



Analyzing users' music listening habits and the effect of track number on song popularity

Noah Picard, Elbert Wang, Emily Kasbohm, Matt McAvoy

Summary / Abstract

In this project, our group proposed, investigated, and tested two hypotheses about the Spotify dataset, as well as building a personalized Music Recommendation app. Our music-centric hypothesis tested whether track number influences the popularity of a song relative to an album's mean popularity. Our user-centric hypothesis asked whether a user will listen to a higher number of genres per artist as the user's "hipster score" increases. Our music recommendation app uses d3 to visualize your top 5 artist recommendations and various statistics about the genres and artists you listen to.

Music-centric Hypothesis

Data

Using Spotify's Web API, we used the random artist endpoint, the artist's albums endpoint, album's tracks endpoint, and track number and track popularity fields to gather data for our music-centric hypothesis that tried to find a correlation between track number and track popularity over a sample of artists.

For the Music-Centric testing, we took a sample of 1000 artists and one of their albums, totaling in around 13,000 tracks. We kept track of:

- Track Number in Album
 - Mean: 7.79, StDev: 4.78
- Track Popularity Score
 - Mean: 38.86, StDev: 17.18
- Artist Popularity Score
 - Mean: 72.35, StDev: 6.76

Hypothesis

Does track number influence the popularity of a song relative to an album's mean popularity?

Methodology

We gathered track numbers and popularity scores for the 100 albums, produced top-ranked artist, and compiled dictionary mapping album ids to a list of its tracks with their respective track numbers and track popularities. Creating a bar chart for the mean stdev for the track from the mean popularity for each album, we saw a curve similar to our final resulting graph (Figure 1), with high values for the first few track and a clear downward trend.

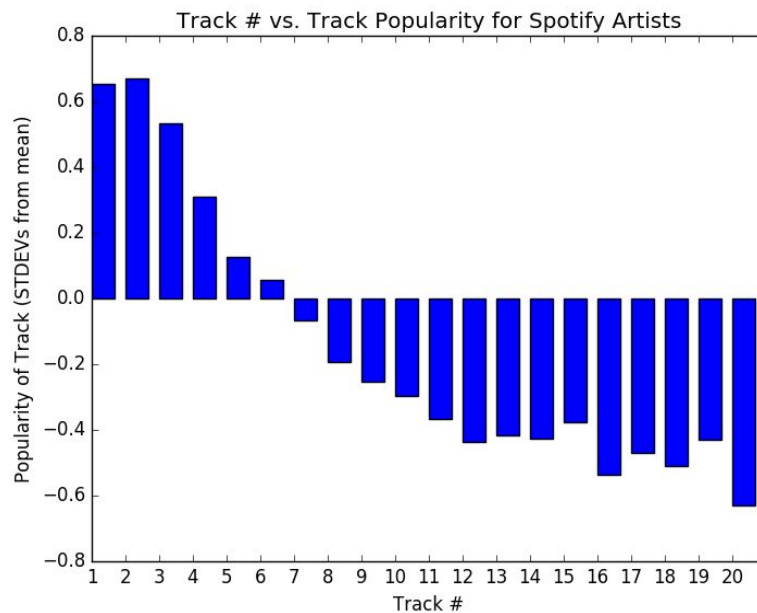


Figure 1: Comparison of track number to mean standard deviation of track popularity from album mean. Note the quasi-linear downward trend.

We continued by gathering the most recent albums from the top 1000 artists, and created graphs for the total of 12640 tracks. For each track number, we took the gaussian formed by the track popularities of tracks with that track number, and performed a two-tailed t-test against the null hypothesis that they had a mean of exactly 0 stdevs from the album mean, giving us the t-statistics visible in Table 1.

track #	stdev	mean	samples	t-statistic
0	0.653	1.136	986	18.059
1	0.671	0.971	940	21.196
2	0.533	0.943	937	17.317
3	0.311	0.925	937	10.308
4	0.127	0.854	929	4.538
5	0.056	0.891	922	1.919
6	-0.067	0.883	880	-2.257
7	0.194	0.85	856	-6.68
8	-0.255	0.83	841	-8.907
9	-0.297	0.825	815	-10.283
10	-0.367	0.812	734	-12.258
11	-0.435	0.774	618	-13.977
12	-0.415	0.817	502	-11.385
13	-0.425	0.818	421	-10.67
14	-0.379	0.882	352	-8.036
15	-0.537	0.766	294	-12.016
16	0.469	0.876	224	-8.005
17	-0.508	0.949	180	-7.187
18	-0.429	0.983	147	-5.284
19	-0.628	0.833	125	-8.426

Table 1: t-statistic of track numbers vs popularity

Results

With a confidence interval of 95% (alpha: 0.05), this allowed us to reject the null hypothesis for all track numbers (except 5), and say that each track numbers stdevs varied from the assumed random value of zero. Then, to verify that there was a downward slope in popularity as track number increased, we drew a regression line through the scatterplot of all tracks (Figure 2), and in a t-test against the null hypothesis that the slope of this line was 0, we got a p-value of 0.0 (confidence interval of 99.9999%). Although we see that our regression line only has an R2: 0.146, this p-value allows us to reject the null hypothesis, and say with confidence that the track number of a song influences its popularity relative to the mean popularity of the album.

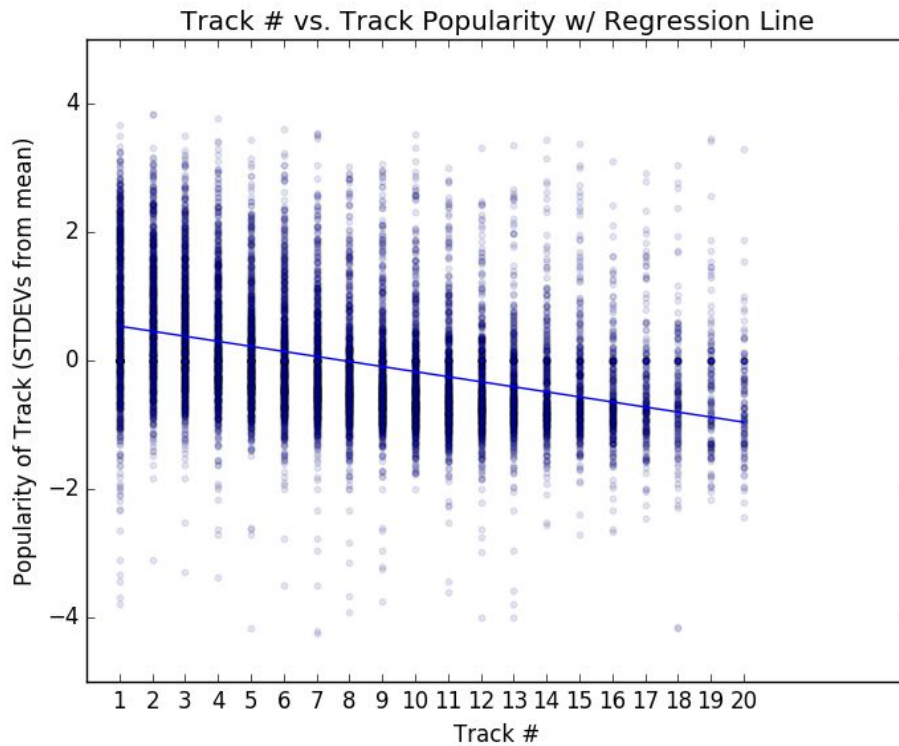
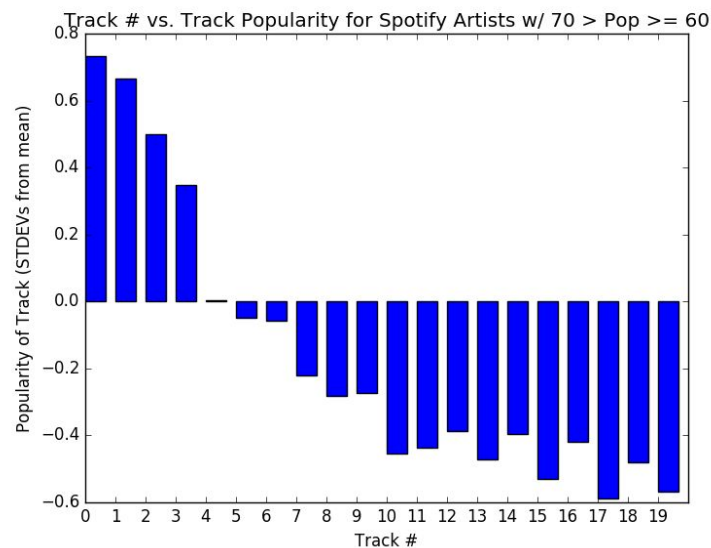
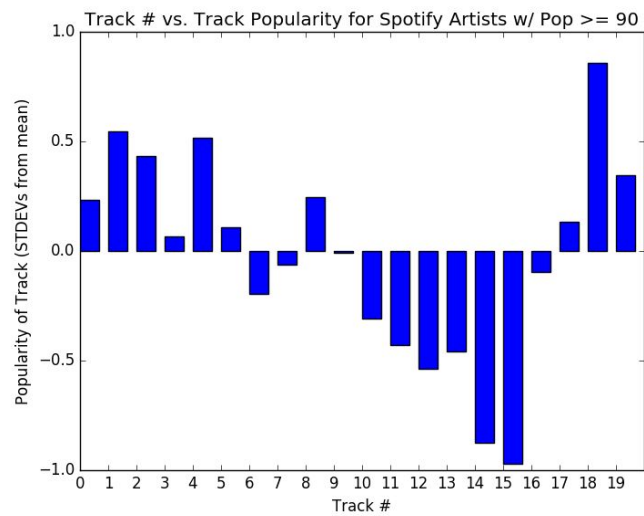
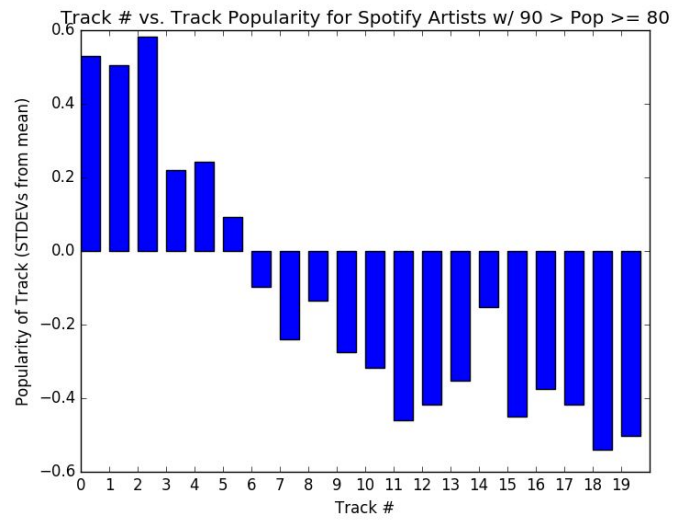
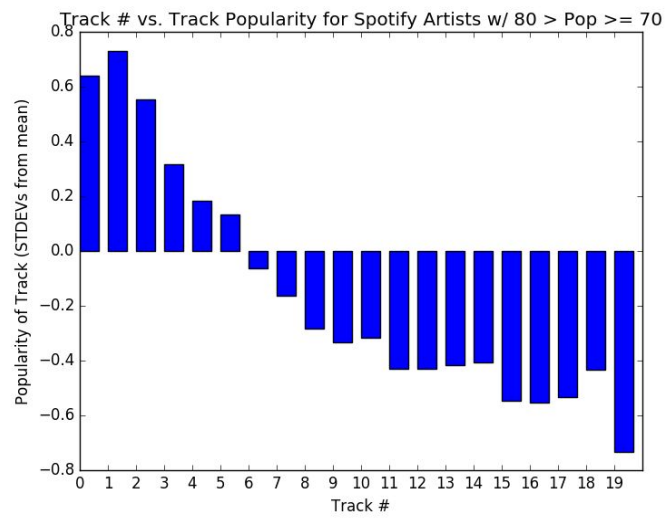


Figure 2: Distribution of track popularities for each track number, showing downward-sloping regression line of popularity with respect to track number. slope: -0.0787, intercept: 0.6134, r-squared: 0.1468, stderr: 0.00169

We also created separate graphs that plots track number vs. track popularity, separated by artist popularity. We tried to see if this correlation held across artist popularities, and to see if an artist's popularity affects where tracks are placed on an album.





User-centric Hypothesis

Data

Using Spotify's Web API, we gathered all tracks from all of a user's public playlists, getting all artists for those tracks.

For each user, we found:

- User's Hipster Score - The inverse of the mean popularity of all tracks listened to by the user, out of 100. ($100 - \text{mean}(\text{popularities of all tracks})$)
- Users' genre count: For each artist, Spotify's artist endpoint returns a list of genres. The genre count is count of genres of all artists from their public playlists.

For the User-Centric testing, for each user, the information we used consists of:

- User's Hipster Score = ($100 - \text{mean}(\text{popularities of all tracks in user playlists})$)
 - Mean: 53.55, StDev: 17.28
- Count of Genres of all of a user's artists
 - Mean: 53.74, StDev: 49.34
- Number of Artists listened to
 - Mean: 206.66, StDev: 248.92

Hypothesis

As a user's "hipster score" increases, will the user listen to a higher number of genres per artist?

Methodology

We collected 370 user ids, and acquired each user's playlists. We then found the average popularity of the playlist's songs, and the number of genres the user listened to. Using this, we created the distribution in Figure 3. Trying to form a linear correlation was clearly infeasible (when t-testing the slope against a null hypothesis of 0, p-value: 0.308), but the distribution's bell curve shape led us to wonder if there was a third factor influencing this dataset: the number of artists the user listened to. To check this, we plotted hipster score vs number of artists (Figure 4), and found that as the number of artists listened to increased, the users' hipster scores fell toward the mean of 50.

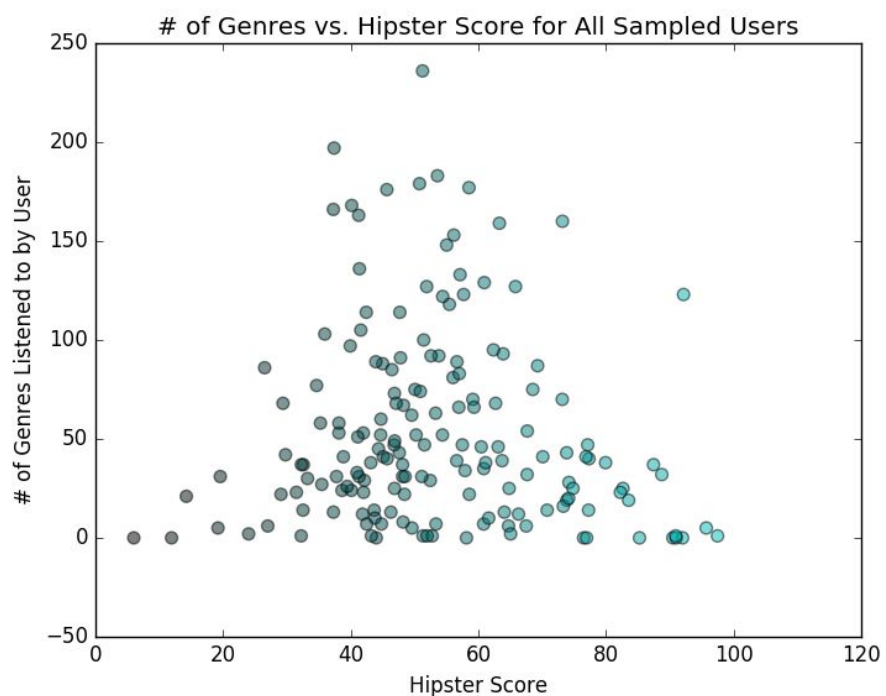


Figure 3: Comparison of number of genres and hipster scores. Note the bell curve shape but no linear correlation.

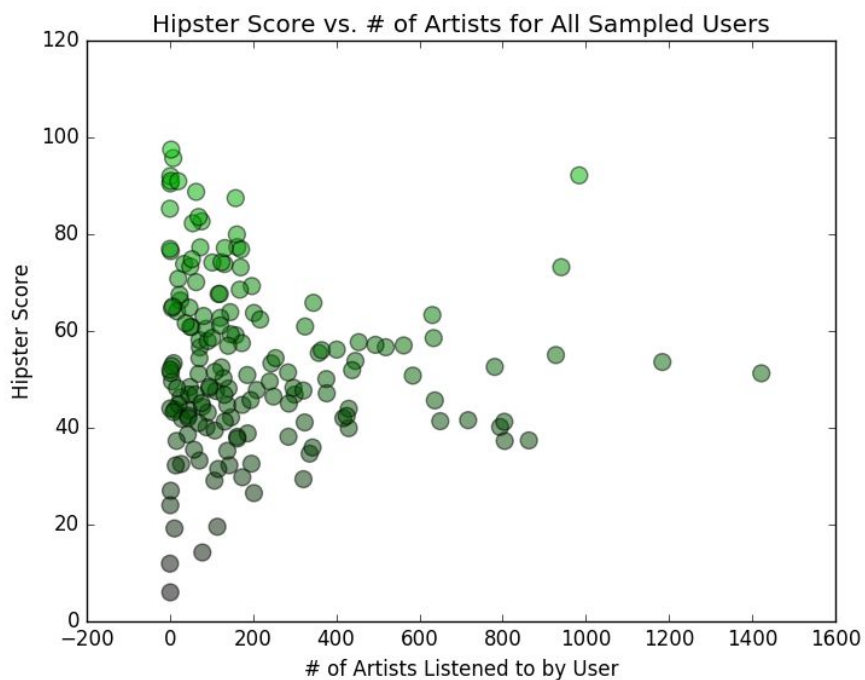


Figure 4: Comparison of hipster score and number of artists listened to. Note the hipster score converges toward 50.

Results

Following Figure 4, hipster scores converged toward 50 because each artist the user listened to could be considered as a random selection of popularity, the mean of which would approach the overall mean, according to the Law of Large Numbers. Next, we tested the hypothesis that as the number of artists increased, so would the number of genres (Figure 5).

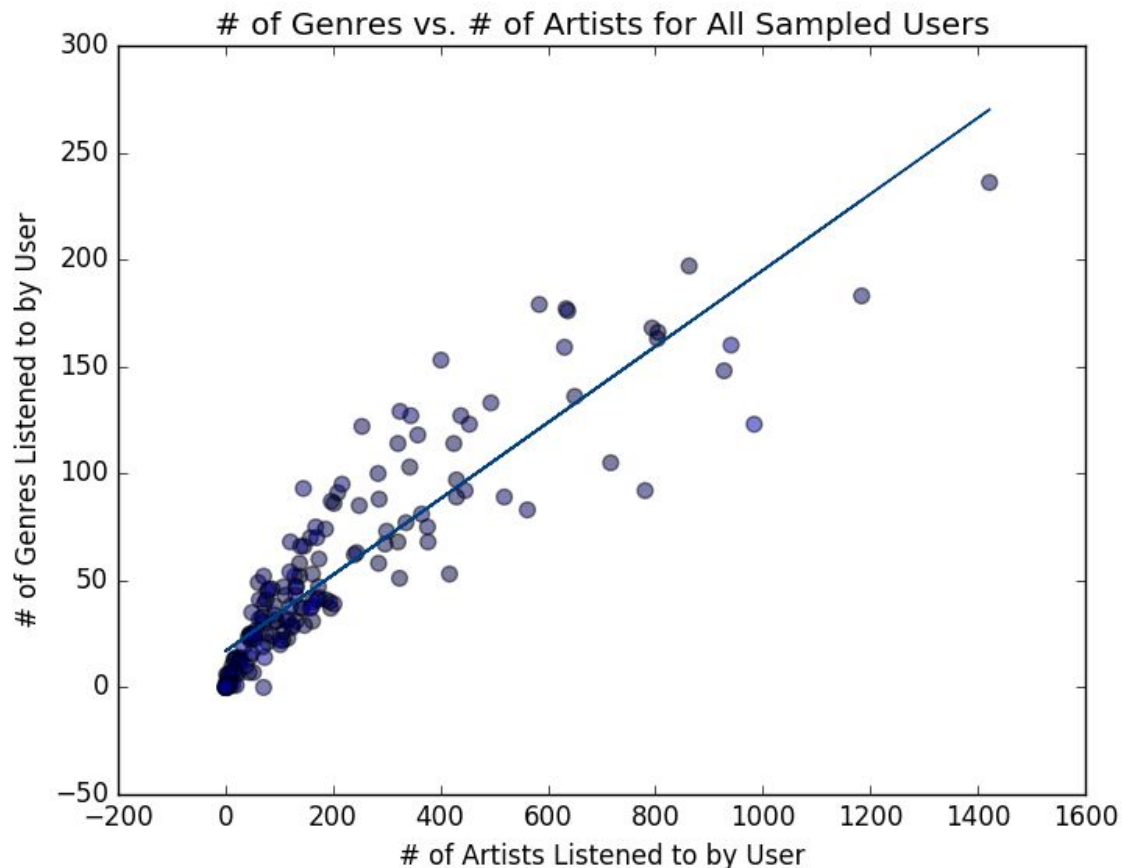


Figure 5: A clear linear correlation between the number of genres a user listens to and the number of artists a user listens to.

Here we saw a clear correlation, with an R^2 : 0.807. Knowing this, we decided to divide our count of genres by the number of artists listened to by the user, result in Figure 6. We did this because we hypothesized that a user who had a high hipster score would listen to a higher number of genres per artist (ie, if every artist was a different genre they would have a genre per artist of 1.0). Here we reached our final conclusion: since our regression line for this chart has a p-value of 0.718, and a correlation score of 0.028, there is no statistically significant correlation between hipster score and number of genres per artist, so we do not reject the null hypothesis.

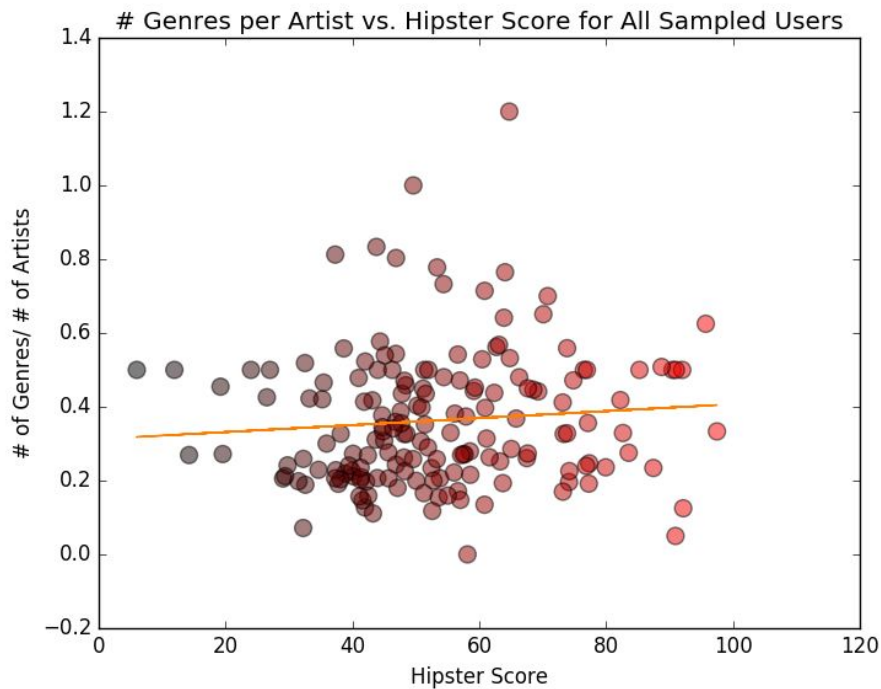


Figure 6: We expected to see some kind of correlation (wider variety of genres per artist, higher hipster score), but there was no strong linear correlation.

Challenges and Surprises

Our first challenge was coping with the privacy measures in place by the various APIs we were trying to use. Originally, we wanted to use the Facebook Graph API so we would be able to compare musical tastes among friends. However, some features were restricted only to those who made use of the API itself (by way of the User Access Tokens). We would only have access to our friends' data if those friend also used our app, had we created an app. Ultimately, while we had included it in our midterm report, we chose to abandon the Facebook Graph API aspect of the project due to these complications with permissions.

The APIs by default operate asynchronously, so getting user data synchronously presented an important challenge.

After the templates, server client connection, test calls, and cleaning were individually completed, the real trouble with callbacks became apparent. Because we intended to make a separate call to get related artists for each top artist, and each callback function from the call needed to add the related artists to the dictionary of counts, and only get the max artists after all calls were completed, we needed some way to keep track of the calls and their status. We were able to resolve this through the use of JavaScript promises.

Spotify's Web API rate limited us very frequently while we were trying to collect data, which made it difficult to resolve all of the JavaScript promises in one go. This led to much trial and error in seeing how much data we could collect, and ultimately we used JavaScript's `setInterval` and `setTimeout` methods to call Spotify's API at longer intervals so we would not be rate limited. We had to write all of our functions recursively to use `setInterval()`.

There were also a few challenges associated with visualizing the relationships between a user's top artists and the artists that we recommended to them. These included relatively straightforward issues, such as getting the artist names to properly align over the circles, to more difficult problems such as the best way to set artist images as the backgrounds for their respective nodes with d3, which we resolved through the use of SVG patterns.

For our hipster-score information, we were surprised to find the immediate lack of correlation between hipster score and number of genres, and then spent a great deal of time investigating the reasons for this (which included the previously mentioned artist count, but also that many artists, especially unpopular ones, did not have any listed genres, while others had up to 7 genres. This led to a series of filtering and testing until we settled on the final visualization (and negative results).

With regard to our proposal, there were some aspects of our planned visualizations that we were not able to implement, due to both time constraints and changes in the direction of our project to accommodate more investigative hypotheses. While it would have been interesting to create something similar to the [the Elvisualization](#), or a bubble graph where bubble sizes were based on how frequently a user listened to an artist and artist bubbles were clustered by genre, the post-proposal adjustments to our project by definition implied a need to change the nature of our proposed visualizations.

Future Directions

The Spotify Web API gives developers access to a variety of different aspects of the service, including artists, albums, and tracks. While we were able to make a recommendation system for artists, an additional aspect that we considered was recommending specific tracks for a particular user. This could be a useful expansion on our current application. Additionally, each Spotify track comes associated with a host of unusual metadata for audio features, including "acousticness," "danceability," and "speechiness". It could be interesting to investigate some of these features for correlations with popularity for the album or artist the track belongs to.

Furthermore, in the "Bells and Whistles" portion of our original project proposal, we mentioned that other APIs, including those for GraceNote and Last.fm, contained additional interesting insights on artist and track metadata. An interesting future direction would be to harness some of these capabilities, such as track moods or artist tags, to categorize artists in unconventional ways.

Finally, it would be interesting if we could take user preferences into account for the artists on which we base our recommendation system. This would allow the user to “seed” the application and look at whichever subset of the Spotify music sphere they may desire.

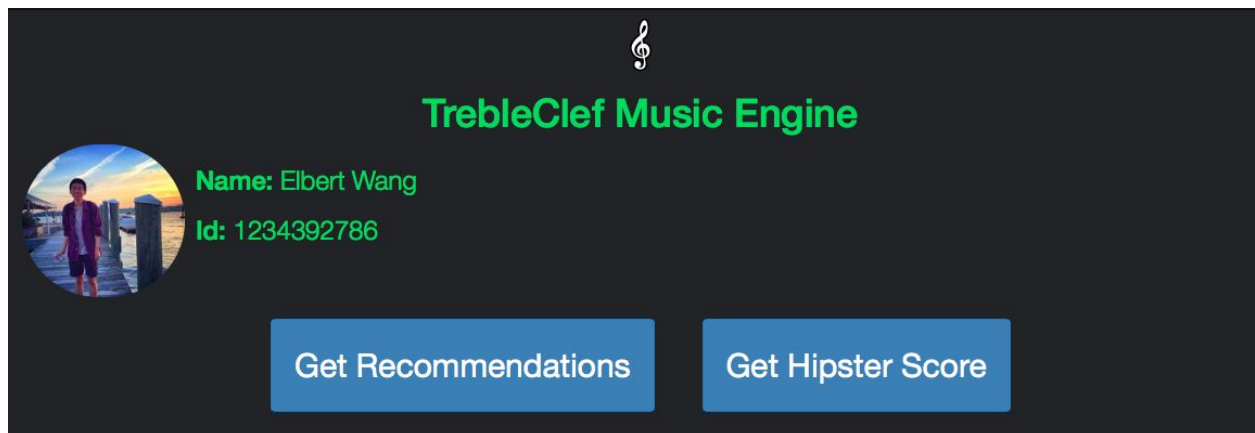
Open Questions to Explore

Is there a correlation between the music a user listens to (artist name and popularity, song popularity) and their popularity/number of followers?

Will artists grouped by related artists be similar to artists grouped by genre?

Does the relationship we found between track number and popularity change as an artist's number of followers increases?

Our Recommendation App



Feature 1: Get Recommendations

A general overview of the algorithm is as follows: when a user logs in, we use Spotify's API to get a user's top artists, then maps all top artists to their related artists. Then we count up how many times each related appears, and we recommend the related artists with the highest counts. We guess that users will like artists that their top artists are most related to.

We created a visualization that maps relations between recommended artists and top artists.

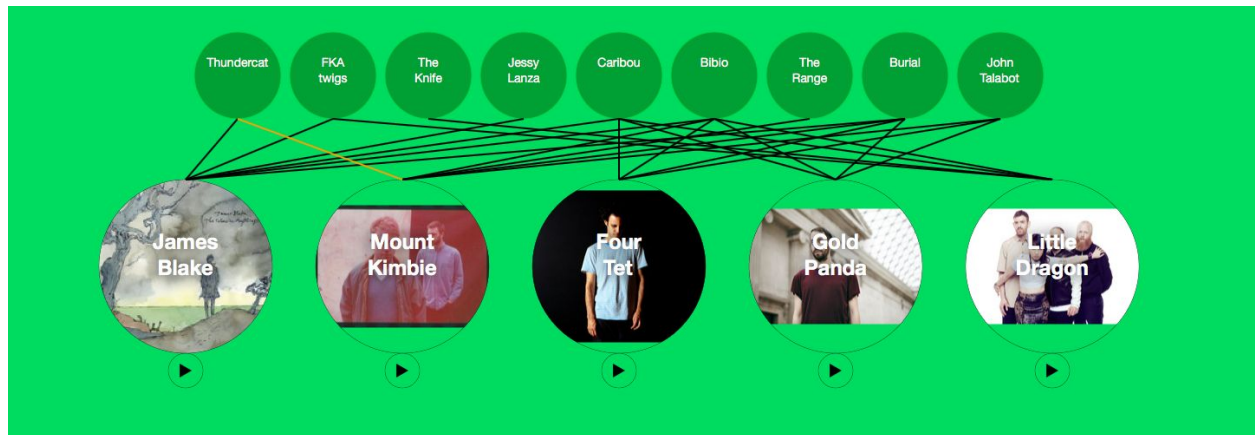


Figure 7: The bottom circles are the recommended artists, and the top circles are some of your top artists. The lines represent that the two artists are related according to Spotify's API. The bottom play button will allow you to play and pause a sample of that artist's most popular song.

Recommended Artists









 <p>James Blake Artist genre: indie r&b Spotify URL: spotify:artist:53KwLdlmriCelAZMaLVZqU Related artists: Caribou, Biblo, Jessie Lanza, Burial, Thundercat, FKA twigs</p>	 <p>I Need A Forest Fire James Blake, Bon Iver 0:00</p>
 <p>Mount Kimbie Artist genre: Spotify URL: spotify:artist:3NUtpWpGDoffm3RCGhSHH Related artists: Biblo, The Range, John Talabot, Burial, Thundercat</p>	 <p>Before I Move Off Mount Kimbie 0:00</p>
 <p>Four Tet Artist genre: Spotify URL: spotify:artist:7Eu1xygG6nJtLLHbZdQOh Related artists: Caribou, Biblo, John Talabot, Burial</p>	 <p>SeeSaw (Club Version) Jamie xx, Four Tet, Romy 0:00</p>
 <p>Gold Panda Artist genre: Spotify URL: spotify:artist:6xS3zemJD9h94IueOvGqVh Related artists: Caribou, Biblo, John Talabot, Burial</p>	 <p>You Gold Panda 0:00</p>

Figure 8: More information on each recommended artist

Feature 2: Get Hipster Score

Your hipster score is an index based on the mean of the popularities of all artist that you listen to in your public playlists. We also allowed users to see how many genres they listen to, the names of all the genres, their most popular artist, and their most obscure artist, both of which are based on popularity scores.

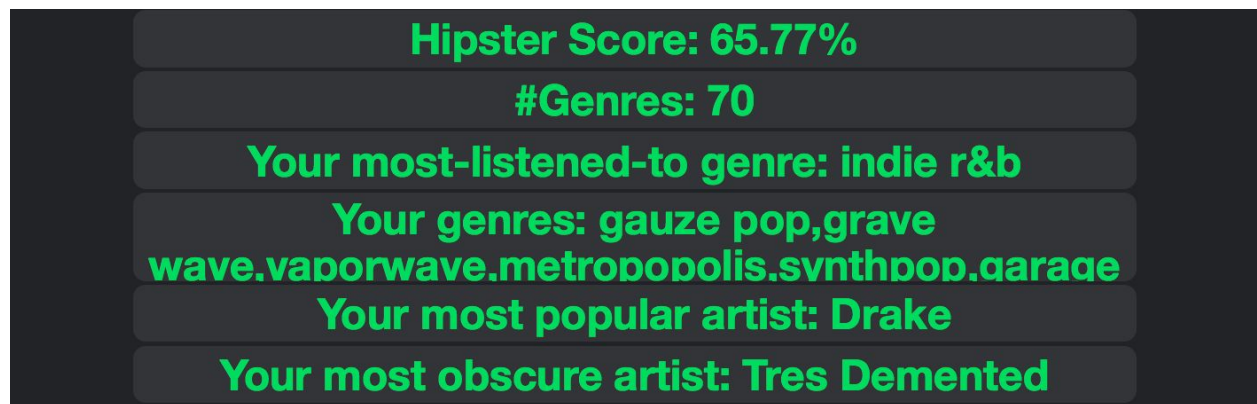


Figure 9: User statistics

Check out our GitHub repository!

You can see all of our code [here](#).

This final report is also available as a [blog post](#).