

Daily Log

Monday September 30

Since I wasn't happy with the efficiency of the library I'm currently using to download songs from Spotify (Spotify-Downloader), I spent the class searching GitHub for more a better library. I tried a couple of them out, but I the installation process was so tedious and in the end I found it a lot better to just keep using Spotify-Downloader. It works fine even though it's a little slow in my opinion. In the end my goal is to download usable sound files, and it accomplishes that.

Tuesday October 1

Spotify-Downloader downloads songs as M4A files, which is a format used mainly by iOS and MacOS devices. In order to use the librosa library to analyze audio files and perform calculations, my song files need to be in WAV format (or MP3, but WAV is preferable). The library doesn't work with M4A files as far as I can tell. As such, I spent today's class trying to find a suitable audio file format converter. My criteria were the same as for my Spotify-Downloader: a Python library that I could actively code with. As of today, I hadn't found one. On the plus side, I was able to calculate the MFCC of a sample song file from the librosa library, so now I know how to do that.

Thursday October 3

I discovered that Spotify-Downloader has a class just for converting audio files to different formats! I tried to convert one of my previously downloaded M4A files to a WAV file, but it didn't work. However, after reading the error message, I noticed that I was being told I was missing a library called FFmpeg. I looked it up and it turned out to be a library needed by Spotify-Downloader to process audio and video files. I downloaded it using Homebrew and I stopped getting that specific error... but now I'm getting one that says my program can't find the file I'm inputting even though I put it in the same directory as my Python script. I'll look into that next week.

```

4 import spotdl.convert
5 import librosa
6
7 spotdl.convert.song("narco.m4a", "tuyo.wav", "Senior_Research", trim_silence=False)
8 # y, sr = librosa.load("tuyo.m4a")
9 # mfcc = librosa.feature.mfcc(y=y, sr=sr)
10 # print (mfcc[0]) |

```

Figure 1: Attempting to convert an existing M4A file to a WAV file.

Timeline

Date	Goal	Met
9/16/19-9/20/19	Figure out a way to extract data from playlists with several songs without having to input each song's URI individually	Yes
9/23/19-9/27/19	Build a logistic curve that takes features of several popular songs to plot another song's potential popularity on the curve (I'll be comparing a few different methods of supervised learning before I integrate MFCC)	No, first I have to figure out how to obtain MFCC from a song on the Hot 100 playlist with librosa
9/30/19-10/04/19	Learn how to extract MFCC data using the Librosa Python library	Yes, but I need to convert the MP3 song files I've collected into WAV files before I can calculate MFCC with them
10/07/19-10/11/19	Begin building logistic curve using MFCC and song features as inputs	In progress
10/14/19-10/18/19	Begin building neural network using MFCC and song features as inputs	In progress

Reflection

I discovered a crucial problem with my code, which was that I was working with the wrong audio file format for the library I'm using for audio analysis, but luckily I found a solution to that problem. This week I learned that GitHub is my friend: literally every useful library I've worked with so far has been found on GitHub. As someone who has never previously worked with packages/libraries installed from GitHub, I'm really impressed with its usefulness. When I started my project I hadn't realized how much work it would take just to get useable sound files for my calculations and algorithms, but I've been solving whatever issues I come across and I know the rest will come once I'm past this phase.