Book Recommender System

Jean Chao

Problem Statement

"Develop a **personalized book recommendation system** that accurately suggests books based on individual **user preferences, rating, and reading history**, thereby **improving the overall reading experience** and helping users discover new authors and genres efficiently."

Assumptions/Hypothesis

1. User Preferences

The recommendation model assumes that **users' preferences remain relatively consistent** over time.

2. Item Similarity

When doing the collaborative filtering models, we assume that books that are frequently rated or liked by **similar users tend to have similar characteristics**.

3. Cold Start Problem

The model assume that there is a **sufficient amount of data available** for users and books to provide accurate and meaningful recommendations.

4. Data Quality

It is assumed that the dataset used for training the model is **good quality with accurate and reliable information** of books, users, and ratings.

Data Overview

The three datasets are from Kaggle include books, ratings, and users.

For books dataset, we **drop three URL columns** which are unnecessary columns for the model.
For rating dataset, we **extract the country** from the location column by using split ' , '
For users dataset, we **drop the Age column** which are unnecessary columns for the model.

We also **check for missing value** and **duplicated value**.

Remaining features for each dataset:

Books

- ISBN
- Book-Title
- Book-Author
- Year-Of-Publication
- Publisher

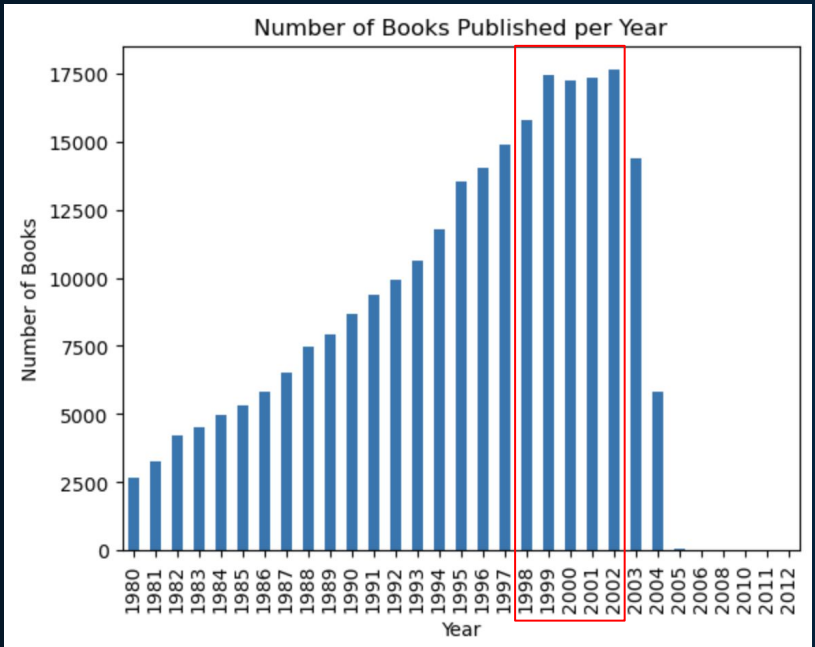
Ratings

- User-id
- ISBN
- Book-rating

Users

- User-id
- Location

Number of Books Published per Year



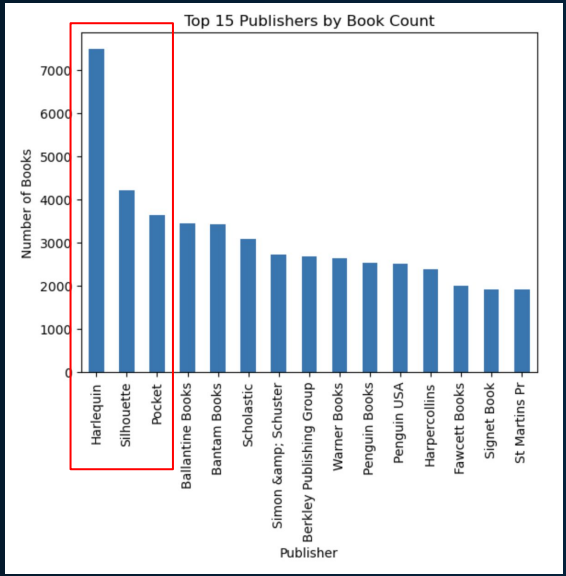
When **cleaning the book dataset**, the "year of publication" includes some errors:

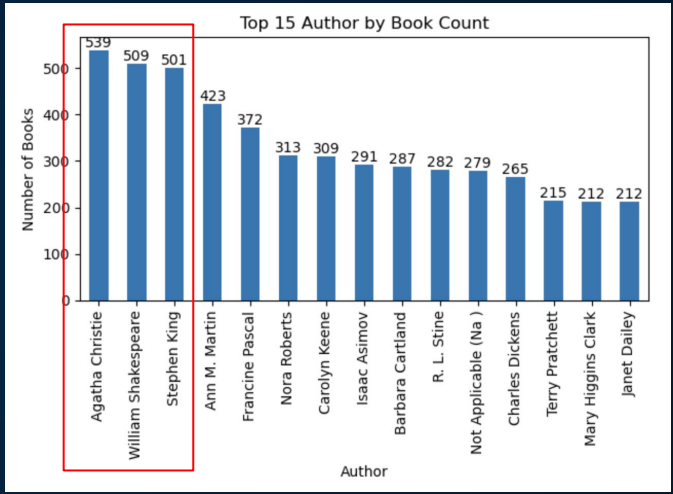
- future years such as 2038, 2050, etc
- 0 year
- after year 2012, there is missing values between 2012 and 2020

Therefore, we only include year from 1980 and 2012.

The years 1998 to 2002 are the years in which most books were published.

Top 15 Publishers & Top 15 Authors





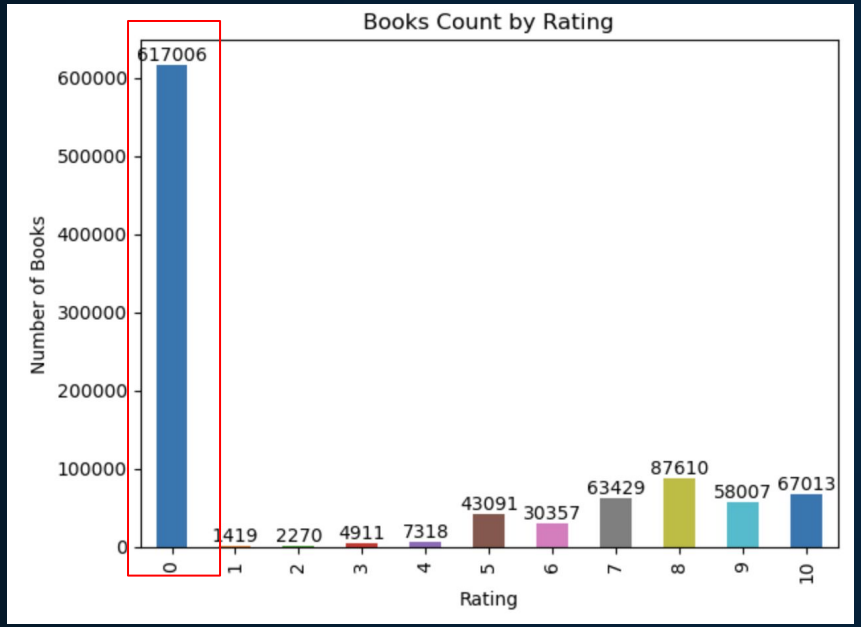
Top 3 Publishers who have published books

1. Harlequin (7533 books)
2. Silhouette (4220 books)
3. Pocket (3905 books)

Top 3 Authors authored books by count

1. Agatha Christie (539 books)
2. William Shakespeare (509 books)
3. Stephen King (501 books)

Books Count by Rating



By plotting the bar chart, we found out:

- 0 rating has the highest books count (617006 books)

- We will **remove books with zero ratings** from the dataset as these are books that are not rated by users

Top 10 Most Rated books by users

	Book-Title	Book-Rating
4	Harry Potter and the Sorcerer's Stone (Harry P...	8.936508
15	The Secret Life of Bees	8.477833
8	The Da Vinci Code	8.439271
10	The Lovely Bones: A Novel	8.185290
14	The Red Tent (Bestselling Backlist)	8.182768
18	Where the Heart Is (Oprah's Book Club (Paperba...	8.142373
6	Life of Pi	8.080357
1	Angels & Demons	8.016129
12	The Notebook	7.897611
3	Divine Secrets of the Ya-Ya Sisterhood: A Novel	7.876161

By grouping by the book title, and calculating the average book rating, we found out:

- Harry Potter and the Sorcerer's Stone has been rated the highest by users with an 8.94 score
- Followed by the second book The Secret life of Bees with 8.48 and The Da Vinci Code with 8.44.

Feature Engineering

Before running the model, we did further clean our dataset.

1. From the EDA, we notice that there are zero rating books, therefore we will remove those since they are not being rated by users.

```
complete_df = complete_df[complete_df["Book-Rating"]>0]
complete_df["Book-Rating"].describe()
```

2. To ensure our dataset is credible, a book that receives high ratings but has only been rated by one or two users lacks credibility. Therefore, we will only include books with more than 20 ratings.

```
credible_book = complete_df['Book-Title'].value_counts()
credible_book = credible_book[credible_book > 20].index

complete_df = complete_df[complete_df['Book-Title'].isin(credible_book)]
```

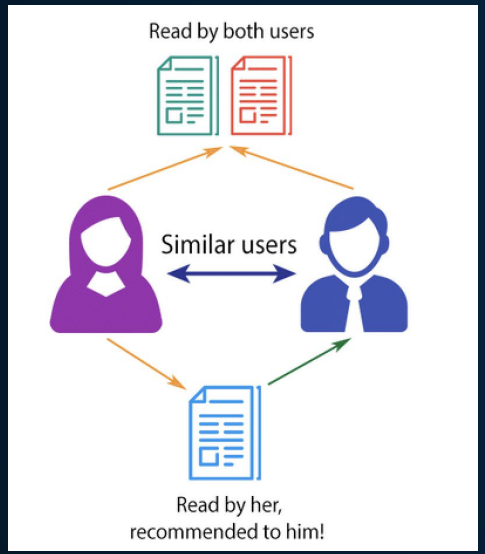
3. On the other hand, a user who has only rated one or two books is not considered a credible user. Therefore, we will only include those users who have given ratings to more than 50 books.

```
credible_user = complete_df['User-ID'].value_counts()
credible_user = credible_user[credible_user > 50].index

complete_df = complete_df[complete_df['User-ID'].isin(credible_user)]
```

Building Recommendation Model

Collaborative Filtering



- Based on the idea that people with similar preferences in the past will have similar preferences in the future.

- Collaborative filtering can be implemented using two main approaches:

1. **User-based filtering**: compares one user's preferences to those of other users in order to identify similar users and make recommendations based on their shared interests.

2. **Item-based filtering**: recommending items that are similar to the ones a user has already shown interest in.

Result (Item-based Collaborative Filtering)

Pivot Table

User-ID	254	638	1424	1733	1903	2033	2110	2276	2766	2977	...	273113	27
Book-Title													
1984	9.0	0.0	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	0.0	...	0.0	
2nd Chance	0.0	9.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	0.0	...	0.0	
A Bend in the Road	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.0	0.0	...	5.0	
A Case of Need	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	

- We first pivot the dataset to by setting our `index='Book-Title',columns='User-ID',values='Book-Rating'`
- Use **cosine similarity** to calculate similarity between users.
- We then recommend 5 books to users based on the recommender we built.

1. Recommend 5 books to users who have read "**The Secret Life of Bees**"

```
[['Good in Bed', 'Jennifer Weiner'],
 ['Wicked: The Life and Times of the Wicked Witch of the West',
  'Gregory Maguire'],
 ['The Rapture of Canaan', 'Sheri Reynolds'],
 ['Girl in Hyacinth Blue', 'Susan Vreeland'],
 ['The Da Vinci Code', 'Dan Brown']]
```

2. Recommend 5 books to users who have read "**The Notebook**"

```
[['The Rescue', 'Nicholas Sparks'],
 ['A Walk to Remember', 'Nicholas Sparks'],
 ['The Five People You Meet in Heaven', 'Mitch Albom'],
 ["Suzanne's Diary for Nicholas", 'James Patterson'],
 ['Message in a Bottle', 'Nicholas Sparks']]
```

Result (User-based Collaborative Filtering)

We set user id = 136382 as our target user.

Here is the top 10 book recommendation for the user:

	ISBN	weighted_rating	Book-Title
0	0515120898	7.503732	The Pull of the Moon
25	0525945210	7.503732	A Man Named Dave: A Story of Triumph and Forgi...
45	0345423097	7.503732	Joy School (Ballantine Reader's Circle)
67	0866852611	7.503732	Love Story (Arabic)
68	051511992X	7.503732	That Camden Summer
103	0375412824	7.503732	The Dive From Clausen's Pier (Alex Awards)
158	0515102636	7.503732	Morning Glory
184	0811801802	7.503732	Sabine's Notebook: In Which the Extraordinary ...
217	0805061762	7.503732	A Gentle Madness : Bibliophiles, Bibliomanes, ...
222	0684842327	7.503732	NEEDLES : A MEMOIR OF GROWING UP WITH DIABETES

- Before creating recommendation system, we first pivot the dataset to by setting our

```
index='User-ID',columns='Book-Title',values='Book-Rating'
```

- By calculating the **weighted rating**, it predict a user's rating for an item by considering the ratings of similar users in the neighborhood.

Future Work

1. Instead of just using cosine similarity, we can **explore other similarity metrics** such as pearson correlation or adjusted cosine similarity to see the difference.

2. Explore **other recommendation algorithms** such as hybrid approaches or matrix factorization methods to improve the recommendation accuracy.

3. Get **more data that related to genre** of the books, we can check the recommendation result based on that and help users discover new genres efficiently.

4. To better test the accuracy of our recommendation model, we can also do the **A/B testing** which requires larger user base but allows you to compare the performance of the new system and the existing one.

Thank you!

References

Data link:
https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset?datasetId=1004280

https://www.researchgate.net/figure/Content-based-filtering-and-Collaborative-filtering-recommendation_fig3_331063850