Basic Info

The title of our project is SpotifyMDb.

Our team:

Steffen Dreesen	C10630953	sdreese@clemson.edu
Nick Macris	C13084809	nmacris@clemson.edu
Jacob Madsen	C14645719	jmmadse@clemson.edu

Project repo: https://github.com/steffendreesen/SpotifyMBd

Overview and Motivation

Our project was motivated by our collective interest in music and film. Having interests in both of these areas, we naturally gravitated towards studying the intersection of these two topics: movie soundtracks. The soundtrack of a movie can have a significant impact on the viewer's reception of the film, which in turn determines its financial success. For this reason, we believe our study has potential for genuinely valuable discovery. We also knew of publicly available databases and subsequent APIs (IMDb and Spotify) which could provide us with a large and interesting dataset. For all the above reasons, we decided that film soundtracks would be a promising dataset to study.

Related Work

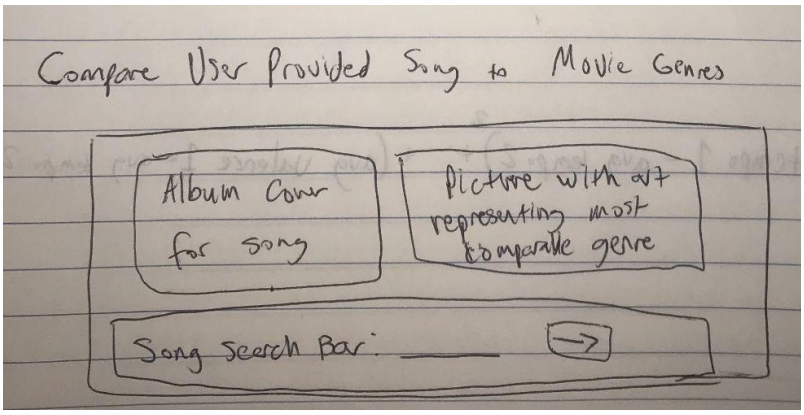
One of our major inspirations for this project was IMDb, which is a well known web based database containing a large amount of information about movies, film, homevideos, and other platforms as well. As stated above, two of our common interests as a group are music and film, so a resource like IMDb is very useful, but we noticed there was not a ton of information on IMDb related to the soundtracks. For example, if you were to type a movie into the search bar, when you go to look for information on the soundtrack for that movie, the only information you

would find are song name, song writers, performers, and a record label. We felt there was a lot more compelling information to soundtracks than just this basic information, so this was our main inspiration in deciding to create visualizations exploring the relationship between film genre and what types of music is selected for soundtracks.

Questions

Overall, our main objective in completing this project was to study the relationship between film genre and soundtrack selection. More specifically, we wanted to create meaningful visualizations that allow the viewer to easily observe relationships and trends between movie genre and song selection. For example, how do the soundtracks of Comedies differ from Horror films? We wanted to create a set of visualizations that clearly displays the soundtrack differences in these two movie genres.

Finally, we wanted to provide the user an interactive experience that allows them to study the relationship between film genre and soundtrack selection in real time. Initially, our main objective in this regard was a film genre recommendation service. This service would take a user-submitted song and generate a visualization summarizing which film genres are most likely to have a soundtrack containing that song. Our goal for this objective was to provide the user with a fun mechanism to study the relationship between a film's genre and its soundtrack.



Initial Sketch of Genre Recommendation Service

However, we felt that this design decision would not allow for a visual enough user experience, as the user would be required to paste in their own songs, etc. For this reason, and several more listed below in the Design Evolution section, we switched our focus to visually conveying the similarity, or dissimilarity, between all the film genres and their soundtracks. Essentially, we shifted our focus with this objective to answering the following set of questions: how similar are the soundtracks of <Genre 1> movies and <Genre 2> movies? We felt that this was a more interesting question to approach as it could more easily interact with other visualizations, and perhaps give insight to the user as to which genres and soundtrack attributes to study.

Data

Movie titles, genres and soundtracks were collected through the IMDb API. We used the Spotify API to collect attributes associated with the songs in each movie's soundtrack. These attributes include the following: track title, artist, valence, tempo, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, and liveness. A description of these attributes is provided below:

valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
danceability	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
energy	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel

	fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
instrumentalness	Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
liveness	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
loudness	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.
speechiness	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
tempo	The overall estimated tempo of a track in beats per minute (BPM). In musical

	terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

The data was scraped using a Python script utilizing the IMDb and Spotify API and exported into a csv file for further processing. Our dataset is 8800KB large and contains all of the above information for 12,291 films.

Fortunately, we did not have to do any substantial data cleanup for our dataset. Much of the common data cleaning processes were implemented programmatically in our Python script. We did, however, reduce the size of our dataset by filtering out any movies with only one song available on Spotify. We felt that these points were too information-sparse and did not accurately represent the true average energy of the soundtrack, average valence of the soundtrack, etc. After applying this filter, we ended up with 6437 data points.

We also made use of several derived quantities from our dataset. For each movie genre in our database, we computed the average of each attribute available; for example, average tempo, average danceability, etc. Note that, for each film, the value for each attribute is an average of all the songs in that movie's soundtrack. We made use of these derived averages to compute similarity scores between film genre soundtracks (described more below). These averages and similarity scores were computed within the Excel file using Visual Basic.

Exploratory Data Analysis

Initially, we used very simple visualizations to study our dataset. By using bar charts, we were able to plot a film genre on the x axis, and then use the y axis as a scale for whichever attribute was selected by the user. One major insight we gathered was that there were definitely trends worth looking at in the dataset, but sometimes they were hard to find, and it would be entirely unintuitive to expect a user to find these trends on their own, so we knew we needed to have some sort of way to help the user spot these trends as well. In the long run, we believe this helped us come up with the idea of using a film genre similarity matrix, where the user is able to use color to distinguish how similar two soundtracks are, so they can filter the scatterplot to more

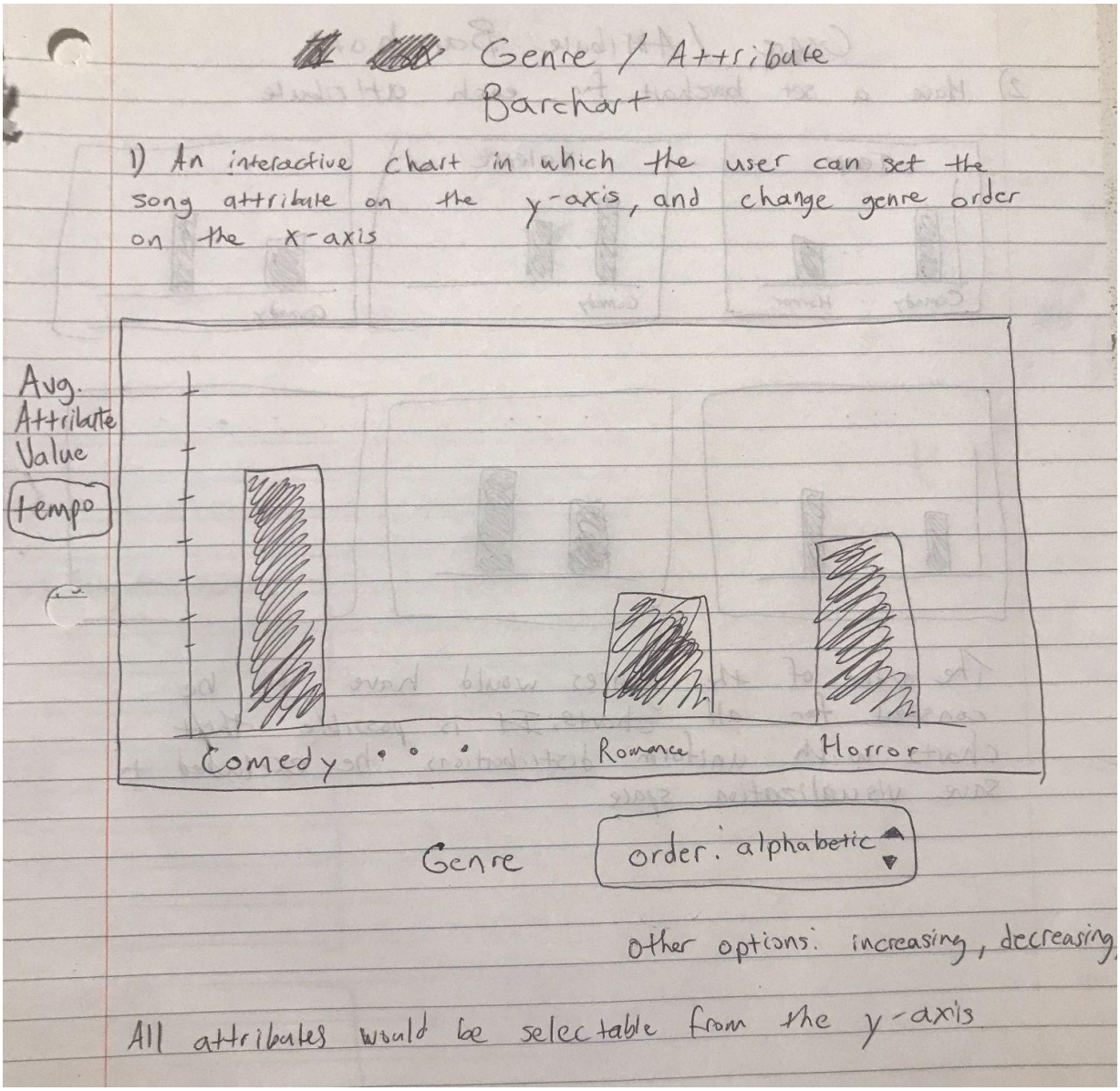
easily view this visualization. This insight helped inform our design because it made us realize that when we are dealing with a dataset as large as ours, we need to make it as easy as possible for the user to spot these trends within the dataset, so that they have a pleasant experience with our product.

Design Evolution

Our initial design goal was to provide three visualizations:

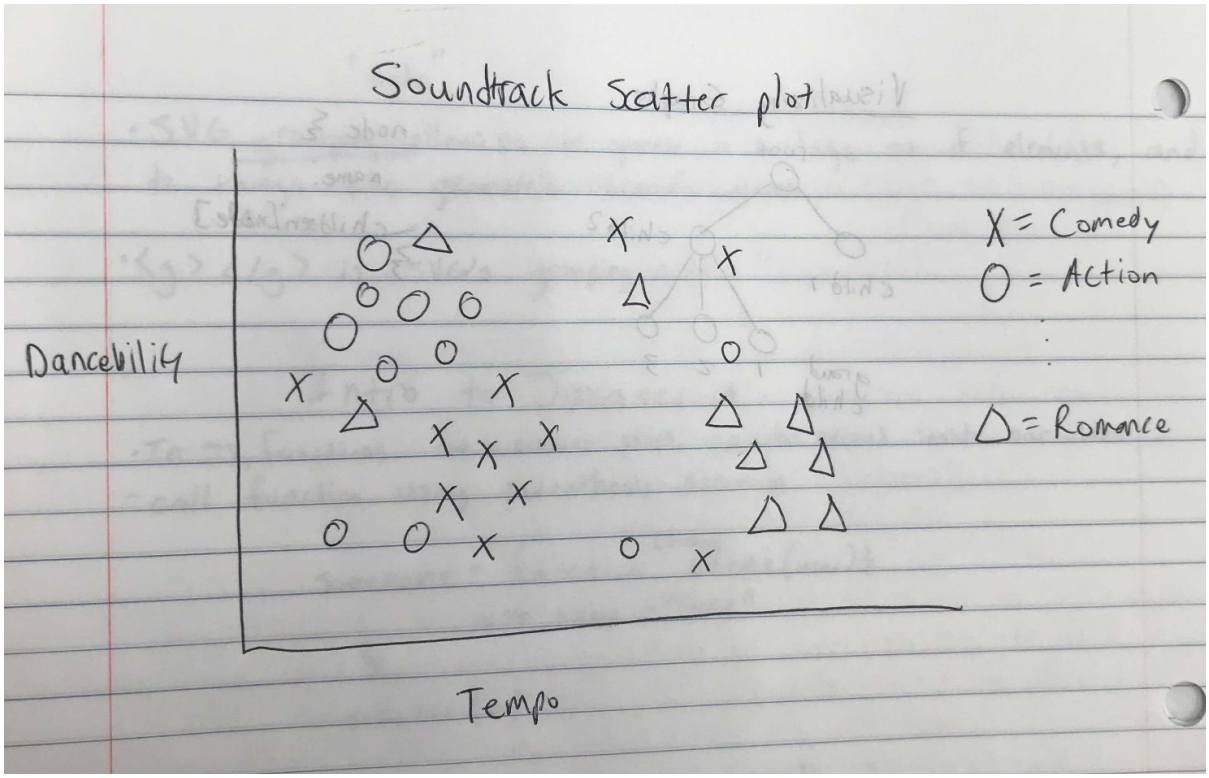
1. Interactive Film Genre Soundtrack Attribute Analyzer
    a. User selects song attribute
2. Film Genre Soundtrack Similarity Matrix
3. Song to Film Genre Fitness Chart
    a. User provides song

Our initial plan for the first visualization was an interactive bar chart. The user would be provided a dropdown menu allowing them to select a song attribute: valence, tempo, energy, etc. The x-axis would be labeled with the film genre, and the y-axis would provide a scale for the selected attribute. The user would then be able to switch between the provided song attributes and observe visually how the typical soundtrack for each film genre makes use of the selected song attribute. Our goal was for the difference in heights of each bar to allow the user to quickly discern whether film <Genre 1> typically contains more happy songs in its soundtrack than <Genre 2>, for example.



Initial Sketch of Interactive Film Genre Soundtrack Attribute Analyzer

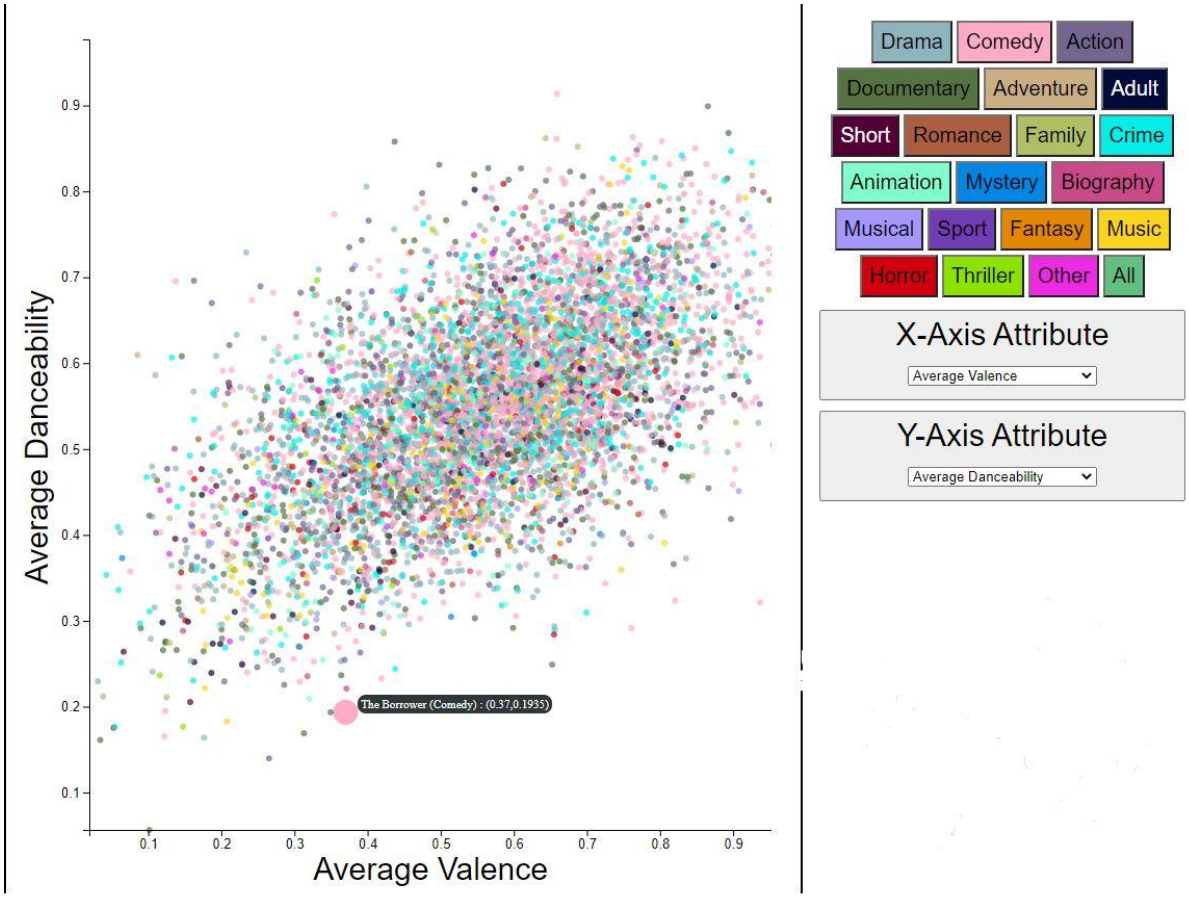
We soon realized that this design was a poor visual representation of our dataset as a whole. We felt that showing only one genre of movie at a time was too restrictive and prevented the user from identifying clustering/trends across the entire dataset. For this reason, we decided to change the visualization to a scatter plot. The new visualization would plot every data point on the graph based on two user-selected attributes. Further, the shape of each point would encode the genre of the movie, allowing the user to identify trends across genres.



Initial Scatter Plot Sketch

Initial Scatter Plot Implementation

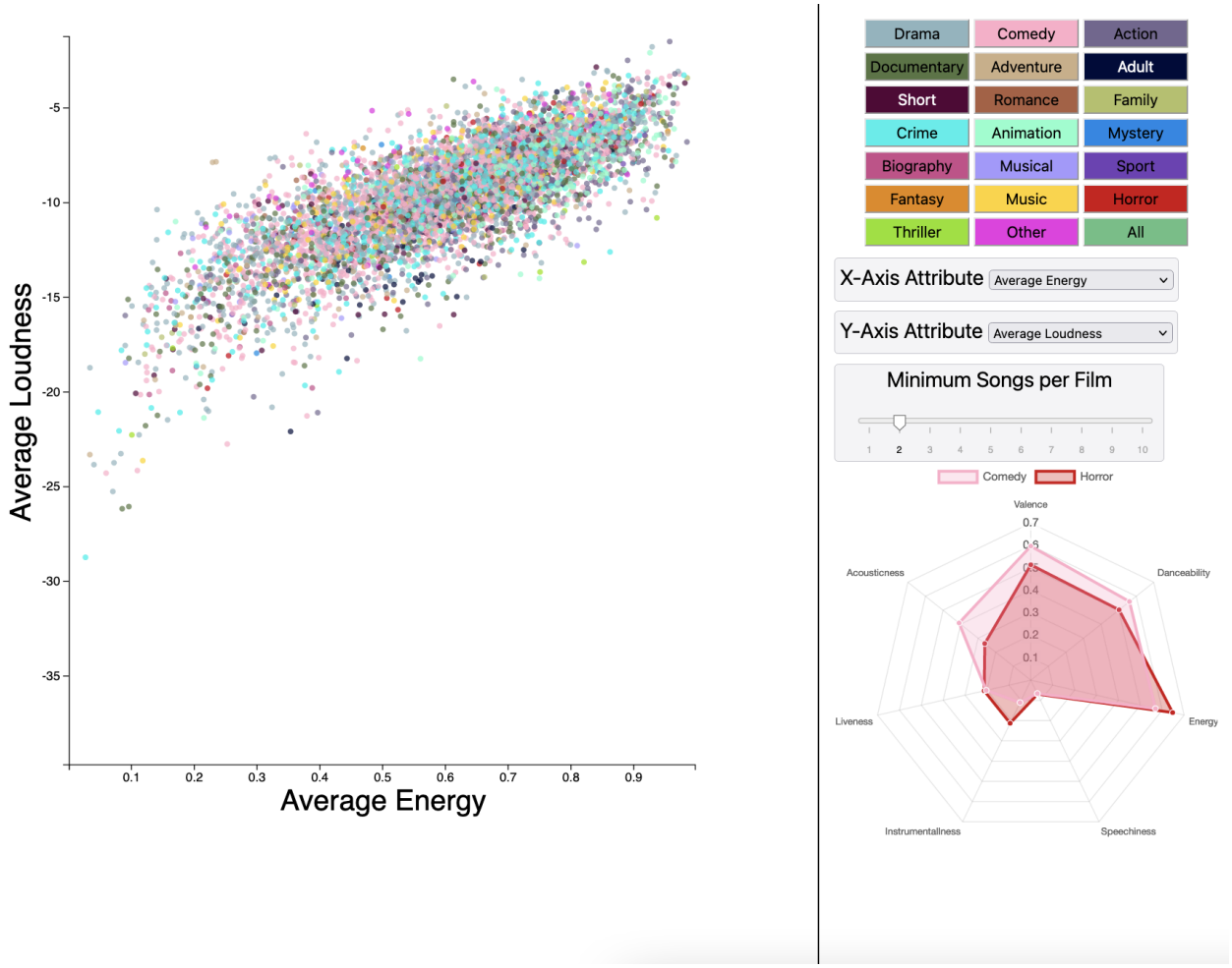
Overall we were much happier with the alternative visualization, especially how it displayed the entire dataset all at once. Since we were working with 20+ movie genres, we decided to make use of color instead of shape to visually encode the genre of each movie. Unfortunately, due to the large number of genres, we were finding it difficult to create a color map which clearly distinguished each genre from each other. This problem would later be resolved during the development of our second visualization (detailed below) when we decided to remove several genres with limited numbers of movies.



Updated Scatter Plot Implementation

In order to provide the user with more tools to interact with the visualization, we added genre selection buttons and attribute selectors. The genre selection buttons hide and show points of that genre when they are clicked. We were happy with the consistency of colors between the points and their associated genre selection button, however we did not like the actual colors we made use of to do this. Further, the attribute selectors provided the user a dropdown that allowed the

attribute associated with each axis to be changed. This would successfully change the scale of the axis if needed, and reposition all the points according to the updated attribute. We were very happy with this result, however we were slightly disappointed that there were relatively few obvious trends resulting from the attribute combinations we tried. This problem would later lead us to creating an interaction between our second visualization (genre similarity matrix) and the main scatter plot. This interaction is detailed later in the Design Evolution section.

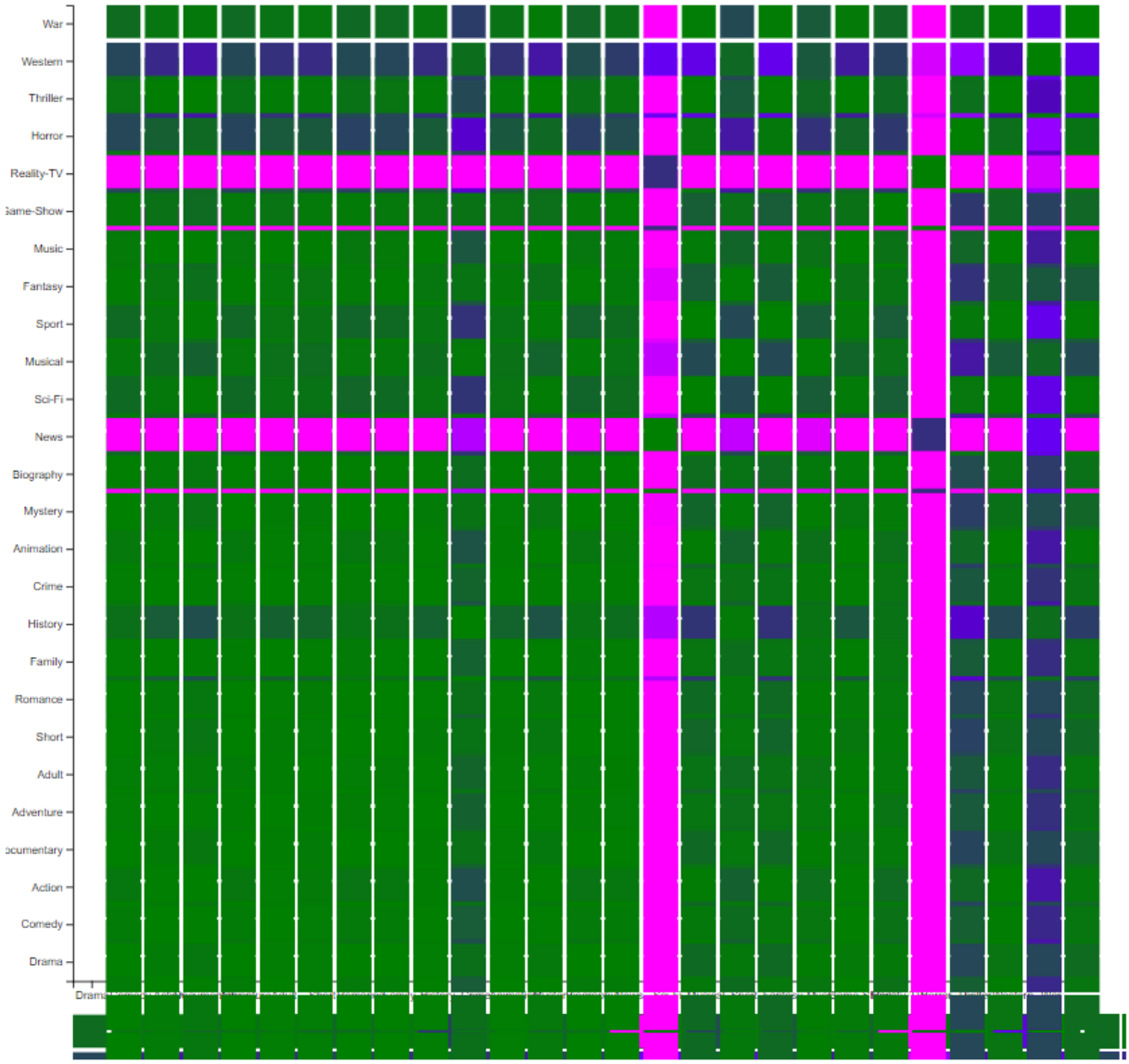


Final Scatter Plot Implementation

Our final scatter plot design includes an aesthetically improved “control section” on the right which allows for easy user-selected alterations to the scatter plot. We also implemented a hovering feature that displays the name of the movie and the corresponding (x,y) attribute values when each dot is hovered over. Additionally, we made use of the website iWantHue to generate 20 maximally distinct colors that we encoded to the genres. Finally, we added a slider below the

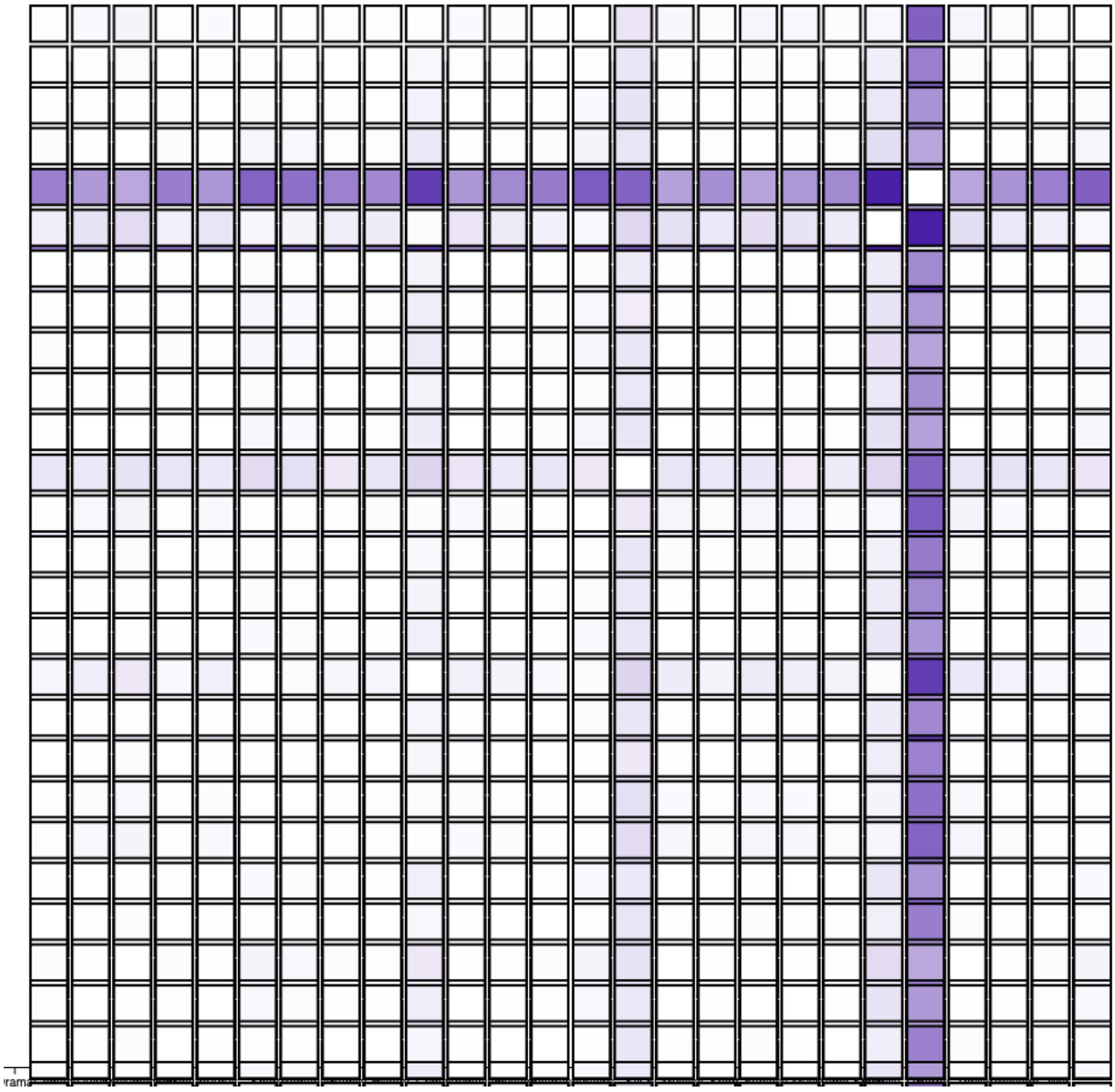
axis selectors that allows the user to control the minimum number of songs required for each film. For example, if the user selects one track, several more points will appear on the scatter plot. Overall, we are quite happy with the final scatter plot design and feel that it provides the user with plenty of interactive options allowing them to explore the dataset.

In the second visualization, our intention was to create a similarity matrix that would compare each film genre with each other based on a similarity function. We defined the similarity function in the following manner: dissimilarity = (avg_valence_genre1 - avg_valence_genre2)^2 + (avg_tempo_genre_1 + avg_tempo_genre2)^2 + … and the smaller the value of dissimilarity the more closely related the soundtracks are between the film genres. The color of each cell in the matrix would then be determined by this similarity function. Our goal for the similarity matrix and use of color was to allow viewers to quickly spatially and comparatively recognize which film genres typically make use of similar soundtracks and which do not.



Initial Similarity Matrix Implementation

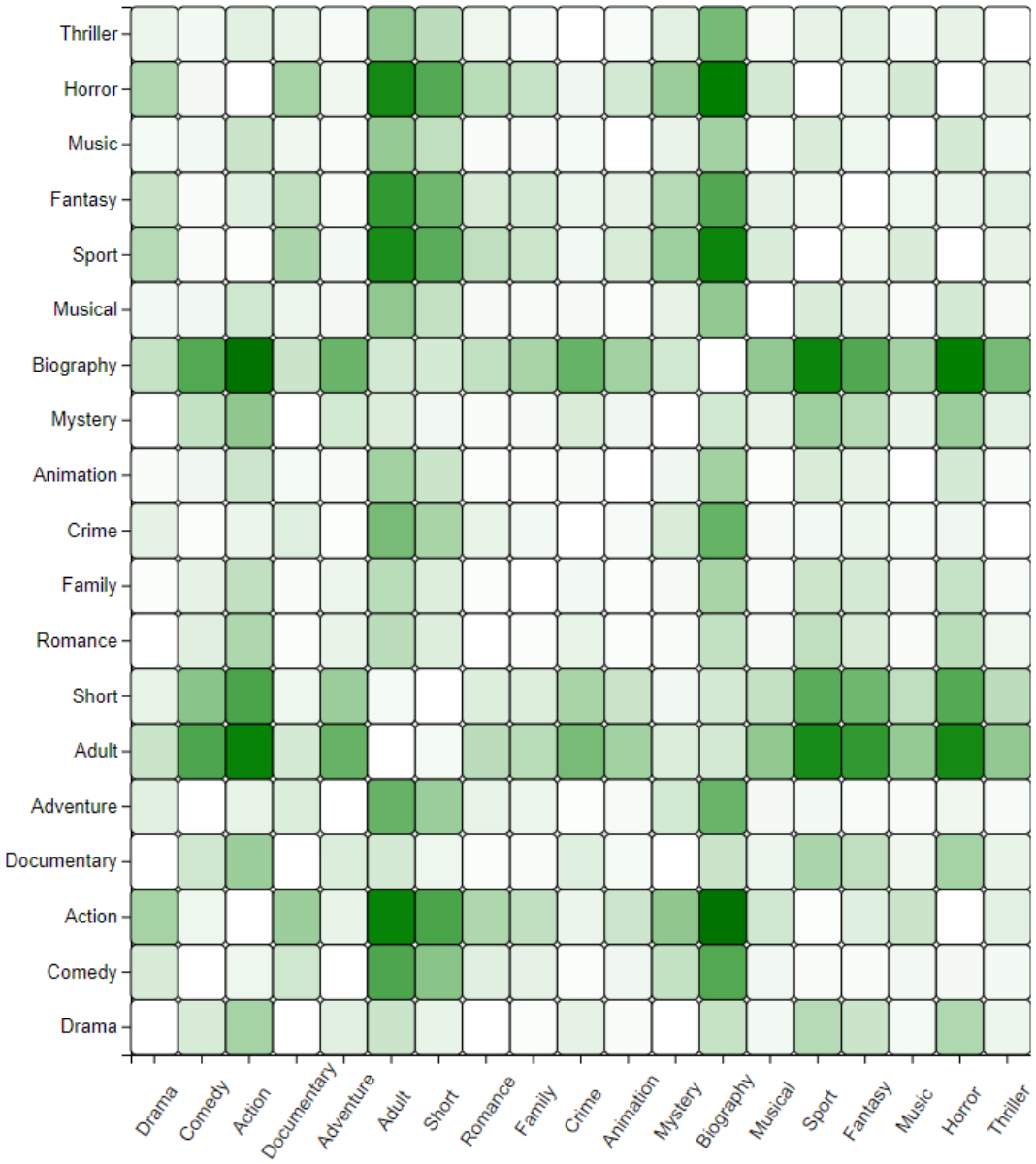
Our initial similarity matrix had several issues. To begin with, we felt the use of multiple colors was unintuitive. For example, there was no way for the user to know which color indicated greater similarity and which indicated less similarity. To rectify this, we decided to use only one color and vary the saturation depending on similarity.



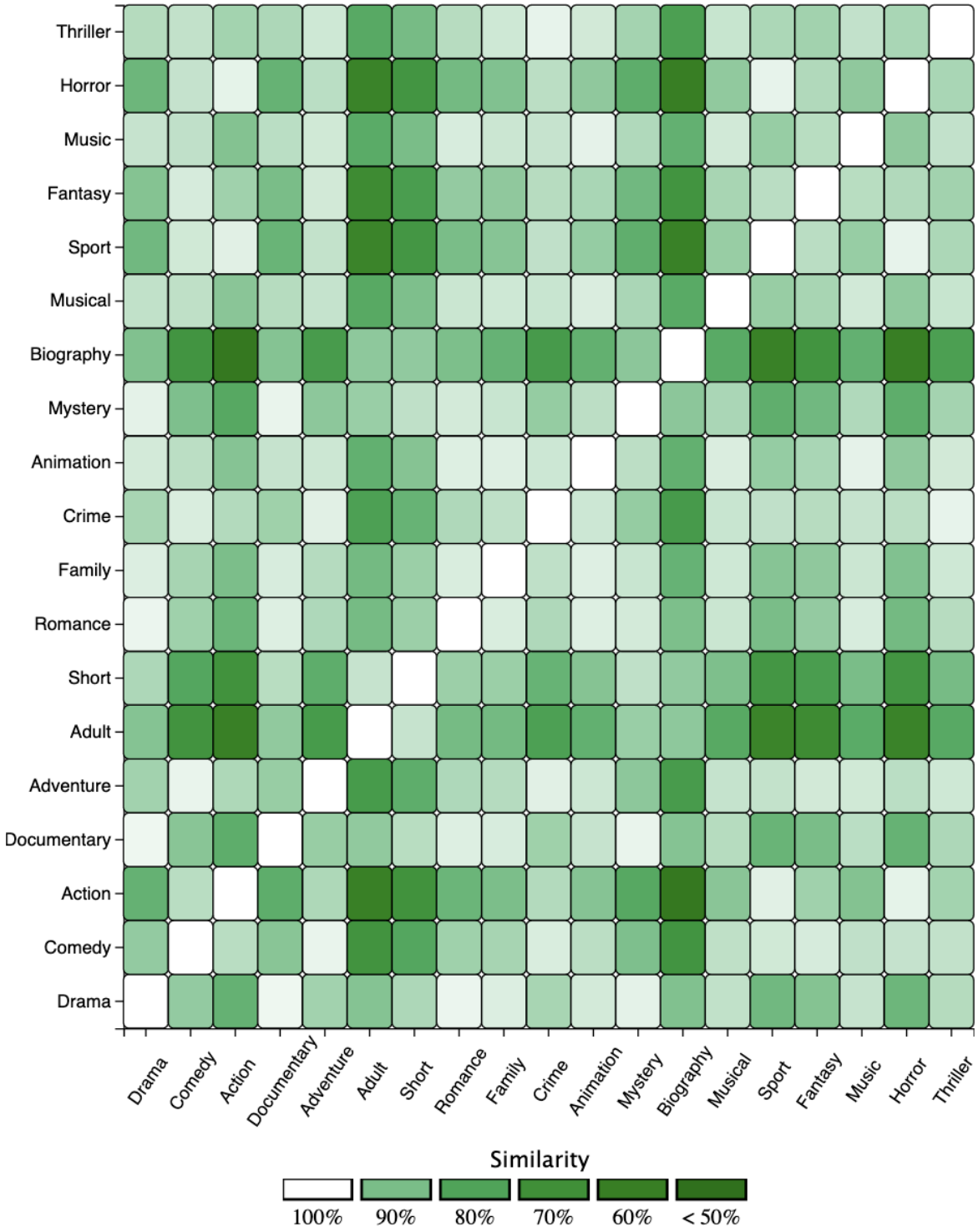
Improved Similarity Matrix

This visualization still had some bound issues with the cells running off the axes, but we were happy with the change to a single color. Unfortunately, the matrix at this point was relatively uninteresting due to the large dissimilarity between the Reality-TV and Game-Show genres and the rest of the movie genres. After studying the dataset, we found that these two genres had fewer than 10 movies each, so we felt the best approach was to remove them entirely. To further justify this decision, we remark that the original intention of the project is to study the relationship between film genre and soundtrack selection; however, Reality-TV and

Game-Shows are not truly films to begin with. For that reason, we felt that removing them was the best option.



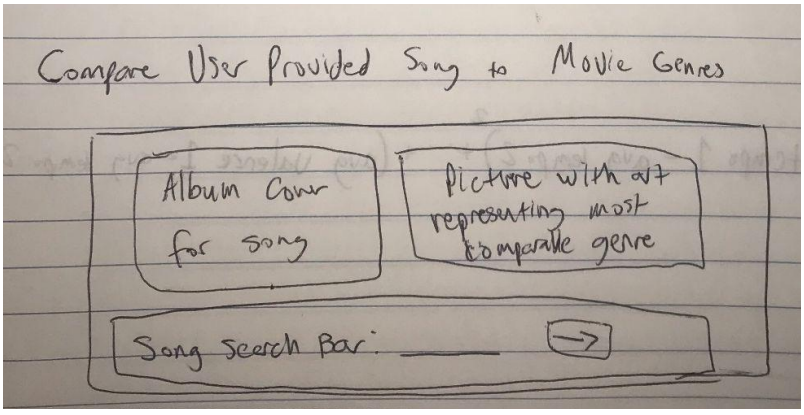
Improved Similarity Matrix



Final Similarity Matrix

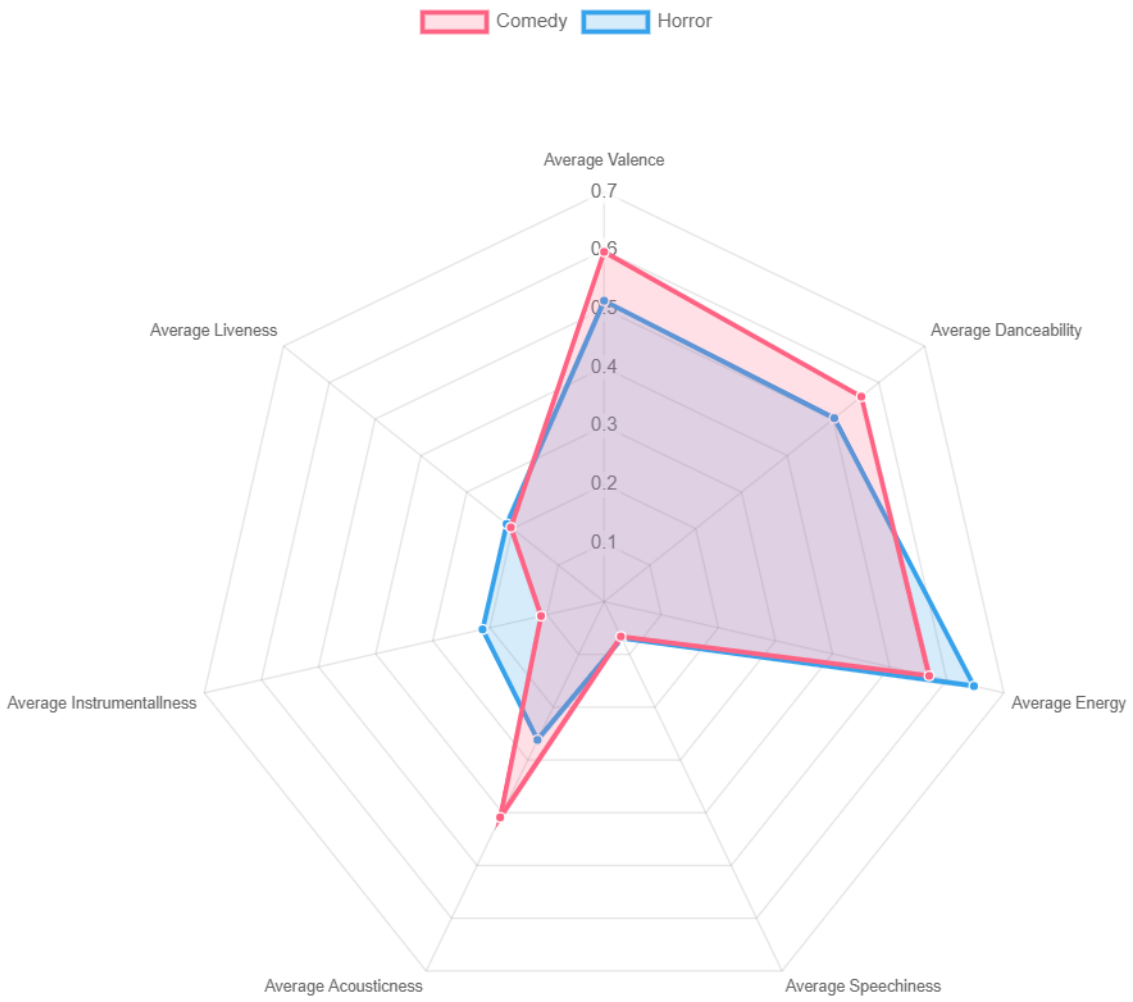
After removing a few sparse genres, changing the color scheme, and adding a legend, we arrived at our final similarity matrix design. We were much more satisfied with how clean the single color looked, and after removing a few genres the matrix is actually informative. For example, the cell marking the intersection of Biography movies and Action movies is quite saturated, and as you would expect the typical soundtracks of these two film genres are very different. At this point in the project, we realized that a standalone visualization to highlight the differences between the two selected genres' soundtracks would be extremely helpful, as this would provide insight to the user as to which attributes they should plot on the scatter plot axes. For this reason, we scrapped our initial design plan for our third visualization and shifted our focus to addressing this issue.

The original intent for the third visualization was a bar chart that would accept a user-provided Spotify song and generate a bar chart based on which film genre's typical soundtrack would be most likely to contain the user-provided song. The x-axis was to be labeled with each film genre, and the y-axis would provide a scale for the fitness function that we would define in the following manner: fitness = (input_song_valence - genre_valence)^2 + (input_song_tempo - genre_tempo)^2 + ... and the smaller the value of fitness the more likely the song would appear in that film genre's typical soundtrack. This fitness function would provide a fitness value for each film genre which would then determine the height of each bar in the bar chart. Our goal was for the difference in heights of each bar to allow the user to quickly discern whether "Back in Black" by AC/DC is more likely to appear in an Action film's typical soundtrack versus a Romance, for example.



Initial Sketch of Genre Recommendation Service

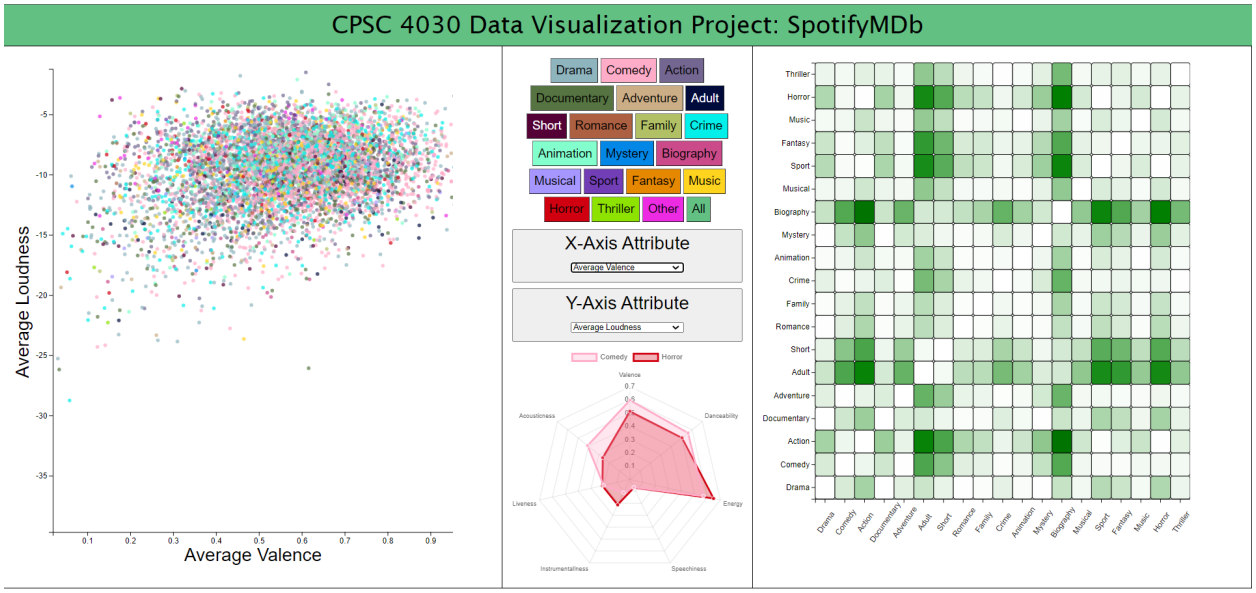
However, as stated above, we realized that a visualization that would portray the exact attributes where two movie genres' typical soundtracks differ would be much more valuable. For this reason, we decided to move forward with a radar visualization. The goal of this radar visualization was to visually highlight the key attributes that caused two genres' typical soundtracks to be dissimilar.



Radar Implementation

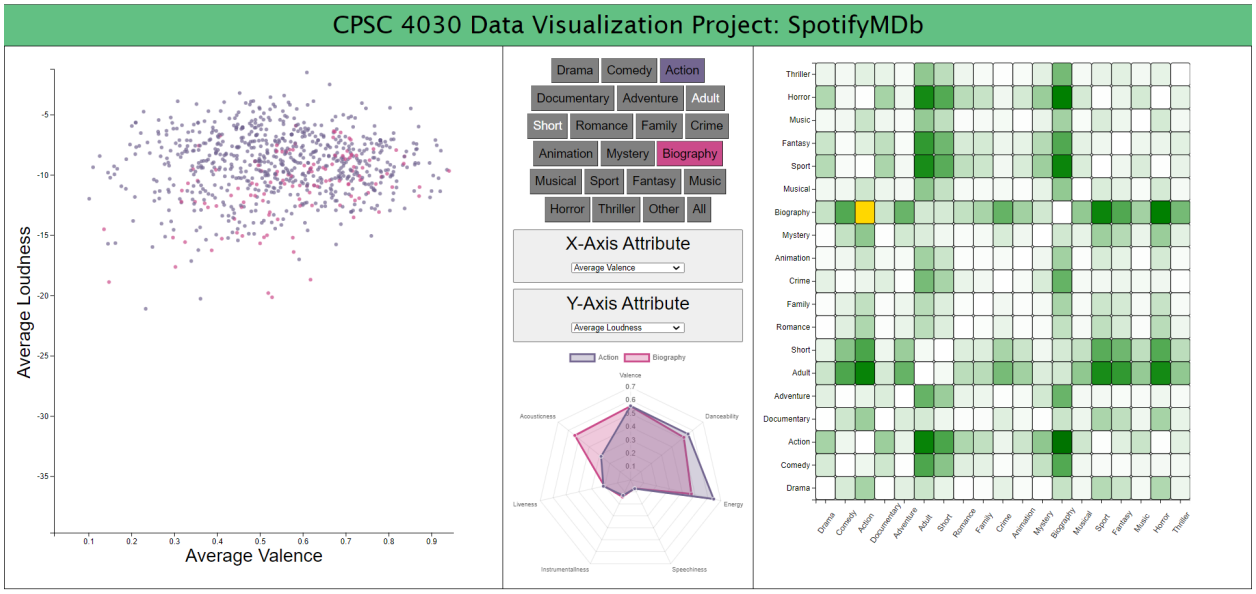
Surprisingly, our first implementation of the radar visualization was nearly exactly the same as we initially envisioned. For consistency, we associated the color of each genre in the radar with the same color used in the scatter plot visualization.

Implementation



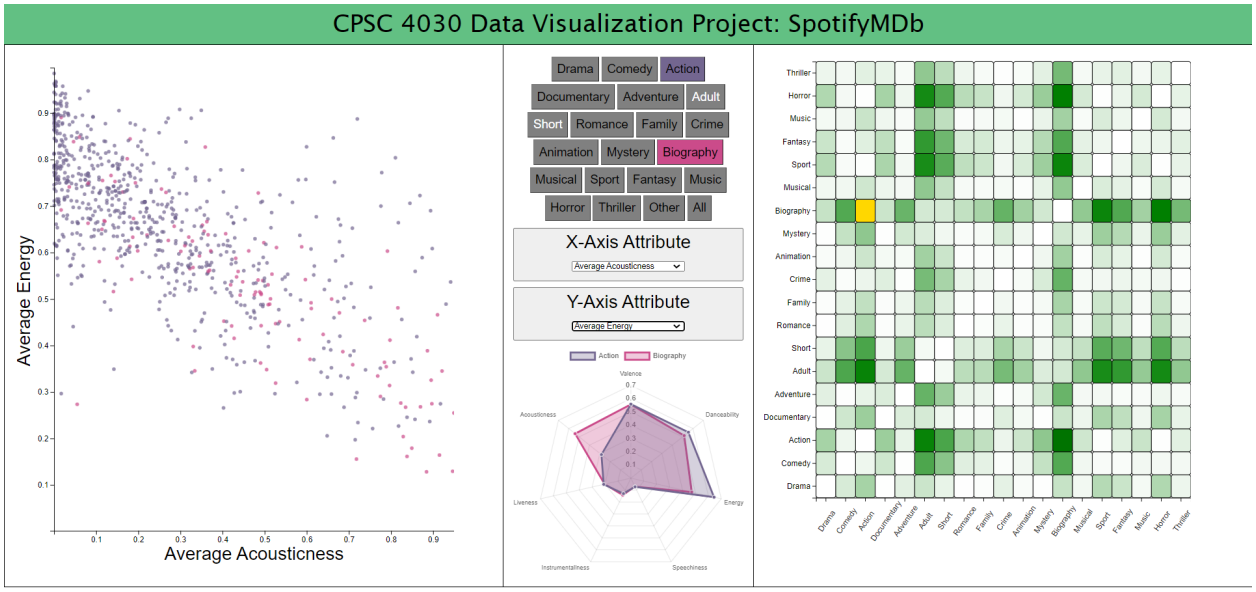
Final Site Layout

The intended use of our set of visualizations is for the user to begin with the similarity matrix. When the user finds a cell they determine to be interesting, i.e. a more saturated cell such as the intersection of Biography and Action, they can click on it to automatically adjust the other two visualizations.



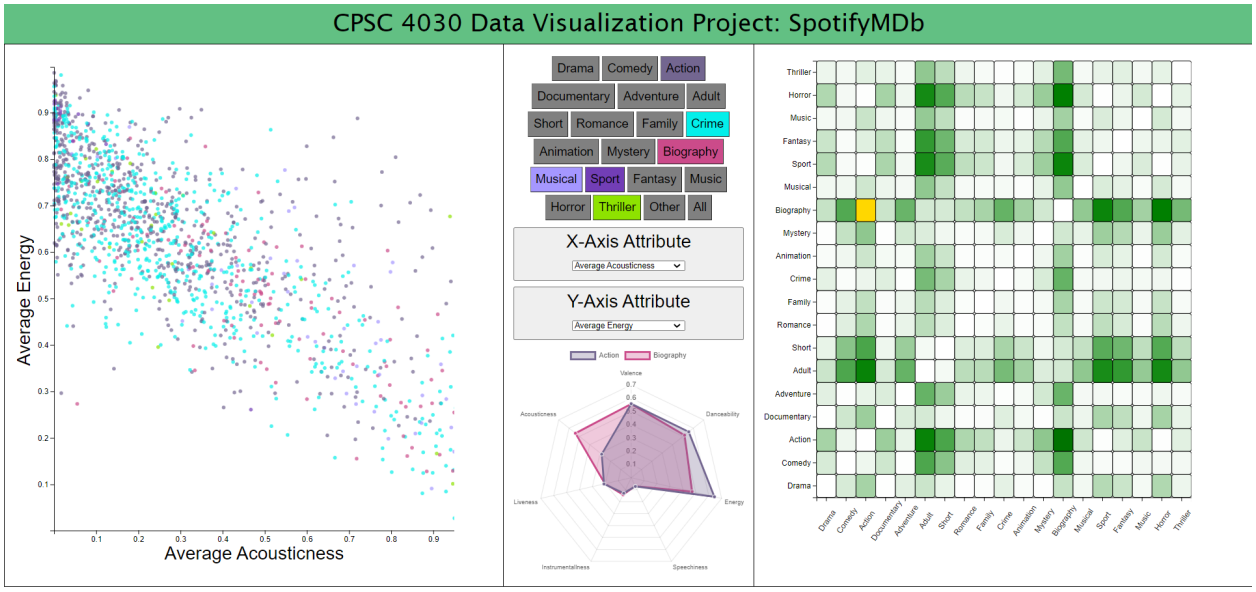
Choosing a Similarity Matrix Cell

When the user selects a cell, it is highlighted in gold in the matrix. Additionally, the radar in the center of the page updates to show the two selected genres--in this case, the average attributes of the Biography and Action genres' soundtracks are plotted. In the same way, the scatter plot updates to only display dots of the two selected genres. The radar allows the user to determine the exact attributes where the two genres' typical soundtracks differ, providing them insight into the attributes they may wish to plot on the axes of the scatter plot. For example, the radar indicates that the average soundtracks of Biography and Action films differ the most in acousticness and energy.



Updating Scatter Plot Axes

Following this new insight, the user can plot average accousticness and average energy on the scatter plot axes. Note the negative linear correlation between the two attributes, as well as the larger quantity of pink (Biography) dots in the lower right and higher quantity of purple (Action) dots in the top left. By following this flow of interaction, the user has discovered an interesting trend in the dataset with a visual representation to support their findings.



Adding/Removing Genres

To further explore their newly found trend, the user may add or remove movie soundtracks of different genres. In doing so, they may find that certain genres do or do not follow the same correlation. At any time, the user is free to select a different cell in the similarity matrix, add/remove genres in the scatter plot, as well as adjust which attributes are plotted on the axes.

Evaluation

One specific way the team learned about the data by using our visualizations is to see which specific attributes differ from soundtrack to soundtrack based on the genre of the film. Further, our visualizations also allows the user to see by how much each attribute differs from genre to genre, as well as from movie to movie within the same genre. This allowed us to learn even more about the data since it invites the user to add extra context by having a scale to attach the differences between the soundtracks to. Our main question for this project was how the genre of a film relates to the songs chosen to be in its soundtrack. We believe that we have answered this question throughout our visualizations, and the visualizations allow the user to see which specific attributes differ between a soundtrack for one movie genre, and a soundtrack for another movie genre. Overall, we are very happy with the result of this project, and we would only make slight modifications had we had additional time. Some of these slight modifications would involve

more HTML styling to have the website itself look a little more visually appealing. As we neared the completion of the project, we realized that the layout of our site could be improved.