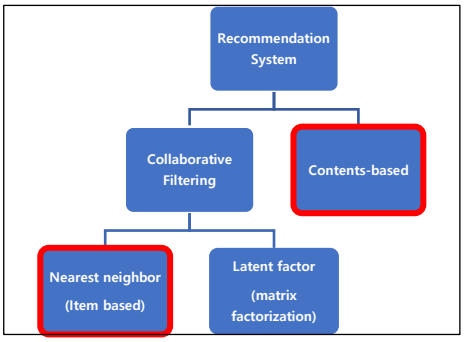
Animation Recommendation System using Contents based & Collaborative (Item based) filtering

20191314 Jihyuk-Choi, 20201181 Jihwan-Oh

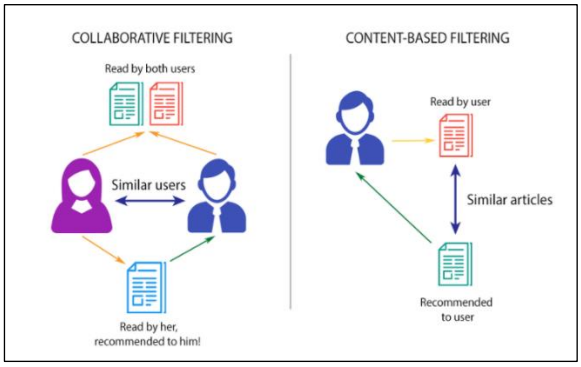
Abstract: **We used an animation database to create our own animation recommendation system, content-based model and item-based collaborative model. Analyze each of the two models we have created and look at what we will explore further.**

I. INTRODUCTION & USEFUL HINTS



<Figure.1. Recommendation system>

Recommendation system is kind of information filtering technology, information that a specific user may be interested in is recommended. The recommendation system is divided as shown Figure 1. We focused in contents based and collaborative(item-based) filtering model.

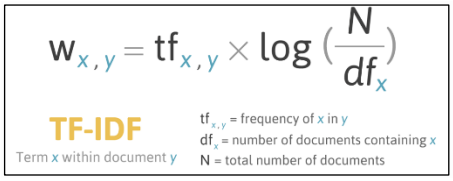


<Figure. 2 Difference between the two models>

Our goal is to understand the principles of various recommender systems and develop our own animated recommender systems. And it is to learn recommendation models other than latent factor collaborative filtering learned in class. Finally, we will create our own animation recommendation system through a model using content-based filtering and collaborative filtering then analyze them.

A. Contents-Based Filtering

The content-based filtering model is a simple algorithm in which content similar to content that users like is recommended. Then, how should we define similar content here? Text mining is used to analyze the characteristics of contents such as genres and synopsis and classify them. At this time, we use a text mining technique called Term Frequency-Inverse Document Frequency (TF-IDF).

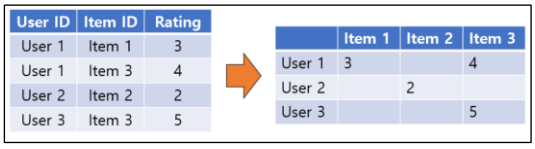


<Figure.3. TF-IDF>

TF-IDF stands for Term Frequency-Inverse Document Frequency, which is a statistical value indicating how important a particular word is by giving different weights for each word. The larger the frequency of occurrence of words, the greater the weight is given. In the code, TF-IDF Vectorizer functions to vectorize text information.

B. Item-Based Collaborative Filtering

Collaborative Filtering (CF) is an algorithm that predicts based on taste information collected from many users. The reasons we selected Item-Based Collaborative Filtering as our recommendation algorithm are as follows.



User ID	Item ID	Rating
User 1	Item 1	3
User 1	Item 3	4
User 2	Item 2	2
User 3	Item 3	5

	Item 1	Item 2	Item 3
User 1	3		4
User 2		2	
User 3			5

<Figure.3. Changes in the User-Item Rating matrix >

Item-Based Collaborative Filtering is commonly known as the filtering used by most services such as Netflix and Amazon. Therefore, we thought that we would be able to learn about the basic algorithms used in real industrial settings.

It is also the same method as User-Based Collaborative Filtering used in class but uses a different similarity matrix. Therefore, we thought that it could be a good comparison target with User-Based Collaborative Filtering that we used in class.

C. Cosine similarity

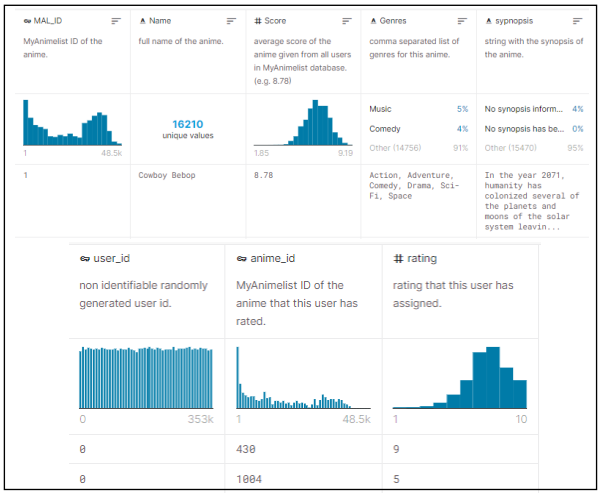
$$\mathrm{Sim}(u,v)^{\cos} = \frac{\vec{r}_u \cdot \vec{r}_v}{\|\vec{r}_u\| \cdot \|\vec{r}_v\|} = \frac{\sum_{i \in I_u \cap I_v} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_u \cap I_v} r_{ui}^2} \cdot \sqrt{\sum_{i \in I_u \cap I_v} r_{vi}^2}}$$

<Figure.4. Cosine similarity>

Cosine similarity is used to measure the similarity between contents. Cosine similarity is as follows. The closer the value is to 1, the more similar it is, and the closer it is to 0, the more different it is. In addition, this is used to create a cosine similarity matrix between contents.

II. PROCESS 1 – Contents Based

A. Data Preprocessing



<Figure.5. Data set>

In the case of contents-based models, basically, all we need is information on contents such as animation names, genres, synopsis, etc. However, user-rating data will also be used for further analysis. This dataset contains 57 million ratings applied to 16.872 animations by 310.059 users.

```
df.head()
```

	anime_id	Name	Genres	sypnopsis
0	1	Cowboy Bebop	Action, Adventure, Comedy, Drama, Sci-Fi, Space	In the year 2071, humanity has colonized sever...
1	5	Cowboy Bebop: Tengoku no Tobira	Action, Drama, Mystery, Sci-Fi, Space	other day, another bounty—such is the life of ...
2	6	Trigun	Action, Sci-Fi, Adventure, Comedy, Drama, Shounen	Vash the Stampede is the man with a $$60,000,0...
3	7	Witch Hunter Robin	Action, Mystery, Police, Supernatural, Drama, ...	ches are individuals with special powers like ...
4	8	Bouken Ou Beet	Adventure, Fantasy, Shounen, Supernatural	It is the dark century and the people are suff...

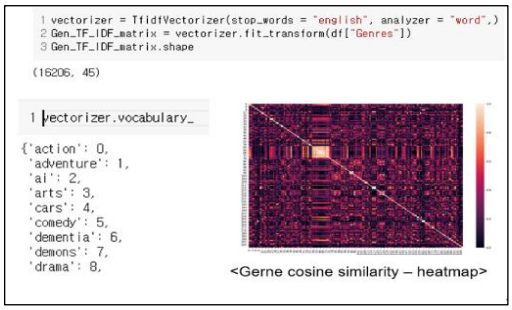
```
df2.head()
```

	user_id	anime_id	rating
0	0	430	9
1	0	1004	5
2	0	3010	7
3	0	570	7
4	0	2762	9

<Figure.6. Preprocessed data>

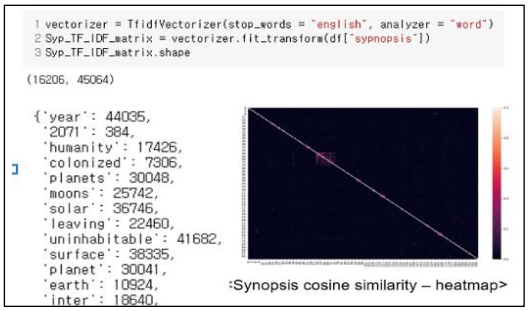
Then we preprocessed this data set. Both datasets here remove null values. In the animation information data, the score means the average rating score of the animation. It is data used in many contents-based recommendations, but I do not use this data because we will analyze it in a different way. Therefore, the column is deleted from the data.

B. Text mining & Calculate cosine similarity



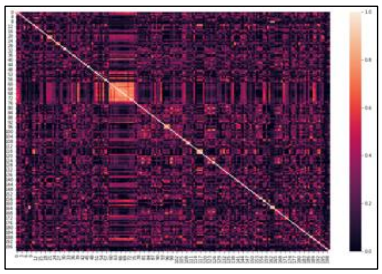
<Figure.8. Genre Text-mining>

The data that can do text mining to our data is genre and synopsis. First, text mining shows that words corresponding to the genre are vectorized as follows. Then, we used heatmap to find out how similar the cosine similarity of the content was compared based on the genre. The red, the higher the relationship, and the darker the color, the less the relationship.



<Figure.9. Genre Text-mining>

In the same way, it can be seen that synopsis was clearly darker than the genre as a result of heatmap analysis. But the dark value is not zero. Due to the nature of the synopsis composed of many sentences, the similarity is inevitably smaller than that of the genre composed of specific words. Therefore, genre text mining cannot be said to be more meaningful.



<Figure.10. Combination>

However, We thought it would be good if the proportion of the genre was a little larger. Therefore, we decided the final contents similarity in consideration of 80% of genre and 20% of synopsis. As this ratio varies, the final recommendation result may vary.

C. Results 1: Contents recommendation

	anime_id	Name	Genres	synopsis	similarity_score
10	20	Naruto	Action, Adventure, Comedy, Super Power, Martia...	oments prior to Naruto Uzumaki's birth, a huge...	1.000000
1508	1735	Naruto: Shippuuden	Action, Adventure, Comedy, Super Power, Martia...	It has been two and a half years since Naruto ...	0.872604
11346	34566	Boruto: Naruto Next Generations	Action, Adventure, Super Power, Martial Arts, ...	Following the successful end of the Fourth Shi...	0.856797
6158	13667	Naruto: Shippuuden Movie 6 - Road to Ninja	Action, Adventure, Super Power, Martial Arts, ...	urning home to Konohagakure, the young ninja c...	0.853405
8831	28755	Boruto: Naruto the Movie	Action, Comedy, Martial Arts, Shounen, Super P...	The spirited Boruto Uzumaki, son of Seventh Ho...	0.827924
4598	8246	Naruto: Shippuuden Movie 4 - The Lost Tower	Action, Comedy, Martial Arts, Shounen, Super P...	signed on a mission to capture Mukade, a miss...	0.819015
5518	10686	Naruto: Honoo no Chuunin Shiken! Naruto vs. Ko...	Action, Adventure, Martial Arts, Shounen, Supe...	Naruto faces off against his old pupil Konoham...	0.812497
10244	32365	Boruto: Naruto the Movie - Naruto ga Hokage ni...	Action, Comedy, Super Power, Martial Arts, Sho...	Bundled with the limited edition of Blu-ray/DV...	0.807435
3904	6325	Naruto: Shippuuden Movie 3 - Hi no Ishi wo Tsu...	Action, Comedy, Martial Arts, Shounen, Super P...	Ninjas with bloodline limits begin disappearin...	0.797468
11640	35072	Boruto: Jump Festa 2016 Special	Action, Adventure, Comedy, Super Power, Martia...	The special anime adaptation of Boruto will be...	0.200624

<Figure.11. Contents based recommend system (basic)>

We first received input as an animation and recommended the top 10 animations with the highest similarity to input animation. If entered the animation "Naruto" as input, and most of the series related to Naruto came out as output. This value may also be meaningful, but we will want to get results except for the related series.

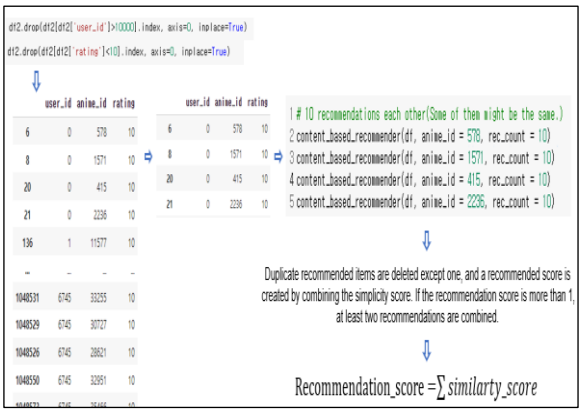
	anime_id	Name	Genres	synopsis	similarity_score
11	21	One Piece	Action, Adventure, Comedy, Super Power, Drama...	'Gol D. Roger was known as the "Pirate King." t...	0.554792
11666	35104	Christmas Carol	Kids, Supernatural	anime adaptation of Charles Dickens' A Christm...	0.479910
5537	10717	Towa no Quon 6: Towa no Quon	Action, Sci-Fi, Super Power, Supernatural	The story follows a boy named Quon and others ...	0.390219
1515	1742	Souryuuden	Mystery, Super Power, Supernatural, Drama	The four Ryudo brothers hold an ancient secret...	0.328704
11372	34629	MiniGARO	Comedy	short web-anime created for the "miniGARO" pro...	0.155219
8855	28853	Pokemon: Pikachu to Pokemon Ongakutai	Adventure, Kids, Fantasy	Pikachu and the rest of the group practice sin...	0.121063
3917	6372	Higashi no Eden Movie I: The King of Eden	Comedy, Drama, Mystery, Romance, Slice of Life...	fter preventing Japan's destruction, Akira Tak...	0.037810
6179	13795	The Green Wind	Fantasy	Promotion video for 21st Century Museum of Con...	0.000000
4613	8312	Piece	Music	music video to the song Piece by singer Aragak...	0.000000
10269	32420	Ken-chan	Music, Kids	music video for the song "Ken-chan" by Yuuki K...	0.000000

<Figure.12. Contents based recommend system (Except series)>

Therefore, we made top 10 recommendation system except for the series like Figure 12.

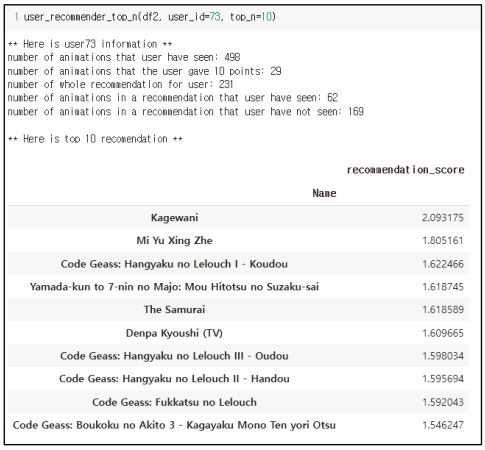
D. Results 2: Contents recommendation for User

Until then, only contents data was needed, but now we are going to make customized recommendations for users using user-rating data. Use user id from 1 to 10000 as data and consider only 10 rating points. Animation with a rating of 10 points is an object that we must consider when recommending it to that user.



<Figure.13. Principle of recommended to users>

The principles recommended to users are shown Figure 13. In this figure, users with zero user id have a total of four animations rated with 10 points out of all animations. After that, the contents-based recommendation created above is recommended for each of the four animations. Then the total number of recommendations will be 40. At this time, among the 40 recommended animations, the ranking and simplicity are different, but there may be animations recommended due to overlapping items. We focus on this case. Duplicate recommendations mean that they are items that must be recommended to the user. Therefore, the recommendation score is defined as a new recommended score, which is calculated as the sum of the overlapping mothers' similarities. Items that do not overlap will have a low recommendation score. In addition, since the maximum value of simplicity is 1, more than 1 means that the recommendation score has been recommended at least twice. In this way, the top 10 animations are recommended in the order of high recommendation scores.



```
user_recommender_top_n(df2, user_id=73, top_n=10)

++ Here is user73 information ++
number of animations that user have seen: 498
number of animations that the user gave 10 points: 29
number of whole recommendation for user: 231
number of animations in a recommendation that user have seen: 62
number of animations in a recommendation that user have not seen: 169

++ Here is top 10 recomendation ++
```

Name	recommendation_score
Kagewani	2.093175
Mi Yu Xing Zhe	1.805161
Code Geass: Hangyaku no Lelouch I - Koudou	1.622466
Yamada-kun to 7-nin no Majo: Mou Hitotsu no Suzaku-sai	1.618745
The Samurai	1.618589
Denpa Kyoushi (TV)	1.609665
Code Geass: Hangyaku no Lelouch III - Oudou	1.598034
Code Geass: Hangyaku no Lelouch II - Handou	1.595694
Code Geass: Fukkatsu no Lelouch	1.592043
Code Geass: Boukoku no Akito 3 - Kagayaku Mono Ten yori Otsu	1.546247

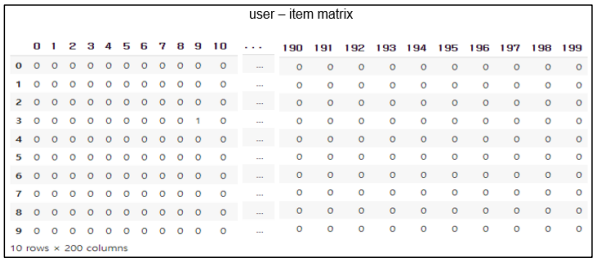
<Figure.14 Contents based recommend system for users>

Here, the information of user 73 is shown Figure. 14. All animations seen by user No. 73 were 498 and 29 of them were given 10 points. At this time, the number of recommended animations different from each other

is 231 by removing duplication. There are 62 animations that he saw here and 169 animations that he didn't see. Finally, there is the top 10 animations recommended to user 73.

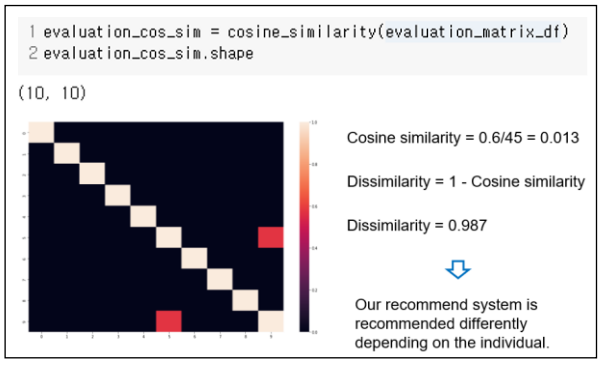
E. Evaluation Our Recommendation System

Finally, we will evaluate the performance of our contents-based service. First, there is 'personalization' as a quantitative evaluation. Personalization is an indicator for evaluating whether the recommendation system recommends the same product to other users equally or differently. That is, the dissimilarity between the recommended product lists of users is obtained.



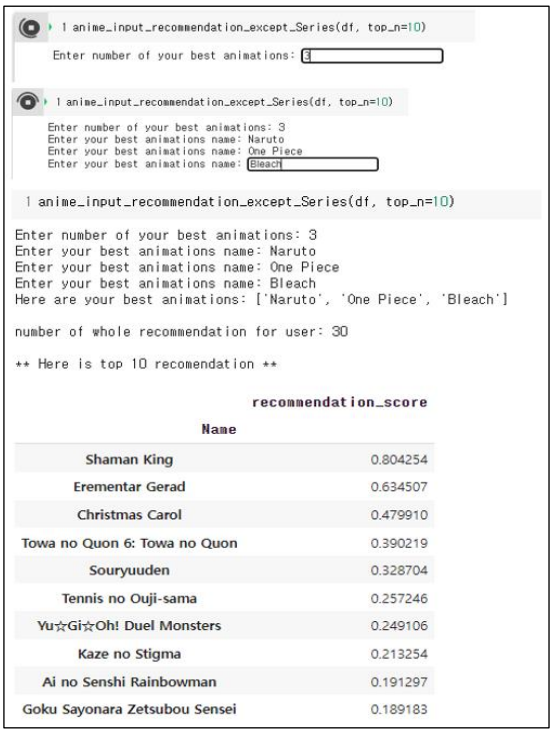
<Figure.15 User-item matrix through contents based model>

The figure 15 is a user-item matrix, indicating 1 or 0 if the item is recommended to the user. By calculating cosine simplicity in this way, you can see how differently animation was recommended among users. The total number of people was too large for computer computation, so we randomly selected 10 users and 200 items. The result heatmap is as follows.



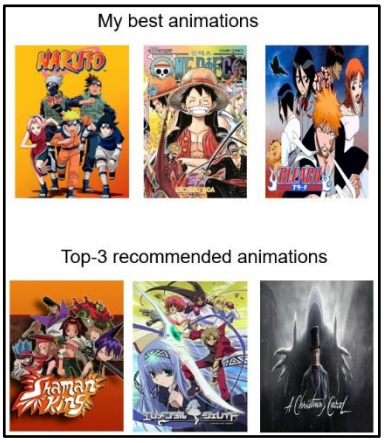
<Figure.16 Personalization result as heat map & cosine similarity>

It can be seen that most of them are zero. Cosine similarity is calculated at 0.013, and Dissimilarity is 0.987. It can be said that the higher the Dissimilarity, the better our recommended model. However, since this is an analysis of some of the user data, reliability may not be great.



<Figure.17 Input animation recommend system>

Therefore, we will now conduct a qualitative evaluation. In order to make a qualitative evaluation, one more recommendation system was created to receive animation titles directly. This can be recommended not only for users on the data but also for actual you. For example, We enjoyed watching One Piece, Naruto, and Bleach the most, so We put them in input shown Figure 17.



<Figure.18 Result of recommendation for qualitative evaluation>

Here, One Piece, Naruto, and Bleach animations are as follows, and the top three recommended animations are as follows. When comparing these, it can be seen that similar genres are similar in action and adventure, and similar animations are recommended in the animation poster. Therefore, the performance of our recommendation system seems to be quite good.

III. PROCESS 2 – Collaborative (Item Based)

A. Data Preprocessing

We tried to preprocess the data to eliminate cases where we don't evaluate scores in the dataset we use. However, the amount of data we collected was so great that we found that it would take about 76 days, calculated by computer, to analyze all the data. So instead of using all the data, we decided to compare the three groups that evaluated animation. The three selected groups are as follows.

1. A group that rated 100 animations
2. A group that rated between 2000-2300 animations
3. Group that rated over 5000 animations

Based on our experience, we divided the data into a group that watched animation for 3-4 years and a group that watched about 20% or 50% of the total animation.

rating_matrix

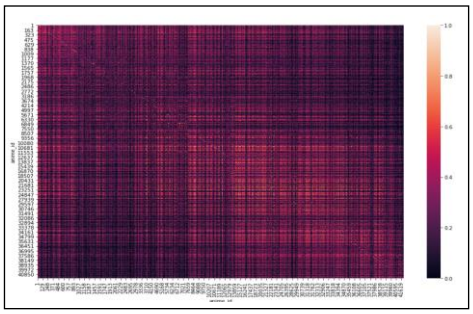
anime_id / user_id	1	5	6	7	8	15	16	17	18
1946	NaN	NaN	9.0	7.0	NaN	NaN	8.0	NaN	NaN
6783	7.0	8.0	5.0	NaN	NaN	NaN	0.0	NaN	9.0
8326	8.0	0.0	6.0	NaN	NaN	0.0	0.0	NaN	0.0
9528	10.0	10.0	10.0	NaN	NaN	8.0	10.0	0.0	10.0
9623	6.0	8.0	8.0	8.0	8.0	8.0	9.0	7.0	8.0

<Fig. 19. User-Item Rating matrix of Group 2>

If the User-Item Rating matrix is implemented for each selected group, there are many empty values as follows. All of these values are changed to 0 to calculate the similarity.

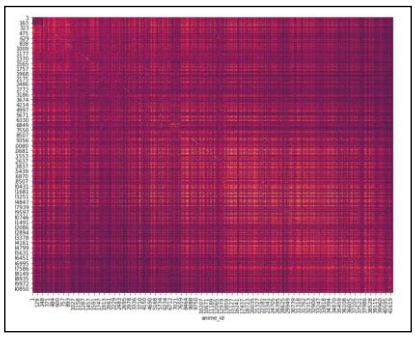
B. Calculate Item-Item Similarity

We implemented similarity based on cosine similarity, which is mainly used to calculate similarity.



<Fig. 20. Cosine similarity of Group 2>

And, knowing that cosine similarity has a disadvantage that it is difficult to consider scale, we calculated and compared the two types of cosine similarity that improved it. And it was decided to use the second improved cosine similarity, which showed better values after comparing the RMSE scores of the predictive matrix.



<Fig. 21. Adjusted Cosine similarity 1 matrix>

$$sim(i,i')_{ACOS} = \frac{\sum_{u \in U} \left(r_{u,i} - \bar{r}_i\right) \cdot \left(r_{u,i'} - \bar{r}_{i'}\right)}{\sqrt{\left(r_{u,i} - \bar{r}_i\right)^2} \cdot \sqrt{\left(r_{u,i'} - \bar{r}_{i'}\right)^2}}$$

<Fig. 22. Adjusted Cosine similarity equation [3]>

<Fig. 23. Adjusted Cosine similarity 2 matrix [1][3]>

$$\text{sumDiffer}(u_a, u_b) = \sqrt{\sum_{i \in I_{a,b}} (R_{a,i} - R_{b,i})^2 / M}$$

$$\omega(u_a, u_b) = \lambda^{\text{sumDiffer}(u_a, u_b)}, \quad 0 < \lambda < 1$$

$$\text{Imp\_sim}(u_a, u_b) = \text{sim}(u_a, u_b) \bullet \omega(u_a, u_b)$$

Fig. 24. Adjusted Cosine similarity 2 equation

Fig. 24 is an expression for the User-User Rating matrix, but in this expression, User and Item are calculated as interchangeable.

C. Predict

$$\widehat{R_{u,i}} = \sum^{N}(S_{i,N} * R_{u,N}) \,/\, \sum^{N}(|S_{i,N}|)$$

<Fig. 25. Personalized prediction rating equation [4]>

```
def predict_top_n(ratings_arr, item_sim_arr, n=20):
    pred = np.zeros(ratings_arr.shape)
    for col in range(ratings_arr.shape[1]):
        top_n_item = [np.argsort(item_sim_arr[:,col][:-
        for row in range(ratings_arr.shape[0]):
            pred[row,col] = item_sim_arr[col,:][top_n_i
            pred[row,col] /= np.sum(np.abs(item_sim_arr
    return pred
```

<Fig. 26. Personalized prediction rating equation Code>

We implemented the prediction matrix by implementing python code based on the Personalized prediction rating equation. (N=20) And implemented code that recommends animations that the user has not watched.

```
def get_unseen_ani(rating_matrix,userId):
    user_rating = rating_matrix.loc[userId,:]
    already_seen = user_rating[user_rating>0].index.tolist()
    ani_list = rating_matrix.columns.tolist()
    unseen_list=[anime for anime in ani_list if anime not in already_seen]

    return unseen_list
```

```
def recomm_anime(pred, userId, unseen_list, top_n=10):
    recomm_anime = pred.loc[userId, unseen_list].sort_values(ascending=False)[:top_n]
    return recomm_anime

unseen_list = get_unseen_ani(rating_matrix, user_id[3])

recomm_anime = recomm_anime(ratings_pred_matrix, user_id[3], unseen_list, top_n = 10)
```

```
title = []
ani2 = ani.set_index('MAL_ID')
for i in recomm_anime.index:
    title.append(ani2['Name'][i])
anime_recomm = pd.DataFrame(data = recomm_anime.values, index=title, columns=['pred_score'])
anime_recomm
```

Fig. 27. Animation recommendation code that users do not see

D. RESULT

The accuracy of the results could not be confirmed by checking the recommended list of movies that users have not seen because the number of movies they have seen is too large. Therefore, we chose the RMSE method for measurement. And as a result of comparing the values of RMSE for each group, the following graph was obtained.

It can be seen that the more the group watched the animation, the lower the RMSE level. However, I was able to confirm that the RMSE value was at least 2 or higher. This is a fatal error for grade scores ranging from 0-10.

An analysis of the causes revealed that the number of unwatched films was greater than the number of films watched, and the total number of films was too large. As a result, when selecting 20 with high similarity for prediction, a significant number of movies that users did not watch were selected. Therefore, this part was counted as 0, resulting in significant measurement error.

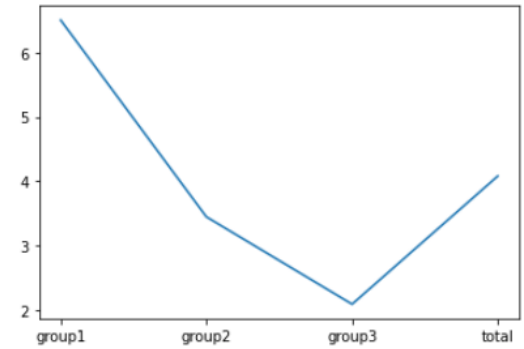


Fig. 28. RMSE Scores

IV. REGRETTABLE POINT

There are several regrettable points.

1. Try Jaccard or Pearson or any other similarity in the reference.

Through several papers on similarity that we saw in [3], we found that there are countless methods for calculating similarity, and that they are being improved to eliminate the shortcomings of the existing methods. This time I used similarity based on cosine similarity, but I'd like to try additional other methods as well.

2. Measure it using all rating data.

It's a pity that we didn't have time to utilize all the data. We felt that this part was disappointing because the number of movie data used for each data was different as the data was only partially extracted. (since movies not seen in the group were omitted from the data)

3. Comparison of user-based collaborative filtering rather than item-based collaborative filtering.

It would be better if the accuracy was compared using two similarities(item-item, user-user) of the same data. And I think that better results might have been obtained if the accuracy was measured by linking the two.

4. Apply by changing the prediction method.

The problem with the current method is that the predicted value is calculated as 0 if the user did not watch the animation included in the top 20 similarity. At this time, if a non-zero user rating average was provided or if the error rate was predicted based on the error of the actual values of the rating average, the error rate would have been less.

V. Reference

[1] An improved collaborative recommendation algorithm based on optimized user similarity(2016)

[2] https://en.wikipedia.org/wiki/Collaborative_filtering#References

[3] https://data-science-hi.tistory.com/150

[4] https://tpwkcorqhd.tistory.com/37

[5] https://techblog-history-younghunjo1.tistory.com/117

[6]https://www.kaggle.com/hernan4444/anime-recommendation-database-2020

[7] Poonam B. ,R. M. Goudar-Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System International Journal of Computer Applications (0975 – 8887) Volume 110 – No. 4, January 2015