Manufactured Taste: How Spotify Shapes What We Listen To

Eleanor Adams, Harris Bubalo, Antonio Caceres, Daniella Calle

Northeastern University, Boston, MA, USA

Abstract

In our investigation of how featured Spotify playlists influence user taste, we used our database of 50 Spotify-generated playlists and ~1000 user playlists to find that the average value of audio metrics were nearly identical across both types of playlists, and the duration of songs were very similar as well. However, the top genres between both types of playlists differ, and there are more distinct artists across Spotify playlists.

Introduction

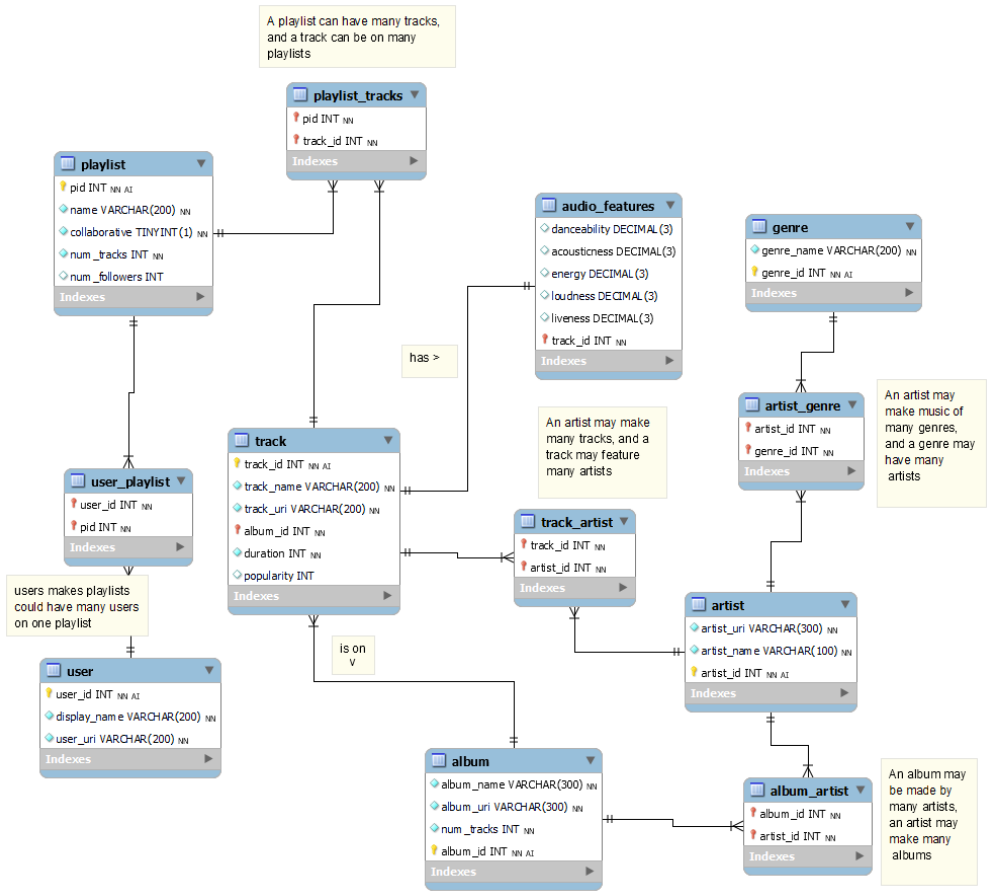
In the physical age of media, music discovery was irrefutably much more involved than it is today. Unlike now, where millions of songs and artists are available at our fingertips, audiophiles then found new music by either picking up an interesting-looking LP from the local record store, by listening to radio advertising, or from word-of-mouth recommendations. Since then, music streaming has become ubiquitous, with it making up 83% of the music industry's total revenue[1].

With such a pervasive form of media consumption, the question is raised: How do music streaming powerhouses, such as Spotify, influence music taste amongst consumers? Companies like Spotify already dictate the way money is made in the music industry, so it would be significant to see if they also decide what music is culturally relevant, and thus they would also dictate how *music* itself is made in the future.

To approach this question, we created a database of Spotify playlists; Some playlists (50) are all those created and featured by Spotify itself, and the rest (~1000) are user-generated playlists. With this, our main use cases involve analytically comparing user-generated and Spotify-generated playlists, with regard to common artists, genres, track popularity, audio metrics, and more. Through this overlap, we may examine the ways in which user taste reflects the music which Spotify pushes to its listeners.

Database Design

Our database design revolves around the playlist entity, as our project observes user-generated and Spotify-generated playlists. Playlists are made by one or more users, and each playlist naturally has one or more tracks within it. Apart from typical track attributes, such as name and duration, the Spotify API offers specific audio metrics (danceability, energy, etc.) which we store in the audio_features entity. Furthermore, each track is made by one or more artists and is on exactly one album. Albums are also made by one or more artists. Lastly, each artist is described as making music for one or more genres. Most key entities have a unique "uri" attribute; This is for ease of use when it comes to loading in data from the API, as each track, artist, and album is represented by a unique URI in the Spotify API.



Data Sources and Methods

To populate our database, we used two data sources: The Spotify Web API[3] and the Spotify Million Playlist Dataset[2]. The API provides access to all playlists generated by Spotify itself, as well as all track, artist, album, genre, and audio metric information needed for our analysis. The Million Playlist Dataset is a giant collection of user playlists, represented as JSON files, which we used 1000 of since loading information from the API turned out to be a slow process. Because of the organized nature of these two sources, no cleaning was necessary. We used Python and the Spotipy library to access API information from a Python notebook, wherein we created a script to read and load JSON information from both Spotify and user playlists into our database.

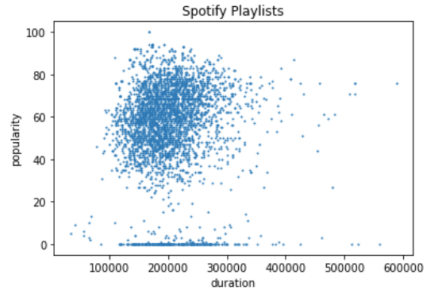
Use Cases

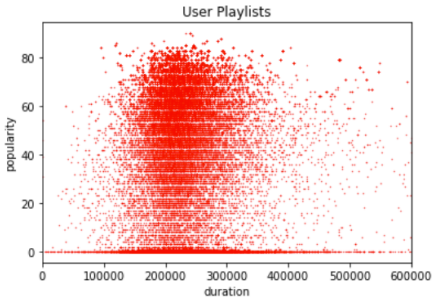
Because our database consists of playlist, track, artist, album, and genre information, a user with access to the database could ask any questions about artist/track/genre appearance frequency, which songs are the most popular, what songs have similar audio features, etc. However, because of the comparative nature of our study, we focused on the following questions:

1. How are track duration and popularity related, and are the tracks on Spotify and user playlists similarly long/popular?

```sql
-- duration and popularity of tracks on Spotify-generated playlists
select
    duration,
    popularity
from track join playlist_tracks using (track_id)
    join playlist using (pid)
    join user_playlist using (pid)
where user_id = 1;
```

```sql
-- duration and popularity of tracks on user-generated playlists
select
    duration,
    popularity
from track join playlist_tracks using (track_id)
    join playlist using (pid)
    left join user_playlist using (pid)
where user_id is null;
```





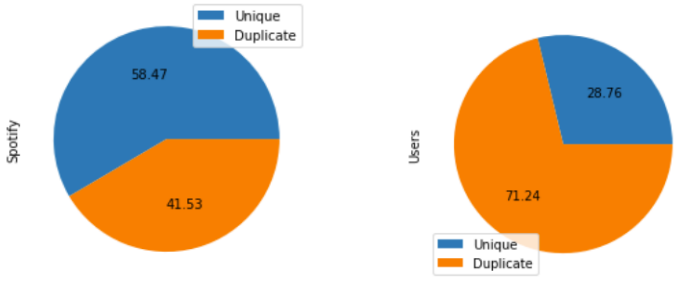
(Note: Popularity is a metric within the Spotify API. It is a value from 0 to 100. How it is determined is not disclosed by Spotify.)

Universally, most tracks fall within the 100,000 to 300,000 millisecond mark, and the most popular songs are centered there as well. User playlists have more duration diversity, as there are more points that fall below and above the aforementioned region. On top of this, users seem to enjoy less popular songs as well; The Spotify data has a gap below the popularity value of roughly 30, while user tracks are spread throughout the popularity range. While these deviations exist, most user tracks still fall within the common duration region, reflecting the Spotify playlist data to some extent.

2. What percentage of tracks on Spotify and user playlists are by unique (distinct) artists?

```sql
-- percent of unique vs. duplicate artists in Spotify playlists
select
    count(distinct artist_id) / count(artist_id) * 100 as 'pct_unique',
    (count(artist_id) - count(distinct artist_id)) / count(artist_id) * 100 as 'pct_duplicate'
from artist join track_artist using (artist_id)
    join track using (track_id)
where track_id in
(select track_id
from playlist_tracks join playlist using (pid)
join user_playlist using (pid))
order by count(artist_id) desc;
```

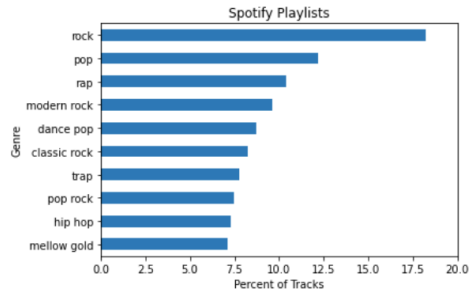
```sql
-- percent of unique vs. duplicate artists in user playlists
select
    count(distinct artist_id) / count(artist_id) * 100 as 'pct_unique',
    (count(artist_id) - count(distinct artist_id)) / count(artist_id) * 100 as 'pct_duplicate'
from artist join track_artist using (artist_id)
    join track using (track_id)
where track_id in
(select track_id
from playlist_tracks join playlist using (pid)
left join user_playlist using (pid)
where user_id is null)
order by count(artist_id) desc;
```



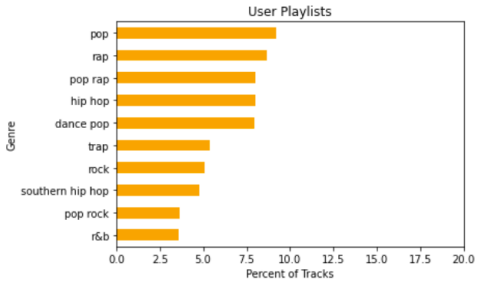
Despite the diversity of popularity and duration, user playlists are a lot less diverse in terms of distinct artists. Over 58% of tracks on Spotify-generated playlists are from different artists, while the same can be said about only 29% of tracks on user playlists. Intuitively, this could make sense, as Spotify playlists should encompass a wider range of artists in order to introduce users to new artists. Alternatively, this discrepancy could be explained by the sheer size of the user sample; With over 65,000 user-curated tracks, the probability of repeating artists is much higher.

3. What are the top genres on Spotify and user playlists, and what percentage of the tracks do they make up?

```sql
-- percent of genres on Spotify playlists
with spotify_genres as (select
    genre_name, count(genre_id) as 'genre_count'
from genre join artist_genre using (genre_id)
    join artist using (artist_id)
    join track_artist using (artist_id)
    join track using (track_id)
where track_id in
(select track_id
from playlist_tracks join playlist using (pid)
join user_playlist using (pid))
group by genre_name
order by genre_count desc)
select
    genre_name,
    genre_count * 100 / (select count(track_id) from track join playlist_tracks using (track_id)
                    join playlist using (pid)
                    join user_playlist using (pid)) as 'pct'
from spotify_genres
limit 10;
```




```sql
-- percent of genres on user playlists
with user_genres as (select
    genre_name, count(genre_id) as 'genre_count'
from genre join artist_genre using (genre_id)
    join artist using (artist_id)
    join track_artist using (artist_id)
    join track using (track_id)
where track_id in
(select track_id
from playlist_tracks join playlist using (pid)
left join user_playlist using (pid)
where user_id is null)
group by genre_name
order by genre_count desc)
select
    genre_name,
    genre_count * 100 / (select count(track_id) from track join playlist_tracks using (track_id)
                    join playlist using (pid)
                    left join user_playlist using (pid) where user_id is null) as 'pct'
from user_genres
limit 10;
```

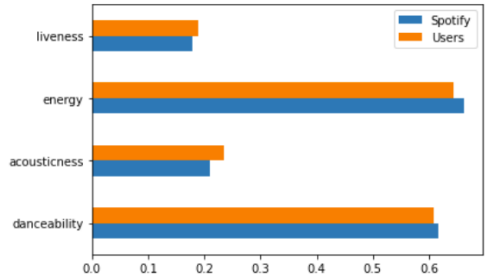


As expected, both Spotify and user playlists have popular genres like rock, pop, and rap within their most prevalent genres. However, Spotify playlists are particularly more rock-oriented, with various rock subgenres within the top 10 and with rock itself making up nearly 20% of all tracks. User playlists, on the other hand, seem to be more pop, rap, and hip hop oriented. On top of this, the top genre throughout user playlists, pop, makes up less than 10% of all tracks. This implies that user playlists may have greater genre diversity amongst their tracks.

4. What are the average audio feature values for Spotify playlists and for user playlists?

```sql
-- average audio metrics from Spotify playlists
select
    avg(danceability),
    avg(acousticness),
    avg(energy),
    avg(liveness)
from audio_features join track using (track_id)
    join playlist_tracks using (track_id)
    join playlist using (pid)
    join user_playlist using (pid)
where user_id = 1;
```

```sql
-- average audio metrics from user playlists
select
    avg(danceability),
    avg(acousticness),
    avg(energy),
    avg(liveness)
from audio_features join track using (track_id)
    join playlist_tracks using (track_id)
    join playlist using (pid)
    left join user_playlist using (pid)
where user_id is null;
```



(Note: Liveness, energy, acousticness, and danceability are all metrics from the Spotify API. They are all on a 0 to 1 scale. Liveness is the quality of sounding like "live" music, energy is how energetic the song sounds, acousticness is how much acoustic instruments/sounds are used, and danceability is how likely someone is to dance during the song. Spotify does not disclose how such metrics are calculated.)

Despite some of the differences identified in our other visualizations, the average value of each audio feature is nearly identical between Spotify playlists and user playlists. This is significant to note, as even if there may be an observable difference in the genres and artists on Spotify and user playlists, the *feeling* of the music on both is very similar.

Conclusions

In all, we were able to effectively create and use a database to compare the qualities of both Spotify and user playlists. While there were some similarities, such as the near identical audio features and the amount of tracks within the 100,000 to 300,000 millisecond duration range, differences in artist and genre diversity seem to suggest that this matter is not as simple as user taste being a one-to-one reflection of what music Spotify features. However, the influence of companies like Spotify remains undeniable, so further research would be beneficial, especially due to the following limitations of our study:

- **Sample Size:** Loading data from the API was a very slow process, which prevented us from loading a greater number of playlists from the Million Playlist Dataset.
- **Missing Metrics:** The API was strangely missing metrics that could have been useful, such as number of track streams, artist followers, artist monthly listeners, etc.
- **Data Recency:** The Million Playlist Dataset was created in 2017, so the songs within the playlists are a few years out of date, culturally.
- **User-Specific Operations:** To better examine how Spotify recommends music to its users, we would need to be granted case-by-case access to user-specific functions (e.g. music recommendation algorithms based on the music the user listens to) in the API, which we lacked.

Author Contributions

All of us worked on the initial database design. Harris and Antonio worked on the Python scripts and loading data into the database. Daniella and Eleanor worked on writing the queries. Harris worked on the presentation and the data visualizations.

References

1. Götting, Marie Charlotte. "Topic: Music Streaming." *Statista*, 9 Nov. 2021, https://www.statista.com/topics/6408/music-streaming/#topicHeader__wrapper.
2. "Spotify Million Playlist Dataset Challenge: Challenges." *AIcrowd*, https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge#citation.
3. "Web API." *Spotify for Developers*, https://developer.spotify.com/documentation/web-api/.