CSCI 5523
Group 10

Final Report: Spotify Playlist Analysis

Our group's proposal went through many iterations. Our original proposal intended to predict a titanic passenger's survival based on ticket and demographic information. We were aiming to create a classifier that identifies the likelihood of survival for Titanic ship passengers informed by ticket level indicative of location on a cruise ship using this Titanic Dataset from Kaggle. This interested us at first because the idea of working with data from the Titanic was information all group members were excited to work with.

The motivation of this original idea was hoping to improve the safety of cruise ships and large aquatic vessels with internal compartments. Through the years, ships and cruises have had numerous improvements to help aid in dangerous possibilities of sinking and emergency evacuations. We thought we could research commonalities with surviving off sinking ships to help improve the safety of boarding these vessels (statista.com). We quickly realized that there was not enough information on this material so we changed our approach.

We then moved to research music specifically involving the popular app, Spotify. There was a good amount of data for this idea and it was something we were all familiar with in using. We thought we could use the general idea of our previous idea and alter it to fit this new material. We wanted to play specific genres in our playlists, but didn't know the genres of songs. So, we grouped by genre per playlist and had thought, what is a genre anyway? We came to the definition that it is based on vibes and is something that helps people categorize the songs that they like, self identify, and is a point of comparison for songs and musicians. There was no classification based on genre, so we are making them.

The motivation of involving this new data was to have common genres which contained similar and precise characteristics. Genres can be useful to both people and music as it gives more information about what someone likes and helps group together songs. This is a point of comparison between different songs and artists to help identify people's tastes in music. Basically, we wanted to see if songs with similar attributes were similar to existing genres, is genre quantifiable?

The main change is the data set but we are able to continue with our plan of selecting features, scaling, separation, and discretization/binarization. Overall, the main challenges were that the column types had different formats so this made it hard to understand and compare. This caused issues when trying to analyze the artists that sang the song and the year they were produced in. The different variables were also hard to compare as they weren't uniform. There was a lot of data preprocessing that needed to be done so that we were able to compare them. Next, we do not know every single song that exists so our data could have had errors in it. To overcome this, we had to go through each song and listen to it to prove that the data was correct. This became pretty tedious and time consuming but helped us form the data correctly. Also, artists tend to switch genres a lot throughout their careers, this can make it difficult to scale what their goals were for the songs. Many artists have hidden meanings in their songs and this can have a heavy impact on what genre they relate to. Finally, most of the songs in the data were pop songs which made it hard to categorize them due to the skew.

The data provides us with information that we can use to group similar songs together to create a "Playlist made just for you" feature. We were hoping to use the training data and compare it to the testing data to further our results. The new data set was found on Kaggle, Spotify Top Songs and Audio Features but it was hard at first to understand what some of the dataset terms meant and how they would be used in our analysis, terms like danceability, speechiness, or acousticness. From our understanding, these terms were given a value on a scale of 0 to 1 where 0 is very poor and a score of 1 is very good and the scaling was created from Spotify API, a website designed to allow data mining interactions between Spotify music and audio features.

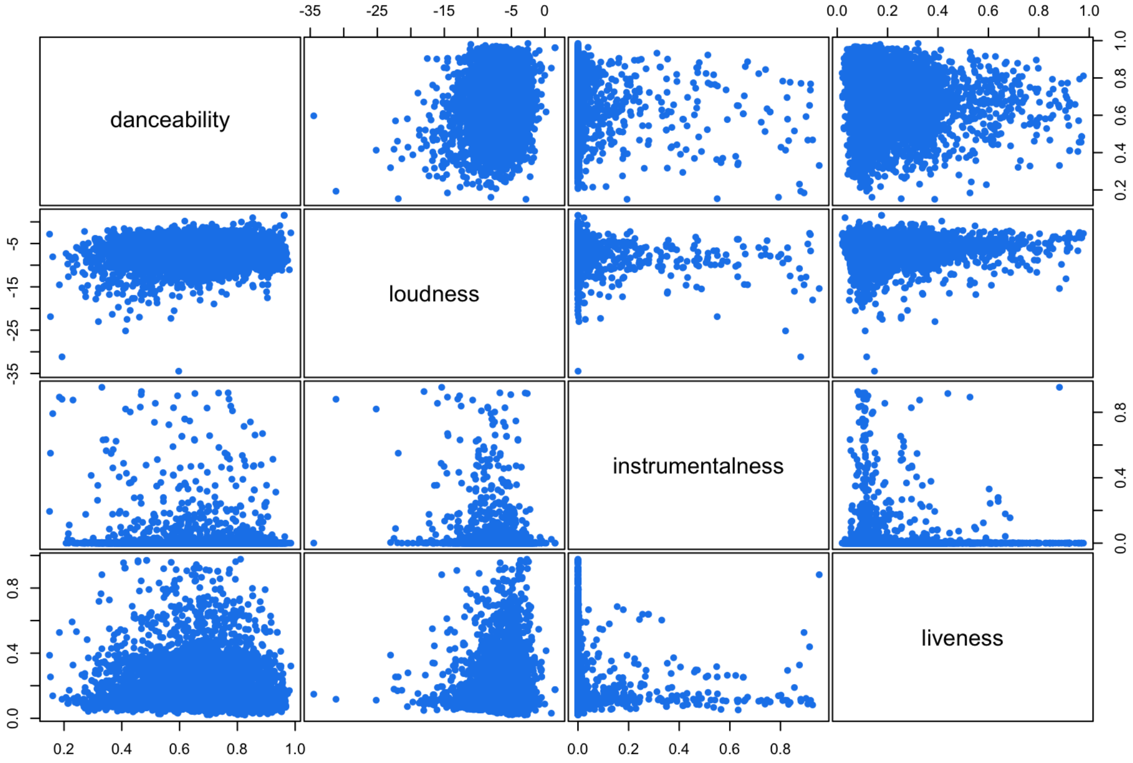


Figure 1: Pairwise Plot of song features determining the energy portrayed through the songs.

This data has a lot of information that we are able to apply to our project. There were no "NA" or missing values in our dataset so no adjustments were needed for that. However, there were multiple songs repeated more than once that we needed to remove to avoid songs showing up more than once in a playlist. We also spent time listening to some songs we didn't know to make sure that when they were clustered with other genres that it was reasonable for them to be there. There were quite a few holiday songs that were clustered in different genres which was interesting to note. This emphasized that the clusters were based on the feature statistics and gave further intuition for us to understand how the clusters were being formed.

In our exploratory data analysis, we found categories of note including duration in milliseconds, danceability, energy, liveness, the presence of explicit content, tempo, and valence. The release year of songs ranged from 1998 to 2020. Descriptive statistics included, on average, songs that had an average tempo of 120.12 beats per minute (minimum of 60.2, maximum of 210.85, standard deviation of 26.97), a loudness of -5.51 decibels (minimum of -20.51, maximum of -0.28, standard deviation of 1.93), a duration of 228748.12 milliseconds (minimum of 113000.00, maximum of 484146.00, standard deviation of 39136.57), which is about 3 minutes and 49 seconds, and 14 distinct genres. The most abundant genre was pop with 1632 songs, followed by hip hop with 778 songs and R&B with 452 songs. The least common genres were classical (1 song), jazz (2 songs), blues (4 songs), easy listening (7 songs), and world/traditional (10 songs). Additionally, we found the number of tracks associated with each artist and record label. These features will help us continue through this project and learn more about the data found.

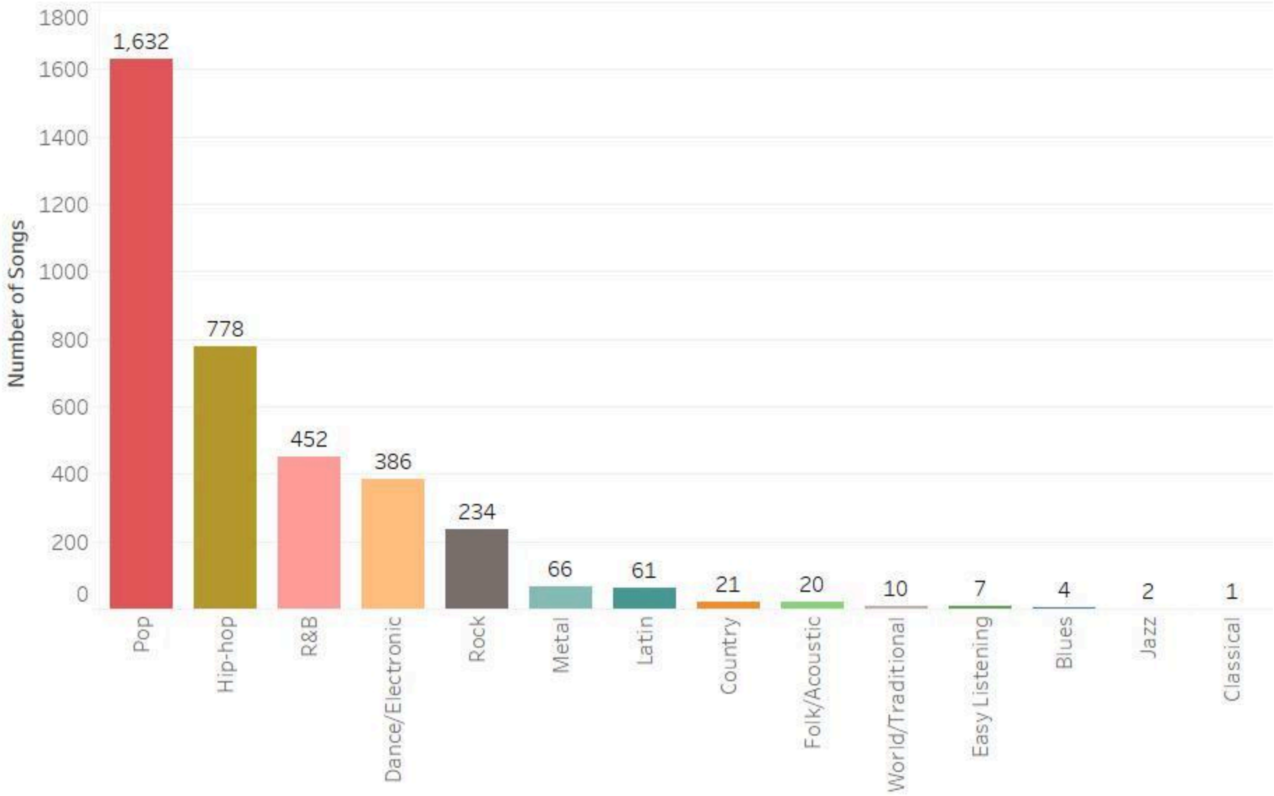
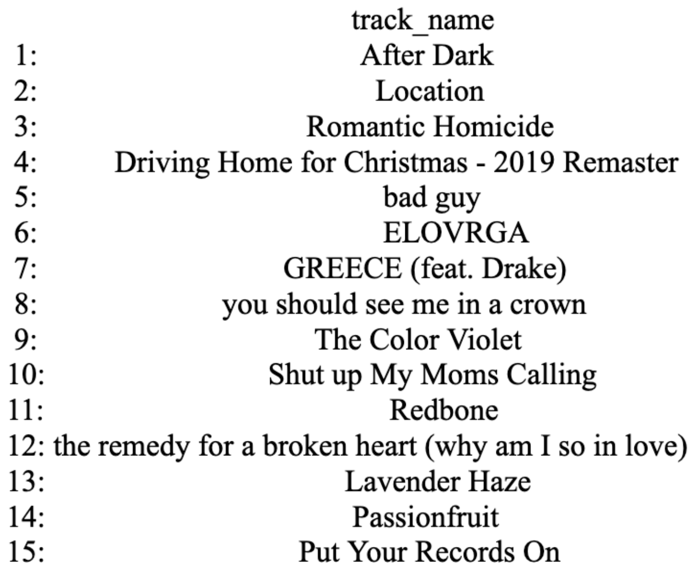


Figure 2: Base truth distribution of genres with number of songs per genre on the y-axis and genre on the x-axis.

With this data, we had initially attempted to use K-means and we found pretty much all that we expected. There were similar numbers across the various features within clusters but it was a struggle to actually evaluate the clusters as independent playlists. It was important that we filtered the songs in this list to ones that we actually knew. We had planned to further filter the data so that we can evaluate it better. There were some challenges with this data set involving the songs because obviously we don't know every song that exists. It was tedious but important for us to research the songs and listen to them so as to prove that the information was correct. After using K-means, we explored some other algorithms such as hierarchical complete clustering and hierarchical single clustering.

Using data from Spotify works well for this project, as there is the feature that predicts playlists that are made up of new songs similar to what the user listens to. Spotify is able to learn from frequently listened to songs and find songs that share similar characteristics. It then forms them into a custom playlist for the user so that they can listen to new music. We wanted to try and replicate this Spotify feature but using different feature selections to create playlists based on genre. It would be interesting to see how Spotifies algorithm works and what it uses as its group truth to make playlists. It would be especially interesting to compare and contrast their algorithm with the ones we've created mentioned further in this paper. However, their playlists are more likely to be larger than the ones we've created here which may hinder the amount of comparing we could do between the two.

For example, this first cluster generated by K-means with 100 centers is an example of the type of playlist generated by our algorithm (Figure 3).



```
                    track_name
 1:                  After Dark
 2:                    Location
 3:           Romantic Homicide
 4:   Driving Home for Christmas - 2019 Remaster
 5:                     bad guy
 6:                     ELOVRGA
 7:          GREECE (feat. Drake)
 8:      you should see me in a crown
 9:             The Color Violet
10:          Shut up My Moms Calling
11:                     Redbone
12: the remedy for a broken heart (why am I so in love)
13:               Lavender Haze
14:                Passionfruit
15:          Put Your Records On
```

Figure 3: An example playlist created from K-means clustering.

The tracks seem to relate but using multiple types of clustering has helped provide us with a better idea of which clustering technique gives us the best playlist based on its cohesiveness. Also, as mentioned before, making large playlists grouped by genre would take an excessive amount of time to listen to each song, so smaller playlists like this were easier to validate than listening to a large playlist.

The next steps after using different algorithms were to do more data preprocessing with the different categories and data found in the set. We have used K-means with centroids for k equaling 5, 10, and 15 and clustered by genre, which can be seen in Figure 4. We show that for K-means with fewer amounts of centroids (Figure 4A & B), the clusters do not form for specific genres, but rather have a proportion of each genre in its cluster. We still show this for K-means with 15 centroids and that clusters are starting to separate more based on specific genres (Figure 4C). With more time, we could look into how the algorithm would perform by using a large number of centroids, and if that makes a difference. Some issues may arise that with a large number of centroids, the playlists would be composed of too few songs, moving away from our goal of quantifying what a genre is. K-means was useful for this data and gave us more intuition of the data with more clusters.

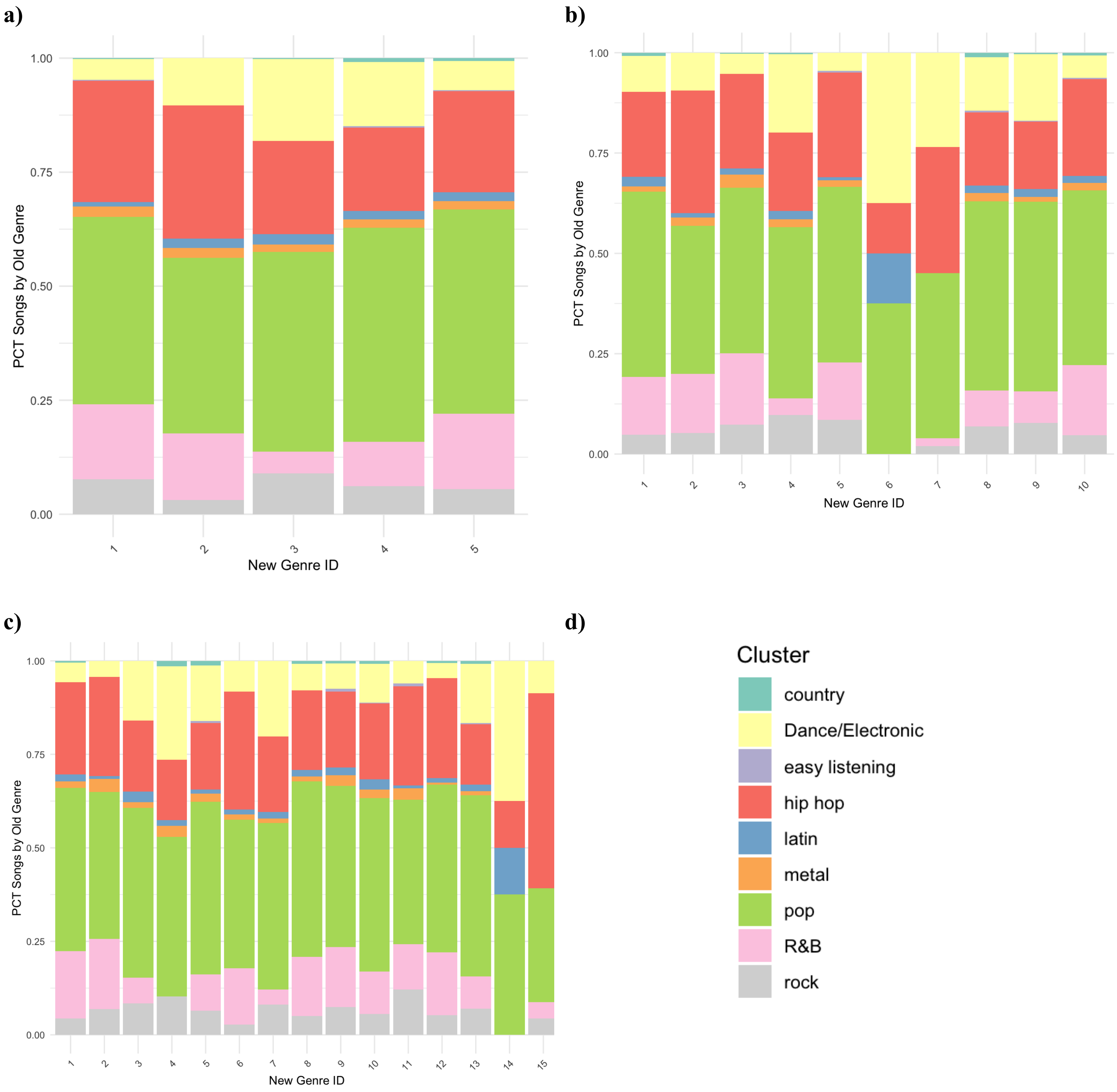


Figure 4: K-means clusters for a) 5 centroids, b) 10 centroids, and c) 15 centroids. d) Cluster color corresponds to specific genres.

Another algorithm we used to cluster out data was hierarchical complete clustering, seen in Figure 5. Similar to K-means we used three different clusters to cluster the data (5, 10, and 15). Also, with only 5 clusters we see proportions of genres similar to K-means, where there are no distinct genres composing just one cluster. However, with a greater number of clusters we start to see more groupings with fewer number of genres, indicating a better separation into their specific genres/clusters. We also start to develop intuition of which genres the algorithm deems more alike, those seen in Figure 5C like hip hop and dance/electronic in Genre ID 10.

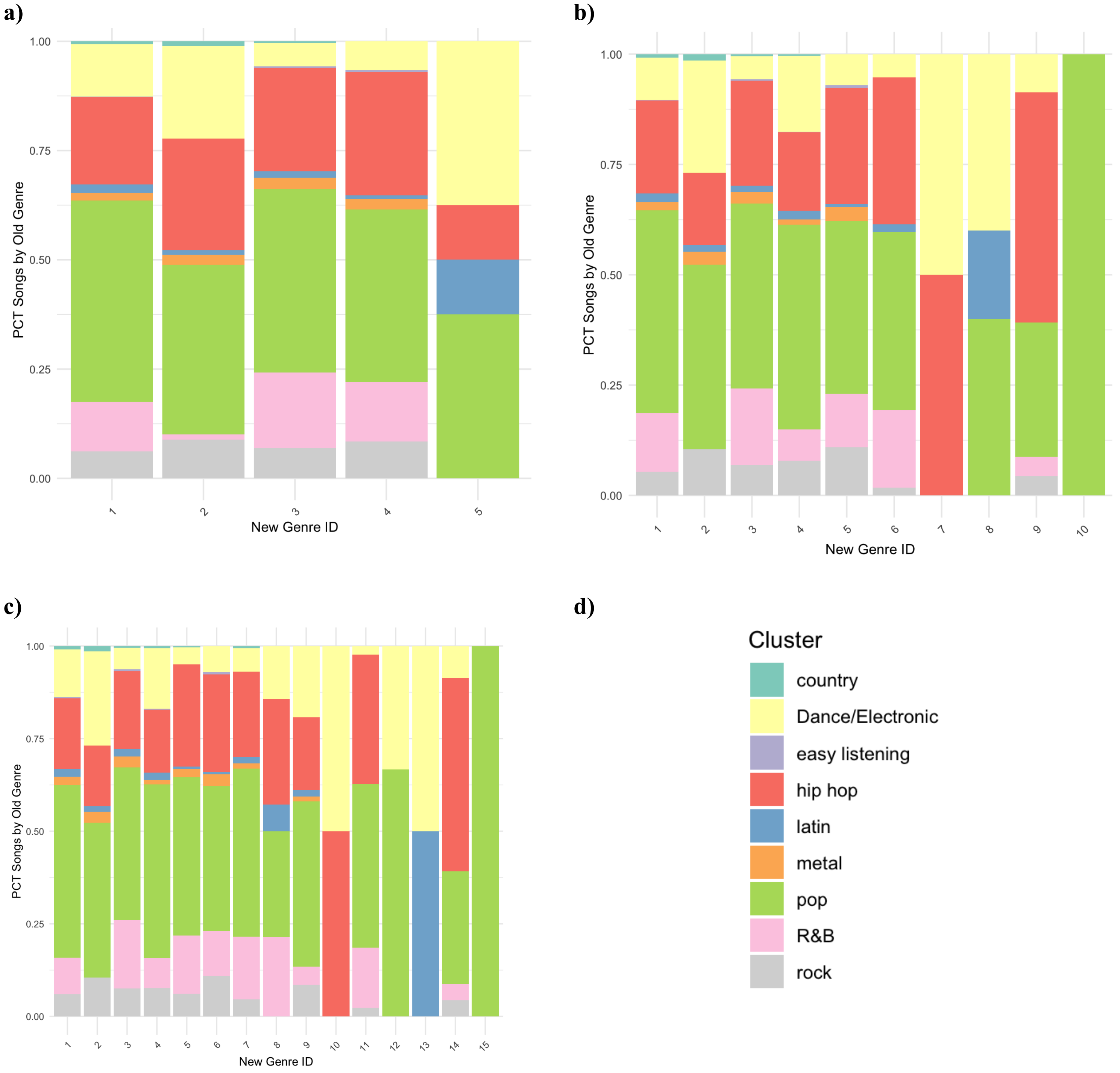


Figure 5: Hierarchical complete clustering for a) 5 centroids, b) 10 centroids, and c) 15 centroids. d) Cluster color corresponding to specific clusters.

Lastly, we used the hierarchical single clustering to cluster the Spotify data, seen in Figure 6. Even with a smaller amount of clusters, we already see distinct groups of genres. The distinction becomes even greater with larger amounts of clusters, like in Figure 6C, we have 8 clusters with only one genre, suggesting that if we increase the amount of clusters, our data could eventually have one genre per cluster, leading to distinct genre playlists.

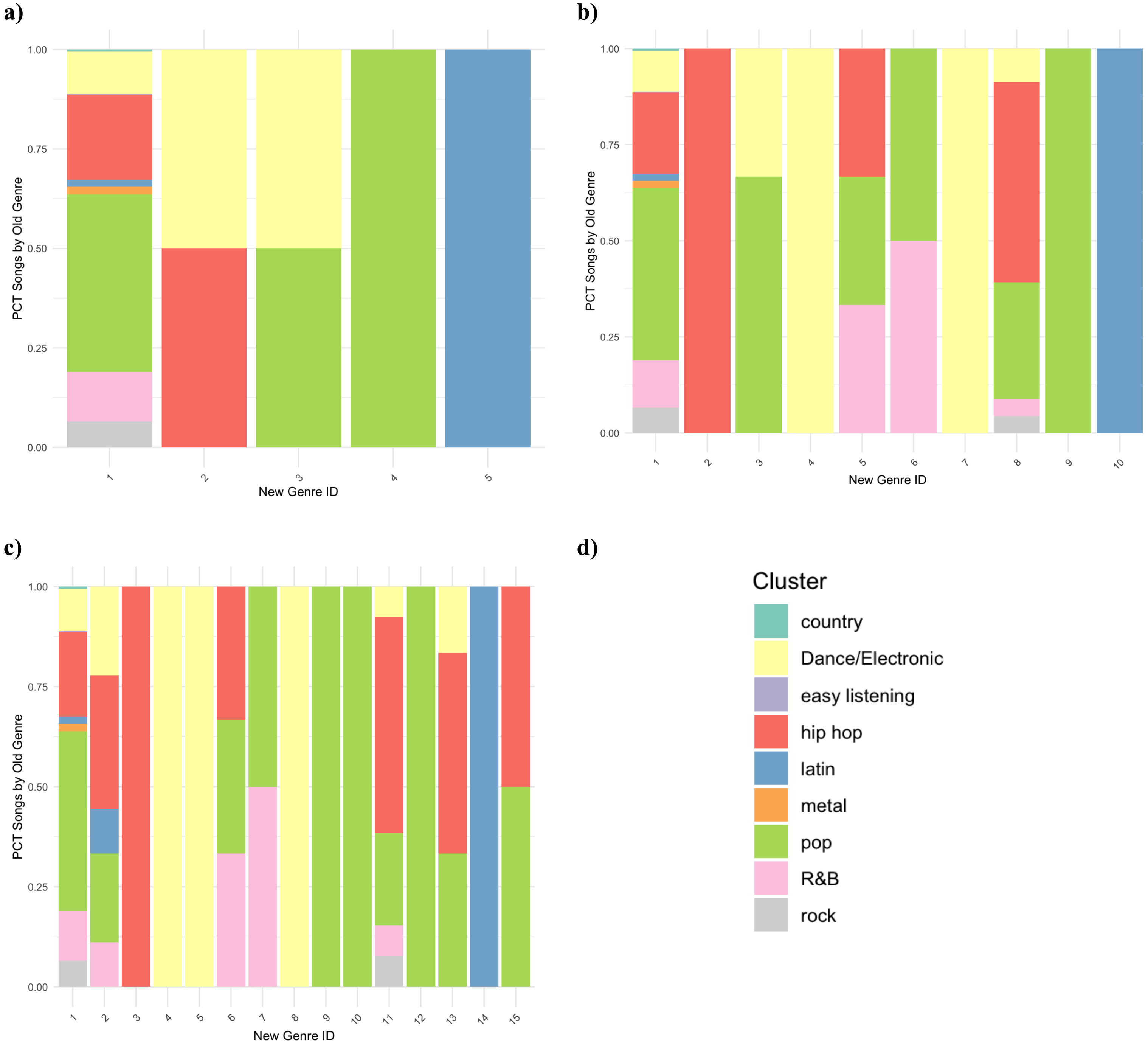


Figure 6: Hierarchical single clustering for a) 5 centroids, b) 10 centroids, and c) 15 centroids. d) Cluster color corresponding to specific clusters.

Beyond these visual assessments, we looked for a more objective "ranking" of genre systems. Figure 7 displays the intra- and inter-cluster sums of squares of: marked genres, K-means genres (K = 5), hierarchical single-linkage genres (5 clusters), and hierarchical complete-linkage genres (5 clusters). Even though the original genres had 9 more groups, it was substantially out-performed by all of the other methods and the other methods were more effective in capturing underlying patterns and relationships within the data, resulting in more cohesive genre groupings.

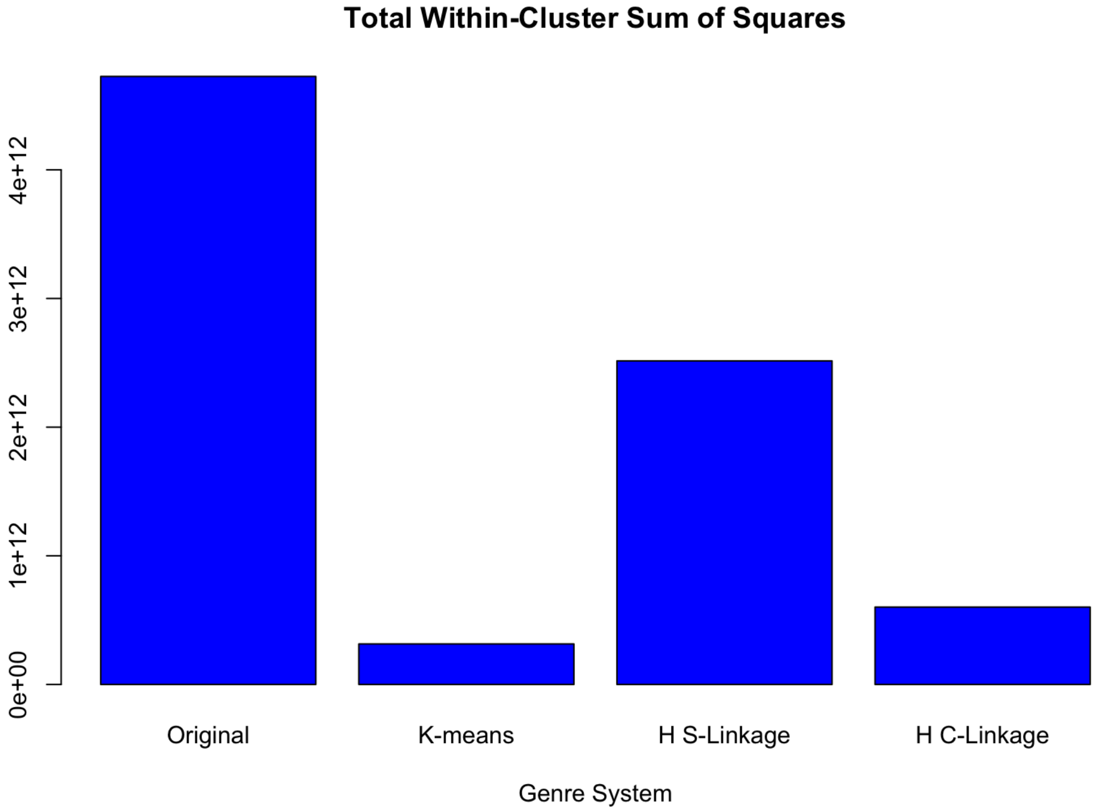


Figure 7: Bar plot of genre systems by total within-cluster sum of squares

In conclusion, we found that assigned/popular genres do a relatively poor job of grouping songs into categories. K-means, hierarchical clustering with single linkage, and hierarchical clustering with complete linkage all created groups of songs with more cohesion among the measured properties. Certain of these groupings correlated with the assigned genres (primarily Latin music). However, the assigned genres were significantly more spread out. This is likely due to various factors including artist consistency (a whole album or discography being put in the same genre) and era (how different pop is now from in the early 2000's).

Limitations of our work primarily revolve around the exact features given in the dataset. There may be other quantifiable aspects of songs that lead to more popular genre breakdowns.

Further research could identify these variables and test how the original genres hold up. Additionally, listening to the songs to see if the clusters made/playlists were of similar style and belonged where they were placed by the algorithm was time consuming. In a future study, it would be beneficial to find another way to validate our playlists that could take less time but still be a viable way to see how the algorithms were performing.

References:

Titanic Dataset
Large Vessel Statistics
Spotify Top Songs and Audio Features
Introduction to Data Mining, Second Edition*,* P.N. Tan, M. Steinbach, A. Karpatne, V. Kumar.