Journal Report 4
9/23/19-9/27/19
Victoria Agrinya
Computer Systems Research Lab
Period 1, White

Daily Log

Monday September 23

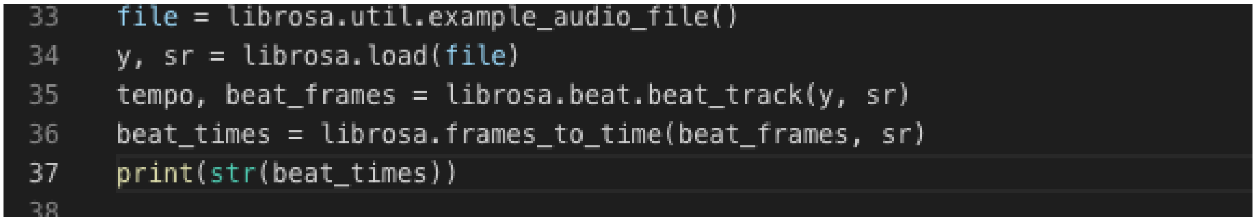
I spent the class exploring the librosa Python library that I'm using for in-depth audio analysis (namely MFCC). It includes an example track for analysis as well as an in-depth tutorial, so I followed that to get some pretty basic tempo data from the sample track. I'm still figuring out how to get MFCC, which takes a few more inputs and is a fair bit more complicated.

Tuesday September 25

I realized that in order for librosa to calcualate a song's MFCC, it has to take an actual audio file as input. Spotify's API doesn't allow developers to download full songs to devices, and I couldn't find a way to just download a short preview clip of a song, so I spent the class looking for a way to download the playlist's contents. There are plenty of services like this online, but I ahd specific criteria: preferably I wanted it to be a library that I could actually code with, and I wanted to be able to identify a specific Spotify playlist to download songs from. I didn't find anything this class.

Thursday September 27

I found a Python library on Github that will accomplish what I wanted last class. It's called spotify-downloader and it downloads the Youtube versions of songs as MP3 files using a Spotify song ID or playlist as input. There are a couple of downsides: I have to use it in the terminal (which I don't like doing, but I'll survive) and it takes several seconds to find and download each song, so it'll take a while to get the new Hot 100 playlist each week. I'm still searching for a more optimal solution to my downloading problem, but for now this is what I'm working with. For now I'll just have to download the playlist before class so I'm not spending too much time waiting for it to load when I could be working.



```
file = librosa.util.example_audio_file()
y, sr = librosa.load(file)
tempo, beat_frames = librosa.beat.beat_track(y, sr)
beat_times = librosa.frames_to_time(beat_frames, sr)
print(str(beat_times))
```

Figure 1: Song tempo calculation using the librosa audio analysis library.

Timeline

Date	Goal	Met
9/9/19-9/13/19	Learn how to use Spotify Web API with Python and collect preliminary song feature data from several Billboard Top 100 songs	Mostly, still have to figure out how to get data from a playlist
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	In progress
10/07/19-10/11/19	Begin building logistic curve using MFCC and song features as inputs	In progress

Reflection

This week I really discovered how many little details can go into one research project. I initially hadn't realized how many relatively small obstacles I could run into that would take up my time. I'm still happy with my progress, though. I face whatever issues I come across head-on and by the end of every week so far I've accomplish tasks that seemed difficult on Monday. My biggest obstacle so far is one that I'll tackle next week: figuring out what components of an audio file go into an MFCC calculation and using librosa to do calculate a song's MFCC.