Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12441
e-ISSN : 2541-2019
p-ISSN : 2541-044X

A proposed User-Based Approach for eBooks Recommendation Using a Weighted Nearest Neighbor Technique

Abdullah Mohammed Saleh[1)*], Alaa Yaseen Taqa[2)]
[1)]College of Computer Sciences and Mathematics, University of Mosul, Nineveh, Iraq
[1,2)] College of Education for Pure Sciences, University of Mosul, Nineveh, Iraq
[1)] abdullah.20csp39@student.uomosul.edu.iq [2)] Alaa.taqa@uomosul.edu.iq

Submitted : May 21, 2023 | **Accepted** : Jun 7, 2023 | **Published** : Jul 1, 2023

Abstract: Large book data stores were beneficial for our support systems but posed significant challenges for useful information retrieval. This issue was resolved by collaboratively filtering data depending on user needs. This study suggested a user-based methodology for recommending eBooks. The selected dataset was pre-processed, and Cross-validation was used to build a user-user similarity matrix. Three nearest neighbor algorithms (KNN Basic, KNN with Means and KNN with ZScore) were used, and weighted KNN was proposed for rating prediction. In this technique, the weight of each user was calculated based on its distance from the intended user. The evaluation process depends on the user-item matrix and user-user matrix for prediction. The proposed recommendation system was tested on the book-crossing dataset, and the results were evaluated using the root mean square error and the mean absolute value of error. The results show that the error rate of the proposed model is the lowest compared to the other methods used, specifically when using the Pearson-Baseline technique. Since the root mean square error is 1.647 and the mean absolute value of errors is 1.253. When using the cosine technique, the root mean square error is 1.742, and the mean absolute value of errors is 1.328.

Keywords: Book-Crossing, Collaborative filtering, eBooks recommender, KNN with Weight, Pearson correlation, User-Based.

INTRODUCTION

Book recommendation systems (BRMS) are becoming increasingly popular to help readers discover new books they may enjoy. These systems use algorithms to analyzed user data and suggest titles that match their interests. The aim is to provide personalized recommendations tailored specifically for each reader, making it easier for them to find the perfect book for their taste (Ricci, Rokach, & Shapira, 2022). Introducing a BRMS can be invaluable in any library or bookstore setting. Such a system helps customers quickly locate titles that suit their preferences, leading them towards more meaningful reading experiences and potentially expanding their literary horizons further than ever before possible (Resmi, Hermanto, & Ghozali, 2022). By allowing users access to detailed information about recommended books—including author bios, reviews from other readers, and ratings from trusted sources like Goodreads—a well-designed recommendation system can become an essential part of the customer experience at any library or store offering digital content services such as e-books and audiobooks. Successful implementation of this type of technology requires careful consideration when designing its architecture: What kind of data should be collected? How will it be used? What feedback mechanisms

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12441
e-ISSN : 2541-2019
p-ISSN : 2541-044X

should exist so users can refine results over time? Answering these questions correctly is critical to creating a compelling book recommendation engine that provides accurate results while maintaining privacy standards set forth by applicable laws and regulations on personal data protection.

LITERATURE REVIEW

A collaborative BRMS is a recommendation system that uses user preferences and ratings to suggest books to readers. It is a popular tool for online bookstores, libraries, and other organizations that need to recommend books to their users. This literature survey will review the various studies conducted on collaborative BRMS (Mathew, Kuriakose, & Hegde, 2016). One of the earliest studies on collaborative BRMS was conducted by Resnick et al. (1994). They developed a system called GroupLens, which used content-based and collaborative filtering techniques to recommend books to readers. The system was evaluated using a dataset of over 2,000 books and more than 1,000 readers. The results showed that the system could make accurate recommendations, with a recommendation accuracy of over 70% (Ahmed & Letta, 2023). In Thi Thanh Sang Nguyen (Sang Nguyen, 2019), Naive Bayes for book recommendation was implemented with acceptable runtime and accuracy. For classifier models, numeric and string types are inefficient. The word embedding method can be used to represent book titles better. Search engines, digital libraries, and e-commerce sites that sell books all need book RMS. Avi Rana and K. Deeba (Rana & Deeba, 2019) proposed a recommendation that uses Jaccard similarity to give more accurate recommendations using CF. Compact datasets proved more precise than complete datasets in the proposed algorithm. Yiu-Kai Ng (Ng, 2020) created a web application that suggests reading books for kids. They combined matrix factorisation and content-based methods to address the cold-start issue. This model made some grade-level predictions on the books too. Another study on collaborative BRMS was conducted by Adomavicius and Tuzhilin (2005). They developed a system called the Recommender System Toolkit (RST), which used a combination of content-based and collaborative filtering techniques. The system was evaluated using a dataset of over 1,000 books and more than 500 readers. The results showed that the system could make accurate recommendations, with a recommendation accuracy of over 80%.

Finally, a study was conducted by Dhiman Sarma and Tanni Mittra et al. (Sarma, Mittra, & Hossain, 2021); the clustering algorithms were used to improve the RMS prediction capacity. The datasets were obtained from Kaggle's Goodreads-books repository and processed by machine learning algorithms, including approximately 900,000 ratings of 10,000 books. Sensitivity, Specificity, and F1-Score were calculated for the proposed model's algorithms. The average sensitivity and specificity were 49.76% and 56.74%, respectively. Overall, the studies reviewed in this literature survey demonstrate that collaborative BRMS can effectively recommend books to readers. The studies also show that the accuracy of the systems can be improved by using more sophisticated algorithms and larger datasets.

Collaborative Recommendation System

A collaborative system is a type of system that uses the ratings and reviews of other users to generate recommendations (Ghannadrad, Arezoumandan, Candela, & Castelli, 2022).

- *User-Based Collaborative Filtering:* This type of collaborative filtering looks at the ratings and preferences of other users who are similar to the user. It then uses this information to recommend items that other similar users like (Hikmatyar & Ruuhwan, 2020).
- *Item-based Collaborative Filtering:* When using collaborative filtering, related products are suggested based on the items that the user has previously liked. It bases its recommendations on the evaluations and preferences of other users who have enjoyed such products (Tewaria, 2020).

Data collection

The dataset plays a significant role as it is given as input to the machine learning model, and output is predicted based on the data in the dataset. Generally, the dataset used in a BRMS would depend on the type of system being used. The collaborative filtering literature frequently makes use of this dataset. It was retrieved in 2004 in under four weeks via the Book-Crossing website, which is of ".csv" format:

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12441
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Books.csv, Ratings.csv, Users.csv (Ziegler, McNee, Konstan, & Lausen, 2005). The following is a description of dataset files:

- Books.csv: Contains the attributes (Book-Title, ISBN, Book-Author, Year-Of-Publication, Publisher, Image-URL-S, Image-URL-M Image-URL-L)
- Ratings.csv: Contains the attributes (User-ID, ISBN, Book-Rating)

Distributions of the number of ratings per book (Fig. 1) and user (Fig. 2) are strongly right-skewed, showing that most books/users have/are given few ratings. At the same time, there are some outliers with values very distant from the mean that are responsible for a long tail of the distribution.

- Users.csv: Contains the attributes (User-ID, Location, Age)

A total of 278,858 users have contributed 1,149,780 ratings for 271,379 books. The scale goes from 1-10. (10 is the highest). Researchers can expand the dataset by including book details (e.g., book summaries and reviews from other Websites). The Book-Crossing dataset is helpful for collaborative BRMS because it provides a large amount of data that can be used to create accurate recommendations (Pujahari & Sisodia, 2019). By analysing the ratings of books by different users, a recommendation system can learn the preferences of other users and make recommendations accordingly.

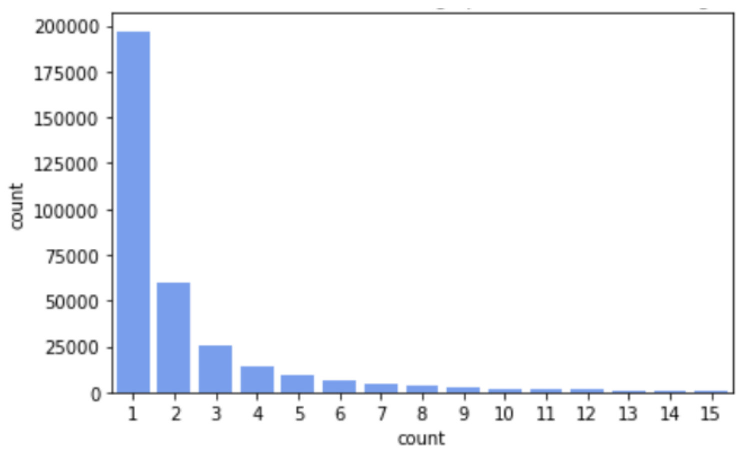

Fig.1 . Distribution of number of ratings per book (<=15 ratings)

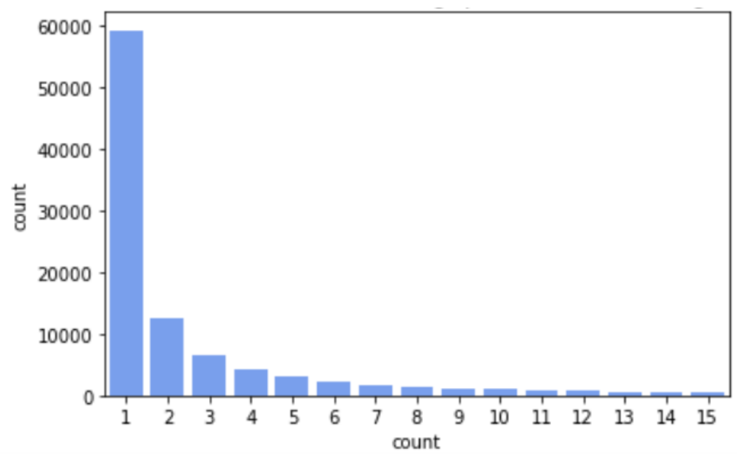

Fig. 2. Distribution of the number of ratings per user (<=15 ratings)

The dataset also provides information about the books, which can be used to determine the similarity between books. This can be used to create personalised recommendations that consider the user's preferences and the similarity between users (Hariadi & Nurjanah, 2018). Table 1 illustrates the description of the book-crossing dataset.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12441
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Table 1. Description of Book-Crossing Datasets

Features	Book-Crossing
Rating range	1-10
Demographics	Locations and ages
Metadata	Title, authors, year, publisher, and image of the cover
Description	N/A
Users	278,858
Items	271,379
Ratings	1,149, 780
Ratings/Users	4.123
Ratings/Items	4.236

Data Cleaning

Pre-processing the book-crossing dataset is essential in developing Collaborative BRMS. This process involves cleaning and organising the data, removing irrelevant information, and ensuring the data is in a suitable format for the system. Pre-processing the book-crossing dataset involves removing duplicates, correcting errors, and transforming the data into a format compatible with the system (Bhaskaran, Marappan, & Santhi, 2020). Additionally, the data must be normalised to compare different data points accurately. Once the data is pre-processed, the system can then use it to generate book recommendations. Pre-processing the book-crossing dataset is essential in developing Collaborative BRMS, as it ensures that the data is accurate and ready for the system to use (Saleh, Dharshinni, Perangin-Angin, Azmi, & Sarif, 2023).

Similarity Techniques

The computation of user similarities is one of the most important elements that significantly influences CF performance. Based on their past preferences or tastes, the users are thus represented as vectors of ratings. Comparing the corresponding vector similarities between two users describes their similarity. The module has functions for computing the similarity between two persons or items using a variety of metrics, including Pearson correlation, mean square difference, and cosine similarity. (Suryakant & Mahara, 2016)(Feng, Fengs, Zhang, & Peng, 2018).

1. Cosine similarity:

Under this method (Breese et al., 1998), a user is represented by his unique vector of ratings, and an item is represented by a vector of ratings rated by the set of users. The cosine metric gives the similarity value between two vectors that represent two users (or items)(Saeed & Taqa, 2022). A correlation between the two variables is strong if the value is near one. A number around 0 denotes the absence of a correlation (independent variables). The CF recommender system typically uses something that resembles it. Equation (1) defines the formula for the cosine similarity between two users u and v. (Suryakant & Mahara, 2016)(Fkih, 2021).

$$\mathrm{COS\_Sim}(i,j) = \frac{\overrightarrow{R_u} \times \overrightarrow{R_v}}{\|\overrightarrow{R_u}\| \times \|\overrightarrow{R_u}\|} = \frac{\sum_{i \in I_u \cap I_v} R_{ui} \times R_{vi}}{\sqrt{\sum_{i \in I_u \cap I_v} R_{ui}^2} \times \sqrt{\sum_{i \in I_u \cap I_v} R_{vi}^2}} \quad (1)$$

Where the sets of items rated by users u and v, respectively, are represented by Iu and Iv, and the set of items rated by both u and v is represented by IuIv. Users u and v, respectively, rated item I with the values Rui and Rvi. (Fkih, 2021).

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12441
e-ISSN : 2541-2019
p-ISSN : 2541-044X

2. Mean squared difference similarity:

The similarity metric may be off in some circumstances. Mean Squared Difference (MSD), in contrast to Jaccard, emphasises absolute ratings. Jaccard, another well-known CF metric, takes into account the number of objects that have been rated by two users rather than the ratings, meaning that the more co-rated items there are, the more similar the items are. The Jaccard and MSD formulas are shown in Equations (2) and (3), respectively (Fkih, 2021).

$$Jaccard\_Sim(u,v) = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} \quad (2)$$

$$MSD\_Sim(u,v) = 1 - \frac{\sum_{i\in I|_u \cap I_v|}(R_{ui} \times R_{vi})^2}{\sqrt{\sum_{i\in I_u \cap I_v} R_{ui}^2} \times \sqrt{\sum_{i\in I_u \cap I_v} R_{vi}^2}} \quad (3)$$

3. Pearson correlation similarity:

Karl Pearson devised this metric to evaluate linear correlations, and it quickly gained popularity in the statistical community. A number between -1 and 1 is produced using the PCC formula, with 1 denoting a high positive correlation, -1 a robust negative correlation, and 0 no connection. The following Equation (4) calculates the similarity between two users, u and v (Logesh, Subramaniyaswamy, Malathi, Sivaramakrishnan, & Vijayakumar, 2018):

$$PCC_{Sim(i,j)} = \frac{\sum_{i\in I_{uv}}(R_{u,i} - \overline{R_u}) \times (R_{v,i} - \overline{R_v})}{\sqrt{\sum_{i\in I_{uv}}(R_{u,i} - \overline{R_u})}^2 \times \sqrt{\sum_{i\in I_{uv}}(R_{v,i} - \overline{R_v})}^2} \quad (4)$$

Where Iuv denotes the set of items frequently rated by both u and v. The symbols (Ru) and (Rv) represent the mean ratings of users u and v for the item i in Iuv. R(u,i) and R(v,i) are user ratings for the same item i from users u and v (Logesh et al., 2018).

4. Pearson_Baseline correlation similarity:

By focusing on baselines rather than means, this measurement calculates the Pearson correlation coefficient between all pairs of users (or objects). The shrinkage parameter prevents overfitting when few ratings are provided. Equation (5) shows the pearson-baseline correlation coefficient formula(Suryakant & Mahara, 2016)(Fkih, 2021):

$$PCC\_Baseline\_Sim(u,v) = \frac{\sum_{i\in I_{uv}}(R_{u,i} - B_{u,i}) \times (R_{v,i} - B_{v,i})}{\sqrt{\sum_{i\in I_{uv}}(R_{u,i} - B_{u,i})}^2 \times \sqrt{\sum_{i\in I_{uv}}(R_{v,i} - B_{v,i})}^2} \quad (5)$$

B(u,i) is a baseline estimate that takes into consideration the user and item effects for an unknown rating R(u,i) as illustrated in Equation (6):

$$Bui = \mu + Bu + Bi \quad (6)$$

Bu and Bi parameters represent the observed and item-specific deviations from the average.

Evaluation Model

Evaluation metrics are used to assess learning algorithms and are a critical machine learning component. Because we are forecasting output in the recommendation, the error measurements are much different. The error metrics are produced as usual by comparing the model predictions to the actual values of the target variables and determining the average error. The proposed system's accuracy is often measured using two standards: mean absolute error (MAE) and root mean square error (RMSE) (Alita, Putra, & Darwis, 2021). The mean deviation Equation (7) is as follows:

$$\text{MAE} = \frac{\sum_{i=1}^{n}|p_i - q_i|}{n} \quad (7)$$

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

Which is the projected user rating, and which is the actual user rating? The lower the variation from the average number, the closer the recommendation algorithm's projected score is to the actual score. The root mean square error Equation (8) is as follows:

$$\text{RMSE} = \frac{\sum_{i=1}^{n}(p_i - q_i)^2}{n} \quad (8)$$

In which represents the test data, the size of the test data set, the user, the book, the user's actual score for the book, and the user's predicted score (Xia, Li, & Liu, 2020).

METHOD

In this research, a user-based model was proposed for e-books recommendation, as illustrated in Fig. 3, the selected dataset (book-crossing) was used, and pre-processing was performed on it. After that, the concept of Cross-validation, where the data was divided into three folds, every time two folds were used to build a user-user similarity matrix (using the cosine, mean squared difference (MSD), Pearson and Pearson_baseline correlation techniques) and the remaining Fold for testing.

Three nearest neighbour algorithms (KNN Basic, KNN With Means and KNN with Z-Score) were used for rating prediction (Sütçü, KAYA, & ERDEM, 2021); all of these algorithms are directly derived from an essential nearest neighbours approach (Mahmud, Hermanto, & Nugroho, 2023). To predict ratings, the KNN With Means considers the mean ratings of each user, whereas the KNN with z-score considers the z-score normalisation of each user. The basic KNN formula is illustrated in equation 9 (Shuxian & Sen, 2019).

$$\widehat{R_{ui}} = \frac{\sum_{v \in N_i^k(u)} R_{vi}}{k} \quad (9)$$

In this paper, KNN with weight was proposed to predict ratings. The weights were calculated for each user, similar to the required user, depending on the distance between them illustrated in Equation 10. The rating prediction Equation (11) was derived from the above basic Equation (9). The distance is computed as follows: $\text{Distance}(\text{u}, \text{v}) = 1 - \text{Similarity}(\text{u}, \text{v})$

$$Weight(v) = \frac{1}{Distance(u,v)^2} \quad (10)$$

$$\widehat{R_{ui}} = \frac{\sum_{v \in N_i^k(u)} W_v . R_{vi}}{\sum_{v \in N_i^k(u)} W_v} \quad (11)$$

Where *u* represent the active user, v is the similar user, *Rvi* is the rating of user *v* to item *i*, *N* is the set of similar users, *k* represents the number of users in *N*, and *Wv* is the importance of user *v* to user *u*.

As illustrated in Fig. 4, the user-user similarity matrix is continuously updated based on user behaviour (user-item interaction) to determine the nearest neighbours in the recommendation process. In contrast, the evaluation process depends on the user-item matrix (testing set/ available or actual ratings) and the user-user matrix for prediction.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12441
e-ISSN : 2541-2019
p-ISSN : 2541-044X

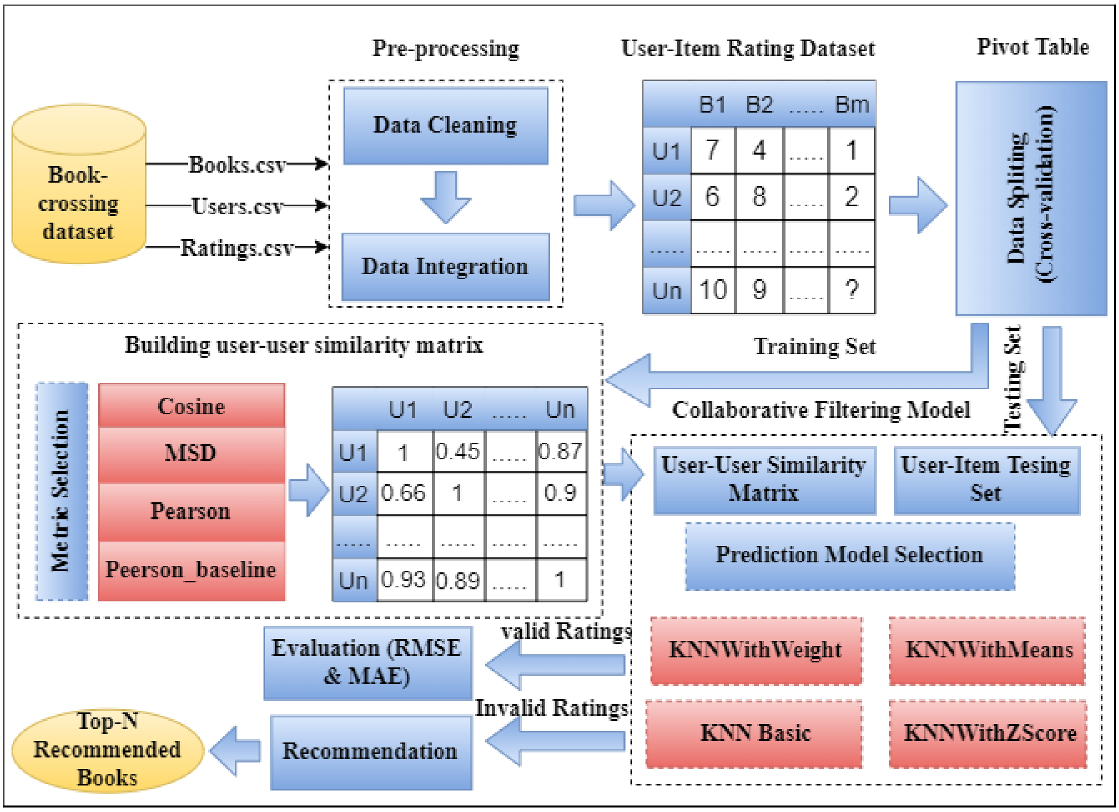

Fig. 3. The Proposed Model

The following algorithm shows the basic steps of the proposed user-based collaborative filtering model.

Proposed Algorithm:

1. Read dataset files (Users.csv, Books.csv, Ratings.csv) as U_m, B_n and R, respectively, where m is the number of users, and n is the number of books.
2. Pre-processing / **Cleaning** (Delete null values, records with low ratings, unnecessary characters (handling) and replace invalid data)
3. Data integration (build User-book matrix (UB_{mn}) from the above three files)
4. Dataset splitting into three parts using cross-validation
5. For all user's u_i in UB_{mn} **do**
6. For all users u_j in UB **do**
7. Compute u_i-u_j similarity using Cos-Sim(u_i,u_j)
8. Compute u_i-u_j similarity using MSD-Sim(u_i,u_j)
9. Compute u_i-u_j similarity using Pearson-Sim(u_i,u_j)
10. Compute u_i-u_j similarity using PearsonBL-Sim(u_i,u_j)
11. **End for**
12. Find the nearest neighbours (k) of the user u_i, , default k=40
13. Compute Rating Prediction of user u_i using (KNNwithMeans, KNNWithZscore, KNNBaseline and KNNwithweight)
14. End for
15. Evaluate the model using RMSE and MAE based on predicted ratings and actual ratings of user u_i
16. Recommend Top-N books that have the larger rating prediction to user u_i

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12441
e-ISSN : 2541-2019
p-ISSN : 2541-044X

RESULT

As shown in the proposed model, the training and testing process was done on the book-crossing data set after dividing it into 3 sections using cross validation. Table 2 shows the test results of the proposed model (KNN with weight) and other used models. In this paper RMSE and MAE measures were used to analyze and compare these models to determine the best fit for the data. In KNN with weight, when using centered cosine (pearson) baseline metric RMSE is 1.647 and MAE is 1.253. The second best method is KNN basic a special when using pearson similarity metric, where RMSE is 1.784 and MAE 1.330, 1.368 for pearcon and pearson-baseline respectively.

Table 2. Experimental Results

Similarity Techniques	Evaluation Metric	KNN with Weight	KNN Basic	KNN With ZScore	KNN With Means
Cosine	RMSE	1.742	1.78	1.801	1.813
	MAE	1.328	1.365	1.34	1.358
MSD	RMSE	1.710	1.797	1.847	1.835
	MAE	1.281	1.334	1.398	1.388
Pearson	RMSE	1.689	1.787	1.833	1.828
	MAE	1.269	1.330	1.389	1.382
Pearson Baseline	RMSE	1.647	1.784	1.846	1.844
	MAE	1.253	1.368	1.418	1.414

The chart in Fig. 4 illustrates the results based on RMSE, whereas Fig. 5 illustrates results based on MAE. Note from that the KNN with weight is best, when using similarity measures pearson-baseline, pearson, MSD and cosine. But in KNN With Means and KNN with Z-Score the cosine similarity measure was more accurate than MSD, pearson and pearson-baseline.

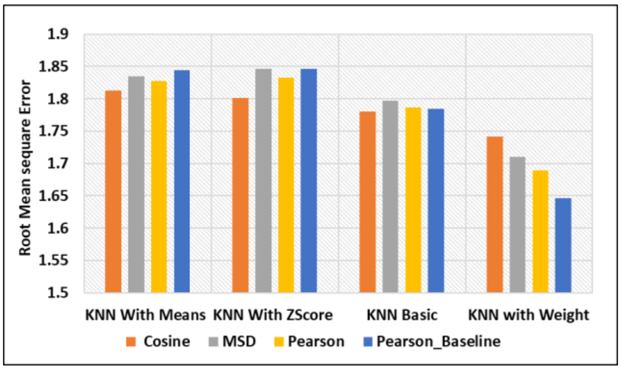

Fig. 4. Results evaluation using RMSE

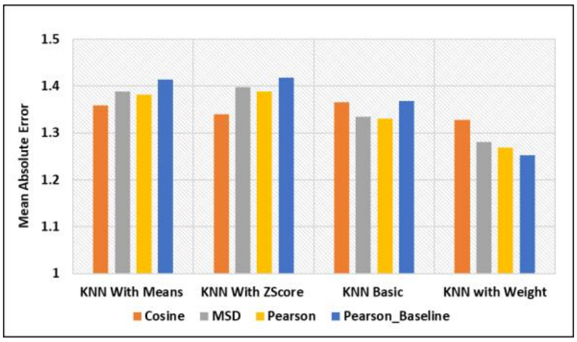

Fig. 5. Results evaluation using MAE

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12441
e-ISSN : 2541-2019
p-ISSN : 2541-044X

DISCUSSIONS

Based on the test results obtained in the implementation, it was found that the proposed method, which is derived from the nearest-neighbor method, is the most accurate type of technique used to implement memory-based recommendation models. As the RMSE when using centered cosine (pearson) baseline metric is 1.647 and MAE is 1.253. The second best method is KNN basic a special when using pearson similarity metric, where RMSE is 1.784 and MAE 1.330, 1.368 for pearcon and pearson-baseline respectively. It is possible to take advantage of these models when building recommendation systems for all applications, because they depend on the behavior (ratings) of user, and to increase the efficiency of the system, they can be hybridized with content-based models that solve the problem of cold start for new users.

CONCLUSION

Recommendation systems encourage people to make lifestyle choices take every day. E-books are among the richest and most diverse content, making choosing or recommending one difficult. And so, building a good recommendation system requires a good nomination form, An appropriate similarity metric and the use of a good evaluation metric to improve the ability of systems to predict which items are appropriate for users. Collaborative filtering techniques are among the most important techniques for solving the over-specialisation problem that content-based technologies suffer from. For that, this paper introduces memory-dependent collaborative filtering techniques derived from the basic KNN algorithm. Various accuracy measures (such as RMSE and MAE) were used to analyse and compare these models to determine the best fit for the data. In terms of ease and speed, the results showed that the error rate in the proposed weighted KNN model is the lowest compared to the other methods used, specifically when using the Pearson-Baseline technique. Since the RMSE is 1.647 and the MAE is 1.253. While using the cosine technique, the RMSE is 1.742, and the MAE is 1.328.

REFERENCES

Ahmed, E., & Letta, A. (2023). Book Recommendation Using Collaborative Filtering Algorithm. *Applied Computational Intelligence and Soft Computing*, *2023*, 12. doi:10.1155/2023/1514801

Alita, D., Putra, A. D., & Darwis, D. (2021). Analysis of classic assumption test and multiple linear regression coefficient test for employee structural office recommendation. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*. doi:10.22146/ijccs.65586

Bhaskaran, S., Marappan, R., & Santhi, B. (2020). Design and comparative analysis of new personalized recommender algorithms with specific features for large scale datasets. *Mathematics*. doi:10.3390/math8071106

Feng, J., Fengs, X., Zhang, N., & Peng, J. (2018). An improved collaborative filtering method based on similarity. *PLoS ONE*. doi:10.1371/journal.pone.0204003

Fkih, F. (2021). Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison. *Journal of King Saud University – Computer and Information Sciences*. doi:10.1016/j.jksuci.2021.09.014

Ghannadrad, A., Arezoumandan, M., Candela, L., & Castelli, D. (2022). Recommender Systems for Science: A Basic Taxonomy. *CEUR Workshop Proceedings*.

Hariadi, A. I., & Nurjanah, D. (2018). Hybrid attribute and personality based recommender system for book recommendation. *Proceedings of 2017 International Conference on Data and Software Engineering, ICoDSE 2017*. doi:10.1109/ICODSE.2017.8285874

Hikmatyar, M., & Ruuhwan. (2020). Book Recommendation System Development Using User-Based Collaborative Filtering. *Journal of Physics: Conference Series*. doi:10.1088/1742-6596/1477/3/032024

Logesh, R., Subramaniyaswamy, V., Malathi, D., Sivaramakrishnan, N., & Vijayakumar, V. (2018). Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method. *Springer Nature*. doi:10.1007/s00521-018-3891-5 ORIGINAL ARTICLE Enhancing

Mahmud, F. G., Hermanto, T. I., & Nugroho, I. M. (2023). IMPLEMENTATION OF K-NEAREST

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 8, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12441
e-ISSN : 2541-2019
p-ISSN : 2541-044X

NEIGHBOR ALGORITHM WITH SMOTE FOR HOTEL REVIEWS SENTIMENT ANALYSIS. *SinkrOn*, *8*(2), 595–602. doi:: https://doi.org/10.33395/10.33395/sinkron.v8i2.12214

Mathew, P., Kuriakose, B., & Hegde, V. (2016). Book Recommendation System through content based and collaborative filtering method. *Proceedings of 2016 International Conference on Data Mining and Advanced Computing, SAPIENCE 2016*. doi:10.1109/SAPIENCE.2016.7684166

Ng, Y. K. (2020). CBREC: A book recommendation system for children using the matrix factorisation and content-based filtering approaches. *International Journal of Business Intelligence and Data Mining*. doi:10.1504/IJBIDM.2020.104738

Pujahari, A., & Sisodia, D. S. (2019). Modeling Side Information in Preference Relation based Restricted Boltzmann Machine for recommender systems. *Information Sciences*. doi:10.1016/j.ins.2019.03.064

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

Resmi, M. G., Hermanto, T. I., & Ghozali, M. Al. (2022). Product Recommendation System Application Using Web-Based Equivalence Class Transformation (Eclat) algorithm. *SinkrOn*, *7*(3), 957–961. doi:https://doi.org/10.33395/sinkron.v7i3.11454

Ricci, F., Rokach, L., & Shapira, B. (2022). *Recommender Systems Handbook*. (F. Ricci, L. Rokach, & B. Shapira, Eds.) (Third). Springer Science. doi:10.1007/978-1-0716-2197-4

Saeed, A. A. M., & Taqa, A. Y. (2022). A proposed approach for plagiarism detection in Article documents. *SinkrOn*, *7*(2), 568–578. doi:10.33395/sinkron.v7i2.11381

Saleh, A., Dharshinni, N., Perangin-Angin, D., Azmi, F., & Sarif, M. I. (2023). Implementation of Recommendation Systems in Determining Learning Strategies Using the Naïve Bayes Classifier Algorithm. *Sinkron*. doi:10.33395/sinkron.v8i1.11954

Sang Nguyen, T. T. (2019). Model-based book recommender systems using Naïve Bayes enhanced with optimal feature selection. *ACM International Conference Proceeding Series*. doi:10.1145/3316615.3316727

Sarma, D., Mittra, T., & Hossain, S. (2021). Personalized Book Recommendation System using Machine Learning Algorithm. *International Journal of Advanced Computer Science and Applications*. doi:10.14569/IJACSA.2021.0120126

Shuxian, L., & Sen, F. (2019). Design and Implementation of Movie Recommendation System Based on Naive Bayes. *Journal of Physics: Conference Series*, *1345*(4). doi:10.1088/1742-6596/1345/4/042042

Suryakant, & Mahara, T. (2016). A New Similarity Measure Based on Mean Measure of Divergence for Collaborative Filtering in Sparse Environment. *Elsevier B.V.* doi:: 10.1016/j.procs.2016.06.099

Sütçü, M., KAYA, E., & ERDEM, O. (2021). Movie Recommendation Systems Based on Collaborative Filtering : A Case Study on Netflix İşbirlikçi Filtreleme Temelinde Film Öneri Sistemleri : Netflix. *Journal of Institue Of Science and Technology*, *37*, 367–376.

Tewaria, A. S. (2020). Generating Items Recommendations by Fusing Content and User- Item based Collaborative Filtering Item based Collaborative Filtering. *Elsevier B.V.*, *167*, 1934–1940. doi:10.1016/j.procs.2020.03.215

Xia, H., Li, J. J., & Liu, Y. (2020). Collaborative Filtering Recommendation Algorithm Based on Attention GRU and Adversarial Learning. *IEEE Access*, *8*, 208149–208157. doi:10.1109/ACCESS.2020.3038770

Ziegler, C.-N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving Recommendation Lists Through Topic Diversification. Chiba, Japan: International World Wide Web Conference Committee (IW3C2).

*name of corresponding author