Inspiration

Many times, it is very difficult to pick a song that is perfect for your current mood. Especially during such an unprecedented time in our world, it is hard for people to understand how they feel, let alone understand how to feel better. A plethora of research studies have proven that music is truly one of the best ways to brighten someone's day and MoodMatch aims at doing just that. This issue of not being able to understand one's true emotions and grasp what is needed to acquire a more positive mindset inspired me to create MoodMatch.

What it does

MoodMatch is a Facebook Messenger ChatBot that recommends songs based on a user's current mood. MoodMatch understands the user's current emotions by asking several personalized questions and using novel sentimental analysis algorithms to deliver several songs that will cheer up the mood of the user.

How I built it

Before this project, I was very new to the ideas of creating a project. This project introduced me to the end-to-end programming pipeline. I started by first mapping out what I had to do:
1. Obtain a list of songs, artists, and lyrics of the songs
2. Perform sentimental analysis on the songs to understand what kinds of feelings are exhibited by the song
3. Create a messenger bot
4. Program the messenger bot to ask the user questions
5. Process the results (sentimental analysis on the chosen choices)
6. Recommend songs with similar sentiment levels or perhaps higher sentiment levels to cheer one up

I followed this process thoroughly and used the Microsoft Azure Cognitive Science API to perform sentimental analysis. I also used various APIs including Spotify, Genius Lyrics, and various others.
Originally, I had meant for MoodMatch to pop up a Messenger Webview and encourage users to fill out a form that would then allow for the processing of data, however, Facebook's innovative Quick Replies module on Messenger allowed me to make MoodMatch more interactive and personalized for users.

Challenges I ran into

As I was very new to the end-to-end programming process, I ran into many challenges including not being able to understand what was wrong. My first challenge in this journey of completing MoodMatch was deploying my messenger bot on a Heroku server. Many times, I received an Error 400 or Error 500 stating that the /GET request cannot be found. However, once I got over this hump by fixing a few lines in my code, I was faced with several other tough mountains:

1. Obtaining results from the HTML form and using ExpressJS to deliver POST requests
2. Returning a list of songs and executing my algorithm on an ASYNC function
3. Transitioning from a bland messenger webview to more interactive Quick Replies

I often asked friends or even posted on StackOverflow to get some of my questions answered and get over these challenges. This project truly wouldn’t have been possible without others who helped me get over these challenges.

Accomplishments that I'm proud of

Creating a bot was not something I could've imagined me doing a few months ago, but now I have fully learned one example of the end-to-end programming pipeline. Getting over the aforementioned challenges and not giving up throughout the process has not only given me more technical knowledge, but it has impacted me mentally as I am more confident in my coding ability after this project.

What I learned

I fully learned the end to end programming pipeline of sorts. In particular, I learned the ins and outs of the various APIs I worked with and understanding how to deploy/manage server-based applications like a messenger chatbot.

What's next for MoodMatch

A lot is in store for MoodMatch, ranging from automating the song updates as well as adding more factors that may influence someone's choice of song. I plan on having an autorefresh of sorts for the list of songs already scraped off the web right now so that users will continuously get different songs. Perhaps even employing an account scenario in which users are never recommended the same song on multiple occasions is also one thing I plan to work on. Lastly, a lot of factors go into one’s feelings and current emotions, which definitely cannot be grasped by the current 2 questions asked my MoodMatch. I plan on adding more of these questions and even an open-ended question of sorts, in which users can type in their response, and emotional analysis will be performed on their inputs. In fact, a few of these are already in the works so be on the lookout for upcoming updates ;)