



Group Recommender System using Matrix Factorization Technique for Book Domain

Moh Naufal Mizan Saputro, Z. K. A. Baizal*

School of Computing, Informatics, Telkom University, Bandung, Indonesia

Email: ¹mizansaputro@student.telkomuniversity.ac.id, ^{2,*}baizal@telkomuniversity.ac.id

Correspondence Author Email: baizal@telkomuniversity.ac.id

Abstract—A recommender system helps users to select the desired items by analyzing the user's habit of interacting with the system. Recommender system also help the group of users for selecting items due to information overloads. Group Recommender System (GRS) is designed to identify all preferences within a group. An aggregation strategy is needed to accommodate all user preferences in a group. GRS is required in many cases, for example in the book domain, a bookstore recommends a list of books through a display for a group of visitors. We design a GRS for the book domain using Matrix Factorization technique. We utilize three methods to design GRS, such as After Factorization (AF), Before Factorization (BF), and Weighted Before Factorization (WBF). These three approaches were applied to three different group categories, i.e., small groups, medium groups, and large groups. We aim to find the best approach for each group category in this research. The evaluation metrics used are precision and recall in building this GRS. The results of this research indicate that a small group is suitable for using all three approaches, AF methods is the best approach methods for medium groups, and the best approach method for large groups is WBF.

Keywords: Group Recommender System; Matrix Factorization; After Factorization; Before Factorization; Weighted Before Factorization

1. INTRODUCTION

Data on the internet is huge and continues to increase along with the number of internet users [1]. However, not all data available on the internet can be used or provide satisfactory results for the user [1]. To overcome this problem, a recommender system has been developed to assist users for selecting required items, where the recommended items are retrieved from a database that contains a very large set of items [2]. A recommender system is a type of information filtering system that is used to predict the rating or preference a user will give to an item [3]. Recommender systems are applied to various products and services on the internet, for example movies, videos, music, and social media [2][4]. The recommender system provides personalized item recommendations by analyzing the user's habits of interacting with the system [5]. However, there are conditions where a recommender system is needed to recommend an item to a group of users who have different preferences, so that a recommender system for groups is also an issue that is equally important to address [6].

Many e-commerce platforms utilize recommender systems to optimize their operations, and one of the most commonly employed paradigms for constructing these systems is Collaborative Filtering (CF) [7][8]. CF recommendations are founded on the principle that a user is likely to appreciate items favored by another user who shares similar preferences. This approach leverages the (implicit or explicit) feedback provided by each individual user to suggest items that are popular among users with comparable tastes [9][10]. The widespread adoption of these implementations has propelled the field of recommender systems, resulting in the development of faster and more precise recommender systems [11]. Numerous conventional CF techniques rely on the foundation of Matrix Factorization (MF) [12].

Group Recommender System (GRS) is a model that integrates all preferences of group members [6][13]. To design the model, an aggregation strategy is used that fits the needs of the group [6]. GRS is more complicated than individual recommender systems [14]. Some examples of problems that require GRS include watching movies together, choosing tourist destinations together, choosing radio stations together, and choosing music together [8]. Many studies discuss the recommender system for the book domain. One of these studies utilizes the Convolutional Neural Network (CNN) algorithm in building a recommender system [15]. This algorithm is used because it can overcome the problem of data sparsity. GRS is also needed in the book domain, for example, a bookstore increases sales by displaying book recommendations on the LCD by looking at nearby customer preferences. Based on these problems, we created a GRS with a book domain. We designed the GRS using the Collaborative Filtering (CF) paradigm with the Matrix Factorization (MF) method.

Collaborative filtering plays a crucial role in enhancing the recommendation environment through the utilization of matrix factorization (MF) decomposition technology, which has been proven to be one of the most effective recommendation strategies. However, despite its success as a method employed in recommendation systems, SVD-based methods encounter the issue of data sparsity, resulting in inaccurate prediction of ratings. R.Barathy., et al. [16] built a recommendation system using the CF paradigm with the MF method optimized using the SVD method.

We conducted this research with the aim of building an accurate GRS to recommend an item using the MF paradigm CF method with three approaches, i.e., After Factorization (AF), Before Factorization (BF), Weighted Before Factorization (WBF) to be applied in various group categories. In this study, we specifically focus on the



book domain. The GRS that we build uses three approaches that are applied to three different group categories with the aim of knowing the best approach for each group category. We build GRS with this model to find out the advantages and disadvantages of each approach used in the system.

We organize this research as follows: Section 2 is a research methodology that describes the steps and methods we use in developing the GRS for the book domain. Section 3 is a description of the results and discussion of the research we conducted. Section 4 is a conclusion and feature work.

2. RESEARCH METHODOLOGY

2.1 Research Stages

The system design modeling flow in building GRS in this research can be seen in Figure 1. The GRS built in this research used the goodbooks-10k dataset.

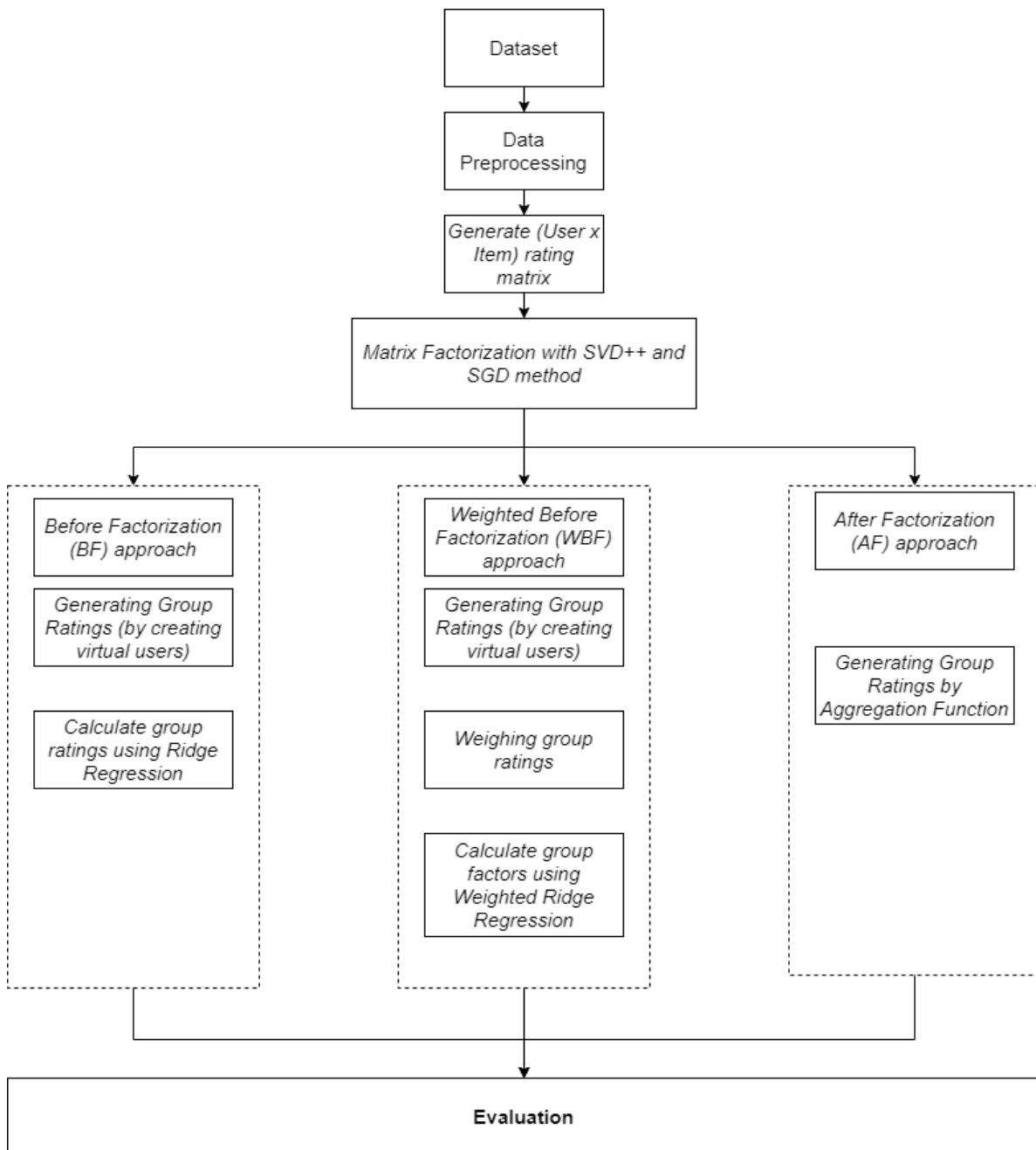


Figure 1. Book GRS design modeling flow

We do data preprocessing on the dataset so that the dataset we use in building GRS has good accuracy. We also generate groups by constructing a user-item rating matrix. After that we did matrix factorization using the AF, BF, and WBF approaches. We evaluate the three approaches to find the best approach for each group.



This research introduces a novel approach for generating recommendations tailored to a collective of users through the utilization of matrix factorization (MF) based collaborative filtering (CF). The fundamental concept behind MF models revolves around the decomposition of the primary rating matrix into multiple matrices, which effectively capture the interactions between users and items. A visual representation of the factorization process is depicted in Figure 2, where the rating matrix is illustrated on the left-hand side, while the right-hand side showcases the bifurcation of the matrix into two distinct entities. The first entity signifies the users residing within the latent factor space, while the second entity represents the items under consideration.

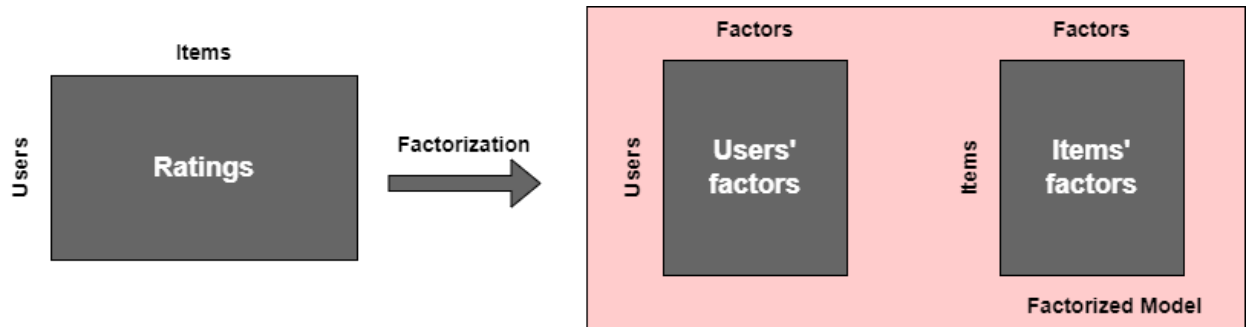


Figure 2. Illustrative demonstration of the Matrix Factorization process

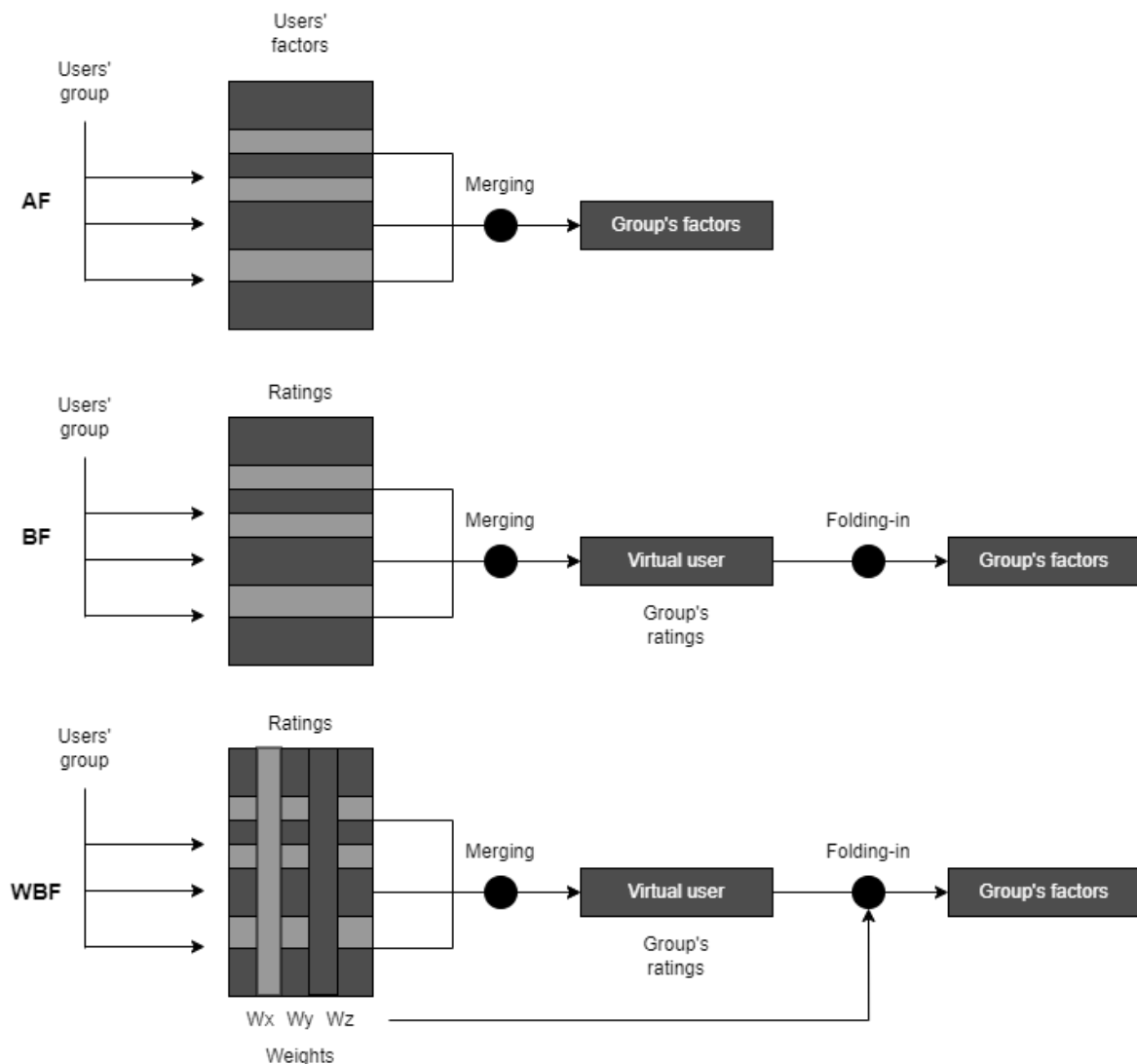


Figure 3. GRS approach with MF method

The principal element of the proposed methodology revolves around calculating the factors that capture the group's interactions with items within the latent factor space. We have devised three distinct approaches to compute these factors, which can be categorized based on the point at which users' data is unified with the group's data. A summary of the proposed methods is presented in Figure 3.



- a. After Factorization (AF), the process of computing recommendations for a group of users using a Matrix Factorization (MF) model is carried out in the most straightforward manner. It involves combining the factors of individual users belonging to the group to obtain the factors specific to the group. This particular approach serves as the fundamental comparison point for the Group Recommender System (GRS) employing Collaborative Filtering (CF) based on matrix factorization.
- b. Before Factorization (BF), the modeling of a group of users is accomplished by constructing a virtual user that encapsulates the item preferences of the users within the group. In order to determine the factors associated with the group, the folding-in technique is employed on the virtual user, effectively incorporating it into the factorized model.
- c. Weighted BF (WBF) is an expanded version of BF that introduces a weighting mechanism to each item rated by the virtual user. These weights are determined by considering the number of ratings received by each item from the users within the group, as well as the level of agreement among these ratings. When computing the latent factors specific to the group, items with higher weights exert a more significant influence compared to those with lower weights.

2.2 Collaborative Filtering

A CF-based Recommender System (RS) utilizes ratings assigned by a group of users to various items [7]. It recommends items that the target user has not yet considered but is likely to appreciate. These ratings are stored in an $m \times n$ matrix, where m represents the number of users and n represents the number of items, as shown in Table 1. The matrix rows contain the ratings provided by users for each item, while the columns store the ratings received by each item. When a new user joins the system, a new empty row is added to the matrix. Similarly, when a new item is added to the catalog, a new empty column is appended to the matrix.

Table 1. An illustration of a rating matrix

User/Item	RS Book	Dilan	Harry Potter	Puisi Malam
Reisa	5	3	5	?
Kina	3	5	?	4
Zota	?	5	4	3
Laza	1	?	5	4

A Collaborative Filtering (CF) system generates recommendations by leveraging the relationships and similarities among users or items. These relationships are derived from the user-item interactions managed by the Recommender System (RS). Consequently, the RS infers the ratings that the target user would assign to items that have not yet been evaluated. Subsequently, the items are ranked based on the estimated rating scores, and the high-ranking items are recommended to the targeted user.

The CF algorithms can be categorized into two primary classes: memory-based and model-based approaches [9]. Memory-based algorithms employ heuristics to analyze a rating matrix and generate recommendations, while model-based algorithms derive a model from the rating matrix and utilize it for item recommendations. Nearest Neighbor (NN) strategies predominantly represent memory-based algorithms, whereas model-based algorithms predominantly utilize Matrix Factorization (MF) techniques. The process followed by a memory-based algorithm involves three steps:

- a. Calculating the similarity between users or items
- b. Creating a neighborhood consisting of similar users or items
- c. Generating recommendations by sorting similar items

On the other hand, MF assumes that the original rating matrix values can be approximated by multiplying matrices with latent features that capture the underlying data patterns. Several successful MF algorithms, such as SVD++, SGD, and ALS, have been employed in CF.

2.3 Matrix Factorization

The MF model maps users and items joint latent factor space to represent user-item interactions. We use the MF model by factoring the rating matrix. Let $\vec{q}_i = (q_{i,1} \dots q_{i,K})$ be the factor vector of item i , and b_i is bias of item i . Meanwhile, suppose $\vec{p}_u = (p_{u,1} \dots p_{u,K})$ the factor vector of user u . Let b_u be bias of user u which is independent of any interaction. The system minimizes equation (1) for a known set of ratings for learning the factor vectors (\vec{q}_i and \vec{p}_u) and bias (b_i and b_u):

$$\min_{\vec{p}_u, \vec{q}_i, b_u, b_i} \sum_{r_{u,i} \neq \cdot} (r_{u,i} - \mu - b_u - b_i - \vec{p}_u \top \vec{q}_i)^2 + \lambda (\|\vec{p}_u\|^2 + \|\vec{q}_i\|^2 + b_u^2 + b_i^2) \quad (1)$$

Where $r_{u,i}$ is the user u 's training rating for item i , μ is the dataset's average rating, and λ is the training process control parameter. The following phrase can be used to determine user u 's prediction for item i ($m_{u,i}$) after the MF has been learned:

$$m_{u,i} = \mu + b_i + b_u + \vec{p}_u \top \vec{q}_i \quad (2)$$



It is necessary to compute the group factor vector (\vec{p}_G) and group bias (b_G) in order to produce group predictions. Following the factoring of the groups, the following formula can be used to determine group G 's predicted for item i :

$$m_{u,i} = \mu + b_i + b_G + \vec{p}_G \top \vec{q}_i \tag{3}$$

We determine the recommendations for group G (R_G) using the predicted values as the set of N items with predicted values but not assessed by every member of the group. The following expressions must be satisfied:

$$\#R_G \leq N \tag{4}$$

$$\forall i \in R_G, \forall u \in G: r_{u,i} = \bullet \tag{5}$$

$$\forall i \in R_G, \forall j \notin R_G: m_{G,i} \geq m_{G,j} \tag{6}$$

2.4 Precision and Recall

Precision and recall are the most frequently employed classification metrics within recommender systems [17][18]. Table 2 comprises two types of data: (1) user rating data, representing evaluations of items already consumed by the members of groups $g_1, g_2, and g_3$. Each group consists of three users. Additionally, (2) predictions of item ratings are included for items $t_1 and t_2$. It is important to note that, for the sake of simplicity, we assume that each group member has provided a rating for every item consumed by them.

Table 2. Illustrates the ratings $r(u_i, t_j)$ and predictions $\hat{r}(u_i, t_j)$ corresponding to items t_1 and t_2 .

Groups	Group members	Ratings $r(u_i, t_j)$			Predictions $\hat{r}(u_i, t_j)$		
		t_1	t_2	...	t_1	t_2	...
g_1	u_1	4.5	2.5	...	3.4	3.8	...
	u_2	3.5	4.5	...	3.7	4.4	...
	u_3	4.5	4.0	...	4.4	3.9	...
	u_4	3.5	2.5	...	3.8	2.6	...
g_2	u_5	4.0	4.5	...	3.7	4.4	...
	u_6	4.5	3.5	...	4.5	3.7	...
	u_7	4.5	3.5	...	3.4	3.8	...
g_3	u_8	3.5	2.5	...	3.7	4.4	...
	u_9	4.0	3.5	...	4.4	3.9	...
...	

The user-specific ratings and predictions are consolidated to form (1) a group rating $r(g_i, t_j)$ and (2) the corresponding group predictions $\hat{r}(g_i, t_j)$ derived from an aggregated predictions-based group recommender system, as shown in Table 3. In a standard group recommendation scenario, a random collection of item ratings at the group level is set aside and utilized as a test set. For the purpose of our example, we assume that, for the groups $g_1, g_2, and g_3$, the ratings pertaining to item $t_1 and t_2$ have been chosen as "holdouts". The rating predictions $\hat{r}(g_i, t_j)$ (presumed to be furnished by a group recommender) are illustrated in Table 3.

Precision refers to the ratio of the number of relevant recommended items (true positives) to the total number of recommended items.

Table 3. The test set in this example consists of group ratings $r(g_i, t_j)$ and corresponding group predictions $\hat{r}(g_i, t_j)$.

Groups	Ratings $r(g_i, t_j)$		Predictions $\hat{r}(g_i, t_j)$	
	t_1	t_2	t_1	t_2
g_1	4.2	3.7	3.8	4.0
g_2	4.0	3.5	4.0	3.6
g_3	4.0	3.2	3.8	4.0

Recall, on the other hand, represents the ratio of the number of relevant recommended items to the total number of relevant items. These metrics are typically evaluated at a specific level denoted as k , which corresponds to the length of the recommended item list. The precision of a GRS, which suggests k items to a group g , can be defined as follows:

$$precision@k(g) = \frac{|predicted_k(g) \cap relevant(g)|}{k} \tag{7}$$

$$recall@k(g) = \frac{|predicted_k(g) \cap relevant(g)|}{|relevant(g)|} \tag{8}$$



Where predicted $k(g)$ denotes a list of k items recommended to group g , and $relevant(g)$ represents all items that are relevant to group g . The calculation of precision and recall is outlined in Table 4, which is derived from the test dataset defined in Table 3.

Table 4. Precision and recall values in the example scenario

Groups	Precision	Relevant	precision@2	recall@2
g_1	2	2	1.0	1.0
g_2	2	1	0.5	1.0
g_3	2	1	0.5	1.0
overall	6	4	0.67	1.0

2.5 Related Works

Ortega., et al. builds GRS for movie domain using the CF paradigm with the MF method with three different aggregation strategy approaches, specifically After Factorization (AF), Before Factorization (BF), Weighted Before Factorization (WBF). The three MF approaches are applied to three different categories of groups, such as small groups consisting of two to four users, medium groups consisting of five to eight users, and large groups consisting of nine to twelve users. The methods are then compared with each other to determine the best method for each group category. In addition, the MF method was also compared with the KNN method. The dataset used is sourced from Netflix and MovieLens. The results of this study indicate that the AF method is best used for small groups and the WBF method is the best approach method for medium groups. In addition, the BF method is best used for large groups.

There are many studies using the CF paradigm with MF techniques with different methods for building GRS. Le Nguyen Hoai Nam., et al. [19] built a GRS using the MF model latent factor technique with the CF paradigm. The purpose of this research is to calculate virtual users with the MF model latent factor technique. Zeshan Aslam Khan., et al. [20] used the MF technique with the Stochastic Gradient Descent (SGD) and Fractional Stochastic Gradient Descent (FSGD) methods. The purpose of this research was to compare the convergence and accuracy of the standard SGD method with the FSGD.

Matrix Factorization (MF) is a commonly employed model-based approach within the field of recommender systems. It represents both items and users through vectors derived from item rating patterns. Over the past few years, MF models have gained significant popularity due to their outstanding attributes, including enhanced accuracy, improved capacity to capture item-user data characteristics, ease of implementation, scalability, flexibility in modeling diverse real-life scenarios, and the ability to incorporate supplementary information [21]. The MF method has many techniques for building recommender system. J. Jiao., et al. [21] conducted research to a new adaptive learning rate (ALR) function is introduced. The function combines the exponential and linear functions, and it is subsequently applied to the SVD++ recommendation algorithm. Christina and Z. K. A. Baizal built a recommender system for the book domain using the SVD algorithm and combined with the Slope One algorithm [22]. We use SVD++ to perform MF in this research.

All research requires evaluation metrics to find out how well the model has been made. Z. Fayyaz., et al. [17] conducted a research to analyze and conclude various evaluation metrics for recommender systems. In building a recommender system, one must choose evaluation metrics and make a comprehensive analysis according to the specific tasks and objectives of the recommending system. We use precision and recall as evaluation metrics to compare and analyze three different aggregation strategies to be applied to three different groups in our GRS.

3. RESULT AND DISCUSSION

3.1 Data

This dataset is a collection of ten thousand popular book ratings that have `book_id`, `user_id`, and `rating` features as shown in Table 5. The `book_id` feature has id from 1 to 10000 and the `user_id` feature has id from 1 to 53424. Each user has given a rating, where the user has rated at least 2 books. The median rating per user is 8.

Table 5. Dataset before data preprocessing

book_id	user_id	rating
1	314	5
1	439	3
1	212	4
1	543	2
1	588	5

We reduce the dataset by filtering the user id from 1-53424 to 1-10000, where the user taken is the user who has the most rating of an item. Obtained 611890 records, where previously there were 981756 records. After that we made the `book_id` and `user_id` indexes sequential starting from 0. The positions of the `book_id` and `user_id`



columns were swapped to make it easier to read the dataset. We divide the dataset into 80% for the train dataset and 20% for the random test dataset. The results of preprocessing data can be seen in Table 6.

Table 6. Dataset after data preprocessing

book_id	user_id	rating
313	0	5
438	0	3
212	0	2
111	0	1
587	0	5

3.2 Creating Groups

We have to ensure that there are enough items to test in order to get the best evaluation results from each approach used in this research. So we've set the threshold to 50, which basically means that there are at least 50 books in the test dataset that have been rated by at least one member of the group. We divided the groups into three categories, such as small groups consisting of three members, medium groups consisting of five members, and large groups consisting of eight members.

3.3 After Factorization (AF)

By combining the factors of each user involved in the group, the AF method considers the user group. Because this method only incorporates AF-generated data and not rating data, users are categorized when the MF model is constructed.

Let $G = \{u_1 \dots u_n\}$ be the set of users that make up group G, Let $\vec{p}_u = (p_{u,1}, \dots, p_{u,k})$ be the user u's factor vector, and let b_u be the user u's bias definition. The definition of \vec{p}_G is the group G factor vector,

$$\vec{p}_G = \begin{pmatrix} h(p_{u_1,1}, \dots, p_{u_n,1}) \\ \vdots \\ h(p_{u_1,K}, \dots, p_{u_n,K}) \end{pmatrix} \tag{9}$$

b_G is referred to as group G's bias,

$$b_G = h(b_{u_1}, \dots, b_{u_n}) \tag{10}$$

3.4 Before Factorization (BF)

The BF approach involves aggregating the preferences of a group of users, denoted as $G = \{u_1, \dots, u_n\}$, into a virtual user, denoted as u_G . This virtual user is created by combining the ratings provided by individual users within the group G. Rating data is used for aggregation. Prior to the MF procedure, aggregation is performed. This strategy is built on two steps, specifically,

- Simulates the rating made by the virtual user u_G on the item, $r_{G,i}$. The following is how this step is carried out using the particular aggregation function h,

$$r_{G,i} = h(r_{u_1,i}, r_{u_2,i}, \dots, r_{u_n,i}) \tag{11}$$

Where $r_{u_1,i}, r_{u_2,i}, \dots, r_{u_n,i}$ is the rating observed from group G users for item i.

- Calculates the virtual user factor vector ($\vec{p}_G = (p_{G,1}, \dots, p_{G,k})$) and virtual user bias (b_G) after rating ($r_{G,i}$) is determined. This step can be completed with the following mathematical notation,

$$\min_{\vec{p}_G, b_G} \sum_{r_{G,i} \neq \bullet} (r_{G,i} - \mu - b_G - b_i - \vec{p}_G \top \vec{q}_i)^2 + \lambda (\|\vec{p}_G\|^2 + \|\vec{q}_i\|^2 + b_{u_G}^2 + q_i^2) \tag{12}$$

With the values q_1, b_i, μ calculated before the learning stage. The mathematical notation above can be simplified to,

$$\min_{\vec{p}_G, b_G} \sum_{r_{G,i} \neq \bullet} ((r_{G,i} - \mu - b_i) - (\vec{p}_G \top \vec{q}_i + b_G))^2 + \lambda (\|\vec{p}_G\|^2 + b_G^2) \tag{13}$$

By definition,

$$s_{G,i} = r_{G,i} - \mu - b_i \tag{14}$$

$$\vec{p}_G^* = (\vec{p}_G, b_G) = (p_{G,1}, \dots, p_{G,K}, b_G) \text{ extends vector } \vec{p}_G \tag{15}$$

$$\vec{q}_i^* = (\vec{q}_i, 1) = (q_{i,1}, \dots, q_{i,K}, 1) \text{ extends vector } \vec{q}_i \tag{16}$$

The previous expression can be derived as follows by definition,



$$\min_{\vec{p}_G} \sum_{s_{G,i} \neq \bullet} (s_{G,i} - \vec{p}_G \cdot \vec{q}_i)^2 + \lambda \|\vec{p}_G\|^2 \quad (17)$$

This reduction complies with the Ridge Regression definition. The virtual user u_G is considered to have rated the item as $\{1, \dots, n_G\}$ and defined matrix A as,

$$A = \begin{pmatrix} q_{1,1} & \dots & q_{1,k} & 1 \\ q_{2,1} & \dots & q_{2,k} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ q_{n_G,1} & \dots & q_{n_G,k} & 1 \end{pmatrix} \quad (18)$$

So,

$$\begin{pmatrix} \vec{p}_G \\ b_G \end{pmatrix} = \begin{pmatrix} p_{G,1} \\ p_{G,2} \\ \vdots \\ p_{G,k} \\ b_G \end{pmatrix} = (A^T A + \lambda I)^{-1} A^T \begin{pmatrix} s_{G,1} \\ s_{G,2} \\ \vdots \\ s_{G,n_G} \end{pmatrix} \quad (19)$$

3.5 Weighted Before Factorization (BF)

Each item will receive weight in the WBF approach based on user group ratings. Priority items are items that are rated the most and items that have the same rating as the user group. Defined $w_{G,i}$ as the weight of item i for group G as,

$$W_{G,i} = \frac{\#\{u \in G | r_{u,i} \neq \bullet\}}{\#G} \cdot \frac{1}{1 + \sigma_{G,i}} \quad (20)$$

Where # indicates the cardinality of a set and $\sigma_{G,i}$ represents the standard deviation of the ratings of group G members for item i. The difference between the BF and WBF approaches is in the second step. In the WBF approach, we calculate the virtual user vector (\vec{p}_G) and the virtual user bias (b_G) with the objective function as follows,

$$\min_{\vec{p}_G, b_G} \sum_{r_{G,i} \neq \bullet} w_{G,i} (r_{G,i} - \mu - b_G - \vec{p}_G \cdot \vec{q}_i)^2 + \lambda (\|\vec{p}_{u_G}\|^2 + \|\vec{q}_i\|^2 + b_{u_G}^2 + q_i^2) \quad (21)$$

This expression is equivalent to Weighted Ridge Regression, so

$$\begin{pmatrix} \vec{p}_G \\ b_G \end{pmatrix} = \begin{pmatrix} p_{G,1} \\ p_{G,2} \\ \vdots \\ p_{G,k} \\ b_G \end{pmatrix} = (A^T W A + \lambda I)^{-1} A^T W \begin{pmatrix} s_{G,1} \\ s_{G,2} \\ \vdots \\ s_{G,n_G} \end{pmatrix} \quad (22)$$

3.6 Evaluation Metrics

We specify group G's precision and recall as follows to evaluate the accuracy of recommendations produced for user groups,

$$precision_G = \frac{\#TP_G}{\#(TP_G \cup FP_G)} \quad (23)$$

$$recall_G = \frac{\#TP_G}{\#T_G} \quad (24)$$

Where TP_G is the set of true positive recommendations, FP_G is the set of false positive recommendations, and T_G is the expected set of recommendations, as shown in the notation below,

$$FP_G = \{i \in R_G | \exists g \in G, \text{ such that } \hat{r}_{g,i} < \theta\} \quad (25)$$

$$TP_G = \{i \in R_G | \exists g \in G, \text{ such that } r_{g,i} \neq \bullet \text{ and } \forall u \in G \hat{r}_{u,i} \neq \bullet \rightarrow \hat{r}_{u,i} \geq \theta\} \quad (26)$$

$$T_G = \{i \in I | \exists g \in G, \text{ such that } r_{g,i} \neq \bullet \text{ and } \forall u \in G \hat{r}_{u,i} \neq \bullet \rightarrow \hat{r}_{u,i} \geq \theta\} \quad (27)$$

We define θ as a threshold used to determine whether a user has a preference for or against an item. Dataset has a feature rating with a scale from 1-5. We set θ to 4 in this research. The user test rating for item i is indicated by $\hat{r}_{u,i}$, and R_G is the set of suggested items for group G (the test rating was not taken into account while determining the recommended items). The parameters that we apply in building the GRS are as follows:

- a. Matrix Factorization Hyperparameters:
 1. No. of factors = 15
 2. Lambda (regularization) = 0.1



3. Neta (learning rate) = 0.05
- b. User Satisfaction Threshold: 4
- c. Group Parameters:
 1. Small group = 3 users
 2. Medium group = 5 users
 3. Large group = 8 users
- d. No. of Recommendations per user = 50

3.7 Result

The results of the evaluation of the GRS that we built in this research can be seen in Table 7 and Table 8. We evaluated the three methods (AF, BF, WBF) by randomly creating 50 groups, where one member could be included in many groups.

Table 7. Precision

Methods	Small Group (K=3)	Medium Group (K=5)	Large Group (K=10)
AF	1	0.5	0.944
BF	1	0.333	1
WBF	1	0.333	1

Table 8. Recall

Methods	Small Group (K=3)	Medium Group (K=5)	Large Group (K=10)
AF	0.0019	0.0044	0.0076
BF	0.0019	0.0009	0.0063
WBF	0.0019	0.0009	0.0087

Table 7 and Table 8 show the results of the precision and recall of each approach for small, medium and large groups. All three approaches have the same precision and recall for small groups. The conclusion from the evaluation results is that these three approaches are suitable for small groups in the goodbooks-10k dataset. We can also see that the AF approach is better than the BF and WBF approaches for the medium group. This can be seen in the precision and recall in Table 7 and Table 8. The WBF approach is also the most effective one for large groups. Based on Table 7, we get good precision results. One of the things that can happen is because the GRS built in this research uses a quality dataset after preprocessing data.

4. CONCLUSION

This research builds a group recommender system for the book domain using the Collaborative Filtering paradigm with the Matrix Factorization method. The dataset used in this research is sourced from goodbooks-10k. We use three approaches, such as AF, BF, and WBF to be applied to three different categories of groups. The three categories of groups have many different members, where small groups consist of three users, medium groups consist of five users, and large groups consist of ten users. The approach methods are compared to find out the best approach method for each group category. The GRS that we have developed can be applied to any dataset that includes user_id, item_id, and rating features. Nevertheless, it is important to note that the research outcomes may vary. The specific findings of our research are based on the goodbooks-10k dataset, which has been subjected to preprocessing techniques in order to enhance the accuracy of the GRS that we have constructed. As future work, we propose that further research can build a GRS by utilizing the method we apply to handle more features, such as genre, price, reading time, etc.

REFERENCES

- [1] S. Reddy, S. Nalluri, S. Kuniseti, S. Ashok, and B. Venkatesh, "Content-based movie recommendation system using genre correlation," *Smart Innovation, Systems and Technologies*, vol. 105, pp. 391–397, 2019.
- [2] J. Zhang, Y. Wang, Z. Yuan, and Q. Jin, "Personalized real-time movie recommendation system: Practical prototype and evaluation," *Tsinghua Sci Technol*, vol. 25, no. 2, pp. 180–191, 2020.
- [3] R. Ahuja, A. Solanki, and A. Nayyar, "Movie recommender system using k-means clustering and k-nearest neighbor," *Proceedings of the 9th International Conference On Cloud Computing, Data Science and Engineering, Confluence 2019*, pp. 263–268, 2019.
- [4] A. Delic, J. Neidhardt, T. N. Nguyen, and F. Ricci, "An observational user study for group recommender systems in the tourism domain," *Information Technology and Tourism*, vol. 19, no. 1–4, pp. 87–116, 2018.
- [5] M. Quadrana, P. Cremonesi, and D. Jannach, "Sequence-aware recommender systems," *ACM Comput Surv*, vol. 51, no. 4, 2018.



- [6] S. Dara, C. R. Chowdary, and C. Kumar, "A survey on group recommender systems," *J Intell Inf Syst*, vol. 54, no. 2, pp. 271–295, 2020.
- [7] M. Srifi, A. Oussous, A. A. Lahcen, and S. Mouline, "Recommender systems based on collaborative filtering using review texts-A survey," *Information (Switzerland)*, vol. 11, no. 6. MDPI AG, Jun. 01, 2020.
- [8] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, "A review on deep learning for recommender systems: challenges and remedies," *Artif Intell Rev*, vol. 52, no. 1, pp. 1–37, Jun. 2019.
- [9] T. Cunha, C. Soares, and A. C. P. L. F. de Carvalho, "Metalearning and Recommender Systems: A literature review and empirical study on the algorithm selection problem for Collaborative Filtering," *Inf Sci (N Y)*, vol. 423, pp. 128–144, Jan. 2018.
- [10] R. Logesh, V. Subramaniaswamy, D. Malathi, N. Sivaramakrishnan, and V. Vijayakumar, "Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method," *Neural Comput Appl*, vol. 32, no. 7, pp. 2141–2164, Apr. 2020.
- [11] D. Kluver, M. D. Ekstrand, and J. A. Konstan, "Rating-based collaborative filtering: Algorithms and evaluation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2018, pp. 344–390.
- [12] W. Chen, F. Cai, H. Chen, and M. D. E. Rijke, "Joint neural collaborative filtering for recommender systems," *ACM Trans Inf Syst*, vol. 37, no. 4, Aug. 2019.
- [13] E. Yalcin, F. Ismailoglu, and A. Bilge, "An entropy empowered hybridized aggregation technique for group recommender systems," *Expert Syst Appl*, vol. 166, no. October 2020, p. 114111, 2021.
- [14] A. Pujahari and D. S. Sisodia, "Aggregation of preference relations to enhance the ranking quality of collaborative filtering based group recommender system," *Expert Syst Appl*, vol. 156, Oct. 2020.
- [15] A. Putri, Z. K. Abdurahman Baizal, and D. Rischasy, "Book Recommender System using Convolutional Neural Network," *Institute of Electrical and Electronics Engineers (IEEE)*, Mar. 2023, pp. 1–6.
- [16] Sri Eshwar College of Engineering and Institute of Electrical and Electronics Engineers, 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS).
- [17] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation systems: Algorithms, challenges, metrics, and business opportunities," *Applied Sciences (Switzerland)*, vol. 10, no. 21, pp. 1–20, Nov. 2020.
- [18] A. Felfernig, L. Boratto, M. Stettinger, and M. Tkalčić, "Evaluating Group Recommender Systems," 2018, pp. 59–71.
- [19] L. N. H. Nam, H. T. H. Vy, L. H. My, L. T. T. Mai, H. T. Gia, and H. L. T. K. Nhung, "An approach to improving group recommendation systems based on latent factor matrices," *ACM International Conference Proceeding Series*, pp. 98–105, 2019.
- [20] Z. A. Khan, N. I. Chaudhary, and S. Zubair, "Fractional stochastic gradient descent for recommender systems," *Electronic Markets*, vol. 29, no. 2, pp. 275–285, 2019.
- [21] 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, 2020.
- [22] Christina and Z. K. A. Baizal, "Book Recommender System Using Singular Value Decomposition Combined with Slope One Algorithm," 2022 10th International Conference on Information and Communication Technology (ICoICT), Bandung, Indonesia, 2022, pp. 346-350.