Alex Swenson and Vaibhav Guglani
Prof. Rachlin
DS 4300
17 June 2020

Overview

Description:

For our project, we built a recommendation engine for music based on Vab's spotify library. The data collection is presented in the 'data_collection' file, the analysis is in the 'analysis' file, and the machine learning and actual recommendation is done in the 'analysis_file'. Libraries used include pymongo, pandas, numpy, requests, datetime, urllib.parse, base64, sys, pprint, spotipy, matplotlib, and sklearn.

Data Acquisition, Cleaning, and Storage:

To collect data, we created a spotify developer account in order to get credentials for authorization to access the spotify API. For retrieving data like artist, track, and album information, we created a spotify client based on this youtube video: https://www.youtube.com/watch?v=xdq6Gz33khQ. As Vab did not have authorization for personal data, we used the spotipy library to access Vab's personal spotipy data. We could have used spotipy for everything, but we built a separate client to customize it to the needs of the project. Additionally, queries into the spotify API retrieved heavily nested structures that we had to clean in order to retrieve meaningful data. We handled a lot of the cleaning in the client itself.

First, we retrieved Vab's library, and all relevant audio features such as energy, danceability, and loudness and stored it in a dictionary of dictionaries for easy import to mongodb. Then we created a list of potential recommendations upon which we would later run our classifier, and retrieved the same features. To create a list of potential recommendations, we went through the top tracks of our top artists, their related artists, and new releases. We made sure the tracks we added to these potential recommendations were not included in Vab's library previously.

Now, we needed to collect data on songs Vab disliked. To do this, we found a kaggle dataset on hits and flops, and we included data for songs after 2000 as we thought those were more relevant to Vab, and most people. We assumed that Vab disliked all 'flopped' songs and considered those to be our 'disliked' songs on which we would train our classifier. We removed any songs from this dataset that included artists in Vab's library, to account for some flops Vab may have liked.

We then dumped all this data into mongo. We created a separate spotify database, and created separate collections in Mongo for each dataset, for ease of import back into Python.

Machine learning and actual recommender function:

The data we stored in Mongo was retrieved here and stored into dataframes for machine learning purposes.

We made a classifier that would go through our potential recommendations and return songs it predicted to us. After testing various classifiers, some tuning and preprocessing, we got best results with the DecisionTreeClassifier with an accuracy of a little over 80%.

While it looks like a good accuracy rate, it could be misleading, as the way our dataset was arranged, Vab 'disliked' about 80% of the music, and the classifier could be 80% accurate music by just rejecting all recommendations. However this was not the case, as we used another measure, the AUCROC, which measures the probability that the classifier would predict a true positive instead of a false positive and a true negative over a false negative. Our AUC was 0.71, which meant that there was a 71% chance of this being true. The model recommended about 250 songs to me from about 800 potential recommendations. The most important features were instrumentalness and danceability, according to the model.

Analysis:

For conducting some analysis on our results, we first created a correlation matrix for our features to see if it could impact our model. The only major correlation seemed to be between loudness and energy, and the model reflected that too, as in its most important features, those two metrics were right next to each other in rank.

We wanted to compare Vab's library to music in general, so on kaggle we found a dataset of 136000 songs, and used that as a standard. We created z scores for Vab's library, flops, and hits, and plotted them to compare them to each other. Vab's music seemed to be similar to hits, and more danceable than the standard (which was 0 for each feature on the plot).

Finally, we compared this to the songs actually recommended by the algorithm to see if the model recommended music similar to the library. As shown in the plot, this was the case.