Interactive Spotify

Joonho Ko
joonhok@mit.edu
Massachusetts Institute of Technology

ABSTRACT

In this paper, I introduce a tool, Interactive Spotify [1], developed as the final project for MIT's Spring 2020 class Interactive Data Visualization (6.894). This tool is designed to offer audio-visualizations of a subset of the user's personalized Spotify data, this subset being all playlists as well as the user's top songs for three different time periods. Using Interactive Spotify allows users to compare and find correlations in certain *audio features* for the music they listen to.

1 INTRODUCTION

Spotify is a music streaming platform that has seen widespread use due to its extensive library of over 50 million tracks [2] as well as being a powerful tool for discovering new music. Spotify allows users to create and add songs to playlists, while also providing curated playlists unique to each user. However, Spotify does not provide visualizations for user's playlists and songs, nor does it expose music preferences for a given user's data.

My project, Interactive Spotify (IS), aims to provide an interactive audio-visualization of the user's personal Spotify data by exposing certain aspects of Spotify's song data that are hidden in its main application, namely the *audio features* for a given song. Audio features are abstract categorical descriptions of songs, such as energy or danceability, that are expressed as a numeric value between 0.0 and 1.0. IS allows a user to concretely view these otherwise hidden audio features and compare and correlate them with the various playlists in the user's library. Although these audio features are not visible in the main application, they are available through Spotify's API, and Spotify does refer to these features in official blog posts about listening trends over time [3].

The project is an *audio*-visualization because it uses not just a visual mode of display, but an aural mode as well – users can listen to songs in real time as they interact with them in the visualization.

2 RELATED WORK

Currently, one of the most popular music-habit tracking websites is *last.fm* (https://last.fm), which, when registered with Spotify, keeps a record of the tally of all the songs that the user has listened to since registering. It also provides a profile page of mini-visualization modules providing top-level information such as number of different songs, artists, and albums played, top genres, and listening times during the day. *last.fm* is similar to IS in that it has visualizations of personal Spotify data, but differs from IS in the type of data that is visualized. Unfortunately, *last.fm* requires a paid subscription to view data older than a week.

3 METHODS

Since Interactive Spotify shows visualizations that are personalized for each user, a user needs to be able to authenticate with Spotify in the app in order to get the personalized data. This meant that the project needed to be run as a web app instead of a statically hosted website on GitHub Pages. I wrote the server using Node and Express, a JavaScript web app framework. To make IS publicly accessible, I deployed the server using Heroku.

Authentication to Spotify and accessing user data was done through calls to the Spotify API in the server. Spotify has an endpoint for getting the top tracks for a user, as well as endpoints for getting audio features for songs, getting top-level information (e.g. name, artist, album, track duration) for songs, and getting all playlists in the user's library. The actual visualization of the data was done with D3 (https://d3js.org).

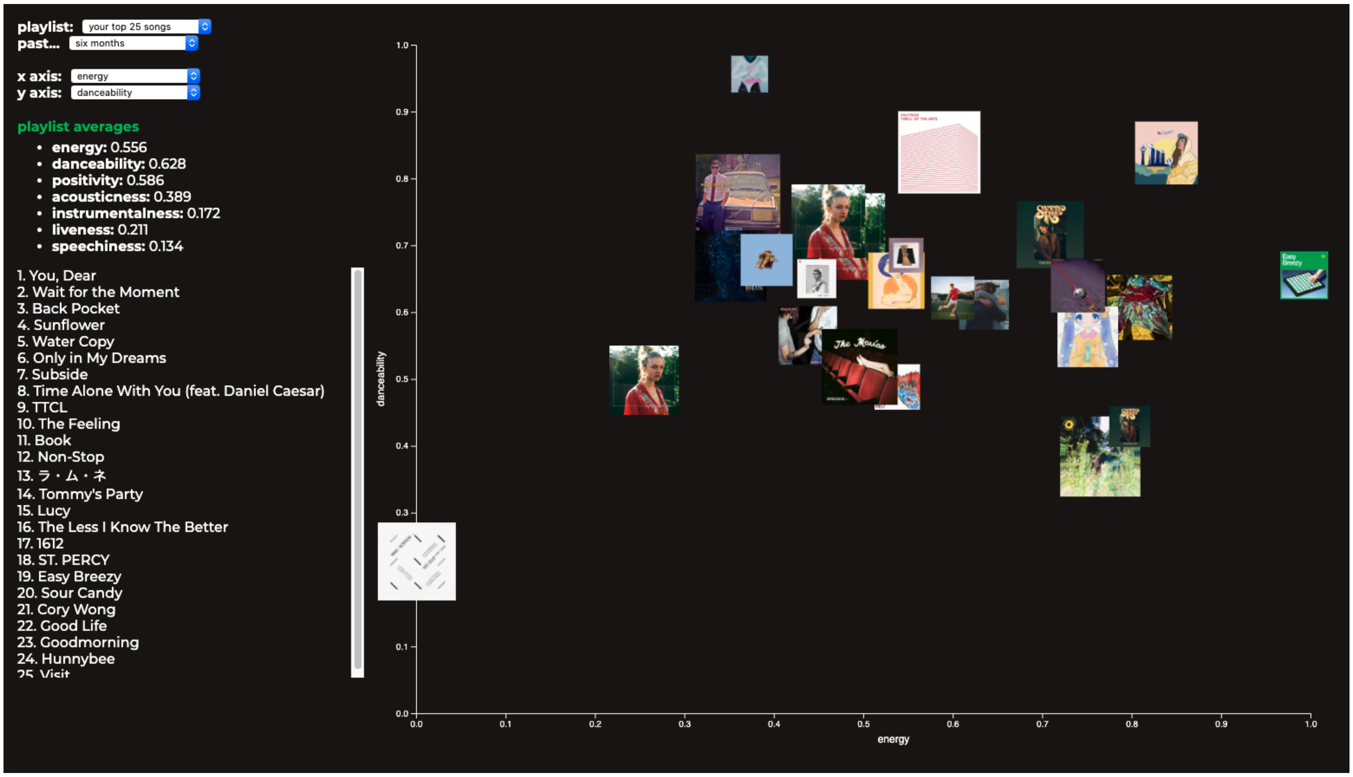
4 RESULTS

When the user first views their visualization (Figure 1), they are shown their top 25 songs in the last six months. This was chosen to be a neutral, universally interesting dataset. On the left, the user can choose to select any playlist in their current library, or view their top 25 songs for three different time periods – the last month, last six months, or the last several years. These time periods correspond with the three options for time ranges offered by the Spotify API.

The data itself is shown as a 2-dimensional scatterplot, with the album artworks for tracks as the scatterplot points. The axes of the plot, which are audio features, are configurable by the user on the left-hand side. IS shows seven different audio features, those being *energy*, *danceability*, *positivity*, *acousticness*, *instrumentalness*, *liveness*, and *speechiness*. To some degree the feature names are self-explanatory, but IS also has a landing page after authentication that have longer definitions for these terms. Not all audio features are created equal, and some are more meaningful than others – I chose energy and danceability to be the default starting axes out of pure personal opinion that those produced the most balanced visualizations, although positivity is another good choice.

The left-hand side also has a list of the songs in the currently selected playlist. When a user hovers over a song name in the list, several things occur: First, the song is highlighted in the list. Second, top-level song info such as artist name, album name, global popularity, and song length are shown in the lower-left corner. Third, the song is 'highlighted' in the plot itself, as the album cover gets slightly larger and all other album covers get greyed out. Lastly, an audio preview of the song begins to play, allowing users to actually *listen* for the audio features of the particular song that they are hovering over. These effects are removed when the user stops hovering. These effects also occur when a user hovers over an album cover on the plot itself.

, ,



playlist: your top 25 songs
past... six months

x axis: energy
y axis: danceability

playlist averages

- **energy:** 0.556
- **danceability:** 0.628
- **positivity:** 0.586
- **acousticness:** 0.389
- **instrumentalness:** 0.172
- **liveness:** 0.211
- **speechiness:** 0.134

1. You, Dear
2. Wait for the Moment
3. Back Pocket
4. Sunflower
5. Water Copy
6. Only in My Dreams
7. Subside
8. Time Alone With You (feat. Daniel Caesar)
9. TTCL
10. The Feeling
11. Book
12. Non-Stop
13. ラ・ム・ネ
14. Tommy's Party
15. Lucy
16. The Less I Know The Better
17. 1612
18. ST. PERCY
19. Easy Breezy
20. Sour Candy
21. Cory Wong
22. Good Life
23. Goodmorning
24. Hunnybee
25. Visit

Figure 1: My top 25 songs on Spotify for the past six months.

5 DISCUSSION

When having my friends 'playtest', I was surprised by how many users actually appreciated being able to visualize their Spotify data in this way. It became a fun activity to be able to share each other's screenshots of our plots, both as a visual way of showing one's top songs, as well as viewing people's preferences for certain audio features – several people boasted about being in the "far right gang", which are songs with high energy values in the default visualization, since energy is the default x-axis. By contrast, Figure 1 (my data) seems to have a balanced range of energetic songs, while generally having moderate-high danceability.

I found that this project was also great at skimming over a new playlist at a more comprehensive glance than going through it on Spotify. Whereas on Spotify, songs start at the beginning and you're usually in the dark as to how a song will sound, with this project you immediately get a sense of how the playlist and the individual songs will sound by looking at the audio features – in addition, the song previews on hover usually start playing at a characteristic point in the middle of the song, so you get a feel for the song and the playlist much more quickly, which is useful when trying to discover new music.

6 FUTURE WORK

I think the project could benefit from having a more comprehensive set of audio features for users to play around with, and also think the project could use more types of visualizations themselves, and not just an interactive scatterplot.

I also think it would be very interesting for users to be able to visualize other users' data. Currently, since access to a user's data requires authentication info for that user, this is not possible. However, if I made Interactive Spotify a full-fledged web app by allowing users to create a profile and hooked up a database onto the backend, Spotify data could be cached to the database and viewable by other users (with permission, of course).

Finally, I hope to expand the app by allowing users to filter by musical genre, as well as generate recommended songs for a user by using the average values of a playlist's audio features and a user's top artists and tracks as a seed.

It remains a question as to how two playlists can be easily visually compared on the same plot.

REFERENCES

[1] Joonho Ko. 2020. *Interactive Spotify.* https://interactive-spotify.herokuapp.com
[2] Spotify. 2020. *Company Info.* https://newsroom.spotify.com/company-info/
[3] Spotify. 2020. *How Social Distancing Has Shifted Spotify Streaming.* https://newsroom.spotify.com/2020-03-30/how-social-distancing-has-shifted-spotify-streaming/