Competition 6: Prediction and Recommendation of Users' Potential Book Borrowing Outcomes Based on University Library Loan Data

1. Competition Background

With the continuous advancement of digitalization in university libraries, borrowing data has become a crucial resource reflecting users' reading interests and behavioral habits. Libraries have accumulated vast amounts of borrowing records. How to leverage this data to uncover users' latent needs and provide accurate book recommendations has become a key issue in enhancing library service quality and user experience. By employing artificial intelligence and data mining technologies, developing efficient book recommendation algorithms can not only optimize resource allocation but also offer personalized reading suggestions to users, thereby promoting the development of campus reading culture.

2. Competition Application Scenario

In the daily operations of university libraries, users such as students and teachers access book resources through the borrowing system. Their borrowing records contain rich information about their interest preferences and behavioral patterns. For example, a user may frequently borrow books on computer science or concentrate on borrowing textbooks related to specific courses during a particular semester. Traditional manual recommendations or simple classification methods struggle to fully capture user needs, often resulting in recommendations that do not align with user interests. Recommendation algorithms based on borrowing data can analyze historical borrowing records, book categories, borrowing times, and other features to predict users' potential reading needs and recommend books that match their interests. This enhances borrowing efficiency and user satisfaction while providing data support for library collection management and procurement decisions.

3. Competition Information

This challenge is presented by the MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, designed in collaboration with actual library data and needs.

4. Competition Task

Participants are required to use the library borrowing dataset provided by the organizers to design and implement artificial intelligence algorithms that predict users' potential future book borrowing needs and provide personalized recommendations.

Specific tasks include:

1. User Interest Modeling: Analyze users' reading preferences and behavioral patterns based on their historical borrowing data to construct user interest profiles.

2. Book Recommendation Prediction: Predict books that users may be interested in the future based on their interest profiles and book information, and generate a recommendation list.

5. Dataset and Data Description

5.1 Data Source

The data comes from real library borrowing records, covering the borrowing behaviors of various users, including students and teachers, ensuring diversity and representativeness. The data has been anonymized to protect user privacy.

5.2 Data Overview

A total of 88,378 borrowing records from 1,451 users and 58,549 books are provided for model training and validation. The test set contains approximately 14,510 records for final evaluation.

The interaction data for this competition will be released in phases based on user groups. During the preliminary stage, complete interaction data for 600 users, along with all user and book information, will be provided. The semi-final stage will add interaction data for 400 new users, and the final stage will release the remaining interaction data for all users. User and book information will be provided completely only during the preliminary stage and will not be supplemented in subsequent stages.

Sample data can be accessed via:

https://pan.baidu.com/s/1azEZ-oYNnQTnyEbnKphk5w?pwd=q7sh for, the official data will be available after registration.

5.3 Data Format

The dataset is stored in CSV format and divided into three files:

The details are as follows:

(1) book.csv: Book information, including book_id (Book ID), title (Book Name), author, publisher, first-level category, and second-level category.

(2) inter.csv: Borrowing interaction records, including inter_id (Interaction ID), user_id (User ID), book_id (Book ID), borrowing time, return time, renewal time, and renewal count.

(3) user.csv: User information, including borrower (User ID), gender, DEPT (Department), grade, and type (e.g., undergraduate/graduate).

6. Algorithm Design Requirements

6.1 Model Type

Participants are encouraged to adopt machine learning or deep learning algorithms, such as Collaborative Filtering (CF), Matrix Factorization (MF), Deep

Neural Networks (DNN) and their variants. They can also leverage large language models (LLM) to extract content and understand semantics from the text in the book dataset, mine key information in the text, and further enrich the recommendation dimensions. Enhance the accuracy and personalization of the recommendation system to provide users with book recommendation services that better meet their needs. For example, content-based recommendation models can be used to analyze book features, while Graph Neural Networks (GNN) or Transformer models can capture users' borrowing behaviors.

6.2 Innovation

Innovative algorithm architectures or improvements to existing recommendation algorithms are encouraged to enhance the accuracy and personalization of book recommendations. Examples include designing novel feature extraction methods to better represent user interests and book attributes, or employing multi-dimensional data fusion (e.g., combining user information, borrowing time, and book categories) to optimize recommendation effectiveness.

6.3 Scalability

The algorithm should have good scalability, be capable of running on computing devices with different configurations, and perform stably when processing large-scale borrowing data. For instance, the algorithm should be capable of running efficiently on both regular workstations and cloud servers, and its performance should not decline significantly when the number of users or borrowing records increases.

7. Performance Metrics Requirements

7.1 Primary Metrics

(1) Precision (P): Measures the proportion of books in the library's recommendation list that were actually borrowed by users (i.e., ground-truth borrowings in the test set) out of all recommended books. It reflects the accuracy of the recommendations, indicating how many of the recommended books users truly end up borrowing.

(2) Recall (R): Measures the proportion of books in the library's recommendation list that were actually borrowed by users out of all books actually borrowed by users in the test set. It reflects the system's ability to capture users' real borrowing behaviors, indicating how many of the actually borrowed books were successfully recommended.

7.2 Secondary Metrics

(1) Model Size: The file size of the trained model, serving as an important

indicator of model complexity and storage requirements. A smaller model size suggests lower complexity, facilitating deployment across different devices and environments.

8. Functional Requirements

8.1 Accuracy

The algorithm must achieve high accuracy in predicting users' potential borrowing needs, ensuring that recommended books closely match user interests. On the test set, the F1 Score of the recommendation results must reach at least 0.12.

8.2 Reliability

The algorithm should operate stably and deliver reliable recommendations when processing data from users of different departments, grades, and borrowing habits. Even in the presence of noise in borrowing records or anomalous behaviors, the algorithm should not exhibit significant performance fluctuations, maintaining the accuracy and stability of its recommendations.

8.3 Interpretability

The algorithm should possess a certain level of interpretability, providing users or library administrators with explanations for the recommendation results. For example, it can employ visualization techniques to show how users' interests align with the recommended books, or conduct feature importance analysis to illustrate which borrowing features the model relied on when generating the recommendations. Such interpretability helps users understand the recommendation process and enhances their trust in the system.

8.4 Robustness

The algorithm must demonstrate strong robustness against outliers and missing values in the data. Even when some borrowing records have missing timestamps or user data is incomplete, the algorithm should still ensure the reliability of its recommendations without suffering significant performance degradation due to minor data imperfections.

8.5 Multimodal Fusion Capability

If participants employ multimodal data fusion methods (e.g., integrating user information, book categories, borrowing times, etc.), the algorithm should effectively consolidate different types of data. The fusion should significantly enhance the accuracy and personalization of recommendations, demonstrating efficient utilization of multi-source information.

9. Development Environment

9.1 Programming Language

Python is required. It is recommended to use Python 3.6 or higher due to its rich ecosystem of scientific computing libraries and machine learning framework support.

9.2 Machine Learning Framework

The use of TensorFlow 2.x or PyTorch 1.x is recommended. These frameworks are widely adopted in the field of machine learning due to their high computational efficiency and comprehensive APIs, which facilitate model building, training, and deployment.

9.3 Computing Resources

Participants may use local workstations or cloud computing platforms for development and training. For local workstations, it is recommended to equip NVIDIA GPUs (e.g., GTX 10 series or above, or RTX series) to accelerate model training. For cloud platforms, options such as Alibaba Cloud Tianchi, AWS SageMaker, or Google Colab can be selected, offering flexible computing resource configurations.

9.4 Dependency Libraries

The RecBole framework can be used, along with its supporting libraries, including PyTorch (for model training), NumPy and Pandas (for data processing), as well as RecBole's built-in utility modules, to facilitate rapid development and experimentation of recommendation systems.

10. Evaluation Criteria

10.1 Input Data Format Requirements

Participants' algorithms must correctly read the CSV-format borrowing datasets provided by the organizers, which include the following files:

book.csv (Book Information): should parse fields such as book ID, title, author, publisher, and category.

inter.csv (Borrowing Interaction Records): should extract information including borrower (user ID), book ID, borrowing time, return time, renewal time, and number of renewals.

user.csv (User Information): should accurately read user attributes such as gender, department, grade, and user type.

train_data.csv (Training Set), valid_data.csv (Validation Set), test_data.csv (Test Set): should process borrowing records in these datasets for model training and evaluation.

The algorithm must be able to effectively parse these CSV files and fully extract

all relevant fields to provide reliable data support for subsequent recommendation tasks.

10.2 Output Data Format Requirements

Participants must save their prediction results in a CSV file and compress the submission into a ZIP file. Each row in the CSV file should represent a single predicted recommendation record and contain two fields in order: “user_id” and “book_id.”

“user_id”: uniquely identifies the user. The data type should be a string, and the value must match the user ID format provided in the dataset. The ID must be valid, without any fabricated or incorrect entries.

“book_id”: represents the book ID. The data type should be a string, consistent with the format of book IDs in the competition dataset, and must correspond to valid entries.

Recommendation constraint: each user can only be recommended one book. That is, the same “user_id” must appear only once in the CSV file.

The CSV file must be correctly formatted, with no extra header rows (if a header is included, only the first row is allowed, and it must be exactly “user_id,book_id”). The file must not contain blank rows, invalid characters, or formatting errors, in order to ensure proper evaluation. The file should be encoded in UTF-8 to guarantee character compatibility.

10.3 Evaluation Metrics

The evaluation is based on comparing the recommendation results with the users’ actual borrowing records in the test set. Both Precision (P) and Recall (R) are used, with the following definitions:

10.3.1 Precision (P)

Definition: measures the proportion of books in the library’s recommendation list that were actually borrowed by users (i.e., ground-truth borrowings in the test set). It reflects the accuracy of the recommendations — that is, how many of the recommended books users truly borrowed.

Formula:

$$P = \frac{TP}{TP + FP}$$

where TP (True Positives) is the number of recommended books that were actually borrowed by users, and FP (False Positives) is the number of recommended books that were not borrowed.

10.3.2 Recall (R)

Definition: measures the proportion of books in the library's recommendation list that were actually borrowed by users, relative to all books the users borrowed in the test set. It reflects the system's ability to capture users' real borrowing behavior — that is, how many of the actually borrowed books were successfully recommended.

Formula:

$$R = \frac{TP}{TP + FN}$$

where TP (True Positives) is the number of recommended books that were actually borrowed, and FN (False Negatives) is the number of books borrowed by users in the test set but not recommended by the system.

10.3.3 Final Score

F1 Score

Definition: the F1 score is the harmonic mean of Precision and Recall, providing a balanced evaluation of the recommendation system's performance. It avoids the bias of relying solely on Precision or Recall.

Formula:

$$F1 = \frac{2 \times P \times R}{P + R}$$

10.4 Valid Results

For each competition phase, an F1 score higher than 0.0055 is considered a valid result. The threshold of 0.0055 ensures that the recommendation algorithm has practical application value.

The prize allocation baseline is determined by the number of teams achieving valid results. The threshold itself may be modified by the organizers based on the overall performance of participants.

11. Problem-Solving Approach

11.1 Data Preprocessing

Extracting interaction data and constructing a graph structure.

Borrowing records are processed to extract the user ID and book ID from each record, forming user–book interaction pairs. By representing the data as a graph, users and books are treated as nodes, while borrowing behaviors are regarded as edges connecting them. This yields a user–book bipartite graph. Analyzing this graph structure helps uncover similarities among users and associations among books, providing richer feature information for the recommendation algorithm.

Incorporating user and book information into LLM prompts.

From the user information table, attributes such as gender and major can be collected. For the book recommendation task, these attributes are integrated into prompts in a meaningful way. For example, one may construct a prompt such as "Recommend relevant books for a male student majoring in computer science, class of 2023." Carefully designed prompts can guide LLM-based recommendation services to generate results more closely aligned with user needs. Alternatively, a sequential recommendation approach may also be adopted.

11.2 Model Training

Select an appropriate machine learning framework (e.g., TensorFlow or PyTorch) to build the model, and configure reasonable training parameters such as learning rate, number of iterations, and batch size. During training, apply cross-validation with the validation set to evaluate and fine-tune the model, preventing overfitting.

11.3 Model Ensemble and Optimization

Consider combining multiple models of different structures or training stages, using methods such as voting or weighted averaging to integrate predictions and improve overall accuracy. Meanwhile, models can be optimized based on performance evaluation metrics, for instance, by adjusting the model architecture or increasing the amount of training data.

12. Reference Resources

12.1 Books

(1) *Deep Learning*, authored by Ian Goodfellow, Yoshua Bengio, and Aaron Courville. This book systematically introduces the fundamental concepts, model architectures, and training methods of deep learning, providing substantial guidance for understanding and applying neural networks.

(2) *Deep Learning with Python*, authored by François Chollet. Through numerous code examples, this book explains in detail how to develop deep learning models using Python and the Keras framework, making it suitable for beginners to get started quickly.

12.2 Online Courses

(1) "Deep Learning Specialization" on Coursera, taught by Professor Andrew Ng. The course covers several key areas of deep learning, including the foundations of neural networks, convolutional neural networks, and recurrent neural networks, with rich content and strong practical orientation.

(2) "Introduction to Artificial Intelligence" on edX. This course provides

introductory knowledge of artificial intelligence and machine learning, including algorithm principles, model training, and application cases, helping participants build a comprehensive knowledge base.

12.3 Academic Papers

Participants are encouraged to search academic databases for the latest research papers on recommender systems, such as “Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning”, to learn about state-of-the-art techniques and methodologies in the field.

Attention should also be given to papers presented at leading AI conferences (e.g., AAAI), to stay updated with the latest research trends and innovations.

13. Submission Requirements

13.1 Algorithm Code

Submit complete Python code covering all stages, including data preprocessing, model training, and recommendation result generation.

The code must follow the PEP8 coding standard and include detailed comments and documentation to ensure that reviewers can understand and execute it.

13.2 Technical Report

Submit a technical report in PDF format, with a minimum length of 3,000 words. The content should include algorithm design ideas, model architecture diagrams, experimental settings (e.g., hyperparameter choices, data processing methods), performance analysis, as well as a discussion of innovations and limitations.

13.3 Recommendation Data

Submission data must be named .zip. The submission.csv file should be compressed into a ZIP file, where the first column is user_id, and the second column is book_id.

13.4 Model File

Submit the trained model files along with documentation for loading and usage. This must include the runtime environment (e.g., Python version) and dependency libraries.

The model must be able to run in the specified environment and generate recommendation results successfully.

14. Updates and Q&A

14.1 Updates and Inquiries

The competition task description will not be updated. For any task-related questions, participants may contact the responsible person via email:

nuaa_niurongbing@nuaa.edu.cn

14.2 Intelligent Assistant

An online intelligent assistant will be established for each task to facilitate participants' inquiries and provide timely assistance.

15. Competition Process and Award Settings

15.1 Registration Phase

Participants must complete registration on the official competition website and submit personal or team information.

15.2 Preliminary Round

Participants use the training dataset provided by the organizers to design algorithmic models and validate/tune their methods using the preliminary test set. During this stage, the number of daily submissions is unlimited, however the preliminary leaderboard is refreshed every hour.

15.3 Semifinal (Provincial Round)

After the preliminary round, the semifinal round will commence, with the semifinal dataset made available for download. Only teams that submitted valid results in the preliminary round may advance to the semifinal. During this stage, participants must tune their models using the semifinal dataset and submit inference results on the semifinal test set. The semifinal lasts for three days, with each team allowed up to two submissions per day. The leaderboard is refreshed every hour.

15.4 Announcement of Semifinal Results

Semifinal results will be published on the official competition website. The number of teams advancing to the semifinal serves as the baseline for awarding prizes. In accordance with the award ratio set for the provincial round, first, second, and third prizes will be granted (with provincial-level award certificates). However, any submission with algorithm performance below the baseline reference score provided by the organizers will be deemed invalid and not eligible for awards. Teams receiving first and second prizes in the semifinal will advance to the national final.

15.5 Final Round (National Competition)

(1) Online Evaluation: Teams advancing to the final round will be ranked on the final leaderboard. Based on the number of teams in the final, and in accordance with the award ratio set for the national competition, the organizers will determine the list of candidates for the national first prize and the winners of the second and third prizes (with national-level award certificates).

(2) Final Submission: Candidate teams for the national first prize must submit

technical documentation, algorithm code and model files, demonstration videos, and supplementary materials within the specified timeframe. No modifications or additional submissions will be accepted after the deadline.

(3) Final Review: A professional review committee will reproduce and verify the submitted results of the candidate teams for the national first prize. If necessary, participants may be required to provide further explanations during the review process.

(4) On-site Final Defense: Candidate teams for the national first prize must submit their finalized documentation, algorithm code and model files, demonstration videos, and supplementary materials by the specified deadline, and participate in the offline defense at the national final. The final ranking and winners of the national first prize will be determined based on both algorithm performance scores and offline defense scores. Teams failing to participate in the offline defense will be considered as having forfeited their prize. National first-prize winners will be awarded honorary certificates.

16. Additional Notes

16.1 Fairness

Any form of cheating is strictly prohibited, including but not limited to data leakage, overlap between model pre-training data and test data, or plagiarism of others' code. Once discovered, the participant's qualification will be immediately revoked, and relevant responsibilities will be pursued.

16.2 Intellectual Property

All submissions must be original and must not have won awards in other competitions or been publicly published. The organizers reserve the right to display and promote submitted works for competition-related activities; however, the intellectual property rights remain with the participants.

17. Contact Information

Competition Q&A Group (QQ): 614278600

Email: tushujieyue233@163.com

Official Registration Website: www.aicomp.cn