Capstone Project-4

Book Recommendation System

By

Neeraja C

Data Science Trainee|AlmaBetter

What actually is Recommendation System

A recommendation engine is a class of machine learning which offers relevant suggestions to the customer. Before the recommendation system, the major tendency to buy was to take a suggestion from friends. But Now Google knows what news you will read, Youtube knows what type of videos you will watch based on your search history, watch history, or purchase history.

A recommendation system helps an organization to create loyal customers and build trust by them desired products and services for which they came on your site. The recommendation system today are so powerful that they can handle the new customer too who has visited the site for the first time. They recommend the products which are currently trending or highly rated and they can also recommend the products which bring maximum profit to the company.

Types Of Recommendation System

Mainly three types of recommendation systems in machine learning based on filtering are used to suggest product and services to the consumers.

Content Filtering

Collaborative Filtering

Hybrid Filtering

1. Content Filtering: In this algorithm, we try finding items look alike. Once we have item look like matrix,we can easily recommend alike items to a customer, who has purchased any item from the store.
2. Collaborative Filtering: Here, we try to search for look alike customers and offer products based on what his/her lookalike has chosen.This algorithm is very effective but takes a lot of time and resources.
3. Hybrid Filtering (Content Filtering + Collaborative Filtering): Both Content Filtering & Collaborative Filtering is used for the purpose. you-tube uses this algorithm for their strong recommendation system.

Book Recommendation System

A book recommendation system is a type of recommendation system where we have to recommend similar books to the reader based on his interest. The books recommendation system is used by online websites which provide e-books like google play books, open library, good Read's etc.

In this project, we will use the Collaborative based filtering method to build a book recommender system.

Collaborative Filtering (CF) is currently the most widely used approach to build recommendation systems and uses the users' behavior in the form of user-item ratings for predictions.

Dataset Description

The Book-Crossing dataset comprises 3 files.

- Users Contains the users. Note that user IDs (User-ID) have been anonymized and map 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, Book-Author, Year-Of-Publication, Publisher), 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 different flavors (Image-URL-S, Image-URL-M, Image-URL-L), i.e., small, medium, 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.

Data Preprocessing:

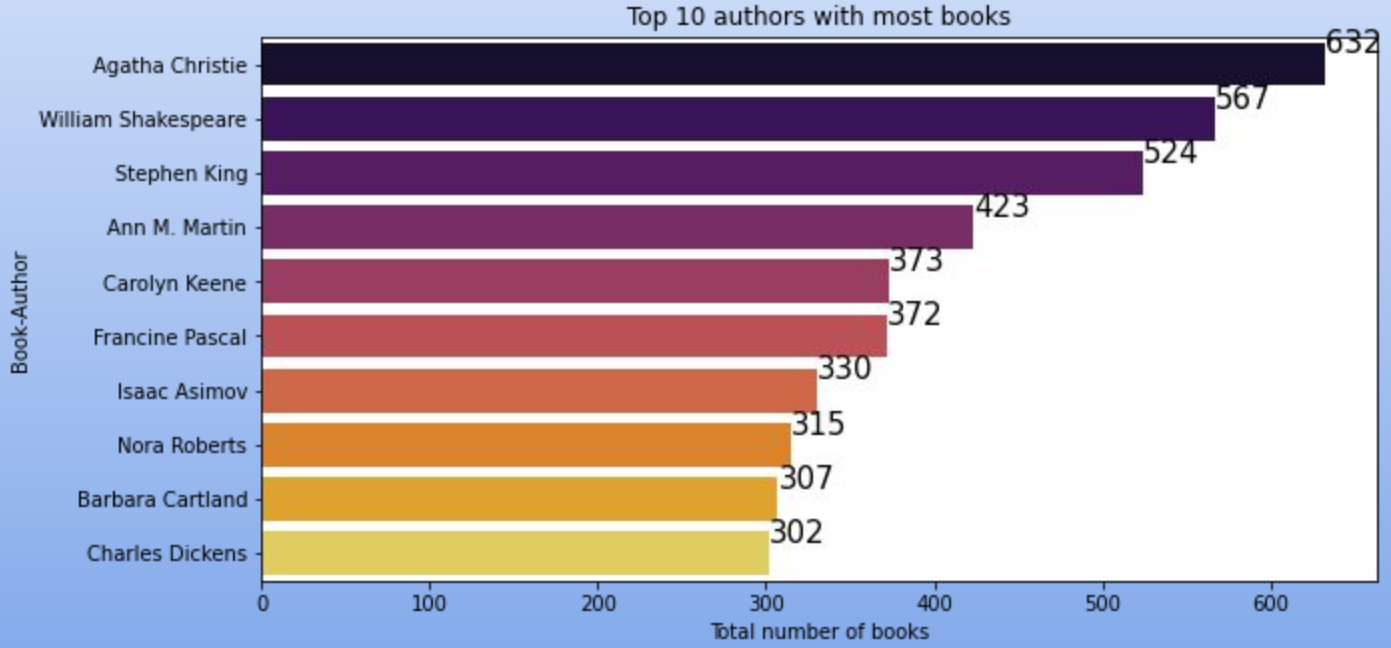
we have some extra columns which are not required for our task like image URLs. And we will rename the columns of each file as the name of the column contains space, and uppercase letters so we will correct as to make it easy to use.

Exploratory Data Analysis

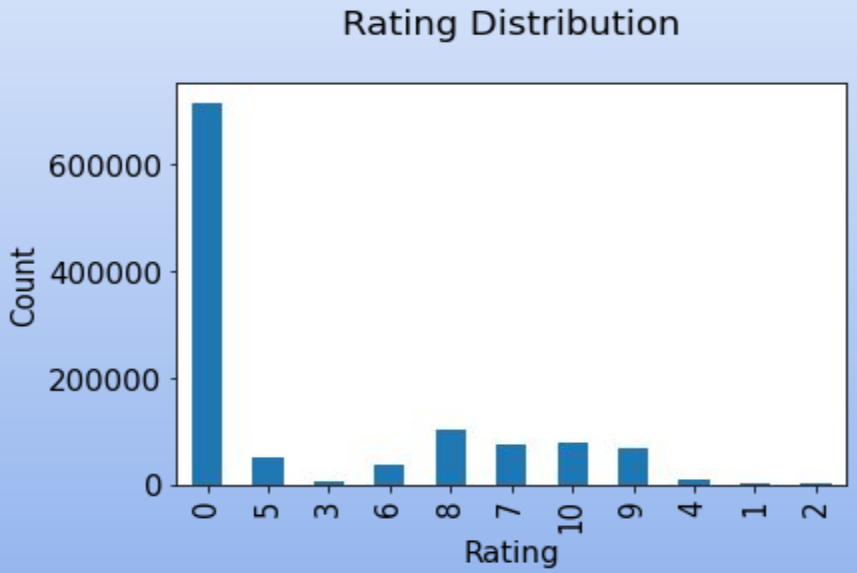
We don’t need all the books and all the users for modeling.

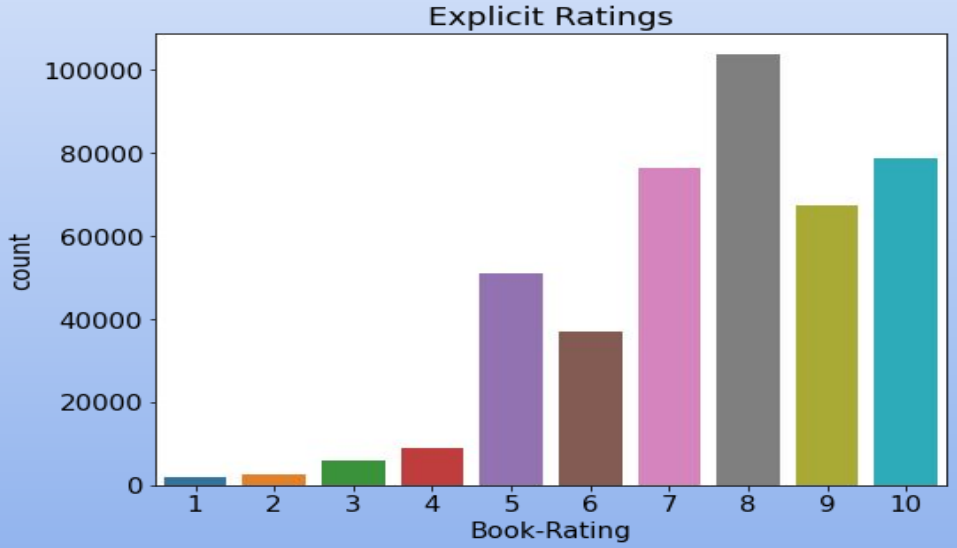
So what we have to do is we have to decrease the number of users and books because we cannot consider a user who has only registered on the website or has only read one or two books. On such a user, we cannot rely to recommend books to others because we have to extract knowledge from data. So what we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

Top 10 Authors



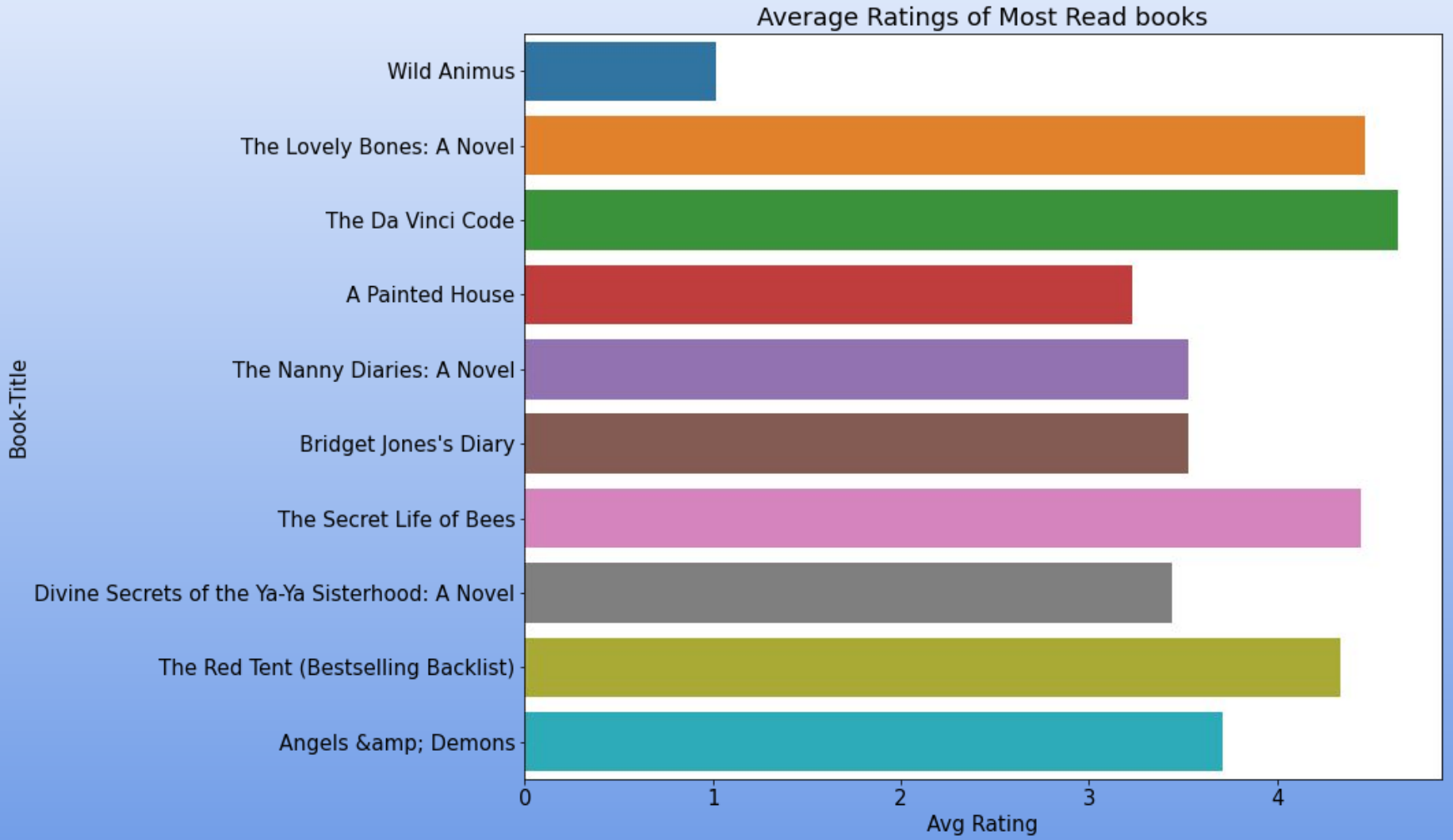
These are the top 10 authors with most number of books.

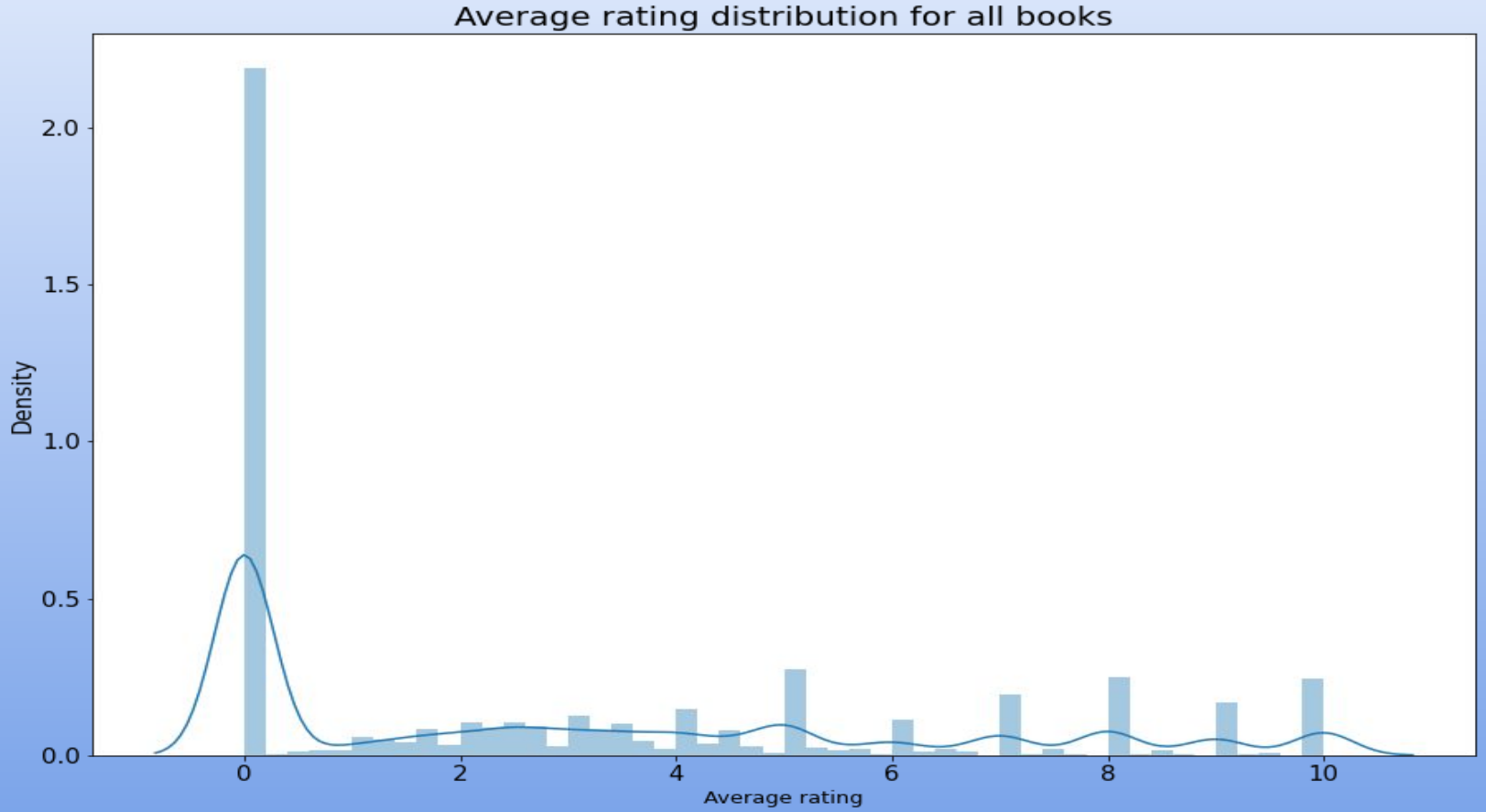


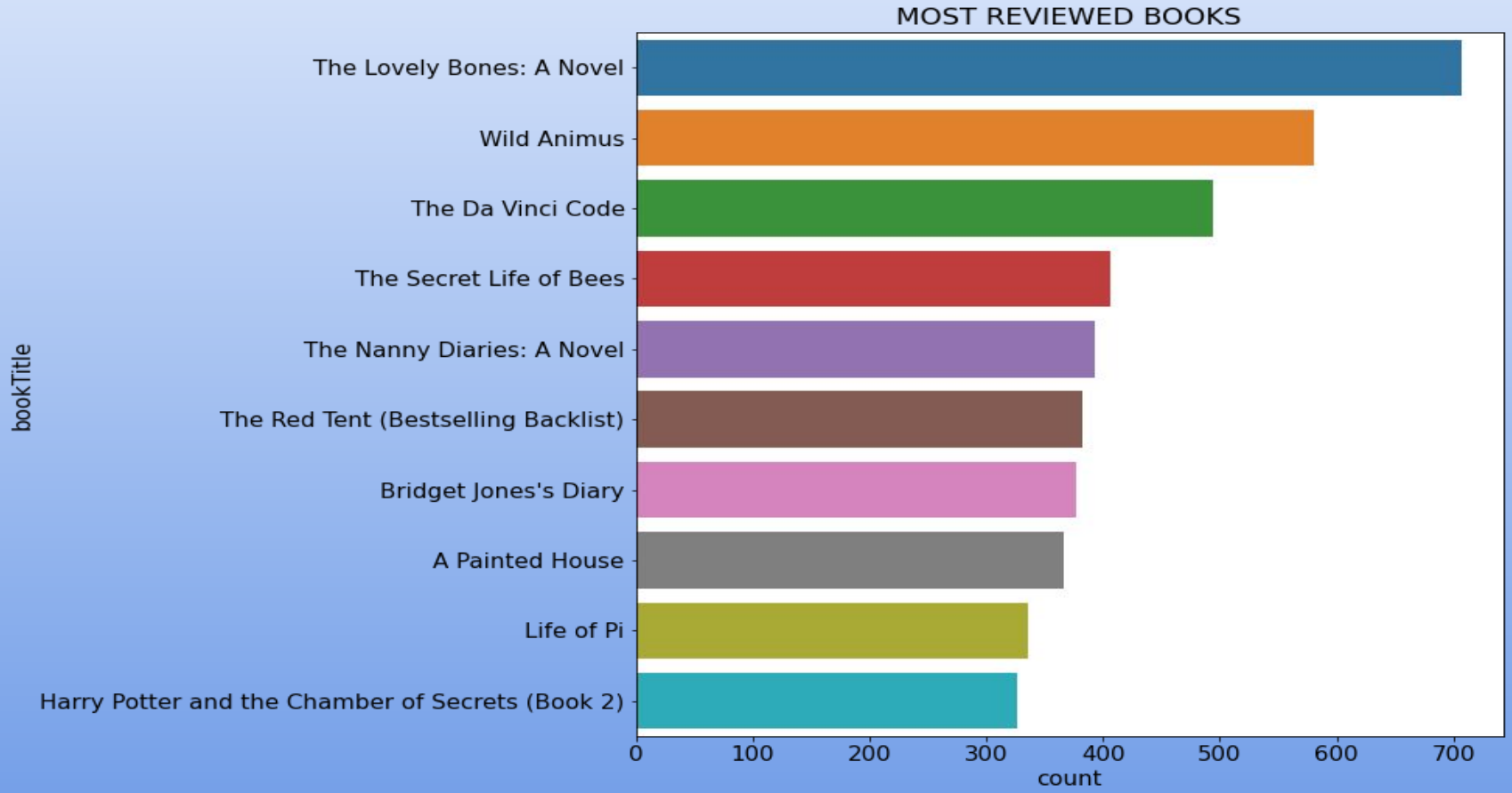


Zero rating is more in dataset. So we can say that most of the people leaving without give rating to the books.

Most of the books have high ratings


Average Ratings of Most Read books
Wild Animus
The Lovely Bones: A Novel
The Da Vinci Code
A Painted House
The Nanny Diaries: A Novel
Bridget Jones's Diary
The Secret Life of Bees
Divine Secrets of the Ya-Ya Sisterhood: A Novel
The Red Tent (Bestselling Backlist)
Angels & Demons
Book-Title
0
1
2
3
4
Avg Rating

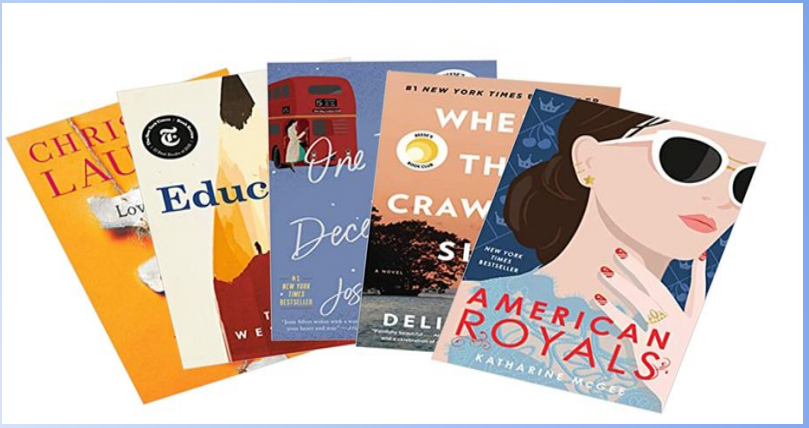

Average rating distribution for all books
Density
2.0
1.5
1.0
0.5
0.0
0
2
4
6
8
10
Average rating


MOST REVIEWED BOOKS
The Lovely Bones: A Novel
Wild Animus
The Da Vinci Code
The Secret Life of Bees
The Nanny Diaries: A Novel
The Red Tent (Bestselling Backlist)
Bridget Jones's Diary
A Painted House
Life of Pi
Harry Potter and the Chamber of Secrets (Book 2)
bookTitle
0
100
200
300
400
500
600
700
count

Popularity Based Recommendation System

It is a type of recommendation system which works on the principle of popularity or anything which is on trend.

These systems check about the items which are in trend or are most popular among the users and directly recommend those.



Top 5 Popular books are:

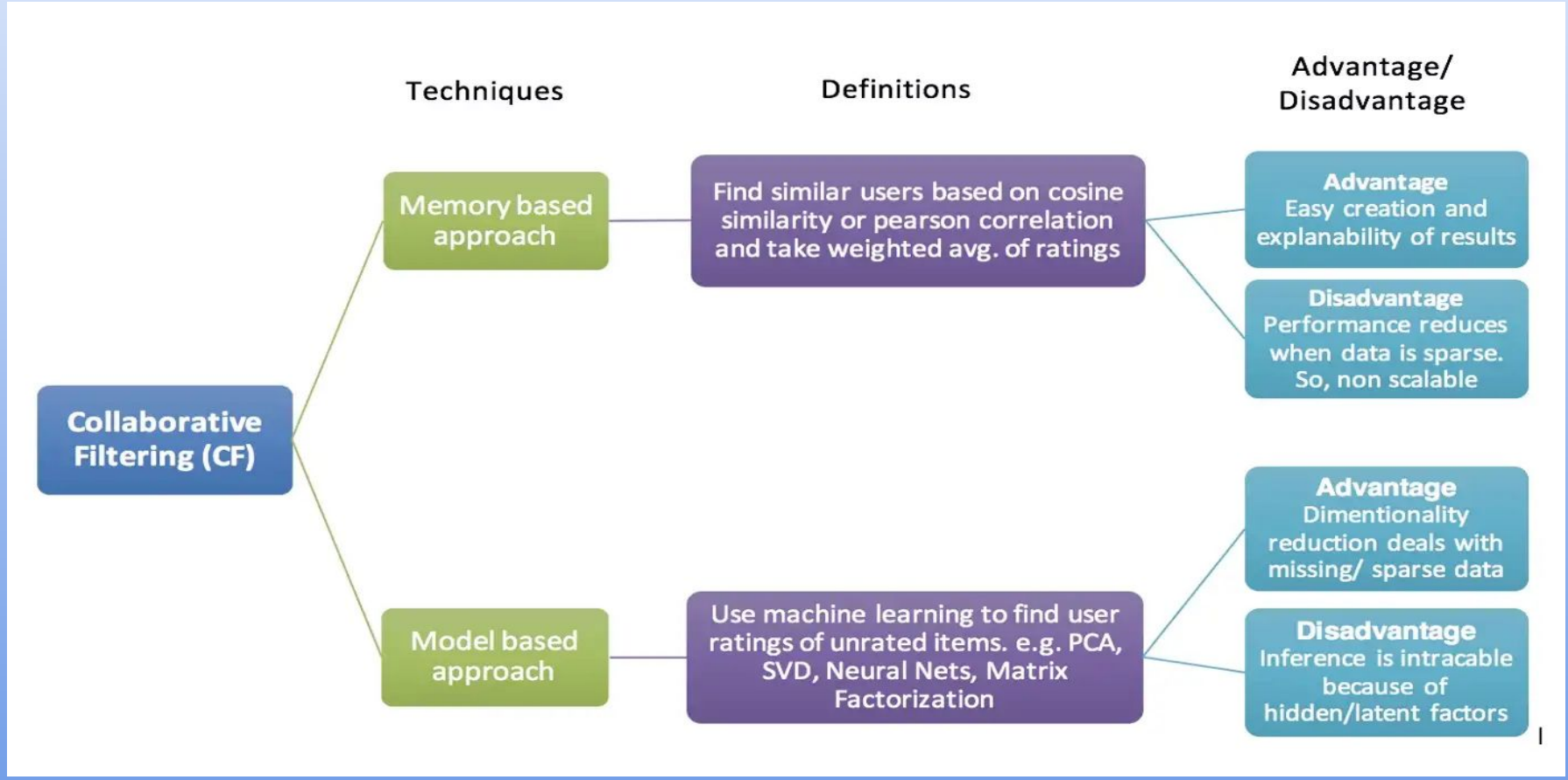
	ISBN	Book-Rating	Book-Title	Book-Author	Year-Of-Publication	Publisher
0	0316666343	707	The Lovely Bones: A Novel	Alice Sebold	2002	Little, Brown
1	0971880107	581	Wild Animus	Rich Shapero	2004	Too Far
2	0385504209	487	The Da Vinci Code	Dan Brown	2003	Doubleday
3	0312195516	383	The Red Tent (Bestselling Backlist)	Anita Diamant	1998	Picador USA
4	0060928336	320	Divine Secrets of the Ya-Ya Sisterhood: A Novel	Rebecca Wells	1997	Perennial

Collaborative Filtering

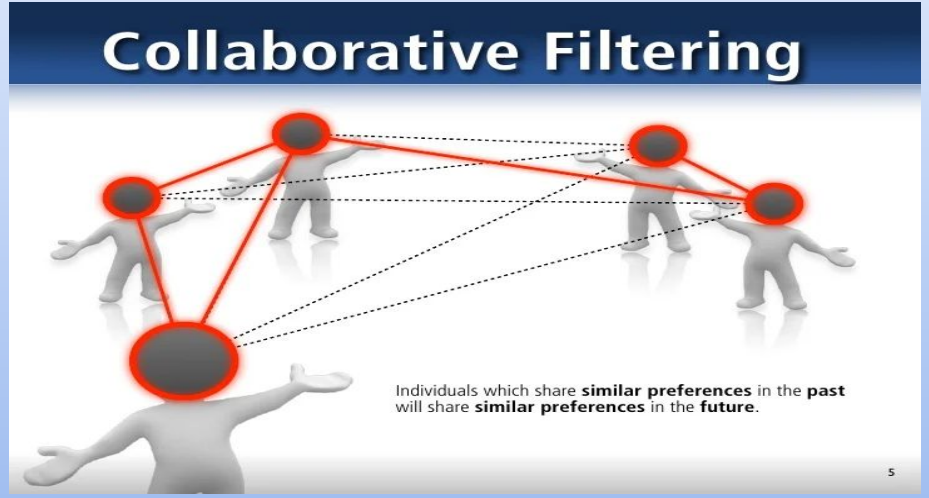
Collaborative based filtering recommender systems are based on past interactions of users and target items. In simple words here, we try to search for the look-alike customers and offer products based on what his or her lookalike has chosen.

Let us understand with an example. X and Y are two similar users and X user has watched A, B, and C movie. And Y user has watched B, C, and D movie then we will recommend A movie to Y user and D movie 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 not at all related to your history but then also it recommends it because the other person similar to you has watched it.


Techniques
Definitions
Advantage/ Disadvantage
Collaborative Filtering (CF)
Memory based approach
Find similar users based on cosine similarity or pearson correlation and take weighted avg. of ratings
Advantage
Easy creation and explanability of results
Disadvantage
Performance reduces when data is sparse. So, non scalable
Model based approach
Use machine learning to find user ratings of unrated items. e.g. PCA, SVD, Neural Nets, Matrix Factorization
Advantage
Dimentionality reduction deals with missing/ sparse data
Disadvantage
Inference is intracable because of hidden/latent factors

Approach 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 this 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.

First we will count the rating of each book so we will group data based on title and aggregate based on rating. Finally, we have a dataset with that user who has rated more than 200 books and books that received more than 50 ratings.

We convert the rating column to a 2D matrix. The matrix will be sparse because not every user rated every book.

We have prepared our dataset for modeling.

Memory-Based Approach

The recommendation system in its core algorithm uses a fundamental mathematical metric called “similarity”, which compares and quantifies the similarity between two items: user selected vs rest of items in the catalog. The list of items with high similarity values to the ones that the user selected are recommended.
There are many similarity metrics used in recommendation systems.The most commonly used similarity metric is Cosine similarity.

Cosine Similarity

In data analysis, cosine similarity is a measure of similarity between two sequences of numbers. For defining it, the sequences are viewed as vectors in an inner product space, and the cosine similarity is defined as the cosine of the angle between them, that is, the dot product of the vectors divided by the product of their lengths.

It follows that the cosine similarity does not depend on the magnitudes of the vectors, but only on their angle. The cosine similarity always belongs to the interval [-1,1].

For example, two proportional vectors have a cosine similarity of 1, two orthogonal vectors have a similarity of 0, and two opposite vectors have a similarity of -1. The cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0,1].

For example, in information retrieval and text mining, each word is assigned a different coordinate and a document is represented by the vector of the numbers of occurrences of each word in the document. Cosine similarity then gives a useful measure of how similar two documents are likely to be, in terms of their subject matter, and independently of the length of the documents.

The technique is also used to measure cohesion within clusters in the field of data mining.
One advantage of cosine similarity is its low complexity, especially for sparse vectors: only the non-zero coordinates need to be considered.

Model-Based Approach

Let's look at how a recommendation problem can be translated into a matrix decomposition context.

The idea behind such models is that the preferences of users can be determined by a small number of hidden factors. We can call these factors Embeddings.

Embeddings are low dimensional hidden factors for items and users.

For e.g., say we have 5-dimensional embeddings for both items and users (5 chosen randomly, this could be any number - as we saw with PCA and dimensionality reduction).

For user-X & movie-A, we can say those 5 numbers might represent 5 different characteristics about the movie, e.g.:

- How much movie-A is political
- How recent is the movie
- How much special effects are in movie A
- How dialogue driven is the movie
- How linear is the narrative in the movie

In a similar way, 5 numbers in the user embedding matrix might represent:

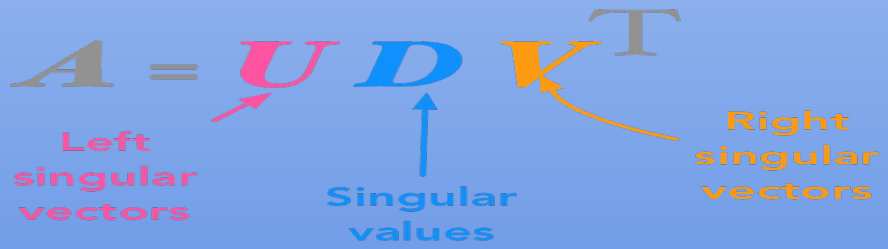
- How much does user-X like sci-fi movies
- How much does user-X like recent movies … and so on

Now the big question is, how do we obtain these embedding matrices? One of the most common factorization techniques in the context of recommender systems is Singular Value Decomposition.

Singular Value Decomposition

Singular-Value Decomposition or SVD is a common and widely used matrix decomposition method. All matrices are able to be factored using an SVD, which makes it more stable than other methods, such as the eigendecomposition. As such, it is often used in a wide array of applications including compression and data reduction.

In simple terms, SVD is the factorization of a matrix into 3 matrices. So if we have a matrix A, then its SVD is represented by the equation:



With SVD, we turn the recommendation problem into an Optimization problem that deals with how good we are in predicting the rating for items given a user.

SciPy has a straightforward implementation of SVD to help us avoid all the complex steps of SVD. We can use the svds() function to decompose a matrix.

SVD decreases the dimension of the utility matrix by extracting its latent factors.

Essentially, we map each user and each item into a latent space with lower dimension. Therefore, it helps us better understand the relationship between users and items as they become directly comparable.

I have successfully built a book recommendation system which recommends books corresponding to each user.

SVD is not a perfect solution, but when we have enough users and items, we are able to gain valuable insights about the underlying relationships found in our data.

Conclusion

This is a wonderful Unsupervised learning project. Here I have to build a reliable Book Recommendation system and I have done lots of preprocessing.

There was 3 files in the dataset so I have to combine that files as per our need. There was some extra columns which were not required for our task. So after all these pre-processing steps I have used Collaborative based filtering method to build a book recommender system.

By this project I am able to

- Compare and contrast the advantages/disadvantages of memory vs. model-based recommender systems
- Describe how memory-based collaborative filtering methods work
- Describe how model-based collaborative filtering models work
- Explain how SVD is able to extract meaning with latent factors
- Implement SVD using SciPy