

An Effective Book Recommendation System using Weighted Alternating Least Square (WALS) Approach

Kavitha V K*, Dr.Sankar Murugesan

Department of Computer Science and Engineering,
Veltech Rangarajan Dr Sagunthala R & D Institute of Science and Technology, Avadi, Chennai, India

Abstract—Book recommendation systems are essential resources for connecting people with the correct books, encouraging a love of reading, and sustaining a vibrant literary ecosystem in an era when information overload is a prevalent problem. With the emergence of digital libraries and large online book retailers, readers may no longer find their next great literary journey without the help of customized book suggestions. This work offers a novel way to improve book recommendation systems using the Weighted Alternating Least Squares (WALS) technique, which is intended to uncover meaningful patterns in user ratings. The suggested approach minimizes the Root Mean Square Error (RMSE), a crucial indicator of recommendation system (RS) performance, in order to tackle the problem of optimizing recommendations. By representing user-item interactions as a matrix factorization problem, the WALS approach improves the recommendation process. In contrast to conventional techniques, WALS adds weighted elements that highlight specific user-item pairings' significance, increasing the recommendations' accuracy. Through an empirical study, the proposed approach demonstrates a significant reduction in RMSE when compared to standard RS, highlighting its effectiveness in enhancing the quality of book recommendations. By leveraging weighted matrix factorization, the proposed method adapts to the nuanced preferences and behaviors of users, resulting in more accurate and personalized book recommendations. This advancement in recommendation technology is poised to benefit both readers and the book industry by fostering more engaging and satisfying reading experiences.

Keywords—Recommendation system; user ratings; matrix factorization; alternating least square; weighted matrix factorization

I. INTRODUCTION

A technology that makes recommendations based on its ability to infer user preferences is called a recommendation system (RS), since it uses all of the information that is accessible at any given time to predict items. In light of the current situation of "information overload," which is defined as an instance in which there is an excessive volume of information to absorb and interpret, when it comes to decision-making, RS can be very beneficial because it provides unique and personalized recommendations [1]. The algorithm utilized determines the recommendation outputs. Recommender systems can be roughly categorized as content-based filtering, collaborative filtering, and hybrid filtering system [2].

A. Content Based Filtering in Recommendation System

In content-based filtering, recommendations are made using previous user selections. The recommender may receive implicit user choices, such as ratings, or explicit user choices, such as surfing patterns. Using the data provided by the user, a user profile is crafted and utilized to inform future recommendations derived from the content of input objects. Items that are particularly suitable to the user are identified using content-based RS by examining item depictions. Recommendation framework details frequently vary depending on how objects are represented. This methodology offers the advantage of being able to explain its suggestions and suggest previously unrated items to individuals with unique interests. When a user provides input on particular items in relation to user-related content, content-based RSs create user preferences, profiles, interests, and impressions based on that information [3]. Products will be suggested to the user based on their preference profile if they match products that have received exceptional appraisals in the past. The increase in user profile proficiency is a crucial component of CBF recommender frameworks [4]. The creation of user profiles has been linked to various learning approaches.

B. Collaborative Filtering in Recommendation System

The process of collaborative filtering involves gathering user evaluations or judgments about various items. Moreover, users with comparable tastes are found in a sizable population. People with similar tastes are merged in order to recommend new products and assist other users in making better selections [5]. CF algorithms are further categorized into model-based and memory-based techniques.

The most suitable definition of CF is "joint effort between individuals to help each other perform filtering by recording their responses to materials they read" [6]. CF approaches play a critical role in the recommendation, even if they are frequently used in conjunction with other filtering approaches such as knowledge-based, content-based, or social-based [7]. It allows users to rate various elements (such as songs, movies, videos, and so on) on a content-based network so that, once sufficient data is stored on the system, recommendations can be sent to each user based on the information offered by those who believe they have the most practical communication with them. Model-based approaches offer recommendations by evaluating the parameters of quantifiable models for user evaluation. The most often used models include matrix

factorization (MF), neural networks, fuzzy systems, Bayesian classifiers, latent features, and genetic algorithms [8].

1) *Collaborative filtering using matrix factorization*: MF is the core for some of the most successful latent factor model realizations. In its simplest form, MF assigns vectors of factors based on implicated information from item rating patterns to both items and users. A recommendation is produced when item and user factors have a high degree of correlation. RS rely on many forms of input data, which are regularly disposed in a matrix where users are represented by one dimension and items of interest by the other. A user-item interaction matrix is factorized into two lower-dimensional matrices via MF models, which can subsequently be utilized to anticipate missing values in the original matrix [9]. The main phases of the MF approach are described below.

a) *The User-Item Interaction Matrix*: In CF, we start with a user-item interaction matrix, often denoted as R . Each row represents a user, each column represents an item, and the values in the matrix depicts user ratings, purchase history, or some other form of interaction. The matrix is typically sparse because not all users have interacted with all items. Let R be an $m \times n$ matrix, where m = number of users, n = number of items and $R[i,j]$ is the interaction (rating or preference) of user i with item j .

b) *MF*: The goal is to factorize the user-item interaction matrix R into two lower-dimensional matrices, typically denoted as U (for users) and V (for items). The concept involves depicting users and items in a reduced-dimensional space, where the dot product of their representations serves as an approximation for the entries in the original matrix.

Let U be an $m \times k$ matrix, where m = number of users, k = the number of latent features (a hyper parameter to be chosen).

Let V be an $n \times k$ matrix, where n = number of items, k = the number of latent features. The approximation of the user-item interaction matrix R is given by the dot product of U and V :

$$R \approx UV^T \quad (1)$$

Here, R is approximated by the dot product of U and the transpose of V .

2) *Objective function*: MF models are trained by minimizing a loss function. A common loss function is the Mean Squared Error (MSE) between the actual user-item interactions (R) and the predicted interactions UV^T .

$$MSE = \sum_{(i,j)} (R[i,j] - (UV^T)[i,j])^2 \quad (2)$$

The objective is to find U and V that minimize this error.

a) *Regularization*: To prevent over fitting, regularization terms are often added to the loss function. L2 regularization is commonly used, and it penalizes large values in U and V :

$$\begin{aligned} \text{Regularized MSE} &= \sum_{(i,j)} (R[i,j] - (UV^T)[i,j])^2 + \lambda(\|U\|^2 \\ &+ \|V\|^2) \end{aligned} \quad (3)$$

Here, λ is a hyper parameter that controls the strength of regularization, and $\|U\|^2$ and $\|V\|^2$ represent the Frobenius norms of U and V , respectively.

b) *Optimization*: MF models are typically trained using optimization techniques like gradient descent to minimize the regularized loss function.

c) *Prediction*: Once the U and V matrices are learned, predictions for user-item interactions can be made by taking the dot product of the corresponding user and item vectors:

$$\text{Prediction for user } i \text{ and item } j: (UV^T)[i,j]$$

MF frameworks, such as Singular Value Decomposition and Alternating Least Squares (ALS), are widely utilized in RSs. They are effective at capturing latent patterns and making personalized recommendations based on user-item interactions.

C. Hybrid Filtering

A hybrid strategy integrates content and collaborative filtering-based techniques to achieve successful recommendation outcomes. The integration of different algorithms benefits from their complementary advantages. In fact, a hybrid strategy outperforms the traditional approach in handling data sparsest and cold start issues. Combining multiple recommender system approaches entails developing two unique recommender systems that rely on content-based and collaborative methodologies. The emergence of RS has demonstrated the value of hybrid processes, which combine several approaches to maximize the advantages of each method. In order to overcome data sparsest and cold start difficulties, CF-based approaches inevitably involve abuse of the extract information source associated with the items our users use. Hybrid recommenders are an interesting problem that proposes a reasonable hypothesis to which one may respond similarly by looking into more recent opportunities.

D. Deep Learning in Recommendations

The volume of data generated in the last few years has increased dramatically compared to previous years. This has led to a greater awareness of the term "big data," which indicates to the vast quantity of unstructured information that is generated and needs more investigation. In recent times, there has been a growing focus on machine learning, specifically deep learning, because of its potential to improve the way large amounts of data are processed and because it can be used to model complex data sets like texts and images. RSs rely heavily on machine learning algorithms, which analyze item and user data to generate specific recommendations. Because of the exponential rise in data availability, the advancement of algorithms, and the availability of more computer resources like GPUs, deep learning has become a promising tool for different data domains. Deep learning models have been effectively utilized in computer vision, speech recognition, and NLP applications. Deep learning has recently been applied to RSs. Despite the significant progress made in the previous twenty years, standard RSs remain unsatisfactory in adequately modeling complicated (e.g., nonlinear) interactions between users and items. Deep neural networks, on the other hand, are universal function approximates that can represent any continuous function. Restricted Boltzmann Machine (RBM) is one of the earliest works that uses the deep learning concept for

collaborative filtering [10]. Nowadays, a wide range of DL models, including Auto encoder (AE), Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Generative Adversarial Network (GAN), are utilized to boost the quality of recommendations.

In the modern world, book recommendation system has become increasingly sophisticated and invaluable, leveraging user-generated book ratings and reviews to guide readers towards literary treasures that resonate with their tastes. These systems harness advanced algorithms to analyze vast databases of reader feedback, identifying patterns and correlations in book preferences. By considering factors such as genre, author, writing style, and even thematic elements, these recommendation engines can suggest books that are highly likely to captivate and engage individual readers. As a result, book enthusiasts can effortlessly discover new and exciting literary adventures, making the most of the digital age's abundance of reading options, while also fostering a sense of community among readers who share similar literary passions. This paper introduces an effective book recommendation system based on WALS method.

The primary contribution of the paper includes:

- Development of an effective book recommendation system using CF.
- Utilization of WALS matrix factorization method to avoid sparsest problems.
- Model evaluation using Good Books Dataset.

II. LITERATURE REVIEW

Dhiman Sarma et al. [11] developed clustering approaches to improve the prediction capabilities of RSs. The datasets were gathered from the Kaggle repository for Goodreads books. The F1 score was 52.84%, whereas the average sensitivity and specificity were 49.76% and 56.74%, respectively. According to the simulation results, the suggested method can more effectively exclude boring books from the list of recommended reading. Yiu-Kai Ng [12] created a book recommendation approach for young readers using MF and content-based filtering techniques. In order to evaluate the effectiveness of recommenders that suggest books to K-12 readers, a benchmark dataset was developed that includes metadata, readability levels, and user and book counts. The error values are computed for the suggested methodologies. Madhuri Kommineni et al. [13] developed a straightforward, intelligible system for book recommendations to assist readers in suggesting the suitable book to be studied next. The introduced method dedicates on training, feedback, management, conveying, arrangement, and providing the user with helpful information to support the recommendation of data items and decision-making. The effectiveness of similarity metrics in book recommendations to users was assessed utilizing the User-Based CF technique. According to the simulation results, a normal distribution after the accuracy rating implies consistency and efficiency.

Sunny Sharma et al. [14] developed a hybrid system-based book RS that predicts recommendations. The suggested method combines content-based and CF. The proposed work is

tested in an online recommender system, enabling us to determine the actual user approval rate of the recommendations made. Based on simulation findings, it was shown that the suggested hybrid filtering strategy works better than both content-based and traditional CF. A personalized recommendation algorithm was realized by Yihan Ma et al. [15] by introducing a large and DL to the book recommendation system sector. Initially, the records' readers' and books' information were gathered. LR and DNN networks are trained together to produce the suggested basic recommendation model. The transformation of the double label into numerous labels enhanced the suggested model. Ultimately, a set of comparison studies was created to confirm the viability of the suggested method. The simulation findings demonstrated that the suggested RS's accuracy is noticeably higher than that of the current techniques. Kiran R et al. [16] suggested a unique DL hybrid recommender system to close the loopholes in CF systems and use DL to reach state-of-the-art predicted accuracy. The method learns non-linear latent variables by representing users and items using embeddings. The method reduces the cold start issue by assimilating side information about users and items into an extremely DNN. The suggested approach combines an increasing weight decay with a lower learning rate. The values are cyclically changed throughout epochs to further increase accuracy. Extensive experiments are carried out across multiple datasets. The outcomes demonstrated that in both non-cold start and cold start scenarios, the suggested solution performs better than the current approaches.

Tulasi Prasad Sariki and G. Bharadwaja Kumar [17] developed an efficient framework to enhance the recommendations given in the book domain by carefully combining natural language processing and DL approaches. The feature space for the existing CBFs has been improved by augmenting them with other relevant attributes by employing this wise combination. The different models that are suggested in the current framework are effective in making use of the book's entire textual material. According to the simulation results, the expanded framework outperforms the baseline models in terms of total suggestion accuracy, with an improvement of 18%. Furthermore, it can be concluded that the suggested approach performs 6% better than the cutting-edge models, including the NCF with Content Embedding Model. An algorithm for digital library recommendations was presented by Fikadu Wayesa et al. [18]. A hybrid book recommendation system using new user profile data was proposed in this study. This hybrid RS merges elements of collaborative and content-based approaches, leveraging user profile data and pattern relationships among users. It uses a content-based component to suggest recommendations to users in the same cluster who have rated books in similar categories. These recommendations are based on the book features and content type information, offering a personalized experience for users with shared interests. A set of comprehensive experiments using information retrieval (IR) assessment criteria are used to measure the efficacy of the suggested technique. The results showed that the recommended strategy outperforms the current techniques, with significant advancements. Dongjin Hou [19] developed a personalized book recommendation method using DL models and the

features and rules governing user savings at university libraries. The deep auto encoder (DAE) is initially improved by the LSTM in order to enable the framework to retrieve the temporal aspects of the data. The resultant book recommendation for the present user is then obtained by using the Softmax function. The suggested approach is validated on the basis of real library lending data. The findings demonstrated that the suggested strategy outperforms a number of other RSs.

Minyu Liu [20] developed a deep-belief network-based book recommendation system. Personalized service suggestions are based on the scenario ontology, which is calculated using the ontology similarity calculation approach. The scenario ontology is then derived based on the library and users' information demands. A deep learning-based personalized intelligent RS idea for book services was proposed by Weiwei Yang [21]. By applying intelligent technologies like machine learning, users can evaluate their big data, create user profiles, correlate resources and users with their profiles, and receive individualized intelligent services. In order to provide users with personalized intelligent book recommendation services that primarily satisfy their potential demands and match their interests, the paper first retrieved users' personalized interests. Next, it identified users' current personalized potential book demands using the plain Bayesian algorithm. Finally, it provided users with personalized book recommendation services. A content-based scientific article recommendation (C-SAR) framework based on a DL approach was developed by Akhil M. Nair et al. [22]. The primary objective of the proposed model was to identify papers by comparing their titles. The most often recurring set of documents from a comparable collection were filtered using an association rule mining Apriori approach, and the Gated Recurrent Unit method was used to determine how similar the documents were. The simulation results revealed that the model performed better than current models that make use of user representations and basic K-means clustering.

A decision tree-based book recommendation system framework was proposed by Anant Duhan and Dr. N. Arunachalam [23]. The effectiveness of a classifier is a major factor in book recommendation systems. Real user data was used to test the system. According to the simulation results, the decision tree classifiers outperformed because of their strong learning capacity, quick classification speed, high accuracy, and straightforward design. A cross-domain book recommendation system employing sequential pattern mining and rule mining was proposed by Taushif Anwar and V. Uma [24]. The suggested approach integrates Wpath, CF, and SPM to recommend the most favored items from many domains with greater recommendation accuracy. Wpath is used in the proposed study to assist in determining the semantic similarity between items that belong to different domains. Topseq criteria are used to extract the sequence's favored items, while the Prefix Span technique assists in retrieving frequently occurring sequences. Initially, the RMSE is used to compare the errors. Finally, the algorithm for pattern mining is examined. The outcome shows that, as compared to the CF-KNN technique, CD SPM performed better. Chendhur et al. [25] introduced an improved CF strategy based on user preferences. The key

objective of this suggested method is to make book recommendations to users based on their reading preferences and to boost the accuracy of those recommendations by enhancing the computational techniques used in the collaborative filtering algorithm. There are four major modules in the proposed approach. The book crossing dataset is used to assess the suggested method. The experimental findings demonstrated that the suggested approach offers the user effective book recommendations.

Neighborhood-Based Collaborative Filtering (NBCF) has its limitations, including a restricted range of recommendations, vulnerability to data sparsest, challenges with handling cold-start scenarios, and relatively slow prediction processing. Model-Based Collaborative Filtering (MBCF) necessitates parameter tuning, entails resource-intensive training phases, and exhibits a sluggish training process. Calculating similarities in graph-based collaborative filtering (GBCF) for large-scale commercial applications can be resource-intensive in terms of both space and time. The computational complexity involved in these calculations can result in significant costs, making the adoption of such systems expensive for businesses. Hybrid-Based Collaborative Filtering (HBCF) involves calculating similarities in large commercial applications, which can be computationally expensive in terms of both space and time complexity. The complexity and adoption of these systems often come with significant costs.

III. MATERIALS AND METHOD

The proposed book recommendation system utilized MF method using WALS method. The detailed block diagram of the suggested methodology is visualized in Fig. 1. The suggested book recommendation system methodology begins with dataset collection and preprocessing to ensure data quality. Data preprocessing involves data cleaning, handling missing values, and outlier detection. Data cleaning is the process of guaranteeing the absence of errors and discrepancies within a dataset. Missing values are imputed or removed, and outliers, which can distort recommendations, are identified and addressed. EDA is crucial to understand the dataset better. It includes visualizations and statistical analysis to uncover patterns, trends, and relationships within the data. EDA can reveal insights such as popular book genres, user preferences, and rating distributions. In this paper, MF with techniques like WALS is applied to generate personalized recommendations based on user-item interactions. In this approach, the user-item interaction data is deployed as a matrix. MF techniques like WALS are applied to decompose this matrix into two lower-dimensional matrices—one denoting user and the other denoting books. The latent factors extracted from these matrices capture user preferences and item characteristics. The performance of the RS is rigorously assessed in terms of Root Mean Squared Error (RMSE) to refine the model and enhance its effectiveness in suggesting books that align with users' preferences and interests.

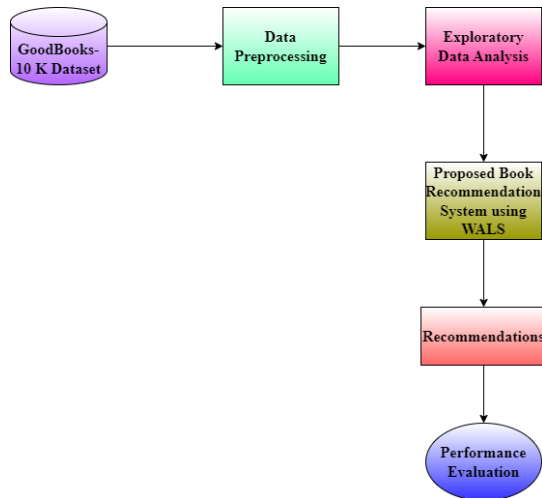


Fig. 1. Block diagram of the proposed book recommendation system.

A. Dataset Description

The goodbooks-10k dataset has been utilized in this work, which comprises ratings for a collection of ten thousand popular books. Typically, each book is associated with approximately 100 reviews, although there are instances of books with fewer ratings. These ratings are on a scale of one to five. The book IDs and user IDs in this dataset follow a sequential numbering scheme, ranging from 1 to 10000 for books and 1 to 53424 for users. It's noteworthy that all users have provided at least two ratings, and the median number of ratings per user is eight. The dataset is organized into various

folders, including "ratings.csv" for ratings data, "to_read.csv" containing user-to-book pairs indicating books marked as "to read," "books.csv" providing metadata for each book (e.g., goodreads IDs, authors, titles, average ratings), "book_tags.csv" containing tags, shelves, and genres granted to books by users, and "tags.csv" which maps tag IDs to their corresponding names. For our book recommendation task, we specifically leveraged the "ratings.csv" file. Table I summarizes the key attributes within the dataset, and Fig. 2 offers a visual representation of the dataset structure.

TABLE I. FEATURE DESCRIPTION OF DATASET

Features	Description
bookID	Unique identification number for each book
title	Name under which book was published
authors	Name of the Authors of the book
average_rating	Average rating of the book received in total.
isbn	International standard book number
isbn13	13-digit isbn to identify the book
language_code	Primary Language of the book
num_pages	Number of pages the book contains
ratings_count	Total Number of ratings the book received.
text_reviews_count	Total number of written reviews received.
publication_date	Date when the book was first published
publisher	Name of the Publishers

bookID	title	authors	average_rating	isbn	isbn13	language_code	num_pages	ratings_count	text_reviews_count	publication_date	publisher
0	1 Harry Potter and the Half-Blood Prince (Harry ...	J.K. Rowling/Mary GrandPré	4.57	0439785960	9780439785969	eng	652	2095690	27591	9/16/2006	Scholastic Inc.
1	2 Harry Potter and the Order of the Phoenix (Har...	J.K. Rowling/Mary GrandPré	4.49	0439358078	9780439358071	eng	870	2153167	29221	9/1/2004	Scholastic Inc.
2	4 Harry Potter and the Chamber of Secrets (Harry...	J.K. Rowling	4.42	0439554896	9780439554893	eng	352	6333	244	11/1/2003	Scholastic
3	5 Harry Potter and the Prisoner of Azkaban (Harr...	J.K. Rowling/Mary GrandPré	4.56	043965548X	9780439655484	eng	435	2339585	36325	5/1/2004	Scholastic Inc.
4	8 Harry Potter Boxed Set Books 1-5 (Harry Potte...	J.K. Rowling/Mary GrandPré	4.78	0439682584	9780439682589	eng	2690	41428	164	9/13/2004	Scholastic

Fig. 2. Visualization of dataset.

B. Data Preprocessing and Exploratory Data Analysis (EDA)

Data preprocessing is a vital and often time-consuming phase in the data analysis process, as it plays a key role in improving data quality by addressing issues such as missing values, outliers, inconsistencies, and errors. The precision and dependability of subsequent analyses are significantly shaped by the quality of the data. Missing data, in particular, can significantly affect data analysis quality. There are several methods to manage missing data, including data deletion and imputation. Managing missing values is a critical aspect of data preprocessing since it directly impacts the dataset's quality and

reliability. Various strategies are available to address missing data, such as eliminating rows with missing values, filling in missing values with measures like the mean, median, or mode, or utilizing more advanced techniques like regression imputation or predictive modeling.

Identifying and managing outliers is a critical component of data preprocessing, serving to pinpoint data instances that exhibit substantial deviations from the bulk of the dataset.

These outliers represent data points that are exceptional or divergent from the norm and have the potential to exert a substantial influence on the outcomes of data analysis, the

performance of learning models, and the integrity of statistical inferences.

Exploratory Data Analysis (EDA) is a fundamental stage in data analysis dedicated to providing a comprehensive overview of a dataset's essential features. This process serves as a crucial step in gaining profound insights into the data prior to employing more advanced statistical and learning models. To start the EDA process, descriptive statistics are employed to offer an initial snapshot of the data's central tendencies and variability. Key descriptive statistics often encompass metrics

like the mean, median, mode, standard deviation, and range. Fig. 3 illustrates the descriptive statistics of the collected dataset, offering a concise representation of these critical data characteristics.

Data visualization is a powerful EDA technique that involves creating visual representations of data. This includes scatter plots, line charts, bar plots, heatmap, and more. The visualization of top 10 authors with maximum book published is shown in Fig. 4. The most occurring book in the collected dataset is visualized in Fig. 5.

	average_rating	num_pages	ratings_count	text_reviews_count	year
count	11123.000000	11123.000000	1.112300e+04	11123.000000	11123.000000
mean	3.934075	336.405556	1.794285e+04	542.048099	2000.169019
std	0.350485	241.152626	1.124992e+05	2576.619589	8.247227
min	0.000000	0.000000	0.000000e+00	0.000000	1900.000000
25%	3.770000	192.000000	1.040000e+02	9.000000	1998.000000
50%	3.960000	299.000000	7.450000e+02	47.000000	2003.000000
75%	4.140000	416.000000	5.000500e+03	238.000000	2005.000000
max	5.000000	6576.000000	4.597666e+06	94265.000000	2020.000000

Fig. 3. Descriptive statistics of dataset.

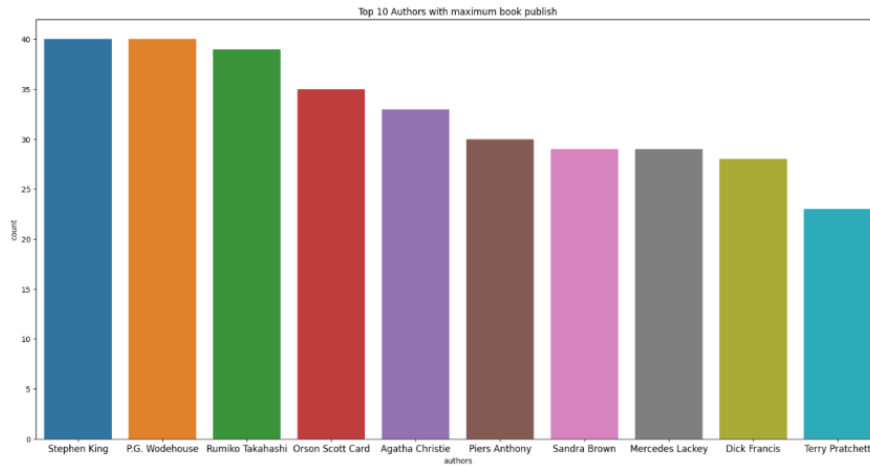


Fig. 4. Data visualization.

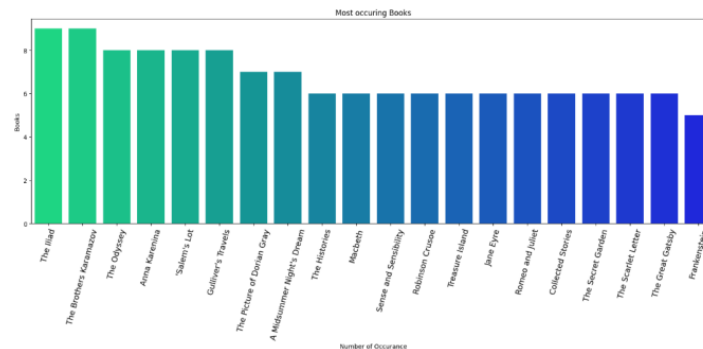


Fig. 5. Visualization of most occurring books in the dataset.

The distribution plot of rating variable from the dataset is shown in Fig. 6. Distribution plots, such as histograms and kernel density estimates (KDE), are essential tools during

EDA. It offers insightful information on the type of data, its properties, and any possible trends or problems.

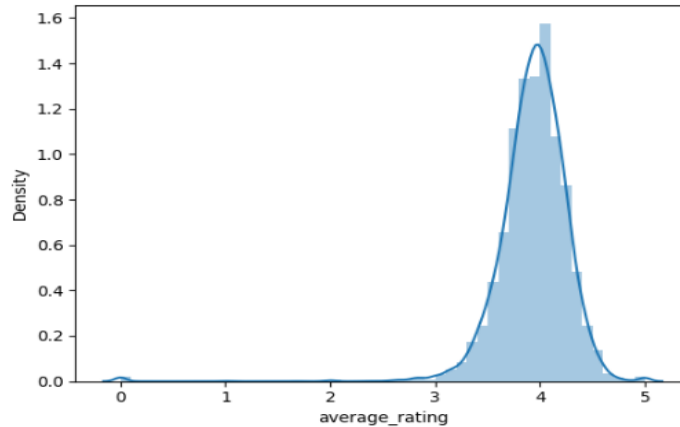


Fig. 6. Distribution Plot of Ratings from dataset.

C. Proposed Book Recommendation System Using Weighted Alternating Least Square (WALS) Method

WALS is a popular MF technique frequently employed in RSs, including those used in book recommendations. Its primary objective is to break down the user-item interaction matrix into two lower-dimensional matrices: one for users and another for items. These matrices capture latent factors that encapsulate user preferences and item characteristics. WALS serves as an extension of the Alternating Least Squares (ALS) method, a widely utilized collaborative filtering approach [26].

To understand the ALS method, consider a rating data matrix R with dimensions $m \times n$, where there are n users and m items. The entry R_{ui} at the $(u, i)^{th}$ position within the matrix R corresponds to the rating given by user u to item i . Since this matrix captures user-item interactions, it is naturally sparse due to the fact that not all users rate or interact with every item, leading to numerous empty or missing values. MF offers a viable solution to the intricate problem of handling sparse matrices. It involves decomposing the R matrix into two k -dimensional vectors, often specified as "factors." These factors play a vital role in capturing underlying patterns and relationships between users and items, enabling more effective recommendations.

- x_u is k dimensional vectors summarizing's every user u .
- y_i is k dimensional vectors summarizing every user i .

$$\text{Let, } r_{ui} \approx x_u^T y_i \quad (4)$$

$$x_u = x_1, x_2, x_3, \dots, x_n \in \mathbb{R}^k \quad (5)$$

$$y_i = y_1, y_2, \dots, y_n \in \mathbb{R}^k \quad (6)$$

Eq. (4) can be assembled as an optimization problem to find:

$$\text{argmin } \sum_{r_{ui}} (r_{ui} - x_u^T y_i)^2 + \lambda (\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2) \quad (7)$$

The regularization factor denoted as λ is commonly referred to as "weighted λ regularization." This parameter, λ , serves the purpose of mitigating over fitting, and its value can be customized to fine-tune the model's performance in mitigating over fitting. The default value for λ is typically set at 1, but it can be modified as needed to achieve optimal results.

When the set of variables x_u is held constant, the objective function for y_i becomes convex. Conversely, when the set of variables y_i is constant, the objective function for x_u also becomes convex. By iteratively optimizing x_u and y_i using this alternating approach until convergence, we employ a technique known as Alternating Least Squares (ALS) to determine the optimal values for these variables.

WALS is a MF approach widely employed in RSs to model user-item interactions. The primary objective of WALS is to factorize the user-item interaction matrix while optimizing a weighted least squares objective function. By iteratively updating latent factor matrices for users and items, WALS discovers hidden patterns that capture user preferences and item characteristics. The incorporation of weights allows for flexibility in handling missing data and emphasizing the importance of certain interactions. With its alternating least squares optimization strategy, WALS efficiently generates recommendations by approximating missing entries in the interaction matrix, making it a valuable tool for personalized and efficient RSs.

The goal of WALS is to approximate the user-item interaction matrix R as a product of two lower-rank matrices, U (user matrix) and V (item matrix), where:

$$R \approx U * V^T \quad (8)$$

R is the user-item interaction matrix, where $R(i, j)$ represents the interaction or rating of user i for item j .

WALS aims to factorize the user-item matrix R into two matrices, U and V , such that the approximation $R \approx U * V^T$ holds.

WALS uses a weighted least squares approach to optimize the factorization. It introduces a weight matrix W , where $W(i, j)$ is the weight associated with the interaction between user i and item j . This weight can be set based on various criteria, like the confidence of the user's rating or the number of interactions. The objective function to minimize in WALS is the weighted least squares loss, defined as follows:

$$L(U, V) = (R(i, j) - U(i, :) * V(j, :))^2 \quad (9)$$

Here, $U(i, :)$ represents the i^{th} row of the user matrix U , and $V(j, :)$ represents the j^{th} row of the item matrix V .

To minimize the loss function, WALS uses an alternating optimization approach. It alternates between updating U and V while keeping the other matrix fixed.

Updating U : For each user i , the update rule for $U(i, :)$ is given by:

$$U(i, :) = (\sum(W(i, j) * V(j, :)^T * V(j, :)))^{-1} * \sum(W(i, j) * R(i, j) * V(j, :)) \quad (10)$$

Updating V : For each item j , the update rule for $V(j, :)$ is given by:

$$V(j, :) = (\sum(W(i, j) * U(i, :)^T * U(i, :)))^{-1} * \sum(W(i, j) * R(i, j) * U(i, :)) \quad (11)$$

The alternating optimization process continues until convergence criteria are met, such as a maximum number of iterations, or a small change in the loss function. Once the factorization is complete, you can make recommendations for users by computing the predicted ratings as follows:

$$R_{\text{pred}}(i, j) = U(i, :) * V(j, :)^T \quad (12)$$

In practice, regularization terms are often added to the objective function to prevent over fitting. These regularization terms penalize large values in U and V . The step-by-step operation of WALS is shown below.

Step 1- Initialization: The process starts with the random initialization of matrices representing user features and item features in the algorithm.

Step 2- Alternating optimization: The algorithm alternates between optimizing the user-feature and item-feature matrices.

Step 3- User-feature optimization: The algorithm updates the user-feature matrix while keeping the item-feature matrix fixed. It does this by solving a least-squares problem using the weighted ratings of each user. The weight assigned to each rating is based on the confidence level of the user's rating.

Step 4- Item-feature optimization: The algorithm updates the item-feature matrix while keeping the user-feature matrix fixed. It does this by solving a least-squares problem using the weighted ratings of each item. The weight assigned to each rating is based on the confidence level of the user's rating.

Step 5- Convergence: The algorithm repeats steps 3 and 4 until the disparity between the predicted ratings and actual ratings is minimized, or a predetermined number of iterations is attained.

Step 6- Prediction: Once the algorithm has converged, it can use the user-feature and item-feature matrices to predict the ratings of unseen user-item pairs.

IV. RESULTS AND DISCUSSION

A. Hardware and Software Setup

The proposed book recommendation system is implemented on Google Collaboratory platform. Google Collaboratory, is a cloud-based platform that provides free access to a virtual environment with a variety of resources, including a Nvidia K80 GPU with 16GB of GPU memory. Colab supports Python, a versatile and widely-used programming language, and it comes pre-installed with popular libraries like TensorFlow and Keras. The key parameter of WALS utilized by the proposed book recommendation system is tabulated in Table II.

TABLE II. PARAMETERS

Parameters	Explanation
Rank of the factorized matrix (K)	The number of latent factors or dimensions used in the matrix factorization process.
Regularization parameter	Hyper parameter used to control the degree of regularization applied to the coefficients or during training.
Number of iterations	Determines how many times the algorithm will update its parameters or make incremental changes during the training process.

B. Results

The effectiveness of the suggested book RS is assessed through the use of RMSE, a broadly adopted measurement for analyzing the performance of RS. RMSE serves as a means to gauge the accuracy of a RS's predictions by quantifying the discrepancy between the system's predicted values and the

actual user preferences, ratings, or interactions with items, such as books, movies, or products. In the realm of RS, RMSE is employed to determine how closely the system's predictions align with the real interactions of users with items, generally structured in a user-item interaction matrix where rows represent users, columns represent items, and the matrix cells contain user interactions, which may include ratings, purchase

history, clicks, or any other relevant user-item engagement data. The RS predicts the missing values in the interaction matrix. These predictions represent the system's estimate of how a user would rate or interact with items they have not yet interacted with. Actual user ratings or interactions for some items are available in the dataset. These are the ground-truth values that the RS aims to predict. For each user-item pair where both actual and predicted values are available, RMSE calculates the squared difference between the actual and predicted values. The squared differences are then averaged across all user-item pairs. The RMSE value is computed by initially calculating the average and then finding the square root of that average. RMSE can be expressed as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (r_i - \hat{r}_i)^2} \quad (13)$$

Table III contains a tabulation of the suggested book recommendation system's performance under various conditions.

The model's performance is significantly affected by the choice of hyper parameters, particularly λ and the number of iterations. When λ is set at 0.1, the RMSE diminishes as the number of iterations rises, with the lowest RMSE achieved at 20 iterations. However, for a lower λ of 0.01, the RMSE is only marginally higher, indicating that a more conservative regularization may be preferred. Furthermore, the training and testing times show an increase as the number of iterations and λ value grow, indicating a trade-off between model accuracy and computational efficiency. The best-performing

configuration seems to be with $\lambda = 0.01$ and 20 iterations, achieving a reasonable RMSE while maintaining a relatively lower computational cost. In summary, the WALs method appears promising for book recommendation. The graphical representation of RMSE values under $K=5$ and 10 iterations are shown in Fig. 7 also the training time distribution, testing time distribution is outlined in Fig. 8, Fig. 9. The recommendation output is tabulated in Table IV.

TABLE III. PERFORMANCE OF PROPOSED BOOK RECOMMENDATION SYSTEM

K	λ	Number of Iterations	RMSE	Training Time	Testing Time
5	0.1	10	3.880	225.670	0.00168
10	0.1	10	3.833	413.3266	0.00176
20	0.1	10	3.77	832.388	0.001498
5	0.01	10	3.879	227.261	0.001847
5	0.001	10	3.878	221.310	0.001836
5	0.001	20	3.8787	448.85	0.001432
5	0.001	25	3.878	567.545	0.00155
5	0.001	30	3.8795	686.3248	0.001526

TABLE IV. RECOMMENDATION OUTPUT

User ID	Recommended Books
1	[1632, 3988, 1775, 1548, 3051, 2260, 1229, 1478, 1765, 2582]

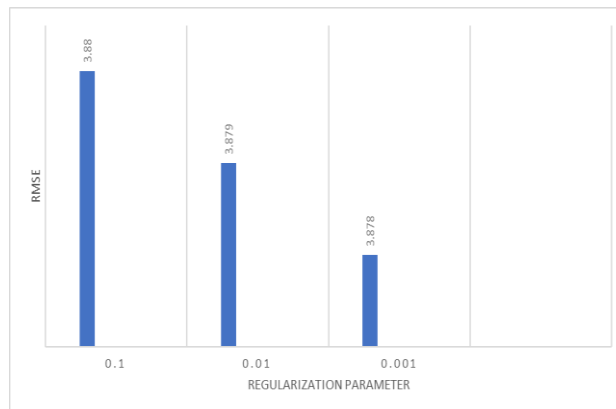


Fig. 7. RMSE under K=5 and no. of iterations=10.



Fig. 8. Training Time Distribution (K=5, number of iterations=10).

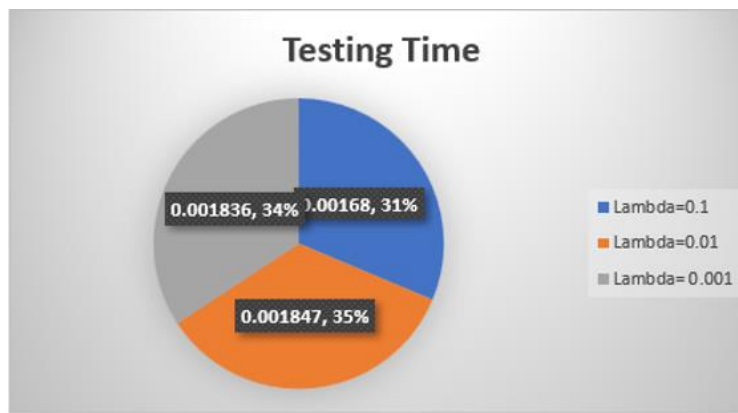


Fig. 9. Testing Time Distribution (K=5, number of iterations=10).

C. Discussion

The assessment of the suggested book recommendation system's performance, evaluated through Root Mean Square Error (RMSE), reveals insights into its effectiveness across varied hyper parameter configurations. Results from Table III showcase RMSE values ranging from 3.770 to 3.880, indicating a moderate accuracy in predicting user-item interactions. Notably, λ and the number of iterations significantly impact the system's performance, with higher λ values leading to lower RMSE but increased computational costs. Conversely, lower λ values result in marginally higher RMSE but offer computational efficiency. Graphical representations aid in understanding these trends. Ultimately, a balance between RMSE and computational resources is achieved with $\lambda = 0.01$ and 20 iterations, highlighting the promise of the Weighted Alternating Least Squares (WALS) method for book recommendations. Overall, hyper parameter tuning plays a crucial role in optimizing recommendation system performance, offering opportunities for enhancing user satisfaction and engagement in book recommendations and beyond.

V. CONCLUSION

In the modern digital era, book recommendation systems have become integral to elevating the reading experience for individuals. These systems leverage advanced algorithms, user data, and content analysis to provide personalized book suggestions, helping readers discover new titles that align with their preferences and interests. This paper presents an effective book recommendation system using the WALS method using ratings. WALS is a MF technique commonly used in collaborative filtering-based RSs. The major goal is to reveal hidden elements that capture the fundamental traits of users and books by breaking down the matrix of user-item interactions. Unlike traditional alternating least squares, WALS introduces the concept of weighting, allowing it to assign different levels of importance to user-item interactions. This weighting factor can reflect the confidence or reliability of a user's rating, making WALS particularly useful in scenarios with sparse or noisy data. By iteratively optimizing these latent factors, WALS refines its recommendations, ultimately providing users with personalized suggestions by estimating their preferences based on the discovered latent features. The

proposed method yielded outstanding results with considerably diminished RMSE values. This achievement underscores the efficacy of WALS in enhancing the accuracy of book recommendations, allowing users to discover books that align more closely with their individual preferences. A smaller RMSE value indicates that the system is more proficient at predicting user behavior, resulting in a more gratifying and individualized reading experience.

REFERENCES

- [1] Eckhardt, A. (2009). Various aspects of user preference learning and recommender systems. In DATESO (pp. 56-67).
- [2] Bobadilla, J et al. 2013, 'Recommender systems survey', Knowledge Based Systems, vol. 46, pp. 109-132.
- [3] Lops, P et al. 2019, 'Trends in content-based recommendation: Preface to the special issue on Recommender systems based on rich item descriptions', User Modeling and User-Adapted Interaction, vol. 29, no. 2, pp. 239-249.
- [4] Pazzani, M. J. and Billsus, D. (2007), Content-based recommendation systems, in 'The adaptive web', Springer, pp. 325-341.
- [5] Deldjoo, Y., Elahi, M., Cremonesi, P., Garzotto, F., Piazzolla, P. and Quadrana, M. (2016), 'Content-based video recommendation system based on stylistic visual features', Journal on Data Semantics 5(2), 99-113.
- [6] Cai, Y., Leung, H.-f., Li, Q., Min, H., Tang, J. and Li, J. (2014), 'Typicality-based collaborative filtering recommendation', IEEE Transactions on Knowledge and Data Engineering 26(3), 766-779.
- [7] Murali, MV, Vishnu, TG, & Victor, N 2019, 'A Collaborative Filtering based Recommender System for Suggesting New Trends in Any Domain of Research', in, 2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019, Institute of Electrical and Electronics Engineers Inc., pp.550-553.
- [8] C, ano, E. and Morisio, M. (2017), 'Hybrid recommender systems: A systematic literature review', Intelligent Data Analysis 21(6), 1487-1524.
- [9] Melville, P. and Sindhvani, V. (2011), Recommender systems, in 'Encyclopedia of machine learning', Springer, pp. 829-838.
- [10] da Silva, E. Q., Camilo-Junior, C. G., Pascoal, L. M. L. and Rosa, T. C. (2016), 'An evolutionary approach for combining results of recommender systems techniques based on collaborative filtering', Expert Systems with Applications 53, 204-218.
- [11] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In Proceedings of the 24th international conference on Machine learning, pages 791-798. ACM, 2007.
- [12] Sarma, D., Mitra, T., & Hossain, M. S. (2021). Personalized book recommendation system using machine learning algorithm. International Journal of Advanced Computer Science and Applications, 12(1).

- [13] Ng, Y. K. (2020). CBRec: a book recommendation system for children using the matrix factorization and content-based filtering approaches. *International Journal of Business Intelligence and Data Mining*, 16(2), 129-149.
- [14] Kommineni, M., Alekhya, P., Vyshnavi, T. M., Aparna, V., Swetha, K., & Mounika, V. (2020, January). Machine learning based efficient recommendation system for book selection using user based collaborative filtering algorithm. In *2020 Fourth International Conference on Inventive Systems and Control (ICISC)* (pp. 66-71). IEEE.
- [15] Sharma, S., Rana, V., & Malhotra, M. (2022). Automatic recommendation system based on hybrid filtering algorithm. *Education and Information Technologies*, 1-16.
- [16] Ma, Y., Jiang, J., Dong, S., Li, C., & Yan, X. (2021, May). Book recommendation model based on wide and deep model. In *2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID)* (pp. 247-254). IEEE.
- [17] Kiran, R., Kumar, P., & Bhasker, B. (2020). DNNRec: A novel deep learning-based hybrid recommender system. *Expert Systems with Applications*, 144, 113054.
- [18] Sariki, T. P., & Guntur, B. K. (2022). An Aggrandized Framework for Enriching Book Recommendation System. *Malaysian Journal of Computer Science*, 35(2), 111-127.
- [19] Wayesa, F., Leranso, M., Asefa, G., & Kedir, A. (2023). Pattern-based hybrid book recommendation system using semantic relationships. *Scientific Reports*, 13(1), 3693.
- [20] Hou, D. (2022). Personalized book recommendation algorithm for university library based on deep learning models. *Journal of Sensors*, 2022.
- [21] Liu, M. (2022). Personalized Recommendation System Design for Library Resources through Deep Belief Networks. *Mobile Information Systems*, 2022.
- [22] Yang, W. (2022). Personalized intelligent recommendation algorithm design for book services based on deep learning. *Wireless Communications and Mobile Computing*, 2022, 1-8.
- [23] Nair, A. M., Benny, O., & George, J. (2021). Content based scientific article recommendation system using deep learning technique. In *Inventive Systems and Control: Proceedings of ICISC 2021* (pp. 965-977). Springer Singapore.
- [24] Duhan, A., & Arunachalam, N. (2023). Book Recommendation System Using Machine Learning (No. 10012). EasyChair.
- [25] Anwar, T., & Uma, V. (2022). CD-SPM: Cross-domain book recommendation using sequential pattern mining and rule mining. *Journal of King Saud University-Computer and Information Sciences*, 34(3), 793-800.
- [26] Chendhur, K. K., Priya, V., Priya, R. M., & Lakshmi, S. L. (2021). Book recommender system using improved collaborative filtering. *International Journal of Research in Engineering, Science and Management*, 4(4), 51-56.