Song Recommendation System

DWAYNE MCFARLANE

JAN 24 2021

Context--Why Recommender Systems?

As technology gradually plays a more pronounced role in the lives of everyday citizens, companies are continuously on the hunt for methods that will allow them to optimize the user experience on their respective platforms. Major organizations such as Facebook, Amazon, Spotify, Apple Music, Youtube and many others increase their revenues by increasing the amount of time on and engagement interactions with their digital platforms. One popular type of system that helps these organizations and many others improve the user experience is called a recommendation system. A recommendation system is a set of various algorithms that are used to recommend relevant products or services to interested users.

Examples of Recommendation Systems

NETFLIX

Popularity

amazon

collaborative filtering

Problem Statement & Objective

Problem Statement

Build a recommendation system to propose the top 10 songs for a user based on the likelihood of listening to those songs.

Objective

Explore and utilize various libraries and data science concepts to build a recommendation system that best achieves the goals stated in the problem statement. We will utilize data science methods of Collaborative filtering and content-based methods to build a recommendation system that is suitable for various users. In doing so, we hope to have several business benefits that would increase user interaction, user engagement and potential revenue increase.

DataSet & Limitations

The core data is the Taste Profile Subset released by The Echo Nest as part of the Million Song

Dataset. There are two files in this dataset. One contains the details about the song id, titles, release, artist name and the year of release. Second file contains the user id, song id and the play count of users.

File – song_data

1. song_id - A unique id given to every song
2. title - Title of the song
3. Release - Name of the released album
4. Artist_name - Name of the artist
5. year - Year of release

File - count_data

1. user _id - A unique id given to the user
2. song_id - A unique id given to the song

Limitations:

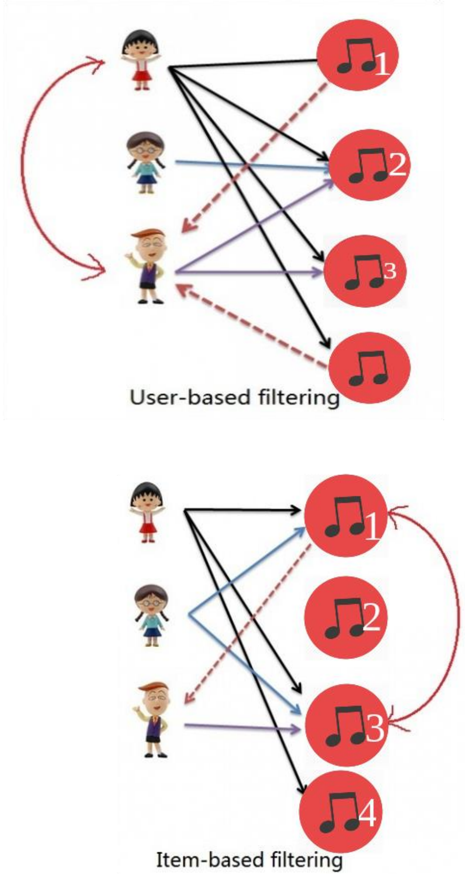
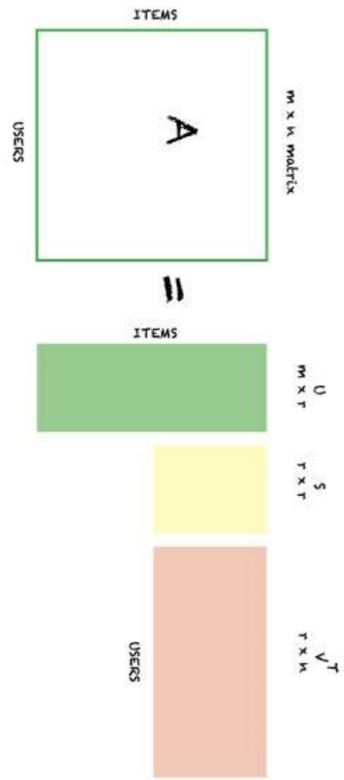
One of the issues that exists in this dataset is the fact that we do not have explicit ratings data to utilize in our recommendations. What we do have is number of times a song is played. While this is great for popularity based recommendation systems, this poses some hurdles for the more user-centric collaborative filtering methods. We will have to transform the data in a way that would make it applicable to the Nearest neighbor and Single Value Decomposition factorized matrix.

Data Collection Methods

Implicit – Bahavior Driven

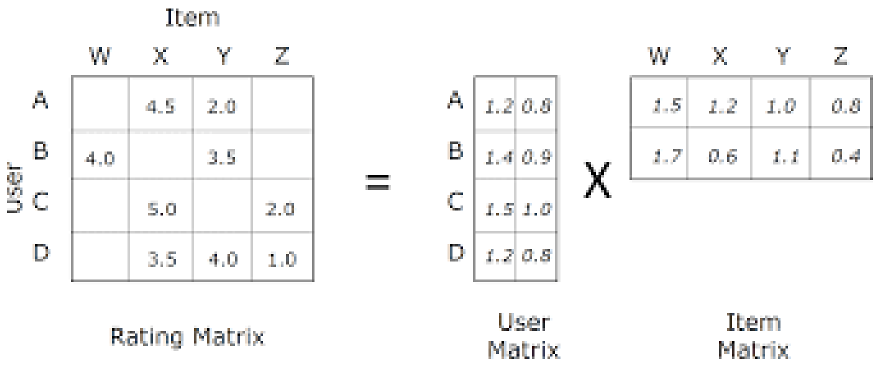
Explicit

Types of Recommender Systems

Popularity	Classification Model	NN Collab Filtering	Matrix Factor CF
- Most played songs - No personalized Recommendations -User Features can be used -Song Features can be used - Very scalable	-Uses Features of both Items and Users -Information is not high quality -Not scalable		

Matrix Factorization

For matrix A that contains the data for n users x m songs. This matrix can be decomposed uniquely into 3 matrices; let’s called them **U**, **S**, and **V**.



In terms of our song recommender system:

- **U** is an *n* users x *r* user-latent feature matrix

- **V** is an *m* songs x *r* song-latent feature matrix

- **S** is an *r x r non-negative* diagonal matrix containing the singular values of the original matrix.

A singular value represents the importance of a specific feature in predicting a user preference.

A *Diagonal matrix* is a matrix in which the entries outside the main diagonal are all zero.

Why SVD?

Reduces the number of features of a dataset

Allows us to better identify important and relevant features

Saves on storage and processing costs for extremely large dataset/user base

Next Steps

Try various other methods and tests to evaluate different recommendation system approaches and how they each affect operations and business goals and revenues.

A/B testing

Latency testing

Recall vs Coverages

Offline /Online

KPI, CTR, ROI, ROR