Book Recommendation System

By – Ranajay Biswas

Data Science Trainee,

Almabetter, Bangalore

Abstract:

A recommendation system broadly recommends items to the user best suited to their tastes and traits. It uses the user's previous data and other user's data to give new recommendations.

Book Recommendation systems are popular recommendations system as most people have a very limited time that they spend on trying out and reading new books. So, when they visit an online bookstore or just simply search on the internet about some book, it becomes important to utilize this opportunity to make recommendations that are similar to what they would like.

Also it is important to consider books that are worth reading, meaning books that are popular for being good among other readers.

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.

Data Summary & Attributes:

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.

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 O.

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.

Approach:

Our approach to solve this problem is going to be -

- Understanding the dataset, different rows and columns.
- During the EDA, we will try to find popular books and authors, where most of our readers come from. By calculating the number of votes and average ratings, we will find popular books in the data. Statistical methods and Visualizations are going to be very helpful in this EDA process.
- In the pre-processing step, we shall filter the most important features, make necessary transformations, create or omit features as needed. Depending on the features we choose, we find best approaches for text pre-processing.
- Clustering techniques will be used to identify different groups that books can belong to, based on rating and popularity.
- There is a choice to be made when it comes which recommendation system to use, as the data have both user-interaction related features and also content-related features. So, we try both recommender techniques and see how it goes.
- Then we conclude the project with an overview and discussion about the observations that we make, about the performance of our models and how this project can prove to be useful from a business standpoint.

Steps Involved:

- ## Loading & Understanding the Data

  There were 3 csv files containing the books data, ratings and users data. Those were loaded in our work environment as a data frame using pandas. We checked top 5 rows and bottom 5 rows to get an initial idea about the data.

- ## Exploratory Data Analysis

  Exploratory Data Analysis (EDA) is an approach to analyze the data using visual techniques. It is used to discover trends, patterns, or to check assumptions with the help of statistical summary and graphical representations.

  - The amount of null values are almost negligible in the books data.
  - No missing values in the ratings data.
  - For the users' data, we see that Age column has many null values. Almost 40 percent data is missing for this column.
  - Checking for duplicates, we found no duplicates in the data.
  - During the EDA, we used different plots and charts to visualize the patterns and trends that are present in the data.
  - Using histograms and boxplots, we found the distribution of users' age. Most of the readers are below 45 years of age. We can also see that there are many nonsensical values like all the values above 85 or 90. This data was not collected correctly.
  - With a lineplot, we visualized the number of book published each year.
  - Using barplots, we visualized the most popular authors, the publishers that have published most books and the top countries where most of the users come from etc.
  - We used countplots to see the top rated books and most rated books.
  - During Clustering Analysis, we used distplot, boxplots and barplots for visualizing different patterns in those clusters.

- ## Pre-Processing & Feature Engineering:

Feature Creation:

Doing groupby and taking a mean of the ratings, we found the average ratings of each book and stored the values by creating a column avg ratings**.** We counted the occurrence of rating and calculated how many times a book has been rated. We stored these values in num rated column.

Feature Selection:

For doing the Content based and Collaborative filtering, we selected **Book-Title, Book-Author, Average Ratings, Number of ratings** columns.

Feature Transformation: We used lowercasing, English stopwords removal and **TFIDF** vectorization on top of the Book-Title and Book-Author columns to get their vector forms that can be used for modelling.

TF-IDF stands for Term Frequency Inverse Document Frequency of records. It can be defined as the calculation of how relevant a word in a series or corpus is to a text. The meaning increases proportionally to the number of times in the text a word appears but is compensated by the word frequency in the corpus (data-set).

- **Modelling:**

Clustering:

- **K-Means Clustering :**

K-Means is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The best value for K in our experiment was **4** and the Silhouette score was 0.65

Silhouette Score varies from -1 to +1. A positive Silhouette Score means clusters are well separated.

Recommender Systems:

- **Technique 1: Content Based Filtering**

To create a content based recommender system, we found the **Cosine similarities** between the data points. Based on the book that user gives as input, our system calculates the similarities and recommends top 5 most similar books.

Content-based filtering uses item features to recommend other items similar to what the user likes, based on their previous actions or explicit feedback. Content-based filtering uses similarities in products, services, or content features, as well as information accumulated about the user to make recommendations.

We selected **Book-Title, Book-Author** features, then made TFIDF transformation on these columns. Finally, we applied Cosine Similarities to calculate the similarities or distances between each book. Also, we only selected books that have been rated at least 50 times for this method of modelling, since we want to recommend books that have some popularities.

- **Technique 2: Collaborative Filtering(CF)**

Collaborative filtering technique is a **recommendation system** that creates a prediction based on a user's previous behaviours.

We selected Avg ratings and num ratings columns to create a user-interaction matrix. We selected only those users who have rated at least 200 books and only those books that have at least 50 votes. That way our collaborative recommender system will be intelligent.

- **Technique 1: Hybrid Recommendation System**

A hybrid recommendation system is a special type of recommendation system which can be considered as the combination of the content and collaborative filtering method. Combining collaborative and content-based filtering together may help in overcoming the shortcoming we are facing at using them separately and also can be more effective in some cases.

Hybrid recommender system approaches can be implemented in various ways like by using content and collaborative-based methods to generate predictions separately and then combining the prediction or we can just add the capabilities of collaborative-based methods to a content-based approach.

In this approach, we took the predictions from both of our previous recommender systems and found the common recommendations. If somehow, no common recommendations found, then it returns the recommendation from the collaborative system.

- # Conclusion:
  - We performed EDA on the 3 different datasets. Made visualizations and found insights from the data. Performed different pre-processing techniques to prepare the data for send it to the recommendation systems.
  - Implemented Content based, Collaborative and Hybrid Recommendation systems that are being able to efficiently recommend similar books.
- In the EDA, we found most of the readers are from USA, Canada, UK, Germany and Spain. We found the top authors and publishers. Discovered the top rated books and most number of rated books.
- Many records were missing for users' age. Many of the existing information was invalid for this particular feature as well.
- We noticed that for many books, very small number of ratings were present which does not really give us an idea if those books are actually good or not. Also not all the users should be considered when building a recommendation system. We want genuine, credible and unbiased users. So, we declared thresholds in terms of selecting users and books that will be considered for recommendations.

Suggestions:

- More focus can be given to the users who come from the top 10 countries. These are the users that read and rate the books. So, these are more important users.

- The data regarding the users' age needs to be extracted correctly. Then it can help us with doing many analysis and also to recommend similar content to similar age group people.
- Both Collaborative and content based filtering are performing efficiently. But if we want to be surer about the recommendations that are being made, then the Hybrid approach can be used.
- We found the best results for user and book-ratings threshold that is in the final notebook. But as the data grows, we can tweak with those thresholds to consider and recommend newer books as well.

With that, we have reached the end of this project. We hope the analysis that we performed helped to understand people's reading habits. We found areas where more care should be given and explained how the recommendation systems can be used efficiently to provide books recommendations to readers.

References-

1. AlmaBetter
2. Kaggle
3. MachineLearningMastery
4. GeeksforGeeks
5. Analytics Vidhya