MyMovie

MOVIE RECOMMENDATION SYSTEM

PROJECT REPORT

FOR MOVIE LOVERS
PICK YOUR FAVORITES
ENJOY YOUR FAVORITE MOVIE TIME

Movie For You

Report date | 2022 10.28

Team members/Cai zimoA0261828B /Hu ZhiqingA0261637B/ Yu hanchunA0261716B/Lu qinwenA0261847W

Contents

0. ABSTRACT 3
1. INTRODUCTION 3
1.1 Background and objetcives 3
1.2 Movie recommendation systems development status 3
1.3 Commercial value and market research 4
2. PROBLEM DESCRIPTION 5
3. RECOMMENDATION SYSTEMS 6
3.1 Common recommending methods 6
3.2 System architecture 6
3.3 Recommendation algorithm 7
3.3.1 Content-based recommendation 7
3.3.2 Collaborative Filtering 8
3.4 Data collection 9
3.4.1 Web Spider 9
3.4.2 MovieLens 10
3.4.3 User data 11
3.5 Data preprocessing 12
3.6 Cold start 13
3.7 Model performance evaluation 14
4. CONCLUSIONS AND FUTURE WORK 14
4.1 Conclusions 14
4.2 Future work 15
5. REFERENCES 15
6. APPENDIX 16
6.1 Project proposal 16
6.2 Mapped system functionalities 20
6.3 Installation and user guide 21
6.3.1 Requirement 21
6.3.1.1 Golang server configuration 21
6.3.1.2 Python server configuration 22
6.3.1.3 Java server configuration 23
6.3.1.4 Redis configuration 24
6.3.1.5 Mysql Configuration 25
6.3.1.6 Docker configuration 26
6.3.1.7 Browser requirement 28
6.3.2 Best Practice 28
6.3.2.1 User information 28
6.3.2.1.1 Login/Logout 28
6.3.2.2 Functions of system 29
6.3.2.2.1Search movies 29
6.3.2.2.1Recommend movies 30
6.3.2.2.3 Rating a movie 31

6.3.2.2.4 Performance of system 32
6.3.3 Interface document 34
6.3.4 Necessary SQL 38
6.4 Individual project report 38
6.4.1 Cai Zimo A0261628B 38
6.4.2 Lu Qinwen A0261847W 40
6.4.3Hu Zhiqin A0261637B 41
6.4.4Yu Hanchun A0261716E 42

0. ABSTRACT

As a popular art form widely enjoyed by the massive, film has developed rapidly in recent years. However, the numerous variety and huge number of films became dizzying for people to choose to watch. It is often the case that one might find it difficult to watch a movie since it always takes too much time to pick his movie. Although there are various video websites, most of them are piled up with popular information and lack personalized recommendations. Our team takes this as an opportunity to integrate multiple recommendation algorithms to make recommendations in appropriate situations, and help users find their favorite movies conveniently.

1. INTRODUCTION

1.1 Background and objectives

With the rapid development of the economy, movies have entered people's lives and are loved and welcomed by the masses. We enjoy the fun of movie boost and suffer from picking one movie to watch at the same time since all of us must be familiar with the scene: Endlessly scrolling through Netflix, watching trails on YouTube, looking up IMDb ratings, wasting half an hour and still cannot decide what to watch.

Although there are various video websites, most of them are piled up with popular information and lack personalized recommendations. It is difficult for users, especially those who suffer from decidophobia, to find movies that meet their interests and tastes.

So this project takes this as an opportunity to integrate multiple recommendation algorithms to make recommendations in appropriate situations and help users find their favorite movies in different situations in a simple manner.

1.2 Movie recommendation systems development status

Now there are many big movie recommendation websites and softwares, such as Netflix, PickAMovieForMe, Douban and so on. After analyzing their website we found that they have their own Characteristics.

In addition to knowing what a user is watching on Netflix, it also looks at things like how long the user watched, which device the user is watching Netflix on, and how long he watched.When the user enter a search query, the top results it returns are based on the actions of other members who have entered the same or similar queries.

PickAMovieForMe, a well-known movie recommendation website, uses quiz-based movie picker to find movies that fit in the users' mood,occasions and

individual preferences.

As a famous movie evaluation social platform in China, Douban applies the item-based collaborative filtering algorithm to make personalized movie recommendations for users based on movie data and user viewing and review data.

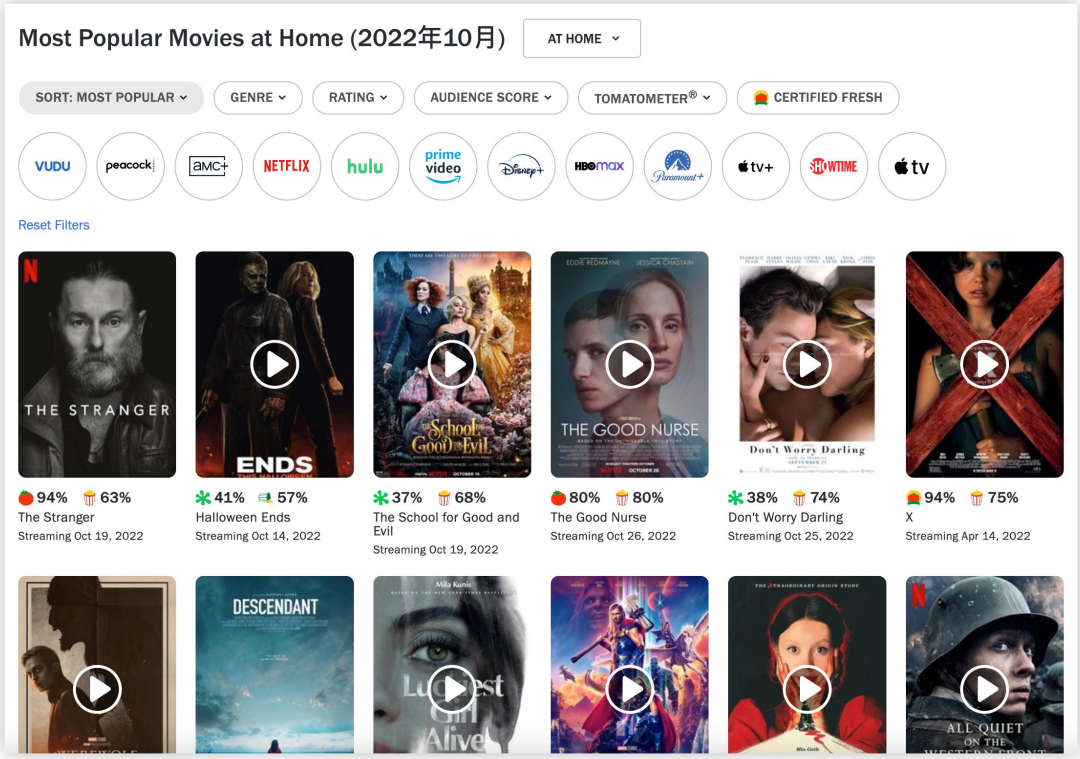


Fig.1 movies from rottentomatoes.com

1.3 Commercial value and market research

The core of the movie recommendation system is the recommendation algorithm. The recommendation algorithm is widely used in the Internet industry, and the commercial value of recommended system is reflected in various aspects.

One well-known case is TikTok. Ad targeting using the recommendation algorithm has undoubtedly played an important role in the gradual rise of TikTok's advertising revenue. According to ByteDance's annual financial statements, as of 2020, TikTok used targeted advertising strategies to generate $202.1 billion in revenue in 4.6 years. With the help of the recommendation algorithm, the precise placement of advertisements can be realized, and the advertisements can generate greater commercial value by matching the huge traffic of personalized recommendations and the precise interests of users. Similarly to this, the e-commerce platform recommends products to the people who may buy them through personalized recommendation, promotes the purchase of the products, and obtains more shares from the merchants. Similar to how the role of a real estate agent can be realized through e-commerce.

Movie recommendation systems also have a huge market. Netflix as a large

American subscription streaming service company holds a Netflix Prize competition every year. Netflix's million-dollar Netflix Prize competition launched in 2009 brings recommendation systems into business. It attaches great attention to this technology by spending millions of dollars to analyze recommendation systems which help it save more than 1 billion dollars each year. This shows the great value of the movie recommendation system.

The commercial value of the recommendation system is reflected in the realization of advertising value-added services, user growth, the creation of user and platform boding, and labor cost savings.

While our recommendation system brings a great number of users to the platform, it can generate revenue from it. For example, large amount of users will attract a large number of merchants to make advertisements, and the platform only has to charge for advertisements to obtain considerable advertising revenue.

2. PROBLEM DESCRIPTION

With the development of information technology and the Internet, people have gradually moved from an era of information scarcity to an era of information overload. In this era, both information consumers and information producers are facing great challenges. As an information consumer, it is very difficult to find the information you are interested in from a large amount of information; As an information producer, it is also a very difficult thing to make the information produced stand out and attract the attention of the majority of users. A recommendation system is an important tool to solve this contradiction. The task of the recommendation system is to contact users and information. On the one hand, it helps users find valuable information for themselves, and on the other hand, information can be displayed in front of users who are interested in it, so as to achieve a win-win situation for information consumers and information producers.

Not just used in movie recommendation， the recommendation system is widely used in other Internet industries. In abstract, the recommendation algorithm is a fitting function for content satisfaction, involving user characteristics and content characteristics. The collaborative filtering algorithm and Content-based recommendation algorithm are the two most widely used recommendation algorithms in movie systems.

After much consideration and discussion, we notice that the issues we need to consider include the following aspects: what kind of data is used to train the machine to judge the movies that the user may like, how to obtain the data set of the user's watching movies, and how to understand the user's favorite without infringing on the user's privacy type of movie, etc.

As a result, our team aims to design a personalized recommendation system based on the combination of Collaborative Filtering algorithm and Content-based recommendation algorithm and build the supporting system architecture and server system.

3. RECOMMENDATION SYSTEMS

3.1 Common recommending methods

Generally speaking, we may decide which movie to watch in the end in the following ways:

Ask a friend. We may open the chat tool, find some good friends who often watch movies, and ask them if they have any movies to recommend. Also, we can log in to the microblog, post on the topic, "I want to see a movie", and wait for movies recommended by the enthusiasts. This method is called social recommendation in the recommendation system, that is, let friends recommend items to themselves.

Use the search engine, to search for their favorite actors. We can search for our favorite actors and directors to see movies we have not seen in the returned results. Pretty much similar to this, the recommendation system can analyze users' movie history, find the actors and directors they might like, and then recommend other movies. This recommendation method is called content-based recommendation in the recommendation system.

check the ranking list, such as the IMDB ranking list to see what other people are watching and what other people like, and then find a movie that is widely praised. This method can be further expanded: if you can find a group of users with similar historical interests and see what movies they are watching recently, the results may be more consistent with your interests than the broad popular leaderboards. This method is called collaborative filtering-based recommendation.

The above three methods are the essence of our recommendation system, which connects users and items in a certain way.

3.2 System architecture

The core of the recommendation system is the data and model algorithm：

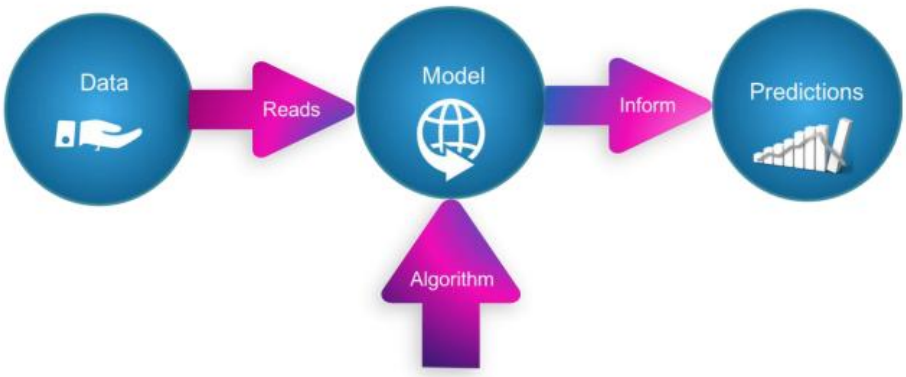


Fig.2 how the system recommends

By burying a point in the movie website system, users' click events (such as which movie they like or the rating of a movie) are obtained and the information is sent to the recommendation system. The recommendation system makes corresponding processing according to the information, and stores the recommendation results in the MySQL database. The web front end displays the recommended movie to users through the query database.

After much consideration, the structure of our designed recommendation system is shown below:

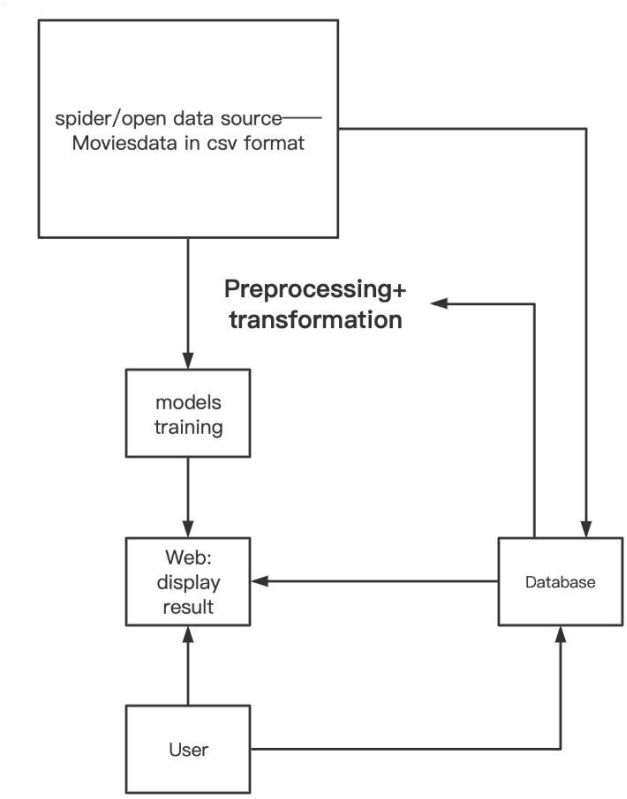


Fig.3 system architecture

3.3 Recommendation algorithm

the recommendation algorithm is a fitting function for content satisfaction, involving user characteristics and content characteristics.

3.3.1 Content-based recommendation

The basic principle of the Content-based recommendation algorithm is to obtain the user's interest preference according to the user's historical behavior, and recommend the target object similar to his interest preference for the user.

Three core steps of content-based personalized recommendation:

Recommend based on user historical behavior records

Both user and target features are represented by explicit labels, which are used to make recommendations.

Users and objects are embedded in the same vector space, and recommendation is made based on vector similarity

Thus,we have to complete its item profiles,user profiles and similarity measure.However,what we also need to pay attention is that the algorithm cannot recommend content other than the user's interests, and the user must have a wider range of interests for recommendation.

After much consideration, we decided to build the recommenders based on movie genre and then calculate the similarity between movies based on certain metrics and to recommend movies that are most similar to a particular movie that the user likes.We convert the target feature into a vectorized representation, and calculate the cosine value of the two feature vectors. The larger the cosine value, the smaller the angle between the vectors, indicating the higher the similarity.The calculation formula of similarity can be expressed as follows:

$$u(c,s) = \cos(\omega_c,\omega_s) = \frac{\sum_{i=1}^{K} \omega_{i,c}\omega_{i,s}}{\sqrt{\sum_{i=1}^{K} \omega_{i,c}^{2} \sum_{i=1}^{K} \omega_{i,c}^{2}}}$$

Firstly, we model the properties of the movies, taking advantage of the genre of the movie. Secondly we And then we built profiles of the movies, which are included in metadata dumps we create separately for each movie. Then according to the learned preferences of the user: like comedy, action novels, and then through similarity calculation, and recommend movies with high similarity to the user.

3.3.2 Collaborative Filtering

We use two CF algorithms one is a user-user-based CF algorithm whose basic principle is to find a set D similar to the user's score for a user, and then estimate the user's score based on set D, the other is item-item based, whose core is to to find other similar items for an item S, and estimate the item S' rating based on similar items.

Two steps of the Users CF algorithm:

1. First, calculate the similarity between the target user and users A, B, C, and D according to the previous user's rating of the movie (or the existing user vector), and find the n users that are most similar to the target user.

2. According to the similarity of the ratings of the n users to the movie X and the target user, the target user's rating of the movie X will be guessed. If the rating is relatively high, movie X will be recommended to the target user, otherwise, it will not be recommended.

To measure the similarity, the two algorithms are similar, both of them use the calculated value of vector cosine angle as similarity.

3.4 Data collection

3.4.1 Web Spider

movies

movieId	title	genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	Comedy|Drama|Romance
5	Father of the Bride Part II (1995)	Comedy
6	Heat (1995)	Action|Crime|Thriller
7	Sabrina (1995)	Comedy|Romance
8	Tom and Huck (1995)	Adventure|Children
9	Sudden Death (1995)	Action
10	GoldenEye (1995)	Action|Adventure|Thriller
11	American President, The (1995)	Comedy|Drama|Romance

Fig.4 Movies dataset before crawling

id	name	url	time	genre	release_time	intro	directors	writers	stars
1	Toy Story	https:	1h 21min	Animatior	22 November 1995 (I	A cowboy dc	John Lasset	John Lasseter|Pe	Tom Hanks|Tim Allen|Don Rickles
10	GoldenEy	https:	2h 10min	Action|Ad	17 November 1995 (U	Years after a	Martin Camp	Ian Fleming|Micha	Pierce Brosnan|Sean Bean|Izabella Sc
100	City Hall (	https:	1h 51min	Drama|Th	16 February 1996 (US	The acciden	Harold Beck	Ken Lipper|Paul S	Al Pacino|John Cusack|Bridget Fonda
100044	Human Pl	https:	50min	Documen	TV Mini-Series (2011	Like all life fc	(NULL)	(NULL)	John Hurt|Roger Munns
100068	Comme u	https:	1h 24min	Comedy|[	7 March 2012 (Franc	A veteran ch	Daniel Coher	Daniel Cohen|Oliv	Jean Reno|Michaël Youn|Raphaëlle Ag
100083	Movie 43	https:	1h 34min	Comedy	25 January 2013 (US	A series of ir	Elizabeth Ba	Rocky Russo|Jere	Emma Stone|Stephen Merchant|Richa
100106	The Perv	https:	2h 16min	Documen	4 October 2013 (UK)	Philosopher	Sophie Fienr	Slavoj Zizek	Slavoj Zizek
100159	Sightseer	https:	1h 28min	Adventur	30 November 2012 (I	Chris wants	Ben Wheatle	Alice Lowe|Steve	Alice Lowe|Kenneth Hadley|Steve Ora
100163	Hansel &	https:	1h 28min	Action|Fa	25 January 2013 (US	Hansel & Gr	Tommy Wirk	Tommy Wirkola	Jeremy Renner|Gemma Arterton|Peter
100194	Jim Jeffe	https:	1h 1min	Comedy	TV Special 30 Octob	Foul-mouthe	Jay Karas	Jim Jefferies	Jim Jefferies|Al Jackson
100226	Why Stop	https:	1h 25min	Comedy|[	17 August 2012 (USA	When a colle	Phil Dorling|I	Phil Dorling|Ron M	Jesse Eisenberg|Melissa Leo|Tracy Mo
100277	Tabu (20	https:	1h 58min	Drama|Ro	5 April 2012 (Portuga	A restless re	Miguel Gome	Miguel Gomes|Ma	Telmo Churro|Miguel Gomes|Hortência
1003	Extreme M	https:	1h 58min	Crime|Dra	27 September 1996 (	A young doc	Michael Apte	Michael Palmer	Hugh Grant|Gene Hackman|Sarah Jes
100302	Upside D	https:	1h 49min	Drama|Fa	7 March 2013 (China	Adam and Ei	Juan Solanas	Juan Solanas|Jua	Jim Sturgess|Kirsten Dunst|Timothy S
100304	The Liabi	https:	1h 22min	Comedy|(	17 May 2013 (UK)	When 19-ye	Craig Viveiro	John Wrathall	Tim Roth|Jack O'Connell|Peter Mullan
100306	Angst (19	https:	1h 27min	Drama|Hc	1983 (USA)	A troubled m	Gerald Kargl	Zbigniew Rybczy	Erwin Leder|Robert Hunger-Bühler|Sil
100326	Stand Up	https:	1h 35min	Comedy|(	1 February 2013 (USA	A pair of agi	Fisher Steve	Noah Haidle	Al Pacino|Christopher Walken|Alan Arl
100383	Side Effe	https:	1h 46min	Crime|Dra	8 February 2013 (US	A young wor	Steven Sode	Scott Z. Burns	Rooney Mara|Channing Tatum|Jude La

Fig.5 Movies dataset baftercrawling

```python
import requests
from bs4 import BeautifulSoup
import unicodedata
import logging
import csv
import time


class Model():
    def __init__(self):
        # 请求头
        self.headers = {
            'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.162 Safari/537.
        }
        # 存放每一步电影的id和imdb的id
        self.movie_dct = {}
        # 存放已经处理完的movie id
        self.white_lst = []
        # 电影详情的初始url
        self.url = 'https://www.imdb.com/title/'
        self.movie_csv_path = '../ml-latest-small/links.csv'
        # 海报的保存路径
        self.poster_save_path = './poster'
        # 电影信息的保存文件
        self.info_save_path = './info/info.csv'
        # logging的配置, 记录运行日志
        logging.basicConfig(filename="run.log", filemode="a+", format="%(asctime)s %(name)s:%(levelname)s:%(message)s",
                            datefmt="%Y-%m-%d %H:%M:%S", level=logging.INFO)
        # 表示当前处理的电影
        self.cur_movie_id = None
        self.cur_imdb_id = None
```

Fig.6 part of the crawler code

3.4.2 MovieLens

This dataset describes 5-star ratings and free-text tagging activity from MovieLens(http://movielens.org), a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users. All users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided.

The data are contained in the files `links.csv`, `movies.csv`, `ratings.csv` and `tags.csv`.

Our website records the necessary user data to supplement the content of this section.

movieId	title	genres
1	Toy Story (1	Adventure | Animation | Children | Comedy | Fantasy
2	Jumanji (19	Adventure | Children | Fantasy
3	Grumpier C	Comedy | Romance
4	Waiting to	Comedy | Drama | Romance
5	Father of th	Comedy
6	Heat (1995)	Action | Crime | Thriller
7	Sabrina (19	Comedy | Romance
8	Tom and H	Adventure | Children
9	Sudden Dec	Action
10	GoldenEye	Action | Adventure | Thriller
11	American P	Comedy | Drama | Romance
12	Dracula: D	Comedy | Horror
13	Balto (1995)	Adventure | Animation | Children
14	Nixon (199	Drama
15	Cutthroat I	Action | Adventure | Romance
16	Casino (199	Crime | Drama
17	Sense and S	Drama | Romance
18	Four Room	Comedy
19	Ace Ventur	Comedy
20	Money Trai	Action | Comedy | Crime | Drama | Thriller
21	Get Shorty	Comedy | Crime | Thriller

Fig.7 part of movie.csv file

3.4.3 User data

Users can enter the movie details page to view the movie information such as actors, screenwriters, directors, etc., and rate the movie. The scoring data will be recorded and saved in the background and used as an important user data source for the recommendation system.

Fig.8 user rating page

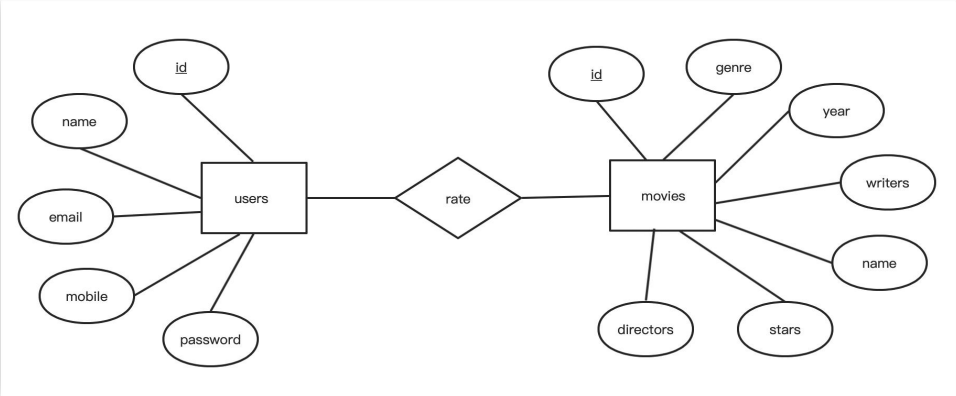


Fig.9 database ER sketch

3.5 Data preprocessing

ETL (Extract-Transform-Load), which is used to describe a series of processing processes between a data source and final storage, is used in data preprocessing. It generally goes through three stages extraction, transformation, and loading.

As mentioned above, the dataset we build contains three different sheets, we extracted data from Spider, and MovieLens and combined them with user data collected ourselves from our website by burying a point on the client side, and then uploading to the log collection web service through the HTTP protocol.

For missing movie rating values, we marked them as 0 for missing rating values. We sorted out the data of the movie year, genre, etc., and loaded our users' and movies' data in our database for further use.

First, we create a user-item matrix of users and movies in order to map the ratings of all users to all movies. For data that users have not rated, that is, missing values problem, we mark them as 0 to solve the problem. Then, according to the content-based recommendation algorithm, we use the movie genre and producing year as the main feature values. Another process is to divide the year of the movie into 3 types: 90s, 00s, and 10s, the type is also divided into Action and the other 24 species, and we use the one hot encoding format for marking. In this way, we can extract the characteristics of the movie, and analyze the similarity between the movies.

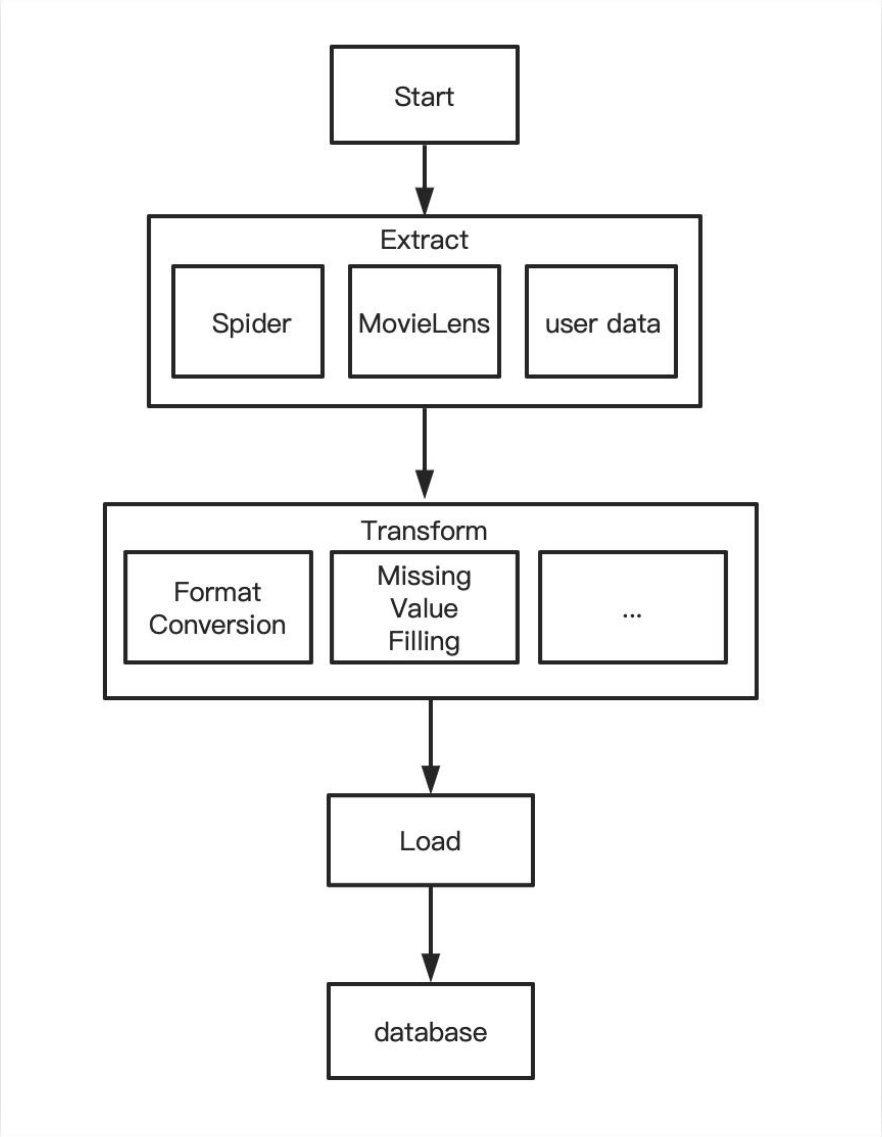


Fig.10 data preprocessing

3.6 Cold start

As for the recommender, we notice that there is a cold start issue, which means the recommender cannot work without users' data at the beginning. The usual practice is to ask users to choose their own topics of interest, groups, products, personalities, favorite music types, and other information, such as MovieLens:

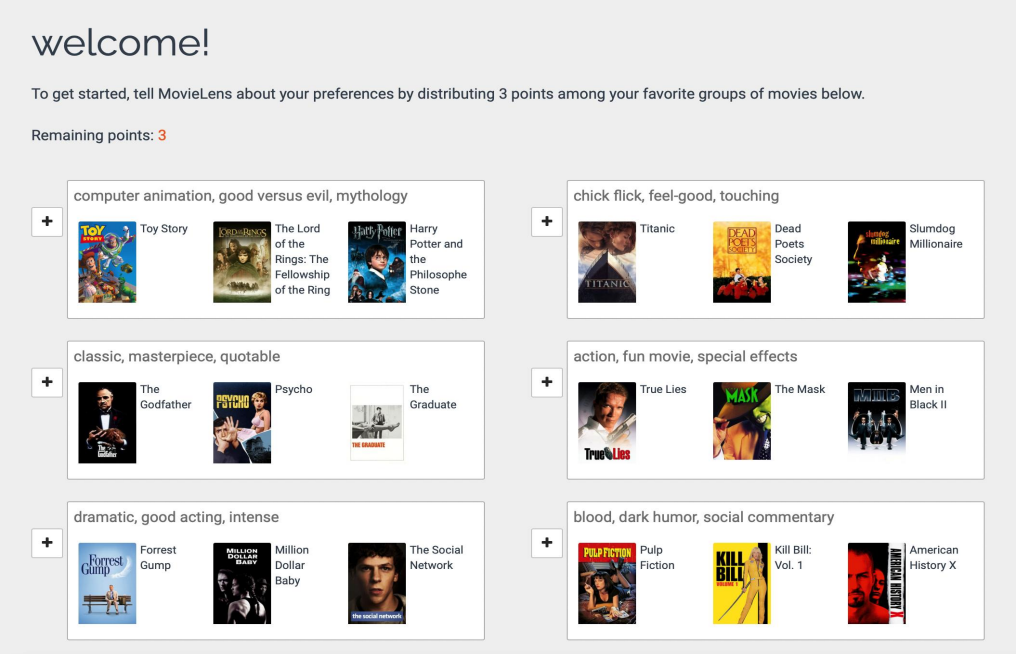


Fig.11 MovieLens start page

3.7 Model performance evaluation

We mainly used MSE and RMSE as Model performance evaluation methods.

MSE: The expected value of the squared difference between the predicted value and the actual value. The smaller the value, the higher the model accuracy.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

RMSE: It represents the sample standard deviation of the difference between the predicted value and the observed value. The smaller the value, the higher the model accuracy.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}$$

4. CONCLUSIONS AND FUTURE WORK

4.1 Conclusions

Overall, our team combined a content-based recommend algorithm and a collaborative filtering algorithm, both are core algorithms of our movie recommendation system.

As for the supporting system design:

Front-end: Vue CLI, make a request to the back-end, get the information from

the back-end for page display.

we designed three different back-end servers, using python, java, and go, they are:

1. Spring Boot: Based on Java language, it is mainly responsible for movie display, search, and recommendation functions

2. Gin: Based on the Go language, mainly responsible for user login and registration, movie rating functions

3. Flask: Based on the Python language, it is mainly responsible for the preprocessing and training of the recommended model, and returns the results to Spring Boot

4.2 Future work

Obviously, up to now, the recommendation system has functions that could meet the expectation of a demo. However, when the project is actually launched, it will bear high concurrence, so more operations are needed to optimize such scenarios. For example, we can design some multi-level caches, to reduce part of the pressure on the cache. However, at the time the resulting inconsistency between the database and the cache also needs to be considered.

At the same time, as a recommendation system, the performance of the recommendation algorithm itself has yet to be optimized. Whether the method of recommendation based only on similarity can be replaced by the algorithm with better performance is also one of the directions to be studied in the future.

5. REFERENCES

O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh,Recommendation systems: Principles, methods and evaluation,Egyptian Informatics Journal,ISSN 1110-8665

Pazzani M J, Billsus D. Content-based recommendation systems[M]//The adaptive web. Springer, Berlin, Heidelberg, 2007: 325-341.

Lops P, Jannach D, Musto C, et al. Trends in content-based recommendation[J]. User Modeling and User-Adapted Interaction, 2019, 29(2): 239-249.

Koren Y, Rendle S, Bell R. Advances in collaborative filtering[J]. Recommender systems handbook, 2022: 91-142.

Herlocker J L, Konstan J A, Riedl J. Explaining collaborative filtering recommendations[C]//Proceedings of the 2000 ACM conference on Computer supported cooperative work. 2000: 241-250.

Wang Z, Yu X, Feng N, et al. An improved collaborative movie recommendation system using computational intelligence[J]. Journal of Visual Languages & Computing, 2014, 25(6): 667-675.

Shcherbakov M V, Brebels A, Shcherbakova N L, et al. A survey of forecast error measures[J]. World applied sciences journal, 2013, 24(24): 171-176.

Khanal S S, Prasad P W C, Alsadoon A, et al. A systematic review: machine

learning based recommendation systems for e-learning[J]. Education and Information Technologies, 2020, 25(4): 2635-2664.

TMDB：https://www.themoviedb.org/

6. APPENDIX

6.1 Project proposal

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

Date of proposal: 30th September 2022
Project Title: ISS Project – My movies, intelligent movies recommendation system
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: 1.background and objetcives With the rapid development of economy, movies have entered people's lives and are loved and welcomed by the masses.We enjoy the fun of movie boost and suffer from picking one movie to watch at the same time, since all of us must be familiar with the scene： Endlessly scrolling through Netflix,watching trails on YOuTube,looking up IMDb ratings,wasting half an hour and still cannot decide what to watch. Although there are various video websites, most of them are piled up with popular information and lack of personalized recommendations. It is difficult for users, especially one suffers from decidophobia, to find movies that meet their personal interests and tastes. So this project takes this as an opportunity to integrate multiple recommendation algorithms to make recommendations in appropriate situations, and help users find their favorite movies in different situations in a simple manner.

2.movie recommendation systems development status

Now there are many big movie recommendation websites and software, such as Netflix, PickAMovieForMe, Douban, and so on. After analyzing their website we found that they have their own Characteristics.

In addition to knowing what a user is watching on Netflix, it also looks at things like how long the user watched, which device the user is watching Netflix on, and how long he watched. When the user enters a search query, the top results it returns are based on the actions of other members who have entered the same or similar queries.

PickAMovieForMe, a well-known movie recommendation website, uses a quiz-based movie picker to find movies that fit the users' moods, occasions and individual preferences.

As a famous movie evaluation social platform in China, Douban applies the item-based collaborative filtering algorithm to make personalized movie recommendations for users based on movie data and user viewing and review data.

3. commercial value and market research

The recommendation system is widely used in the Internet industry, the commercial vlue of recommend system is reflected in various aspects.

According to ByteDance's annual financial statements, as of 2020, tiktok used targeted advertising strategies to generate $202.1 billion in revenue in 4.6 years.

The usage of recommend system is reflected in various aspects. Ad targeting using the recommendation algorithm has undoubtedly played an important role in the gradual rise of TikTok’s advertising revenue.

Netflix's million-dollar Netflix Prize competition launched in 2009 brings recommendation systems into business. It attaches great attention in this technology by spending millions of dollars to analyse recommendation systems which help it saving more than 1 billion dollars each year.

What's more, not just used in movie selecting, the recommendation system is widely used in the Internet industry. According to ByteDance's annual financial statements, as of 2020, TikTok used targeted advertising strategies to generate $202.1 billion in revenue in 4.6 years.

With the help of the recommendation system, the precise placement of advertisements can be realized, and the advertisements can generate greater commercial value by matching the huge traffic of personalized recommendations and the precise interests of users. Similar to this, the e-commerce platform recommends products to the people who may buy them through personalized recommendation, promotes the purchase of the products, and obtains more shares from the merchants. Similar to how the role of a real estate agent can be realized through e-commerce.

To conclude, the commercial value of the recommendation system is reflected in the realization of value-added services, user growth, improving advertising efficiency, increasing exposure, reducing advertising costs, and saving labor costs.

Requirements Overview:

- Research ability

- Programming ability
- System integration ability
- Deep understanding in computer network\operation system

Resource Requirements (please list Hardware, Software and any other resources)

Hardware proposed for consideration:

- GPU, CPU,Router
- Application container, e.g. Docker
- Open source data
- Github
- Cloud Server(tencent)

Number of Learner Interns required: (Please specify their tasks if possible)

LU QINWEN: back-end development

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.

Team Formation & Registration

Team Name: MyMovies-0-Grioup 16
Project Title (repeated): MyMovies--movie recommendation system
System Name (if decided): MyMovies
Team Member 1 Name: LU QINWEN
Team Member 1 Matriculation Number: A0261847W
Team Member 1 Contact (Mobile/Email): e0983241@u.nus.edu
Team Member 2 Name: HU ZHIQING
Team Member 2 Matriculation Number: A0261637B

Team Member 2 Contact (Mobile/Email): e0983031@u.nus.edu
Team Member 3 Name: YU HANCHUN
Team Member 3 Matriculation Number: A0261716E
Team Member 3 Contact (Mobile/Email): e0983110@u.nus.edu
Team Member 4 Name: CAI ZIMO
Team Member 4 Matriculation Number: A0261628B
Team Member 4 Contact (Mobile/Email): zimocai06@u.nus.edu

6.2 Mapped system functionalities

Data presentation ——Knowledge Models in MR

Recommendation Algorithm—— Knowledge Discovery\Deductive Reasoning in MR

Micro services——Distributed Computing in MR

Redis cache——NoSQL Databases

CB recommendation——Analogical Reasoning in MR\Content-Based Filtering Example in RS

HTTP/TCP transportation——Data Structure: JSON in MR

Representation of relationship among users and how to trace(DFS\BFS)——Uninformed Search Techniques (Graph) in RS

Let system learn about the user data(in order to predict)——Machine Learning in RS

Market survey——Market Basket Analysis in RS

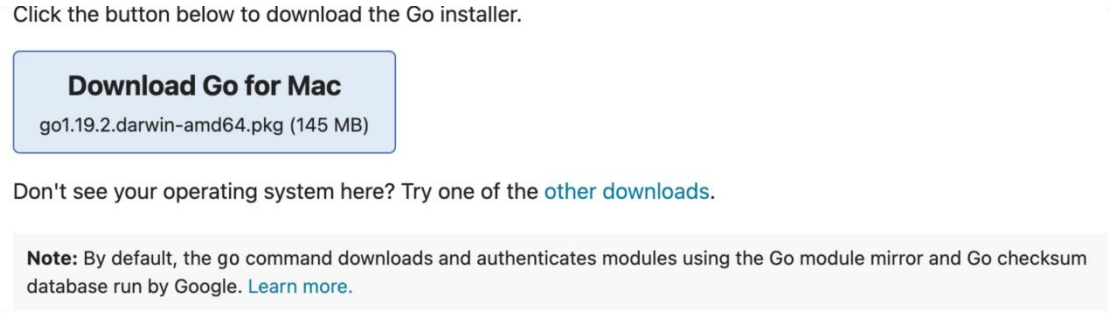
Prediction made by system——Concepts and techniques that enable systems to mimic human reasoning in CS

Computing similarity——Cosine Similarity in CS

6.3 Installation and user guide

6.3.1 Requirement

6.3.1.1 Go server configuration



(1) Open the package file you downloaded and follow the prompts to install Go.

(2) The package installs the Go distribution to /usr/local/go. The package should put the /usr/local/go/bin directory in your PATH environment variable. You may need to restart any open Terminal sessions for the change to take effect.

(3) Verify that you've installed Go by opening a command prompt and typing the following command:

(4) $ go version

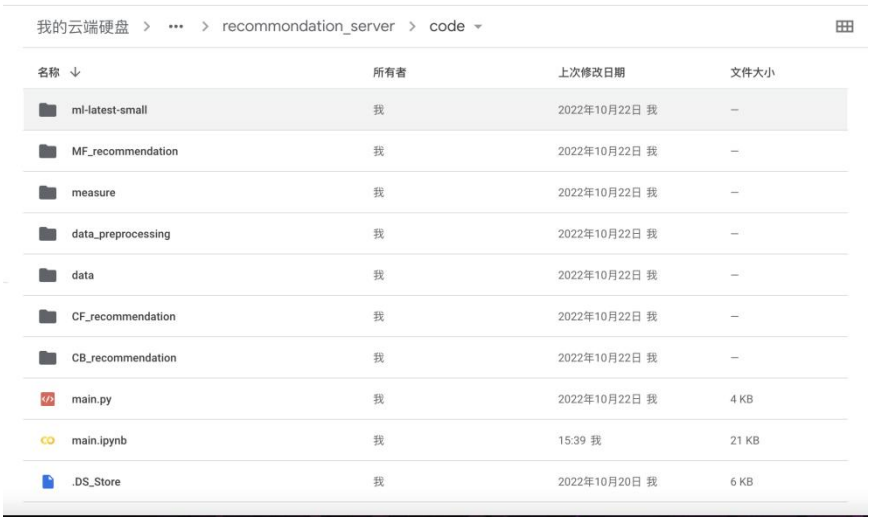
```
qinwenlu@QinwendeMacBook-Pro.local:~ $ go env
GO111MODULE="on"
GOARCH="arm64"
GOBIN=""
GOCACHE="/Users/qinwenlu/Library/Caches/go-build"
GOENV="/Users/qinwenlu/Library/Application Support/go/env"
GOEXE=""
GOEXPERIMENT=""
GOFLAGS=""
GOHOSTARCH="arm64"
GOHOSTOS="darwin"
GOINSECURE=""
GOMODCACHE="/Users/qinwenlu/go/pkg/mod"
GONOPROXY=""
GONOSUMDB=""
GOOS="darwin"
GOPATH="/Users/qinwenlu/go"
GOPRIVATE=""
GOPROXY="https://goproxy.cn,direct"
GOROOT="/opt/homebrew/Cellar/go/1.17.8/libexec"
GOSUMDB="sum.golang.org"
GOTMPDIR=""
GOTOOLDIR="/opt/homebrew/Cellar/go/1.17.8/libexec/pkg/tool/darwin_arm64"
GOVCS=""
GOVERSION="go1.17.8"
GCCGO="gccgo"
AR="ar"
CC="clang"
CXX="clang++"
CGO_ENABLED="1"
GOMOD="/dev/null"
CGO_CFLAGS="-g -O2"
CGO_CPPFLAGS=""
CGO_CXXFLAGS="-g -O2"
CGO_FFLAGS="-g -O2"
CGO_LDFLAGS="-g -O2"
PKG_CONFIG="pkg-config"
GOGCCFLAGS="-fPIC -arch arm64 -pthread -fno-caret-diagnostics -Qunused-arguments -fmessage-length=0 -fdebug-prefix-map
```

(5) Confirm that the command prints the installed version of Go.

6.3.1.2 Python server configuration

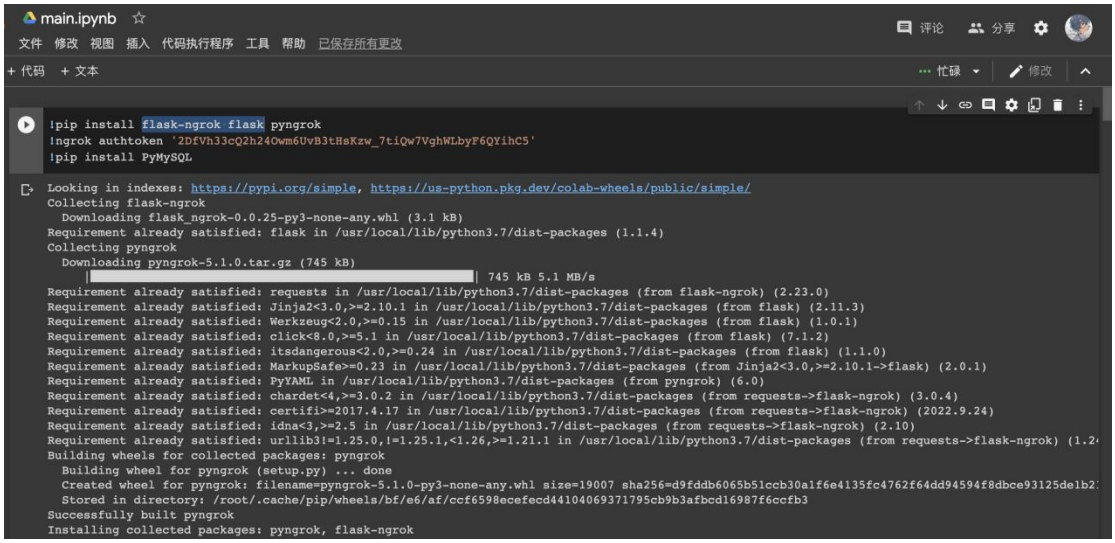
For simplifying the configuration process of our system, we used the Flask framework and deployed the python-based server for recommendation-related service in Google Colaboratory. The configuration steps are as follows:

(1) Register and sign in google account and upload the project files to the google drive.



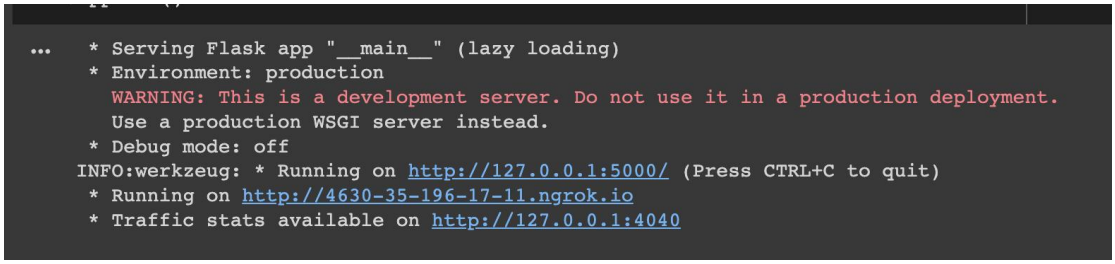
(2) Open main.ipynb file in google colab.

(3) Run the code in main.ipynb for installing the necessary libraries, like ngrok, pymysql and flask.



(4) Run the following code to import the model files in the project, load preprocessed data, configure the MySQL connecter and prepare the top K recommendation.

(5) Run the flask-related code block and activate the server.

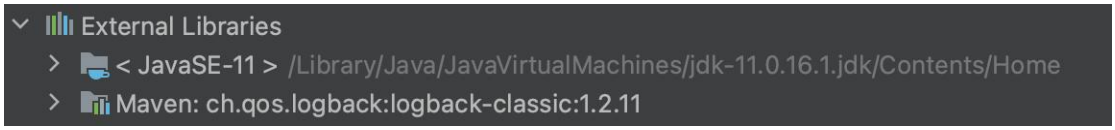


6.3.1.3 Java server configuration

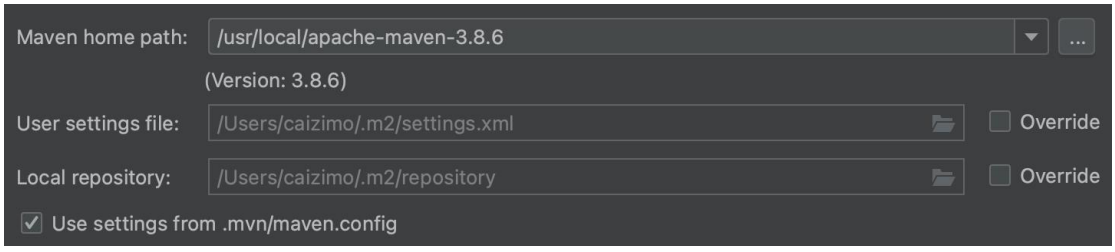
Install Java 11

Link:

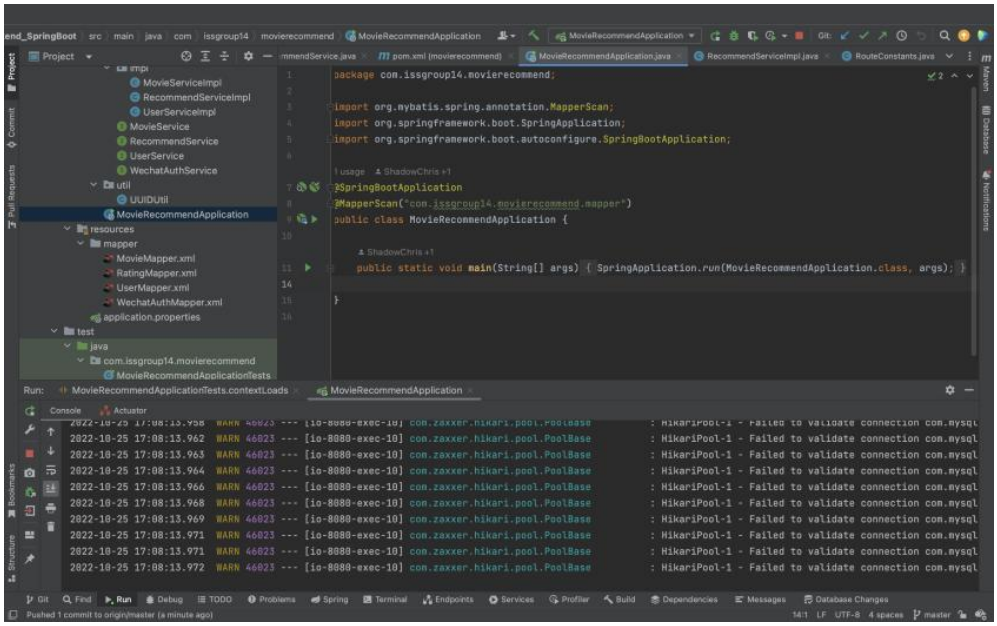
https://www.oracle.com/sg/java/technologies/javase/jdk11-archive-downloads.html



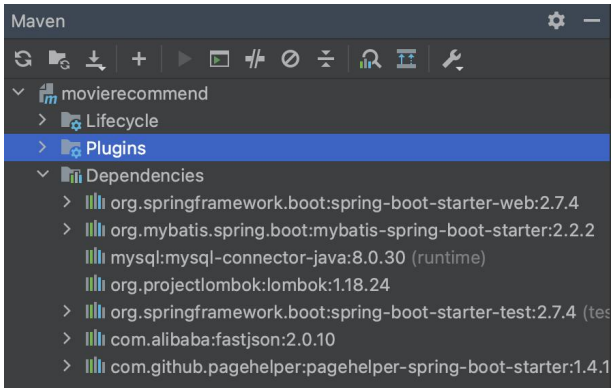
(1) Install Maven 3.8.6



(2) Import and run the project



(3) The dependencies are as follows:



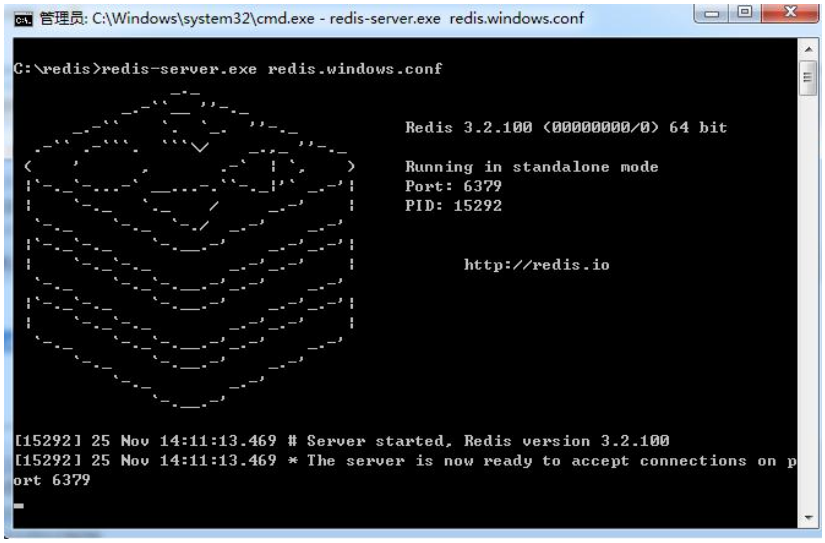
6.3.1.4 Redis configuration

Download address: https://github.com/tporadowski/redis/releases

Redis supports 32-bit and 64 bit. This needs to be selected according to the actual situation of your system platform. Here we download Redis-x64-xxx Zip the package to disk C. After decompression, rename the folder Redis.

Redis supports 32-bit and 64 bit. This needs to be selected according to the actual situation of your system platform. Here we download Redis-x64-xxx Zip the package to disk C. After decompression, rename the folder Redis.

Open a cmd window and use the cd command to switch the responding directory to run it:



At this time, another cmd window is opened, and the original one should not be closed, otherwise the server will not be accessible.

Switch to the Redis directory to run:

redis-cli.exe -h 127.0.0.1 -p 6379

6.3.1.5 Mysql Configuration

First, we use the yum command to install MySQL on Centos7. Note that the MySQL database in CentOS 7 has been removed from the default program list, so we need to download the Yum resource package on the official website before installing it. The download address is: https://dev.mysql.com/downloads/repo/yum/

wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm

rpm -ivh mysql-community-release-el7-5.noarch.rpm

yum update

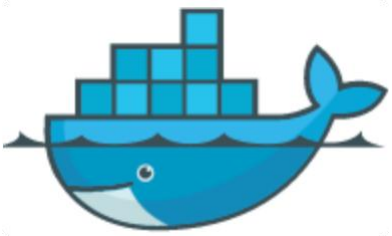
yum install mysql-server

systemctl start mysqld

You can also use Cloud MySql Server:

Note: If we start the MySQL service for the first time, the MySQL server will be initialized first.

6.3.1.6 Docker configuration



Install cmd :

curl -fsSL https://get.docker.com | bash -s docker --mirror Aliyun

Update apt package index:

$ sudo apt-get update

Install the apt dependency package to obtain the warehouse through HTTPS:

```
 sudo apt-get install \
     apt-transport-https \
     ca-certificates \
     curl \
```

gnupg-agent \
software-properties-common

Test whether the Docker is successfully installed. Enter the following instructions and print out the following information. The installation is successful:

$ sudo docker run hello-world

Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525:
Pull complete

Digest:
sha256:c3b4ada4687bbaa170745b3e4dd8ac3f194ca95b2d0518b417fb47e5879d9b5f
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:

https://hub.docker.com/

For more examples and ideas, visit:

https://docs.docker.com/get-started/

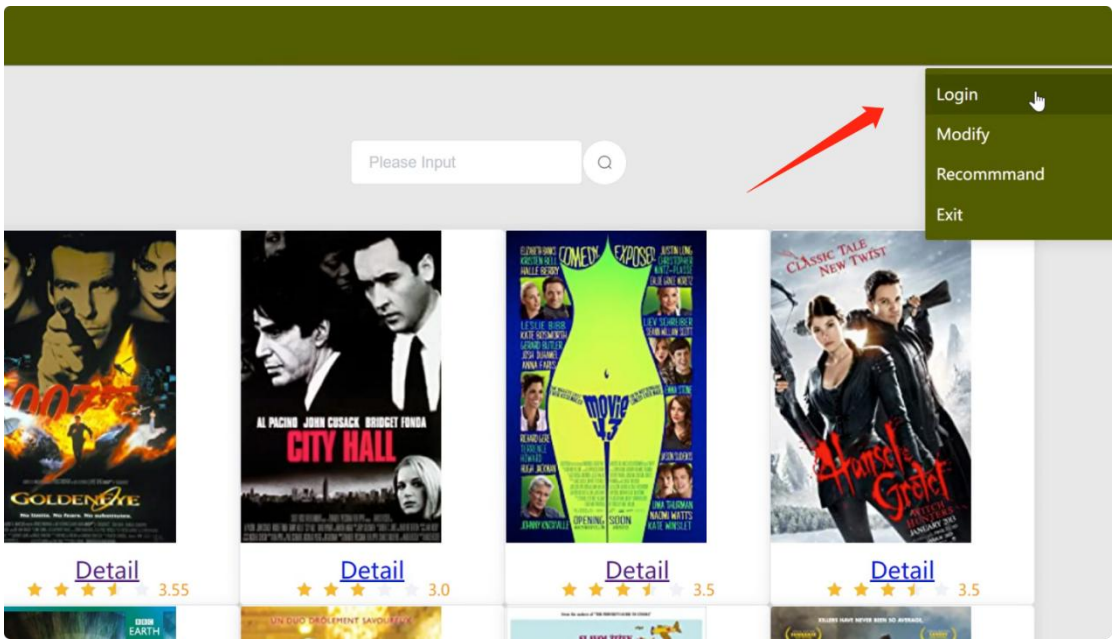
6.3.1.7 Browser requirement

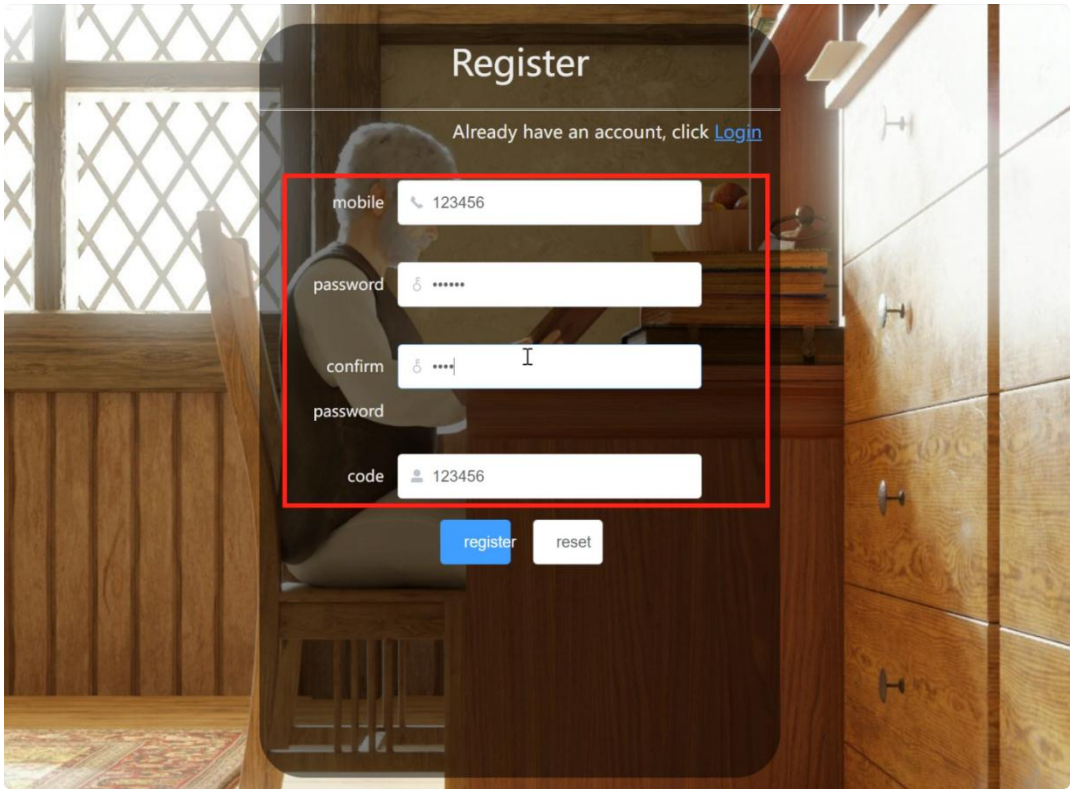
IE10、IE11，Edge、Firefox、Chrome、safari、opera

6.3.2 Best Practice

6.3.2.1 User information

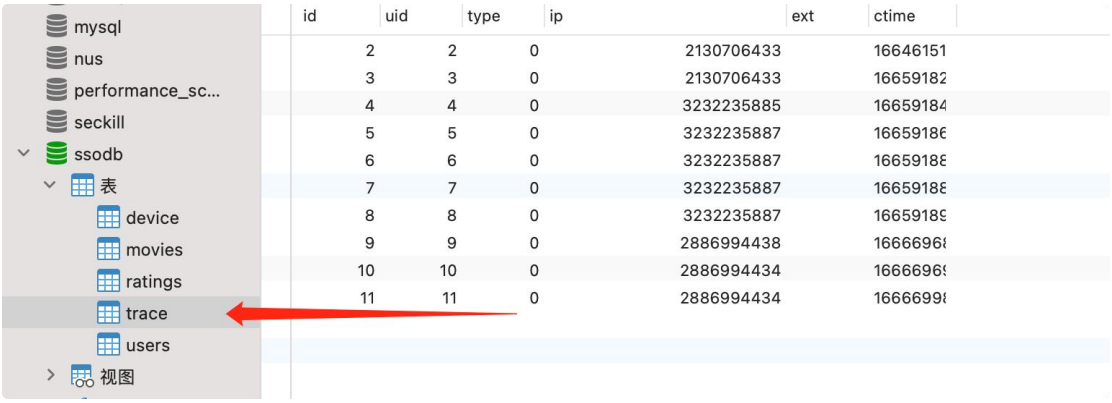
6.3.2.1.1 Login/Logout





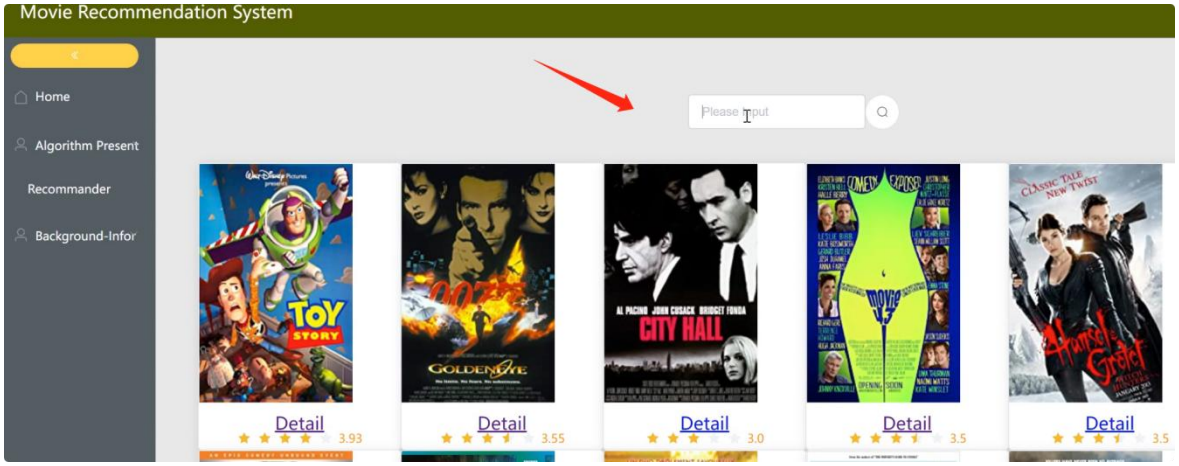
If users need to log in to obtain personalized recommendation results, new users need to complete the registration function before logging in for the first time.

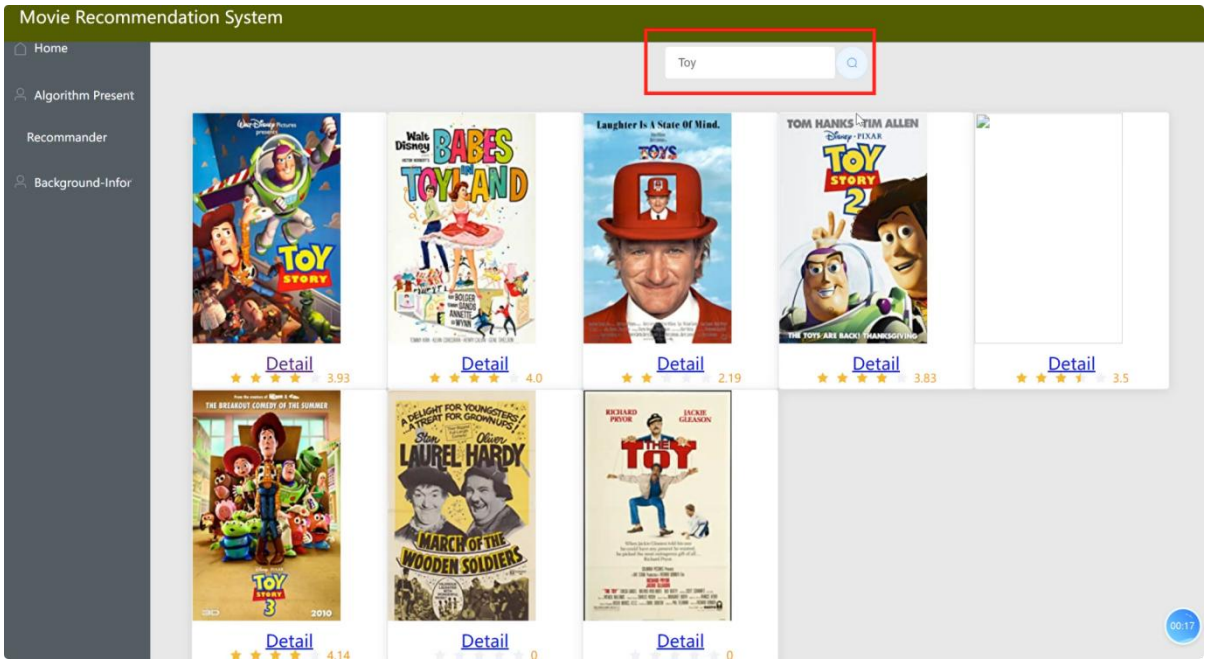
6.3.2.1.2 Trace user ID



6.3.2.2 Functions of system

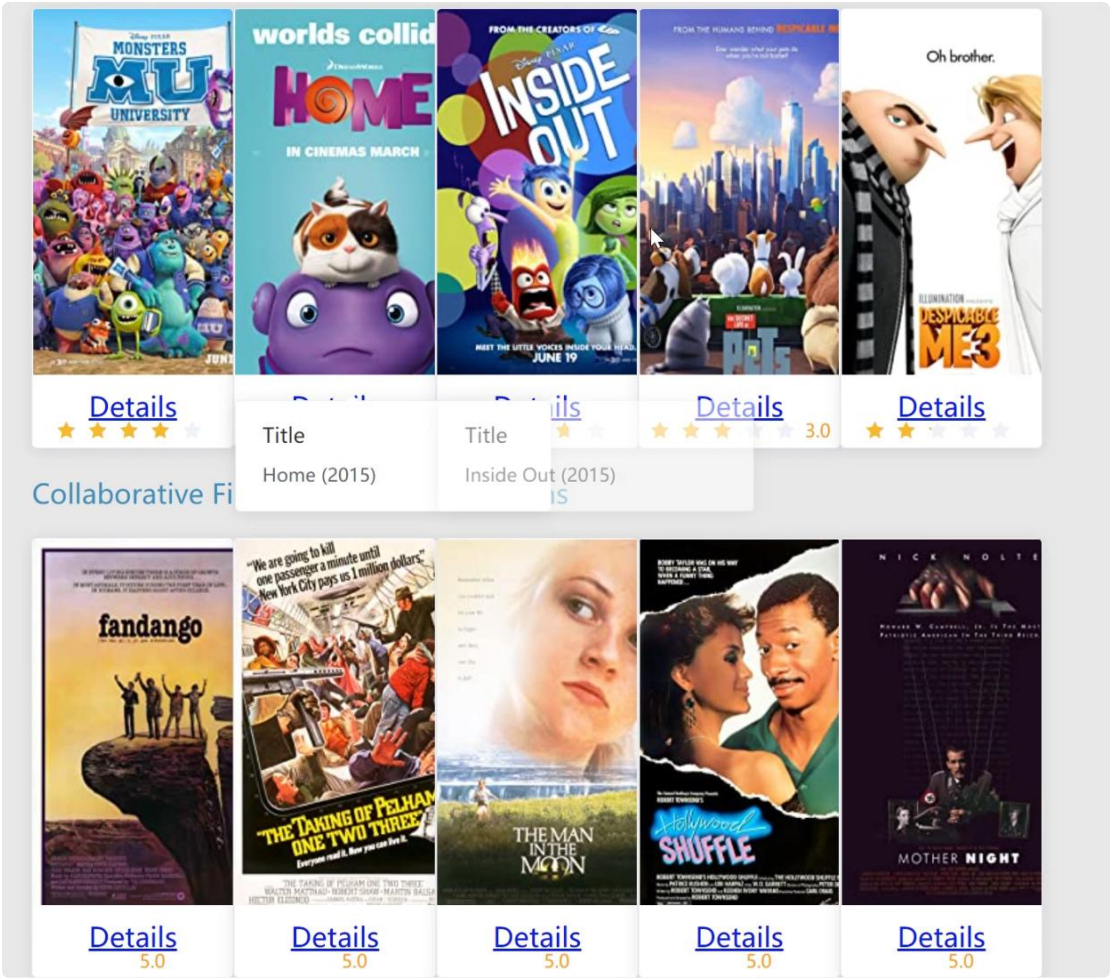
6.3.2.2.1Search movies





Enter the name of movie that you want to search in the input box to get relevant matching results.

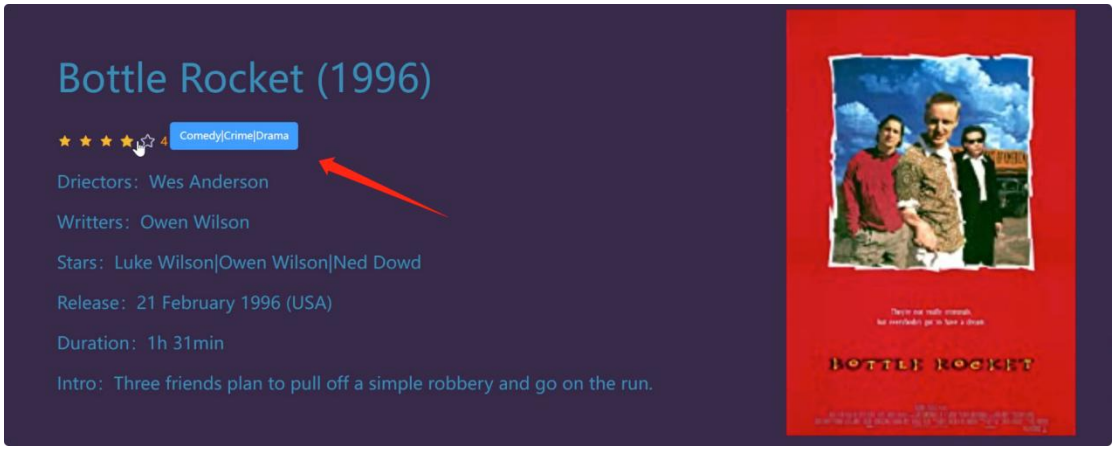
6.3.2.2.1Recommend movies



The recommendation system will select and recommend movies that may be of interest to users according to their personalized data.

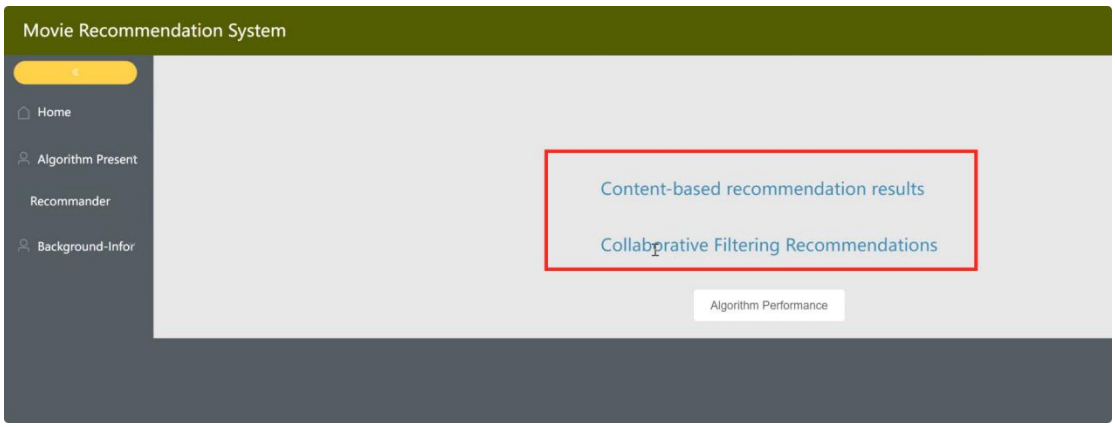
It is based on the recommendation algorithm. We also implement the high-performance TopK algorithm. The result will vary from person to person.

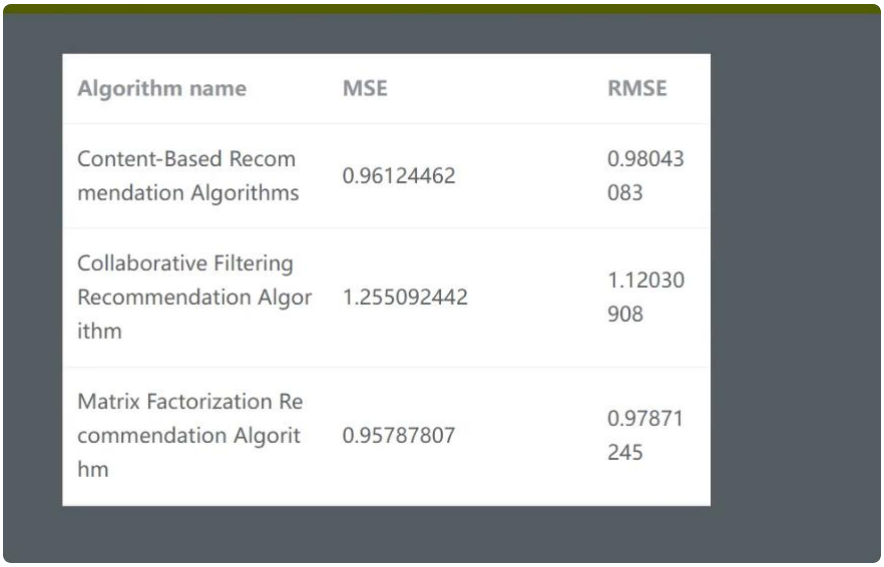
6.3.2.2.3 Rating a movie



The system will record the users' data based on user's feedback. On the movie details page, users can rate a movie by hovering and dragging the mouse ; After rating successfully, there will be corresponding prompt information

6.3.2.2.4 Performance of system





Algorithm name	MSE	RMSE
Content-Based Recommendation Algorithms	0.96124462	0.98043083
Collaborative Filtering Recommendation Algorithm	1.255092442	1.12030908
Matrix Factorization Recommendation Algorithm	0.95787807	0.97871245

We also provide the administrator with a visual page to view the performance of the current algorithm.

MSE （Mean Squared Error）

$$\frac{1}{m}\sum_{i=1}^{m}(y_i-\hat{y}_i)^2$$

RMSE（Root Mean Squard Error）

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i-\hat{y}_i)^2}$$

6.3.3 Interface document

Method	URL	parameters	remarks

POST	/signup/mobile	{
		"mobile":"1314520",
		"passwd":"123456",
		"code":"123456"
		}
			Sign up

POST	/login	{
		"mobile":"1314520",
		"passwd":"123456"
		}
			Log in

POST	/logout	clean token and cookie	Log out

POST	/login/mobile	{
		"mobile":"1314520",
		"passwd":"123456",
		"code":"123456"
		}
			Log in by mobile phone

POST	/rating/rateAMovie	{
		"movieId":"1314520",
		"rating":"5.0",
		"userId":"123456"
		}
			Rate a movie

POST	/rating/updateAMovieRating	{
		"movieId":"1314520",
		"rating":"3.0",
		"userId":"123456"
		}
			Change a rate for a movie

POST	/home/queryMovieList	{
		"pageNum": "1",
		"pageSize": "10"
		}
			Query movie list by page (sorted by movie ID by default)
			Parameters:
			Page
			Several data items on one page

POST	/home/queryMovieList	{
		"name": "Toy Story 2 (1999)",
		"genre": "Animation|Adventure|Comedy",
		"releaseTime": "24 November 1999 (USA)",
		"directors": "John Lasseter|Ash Brannon|1 more credit",
		"writers": "John Lasseter|Pete Docter",
		"stars": "Tom Hanks|Tim Allen|Joan Cusack",
		"pageNum": "1",
		"pageSize": "10"
		}
			Fuzzy search for a movie according to one of the keywords (movie name/type/release year/director/screenwriter/star), and return the matching results in pages

GET	/movie/{movie_id}	None	Query movie details according to movie ID
POST	/recommend/get-content-based-rec	{ "k": 10, "userId": 10,	Content-based personalized recommendation, returns k movies

		"pageNum": "1", "pageSize": "10" }	recommended by the system under the current user through paging

POST	/recommend/get-collaborative-filtering-rec	{
		"k": 10,
		"userId": 10,
		"pageNum": "1",
		"pageSize": "10"
		}
			Personalized recommendation based on collaborative filtering returns k movies recommended by the system under the current user through paging

GET	/recommend/preprocess	None	Update the preprocessing data of the recommendation system
GET	/recommend/getPerformance	None	Get the performance score (MSE, RMSE) of the two algorithms

Flask API Document

Method	URL	Parameters	Remark
GET	/preprocessing	None	Update the preprocessed data of the model

POST	/cb-recommend	{
		"userId":1,
		"k": 10
		}
			Get the top10 movie ID list returned by the user with ID 1 through the content based

			recommendation algorithm Return value: list of k movieIds

POST	/cf-recommend	{
		"userId":1,
		"k": 10
		}
			Get the top10 movie ID list returned by the user with ID 1 through the recommended algorithm of collaborative filtering
			Return value: list of k movieId

GET	/get-recommend-performance	None	[{ "MSE": 1.3207583528140168, "Name": "CB", "RMSE": 1.1492425126203853 }, { "MSE": 1.567236810564991, "Name": "CF", "RMSE": 1.251893290406571 }]

6.3.4 Necessary SQL

```
CREATE TABLE `movies` (
  `movieId` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  `title` varchar(255) DEFAULT NULL,
  `genres` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`movieId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `users` (
  `id` bigint unsigned NOT NULL AUTO_INCREMENT COMMENT '主键',
  `name` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL DEFAULT '' COMMENT '用
  `email` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL DEFAULT '' COMMENT '
  `mobile` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL DEFAULT '' COMMENT '
  `passwd` varchar(40) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '密码',
  `salt` char(4) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '盐值',
  `ext` text CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '扩展字段',
  `status` tinyint NOT NULL DEFAULT '0' COMMENT '状态（0：未审核,1:通过 10删除）',
  `ctime` int unsigned NOT NULL DEFAULT '0' COMMENT '创建时间',
  `mtime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '修改时间',
  PRIMARY KEY (`id`),
  KEY `ctime` (`ctime`)
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `trace` (
  `id` bigint unsigned NOT NULL AUTO_INCREMENT COMMENT '主键',
  `uid` bigint unsigned NOT NULL DEFAULT '0' COMMENT '用户主键',
  `type` tinyint NOT NULL DEFAULT '0' COMMENT '类型(0:注册1::登录2:退出3:修改4:删除)',
  `ip` int unsigned NOT NULL COMMENT 'ip',
  `ext` varchar(1000) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '扩展字段',
  `ctime` int unsigned NOT NULL DEFAULT '0' COMMENT '注册时间',
  PRIMARY KEY (`id`),
  KEY `UT` (`uid`,`type`) USING BTREE
) ENGINE=MyISAM AUTO_INCREMENT=12 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `ratings` (
  `user_id` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  `movie_id` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  `rating` varchar(255) DEFAULT NULL,
  `timestamp` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`user_id`,`movie_id`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

6.4 Individual project report

6.4.1 Cai Zimo A0261628B

(1) Personal Contribution

In this project, I am mainly responsible for data collection, front-end web page development, and back-end server development.

For data collection, I used MovieLens small latest as our basic dataset and used a web crawler to capture more detailed information about all movies based on the dataset. Finally, I imported the file data into MySQL database, so as to facilitate the display of movie information in this system.

As for web page development, I participated in the development of the front-end system based on Vue, adjusted the layout of the home page and beautified the interface, and was responsible for connection with the back-end interface.

In the back-end development, I used Spring Boot, a Java-based web framework, as one of our servers and developed movie-related services. In this server, I mainly realized the functions that processed requests from the front end, obtained data from the database and Flask server and responded to the front end after processing. The main functions include: inserting and modifying movie information, displaying movie lists, searching movies, displaying movie details, obtaining personalized recommended movie lists, etc.

At the same time, I deployed the recommendation algorithm model on Google Colab, successfully ran the algorithm model on Google Colab using Flask and Ngrok, and exposed the interface to the network, so that the Spring Boot server can successfully interact with the Flask server and transmit data.

(2) Useful knowledge & Application

In this group project, I learned a lot of practical knowledge and generated many good ideas for application in other scenarios.

The first is about the algorithm. Through this project, I have systematically studied the theories and developed recommendation models in practical terms, such as content-based recommendation algorithms and collaborative filtering recommendation algorithms. Through theoretical study and actual code implementation, I have a deeper understanding of the underlying principles of the recommendation algorithm and learned how to apply the algorithm model to the actual project system. Through the knowledge of the recommendation system I learned, I realized that it is not only movie recommendation, but can be applied to all fields containing specific content production, such as news, video, merchandise, music, and other fields. We can infer users' favorite preferences based on user's operation behavior (such as browsing, clicks, and screen time) and content category characteristics (such as sports news and international news, jazz music, and pop music), so as to recommend users more content of interest.

Then, I understood and learned some practical technical software and tools. For example, I learned how to deploy a Python-based algorithm model to Google Drive and successfully import the project and run it using Colab. Google Colab is equipped with many machine learning-related configurations and provides each user with part of the computer resources required to run the model so that users can save the time to deploy the environment and the cost of purchasing GPUs. Meanwhile, I learned how to use Ngrok. Ngrok is a globally distributed reverse proxy that allows us to expose

our local server interface to the Internet. As for applications, we can apply Google Colab and Ngrok to various fields of backend development, for example we can use Google Colab as A lightweight server that performs high-performance computing and can be quickly developed to form demos in business development in various fields.

Finally, at the database level, I encountered and solved some problems caused by the design of business functions during development, such as the movie paging query function of this project. In past development experience, I adopted this strategy to solve the paging problem: first, use SQL statements in the DAO layer to query the all data of the movies to the Service layer, and use the PageHelper plugin in the Service layer to perform data paging. But in this system, it takes nearly 6 seconds for the system to execute each query function. This means that if a user uses our system, every time they click the next page, they need to wait 6 seconds. Obviously, this will greatly mess with the user experience. After analyzing the principle of the strategy, I found the problem: in the previous development experience, our system has less database information, so the query efficiency of this strategy is good. However, there are about 10,000 lines of data in the movies table of this project. When the amount of information in the database table is very large, some common query strategies may cause the system to run very slowly. Therefore, we changed the strategy to add keys LIMIT and OFFSET to the SQL query statement and perform a paging query from the DAO layer. When using this strategy, the system queries the database for the relevant page of data, rather than querying the entire database table. After using this method, the function time was reduced from 6 seconds to 0.2 seconds, and the efficiency was increased by 30 times! In the end, we successfully optimized this function, making the performance of the whole system superior.

6.4.2 Lu Qinwen A0261847W

(1) personal contribution to group project ?

- Dispatch the tasks to my teammates fluently. Get the whole team to have a cordial working relationship.
- Build a high-performance Go back-end server.
- Design the architecture of the system, using the idea of micro services.
- Search and study the recommendation algorithm(CB\CL), get them known by my teammates.

(2) what learnt is most useful for you ?

In this project, I have a better understanding in back-end development:

- I have a deeper understanding in build a movie recommendation system.
- I get to know the some useful algorithm in recommendation.
- I have a deeper understanding in GO and Gin.
- I know how to write a high-performance sorting algorithm.
- I can optimize the slow SQL.

(3) how you can apply the knowledge and skills in other situations or your

workplaces?

I will take the TopK sorting algorithm as an example:

Before this project, I have little idea of why I must learn the data structure and algorithm. But this time, when I have to handle millions of data, every detail becomes important.

In this system, I use a heap. Heap is a data structure that has several very important applications: priority queue, top K, and median.

How to implement a priority queue? There are many methods, but the heap is the most direct and efficient. This is because the heap and priority queue are very similar. A heap can be regarded as a priority queue. Many times, they are just conceptual distinctions. Inserting an element into the priority queue is equivalent to inserting an element into the heap; Taking the element with the highest priority from the priority queue is equivalent to taking the top element of the heap.

Therefore, we can always maintain a K size small top heap. When data is added to the collection, we compare it with the elements at the top of the heap. If it is larger than the top element, we will delete the top element and insert it into the heap; If it is smaller than the top element, it will not be processed. In this way, whenever we need to query the current top K big data, we can immediately return it.

So now, when I meet a similar situation again, I have a firm belief that I can solve it smoothly.

6.4.3 Hu Zhiqing A0261637B

(1) personal contribution to group project ?

- Learn and research content-based recommendation algorithms and collaborative filtering algorithms and perform model evaluation
- Conduct market research and business analysis and project report
- Assist teammates to research and develop python Flask server

(2) what learnt is most useful for you ?

In this project, I have a better understanding of recommendation algorithm

- I have a wider and deeper understanding of different kinds of recommendation algorithms, and implemented collaborative filtering algorithm and content-based recommendation algorithm
- I understand how the recommendation process from data to model to prediction results is completed through the front and back ends, and have a clearer understanding of designing a complete system architecture
- I understand and learn how to deal with the cold start problem
- I have new thinking on how to use big data platform for system optimization

(3) how you can apply the knowledge and skills in other situations or your workplaces?

The inspiration for this project for me mainly lies in two aspects: how to complete the design and implementation of the recommendation algorithm and the

architectural design ideas for realizing the whole system.

Not only did I have a more comprehensive understanding of the whole of front-end page visualization, back-end business processing, database data support, environment installation, and deployment, but I also deeply felt that the power of teamwork is more important than the sum of individuals. The most important thing is that the project has stimulated my interest in the development and design of big data recommendation systems. I plan to use the knowledge of recommendation algorithms I have learned to continue to use a hybrid method of CF and CB algorithms to continue to improve the project: system initialization, Offline recommendation, real-time recommendation, and business realization are triggered by four aspects. Some components of Spark's ecological services are used. MongoDB manages the business database in the platform and serves as the main database. At the same time, I hope that models such as means and deep learning can be combined for recommendation in future learning research, so as to improve the accuracy and diversity of recommendations.

I think this will give me a huge boost in my algorithmic and system design abilities.

6.4.4Yu Hanchun A0261716E

1.Personal Contribution

In the project, I was mainly involved in front-end design as well as video production and editing. In the front-end design, HTML CSS JavaScript is used to realize part of the interface design and functions, and vue.js is used to integrate it into the web to complete the docking of the back-end interface to realize the function.

In the closing work of the project, the production of the report video was completed, the materials in the work were summarized, and the video was dubbed to better present the overall work situation of the group.

2.Useful knowledge & Application

In the project, I learned the syntax of HTML CSS, and JavaScript, and learned to write basic front-end statements. At the same time, I further learned the Vue.js framework. Vue.js is a progressive framework for building user interfaces. Unlike other heavyweight frameworks, It adopts a bottom-up incremental development design. Vue's core library only focuses on the view layer and is very easy to learn and integrate with other libraries or existing projects. On the other hand, Vue is fully capable of driving complex single-page applications developed with single-file components and libraries supported by the Vue ecosystem. The goal of Vue.js is to enable reactive data binding and composed view components with the simplest possible API. Vue.js itself is not a catch-all framework - it only focuses on the view layer. So it is very easy to learn and very easy to integrate with other libraries or existing projects. On the other hand, Vue.js can also drive complex single-page applications when used with related tools and supporting libraries. Therefore, using this framework, the front-end code can be integrated in an orderly manner to further build a complete front-end interface. This plays a vital role in our project packaging.

How to get the data of the back-end in the front-end, this time needs Axios. It is an interactive way of Web data, which can convert request and response data, cancel requests, automatically convert JSON data, etc., and support Promise API.

Through this group project, I have a deep understanding of the standard industrial process of developing a product, the realization, and coordination of requirements in-group cooperation, the normative requirements of code, the writing of document reports, work reports, etc., all for the future. laid the foundation for a business career.