Book Recommendation System

Clesh Mayoulou[1], Mahamat Habib[2], Bahar Youssouf[3]
Computer Science & IT Jain (Deemed-to-be University)

Abstract— The Academic Book Recommendation System is a web/application platform designed to assist users in discovering and categorizing books based on their contents within the academic domain. This system will use the content-based recommendation system to filter out the books of the same contents and group them in the same categories. It will analyze the subject matter, themes, and key concepts of each book, providing users with personalized and relevant recommendations. It will enable personalized recommendations where the user is able to search a book by typing the title of the book, name of the author, or category in which the book belongs (e.g., programming, business, science, mathematics, arts, and language). Users will be able to find the books of their preference based on the contents they are interested in. Our Academic recommendation system will provide features like author's names, book abstracts, and book descriptions. To access our web platform, users have to register, log in to their account, and access the contents. Users will be able to browse through multiple subjects and find some recommended books based on their interests. Features on a user-registered account include wish listing, sharing, and feedback. To implement our project, we will make use of Content-Based Filtering (CBF), which recommends books by comparing the content of the books to user preferences.

I. INTRODUCTION

The Academic Book Recommendation System represents a state of the art web-based platform meticulously engineered to facilitate the discovery of academic books precisely tailored to the nuanced interests of students, researchers, and academicians. In today's dynamic academic milieu, the capacity to access pertinent resources with both speed and efficacy is not merely advantageous but indispensable.

Collaborative filtering and content-based filtering are two main approaches. The example should integrate both. For collaborative filtering, matrix factorization with Surprise makes sense. Content-based can use TF-IDF or something similar on book descriptions.
The user probably wants to see actual code snippets. I'll need to include data loading, preprocessing, model training, and combining the recommendations. They might want to know how to evaluate the model too, so adding RMSE or other metrics would be helpful.

The system provides prompt recommendations for any services and goods the customer requests "Whatever and Whenever" thanks to these portable technologies. Having a observation on the scenario when a user wishes to choose an e-book or other reading content but lacks the necessary personal experience. Unfortunately, because there were so many things or e-resources available, they were unable to find the right ones. The system that suggests is what we have suggested to avoid this situation. Fig.1 depicts the flow of the recommender system

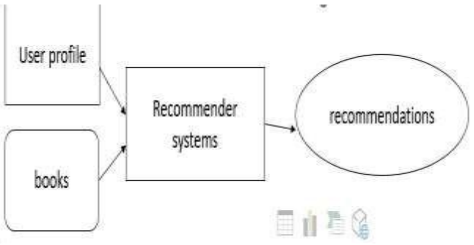


Fig. 1. Flow of a Recommender System

RS was first introduced by Resnick and Varion [1] for developing a collaborative filtering method [2]. Eventually, several researchers utilized the phrase and gave it different definitions. On the internet, numerous recommendation systems are presented and used in different sectors, and each one aims to accurately anticipate in accordance with user interest. Several researchers also looked into the novelty, privacy, scalability, accuracy, usability, and other aspects of RS. Many methods, including collaborative filtering, knowledge-based, demographic, and hybrid methods, can be used to make recommendations.

Collaborative Filtering (CF) is the method for creating recommendations that is most generally used and approved [3]. The CF technique finds comparable products that other users with similar interests favor based on prior interactions (rating, remark) [4]. Based on profile attributes, content-based filtering determines the user's choice based on the items' contents [5]. Knowledge-based works rely on knowledge expertise regarding the user and the item and suggest product based on which items are connected to the user's interests. When suggesting an item, Demographic RS bases its operation on user demographic information from their profiles (such as gender, age, location, etc.).

This article sections are structed as follows: A literature Review on recommender systems is included in Part II. many similarity methods engaged in the RS are presented in Section III. The suggested model of the suggestion of books for online learning is illustrated in Section IV and the words are briefly defined in Section V. The dataset, experimental evaluation, and result analysis are all described in Part VI. The section of the essay that concludes is Section VII.

II. LITERATURE REVIEW

The field of recommendation system has expanded significantly over the past 20 years. Customers have overloaded by the information which is addressed by offering unique, specialized material and repair advice. A few modern cutting-edge techniques are shown in the analysis sections.

Proposal frameworks deal with providing relevant advice in surprisingly intelligent ways. Typically, association rules, content-based filtering, and collaborative filtering are used. Moreover, a straightforward recommendation system for mobile applications is created without sacrificing rating, size, or permission features. There are two models provided, one based on worker incentive features and the other on worker job features. Using real-world datasets, the suggest strategy is compared to several connected methods.

Users decide on a set of data through collaborative filtering. When creating the user want facts set to represent the different facets of the user's profile, special characteristics are taken into consideration. It is a technique for providing users with up-to-date hints from a dataset based on how closely their side interest profiles resemble another user. It is quite tough to side-features for a product or query, and clean that technology cannot be controlled. Customers' information is used by a content-based recommender, either openly or through verification. A user profile is created using this data, and it is then utilized to provide recommendations to the user. The algorithm recognizes customer’s various interests and make suggestions based on the interests of the similar user interests. The capacity of this paradigm to increase the users' present benefits is confined. A huge data set of things can be explored using association rule mining to uncover fascinating associations and interactions between the products. It is a technique for searching for recurrent patterns, parallels, or similarities between datasets from different databases, including relational databases, transactional databases, and various kinds of repositories.

The foremost weakness of the prior methods is the only address one application, namely, an existing client receiving recommendations from the dataset, and do not address usage scenarios like proposing a new book to a new customer or a generic counsel based on the dataset. Our system includes each of these programs, giving users a selection of options for picking the ideal book to read.

The primary shortcoming of the earlier solutions is that they only handle one application, namely, a current customer receiving suggestions from the dataset and do not deal with use scenarios such as recommending a new book to a brand-new customer, or a generic advice based on the dataset. Each of these applications is included in our system, providing customers with a variety of options for selecting the appropriate book to read.

III. METHODOLOGY

Algorithm used in "Book Recommendation System" initiative aims to assist users in selecting the right choice of books that piques their interest and so motivate them to learn more. In this case, we're using Cosine similarity, KNN, and Pearson correlation. Seeing similarities across the books is becoming more and more routine. As previously indicated, the major three use cases for this system are suggestions for current users, recommendations for new users, and ratings for newly uploaded books. Several approaches are used to deal with each of these. The main method used in this project is collaborative filtering based on users. Depending upon the ratings given to the product by another reader who share the target user's preferences, the system predicts what a user will like. Fig 2 depicts the difference between the content based filtering and collaborative based filtering.

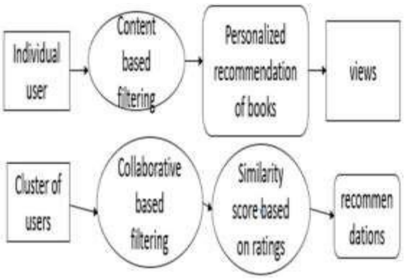

Individual user
Content based filtering
Personalized recommendation of books
views
Cluster of users
Collaborative based filtering
Similarity score based on ratings
recommendations

Fig. 2. Content based vs Collaborative based Filtering

A. Dataset Description

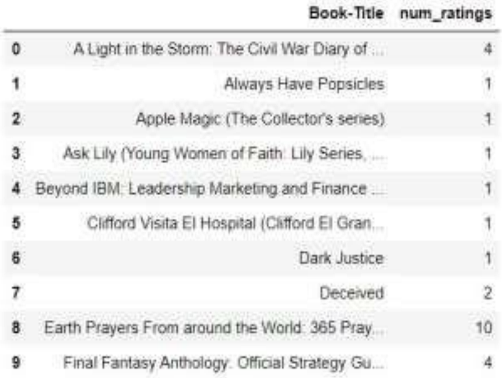
For the dataset, three csv files are taken. First one is the books file. The attributes of the books dataset are ISBN, Book title, Book author, year of publication, publisher, image URL small, image URL medium, image URL large. The second dataset is user's dataset. The user's dataset contains the user ID, location, age. The third dataset is the ratings dataset. The ratings dataset contains the user ID, ISBN of the book and the ratings provided by the users for the particular book.

B. Algorithm

1)Pearson correlation: To gauge the linear correlation between variables, co-efficient of Pearson R correlation is utilized. Its purpose is to build a recommendation using rating counts.

depending upon the ratings and the count of rating, a graph of rating distribution is drawn, and most individuals have given ratings of 0.

In order to obtain the count in desc order, first group the rating data frame by ISBN, then take the book rating column. Hence, if more people rate a book, it must be a very well-known book overall. In order to do that, we have created a data frame using all the ISBN numbers, then combined it with the books dataset's most popular titles using the ISBN field. Therefore, in conclusion, we can find the top 5 books based on the number of ratings here. The ratings mean as well as the ratings count were discovered for the correlation. Then, present the data in desc order after generating two fields in data frames: rating of the book and rating count. As a result, if we are using the rating count to determine popularity and my book does not have a decent rating but the greatest number of people have given it a rating, it cannot be a widely read book, so we should not promote it. In order for them to make recommendations, they must take into account the rating average and the amount of ratings. We have some factual importance where users with less than 200 evaluations and books with fewer than 100 evaluations are disallowed in contemplation to create the best recommendation system. At that time, having the opinion to combine user id and ISBN using the pivot table; therefore, by using the indexes as user id on the column features, it would show if the individual has provided any ratings or not. When the pivot is used, the rating table is actually transformed into a 2D matrix if the user has not provided any ratings, which is basically represented as nan. Then a correlation between the ratings and the ratings average is discovered. Fig 3 depicts the output of pearson algorithm.



	Book-Title	num_ratings
0	A Light in the Storm: The Civil War Diary of ...	4
1	Always Have Popsicles	1
2	Apple Magic (The Collector's series)	1
3	Ask Lily (Young Women of Faith: Lily Series, ...	1
4	Beyond IBM: Leadership Marketing and Finance ...	1
5	Clifford Visita El Hospital (Clifford El Gran...	1
6	Dark Justice	1
7	Deceived	2
8	Earth Prayers From around the World: 365 Pray...	10
9	Final Fantasy Anthology: Official Strategy Gu...	4

Fig. 3. Output of the Pearson algorithm

2) KNN: KNN is a machine learning estimator used to find set of alike readers depending upon basic book ratings and provide data using the average rating of the top k nearest neighbors. KNN is a machine learning estimator used to find groups of similar users based on basic book ratings and provide data utilizing the average rating of the highest k nearest neighbors.

In this suggestion system, similar readers are found using similarity of cosine by converting the table to a two-dimensional matrix and then filling in the gaps with NULL. The network data frame's values (ratings) were then converted into a SciPy sparse grid for more accurate computations. Finally, this computation will determine the similarity of the cosine between rating vectors. Unsupervised techniques with sklearn.neighbors are used to get the Closest Neighbors.

To determine how comparable two items are, one can use the cosine similarity measurement. In terms of numbers, it calculates the intersection cosine of two vectors that are visualized into a multidimensional space. The yield value ranges from 0 to 1. (0 denotes the absence of comparability, while 1 suggests that both elements are equally rated at 100.

A collaborative filtering recommendation system based on items is developed using KNN. The five books with the highest similarity scores to the selected book are listed. Since similarity is determined by distance, the higher the value, the less similarity there is. Following that, they are put in ascending order. When a user types in their favorite book from the dataset, the algorithm suggests the user's next five most related novels.

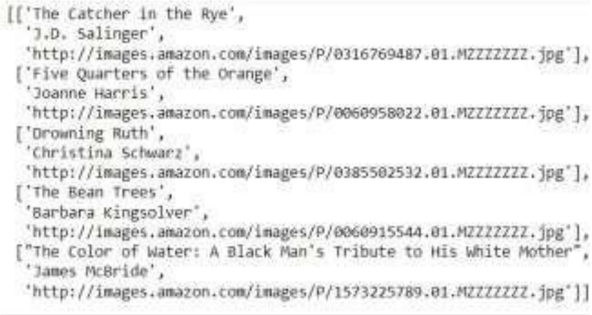
3) Collaborative filtering: Collaborative filtering is one of the methods we use to forecast the book ratings from an already subscribed user by computing the resemblance among readers. The fundamental concept of collaborative filtering is this. Let us say we want to make recommendations for user x. To start, we identify a lot of other users that have user x's likes and dislikes. K Nearest Neighbor is used to accomplish this. An unsupervised learning algorithm is nearest neighbor. It is imported as sklearn.neighbors from the sklearn library. When determining users who are similar, the user's previous ratings of the books are considered.

Three new features were developed. One to find the KNearest Neighbors, second is to forecast the average ratings of a specific book, and the third is to recommend the user the best-rated books. This information is transformed into a cross tabulation with ISBN columns and the id of the user as the index, and all 0 ratings are eliminated. The data is essentially represented as a matrix in the pivot table. It is determined how many books each user has reviewed. The system receives a user's distinctive ID, a distinctive ISBN of book, the number of products the reader has reviewed, and the cross tabulation of ratings.

The standard deviation and the user ratings mean are used to calculate the book ratings. The average of all the user-rated books is represented by the mean rating. The weighted average is calculated using both the mean rate of the mean of the comparable readers of the given book and the measure of their similarity. The distance produced via K Nearest Neighbor is used to determine the similarity value.

Prediction Rate = Rate of the Mean + Standard Deviation.

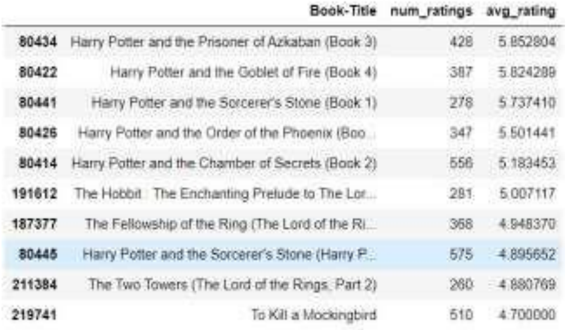
The user is then shown the top 10 books after the ratings are organized in descending order. If those projected ratings are fewer than 6, a message asking the user to examine some general suggestions determined by Pearson correlation is displayed. Fig 4 depicts the output of collaborative Filtering output



```
[['The Catcher in the Rye',
  'J.D. Salinger',
  'http://images.amazon.com/images/P/0316769487.01.MZZZZZZZ.jpg'],
 ['Five Quarters of the Orange',
  'Joanne Harris',
  'http://images.amazon.com/images/P/0060958022.01.MZZZZZZZ.jpg'],
 ['Drowning Ruth',
  'Christina Schwarz',
  'http://images.amazon.com/images/P/0385502532.01.MZZZZZZZ.jpg'],
 ['The Bean Trees',
  'Barbara Kingsolver',
  'http://images.amazon.com/images/P/0060915544.01.MZZZZZZZ.jpg'],
 ["The Color of Water: A Black Man's Tribute to His White Mother",
  'James McBride',
  'http://images.amazon.com/images/P/1573225789.01.MZZZZZZZ.jpg']]
```

Fig. 4. Collaborative Filtering output

4) Suggestion of books to the New Users: A few books new user who has been added to this recommendation system. By dividing the total number of ratings for each book by the total number of ratings, the average ratings are determined. Afterwards, they are arranged in the order of decrement. Following the users' ratings of these books, collaborative filtering Future recommendations for that person can be enhanced using this rating. The top 10 books, as determined by the total average rating of all the books, are presented to each should also be recommended to any new users who need to be added to the data set so that they can rate them.

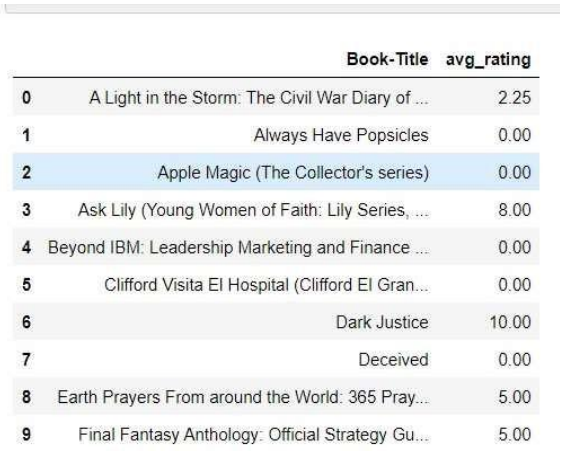


	Book-Title	num_ratings	avg_rating
80434	Harry Potter and the Prisoner of Azkaban (Book 3)	428	5.852804
80422	Harry Potter and the Goblet of Fire (Book 4)	387	5.824289
80441	Harry Potter and the Sorcerer's Stone (Book 1)	278	5.737410
80426	Harry Potter and the Order of the Phoenix (Boo...	347	5.501441
80414	Harry Potter and the Chamber of Secrets (Book 2)	556	5.183453
191612	The Hobbit : The Enchanting Prelude to The Lor...	281	5.007117
187377	The Fellowship of the Ring (The Lord of the Ri...	368	4.948370
80445	Harry Potter and the Sorcerer's Stone (Harry P...	575	4.895652
211384	The Two Towers (The Lord of the Rings, Part 2)	260	4.880769
219741	To Kill a Mockingbird	510	4.700000

Fig.5 Highest average rated books

5) Inserting New Book: To be considered for a user suggestion and to be part of the data set, a new book needs to have a rating. The average ratings of all the author's other books that are already part of the data set are used to compute this score. According to their

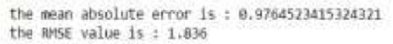
author, books are sorted. After adding up all of the ratings for each book, a percentage of the total ratings is calculated. It is also possible to see how many people read the book by that author by looking at the overall rating count. The new book that was added to the data collection is then given this rating as its initial rating.



	Book-Title	avg_rating
0	A Light in the Storm: The Civil War Diary of ...	2.25
1	Always Have Popsicles	0.00
2	Apple Magic (The Collector's series)	0.00
3	Ask Lily (Young Women of Faith: Lily Series, ...	8.00
4	Beyond IBM: Leadership Marketing and Finance ...	0.00
5	Clifford Visita El Hospital (Clifford El Gran...	0.00
6	Dark Justice	10.00
7	Deceived	0.00
8	Earth Prayers From around the World: 365 Pray...	5.00
9	Final Fantasy Anthology: Official Strategy Gu...	5.00

Fig. 6. Highest-rated authors in terms of average

From the ratings pivot table five readers were chosen randomly . The five users anticipated and saved ratings of the previously reviewed books . Also, each of the dataset's actual evaluations was saved. There were 666 total ratings. These served as a basis for calculating the Root Mean Squared Error and Mean Absolute Error values.



```
the mean absolute error is : 0.9764523415324321
the RMSE value is : 1.836
```

Fig. 7. RMSE and MAE

From the Fig.7 Root Mean Squared Error value is equal to 1.836 and Mean Absolute Error values is equal to 0.976. RMSE and MAE are two different measures of error, a numerical comparison between them (which is involved in asserting that MAE is "lower" than RMSE) does not seem meaningful. That line must have been fit according to some criterion: that criterion, whatever it is, must be the relevant measure of error.

IV. EXPERIMENTAL SETUP AND RESULTS

A. Performance Assessment

Two measures are used to evaluate the collaborative filtering's performance: -

1)RMSE: RMSE is a widely used and accepted approach for determining a model's error.

2) MAE: The only difference between this method and the former is that with MAE, In the case of linear continuous variables, the error is estimated.

V. CONCLUSIONS AND FUTURE WORK

This research suggests a machine learning algorithm named collaborative filtering mechanism for a recommendation of books. By giving book recommendations, people's reading habits improve, which boosts their vocabulary, expertise, and knowledge. This research proves the efficiency of the distributed and parallel computing concepts for e-commerce enterprises and shows that these concepts can be applied in various data-driven system designs. In addition to e-commerce, the proposed framework is also relevant to other business models in the news, hospitality, and music industries. Furthermore, in information systems, query data processing can also use this framework to increase scalability in processing time and decrease the opportunity cost inherent in waiting times. Our designed system makes the most of the information's unique alternatives to provide a speedy response and high-caliber recommendations. Collaborative Filtering is the algorithm used by the system to generate recommendations. The cosine similarity approach is used to precisely quantify the similarities between the users. Based on the average ratings calculated and gathered from the various users the top rated gooks are suggested for the book readers.

This paper proposes a new architecture for a product recommendation system in a large e-commerce environment, utilizing a collaborative filtering algorithm implemented in Apache Spark and cloud environments. While KNN is more straightforward and may work well in smaller datasets or scenarios with dense data, it struggles with computational overhead as dataset sizes grow. It offers complete satisfaction.

REFERENCES

[1]. Kurmashov, Nursultan, Konstantin Latuta, and Abay Nussipbekov. ”Online book recommendation System.” In 2015 Twelve International Conference on Electronics Computer and Computation (ICECCO), pp. 1-4. IEEE, 2015.

[2]. Tewari, Anand Shanker, and Kumari Priyanka. ”Book recommendation system based on collaborative filtering and association rule

mining for college students.” In 2014 International Conference on Contemporary Computing and Informatics (IC3I), pp. 135-138. IEEE, 2014.

[3]. Mohamed, Marwa Hussien, Mohamed Helmy Khafagy, and Mohamed Hasan Ibrahim. ” Recommender systems challenges and solutions survey.” In 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), pp. 149-155. IEEE, 2019.

[4]. Badriyah, Tessy, Erry Tri Wijayanto, Iwan Syarif, and Prima Kristalina. ”A hybrid recommendation system for E-commerce based on product description and user profile.” In 2017 Seventh International Conference on Innovative Computing Technology (INTECH), pp. 95-100. IEEE, 2017.

[5]. Ahuja, Rishabh, Arun Solanki, and Anand Nayyar.” Movie recommender system using KMeans clustering and K-Nearest Neighbor.” In 2019 9th International Conference on Cloud Computing, Data Science Engineering (Confluence), pp. 263-268. IEEE, 2019.

[6]. M¨obius Data Scientist at Healthcare Melbourne, Victoria, Australia” Book Recommendation Dataset Build state-of-the-art models for book recommendation system”, kaggle, 2016.

[7]. Kurup, Ayswarya R., and G. P. Sajeev. ”Task recommendation in rewardbased crowdsourcing systems.” In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1511-1518. IEEE, 2017.

[8]. Keerthana, N. K., Shriram K. Vasudevan, and Nalini Sampath. ”An Effective Approach to Cluster Customers with a Product Recommendation System.” Journal of Computational and Theoretical Nanoscience Vol. 17, No. 1, pp. 347-352.IEEE, 2020.

[9]. Jisha, R. C., Ram Krishnan, and Varun Vikraman. ”Mobile applications recommendation based on user ratings and permissions.” In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 10001005.IEEE, 2018.

[10]. Murali, M. Viswa, T. G. Vishnu, and Nancy Victor. ”A Collaborative Filtering based Recommender System for Suggesting New Trends in Any Domain of Research.” In 2019 5th International Conference on Advanced Computing Communication Systems (ICACCS), pp. 550- 553. IEEE, 2019.