Anime Adaptive Recommendation System

Xiang Liu, xil199@pitt.edu
Ying Ben, bey8@pitt.edu
Yuling Chen, yuc104@pitt.edu
Qizhou Huang, qih31@pitt.edu

1. Introduction

Recommender systems were designed to helping users find what they would like and give them a quick access to the world they interested. In this project, we used the data of anime and tried to implement the recommendation by analyzing the history ratings of users and favorite genres of anime. We choose the item-based collaborative filtering algorithm that extracts the relationships between items based on the variety of ratings made by different users. In other words, we assume that if users give a higher scale rating for an anime, they may also like other animes that have similar ratings. The bigger challenges of this project are the huge dataset which contains more than 6 million data, we cannot deal with it on the real time and it also will influence the efficiency of our system. Besides, there are so many kinds of animes, some genres are uncommon so that it’s hard to find the related information about them.

2. UI design

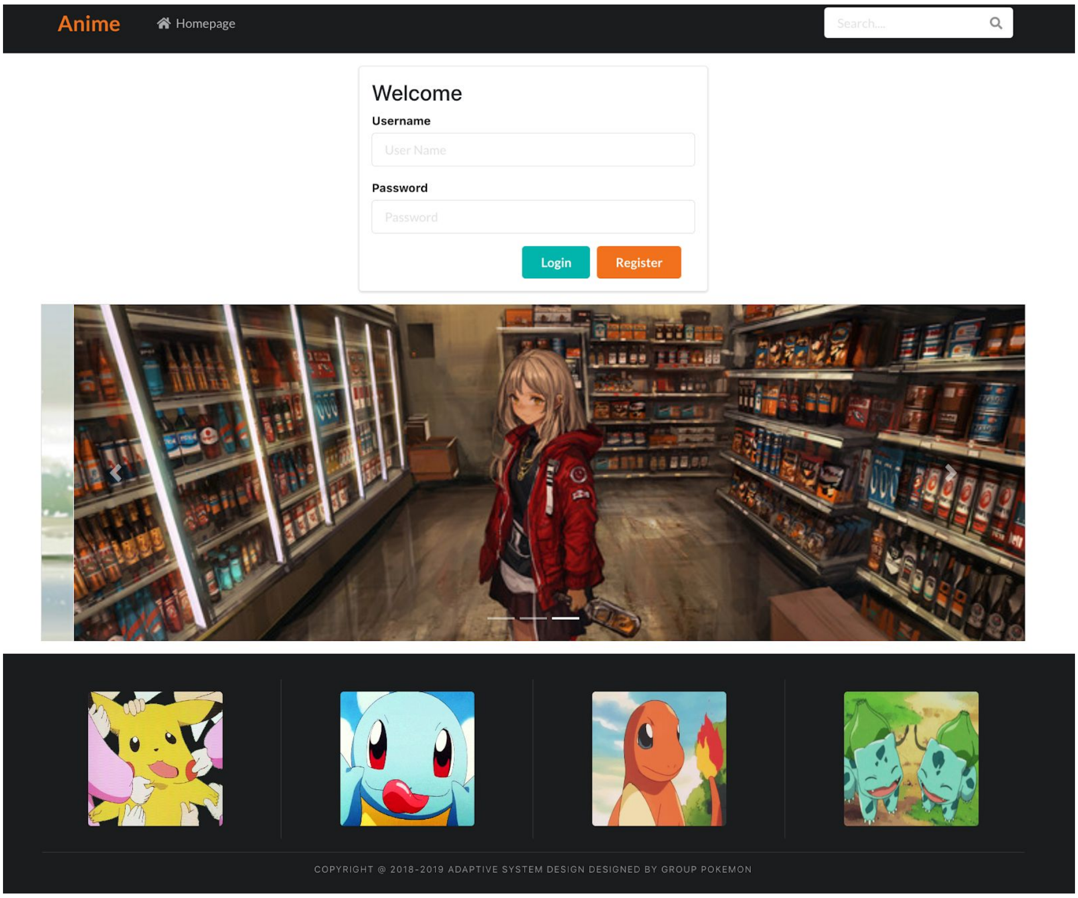

Figure 1: Homepage

This is our homepage. Every user needs to register by setting the username and password at the first time, or they can log in with the existing account. The last four cartoon images in footer represent every member of our group.

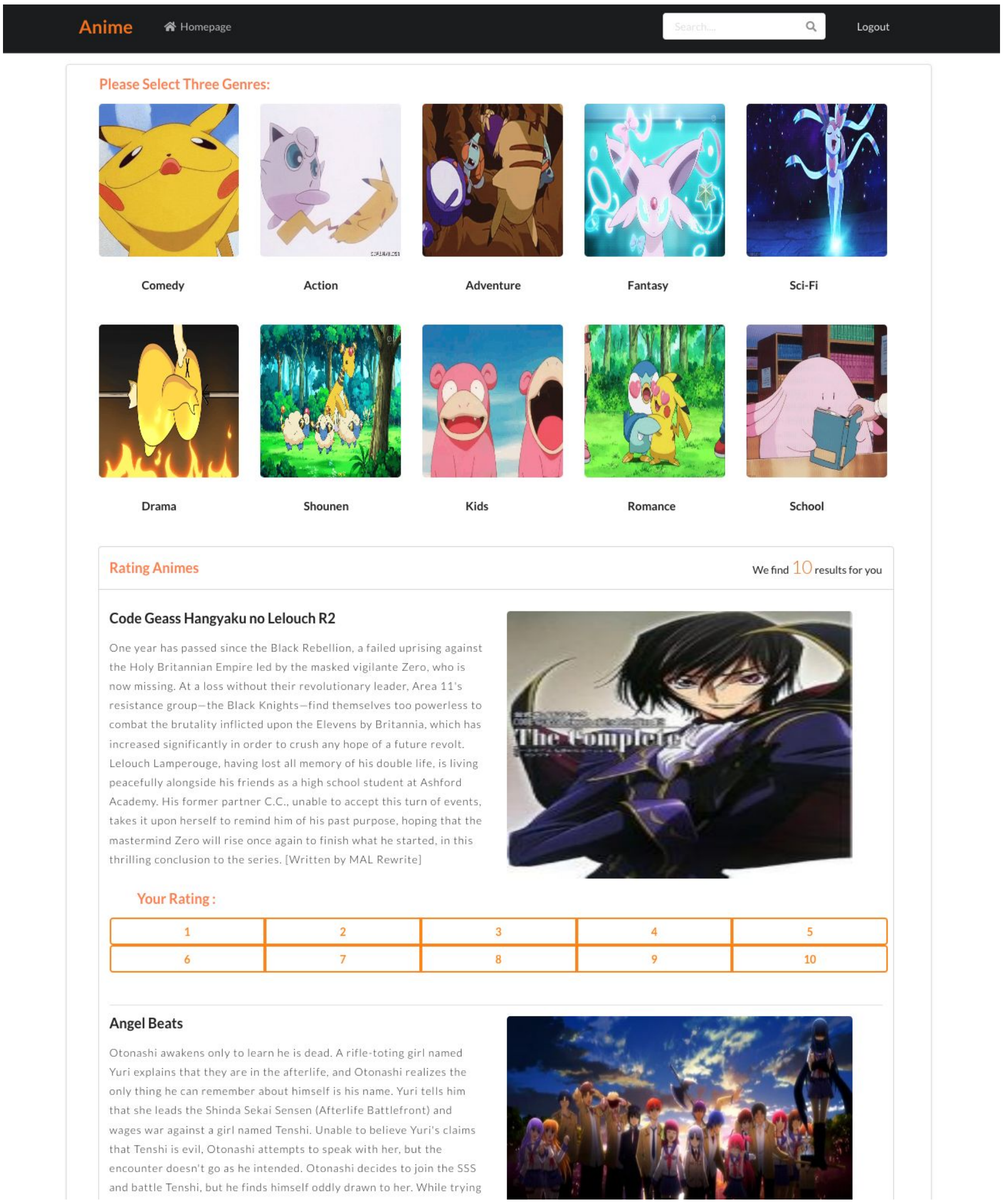

Figure 2: Start page for buidling user profile

After logging in successfully, users need to select at most three genres they like from 10 genres that are common for them. And then we select the top 10 animes dynamically according to the number of ratings on the database in order to know more about the users' preferences. Users can read the description of this anime and rate it from 1-10. Only after they completed the whole process in this page can they move to the next page. It is a good way to solve the problem of a cold start. Only if the users give us some information from them that we can give the better recommendation for them. They can also choose not to participate at all, they can only get trending recommendation though.

Figure 3: Results of our recommendation

In the results page, we provide three recommendations for users. They can view the results based on the top genres they chose. It can give comprehensive information for users to choose from. Besides, we also provide the recommendation of trending by dynamically show animes with top 30 amount of members. The user can use the feature of a filter to select the animes based on genre, duration or type. It will help them find the animes they like in an efficient way. At last, we have the recommendation based on ratings which use collaborative filtering.

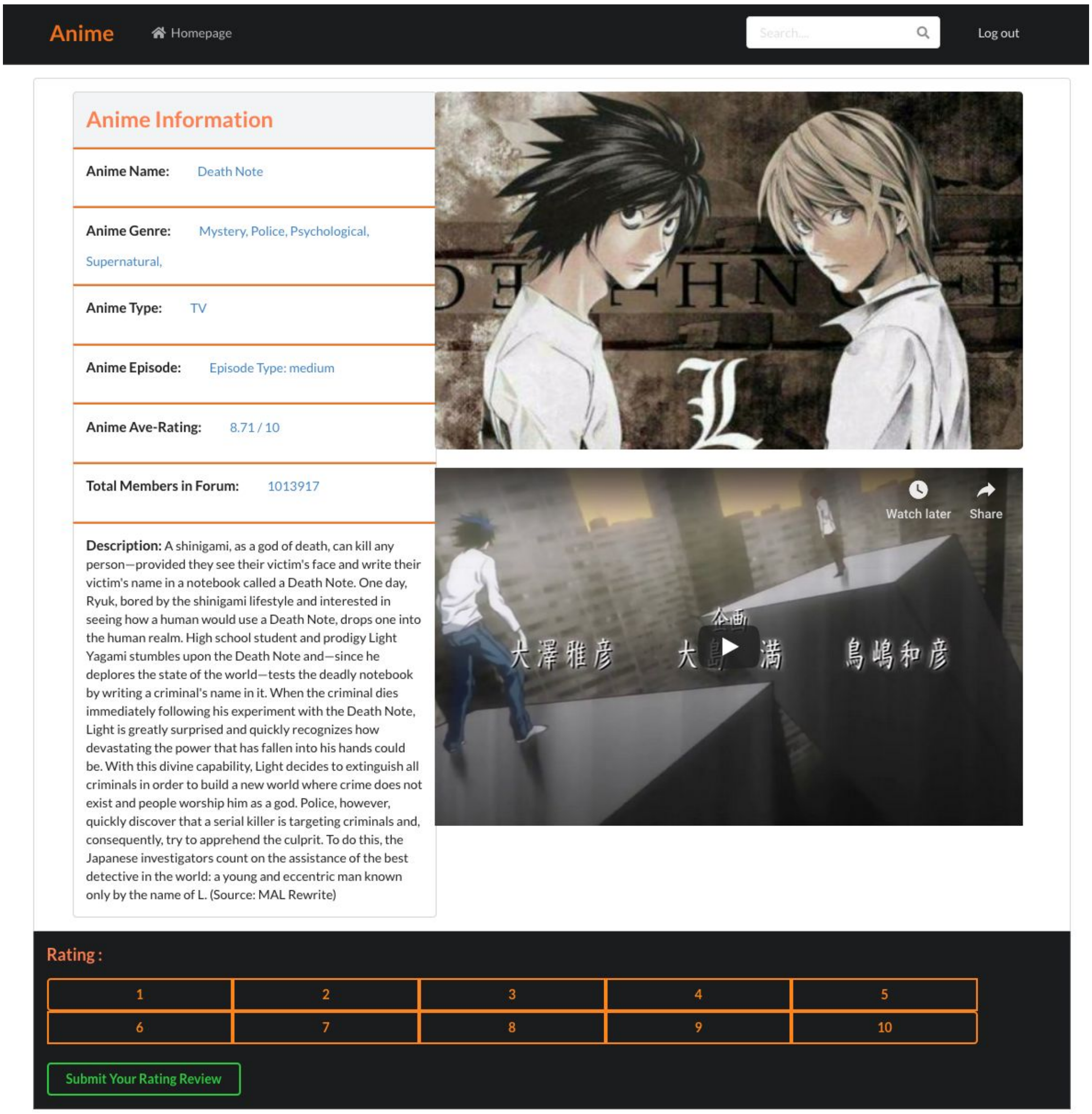

Figure 4: Detail page for animes

When clicking the specific anime, the page will be redirected to the new page to show major information of this anime, such as episode, average rating, even can watch the short related videos exported from Youtube. Users also can submit their rating on this page. It's accord with the principle that users normally will give an actual rating after receive enough information and it will stimulate users to give feedback at the same time.

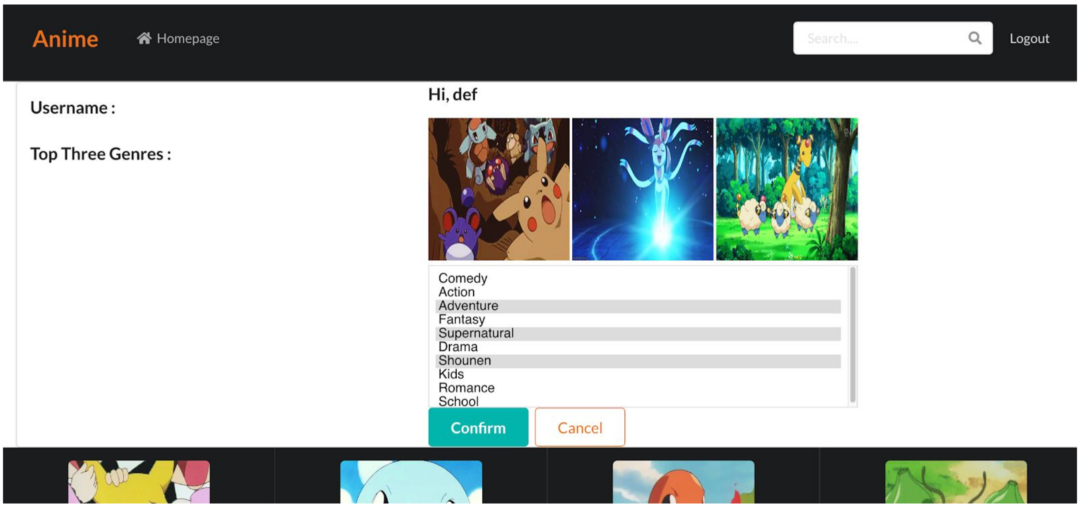

Figure 5: Checking user profile

Users can change the genres they chose before anytime just by click the new genres and then press the "Confirm" button. After that, our database will be refresh again and the results of the recommendation also will change. It shows the feature of adaptive in this system.

3. Data preprocessing and database design

3.1 Data source

The dataset contains information about 73,516 users' rating on 12,294 anime, 7813738 rating records in total and some basic introduction of the animes [1]. Myanimelist.net for providing anime data and user ratings. This data describes the information of each animes according to the following features:

Animeinfo.csv

- anime_id - myanimelist.net's unique id identifying an anime.
- name - full name of anime.
- genre - comma separated list of genres for this anime.
- type - movie, TV, OVA, etc.
- episodes - how many episodes in this show. (1 if movie).
- rating - average rating out of 10 for this anime.
- members - number of community members that are in this anime's "group".

anime_id	name	genre	type	duration	rating	members	url
1535	Death Not	Mystery, P	TV	medium	8.71	1013917	http://img4.imgtn.bdimg.com/it/u=4255424300,1815115452&fm=26&gp=0.jpg
11757	Sword Art	Action, Ad	TV	medium	7.83	893100	http://img0.imgtn.bdimg.com/it/u=1162749346,407366774&fm=26&gp=0.jpg
16498	Shingeki n	Action, Dr	TV	medium	8.54	896229	http://img3.imgtn.bdimg.com/it/u=1036097091,3780344626&fm=26&gp=0.jpg
1575	Code Geas	Action, Me	TV	medium	8.83	715151	http://img4.imgtn.bdimg.com/it/u=3973500382,1278873814&fm=26&gp=0.jpg
226	Elfen Lied	Action, Dr	TV	short	7.85	623511	http://img2.imgtn.bdimg.com/it/u=1448394834,2788819760&fm=26&gp=0.jpg
6547	Angel Beat	Action, Co	TV	short	8.39	717796	http://img2.imgtn.bdimg.com/it/u=1131750086,3941474638&fm=26&gp=0.jpg
20	Naruto	Action, Co	TV	long	7.81	683297	http://img3.imgtn.bdimg.com/it/u=2687293353,2319663022&fm=26&gp=0.jpg
121	Fullmetal A	Action, Ad	TV	medium	8.33	600384	http://img5.imgtn.bdimg.com/it/u=3721240610,3172364047&fm=26&gp=0.jpg
5114	Fullmetal A	Action, Ad	TV	medium	9.26	793665	http://img1.imgtn.bdimg.com/it/u=1102831640,112890545&fm=26&gp=0.jpg
4224	Toradora	Comedy, R	TV	medium	8.45	633817	http://img5.imgtn.bdimg.com/it/u=2526573885,2542072842&fm=26&gp=0.jpg
2904	Code Geas	Action, Dr	TV	medium	8.98	572888	http://img5.imgtn.bdimg.com/it/u=1581483358,2916872153&fm=15&gp=0.jpg
8074	Highschoo	Action, Ec	TV	short	7.46	535892	http://img5.imgtn.bdimg.com/it/u=473593451,3156264264&fm=26&gp=0.jpg
199	Sen to Chil	Adventure,	Movie	short	8.93	466254	http://img2.imgtn.bdimg.com/it/u=896929401,2873064191&fm=26&gp=0.jpg
10620	Mirai Nikki	Action, My	TV	medium	8.07	657190	http://img3.imgtn.bdimg.com/it/u=403393653,1110972858&fm=26&gp=0.jpg
2167	Clannad	Comedy, D	TV	medium	8.3	566690	http://img0.imgtn.bdimg.com/it/u=931467689,1256674891&fm=26&gp=0.jpg
9919	Ao no Exo	Action, De	TV	medium	7.92	583823	http://img1.imgtn.bdimg.com/it/u=2983586010,3774030905&fm=26&gp=0.jpg
11111	Another	Horror, My	TV	short	7.88	534657	http://img3.imgtn.bdimg.com/it/u=3236724984,2231923873&fm=26&gp=0.jpg
3588	Soul Eater	Action, Ad	TV	medium	8.08	580184	http://img1.imgtn.bdimg.com/it/u=2067850575,1434412300&fm=26&gp=0.jpg
2001	Tengen To	Action, Ad	TV	medium	8.78	562962	http://img3.imgtn.bdimg.com/it/u=3086936795,4029823376&fm=26&gp=0.jpg
9253	SteinsGate	Sci-Fi, Thri	TV	medium	9.17	673572	http://img2.imgtn.bdimg.com/it/u=3525434748,2848367454&fm=26&gp=0.jpg
853	Ouran Kou	Comedy, H	TV	medium	8.39	422271	http://img0.imgtn.bdimg.com/it/u=3209593336,368969426&fm=26&gp=0.jpg
849	Suzumiya I	Comedy, M	TV	short	8.06	428569	http://img0.imgtn.bdimg.com/it/u=1068981307,880648277&fm=26&gp=0.jpg
6746	Durarara	Action, My	TV	medium	8.38	556431	http://img4.imgtn.bdimg.com/it/u=3399568019,3400791297&fm=26&gp=0.jpg
19815	No Game I	Adventure,	TV	short	8.47	602291	http://img0.imgtn.bdimg.com/it/u=1311597243,925157547&fm=26&gp=0.jpg
22319	Tokyo Ghc	Action, Dr	TV	short	8.07	618056	http://img5.imgtn.bdimg.com/it/u=4142216676,129239669&fm=26&gp=0.jpg
4181	Clannad A	Drama, Far	TV	medium	9.06	456749	http://img1.imgtn.bdimg.com/it/u=2132196663,3742505294&fm=26&gp=0.jpg
30	Neon Gene	Action, De	TV	medium	8.32	461946	http://img3.imgtn.bdimg.com/it/u=2738715147,892108512&fm=26&gp=0.jpg
431	Howl no U	Adventure,	Movie	short	8.74	333186	http://img1.imgtn.bdimg.com/it/u=3894347254,319076505&fm=15&gp=0.jpg
813	Dragon Ba	Action, Ad	TV	long	8.32	375662	http://img3.imgtn.bdimg.com/it/u=2253604519,389482574&fm=26&gp=0.jpg

Figure 6: Animeinfo.csv after preprocessing

Rating.csv

- user_id - non-identifiable randomly generated user id.
- anime_id - the anime that this user has rated.
- rating - rating out of 10 this user has assigned (-1 if the user watched it but didn't assign a rating).

user_id	anime_id	rating
71537	1535	10
70725	1535	9
70953	1535	10
68084	1535	8
67991	1535	10
66936	1535	10
66492	1535	10
66506	1535	9
69076	1535	10
66957	1535	8
66989	1535	9
65108	1535	5
64372	1535	8

Figure 7: Rating.csv after preprocessing

3.2 Data Preprocessing

Rating contains -1, but based on our algorithm design and the data description, we take it as watching history data instead of a rating, so we remove all the -1 record.

In the anime table, we convert some special characters in the name to make sure the data could be imported to the database.
For the genre field, we did statistics about it, get representative top 10 genres which used in the user profile building section. In the original dataset, episodes fields range in 1 to hundreds, in order to better recommendation, we transfer it to duration, for example, short means the animes last less than one season. And we crawl posters based on the name for showing in the front-ends.

For the recommendation, when we did the algorithm testing using real-time calculation, the whole dataset is too huge, so we Select top 50 animes with most rating records and the users with more ratings, the 167 users all graded more than 45 animes of the top 50 to improve running speed.

3.3 Database design and ER diagram

From the introduction above, the dataset of our project is consist of three tables, user, rating and anime information. The user table records the information of userid , username, and password which are used to login to a user profile. User table also records the top 3 favorite genres selected by users. Next one is anime information table, this table includes detail information of animes like id, name, genre, type, duration and anime images. it also contains information associated with users like user rating towards the animes and the number of users who rated this anime. The last table is the rating table which contains the records of users' rating. For example, the rating of anime 1 from the user whose id equals to 123. The new user table stored the more user information includes the user_id, username, user password and top 3 favorite genres for the recommendation, using the user_id as the primary key. In the anime table, anime_id is the primary key.

We use MySQL to build the data warehouse of our project, here is the ER diagram of our project.

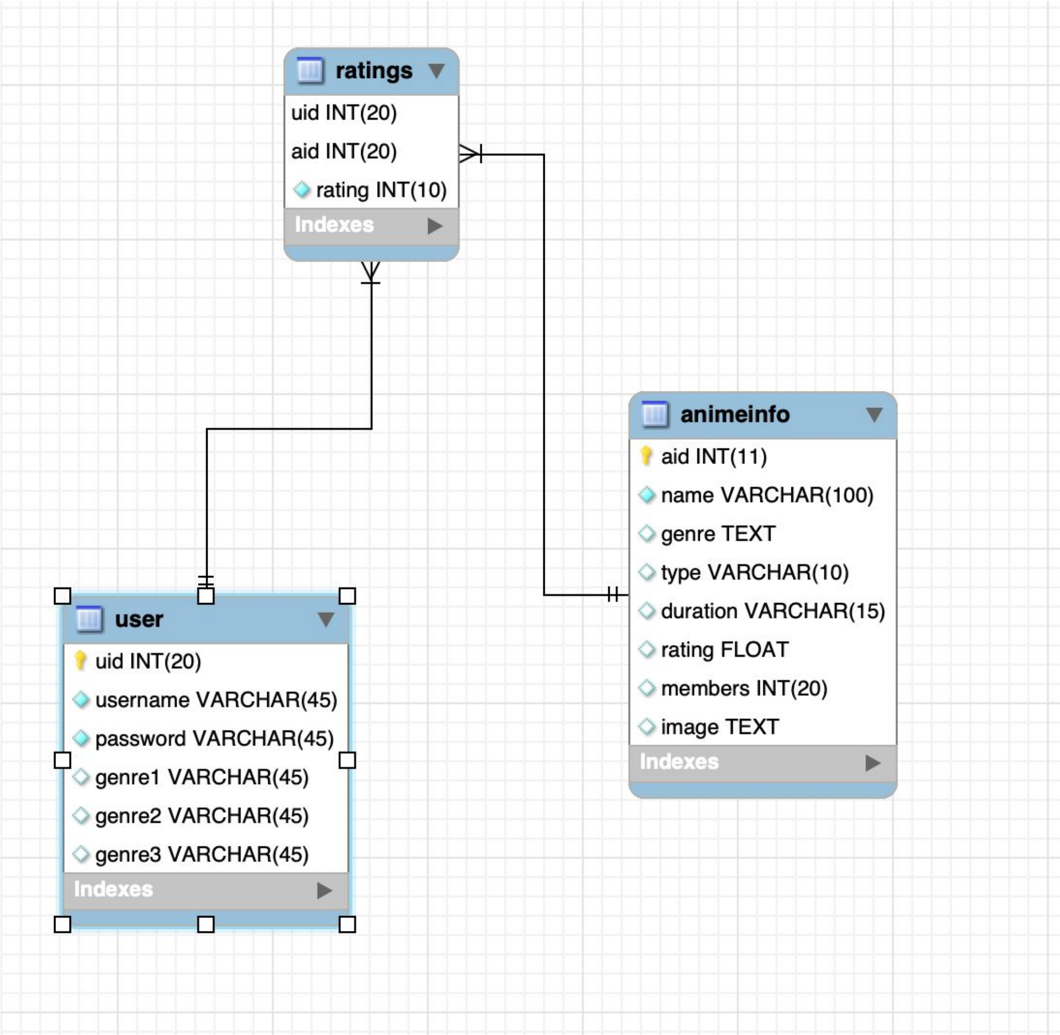

Figure 8: ER diagram of data

4. Algorithm design & Literature analysis

In this section, we will introduce the path to develop the algorithm for the system from three phase.

4.1 Phase One

Firstly, based on the rating history, we calculate the relationship between animes using co-occurrence. For example, M2-M3 means there are two users both rating M2&M3.

	M1	M2	M3	M4	M5
M1	2	2	1	1	0
M2	2	4	2	2	1
M3	1	2	2	0	1
M4	1	2	0	2	0
M5	0	1	1	0	1

Figure 9: co-occurrence matrix

Then we normalize the table.

	M1	M2	M3	M4	M5
M1	2/6	2/6	1/6	1/6	0
M2	2/11	4/11	2/11	2/11	1/11
M3	1/6	2/6	2/6	0	1/6
M4	1/5	2/5	0	2/5	0
M5	0	1/3	1/3	0	1/3

Figure 10: Normalized co-occurrence matrix

Finally, multiply Co-matrix by rating matrix to give the non-rated anime a prediction rating. Recommend user the top 10 animes in the prediction results.

Co Matrix

	M1	M2	M8
M1	2/4	2/4	0
M2	2/8	4/8	2/8
M8	0	2/4	2/4

Rating Matrix

Movie	User B rating
M1	3
M2	7
M3	8

Figure 11: Prediction

4.2 Phase Two - pure cosine similarity

Based on 4.1, we started to consider each rating values itself as well. After reading a paper written by Sarwar, I find many method to calculate similarity. One is called pure cosine similarity. So once we find the mutual ratings for two animes, the similarity is calculated as the formula below [2].

$$sim(i,j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2}$$

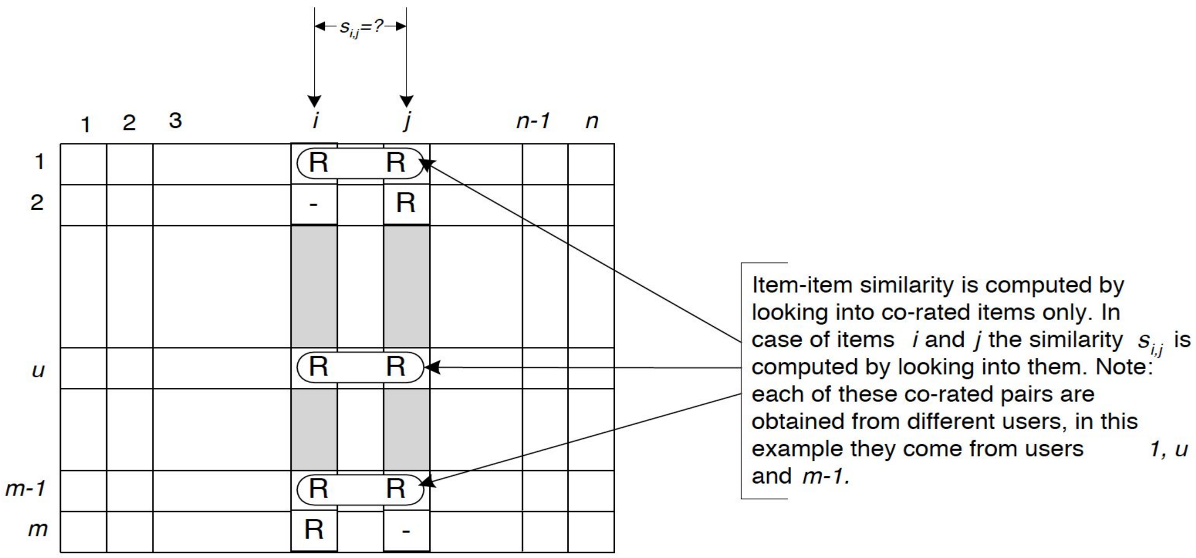

Figure 12: Pure cosine similarity

4.3 Phase Three - adjusted cosine similarity

Based on the progress 4.2, we still have space to increase the performance. Because the pure cosine similarity ignored the fact that certain users' rating preference. For example, some the users may like to rate higher and some may rate lower. Here, we used adjusted cosine similarity to overcome this issue. We simply calculate each user's average rating for each mutual rating record and make the rating subtracts average rating to see how the rating was different from the user's normal behavior. The formula is as below [2].

$$sim(i, j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R_u})(R_{u,j} - \bar{R_u})}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R_u})^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R_u})^2}}.$$

Here $\bar{R_u}$ is the average of the u-th user's ratings.

Figure 13: adjusted cosine similarity

After we get the similarity, we simply use "Weight Sum" method to get the predicted rating for each unrated animes based on what current user have rated. The formula is as below [2].

$$P_{u,i} = \frac{\sum_{\text{all similar items, N}}(s_{i,N} * R_{u,N})}{\sum_{\text{all similar items, N}}(|s_{i,N}|)}$$

Figure 14: Weight Sum Prediction

5. Implementation

The system is an online web application developed with Node.js and express.js. The database is using MySQL because there is a lot of calculation using join action and complex SQL. MySQL is a typical choice for the relational database.

Plus, to make application filled with more contents for user experience, we also use youtube API and Kitsu API to dynamically generate the updated video and description for related animes.

5.1 Implementation of item collaborative filtering

We tried two ways to calculate the similarity and used list and object to operate the result from the database to get the top 10 animes for the recommendation based on animes' rating table. The detailed steps are listed in the table below.

Input	Process	Output
Database access	Find the animes list that rated A and unrated B by current login user	rated list A, unrated list B, represented by anime id
rated list A, unrated list B	For every unrated anime b in list B, calculate the similarity with every anime a in list A	double for loop inside to have pair (b,a) in each iteration
pair (b,a)	For each (b,a) pair, find their mutual ratings vector ub and vector ua that rated by same user	vector (ua), vector (ub), represented by ratings
vector (ua), vector (ub)	calculate the pure or adjusted cosine similarity (for adjusted, each rating in vector subtracts the average rating for related user)	similarity(b,a)=> similarity list [(b,a1), (b,a2),(b,a3)...] that a1, a2, a3 from A
similarity list [(b,a)...], the current user rating for every elements in list A	For each b, using weighted sum and similarity array with each element in list A to calculate the predicted ratings for current user	for current user, get the predicted rating for every elements in list B
all predicted ratings for animes in B	sort based on value of rating in desc and get top 10's anime ID	send the selected anime IDs to front end page

Table 1. Algorithm implementation process

5.2 Implementation of other recommendation

Besides collaborative filtering based on animes, we also implemented other static or dynamic recommendation that can be helpful for users to discover new animes. The details are listed in the table below.

name	principle	application
genres selection pool	prefixed with genres of most top 10 number of rated users	user can select or update their top three genres from genres selection pool
initial animes pool	dynamically show animes with most 10 amount of ratings from	user can initially rate animes from this pool

	rating table	
trending pool	dynamically show animes with top 30 amount of members, filters include genre, duration and type	user can view trending animes combined with filter function
genres recommendation pool	dynamically show union of each genre for top 5 animes based on avgerage rating	user can view animes recommended based on user's top three genres

table 2. Other Algorithm implementation process

6. Conclusion, Limitation & Testing

6.1 Testing

We briefly tested our system with smaller data as well as bigger data. From the response time aspect, it works perfectly with smaller data with the adjusted method. However, with bigger data more than 100,000 ratings, the adjusted method won't response while pure cosine method will response after around one minute. With more than 6 million rating records, none of the methods works anymore. From the aspect of accuracy, due to the limitation of computation, we didn't test heavily. We can clearly see the adjusted method has better performance in the next section and the anime results are relatively promising as well.

6.2 Conlusion

In this project, we used the data of anime and tried to implement the recommendation by analyzing the history ratings of users and favorite genres of anime. We choose the item-based collaborative filtering algorithm that extracts the relationships between items based on the variety of ratings made by different users.

Our project is strictly associated to what we learned in class, In order to simulate the item-based collaborative filtering recommendation, our group devoted time and energy to find the suitable algorithm, after experiment and evaluation, we learned how the item based collaborative filtering works and how to implement it. Besides this, from the algorithm part, we know that various methods to calculate the item similarity between two animes.

When we use the weight sum method to calculate the predicted ratings for unrated animes. If a user only has one rating for certain anime, then the predicted value for other animes will be the same as what he just rated because the cosine value was eliminated by numerator and denominator. So we should have a certain rule to avoid user have only one rating. As we can see figure, user rated only one anime with rating 9. Even though the unrated animes have different similarity value, they all ended up with same absolute rating value 9.

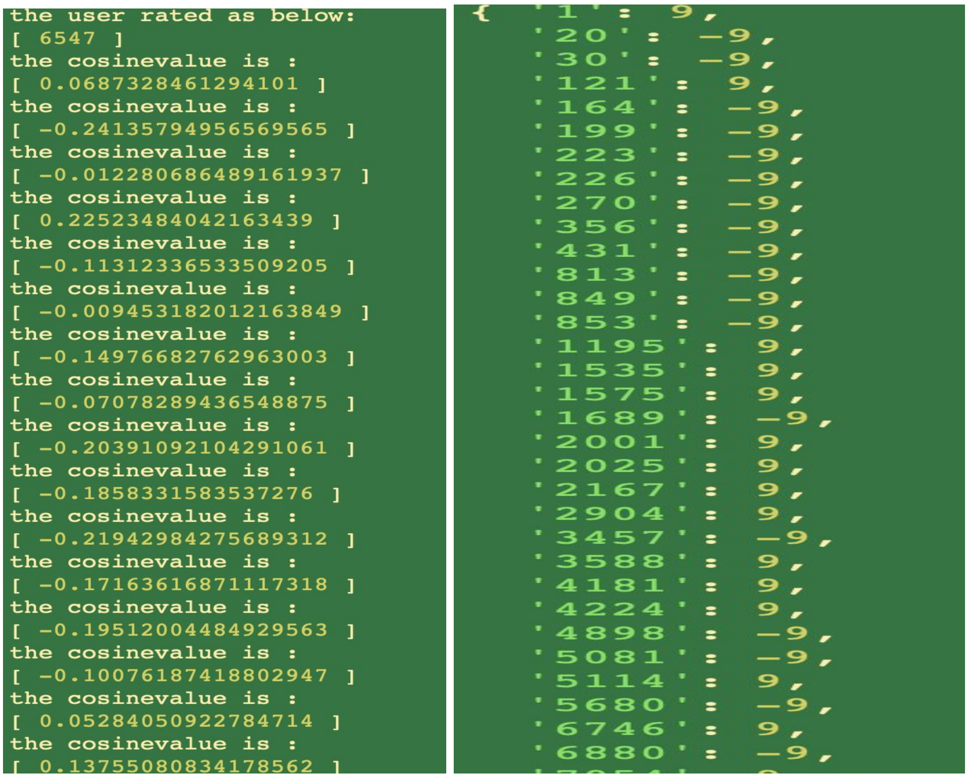

Figure 15. different cosine ended with same rating

Another place needs to focus on is that when we calculate the similarity between two animes, we find two animes that share the same user that rated both of them. Some pairs may have many users in the vector whereas other pairs may have only two or three pairs. It's not convincing to compare similarity with only two or three pairs when you have millions of users. So we suggest using a certain threshold like 1% of the whole user base as a rule to decide whether further the calculation of similarity.

Also, a user is unpredictable, different users have different rating habits, so in our algorithm, we used adjusted cosine similarity instead of purely cosine similarity. In other words, we focus on the tendency of user rating instead of the absolute user rating result. This will make our system have more convincing performance. As we can see from figure 16 and 17, with same rating on same animes, the similarity of same unrated animes using adjusted cosine method vary hugely compared with using pure cosine method. This illustrated that the user rating habit have extremely impact on this anime dataset and adjusted cosine method is much more convincing here. This result applies with full dataset as well.

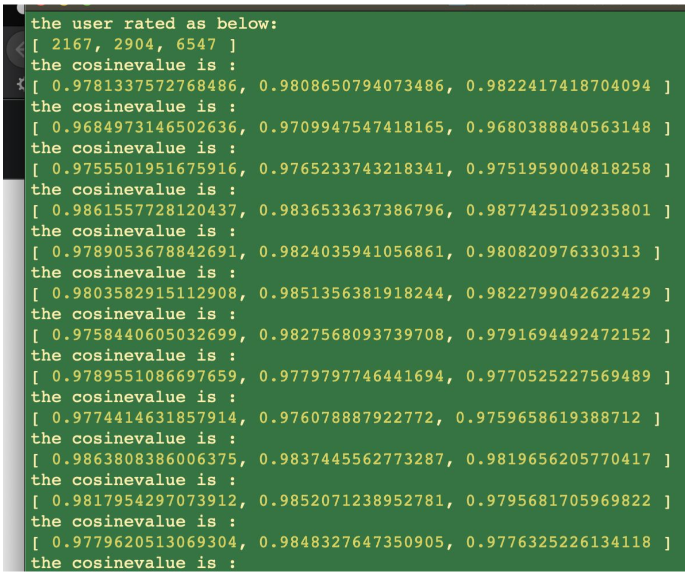

Figure 16. similarity of pure cosine

Figure 17. similarity of adjusted cosine

Last, since item-based similarity does not change frequently, our group suggest to use pre-calculated similarity table in the database and using scripts to update weekly (especially for animes). This operation will increase the efficiency of our system.

6.2 Limitation and future work

As we talked before, the efficiency of our system still has the potential to improve. For now, For the recommendation, when we are testing the algorithm with real-time calculation, the whole dataset is too large to handle, That's why we implement the system using our small-scale dataset first. From the result, we can tell that this small-scale experiment is successful. In future work, our project team is practicing to enlarge the scale of our system to operate on a larger dataset.

Contribution

Ben Ying: Data preprocess, data crawler development, implement first phase recommendation algorithm

Qizhou Huang: requirment anaysis, font-end design and project materials collection.

Xiang Liu: Pure cosine & adjusted cosine algorithm implementation, back end.

Yuling Chen: Analyzing the requirements in depth, design and draw the prototype of interface iteratively

References

[1]Anime Recommendations Database. (2018). Retrieved from https://www.kaggle.com/CooperUnion/anime-recommendations-database

[2]Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. (2001). Item-based Collaborative Filtering Recommendation Algorithms. In: 10th international conference on World Wide Web. [online] pp.285-295. Available at: http://10.1145/371920.372071 [Accessed 9 Dec. 2018].