Capstone II Report
Truc Tran

Spotify Music Recommendation System

1. Data

The dataset contains 1,000,000 playlists, including playlist titles and track titles, created by users on the Spotify platform between January 2010 and October 2017. Each playlist contains some tracks from 5 to 250 of them. The dataset includes playlist information such as playlist name, pid, number of albums… and track information such as track name, artist name, track_uri, duration..... A single file that holds many JSON files of Spotify playlists and its track information and can be downloaded through ALcrowd website https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge/dataset_files

Note: A playlist can contain duplicate tracks, and one track can be included in many playlists.

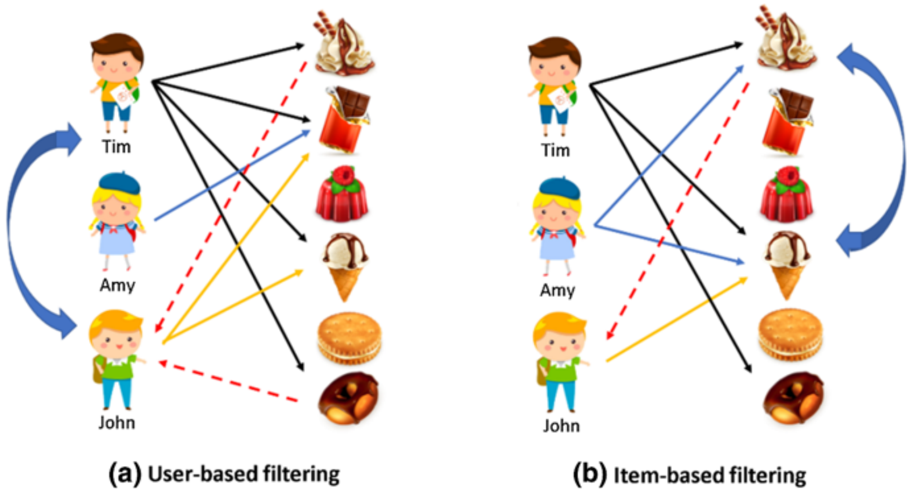
2. Method

The two popular methods for recommendation engines are:

1. Collaborative filtering models:

This recommendation system relies on the user's dataset, and item feedback includes explicit feedback and implicit feedback. The explicit feedback is more accurate than implicit feedback that contains the user's interest, rankings such as star rating, and implicit data from user behavior without ratings or detail needed. There are three algorithms:

Algorithms	Recommendation
User-User	recommend items based on similar users liking it
Item-Item	Recommend items that are similar to the one the user had used before or similar
User-Item	Recommend based on the combination of both approaches and matrix factorization techniques.



Picture Source: https://predictivehacks.com/item-based-collaborative-filtering-in-python/

2. Content-based models:

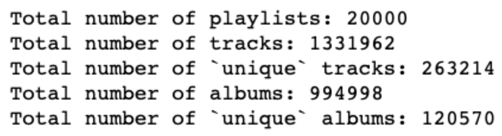
   This technique uses similarities in features to make decisions that are based on user preferences for product features.

3. Data Cleaning

   The Spotify dataset is an organized dataset that already shuffles the playlist and assigns each playlist to an identification number. Most of the dataset is clean. The only missing column is description (all NaN values). So, I only need to drop the description column.

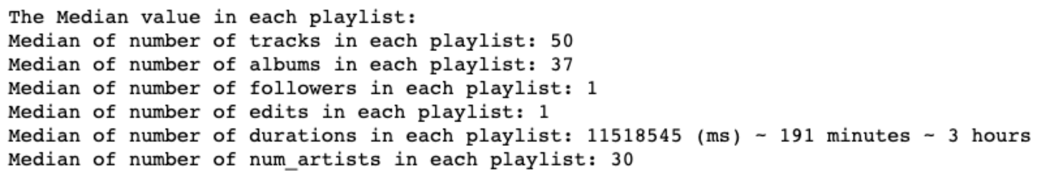
4. EDA

Dataset Information

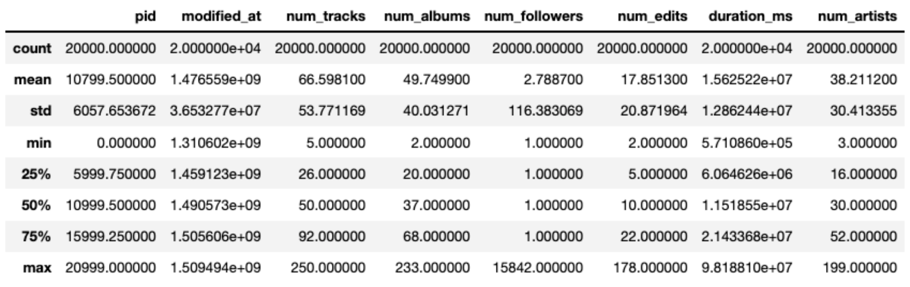


```
Total number of playlists: 20000
Total number of tracks: 1331962
Total number of `unique` tracks: 263214
Total number of albums: 994998
Total number of `unique` albums: 120570
```

Statistical Features

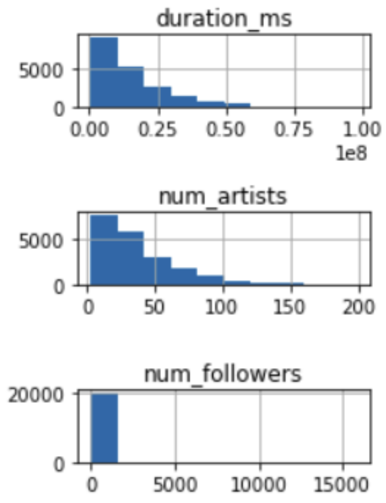


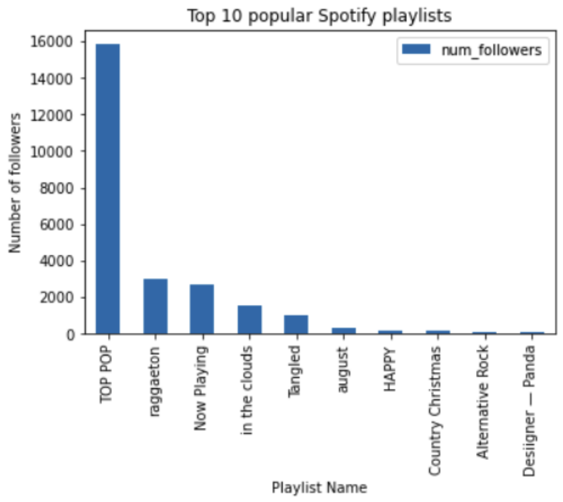
```
The Median value in each playlist:
Median of number of tracks in each playlist: 50
Median of number of albums in each playlist: 37
Median of number of followers in each playlist: 1
Median of number of edits in each playlist: 1
Median of number of durations in each playlist: 11518545 (ms) ~ 191 minutes ~ 3 hours
Median of number of num_artists in each playlist: 30
```

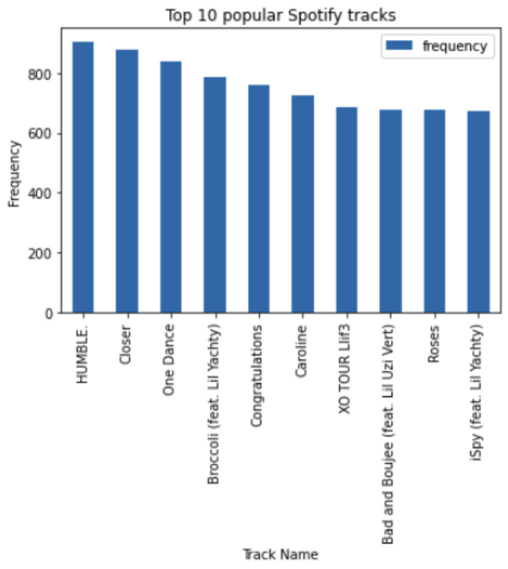


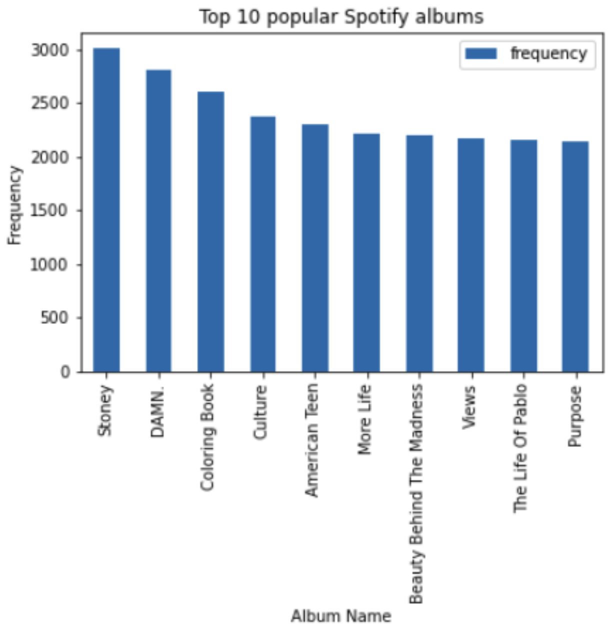
	pid	modified_at	num_tracks	num_albums	num_followers	num_edits	duration_ms	num_artists
count	20000.000000	2.000000e+04	20000.000000	20000.000000	20000.000000	20000.000000	2.000000e+04	20000.000000
mean	10799.500000	1.476559e+09	66.598100	49.749900	2.788700	17.851300	1.562522e+07	38.211200
std	6057.653672	3.653277e+07	53.771169	40.031271	116.383069	20.871964	1.286244e+07	30.413355
min	0.000000	1.310602e+09	5.000000	2.000000	1.000000	2.000000	5.710860e+05	3.000000
25%	5999.750000	1.459123e+09	26.000000	20.000000	1.000000	5.000000	6.064626e+06	16.000000
50%	10999.500000	1.490573e+09	50.000000	37.000000	1.000000	10.000000	1.151855e+07	30.000000
75%	15999.250000	1.505606e+09	92.000000	68.000000	1.000000	22.000000	2.143368e+07	52.000000
max	20999.000000	1.509494e+09	250.000000	233.000000	15842.000000	178.000000	9.818810e+07	199.000000

Distribution of


duration_ms
num_albums
num_artists
num_edits
num_followers
num_tracks
1e8


Top 10 popular Spotify playlists
num_followers
Number of followers
16000
14000
12000
10000
8000
6000
4000
2000
0
TOP POP
raggaeton
Now Playing
in the clouds
Tangled
august
HAPPY
Country Christmas
Alternative Rock
Desiigner — Panda
Playlist Name


Top 10 popular Spotify tracks
frequency
Frequency
800
600
400
200
0
HUMBLE.
Closer
One Dance
Broccoli (feat. Lil Yachty)
Congratulations
Caroline
XO TOUR Llif3
Bad and Boujee (feat. Lil Uzi Vert)
Roses
iSpy (feat. Lil Yachty)
Track Name



5. Algorithm & Machine Learning

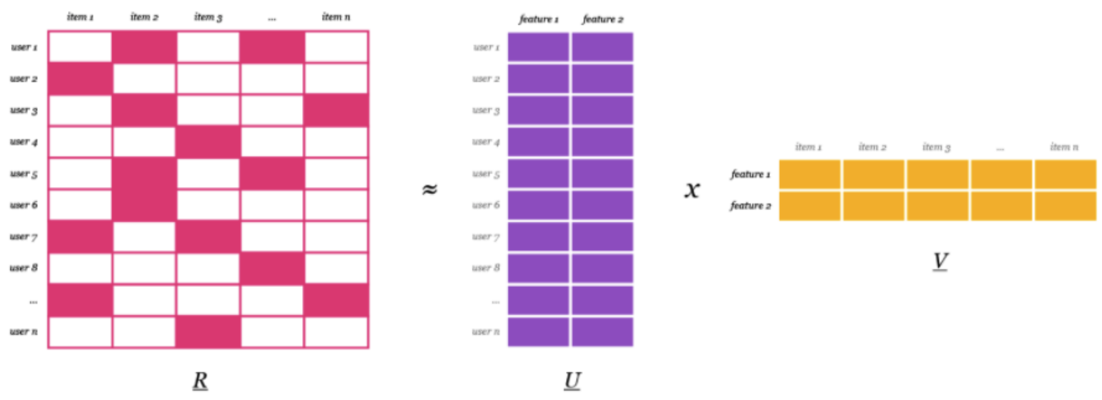
Modeling:

1. Alternative Least Squares

This recommendation model is based on the ALS algorithm that computes a matrix factorization of the user-item matrix. Here the playlist is treated as a user, and each song is treated as an item. The Alternating Least Square is built for large-scale collaborative filtering problems. We are dealing with all missing data in our sparse matrix with implicit data of music playlists from Spotify users. It is an optimization solution for dealing with the spareness of rating data, and it is simple and easy to scale on large datasets.

Starting with the original data of playlists and track R, ALS finds the other two matrices U and V, to make the equation ***R ≈ U x V*** perfect. U and V are used to calculate similarity scores for items and make recommendations based on those scores.

Similar items:	score = $V \,.\, V_i^T$
Making Recommendations:	score = $U_i \,.\, V_{\blacksquare}^T$



R $_{u\,x\,i}$: playlists and tracks matrix
U $_{u\,x\,f}$ users and hidden features matrix
V $_{f\,x}$ I items and hidden features matrix

Source: https://medium.com/radon-dev/als-implicit-collaborative-filtering-5ed653ba39fe

2. Logistic Matrix Factorization
   Another collaborative filtering recommender model for implicit data is Logistic Matrix Factorization (LMF), which works similarly to the ALS model. Both models perform matrix factorization, with LMF using a slightly different factorization algorithm. True to its name, LMF borrows some ideas from logistic regression and uses a different loss metric than ALS.

3. Item-based filtering
   Cosine similarity is used because data is sparse, and many ratings are undefined.
   To recommend music to playlists, we want to recommend similar songs and are likely to appear in the same playlists. Cosine similarity gives a number range [0, 1], indicating how similar two songs are. 0.0 means the songs are in entirely different playlists, 1.0 means the songs are included in all the same playlists.

   - song-song similarity we take column as a vector

	Song #1	Song #2	Song #3	……….	Song #n

Playlist #1	✓		✓	....	
Playlist #2				....	✓
Playlist #3	✓	✓	✓	....	✓
....	....	....	....	....	....
Playlist #m	✓	✓		....	

*** ‘0’ means the song is not in the playlist, ‘1’ indicates the song is in that playlist

Metric Evaluation & Results:

The Spotify music recommendation system is following metrics to evaluate the models based on Spotify ResSys rules:

- G and the ordered list of recommended tracks by R.
- The size of a set or list is denoted by |·|, and we use from:to-subscripts to index a list.

R-PRECISION

R-precision is the number of retrieved relevant tracks divided by the number of known relevant tracks (i.e., the number of withheld tracks):

$$\text{R-precision} = \frac{|G \cap R_{1:|G|}|}{|G|}$$

The metric is averaged across all playlists in the challenge set. This metric rewards total number of retrieved relevant tracks (regardless of order).

RECOMMENDED SONGS CLICKS

Recommended Songs is a Spotify feature that, given a set of tracks in a playlist, recommends 10 tracks to add to the playlist. The list can be refreshed to produce 10 more tracks. Recommended Songs clicks is the number of refreshes needed before a relevant track is encountered. It is calculated as follows:

$$\text{clicks} = \left\lfloor \frac{\arg\min_i\{R_i : R_i \in G|\} - 1}{10} \right\rfloor$$

If the metric does not exist (i.e. if there are no relevant tracks in R, a value of 51 is picked (which is 1 greater than the maximum number of clicks possible).

NORMALIZED DISCOUNTED CUMULATIVE GAIN (NDCG)

Discounted Cumulative Gain (DCG) measures the ranking quality of the recommended tracks, increasing when relevant tracks are placed higher in the list. Normalized DCG (NDCG) is determined by calculating the DCG and dividing it by the ideal DCG in which the recommended tracks are perfectly ranked:

$$DCG = rel_1 + \sum_{i=2}^{|R|} \frac{rel_i}{\log_2 i}$$

The ideal DCG or IDCG is, in our case, equal to:

$$IDCG = 1 + \sum_{i=2}^{|G \cap R|} \frac{1}{\log_2 i}$$

If the size of the set intersection of G and R is empty, then the IDCG is equal to 0. The NDCG metric is now calculated as:

$$NDCG = \frac{DCG}{IDCG}$$

Model	**R-Precision**	**Recommended Song Clicks**	**NDCG**	**Run time**
1. Item-based Filtering (Cosine Similarity)	0.0424	38.55	0.0366	82000 s
2. Alternative Least Square Matrix Factorization	0.0056	47.8523	0.0497	110 s
3. Logistic Matrix Factorization	0.0055	47.9375	0.0521	120 s

* NDCG indicates out of recommendation songs, how close the recommendation songs are at the beginning of the hidden list

6. Prediction

   Based on the size of the original playlist, the recommendation will produce a list of k songs where k = test_size * 15. This Spotify music recommendation system will take a playlist id and the portion of its tracks and return prediction on the continuous songs of the playlist. The recommended songs will return the list of songs that are likely to appear on the playlist in descending order.

7. Future Improvement
   In the future, I would spend more time optimizing the run time for Item-based Filtering because this model gives better results, but it is not scalable. In addition, I will measure the NCDG value for this model.

   I want to try other methods or models, use Spotify track dataset on audio features, and other songs to see if it returns better results or improves the run time.

8. Credits
   I want to thank Spotify for providing their user music dataset, an excellent resource for education purposes, and the AIcrowd website to host the challenge. I would especially like to thank my Springboard mentor Kevin Glen for advising this recommendation system project and fantastic support.