The Investigation of Music Recommendation Systems

Shengqi Yu
Hangzhou Xuejun High School, Hangzhou, Zhejiang, China
may.zhao@shgl-link.com

Abstract. A recommendation system represents a machine learning technique that examines and presents user preferences derived from their online activities. In today's digital landscape, recommendation systems find widespread use in short video platforms. These systems scrutinize user actions such as search history and viewing patterns to proactively suggest videos aligned with individual tastes. Leveraging an effective recommendation system to propose high-quality videos tailored to user preferences proves advantageous in sustaining user engagement and boosting platform traffic. Nonetheless, users with specialized interests or newcomers to the platform may receive less accurate recommendations due to limited data availability. The development of an artificial intelligence-based recommendation system can significantly expedite product searches, enhance search precision, foster user loyalty, augment average order values, and bolster overall conversion rates. In the realm of artificial intelligence applications, recommendation systems have become indispensable components. This paper delves into the algorithms for automatically analyzing and recommending music based on users' historical playback records. The study employs a random forest model to train the recommendation algorithm.

Keywords: recommendation system, music, machine learning.

1. Introduction

A recommendation system is a machine learning technique that analyzes and displays user preferences based on their browsing history. Nowadays, recommendation systems are commonly used in short video applications, where they analyze user behaviors such as search history and viewing history to proactively recommend videos that align with user preferences. Using an effective recommendation system to suggest high-quality videos that match user preferences is beneficial for maintaining user engagement and increasing platform traffic. However, for users with niche interests or new users, the lack of sufficient data may lead to less accurate recommendations. Building an artificial intelligence recommendation system can significantly shorten product search time, improve search relevance, enhance user loyalty, enhance average order value, and increase overall conversion rates. Currently, recommendation systems are an essential module in the field of artificial intelligence applications.

In recent times, there has been a growing prevalence of intelligent algorithms. By harnessing the power of these algorithms, it becomes feasible to offer accurate feedback regarding particular user actions and suggest content that aligns with their interests. This research endeavors to delve into the algorithms utilized in recommendation systems by examining the historical playback data of music. The aim is to leverage this analysis to generate personalized recommendations for users, enhancing their music streaming experience.

2. Method

2.1. Dataset Introduction

songs.csv shown in Figure 1 contains data about songs. It includes song IDs, length, genre, composer, performer, and language codes. test.csv shown in Figure 2 contains a substantial amount of data regarding music playback by platform users. It includes user IDs, music IDs, the source of music resources, and the mode of playback. To facilitate analysis of individual users, the data has been filtered to retain only the playback data under a single user.

id	msno	song_id	source_syst	source_scre	source_type
232	//4hBneqk/	N2nNEISm	search	Search	album
234	//4hBneqk/	BUQwTuzZ	search	Search	album
553	//4hBneqk/	hymjSPprk:	search	Search	album
554	//4hBneqk/	ZmgY6NwI	search	Search	album
555	//4hBneqk/	d4oUZEZh:	search	Search	album
556	//4hBneqk/	Ag6LeaguF	search	Search	album
95272	//4hBneqk/	S4AINh7ko	search	Search	album
95273	//4hBneqk/	sVoiMDj0E	search	Search	album
95274	//4hBneqk/	ZP/XuyUoh	search	Search	album
98932	//4hBneqk/	emWCsOD	search	Search	album
170820	//4hBneqk/	ucvxpS8Bla	my library	Local playli	local-library
170821	//4hBneqk/	tPQs4Kqi4r	my library	Local playli	local-library
170822	//4hBneqk/	cJbwHYqah	my library	Local playli	local-library
170823	//4hBneqk/	eUDIAZSeT	my library	Local playli	local-library
170824	//4hBneqk/	Q4erUp5sz	my library	Local playli	local-library
170825	//4hBneqk/	fp6ii6/DRal	my library	Local playli	local-library
170826	//4hBneqk/	msJWKceA	my library	Local playli	local-library
170827	//4hBneqk/	9BNHcETI5	my library	Local playli	local-library

Figure 1. The sample data of songs.csv (Photo/Picture credit: Original).

2.2. Computing Similarity

2.2.1 Approach

The idea is to take a song from the historical record library as a sample and compare its various indicators with the song information in the music database. Assign a value of 1 to identical features and 0 to different features. Calculate a score for each song as the song similarity and recommend songs with higher similarity.

2.2.2 Code Implementation

1) Read the data and store the information of recommended songs in a dataframe called “rec”.

2) Traverse the song information table and calculate the similarity between each song and the sample song.

3) Record the song with the highest similarity for recommendation.

Problems and Algorithm Limitations

The weights of different indicators in preference calculation may vary, and it is not appropriate to assign the same weight to different indicators hastily. Comparing the similarity based on a single song from the historical records does not involve a comprehensive analysis of all the data, resulting in preference recommendations that lack universality.

3. Random Forest

Random Forest is a classifier that consists of multiple decision trees and can be used for both classification and regression problems, as well as dimensionality reduction tasks. It exhibits good tolerance towards outliers and noise and provides better prediction and classification performance compared to a single decision tree [1].

3.1. Random Sampling and Training Decision Trees

To generate a representative sample with a size of N, conduct N iterations of random sampling with replacement, extracting one sample each time. This process results in N diverse samples. Subsequently, these selected N samples are employed to train a decision tree, which serves as the root node of the decision tree [2-5].

3.2. Attributes for Node Splitting

When each sample comprises M attributes, at each node splitting point in the decision tree, a random selection of n attributes is made from the total M attributes. From this subset of n attributes, one specific attribute is chosen as the splitting attribute for the given node.

3.3. Repeating Step 2 Until no Further Splitting is Possible

The process of forming the decision tree involves splitting each node according to Step 2. This process continues until no further splitting is possible.

3.4. Building a Large Number of Decision Trees to Form a Forest

By following Steps 1 to 3, this study built a large number of decision trees, forming a random forest.

3.5. Code implementation steps

This process consists of many steps including: 1) Modules Loading 2) Config 3) Data Loading 4) Filling Null Values 5) Feature Selection 6) The output is a dataframe including the information of music it recommended. The results of the model can be found in Figure 3.

msno	city	bd	gender	registere(	registrat:	expiration_date
XQxgAYj3k	1	0		7	20110820	20170920
UizsfmJb9r	1	0		7	20150628	20170622
D8nEhsIOB9	1	0		4	20160411	20170712
mCuD+tZ1hI	1	0		9	20150906	20150907
q4HRBfVSs	1	0		4	20170126	20170613
zgPOEyUn5a	13	43	female	9	20120703	20171006
Sw9AT8QoR4	1	0		4	20160326	20160329
pg6bT2XZk9	1	0		7	20160130	20170930
kfk1AdTNH2	1	0		7	20110111	20170930
tscijwx4dl	1	0		7	20160217	20170613
GoFYKhcq8(	12	28	male	9	20060531	20170816
Hiw71XZQS(	1	0		7	20150909	20170909
V6QgFYHgs1	1	0		7	20120105	20170918
RoSfblbwJ1	15	33	female	3	20150915	20171019
N+dxmo1qvI	1	0		3	20140412	20170815
vUzJAyFEu(	13	20	male	3	20140410	20170427
Ev+ouoRTpA	13	30	male	9	20070330	20170612
/NFM1PP8F\	1	0		7	20121228	20170930

Figure 2. The result of the model (Photo/Picture credit: Original).

4. Discussion

4.1. Principles of Intelligent Algorithm-Based Recommendations

The process of building recommendation models involves associating labels or profiles with the objects being recommended. However, a challenge arises when dealing with newly registered users, as the system lacks sufficient data to understand their interests and preferences. This is known as the "cold start" problem in recommendation systems. To address this problem, various strategies can be employed. User clustering techniques can group similar users together based on limited available information, allowing the system to make recommendations based on patterns observed in the clustered groups. Classification methods can also predict the interests or preferences of new users based on demographic or registration information. Open programming interfaces can encourage user engagement and facilitate data collection, helping the system gather more reliable data and reduce the impact of the cold start problem [6-9].

4.1.1 User-Based Collaborative Filtering Model

User-based collaborative filtering is a technique commonly used in video promotion that relies heavily on user information. The approach involves gathering essential user details such as gender, address, age, and basic interests to create an initial user profile. During the process of distributing

content, the model uses the similarities between users to suggest relevant and similar content to each user. By identifying users with similar characteristics and preferences, the system can personalize recommendations that align with their individual preferences. The accuracy of the platform's predictions concerning user needs improves as the quality and depth of the user information provided increases. By incorporating more precise and detailed user information, the platform gains a better understanding of individual user preferences and can make precise recommendations based on their unique interests. Consequently, as the user information becomes more comprehensive, the platform becomes more proficient in predicting and fulfilling user needs, ultimately enhancing the overall user experience.

4.1.2 Short Video User Behavior Prediction Algorithm Based on Multi-task Learning and User Interest Change

This algorithm focuses on predicting user click-through rates for various interactions during short video viewing, such as reading comments, liking, clicking on profiles, and sharing. It recognizes that user interests in short videos can change over time, and accounting for this dynamic nature is crucial for accurate predictions. The algorithm builds upon the MMOE model by integrating a mechanism to consider user interest changes. It achieves this by incorporating sorted user historical behavior sequences as a corpus and training word embedding models using the Word2Vec algorithm. This approach allows the algorithm to gain insights into users' dynamic interests and effectively capture the fluctuations in their interests over time.

The algorithm combines two types of features to predict user behavior. Firstly, it generates statistical features through feature engineering, which provide relevant insights into user interactions. Additionally, it utilizes dynamic interest features generated through the word embedding model, which capture the evolving interests of users. Both sets of features are then fed into the MMOE model, which ultimately produces predictions regarding user behavior during short video viewing [10].

5. Conclusion

In conclusion, the creation of an artificial intelligence-driven recommendation system holds the potential to greatly accelerate product searches, improve search accuracy, cultivate user loyalty, increase average order values, and boost overall conversion rates. In the field of artificial intelligence applications, recommendation systems have evolved into essential components. This paper has delved into the intricacies of algorithms for the automated analysis and recommendation of music based on users' historical playback records, with the study utilizing a random forest model to train the recommendation algorithm.

References

[1] Fan H. Research on Random Forest Model Based on Grey Wolf Optimization. 2022(6):45-47

[2] Rigatti S J. Random Forest. Journal of Insurance Medicine, 2017, 47(1): 31-39.

[3] Biau G, Scornet E. A random forest guided tour. Test, 2016, 25: 197-227.

[4] Belgiu M, Drăguţ L. Random Forest in remote sensing: A review of applications and future directions. ISPRS journal of photogrammetry and remote sensing, 2016, 114: 24-31.

[5] Oshiro T M, Perez P S, Baranauskas J A. How many trees in a random forest? Machine Learning and Data Mining in Pattern Recognition: 8th International Conference, MLDM 2012, Berlin, Germany, July 13-20, 2012. Proceedings 8. Springer Berlin Heidelberg, 2012: 154-168.

[6] Ye K. Social influence and regulation of short video intelligent recommendation algorithm technology, 2021 (9), 28-3.

[7] Lan X Y. Analysis of Tiktok short video recommendation mode based on intelligent algorithm. Video Engineering, 20203 47 (6), 152-154, 158

[8] Xu J, etc. Research on collaborative filtering cold start recommendation algorithm based on item semantics. Small microcomputer system, 2021, 42 (11):2246-2251.

[9] Han S, et al. Cold Start Recommendation Algorithm Based on Fusion similarity and Hierarchical clustering. Microcomputer System, 2022, 43(5):985-991.

[10] Ma J, et al. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM Press, 2018: 1930-1939