Midterm 4: Spotify Data Analysis

Preston Yoshino, Joe Chanis, Joshua Chung

Our group has created a data exploration tool that allows users to explore Spotify data through a clean and intuitive interface, allowing us to display complex concepts seamlessly. It starts with our two datasets from Kaggle. The first of which is a dataset containing song information. This table contains basic information like name, release date, artist, popularity, etc. It also has some more niche features like energy, tempo, danceability, etc. Our second dataset is about user behavior, which gives us information like if the user skipped the song, how long they listened to it for, how did the song start (autoplay, playlist, search, etc). These two datasets were then crossjoined by song.

The next part of our data cleaning process was creating summary statistics for each of our songs. Since our primary interest lies at the song level, not necessarily at the individual user level. Thus, we grouped our data by song to create new features like skip percentage and average time listening.

Moving into the R-Shiny app we highlight 4 main features: A page allowing users to investigate the song components that influence skip percentage, a page for investigating user engagement by genre, a page that models genre popularity over time, and a page that allows users to inspect principle components of numeric predictor combinations and the corresponding clusters.

In our first feature we utilize a generalized linear model backed by a quasibinomial distribution. In our research, we found that GLMs are great for percentage outcomes as they can handle non-normal variances, and provide a link function to keep our responses bounded within 0 and 1. We also opt for a quasibinomial distribution as our background research suggested it's used commonly for percentage data that's derived from binary outcomes, and providing more flexibility in modeling variance. Users are able to see the summary output of our fitted GLM model. They can also view plots of the individual variables they selected against skip percentage.

Our second and third features are fairly straight forward. The second, allows users to select groups of genres and see the summary statistics of skip percentage and playing time percentage per genre. The summary statistics are delivered visually through a box plot. The third feature then gives the user the option to choose a genre and inspect how its popularity has changed over time. The first panel of this page displays a linear regression model that plots time against popularity for data where the genre is of the user's selection. The second page is then a visual representation of the data points and our fitted line.

Our last feature covers PCA and clustering for the data related to the components of a song. Users are able to select different components and see the principal components that are computed based on their selections. You can also see the different loadings for each principal component. This is paired with a scree plot to see the explained variance by each principal component and the recommended number of principal components. On the next tab the user is then able to take the top two principal components and perform k-means clustering. Clusters are

displayed visually for the user. They are also able to manipulate the amount of clusters in the model.

The difficulty rating I will assign to this project is an A. Our data cleaning was fairly extensive as we took two datasets and used a join to combine them, while also incorporating various dplyr and lubridate functions to transform and manipulate our data. However, the driving force for our A-level assessment is our R-Shiny app. We go beyond the recommended 3 data-exploration features, providing four distinct ways to explore our data, with many of our features being multi-dimensional, such that one might argue that our sub-features could even be their own page. We combine simple analysis tools such as features 2 and 3, with more complex and nuanced tools like feature 1 and feature 4 which go beyond the areas of data science covered in class. Our first feature required the research of various modeling algorithms and distributions to choose a proper model for our data. It also required us to learn how to interpret the summary of our model and translate this into intuitive explanations for our user. Our fourth feature also displays A-level difficulty, as even though the concepts within the page were covered in class, these were widely regarded as the most difficult, but we incorporate them to allow the user to see how various components of a song can be simplified into a handful of components. We also provide a visual way for users to see how songs are structurally similar through a plot produced by K-means clustering. Lastly, our app has an intuitive interface and consistent layout. We see a traditional navigation bar for switching between pages, tabs on each page for switching between sub-features, then a side panel containing all user input fields. Embedded in each of our components is also text to help the user understand our app and what we are showing them. Behind the scenes, our code is neatly formatted and well documented.

Our app provides users a clean and intuitive way to interact with the spotify data and receive both written and visual analysis. We cover various data science concepts, from both in class and out of class, that allow the users to explore the data from multiple different angles. If time were unlimited, I would love to go through our server code, and better implement reactivity so that the rerendering of plots and summaries were computationally optimal. The project could also be further improved through more complex modeling of listening data, through different algorithms. Potentially using machine learning to suggest song compositions that will maximize user engagement. Though, the necessary knowledge for this goes far beyond the extent of this project and would most likely require further coursework.