1 Introduction

As companies have gathered more data about consumer behavior in recent years, they have effectively refined and utilized recommendation engines to improve their products. Online commerce companies such as Amazon were the first to leverage recommendation engines, but these engines have since spread to the media sector. Indeed, users of YouTube and Netflix can find customized suggestions of video clips or television shows to watch. These recommendations, extracted from user behavior and history, aim to improve the experience of the user and encourage continued consumption. Recommendation engines have also spread to the music sector. Spotify and Apple Music, the two most popular music streaming services, have embedded their song recommendation engines in the form of personalized playlists and radios based on a given playlist or song.

2 Project Details

This project intends to build a music recommendation engine that utilizes collaborative filtering implements content-based filtering. These methods are two of the most popular music recommendation techniques. Collaborative filtering (CF) uses the ratings and behaviors of other similar users in the system to predict the preferences for a given user. Previously, CF would focus on explicit feedback like giving a thumbs down or thumbs up rating to a song. Now, CF relies on implicit feedback that depends on the user’s listening behavior.

I will aim to implement an item-based collaborative filtering method. Item-based collaborative filtering recommends a song after analyzing the behaviors of other users who looked at the same song. Pompeu Fabra University in Spain has a department called the Music Technology Group, which released a data set collected from Last.fm that contains activity and profile data of 360,000+ users. I will first process this data (probably filter out outliers, remove those that contain a lack of information, etc). Then, I will build some models such as k-Nearest Neighbors and Alternating Least Squares, fit the models, and then look at the results. For example, with the k-Nearest Neighbors model, we can look at the cosine similarities of the
suggested songs to see how close they are to the queried song. The songs with the best cosine similarities will be recommended.

Although CF generally outperforms other techniques of music recommendation, it is not without its disadvantages, namely the cold start problem (lack of data for a song) and gray sheep users (users who have very atypical tastes won’t receive accurate suggestions). To address the disadvantages of CF, many recommendation engines also use content-based filtering. Content-based filtering utilizes the musical data of a song and recommends other songs that share similar musical components. From using convolutional neural networks to classifier models, there are a variety of ways that content-based filtering can be implemented. Nevertheless, content-based filtering depends on the features extracted by the system from the music, and if those features fail to effectively represent the data, then the suggestions would not be optimal.

The next step in the project will be to add a content-based filtering component to the recommendation engine. Two data sets I will likely use are the GTZAN and Free Music Archive (FMA). The former contains 1,000 half-minute song excerpts and the later contains 106,574 songs of 150+ genres, all of which have already been classified. First, the songs will be transformed into a form called mel-spectrogram. Mel-spectrogram uses a logarithmic scale to represent the time-frequency of sound. Then, we will build a convolutional neural network that will determine important features of the songs and then output the probability distribution of the song’s genre. These probability distributions will be plotted in a high-dimensional space of genres so that we can compare the song against all the other songs. Given a queried song, we can use this high-dimensional space, calculate the closest $n$-songs, and output those as the recommended songs.

The final recommended list of songs will be a combination of the songs suggested by the collaborative filtering and content-based filtering methods.

3 Timeline

2/04/2019 - 02/25/2019 (3 weeks) - Implementation of Collaborative Filtering
2/25/2019 - 3/18/2019 (1 week) - Process and collect data for content-based filtering
3/18/2019 - 3/25/2019 (1 week) - Implement data transformation into mel-spectrogram
3/25/2019 - 4/15/2019 (3 weeks) - Implementation of CNN for content-based filtering model
4/15/2019 - 4/29/2019 (2 weeks) - Refinement of engine

4 Deliverables and Stretch Goals

The deliverables for my project include a project report and the code I wrote for the project.

There are many stretch goals that I could implement if I have some extra time. The collaborative filtering and content-based filtering could take into account more complex statistical techniques. Moreover, we could
implement other music recognition methods such as feature-based recommendation that could improve our engine.