Hybrid Collaborative Movie Recommendation System

Conference Paper · September 2023

DOI: 10.1109/AiDAS60501.2023.10284679

Ariff Md. Ab. Malik
Universiti Teknologi MARA

Shuzlina Abdul Rahman
Universiti Teknologi MARA

Tajul Rosli Razak
Universiti Teknologi MARA

Sharifalillah Nordin
Universiti Teknologi MARA

Hybrid Collaborative Movie Recommendation System

Mohamad Riduan Mas Husin
College of Computing, Informatics and Mathematics
Universiti Teknologi MARA
Shah Alam, Malaysia.
Email: mohamadriduanmashusin@gmail.com

Tajul Rosli Razak
Research Initiative Group of Intelligent Systems
College of Computing, Informatics and Mathematics
Universiti Teknologi MARA
Shah Alam, Malaysia.
Email: tajulrosli@uitm.edu.my

Ariff Md Ab Malik*
Research Initiative Group of Intelligent Systems
Faculty of Business and Management
UiTM Selangor
Puncak Alam, Selangor, Malaysia.
Email: ariff215@uitm.edu.my

Sharifalillah Nordin
Research Initiative Group of Intelligent Systems
College of Computing, Informatics and Mathematics
Universiti Teknologi MARA
Shah Alam, Malaysia.
Email: sharifalillah@uitm.edu.my

Shuzlina Abdul-Rahman
Research Initiative Group of Intelligent Systems
College of Computing, Informatics and Mathematics
Universiti Teknologi MARA
Shah Alam, Malaysia.
Email: shuzlina@uitm.edu.m

*Corresponding author

2023 4th International Conference on Artificial Intelligence and Data Sciences (AiDAS) | 979-8-3503-1843-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/AIDAS60501.2023.10284679

***Abstract*—A recommendation system applies the data to the information discovery techniques and personalises the products for recommendations. Typically, movie recommendation systems predict what movies a user wants based on the characteristics of previously liked movies. These recommendation systems are helpful for organisations that gather data from many users and wish to offer the best possible recommendations effectively. This study developed a hybrid movie recommendation system using collaborative and content-based filtering involving Matrix Factorisation, TF-IDF. In addition, this study aims to extract the CBF features into CF to improve the recommendation system engine's performance and personalise the user's movie recommendations. The performance of the trained model was measured using RMSE, precision, recall, and F1 score. The trained model produced a low RMSE value with high precision, recall, and F1 score values. The paper's contribution is that the proposed hybrid recommendation provides a pathway for the users by helping them find movies that meet their interests.**

***Keywords*—recommendation system; collaborative filtering; content-based filtering; hybrid; matrix factorisation**

I. Introduction

Movies have become a favorite of many individuals worldwide because of their different new and interesting genres that attract audiences to watch them [1]. The existence of streaming entertainment service companies offers movies and has become a flexible medium for the public to watch movies online or can be downloaded. A lot of great movies have been produced in various genres, whether classic or modern concepts, in satisfying audiences. From year to year, the production and number of movies have increased from all over the world. Movies are an entertainment platform that can help people to enjoy their day. Nowadays, movies can be easily accessed and watched by movie fans through online streaming services [1]. However, most of these companies are struggling to attract more potential movie fans to extensively use their online services and offer a more variety of movie options. Thus, it may cause losses to their businesses [2]. In the context of human beings, watching movies can be a healing process tool from daily life stress and can stimulate individual happiness and human relationship [3]. It is important to ensure the correct movies are to be watched by the right audiences according to their current emotion and psychology. From the perspective of these streaming companies, they could not attract movie fans to use their applications although they are offering an extensive number of movies to be watched. This is because their potential audiences have difficulties finding their best selection and suitable movies to be watched by them subject to their interest and mood [4]. While searching for the best movies to watch, they may confuse about which movies should they watch first, and this process may take too much time which may contribute to a lack of interest in them to continue watching the movies [1]. Some of them could lose interest in watching the movies because they are no longer interested to keep watching and become tired [5].

To improve this situation, a lot of recommendation systems have been developed and introduced [6]. These systems are developed with the purpose to help potential or existing audiences to find their searched items that can fulfill their interests. Several system methods have been used to make better item suggestions in satisfying the audience's interest, especially in the context of the movie recommendation system. Two methods were commonly used in previous studies, namely Content-Based Filtering (CBF) and Collaborative Filtering (CF). CBF is a method that works with information provided by the users either explicitly through the rating process or indirectly by clicking on a link. This method will create a profile based on the information given by the users can be used for developing an audience suggestion [7]. Meanwhile, CF is a method that can explore the similarities between other users to make a recommendation [8]. The core functions of the recommendation systems are i) analyzing user data, and ii) collecting valuable information for further predictions. Therefore, a recommendation system is an algorithm to be used for processing and recommending the related item options to the users [9][26]. This system can help users to find the information based on the user's preferences and personalize it subject to each user [10].

This paper presents a combination of two different algorithms which are CBF and CF to produce a movie recommendation system for movie fans in meeting their interest in the movies. This paper aims to extract the CBF's feature into CF to enhance the performance of the recommendation system engine to personalize the movies according to the user's preferences. The weakness of CBF is the movies that will be recommended will be the same and the contents may not surprise the users while for the CF, CF will have difficulties if the users do not rate any movies or the new users that do not rate any movies yet. Thus, this paper proposes to combine the strength of both CF and CBF to overcome both algorithms' weaknesses and be able to satisfy the users and the movies to able to meet the user's interest in the movies. This paper is organized as follows: Section 2 describes the research methodology; Section 3 presents the results and findings while Section 4 discusses the conclusion of the research.

II. RESEARCH METHODS

A. Data Preparation

The MovieLens dataset that is being used was gained from Kaggle. The GroupLens Research Project at the University of Minnesota collected MovieLens dataset. MovieLens dataset gained names as "ml-latest-small" which has 100836 ratings, 3683 tag applications, and 9742 movies. The dataset was created by 610 users which was all of users had rated at least 20 movies. The dataset does not have demographic information and the movie's genres are varied such as comedy, drama, and action. The dataset was cleaned by checking missing values and duplicate values using the Python function. Three different files were created for the recommendation process which are the movies file, rating file, and tags file. By using these files, two different data frames had been created, CF and CBF. For the movie file, three attributes were used which are i) *movie Id*, ii) *the title of the movies*, and iii) the *genres of the movies*. While for the rating file, only two attributes were used which are i) *movie Id* and ii) *rating of the movies*. Meanwhile, for the tag file, i) *movie Id*, and ii) *tag of the movie* were used. The CF data frame was created by combining the movie file and the rating file and used the *movie Id* attribute as a primary key. These two files were suitable because the CF engine made recommendations by finding similar users that have the same interest. The second data frame was created for CBF by combining the movie file and tag file and used the *movie Id* attribute as a primary key. These two files were chosen to be the second data frame because the CBF will make recommendations by finding the similarity of the movie's content.

B. Model Development

1) Identifying Keywords: In the development phase, a CBF engine was applied to find similarities in the content in producing recommendations. To help the engine find the similarity of content, a keyword column was created and added to the existing CBF data frame. The keyword column named 'related' was created and contains genres, movie title, tags, and movie released years. Some movies that had combined genres were separated before being inserted into related columns to make it easier for the CBF engine to find similar genres. An example of a combined genre is a romantic comedy. After separating it, the movie will have two separate genres which are romantic and comedy. Besides that, the movie's release year was extracted from the title column and added to the related column. Thus, the engine will be allowed to find movies that were released in the same year. The title of the movies also was added to the related column and assigned as a keyword. The main reason for adding the movie's title is because some of these movies had been produced in series. Then, these titles were separated word by word to use as several keywords for the similarity assessment. For instance, the title of the Toy Story movie will be separated word by word and become "Toy" and "Story". Next, the tags file was added to the related column. Before adding to the related column, the timestamp attribute will be removed. The tags were added into the data frame by using movie Id to match with the same movie Id that already exists in the data frame. A similar word separation process was used to allow the engine to find the similarity of the movies' title. All keywords in the related column were separated word by word and had transformed into small letters. These actions were done to help the CBF find the similarity of the keywords easier.

2) CBF Engine Development: Both TF-IDF and Cosine Similarity were used for building CBF Engine. TF-IDF technique was the first technique used to build CBF engine. The words in the keyword columns in the CBF data frame need to change into numbers which were known as text vectorization. After transforming the words into numbers, then machine learning can understand to continue the text analysis process. The TF-IDF is used to extract keywords and retrieve information from words. TF-IDF approach was focused on the related column that contains all the keywords for particular movies. The words with the highest scores are the most relevant, and hence can be considered keywords for that word. CBF always uses the TF-IDF method. In a collection of documents, the word relevant in the document will be evaluated by the TF-IDF as a statistical measurer, as in Equation (1). This is achieved by multiplying two metrics [12]. These metrics are:

- The number of times in a document that a word appears (TF).
- The word's inverse document frequency over a set of documents (IDF).

The result of this multiplication will represent the TF-IDF score of a word in a document. In a specific document, the higher the score, the more important the word is [12]. For instance, the use of TF-IDF to weigh the various terms that suit the question of "furry" which would be more relevant than "cat". Then "the furry kitten" can be selected as the best match eventually [13].

$$TF - IDF = TF \times IDF \quad (1)$$

The second technique used to develop the CBF engine is Cosine Similarity. This method is used for matching comparable movies based on the most common word counting. The Cosine Similarity represents the orientation (angle) of the words when plotted on a multi-dimensional space, where each dimension corresponds to a word. As result, the greater the resemblance, the smaller the angle is [14]. By using the keywords, the second engine will search the similar movies to recommend by applying Cosine Similarity as in Equation (2). Similar movies that match with the keywords will be recommended to users. Cosine Similarity is a metric used to determine the similarity, regardless of their size, the two items or documents are. It tests the cosine of an angle projected in multi-dimensional space between two vectors [15].

$$\text{Similarity } (p, q) = \cos\,\theta = \frac{p \cdot q}{\|p\|\|q\|} = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2}\sqrt{\sum_{i=1}^{n} q_i^2}} \quad (2)$$

Mathematically, the angle of the cosine between two vectors is determined from the point product of the two vectors divided by the magnitude product of the two vectors [15]. The output will always range from -1 to 1 because of the discovery of the cosine of two vectors, where -1 indicates that two objects are different and 1 indicates that two objects are completely similar [16].

3) Matrix Factorisation: The latent factor model is the foundation for most Matrix Factorisation (MF) models. Although certain research has employed a variety of dimensionality reduction strategies, the MF approaches have been proven to be the most accurate techniques in reducing the problem of high levels of sparsity in recommendation system databases [17]. The purpose of MF is to find the set of features for movies that determine how users rate each movie [18]. It can be done by predicting the influence of these features on the user's rating of the movie together with the number of features in each movie [19]. The Movie's features can be any elements such as 'violence', 'romance', 'comedy', 'sci-fi', 'slice of life', 'fantasy', or even abstract elements that may be unique to the movie. Table I shows the example of user feature preferences how the audiences rate the features' elements of the movies. In this example, User1 and User3 preferred romance and comedy movies by giving the highest rating as compared to the other features. While User2 and User4 highly rated 'sci-fi' and 'fantasy' elements for their preferred movies compared to romance and comedy movies.

TABLE I. EXAMPLE OF USER FEATURE PREFERENCES

User	Fantasy	Sci-fi	Romance	Comedy
User1	0	0	4	5
User2	5	4	0	0
User3	0	0	5	4
User4	5	5	0	1

Table II shows an example of movie feature elements, where the engine will be able to guess User2 and User 4 may favour "Avatar" and "Avenger" movies. On the other hand, User1 and User3 may prefer "Love and Basketball" and "Just go with it". This is because each user has his or her feature preferences that are rated for the preferrable movies.

TABLE II. EXAMPLE OF MOVIE FEATURE ELEMENTS

Movie	Fantasy	Sci-fi	Romance	Comedy
Love and Basketball	0	0	5	3
Just go with it	0	0	3	5
Avatar	5	3	0	0

Avengers	2	5	0	1

Table III shows an example of the users' movies ratings. From this table, "?" indicates that the movie had not been rated by the users either because they have not watched the movie or are just reluctant to provide feedback for it. A movie with the highest rating is one of the users' preferred movies or would like to watch and vice versa. The goal for this machine learning algorithm is to predict how the user would rate the movie they have not watched so that the movie recommendation engine can recommend it or not.

TABLE III. EXAMPLE OF USER'S MOVIE RATING

User	Love and Basketball	Just go with it	Avatar	Avenger
User 1	?	0	4	?
User 2	0	0	?	5
User 3	?	5	0	0
User 4	5	?	?	0

The CF uses MF to predict user movie ratings by predicting the numbers from Tables 1 and 2. The motivation for doing this is that the number of features was often smaller than the number of users and movies in a given dataset to produce a rating table or matrix. If Table 1 and 2 can be represented as the matrices P and Q, then the user's movie rating table, which is Table 3, can be derived by the multiplication of P and the transpose of Q. This can be written as "np. dot(P, Q.T)" in Python NumPy. Then, uses the result matrix to fill in the missing values in Table 3. The usage of MF is to predict the values of the matrices P and Q by initializing them with random numbers and compute P*Q(transpose). The result will be used to determine how much the real ratings are missing. This can be achieved by comparing the predicted ratings with the ratings from the training dataset. The prediction error will be used to determine the update of the matrices P and Q. This process will be repeated until the error is small enough.

4) Hybrid Collaborative Filtering (HCF) Engine: Fig. 1 shows the HCF engine's flow which is HCF engine is created by extracting CBF feature into CF engine using Python function. HCF engine provide recommendation using previous two engine's strength. The purpose of the third engine, the HCF engine, is to support each weakness of the engines by producing recommendations using the strength from each engine [20]. This engine can be set up by recommending movies that are rated at least 3.0 by users.

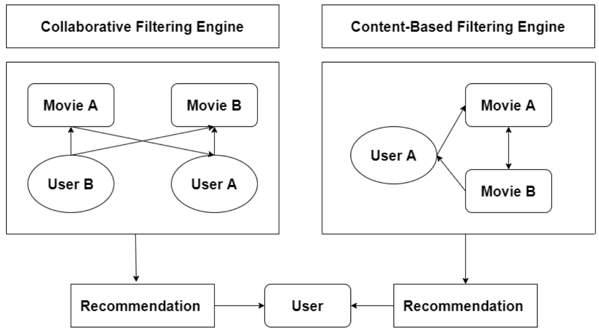

Fig. 1. HCF engine's flow

C. Testing & Documentation

This phase is the final phase where the functionality of the system will be checked. The system has to go through an adjustment process after the tested results were not satisfying the project requirement. Root Mean Square Error (RMSE), *precision*, *recall,* and *F1 score* will be applied to evaluate the performance of the system to adjust if the performance of the trained model is not satisfying. Each conducted process and step need to be recorded and documented. RMSE, as in Equation (3), shows the model's absolute fit to the data, in determining how close the data points observed are to the expected values of the model. Lower RMSE values suggest a better match. RMSE is a good measure of how accurately the model predicts the response [21].

$$RMSE = \sqrt{\overline{(f - o)^2}} \quad (3)$$

Where:

- *f* = forecasts (expected values or unknown results)
- = observed values (known results).

The standard deviation of unexplained variance can be calculated using RMSE, and it has the advantage of being in the same units as the response variable [21]. Residuals are a measure of how far the data points are from the regression line where RMSE is a measurement of how spread out these residuals are. In other words, it indicates how tightly the data is clustered around the line of best fit. In climatology, forecasting, and regression analysis, root mean square error is widely used to check experimental results. The proportion of data points of our model claims are relevant that can be referred as *precision* [22]. *Precision* score gained after the division of True Positive (TP) and the addition of True Positive (TP) and False Positive (FP) as explained in Equation (4).

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

The ability to locate all relevant instances in a dataset is referred to as recall [22]. Recall score gained after the division of True Positive (TP) and the addition of True Positive (TP) and False Negative (FN), as Equation (5).

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

The *F1-score* is a metric for how accurate a model is on a given dataset and can be defined as the harmonic mean of the model's precision and recall. It is a technique of combining the model's *precision* and *recall*. *F1 score* can be calculated as explained in Equation (6).

$$F1 = 2\left(\frac{Precision*Recall}{Precision+Recall}\right) \tag{6}$$

III. Results and Discussion

This section presents the result of the study by first explaining the Collaborative Filtering (CF) engine, followed by the Content-Based Filtering (CBF) engine and lastly the Hybrid Collaborative Filtering (HCF) engine.

A. Collaborative Filtering (CF) Engine

Table IV shows the sample output of the CF engine. By inserting *user Id*, the movies were recommended to users based on previous movies that had been watched and rated with a high rating. Then the CF engine will find similar users that had rated similar movies with a high rating for a recommendation.

TABLE IV. CF ENGINE'S OUTPUT

User	Movies Recommendation
User1	Mickey's The Prince and the Pauper (1990)
	Red 2 (2013)
	Last Song, The (2010)
	Train to Busan (2016)
	Miss Potter (2006)
	We Don't Live Here Anymore (2004)
	Waterdance, The (1992)
	Pat Garrett and Billy the Kid (1973)
	Snowden (2016)
	Endless Summer, The (1966)

Fig. 2. CF engine's output

B. Content-Based Filtering (CBF) Engine

Table V shows the result of the CBF engine. For the first recommendation, the movie's title 'Toy Story' that was inserted as the keyword to find similar movies. The CBF engine will find similar movies which are similar either based on *genres*, *tags*, *titles*, or *movie's released year*. Subject to the inserted movie's title, similar movies were recommended.

TABLE V. CBF ENGINE'S OUTPUT

Movie's Title	Movie Recommendation
Toy Story	Toy Story 2
	Toy Story 3
	Toy, The
	Fun
	Toy Soldiers
	Nelly & Monsieur Arnaud
	In Search of the Castaways
	NeverEnding Story, The
	We're Back! A Dinosaur's Story
	Wild, The

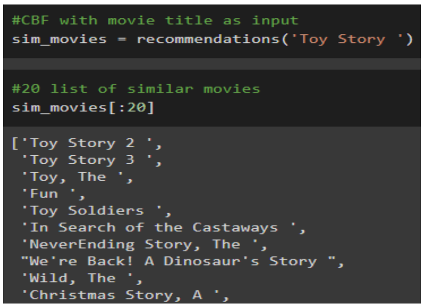

Fig. 3. CBF engine's output

C. Hybrid Collaborative Filtering (HCF) Engine

Table VI shows the outputs using the hybrid recommendation that combines the strength of CF and CBF engines. The input is needed to produce recommendations are *user id* and their favourite *movie' title*.

TABLE VI. HCF ENGINE'S OUTPUT

User	Current Favourite Movie	Movie Recommendation
User50	Cars	Cars 2
		Riding in Cars with Boys
		Monster House
		Curious George

		The Fox and the Hound 2
		Golmaal ', 'Aquamarine
		Ultimate Avengers
		Lifted
		Ice Age 2: The Meltdown
		Children of Men

```
print(hybrid('Toy Story ',5))

['Toy Story 2 ', 'Toy Story 3 ', 'Toy, The ', 'Fun ', 'Toy Soldiers ', 'In Search of the Castaways ', 'NeverEnding Story, The ', 'Wild, The ',

print(hybrid('Train to Busan ',1))

['The D Train ', 'Train of Life (Train de vie) ', "Man on the Train (Homme du train, L') ", 'How to Train Your Dragon 2 ', 'Runaway Train ', '
```

Fig. 4. HCF engine's output

D. *Performance Evaluation*

To evaluate the trained model, testing and evaluation had been implemented. The performance of the trained model was measured by calculating RMSE, *precision*, *recall*, and *F1 score*. To measure the performance of the model using RMSE, the smaller the RMSE value, the more accurate the model. Also, the higher *precision*, *recall*, and *F1 score*, the better the performance [23].

TABLE VII. RESULT OF PERFORMANCE EVALUATION

Performance Evaluation	Value
Root Mean Square Error (RMSE)	0.13
Precision	0.87
Recall	0.99
F1 score	0.93

The standard deviation of unexplained variance can be calculated using RMSE, and it has the advantage of being in the same units as the response variable. Low RMSE values indicate a better match [21]. If the major objective of the model is to predict, RMSE is a good indicator of how accurately the model predicts the response. It is the most important criterion for fit. Therefore, the trained model gained a low RMSE (0.13) value indicating the better model predict response. *Precision* is a measure of result relevancy in information retrieval, while *recall* is a measure of how many truly relevant results are returned [8]. High scores for both indicate that the classifier can produce accurate results (high precision) as well as most of all positive results (high recall). The weighted average of *precision* and *recall* is *F1 Score* [22]. As a result, this score will consider both *false positives* (FP), and *false negatives* (FN) values. With the high *precision* (0.87) together with high *recall* (0.99) and *F1 Score* (0.93), it has shown that this model has better performance in recommending the movie suggestion.

IV. CONCLUSION

This study demonstrated the hybrid development of a movie recommendation system employing collaborative and content-based filtering by applying Matrix Factorisation, TF-IDF, and cosine similarity. This hybrid movie recommendation system assists potential audiences in minimizing their time and avoiding confusion in selecting their favourite movie. The model would also suggest the appropriate movies that suit their interest. Also, based on the available evidence, it has been determined that this model performs better when recommending films. For future work, one may adopt Federated Learning (FL) in the recommendation system. It can provide more opportunities to train the model on edge devices while maintaining the user's privacy because the data never leaves the user's premises.

ACKNOWLEDGEMENTS

The author would like to thank the College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Malaysia, for the support throughout this research.

REFERENCES

[1] V. Advani, "Excerpts From a Masterclass on Movie Recommendation System." 2020, [Online]. Available: https://www.mygreatlearning.com/blog/masterclass-on-movie-recommendation-system/.

[2] K. Goltsman, "Under the Hood of Netflix Recommender," no. 9624670, pp. 4–7, 2017.

[3] Breo Box, "10 benefits of watching movies." 2020, [Online]. Available: https://www.breobox.com/blogs/news/10-benefits-of-watching-movies.

[4] N. Raval and V. Khedkar, "A review paper on collaborative filtering based moive recommedation system," *Int. J. Sci. Technol. Res.*, vol. 8, no. 12, pp. 2507–2512, 2019.

[5] Parrot Analystics, "Investigating the reasons why Americans stop watching TV shows." 2018, [Online]. Available: https://www.parrotanalytics.com/insights/investigating-the-reasons-why-americans-stop-watching-tv-shows/.

[6] J. K. Kim, I. Y. Choi, and Q. Li, "Customer satisfaction of recommender system: Examining accuracy and diversity in several types of recommendation approaches," *Sustain.*, vol. 13, no. 11, 2021, doi: 10.3390/su13116165.

[7] K. Luk, "Introduction to TWO approaches of Content-based Recommendation System." 2019, [Online]. Available: https://towardsdatascience.com/introduction-to-two-approaches-of-content-based-recommendation-system-fc797460c18c.

[8] G. Miryala and R. Gomes, "Available Online at www.jgrcs.info

Comparative Analysis Of Movie Recommendation System Using," vol. 8, no. 10, pp. 10–14, 2017.

[9] B. Rocca, "Introduction to recommender systems." 2019, [Online]. Available: https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada.

[10] T. S. K. K.B Vamsi Anupriya Koneru, D.Siddhabhi, "Movie Recommendation System Using Machine Learning Algorithms," *TEST Eng. Manag.*, vol. 83, no. 1, pp. 2414–2420, 2020, doi: 10.22105/riej.2020.226178.1128.

[11] Bindhu, "Content-Based Recommendation System." 2019, [Online]. Available: https://medium.com/@bindhubalu/content-based-recommender-system-4db1b3de03e7.

[12] B. Stecanella, "What is TF-IDF." 2019, [Online]. Available: https://monkeylearn.com/blog/what-is-tf-idf/.

[13] K. T. Ucar, "How to Calculate TF-IDF (Term Frequency–Inverse Document Frequency) in Python." 2018, [Online]. Available: https://iyzico.engineering/how-to-calculate-tf-idf-term-frequency-inverse-document-frequency-from-the-beatles-biography-in-c4c3cd968296.

[14] S. Prabhakaran, "Cosine Similarity – Understanding the math and how it works (with python codes)." 2018, [Online]. Available: https://www.machinelearningplus.com/nlp/cosine-similarity/#2whatiscosinesimilarityandwhyisitadvantageous.

[15] E. Zeytinci, "How to Build a Content-Based Movie Recommender System." 2019, [Online]. Available: https://towardsdatascience.com/how-to-build-a-content-based-movie-recommender-system-92352f5db7c6.

[16] R. Khanna, "MachineX: Cosine Similarity for Item-Based Collaborative Filtering." 2019, [Online]. Available: https://dzone.com/articles/machinex-cosine-similarity-for-item-based-collabor.

[17] D. Chen, "Recommender System — Matrix Factorization." 2020, [Online]. Available: https://towardsdatascience.com/recommendation-system-matrix-factorization-d61978660b4b.

[18] P. Pantola, "Recommendation Using Matrix Factorization." p. 4, 2018, [Online]. Available: https://medium.com/@paritosh_30025/recommendation-using-matrix-factorization-5223a8ee1f4.

[19] D. Bokde, S. Girase, and D. Mukhopadhyay, "Matrix Factorization model in Collaborative Filtering algorithms: A survey," *Procedia Comput. Sci.*, vol. 49, no. 1, pp. 136–146, 2015, doi: 10.1016/j.procs.2015.04.237.

[20] A. Lineberry and C. Longo, "Creating a Hybrid content-collaborative Movie Recommender using deep learning." 2018, [Online]. Available: https://towardsdatascience.com/creating-a-hybrid-content-collaborative-movie-recommender-using-deep-learning-cc8b431618af.

[21] J. Moody, "What does RMSE really mean?" 2019, [Online]. Available: https://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e.

[22] W. Koehrsen, "Beyond Accuracy: Precision and Recall." 2018, [Online]. Available: https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c.

[23] A. Chugh, "MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better?" 2020, [Online]. Available: https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e.

[24] D. R. Brendan McMahan, "Federated Learning: Collaborative Machine Learning without Centralized Training Data." 2017, [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.

[25] ODSC, "What is Federated Learning?" 2020, [Online]. Available: https://medium.com/@ODSC/what-is-federated-learning-99c7fc9bc4f5.

[26] P. N. S. Norizad, Y. Mahmud, N. M. Noh., & S. Abdul-Rahman, (2022, September). Acne Treatment Recommender Fuzzy Knowledge-Based System with Image Processing. In *2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS)* (pp. 89-94). IEEE.