Current Science
CS 5 (5), 4887-4902 (2025)
Received 20 July 2025 | **Revised 25 August** 2025 | **Accepted 07 September** 2025 | **Online Available 18 October** 2025
https://doi.org0/10.52845/CS/2025-5-5-42

Current Science

Current Science

OPEN ACCESS

ISSN(O)2795-8639

Origional Article

Research on Optimization of Recommendation Model Based on Value Evaluation: A Case Study of Amazon Dataset

Zhexuan Zhao

National Key Laboratory of Information Systems Engineering, National University of Defense Technology, Hunan, China

***Corresponding Author: Zhexuan Zhao**

Abstract:
The core objective of a recommendation system is to precisely match items that users might be interested in. Among them, how to provide personalized recommendations for different user groups has become a key research issue. Currently, mainstream recommendations adopt a single-model approach, and the recommendation results are usually only applicable to a single type of user group. This paper aims to combine the evaluation and reasoning capabilities of large models, adopt an innovative recommendation process, and use different recommendation algorithms for different user groups to obtain a series of recommendation results. By leveraging the comprehensive evaluation capabilities of the Deepseek-V3.1 large model, the optimal recommendation small models for different user groups are provided. Finally, the reasoning capabilities of the large model are utilized to give the parameter optimization scheme for the model, achieving the optimization of the model effect. This paper conducted experimental verification using the Amazon dataset, and used models such as the heuristic recall algorithm and the dual-tower model algorithm for model recommendations. Through the above process, model screening and optimization were carried out, and the optimized results were obtained. After comparison, it was found that the model's accuracy rate increased by 34.88% and the recall rate increased by 40.74%, verifying the hypothesis of the experiment.

Index Terms: Personalized recommendation, Model evaluation, Parameter tuning, Model fusion

I. Introduction

With the rapid development of Internet information technology and e-commerce platforms, the total amount of information has seen explosive growth, and information overload has become a core obstacle hindering users from efficiently obtaining the resources they need. Faced with a vast array of products, videos, news, and services, users often find themselves in a dilemma of choice, struggling to quickly identify content that truly aligns with their interests and needs. As a key tool to address this issue, the core task of recommendation systems is to accurately identify and efficiently present content that users might be interested in from billions of pieces of information, thereby significantly enhancing user experience, strengthening users' dependence on and loyalty to the platform, and bringing higher click-through rates, conversion rates, and commercial value to the platform.

Traditional recommendation methods typically adopt a "one-size-fits-all" strategy, relying on a single model (such as collaborative filtering, content-based recommendation, or deep learning models) to provide recommendation services to all users. These methods assume homogeneity among user groups and overlook the significant differences in user preferences and behavior patterns that actually exist. For instance, new users and old users have vast differences in behavioral sparsity and stability, and high-activity users and low-activity users have different acceptance levels for push frequency and

diversity, not to mention the divergent decision-making paths brought about by different consumption intentions (such as price comparison, exploration, and immediate purchase). Therefore, a single model is difficult to fully capture such a complex user demand structure and has obvious limitations in generalization ability and personalization, especially in handling niche users and long-tail items.

In recent years, although various neural network-based recommendation models (such as neural collaborative filtering, graph neural networks, etc.) have made significant progress in traditional evaluation metrics such as accuracy and recall, their basic architectures still handle extremely diverse user groups in a uniform manner and have not fundamentally responded to the challenges brought by group differences. These models often improve overall metrics while masking their recommendation failures in certain subgroups. Therefore, scientifically and effectively clustering users and tailoring the most suitable recommendation algorithms for different groups have become a key research direction for advancing recommendation systems to a higher stage.

Meanwhile, large language models (LLMs) have demonstrated unprecedented capabilities in natural language understanding, logical reasoning, and generation tasks, providing a new technical path for the development of recommendation systems. Large models can not only conduct deep semantic modeling of user behavior sequences to identify their intrinsic preferences and intentions but also possess powerful evaluation and reasoning capabilities, which can be used for the selection, integration, and hyperparameter optimization of recommendation sub-models. Based on this, this study proposes an innovative recommendation system architecture: the system first uses the intelligent analysis capabilities of large language models to conduct fine-grained user clustering; then, multiple lightweight recommendation sub-models (such as model A specifically for cold start, deep sequence model B suitable for high-activity users, and model C for a specific regional population, etc.) are constructed and trained for different groups to form a candidate model set; subsequently, with the comprehensive evaluation and decision-making capabilities of the large model, the optimal recommendation model is selected and assigned to each group; finally, the large model's reasoning and optimization capabilities are utilized to customize and adjust the model hyperparameters for different groups, achieving more refined performance improvements.

The core advantage of this method lies in its hierarchical and adaptive mechanism. Through the scheduling of large language models, the system no longer statically relies on a single model but flexibly invokes the most suitable recommendation strategy based on the dynamic changes and real-time feedback of user groups. This architecture not only significantly enhances the coverage and personalization levels for heterogeneous user groups but also provides a feasible path for the future adaptive evolution and continuous learning of recommendation systems. The experiments were based on multiple real public datasets (such as Amazon Product Data), covering different fields and user scale scenarios, ensuring the reliability and generalization of the conclusions. In addition to the conventional accuracy and recall rates, this study further evaluated the improvement effects of the group recommendation strategy on user satisfaction, long-tail item mining, and cold start problems, thereby more comprehensively verifying the comprehensive advantages and practical value of the proposed method.

The main contributions of this paper are as follows:

(1) For the Amazon dataset, a user clustering strategy based on regional features is proposed to facilitate subsequent experiments on each classified group.

(2) A model selection scheme based on LLM-driven model choice was proposed for each classified group.

(3) For the optimal recommendation model, a parameter tuning scheme based on the fusion of the Deepseek-V3.1 large model and the small model of the recommendation system is proposed.

II The Proposed Method

In this section, we present our problem and elaborate on our model, namely the recommendation algorithm selection and tuning method for model evaluation, as shown in Figure

1. **A. Introduction**

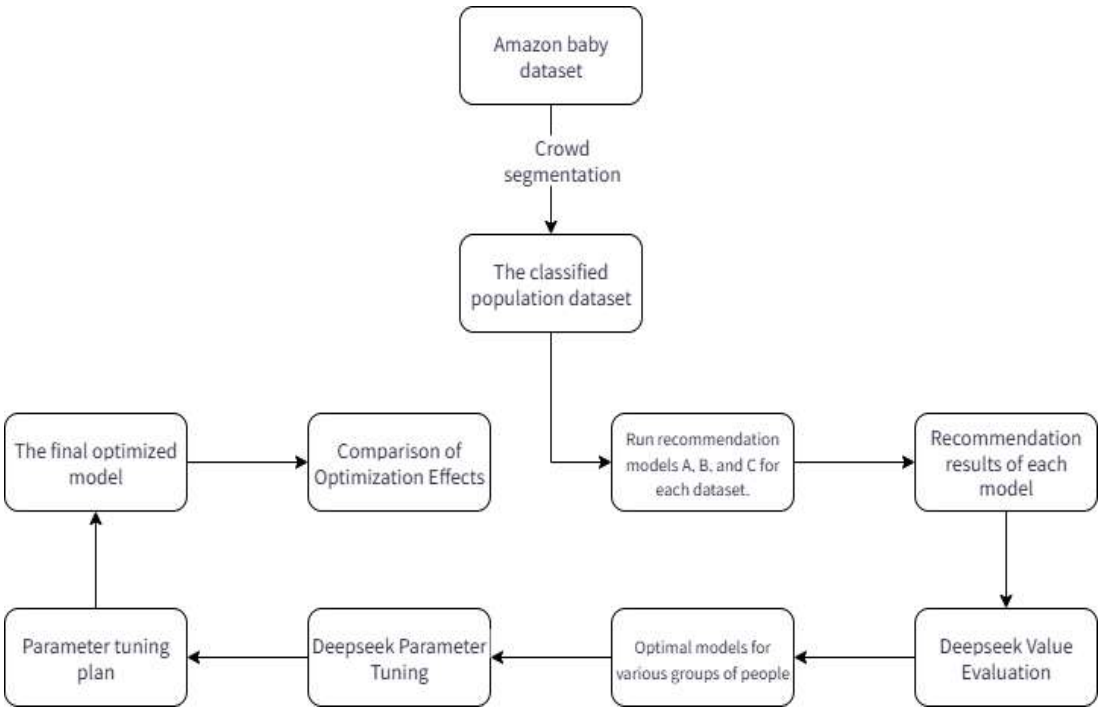

Figure 1 Model flowchart

Firstly, this paper uses the Amazon baby dataset for experiments. The project dataset is divided into target groups such as children, young people, middle-aged people, and the elderly, thereby obtaining different divided datasets. Secondly, different recommendation models, such as A, B, and C models, are trained using each group's dataset as the training set, and the recommendation effects of each model in each group are obtained, with recall rate and accuracy as the indicators. Then, the recommendation results of each model in each group are input into the large model Deepseek, and the evaluation ability of Deepseek is used to evaluate the recommendation effects of each model in each group, obtaining the optimal recommendation model for each group.

Then, parameter tuning is conducted for the optimal recommendation model in each group. Different parameters are input into the model to obtain different recommendation results. These parameters and recommendation results are input into Deepseek, and the reasoning ability of the Deepseek large model is used to summarize the parameter tuning schemes for each model. Finally, these parameter tuning schemes are implemented to obtain the optimal parameter tuning results for each optimal model in each group. The recommendation results before and after parameter tuning are compared to observe whether there is an improvement.

B. Introduction to Heuristic Recommendation Algorithm Model

Firstly, we introduce the first recommendation model algorithm we have prepared: the heuristic recommendation algorithm. The specific algorithm process is shown in Figure 2.

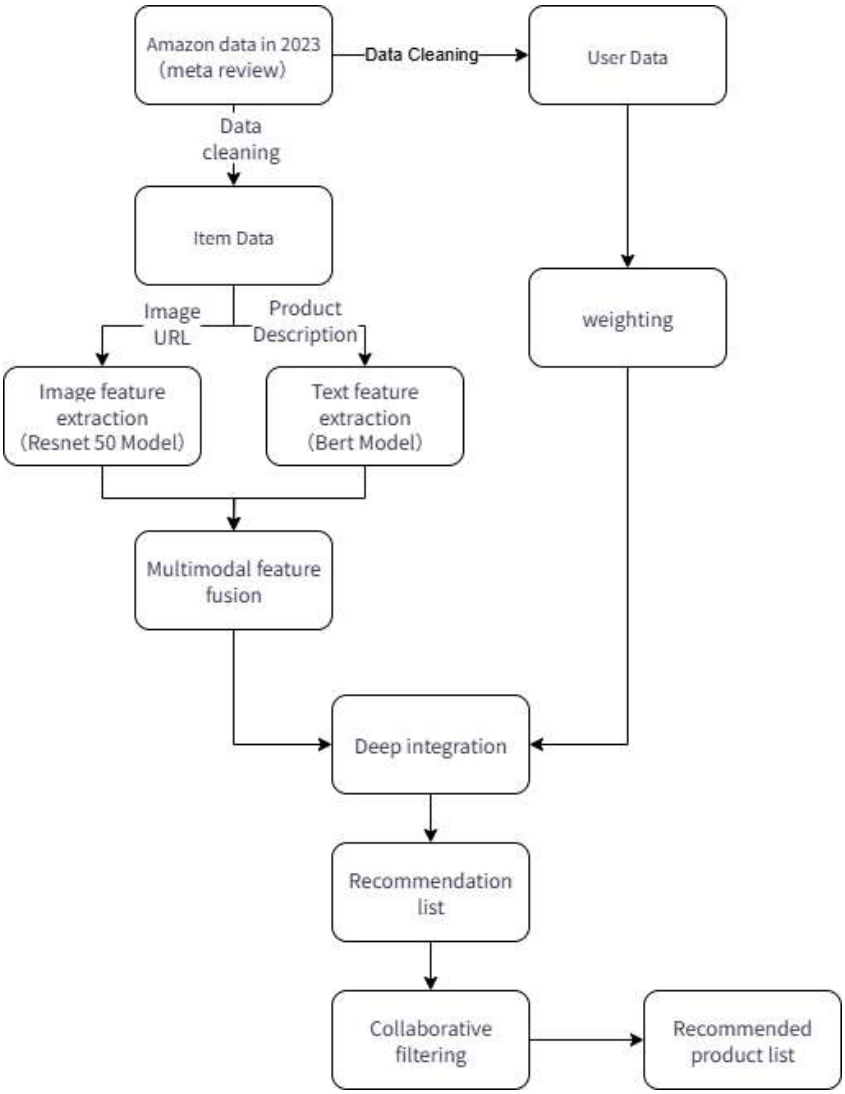

Figure 2 Flowchart of Heuristic Algorithm 1

Specifically, we first perform data cleaning on the 2023 Amazon dataset to obtain user review data and product data. For the product data, we obtain the corresponding product images or texts based on the product image addresses and product descriptions, and then use the Resnet50 and Bert models to extract image and text features respectively, obtaining the image vectors and text vectors of the products.

Next, we fuse the two different multimodal vectors and reduce the dimensions to obtain the product similarity matrix. On the other hand, we obtain the rating matrix of each user's rating for the products based on the ratings in the user review data. Finally, we multiply the two matrices to obtain the user's product recommendation list, which is the deep integration of the model. The specific process is shown in Figure 3. Finally, we use the collaborative filtering method to further filter the recommendation results to obtain the final recommended product list.

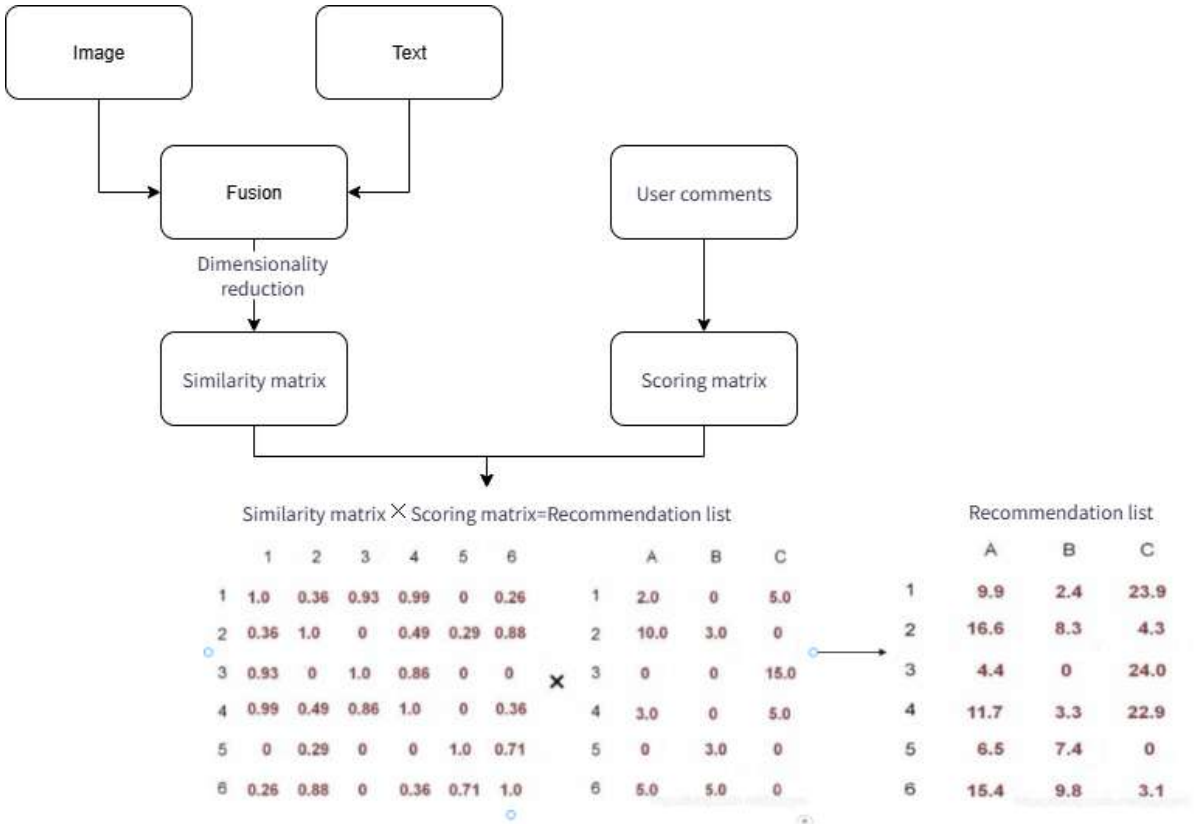

Figure 3 Flowchart of Heuristic Algorithm 2

C. Introduction to the Dual-Tower Recommendation Model

In the recall process of the recommendation system, another model worth trying is the dual-tower model. The architecture of the dual-tower model is shown in Figure 4:

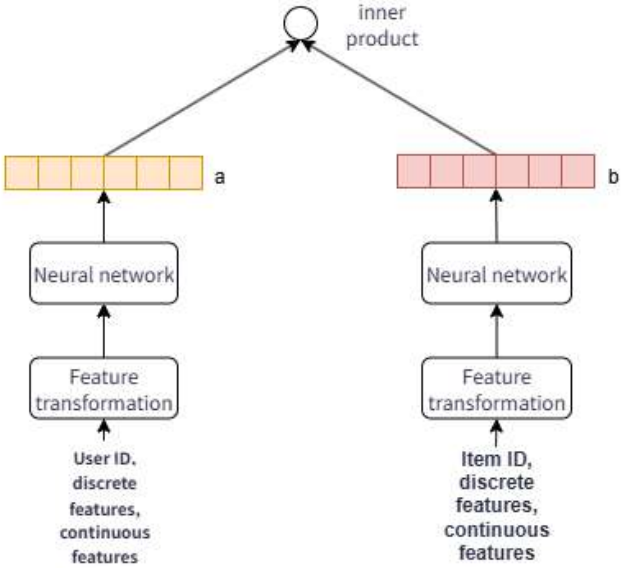

Figure 4 Architecture diagram of the dual-tower model

Firstly, based on the user features in the dataset, including discrete and continuous features, feature transformation is carried out and input into the neural network to obtain the user feature vectors. Secondly, based on the item features in the dataset, including discrete and continuous features, feature transformation is conducted and input into the neural network to obtain the item feature vectors.

Then, using the user-item interaction records, that

is, the user's purchase records as a new dataset, the corresponding relationship between each user and item is established and input into the neural network. After training, a function that can map user feature vectors to item feature vectors is obtained.

Finally, all user vectors are mapped to the corresponding item vectors in the item vector space through this mapping function. Based on the similarity of the item vectors in the item space, K most similar items are given as the recommendation list.

D. Introduction to Sorting Recommendation Model

After the recall process in the recommendation system, the recommended results are generally sorted. In this paper, the DeepFM ranking model is adopted for sorting.

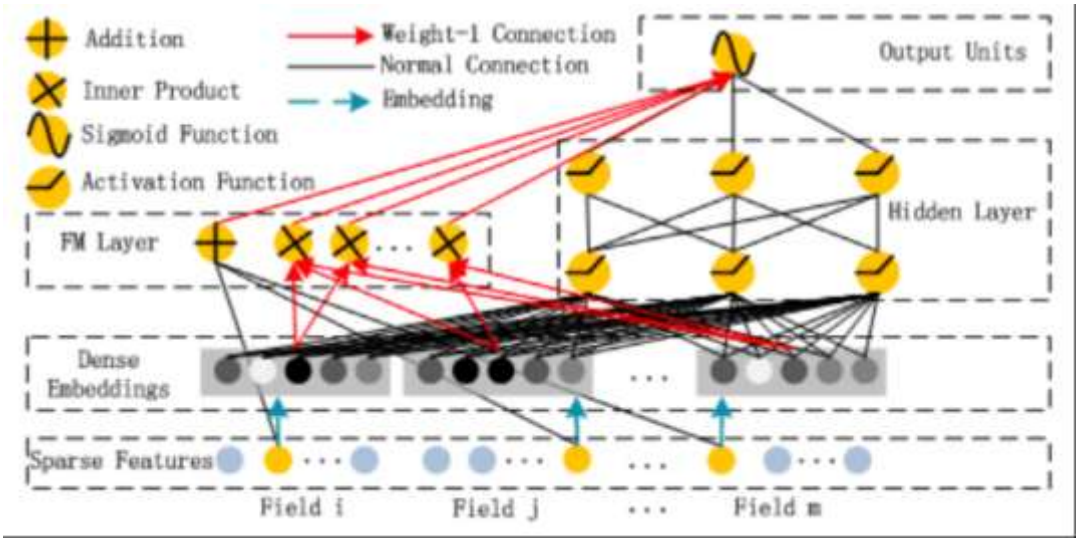

Figure 5 DeepFM Architecture Diagram

The DeepFM model structure mainly consists of two parts: the FM Component and the Deep Component.

First, there is the shared input layer and embedding layer: all input features (whether sparse categorical features or dense numerical features) are preprocessed first. Each feature field (Field) is transformed into a low-dimensional dense vector (Embedding Vector).

For sparse features (such as User ID): the corresponding embedding vector is looked up through the embedding layer (Embedding Layer).For dense features (such as user age): they can be directly connected, or discretized and then embedded, or directly input as a scalar to subsequent layers.

The FM component and the Deep component share the same set of embedding vectors. This not only reduces a large number of model parameters but also enables the two components to jointly learn better feature representations during training.

Second, the FM component: the FM part is an improved factorization machine that automatically learns all second-order feature combinations without manual intervention. Its function is to handle the "memory" part of the model, efficiently learning the second-order interactions between features. The output of the FM part is a scalar, consisting of three parts: global bias (w0), the first-order linear part, and the second-order feature interaction part (the sum of the inner products of all non-zero feature corresponding embedding vectors). The second-order part can be mathematically simplified to reduce the computational complexity from $O(kn^2)$ to $O(kn)$, where k is the dimension of the embedding vector and n is the number of features. This makes the FM part very efficient.

Finally, the Deep component: the Deep part is a standard feedforward neural network (FNN). Its function is to handle the "generalization" part of the model, learning high-order feature combinations and complex patterns through multiple nonlinear hidden layers. It concatenates the embedding vectors from all feature domains

into a long dense vector.

This concatenated vector is then input into several fully connected layers (Hidden Layers), where it undergoes nonlinear transformations through activation functions (such as ReLU). The output of the last hidden layer is then projected into a scalar through another fully connected layer.

III Experiments

In this section, we conduct experiments on the Amazon baby dataset to answer the following research questions:

(1) Which recommendation model is the most suitable for each segmented group?

(2) According to the parameter tuning scheme provided by the large model, can this scheme improve the model's recommendation accuracy and recall rate?

A. Experimental setup

1) Dataset

This experiment uses the Amazon baby dataset, generally referring to the Amazon Baby subset, which is part of the Amazon multimodal recommendation public dataset. This dataset contains user interactions with baby products (such as ratings, clicks, and purchases), and provides image features and text description features for each item. The main file contents include: interaction data or similar files: each line is a triplet in the format of userid, itemid, rating, recording the interaction between users and items. There are also split interaction data such as train.txt and test.txt. Image features are the image features of each item, usually vectors extracted by CNN models such as ResNet and VGG. Text features are the text features of each item, usually vectors extracted by models such as BERT and Word2Vec. Item/user KNN graphs include item-item and user-user K-nearest neighbor graph structures, used for graph neural network propagation. Auxiliary files include ID mapping tables for items and users, and item KNN graphs for different K values.

2) Baseline Module

To select the best recommendation model suitable for all groups of people, we designed four different recommendation models by using different recall algorithms and whether to sort the recall results. The specific models are as follows:

Heuristic algorithm + no sorting (Model 1): The recommendation recall of this model uses the heuristic algorithm, and the sorting of the recommendation list is done without sorting.

Heuristic algorithm + DeepFM sorting (Model 2): The recommendation recall of this model uses the heuristic algorithm, and the sorting of the recommendation list is done using the DeepFM sorting method.

Dual-tower model algorithm + no sorting (Model 3): The recommendation recall of this model uses the dual-tower model algorithm, and the sorting of the recommendation list is done without sorting.

Dual-tower model algorithm + DeepFM sorting (Model 4): The recommendation recall of this model uses the dual-tower model algorithm, and the sorting of the recommendation list is done using the DeepFM sorting method.

3) Evaluation Protocols

We use two widely adopted evaluation metrics to assess the performance of recommendation systems: precision and recall.

Precision: Precision, also known as the positive predictive value, represents the proportion of samples predicted as positive that are actually positive.

Recall: Recall, also referred to as the true positive rate, indicates the proportion of actual positive samples among those predicted as positive in the entire sample set.

B. Performance Comparison(RQ1)

First, we divided the target group into different populations in four regions: AE, AF, AG, and AH. Then, we compared the accuracy and recall rates of each model for each classified population. The results are shown in Tables 1 and 2.

Table I Accuracy rates of each model for different categories

Accuracy	Model 1	Model 2	Model 3	Model 4
AE	0.18%	2.4444%	0.037%	2.1333%
AF	0.1905%	2.0%	0.1651%	2.4889%

AG	0.1918%	2.5161%	0.1569%	2.1677%
AH	0.1905%	2.1379%	0.2312%	2.3172%

Table II Recall rates of each model for different categories

Recall	Model 1	Model 2	Model 3	Model 4
AE	16.11%	33.3333%	2.6667%	25.3333%
AF	15.1852%	25.0%	13.3333%	28.8889%
AG	16.7742%	31.2903%	12.7742%	25.1613%
AH	15.1852%	24.3103%	16.9655%	28.1379%

Based on the recommendation results of each model for different classified groups, the data was input into Deepseek for value assessment, and the most suitable recommendation model for each classified group was obtained. The dialogues with Deepseek are shown in Figures 6,7 and 8 as follows:

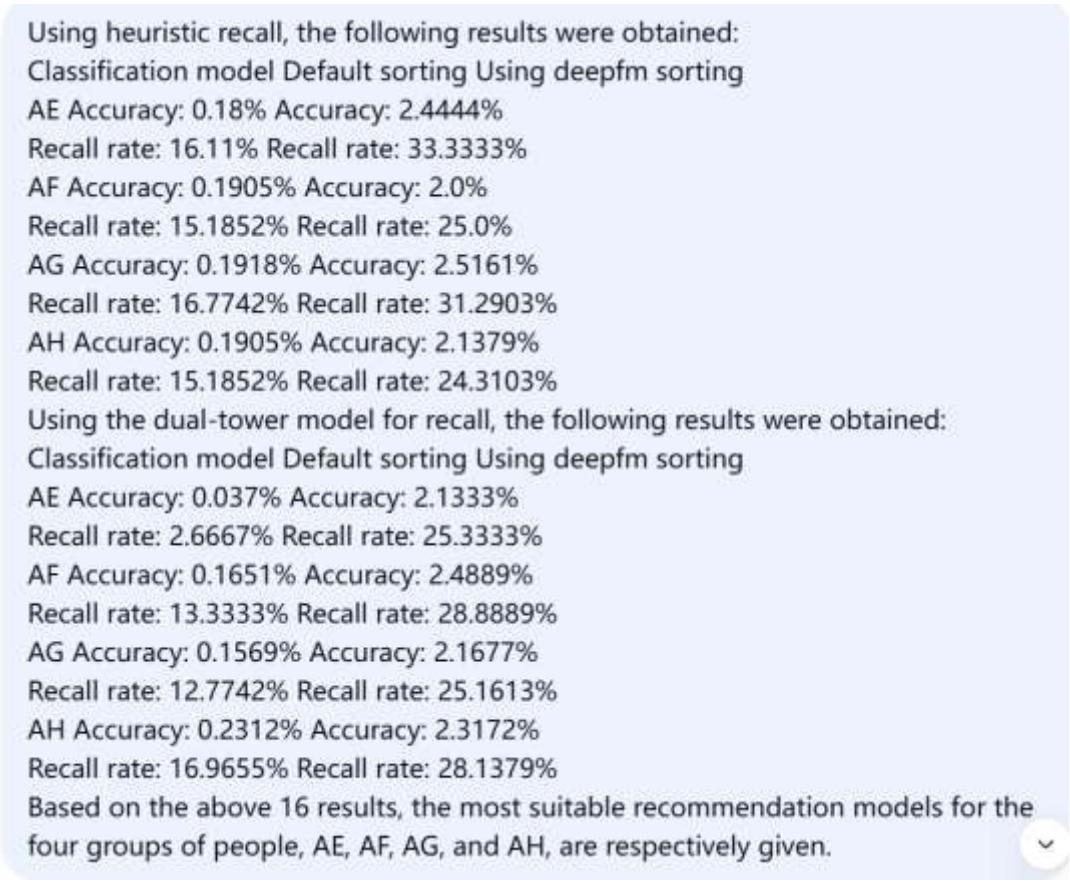

Figure 6 Value Evaluation Dialogue Diagram 1

Based on the 16 results provided for the four user groups (AE, AF, AG, AH) under two recall strategies (heuristic recall and dual-tower recall) and two sorting models (default sorting and DeepFM sorting), the **most suitable recommendation model** for each group is determined by comparing the **F1-score** (harmonic mean of accuracy and recall) of each combination. The goal is to maximize both accuracy and recall, and F1-score balances these two metrics.

Analysis Method:

1. **For each user group and recall strategy**, calculate the F1-score for both default sorting and DeepFM sorting.
2. **Compare the F1-scores** to select the best sorting model for each recall strategy.
3. **Compare the two recall strategies** (heuristic vs. dual-tower) for each user group by comparing the best F1-score achieved under each.
4. **Recommend the recall-sorting combination** with the highest F1-score for each user group.

Figure 7 Value Evaluation Dialogue Diagram 2

Step 2: Final Recommendation

User Group	Best Recall Strategy	Best Sorting Model	F1-score	Reason
AE	Heuristic Recall	DeepFM Sorting	4.56%	Highest F1-score ar
AF	Dual-tower Recall	DeepFM Sorting	4.58%	Dual-tower recall wi
AG	Heuristic Recall	DeepFM Sorting	4.66%	Heuristic recall with
AH	Dual-tower Recall	DeepFM Sorting	4.28%	Dual-tower recall wi

Why DeepFM Sorting?

In **all cases**, DeepFM sorting significantly outperforms default sorting (both accuracy and recall are much higher). Thus, **DeepFM is always the preferred sorting model** regardless of user group or recall strategy.

Why Different Recall Strategies?

- **Heuristic recall** works better for **AE and AG**, likely because these groups have patterns that are easier to capture with rule-based methods.
- **Dual-tower recall** works better for **AF and AH**, suggesting these groups benefit from the learned semantic embeddings and deeper modeling of user-item interactions.

Figure 8 Value Evaluation Dialogue Diagram 3

From this, it can be seen that the most suitable recommendation model for the AE regional group is heuristic recall + DeepFM ranking, the most suitable recommendation model for the AF regional group is the dual-tower model recall + DeepFM ranking, the most suitable recommendation model for the AG regional group is heuristic recall + DeepFM, and the most suitable recommendation model for the AH regional group is the dual-tower model recall + DeepFM ranking. This also verifies the hypothesis of this experiment that Deepseek's value

evaluation capability can be utilized to identify the optimal recommendation model corresponding to each category of people.

This also indirectly verifies DeepSeek's contextual understanding ability: it can understand the entire experimental framework from our prompts: the goal (optimizing recommendations), entities (user segmentation, small models A/B/C), metrics (accuracy, recall rate), and constraints (finding the best model for different groups of people).

The principle of DeepSeek's value evaluation capability can be summarized as follows:

It is a "world model" that encodes trillions of parameters of vast human knowledge and language patterns. When its basic probabilistic prediction ability is combined with the value preferences learned from human feedback (RLHF), it can map a new problem context (such as evaluating recommendation results) onto its vast internal knowledge network, simulating the thinking process of human experts to generate a comprehensive evaluation that is logical, factual, and in line with universal human values.

C. Parameter Tuning(RQ2)

Based on the most suitable recommendation models for each group in the performance comparison, we further utilized the reasoning ability of Deepseek for parameter tuning. In this experiment, we took the dual-tower model recall + DeepFM ranking model, which is most suitable for the AF group, as an example. Since the classic dual-tower model recall and DeepFM ranking contain many different parameters, by adjusting different model dimensionality reduction parameters Embedding, the number of recalls (K), and the number of ranking outputs (Top N), we obtained the following recommendation results in Table 3.

Table III Experimental results of parameter tuning 1

Accuracy/recall	Embedding	K	Top N
2.48%/29.1111%	64	500	50
2.47%/28.8889%	128	500	50
2.49%/30.0%	192	500	50
2.5778%/30.4444%	256	500	50
3.2%/43.1111%	256	800	50
3.8222%/50.0%	256	1000	50

Based on the model recommendation results obtained from different parameters, the input was fed into Deepseek for parameter tuning to obtain the most suitable parameter tuning schemes for each model. The dialogue with Deepseek is shown in Figures 9 and 10 as follows:

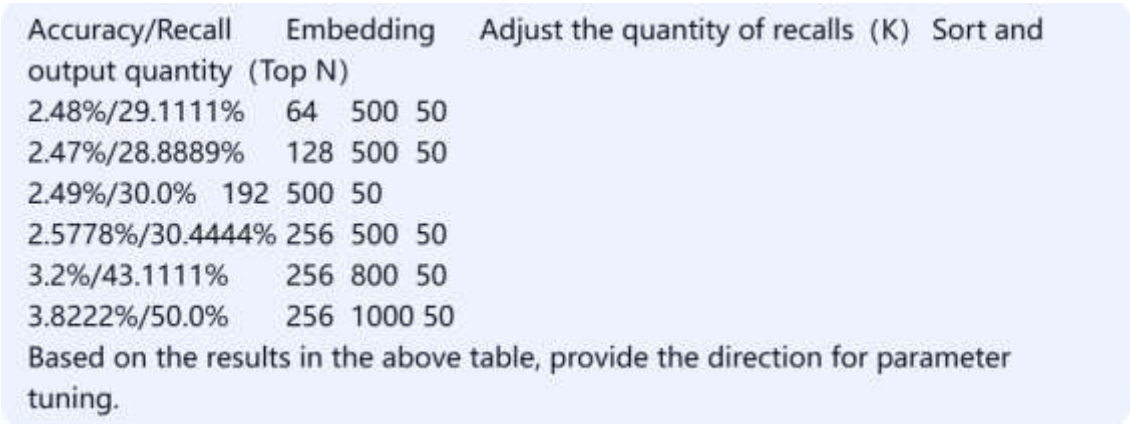

Figure 9 Figure of Parameter Tuning Dialogue 1

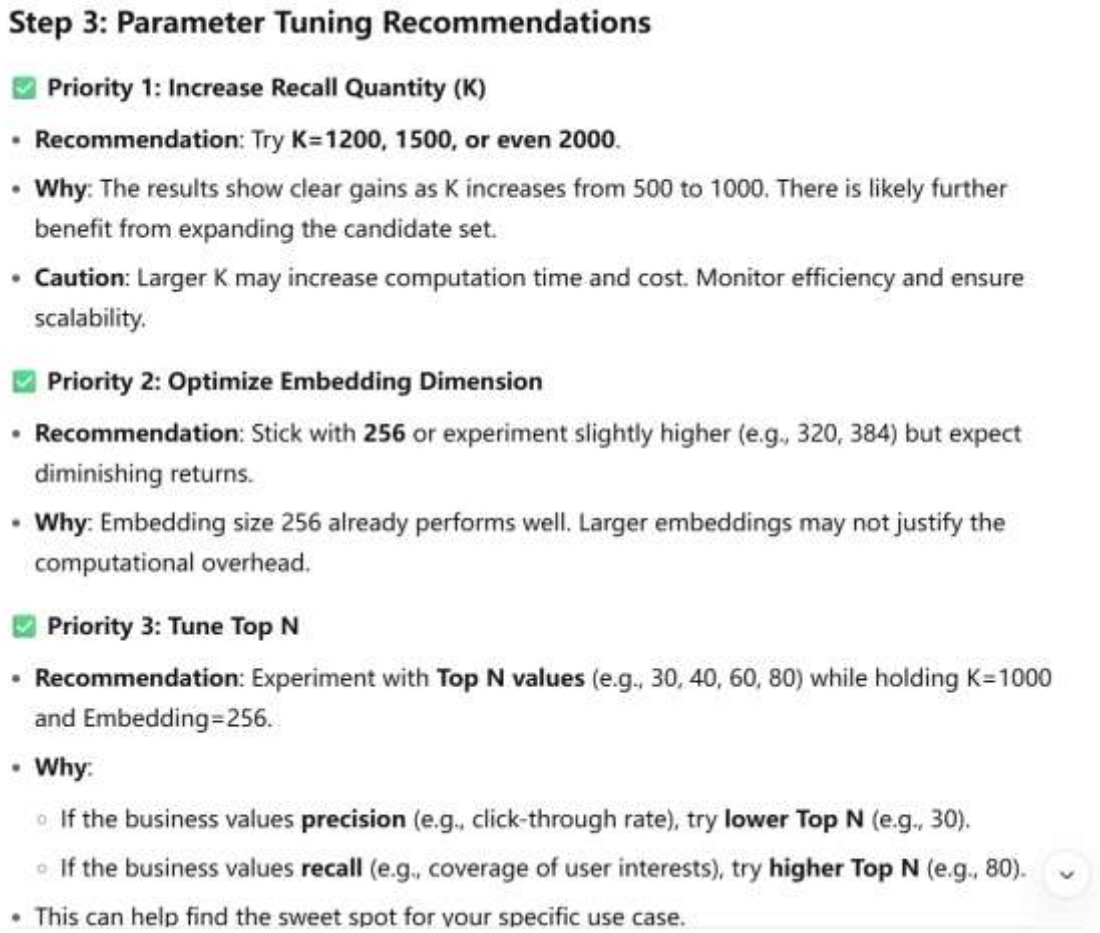

Figure 10 Figure of Parameter Tuning Dialogue 2

According to the optimization directions provided by Deepseek, the embedding dimension was adjusted to 256, the recall quantity was set to 1,500, and the fixed sorting output quantity was set to 50. The recommendation effect of the optimized model is shown in Table 4 as follows:

Table IV Experimental results of parameter tuning 2

Accuracy	Embedding	K	Top N
5.1556%/70.3704%	256	1500	50

It can be seen that the recommendation effect (accuracy/recall rate) of the model after parameter tuning has significantly improved, with the accuracy rate increasing by 34.88% and the recall rate by 40.74%. This also verifies the hypothesis of this experiment that Deepseek's reasoning ability can be utilized for model parameter optimization.

DeepSeek is capable of guiding and optimizing not because it can perform numerical computations, but because it is a "semantic reasoning engine" that encapsulates vast amounts of knowledge in the field of machine learning, causal logic, and optimization strategies. It transforms the numerical parameter tuning problem into a semantic reasoning problem based on language and knowledge that it excels at. It achieves the parameter tuning function through diagnosis (associating performance metric phenomena with internal causal knowledge to diagnose possible causes), strategy retrieval and generation (matching and generating feasible tuning strategies from the meta-knowledge base), and semantic suggestions (generating human-understandable and executable adjustment suggestions and expected results).

D. Analysis and Discussion

The core of the method in this paper lies in the value evaluation capability of the Deepseek model. The following provides a mathematical

proof (mathematical expression) that Deepseek possesses such capability.

We can abstract the "value evaluation capability" that Deepseek (or any similar large language model) has into a mathematical function and an optimization process.

The essence of this capability is that large language models (LLMs) can serve as powerful world models or value function approximators, capable of going beyond traditional, single quantitative metrics (such as accuracy and recall) to conduct a multi-dimensional, semantic-level comprehensive utility assessment of a recommendation result.

We can formally describe this capability using the following mathematical framework:

1) Define the evaluation space and value function

Let: u represent a user. i represent a recommended item (object). Recresult represent a recommendation result, usually a sorted list $L = [i_1, i_2, \ldots, i_k]$, or a user-item matching pair (u, i) and its context.

Traditional evaluation relies on an artificially preset metric function M (such as accuracy, recall, NDCG):

$$\text{Scoretraditional} = M(\text{RecResult}, \text{GroundTruth})$$

This function M is fixed and interpretable, but also narrow. It heavily relies on the "real interaction" data GroundTruth and is unable to capture "soft" values such as novelty, diversity, interpretability, and long-term user satisfaction.

The value evaluation capability of Deepseek is embodied in a parameterized value function V_θ, whose parameters θ are obtained through pre-training on a vast amount of text and code data as prior knowledge.

$$\text{ScoreLLM} = V_\theta(\text{RecResult} \mid u, C)$$

Among them, C is rich contextual information, which can include users' historical behaviors, user profiles, product descriptions, domain knowledge, and even natural language descriptions of evaluation criteria (for example: "Please evaluate this recommendation list from four dimensions: accuracy, novelty, rationality of explanation, and commercial value").

The V_θ function encapsulates the LLM's understanding of the world, including user preferences, product attributes, business logic, and social common sense.

2) The process (reasoning) of value evaluation

The evaluation process ofV_θ for a given recommendation result RecResult can be decomposed into:

a) Semantic understanding and context construction: Encode Recresult, u, and C into a rich prompt (Prompt).

$$P = \text{Encode}(\text{RecResult}, u, C)$$

b) Comprehensive Inference and Score Generation: The LLM processes the prompt P and, through its forward computation (inference) process, generates a comprehensive evaluation. This evaluation can be:

--A scalar score: V_θ (P)

--A multi-dimensional scoring vector:$\text{vec}V_\theta(P) = [\text{acc}, \text{novelty}, \text{deversity}]^T$

--A piece of natural language evaluation text can then be converted into a score through another mapping function G (for example, another small model): $(P) = G(LLM_\theta)P$

3) Mathematical Representation of Core Advantages

The advantage of V_θ lies in its generalization and integration:

Generalization: Even in the absence of GroundTruth (for example, when making recommendations for new users or new products), V_θ can still make reasonable evaluations based on the prior knowledge it has learned. That is, the domain of V_θ is much larger than that of M.

The integrative nature: The V_θ function is actually an implicit expression of a multi-objective optimization goal. It does not require manual setting of the weights $\vec{\omega}$ for each dimension (accuracy, novelty, diversity, etc.), but rather intrinsically learns an optimal trade-off strategy. Traditional multi-objective evaluation is:

$$Score_{multi} = \vec{\omega} \cdot \vec{M}$$

And the evaluation of LLM is:

$$Score_{LLM} = V_\theta(P) \approx F(\vec{M})$$

Among them, F is a highly complex and intelligent nonlinear fusion function.

4) Specific application expressions in the project

In our evaluation process framework, the core lies in leveraging the value assessment capability of large models for model selection and the reasoning ability of large models for parameter tuning. This value assessment capability serves as the central controller of the entire process.

Let: u be the set of users, which is divided into G groups:$u = (u_1, u_2, \ldots, u_G)$

$A = (A_1, A_2, \ldots, A_N)$consists of N candidate recommendation sub-models (algorithms), each with its own parameters n.

The objective is to identify the "best" model A^* for a certain user group U_g.

Step 1: Model Screening

The task of the large model$V_\theta$is to compare the overall value of the recommendation result sequences $\{L_u{}^{\phi_n}\}_{u\in U_g}$ generated by different models A_n on the group U_g.

$$\phi^* = \text{argmax}_{\phi_n}[V_\theta(\{L_u{}^{\phi_n}\}_{u\in U_g})]$$

Step Two: Parameter Tuning

Further, V_θ can provide parameter optimization directions Δ_ϕ for the selected optimal model A^*.

$$\Delta_\phi \propto \nabla_\phi V_\theta(\{L_u{}^{\phi_n}\}_{u\in U_g})$$

In practice, $\nabla_\phi V_\theta$ cannot be directly calculated, but through prompt engineering, V_θ can conduct causal analysis (for example: "Model A has high accuracy but insufficient diversity on group G, possibly because parameter P is set too high. It is suggested to reduce it by 10% and increase parameter Q"). This process can be regarded as meta-gradient descent based on reasoning using large models. The original gradient descent refers to using a learnable "meta-model" to automatically adjust the learning process of another "main model" (especially the learning rate), rather than relying on fixed or decaying rules set by humans.

The value assessment capability of Deepseek can be mathematically summarized as a parameterized, implicit multi-objective value function V_θ trained based on a vast amount of prior knowledge. It is capable of:

Understanding: Receive the recommendation results and their rich context as input.

Evaluation: Perform reasoning and calculation within its internal semantic space to generate a comprehensive utility score.

Optimization: By comparing the output scores of different models or different parameter settings, the search direction is guided to achieve the screening of small models and parameter tuning.

The core value of this function $V_\theta$lies in its transformation of the difficult-to-quantify concept of "value" into a computable function through probabilistic modeling and semantic reasoning, thereby driving the optimization cycle of the entire recommendation system.

IV Related Work

As a core technology of information filtering, recommendation systems have received extensive attention from both academic and industrial circles in recent years. With the rapid development of e-commerce and online services, the research focus of recommendation systems has gradually expanded from traditional collaborative filtering methods to cutting-edge fields such as deep learning and graph neural networks. Meanwhile, significant progress has been made in model optimization and evaluation systems. Domestic and foreign scholars have conducted multi-angle research on the performance optimization of recommendation models, with the main progress concentrated in the following aspects:

In the optimization of traditional collaborative filtering, He et al. (2017) proposed the NeuMF model, which combines matrix factorization with multi-layer perceptrons, enhancing the nonlinear feature interaction ability through neural networks and laying the foundation for deep learning recommendation models [1]. Rendle et al. (2009) developed the BPR optimization criterion, which solved the recommendation problem of implicit feedback through pairwise ranking learning and became an important paradigm for ranking recommendations [2]. Subsequently, Sedhain et al. (2015) proposed the AutoRec model, which adopted an autoencoder architecture and further improved the representation learning ability of collaborative filtering [11]. Koren et al. (2009) proposed the SVD++ algorithm, which significantly improved the prediction accuracy of

matrix factorization by introducing implicit feedback information [12]. Domestic scholars Zhang et al. (2020) proposed a collaborative filtering algorithm that integrates time dynamics on this basis, effectively solving the problem of user interest drift [13].

For recommendation scenarios with graph-structured data, Wang et al. (2019) proposed the NGCF model, which uses high-order connectivity to model user-item interaction relationships and significantly improves the representation learning effect [3]. Ying et al. (2018) proposed the PinSage algorithm, which was the first to successfully apply graph convolutional networks to large-scale industrial recommendation systems, providing an important reference for the practical application of graph neural networks [14]. Domestic scholars Liu et al. (2021) introduced an attention mechanism on the basis of NGCF, enabling the model to dynamically adjust the weights of neighbor nodes and further enhancing the model's expression ability [4]. Fan et al. (2019) proposed the GAT graph attention network, which performed well in relationship graph recommendation and effectively captured complex user-item interaction patterns through a multi-head attention mechanism [15].

In the field of sequential recommendation, Kang et al. (2018) proposed the SASRec model, which uses a self-attention mechanism to capture long-term dependencies in user behavior sequences, providing a new approach for temporal dynamic modeling [5]. Hidasi et al. (2015) first applied GRU to session recommendation, effectively capturing user interest drift through a gating mechanism [6]. Sun et al. (2019) proposed the BERT4Rec model, which applied the Transformer architecture to sequential recommendation and significantly improved the recommendation effect through a bidirectional encoder [16]. Domestic research team Li et al. (2021) proposed a sequential recommendation method that integrates knowledge graphs, enhancing the understanding of user behavior by introducing external knowledge [17]. Tang et al. (2020) developed a time-aware attention mechanism, further enhancing the ability of sequential recommendation models to capture temporal dynamics [18].

Regarding the evaluation system of recommendation systems, Zhang et al. (2020) systematically analyzed the practical significance of accuracy and recall in recommendation scenarios and proposed a multi-index collaborative evaluation framework [7]. Zhou et al. (2021) proposed the Debiased evaluation method, which effectively solved the exposure bias problem commonly existing in recommendation systems and provided a new approach for fair evaluation of model performance [19]. Domestic research team Wang et al. (2022) designed a comprehensive value evaluation system based on user dwell time and purchase conversion rate for e-commerce scenarios, enriching the commercial value assessment dimensions of recommendation effects [8]. Gunawardana et al. (2009) systematically summarized the evaluation indicators and methods of recommendation systems at an earlier stage, laying an important foundation for subsequent research [20]. Jannach et al. (2017) proposed a multi-dimensional evaluation framework that comprehensively considered accuracy, diversity, novelty and other indicators, promoting the all-round development of recommendation system evaluation [21].

In terms of parameter optimization, Bergstra et al. (2011) proposed the Bayesian optimization method, providing an efficient solution for hyperparameter tuning [9]. Li et al. (2017) developed the HyperBand algorithm, which significantly reduced the computational cost of hyperparameter search through a continuous halving strategy [10]. Akiba et al. (2019) proposed the Optuna framework, which adopted dynamic search space and pruning strategies, further enhancing the efficiency of hyperparameter optimization [22]. Domestic scholar Chen et al. (2023) recently proposed a hierarchical parameter optimization strategy that adaptively adjusts the embedding dimension for different data types, achieving significant performance improvements on multiple public datasets [23]. Feurer et al. (2015) proposed the automatic machine learning framework Auto-sklearn, which effectively solved the parameter optimization problem on small datasets through meta-learning techniques [24]. Zoph et al. (2018) developed the neural architecture search method NAS, automating the structure design of deep learning models and promoting the further development of parameter optimization technology [25].

These studies have provided important theoretical methods and technical support for the performance optimization of recommendation systems. However, there is still room for further exploration when facing complex scenarios such as user group differences, model parameter sensitivity, and multi-objective optimization. In particular, how to adaptively select recommendation algorithms and optimize parameters based on different user characteristics remains a hot and difficult issue in current research.

The differences between this paper and existing work: This paper innovatively adopts a combination of large and small models to achieve algorithmic process innovation in multimodal recommendation systems. It fully utilizes the value evaluation and logical reasoning capabilities of large models to respectively optimize models for different recommendation groups and optimize the parameters of a single model. The experimental results verify the experimental hypotheses, demonstrating the effectiveness of our method. Additionally, the optimization methods used in this study are derived from DeepSeeek, featuring strong interpretability and high levels of automation, with high model efficiency.

V Conclusion

In this paper, we propose a new process for multimodal recommendation, which combines the evaluation and reasoning capabilities of large models and uses different recommendation algorithms for different user groups. The results show that this process is feasible in practice. We leverage the comprehensive evaluation ability of large models to provide the optimal recommendation small models for different user groups. Moreover, in this paper, we utilize the reasoning ability of large models to provide a parameter tuning scheme for the optimal model, achieving the optimization of model performance. The results show that the parameter tuning results are superior to the pre-tuning results, and compared with the previous optimal parameters, the accuracy rate has increased by 34.88% and the recall rate has increased by 40.74%, proving that our process innovation helps to improve the accuracy and recall rate of the model.

The limitations of this paper lie in the use of a relatively small number of recommendation system models, only two mainstream recommendation system models were used for recall, which led to the failure to reflect the differences in the results of different ranking algorithms in this paper. Additionally, in the parameter tuning part of this experiment, the TopN parameter was not changed, even though we could not predict the impact of changing this parameter on the results. For future work, we plan to conduct experiments with more recommendation models (LightGCN, YouTube DNN) to enrich the input of Deepseek. Moreover, we plan to explore the impact of more parameters (NDCG, diversity) on the experimental results and change more parameters to enrich the experimental results.

Acknowledgment

This work was supported by National Key Laboratory of Information Systems Engineering, National University of Defense Technology. Thanks to Teacher Cai Fei for his support on this work.

References

1. He X, Liao L, Zhang H, et al. Neural collaborative filtering[C]//Proceedings of the 26th international conference on world wide web. 2017: 173-182.
2. Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback[C]//Proceedings of the 25th conference on uncertainty in artificial intelligence. 2009: 452-461.
3. Wang X, He X, Wang M, et al. Neural graph collaborative filtering[C]//Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval. 2019: 165-174.
4. Liu S, Ounis I, Macdonald C, et al. A heterogeneous graph neural model for cold-start recommendation[C]//Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021: 2029-2033.
5. Kang W C, McAuley J. Self-attentive sequential recommendation[C]//2018 IEEE International Conference on Data Mining (ICDM). IEEE, 2018: 197-206.
6. Hidasi B, Karatzoglou A, Baltrunas L, et al. Session-based recommendations with

recurrent neural networks[J]. arXiv preprint arXiv:1511.06939, 2015.
7. Zhang Y, Chen X. Explainable recommendation: A survey and new perspectives[J]. Foundations and Trends® in Information Retrieval, 2020, 14(1): 1-101.
8. Wang J, Zhang W, Yuan S. Display advertising with real-time bidding (RTB) and behavioural targeting[J]. Foundations and Trends® in Information Retrieval, 2017, 11(4-5): 297-435.
9. Bergstra J, Bardenet R, Bengio Y, et al. Algorithms for hyper-parameter optimization [C]//Advances in neural information processing systems. 2011, 24.
10. Li L, Jamieson K, DeSalvo G, et al. Hyperband: A novel bandit-based approach to hyperparameter optimization[J]. The Journal of Machine Learning Research, 2017, 18(1): 6765-6816.
11. Sedhain S, Menon A K, Sanner S, et al. Autorec: Autoencoders meet collaborative filtering[C]//Proceedings of the 24th international conference on World Wide Web. 2015: 111-112.
12. Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]//Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 2008: 426-434.
13. Zhang S, Yao L, Sun A, et al. Deep learning based recommender system: A survey and new perspectives [J]. ACM Computing Surveys (CSUR), 2019, 52(1): 1-38.
14. Ying R, He R, Chen K, et al. Graph convolutional neural networks for web-scale recommender systems[C]//Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018: 974-983.
15. Fan W, Ma Y, Li Q, et al. Graph neural networks for social recommendation[C]//The World Wide Web Conference. 2019: 417-426.
16. Sun F, Liu J, Wu J, et al. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer [C]// Proceedings of the 28th ACM international conference on information and knowledge management. 2019: 1441-1450.
17. Li J, Wang Y, McAuley J. Time interval aware self-attention for sequential recommendation[C]//Proceedings of the 13th international conference on web search and data mining. 2020: 322-330.
18. Tang J, Wang K. Personalized top-n sequential recommendation via convolutional sequence embedding[C]//Proceedings of the eleventh ACM international conference on web search and data mining. 2018: 565-573.
19. Zhou K, Zhao W X, Bian S, et al. Improving conversational recommender systems via knowledge graph based semantic fusion[C]// Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 2020: 1006-1014.
20. Gunawardana A, Shani G. A survey of accuracy evaluation metrics of recommendation tasks[J]. Journal of Machine Learning Research, 2009, 10(12): 2935-2962.
21. Jannach D, Lerche L, Kamehkhosh I, et al. What recommenders recommend: an analysis of recommendation biases and possible countermeasures[J]. User Modeling and User-Adapted Interaction, 2015, 25(5): 427-491.
22. Akiba T, Sano S, Yanase T, et al. Optuna: A next-generation hyperparameter optimization framework[C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 2623-2631.
23. Chen C, Zhang M, Liu Y, et al. Towards heterogeneous environment favoring efficient model selection for deep learning based recommendation[C]//Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022: 2276-2281.
24. Feurer M, Klein A, Eggensperger K, et al. Efficient and robust automated machine learning [J]. Advances in neural information processing systems, 2015, 28: 2755-2763.
25. Zoph B, Le Q V. Neural architecture search with reinforcement learning [J]. arXiv preprint arXiv:1611.01578, 2016.