Book Recommendation System

Mohammed Javeed
Data science trainee,
Alma Better, Bangalore

Abstract:

Recommendation systems are used in hundreds of different services - everywhere from online shopping to music to movies, Libraries. For instance, the online retailer Amazon had a heavy hand in developing collaborative filtering algorithms that recommend items to users. Music services like Pandora identify up to 450 uniquely identifying characteristics of songs to find music similar to that of their users' preferences. Other music streaming services, such as Spotify, heavily rely upon the music selections of similar users to make weekly song recommendations and personalized radio stations. Netflix, a popular television and movie streaming service, uses these systems to recommend movies that viewers may enjoy. We can see how recommendation systems have a surprisingly large impact on the materials consumers engage with over the course of their daily lives.

For this capstone project we have dataset of books in three files Users, Ratings and Books, we analyzed three dataset systems that predict how users will rate specific books. Our system that we created makes these predictions based Collaborative filtering by using KNN model. From that we accurately predict top five books on users' reactions to books, we've integrated several strategies in the field of recommendation systems.

Table of Contents

- Problem Statement
- Introduction to a Recommendation system
- Types of Recommendation system
- Book Recommendation System
  - Content-Based Filtering
  - Collaborative-based filtering
  - Hybrid filtering
- Hands-on Recommendation system
  - Dataset Description
  - Preprocess data
  - Perform EDA
  - Predictions

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.

Content 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

There are three major approaches for recommendation systems: (i) content-based, (ii) collaborative, and (iii) hybrid. Broadly, recommendation systems that implement a content-based (CB) approach recommend items to a user that are similar to the ones the user preferred in the past. On the other hand, recommendation systems that implement collaborative filtering (CF) predict users' preferences by analyzing relationships between users and interdependencies among items; from these, they extrapolate new associations. Finally, hybrid approaches meld content- based and collaborative approaches, which have complementary strengths and weaknesses, thus producing stronger results.

Method

Our first task was to model the books in our datasets. We chose two different approaches one is to find top most 50 books in our dataset shows and second is to display users on the basis of collaborative filtering top most 5 similar books on the basis of cosine distance (vector). For both approaches, we preprocessed the text of all descriptions and reviews we had for a particular book to remove or cleaning dataset

The final piece of the collaborative-based system that we implemented was feature selection based on the set of books that a user had rated in our dataset. Though we could have used every feature in whichever vector(s) our classifiers were considering, there were so many features in any given book vector and so little training data that it made more sense to apply the heuristic of choosing the top 5 features that, across the books a user has read in our training dataset.

3. Objective

- The general objective of the project is to develop a web application to search top 50 books and By users on the basis of reading shows the top 5 books on the basis of collaborative filtering

- To provide platform to view the books on the webpage.

- To purchase books online.
- To recommend a book to a user/buyer

- The Book Recommendation System aims to provide the best suggestion to the user by analyzing the buyer's interest. The quality and the content are taken into consideration by employing popularity based, association rule mining and collaborative filtering.

4. Project Overview

The booming technology of the modern world has given rise to the enormous book websites. This makers the buyers to choose the best books to read as books play a vital role in many people's life. The various kinds of books come into existence on day to day basis. So in order to eliminate this critical situation the recommendation system has been introduced in which the suggestion on the various books can be provided based on the analysis of the buyer's interest. The Book Recommendation System is an intelligent algorithm which reduces the overhead of the people. This provides benefit to both the seller and the consumer creating the win-win situation. The E-commerce site to network security, all demands the need for the recommended system to increase their revenue rate. The content filtering, association rule mining and collaborative filtering are the various decision making techniques employed in the recommendation system as it helps buyers by the strong recommendations as there are various books, buyer's sometimes cannot find the item they search for. The Book Recommendation System is widely implemented using search engines comprising of data sets.

5. Types of Recommendation System

A recommendation system is usually built using 3 techniques which are content-based filtering, collaborative filtering, and a combination of both.

1) Content-Based Filtering
2) Collaborative-based Filtering
3) Hybrid Filtering Method

1) Content-Based Filtering

Overview CB recommendation strategies consist of a few high-level steps: First, a quantitative model is generated to represent each of the products. The model should represent something about the actual content of the product. Next, using the model and feedback (usually in the form of user ratings or reviews), the algorithm builds a user profile. The profile is essentially a classifier that learns how to accurately predict a user's ratings for the modeled items. Finally, each of the modeled products a user has not yet tried is presented to the user profile to determine an approximate rating, and the highest-rated objects become the most highly recommended items for that user.

The algorithm recommends a product that is similar to those which used as watched. In

simple words, In this algorithm, we try to find finding item look alike. For example, a person likes to watch Sachin Tendulkar shots, so he may like watching Ricky Ponting shots too because the two videos have similar tags and similar categories.

Only it looks similar between the content and does not focus more on the person who is watching this. Only it recommends the product which has the highest score based on past preferences.

2) Collaborative-based Filtering

While CB systems recommend items with similar features to users (e.g. books with similar contents or writing styles), CF systems predict user preferences by analyzing past relationships between users and interdependencies among items [8]. More specifically, CF works under the belief that if a person A has the same opinion as a person B on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person. CF systems can be categorized into either non-probabilistic or probabilistic 4 algorithms. There are a large number of CF algorithms.

We've explored two: UV-decomposition and k nearest neighbor.

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.

3) Hybrid Filtering Method

It is basically a combination of both the above methods. It is a too complex model which recommends product based on your history as well based on similar users like you.

There are some organizations that use this method like Facebook which shows news which is important for you and for others also in your network and the same is used by Linkedin too.

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 ebooks like google play books, open library, good Read's, etc.

Practical Implementation of Recommendation System

Let's make our hands dirty while trying to implement a Book recommendation system using collaborative filtering.

Dataset Description

We have 3 files in our dataset.

- Books – first are about books which contain all the information related to books like an author, title, publication year, etc.
- Users – The second file contains registered user's information like user id, location.
- Ratings – Ratings contain information like which user has given how much rating to which book.

So based on all these three files we can build a powerful collaborative filtering model. Let's get started.

Load Data

Let us start while importing libraries and load datasets. While loading the file we have some problems like.

So while loading data we have to handle these exceptions and after running the below code you will get some warning and it will show which lines have an error that we have skipped while loading.

Preprocessing Data

Now in the books file, we have some extra columns which are not required for our task like image URLs. And we will rename the require 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.

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.

So this is achieved using Matrix Factorization, we will create one matrix where columns will be users and indexes will be books and value will be rating. Like we have to create a Pivot table.

A big flaw with a problem statement in the dataset

If we take all the books and all the users for modeling, Don't you think will it create a problem? 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

Exploratory Data Analysis

The dataset is reliable and can consider as a large dataset. We have 271360 books data and total registered users on the website are approximately 278858 and they have given near about 1149780 rating. Hence we can say that the dataset we have is nice and reliable.

After that we create new column Avg rating and Number of rating on this basis of columns we clearly shows the top 50 books this is known as popularity based recommendation system

After this we go to use Collaborative filtering to find top 5 similarity books on the basis of users reading.

So let's get with analysis and prepare the dataset as we discussed for modeling. Let us see how many users have given ratings and extract those users who have given more than 200 ratings.

Step-1) Extract users and ratings of more than 200

When you run the above code we can see only 105283 peoples have given a rating among 278858. Now we will extract the user ids who have given more than 200 ratings and when we will have user ids we will extract the ratings of only this user id from the rating data frame and we get 526356 ratings above 200.

Step-2) Merge ratings with books

So 900 users are there who have given 5.2 lakh rating and this we want. Now we will merge ratings with books on basis of ISBN so that we will get the rating of each user on each book id and the user who has not rated that book id the value will be zero.

step-3) Extract books that have received more than 50 ratings.

Now dataframe size has decreased and we have 4.8 lakh because when we merge the dataframe, all the book id-data we were not having. Now we will count the rating of each book so we will group data based on title and aggregate based on rating.

we have to drop duplicate values because if the same user has rated the same book multiple times so it will create a problem. Finally, we have a dataset with that user who has rated more than 200 books and books that received more than 50 ratings. the shape of the final dataframe is 59850 , rows and 9 columns.

Step-4) Create Pivot Table

As we discussed above we will create a pivot table where columns will be user ids, the index will be book title and the value is ratings. And the user id who has not rated any book will have value as NAN so impute it with zero.

We can see the more than 11 users have removed out because their ratings were on those books which do not receive more than 50 ratings so they are moved out of the picture.

Finally books remaining 742 and users is 888

Because of so many zeroes if we calculate distances then so many time will take so we use sparse matrix to simplify our distance calculation for k nearest neighbor algorithm

Modeling

K Nearest Neighbors (kNN) the second collaborative filtering algorithm we implemented is a memory-based algorithm known as kNN. Nearest neighbor algorithms are a family of popular non-probabilistic CF techniques. This technique computes the similarity between either users (user-to-user based CFs) or items (item-to-item based CFs) that are represented by columns or rows in a user-item matrix with similarity metrics such as cosine, adjusted cosine, or Euclidean. This algorithm then predicts unknown ratings based on known ratings of similar users or items. The similarity values "provide the means to give more or less importance to these neighbors in the prediction".

We have prepared our dataset for modeling. We will use the nearest neighbors algorithm which is the same as K nearest which is used for clustering based on Euclidian distance.

But here in the pivot table, we have lots of zero values and on clustering, this computing power will increase to calculate the distance of zero values so we will convert the pivot table to the sparse matrix and then feed it to the model.

Now we will train the nearest neighbor's algorithm. Here we need to specify an algorithm which is brute means find the distance of every point to every other point.

Let's make a prediction and see whether it is suggesting books or not. We will find the nearest neighbors to the input book id and

after that, we will print the top 5 books which are closer to those books. It will provide us distance and book id at that distance. Let us pass Naked which is at 358 indexes.

Let us print all the suggested books.

```
358
Recommendations for Naked:
1: Me Talk Pretty One Day, with distance of 0.6545588833986331:
2: 1984, with distance of 0.8039283491773567:
3: Big Trouble, with distance of 0.806687213041357:
4: Bastard Out of Carolina, with distance of 0.8135603168032868:
5: Turtle Moon, with distance of 0.8207052172096709:
```

Hence, we have successfully built a book recommendation system.

6. Conclusion:

That's it! We reached the end of our exercise. Starting with loading the data so far we have done EDA, null values treatment, creating new columns, Use popularity based filtering and collaborative filtering and then model building of KNN by using Cosine similarity.
In all of these models our accuracy to give us better recommendations.
All of our systems– purely purely collaborative-filtering, and popularity based. Looking back on the project, one thing that we might have chosen to do differently in retrospect would have been to spend more time searching for a dataset of ratings with a higher rating variance per user. Had we been able to find such a dataset, our implementations of algorithms would have been tested on data that would have been more representative of what a typical commercial recommendation system could access in creating its predictions. However, given the data that was available to us, as well as the results our various approaches produced, our systems were largely successful, providing insight into how the different systems we regularly use work and the varying algorithms that make that possible.

References-

1. Analytics Vidhya, Medium
2. Kaggle
3. E. A. Laksana, “Collaborative Filtering dan Aplikasinya,” J. Ilm. Teknol. Inf. Terap., vol. 1, no. 1, pp. 36–40, 2014.