Enhancing Book Recommendation Accuracy through User Rating Analysis and Collaborative Filtering Techniques

An Empirical Analysis

Jirat Chongwarin
Khon Kaen University, Thailand

Paiboon Manorom
Khon Kaen University, Thailand

Vispat Chaichuay
Khon Kaen University, Thailand

Tossapon Boongoen
Aberystwyth University, United Kingdom

Chunqiu Li
Beijing Normal University, China

Wirapong Chansanam
Khon Kaen University, Thailand

Abstract: Since online electronic books have become popular, book recommendation systems have been invented and challenged to handle the high demand from users in the digital era. This study aimed to develop and evaluate a book recommendation model using data mining techniques through RapidMiner Studio. The datasets used were comprised of 981,756 user ratings. Before conducting the data analytics, the data was pre-processed to eliminate duplicates and retain only the highest ratings. Collaborative Filtering (CF) techniques, particularly k-Nearest Neighbours (k-NN) and Matrix Factorization (KF), were employed to elicit insightful information for development and to highlight their capabilities in handling enormous datasets. Furthermore, statistical analysis, visualization, elementary modelling, and model combinations were investigated to compare their performance.

To reinforce creditability, modelling techniques and parameter adjustments were integrated to optimize the performance of the algorithms, since the results indicated that different model settings and data partitions impacted the effectiveness of the recommendation system.

https://doi.org/10.18080/jtde.v12n3.976

Additionally, these results demonstrated the potential of hybrid models in improving the accuracy and efficiency of recommendation systems and highlighted the trade-off between algorithmic approaches and dataset characteristics that interplay in optimizing the performance of recommendation systems.

Keywords: collaborative filtering, k-Nearest Neighbours (k-NN), matrix factorization, RapidMiner Studio, data mining techniques.

Introduction

Book Recommendation Systems

Recent recommendation systems (RSs) have significantly improved to provide tailored suggestions to enhance user experience and satisfaction. As a result, this study comprehensively explores various RS methodologies through hybrid systems, which integrated multiple filtering techniques to address the recommendation system challenges, such as scalability, accuracy, and new user integration. The recommendation systems for books have been developed for a decade to improve user (reader) support and precise decision-making for the right target (buyer). Additionally, RSs have been implemented in both commercial business and educational organizations. For instance, in the book recommendation domain, Mathew *et al.* (2016) integrated content-based filtering, collaborative filtering, and association rule mining to resolve the inefficiencies in existing systems by providing more precise book recommendations to align with individual preferences.

The collaborative filtering (CF) technique was introduced to make precise recommendations to boost RSs for higher performance. CF is considered a critical feature in recommendation systems because it can suggest items based on user interaction patterns and operate through user- and item-based approaches by employing similarity metrics to generate recommendations. For example, Rana & Deeba (2019) introduced a method that enhances the accuracy of CF by incorporating Jaccard Similarity to evaluate user overlap between products, improving the effectiveness of recommendations on electronic book (e-book) platforms like Amazon and Goodreads. The latest advances in recommendation systems from this study have significantly improved the RS's ability to provide tailored suggestions that enhance user experience. Another extensive study in recommendation systems was found in You *et al.* (2023), which introduced the Deep Ranking Weighted Multi-Hashing Recommender (DRWMR) System. The system enhances recommendation accuracy and offers explanations for its suggestions, addressing key challenges in CF systems. To confirm this feature, the study

https://doi.org/10.18080/jtde.v12n3.976

by Kumar *et al.* (2023) highlighted the importance of hybrid algorithms that combine CF with content-based filtering to improve recommendation accuracy and personalization to leverage the strengths of these approaches.

Furthermore, Zhao *et al.* (2022) addressed the challenge of sparse data in CF. They proposed a new item-based CF algorithm to handle the vague datasets with Kullback-Leibler (KL) divergence for measuring item similarity and a novel neighbour selection technique. The study improved prediction accuracy to enhance adaptability to sparse data by integrating more comprehensive rating information, as demonstrated by superior performance on benchmark datasets by the Bayesian Personalized Ranking Smart Linear Model (BPRSLIM). The BPRSLIM is a model-based CF algorithm that enhances accuracy by reconstructing the user-item matrix using item-feature information. This model was adopted by the study of Isinkaye (2023), which conducted extensive experiments to reveal significant improvements in recommendation accuracy, especially in precision and normalized Discounted Cumulative Gain, by 30.6% and 22.1%, respectively.

Nonetheless, although RSs have been developed to be compatible with sparse data, some limitations of explicit ratings in sparse data environments remain, such as in the study of Al-Ghuribi *et al.* (2023), which utilized sentiment analysis of user reviews to derive implicit ratings. This study proposed methods considering sentiment degrees and aspect-sentiment word pairs by combining these with explicit ratings to improve CF performance from Amazon and Yelp datasets. This study demonstrated that these approaches significantly enhanced the accuracy of CF rating prediction algorithms.

Collaborative Filtering integrations

Patel & Patidar (2018) conducted a comparative assessment by exploring a hybrid recommendation model for an online book portal that adapts to user demand and book quality. Furthermore, the study of Mankar *et al.* (2023) emphasized the inclusion of the k-Nearest Neighbours (k-NN) machine-learning model within a CF framework to categorize books based on user preferences to boost customer attraction and revenue for online bookstores by enabling users to select the most fitting books. Devika *et al.* (2021) described a CF system to encourage reading habits by matching book selections with user interests through extensive datasets and machine learning techniques. Correspondingly, Bharathi (2019) developed a hybrid RS for video streaming that integrates user feedback with content and CF algorithms to overcome existing limitations and enhance personalization. Apart from integrating machine learning and algorithms, CF was also integrated with social media platforms for higher prediction efficacy, as in the study of Singh & Banerjee (2023), which enhanced e-commerce book recommendations by integrating item-to-item in the CF

https://doi.org/10.18080/jtde.v12n3.976

approaches. Social media interactions and sentiment analysis from Goodreads, Amazon reviews, and Twitter were the main data platforms for data collection and analysis. Based on the study results, the model has achieved high accuracy in predicting user preferences.

Furthermore, when addressing the complex task of predicting multi-label interests in hybrid news recommendation systems, Saravanapriya *et al.* (2022) proposed a deep learning approach using a multi-label Convolutional Neural Network (CNN) to predict user preferences. This feature forecasted the user interests across 15 labels and identified popular news articles by clustering labels mined from social media platforms, such as Facebook and Twitter. The system was integrated with the latest news feeds to generate recommendations to show prediction performance improvement by 5.87%, 12.09%, and 18.49% through machine learning, measured by Support Vector Machine, Decision Tree, and Naïve Bayes, respectively. Sarimehmetoğlu & Erdem (2023) developed a deep learning-based system through YOLOv5 to extract book titles from promotional videos on social media, facilitating efficient video navigation and enhancing text extraction and video analysis. Additionally, the study of Ifada *et al.* (2019) investigated the probabilistic-keyword CF approaches to improve library book recommendations by addressing the sparsity issue in the traditional CF approaches. This study employed circulation and keyword matrices, integrated them into a keyword model, and then adapted a probabilistic technique to generate top-N book recommendations. The results indicated that the performance of the user- and item-based CF methods obtained higher performance than that of other approaches.

Similarly, the study of Tian *et al.* (2019) developed a personalized book recommendation system for college libraries through hybrid algorithms, which integrated CF and content-based methods to improve user-item scoring matrices and clustering to address data sparsity and accuracy. Additionally, Al-Hagery (2020) proposed a hybrid RS that integrates clustering algorithms, user-based CF, and cosine similarity measures to illustrate significant improvements in product recommendation accuracy and reliability. The study of Rao *et al.* (2021) described a hybrid RS that offers personalized book and audiobook suggestions by combining user profiles with machine learning, which corresponded with the study by Raghavendra & Srikantaiah (2022), which addressed the cold-start problem through demographic data to enhance the quality and accuracy of book recommendations.

The recommendation system challenges

Although RSs have been adopted with extensive integration and in various contexts, some challenges in handling user preferences correspond with current customer demand in the digital era. For example, the study of Pasricha & Solanki (2019) identified that influential social network users enhanced book RSs by providing a novel approach to generate accurate

https://doi.org/10.18080/jtde.v12n3.976

recommendations even without extensive purchase history. Similarly, Bharathipriya *et al.* (2020) conducted a collaborative filtering RS with classification and clustering algorithms to categorize similar customers and offer personalized product suggestions to enhance Business-to-Consumer relationships. To overcome these challenges, Hikmatyar (2020) developed a desktop-based book search recommendation system for the University of Struggle library using Python and MySQL, which employed a user-based CF method, which systematically recommended books by analyzing user-profiles and lending patterns, enhancing search efficiency and accuracy. Similarly, Muneer *et al.* (2022) also developed a content-based filtering RS tailored for tour spots that aligns destination suggestions with user preferences, enhancing tourist satisfaction. Furthermore, Munaji & Emanuel (2021) leveraged user ratings and Pearson correlation to gauge user similarity, tailoring restaurant recommendations to user preferences.

To enhance the efficacy ratios, some recent studies employed the integration of machine learning algorithms, such as the study of Chaturvedi *et al.* (2023), which developed a hybrid book recommender system that employs K-means and Gaussian mixture models for clustering, utilizing Term Frequency Inverse Document Frequency (TF-IDF) vectorization for content analysis and achieved high accuracy in recommending books during the COVID-19 pandemic and in the context of e-book availability challenges. Furthermore, Sarma *et al.* (2021) developed a recommendation system that utilizes clustering with K-means, Cosine Distance and Cosine Similarity to suggest books based on similarity, improving search efficacy in traditional user-based systems.

Emerging technologies and frameworks

Ang & Haw (2021) investigated recommender systems for enhancing public understanding and evaluating the accuracy of various methods in predicting item ratings. The study employed Python to create prototypes with graphical interfaces, enabling visualization and assessment of these systems' effectiveness for commercial implementation. In parallel, Li *et al.* (2023) introduced BookGPT, a framework that integrates large language model technology, especially generative pre-trained transformers (GPTs), into book recommendation systems to explore the application of ChatGPT for book rating, user rating, and summary of book recommendations, and to examine the potential of GPTs to enhance traditional models, especially in less data-rich contexts. Additionally, the studies of Khan *et al.* (2017) and Rajalakshmi *et al.* (2024) explored the effectiveness of CF, content-based methods, and hybrid algorithms in recommending products tailored to user interests, especially in the competitive environment of online platforms. The results indicated the dynamic nature and critical role of

https://doi.org/10.18080/jtde.v12n3.976

RSs in refining e-commerce and other online service platforms to confirm that the users received the most relevant and high-quality recommendations.

From the existing studies, we observed the dynamic evolution of recommendation systems emphasize the critical role of CF and hybrid approaches in delivering accurate and personalized recommendations amidst diverse data challenges. As a result, this study aimed to identify a significant gap in the scalability and integration of new users within existing recommendation systems. Despite advances in collaborative filtering (Jain & Vishwakarma, 2017), content-based filtering and hybrid models have struggled to manage the vast amounts of data generated by increasing users and products. This issue was further compounded by the cold-start problem involving new users or items with limited historical data. As a result, the data was complex and could not be integrated effectively into the recommendation algorithms. These difficulties have obstructed the system's ability to provide accurate recommendations from the onset of user interaction, impacting user satisfaction and system efficiency. Thus, these problems attract the researcher's curiosity to develop a more robust RS by integrating advanced machine-learning algorithms and leveraging demographic data to enhance system scalability and improve the accuracy for initial new users and of product recommendations.

Material and Methodology

To develop and evaluate a book recommendation model utilizing data-mining techniques, the researchers employed a comprehensive approach:

a) Dataset Selection: in this study, the researchers employed a well-established dataset pertinent to book recommendations, ensuring it contains a diverse array of books, user ratings, and associated metadata to facilitate a robust analysis.

b) Data Pre-processing: this step involves cleaning the data by removing duplicates, handling missing values, and filtering irrelevant information. The data was transformed and normalized to prepare it for effective analysis.

c) Technique Selection: the researchers selected appropriate data-mining techniques; collaborative filtering was employed in this study.

d) Model Development: this study employed the chosen techniques; the researchers constructed several models to predict user preferences. This involved setting up algorithms in a data-mining tool, RapidMiner, and configuring them with the necessary parameters.

e) Model Training and Validation: the models were trained using a portion of the dataset designated as the training set. The researchers then validated these models on a separate validation set to gauge their accuracy and effectiveness.

https://doi.org/10.18080/jtde.v12n3.976

f) Performance Evaluation: the researchers evaluated each model's performance through accuracy, precision, recall, F1 score, and the Area under the Receiver Operating Characteristics (ROC) curve (AUC). These metrics facilitated determining the model's effectiveness in recommending books that align with user preferences.

g) Model Optimization: based on the performance evaluations, models are fine-tuned and optimized by adjusting parameters, retraining, or incorporating additional features.

Data

The dataset in this study was in rating.csv format and was comprised of rating information for ten thousand widely read books. These ratings were acquired through online platforms, and the scoring system was ranked from 1 to 5, as Foxtrot (2024) documented. The dataset comprises 981,756 rows and 3 attributes; the specifics of each attribute are shown in Table 1.

Table 1. The dataset details

Attribute	Description	Type	Value
Book_id	Book code	Integer	1-10000
User_id	User ID	Integer	1-53424
rating	Score rating by user	Integer	1-5

Upon importing the dataset into RapidMiner Studio, the initial step involves Exploratory Data Analysis to survey and examine the data. Data visualization techniques were employed to facilitate comprehension of the dataset. It was observed that most users rated books between 3 and 5 points, with no ratings lower than 1 point. Subsequently, an analysis was conducted to determine the frequency of ratings per user-book pair. The study revealed that users rated books at least twice and as many as 200 times. Then, plotting this data on a line graph demonstrated high density within the range of 2 to 100 ratings by indicating a common rating frequency among users.

Furthermore, an examination of each book's ratings revealed that the least-rated books garnered 8 ratings, while the most-rated books received 100 ratings. Notably, the line graph depicting this data exhibited dense clusters within the 60 to 100 rating range, which suggested a consistent rating pattern across most books. Overall, these findings provide valuable insights into user behaviour and rating distribution within the dataset for developing the book recommendation model.

https://doi.org/10.18080/jtde.v12n3.976

Prepare data to be analysed

Data cleaning was prioritised in the preliminary stages of data preparation for analysis to ensure optimal quality. Upon scrutinizing the dataset, it was observed that certain users had submitted multiple ratings for the same book, which contradicts the intended protocol of a single rating per book per user; as a result, 4,487 duplicates were removed and only the highest-rated entry for each book by a user remained, with a total of 979,478 rows. Subsequently, a filtration process was employed for data rating distribution for each user to enhance the accuracy of performance. The analysis revealed that a substantial volume of rating data for each user is essential for the approach to selection. In this study, collaborative filtering techniques considered a user's historical preferences, and data filtering was focused on users with ratings for more than 20 books. To confirm the completion of the data filtration process, the datasets were marked for creating a book recommendation model comprising 720,212 rows, which contained user_id information for 14,612 individuals and book_id data for 9,999 books.

Build and measure the performance of a recommendation model

To develop a recommendation model, the study leveraged a dataset comprising book ratings from users, a common approach for such systems. CF techniques, particularly k-Nearest Neighbour (k-NN) and Matrix Factorization, were implemented, along with a Model Combiner (k-NN + MF) to evaluate their performance through RapidMiner Studio 10.1. The rating.csv dataset was imported via the ReadCSV operator, and the attributes were then defined through the Set Role operator: ratings serve as labels, user_id as user identification, and book_id as item identification. The datasets were split into training data (90%) and test data (10%) using the Split Data operator. The modelling process was launched through the Recommenders extension operator, which was focused on Collaborative Filtering Item Recommendations.

The chosen algorithm was trained on the training dataset to generate a recommendation model. This model was then applied to the test data by providing each user with the top 15 book recommendations. Next, performance evaluation was conducted using the Performance Operator (Item Recommendation) by measuring metrics such as Area Under the ROC Curve (AUC); Precision at 5 (prec@5), which measures the proportion of relevant items among the top 5 recommendations; Precision at 10 (prec@10), which measures the proportion of relevant items among the top 10 recommendations; and Precision at 15 (prec@15), which measures the proportion of relevant items among the top 15 recommendations. Then, the Normalized Discounted Cumulative Gain (NDCG) was used to evaluate the ranking quality of the recommendations through the position of relevant items, with the Mean Average Precision

https://doi.org/10.18080/jtde.v12n3.976

(MAP) used to calculate the average precision for each query, and then the mean across all queries. The parameter modelling process in this study was implemented through the RapidMiner software.

Based on this study, the researchers proposed the development of a recommender model utilizing the k-Nearest Neighbour (k-NN) technique, which was divided into two formats as follows:

a) User-based focuses on users within the system who express their preferences for books in comparable manner, as demonstrated by Bošnjak *et al.* (2011) and Tang & Wen (2015). The user-based k-NN model leverages the concept that users with similar preferences or behaviours can provide valuable recommendations to one another by identifying and analysing similarities. The model can suggest to a user items that similar users had liked or interacted with, to enhance the overall user experience. The development of this model involves several key steps and operators for simulation purposes. Initially, user data was pre-processed to ensure its quality and relevance to calculate the similarity between users using various similarity measures, such as cosine similarity, Pearson correlation, or Euclidean distance. Once the similarities were determined, the k-NN algorithm was applied to identify the nearest neighbours for each user. The model then aggregated the preferences of these neighbours to generate personalized recommendations. Various operators in RapidMiner Studio facilitated the tasks, such as data normalization, similarity computation, model training, and evaluation, throughout this process.

b) The item-based approach took into account books that were consistently similarly preferred by multiple users, as demonstrated by various operators utilized in constructing the model (Bošnjak *et al.*, 2011; Tang & Wen, 2015). The model was constructed using similar operators to those used in the user-based k-NN model, emphasized the process of calculating the similarity between items rather than users, by focusing on item-to-item relationships. The model can recommend items to users based on the similarity of items that they have previously interacted with. The implementation in this study revealed that the practical application of item-based collaborative filtering techniques has highlighted the key steps, such as data pre-processing, similarity calculation, and recommendation generation. This approach was particularly useful in scenarios where item metadata was rich and user interactions were sparse in order to provide a robust alternative to user-based methods.

The study also proposed developing a recommender model utilizing the Matrix Factorization technique (Mihelčić *et al.*, 2012) for incorporating different operators.

https://doi.org/10.18080/jtde.v12n3.976

The matrix factorization technique is popular in recommender systems for predicting user preferences. The technique involves decomposing a large matrix into smaller, more manageable matrices, typically representing user-item interactions. These smaller matrices capture latent factors influencing user preferences, enabling more accurate predictions. It highlighted the process of selecting and tuning these parameters, such as the number of latent factors, regularization terms, and learning rates, which were crucial for improving the model's accuracy and efficiency. This visualization helps us understand how the matrix factorization model operates and how it can be fine-tuned to predict better user preferences based on the hidden patterns in the data.

Results

As in the study objectives, this study aimed to investigate the development of recommendation models utilizing k-NN, Matrix Factorization (MF), and a combined approach of k-NN and MF techniques in order to assess the algorithms' effectiveness. The study results from the implementation of these algorithms are illustrated in the following subsections.

Performance of the k-Nearest Neighbour (k-NN) algorithm

The study explored recommendation modelling employing the k-NN method, which groups data by considering the closest k neighbours. The k-NN operator used the Cosine Similarity method to calculate similarity values. In this method, an angle of 0° indicates high similarity between variables, while an angle of 90° signifies independence or no similarity at all (Bošnjak *et al.*, 2011; Tang & Wen, 2015). The rating.csv dataset imported into the program comprised 720,212 rows, encompassing information on 14,612 users, 9,999 book data entries, and book rating data ranked from 1 to 5 points. The researchers experimented with different values of k to identify the optimal model performance through random selection. The k values of 5, 15, 25, 35, 45, 55, 65, 75, 85, 95, 105, 125, 155, 175, and 205 were evaluated. Subsequently, the dataset was divided into training and testing data in three proportions: 90:10, 80:20, and 75:25. This approach ensured a comprehensive examination of recommendation modelling effectiveness while maintaining computational efficiency.

The user-based approach relies on user data within the system, which assessed preferences for analogous books. The outcomes derived from model processing shown in Table 2 were achieved by partitioning the dataset with a ratio of 90:10.

Table 2 shows the results from random testing to find the best k value for the User k-NN technique through a 90:10 data split. The study revealed a correlation between the k values and the model's performance, with performance tending to improve with higher k values. However, an interesting observation emerged regarding the accuracy at position 5 (prec@5),

https://doi.org/10.18080/jtde.v12n3.976

as there was a positive correlation between accuracy and k values, until reaching Scenario 6 where k was set to 55. In this context, *'position'* refers to the rank of items in the recommendation list, such as the top 5 recommendations. A '*scenario*' refers to a specific experimental setup or configuration used in the study to test the performance of the recommendation model under different conditions. At this point, the results show some repetitions, particularly noticeable in accuracy values at positions 10 and 15, which started repeating from Scenario 6 and Scenario 8, respectively. Similarly, the Normalized Discounted Cumulative Gain (NDCG) value gradually increases until Scenario 14, where it reaches the high point. Conversely, the mean average precision (MAP) shows a consistent increase until Scenario 14, which was slightly decreased in Scenario 15 compared to Scenario 14. Consequently, the researchers ceased randomizing k values after Scenario 15. Thus, the performance results for each k value indicate that Scenario 13, which corresponds to a k value of 155, produces the optimal performance.

Table 2. Optimal k-value results from random 90:10 testing

Metric	Value
AUC	0.812
prec@5	0.078
prec@10	0.068
prec@15	0.066
NDCG	0.327
MAP	0.064

Subsequently, the data splitting ratio was adjusted to 80:20 for the training and testing datasets to ensure consistency with the random k value used in the initial 90:10 split, as shown in Table 3.

Table 3. Optimal k-value results from random 80:20 testing

Metric	Value
AUC	0.889
prec@5	0.198
prec@10	0.146
prec@15	0.121
NDCG	0.437
MAP	0.119

Table 3 shows the results from random testing to find the best k value of the User k-NN technique in the 80:20 data split. The results show a consistent trend across experiments where the 80:20 ratio corresponds to higher k values. Consequently, the model's performance metric is significantly increasing, particularly the Area Under the Curve (AUC), with increasing values of k. Moreover, the accuracy at position 5 (prec@5) shows a positive

https://doi.org/10.18080/jtde.v12n3.976

correlation with the increment of k. However, when considering the results, after reaching Scenario 11 with a k value of 105, subsequent scenarios, such as Scenario 15, show declining results, with accuracy values at positions 10 and 15 showing signs of stagnation. Conversely, the Normalized Discounted Cumulative Gain (NDCG) values show a gradual improvement, culminating in repeated outcomes from Scenario 15 onwards. The Mean Average Precision (MAP) values show a similar pattern of gradual improvement until Scenario 15, after which there are recurring outcomes from previous scenarios. Consequently, the results show the best performance in Scenario 14, where the dataset was divided into an 80:20 ratio and k was set to 175 — this emerged as the preferred configuration.

Table 4 shows the result of the data analysis process where the ratio of partitioning the data into training and testing sets was adjusted from a 90:10 split to a 75:25 split. The adjustment was made using the same random seed to ensure consistency in data sampling between experiments. This practice helps maintain the randomness of the data split while allowing for reproducible results when the experiment is rerun or reviewed. Table 4 shows that by comparing the model's performance under different data splits by using the same random seed across different experiments, variations in model performance can be attributed more directly to changes in the training-testing split, leading to more reliable conclusions about the model's robustness and scalability.

Table 4. Optimal k-value results from random 75:25 testing

Metric	Value
AUC	0.933
prec@5	0.127
prec@10	0.105
prec@15	0.093
NDCG	0.392
MAP	0.082

The k-Nearest Neighbours (k-NN) algorithm was used to assess books based on user ratings. The dataset was partitioned in a 90:10 ratio, with 90% used for training and 10% for testing. Table 5 shows the results for the optimal k value from random 90:10 testing. This table provides key metrics that reflect the model's performance when applying the optimal k value. The k values were balanced between bias and variance to best generalize unseen data.

Table 5. Optimal k-value results from random 90:10 testing

Metric	Value
AUC	0.915
prec@5	0.154
prec@10	0.117
prec@15	0.098

https://doi.org/10.18080/jtde.v12n3.976

Metric	Value
NDCG	0.396
MAP	0.093

After optimizing the k value for the User k-NN technique, the focus shifted to evaluating the Item k-NN technique. Table 6 shows the random optimal k value evaluation results in the Item k-NN technique. The analysis across various test scenarios indicates that Scenario 2, which sets k at 6, provides this study's best results. These new scenarios were designed to evaluate the impact of different k values on the performance of the Item k-NN technique. The model's performance in this scenario is primarily assessed through the AUC, a key metric for evaluating classifier performance. The AUC shows a notable upward trend from Scenario 1 through Scenario 5 and highlights an improvement in model efficacy as the experiments progressed. However, while the AUC has improved, the other efficiency metrics declined from their original points in subsequent scenarios. This suggests a trade-off between the AUC and other performance indicators: while the model becomes better at distinguishing between classes (as shown in the rising AUC), it might lose efficiency in other areas due to overfitting or increased computational demand. Based on these observations, it can be inferred that using a data splitting ratio of 75:25, Scenario 2 with k = 6 significantly shows the most balanced and effective outcomes. This setting seems to optimize the trade-offs to achieve a high accuracy without excessively compromising on other performance aspects. This conclusion highlights the importance of tuning the k parameter to align with specific data distributions and objectives, demonstrating that even small adjustments can significantly impact the performance of k-NN models.

Table 6. Performance metrics across different k values

Scenario	K	AUC	PREC @5	PREC @10	PREC @15	NDCG	MAP	Execute Time
Scenario1	5	0.743	0.257	0.203	0.170	0.450	0.166	2.02
Scenario2	6	0.755	0.259	0.205	0.172	0.453	0.168	2.19
Scenario3	7	0.765	0.257	0.204	0.172	0.454	0.167	2.28
Scenario4	8	0.773	0.254	0.202	0.171	0.454	0.166	2.47
Scenario5	9	0.780	0.250	0.200	0.169	0.453	0.165	2.57

Performance of the Matrix Factorization model

Matrix Factorization involves decomposing a matrix into smaller matrices, and as a result, the outcomes of these matrices are equal to those of the original matrix. For instance, in the context of user-book ratings, the rating matrix (R) is decomposed into two matrices: the user matrix (X) and the book matrix (Y). The multiplication of these X and Y matrices yields the original rating matrix (R). This decomposition is particularly useful in areas like

https://doi.org/10.18080/jtde.v12n3.976

recommendation systems, signal processing, and data compression. The concept is to simplify a complex matrix into easier to analyze and manipulate components.

In the context of recommendation systems, such as a user-book rating system, matrix factorization assists in predicting unknown ratings by users on a set of books based on the available ratings. The rating matrix, *R*, represents each user's ratings of different books. This matrix is typically sparse, but its meaning might reflect most entries, because not every user has rated every book.

The decomposition involves breaking down the rating matrix into two lower-dimensional matrices, X and Y, often referred to as the user and book matrices, respectively. The user matrix X refers to the latent factors of the users, and the book matrix Y refers to the latent factors of the books. These latent factors include user preferences and book characteristics that cannot be observed directly.

The number of latent factors is much smaller than the number of users or books, and these factors contain the underlying patterns in the data. For instance, in a simple model, each row of X might correspond to a user affinity towards various genres, while each row of Y might correspond to the book genres.

The outcomes of X and the transpose of Y (denoted as X * Y^T, where Y is structured similarly to X) approximates the original matrix R. This outcome provides a prediction of the ratings from a user to a book based on the similar preferences of other users and the attributes of the books. The aim is to fill in the missing entries in the original ratings matrix with these predictions, providing personalized recommendations to users based on their preferences from the data.

Matrix factorization techniques typically were employed in optimization algorithms to investigate the best approximation of R by minimizing the difference between the actual ratings and the predictions. This process is regulated by parameters that control overfitting to assure that the model was generalized adequately to new data through common approaches, including Singular Value Decomposition, Non-negative Matrix Factorization, and Alternating Least Squares, each suited to different data types and applications.

The RapidMiner software incorporates a crucial parameter called '*Iteration*' to control the number of cycles the algorithm runs to reduce discrepancies between predicted outputs and actual data. This parameter is vital, as it directly influences the model's effectiveness by adjusting the weights within the user and book matrices, which is essential for predicting outcomes based on user preferences and item characteristics. The researchers did not adhere to a fixed iteration number in the study, but reached a consensus to choose a method where iteration values were selected randomly within a predetermined range to find the optimal

https://doi.org/10.18080/jtde.v12n3.976

setting for model accuracy instead. The iteration values tested were 5, 15, 25, 35, 45, 55, 65, and 75. This empirical approach tests each configuration to see which yields the best training speed and prediction accuracy balance. To evaluate the effectiveness of these iteration settings, the data was split into training and testing subsets with a 90:10 ratio. This split allows the model to learn from the majority of the data (training set) and then validate the learned patterns on the smaller portion (testing set), which helps assess the model's performance in simulating real-world operations. The results of these experiments are shown in Table 7. Typically, these results would show trends indicating whether higher iterations lead to better performance or if there are diminishing returns after a certain point. This analysis is fundamental in optimizing machine learning models for practical applications to ensure they are both efficient and effective.

Table 7. The random test results of the number of repetitions of the matrix factorization model that divides the dataset in a ratio of 90:10

Scenario	Iteration	AUC	PREC @5	PREC @10	PREC @15	NDCG	MAP	Execute Time
Scenario1	5	0.838	0.017	0.015	0.013	0.196	0.018	2.07
Scenario2	15	0.841	0.017	0.014	0.013	0.196	0.017	1.27
Scenario3	25	0.842	0.017	0.014	0.013	0.195	0.017	1.36
Scenario4	35	0.842	0.017	0.014	0.013	0.196	0.017	1.31
Scenario5	45	0.842	0.017	0.014	0.013	0.196	0.017	2.04
Scenario6	55	0.842	0.017	0.014	0.013	0.196	0.017	2.04
Scenario7	65	0.842	0.017	0.014	0.013	0.196	0.017	2.12
Scenario8	75	0.842	0.017	0.014	0.013	0.196	0.017	2.15

Table 8. Optimal k-value results from random 90:10 testing

Metric	Value
AUC	0.942
prec@5	0.198
prec@10	0.148
prec@15	0.123
NDCG	0.442
MAP	0.120

Table 8 shows the results of testing various iteration values of the matrix factorization model. The initial dataset was divided into a 90:10 ratio, and then was divided into a 75:25 ratio for training and testing (Table 9). The results reveal enhanced performance when setting higher values for latent factors: running the matrix factorization model for 15 iterations and setting the number of latent factors to 100 resulted in the model achieving a peak efficiency value, with an AUC of 0.916. The latent factors refer to the hidden features that represent underlying patterns in the data, and increasing their number can help the model capture more complex relationships. However, other efficiency metrics have declined compared to those in Scenario

https://doi.org/10.18080/jtde.v12n3.976

6, where a different number of latent factors was used. With the same latent factor set at 100 but only two iterations, exploration into other scenarios mirroring this configuration resulted in Scenarios 3, 9, and 12 showing high model efficiency. The comparative assessment of performance outcomes identified Scenario 12 as yielding the most favourable results, which was achieved by setting the iteration count to 5 and the number of latent factors to 100.

Table 9. The random test results of the number of repetitions of the matrix factorization model that divides the data set in a ratio of 75:25

Scenario	Iter-ation	num factor	AUC	PREC @5	PREC @10	PREC @15	NDCG	MAP	Execute Time
Scenario1	1	25	0.613	0.014	0.012	0.01	0.217	0.009	1
Scenario2	1	50	0.679	0.047	0.037	0.031	0.252	0.027	1.14
Scenario3	1	100	0.714	0.09	0.069	0.057	0.29	0.05	2.03
Scenario4	2	25	0.832	0.101	0.081	0.07	0.32	0.059	1.03
Scenario5	2	50	0.856	0.156	0.123	0.103	0.369	0.093	1.19
Scenario6	2	100	0.873	0.201	0.155	0.13	0.41	0.123	2.45
Scenario7	3	25	0.867	0.103	0.085	0.073	0.33	0.063	1.04
Scenario8	3	50	0.889	0.155	0.123	0.105	0.378	0.095	1.27
Scenario9	3	100	0.899	0.199	0.157	0.133	0.419	0.126	3.24
Scenario10	5	25	0.882	0.09	0.076	0.067	0.326	0.057	1.07
Scenario11	5	50	0.904	0.146	0.118	0.101	0.376	0.091	1.42
Scenario12	5	100	0.911	0.194	0.154	0.132	0.419	0.124	4.5
Scenario13	15	25	0.886	0.081	0.069	0.062	0.32	0.053	1.23
Scenario14	15	50	0.91	0.136	0.112	0.097	0.372	0.087	3
Scenario15	15	100	0.916	0.187	0.149	0.128	0.415	0.12	11.51

Performance of the Model Combiner (k-NN + MF) technique

The results from the integrated three-algorithms model were introduced as the Combiner Model, connecting the strengths of each component. The results show how the Model Combiner operator integrates these models, functioning in parallel and producing outcomes through a weighted average approach. The model's efficacy is evaluated against the default parameters and those parameters optimized through prior experiments.

Table 10 shows that the learning data set was partitioned alongside the test data set at ratios of 90:10, 80:20, and 75:25 to compare the results. Observations indicate that Scenario 1, adhering to default parameters and a 90:10 ratio, obtained notably lower performance metrics, including Accuracy at Positions 5, 10, and 15, compared to other scenarios. Conversely, Scenarios 2 to 4, wherein model variables from previous experiments are optimized, show improved performance in Scenario 2, which obtained the highest AUC value of 0.937, yet lags behind Scenario 4 in accuracy metrics. Conversely, Scenario 4, aligning with parameters derived in previous sections, emerges as the optimal configuration. Therefore, User k-NN is set to k = 205, Item k-NN to k = 5, and Matrix Factorization to iterations = 5 and number of latent factors = 100, alongside a 75:25 learning-test data split.

https://doi.org/10.18080/jtde.v12n3.976

Table 10. The experimental results of Model Combiner between User k-NN and MF according to the specified experiment

Scenario	Train /Test Data-set	K-user	K-item	Num Factor	Iteration	AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP	Execute Time
1	90/10	80	80	10	30	0.93	0.124	0.087	0.07	0.362	0.136	39.53
2	90/10	205	5	100	5	0.937	0.16	0.115	0.093	0.405	0.174	45.53
3	80/20	205	5	100	5	0.932	0.256	0.196	0.163	0.481	0.194	46.11
4	75/25	205	5	100	5	0.929	0.293	0.229	0.192	0.506	0.201	45.41

In conclusion, Scenario 4 offers the most favourable outcomes, emphasizing the significance of parameter optimization in enhancing model performance, as shown in Table 11.

Table 11. Comparison of performance results across different scenarios

Scenario	Data Split	Parameters	AUC	Accuracy @ Position 5	Accuracy @ Position 10	Accuracy @ Position 15
1	90:10	Default Parameters	Lower	Lower	Lower	Lower
2	80:20	Optimized Parameters from previous experiments	0.937	Higher	Higher	Higher
3	75:25	Optimized Parameters from previous experiments	Higher	Higher	Higher	Higher
4	75:25	User k-NN: k=205, Item k-NN: k=5, Matrix Factorization: Iteration=5, Latent Num Factor=100	Highest	Highest	Highest	Highest

Comparative analysis of model performance

The experimental findings above establish a book recommendation model derived from the goodbooks-10k dataset. Employing the Model Combiner technique yields optimal performance outcomes by integrating three techniques: Item k-NN, User k-NN, and Matrix Factorization. The most efficient settings were ascertained through rigorous parameter configuration across these methodologies. Consequently, the model integration technique emerged as the top performer among all methodologies. This technique entails a 75:25 ratio allocation for learning and test data, alongside parameter specifications for each of the three constituent techniques: the User k-NN technique is parameterized with k = 205, the Item k-NN technique with a k value of 5, and the Matrix Factorization technique with an iteration value of 5 and a hidden factor number of 100. Under these settings, the model achieves the

https://doi.org/10.18080/jtde.v12n3.976

following performance metrics: AUC = 0.942, prec@5 = 0.198, prec@10 = 0.148, prec@15 = 0.123, NDCG = 0.442, and MAP = 0.120, as shown in Table 12. This comprehensive analysis highlighted the efficacy of the model integration technique through parameter optimization across multiple methodologies for superior performance in book recommendation systems.

Table 12. The summary of model performance measure results

Model	AUC	PREC@5	PREC@10	PREC@15	NDCG	MAP
User k-NN	0.889	0.198	0.146	0.121	0.437	0.119
Item k-NN	0.933	0.127	0.105	0.093	0.392	0.082
Matrix Factorization	0.915	0.154	0.117	0.098	0.396	0.093
Model Combiner	0.942	0.198	0.148	0.123	0.442	0.120

Table 12 shows a detailed comparative analysis to evaluate the effectiveness of various recommendation system models, employing the techniques k-NN for both user-based and item-based approaches, matrix factorization, and a hybrid approach that combines these methods. The analysis was based on how well each model predicted user preferences, leveraging the goodbooks-10k dataset—a popular dataset for benchmarking recommendation systems. The key findings highlight that the hybrid model demonstrates superior performance when compared to the individual models (Ngoendee & Charoenruengkit, 2020). This enhancement in performance is attributed to the synergistic effects that arise when these techniques are integrated, and each approach compensates for the weaknesses of the others; for example, matrix factorization can address the sparsity problem in user-item matrices, which is a limitation in k-NN approaches.

Additionally, the study involved experimentally determining the optimal parameters for each recommendation technique. This involved tuning parameters such as the number of neighbours in k-NN models and the regularization and factorization features in matrix factorization models. Identifying these optimal configurations was crucial for enhancing the overall accuracy and efficiency of the recommendation systems. The research indicates that a well-tuned hybrid recommendation model utilizes the strengths of user-based k-NN, item-based k-NN, and matrix factorization to deliver the most effective recommendations, promising optimal outcomes for users and businesses employing this model. This finding is significant, as it guides the development of more robust, accurate, and efficient recommendation systems in practical applications.

Conclusions

In this research, the researchers developed a book recommendation model focusing on Collaborative Filtering techniques, particularly k-NN and Matrix Factorization. The study

https://doi.org/10.18080/jtde.v12n3.976

aimed to enhance model performance by employing data mining techniques and optimizing parameters. The optimal configurations for user-based k-NN, item-based k-NN, and matrix factorization models each produced strong results. The Model Combiner, which integrated all three methods, achieved the highest AUC, 0.942. These results highlight the potential of hybrid models in improving the accuracy and efficiency of recommendation systems. Despite achieving modest accuracy metrics, the study emphasizes the importance of exploring alternative and hybrid methods to address common challenges like scalability and the cold-start problem. Future studies should focus on integrating advanced machine learning techniques, incorporating diverse data sources, and ensuring ethical considerations, such as user privacy and fairness, to enhance recommendation systems.

Acknowledgment

The authors deeply thank Khon Kaen University for its invaluable support. This work was generously funded by a scholarship under the Research Assistant Program at Khon Kaen University, Thailand (Grant No. RA2566-D203). We also sincerely thank Khon Kaen University for facilitating this research through Announcement No. 2580/2563 on "The Criteria for the Acceptance of an Inbound Visiting Scholar from a Foreign Institution or Organization to Khon Kaen University". This support has been crucial to the success of our research project.

References

Al-Ghuribi, S. M., Noah, S. A., & Mohammed, M. A. (2023). An experimental study on the performance of collaborative filtering based on user reviews for large-scale datasets. *PeerJ Computer Science*, *9*, e1525. https://doi.org/10.7717/peerj-cs.1525

Al-Hagery, M. A. (2020). A novel based-approach composed of clustering algorithm & cosine similarity for products recommendation. *International Journal of Education and Information Technologies*, *14*, 133–141. https://doi.org/10.46300/9109.2020.14.16

Ang, J. Y., & Haw, S. C. (2021). Comparative analysis of techniques used in book-based recommender system. In Proceedings of the 5th International Conference on Digital Technology in Education (pp. 87–92). ACM. https://doi.org/10.1145/3488466.3488475

Bharathi, Y. D. (2019). Recommendation system for video streaming websites based on user feedback. *International Journal of Engineering and Advanced Technology*, *8*(6), 1317–1320. https://doi.org/10.35940/ijeat.F8516.088619

Bharathipriya, C., Swathi, B., & Jency, X. F. (2020). Product recommendation framework based on customer review using collaborative filtering techniques. *Journal of Mechanics of Continua and Mathematical Sciences*, (Special Issue 7), 58–71. https://doi.org/10.26782/jmcms.spl.7/2020.02.00004

Bošnjak, M., Antulov-Fantulin, N., Šmuc, T., & Gamberger, D. (2011, June). Constructing recommender systems workflow templates in RapidMiner. In Proceedings of the 2nd RapidMiner Community Meeting and Conference (pp. 101–112). Shaker Verlag. https://www.shaker.eu/Online-Gesamtkatalog-Download/2024.08.10-08.53.51-66.249.66.88-radE74BB.tmp/3-8440-0093-3_INH.PDF

Chaturvedi, A., Subramanian, S. P., Kumar, A., & Mishra, B. (2023). Synergizing collaborative filtering and popularity scores in a machine learning approach for content suggestions. In 2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 442–449). IEEE. https://doi.org/10.1109/ICECA58529.2023.10395086

Devika, P. V., Jyothisree, K., Rahul, P. V., Arjun, S., & Narayanan, J. (2021, July). Book recommendation system. In 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1–5). IEEE. https://doi.org/10.1109/ICCCNT51525.2021.9579647

Foxtrot. (2024). Goodbooks-10k. Kaggle. https://www.kaggle.com/zygmunt/goodbooks-10k

Hikmatyar, M. (2020). Book recommendation system development using user-based collaborative filtering. *Journal of Physics: Conference Series*, *1477*(3), 032024. https://doi.org/10.1088/1742-6596/1477/3/032024

Ifada, N., Susanto, L. R., Saputro, P. A., Nugraha, A. P., Muflikhah, L., Fauzi, M. A., & Adinugroho, S. (2019). Enhancing the performance of library book recommendation system by employing the probabilistic-keyword model on a collaborative filtering approach. *Procedia Computer Science*, *157*, 345–352. https://doi.org/10.1016/j.procs.2019.08.176

Isinkaye, F. O. (2023). Harnessing Item Features to Enhance Recommendation Quality of Collaborative Filtering. *Journal of Applied Intelligent System*, *8*(2),162–172. http://dx.doi.org/10.33633/jais.v8i2.7915

Jain, A., & Vishwakarma, S. K. (2017). Collaborative filtering for movie recommendation using RapidMiner. *International Journal of Computer Applications*, *169*(5), 29–33. https://www.ijcaonline.org/archives/volume169/number6/jain-2017-ijca-914771.pdf

Khan, B. M., Khan, M. T., Malik, A. B., & Ahmad, K. S. (2017). Collaborative filtering based online recommendation systems: A survey. In 2017 International Conference on Information and Communication Technologies (ICICT) (pp. 125–130). IEEE. https://doi.org/10.1109/ICICT.2017.8320176

Kumar, P., Gupta, M. K., Rao, C. R. S., Bhavsingh, M., & Srilakshmi, M. (2023). A Comparative Analysis of Collaborative Filtering Similarity Measurements for Recommendation Systems. *International Journal on Recent and Innovation Trends in Computing and Communication*, *11*(3s), 184–192. https://doi.org/10.17762/ijritcc.v11i3s.6180

Li, Z., Xiao, Q., Liu, Y., Lin, Y., & Luo, F. (2023). BookGPT: A general framework for book recommendation empowered by large language model. *Electronics, 12*(22), 4654. https://doi.org/10.3390/electronics12224654

Mankar, K., Wankhade, P., Bahurupi, S., Jirapure, S., Deshmukh, A., & Thakare, S. (2023). Web based book recommendation system using collaborative filtering. In 2023

International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 1–6). IEEE. https://doi.org/10.1109/ESCI56872.2023.10099750

Mathew, P., Kuriakose, B., & Hegde, V. (2016). Book recommendation system through content-based and collaborative filtering method. In 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE) (pp. 47–52). IEEE. https://doi.org/10.1109/SAPIENCE.2016.7684166

Mihelčić, M., Antulov-Fantulin, N., Bošnjak, M., & Šmuc, T. (2012). Extending RapidMiner with recommender systems algorithms. Proceedings of the 3rd RapidMiner Community Meeting and Conference (RCOMM 2012). http://bib.irb.hr/datoteka/596976.rcomm2012_recommenders.pdf

Munaji, A. A., & Emanuel, A. W. (2021). Restaurant recommendation system based on user ratings with collaborative filtering. *IOP Conference Series: Materials Science and Engineering*, *1077*(1). https://doi.org/012008. 10.1088/1757-899X/1077/1/012026

Muneer, M., Khan, M. A., Manzoor, S., Naeem, M., & Saeed, H. (2022). Tour spot recommendation system via content-based filtering. In 2022 16th International Conference on Open Source Systems and Technologies (ICOSST) (pp. 1–6). IEEE. https://doi.org/10.1109/ICOSST57195.2022.10016820

Ngoendee, W., & Charoenruengkit, W. (2020). Book recommendation with data mining using RapidMiner [Master's project, Srinakharinwirot University]. Institutional Repository. http://ir-ithesis.swu.ac.th/dspace/bitstream/123456789/1232/1/gs591130027.pdf

Pasricha, H., & Solanki, S. (2019). A new approach for book recommendation using opinion leader mining. In Proceedings of the International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT 2018) (pp. 501–515). Springer Singapore. https://doi.org/10.1007/978-981-13-5802-9_46

Patel, D., & Patidar, H. (2018). Hybrid recommendation solution for online book portal. *International Journal for Research in Applied Science and Engineering Technology*, *6*(5), 1367–1373. http://doi.org/10.22214/ijraset.2018.5225

Raghavendra, C. K., & Srikantaiah, K. C. (2022). Switching hybrid model for personalized recommendations by combining users demographic information. *Journal of Theoretical and Applied Information Technology*, *100*(3), 825–835. http://www.jatit.org/volumes/Vol100No3/20Vol100No3.pdf

Rajalakshmi, S., Indumathi, G., Elias, A., & Priya, G. S. (2024). Personalized Online Book Recommendation System Using Hybrid Machine Learning Techniques. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(15s), 39–46. https://www.ijisae.org/index.php/IJISAE/article/view/4712

Rana, A., & Deeba, K. (2019). Online book recommendation system using collaborative filtering (with Jaccard similarity). *Journal of Physics: Conference Series*, *1362*(1). https://doi.org/012130. 10.1088/1742-6596/1362/1/012130

Rao, B., Bhargava, P., Panchal, D., Rane, S., & Lalwani, P. (2021). Book recommendation system with relevant text audiobook generation. *International Journal of Creative Research Thoughts (IJCRT)*, *42*(1), 398–404. https://ijcrt.org/papers/IJCRT2107170.pdf

Saravanapriya, M., Senthilkumar, R., & Saktheeswaran, J. (2022, July). Multi-label Convolution Neural Network for Personalized News Recommendation based on Social Media Mining. *Journal of Scientific and Industrial Research (JSIR)*, *81*, 785–797. http://op.niscair.res.in/index.php/JSIR/article/download/46261/465480990

Sarimehmetoğlu, B., & Erdem, H. (2023). Extracting Book Titles From Book Recommendation Videos Using a Deep Learning Approach. *MANAS Journal of Engineering*, *11*(2), 229–234. https://doi.org/10.51354/mjen.1369636

Sarma, D., Mittra, T., & Shahadat, M. (2021). Personalized book recommendation system using machine learning algorithm. *International Journal of Advanced Computer Science and Applications*, *12*(6), 212–219. https://doi.org/10.14569/IJACSA.2021.0120126

Singh, K. K., & Banerjee, I. (2023). Integrated personalized book recommendation using social media analysis. *Parikalpana KIIT Journal of Management*, *19*(1), 106–123. http://doi.org/10.23862/kiit-parikalpana/2023/v19/i1/220834

Tang, Z., & Wen, Z. (2015). Recommendation system based on collaborative filtering in RapidMiner. *Computer modelling & new technologies*, *18*(11), 1004–1008. http://www.cmnt.lv/upload-files/ns_19art162.pdf

Tian, Y., Zheng, B., Wang, Y., Zhang, Y., & Wu, Q. (2019). College library personalized recommendation system based on hybrid recommendation algorithm. *Procedia CIRP*, *83*, 490–494. https://doi.org/10.1016/j.procir.2019.04.126

You, Z., Hu, H., Wang, Y., Xue, J., & Yi, X. (2023). Improved Hybrid Collaborative Filtering Algorithm Based on Spark Platform. *Wuhan University Journal of Natural Sciences*, *28*(5), 451–460. https://doi.org/10.1051/wujns/2023285451

Zhao, W., Tian, H., Wu, Y., Cui, Z., & Feng, T. (2022). A New Item-Based Collaborative Filtering Algorithm to Improve the Accuracy of Prediction in Sparse Data. *International Journal of Computational Intelligence Systems*, *15*, a15 https://doi.org/10.1007/s44196-022-00068-7