THE BOOK CLUB PROJECT

Author: Tais Pancier

INTRODUCTION

- A book club is a group of people who meet to discuss a book or books that they have read and express their opinions, likes, dislikes, etc. In today's hectic life, people don't have much time to meet in person to discuss books.

OBJECTIVE

- The purpose of this project is to create a virtual book club with a system that will recommend books to read based on ratings and also other readers to discuss books with using Unsupervised Learning.
- This model can be implemented by a book store wanting to increase their sales or by a library wanting to increase traffic and community engagement.

DATA SOURCE

- The dataset obtained in Kaggle consists of 3 main files:
  - Book metadata: ten thousand XML files
  - User ratings for each book: csv file
  - Book tags: csv file
- The book metadata (author, language, year, etc) dataset contains ten thousand books.
- The ratings dataset contains reviews for all the book by user id. All users have made at least two ratings.
- Ratings go from one to five. Both book IDs and user IDs are contiguous. For books, they are 1-10000, for users, 1-53424.
- There are also books marked to read by the users (wish list) and tags.

DATA CLEANING

- The XML files were read using Python object xml.etree.ElementTree and saved in 2 datasets: books and authors.
- The raw books dataset had 10000 rows and 34 columns. The dataset after cleaning has 10000 rows, 10 columns.
- The raw ratings dataset had 5976479 rows and 3 columns. The dataset after cleaning has 705914 rows and 3 columns. Some rows were removed because they contained books that were not in the 10K book dataset.
- Steps used to clean the dataset: dropped columns that were not relevant for the analysis and machine learning model, change type in numerical columns, filled missing values.

EXPLORATORY DATA ANALYSIS

1. Summary statistics for Books:

	average_rating	ratings_count	text_reviews_count
count	10000.000000	1.000000e+04	10000.000000
mean	4.002224	5.404310e+04	2385.520300
std	0.254406	1.574966e+05	5069.490366
min	2.470000	2.718000e+03	3.000000
25%	3.850000	1.357700e+04	542.000000
50%	4.020000	2.116850e+04	1138.500000
75%	4.180000	4.108200e+04	2267.000000
max	4.820000	4.784860e+06	141502.000000

	book_id	title	country_code	language_code
count	10000	10000	10000	10000
unique	10000	9964	1	25
top	33288638	Selected Poems	GB	eng
freq	1	4	10000	7432

- ratings_count mean is much higher than text_reviews_count, indicating that people tend to rate the books more often than providing a text review
- average rating is high in the 10K books set
- only country in the dataset is GB
- “Selected Poems” title has the highest mode
- 25 unique language codes, with “eng" being the most common

EXPLORATORY DATA ANALYSIS

2. Summary statistics for Ratings:

	rating
count	705914.000000
mean	3.965466
std	1.007613
min	1.000000
25%	3.000000
50%	4.000000
75%	5.000000
max	5.000000

	user_id	book_id
count	705914	705914
unique	53400	812
top	46521	1
freq	35	22806

- Ratings vary from 1 to 5
- 705914 ratings for 812 books and 53400 users
- User id 46521 has the highest number of ratings: 35
- Book id 1 (Harry Potter - of course!) has the highest number of ratings: 22806

EXPLORATORY DATA ANALYSIS

3. Summary statistics for Wish List (books to read):

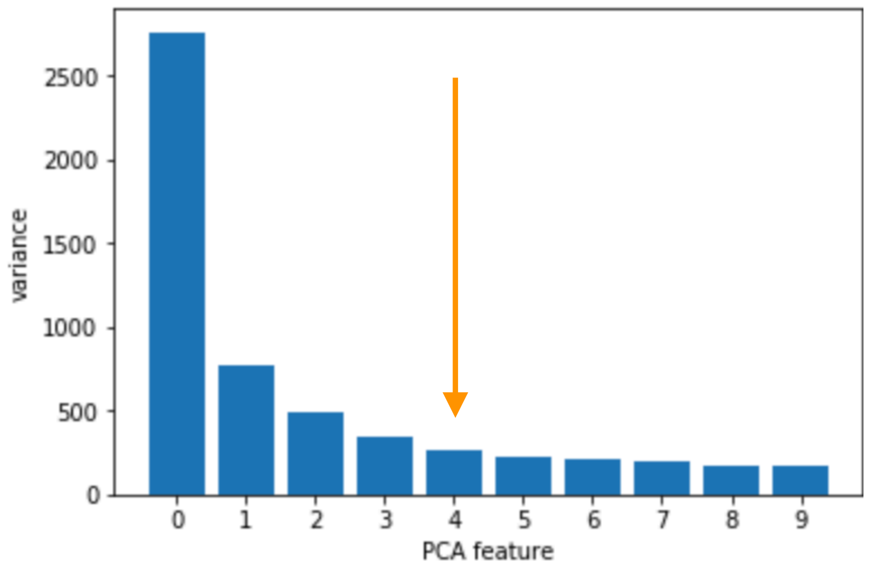
	average_rating	ratings_count	text_reviews_count
count	87493.000000	8.749300e+04	87493.000000
mean	4.028561	2.382774e+05	4522.605980
std	0.306165	5.170892e+05	7601.295942
min	2.800000	5.327000e+03	28.000000
25%	3.820000	1.930100e+04	623.000000
50%	3.970000	4.462600e+04	1603.000000
75%	4.220000	1.658920e+05	4134.000000
max	4.770000	4.607944e+06	58776.000000

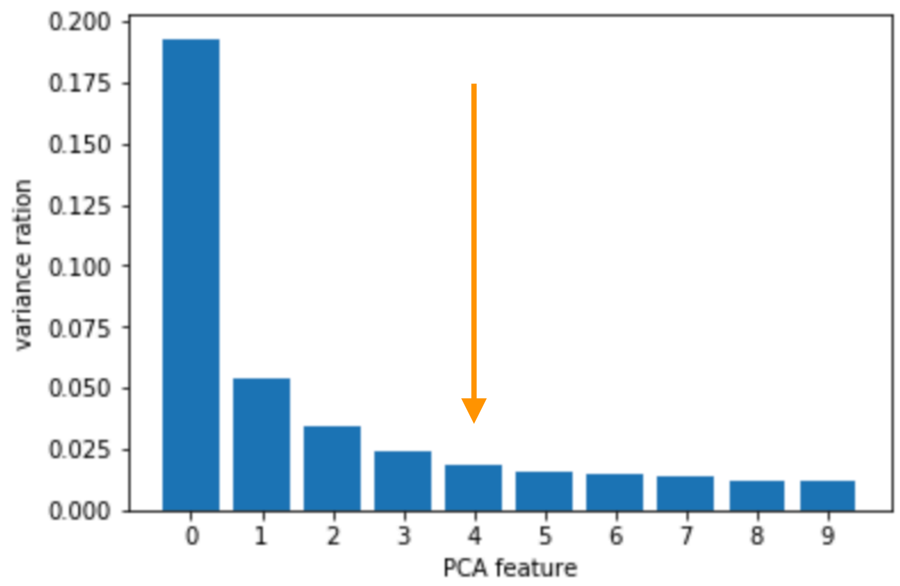
	user_id	book_id	authors
count	87493	87493	87493
unique	32806	811	403
top	12483	13	1077326
freq	15	1812	7137

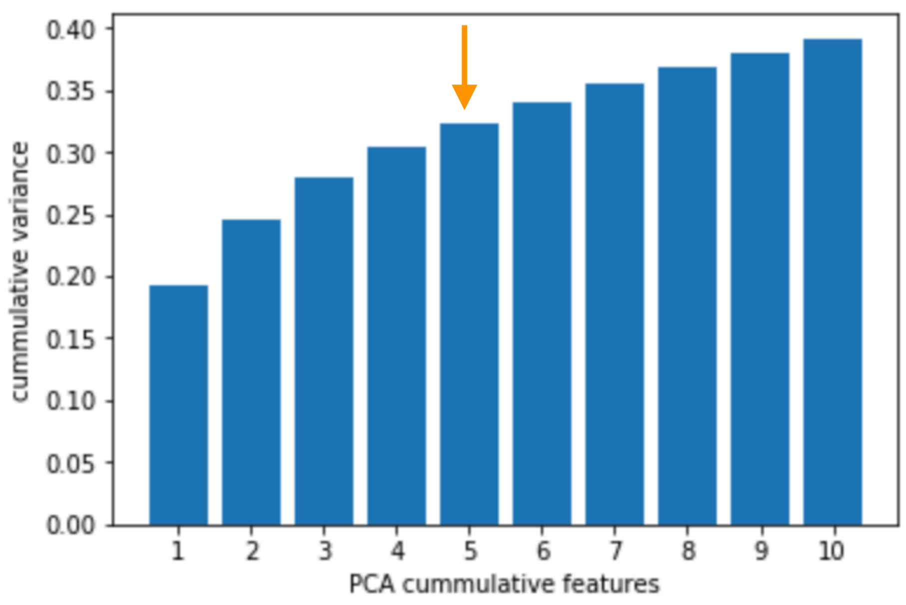
- user id 12483 has the highest number of books in the wish list: 15
- book id 13 (The Ultimate Hitchhiker's Guide to the Galaxy) shows up 1812 in books to read lists
- author 1077326 (J.K. Rowling) has the highest frequency in wish lists: 7137

UNSUPERVISED LEARNING – PCA

Intrinsic Dimension with PCA:







➤ Based on plots above, intrinsic dimension of 5 features was considered.

UNSUPERVISED LEARNING – RECOMMENDER SYSTEM

Description:

- NCM (Non-negative Matrix Factorization) with n_components = 5 and the cosine similarity was used to build a book recommender system based on user ratings.
- Besides recommending books that users could read, this model also recommends other readers with similar book rating profile so people can connect and exchange information about books.
- Cosine similarity was used to evaluate the distance between books in the wish list and the full ratings database and suggest the most similar books. For each book in the wish list, 2 similar books were suggested.
- Cosine similarity was used to evaluate the distance between users based on the ratings they provided and suggest the most similar ones.

UNSUPERVISED LEARNING – RECOMMENDER SYSTEM

1. Recommender system: Books

The following steps were executed as part of this model:

➤ Generate matrix with book ids as rows and user ids as columns (first 5 records shown below)

book_id	1	2	3	4	5	6	7	8	9	...	53415	53416	53417	53418	53419	53420	53421	53422	53423	53424
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	...	0.0	0.0	4.0	5.0	4.0	4.0	4.0	4.0	4.0	4.0
2	0.0	5.0	0.0	5.0	0.0	0.0	0.0	0.0	4.0	...	0.0	0.0	0.0	0.0	5.0	5.0	5.0	5.0	5.0	5.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	...	0.0	0.0	0.0	0.0	3.0	3.0	0.0	0.0	0.0	4.0
5	0.0	5.0	0.0	4.0	0.0	0.0	3.0	3.0	5.0	...	0.0	0.0	0.0	0.0	3.0	2.0	4.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0

➤ Generate dataframe with 5 NMF (normalized) features (first 5 records below)

book_id	0	1	2	3	4
1	0.000000	0.000000	0.993209	0.000000	0.116345
2	0.493134	0.224524	0.840481	0.000000	0.000000
3	0.018363	0.000000	0.998842	0.044472	0.000000
5	0.000000	0.977175	0.000000	0.108214	0.182810
6	0.020572	0.000000	0.035259	0.000000	0.999166

UNSUPERVISED LEARNING – RECOMMENDER SYSTEM

➤ Look in the normalized DataFrame for books in the wish list (example below):

- User id: 9

- 3 books in wish list of user 9: 8 (Harry Potter Boxed Set, Books 1-5) , 3476 (Icy Sparks), 112 (Children of Dune (Dune Chronicles #3))

book_id	0	1	2	3	4
8	0.000000	0.988645	0.0	0.147188	0.030282
3476	0.255862	0.953529	0.0	0.109457	0.115479
112	0.000000	0.000000	0.0	0.019758	0.999805

➤ Drop from normalized dataset books already rated by user 9

➤ Compute cosine similarity between books in the wish list and the normalized dataset:

UNSUPERVISED LEARNING – RECOMMENDER SYSTEM

➤ Show 2 recommendations for each book in the wish list, with corresponding book name:

- wish_id is a book id in the wish list
- wish_title is a book title in the wish list
- similar_id is a recommended book id
- similar_title is a recommended book title

	wish_id	wish_title	similar_id	similar_title	similarity
0	8	Harry Potter Boxed Set, Books 1-5 (Harry Potte...	1420	Hamlet	0.999191
1	8	Harry Potter Boxed Set, Books 1-5 (Harry Potte...	5369	The Amber Room	0.999097
2	3476	Icy Sparks	903	The Egypt Game (Game, #1)	0.998330
3	3476	Icy Sparks	9998	The Woman in the Dunes	0.990695
4	112	Children of Dune (Dune Chronicles #3)	3466	The Wedding (The Notebook, #2)	0.999986
5	112	Children of Dune (Dune Chronicles #3)	2530	Baltasar and Blimunda	0.999980

UNSUPERVISED LEARNING – RECOMMENDER SYSTEM

1. Recommender system: Users

The following steps were executed as part of this model:

➤ Generate matrix with user ids as rows and book ids as columns (first 5 records shown below)

user_id	1	2	3	5	6	8	10	11	13	...	9854	9864	9865	9912	9913	9914	9915	9943	9957	9998
1	0.0	0.0	0.0	0.0	0.0	0.0	4.0	5.0	4.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	5.0	0.0	5.0	0.0	4.0	5.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	5.0	0.0	4.0	0.0	4.0	5.0	4.0	4.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	4.0	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

➤ Generate dataframe with 5 NMF (normalized) features (first 5 records below)

	0	1	2	3	4
user_id					
1	0.000000	0.354922	0.059074	0.933028	0.0
2	0.781226	0.594015	0.021383	0.190721	0.0
3	0.000000	1.000000	0.000000	0.000000	0.0
4	0.603214	0.738703	0.107796	0.280767	0.0
5	0.000000	0.000000	0.000000	0.000000	1.0

UNSUPERVISED LEARNING – RECOMMENDER SYSTEM

➤ Look in the normalized DataFrame by user and compute cosine similarity:

- User id: 9
- User_ids below are the recommended readers with largest similarity:

```
user_id
9          1.000000
6445       0.999219
52779      0.998622
4200       0.997966
9992       0.997526
50147      0.997012
14205      0.996186
29404      0.995741
5335       0.995724
1979       0.995468
```

UNSUPERVISED LEARNING – RECOMMENDER SYSTEM

➤ Shows other users together with the highest and lowest book ratings (example below for similar user id = 6445):

- Highest ratings:

	user_id	book_id	rating	title
304673	6445	24	5	In a Sunburned Country
45446	6445	36	5	The Lord of the Rings: Weapons and Warfare
304685	6445	119	5	The Lord of the Rings: The Art of The Fellowsh...
304684	6445	13	5	The Ultimate Hitchhiker's Guide to the Galaxy
304682	6445	93	5	Heidi

- Lowest ratings:

	user_id	book_id	rating	title
69477	6445	1381	2	The Odyssey
36616	6445	26	3	The Lost Continent: Travels in Small-Town America
211795	6445	34	3	The Fellowship of the Ring (The Lord of the Ri...
304668	6445	30	3	J.R.R. Tolkien 4-Book Boxed Set: The Hobbit an...
77275	6445	291	4	The Broken Wings

➤ **Interpretation:** User 9 will get the message that based on their ratings, user 9 would enjoy chatting with user 6445. User 6445 highest and lowest rated books are shown to give a starting point for the conversation.

PROPOSED NEXT STEPS

- Improve the model so it addresses diversity, serendipity, novelty, robustness and scalability in the book recommendations
- Collect feedback from users on the recommendations to evaluate the mode
- Collect additional data related to the ratings, for example time stamp. With the time stamp it will be possible to implicitly determine if a user rated a book after it was recommended to them, which is an indication of a good recommendation performance and could be used as a prediction in the model evaluation.
- Explore ranking evaluation methods such as Spearman, Kendall or Utility