Justin Tisch

Self-Assessment Two

Part A:

       After Chris laid the groundwork with Spotify API integration, I took the lead on researching and developing our music recommendation approach. Our initial plan involved using NLP and machine learning techniques, similar to ones I had previously worked with in class, combined with Spotify's audio feature data. Unfortunately, Spotify deprecated most of the features we planned to use. After coordinating several team discussions, we pivoted to a user similarity-based recommendation system built on genre distribution and listening history. I explored and compared multiple clustering algorithms, eventually concluding that artificial data wouldn't be sufficient to support our system at launch. To address this, I wrote a script that uses users top 100 playlists from and Spotify charts, allowing us to simulate about 500 realistic users using the same generation pipeline built for live users. I then handed off most of the clustering refinement to Noah, but we continued working closely to align our methods. In parallel, I developed and tested the playlist generation logic, which required me to collaborate with Chris and Seamus to integrate with the database and frontend systems.

       Originally, I had a defined scope in mind for my contributions, but the API pivot pushed me to learn and adapt well beyond that. I picked up practical skills across backend systems, database design, and cross-functional teamwork. For example, I had no initial plans to interact with our database, but while refining our clustering system, I needed to add a secondary index, which forced me to learn the inner workings of DynamoDB. I also got to apply knowledge from my IDA) class in a real-world setting, especially around clustering strategies and vector-based user profiling. Throughout the project, I had to account for practical limitations like API rate limits and AWS costs, which added another layer of design consideration. This experience strengthened both my technical problem-solving and my ability to contribute flexibly in a team environment.

Part B:

       As a group we flipped the typical Spotify experience, rather than just consuming algorithmically generated playlists, we gave users insight into the music they listen to and let them discover music through others with similar or contrasting tastes to discover better recommended music they may not already know. We accomplished this by building a web application that uses user clustering to generate personalized playlists based on genre distributions. Users could inspect their own listening patterns, view clusters of similar listeners, and explore playlists from both similar and dissimilar groups. Despite a major shift in direction after Spotify deprecated key audio features, we were able to pivot effectively, build a functioning system, and integrate backend, frontend, and database components into a cohesive product. Each

team member brought a unique skill set, and we all had to learn new tools and technologies to get the project finished.

Throughout the project, I gained a deeper appreciation for the complexity of group work. One of the biggest challenges was syncing efforts across the backend, bootstrapping, and frontend, especially when expectations and assumptions didn't fully align. I often found myself building placeholder logic, abstracting functionalities, or refactoring code to make sure future development was possible as planned or unplanned. That said, we stayed communicative and adjusted as needed. I believe I contributed a significant amount to the project across backend development, user data modeling, and problem-solving. Chris deserves recognition for the Spotify API integration, frontend capabilities, hosting, and helping glue pieces together. Noah really stepped up on clustering logic, I learned a lot about how to use clustering in a run time environment and not just data inspection. Seamus implemented and integrated the database well and helped make it easier for me to extend its functionality later on, he also helped add some key functionalities to the front end which were later instrumental to the verification of our back-end systems and final demo. All of this happened while things were constantly changing but we all coordinated and developed work I am proud of. While the process wasn't perfect, we supported each other well, adapted quickly, and ended up delivering something we're all proud of.