Book Recommendation System

Shrinidhi Choragi
Data Science Trainee, AlmaBetter
Bangalore.

Abstract

A recommender system can be called an intelligent piece of code that studies the data to recommend the products. It is an information filtering technology used in a wide range of platforms as per the interest of users and is implemented in applications like movies, music, venue, books, research articles, tourism, and social media in general.

The recommendation system is constructed using various approaches like popularity-based(trend) recommendations, collaborative filtering recommendations, content-based filtering approaches, or as a hybrid of any of the mechanisms.

Collaborative filtering helps in finding the adjacent neighbors of a customer, concerning the user's rating history and tries to generate the most optimally suitable recommendation for the user. In the case of the content-based filtering approach, a consumer profile is prepared by procuring the contents of items rated by the user and the system will propose recommendations that match the customer profile.

Therefore, this study attempts to build one such system to recommend books.

Keywords: *Recommendation, Collaborative Filtering, Content-Based Filtering, Popularity.*

Problem Statement

The main objective is to create a machine learning model to recommend relevant books to users based on different algorithms.

The following points are explored:

- What is a recommendation system and how does it work?
- What hypothesis can be made from the data?
- Explore/understand the different algorithms that recommend books

Introduction

When you read some news, watch a movie on Netflix, or simply buy something on Amazon you will get some messages like:

- You will also probably like this
- Frequently bought together
- Products related to this item
- Customers who bought this item also bought
- Because you have seen X you might also like Y
- Recommended for you

Recommendation systems use specialized algorithms and machine learning solutions. Driven by the automated configuration, coordination, and management of machine learning predictive analytics algorithms, the recommendation system can wisely select which filters to apply to a particular user's specific situation. It facilitates marketers to *maximize conversions and average order value.*

During the last few decades, with the rise of Youtube, Amazon, Netflix, and many other such web services, recommender systems have taken more and more place in our lives. From e-commerce (suggest to buyers articles that could interest them) to online advertisement (suggest to users the

right contents, matching their preferences), recommender systems are today unavoidable in our daily online journeys.

In a very general way, recommender systems are algorithms aimed at suggesting relevant items to users (items being movies to watch, text to read, products to buy, or anything else depending on industries).

Recommendation systems are critical in some industries as they can generate a tremendous amount of income when they are efficient or also be a way to stand out significantly from competitors. The main objective here is to create a book recommendation system for users.

Dataset

We are using the Book-Crossing dataset to train and test our recommendation system. Book-Crossings is a book rating dataset compiled by *Cai-Nicolas Ziegler*. It contains 1.1 million ratings of 271,360 books by 278858 users. The ratings are on a scale from 1 to 10. The Book-Crossing dataset comprises three different files.

- ***Users***

This *.csv* file contains the users. Note that user IDs (User-ID) have been anonymized and mapped to integers. Demographic data is provided (Location, Age) if available. Otherwise, these fields contain NULL values.

- ***Books***

Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (Book-Title, Publisher, Year-Of-Publication, Book-Author), obtained from Amazon Web Services. Note that in the case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three flavors (Image-URL-S, Image-URL-M, Image-URL-L), i.e., small, medium, and large. These URLs point to the Amazon website.

- ***Ratings***

Contains the book rating information. *Ratings (Book-Rating) are either explicit, expressed on a scale from* 1–10 (higher values denoting higher appreciation), or *implicit, expressed by* 0.

Methodology

Data Preprocessing and Exploratory Data Analysis

Data pre-processing gives the feel of the data. If the data is messy, it is improved by sorting and deleting extra rows and columns. This stage generally involves data cleaning, merging, sorting, looking for outliers, looking for missing values in the data, and imputing missing values.

The primary goal of EDA is to support the analysis of data before making any conclusions. It may aid in the detection of apparent errors, as well as a deeper understanding of data patterns, the detection of outliers or anomalous events, and the discovery of interesting relationships between variables.

1. Missing Value Analysis

Values that are reported as missing may be due to a variety of factors. This lack of answers would be considered missing values. These data values may be deleted or data imputation can be done to replace them.
There are missing values in some feature variables namely *Year-Of-Publication*, *Age*, *Book-Author*, *and Publisher* that are to be imputed.

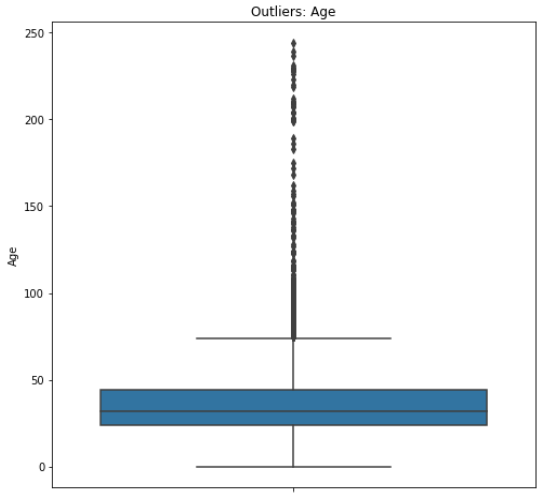
- The missing values in the feature

variable- *Age* are imputed with the median value.

- Also, for *Year-Of-Publication* we observed that the year mentioned was beyond 2020 for some entries whereas the dataset was created in 2004. The '0' values present in the same column don't make sense either. So, the anomalous entries are replaced with the median value.

2. Outlier Analysis

An outlier is an observation of a data point that lies an abnormal distance from other values in a given population. It is an abnormal observation during the Data Analysis stage, that the data point lies far away from other values.
There are outliers in the *Age* feature of the Users dataset.



The box plot shows that there are entries of age beyond 100. Therefore to make the data valid, the values greater than 80, and lesser than 10 are replaced with the median value.

3. Data Validity

The data in the "Ratings" dataset included ratings of the books that are not present in the "Books" dataset. Therefore ratings corresponding to the ISBN values that are outside the Books dataset are dropped.
Since the "Ratings" dataset contains the book rating information where ratings (*Book-Rating*) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0.

- Only the explicit ratings are considered to be effectively rated by the user, where the least rating possible is 1.
- '0' ratings indicate no user involvement in the book ratings. Hence the implicit ratings are dropped.

4. Merge Datasets

The "Ratings" and the "Books" dataset are merged on the common feature "ISBN". The resultant dataset is then merged with the "Users" dataset using the common feature "*UserID*".

5. Feature Engineering

The new features "*Average-Rating*" and "*Book-Rating-Count*" are formed using the "*Book-Rating*" feature information.

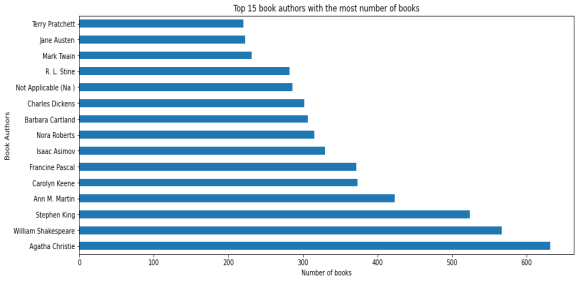
- The data is grouped based on ISBN to obtain the average ratings per book using mean transformation.
- Similarly, grouping based on "ISBN" and obtaining the count of ratings gives the "*Book-Rating-Count*" for each book.

Information regarding country and state names is extracted from the existing column "*Location*", and the column is dropped later.

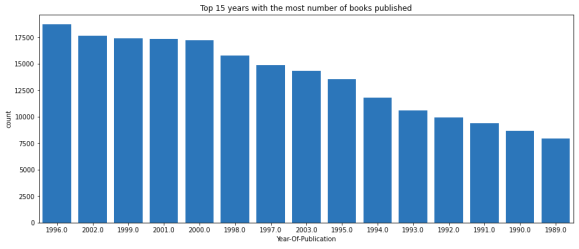
6. Feature Visualization

Hypothesis/Observations: Exploratory Data Analysis

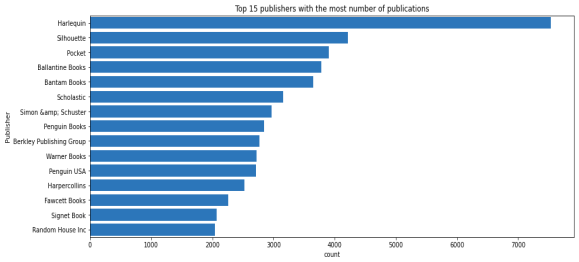
1. ***Agatha Christie, William Shakespeare, and Stephen King are the authors with the most number of books being published.***



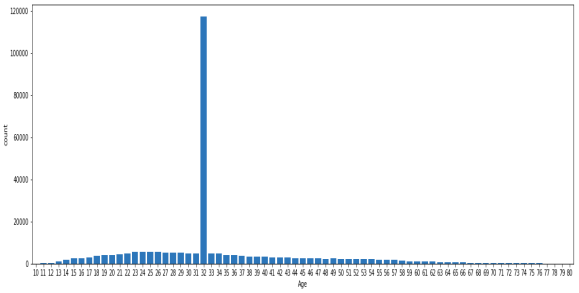
2. ***1996, 2002, and 1999 mark the years with the most number of books published.***



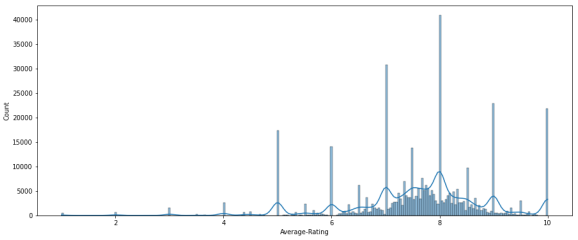
3. ***Harlequin, Silhouette, and Pocket have published the most number of books.***



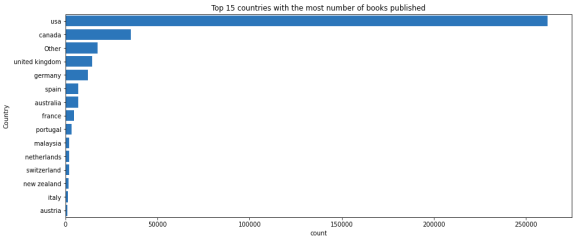
4. ***The information regarding the age of the author doesn't add any value to the analysis, therefore the feature is dropped.***

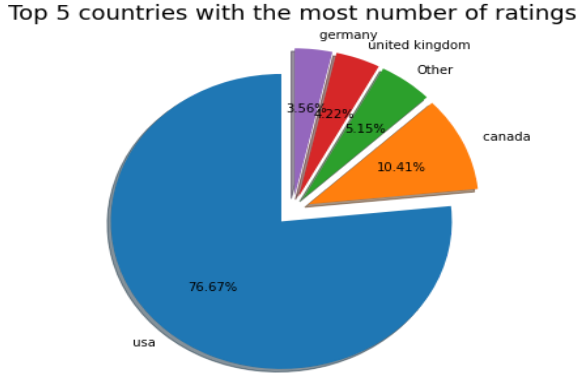


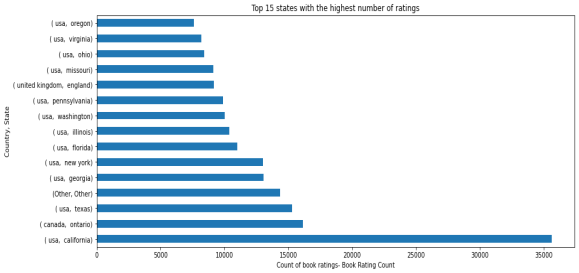
5. ***The mean average rating of the books is around 8, and the distribution is left-skewed.***



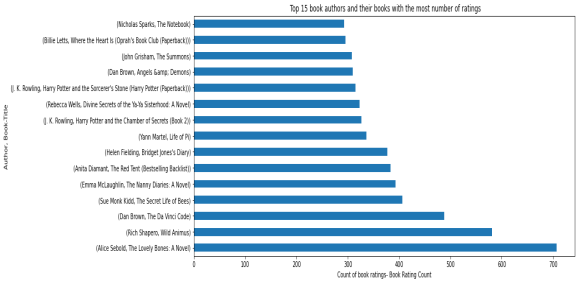
6. ***USA (California, Texas), Canada(Ontario), Uk, and Germany host the largest number of book publications and have obtained a large number of ratings as well.***



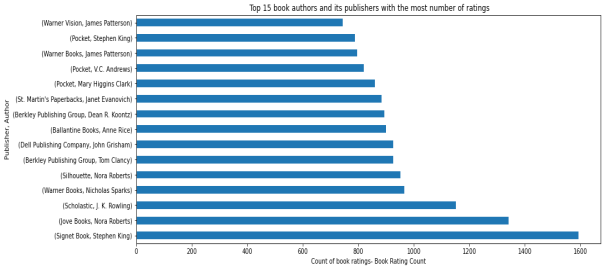




7. ***The Lovely Bones: A Novel by Alice Sebold, Wild Animus by Rich Shapero, and The Da Vinci Code by Dan Brown are the books that are among the highest-rated ones. These could be the most popular ones among the wide range of users.***



8. ***Stephen King, Nora Roberts, and J.K.Rowling are the authors with the most ratings.***



Recommender Systems

What is a Recommender System?

Recommender systems can forecast user ratings, even before they have provided one, making them an effective tool. Mainly, a recommendation system processes data through four phases as follows:

1. Collection

Data collected can be explicit (ratings and comments on products) or implicit (page views, order history, etc.) or any other form of item features.

2. Storing

The type of data used to create recommendations can help you decide the kind of storage you should use- NoSQL database, object storage, or standard SQL database.

3. Analyzing

The recommender system finds items with similar user engagement, users with similar item preferences, trending item purchases, user behavior and history, and data after analysis.

4. Filtering

This is the last step where data gets filtered to access the relevant information required to provide recommendations to the user. To enable this, you will need to choose an algorithm suiting the recommendation system.

How does it work?

There are a bunch of techniques that can be used when it comes to creating a recommendation system, like quantizing the popularity, user content-based, or making use of similarities between users/items. Few of the techniques are explored in this study based on suitability with the dataset.

1. ***Recommender System: Popularity***

The easiest approach would be to recommend the most popular items in the store. For example, for Netflix it would be the most popular movies, for Amazon the most sold items, for YouTube the most trending videos, and so on. It works on the principle of popularity and or anything which is in trend.

The popularity is quantified using the Average Ratings of the books and using book rating count as the threshold for filtering the data in the implementation of this approach.

Advantages of this approach include simplicity, and less computational usage and they are also really easy to keep updated. Otherwise, there is a lack of personalization, and poor accuracy in recommendations, which also means fewer profits generated.

2. ***Recommendation System: Weighted Average***

The recommendation index used for our books dataset is a weighted average rating. It is one of the types of popular recommendations.

The weighted Average is calculated as follows:

$$W = \frac{Rv + Cm}{v + m}$$

Where,

W is Weighted Average

v is the number of ratings for the book

m is the threshold- the minimum number of ratings required to be listed in the chart

R is the average rating of the book

C is the mean average ratings across the whole dataset- mean(R)

For the calculation of m, the 90th percentile is used as the threshold/cutoff. This means that a book is considered for a recommendation if it has more ratings than at least 90% of the books in the dataset.

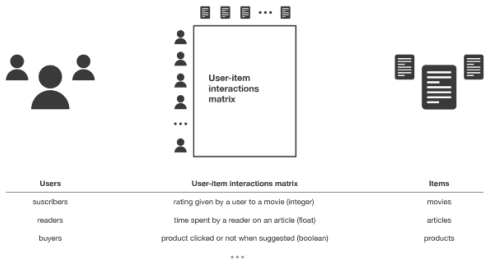
The weighted average score is then calculated for each of the books and recommendations are done based on the highest scores.

This approach is not sensitive to the interests and tastes of a particular user. It is not personalized and the system would recommend the same sort of products/books which are solely based upon the calculated formula of score to every other user. Hence to include user personalization different algorithms are explored.

3. ***Recommendation System: Collaborative-Filtering***

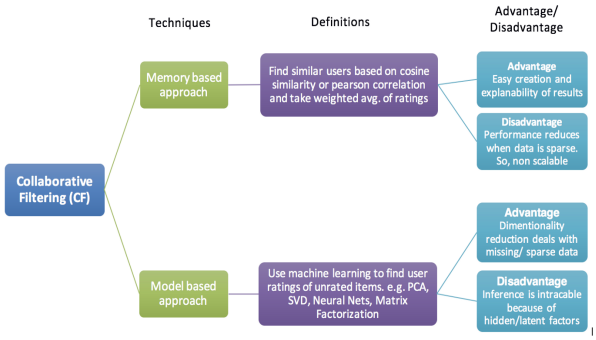
The collaborative filtering method is based on gathering and analyzing data on users' behavior. This includes the user's online activities, interaction with different items, and predicting ratings and insightful recommendations based on the similarity with other users.

For example, if user A likes to read "*Harry Potter*", "*The Wonderful of Wizard*", and "*The Secret Garden*" while user B likes "*Harry Potter*", "*The Little Prince*" etc., they have similar interests. So, it is highly likely that B would like "*The Secret Garden*" and A would enjoy "*The Little Prince*". This is how collaborative filtering takes place.



Two kinds of collaborative filtering techniques used are:

1. Memory-based collaborative filtering
2. Model-based collaborative filtering



One of the main advantages of this recommendation system is that it can recommend complex items precisely without understanding the object itself.

3a. ***Memory-Based Collaborative Filtering***

The main characteristic of this approach is that it extracts information from the user-item interaction matrix and they assume no model to produce new recommendations.

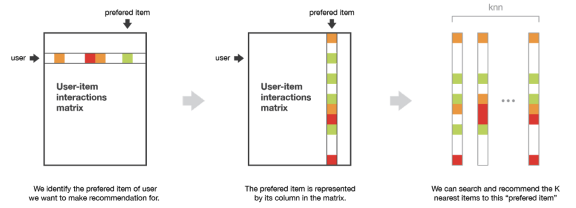
It consists of two different types of approaches based on User-User and Item-Item interactions. The latter is explored in our study.

Item-Item Memory-Based Collaborative Filtering

The idea here is to find items similar to the ones the user already "positively" interacted with.

- Two items are considered to be similar if most of the users that have interacted with both of them did it similarly.

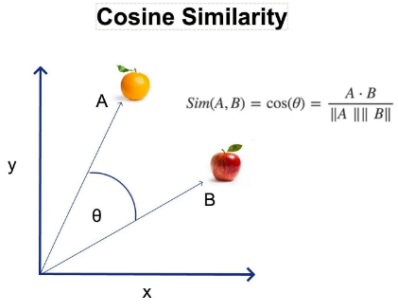
This method is said to be "item-centered" as it represents items based on interactions users had with them and evaluates distances between those items.



Suppose we wish to make a recommendation to a user. We study the user preferences, and the items the user mostly interacted with, represent this as an interactive vector, and find the similarities with other items represented as vectors. Therefore, the item vectors most similar to the interacted item vector that are new to our user of interest are considered for recommendation.

The similarity between items is found using various similarity techniques like cosine similarity/Jaccard similarity/Pearson's correlation coefficient, etc.

Cosine Similarity is explored in this case. The ultimate reason behind using cosine is that the value of cosine will increase with decreasing value of the angle between which signifies more similarity.



- The similarity is the cosine of the angle between the two vectors of the item vectors of A and B
- It determines whether two vectors are pointing in roughly the same direction. Hence we can understand how the two books are similar.
- The closer the vectors, the smaller will be the angle, and the larger the cosine

Memory-based collaborative filtering approaches that compute distance relationships between items or users have these two major issues:

- It doesn't scale particularly well to massive datasets, especially for real-time recommendations based on user behavior similarities which take a lot of computations.
- Rating matrices may be overfitting to noisy representations of user tastes and preferences. When we use distance-based "neighborhood" approaches on raw data, we match to sparse low-level details that we assume represent the user's preference vector instead of the vector itself.

3b. ***Model-Based Collaborative Filtering***

Model-based recommendation systems involve building a model based on the dataset of ratings. This approach potentially offers the benefits of both speed and scalability. This type of collaborative approach only relies on user-item interaction information and assumes a latent model supposed to explain these interactions.

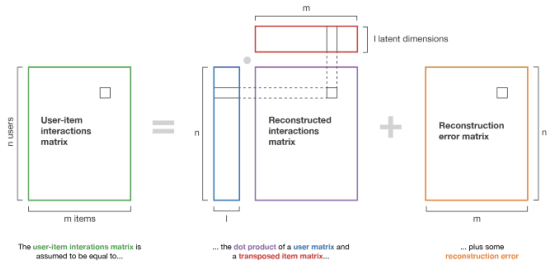
Therefore to overcome the disadvantages of memory-based techniques, dimensionality reduction techniques are used to derive the tastes and preferences from the raw data, otherwise known as doing low-rank matrix factorization. Perhaps the more popular technique for dimensionality reduction in machine learning is Singular Value Decomposition.

***Singular Value Decomposition* (SVD)**

- SVD is a method of linear algebra that has been generally used as a dimensionality reduction technique in machine learning.
- It is a matrix factorization technique which reduces the number of features of a dataset by reducing the space dimension from N-dimension to K-dimension (where K<N).
- It uses a matrix structure called ***Pivot-Table***, where each row represents an item, and each column represents a user. The elements of this matrix are the ratings that are given to items by users.

(Pivot Table is a utility matrix that consists of an index, columns, and values.)

- It can be formatted to use for calculating similarity/correlation. As the similarity will be higher we can use them as our recommendation.
- It is a sparse table.



The factorization of this matrix is done by the SVD. It finds factors of matrices from the factorization of a high-level (user-item-rating) matrix. The singular value decomposition is a method of decomposing a matrix into three other matrices as given below:

$$A = USV^T$$

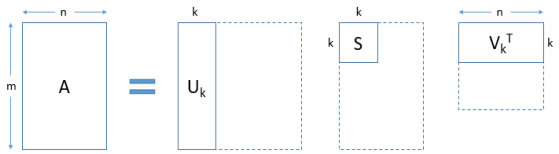
Where,

A is a m x n utility matrix,

U is a m x k orthogonal left singular matrix, which represents the relationship between users and latent factors,

S is a k x k diagonal matrix, which describes the strength of each latent factor.

V is a k x n diagonal right singular matrix, which indicates the similarity between items and latent factors.



The latent factors here are the characteristics of the items, for example, the ratings of the books. The SVD decreases the dimension of the utility matrix A by extracting its latent factors. It maps each user and each item into a k-dimensional latent space (k<N).

SVD is efficient, with a hierarchical basis, ordered by relevance. Therefore performs quite well.

Cold Start Problem

Cold start is a potential problem in computer-based information systems which involves a degree of automated data modeling. Specifically, it concerns the issue that the system cannot draw any inferences for users or items about which it has not yet gathered sufficient information. Recommender systems form a specific type of information filtering (IF) technique that attempts to present information items (e-commerce, films, music, books, news, images, web pages) that are likely interesting to the user. Typically, a recommender system compares the user's profile to some reference characteristics. Depending on the system, the user can be associated with various interactions: ratings, bookmarks, purchases, likes, number of page visits, etc.

There are three instances of a cold start:

1. **New community**: refers to the start-up of the recommender, when, although a catalog of items might exist, almost no users are present and the lack of user interaction makes it very hard to provide reliable recommendations.
2. **New item**: a new item is added to the system that might have some content information but no interactions are present.
3. **New user**: a new user registers and has not provided any interaction yet, therefore it is not possible to provide personalized recommendations.

Hence, possible solutions are

- Recommending random items to new users or new items to random users (random strategy).
- Recommending popular items to new users or new items to the most active users (maximum expectation strategy)

- Recommending a set of various items to new users or a new item to a set of various users (exploratory strategy)
- Using a non-collaborative method for the early life of the user or the item.

Understanding

Therefore, the recommender system based on popularity, and/or weighted average are highly essential to avoid the cold start problem. To get started with the system, under all three cases of cold-start as discussed, methods based on popularity, and weighted average are handy to form a hybrid model along with other techniques that make use of user preferences.

Evaluation Metrics and Visualizations

Evaluation methods for recommender systems can mainly be divided into two sets:

- The evaluation is based on well-defined metrics

If the recommender system is based on a model that outputs numeric values such as rating predictions.

RMSE is used in our model.

- The evaluation is mainly based on human judgment and satisfaction estimation.

If the recommender system is not based on numeric values and only returns a list of recommendations

Our analysis will focus on book recommendations based on the Book-Crossing dataset. ***To reduce the dimensionality of the dataset, avoid running into memory errors and bring significance we will focus on users with at least 60 ratings and books with at least 10 ratings. This gives us the filtered data, that is used for collaborative filtering techniques.***

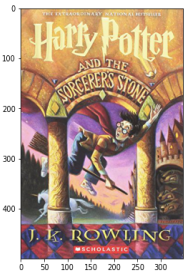
Results

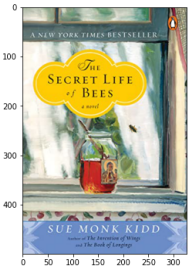
<u>Popularity-Based Recommender System</u>

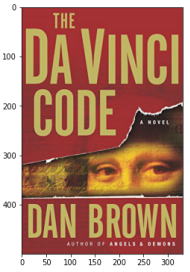
The recommendation based on popularity is done by considering the *Average Rating* of the book. Initially, data is filtered by considering the books that are having a count of Book Rating greater than the popularity threshold (minimum number of ratings for a book to be considered for the recommendation). Then the available list of books is sorted in decreasing fashion based on the *Average Rating.*

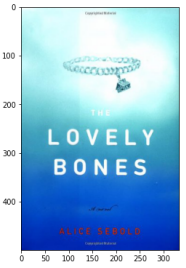
The most popular books recommended are as follows: For "popularity threshold"= 300

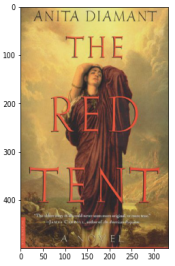
	Book-Title	Book-Author	Publisher	Average-Rating	Book-Rating-Count
0	Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))	J. K. Rowling	Arthur A. Levine Books	8.939297	313
1	The Secret Life of Bees	Sue Monk Kidd	Penguin Books	8.452769	307
2	The Da Vinci Code	Dan Brown	Doubleday	8.435318	487
3	The Lovely Bones: A Novel	Alice Sebold	Little, Brown	8.185290	707
4	The Red Tent (Bestselling Backlist)	Anita Diamant	Picador USA	8.182768	383











We observe that the above-mentioned books are highly popular ones(refer to data visualization).

Weighted Average Recommender System

The books are arranged in decreasing fashion of weighted average calculated according to the formula as discussed.

The most recommended books according to the weighted average method are as follows:

	Book-Title	Book-Author	Publisher	Average-Rating	Book-Rating-Count	Weighted-Average-Score
0	Harry Potter and the Goblet of Fire (Book 4)	J. K. Rowling	Scholastic	9.262774	137	8.741835
1	Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))	J. K. Rowling	Arthur A. Levine Books	8.939297	313	8.716469
2	Harry Potter and the Order of the Phoenix (Book 5)	J. K. Rowling	Scholastic	9.033981	206	8.700403
3	To Kill a Mockingbird	Harper Lee	Little Brown & Company	8.943925	214	8.640679
4	Harry Potter and the Prisoner of Azkaban (Book 3)	J. K. Rowling	Scholastic	9.082707	133	8.609690

Figure: Top 10 highly recommended books based on the weighted average

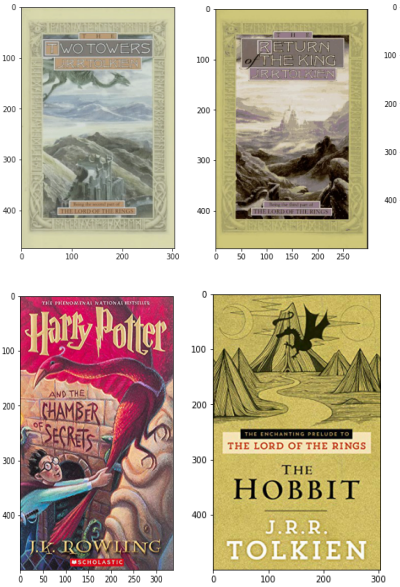
Memory-Based Recommender System

The data is filtered to consider the significant ratings. It is done by considering the users with at least 60 ratings and books with at least 10 ratings. Given a book of interest, to a user, similar books are recommended based on the highest similarity scores that are computed using the pivot table.

Therefore 5 books similar to the book "The Lord of the Rings Part1" are given as follows:

```
collaborative_memory_based('The Fellowship of
the Ring (The Lord of the Rings, Part 1),5)
```

	Book-Title	Book-Author	Publisher	Average-Rating	Book-Rating-Count	Similarity-Score
0	The Two Towers (The Lord of the Rings, Part 2)	J. R. R. Tolkien	Houghton Mifflin	9.500000	4	0.260915
1	The Return of the King (The Lord of the Rings, Part 3)	J. R. R. Tolkien	Houghton Mifflin	9.750000	4	0.260915
2	Harry Potter and the Goblet of Fire (Book 4)	J. K. Rowling	Scholastic	9.262774	137	0.260915
3	Harry Potter and the Chamber of Secrets (Book 2)	J. K. Rowling	Scholastic	8.783069	189	0.260915
4	The Hobbit : The Enchanting Prelude to The Lord of the Rings	J.R.R. TOLKIEN	Del Rey	8.739130	161	0.260915



We observe that the recommended books similar to "*The Lord of the Rings Part1*" *are* all fantasy fiction books.

Model-Based Recommender System

The data is filtered to consider the significant ratings. It is done by considering the users with at least 60 ratings and books with at least 10 ratings. Given a UserId, the recommender suggests the kind of books the user might like that he hasn't interacted with before. The model considers the predictions that are reduced in dimension using SVD.

Therefore given a UserId, the recommender gives the following results:

1. UserData: The information of books the user has rated before
2. The “n” number of book suggestions according to the interest of the user.

The user data and recommendations to UserId-1424 are as follows:

```
user_data,recos=collaborative_model_based(ratings_predicted,1424,final_filtered_data,5)
```

User Data: 1424 (Books rated by the user)

	Book-Title	Book-Author	Publisher	Book-Rating	Average-Rating	Book-Rating-Count
0	A Walk in the Woods: Rediscovering America on the Appalachian Trail (Official Guides to the Appalachian Trail)	Bill Bryson	Broadway Books	8	8.207547	106
1	Prey	Michael Crichton	Avon Books	8	7.571429	77
2	A Map of the World	Jane Hamilton	Anchor Books/Doubleday	7	7.000000	111
3	The Poisonwood Bible	Barbara Kingsolver	HarperTorch	7	8.264706	34
4	The Name of the Rose: including Postscript to the Name of the Rose	Umberto Eco	Harvest Books	8	8.523810	21
5	The Joy Luck Club	Amy Tan	Prentice Hall (K-12)	6	8.195876	194
6	Plain Truth	Jodi Picoult	Washington Square Press	8	8.148936	47
7	The Poisonwood Bible: A Novel	Barbara Kingsolver	Perennial	7	8.178899	218
8	The Bean Trees	Barbara Kingsolver	HarperTorch	5	7.861111	72
9	Memoirs of a Geisha Uk	Arthur Golden	Trafalgar Square	8	8.174419	86
10	Year of Wonders	Geraldine Brooks	Penguin Books	7	8.318182	88

Recommendations to User 1424:

	Book-Title	Book-Author	Publisher	Average-Rating	Book-Rati
0	1984	George Orwell	Signet Book	8.772277	
1	1st to Die: A Novel	James Patterson	Little Brown and Company	7.661017	
2	2010: Odyssey Two	Arthur C. Clarke	Del Rey Books	7.413793	
3	2061: Odyssey Three	Arthur C. Clarke	Del Rey Books	7.666667	
4	2nd Chance	James Patterson	Warner Vision	7.722222	

Evaluation of model-based collaborative filtering recommendation system

The predictions are based on the SVD model used for evaluating the model. The SVD model is hyperparameter tuned to obtain the ***RMSE value of 1.50***

“The deviation between the predicted value and the real value of different users and items are measured using RMSE”

The prediction of the rating of the book that the user has already interacted with is as follows:

- Prediction of the rating of the book “The Poisonwood Bible” rated by user 1424.

```
model.predict("The Poisonwood Bible",1424)
output:
Prediction(uid='The Poisonwood Bible', iid=1424, r_ui=None, est=7.382737779974534, details={'was_impossible': False})
```

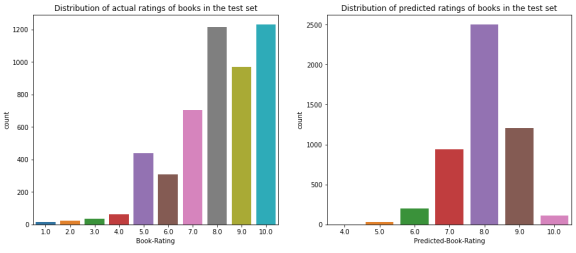
Estimated rating: 7.38
Actual rating: 7

- Prediction of the rating of the book “1st to Die: A Novel” that user 1424 might rate:

```
model.predict("1st to Die: A Novel",1424)
output:
Prediction(uid='1st to Die: A Novel', iid=1424, r_ui=None, est=7.672810426877989, details={'was_impossible': False})
```

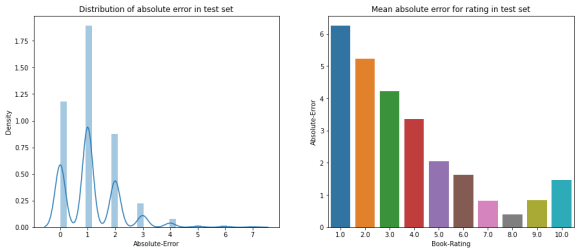
Estimated rating: 7.67

Plotting Actual and Predicted Book Rating



- Most of the books are rated as 8. In actuality, the ratings are widely distributed between 4-10.
- Whereas in prediction the ratings are mostly 8.

Plotting the absolute error between actual and predicted ratings



There is a large absolute error in lower rating values compared to that of higher rating values.

Conclusion

The book recommendation is carried out based on the dataset providing the users, ratings, and book information. Data preprocessing helped in obtaining unerring data. Exploratory data analysis supports the hypothesis based on the visualizations of the book ratings, authors, and publishers.

Explored different kinds of recommendation systems where the popularity-based systems emphasize the system free of cold-start problems. Therefore a hybrid model of popularity-based and collaborative filtering models can be considered for future exploration. The availability of book summaries and genres would have helped in implementing the hybrid model of content-based and collaborative filtering systems.

Note: Kindly zoom in on the tables in the visualization and result images.

References

- Cold Start Problem
- Collaborative Filtering
- Krish Naik - Recommender System tutorial
- Evaluation Techniques
- Weighted-Average-Score Recommendation System
- Intro to Recommendation System