Enhanced Book Recommender System -A Content based Approach

Daljeet Kaur Khanduja[1] Surjeet Kaur[2]
[1]Professor [2]Associate Professor
[1,2]Department of Mathematics
[1]Sinhgad Academy of Engineering, Kondhwa, Pune, Maharashtra, India [2]SIES College of Arts, Science and Commerce (Autonomous), Sion, Mumbai, Maharashtra, India

Abstract — People require some tools to seek and collect useful information because it is difficult in this information era to sift through the large amount of material that is available on internet platforms. One of these tools is referred to as a recommendation system, which is a potent software method that assists in quickly traversing through large volumes of data to determine users' interests and give the necessary information. In this paper the three approaches for a content-based book recommendation system are discussed. In the first strategy, a simple system for recommending books is created, with certain input criteria like author, publisher, language, average rating, and recommendations for the books as the output. In the second method the K nearest neighbor (KNN) algorithm and cosine similarity are used to develop a book recommendation system based on several attributes. In the third method, the user enters a book title, which is then translated into vectors using Term Frequency-Inverse Document Frequency (TF-IDF), which then calculates the cosine similarity between related books before proposing related and similar books to the user. By implementing the above strategies, the user will be able to view book recommendations based on various attributes.

Keywords: Content based Approach, Cosine Similarity, K Nearest Neighbor (KNN), Term Frequency-Inverse Document Frequency

I. INTRODUCTION

Recent technological developments have made it possible to obtain a vast amount of internet information more quickly due to the presence of online services. Finding relevant and helpful stuff on the internet is more difficult because of the data overload. Even if customers adore your movie, product, or job opportunity, they might not be aware that it even exists. According to the late Steve Jobs, "People frequently don't know what they want until you show it to them." Recommendation systems are a subset of machine learning that provide customers with pertinent suggestions. A recommendation system aims to accurately forecast a user's interests by suggesting products that fit their interests. The objective of the recommender system is to expose the user to a wide range of products and opportunities that they might not have thought to look for on their own. Despite the fact that the majority of businesses produce a lot of data, very few are able to turn that data into insightful knowledge. This is where recommendation systems can help, as they can give their consumers the most pertinent content. Since they enable consumers to find goods they might not have otherwise, recommender systems are a helpful and viable alternative to search algorithms. Recommender systems often serve as information filtering tools, providing users with pertinent and tailored content or information. The main goal of recommender systems is to minimize the user's time and effort spent looking for pertinent information online.

Currently, recommender systems are utilized more and more in a variety of applications, including the web[1,2], books[3], e-learning[4,5,6], tourism[7,8], movies[9], music[10], e-commerce, news, specialized research resources[11], television shows[12,13], etc. To provide customers with individualized recommendations across a range of applications, it is crucial to develop superior and unique recommender systems since many classifications are possible in recommendation process[14,15,16].A recommendation system consists of background data—the knowledge the system has before the recommendation process starts—input data—the knowledge the user must provide to the system for it to generate a recommendation—and an algorithm that combines background and input data to produce recommendations. Analyzing user data and extracting pertinent data for additional predictions are the major functions of recommendation systems [17].This broad overview shows that for recommendation systems to function well, the following two prerequisites must be met:

1) Details regarding the users' choices.
2) A way to assess whether a user will find something interesting.

External data including user profiles, purchase history, and product ratings are typically included in the users' information [18]. Depending on the type of recommendation system, a user's interest in a particular item can be ascertained.

The paper is structured as follows. Section 2 presents a literature review on recommendation approaches. Section 3 discusses the collaborative and content-based filtering approach. In section 4 the algorithms used in the proposed work are discussed, Section 5 discusses the research methodology. In Section 6 results are presented and Section 7 concludes the work.

II. LITERATURE REVIEW

Using the Content based (CB) and Collaborative filtering (CF) methods to suggest books, a work entitled "Book Recommendation System" was suggested by Sushma Rjpurkar et al. (2015) [19]. In order to discover associations and correlations between a collection of elements, the author of this work used associative rule mining. They constructed a system using CF and CB methodologies. "A Proposed Hybrid Book Recommender System" by Suhas Patil et al. (2016) [20] constructed a hybrid recommender system using demographic, collaborative filtering, and content-based data. The author built a system in this work using approaches including collaborative filtering, content-based analysis, and 14 demographic analysis. Rarely do these techniques combine their features to create a superior recommendation system. A work titled "Personalized Book Recommendation

System using Machine Learning Algorithm" was proposed by Dhirman Sarma, Tanni Mittra, Mohammad Shahadat Hossain, et al. in 2019 [21] constructed a system using machine learning algorithms. System construction with UV Decomposition and KNN, an article titled "Book Recommendation System" was proposed by Jinny Cho, et al. (2016) [22]. The author of this study employs two approaches: collaborative filtering (CF) and content-based (CB). They employed the UV-Decomposition and K Nearest Neighbors (KNN) algorithms. They got a result that was 85% accurate.Kurmashov et al. (2015) [23] assessed the system using an online survey and employed CF based on Pearson correlation coefficient to propose books to readers online. Mathew and others (2016)[24] suggested a system that records the user's book purchases in detail. A hybrid algorithm employing collaborative filtering, content-based filtering, and association rules generates book suggestions using these book contents and ratings. This paper suggests a clear, easy-to-understand approach for book recommendations that aids readers in recommending the right book[25]. They tested the effectiveness of similarity measures while recommending books to users using a user-based mainly cooperative Filtering technique. The overall architecture of the intended system is introduced, and a model-style implementation is defined. The current systems don't examine the suggestion data and can't give readers enough information to decide whether or not to recommend a book[26]. Some systems also don't provide readers with a way to provide input, which could dampen their enthusiasm. They created a cutting-edge book recommendation system to address these problems. The majority of websites typically don't propose books that the buyer would be interested in reading. Consumers typically receive a variety of data and recommendations, but the majority of these are irrelevant]. In the paper[27], a novel method for recommending books to customers is presented. To create affordable and practical recommendations, this technique combines the possibilities for cooperative filtering, association rule mining, and content filtering.

III. RECOMMENDATION TECHNIQUES

Every recommendation system, in general, uses a certain procedure to provide recommendations. Two sources of information are known to be required as input for the recommendation process if the procedure is seen as a "black box." These data sources include user profiles and product or item information. Ideally, the data kept in user profiles relates to their choices and is provided voluntarily by the user. However, this data can also be gleaned from other outside sources like websites, purchasing patterns, etc. A database of users and goods can be used as an input to the recommendation algorithm, and the algorithm's output will be the recommendations. As in our situation, the inputs are a database of books with different attributes, and the output is a list of recommended books.

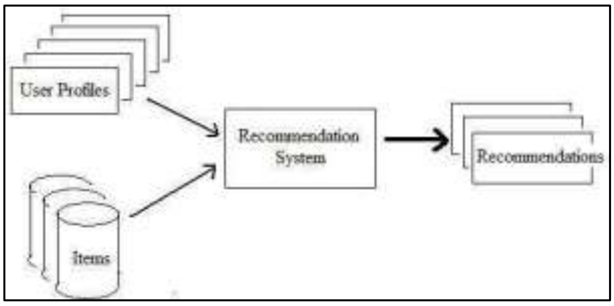

Fig. 1: Recommendation process as a black box

In the development of machines, a variety of recommender system methodologies are used, including content-based approach, collaborative approach, and hybrid approach (combining the aforementioned two ways). Different information sources are used by the current recommendation systems, such as collaborative filtering and content-based filtering, to provide recommendations.

A. *Collaborative Filtering:*

The collaborative filtering approach uses data collection and analysis based on various aspects, such as opinion expressed in the form of ratings given by other users for a particular book and user's past behavior towards the system, which includes books the user has previously read, preferences or activities, as well as similar decisions made by other users. The method then predicts what users will like based on their similarities with other users. Then, this model is used to forecast the ratings or products that the user may be interested in.

Limitations of Collaborative Filtering:

- Cold Start issue: In order for users to offer precise recommendations, collaborative filtering systems frequently need a substantial amount of previous data [28].
- Scalability: Collaborative filtering offers suggestions for a range of settings with billions of users and items. As a result, computing recommendations requires a significant amount of computing power.
- Sparsity: The quantity of goods offered on well-known websites like amazon or flipkart is very high. Because of this, the majority of active users only rate a small portion of the overall database. As a result, the most popular things receive extremely few ratings. For collaborative filtering to produce accurate predictions, a sizable dataset of active users who have already evaluated the product is needed.

B. *Content Based Filtering:*

This strategy is based on using the item description and user profile as a foundation. Based on the user's past behavior or explicit feedback, content-based filtering uses item features to suggest additional products that are similar to what the user already likes. When making suggestions to one user, content-based filtering does not require information from other users. In a content-based recommender system, the things are described using keywords, and the user's preferences are shown in their profile. In other words, these algorithms aim to suggest products that a user may enjoy or be interested in right now. In content-based recommender systems all the data items are gathered into various item profiles according to

their features or descriptions. For instance, the features of a book would be the author, publisher, etc. A movie's features might include things like the director, actors, etc. The other items in the item profile that receive a favorable rating from a user are then combined to create the user profile. This user profile aggregates all item profiles whose products have received high user ratings. The user is then given recommendations for products based on this user profile. •
Limitations of Content based filtering:
To make an accurate recommendation using this strategy, thorough knowledge of the item's features is required. It is possible that not all items may always have access to this knowledge or information.

- The ability of this strategy to broaden the users' current preferences or interests is limited.
- Content-based filtering has an overspecialization issue because it frequently recommends the same kinds of products.
- Compared to collaborative filtering, users using content-based filtering are less likely to rate the items, making it more difficult to obtain user input.

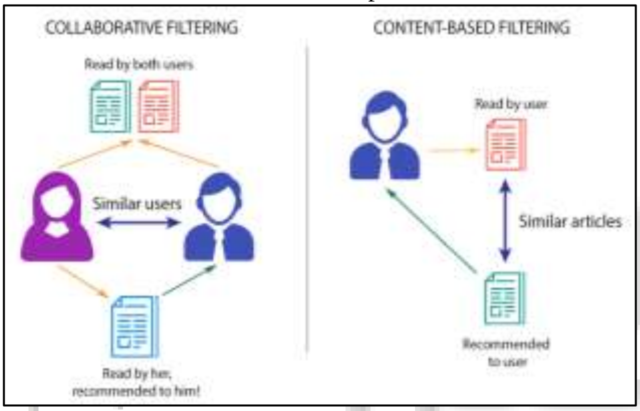

Fig. 2: Collaborative and Content based Filtering

Due to the fact collaborative filtering only offers item recommendations based on user ratings, recommender systems adopting this approach are particularly vulnerable to the cold start issue[29]. As there is no history data available for a new item or new person, the system would not be able to provide useful recommendations when they are added to the system[30]. On the other hand, the content-based filtering strategy is less vulnerable to the cold start issue and can still produce reliable product suggestions even in the absence of significant user ratings. This is because with this method, appropriate recommendations are generated based on the metadata of new products rather than by considering user activities or interactions[31].

C. *Hybrid Recommender System:*

This system combines a collaborative data filtering strategy with a content-based approach. There are numerous ways to implement hybrid techniques, including creating content-based and collaborative-based predictions separately, merging them by giving collaborative-based approaches content-based capabilities, and vice versa, or combining the approaches into a single model. The major goal of the hybrid approach is to combine collaborative and content-based filtering to increase the accuracy of recommendations. This system combines a content-based approach with a collaborative data filtering strategy. Implementing a hybrid strategy can be done in a few different ways, including independently creating content- and collaboratively based predictions, integrating them by adding content-based capabilities to a collaborative-based approach (and vice versa), or combining the approaches into a single model. The major goal of the hybrid approach is to combine collaborative and content-based filtering to increase the accuracy of recommendations.

IV. ALGORITHMS USED IN THE PROPOSED WORK

A. *K Nearest Neighbor (KNN) Algorithm*

The k Nearest Neighbors (KNN) algorithm is the most popular machine learning algorithm for supervised learning that can be used to solve classification and regression issues. It stores all the available cases and classifies the new data or case based on a similarity measure. The KNN algorithm is predicated on the notion that comparable/related objects are close together and uses the constant k to denote the number of neighbors. By identifying the closest neighbor nearby, the algorithm attempts to ascertain which category a new unknown data item belongs to. By identifying the closest neighbor, the algorithm attempts to ascertain which category a new unknown data item belongs to. It is predicated on the assumption that the nearest neighbor to and by the unknown data point will have similar characteristics. The algorithm calculates the distance between each data point in the region closest to the unknown data point to create predictions.

The graphic below shows how identical data points are frequently closely related. KNN uses Euclidean distance to calculate the distance between points to represent the idea of similarity (sometimes referred to as distance, proximity, or closeness)

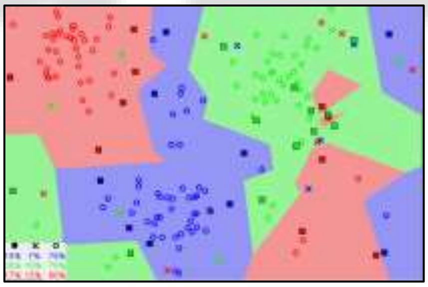
Fig. 3: Image showing how similar data points typically exist close to each other

B. *Similarity Measures*

The distance metric is applied to determine similarity. The closest points are the most comparable, while the furthest points are the least significant. The degree of resemblance is arbitrary and heavily influenced by the application and domain.

Regardless of the size of the documents, cosine similarity is a statistic used to determine how similar they are. It calculates the cosine of the angle formed by two vectors that are projected onto a multidimensional space. The cosine similarity increases with decreasing angle.

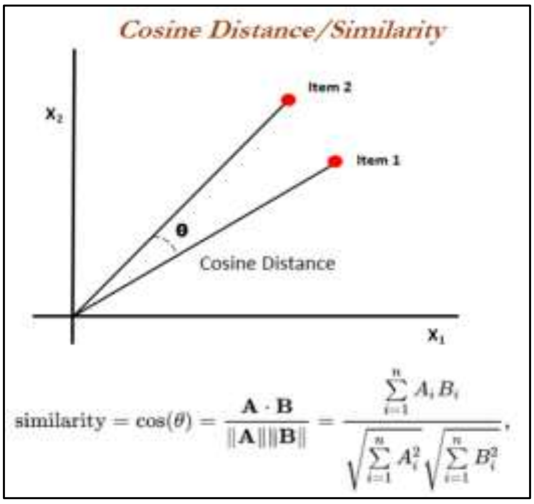

Fig. 4: Cosine Similarity

The cosine similarity is useful since it increases the likelihood that the two comparable documents will be oriented closer together, even if they are separated by a large Euclidean distance because of the size of the documents.

C. Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency (TF): It is defined as frequency of words in the current document to the total number of words in the document. When a word appears more frequently in a document, its weight is increased, therefore, to normalize it, its frequency is divided by the length of the document.

$$\mathrm{Tf}(t) = \frac{\text{Frequency occurence of term t in document}}{\text{Total number of terms in document}}$$

Inverse Document Frequency (IDF): It is defined as the proportion of total documents to documents where a word appears frequently. It denotes a word's rarity; the higher the IDF, the fewer frequently the term appears in the text. It helps to give uncommon terms in the papers a better score.

$$\mathrm{Idf}(t) = log_{10}\left(\frac{\text{Total Number of documents}}{\text{Number of documents containing term t}}\right)$$

To prevent division by zero when there are no terms in the text, certain variations of the IDF definition add 1 to the denominator.

V. RESEARCH METHODOLOGY

This main goal of this paper is to make it quick and simple for individuals to get good book recommendations. To build the book recommendation system the books.csv data set from Goodreads has been implemented in this paper. The data set includes a variety of attributes, including the book's ID, author, average rating, total number of ratings, language, and publisher. This paper examines three strategies for recommending books to readers.

A. Approach1: Using particular attributes to recommend books to users.

The goal of the system is to create a recommendation engine that can make more accurate recommendations to the users.

The first method involves developing a straightforward content-based book recommendation engine in Python and compiling it in a Jupyter notebook to suggest highly rated books based on factors like the author, publisher, language, average rating, etc. The dataset's summary is

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11127 entries, 0 to 11126
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   bookID              11127 non-null  int64
 1   title               11127 non-null  object
 2   authors             11127 non-null  object
 3   average_rating      11127 non-null  object
 4   isbn                11127 non-null  object
 5   isbn13              11127 non-null  object
 6   language_code       11127 non-null  object
 7     num_pages         11127 non-null  object
 8   ratings_count       11127 non-null  int64
 9   text_reviews_count  11127 non-null  int64
 10  publication_date    11127 non-null  object
 11  publisher           11127 non-null  object
 12  Unnamed: 12         4 non-null      object
dtypes: int64(3), object(10)
memory usage: 1.1+ MB
```

This technique involves the following steps:

1) Setting up the environment:

All of the Python libraries used in this work were imported in this phase. Imported libraries include the Matplotlib and seaborn libraries for data visualization, the panda's library for reading and manipulating data, and the ipywidgets library for building interactive plot creation. To make our plots more interactive, we use interactive HTML widgets called ipywidgets.

2) Understanding the data set:

In this step, various attributes of the data set, such as the book ID, title, author, publisher, average rating, etc., are analyzed along with key characteristics, such as the number of variables in the data set, any missing values, and any inaccuracies, to understand the nature of the data. The summary statistics are then calculated.

3) Preprocessing:

To remove duplicates, missing values, and Nan values from the dataset, the data is cleaned and preprocessed.

4) Feature Engineering:

Feature engineering is used to reduce the size of the feature set and extract significant features such as traits, properties, and attributes from the raw data. This makes computing more practical and straightforward.

5) Exploratory data analysis and Visualization

The books are recommended based on a variety of characteristics after the data has been processed and numerous insights have been extracted from it.

VI. RESULTS AND DISCUSSIONS: RECOMMENDING BOOKS BASED ON VARIOUS ATTRIBUTES.

A count plot was created to illustrate the books that appear the most frequently. To visualize the most occurring books a count plot has been plotted and it has been observed that Illiad and The Brothers Karamazov are the most occurring books.

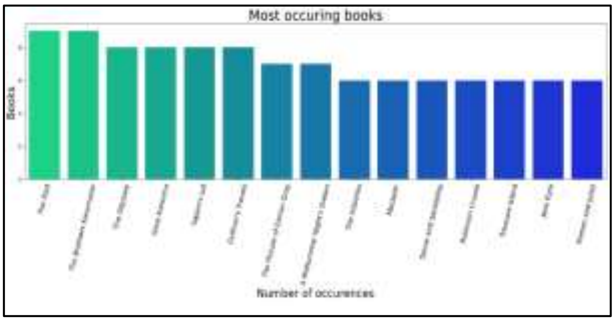

Fig. 5: Count Plot of Most Occurring Books

A dispersion plot was created to illustrate the average rating of the books, and it was found that the majority of them have an average rating between 3 and 5, with very few having a rating below 3.

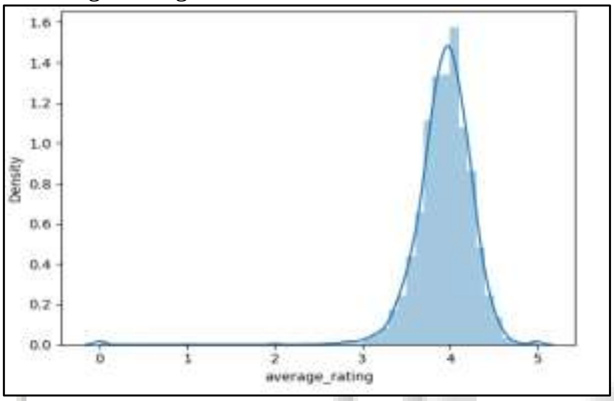

Fig. 6: Dispersion Plot of Average Rating of Books

A. Recommending books based on the attribute author

Using the Seaborn Library and the attribute author, the top 10 writers with the most books published were visualized using a count plot. The results show that Stephen King and P.G. Woodhouse are the two authors that have the most books published.

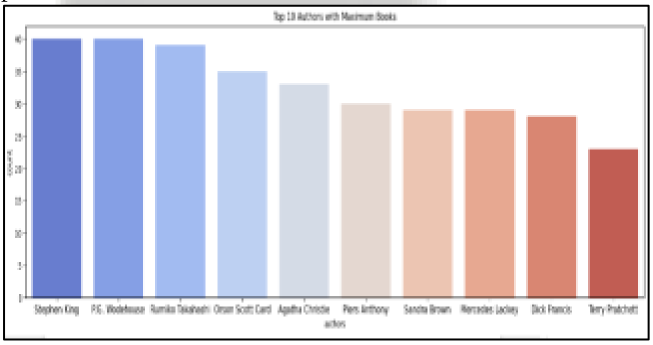

Fig. 7: Count Plot of Top 10 Authors with Maximum Books

By defining the recommend function and calling it with the author's name as a parameter, we were able to recommend books based on a certain author and get the results accordingly. But the same procedure must be done if we want to suggest books based on works by other authors. In order to solve this issue, we used Python's ipywidgets package to build an interactive function that gives us the appropriate output in the form of a drop-down menu from which we can choose any author and have the results update accordingly.

author_name: Stephen King

	title	average_rating
1575	Carrie / 'Salem's Lot / The Shining	4.54
3150	The Green Mile	4.44
9867	Different Seasons	4.35
8653	Different Seasons	4.35
3155	On Writing: A Memoir	4.33
6019	The Dark Tower (The Dark Tower #7)	4.28
1459	The Drawing of the Three (The Dark Tower #2)	4.23
3489	The Shining	4.22
3487	The Shining	4.22
3156	The Shining	4.22

So, using this strategy, we have developed a simple content-based recommender engine to suggest books based on specific criteria, such as average rating, author, publisher, and language.

B. Method 2: Recommendation of books to users using KNN

In the second strategy, we employ the KNN algorithm with cosine similarity to suggest books to readers based on language code and ratings count.

In order to build a recommendation engine based on language code and rating count, we preprocess the data. The values are then scaled down using the Minmax scaler from the sklearn library, which will assist to reduce the bias for some of the books that have too many attributes. The dataset is prepared to train the model using the KNN algorithm after preprocessing. The parameters are specified, and the neighbors package is imported from the sklearn library. The distance between the data points is calculated using the cosine metric and the brute force approach. The model is then ready for suggestions after the features-scaled data is fitted and the k nearest neighbors are determined for each data point. The ipywidgets package is used to build an interactive function that generates recommendations. We can select any book from the drop-down menu, and the function will provide suggested books based on different books that people have expressed interest in.

book_name: The Iliad

```
['The Iliad',
 'The Call of the Wild',
 "She's Come Undone",
 'The Fountainhead',
 'Beloved',
 'Othello']
```

C. Method 3: Recommendation of books through book title.

Using the book title as the input argument, a recommendation engine is created in this method. The basic concept is to turn the texts or words into vectors and represent them in a model of vector space. We establish a dictionary of terms (also known as a bag of words), which is present throughout the entire document space, in order to determine how frequently the term or word appears in the document as well as to represent the content in vector form. We disregard some frequent words, also known as stop words, because they will not aid us in choosing vital terms and the subject of a document. To determine the TF.IDF score for each of the text's remaining words, we first generate a vector. The books' titles serve as the content for TFIDF, which is then utilized to construct a cosine similarity matrix and provide recommendations based on the similarity of the books.

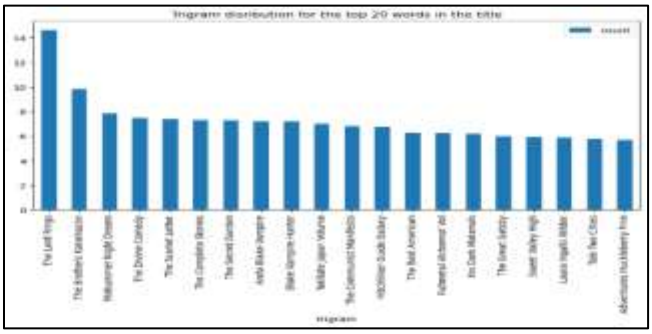

Fig.8 Trigram distribution for top 20 words in the title

```
Recommending2book similar toHarry Potter and the Prisoner of Azkaban (Harry Potter  #3)
____________________
You may also like to read: Harry Potter and the Prisoner of Azkaban (Harry Potter  #3)(score:)1.0000000000000002)
You may also like to read: Harry Potter Collection (Harry Potter  #1-6)(score:)0.470130299691641)
```

Therefore, in this approach by enabling the user to quickly find the book they are seeking for, this strategy helps improves the effectiveness of the system. As a result, the volume of data that must be transmitted may be reduced, resulting in a reduction in system operating expenses.

VII. CONCLUSION

While discussing numerous techniques for creating recommender systems, this study focused on books and utilized a content-based filtering strategy on books.csv from the Goodreads dataset. We have innovated and altered the recommendation mechanisms in our proposed system. To propose books, this book recommendation system considered specific factors including author, publisher, and language code, average rating. KNN and cosine similarity were then used to construct a book recommender system that gives users more pertinent recommendations. In order to determine the significance of any book and recommend similar books, the notions of (TF- IDF) term frequency and inverse document frequency have been implemented.

REFERENCES

[1] Castellano G, Fanelli AM, Torsello MA. NEWER: A system for neuro-fuzzy web recommendation. Appl Soft Computing. 2011;11:793–806
[2] Ochi P, Rao S, Takayama L, Nass C. Predictors of user perceptions of web recommender systems: How the basis for generating experience and search product recommendations affects user responses. Int J Hum Computing Stud. 2010;68:472–82.
[3] Crespo RG, Martínez OS, Lovelle JMC, García-Bustelo BCP, Gayo JEL, Pablos PO. Recommendation system based on user interaction data applied to intelligent electronic books. Computers Hum Behaviour. 2011;27:1445–9
[4] Wang SL, Wu CY. Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system. Expert Syst Appl. 2011;38:10831–8
[5] Salehi M, Kamal Abadi IN. A hybrid attribute–based recommender system for e–learning material recommendation. IERI Procedia. 2012;2:565–70.
[6] Bobadilla J, Serradilla F, Hernando A. Collaborative filtering adapted to recommender systems of e-learning. Knowledge Based Syst. 2009;22:261–5.
[7] Lorenzi F, Bazzan ALC, Abel M, Ricci F. Improving recommendations through an assumption-based multiagent approach: An application in the tourism domain. Expert System Appl. 2011;38:14703–14
[8] Isinkaye FO, Folajimi YO, Ojokoh BA. Recommendation systems: Principles, methods and evaluation. Egyptian Inform J. 2015;16:261–73.
[9] Bobadilla J, Serradilla F, Bernal J. A new collaborative filtering metric that improves the behaviour of recommender systems. Knowledge-Based Syst. 2010;23:520–8.
[10] Porcel C, Moreno JM, Herrera-Viedma E. A multi-disciplinary recommender system to advice research resources in University Digital Libraries. Expert System Appl. 2009;36:12520–8.
[11] Yoshii K, Goto M, Komatani K, Ogata T, Okuno HG. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. IEEE Trans Audio Speech Lang Process. 2008;16:435–47.
[12] Shin C, Woo W. Socially aware tv program recommender for multiple viewers. IEEE Trans Consum Electron. 2009;55:927–32.
[13] Bjelica M. Towards TV recommender system: experiments with user modelling. IEEE Trans Consum Electron. 2010;56:1763–9.
[14] Schafer, J. B., Konstan, J. and Riedl, J.: 1999, _Recommender Systems in E- Commerce‘. In: *EC '99: Proceedings of* the First ACM Conference on Electronic Commerce, Denver, CO, pp. 158-166.
[15] Resnick, P., Varian, H., Recommender Systems. Communications of the ACM, 40, 3, (1997), 56-58.
[16] Terveen, L. and Hill, W: 2001, _Human-Computer Collaboration in Recommender Systems‘. In: J. Carroll (ed.): Human Computer Interaction in the New Millenium. New York: Addison-Wesley.
[17] Anne Yun-An Chen and Dennis McLeod, Collaborative Filtering for InformationRecommendation Systems
[18] J. Ben Schafer, Nathaniel Good, and Joseph Konstan et. al. Combining collaborative filtering with personal agents for better recommendations. In Proceedings of the

1999 National Conference of the American Association of Artificial Intelligence, pages 439–436, 1999.
[19] Ms. Sushma Rajpurkar, Ms. Darshana Bhatt and Ms. Pooja Malhotra, “Book Recommendation System” International Journal for Innovative Research in Science & Technology vol.1, issue 11, April 2015
[20] Suhas Patil and Dr. Varsha Nandeo, “A Proposed Hybrid Book Recommender System” International Journal of Computer Applications vol.6 – No.6,Nov – Dec 2016
[21] Dhirman Sarma, Tanni Mittra and Mohammad Shahadat Hossain, ”Personalized Book Recommendation System using Machine Learning Algorithm” The Science and Information Organization vol.12,2019
[22] Jinny Cho, Ryan Gorey, Sofia Serrano, Shatian Wang, JordiKai Watanabe-Inouye, “Book Recommendation System” Winter 2016.
[23] Kurmashov, N., Konstantin, L., Nussipbekov, A. (2015). Online book recommendation System. Proceedings of Twelve International Conference on Electronics Computer and Computation (ICECC)
[24] Mathew, P., Kuriakose, B. And Hegde, V. (2016). Book Recommendation System through content based and collaborative filtering method. Proceedings of International Conference on Data Mining and Advanced Computing (SAPIENCE).
[25] M. Kommineni, P. Alekhya, T. M. Vyshnavi, V. Aparna, K. Swetha and V. Mounika, "Machine Learning based Efficient Recommendation System for Book Selection using User based Collaborative Filtering Algorithm," 2020 Fourth International Conference on Inventive Systems and Control (ICISC), 2020, pp. 66-71, doi: 10.1109/ICISC47916.2020.9171222.
[26] B. Cui and X. Chen, "An Online Book Recommendation System Based on Web Service," 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009, pp. 520-524, doi: 10.1109/FSKD.2009.328.
[27] A. S. Tewari, 27)A. Kumar and A. G. Barman, "Book recommendation system based on combine features of content-based filtering, collaborative filtering and association rule mining," 2014 IEEE International Advance Computing Conference (IACC), 2014, pp. 500-503, doi: 10.1109/IAdCC.2014.6779375
[28] Cover, T and Hart, P., Nearest neighbour pattern classification. Information Theory, IEEE Transactions on, 13(1):21–27, 1967
[29] Linden, B. Smith, AND J. York. Amazon.com recommendations: item- to-item collaborative filtering, IEEE Internet Computing, vol. 7, pp. 76-80, 2003.
[30] B. Smith and G. Linden. Two decades of recommender systems at amazon.com, IEEE Internet Computing, vol. 21, no. 3, pp. 12-18, 2017.
[31] Adomavicius, G.; Tuzhilin, A., "Toward the next generation of recommender systems: a survey of the state of- the-art and possible extensions," Knowledge and Data Engineering, IEEE Transactions on , vol.17, no.6, pp.734,749, June 2005