Honors College Thesis
Submitted in partial fulfillment of the requirements for
graduation from the Honors College

How can we use techniques from data analytics to study data on music listener behavior, produce visualizations of trends, and build predictions on listener preferences?

Matthew Van Praagh
Dr. Anil Venkatesh, Adviser
Dr. Juan Jaramillo, Reader
Dr. Sukun Li, Reader
May 19, 2023

Abstract

Recommendation systems to improve user's listening experience have become increasingly common, and improvements are constantly being made to improve the accuracy of the algorithms that display music it thinks a particular user will enjoy. Most current musical recommendation systems employ a combination of factors for optimized playlists of songs, taking into account music that user already listens to, along with music that other accounts with similar "listening profiles" consistently play. No matter the combination of attributes that forms these recommendations, however, there exists some implicit biasing factor that restricts these listening profiles from completely reflecting a user's listening preferences. There is an exorbitant amount of configurations that could be created, with each configuration assigning different weights to the factors in their respective models, so the ongoing question is as follows: which configuration is the "best" at predicting a particular user's listening preferences? Using a combination of datasets from Spotify's and Last.fm's publicly accessible listener data, we look for trends and commonalities between users' listening histories, and attempt to build a reasonably robust system that uses code to connect users' tastes together, and recommend songs that one subject enjoys to the other. Additionally, visualizations are an important aspect of transparency in the field of data, and this project demonstrates several visualizations with varying effects on their audience. Upon creating several visualizations of these datasets, it becomes apparent that certain graphs are more accessible than others, and another facet of this project's question is to discover the reason for this, and potentially utilize these graphs to enhance a user's overall experience with certain projections.

Table of Contents

I. Introduction

With a large competition of services, the significance of an application's ability to appropriately recommend new music to its users has become a fundamental feature. In a content-driven world, users of applications across multiple forms of media are searching for robust recommendations, and these systems provide a sense of personalization not offered by alternatives like radio stations. Rather than offering listeners a broad range of music that may not be entirely enjoyed, a music recommendation system typically offers music that prior data has indicated that user would likely enjoy. (Jena, 2022).

Two types of recommendation systems exist, and their differences are denoted by the userbase from which the data is derived, although many systems utilize features of both systems. A content-based recommendation system employs data from one user's listening history to generate recommendations that are stylistically similar to that history. Some features that are explored in content-based recommendation systems include methodical features like genre, BPM, and pitch, but also more exploratory features, like keywords that are used in a particular song's lyrics, the playlists a user places that song into, and the title of the song. Comparatively, a collaborative recommendation system relies on a wider community of listeners to build lists of songs for one user based on music that is enjoyed by users who have similar listening histories. In the case of a collaborative recommendation system, users who have a large overlap of songs previously listened to may be grouped together and marked for their similar tastes, and the system may recommend music that one user has listened to frequently, but the other user may not have discovered yet. On the larger scale, several users are separated based on their listening histories, which provides a grander sample size for recommending music to a particular user.

(Ripenko & Hasiuk, 2022). Both forms of recommendation systems are useful in suggesting music to users, and features from personal and collaborative systems have been utilized throughout several platforms.

In particular, Spotify is renowned for its extensive and wide-spreading recommendation systems, which employs machine learning (ML), natural language processing (NLP), and convolutional neural networks (CNN) to make educated guesses regarding how a user may tend to listen. Each three of these features are used in conjunction with one another to build recommendations for Spotify's users and utilize information from both the personal and community level. ML is utilized to "analyze user behavior and group people together based on musical preferences." (Mixson, 2021). Subsequently, ML is also used to learn the attributes of songs that are listened to by a person and recommend music that falls within similar brackets as those songs already on record. NLP is wrapped into this process, as this feature scans across various articles and blog posts to discover language that is used to describe certain music. Songs associated with similar keywords will be grouped together and suggested towards users whose listening histories indicate their tastes are aligned with those keywords. Lastly, CNN is used for the technical aspects of the music being listened to; thus, CNN is responsible for processing data on musical key, BPM, volume, and other objective details about a track. Through a combination of all of these factors, Spotify is capable of "learning" a user's music taste and accessing extraneous information to recommend a particular user additional music. (Mixson, 2021).

Despite the efficacy of Spotify's recommendation algorithm in its current state, there are certain problems preventing the system to recommend music the user will always enjoy. A study of Last.fm users conducted by Wundervald (2021) concluded that many of Spotify's automatically generated playlists tend to favor music that is more popular. Because of the native popularity bias of Spotify's music recommendation algorithm, songs with more listeners have a broader listening audience, which leads to a wider potential audience of recommendations, as opposed to the song with fewer listeners that has a smaller sample size of users that could be recommended that song. This problem constitutes a possible violation of fairness in Spotify's algorithm, an ongoing problem faced by the machine learning community in understanding how this metric can be measured. In the case of popularity bias, a fine balance must be found so songs of all levels of popularity are equitably weighted, and skewing too far in either direction of popularity could lead to unintended unfairness in the algorithm. (Wundervald, 2021).

Another prominent shortcoming of the recommendation systems of Spotify, and other music applications alike, is the integration of user personality. The demographic of a user has been shown to hold an influence in certain recommendation system, with features including race, gender, ethnicity, country of origin, and age. Melchiorre et al. (2020) conducted a study that compares the personality traits of a subsection of Twitter users to music in which these users have self-reported enjoyment. Conclusions of this study yielded results that showed how different models that were trained on the same data led to different biases from one another. On some models tested in the study, certain internal personality attributes proved more significant to a recommendation system than other attributes. Their traits of conscientiousness and extraversion were heavily weighted in certain models, but were mostly ignored in others. Because of this difference in weighting, it is revealed that all recommendation systems contain implicit biases, and rectifying these biases is a subjective matter dependent on the type of system used to attribute music towards specific users. (Melchiorre et al., 2020).

Although recommendation systems may not ever reach theoretical perfection, an understanding of user personality could enhance recommendations. Extending beyond the technical aspects of a song could allow for a more comprehensive list of music that is more prevalent for recommendation systems, thus signifying the importance of NLP in Spotify's recommendation system. Qiu and Jia (2022) tested for the significance of user-authenticated personality against a model that purely uses CNN to track the internal attributes of a user's listening taste. By extracting information about the listener's mood while listening, Qiu and Jia were able to improve the model for recommendation of music, with a heightened accuracy rate of 96.7%, as opposed to a standard 77%. For this study, accuracy was defined as a function R with respects to a music listening dataset. The results of this study demonstrated an improved recommendation system when personalized attributes were implemented, which shows the importance of non-objective elements in recommendation algorithms. (Qiu and Jia, 2022).
One other important attribute to involve within the process of recommendation systems is the inclusion of repeat products. Although a user has listening history recorded for a particular song, this fact does not imply that song can not be recommended to that user once more. Kotzias et al. (2019) tested the accuracy of prediction models when repeat events occurred and found that recommendation systems were ultimately improved when considering events that had already been tracked. For this experiment, multiple types of data were considered, including location data, social media data, and music listening data. In each of these instances, a mixed model that takes repeat events into account demonstrated an increased efficiency when compared to a system that simply introduces new events. (Kotzias et al., 2019). When considering the construction of a new recommendation system, repeat events should not automatically be

removed from a list of recommendations for a particular user. Multiple reasons to defend the inclusion of repeat items in recommendation systems relate to the interpersonal quality of certain listening habits. For example, a particular song may hold special importance to a user on a specific day, due to a concert performance or a special personal occasion. Studies have previously shown that averages between 51% and 69% of a user's listening history have included repeat songs. This means that, for the typical music listener, over half of the songs they choose to listen to are being listened more than once. (Kosetsu & Goto, 2020). Therefore, a justifiable recommendation system should take repeat listens into account.

Recommendation systems must also offer personalization to the extent of adjusting to their style of listening. Ter Bogt et al. (2010) conducted a study between different types of listeners to illustrate how certain listening styles could adjust how a certain user's choices of music is affected. Results of this experiment showed differences between pairs of attributes suggesting how one listening quality holds bearing on other qualities. Of note was the differences between time spent listening to music, music style preferences, connection of the user to music, and emotions experienced during listening. Conclusions of this experiment yielded three separate groups of listeners, all characterized by different styles of listening; the three groups were defined as "High-Involved", "Medium-Involved", and "Low-Involved", where each of these categories were aptly named after the data demonstrated by their listening statistics and personally reported feelings. Each distinct group reported different listening scores according to genre, as certain age groups were found to have higher affinities with some times spent listening, and were sorted to these groups accordingly. (Ter Bogt et al., 2010).

Recommending familiar songs to users is an efficient practice to the familiar listener who enjoys a particular subsection of certain genres and does not wish to explore more genres. Even for the user who is comfortable within their listening habits, many novice recommendation systems do not account for the change in a person's music taste over time. Therefore, to achieve more success within the personalization of a music recommender, more emphasis must be placed on the exploration of further genres. It has been demonstrated that the common music application user is interested in expanding their taste into other genres, even if these genres include commonality with the genres they typically revisit. Consequently, it is vital for recommendation systems to "nudge" users in the direction of exploration, depending on the personalization of the system for each user. (Yu & Willemsen, 2022).

Exploration of several genres has shown importance for a significant portion of users and should thus be considered in recommendation systems. Fortunately, several industry giants, such as Spotify, have implemented this functionality into their personalized recommenders, specifically splitting up internal genres into user-digestible playlists which are suggested to users dependent on the time of day, and their recent listening history. Spotify prioritizes user feedback over all other attributes to build a comprehensive taste profile for its users, and whether this feedback is active, like saves, downloads, or shares, or passive, like length listened to or repeat listens, these features are adopted into the taste profile Spotify builds. Many features are co-opted into this profile, including assumed demographics about the user themselves, most listened genres, most fringe genres, hours most spent listening to music, most listened tracks over certain stretches of time, and various other attributes. (Pastukhov, 2022). Its recommender is also helped by the splitting into several smaller playlists each focused on a certain theme. Several playlists are generated for users based on several factors like genre, time of day, and season. Understanding these groupings of attributes is paramount to comprehending the overall structure of Spotify's recommender.

The culmination of Spotify's several machine learning-oriented methods of recommendations is several playlists personalized to the users who utilize them. Two specific instances of machine learning implemented as techniques for their recommendation system are collaborative filtering and reinforcement learning. The first filtering method compares the similarities of listener profiles between many of Spotify's own users and determines which songs would be most enjoyed by specific members of a subset. Once users are grouped together contingent on their prior listening history, they continue to be grouped together moving forward; thus, any new song that is found by one user will more likely be recommended to users who have been noted as similar. Reinforcement learning is a machine learning technique that takes user feedback into account, intaking information based on the actions the user takes when interacting with a certain song. These actions can be explicit, such as clicking the green heart next to a song's name to indicate the user likes it, or passive, like leaving a song playing in the background and refusing to skip past it. Negative feedback can also help inform the algorithm to restrict a song from being repeatedly recommended to a user. If a song is consistently skipped by the user, or the user decides to click the "Hide song" button whenever they are greeted with it, reinforcement learning will understand that song is not being enjoyed by the user, and will consequently implement the user's distaste into the recommendation algorithm. Thus, that song, and perhaps songs created by that artist or falling within a similar genre, will not be as steadily recommended in the future. (Nagpal, 2023).

Improving the user experience is at the forefront of most developers' list of priorities when designing recommender systems. Determination of what is considered "good" user experience according to visualizations of applications is largely dependent on their usability and aesthetics. Both aspects, however, are integral to the user experience, and including one without the other limits quality of experience. Usability is important in terms of navigating applications, and significant in a user's capability of understanding the interface they are interacting with. However, studies show that many UX (user experience) companies have been transferring away from usability as an evaluation target, and shifting gears to prioritizing more subjective, experiential-based methods (Brown et al., 2017). This is largely because of a shift in the importance of certain stakeholders when considering UX of music-based applications. Within these applications, stakeholders are defined as performers, audiences, designers, and composers. Performers and audiences were found to be the most significant contributors to the user experience angle, and each is respectively defined as participants with agency in auditory interaction, and as participants without free agency in the evaluation of music, but holding involvement in its consumption. Performers initially seemed to prefer usability of UX above all other elements, and while it was still the most prominent feature advancing into 2016, other features like aesthetics and engagement began gaining traction. For audience, aesthetics was prioritized over all other categories, and an emotional response triggered by the user interface was also demonstrated as significant (Brown et al., 2017). A company's comprehension of its stakeholders is vital in ensuring the largest profit, so application designers can know which elements of UX to prioritize within their models.

Spotify's user experience elements behave according to many tools of proper data visualization. Tang & Yang (2017) observe the various design elements of Spotify's user interface, and how they abide by certain rules of information visualization. Researchers looked specifically at 29 users, each with separate prior experiences with Spotify's interface, and measured their satisfaction with the layout of the application. The study examined both Spotify's capacity for exploratory behavior, and how each of the users interacted with certain design elements. Results from the experiment demonstrated how most users were content with the application's search engine, and equally satisfied with a user's ability to access similar music to what was already being listened to. Specific parts of Spotify's UI that cater to a user's ability to expand on their music taste include the "Radio" feature, as well as the multiple personalized user playlists. Design-wise, this paper demonstrates how Spotify's user interface is conducive to a pleasant user experience, for both experienced and new users. (Tang & Yang, 2017).

For visualizations to be considered graphically cohesive, a number of considerations need to be taken. Tufte outlines six principles of graphical integrity within his research on proper etiquette in data visualization. Proper visualizations should include, according to Tufte, efficient numerical representation, unambiguous and detailed labels, focus on data variation rather than design variation, inclusion of standardized units for time-series data, appropriate usage of dimensionality, and a suitable ratio between filled and blank space. Tufte also remarks on the lack of intuitiveness of what he describes as "chartjunk," which vaguely describes all features of a visualization that can be considered distracting, or do not enhance the experience of observing the visual. Each of his theories regarding appropriate visualizations have grounds within the statistical fields, as these rules are commonly upheld in most common visualizations of today's expansive data. (Tufte, 1986).

One of Spotify's most famous features, Spotify Wrapped, is known for the succinct manner in which it organizes data considered interesting by its users. Aside from presenting the user with an itemized list of the music they listened to most commonly throughout the previous year, Spotify Wrapped typically profiles the user according to attributes strongly associated with users akin to their taste. Galant (2020) writes about why the user experience of previous Spotify Wrapped features have proven so successful, and how its ability to tell a story with data has proven captivating for users. From a user experience angle, Spotify Wrapped's design can be boiled down to three key features: its strong visual focus, simplified social sharing, and interactive and dynamic presentation. Its usage of bright, yet complementary colors soothe the user's eyes, and capture their attention for aesthetically pleasing slides of information. The intuitiveness of Wrapped's layout is also conducive to sharing one's data on social media, which continues to feed into its ongoing popularity. (Galant, 2020). Because of its visually pleasing layout, many users of Spotify would share this interface on other social media applications to discuss their favorite musicians, or interact with other users regarding their data. Regardless of the reason, Spotify Wrapped is framed in a way that inspires users to share their results on social media, which drives user engagement higher as each subsequent share heightens awareness of the feature. When compared to Tufte's main keys of a good visualization, Spotify Wrapped checks all the boxes, and has been proven through point of popularity to succeed as a proper visualization of user data that can be widespread to the general public. (Galant, 2020).

Ultimately, decent visualization of data enhances the usability of the application associated with it, which is integral to a user-friendly recommendation system. For a recommender to be understood by its recipient, the data needs to be fed to the user in a digestible fashion, which typically takes form in a playlist. Spotify typically lists 30 unorganized songs for each of its playlists that match its overall theme, whether that theme be a specific mood, songs newly recommended for a particular user, or songs that user has already shown repeated listening history of. Playlists are user-comprehensible visualizations, due to their interactive list format that allows their users to scroll through their contents, and solely interact with each individual track featured on the playlist, whether to show approval or disapproval of that track. With this information, Spotify can enhance their recommendation system based on the information it is fed, and adjust their future recommendations to accommodate the previous feedback. The playlist is cleanly laid out for the user with an easy-to-understand format where the connection between the songs is obvious, and the options for the user to interact with these songs are plainly visible. Because of these clean visualizations, data regarding the accuracy of a person's enjoyment of particular songs can be more trustworthy, and these visualizations lead to an overall greater quality of internal recommendations.

II. Data Processing

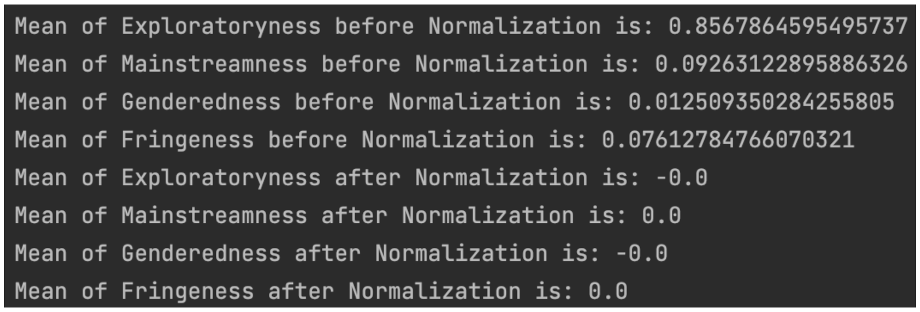
Feature-Based Data Collection

The first step in data retrieval was to access an existing music listening dataset which lists characteristics of a particular user's listening habits, and produces values that represent many users' music tastes, according to predetermined characteristics. A dataset of this nature was found within the "Music Listening History Dataset," a collection of music listening data which displays listening features of 563,293 users of the app Last.fm, an application that collects and aggregates every song listened to by a particular user since their initial sign-up for the application. The dataset contains two separate subsections of data: the first of these subsections displays information about various characteristics offered by users. Each row of this dataset represents a user record, and each user is paired with data that presents their self-reported ages, genders, date signed up for Last.fm, and internal user identification number. This information is useful in understanding the demographics of each user included in the study, and could be significant when compared against the user attributes, to understand how each user is connected to one another based on their demographics and listening habits.

The other subsection takes an amalgamation of each user's listening data, and scores them between a value of zero to one based on four automatically generated key attributes: mainstreamness, exploratoryness, genderedness, and fringeness. Each of the 563,293 users is scored according to these four attributes on the levels of album listens, artist listens, and track listens, creating a total of twelve attributes to examine per user. To understand these attributes more clearly, it is imperative to describe the context of each characteristic. A user that scores high in mainstreamness must typically listen to music that is widely listened to by other users of the website. If a user mostly listens to music that is considered "mainstream," that user will be scored closer to one than to zero, and the opposite is true for a user who listens to music that is less popular. A user that scores high in exploratoryness must listen to a wide range of artists and genres, and their listening history should reflect a willingness to expand beyond a few musical stylings. In this regard, a user who listens to music spanning many different genres and times would score closer to one, while a user who remains rigid within one genre or artist would score closer to zero. Genderedness refers to the typical listening demographic of the music a particular user has listened to. It is important to note that genderedness does not refer to the gender identities of the artists who create these songs, but rather the identities of the audience that consumes their music. If a user typically listens to male-coded music, they will score closer to zero in this category, but a user who has mostly listened to female-coded music will score closer to one. Fringeness refers to a user's history in listening to genres that are not particularly common, and a given user will score highly in this category if they tend to consume music belonging to an unpopular style of music, with special attention to the specific genre.

Feature-Based Data Processing

By observing visualizations regarding each of the four attributes on their own, we can learn more information about each singular attribute, and additionally learn more about how they correlate with one another. First, it is important to note that each attribute starts as biased in a certain direction. Although each attribute is measured on a scale between 0 and 1, each attribute is closer to either 0 or 1 than somewhere in between. It is important to normalize the data before making informative decisions regarding the stories they tell, so the data can be spread far enough to capture a reasonable deviation from the mean, and each attribute is being compared on the same scale. Because of this, the first step to processing the above data is to normalize it using a z-score standardization formula for each entry: $\frac{x-\mu}{\sigma}$. This transformation of the data will shift each overall attribute to have a mean of 0 and a standard deviation of 1. With each observed attribute using the same metric, it becomes easier to make conclusions about the data. Figure 1 shows how the means of each attribute are shifted after a z-score standardization is applied:



```
Mean of Exploratoryness before Normalization is: 0.8567864595495737
Mean of Mainstreamness before Normalization is: 0.09263122895886326
Mean of Genderedness before Normalization is: 0.012509350284255805
Mean of Fringeness before Normalization is: 0.07612784766070321
Mean of Exploratoryness after Normalization is: -0.0
Mean of Mainstreamness after Normalization is: 0.0
Mean of Genderedness after Normalization is: -0.0
Mean of Fringeness after Normalization is: 0.0
```

Figure 1. Comparison of the means of each attribute before and after normalization

As the output shows, the nature of each attribute was drastically shifted before any further analysis of the data was conducted. To show the full nature of how the data has largely altered the overall spread of the data, Figure 2 shows each attribute before normalization, and Figure 3 shows each attribute after normalization.

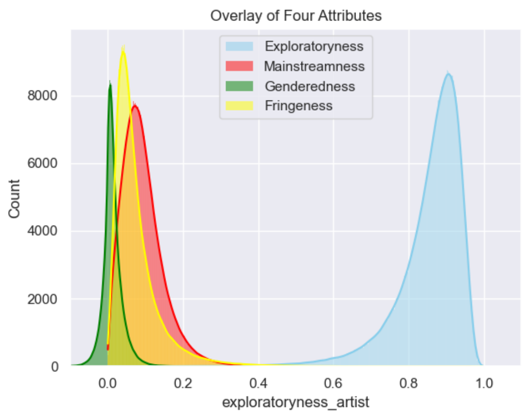


Figure 2. Overlay of Four Attributes before normalization, to demonstrate the difference in data behavior.

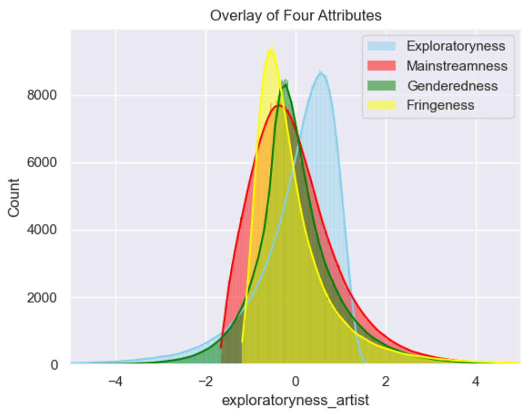


Figure 3. Overlay of Four Attributes after normalization, each with mean 0 and standard deviation 1 with roughly similar behaviors.

Because each of these attributes are now overlaid, and do not show bias in either direction, it is now reasonable to compare between these attributes to observe any existing trends. Correlation plots were built comparing each pair of the four attributes, and scatter plots were constructed to demonstrate where each entry in the dataset landed as a combination of the attributes from the pair. Figure 4 shows the correlation between exploratoryness and mainstreamness.

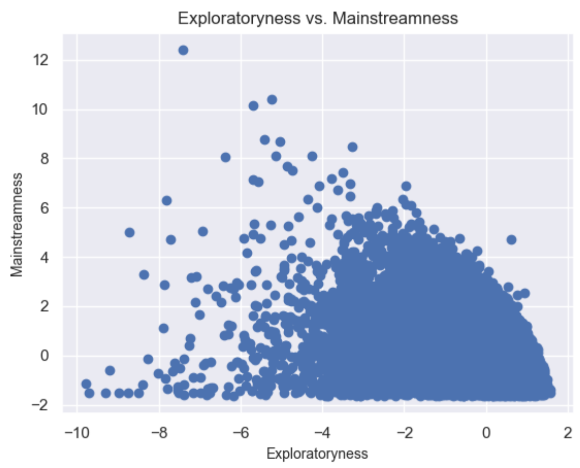


Figure 4. Scatter plot comparing exploratoryness against mainstreamness

Although this figure tells its audience some information about its contents, we concluded that this visualization was confusing in its current format. Considering that roughly 500,000 users were taken into account for this visualization, when each user is displayed as a point on a scatter plot, the result is an image that is condensed into one area, and very few points are identifiable. Because of the size of this dataset, a heatmap may make more sense to include, since heatmaps are better indicators of density within a certain region of the grid. For this visualization, a scikit package was used to construct the heatmap.

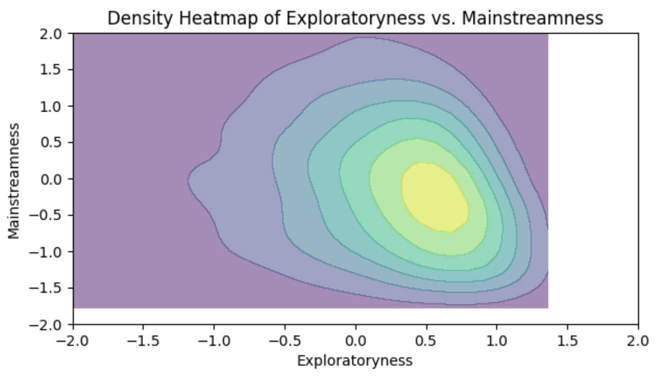


Figure 5. Density heatmap comparing exploratoryness and mainstreamness. By utilizing color and density, this visualization constructs a clearer depiction of data compared to Figure 4.

This visualization does a better job of showing where most of the data is concentrated, while also demonstrating the general downward directionality of the entire correlation. For this reason, heatmaps were preferred for data collection from each pair of attributes. Figure shows each original scatterplot of pairs of attributes compared against their respective heatmaps. Note that each heatmap exclusively looks at points between -2.0 and 2.0, which comprises the majority of the dataset, where most outliers are removed.

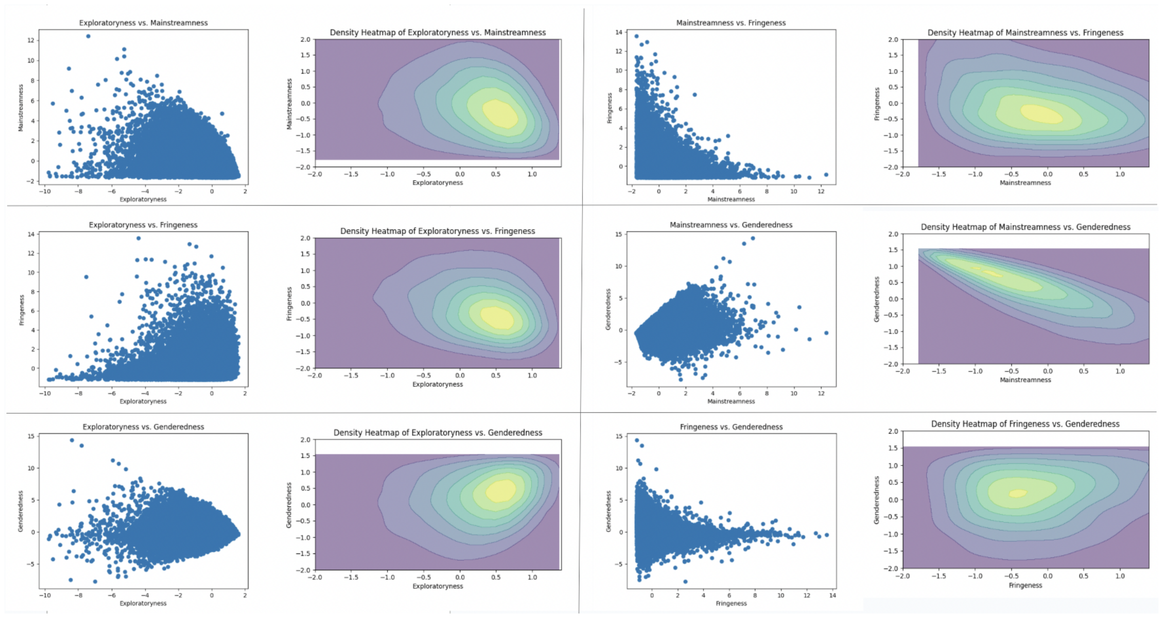
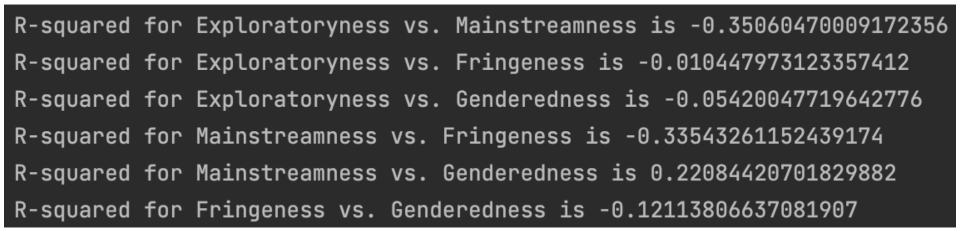


Figure 6. Comparison of the scatterplots vs. heatmaps of each pair of attributes. Each heatmap extends from -2 to 2 on the y- and x-axis.

Although a visualization of each pair of attributes has been created, it is difficult to discern correlations between some of them. When viewing the correlation coefficient for each of these pairs of attributes, where the absolute value of the coefficient represents a strong linear relationship, it becomes obvious that some of these attributes do not have a strict linear correlation with one another.



```
R-squared for Exploratoryness vs. Mainstreamness is -0.35060470009172356
R-squared for Exploratoryness vs. Fringeness is -0.010447973123357412
R-squared for Exploratoryness vs. Genderedness is -0.05420047719642776
R-squared for Mainstreamness vs. Fringeness is -0.33543261152439174
R-squared for Mainstreamness vs. Genderedness is 0.22084420701829882
R-squared for Fringeness vs. Genderedness is -0.12113806637081907
```

Figure 7. R-squared values for each pair of attributes to demonstrate their linear correlations.

From these scores, we can conclude that a linear relationship between some of these attributes may not exist, while other pairs of attributes may have a weak correlation. The R-squared value of -0.01 between exploratoryness and fringeness indicates that no linear relationship exists between these two attributes, but this does not mean these two variables are completely unrelated. The largest absolute value of a linear correlation coefficient exists between exploratoryness and mainstreamness, with a coefficient of -0.35. This indicates a weak negative correlation between the two attributes that reveals, in certain cases, that an increase in a user's explorative nature with music may decrease the amount of mainstream music they consume, and vice versa. We can not be confident in an absolute correlation between the two variables, however.

Principal Component Analysis

From this step, a principal component analysis of the data was commenced. Each attribute of the data was selected as a dimension, and a four-dimensional graph that compares all elements of the data is created. From this four-dimensional figure, the graph is flattened into two dimensions by taking the "principal component" of the data, which minimizes the distance between the original four-dimensional figure and its projection into two dimensions. The principal component is selected algorithmically in the code of this project, and is compared against another principal component. This additional component is defined as the second smallest component when comparing the distance between the original figure and its projection into two dimensions. This principal component must also be uncorrelated with the first component, so the resulting component is orthogonal to the original (Lever et al., 2017).

Constructing a principal component analysis of this data can inform us which variable is most informative in determining a user's listening habits within this data. When PCA was applied to the Music Listening History Dataset, the following figures were created:

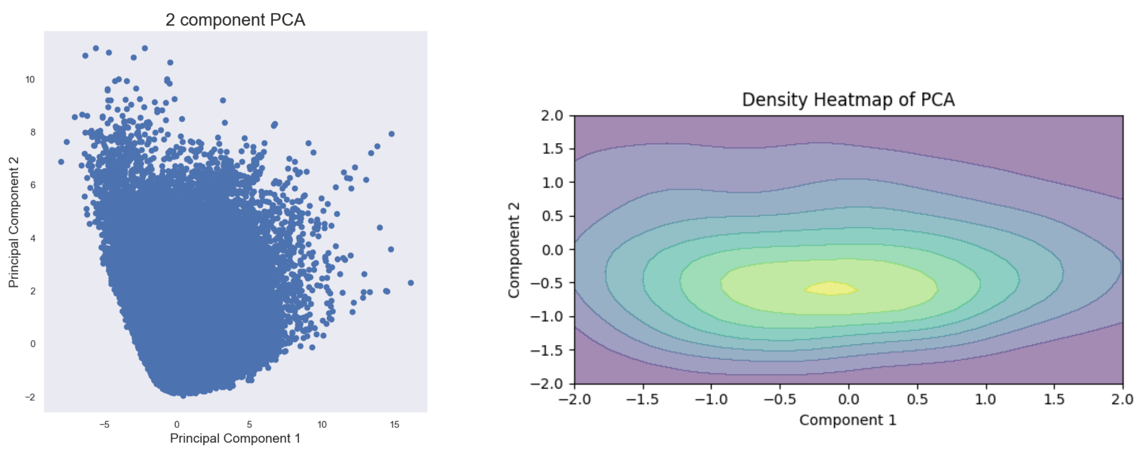
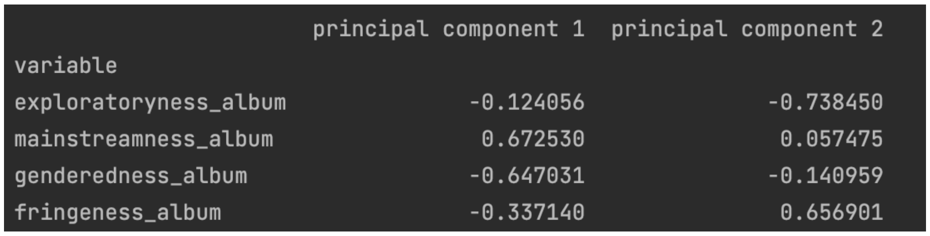


Figure 8. Principal Component Analysis of four attributes displayed through scatter plot and heatmap. The density heatmap shows a concentration around (x, y) = (-0.2, -0.5).

The following figure shows the amount of each variable used in the formation of each principal component. The values for each component represent the eigenvectors for each variable with regards to that component, demonstrating the correlation between each variable and its respective component.

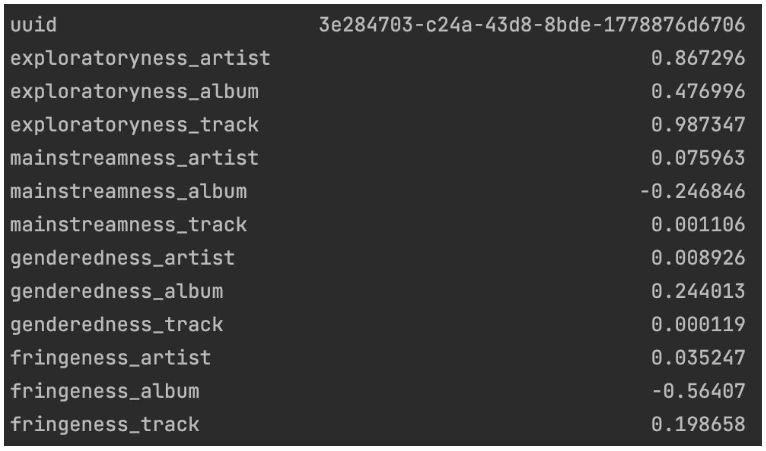


```
                      principal component 1  principal component 2
variable
exploratoryness_album              -0.124056              -0.738450
mainstreamness_album                0.672530               0.057475
genderedness_album                 -0.647031              -0.140959
fringeness_album                   -0.337140               0.656901
```

Figure 9. List depicting each attribute's effect on each principal component.

Each variable is significant for at least one component, as mainstreamness and genderedness are shown to have eigenvectors of 0.67 and -0.65 for principal component 1, while exploratoryness and fringeness are shown to have eigenvectors of -0.74 and 0.66 for principal component 2. These values demonstrate the combination of variables that will offer the best approximation of where a user within the dataset can find themselves within the grid in Figure [].

Grouping users together based on these internal attributes is now possible due to the principal component analysis performed. If a particular user's scores were substituted into the equation for principal component 1, it becomes possible to determine their position with respects to all four attributes. Consider user 3e284703-c24a-43d8-8bde-1778876d6706, who scored the following normalized values for each attribute. Recall this analysis was conducted on the album level, so the variable values for albums are the ones that will be utilized.



```
uuid                       3e284703-c24a-43d8-8bde-1778876d6706
exploratoryness_artist                                 0.867296
exploratoryness_album                                  0.476996
exploratoryness_track                                  0.987347
mainstreamness_artist                                  0.075963
mainstreamness_album                                  -0.246846
mainstreamness_track                                   0.001106
genderedness_artist                                    0.008926
genderedness_album                                     0.244013
genderedness_track                                     0.000119
fringeness_artist                                      0.035247
fringeness_album                                       -0.56407
fringeness_track                                       0.198658
```

Figure 10. Scores for each attribute for sample user from the Music Listening History Dataset.

Principal component 1 can be defined as PC = $-0.142i_1 + 0.675i_2 - 0.643i_3 - 0.333i_4$, where i represents each respective attribute from the data. Using the data given from the user, we can determine their value for the first principal component as -0.204. Similarly, principal component 2 can be defined as PC = $-0.724i_1 + 0.053i_2 - 0.134i_3 + 0.675i_4$. When considering the above user, the value for their second principal component would be -0.010. Thus, on the image from Figure [], this user's listening data would exist at (x, y) = (-0.204, -0.010), and it may be possible to craft a recommendation system comparing other users that are geometrically close to that user.

Song-Based Data Collection

After collecting more information about the listening history of this entire dataset, it became possible to generate recommendations to certain users based on the type of music that users with similar scores enjoyed listening to. However, the data included no information about the specific songs or albums listened to by its participants, which complicated the matter of recommending any user a particular list of songs. To make headway in this direction, it became important to search for data where we have specific information about the listening histories of the users. Since all of a Last.fm user's listening history is public source under the permission of its terms of service, it is possible to extract information about the tastes of any desired number of users. For the sake of this study, the listening history of 100 randomly selected Last.fm users was considered, and downloaded using a web compiler converting each listening history to a comma-separated value file. In the outgoing text file, each record includes the name of a song, its artist, its album, and the time that song was listened to by a given user.

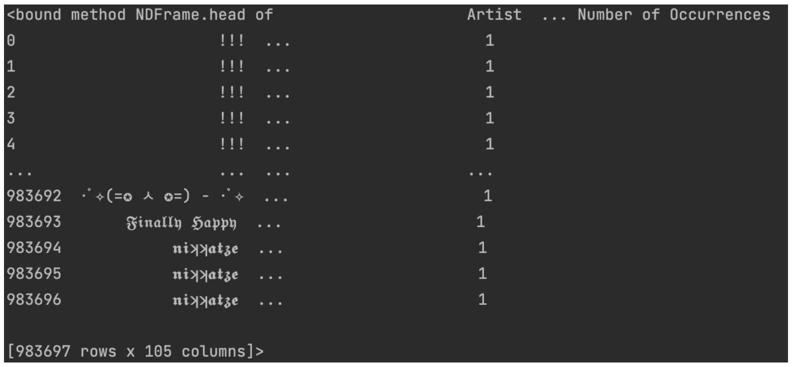
The process of selecting usernames for this study was purposefully as random as possible, although could still be prone to human error. For some users included in the dataset, there was an emphasis for more "committed" users who extensively listen to one artist at a time, and may not have much variability in their listening preferences. Last.fm's interface allows for some of a particular artist's top listeners to be displayed, meaning users who have listened to that particular artist more than most others will be shown on that user's homepage. For the artists selected in this study, a wide range of genres, audiences, popularities, and years of popularity was attempted. For example, some users were avid listeners of 2010s pop artist Taylor Swift, other users were avid listeners of 1980s thrash metal band Megadeth, and others were primary listeners of more obscure artists. The selection process also accounted for different languages, as some chosen artists typically had lyrics in Spanish, Russian, or Portuguese. Another selection strategy was to search for certain users, and then collect the data of other users that the original user was "friends" with. One preconception held before entering this project was that users who interact with one another on a regular basis would share a somewhat similar music taste, and tend to listen to the same style of music. This hypothesis could be tested on the user level, if a researcher was interested in comparing the music taste of particular users. This factor did not come up throughout the experiment, as most of this research was conducted on the song level, although this could be a potential direction for further research in how socially connected people may share interest in certain styles of music. Although this listener data does not accurately represent the entire listening population, nor the listening population of Last.fm users, it includes enough variation for more in-depth research about these users.

Song-Based Data Processing

The formation of this dataset allows for more nominal information when compared to the Music Listening History Dataset. That dataset, constructed by listening analysts Vigliensoni and Fujinaga, includes holistic information that scores individual users against internal attributes that are selected using their own equations. These calculations are useful in obtaining broader information about how demographics listen to categories of music, especially against the internal qualitative measures utilized by Last.fm within their own personal recommendation system. One weakness of this dataset, however, is that it does not include specific information on the song level about a user's listening habits. There is no publicly accessible information about the specific artists, songs, or albums listened to by a particular user, which would introduce a difficulty for an outer recommendation system using this dataset to make conclusions about specific songs that could be recommended to specific users. In the dataset using nominal information about the music listened to by a chosen subset of users, there would be extensive information about which songs are enjoyed by which users, and it becomes simpler to connect users from this dataset together when analyzed on a song level. The benefits of a recommendation system that includes information about specific past songs listened to by users is that this system could immediately pick out songs that already exist in the dataset, and suggest these songs to users who have not yet listened to them, but exhibit a music listening history that implies they would enjoy that song. With the Music Listening History Dataset, songs would have to be introduced from outside the immediate dataset, scored against their internal attributes, and then assigned to the playlists of certain users depending on how closely the scores of their attributes match the attributes of that particular song. The introduction of a personalized song-level recommendation system would eliminate the need to consult a resource where each song is scored against these attributes, and also introduces a sense of personalization and community-based recommendation that is absent from the original model. The construction of a master dataset that contains all 100 text files of the users of this dataset would combine existing song information, and instantly return values for each song when contrasted against each user.

III. Analysis

Once all 100 text files were combined into one master dataset, there were a total of 12,852,171 records of any song being listened to by any of the included users. Every song that was listened to more than one time was then combined, so there were 983,967 unique songs listened to by any number of these users. With this new list of combined music, it was possible to construct a new dataset with 983,967 rows and 101 columns, where each row corresponded to any song, and each column referred to one of the 100 listeners, with an additional column accounting for the name of a particular song. In each combination of song and user, there was a numerical value of either 0 or 1, which demonstrated whether or not that user has ever listened to that song within their listening history. The head of this dataset is included in Figure 11.



```
<bound method NDFrame.head of                  Artist  ... Number of Occurrences
0                    !!!  ...                     1
1                    !!!  ...                     1
2                    !!!  ...                     1
3                    !!!  ...                     1
4                    !!!  ...                     1
...                  ...  ...                   ...
983692  ·˚✧(=✪ ⅄ ✪=) - ·˚✧  ...                     1
983693       𝔉𝔦𝔫𝔞𝔩𝔩𝔶 ℌ𝔞𝔭𝔭𝔶  ...                     1
983694            𝖓𝖎𝖐𝖐𝖆𝖙𝖟𝖊  ...                     1
983695            𝖓𝖎𝖐𝖐𝖆𝖙𝖟𝖊  ...                     1
983696            𝖓𝖎𝖐𝖐𝖆𝖙𝖟𝖊  ...                     1

[983697 rows x 105 columns]>
```

Figure 11. Head of the dataframe containing 983,696 songs, showing the artist of a particular song and number of users who had listened to that song.

Jaccard Index Scoring

The dataset in its current form allows each song to be compared to each user, and returns 0 if there exists no listening history from that user of that song, and 1 if a history is tracked. However, in recommendation systems, it would be prudent to compare songs against each other, rather than compare them to a list of predetermined users. This would be advantageous for a recommendation system, because songs that are closely related could be recommended from one user to another. In this sense, it is possible to create a large-scale matrix that compares each individual song against one another, and assigns a score to each pair depending on their similarity in users. For the purposes of this project, a matrix would be constructed of 983,967 rows and 983,967 columns, where each song within the dataset is tested against any other song within the dataset for its similarity in listeners. This measure is calculated using the Jaccard index, a formula measured as $\frac{(A \cap B)}{(A \cup B)}$. The Jaccard index of two sets measures the similarity between them dependent on the members of each set. It is useful within the context of this dataset, due to the asymmetrical nature of the listener data. The Jaccard index can be used for asymmetric data of binary value, where each value is weighted separately. In the instance of the listener data from this project, a score of “0” represents no listener data tracked for a particular song from that user, while a score of “1” represents the presence of listener data. The nature of this data is asymmetric, since a score of 1 would be more informative than a score of 0 for building a recommendation system. The presence of a track being listened to is more informative than the lack of listening, since the tracking of a particular song can be used to recommend songs the system may deem similar. For the purpose of this dataset, this measure gives a score between zero and one that judges the similarities between two records, based on the overlap of listeners. The total number of users who have listened to both songs will be divided by the total number of users who have listened to at least one song.

Because the denominator will always be equal to or greater than the numerator, the largest possible Jaccard score is 1, which would demonstrate that every user who has listened to either song has also listened to the other one. Alternatively, a Jaccard score of 0 would indicate the two songs included in this study would have no listeners in common with one another, and would be treated as entirely dissimilar. Because each song in the dataset is being compared with every other song, this measure would offer a concrete value to each pair of songs showing their similarity to one another, based on the amount of listeners they share in common. In this sense, the scores of interest in this study would be closer to, but not precisely, one. A pair of songs with a Jaccard score of 0.9, for example, could indicate that 9 users have listened to both songs, and 10 users have listened to either of the two songs. Thus, a recommendation system would be able to take that score and suggest one of these songs to the user who has not listened to both yet, as the Jaccard score indicates these two songs have a high similarity. However, a Jaccard score of 1 would prove uninformative in the context of recommendation systems in this regard, as each listener who has already listened to one song has also already listened to the other, and thus would not benefit from being recommended a song they already have a history of listening to.

A sample of 100 randomly selected songs was taken, and their Jaccard scores were lined up in a distribution to demonstrate the spread of scores seen within the data.

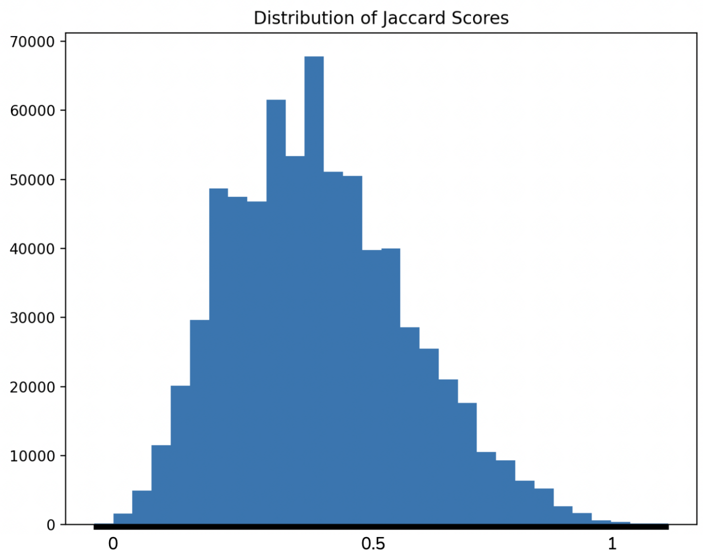


Figure 12. Distribution displaying the amount of Jaccard scores witnessed within the data.

Within this diagram, it is evident that most of the Jaccard scores occur within the range of 0.2-0.4, which is also shown by the following chart which displays the most common Jaccard index scores, and the amount of times they occur within these rows:

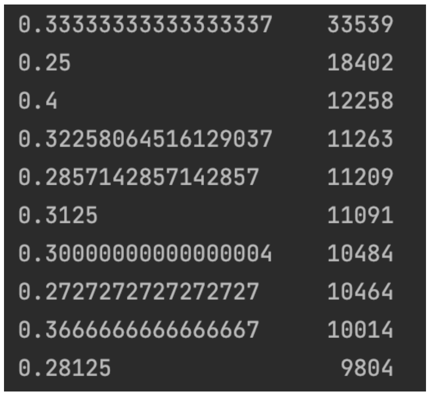

```
0.33333333333333337    33539
0.25                   18402
0.4                    12258
0.32258064516129037    11263
0.2857142857142857     11209
0.3125                 11091
0.30000000000000004    10484
0.2727272727272727     10464
0.3666666666666667     10014
0.28125                 9804
```

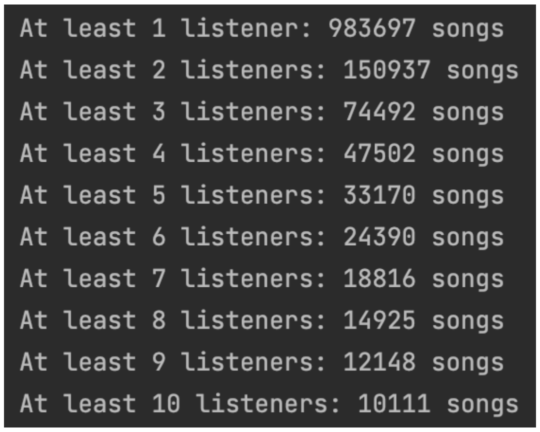
Figure 13. List of most commonly occurring Jaccard scores, with their total number of occurrences.

The most common Jaccard index, therefore, refers to $\frac{1}{3}$, where some factor of three songs were listened to by at least one user, but an equivalent factor of one song was listened to by both users. For this data, only songs with at least 3 listeners were considered, thus explaining the absence of 0.5 from the most commonly occurring Jaccard scores. Although they are less commonly occurring, a measure of strong recommendation would occur further to the right on the distribution, as the greater Jaccard index implies the belief that multiple people who listened to one of the songs have also listened to the other.

Song-Pair Matrix Creation

Based on these individual Jaccard scores, it becomes possible to measure the distances between songs geometrically. This can be accomplished by computing the dissimilarity between each pair of songs, which can be taken as the reciprocal of the original Jaccard score. When inverted, each score can be interpreted as the geometrical distance between one song and another, and if a particular song scores closer to 0, this means it shares most of its listeners in common. Any song when paired with itself should give a dissimilarity score of 0, which would imply they have no distance from one another. When each of these scores are converted, a localization function computes a set of planar coordinates that optimizes the distances of songs between each other. As a result, each individual song is given a position on this plane, and any other song that is geometrically close to that point could be considered similar. For the sake of a recommendation system, this means that any song that is close to a particular track would be considered as a strong recommendation for any user who enjoys the original track.

Internally, there was also an additional column that counted the total occurrences of 1s in each row; this column represented how many users listened to the particular song that was attributed to that row. The inclusion of this column expedites the process of filtering out songs depending on how many users have listened to that particular song. For memory reasons, it will prove efficient to only look at songs that have a certain number of listeners. By raising the lower limit of listeners necessary for a song to be included in the dataset, the processing speed of the machine in analyzing the matrix would benefit. Following is a list of how many songs were included under each parameter of a certain number of listeners.



```
At least 1 listener: 983697 songs
At least 2 listeners: 150937 songs
At least 3 listeners: 74492 songs
At least 4 listeners: 47502 songs
At least 5 listeners: 33170 songs
At least 6 listeners: 24390 songs
At least 7 listeners: 18816 songs
At least 8 listeners: 14925 songs
At least 9 listeners: 12148 songs
At least 10 listeners: 10111 songs
```

Figure 14. Amount of songs existing within the dataset that possess a related minimum number of listeners.

Although increasing the lower limit of listeners drastically reduces the number of analyzable songs, the tradeoff is a reduction in memory usage. However, the removal of songs with a certain amount of listeners does not automatically create a worse model. Because some pairs of songs in the dataset only have one listener, these songs are only able to have a Jaccard score of 0 or 1, and both of these are uninformative for recommendation purposes. By removing songs with only one listener, the dataset becomes less diluted, and the overall performance of the model increases.

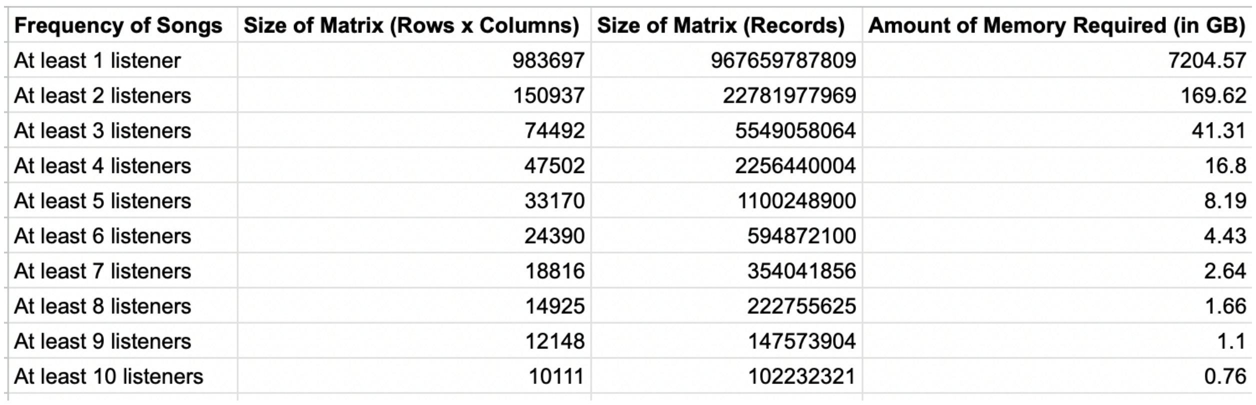
Memory Allocation

For the start of this portion of the project, the matrix was run on PyCharm CE on a machine with 8 gigabytes of RAM (Random-access memory). However, this machine was only capable of analyzing 25,000 songs before running into a runtime error: “MemoryError: Unable to allocate x GiB for an array with shape (y, z) and data type float64”, where x refers to the amount of gigabytes required to run the specified array, and y and z respectively represent the rows and columns of the requested matrix. Upon further investigation, Python threw an error due to the large size of the dataset surpassing the processing power of the machine being operated on. Since the overall dataset consisted of 983,697 individual records, this means the machine was only capable of localizing 2.5% of the songs included in the dataset, and 0.065% of the overall comparisons between separate songs. With less than one thousandth of the data being processed, reasonable conclusions that encompass the entire dataset are impossible. Even if one was excluding songs with no more than 2 listeners, this study would only include 25,000 of a total 150,937 songs of interest, an inclusion rate of approximately 16.6%. This is a rather small subsection of the dataset that is not suitable for making assumptions about the overall listening population. To combat this issue of this subset’s inability to represent the broader data, the dataset was transferred to a machine with 16 GB of RAM. Even at 16 GB, however, roughly 45,000 songs were analyzed in the dataset, which still consists of less than half of the song population with two or more listeners, and a mere 4.5% of overall pairs of songs. Reading in a matrix of this size proved problematic for machine handling, so a comparison between the two machines was run to determine the maximum number of songs that could be analyzed by each machine.

To compare the performance of the code on a separate device, the file computing the Jaccard indices of each pair of songs was run on a computer with 128 GB of storage. The machine was capable of analyzing every pair of songs with at least 3 listeners, which consists of 74,492 songs, with the file taking up roughly 41 GB of space. Therefore, every song with at least three listeners was capable of being compared with any other song with at least three listeners without machine limitations. When attempting to compute the matrix of every song with at least two listeners, however, the machine encountered a runtime error, noting that it was unable to allocate enough memory to process the matrix. By comparing the performances of each of the three machines, it is possible to estimate the memory requirement as a function of the number of songs analyzed.

When the code was run on the 8 GB machine, the error message specified that 16.8 GB were required to run a matrix of size 47,502 x 47,502. Even when adjusting the size of the matrix, it stated that 8.20 GB were required to run a matrix of size 33,170 x 33,170. Thus, in this experiment, an 8 GB machine would only be capable of running a matrix containing songs with at least 6 listeners. When transferring over to the 16 GB machine, the code was still not executable, but was quickly processed in the 128 GB sphere. Running the dataset that included at least 3 songs was possible on the 128 GB machine, but every song with at least 2 reported listeners was still not possible. Using the runtime errors from each machine, however, it is possible to estimate the amount of storage space required to form a matrix from the entire dataset.

From the previous error messages, it is demonstrated that 8.20 GB of RAM is required to process the matrix of size 33,170 by 33,170, and 16.80 GB of RAM is required to process the matrix of size 47,502 by 47,502. In the first case of 8.20 GB, 1,100,248,900 comparisons between songs are included, while the second case of 16.80 GB creates 2,256,440,004 records of comparisons. The proportion of gigabyte usage to included individual records appears consistent: 16.80 GB is approximately 2.049 times larger than 8.20 GB, and 2.256 billion is approximately 2.051 times larger than 1.100 billion. This is further exacerbated by the file of size 74,492 by 74,492 using 41.30 GB of RAM. Once more, the amount of increased records is proportional to the amount of increased memory: there is a 2.459x increase in records, and a 2.458x increase in memory when compared to the next largest matrix. Using these figures, the following chart was constructed containing the amount of memory required for each subset of songs.



Frequency of Songs	Size of Matrix (Rows x Columns)	Size of Matrix (Records)	Amount of Memory Required (in GB)
At least 1 listener	983697	967659787809	7204.57
At least 2 listeners	150937	22781977969	169.62
At least 3 listeners	74492	5549058064	41.31
At least 4 listeners	47502	2256440004	16.8
At least 5 listeners	33170	1100248900	8.19
At least 6 listeners	24390	594872100	4.43
At least 7 listeners	18816	354041856	2.64
At least 8 listeners	14925	222755625	1.66
At least 9 listeners	12148	147573904	1.1
At least 10 listeners	10111	102232321	0.76

Figure 15. Table listing memory values required to run pair-of-songs matrix

To analyze every song that has been listened to by at least one of these 100 selected users, a machine with at least 7204.57 gigabytes of memory is required, or 7.20457 terabytes. However, not all of the information is strictly necessary to construct a recommendation system. Of the 983,697 songs with recorded listening histories, 832,760 of these songs have only been listened to by one user. These songs are *hapax legomena* within this dataset, as they only occur once in the context of listeners, and thus may not be significant for results. In the context of Jaccard scores, these records are only capable of having scores of 0 or 1, which are both uninformative for creating recommendation systems powered by similarity in songs. A Jaccard score of 1 indicates that everyone who has already listened to at least one song in the pair has already listened to both, and a score of 0 indicates that everyone who has already listened to at least one song in the pair has not listened to any of the others. In both cases, the song with 1 listener would not be included in a recommendation system for any user, and would thus hold minimal significance in the construction of a recommendation system.

As the number of listeners for a particular song increases, the number of possible Jaccard index scores for that song coincides. Any song with exactly two listeners has possible Jaccard scores of 0, 0.5, and 1, respectively corresponding to the cases where 0 of its 2 listeners were reported listening to another song, 1 of its 2 listeners were reported, and both listeners were reported. With each additional listener to a particular song, another possible Jaccard index score is achievable. By this metric, a song with N amount of listeners is capable of N+1 possible Jaccard index scores. By extension, a song with a larger amount of listeners tells the most information about comparisons between that song and another. It is particularly valuable when comparing two songs that both have a substantial number of listeners, as there can be a larger confidence in the true similarity between the two songs. If two songs both have a large group of listeners with a recorded experience of listening to either song, but a low amount of listeners in common, then one can be confident that the two do not share a strong common audience. Including more listeners also increases the highest possible Jaccard score index of those two songs. When there are only three listeners between a pair of songs, the highest possible informative (non-1) score is 0.75, but with nine listeners between a pair of songs, the highest informative score is 0.9, which a recommendation system would prefer.

For these reasons, a song with more listeners is more informative for the recommendation system, but even songs with a low amount of listeners can still prove informative. It is possible that many songs with a high number of listeners do not share overlap with other songs with a high number of listeners. For instance, there may be ten fans of a genre like metal who have all listened to the same track. There may also exist 15 fans of pop music who have listened to another track. Even though both of these songs have a high number of listeners individually, their overlap may not be strong, which would lead to a noticeably low Jaccard index score. In this sense, the recommendation system may opt for a song which has a lower number of listeners, but still shares a majority of those listeners in common. Because of this, any song with at least two listeners could be informative for this system, and a machine that can appropriately analyze the 150,937 songs with at least 2 listeners would be considered thorough.

Only songs with at least three listeners were considered, but due to the potentially counterproductive nature of analyzing songs with only one or two listeners, this would be considered thorough enough to craft a reasonable song-based recommendation system. After each song was placed in a 74,492 row matrix, each of their scores was converted into a distance that subtracted its Jaccard index score from 1. At this point, a dissimilarity graph was created to algorithmically sort the optimal geometric distances between songs on the grid, so songs with low distances would appear close to one another, and songs on opposite sides of the map would have a high distance. As distances were calculated as the reciprocals of Jaccard index scores, songs with a low distance score would reportedly have a high Jaccard index score with those around it. The grid was created using the scikit-learn package “manifold”, which is useful in linear dimensionality reduction. Much like the multidimensionality of attributes in the Last.fm Historical Dataset, the matrix of pairs of songs creates a graph of several dimensions.

IV. Results

To begin this analysis, a subsection of songs that each have at least 23 listeners in the dataset was taken. Only 1,701 songs were included in this sample, and therefore the sample would not be representative of the entire listening population, and prone to popularity bias, where songs with more listeners are typically favored within recommendation systems. Creation of a plot for these songs yields the following graph:

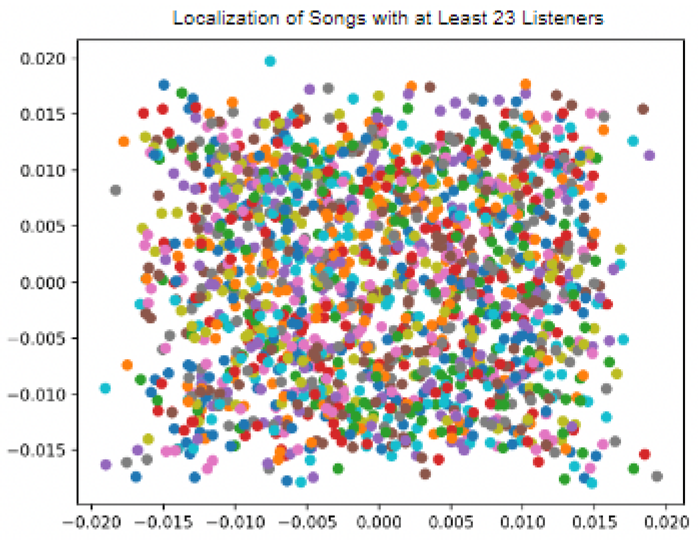


Figure 16. Song space for each song with at least 23 listeners

To test that songs with similar attributes were being appropriately grouped together, some songs created by the same artist were labeled. Theoretically, songs created by one artist would appear somewhat similarly to one another on this graph. To test this theory, labels were created for multiple songs created by Taylor Swift.

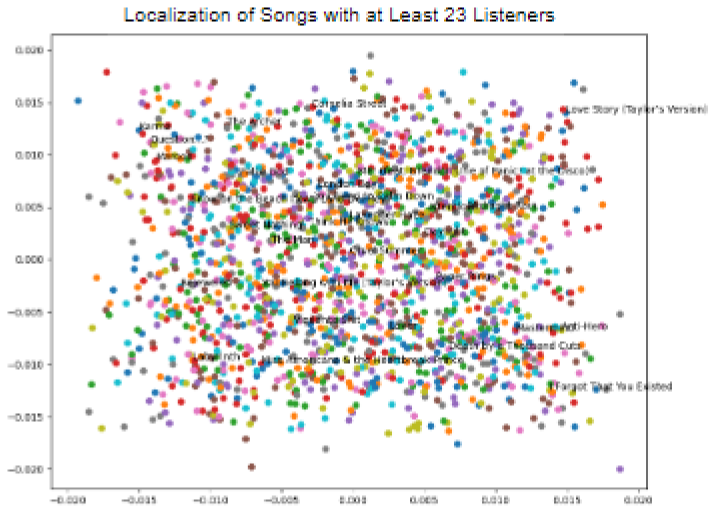


Figure 17. Song space for each song with at least 23 listeners, with some songs created by the same artist, Taylor Swift, possessing labels.

Although many of the labeled songs on this chart are somewhat close to one another, there is an unexpected amount of clustering across the entire graph. It is quite likely that songs authored by the same artist would appear quite close to each other on this graph, but that is not the case for these songs. Even consistency of albums seems to be ignored, as “I Forgot That You Existed” and “The Archer” come from the same album, but appear to be on opposite sides of the graph. To further test for accuracy of this graph, random songs were labeled, and found more information about each of them to see if their distances from one another was believable. Following is a graph with randomly labeled songs from the dataset.

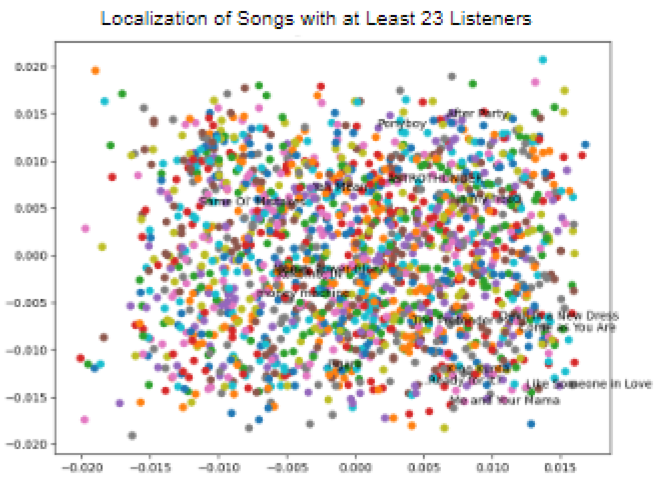


Figure 18. Song space for each song with at least 23 listeners, with randomly selected songs possessing labels.

Near the center of the graph, we see three songs appear quite close together: “Money Power Glory”, “Kokomo, IN”, and “money machine”. Authored by Lana Del Rey, Japanese Breakfast, and 100 gecs respectively, all three do have an overlap in terms of genre. However, there is a quantitative difference between the three, as “money machine” is considered by songbpm.com to have high energy and high danceability with a BPM of 99, while “Money Power Glory” has low energy and very low danceability at a BPM of 137, and “Kokomo, IN” has high energy and medium danceability with a BPM of 101. Of these songs, “Kokomo, IN” and “money machine” have a Jaccard index score of 0.5, “money machine” and “Money Power Glory” have a Jaccard index score of 0.296, and “Kokomo, IN” and “Money Power Glory” have a score of 0.431. With these scores, these songs should not be close to one another, which could be an indicator that either more songs need to be included, or the principle component analysis is not functioning as intended.

When the sample size was increased to include any song with at least 10 listeners, a much denser scatter plot appears, consisting of 12,148 songs. The overall plot appears as the following:

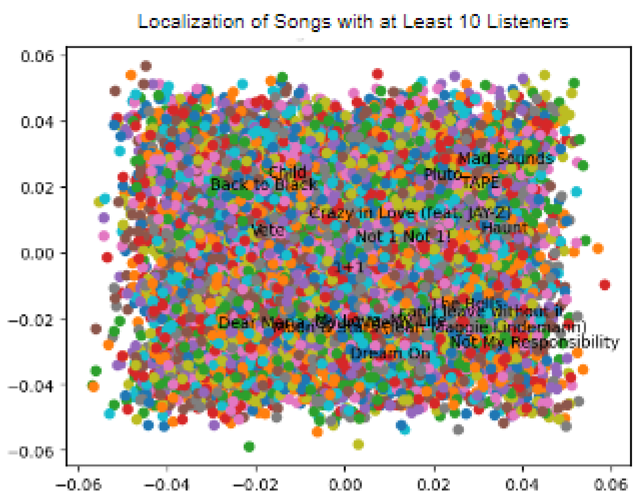


Figure 19. Song space containing every song listened to by at least ten users from the dataset, with randomly selected songs labeled.

Upon visual inspection, the accuracy of this graph appears to have increased, with songs under a similar genre appearing close to one another, like “Moon & Stars (feat. Maggie Lindemann)” and “can’t leave without it”, as they are both classified under the genre of hip-hop. Upon analyzing the file containing the several distances between pairs of songs, it appears these two songs have a Jaccard index score of 0.75, which is a higher score than those similarly grouped in the previous run. In addition, although they come from separate genres and energy scores, “Not My Responsibility,” a song by pop artist Billie Eilish, has a significant correlation with both of these songs, at 0.55 and 0.68, respectively. To test this theory further, it is possible to zoom into a smaller subsection of the graph, and check the similarity between those. For this purpose, we zoom into a portion of the graph, and find a subsection of points.

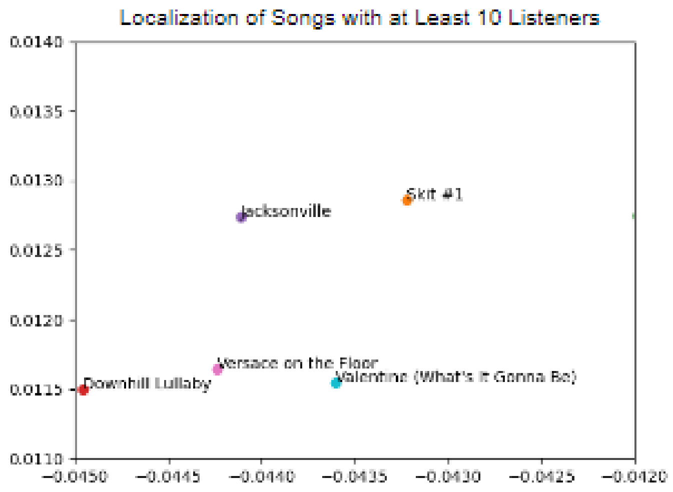
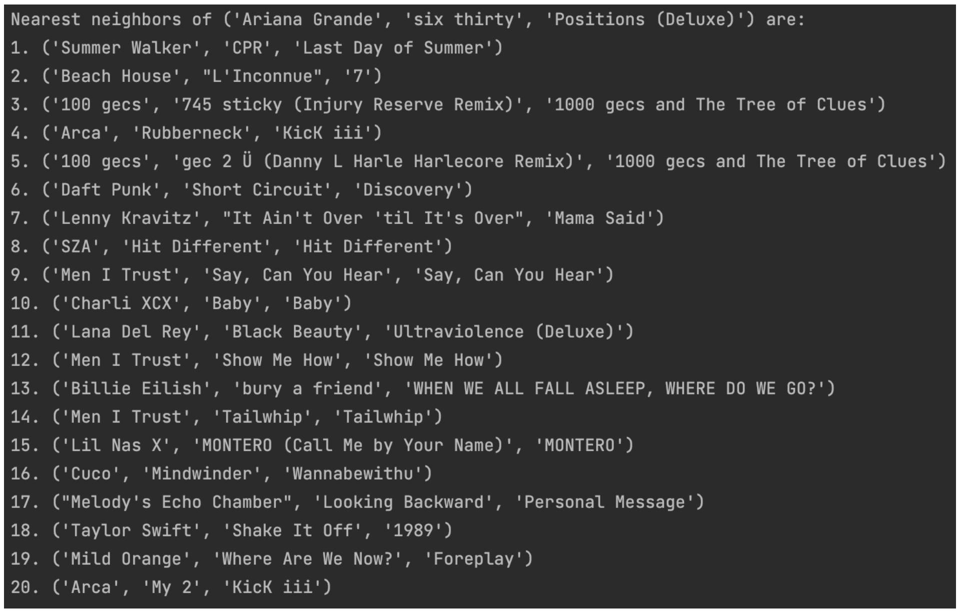


Figure 20. Zoomed in section of song space from Figure 19, for songs between x = [-0.045, -0.042] and y = [0.011, 0.014]

All songs listed within this diagram are somewhat similar to one another, as they are all medium-energy pop songs, aside from “Skit #1”. The process was repeated for certain other subsections of the graph, and each subsection contained songs with similar energies and genres, according to outside metadata sources. Based on this, it is reasonable to conclude that the inclusion of more songs within this model led to a more complete image of the song space, that could be even more accurate with more songs included.

Branching from this step, it is important to store the information of similar songs in a list format, in addition to displaying them geographically. Using the list of localizations as scores for how similar certain songs are to one another, it becomes possible to construct a list of songs that have the lowest distance to any given song, above 0. For any song in the dataset, a list of recommendations can be given based on similarity to that particular song. Following is a list of recommendations that are similar to “six thirty” by Ariana Grande.



```
Nearest neighbors of ('Ariana Grande', 'six thirty', 'Positions (Deluxe)') are:
1. ('Summer Walker', 'CPR', 'Last Day of Summer')
2. ('Beach House', "L'Inconnue", '7')
3. ('100 gecs', '745 sticky (Injury Reserve Remix)', '1000 gecs and The Tree of Clues')
4. ('Arca', 'Rubberneck', 'KicK iii')
5. ('100 gecs', 'gec 2 Ü (Danny L Harle Harlecore Remix)', '1000 gecs and The Tree of Clues')
6. ('Daft Punk', 'Short Circuit', 'Discovery')
7. ('Lenny Kravitz', "It Ain't Over 'til It's Over", 'Mama Said')
8. ('SZA', 'Hit Different', 'Hit Different')
9. ('Men I Trust', 'Say, Can You Hear', 'Say, Can You Hear')
10. ('Charli XCX', 'Baby', 'Baby')
11. ('Lana Del Rey', 'Black Beauty', 'Ultraviolence (Deluxe)')
12. ('Men I Trust', 'Show Me How', 'Show Me How')
13. ('Billie Eilish', 'bury a friend', 'WHEN WE ALL FALL ASLEEP, WHERE DO WE GO?')
14. ('Men I Trust', 'Tailwhip', 'Tailwhip')
15. ('Lil Nas X', 'MONTERO (Call Me by Your Name)', 'MONTERO')
16. ('Cuco', 'Mindwinder', 'Wannabewithu')
17. ("Melody's Echo Chamber", 'Looking Backward', 'Personal Message')
18. ('Taylor Swift', 'Shake It Off', '1989')
19. ('Mild Orange', 'Where Are We Now?', 'Foreplay')
20. ('Arca', 'My 2', 'KicK iii')
```

Figure 21. List of 20 songs considered "most similar" to "six thirty" by Ariana Grande according to the created recommendation system

It is possible to find this for any song in the dataset. Each song on this list has a reasonably low distance when compared with the original. "CPR" has the lowest distance at 0.1754, followed by "L'Inconnue" at 0.1786. Even the lowest song on this list, "My 2", has a distance of 0.2373, which indicates a reasonable crossover in audience between the two songs. This functionality could be used for something similar to Spotify's "song radio" feature, which lists songs that are similar to the one currently played by the listener. This is one way for a recommendation system to function, given the particular user already knows a song they are aiming to find similarities with.

V. Discussion

On a broader scale, though, a combination of these songs is required to create a thorough list of recommendations based on a user's prior listening history. If a user had a listening history of roughly ten songs, for example, a list of recommendations would be created based on songs that are similar to a majority of those ten. On the map of songs plotted near similar ones, recommendations may be represented as points that fall close to a majority of the other songs that have already been listened to. A convex hull may subsequently be created between those ten songs, and any song that falls within the borders created by the convex hull could be considered for recommendation. This is because, if a song falls within a convex hull of ten songs that are presumably already close together, as they share at least one listener, then any song within that hull may be stylistically close to most of those ten initial songs.

Data Visualization

For this project, a variety of visualizations were generated, each demonstrating a varied level of potential comprehension. For the scatterplots of certain attributes within the Music Listening History Dataset, it became evident that the heatmap was ultimately more informative than the generic scatterplot. Within the heatmap, the audience is capable of finding where the majority of the data lie, and having a clearer understanding of the correlation between the two variables. Even in the principal component analysis, the heatmap rendition was ultimately more informative in telling the audience the shape of the graph, and pinpointing exactly where most of the attributes lay in the two-dimensional sphere. Data visualization holds an integral part in user comprehension, which is key in music recommendation systems. The most familiar form of recommendation visualizations to the modern music application user is the playlist, where a list of songs is cleanly laid out for the user to scroll through and pick out their favorites.

However, other visualization techniques could be informative, even on the user level, in offering information about songs and artists that user may enjoy. A song space like the one generated in Figure 19 could help that user manually find similar songs, if the map offered a level of interaction where the user could click on a particular song, and find its nearest neighbors. If users were interested in learning how their listening habits have changed over time, a time-series line graph could offer information in this regard. For example, one line could represent the average energy of the songs that user listened to over time, and another line could represent how other attributes have also been affected.

Visualizations are key in aiding users in understanding their listening habits, and also in music recommendation systems. Alternatively, they are also useful for the workers behind these applications so they can understand their audiences better. With clearer visualizations, personalization of playlists could be enhanced so that these companies can cater to their user base and their interests more effectively. One example of this in action is Spotify's "Wrapped" feature, which collects all the listening data for each of its users throughout a given year, and presents them with various visualizations that tend to pique the user's interest, while also fueling Spotify's internal recommendation algorithm. Some information presented in Spotify's Wrapped feature includes the user's top genres, most frequently listened songs, overall energy of songs, and other unique features only available to active users of the application. The visual API of the feature is integral to its success, as its flashy colors and clean format are simple for users to digest, and unquestionable in their true meanings. With more ambiguous visualizations of Spotify's internal data, a user may feel confused by what is presented, and therefore lose interest. However, because the data is cleanly presented, there is a benefit to both the user and the business.

VI. Conclusion

The recommendation system crafted for this project is useful in creating lists of songs similar to certain songs already included in the dataset, and could be modified to create a list of songs based on the listening history of a particular user. The system operates under certain preconceptions, and includes a fair share of limitations that could improve the system if addressed. The system does not take into account the recency of listens to a particular song, nor does it weight the frequency of a particular song. With more weight attributed to songs that have been frequently played by a user within the study, that song could become more recommendable to users with a similar listening history. Any song that has ever been listened to by a user has been weighted equally, which should be rectified in any legitimate practice of this system. Additionally, there are machine limitations to how quickly the csv files can be processed, and the inclusion of 100 users, while extensive enough for the study's purposes, is ultimately a negligible subsection of a website's entire user base. There needs to be a tradeoff between the amount of users analyzed for the study, and the number of songs included in the Jaccard indexing, and several experiments changing these two variables could yield a more optimal comparison between the two.

This recommendation system is useful in checking for currently similar songs to any particular song in the dataset, and also checking how a particular user scores within the four attributes of Last.fm's internal data. With this data tracked, it is plausible for a user to land themselves somewhere within the cross-section of these four attributes based on the songs they most frequently revisit. A robust recommendation system considers several factors, such as frequency and recency of listening, a preventative measure against popularity biases, and priorities on both the user and track level. With a strong recommendation system, and tactically robust visualizations to accompany it, music can be more correctly attributed to users of that system.

References

Avidon, E. (2019). *Music Data Visualization Comes to Life in Tableau Collection.* TechTarget. https://www.techtarget.com/searchbusinessanalytics/news/252465871/Music-data-visualization-comes-to-life-in-Tableau-collection

Bedre, R. (2021). *Principal component analysis (PCA) and visualization using Python (Detailed guide with example).* https://www.reneshbedre.com/blog/principal-component-analysis.html

Bode, A. (2021). *Visualizing Spotify Data with Python and Tableau.* Towards Data Science. https://towardsdatascience.com/visualizing-spotify-data-with-python-tableau-687f2f528cdd

Chang, E. (2021). *Building a Song Recommendation System with Spotify.* Towards Data Science. https://towardsdatascience.com/part-iii-building-a-song-recommendation-system-with-spotify-cf76b52705e7

Ekstrand, M., Tian, M., Azpiazu I.M., Ekstrand J., Anuyah O., McNeill D., Pera M.S. (2018). *All The Cool Kids, How Do They Fit In? Popularity and Demographic Biases in Recommender Evaluation and Effectiveness.* Proceedings of Machine Learning Research. http://proceedings.mlr.press/v81/ekstrand18b/ekstrand18b.pdf

Galant, S. (2020). *Data Spotlight: How Spotify Wrapped Makes Music Data Feel Personable.* Springboard. https://www.springboard.com/blog/data-science/spotify-data-insights/

Gustafsen, A. (2022). *Principal Component Analysis.* Towards Data Science. https://towardsdatascience.com/principal-component-analysis-fbce2a22c6e0

Jaadi, Z. (2022). *A Step-by-Step Explanation of Principal Component Analysis (PCA).* BuiltIn. https://builtin.com/data-science/step-step-explanation-principal-component-analysis

Jena, A. (2022). *Music Recommendation System - Revolutionizing the Music Streaming Industry.* Muvi. https://www.muvi.com/blogs/music-recommendation-system

Kotzias, D., Lichman, M., Smyth, P. (2018). *Predicting Consumption Patterns with Repeated and Novel Events.* IEEE Transactions on Knowledge and Data Engineering. PP. 1-1. 10.1109/TKDE.2018.2832132.

Langer, T. (2008). *Music Information Retrieval & Visualization*, http://www.mmi.ifi.lmu.de/lehre/ws0809/hs/docs/langer.pdf.

Lever, J., Krzywinski, M., Altman, N. (2017). *Principal component analysis.* Nat Methods. https://doi.org/10.1038/nmeth.4346

Liang, Y., Willemsen, M.C. (2023). *Promoting Music Exploration through Personalized Nudging in a Genre Exploration Recommender.* International Journal of Human-Computer Interaction. *39*(7), 1495-1518. https://doi.org/10.1080/10447318.2022.2108060

Melchiorre, A., Zangerle, E., Schedl, M. (2020). *Personality Bias of Music Recommendation Algorithms.* Association for Computing Machinery. https://doi.org/10.1145/3383313.3412223

Mixson, E. (2021). *How Spotify Uses Data to Keep You Listening.* Data Analytics & Network. https://www.aidataanalytics.network/data-monetization/articles/data-visualization-monetization-and-personalization-spotify

Nagpal, M. (2023). *How Spotify uses Machine Learning?* ProjectPro. https://www.projectpro.io/article/how-spotify-uses-machine-learning/687

Oza, H. (2021) *How Spotify Is Using Big Data for Better Customer Experience.* HData Systems. https://www.hdatasystems.com/blog/how-spotify-is-using-big-data.

Pastukhov, D. (2022). *Inside Spotify's Recommender System: A Complete Guide to Spotify Recommendation Algorithms.* Music Tomorrow. https://www.music-tomorrow.com/blog/how-spotify-recommendation-system-works-a-complete-guide-2022

Qiu, H., Jia, X. (2022). *Western Music History Recommendation System Based on Internet-of-Things Data Analysis Technology.* Mobile Information Systems. https://doi.org/10.1155/2022/8920599

Reevesman, A. (2022). *Spotify Wrapped: Data Visualization and Machine Learning on Your Top Songs.* Towards Data Science. https://towardsdatascience.com/spotify-wrapped-data-visualization-and-machine-learning-on-your-top-songs-1d3f837a9b27.

Ripenko, S., Hasiuk, N. (2022). *Music recommendation system: all you need to know.* Tech Guide. https://www.eliftech.com/insights/all-you-need-to-know-about-a-music-recommendation-system-with-a-step-by-step-guide-to-creating-it/

Soman, A. (2020). *Exploring the Tale of Music Through Data Visualization.* Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/12/exploring-the-tale-of-music-through-data-visualization/

Tang, M., Yang, M. (2017). *Evaluating Music Discovery Tools on Spotify: The Role of User Preference Characteristics.* Journal of Library and Information Studies 15:1 pp.1-16. doi: 10.6182/jlis.2017.15(1).001

Ter Bogt, T.F.M., Mulder, J., Raaijmakers Q., Gabhainn, S. (2010). *Moved by music: A typology of music listeners.* Psychology of Music. doi:10.1177/0305735610370223

Thakur, R. K. (2021). *Spotify Data Visualization and Analysis Using Python.* Geek Culture. https://medium.com/geekculture/spotify-data-visualization-and-analysis-using-python-4af81c5531a7

Tsukuda, K., Goto, M. (2020). *Explainable Recommendation for Repeat Consumption.* Association for Computing Machinery. https://doi.org/10.1145/3383313.3412230

Tufte, E. R. (2018). *The visual display of quantitative information*. Graphics Press.

Wundervald, B. (2021). *Cluster-based quotas for fairness improvements in music recommendation systems.* International Journal of Multimedia Information Retrieval. doi: 10.1007/s13735-020-00203-0

Yang, Y., Liu, J. (2013). *Quantitative Study of Music Listening Behavior in a Social and Affective Context.* IEEE Transactions on Multimedia. doi: 10.1109/TMM.2013.2265078