SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I10P104

Volume 11 Issue 10, 43-57, October 2024

Original Article

Enhancing Book Recommendation Systems: A Deep Dive into Weighted Alternating Least Square (WALS) and Neural Collaborative Filtering (NCF) with Feature Optimization

Kavitha V K[1], Sankar Murugesan[2]

[1,2]*Department of Computer Science and Engineering,*
Veltech Rangarajan Dr Sagunthala R & D Institute of Science and Technology, Tamilnadu, India.

[1]*Corresponding Author : vtd918@veltech.edu.in*

Received: 04 August 2024 Revised: 05 September 2024 Accepted: 04 October 2024 Published: 30 October 2024

Abstract - *A Recommendation System (RS) is a kind of data filtering framework that can forecast user preferences or ratings for various categories, including music, movies, books, social media tags, books, and general products. A book recommendation system is essential to connect readers with appropriate books, encourage a love of reading, and preserve an exciting literary community. With the rise of online bookstores and digital libraries, readers would not be able to discover their next outstanding literary adventure without personalized book recommendations. This work primarily aims to present a comparative analysis of the performance of suggested book recommendation systems employing the Neural Collaborative Filtering (NCF) approach with feature optimization and the Weighted Alternating Least Square (WALS) approach. The proposed models were evaluated on the GoodBooks-10Kdataset. Root Mean Square Error (RMSE) values were employed to compare the models' performances. A system that is better at forecasting user behavior will provide a more satisfying and customized reading experience; a decreased RMSE score indicates this. The simulation outcome indicates that the suggested method produced excellent outcomes with significantly lower RMSE values. It also demonstrates that NCF with feature optimization exhibits superior recommendation performance regarding RMSE, outperforming WALS consistently with lower values. This outcome demonstrates how the recommended techniques can enhance the effectiveness of book recommendations and help users select books that are more compatible with their own tastes.*

Keywords *- Alternating least square method, Collaborative filtering, Matrix factorization, Neural collaborative filtering, Recommendation system, Root Mean Square Error.*

1. Introduction

Recommendation Systems (RSs) provide product or item recommendations based on a user's particular interests, a group of related products, or comparable users. These kinds of data can be acquired implicitly through user behavior and user-item interaction or directly through user-item ratings [1-3]. Academics and educators employ recommender systems to quickly sort through the necessary material [4-9].

An RS uses an information filtering method to build a suggestion model based on data derived from a user's browsing history and interests, as well as their physiological profile. Information filtering frameworks use recommendation algorithms to make recommendations for books, music, and movies depending on user requests. These algorithms suggest products personalized to each user's unique interests. Figure 1 depicts the recommendation function's visualization.

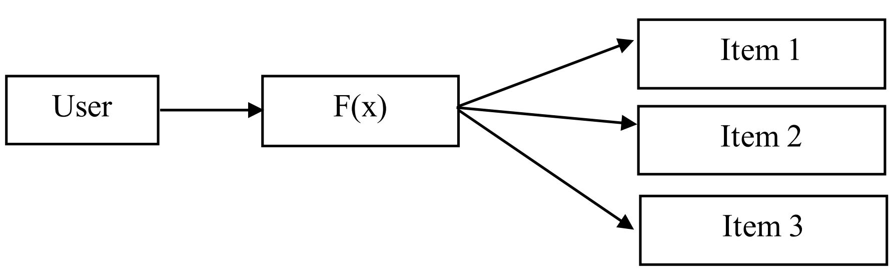


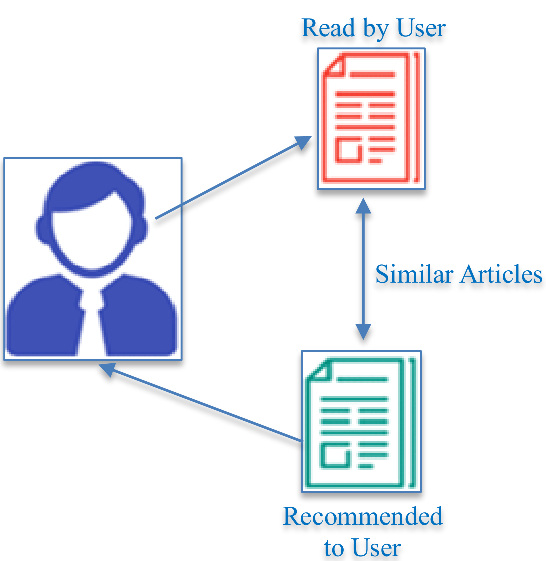
Fig. 1 Recommendation function

The basic idea behind content-based filtering is to recommend products based on users' interests [10]. One popular approach to making user recommendations is the Collaborative Filtering (CF) methodology. The methodology focuses on recommending publications based on previously conducted searches and metadata. It involves creating a user-item rating matrix with each user's preferences for the objects they have viewed. The relationship between comparable users determines how unseen items should be rated. Systems for CF can generate original recommendations and are independent of domains [11]. Depending on the model and system memory, CF employs two different techniques [12]. The memory-based approach uses the system's memory to provide predictions. The memory-based technique envisages items for the current user based on their shared preferences. Once the current user has rated an item, the method creates a matrix based on their ratings and then generates new recommendations based on those ratings. Figure 2 illustrates the visualization of content-based filtering and CF in an RS.

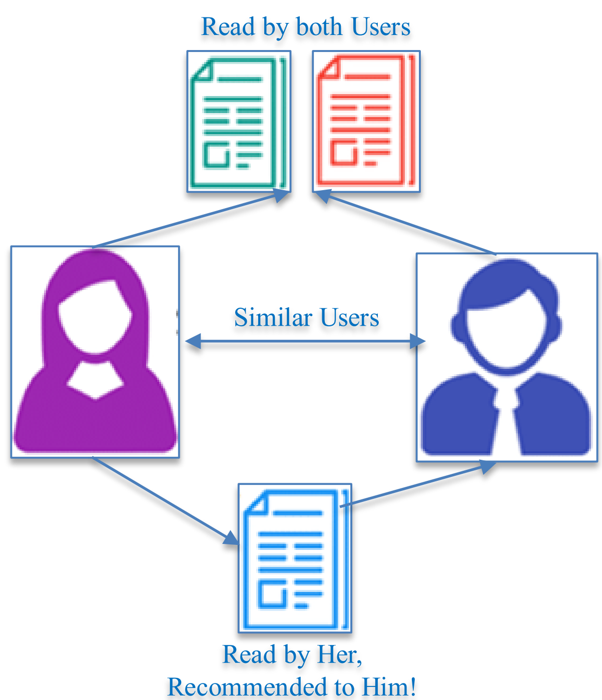
Book RSs are vital in improving the accessibility and customization of reading experiences in the modern world. These systems can customize recommendations based on user preferences by utilizing advanced algorithms and data analytics, which makes literature discovery more effective and enjoyable. This enables publishers and authors to reach their intended audiences and readers to locate intriguing and relevant books. Additionally, RSs increase the global reach of literary work by bridging the gap between digital platforms and traditional libraries. Personalized recommendations can potentially improve learning and encourage lifetime reading habits in educational settings, highlighting their value even more. Hence, the main objective of the suggested research work is to offer a comparative analysis of the proposed book RSs. The major contribution of the paper includes:

- Develop an efficient book RS by leveraging CF techniques to enhance the precision and relevance of suggested readings based on user preferences and behaviours.
- Utilize the Weighted Alternating Least Squares (WALS) Matrix Factorization (MF) method to address and mitigate sparsity issues, guaranteeing robust recommendations even with limited user-item interaction data.
- Create an RS founded on the Neural Collaborative Filtering (NCF) approach with feature optimization, integrating deep learning methods to grab complex user-item interactions and boost recommendation performance.
- Conduct a comprehensive performance comparison of the proposed book RSs, evaluating their effectiveness and scalability to identify the most suitable solution for various application contexts.

The remaining section of the paper is arranged as follows: Section 2 delivers a review of the literature, highlighting topics that require further investigation. Section three offers a thorough explanation of the methodology. The fourth section thoroughly discusses the results of the suggested technique. In Section Five, the paper concludes with a summary of the findings.



(a)



(b)

Fig. 2(a) Content-based filtering, and (b) CF in an RS.

2. Related Works

Afoudi Yassine et al. [13] developed an approach for building a hybrid recommender framework that integrates CF with the neural network techniques of content-based approaches and self-organizing maps. The movie dataset served as the evaluation platform for the suggested RS. The simulation results revealed that using self-organizing maps with CF instead of K-means clustering with CF resulted in lower RMSE in most clusters. Furthermore, it demonstrated that the proposed RS had increased the quality and effectiveness of movie RSs.

Fikadu-Wayesa et al. [14] offered CF and content-based filtering with semantic relationships to identify the relationships and provide readers in a digital library with knowledge-based book recommendations. By grouping the patterns in a semantically equivalent way, the clustering algorithm was able to grab the similarity between the books that the new user had extracted. The effectiveness of the proposed model is evaluated using an extensive set of experiments that employ Information Retrieval (IR) assessment criteria. The simulation findings demonstrated that the suggested approach performed significantly better than the most advanced models.

Anant Duhan and N. Arunachalam [15] introduced an RS framework based on decision trees. Real-time user data was employed to test the proposed solution. Based on the experimental findings, the suggested method might offer personalized recommendations for consumers. The CF method used in this work yielded positive outcomes in terms of accuracy and efficiency. The suggested strategy outperformed the existing methods.

Sunny Sharma et al. [16] proposed a book RS that predicts recommendations based on a hybrid method. There are three stages to the suggested approach, which combines content-based and CF. The first phase detects the objects, while the second phase matches profiles to identify individuals comparable to the current user. The last step makes suggestions to the intended user by evaluating the prediction value for each item utilizing the Resnick prediction equation. According to the simulation results, the recommended system outperformed the existing methods and performed effectively.

Tulasi Prasad Sariki and G. Bharadwaja Kumar [17] suggested an expanded architecture with 3 concurrent phases to enhance the recommendation procedure. The NER module extracted the named entities from the full book content, the essential semantic units that provide hints on potential reading choices for other related books. During the feature extraction phase, analysis of the book's front cover included identifying images, text, and captions to determine its genre. Stylometric analysis extended the feature set by evaluating the author's literary style, aiming to identify authors with similar writing styles to the current author of the book. According to the simulation findings, the suggested model outperformed the existing model by 6%. The stylometry and visual features improved the recommendation process compared to baseline and hybrid models.

Minyu Liu [18] offered deep belief networks for customized book suggestions. It examined library features in the first phase and sorted out changes in development along with digital library features. An investigation of existing deep-belief network-based personalized recommendation services yielded the parameters influencing library patrons' adoption of these services. Finally, deep belief networks are used to build digital library scenarios that are predicated on the development of a customized RS for library materials. According to the simulation findings, the proposed deep belief network-based RS produced excellent recommendation outputs.

Taushif Anwar and V. Uma [19] developed a Cross-Domain book RS employing Sequential Pattern Mining (CD-SPM) and rule mining. The proposed work utilized Wpath to support defining the semantic similarity between items that belong to diverse domains. The PrefixSpan technique is utilized to retrieve frequently occurring sequences. Initially, five distinct movie categories were used for the error comparison using the RMSE, and the findings showed that the suggested system produced a lower error rate. Finally, precision, recall, and F1 Score metrics were considered when evaluating the pattern mining method. According to the simulation results, the CD-SPM outperformed the CF-KNN technique.

Dhiman Sarma et al. [20] provided a book RS for online users who rated a book using the clustering method and then discovered a book's similarity to recommend a new one. The suggested system employed the K-means Cosine Distance function and the Cosine Similarity function to measure distance and evaluate the similarity between the book groups. Simulation findings demonstrated that the introduced system can more efficiently remove uninteresting novels from the list of recommended books.

Yihan Ma et al. [21] developed a personalized recommendation algorithm for book RSs by introducing wide and deep models. The first step is to gather reader and book information. Subsequently, the fundamental recommendation model is acquired through the combined training of LR and DNN networks, following an analysis of the wide and deep models' structures and concepts. A set of comparative studies were ultimately designed to authorize the effectiveness of the suggested approach. Based on the simulation results, the accuracy of the suggested book recommendation framework outperformed both hybrid and classic recommendation algorithms by a large margin.

Akhil M. Nair et al. [22] developed the Content-based Scientific Article Recommendation (C-SAR) model utilizing a deep learning methodology. The Apriori algorithm was utilized for association rule mining to identify frequently co-occurring sets of documents within a similar dataset. Subsequently, similarity levels between these documents were assessed using the Gated Recurrent Unit (GRU) technique. The experimental findings showed that the suggested approach surpassed existing models that used user representations and simple K-means clustering.

The current book, RS, faces several limitations that hinder its effectiveness. Firstly, they are restricted to recommending only ten books per query and are confined to a narrow domain, primarily focusing on computer science books. These systems require extensive real-time data, which raises privacy concerns as unnecessary information like emails, passwords, and user names is collected. Moreover, they are limited to library and e-library settings and fail to leverage clustering techniques for improved recommendations, making them unsuitable for broader applications such as e-commerce platforms. Additionally, the existing models only consider academic books, disregarding factors like book-length, borrowing duration, and clustering, leading to overfitting issues on small datasets, such as one containing just 4,612 books. To address these shortcomings, an enhanced book RS is necessary to provide more accurate, diverse, and scalable recommendations across various domains while ensuring privacy and effective data utilization.

3. Materials and Methods

The primary objective of this study is to present a comparative evaluation of suggested book RSs. In this proposed work, two different book RS have been introduced and provided a performance comparison of these two RSs. The WALS approach and NCF with feature optimization were utilized for effective RS. The block schematics of the Book RS are visualized in Figure 3.

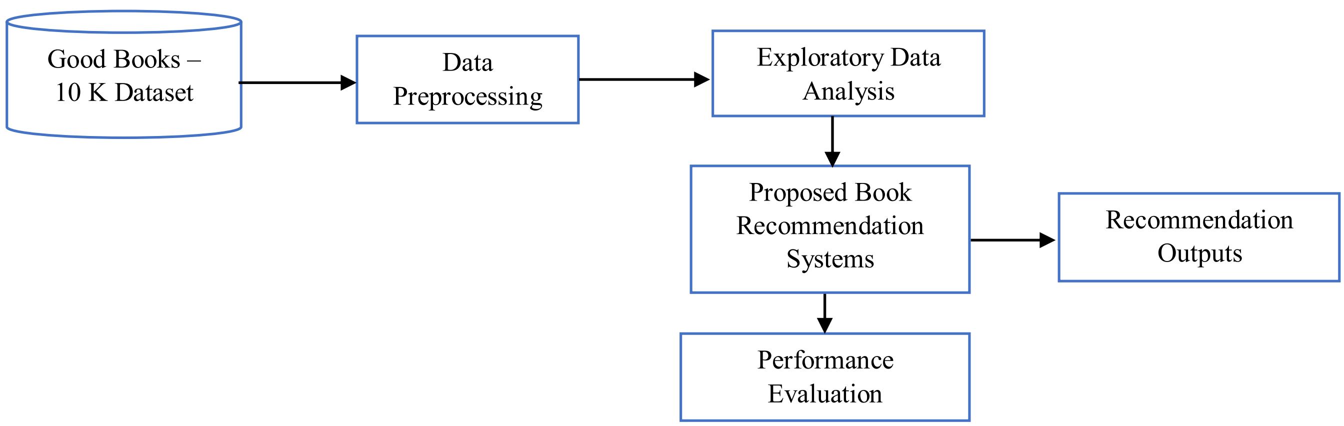


Fig. 3 Block diagram of proposed book RSs

3.1. Dataset Description

This study uses the goodbooks-10k [23] dataset, which includes ratings for ten thousand popular books. Each book averages 100 reviews, but some have fewer. A scale of one to five is employed for these ratings.

This dataset utilizes a sequential numbering scheme for the book IDs and user IDs, with book IDs ranging from 1 to 10000 and user IDs from 1 to 53424. It is fascinating that every user has at least two ratings, with an average of eight ratings per user.

Several folders categorize the dataset. Specifically, the proposed book RS utilized the "ratings.csv" file. Table 1 lists the primary characteristics of the dataset, while Figure 4 visually depicts its structure.

Table 1. Feature description of dataset

Features	Description
Book ID	Identification of the Number of Books
Title	Name of the Book
Authors	Author Name
Average_Rating	Average Rating of the Book
ISBN	International Standard Book Number
ISBN_13	13 Digits ISBN to Recognize the Book
Language_Code	Primary Language of the Book
Num_Pages	Page Count
Ratings_Count	Total Count of Ratings
Text_Reviews_Count	Total Count of Received Reviews
Publication_Date	Publication Date
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. 4 Visualization of dataset

3.2. Data Pre-Processing and Exploratory Data Analysis (EDA)

Data pre-processing is an essential and challenging stage in data analysis, aimed at undertaking problems like errors, missing values, outliers, and inconsistencies to improve data quality. Missing data can be greatly influencing the quality of data analysis. Two common techniques usually employed to handle missing data are imputation and data deletion. The management of missing values is crucial to data pre-processing because it has an immediate impact on dataset quality and reliability.

One can employ various methods to address missing data. These include eliminating rows that contain missing values, filling in the missing values with metrics, or utilizing more advanced techniques like regression imputation or predictive modeling. An essential part of data pre-processing is locating and handling outliers. Data points that substantially differ from the rest of the dataset are known as outliers, and their existence can have a big impact on the results of data analysis. These anomalies may affect the validity of statistical results, which can also impair learning model performance.

An ultimate step in the data analysis process offering a wide-ranging overview of all the significant features of a dataset is the aim of Exploratory Data Analysis (EDA). This process is necessary to get deeper perceptions into the data before using more complex statistical and learning frameworks.

Descriptive statistics are first used in the EDA process to give an overview of the main trends and data variability. In descriptive statistics, key metrics include the range, mean, mode, median and standard deviation. The descriptive statistics of the gathered data are shown in Figure 5, which offers an overview of these important data points.

	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. 5 Descriptive statistics of dataset

Providing visual representations of data is a powerful EDA technique called data visualization. This includes line charts, scatter plots, bar graphs, heat maps, etc. Figure 6 displays a visualization of the top 10 writers with the most published books. Figure 7 displays the book that appears the most frequently in the collected data.

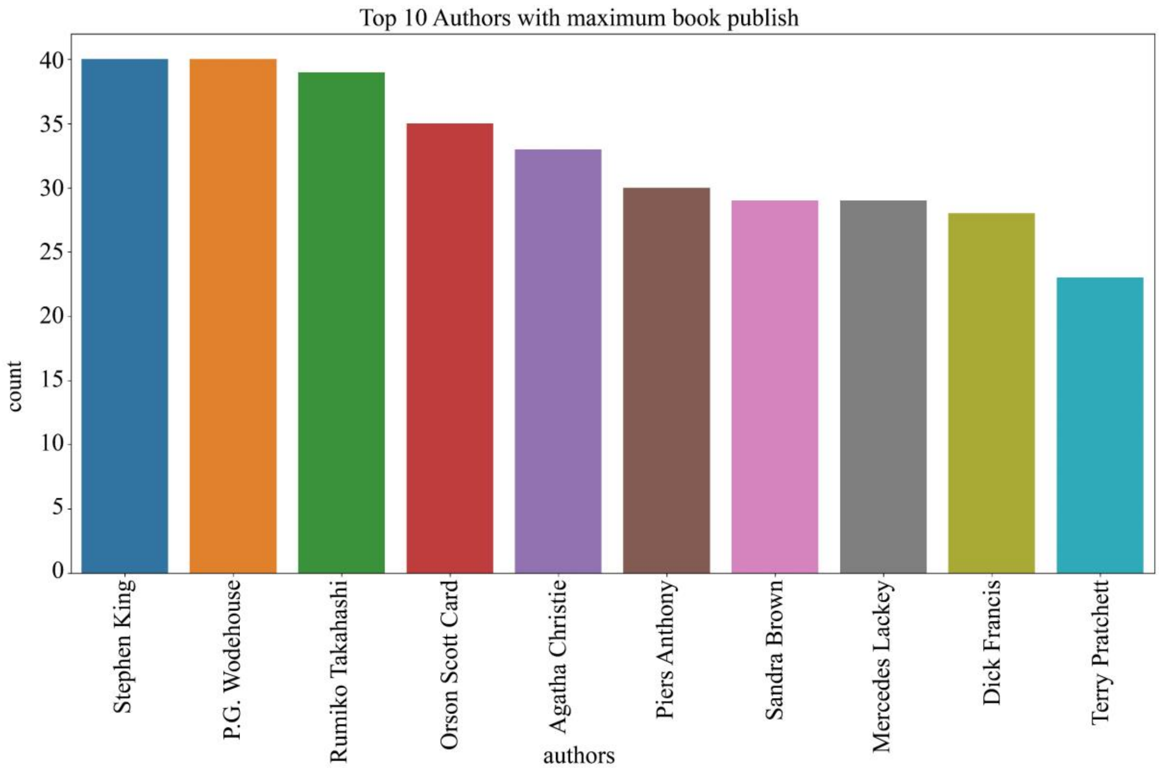


Fig. 6 Visualization of data

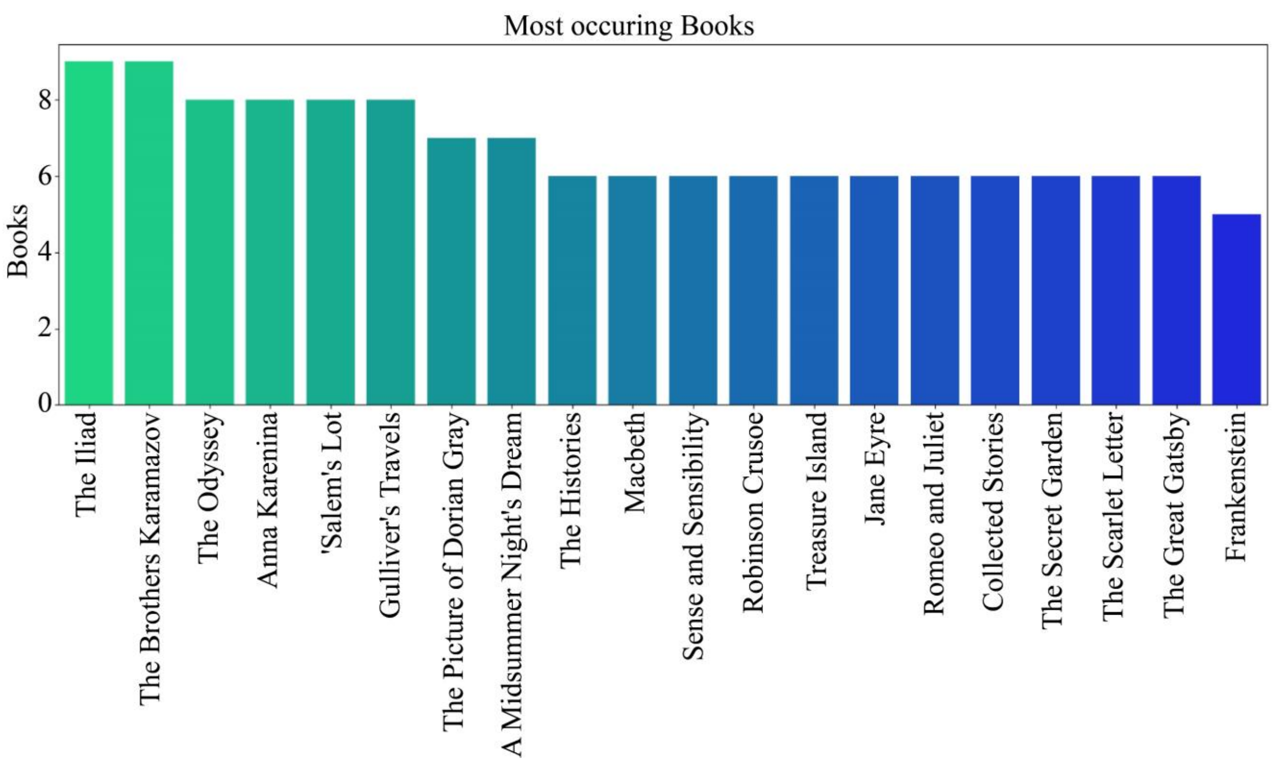


Fig. 7 Illustration of most occurring books in the dataset

Figure 8 displays the dataset's distribution plot of the rating variable. EDA requires the use of distribution charts, such as histograms. It provides enlightening details about the kind of data, its characteristics, and any potential trends or issues.

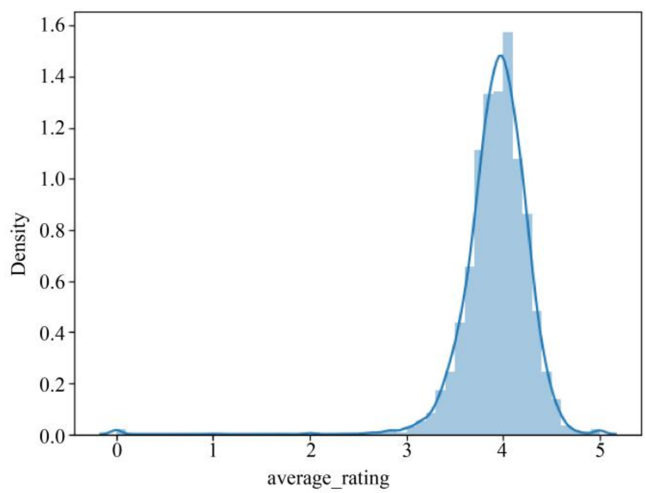


Fig. 8 Distribution plot of ratings from dataset

3.3. Effective Book RS Using Weighted Alternating Least Square (WALS) Approach

CF aims to deduce user preferences for items based on a large set of other user's preferences as well as the user's previous choices. The goal is to forecast users' unobserved preferences for things in a partially observed N × M user-item rating matrix Y, where there are N users and M items.

MF is an essential approach to complete the user-item rating matrix to anticipate the unobserved entries and simultaneously approximate the observable entries under some loss measure [24]. The objective is to determine the low-rank (or low-norm) latent components V for items and U for users. This can be approached in multiple ways. The entries in factor matrices U and V must not be negative to approximate Y using only additive combination factors. The basic concept is diminishing the squared sum distance between the related prediction and the observed entry by learning factor matrices U and V. The problem of optimization is expressed in Equation (1).

$$\min_{U\geq 0, V\geq 0} J(U,V) = \sum_{(i,j)\in\Omega}(y_{ij} - U_i V_j^T)^2 \qquad (1)$$

The most popular strategy for CF is a recommendation model based on the MF approach. MF uses Machine Learning (ML) techniques to construct a statistical model based on observed user rating data to provide recommendations. MF characterizes users and items using latent factors resulting from the observed sparse rating patterns. MF models the sparse user-item rating matrix $R_{m\times n}$ by mapping users (m) and items (n) into a subspace of latent variables of dimension k. This mapping aims to record the hidden qualities of both users and items and, hence, explain the observed rating patterns. The components of each item i are represented by a vector $p_i = R^k$, the values of which indicate how much of those factors the item i contains. Similarly, every user u is similarly represented by a vector $q_u = R^k$, where q_u quantifies how interested a user u is in each of the k latent components. The dot product of these two factor vectors is employed to estimate ratings:

$$r'_{ui} = q_u {p_i}^T \qquad (2)$$

In contrast to user- or item-based recommenders, the Alternating Least Squares (ALS) method [25] determines the best factor weights to reduce the least squares between the anticipated and actual ratings. This makes it possible to identify the latent variables that account for the observed users' item ratings. ALS gains knowledge of these parameters by lowering the observed reconstruction error of the ratings. Low-rank user (U) and product (V) variables are multiplied to create the rating matrix (R). These factors can then be multiplied to calculate the unknown ratings. Figure 9 displays the architecture of the ALS system.

A matrix R with n users and m items can be used to represent the rating data. In matrix R, the $(u,i)^{th}$ entry is r_{ui}, indicating that $(u,i)^{th}$ ratings for i^{th} item by u user. The R matrix is sparse because certain items have a low number of user ratings. As a result, the R matrix has the highest number of missing values. This sparse matrix problem has an MF solution. The term "factors" refers to the two k-dimensional vectors.

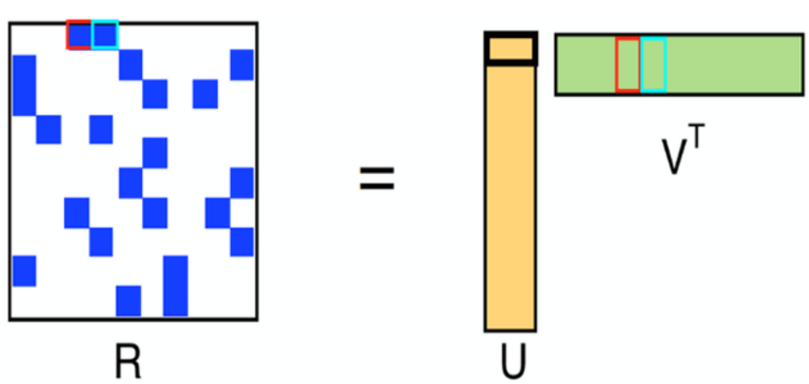


Fig. 9 ALS system architecture

- A k-dimensional vector called x_u summarizes each user u.
- A k-dimensional vector called y_i summarizes each item i.

$$r_{ui} \approx x_u^T Y_i \quad (3)$$

$$x_u = x_1, x_2, \ldots . x_n \in R^k \quad (4)$$

$$y_i = y_1, y_2, \ldots . y_n \in R^k \quad (5)$$

Equation (3) is formulated as an optimization problem to find:

$$argmin \sum_{r_{ui}} (r_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2) \quad (6)$$

This regularization factor, λ, is used to address the overfitting issue. When the set of variables x_u remains unchanged, the objective function for y_i is convex. Similarly, if the set of variables y_i stays constant, the objective function for x_u is also convex.

Consequently, by repeating the previously described process until convergence, the optimal values of x_u and y_i can be obtained. This is known as ALS.

In RSs, WALS is a popular MF algorithm. The factorization of the user-item rating matrix algorithm produces two matrices: an item-feature matrix and a user-feature matrix. The purpose of WALS is to multiply two matrices to estimate the user-item interaction matrix R:

$$R \approx U * V^T \quad (7)$$

R denotes the user-item interaction matrix, and $R(i, j)$ denotes the rating of user i for item j.

The goal of WALS is to factorize the user-item matrix R into two matrices, U and V so that $R \approx U * V^T$ can be approximated.

WALS optimizes the factorization using a weighted least squares method. The weight matrix W is introduced, where $W(i, j)$ represents the weight connected to the interaction between item j and user i. A number of factors, such as the quantity of interactions or the degree of confidence in the user's rating, can determine this weight. The objective function in WALS that requires minimization is the weighted least squares loss, as expressed below:

$$L(U, V) = \sum W(i, j) * (R(i, j) - U(i, :) * V(j, :)^T)^2 \quad (8)$$

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

WALS uses an alternate optimization technique to decrease the loss function. It substitutes between updating U and V while maintaining the other matrix fixed.

Updation of U: The update formula for $U(i, :)$ is specified for each user i.

$$U(i, :) = (\sum(W(i, j) * V(j, :)^T * V(j, :)))^{-1} * \sum(W(i, j) * R(i, j) * V(j, :)) \quad (9)$$

Updation of V: For each item j, the update rule for $V(j, :)$ is provided by:

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

The optimization process proceeds iteratively until certain criteria are met, such as hitting the maximum iteration limit or the loss function stabilizing with minimal change, ensuring convergence. Once the factorization is finalized, the predicted ratings are computed to provide user recommendations.

$$R_{pred}(i, j) = U(i, :) * V(j, :)^T \quad (11)$$

The objective function frequently includes regularization elements to prevent overfitting. Large values in V and U are impacted by these regularization parameters.

The algorithm weights each rating according to the user's confidence level. The algorithm typically gives larger weights to ratings with high levels of confidence and fewer weights to those with low levels. This enables the algorithm to prioritize trustworthy ratings and invalidate false ones. WALS is a potent algorithm that generates individualized suggestions based on ratings from user items. It can generate more accurate and dependable recommendations than regular ALS by including confidence levels in the optimization process.

The proposed WALS book RS is now operational on the Google Collaboratory platform. Cloud-based Google Collaboratory provides free access to virtual environments with many resources, such as a 16GB Nvidia K80 GPU. Python is a well-liked and flexible programming language that is supported by Colab. Pre-installed libraries like TensorFlow and Keras are also included. The major parameters utilized by the proposed WALS model are the Rank of the factorized matrix (k), Number of iterations and Regularization parameters (λ).

3.4. Book Recommendation System Using Neural Collaborative Filtering (NCF) with ANOVA Optimization

Data is initially gathered from pertinent sources during this phase to ensure a diverse and representative collection. The next step involves preparing the data, which includes cleaning, normalization, and formatting the data into an analysis-ready format. The next step involves using Exploratory Data Analysis (EDA), which helps identify key features by spotting patterns, trends, and insights in the dataset. The dataset is then separated into training, validation, and test sets to enable reliable model evaluation. ANOVA is used in feature optimization to identify the most important features that affect the recommendation process. The recommended Neural Collaborative Filtering (NCF) with feature optimization is the central component of the technique; it incorporates a CNN to recognize complex patterns and relationships in the data, improving the recommendation performance. Finally, the effectiveness and dependability of the book recommendation system are confirmed by evaluating the RMSE values. The detailed block diagram of the book recommendation system using NCF with ANOVA optimization is depicted in Figure 10.

A statistical technique called Analysis of Variance (ANOVA) is used for feature optimization to identify the most important features in a predictive modeling scenario, especially when there are a lot of input variables. It aids in selecting a subset of features that have the most vital effect on the model's overall performance [26]. The statistical technique known as ANOVA is employed to see if the averages of various groups differ in any significant way. This is accomplished by comparing the degree of variability within each group to the variability between these group averages. It indicates that these groups have significant mean differences if the variability between them is much larger than the variability within them. When an ANOVA is used, the variability within and between groups is compared to determine the F-statistic. If the F-statistic is higher than a predefined critical value, then there are significant differences between the group means. ANOVA is a hypothesis-testing method that splits a dataset's total variance into two main parts: the variance resulting from group differences and the variance originating from group differences. ANOVA serves as an effective tool for feature optimization, enabling the determination of significant differences in the target variable's average values across various categories or levels associated with a particular feature. Figure 11 shows the cause of variation in ANOVA optimization.

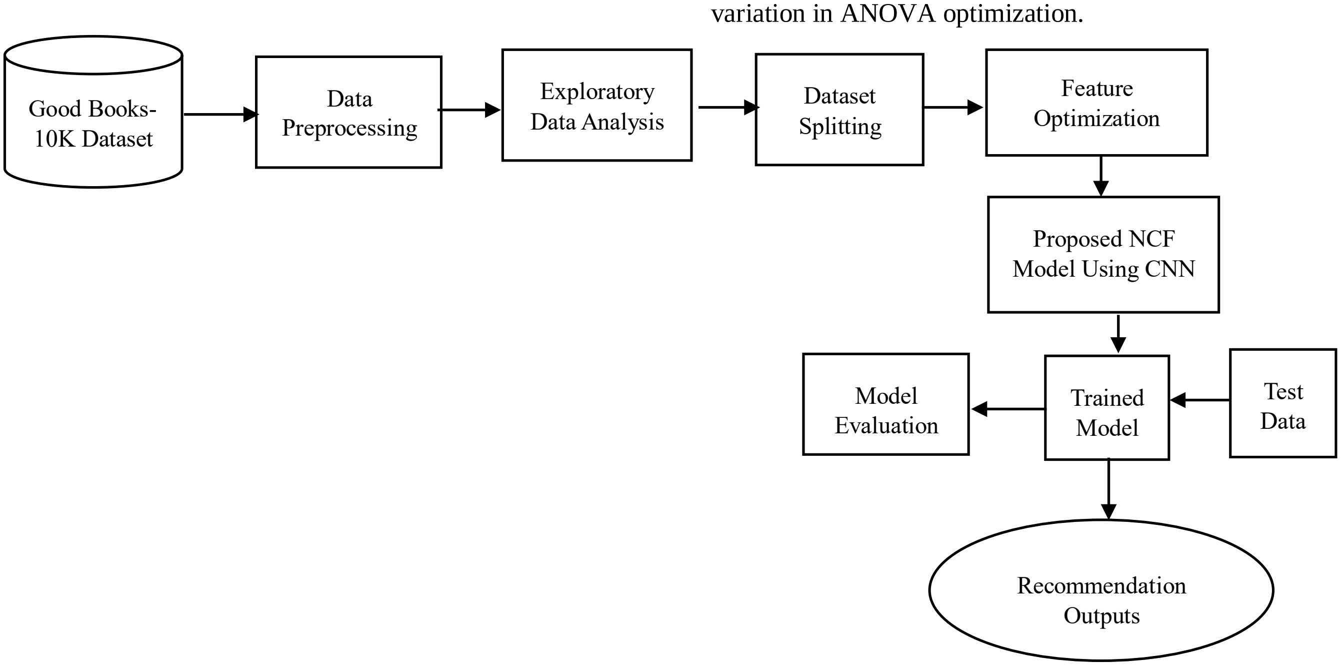


Fig. 10 Proposed book recommendation system using NCF with ANOVA optimization

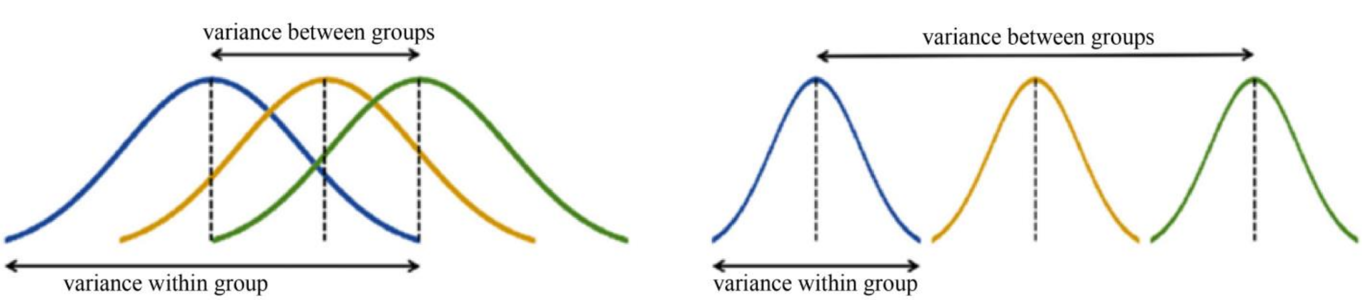


Fig. 11 Source of variation

ANOVA comprises testing two hypotheses:

3.4.1. Null Hypothesis (H_0)

The premise of this hypothesis is that the means of the target variable in each of the feature's categories or levels under analysis do not differ significantly from one another.

3.4.2. Alternative Hypothesis (H_a)

According to this hypothesis, there may be statistically significant variations in the target variable's means between the different feature categories or levels.

An F-statistic, or the ratio of the variation within groups to the variance between groups, is computed using an ANOVA. An elevated F-statistic suggests that the groups differ significantly, which suggests that the feature is a strong contender for feature optimization. A p-value, which measures the likelihood of finding such substantial differences by pure chance, is computed using the F-statistic. The null hypothesis (H_0) should be rejected if the p-value is less than the selected crucial level, indicating that the characteristic significantly affects the target variable. A subset of the most pertinent features should be chosen for the predictive model based on the ranking.

ANOVA feature optimization is a useful method for determining which characteristics in a dataset are most pertinent by evaluating their influence on a target variable. It ensures statistical rigor in the feature selection process and aids in constructing more effective and understandable predictive models.

NCF for book recommendation builds upon conventional MF techniques [27] due to its capacity to record rich user-book interactions using neural networks. As a result, NCF learns intricate, nonlinear correlations by multiplying users and books into high-dimensional vector spaces and combining them using neural layers. This technique also yields more accurate recommendations because the model can learn higher-order feature interactions that linear models could miss [28]. Consequently, the model often enhances RSs by striving to forecast user preferences precisely.

The basic idea behind the NCF framework is to replace the MF (or dot product) function with a neural network, which can learn and approximate any data distribution. The primary components of the NCF model are a neural network and a Generalized Matrix Factorization (GMF). Figure 12 illustrates the general framework of the NCF model.

A variety of user and item modeling, including context-aware, content-based, and neighbor-based modeling, can be supported by modifying the two feature vectors (v_u^U and v_i^I) that make up the bottom input layer. These vectors denote user u and item i, respectively.

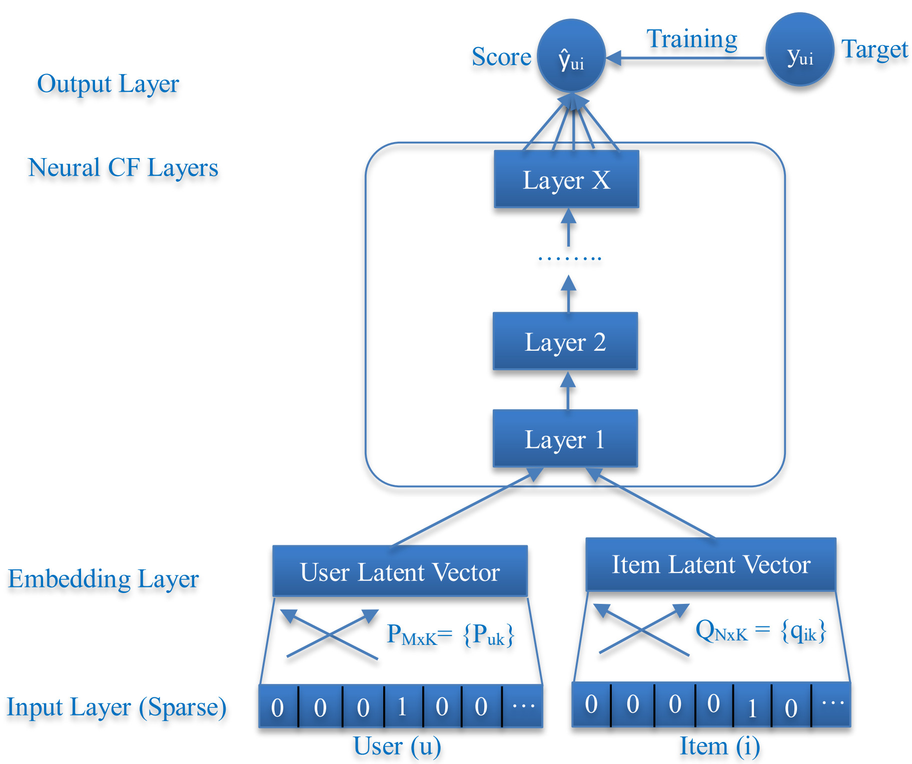


Fig. 12 General framework of NCF model

In the case of the latent factor approach, the embedding layer is a fully connected layer situated between the input layer and converts sparse representations to dense vectors. The final vectors obtained for the users and items are the representations of the latent vectors. These latent vectors are fed into a multi-layer neural network called the Neural Collaborative Filtering (NCF) layers, which can be customized in each layer to learn specific latent features of the user-item interactions. The final hidden layer, also known as the output layer X, governs the framework's ability. The predicted score y_{ui} is produced by the output layer, and training is done in order to diminish the pointwise loss between the predicted score $\hat{y}_{ui}$and the actual target value of the corresponding variable.

The NCF model can be expressed as,

$$\hat{y}_{ui} = f(P^T.V_u^U, Q^T.V_i^I|P, Q, \ominus_f) \tag{12}$$

Where $P \in R^{M\times K}$ and $Q \in R^{N\times K}$, representing the latent factor matrix for users and items, respectively; and $\ominus_f$ represents the model parameters of the interaction function f. Since the function f is expressed as a multi-layer neural network, it can be expressed as

$$f(P^T.V_u^U, Q^T.V_i^I) = \phi_{out}(\phi_X\left(\ldots.\phi_2\left(\phi_1(P^T.V_u^U, Q^T.V_i^I)\right)\ldots\right)) \tag{13}$$

Where ϕ_{out} and ϕ_X respectively implies the mapping function for the output layer and x^{th} NCF layer and there are X neural CF layers in total.

In particular, the recommendation framework uses the given NCF model with respect to user and item (book) information. First, a unique embedding layer with a size of 50 must be defined for both users and books. Once the system converts the categorical data into dense vector forms and reorganizes it, it passes through embedding layers that merge the two features. The system combines these vectors, creating one vector representing the user and another representing the book. Next, a few dense layers with ReLU activation functions receive the concatenated vector, separating important interactions for users and books from each dense layer with ReLU activation. The output layer produces a single score, which could be either the expected rating or the preferred score. Table 2 contains a tabulation of the hyperparameters used by the proposed NCF model. The Python programming language implements the suggested NCF-based book RS on Google Collaboratory.

Table 2. Hyperparameters utilized by the NCF model

Parameters	Value
Optimizer	Adam
Number of Epochs	10
Activation Function	Rectified Linear Unit
Metrics	Root Mean Squared Error (RMSE)
Loss Function	Mean Square Error (MSE)
Batch Size	64

4. Results and Discussion

The effectiveness of the recommended RSs was measured utilizing the RMSE values. It computes the mean difference between the actual values and the values predicted by a model. It provides an approximation of the accuracy or how well the model predicts the intended outcome. A model is considered better when its RMSE value is lower. A perfect model, or hypothetical model, would have an RMSE value of 0. Equation (14) is utilized to express the RMSE value.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|y_i - \hat{y}_i\|^2} \tag{14}$$

When the i^{th} measurement is denoted by y_i, the number of data points is N, and the related prediction is $\hat{y}_i$.

When the predicted and actual numbers are exactly the same, the result is 0. Low RMSE values show that the model has more accurate predictions and matches the data well. Higher levels, on the other hand, indicate greater error and less accurate forecasts. Table 3 lists the effectiveness of the suggested WALS-based book RS.

Table 3. Performance evaluation of book recommendation system using WALS approach

K	Number of Iterations	λ	Training Time	RMSE	Testing Time
5	10	0.1	225.670	3.880	0.00168
10	10	0.1	413.3266	3.833	0.00176
20	10	0.1	832. 388	3.77	0.001498
5	10	0.01	227.261	3.879	0.001847
5	10	0.001	221.310	3.878	0.001836
5	20	0.001	448.85	3.8787	0.001432
5	25	0.001	567.545	3.878	0.00155
5	30	0.001	686.3248	3.8795	0.001526

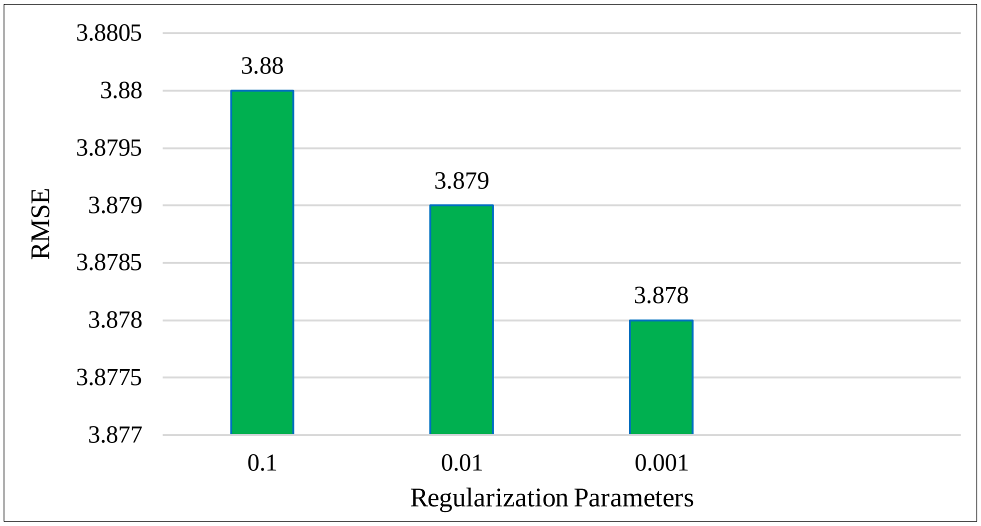
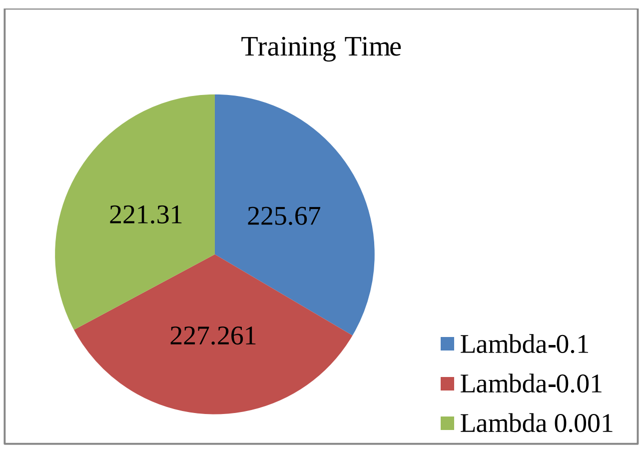
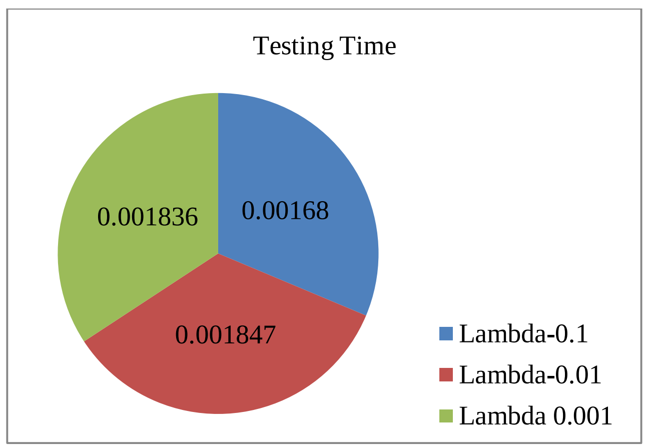


Fig. 13 RMSE under K=5 and no. of iterations=10

The number of iterations and the selection of hyperparameters, especially λ, have a significant impact on the suggested book RS outcomes. The RMSE falls with increasing iterations; if set at 0.1, it reaches a minimum after 20 iterations. But with a lesser value of 0.01, the RMSE just marginally increases, suggesting a possible preference for a more conservative regularization. When the number of iterations and λ value increase throughout the training and testing periods, there is also a conflict between efficiency and model correctness. Regarding RMSE and computation cost, it seems that λ = 0.01, and 20 iterations produce the best results. A graphical depiction of RMSE values under K = 5 and 10 iterations can be seen in Figure 13. The training and testing time distributions of the suggested WALS-based book RS are displayed in Figure 14.



(a)



(b)

Fig. 14 (a) Distribution of training time, and (b) Distribution of testing time (Under K=5, number of iterations=10).

The performance of the proposed book RS using NCF is tabulated in Table 4. A strong correlation has been seen between the number of epochs, training time, and the RMSE in the data provided for the proposed book RS using NCF. With increased training, the RMSE constantly drops from 1.1520 to 0.9989 as the number of epochs increases from 5 to 20. This, however, results in a much longer training time 222.027 seconds for 5 epochs rises to 622.041 seconds for 20 epochs. With a minor rise from 14.7194 seconds at 5 epochs to about 20.68 seconds at 20 epochs, the testing time is still comparatively constant.

Higher epochs, which typically require more computer power and longer training cycles, improve model performance. The training time, however, increases significantly, from 222.027 seconds for 5 epochs to 622.041 seconds for 20 epochs. The testing time remains relatively constant, slightly increasing from 14.7194 seconds at 5 epochs to roughly 20.68 seconds at 20 epochs. Model performance generally improves with longer training phases and higher epochs requiring more processing power. The loss plot of the proposed NCF-based book recommendation system is depicted in Figure 15.

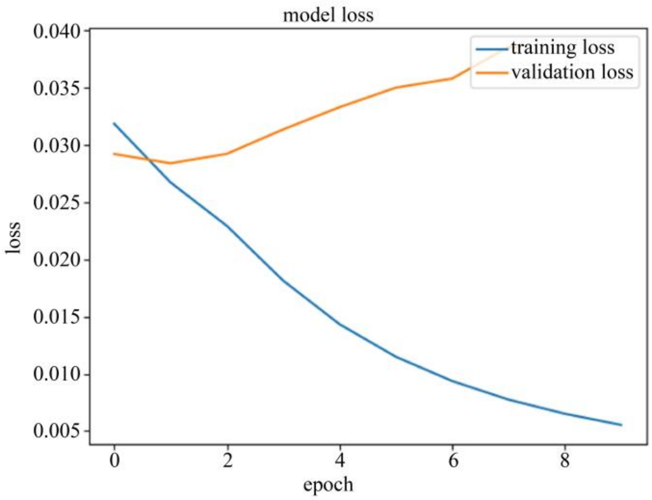


Fig. 15 Loss plot of the proposed book recommendation system using NCF

The performance comparison of the proposed book RS was tabulated in Table 5. The comparison of the NCF and WALS approach shows different performance characteristics regarding testing time, training time, and RMSE for a book RS. Overall epochs, WALS shows much higher training times than NCF, ranging from 413.3266 to 832.388 seconds, while NCF shows training times between 222.027 and 622.041 seconds. WALS, on the other hand, has testing times that are significantly shorter (between 0.0014 and 0.0018 seconds) than NCF (between 14.7194 and 20.68 seconds). NCF exhibits superior recommendation performance regarding RMSE, outperforming WALS consistently with lower values across all epochs. Regarding RMSE, Figure 16 offers a graphical depiction of the performance comparison of the suggested book RS.

Table 4. Performance of proposed book recommendation system using NCF

Epochs	Batch Size	Training Time	Testing Time	RMSE
5	64	222.027	14.7194	1.1520
10	64	309.48	20.5226	1.0845
20	64	622.041	20.68	0.9989

Table 5. Performance comparison of proposed book recommendation systems

Proposed Models	Number of Epochs	Batch Size	Training Time	Testing Time	RMSE
WALS	5	64	448.85	0.001432	3.8787
	10	64	413.3266	0.001776	3.833
	20	64	832.388	0.001498	3.77
NCF	5	64	222.027	14.7194	1.1520
	10	64	309.48	20.5226	1.0845
	20	64	622.041	20.68	0.9989

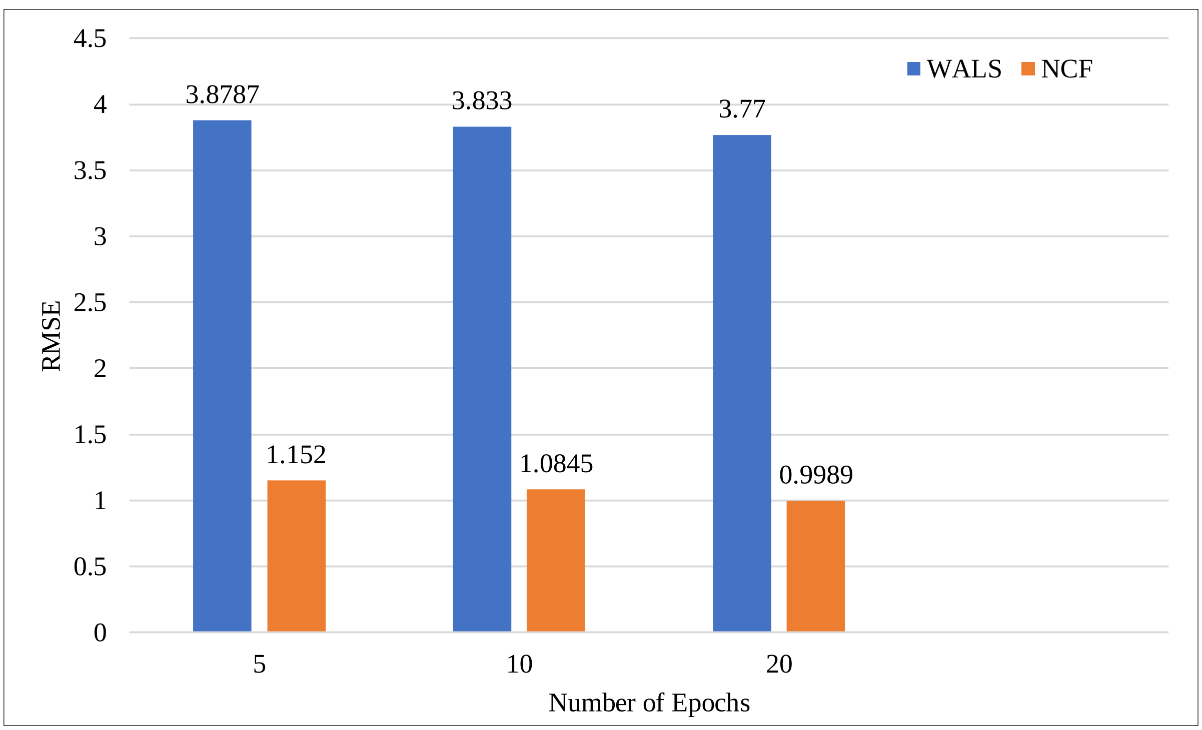


Fig. 16 Performance comparison of proposed book RSs

5. Conclusion

Book RSs have become essential to improving the reading experiences of people in the current digital world. These systems generate personalized book recommendations using sophisticated algorithms, user data, and content analysis. This allows readers to find new books that match their tastes and preferences. This paper compared the Neural Collaborative Filtering (NCF) with feature optimization and Weighted Alternating Least Squares (WALS) methods for book RSs. A performance evaluation based on the RMSE value was conducted on the GoodBooks-10K dataset to test the proposed models. A lower RMSE value indicates a system that can better predict user behavior and deliver a more tailored and enjoyable reading experience. The simulation results showed that the RS yielded remarkable outcomes, including a significant reduction in RMSE values. The NCF technique surpassed WALS in recommendations by consistently generating lower RMSE values. This outcome demonstrated how well the recommended strategies improve book recommendation performance and help users select books that are more compatible with their personal preferences. In the future, further enhancements can be made by incorporating advanced deep learning models and natural language processing techniques to improve the understanding of user preferences and the semantic relationships between books.

Acknowledgements

The author expresses profound appreciation to the supervisor for providing guidance and unwavering support throughout the course of this study.

References

[1] Keunho Choi et al., “A Hybrid Online-Product Recommendation System: Combining Implicit Rating-Based Collaborative Filtering and Sequential Pattern Analysis,” *Electronic Commerce Research and Applications*, vol. 11, no. 4, pp. 309-317, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[2] Seok Kee Lee, Yoon Ho Cho, and Soung Hie Kim, “Collaborative Filtering with Ordinal Scale-Based Implicit Ratings for Mobile Music Recommendations,” *Information Sciences*, vol. 180, no. 11, pp. 2142-2155, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[3] Edward Rolando Núñez-Valdéz et al., “Implicit Feedback Techniques on Recommender Systems Applied to Electronic Books,” *Computers in Human Behavior*, vol. 28, no. 4, pp. 1186-1193, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[4] Amy J.C. Trappey et al., “Intelligent Patent Recommendation System for Innovative Design Collaboration,” *Journal of Network and Computer Applications*, vol. 36, no. 6, pp. 1441-1450, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[5] Yingjie Wang et al., “A Trust-Based Probabilistic Recommendation Model for Social Networks,” *Journal of Network and Computer Applications*, vol. 55, pp. 59-67, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[6] Liang Zhu et al., “SEM-PPA: A Semantical Pattern and Preference-Aware Service Mining Method for Personalized Point of Interest Recommendation,” *Journal of Network and Computer Applications*, vol. 82, pp. 35-46, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[7] Fariba Aznoli, and Nima Jafari Navimipour, “Cloud Services Recommendation: Reviewing the Recent Advances and Suggesting the Future Research Directions,” *Journal of Network and Computer Applications*, vol. 77, pp. 73-86, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[8] Da Cao et al., “Cross-Platform App Recommendation by Jointly Modeling Ratings and Texts,” *ACM Transactions on Information Systems*, vol. 35, no. 4, pp. 1-27, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[9] Johan Bollen et al., “Usage Derived Recommendations for a Video Digital Library,” *Journal of Network and Computer Applications*, vol. 30, no. 3, pp. 1059-1083, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[10] Michael J. Pazzani, and Daniel Billsus, *Content-Based Recommendation Systems*, The Adaptive Web. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol. 4321, pp. 325-341, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[11] Mohammed F. Alhamid et al., “RecAm: A Collaborative Context-Aware Framework for Multimedia Recommendations in an Ambient Intelligence Environment,” *Multimedia Systems*, vol. 22, no. 5, pp. 587-601, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[12] F.O. Isinkaye, Y.O. Folajimi, and B.A. Ojokoh, “Recommendation Systems: Principles, Methods and Evaluation,” *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261-273, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[13] Yassine Afoudi et al., “Hybrid Recommendation System Combined Content-Based Filtering and Collaborative Prediction Using Artificial Neural Network,” *Simulation Modelling Practice and Theory*, vol. 113, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] Fikadu Wayesa et al., “Pattern-Based Hybrid Book Recommendation System Using Semantic Relationships,” *Scientific Reports*, vol. 13, no. 1, pp. 1-12, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Anant Duhan, and N. Arunachalam, “Book Recommendation System Using Machine Learning,” *AIP Conference Proceedings*, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Sunny Sharma, Vijay Rana, and Manisha Malhotra, “Automatic Recommendation System Based on Hybrid Filtering Algorithm,” *Education and Information Technologies*, vol. 27, no. 2, pp. 1523-1538, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Tulasi Prasad Sariki, and G. Bharadwaja Kumar, "An Aggrandized Framework for Enriching Book Recommendation System," *Malaysian Journal of Computer Science*, vol. 35, no. 2, pp. 111-127, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Minyu Liu, "Personalized Recommendation System Design for Library Resources through Deep Belief Networks," *Mobile Information Systems*, vol. 2022, no. 1, pp. 1-9, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Taushif Anwar, and V. Uma, "CD-SPM: Cross-Domain Book Recommendation Using Sequential Pattern Mining and Rule Mining," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 3, pp. 793-800, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Dhiman Sarma, Tanni Mittra, and Mohammad Shahadat Hossain, "Personalized Book Recommendation System Using Machine Learning Algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 1, pp. 1-8, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] Yihan Ma et al., "Book Recommendation Model Based on Wide and Deep Model," *2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID)*, Guangzhou, China, pp. 247-254, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[22] Akhil M. Nair, Oshin Benny, and Jossy George, "Content Based Scientific Article Recommendation System Using Deep Learning Technique," *Inventive Systems and Control: Proceedings of ICISC 2021*, pp. 965-977, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] Kaggle, Goodbooks-10k, Ten Thousand Books, One Million Ratings. Also Books Marked to Read, and Tags, 2017. [Online]. Available: https://www.kaggle.com/datasets/zygmunt/goodbooks-10k

[24] Behnoush Abdollahi, and Olfa Nasraoui, "Explainable Matrix Factorization for Collaborative Filtering," *WWW '16 Companion: Proceedings of the 25th International Conference Companion on World Wide Web*, pp. 5-6, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[25] Gábor Takács, and Domonkos Tikk, "Alternating Least Squares for Personalized Ranking," *RecSys '12: Proceedings of the Sixth ACM Conference on Recommender Systems*, pp. 83-90, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[26] Tae Kyun Kim, "Understanding One-Way ANOVA using Conceptual Figures," *Korean Journal of Anesthesiology*, vol. 70, no. 1, pp. 22-26, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[27] Xiangnan He et al., "Neural Collaborative Filtering," *WWW '17: Proceedings of the 26th International Conference on World Wide Web*, pp. 173-182, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[28] Steffen Rendle et al., "Neural Collaborative Filtering vs. Matrix Factorization Revisited," *RecSys '20: Proceedings of the 14th ACM Conference on Recommender Systems*, pp. 240-248, 2020. [CrossRef] [Google Scholar] [Publisher Link]