Book recommendation system

D . Venkata Nithin Datta.

SRM UNIVERSITY

Abstract

In recent years, as the amount of information available on the internet is growing quickly people need certain tools in order to find and obtain the correct information. To acquire the information without having to look through all the goods or products, we need something called a recommendation system. The development of computing has made it possible to provide users with personalised recommendation lists based on their activities. In comparison to travelling to a store and making their own purchases, internet recommendations give a simpler and faster way to make purchases, and online transactions are quite quick. By offering products that consumers can use right away, this recommendation system makes their lives easier. The top items are usually recommended to customers using a recommendation system. Nowadays, online book selling websites compete with one another in a variety of ways. In this study, we propose a book recommendation system that allows both habitual and infrequent readers to easily obtain results that reflect their interests using their own content of interest as queries. The book recommendation system must suggest books that are of interest to the buyer. A simple and user-friendly book recommendation system that assists readers in selecting the best book to read next. Furthermore, Our discussion is based on the book recommendation system (BRS), which uses content based filtering (CBF), Popularity based filtering, Recommendation using Average weighted Rating and K-Nearest Neighbour algorithm.

Literature review

[1] Developed a system for determining recommended books based on the similarities between the topics and feelings found in books and the interests of users.**[2]** The cold start problem and matrix sparsity are two problems that are almost guaranteed to occur in a recommendation system, and this paper aims to solve the former. This paper focuses on the problem of cold start and attempts to implement personalised book recommendation with students' course selection, which is required for students during their college time. This model employs a user-based collaborative filtering algorithm. **[3]** This paper suggests a collaborative filtering mechanism for a book recommendation system. Based on carefully computed and retrieved reviews from several users, the highest rated books are suggested to the customer. **[4]** A combination of content and collaborative filtering techniques is applied. The system really assesses the quality of the books it recommends based on the ratings provided by the current users. It also uses an association rule mining algorithm to uncover interesting associations and relationships among a big data set of books and to provide effective book recommendations. **[5]** This paper considers two important factors for the collaborative filtering algorithm based on books: user activity and time. The top-k users with the highest similarity are chosen as the recommendation set in this paper. The lowest mean square deviation can be obtained by varying the k value. A hybrid model may improve the performance of the recommendation system by combining different methods. **[6]** In the suggested framework, people are only recommended when K randomly selected individuals are filtered based on how similar they are to each other and collaborative filtering is only used on those users.[7] In order to obtain better performance, this project uses the KNN algorithm and association rule mining to help solve the problem of data sparcity. **[7]** In this paper, we propose a book recommendation that, through collaborative filtering, provides users with recommendations on various genres based on information about their preferences provided during registration. The benefit of this system is its speed and simplicity

Introduction

There are a lot of data available nowadays in the information sector, but none of them are of any use until they are turned into information that is insightful or helpful. Therefore, it is essential to examine this vast volume of data and draw out relevant information. Users are able to make use of data from numerous different dimensions. For this reason, recommendation is one of the helpful tools that suggests a product or object to a user. Recommendation is used in many applications, including those for movies, music, news, books, and other products. With the rapid growth of the online book market, it is a practical challenge to suggest accurate books based on users' past ratings or purchases. Good tailored recommendations can enhance the user experience in new ways. These suggestions largely fall into two categories. Collaboration is the first, and content-based filtering is the second. In collaborative filtering, a user is given product recommendations based on the preferences of other users who share similar tastes. Based on the users' favourite features of other content or the description or content of a specific item, a content-based recommendation engine makes suitable recommendations to the users. KNN algorithm is also used for recommendation. For this, a dataset is used, and some of the aforementioned algorithms are applied to produce recommendations.

Methodology

Dataset Collection/Dataset description

The dataset is comprised of three csv files namely: Users, Books, Ratings

Users Dataset.

This data set consists of following attributes, User-ID (unique for each user), Location (contains city, state and country separated by commas) and Age

```
users.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 278858 entries, 0 to 278857
Data columns (total 3 columns):
 #   Column   Non-Null Count   Dtype
---  ------   --------------   -----
 0   User-ID  278858 non-null  int64
 1   Age      278858 non-null  float64
 2   Country  278858 non-null  object
dtypes: float64(1), int64(1), object(1)
memory usage: 6.4+ MB
```

Books Dataset.

The books dataset consists of ISBN (unique for each book), Book-Title, Book-Author, Year-Of-Publication, Publisher, Image-URL-S, Image-URL-M, Image-URL-L

```
books.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271360 entries, 0 to 271359
Data columns (total 5 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   ISBN                 271360 non-null  object
 1   Book-Title           271360 non-null  object
 2   Book-Author          271360 non-null  object
 3   Year-Of-Publication  271360 non-null  float64
 4   Publisher            271360 non-null  object
dtypes: float64(1), object(4)
memory usage: 10.4+ MB
```

Ratings Dataset.

Ratings dataset consists of following attributes User-ID, ISBN, Book-Rating

	User-ID	Location	Age
0	1	nyc, new york, usa	NaN
1	2	stockton, california, usa	18.0
2	3	moscow, yukon territory, russia	NaN
3	4	porto, v.n.gaia, portugal	17.0
4	5	farnborough, hants, united kingdom	NaN

```
ratings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1149780 entries, 0 to 1149779
Data columns (total 3 columns):
 #   Column       Non-Null Count    Dtype 
---  ------       --------------    ----- 
 0   User-ID      1149780 non-null  int64 
 1   ISBN         1149780 non-null  object
 2   Book-Rating  1149780 non-null  int64 
dtypes: int64(2), object(1)
memory usage: 26.3+ MB
```

Dimensions of datasets

```
print("Books Data:    ", books.shape)
print("Users Data:    ", users.shape)
print("Books-ratings: ", ratings.shape)
```

```
Books Data:     (271360, 8)
Users Data:     (278858, 3)
Books-ratings:  (1149780, 3)
```

Data Pre-processing

Users Dataset :

In the users_dataset we have the following feature variables.

User ID (unique for each user)

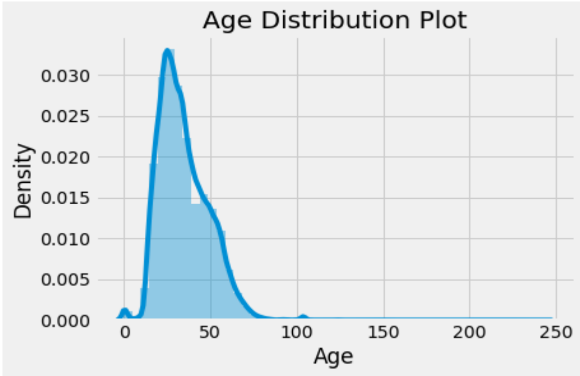
Location (contains city, state and country separated by commas)

Age

Out of these features, User-ID is unique for each user, Location contains city, state and country separated by commas and we have Age given across each user.

Age:

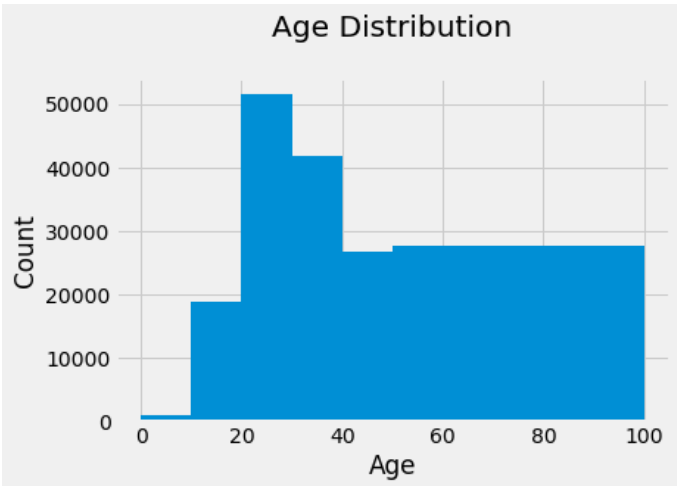
Let's understand the Age distribution of the given user dataset. By using a distplot for the Age column we can get the distribution of ages and the density. Below is the distplot.



From the given distplot we can see that we have outliers in the Age column, we will treat these outliers in the coming section of outlier treatment.

Now let’s have a look at the age distribution in a range of 0 to 100 by plotting a histogram

From the given histogram plot for age, we see that the distribution is right skewed. This information will be used further in imputing the Null values in the Age column.



Location:

Now let’s deep dive into our location column. This column has city, state and country separated by commas. We will first segregate these into different columns and we will introduce a new column “Country” so that we can analyse on the basis of the country of different users. The following code will separate the Country from the location.

```
for i in users:
    users['Country']=users.Location.str.extract(r'\,+\s?(\w*\s?\w*)\"*$')
```

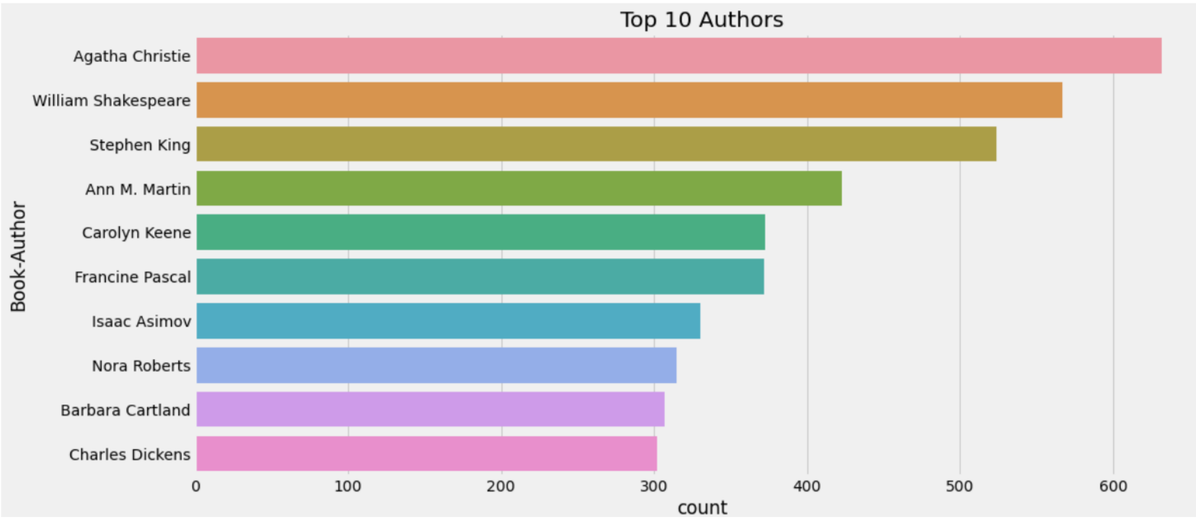
There are mis-spellings in some of the country names. We will first correct these and then plot the top 10 countries from where we have the maximum number of users. The following countplot shows the top 10 countries, here we analyzed that the maximum number of users belong to the USA.

For books Dataset

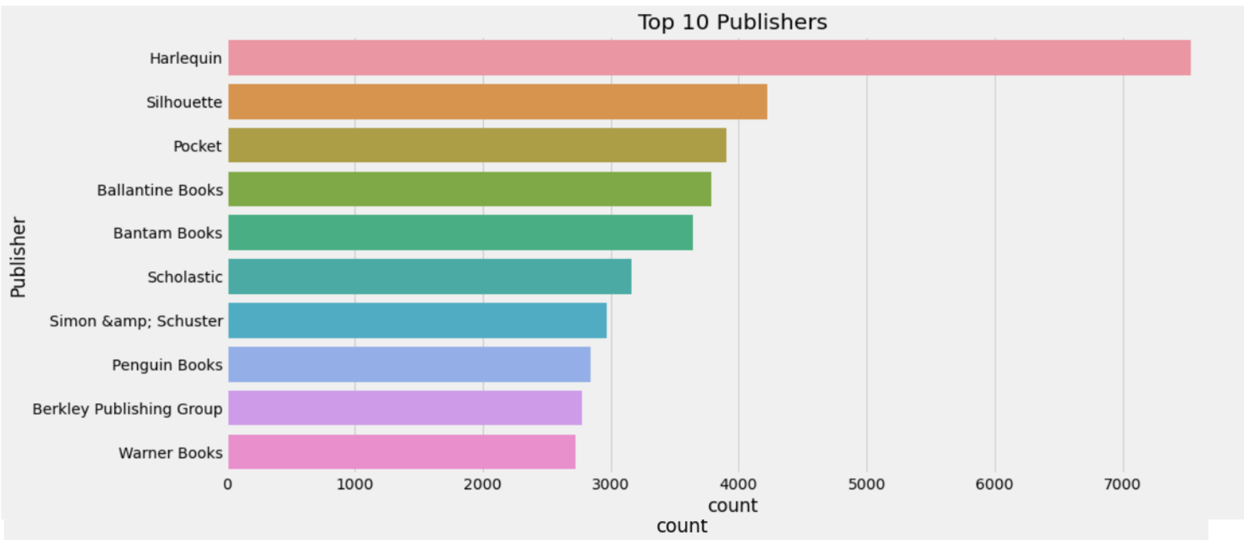
In the books_dataset we have the following feature variables.

1) ISBN (unique for each book)
2) Book-Title
3) Book-Author
4) Year-Of-Publication
5) Publisher
6) Image-URL-S
7) Image-URL-M
8) Image-URL-L

From the count plot, let's find the top 10 Book-Author and top 10 Book-Publishers. Further we find that both the plots are skewed and the maximum number of books are from top 10 Book-Authors and top 10 Book-Publishers.



Check for the unique years of publications. Two values in the year column are publishers.



Also, for three tuples name of the author of the book was merged with the title of the book. Manually set the values for these three above obtained tuples for each of their features using the ISBN of the book.

```
[0, 1376, 1378, 1806, 1897, 1900, 1901, 1902, 1904, 1906, 1908, 1909, 1910, 1911, 1914, 1917, 1919, 1920, 1921, 1922, 1923, 192
4, 1925, 1926, 1927, 1928, 1929, 1930, 1931, 1932, 1933, 1934, 1935, 1936, 1937, 1938, 1939, 1940, 1941, 1942, 1943, 1944, 194
5, 1946, 1947, 1948, 1949, 1950, 1951, 1952, 1953, 1954, 1955, 1956, 1957, 1958, 1959, 1960, 1961, 1962, 1963, 1964, 1965, 196
6, 1967, 1968, 1969, 1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 198
7, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2008, 201
0, 2011, 2012, 2020, 2021, 2024, 2026, 2030, 2037, 2038, 2050]
```

Dropping last three columns containing image URLs which will not be required for analysis.

Convert the type of the years of publications feature to the integer.

By keeping the range of valid years as less than 2022 and not 0, replace all invalid years with the mode of the publications that is 2002.

By keeping the range of valid years as less than 2022 and not 0, replace all invalid years with the mode of the publications that is 2002.

For Ratings Dataset

In the ratings_dataset we have the following feature variables.

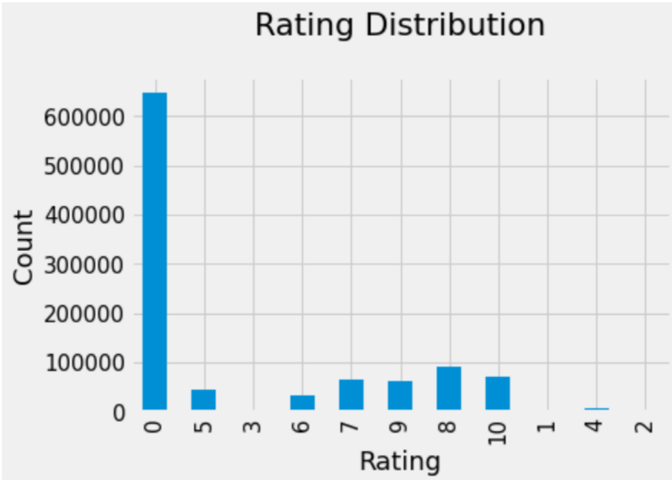
1) User-ID
2) ISBN
3) Book-Rating

```
ratings.head()
```

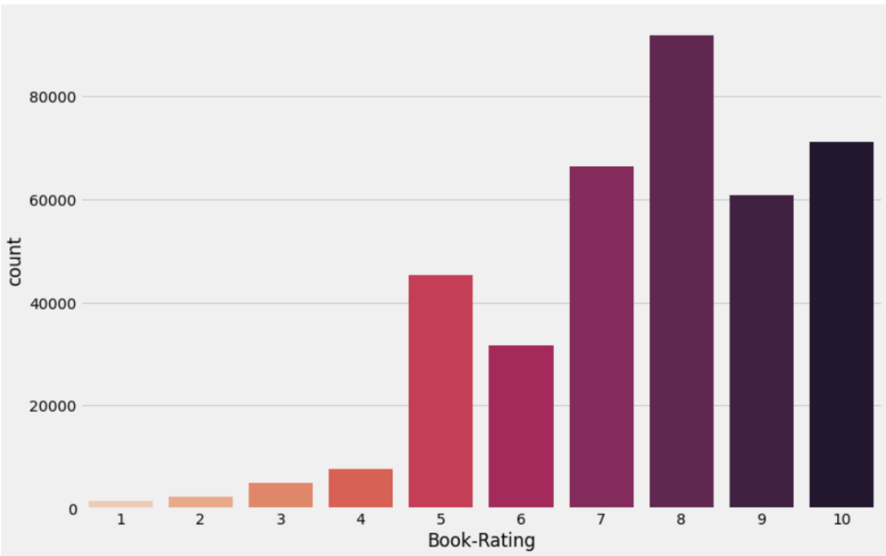
	User-ID	ISBN	Book-Rating
0	276725	034545104X	0
1	276726	0155061224	5
2	276727	0446520802	0
3	276729	052165615X	3
4	276729	0521795028	6

Let's find the distribution of ratings frequency in our ratings dataset. From the following frequency plot, we find that most of the ratings are 0 which is implicit rating. The following code snippet will give us the frequency distribution.

The ratings are very unevenly distributed, and the vast majority of ratings are 0 .As quoted in the description of the dataset - BX-Book-Ratings contains the book rating information. Ratings are either explicit, expressed on a scale from 1-10 higher values denoting higher appreciation, or implicit, expressed by 0. Hence segregating implicit and explicit ratings datasets.



Book required by the user




```
bookName = input("Enter a book name: ")
number = int(input("Enter number of books to recommend: "))

# Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))
```

```
Enter a book name: Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))
Enter number of books to recommend: 5
```

Recommendation methods

Popularity based filtering

The popularity-based recommendation system, as its name implies, follows trends. In essence, it makes use of current fashion products. For instance, there is a likelihood that it will offer a product to a newly registered user if that product is one that every new user typically purchases. It doesn't have cold start issues, so it can make product recommendations for a variety of different filters right away. The past information about the user is not required. It makes recommendations based on the most popular books across the entire collection, the most popular books by the same author, the publisher of the specified book, and the most popular books by year.

Recommended books by popular in the whole collection

```
print("Top", number, "Popular books are: ")
popularity_based(Final_Dataset, number)
```

```
Top 5 Popular books are:
```

	ISBN	Book-Rating	Book-Title	Book-Author	Year-Of-Publication	Publisher
0	0316666343	707	The Lovely Bones: A Novel	Alice Sebold	2002.0	Little, Brown
1	0971880107	581	Wild Animus	Rich Shapero	2004.0	Too Far
2	0385504209	487	The Da Vinci Code	Dan Brown	2003.0	Doubleday
3	0312195516	383	The Red Tent (Bestselling Backlist)	Anita Diamant	1998.0	Picador USA
4	0060928336	320	Divine Secrets of the Ya-Ya Sisterhood: A Novel	Rebecca Wells	1997.0	Perennial

Recommended Books by same author, publisher of given book name

```
Books by same Author:

Harry Potter and the Goblet of Fire (Book 4)
Harry Potter y el cÃ¡liz de fuego
Harry Potter and the Order of the Phoenix (Book 5)
Harry Potter and the Chamber of Secrets (Book 2)
Harry Potter and the Prisoner of Azkaban (Book 3)


Books by same Publisher:

The Seeing Stone
The Slightly True Story of Cedar B. Hartley: Who Planned to Live an Unusual Life
Harry Potter and the Chamber of Secrets (Harry Potter)
The Story of the Seagull and the Cat Who Taught Her To Fly
The Mouse and His Child
```

Average Weighted Rating

Weighted score can be calculated using the below formula for all the books and recommended the books with the highest score.

$$score= t/(t+m)*a + m/(m+t)*c$$

where,

t represents the total number of ratings received by the book

m represents the minimum number of total ratings considered to be included a represents the average rating of the book and,

c represents the mean rating of all the books.

Here we used 90th percentile as our cutoff. In other words, for a book to feature in the charts, it must have more votes than at least 90% of the books in the list.

```
C= Final_Dataset['Avg_Rating'].mean()
m= Final_Dataset['Total_No_Of_Users_Rated'].quantile(0.90)
Top_Books = Final_Dataset.loc[Final_Dataset['Total_No_Of_Users_Rated'] >= m]
print(f'C={C} , m={m}')
Top_Books.shape
```

```
C=7.626700569504765 , m=64.0

(38570, 11)
```

We see that there are 38570 books which qualify to be in this list. Now, we need to calculate our metric for each qualified book. To do this, we will define a function, **weighted_rating()** and define a new feature score, of which we'll calculate the value by applying this function to our DataFrame of qualified books:

	Book-Title	Total_No_Of_Users_Rated	Avg_Rating	Score
0	Harry Potter and the Goblet of Fire (Book 4)	137	9.262774	8.741835
1	Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))	313	8.939297	8.716469
2	Harry Potter and the Order of the Phoenix (Book 5)	206	9.033981	8.700403
3	To Kill a Mockingbird	214	8.943925	8.640679
4	Harry Potter and the Prisoner of Azkaban (Book 3)	133	9.082707	8.609690

Content based Recommendation

Content-based filtering algorithms are designed to recommend products based on the accumulated knowledge of users. It is crucial to include an important characteristic of products in the system because this technique compares user interest with product attributes. Prior to building a system, choosing each buyer's preferred features should be the top priority. For this filtering, it is necessary to employ these two procedures. The user is first given a list of features from which to choose the most appealing features. Second, the algorithms compile the customer's behavioural data by keeping track of all the products the user has previously selected. Content based recommendation system filter the entire set of books from the dataset based on the content of the book, where buyer is interested to buy.

```
Recommended Books:

Harry Potter and the Sorcerer's Stone (Book 1)
Harry Potter and the Goblet of Fire (Book 4)
Harry Potter and the Chamber of Secrets (Book 2)
Harry Potter and the Prisoner of Azkaban (Book 3)
Harry Potter and the Order of the Phoenix (Book 5)
```

K-Nearest Neighbour algorithm

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

```
Recommended books:

Harry Potter and the Chamber of Secrets (Book 2)
Harry Potter and the Prisoner of Azkaban (Book 3)
Harry Potter and the Goblet of Fire (Book 4)
Harry Potter and the Order of the Phoenix (Book 5)
The Fellowship of the Ring (The Lord of the Rings, Part 1)
```

Conclusion

Majority of the readers were of the age bracket 20-35 and most of them came from North American and European countries namely USA,Canada,UK, Germany and Spain. If we look at the ratings distribution, most of the books have high ratings with maximum books being rated 8. Ratings below 5 are few in number.Author with the most books was Agatha Christie,William Shakespeare and Stephen King.

References

[1] T. Fujimoto and H. Murakami, "A Book Recommendation System Considering Contents and Emotions of User Interests," 2022 12th International Congress on Advanced Applied Informatics (IIAI-AAI), 2022, pp. 154-157, doi: 10.1109/IIAIAAI55812.2022.00039.

[2] J. Qi, S. Liu, Y. Song and X. Liu, "Research on Personalized Book Recommendation Model for New Readers," 2018 3rd International Conference on Information Systems Engineering (ICISE), 2018, pp. 78-81, doi: 10.1109/ICISE.2018.00022.

[3] P. Devika, K. Jyothisree, P. Rahul, S. Arjun and J. Narayanan, "Book Recommendation System," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021, pp. 1-5, doi: 10.1109/ICCCNT51525.2021.9579647.

[4] P. Mathew, B. Kuriakose and V. Hegde, "Book Recommendation System through content based and collaborative filtering method," 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), 2016, pp. 47-52, doi: 10.1109/SAPIENCE.2016.7684166.

[5] Y. Lu and Y. Lu, "Book recommendation system based on an optimized collaborative filtering algorithm," 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA), 2022, pp. 1-4, doi: 10.1109/CVIDLICCEA56201.2022.9824088.

[6] S. J. Mehta and J. Javia, "Threshold based KNN for fast and more accurate recommendations," 2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS), 2015, pp. 109-113, doi: 10.1109/ReTIS.2015.7232862.

[7] N. Kurmashov, K. Latuta and A. Nussipbekov, "Online book recommendation system," 2015 Twelve International Conference on Electronics Computer and Computation (ICECCO), 2015, pp. 1-4, doi: 10.1109/ICECCO.2015.7416895.

[8] https://www.ijresm.com/Vol.2_2019/Vol2_Iss6_June19/IJRESM_V2_I6_91.pdf

[9] https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset