October 13th, 2024:

* Finished exploring the dataset with the Demo.ipynb notebook
* Started working on the main project with the first step is to clean the raw data and turn it into a formal format.
* Understood the essence of the content-based filtering approach.
* Obstacles:
  + With the demo, I realized that the dataset I chose wasn’t appropriate for our approach, content-based filtering, since we only have one playlist to compare the content to. Therefore, I need to use the offered raw\_data.csv in the Github.
  + This is also a note to future work: the Github repo offers everything you need to follow the instructions and complete the project, trying to avoid create new data or tool by your own if it doesn’t serve the learning purposes (it would just take time).
* Future:
  + Spend time on the Extracting features document and Finish the preprocessing step.

October 14th, 2024

* Produced a processed dataset.
* Cleaned and chose useful features from it.
  + dedup by creating an unique column from song name and song artist
  + learned how to use column\_name.apply(lambda x: x\_do\_sth)
  + there is a different between .apply() and list comprehension.
* Collaborative Filtering (CF) | content-based CF:
  + Recommend songs based on the overlap of songs in playlists in the dataset.
  + In the context of song prediction, we would look at look the similarities in the songs of each playlist and recommend a song in one playlist if the similarity of songs is high with another playlist and that song is not in the other playlist. => This would limit the songs recommended within one playlist.
  + There are other ways for CF, such as item-item CF and model-based CF.

A diagram of a system pipeline

Description automatically generated

* Since TF-IDF calculates the importance of words based on the popularity and prevalence of words among a corpus of songs, we can use TF-IDF values as weights for different genres. Genres that are popular among many songs will hold less value and genres with less appearance will hold more value (IDF). TFs are relatively useless in this case because a genre only appears one time for each song, so TF for every word in a song is always 1.
* Obstacles:
  + 429 Error (SpotifyException: http status: 429, code:-1) | indicates that I’ve reached Spotify’s rate limit. The HTTP status code 429 means "Too Many Requests." This happens when too many API requests are made in a short period of time. => This is why you need to divide the dataframe into 3 parts and run 3 for-loops separatedly.
    - I reached the rate limit and have been hold for a while. To avoid wasting time, I would just use the processed\_data from GitHub.
  + Files are too large, cannot push to GitHub.
    - Apply “git config http.postBuffer 524288000” to increase the buffer to 500MB.
* Future:
  + Sentiment Analysis.

October 15th, 2024

* Finished content-based filtering, successfully generated a list of recommended songs using consine\_similarity from sklearn.
* Obstacles:
  + Don’t understand how the content-based filtering code works.
* Future:
  + Read thru the article and understand the issue.
  + Double-check if the code (for the last part) is correct.
  + Explore K-means