify's API with Python | Write a Program to Display Artist, Tracks, and More," YouTube, Dec. 7, 2022. [Online]. Available: https://www.youtube.com/watch?v=WAmEZBEeNmg

[4] Z. Rushirajsinh, "The Elbow Method: Finding the Optimal Number of Clusters," Medium, Nov. 4, 2023. [Online]. Available: https://medium.com/@zalarushirajsinh07/the-elbow-method-finding-the-optimal-number-of-clusters-d297f5aeb189