Exploratory Data Analysis

The million playlist data contains 1000 csv files, each of them includes information of 1000 playlists. Each row represents a single song. Playlist id (which playlist it is in), position id (index of this song in the playlist), track name, track uri (potentially useful for scraping), artist name, artist uri, album name, album uri, and duration (in ms) are used to describe a song. The whole dataset is very large (about 10G), so we select a subset (first 100 out of 1000 csv files) to do EDA.

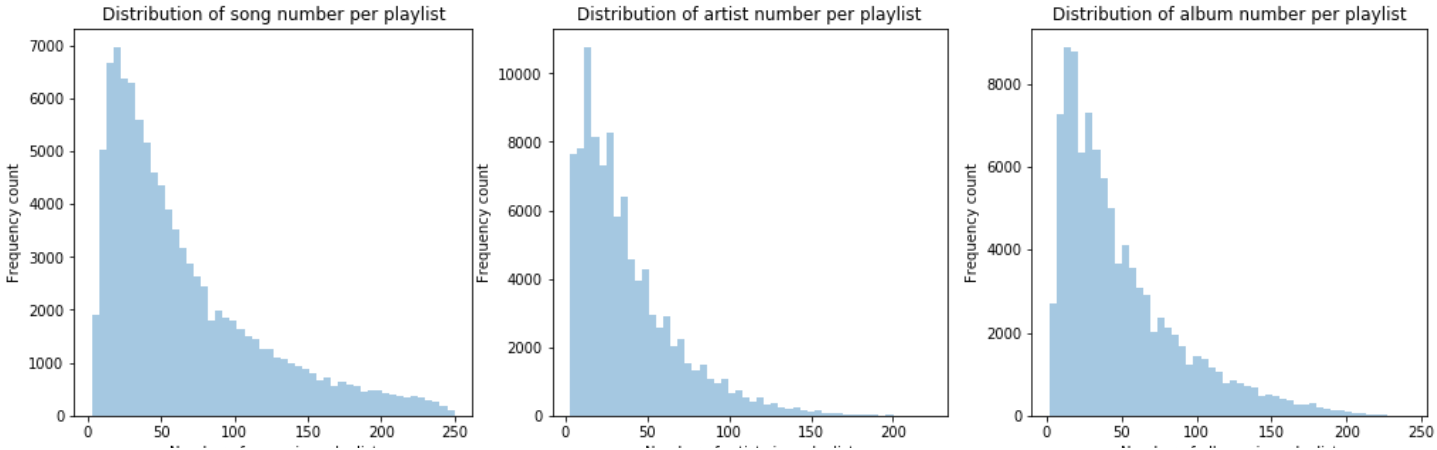
In these 100 csv files, with 100000 playlists, we see 6677800 tracks records. However, there are cases that the exact same song could appear in the same playlist for more than one time. After accounting for this, we see that there are 6589079 actual records, and there are 65.89079 songs on average per playlist. Similarly, we see that there are 38.12169 different artists involved on average per playlist, and 49.78987 different albums involved on average per playlist.

	Number of songs per playlist	Number of artists per playlist	Number of albums per playlist
count	100000.000000	100000.000000	100000.000000
mean	65.890790	38.121690	49.789870
std	52.849538	30.171495	39.888158
min	3.000000	3.000000	2.000000
25%	26.000000	16.000000	20.000000
50%	49.000000	30.000000	37.000000
75%	91.000000	52.000000	68.000000
max	250.000000	222.000000	242.000000

Over these 100000 playlists, there are 681805 different songs, 110063 different artists and 271413 different albums covered. Next, we explore the distribution of frequency.

The distributions are similarly right-skewed. There are few playlists that are unreasonably long (up to 200 songs).

We are curious about what songs and which artists are most popular. To better distinguish songs with the same name, we put track name and artist name together.



As shown in Supplemen 1 and 2, we see that the TOP 50 songs belong to 39 different artists, and 25 of them rank TOP 50 as artist. Among the TOP 50 songs, 36 of them belong to TOP 50 artists. The popularity of songs and corresponding artists are highly correlated.

Baseline model

Our baseline model was built using collaborative filtering. Specifically, we recommend songs to a user/existing playlist based on other similar playlists. When building our baseline model, we used the JSON version of the million playlist dataset, provided by the TF, instead of the CSV version. Even though the JSON version is substantially larger in size, we decided to use the JSON version because it contains the playlist name and number of followers, which could provide information about the vibe/genre of the playlist as a whole. This is useful when we build content based model during the next step. Additionally, a playlist with a large number of followers might be a signal that songs in this playlist should be recommended to others more frequently.

Due to lack of computational power and time constraints, the baseline model only utilized 50 JSON files, which include 50,000 playlists, more than 400,000 unique songs. 80% were used for training and the rest were used for validating.

We first constructed the interaction matrix, or utility matrix. For each element, 1 indicates a song is in a playlist and 0/blank indicates a song not in the matrix. Since on average there are around 50 songs in a playlist, it is a very sparse matrix. After matrix factorization, using a package called Spotlight, we were able to fill in the blanks in the matrix by giving predictions based on existing interactions.

After the full utility matrix is determined, for each playlist in the validation set, we randomly selected 70% songs for prediction (X) and 30% as the ground truth (y). We compared how similar this playlist is to each playlist in the utility matrix using cosine similarity. Then we were able to calculate the weighted score of each song in the dataset and recommend the top 250 scoring songs to this playlist. For evaluation, we used R-Precision: for the same amount of songs in the prediction as in the ground truth, we calculated how many songs in the prediction are also in the ground truth. After evaluating 200 playlists in the validation set, our baseline model scored 0.0079, which ranks 100 in the Spotify RecSys Challenge 2018.

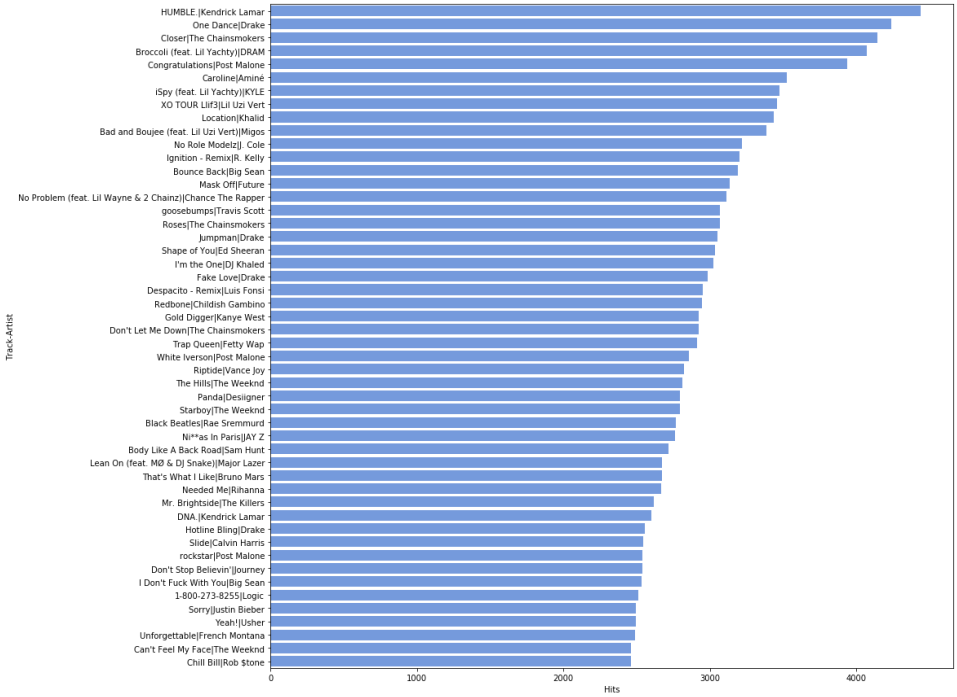
Revised Project Goal

In this century of information explosion, thanks to the music streaming services provided by Spotify, Pandora and etc., we now have access to tens of millions of digital songs online. However, one problem posed by the rapidly increasing amount of digital music is that it becomes harder for users to screen out the desirable songs from the massive music library. Music recommender systems are thus developed to automatically searching the music library and recommend the songs users might like based on the users' music listening history. The two most prevalent methods used in recommender systems are collaborative filtering method and

content-based filtering method. The idea of collaborative filtering method is to recommend to a user the songs that are liked by the users with similar taste in music. While in content-based filtering, songs are recommended to a user according to similar features of the music within that user's playlist. Thus, collaborative filtering method suggests similar songs on a user behavioral basis, and content-based recommends similar songs on the basis of song properties. In our project, we want to apply both methods to our million playlist dataset and compare how the performance of the two methods differ in predicting the songs each user might like. And we want to figure out a way to combine these two methods to make more appropriate music recommendations.

Supplement

Supplement 1: The top 100 hits



Supplement 2: The top 100 artists