Data driven Book Recommender System: An Example of Simple Un-supervised Learning

-By Sayan Das

JUNE 1, 2023

Contents

Introduction

In today's digital age, the abundance of books and online platforms offering a wide range of reading materials can often leave readers feeling overwhelmed and uncertain about which books to choose. It is in this context that the development of intelligent book recommender systems becomes increasingly valuable, guiding readers through the vast literary landscape and helping them discover personalized reading recommendations.

The aim of this project is to design and build an advanced book recommender system that leverages the power of artificial intelligence and un-supervised machine learning algorithms to provide accurate, tailored recommendations to individual readers. By analyzing vast amounts of data, such as book metadata, user preferences, ratings, and reading habits, our system will strive to understand readers' unique tastes and deliver book suggestions that align with their interests..

Our book recommender system also holds promise for libraries, bookstores, and online platforms. By integrating our system into their services, these entities can enhance customer satisfaction, increase engagement, and improve overall user experience by providing accurate and relevant book recommendations tailored to each user's tastes.

The Datasets

For the purpose of building up the recommender system we have used 3 datasets available in Kaggle website. The link to the dataset is given below –

https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset

This dataset contains 2 datasets

1. Books dataset – contains
   I. ISBN – unique identification number assigned to the books
   II. Book title – title of the books
   III. author name – author of the books
   IV. year of publication – year of the book publication
   V. publisher name – name of the publisher of the book
   VI. book image url's – url to the images of the book cover (there are 3 urls for each books, depending of the image size)

This books dataset contains about 271360 books with unique ISBN

2. Rating dataset – contains
   I. user-id - unique identifying number assigned to the user
   II. ISBN - unique identification number assigned to the books
   III. Ratings – rating given by the user to the book

This dataset have ratings given by 105283 unique users.

We will be jointly analysing this 2 datasets to build the recommendation system But before we go into analysis lets take a look in to the different methods of data-driven recommendations.

Different Types of Recommendations

Based on the methodology of suggesting the items there are different kind of recommendation algorithm that are available. Each of the algorithm needs different kind of requirement of the dataset. Based on the data available to us we are going to apply

1. Popularity Based Recommendation
2. Collaborative filtering Based Recommendation

Let us get a brief idea about these two kinds of recommendation algorithm.

Popularity Based Recommendation

This kind of recommendation one tries to find out the top n item that are most popular. For this kind of recommendation, we need to calculate some sort of popularity measures for each of the items, this popularity score should be such that, the popularity score should be reflected on the score. That is, the items with more popularity should have higher values of the popularity score. After calculating the popularity score we choose the top n items with highest popularity score and recommend to the users.

Collaborative Filtering Based Recommendation

In the case of collaborative based filtering, we try to find out which items are most likely, according which kind of users uses that item. Suppose we have two users, user 1 and user 2. The user 1 used item 1, item 5, item 9 and item 7 in the past, similarly user 2 used item 2, item 5, item 9 and item 7. Since most of the items that user 1 and user 2 are common (item 5, item 9, item 7), we can have the idea that user 1 and user 2 has similar preference of items. Now we can suggest item 2, that is used by user 2 only, to user 1 with the hope that user 1 may like it or in need of it. Here our entire job is to find items used by similar users.

We have applied the two types of recommendation in the dataset.

Data Exploration

After exploring the datasets, we had found out the followings –

1. The dataset is based on ratings of 105283 unique readers giving ratings to 271360 books with unique ISBN
2. Only the books dataset have 2 missing value in the Book Author column, 2 missing value in the publishers columns and 3 missing values in the Large image url column. We have removed these missing observations
3. Books datasets have several duplicated books, that is books having same title and different ISBN number. We have removed those duplicated items.

Popularity based recommendations

In this kind of recommendation techniques our target is to score the books so that the score reflects the popularity of that books. Now the popularity is reflected by the rating of readers on the book. So, it’s a good idea to build the popularity score based on the ratings.

As a start we may take the average ratings, over the different users, of the books and call it as the popularity score. But in this way, we will face some issue. Consider those books that are read by a very few people. If we average the user ratings of such a book, the result may be biased to that few readers opinion. It kind of the biasedness is not desirable for a good recommender.

To remove this issue, we can consider the number of votes for a book. We can only consider those books that got at least 100 votes, that is at least 100 people has rated that book. Now among these selected books we can develop a formula for the popularity score.

For this project we have used the following formula for the popularity score.

$$Popularity\ Score = \frac{vR}{(v+m)} + \frac{mC}{(v+m)} \quad \ldots (1)$$

Where,

v = number of vote of the book

R = average rating of the book

m = minimum number of votes required to be listed in the list

C = mean rating across the whole report

Step for recommendation

1. Set a value for m. here we have taken m=100
2. Count the numbers of votes or the number of people who have rated, for each of the book
3. Filter out those books that has the vote(v) count at least m. After filtering in this way we have only 594 books.
4. Calculate the average ratings(R) for each of these selected books
5. Calculate the mean vote across all the books(C)
6. Calculated the popularity score using the formula (1)
7. Sort the selected books according to the descending order of the popularity score
8. Recommend the top-n books in the formed list

Collaborative filtering-based recommendations

In this kind of recommendation procedure, we have to build up a model, based on the similarity between the books, so that given a book the model will return a list of n most similar books to the given one.

Here the main issue is to calculate the similarity between the books. In this kind of recommendation, we will calculate the similarity between the books according the similar readers reading the books. Suppose for example, say we have two readers r1 and r2, reading 5 books b1- b5 and r1 gives a high rating to the books b2 and b4 where as r2 gives a high rating to b1, b3, b5. Now b2 and b4 are kind of similar in the sense r1 likes them and also b1, b3, b5 are kind of similar in the sense r2 likes them.

Now say there are another new user r3 who have read b3 then we can have the idea that since r3 read b3, he has a preference like r2. So, we can recommend b1 and b5, the other books read by r2, to r3.

The basic idea is shown above. But here we will consider the preference of not only two readers but a lot many readers.

Step for recommendation

1. In the first step, we will filter out those readers who have rated at least 100 books (capturing knowledgeable readers) and those books that are rated by at least 100 readers (capturing book popular among the users). After the filtering we have got 1433 users rating 594 book
2. In the next step we will for a matrix whose rows represented books, columns represents readers and the values are the ratings. The matrix will be like the following-

	Reader 1	Reader 2	…	Reader k
Book 1	R_{11}	R_{12}	…	R_{1k}
Book 2	R_{21}	R_{22}	…	R_{2k}
…	…	…		…
Book n	R_{n1}	R_{n2}	…	R_{nk}

   Observe that here every book is represented by a row vector of order k, denoting the rating for the book given by k users. Here k = 1433 and n = 594.

3. We will find the pairwise similarity score of the books represented by the row vectors in k dimension. Here we will use the cosine similarity as the similarity score. The formula of similarity between two vectors a and b is given by

$$cosine\ similarity\ score\ = \frac{a.b}{|a||b|}$$

   The value lies between -1 and 1. Higher the value of the measure is, higher the similarity between a and b. Here we obtain the similarity between the book in the form of a nxn(that means here 594x594) symmetric matrix.

4. Given a name of the book find out the row or column that corresponds the given book, extract the row or column. This represents the similarity scores of the given books to all other selected books. Sort the values in the descending order. In this way we will find the most similar books to the given book at the top.
5. Select n books from the top and recommend it.

Conclusion

In conclusion, the integration of popularity-based and collaborative filtering-based approaches in our book recommendation project has yielded promising results. By leveraging the collective wisdom of the user community, popularity-based recommendations offer a reliable starting point for users to discover widely acclaimed and popular books. This method ensures that users are exposed to titles that have received high overall ratings and widespread recognition, increasing the likelihood of finding engaging reads.

On the other hand, collaborative filtering-based recommendations have provided a personalized touch to the recommendation process. By analyzing individual user preferences and employing similarity-based algorithms, this approach tailors suggestions to align with users' specific tastes and reading habits. Collaborative filtering accounts for the unique characteristics and preferences of each user, enhancing the relevance and enjoyment of the recommended books.

By combining the strengths of both popularity-based and collaborative filtering-based methods, our book recommendation system has demonstrated its ability to cater to a wide range of users. The popularity-based approach ensures that users have access to popular and acclaimed titles, while collaborative filtering personalizes the recommendations based on individual preferences.

We have build a simple flask app to recommend books to the users. The links to the codes are given below.

References

Here we have the GitHub link for the project and link for the article that we consulted for he project-

1. Dataset: https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset
2. Article used: https://medium.com/the-owl/recommender-systems-f62ad843f70c