COGS 118B Final Project Paper

Code

Our code base for this project can be found at the following link.

Introduction and Motivation

This project focuses on analyzing the relationship between a song's quantitative features and its genre. The purpose of this project is to determine the extent to which a song's genre is determined by its audio characteristics. We reach our goal using a dimensionality-reduction algorithm, Principal Component Analysis (PCA), and two unsupervised clustering algorithms, K-Means Clustering and the Gaussian Mixture Model (GMM). These algorithms are performed on a dataset named *Spotify - All Time Top 2000s Mega Dataset*[1]. In this report, we outline how we processed this dataset, used aforementioned algorithms on it, as well as the conclusions drawn from this.

We were motivated to do this project after asking ourselves what distinguishes songs that belong in different genres from each other, since people's music preferences are often based on genre. Also, with the increasing role played by technology in the world of music, genres evolve over time without breaking away from the audio characteristics that they centered themselves on without technology. Therefore, we thought it would be relevant to use clustering methods in an attempt to find out whether these characteristics truly determine which genre a song belongs to. For songwriters and musicians, this information can be useful, as they would be able to know how to use audio characteristics if they want to belong to a specific genre.

Related Work

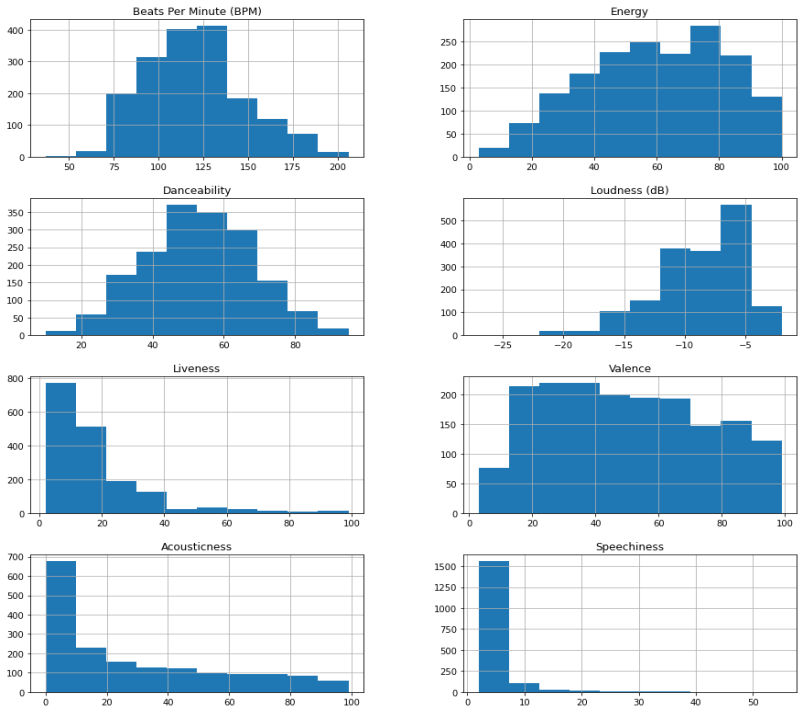
The most common use of Machine Learning in music revolves around song recommendation. While most of these algorithms were initially revolving around relational algorithms based on users, newer projects that have come to fruition focus on trends related to features associated with the actual songs. We found two articles that attempt to build content-based filtering instead of the commonly used collaborative filtering. The first one was written by Aimun Khan on towardsdatascience.com[2], and the second one was written by Kim, Kim, and Yun from Stanford University[3]. In both of these projects, we see quantitative features of songs being used to cluster them into genres, which inspired our project.

Methods

Exploring and Preparing the Data

Our dataset contained 1994 rows (songs and albums) and 14 columns (features). Upon checking for missing values, we found that they were none. For our clustering, we only kept quantitative values, removing non-audio characteristics such as year or popularity as well as length, since it was not normalized between songs and albums. We ended up with 8 features that would serve as input for our clustering algorithms: acousticness, beats per minute, danceability, energy, liveness, loudness,

Figure 1.



speechiness, and valence. The distribution of their values can be seen in *Figure 1* above.

Genre	# Entries
Rock	857
Pop	388
Adult Standards	123
Metal	93
Indie	79
Dutch Cabaret	51
Soul	45
Permanent Wave	38
British Invasion	36
Hip Hop	29

Table 1.

We then focused on the dataset column that denoted genre and found 149 different genres with Album Rock being the most popular with 413 entries. Several genres contained only 1 entry. This led to us needing to prepare our data, as 149 genres were too many for clustering. We reduced this number in two ways. First, we grouped sub-genres together. For example, Album Rock and Alternative Rock got grouped together under Rock. Second, after this step, we only kept genres with more than 20 songs to avoid sparse clusters. This removed 255 entries from the dataset, leaving us with the 10 most popular genres, which are displayed, along with the number of entries in them, in *Table 1* on the left.

Principal Component Analysis

Principal Component Analysis (PCA) focuses on dimensionality reduction. Once we reduce the dimensionality of the dataset by removing the redundant features, we are left with the dimensions (features) that are the most relevant to the genre, which is our ultimate goal. In our project, we used PCA once, and used the results from PCA in a few different ways.

The first few steps of PCA were to normalize the matrix by subtracting the mean, find the covariance matrix by multiplying the normalized matrix with its transpose, and obtain the eigenvectors and eigenvalues of the covariance matrix using the *np.eig* command. Using the normalized eigenvalues of the covariance matrix, we graphed a scree plot (*Figure 2*) to show the principal components that contributed the most to the overall variance of our matrix. We then used the eigenvectors to create a heat-map (*Figure 3*) in order to see how much each feature contributed to the new principal components. From that, we were able to determine how much each feature contributed to the overall variance by multiplying the contribution of each feature to all the new principal components by the percentage of variance that the components contribute to and adding up the results. We were then able to deduce that the three most important features that contributed to genre clustering were energy, liveness, and valence.

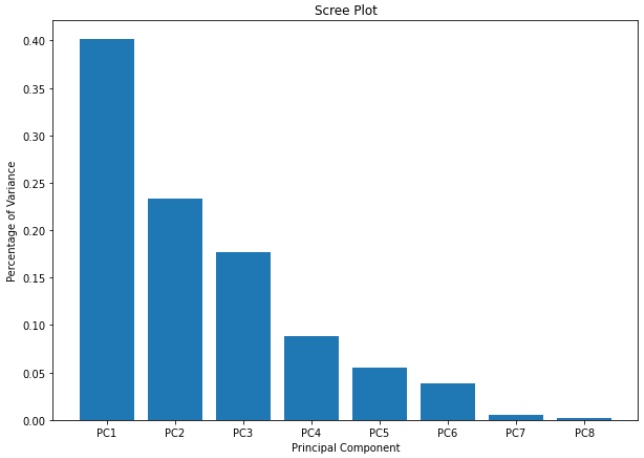


Figure 2.

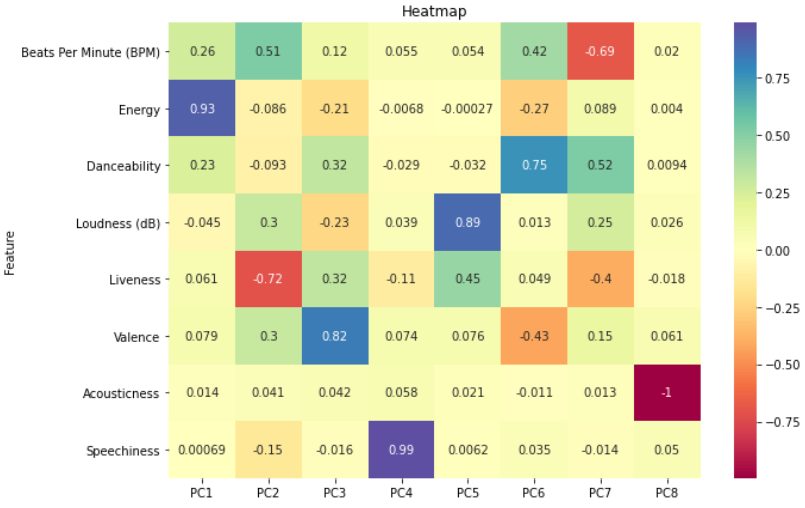


Figure 3.

K-Means Clustering

K-Means Clustering is a clustering algorithm that defines a certain amount of clusters and clusters data by iteratively moving the centers of these clusters to the mean value of the data points closest to them. In our K-Means Clustering implementation, we set the number of clusters to 10, one for each genre. For our data points, we used the dataset values dependent on the number of features being used, as we ran our algorithm twice (once with all 8 features and once with the most 3 important ones obtained from PCA).

To implement K-Means Clustering, we started by initializing random cluster centroids. Then, in a while loop, we implemented the three steps we learned about in class. The first one was to calculate the Euclidean distance between each datapoint and centroid and store them in a matrix. The second one was to assign each datapoint to a cluster by seeing which centroid each cluster was closest to. This defined our rank matrix. The third step was to move each centroid to the mean of all the data points that are assigned to the cluster it corresponds to. The while loop kept iterating until the location of the centroid converged (when all centroids collectively move a distance less than 10^{-6} from the previous iteration). We started by writing this while loop completely unvectorized and, once we knew it was working, we vectorized it using the implementation from Homework 2 for help. Once we had our final rank matrix, we iterated through it to extract the number of datapoints in each cluster as well as the data points themselves. These two values were returned to be used in analysis.

Gaussian Mixture Model

Gaussian Mixture Models (GMM's) have a plethora of different applications. In the case of this project, a GMM was used to cluster the songs into their genres. GMM's generally involve a superposition of many gaussian distributions and take advantage of Expectation-Maximization (EM), a known method to find the maximum-likelihood estimations for model parameters when the given data has latent variables, has missing data, or is incomplete. We can think of GMM Clustering as an EM algorithm for a mixture of Gaussians. There is an iterative search for the covariance, weights, and means of the unlabeled data. The two steps in an EM algorithm are (1) the Expectation step, which is used to compute the expected likelihood of the data with respect to any possible missing data, and (2) the Maximization step, which finds the maximum-likelihood vector given the observed data, as well as computes the probability density function for the unobserved data. This applies to GMM Clustering as such: (1) The Expectation step computes the weights for each Gaussian for each datapoint, and (2) the Maximization step modifies the parameters of each Gaussian in order to be able to maximize the likelihood for the given data.

Similarly to K-Means Clustering, we opted to execute GMM Clustering two times, once with all 8 features and once with the most 3 important ones obtained from PCA. Because we had already implemented K-Means Clustering and PCA from scratch, we used GMM Clustering, not as the main focus of the project, but to provide a more in-depth analysis. For this reason, we used the *GaussianMixture* class from the *sklearn.mixture* library to implement it.

Results

For the result of our clustering algorithms, we expected the number of datapoints in each cluster to somewhat match the number of songs that belong to each genre in our dataset. For any given cluster, we also expected a large portion of the data points in it to belong to a single genre, with a small number belonging to other genres, due to the potential error margin that comes with clustering.

Table 2.

Cluster #	# Entries with 8 Features	# Entries with 3 Features
1	346	221
2	238	215
3	194	210
4	190	200
5	177	196
6	158	165
7	154	146
8	113	142
9	86	142
10	83	102

For K-Means Clustering, we did not obtain our expected results using all 8 of our features nor using the most important 3 features, as defined by PCA. The number of datapoints in each cluster for both runs of our algorithm is outlined in *Table 2* above.

As can be seen, the number of datapoints in each cluster does not match the number of songs in each genre in our dataset. This tells us that our data was not classified correctly based on genre. Neither run of our algorithm performed particularly better, so we decided to use the one with 3 features for further experiment, as the features used for it are the most important ones. We decided to inspect the genre distribution in each cluster for that run of our algorithm. We found that Rock was the most dominant genre in every cluster, making up between 38% and 56% of the songs in each cluster. This further confirmed that clustering based on song genre did not occur the way we expected it to.

For GMM Clustering, the number of datapoints belonging to each cluster for our two runs of the algorithm can be observed in *Table 3* on the right.

Table 3.

Cluster #	# Entries with 8 Features	# Entries with 3 Features
1	761	335
2	410	294
3	141	291
4	125	183
5	96	146
6	92	141
7	39	114
8	30	107
9	23	67
10	22	61

As can be seen, the number of datapoints using the three most important features also did not match our expected results, telling us that songs were not clustered based on genre. However, the results using all eight features looked promising as the numbers somewhat closely matched. In order to know if they actually did, we looked at the genre distribution in each cluster another time for that run of the algorithm. From this, we saw that Rock was, once again, the most popular genre in every single cluster, making up between 40% and 70% of the songs in each cluster. If clustering based on genre was done correctly, each genre would be the most popular in exactly one cluster, telling us that the similar numbers were just a coincidence and that clustering based on genre did not happen for our GMM Clustering either.

Because clustering songs based on their genre using features related to their audio characteristics was unsuccessful with two different methods of clustering, we thought that there may be another non-quantitative feature that defines genre. Due to this, we decided to look at all the artists in our dataset and see how many genres their songs fit in. We found that our dataset contained songs from 591 artists, meaning that at least some of the artists had more than one song in the dataset, and that the number of different genres that one artist's songs fit into never exceeded 1. From this, we deduced that the artist of a song seems to determine genre more than the actual audio characteristics of the song.

Discussion

From our results, we learned that song features that relate to audio characteristics do not seem to be the factor that determines a song's genre. This could be seen from the fact that none of the clustering methods we used were accurate when attempting to cluster the genres of the songs in our dataset. Instead, the way we determine a song's genre seems to be more superficial, as our results showed that it is predominantly determined by the artist of the song. From this, we learned there may be a general bias when determining the genre of a song, as our perception of artists affects the way we categorize their songs.

We believe that our clustering algorithms did not work for two reasons. The first one can be seen when looking at the distribution of the values of each quantitative feature (*Figure 1*). We see that there are no visible categories that can be perceived in these distributions, telling us that this data may not be clusterable. The second reason is that the number of songs in each genre we used for clustering was not evenly weighted.

This led to the most popular genres, such as Rock, having a larger presence in our dataset than the least popular ones, such as Hip Hop. This means that there is a higher probability for Rock songs to have vastly different audio features than for Hip Hop songs, which impacted our clustering negatively. Due to this, one improvement that we could have made would be to prepare our data differently, so that the number of songs in each genre is comparable for all genres to be used in our clustering. Another improvement we could have made would have been to explore other important features that came out of PCA. For both of our clustering algorithms, the algorithms did not improve when ran with the features extracted from PCA, which leads us to think that it could have been worth it to explore features other than the 3 we used that were still important as determined by PCA results..

If we had more time for this project, there are several next steps that we could have taken. One of them would be to find a way to explore qualitative features in more depth, such as year of release or location of the artist (for which we would need another dataset). As shown in our results, we saw that a brief look at the relationship between artists and genre led to promising results. This is a good enough reason to believe that an algorithm clustering qualitative features of a song would yield results that would match the expected results better.

Another step we could have taken, if given more time, would have been to draw a timeline through the years of the progression of how quantitative features of a song influence genre clustering. This could have been done using the dataset we used, as the year of release of each song and album is provided. As discussed in this Complex article[4], it is becoming increasingly more difficult to determine which genre an artist fits into. Hence, it would be interesting to see if genre clustering based on audio characteristics would yield to clearer clustering with older songs than newer songs.

The two steps discussed above are two of many steps that could be taken to improve our project. This shows that, while we learned a lot from our experiments, there is plenty of room to expand on what we did in this project.

Author Contributions

- **Bryan Truong:** Coded, performed and analyzed K-Means Clustering, and coded algorithm for analyzing genre percentages, which was used in K-Means Clustering and GMM.
- **Darren Wu:** Coded, performed, and analyzed PCA, and edited video.
- **Manan Surana:** Coded, performed, and analyzed K-Means Clustering, and analyzed the relationship between artist and genre.
- **Nour Yehia:** Explored and prepared the data, prepared clustering results for analysis, helped with vectorizing K-Means Clustering, and helped with GMM results analysis.
- **Rahul Bhattacharya:** Researched datasets, helped explore and prepare the data, and oversaw the creation of the report and presentation.
- **Zakaria Gorgis:** Explored data, coded, implemented and analyzed Gaussian Model Mixtures, and worked on writing parts of report

References

1. https://www.kaggle.com/iamsumat/spotify-top-2000s-mega-dataset
2. https://towardsdatascience.com/music-to-my-ears-an-unsupervised-approach-to-user-specific-song-recommendation-6c291acc2c12
3. http://cs229.stanford.edu/proj2015/129_report.pdf
4. https://www.complex.com/pigeons-and-planes/2019/12/whats-next-in-music-everything