SpotifyStats

Chris Laney, Noah Falanga, Seamus Collins, Justin Tisch

Advised by; Prof William Hawkins III

Goals

The overarching goal of this project is to create a dashboard for a Spotify users listening history as well as an algorithm for recommending music to them.

- User interface with custom made graphs to display users listening history in an engaging and aesthetically pleasing way.
- Recommendation algorithm based on data we are able to pull from users accounts along with macro trends in listening.
- All fueled by a large parser to distribute the data between the historic data and recommendation algorithm

Intellectual Merits

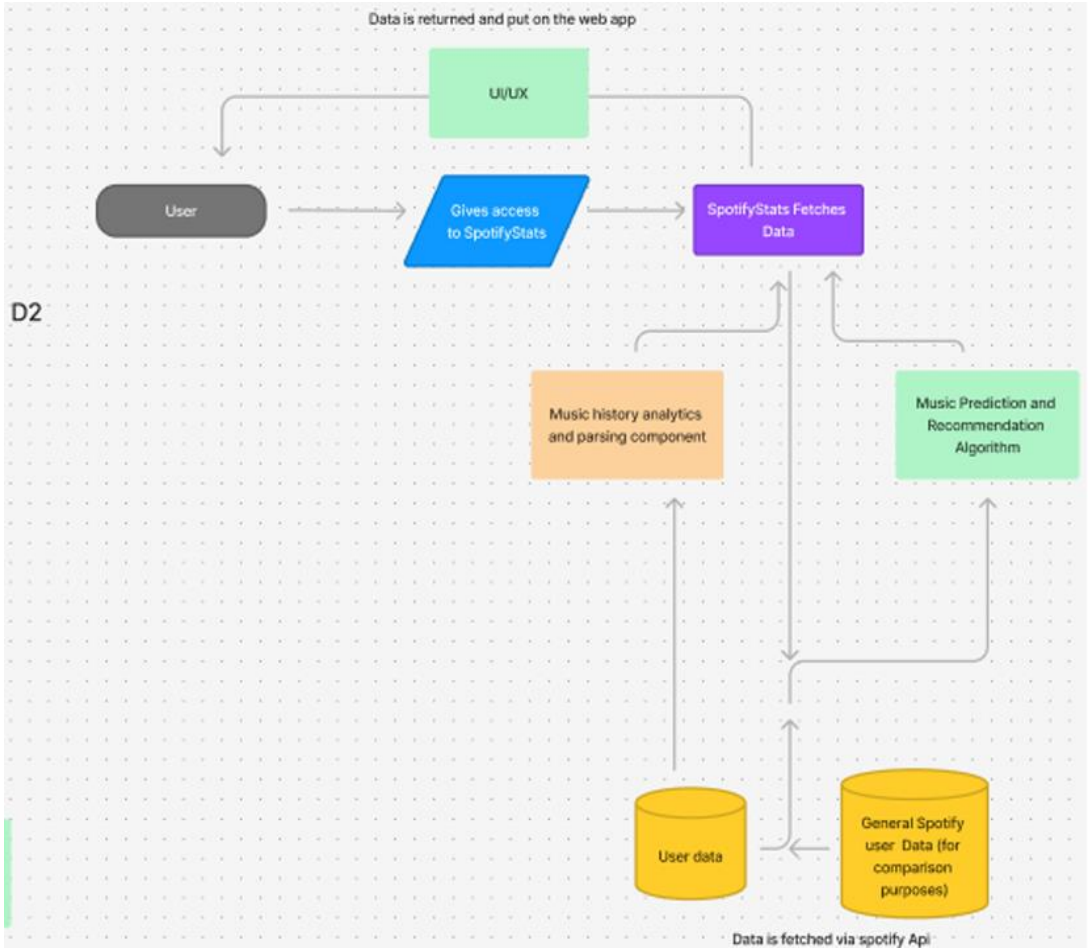
- **Recommendation Algorithm**, due to the ever changing nature of what data we are allowed to pull from users Spotify accounts, we must come up with a creative way of interpolating their data
- Using dimensionality reductions, and comparing users based on similarity of their listening history we are able to recommend songs to users based on where their interests align.
- **User Interface** the attractive part of this website is the user interface, we are striving for an engaging experience where users can take a tour of their data in a visual way. This will be done by creating custom visualizations to best display the metric we deem important

Broader Impacts

- This project aims to help music enjoyers analyze their data in a more open sourced way, without large industry players telling them what they should and should not listen to. Of course we are taking current macrotrends into account but also trying to create transparency about what other users are listening to.

Design Specifications

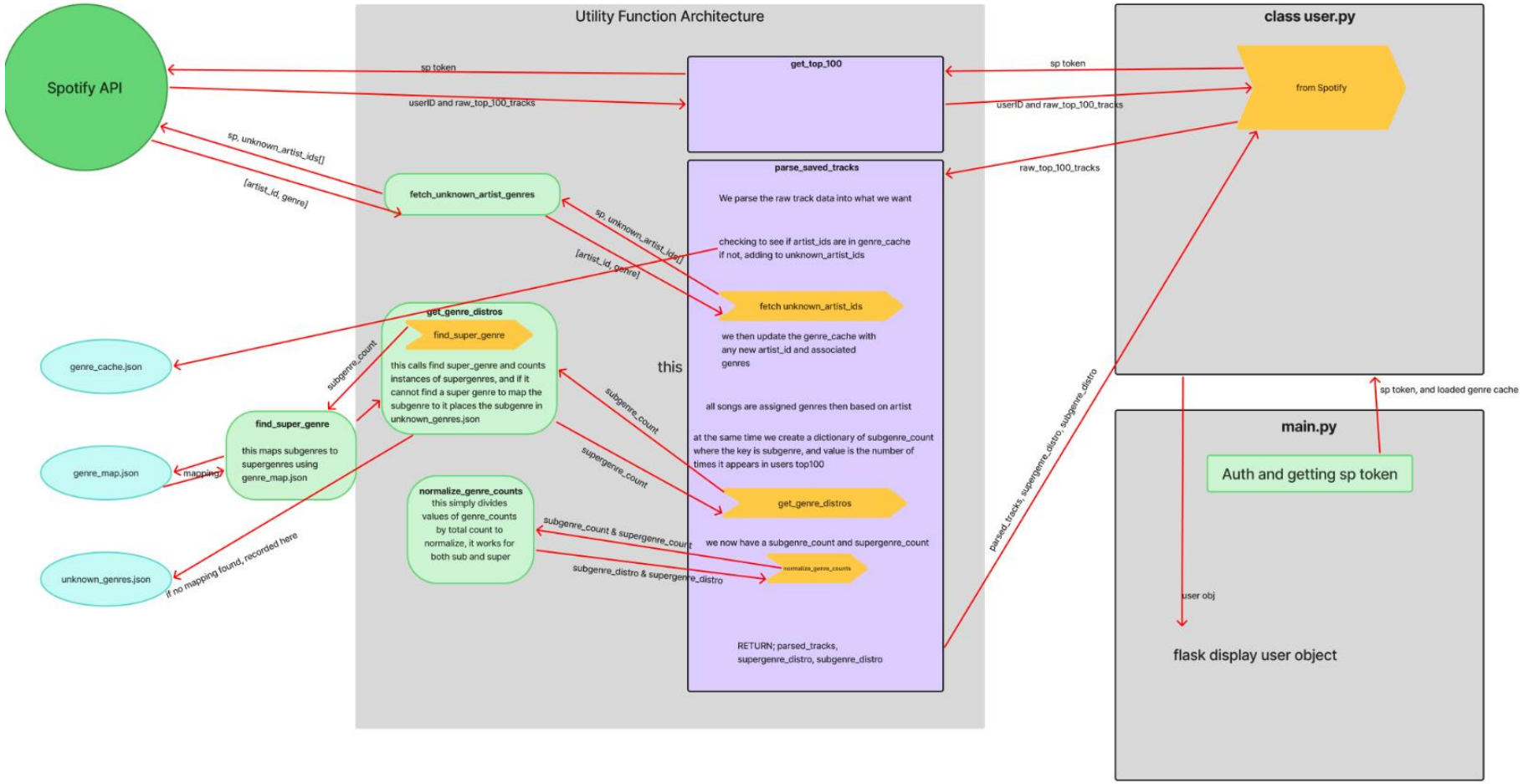
This is a high level architecture of our system design.



Some important keys;

- Purple Box – This is the large parser that requests, parses and distributes data to locations below
- Orange Box – This is the parser and api for visualizing the historical analysis
- Green Box – This is where the recommendation algorithm is, and once ran, returns data to front end

Parser Architecture



Technologies

- Flask: We are using the flask framework to create the server and its front end as well as its backend
- Programming Languages: We are using python for the front and back end of our application.
- Data Science Stack: We are using a variety of python libraries to help devise our algorithms; namely, we are using PyTorch for vectorization, sci-kit for data formulation and aid in creating learning algorithms, and matplot.lib for plotting

Completed Milestones

- Basic System Architecture
  - Implemented core parser architecture for Spotify data retrieval
  - Set up Flask server infrastructure for frontend and backend
  - Established basic user authentication with Spotify API
  - Created initial data pipeline for processing user listening history
- Frontend Development
  - Developed basic UI framework for displaying user statistics
  - Implemented preliminary data visualization components
  - Created user authentication flow and session management
- Backend Infrastructure
  - Built core data processing pipeline
  - Implemented initial version of genre-based analysis
  - Set up database structure for user data storage
  - Created basic API endpoints for data retrieval

Future Milestones

- February 2025
  - Complete clustering algorithm implementation
  - Develop interactive visualization for user clusters
  - Implement initial version of playlist generation algorithm
  - Begin Spotify app publication process
- March 2025
  - Refine recommendation system based on genre distributions
  - Enhance UI/UX with additional interactive features
  - Scale system architecture for larger user base
  - Complete Spotify app review process
- April 2025
  - Launch beta version with expanded user capacity
  - Implement advanced data analytics features
  - Add social features for sharing and comparing music tastes
  - Begin gathering user feedback for further improvements

Results

Completed Deliverables

- Core Infrastructure
  - Successfully implemented Flask-based web application
  - Developed secure Spotify API integration
  - Created robust data parsing system for user statistics
  - Established efficient database architecture
- User Interface
  - Built responsive dashboard for displaying listening history
  - Implemented initial data visualization components
  - Created user authentication and profile management
  - Developed basic statistical analysis displays
- Data Processing
  - Implemented genre-based analysis system
  - Created preliminary user similarity calculations
  - Developed data caching and optimization systems
  - Built foundation for recommendation engine

In Progress

- Algorithm Development
  - Finalizing clustering algorithm for user grouping
  - Implementing interactive visualization for user clusters
  - Developing intracluster playlist generation system
  - Testing and optimizing recommendation accuracy
  - Platform Scaling
  - Preparing Spotify app publication materials
  - Optimizing system for larger user base
  - Enhancing data processing efficiency
  - Implementing advanced caching mechanisms

Next Steps

- Complete and deploy clustering visualization
- Finalize playlist generation algorithm
- Submit for Spotify app review
- Scale beyond current 25-user limitation

Challanges

- The most significant challenge we have faced thus far is working with compliance to Spotify's changing of it's API; in November, Spotify restricted use of some of its statistics for song features, which was key to our learning algorithm. We are bypassing this pivot with a focus on using similarity as a function of a users' genre distribution instead. While this will provide a less accurate representation of a user's true similarity with other users, it should still provide a decent comparison, which may even prove to provide a more interesting grouping of users.