Final Project Reflection

I made a Python Flask application that takes an input of YouTube music and returns recommendations based on a machine learning model. An audioset dataset of YouTube songs gets passed into an embedding generator, which converts the data into a vector, and passes it into a recommendation engine with a library that performs an Approximate Nearest Neighbor algorithm to find a list of the closest recommendations in the dataset (using metadata). Flask and HTML/CSS were used to create a web application that takes an input and a number of songs we want to be recommended and returns a list of videos that most closely matches the inputted music. The GitHub repository with all the code, data, and usage information is located at https://github.com/Andre14254/music-recommender.

Overall, my recommender worked well on a small scale. In the demo (demo.mp3 in GitHub), I tested a violin song and a piano song and got recommendations for violin and piano songs, respectively, indicating that my model was doing the right thing. However, in the real world, recommenders are highly complicated. The Spotify algorithm involves optimizing a machine learning model for business goals: user retention, time spent, and revenue. It uses two components: content-based filtering and collaborative filtering. Content-based filtering examines the track's content and collaborative filtering examines the connection between tracks that users create. Spotify then analyzes the metrics on the characteristics of the tracks, such as danceability, energy, and valence (positiveness). Then, sections, bars, beats, segments, and tatums are analyzed. Finally, the model is applied to lyrics analysis (themes and meaning), web-crawled data (how people describe the music online), and user-generated playlists (mood, style, and genre). Spotify uses all this analysis in generating playlists using different algorithms like

user-entity affinity (how much does the user like an artist or track), item similarity (similarity between artists and tracks), and item clustering (how to split tracks and artists into different groups) (Pastukov, 2022).

Ultimately, I was interested in how recommender algorithms work in the real world and I learned about the complexity of using dozens of algorithms and models to create the best recommendations for the user. I enjoyed the process of learning how to build a model and using that in a web application. I took away a lot of skills from this project. I learned how to use a Python library (Approximate Nearest Neighbor) for an algorithm implementing cosine similarity/angular distance and how to create a web application with Python Flask and HTML/CSS. In learning about the Spotify algorithm, I liked how it was based on their goals of generating revenue relating to user satisfaction.

Works Cited

Pastukhov, Dmitry. “How Spotify's Algorithm Works? A Complete Guide to Spotify Recommendation System [2022]: Music Tomorrow Blog.” *How Spotify's Algorithm Works? A Complete Guide to Spotify Recommendation System [2022] | Music Tomorrow Blog*, https://www.music-tomorrow.com/blog/how-spotify-recommendation-system-works-a-complete-guide-2022.