Machine Learning as a Lens on Song Analysis: An Unsupervised Clustering Approach

Project Group: 149

F.R. Signorelli **R.M.A. van Nee** **P.M. Ketterman** **A.C.T.D. Ramautar** **V. Li**

Abstract

In this report, our project group explores using unsupervised clustering techniques on various characteristics found in the Spotify API to identify different music genres. The ultimate goal of this report is to demonstrate how unsupervised clustering can be applied to analyze and classify music using the rich datasets available through the Spotify API. This report aims to reach an audience with some knowledge of the core concepts of machine learning processes and algorithms in general.

1 Introduction

Unsupervised systems have received considerable attention in machine learning as they provide a unique approach to data analysis. This unique approach is realized by the system identifying underlying patterns and structures without explicit labels. Conversely, most existing research has focused on supervised learning methods, which require labeled data for training. Unsupervised techniques offer an opportunity to gain new understanding in complex and unlabeled datasets (1). In this paper, we explore the potential of unsupervised clustering in analyzing musical data obtained from Spotify, one of the most well-liked music streaming services with more than 365 million monthly active users (2) (3). Through a comparative study, we aim to provide insight into music analysis using different clustering techniques and contrast the results with human-defined genre classifications.

The music world presents a complex and diverse set of data, traditionally analyzed using domain-specific knowledge, such as music theory or cultural context. Unsupervised learning systems offer a complementary approach to that traditional knowledge, allowing for the discovery of emergent patterns and relationships by examining the implicit structure of the data. We will use and compare multiple clustering techniques, applying these algorithms to a robust set of numerical acoustic features and even employing a technique that explores a linguistic approach to categorization. By providing, comparing, and studying these results, we intend to gain knowledge of the musical data in addition to commonalities and differences in human-made and machine-made music analysis and its genre classification.

2 Data Inspection and Preparation

2.1 The Dataset

In today's data-driven research, datasets play an essential role. There are many different types of datasets, from small, regulated datasets to large, complex, and unstructured ones. Both the public and private sectors are publishing data, which spans a wide range of industries, from the life sciences to the media or government data. Despite the fact that datasets are more widely available, it can still be difficult for researchers to discover ones that are appropriate for their particular research objectives. The supplied datasets might not always be suitable for the study topic (4).

2.2 Data Inspection

The dataset was made by merging a dataset found on Kaggle and adding to this some of our own Spotify tracks. The original Kaggle dataset has 114001 unique Spotify tracks over a range of 125 different genres (5). The data is in CSV format and each track has some audio features. Audio features are a set of measurements that describe various characteristics of a piece of music. The audio features which were included in the Kaggle dataset were the track id, artist, album name, track name, popularity, duration in milliseconds, if the track is explicit, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, time signature and track genre. A description of each feature can be found in the appendix.

The graph depicting the top 20 most frequently occurring genres in our dataset can be found in Figure 1. It is noteworthy that the majority of genres exhibit analogous proportions, as the initial dataset evinced similar percentages among the various genres.

3 Data Inspection and Preparation

3.1 The Dataset

In today's data-driven research, datasets play an essential role. The public and private sectors publish data spanning various industries, from the life sciences to the media or government data. There are many forms a dataset can take, from small regulated datasets to large, complex, and unstructured ones. Although datasets of significant size are more widely available, it can still be difficult for researchers to acquire clean data appropriate for their particular research objectives. The supplied datasets might not always be well suited to the study topic (4).

3.2 Data Inspection

The dataset in this project was created by merging a dataset found on the website Kaggle and then adding a sample of Spotify tracks from playlists contributed by our project group members. The original Kaggle dataset has 114001 unique Spotify tracks over a range of 125 different genres (5). The data is in Comma-Separated Value (CSV) format, and each track has multiple distinct audio features. Audio features, in this case, are a set of measurements that describe specific characteristics of a piece of music. The audio features included in the Kaggle dataset were the track ID, artist, album name, track name, popularity, duration in milliseconds, if the track is explicit, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, time signature and track genre. The audio features comprise several data types, including boolean (true/false), integers (whole numbers), floating-point decimals (numbers with decimal values), and text strings (titles and sentences). Descriptions of each feature are in the appendix.

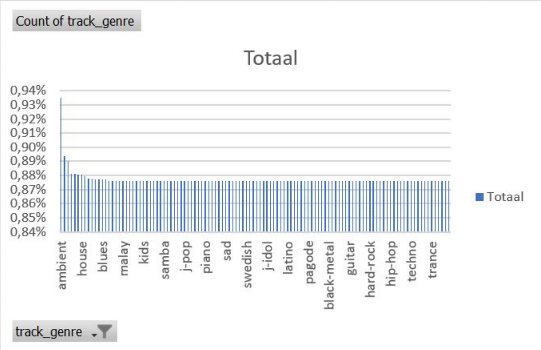

Figure 1: Top 20 Genres in Dataset

The graph depicting our dataset's top 20 most frequently occurring genres is in Figure 1. Notably, most genres exhibit similar proportions, as the initial dataset indicated similar percentages among the various genres.

The genre 'ambient' occurs most frequently. Beyond that, the genres 'house' and 'blues' occur second and third most often.

3.3 Data Preparation

The dataset used for this study required tremendous alteration before it was ready for use in our clustering experiments. The original Kaggle dataset included a relatively equally distributed amount of tracks for each genre. In order to increase the variability of our dataset and avoid having the same number of songs for every different genre, we intentionally introduced noise to make the dataset more heterogeneous. This decision was based on the understanding that outliers can hold significant importance. We integrated our data using the Spotify API (Application Programming Interface) (6). The Spotify API is a powerful tool that enables developers to access Spotify's vast music library and integrate Spotify features into their applications.

The final dataset contains 115382 original tracks, of which 114001 tracks are from the original Kaggle dataset, and the other 1381 tracks originated in our playlists. We initially tried to add more than 3000 tracks to the dataset, but we only managed to add 46% of this successfully. This shortcoming is due to limitations and errors we observed while retrieving the data using Spotify's API endpoints.

Apart from the number of original tracks, the original Kaggle dataset's audio features also needed altering. Modifying the dataset was necessary because we sought numerical features almost exclusively in our study since categorical features are notoriously difficult to measure against numerical data. Below is described how we decided which features to include and exclude:

Some features we obtained from Spotify were already in decimal form from 0 to 1, such as danceability, speechiness, and instrumentalness. Others, such as loudness, tempo, or duration, were numerical but had different ranges that we had to normalize so that all features could be combined without having too many different weights and skewing the results. We eliminated the following features that were not numerical: the artist name, album name, track name, and track genre were all formatted as text strings, and the single true/false value of whether the track contained explicit language. The explicit language feature could be numerical, but we decided not to include it because we did not find it necessary for our research, just like the numerical feature track ID.

There were some tricky features as well for which we had to decide whether to remove them or keep them. The 'mode' signifies minor or major in terms of musical key, and we decided to keep this because we judged it to be a valuable feature, and it was already binary (0 or 1). A feature represented by numbers but not numerical was 'key.' It indicates the song's key, using different numbers to indicate different notes; however, there is no correspondence between notes in the manner expected between numbers. We judged it best to remove it as we removed explicitness, album name, and artist name, all categorical features we did not need because they made the dataset heavier and more difficult to cluster.

We decided not to use one-hot encoding to turn categorical features into numerical ones since it would have significantly increased the sparsity of our dataset and the number of features. Furthermore, since all features resulting from one-hot encoding would be either 0 or 1, they would have more weight in the dataset than those with decimal values between 0 and 1.

4 Research question

The primary goal of this study is to investigate and compare unsupervised numerical clustering, title-based unsupervised clustering, and human-made genre classification. As a result, the research question best formulated to capture the core of this user study is thus: *"What is the relative effectiveness of unsupervised numerical clustering and title-based unsupervised clustering techniques given pre-defined human-made genre classification?"*. The study endeavors to provide insights that would be of interest to researchers studying musicology, academics, and practitioners in the field of machine learning.

5 Methods

5.1 Numerical Clustering: K-Means and EM

While considering the possible techniques to implement the unsupervised clustering, which included hierarchical clustering and DBSCAN, we chose two straightforward and logical approaches to our problem: K-Means and Expectation–maximization (EM). An important reason for this was to obtain clean and simple cluster divisions we would be able to visualize and understand. K-Means works by iteratively computing K-centroids (representative centers of the cluster) and assigning every data point to one of them, then picking the new mean of every cluster and making it a centroid for the latest iteration until the means converge with the centroids. One possible limitation of the algorithm is that the final clustering result depends on the random initial centroids. To circumvent this, we used the variant K-Means++ which selects centroids sequentially, computing the squared distance between each data point and its nearest centroid and choosing the next centroid with probability proportional to this distance. EM treats data as a combination of Gaussian distributions and estimates their parameters by finding the best fit for the observed data. The algorithm alternates between expectation and maximization, using probabilities to assign data points to Gaussian distributions and refining distribution parameters and proportions until convergence.

Remembering that EM has the potential for more variegate shapes and distributions, the two algorithms are similar in their iterative clustering systems based on n numerical features. We are utilizing both, expecting their similar outputs to make our results more robust. Principal Component Analysis (PCA) is also applied to obtain and compare these results. PCA is used to reduce the dimensionality of data by identifying patterns and relationships among variables. PCA transforms a set of correlated variables into a smaller set of uncorrelated variables, known as principal components, while retaining as much of the original information as possible.

5.2 Picking the Number of Clusters

As the methods are unsupervised, we do not need to divide our dataset into training and test data, as there is no target feature to predict, and thus error calculation is not required. We must look for and select the hyper-parameter *k* indicating the number of different clusters. An optimal number of *k* should make the clustering compact, separate, and generalizable. A suitable measure for compactness is the within-cluster sum of squares (WCSS), which represents how distant the points of each cluster are from each other (7).

The most popular technique to find the optimal *k* in such situations is the Elbow method, which consists of plotting the different possible numbers of clusters *k* against their corresponding WCSS (8). By observing Figure 2, one should be able to find an elbow point that indicates the maximal *k*, after which adding more clusters does not lead to significant improvement.

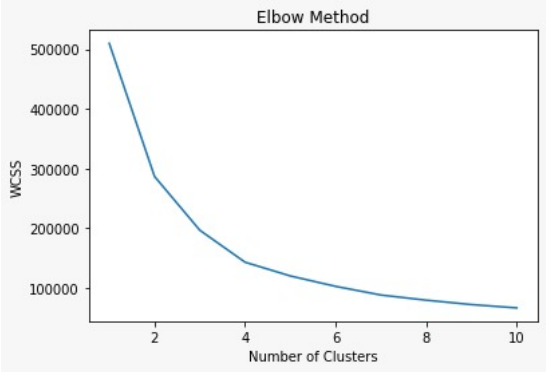

Figure 2: Elbow Visualisation

In this picture, obtained from applying the method on our dataset, it is visible that the elbow point is k=4, so we picked four as the ideal number of clusters. We chose to keep *k* constant for all the different clustering methods since the goal is to compare our cluster results.

5.3 Title-Based Clustering: LDA

The Latent Dirichlet Allocation (LDA) algorithm, first introduced in 2003, is another unsupervised machine learning method for analyzing text data by discovering the underlying topics in a collection of documents by identifying the standard sets of words that frequently appear associated together in the same document (9).

LDA functions by assuming that each instance in the collection is a mixture of various topics and that each topic is distributed over a set of words. It then tries to find the best set of topics that can explain the words in the documents. This process involves estimating the probabilities of words belonging to topics and topics to instances. Once topics and their associated probabilities are estimated, the algorithm can cluster documents based on their topic distributions.

This algorithm was chosen because we wanted to cover different ways in which machines can perform clustering, not only basing it on numerical features but also on natural language processing. Several studies have demonstrated the effectiveness of LDA-based title clustering in various domains (10).

We expect to see differences in this clustering compared to the numerical one due to the different nature of the technique. However, we set the number of themes and classes *k* to be still equal to four because we wanted the two clustering types to be comparable.

6 Experimental Setup

Throughout this research, we utilized the programming language Python to write and perform our code. Python has many libraries available to aid users in performing operations, and we used several throughout the research. We used the spotipy library to interact easily with Spotify's API: pull the playlists from their database and retrieve all of the features for each song. In all the clustering tasks, we first utilized the Python package pandas to turn our dataset into a data frame we could easily modify. Second, to perform K-Means clustering and EM, we used the modules and classes contained in sklearn, a library dedicated to classic machine learning approaches. From sklearn, we imported the class Standard Scaler to preprocess our data, the Gaussian Mixture and Kmeans classes to perform the EM and the K-Means clustering, and finally, the PCA class to perform the principal component analysis, which was necessary to obtain a 2d visualization of the clustering. To perform LDA, apart from pandas and sklearn, we also used gensim and nltk (Natural Language Toolkit) for processing the phrases that may occur in titles. Gensim is a package that contains functionality to transform strings into vectors and calculate the similarity between documents, which between its modules contains a class called LdaModel we utilized in performing the LDA clustering. Nltk was used to pre-process the data to perform the LDA. This pre-processing included lowercasing, translating, lemmatizing, and tokenizing all the words so they could be processed. We also used Excel and CSV files throughout the project to store datasets, interact with the features, and plot graphs.

7 Result Analysis

The first results we gathered showed a similarity in clustering between the K-Means algorithm and the EM algorithm. To verify it, we used a two-sided approach. On one side, we use PCA to plot the results in two dimensions and observe the clustering and the associated separation boundaries between clusters. As can be viewed in their respective graphs Figure 3 and Figure 4, the two algorithms follow the same general outline, but some clusters take up more space and contain more outliers. It is notable that though the centroid and the exact individual distribution of the clusters do not match, this is not necessarily an indicator that the clusters are dissimilar. Cluster four takes much larger space in EM, allowing it to take a more irregular shape than in K-Means. In contrast, some clusters are very tight and have strongly delineated boundary lines. An example is the cluster in the below-middle area of the graph, cluster one in Figure 3, and cluster four Figure 4. Also, it is interesting to notice that the points more distant from the others in the upper cluster in each graph have a significant difference in classification, being assigned chiefly to the top-left cluster in K-Means and the right one in EM. Only the Top 20 genres in each cluster from EM and LDA are pictured in Figure 6 through Figure 11 due to the genre clustering patterns in EM and K-means being nearly identical. Figure 6 through Figure 9 accompany the plot in Figure 5 and Figure 10 through Figure 13 accompany the plot in Figure 4.

Another method we used to compare EM and K-Means is through correlation metrics between clustering techniques. Measuring through the Adjusted Rand Index (ARI) and the Normalized Mutual Information (NMI), which both range between 0 for no similarity and 1 for perfect coincidence, we obtained respectively 0.88 and 0.84 (11) (12).

As the graph in Figure 5 shows, plotting the principal components of LDA, as predicted, had a very different distribution. Cluster 3

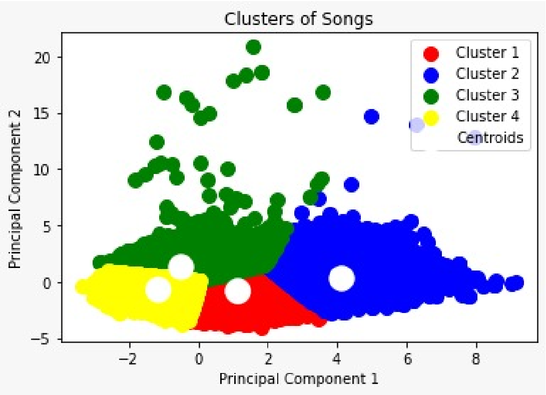

Figure 3: K-Means Clustering Plot

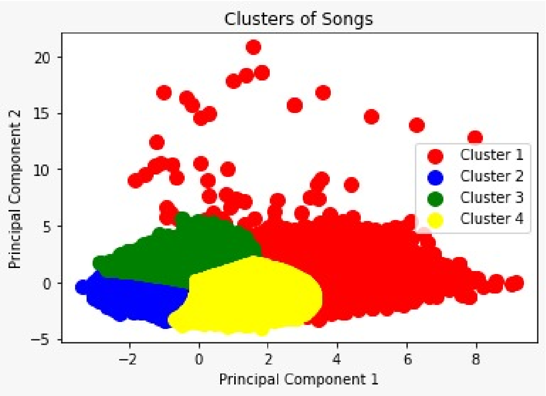

Figure 4: EM Clustering Plot

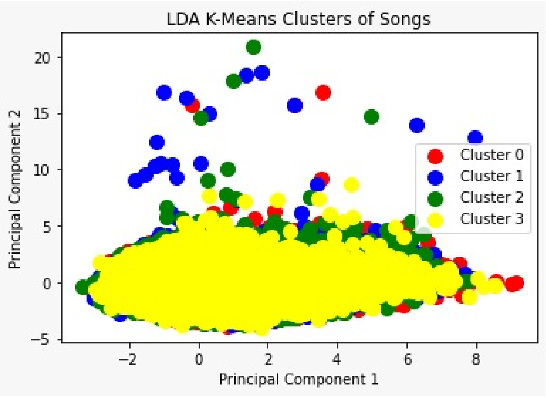

Figure 5: LDA Clustering Plot

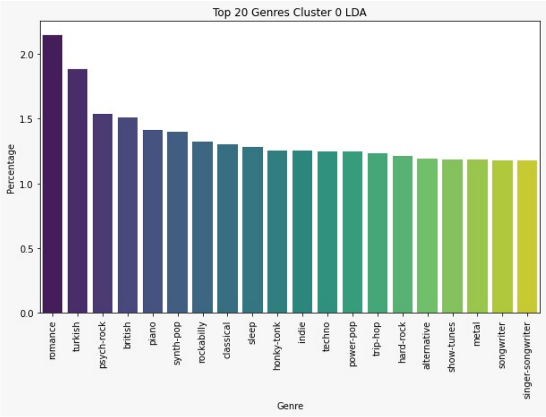

Figure 6: Cluster 0 LDA

was plotted as the last, which explains why it seems so predominant. We notice that the distributions all overlap. No absolute boundaries correspond to the principal components. This lack of boundary makes sense since the K-Means and EM were based on the features from which the principal components were extracted, while LDA was based solely on the unrelated feature of the song title. When we computed ARI and NMI with either K-Means or EM, we obtained 0. Unsurprisingly, the correlation was small, but these meager results pushed us to analyze how the LDA performed even more. We obtained the top 10 topic words for every different cluster, and we noticed that, unfortunately, while some words were legitimate indicators such as ’Christmas’ or ’night’ or ’love,’ many others were just stopwords in other languages, such as ’el’ or ’que,’ and words related to music production such as ’live’ or ’remaster’. It seemed this made LDA useless, but then we plotted the genres against our clustering results. Likely, the stop-words and music production words were included in most of the different genres, and even if they impacted the selection, they did not ruin it. It is also intriguing how the LDA algorithm seemingly divided music based on culture.

8 Discussion

As mentioned above, we adopted an approach that capitalized on existing tools such as spotipy, pandas, and sklearn. Using such tools enabled us to avoid the time-consuming pro-

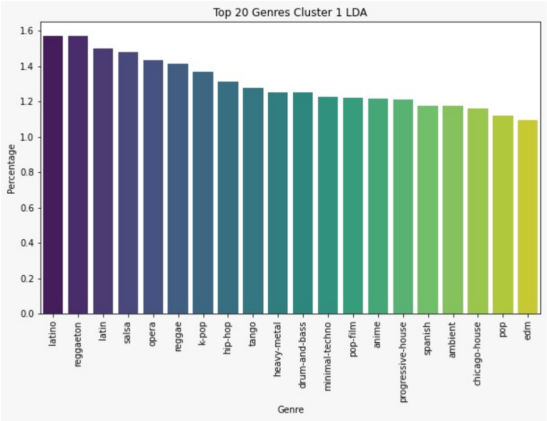

Figure 7: Cluster 1 LDA

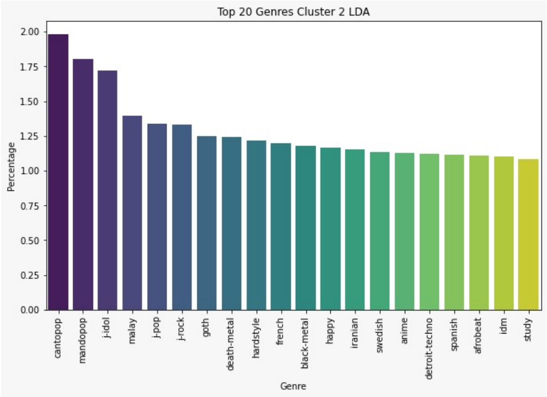

Figure 8: Cluster 2 LDA

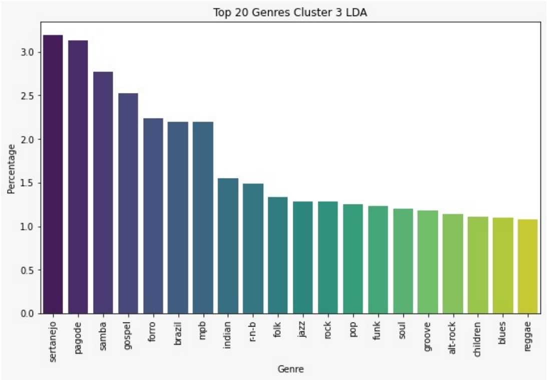

Figure 9: Cluster 3 LDA

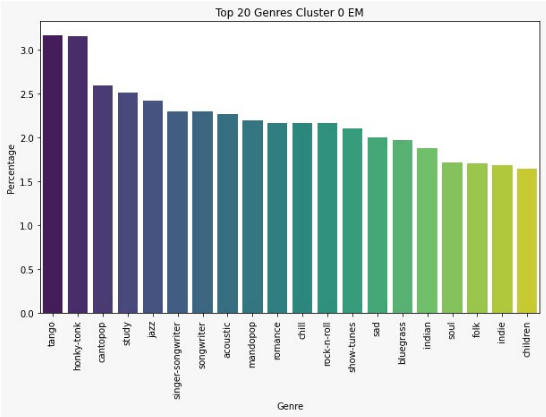

Figure 10: Cluster 0 EM

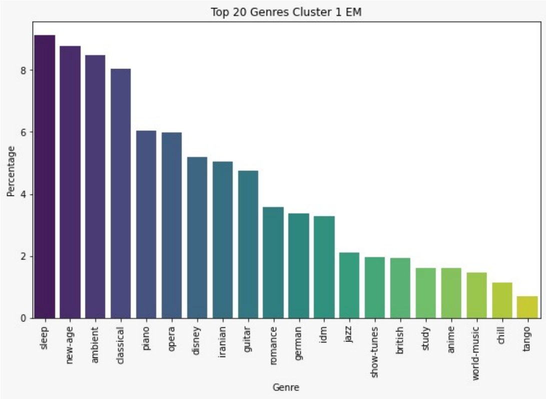

Figure 11: Cluster 1 EM

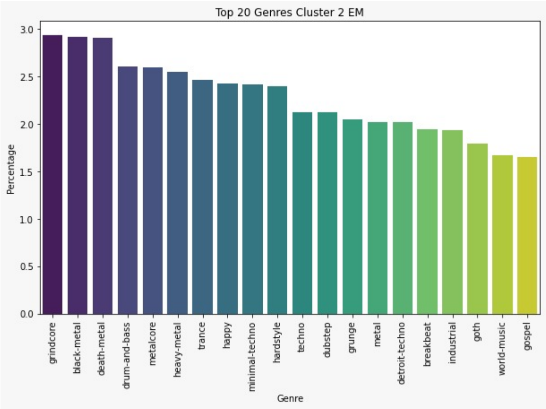

Figure 12: Cluster 2 EM

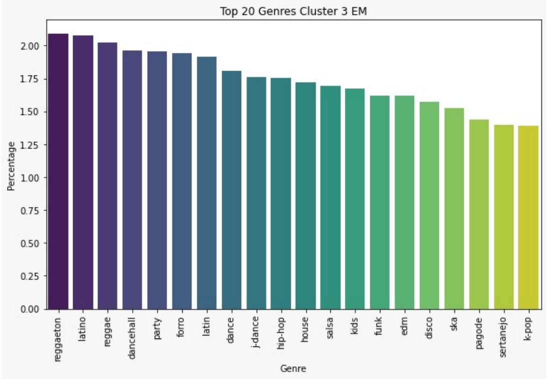

Figure 13: Cluster 3 EM

cess of developing functionalities from scratch. This approach allowed us to focus on higher-level tasks and analyses, yielding efficient data pre-processing, clustering, and results visualization.

However, we still faced some challenges during the implementation. The first obstacle was that each tool has a learning curve due to individual functionality and syntax, making it challenging to use them without a certain degree of background knowledge. More specifically, we had issues with LDA, where we assumed that it did not work because it included stop-words in the output. However, we were pleasantly surprised that the clustering was efficient, highly interpretable, and sensible to the human eye upon displaying findings using a genre-based clustering and selecting the top 20 genres. The second challenge was finding errors due to employing several libraries and tools, resulting in a more complicated code base.

In addition, despite our efforts to make the dataset more heterogeneous, as described in section 2.3, Data Preparation, the dataset still needed to be more homogeneous. We should have added many more tracks than we were able. Our efforts did not significantly impact the dataset, and we managed to alter it only by approximately 1.01%.

Nevertheless, we are glad that K-means and EM are similar as expected. Additionally, when employing LDA in future iterations, we realized the necessity to minimize the impact of stop words and musically-specific keywords like "remix." We also understood that integrating natural language processing with acoustic and numerical approaches will help achieve a more comprehensive understanding. Moreover, developing additional acoustic features and doing sequential model analysis on audio recordings might significantly improve the precision and thoroughness of our study.

9 Conclusion

While this study comparing unsupervised numerical clustering, title-based unsupervised clustering, and human-made genre classification provides some valuable insights into different clustering techniques in music genre classification, it is crucial to recognize that these techniques may oversimplify the complexity of music genres and their ability to be associated together through simple numeric and coded results. While the algorithms do put together groupings that make sense and they do often successfully identify the similarities between genres and put them together, even in the top 20 results it sometimes makes groupings that do not quite fit.

In order to provide a more complex and thorough understanding of music genre classification, it is advised that future research in this area take into account subjective factors and other pertinent features, such as cultural context and personal taste. Subjective opinions may influence the human-made genre classification used as a comparison and may not represent the larger music community's classifications. As such, the study's results may be limited in their generalizability and applicability beyond the specific comparison techniques chosen.

Furthermore, the study's focus on only three classification methods may limit the scope of the analysis, as there may be other methods or techniques that could yield more accurate or meaningful results.

In a sense, our experiment served as a test run to see what is possible with unsupervised clustering systems using them in different ways and obtain a deeper understanding of the possibilities and limitations of different clustering methods.

References

[1] Y.-S. Luo, X.-L. Zhao, T.-X. Jiang, Y.-B. Zheng, and Y. Chang, “Hyperspectral mixed noise removal via spatial-spectral constrained unsupervised deep image prior,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 9435–9449, 2021.

[2] “Spotify kernel description.” https://www.spotify.com. Accessed: 2023-03-30.

[3] L. Moore, “Audio streaming application performance: A comparative study of spotify and youtube music,” 2021.

[4] A. Assaf, R. Troncy, and A. Senart, “Roomba: An extensible framework to validate and build dataset profiles,” in *The Semantic Web: ESWC 2015 Satellite Events: ESWC 2015 Satellite Events, Portorož, Slovenia, May 31–June 4, 2015, Revised Selected Papers 12*, pp. 325–339, Springer, 2015.

[5] “Kaggle spotify tracks datasets kernel description.” www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset. Accessed: 2023-03-30.

[6] “Spotify api for developers.” https://developer.spotify.com/. Accessed: 2023-03-31.

[7] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[8] A. Vysala, D. Gomes, *et al.*, “Evaluating and validating cluster results,” *arXiv preprint arXiv:2007.08034*, 2020.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[10] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, “Towards k-means-friendly spaces: Simultaneous deep learning and clustering,” in *international conference on machine learning*, pp. 3861–3870, PMLR, 2017.

[11] J. E. Chacón Durán and A. I. Rastrojo Sagundo, “Minimum adjusted rand index for two clusterings of a given size,” 2023.

[12] A. Amelio and C. Pizzuti, “Correction for closeness: Adjusting normalized mutual information measure for clustering comparison,” *Computational Intelligence*, vol. 33, no. 3, pp. 579–601, 2017.

A Appendix

Below are descriptions for each audio feature in the dataset found on Kaggle (5).

track-id: The Spotify ID for the track.

artists: The artists' names who performed the track. If there is more than one artist, they are separated by a ';'.

album-name: The album name in which the track appears.

track-name: Name of the track.

popularity: The popularity of a track is a value between 0 and 100, with 100 being the most popular. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past. Duplicate tracks (e.g. the same track from a single and an album) are rated independently. Artist and album popularity is derived mathematically from track popularity.

duration-ms: The track length in milliseconds.

explicit: Whether or not the track has explicit lyrics (true = yes it does; false = no it does not OR unknown).

danceability: 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.

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. For example, death metal has high energy, while a Bach prelude scores low on the scale.

key: The key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. $0 = C, 1 = C/D, 2 = D$, and so on. If no key was detected, the value is -1.

loudness: The overall loudness of a track in decibels (dB) mode: Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0

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. 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

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

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

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

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)

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

time-signature: An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure). The time signature ranges from 3 to 7 indicating time signatures of 3/4, to 7/4.

track-genre: The genre in which the track belongs.