PLOS ONE

RESEARCH ARTICLE

Knowledge distillation for multi-depth-model-fusion recommendation algorithm

Mingbao Yang[1], **Shaobo Li**[2]*, **Peng Zhou**[3]*, **JianJun Hu**[4]

1 State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang, China, **2** State Key Laboratory of Public Big Data, Guizhou University, Guiyang, China, **3** College of Mechanical Engineering, Guizhou University, Guiyang, China, **4** Department of Computer Science and Engineering, University of South Carolina, Columbia, South Carolina, United States of America

* lishaobo@gzu.edu.cn (SL); pzhou@gzu.edu.cn (PZ)

Check for updates

OPEN ACCESS

Citation: Yang M, Li S, Zhou P, Hu J (2022) Knowledge distillation for multi-depth-model-fusion recommendation algorithm. PLoS ONE 17(10): e0275955. https://doi.org/10.1371/journal.pone.0275955

Editor: Brij Bhooshan Gupta, Asia University, TAIWAN

Received: July 15, 2022

Accepted: September 27, 2022

Published: October 25, 2022

Copyright: © 2022 Yang et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All datasets can be accessed at https://grouplens.org/datasets/movielens.

Funding: This paper was supported by the National Key Research and Development Program of China under Grant 2020YFB1713300. In addtion, the recipient is Proffsor Shaobo Li. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Recommendation algorithms save a lot of valuable time for people to get the information they are interested in. However, the feature calculation and extraction process of each machine learning or deep learning recommendation algorithm are different, so how to obtain various features with different dimensions, i.e., how to integrate the advantages of each model and improve the model inference efficiency, becomes the focus of this paper. In this paper, a better deep learning model is obtained by integrating several cutting-edge deep learning models. Meanwhile, to make the integrated learning model converge better and faster, the parameters of the integrated module are initialized, constraints are imposed, and a new activation function is designed for better integration of the sub-models. Finally, the integrated large model is distilled for knowledge distillation, which greatly reduces the number of model parameters and improves the model inference efficiency.

1. Introduction

As human beings enter the information age, people are producing a large amount of information all the time, and they also spend a lot of time browsing information on the Internet. How to let users find the information that users are interested in from the massive Internet information has become an important topic of current research. Similarly, in the aviation equipment manufacturing industry, it is also faced with the problem of how to find better upstream service providers from the massive services. The aviation equipment manufacturing cloud service platform has attracted a large number of aviation equipment manufacturing service providers to settle in. How to allow users to obtain the information they are interested in from a large number of manufacturing cloud services has become an important research content for platform developers. The recommendation algorithm is the solution important approach to this problem.

However, less research has been conducted by researchers to integrate recommendation models with various advantages to enhance model recommendation, and the integration method of manually adjusting the weight hyperparameters occupied by each depth model is inflexible and time-consuming. In addition, the integrated recommendation models are larger and the model inference speed is slower. Therefore, this study designs a new activation function to better integrate the models, and uses distillation learning to compress the integrated models. For the convenience of subsequent representation, the integrated algorithm is called MultiModel and the distilled algorithm is abbreviated as KDMRA. the main contributions of MultiModel and KDMRA are as follows:

- Add a fully connected layer, integrate the advantages of the three cutting-edge deep learning models which are DeepFM, DIN, and MMDIN models, and use the gradient descent method to update the parameters of the fully connected layer.
- In order to make the ensemble model converge faster and better and simulate the scenario where each model votes, we initialize the parameters of the model and impose limits on the parameter variation range.
- In order to make the output domain of each sub-model output as [0, 1], the output value domain of the integrated model is also [0, 1], we designed a new activation function.
- Due to the large number of parameters of the integrated model, which requires large storage space and long training time, this paper performs knowledge distillation and compression on the integrated model.

This paper uses the famous open source MovieLens dataset to validate the model. Experiments show that the integrated model has better effect than other cutting-edge deep learning recommendation models, and after the model is compressed, the model storage space is saved and the model prediction effect is improved. In addition to movie recommendation, knowledge distillation recommendation method based on multi-model fusion can also be applied to recommendation in various fields such as text, speech and video.

The rest of the paper is organized as follows: Section 2 focuses on the research work related to the recommendation algorithm. Section 3 focuses on the relevant implementation details of the model. Section 4 focuses on the data set description, baseline algorithm, relevant metrics and experimental analysis of the experiment. Finally, this paper summarizes and outlooks the experimental research.

2. Related works

The related work section focuses on machine learning and deep learning related recommendation work.

2.1. Machine learning recommendation algorithms

In order to allow users to better obtain the information they are interested in, from machine learning to deep learning, experts and scholars have proposed various methods. Cui et al. [1] proposed a method of clustering using the K-means algorithm of cuckoo search, and then using collaborative filtering for recommendation. Since user preferences change over time, Hwangbo et al. [2] proposed a user preference decay function to simulate the changes in user interests, and improved the collaborative filtering algorithm. Given that most recommendation models only predict a single rating, Nassar et al. [3] proposed a deep multi-criteria collaborative filtering model, which achieved good recommendation results. Chonghuan [4] proposed a method of clustering users first, and then using matrix factorization techniques to recommend users. Unlike most user clustering methods that use user attributes for clustering, Chen [5] uses user historical decision-making features to cluster users, and then make recommendations. In order to improve the efficiency of code development, Nguyen et al. [6] built a

Persona personalized code recommendation model based on user coding preferences, experience, style, etc.

To solve the cold start problem, Nilesh et al. [7] proposed a content-based recommendation method to recommend recipes. Patra et al. [8] implemented a content-based recommendation method for recommending medical datasets. In order to solve the problem that only accurate keyword matching technology is used for recommendation in content-based recommendation, resulting in that the recommendation result field does not focus on the user's interest field, Liu et al. [9] proposed a method that uses both keywords and popularity. recommended method. Harshvardhan et al. [10] found that there is a correlation between user access content and time, so both ratings and access time are used as inputs, and a Boltzmann machine containing time information is used to build a recommendation system. Zihayat et al. [11] proposed a news recommendation system combining utility model and probabilistic topic model, which solved the cold start problem of items. Xiao et al. [12] combined association rule-based recommendation, content-based recommendation, and collaborative filtering-based recommendation to propose a hybrid personalized recommendation system [13]. Fernández-García et al. [14] then recommend a mix of music in multiple dimensions of music genre, theme, and voice tone.

The problem of recommendation performance and incremental model updates in big data scenarios is also a hot research topic nowadays. In order to improve the disease diagnosis and treatment effect of inexperienced doctors, Chen et al. [15] clustered disease symptoms and then made recommendations. At the same time, big data technologies such as spark were used to improve the efficiency of system recommendation. To solve the problem that traditional recommender system models cannot be incrementally updated with increasing data, Khalid et al. [16] proposed a novel online recommendation algorithm that votes on active learners and hyperspheres. In order to solve the performance problem of course recommendation in big data scenarios, Zhang et al. [17] proposed a distributed association rule mining algorithm, and used Hadoop for distributed storage and spark for distributed memory computing, which improved the recommendation effect and efficiency of the algorithm. However, because machine learning models often require a lot of effort to extract features, and are limited to a relatively simple model structure, it is difficult to better fit and express massive data.

2.2. Deep learning recommendation algorithms

The development of deep learning and reinforcement learning [18] has brought the research on recommendation algorithms into a new era. To better capture the higher-order feature interactions in the co-occurrence matrix, Lee et al. [19] proposed a deep learning recommendation model based on convolutional neural networks. In order to improve the effect of group recommendation, Huang et al. [20] designed a multi-attention neural network to capture the internal social characteristics of groups. Since user interests shift over time, Ahmadian et al. [21] proposed a temporal reliability measure to evaluate changes in user interests. Lin et al. [22] used L0 regularization to constrain the recommendation network parameters and proposed a new convex optimization method for recommendation networks. Zhang et al. [23] integrated user location features into the recommendation model, which improved the model recommendation effect. Huang et al. [24] proposed a recommendation model that simultaneously considers taxi GPS trajectories, passenger locations, road conditions, and revenue to implement a passenger recommendation method for taxi drivers. To better push carpooling requirements for recommendations, Su et al. [25] propose recommendation models that incorporate semantic trajectories. Li et al. [26] added the topological structure of users on social networks based on the use of rating data to implement personalized recommendations for users. Lee et al. [27] proposed a self-supervised representation learning method to encode items and companies and make recommendations by computing the similarity between them. Chiu et al. [28] proposed a preprocessing model based on unsupervised learning, combined with deep learning recommendation to provide users with personalized product recommendations. In order to utilize heterogeneous information, Shi et al. [29] designed a meta-path-based random walk algorithm to generate Embedding, which is further input into the model for training and prediction.

In order to alleviate the data sparsity and cold start problems in recommender systems, Ma et al. [30] constructed a knowledge graph and made recommendations through various information such as user-item, neighbor-neighbor, etc. Ye et al. [31] obtained low-dimensional representations of various entities by constructing a knowledge graph, and then input them into a neural decomposition machine for recommendation. Since traditional recurrent neural networks can only rely on linear transformations in each session to train recommendation models, Gwadabe et al. [32] propose a graph neural network-based recommender system that simultaneously uses non-sequential interactions and sequential The interactive information is used for model training, which improves the model recommendation effect. To solve the data sparsity and cold-start problems, Li et al. [33] investigated the construction and mining of higher-order semantic information in knowledge graphs and applied them to scholar recommendations. To solve the problem that GNNs are difficult to capture the implicit information in the interaction direction dimension, Chang et al. [34] propose a novel graph neural network for reciprocal recommendation. Since a similar group of users may also have some different interests, Ali et al. [35] used Pearson similarity to assign users to multiple interest groups, and combined group information and other information for recommendation, which improved the recommendation effect. Since customer reviews play an important role in recommendation decisions, Karthik et al. [36] proposed a fuzzy recommendation system based on sentiment analysis and customer interests. Moscato et al. [37] identify the user's emotion and recommend music to the user according to the user's emotion. Abbasi-Moud et al. [38] performed sentiment analysis and semantic clustering on user reviews, extracted user preferences, and used these user preferences to recommend users. To address the cross-domain problem in recommendation using sentiment analysis, Osman et al. [39] proposed a domain-adaptive recommendation algorithm using contextual information.

In order to protect user privacy, Qi et al. [40] designed an enhanced locality-sensitive hash recommendation algorithm. Gong et al. [41] also improved the local-sensitive hashing algorithm to protect user privacy from multiple dimensions. In order to protect data privacy, Chen et al. [42] proposed a new privacy-preserving method for recommender systems based on context awareness. Li et al. [43] used geographic information to effectively improve the system recommendation effect. To protect user privacy, Huo et al. [44] proposed a location-based privacy-preserving algorithm, which was integrated into the recommender system, and achieved a good balance between privacy and recommendation accuracy. In order to explain the recommendation results, Hsu et al. [45] proposed an explanatory recommendation system based on a knowledge graph to recommend funds.

3. KDMRA

The overall implementation process of KDMRA is shown in Fig 1. The algorithm mainly includes the teacher model part (the upper part of the figure) and the student model part (the lower part of the figure). The teacher model mainly integrates three mainstream recommendation algorithms, namely DIN (left), DeepFM (middle) and MMDIN (right). The student model is implemented using a simple shallow DIN model. Each module is described in detail below.

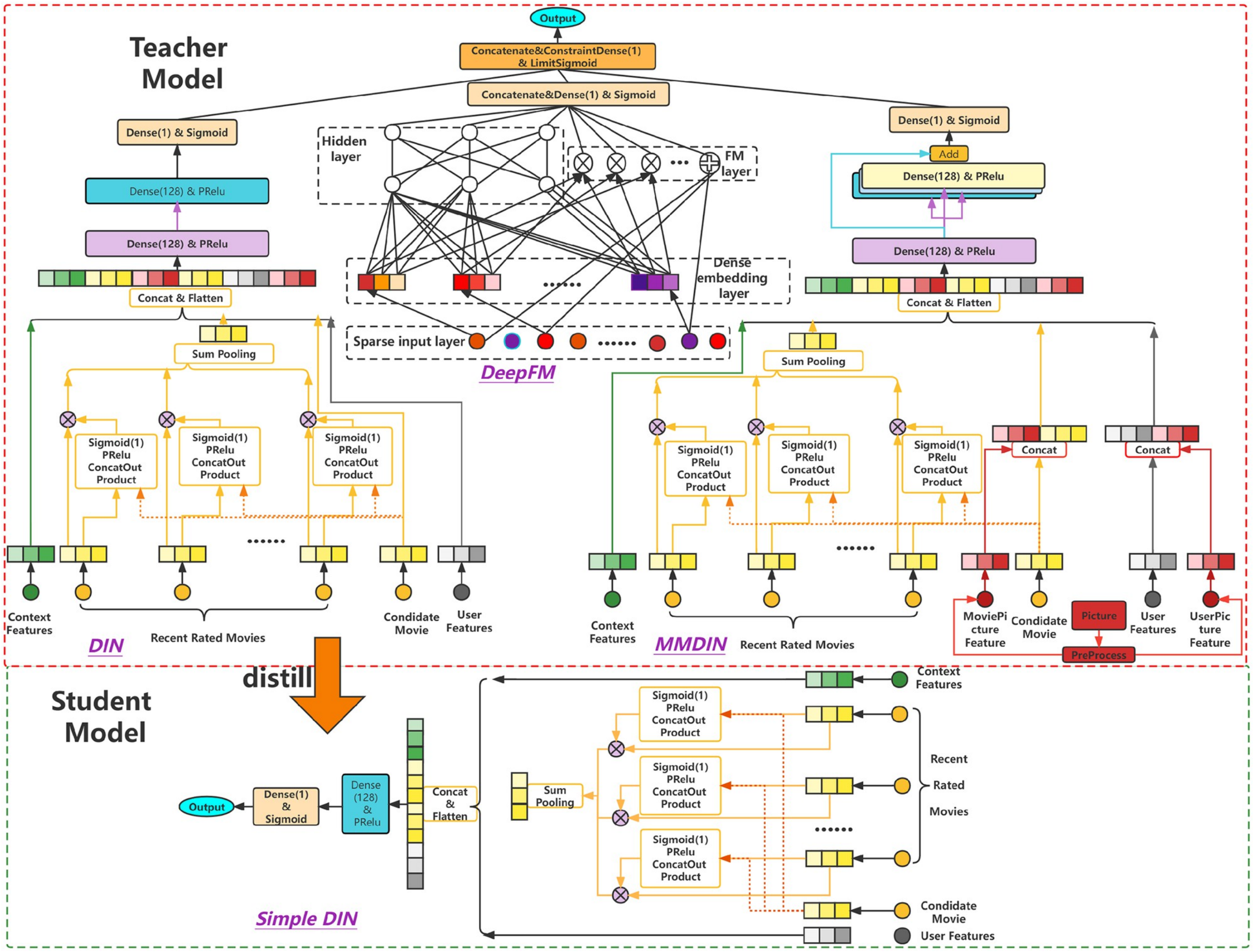

Fig 1. The structure of the proposed algorithm.

https://doi.org/10.1371/journal.pone.0275955.g001

In addition, the notations used in this paper and their meanings are shown in Table 1.

3.1. DIN module

The advantage of DIN is that it introduces an attention mechanism to avoid treating items that need to be rated equally. The main idea is to use the 5 movies recently rated by the user to do the outer product with the current candidate movies and weighted pooling to obtain the User's recent points of interest. As shown in the model in the upper left corner of Fig 1, the original input sparse features and non-numeric data are encoded to form dense feature vectors. Then, take the outer product of the current movie and the last 5 movies respectively (as shown in the yellow-edged parts in the upper left corner of Fig 1), and then connect them for PRelu and sigmoid activation to get the similarity between the current movie and the recently rated movie, and weight the similarity to each recently rated movie. above, do summation pool-ing. Finally, it is connected with user features and scene features, and is fitted and expressed through a multi-layer neural network (as shown in the three parts in the upper left corner of Fig 1) to obtain the final scoring result.

Table 1. Annotations.

Notations	Descriptions
V	Current parameter value
P	Previous parameter value
R	Rate of change
Vmin	Minimum value for parameter qualification
Vmax	Maximum value for parameter qualification
b	Initial Factor
k	Positive adjustment factor
Lsum	Total loss
L1	Loss between soft value and model predicted value
L2	Loss between true value and predicted value
α	Distillation factor
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
F1-Score	Summed average of precision and recall
TPR	True Positive Rate
FPR	False Positive Rate
G-Mean	Geometric mean of positive sample recall and negative sample recall
PR-AUC	The area under the PR curve
ROC-AUC	The area under the ROC curve

https://doi.org/10.1371/journal.pone.0275955.t001

3.2. DeepFM module

DeepFM improves the Wide&Deep model, changing the wide part to the FM part, so that the features can be combined and crossed, and then the item scores can be predicted. The DeepFM model structure is shown in the middle area of the upper part of Fig 1. Numerical features can be directly input to the next layer for operation, while non-numerical features are densified to obtain corresponding embedded representations. Some of the features are directly combined with feature crossover (as shown in the FM layer in Fig 1), and finally connected to the fully connected layer for activation and output; the other part is subjected to complex fitting and expression through a multi-layer neural network (as shown in the Hidden layer in Fig 1), and finally connected to the output layer. DeepFM not only ensures that the model has strong memory ability and generalization ability, but also enables the model to have the ability to predict the score by combining features.

3.3. MMDIN module

On the basis of DIN, the MMDIN module introduces the image features of items to predict user scores, and adds a multi-head mechanism to enable the model to extract features from different dimensions.

The MMDIN model is mainly divided into a multi-modal module, an attention module and a multi-layer neural network module. The multi-modal module (as shown in Fig 1 MMDIN module in red parts) is mainly responsible for extracting the color features of the image; the attention mechanism module (as shown in the yellow part of the MMDIN module in Fig 1) is responsible for extracting items that the user may be interested in from the items recently rated by the user. The multi-layer neural network module mainly adopts the Reset structure to alleviate the problem of gradient disappearance when the number of model layers becomes deeper, making it difficult for the scoring prediction to become worse as the model becomes deeper. At the same time, the multi-head mechanism (as shown by the trigeminal arrow in the MMDIN module in Fig 1) is adopted in the multi-layer neural network, so that the model can better extract features from the data from multiple dimensions.

3.4. Integrated learning module

The integrated learning module is mainly responsible for integrating the advantages of the DeepFM, DIN, and MMDIN models (as shown in the orange parts in the top half of Fig 1), so that the model has a better prediction effect, and also makes the algorithm more robust and stable. The ensemble learning module uses the fully connected layer to perform weighted voting on the prediction results of the three models to obtain the final prediction result. Among them, the weights are adaptively adjusted by the gradient descent method. In order to simulate the voting scenarios of each model, so that the model can converge better and faster, the parameters of the fully connected layer are specified and initialized, constraints are imposed, and the rate of change is set to specify the rate of parameter change. The parameter calculation method is as shown in Eq 1 shown. Among them, V represents the current parameter value, P represents the previous parameter value, R represents the rate of change, V_{min} represents the minimum value limited by the parameter, and V_{max} represents the maximum value limited by the parameter.

$$V = (1 - R)^{*}P + R^{*}\begin{cases} V_{\min} & P < V_{\min} \\ P & V_{\min} \leq P \leq V_{\max} \\ V_{\max} & P > V_{\max} \end{cases} \tag{1}$$

In addition, in order to make the output range of each sub-model be [0, 1], the output range of the integrated model is also [0, 1], that is, the final output score range, which is more in line with the actual change scenario, we study A new excitation function is designed, and the calculation method of the excitation function is shown in formula 2.

$$y = \frac{e^{kx}}{e^{kx} + b} \tag{2}$$

Among them, x is the input, y is the output, b is the initial factor, and k is the proportional adjustment coefficient. The closer the curve is to y = x, the better the activation effect of the activation function. Set the initial factor b = 100, adjust k, and get the activation effect of the activation function as shown in Fig 2. It can be intuitively obtained from Fig 2 that ask increases, the degree of the curve close to the straight line y = x becomes smaller and smaller. After reaching the minimum value, ask increases, the curve begins to move away from the straight line y = x. This paper takes b = 100, k = 9.24 for experiments.

3.5. Student module

The student module is mainly implemented with a simple shallow DIN structure. The advantage of knowledge distillation lies in the use of soft labels to guide the convergence of students' models, enabling students to make accurate predictions on samples they have not seen before. The loss function design of the distillation model is shown in Eq 3. Among them, L_{sum} represents the total loss, L_1 represents the loss between the soft value and the model predicted value, L_2 represents the loss between the true value and the predicted value, and α is the distillation

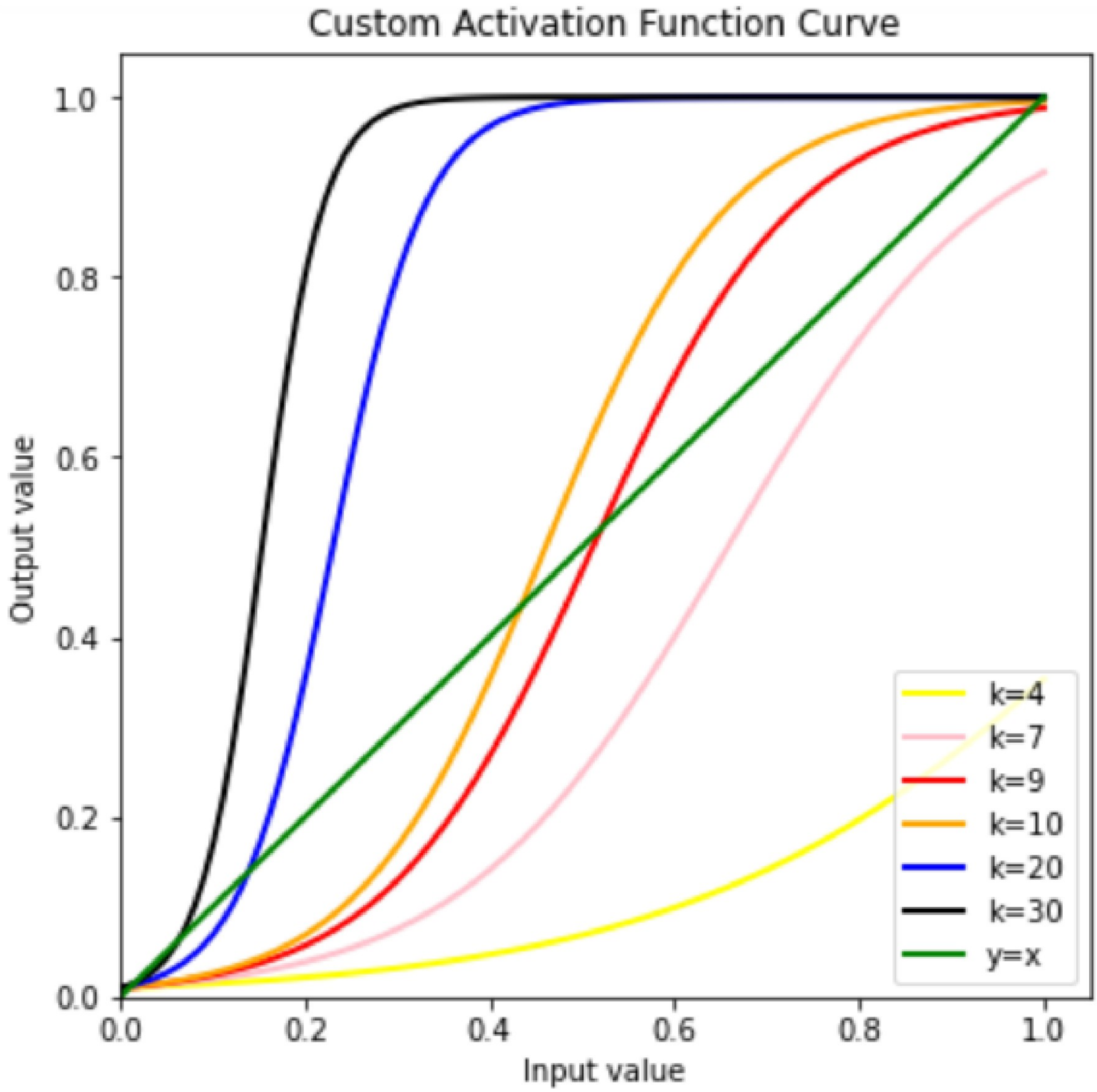

Fig 2. Custom activation function change curve with the k value changing.

https://doi.org/10.1371/journal.pone.0275955.g002

coefficient.

$$L_{\text{sum}} = L_1\alpha + L_2(1 - \alpha) \tag{3}$$

At this point, the KDMRA algorithm is introduced, and its main implementation process is sorted out as shown in Algorithm 1.

Algorithm 1: KDMRA

```
 Input: user information, item information, user item rating
 Output: Item score prediction
Initialize parameters
  1. User features, item features and user-item relationship features pre-processing.
  2. The processed features are input into DIN, DeepFM, and MMDIN models for prediction.
```

```
    2.1 The DIN model calculates the current candidate movie attention score from the user's recently viewed movies, weighted pooling to the current movie code, and input to the MLP layer along with user features, movie features and scene features for fitting and output.
    2.2 After thickening the ID-type features, the DeepFM model cross-combines the shallow features and connects them to the fully connected layer, while the other part performs complex crossover by MLP and then connects to the fully connected layer for output.
    2.3 MMDIN adds the use of picture-related features to DIN and uses multiple header Heads to extract the features, which are finally output to the MLP layer for fitting and output.
3   3. Integration: Integration of multiple models using custom activation functions that map both the input and output domains to the [0,1] range, simulating a voting scenario with dynamic weights to map the inputs to the outputs.
  4. Train and save models for evaluation.
4 Return
```

4. Experiments

4.1. Dataset description

This paper uses the open-source MovieLens dataset to verify the model effect. The distribution information of the dataset is shown in Table 2.

The MovieLens dataset mainly includes a movie dataset and a user rating dataset for movies. Among them, the user movie rating dataset mainly includes attributes such as user id, movie id, the user's rating of the movie, and the timestamp of the rating; the movie dataset mainly includes movie id, title, movie style, release year, and movie poster. As shown in Table 2, all the reviews of the first 1000 movies are selected, and after cleaning and preprocessing the movie dataset, the dataset has a total of 5,048,839 rating records, of which 20% are selected for test verification. Among them, in the training set and the test set, the proportion of user praise is 54.44% and 54.48% respectively, indicating that the distribution of the test set data and the training set data is basically the same.

4.2. Baseline algorithm

In the following experiments, we use five cutting-edge deep learning recommendation algorithms as benchmark algorithms for experimental comparison and analysis.

- NeuralCF: NeuralCF uses a multi-layer neural network to replace the simple interoperability layer in matrix decomposition, and at the same time solves the limitation that only rating information can be used in matrix decomposition so that user-side features and item-side features can be better crossed.
- Embedding MLP: It mainly includes a feature input layer, Embedding layer, Stacking layer, multi-layer residual network layer, and scoring output layer. The model mainly realizes the simultaneous input of numerical features and non-numerical features into the model for

Table 2. Dataset distribution.

	Training dataset	Testing dataset	Positive rate on training dataset	Positive rate on testing dataset	Total Num
MovieLens	4,039,104	1,009,235	54.44%	54.48%	5,048,339

https://doi.org/10.1371/journal.pone.0275955.t002 training, and uses residual connections to alleviate the problem of network gradient disappearance.

- DeepFM: On the basis of the WideDeep model, the FM layer is added, so that the model has a better recommendation effect in scenarios where the combined features have a great influence on the prediction results.
- DIN: The e-commerce advertising recommendation model released by Ali, which introduces an attention mechanism, takes the outer product of the user's recently rated products and candidate products to obtain the attention score, and then weights it to the recently rated products and performs summation and pooling. It is finally input into the MLP model for scoring prediction.
- MMDIN: On the basis of DIN, multi-head mechanism and multi-modal method are added. The multi-head mechanism enables the model to extract features from data from multiple dimensions. The multi-modal method mainly mines image features to train the model, which enriches the feature library that the model can use.

4.3. Experiment setup and evaluation index

This experiment runs in Ubuntu 18.04 and Cuda 11.1 environment, which is configured with 8-core CPU, 16G memory, GeForce RTX 2060 super graphics card, and 8G memory size. Developed with Python 3.8.1, Tensorflow 2.5 and Spark 2.4.3.

The experimental evaluation mainly uses comprehensive indicators such as PR-AUC, ROC-AUC, and F1-Score, as well as single indicators such as Accuracy, Recall Rate, and Precision. The formulas for precision and recall are shown in Eqs (4) and (5).

$$\mathrm{Pr}ecision = \frac{TP}{TP + FP} \tag{4}$$

$$\mathrm{Re}call = \frac{TP}{TP + FN} \tag{5}$$

The F1-Score is the summed average of the precision rate and recall rate. F1-Score is the summed average of precision and recall, which is a comprehensive evaluation index of precision and recall, and its calculation formula is shown in Eq (6).

$$F1 - Score = \frac{2 * \mathrm{Pr}ecision * \mathrm{Re}call}{\mathrm{Pr}ecison + \mathrm{Re}call} \tag{6}$$

G-Mean is also a comprehensive evaluation index of the recommendation model, which has a better evaluation effect on the recommendation scenario in the case of data imbalance, and its calculation formula is shown in Eq (7).

$$G - Mean = \sqrt{\frac{TP}{TP + FN} + \frac{TN}{TN + FP}} \tag{7}$$

The true positive rate (TPR), false positive rate (FPR), and equilibrium accuracy were calculated as shown in Eqs (8), (9) and (10), respectively.

$$TPR = \frac{TP}{TP + FN} \tag{8}$$

$$FPR = \frac{FP}{FP + TN} \tag{9}$$

$$\text{Balanced Accuracy} = \frac{TPR + TNR}{2} \tag{10}$$

The PR curve is a combined dynamic assessment of precision and recall, while the ROC curve is a combined dynamic assessment of true-positive and false-positive rates. Two metrics, PR-AUC and ROC-AUC, indicate the magnitude of the area enclosed by the PR and ROC curves and the coordinate axis, respectively. In this experiment, the Adam optimizer is used to optimize each model.

4.4. Evaluation of experimental results

4.4.1. Comparison of key evaluation indicators for recommendation algorithms. In order to evaluate the pros and cons of the model, this study uses ROC-AUC, PR-AUC, FScore and other comprehensive indicators to comprehensively evaluate the model, and uses single evaluation indicators such as Precision, Recall, Accuracy, and Loss to evaluate the model unilaterally. Among them, the industry-recognized and most used indicator is the ROC-AUC indicator. This experiment is carried out on the MovieLens, and the evaluation results are shown in Table 3 below.

It can be seen from Table 3 that in the experimental evaluation results of MovieLens, the multimodel integration algorithm MultiModel has the highest value in the three comprehensive evaluation indicators ROC-AUC, PR-AUC and F1-Score, and the single evaluation indicators Precision, Recall, Accuracy, Balanced Accuracy, Loss, KDMRA, MMDIN and DIN models have their own strengths. As a compressed version of the multi-model ensemble model MultiModel, KDMRA also has good results. In addition, in the evaluation of 1,009,235 test data, the MultiModel model took 238s, while KDMRA used only 42s, significantly reducing the inference time required by the multi-model fusion algorithm. It can be seen that on the MovieLens dataset, the comprehensive performance ranking of each model is: MultiModel > MMDIN > KDMRA > DIN > DeepFM > Embedding MLP > NeuralCF. Among them, the confusion matrix of KDMRA is shown in Fig 3. As can be seen from Fig 3, the KDMRA accuracy is basically consistent with the accuracy in Table 3.

4.4.2. Precision and recall rate curve of each model. In order to analyze the recommendation effect of each model more intuitively, in this experiment, after shuffling the MovieLens

Table 3. Comparison of key evaluation indicators for recommendation algorithms.

Methods	ROC-AUC	PR-AUC	F1-Score	Precision	Recall	Accuracy	Balanced Accuracy	Loss	Evaluation Time(s)
NeuralCF	0.7319	0.7537	0.7358	0.6738	0.7701	0.6717	0.6612	0.6030	41
EmbeddingMLP	0.7610	0.7786	0.7483	0.7060	0.7576	0.6961	0.6906	0.5792	**40**
DeepFM	0.7799	0.7947	0.7614	0.7134	0.7872	0.7118	0.7041	0.5598	41
DIN	0.7897	0.8028	0.7671	0.7064	**0.8263**	0.7183	0.7083	0.5582	42
MMDIN	0.8010	0.8156	0.7724	0.7334	0.7877	**0.7284**	0.7222	**0.5370**	46
Multi-Model	**0.8018**	**0.8171**	**0.7736**	0.7549	0.7354	0.7258	0.7226	0.6250	238
KDMRA	0.8007	0.8149	**0.7736**	**0.7577**	0.7262	0.7243	**0.7231**	0.5690	42

https://doi.org/10.1371/journal.pone.0275955.t003

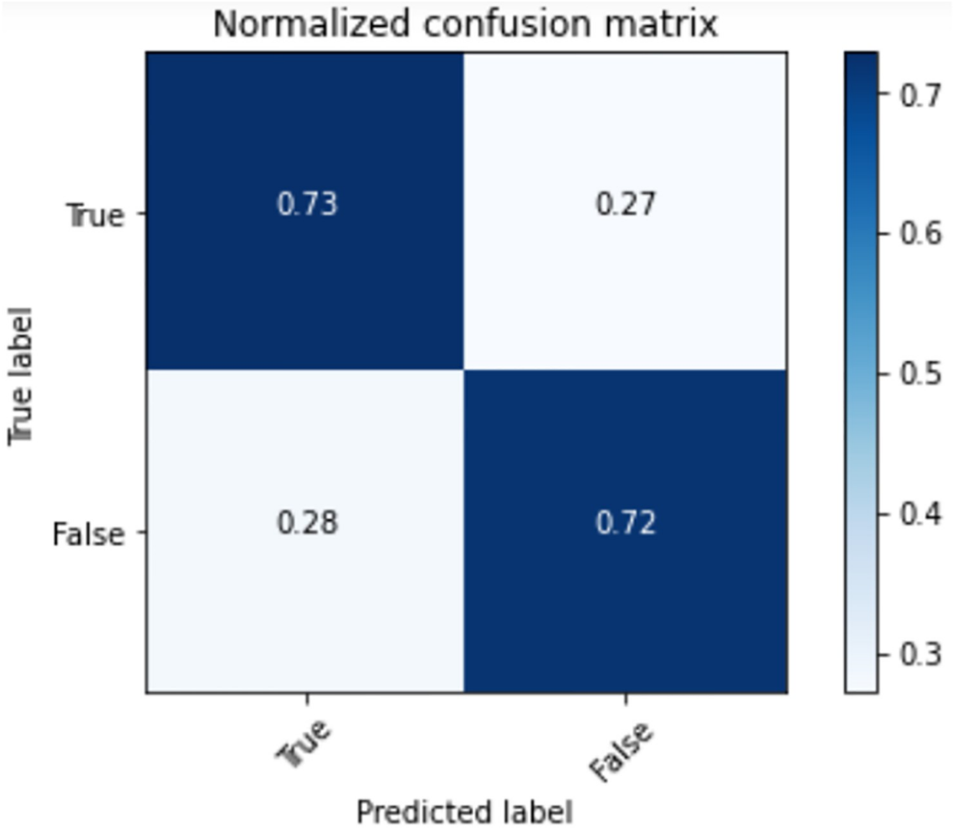

Fig 3. Confusion matrix of KDMRA.

https://doi.org/10.1371/journal.pone.0275955.g003

datasets, 20,000 pieces of data were randomly selected for evaluation and PR curve was drawn, as shown in Fig 4.

In the PR curve, the larger the area under the curve, the better the model recommendation effect. It can be seen from Fig 4 that among the 20,000 pieces of rating data extracted this time, the favorable rate is 0.5442, which is equal to the favorable rate on the entire dataset (as shown in Table 2). The uppermost curve in Fig 4 is the yellow curve, which represents the curve where the MultiModel is located. This is followed by the red and yellow curves, representing the curves where MMDIN and KDMRA are located, respectively. The overall ranking of model recommendation effects is: MultiModel > MMDIN > KDMRA > DIN > DeepFM > Em bedding MLP > NeuralCF. Among them, when Threshold is 0.2342, the KDMRA model achieves the best precision rate and recall rate, which are 0.6890 and 0.8819, respectively, and the corresponding PR-AUC value is 0.8124 (the formula for calculating each indicator is shown in subsection 4.3).

It can be seen from Fig 4 that the MultiModel algorithm has a better recommendation effect than other models on the MovieLens dataset. The model compressed by the knowledge distillation method also has a good recommendation effect.

4.4.3. Receiver operating curve of each model. The ROC curve is a recognized indicator in the recommendation system that can comprehensively reflect the comprehensive recommendation effect of the recommendation model. Like the PR curve, the larger the area under the ROC curve, the better the model recommendation effect. The recommendation effects of each model on the MovieLens datasets are shown in Fig 5.

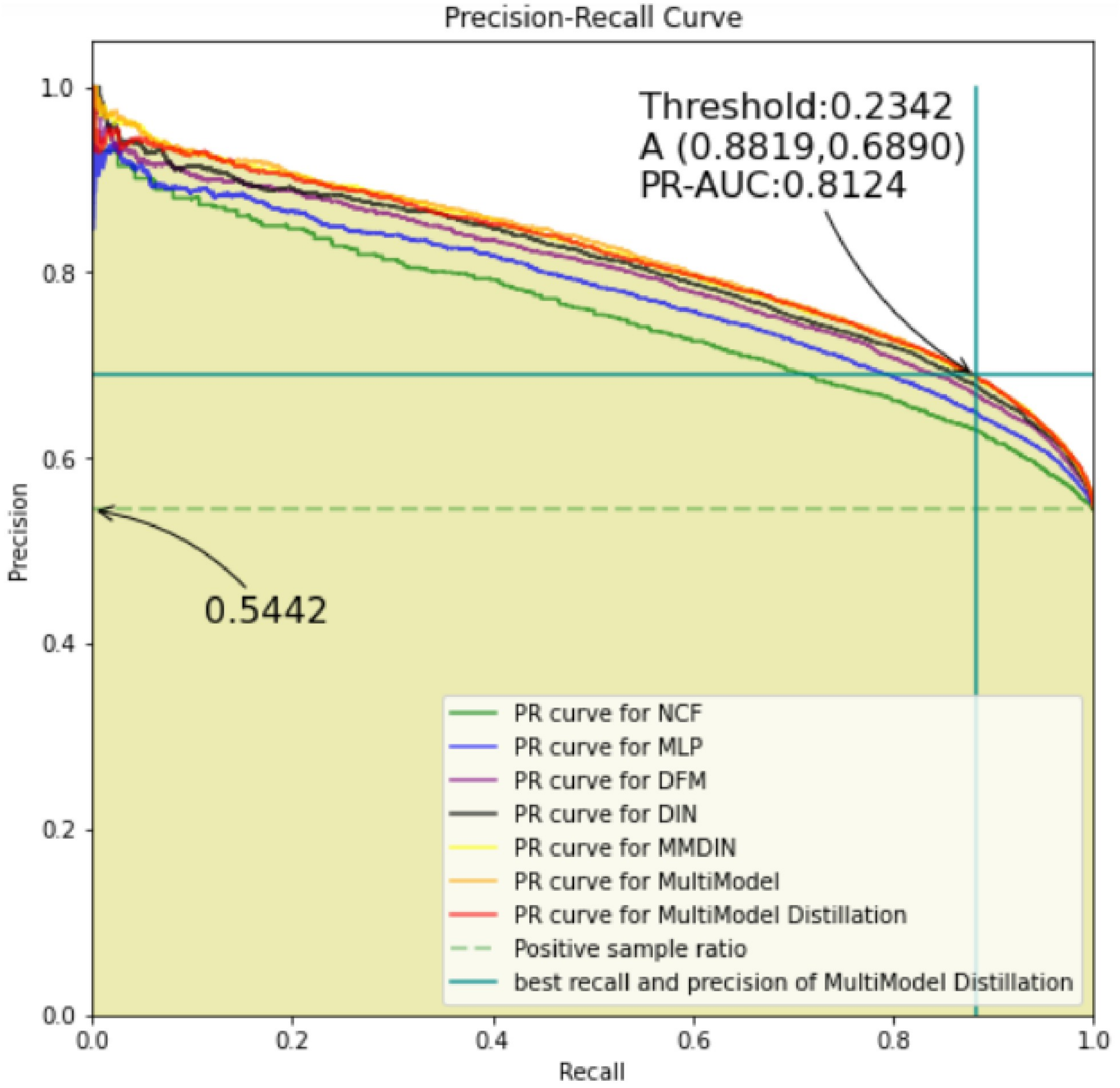

Fig 4. The precision and recall rate curve of each model on the MovieLens dataset.

https://doi.org/10.1371/journal.pone.0275955.g004

In Fig 5, the three curves of yellow, red and orange basically overlap, representing the curves of MultiModel, MMDIN and KDMRA respectively, and the yellow curve is slightly higher than the red curve, followed by the orange curve. The following are the curves of black, purple, blue, and green, respectively ghostwriting the curves of DIN, DeepFM, Embedding MLP, and NeuralCF, that is, the recommended effect size order is: MultiModel > MMDIN > KDMRA > DIN > DeepFM > Embedding MLP > NeuralCF. When Threshold is 0.1658, the true positive rate and false positive rate of KDMRA model are 0.8646 and 0.3270, respectively, the ROC-AUC value at this time is 0.8362, and the recall rate is 0.8646 (the formula for calculating each indicator is shown in subsection 4.3).

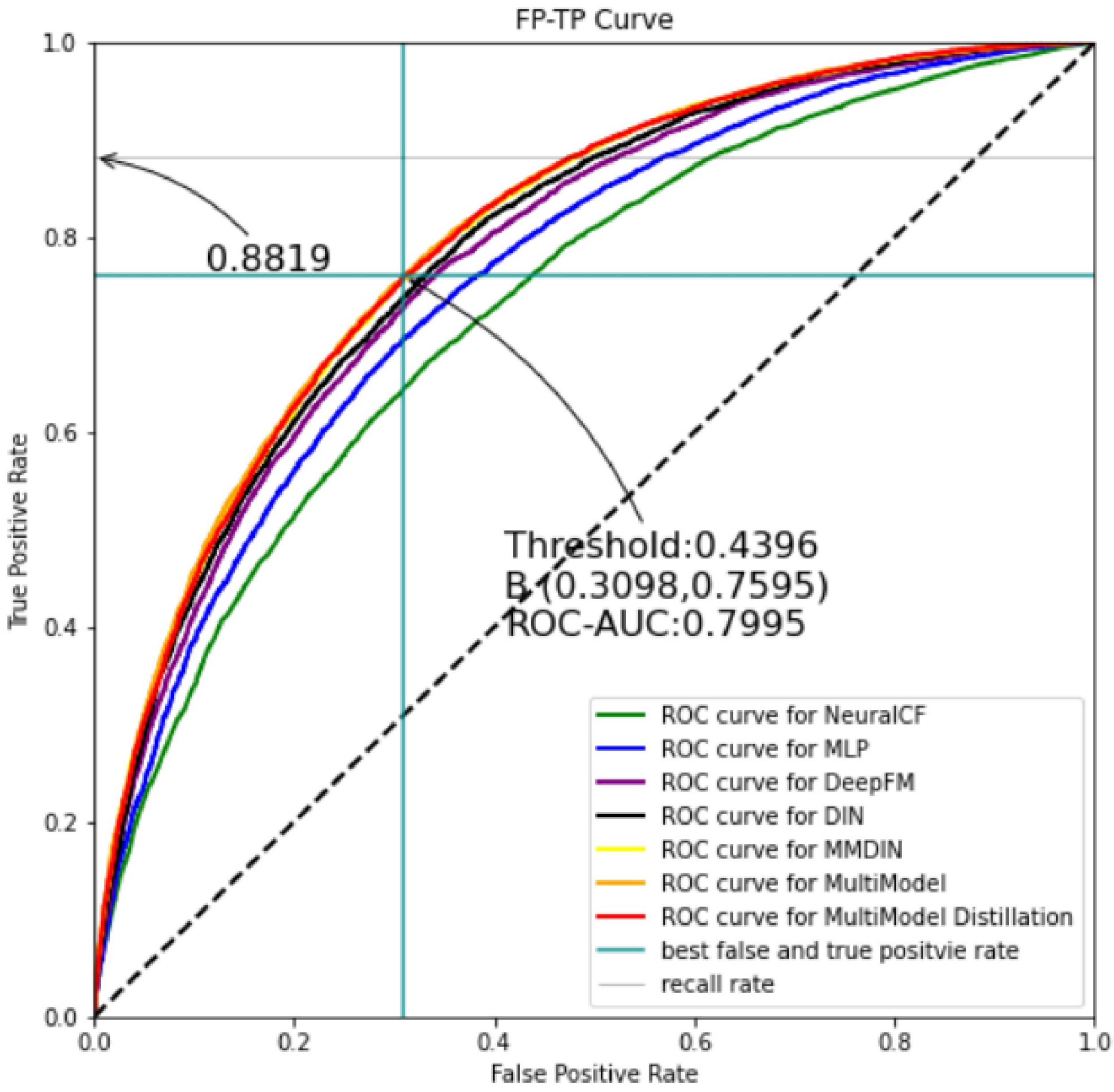

Fig 5. Receiver operating curves of each model on the MovieLens dataset.

https://doi.org/10.1371/journal.pone.0275955.g005

It can be seen from Fig 5 that the MultiModel model has a better comprehensive recommendation effect than other models, and the KDMRA model after knowledge distillation also has a better effect.

4.4.4. Model intuitive prediction effect analysis. In order to show the model prediction effect more intuitively, in this experiment, after shuffling the MovieLens datasets, 20,000 pieces of data are randomly selected for scoring prediction, and then the scatter plot is drawn as shown in Fig 6. Among them, the ordinate represents the score prediction result, and the abscissa represents any floating point number between 0.0 and 1.0.

It can be seen from Fig 6 that the green points represent the points that the user actually rated as positive, and these points are mainly distributed in the upper part of the image, that is,

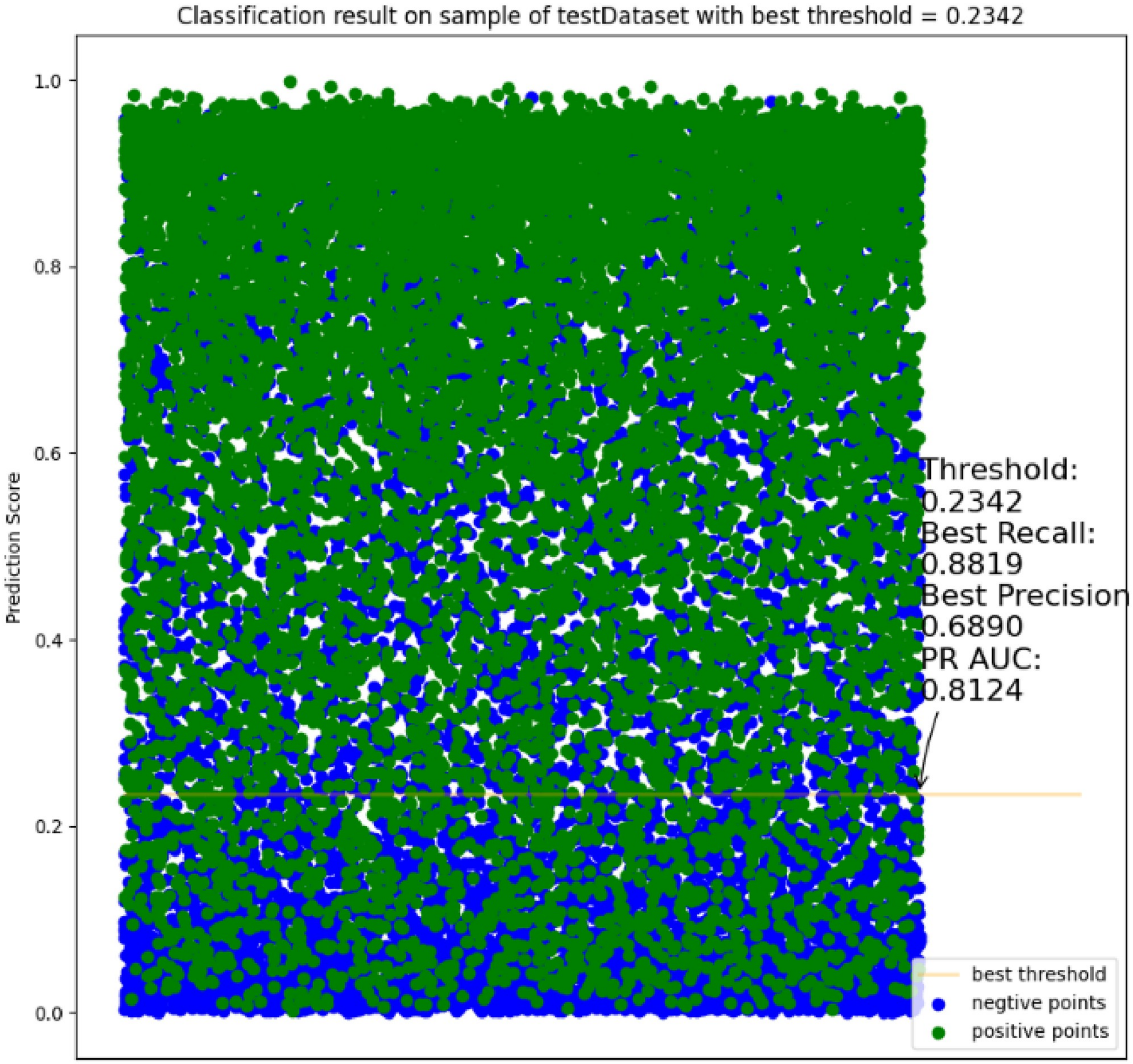

Fig 6. Model intuitive classification effect on the MoiveLens dataset.

https://doi.org/10.1371/journal.pone.0275955.g006

the KDMRA model predicts a relatively high score. The blue dots represent the points where the user's actual rating is negative. These points are mainly distributed below the image, that is, the rating predicted by the model is also relatively low, which indicates that the rating prediction result of the model is reliable. Among them, when the threshold is 0.2342, the model achieves the best recall rate and precision rate, which are 0.8819 and 0.6890, respectively, and the PR-AUC value at this time is 0.8124 (the formula for calculating each indicator is shown in subsection 4.3).

4.4.5. Effectiveness of MultiModel and its distillation model. In order to further conduct an intuitive and comprehensive evaluation of MultiModel and its distillation model, this experiment draws a columnar comparison chart of the comprehensive evaluation indicators of each model for comparison, as shown in Fig 7.

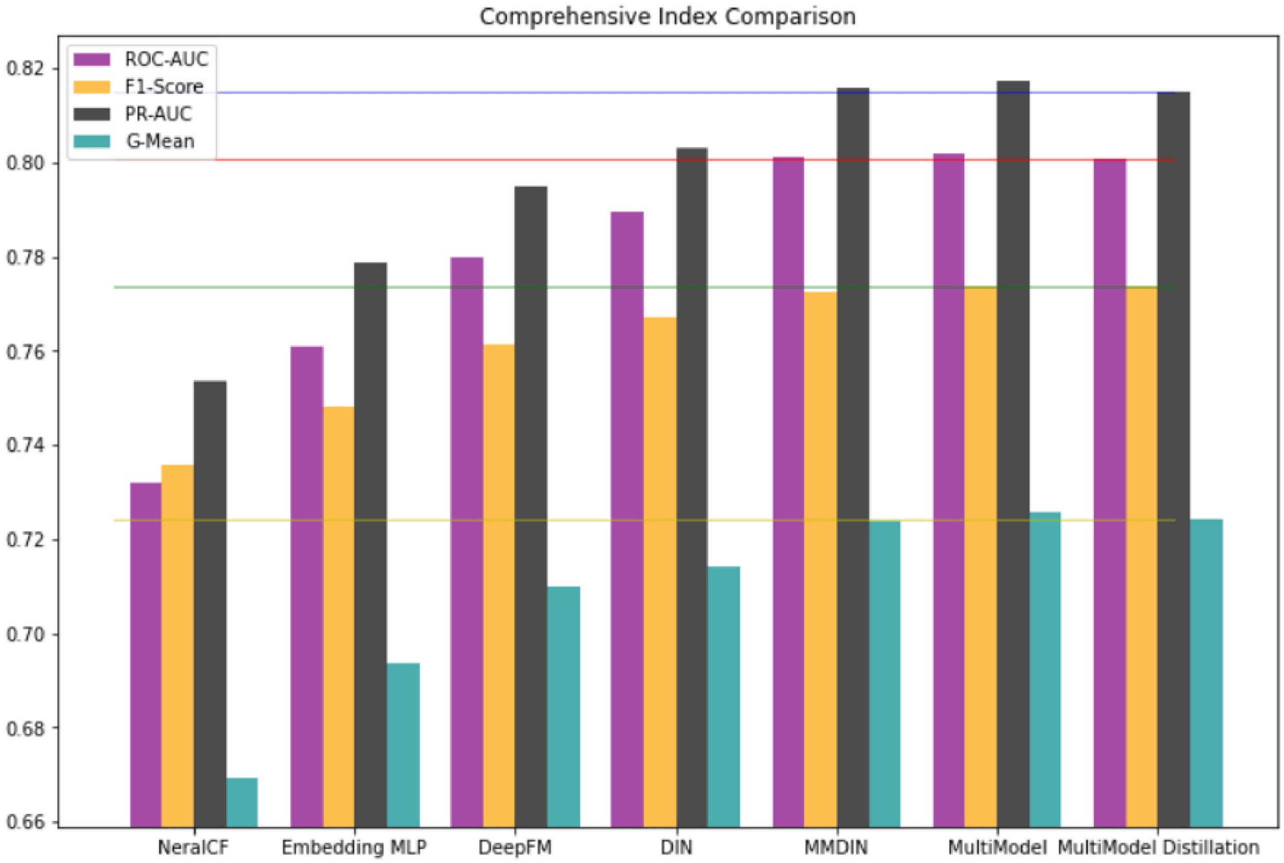

Fig 7. Comparison of comprehensive indicators of each model on the MovieLens dataset.

https://doi.org/10.1371/journal.pone.0275955.g007

It can be seen intuitively from Fig 7 that on the whole, the MultiModel model has high values in various comprehensive indicators such as ROC-AUC, F1-Score, PR-AUC, and G-Mean (the formula for calculating each indicator is shown in subsection 4.3). The compressed KDMRA model also has higher values.

In summary, through the PR curve, receiver operating curve and comprehensive evaluation index of the model, it can be concluded that MultiModel has a better recommendation effect than other models, and its distilled KDMRA model also has a better recommendation effect.

5. Conclusions and future work

In this study, we integrate the advantages of several models to obtain a model with better recommendations. In order to make the model converge faster and better, we initialize and impose constraints on the model parameters. In addition, we design a new activation function to better simulate the sub-model voting scenario. Finally, to reduce the number of model parameters and improve the model inference speed, we distill the integrated large model with the knowledge to improve the recommendation efficiency. In conclusion, the multi-model fusion knowledge distillation recommendation algorithm effectively improves the system recommendation effect and recommendation system, and the method can be applied in various fields such as e-commerce recommendations, advertising recommendations, and long and short video recommendations.

In the future, we plan to continue in-depth research on recommender systems in the following aspects. First, study the recommendation of lifelong learning to improve the recommendation effect of the model after updating the model parameters with incremental data. Then, we study the effect of the variable-length self-attention mechanism on the improvement of recommendation effect. Finally, we study multi-modal reinforcement learning recommendations by simulating human learning by interacting with the environment.

Supporting information

S1 Text. Supporting data for all the figures and tables in this text.
(DOCX)

Author Contributions

Conceptualization: Shaobo Li, Peng Zhou, JianJun Hu.

Data curation: Mingbao Yang.

Funding acquisition: Shaobo Li.

Methodology: Mingbao Yang, Peng Zhou.

Supervision: Shaobo Li, Peng Zhou.

Validation: Mingbao Yang.

Visualization: Mingbao Yang.

Writing – original draft: Mingbao Yang.

Writing – review & editing: Mingbao Yang, Peng Zhou.

References

1. Cui Z., Xu X., Xue F., Cai X., Cao Y., Zhang W., et al., Personalized Recommendation System Based on Collaborative Filtering for IoT Scenarios, IEEE Trans. Serv. Comput. 13 (2020) 685–695. https://doi.org/10.1109/TSC.2020.2964552
2. Hwangbo H., Kim Y.S., Cha K.J., Recommendation system development for fashion retail e-commerce, Electronic Commerce Research and Applications. 28 (2018) 94–101. https://doi.org/10.1016/j.elerap.2018.01.012
3. Nassar N., Jafar A., Rahhal Y., A novel deep multi-criteria collaborative filtering model for recommendation system, Knowledge-Based Systems. 187 (2020) 104811. https://doi.org/10.1016/j.knosys.2019.06.019
4. Chonghuan X., A novel recommendation method based on social network using matrix factorization technique, Inf. Process. Manage. 54 (2018) 463–474. https://doi.org/10.1016/j.ipm.2018.02.005
5. Chen T., A fuzzy ubiquitous traveler clustering and hotel recommendation system by differentiating travelers' decision-making behaviors, Applied Soft Computing. 96 (2020) 106585. https://doi.org/10.1016/j.asoc.2020.106585
6. Nguyen T.T., Nguyen T.T., PERSONA: A personalized model for code recommendation, PLOS ONE. 16 (2021) e0259834. https://doi.org/10.1371/journal.pone.0259834
7. N. Nilesh, M. Kumari, P. Hazarika, V. Raman, Recommendation of Indian Cuisine Recipes Based on Ingredients, in: 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW), 2019: pp. 96–99.
8. Patra B.G., Maroufy V., Soltanalizadeh B., Deng N., Zheng W.J., Roberts K., et al., A content-based literature recommendation system for datasets to improve data reusability–A case study on Gene Expression Omnibus (GEO) datasets, Journal of Biomedical Informatics. 104 (2020) 103399. https://doi.org/10.1016/j.jbi.2020.103399
9. Liu H., Kou H., Yan C., Qi L., Keywords-Driven and Popularity-Aware Paper Recommendation Based on Undirected Paper Citation Graph, Complexity. 2020 (2020) 2085638. https://doi.org/10.1155/2020/2085638

10. Harshvardhan G., Gourisaria M.K., Rautaray S.S., Pandey M., UBMTR: Unsupervised Boltzmann machine-based time-aware recommendation system, Journal of King Saud University—Computer and Information Sciences. (2021). https://doi.org/10.1016/j.jksuci.2021.01.017

11. Zihayat M., Ayanso A., Zhao X., Davoudi H., An A., A utility-based news recommendation system, Decision Support Systems. 117 (2019) 14–27. https://doi.org/10.1016/j.dss.2018.12.001

12. Xiao J., Wang M., Jiang B., Li J., A personalized recommendation system with combinational algorithm for online learning, J Ambient Intell Human Comput. 9 (2018) 667–677. https://doi.org/10.1007/s12652-017-0466-8

13. Walek B., Fajmon P., A hybrid recommender system for an online store using a fuzzy expert system, Expert Systems with Applications. (2022) 118565. https://doi.org/10.1016/j.eswa.2022.118565

14. Fernández-García A.J., Rodriguez-Echeverria R., Preciado J.C., Perianez J., Gutiérrez J.D., A hybrid multidimensional Recommender System for radio programs, Expert Systems with Applications. 198 (2022) 116706. https://doi.org/10.1016/j.eswa.2022.116706

15. Chen J., Li K., Rong H., Bilal K., Yang N., Li K., A disease diagnosis and treatment recommendation system based on big data mining and cloud computing, Information Sciences. 435 (2018) 124–149. https://doi.org/10.1016/j.ins.2018.01.001

16. Khalid A., Lundqvist K., Yates A., Ghzanfar M.A., Novel online Recommendation algorithm for Massive Open Online Courses (NoR-MOOCs), PLOS ONE. 16 (2021) e0245485. https://doi.org/10.1371/journal.pone.0245485 PMID: 33481886

17. Zhang H., Huang T., Lv Z., Liu S., Zhou Z., MCRS: A course recommendation system for MOOCs, Multimed Tools Appl. 77 (2018) 7051–7069. https://doi.org/10.1007/s11042-017-4620-2

18. Vijayakumar P., Jegatha Deborah L., Rajkumar S. C., Deep Reinforcement Learning-Based Pedestrian and Independent Vehicle Safety Fortification Using Intelligent Perception, Int. J. Softw. Sci. Comput. Intell. 14 (2022) 1–33. https://doi.org/10.4018/IJSSCI.291712

19. Lee S., Kim D., Deep learning based recommender system using cross convolutional filters, Information Sciences. 592 (2022) 112–122. https://doi.org/10.1016/j.ins.2022.01.033

20. Huang Z., Xu X., Zhu H., Zhou M., An Efficient Group Recommendation Model With Multiattention-Based Neural Networks, IEEE Trans. Neural Netw. Learn. Syst. 31 (2020) 4461–4474. https://doi.org/10.1109/TNNLS.2019.2955567 PMID: 31944999

21. Ahmadian S., Joorabloo N., Jalili M., Ahmadian M., Alleviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach, Expert Systems with Applications. 187 (2022) 115849. https://doi.org/10.1016/j.eswa.2021.115849

22. Lin J., Li Y., Lian J., A novel recommendation system via L0-regularized convex optimization, Neural Comput & Applic. 32 (2020) 1649–1663. https://doi.org/10.1007/s00521-019-04213-w

23. Zhang Y., Yin C., Wu Q., He Q., Zhu H., Location-Aware Deep Collaborative Filtering for Service Recommendation, IEEE Trans. Syst. Man Cybern. -Syst. 51 (2021) 3796–3807. https://doi.org/10.1109/TSMC.2019.2931723

24. Huang Z., Shan G., Cheng J., Sun J., TRec: an efficient recommendation system for hunting passengers with deep neural networks, Neural Comput & Applic. 31 (2019) 209–222. https://doi.org/10.1007/s00521-018-3728-2

25. Su J., Li J., Semantic Trajectory Frequent Pattern Mining Model: The Definitions and Theorems, Int. J. Semant. Web Inf. Syst. 18 (2022) 1–20. https://doi.org/10.4018/IJSWIS.297031

26. Li Z., Xiong F., Wang X., Chen H., Xiong X., Topological Influence-Aware Recommendation on Social Networks, Complexity. (2019) 6325654. https://doi.org/10.1155/2019/6325654

27. Lee J., Sohn S.Y., Recommendation system for technology convergence opportunities based on self-supervised representation learning, Scientometrics. 126 (2021) 1–25. https://doi.org/10.1007/s11192-020-03731-y

28. Chiu M.-C., Huang J.-H., Gupta S., Akman G., Developing a personalized recommendation system in a smart product service system based on unsupervised learning model, Computers in Industry. 128 (2021) 103421. https://doi.org/10.1016/j.compind.2021.103421

29. Shi C., Hu B., Zhao W.X., Yu P.S., Heterogeneous Information Network Embedding for Recommendation, IEEE Trans. Knowl. Data Eng. 31 (2019) 357–370. https://doi.org/10.1109/TKDE.2018.2833443

30. Ma X., Dong L., Wang Y., Li Y., Zhang H., MNI: An enhanced multi-task neighborhood interaction model for recommendation on knowledge graph, PLOS ONE. 16 (2021) e0258410. https://doi.org/10.1371/journal.pone.0258410 PMID: 34710122

31. Ye Q., Hsieh C.-Y., Yang Z., Kang Y., Chen J., Cao D., et al., A unified drug–target interaction prediction framework based on knowledge graph and recommendation system, Nat Commun. 12 (2021) 6775. https://doi.org/10.1038/s41467-021-27137-3 PMID: 34811351

32. Gwadabe T.R., Liu Y., Improving graph neural network for session-based recommendation system via non-sequential interactions, Neurocomputing. 468 (2022) 111–122. https://doi.org/10.1016/j.neucom.2021.10.034

33. Li P., Li T., Wang X., Zhang S., Jiang Y., Tang Y., Scholar Recommendation Based on High-Order Propagation of Knowledge Graphs, Int. J. Semant. Web Inf. Syst. 18 (2022) 1–19. https://doi.org/10.4018/IJSWIS.297146

34. Chang Y., Shu L., Du E., Chen C., Zhang Z., Zheng Z., et al., GraphRR: A multiplex Graph based Reciprocal friend Recommender system with applications on online gaming service, Knowledge-Based Systems. 251 (2022) 109187. https://doi.org/10.1016/j.knosys.2022.109187

35. Ali Y., Khalid O., Khan I.A., Hussain S.S., Rehman F., Siraj S., et al., A hybrid group-based movie recommendation framework with overlapping memberships, PLOS ONE. 17 (2022) e0266103. https://doi.org/10.1371/journal.pone.0266103 PMID: 35358269

36. Karthik R.V., Ganapathy S., A fuzzy recommendation system for predicting the customers interests using sentiment analysis and ontology in e-commerce, Applied Soft Computing. 108 (2021) 107396. https://doi.org/10.1016/j.asoc.2021.107396

37. Moscato V., Picariello A., Sperli G., An Emotional Recommender System for Music, IEEE Intell. Syst. 36 (2021) 57–68. https://doi.org/10.1109/MIS.2020.3026000

38. Abbasi-Moud Z., Vahdat-Nejad H., Sadri J., Tourism recommendation system based on semantic clustering and sentiment analysis, Expert Systems with Applications. 167 (2021) 114324. https://doi.org/10.1016/j.eswa.2020.114324

39. Osman N.A., Noah S.A.M., Darwich M., Mohd M., Integrating contextual sentiment analysis in collaborative recommender systems, PLOS ONE. 16 (2021) e0248695. https://doi.org/10.1371/journal.pone.0248695 PMID: 33750957

40. Qi L., Wang X., Xu X., Dou W., Li S., Privacy-Aware Cross-Platform Service Recommendation Based on Enhanced Locality-Sensitive Hashing, IEEE Trans. Netw. Sci. Eng. 8 (2021) 1145–1153. https://doi.org/10.1109/TNSE.2020.2969489

41. Gong W., Qi L., Xu Y., Privacy-Aware Multidimensional Mobile Service Quality Prediction and Recommendation in Distributed Fog Environment, Wirel. Commun. Mob. Comput. (2018) 3075849. https://doi.org/10.1155/2018/3075849

42. Chen J., Liu L., Chen R., Peng W., Huang X., SecRec: A Privacy-Preserving Method for the Context-Aware Recommendation System, IEEE Transactions on Dependable and Secure Computing. (2021) 1–1. https://doi.org/10.1109/TDSC.2021.3085562

43. Li K., Wei H., He X., Tian Z., Relational POI recommendation model combined with geographic information, PLOS ONE. 17 (2022) e0266340. https://doi.org/10.1371/journal.pone.0266340 PMID: 35427385

44. Huo Y., Chen B., Tang J., Zeng Y., Privacy-preserving point-of-interest recommendation based on geographical and social influence, Information Sciences. 543 (2021) 202–218. https://doi.org/10.1016/j.ins.2020.07.046

45. Hsu P.-Y., Chen C.-T., Chou C., Huang S.-H., Explainable mutual fund recommendation system developed based on knowledge graph embeddings, Appl Intell. (2022). https://doi.org/10.1007/s10489-021-03136-1