CS 410: Text Information Systems
Course Project Progress Report: **Spotify Mood Ring**

Current Progress

As of today, we have completed the discovery phase of our project. There were a few different components involved within our discovery period, which included research and analysis of the various tools and technologies we are going to use.

Each of us did our own research into Chrome Extension development, including setting up our development environments, and looking into the different programming languages we would need to use like Python and JavaScript. Then, we got into researching the API documentation for the key APIs we'd be using. Specifically, we assessed what our use of the following technologies would be:

Spotify API: We spent several hours reading through the API documentation given for the Spotify for Developers Web API. We found several endpoints that will be off use to us in terms of fetching relevant data:

- GET /me/player/recently-played - Get a users recently played tracks
    - This endpoint allows us to specify a limit on the number of tracks returned as well as select a specific time frame
- GET /me/player/queue - Get the list of objects that make up the user's queue
- GET /me/player/currently-playing - Get the object currently being played on the user's Spotify account

Genius API: In addition to researching the Spotify API, we also looked into the Genius API methods we could utilize in conjunction with Spotify. It was discovered that the Genius API does not provide a way to download song lyrics themselves, so instead we looked into a tool called BeautifulSoup.

- GET /search - the search capability covers all content hosted on Genius (all songs). This allows us to search documents hosted on Genius and retrieve a particular song API path.
- GET /songs/{id} - we use the result of the search to fetch the song record from Genius and get the URL for where the lyrics are hosted
- BeautifulSoup: in order to get the song lyrics themselves we have to perform some web scraping. We have to send the webpage we retrieve from the Genius API into this BeautifulSoup tool to retrieve the text data corresponding to the lyrics of the song.

After researching all of the APIs that we were going to utilize for text data retrieval, we composed a plan for how each of the above endpoints that we gathered will be used and which tools we would need to perform the sentiment analysis on the retrieved lyrics.

NLTK (VADER): This Python library allows us to do text processing and sentiment analysis on the lyrics of the songs we retrieve. We use the Sentiment Intensity Analyzer class in the library to evaluate a string, with the function polarity_scores. The function returns a decimal between -1 and 1, with negative numbers being considered “bad” and one being considered “good”.

PyScript: This is a framework that allows Python code to be used within HTML. By utilizing this framework, we will be able to run Python code for our extension. This framework has a Cython powered interpreter called Pyodide, which has many Python libraries installed, especially those related to data science. This allows us to leverage the packages BeautifulSoup and NLTK with no difficulties. We also can interact with APIs by using the pyfetch package in PyScript.

Remaining Work

In terms of the work remaining, we have the developmental side of our project to implement. This entails pulling information on the current track from the Spotify API and then pulling url information for the current song via the Genius API. Using the url we received via the genius API, we will perform web scraping using BeautifulSoup to get the actual lyrics, which will be used for our sentiment analysis.

The next step would be the integration of the sentiment analysis tool, the NLTK library VADER. This is a pretrained model and the NLTK library is supported by the interpreter used in the JavaScript wrapper we will use for this Python code in the extension, PyScript. We will develop the extension after this, integrating our Python code via PyScript. Finally, we would need to deploy the extension, after thorough testing.

Once all developmental and documentation work, which will be done concurrently with development, has been completed, we will move to the tutorial presentation stage of the project.

Challenges/Issues

Some challenges that we came across include the following:

- Deciding on what kind of sentiment analytics libraries to use was also a challenge, given how many different ones there are right now.
- There may be some challenges we face with integrating Python code into a Javascript based Chrome extension. We will have to ensure that our sentiment analysis code can integrate smoothly as a Chrome extension.
- There was limited scope to what we could achieve given the time constraint of the project, so we decided against building the recommendation system that we had thought about adding initially.