Executive Summary

In this problem of developing a song recommendation system, it was observed that 2 of the collaborative filtering methods namely- User-User Similarity and Optimized SVD (Matrix Factorization) methods have offered the best possible performance. Precision of 44.7% indicates that out of recommended songs, 44% are heard by the user. Similarly, with Optimized SVD models- When it ensures 43.4% precision means on an average, this recommendation system is good enough to drive 43.4% of the business.

Initially, I had proposed 3 various ways in which play_count data can be considered.

1. Keeping only play_ount values between 1-5
2. Keeping play_count values which have occurred for more than one time (this approach gave maximum F1-score, recall and Precision)
3. Applying min-max scaler to play_count

Finally, the final recommendation system created is a hybrid recommendation system created by generating recommendations in following ratio:

- 2 songs from Popularity Based Recommendation
- 4 songs from User-User Similarity Based Recommendation
- 4 songs from Optimized SVD model based Recommendation

Algorithm	**Cut-Off**	**RMSE**	**Precision**	**F_1 Score**
User User Similarity Based	1.5	4.1043	0.447	0.584
Optimized SVD Model	2	3.9342	0.434	0.586

A user’s song taste may change over time. As a result it would be a good idea to include time in some or the other way while recommending songs to the users. Additionally, there are other algorithms applicable for recommendation systems, as mentioned below, which can be implemented to enhance the performance and personalization effect of this recommendation system.

1. Non-Negative Matrix Factorization
2. Neural Collaborative Filtering
3. Graph Neural Networks

Further, on integration of such a recommendation system on the company’s web platform/ application, it would be mandatory to perform A/B testing which would ensure, whether this selected model for recommendation system is performing well or not.

In A/B testing, a group of users is provided with type ‘A’ recommendation system (existing) where as remaining group of users is provided with type ‘B’ recommendation systems (newly developed).

After that, various user activities like time spent by users using the app/ website, or total play time per day, and so on can be tracked for each group of users to make a decision whether this new recommendation system is performing better than the previous one. If yes, then it is deployed at a full scale for all the users.

Problem and Solution Summary

As expected, performance of Matrix Factorization methods works well, and provides a good F1-score for this type of dataset. The biggest challenge in this dataset was to interpret the play_count column and use it as an interaction between users and songs. In reality, a user may listen to a song once and may or may not like it. So a play_count of 1 does not indicate user has liked the song today. At the same time, there may come incidences where tomorrow the same user, might listen to this song while hanging out with friends, and by listening to the song again and again, starts liking it. Then he would come back to the app and listen to the song. In such a case, time plays a very important role. A particular recommendation may be making sense today but not tomorrow. The solution provided here however, does not cover time interactivity while generating recommendations, but it would be good to have time related data.

In the solution provided, personalization is covered to some extent by the User-User similarity method as well as SVD model. User-User similarity recommends songs based on other similar users taste. At the same time, SVD model attempts to capture hidden latent features like genre of the song, i.e. attempts to match a song of particular genre with a user who likes that genre. We may not be able to identify the genre of song/ users choice, but SVD model implicitly works in providing relevant recommendations.

The need of recommendation system arises from the fact that companies need to recommend songs and generate more revenue, as well as users need appropriate songs in their recommendation list. A good recommendation system should be a balance of both. Naturally, relevant recommendations to the user, will ensure user spending more time using the website/ application. As a result, the given solution combining recommendation of most popular songs, along with personalized recommendation would be a great solution. Approximately 40-50% of business should be driven by recommendation system.

Recommendations for Implementation

This being a sparse data matrix with a large number of users and songs available on the website/ app, would require lot of computational power. A cloud based deployment service like AWS or GCP would be recommended. The cost of investing in such platforms versus the revenue generated by such a recommendation system must be compared. This is an experimental design of recommendation system which may or may not turn out to be very great. Other business in competition, may come up with newer and newer techniques for recommendation systems. This is one of the fundamental risks, beyond the control of our business. Challenges include, less data available for users. This can be addressed by initiating a short survey with the users asking them their choice of songs. More the data available for recommendation systems, more better is their performance. In order to solve this problem one step ahead, user data such as age, gender, geographic location may also be captured, as these factors play a key role in shaping human choice.