Capstone Project- 4

BOOK RECOMMENDATION SYSTEM

Prerit Tyagi
AlmaBetter, Bangalore

ABSTRACT

Recommendation System (RS) is software that suggests similar items to a purchaser based on his/her earlier purchases or preferences. RS examines huge data of objects and compiles a list of those objects which would fulfill the requirements of the buyer. Nowadays most ecommerce companies are using Recommendation systems to lure buyers to purchase more by offering items that the buyer is likely to prefer. Book Recommendation System is being used by Amazon, Barnes and Noble, Flipkart, Goodreads, etc. to recommend books the customer would be tempted to buy as they are matched with his/her choices. The challenges they face are to filter, set a priority and give recommendations which are accurate. RS systems use Collaborative Filtering (CF) to generate lists of items similar to the buyer's preferences. Collaborative filtering is based on the assumption that if a user has rated two books then to a user who has read one of these books, the other book can be recommended (Collaboration). CF has difficulties in giving accurate recommendations due to problems of scalability, sparsity and cold start. Therefore this paper proposes a recommendation that uses Collaborative filtering with Jaccard Similarity (JS) to give more accurate recommendations. JS is based on an index calculated for a pair of books. It is a ratio of common users (users who have rated both the two books individually. Larger the number of common users higher will be the JS Index and hence better recommendations. Books with high JS index (more recommended) will appear on top of the recommended books list.

Keywords: Similarity index, filtering techniques, recommender system, Jaccard Similarity.

INTRODUCTION

Recommendation system filters information by predicting ratings or preferences of consumers for items that the consumer would like to use. It tries to recommend items to the consumer according to his/her needs and taste. RS mainly uses two methods to filter information - Content-based and Collaborative filtering. Content-based filtering involves recommending those items to a consumer which are similar in content to the items that have already been used by him/her. First, it makes a profile of the consumer, which consists of his/her taste. Taste is based on the type of books rated by the consumer. The system analyses the books that were liked by the consumer with the books he had not rated and looks for similarity. Out of these unrated books, the books with the maximum value of similarity index will be recommended to the consumer. Paul Resnick and Hal Varian were the ones who suggested Collaborative filtering algorithm in 1997. It became popular amid the various frameworks available at that time. A complete RS contains three main things: user resource, item resource and the recommendation algorithm. In the user model, the consumers'

interests are analysed, similarly, the item model analyses the items' features. Then, the characteristics of the consumer are matched with the item characteristics to estimate which items to recommend using the recommendation algorithm. The performance of this algorithm is what affects the performance of the whole system. In memory-based CF, the book ratings are directly used to assess unknown ratings for new books. This method can be subdivided into two ways: User-based approach and Item-based approach.

BACKGROUND

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.

DATA

The dataset is comprised of three CSV 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 case of several authors, only the first is provided. URLs linking to cover images
- 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.

METHODOLOGY

- **DATA PREPROCESSING:**

1. In books data pre-processing we handled missing values and duplicated values. Handled outliers in Year-of-Publication column. Corrected values in Book-Author, Book-Title and Publisher Column. Dropped image-url-s, image-url-m and image-url-l columns as they were not of much importance.
2. In user data pre-processing, we took the age group of 6 to 90. Extracted city, state and country features from Location column. Handled outliers, missing values and duplicated values in age columns.
3. In rating data pre-processing, we separated book rating into rating implicit and rating explicit. Handled outliers, missing values and duplicate values in book rating.

- **EXPLORATORY DATA ANALYSIS**

In statistics, exploratory data analysis is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task. Data visualization is the graphic representation of data. It involves producing images that communicate relationships among the represented data to viewers of the images. This communication is achieved through the use of a systematic mapping between

graphic marks and data values in the creation of the visualization. This mapping establishes how data values will be represented visually, determining how and to what extent the property of a graphic mark, such as size or color will change to reflect changes in value of datum.

- **FEATURE ENGINEERING**

1. **Handling Missing Values:** Values that are reported as missing may be due to a variety of factors. These lack of answers would be considered missing values. The researcher may leave the data or do data imputation to replace them. Suppose the number of cases of missing values is extremely small; then, an expert researcher may drop or omit those values from the analysis. But here in our case we had a lot of data points which were having missing values in different feature columns.

2. **Imputing Values:** Since, we have missing values in some feature variables, we need to impute them.
    - ✓ Year-Of-Publication
    - ✓ Book-Author
    - ✓ Publisher
    - ✓ Age

- **Year of Publication**: From the analysis part we get that the Year-Of_publication was wrongly mentioned for some of the rows. Diving deep into the Books_df we got to know that for these rows there was actually a column mismatch. Also, for Year-Of-Publication we observed that the year mentioned was beyond 2020 for some entries whereas the dataset was created in 2004. So, for the anomalous entries we first filled them with Nan values. Then we also observed that the Year-Of_publication graph was right skewed so we imputed the Nan values with the median. For Book-Author and Publisher missing data since there were a very few missing values we could search the internet and impute the values.

- **AGE:** An outlier is an observation of a data point that lies an abnormal distance from other values in a given population. It is an abnormal observation during the Data Analysis stage, that data point lies far away from other values. From the dataset we found that there were outliers in the Age column and the Age range exceeded beyond 100 which we all know is not sensible at all. Having a look at the distribution of our outliers and the missing number of values we came to the point that we cannot afford to just drop off these Nan and outliers because even these users have given ratings and we have User-Item interactions for these users. So, we concluded not to drop off these values, instead we first imputed the outliers with Nan values then we imputed all the Nan values with some sensible number. Here we took an interesting approach. Instead of simply imputing with a single value we grouped our user information based on country and took the median of the age then we imputed these values in place of NaNs. Also, we decided to impute with median because earlier we saw that the age distribution is positively skewed. Age value's below 5 and above 100 do not make much sense for our book rating case...hence replacing these by NaN. Next step was to impute these NaNs with median grouped on country values.

- **POPULARITY BASED FILTERING**

It is a type of recommendation system which works on the principle of popularity and or anything which is in trend. These systems check about the books which are in trend or are most popular among the users and directly recommend them.

Why Is this model relevant?

The answer to this is the Cold-Start Problem. Cold start is a potential problem in computer-based information systems which involves a degree of automated data modelling. Specifically, it concerns the issue that the system cannot draw any inferences for users or items about which it has not yet gathered sufficient information. The cold start problem is a well known and well researched problem for recommender systems. Recommender systems form a specific type of information filtering (IF) technique that attempts to present information items (e-commerce, films, music, books, news, images, web pages) that are likely of interest to the user. Typically, a recommender system compares the user's profile to some reference characteristics. These characteristics may be related to item characteristics (content-based filtering) or the user's social environment and past behavior (collaborative filtering). Depending on the system, the user can be associated with various kinds of interactions: ratings, bookmarks, purchases, likes, number of page visits etc.

A popularity based model does not suffer from cold start problems which means on day 1 of the business also it can recommend products on various different filters. There is no need for the user's historical data. The popularity index used for our books dataset was weighted rating.

WR = [(v * R)/(v + m)] + [(m * c)/(v + m)]

where,
v is the number of votes for the books;
m is the minimum votes required to be listed in the chart;
R is the average rating of the book;
and C is the mean vote across the whole report.

Using this popularity metric we can calculate the top books that could be recommended to a user.

	Book-Title	Total_No_Of_Users_Rated	Avg_Rating	Score
0	Harry Potter and the Goblet of Fire (Book 4)	137	9.262774	8.741835
1	Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))	313	8.939297	8.716469
2	Harry Potter and the Order of the Phoenix (Book 5)	206	9.033981	8.700403
3	To Kill a Mockingbird	214	8.943925	8.640679
4	Harry Potter and the Prisoner of Azkaban (Book 3)	133	9.082707	8.609690
5	The Return of the King (The Lord of the Rings, Part 3)	77	9.402597	8.596517
6	Harry Potter and the Prisoner of Azkaban (Book 3)	141	9.035461	8.595653
7	Harry Potter and the Sorcerer's Stone (Book 1)	119	8.983193	8.508791
8	Harry Potter and the Chamber of Secrets (Book 2)	189	8.783069	8.490549
9	Harry Potter and the Chamber of Secrets (Book 2)	126	8.920635	8.484783
10	The Two Towers (The Lord of the Rings, Part 2)	83	9.120482	8.470128
11	Harry Potter and the Goblet of Fire (Book 4)	110	8.954545	8.466143
12	The Fellowship of the Ring (The Lord of the Rings, Part 1)	131	8.839695	8.441584
13	The Hobbit : The Enchanting Prelude to The Lord of the Rings	161	8.739130	8.422706
14	Ender's Game (Ender Wiggins Saga (Paperback))	117	8.837607	8.409441
15	Tuesdays with Morrie: An Old Man, a Young Man, and Life's Greatest Lesson	200	8.615000	8.375412
16	Charlotte's Web (Trophy Newbery)	68	9.073529	8.372037
17	Dune (Remembering Tomorrow)	75	8.973333	8.353301
18	A Prayer for Owen Meany	181	8.607735	8.351465
19	Fahrenheit 451	164	8.628049	8.346969

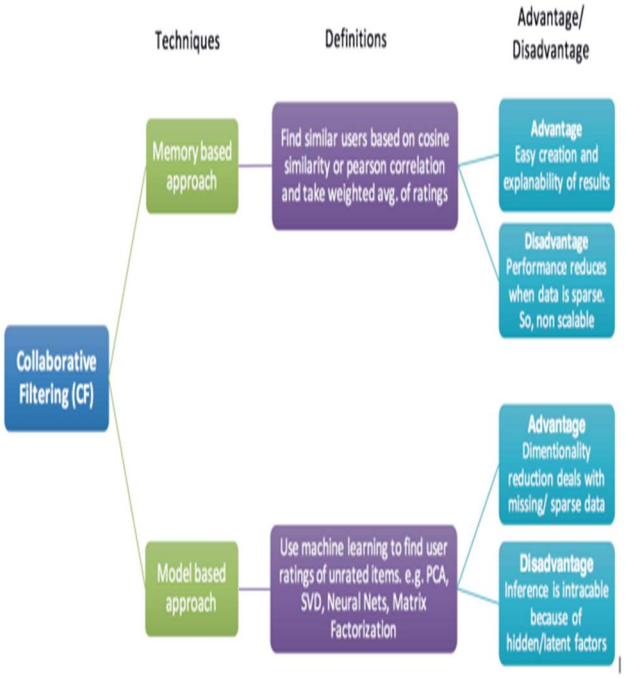
Hence it was observed that the Harry Potter series was the most popular book series. Other than that books like To Kill A Mockingbird, The Lord of the Rings and Dune were among the top books.

- **COLLABORATIVE FILTERING**

It is considered to be one of the very smart recommender systems that work on the similarity between different users and also items that are widely used as an e-commerce website and also online movie websites. It checks about the taste of similar users and makes recommendations.

The similarity is not restricted to the taste of the user, moreover there can be consideration of similarity between different items also. The system will give more efficient recommendations if we

have a large volume of information about users and items. There are various types of collaborative filtering techniques as mentioned in the diagram given below.



Types of collaborative filtering approaches

1. Model Based Approach:

Model-based recommendation systems involve building a model based on the dataset of ratings. In other words, we extract some information from the dataset, and use that as a "model" to make recommendations without having to use the complete dataset every time. This approach potentially offers the benefits of both speed and scalability.

Model based collaborative approaches only rely on user-item interactions information and assume a latent model supposed to explain these interactions.

Model based approach hence involves building machine learning algorithms to predict user's ratings. They involve dimensionality reduction methods that reduce high dimensional matrices containing an abundant number of missing values with a much smaller matrix in lower-dimensional space. To understand this further lets understand what matrix factorization is.

a) Matrix Factorization:

The main assumption behind matrix factorisation is that there exists a pretty low dimensional latent space of features in which we can represent both users and items and such that the interaction between a user and an item can be obtained by computing the dot product of corresponding dense vectors in that space.

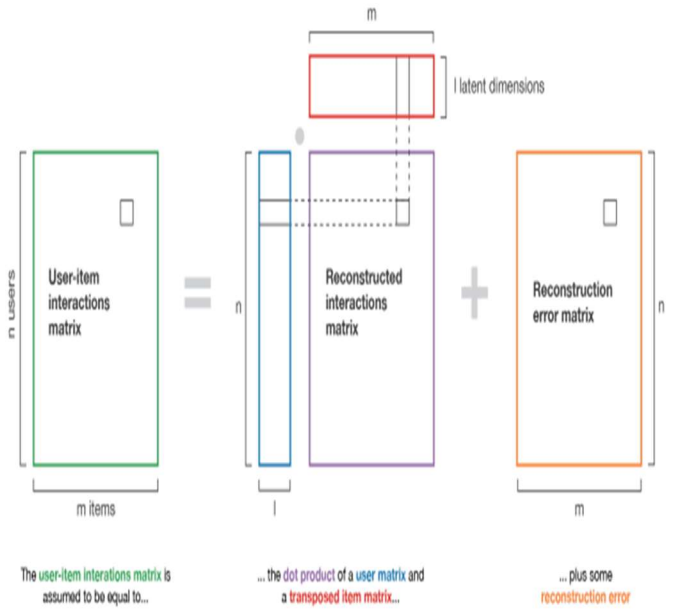


Illustration of the matrix factorization method.

b) Non-Negative Matrix Factorization (NMF):

Non-negative matrix factorization also non-negative matrix approximation is a group of algorithms in multivariate analysis and linear algebra where a matrix V is factorized into (usually) two matrices W and H, with the property that all three matrices have no negative elements. This non-negativity makes the resulting matrices easier to inspect. Also, in applications such as processing of audio spectrograms or muscular activity, non-negativity is inherent to the data being considered. Since the

problem is not exactly solvable in general, it is commonly approximated numerically.

Using NMF for our test data we observed that the RMSE and MAE scores are quite low.

c) Singular Value Decomposition (SVD):
Singular value decomposition also known as the SVD algorithm is used as a collaborative filtering method in recommendation systems. SVD is a matrix factorization method that is used to reduce the features in the data by reducing the dimensions from N to K where (K<N).

Why SVD? There are 3 primary reasons for that:
- It's very efficient
- The basis is hierarchical, ordered by relevance
- It tends to perform quite well for most data sets

We observed that the SVD model performed considerably better than the NMF model in terms of very low RMSE and MAE scores for the test set as well as in terms of lower test time taken for execution.

2. Memory Based Approach

The main characteristics of user-user and item-item approaches is that they use only information from the user-item interaction matrix and they assume no model to produce new recommendations.

a) User-User:

In order to make a new recommendation to a user, the user-user method roughly tries to identify users with the most similar "interactions profile" (nearest neighbours) in order to suggest items that are the most popular among these neighbours (and that are "new" to our user). This method is said to be "user-centred" as it represents users based on their interactions with items and evaluates distances between users.

Evaluation of this model is a bit different from the usual evaluation metrics in general. In Recommender Systems, there are a set of metrics commonly used for evaluation. We choose to work with Top-N accuracy metrics, which evaluates the accuracy of the top recommendations provided to a user, compared to the items the user has actually interacted with in the test set. This evaluation method works as follows:

- ✓ For each user
- ✓ For each item the user has interacted in test set
- ✓ Sample 100 other items the user has never interacted with.
- ✓ Ask the recommender model to produce a ranked list of recommended items, from a set composed of one interacted item and the 100 non-interacted items
- ✓ Compute the Top-N accuracy metrics for this user and interacted item from the recommendations ranked list
- ✓ Aggregate the global Top-N accuracy metrics

Below mentioned is a list of all evaluations performed for this model.

```
[225] user=int(input("Enter User ID from above list for book recommendation  "))
      model_recommender.recommend_book(cf_recommender_model,user)

Enter User ID from above list for book recommendation  2033
Recommendation for User-ID =  2033
        ISBN                                          Book-Title  recStrength
0  0440211727                                       A Time to Kill        0.212
1  059035342X  Harry Potter and the Sorcerer's Stone (Harry P...        0.196
2  0316666343                           The Lovely Bones: A Novel        0.170
3  0804106304                                   The Joy Luck Club        0.158
4  0590353403      Harry Potter and the Sorcerer's Stone (Book 1)        0.149
5  0836217691  Homicidal Psycho Jungle Cat: A Calvin and Hobb...        0.145
6  0836218051                     The Essential Calvin and Hobbes        0.143
7  0446310786                               To Kill a Mockingbird        0.142
8  0399501487                                   Lord of the Flies        0.137
9  0345370775                                      Jurassic Park        0.134
```

For a random set of users, the recall strength associated with its results is mentioned above. User 0446310786 has high recall capacity for certain books like To Kill a MockingBird meaning there is a high chance of this book being recommended. Similarly, for users lower than that have lesser recall strength meaning the corresponding book being recommended has a lower chance.

Similarly some other metrics used are:
1. Recall@K: Shows proportion of relevant items found in the top-k recommendations.
2. Hit@K: Proportion of the seen data being present in the top-k recommendations.

	hits@5_count	hits@10_count	interacted_count	recall@5	recall@10	User-ID
10	258	336	1389	0.186	0.242	11676
31	185	242	1138	0.163	0.213	98391
45	21	28	380	0.055	0.074	189835
30	84	101	369	0.228	0.274	153662
70	32	35	236	0.136	0.148	23902
7	26	46	204	0.127	0.225	235105
47	25	29	203	0.123	0.143	76499
50	25	34	193	0.130	0.176	171118
42	62	73	192	0.323	0.380	16795
43	24	31	188	0.128	0.165	248718

b) Item-Item:

To make a new recommendation to a user, the idea of the item-item method is to find items similar to the ones the user already "positively" interacted with. Two items are considered to be similar if most of the users that have interacted with both of them did it in a similar way. This method is said to be "item-centred" as it represents items based on interactions users had with them and evaluates distances between those items.

For the scope of our project, we used the K-Nearest Neighbours algorithm. kNN is a machine learning algorithm to find clusters of similar users based on common book ratings, and make predictions using the average rating of top-k nearest neighbors. We convert our table to a 2D matrix, and fill the missing values with zeros (since we will calculate distances between rating vectors). We then transform the values(ratings) of the matrix data frame into a scipy sparse matrix for more efficient calculations.

Testing model and making Recommendations:
In this step, the kNN algorithm measures distance to

determine the “closeness” of instances. It then classifies an instance by finding its nearest neighbors, and picks the most popular class among the neighbors.

Recommendations for Harry Potter and the Sorcerer's Stone (Book 1)

What do we observe after implementing both the memory based techniques?

The user-user method is based on the search of similar users in terms of interactions with items. As, in general, every user has only interacted with a few items, it makes the method pretty sensitive to any recorded interactions (high variance). On the other hand, as the final recommendation is only based on interactions recorded for users similar to our user of interest, we obtain more personalized results (low bias).

Conversely, the item-item method is based on the search of similar items in terms of user-item interactions. As, in general, a lot of users have interacted with an item, the neighbourhood search is far less sensitive to single interactions (lower variance). As a counterpart, interactions coming from every kind of user (even users very different from our reference user) are then considered in the recommendation, making the method less personalised (more biased). Thus, this approach is less personalized than the user-user approach but more robust.

CONCLUSION

- 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.
- Author with the most books was Agatha Christie, William Shakespeare and Stephen King.
- Amongst the memory based approach, item-item CF performed better than user-user CF because of lower computation requirements .
- For modelling, it was observed that for model based collaborative filtering SVD technique worked way better than NMF with lower Mean Absolute Error (MAE) .

FUTURE SCOPE

- Given more information regarding the books dataset, namely features like Genre, Description etc, we could implement a content-filtering based recommendation system and compare the results with the existing collaborative-filtering based system.
- We would like to explore various clustering approaches for clustering the users based on Age, Location etc., and then implement voting algorithms to recommend items to the user depending on the cluster into which it belongs.

REFERENCES

- Geeksforgeeks
- Researchgate.net
- Analytics Vidhya
- Github profiles