Capstone Project-4

BOOK RECOMMENDATION SYSTEM

**Abstract**

When we want to read a new book, we generally ask our friends or classmates or may search all the books available in a library (as if we can). After all, asking and searching, we may still not find any book of our preference as not everyone has the same interests. For such situations, we need a system which takes our choices into consideration and suggests to us some good books.

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.

# 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.

## ****Dataset****

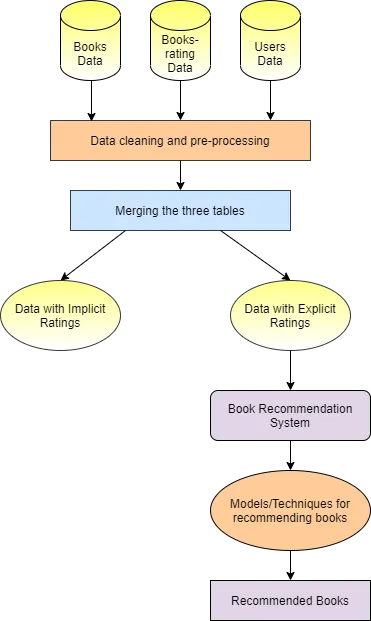
The dataset we have used in this work is the data comprises three tables: -

Books- It has 8 columns; ISBN, Book title, Book author, Year of publication, Publisher, and three columns for Book cover Image URLs representing three different versions (small, medium, and large).

Users- Contains the user’s information. It consists of 3 columns UserID, Location, and age.

Ratings- Contains the information on ratings of the books. It consists of 3 columns UserID, ISBN, and Book Rating.

The workflow of our project is:



**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 as 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.

# Ratings Table

Check for null values in the table.

Check for the 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.

# 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, three tuples have the name of the author of the book 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 number of the book.

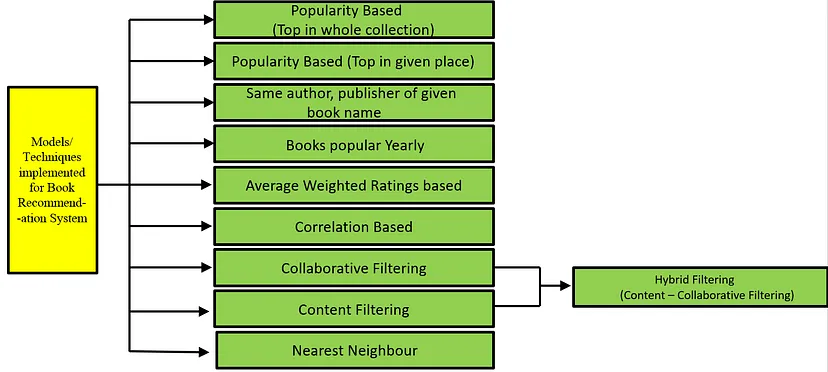
Convert the type of the years of publications feature to the integer.

By keeping the range of valid years as less than 2022 and obviously 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.

# Recommendation Models

We started building some basic recommendation systems and then implemented collaborative and content-based filtering methods as well.

****

## Popularity Based (Top 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.

## Popularity Based (Top In a given place)

We have filtered the dataset according to a given place (city, state, or country) and then sorted it 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 the 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.

## Books popular 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.

## Correlation Based

For this model, we have created the correlation matrix for which we needed to reduce the dataset (because of limited resources). So we have considered only those books which have total ratings of more than 50. Then from this data, we have created a user-book rating matrix. For the input book using the correlation matrix, top books are recommended.

## Nearest Neighbours Based

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.

## ****Collaborative Filtering (User-Item Filtering)****

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 (because of limited resources).

## Content-Based Filtering

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

## ****Hybrid Recommendation System****

We have built a hybrid recommendation system using 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.

# Conclusion

* 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-user CF** because of lower computation