Recommendation System for Digital Marketers

Ramya A[1], Mahalakshmi L[2]

[1]Student, Department of Computer Applications, Hindusthan College of Engineering andTechnology, Coimbatore.
[2]Assistant Professor, Department of Computer Applications, Hindusthan College of Engineering and Technology, Coimbatore, mahalogu@gmail.com

Abstract: This Recommendation system is built to recommend the book to customers based on their history of purchases. The Recommendation system comes under the Machine learning Process, which is divided into supervised and unsupervised learning, where this recommendation comes under the unsupervised machine learning. In unsupervised machine learning the recommendation engine is built using a Popularity based filtering algorithm along with a collaborative filtering algorithm. In the collaborative filtering algorithm the model-based approach is used where Cosine similarity is used as a metric machine learning and the products are recommended for the book.csv, user.csv, and ratings.csv datasets. The ML model is executed in the Jypyter notebook and the output is verified. Then a webpage for the user's interface is designed using HTML the user interface collects the inputs from the user's dataset and transfers it to the machine model for analysis and then returns its output to the user interface page itself. Thus, the Recommendation Engine built will recommend the book to each user based on their previous purchase records and thus improve the company sale. Thus, this project focuses on increasing the sale of the company by recommending the book which the customer may like to buy. This is how this Recommendation Engine works.

Keywords: Recommender System, collaborative algorithm, Cosine similarity.

I. Introduction

Recommender systems are an important class of machine learning algorithms that offer "relevant" suggestions to users. Categorized as either collaborative filtering or a content-based system, Practically, recommender systems encompass a class of techniques and algorithms that can suggest "relevant" items to users. Ideally, the suggested items are as relevant to the user as possible, so that the user can engage with those items: YouTube videos, news articles, online products, and so on. Items are ranked according to their relevancy, and the most relevant ones are shown to the user. The relevancy is something that the recommender system must determine and is mainly based on historical data. Recommender systems are generally divided into two main categories: collaborative filtering and content-based systems. Pictorial representation of types of recommendation system is shown in Figure 1.

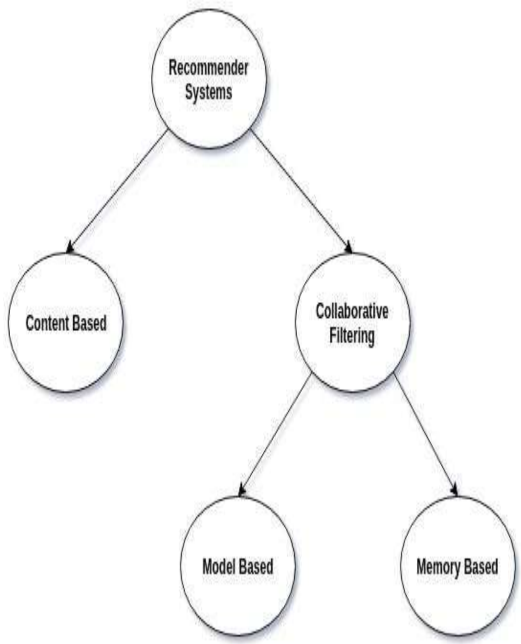

Figure 1. A tree of the different types of Recommender Systems

Collaborative Filtering Systems

Collaborative filtering methods for recommender systems are methods that are solely based on past interactions between users and the target items. Thus, the input to a collaborative filtering system will be all historical data of user interactions with target items. This data is typically stored in a matrix where the rows are the users, and the columns are the items. The core idea behind such systems is that the historical data of the users should be enough to make a prediction. i.e we don't need anything more than that historical data, no extra push from the user, no presently trending information, etc.

This project is based on the collaborative filtering system, where the data sets of the Book, User, Rating are used to build the recommendation engine using the Cosine similarity algorithm, the dataset is tested and trained and the recommendation model is built and books are recommended based on the user history of purchases.

Problem Domain

Recommender systems are an inseparable part of any medium-sized or big e-commerce website. The system's role is to suggest relevant new content to help users find exactly what they seek. A good recommendation boosts sales, AOV, and conversion rates, because it generates automatic content advice based on the user's preference. To make an informed suggestion, a recommendation system has to understand as much as possible about the user's needs.

Recommending books using a Machine learning algorithm is the main goal of this project. Books are recommended by the similarity model and we are going to train and build using various features such as user ratings, book descriptions, book titles, etc. The system groups users into clusters so that each data point within the cluster is similar and dissimilar to the data point in the other cluster. The system we would like to develop will also be able to find an average rating for each cluster and it is going to find top-rated books for users from each cluster. All these books shortlisted by the system will be used for the training model. The prediction model needs to be trained to produce better results.

The next stage is narrowing down the data by ranking the candidates. Optimizing the process as shown in the above image is what makes a recommendation successful. This is a task for artificial intelligence. The most precise recommendation systems utilize self-learning models that register, analyze and interpret everything there is to know about user preferences.

Machine learning algorithms pave the way for personalized recommendations. Machine learning, a subset of artificial intelligence, is a process through which a system explores patterns and connections occurring in vast historical data volumes. This way it can delve deep into complex matters, such as human behavior, and understand them better. To produce personalized content, recommendation systems must be trained by algorithms.

I. Algorithm used

Collaborative filtering method

The algorithm used in this recommendation system is a collaborative filtering model.

Collaborative filtering recommender systems have played a significant role in the rise of web services and content platforms like Amazon, Netflix, YouTube, etc. in recent years. In this age of information, knowing what the customer wants before they even know it themselves is nothing short of a superpower. As the name suggests, recommender system algorithms are used to offer relevant content or product to the consumer based on their taste or previous choices.

Collaborative filtering solely uses past interactions between the customers and the products they've used to recommend new items. Item features are not important since user-item interactions are used and are stored in the user-item interactions matrix. In collaborative filtering, all the users are taken into consideration and people with similar tastes and preferences are used to suggest new and specific products to the primary customer. It helps companies and customers keep up with what's trending.

II. Objective

The main objective to build this recommendation system is to recommend books to the user based on their interest, it also saves the time of the user by surfing unliked or uninterested products. To recommend similar products to the users based on their past interactions and ratings with the books. The website shows the top 50 books on the homepage and also has a listing page that contains a list of the books. The factorization matrix is built and the product is reduced based on the user's interest and only the product liked by the customers is recommended to the user. The purpose of recommenders is often summarized as "help the users to find the relevant items", and the predominant operationalization of this goal has been to focus on the ability to numerically estimate the users' preferences for unseen items or to provide users with item lists ranked in accordance to the estimated preferences.

III. Scope

This recommendation engine gives the solution to the customers by showing relevant book recommendations in real time. Powerful data filtering tools and recommendation systems use algorithms and data analysis techniques to recommend the most relevant products/items to a particular user. The main aim of any recommendation engine is to stimulate demand and actively engage users. Primarily a component of an eCommerce personalization strategy, recommendation engines dynamically populate various products onto websites, apps, or emails, thus enhancing the customer experience. These kinds of varied and omnichannel recommendations are made based on multiple data points such as customer preferences, past transaction history, attributes, or situational context. Recommendation engines today serve as the key to the success of any online business. But, for a sound recommendation system to make relevant recommendations in real-time requires powerful abilities to correlate not just the product but also customer, inventory, logistics, and social sentiment data. All recommender systems can be powerful tools for any e-commerce business, and rapid future developments in the field will increase their business value even further. With a wide range of business applications, including anticipating seasonal purchases based on recommendations, determining essential purchases, and offering better suggestions to customers, brands can leverage recommender systems for two key areas brand loyalty and enhanced customer retention.

IV. System analysis

Existing System

The Existing book recommendation engines recommend the top-stared books on the website and engines make use of conventional algorithms for recommendations. A Content-based Recommendation Engine, the system generates recommendations from the source based on the features associated with products and user information. Content-based recommenders treat recommendation as a user-specific classification problem and learn a classifier for the user's likes and dislike based on product features.

Drawbacks

- The products recommended were the same for all users and some customers may not like that recommendation
- The best products were recommended by some overrated books without the user's preference.

Proposed System

All the drawbacks in the existing system have been overcome in the proposed system. This proposed system is based on Collaborative Filtering Model. Recommendation engines and suggestions are generated based on ratings given by the group of people. It locates peer users with a rating history similar to the current user and generates recommendations for the user.

Advantages

- The products recommended are different from user to user. It analyses the user's preference for recommending books.
- The products recommended are based on the user's history, So there is a high chance for the customer to purchase that book. That increases the rate of purchase.
- It saves time for the user by recommending users interested in the book

V. Module description

Collaborative Filtering Method

The algorithm used in this recommendation system is a collaborative filtering model. Collaborative filtering recommender systems have played a significant role in the rise of web services and content platforms like Amazon, Netflix, YouTube, etc. in recent years. In this age of information, knowing what the customer wants before they even know it themselves is nothing short of a superpower.

Collaborative filtering solely uses past interaction between the customers and the products they've used to recommend the new item. Item features are not important since user-item interactions are used and are stored in the user-item interaction matrix. In collaborative filtering, all the users are taken into consideration and people with similar tastes and preferences are used to suggest new and specific products to the primary customer. It helps companies and customers keep up with what's trending.

User-item interaction matrix

In collaborative filtering, we ignore the features of an individual item. Instead, we focus on a similar group of people using the item and recommend other items that the group likes. Similar users are divided into small clusters and are recommended new items according to the preferences of that cluster. Let’s understand this with an easy product recommendation

Table 1: User-Item Matrix

USERS	PRODUCT 1	PRODUCT 2	PRODUCT 3	PRODUCT 4
USER 1	5	4		5
USER 2	4		3	
USER 3		1		2
USER 4	1	2		

The infer from this user-item matrix is shown in table 1:

- Users 1 and 2 liked Product1. Since User 1 liked products 2 and 4 a lot, there's a high chance of User 2 enjoying the same.
- Users 1 and 3 have opposite tastes.
- Users 3 and 4 both disliked Product 2, so there's a high chance User 4 will also dislike Product 4.
- User 3 might dislike Product 1.

This is the logic behind employing a user-item interaction matrix - to find clusters of similar users through collaborative filtering.

Types of collaborative filtering

The two types of collaborative filtering approaches are:

1. Memory-based collaborative approach
2. Model-based collaborative approach

The collaborative approach used in this recommendation system is memory-based, in model-based we have used Cosine similarity to find the user's ratings and rated items, under factorization.
1)Memory-based collaborative approach

In memory-based collaborative filtering, only the user-item interaction matrix is utilized to make new recommendations to users. The whole process is based on the user's previous ratings and interactions. Memory-based filtering consists of 2 methods: user-based collaborative filtering and item-based collaborative filtering.

2) Model-based collaborative approach

In the model-based approach, machine learning models are used to predict and rank interactions between users and the items they haven't interacted with yet. These models are trained using the interaction information already available from the interaction matrix by deploying different algorithms like matrix factorization, deep learning, clustering, etc.

Matrix factorization

Matrix factorization is used to generate latent features by decomposing the sparse user-item interaction matrix into two smaller and dense matrices of user and item entities. Since not all the products are viewed and rated by every user, we end up with a sparse matrix. To create a model for our matrix, we can assume that:

- There exist some latent features that can differentiate between good and bad movies.
- These features can help us understand user choices (the higher the value, the higher the preference).

We do not provide these features explicitly, but let the model discover the useful features and make its user and item matrices. As the features are learned and not provided, they have mathematical correlation and meaning but no intuitive understanding.

Cosine Similarity

Cosine similarity is used as a metric in different machine learning algorithms like the KNN for determining the distance between the neighbor, in recommendation systems, it is used to recommend movies with the same similarities and for textual data, it is used to find the similarity of texts in the document Hamming distance can be used as a metric for KNN, recommendation systems, and textual data. But hamming distance considers only the character type of data of the same length but cosine similarity can handle variable length data. When considering textual data the Hamming distance would not consider the frequently occurring words in the document and would be responsible for yielding a lower similarity index from the text document while cosine similarity considers the frequently occurring words in the text document and will help in yielding higher similarity scores for the text data. Cosine similarity in machine learning can be used for classification tasks wherein it can be used as a metric in the KNN classification algorithms to find the optimal number of neighbors and also the KNN model that is fitted can be evaluated against different classification machine learning algorithms and can be used to evaluate various performance parameters like the accuracy score, AUC score, and the classification report can also be obtained to evaluate other parameters like precision and recall.

knn_model=KNeighborsClassifier(metric='cosine')

VI. Methodology

Collaborative filtering is the most common technique used when it comes to building intelligent recommender systems that can learn to give better recommendations are more useful information about the user is collected. To experiment with recommendation algorithms, you'll need data that contains a set of items and a set of users who have reacted to some of the items.

	i_1	i_2	i_3	i_4	i_5
u_1	5		4	1	
u_2		3		3	
u_3		2	4	4	1
u_4	4	4	5		
u_5	2	4		5	2

Figure 2: User Matrix

The matrix shown in Figure 3 shows five users who have rated some of the items on a scale of 1 to 5. For example, the first user has given a rating of 4 to the third item. In most cases, the cells in the matrix are empty, as users only rate a few items. It's highly unlikely for every user to rate or react to every item available. A matrix with mostly empty cells is called **sparse**, and the opposite to that (a mostly filled matrix) is called **dense**. Here we used Model-based approaches, which involve a step to reduce or compress the large but sparse user-item matrix.

Dimensionality Reduction

In the user-item matrix, there are two dimensions:

1. The number of users
2. The number of items

If the matrix is mostly empty, reducing dimensions can improve the performance of the algorithm in terms of both space and time.

Cosine Similarity

Cosine similarity is a metric, helpful in determining, how similar the data objects are irrespective of their sizeIn cosine similarity, data objects in a dataset are treated as a vector. The formula to find the cosine similarity between two vectors is – Cos(x, y) = x.y / ||x|| * ||y|| where,

- **x . y** = product (dot) of the vectors 'x' and 'y'.
- **||x||** and **||y||** = length of the two vectors 'x' and 'y'.
- **||x|| * ||y||** = cross product of the two vectors 'x' and 'y'.

For example to find the similarity between two vectors – **'x'** and **'y'**, using Cosine Similarity.

The 'x' vector has values, **x = { 3, 2, 0, 5 }**

The 'y' vector has values, **y = { 1, 0, 0, 0 }**

The formula for calculating the cosine similarity is: **Cos(x, y) = x.y / ||x|| * ||y||**

x . y = 3*1 + 2*0 + 0*0 + 5*0 = 3

||x|| = √ (3)^2 + (2)^2 + (0)^2 + (5)^2 = 6.16

||y|| = √ (1)^2 + (0)^2 + (0)^2 + (0)^2 = 1

∴ Cos(x, y) = 3 / (6.16 * 1) = 0.49

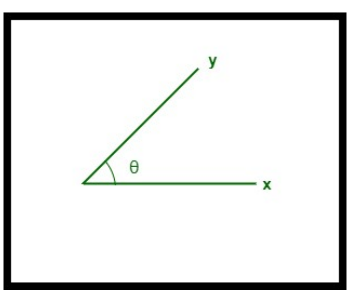

Figure 3: Cosine similarity between two vectors

The dissimilarity between the two vectors ‘x’ and ‘y’ is given by –

$\therefore$ Dis(x, y) = 1 - Cos(x, y) = 1 - 0.49 = 0.51

- The cosine similarity between two vectors is measured in ‘θ’.
- If $\theta = 0°$, the ‘x’ and ‘y’ vectors overlap, thus proving they are similar.
- If $\theta = 90°$, the ‘x’ and ‘y’ vectors are dissimilar.

This similarity is shown in Figure 3.

Thus the engine is built using a collaborative filtering model, the recommendation, commands are the machine learned to recommend books based on the cosine similarity algorithm, thus the engine recommends items when the user id is given as input. The outcome of the project is shown in Figure 4,5and 6.

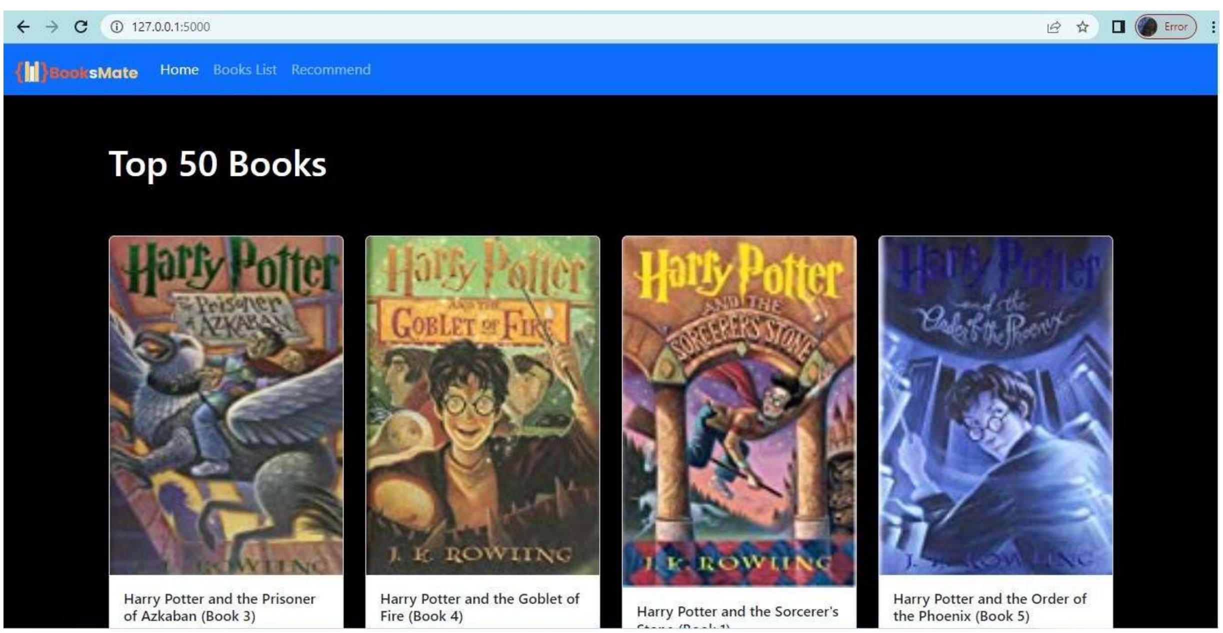

Figure 4 Home page

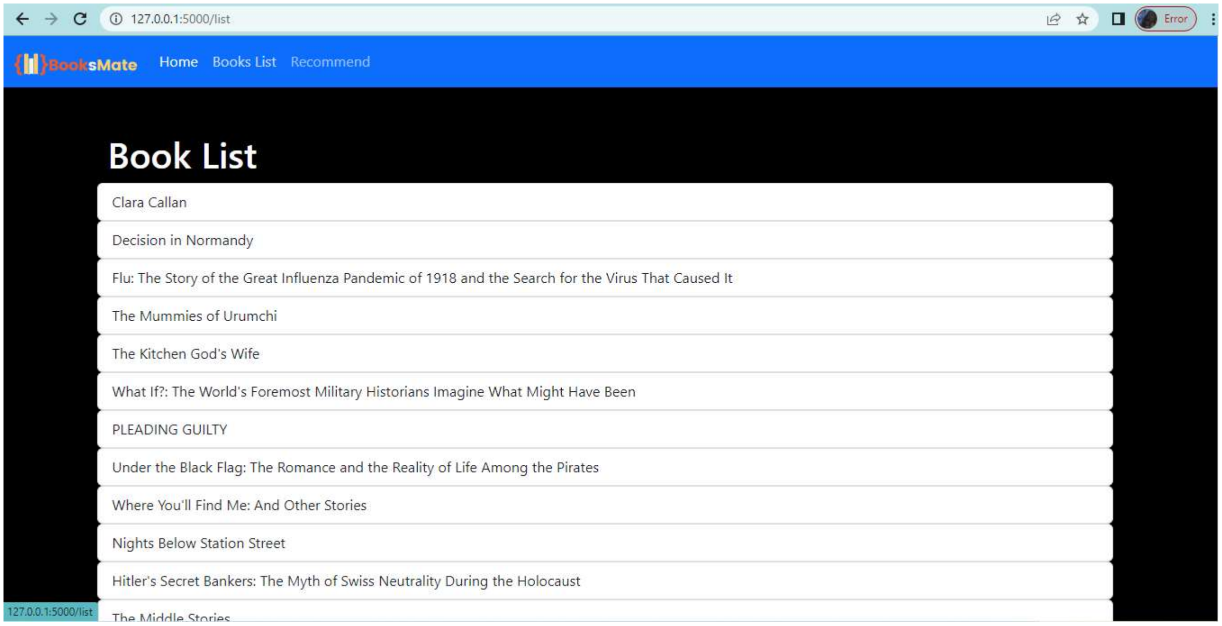

Figure 5 List of Books

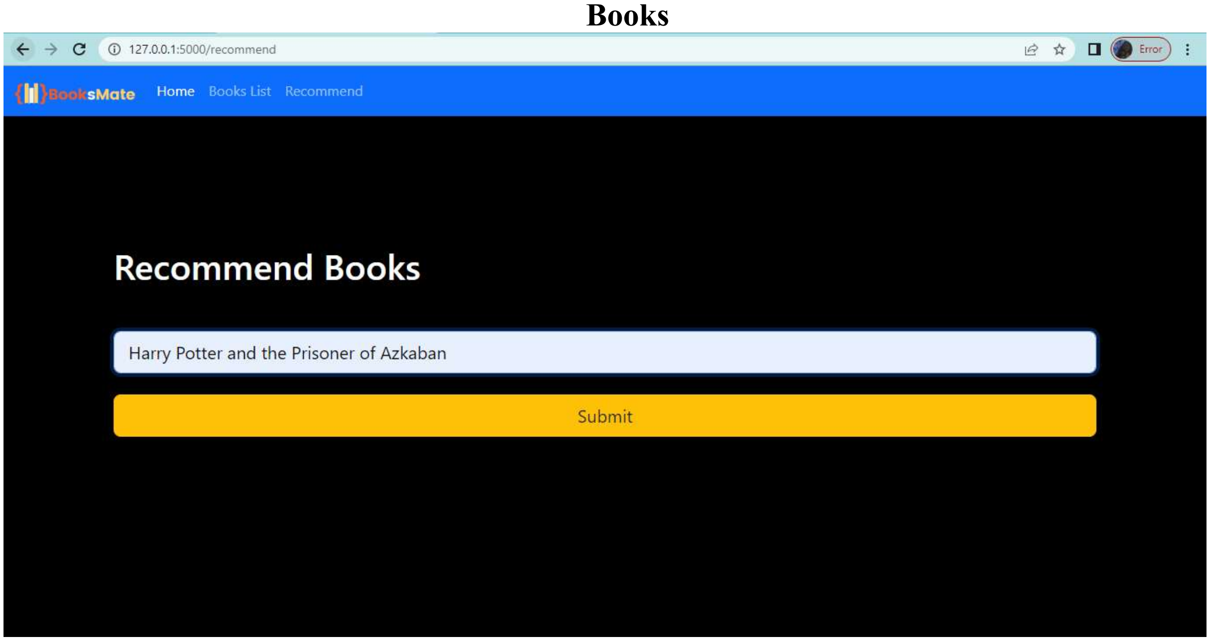

Figure 6 Recommendation Engine

VII. Conclusion and future enhancement

Even with the adaptation of a fitting algorithm for the recommendation, the RS faces an obstacle because of the large quantity of data that needs to be handled. According to the experimental results, the proposed algorithm with a compact dataset was more accurate than existing algorithms with full datasets. In addition, a popularity-based algorithm measures the similarity of books which gives more accurate results. It also proves that the recommendation system would help to increase the rate of sales. The recommendation system proposed here takes the number of users who have rated the books into account, without factoring in the absolute rating. Due to this, a recommendation might arise from a book that a user has given a low rating to, in which case a book might be recommended from a genre that the user dislikes. This recommendation system relies on the ratings given by users. So, trust is a major issue, like whether the feedback and rating given by the user are genuine or not. This recommendation system does not solve the trust issue.

Reference

1. Avirana, K.Deeba (2019) SRM Institute of Science and Technology, reviewed an Online Book Recommendation System using Collaborative Filtering(with Jaccard similarity).

2. Mr. Atish Bhosale, Ms. Komal Patil, Ms. Amruta Gaikwad, Mr.Riddhesh Deshmukh (2013) Jansons Institute of Technology Coimbatore, research on Book Recommendation System with Hybrid Approach

3. Anis Mansurma K S, Kavitha V K, Juby Mary Abraham, Sam G Benjamin (2021) APJ Abdul Kalam Technological University, Sasthamcotta, Kerala, survey paper on BOOK RECOMMENDATION SYSTEM.

4. K. Vasantha Rayudu, K. Sunil Kumar, G. Sai Kumar (2019) Anil Neerukonda Institute of Technology and Sciences, Studied on Online Book Recommendation System using Collaborative Filtering Algorithm

5. K. M. Kavin Chendhur, V. Priya, R. Mohana Priya, S. Lavanya Lakshmi (2021) Dr. Mahalingam College of Engineering and Technology, Pollachi, reviewed on Book Recommendation System using Improved Collaborative Filtering.

6. Nursultan Kurmashov (2015) Engineering and Natural SciencesSueyman Demirel University Kaskelen, Kazakhstan, studied on Online Book Recommendation System.

7. Zhi Hui Wang (2021) Wuxi Vocational College of Science and Technology, China Studied on Book Recommendation Algorithm Based on Collaborative Filtering and Interest Degree.

8. Dhiman Sarma, Tanni Mittra, Mohammad Shahadat Hossain(2021) Rangamati Science and Technology University, Bangladesh, studied on Personalized Book Recommendation System using Machine Learning Algorithm.

9. Fikadu Wayesa , Mesfin Leranso , Girma Asefa , Abduljebar Kedir , Ayodeji Olalekan Salau (2022) studied on Hybrid Book Recommendation System using Semantic Relationships.

10. Joeran Beel, Bela Gipp, Stefan Langer, Corinna Breitinger (2016) Otto-von-Guericke University, Magedeburg, Germany, studied on Book Recommendation System using Collaborative Based Filtering.