uniTunes

Katie Kwan, Ambika Miglani, Rebecca Zuo, Rachel Souza

The Problem

Student musicians do not have a platform for themselves. It is very difficult to discover other student musicians who they might want to collaborate with.

The Solution

A platform that is built specifically for student artists. Using a recommendation algorithm, they can discover other student artists.

Databases

User Database (users):

- Contains a username, email, password, and top genre field.
- Username, email, and password are filled in when the user creates an account. Is it stored to verify their account when they log in.

Songs Database (brown_songs):

- Used the spotify API to convert a spotify playlist of student songs into a database
- Contains fields for name, genre, popularity, duration, danceability, energy, loudness, speechiness, acousticness, liveness, tempo

Song Database

genre	popularity	duration	album_art	danceability	energy	loudness	speechiness	acousticness	liveness	tempo
hiphop pop	35	182822	https://i.scdn.co/image/ab6…	0.888	0.458	-10.61	0.245	0.367	0.0936	123.998
hiphop pop	18	154012	https://i.scdn.co/image/ab6…	0.968	0.401	-11.295	0.356	0.482	0.117	120.009
latin pop	38	189545	https://i.scdn.co/image/ab6…	0.443	0.307	-11.665	0.0366	0.923	0.111	115.311
latin pop	17	166736	https://i.scdn.co/image/ab6…	0.764	0.343	-9.206	0.155	0.0429	0.145	95.168
acoustic pop	0	181315	https://i.scdn.co/image/ab6…	0.571	0.156	-16.75	0.273	0.864	0.344	136.593
indie	7	227983	https://i.scdn.co/image/ab6…	0.48	0.593	-7.32	0.0329	0.303	0.206	129.955
electronic	9	213494	https://i.scdn.co/image/ab6…	0.601	0.648	-12.628	0.202	0.00242	0.103	127.073
electronic	5	243421	https://i.scdn.co/image/ab6…	0.857	0.221	-14.702	0.0786	0.0359	0.143	108.038
dance pop	0	252917	https://i.scdn.co/image/ab6…	0.704	0.35	-16.193	0.121	0.344	0.351	117.211
dance pop	0	168923	https://i.scdn.co/image/ab6…	0.839	0.431	-12.782	0.215	0.569	0.112	129.982
indie folk	3	224952	https://i.scdn.co/image/ab6…	0.589	0.342	-7.553	0.0323	0.852	0.183	132.83
indie folk	1	216455	https://i.scdn.co/image/ab6…	0.623	0.225	-9.394	0.0343	0.888	0.131	78.17
indie r&b	62	159130	https://i.scdn.co/image/ab6…	0.736	0.503	-11.249	0.174	0.637	0.104	92.032

User Database

username	password	email	genre
ambikamiglani	tazo	ambika_miglani@brown.edu	null
katiekwan	hello	kkwan@brown.edu	null
rebecca zuo	testing	rzuo@brown.edu	null
rachel souza	testingtesting	rsouza@brown.edu	null

The Algorithm

- Uses a KD-Tree on python-generated clusters of songs to find the nearest song to a given input

- Features:
  - Sentiment analysis on genre and title.
  - Distance metric for attributes: acousticness, danceability, duration, genre, title, energy, tempo, loudness, popularity, speechiness.

Demo

Main Problems

Front end:

- User authentication → we could not get firebase to work, when we log out of an account sometimes it stays logged in even if another user logs in.

- Sending song preferences to the back end → our recommendations page does not receive the value of the preferences received by the front end

- Adding an upload page → Ideally would have implemented a page where artists can upload their own music using any link (youtube, spotify, soundcloud), but it was difficult to gather the data for each song for the recommendation algorithm if they were uploaded on different platforms. With the spotify API, we could have consistent values to compare.

Main Problems

Back end:

- Sentiment Analysis → configuring maven dependencies
- Should have had everyone write tests vs. one person.
    - hard to verify the accuracy of our algorithm, to objectively know if the suggested songs are correct.
- Edge cases for songs, certain songs, missing entries, weird formatting, etc.
- Having unique identifier for songs vs. dummy nodes, figuring out how to exclude that when running KdTree.
- Figuring out what attributes were most important.