Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM)
Vol 7, No.1, March 2024, pp. 157 – 166
p-ISSN: 2614-3372 | e-ISSN: 2614-6150

Web-Based Movie Recommendation System Using Content-Based Filtering and Cosine Similarity

[1]Zico Fachreza Meillano, [2] Donni Richasdy, [3]Hasmawati
[1,2,3]Faculty of Informatics, Telkom University Bandung
Email: [1]zicofachreza@students.telkomuniversity.ac.id, [2]donnir@telkomuniversity.ac.id, [3]hasmawati@telkomuniversity.ac.id

Article Info

Article history:
Received Nov 15th, 2023
Revised Feb 29th, 2024
Accepted Mar 12th, 2024

Keyword:
Content-Based Filtering
Cosine Similarity
Count Vectorization
Movie
Recommendation System

ABSTRACT

Movie are one of the most popular entertainment media among people and are often chosen as activities during weekend holidays. As time goes by, world cinema continues to develop with various interesting and entertaining genres, stories and visuals. Because film is one of the entertainment media that can relieve stress from work assignments or lectures and now film production is also growing so that more and more films are being produced until finally people are confused about choosing the film they will watch. To resolve the obstacles faced, movie information is needed that can help people find movies that suit user preferences, so users need a system that can recommend movies. In this research, the author used the content-based filtering method to find movie recommendations. The substance utilized is the movie genre. The Check Vectorization calculation is utilized to discover the term/word weight values in each record and after that these values are utilized as factors within the Cosine closeness to discover similitudes between archives. As a result of this last project the system can generate a kind of recommendations for the 10 most similar movies. The test results from this final project are that the system is running well and is reliable with an alpha test result of 100%, and a reliability test result of 0.7.

Corresponding Author:
Zico Fachreza Meillano,
Faculty of Informatics, Telkom University,
Telekomunikasi Street 1, Sukapura, Kec. Dayeuhkolot, Kabupaten Bandung, Jawa Barat, Indonesia, 40257.
Email: zicofachreza@students.telkomuniversity.ac.id

DOI: http://dx.doi.org/10.24014/ijaidm.v7i1.27390

1. INTRODUCTION

A movie is a motion picture may be an arrangement of pictures that, when shown on a screen, make the dream of moving pictures beneath the impact of the phi impact. This optical dream powers the watcher to see persistent development between diverse objects in sequence. The motion picture generation handle could be a combination of craftsmanship and industry. A motion picture can be made by shooting genuine scenes, drawings or "little" models with a motion picture camera; utilizing conventional activity methods, CGI and computer activity or a combination of a few existing strategies and other visual impacts, making a movie requires a very large budget [1]. As of December 2018, 5,980,614 movie titles had been released [2]. Because until now there are too many movies in various genres so people are confused about watching the movies they want. So, with the presence of a movie recommendation system, it is hoped that it can help users determine movies that are suitable and similar to the movies they like. A recommendation system is a method that can provide information or recommendations that suit the user's tastes based on information received from the user. In creating a recommendation system, two recommendation methods can be used, namely collaborative filtering and content-based filtering [3]. There is previous research that is similar but has different methods, namely movie recommendations using collaborative filtering. This method takes input from the user, cross-checks his/her past history/behavior, and recommends a list of similar movies [4].

The method used in this thesis is content-based filtering. This method provides recommendations to users by determining films based on the user's preferences. The content that will be used in this final assignment is the film genre. The algorithm used to calculate the similarity of each film is Cosine similarity. The document used is a film genre document. To get recommendation results, several stages must be carried out. The initial stage is the process of cleaning film genre documents using a preprocessing process. Several stages are passed in the preprocessing process to get cleaner data, namely cleaning data and merging data. After this process, weighting must be carried out using count vectorization and then the similarity distance is calculated using Cosine similarity. The similarity will be seen from each film genre, so this similarity value will be used to find recommendations so that the framework can create yield within the shape of film suggestions to the client.

Based on the background above, the problem can be formulated as follows. How the Content-based filtering method can recommend films to users based on film genre lyrics and what performance results from applying the Content-based filtering method in providing film recommendations for users. Based on the problem formulation above, there are several goals to be achieved, namely applying the content-based filtering method to provide film recommendations for users and to determine the performance resulting from the content-based filtering method in providing film recommendations using the cosine similarity algorithm.

2. RESEARCH METHOD

The system that will be created in this final project is a system that can provide movie recommendations by applying one of the recommendation system methods. The method used is content-based filtering with the cosine similarity algorithm.

The data used is a film dataset taken from several sites on the internet. This film dataset will go through a preprocessing stage, then the data that has been preprocessed will undergo count vectorization and finally the similarity value of the data is calculated using the cosine similarity algorithm.

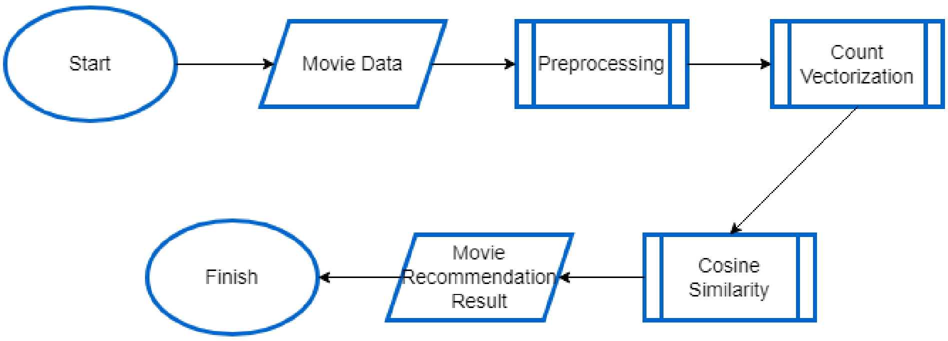

Figure 1. System Design Flowchart

Starting from collecting data on the director's name, movie title, actor's name, and movie genre. The dataset was taken via the website. The movie dataset will be used for preprocessing to obtain data that is ready to be used. Next, the data will go through the count vectorization stage, then the final stage is to find the similarity value using the cosine similarity algorithm.

2.1. Content-Based Filtering

The content-based filtering method is based on information about the content of a particular object [5]. For example, if a user likes A and B is similar to A, the user is likely to like B [6]. The Content-based filtering method creates a user profile based on the attributes that make up an object [7].

The system carries out its evaluation based on an analysis of the similarity of the user's image to the component vectors that make up the object [7]. If the item will be liked by the user then the system will recommend the item to the user. In a film recommendation system with a content-based filtering method, it calculates similarities between films, recommends films that are similar to favorite films, and deletes films with skipped films [8]. The Content-Based Filtering method has a process algorithm which will be explained below.

1. An item will be grouped based on the vector of its constituent components
2. Users will rate and determine whether they like the item or not
3. The system will form a user profile based on the component vector values of an item. Creating user profiles can use the Count Vectorization algorithm. TF is the number of terms in a document.

The system carries out its evaluation based on an analysis of the similarity of the user's image to the component vectors that make up the object [9]. If the item will be liked by the user then the system will recommend the item to the user. In a film recommendation system with a content-based filtering method, it calculates similarities between films, recommends films that are similar to favorite films, and deletes films that are skipped [10].

2.2. Preprocessing

Preprocessing is the first process carried out to apply the verbal weighting method [11]. This process aims to process data into data that is ready to be used by subsequent processes. What is done in the preprocessing process is data cleaning and data merging. In the image below, a flowchart of the preprocessing process will be shown. The preprocessing stages used in this final project is attached in Figure 2.

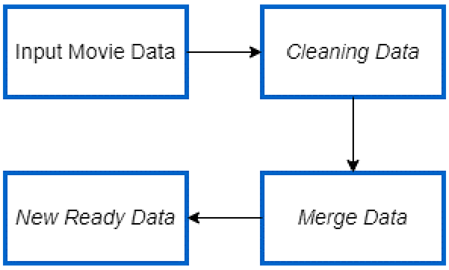

Figure 2. Preprocessing Flowchart

Figure 2 illustrates the flow of the preprocessing process starting from raw film data and then going through the data cleaning and data merging stages. The following is a more detailed explanation of the stages in preprocessing.

2.2.1. Cleaning Data

Cleaning Data is cleaning problematic data or various errors and violations [12]. Cleaning data also the stage of deleting unnecessary data. Deletion in the data cleaning process is a component of film data that has been obtained previously.

Table 1. Cleaning Data Process

Data	Wipe
IMDB 5000 Movie Dataset	Color, num_critic_for_reviews, duration, director_facebook_likes, actor_3_facebook_likes, actor_1_facebook_likes, gross, num_voted_users, cast_total_facebook_likes, facenumber_in_poster, plot_keywords, movie_imdb_link, num_user_for_reviews, language, country, content_rating, budget, title_year, actor_2_facebook_likes, imdb_score, aspect_ratio, movie_facebook_likes.
The Movies Dataset	Adult, belongs_to_collection, budget, homepage, imdb_id.
List of Movies in 2018	Opening, Opening.1, Production company, Ref.
List of Movies in 2019	Opening, Opening.1, Production company, Ref.

2.2.2. Merge Data

Merge Data is a process of combining components and data content from each dataset used. The first step in this merge data method is to determine the time-dependent first-order and second-order statistics of the secondary datasets [13]. The results of combining the data components consist of director_name, actor_1_name, actor_2_name, actor_3_name, genres, movie_title, comb.

2.3. Count Vectorization

Count Vectorization is a method that simply counts the number of words that appear in a document [14]. The Term Frequency (TF) value is obtained from the number of each term in each existing document. The TF value is sought to determine the number of existences of a term in a document. The following is the formula that will be used to calculate the Count Vectorization value using the formula:

$$||\mathcal{X}||_1 = \sum_i |xi|^1 \tag{1}$$

The following is an illustration of what happens in the Count Vectorization process.

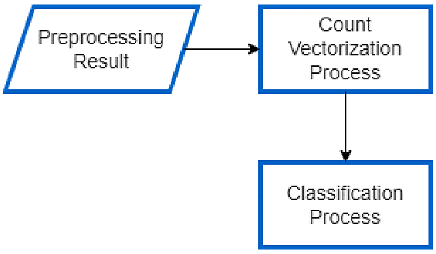

Figure 3. Count Vectorization Flowchart

Figure 3 is a flowchart of Count Vectorization. The following is a simple example of Count Vectorization calculations.

Table 2. Example Sentences from Preprocessing Results

Document	Film Genre
D1	Action Adventure Fantasy
D2	Action Adventure Thriller
D3	Animation Drama Family Fantasy
D4	Adventure Family Fantasy Mystery
D5	Drama Romance

In the table is data that has gone through a preprocessing process. The data taken as a calculation sample is only film genres. Five samples of film genres were taken, named D1 to D5. Next, this data will be used for the calculation stage of word occurrences. The first stage in the Count Vectorization method is to see how many times a word appears (term frequency) of a word. For example, the word "Action" appears once in D1, then the value of tf = 1. The following is an example of Count Vectorization calculation for the word "Action".

Table 3. Calculation of Term Frequency from Count Vectorization (N=5)

Term	Tf				
	D1	D2	D3	D4	D5
Action	1	1	0	0	0
Adventure	1	1	0	1	0
Fantasy	1	0	1	1	0
Thriller	0	1	0	0	0
Animation	0	0	1	0	0
Drama	0	0	1	0	1
Family	0	0	1	1	0
Mistery	0	0	0	1	0
Romance	0	0	0	0	1

2.4. Cosine Similarity

The recommendation algorithm used in this final project is cosine similarity. Cosine similarity is one of the most popular methods that is often used in documenting text for information retrieval and grouping purposes [15]. Cosine similarity may be a degree of closeness between two tall dimensional vectors [16]. Cosine similarity calculates vectors that are related to each other [17]. Cosine similarity is also used to calculate the similarity of vectors in documents. The resulting similarity value is between 0 and 1 [18]. A value of 0 indicates the user profile and items are very irrelevant while a value of 1 is the opposite. Next, the document with the closest similarity value will be used as a recommendation.

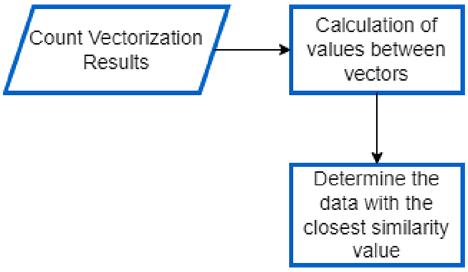

Figure 4. Cosine Similarity Flowchart

Figure 3 is a flowchart of the cosine similarity algorithm. The input for the cosine similarity calculation process is the value obtained from the tf calculation in count vectorization to be used as a vector. The following is a further explanation regarding the cosine similarity calculation process:

1. The weight values for the terms/words in the song genre document in the count vectorization calculation will then be used to calculate the cosine similarity value. The results of the term calculation will be used as a vector variable in the cosine similarity calculation.
2. The values between vectors will be calculated using the cosine similarity formula to get the closest value, these values will then be displayed in table form.

   The following is a simple calculation of cosine similarity of D1 and D2.

   $D1 \times D2$ = (1 × 1) + (1 × 1) + (1 × 0) + (0 × 1) + (0 × 0) + (0 × 0) + (0 x 0) + (0 x 0) + (0 x 0) = 2

   $\sqrt{\mathrm{D1}^2} = \sqrt{1^2+1^2+1^2+0^2+0^2+0^2+0^2+0^2+0^2}$ = 1.73

   $\sqrt{\mathrm{D2}^2} = \sqrt{1^2+1^2+0^2+1^2+0^2+0^2+0^2+0^2+0^2}$ = 1.73

   $Similarity$ (D1, D2) = $\frac{2}{1.73 \times 1.73}$ = 0.67

   The results of similarity calculations between vectors D1 to D5 will be presented in table form as follows.

   Table 4. Similarity Calculation Results

SIM	D1	D2	D3	D4	D5
D1	-	0.67	0.29	0.58	0
D2	0.67	-	0	0.29	0
D3	0.29	0	-	0.5	0.35
D4	0.58	0.29	0.5	-	0
D5	0	0	0.35	0	-

   In the table is the distance value between vectors (D1-D2, and so on). By using the Cosine similarity formula, the similarity value between each document is obtained.

3. Next, sort the data resulting from the similarity value calculation from closest to furthest which has been searched using the cosine similarity formula.

   Table 5. Sequence of Similarity Calculation Results

D1	0.67	0.58	0.29	0
D2	0.67	0.29	0	0
D3	0.5	0.35	0.29	0
D4	0.58	0.5	0.29	0
D5	0.35	0	0	0

4. After sorting based on the closest similarity value to each document, namely D1-D5, the document that has the closest value will be determined based on the value of the recommended document.

   Table 6. Closest Document Sequence

D1	D2	D4	D3	D5
D2	D1	D4	D3	D5
D3	D1	D2	D4	D5
D4	D1	D2	D3	D5
D5	D1	D2	D4	D3

   Based on Table 6 it can be concluded :
   - D1 with a similarity value with itself of 1, then the closest is D2.
   - D2 with a similarity value with itself of 1, then the closest is D1.
   - D3 with a similarity value with itself of 1, then the closest is D1.
   - D4 with a similarity value to itself of 1, then the closest is D1.
   - D5 with a similarity value to itself of 1, then the closest is D1.

Based on the comparison above, it can be concluded that the documents with the closest Cosine similarity values to all documents are D1 and D2.

After getting the Cosine similarity value, the system can provide recommendations to users. For example, from the 5 documents above, namely D1-D5 are film genres. So if the user inputs the film genre D1, then the user presses the enter button, the system will provide film recommendations in the order D2, D4, D3, and D5. The order given by the system is based on the similarity value closest to furthest from D1.

3. RESULTS AND ANALYSIS

3.1. Dataset

The data needed for this final assignment are director_name, actor_1_name, actor_2_name, actor_3_name, genres, movie_title, and comb data. The data is taken via a website that has gone through a preprocessing process so that the final data is as below.

Table 7. Dataset Examples

director_name	actor_1_name	actor_2_name	actor_3_name	genres	movie_title	comb
James Cameron	CCH Pounder	Joel David Moore	Wes Studi	Action Adventure Fantasy Sci-Fi	avatar	CCH Pounder Joel David Moore Wes Studi James Cameron Action Adventure Fantasy Sci-Fi
Gore Verbinski	Johnny Depp	Orlando Bloom	Jack Davenport	Action Adventure Fantasy	pirates of the caribbean: at world's end	Johnny Depp Orlando Bloom Jack Davenport Gore Verbinski Action Adventure Fantasy

Table 7 is an example of a film dataset where there are columns director_name, actor_1_name, actor_2_name, actor_3_name, genres, movie_title, and comb data. The final data that will be used to look for similarities between films so that film recommendations are obtained is comb data, which is a combination of all data.

3.2. Interface Design Implementation

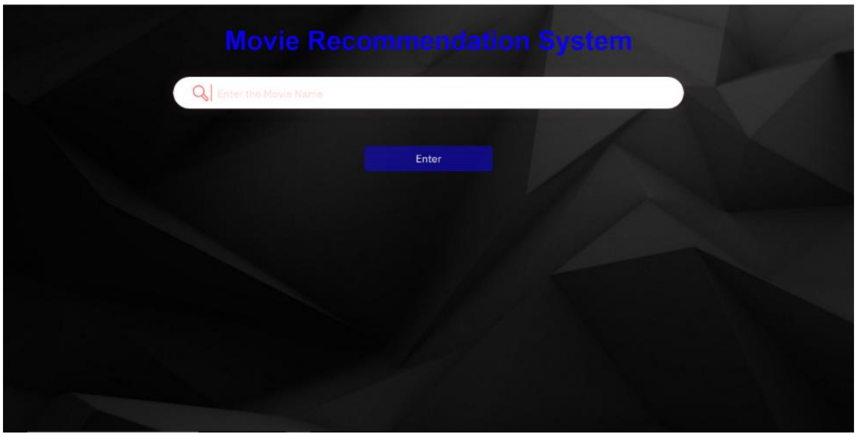

Figure 5. Main Page of The Website

Figure 5 is the main page of the website where there is a search column for movie titles. So users have to input the title of the movie first on the home page of the website. Then after inputting the title of the movie you want to search for, the user presses the enter button so that the title of the movie entered can be processed by the system.

Figure 6 is the movie recommendation list page where the list will display movie titles sorted based on the largest to smallest similarity value with the movie title entered by the user. So in the website will display 10 recommended movie titles that are similar to the movie titles you like.

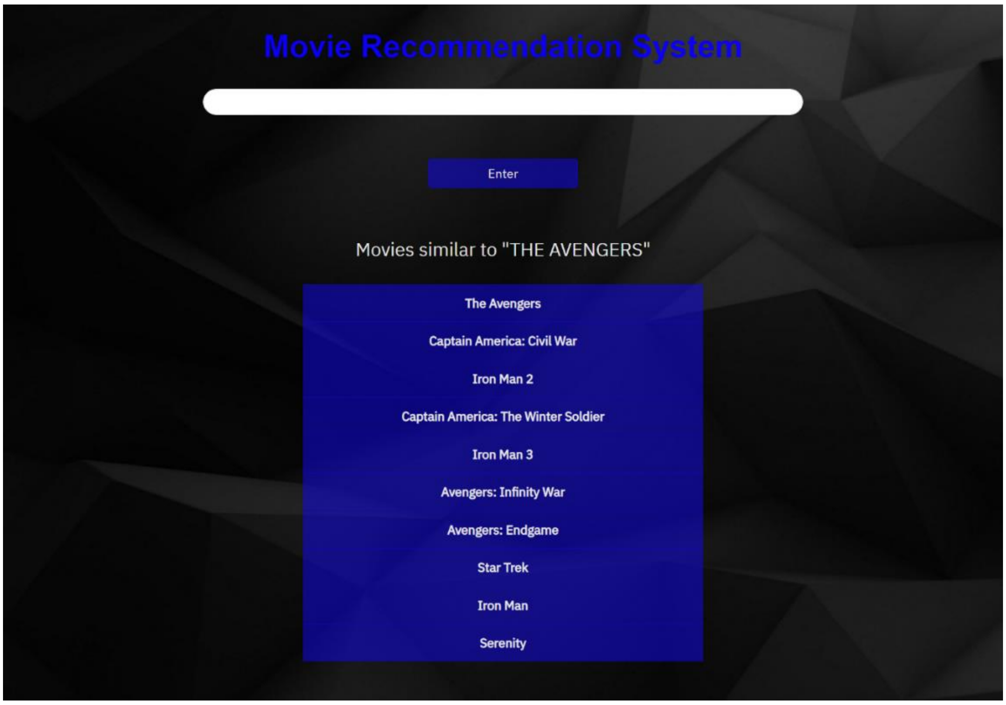

Figure 6. Movie Recommendation List Page

3.3. Alpha Testing

The The purpose of alpha testing is to find out whether the film recommendation system with Content-Based filtering is suitable for use as a website. This testing is carried out by testing all the features of each menu on the website.

The alpha testing scenarios that will be used can be seen in the table below.

Table 8. Alpha Scenario

Tested Features	Testing Details	Test Type
Opening the Website	Displays the main page of the website	Black Box
Enter the movie title in the search column	Displays the title of the film you want to search for	Black Box
Pressing the enter key	Navigate to search results	Black Box
Recommendation Result Display	Displays 10 similar film recommendations	Black Box

In the test scenario which can be seen in table 8, there are 4 features being tested. Alpha testing is carried out in a black box manner, namely by carrying out tests only observing the execution results and checking the functionality of the website being built.

The results of alpha testing on the film recommendation system website can be seen in the table 9.

Table 9. The Result of Alpha Testing

Input Data	Expected Result	Observation Result	Conclusion
Opening the Website	Displays the main page	Can display the main page	Succeed
Enter the movie title in the search column	Displays the title of the film you want to search for	Can display the title of the film that has been input and wants to be searched	Succeed
Pressing the enter key	Navigate to search results	Can direct to search results for film titles	Succeed
Recommendation Result Display	Displays 10 similar film recommendations	Can display 10 similar film recommendations	Succeed

The following is a calculation of the value from the alpha test results.

$$\text{Alpha Test} = \frac{\text{results received total}}{\text{expected total}} \times 100\% \quad (1)$$

$$\text{Alpha Test} = \frac{4}{4} \times 100\% \\ = 100\%$$

From the alpha test results, it can be concluded that the system can run well with alpha test results of 100%.

3.4. Beta Testing

Beta testing aims to try whether a website can run well or not. Beta testing is carried out by conducting user assessments of a system being built. One example is by asking users to fill out a form/questionnaire regarding user satisfaction with the website being built. Questionnaires were distributed using the Random Sample Technique where sample members from the population were randomly selected. One method of conducting beta testing is usability testing. The questionnaire distributed to 40 respondents had 10 questions containing usability aspects. Beta testing is also used to validate the usability and reliability of the software being created.

3.4.1. Validity Test

The essence of all validation techniques is to increase certainty about the questionnaire asked to respondents [19]. The way to determine the results is by comparing the Rcount and Rtable values. If the value Rcount > Rtable then it is declared valid and vice versa. The R table used is the product moment R table with a significance level of 5%. The number of respondents was 40 people. This aims to determine whether or not the questions asked to respondents are valid. If the number of respondents is 40 people, the df (degree of freedom) value is df = 40 – 2, namely df = 38. This df value will then be used to determine the Rtable value. So we get a value of Rtable = 0.312 with df = 38. If the value of Rcount > Rtable then the question is considered valid, and vice versa. The following are the results of calculating the validity of the 10 questions asked to respondents.

Table 10. Questionnaire Validity Test Results

Question To	Rcount	Rtable	Result
1	0,554976	0,312	Valid
2	0,479783	0,312	Valid
3	0,347166	0,312	Valid
4	0,535382	0,312	Valid
5	0,553378	0,312	Valid
6	0,362159	0,312	Valid
7	0,651446	0,312	Valid
8	0,374572	0,312	Valid
9	0,638957	0,312	Valid
10	0,769536	0,312	Valid

Based on the table above, it can be concluded that all questions asked to respondents are considered valid and worth asking respondents because all the Rcount values from the 10 questions are Rcount > Rtable.

3.4.2. Reliability Test

Reliability testing is widely applied by developers to verify the reliability requirements of answers to questionnaires submitted to respondents [20]. Based on the questionnaire, it is considered reliable if the R11 value is greater than 0.6. The author carried out a reliability test on each questionnaire question item because each question item was declared valid. The reliability test results can be obtained as in the table 11.

Table 11. Questionnaire Reliability Test Results

Number of Item Variants	Total Number of Variants	R11	Comparison StandardValue	Result
3,583333	9,771795	0,703665	0,6	Reliable

Based on the table above, it can be concluded that all answers to the questions/questionnaires asked of respondents are consistent or stable so they are considered reliable because the Cronbach Alpha (R11) value is greater than the comparison standard value, namely 0.703665 > 0.6.

4. CONCLUSION

From the tests that have been carried out from collecting movie data, then through stages of preprocessing, count vectorization, and cosine similarity to get movie recommendation results. It can be concluded that the system created has succeeded in answering the objectives of the research, namely from the alpha testing gives results that the system and features on the movie recommendation website can run well and run according to user input, the system designed is successful in providing movie recommendations based on term and similarity values of the movie genre, the system can run well with an alpha test result of 100%. From the beta test, the results of the validity test showed that all questions asked to respondents are considered valid and worthy of being asked to respondents, and the results of the reliability test showed that all questionnaire answers submitted to respondents are consistent or stable so they are considered reliable with a value 0.703665. If anyone wants to continue this research, many features must be added, developed, and expanded. First, you can create a movie recommendation system with more recent year coverage. Second, Add other features and add useful movie information for users.

ACKNOWLEDGEMENTS

The author would like to thank Telkom University Bandung for supporting the implementation of this entire series of research.

REFERENCES

[1] Saveetha. D, G. Maragatham. Movie Rating System based on Blockchain. *IEEE 2021 International Conference on Computer Communication and Informatics (ICCCI)*. 2021: 1–3.
[2] Press Room - IMDb. Accessed: Jan. 24, 2024. [Online]. Available: https://www.imdb.com/pressroom/stats/
[3] F. Ricci, L. Rokach, and B. Shapira. Introduction to Recommender Systems Handbook. *Recommender Systems Handbook.* Boston. MA: Springer US. 2011: 1–35.
[4] Institute of Electrical and Electronics Engineers and Hindusthan Institute of Technology. *Proceedings of the International Conference on Electronics and Sustainable Communication Systems*. ICESC. 2020 : 02-04.
[5] T. Rutkowski, J. Romanowski, P. Woldan, P. Staszewski, R. Nielek, and L. Rutkowski. A Content-Based Recommendation System Using Neuro-Fuzzy Approach. *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. 2018: pp. 1–8.
[6] F. Fessahaye *et al.*. T-RECSYS: A Novel Music Recommendation System Using Deep Learning. *2019 IEEE International Conference on Consumer Electronics (ICCE)*. 2019: 1–6.
[7] T. Badriyah *et al.*. Konferensi Nasional Sistem Informasi 2018 STMIK Atma Luhur Pangkalpinang. 2018.
[8] Y. Hu and M. Ogihara. NEXTONE PLAYER: A MUSIC RECOMMENDATION SYSTEM BASED ON USER BEHAVIOR. 2011.
[9] T. Badriyah *et al.*. Konferensi Nasional Sistem Informasi 2018 STMIK Atma Luhur Pangkalpinang. 2018.
[10] Y. Hu and M. Ogihara. NEXTONE PLAYER: A MUSIC RECOMMENDATION SYSTEM BASED ON USER BEHAVIOR. 2011.
[11] R. R. M. Putri, R. Y. Herlambang, and R. C. Wihandika. Implementasi Metode K-Nearest Neighbour Dengan Pembobotan TF.IDF.ICF Untuk Kategorisasi Ide Kreatif Pada Perusahaan. *Jurnal Teknologi Informasi dan Ilmu Komputer*. 2017; 4(2): 97.
[12] L. Ma, Q. Pei, L. Zhou, H. Zhu, L. Wang, and Y. Ji. Federated Data Cleaning: Collaborative and Privacy-Preserving Data Cleaning for Edge Intelligence. *IEEE Internet Things J*. 2021; 8(8): 6757-6770.
[13] Y.-T. Lin, A. E. Newhall, T. F. Duda, P. F. J. Lermusiaux, and P. J. Haley. Merging Multiple-Partial-Depth Data Time Series Using Objective Empirical Orthogonal Function Fitting. *IEEE Journal of Oceanic Engineering*. 2010; 35(4): 710-721.
[14] M. Zaynurroyhan, A. Pambudi, and S. Artikel. Jurnal Explore IT|8 Perbandingan TF-IDF dengan Count Vectorization Dalam Content-Based Filtering Rekomendasi Mobil Listrik INFO ARTIKEL ABSTRAK Katakunci: Content-Based Filtering K-Nearest Neighbor Mobil Listrik.
[15] C. S. D. Prasetya. Sistem Rekomendasi Pada E-Commerce Menggunakan K-Nearest Neighbor. *Jurnal Teknologi Informasi dan Ilmu Komputer*. 2017; 4(3): 194.
[16] S. Zhu, J. Wu, H. Xiong, and G. Xia. Scaling up top-K cosine similarity search. *Data Knowl Eng*. 2011; 70(1): 60-83.
[17] Ş. ÖZTÜRK BİRİM and A. TÜMTÜRK. A Novel Algorithmic Similarity Measure for Collaborative Filtering: A Recommendation System Based on Rating Distances. *Academic Platform Journal of Engineering and Smart Systems*. 2022; 10(1): 57-69.
[18] ST. , MT. G. R. E. R. S. M. Ade Romadhony. Analisis dan Implementasi House Recommendation Menggunakan Content-Based Recommender System.
[19] J. W. Duran and J. J. Wiorkowski. Estimating the correctness of computer programs. *Inf Sci (N Y)*. 1982; 26(2): 141-153.
[20] J. Zhang, X. Ma, and Y. Zhao. Reliability Demonstration for Long-Life Products Based on Hardened Testing Method and Gamma Process. *IEEE Access*. 2017; 5: 19322-19332.

BIBLIOGRAPHY OF AUTHORS

Zico Fachreza Meillano, is a student at Telkom University Bandung, Informatics study program. Has an interest in the field of AI, especially Recommendation Systems. During college, I interned at Automate All as a Telegram chatbot researcher and at PT. Pertamina Patra Niaga as data processor.

Donni Richasdy, S.T., M.T., is an alumnus of the Telkom Institute of Technology (S1) and the Bandung Institute of Technology (S2). He works as a permanent lecturer at Telkom University majoring in informatics with an interest in Data Engineering and Natural Language Processing. He also has a position as an expert assistant in the field of Data Science.

Hasmawati, S.Kom., M.Kom., is an alumnus of Binus University (S1) and Telkom University (S2). He works as a permanent lecturer at Telkom University majoring in Informatics with an interest in the fields of Text mining, Natural language processing, and Machine learning. He also holds a position as an expert assistant in the field of Data Science.