JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1041-1049
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6049

Collaborative Filtering Based Food Recommendation System Using Matrix Factorization

Muhammad Bayu Samudra Siddik*, Agung Toto Wibowo

School of Computing, Informatics, Telkom University, Bandung, Indonesia
Email: 1,*bayusiddik@student.telkomuniversity.ac.id, 2agungtoto@telkomuniversity.ac.id
Correspondence Author Email: bayusiddik@student.telkomuniversity.ac.id

Abstract−A recommendation system is a method that provides suggestions of items that might users like. There are many domains that can be recommended, one of the most demanded domains by users today is food. In the era of big data, food choices from the large amount of data make it difficult for users to choose the right food for them. The collaborative filtering (CF) approach is considered capable of providing accurate and high quality item suggestions. One of the algorithms that can provide good performance results from the CF approach is Matrix Factorization (MF). This study aims to test a dataset that contains product ratings of food items using three MF algorithms, which are Singular Value Decomposition (SVD), SVD with Implicit Ratings (SVD++), and Non-Negative Matrix Factorization (NMF). Different latent factors are also used for the purpose of improving the performance of the proposed recommendation system algorithm. The dataset used is Amazon Fine Food Reviews. The study shows NMF and SVD++ as the best algorithm for generating user rating predictions for items. NMF has the smallest average prediction error as measured by MAE which is 0.7311. While SVD++ obtains the smallest prediction error value of 1.0607 as measured using RMSE. In addition to these results, the top-n evaluation also shows that the proposed algorithm performs quite well. The hit ratio value for each different n-item always increases proportionally to the number of recommended n-items. The highest hit ratio value is generated from the SVD++ algorithm of 0.0025 on n-item recommendations of 25 items. Overall it can be said that the proposed algorithm has good performance in providing item recommendations.

Keywords: Food Recommender System; Collaborative Filtering; Matrix Factorization; MAE; RMSE.

1. INTRODUCTION

A recommendation system is a method that can provide suggestions based on favorite and useful items for users [1]–[3]. The recommendations made are meant to aid users in making decisions about things like what to buy, what news to read, and what music to listen to. Food is certainly one of the domains that can be recommended to users. Food is an essential component of human life [4], which makes the food domain in much demand by users. Nowadays, food choices are important in meeting a variety of needs such as nutrition, calories, and taste [5].

In the era of big data, there are many food options available both on the internet and in commercial restaurants. This huge selection of food makes it difficult for users to choose one of the many food items available. In order to provide food suggestions to users, a food recommendation system is created that can provide recommendations according to user preferences. The food recommendation system offered is collaborative filtering (CF) [6]. The CF method suggests products to active users based on what other users who share their likes have liked (similarity) [2], [3]. The fundamental idea of CF is that users with similar behavior patterns are more likely to share similar preferences, so a user's past choices can be used to make recommendations to another user [7]. The most widely used algorithm of the CF approach method is Matrix Factorization (MF) [8].

In our study conducted, we propose three MF derivative algorithms to create a food recommendation system model based on the CF approach. To build a good food recommendation model, we limited this study to using the Amazon Fine Food Reviews dataset. We chose this dataset because the features contained in the data are in accordance with the model built. In addition, the MF derivative algorithms that we use are Singular Value Decomposition (SVD) [9], SVD with Implicit Ratings (SVD++) [10], and Non-Negative Matrix Factorization (NMF) [11].

The purpose of our study is to implement three types of CF-based MF algorithms in a food recommender system and compare the performance results with the BaselineOnly [12] algorithm based on the values obtained from the error evaluation results. The performance of the implemented CF method is measured using Mean Absolute Error (MAE) [13] and Root Mean Squared Error (RMSE) [14]. Furthermore, to evaluate the model, we apply the top-*n* recommendation and measure the results using the hit ratio [15] so that we know the performance of the model that has been built.

The study conducted is guided by previous studies on recommender systems that use a collaborative filtering approach. Basically, there are various kinds of recommendation system algorithms to suggest items to users. Anand, R. and Beel, J. [16] conducted a study to compare the performance of prediction algorithms using several recommendation system algorithms. One of the algorithms they use is the BaselineOnly algorithm. They used three different datasets and two evaluation matrices, MAE and RMSE. They also measured the estimated running time of the program. Their results show that the BaselineOnly algorithm is able to produce a quite good prediction with MAE = 0.7479 and RMSE = 0.9433 run from the MovieLense100K dataset. Tran, D.T. and Huh, J.H. [17] also conducted a study using the same BaselineOnly algorithm. Their research also shows quite good

Submitted: **01/04/2023**; Accepted: **14/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1041-1049
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6049

results, by running a train size of 90%, the BaselineOnly algorithm used has the lowest error rates value. The MAE value obtained is 0.79, MSE is 1.13, and RMSE is 1.06.

The study [18] said that one of the most successful methods for creating recommendation systems is the collaborative filtering method. The study conducted by Mustafa, N. et al. presents three primary types of CF approaches which is memory-based, model-based, and hybrid collaborative filtering algorithms to discuss in more detail the challenges, the solution to the challenge, and an analysis of the algorithm's challenge-solving potential and predictive performance.

Rajabpour, N. et al. [19] developed an application aimed at tourists arriving in a country. The application provides food recommendations to tourists according to their preferences. A food recommendation system designed with collaborative filtering and content-based filtering approach. Then the built system is tested by a number of users before being appraised through empirical experiments. In addition, precision and recall matrices are used to measure the integrity of the recommendation system algorithm. The results of this study provide fairly good accuracy in recommending food items according to user preferences, which is 86.3%. Yu, C. et al. [20] said that a good food recommendation should not only make users feel satisfied with the suggested items but should also be able to increase the sales revenue of the recommended products. In their study, they also built a food recommendation system using an improved collaborative filtering approach. The recommendation system they built was successfully implemented on the Zhuoji booking platform and received good feedback from users. This can be achieved because the model they use has a coverage ratio value of 83.2%. But besides the large coverage ratio value they have, the accuracy value is only able to reach 74.3%.

Vivek, M. B. et al. [3] conducted a study related to food recommendation systems with a machine learning based collaborative filtering approach. They apply the Tanimoto Coefficient Similarity and LogLikelihood Similarity methods to calculate item-based similarity based on user preferences. Meanwhile, to calculate user based similarity, they use Euclidean Distance and Pearson Correlation. Their study concludes that a user-based approach can provide better recommendations than an item-based approach. This can happen because there is high interaction between users and the items of data they use. The last, Massimo, D. et al. [21] built a food recommendation system by inputting user preferences as ratings and tags. Matrix Factorization serves as the recommended algorithm. The evaluation results measured by MAE showed a poor value. The MF user tags implemented are only worth 0.979 which shows that the performance of the method used has not met the expected prediction results.

Based on the various studies described above, the BaselineOnly and Matrix Factorization algorithms have a good enough ability to recommend items. From the above studies as well, the Matrix Factorization algorithm has the possibility of producing better recommendations than the BaselineOnly algorithm. In addition, Matrix Factorization with CF approach is widely used in current recommender systems [13]. So in this study we focus on comparing the performance results of the BaselineOnly and Matrix Factorization algorithms that we propose using data from Amazon e-commerce that we get from Kaggle.com.

2. RESEARCH METHODOLOGY

2.1 Model Design

The food recommendation system created requires several stages that must be performed well. In Figure 1 there are stages of the food recommendation system using the CF method which starts from load datasets. In this section, we load the dataset that has been downloaded from Kaggle.com. After that, data preprocessing is carried out, this stage aims to make the data used easy to read and facilitate the data analysis process. There are several steps we took to improve the structure of the data in data preprocessing. Furthermore, the collaborative filtering modeling stage using the matrix factorization algorithm, this stage consists of two steps, namely updating the P and Q matrices, where P represents the relationship between users and features while Q represents the relationship between items and features. After that, the rating prediction step is continued, this stage aims to get the approximate value between the user and the item. In the final stage, we evaluate the predictive values generated by the algorithm using MAE and RMSE measurement.

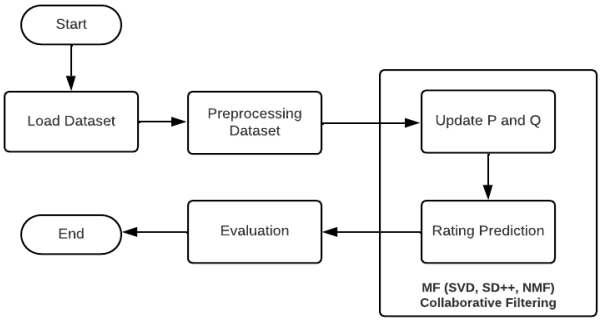

Figure 1. Flowchart Design of the conducted research.

Submitted: **01/04/2023**; Accepted: **14/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1041-1049
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6049

2.2 Load Dataset

The data used in this study is the Amazon Fine Food Reviews Dataset, which is review data for food products sold on Amazon e-commerce. The dataset was obtained from website Kaggle.com which is https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews in csv format. This dataset has the attributes and variables needed for the study. There are 10 columns and more than 500,000 rows in the dataset that will be processed into the data preprocessing to make the data easier to analyze. The column attributes contained in the dataset include Product ID which consists of 74,258 data. The product data referred to here is food data. Then there is User ID data which consists of 256,059 data. Next there is Product Rating data which consist 568,454 data. The Product Rating here is the interaction between users who give a value or rating to food products. Lastly, there is a collection of Metadata consisting of 568,454 data. In addition, dataset details can be seen in Table 1.

Table 1. Summary of the data used in this experiments

Attribute Name	Number of Data
Product ID	74,258
User ID	256,059
Product Rating	568,454
Metadata (Helpfulness Numerator and Denominator, Time, Text)	568,454

2.3 Preprocessing Data

In the data preprocessing, we perform the attribute selection process from the required dataset. Data preprocessing is carried out to improve incomplete and irregular data structures which involve validation and data imputation processes. From the details of the dataset in Table 1, the attributes needed for this study are Product ID, User ID, and Product Rating. Attributes incorporated in metadata are not required for the model-based collaborative filtering MF algorithm. We split the dataset using the cross validation (CV) method, which is a method used to evaluate the performance of an algorithm by splitting the data into two sections, which are training data and testing data. We split the dataset with 80% training data and 20% testing data. Five iterations were performed to maximize the estimated error rate. The original dataset that we used has 256,059 userId, 74,258 itemId, and 568,454 rows of interaction ratings from users for certain items. But in top-*n* recommendation modeling, we scrape the data into 31,032 userId, 16,926 itemId, and 37,000 user rating interaction rows. We decided to do this because the complete data that we use is too large to be processed in our system. An example of the dataset after preprocessing can be seen in Table 2. The userID column contains the identity of the user, the itemID column contains the identity of the item, and the rating column contains the item rating rated by the user.

Table 2. An example of a dataset after preprocessing**.**

userID	itemID	rating
256056	74254	5
29529	74255	2
256057	74256	5
163453	74256	5
256058	74257	5

2.4 Matrix Factorization Algorithms

This section consists of two main parts, namely updating the P and Q matrices and then calculating the prediction matrix to get the user's prediction value for the item. In this section, we use three MF derived algorithms from the surprise library [22], which are SVD, SVD++, and NMF. To generate a rating prediction matrix, MF performs a number of steps, beginning with the explicit reference step, which initializes the user and item matrices randomly. After that, MF goes through the rating matrix step which is obtained from the user's assessment of the items in the previous step. The next step is to update both latent factor matrices P and Q, using the Stochastic Gradient Descent (SGD) algorithm, an optimization algorithm used to find the most suitable model parameters between the predicted value and the actual value. The last step is to form a prediction rating matrix which is obtained by multiplying both latent factor matrices in the previous P and Q update steps. The purpose of MF is to reduce the loss function (rating difference between the actual matrix and the prediction matrix) which is processed at the P and Q update steps using the SGD algorithm.

2.4.1 Singular Value Decomposition

Singular Value Decomposition (SVD) is the earliest version of MF [13]. In collaborative filtering-based recommendation systems, SVD is one of the most popular MF algorithms [23]. SVD is used to factor an X matrix into three matrices. In general, the SVD formula can be written as follows:

$$X = P \cdot \Sigma \cdot Q^T \quad (1)$$

Submitted: **01/04/2023**; Accepted: **14/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1041-1049
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6049

Where X denotes the original matrix, P denotes the orthogonal matrix, Σ denotes the diagonal matrix, and Q^T denotes the transpose of the orthogonal matrix.

From equation (**Error! Reference source not found.**), we can see that SVD is an algorithm used to reduce a dimension. Dimensionality reduction techniques such as SVD [24] can quickly generate good quality item recommendations. An illustration of SVD can be seen in Figure 2.

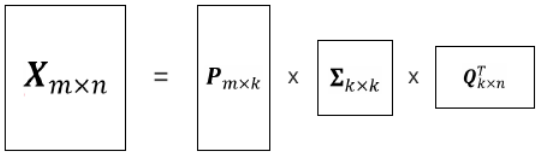

Figure 2. SVD performs dimension reduction on a matrix that is split into several matrices.

From Figure 2 we can see that the SVD is generated from the reduction of the several matrices that have different dimensions. The orthogonal matrix $\boldsymbol{P}$ has dimension $m \times k$, a diagonal matrix $\boldsymbol{\Sigma}$ has dimension $k \times k$, and the transpose of an orthogonal matrix $\boldsymbol{Q}$ has dimension $k \times n$.

2.4.2 SVD with Implicit Ratings

SVD with Implicit Ratings (SVD++) is a derivative of the SVD model [23] that considers implicit ratings from the user. Implicit ratings provided can be in the form of browsing history or items that the user has rated, an illustration is in Figure 3.

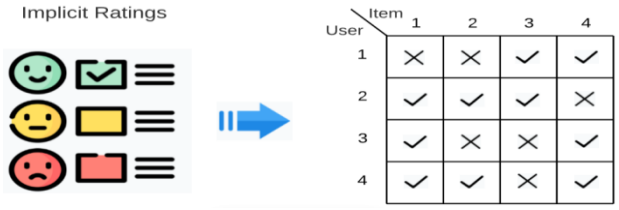

Figure 3. An illustration of SVD++ that considers implicit ratings to provide optimal recommendation results.

Implicit ratings has proven useful for prediction algorithms [25], as general recommendation systems will achieve better performance if more information is included [13]. The implicit rating given is in the form of user feedback or evaluation. In Figure 3, there are three types of implicit ratings given by users. A green smiling face indicates that a user has the highest rating for an item. A yellow neutral face indicates that the user has an average rating for an item. And finally, a red frowning face indicates that the user dislikes the item they have rated. Furthermore, SVD++ can combine the average item ratings, user-item bias, and implicit rating information to provide better recommendation accuracy results [13], [23]. The SVD++ equation can be written in equation (**Error! Reference source not found.**) as follows:

$$\hat{\boldsymbol{r}}_{\boldsymbol{ui}} = \boldsymbol{\mu} + \boldsymbol{b_i} + \boldsymbol{b_u} + \boldsymbol{q_i^T}\left(\boldsymbol{p_u} + |\boldsymbol{N(u)}|^{-\frac{1}{2}}\sum_{\boldsymbol{j\epsilon N(u)}}\boldsymbol{y_j}\right) \quad (2)$$

Where $\hat{r}_{ui}$ denotes the user rating predictions, μ denotes the global average rating, b_i denotes the item offset matrix (i), b_u denotes the user offset matrix (u), q_i^T denotes the transpose matrix of latent-features item (i), p_u the user latent features matrix (u), $|N(u)|^{-\frac{1}{2}}$ denotes the set of items from the user history (u), y_j denotes the implicit ratings item, $|N(u)|^{-\frac{1}{2}}\sum_{j\epsilon N(u)} y_j$ denotes the eigenvector of the implicit user ratings (u), and $p_u + |N(u)|^{-\frac{1}{2}}\sum_{j\epsilon N(u)} y_j$ denotes the latent-features matrix of the user (u) that provides implicit ratings.

2.4.3 Non-Negative Matrix Factorization

Non-Negative Matrix Factorization (NMF) also factorizes the original rating matrix into two matrices. NMF is an algorithm that implements a non-negative low-rank matrix into two latent factor matrices that also have non-negative elements [11]. In general, the NMF formula can be written as follows:

$$V \approx P \times Q \quad (3)$$

Where V denotes the original matrix, P and Q denote the latent features matrices between users and items. From equation (**Error! Reference source not found.**), it can be seen that the purpose of the NMF is to determine the two non-negative matrix factors P and Q from the non-negative matrix V [13].

2.4.4 Benchmark Method

The comparison method used in this study is the BaselineOnly [22] model. In the BaselineOnly model, to calculate the rating prediction can be written in equation (**Error! Reference source not found.**) as follows:

Submitted: **01/04/2023**; Accepted: **14/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1041-1049
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6049

$$\hat{r}_{ui} = b_{ui} = \mu + b_u + b_i \quad (4)$$

Where $\hat{r}_{ui}$, bui denotes the prediction rating, μ denotes the global average rating, b_u and b_i denotes the deviation value from the user (u) and the item (i). The baseline prediction for the unknown rating $\hat{r}_{ui}$ is denoted by b_{ui}. Then if the user u is unknown, then the bias b_u is considered 0 (zero). The same applies to item i with bias b_i.

2.5 Evaluation

There are two types of evaluation metrics that we use in this study to measure the proposed algorithm's performance, they are Mean Absolute Error and Root Mean Squared Error. Both evaluation metrics were chosen because they fit the model we built.

2.5.1 Mean Absolute Error

Mean Absolute Error (MAE) is one of the metric evaluation methods used to measure the accuracy value of a prediction model. MAE is developed from an average error metric that is commonly used in measuring multivariate vectors (vector-to-vector) [26]. The MAE value represents the average absolute error between the predicted value and the actual value. Intuitively, MAE calculates the average error value by giving equal weight to all data, as we can see in equation (**Error! Reference source not found.**) as follows:

$$\boldsymbol{MAE = \frac{1}{n}\sum_{i=1}^{n}|p_{ui} - r_{ui}|} \quad (5)$$

Where MAE denotes the Mean Absolute Error value, n denotes the total number of data, p_{ui} denotes the predicted value of the user (u) against the item (i), r_{ui} denotes the real value of the user (u) against the item (i).

2.5.2 Root Mean Squared Error

Root Mean Squared Error (RMSE) is a measurement model that emphasizes significant errors. RMSE is a similar measurement to MAE that has the same scale as the original rating. But the errors in RMSE are squared first and then summed. The RMSE can be defined as the square root of the average of the squared differences between predicted and actual values [27]. The RMSE formula can be written in equation (**Error! Reference source not found.**) as follows:

$$\boldsymbol{RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_{ui} - r_{ui})^2}} \quad (7)$$

Where $RMSE$ denotes the Root Mean Squared Error value, n denotes the total number of data, p_{ui} denotes the predicted value of the user (u) against the item (i), r_{ui} denotes the real value of the user (u) against the item (i).

3. RESULT AND DISCUSSION

We ran the test using k-fold cross validation with k = 5, the number of epochs = 20, then from the total 568,454 data we also split the data by 80% for training data and 20% for testing data. The three algorithm models are trained in one testing process so that we can find out the difference in their performance in rating prediction. The output of the developed model is then measured using two types of evaluation, which are MAE and RMSE implemented using the surprise library [22]. If the evaluation value is getting smaller, it shows that the prediction error value is also getting smaller and the prediction accuracy is getting better. But if the evaluation value is greater, then the prediction error value is also large and the accuracy of the model used is getting worse. The performance results of the three algorithms are summarized in Table 3.

Table 3. Algorithm performance results using num of epochs = 20, k-fold = 5 and various of n-latent factors.

(a) MAE

Algorithms	n-factors	*fold-k*					Average
		1	2	3	4	5	
BaselineOnly	-	0.9009	0.9041	0.9007	0.9042	0.9039	0.9027
SVD	10	0.8212	0.8244	0.8241	0.8207	0.8201	0.8221
	15	0.8173	0.8177	0.8181	0.8142	0.8154	0.8165
	20	0.8115	0.8126	0.8136	0.8104	0.8104	0.8117
SVD++	10	0.7602	0.7623	0.7575	0.7650	0.7632	0.7616
	15	0.7561	0.7584	0.7581	0.7562	0.7568	0.7571
	20	0.7516	0.7495	0.7548	0.7564	0.7528	0.7530
NMF	10	1.1767	1.1697	1.1727	1.1783	1.1785	1.1752
	15	0.8698	0.8664	0.8610	0.8675	0.8636	0.8656
	20	**0.7302**	**0.7311**	**0.7320**	**0.7325**	**0.7295**	**0.7311**

Submitted: **01/04/2023**; Accepted: **14/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1041-1049
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6049

(b) RMSE

Algorithms	n-factors	*fold-k*					Average
		1	2	3	4	5	
BaselineOnly	-	1.1658	1.1704	1.1681	1.1709	1.1711	1.1693
SVD	10	1.1073	1.1114	1.1097	1.1037	1.1045	1.1073
	15	1.1046	1.1064	1.1031	1.1024	1.1003	1.1034
	20	1.1005	1.0998	1.1026	1.0970	1.0981	1.0996
SVD++	10	1.0642	1.0648	1.0632	1.0713	1.0681	1.0663
	15	1.0629	1.0629	1.0646	1.0625	1.0630	1.0631
	20	**1.0588**	**1.0567**	**1.0631**	**1.0653**	**1.0599**	**1.0607**
NMF	10	1.4041	1.3959	1.3981	1.4050	1.4064	1.4019
	15	1.1773	1.1745	1.1716	1.1767	1.1682	1.1737
	20	1.1208	1.1191	1.1224	1.1231	1.1169	1.1205

Table 3 summarizes the performance of prediction results for each algorithm. N-factor indicates the number of latent features for each algorithm. Then, fold-k indicates the iterations performed on each algorithm. After that, average indicates the mean value of all iterations. From Table 3 (a), it can be seen that from several differences in n-factor and fold-k, the SVD algorithm is able to produce the best prediction value at n-factor = 20 then iterations at fold-k = 4 and fold-k = 5 with each value of MAE = 0.8104. The best average MAE value of the SVD algorithm is 0.8117. Furthermore, for the SVD++ algorithm, it produces the best prediction value with n-factor = 20 and iteration on fold-k = 2 with MAE value of 0.7495. The best average MAE value for the SVD++ algorithm is 0.7530. In addition, for the last matrix factorization algorithm in Table 3 (a), NMF produces the best prediction value with n-factor = 20 and k-fold = 5 with MAE value of 0.7295. The best average MAE value for the NMF algorithm is 0.7311.

In terms of RMSE which is presented in Table 3 (b), each algorithm has different values from the previous evaluation results. The SVD algorithm produces the best prediction value with n-factor = 20 and iteration on fold-k = 4 with RMSE value of 1.0970. The best average RMSE value for all iterations of the SVD algorithm is 1.0996. Then, for the SVD++ algorithm, the best prediction value is found with n-factor = 20 and iteration on fold-k = 2, with RMSE value of 1.0567. The average RMSE value for all iterations of the SVD++ algorithm is 1.0607. Lastly, the best predicted value for the NMF algorithm is obtained with n-factor = 20 and k-fold = 5, with an RMSE value of 1.1169. Of all the iterations, the best average RMSE value for the NMF algorithm is 1.1205.

Based on the error evaluation results for each algorithm in Table 3, we can see that there are differences in the performance results of the algorithms that provide rating predictions. Table 3 (a), shows that the best prediction algorithm measured by MAE is NMF with an average value of 0.7311, while in Table 3 (b), the best prediction algorithm as measured by RMSE is SVD++ with an average value of 1.0607. This can happen because both evaluation metrics have different parameters for making measurements, MAE measures the absolute error value while RMSE measures the squared error value. From the MAE, the NMF algorithm is able to produce the smallest error value than other algorithms. The resulting value can be said to be quite good because it is still far from the value of 1.0.

On the other side, we see two identical algorithms, they are SVD and SVD++. Both also produce a good MAE value, but SVD++ can be better than SVD. In Table 3 (a), SVD has the smallest average MAE value of 0.8117 while SVD++ has the smallest average MAE value of 0.7530. The SVD++ algorithm is able to outperform SVD because SVD++ considers ratings from the user so that the prediction performance is better than the SVD base model.

From the MAE result in Table 3 (a), we can also see that the three algorithms proposed in this study are able to outperform the performance of the BaselineOnly model. But there is something interesting about the RMSE results in Table 3 (b), the baseline model used is able to outperform the performance of the NMF algorithm with the number of latent factors n = 10 and n = 15. But at a latent factor of n = 20, NMF again outperforms the performance of the baseline model. From this, we can know that latent factors affect the performance of the algorithm in producing a good item prediction. From Table 3 presented, it is known that a larger number of latent factors will make the algorithm perform better. Also from the overall results shown in Table 3, it is known that MAE always gives a good error evaluation value for each predictive algorithm model built.

3.1 Top-N Recommendation

A good recommendation system must be able to find a set of *n*-items that might interest users. For this purpose, we evaluate each recommendation algorithm using top-*n* recommendations, to determine the quality of the recommendation system. To perform the evaluation, we used the three algorithms we proposed earlier and one baseline algorithm. To study recommendation performance against items, we use the Leave One Out Cross Validation evaluation protocol [28] from the surprise library [22]. We split the dataset into two sections: the

Submitted: **01/04/2023**; Accepted: **14/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1041-1049
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6049

training set and the testing set. From this data, we leave one observation data taken from the testing data. From this one ”out” data observation taken from the testing set, this method is called leave-one-out.

3.2 Hit Ratio

To evaluate our top-*n* model, we calculated it using the hit ratio which is also implemented with the surprise library [22]. Where if the user provides an interaction by rating one of the *n* items that are recommended, it is considered a ”hit”. To calculate the hit ratio we use all the items in the training data. Because it uses leave-one-out cross validation, there will be one item removed intentionally. Then the other items are used to recommend *n* items in the top-*n* recommendation model. If the resulting top-*n* recommendation list contains items that were removed from the training data, then it counts as a hit, if not, it does not count as a hit. The formula for calculating the hit ratio can be written in equation **Error! Reference source not found.** as follows.

$$HR = \frac{u_{hit}}{u_{all}} \tag{8}$$

Where HR denotes the hit ratio value, u_{hit} denotes the number of users whose answers are in top-*n* list (hits), and u_{all} denotes the total number of users in the testing dataset.

In our top-*n* model, we used various values of *n*. Details and evaluation results of the top-*n* model that we built, can be seen in Table 4. To find out whether the performance of the algorithm is good or not, we can see it from the resulting HR value. The value of HR = 1.0 indicates that the algorithm has good performance because it can recommend hidden items. While, if the value of HR = 0.0 indicates that the algorithm has poor performance because it does not recommend any hidden items. From the evaluation results shown in Table 4, there are some differences in the HR values resulting from each algorithm. At n-items = 10, SVD produced the highest HR value of 0.0012, which produces the highest HR value at n-items = 15 is the BaselineOnly model of 0.0016. While at n-items = 20, there are two algorithms that produce the highest HR values, which are BaselineOnly and SVD each of 0.0020. Last, at the largest n-item value for n-items = 25, the highest HR value is obtained by SVD++ of 0.0025. From these results, the SVD and BaselineOnly algorithms are able to provide two superior values compared to the other algorithms. The SVD algorithm produces the highest HR values for n-items = 10 and n-items = 20. Meanwhile, the BaselineOnly algorithm produced the highest HR values for n-items = 15 and n-items = 20.

Table 4. Hit ratio performance over test set using different n-items values.

Algorithm	*n-items*			
	10	**15**	**20**	**25**
BaselineOnly	0.0010	**0.0016**	**0.0020**	0.0023
SVD	**0.0012**	0.0015	**0.0020**	0.0023
SVD++	0.0010	0.0015	0.0019	**0.0025**
NMF	0.0003	0.0005	0.0008	0.0010

In addition, Figure 4 shows that if the value of top-n recommended items gets larger, then the hit rate value gets higher. This happens because the probability of a correct answer on the recommended *n*-items also increases. From Figure 4, it shows a line chart of hit ratio that keeps increasing proportionally to the number of n-items. The BaselineOnly, SVD, and SVD++ algorithms all have hit ratio values that are close to each other. But at larger n-items, SVD++ outperforms all existing recommendation algorithms. From Table 4, it is known that SVD++ has the highest hit ratio with a value of 0.0025 which is in the number of *n*-items = 25.

On the other hand, the hit ratio results of the NMF algorithm are always at the bottom compared to the other algorithms as shown on the line chart in Figure 4. Although the results also always increase, but from all the results shown the NMF algorithm cannot outperform the hit ratio results of existing algorithms. This happens because the NMF Algorithm tends to be unable to capture complex patterns in the data. This makes the predictions generated by the NMF algorithm tend to be inaccurate, so the resulting hit ratio is also low. But from the overall results obtained, all the proposed algorithms can make recommendations well. In addition, the algorithm can provide recommendations for hidden items as evidenced by the HR value of each algorithm.

Submitted: **01/04/2023**; Accepted: **14/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1041-1049
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6049

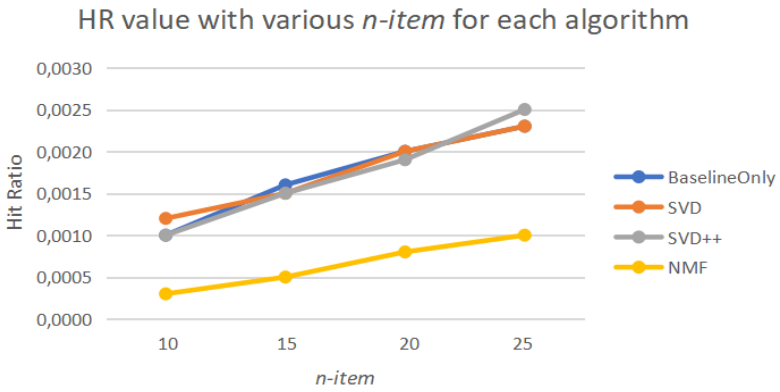

Figure 4. The HR evaluation results with different *n-item* values for each algorithm.

4. CONCLUSION

Based on the evaluation results and comparison of prediction error values that we have performed, we conclude that the recommendation system of the three algorithms we propose is able to predict well. From the evaluation results using 5 iterations of cross validation, the three algorithms we propose can outperform the baseline algorithm's predictive performance. From the results of the MAE evaluation, the prediction algorithm that produces the best performance in our food recommendation system is NMF. Out of 5 iterations, NMF always outperforms the performance of other algorithms with an MAE value of 0.7295 at fold-k = 5 and n-factor = 20. The average MAE value of the NMF algorithm is = 0.7311. While the algorithm that provides the best prediction of RMSE results is SVD++. Throughout the 5 iterations, SVD++ consistently outperformed the other algorithms with an RMSE value of 1.0567 on fold-k = 2 and n-factor = 20. The average RMSE value of the SVD++ algorithm is = 1.0607. In addition, all our proposed algorithms are able to provide hidden item recommendations as evidenced by the HR value that always increases with the number of *n*-items recommended. The BaselineOnly algorithm has two HR values found at n-item = 15 with HR = 0.0016 and at n-item = 20 with HR = 0.0020. The SVD algorithm also has two high hit ratio values found at n-item = 10 with HR = 0.0012 and at n-item = 20 with HR = 0.0020. Then the SVD++ algorithm has the highest hit ratio value of all existing values at n-item = 25 with a value of HR = 0.0025. Finally, although the NMF algorithm never gets a high HR value, the HR value generated from the NMF algorithm is also quite good. This result shows that the recommendation system we built has good performance. The suggestion that we can give to this study is that the study only uses three types of matrix factorization algorithms. So future work can be developed by adding several other types of matrix factorization algorithms to see a wider comparison of prediction results.

REFERENCES

[1] K. Haruna *et al.*, "Context-aware recommender system: A review of recent developmental process and future research direction," *Applied Sciences*, vol. 7, no. 12, p. 1211, 2017.
[2] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: introduction and challenges," *Recommender systems handbook*, pp. 1–34, 2015.
[3] M. B. Vivek, N. Manju, and M. B. Vijay, "Machine learning based food recipe recommendation system," in *Proceedings of International Conference on Cognition and Recognition: ICCR 2016*, Springer, 2018, pp. 11–19.
[4] W. Wang, L.-Y. Duan, H. Jiang, P. Jing, X. Song, and L. Nie, "Market2Dish: health-aware food recommendation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 1, pp. 1–19, 2021.
[5] A. Al-Ansi, H. G. T. Olya, and H. Han, "Effect of general risk on trust, satisfaction, and recommendation intention for halal food," *Int J Hosp Manag*, vol. 83, pp. 210–219, 2019.
[6] Y. Afoudi, M. Lazaar, and M. Al Achhab, "Collaborative filtering recommender system," in *Advanced Intelligent Systems for Sustainable Development (AI2SD'2018) Volume 5: Advanced Intelligent Systems for Computing Sciences*, Springer, 2019, pp. 332–345.
[7] D. Bianchini, V. De Antonellis, N. De Franceschi, and M. Melchiori, "PREFer: A prescription-based food recommender system," *Comput Stand Interfaces*, vol. 54, pp. 64–75, 2017.
[8] R. Mehta and K. Rana, "A review on matrix factorization techniques in recommender systems," in *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*, IEEE, 2017, pp. 269–274.
[9] X. Yuan, L. Han, S. Qian, G. Xu, and H. Yan, "Singular value decomposition based recommendation using imputed data," *Knowl Based Syst*, vol. 163, pp. 485–494, 2019.
[10] J. Jiao, X. Zhang, F. Li, and Y. Wang, "A novel learning rate function and its application on the SVD++ recommendation algorithm," *IEEE Access*, vol. 8, pp. 14112–14122, 2019.
[11] N. Ifada, D. R. M. Alim, and M. K. Sophan, "NMF-based DCG Optimization for Collaborative Ranking on Recommendation Systems," in *Proceedings of the 2019 2nd International Conference on Machine Learning and Machine Intelligence*, 2019, pp. 7–11.

Submitted: **01/04/2023**; Accepted: **14/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1041-1049
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6049

[12] M. C. Keshava, S. Srinivasulu, P. N. Reddy, and B. D. Naik, "Machine learning model for movie recommendation system," *Int. J. Eng. Res. Tech*, vol. 9, pp. 800–801, 2020.
[13] X. Li *et al.*, "Application of intelligent recommendation techniques for consumers' food choices in restaurants," *Front Psychiatry*, vol. 9, p. 415, 2018.
[14] R. Ahuja, A. Solanki, and A. Nayyar, "Movie recommender system using k-means clustering and k-nearest neighbor," in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, 2019, pp. 263–268.
[15] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, "Deep item-based collaborative filtering for top-n recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 3, pp. 1–25, 2019.
[16] R. Anand and J. Beel, "Auto-surprise: An automated recommender-system (autorecsys) library with tree of parzens estimator (tpe) optimization," in *Proceedings of the 14th ACM Conference on Recommender Systems*, 2020, pp. 585–587.
[17] D. T. Tran and J.-H. Huh, "New machine learning model based on the time factor for e-commerce recommendation systems," *J Supercomput*, pp. 1–46, 2022.
[18] N. Mustafa, A. O. Ibrahim, A. Ahmed, and A. Abdullah, "Collaborative filtering: Techniques and applications," in *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, IEEE, 2017, pp. 1–6.
[19] N. Rajabpour, A. Naserasadi, and M. Estilayee, "TFR: a tourist food recommender system based on collaborative filtering," *Int J Comput Appl*, vol. 975, p. 8887, 2018.
[20] C. Yu, Q. Tang, Z. Liu, B. Dong, and Z. Wei, "A recommender system for ordering platform based on an improved collaborative filtering algorithm," in *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, IEEE, 2018, pp. 298–302.
[21] D. Massimo, M. Elahi, M. Ge, and F. Ricci, "Item contents good, user tags better: Empirical evaluation of a food recommender system," in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, 2017, pp. 373–374.
[22] N. Hug, "Surprise: A Python library for recommender systems," *J Open Source Softw*, vol. 5, no. 52, p. 2174, 2020.
[23] M. Jallouli, S. Lajmi, and I. Amous, "When contextual information meets recommender systems: extended SVD++ models," *International Journal of Computers and Applications*, vol. 44, no. 4, pp. 349–356, 2022.
[24] X. Guan, C.-T. Li, and Y. Guan, "Matrix factorization with rating completion: An enhanced SVD model for collaborative filtering recommender systems," *IEEE access*, vol. 5, pp. 27668–27678, 2017.
[25] M. Ge, M. Elahi, I. Fernaández-Tobías, F. Ricci, and D. Massimo, "Using tags and latent factors in a food recommender system," in *Proceedings of the 5th International Conference on Digital Health 2015*, 2015, pp. 105–112.
[26] J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C.-H. Lee, "On mean absolute error for deep neural network based vector-to-vector regression," *IEEE Signal Process Lett*, vol. 27, pp. 1485–1489, 2020.
[27] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geosci Model Dev*, vol. 15, no. 14, pp. 5481–5487, 2022.
[28] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

Submitted: **01/04/2023**; Accepted: **14/07/2023**; Published: **23/07/2023**