**Book Recommendation System**

**Sujit Musale**

**Data science trainees,**

**AlmaBetter, Bangalore**

**Abstract:**

A recommendation system is one of the top applications of data science. Every consumer Internet company requires a recommendation system like Netflix, Youtube, a news feed, etc. What you want to show out of a huge range of items is a 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.

***Keywords: machine learning, regression rating, Books, Users***

**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**

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.

**3. Approach:**

* **Data Loading and cleaning**

Firstly, after loading the dataset we preprocessed the raw data to have better understanding of the features, and make the data high quality. We performed the following steps in order to clean our dataset:

* + **Null values**: User data set having null value in age are treated by doing normal distribution of data. Books data set having null value dropped as count is low bur rating data set don’t have any null value..
  + **Duplicated values**: There is no any duplicate value in all three data set So just checked.
  + **Improper format**: Most of the location name from user data set are having spell mistake or wrong data so replaced with correct data.
  + **Data merging:** Data merging done as per required data like rating data merged with book data and then this merged data again merged with User data in this process we found most of the rating data is out of our scope
  + **Feature engineering**: we drop the image URL data column’s as we are not having any use of that.
* **Exploratory Data Analysis**

We performed EDA on individual data set like user, books and rating and come up with some important conclusion from analysis:

* + Sum of the users are having age more than 100 and up to 250 year so that we treated as outliers.
  + From user we also find various age group and there respective percentage with help of piechart.
  + We checked rating count for 0 to 10.
  + We find the top books and authors who are having large count of users.

Also performed EDA with merged data set.

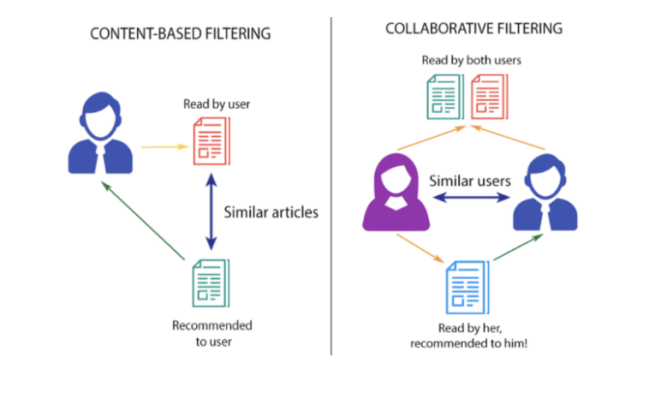
* + We count number of rating to the book authors..
  + Top publisher and total book title count that they have.

**4. Fitting Models**

Below are the some model that we used for Filtering.

1. **Collaborative filtering - Memory Based.(Matrix Factorization)**
2. **Collaborative filtering - Model Based. (KNN algorithm)**
3. **Popularity Based Filtering(Weighted average)**

* **Type of filtering**

****

**1. Content based filtering-** This method uses only information about the description and attributes of the items users has previously consumed to model user's preferences. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past (or is examining in the present). In particular, various candidate items are compared with items previously rated by the user and the best-matching items are recommended.

Its use TF-IDF vectorizer techniques.

Term Frequency (TF) - How many times a particular word appears in a single document?

Inverse Document Frequency (IDF) - It is calculated by taking the log of {number of docs in your corpus divided by the number of docs in which this term appears}. This takes care of words which are present rarely across the corpus.

**2. Collaborative based filtering**

This method makes automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). The underlying assumption of the collaborative filtering approach is that if a person A has the same opinion as a person B on a set of items, A is more likely to have B's opinion for a given item than that of a randomly chosen person.

Collaborative Filtering (CF) has two main implementation strategies:

**Memory-based**

This approach uses the memory of previous users interactions to compute users similarities based on items they've interacted (user-based approach) or compute items similarities based on the users that have interacted with them (item-based approach).

A typical example of this approach is User Neighbourhood-based CF, in which the top-N similar users (usually computed using Pearson correlation) for a user are selected and used to recommend items those similar users liked, but the current user have not interacted yet. This approach is very simple to implement, but usually do not scale well for many users.

**Model-based**

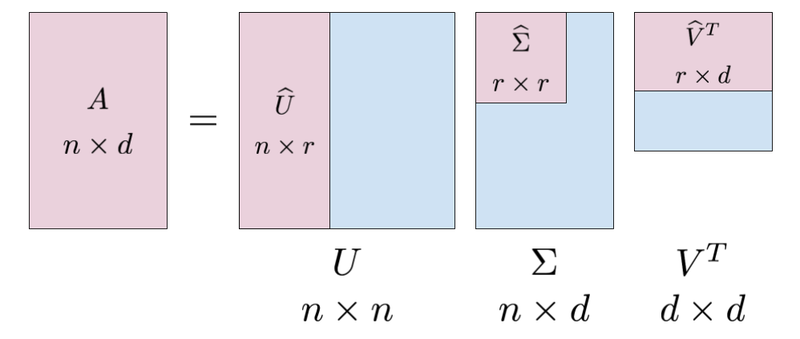
In this approach, models are developed using different machine learning algorithms to recommend items to users. There are many model-based CF algorithms, like Neural Networks, Bayesian Networks, Clustering Techniques, and Latent Factor Models such as Singular Value Decomposition (SVD) and Probabilistic Latent Semantic Analysis.

* **Model Building**

**Matrix factorization**

Latent factor models compress user-item matrix into a low-dimensional representation in terms of latent factors. One advantage of using this approach is that instead of having a high dimensional matrix containing abundant number of missing values we will be dealing with a much smaller matrix in lower-dimensional space.

A reduced presentation could be utilized for either user-based or item-based neighborhood algorithms. There are several advantages with this paradigm. It handles the sparsity of the original matrix better than memory based ones. Also comparing similarity on the resulting matrix is much more scalable especially in dealing with large sparse datasets.



U is an n×n unitary matrix

Σ is a diagonal n×d matrix with non-negative real numbers on the diagonal.

V is an d×d unitary matrix and VT is the transpose of V.

* **KNN Filtering**

Which is also called the memory-based approach. utilizes the entire user-item database to generate predictions directly, i.e., there is no model building.

This approach includes both

• User-based methods

• Item-based methods

* **Weighted avg rating method:**

We use below formula to calculate filtering Weighted average rating method Using Weightedaverage for each Book’s Average Rating.

**W = Rv + Cm/(v + m)**

where

W= Weighted Rating

R = Average of the Books rating

v = No of people who have rated the books (number of votes)

m = minimum no of votes to be listed

C = the mean rating across all the books

**5. Conclusion:**

The initial step, of our project was Data preprocessing of the three datasets-books, users and ratings ,wherein we removed duplicates and imputed the missing values & invalid entries with appropriate values, corrected spellings. Then, we performed Exploratory Data Analysis to find out the countries with maximum users, popular books, popular authors and popular publishing companies. We also analyzed the rating distribution, age distribution of users and the popular books amongst various age groups. Then, we used Popularity-based approach, Collaborative filtering approach to built different types of recommendation models. We evaluated the performance of Singular Value Decomposition based recommender and obtained a Global Recall@5 of 30 % and Recall@10 of 41%