Animation recommender using content-based filtering

Sujin Jeon
School of energy and chemical engineering
UNIST
jeonsujin37@unist.ac.kr

Yeongho Lee
School of Industrial Engineering
UNIST
lyh1030@unist.ac.kr

Youngbin Lee
School of Industrial Engineering
UNIST
young@unist.ac.kr

Abstract

In this study, a recommender system was developed to predict user ratings on animations. Content-based methods are used on animations dataset. Two representative methods for content-based filtering, vector space model and regression model, are conducted on the same dataset and their performances are compared. To get the best results on each model, different sets of features are used. Frequency matrix of genres using Bag-of-words and TF-IDF weights of synopsis are used for vector space model for effective computation of cosine similarities between feature vectors. For regression model, XGBoost, meta features such as date, duration, and rank are used for better prediction. As a result of analysis, regression model outperformed vector space model. It seems that it is not sufficient to use only genres and synopsis for identifying similar contents, but the meta features may play more important role.

I. INTRODUCTIONS

The objective of this study is to develop the best content-based filtering method for recommending animations using meta features (e.g., rank, date, duration) and text descriptions (e.g., genre, synopsis) with rating data from the website 'myanimelist.net' which is one of the biggest community for reviewing Japanese animations.

Recommender system is a technique that is widely used in a variety of services and movie recommendation is one of the most popular application. It seeks to predict the rating or preference that a user would give to a content. In this study, recommender system is developed for animations.

There are two major types of recommender systems and we focused on content-based filtering. A content-based filtering predicts user rating on a content by identifying similar contents. It is expected to work well with the dataset we used. "Anime Recommendation Database 2020" on Kaggle contains information about 17,562 animations and more than 100 million ratings from 325,772 different users. Content-based filtering is conducted for each user and there will be sufficient amount of data for training and testing.

Two methods are used for animation recommendation. One is a vector space model that identifies similar animations using text decriptions to predict the rating. The other is a regression model (XGBoost) that predicts the rating with meta features and text descriptions. And the results of two methods are compared to see which works better.

II. LITERATURE REVIEW

A. Contents-based Filtering [1][2][3]

A content-based approach analyzes the contents of items directly, analyzes the similarity between items and items, or items and users' preferences, and recommends items to customers based on this method. Contents-based Filtering have the following characteristics:

It identifies preferred items through profile information entered by users, scores evaluated by users, or information generated based on past purchases, and measures the similarity between classified item categories and users' preferred items through pre-selected criteria. Finally, the item that corresponds to the highest similarity is recommended to the customer [4]. Due to the ease of analysis and the availability of metadata, this method is widely used to recommend text-based news or Internet articles as well as movies, music, and books.

While collaborative filtering requires evaluation score data from multiple people to find neighboring users with similar tastes to recommended customers, content-based approaches use their own historical purchase history or profile information to identify recommended customers' preferences [3].

(1) Use of Independent Information

This is known as a technique that can be useful in a lack of information from other users, as it requires only independent information from the customer to be recommended. However, if customer's past purchasing history requires customer to identify preferences, customer must have enough item evaluation scores to identify your preferences. It has the disadvantage that the performance of recommendations cannot be guaranteed if there is a lack of past purchasing history, and that it is impossible to implement a content-based approach recommendation system if both purchasing history and profile information do not exist [5][6].

(2) New Item

If a newly added item has not yet been purchased and a score does not exist, the item cannot be included in the recommendation list until someone gives it a score in the collaborative filtering, and this problem is called First rater [7]. If there are many items whose evaluation scores do not exist, collaborative filtering repeatedly recommends only items released in the past, and the number of items included in the recommendation list is also limited, which can reduce the user's recommendation satisfaction. However, in a content-based filtering, the properties between items are identified and assigned to categories, even if evaluation scores for items do not exist. Finally, the First rater problem does not arise because items in categories that are highly similar to the user's preferred item attributes are recommended to the user.

(3) Over Specialization

The content-based Filtering finds similar products based on the past purchasing history of customers who are recommended, so only products that are too similar to those purchased previously are recommended without reflecting other users' tastes or preferences. In other words, there is no opportunity to access items of new attributes, so the diversity of recommended products cannot be guaranteed, and this problem is called over specialization [8].

B. Content-Based Collaborative Filtering for News Topic Recommendation [9]

In this paper, they propose a Content-based Collaborative Filtering approach (CCF) to bring both Content-based Filtering and Collaborative Filtering approaches together. They found that combining the two is not an easy task, but the benefits of CCF are impressive. On one hand, CCF makes recommendations based on the rich contexts of the news. On the other hand, CCF collaboratively analyzes the scarce feedbacks from the long-tail users. They tailored this CCF approach for the news topic displaying on the Bing front page and demonstrated great gains in attracting users.

Some noisy contents lead to bad performance in the CCF approaches. For the CCF approach in this work, they use θ to measure the similarity between two pieces of news, and only the similar piece of news shall be used in the CCF methods. Similarity checking strategy as described would ensure no noisy contents would be used in the learning of the CCF models, although the computational complexity might be an issue.

C. Contents-based filtering for recommendation systems using multi-attribute networks. [10]

In this paper, they propose a novel CBF method that uses a multiattribute network to effectively reflect several attributes when calculating correlations to recommend items to users. In the network analysis, they measure the similarities between directly and indirectly linked items. Moreover, their proposed method employs centrality and clustering techniques to consider the mutual relationships among items, as well as determine the structural patterns of these interactions. This mechanism ensures that a variety of items are recommended to the user, which improves the performance. They compared the proposed approach with existing approaches using MovieLens data, and found that their approach outperformed existing methods in terms of accuracy and robustness. Their proposed method can address the sparsity problem and over-specialization problem that frequently affect recommender systems. Furthermore, the proposed method depends only on ratings data obtained from a user's own past information, and so it is not affected by the cold start problem.

D. Vector Space Model

Vector space model, which express documents, or some contents in vectors using tf-idf method, bag of words model and other various methods and measuring similarity, has been used for information retrieval until now since it was proposed.[17] It is also used for some recommendation system, especially contents-based filtering by expressing contents of products as vector and compare similarity.[18]

But after the vector space model is proposed, some limitations of assumptions also pointed out.[19]

For example, vector space model assumes each vector feature are orthogonal, which means each word are independent, but in some cases it is not true. For this limitation generalized vector space model [20] was proposed. Also, high dimensionality is another problem. One of strategies for that problem is randomly select features. Or PCA, SVD and other techniques can be used. However even if the model has these limitations, it has merit in that it is intuitive method, the heuristic model.

In information retrieval for some documents, it usually only considers the contents itself of the documents for their vector feature. But for recommendation system of movie, books and etc, contents may not only include the contents itself, but also the other features like genre, actor, writer etc can be regarded as contents. Therefore, combining different kind of features which may have high dimensionality and different importance for representation of document/contents is another limitation. Also, in that case measuring similarity procedure may have to be done in more sophisticated manner because of each features' different importance and characteristics like data type, range, and etc.

So, method of giving weight to feature with their importance was also proposed.[21] But in that research, weight was also based on the other user's information, by the collaborative social network. So, in this project, we considered simpler way of weighting features, which just find weights regardless of user-user information.

Therefore in this project, for Vector Space Model we vectorized two kind of feature which can be regarded with different importance for similarity measure. One is synopsis and the other is genre, which is usually regarded importantly for user preference. We tried to find the weight of importance for measuring similarity.

III. METHOD

For a content-based recommendation, two different methods are used. A dataset example for a single user is given the Table 1. There are animations that a user has seen, and each animation has its features and rating.

TABLE 1

Animation	Feature_1	Feature_2	Feature_3	Rating (predict)
A	5	3	0	10
B	8	0	0	5
C	0	0	2	9
D	0	0	2	9
E	0	0	2	9
F	0	0	2	5
...	...	...	...	...

A. Content-based recommendation

This is a method for recommending contents using information about both contents and user preferences. Unlike a collaborate filtering that does not require any information about the contents, it is more explainable when recommending contents to users because it gives recommendations based on similar contents identified. This method can measure similarities between contents in different ways.

B. Vector space model

This method identifies similar contents by calculating cosine similarities of feature vectors of contents. To use this method, you first need to vectorize features. For text description features like synopsis of a movie, TF-IDF (Term Frequency - Inverse Document Frequency) can be used to measure the importance of words, excluding the stop words such as 'a', 'the', 'on'. By using TF-IDF, each text description feature is represented by a real-valued vector of TF-IDF weights. Then, nearest neighbors of a content can be found by calculating cosine similarities between feature vectors. And the weighted average of ratings for nearest neighbors are used to predict a user's unseen content.

Cosine-Similarity [15]: Representing items as vectors on a coordinate space and measures angles between vectors and gives out their cosine value. This has merits in that it is efficient for sparse matrix. Vectors of two items with attributes are compared in cosine similarity function as follows:

FORMULA 1

$$u(c,s) = cos(\vec{w_c}, \vec{w_s}) = \frac{\vec{w_c} \cdot \vec{w_s}}{||\vec{w_c}|| \times ||\vec{w_s}||} =$$

$$= \frac{\sum_{i=1}^{K} \vec{w_{ic}} \vec{w_{is}}}{\sqrt{\sum_{i=1}^{K} w_{ic}^2} \sqrt{\sum_{i=1}^{K} w_{is}^2}}$$

The more similar two items are, smaller the angle between their vectors.

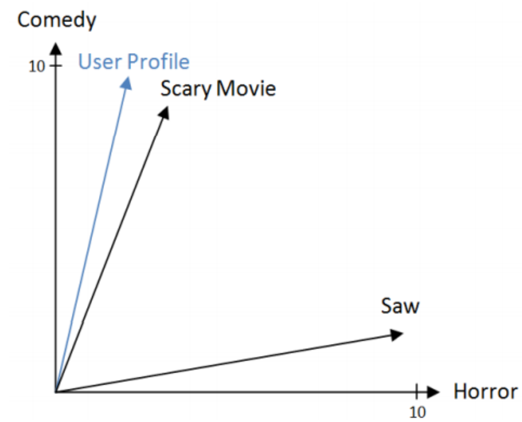

Fig. 1 Cosine Similarity in movie data.

Similarity search requires detailed information about the items. Better described items lead to more accurate recommendations [16].

C. Regression model

This method identifies similar contents by simply using a regression model. In other words, the model predicts rating for an unseen item based on regression using common features that are shared by contents. Thus, the process will be feature engineering, training the model, and make predictions on test set.

XGBoost [11]: Tree boosting is a highly effective and widely used machine learning method. In this paper, they describe a scalable end-to-end tree boosting system called XGBoost, which is used widely by data scientists to achieve state-of-the-art results on many machine learning challenges. They propose a novel sparsity-aware algorithm for sparse data and weighted quantile sketch for approximate tree learning. More importantly, they provide insights on cache access patterns, data compression and sharding to build a scalable tree boosting system. By combining these insights, XGBoost scales beyond billions of examples using far fewer resources than existing systems.

Therefore, in this proejct, content-based filtering product recommendation using XGBoost machine learning algorithm proposed to recommend items to a user based on item metadata [12][13][14].

IV. Application

A. Data preprocessing

There are raw data for animations and user ratings. Animations data has meta features about animations such as animation name, average score, genres, type, duration, date, synopsis, and so on. User ratings data has information about user preferences that who rated how much to which animation. For example, columns are user id, animation id, and rating (1~10). For analysis, these two different datasets are combined so that each user has its own dataset containing animation features and ratings that a user have seen. Before combining two datasets, users are filtered. We only use those who rated more than 5000 animations, so that the data is enough for training models.

In pre-processing, the feature 'Genres' is converted to a frequency matrix using Bag-of-Words, and 'Synopsis' are represented as TF-IDF. There are 36 features for 'Genre' and 100 features for 'Synopsis'. 'Synopsis' has much more feature originally, but it was limited to have only 100 features because of computations later in the vector space model. Also, we did not want the regression model to have too much high dimensions. As a result, the Fig 2 below shows the resulting dataset that is for vector space model. Here, only the text description features are used, which are 'Genres' and 'Synopsis'. Note that the Fig 2 is an example for a single user and other users have their own dataset like this which the same features.

Fig. 2 dataset example for vector space model.

Similarly, dataset is combined for regression model as well. The difference here is that meta features are included. There are 33 features including One-Hot-Encoded features for categorical features. And the Fig 3 below shows the resulting dataset for a single user.

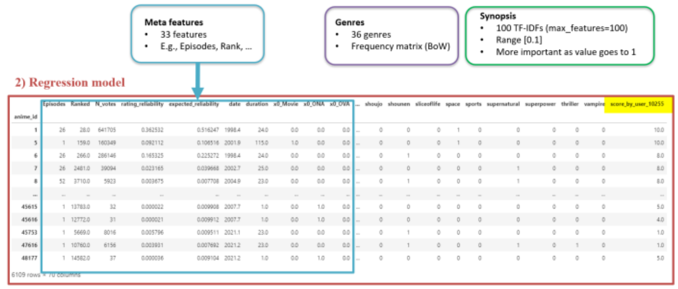

Fig. 3 dataset example for regression model.

B. Vector Space Model

This model was built with the assumption that user's preference to new animation will be close to similar animations' preference and that similarity is mostly determined by genre and synopsis.

Each users' data were split using train_test_split module in sklearn.model_selection package with test_size = 0.2. Each users' train set was used for finding better weight which represent importance in similarity measure for genre and synopsis. Candidates are (0.3, 0.7), (0.5, 0.5), (0.7, 0.3) for (genre, synopsis).

Higher weight for an attribute is regarded as higher importance of it in similarity measure. For example, if (0.3,0.7) is optimal weight, synopsis is more important than genre in similarity measure.

Weight was determined by comparing mean MSE value of real rating and predicted rating of all users' train set with each weight candidates and determine argmin as optimal weight. Mean MSE value of result is shown in TABLE 2.

TABLE 2
MEAN MSE VALUE FOR EACH CANDIDATE OF WEIGHT

Genre : Synopsis	0.3:0.7	0.5:0.5	0.7:0.3
MSE	2.2565	2.2278	2.3361

As a result, the weight candidate (0.5, 0.5) shown lowest mean MSE value so it was chosen for our weight. Actually giving this weight is just same with not giving weight because its ratio is 1:1. This weight is given by multiplying each weights to each attributes(genre, synopsis) vectors, which was generated with Bag-of-Words and TF-IDF method.

After giving weight the values of matrix become like below. Can see that each value in each attributes got weights, multiplied by 0.5 and 0.5.

adventure	cars	comedy	...	95	96	97
1	0	1	...	0.000000	0.000000	0.219417
0	0	0	...	0.000000	0.000000	0.000000
1	0	1	...	0.000000	0.000000	0.000000
0	0	0	...	0.294035	0.000000	0.250298

adventure	cars	comedy	...	95	96	97
0.5	0.0	0.5	...	0.000000	0.000000	0.109708
0.0	0.0	0.0	...	0.000000	0.000000	0.000000
0.5	0.0	0.5	...	0.000000	0.000000	0.000000
0.0	0.0	0.0	...	0.147018	0.000000	0.125149

Fig. 4 user matrix given weight.

Another hyperparameter value k, which is number of nearest animations for rating prediction, was determined as 10 by intuition.

With this weight and k, predicted ratings of animations in each users' test set. For prediction, first calculated cosine similarity of target animation and other already seen animations and based on that values found top 10 similar animations. Here, actually target animations are also already seen animations for each users, but for evaluation regarded as unseen animations and each animations in test set was excluded for measuring similarity each other. After finding 10 nearest animations, predict rating with weighted average of that 10 animations' rating. Here the weight was calculated with formula below.

FORMULA 2

$$\hat{r_t} = \sum_{i \in A} w_i r_i$$

$$w_i = \frac{s(a_t, a_i)}{\sum_{j \in A} s(a_t, a_j)}$$

$\hat{r_t}$ is estimated rating of target animation.
A is target's 10 similar animations' set.
s(a,b) is cosine similarity of a and b.
a_i is vector of animation i.

C. Regression Model (XGBoost)

We changed all datasets to numerical data to use XGBregressor. Source and genre features matrix and synopsis matrix used as a feature of the XGB model. In this model, our y-value is each user's rating, so we will use XGBregressor to predict it. When training the XGBoost, they are 3 types of parameter general, boost, learning task. We should choose best parameters to get well training model. Used the OPTUNA[22] to tune the best parameters.

```
params = {
    'n_estimators': 100,
    'max_depth': 24,
    'learning_rate': 0.09905592273886195,
    'subsample': 0.8704369112806065,
    'colsample_bytree': 0.9932309296458037,
    'objective': 'reg:squarederror',
    'gamma': 7,
    "eval_metric" : 'rmse',
    'seed': 2021,
    'tree_method': 'gpu_hist'
    }
```

Fig. 5 Hyperparameters in XGB Model.

And then use tuned parameters and train the model. When we ploted 10 features that influenced the prediction, and we can see that 'Ranked' feature was the most important. Also, we could see that the features we created had a huge impact on prediction.

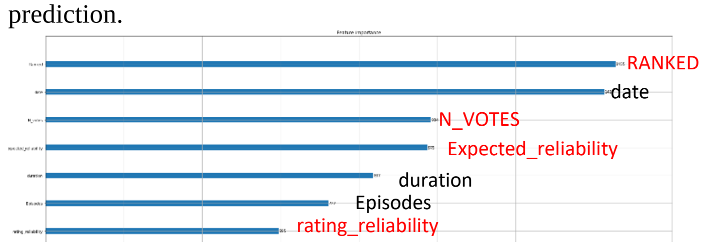

Fig. 6 Features Importance in XGB Model.

V. Results and Discussion

After finishing prediction, got MSE, RMSE, MAE value for evaluation, and got mean and standard deviation value of all users.

A. Vector Space Model

TABLE 3
MEAN AND STANDARD DEVIATION OF MSE, RMSE AND MAE

	MEAN	STD
MSE	2.3411	1.5491
RMSE	1.3799	0.6611
MAE	1.0907	0.5489

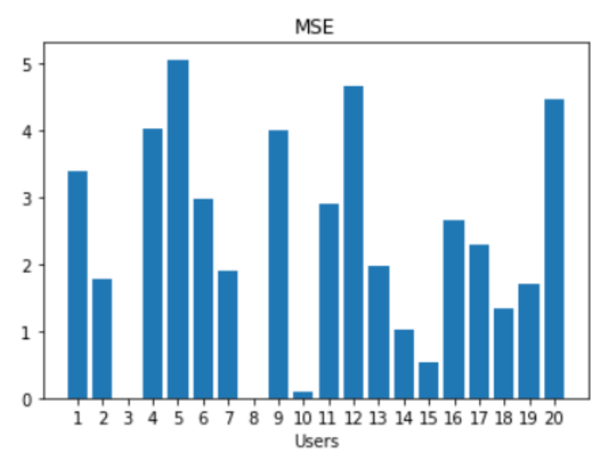

Fig. 7 MSE of all users in Vector Space Model.

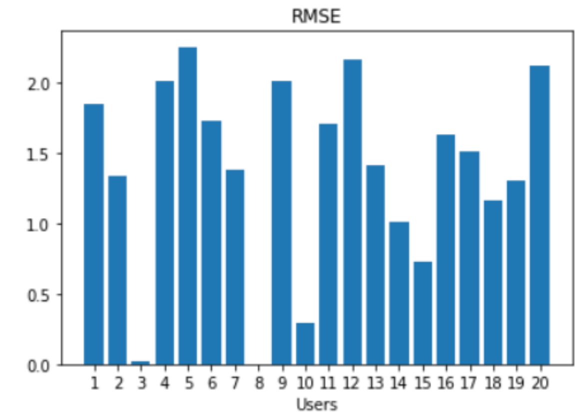

Fig. 8 RMSE of all users in Vector Space Model.

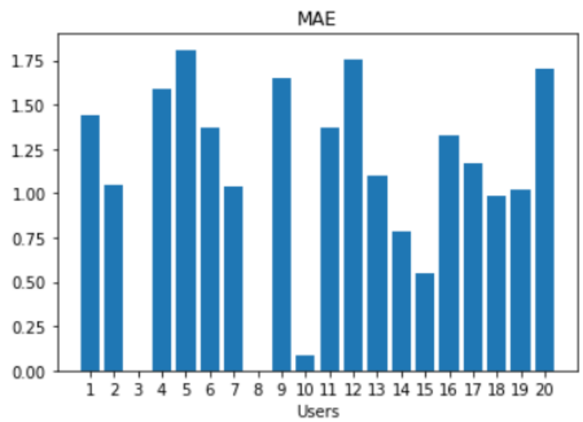

Fig. 9 MAE of all users in Vector Space Model.

Rating range is between 1~10. Therefore can think that the RMSE, MAE value is quite big for rating prediction, which means this model performed generally not good. Also there's big variance between users, which means the performance of this model may depending on users' characteristic.

As we assumed that genre and synopsis are most representative attributes for similarity measure and most of users will prefer similar animations to their previously preferred one, the result can be understood as except for some users, this assumption is not true and their preference is not highly affected by the genre/synopsis of animation.

B. Regression Model (XGBoost)

After training the XGB model, predict the results about each user. The Fig 11 below is evaluation metrics(MAE,RMSE) obtained from the XGB model.

	length	explained_variance_score	r2_score	MSE	RMSE	RMSLE
user_10255	6109	0.425292	0.425271	2.481537e+00	1.575290	0.291548
user_38143	5527	0.470044	0.469755	1.327218e+00	1.152049	0.156620
user_57684	5421	1.000000	0.000000	8.645884e-07	0.000930	0.000085
user_64807	5396	0.396844	0.395606	3.143819e+00	1.773082	0.335699
user_68042	10005	0.001036	0.000965	4.954098e+00	2.225780	0.419452
user_71931	5431	0.436715	0.435585	2.016030e+00	1.419870	0.275604
user_107650	7152	0.451997	0.451608	1.613498e+00	1.270235	0.306249
user_162615	10902	1.000000	0.000000	7.914878e-08	0.000281	0.000026
user_182280	6915	0.437491	0.437491	2.748271e+00	1.657791	0.352705
user_189037	11155	0.032947	0.032843	7.411416e-02	0.272239	0.041249
user_190748	5615	0.239602	0.238572	2.662431e+00	1.631696	0.224717
user_259790	7924	0.398619	0.398341	3.304818e+00	1.817916	0.357714
user_276953	5157	0.431963	0.431316	1.325122e+00	1.151139	0.196041
user_277841	6590	0.442279	0.441270	6.838164e-01	0.826932	0.136276
user_283786	9712	0.403905	0.402969	4.094643e-01	0.639894	0.081999
user_291207	7011	0.524727	0.524488	1.584389e+00	1.258725	0.240673
user_300428	6098	0.651927	0.651765	1.093353e+00	1.045635	0.247904
user_312302	5665	-0.017065	-0.017095	1.274061e+00	1.128743	0.120088
user_328195	5644	0.537783	0.537553	1.198408e+00	1.094718	0.187122
user_336459	7276	0.219143	0.217053	3.595151e+00	1.896088	0.361083

Fig. 10 Results in XGB Model.

C. Comparison of two methods

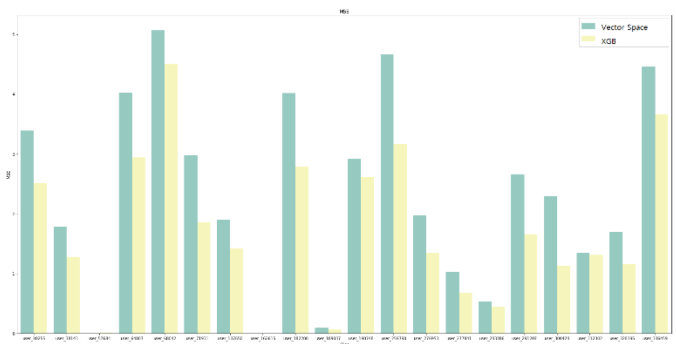

Fig. 11 MSE of all users in Vector Space Model and XGB.

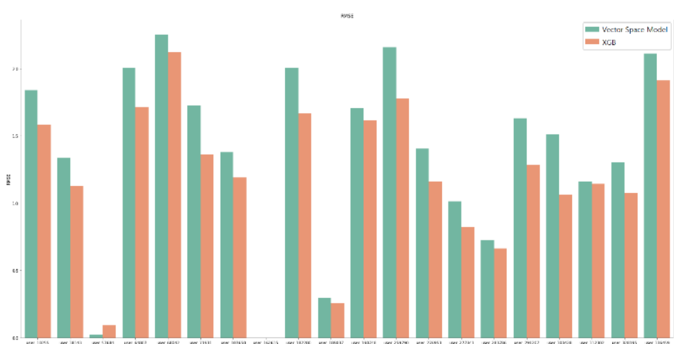

Fig. 12 RMSE of all users in Vector Space Model and XGB.

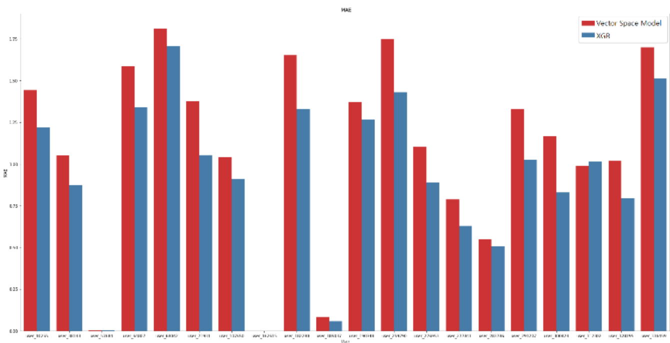

Fig. 13 MAE of all users in Vector Space Model and XGB.

Can see that generally XGBoost performs better than Vector Space Model.

It is because the Vector Space Model uses only genre and synopsis, while the XGB uses all features. Especially, the feature that most influenced the results in XGBoost model is 'Ranked' feature, so Vector Space Model which didn't consider that feature may performed bad.

And because among about 28,000 words in all synopsis data, only considered 100 words for similarity measure, some important words for some animations may not be included, which made synopsis attributes useless for similarity measure and lower the performance of Vector Space Model.

Also as can see in Fig.14, some extremely low errors were caused by some users of extreme cases, who rated all the animations as 10, or except 2 or 3 animations rated all 10.

	score_by_user_162615
mean	10.0
std	0.0

	score_by_user_57684
mean	9.999447
std	0.040746

score_by_user_57684	
10.0	5420
7.0	1

Fig. 14

One of the reason of big variance between users in performance might be the difference of number of rated animations among users. Usually the model performed well for users who rated many animations, so had bigger train set for XGBoost model.

VI. Conclusion

Can conclude that XGBoost model, the Ensemble model perform better than the heuristic model, the Vector Space model. Using only genres and synopsis for identifying similar contents seems not sufficient, but the meta features may play more important role.

In Vector Space Model, there were computation limitation. For running model for one user, it took about 1 minute, which is too long time for real world application.

Also for improvement, more words can be considered for synopsis feature and it can make the synopsis' features have more impact on similarity measure, and change the optimal weight. Also more candidates for weight between genre and synopsis can be considered. And clustering or specifying users whose preference are more likely to be affected by genre, synopsis or other contents of animations and use Vector Space Model only for that kind of users can be tried.

Also, because in XGBoost Model the 'Ranked' feature had highest feature importance, it can be understood that the rating trend of users to specific animation are similar to that animations' average preference. So to get more precise relative preference of one user to specific animations, procedure of adjusting the rating of each animation from users by subtracting the average rating of all users and then also subtracting the average of that user's rating can be considered.

We tried recommendation system for animations which have some special attributes like source, type of distribution and etc, and also have users kind of maniac, so domain knowledge and users' characteristics exploration may important.

References

[1] Melville, Prem, and Vikas Sindhwani. "Recommender systems." Encyclopedia of machine learning 1 (2010): 829-838.
[2] Asanov, Daniar. "Algorithms and methods in recommender systems." Berlin Institute of Technology, Berlin, Germany (2011).
[3] Pazzani, Michael J., and Daniel Billsus. "Content-based recommendation systems." The adaptive web. Springer, Berlin, Heidelberg, 2007. 325-341.
[4] Lang, Ken. "Newsweeder: Learning to filter netnews." Machine Learning Proceedings 1995. Morgan Kaufmann, 1995. 331-339.
[5] Bobadilla, Jesús, et al. "Recommender systems survey." Knowledge-based systems 46 (2013): 109-132.
[6] Khusro, Shah, Zafar Ali, and Irfan Ullah. "Recommender systems: issues, challenges, and research opportunities." Information Science and Applications (ICISA) 2016. Springer, Singapore, 2016. 1179-1189.
[7] Moreno, María N., et al. "Mining semantic data for solving first-rater and cold-start problems in recommender systems." Proceedings of the 15th Symposium on International Database Engineering & Applications. 2011.
[8] Balabanović, Marko, and Yoav Shoham. "Fab: content-based, collaborative recommendation." Communications of the ACM 40.3 (1997): 66-72.
[9] Lu, Zhongqi, et al. "Content-based collaborative filtering for news topic recommendation." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 29. No. 1. 2015.
[10]Son, Jieun, and Seoung Bum Kim. "Content-based filtering for recommendation systems using multiattribute networks." Expert Systems with Applications 89 (2017): 404-412.
[11]Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.
[12]Shahbazi, Zeinab, and Yung-Cheol Byun. "Product Recommendation Based on Content-based Filtering Using XGBoost Classifier." Int. J. Adv. Sci. Technol 29 (2019): 6979-6988.
[13]Baraniak, Katarzyna. "ISMIS 2017 Data Mining Competition: Trading Based on Recommendations-XGBoost Approach with Feature Engineering." Intelligent Methods and Big Data in Industrial Applications. Springer, Cham, 2019. 145-154.
[14]Li, Wenqiang, et al. "Meta-learning strategy based on user preferences and a machine recommendation system for real-time cooling load and COP forecasting." Applied Energy 270 (2020): 115144.
[15] Asanov, Daniar. "Algorithms and methods in recommender systems." *Berlin Institute of Technology, Berlin, Germany* (2011).
[16] Schwarz, Mykhaylo, Mykhaylo Lobur, and Yuriy Stekh. "Analysis of the effectiveness of similarity measures for recommender systems." *2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*. IEEE, 2017.

[17] Salton, Gerard, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing." *Communications of the ACM* 18.11 (1975): 613-620.

[18] Van Meteren, Robin, and Maarten Van Someren. "Using content-based filtering for recommendation." Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop. Vol. 30. 2000.

[19] Raghavan, Vijay V., and SK Michael Wong. "A critical analysis of vector space model for information retrieval." Journal of the American Society for information Science 37.5 (1986): 279-287.

[20] S. K. M. Wong, Wojciech Ziarko, Patrick C. N. Wong, Generalized vector spaces model in information retrieval, (1985)

[21] Debnath, Souvik, Niloy Ganguly, and Pabitra Mitra. "Feature weighting in content based recommendation system using social network analysis." Proceedings of the 17th international conference on World Wide Web. 2008.

[22] Akiba, Takuya, et al. "Optuna: A next-generation hyperparameter optimization framework." Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019.