Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 245 (2024) 860–868

Procedia Computer Science

www.elsevier.com/locate/procedia

9th International Conference on Computer Science and Computational Intelligence 2024 (ICCSCI 2024)

Movie Recommendation System: A Comparison of Content-Based and Collaborative Filtering

Hans Hendersen Kurniawan[a], William Susanto Lukman[a]*, Renaldy Fredyan[a], Muhammad Amien Ibrahim[a]

[a]*Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia*

Abstract

Watching movies has been our source of entertainment for decades. In this modern world, watching movies has never been more convenient with the existence of video streaming services. This convenience also comes with new problems because there are a lot of movies and series on the platform that exploring it on our own will take time and effort. To solve that problem, companies use recommendation systems to improve user experience. In the field of recommendation systems, there are two famous approaches, content-based filtering, and collaborative filtering. This research aims to compare both methods and find the best possible method to use in a video streaming service platform. We found that the difference between the two is located on the data that they need for the model to work properly. Content-based recommendation systems have higher accuracy when using a limited amount of data while collaborative filtering needs more data about user's behavior. Other than that, both methods give different recommendations because content-based only gives recommendations that are like the content that the user likes while collaborative filtering recommends contents that other users like.

Keywords: Recommendation System; Content-based; Collaborative Filtering.

* Corresponding author.
E-mail address: williamsusantolukman@gmail.com

10.1016/j.procs.2024.10.313

1. Introduction

Recently, a lot of video streaming services such as Netflix, YouTube and Amazon Prime Video have grown rapidly [1]. There are a large selection of movies and series on these platforms and because of that, consumers usually have a difficult time choosing a show to watch that matches their taste. Finding shows that are similar to their favorite show can be a hustle and can cause them to lose interest in watching more shows.

Regarding the statement above, video streaming services want to solve the problem by using recommendation systems to improve user experience and increase watch time. The challenge for designing a recommendation system is making the right recommendations to the right user to help consumers explore the huge amount of shows in their library.

There are several approaches to make a recommendation system. One of them is content-based filtering [2], in which users are recommended shows that are similar to the shows that they had watched before. The problem is that the system always gives similar recommendations for the user and because of that, they don't discover any of the other genres that they might like.

Apart from that, there is collaborative filtering approach where recommendation systems use the technique of matching users with similar interests and making recommendations from that basis. This technique requires a lot of user demographic information. The problem is for new users, they don't have a watch history and because of that, the system has nothing to compare it to. For new movies, this problem also exists because there are no ratings yet. In this case, video streaming services usually recommend the most popular and highest rated shows to the user hoping that they will be interested in watching one.

In this paper, we conducted an experiment to compare both approaches for recommendation systems based on their metrics score. We also want to see the reasoning behind the recommendations that the system gives. We believe that the content-based approach is more appealing to the user because it recommends something that is special to the user and not seeing the user as a part of a group and thus make them feel special

2. Related Works

This section reviews literature about comparison methods related to our research in order to develop and provide methods to make a recommendation system for study.

2.1. Collaborative Filtering

A popular recommendation algorithm called collaborative filtering (CF) builds its forecasts and suggestions on the opinions or actions of other users on the network. This strategy is based on the fundamental idea that the opinions of other users can be chosen and combined in a way that reasonably predicts the choice of the current user. They make the intuitive assumption that if people concur on any item's relevance or quality, they will likely agree on other items as well. If a group of people shares the same interests as Mary, for instance, Mary is likely to enjoy the items on their list that she hasn't seen yet [3].

Correlation-rate, or Co-Rate, is what collaborative filtering is used to determine which user's feature is closest to all users in the database. Thus, a primary cause of important issues like scalability issues. The system will have more active users, which will lead to user similarity and longer processing times, which presents a problem for the designers. Therefore, in the case of a larger system, this solution can be realized by designing the system to divide users into groups prior to engaging in collaborative filtering, which will speed up the process of processing user similarity [3].

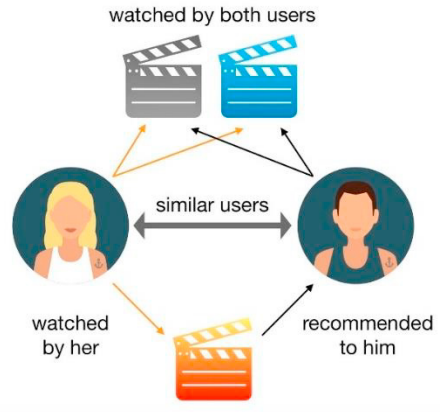

Fig. 1. Collaborative Filtering Illustration

2.2. Content Based

One of the most popular approaches to make a recommendation system is a content-based approach. The system recommends shows based on the user's movie preference characteristics. The data that is needed for this approach comes only from the item such as name, description or in this case, the genre of the movie [4].

From a psychological point of view, the user usually prefers this system because it makes them feel special [2]. Collaborative filtering usually treats the users as a group of people with the same interests and gives that group of people similar recommendations. Content-based on the other hand, makes the recommendation based on every user's specific taste in movies and it implies that this recommendation is for this specific user only [2]. That way, the user will feel as if this recommendation is more personalized and they tend to trust this approach more.

To determine whether to recommend the show or not, the system takes information from the user's watch history. Their favorite genre, their ratings of certain movies and all of the movie information on the platform is necessary for this approach to work. All of this information will be transformed into a uniform metric and the system will compare the result of the transformation [5].

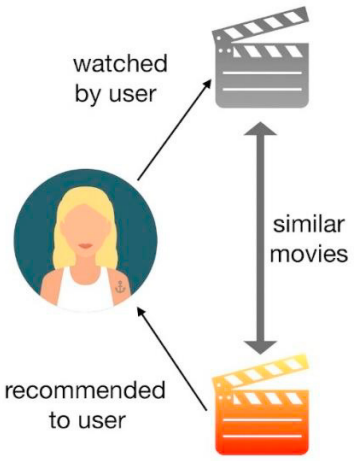

Fig. 2. Content-Based Illustration

3. Method

To answer the research hypothesis, a quantitative method is used which compares recommendation systems type: content-based vs collaborative filtering that provide high accuracy results.

3.1. Cosine Similarity

To make a recommendation system, building a machine learning model is necessary. There are a lot of machine learning models that can be used. For a content-based recommendation system, the model needs to assess the similarity between items or in this case, movies. That is because the content-based recommendation system compares the movies that the user has seen and rated before to other movies that the user hasn't rated [6].

In this project, the system will use cosine similarity to determine the recommendations it will give. Cosine similarity will measure the angle of cosine between two objects. It compares two objects on a normalized scale by finding the dot product of the two [6].

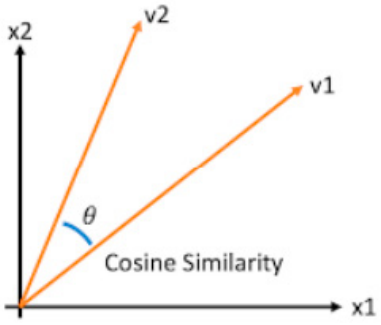

Fig. 3. Cosine Similarity Illustration

From fig. 3., it can be determined that the angle between v1 and v2 is relatively small. Using cosine similarity, the system will draw a conclusion that v1 and v2 are similar to each other because of the angle between them. If the angle between the two is larger, then they will be less similar to each other [6].

3.2. Word Embedding-Based

This section is a demonstration of the suggested similarity prediction approach. Key presumptions that have been made are as follows [7]:

- The extracted features from the movies can represent the movies [7].
- The history of the activities by the user (seen movies, movies similarity) can imply user preferences [7].

As a result, the similarity between movies may be found by measuring the similarity between features that have been extracted from movies. Considering that the films mi and mj share similarities, the following is the description of the formulation used to measure the similarity based on attributes extracted [7][10]:

$$Sim\ mi\ ,mj\ \equiv Tij,Gij,Dij,Aij,Pij$$

Where Tij,Gij, Dij, Aij , and Pij represent the similarity of features such as title, genre, directors, actors and plots of movie mi and movie mj, respectively. The methods for measuring the similarity of each feature are described as follows [7].

3.3. Dataset

The dataset that is used for this project is a dataset about movies. This dataset contains 6 different csv files about movies, such as title, date of release, budget, revenue, credits and many more. There are about 45.000 movie titles and more than 100.000 ratings from viewers.

Among those csv files, we only need 3 files that contain ratings, information about the movies and the credits. The ratings are for the recommender system, especially for the collaborative filtering approach. Other than that, ratings can also be used to recommend high rated movies to new users. The csv file containing film information will be used to see which movie is like the one that the user is watching. Movies that are directed by the same directors or cast the same actor can also be considered for recommendation by using the credits.

The format is .csv, which is easy to use because it doesn't need any other additional library to read the file. This dataset also has a CC0 1.0 universal license, which means that this dataset can be used without asking for permission and is a trustworthy source of information.

Table 1. Movies Dataset.

	Adult	Belongs_to_collection	Budget	Genres	Imdb_id	Orginal_language	Original_title	Overview
0	False	{'id': 10194, 'name': 'Toy Story Collection', …	30000000	[{'id': 16, 'name': 'Animation'}, {'id': 35, 'name': 'Comedy'}, {'id': 10751, 'name': 'Family'}]	tt0114709	en	Toy Story	Led by Woody, Andy's toys live happily in …
1	False		65000000	[{'id': 12, 'name': 'Adventure'}, {'id': 14, 'name': 'Fantasy'}, {'id': 10751, 'name': 'Family'}]	tt0113497	en	Jumanji	When siblings Judy and Peter discover an enchanted …
2	False	{'id': 119050, 'name': 'Grumpy Old Men Collection',…	0	[{'id': 10749, 'name': 'Romance'}, {'id': 35, 'name': 'Comedy'}]	tt0113228	en	Grumpier Old Men	A family wedding reignites the ancient feud …

Table 2. Ratings Dataset.

	UserId	MovieId	Rating	Timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185
4	1	1172	4.0	1260759205

Table 3. Credits Dataset.

	Cast	Crew	Id
0	[{‘cast_id:14,’ ‘character’: ‘Woody (voice)’, …	[{'credit_id': '52fe4284c3a36847f8024f49', 'department': 'Directing', …	862
1	[{'cast_id': 1, 'character': 'Alan Parrish', …	[{'credit_id': '52fe44bfc3a36847f80a7cd1', 'department': 'Production', …	8844
2	[{'cast_id': 2, 'character': 'Max Goldman', …	[{'credit_id': '52fe44779251416c91011acb', 'department': 'Directing', …	15602
3	[{'cast_id': 1, 'character': "Savannah 'Vannah' Jackson", …	[{'credit_id': '52fe44959251416c75039ed7', 'department': 'Sound', …	31357
4	[{'cast_id': 1, 'character': 'George Banks', …	[{'credit_id': '52fe4292c3a36847f802916d', 'department': 'Directing', …	11862

3.4. Evaluation Principle

To evaluate the quality of the predictions, you must compare these ranks or scores against some ground truth. These target values reflect the actual success of recommendations. We often refer to them as relevance scores. All of the available data was divided into training and test sets in chronological order. To train a model and forecast, for example, future transactions by a certain user, you need train data.

Let's say you created a set of predictions for each user by training the product e-commerce recommendation model using a portion of their viewing history. Subsequently, you may contrast the system's predicted user preferences for products with the actual transaction history of the items that users have purchased outside of the training set.

By logging significant user behaviors, you may capture the ground truth as it happens when you monitor live recommendation systems in production. How well an item fits a user profile or inquiry is reflected in its relevance. Relevant items for e-commerce sites are products you are probably going to purchase.

The output of the recommendation system is intended to predict what other items that the user will like. In this case, the evaluation metric that will be used is Root Mean Squared Error [8].

$$Root\ Mean\ Squared\ Error = \sqrt{\frac{\sum_{(u,i)\in T}(\hat{r}_{ui} - r_{ui})^2}{N}}$$

RMSE will calculate the difference between actual user ratings (r_{ui}) and the ratings that the system will predict ($\hat{r}_{ui}$) [8]. Thus, we can evaluate whether the system gives accurate recommendations or not. Other than RMSE, Mean Absolute Error is also recommended to use as an evaluation metric.

$$Mean\ Absolute\ Error = \frac{\sum_{(u,i)\in T}|\hat{r}_{ui} - r_{ui}|}{N}$$

The difference between the evaluation metrics is that all individual residuals in MAE are equally weighted while in RMSE, larger errors will have more penalty compared to smaller errors.

To measure the classification capability of the model, we will also use a classification metric which is accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy measures the percentage of correct classifications (true positive and true negative) when compared to all of the classifications that has been made (true positive, true negative, false positive, and false negative).

4. Results

After making the machine learning model and measuring the evaluation metric, here are the results of this experiment. Using Root Mean Squared Error as an evaluation metric, the results are:

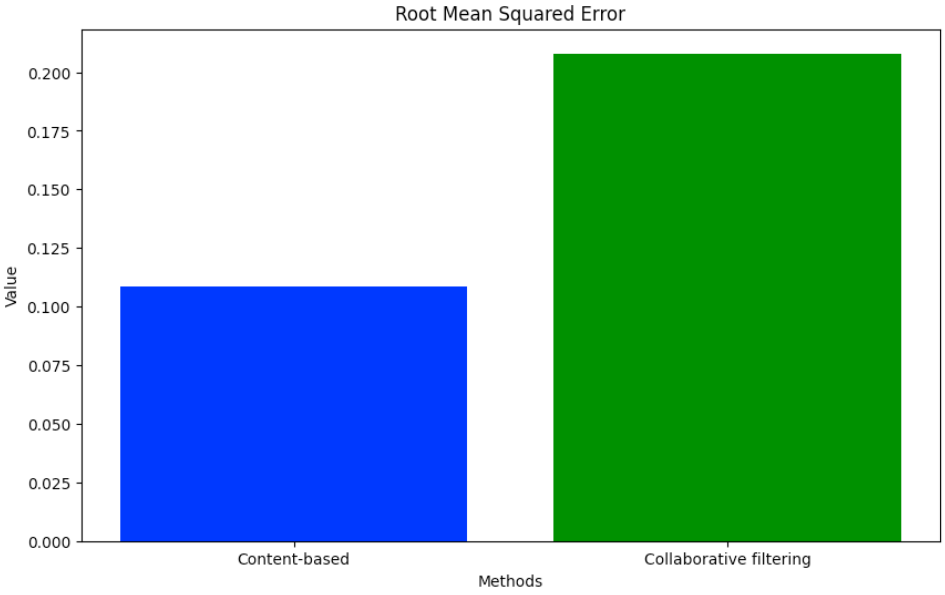

Fig. 4. Result of RMSE

Based on figure 4, content-based approach has half the RMSE score of collaborative filtering. That means that content-based can estimate the real value better than collaborative filtering. Using Mean Squared Error as an evaluation metric, the results are:

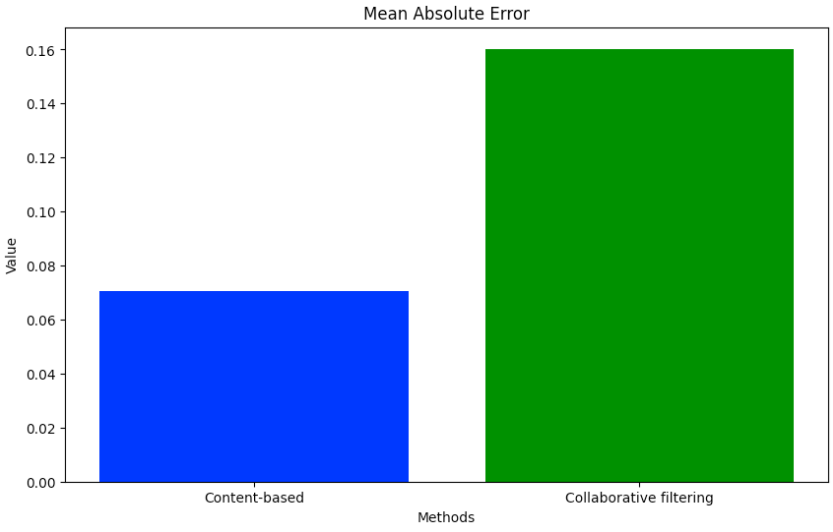

Fig. 5. Result of MAE

Based on figure 4, the result of the Mean Absolute Error test is almost identical to the Root Mean Squared Error with content-based having half the MAE score of collaborative filtering. This would also mean that content-based produces better estimation of the real value than collaborative filtering.

Using accuracy as an evaluation metric, the results are:

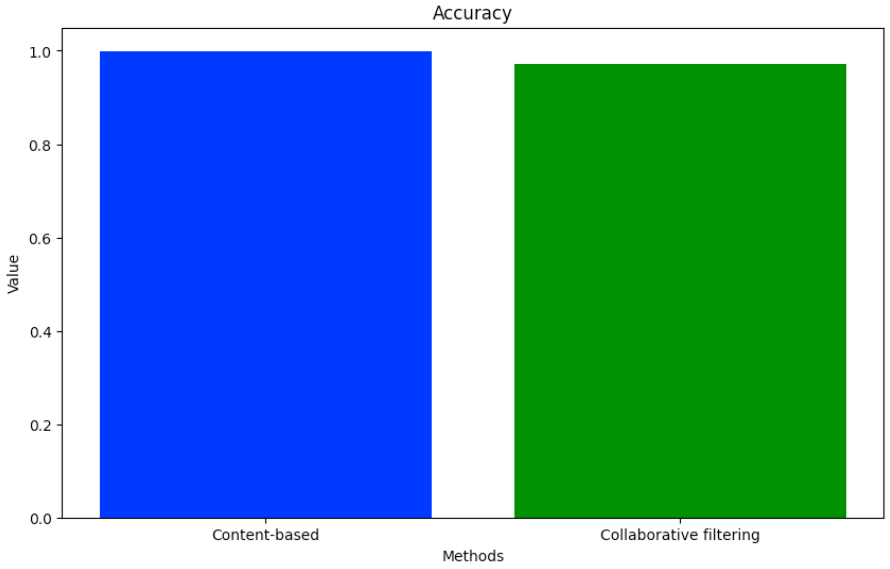

Fig. 6. Result of accuracy

Accuracy measures how many of the classifications a model makes are correct. There is a slight difference between content-based and collaborative filtering, but content-based still comes on top.

5. Discussion

Looking at the results of this experiment, content-based recommendation system is better in all the evaluation metrics. This experiment was run with very limited data and that may cause the underperformance of the collaborative filtering. Collaborative filtering needs a lot of data from users and the characteristics of the movies while content-based only needs data that are related to the movie itself. When your platform is brand new, we recommend that you use a content-based recommendation system. After your platform has developed, you should add a different recommendation by using collaborative filtering. That way, users won't get bored of the same recommendations and still have a recommendation that is accurate.

References

[1] Goyani, M., & Chaurasiya, N. (2020). A Review of Movie Recommendation System: Limitations, Survey and Challenges. Electronic Letters on Computer Vision and Image Analysis, 19(3), 18–37. https://doi.org/10.5565/rev/elcvia.1232

[2] Liao, M., Sundar, S. S., & Walther, J. B. (2022, April 29). User Trust in Recommendation Systems: A comparison of Content-Based, Collaborative and Demographic Filtering. Conference on Human Factors in Computing Systems - Proceedings. https://doi.org/10.1145/3491102.3501936

[3] Phorasim, P., & Yu, L. (2017). Movies recommendation system using collaborative filtering and k-means. International Journal of Advanced Computer Research, 7(29), 52–59. https://doi.org/10.19101/IJACR.2017.729004

[4] Pradeep, N., Rao Mangalore, K. K., Rajpal, B., Prasad, N., & Shastri, R. (n.d.). Content Based Movie Recommendation System. www.riejournal.com

[5] Abdul Hussien, F. T., Rahma, A. M. S., & Abdul Wahab, H. B. (2021). Recommendation Systems for E-commerce Systems An Overview. Journal of Physics: Conference Series, 1897(1). https://doi.org/10.1088/1742-6596/1897/1/012024

[6] Singh, R. H., Maurya, S., Tripathi, T., Narula, T., & Srivastav, G. (2020). Movie Recommendation System using Cosine Similarity and KNN. International Journal of Engineering and Advanced Technology, 9(5), 556–559. https://doi.org/10.35940/ijeat.E9666.069520

[7] Nguyen, L. V., Nguyen, T. H., & Jung, J. J. (2020). Content-Based Collaborative Filtering using Word Embedding: A Case Study on Movie Recommendation. ACM International Conference Proceeding Series, 96–100. https://doi.org/10.1145/3400286.3418253

[8] Avazpour, I., Pitakrat, T., Grunske, L., & Grundy, J. (2014). Dimensions and Metrics for Evaluating Recommendation Systems. Springer.

[9] Gupta, M., Thakkar, A., Aashish, Gupta, V., Rathore, D. P. S. (2020). Movie Recommender System Using Collaborative Filtering(ICESC 2020) : 02-04, July 2020.

[10] Anwar, T., & Uma, V. (2021). Comparative study of recommender system approaches and movie recommendation using collaborative filtering. International Journal of System Assurance Engineering and Management, 12(3), 426–436. https://doi.org/10.1007/s13198-021-01087-x

[11] Reddy, S., Nalluri, S., Kunisetti, S., Ashok, S., & Venkatesh, B. (2019). Content-based movie recommendation system using genre correlation. Smart Innovation, Systems and Technologies, 105, 391–397. https://doi.org/10.1007/978-981-13-1927-3_42

[12] Sujithra Alias Kanmani, R., Surendiran, B., & Ibrahim, S. P. S. (2021). Recency augmented hybrid collaborative movie recommendation system. International Journal of Information Technology (Singapore), 13(5), 1829–1836. https://doi.org/10.1007/s41870-021-00769-w

[13] Joseph, A., & Benjamin, J. (2022). Movie Recommendation System Using Content-Based Filtering And Cosine Similarity. (n.d.). https://doi.org/10.5281/zenodo.6791117

[14] Ahuja, R., Solanki, A., & Nayyar, A. (2019). Movie recommender system using k-means clustering and k-nearest neighbor. Proceedings of the 9th International Conference On Cloud Computing, Data Science and Engineering, Confluence 2019, 263–268. https://doi.org/10.1109/CONFLUENCE.2019.8776969

[15] Furtado, F., Singh A. (2020). Movie Recommendation System Using Machine Learning. https://doi.org/10.22105/riej.2020.226178.1128

[16] Ojokoh, B. A., & Aboluje, O. O. (2020). A collaborative content-based movie recommender system. In Int. J. Business Intelligence and Data Mining (Vol. 17, Issue 3).

[17] Davagdorj, K., Park, K. H., & Ryu, K. H. (2020). A Collaborative Filtering Recommendation System for Rating Prediction. Smart Innovation, Systems and Technologies, 156, 265–271. https://doi.org/10.1007/978-981-13-9714-1_29

[18] Salmani, S., & Kulkarni, S. (2021). Hybrid Movie Recommendation System Using Machine Learning. Proceedings - International Conference on Communication, Information and Computing Technology, ICCICT 2021. https://doi.org/10.1109/ICCICT50803.2021.9510058

[19] Fulzele, H., Bhoite, M., Kanfade, P., & Yadav, A. (2023). Movie Recommender System using Content Based and Collaborative Filtering. In International Journal of Innovative Science and Research Technology (Vol. 8, Issue 5). www.ijisrt.com

[20] Yin, L. J., Safar, N. Z. M., Kamaludin, H., Abdullah, N., Yusof, M. A. M., & Supriyanto, C. (2023). Adopting Machine Learning in Demographic Filtering for Movie Recommendation System. Journal of Soft Computing and Data Mining, 4(1), 1–12. https://doi.org/10.30880/jscdm.2023.04.01.001