Book Recommendation System

Shubham Srivastava
Data science trainee,
AlmaBetter, Bangalore

Abstract:

Today the amount of information in the internet growth very rapidly and people need some instruments to find and access appropriate information. One of such tools is called recommendation system. Recommendation systems help to navigate quickly and receive necessary information. Generally they are used in Internet shops to increase the profit. Personal recommendation systems have been emerged to conduct effective search which mine related books based on user rating and interest. Most of these existing systems are user-based ratings where content-based and collaborative based learning methods are used.

***Keywords—recommendation system , collaborative filtering, book*.**

1.Problem Statement

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). Recommender systems are really critical in some industries as they can generate a huge amount of income when they are efficient or also be a way to stand out significantly from competitors.

The main objective is to create a book recommendation system for users.

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.

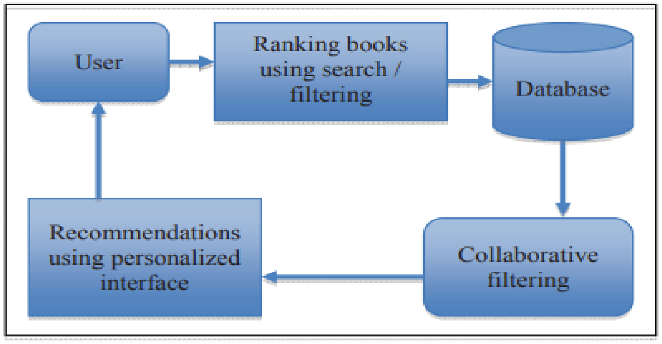
● Ratings :

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.

2. Introduction

Nowadays the amount of information especially in Internet growth very rapidly. Finding necessary information becomes more difficult. Recommendation systems aim to solve this kind of problems. With the help of them one can quickly access relevant information without searching the web manually. As such many web sites today benefit from recommendation systems to promote and sell their products. There is a wide range of products like music, movies, articles and etc. that can be recommended to the customer based on their profiles in internet shops or even social networks, browsing history such as visited links, browsing activity like number and time of visits and other online behavior. Online shops are increasing their sales using such technologies.

Recommendation systems use different kind of approaches to provide relevant recommendation. Traditionally collaborative filtering and content based filtering are used.
The collaborative filtering, in contrast, doesn’t rely on content and matches items with users based on the idea that those users who agreed in the past will also agree in the future. The data about their preferences can be collected upon ratings they give on the items. One of the successful implementations and use of collaborative filtering was done by Amazon company that recommends their wide range of products in a very efficient way.


User
Ranking books using search / filtering
Database
Collaborative filtering
Recommendations using personalized interface

3. Steps involved:

- **Data Cleaning and Preprocessing**

The dataset consists of three tables; Books, Users, and Ratings. Data from all three tables are cleaned and preprocessed separately as defined below briefly:

For Books Table:

- Drop all three Image URL features.
- Check for the number of null values in each column. There comes only 3 null values in the table. Replace these three empty cells with ‘Other’.
- Check for the unique years of publications. Two values in the year column are publishers. Also, for three tuples name of the author of the book was merged with the title of the book. Manually set the values for these three above obtained tuples for each of their features using the ISBN of the book.
- Convert the type of the years of publications feature to the integer.
- By keeping the range of valid years as less than 2022 and not 0, replace all invalid years with the mode of the publications that is 2002.
- Upper-casing all the alphabets present in the ISBN column and removal of duplicate rows from the table.

For Users Table:

- Check for null values in the table. The Age column has more than 1 lakh null values.
- Check for unique values present in the Age column. There are many invalid ages present like 0 or 244.
- By keeping the valid age range of readers as 10 to 80 replace null values and invalid ages in the Age column with the mean of valid ages.
- The location column has 3 values city, state, and country. These are split into 3 different columns named; City, State, and Country respectively. In the case of null value, ‘other’ has been assigned as the entity value.

- Removal of duplicate entries from the table.

For Ratings Table:

- Check for null values in the table.
- Check for Rating column and User-ID column to be an integer.
- Removal of punctuation from ISBN column values and if that resulting ISBN is available in the book dataset only then considering else drop that entity.
- Upper-casing all the alphabets present in the ISBN column.
- Removal of duplicate entries from the table.

Algorithms Implemented

Popularity Based Recommendation :

- Popular in the Whole Collection

We have sorted the dataset according to the total ratings each of the books have received in non-increasing order and then recommended top n books.

- Popular at a Given Place

The dataset was filtered according to a given place (city, state, or country) and then sorted according to total ratings they have received by the users in decreasing order of that place and recommended top n books.

- Books By the Same Author, Publisher of Given Book Name

For this model, we have sorted the books by rating for the same author and same publisher of the given book and recommended top n books.

- Popular Books Yearly

This is the most basic model in which we have grouped all the books published in the same year and recommended the top-rated book yearly.

Recommendation using Average Weighted Rating:

We have calculated the weighted score using the below formula for all the books and recommended the books with the highest score.

$$\text{score} = t/(t+m) * a + m/(m+t) * c$$

where,
t represents the total number of ratings received by the book
m represents the minimum number of total ratings considered to be included
a represents the average rating of the book and,
c represents the mean rating of all the books.

User-Item Collaborative Filtering Recommendation:

Collaborative Filtering Recommendation System works by considering user ratings and finds cosine similarities in ratings by several users to recommend books. To implement this, we took only those books' data that have at least 50 ratings in all.

Correlation Based Recommendation:

For this model, we have created the correlation matrix considering only those books which have total ratings of more than 50. Then a user-book rating matrix is created. For the input book using the correlation matrix, top books are recommended.

Nearest Neighbour Based Recommendation:

To train the Nearest Neighbours model, we have created a compressed sparse row matrix taking ratings of each Book by each User individually. This matrix is used to train the Nearest Neighbours model and then to find n nearest neighbors using the cosine similarity metric.

Content Based Recommendation:

This system recommends books by calculating similarities in Book Titles. For this, TF-IDF feature vectors were created for unigrams and bigrams of Book-Titles; only those books' data has been considered which are having at least 80 ratings.

Hybrid Approach (Collaborative+Content) Recommendation:

A hybrid recommendation system was built using the combination of both content-based filtering and collaborative filtering systems. A percentile score is given to the results obtained from both content and collaborative filtering models and is combined to recommend top n books.

- **Providing Recommendations**

  After learning user preferences the system provides recommendations. Collaborative filtering is an approach to make recommendations on different items for users by collecting a bunch of information about their preferences. As it was mentioned before the idea of a collaborative filtering is that if two users have same preferences on a particular item then most likely that they will have same opinions on other items rather than with some other random user.

4. Conclusion:

Finally, we derive conclusion based on results shown through recommendation lists we obtained is that:

In EDA, the Top-10 most rated books were essentially novels. Books like The Lovely Bone and The Secret Life of Bees were very well perceived.

Majority of the readers were of the age bracket 20-35 and most of them came from North American and European countries namely USA, Canada, UK, Germany and Spain.

If we look at the ratings distribution, most of the books have high ratings with maximum books being rated 8. Ratings below 5 are few in number.

Author with the most books was Agatha Christie, William Shakespeare and Stephen King.

For modelling, it was observed that for model based collaborative filtering SVD technique worked way better than NMF with lower Mean Absolute Error (MAE) .

Amongst the memory based approach, item-item CF performed better than user   item CF because of lower computation.

References-

1. MachineLearningMastery
2. GeeksforGeeks
3. Analytics Vidhya