



Available online at :
<http://ejournal.amikompurwokerto.ac.id/index.php/telematika/>

Telematika

Accredited SINTA “2” Kemenristek/BRIN, No. 85/M/KPT/2020



Use of Hybrid Methods in Making E-commerce Product Recommendation Systems to Overcome Cold Start Problems

Budi Santosa¹, Muhamad Azam Fuadi², Mangaras Yanu Florestiyanto³, Vynska Amalia Permadi⁴, Wilis Kaswidjanti⁵

^{1,2,3,4,5} Informatic, Faculty of Industrial Engineering
 UPN Veteran Yogyakarta, Yogyakarta, Indonesia

E-mail: dissan@upnyk.ac.id¹, 123170059@student.upnyk.ac.id^{2*}, mangaras.yanu@upnyk.ac.id³, vynspermadi@upnyk.ac.id⁴, wilisk@upnyk.ac.id⁵

ARTICLE INFO

History of the article:

Received July 20, 2022

Revised March 3, 2023

Accepted August 28, 2023

Keywords:

E-Commerce, Recommendation System, Hybrid, Collaborative Filtering, Cold Start

Correspondece:

E-mail:

123170059@student.upnyk.ac.id

ABSTRACT

The large number of users and the items offered in e-commerce make it difficult for buyers to choose the right items and sellers to offer their items to the right buyers. To overcome this problem, a system that can offer and recommend goods automatically, namely a recommendation system is needed. One of the most popular methods used to create a recommendation system is collaborative filtering, the recommendations are created based on similarities in user behavior. Unfortunately, this method has a weakness, namely cold start, where the recommendations will be inaccurate on data that has a lot of new users and items due to minimal historical data regarding user behavior. This problem will be tried to be solved in this study using a hybrid method, where this method combines more than 1 method to create a list of recommendations so that it will cover the shortcomings of each method. This study uses Amazon's e-commerce product and transaction data. The use of the hybrid method in this study can overcome the cold start problem by using switching and mixed methods, by not using the collaborative filtering model on new user recommendations or users who have little interaction. New users will receive recommendations based on the combination of popularity-based and content-based filtering models. This can be seen from the Mean Absolute Error (MAE) value of the model, where the MAE value for the data with a minimum user has at least 3 times rating is 0.566883, for the minimum 7 times, the MAE value is smaller, 0.487553.

INTRODUCTION

In this modern era, buying and selling transactions can be carried out without direct meetings between sellers and buyers, and this can be done in e-commerce. It is an online trading application for direct and indirect buying and selling transactions between sellers and buyers (Aan & Permana, 2019). The problem that occurs in e-commerce is that buyers will have difficulty in choosing products because the products offered in e-commerce are very many, while sellers have difficulty offering their products to the right buyers because of the large number of users on e-commerce (Prasetya, 2017).

These e-commerce problems can be overcome in several ways, for example by using a decision support system (Noviansyah et al., 2019). However, decision support systems require more specific data to provide recommendations and also do not look at user behavior or historical data. For that we need a system that can recommend the right product to the right buyer as well as from the many diverse products, the system is a recommendation system. The recommendation system in e-commerce can analyze data from a

product and prospective buyers, and then provide a recommendation to the prospective buyer (Knijnenburg et al., 2012). The use of recommendation systems in e-commerce is said to increase product sales and build buyer loyalty (Mobasher, 2007).

Making a recommendation system can use the method of decision support systems in general such as Simple Additive Weighting (SAW) and Weighted Product (WP) (Arvianti et al., 2019). This method is suitable when used on a set of similar items and has the same attributes. However, the most popular method for making a recommendation system is a method that can be used when the choice of items is not only 1 type, like the methods that create recommendations based on the similarity of user behavior and similarities between items, such as collaborative filtering to find similarities between user behavior and content-based filtering. to find similarities between items (Arvianti et al., 2019). In addition to these two methods, there is one more method, namely hybrid which combines several methods at once in making a recommendation system.

Several studies have been conducted to create a recommendation system. One of the studies was conducted using the collaborative filtering method. Collaborative filtering carried out in this study uses Pearson similarity which has been modified by adding weighting to the number of the same items that are rated, the more the number of the same items, the higher the weight. This modified Pearson similarity is then used to find the similarity value in KNN collaborative filtering. The $K=40$ test resulted in a Mean Absolute Error value of 0.821 on the use of cosine similarity, 0.838 on the use of Jaccard similarity, 0.821 on the Pearson similarity, and also 0.695 on the modified Pearson similarity (Zhou et al., 2020). From the test results, it can be seen that the weighting of Pearson similarity can reduce the error value that occurs in the recommendation system model. The collaborative filtering method that relies on interactions between users and items has a weakness, namely cold start, where recommendations will be inaccurate on data that has a lot of new users and items because it doesn't have much interaction (Guo, 2013). Another study was conducted using a hybrid method to build e-commerce recommendation systems. The hybrid method used is built by combining content-based filtering and collaborative filtering, both using the cosine similarity algorithm. In the test, the research resulted in an f-measure value at $k = 10$ with content-based filtering of 0.103, collaborative filtering of 0.036, and hybrid method of 0.093. The greater the value of K , the f-measure in content-based filtering and hybrid will decrease but the decrease in the hybrid method is not too significant, while in collaborative filtering it will increase but the value remains the lowest compared to the other two methods. This is because the collaborative filtering used is not able to handle sparse data (Prasetya, 2017).

Based on previous research, the use of the hybrid method is believed to be able to handle problems such as cold-starts that occur in collaborative filtering (Thorat et al., 2015). So, in this study a hybrid method will be used by combining collaborative filtering using the K-Nearest Neighbor and Pearson Similarity algorithms with content-based filtering using the Tf-Idf and Cosine Similarity algorithms, and also the popularity-based method linearly, namely by combining the results of rating predictions on each method for creating e-commerce product recommendations. The use of a linear hybrid method is expected to overcome the cold-start problem that occurs in collaborative filtering.

RESEARCH METHODS

The research method used in this study is a quantitative method where there are quantitative calculations at several stages such as pre-processing the data and making models. The quantitative method

in this study consists of the stages of problem identification, literature study, data collection, data pre-processing, modeling for the system, implementation and testing of the system, and finally conclusions and suggestions.

1. Data Collecting

Data collection is a process to find information used in this research. The information or data needed for this research is about user data, products, and also activities carried out by users on products in e-commerce. The data is obtained by downloading the data that is already available on Jianmo Ni's website (Ni et al., 2020). The data is secondary because it is obtained from existing sources, sourced from the Amazon website.

The data obtained contains the details of a product along with the assessment given by the user to the product which was taken from 1996 to 2018 as much as 233.1 million data. Due to the large amount of data available when taking all categories, this study only takes data from several categories, including: fashion, all beauty, appliances, luxury beauty, and prime pantry. From several categories taken, data obtained as many as 271844 rows and 16 columns for product detail data. As for the product review data obtained as many as 2819478 rows and 14 columns. The overview of the data that has been obtained can be seen in Figure 1.

meta_data												
asin	brand	category	date	description	feature	fit	main_cat	price	rank	title	imageURL	imageURLHighRes
6546546450	idea village		NaN	["Loud 'N Clear Personal Sound Amplifier allow...		NaN	All Beauty	NaN	2,938,573 in Beauty & Personal Care (Loud 'N Clear™ Personal Sound Amplifier	NaN	NaN
7178680776	NaN		NaN	["No7 Lift & Luminate Triple Action Serum 50ml...		NaN	All Beauty	\$44.99	872,854 in Beauty & Personal Care (No7 Lift & Luminate Triple Action Serum 50...	NaN	NaN
7250468162	No7		NaN	["No7 Stay Perfect Foundation now stays perfec...		NaN	All Beauty	\$28.76	956,696 in Beauty & Personal Care (No7 Stay Perfect Foundation Cool Vanilla by No7	NaN	NaN
7367905066	NaN		NaN			NaN	All Beauty	NaN	1,870,258 in Beauty & Personal Care (Wella Koleston Perfect Hair Colour 44/44 Mediu...	NaN	NaN

Figure 1. Product's Detail Data

2. Data Preprocessing

The data obtained is stored in separate tables for each category and also between product details and product reviews, so it requires a merging process first. The first table merging is done by combining all product details into 1 table and also all product reviews into 1 table. The two separate tables are then combined into 1 table based on the productId column contained in both tables. The resulting table will then go through the process of deleting duplicate data.

2.1. Table Merging

The data obtained is stored in separate tables for each category and between product details and product reviews, so it requires a merging process first. The first table merging is done by combining all product details into 1 table and all product reviews into 1 table. The two separate tables are then combined into 1 table based on the productId column contained in both tables. The resulting table will then go through the process of deleting duplicate data.

2.2. Column Modification

The column modifications carried out include the process of filling in null data, deleting columns that will not be used, deleting duplicate data, filtering data to be used in each column, changing column names according to those that will be used, and creating new columns from column values that have been used. There is. The new column created is the popularity column which contains the popularity value of a product and is used for recommendations using popularity-based filtering. Popularity value is obtained from the number of interactions made by the user as a whole on an item (Wang et al., 2019). In the data used in this study, the popularity value is sought by calculating the total reviews on a product multiplied by the percentage of users who review the total number of users. The equation to find the popularity value of a product can be seen in the following equation.

$$P_i = (\sum_{i=1}^n (r + r_n)) \times n \quad (1)$$

From the above equation, P_i is the popularity value obtained from item i . Meanwhile, for n is the number of ratings given to item i , and r is the rating value given to item i .

2.3. Data Filtering

The data obtained still has a cold start problem where there are users who have not given a rating or the rating is still small. To see the difference in the results of the system made using data that has a cold-start problem or data that has gone through the filtering stage to overcome cold-start, then from the data obtained previously, new data will be created which each user or product already has at least 3 rating, 5 rating, and 7 rating. Filtering is done by deleting data that does not meet the rating limit.

3. Data Splitting

The final step for data pre-processing is data splitting. Splitting data is done by dividing the data into train data and test data. This process produces 80% of the data as training data and 20% of the data as test data. Both types of data will be used in the process of making the KNN Collaborative Filtering model and measuring the error rate of the model.

3.1. KNN User Based Collaborative Filtering

Collaborative filtering is divided into two types of algorithms, namely model-based and memory-based (Cacheda et al., 2011). The model-based algorithm uses a latent factor model with a high generalization capacity to predict rating of an item that have not been assigned by the user, the most frequently used example is matrix factorization as in the Singular Value Decomposition algorithm (Veras De Sena Rosa et al., 2020). While the memory-based algorithm uses the similarity function to find similar users or items according to the rating, the similarity between users or items is then the basis for predicting the rating (Zarei & Moosavi, 2019). The most common example of using memory-based is with K-Nearest Neighbors.

This study uses the K Nearest Neighbor (KNN) algorithm as a collaborative filtering algorithm to develop a hybrid model. The KNN algorithm which is a memory based collaborative filtering is used because it has calculations that are not too complex when compared to models based on collaborative filtering such as matrix factorization. K Nearest Neighbor (KNN) is a classification algorithm that works by looking at the K closest neighbors of a sample to determine what category the sample belongs to, if the majority of K of its closest neighbors are in a category, then the sample will be assumed to fall into that category (Bilge et al., 2013). In collaborative filtering that gives recommendations by the similarity of points of interest, the KNN algorithm is suitable to be used to find users or items that have similarities with

each other. KNN Collaborative Filtering combines collaborative filtering algorithms by using the KNN algorithm to find neighbors that have similarities, with the steps taken including: calculating user similarity values, selecting K Nearest Neighbors, and finally predicting scores or ratings based on user similarity (Cui, 2017). These three steps can be summarized into 2 stages as follows:

1) User Similarity Calculation Stage:

Similarity calculations between users are carried out to determine the closest neighbors of a user for further use in the rating prediction stage. This stage begins by creating a user-item matrix that is created by using the user as the index and the item as the column whose value is filled with the rating of an item from user. The example of the user-item matrix can be seen in table 1 below where U is the user, I is the item or product and r_{ui} is the rating of the user U on item I.

Table 1. User – Item Matrix

	I_1	I_2	I_j	I_{n-1}	I_n
U_1	r_{u1}	r_{u2}	r_{uj}	$r_{u(n-1)}$	r_{un}
U_2	r_{21}	r_{22}	r_{2j}	$r_{2(n-1)}$	r_{2n}
.....
U_{n-1}	$r_{(n-1)1}$	$r_{(n-1)2}$	$r_{(n-1)j}$	$r_{(n-1)(n-1)}$	$r_{(n-1)n}$
U_n	r_{n1}	r_{n2}	r_{nj}	$r_{n(n-1)}$	r_{nn}

The matrix that has been created is then used to count the similarity between users based on the rating given to the products. Search similarity between users in this study using Pearson similarity. Pearson Similarity or commonly called Pearson Correlation Coefficient is also an algorithm for finding similarity values of items or users. This algorithm calculates the similarity value based on how well the data or items match on a straight line (Geetha et al., 2018). In Pearson similarity, ratings that will be used as a reference only on the same group of items are rated by the reference user (u) and the target user (v), other items that are only rated by one user will be ignored (B. Guo et al., 2017). The workings of this algorithm begin with finding groups of items that have been assessed by both users, then the rating groups will be the reference for finding similarities between users for user-based collaborative filtering or items for item-based collaborative filtering. This study uses user-based collaborative filtering with the following Pearson similarity equation (Bourkougou & Achbarou, 2018).

$$Sim_{(u,v)} = \frac{\sum_{i \in u,v} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in u,v} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in u,v} (r_{v,i} - \bar{r}_v)^2}} \tag{2}$$

From the equation above, $Sim_{(u,v)}$ is the similarity score of user u and user v. $r_{u,i}$ is the rating from user u to item I, and \bar{r}_u is the average rating from every item that has been rated by user u. While for $r_{v,i}$ and \bar{r}_v , same like the $r_{u,i}$ and \bar{r}_u but for user v.

2) Rating Prediction Stage

The similarity values between users that have been obtained are then used to predict product ratings that have never been rated by users before. The Pearson similarity value that has been obtained previously is then used to create a user-user matrix that shows who a user can be said to be similar which will later be used to find the closest k users (Bourkougou & Achbarou, 2018). The matrix will contain a value between -1 to 1 which is the Pearson value between users obtained in the previous process. A value of 1 means that the user has a high similarity, -1 means that the user is contradictory to each other, and 0 means that the user has no resemblance or relationship at all (Theodorus & Budiyanto Setyohadi, 2016). The matrix can be seen in table 2.

Tabel 2. User – User Matrix

	U_1	U_2	U_j	U_{n-1}	U_n
U_1	1	0	0	0
U_2	0	1	1	0
.....
U_{n-1}	0	1	1	0
U_n	0	0	0	1

The matrix that has been created will make it easier for the system to find K nearest neighbors from a user that will be used for rating prediction. Predicted product ratings that have never been rated by a user will refer to other similar users or to the nearest neighbors of that user. The equation for predicting ratings on KNN User-Based Collaborative Filtering can be seen in the following equation (Zhou et al., 2020).

$$\text{Predict}_{(u,i)} = \bar{r}_u + \frac{\sum_{v=1}^k \text{Sim}(u,v)(r_{v,i} - \bar{r}_v)}{\sum_{v=1}^k |\text{Sim}(u,v)|} \quad (3)$$

From the equation above, $\text{Predict}_{(u,i)}$ is the predicted value of user u rating for item i. \bar{r}_u is the average rating from every item that has been rated by user u. $\text{Sim}_{(u,j)}$ is the similarity value of user u and user v. Then $r_{v,i}$ and \bar{r}_v sequentially are user v rating for item I and the average rating from every item that has been rated by user v.

3.2. Content Based Filtering

Content-based filtering is a method that uses descriptions of items and/or user profiles such as gender, occupation, age, or other descriptions from users to generate a list of recommendations (Geetha et al., 2018). Because the data that is mostly used is text, it is generally necessary to pre-process it first to extract the required digest. Common pre-processing such as keyword extraction, tokenization, stopword removal, stemming, and weighting (Suka Parwita et al., 2018). After pre-processing this data, then the data from the target item will be searched for its similarity value to other items using the similarity function. Recommendations are made based on the similarity value. This study uses the TF-IDF algorithm to pre-process the text into vectors, then the similarity value between the item vectors is calculated using cosine similarity. Content-based filtering in this study includes 2 main stages as follows.

1) Text Vector Creation Stage

The description of the item is created by combining the title, main_cat, and brand fields. The product description that has been obtained will be used to create a vector that represents the product. TF IDF algorithm will be used to create the vector. This algorithm works by calculating the Term Frequency (TF) and Inverse Document Frequency (IDF) values for each word in each document in the corpus (Larasati & Februriyanti, 2021). Term frequency itself is the total frequency of a term appearing in a document, while Inverse Document Frequency is a value to see how often a term appears in a corpus, the more often a term appears, the term will be considered a less important term and the value will be smaller (Kusuma & Musdholifah, 2021). The TF-IDF value of a term in a document is obtained using equation below.

$$\text{TF-IDF}_{d,t} = \text{Freq}_{d,t} \left(1 + \log\left(\frac{N}{dFREQ_t}\right)\right) \quad (4)$$

$\text{TF-IDF}_{d,t}$ in the equation above is the weight of term t on document d. N is the total number of documents. $\text{Freq}_{d,t}$ is the number of terms t in a document d, while $dFREQ_t$ is the number of documents in which term t appears.

2) Item Similarity Calculation Stage

The TF IDF value of that has been obtained from each term will then be used as a vector component to calculate the similarity value between products using cosine similarity algorithm. This algorithm calculates the similarity between items by representing each item as a vector in an N-dimensional space, then the similarity value between the items is measured based on two angles between the vectors, this angle is called the cosine value (Badriyah et al., 2017). Based on arithmetic principles, two vectors are said to be equal if they form an angle of 0 or the cosine value is 1. The item vector that has been formed from the document vectorization process using TF IDF in the previous process can be seen for example in equations (5) and (6) below. where each vector component is the value of the TF IDF of each term, while the equation for knowing the value of cosine similarity between items can be seen in equation below (Parwita, 2019).

$$\vec{a} = (a_1, a_2, a_3, \dots, a_n) \quad (5)$$

$$\vec{b} = (b_1, b_2, b_3, \dots, b_n) \quad (6)$$

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} \quad (7)$$

\vec{a} and \vec{b} is the representation of vector a and vector b, while a_i dan b_i is the component of the vector (the weight of the i-th term of each document). $Sim_{(a,b)}$ in the equation above is the similarity value between item a and item b, and n is the dimension of the vector (the total number of terms in the document).

3.3. Generating Hybrid Recommendations

The two models that have been built before, namely the collaborative filtering model and content-based filtering, as well as the popularity column that has been created in data pre-processing, will be used to make recommendations by using the userId as a parameter to make recommendations for the user. The making of hybrid recommendations in this study uses switching and mixed methods, namely by dividing the recommendations into 2 processes based on the user who will be the target of providing recommendations. If the user is an old user who already exists on the system, recommendations will be executed by combining collaborative filtering, content-based filtering, and also popularity based. However, if the user is not registered in the system, the recommendations that will be given by combining popularity-based and content-based filtering only. The flow of making hybrid recommendations in this study described by Figure 2.

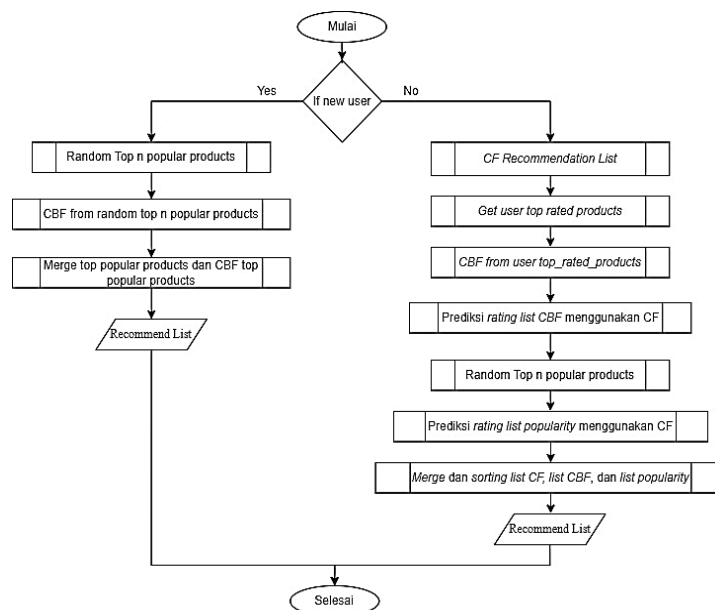


Figure 2. Making of Hybrid Recommendation Flowchart

Making recommendations begins with checking first whether the user is registered in the database or not. If not, a recommendation will be made for new users, while if it is registered, a recommendation will be made for old users.

RESULTS AND DISCUSSION

Mean Absolute Error (MAE) will be used to test hybrid recommendation system. The mean absolute error value is sought to find out how many percentage errors are made by the system in predicting the rating. This test was conducted on the ability of collaborative filtering in predicting ratings because the order of recommendations is based on rating predictions. This test is also a test to see how well the system handles cold start problems.

The mean absolute error test is carried out through several scenarios with the aim of comparing the values generated in each scenario. The scenarios are divided based on the filtering performed on the data used to create a collaborative filtering model that will be used in the hybrid recommendation system. The first scenario is carried out using data that has gone through filtering ratings at least 3 times, the second scenario uses data that has passed filtering ratings at least 5 times, and the last scenario uses data that has passed filtering ratings at least 7 times. Each scenario also has a sub-scenario which is divided based on the use of the min_k and K parameters. This is done to see the best parameter that produces the smallest value. Table 3 below displayed the result of the mean absolute value.

Table 3. Mean Absolute Error Result

No	Dataset Name	Data Count	User Count	Product Count	$Min K$	$K (max)$	MAE
1	Rated3_user_test	4734	2966	2611	1	10	0.566911
2	Rated3_user_test	4734	2966	2611	1	15	0.566911
3	Rated3_user_test	4734	2966	2611	1	25	0.566911
4	Rated3_user_test	4734	2966	2611	3	10	0.566883
5	Rated3_user_test	4734	2966	2611	3	15	0.566883
6	Rated3_user_test	4734	2966	2611	3	25	0.566883
7	Rated3_user_test	4734	2966	2611	5	10	0.566883
8	Rated3_user_test	4734	2966	2611	5	15	0.566883
9	Rated3_user_test	4734	2966	2611	5	25	0.566883
10	Rated5_user_test	2466	1208	1516	1	10	0.506765
...
18	Rated5_user_test	2466	1208	1516	5	25	0.506603
19	Rated7_user_test	1641	649	1112	1	10	0.488732
...
27	Rated7_user_test	1641	649	1112	5	25	0.487553

Table 3 below shows the MAE values of the model created using a combination of the parameters as well as the datasets listed above. The greater the mean absolute error, the more invalid the recommendation because the rating prediction is less accurate.

The values in the table of test results that have been shown previously, show the effect of the hybrid method in dealing with the cold start problem. Mean absolute error test results shown in table 3, shows the comparison of the error rates between the collaborative filtering models used in the hybrid method which also shows the effect of the hybrid method to overcome cold start problem.

Cold start problem can be overcome by using a collaborative filtering model that has gone through the filtering process for the number of ratings first. This is proven by means of testing the mean absolute error where the collaborative filtering model used is the model with the lowest error rate of the three collaborative filtering models that have been made using 3 different datasets before. The model with the

least mean absolute error is the model that is trained using data that has been filtered at least 7 times by rating it compared to data that only goes through filtering 3 and 5 times rating.

The less user-product interaction in the dataset, the higher the error rate of the collaborative filtering model for providing recommendations. Vice versa, the more user-product interactions, the lower the error rate. This is because the collaborative filtering model is made based on giving a rating from a user for a product and then looking for other users who have similarities in giving ratings. If the user has not provided an interaction/rating, it will be difficult to find other users who are similar to the user, so that many recommendations will be invalid judging from the level of system error in predicting product ratings that have never been rated by the user. This is shown through the graph in Figure 3.

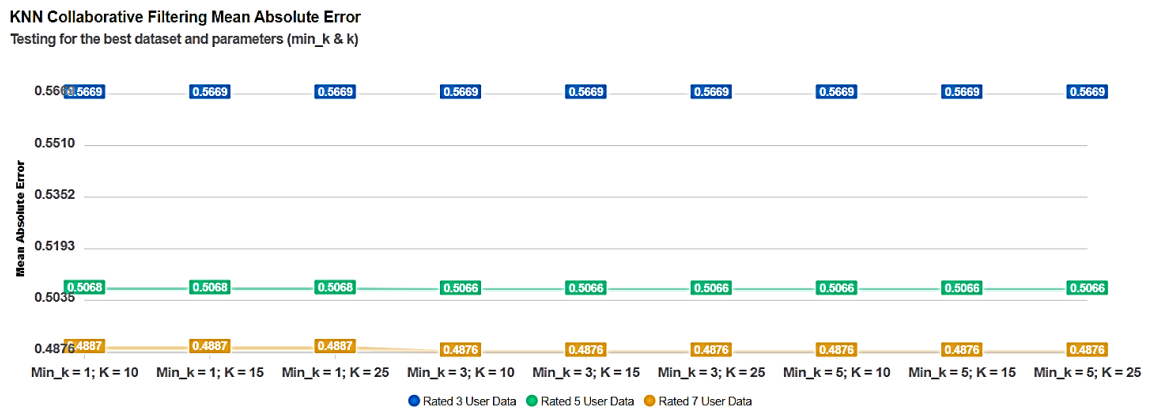


Figure 3 Mean Absolute Error Line Chart

The figure 3 shows, that the model created using the rated3 dataset or at least a user gives a rating for the product as much as 3x, the error rate or MAE value will be greater than the model created using the rated5 and rated7 dataset, where the frequency of users giving more ratings. The graph above also shows that the most optimal min_K parameter is 3, because after trying to use a value of 5 the result does not change, whereas when changing from 1 to 3 there is a change. It is also known that the K parameter does not have much effect on the modeling because the values of 10, 15, and 25 do not change anything in the graph above.

In addition to using filtering on the dataset, cold start in this study was also overcome by using popularity based for new users. New users will receive recommendations generated from the content-based filtering and popularity-based models. This aims to avoid rating prediction errors made to new users due to the lack of interaction made by new users. Because the collaborative filtering model provides recommendations based on rating predictions for products that have never been rated by a user.

Based on the research that has been done and the discussion that has been described, the results of the research can be seen from the testing of models and systems that have been discussed previously in the implementation and results section. The results of these tests and discussions indicate that the use of the hybrid method is able to overcome the cold-start problem that occurs in collaborative filtering

CONCLUSIONS AND RECOMMENDATIONS

Based on the analysis and testing conducted in this study, the use of the hybrid method in the recommendation system is able to overcome the cold start problem that occurs in collaborative filtering. This can be seen from the value of the Mean Absolute Error (MAE) model, where the filtering rating of at least 3 times obtained an MAE value of 0.566883, a minimum of 5 times of 0.506603, and a minimum of

7 times obtaining the smallest MAE value of 0.487553. This value is obtained in the min_k parameter of 5 and K of 25. The less user-item interaction, the higher the prediction error rate, which means there is a cold start problem. The hybrid method in this study is able to overcome this by using switching and mixed methods, namely by not using the collaborative filtering model on new user recommendations or users who have small interactions. New users will get recommendations from a combination of popularity-based and content-based filtering models.

Further research is recommended to use model-based methods such as matrix factorization and neural networks which will make it easier to update the model and also consume less memory compared to the model used in this study to develop hybrid recommendations.

REFERENCES

- Aan, A., & Permana, J. (2019). *Usability Testing Pada Website E-Commerce Menggunakan Metode System Usability Scale (SUS) (Studi Kasus : Umkm Buleleng.Com)*. 8(2), 149–158.
- Arvianti, Q. R., Baizal, Z. K. A., & Tarwidi, D. (2019). Tourism Recommender System Using Item-Based Hybrid Clustering Method (Case Study : Bandung Raya Region). *Journal of Data Science and Its Applications*, 2(2), 95–101. <https://doi.org/10.34818/JDSA.2019.2.35>
- Badriyah, T., Wijayanto, E. T., Syarif, I., & Kristalina, P. (2017). A hybrid recommendation system for E-commerce based on product description and user profile. *7th International Conference on Innovative Computing Technology, INTECH 2017, Intech*, 95–100. <https://doi.org/10.1109/INTECH.2017.8102435>
- Bilge, A., Kaleli, C., Yakut, I., Gunes, I., & Polat, H. (2013). A survey of privacy-preserving collaborative filtering schemes. *International Journal of Software Engineering and Knowledge Engineering*, 23(8), 1085–1108. <https://doi.org/10.1142/S0218194013500320>
- Bourkhouk, O., & Achbarou, O. (2018). Weighting based approach for learning resources recommendations. *International Journal on Informatics Visualization*, 2(3), 104–109. <https://doi.org/10.30630/joiv.2.3.124>
- B.Thorat, P., M. Goudar, R., & Barve, S. (2015). Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. *International Journal of Computer Applications*, 110(4), 31–36. <https://doi.org/10.5120/19308-0760>
- Cacheda, F., Carneiro, V., Fernández, D., & Formoso, V. (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web*, 5(1). <https://doi.org/10.1145/1921591.1921593>
- Cui, B.-B. (2017). Design and Implementation of Movie Recommendation System Based on Knn Collaborative Filtering Algorithm. *ITM Web of Conferences*, 12, 04008. <https://doi.org/10.1051/itmconf/20171204008>
- Geetha, G., Safa, M., Fancy, C., & Saranya, D. (2018). A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System. *Journal of Physics: Conference Series*, 1000(1). <https://doi.org/10.1088/1742-6596/1000/1/012101>
- Guo, B., Xu, S., Liu, D., Niu, L., Tan, F., & Zhang, Y. (2017). Collaborative filtering recommendation model with user similarity filling. *Proceedings of 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference, ITOEC 2017, 2017-Janua*, 1151–1154. <https://doi.org/10.1109/ITOEC.2017.8122536>
- Guo, G. (2013). Improving the performance of recommender systems by alleviating the data sparsity and cold start problems. *IJCAI International Joint Conference on Artificial Intelligence*, 3217–3218.
- Knijnenburg, B. P., Willemsen, M. C., Gantner, Z., Soncu, H., & Newell, C. (2012). Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22(4–5), 441–504. <https://doi.org/10.1007/s11257-011-9118-4>
- Kusuma, H. S., & Musdholifah, A. (2021). Recommendation System for Thesis Topics Using Content-based Filtering. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 15(1), 65. <https://doi.org/10.22146/ijccs.62716>
- Larasati, F. B. A., & Februariyanti, H. (2021). *Sistem Rekomendasi Product Emina Cosmetics Dengan Menggunakan Metode Content-Based Filtering*. 4(1).
- Mobasher, B. (2007). Data mining for Web personalization. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4321 LNCS, 90–135. https://doi.org/10.1007/978-3-540-72079-9_3
- Ni, J., Li, J., & McAuley, J. (2020). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural*

- Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 188–197. <https://doi.org/10.18653/v1/d19-1018>
- Noviansyah, M. R., Suharso, W., Chandranegara, D. R., Azmi, M. S., & Hermawan, M. (2019). Sistem Pendukung Keputusan Pemilihan Laptop Pada E-Commerce Menggunakan Metode Weighted Product. *Prosiding SENTRA (Seminar Teknologi Dan Rekayasa)*, 0(5), 43–53. <http://research-report.umm.ac.id/index.php/sentra/article/view/3025>
- Parwita, W. G. S. (2019). Pengujian Akurasi Sistem Rekomendasi Berbasis Content-Based Filtering. *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, 14(1), 27. <https://doi.org/10.30872/jim.v14i1.1272>
- Prasetya, C. S. D. (2017). Sistem Rekomendasi Pada E-Commerce Menggunakan K-Nearest Neighbor. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 4(3), 194. <https://doi.org/10.25126/jtiik.201743392>
- Suka Parwita, W. G., Prami Swari, M. H., & Welda, W. (2018). Perancangan Sistem Rekomendasi Dokumen Dengan Pendekatan Content-Based Filtering. *Computer Engineering, Science and System Journal*, 3(1), 65. <https://doi.org/10.24114/cess.v3i1.7855>
- Theodorus, A., & Budiyanto Setyohadi, D. (2016). User-Based Collaborative Filtering Dengan Memanfaatkan Pearson- Correlation Untuk Mencari Neighbors Terdekat Dalam Sistem Rekomendasi. *Thesis Magister Teknologi Informasi Universitas Atma Jaya Yogyakarta*, 1–6. <http://e-journal.uajy.ac.id/8924/>
- Veras De Sena Rosa, R. E., Guimarães, F. A. S., Mendonça, R. D. S., & Lucena, V. F. de. (2020). Improving Prediction Accuracy in Neighborhood-Based Collaborative Filtering by Using Local Similarity. *IEEE Access*, 8, 142795–142809. <https://doi.org/10.1109/ACCESS.2020.3013733>
- Wang, C. D., Deng, Z. H., Lai, J. H., & Yu, P. S. (2019). Serendipitous recommendation in e-commerce using innovator-based collaborative filtering. *IEEE Transactions on Cybernetics*, 49(7), 2678–2692. <https://doi.org/10.1109/TCYB.2018.2841924>
- Zarei, M. R., & Moosavi, M. R. (2019). A Memory-Based Collaborative Filtering Recommender System Using Social Ties. *4th International Conference on Pattern Recognition and Image Analysis, IPRIA 2019*, 263–267. <https://doi.org/10.1109/PRIA.2019.8786023>
- Zhou, W., Li, R., & Liu, W. (2020). Collaborative Filtering Recommendation Algorithm based on Improved Similarity. *Proceedings of 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference, ITOEC 2020, Itoec*, 321–324. <https://doi.org/10.1109/ITOEC49072.2020.9141788>