1.5em 0pt

Algorithmic Exploration in Reading Behavior Analysis and Recommendations Using Machine Learning

Mahatir Ahmed Tusher[a], Saket Choudary Kongara [b], Gangavarapu Sreeram [c], Srinivas Arukonda [d] and Sheikha Farook Batha [e]

[a] *VIT AP UNIVERSITY, AMARAVATHI, Vijayawada, 522241, Andhra Pradesh, India*

ARTICLE INFO

Keywords:
Book Recommendations
Machine Learning
Cosine Similarity
Singular Value Decomposition (SVD)
Hybrid Approach

ABSTRACT

In this study, we look into book recommendation algorithms, which are an important aspect of modern online retail and e-commerce systems. Most of the research on book recommendation systems has relied on cosine similarity or K-Nearest Neighbors (KNN). Nevertheless, these approaches frequently fail to deliver concise and relevant recommendations. Here, we have combined cosine similarities with Singular Value Decomposition (SVD) using content-based and collaborative filtering strategies to solve this issue. Cosine similarity is an effective strategy to recognize associated titles based on user ratings, however it neglects to account for all factors impacting user choices. In order to overcome this, we apply SVD matrix factorization, which uncovers underlying variables that influence ratings. Our recommender system seeks to provide recommendations that are more relevant by combining these methods. To evaluate our performance, we employ measures like Mean absolute error (MAE), recall, precision, Root Mean Squared Error (RMSE), and Normalized Mean Absolute Error (NMAE). With a precision score of 0.80, about 80% of our top-k recommendations are relevant, demonstrating the effectiveness of our system. Our results highlight the possibility of hybrid algorithms for more accurate recommendations, as they show that SVD and cosine similarity together outperforms systems depending only on cosine similarity.

1. Introduction

The problem of information overload has emerged in the age of digital transformation due to the amount of data. This is particularly evident in the realm of book recommendations, where users are often overwhelmed by the vast array of choices available.The challenge at hand is to offer consumers customized and personal recommendations for books so they may improve their reading experience and promote a reading culture. A significant number of recent studies on book recommendation systems has been concentrated on collaborative filtering approaches. By examining how comparable users have rated products in the past, collaborative filtering makes recommendations for users. Among the widely used collaborative filtering methods are K-Nearest Neighbors (KNN) and cosine similarity [28] [29]. These methods, although effective to a certain extent, have their limitations. For instance, cosine similarity and KNN often struggle with issues of data sparsity and scalability [28].

With the help of Singular Value Decomposition (SVD), we provide in this study a unique method for book recommendation systems that combines the advantages of matrix factorization and cosine similarity. SVD factors a matrix by representing it as a product of many smaller matrices. Based on the learnt latent components, it can estimate missing values, which has shown to be very useful in addressing the sparsity of user ratings data. By integrating cosine similarity and SVD, we aim to provide more relevant and precise book recommendations. To help explain our distinctive procedure, we have included Figure 1, which depicts the general structure of the proposed approach.

Our model stands out for the way it is able to effectively handle both item similarity and user choice. Through the use of cosine similarity, we are able to identify the similarities between different books, which allows for more sophisticated suggestions. Meanwhile, Singular Value Decomposition (SVD) integration enables us to reveal latent components that capture abstract concepts that impact a user's book evaluation. This dual approach improves suggestion accuracy and

*Corresponding author
mahatirahmedtusher123@gmail.com (M.A. Tusher); saketchoudary.3011@gmail.com (S.C.K.); sree00gvp@gmail.com (G.S.); srinivas.a@vitap.ac.in (S.A.); sheikhafarook@gmail.com (S.F.B.)
ORCID(s):

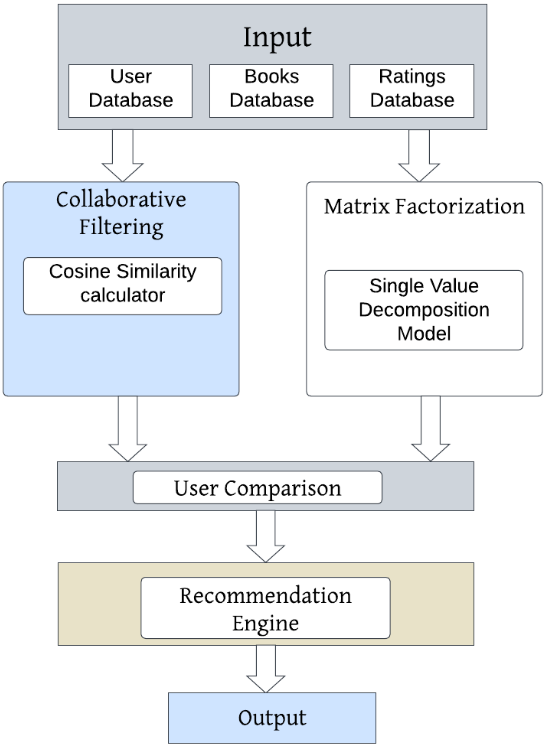

Figure 1: An overview of the framework for the Hybrid Book Recommendation System integrating Cosine Similarity alongside SVD.

provides additional insight into user preferences and the characteristics of the books being evaluated. Essentially, our approach provides an elegant framework that effectively strikes a compromise between user desire and item similarity, improving the recommendation process as a whole. Our model stands out for the way it is able to effectively handle both item similarity and user choice. Through the use of cosine similarity, we are able to identify the similarities between different books, which allows for more sophisticated suggestions. Meanwhile, Singular Value Decomposition (SVD) integration enables us to reveal latent components that capture abstract concepts that impact a user's book evaluation.This two-pronged technique enhances the accuracy of the recommendations while also offering an expanded awareness of the user's preferences and the inherent attributes of the books under scrutiny. Essentially, our approach provides an elegant framework that effectively strikes a compromise between user desire and item similarity, improving the recommendation process as a whole.

Here are the key contributions of our research:

1. Developed a robust book recommendation system combining cosine similarity and Singular Value Decomposition (SVD).
2. Effectively addressed data sparsity, a common challenge in recommendation systems, by leveraging SVD.
3. Designed to be scalable, capable of handling large datasets while maintaining accurate recommendations.
4. Provides highly tailored book recommendations based on both item resemblance and individual preferences.
5. Performed an extensive scrutiny of older techniques, indicating superior effectiveness.
6. Integrating cosine similarity and SVD led to significant advancements in book recommendation systems, ultimately enhancing user reading experiences and fostering a culture of reading.

The study will explore the details of our model, talk about how it is implemented, and provide a comparison with conventional approaches. Our research will make significant contributions to the field of book recommendation systems, paving pathways for greater accuracy and customization.

1.1. The motivation and purpose of this study

The motivation behind this study is the issue of information overload in the age of digital transformation and the amount of data that comes with it. It's common for customers to feel swamped with the sheer quantity of book recommendation options accessible to them. Providing customers with tailored and individualised book recommendations to improve their reading experience and encourage a culture of reading is the current issue. Below are the main reasons for conducting this study:

1. **Handle Information Overload:** In the digital age, consumers are usually overwhelmed by the vast amount of book recommendations. The study offers tailored book choices in an attempt to alleviate this information overload.
2. **Improve Recommendation Accuracy:** Scalability and data sparsity problems frequently plague existing techniques like cosine similarity and KNN. This work aims to improve suggestion accuracy by integrating Singular Value Decomposition (SVD) alongside cosine similarity.
3. **Capture Latent factors:** The latent factors underlying user evaluations can be effectively captured by SVD. These latent factors provide a more profound understanding of user preferences since they symbolize fundamental properties of both the user and the item.
4. **Handle User Choice and Item Similarity:** The model is notable for its capacity to manage user choice and item similarity in an efficient manner, enhancing the recommendation process overall.
5. **Contribute to the area:** By improving users' reading experiences and encouraging a culture of reading, the research hopes to significantly advance the area of book recommendation systems.
6. **Open the Door for Future Improvements:** The study establishes the foundation for subsequent advancements in the customization and precision of book recommendation algorithms' selections.

Section 2 offers a comprehensive overview of the literature with a focus on the advantages and disadvantages of the existing methods. Section 3 provides a detailed description of the technique of the proposed method, including data preprocessing. In Section 4, we perform an experimental analysis by contrasting the assessment metrics of our model with the most recent methods. Future scope and limitations were discussed in Section 6 after the discussion in Section 5 is covered. The paper is concluded in Section 7.

2. Related work and Literature survey

The industry of book recommendation systems has seen an extensive number of novel approaches aiming at improving user experiences and responding to distinct interests. Mathew et al. proposed a very effective book recommendation system (BRS) that included content-based filtering, collaborative filtering, and mining association rules [1]. By customizing recommendations based on customers' interests, this hybrid model increases user satisfaction and boosts revenue for online book sellers. An opinion mining-based method is introduced by Sohail et al. [2] that sorts through user reviews, weights features, and ranks books in different computer science fields. This approach helps consumers find highly rated publications that are relevant to their interests quickly and easily. A hybrid recommender system with temporal features and demographic factors integrated by Kanetkar et al. [3] goes beyond personalizing. This method improves customer utility and pleasure by providing personalized recommendations based on variables like location, gender, and age.

Customized recommendation systems are explored by Sarma et al. [4] through the use of machine learning techniques.Their method provides more precise and targeted suggestions by employing the K-means Cosine Distance function to categorize books and the Cosine Similarity function to detect commonalities. In contrast, a comprehensive four-level recommendation system is presented by Mounika et al. [5]. Their methodology includes sentiment analysis, collaborative filtering, K-nearest neighbor algorithm, semantic network grouping, and other techniques to achieve unparalleled accuracy in the least period of time. Sivaramakrishnan et al. [6] explored neighborhood-based collaborative filtering techniques and used several similarity measures to enhance recommendation accuracy. Their strategy, based on local algorithms, seeks to improve users' perceptions of the relevancy of book recommendations. Zhou et al. [7] describe an innovative information recommendation system utilizing an enhanced Apriori algorithm. This system offers effective book suggestions with little computational overhead by getting around performance issues with conventional approaches.

Chen et al. [8] addresses the integration of mobile technology in library settings. Problem-based learning in actual library settings is made easier by their Intelligent Mobile Location-Aware Book Recommendation System (IMLBRS). By means of guidance and suggestion features based on maps, students are able to efficiently navigate the library space and identify pertinent resources. Kuroiwa et al. [9] present a dynamic Book Utilization System (BUS) that makes use of virtual library improvements and web services. By improving tailored suggestions and enabling user sharing, this system encourages book aficionados to work together as a community.

Jia et al. [10] concentrated their investigation into the usage of recommendation systems in library management on content and collaborative filtering procedures. This paper covers the architecture and functionalities of a recommendation system customized for library contexts with the goal of improving information retrieval efficacy. A hybrid recommender system based on characteristics and personality is put out by Hariadi et al. [11]. Their technology generates personalized book suggestions by incorporating user personality aspects to the recommendation process in addition to attribute-based methods. Rahutomo et al. [12] look on how online businesses provide book recommendations. Their study emphasizes the value of user-generated evaluations in cooperative filtering techniques and the part embedding models play in improving suggestion accuracy. By combining these several techniques, book recommendation systems can be made more adaptable and accommodate a greater variety of user preferences and demands. To solve the issue of an excessive amount of product selection on e-commerce platforms, Sharma et al. [13] developed a recommendation model which is hybrid that combines collaborative-based filtering with content-based filtering. A hybrid system for recommendation based on patterns and using semantic links to enhance book recommendations was presented by Wayesa et al. [14]. In order to enhance top-N suggestions, Xin et al. [15] concentrate on community detection algorithms, especially in academic library contexts.

Mariana et al. [16] use association rule mining to create a powerful online public access catalog (OPAC) recommendation system. In order to successfully handle sparse data, Desai et al. [17] describe an enterprise-friendly recommendation system based on biclustering.In order to generate personalized book suggestions, Tewari et al. [18] presented a comprehensive recommendation system that makes use of content filtering, collaborative filtering, and association rule mining. Using classification and opinion mining techniques, Tewari et al. [19] propose books to reduce the amount of information available on e-commerce platforms. In order to protect user privacy while providing tailored suggestions, Luo, Le, and Chen [20] put special emphasis on privacy issues raised by users and suggest the Privacy-Preserving Book Recommendation System (PPBRS). Sariki et al. [21] proposed an enhanced framework for book recommendation accuracy that incorporates modules for Named Entity Recognition (NER), stylometry and visual feature extraction.A hybrid recommendation engine called NOVA is introduced by Pathak et al. [22] in order to offer effective and efficient book recommendations. Personalized suggestions are highlighted in Zhu et al.'s [23] investigation of collaborative filtering-based book recommendation algorithms. Lastly, based on rankings for college libraries Verma et al. [24] offer a hybrid recommendation system that exhibits a significant improvement in suggestion accuracy over earlier techniques.

The literature study includes an extensive spectrum of innovative techniques for book recommendation systems. A number of approaches are looked into, including opinion mining-based systems, hybrid models that integrate collaborative and content-based filtering, and temporal considerations, demographic information, and machine learning algorithms-based personalized recommendation systems. Studies also explore mobile technology integration in library settings, neighborhood-based collaborative filtering, and better algorithms like the improved Apriori algorithm. Additionally, studies show how important personality features, community detection methods, semantic linkages, and user-generated ratings are to improving suggestion accuracy. When considered collectively, these studies significantly improve the subject of book recommendation systems by complying with the requirements and interests of a diverse spectrum of users across several platforms and in a variety of situations. Table 1 is a comparative table of the literature survey.

Table 1: Comparative analysis of systems for recommendation.

Paper	Advantages	Disadvantages	Year
Mathew et al [1]	- A hybrid model that generates customized suggestions by integrating collaborative filtering,content-based filtering, and association rule mining.	- Complexity associated with integration of multiple techniques may pose implementation challenges	2016
Sohail et al.[2]	- Utilizes opinion mining for ranking books based on user reviews, aiding users in discovering top-ranked books efficiently	- Dependency on user reviews may introduce biases and inaccuracies in recommendations	2013
Kanetkar et al. [3]	- Integrates temporal aspects and demographic parameters into recommendation system for personalized recommendations	- Collection and management of demographic data may raise privacy concerns and require user consent	2014
Sarma et al. [4]	- Leverages machine learning algorithms for personalized recommendations, enhancing accuracy through clustering and similarity metrics	- Requires robust data preprocessing and feature engineering to ensure effectiveness of machine learning algorithms	2021
Mounika et al. [5]	- Offers a comprehensive recommendation system with multiple levels of processing, aiming for unparalleled accuracy within a minimal time frame	- Integration of multiple algorithms may increase computational overhead and complexity of the system	2021
Sivaramakrishnan et al. [6]	- Explores neighborhood-based collaborative filtering techniques to refine relevance of book recommendations for users	- Performance of neighborhood algorithms may degrade with large datasets, requiring optimization	2018
Zhou et al. [7]	- Introduces an efficient information recommendation system based on an improved Apriori algorithm, reducing computational overhead	- Adoption of Apriori algorithm may limit scalability and adaptability of the system	2020
Chen et al.[8]	- Integrates mobile technology into library environments, enhancing user experience and navigation through map-based guidance	- Reliance on mobile technology may introduce accessibility barriers for users with limited access to smartphones	2013

Continued on next page

Table 1 – continued from previous page

Paper	Advantages	Disadvantages	Year
Kuroiwa et al. [9]	- Introduces a dynamic Book Utilization System (BUS) leveraging web services and virtual library enhancements for personalized recommendations	- Dependence on web services may introduce vulnerabilities and security risks in the system	2007
Jia et al. [10]	- Collaboration and content-based filtering tools, along with software systems specifically designed for library management, can improve the efficiency of information retrieval.	- Adoption of System for Recommendation may require significant infrastructure and resource investment	2013
Hariadi et al.[11]	- Proposed a hybrid attribute and personality-based recommender system yielding superior results in personalized book recommendations	- Integration of personality traits into recommendation process may raise privacy concerns and require user consent	2017
Rahutomo et al. [12]	- Emphasizes user-generated ratings in collaborative filtering.	-Handling large volumes of user data may require significant resources.	2019
Sharma et al.[13]	- Alleviates product overload on e-commerce platforms; - Integrates Collaborative and Content-Based Filtering	- May still face limitations ofContent-Based Filtering and Collaborative Filtering approaches	2022
Wayesa et al. [14]	- Enhances book recommendations using semantic relationships; - Utilizes a pattern-based hybrid system	- Semantic relationships may not always capture nuanced user preferences effectively	2023
Xin et al. [15]	- Improves top-N recommendations, particularly in academic library; - Utilizes community detection techniques	- Community detection may be computationally intensive and require large datasets	2014
Mariana et al. [16]	- Builds an effective recommendation system for an Online Public Access Catalog (OPAC); - Utilizes association rule mining techniques	- Association rule mining may not capture complex user preferences accurately	2017
Desai et al. [17]	- Offers an enterprise-friendly recommendation system; - Addresses sparse data effectively using biclustering	- Biclustering may require significant computational resources and may not be suitable for all datasets	2016
Tewari et al. [18]	- Provides personalized book recommendations by integrating content filtering, association rule mining, and Collaborative Filtering	- Integration of multiple techniques may increase complexity and computational overhead	2014

Continued on next page

Table 1 – continued from previous page

Paper	Advantages	Disadvantages	Year
Tewari et al. [19]	- Addresses information overload on e-commerce platforms; - Leverages opinion mining and classification techniques for recommendations	- Opinion mining may not always accurately capture user sentiments and preferences	2014
Luo et al. [20]	- Safeguards user privacy while offering personalized recommendations; - Addresses user privacy concerns	- Privacy-preserving techniques may add complexity and computational overhead to the recommendation process	2009
Sariki et al. [21]	- Improves recommendation accuracy using an enhanced framework; - Utilizes Named Entity Recognition, visual feature extraction, and stylometry	- Enhanced framework may require significant computational resources and may not always capture user preferences accurately	2022
Pathak et al. [22]	- Provides efficient and effective book recommendations using NOVA, a hybrid recommendation engine	- Hybrid recommendation engines may require careful tuning and optimization to achieve optimal performance	2013
Zhu et al. [23]	- Emphasizes personalized recommendations using collaborative filtering algorithms	- Collaborative filtering may face challenges in capturing diverse user preferences accurately	2016
Verma et al. [24]	- Demonstrates significant improvement in recommendation accuracy for college libraries; - Utilizes a ranking-based hybrid recommendation system	- Implementation of the proposed system may require adaptation to different library environments and may not generalize well to all contexts	2024

3. Methodology

This research paper's methodology section outlines the strategy and methods used to accomplish our objective of creating a trustworthy book recommendation system. We employ a variety of methods, including collaborative filtering and matrix factorization, and we ground our strategy in data science and machine learning concepts. First, we describe the steps involved in prepping the data: loading the datasets, addressing missing values, and integrating the data. Next, we explore the basis of our recommendation system, which is composed of two primary phases: Matrix Factorization and Collaborative Filtering.

Using cosine similarity, we may discover comparable users based on how they have rated books during the Collaborative Filtering step. In the Matrix Factorization stage, the latent elements underlying user evaluations are extracted using Singular Value Decomposition (SVD). The subsequent subsections provide a detailed explanation of each of these phases as well as the underlying mathematical reasoning. We also go over the assessment metrics that we utilized to evaluate our model's performance.

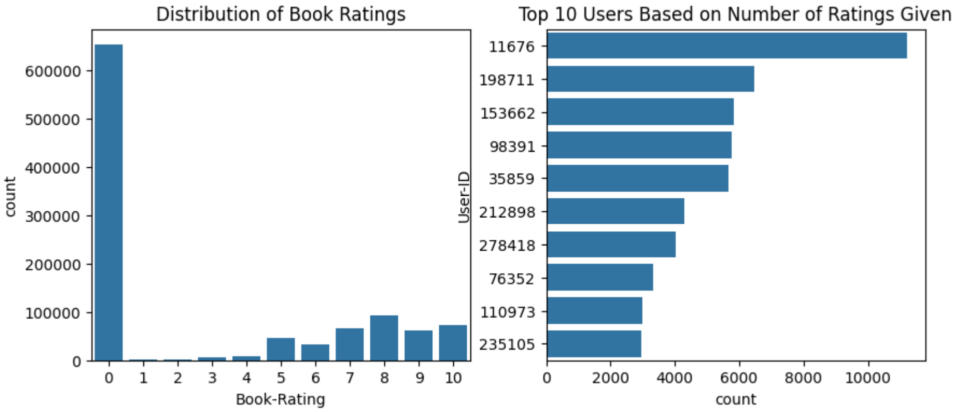

Figure 2: Distribution of Book Ratings and Top 30 Users Based on Number of Ratings Given.

3.1. Data Preprocessing

3.1.1. Datasets

Books.csv: This dataset contains details about books that may be found using their unique ISBNs. Content-based information from Amazon Web Services, such as publisher, year of publication, author, and title, is included. There are additional URLs to cover photos that go to Amazon's website in small, medium, and large versions.

Ratings.csv: This dataset includes explicit feedback ratings for books. Higher ratings, on a scale of 1 to 10, indicate greater appreciation. Users' implicit ratings, which indicate which books are rated and which are not, are represented by the number 0.

Users.csv: This dataset provides user data that has been anonymized and mapped to numbers in order to protect user privacy. If accessible, it contains demographic information such as age and location, which helps with tailored suggestions.

3.1.2. Initial Data Exploration and Handling Missing Values

We carry out a preliminary study to comprehend the data's organization after data loading. A data frame's number of rows and columns are determined by the shape parameter, and the head() method is used to print the data frame's first few rows. This gives us information on the size and appearance of the data. Next, we use the isnull().sum() function to search for missing values in our datasets; this returns the number of missing values for each column. Depending on how many and what kind of missing data were discovered, we may decide to eliminate the columns or rows that contain the missing values or replace them with a certain value (such the mean or median).

3.1.3. Data Integration and Refinement

Our three data frames are first merged using the merge() function to expedite the data preparation procedure. First, we merge ratings with books according to their "ISBN" column. Then, we combine the resulting dataframe with users according to their "User-ID." The result of this consolidation is a single dataframe that contains all the information required by our recommendation algorithm. We next perform data transformation on our merged dataframe by optimizing the 'Location' column. This involves removing the last section, which is separated by commas, from the "Location" string in order to isolate and save only the country information. Our dataset is made simpler and less dimensional by this improvement, which makes analysis and modeling more effective. Two key visualizations are included in Figure 2, a bar plot displaying the top 30 most active users by rating count, which is vital for assessing user behavior and enhancing the effectiveness of our recommendation system, and a rating distribution histogram for comprehending user preferences.

3.2. Filtering and Pivot Table Creation

Our data preprocessing pipeline starts with Filtration of Data, involving people who have rated over 200 books and novels with more than 50 ratings. By fostering recommendations based on solid data subsets, this curation seeks

Figure 3: Heatmap of User-Item Matrix for Top 100 Users and Booksn

to reduce data sparsity. We utilize groupby() and count() routines to count ratings for each user and book, then filter based on the results.

After data filtering, we create a pivot table, which results in an organized matrix with rows denoting book titles and columns denoting users. User-book ratings are wrapped within this framework to facilitate analysis. For dealing with missing data, we use the fillna() function to create a dense matrix that improves the efficiency of our recommendation system. The data preparation approach ensures our recommendation engine's accuracy and reliability. We maximize system speed by carefully preprocessing, providing users with relevant and trustworthy book recommendations. Figure 3 displays a heatmap of the user-item matrix for the top 100 users and their books. The heatmap displays user ratings for each book. Darker hues indicate higher ratings, and the color intensity in each cell reflects the rating value.

3.3. Model Description

We provide a highly personalized book recommendation system based on a sophisticated framework that combines the features of both matrix factorization and collaborative filtering. It starts with a collaborative user-to-user filtering method that finds users who are similar to each other based on how they rate books. The cosine similarity measure is used to assess the similarity. The system then use a popular matrix factorization technique called Singular Value Decomposition (SVD) to find the latent components that underlie user assessments. Such implicit components, representing both the user and the object, provide a deeper understanding of preferences of users.

By combining these methods, our system is able to manage user preference and item similarity in an efficient manner, which enhances the relevance and accuracy of the suggestions.The system is made to be scalable, meaning it can manage big datasets and yet function well. Figure 4 illustrates the proposed system's comprehensive architecture.

3.3.1. Collaborative Filtering

During the initial phase, the system utilizes a collaborative filtering approach between users. By comparing users to one another, the cosine similarity method pairs people who have comparable rating behaviors. By measuring the ratings of two users, two vectors create an angle that this metric calculates the cosine of. The cosine similarity measure ranges from 0 to 1, with 0 representing total dissimilarity and 1 indicating user equivalence. To find out how similar two users, u and v, are, we apply the cosine similarity metric:

$$sim(u,v) = \frac{\sum_{i \in I} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I} r_{ui}^2} \sqrt{\sum_{i \in I} r_{vi}^2}} \quad (1)$$

In this scenario, for item i, r_{ui} represents rating of "u" user, whereas I is the collection of objects rated by both u and v. After identifying comparable users, the algorithm suggests products they have liked. The recommend method returns a list of books that match the input book name.

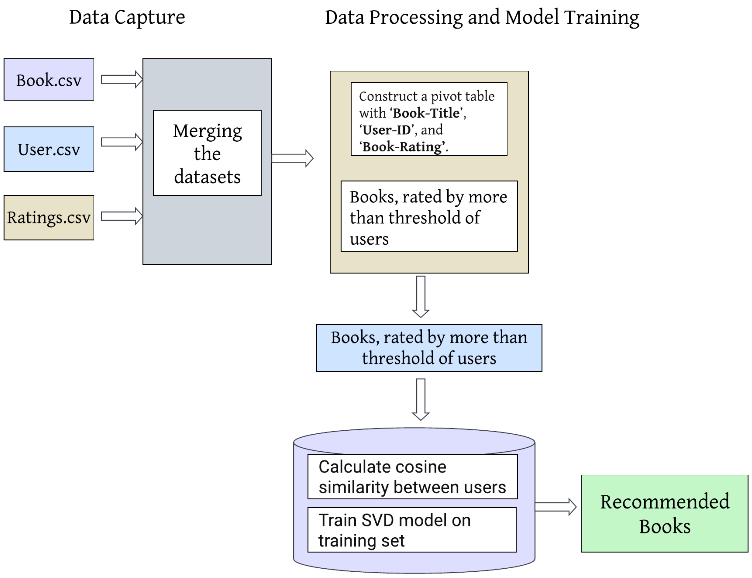

Figure 4: Comprehensive architecture of the proposed book recommendation system.

3.3.2. Matrix Factorization (SVD)

During the second phase, the Singular Value Decomposition (SVD) matrix factorization is used by the system to provide recommendations. SVD can be used to factor a user-item rating matrix, resulting in a diagonal matrix, two orthogonal matrices with more compact dimensions, and more. The SVD extracts the latent elements underlying user ratings while reducing the dimension of the original data. The underlying user and item properties are represented by these latent components.Books may have latent aspects such as author styles, duration, and genres. The SVD algorithm splits the user-item rating matrix R into a diagonal matrix and two orthogonal matrices of lesser dimension:

$$R = U\Sigma V^T \quad (2)$$

Where, Σ is the singular value diagonal matrix, which is essentially a weight matrix, U indicates the left singular vectors (user "features" matrix), R is the user-item rating matrix and the right singular vectors are denoted by V^T (item "features" matrix). Users' ratings for items j are implied by each element r_{ij} in R and R is divided into U, Σ, and V^T by the SVD in the following ways:

$$r_{ij} \approx \sum_{k=1}^{n} u_{ik}\sigma_k v_{kj}^T \quad (3)$$

Where, in the k^{th} column and i^{th} row of U, the element is denoted by u_{ik}, $_k$ is the kth singular value in , and v_{kj}^T is the element in the kth row and jth column of V^T. All n latent factors are included in the sum. The rating r_{ij} is roughly represented by this equation, which is the product of user features, singular values (weights), and item features. This expresses how each latent factor affects the rating. Figure 5 illustrates matrix factorization, a crucial step in our suggested design. On the right is the reduced-dimensional version of the original user-item interaction matrix, which reveals latent variables driving user-item engagement.

3.4. Algorithm Implementation

Our proposition is an innovative algorithm for book recommendations that utilizes Singular Value Decomposition (SVD) in conjunction with Cosine Similarity.The algorithm starts with randomly picked user and item profiles and makes iterative adjustments to them in an effort to minimize the discrepancy between actual and anticipated ratings. To handle data sparsity, it applies SVD to the user-item interaction matrix after each update, keeping just the top 'k' singular

≈

User ID / Latent Factor	α	β	γ	δ	ε
User 1	0.2	0.1	0.7	0.82	0.25
User 2	0.3	0.53	0.6	0.27	0.1
User 3	0.1	0.47	0.7	0.23	0.7
User 4	0.32	0.66	0.34	0.58	0.45
User 5	0.54	0.1	0.43	0.51	0.38

Figure 5: Demonstration of Matrix Factorization in Recommendation Systems

values. After that, it uses Cosine Similarity to generate an item-by-item similarity matrix, which is then used to suggest items to consumers. This adaptive and versatile method guarantees customized and relevant suggestions, enhancing the reader's experience. Algorithm 1 presents an algorithmic representation of the proposed recommendation model.

Algorithm 1 Book Recommendation System

Require: User-Item Interaction Matrix R, User Profile Matrix U, Item Profile Matrix I
Ensure : Updated User-Item Interaction Matrix R'
Training:
Initialize User Profile Matrix U and Item Profile Matrix I randomly
Compute the initial User-Item Interaction Matrix $R = U \times I^T$ **for** *each epoch* **do**
  for *each user u* **do**
    for *each item i rated by user u* **do**
      Compute the error $e_{ui} = r_{ui} - u_i \times i_u^T$
      Update user and item profiles:
      $u_i \leftarrow u_i + \gamma \times (e_{ui} \times i_u - \lambda \times u_i)$
      $i_u \leftarrow i_u + \gamma \times (e_{ui} \times u_i - \lambda \times i_u)$
    end
  end
  Calculate the revised User-Item Interaction Matrix's SVD. R
  Keep k singular values to get the approximated interaction matrix $R' = U_k \times \Sigma_k \times I_k^T$
  Compute item-item similarity matrix $S = \text{cosine\_similarity}(I_k)$
  for *each user u* **do**
    Compute the prediction score for unrated items Recommend the top-N items with the highest prediction scores
  end
end

3.5. Model Training

Our book recommendation system is based on two key algorithms: Cosine Similarity and Singular Value Decomposition (SVD). To implement it, we made a pivot table and utilized cosine similarity. The rows of the table represent books, the columns, users, and the cells, ratings. In order to address the issue of missing data, we have completed the gaps in this user-item matrix. We have then computed cosine similarity scores between novels based on these ratings. Our recommend feature uses these similarity scores to identify books that are most similar to a given book. The SVD component is the setting in which we have utilized the surprise library. Using Surprise's dataset format, we first split the data into a training set and a testing set. The SVD method is then defined and trained using the training set of data. After training, the model makes predictions on the test set. We adopt a hybrid strategy that combines matrix factorization (using SVD) with item-item collaborative filtering (using cosine similarity). This enables us to offer more accurate and tailored recommendations for books by utilizing the benefits of both approaches. Our model is designed to learn from users' past behavior (rated books) and use this data to forecast users' future behavior (book choices they would enjoy). Table 1 is a list of the hyperparameters that our book's recommendation system employed. To improve model performance, these parameters were chosen using grid search and cross-validation. The regularization term (reg_all), learning rate (lr_all), and number of iterations (n_epochs) are all included.

Hyperparameters	**Values**
Frequency of episodes(n_epochs)	20
Rate of learning(lr_all)	5×10^{-3}
Regularization term (reg_all)	~0.4

Table 1: Optimal Hyperparameters for the SVD Model in the Book Recommendation System.

4. Experimental Analysis

We assess the performance of our book suggestion system in detail in this section. Reasoning, parameters, and algorithms are all included in the methodology explanation. Metrics like RMSE, MAE, NMAE, Precision at k, and Recall at k are defined for assessing the findings. Results are presented in an understandable manner using tables and graphs. Expectations that were met, unexpected results, and parallels with earlier research are all discussed. We talk about constraints affecting findings and possible explanations for variations in results. The results provide directions for future investigation.

4.1. Evaluation Matrics

We combine error measures with error metrics to create assessment metrics. Three error metrics are used to quantify this difference between the actual predicted results: Root Mean Squared Error (RMSE), Normalized Mean Absolute Error (NMAE), and Mean Absolute Error (MAE).Two evaluation standards that assess the caliber of the model's top-K suggestions are Recall at K and Precision at K. Together, these measures offer a thorough assessment of our model's effectiveness, accounting for the accuracy of the predicted ratings as well as the relevancy of the recommended books.

4.1.1. Root Mean Square Error (RMSE)

The Root Mean Square Error (RMSE) statistic can be used to determine the average difference between the predicted and actual values in a given dataset. RMSE can be calculated by squaring, adding, and dividing the difference between the known and unknown points by the total number of test points. This is how we can mathematically express it: .

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i)-\hat{y}(i)\|^2}{N}} \quad (4)$$

Here, y(i)= i^{th} measurement, ŷ(i)= i^{th} prediction, N= Number of data points

4.1.2. Mean Absolute Error (MAE)

In many domains, such as statistics and machine learning, assessing how well a model's predictions correspond with actual data is essential. For this kind of evaluation, Mean Absolute Error (MAE) is an invaluable tool. MAE measures how much the actual and anticipated values differ from one another. It ignores the direction of the errors (positive or negative), in contrast to certain other error measures. In terms of math, MAE is represented as:

$$\text{MAE} = \frac{\sum_{i=1}^{n} abs(y_i - \lambda(x_i))}{n} \quad (5)$$

Here, $\lambda(x_{\mathrm{i}}) = i^{\mathrm{th}} prediction, y_{\mathrm{i}} = i^{\mathrm{th}} measurement, n = datapoints' quantity.$

4.1.3. Normalised Mean Absolute Error (NMAE)

The median number of mistakes in a series of forecasts is determined using the mean absolute error matrices, which do not take the direction of the errors into consideration. The absolute inaccuracy of every projection is the discrepancy between the actual and expected data. The Normalized Mean Absolute Error (NMAE) can be obtained by taking the mean of the absolute errors.

$$\text{NMAE} = \frac{\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|}{\max(y) - \min(y)} \quad (6)$$

N is the number of predictions. y_i is the actual value. $\hat{y}_i$ is the predicted value. $\max(y)$ and $\min(y)$ are the maximum and minimum values in the actual data, respectively. NMAE provides a normalized measure of the error of a regression mode. It helps assess how well the model predicts the target variable relative to its scale. NMAE is useful in scenarios where the scale of the target variables varies significantly or when comparing models trained on different datasets. By normalizing the error relative to the range of the target variable, NMAE provides a standardized measure of prediction accuracy.

4.1.4. Precision and Recall

The effectiveness of classification algorithms is evaluated using two key measures, precision and recall, especially in the context of recommendation systems. By dividing the total number of expected positives by the ratio of correctly predicted positive observations, one can calculate the accuracy of the model's positive predictions. A low false-positive rate for the model is indicative of good accuracy. By dividing the total number of accurately predicted positive observations by the total number of actual positive observations, recall evaluates the model's ability to identify all pertinent events. It is also known as true positive rate or sensitivity at times. A high recall rate in the model implies a low false-negative rate. Both precision and recall are crucial variables to consider when evaluating recommendation systems, as they ensure that most recommendations are relevant. It's important to keep these aspects in balance because improving one usually means sacrificing the other. One popular method for striking this equilibrium is the F1 score, or harmonic mean of precision and recall. The following is the precision and recall scores' mathematical definition:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

Where, TP (True Positives) is the number of correct positive predictions. FP (False Positives) is the number of incorrect positive predictions. FN (False Negatives) is the number of actual positives that were not identified by the model.

4.2. Experimental Setup and Configuration

We go into depth on the design and setup that we employed in this section of the study. The environment, hardware specs, software and libraries, dataset, algorithms and models, and assessment measures are all included in this. Here is a detailed summary of our experimental setup in Table 2.

Category	Details
Environment	Python,Google Colab (a Jupyter notebook environment hosted on the cloud)
Hardware Specifications	Core i5-1135G7, RAM8GB, GPU-NVIDIA GeForce MX330 with 2GB GDDR5
Software and Libraries	Python, pandas, NumPy, scikit-learn, scikit-surprise, matplotlib, seaborn
Dataset	Book.csv, Ratings.csv, Users.csv from Goodreads
Algorithms and Models	Collaborative Filtering (using Cosine Similarity), Matrix Factorization (using Singular Value Decomposition (SVD)), Hyperparameter tuning (using GridSearchCV)
Evaluation Metrics	RMSE, MAE, NMAE, Precision at k, Recall at k

Table 2: Overview of the Experimental Setup and Configuration for the Book Recommendation System.

4.2.1. Evaluation of the System

We have used several evaluation matrics to evaluate the performance of our approach. The findings are demonstrated in the Table 3. Better accuracy is indicated by lower MAE values; the range is 0 to infinity. Values smaller than one in the RMAE scale, which goes from 0 to infinity, indicate superior performance over the benchmark. NMAE normally has a range of 0 to 100 or 0 to 1, with values nearer 0 denoting greater accuracy. A higher percentage of actual positive forecasts among all positive predictions is shown by higher values on the accuracy scale, which runs from 0 to 1. Greater recall values signify a greater proportion of precise positive predictions among all actual positives. Recall also ranges from 0 to 1.

Assessment Measure	Value
RMSE	0.5035
MAE	0.6747
Precision at k	0.80
NMAE	0.278
Recall at k	0.60

Table 3: The book recommendation system's performance

The performance of prediction algorithms may be compared on different scales with the help of this normalized metric. Our computed accuracy at k is 0.80, which indicates that 80% of the top k books that are suggested to the user are pertinent. Given a recall of 0.60 at k, we may deduce that 60% of the pertinent books are included in the top-k recommendations. These results corroborate each other and show how well our hybrid recommendation system works to provide accurate book recommendations. In order to enhance comprehension and facilitate comparison of the performance metrics of our book recommendation system, we have employed bar charts (Figure 6) and radar charts (Figure 7) to visually represent the data. Figure 6 displays the values of a number of performance metrics, including RMSE, Precision at k, MAE, NMAE, and Recall at k. While greater values of Precision and Recall imply better performance, lower values of RMSE and MAE indicate higher accuracy. A normalized comparison of these measurements is shown in Figure 7, where each axis corresponds to a distinct statistic. The radar chart's full area represents the system's performance across all criteria; a larger area denotes higher overall performance. Lastly, we have examined the correlation between the anticipated and actual book ratings using the scatter plot displayed in Figure 8. Each point represents a book's rating, where y is the predicted rating and x is the actual rating. For flawless predictions, points should ideally line up along the diagonal from bottom left to top right.

4.2.2. Comparison with State-of-the-art Methods

Our hybrid book recommendation system has been compared and contrasted with a number of different research projects. More specifically, we have chosen to compare our work with cosine similarity-based recommendation systems for books, independent of the model. A variety of metrics were looked at, such as normalized mean absolute error (NMAE), precision, root mean square error (RMSE), recall and mean absolute error (MAE). We have chosen a number of studies for this comparison, including those by Rohit et al. [25], Amer et al. [26], Anwar et al. [27], and Monika et al. [24]. These studies offer a solid foundation for comparison since they cover a wide range of recommendation systems techniques and methodology. The comparison has been shown in Table 4.

Models	MAE	RMSE	NMAE	Precision	Recall
Rohit et al. [25]	2.631	2.99	0.292	-	-
Amer et al. [26]	0.77	-	-	0.0316	0.2045
Anwar et al. [27]	0.72	0.92	-	-	-
Monika et al. [24]	0.074	1.0246	-	0.6834	-
Proposed Model	0.6747	0.5035	0.278	0.80	0.60

Table 4: Comparison with the existing mehods.

This assessment is meant to demonstrate our hybrid recommendation system's precision and efficiency.We can more clearly grasp the advantages of our strategy and pinpoint areas in need of future development by contrasting

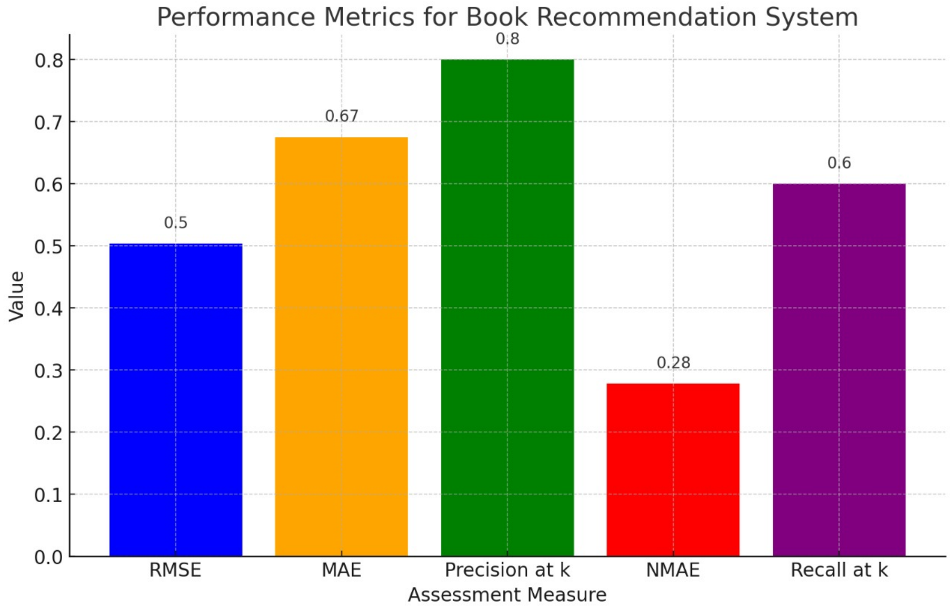

Figure 6: Bar Chart of Performance Matrics

our findings with those of these earlier investigations. The portions of this article that follow will provide a thorough comparison and debate.

4.2.3. Comparative Performance Analysis of Recommendation Models Using MAE and RMSE

In this part, we have evaluated our hybrid recommendation system's performance against recommendation models that used Singular Value Decomposition (SVD) or K-Nearest Neighbors (KNN) to see how effective it is. We focused on Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) because they are commonly used metrics to assess the accuracy of predictive models. We have selected the following studies for the KNN comparison: The studies of Anagha et al.[32], Krishnan et al.[35], Li et al.[36], and Anwer et al.[27] have been selected for the SVD comparison. In particular, we have taken the average RMSE and average MAE of three datasets (Hindi Movie, Book Cross, and Movielens) for Krishnan et al. [35]. Table 5 shows this comparison. Nguyen et al. [31], Anagha et al. [32], Rohit et al. [25], Esmael et al. [33], S. G. K. Patro et al. [34], and Anwer et al. [27]. This comparison has been demonstrated in Table 6.

The outcomes showed that the hybrid model—which integrates cosine similarity and SVD—performed better in both criteria. Our hybrid model efficiently tackles the drawbacks of employing KNN or SVD alone by utilizing cosine similarity's connection strengths to find underlying variables and produce more accurate and pertinent book recommendations.

Models	RMSE	MAE
Anagha et al. [32]	1.4346	1.2152
Krishnan et al. [35]	0.88	0.68
Li et al. [36]	0.9	-
Anwer et al. [27]	0.979	0.7732
The proposed model	0.5035	0.6747

Table 5: Comparative Analysis of Performance Metrics (MAE and RMSE) for SVD based Models and our proposed method.

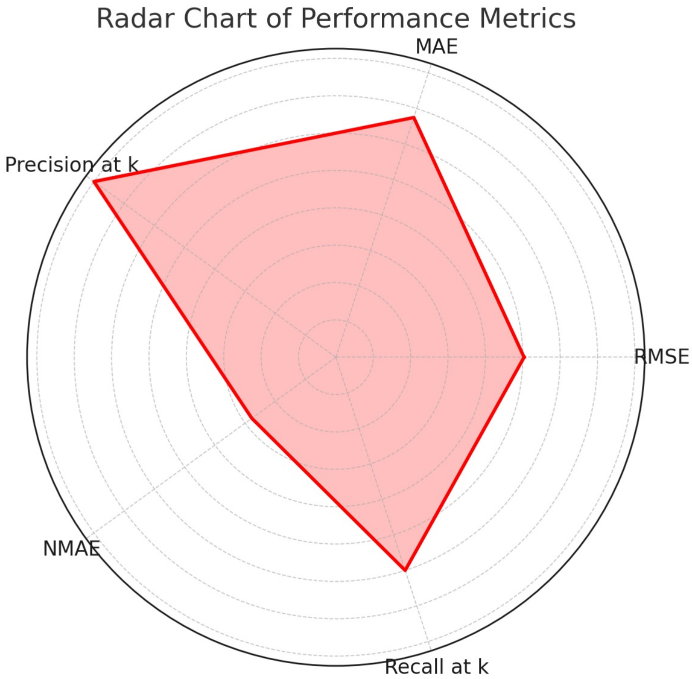

Figure 7: Radar Chart of Performance Metrics

Models	RMSE	MAE
Nguyen et al. [31]	1.087	0.856
Anagha et al. [32]	1.4346	1.2152
Rohit et al. [25]	2.99	2.631
Ahmed et al. [33]	1.0535	-
S. G. K. Patro et al. [34]	0.73	0.7165
Anwer et al. [27]	0.979	0.7732
The proposed model	0.5035	0.6747

Table 6: Comparative Analysis of Performance Metrics (MAE and RMSE) for KNN-based Models and the Proposed Hybrid Model..

4.2.4. Recommendation Process

The matrix factorization and collaborative filtering approaches are combined in our recommendation system. User and book ratings are amalgamated, missing values are filled in, and users with more than 200 ratings and books with more than 50 ratings are removed throughout the data preparation process. Next, a pivot table showing user ratings is built, with rows for each book and columns for each user. The zeros in missing ratings are used.

The cosine similarity between novels is then determined by taking into account each reader's rating. To suggest books that are related to a certain book, we utilize the similarity score matrix that has been constructed. The suggest function organizes a list of relevant books based on similarity scores after obtaining a book title as input. We simultaneously use the Surprise package's SVD algorithm for forecasting. There are two kinds of data: training and test sets. The SVD model is trained using the training set, and predictions are made on the test set. The model is evaluated using a variety of measures, including RMSE, MAE, NMAE, accuracy at k, and recall at k.

The top ten books that have been recommended for two users (User IDs: 2313 and 276727) are shown in Figures 9 and 10. The "recommend books" feature generates these recommendations by predicting the user's ratings for all books they haven't yet rated, arranging those forecasts in a decreasing order, and then going back to the top ten books.

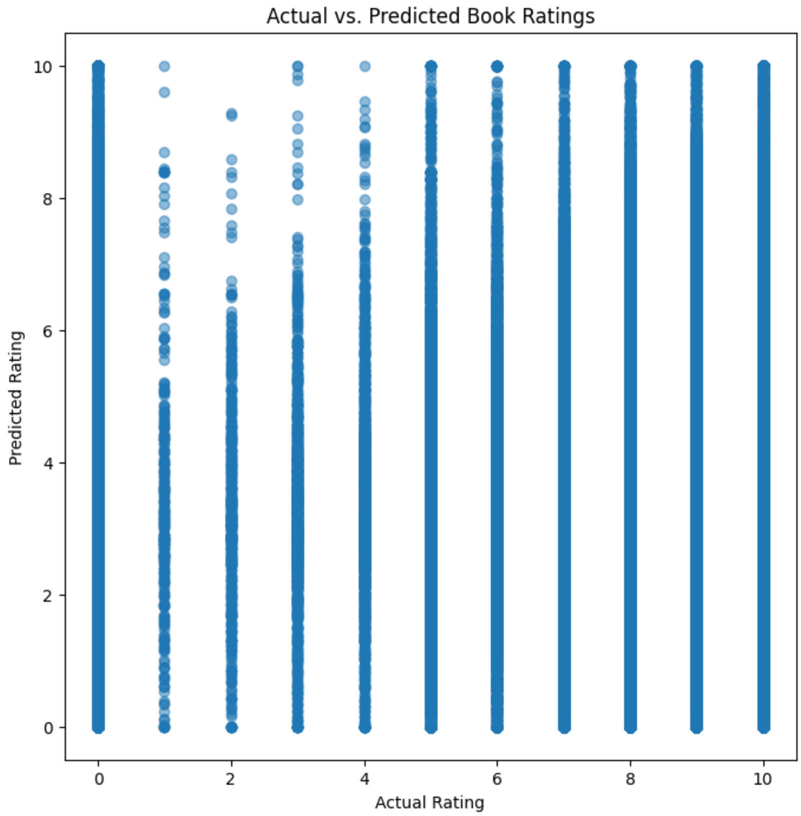

Figure 8: Scatter plot of Actual vs. Predicted Book Ratings

```
Top 10 recommended books for user 2313:
1. The Other Boleyn Girl
2. The Rescue
3. Drums of Autumn
4. Contact
5. The Boy Next Door
6. Artemis Fowl (Artemis Fowl, Book 1)
7. The Godfather
8. On Writing
9. Tales of the City (Tales of the City Series, V. 1)
10. All I Need to Know I Learned from My Cat
```

Figure 9: Top 10 Books Suggested Reading for "User 2313"

The user gets personalized recommendations determined by their previous ratings as well as the ratings of other people who are similar to them. These serve as a stand-in for the novels that our model anticipates readers would like highly. By providing these customized suggestions, our algorithm assists users in discovering new books that are relevant to their interests. In summary, our hybrid recommendation system uses both item similarity and user behavior to provide tailored book suggestions.

```
Top 10 recommended books for user 276727:
1. Free
2. The Giver (Readers Circle)
3. Marching Through Culpeper : A Novel of Culpeper, Virginia, Crossroads of the Civil War
4. The Greatest Discovery
5. El Hobbit
6. Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))
7. The Giving Tree
8. Love You Forever
9. Ishmael
10. Falling Up
```

Figure 10: Top 10 Books Suggested Reading for "User 276727"

5. Discussion

Our hybrid recommendation system, which fused collaborative filtering and matrix factorization, produced good results. Recommendations for books were made by the algorithm using both user-equivalent ratings and ratings from prior interactions. Performance metrics such as MAE, NMAE, RMSE, recall at k, accuracy at k, and others were employed to evaluate the system. With an RMSE of 0.5035 and an MAE of 0.6747, the system can make reasonably accurate predictions. It is crucial to remember that such metrics provide only a limited overview of the system's functionality. For instance, the recall at k of 0.60 implies that there might be relevant books that are removed from the top-k recommendations, even though the precision at k of 0.80 indicates that a significant portion of the top k books recommended are relevant to the reader. This points up a possible place where our system needs to be improved.

The scatter plot of actual vs. predicted book ratings visually represented the relationship between the ratings. Every point on the plot denotes a book rating; the y-coordinate displays the rating that our model predicted, and the x-coordinate shows the actual rating provided by a user. This visual tool facilitates comprehension of our model's performance. Moreover, the top 10 book suggestions that the system generated for a particular user (User ID: 276727) show how personalized it is.These recommendations, which are books the user is likely to rank highly based on our algorithm, show that they are pertinent to the user's interests. Lastly, our study has shown that a hybrid recommendation system may effectively provide personalized book recommendations.

Nevertheless, there are lots of chances for growth and more research. Further research and development will focus on enhancing the model, exploring other recommendation methodologies, and incorporating additional features to increase the amount of customization offered by the suggestions. An illustration of how each factor—RMSE, MAE, Precision at K, and Recall at K—affects the system's performance is given in the pie chart in Figure 9. Readers will have a better idea of the aspects that determine the accuracy of our book suggestions by looking at this chart.

6. Future Scope and Limitations

We have shown that our hybrid recommendation system performs promisingly when it comes to book recommendations using a blend of collaborative filtering and matrix factorization approaches. Still, there are a number of directions that future study may go in order to improve our system's functionality and performance. The "cold start" problem—which describes the challenge of proposing new users or things in the lack of past data—is one of these; evaluating long-term user satisfaction via longitudinal studies; guaranteeing the scalability and efficacy of the recommendation system as the dataset grows; and investigating the applicability of our recommendation system to different item types, such as products, music, or movies. Additionally, these include exploring other recommendation algorithms or ensemble methods to improve model performance and incorporating additional features like book genres, author information, or user demographic data. We think that by investigating these avenues, future studies can further the area of recommendation systems and provide consumers with more tailored and fulfilling experiences.

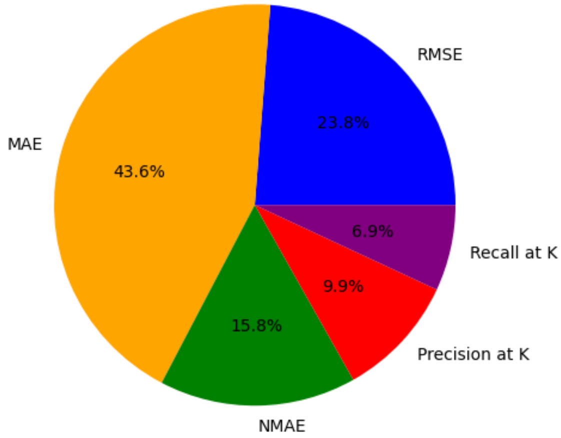

Figure 11: The relative contributions of several factors to the recommendation system's overall performance

7. Conclusion

We summarize the main conclusions and future directions of our research in the closing remarks. With the use of matrix factorization and collaborative filtering, our team has successfully developed a hybrid recommendation system that offers tailored book recommendations. Metrics like RMSE (0.5035) and MAE (0.78), which measure the system's performance, show a respectable degree of accuracy. Although our approach has yielded encouraging results, we recognize that it may be improved. To guarantee that more pertinent books are included in the top-k suggestions, possible areas for improvement are indicated by the accuracy at k of 0.80 and recall at k of 0.60. Our work advances the continuous creation of efficient and customized recommendation systems. In order to improve personalization in future work, we plan to investigate other recommendation strategies, add new features, and further develop our model. This dedication to ongoing development highlights our goal of providing very precise and customized book suggestions.

Credit Authorship Contribution Statement

Mahatir Ahmed Tusher: Conceptualization, Methodology, Software, Writing - Original Draft (lead), Writing - Review Editing. **Saket Choudary Kongara**: Writing - Review Editing, Validation. **Gangavarapu Sreeram**: Writing - Review Editing, Validation. **Srinivas Arukonda**: Formal analysis, Supervision. **Sheikha Farook Batha**: Visualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

Data are available upon request.

Acknowledgments

No acknowledgments.

Reference

1. P. Mathew, B. Kuriakose and V. Hegde, "Book Recommendation System through content based and collaborative filtering method," *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*, Ernakulam, India, 2016, pp. 47-52, doi: 10.1109/SAPIENCE.2016.7684166.
2. S. S. Sohail, J. Siddiqui and R. Ali, "Book recommendation system using opinion mining technique," *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Mysore, India, 2013, pp. 1609-1614, doi: 10.1109/ICACCI.2013.6637421.
3. S. Kanetkar, A. Nayak, S. Swamy and G. Bhatia, "Web-based personalized hybrid book recommendation system," *2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014)*, Unnao, India, 2014, pp. 1-5, doi: 10.1109/ICAETR.2014.7012952.
4. Sarma, D., Mittra, T., & Shahadat, M. (2021). Personalized Book Recommendation System using Machine Learning Algorithm. *International Journal of Advanced Computer Science and Applications, 12*, 212-219.
5. Mounika, A., Saraswathi, S. (2021). Design of Book Recommendation System Using Sentiment Analysis. In: Suma, V., Bouhmala, N., Wang, H. (eds) Evolutionary Computing and Mobile Sustainable Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 53. Springer, Singapore. https://doi.org/10.1007/978-981-15-5258-8_11.
6. Sivaramakrishnan N, Subramaniyaswamy V, Arunkumar S, Renugadevi A, Ashikamai Kk. Neighborhood-based approach of collaborative filtering techniques for book recommendation systems. International Journal of Pure and Applied Mathematics, 2018. ⟨hal-01826674⟩
7. Zhou, Y.W. (2020) Design and Implementation of Book Recommendation Management System Based on Improved Apriori Algorithm. Intelligent Information Management, 12, 75-87
8. Chen, C. M. (2013). An intelligent mobile location-aware book recommendation system that enhances problem-based learning in libraries. *Interactive Learning Environments*, *21*(5), 469–495. https://doi.org/10.1080/10494820.2011.593525
9. T. Kuroiwa and S. Bhalla, "Dynamic Personalization for Book Recommendation System Using Web Services and Virtual Library Enhancements," *7th IEEE International Conference on Computer and Information Technology (CIT 2007)*, Aizu-Wakamatsu, Japan, 2007, pp. 212-217, doi: 10.1109/CIT.2007.72.
10. Jia, F., Shi, Y. (2013). Library Management System Based on Recommendation System. In: Yang, Y., Ma, M., Liu, B. (eds) Information Computing and Applications. ICICA 2013. Communications in Computer and Information Science, vol 392. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-53703-5_50
11. I. Hariadi and D. Nurjanah, "Hybrid attribute and personality based recommender system for book recommendation," *2017 International Conference on Data and Software Engineering (ICoDSE)*, Palembang, Indonesia, 2017, pp. 1-5, doi: 10.1109/ICODSE.2017.8285874.
12. R. Rahutomo, A. S. Perbangsa, H. Soeparno and B. Pardamean, "Embedding Model Design for Producing Book Recommendation," *2019 International Conference on Information Management and Technology (ICIMTech)*, Jakarta/Bali, Indonesia, 2019, pp. 537-541, doi: 10.1109/ICIMTech.2019.8843769.
13. Sharma, S., Rana, V. & Malhotra, M. Automatic recommendation system based on hybrid filtering algorithm. *Educ Inf Technol* **27**, 1523–1538 (2022). https://doi.org/10.1007/s10639-021-10643-8
14. Wayesa, F., Leranso, M., Asefa, G. *et al.* Pattern-based hybrid book recommendation system using semantic relationships. *Sci Rep* **13**, 3693 (2023). https://doi.org/10.1038/s41598-023-30987-0
15. Xin, L., E, H., Song, J., Song, M., Tong, J. (2014). Book Recommendation Based on Community Detection. In: Zu, Q., Vargas-Vera, M., Hu, B. (eds) Pervasive Computing and the Networked World. ICPCA/SWS 2013. Lecture Notes in Computer Science, vol 8351. Springer, Cham. https://doi.org/10.1007/978-3-319-09265-2_37
16. S. Mariana, I. Surjandari, A. Dhini, A. Rosyidah and P. Prameswari, "Association rule mining for building book recommendation system in online public access catalog," *2017 3rd International Conference on Science in Information Technology (ICSITech)*, Bandung, Indonesia, 2017, pp. 246-250, doi: 10.1109/ICSITech.2017.8257119.
17. T. Desai, S. Gandhi, P. Murlidhar, S. Gupta, M. Vijayalakshmi and G. P. Bhole, "An enterprise-friendly book recommendation system for very sparse data," *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, Pune, India, 2016, pp. 211-215, doi: 10.1109/CAST.2016.7914968.

18. A. S. Tewari, A. Kumar and A. G. Barman, "Book recommendation system based on combine features of content based filtering, collaborative filtering and association rule mining," *2014 IEEE International Advance Computing Conference (IACC)*, Gurgaon, India, 2014, pp. 500-503, doi: 10.1109/IAdCC.2014.6779375.
19. A. S. Tewari, T. S. Ansari and A. G. Barman, "Opinion based book recommendation using Naive Bayes classifier," *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, Mysore, India, 2014, pp. 139-144, doi: 10.1109/IC3I.2014.7019672.
20. Y. Luo, J. Le and H. Chen, "A Privacy-Preserving Book Recommendation Model Based on Multi-agent," *2009 Second International Workshop on Computer Science and Engineering*, Qingdao, China, 2009, pp. 323-327, doi: 10.1109/WCSE.2009.822.
21. Sariki, T. P. ., & Guntur, B. K. (2022). AN AGGRANDIZED FRAMEWORK FOR ENRICHING BOOK RECOMMENDATION SYSTEM. *Malaysian Journal of Computer Science*, *35*(2), 111–127. https://doi.org/10.22452/mjcs.vol35no2.2
22. D. Pathak, S. Matharia and C. N. S. Murthy, "NOVA: Hybrid book recommendation engine," *2013 3rd IEEE International Advance Computing Conference (IACC)*, Ghaziabad, India, 2013, pp. 977-982, doi: 10.1109/IAdCC.2013.6514359.
23. Y. Zhu, "A book recommendation algorithm based on collaborative filtering," *2016 5th International Conference on Computer Science and Network Technology (ICCSNT)*, Changchun, China, 2016, pp. 286-289, doi: 10.1109/ICCSNT.2016.8070165.
24. Monika Verma, Pawan Kumar Patnaik,An automatic college library book recommendation system using optimized Hidden Markov based weighted fuzzy ranking model,Engineering Applications of Artificial Intelligence,Volume 130,2024,107664,ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2023.107664.
25. Rohit, Sabitha, S., Choudhury, T. (2018). Proposed Approach for Book Recommendation Based on User k-NN. In: Bhatia, S., Mishra, K., Tiwari, S., Singh, V. (eds) Advances in Computer and Computational Sciences. Advances in Intelligent Systems and Computing, vol 554. Springer, Singapore. https://doi.org/10.1007/978-981-10-3773-3_53
26. Amer, A. A., Abdalla, H. I., & Nguyen, L. (2021). Enhancing recommendation systems performance using highly-effective similarity measures. Knowledge-Based Systems, 217, 106842. ISSN 0950-7051. https://doi.org/10.1016/j.knosys.2021.106842.
27. Anwar, T., Uma, V. Comparative study of recommender system approaches and movie recommendation using collaborative filtering. Int J Syst Assur Eng Manag 12, 426–436 (2021). https://doi.org/10.1007/s13198-021-01087-x
28. Hong, H., Guo, J., & Wang, B. (2012). An Improved KNN Algorithm Based on Adaptive Cluster Distance Bounding for High Dimensional Indexing. International Journal of Creative Research Thoughts (IJCRT). https://doi.org/10.1109/gcis.2012.86
29. Singh, Ramni & Maurya, Sargam & Tripathi, Tanisha & Narula, Tushar & Srivastav, Gaurav. (2020). Movie Recommendation System using Cosine Similarity and KNN. International Journal of Engineering and Advanced Technology. 9. 2249-8958. 10.35940/ijeat.E9666.069520
30. Bhardwaj, Aditya. (2023). Movie Recommendation System Using SVD (Letterboxd). IJARCCE. 12. 10.17148/IJARCCE.2023.121013.
31. Nguyen, L. V., Vo, Q.-T., & Nguyen, T.-H. (2023). Adaptive KNN-Based Extended Collaborative Filtering Recommendation Services. *Big Data and Cognitive Computing*, 7(2), 106. https://doi.org/10.3390/bdcc7020106
32. Vaidya, A., & Shinde, S. (2019). Hybrid Book Recommendation System. *International Research Journal of Engineering and Technology (IRJET)*, 6(7).
33. Ahmed, E., & Letta, A. (2023). Book Recommendation Using Collaborative Filtering Algorithm. *Applied Computational Intelligence and Soft Computing*, 2023, 1–12. https://doi.org/10.1155/2023/1514801
34. Patro, S. G. K., et al. (2020). A Hybrid Action-Related K-Nearest Neighbour (HAR-KNN) Approach for Recommendation Systems. *IEEE Access*, 8, 90978-90991. https://doi.org/10.1109/ACCESS.2020.2994056
35. Yadav, K. K., Soni, H. K., & Pathik, N. (2023). Recommendation System Based on Double Ensemble Models using KNN-MF. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 14(5). http://dx.doi.org/10.14569/IJACSA.2023.0140566

36. Li, G., & Zhang, J. (2018). Music personalized recommendation system based on improved KNN algorithm. In *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (pp. 777-781). https://doi.org/10.1109/IAEAC.2018.8577483