OPEN ACCESS
Indo-JC Indonesia Journal of Computing
ISSN 2460-9056
socj.telkomuniversity.ac.id/indojc

Ind. Journal on Computing
Vol. 7, Issue. 1, April 2022. pp. 1-14
doi:10.34818/indojc.2022.7.1.599

Non-Negative Matrix Factorization Based Recommender System using Female Daily Implicit Feedback

Hani Nurrahmi[#1], Agung Toto Wibowo[#2], Selly Meliana[#3]

#School of Computing, Telkom University
Jl. Telekomunikasi No. 1, Bandung 40257, Indonesia

[1] haninurrahmii@telkomuniversity.ac.id
[2] agungtoto@telkomuniversity.ac.id
[3] sellym@telkomuniversity.ac.id

Abstract
Recommender Systems is widely used by e-commerce and social media, to provide recommendations of items/products that are probably to be the interest to users. The recommended items are usually selected from thousands or even millions of other items. One of the recommender system algorithms that can be implemented is Non-negative Matrix Factorization (NMF) which receives explicit feedback in the form of user ratings. Although this method is effective, however, there are problems faced by explicit feedback as input. For example, there are users who act as grey-sheep or black-sheep by providing dishonest ratings as explicit feedback. On the opposite, dishonest feedback least frequently occurs in implicit feedback. Therefore, in this study, we used implicit feedback to recommend products by taking the implicit feedback obtained from the Discovery menu in Female Daily's mobile application as a case study. There are three types of implicit feedback in the Discovery menu: a. View Product Detail, b. View Review Detail, and c. Add to Wishlist. We then experimented with the NMF algorithm provided by the Surprise library using two implicit ratings weighting scenarios: a. accumulative weighing and b. maximum weighting. We combined several NMF parameters (such as K-Latent Factors and bias value) and run our experiment in 5-fold cross-validation. The best performance result in accumulative weighting is MSE = 1.2698, NMSE = 0.2116, RMSE = 1.1268, MAE = 0.7829. Meanwhile, the best performance result in maximum weighting is MSE = 0.6609, NMSE = 0.2203, RMSE = 0.8129, MAE = 0.5873.

Keywords: recommender systems, implicit feedback, Non-negative Matrix Factorization, Female Daily

Abstrak
Recommender Systems atau sistem pemberi rekomendasi banyak digunakan oleh *e-commerce* dan media sosial, untuk memberikan rekomendasi item/produk yang menarik bagi pengguna. Item yang direkomendasikan biasanya dipilih dari ribuan bahkan jutaan item lainnya. Salah satu algoritma sistem rekomendasi yang dapat diimplementasikan adalah *Non-negative Matrix Factorization* (NMF) yang menerima umpan balik eksplisit berupa penilaian pengguna. Meskipun metode ini efektif, namun ada masalah yang dihadapi oleh umpan balik eksplisit sebagai input. Misalnya, ada pengguna yang bertindak sebagai *grey-sheep* atau *black-sheep* dengan memberikan penilaian tidak jujur sebagai umpan balik eksplisit. Sebaliknya, umpan balik yang tidak jujur paling jarang terjadi dalam interaksi implisit. Oleh karena itu, dalam penelitian ini, kami menggunakan umpan balik implisit untuk merekomendasikan produk dengan mengambil interaksi implisit yang diperoleh dari menu "Discovery"" di aplikasi seluler "Female Daily" sebagai studi kasus. Ada tiga jenis umpan balik implisit dalam menu Discovery: a. *View Product Detail*, b. *View Review Detail*, dan c. *Add to Wishlist*. Kami kemudian bereksperimen dengan menggunakan algoritma NMF yang disediakan oleh library "Surprise" menggunakan dua skenario pembobotan peringkat implisit, yaitu: a. pembobotan akumulatif dan b. pembobotan maksimal. Kami menggabungkan beberapa parameter

Received on Januari, 2022. Accepted on March, 2022

NMF (seperti *K-Latent Factors* dan nilai bias) dan menjalankan eksperimen kami dalam 5-fold cross-validation. Hasil performansi terbaik pada pembobotan akumulatif adalah MSE = 1.2698, NMSE = 0.2116, RMSE = 1.1268, MAE = 0.7829. Sedangkan hasil performansi terbaik pada pembobotan maksimal adalah MSE = 0.6609, NMSE = 0.2203, RMSE = 0.8129, MAE = 0.5873.

Kata Kunci: recommender systems, implicit feedback, NM

I. INTRODUCTION

RECOMMENDER System is a technique that is able to select items that are most likely to attract the interest of a user [6], [15]. One of the motivations to utilize a recommendation system is a user is often faced with a large selection of items. In the real world, the items can reach thousands even millions [18]. It is difficult for a user to views and evaluate items one by one to determine which item is suitable for him or herself.

In recent years, many e-commerce and social media have been using recommender systems in their application. For example, Amazon[1] uses a recommender system to recommend products that are most likely to attract (will buy) the interest of their users. Netflix[2] and YouTube[3] recommend videos based on users' watching history.

The recommended products are not limited to videos or products being sold, but there are also items that users need to explore their review such as beauty products that were brought by Female Daily[4]. Female Daily launch a mobile application namely Discovery, where people communicate, discuss and share their opinion about beauty products. Recommender systems were needed to recommend more than forty-eight thousand beauty products to users.

Female Daily is Indonesia's largest beauty destination to discover, share, and buy beauty products. It begins with a personal blog that shares about beauty products, now Female Daily has become a large community for the beauty products enthusiast in Indonesia. Female Daily has two platforms: website and mobile application. The website platform focuses on beauty products review, articles, and a web page for buy beauty products. Whereas in mobile application platforms it has recommender systems that can recommend beauty products based on user demographic data e.g. skin type, skin tone, hair type, etc.

There is “Discovery” menu in the Female Daily mobile application. In this menu, the users can see posts in the form of photos that had been posted by other users. Users can give comments to each other posts, click the like button on a post, and check the product that had been tagged by the post owner. This menu shows the posts randomly. It does not consider the user's preferences or feedback; thus the problem is the users cannot see the posts from a user who has the same preferences or concerns. Based on the problem, a recommendation system is needed that can display posts based on users' preferences.

Generally, in providing recommendations, the recommender system will process some of the input that has been provided by the user. There are two types of input, i.e., explicit feedback and implicit feedback [2]. Explicit feedback is generally obtained by asking the users how well a particular item is directly (e.g. rating system). Implicit feedback is generally provided by users unknowingly, through interaction with an item (e.g. click or buy an item). The similarity of the two interactions are a) both of them show the preferences of users toward a particular item, b) both of them are converted to a sparse matrix notation since there is no item that obtains feedback from all users and there is no user that can provide feedback to all items.

One of the recommender system techniques that can be used for recommending items is Matrix Factorization [21]. In Matrix Factorization, a matrix input will be converted to two other matrices which is the result of the multiplied of these two matrices will be approached the matrix input's value in dense form. One of the algorithms that can be used for matrix input conversion is Non-Negative Matrix Factorization (NMF) [9]. Generally, the matrix factorization technique is used for rating prediction towards explicit feedback. However, in this research, the recommender system was developed using NMF to recommend products towards users' implicit feedback in the Female Daily mobile application. There are three implicit feedbacks that have been used in this research: a. View Product Detail, b. View Review Detail, and c. Add to Wishlist.

We assumed that the use of implicit feedback can give a suitable recommendation since those feedbacks represents the honest preferences of users. Besides, we analyze this NMF algorithm towards implicit feedback to discover a new way in recommender systems.

II. LITERATURE REVIEW

A. Recommender System

Recommender System is a technique that is able to select items that are most likely to attract the interest of a user [6], [15]. In general, the recommender system is beneficial in a condition where the user needs to select one or two items among thousands even millions of other items. Without recommender systems, the user needs extra time and effort to evaluate and choose the most suitable items for herself [6]. Nowadays, recommender system can be found in e-commerce (e.g. Amazon.com, the recommender system works to recommend products), movie streaming services (e.g. Netflix, the recommender system works to recommend movies), video sharing websites (e.g. YouTube, the recommender system works to recommend videos that have been shared by their users), or even in digital music streaming service (e.g. Spotify) [17].

To recommend items, recommender systems need to handle a specific task. There are several tasks that can be processed by recommender systems [6], i.e.:

1). Rating Prediction based on user preferences.
To obtain rating prediction, recommenders systems take user preferences in a form of user-item-interactions (such as rating) as input. These inputs are then estimated by recommenders systems using a specific algorithm such as Matrix Factorization [21]. To recommend items, recommender systems are then selected n numbers of items that obtain the highest rating as suggestions. The performance of this task is measured using specific metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

2). Discovering the suitable items for a user.
The recommender system's objective of this task is to select items that have good preferences for a user thus the user is interested to play, buying, or clicking the items. This task is necessary since in many systems it is impossible to show or recommend all good items provided simultaneously. The recommendation system has to provide it in the form of a ranked list of items [6]. The performance of this task is measured using specific metrics such as Normalized Discounted Cumulative Gain (NDCG), recall@N, and precission@N.

3). Recommending sequence of items.
This task can be processed when there is a relation between the items in the order of consumption. E.g. sequential movie or sequential music that is formed as a playlist.

B. Explicit Feedback and Implicit Feedback

In general, recommender systems obtain the input in many forms, i.e. explicit feedback and implicit feedback. Explicit feedback is the input provided by the user towards the item consciously. Some researchers had used explicit feedback in their research [3], [5], [8], [14], [19]. Users are asked to evaluate how well the products are in the form of stars rating, answering a questionnaire, product reviews, like-dislike rating, or comments. The explicit feedback data can be converted to an interaction matrix. Since many users never give ratings or interact with items, the data will be converted to a sparse matrix.

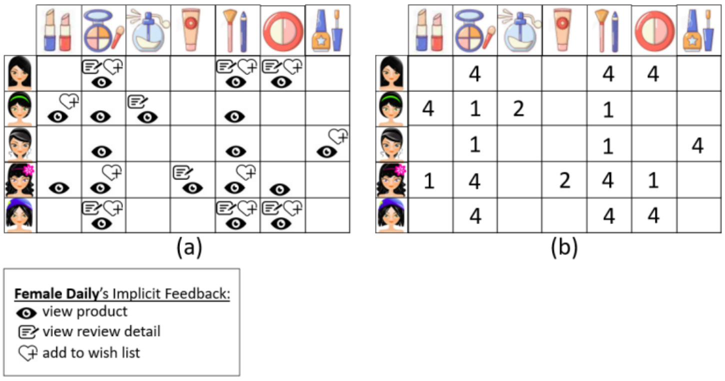

Fig. 1. The Female Daily Implicit Feedback Illustrations. (a) The Implicit Interactions occurs in Female Daily's Discovery. (b) The Converted Interactions With The Maximum Interaction Scenario (Described in Section Data Preprocessing)

Recommender systems collect the implicit feedback during users' interaction into items unconsciously. For instance, recommender systems can log the e-commerce user's actions such as “clicks a thumbnail” of a product to see the details of the product. Other interactions such as play, pause, skip provided by a user when they watch a movie are also considered as implicit interactions. These interactions are saved by the system as implicit feedback. The advantage of this feedback is the user does not need to identify his preferences intentionally through answering questionnaires, giving comments, reviewing products, or giving ratings.

In the Female Daily mobile application, there are different types of implicit feedback, e.g. "View Product", "View Review Detail", and "Add to Wishlist" (see Figure 1). When users enter the Female Daily application, they obtain a list of beauty products (e.g. lipstick, whitening powder). In response, users can interact with those beauty products in different ways. A user can view a product by opening the image provided by the female daily. The user can also view the review detail on a product, and then add the product to their Wishlist. These implicit interactions are logged separately by Female Daily.

There are differences between explicit and implicit feedback:

a. To obtain explicit feedback, users intentionally identify their preference for an item. On the opposite, to obtain implicit feedback the users unknowingly identify their preference through interaction towards an item, e.g., view product, add to Wishlist, buy a product, listen to a song, watch the video later, etc.
b. The workload. To obtain the explicit feedback, the user is given the additional workload since the user has to fill out a preference consciously and apart from the activity of its interaction with the item (e.g., select the star rating, reviewing a product, comment, etc.).
c. The integrity. to obtain explicit feedback, sometimes the information that is given by the user can be wrong. For example, a user may incorrectly give a star rating because one of them does not understand the meaning or sometimes the user also does not know the reason why they like certain items. However, the implicit feedback is relatively more reliable because it is taken based on user behavior and follows a certain pattern obtained by comparing the behavior of other users.

C. Recommender System Techniques

There are several studies that have been published to discover recommender system techniques, i.e., studies in [3], [6], [9], [10], [11], used Collaborative Filtering technique, and studies in [1], [13], [20] used Content-Based Filtering technique. Each technique has a different type of input and the process of suitable items selection for users. These are the brief explanation of each recommender system technique.

1) Collaborative Filtering: The Collaborative Filtering [3], [6], [9], [10], [11] receives input in the form of the matrix of user interactions with items. In many literature studies, the interaction that had been used is the explicit feedback in the form of a rating. In general, collaborative filtering can process the rating prediction task. To recommend an item, collaborative filtering will recommend items based on items that other users with similar tastes liked in the past. One of the implemented algorithms in collaborative filtering is matrix factorization [9], [11].

2) Content-Based Filtering: The Content-Based Filtering [1], [13], [20] receives input in the form of item description from all items and the item description from items that have a positive preference for a user. For instance, the item description on the movie domain is a list of actors, directors, genre, or movie's synopsis for a movie. These item descriptions will be the input of the recommender system in the movie domain. In Content-Based Filtering, if a user is known to have a positive preference for an item, other items with similar attributes will be recommended by the system to her.

3) Demographic Filtering: Demographic Filtering [4], [16] technique groups users based on the similarity of their attributes. For instance, the recommendation system will collect data in the form of age, domicile, gender, and some other personal information. Items will be recommended to a user if it is known that other users with the same demographic have a positive preference for the item. While the hybrid approach [20] combines several approaches in selecting items that are estimated to be the preferences of a user.

D. Matrix Factorization

The matrix factorization is used by the recommendation system in completing the task rating predictions. As a part of the collaborative filtering method, matrix factorization can provide good accuracy. Matrix factorization accepts input in the form of an interaction matrix (R) which describes the preference values (e.g., feedback as seen in Figure 1) from a user to an item. One of the advantages of matrix factorization is its ability to estimate all interactions from all users on all items [21]. It should be noted that in matrix factorization, the R matrix is a sparse matrix, where many cell values are unknown.

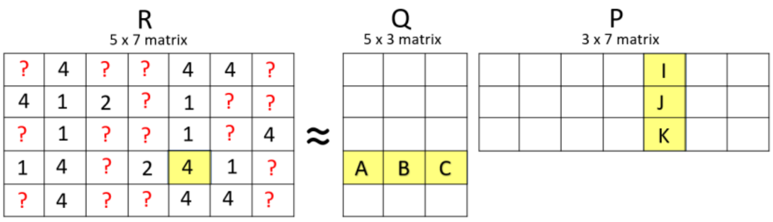

Fig. 2. The Illustration of Matrix Factorization

Formally, as seen in Figure 2, the factorization of the matrix can be defined as given a matrix containing user interactions with items R, then look for two other matrices, i.e. Q and P; where Q and P describe the user and item latent factors with the result of the dot product operation of Q and P approaching the original matrix (R) [21]. The values in the R matrix represent the interaction value/rating (r_{ui}) of a user u on an item i. The rating prediction ($\hat{r}_{ui}$) obtained by equation (1) is the result of multiplying the value of the latent vector q_i^T item on the item i against user u's latent vector p_u.

$$\hat{r}_{ui} = q_i^T p_u \quad (1)$$

E. Non-Negative Matrix Factorization

Linear Dimensionality Reduction (LDR) is the basic technique to perform data analysis. The LDR technique is used to analyze large-dimensional data and has been widely implemented in the fields of statistics and machine learning [9]. LDR is built on three matrices, i.e.:

1). Matrix $V \in R^{p*n}$
Each column represents data points where $V(:,j) = v_j$ for $1 \leq j \leq n$.

2). Matrix $W \in R^{p \times r}$
Each column represents a base element was $W(:,k) = w_k$ for $1 \leq k \leq r$.

3). Matrix $H \in R^{r \times n}$
Each column represents the coordinates of the data point $V(:,j)$ on the W base, i.e. $H(:,j) = h_j$ for $1 \leq j \leq n$.

Non-Negative Matrix Factorization (NMF) is an LDR that disjoint a high-dimensional matrix R with $p \times n$ dimensions into two low-dimensional matrices, i.e. Q and P (as seen in Figure 2), that have dimensions $p \times r$ and $r \times n$, respectively, where $v_{ij} \geq 0$, $w_{ij} \geq 0$, $h_{ij} \geq 0$, and $r < (p, n)$.

The recommender system is one of the systems that apply NMF in the prediction process. Typical input from a recommendation system is user-item interaction, i.e. data on user preferences for certain items. Data is stored in a matrix where each row represents a user and each column represents an item. The obstacle that is often faced when using this technique is the problem of sparsity. The preference matrix will have a lot of sparse data because not all users rate every item.

The application of NMF is said to be able to minimize these problems. Implementation using NMF also has low computational complexity [12]. The paper also mentions that implementation using NMF is relatively easy and has very low computational complexity.

The application of NMF allows a recommendation system to predict what items are liked by a particular user, even though the user has never been exposed (read, watched, etc.) by the item in question. NMF is used to predict preference matrix cells that are mostly empty due to the sparsity problem by parsing the main matrix, R, into a Q matrix containing user latent factors and a P matrix containing item latent factors as seen in Fig. 2.

One of the libraries that can be used to implement NMF is Surprise [7]. Surprise is a Python Scikit package for recommender systems. Surprise provides implementations of various matrix factorization models complete with matrix factorization algorithms and associated initialization methods. Some of the modules that can be used i.e., SVD (surprise prediction algorithms matrix factorization SVD), SVD++ (surprise prediction algorithms matrix factorization SVDpp) and NMF algorithms [7] that contains a collaborative filtering algorithm based on NMF (surprise prediction algorithms matrix factorization NMF).

III. RESEARCH METHOD

In this research, several systematic steps are planned to utilize implicit feedback data into a recommendation system input. As explained in the introduction, implicit feedback has the advantage that user interaction with items is an honest preference. However, implicit feedback also has a drawback, namely the absence of negative interactions which indicate that a user does not like certain items.

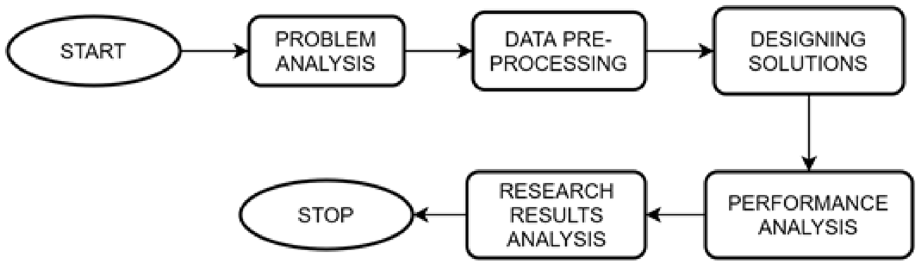

Fig. 3. Research Method Design

Considering the existing strengths and weaknesses, then the research will be carried out in stages according to Figure 3 that can be described as follows:

1). Problem analysis. At this stage, the author analyzes the implicit feedback data from Female Daily to obtain the characteristics of the data and determine the weighting process based on user interactions. In addition, the author also designed a solution based on a problem analysis and literature study that focuses on the field of recommendation systems, especially those that utilize Non-Negative Matrix Factorization and implicit feedback data processing.
2). Data pre-processing. The implicit feedback data from the Discovery menu will be processed in terms of weighting scenarios that can distinguish the level of preference of a user to the products offered. The difference of the level of users' preference is expected to describe the users' usual conditions, likes, and really likes the product. From this stage, it is hoped that several data pre-processing scenarios will optimize the system's performance. As seen in Figure 4, there are two steps that will be processed in data pre-processing: 1) giving weight to implicit feedback with accumulative interaction weighting scenario, and 2) giving weight to implicit feedback with maximum interaction weighting scenario. The output of this stage is the data that have been weighted.
3). Solution design. At this stage, the most appropriate feature is determined in the process of forming a rating classification model based on a review. The author used the NMF algorithm as the rating prediction for weighted implicit feedback data, as seen in Figure 4.
4). Performance testing and analysis. At this stage, test scenarios and evaluation of test results from several proposed models are carried out previously. As seen in Figure 4, the author used MSE, RMSE, and MAE as the performance evaluation.
5). Analyze the results and draw conclusions from the results of experiments carried out in the previous stage.

IV. RESULTS AND DISCUSSION

In this research, we carried out systematic steps to utilize Female Daily's Discovery implicit feedback data as an input. We then bring the recommendation system over this data in a form of rating predictions. We followed the following stages: a. Problem Analysis, b. Data Pre-processing, and c. Solution design using Non-Negative Matrix Factorization (NMF). Each stage can be described as follows.

A. Problem Analysis

In this stage, we identified problems that might arise during the research process. One of them is what types of implicit interactions that available in Female Daily's Discovery, and what types of implicit interactions can be used to build implicit interaction matrices. This implicit matrix should be related to each other and shows a correlation as in the case of the rating in the recommendation system.

In this research, we did not use explicit feedback because a) the research that has been using explicit feedback (rating) is often discussed by other researchers (see in Literature Review), b) Female Daily did not collect explicit feedback in the Discovery Menu.

From the data provided by Female Daily, there are several interaction Tables that have positive action. This action indicates that a user performs a certain action that corresponds to another user, a brand, a post item, and also a product. These existing data include

1. User-to-user interactions are a form of interactions of a user to another user, e.g. user A likes a post by user B or user A views user B profiles.
2. User-to-item interaction is a form of interaction made by a user with other items. In Female Daily's Discovery data, the items can be a form of products, brands, or posts. There are several user interactions with the product, i.e.: “view review list”, “view product details”, “view review details”, and “add to Wishlist”. While the interaction of the post can be in the form of a tap on feed items and like posts. The interaction with the brand can be in the form of a product list view.

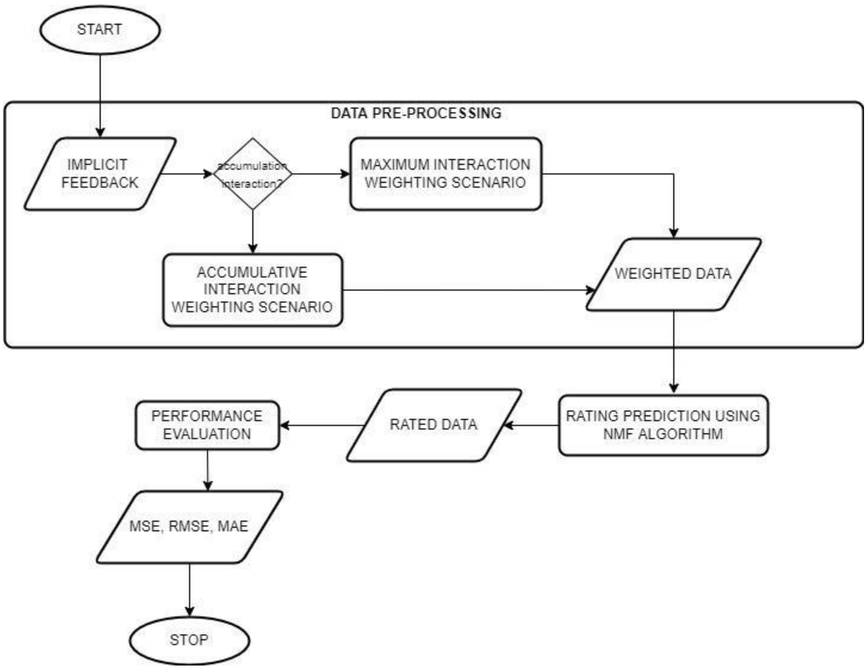

Fig. 4. Problem Solutions Flowchart

From the several types of interactions provided by Female Daily, we chose the interactions that can be converted into “rating like" interactions. We then selected the interactions which related to products rather than another type of user-to item interactions. The users to product interaction have similar characteristics with rating systems which shows strong and low preferences of the user to the product.

In Female Daily's Discovery data, the users can interact with products in different ways: a. “add to Wishlist”, b. “view review detail”, c. “view product detail”, and d. “view review list”. Using these types of interactions, we then create an order of preferences signals. This kind of order simulates the rating preference in recommender systems. Due to the insufficient number of data provided by Female Daily over the “view review list”, we then discard those data and use only three other types of interactions. It is argued that the “add to Wishlist” becomes the highest interaction compared to other interactions since it shows the strongest preferences compared to other interactions. In some cases, a user can interact multiple times with a product in different ways. For example, a user U_i can interact with a product P_1 in a form of “add to wishlist”, “view review detail”, and “view product detail” simultaneously. It is also possible for a user U_i to only interact to another product P_2 in two one of them, such as only interact in a form of “view product detail”.

From the “add to Wishlist”, “view review detail”, and “view product detail” interactions, we converted these data into a pseudo-implicit rating by assigning a weight to the interactions. In our experiment, we replaced each positive interaction from a user to a product in a form of ``view product detail'' interaction into “1”, “view review detail” interaction into “2”, and “add to Wishlist” interaction into “4”. We used a weight of 4 (instead of 3) on the “add to Wishlist” interaction because we need to avoid the inconsistency of the accumulated data of a user who has an interaction of product detail view and detail review view on the same product (implicit value 3). The final implicit feedback data is the sum of the weights of a user's interaction with a product.

B. Data Pre-processing

All implicit feedback we obtained from Female Daily is positive feedback. This feedback represented the users' preferences for specific products. These data are relatively different with explicit feedback such as rating. In explicit feedback such as rating between 1 to 5, there are negative preferences represented with rating 1 or 2. The positive preferences in such explicit feedback can be represented with a rating of 4 or 5.

Meanwhile, all interactions obtained from Female Daily are considered positive interactions. Take the “add to Wishlist” interaction as an example, all recorded interactions indicate that the users provide a positive signal to add the appropriate products to their Wishlist. On the opposite, Female Daily has no ability to record the negative signal since there is no clue whether a user really does not want that product, or just has no intention to add the product to their Wishlist. Female Daily will record each time the positive interactions have occurred, and if the user does not interact with the addition to the Wishlist on a product, then the female daily does not record the interaction. Each interaction type is recorded in a separate log file and needs to be combined into a Table to facilitate the process of predicting its implicit feedback rating.

	username	productid	viewProduct	viewReviewDetail	addToWishList	implicite	userid
0	Arinkaghita01	44609	1	0	0	1	5023
1	ms_auror	50250	1	0	0	1	147000
2	ningtyasss	208	1	0	0	1	157569
3	hafsahriutami	25164	1	0	0	1	111931
4	intanpuspitasari14	42473	1	1	0	3	121156
5	nilamkhrnsyh	527	1	1	0	3	156955
6	tashyasaskirana	26513	1	0	0	1	199114
7	hafsahriutami	14857	1	0	0	1	111931
8	nandasabila10_	6230	1	0	0	1	152816
9	adrinebianca	10665	1	1	0	3	48623

Fig. 5. The data sample from preprocessed data interactions

We concatenated all implicit feedback types and converted the recorded implicit interactions into a Table, as illustrated in Figure 5. Figure 5 also shows the sample data from interactions that have been preprocessed. As seen in the image, there are several fields. The username, productid, and *userid* fields are the identities of the user and product corresponding to the implicit feedback. The “viewProduct”, “viewReviewDetail”, and “addToWishList” fields are the history of implicit interactions of a user with the product. The values in “viewProduct”, “viewReviewDetail”, and “addToWishList” fields become a boolean value with a value of 1 indicating an interaction has occurred and 0 indicating no interaction. The values in the “implicit” column represent the values obtained from a calculation of weighting scenarios. We use two weighting scenarios namely: a. accumulative interaction and b. maximum interaction.

In the accumulative interaction scenario, the implicit feedback value will use the sum of the weights from the appearance of 3 types of interactions given by a user to the product. With this mechanism, the interaction value will range from 1 to 7. A value of 7 is obtained when a user provides 3 interactions over the same product at once. This value is obtained as a summation of 1 on the “view product” interaction, 2 on the “view review detail” interaction, and 4 on the “add to Wishlist” interaction.

In the maximum interaction scenario, the interaction value will be cut to 4. This means that if a user performs 3 interactions over the same product at once the value will be set to 4.

This value is obtained as a maximum value of 1 on the “view product” interaction, 2 on the “view review detail” interaction, and 4 on the “add to Wishlist” interaction. This value will be the same as the value obtained when a user only interacts with the add to Wishlist.

From the results of preprocessing data, an implicit rating dataset is obtained that can be described as follows:

1. The implicit rating consists of 6,113,259 lines of implicit interaction given by 219,205 users to 48,048 products registered in the Discovery Female Daily application.
2. From the 6,113,259 implicit feedbacks, there are 3,892,847 view product detail interactions, 2,414,678 view review details and 742,425 add to wish lists given by a user to the products listed.
3. From point 2 above, two datasets are prepared that describe the accumulative interaction scenario and maximum interaction scenario. The accumulative interaction scenario will have an implicit value range ranging from 1 to 7. While the maximum interaction scenario has an implicit value range ranging from 1 to 4. The distribution of each value in the two scenarios can be seen in Table 1.

TABLE I
THE FREQUENCY OF IMPLICIT FEEDBACK IN THE DATASET WITH THE ACCUMULATIVE INTERACTION SCENARIO AND THE MAXIMUM INTERACTION SCENARIO

Implicit Value	**Scenarios**	
	Accumulative Interaction	**Maximum Interaction**
1	3,094,241	3,094,241
2	1,705,101	1,707,101
3	571,492	571,492
4	433,745	742,425
5	170,595	
6	81,566	
7	56,519	

C. Solution Design

The preprocessed dataset in Table 4 is then converted into a sparse matrix of implicit interactions consisting of 3 columns namely:" userid", "productid", and "implicit" consisted 6,113,259 rows. Normally, the recommender systems sparse matrix using rating as a targeted class. In this research, we used implicit values as "rating" like data. The sparse matrix then becomes the input of the NMF algorithm. The NMF will produce user and product latent feature matrices before we can use them to predict the implicit values.

In this research, we used the NMF algorithm provided by the Surprise library (http://surpriselib.com/). Surprise library has been equipped with k-cross-fold validation evaluation, error calculation with metric mean square error (MSE), root mean square error (RMSE), and also mean absolute error (MAE). Therefore, we carried out experiments over two implicit weighting mechanisms: a. accumulative interaction and b. maximum interaction mentioned in section Data Pre-Processing.

D. Experiment Scenario

In this research process, we conducted several evaluations to find the best parameters of the Non-Negative Matrix Factorization (NMF) algorithm, indicated with the smallest error value for implicit feedback predictions. The dataset used is Female Daily interaction data which has been described in section Data Pre-processing. We conduct our experiments as follows:

1. In accordance with the previous description, we constructed the dataset in two scenarios, i.e.: a) accumulative interaction scenario and b) maximum interaction scenario. In the accumulative scenario, the rating values range from 1 to 4. Meanwhile, in the maximum scenario, the rating values range from 1 to 7.
2. We calculated the error metric value in the form of Mean Square Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE) in all scenarios. In the context of rating prediction, an accurate model is needed so that it can predict positive preferences (high ratings) and negative preferences (low ratings). Error-based measurements (MAE, RMSE, MSE) provide an illustration of how much error is in the preference prediction. In order to compare the accumulative and maximum scenarios, we also calculate the Normalize Mean Square Error (NMSE). The NMSE is calculated by dividing the MSE by the possible maximum rating minus the possible minimum rating in each scenario.
3. For each parameter's combination, we run 20 times 5-fold-cross-validations and reported the performance metric explained in point 2. We provide the average and best performance metric as shown in Table II.

4. The NMF parameter that will be tested is the k value which represents the dimensions of the latent feature of NMF. For example, if k is taken as 5, then the result of matrix factorization with the NMF algorithm will produce a user's latent feature matrix with dimensions of 219,205 x 5, and the product's latent feature matrix of 5 x 48,048. A value of 219,205 represents the number of users in the dataset, a value of 5 represents the value of latent features, and a value of 48,048 represents the number of products in the dataset. We experimented the value of k = {5, 10, 15, and 20} .
5. In addition to the NMF parameters, in the Surprise library, there is a biased parameter. The biased parameter in the NMF algorithm in the Surprise library is a Boolean variable that states whether the algorithm will use the baselines value or not [cite surprise]. In this research, we conducted both biased values (biased = {true, false}) and reported the obtained performances.

The five scenarios will be implemented by the researcher as a research evaluation process and will be reported in the next sub-chapter.

E. Experiment Results and Analysis

Based on the scenario as mentioned in the section Experiment Scenario, we conducted our experiment over the Female Daily implicit feedback dataset. The experiment results can be seen in Table II.

Table II shows the average and the best performance measurement results, measured by using the mean square error (MSE), normalized mean square error (NMSE), root mean square error (RMSE), and mean absolute error (MAE) with 20 times 5-fold-cross-validations setting. The first column in Table II shows the implicit rating weighting scenario. The second column states the utilization of the NMF algorithm's bias value in the Surprise library. While the third column states the value of k-latent as one of the NMF parameters. In our experiment, we used the value of k = {5, 10, 15, 20}. The fourth, fifth, and sixth columns, respectively, are the average values of the MSE, NMSE, RMSE, and MAE of the data testing performed in the 20 times 5-fold-cross-validations scenario. Meanwhile, the seventh, eighth, and ninth columns respectively state the best values of MSE, NMSE, RMSE, and MAE.

To get a better understanding of our results, we divided our analysis in Table II into two parts, following the existing implicit feedback weighting scenario. We considered the range of values in the accumulative interaction scenario, and this is different from the maximum interaction scenario. The implicit feedback value in the accumulative interaction scenario ranged from 1 to 7, while the implicit feedback value in the maximum interaction scenario ranged from 1 to 4. This difference will affect the error values expressed by MSE, NMSE, RMSE, and MAE.

TABLE II
THE AVERAGE DATA AND THE BEST PERFORMANCE MEASUREMENT RESULTS

Parameters			Average				Best			
Weighting Scenario	Biased	k	MSE	NMSE	RMSE	MAE	MSE	NMSE	RMSE	MAE
Accumulative	FALSE	5	1.2958	0.2160	1.1383	0.7890	1.2914	0.2152	1.1364	0.7877
Accumulative	FALSE	10	1.3123	0.2187	1.1455	0.7971	1.3040	0.2173	1.1419	0.7951
Accumulative	FALSE	15	1.3142	0.2190	1.1464	0.7990	1.3088	0.2181	1.1440	0.7975
Accumulative	FALSE	20	1.3116	0.2186	1.1453	0.7989	1.3073	0.2179	1.1434	0.7974
Accumulative	TRUE	5	1.2808	0.2135	1.1316	0.7861	1.2698	0.2116	1.1268	0.7829
Accumulative	TRUE	10	2.3075	0.3846	1.4691	1.0063	1.5212	0.2535	1.2334	0.8525

Accumulative	TRUE	15	2.3815	0.3969	1.5060	1.0283	1.6407	0.2734	1.2809	0.8764
Accumulative	TRUE	20	2.3768	0.3961	1.5071	1.0288	1.6425	0.2737	1.2816	0.8818
Maximum	FALSE	5	0.7155	0.2385	0.8458	0.6262	0.6904	0.2301	0.8309	0.6065
Maximum	FALSE	10	0.6956	0.2319	0.8340	0.6094	0.6844	0.2281	0.8273	0.5907
Maximum	FALSE	15	0.6862	0.2287	0.8284	0.5956	0.6819	0.2273	0.8258	0.5876
Maximum	FALSE	20	0.6818	0.2273	0.8257	0.5894	0.6794	0.2265	0.8243	0.5870
Maximum	TRUE	5	0.6637	0.2212	0.8147	0.5884	0.6609	0.2203	0.8129	0.5873
Maximum	TRUE	10	0.9983	0.3328	0.9874	0.7069	0.7750	0.2583	0.8804	0.6326
Maximum	TRUE	15	1.0855	0.3618	1.0278	0.7380	0.8151	0.2717	0.9028	0.6473
Maximum	TRUE	20	1.0688	0.3563	1.0229	0.7345	0.8402	0.2801	0.9166	0.6567

In Table II, cells with yellow, green, and blue colors represent the best performance values indicated by smaller MSE, NMSE, RMSE, and MAE values when compared to other rows in the evaluation group with the same weighting scenarios and biased values. Cells in blue are the best performance values from the accumulative interaction scenario, while cells in green are the best performance values from the maximum interaction scenario.

In general, with the presence of a performance value, it improves with a decrease in the value of k latent features. These occur in at least three groups, namely the accumulative interaction scenario either with or without using bias and the maximum interaction scenario which is carried out using bias. The three evaluation groups got the best performance value with the value of k latent features being 5. Only one evaluation group, namely the maximum interaction scenario which was carried out without using the bias, experienced a decrease in performance with a decrease in the value of k latent features. The best performance in this group is obtained when the value of the k latent feature is 20.

In both scenarios (accumulative and maximum), the use of bias values can improve performance. This is indicated by the best MSE, NMSE, RMSE, and MAE values or a smaller average when compared to the same k-latent feature value. The best average performance in the accumulative interaction scenario (scenario 1) is MSE = 1.2808, NMSE = 0.2135, RMSE = 1.1316, MAE = 0.7861, and the best performance for each fold in the accumulative interaction scenario is MSE = 1.2698, NMSE = 0.2116, RMSE = 1.1268, MAE = 0.7829 indicated by green cells. While the best average performance in the maximum interaction scenario (scenario 2) is MSE = 0.6637, NMSE = 0.2212, RMSE = 0.8147, MAE = 0.5884, and the best performance for each fold in the maximum interaction scenario is MSE = 0.6609, NMSE = 0.2203, RMSE = 0.8129, MAE = 0.5873. All the best performances in scenario 1 and scenario 2 are obtained by utilizing the bias value, and the value of k-latent feature = 5. Based on the NMSE values, the best weighting scenario is the Accumulative Weighing Scenario. The best NMSE value in average performance is 0.2135 and the best NMSE value in performance for each fold is 0.2116 indicated by green cells in Table II.

V. Conclusion

In the previous sections, we discussed our ideas to conduct experiments over Female Daily's implicit interactions dataset. We used NMF to predict its implicit ratings in two scenarios, namely accumulative interaction and maximum interaction. From the experiment results, we can draw conclusions:

1. The error performance is not related to the value of the k-latent feature. This is indicated by the inconsistent performance improvement when we increase or decrease the value of the k-latent feature. Some experiments improved the performance when we increase the value of the k-latent feature, while others improved the performance when we decrease the value of the k-latent feature,

2. The average value of the best performance in the accumulative interaction scenario (scenario 1) is MSE = 1.2698, NMSE = 0.2116, RMSE = 1.1268, MAE = 0.7829, obtained with biased = true parameter and k-latent feature value = 5, and
3. While the best average performance in the maximum interaction scenario (scenario 2) is MSE = 0.6609, NMSE = 0.2203, RMSE = 0.8129, MAE = 0.5873, obtained with biased = true parameter and k-latent feature value = 5.

In the future, we are keen to explore several experiments over the Female Daily implicit interactions dataset. There are several directions we can follow:

1. It is necessary to compare the results with several rating prediction algorithms such as singular value decomposition (SVD), singular value decomposition plus-plus (SVD++), Matrix Factorization (MF), as well as k-NN,
2. The experiment conducted in this paper is based on rating predictions. There is another recommendation setting that experimented in the top-N recommendations setting,
3. There is also a possibility to combine the NMF algorithm with other algorithms, and carried out as a monolithic or parallel hybrid algorithm.

ACKNOWLEDGMENT

The authors would like to thank Female Daily for their support in developing a recommendation system in their mobile application.

REFERENCES

[1] Anand, P.B. and Nath, R. (2020). Content-Based Recommender Systems. In Recommender System with Machine Learning and Artificial Intelligence (eds S.N. Mohanty, J.M. Chatterjee, S. Jain, A.A. Elngar and P. Gupta).

[2] Badreesh Shetty. An in-depth guide to how recommender systems work. https://builtin.com/data-science/recommender-systems, 2019.

[3] Bobadilla, J., Alonso, S., & Hernando, A. (2020). Deep Learning Architecture for Collaborative Filtering Recommender Systems. Applied Sciences, 10(7), 2441.

[4] Bobadilla, J., González-Prieto, Á., Ortega, F. et al. Deep learning feature selection to unhide demographic recommender systems factors. Neural Comput & Applic **33,** 7291–7308 (2021).

[5] Emre Yalcin, Firat Ismailoglu, Alper Bilge, An entropy empowered hybridized aggregation technique for group recommender systems, Expert Systems with Applications, Volume 166, 2021, 114111, ISSN 0957-4174

[6] Francesco Ricci, Lior Rokach, and Bracha Shapira. Collaborative filtering recommender systems. Recommender Systems Handbook, pages 1–35, 2010

[7] George Pipis. How to run recommender systems in python. https://predictivehacks.com/how-to-run-recommender-systems-in-python/, 2020

[8] Hassanieh, L. A., Jaoudeh, C. A., Abdo, J. B., & Demerjian, J. (2018). Similarity measures for collaborative filtering recommender systems. 2018 IEEE Middle East and North Africa Communications Conference (MENACOMM).

[9] Hernando, A., Bobadilla, J., & Ortega, F. (2016). A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. Knowledge-Based Systems, 97, 188–202.

[10] L. A. Hassanieh, C. A. Jaoudeh, J. B. Abdo and J. Demerjian, "Similarity measures for collaborative filtering recommender systems," 2018 IEEE Middle East and North Africa Communications Conference (MENACOMM), 2018, pp. 1-5

[11] Lara-Cabrera, R., González-Prieto, Á., & Ortega, F. (2020). Deep Matrix Factorization Approach for Collaborative Filtering Recommender Systems. Applied Sciences, 10(14), 4926.

[12] Mehdi Hosseinzadeh Aghdam, Morteza Analoui, and Peyman Kabiri. A novel non-negative matrix factorization method for recommender systems. Applied Mathematics & Information Sciences 9, pages 2721–2732, 2015

[13] Musto, C., Franza, T., Semeraro, G., de Gemmis, M., & Lops, P. (2018). Deep Content-based Recommender Systems Exploiting Recurrent Neural Networks and Linked Open Data. Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization - UMAP '18.

[14] Negin Ghasemi, Saeedeh Momtazi, Neural text similarity of user reviews for improving collaborative filtering recommender systems, Electronic Commerce Research and Applications, Volume 45, 2021, 101019, ISSN 1567-4223

[15] Portugal, I., Alencar, P., & Cowan, D. (2018). The use of machine learning algorithms in recommender systems: A systematic review. Expert Systems with Applications, 97, 205–227.

[16] Rashidul Islam, Kamrun Naher Keya, Shimei Pan, James Foulds. 2019. Mitigating Demographic Biases in Social Media-Based Recommender Systems. In KDD '19: Social Impact Track, August 04–08, 2019, Anchorage, Alaska. ACM, New York, NY, USA, 3 pages.

[17] Rishabh Mall. Recommender system, towards data science. https://towardsdatascience.com/recommender-system-a1e4595fc0f0, 2019

[18] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. ACM Comput. Surv. 52, 1, Article 5 (February 2019), 38 pages

[19] S Hasanzadeh, S M Fakhrahmad, M Taheri, Review-Based Recommender Systems: A Proposed Rating Prediction Scheme Using Word Embedding Representation of Reviews, The Computer Journal, Volume 65, Issue 2, February 2022, Pages 345–354

[20] S. Amara and R. R. Subramanian, "Collaborating personalized recommender system and content-based recommender system using TextCorpus," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 105-109

[21] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. Computer, 42:30–37, 2009