

Song Recommendation Tool

Leonidas Ioannou

2454882i

Proposal

Motivation

Recommender systems are very common practice in today's technological industry. They exist in a wide variety: from music and TV shows to products on online marketplaces and advertising. While these systems are now built-in to the music streaming experience, developing a simple, stand-alone song recommendation application which allows users to influence their recommendations could make discovering music less overwhelming and more intuitive.

Aims

This project is a mix of web development and data science to develop a web application that allows users to input songs they know and rate them, as well as various parameters to get song recommendations. The recommendations are calculated via a taste profile that is built for the user, with song and genre data gathered from the Last.fm API. The data is processed, and the best recommendations are calculated in the back-end. The recommendations are then returned to the front-end and displayed to the user, also giving the user the option to automatically create a Spotify playlist with the recommendations they get.

Progress

- Front-end developed using React (JavaScript, HTML, CSS), back-end developed using Flask (a Python Framework).
- Can get top three genre tags of a song sent from the front-end from the Last.fm API.
- Can display proper track and artist names (in case of grammatical errors or featuring artists), and ratings the user inputs on front-end.
- Implemented a Taste Profile that keeps track of the user's top genre tags.
- Implemented system that gets tracks that correspond to the user's top tags, and various calculations (a system that gives each track a score) to determine which tracks make the best recommendations.
- User can set number of recommendations they want.

- User can set maximum number of recommendations they want from the same artist.
- The top three tags in the Taste profile are also displayed when recommendations are returned.
- Implemented buttons to remove most recently entered song, and to clear all songs input.
- Made wireframes for design.
- Implemented feature that allows users to automatically create a Spotify playlist with their recommendations.
- Users can prioritize some genre tags if they choose to.
- Styled the website based on the wireframes.

Problems and risks

Problems

- Figuring out how to use Spotify and Last.fm APIs.
- Spotify errors when searching for some tracks through the API (no results returned).
- Quality of recommendations was not great for a decent amount of time, although now they are satisfactory (room for improvement still exists).
- Issues with requests from front-end to back-end (timing, proper format)

Risks

- Spotify errors still exist, as it is an issue with the API. **Mitigation:** tweak current workarounds and implement new ones.
- How to give access to participants during evaluation. **Mitigation:** Either do evaluation on my laptop or find a way to host the application.
- Recommendations not being considered good enough by evaluation participants. **Mitigation:** Improve quality of recommendations by finding new criteria to boost track scores.

Plan

Semester 1 Remainder of Week 13

- Write up draft of evaluation process.
- Refactor code for simplicity and efficiency.

Semester 2

- Weeks 1-4: Finish up implementation (remaining features, bug fixes, final styling), conduct testing, finalize evaluation plan, and gather participants.
- Weeks 5-8: Conduct evaluation, write a significant amount of the dissertation, focusing on aspects that concern the implementation itself and the background.
- Weeks 9-11: Write up the rest of the dissertation, including evaluation results, performance measures, and any remaining aspects of the project that must be included.