Recommendation Systems

Music Recommendation Systems

Milestone 2: Refined Insights, Techniques' Comparison, Final Solution Design

Adriana C Vazquez | March 2022

REFINED DATA INSIGHTS

Problem formulation: To build a recommender system that recommends songs based on SONG data and USER - PLAY COUNT data.
Some helpful insights are outlined below

- **We have vast amount of data** 76k unique users and 10k unique songs, which would mean if every user played every song we'd have over 760M entries.
- We can use some of the attributes in song data, in particular **play counts** to use as indicator of **ratings**. Intuitively, songs with higher play count are better liked by a user.
- We can see across songs, which ones are the most **played on average** or do a summation across users to find which songs have the highest **play frequency**
- We can also use song meta data, e.g. name of artists, and song titles
- In building our recommendation systems, it helps that we can do some sanity checks based on the nature of the data, e.g. see if songs with play counts > 1 are recommended to a user, check if recommendations seem to fall within what the user could like based on genres or other similar users, etc.

comparison of techniques

We built several recommendation systems and tuned hyperparameters. results are summarized below.

TABLE GUIDE:

- User-User Similarity-Based Collaborative Filtering (UUCF)
- Item-Item Similarity-Based Collaborative Filtering (IICF)
- Model Based Collaborative Filtering - Matrix Factorization (MSVD)
- Cluster & Content Based Recommendation System (CB)

	UUCF	IICF	MSVD	CB
Untuned	**RMSE: 1.0707** **Precision: 0.403** **Recall: 0.711** **F_1 score: 0.514**	**RMSE: 1.0212** **Precision: 0.327** **Recall: 0.436** **F_1 score: 0.374**	**RMSE: 0.9929** **Precision: 0.428** **Recall: 0.655** **F_1 score: 0.518**	**RMSE: 1.0382** **Precision: 0.397** **Recall: 0.584** **F_1 score: 0.473**
Tuned	**RMSE: 1.0053** **Precision: 0.435** **Recall: 0.771** **F_1 score: 0.556**	**RMSE: 0.9912** **Precision: 0.455** **Recall: 0.583** **F_1 score: 0.511**	**RMSE: 0.9753** **Precision: 0.442** **Recall: 0.642** **F_1 score: 0.524**	**RMSE: 1.0491** **Precision: 0.39** **Recall: 0.565** **F_1 score: 0.461**

scope to improve: for each model, we did basic hyper parameter tuning. For the best performing, it would be recommendable to look into the options more in depth (highlighted) and also look into **hybrid** model options.

understanding

Proposal for final solution

we propose to further tune the user - user similarity collaborative filtering model and explore a hybrid model with SVD

We will explore that in the subsequent stages and ultimately see which model has best performance based on f-1 and RMSE metrics.

pros and cons

User based collaborative filtering: even though it performed best in our data, scalability and sparsity are the primary issues and something to keep in mind if this were to be implemented in a production scenario.

We would like to explore a hybrid model to see if we can improve the best model's performance, but this may mean losing a bit of the interpretability of the model.
However, a very intuitive model may not always result in the best for this particular business case, (e.g. content based although it will recommend songs from the same artist that a user likes, it will not help the user discover new songs that he/she truly may not know).