Data Science Two Project Report

My topic that I covered for this project was Spotify API calls to provide users with a chatbot that can assist them with any music-related questions they may have. However, this wasn't my initial plan. My initial plan was to do something similar to my midterm project (finding the relationship between NFL QB stats and wins), but with NHL goalies. My thought processes was I can use api calls from the NHL/ESPN website to find statistics related to goaltending and a separate website to run the ETL for team wins. However, the research and time I put into this were fruitless. The NHL disbanded its API system years ago, and ESPN has strict regulations regarding data scraping/api calling. So I brainstormed for a bit to find an interesting topic that could replace my previous plan, and while listening to Spotify, I found my answer.

After finding Spotify's API call system and package in Python, I knew I had made the right choice. I started writing functions and running API calls for typical questions a user might ask a Spotify-related chatbot. This includes albums by an artist, genre of an artist, songs by artists, and track info. However, I ran into a mental hurdle when determining the source of my ETL pipeline and local dataset. In order to follow the parameters of the project, I sought a distinct but related dataset I could tie into my other questions. This was harder than it may seem. Data for Billboard's top 100 was either outdated or not externally accessible with API calls. Additionally, while metrics of music performance did seem relative enough at first, the nuances of cleaning the datasets I found for such metrics swayed me against the idea. I then found that Spotify actually has a ranking system for popularity that it grades itself. This provided me with the artist/music performance metric I was looking for to use in my ETL pipeline, and with its close proximity to the other API calls I was running, I figured it would be a perfect fit. However, extracting and transforming the data into a local CSV file proved to be quite a challenge.

Much of the difficulty extracting and transforming the data was expected, however, I ran into an unexpected problem while attempting to extract. Due to the broadness and scope of the possible user questions, I saw it fit to extract an extremely large amount of artists' scores to cover the many possible questions. I believed that the top 20,000 artists would suffice. In hindsight, not only was that too many artists, but I should've adjusted my code beforehand to compensate for the rate limits. Unbeknownst to me, Spotify has an API call rate limit, and around the 3500 mark of artists extracted, I got prohibited from API calls from Spotify for about 18000 seconds, which is around 22 hours. This was very frustrating because I wanted to finish this project quicker so I can free up time to study for other exams. Additionally, I was afraid of running into this problem again and not being able to finish the project in time. I fortunately got

over this issue by simply looking into popularity scores in Spotify and realizing 20000 artists is way too much. I ended up creating a CSV file with 553 artists, which contains household names to niche artists that people could look into.

I made many key discoveries during my time working on this project. The main one I will carry with me into the future is the sheer amount of data available at our fingertips. Most household brands/companies have some sort of integrations with API calls and data scraping, which initially came as a shock to me. As I sit now, I look forward to tapping into these coding programs to create personal coding projects.

Another key insight/discovery I made working through this project is the practicality of some of these Python packages. The Spotify package I found made this project much simpler than it would have been without package integration. I look forward to researching and playing with more Python packages that could assist me in personal initiatives or career goals.

If I had more time to work on this project, I would've liked to tie a time component into a question. I was thinking of something along the lines of "Artist History" that would create a CSV file of all songs/albums releases and their dates. I would've also liked to do an "new release" questions that send newly released songs by certain artists. I could've perhaps done it so it could filter by genre, to give people a chance to pick and choose potentially new music to listen to. I believe that all these features and enhancements are very achievable with the Spotify API, which seems to have a knob and lever for everything.

In conclusion, I really enjoyed working through and creating something so cool. You always see these bots everywhere, but never really know what it takes to create one until you actually do it for yourself. I also enjoyed seeing something I worked so hard on being utilized. My friends started talking to my chatbot which I had joined our discord server, and I thought that was pretty awesome to be able to show something that I created that can be used for practical questions.