Business Report for Music Recommendation System

Kojen Wang 9th June 2022

Executive Summary

This project proposes a hybrid model that combines a tuned user-user similarity-based model and a tuned item-item similarity-based model for the prediction of the top 10 songs for each user on Spotify. It is suggested that the model should only consider data that has users who have listened to at least 90 songs, songs listened by at least 120 users, and songs with no more than 5 play counts. With the sufficient set of observed data, we suggest a robust and trustworthy recommender system that has achieved high accuracy and low error with the test data. The hybrid model we propose provides real-time and offline recommendations to satisfy both the new and returning users' needs, solving the cold-start problem in collaborative filtering. It is recommended that stakeholders consider experimenting and segmenting the user type on the platform before implementing the model. We also suggest stakeholders collect more data points, such as the timestamps of user activities, which greatly helps with the model evaluation and long-term improvement.

Problem Summary

Spotify has put a great emphasis on providing quality recommendations, which helps the company earn nearly 90 % of the revenue from 182 million subscribers. These users rely on Spotify's recommender system for music discovery, and personalized songs and playlists. The record labels and the music professionals also benefit from the music recommender system. The record labels can promote new artists through recommendation sessions. It is shown that over one-third of new artists were discovered this way on Spotify. Furthermore, music professionals can take advantage of music recommendations to understand what listeners prefer and how to compete in the growing market. This helps them strategize around release campaigns, connecting with new audiences, and amplifying the ad's budgets.

The objective of this project is to build a music recommendation system that proposes the top 10 songs for each Spotify user based on the likelihood of listening to those songs. We will take advantage of different models to provide customized and relevant recommendations to users, helping the company enhance listeners' day-to-day experience.

Solution Design

We have explored various models and methods while building recommender systems, including rank-based models, content-based models, user-user similarity-based collaborative learning, item-item similarity-based collaborative learning, model-based collaborative filtering, and cluster-based model. The final proposed solution is the hybrid model that blends the tuned user-user similarity-based and the tuned item-item similarity-based models.

The below figure shows the performance of the hybrid model, which yields the highest values of Precision, Recall, and F1 scores in comparison to other models. This suggests that the hybrid recommender system has been the best-performing model.

Algorithms / Metrics	**User-user Tuned**	**Item-item Tuned**	**SVD Tuned**	**Cluster Tuned**	**Hybrid (User-User Tuned + Item-item Tuned)**
Precision	0.448	0.464	0.419	0.406	**0.467**
Recall	0.29	0.286	0.247	0.234	**0.302**
F1 Score	0.352	0.354	0.311	0.297	**0.367**

Note: For the hybrid model, we define k = 5 for tuned user-user model, and k = 5 for tuned item-item model.

This model is very effective in predicting the top 10 songs that are relevant to different types of users. For new users, we provide real-time recommendations using tuned item-item similarity-based collaborative learning, which do not suffer from the cold-start problem when new users come to the platform. For returning users, we already have enough historical data to customize recommendations, so our model can generate recommendations offline using tuned user-user similarity-based collaborative learning. This ensures us providing more personalized content to the users.

Analysis and Key Insights

Comparing Different Models

We have investigated different models and compared their performance in the below table.

Algorithms / Metrics	**User-user Baseline**	**User-user Tuned**	**Item-item Baseline**	**Item-item Tuned**	**Model Based**	**Model Based Tuned**	**Cluster Based**	**Cluster Tuned**
Run Time	20.9 s		**21.5 s**		19.5 s		**8.19 s**	
RMSE	1.0919	1.0546	1.0919	1.0222	1.0295	1.0177	1.0611	1.0911
Precision	0.404	0.448	0.404	0.464	0.42	0.419	0.411	0.406
Recall	0.263	0.29	0.263	0.286	0.25	0.247	0.245	0.234
F-1 Score	0.319	0.352	0.319	**0.354**	0.313	0.311	0.307	**0.297**

We have observed these patterns:

- There is no significant difference between the RMSE, Precision, Recall, and F-1 score among these models, indicating that their model performance is very similar.
- The F1 score is the highest for the tuned item-item similarity-based model (0.354), followed by the tuned user-user similarity-based model (0.352). We can say that the tuned item-item similarity-based model is performing relatively better among all models.
- The tuned cluster-based model has the lowest F1 score (0.297).
- For the model run time, the item-item similarity-based baseline model has the longest run time (21.5 s). However, the cluster-based model has the shortest run time (8.19 s) but its overall model performance is very similar to other models.

From the above analysis, it is shown that there is no model that its performance dominates over the others. Due to this, we have explored if combining different algorithms would yield better performance in the following section.

Intersection Rate

We define the Intersection Rate to measure how similar the recommendations are provided by two different algorithms. From the below table, it is shown that the user-user baseline model and model-based learning have the highest similarity, while cluster-based learning and user-user baseline model, cluster-based learning and item-item baseline model, or model-based learning and item-item baseline model have the least similarity.

In general, different algorithms provide different lists of recommendations, since the average similarity is ~0.02. This indicates that when 10 songs are recommended, there are about 0.2

songs overlapping between two algorithms. This finding shows the potential of combining different algorithms to optimize the performance of recommender systems.

Algorithms / Algorithms	**User-user Baseline**	**Item-item Baseline**	**Model Based**	**Cluster Based**
User-user	1	0.03	**0.11**	**0.00**
Item-item	0.03	1	**0.00**	**0.00**
Model-based	**0.11**	**0.00**	1	0.02
Cluster-based	**0.00**	**0.00**	0.02	1

Note: The values are the intersection rates between two algorithms.

Evaluating Model Performance with Different Ks

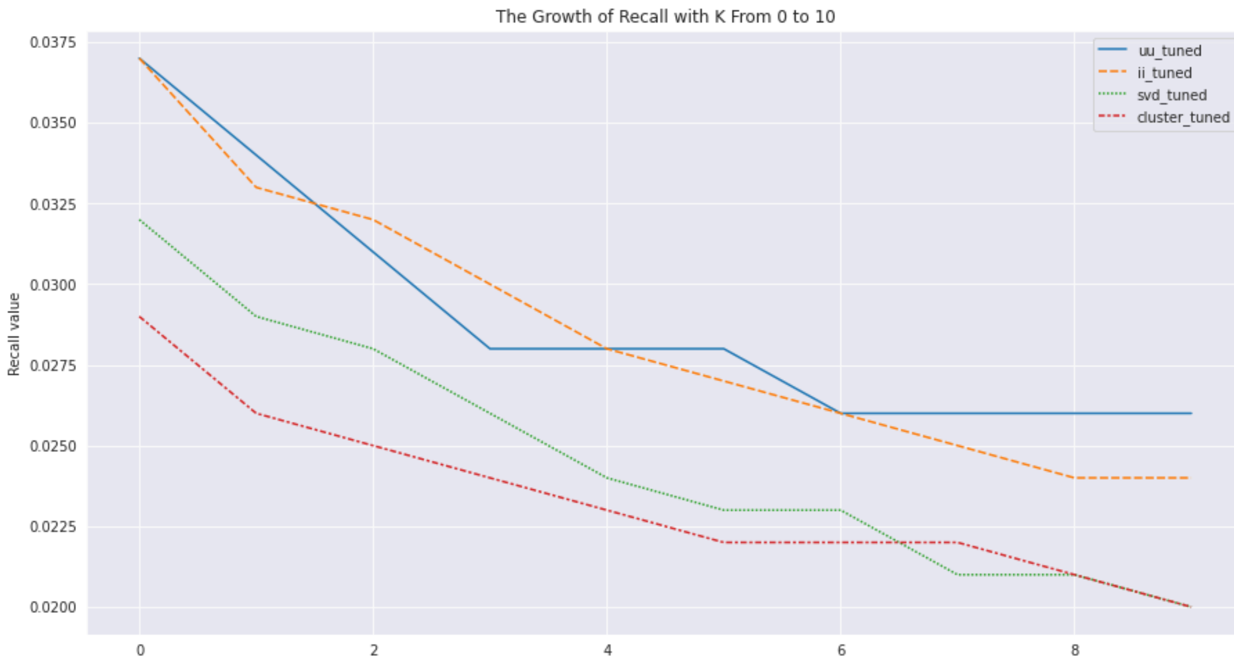
We have analyzed the Recall@K of different models with K from 1 to 10. From the below table, it is shown that the Recall of all these models slightly increases as the value of K increases.

K / Algos	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
User-user Tuned	0.037	0.071	0.102	0.13	0.158	0.186	0.212	0.238	0.264	0.29
Item-item Tuned	0.037	0.07	0.102	0.132	0.16	0.187	0.213	0.238	0.262	0.286
SVD Tuned	0.032	0.061	0.089	0.115	0.139	0.162	0.185	0.206	0.227	0.247
Cluster Tuned	0.029	0.055	0.08	0.104	0.127	0.149	0.171	0.193	0.214	0.234

Note: The values are the Recall@K of different algorithms.

Generally speaking, we can see from the line plot that the growth of Recall for each model has slightly decreased from 0.03 to 0.02 as the value of K increases, indicating that the higher the value of K, the less powerful the predictive model.

We take advantage of this finding and select models with the lower value of K in order to build a hybrid and powerful recommender system. Since the tuned user-user similarity-based model and tuned item-item similarity-based model have the highest overall values of Recall, we combine the two as our final solution.



Evaluating Model Performance with the Hybrid Model

We have calculated the RMSE, Precision, Recall, and F1 score of the hybrid model. From the below table, it is shown that the Precision, Recall, and F1 Score of the hybrid model have been the highest. This indicates that the model performance has greatly improved in comparison to other types of models. For this reason, we propose the hybrid model as the final solution, which is very effective in recommending the top 10 songs that are relevant to Spotify users.

Algorithms / Metrics	**User-user Tuned**	**Item-item Tuned**	**SVD Tuned**	**Cluster Tuned**	**Hybrid (User-User Tuned + Item-item Tuned)**
Precision	0.448	0.464	0.419	0.406	**0.467**
Recall	0.29	0.286	0.247	0.234	**0.302**
F1 Score	0.352	0.354	0.311	0.297	**0.367**

Note: For the hybrid model, we define k = 5 for tuned user-user model, and k = 5 for tuned item-item model.

Limitations and Recommendations for Further Analysis

We recommend stakeholders experiment and segment the user type into new users and returning users before implementing our solution. The hybrid model will provide real-time recommendations to new users and generate recommendations offline for returning users. This mechanism can satisfy both types of users, potentially convert new users into subscribers, and motivate subscribers to continue to pay for the service.

Furthermore, the proposed solution was trained based on historical data. When we implement the solution, it is challenging to analyze the quality of the recommendations without knowing the timestamps of the user's activities. We suggest collecting additional data points to help with further analysis, such as timestamps, music skip times, the number of times that a user has searched for songs, user preferences, and much more. This information can inform whether the user satisfaction, user usage, user retention, or the revenue has increased, helping the company strategize a long-term improvement plan for the recommender system.

The proposed solution might have a risk that it is not ready to be operationalizable due to the fact that the data pipeline has not been built yet. Before implementing the solution, it is suggested to design the machine learning pipeline to retrain models based on the newly observed data and its data update frequency. This will ensure that the recommender system continues producing results consistently, and presents the best songs to the new and returning users at the right time.

To further improve the recommender system, we also recommend fine-tuning the hybrid model by combining different models with different K values. This may help us find another good-performing model if any.

References

- How Spotify's Algorithm Works? A Complete Guide to Spotify Recommendation System [2022]: Music Tomorrow Blog https://www.music-tomorrow.com/blog/how-spotify-recommendation-system-works-a-complete-guide-2022#:~:text=Music%20professionals%20rely%20on%20recommender,them%20to%20amplify%20artist%20discovery.

- Inside Spotify's Content Recommendation Engine - Blog https://exchange.scale.com/public/blogs/inside-the-content-recommendation-engine-at-the-heart-of-spotify

- Why do you need a Real-Time Recommendation System? https://www.bizdata.com.au/blogpost.php?p=why-do-you-need-a-real-time-recommendation-system