

# Spotify Song Final Report

Jared Allen<sup>1</sup>, Rephael Congmon<sup>2</sup>, Kishan Tailor<sup>3</sup>, Kedar Vyas<sup>4</sup>

**Abstract**—This paper seeks to analyze user data from Spotify in various ways. The paper will discuss the main objective for the project, data that was obtained and examined, final results, issues that were faced throughout, future work for the project, as well as an org chart.

**Keywords**—*Spotify, genres, user data, API*

## I. INTRODUCTION AND OBJECTIVE

Spotify is a global music streaming service that has become a popular form of social media today. Users are able to stream their favorite songs and podcasts with ease from their smartphones and computers. The main objective of our final project was to analyze user data using Spotify in various different ways. We wanted to successfully be able to see and evaluate what genres a user listens to and to successfully compare user data. This allows a Spotify user to analyze their friend's music habits with their own, find similarities in music taste, and even discover new artists.

## II. DATA

Our first step was to obtain and collect the data. For this we ended up using the Spotify API to gather a user's fifty top artists, which we then gathered each artists genre from, using Spotify's description of genres. Once we stored a user's top fifty artists and their respective genres, we went on to repeat the steps until we stored all four of our group members' Spotify top artist data. Lastly, we used the data collected by the Spotify API to compare similarities in music from user to user.

## III. MODELS AND ALGORITHMS USED

Since Spotify returned the list of genres fairly jumbled, we had to put in some work consolidating all the returned data into a usable format, so we counted each instance of a genre and then made a dictionary of each genre matching to the number of times appeared. After we had a count of every genre for each user, we also had a running total of the number of genres a user listens to in their top 50 artists. By dividing the common genres by the total amount of genres for both users, we could come up with a percentage similarity between two users.

## IV. RESULTS

Upon the finalization of our project, we were successfully able to compare music taste between two Spotify users. We were able to obtain a percentage number based on the similarities in genres between users. Overall, our team would consider the project a success, and look forward to potentially adding more to our Spotify comparison project.

## V. PRIMARY ISSUES ENCOUNTERED

We encountered a fair number of issues when working on this project. A large one was simply working with node.js,

which none of the team was quite familiar with, but the Spotify API used it, so we learned as we went.

We believe there to be two main issues with the data:

### 1. Genres are too specific and inaccurate:

Since we relied on the Spotify API to return the list of genres to us, we relied on it for comparisons. In the end this resulted in overly specific genres which we believe weren't entirely accurate to the artist.

### 2. Comparison between two users is a fun statistic, but is trivial as far as data analysis goes:

Comparing only two users is quite simple and adding more users to compare to concurrently could be very beneficial in showing the data. A user could see how obscure their music taste in comparison to everyone who's signed in and linked their account, with more statistics and visualizations being available the more users signing up.

## VI. FUTURE WORK

There is a solid amount of extra work that could improve our project. For one, consolidating genres that are similar would more accurately display results. For example, if one person has several different rap genres like Atlanta hip hop and Chicago hip hop, realistically, both of those genres could simply be considered hip hop for the sake of comparison. There would have to be some math to further increase accuracy of results depending on exact genre matches, but some genres that Spotify returned seemed so non-descript. of the artist it led us to believe that some genres existed for a just a few artists.

We also in the future wished to have more stats than simply a percentage. Seeing how many artists or even songs you share of your top 50 would be a fun and interesting stat for users to see. These would be relatively easy to implement, but we believe the comparisons could go even further. Song statistics outside of genre would be intriguing comparisons, like comparing how much instrumental vs lyrical content each user listens to, the BPM of their tracks, the time of day they are listening to music, how many hours in general, etc. There are lots of stats to unpack, and depending on how robust the Spotify API is, there could be a large selection of data to comb through for users to compare themselves with.

If users could add themselves to the large dataset of users inside our database, then they could compare music taste with all other users. There would have to be a base user that bases the comparisons for all others, but it could be

fairly interesting to see if you have a more obscure music taste than other users that have used our web app.

Lastly, we believe some visualizations could really improve the application. Graphs showing how similar the data is could really solidify the point our data makes and would also be interesting and fun to look at for any users.

## VII. ORG CHART

Each team member contributed equally for each task given for the week. Everyone was present in each active working session. The following list below is an estimated timeline for major project milestones:

- **Week 1:** Learn the Spotify API, discuss and framework the project.
- **Week 2:** Begin collected user data using the Spotify API for one user.
- **Week 3:** Organize user data into a list of genres and sub-genres.
- **Week 4:** Proceed to add another user and the user's top artist and top genre data.
- **Week 5:** Finalize datasets and connect front-end and back-end components.
- **Week 6:** Finalize user interface and user experience for webpage and proceed to testing, revising, and completing project.

Similarly, the following is a list of project members and their respective responsibilities:

- **Jared Allen** – Handled the front-end development of the project alongside Kedar. Oversaw the user interface of the webpage and worked with Kishan and Rephael to collect and organize the data obtained from the Spotify API.
- **Rephael Congmon** – Used previous back-end experience and knowledge in flask and node.js to take care of the back-end of the project. Worked with Kishan to handle the back-end of the project with tasks of pulling data from Spotify API and parsing through collected data.
- **Kishan Tailor** – Worked with Rephael to handle the back-end aspect of the project. In charge of creating the statistics for each user's unique data set. Kishan was also in charge of working with Kedar to create the algorithms that will feed into the main app and provide data for Jared to work with.
- **Kedar Vyas** – Handled the front-end development of the project as well as the user experience of the webpage with Jared. Worked with the back-end team to connect components together to create a working project.