Music Recommendation System

By Truc Tran

Outline

- Overview and Context
- Who cares?
- Popular methods for recommendation engine
  - Collaborative Filtering
- Explore Data Analysis
- Modeling
  - Cosine Similarity
  - Matrix Factorization
- Evaluation Metrics
- Business Suggestions

Overview and Context

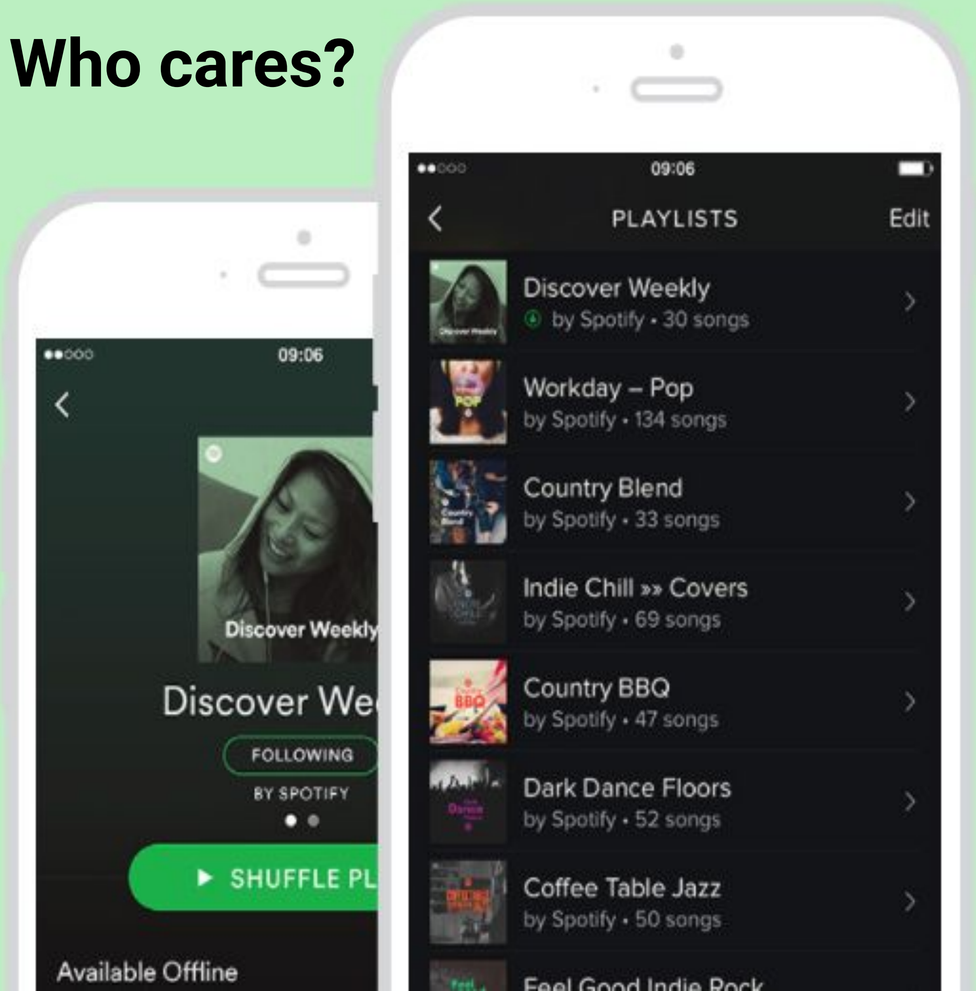
Nowadays music is an essential part of people's lives. Streaming services deliver that music, but they also present more song choices than anyone can sort through. Recommendation systems help users narrow down these choices, and find their future favorites.

Our goal: develop a competitive music recommendation system, to fill user playlists with relevant songs, and improve user experience and retention.

Who cares?

Music Streaming Services

Who cares?

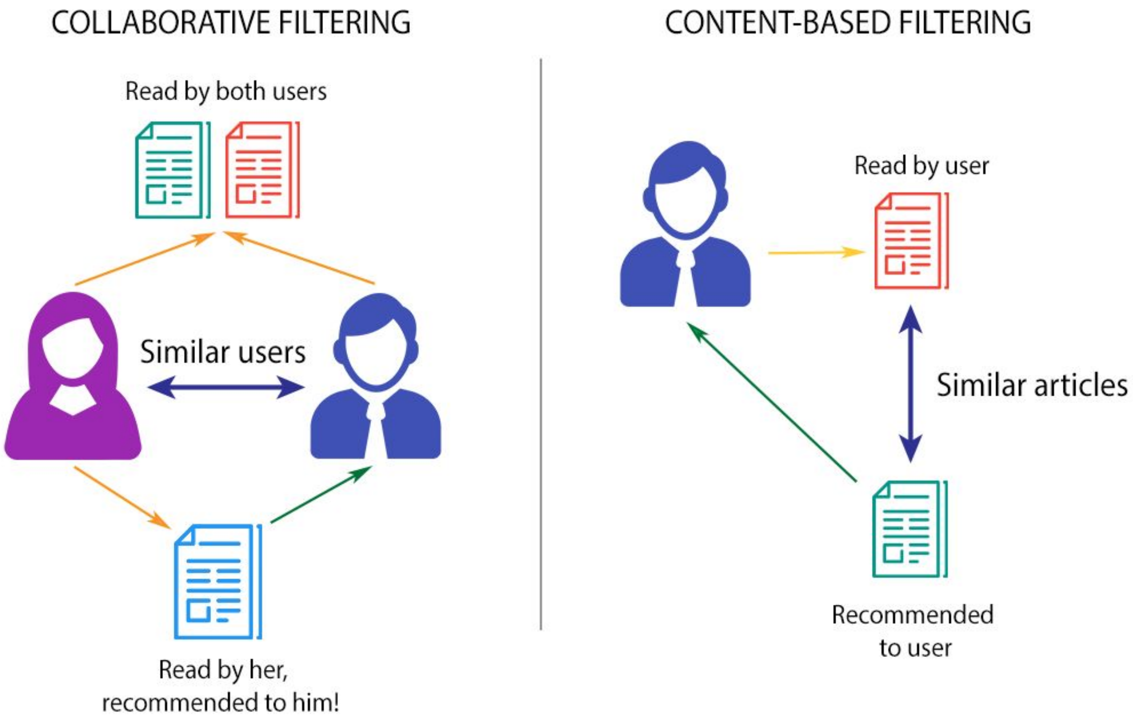


You care

It’s all yours!

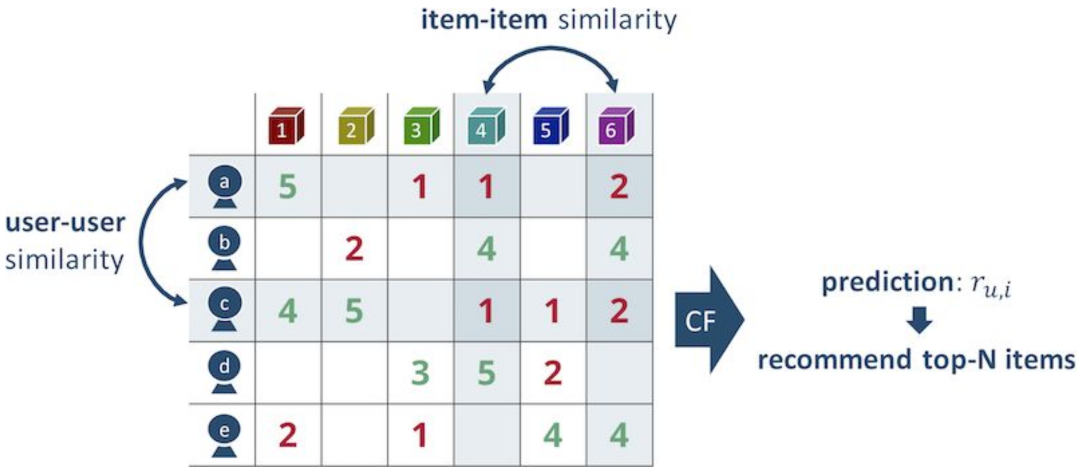
Your playlist is crafted just for **you**, based on the music **you** already love

Popular methods for recommendation engine



Collaborative Filtering

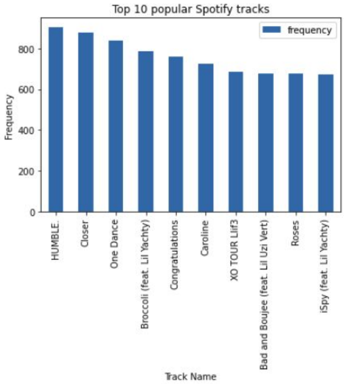
1. User-User: recommend items based on similar users liking it.
2. Item-Item: recommend items that similar to the one the user had used before.
3. User-Item: recommend items based on a combination of both approaches, or matrix factorization techniques.

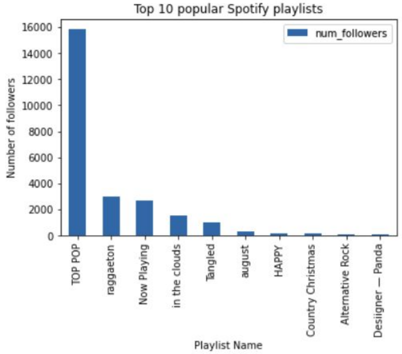


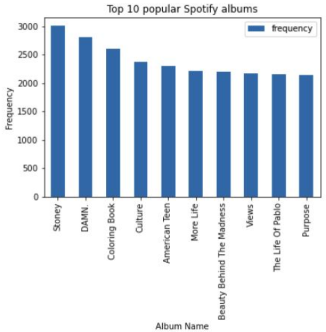
User-Item Matrix

Explore Data Analysis

- Out of 20,000 playlists, there are 1.3 million tracks, but only 260k unique tracks.
- In average, each playlist is around 3 hours, and has 50 songs from 30 artists.
- The most popular track is “HUMBLE”, the most popular playlist is “TOP POP”, and the most popular album is “Stoney”.







Modeling

Two types of models:

- Cosine similarity, where new tracks are discovered and ranked based on mutual inclusion in playlists.
- Matrix factorization, which builds a user-item matrix
    - Alternating Least Squares (ALS)
    - Logistic Matrix Factorization (LMF)

Modeling: Cosine Similarity

New tracks are discovered and ranked based on mutual inclusion in playlists.

This technique gives:

- A real number range [0, 1] indicating how similar two songs are.
- 0.0 means the songs are in completely different playlists, 1.0 means the songs are included in all the same playlists.

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

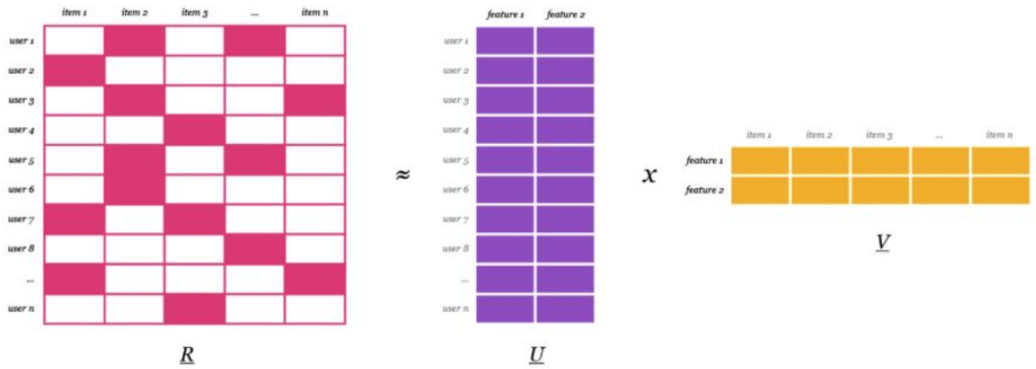
*** '0' means the song is not in the playlist, '1' indicates the song is in that playlist

*** song-song similarity we take column as a vector

Modeling: Matrix Factorization

Matrix Factorization finds two matrices U and V in order to make the equation ***R ≈ U x V*** to be as perfect as possible.

Based on the similarity scores for items, the system can make recommendations.



$R_{u\,x\,i}$: playlists and tracks matrix

$U_{u\,x\,f}$ users and hidden features matrix

$V_{f\,x\,i}$ items and hidden features matrix

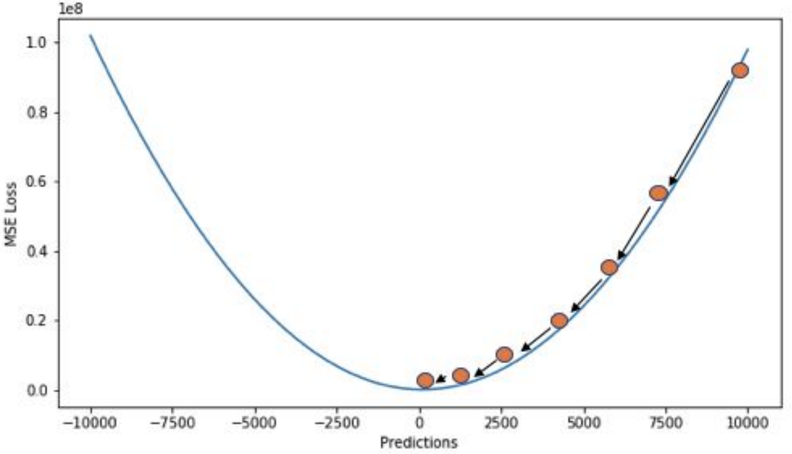
Similar items: score = $V \cdot V_i^T$

Making Recommendations: score = $U_i \cdot V^T$

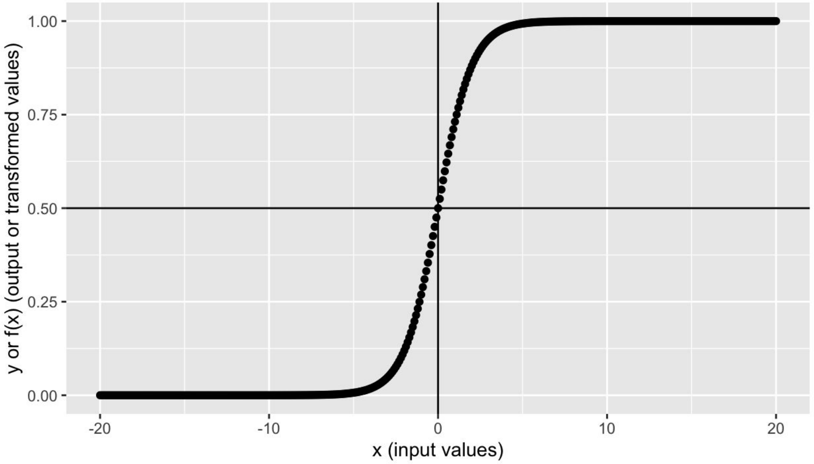
ALS vs LMF

ALS uses Mean Squared Error as its loss function, while LMF uses a logistic function that represents probabilities of tracks appearing in the playlist.

Mean Squared Error Function



Logistic (sigmoid) function



Evaluation Metrics

- **R-Precision**: this metric rewards total number of retrieved relevant tracks (regardless of the order)

- **Recommendation Song Clicks**: is the number of refreshes needed before a relevant tracks is encountered

- **Normalized Discounted Cumulative Gain (NDCG)**: indicates out of recommendation songs, how close the recommendation songs are in the beginning of the hidden list

Results

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

Business Suggestions

Although cosine similarity has better results, matrix factorization is 100 to 1,000 times faster.

- I suggest streaming services use matrix factorization methods for real time music recommendations, on user home screens.
- Businesses with strong data processing capacity can implement cosine similarity to refine recommendations over time, or when accuracy is a priority, such as marketing campaigns.


THANK YOU