Capstone Project Submission

Book recommendation system

Cohort Montreal

Team Member's Name, Email and Contribution:

**Name : **Kumar Mhaske
Email id : kumarmhaske1998@gmail.com

Contribution:

- Introduction to Data
- Data cleaning and pre processing
- Correlation
- Understanding the data
- Implementing Algorithms
- Popularity based recommendation
- Popular in the whole collection
- Popular at a given place
- Books by the same author, publisher of given book name
- Popular books yearly
- Recommendation using average weighted rating
- User item collaborative filtering recommendation
- Correlation based recommendation
- Nearest neighbors based recommendation
- Content based recommendation

Please paste the GitHub Repo link.

GitHub repo link:

https://github.com/KumarMhaske/Book-recommendation-system-

Drive Link :

https://drive.google.com/drive/folders/1Eh92wOYslPmElJwoukKxwiqglbiaJkuV?usp=share_link

Please write a short summary of your Capstone project and its components. Describe the problem statement, your approaches and your conclusions.

(200-400 words)

1. Data Cleaning and Pre-Processing

The dataset consists of three tables; Books, Users, and Ratings. Data from all three tables are cleaned and pre-processed separately as defined below briefly:

For Books Table:

- Drop all three Image URL features.
- Check for the number of null values in each column. There come 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, for three tuples name of the author of the book was 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 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 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.

For 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; 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.

For Ratings Table:

- Check for null values in the table.
- Check for 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.

2. Algorithms Implemented:

Popularity Based Recommendation:

- **Popular 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.

- **Popular at a Given Place**

The dataset was filtered according to a given place (city, state, or country) and then sorted 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 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.

- **Popular Books 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.

Recommendation using Average Weighted Rating

We have calculated the weighted score using the below formula for all the books and recommended the books with the highest score.

$$\text{score}= t/(t+m) * a + m/(m+t) * c$$

where,
t represents the total number of ratings received by the book
m represents the minimum number of total ratings considered to be included
a represents the average rating of the book and,
c represents the mean rating of all the books.

User-Item Collaborative Filtering Recommendation

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.

Correlation Based Recommendation

For this model, we have created the correlation matrix considering only those books which have total ratings of more than 50. Then a user-book rating matrix is created. For the input book using the correlation matrix, top books are recommended.

Nearest Neighbour Based Recommendation

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 neighbours using the cosine similarity metric.

Content Based Recommendation

This system recommends books by calculating similarities in Book Titles. For this, TF-IDF feature vectors were created for unigrams and bigrams of Book-Titles; only those books' data has been considered which are having at least 80 ratings.

Hybrid Approach (Collaborative + Content) Recommendation

A hybrid recommendation system was built using the combination of 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.

3. Libraries Used:

- ipython-notebook - Python Text Editor
- sklearn - Machine learning library
- seaborn, matplotlib - Visualization libraries
- numpy, scipy- number python library
- pandas - data handling library

: