Fuzzy Systems and Soft Computing
ISSN : 1819-4362

BUILDING A BOOK RECOMMENDATION SYSTEM A JOURNEY THROUGH DATA AND USER PREFERENCES

Dr. B. Vasumathi Associate Professor, Department of Computational Sciences, Brainware University, Kolkata, India
Elisha Das PG Student, Department of Computational Sciences, Brainware University, Kolkata,
Eshita Saha PG Student, Department of Computational Sciences, Brainware University, Kolkata,
Moumita Patra PG Student, Department of Computational Sciences, Brainware University, Kolkata, India
Nabami Pal PG Student, Department of Computational Sciences, Brainware University, Kolkata,

ABSTRACT:
In today's digital age, personalized recommendation systems play a pivotal role in enhancing user experience across various domains, including literature and books. This paper explores the development and implementation of a book recommendation system that leverages data analytics and user preferences. By analysing user behaviour, reading patterns, and book metadata, the system aims to deliver tailored recommendations that cater to individual tastes and interests. Key components such as collaborative filtering, content-based filtering, and hybrid approaches are examined in detail, highlighting their respective strengths and limitations. Furthermore, considerations for data privacy, algorithm transparency, and system scalability are addressed to ensure ethical and efficient recommendation processes. Through a systematic approach to data collection, processing, and evaluation, this study underscores the potential of recommendation system to enrich the literary journey for readers while advancing the field of personalized content delivery.

KEYWORDS: *Machine Learning, Rest Api, Flask Framework, Python.*

OBJECTIVES OF THE PROJECT:
1. **Understanding User preference and Behaviour:** How data on user interaction, reading habits, and demographic information inform personalized recommendations.
2. **Technological Foundation:** Exploration of collaborative filtering, content-based filtering, and hybrid approaches in recommendation algorithms.
3. **Ethical Consideration:** Addressing issues such as data privacy, algorithm transparency, and fairness in recommendation system.
4. **Impact and Future Direction:** Examining the transformative potential of recommendation system in enhancing user satisfaction and engagement in the literary domain.

By examining these facts in depth, this project aims to provide a comprehensive overview of the development and implementation of a book recommendation system. It seeks to offer insights that inform practitioners, researchers, and stakeholders interested in leveraging data-driven approaches to enhance user experience and engagement with literary content. Ultimately, the goal is to contribute to the advancement of personalized content delivery, fostering a more enriched and tailored and reading experience for individuals in an increasingly interconnected digital world.

1. INTRODUCTION TO RECOMMENDATION SYSTEMS:
In today's digital age, where information is abundant and accessible at our fingertips, recommendation system has become integral to enhancing user experience across various domains. From streaming services suggesting movies based on viewing history to social media platforms recommending connections and content, these system leveraging data analytics to personalize interaction and cater to individual preferences. Among these applications, book recommendation systems occupy a unique niche, aiming to navigate the diverse literary landscape and guide readers to discover books that resonate with their interest and tastes.

This project embarks on a comprehensive exploration of the intricacies involved in building a sophisticated book recommendation system. It delves into the intersection of the data analytics and user preferences, aiming to unravel how these elements converge to deliver personalized recommendations in the realm of literature. By analysing user behaviour, reading habits, and book metadata, the system endeavours to decipher patterns and preferences that inform its recommendation algorithms. These algorithms, ranging from collaborative filtering to content-based filtering and hybrid approaches, play a pivotal role in distilling vast dataset into actionable insights that drive personalized content delivery.

2. REVIEW OF RELEVANT RESEARCH PAPERS AND STUDIES:

In the realm of book recommendation systems, researchers have explored various methodologies to enhance the accuracy and personalization of recommendations, catering to diverse user preferences and behaviours. This summary highlights key findings and innovations from recent studies in the field. Campos, Dias, and Jorge (2020) delve into user profile modelling in book recommendation systems, emphasizing the significance of understanding user behaviour and preferences. By leveraging data analytics, their study aims to improve recommendation accuracy by creating robust user profiles that capture individual reading habits and interests (Campos, Dias, & Jorge, 2020).

Kumar and Goyal (2021) conduct a comparative study of collaborative filtering techniques for book recommendations. Their research evaluates the effectiveness of different collaborative filtering algorithms in suggesting relevant books based on user interactions. This comparative analysis provides insights into the strengths and limitations of each technique, helping to optimize recommendation systems (Kumar & Goyal, 2021). Li, Lu, and Li (2022) propose an advanced recommendation algorithm that integrates user interest and content characteristics. Their approach aims to deliver more personalized book recommendations by considering both explicit user preferences and implicit content features. This hybrid approach enhances recommendation relevance, thereby improving user satisfaction and engagement (Li, Lu, & Li, 2022).

Campos, Dias, & Jorge (2020) Focuses on user profile modelling to enhance recommendation accuracy by understanding user behaviour and preferences.

3. OVERVIEW OF EXISTING BOOK RECOMMENDATION SYSTEMS

3.1 Content-Based Filtering:

The algorithm recommends a product that is similar to those which used as watched. In simple words, in this algorithm, we try to find finding item look alike. For example, a person likes to read Harry Potter 1st edition, so he may like reading magic related book too because the two books have similar tags and similar categories.

Only it looks similar between the content and does not focus more on the person who is reviewing this. Only it recommends the product which has the highest score based on past preference.

Advantages:

The model does not need any data about others users, since the recommendations are specific to this user. This makes it easier to scale to a large number of users. The model can capture the specific interest of users, and can recommend niche items that very few other users are interested in.

Disadvantages:

Since the feature representation of the items are hand-engineered to some extent, this technique requires a lot of domain knowledge. Therefore, model can only be as good as the hand engineered features. The model can only make recommendations based on existing interest of user. In other words, the model has limited ability to expand on the users' existing interests.

3.2 Collaborative-Based Filtering:

Collaborative based filtering recommender system are based on past interactions of users and target items. In simple words here, we try search for the look-alike customers and offer products based on what his or her lookalike has chosen. Let us understand with an example. X and Y ae two similar user

and X user has read A, B and C book. And Y user has read B, C and D book then we will recommend A book to Y user and D book to X user.

YouTube has shifted its recommendation system from Content-based to Collaborative based filtering technique. If you have experienced sometimes there are also videos which are not related at all to your history but then also it recommends it because the other person similar to you has watched it.

Advantages:

The primary advantages of collaborative filtering are that shoppers can get broader exposure to many different products, which creates possibilities to encourage shoppers towards continual purchases of products.

Disadvantages:

Data sparsity is seen as a key disadvantage of collaborative filtering. Traditional CF algorithms often suffer serious scalability problems. Collaborative filtering is unable to identify synonyms refer to similar items labelled or named differently. In principle, collaborative filters are expected to enhance diversity as they help consumers discover more products, but some may unintentionally do the complete opposite.

Each of these recommendation systems plays a crucial role in providing personalized book recommendations to users, addressing various challenges and leveraging different approaches to enhance user experience and satisfaction in discovering and enjoying books.

4. REQUIREMENT ANALYSIS:

4.1 Python IDLE Jupyter Notebook:

Jupyter notebooks are used for all sorts of data science tasks such as exploratory data analysis, data cleaning and transformation, data visualization, statistical modelling, machine learning, and deep learning.

Jupyter notebook is an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative texts. On the other hand, Python IDLE is an integrated development environment (IDE) that provides a basic interface for writing and executing Python code.

4.2 Flask Framework:

In this article, we will build a REST API in PYTHON using FLASK Framework. Flask is a popular micro framework for building web applications. Since it is a micro-framework, it is very easy to use and lacks most of the advanced functionality which is found in a full-fledged framework. Therefore, building a REST API in Flask is very simple.

5. METHODOLOGIES:

5.1 Dataset Description:

We have 3 files in our dataset which are extracted from some book selling websites. **Books –** First are about which contain all the

information related to books like author, title, image etc.

User – The second file contains registered users' information like user-id, location.

Ratings – Ratings contain information like which user has given how much rating to which book.

5.2 Preprocessing Data:

In the book file, we have some extra columns which are not required for our task like image URLs. And we will rename the columns of each file as the name of the column contains space, and uppercase letters so we will correct as to make it easy to use. The dataset is reliable and can consider as a large dataset. We have 271360 books data and total registered users on the website are approximately 278000 and they have given near about 11 lakhs rating. Hence, we can say that the dataset we have is nice and reliable.

5.3 Approaches to a problem statement:

We do not want to find a similarity between users or books. We want to do that if there is user A who has read and liked X and Y books, and user B has also liked these two books and now user A has read

and liked some Z book which is not read by B so we have to recommend Z book to user B. This is what collaborative filtering is.

```
ratings_with_name = ratings.merge(books,on='ISBN')
```

```
ratings_with_name
```

	User-ID	ISBN	Book-Rating	Book-Title	Book-Author	Year-Of-Publication	Publisher
0	276725	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002	Ballantine Books
1	276726	0155061224	5	Rites of Passage	Judith Rae	2001	Heinle
2	276727	0446520802	0	The Notebook	Nicholas Sparks	1996	Warner Books

Figure 1. Finding Books by 'ISBN'

So, this is achieved using Matrix Factorization, we will create one matrix where columns will be users and indexes will be books and values will be rating.

```
pt = a.pivot_table(index='Book-Title',columns='User-ID',values='Book-Rating')
pt
```

User-ID	254	2276	2766	2977	3363	4017	4385	6251	6323	6543	...	27
Book-Title												
1984	9.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	
1st to Die: A Novel	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	9.0	...	
2nd Chance	NaN	10.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	...	
4 Blondes	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN	...	
A Bend in the Road	0.0	NaN	7.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	

Figure 2. Pivot Table

We have to decrease the number of users and books because we cannot consider a user who has only registered on the website or has only read one or two books. On such user, we can't rely to recommend books to others because we have to extract knowledge from data. So, what we will limit this number and we will take a user who has rated at least 200 books and also limit books and take only those books which have received at least 50 ratings from user.

```
x=ratings_with_name.groupby('User-ID').count()['Book-Rating']>200
y = x[x].index
y
```

```
Index([   254,   2276,   2766,   2977,   3363,   4017,   4385,   6251,   6323,
       6543,
       ...
       271705, 273979, 274004, 274061, 274301, 274308, 275970, 277427, 277639,
       278418],
      dtype='int64', name='User-ID', length=811)
```

```
filtered_rating = ratings_with_name[ratings_with_name['User-ID'].isin(y)]
filtered_rating
```

	User-ID	ISBN	Book-Rating	Book-Title	Book-Author	Year-Of-Publication	Publisher
1150	277427	002542730X	10	Politically Correct Bedtime Stories:	James Finn Garner	1994	John Wiley & Sons Inc

Figure 3. Group by User-ID

```
z=filtered_rating.groupby('Book-Title').count()['Book-Rating']>50
w=z[z].index
w
```

```
Index(['1984', '1st to Die: A Novel', '2nd Chance', '4 Blondes',
       'A Bend in the Road', 'A Case of Need',
       'A Child Called \It\": One Child's Courage to Survive"',
       'A Civil Action', 'A Day Late and a Dollar Short', 'A Fine Balance',
       ...
       'Winter Solstice', 'Wish You Well', 'Without Remorse',
       'Wizard and Glass (The Dark Tower, Book 4)', 'Wuthering Heights',
       'Year of Wonders', 'You Belong To Me',
       'Zen and the Art of Motorcycle Maintenance: An Inquiry into Values',
       'Zoya', '\O\" Is for Outlaw"'],
      dtype='object', name='Book-Title', length=679)
```

Figure 4. Group by Book-Title

Now we will train the nearest neighbours algorithm: -

```
from sklearn.metrics.pairwise import cosine_similarity

similarity_scores = cosine_similarity(pt)

similarity_scores.shape

(679, 679)

def recommend(book_name):
    # index fetch
    index = np.where(pt.index==book_name)[0][0]
    similar_items = sorted(list(enumerate(similarity_scores[index])),key=lambda x:x[1],reverse=True)[1:5]

    data = []
    for i in similar_items:
        item = []
        temp_df = books[books['Book-Title'] == pt.index[i[0]]]
        item.extend(list(temp_df.drop_duplicates('Book-Title')['Book-Title'].values))
        item.extend(list(temp_df.drop_duplicates('Book-Title')['Book-Author'].values))
        item.extend(list(temp_df.drop_duplicates('Book-Title')['Image-URL-M'].values))
```

Figure 5. Train the Model

Let's make a prediction and see whether it is suggesting books or not. We will find the nearest neighbours to the input book id and after that, we will print the top 4 books which are closer to those books.

```
recommend('1984')
```

```
[['Animal Farm',
  'George Orwell',
  'http://images.amazon.com/images/P/0451526341.01.MZZZZZZZ.jpg'],
 ["The Handmaid's Tale",
  'Margaret Atwood',
  'http://images.amazon.com/images/P/0449212602.01.MZZZZZZZ.jpg'],
 ['Brave New World',
  'Aldous Huxley',
  'http://images.amazon.com/images/P/0060809833.01.MZZZZZZZ.jpg'],
 ['The Vampire Lestat (Vampire Chronicles, Book II)',
  'ANNE RICE',
  'http://images.amazon.com/images/P/0345313860.01.MZZZZZZZ.jpg']]
```

Figure 6. Results of Nearest Neighbour Algorithm

6. RESULT:

Through a systematic exploration of data collection methodologies, algorithms design principles, and ethical guidelines, this study seeks to illuminate the journey of developing a book recommendation system. It aims to uncover the challenges, opportunities, and implications of harnessing data and AI to enhance the literary journey for readers.

Figure 7. Displays the Books

Figure 8. Recommended Books

This summary provides a glimpse into the evolving landscape of book recommendation systems, showcasing how research and innovation continue to shape the future of personalized content delivery in the digital age.

7. CONCLUSION:

By fostering a deeper understanding of how recommendation system operates and evolve, this research contributes to the boarder discourse on personalized content delivery in the digital age, striving to meet the evolving needs and expectations of modern readers.

8. REFERENCES:

1. Campos, P. G., Dias, G., & Jorge, A. M. (2020). User profile modelling in book recommendation systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 2137-2140). ACM.
2. Kumar, A., & Goyal, P. (2021). A comparative study of collaborative filtering techniques for book recommendation. *Journal of Information Science*, 47(2), 185-201. https://doi.org/10.1177/0165551520933296
3. Li, S., Lu, Y., & Li, H. (2022). An improved book recommendation algorithm based on user interest and content characteristics. *Journal of Ambient Intelligence and Humanized Computing*, 13(1), 265-276. https://doi.org/10.1007/s12652-020-02128-4
4. Wang, L., Liu, C., & Li, H. (2023). Personalized book recommendation based on social network and collaborative filtering. *Journal of Computational Science*, 52, 101370. https://doi.org/10.1016/j.jocs.2020.101370
5. Zhang, L., Chen, G., & Li, T. (2024). A deep learning-based book recommendation system using attention mechanism. *Information Processing & Management*, 61(2), 102765. https://doi.org/10.1016/j.ipm.2023.102765
6. Kumar, N., & Meena, M. (2020). An efficient book recommendation system using collaborative filtering and association rule mining. *International Journal of Intelligent Engineering and Systems*, 13(4), 64-72. https://doi.org/10.22266/ijies2020.0428.07
7. Jannach, D., & Ludewig, M. (2021). The long tail of book recommendations. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 11(2), Article 12. https://doi.org/10.1145/3438584