Grenze International Journal of Engineering and Technology, January Issue

Book Recommendation System: Hybrid Approach

Gattu Vamshikrishna[1], Peruri Subhash[2], Kota Bigeswar[3] and Dr. P L Srinivasa Murthy[4]
[1-3]Student, Institute of Aeronautical Engineering, Dundigal, Hyderabad, India
Email: gattuvamshikrishna05@gmail.com, subhashperuri14@gmail.com, 21951a1217@iare.ac.in
[4]Professor, Institute of Aeronautical Engineering, Dundigal, Hyderabad, India
Email: plsrinivasamurthy@iare.ac.in

***Abstract*—A hybrid book recommendation system is designed to enhance user experience by providing personalized suggestions based on user preferences. This model integrates content-based and collaborative filtering techniques for improved recommendations. It addresses three scenarios: new users receive recommendations based on book rankings, returning users are served recommendations based on prior interactions, and diverse algorithms handle cases with no prior user interaction. The content-based approach utilizes cosine similarity for a book similarity matrix, while collaborative filtering employs a user-item matrix evaluated using the ALS algorithm. This system demonstrates an effective framework for generating tailored recommendations.**

***Index Terms*— recommender systems, content-based filtering, collaborative filtering, alternating least square.**

I. INTRODUCTION

In today's digital era, the plethora of available literature poses a challenge for readers to uncover books that align with their preferences. To tackle this issue, recommendation systems have emerged as vital tools. Two prominent approaches are content-based filtering and collaborative filtering. Content-based filtering examines intrinsic book characteristics, such as genre, authorship, and synopsis, to recommend similar books based on shared attributes. Collaborative filtering, on the other hand, utilizes user-item interactions like ratings and reviews to identify patterns of similarity between users or items, recommending books that are favoured by users with similar tastes. These approaches are often combined to produce a Hybrid recommendation system [1]. By combining the strengths of both methods, the system aims to provide highly personalized and diverse book recommendations that are tailored to individual user preferences.

The current era has witnessed a growing preference for online shopping, leading to a surge in e-commerce and a plethora of options for consumers. However, the abundance of choices can make it difficult to find suitable items. To address this challenge, recommender systems are beneficial in identifying behaviour patterns based on past history or user interactions. The primary challenge in this context is providing accurate recommendations [2][3]. This paper presents a model that analyse book content and user behaviour to surpass the limitations of traditional recommendation methods and enhance the quality of recommendations. By implementing various algorithms, the system aims to produce precise results. If there is a significant disparity between a user's interests and the recommendations offered, they may not use the system. Demographic techniques can generate personalized recommendations, but they may also raise privacy concerns. Alternatively, popularity-based recommendations can effectively generate high-quality suggestions.

Grenze ID: 01.GIJET.11.1.370

The results can be further refined through collaborative approaches, leading to improved output. Cold start problems are common in many recommendation systems. Therefore, offering top-n recommendations may help alleviate this issue to some extent.

We discussed there are three types of users. Firstly, the new user to deal with the cold start issue, we are recommending the books based on the popularity metrices. The second type of the users are those who do not prefer to rate books. this the case for failure of rating-based method. to tackle this kind of issues we are making book recommendations based on the similarity matrix for a given book ID. this will recommend the top similar books and it is crucial for offering users personalized suggestions based on their preferences. The third type of the user will give the ratings of their interests. We used the Alternating Least Square (ALS is algorithm and incorporates prediction functions for personalized recommendations) as main model.

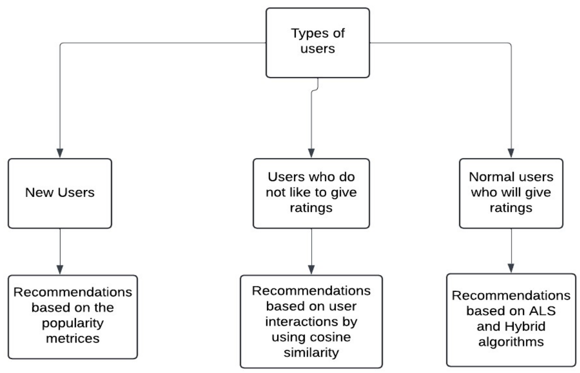

Fig. 1. Types of users.

Experimental results and performance evaluations are presented to demonstrate the effectiveness and superiority of the approach compared to baseline methods. To summarize the underlying approach, we are using hybrid model to provide personalized recommendations to individual user. This system is hybrid of content based as well as collaborative approach of recommender system. We are showing more accurate and more options that will increase the user experience and will raise the possibility of liking books.

II. LITERATURE SURVEY

This section will cover related research on recommender systems as well as the application of content-based and collaborative techniques. Following Netflix's launch of the competition in 2006, recommendation systems gained popularity [4]. Netflix ran a competition called the Netflix Prize from 2006 to 2009 in an effort to enhance its algorithm for suggesting movies. Netflix rewarded any team that could increase the accuracy of their current recommendation algorithm by 10% with a $1 million prize. Thousands of people from all over the world entered the competition, which significantly advanced collaborative filtering methods. The team BellKor's Pragmatic Chaos finally won the prize in 2009 after they were able to surpass the 10% improvement threshold. The concept of recommender systems, intelligent software agents, and personalized search engines has gained widespread acceptance among users who need help finding, organizing, categorizing, filtering, and sharing this enormous amount of data. By exploring the field of intelligent recommender agents on the Internet, the paper [5] addresses the increasing need for tools that help users manage the massive amounts of information that are available online. It highlights the value of recommender systems, intelligent software agents, and personalized search engines that enable users to efficiently find, organize, categorize, filter, and share information. The authors analyze 37 distinct systems and classify them into eight fundamental dimensions in order to present a thorough taxonomy of modern intelligent recommender systems. These dimensions form the basis of a taxonomy used to categorize the systems under study. It concludes cross-dimensional analysis, aiming to provide a framework for researchers to develop their own recommender systems, offering a structured starting point for further innovation in this field.

Recommendation systems come in a variety of forms with various concepts and methods. Recommendation systems have been implemented in a wide range of applications, such as media, e-commerce, transportation, healthcare, and agriculture. The state-of-the-art in recommendation systems today [6] includes a range of advanced methods intended to provide users with tailored content. These systems have a number of difficulties despite their efficacy, such as the data sparsity, scalability, and cold start problems. There are still restrictions like over-specialization and a lack of serendipity, which prevent users from discovering new interests by constantly recommending similar products. Recommendation systems are widely used in business; big businesses like Amazon, Netflix, and Spotify are using them to improve user engagement and satisfaction, which increases sales and customer loyalty. However, in order to process large amounts of data effectively, the implementation of these systems requires a significant amount of computational power and sophisticated algorithms.

Managing extensive datasets, guaranteeing scalability, and preserving recommendation precision as user preferences change are crucial obstacles in the creation and functioning of recommender systems [7][8]. To handle the volume and velocity of data, large datasets require effective data processing and storage solutions, which frequently make use of distributed computing frameworks like Apache Spark. Algorithms that can adjust to growing data volumes without noticeably degrading performance are said to be scalable. It is essential to employ strategies like parallel processing and effective memory management. Retaining recommendation accuracy over time requires iteratively updating models to account for the changing preferences of users.

III. Proposed Method

Our suggested model, a hybrid recommendation system, is made to get around the difficulty of producing precise recommendations. The initial screen will be a generic webpage with various books arranged according to the user's category. The option to receive recommendations for new or returning users is available to the user. The top books will be sent to the new user based on content-based systems and ranking. On the basis of their user ID, the previous user will receive customized recommendations. Based on the user's most recent interactions with the system, we will use hybrid methods and the alternating least square algorithm to generate these personalized recommendations and predictions. The book's details, including the author, publication year, rating, cover page, and publication, are visible to the user. These books are available for use. The Cosine similarity algorithm is being used to populate this section. The same algorithm—cosine similarity—is used to provide the search results.

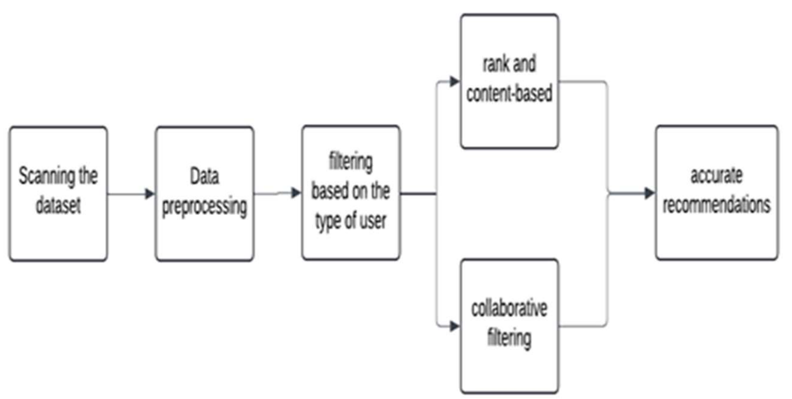

Fig. 2. Steps in the Algorithm

A. Scanning Dataset

In this step, the application searches the entire storage server while also cleaning the data, removing unnecessary information and retaining pertinent information for recommendations. Through the removal of erroneous, malicious, and missing data from the working data set, this process lowers the data sparsity.

B. Preprocessing of Data

In order to guarantee more accurate recommendations, this step also functions as the data correction portion. As per our application, this involves extracting only the data required for recommendations, i.e., books with a rating of more than four and authors with more than thirty books.

C. Filtering by ratings and user-id

The goal of this step is to provide users with the most relevant and optimal search results. The cosine similarity algorithm will return books based on factors such as user ID and rating.

D. Performing content-based filtering

The average rating and the total number of ratings for each book are determined in this step. These characteristics act as metrics to determine how well-liked a book is in our recommendation system. We select authors who have authored more than 30 books and have an average rating higher than 5. Our recommendation system is built on a high-quality dataset, which is ensured by this careful filtration. We generate a one-hot encoding for publisher, author, and year of publication for books. By using this encoding, categorical variables are converted into a format that can be used to measure similarity. We create a one-time encoding for book information like author, publication year, and publisher. This encoding converts categorical variables into a format that allows for similarity measurement. We proceed to measuring the similarity between books using cosine similarity. (Because most of the matrices are 0s and 1s, we will use cosine similarity (or distance) to determine the similarity between the books.) The resulting dataset, now enriched with these features, serves as the foundation for our future endeavors.

E. Performing collaborative filtering

Within the domain of recommendation systems, collaborative filtering is a potent method for offering users tailored suggestions. the User-Item Matrix construction, which are crucial stages in developing a collaborative filtering recommendation system that works. Based on user ratings, ALS (funk SVD) is used to identify users with similar book interests.

F. Serving Recommendations

Finally, we implement our hybrid book recommendation system, which combines both content-based and collaborative filtering approaches [9].

IV. ALGORITHM

A. Cosine Similarity

Cosine similarity [10] is the measurement of the similarity between two vectors. similarity is measured by calculating the cosine of the angle between them. We measure similarities based on the difference between a user’s rating for an item, and their average rating for all items, instead of measuring similarities between people based on their raw rating values.

$$\text{Cos Sim(X,Y)} = \frac{\sum_i (X)_i (Y_i)}{\sqrt{\sum_i {X_i}^2}\sqrt{\sum_i Y_i^2}} \qquad (1)$$

We substitute X with the average of all users' ratings for X sub-I and X bar, and Y with the average of all users' ratings for Y bar. The only thing that makes this approach different from traditional cosine similarity is that instead of focusing only on the raw rating, we are also examining the variance from the mean of each user's rating. Only when a person has rated a significant amount of content can we obtain a baseline or meaningful average of their ratings. Only then can you take the average. The cosine angle ranges from 0 to 1. When the cosine angle is near 1, it indicates a strong similarity between the items.

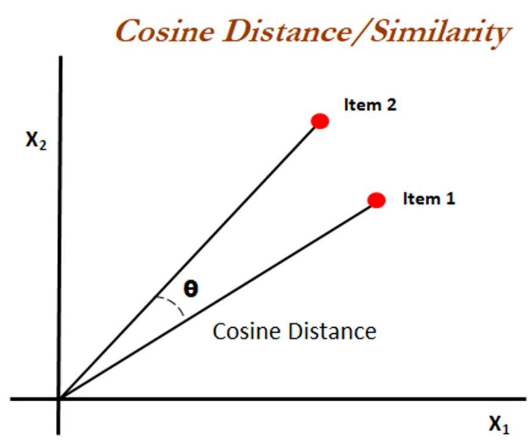

Fig. 3. Cosine Similarity

B. Matrix Factorization

Matrix Factorization (MF) [11], due to its significance as a model in Collaborative Filtering algorithms, numerous studies have been conducted to enhance conventional MF techniques. Variable decomposition, latent variable analysis, and dimension reduction have all recently attracted more attention from MF. Latent factor models form the foundation of most MF models. The rating matrix in latent factor is represented as the product of the item and user matrices. It is discovered that the MF technique reduces the sparsity issue the most accurately.

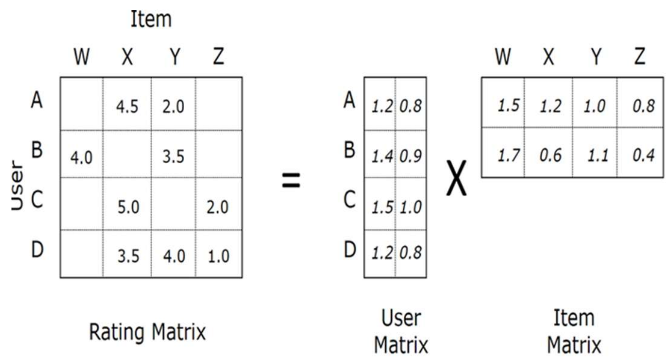

Fig. 4. Matrix Factorization

C. Alternating Least Square

A popular iterative optimization technique for collaborative filtering and matrix factorization tasks is Alternating Least Squares (ALS) [12]. It is especially well-liked for large-scale recommendation systems that use matrix factorization to find latent factors in interactions between users and items. A sparse user-item matrix is frequently factorized using ALS into two lower-rank matrices, one of which represents users and the other of which represents items.

In this user rate the items and the ratings data are used to predict to find ratings for other items. The rating data can be represented as an m × n matrix R where n users and m items. The u_i^{th} entry is r u_i in the matrix R which implies that u_i^{th} ratings for i^{th} item by u user. The R matrix is sparse matrix as items do not receive ratings from many users. Therefore, the R matrix has the most missing values. Matrix factorization is the solution of this sparse matrix problem. There are two k dimensional vectors which are referred to as "factors". x_u is k dimensional vectors summarizing's every user u. y_i is k dimensional vectors summarizing's every item i. Let,

$$r_{u_i} \approx x_u^T Y_i \quad (2)$$

$$x_u = x_1, x_2, x_{3,......,} x_n \in R^k \quad (3)$$

$$Y_i = Y_1, Y_2, Y_{3,......,} Y_n \in R^k \quad (4)$$

Equation 2 can be formulated as optimization problem to find minimum argument

$$\arg min \sum_{r_{u_i}} \left(r_{u_i} - x_u^T\right)^2 + \lambda \left(\sum_u ||x_u||^2 + \sum_i ||y||^2 \right)$$

Here,||x|| denotes the norm of a vector (Sqrt($\sum x_u^2$)), λ is the regularization factor, which is used to solve overfitting problem, is referred to as weighted-λ-regularization. The value of λ can be tuned to solve over-fitting whereas default value is 1.

ALS is an algorithm for matrix factorization and parallelization. Because we use a fairly large dataset in this study, we employ a low-rank matrix factorization method with ALS. Our objective in the matrix factorization problem is to identify a relatively small number that can approximate each book with a lower dimension book vector and each user with a lower dimension user vector. Factors are the name given to these vectors. By simply predicting the multiplication between the user vector and the transpose of the book vector, we can use these factors to predict our user rating for a given book. Therefore, we can create precise and effective rating predictions for recommender systems by applying the matrix factorization technique.

V. Result Precision

We have carried out a series of tests to evaluate the accuracy of book recommendations made to the user by our suggested recommender system. A personalized recommendation system has been developed by us, based on ratings, historical data, and user interests. The recommendation engine is operating on the database of book crossings. The components of the recommendation engine are hybrid, ALS, matrix factorization, and cosine similarity.

TABLE I. Performance metrics

		Cosine similarity	Matrix factorization	ALS	Hybrid
1	RSME	1.8102	1.4346	1.4391	1.4365
2	MAE	1.4932	1.2152	1.2269	1.2233

A. Root Mean Square

Root Mean Square Error (RMSE) puts more emphasis on larger absolute error and the lower the RMSE is, the better the recommendation accuracy.

$$RMSE = \sqrt{\frac{1}{N}\sum_{u_i}\left(p_{u_i} - r_{u_i}\right)^2}$$

Where P is the predicted rating for user p_{u_i} is the predicted rating for user u on item i, r_{u_i} is the actual rating and N is the total number of ratings on the item set.

B. Mean Absolute Error

MAE is the most popular and commonly used; it is a measure of deviation of recommendation from user's specific value. It is computed as follows

$$MAE = \frac{1}{N}\sum_{u_i}\left|p_{u_i} - r_{u_i}\right|$$

Where P is the predicted rating for user p_{u_i} is the predicted rating for user u on item i, r_{u_i} is the actual rating and N is the total number of ratings on the item set. The lower the MAE, the more accurately the recommendation engine predicts user ratings. MAE is comparatively lowest in Hybrid; hence accuracy is more.

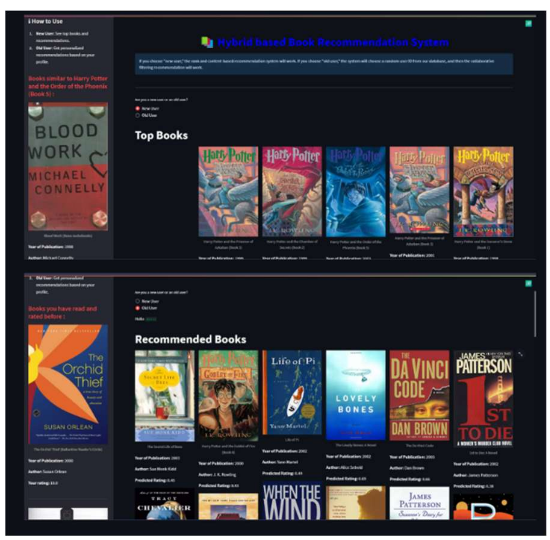

Fig. 5. Web interface displaying Hybrid Book Recommendations

In the above fig.5, to visually represent the functionality and effectiveness of the hybrid book recommendation system, a web-based interface showcasing the recommendation results was developed. The interface dynamically displays personalized book suggestions based on user interactions or book popularity for new users. Screenshots of the web interface, highlighting key features such as ranked recommendations and user-specific suggestions, are included in this section to provide a clear understanding of the system's output.

VI. Conclusion

A comprehensive strategy for raising user satisfaction and engagement in the book recommendation space is represented by the hybrid-based book recommendation system. The system uses book features and user preferences to combine collaborative and content-based filtering techniques to provide precise and customized recommendations. Adding a web application improves accessibility even more and makes it easier for users to find new books to read. All things considered, the system emphasizes how crucial it is to use sophisticated recommendation algorithms to enhance user experiences and encourage participation in the world of book exploration.

References

[1] Erion Çano and Maurizio Morisio “Hybrid Recommender Systems: A Systematic Literature Review” Intelligent Data Analysis Vol.21, No.6, pp. 1487-1524 (2017)

[2] Yonghong Tiana, Bing Zhenga , Yanfang Wanga , Yue Zhanga and Qi Wua “College Library Personalized Recommendation System Based on Hybrid Recommendation Algorithm” (2019).

[3] Qiang Yang, Zhixu Li, An Liu, Guanfeng Liu, Lei Zhao, Xiangliang Zhang, Min Zhang and Xiaofang Zhou "A novel hybrid publication recommendation system using compound information" (2020).

[4] James Bennett and Stanley Lanning “The Netflix Prize” Conference: In KDD Cup and Workshop in conjunction with KDD.

[5] Miquel Montaner, Beatriz López and Josep Lluís de la Rosa “A Taxonomy of recommendation Agents on internet”.

[6] Zeshan Fayyaz, Mahsa Ebrahimian, Dina Nawara, Ahmed Ibrahim and Rasha Kashef "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities.

[7] Xiaoyuan Su, Taghi M. Khoshgoftaar "A Survey of Collaborative Filtering Techniques",(2009).

[8] Marwa Hussien Mohamed, Mohamed Helmy Khafagy and Mohamed Hasan Ibrahim "Recommender Systems Challenges and Solutions Survey" (2019).

[9] Pitiwat Arunruviwat and Veera Muangsin "A Hybrid Book Recommendation System for University Library" (2022).

[10] Ramni Harbir Singh, Sargam Maurya, Tanisha Tripathi, Tushar Narula and Gaurav Srivastav" Movie Recommendation System using Cosine Similarity and KNN" International Journal of Engineering and Advanced Technology (IJEAT) (2020).

[11] Hafid Ahmad Adyatma and Z. K. A. Baizal"Book Recommender System Using Matrix Factorization with Alternating Least Square Method".

[12] Subasish Gosh, Nazmun Nahar, Mohammad Abdul Wahab, Munmun Biswas, Mohammad Shahadat Hossain and Karl Andersson "Recommendation System for E-commerce Using Alternating Least Squares (ALS) on Apache Spark".