JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 3, September 2023, Pp. 729-735

MOVIE RECOMMENDER SYSTEM USING DECISION TREE METHOD

Muhammad Bilal Rafif Azaki[1], Z. K. A. Baizal*[2]

1. Informatics, School Of Computing, Telkom University, Bandung, Indonesia
2. Informatics, School Of Computing, Telkom University, Bandung, Indonesia

Article Info

Keywords: Collaborative Filtering; Recommender System; Movie Recommender System; Decision Tree

Article history:
Received 2 March 2023
Revised 16 March 2023
Accepted 30 March 2023
Available online 1 September 2023

DOI :
https://doi.org/10.29100/jipi.v8i3.3867

* Corresponding author.
Z. K. A. Baizal
E-mail address:
baizal@telkomuniversity.ac.id

ABSTRACT

In this modern era, many things that can be done online, one of which is watching movies. When the number of movies increases, people often find it difficult to decide which movie to watch next. To solve this problem, a useful recommendation system was developed to find movies that one might like based on movies that have been watched before. This research develops a movie recommendation system using Collaborative Filtering (CF) with the Decision Tree algorithm. In this study, the data used were movie data and ratings obtained from the grouplens.org website. Then the movielens dataset is filtered and only saves movies with a rating of more than 50 that are used in the recommendation system. In this study, Mean Absolute Error (MAE) is used as a method to assess the accuracy of the movie recommendation system. Based on the research that has been done, Decision Tree gets better results with an MAE value of 0,942 compared to Collaborative Filtering with an MAE value of 1,242.

I. INTRODUCTION

In the current digital age, the industry is growing rapidly, such as in the movie and technology industry [1]. According to IMDb, the average number of movies produced reached 4584 in 2005, rising to 9387 in 2015 [1]. Due to the increasingly advanced development of movies, finding movies that match the desired interests will be very difficult and can take a lot of time [2]. Therefore, a recommendation system is needed to select movies that are of interest and according to users [3].

A recommender system is a method of helping users find items that suit their needs [4]. In general, recommendation systems are grouped into Collaborative Filtering, Hybrid-Based, Content-Based, and Knowledge-Based [5]. One of the most widely used recommendation systems is Collaborative Filtering (CF). Most Collaborative Filtering takes advantage of similarities between users. Similar users are found by calculating the similarity of ratings from users [6]. Decision Tree is one of the Collaborative Filtering algorithms developed by Ross Quinlan. Decision trees have advantages such as high levels of accuracy, high speed in the classification process, and the ability to learn well in simple development [7].

Jadhav, S. D et. al. [7] used the collaborative filtering method in the book recommendation system. In general, the recommender system uses the KNN algorithm to perform classification but takes a long time to process large datasets. Therefore, this study carried out a combination of Collaborative Filtering and Decision Tree. Due to the fast classification of decision trees, this study successfully recommended a book with more accurate results and less time. Meanwhile, Zhang, J et. al. [8] use the Collaborative Filtering method to implement a recommendation system but has problems with time complexity, so this study uses the Weighted Slope One method to design and implement virtual matrix to obtain recommendation results with reduced time complexity. This research succeeded in building a web-based system that can evaluate the feasibility and accuracy of the system based on the data obtained.

Singh, R. H. et. al. [9] used content-based filtering to illustrate movie recommendation modeling and recommend movies to users. There is also machine learning to handle big data and automate the creation of analytical models. This research successfully implements the KNN Algorithm to provide more accuracy when recommending movies. Meanwhile, Bhalse, N et. al. [10]used the collaborative filtering method to predict the list of top n movie recommendations to active users by using singular value decomposition and cosine similarity to calculate similarity values. This study managed to overcome the large dataset and sparse scoring matrix.

Based on previous research, a movie recommendation system based on Collaborative Filtering has been studied, but the results obtained are not optimal so that further development is needed. In this study, we took a dataset from the grouplens.org website and filtered the data by removing movies with a total rating of less than 50. The purpose of this research is to implement a recommender system using collaborative filtering and Decision Tree algorithms. The testing procedure is performed by testing the movielens dataset, and to evaluate the accuracy of the recommender system, we use the Mean Absolute Error (MAE).

II. RESEARCH METHOD

A. System Design

Fig. 1. Shows the stages of the system, starting with the movielens dataset, followed by the preprocessing of the data. After that, the collaborative filtering process is carried out, and finally the decision tree algorithm is executed.

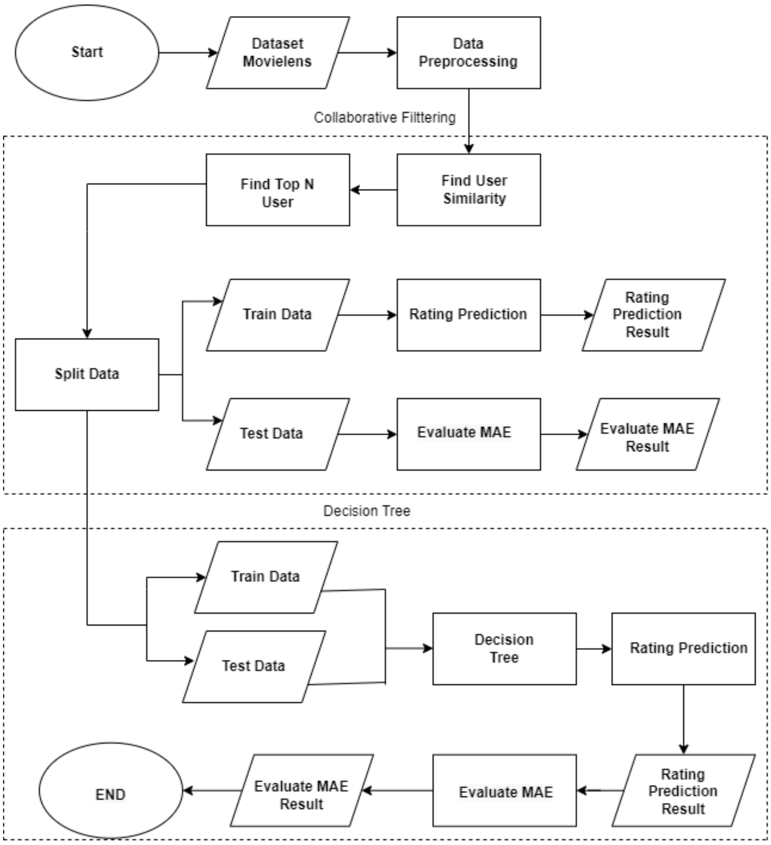


Fig. 1. System Recommendation

B. Dataset

In the research, two datasets were utilized from grouplens.org. The first dataset contained information on 9742 movies, including attributes such as id, title, and genre. The second dataset contained 100836 ratings data, including attributes such as *userId*, *movieId*, rating, and timestamp. Tables 1 and 2 show an examples of movie dataset and rating dataset.

TABLE I
MOVIE DATASET

movieId	Title	Genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance

TABLE II
RATING DATASET

userId	movieId	rating
1	70	3.0
1	101	5.0
1	110	4.0

C. *Data Preprocessing*

At this stage, we combine rating and movie data into a new dataframe and then delete the genres and timestamp columns. After that, we calculated the average from the rating column and filtered out movies that have a total rating of more than 50. The combined dataset has 606 users, 438 movies, 10 unique ratings, and a rating value of 0,5 – 5,0. Table 3 shows an example of a combined dataset of movies and ratings.

TABLE III
MERGED DATASET OF MOVIE AND RATING

movieId	Title	userId	rating
1	Toy Story (1995)	1	4,0
1	Toy Story (1995)	15	2,5
1	Toy Story (1995)	17	4,5

The dataset was then transformed into a matrix with dimensions of 606 x 438. Table 4 shows an example of a rating matrix.

TABLE IV
RATING MATRIX

movieId userId	1	2	3	…	122904
1	4,0	0,0	4,0	…	0,0
2	0,0	0,0	0,0	…	0,0
3	0,0	0,0	0,0	…	0,0
…	…	…	…	…	…
606	2,5	0,0	0,0	0,0	0,0

We normalize the matrix by subtracting the average of each row from the matrix. Movies with a rating less than the average user rating get a negative value, and movies with a rating more than the average user rating get a positive value. Table 5 shows an example of normalization matrix.

TABLE V
NORMALIZATION MATRIX

movieId userId	1	2	3	…	122904
1	2,849	-1,150	2,849	…	-1,150
2	-0,124	-0,124	-0,124	…	-0,124
3	-0,004	-0,004	-0,004	…	-0,004
…	…	…	…	…	…
606	0,512	-1,987	-1,987	…	-1,987

D. *Collaborative Filtering*

Collaborative Filtering is a method used in recommendation systems that utilizes similarities between items or users to generate recommendations. This is done by analyzing the attributes of items or users and looking for similarities with other items or users to provide appropriate recommendations [11]. Collaborative filtering in the recommendation system has several advantages, including having high speed in providing recommendations, being efficient, producing high quality recommendations, and being able to provide accurate recommendations to users [12][13]Collaborative Filtering identifies relationships by comparing the interactions of users or items. When it comes to users, the similarity is known as User-Based Similarity. And when it comes to items, it is referred to as Item-Based Similarity [14][15]. In the research, a Collaborative Filtering experiment was conducted using User-Based Similarity. One way to calculate similarity is by using Pearson Correlation. The following is the formula for finding similarity using Pearson Correlation [16][17]:

$$Sim(i,j) = \frac{\sum_{m\in I_{i,j}}(R_{i,m} - \bar{R}_i)(R_{j,m} - \bar{R}_j)}{\sqrt{\sum_{m\in I_{i,j}}(R_{i,m} - \bar{R}_i)^2}\sqrt{\sum_{m\in I_{i,j}}(R_{j,m} - \bar{R}_j)^2}} \tag{1}$$

Where in equation (1), i and j are the users whose similarity you want to find. $I_{i,j}$ is a set of items rated by users i and j. $\bar{R}_i$ is the average rating of user i on $I_{i,j}$. $\bar{R}_j$ is the average rating of user j in $I_{i,j}$. $R_{i,m}$ is user rating i for item m. $R_{j,m}$ is user rating j for item m [16].

We search for users who are similar to user with userId 1. The similarity between users is determined using a pre-set threshold. Users whose similarity value is greater than the threshold are considered similar. These similar users are then sorted in descending order, from the most similar to the least similar. Table 6 shows the users who have a similarity with userId 1.

TABLE VI
TOP *N* USER

userId	Similarity Value
266	0,403
469	0,398
313	0,378
57	0,371
577	0,359
555	0,355
135	0,346
368	0,343
39	0,341
493	0,332

After that, rating predictions will be made using User-Based Collaborative Filtering and then the prediction results will be evaluated using Mean Absolute Error (MAE).

E. *Decision Tree*

Decision tree is a technique used in Collaborative Filtering, which was developed by Ross Quinlan. Decision trees have the advantages of high accuracy, high classification speed, and strong learning ability for simple constructions [7]. We use the Decision Tree in the Collaborative Filtering method where the target items that have contextual feature values are treated as the test sample to be classified and the other items, which were previously evaluated by active users. Then we make the *movieId* and *userId* columns as feature values and the rating column as label. The following is the formula for calculating predictions with a Decision Tree [18]:

$$Info_{gain}(D) = -\sum_{j=1}^{k} P_j log2(P_j) \tag{2}$$

F. *Performance Evaluation*

In the evaluation phase, the performance of the machine learning model will be evaluated using MAE (Mean Absolute Error) which is a metric that can be used to gauge the performance of a machine learning model. This metric calculates the average absolute difference between the value predicted by the model and the actual value. The model is considered to perform better if the MAE is low [19]. The formula for calculating MAE is as follows:

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \tag{3}$$

In this research, the MAE is calculated by taking the average of the absolute difference between the predicted rating p_i and the actual rating q_i for all the items, where N represents the total number of predicted items.

III. RESULT AND DISCUSSION

A. Collaborative Filtering Experiment

We use the Pearson correlation to calculate similarity between users which has a value between -1 to 1, where -1 indicates that the user has different movie preferences while 1 indicates that the user has the same movie preference. Table 7 shows an example of a user similarity matrix.

TABLE VII
USER SIMILARITY MATRIX

userId userId	1	2	3	…	606
1	1,000	-0,071	0,004	…	0,084
2	-0,071	1,000	-0,017	…	-0,053
3	0,004	-0,017	1,000	…	-0,036
…	…	…	…	…	…
606	0,084	-0,053	-0,036	…	1,000

We divide our dataset into two parts training data and testing data. The training data is used to predict which movies a user has not yet rated. In this data, movies that have been rated by the user are marked as 0, while movies that have not been rated by the user are marked as 1. The testing data is used for evaluation, where movies that have been rated by the user are marked as 1, and movies that have not been rated by the user are marked as 0. Examples of training data and testing data are shown in tables 8 and 9.

TABLE VIII
TRAINING DATA

movieId	Title	userId	rating
318	Shawshank Redemption, The (1994)	40	0
4308	Moulin Rouge (2001)	64	0
1221	Godfather: Part II, The (1974)	424	0

TABLE IX
TESTING DATA

movieId	Title	userId	rating
1968	Breakfast Club, The (1985)	599	1
288	Natural Born Killers (1994)	191	1
8961	Incredibles, The (2004)	534	1

We calculate the predicted rating using User-Based Collaborative Filtering. The following is an example of the rating prediction results from User-Based Collaborative Filtering shown in table 10.

TABLE X
RATING PREDICTION USER-BASED COLLABORATIVE FILTERING

movieId userId	1	2	3	…	122904
1	0,000	12,403	0,000	…	-1,007
2	4,093	2,505	-4,056	…	15,870
3	-4,464	-1,659	-1,602	…	-1,202
…	…	…	…	…	…
606	0,000	5,670	1,139	…	3,698

The accuracy of the predicted results was evaluated using the MAE metric by comparing the predicted ratings to the original ratings. The Collaborative Filtering method achieved the best MAE score of 1,242.

B. Decision Tree Experiment Result

We use the training data and testing data that have been obtained. In the preprocessing stage we take the *movieId* and *userId* columns as features and the rating as label in the training data and testing data. We created a Decision Tree model using the python sci-kit-learn tool. Decision Tree model that has been made is converted into a tree shown in fig. 2.

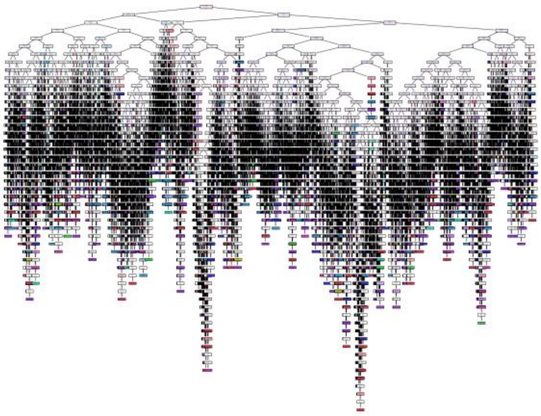
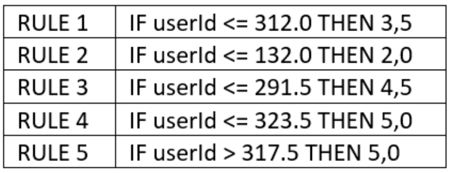


Fig. 2. Decision Tree Model Result

The Decision Tree that has been formed has 5000 rules, if *userId* 1 has the same similarity value as *userId* 599 then *userId* 599 will give the same rating to a certain *movieId* that has been rated by *userId* 1. Fig. 3. Shows an example of 5 rules contained in a Decision Tree.



RULE 1	IF userId <= 312.0 THEN 3,5
RULE 2	IF userId <= 132.0 THEN 2,0
RULE 3	IF userId <= 291.5 THEN 4,5
RULE 4	IF userId <= 323.5 THEN 5,0
RULE 5	IF userId > 317.5 THEN 5,0

Fig. 3. Example Rules of Decision Tree Model

After the decision tree model is created, we trained the model using the training data and we make predictions using testing data. Then we combine the testing data with the predicted results using a Decision Tree model. Table 11 shows an example of rating prediction results using the Decision Tree model.

TABLE XI
RATING PREDICTION DECISION TREE

movieId	userId	rating_prediction
1968	599	3,5
288	191	2,0
8961	534	4,5
5952	416	5,0
356	567	5,0

The accuracy of the predicted results was evaluated using the MAE metric by comparing the predicted ratings to the actual ratings. Previous research resulted in an evaluation score of 1,084 while this study obtained the best MAE score of 0,942.

IV. CONCLUSION

This research implemented a recommendation system using Collaborative Filtering and the Decision Tree algorithm. The dataset was obtained from grouplens.org and contained 9742 movies and 100836 ratings. The results of the tests showed that both methods produced low MAE values, with Collaborative Filtering having a MAE of 1,242 and the Decision Tree having a MAE of 0,942. These results indicate that the system works well with the movielens dataset. It is expected that the findings of this study will aid future researchers in their efforts to improve recommendation systems and benefit the wider community.

REFERENCES

[1] T. Sharma, R. DIchwalkar, S. Milkhe, and K. Gawande, "Movie buzz-movie success prediction system using machine learning model," in *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, Dec. 2020, pp. 111–118. doi: 10.1109/ICISS49785.2020.9316087.

[2] M. Gupta, A. Thakkar, Aashish, V. Gupta, and D. P. S. Rathore, "Movie Recommender System Using Collaborative Filtering," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Jul. 2020, pp. 415–420. doi: 10.1109/ICESC48915.2020.9155879.

[3] S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok, and B. Venkatesh, "Content-based movie recommendation system using genre correlation," in *Smart Innovation, Systems and Technologies*, 2019, vol. 105, pp. 391–397. doi: 10.1007/978-981-13-1927-3_42.

[4] A. Pal, P. Parhi, and M. Aggarwal, "An improved content based collaborative filtering algorithm for movie recommendations," in *2017 Tenth International Conference on Contemporary Computing (IC3)*, Aug. 2017, pp. 1–3. doi: 10.1109/IC3.2017.8284357.

[5] U. Thakker, R. Patel, and M. Shah, "A comprehensive analysis on movie recommendation system employing collaborative filtering," *Multimed Tools Appl*, vol. 80, no. 19, pp. 28647–28672, Aug. 2021, doi: 10.1007/s11042-021-10965-2.

[6] M. Srifi, A. Oussous, A. A. Lahcen, and S. Mouline, "Recommender systems based on collaborative filtering using review texts-A survey," *Information (Switzerland)*, vol. 11, no. 6. MDPI AG, Jun. 01, 2020. doi: 10.3390/INFO11060317.

[7] S. D. Jadhav and H. P. Channe, "Efficient Recommendation System Using Decision Tree Classifier and Collaborative Filtering," *International Research Journal of Engineering and Technology*, 2016, [Online]. Available: www.irjet.net

[8] J. Zhang, Y. Wang, Z. Yuan, and Q. Jin, "Personalized Real-Time Movie Recommendation System: Practical Prototype and Evaluation," 1007. [Online]. Available: http://121.42.174.147:8080/

[9] G. Srivastav, R. H. Singh, S. Maurya, T. Tripathi, and T. Narula, "Movie Recommendation System using Cosine Similarity and KNN," *Article in International Journal of Engineering and Advanced Technology*, no. 9, pp. 2249–8958, 2020, doi: 10.35940/ijeat.E9666.069520.

[10] N. Bhalse and R. Thakur, "Algorithm for movie recommendation system using collaborative filtering," *Mater Today Proc*, Feb. 2021, doi: 10.1016/j.matpr.2021.01.235.

[11] G. Liu and X. Wu, "Using Collaborative Filtering Algorithms Combined with Doc2Vec for Movie Recommendation," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Mar. 2019, pp. 1461–1464. doi: 10.1109/ITNEC.2019.8729076.

[12] R. Ji, Y. Tian, and M. Ma, "Collaborative Filtering Recommendation Algorithm Based on User Characteristics," in *2020 5th International Conference on Control, Robotics and Cybernetics, CRC 2020*, Oct. 2020, pp. 56–60. doi: 10.1109/CRC51253.2020.9253466.

[13] C.-S. M. Wu, D. Garg, and U. Bhandary, "Movie Recommendation System Using Collaborative Filtering," in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, Nov. 2018, pp. 11–15. doi: 10.1109/ICSESS.2018.8663822.

[14] G. Geetha, M. Safa, C. Fancy, and D. Saranya, "A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System," in *Journal of Physics: Conference Series*, Apr. 2018, vol. 1000, no. 1. doi: 10.1088/1742-6596/1000/1/012101.

[15] R. Chen, Q. Hua, Y. S. Chang, B. Wang, L. Zhang, and X. Kong, "A survey of collaborative filtering-based recommender systems: from traditional methods to hybrid methods based on social networks," *IEEE Access*, vol. 6, pp. 64301–64320, 2018, doi: 10.1109/ACCESS.2018.2877208.

[16] Z. Zhao and J. Zhang, "Weighted Slope One Algorithm Optimization Based on User Similarity and Item Similarity," in *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, Jul. 2018, pp. 34–39. doi: 10.1109/FSKD.2018.8686857.

[17] A. Tripathi and A. K. Sharma, "Recommending Restaurants: A Collaborative Filtering Approach," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Jun. 2020, pp. 1165–1169. doi: 10.1109/ICRITO48877.2020.9197946.

[18] S. Linda and K. K. Bharadwaj, "A decision tree based context-aware recommender system," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11278 LNCS, pp. 293–305. doi: 10.1007/978-3-030-04021-5_27.

[19] A. A. Fakhri, Z. K. A. Baizal, and E. B. Setiawan, "Restaurant Recommender System Using User-Based Collaborative Filtering Approach: A Case Study at Bandung Raya Region," in *Journal of Physics: Conference Series*, May 2019, vol. 1192, no. 1. doi: 10.1088/1742-6596/1192/1/012023.