Task 1 Report

Colin Legge, Jacob Ericson, Suqian Wang

For our project, we decided to utilize data from Spotify in order to create an application that would be able to search for music and recommend similar songs. We utilized the Spotipy open source python library that integrates the Spotify developer API in order to gather our data. In order to use this library, we needed to create a Spotify API app on the Spotify developer website so that we could export our API Client ID's to our console in order to make pull requests to Spotify's API. We then wrote a python script to pull playlist data of over 1000 curated Spotify playlists. When iterating across playlists to scrape data, we found that some playlists returned a 404 error. Upon further investigation, this was found to be because some playlists were not made public, but were still picked up by the API. To circumvent this, we simply deleted the playlist ID's that returned the error. Despite this, we still ended up with 1303 usable playlists. Every 200 playlists, we had our program sleep for a couple seconds so as not to overwhelm the Spotify API and have our connection severed. The song data we pulled was then sanitized into a single text file containing only the playlist ID's. Using these playlist ID's, we pulled the song data from each playlist using another python script. However, after we pulled all the song data from the playlists, we found that we were still about 30,000 documents short of our 100,000 document goal. As such, we went back and pulled data from an additional 878 playlists. After pulling the data from these playlists, we ended up with ~127,000 documents of song data files, including song title, artist, album, and various other data pertaining to the songs.

However, a problem arose in our original output format. The information pertaining to 127,000 song data files was all inside a single JSON file. This caused problems due to how long it took to open and extract data. As such, we wrote a script to sanitize our single JSON file into

127,000 separate JSON files for easy access and analysis. This was done successfully after some trial and error with the JSON syntax. We decided to keep our massive JSON file for use at a later time in case we decided that it might be easier to use instead of multiple files. As of right now, we are capable of seeing what songs are from which playlist along with all the other song data. Seeing which playlist a song is in will be vital to us in being able to recommend other songs based on common occurrences of being in the same playlist.