Cluster Analysis of Top Songs

Evan Gray

1) Introduction

Music is a constantly changing field. The sound and feel of popular songs are always in flux and are vastly different from decade to decade. This may reflect broader cultural and technological trends as personal listening devices became more readily available to the average consumer. [1] Likewise, instrumentation and tastes evolved with the growing liberalization and globalization of society as evidenced by the stardom of bands such as the Beatles and Rolling Stones. Out of rock and jazz came hip hop while the vague and intangible sound of pop remained dominant in the ears of music enjoyers. [2] This project seeks to evaluate if we can understand the top music tracks of multiple years through clustering.

2) Data

Although this project had the original intention of using personal, themed playlists to evaluate their cohesiveness, Spotify changed its API data availability and usage policy during the data collection process. The primary purpose of this project is data analysis and evaluation of popular tastes as opposed to the creation of any recommender or predictive models. As a result, audio analysis feature data were unavailable for personal playlists whose themes included folk, country and R & B. To work around this challenge, I found 4 datasets on Kaggle that had been pulled previously. The first two of these datasets were the top 100 songs on Spotify in 2017 and 2018, the third was the top 50 songs on Spotify in 2019 and the last was 30,000 songs from various playlists from 2023 and earlier. [3,4,5,6] The first three playlists were used in the actual analysis and the last was used as a reference dataset to understand the general correlations and distributions of features.

Features of the tracks included the following: danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence and tempo. [7] All of these features were derived through audio analysis at Spotify and are based on features of the data such as rhythm, tempo, sound quality and music theory. I performed some preliminary data analysis on the reference dataset to better understand the relationships between these features. See figure 1 for a correlation heatmap of the data features. It is reasonable that the features tend not to be correlated as this is likely by design. If these features were to be used exclusively, they would have the best utility being constructed with the lack of correlation in mind. Despite this, some relationships remain, mostly expectedly. Valence and Danceability have a slight correlation which makes sense as valence represents a measure of the positivity of a song; it is reasonable that one would want to dance to a happier song as opposed to a sad song. The correlation of loudness and energy also make sense since energy is a complex measure that takes loudness into account. High energy songs tended to be those that could be found in a workout class or playlist. One last major relationship is the anticorrelation of acousticness and energy which makes sense since songs likely to be acoustic tend to be found in low energy environments such as coffee shops.

I also plotted the histograms of each feature to get a sense of the distribution of each feature. Acousticness, Instrumentalness and liveness all tend to favor 0 as these features are confidence measures of how acoustic, instrumental and live each track is. Figures 4, 6 and 8 all seem to maintain the general shapes of the reference dataset.

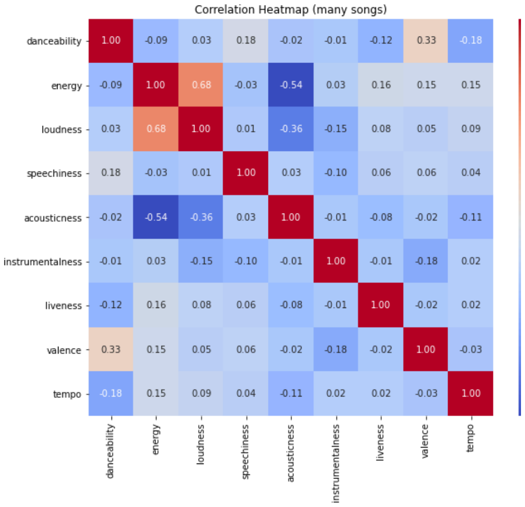


Figure 1: Correlation heatmap of 30,000 track dataset features.

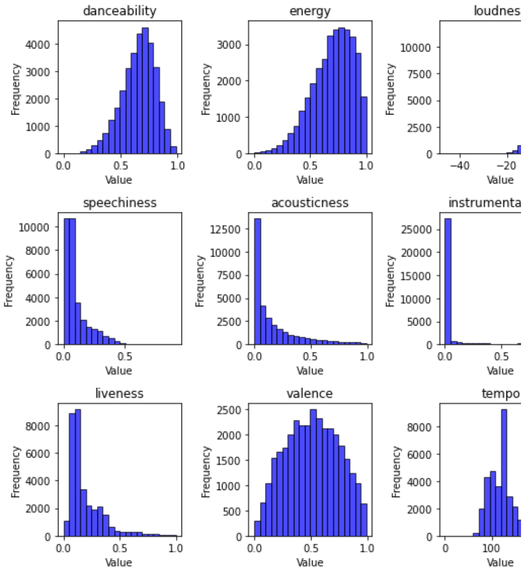


Figure 2: Histograms of each feature of 30,000 track dataset.

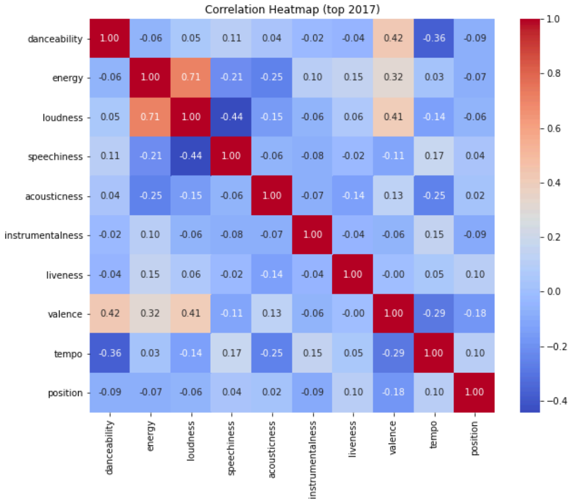


Figure 3: Correlation heatmap of of 2017 dataset features.

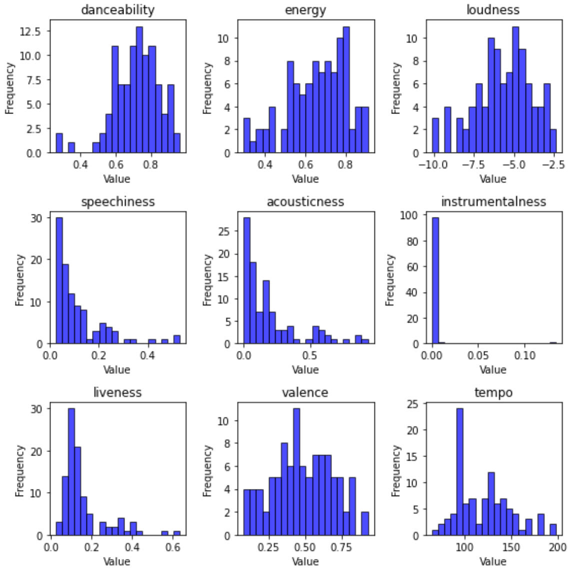


Figure 4: Histograms of each feature of 2017 dataset.

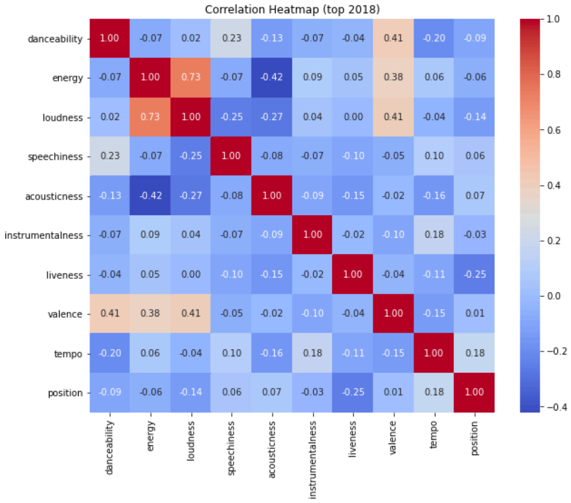


Figure 5: Correlation heatmap of 2018 dataset features.

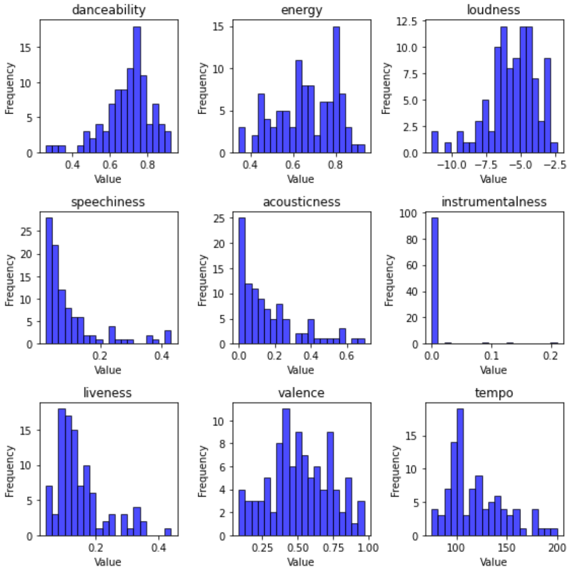


Figure 6: Histograms of each feature of 2018 dataset.

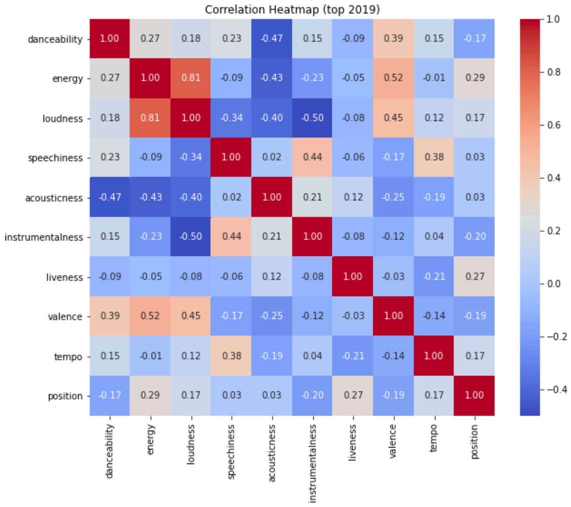


Figure 7: Correlation heatmap of 2019 dataset features.

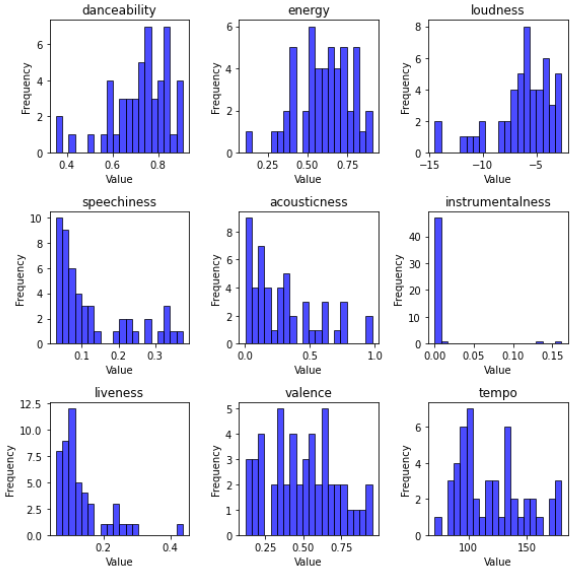


Figure 8: Histograms of each feature of 2019 dataset.

3) Methods

This project uses three main approaches each with their own strengths, weaknesses and unique perspectives. These three approaches include the use of K-Means clustering, agglomerative hierarchical clustering and Gaussian Mixture Models (GMM). These were performed using Sci-kit Learn. [8] Visualizations were made with Matplotlib and

Seaborn. [9,10] I also centered and scaled the data using z-scoring.

For each of the clustering methods, I used a UMAP projection to visually evaluate the grouping of cluster elements and Jaccard similarity to quantitatively evaluate cluster similarity. The Jaccard similarity score between two sets is defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Where:

- A and B are two sets.
- $|A \cap B|$ is the number of elements in the intersection of A and B.
- $|A \cup B|$ is the number of elements in the union of A and B.

The first approach K-Means serves as a starting point for the analysis as it provides potentially informative centroids. [11] An overview of the algorithm is as follows:

1. Initialize k centroids from the data points.
2. Assign each data point to the nearest centroid (in this implementation, based on the Euclidean distance).
3. Recalculate centroids based on the mean of each cluster.
4. Repeat steps 2 and 3 until a predefined set of iterations run or the centroids no longer change significantly.

This algorithm minimizes the within cluster sum of squared distances which can be found below.

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

Where:

- J is the total within-cluster sum of squares (objective function).
- k is the number of clusters.
- C_i is the set of points in cluster i.
- μ_i is the centroid of cluster i.
- $\|x - \mu_i\|^2$ is the squared Euclidean distance between point x and centroid μ_i.

I selected the optimal k using the elbow method whereby I fit each of the years on 2 to 29 values of k. This decision was informed by the size of the population as average cluster sizes below 4 would not be particularly informative. Using the elbow method, I determined that roughly 11 clusters would be sufficient to properly cluster the data. After this point, the within cluster sum of squares tended to be linear so additional clusters would potentially overfit the data.

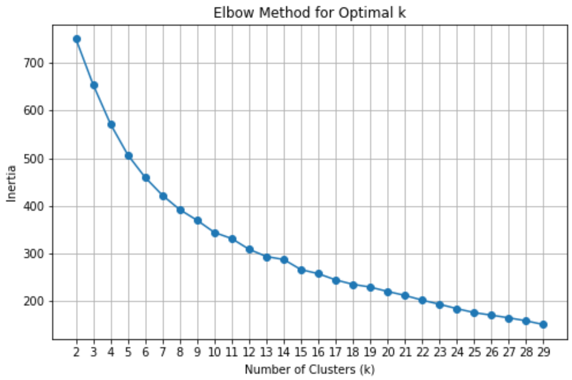


Figure 9: Elbow plot for 2017 data.

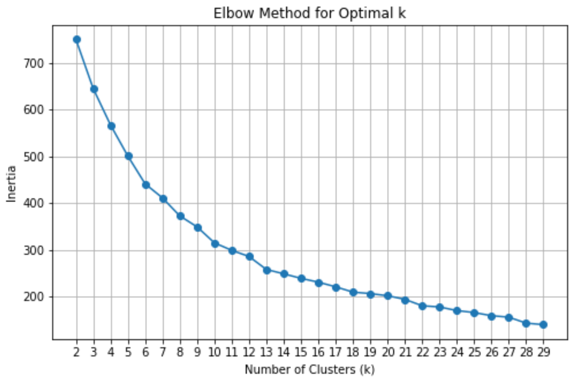


Figure 10: Elbow plot for 2018 data.

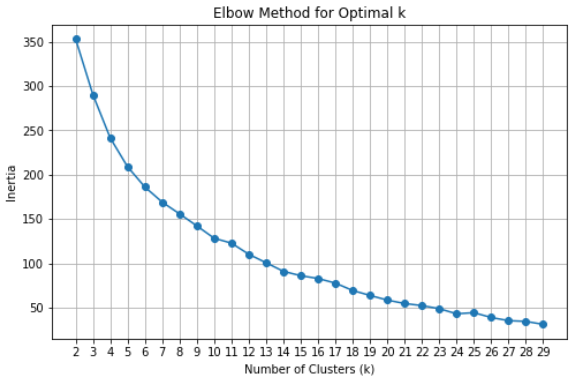


Figure 11: Elbow plot for 2019 data.

I was most interested in this approach as clusters can readily be labeled with the track closest to the centroid. Using this reasoning I used 2 separate methods to apply K-Means clustering to the data. The first method involved concatenating the data from the 3 years of interest and then applying K-Means clustering. The second method involved generating clusters for each year, and grouping clusters from each year based on minimum distance. I framed the trios of clusters as an assignment problem based on minimum overall distance and merged the clusters that formed from each year. A possible weakness of this approach is the assumption that clusters are centered around centroids rather than in more abstract shapes based on feature relationships.

The second approach I used was agglomerative hierarchical clustering. [12] This method is useful because it provides a visual dendrogram by which I could prune the tree for optimal subclusters. An overview of the algorithm is as follows:

1. Initialize every point as its own cluster
2. Compute pairwise distances between points (Euclidean in my implementation)
3. Merge the closest clusters
4. Update distances between new cluster and all clusters based on the following:

$$D(C_i, C_j) = \left(\frac{|C_i| \cdot |C_j|}{|C_i| + |C_j|}\right) \|\mu_i - \mu_j\|^2$$

Where:

- $|C_k|$ is the number of elements in cluster k.
- μ_i is the mean of cluster i.

5. Repeat 2-4 until all clusters have been merged

This approach was primarily of interest due to its informative dendrogram and identification of subclusters. I determined the optimal number of clusters based on the first major split. This resulted in 6 major clusters being identified. One weakness of this method however is the granularity of analysis necessary for determining and interpreting subclusters.

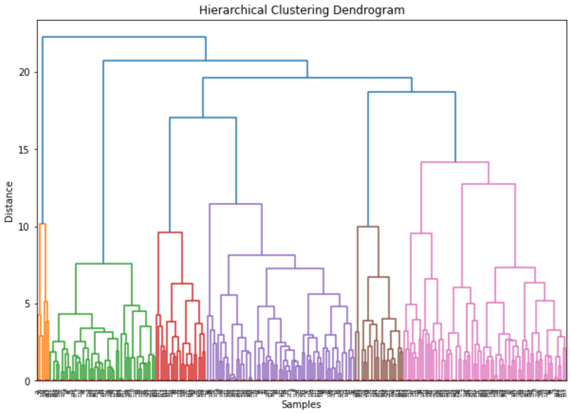


Figure 12: Dendrogram of combined data. Note: I was unable to properly label the x-ticks due to the large number of samples. This contributed to challenges with the approach.

The third and final approach used was Gaussian Mixture Models. [13] This algorithm assumes the data consist of a number of Gaussian distributions and attempts to learn the parameters using the Expectation Maximization algorithm. I was particularly interested in the probabilistic nature of this algorithm as it provides an uncertainty metric for each cluster assignment. An overview of the algorithm is as follows:

1. Determine the number of components (k).
2. Initialize the parameters of the Gaussian with guesses:
    - ○ Means: μ_k
    - ○ Covariances: Σ_k
    - ○ Mixing coefficients: π_k which are the initial probabilities of each cluster
3. Perform Expectation Maximization algorithm:

4. Expectation step:

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)}$$

Where:

- $\mathcal{N}(x_i \mid \mu_k, \Sigma_k)$ is the Gaussian probability density function for cluste given by:

$$\mathcal{N}(x_i \mid \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1}(x_i - \mu_k)\right)$$

- π_k is the mixing coefficient for cluster k.
- μ_k is the mean (centroid) of cluster k.
- Σ_k is the covariance matrix of cluster k.
- γ_{ik} is the responsibility of cluster k for data point x_i.
- K is the total number of clusters.

5. Maximization step:

- **Mixing Coefficient:**

$$\pi_k = \frac{\sum_{i=1}^{n} \gamma_{ik}}{n}$$

- **Mean:**

$$\mu_k = \frac{\sum_{i=1}^{n} \gamma_{ik} x_i}{\sum_{i=1}^{n} \gamma_{ik}}$$

- **Covariance:**

$$\Sigma_k = \frac{\sum_{i=1}^{n} \gamma_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{n} \gamma_{ik}}$$

Where:

- n: Total number of data points.

6. Repeat step 4 and 5 for a predetermined set of iterations or the log likelihood of the data stops improving significantly.

The uncertainty can further be obtained by taking the complementary probability of the responsibility for each assignment. In other words, I subtracted the maximum responsibility (posterior probability) for each cluster assignment from 1 to get a numerical value of uncertainty. I determined the optimal value for k by running the algorithm with each value from 2 to 20 until the cumulative uncertainty for each cluster was below 0.05. This critical value was somewhat arbitrary although based on the concept of significance in statistical tests. 5 clusters was deemed optimal using this method.

4) Results

Below are the UMAP visualizations and Jaccard similarity matrices.

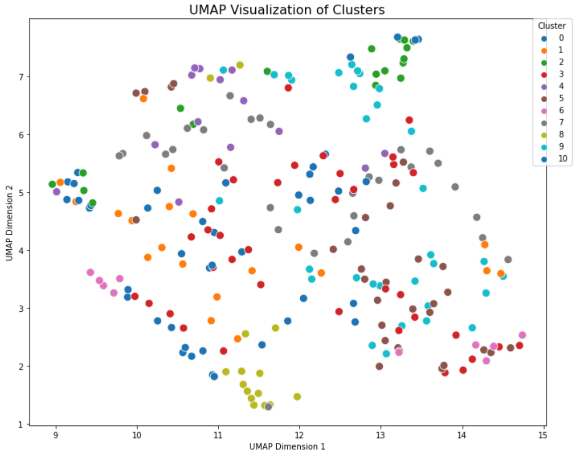


Figure 13: UMAP visualization of K-Means clustering using alignment of centroids.

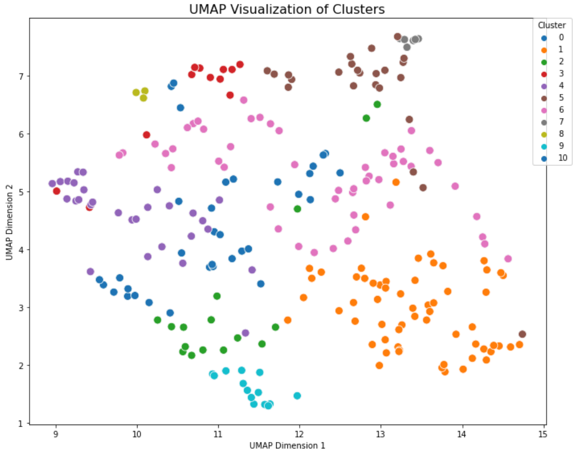


Figure 14: UMAP visualization of K-Means clustering with concatenation.

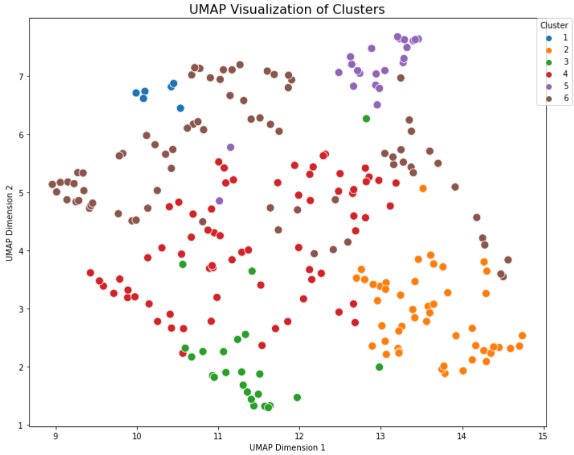


Figure 15: UMAP visualization of agglomerative hierarchical clustering.

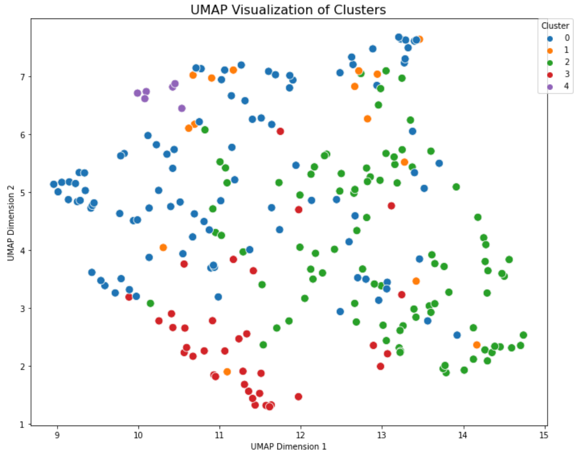
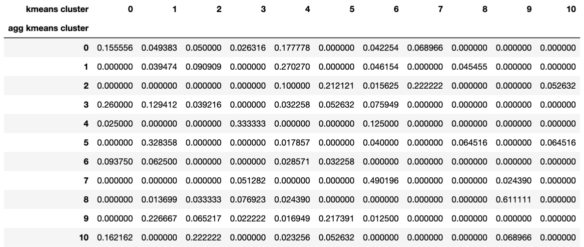
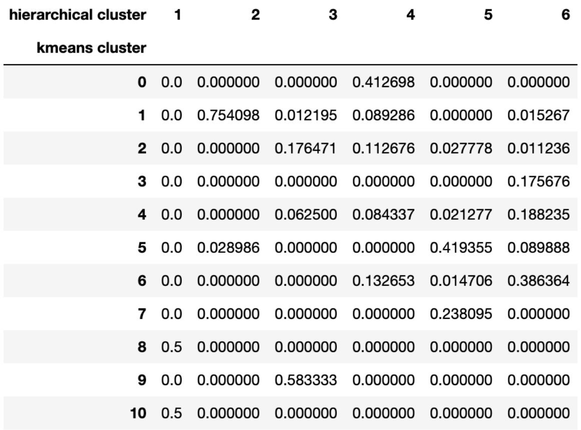


Figure 16: UMAP visualization of GMM clusters.



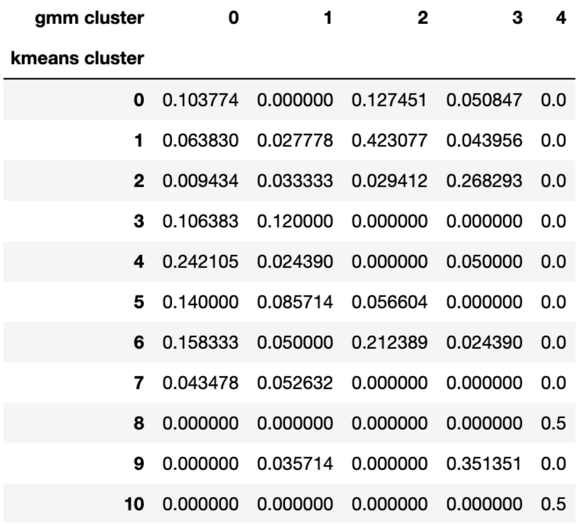
kmeans cluster	0	1	2	3	4	5	6	7	8	9	10
agg kmeans cluster											
0	0.155556	0.049383	0.050000	0.026316	0.177778	0.000000	0.042254	0.068966	0.000000	0.000000	0.000000
1	0.000000	0.039474	0.090909	0.000000	0.270270	0.000000	0.046154	0.000000	0.045455	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000	0.100000	0.212121	0.015625	0.222222	0.000000	0.000000	0.052632
3	0.260000	0.129412	0.039216	0.000000	0.032258	0.052632	0.075949	0.000000	0.000000	0.000000	0.000000
4	0.025000	0.000000	0.000000	0.333333	0.000000	0.000000	0.125000	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.328358	0.000000	0.000000	0.017857	0.000000	0.040000	0.000000	0.064516	0.000000	0.064516
6	0.093750	0.062500	0.000000	0.000000	0.028571	0.032258	0.000000	0.000000	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.051282	0.000000	0.000000	0.490196	0.000000	0.000000	0.024390	0.000000
8	0.000000	0.013699	0.033333	0.076923	0.024390	0.000000	0.000000	0.000000	0.000000	0.611111	0.000000
9	0.000000	0.226667	0.065217	0.022222	0.016949	0.217391	0.012500	0.000000	0.000000	0.000000	0.000000
10	0.162162	0.000000	0.222222	0.000000	0.023256	0.052632	0.000000	0.000000	0.000000	0.068966	0.000000

Table 1: Jaccard similarity between aligned clusters (agg kmeans cluster) and clusters with concatenation (kmeans cluster)



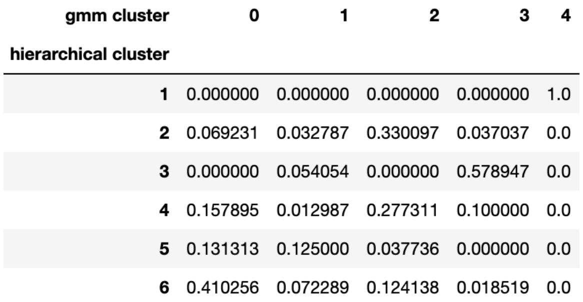
hierarchical cluster	1	2	3	4	5	6
kmeans cluster						
0	0.0	0.000000	0.000000	0.412698	0.000000	0.000000
1	0.0	0.754098	0.012195	0.089286	0.000000	0.015267
2	0.0	0.000000	0.176471	0.112676	0.027778	0.011236
3	0.0	0.000000	0.000000	0.000000	0.000000	0.175676
4	0.0	0.000000	0.062500	0.084337	0.021277	0.188235
5	0.0	0.028986	0.000000	0.000000	0.419355	0.089888
6	0.0	0.000000	0.000000	0.132653	0.014706	0.386364
7	0.0	0.000000	0.000000	0.000000	0.238095	0.000000
8	0.5	0.000000	0.000000	0.000000	0.000000	0.000000
9	0.0	0.000000	0.583333	0.000000	0.000000	0.000000
10	0.5	0.000000	0.000000	0.000000	0.000000	0.000000

Table 2: Jaccard similarity between K-Means clusters with concatenation and agglomerative hierarchical clusters



gmm cluster	0	1	2	3	4
kmeans cluster					
0	0.103774	0.000000	0.127451	0.050847	0.0
1	0.063830	0.027778	0.423077	0.043956	0.0
2	0.009434	0.033333	0.029412	0.268293	0.0
3	0.106383	0.120000	0.000000	0.000000	0.0
4	0.242105	0.024390	0.000000	0.050000	0.0
5	0.140000	0.085714	0.056604	0.000000	0.0
6	0.158333	0.050000	0.212389	0.024390	0.0
7	0.043478	0.052632	0.000000	0.000000	0.0
8	0.000000	0.000000	0.000000	0.000000	0.5
9	0.000000	0.035714	0.000000	0.351351	0.0
10	0.000000	0.000000	0.000000	0.000000	0.5

Table 3: Jaccard similarity between clusters with concatenation (kmeans cluster) and GMM clusters



gmm cluster	0	1	2	3	4
hierarchical cluster					
1	0.000000	0.000000	0.000000	0.000000	1.0
2	0.069231	0.032787	0.330097	0.037037	0.0
3	0.000000	0.054054	0.000000	0.578947	0.0
4	0.157895	0.012987	0.277311	0.100000	0.0
5	0.131313	0.125000	0.037736	0.000000	0.0
6	0.410256	0.072289	0.124138	0.018519	0.0

Table 4: Jaccard similarity between agglomerative hierarchical and GMM clusters

5) Discussion

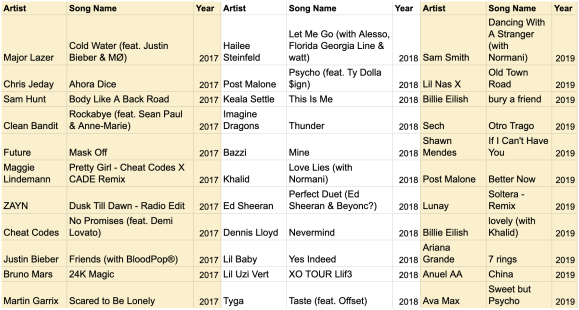
While global distances are relatively unreliable in a UMAP projection, local distances tend to accurately portray neighbors. One of the first major conclusions we can draw from figures 13 and 14, is that the alignment method failed to accurately capture sound groups from year to year. This is supported by table 1 where the maximum similarity between each cluster rarely goes above 0.3. Based on the UMAP projection, I would favor the concatenation approach to data preprocessing. Considering tables 5 and 6, certain songs such as *XO TOUR Llif3* by Lil Uzi Vert, *Thunder* by Imagine Dragons and *China* by Anuela AA appear near cluster centroids for both approaches. This might suggest some consistency albeit slight.

Figures 15 and 16 show limited success in the hierarchical and GMM approaches as elements with close local distances appear in mixed clusters.

Table 2 shows moderate levels of similarity among the clusters with the most similar K-Means cluster having an average similarity of about 0.5 with each hierarchical cluster.

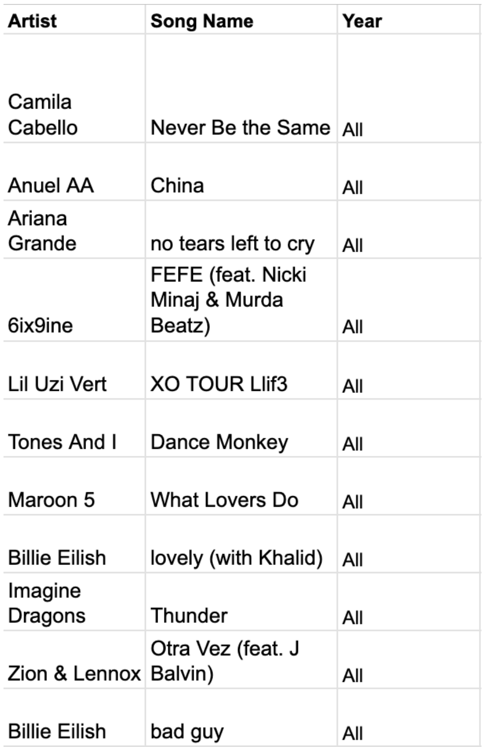
There were several challenges I faced during this project. Perhaps the most difficult was determining what constituted a "good" cluster. I used Jaccard similarity to compare across approaches but this is better suited for comparison than actually evaluating the success of the methods. Further, any success associated with the clustering methods is more likely attributed to the advanced level of audio analysis rather than the methods themselves. I also had some trouble evaluating the clusters as I had not heard of some of the songs and any analysis would be relatively subjective. An in depth analysis of each cluster requires a level of granularity that would be difficult to understand. This meant the hierarchical clustering approach was very difficult to interpret from a practical sense. The subclusters provide interesting information pertaining to the relationships between songs but the subclusters were difficult to ascertain with confidence. Additionally, as the number of clusters increased the specificity of the cluster became increasingly difficult to label from a human perspective.

Speaking from a less quantitative standpoint, however, the features do a good job of identifying intangible "vibes". These clusters tend to transcend genre as hip-hop songs are the songs closest to each centroid. A notable overlap in clusters appears in cluster 4 of the GMM and cluster 1 of the hierarchical clusters. It is particularly interesting that this cluster includes *Thunder* by Imagine Dragons, *Passionfruit* by Drake and *Bad Guy* by Billie Eilish; these are songs that might not be grouped together stylistically but make sense upon listening.



Artist	Song Name	Year	Artist	Song Name	Year	Artist	Song Name	Year
Major Lazer	Cold Water (feat. Justin Bieber & MØ)	2017	Hailee Steinfeld	Let Me Go (with Alesso, Florida Georgia Line & watt)	2018	Sam Smith	Dancing With A Stranger (with Normani)	2019
Chris Jeday	Ahora Dice	2017	Post Malone	Psycho (feat. Ty Dolla $ign)	2018	Lil Nas X	Old Town Road	2019
Sam Hunt	Body Like A Back Road	2017	Keala Settle	This Is Me	2018	Billie Eilish	bury a friend	2019
Clean Bandit	Rockabye (feat. Sean Paul & Anne-Marie)	2017	Imagine Dragons	Thunder	2018	Sech	Otro Trago	2019
Future	Mask Off	2017	Bazzi	Mine	2018	Shawn Mendes	If I Can't Have You	2019
Maggie Lindemann	Pretty Girl - Cheat Codes X CADE Remix	2017	Khalid	Love Lies (with Normani)	2018	Post Malone	Better Now	2019
ZAYN	Dusk Till Dawn - Radio Edit	2017	Ed Sheeran	Perfect Duet (Ed Sheeran & Beyonc?)	2018	Lunay	Soltera - Remix	2019
Cheat Codes	No Promises (feat. Demi Lovato)	2017	Dennis Lloyd	Nevermind	2018	Billie Eilish	lovely (with Khalid)	2019
Justin Bieber	Friends (with BloodPop®)	2017	Lil Baby	Yes Indeed	2018	Ariana Grande	7 rings	2019
Bruno Mars	24K Magic	2017	Lil Uzi Vert	XO TOUR Llif3	2018	Anuel AA	China	2019
Martin Garrix	Scared to Be Lonely	2017	Tyga	Taste (feat. Offset)	2018	Ava Max	Sweet but Psycho	2019

Table 5: Songs closest to each centroid in each year



Artist	Song Name	Year
Camila Cabello	Never Be the Same	All
Anuel AA	China	All
Ariana Grande	no tears left to cry	All
6ix9ine	FEFE (feat. Nicki Minaj & Murda Beatz)	All
Lil Uzi Vert	XO TOUR Llif3	All
Tones And I	Dance Monkey	All
Maroon 5	What Lovers Do	All
Billie Eilish	lovely (with Khalid)	All
Imagine Dragons	Thunder	All
Zion & Lennox	Otra Vez (feat. J Balvin)	All
Billie Eilish	bad guy	All

Table 6: Songs closest to each centroid concatenating all years

6) Future Work

There are 2 main directions I would take an expansion of this project, playlist production and feature extraction.

In the future it would be interesting to examine each cluster and build playlists from a larger reference dataset. I experimented with this approach but it ultimately seemed outside of the scope of the project.

It may be an interesting exercise to manually group songs together and derive unique features from those human-assisted clusters. This might require actual audio files or a large scale labeling effort of songs. One such way to do this is providing a defined number of fundamental groups and providing labellers with a confidence scale, perhaps of 1-5, of how well a given song fits in a playlist/cluster.

A further expansion of this might be a graph based song recommender system. I envision this system randomly selecting neighbors using a multi-armed bandit approach where engagement metrics such as skips and subsequent song choices can influence exploitation and exploration.

The directions that this problem space can go are seemingly endless and I would welcome the opportunity to explore them.

7) References

[1] Publisher, Author removed at request of original. *6. 2 the Evolution of Popular Music*. Mar. 2016. *open.lib.umn.edu*, https://open.lib.umn.edu/mediaandculture/chapter/6-2-the-evolution-of-popular-music/.

[2] Mauch, Matthias, et al. “The Evolution of Popular Music: USA 1960–2010.” *Royal Society Open Science*, vol. 2, no. 5, May 2015, p. 150081. *DOI.org (Crossref)*, https://doi.org/10.1098/rsos.150081.

[3] *Top Spotify Tracks of 2017*. https://www.kaggle.com/datasets/nadintamer/top-tracks-of-2017. Accessed 21 Dec. 2024.

[4] *Top Spotify Tracks of 2018*. https://www.kaggle.com/datasets/nadintamer/top-spotify-tracks-of-2018. Accessed 21 Dec. 2024.

[5] *Top Spotify Tracks of 2019*. https://www.kaggle.com/datasets/nadintamer/top-spotify-tracks-of-2019. Accessed 21 Dec. 2024.

[6] *30000 Spotify Songs*. https://www.kaggle.com/datasets/joebeachcapital/30000-spotify-songs. Accessed 21 Dec. 2024.

[7] *Web API Reference | Spotify for Developers*. https://developer.spotify.com/documentation/web-api/reference/get-audio-features. Accessed 21 Dec. 2024.

[8] *Scikit-Learn: Machine Learning in Python — Scikit-Learn 1.6.0 Documentation*. https://scikit-learn.org/stable/. Accessed 21 Dec. 2024.

[9] *Matplotlib — Visualization with Python*. https://matplotlib.org/. Accessed 21 Dec. 2024.

[10] Waskom, Michael. “Seaborn: Statistical Data Visualization.” *Journal of Open Source Software*, vol. 6, no. 60, Apr. 2021, p. 3021. *DOI.org (Crossref)*, https://doi.org/10.21105/joss.03021.

[11] MacQueen, J. “Some Methods for Classification and Analysis of Multivariate Observations.” *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, vol. 5.1, University of California Press, 1967, pp. 281–98. *projecteuclid.org*, https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings-of-the-Fifth-Berkeley-Symposium-on-Mathematical-Statistics-and/chapter/Some-methods-for-classification-and-analysis-of-multivariate-observations/bsmsp/1200512992.

[12] Ward, Joe H. “Hierarchical Grouping to Optimize an Objective Function.” *Journal of the American Statistical Association*, vol. 58, no.

301, Mar. 1963, pp. 236–44. *DOI.org (Crossref)*, https://doi.org/10.1080/01621459.1963.10500845.

[13] *Gaussian Mixture Model - an Overview | ScienceDirect Topics*. https://www.sciencedirect.com/topics/engineering/gaussian-mixture-model. Accessed 21 Dec. 2024.