COMP9417

Machine Learning and Data Mining

Assignment2

Topic 3.4
Recommender system using collaborative filtering

Group member:	Yuting Cao	z5105048
	Tianrun Zhang	z5103165
	Jie Wang	z5119770
	Qin Huang	z5124183

Introduce

The explosive growth in the amount of available digital information and the number of visitors to the Internet have created a potential challenge of information overload which hinders timely access to items of interest on the Internet This has increased the demand for recommender systems more than ever before. Recommender systems are information filtering systems that deal with the problem of information overload by filtering vital information fragment out of large amount of dynamically generated information according to user’s preferences, interest, or observed behavior about item. Recommender system has the ability to predict whether a particular user would prefer an item or not based on the user’s profile.

The task of the recommendation system is to solve the problem of poor screening effect of search engine when users cannot accurately describe their requirements. Contact the user and information, on the one hand, help users to find valuable information on their own, on the other hand allow information to show he is interested in the crowd, so as to realize the win-win of information providers and users.

In our project, we will design a collaborative filtering recommender system by using given data of this assignment.

1.1 Several simple methods

There are several methods to find similarity users:

1.1.1 Pearson's correlation coefficient

Pearson’s correlation coefficient is a method to avoid grade inflation.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Example:

	Blues traveller	Norah jones	phoenix	The strokes	Weird AI
Clara	4.75	4.5	5	4.25	4
Robert	4	3	5	2	1

This phenomenon is called "fractional inflation" in data mining. Clara gives a minimum of 4, all of her grades range from 4 to 5 and Robert give the grade from 1 to 5. But these two people’s grade has the same trend and they have the same preference. Pearson's correlation coefficient is used to measure the correlation between two variables. Its value is between -1 and 1. So we use this to find similar users.

So in this example ,we can calculate :

Average of x = (4.75 + 4.5 + 5 + 4.25 + 4) /5 = 4.5

Average of y = (1+ 2 + 3+ 4 + 5)/5 = 3

$$\sqrt{\sum_{i=1}^{n}(x_i - x)^2} = \sqrt{(4.75-4.5)^2 + (4.5-4.5)^2 + (5-4.5)^2 + (4.25-4.5)^2 + (4-4.5)^2}$$

$$= \sqrt{0.625}$$

$$\sqrt{\sum_{i=1}^{n}(y_i - y)^2} = \sqrt{(1-3)^2 + (2-3)^2 + (3-3)^2 + (4-3)^2 + (5-3)^2}$$

$$= \sqrt{10}$$

$$r = \frac{(4.75-4.5)(1-3)+(4.5-4.5)(2-3)+(5-4.5)(3-3)+(4.25-4.5)(4-3)+(4-4.5)(5-3)}{\sqrt{0.625}*\sqrt{10}}$$

$$= 0.75$$

1.1.2 Cosine Similarity

This method is suitable for the huge data and the single user’s data is sparse because the data is very tiny compared with the whole data. For example ,a app has more than 15 million songs and the user only listens around 1000 songs ,it’s impossible to calculate the distance, so there is the cosine similarity method:

$$\cos(x,y) = \frac{x \cdot y}{||x|| \times ||y||}$$

In this formula, x and y represent vectors,so there is the example:

	Blues traveler	Norah jones	phoenix	The strokes	Weird AI
clara	4.75	4.5	5	4.25	4
robert	4	3	5	2	1

So $x= (4.75, 4.5, 5, 4.25, 4)$

$y= (4, 3, 5, 2, 1)$

$$||x|| = \sqrt{4.75^2 + 4.5^2 + 5^2 + 4.25^2 + 4^2}$$

$$= 10.09$$

$$||y|| = \sqrt{4^2 + 3^2 + 5^2 + 2^2 + 1^2}$$

$$= 7.416$$

$$\text{Cos}(x, y) = \frac{(4.75*4+4.5*3+5*5+4.25*2+4*1)}{10.09*7.416}$$

$$= 0.935$$

The result is between -1 and 1 and the 0.935 is very close to 1 so there are high match rate of these two users.

1.1.3 Top-k Method

There is another problem that if these two users has some same browsing history or purchase history, but everyone has special interest ,so if use the methods given before , another user may get the recommendation he does not interest. So Top-k method can get the more correct recommendation.

First we can use the Pearson's correlation coefficient to get all the users' relation ,then we first n large result .After that we calculate the weight of each result , this is the influence rate of each user's data .

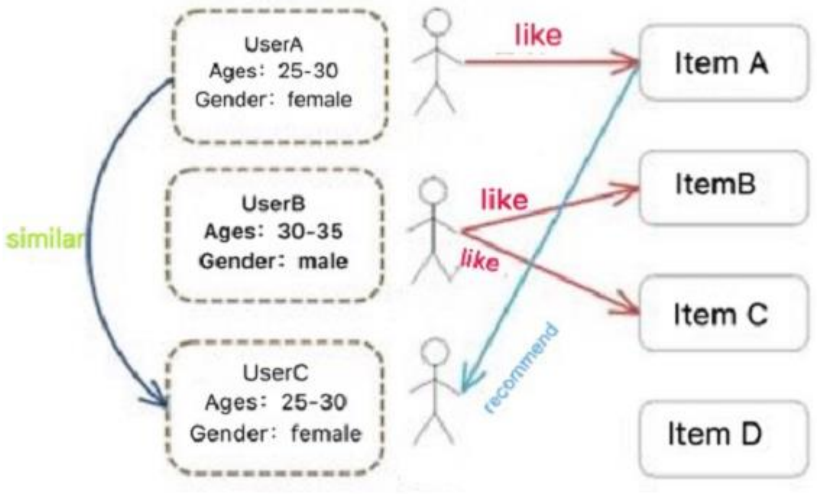
1.2 Recommended System

1.2.1 Emergence of recommended systems

The task of the recommendation system is to solve the problem of poor search results for search engines when users cannot accurately describe their own needs. Contacting users and information helps the user find information that is valuable to him, on the other hand it allows the information to be displayed in the crowd that is interested in this user, so as to achieve a win-win situation between the information provider and the user.

1.2.2 Demographic based recommendation

This is the simplest kind of recommendation algorithm. It simply finds the relevance of the user based on the basic information of other users, and then recommends the other user's favorite items to the current user.



First of all, the system build a model for each user, including the user's basic information, such as the user's age, gender, etc.; then, the system will calculate the user's similarity based on the user's profile, It is obviously to see that the user A's profile is the same as user C. Then the system considers users A and C to be similar users. In the recommendation engine, they can be called "neighbors." Finally, based on the preferences of "neighbor" user groups, they recommend some items to the current user.

Advantages:

1.There is no “Cold Start” problem for new users because the data of current user preferences for items are not used.

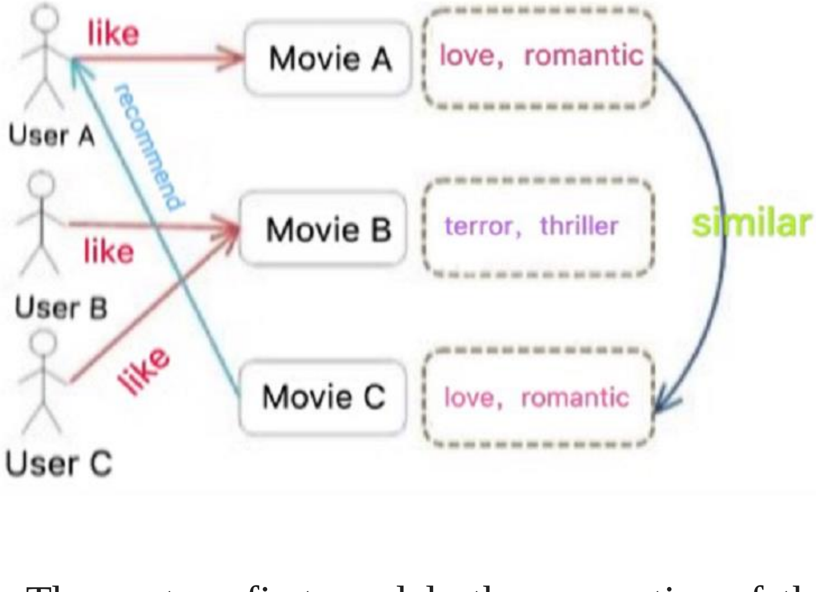
2.This method does not depend on the data of the item itself, so this method can be used in different fields of items. It is domain-independent.

Disadvantages:

This method of classifying users based on user's basic information is too rough, especially in areas where taste requirements are high, such as books, movies, and music, and cannot be well recommended. Another limitation is that this method may involve information that is irrelevant to the information which is helpful for discovering problem itself, such as the user's age, etc. These user information is not well-acquired.

1.2.3 Content based recommendation

Similar to the above method, but this time the center has switched to the item itself. Use the similarity of the item itself instead of the user's similarity.



The system first models the properties of the item (the movie example in the picture), using the type as an attribute. In practical applications, it is obviously too rough based on the type. The Actors, directors, and more properties are also need to be considered. Through similarity calculations, it was found that movies A and C have higher similarities because they all belong to the love class. The system also finds that user A likes movie A, and concludes that user A is also likely to be interested in movie C as well, then recommend movie C to user A.

Advantages:

Easy to implement, no user data is required so there are no sparsity and cold start issues.

Based on the item's own feature recommendation, there is no over-recommended topic.

Disadvantages:

The properties of the items are limited and it is difficult to get more data effectively.

The measure of similarity of items only considers the item itself and has a certain degree of one-sidedness.

Need historical data of the user's items, there is a cold start problem.

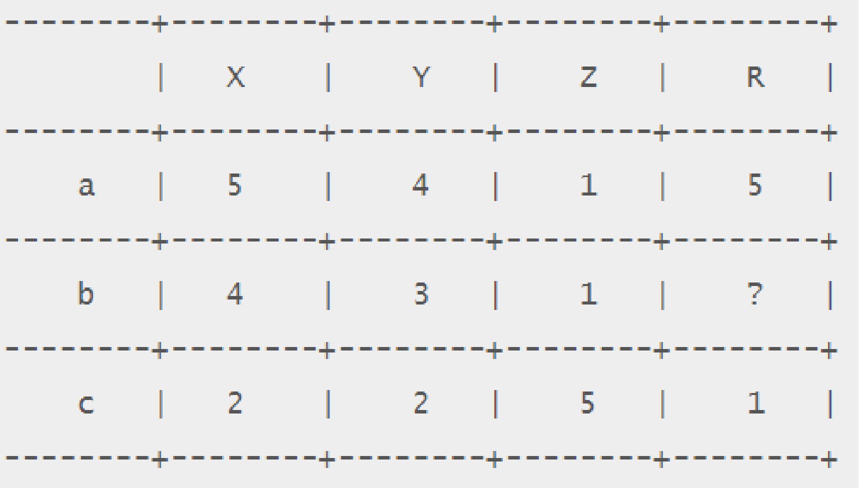
Collaborative Filtering

Collaborative filtering, also referred to as social filtering, filters information by using the recommendations of other people. It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future. A person who wants to see a movie for example, might ask for recommendations from friends. The recommendations of some friends who have similar interests are trusted more than recommendations from others. This information is used in the decision on which movie to see. There are two kinds of collaborative filtering algorithms. One is item-based, and the other is user-based.

2.1 User-based collaborative filtering

So simply to say, the collaborative filtering algorithm is that I would recommend to you the item which is similar to your favorite, the same calculation of user relevance when we are by comparing their relevance to the same item score to calculate.

For example, the table represent the score of each movie(X,Y,Z,R) given by the 3 users a, b and c.



```
--------+--------+--------+--------+--------+
        |   X    |   Y    |   Z    |   R    |
--------+--------+--------+--------+--------+
    a   |   5    |   4    |   1    |   5    |
--------+--------+--------+--------+--------+
    b   |   4    |   3    |   1    |   ?    |
--------+--------+--------+--------+--------+
    c   |   2    |   2    |   5    |   1    |
--------+--------+--------+--------+--------+
```

A user made 5 points for the movie X , movie Y scored 4 points, movie Z gained 1 point, same reasoning as user b and user c , then it is easy to see user a and user b are very similar, but user b have not seen Item R , then we can recommend a very high item R to user a who is very similar to user b. This is the user-based collaborative filtering.

Returning to our collaborative filtering algorithm, we now know that user-based collaborative filtering needs to compare the user's relevance, then how to calculate this correlation, so we can use two users to calculate the relevance of the same item score. For user a and user b, they all evaluate XYZ items. Then, user a can be expressed as (5,4,1) and user b can be expressed as (4,3,1). The classic algorithm is to calculate They are regarded as two vectors, and calculate the angle between two vectors, then calculate the cosine value of the vectors to compare, so the correlation between a and b is:

$$\frac{5*4+4*3+1*1}{\sqrt{5^2+4^2+1^2}*\sqrt{4^2+3^2+1^2}}$$

This value is between -1 and 1, the greater represent, the two users are more relevant. Here it seems that cosine is still good, but consider such a problem, for the differences between users, user d may like to play high scores, user e like to play low scores, user f like to random scores.

Obviously, user d and user e have the same tendency to evaluate the work, so user d and user e should be considered to be more similar, but cosine can only calculate that d and f are more similar. Then we can use the formula of Pearson correlation coefficient to compute the similarity which performs better than cosine similarity.

We use the item-based collaborative filtering ,so the detailed method we talk in the next part.

Implementation And Experimentation

Our system uses Item-based collaborative filtering.

This system has main two steps:

1) Find the relation rate among all the movies depends on the grade users give using the Pearson's correlation coefficient.

2) Our recommendation system get a input user id and find all the movies he gave grade and all the movies he didn't ,then according to the grades he gave calculate each movie's grade which he didn't give the grade using Pearson's correlation coefficient.

3.1 Item-based collaborative filtering

For user-based collaborative recommendation algorithm, with the number of users increases, the calculation time will become longer.

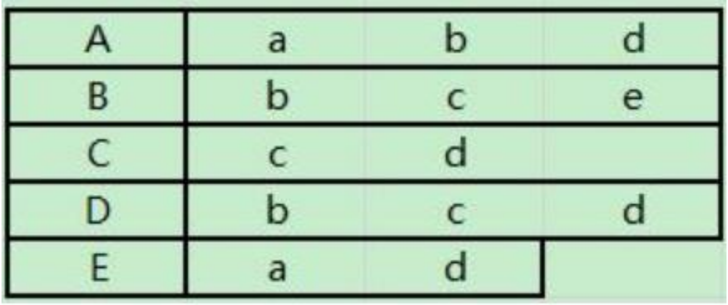
So in 2001, Sawar proposed a collaborative filtering recommendation algorithm based on items. Let |N(i)| denote the number of users who like i , and |N(i) ∩ N(j)| denote the number of users who like item i at the same time also like item j, then the similarity of item i and item j will be :

$W_{ij} = |N(i) \cap N(j)| / |N(i)|$ (1)

There is a problem with the formula (1). When the item j is a very popular product, everyone likes it, then the Wij will be very close to 1, that is, formula (1) will make a big similarity between a lot of items and hot products, so the improved the formula is:

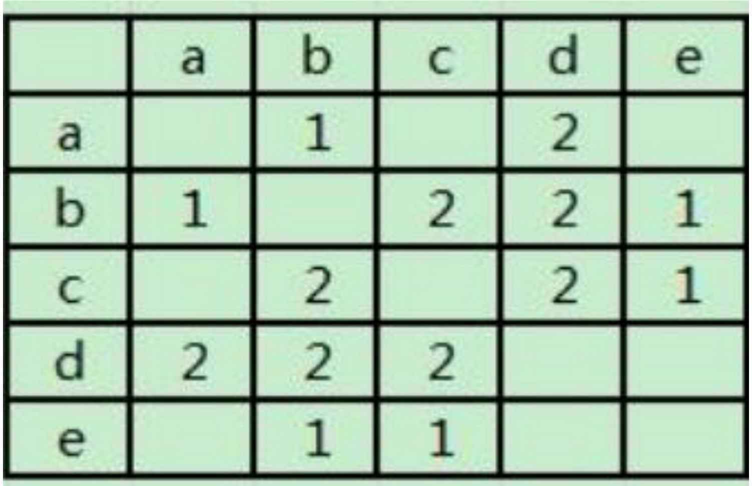
$W_{ij} = |N(i) \cap N(j)| / \mathrm{sqrt}(|N(i)| * |N(j)|)$ (2)

First of all, it is necessary to create a user item inverted list. The uppercase letter is used to indicate the user, and the lowercase letter indicates the item. The built-in inverted list of user items is shown below:



A	a	b	d
B	b	c	e
C	c	d	
D	b	c	d
E	a	d	

The co-occurrence matrix C represents the number of users who like two items at the same time and is calculated based on the user's item countdown table. For example, the following co-occurrence matrix C can be calculated based on the above user item posting table:



	a	b	c	d	e
a		1		2	
b	1		2	2	1
c		2		2	1
d	2	2	2		
e		1	1		

The item that is ultimately recommended is determined by the degree of interest in the forecast. Item j prediction interest degree =

interest degree of the user's favorite item i × similarity of the item i and the item j.

Finally, we need to normalizing the interest degree(rating) of this users.

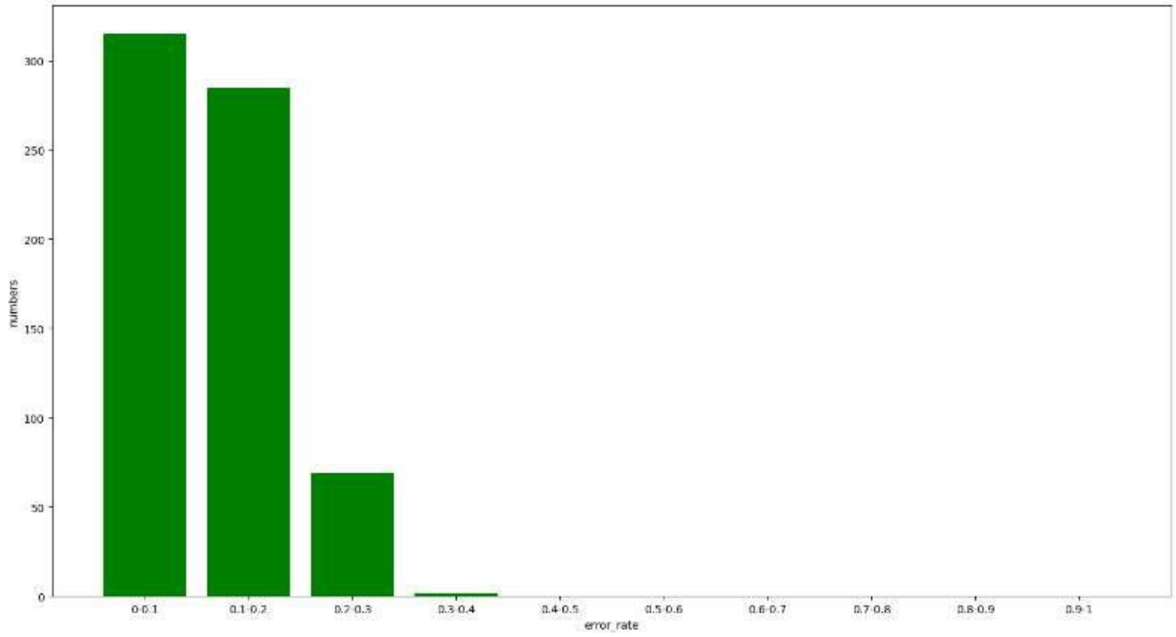
For example, if a user likes items a, b, and c, their interest degrees are 1, 2, and 2. The predicted interest of items d and e are:

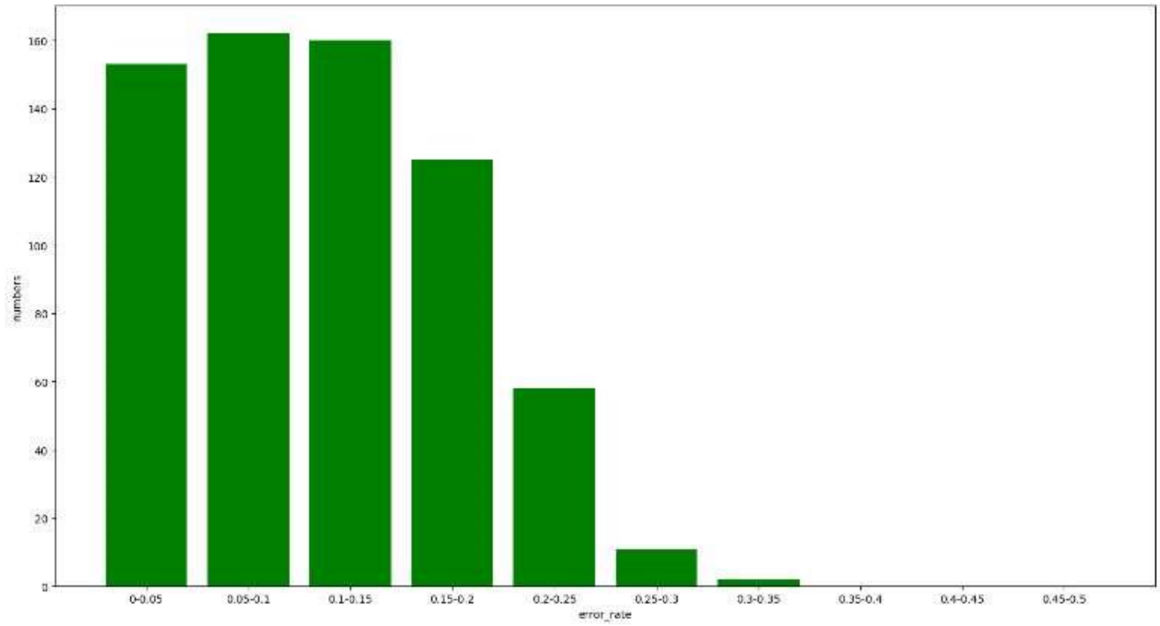
d：(1×0.71+2×0.58+2×0.58)/(0.71+0.58+0.58)=1.620

e：(2×0.58+2×0.58)/(0.58+0.58)=2

So the recommendation system should recommend item e to users.

Results and Analysis





The advantage of item-item system:

Low error rate

In this system, the system needs to give the grade that matches the grade trend that users really give. For example ,if the user gives movie A 3 mark and gives movie B 5mark.Then the recommendation system ‘s grade of these two movies must has the same trend that the grade of movie A is less than movie B.

If not ,this is the error ,so we test thousands of users and calculate the error rate ,and the highest error rate is 0.31 that means the mean correction rate is higher than 87 %.

Achievement

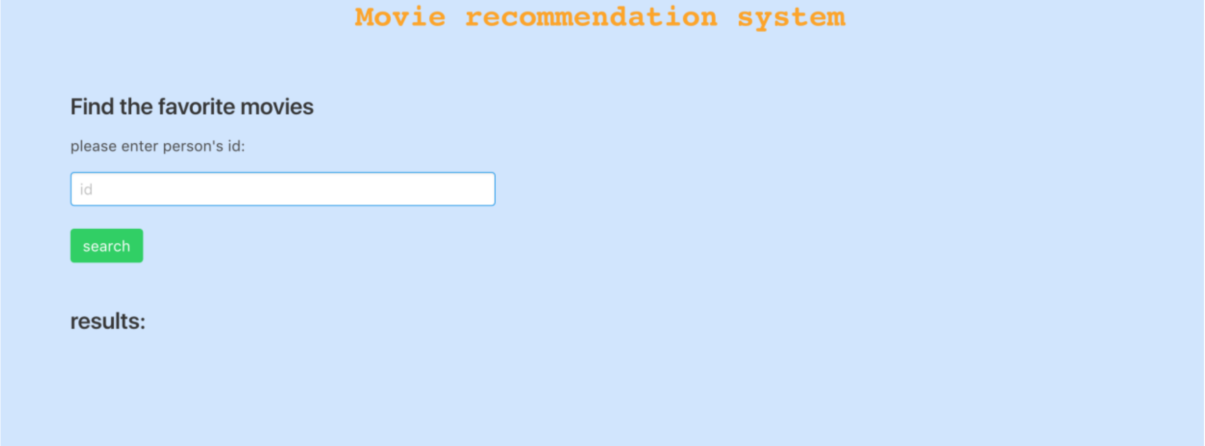
To make the recommendation system more practical and visualized. We use python to write a front page for users to search on it and also we give details of the movies we recommend.

First, we run main.py to set up the movie recommendation system, then double click recommendation.html to enter movie recommendation system page. In this page, we can input the person’s id and click the search button. Then the search results, movie pictures and movie links are showed below. Click the links, a related page will be showed.

Flask module and some other modules are used in main.py. We use recommendation.py to calculate the results and transfer the results into json format. Then data of json format are sent to front end. These pictures which are related to movies are download from website *‘https://www.imdb.com/’* automatically.

5.1 Searching results:

The front page!



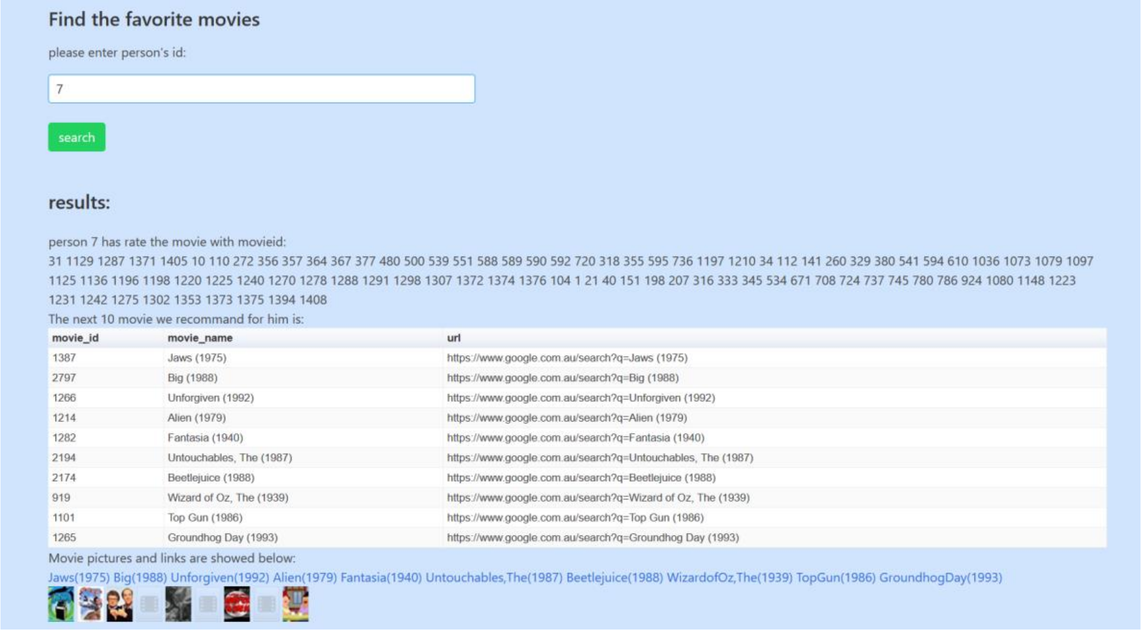
Movie recommendation system

Find the favorite movies

please enter person's id:

results:

The search result and the expected user’s rating. We use computer crawler to get the information of url links and pictures



Find the favorite movies

please enter person's id:

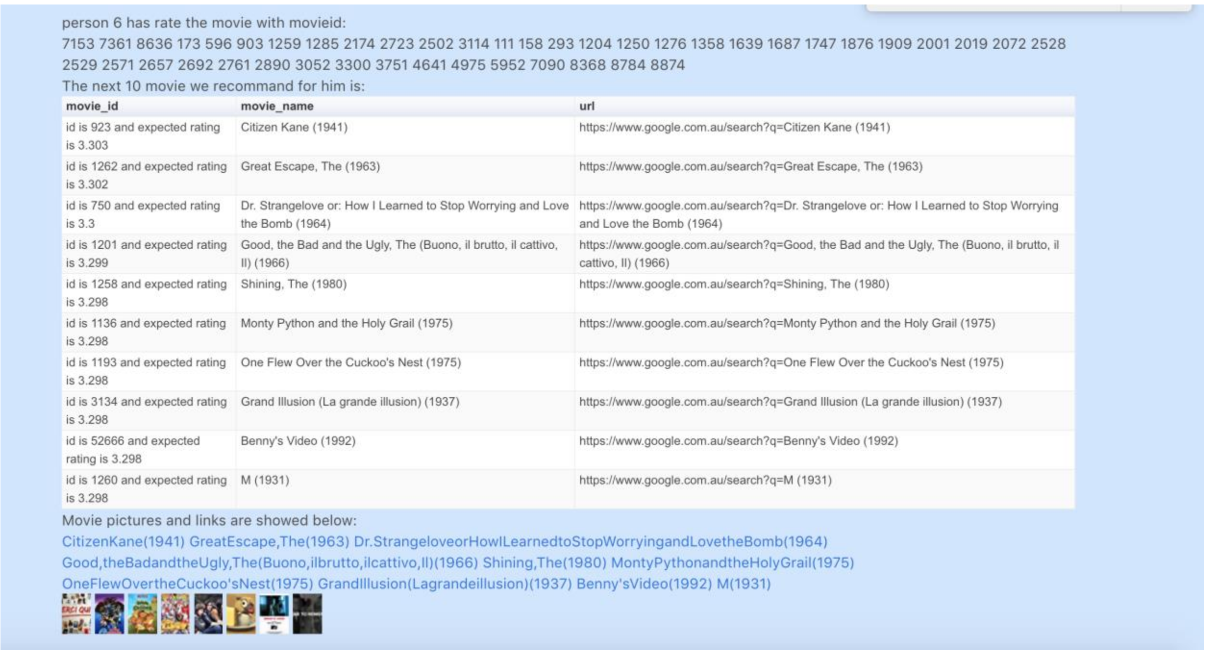
7

results:

person 7 has rate the movie with movieid:
31 1129 1287 1371 1405 10 110 272 356 357 364 367 377 480 500 539 551 588 589 590 592 720 318 355 595 736 1197 1210 34 112 141 260 329 380 541 594 610 1036 1073 1079 1097 1125 1136 1196 1198 1220 1225 1240 1270 1278 1288 1291 1298 1307 1372 1374 1376 104 1 21 40 151 198 207 316 333 345 534 671 708 724 737 745 780 786 924 1080 1148 1223 1231 1242 1275 1302 1353 1373 1375 1394 1408
The next 10 movie we recommand for him is:

movie_id	movie_name	url
1387	Jaws (1975)	https://www.google.com.au/search?q=Jaws (1975)
2797	Big (1988)	https://www.google.com.au/search?q=Big (1988)
1266	Unforgiven (1992)	https://www.google.com.au/search?q=Unforgiven (1992)
1214	Alien (1979)	https://www.google.com.au/search?q=Alien (1979)
1282	Fantasia (1940)	https://www.google.com.au/search?q=Fantasia (1940)
2194	Untouchables, The (1987)	https://www.google.com.au/search?q=Untouchables, The (1987)
2174	Beetlejuice (1988)	https://www.google.com.au/search?q=Beetlejuice (1988)
919	Wizard of Oz, The (1939)	https://www.google.com.au/search?q=Wizard of Oz, The (1939)
1101	Top Gun (1986)	https://www.google.com.au/search?q=Top Gun (1986)
1265	Groundhog Day (1993)	https://www.google.com.au/search?q=Groundhog Day (1993)

Movie pictures and links are showed below:
Jaws(1975) Big(1988) Unforgiven(1992) Alien(1979) Fantasia(1940) Untouchables,The(1987) Beetlejuice(1988) WizardofOz,The(1939) TopGun(1986) GroundhogDay(1993)

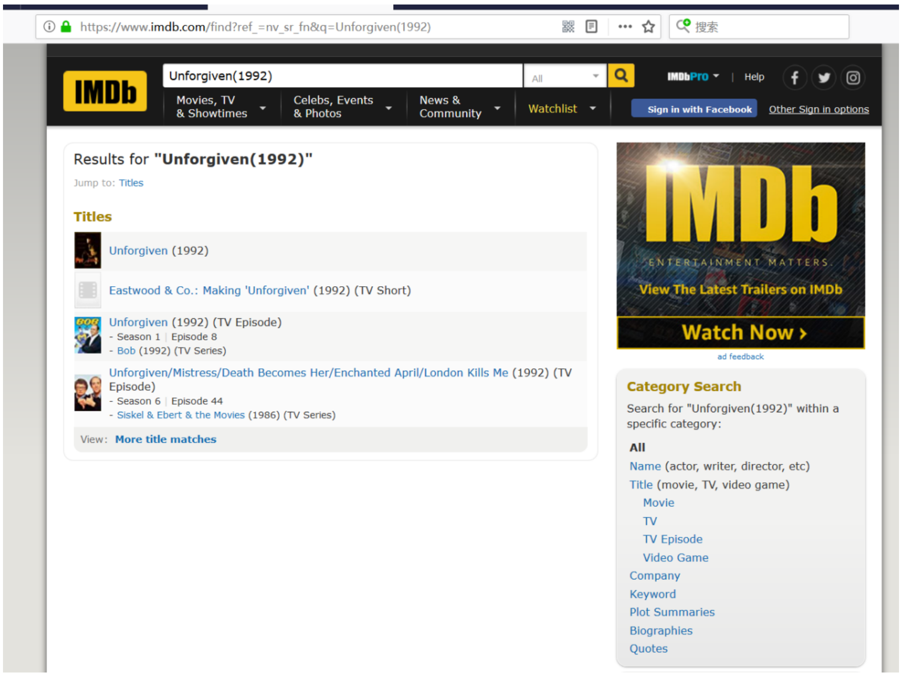


person 6 has rate the movie with movieid:
7153 7361 8636 173 596 903 1259 1285 2174 2723 2502 3114 111 158 293 1204 1250 1276 1358 1639 1687 1747 1876 1909 2001 2019 2072 2528 2529 2571 2657 2692 2761 2890 3052 3300 3751 4641 4975 5952 7090 8368 8784 8874
The next 10 movie we recommand for him is:

movie_id	movie_name	url
id is 923 and expected rating is 3.303	Citizen Kane (1941)	https://www.google.com.au/search?q=Citizen Kane (1941)
id is 1262 and expected rating is 3.302	Great Escape, The (1963)	https://www.google.com.au/search?q=Great Escape, The (1963)
id is 750 and expected rating is 3.3	Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1964)	https://www.google.com.au/search?q=Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1964)
id is 1201 and expected rating is 3.299	Good, the Bad and the Ugly, The (Buono, il brutto, il cattivo, Il) (1966)	https://www.google.com.au/search?q=Good, the Bad and the Ugly, The (Buono, il brutto, il cattivo, Il) (1966)
id is 1258 and expected rating is 3.298	Shining, The (1980)	https://www.google.com.au/search?q=Shining, The (1980)
id is 1136 and expected rating is 3.298	Monty Python and the Holy Grail (1975)	https://www.google.com.au/search?q=Monty Python and the Holy Grail (1975)
id is 1193 and expected rating is 3.298	One Flew Over the Cuckoo's Nest (1975)	https://www.google.com.au/search?q=One Flew Over the Cuckoo's Nest (1975)
id is 3134 and expected rating is 3.298	Grand Illusion (La grande illusion) (1937)	https://www.google.com.au/search?q=Grand Illusion (La grande illusion) (1937)
id is 52666 and expected rating is 3.298	Benny's Video (1992)	https://www.google.com.au/search?q=Benny's Video (1992)
id is 1260 and expected rating is 3.298	M (1931)	https://www.google.com.au/search?q=M (1931)

Movie pictures and links are showed below:
CitizenKane(1941) GreatEscape,The(1963) Dr.StrangeloveorHowILearnedtoStopWorryingandLovetheBomb(1964) Good,theBadandtheUgly,The(Buono,ilbrutto,ilcattivo,Il)(1966) Shining,The(1980) MontyPythonandtheHolyGrail(1975) OneFlewOvertheCuckoo'sNest(1975) GrandIllusion(Lagrandeillusion)(1937) Benny'sVideo(1992) M(1931)

5.2 Related pages:



https://www.imdb.com/find?ref_=nv_sr_fn&q=Unforgiven(1992)

Results for "Unforgiven(1992)"

Jump to: Titles

Titles

Unforgiven (1992)

Eastwood & Co.: Making 'Unforgiven' (1992) (TV Short)

Unforgiven (1992) (TV Episode)
- Season 1 | Episode 8
- Bob (1992) (TV Series)

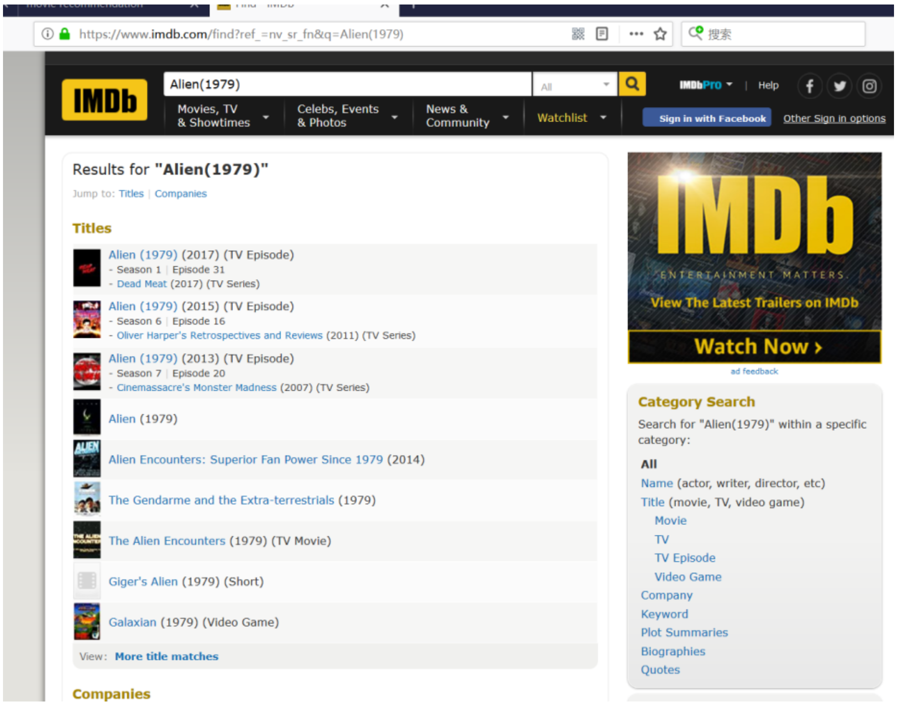
Unforgiven/Mistress/Death Becomes Her/Enchanted April/London Kills Me (1992) (TV Episode)
- Season 6 | Episode 44
- Siskel & Ebert & the Movies (1986) (TV Series)

View: More title matches

Category Search

Search for "Unforgiven(1992)" within a specific category:

All
Name (actor, writer, director, etc)
Title (movie, TV, video game)
Movie
TV
TV Episode
Video Game
Company
Keyword
Plot Summaries
Biographies
Quotes



Results for **"Alien(1979)"**

Jump to: Titles | Companies

Titles

Alien (1979) (2017) (TV Episode)
- Season 1 | Episode 31
- Dead Meat (2017) (TV Series)

Alien (1979) (2015) (TV Episode)
- Season 6 | Episode 16
- Oliver Harper's Retrospectives and Reviews (2011) (TV Series)

Alien (1979) (2013) (TV Episode)
- Season 7 | Episode 20
- Cinemassacre's Monster Madness (2007) (TV Series)

Alien (1979)

Alien Encounters: Superior Fan Power Since 1979 (2014)

The Gendarme and the Extra-terrestrials (1979)

The Alien Encounters (1979) (TV Movie)

Giger's Alien (1979) (Short)

Galaxian (1979) (Video Game)

View: **More title matches**

Companies

Category Search

Search for "Alien(1979)" within a specific category:

All
Name (actor, writer, director, etc)
Title (movie, TV, video game)
Movie
TV
TV Episode
Video Game
Company
Keyword
Plot Summaries
Biographies
Quotes

References

1. J Schafer, D Frankowski, J Herlocker, S Sen “Introduction to recommender systems: Algorithms and evaluation”, ACM Transactions on Information Systems (TOIS), 2007

2. F.O. Isinkaye a, *, Y.O. Folajimi b , B.A. Ojokoh c “Recommendation systems: Principles, methods and evaluation” , Egyptian Informatics Journal , 2015

3. RahulKataryaOm PrakashVerma “An effective collaborative movie recommender system with cuckoo search“ , Egyptian Informatics Journal , 2017