

MASTER OF TECHNOLOGY PROJECT REPORT

Movie Recommend system

TEAM MEMBERS

WANG SIXIANG -

LIU YANJIE -

CAO ZIHAO -

CHEN TIAN YU -

MASTER OF TECHNOLOGY

1.0 EXECUTIVE SUMMARY

With the rapid development of the film industry, all kinds of films on the platform have expanded and increased exponentially. In such a huge movie database, it is very difficult for moviegoers to decide which movie to watch. Recommendation systems are widely used in the Internet industry, with personalized recommendations for takeaway apps, shopping apps, etc. In order to save the time for moviegoers to investigate and analyze the huge number of movies, we built this movie recommendation system. This system will recommend a list of movies suitable for watching based on the preferences of the moviegoers.

Our group consists of four students who love movies and we usually watch movies in our leisure time. However, the current personalized recommendation function of video apps is not very complete. Most of the recommendations are based on the general data of the movie market, and there is no personalized recommendation. So we want to complete a personalized movie recommendation system, which can recommend a movie list according to each user's own preferences

In order to implement this system, we first built a database containing the detailed content of each movie. The data comes from open source movie data on the Internet. Then we use Python to build the front-end UI and back-end data processing system, and connect them. Finally, it can be visited for experience through an independent URL.

We believe that this system has great commercial value. Movies are the most popular leisure activity for people nowadays. Nowadays, there are many movies on the market, and people don't know how to choose a movie that suits them. Through such a recommendation system, it saves the time for viewers to investigate and also allows them to see movies that suit their tastes.

2.0 PROBLEM DESCRIPTION

With the development of the times, the development speed of the film industry is also increasing rapidly. Movies have become indispensable for mass leisure and entertainment. More and more people choose to watch a movie during their break time as a pastime, which is one of the most popular art forms today. Because of the rapid development of the film industry, there are countless movie types on the market for the public to choose from. But not every movie meets personal aesthetic standards, so we need a movie recommendation system to help us make quick decisions.

How to get users' favorite movies? Is it based on the movie genre? Or is it inferred based on the user's favorite movies? Do you need data from other users to support inference?

Members of our group randomly selected video recommendation software commonly used in the market for experience. The current movie recommendation is mainly based on big data, not personalized customization. The recommended movies are all movies with high ratings or with many favorites, but not everyone's movie tastes are the same. Although some movies have high ratings, they are not my favorite types, so the purpose of this system is to personalize the recommendations and make recommendations based on personal preferences.

2.1 PROJECT OBJECTIVE

The purpose of our project is to find a suitable algorithm that can recommend the most suitable viewing list for each user based on their characteristics. This algorithm takes the user's score for each movie and the movies added to the favorite movie list as the element of analysis. All in all, this project will focus more on the customer's own preferences

In order to implement this system, we not only need to obtain various kinds of movie resources from various websites, but also need to obtain other users' information, and recommend movies based on the similarity between users. Therefore, we need to set the system to the type of registration and login so that the information of each user who uses the system can be stored in the database.

3.0 KNOWLEDGE MODELING

Knowledge modeling includes the following aspects:

- (i) Requirement Analysis
- (ii) Design
- (iii) Implementation
- (iv) Testing
- (v) Evolution

Each step is very critical, but mainly focused on design and implementation

31 REUIREMENT ANALYSIS (BUSSINESS VALUE)

Due to the current epidemic, more and more people will choose to enjoy their free time at home. Movie watching has become the first choice for entertainment. According to statistics, the use of video apps has increased significantly during the epidemic. The movie recommendation system will bring better choices to moviegoers. The business value of this system is that it can be combined with existing video apps to bring users a better experience.

32 DESIGN(knowledge representation / knowledge acquisition)

For knowledge modeling, the ‘design’ part includes knowledge representation and knowledge acquisition. The representation of knowledge is a description of knowledge, or a set of conventions on knowledge, a data structure acceptable to computers for describing knowledge.

In order to express ‘how to construct a movie recommendation system’ more clearly, the following inference diagram is constructed. The top-level inference shows how the system selects a movie that the user may want to watch. Mainly determined by the elements in sub-goals.

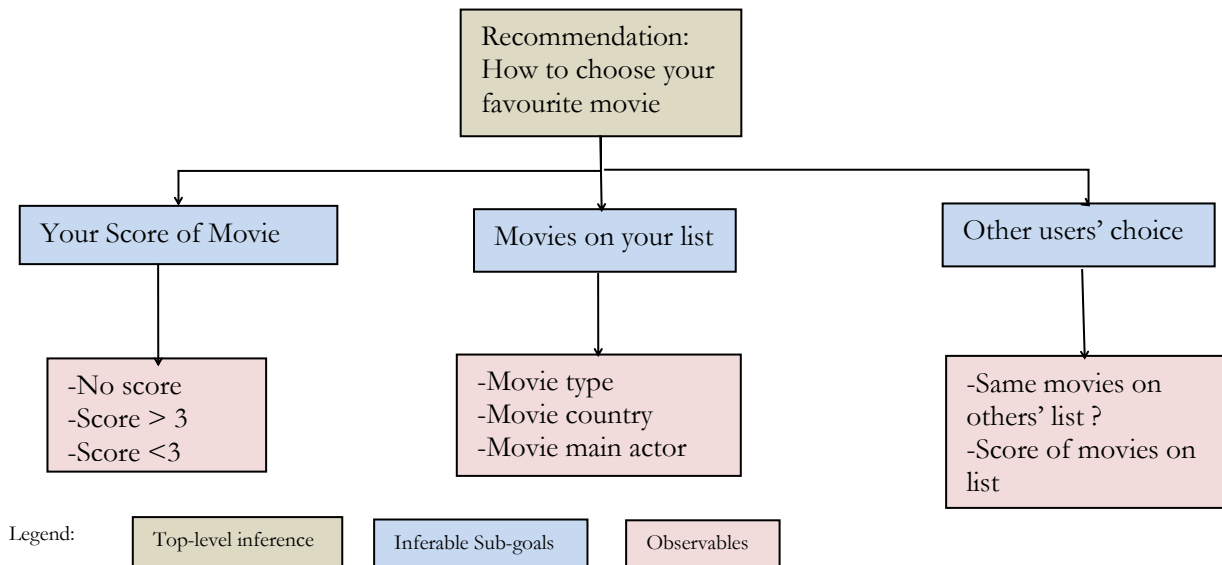


Figure 1. Inference diagram of movie recommend system

The movie recommendation system we built is mainly divided into two parts: i) user information; ii) movie information. The following table shows formal logic rules of the system.

IF No user logged in THEN Jump to login page
--

IF Username and password match THEN Jump to Home page
IF Username and password not match THEN Jump to register page
IF Enter the correct movie name in the search box THEN Jump to the detail page of the movie
IF Give a high mark of a movie THEN Similar movies according to this movie will be recommended
IF Put a movie into movie list THEN Similar movies according to this movie will be recommended

Table 1. Rules (Business Knowledge) of system

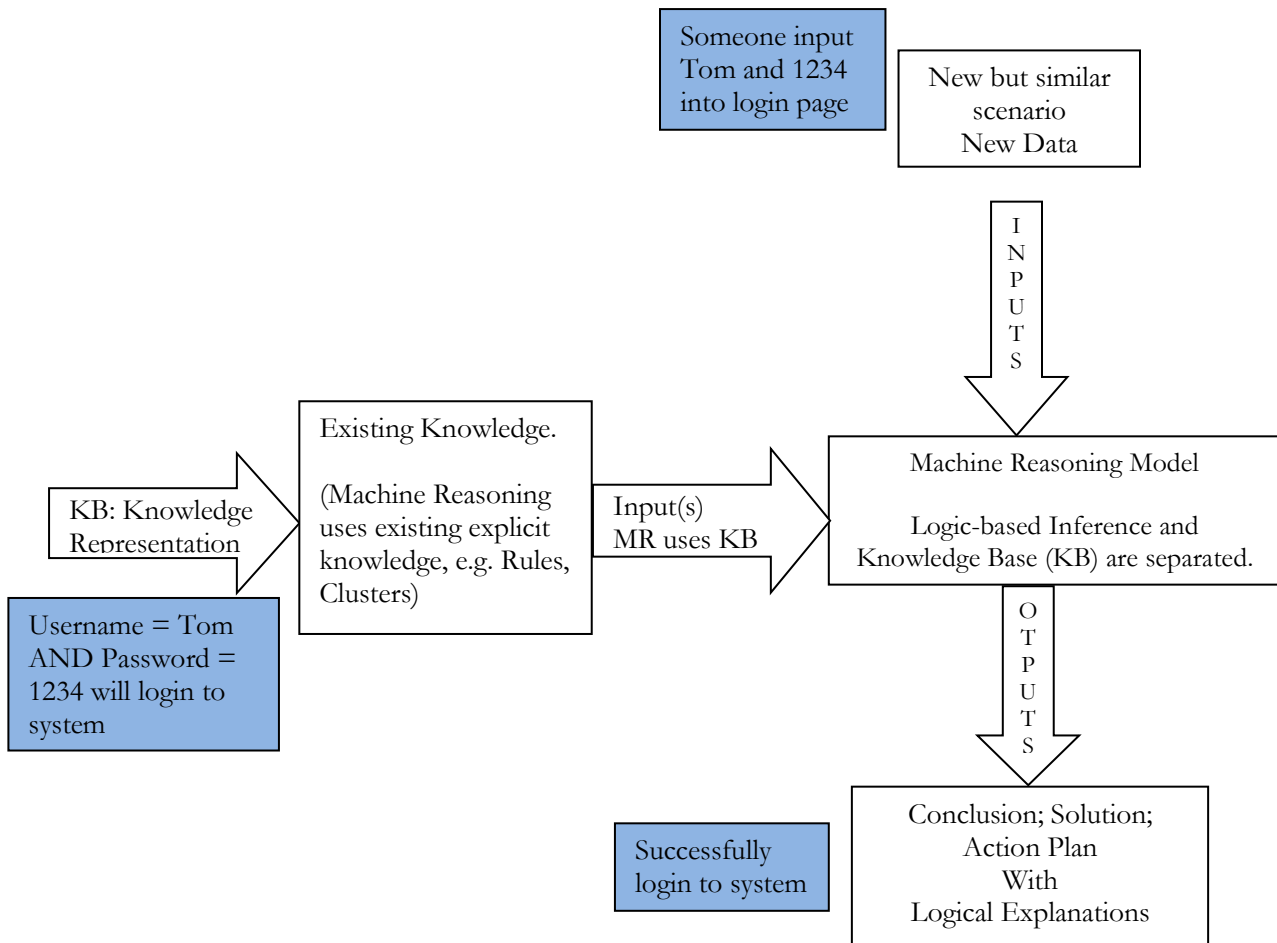


Figure 2. Rule inference

Knowledge Acquisition is the transfer and transformation of problem solving knowledge into a form that can be used

to build intelligent systems. There are two types of knowledge can be used to build KBS/RBS: i) Documented sources (Collection from printed sources/ Machine learning) ii) Undocumented sources (Tacit knowledge that can only be captured by elicitation from human Experts)

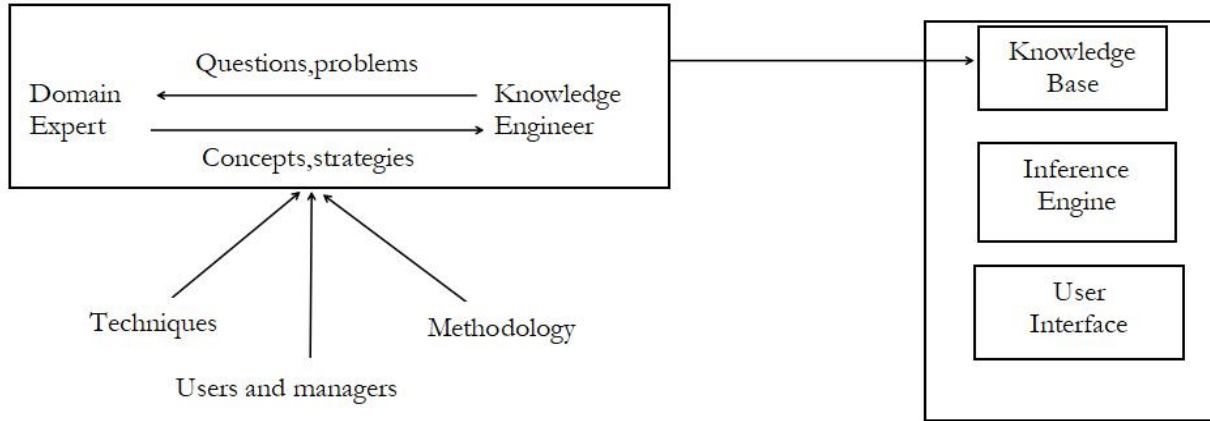
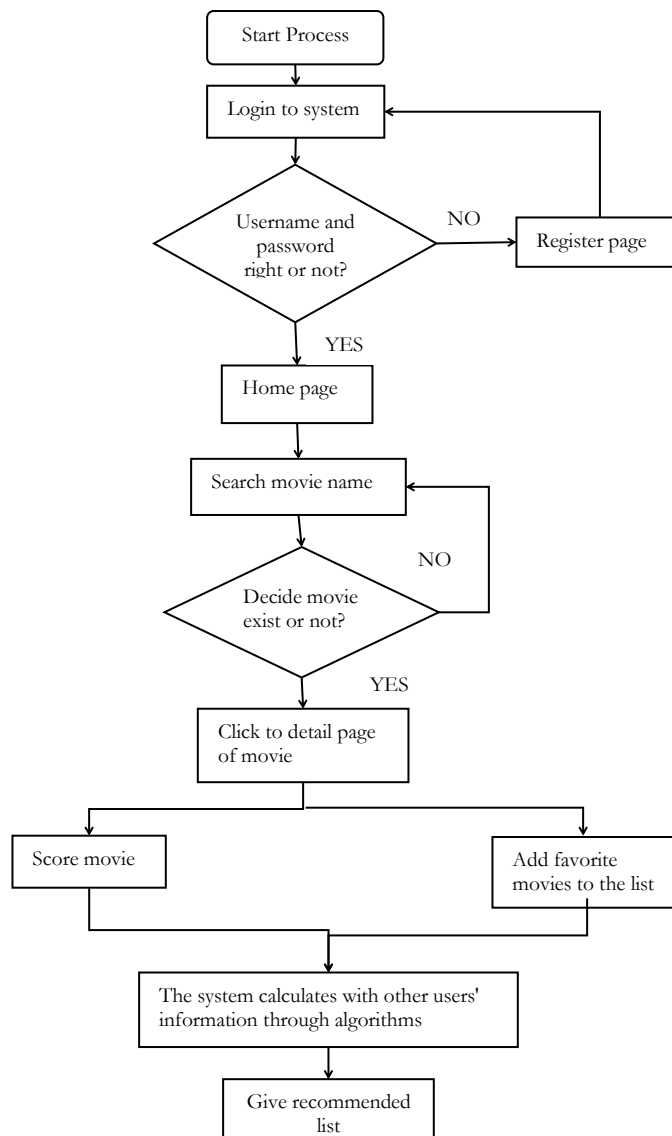


Figure 3. Knowledge Acquisition

33 IMPLEMENTATION(knowledge model)

The key to develop Flowchart to model/capture business process workflow, is to define the sequence of activities, and to identify those points where the flow can go two ways, depending on the circumstances.

The following flowchart shows the running process of the entire system.



34 TESTING(knowledge model)

For this system, we use black-box testing and white-box testing to determine its robustness.

Black box testing, also known as functional testing, data-driven testing, or specification-based testing, is a test from the user's point of view. Testers generally treat the program under test as a black box. We get opinions by letting other students who are not members of the group operate the system.

Test Task	Result
Enter correct movie name on search bar	Display the movie detail page
Enter incorrect movie name on search bar	Result will be empty
Login with correct information	Successfully jump to home page
Login with incorrect information	Alert: Invalid login
Register with correct information form	Successfully jump to home page
Register with incorrect information form(eg. e-mail address)	Alert: incorrect email address
Submit score of movie	The score will be kept in detailed page
Submit without score of movie	Alert: invalid input
Add movie to list	The movie is added in MyList
Remove movie from list	The movie is removed from MyList
Check recommended movie list	The similar movies will be displayed

Table 2: Black-box testing of system

White-box testing is also called structural testing or logic-driven testing. It is a test of how the unit under test works. It designs test cases according to the control structure of the program, and is mainly used for software or program verification. We reviewed the overall code by different team members to confirm the front-end connection. The page jump and algorithm are correct.

4.0 SOLUTION OUTLINE

In the third part, the whole process of knowledge modeling is introduced, and the corresponding functions of the system are described. The main function of the system is to recommend movies that customers are most likely to enjoy, and a collaborative filtering algorithm is used here. This part will introduce the structure of the entire system and the algorithm used to complete the recommended function.

4.1 SYSTEM ARCHITECTURE

The back end of the entire system is built using Python language, and the front end uses html files and css files. Build a database through sqlite. Realize page jump and function realization through POST request. Users can log in to the website and use the system directly through the URL, and the interaction is very friendly.

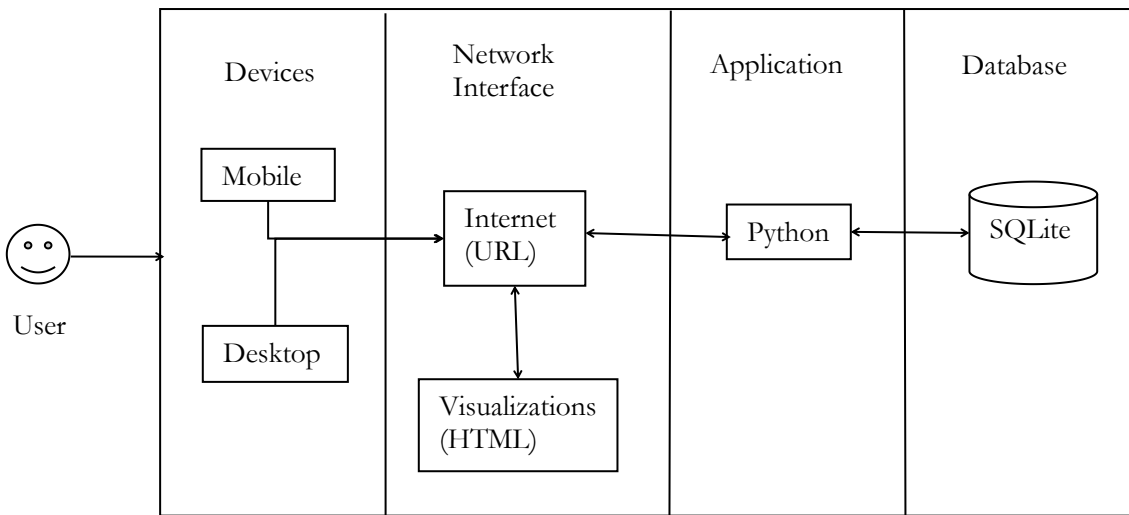


Figure 5: Movie Recommend System Architecture

4.2 RECOMMENDER ALGORITHM

Collaborative filtering means that users can work together to continuously interact with the website, so that their recommendation list can continuously filter out items that they are not interested in, so as to meet their needs more and more. Collaborative filtering recommendation algorithms are divided into two categories, namely user-based collaborative filtering algorithms and item-based collaborative filtering algorithms. When we want to find a new movie to watch, we often ask our friends for their opinions, especially those with similar tastes. So in this system we use a user-based collaborative filtering algorithm.

The user-based collaborative filtering algorithm discovers the user's preferences for products or content (such as product purchase, collection, content review or sharing) through the user's historical behavior data, and measures and scores these preferences. Calculate the relationship between users based on the attitudes and preferences of different users towards the same product or content. Recommend products among users who have the same preferences.

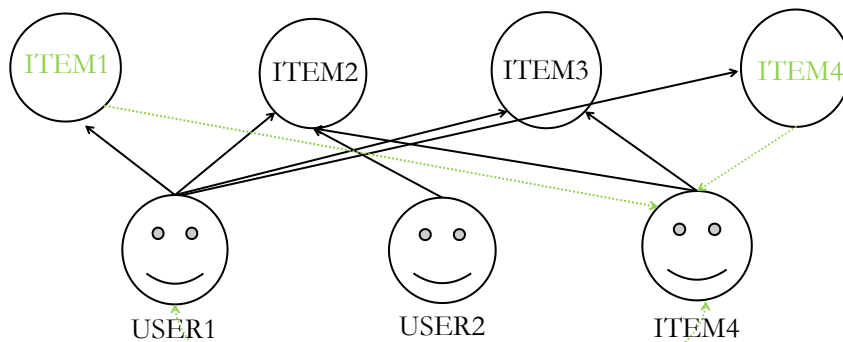


Figure 6: User-based collaborative filtering

When a user A needs to personally recommend a movie, he can first find other users who have similar electronic music hobbies, and then recommend those movies that the user likes but that the user A has not watched. There are two steps to complete this algorithm: a) Find a collection of users with similar interests to the target user. b) Find movies that users in this collection like and that the target user has not heard of and recommend to the target user. The key of step (a) is to calculate the similarity of interest of two users. Here, the collaborative filtering algorithm mainly uses the similarity of behaviors to calculate the similarity of interests. Given user u and user v , let $N(u)$ denote the collection of items for which user u has had positive feedback, and let $N(v)$ be the collection of items for which user v has had positive feedback. Then, we can simply calculate the interest similarity between u and v through the following Jaccard formula or through the cosine formula:

$$w_{uv} = \frac{\sum_{i \in N(u) \cap N(v)} \frac{1}{\log 1 + |N(i)|}}{\sqrt{|N(u)| |N(v)|}}$$

Jaccard formula

$$w_{uv} = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)| |N(v)|}}$$

Cosine formula

After obtaining the interest similarity between users, the UserCF algorithm will recommend the K user favorite items with the most similar interests to the user.

4.3 SYSTEM'S FEATURES

The recommendation system has some notable features that can make it attract more users to use.

SYSTEM'S INTELLIGENCE & ROBUSTNESS

This personalized movie recommendation system is very intelligent, it does not require users to do any investigations themselves. The system will automatically recommend a list of movies most likely to be liked by the user based on the movies added to the favorite list and the rated movies through a collaborative filtering algorithm. And according to the type of movie classification, users can browse a type of movie faster.

The system is also very robust. There are many fault tolerance settings in the system to improve the robustness of the system. When the user name and password do not match when logging in, the system will prompt the user to enter the correct user name and password or register a new account. There will be a prompt when the email format is incorrect during registration. When searching, if the movie name is not entered completely, the movie will be searched intelligently.

EASE OF ACCESS

This recommendation system is web-based, so anyone using any networked device, just enter the correct URL. What's more, the design of the front page is also convenient for users to optimize user experience. When scoring a movie, you can directly select the number of stars without typing it yourself. Adding to the favorite list is also realized by checkbox.

SCALABILITY

Since many new movies appear in the movie industry every year, it is necessary to update movie information at any

time. This movie recommendation system stores data in a database instead of a static page, so it is easy to update information and has good scalability. Not only that, as the number of users increases, the stored user information should also increase accordingly.

4.4 LIMITATIONS

User-based collaborative algorithms still have certain limitations. Next, we need to develop more suitable algorithms to apply to this movie recommendation system.

Performance	It is suitable for occasions with few users. If there are many users, it is very expensive to calculate the user similarity matrix
Area	Areas where the timeliness is strong and the user's personalized interest is not obvious
Real-time	The user has new behavior, which does not necessarily cause immediate changes in the recommendation results
Start	After a new user has behavior on a few items, he cannot be personalized recommendation immediately, because the user similarity table is calculated offline at regular intervals.

Table 3. Limitations of Algorithm

5.0 CONCLUSION & REFERENCES

Through this project, the members of our group have a better understanding of the knowledge learned in class. In the process of practice, I learned how to build a knowledge system, and I also encountered many difficulties in the process. But we finally overcame the difficulty, built an intelligent recommendation system and applied collaborative filtering algorithm.

Before that, we were not good at using actual code. In the process of completing this project, we tried to use Python, HTML and SQL to build a complete web-based system on the front and back ends.

In general, the movie recommendation system can basically realize our initial vision of this system. Users can quickly experience the system through URL.

5.1 IMPROVEMENTS:

If we have more time to spend on this project, we will make the following improvements to make the system more complete:

1) Expand the database

The database of the current system is not large enough, and more data needs to be added to make the data more universal. Not only are popular movies, but many niche movies should also be included in the database, providing users with more different choices. Not only that, the current system only provides the basic information of the movie. If you can add a viewing link or the copyrighted video itself, it will be more convenient for users to use and attract more people.

2) Try multiple algorithms

The algorithm of the current system is relatively simple, and only uses a user-based collaborative filtering algorithm. In order to improve accuracy, multiple algorithms should be used and compared to choose a more suitable algorithm, and even multiple algorithms can be combined to achieve the best recommendation effect.

6.0 REFERENCE

Introduction to Knowledge Modeling: http://www.makhfi.com/KCM_intro.htm

Build a Recommendation Engine With Collaborative Filtering: <https://realpython.com/build-recommendation-engine-collaborative-filtering/>

Build Recommendation Systems: <https://developers.google.com/machine-learning/recommendation/collaborative/basics>

7.0 APPENDICES

APPENDIX A: PROJECT PROPOSAL

PROJECT PROPOSAL

1. TEAM MEMBERS:

LIU YANJIE, WANG SIXIANG, CHEN TIANYU, CAO ZIHAO

2. PROJECT TITLE:

Movie Recommend System

3. PROJECT BACKGROUND:

With the development of the times, the development speed of the film industry is also increasing rapidly. Movies have become indispensable for mass leisure and entertainment. More and more people choose to watch a movie during their break time as a pastime, which is one of the most popular art forms today. Because of the rapid development of the film industry, there are countless movie types on the market for the public to choose from. But not every movie meets personal aesthetic standards, so we need a movie recommendation system to help us make quick decisions.

4. PROJECT OBJECTIVES:

The purpose of our project is to find a suitable algorithm that can recommend the most suitable viewing list for each user based on their characteristics. This algorithm takes the user's score for each movie and the movies added to the favorite movie list as the element of analysis. All in all, this project will focus more on the customer's own preferences

In order to implement this system, we not only need to obtain various kinds of movie resources from various websites, but also need to obtain other users' information, and recommend movies based on the similarity between users. Therefore, we need to set the system to the type of registration and login so that the information of each user who uses the system can be stored in the database.

5. SUMMARY OF METHODS:

The knowledge model is actively used to analyze and adapt to the environment, the needs, habits and interests of users and customers. This can provide higher-quality self-service and effective personalization in various applications ranging from targeted advertising and product recommendations to blogs and news sites.

Collaborative filtering means that users can work together to continuously interact with the website, so that their recommendation list can continuously filter out items that they are not interested in, so as to meet their needs more and more.

APPENDIX B: INSTALLATION & USER GUIDE

1. INSTALLATION

Use pip instructions to install the following packages:

```
asgiref==3.2.7
boto3==1.14.1
botocore==1.17.1
Django==3.0.6
django-storages==1.9.1
docutils==0.15.2
gunicorn==20.0.4
idna==2.9
jmespath==0.10.0
numpy==1.18.5
pandas==1.0.4
Pillow==7.1.2
python-dateutil==2.8.1
pytz==2020.1
recommend==0.2.2
requests==2.23.0
s3transfer==0.3.3
scipy==1.4.1
six==1.15.0
sqlparse==0.3.1
urllib3==1.25.9
whitenoise==5.1.0
```

Use Pycharm to build a system

Use SQLite Studio to manage database

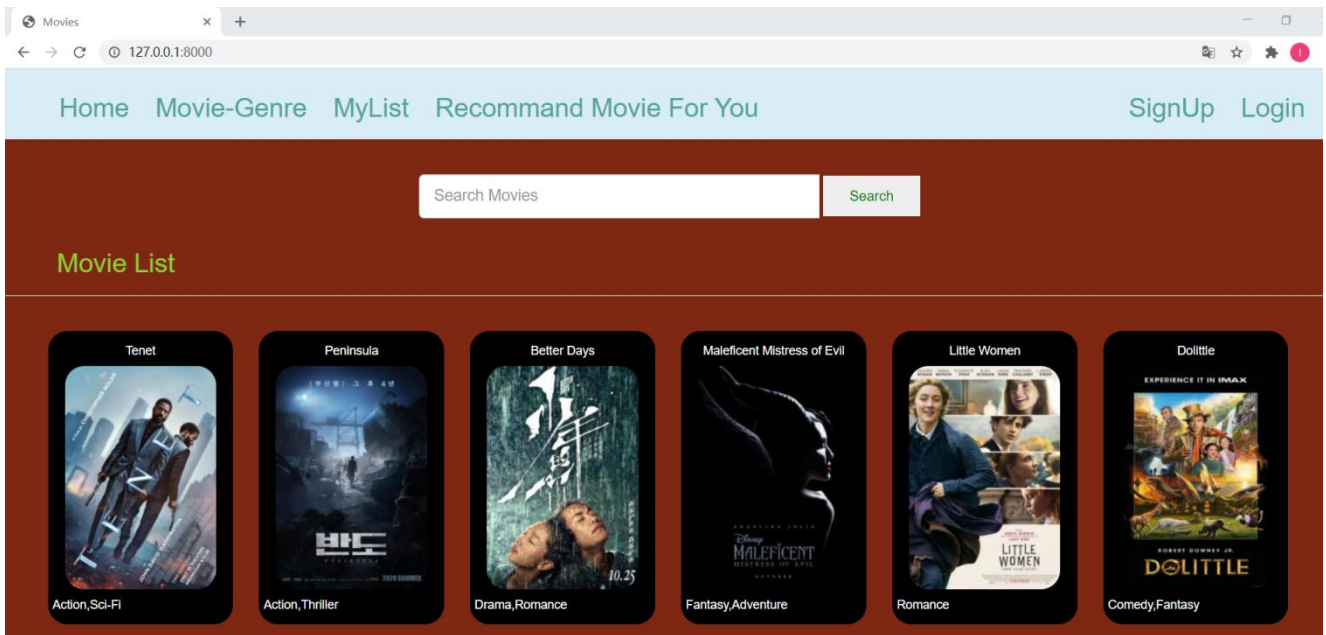
2. USER GUIDE

Run command: python manage.py runserver

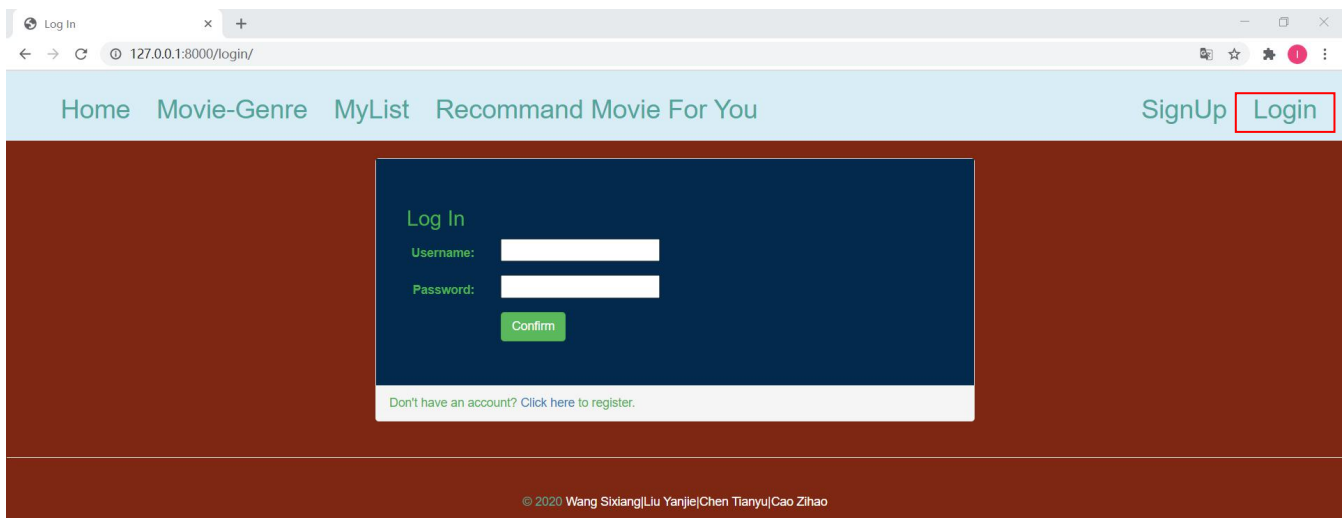
```
(base) C:\Users\S6188\Desktop\NUS-IS\movie_recommender-master>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
October 24, 2020 - 12:22:32
Django version 3.0.6, using settings 'movie_recommender.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[24/Oct/2020 12:22:38] "GET / HTTP/1.1" 200 58213
[24/Oct/2020 12:22:38] "GET /static/recommend/css/bootstrap.min.css HTTP/1.1" 200 121314
[24/Oct/2020 12:22:38] "GET /static/recommend/css/base.css HTTP/1.1" 200 928
[24/Oct/2020 12:22:38] "GET /static/recommend/js/bootstrap.min.js HTTP/1.1" 200 37045
[24/Oct/2020 12:22:38] "GET /media/Tenet.jpg%0A HTTP/1.1" 200 10286
[24/Oct/2020 12:22:38] "GET /media/Peninsula.jpg%0A HTTP/1.1" 200 6491
[24/Oct/2020 12:22:38] "GET /static/recommend/js/jquery.min.js HTTP/1.1" 200 97163
[24/Oct/2020 12:22:38] "GET /media/Better%20Days.jpg%0A HTTP/1.1" 200 13536
[24/Oct/2020 12:22:38] "GET /media/Maleficent%20Mistress%20of%20Evil.jpg%0A HTTP/1.1" 200 4839
[24/Oct/2020 12:22:38] "GET /media/Little%20Women.jpg%0A HTTP/1.1" 200 10424
[24/Oct/2020 12:22:38] "GET /media/Dolittle.jpg%0A HTTP/1.1" 200 11354
[24/Oct/2020 12:22:38] "GET /media/The%20Secret%20Garden.jpg%0A HTTP/1.1" 200 14059
[24/Oct/2020 12:22:38] "GET /media/Legacy%20of%20Lies.jpg%0A HTTP/1.1" 200 10045
[24/Oct/2020 12:22:38] "GET /media/Theatre.jpg%0A HTTP/1.1" 200 11353
```

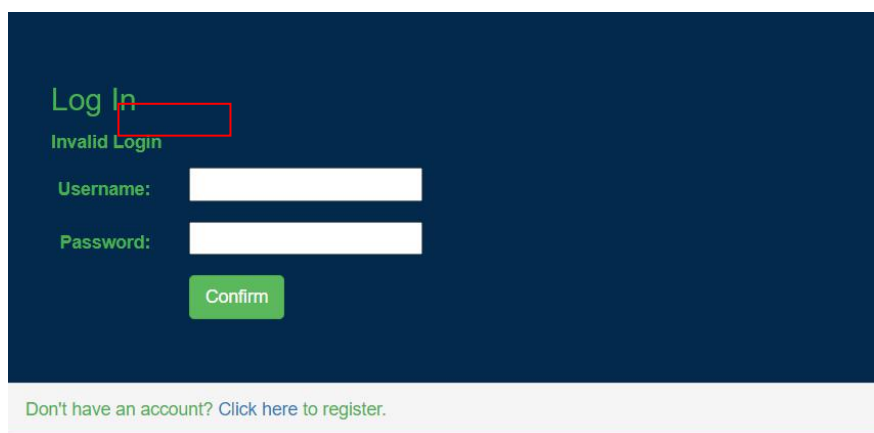
Enter the URL in the browser: <http://127.0.0.1:8000/>



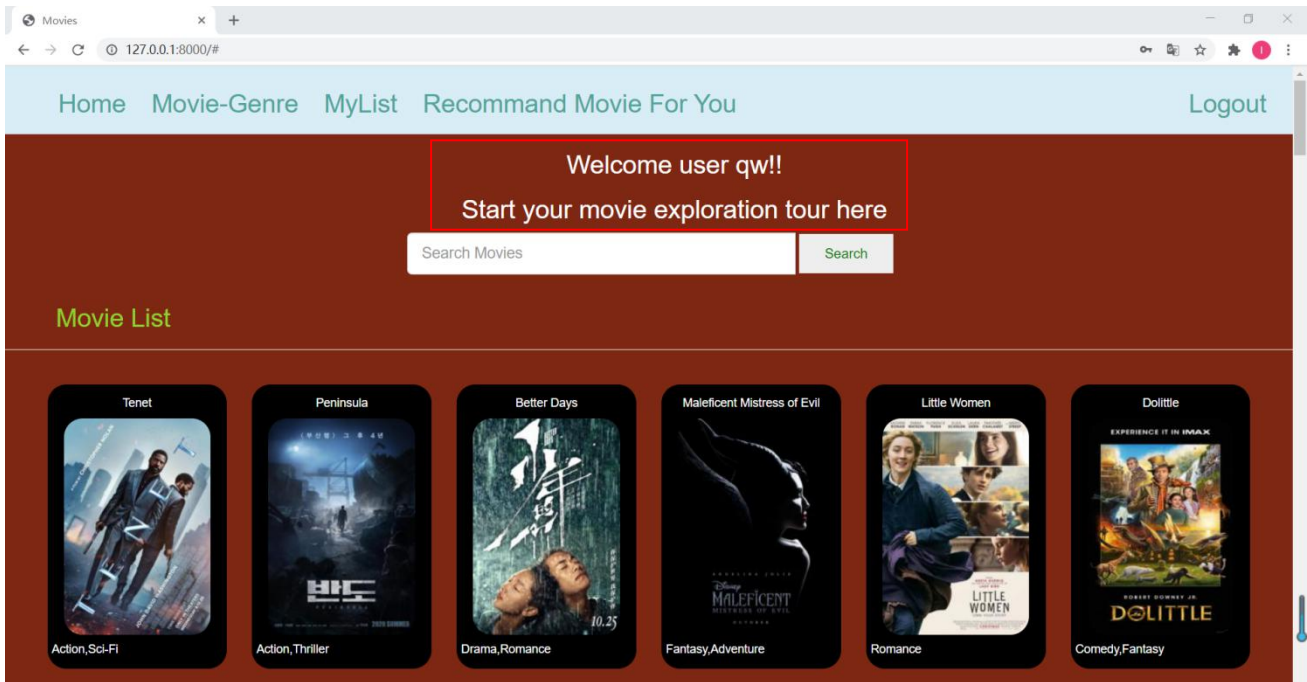
Click 'Login' button



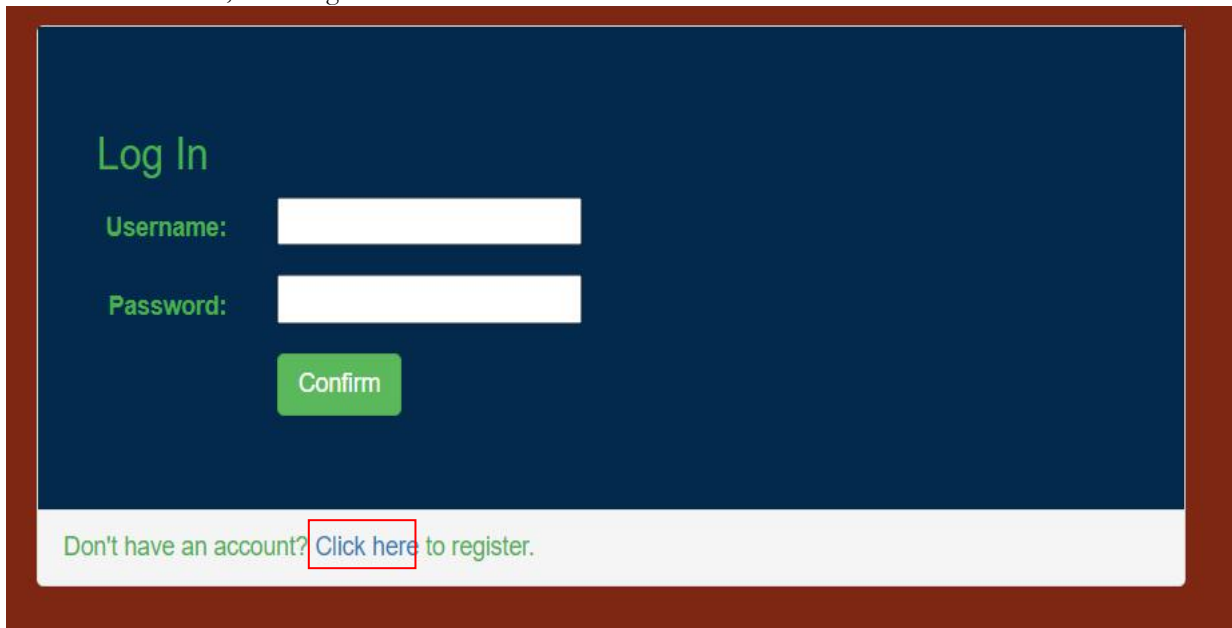
If the username and password are wrong. It will have alert 'Invalid Login'



If the username and password are correct. It will jump to 'Home' page and has this title.



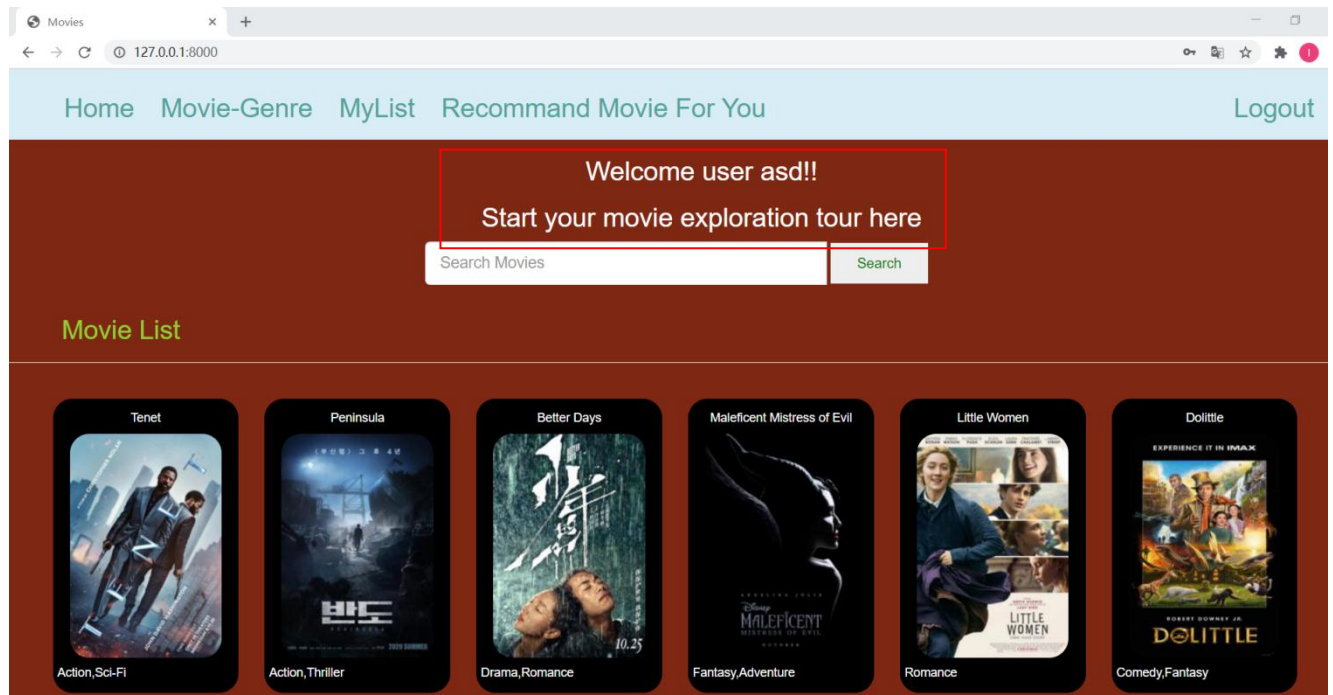
If there is a new user, it can register a new account.



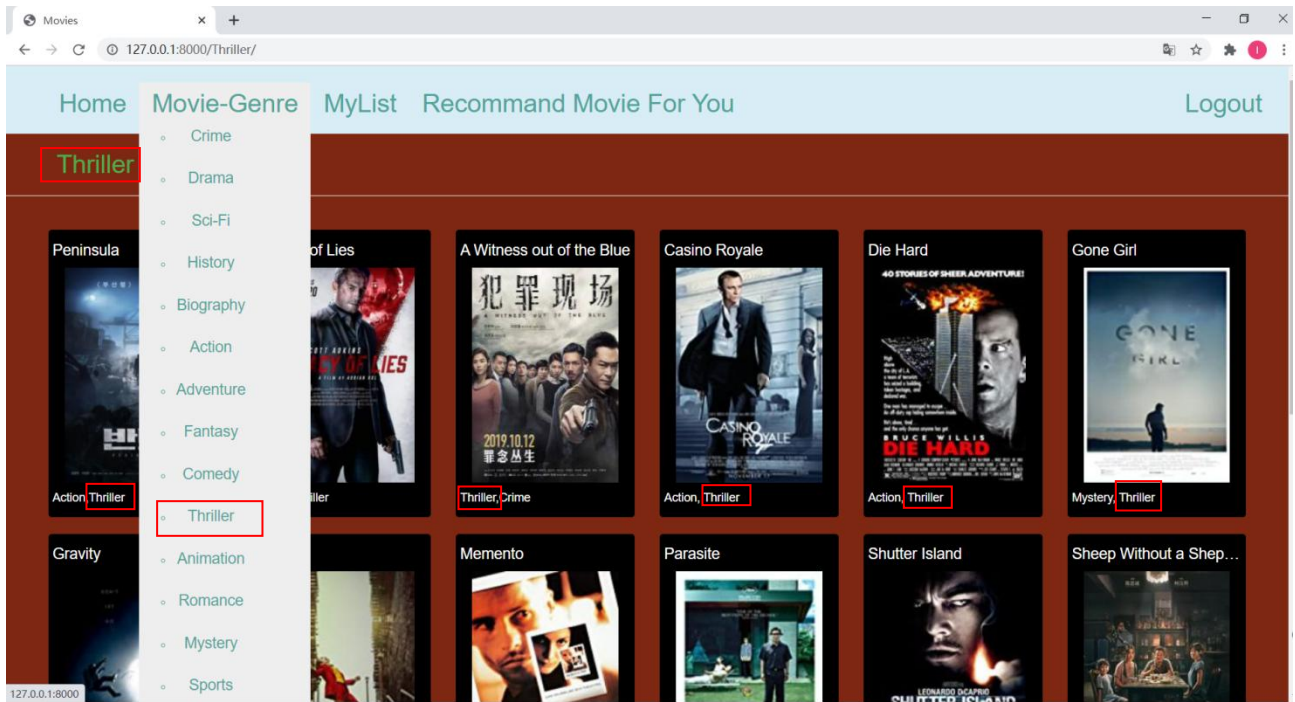
If email address isn't in correct form. It will have alert.(eg. Please include @ in your email)

The screenshot shows a 'Sign up for an account' form on a dark blue background. The form includes fields for Username (filled with 'asd'), Email (filled with 'asa'), and Password (masked with '***'). A green 'Submit' button is at the bottom. A red tooltip with a warning icon and Chinese text is displayed over the Email field, stating: '请在电子邮件地址中包括“@”。“asa”中缺少“@”。' (Please include '@' in the email address. 'asa' is missing '@'). Below the form, a link says 'Already have an account? Click here to log in.'

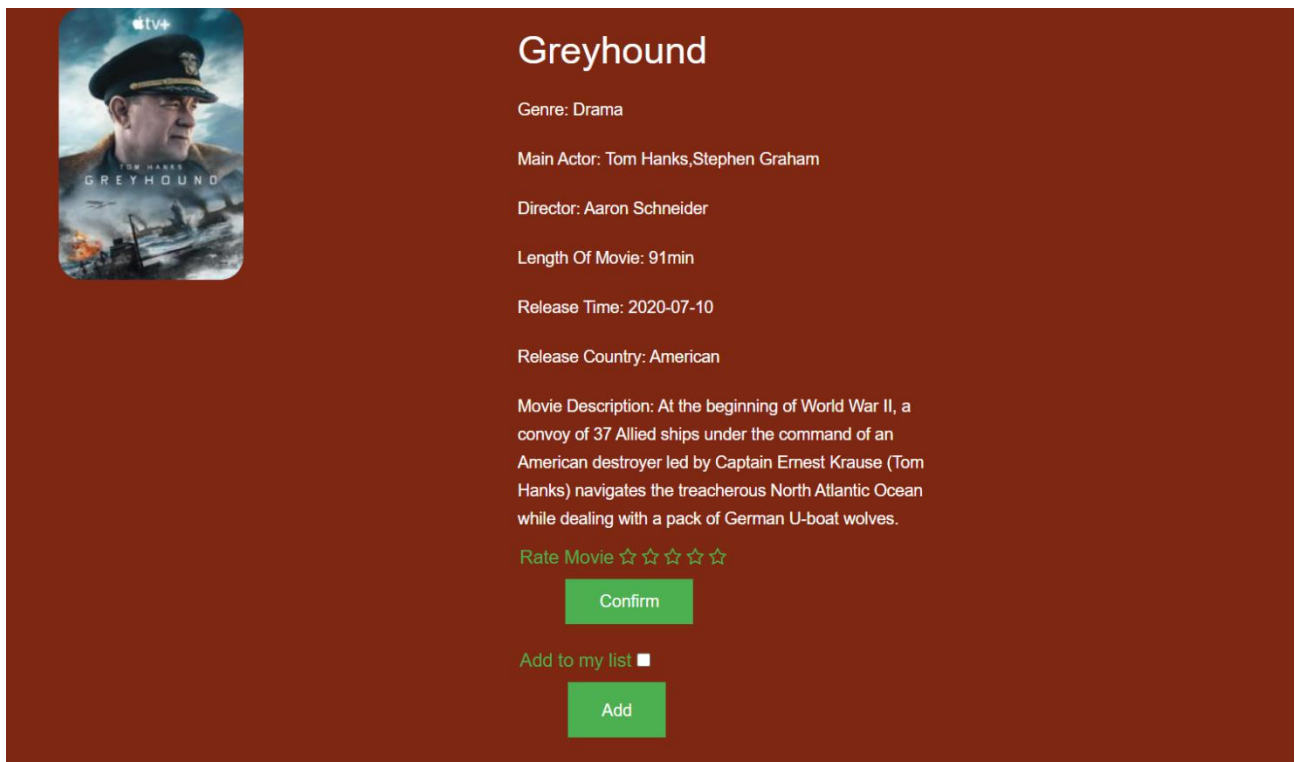
If all the information are correct. It will jump to 'Home' page.



Through 'Movie-Genre', it can check different type of movies.




Click the picture of any one of movie, it will jump to detail page.



In this interface, users can rate the movie.

Success! Rating has been submitted!



Greyhound

Genre: Drama

Main Actor: Tom Hanks, Stephen Graham

Director: Aaron Schneider

Length Of Movie: 91min

Release Time: 2020-07-10

Release Country: American

Movie Description: At the beginning of World War II, a convoy of 37 Allied ships under the command of an American destroyer led by Captain Ernest Krause (Tom Hanks) navigates the treacherous North Atlantic Ocean while dealing with a pack of German U-boat wolves.

Rate Movie ★★☆☆☆


Confirm

Add to my list ☐

Add

It can also add this movie into favourite list.

Success! Movie added to your list!



Greyhound

Genre: Drama

Main Actor: Tom Hanks, Stephen Graham

Director: Aaron Schneider

Length Of Movie: 91min

Release Time: 2020-07-10

Release Country: American

Movie Description: At the beginning of World War II, a convoy of 37 Allied ships under the command of an American destroyer led by Captain Ernest Krause (Tom Hanks) navigates the treacherous North Atlantic Ocean while dealing with a pack of German U-boat wolves.

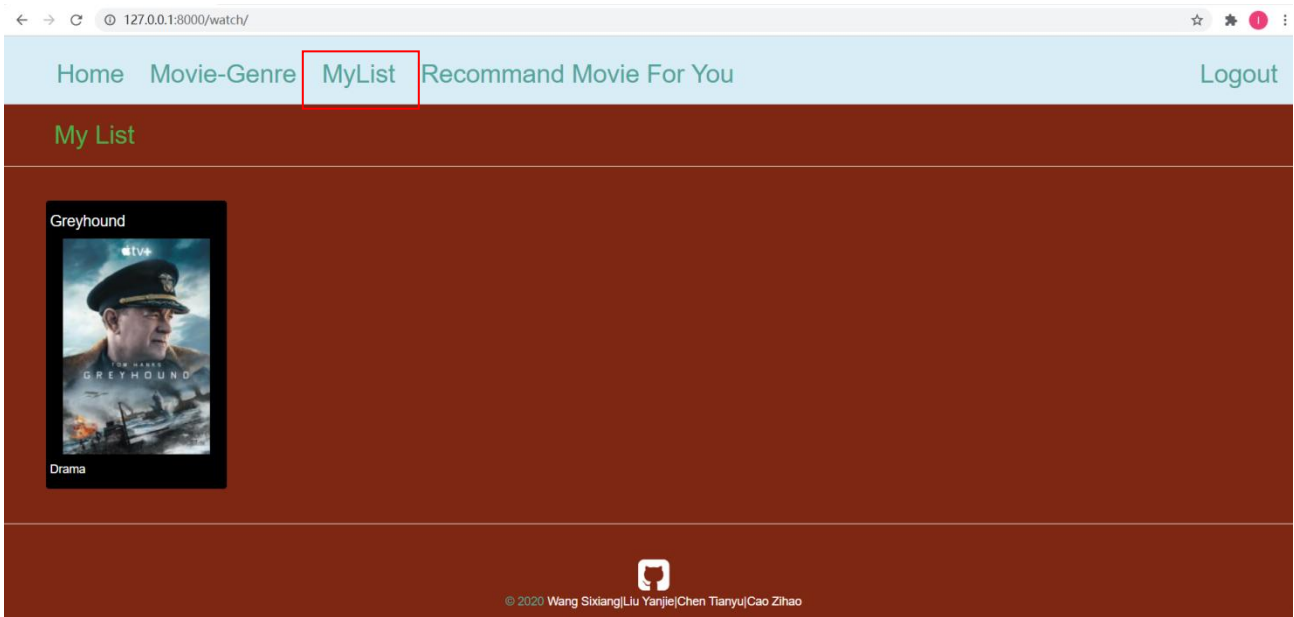
Rate Movie ★★☆☆☆

Confirm

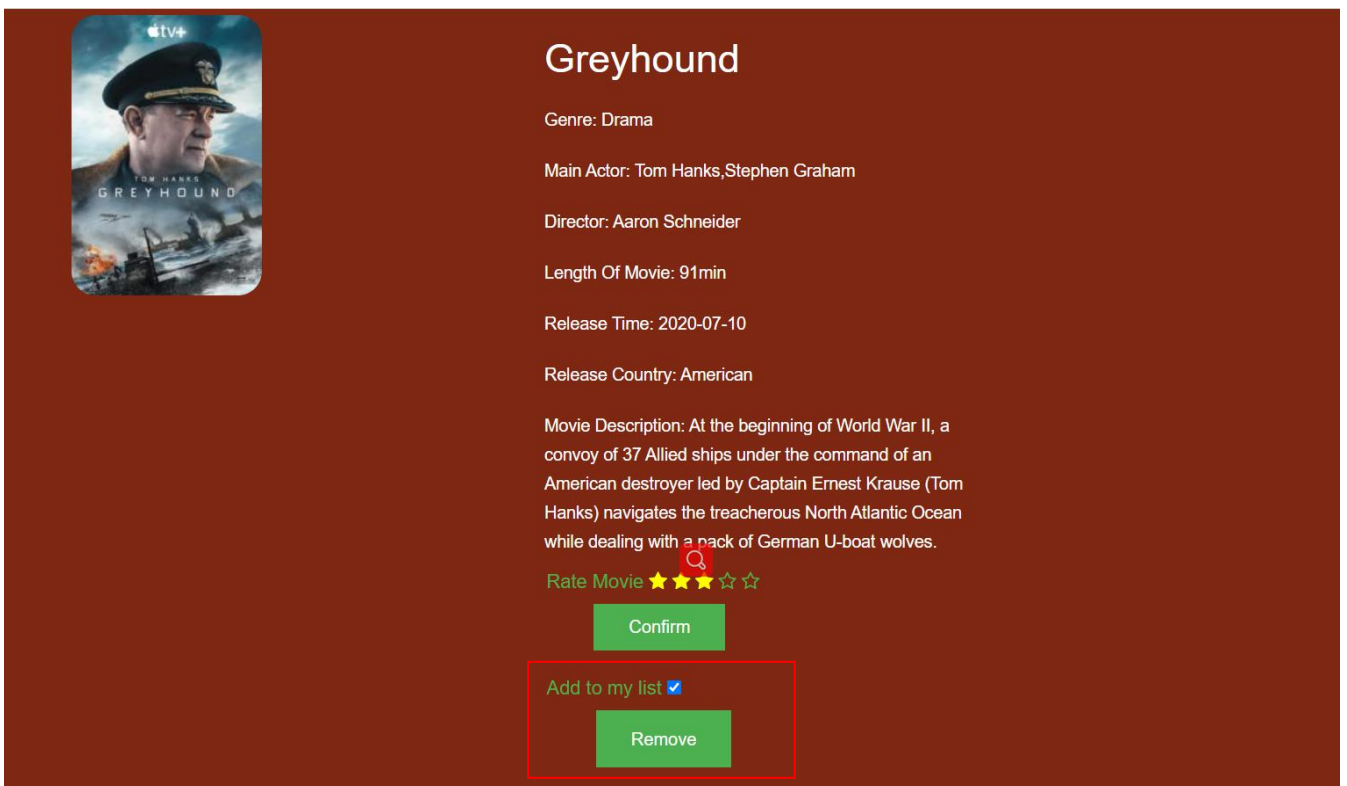
Add to my list ☒

Remove

It can be checked at 'MyList'



Movies in "MyList" can also be removed.



Success! Movie removed from your list!



Greyhound

Genre: Drama

Main Actor: Tom Hanks, Stephen Graham

Director: Aaron Schneider

Length Of Movie: 91min

Release Time: 2020-07-10

Release Country: American

Movie Description: At the beginning of World War II, a convoy of 37 Allied ships under the command of an American destroyer led by Captain Ernest Krause (Tom Hanks) navigates the treacherous North Atlantic Ocean while dealing with a pack of German U-boat wolves.

Rate Movie ★★☆☆☆

Confirm

Add to my list ☐

Add

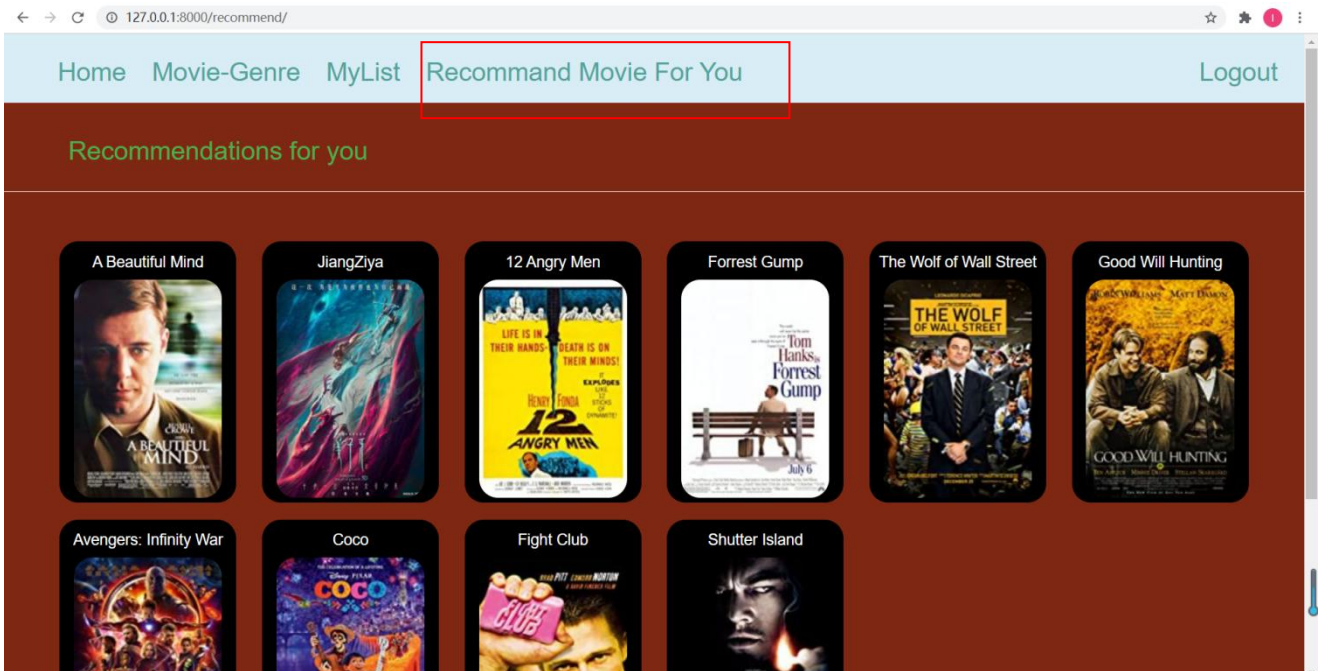
According to users' ratings and movies added to 'MyList', the system will automatically recommend a list of movies for users

Home Movie-Genre MyList Recommend Movie For You

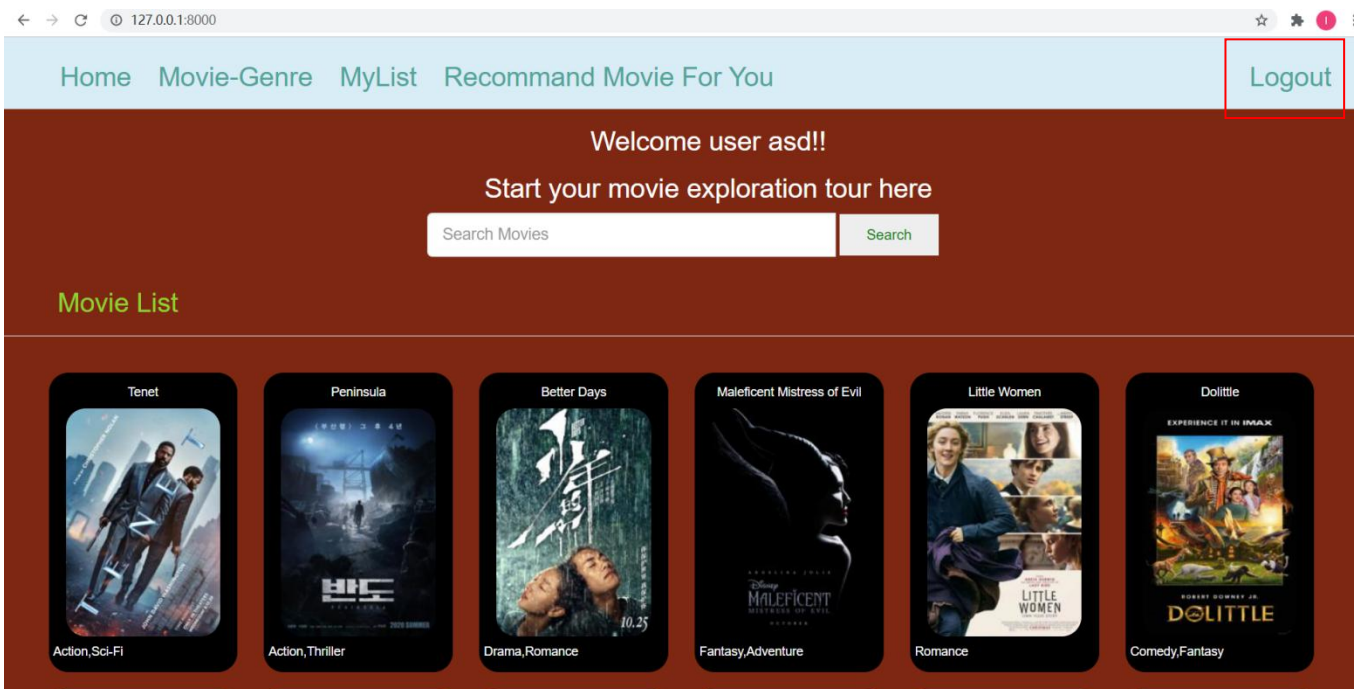
My List



© 2020 Wang Sixiang|Liu Yanjie|Chen Tianyu|Cao Zihao



Finally the user can log out of the system



APPENDIX C: PERSONAL REPORT

Master of Technology in Intelligent Systems

Individual Project Report

Your Name:	LIU YANJIE
Certificate:	Graduate Certificate in Intelligent Reasoning Systems

1. Your personal contribution to the project.

In this project, I built the back end of the system with Wang SiXiang. We used Django, a Python-based web application framework to build the entire system. I finished the algorithm part of recommendation. We chose a user-based collaborative filtering algorithm to implement the recommendation function. Algorithmic calculation through user information data stored in advance in the database. In addition, I also completed the implementation of related functions such as user login and registration. Finally I merged the back-end and front-end code and tested the system. What's more, I finished the report and made the video of the system.

2. What you have learnt from the project.

Through the study of this project, I learned more about the whole process of knowledge modeling. Not only that, I have also accumulated experience in writing code, learned the Django architecture and better understand how to complete the construction of the front and back ends of the complete web page. At the same time, in the process of consulting the literature, I also learned many algorithms for recommendation systems. Especially in the end we use the user-based collaborative filtering algorithm on the system.

Meanwhile, In the process, I learned how to do better group work. It is very important for everyone to have a clear division of labor and complete their tasks on time. When encountering difficulties, team members need to help each other to deal with and solve problems. Timely communication is a key element of group cooperation.

3. How you can apply this in future work-related projects.

In actual work, being familiar with the entire process of knowledge modeling can give a more complete understanding of the process of a project, not just a part of it. Of course, being more familiar with programming codes and algorithms will help the completion of the project.

Not only that, most projects in the future are not individual tasks but group tasks, so learning to get along with group members is a very important skill. Through this group cooperation, I better learn how to run in with group members with different personalities and complete a project together. I think this is very important for future work.

Individual Project Report

Your Name:	WANG SIXIANG
Certificate:	Graduate Certificate in Intelligent Reasoning Systems

1. Your personal contribution to the project.

In this project, I'm responsible for building back-end frameworks and algorithms. The back-end framework we use is Django, and the language is python. Django is an open source model view controller (MVC) style web application framework driven by python programming language. I use model to build up three classes: Movies, Myrating and MyList. Movies represent the movie shown in the website. I And for Myrating class, it has three fields: user, movie and rating. For Mylist class, it also has three class: user, movie and watch (in user's list). These are three objects I create for the website.

The most important part of the framework is the view class, which is responsible for processing the requests sent by the front end. The back end processes these requests and returns the corresponding values to the front end. First I use index function to show all the movies in the home page, which is to show the logo, title and genre of the movie. And next I use a detail function to show details of the movie. In this function, I processed two post requests. The first is to write a function to add movie to Mylist. And another is to submit the user's rating to each movie. And this function also display ratings in the movie detail page. Additionally, I also write some functions to filter different genres of movies and displays each type of movie. This the whole part of back-end work, which is the job of me and Liu Yanjie.

2. What you have learnt from the project.

I learned a lot in this project. First of all, I have accumulated some experience in web development. The back-end framework is an important part. I learned how to use Python's Django framework to build the back-end, and how to interact with the front-end to receive the request from the front-end, and define the function to process the request and return the data to the front-end successfully. This is some basic programming experience of web development. Through the practice of this project, I am more familiar with the whole framework and HTTP protocol, which has consolidated and improved my basic computer ability. And by writing code, my programming ability has also been improved, which will help me a lot in the future.

Secondly, in the process of learning the recommendation algorithm, I learned some principles of the algorithm, that is, the whole process and implementation of collaborative filtering. The algorithm is implemented successfully in Python language. This enables me to further understand the algorithm, and has a certain improvement in the ability of the algorithm

3. How you can apply this in future work-related projects.

First of all, our back-end framework, Django, is a very popular framework that many companies use. Django is a high-level python programming language driven open source model. View, controller style web application framework, which originated from the open source community. With this architecture, programmers can easily and quickly create high-quality, easy to maintain, database driven applications. Therefore, this project has strengthened my understanding and mastery of this framework, which is of great help to my future work.

Secondly, in the process of completing this project, my programming ability has been improved to a certain extent. Internet companies need to have a strong programming ability, so this will help my work in the future. Additionally, the learning of algorithm also has a great help to my future work, it deepens my understanding of machine learning and practical application.

Master of Technology in Intelligent Systems

Individual Project Report

Your Name:	CHEN TianYu
Certificate:	Graduate Certificate in Intelligent Reasoning Systems

1. Your personal contribution to the project.

In this project, I'm responsible for building front-end frameworks. The front end consists of HTML, CSS and JavaScript. My main task is to design the layout and style of web pages through these front-end tools. First, base.html File is the basis of the front-end, it is used to show all the elements of the front-end main interface. There are also many links to jump to other interfaces. Next, in the navigation bar, movie genre is a classification of movie categories, which contains various types of movies. I made each movie category into an HTML file. These files can filter each other's movies through the back-end functions and display them through the front-end. When the user selects any one of the movies, the system will jump to the HTML file to display this type of movie. The mylist button represents the movie added to the user list. The recommend movie for you button indicates the movie recommended to the user. There is a search bar in the middle of the home page. Each movie is displayed with a logo, a movie name and a movie category. In addition, this system design the sign up and login functions. The signup interface requires the user to enter the username, email address and password. After signup, user can enter the username and password to login.

2. What you have learnt from the project.

In this project, I learned to use tools and techniques I know to realize our goal such as HTML, CSS and JavaScript. Besides, I also learned to cooperate with other teammates, and to realize the request brought up the person in charge of designing the system, and modify the website frontend to be fit for movie representation.

3. How you can apply this in future work-related projects.

I can move on to design and implement more functional dynamic websites, as well as make a website of my own to display projects I have done.

Master of Technology in Intelligent Systems

Individual Project Report

Your Name:	Cao Zihao
Certificate:	Graduate Certificate in Intelligent Reasoning Systems

1. Your personal contribution to the project.

I designed a database to store the movie data in the project web page. Our project uses SQLite. It is a light database, is abided by acid relational database management system, its design goal is embedded, and has been used in many embedded products, it takes up very low resources, in embedded devices, may only need a few hundred K internal storage. It can support Windows / Linux / Unix and other mainstream operating systems, and can be combined with many programming languages. So that's why we chose this database.

In our project database, I designed four main tables. The first table is a user table, which stores the user's information. It has 11 fields. This table is used to store all the movies in the project. It is a small movie library. It has 10 fields, which are movie ID, title, genre, logo, description, star, director, length, release time and country. These 10 fields fully describe a movie. The next table is the user Mylist table, which stores the movies each user adds to the movie list. It has four fields: ID, watch, movie ID, and user ID. Where watch is Boolean data, which represents whether the user has added the movie to the list, 1 represents add, and 0 represents removal. The movie ID is the field associated with the movie table, and the user ID is the field associated with the user table. The last table is myrating table. It also has four fields, which are ID, rating, movie_id and user_id. Rating stores the rating of movie for each user. It has value from 1 to 5. And the movie ID and user ID is also associated with the movie table and user table respectively.

2. What you have learnt from the project.

The first is the selection of the tools needed in the project. The quality of the tools largely determines the presentation level of the final effect. Since our data volume in this project is not particularly large, we hope that the selected database will not take up too much space. So despite many attempts, I finally chose the right tool because I knew the importance of the tool.

Master of Technology in Intelligent Systems

Secondly, I have strengthened my understanding of SQL language. Since I seldom contacted with database operation before, I hope to improve myself through this module in this project. I encountered great difficulties in the early stage, but I learned how to meet the requirements of this project through different ways of learning. After several attempts, I built four main tables to satisfy the user information, including the storage of the Movie List and the storage of all the movies associated with it.

Finally, I became more aware of the importance of team cooperation. This project needed to import stored data into the algorithm model, so for different types of data, how to invoke them should be communicated in advance. In addition, when I encountered the problem that the data could not be exported, the team members also provided me with necessary help to make the project proceed smoothly.

3. How you can apply this in future work-related projects.

The application of database is now very extensive. In my previous internship, I called the member database of the target company, and I could create the database by myself and import the required data, so as to reduce unnecessary data acquisition. Too much data would sometimes affect the efficiency of the algorithm, and only the useful ones could be extracted. Better grasp of the use of database can effectively process data and improve work efficiency.

In addition, the film recommendation system can be applied in other recommendation environments, conducive to music, tourism, etc., and can be extended to recommend tourist routes, etc. More complex recommendation effects can be achieved by importing more types of data for training.