Sub Genre Analysis

Sricharan Reddy Varra
sricharan.varra@colorado.edu
University of Colorado
Boulder, Colorado

Sofia Lange
sofia.lange@colorado.edu
University of Colorado
Boulder, Colorado

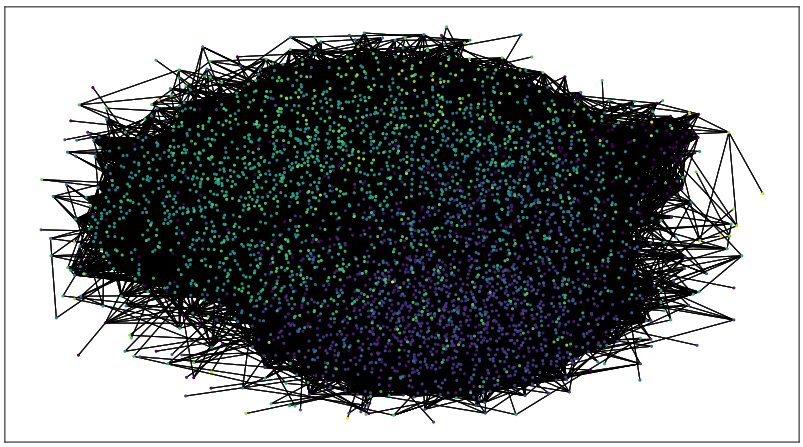

Figure 1: The Louvain Graph Community Detection Method on the *everynoise* Subgenre Network

ABSTRACT

The purpose of the project is to analyze the connectedness and features of different musical subgenres from Spotify. Genres of music can be represented and quantified in many ways. In this work we build a network based on connected genres at the website Every Noise. We compare the genres deemed as similar according to this website to genres found to be similar through K-Means clustering using audio features such as danceability, valence, and tempo. We build a network of nodes and edges based on the connection of genres on the above website and perform graph partitioning on this generated graph. We limited our graph to include only relevant nodes in order to better understand any patterns which might arise. To do this, we took genre rankings from each school in the US and combined these to get the top 75 genres according to university students across the US. We then compared the communities generated on a graph which included only these popular genres using both methods to glean insights into the genres which are most popular in U.S. Schools, and to see if we can understand better the way that musical features influence genre.

Boulder '19, December 17, 2019, Boulder, CO

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

KEYWORDS

data science, k-means, clustering, audio, music, audio features, music genres, subgenres

ACM Reference Format:
Sricharan Reddy Varra and Sofia Lange. 2019. Sub Genre Analysis. In *Proceedings of Boulder '19: CSCI-4502 Data Science Project (Boulder '19).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Contents

1 INTRODUCTION

Understanding similarities between musical genres would be very useful in building a recommendation system. If we can more generally categorize genres into larger groups, it may be easier to have a system which provides a desired amount of variability while still exposing users to music which they would find enjoyable. Making smart, relevant recommendations is of great interest to many large corporations including Apple Music, Spotify, Pandora, and more. Also, if we are able to quantify the classification and clustering of genres, we may also be able to expand this to the analysis of similar artists, which would lead to better recommendations and insights as well.

Furthermore, it is of personal interest to see which musical features are relevant to which subgenres, and which genres are similar according to these features. Perhaps through exploring and grouping these genres we can better understand some genres and what makes them unique, or what connects them to other genres.

2 DATA

2.1 Description

Our data was sourced from the website Every Noise. This website provides the names and associated genres for over 3000 subgenres which exist on Spotify. Furthermore, for each genre, it provides the associated genres which it deems most similar to the given genre. It also provides a sample playlist for each genre with examples of popular representative songs which belong to this genre. We used this playlist to generate musical features for each subgenre, by taking the average features of each song in the sample playlist. We also made use of data from a specific subsection of the website Every School at Once where the top subgenres for each university in the world are listed. To keep matters a bit more local, we narrowed the universities down to just those in the United States. This allows us to analyze data that is more relevant to our target audience for this project.

2.2 Collection Methods

We used a Python webscraper to get the html data from the website, then parsed this data and created csv files with the information which we found to be relevant to our research. This included every name of each subgenre, the genres listed as most similar to this genre, and their weights (some genres are more *similar* than to other similar genres). Also, we performed the same process to get the ranking of genres for each university in the US.

To get the musical features, we used Spotify's API. For each subgenre, we got each track in its associated playlist and took an average of the musical features for all the tracks in this playlist with the idea that this average can be a representation of the average musical features for this subgenre.

3 RELATED WORK

The exploration and understanding of music and music genres via computational methods has been tackled by many researchers. The field of Music Information Retrieval often attempts to automatically classify music based on different features. According to researchers Carlos Silla and Alex Freitas [4], there are a few different ways that this problem is usually handled, because there are five unique feature sets when exploring the musical data domain. Namely, music classification can use the following:

- **Content** - extracted from digital audio files
- **Symbolic** - MIDI formatted songs contain different features pertaining to types and frequencies of instruments
- **Lyrics** - natural language processing on the lyrics of songs
- **Community Meta-Data** - web scraping of data pertaining to songs or artists
- **Hybrid** - combination of above approaches

Music recommendation systems can be based on many features, and may use musical similarity or user similarity to determine the best recommendations. One approach which has parallels to our approach here is the one in [5], where they used k-means clustering on users in order to recommend relevant music.

It would seem, however, that there has not been much research done into the highly granular world of musical subgenres. Perhaps these are grouped and used in proprietary spaces as features in content based filtering systems, but it would appear that there is not a wide spread understanding of what makes genres similar, or even defines genres as different from each other.

3.1 Contributions of This Work

The key understanding that can be taken away from this work is a better grasp of the features that connect different musical subgenres. Another contribution is the analysis and comparison of different methods for grouping subgenres, either based on features, or based on connections from this everynoise website.

One of the unintended contributions of this work is the exploration into comparing SpringRank versus PageRank versus Bordo Counting for combining multiple ranks into a general popularity ranking.

4 EXPLORATORY RESULTS

We first attempted to create a graph based on connecting genres which were deemed as similar by the everynoise website. This created a very complex network of nodes and edges. We attempted to find the best partition of this graph to perhaps find similar groupings of genres and see if there were any interesting communities which we didn't expect. However, since this was such a convoluted and dense network, we really couldn't extract anything interesting from it. View the graph below in the teaser image on page 1.

We also decided to investigate the distributions of different musical features for the subgenres. We wanted to make sure that the features we were using were not too highly correlated, and were showing a good spread that could represent patterns in the data. Below, you can see the histograms for a few chosen features. In

Figure 2, it can be observed that the distribution of instrumentalness is extremely skewed towards very low values. Following that in figure 3 we see that liveness is similarly concentrated in the range 0.0-0.2. We found this pattern for a few of the features and decided that this may mean they would not be as useful in k-means clustering, because k-means uses Euclidean Distance and if all the data points have a value within such a small range, this would most likely mean this feature would have little to no impact on its cluster assignment.

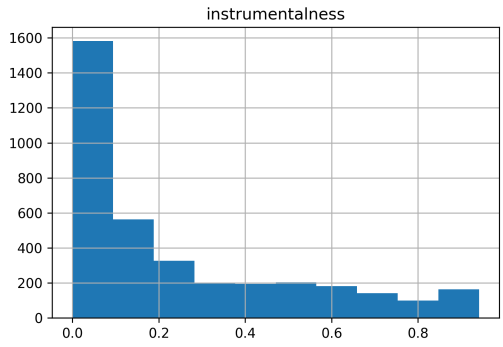

Figure 2: Genre Instrumentalness Distribution

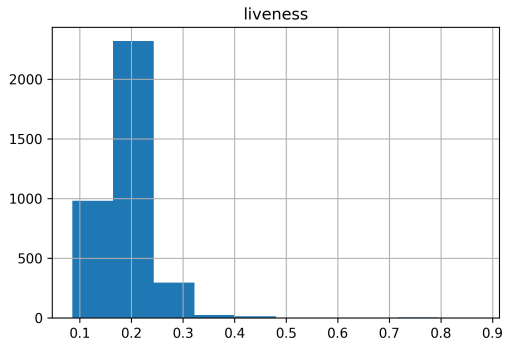

Figure 3: Genre Liveness Distribution

In the following figure (figure 4), the distribution of one of the features we found to be more meaningful for the genres, key, is shown. As you can see, it is distributed normally, which we viewed as a good indicator. A few other features are distributed in a moderately normal fashion similar to the one below.

Finally, we looked at a correlation heat map 5 for all of the features. We figured that using highly correlated features would not be ideal, so decided to try to extract which features were correlated and eliminate one or most if they were representing the same thing for the data. We also listed the most highly or negatively correlated features in Table 1.

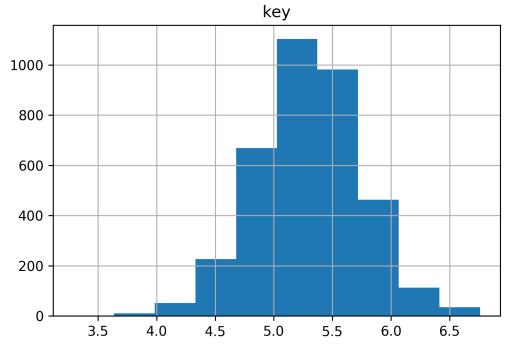

Figure 4: Genre Key Distribution

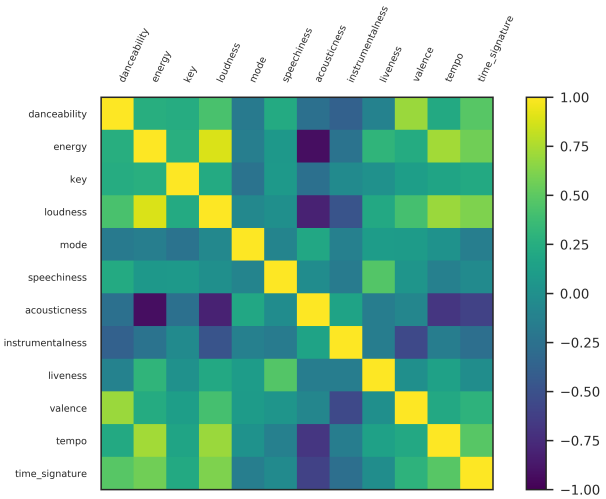

Figure 5: Feature Correlation Heat Map

Table 1: Features with largest correlation

Feature 1	Feature 2	Correlation
Danceability	Valence	0.70
Energy	Tempo	0.73
Energy	Loudness	0.88
Energy	Acousticness	-0.90
Loudness	Accousticness	-0.81

4.1 Feature Selection/Dimensionality Reduction

Using the information gained from looking at the distributions and correlations of features in conjunction to Principal Component Analysis, we decided to limit our dimensions to the following:

(1) Energy
(2) Average of Danceability and Valence
(3) Mode
(4) Key

(5) Tempo

5 METHODOLOGY: DESIGN AND TECHNIQUES

5.1 Principal Component Analysis

PCA reduces the feature vectors for each subgenre $g_i \in \mathbb{R}^{12}$ into 12 eigenvectors. These eigenvectors indicate the principle 'directions' of the data. Each feature was standardized using Z-score normalization.

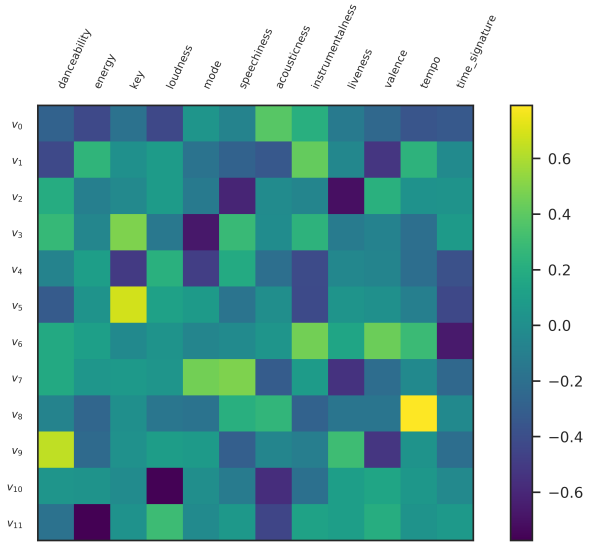

Figure 6: Eigenvector Heatmap

We can get a sense of the impact an eigenvector has with respect to the 12 features. Figure 6 displays this notion. From here we can see that:

- v_5 represents *key* positively
- v_8 represents *tempo* positively
- v_9 represents *danceability* positively

Plotting v_0 against v_1 gives what looks like a single cluster, with only a few subgenres that are scattered far from the central cluster. Many of the other eigenvector projections behave in the same way. Figure 12 shows 10 combinations of various eigenvector projections on the data along with an isolation forest implementation for outliers with respect to different features.

5.2 SpringRank

In order to combine the rankings of 637 different U.S. universities, we explored a few different options. Our first approach was to apply SpringRank Hamiltonian [2] which forms a singular ranking using an input of multiple separate rankings. We consider an edge $i \longrightarrow j$ to signify that i is ranked above j, and A_{ij} is the frequency of this instance. In this context, and edge $i \longrightarrow j$ indicates that subgenre i is ranked higher than subgenre j with The way this works is by structuring a given ranking system with a directed graph and formulating this as a physical 'spring' system trying to minimize

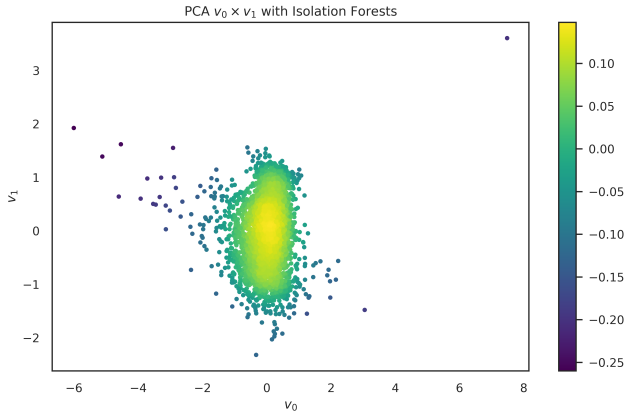

Figure 7: Projection of Eigenvectors $\mathbf{v_0}$ and $\mathbf{v_1}$

the energy of the system 1. There is a desired ordering s_i, s_j that works to minimize this energy and find the optimum ordering s^*.

$$H_{ij} = \frac{1}{2}(s_i - s_j - 1)^2 \tag{1}$$

$$[D^{out} + D^{in} - (A + A^T)]\mathbf{s}^* = [D^{out} - D^{in}]\mathbf{1} \tag{2}$$

Because this is is a sparse convex system, it's relatively easy to find the optimum order of the genres by solving for $\nabla H = 0$ as is shown in equation 2.

However, when we applied SpringRank, we found that nodes were being ranked the highest when they only occurred in 2 out of 637 school rankings, for the genre 'pinoy indie'. By our definition of popularity, this did not make intuitive sense. We wanted to find the genres which would be the most relevant to our target audience of university students, which we thought could be represented by the highest ranked genres across all universities. Since SpringRank wasn't giving us the results we wanted, we had to reassess how we were looking to combine these rankings. We applied two methods.

5.3 PageRank

First, we performed PageRank [3] on a graph similar to the one generated for SpringRank, but with the direction of edges reversed. So, if a genre is at the top of a ranking, it would have in links from each genre below it. Therefore, if a node is frequently at the top of the rankings, it would have a large number of in-links. We thought this would make sense because page rank is based on having a high degree of in-links, so a higher ranking frequently across schools would be associated with a higher page rank.

5.4 Modified Borda Count

Borda Count is a voting mechanism which seeks to combine multiple different rankings to find the overall highest ranked candidate. Basically, for each genre present in the top 10 ranked genres of each school, we counted the number of times it occurred as rank 1, rank 2, etc. Then, we did a weighted sum of these counts, giving higher weight to higher ranking. Interestingly, this method was very comparable to PageRank. When taking the top 75 ranked genres, the

number of genres common to both ranking systems was 56. We did our continued analysis on the genres with the top scores.

5.5 K-Means Clustering

Once we had the set of most popular and relevant genres, we performed k-means on the chosen features for the playlists corresponding to these genres. We chose the number of clusters to be 7 as the graphical Louvain Method produced 7 groups. However, we did also look at the associated elbow plot for clusters in the range from 1 to 10, and this is shown below in Figure 8 below.

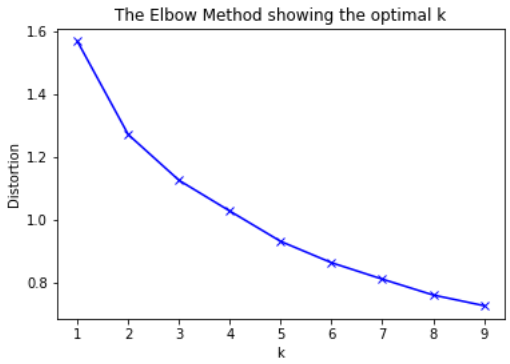

Figure 8: K-Means Elbow Plot

5.6 Louvain Modularity Maximization

Louvain Modularity Maximization [1] is a modified version of Modularity Maximization which extracts community structure in a way that is more scalable for larger sized networks. It solves the issue of 'supercommunities', by having higher counts of smaller communities. In addition it is fast, running in approximatly $O(n \log n)$. Using the top 75 genres generated from the Modified Borda Count method, a network was constructed such that there is an edge from u to v if they were in one or the other's similar genre list and Figure 13 shows this community structure between subgenres within the top 75 U.S. Universities.

6 RESULTS

The clusters found using Louvain Modularity Maximization are shown in Figure 12. The groupings in this method seem to make more intuitive sense. These groupings seem to possibly imply that these genres may be connected based on name. This method was able to differentiate all religious style music, grouped all country style music together, all of the electronic/dubstep is correctly associated, and the emo music is justifiably isolated.

The clusters found using K-Means are listed in Table 3 in the appendix. A few interesting groupings which were identified using K-Means include:

- regional mexican pop, emo, contemporary country
- underground hip-hop, indiefolk, roots worship

In Figures 9 and 10, we have plotted each of the features used for clustering for these combinations, with the addition of loudness, instrumentalness, and accousticness. You can see from these charts that these odd combinations of genres do have a few surprising similarities in terms of average musical features. However, these figures do seem to indicate that perhaps the loudness factor may be playing a role at differentiating these genres, and including it as a feature may make our results more intuitive.

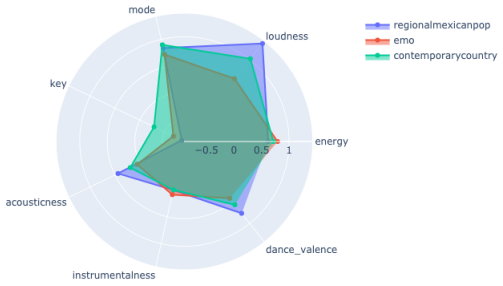

Figure 9: Oddly Similar Genres

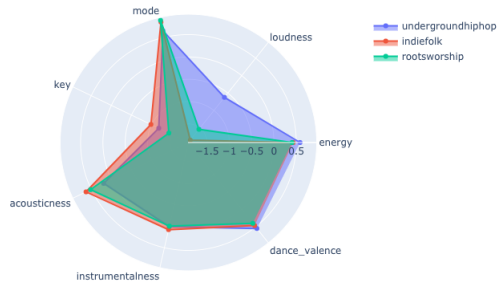

Figure 10: Oddly Similar Genres

The next figure included, Figure 11, shows the features of genres which were clustered together by Louvain Modularity Maximization but were in separate clusters according to K-means. Namely, the following genres:

- edm, popedm, electropop

Interestingly enough, these genres are very very similar in almost all features but because of the differentiation in key, they are clustered separately.

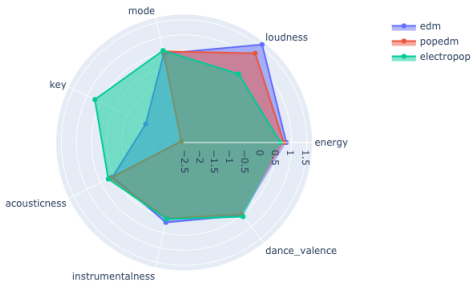

Figure 11: Oddly Similar Genres

7 CONCLUSION

7.1 Key Results

The clusters identified using Louvain Modularity Maximization were clusters which would appear to make more intuitive sense to a human. The subgenres clustered together using this method tended to appear to actually represent a higher overarching genre, or parent genre.

However, the clusters identified using K-means are quite interesting. They show that there may be some unidentified musical similarities between different genres, and this could be useful when developing targeted recommendations.

7.2 Further Work

A few different directions could be explored. Upon completion of this project in particular, it would be interesting to further delve into a recommendation system built upon similar subgenres of music, or to build one which includes this.

Also, it would be of interest to perform K-means with different sets of features included and different distance metrics and see how this influences the outcome of the clusters.

Another simple path which could be explored would be to use the ranking of different schools in the US to build a recommendation system, assigning rank as a rating. Then, you could be recommended genres of music based on the state your school is in, the type of school you attend, etc.

Another venture that would be interesting to see how the popularity of genres varies in general throughout the world, and even throughout sub regions of the US. If we look at more metropolitan areas versus more rural, if we look at smaller liberal arts colleges versus state schools, or even comparing schools with higher versus lower tuition, can we learn anything novel?

Finally, it may be of interest to potential students to be matched with schools which match their preference of music. I know I would have factored this in had I had the opportunity. An interesting app idea or Spotify integration could tell you which school best matches your listening patterns.

8 APPENDIX

8.1 Misc Plots and Tables

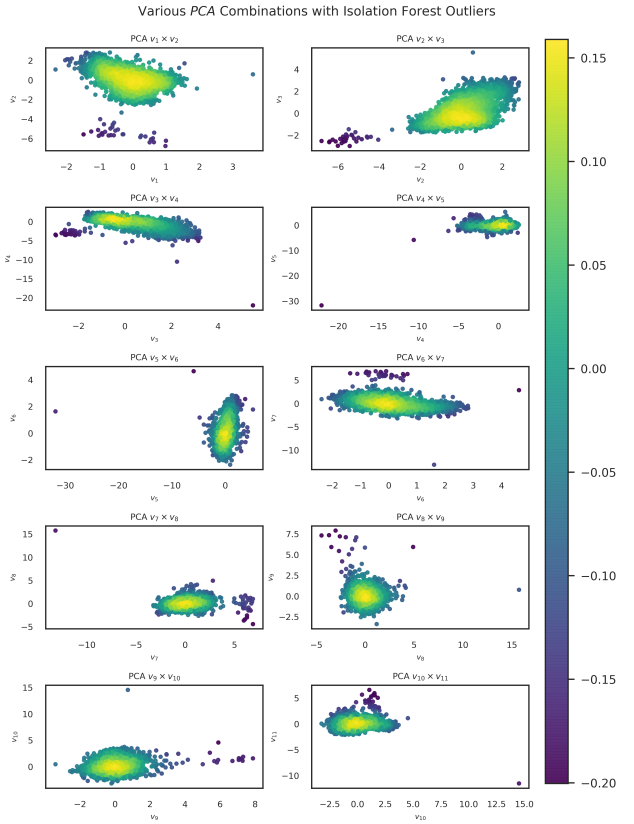

Figure 12: PCA of Various Z-Score Centered Features with Isolation Forest Outlier Detection

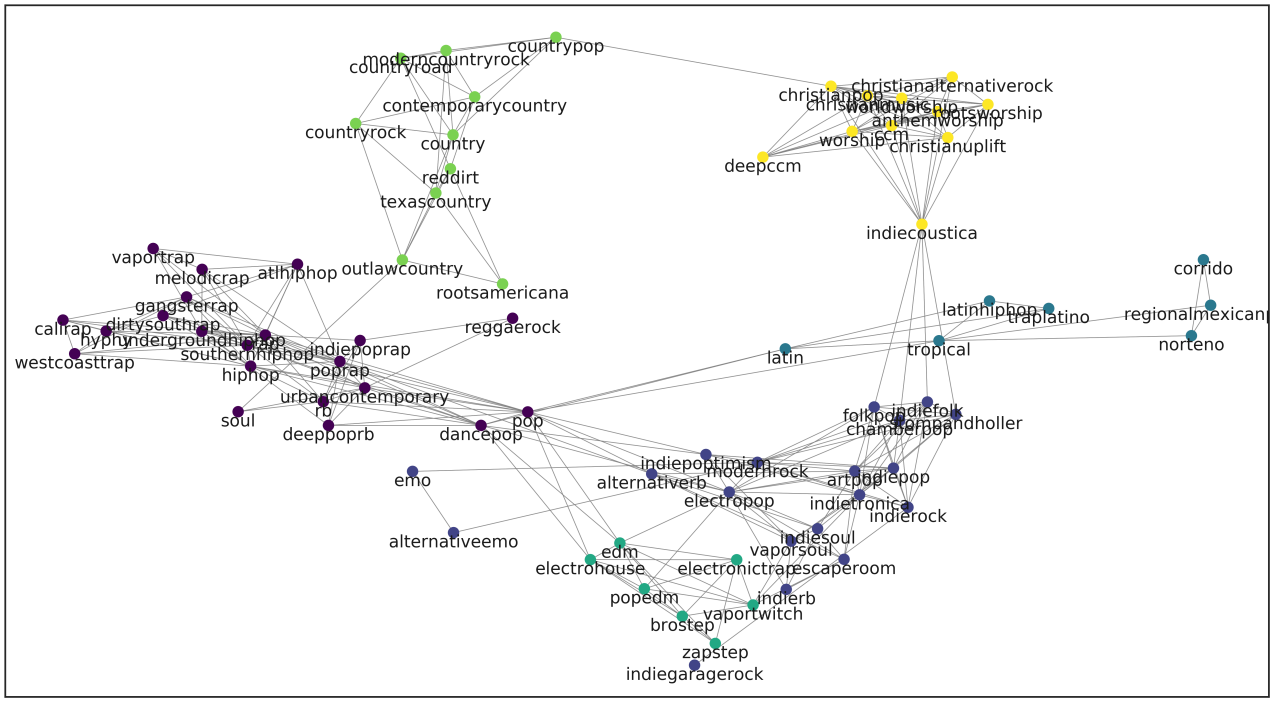

Figure 13: Clusters of Subgenres via Louvain Modularity Maximization

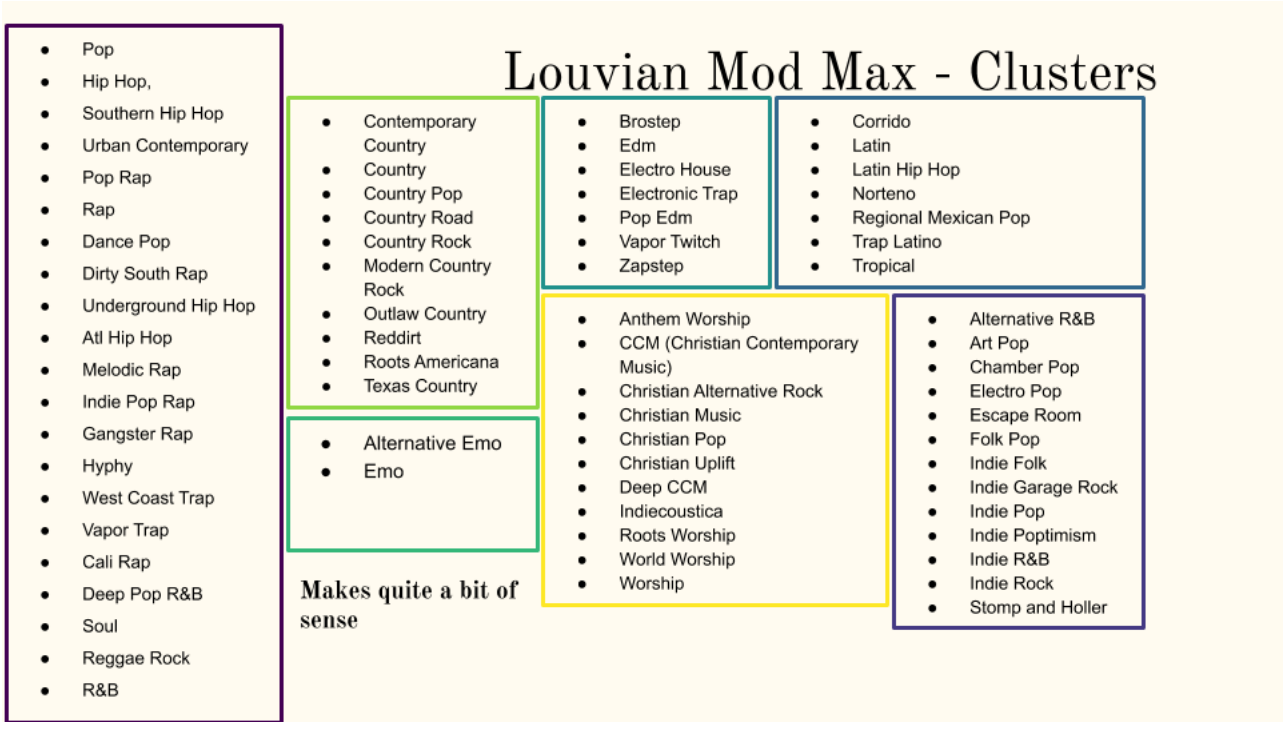

Figure 14: Louvain Clustered Genres

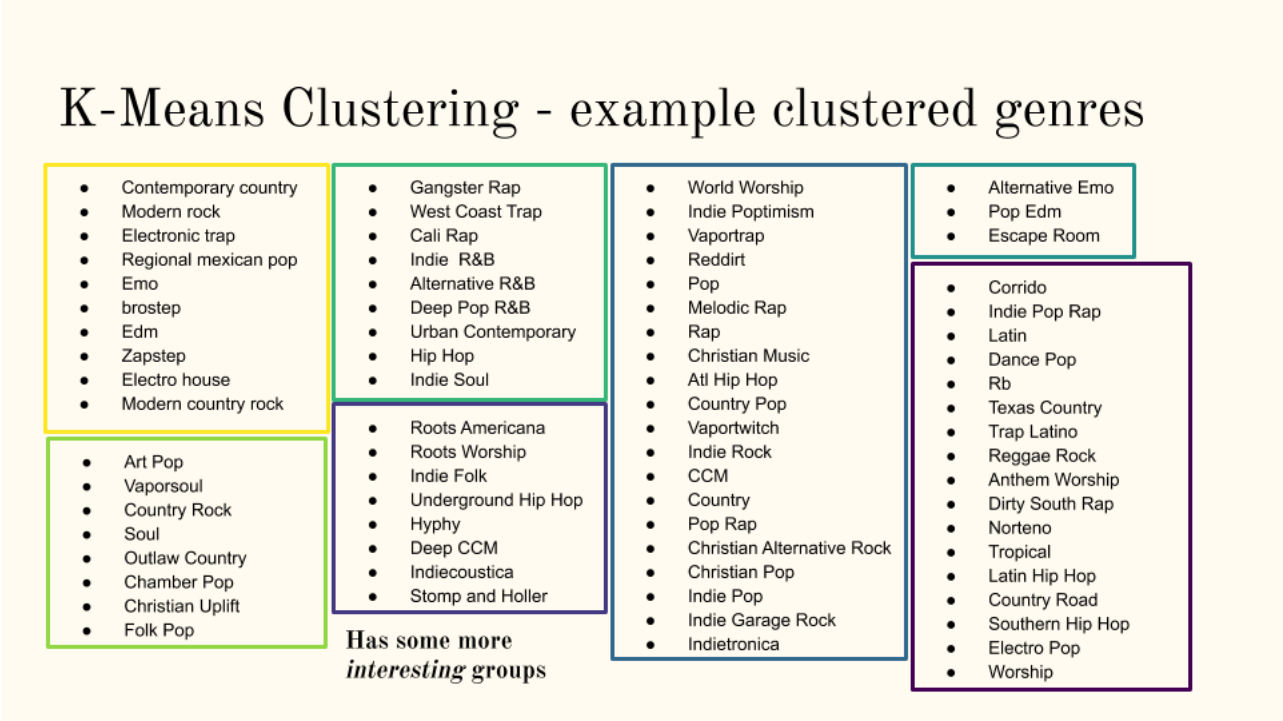

Figure 15: K-Means Clustered Genres

8.2 Honor Code Pledge

"On my honor, as a University of Colorado Boulder student, I have neither given nor received unauthorized assistance."

8.3 Github Repository Link

The GitHub Repository

REFERENCES

[1] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (Sep 2008). https://doi.org/10.1088/1742-5468/2008/10/p10008

[2] Caterina De Bacco, Daniel B. Larremore, and Cristopher Moore. 2018. A physical model for efficient ranking in networks. *Science Advances* 4, 7 (2018). https://doi.org/10.1126/sciadv.aar8260 arXiv:https://advances.sciencemag.org/content/4/7/eaar8260.full.pdf

[3] L. Page, S. Brin, R. Motwani, and T. Winograd. 1998. The PageRank citation ranking: Bringing order to the Web. In *Proceedings of the 7th International World Wide Web Conference*. Brisbane, Australia, 161–172. citeseer.nj.nec.com/page98pagerank.html

[4] C. N. Silla and A. A. Freitas. 2009. Novel top-down approaches for hierarchical classification and their application to automatic music genre classification. In *2009 IEEE International Conference on Systems, Man and Cybernetics*. 3499–3504. https://doi.org/10.1109/ICSMC.2009.5346776

[5] Gurpreet Singh and Rajdavinder Singh Boparai. 2016. A Novel Hybrid Music Recommendation System using K-Means Clustering and PLSA. *Indian Journal of Science and Technology* 9, 28 (2016). http://www.indjst.org/index.php/indjst/article/view/95592