Idalina Sachango

Data Visualization

Tiffany France

12-03-2022

An Analysis of Curated Spotify Playlists

Music has been a favorite pastime of mine for many years. I grew up in a family who greatly valued it as an art form - my dad would often make us sit next to him while he played his favorite songs and would ask us to try and understand the meaning of the lyrics. Anytime we did a family outing, we would drive to Barnes and Nobles and peruse the music section, my parents spending precious minutes finding new CD's they wanted to buy for extended family members. As I got older, this appreciation of music evolved. No longer was it just my parents' influence, now my older brothers were teaching me about their favorite artists and how to enjoy the songs that I liked on my own. Their haphazard lessons on burning CD's, allowed my music taste to transform into something I could consider my own.

As the years progressed and the era of CD's and Limewire were replaced by streaming platforms like Pandora and 8tracks, I not only had access to more music, but the networking style of these platforms made it so I could also share my tastes with others, so I began curating stories through music. I guess in a subconscious way, I started doing online what my parents did for me as a child; I'd hear a song that really moved me, and from the emotion of that track, I'd make a playlist with other songs that fit into the story I felt like was being told. From beginning to end I carefully decided on the order of tracks, the spaces between tracks where I needed to include an instrumental interlude, and what statement the final track needed to say. It was catharsis and poeticism and I held the entire process in very high esteem.

As Spotify and Apple music became the dominant streaming platforms, the control I had over the discovery of my music taste was largely released - Spotify's algorithm, although effective in recommending songs that users like, has made the process of finding music largely passive for individuals like myself who were so attached to their favorite songs and artists. In an attempt to reclaim the same feeling, I've attempted to curate playlists over the years in a similar way through Spotify's streaming services.

For this project, I decided it would be interesting to tell the story of the kind of playlists I make now using Spotify's API. I tend to think my music taste is heterogenous, so I wondered whether or not my assumptions about my listening habits were true and what kind of state I was in emotionally (and temporally) when I made each playlist. I looked at the common genres within and between playlists, the sentiment of the tracks within the playlists, how I described each playlist, and the hours I spent working on each playlist. For these analyses, I gathered track metadata, artist metadata, as well as the general playlist information provided by Spotify through their API.

The process for pulling my Spotify data was relatively straightforward. My Spotify library includes over 100 playlists that I have either made or that I follow, but the Spotify API only allows 50 requests at a time. Since my goal was to tell the story of each playlist, I decided that it would be beneficial to choose only the playlists that I had curated as meticulously as what I described above instead of all 100+ playlists. To this end, I identified 32 playlists that fit this criteria. Next, I manually gathered those 32 playlist ID's through the Spotify desktop interface. Once I had a list of ID's, I began the process of programmatically retrieving data from the Spotify API. The process to use the Spotify API is simple; just like with other social media API's like Twitter, I needed to create a developer account to navigate requests. The challenges manifested in the amount of data I needed to pull per second. I had to do 3 queries at every step. One to grab the track list of the playlist, a second to grab each track's information and metadata, and a third to grab the artist information. This was implemented using nested for-loops (which I wouldn't recommend in most situations) but needed to be done here. Because of the amount of requests I was making, I needed to think critically about how to circumvent rate limits. I chose to implement pauses within my code of 10 to 30 seconds depending on the size of the request. On average, pulling all of the data I needed for 32 playlists took about 1 hour. Next, I had to make my API pulls more efficient by batching the tracks and artists I needed information on. Each request had a certain limit on the batch size and a specific format, so I had to implement code that would concatenate ID's together for a specified batch size and format them appropriately. For reference, I used Spotify's Console Log; an interface that helps you understand the format of requests and the format of responses; Finally, I had to manipulate the data to get it into the format I needed for visualization. Most of this process was uncomplicated, the only anomaly being my decision to rescale all of my feature variables to a zero/one scale to better compare them to each other (zero being the lowest and one being the highest). This scaling did not work for the time signature feature as this feature did not lend itself to being ranked.

I used a mix of charts to display the story behind each playlist. To visualize popular groupings I used force directed bubble clouds. Here, the size of the bubbles denotes the size of the grouping. For example, the genres across all of my playlists are ranked by frequency of appearance. This frequency was used to dynamically calculate the radius of the circles and text. To compare playlists with one another, I used bar charts. With this, I was able to compare song features like 'danceability', 'speechiness', and 'loudness' (among others) for my 32 playlists. Lastly, I used observable code to showcase a playlist timeline. This dot plot showed what eras of music each of my playlists spanned. I used release date as a metric, where the first dot showcases the oldest release date of a song within the playlist and the last dot showcases the most recent release date of a song within the playlist.

Once I decided on the foundational visualizations, I set out to configure the best way to describe the stories behind each playlist. I looked at other Spotify data projects like those by Connor

Rothshild[1] and Brandon Lu[2]. I noted the decisions they made about what they considered important to show, their color schemes, and in Connor's case, how he was able to make his application accessible to other users. I incorporated either some of the same visualizations or some of the same decisions about data to display on tooltips, etc..

Although I am proud of the result, there were a few items I was unable to accomplish stylistically within the timeframe, but would have loved to include. My current homepage was supposed to be a 'catalog' type page, where the user would click on the playlist icon and would be redirected to the playlist specific page pre-filtered. I perused stackoverflow and other documentation but could not get it to work in the way I needed. Additionally, I attempted to make a page where I compared playlists between each other, but I felt like the story being told was not as clear or streamlined as my playlist specific page. With more time, I would have designed the aggregate 'story' with more complexity and with more differentiation between the playlist specific 'story' (I say with more differentiation because some of the visualizations were reused for both pages). Further, it was difficult to think of a story to tell about my streaming data that felt like it had a clear arc from beginning to end. For instance, if I had pulled data from some public repository on a natural phenomenon, I might be able to piece together different sources to create a nice picture. But this data is about me and without telling you my life story, it's hard to convey the reasoning of why they came to be in a way that feels immediately clear. In future iterations, I'd like to work on this more. In all, I am proud of what I was able to do in the 10 weeks we had and I came away with more understanding and definitely more appreciation for web development.

[1] https://github.com/connorrothschild
[2] https://www.brandonlu.com/spotify-data-project

Sources

Lu, Brandon https://www.brandonlu.com/spotify-data-project
Martins, Luís
https://www.linkedin.com/pulse/how-spotify-knows-your-feelings-almost-exploratory-data-martins/
Rothschild, Connor https://github.com/connorrothschild
Spotify API Web API, https://developer.spotify.com/documentation/web-api/quick-start/
Spotify API Console, https://developer.spotify.com/console/
Observable Beeswarm Plot, https://observablehq.com/@d3/beeswarm
Observable Dot Plot, https://observablehq.com/@d3/dot-plot
Observable Color Schemes, https://observablehq.com/@d3/color-schemes