Journal Report 2
9/9/19-9/13/19
Victoria Agrinya
Computer Systems Research Lab
Period 1, White

Daily Log

Monday September 9

Discovered the Spotipy library on Github. Spotipy is a Python library that provides direct access to Spotify's Web API, which is incredibly useful and convenient. I still need my Spotify developer account to access certain things such as user data from Spotify, but Spotipy has a module that allows you to input your Spotify developer credentials and retrieve such data. I attempted to install Spotipy via pip in the command line, but that didn't work.

Tuesday September 10

After spending nearly the entire class period doing research into why I couldn't install Spotipy using pip, I finally figured it out thanks to Stack Exchange: Spotipy was being installed to the Python 2.7 folder on my computer, but since I use Python 3.7, my IDE couldn't find the library and assumed it had never been installed. I worked around this by using a specific command in my terminal that had pip install Spotipy to my Python 3.7 folder. Now I have access to the Spotipy library!

Thursday September 12

I started learning how to use Spotipy. After reading a lot of the library's documentation, I figured out how to create objects and prove authentication via my app client. After tinkering unsuccessfully for some time, I finally managed to retrieve audio feature analysis for individual tracks and learned how to get artist-specific data such as popularity and what genres of music the artist produces. At first I thought that I had to input the artist's actual name as a string, but it turns out that each artist, track, and playlist on Spotify has a unique identifier string known as a URI (uniform resource identifier), which is what the Spotipy functions take as input in order to retrieve data from an artist's profile or audio features from a song.

```python
#Victoria Agrinya
#Last update: 9/12/19

import spotipy
from spotipy.oauth2 import SpotifyClientCredentials


manager = SpotifyClientCredentials("8f408e92f7d24929ae7ac2613ebc11dc", "59dcf725c5494f7a8b31672ff4d46655")
vic = spotipy.Spotify(client_credentials_manager = manager)
id = "spotify:track:2nMeu6UenVvwUktBCpLMK9"
di = vic.audio_features(id)
print (di)
```

Figure 1: Feature extraction using a song's unique Spotify URI.

Timeline

Date	Goal	Met
8/26/19-9/30/19	Learn about CS Research class, procedures, and requirements	Yes
9/2/19-9/6/19	Create an EchoNest developer account and acquire authentication tokens to use their song analysis features	No, but learned that several EchoNest developer tools had been acquired by Spotify and all EchoNest song features can be accessed through the Spotify Web API
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	In progress
9/23/19-9/27/19	Build a logistic curve that takes one specific feature 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)	In progress

Reflection

This week, I found a super useful library that will tremendously benefit my work. It took me a long time to figure out how to install it, but I was able to in the end. I also had to play around with it in order to figure out what sort of data I had to input to get song and artist-specific data, but again, I figured that out. My next step is to figure out how to retrieve data from a playlist, since I'll be using features of songs on the Billboard Hot 100 playlist as metrics for my predictions.