MASTER OF TECHNOLOGY PROJECT REPORT

RecomBook —— A Book Recommendation Syetem

TEAM MEMBERS

FENG YUYUAN

LI CHANGWEI

ZHAO YUCHEN

MASTER OF TECHNOLOGY

1 Business case

1.1Background information

With the development of computer information technology and Internet technology, it has crossed from the previous era of information shortage to the era of information surplus. In this context, it is more and more difficult for people to find interesting information from many information.

Imagine a user who wants to buy a book. Users only need to open an e-commerce bookstore and purchase directly according to the book name.

The premise of this method is that users need to be clear about their needs and know exactly which book they want to buy.

However, if users do not have a clear need, for example, users just want to find their favorite books, users can not tell which book it is and what it looks like. Currently, RecomBook will help users fulfill their needs

1.2 Business objective

RecomBook aims to provide customer a better book shopping experience by recommending them the most suitable book quickly and accurately.

RecomBook starts with users. Starting from users and using the characteristics of the information age, RecomBook record the behavior of each user. The behavior can be previous browsing records, purchase records and evaluation records. These behaviors are recorded and stored as digital portraits of this user.

RecomBook pays more attention to evaluation records because users' scoring information on books can better reflect their preferences.

For a simple example, suppose that user A once gave five-star praise to book A. This behavior was recorded in digital portrait A.

Among other user data, RecomBook found user B, and user B also once gave five-star praise to book A. RecomBook found that their digital portraits were very similar, so RecomBook regarded them as "similar individuals".

Next, RecomBook looks for differences between similar users A and B. It found that user B once gave five-star praise to book B. Therefore, RecomBook recommended book B to user A. I n this process, user A meets the characteristics of "no clear needs" mentioned earlier. User A does not need to provide anything specifically for RecomBook. RecomBook is the active party in the whole process.

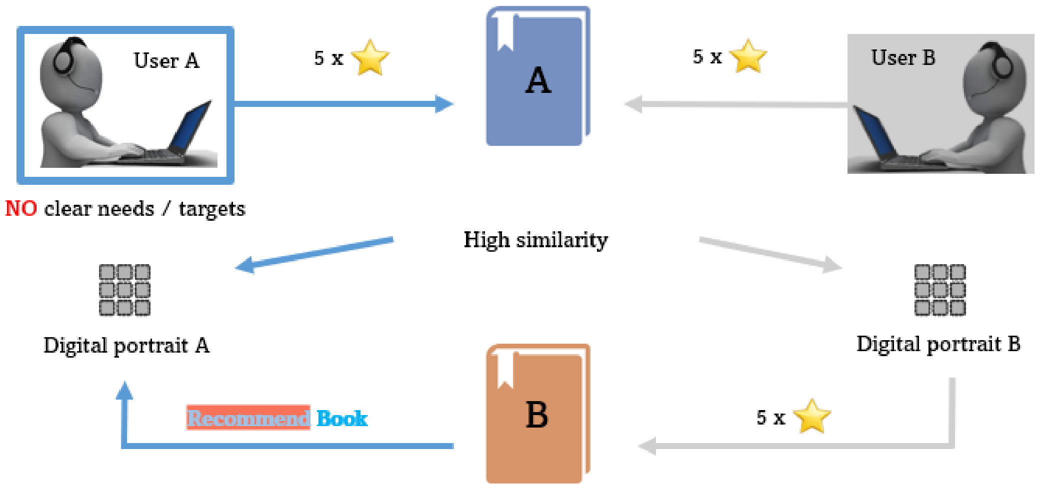


Figure 1 General principle of RecomBook's recommendation

1.3 Financial appraisal

This project is a low-cost Internet project, and the project funds only need server operation outside of the development. The project is non-profit in the initial stage, and the profit method in the middle and late stages is Website monetization.

2 Business Plan

2.1 product development stage

In the early stage of the product, it only has the recommendation function. Users need to score books on RecomBook website and make relevant recommendations according to the score

After accumulating a certain amount of user usage, the purchase or reading interface will be introduced in the middle and later stages of the product, so that users can complete the one-stop experience and facilitate the user's scoring method

2.2 Profit mode

The early stage of the project is non-profit, and will always provide free recommendation services to users. After accumulating a certain number of users, website monetization will be used to make profits, such as adding advertisements, unite authors or publishers pushing new books, etc.

3 Market Reseach

At present, there is no platform for book recommendation in the market, but many platforms or reading software integrate this function, such as bookfinder.com, Amazon Kindle and AI reader.

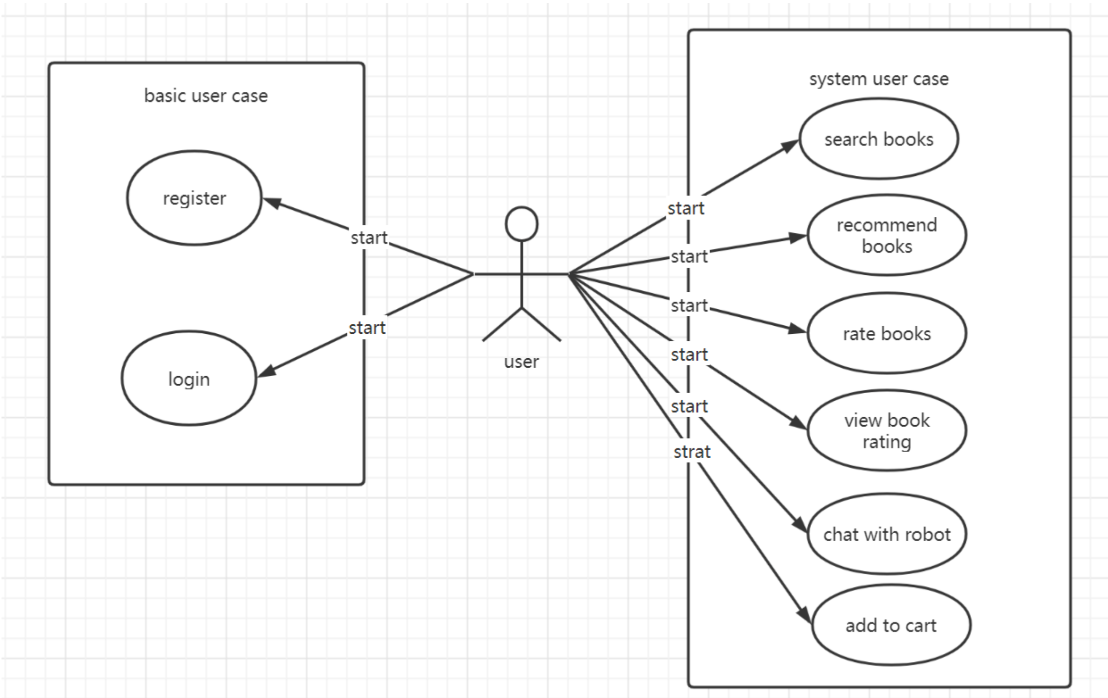
The focus of such websites and APP is more inclined to sales and reading, and the recommendation is only an incidental function. RecomBook completely subverts this business model. We will take recommendation as the leading, reading and shopping as the follow-up, and pay more attention to accurate recommendation. Create a recommendation platform tailored for book shortage readers.

In contrast, we have many disadvantages in the early stage of the project. For example, when there is only recommendation but no purchase or reading interface, the score data collection of new users will be more troublesome. However, with the progress of the project, our advantages will also appear. We have a more specific user group, dedicated to readers looking for the next book. Our recommendation system is more convenient and can be evaluated without buying or reading (the evaluation without buying or reading is only used as the current user's recommendation and is not included in the model calculation to prevent malicious scoring).

4. System design

4.1 User case design:

There are several user cases in our system include register, login, recommend books, rate books, search books, add to book cart, view book rating, chat with robot, and the Use case diagram is as follows:



The use case vocabulary is shown in Table 1:

User case name	Description	Character
Register	Users can register a new user and log in with that user	user
login	Users can log in to the system and use other functions	user
recommend books	Recommend n books for users according to their historical scores and other information	user
rate books	Users rate a specific book	user
search books	The user searches according to the book title, returns the book and can browse the details of the book	user
add to book cart	Add a book to the customer's shopping cart	user
view book rating	View the books that the user has scored and display them	user

chat with robot	Talk with the robot to get book information	user

4.2 Key user case description

recommend books

Use Case Name:	recommend books
Summary:	Recommend n books for users according to their historical scores and other information
Basic Flow:	1. The use case starts when user login. 2. The system calls the recommendation API base on the user id. 3. The user enters the homepage. 4. The system return the top N recommended books for the users. 5. The system shows books in the homepage.
Alternative Flows:	Step 3: The user call the chatbot to do the recommendation. Step 4: The recommendation result will be showed in the chatbox.
Extension Points:	None
Preconditions:	The user is login and he has some rating records.
Postconditions:	None.
Business Rules:	System will do recommendation when user login every time.

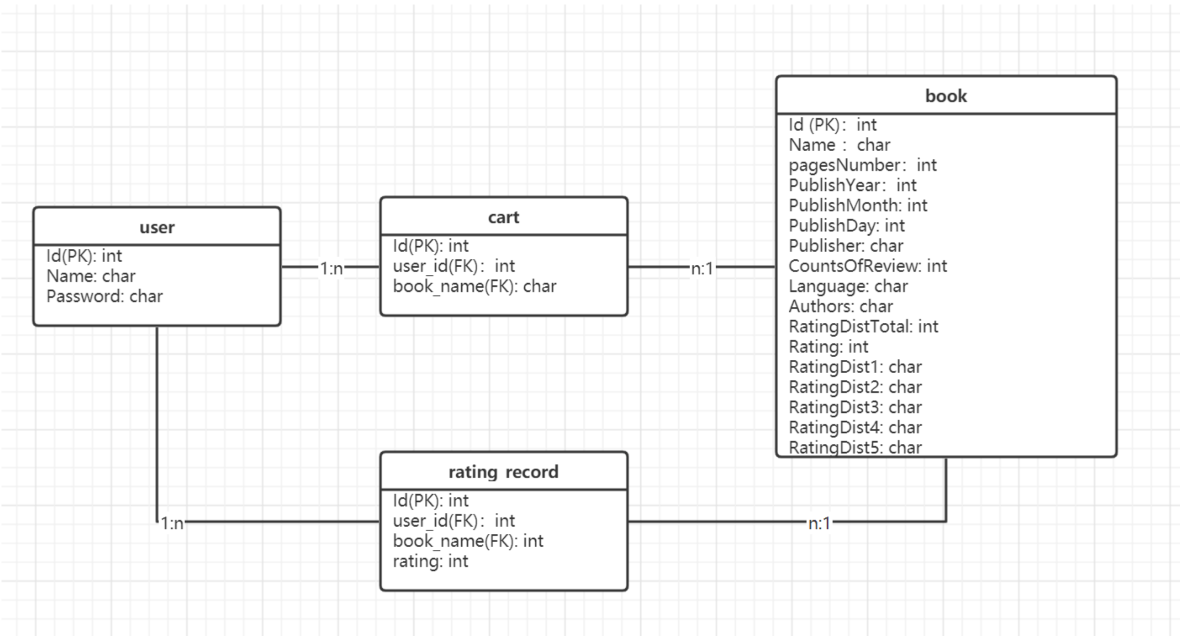
Chat with robot

Use Case Name:	**Chat with robot**
Summary:	Talk with the robot to get book information
Basic Flow:	1. The use case starts when user click the chat. 2. The user can input his question in the chat box. 3. The system will do the intent detection and slot detection. 4. The system will call the API to get the result of question. 5. The system shows result in the chat box.
Alternative Flows:	Step 4: If the question from user can not match any records the system will return none.
Extension Points:	None

Preconditions:	The user is login and he has some rating records.
Postconditions:	None.
Business Rules:	User can not ask some questions which are not relevant with books.

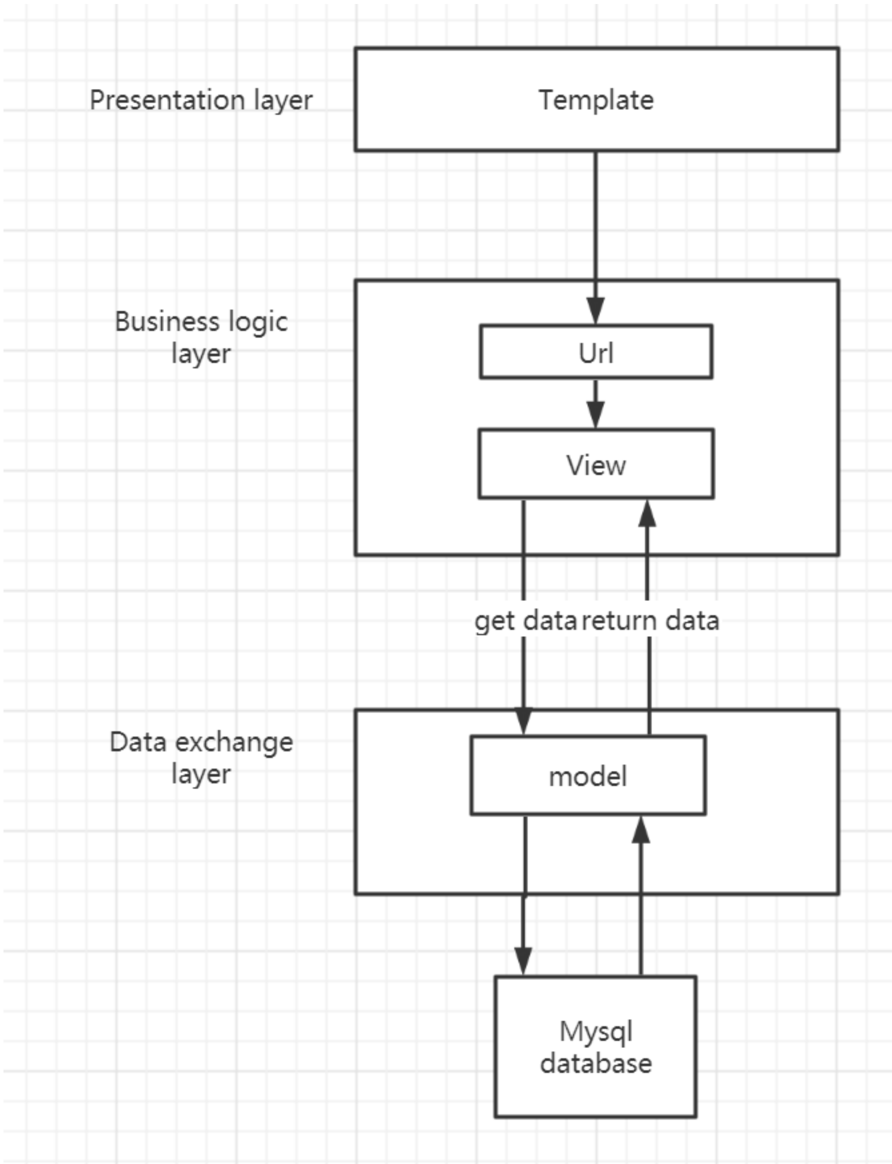
4.3 class diagram design:

There are 4 classes in our system, including use, cart, book, and rating_record. Combine these 4 classes, we design our recommendation system, The class diagram design is as follows:



4.4 software structure design

We use Django to develop the system. The Django framework of Python used in the system design adopts the classic MVT mode, as shown in Figure, where M represents the model, corresponding to the data access layer, which performs all access and verification operations related to the data table; T stands for template, corresponding to presentation layer, which is the page that users can operate between; V represents the view, that is, the business logic layer, which contains the logic related to accessing the model and accessing the appropriate template


Presentation layer
Template
Business logic
layer
Url
View
get data
return data
Data exchange
layer
model
Mysql
database

5 Model design

5.1 Matrix Factorization

5.1.1 Basic theory:

The principle of matrix factorization is relatively simple, which is decomposing a matrix D into the product of u and V. For a specific matrix D with scale m * n, estimate the matrices u and V with scale m * k and N * k respectively, so that the value of UVT is close to matrix D as much as possible. Generally speaking, the value of K should satisfy K ≤ min {m, n} K ≤ min {m, n}, so that the matrix decomposition is meaningful. If in the recommendation system, D represents the user behavior matrix for goods, then u and V represent the user and commodity vectors represented by embedding respectively.

Once factorization is done, we can predict the ratings for all users and all items. The Figure is as follows:

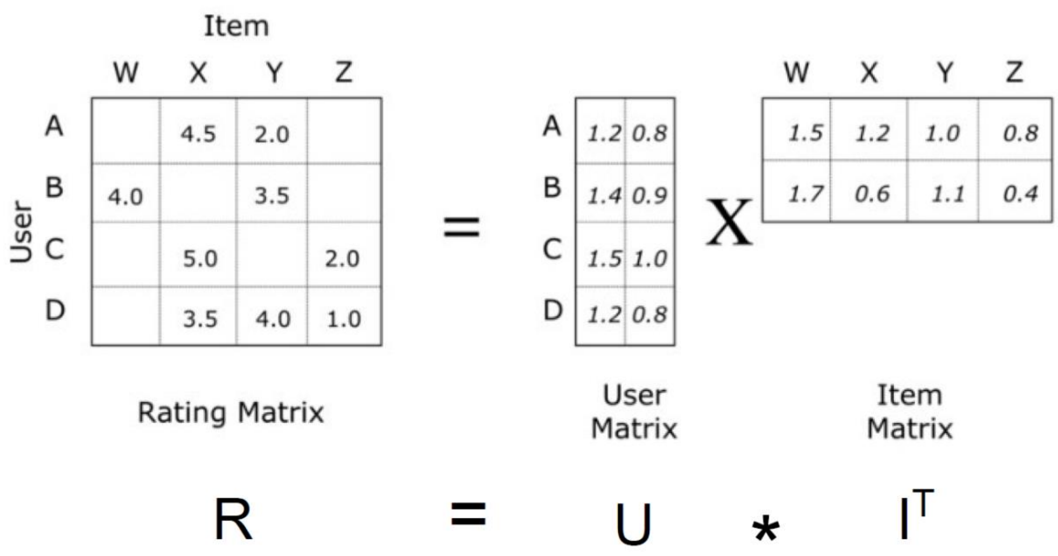


Figure 5-1 factorization principle

5.1.2 Implementation

We use the TensorFlow to implement this model and train the model use our dataset. The key point of this approach is using TensorFlow to build the embedding layer to implement the Matrix decomposition. We set K to be 10, and the structure of the model is as follows:

```
Model: "model_20"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_41 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
input_42 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
embedding_80 (Embedding)        (None, 1, 10)        10000       input_41[0][0]
__________________________________________________________________________________________________
embedding_81 (Embedding)        (None, 1, 10)        999460      input_42[0][0]
__________________________________________________________________________________________________
dot_20 (Dot)                    (None, 1, 1)         0           embedding_80[0][0]
                                                                 embedding_81[0][0]
__________________________________________________________________________________________________
embedding_82 (Embedding)        (None, 1, 1)         1000        input_41[0][0]
__________________________________________________________________________________________________
embedding_83 (Embedding)        (None, 1, 1)         99946       input_42[0][0]
__________________________________________________________________________________________________
add_20 (Add)                    (None, 1, 1)         0           dot_20[0][0]
                                                                 embedding_82[0][0]
                                                                 embedding_83[0][0]
__________________________________________________________________________________________________
flatten_20 (Flatten)            (None, 1)            0           add_20[0][0]
==================================================================================================
Total params: 1,110,406
Trainable params: 1,110,406
Non-trainable params: 0
```

Figure 5-2 structure of the Matrix Factorization model

The input data is the user_id and book_id, and the output data is the predicted score for this book, and use this model we can do the recommendation for the user.

Training

We split the rating_user.csv into train data and test data using split rate 0.8. Use mse to evaluate the performance of the model, and adjust the parameters to minimum the mse. Some optimal parameters are as follows:

Table 5-1: some optimal parameters

Parameters	Valuse
latent dimensionality K	10
regularization penalty	0.0005
optimizer	adam
batch_size	32
epochs	50

5.2 K-Means

K-means clustering algorithm is an iterative clustering analysis algorithm. The steps are to pre-divide the data into K groups, then randomly select K objects as the initial clustering centers, and then calculate each object and each distance between the seed cluster centers, and each object is assigned to the cluster center closest to it. The cluster centers and the objects assigned to them represent a cluster. Each time a sample is allocated, the cluster center of the cluster will be recalculated based on the existing objects in the cluster. This process will continue to repeat until a certain termination condition is met. The termination condition can be that no (or minimum number) of objects are reassigned to different clusters, no (or minimum number) of cluster centers change again, and the sum of squared errors is locally minimum.

In this project, we use K-means based on Silhouette Coefficient to cluster the books by the attributes, for example ratings, page number, author, e.g. After clustering, we give each cluster a tag to represent the books' type, which used in item-based collaborative filtering.

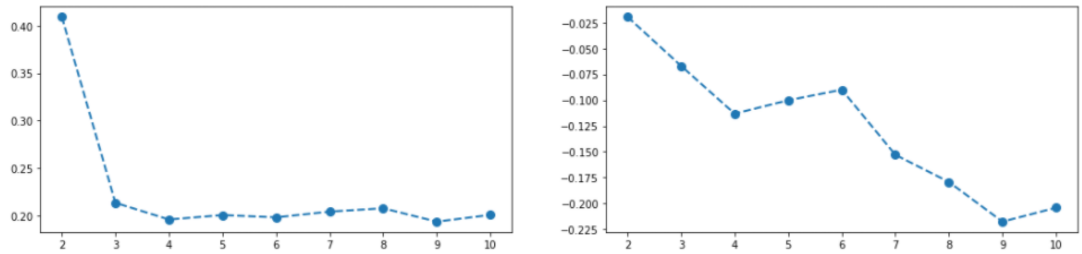


Figure 5-3 left silhouette score with cluster number. Right Minimum silhouette width with cluster number

5.3 RMB-CF

5.3.1 Overview

The objective of this module is to use the recommend model to recommend specific books for this user. The overall process is shown in the Figure 1*: First, use the book data and user ratings data in database to derive a user-rating matrix as the training input to train an RBM-CF model, and then use the personal information of the current user's comment and ratings to form a piece of information, which serves as the input of the trained RBM-CF model. The final output of the RBM-CF model is the top 20 best recommend book of the user. Meanwhile, the user information will also be merged into the user-rating matrix and re-train the model periodically.

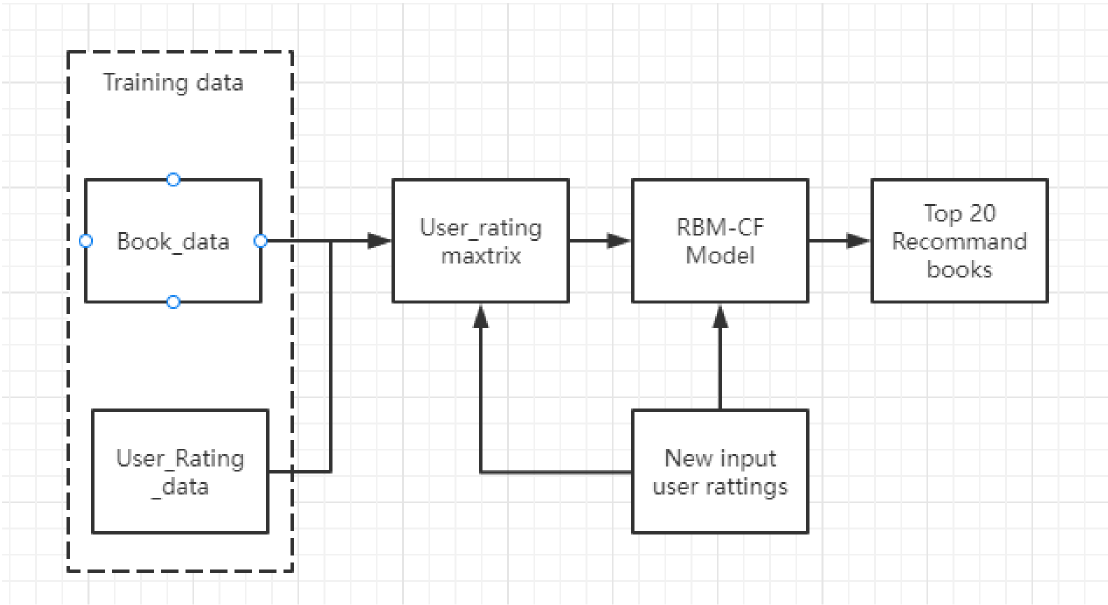


Figure 5-4. process flow of recommend model

5.3.2 User based Collaborative Filtering

User-Based Collaborative Filtering is a technique used to predict the items that a user might like on the basis of ratings given to that item by the other users who have similar taste with that of the target user. Many websites use collaborative filtering for building their recommendation system.

we can split the algorithm into two steps. The first is to find a set of users with similar interests to the target user. The other step is to find the users in this set that the user likes and the target user does not have. Recommend items that have been heard to target users.

To find user groups with similar interests, we must first define indicators to measure the similarity between users. This is also like the K-nearest neighbor algorithm mentioned between us, or the shadow of the clustering algorithm.

5.3.3 Restricted Boltzmann Machine

Restricted Boltzmann Machine (RBM) is a generative stochastic neural network proposed by Hinton and Sejnowski in 1986. The network consists of visible units (corresponding to Visible variables, data samples) and some hidden units (corresponding to hidden variables) are composed. Both visible variables and hidden variables are binary variables, which state takes {0,1}. The entire network is a bipartite graph. Only visible and hidden units will have edges, and there will be no edge connections between visible and hidden units, as shown in the figure 2*:

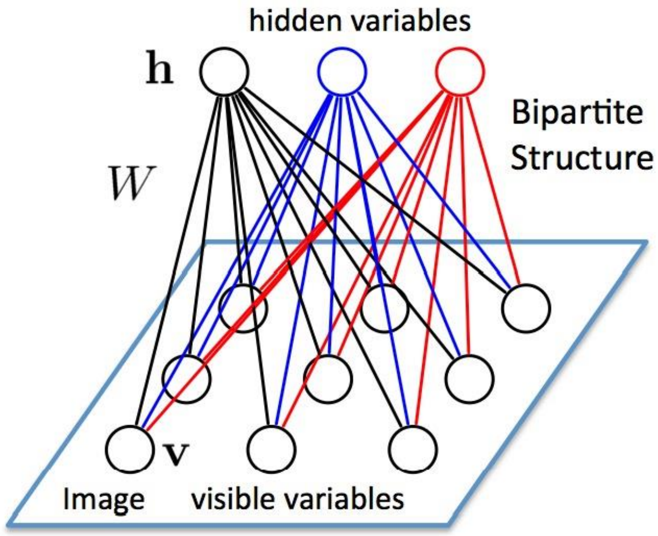


Figure 5-5 Architecture of Restricted Boltzmann Machine

Applying this model to collaborative filtering needs to solve the following two problems:

1. Given that the units in RBM are all binary variables, what if these binary variables are used to model the scores of integer values?

2. User ratings are very sparse, that is, users will only rate a few items (such as book and movies). How to deal with these missing ratings?

5.3.4 RBM-CF (Collaborative Filtering based on Restricted Boltzmann Machine)

R. R. Salakhutdinov et al. proposed a method for collaborative filtering using RBM: Assuming there are m books, m SoftMax units are used as visible units to construct RBM. For each user, different RBMs are used. These different RBMs are only different in visible units, because different users will rate different books. All the visible units of RBM share the same bias and the connection weight w with the hidden units. This method solves the aforementioned problems very well:

Use SoftMax to model the user's rating. SoftMax is a combined visible unit that contains k binary units. The binary unit i will be set to 1 if and only if the user scores the movie i.

If a user does not rate the movie j, then the SoftMax j unit does not exist in the user's RBM.

The model is shown in the figure 3*

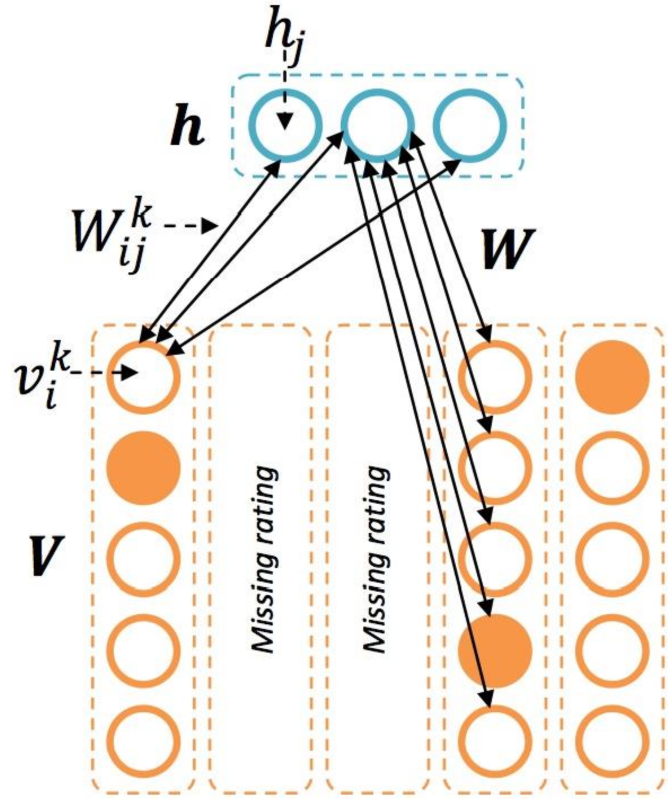


figure 5-6 the RBM-CF model

The conditional probability of the visible unit V and the hidden unit h is:

$$p(v_i^k = 1|h) = \frac{\exp(b_i^k + \sum_{j=1}^{F} h_j W_{ij}^k)}{\sum_{l=1}^{K} \exp(b_i^l + \sum_{j=1}^{F} h_j W_{ij}^l)}$$

$$p(h_j = 1|V) = \frac{1}{1 + \exp(-b_j - \sum_{i=1}^{m} \sum_{k=1}^{K} v_i^k W_{ij}^k)}$$

The learning process of model parameters is very similar to RBM's DC algorithm:

$$\Delta W_{ij}^k = \gamma_w(< v_i^k h_j >^+ - < v_i^k h_j >^T)$$

$$\Delta b_i^k = \gamma_v(< v_i^k >^+ - < v_i^k >^T)$$

$$\Delta b_j = \gamma_h(< h_j >^+ - < h_j >^T)$$

The training process is shown in the following figure:

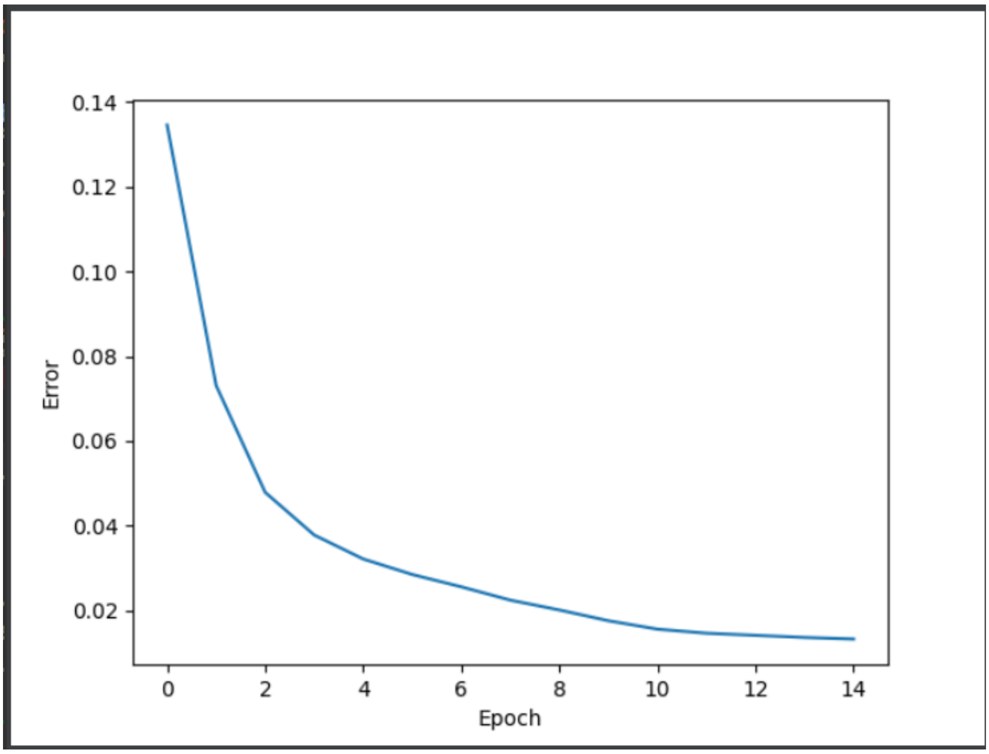


Figure 5-7 training loss with epoch

5.4 Input and output

The input and output of this module are shown in the Table 1. The input is a rating matrix derived from two training data book_data, and user_ratings_data. The books data including the BookID, BookTitle, ISBN, Author, Language, Publisher, CountsOfReview, Ratings. And user ratings data include UserID, BookTitle and Rating. The output of this module is top 20 best recommend book with its recommend score.

Table 5-2: description of input and output of recommend model

	VARIABLE NAME	DESCRIPTION
INPUT Book_data	BookID	The ID of books. Primary Key
	BookTitle	Title of books
	ISBN	International Standard Book Number
	Author	Author of books
	Language	The language of books
	Publisher	The books' publisher
	CountsOfReview	The Total counts of books' review
	Ratings	The total rating of books
INPUT	UserID	The Id of users. Primary key

USER_Ratings_data		
	BookTitle	Title of books. Foreign key
	Ratings	The rating user gave to each book
OUTPUT	BookTitle	Title of books
	Recommend Score	The scores computed by model, the higher the more worthy of recommendation

5.5 Model Hybrid

In this recommendation system, we mainly have two different type model (collaborative filtering and matrix factorization), to get an accurate recommend list, we try making a hybrid on these two models.

First, we use the RBM-CF to out put a recommendation list for with the heist 20 recommendation scores and store the scores respectively. Then, we process these books with matrix factorization, and get another score for each book. Next, we use a weight w to mix these two scores come from two model, and get a final score. Finally, output the list sorted in descending order of final scores.

The processing is illustrated by the following figure:

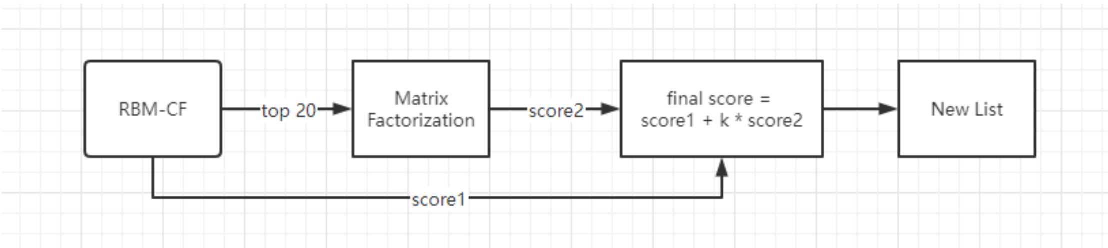


Figure 5-8 model hybrid processing

5.6 Chat bot

5.6.1 Illustration

The chat bot provides a way of dialogue and interaction between the user and the system. The user inputs a string of text to the system, and the system intelligently recognizes the pattern in the text and responds in different ways. Which mainly involves natural language processing technology.

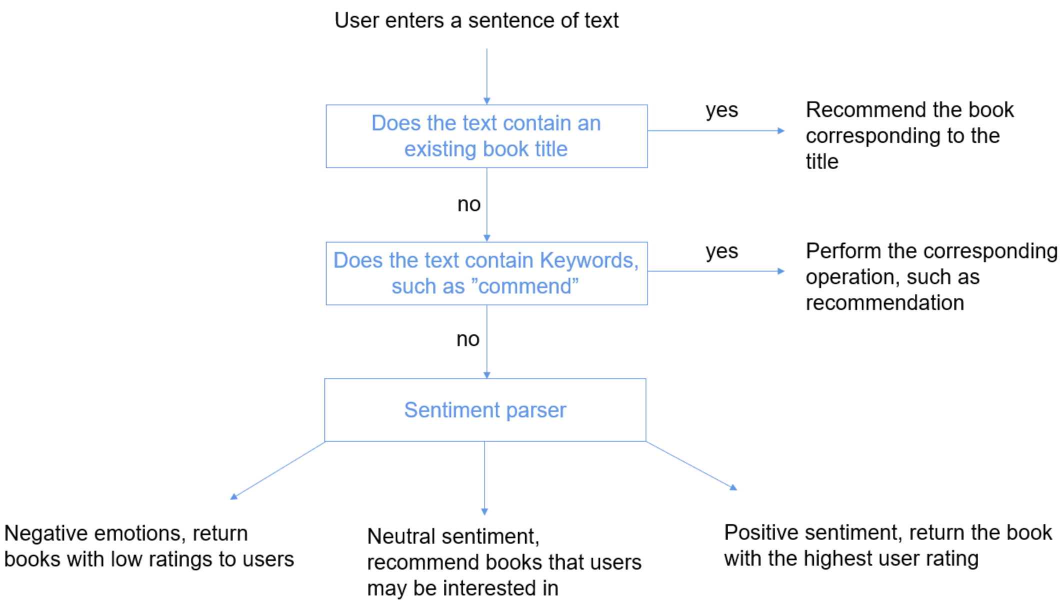


Figure 5-9 chat bot workflow

5.6.2 Sentiment parser workflow

A large number of comment texts are collected, and these comments are marked with emotional tendencies. The sentiment analyzer runs in the following process, and the model part is modeled as shown in the figure below.

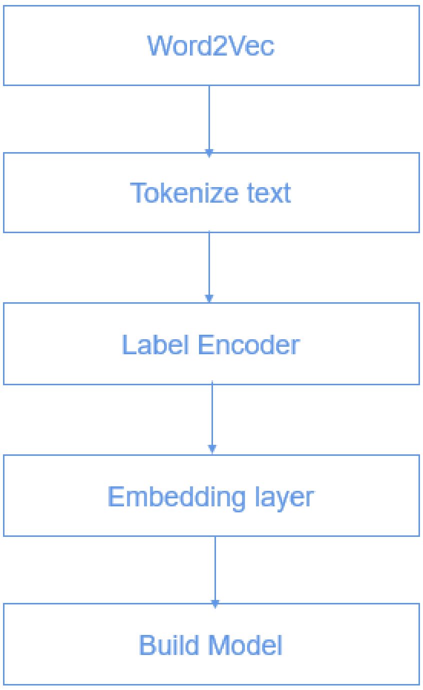


Figure 5-10 Sentiment parser workflow model

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 300, 300)          87125700
_________________________________________________________________
dropout (Dropout)            (None, 300, 300)          0
_________________________________________________________________
lstm (LSTM)                  (None, 100)               160400
_________________________________________________________________
dense (Dense)                (None, 1)                 101
=================================================================
Total params: 87,286,201
Trainable params: 160,501
Non-trainable params: 87,125,700
_________________________________________________________________
```

Figure 5-11 Sentiment parser workflow model details

6 System Development & Implementation in tools

the tools used in development and implementation in each part are listed in the following chart respectively.

Table 6-1 System Development & Implementation in tools

	Components	Tools
System	Frontend	HTML+JS+CSS
	Backend	Python, MySQL
	Web Framework	Django
Recommend Model	RBM-CF/Matrix Factorization	TensorFlow
	K-Means	scikit-learn
Chat Bot	LSTM/NN	Keras

7 future improvements

At the present stage, though we try using K-means to promotion the item type tag to get a better performance on the item-based CF, but the model is still weak in recommendation.

APPENDIX OF REPORT A

Project Proposal

GRADUATE CERTIFICATE: Intelligent Reasoning Systems (IRS)
PRACTICE MODULE: Project Proposal

Date of proposal: 18 September 2021
Project Title: RecomBook, A book recommendation system
Group ID (As Enrolled in LumiNUS Class Groups): 12 **Li Changwei A0231379B** **Feng Yuyuan A0231302A** **Zhao Yuchen A0231417N**
Sponsor/Client: *(Name, Address, Telephone No. and Contact Name)* Institute of Systems Science (ISS) at 25 Heng Mui Keng Terrace, Singapore NATIONAL UNIVERSITY OF SINGAPORE (NUS) Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email: zhan.gu@nus.edu.sg
Background/Aims/Objectives: The proposed intelligent eco-system will provide customer a better book shopping experience by recommending them the most suitable book quickly and accurately.
Project Overview: Our system mainly include following two parts: First is the core function: recommendation system. In the recommendation system, there are three method, user-based RBM-CF, content-based CF and matrix factorization, to derive the recommended list. The second part is a chat bot which build on LSTM and NN, to acquire the customers intent and return the recommend list. The system is based on python website which used the Django framework. The frontend is built in HTML+JS+CSS structure and backend base on python and MySQL.
Resource Requirements (please list Hardware, Software and any other resources) Hardware proposed for consideration: • GPU, etc. Software proposed for consideration: • MySQL 8.0.2.2

- Python 3.8
- TensorFlow 2.5.0
- Kears 2.5.0
- Pandas 1.3.3
- Numpy 1.21.2
- scikit-learn 1.0.1
- pymysql 1.0.2

Methods and Standards:

Procedures	Objective	Key Activities
Requirement Gathering and Analysis	The team should meet with ISS to scope the details of project and ensure the achievement of business objectives.	1. Gather & Analyze Requirements 2. Define internal and External Design 3. Prioritize & Consolidate Requirements 4. Establish Functional Baseline
Technical Construction	· To develop the source code in accordance to the design. · To perform unit testing to ensure the quality before the components are integrated as a whole project	1. Setup Development Environment 2. Understand the System Context, Design 3. Perform Coding 4. Conduct Unit Testing
Integration Testing and acceptance testing	To ensure interface compatibility and confirm that the integrated system hardware and system software meets requirements and is ready for acceptance testing.	1. Prepare System Test Specifications 2. Prepare for Test Execution 3. Conduct System Integration Testing 4. Evaluate Testing 5. Establish Product Baseline
Acceptance Testing	To obtain ISS user acceptance that the system meets the requirements.	1. Plan for Acceptance Testing 2. Conduct Training for Acceptance Testing 3. Prepare for Acceptance Test Execution 4. ISS Evaluate Testing 5. Obtain Customer Acceptance Sign-off
Delivery	To deploy the system into production (ISS standalone server) environment.	1. Software must be packed by following ISS's standard 2. Deployment guideline must be provided in ISS production (ISS standalone server) format

			3. Production (ISS standalone server) support and troubleshooting process must be defined.

Advisor Assigned:

Contact: Mr. GU ZHAN / Lecturer & Consultant
Telephone No.: 65-6516 8021
Email: zhan.gu@nus.edu.sg

APPENDIX OF REPORT B

Mapped System Functionalities against knowledge, techniques and skills of modular courses: MR, RS, CGS

Knowledge based reasoning techniques

We use knowledge-based reasoning to do the book classification and give each book a tag for the item-base collaborative filtering. We built a K-Means model to cluster books by their information.

Informed search optimization

We compute the recommend score by ensemble models, and perform a informed search optimization to recommend the highest scored books

Knowledge Discovery & data mining

We construct three recommendation engines to do the book recommendation. We use the user-based Restricted Boltzmann Machine Collaborative Filtering model, item-based Collaborative Filtering model and Matrix Factorization algorithm to do the data mining which is finding the similarities among each books in the dataset.

System designed with cognitive techniques

We built a chat bot which based on LSTM and NN to analyze the customers' intent and return the best matched result for them. The chat bot support our user-interface of human-mode communication.

APPENDIX OF REPORT C

Individual Reports

Individual Report: LI CHANGWEI (A0231379B)
Personal contribution

As a member of the team, I participated in many aspects from the initial direction assignment and data set screening to the final system development. In terms of data processing, after completing the system design, I integrated the existing data sets according to the class diagram design, and completed the code to import the data into MySQL database for later model training. In terms of recommendation algorithm, I assisted Zhao Yuchen to functionalize CF recommendation algorithm to facilitate back-end call, and use tensorflow to complete the implementation and training of matrix factorization recommendation algorithm, mixing the two output as the recommendation result. In terms of system development, I have completed the work of system design, learned the Django framework of Python as a development tool, and completed the development of most functions of the system, including login, registration, recommendation, search, rating and chat robot UI. For chat bot, I did some intent detection work in this part, we design a small rule base decision approach for this process.

Learning outcome

I have three main gains from this project.
The first is to enhance my understanding of various recommended algorithms and have a deeper understanding of their respective application scenarios, especially CF and matrix factorization algorithms. The complexity of practical problems is often difficult to corpus. Only by enriching their own knowledge system can we choose the best technical means in different problems.
Second, it has improved my data processing ability and training model ability, and also improved my programming skills to a certain extent. In particular, by learning from the achievements of others, I learned the ingenious application of using tensorflow to realize matrix factorization, and saved the process of writing mathematical methods for model fitting, which benefited me a lot. We should learn to make rational use of technology.
The third is to enrich my technical knowledge system. I learned a complete set of processes and methods of system development using Python's Django framework, which also provides me with more learning principles in my future work and study

Knowledge and Skill Application

Both the development experience of recommended algorithm and the development experience of system are of great help to my future work and study. I will continue to apply these technologies to complete new projects. The algorithm is also upgrading step by step. In terms of matrix factorization, I have seen many updated implementation methods, and it is also worth using this experience to further learn.

Individual Report: ZHAO YUCHEN (A0231417N)
Personal contribution

In the whole team project, I play a role in an algorithm builder, and the part I am responsible for is model selection and algorithm implementation. In this project, I completed the collaborative filtering algorithm based on the restricted Boltzmann machine at the core of the recommendation system. At the very beginning, my teammate Li Changwei and I selected the data set together. I analyze the data to select the most suitable and achievable recommendation algorithm among the many available recommendation algorithms. After selecting the data set and algorithm, I preprocessed the data set: deleted redundant data, mapped string-type comment data to numeric data, and generated a user ratings matrix. When building the RBM-CF model, I chose the TensorFlow v2 architecture and implemented the algorithm. In the project report, I am mainly responsible for writing the model part, including overview, user-based collaborative filtering, restricted Boltzmann machine, RBM-CF, input and output.

Learning outcome

I have three main gains from this project.
First of all, I deepened my understanding of recommender systems through this project. When selecting data sets and models, we clarified the data required by different models and the most applicable scenarios by comparing the differences, advantages and disadvantages between various models, At the same time, coding from scratch to implement a recommendation algorithm has greatly deepened my mastery of the algorithm principle from the data level.
Secondly, the process of data preprocessing gave me more experience in processing data of advanced data types like panda DataFrame. In this project, the process of searching for data and learning while preprocessing the data set in this project allowed me to learn many methods and techniques that I didn't know before.
Finally, from a macro perspective, I also deeply realized the gap between a project's idea and realization. Machine learning, deep learning, various algorithms require data support. We have many fantastic ideas, many of which are amazing. But in the actual trial, it was found that there was no data source support, which made it difficult to achieve. A novel idea means that there is no support from the data collected by the predecessors, while the sufficient data means that the current application has been realized by the predecessors' countless times. Innovation will only be reached by breakthrough in technology.

Knowledge and Skill Application

The knowledge and skill I learned in this project will definitely be very useful in future jobs. In fact, I have already got an offer from the AI group in GF securities CO.LTD, and the one of the core businesses for new staff in this group was building recommendation system to recommend suitable securities information to customer. I believe this project experience will bring me great help getting to work quickly.

Individual Report: Feng Yuyuan (A0231302A)

Personal contribution

At the system development level, I completed the wishlist.html on the front-end page and all the logic on the wishlist page. I implemented the wish list function and modified and improved the small functions of other pages. In terms of development, team member Li Changwei is the main force, and I assist him in completing the system.

In addition to development contributions, I am mainly responsible for the chat bot and content-based recommendations. The algorithm logic and model building, training, and testing of chat bot are done by me. Other team members apply the algorithm to the system interface and debug it. The chat bot performs natural language processing on the text entered by the user in the search box. In this part, the algorithm analyzes the user's input semantically and responds intelligently.

In addition, my content also includes the clustering of books, that is, the formation of book communities. People will always like books with certain characteristics. When the system recognizes the characteristics of the user's preference, in this book community, book-based recommendations can be realized.

Learning outcome

In the process of the project, I mainly used Python implementation, including K-means, content-based collaborative filtering and other algorithms. At the same time, I used the Keras architecture to build LSTM and NN to realize the semantic analysis and intelligent response of the chat robot to the user's input. This process has greatly improved my mastery of the algorithms involved, and also deepened my proficiency in Keras architecture. In addition, in terms of system and web page, I used Django to complete the wish list, which is an architecture I have not used before. This project experience makes me quickly learn Django's architecture and add new knowledge.

Knowledge and Skill Applications

In terms of development, the front-end page is written with the learning and use of html, css and JavaScript. The back-end database uses MySQL and performs a certain amount of database operation statements. The overall framework uses python Django. In terms of artificial intelligence algorithms, I learned and used clustering and neural networks. In the selection of the recommendation algorithm, our group learned and adopted a collaborative filtering method.