CSE508 - Information Retrieval Project
Book Recommendation System

Shivanshu Kumar
2019476

Karan Prasad Gupta
2020439

Rishav
2020569

Sarthak Dixit
2020574

D Likhith
2019038

Harsh Kashyap
2020434

1. Problem Statement:

We aim to develop an advanced book recommendation system to address the challenge users face in finding personalized book suggestions amidst a vast array of options. Existing systems often fail to provide accurate recommendations, resulting in user dissatisfaction and reduced engagement. We propose a novel approach leveraging sophisticated algorithms and user-centric design to deliver tailored book suggestions that enhance user satisfaction and engagement.

Motivation:

Online book availability has surged in the digital age, offering users a plethora of options. However, sifting through countless titles to find personalized recommendations is challenging.Traditional recommendation systems often use simplistic algorithms, providing inadequate suggestions.These systems fail to capture the nuances of individual reading preferences, leading to frustration and disengagement.

2. Literature Review

1. https://ieeexplore.ieee.org/document/9579647

The authors discover methods including machine learning techniques like the K-nearest neighbors, Pearson's R Correlation Coefficient, Cosine Similarity through collaborative filtering, to be efficient in deciding the best books for the user based on the query that is provided as the input into their system.

2. https://cs.carleton.edu/cs_comps/1617/book_rec/final-results/paper.pdf

The project on book recommendation system works on the data collected from across the web (Amazon Books, GoogleReads, etc.) and makes use of the Machine Learning classifier models such as Naive Bayes Classifier and the Maximum Entropy Classifier for their Content based approach and the K-nearest neighbor and UV decomposition for their Collaborative filtering based approach for determination of appropriate books according to the user query.

3. Book Recommendation System using Association Rule Mining & Collaborative Filtering

The research paper discusses several Collaborative filtering algorithms like the Jaccard Distance and Pearson' Coefficient along with a novel technique known as Association Mining.

Among all the above mentioned literature, we came to the conclusion that our information retrieval system would make use of parts of each of them, like Cosine Similarity, few of the Machine Learning approaches like Clustering (or K Nearest neighbors) and Pearson's coefficient, and all of the suggested evaluation methods would be considered in deciding which model works the best, which finally would be

allowed in the model combination process where we integrate various models and their features.

3. Methodology

The model that we built is an amalgam of various types of Recommender Systems, it's a hybrid recommender system that mainly includes the functionalities of Content-based and Collaborative-filtering based recommender systems. The overall idea of the approach is to find out the best of the features from the processed datasets (Books, Ratings and Users), performing matrix factorization and using techniques including Cosine Similarity.

Baseline:

1. Collaborative Filtering:

- Identified users who rated more than 200 books.
- Filtered and selected books with more than 50 ratings.
- Created a user-item matrix for collaborative filtering.

Similarity Techniques:

Cosine Similarity:

We computed the cosine similarity between every pair of books and then recommended the top 5 similar books based on cosine similarity.

Pearson's Coefficient:

Calculated Pearson correlation coefficients between books and based on that recommended 5 books.

2. Content-Based Recommendation(using TF-IDF model):

We preprocessed the text data in the Books data frame by lowercasing, punctuation removal, stop-word removal, and stemming. We then concatenated the 'Book-Title,' 'Book-Author,' and 'Publisher' columns into a single text for each book in the Books dataframe. We then utilized the TF-IDF model to learn vector representations of words in the combined text data and then using Cosine Similarity on the reduced sized vectors (through SVD) for finding out the most similar books.

3.Hybrid recommendation method:

The hybrid recommendation method in our system integrates Singular Value Decomposition (SVD) solely for dimensionality reduction, enhancing computational efficiency. Collaborative filtering is employed to analyze user-item interactions and recommend similar books, while content-based filtering utilizes natural language processing to identify textual similarities. Through weighted averaging, the hybrid method seamlessly combines both approaches, ensuring recommendations are tailored and diverse. In the recommendation process, the system first employs SVD to reduce dimensions, then analyzes textual content to identify similar books. Finally, the hybrid recommendation component combines the scores from both methods to deliver personalized suggestions. Evaluation of the hybrid method using book datasets and user interactions demonstrated its effectiveness, with metrics such as precision, recall, and user satisfaction

indicating superior performance compared to traditional methods.

Data Overview:

Users Table:

The Users table has info about users like User-ID, name etc.

```
users.head()
```

	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 Table:

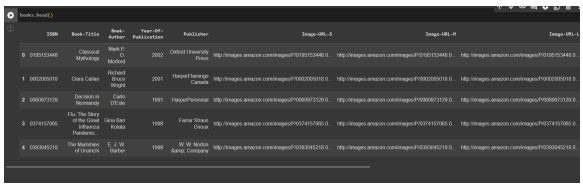
The Ratings table has user ratings for various books, identified by their ISBN.

```
[ ] ratings.head()
```

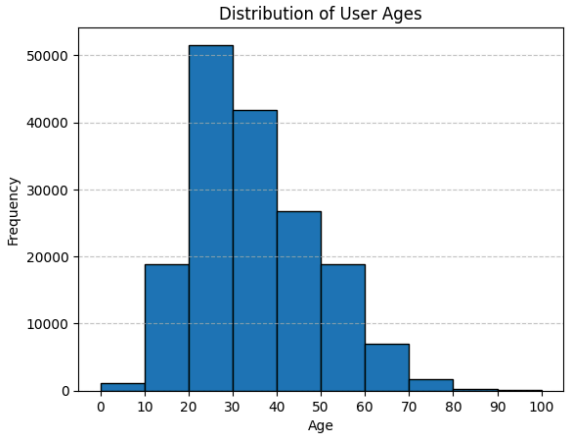
	User-ID	ISBN	Book-Rating
0	276725	034545104X	0
1	276726	0155061224	5
2	276727	0446520802	0
3	276729	052165615X	3
4	276729	0521795028	6

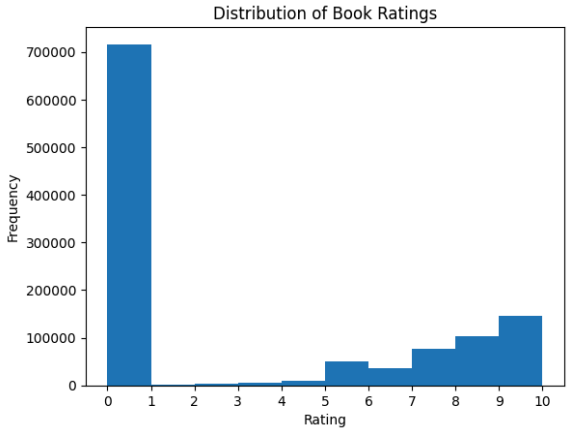
Books Table:

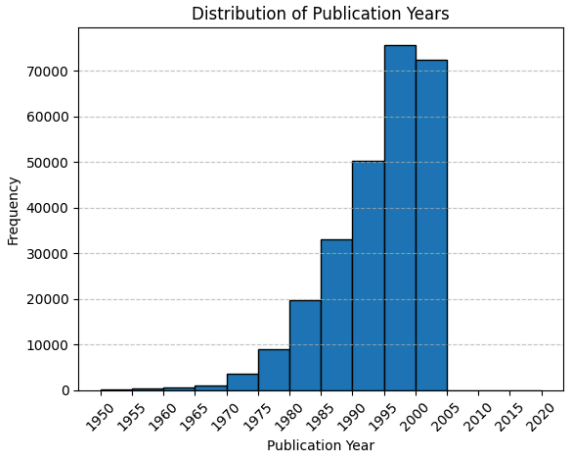
The Books table has details about books, such as Book-Title, Book-Author, and Image-URL-M.



Exploratory Data Analysis:







Data Cleaning:

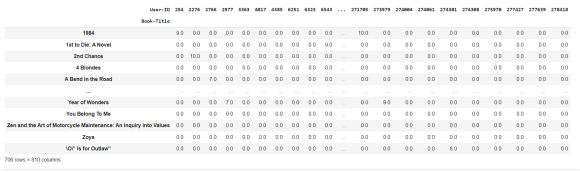
We checked for missing values in the Books, Users, and Ratings tables. No missing values were found.

Data Processing:

- We then merged the Ratings table with the Books table based on the common column 'ISBN.'
- We calculated the number of ratings and the average rating for each book.
- Merged the number of ratings and average ratings dataframes.
- Filtered and selected the top 50 books with a minimum of 250 ratings.

Data Visualization:

Matrix Factorization



1. Collaborative filtering-based methods:-

i. Cosine Similarity:

Recommendation:

	Book-Title	Book-Author	Image-URL-M
0	1984	George Orwell	http://images.amazon.com/images/P/0451524934.0...
1	Angus, Thongs and Full-Frontal Snogging: Confe...	Louise Rennison	http://images.amazon.com/images/P/0064472272.0...
2	Midnight	Dean R. Koontz	http://images.amazon.com/images/P/0425118703.0...
3	Second Nature	Alice Hoffman	http://images.amazon.com/images/P/0399139087.0...
4	Call of the Wild	Jack London	http://images.amazon.com/images/P/1559029838.0...

ii. Pearson's coefficient

Recommendation:

```
recommend_through_pc('Animal Farm')
```

```
Similar books:
1. Book name: 1984, Similarity: 0.24863430168716266
2. Book name: Angus, Thongs and Full-Frontal Snogging: Confessions of Georgia Nicolson, Similarity: 0.2226534505381867
3. Book name: Midnight, Similarity: 0.21992580013893603
4. Book name: Second Nature, Similarity: 0.20647606147712394
5. Book name: Call of the Wild, Similarity: 0.19336425011186748
```

2. Content Based method:

Recommendation:

	Book-Title	Book-Author	Image-URL-S
77519	animal babies	dandi daley mackall	http://images.amazon.com/images/P/1569873364.0...
160434	small animal nutrition	sandie agar	http://images.amazon.com/images/P/075064575X.0...
134756	barneys farm animals	kimberly kearns	http://images.amazon.com/images/P/1570640025.0...
59731	beautiful animal dolls handcrafts treasure	miriam gourley	http://images.amazon.com/images/P/0806960884.0...
208499	animal crackers	hannah tinti	http://images.amazon.com/images/P/0385337434.0...

3. Hybrid based Recommendation:

	Book-Title	Book-Author	Image-URL-S
77519	animal babies	dandi daley mackall	http://images.amazon.com/images/P/1569873364.0...
160434	small animal nutrition	sandie agar	http://images.amazon.com/images/P/075064575X.0...
134756	barneys farm animals	kimberly kearns	http://images.amazon.com/images/P/1570640025.0...
59731	beautiful animal dolls handcrafts treasure	miriam gourley	http://images.amazon.com/images/P/0806960884.0...
208499	animal crackers	hannah tinti	http://images.amazon.com/images/P/0385337434.0...

4. Frontend Implementation

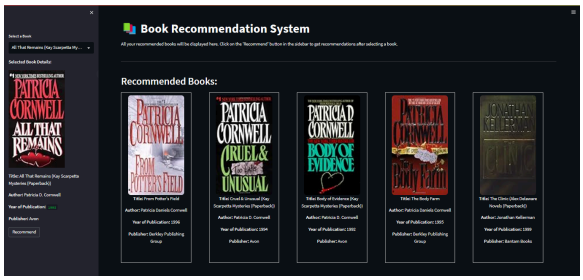
For the frontend part of the book recommendation system, as per our current progress we have designed a single-page interactive and user-friendly web application using Streamlit. It primarily consists of UI elements designed to let users select a book, and view recommended books based on similarity scores computed using collaborative filtering.

Tools Used:

- Streamlit: Streamlit is a Python framework used for easily building interactive web applications and data apps by providing a wide range of UI components and widgets.
- HTML/CSS: HTML and CSS were used with streamlit markdown to

customize the appearance and layout of the web application.

Implementation:



The web application is currently a single-page application with 2 sections: a sidebar and the main page.
The sidebar has a dropdown that allows the user to type/select the book from all the books available, upon selection the details of the selected book are displayed in the sidebar along with the recommend button to get a recommendation based on it.
On Clicking the recommend button, the Python function to fetch recommendations based on the similarity scores is run, and the top 5 books are displayed in the main section containing Book Front Cover, Title, Author name, Year of Publication, and Publisher.

4. Proposed Solution (Novelty):
We proposed and introduced two new features into our recommendation system:

1. Hybrid Model - The model implemented combined the strengths of both the content based and collaborative filtering based system.
2. Personalized User Interface: The user gets to tell the system if the recommendations are meaningful or not, and the recommendations are altered according to the user preference as the values in the similarity matrix are changed, and the recommendations are changed.

Code:
https://colab.research.google.com/drive/1lNqLBAOYQacRGyyKQgljlhgurjAnV38J#scrollTo=na1mT_XXzDtH

5. Evaluation:
For evaluation, the books data and the pivot table were split into Training and Test sets (80:20) respectively for Content based and Collaborative Filtering based techniques, the recommendations from the training set were compared with the test set to assess the metrics.

For the Hybrid model, the metric scores for the two mentioned models were weight averaged.

The three models (Content based, Collaborative Filtering based and Hybrid filtering) are evaluated based on the three metrics:

1. Average Precision
2. Average Recall
3. Average F1-Score

And the best among them is decided on the F1-Score, since it is the combination of both the precision and recall.

Results:

1. For Content Based Recommendations:

```
Average Precision: 0.765137614678893
Average Recall: 0.67889908256881
Average F1-score: 0.719443246331
```

2. For Collaborative Filtering Based Recommendations:

```
Average Precision: 0.7648146564912
Average Recall: 0.6363567848763
Average F1-score: 0.694697281806
```

3. Hybrid Recommendations:

```
Average Precision (Hybrid): 0.7649438397662771
Average Recall (Hybrid): 0.653373703953304
Average F1-score (Hybrid): 0.704595667616
```