Capstone project documentation

Yu-Chien Huang

June 6, 2019

1 Business objective

Recommend songs in an album that the singer should invest the money on to promote under limited budget.

2 About the ML model

Problem: two-class classification problem (a song will be labeled as hit or not hit)

- model: XGBClassifier
- data set: selected 669 albums of either best-selling or with a hit song of the year from 2014-2018. There are in total 8776 tracks in the 669 albums
- data source: I scraped the Billboard best-selling albums and albums with a hit song from 2014-2018. With the album and artist names, I went to spotify API to obtain audio features and popularity scores of the songs
- features: mode, tempo, duration, ordering, acousticness, danceability, energy, liveness, speachiness, valence
- labels: I labeled a track as "hit" (positive) if it's popularity is greater than the albums' overall popularity (popularity scores are provided by the spotify database.)

Some technicalities:

This is a unbalanced class problem (only 751 out of the 8776 tracks are hit in the labeling system.) I wrote a customized precision evaluation metric (the default one only had a precision score 0.2) and used the StratifiedKFold cross-validation in the GridSearchCV to train the XGBClassifier. I managed to have a precision score of 0.79 on the test set. [1]

As it is less likely that there would be any hit song predicted (tracks have to perform better than the album) for a given album, it would not be useful if most of time the model returns nothing. So I also used the probability prediction before turned into a 0/1 label to select the top three tracks in an the album.

[1] Please see the detailed code hosted in my Github repo. Here is the link: MLmodel.ipynb.

3 About the Web App

The web app link: https://stark-mesa-45314.herokuapp.com/predictor. [2]

- Query table: it takes the name of an album and the artist names (note that if the album is not in the spotify data base, nothing will happen; so also be careful about typos or unwanted spaces)
- After hit the submit button,
    1. the link to the the album in Spotify shows and plays the music automatically if you have a Spotify account
    2. the tracks that are predicted as a hit will be shown under *Hit song prediction*; this entry might be empy if no songs are predicted to be in the positive class
    3. The three tracks with the top three probability prediction scores are shown under *Predicted top three in the order of scores*
    4. Finally, the audio features used for the prediction are printed in a table under *Audio Features and popularity scores from Spotify database*, and the feature values are colored to help visualization

[2] Please see the code for the web app in my Github repo: link.