BOOK RECOMMENDATION SYSTEM

PROJECT REPORT

SUBMITTED TO : ZAKARIA SHAMS SIAM

SUBMISSION DATE : 11/06/2024

KEYWORDS

1. Book Recommendation
2. Collaborative Filtering
3. Content-Based Filtering
4. User-Based Collaborative Filtering
5. Item-Based Collaborative Filtering
6. Machine learning
7. User preferences

ABSTRACT

The present world is filled with a vast amount of knowledge and information. As a result, information overload is created. To solve this issue, researchers and computer science and technology developers have created a new solution: recommendation systems. It is a machine learning class that offers necessary suggestions to the people. It has overcome this issue by giving users personalized recommendations based on their opinions and preferences.

Before the recommendation system, people normally took suggestions from their family members, friends, or close ones. But Now social media like Facebook, Twitter, and Instagram know what type of posts we will see, Google knows what kind of information we need and YouTube knows what sort of videos we will watch based on our search history and uses

It helps to create loyal customers and creates trust. Recently, they have become so powerful that they can handle the customers who have entered the site as new members. They also recommend currently trending products.

In the world of machine learning, they are currently one of the most successful applications that is getting widely known and popular

There are millions of books and new books are being created more and more every day, offering a sea of choices. So, it's getting absolutely challenging for people to find the right book based on their choices. Sometimes, making decisions and scrolling for the appropriate book may take so long that it could be time-consuming.

For this, they definitely need the help of a Recommendation system

This project addresses this issue and aims to solve it by implementing and focusing on a collaborative filtering approach. The project offers a quick and intuitive book recommender system that provides personalized book recommendations for each user.

The results highlight the strengths and limitations of each approach, paving the way for future enhancements in book recommendation systems.

The system evaluation results show that the accuracy of the system recommendations is perfect and that a recommender system based on the combination of content-based and collaborative filtering approaches provides more accurate recommendations for the book domain.

INDEX

1. Introduction

Recently, Recommendation systems have been widely used in several different services. From online shopping to book recommendations to music to movies. For instance, music streaming services, such as Spotify, heavily depends on the music selections of similar users to make weekly song recommendations and personalized radio stations. popular television and movie streaming services like Netflix, Amazon Video, Disney+, and HBO Max uses these systems to recommend movies for viewers to enjoy. We can clearly see how recommendation systems have greatly influenced the consumers engaging in their daily lives.

The advantages of this book recommendation system are many. It saves the precious time of customers and is very efficient to use. It provides a large number of choices for books & also recommends books. Users can buy books easily by making online payment
The system-recommending algorithm scales well with co-rated items. The disadvantage is dependent on human ratings for books

In today's digital age, the large number of available books makes it difficult for readers to discover new titles that match their preferences. Recommendation systems provide a solution to this issue by using user data to offer personalized suggestions. Collaborative Filtering (CF) is a popular technique in recommendation systems. It analyzes user behavior to identify patterns and similarities between users or items. This project is focused on creating a book recommendation system using collaborative filtering methods to help users find books that align with their interests.

One of the main motivation for us to work on this project is the fact that direct explicit rating carry less information and are, in a way, a form of information repetition and redundancy. A rating of four may sound fairly good for some users but others may take the same rating harshly.

In this regard, we are willing to analyze the sentiments of reviews made by users on each book and turn that into ratings. Moreover, users hardly give rating explicitly whether it is in the form of stars or reviews. So, we have also taken the user clicks on a particular book into account.

Therefore, the weighted average mean of user clicks, rating, and reviews are taken with priorities in the order of review, rating, and user clicks with review carrying the highest priority, and this average mean is inserted into the user-item interaction matrix.

Finally, books having the highest predicted rating are recommended to the users.

2. Literature Review

Recommender systems are one of the most successful and widespread applications of machine learning.

The first recommender system named “Grundy” was created in 1979 by Elaine Rich. It was the first book recommendation system. She looked for a way to recommend users books they might like. She built a system that asks users specific questions and classifies them into classes depending on their answers. Based on it, they would then get recommendations for books they might like.

Another early recommender system, called a "digital bookshelf", was described in a 1990 technical report by Jussi Karlgren and implemented and worked through from 1994 onwards. Since then the development of making the recommendation system better has been ongoing till now and has been developed.

Recommendations are based on user's past likes/ dislikes & their ratings on other items. The system. These take either a memory-based approach or a model-based approach:

(1) Memory-based approach: Utilizes entire user-item rating information to calculate similarity scores between items or users for making recommendations. These are further either of 2 types:

- **User-based**: Two users are considered similar if their ratings are similar. An item is recommended to a user if he likes the same item another user has.
- **Item-based**: Two items are considered similar if users similarly rate them. An item is recommended to a user based on their past ratings.

(2) Model-based approach: Utilizes user-item rating information to build a model & the model (not the entire dataset) is thereafter used for making recommendations. This approach is preferred in instances where time & scalability are a concern.

Previous research in recommendation systems has explored various approaches and encompasses a wide range of techniques and methodologies including- collaborative filtering, content-based filtering, hybrid systems and deep learning approaches.

Previous studies have evaluated the performance of these different new methods using separate datasets and evaluation metrics, highlighting their strengths and limitations.

While collaborative filtering methods excel in capturing user preferences and generating personalized recommendations, content-based filtering approaches offer advantages in handling cold-start problems and recommending less popular items. Content-based filtering requires rich descriptions of items and a well-organized user profile before making recommendations whereas collaborative filtering suffers from the cold start problem. The hybrid method uses the best of both but tends to be complex. Hybrid systems, combining collaborative and content-based techniques, aim to leverage the complementary

strengths of both approaches to improve recommendation accuracy and diversity

The collaborative filtering method is used to identify relationships between pieces of data. It is used to identify the similarities between user data and items. is a technique that can filter out items that a user might like based on reactions by similar users. It works by searching a large group of people and finding a smaller set of users with tastes similar to a particular user.

Collaborative filtering methods, particularly user-based and item-based approaches, have gained prominence due to their simplicity and effectiveness. User-based collaborative filtering recommends items to a user based on the preferences of users with similar tastes, while item-based collaborative filtering suggests items similar to those the user has previously interacted with. These methods have been applied in diverse domains such as e-commerce, music, and movie recommendations, demonstrating their versatility and applicability

3. Methodology

To build a book recommendation system, a comprehensive methodology was developed. The methodology involves data collection, preprocessing, model development, feature extraction, model training, evaluation metrics, and implementation. Evaluation metrics such as precision, recall, and mean average precision are used to assess the performance of the recommendation models and compare their effectiveness.

The dataset comprises user-book interactions, including ratings and reviews, which are preprocessed to handle missing values and normalize ratings. The collaborative filtering model is developed using item-based filtering because of its efficiency. Similarity scores between books are computed based on various factors such as genre, author, and user ratings. Recommendations are generated for each user by selecting books with the highest similarity scores to those the user has enjoyed previously.

Furthermore, it is important to account for both implicit and explicit feedback from the user. The sentiment of book reviews, along with ratings and clicks, is analyzed to generate a weighted average mean rating. This is used in a user-item interaction matrix for generating recommendations via matrix factorization, eliminating the need for detailed item data and user information.

Types of Collaborative Filtering

There are two main types of collaborative filtering:

1. **User-Based Collaborative Filtering**
2. **Item-Based Collaborative Filtering**

User-Based Collaborative Filtering

This approach recommends items by finding users similar to the target user and recommending items that those similar users have liked.

Example: Imagine a book recommendation system. Here's how user-based collaborative filtering works:

1. **User Preferences:**
    - User A likes books X, Y, Z.
    - User B likes books X, Y.
    - User C likes books X, Z.
    - User D likes books Y, Z.
    - User E likes books X, Y, Z.
2. **Finding Similar Users:**
    - To recommend a book to User F, we first look for users with similar preferences.
    - Suppose User F likes books X and Y.
    - We see that Users A, B, and E also like books X and Y.
3. **Making Recommendations:**

- Since Users A, B, and E have similar tastes to User F, we can recommend books that these users like but User F hasn't read yet.
- If User A, B, and E also like book Z, we can recommend book Z to User F.

Item-Based Collaborative Filtering

This approach recommends items by looking at the similarity between items and recommending items similar to those the target user has liked.

Example: Using the same book recommendation system:

1. **User Preferences:**
   - User A likes books X, Y, Z.
   - User B likes books X, Y.
   - User C likes books X, Z.
   - User D likes books Y, Z.
   - User E likes books X, Y, Z.
2. **Finding Similar Items:**
   - To recommend a book similar to what User F likes, we look at the books User F has rated highly.
   - Suppose User F likes book X.
3. **Making Recommendations:**
   - We find that books Y and Z are often liked by users who like book X.
   - Therefore, we recommend books Y and Z to User F.

Popularity Based Recommender System

[41]:
```
ratings_with_name = ratings.merge(books,on='ISBN')
ratings_with_name
```

[41]:

	User-ID	ISBN	Book-Rating	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	
0	276725	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/ir
1	2313	034545104X	5	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/ir
2	6543	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/ir
3	8680	034545104X	5	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/ir
4	10314	034545104X	9	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books	http://images.amazon.com/images/P/034545104X.0...	http://images.amazon.com/ir
...	...	...	...	...	...	...	...	...	...

[16]:
```
num_rating_df = ratings_with_name.groupby('Book-Title').count()['Book-Rating'].reset_index()
num_rating_df.rename(columns={'Book-Rating':'num_ratings'},inplace=True)
num_rating_df
```

[95]:
```
popular_df = num_rating_df.merge(avg_rating_df,on='Book-Title')
popular_df
```

[95]:

	Book-Title	num_ratings	avg_rating
0	A Light in the Storm: The Civil War Diary of ...	4	2.250000
1	Always Have Popsicles	1	0.000000
2	Apple Magic (The Collector's series)	1	0.000000
3	Ask Lily (Young Women of Faith: Lily Series, ...	1	8.000000
4	Beyond IBM: Leadership Marketing and Finance ...	1	0.000000
...	...	...	...
241066	Ã?Â?lpiraten.	2	0.000000
241067	Ã?Â?rger mit Produkt X. Roman.	4	5.250000
241068	Ã?Â?sterlich leben.	1	7.000000
241069	Ã?Â?stlich der Berge.	3	2.666667
241070	Ã?Â?thique en toc	2	4.000000

241071 rows × 3 columns

[17]:
```
avg_rating_df = ratings_with_name.groupby('Book-Title')['Book-Rating'].agg(lambda x: x.astype(float).mean()).reset_index()
avg_rating_df.rename(columns = {'Book-Rating' : 'avg_rating'}, inplace = True)
avg_rating_df
```

[17]:

	Book-Title	avg_rating
0	A Light in the Storm: The Civil War Diary of ...	2.250000
1	Always Have Popsicles	0.000000
2	Apple Magic (The Collector's series)	0.000000
3	Ask Lily (Young Women of Faith: Lily Series, ...	8.000000
4	Beyond IBM: Leadership Marketing and Finance ...	0.000000
...	...	...
241066	Ã?Â?lpiraten.	0.000000
241067	Ã?Â?rger mit Produkt X. Roman.	5.250000
241068	Ã?Â?sterlich leben.	7.000000

[95]:
```
popular_df = num_rating_df.merge(avg_rating_df,on='Book-Title')
popular_df
```

[95]: Book-Title num_ratings avg_rating

[91]:

User-ID	254	2276	2766	2977	3363	4017	4385	6251	6323	6543	...	271705	273979	274004	274061	274301	274308	275970	277427	277639	2784
Book-Title																					
1984	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	10.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
1st to Die: A Novel	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	9.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
2nd Chance	0.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
4 Blondes	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
A Bend in the Road	0.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
Year of Wonders	0.0	0.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
You Belong To Me	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
Zen and the Art of Motorcycle Maintenance: An Inquiry into Values	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
Zoya	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0

4. Results And Discussion

The developed recommendation system demonstrates promising results in terms of accuracy and user satisfaction. Users receive personalized recommendations that closely align with their interests and preferences, enhancing their overall reading experience. Evaluation metrics such as precision, recall, and accuracy confirm the effectiveness of the system in providing relevant book suggestions. User feedback and satisfaction surveys further validate the system's performance and relevance.

The study's analysis uncovered key findings about various book recommendation techniques.

Collaborative filtering methods demonstrate strong performance in capturing user preferences and generating accurate recommendations based on past interactions

The discussion highlights the implications of the result for improving recommendation accuracy, diversity, and user satisfaction in real-world book recommendation systems.

5. Limitations & Future Works

While the study has shown promising results, it does have its limitations. The evaluation is based on a specific dataset and may not apply to other domains or datasets with different characteristics. The performance of recommendation systems can vary depending on factors such as dataset size, sparsity, and user engagement levels. Future research should explore advanced techniques including deep learning models, reinforcement learning, and context-aware recommendation approaches to further improve recommendation accuracy and relevance. Furthermore, integrating social and contextual information could enhance the personalization and effectiveness of book recommendation systems.

Some viewers who don't read much may only review one book and biased ratings can mislead the viewers to books they won't like, resulting in the inaccuracy of the book recommendation system to the user. In this case we can implement different approaches to solve these problems.

6. Conclusion

In conclusion, this report provides valuable insights into the design and implementation of advanced book recommendation systems, highlighting the strengths and limitations of existing techniques and methodologies. By leveraging collaborative filtering, developers and practitioners can design recommendation systems that deliver personalized and relevant book recommendations to users. Moving forward, continued research and innovation will be essential for addressing the challenges associated with recommendation accuracy, diversity, and scalability, ultimately enhancing user satisfaction and engagement in digital libraries.

7. Contribution

Name	Student ID	Work Done	Percentage
1. Shihab Sarar(Leader)	221036038	Backend coding of Collaborating Filtering.	25%
2. Joynal Abedin	222253038	Backend coding of Popularity Based Filtering.	25%
3. Shondipon Biswas	222113038	Frontend Using HTML, CSS, BOOTSTRAP, JAVASCRIPT.	25%
4. Aditta Morshed	221211038	Report Writing and Analysis.	25%

Citation:

1. **https://ieeexplore.ieee.org/abstract/document/9441927**
2. **https://nevonprojects.com/online-book-recommendation-system-using-collaborative-filtering/**
3. **https://www.analyticsvidhya.com/blog/2021/06/build-book-recommendation-system-unsupervised-learning-project/**
4. **https://www.scribd.com/document/446262056/ONLINE-BOOK-RECOMMENDATION-SYSTEM**
5. **https://www.studocu.com/en-gb/document/university-of-wolverhampton/modern-computer-science/book-recommendation-final-report/34016309**
6. **https://sist.sathyabama.ac.in/sist_naac/aqar_2022_2023/documents/1.3.4/b.e-cse-batchno-46.pdf**

Github:

https://github.com/ShihabXSarar/Book_Recommender_System-with-website