Shao-Min Yeh

Project Proposal

CS 410: Text Information Systems

Spotify Feature Analysis App

1. Team: Shao-Min Yeh (snyeh2@illinois.edu) - Spotify Feature App.
    a. I'm working alone due to taking this last semester remotely. Furthermore, I'm recruiting extensively and will not have much outside time, so I'm doing this project individually to maximize my time usage while making a worthwhile application.
2. Track: Free Topics. I will be doing analysis among songs from a Spotify Dataset. More specifically, I will make a Flask web application that consists of 3 different models a user can choose from:
    a. TF-IDF Search Engine: This model takes a user-submitted query and outputs the 50 most similar songs based on the TF-IDF weighting of lyrics, title, and artist name. Within this, the user can submit different weightings of the three categories (such as to focus on one category specifically) to further optimize their results. The purpose of this model is to aid the searching process of the user for many of their needs (searching for a song, looking at an artists' catalog, finding songs with certain sayings, etc), which is a major pain point in having many songs on Spotify.
    b. Song Similarity: This model takes a user-chosen song and outputs the most similar songs based on cosine similarity. Furthermore, this utilizes TF-IDF of the lyrics and more specific song features (key, energy, modeness, tempo, etc) found

in the dataset. With this, the user can also separate these two weightings or combine them. The purpose of this model is to allow users to find similar songs based on a multitude of different features, which should allow users to find more desired songs.

c. Lyrics Sentiment Analysis: This model takes a user-chosen song and outputs the sentiment analysis (valence) based on that song. More specifically, this outputs the emotion level based on the TF-IDF of lyrics and musical features. This is done by randomly assigning train/test/validation sets and training the model based on the “valence” feature, which measures the positivity of a song from 0.0-1.0 (and can be manipulated as our relevance rating). The purpose of this model is to understand important features to the mood of a song while aiding the user in searching for a song’s valence.

3. Programming Languages:

   a. Backend: I am using python (metapy and sklearn for analysis, pandas and numpy for utility, and flask for the application).

   b. Web application:, I am using HTML template files for pages.

4. Work Allocation:

   a. Preprocessing Data (config, csv, dat, etc files): 1 hour

   b. Search Engine Methods: 6 hours

   c. Song Similarity Methods: 5 hours

   d. Sentiment Analysis Methods: 5 hours

   e. Application Development and Incorporation: 3 hours

   f. HTML Web Pages: 2 hours

g. Tuning Parameters: 1 hour

h. Internal Testing: 1 hour

i. README Documentation: 2 hours

j. Total: 26 hours