Spotify Integrator: A New Approach to Spotify Wrapped

CS316 Final Report

Stephanie You, Abdel Shehata, Madison Nguyen, Sophie Mansoor

Our website, csharp's Spotify Integrator, serves as a mini Spotify Wrapped. It visualizes listening data and reworks the concept of Blends. Our main competitor is Spotify, which is also our inspiration for the project. We believe there were flaws associated with Spotify's data visualization options: Spotify Wrapped, which displays the user's top artists and songs. However, Wrapped stopped collecting data in October and doesn't reflect changes in listening history. With Blend, there's just one singular playlist, and the quality of that playlist isn't ideal since it mashes songs from user listening histories together, regardless of how different they are, thus making it more of a cacophony of songs.

Our tech stack includes SQLAlchemy in the backend, using the Python Flask framework to create a full-stack web application. The frontend is made with HTML/CSS/Javascript. The tables we have include:

User - shows basic information about the user in the database; used for user view/top items

Playlist - lists information about user's playlists; has a many-to-one relationship with songs and user table. Used to view playlists, tracks in playlists, and top items

Song - lists basic song information in the user's recent listening history in the past 6 months (i.e., artist, duration, etc.). Used in top tracks, recommendations, blending features, and top items

Song Characteristics - lists out advanced audio features of songs, including speechiness, valence, danceability, etc. Used in blends and recommendations to get more accurate results.

Top Artists - gets users' most listened-to artists in the past 6 months. Used in top artists, blends, and user view.

User View - contains information about a user's top tracks and artists. Joins with the user table to produce the top items feature.

Blend History - contains detailed information about each blend and when it was executed, including the blend type. Built for the blend history view on the website.

Our application employs K-means in the blending process. There are two types of blends: user blending (including genre blend, normal blend, and extensive blends) and playlist blending. K-means clustering segments songs into clusters based on audio features to identify and group similar tracks, allowing for a more accurate version of the Spotify blend feature. A looping process is implemented to guarantee equal representation for users in the blend.

We also use the K-Nearest Neighbors (KNN) algorithm for recommendations. This process involves standardizing the user's tracks and a collection of popular tracks on specific musical features. A KNN model, utilizing cosine similarity, identifies songs from popular tracks that closely match the user's listening preferences. Users can customize the number of recommended songs, resulting in a tailored list that resonates with their musical tastes.

To evaluate our system, we visualized the data of n=6 people in the Duke community by logging into their Spotify and showing them the data generated. We asked each of them about the accuracy of their top artists, tracks, and playlists and also to listen to their blended playlists and compare them to Spotify's blends. The sample size is small due to the website not being published and Spotify for Developers' limitations in needing to add the user as a developer to use our application. One of the respondents said that they "loved how they can see their top artists and tracks," as it was "more accurate than [their] Spotify Wrapped." One user responded "good stuff" when listening to the blended playlist, as the songs all had a similar vibe and weren't arbitrarily placed like those of Spotify's Blends. Through this feedback, we feel we have a valuable platform we hope to release to the public.

Rubric

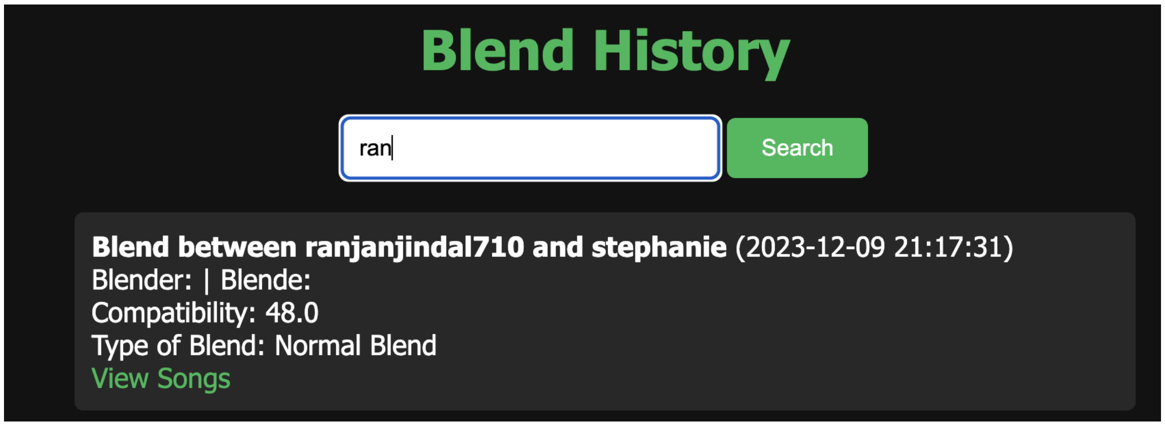
Users Guru (1) - 10 basic + 5 bonus = ____/15 points total

Basic Requirements

- Each user account has a unique ID. Other account information includes display name and image (+1 point) - **perfectly realized.**
- A user has a public view (+5 points) - **perfectly realized.**
    - One thing that was left out was showing the Spotify for the user that had no public playlists due to an error in screen sharing. You can view his Spotify profile here for confirmation.
- An existing user can view their playlists and blend with other users. (+2 points) - **perfectly realized.**
- A user can create a blend with exactly one other user. They can select the blend genre they want, and a link is generated for the blend. (+2 points) - **perfectly realized.**

Possible additional features:

- Search/filter blends based on genre or by people blended with (+1 point) - **perfectly realized.**



Blend between ranjanjindal710 and stephanie (2023-12-09 21:17:31)
Blender: | Blende:
Compatibility: 48.0
Type of Blend: Normal Blend
View Songs

  - Not in the video, but here is the searching mechanism

- Visualization of blends – see top people or top genre blended with (+2 points) - **perfectly realized**; can see top people blended with.
- Users can remove songs they don't want in a Blend (+2 points) - **perfectly realized.**

Songs Guru (1) - 8 basic + 2 bonus = ____/10 points

Basic Requirements

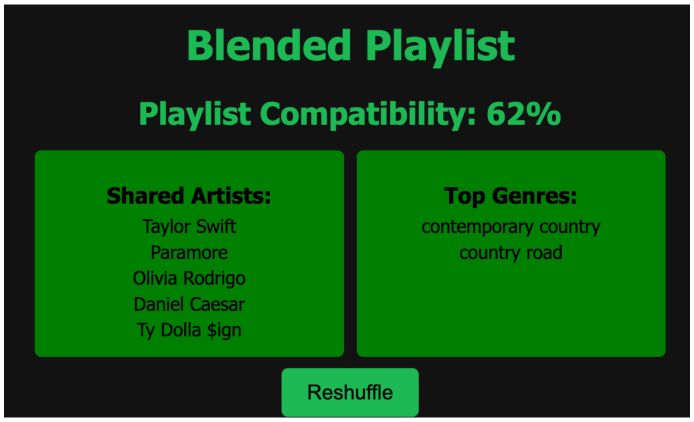
- Each song has a unique ID and has information associated with its Spotify catalog information, along with its audio features (+1 point) - **perfectly realized.**
- Each song can be included at most once in each blend, but a song can be included in multiple blends (+3 points) - **perfectly realized.**
- Blends should focus on a user's recent listening history. Each song used in the blend should take into account the last time the song was played by each user, along with how often it appears in a user's account (i.e., through playlists, saved/liked songs, user's top items) (+4 points) - **perfectly realized.**

Bonus Requirements

- Add recommendations based on songs in your playlists, matched against similar artists and tracks. (+2 points) - **perfectly realized.**

Blending Gurus (1) (2 features - blend history + blends w/ Song Gurus) 16 basic + 4 bonus = ___/20 points
Basic Features:

- The blend history shows the blend of all the app users. The blend history should include a unique ID for a blend, the blender (the person who initiates the blends), the blendee (the person receiving the link), the link to the blend playlist, the type of blend, and the day created. (+1 point) - **perfectly realized.**
- In a blend playlist, you can view the title of each song in the playlist and the users who "share" the song (at most, both users can share the song if it's a common listen between them) (+5 points) - **perfectly realized.**
- A blend is based on the audio features of a song (+5 points) - **perfectly realized.**
- Both users can see common songs, artists, and genres across their playlists (+5 points) - **perfectly realized.**


Blended Playlist

Playlist Compatibility: 62%

Shared Artists:
Taylor Swift
Paramore
Olivia Rodrigo
Daniel Caesar
Ty Dolla $ign

Top Genres:
contemporary country
country road

Bonus Features:

- Implement a % similarity feature (+2 points) - **perfectly realized.**
- Allow users to update the blend if they aren't happy (+2 points) - **perfectly realized.**

Playlist Guru (1) - 13 basic + 2 bonus = ___/15 points
Basic Features:

- Each playlist has a unique ID, owner ID, and number of items (tracks) in the playlist (+1 point) - **perfectly realized.**
- Each user should be able to see the tracks in a playlist and sort the playlist by song ranking, and in alphabetical order by song title (+7 points) - **perfectly realized.**
- A user can see the playlists of users they've blended with (+5 points) - **perfectly realized.**

Bonus Features:

- Blend two playlists based on similarity. (+2 points) - **perfectly realized.**

Extra Implemented Features

- Can see the top artists, public playlists, and top songs of every user in the database (**perfectly realized**).
- Extensive Blend - takes 100 random songs the user has recently listened to and blends with another user (**perfectly realized**).
- Top Genres - can see top artists and music genres listened to inside user profiles **(perfectly realized).**
- Rating: Can add a description and rating for a song, filter all reviews by user, and delete your reviews **(perfectly realized).**

In the future, we aim to optimize the extensive and genre blends since it was our biggest problem due to Spotify's API limitations. If a song has multiple artists, Spotify must fetch the information for every artist and their top genres. Spotify's API rate limit is calculated based on the number of calls your app makes to Spotify in a rolling 30-second window; having to gather 100 songs from each user, along with the genre of every artist with different endpoints, exceeds the fetch request when we tried optimizing it. Also, we would like to turn the project into a music community. In addition to the rating system, we could have a recommendation system where one user recommends songs and artists to another user. Finally, we'd like to improve the HTML/CSS for our websites because our strong suit is not web development.