# Recommender-systems-colaborative-filtering

With the increase in ecommerce and use of internet applications, recommender systems are used in hundreds of algorithms today to suggest products and services to customers and users around the world. There are several online retailers like Amazon, eBay, Netflix and Facebook who vigorously depend on recommender systems in determining similar customers and items that are likely to of interest of given user. Recommendation system helps in filtering the large amount of data generated and provide adequate information which can be used to increase sale or maintain current users’ interests. There are different approaches to implement recommender system like collaborative filtering, content-based filtering, rule-based filtering and hybrid approach. Collaborative filtering gives recommendation based on preference of similar users. Content based approach makes recommendation to user based on items with similar content in user profile or items that user preferred in the past. We are using collaborative filtering approach for building the recommender system for this project and recommend top N book recommendations to the users.

Collaborative filtering predicts user preference by analyzing relationships between different users, finding the most similar users, and suggesting items based on their ratings or preference. We are using collaborative filtering approach on Book crossing dataset to recommend books to users based on preference of similar users.

The Book crossing dataset is an open source dataset, collected by Cai-Nicolas Ziegler from the Book Crossing community. Book Crossing is an act of leaving a book in public place to be picked up and read by others, who then do the same. The website bookcrossing.com is a free online book club which was founded with the intention to encourage reading. The dataset contains the information about 278,858 users (anonymized but with demographic information) providing 1,149,780 ratings about 271,379 books.

Data source: http://www2.informatik.uni-freiburg.de/~cziegler/BX/

We choose this dataset as it has all the required attributes needed to build a recommender system.

Steps followed and problems to be solved:

--> Data Cleaning and filtering

--> Data exploration

--> Model building- Memory based approach: similarity between different users is calculated using KNN, based on similar users ratings, prediction is made. Build a simple recommender system using the best settings.

--> Model building- Model based approach: using Single Value Decomposition SVD, and Negative Matrix Factorization machine learning algorithms to determine similarity.

--> Comparing models

--> Recommending top 10 items based on best model

Collaborative filtering can be implemented in two way: 1. Memory based approach 2. Model based approach In our project we have implemented both to recommend the books to the user. And, compared the advantages and disadvantages of both the approaches.

5.1. MEMORY BASED APPROACH

[1]Memory-Based Collaborative Filtering approaches can be divided into two main sections: user-item filtering and item-item filtering. A user-item filtering takes a particular user, find users that are similar to that user based on similarity of ratings, and recommend items that those similar users liked. In contrast, item-item filtering will take an item, find users who liked that item, and find other items that those users or similar users also liked. It takes items and outputs other items as recommendations.

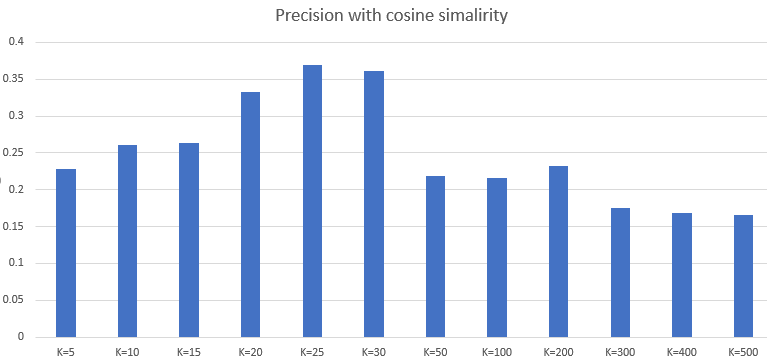
 Item-Item Collaborative Filtering: “Users who liked this item also liked …”  User-Item Collaborative Filtering: “Users who are similar to you also liked …”

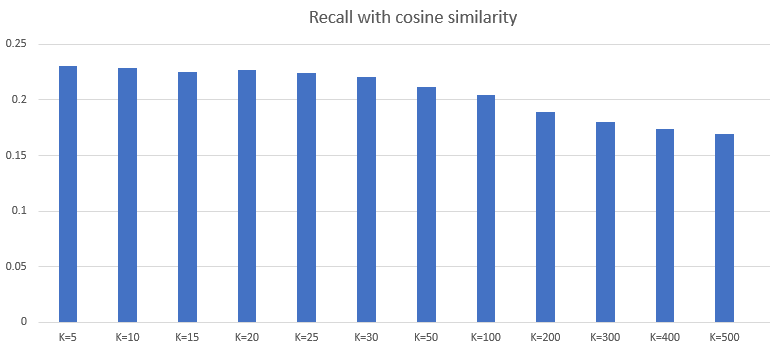
The key difference of memory-based approach from the model-based is that we are not learning any parameter using gradient descent (or any other optimization algorithm). The closest user or items are calculated only by using Cosine similarity or Pearson correlation coefficients, which are only based on arithmetic operations.

After, prepossing the dataset. We merge ratings, books and users into a single dataset. Because for this dataset the only requried fields are unique isbns and user is. Using memory based approach we built a simple popularity based recommendation system. To cope with the computing power of the machine we have reduced the dataset by removing the user who have not rated more than 100 books.

Now, our next step is to build a user-item matrix from the combined ratings table. The function ‘sim’ inputs userID and ratings matrix and returns similarities and indices of k similar users. Here, we use brutes alogrithm to calculate our matrix. This function returns the similarity distance between users. The next function ‘prediction’, for specified user-item predicts rating based on user-based approach. Function ‘recommend’ uses the two functions to recommend books for user-based.

Precision and recall values obtained after building KNN model for various values of K is as plotted below in the bar graph. We observe that for value of K =25 the precision values is maximum. Using this setting the recommender model is built in the following steps.





## **MODEL BASED APPROACH**

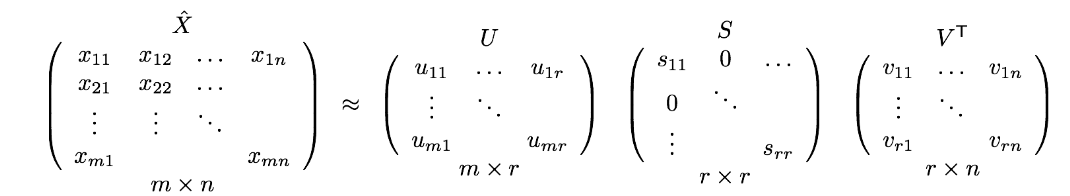
[1]In this approach, CF models are developed using machine learning algorithms to predict user’s rating of unrated items. As per my understanding, the algorithms in this approach can further be broken down into 3 sub-types:

1. Cluster based algorithm
2. Matrix Factorization
3. Deep Learning

**Matrix Factorization (MF):** The idea behind Matrix factorization is the attitudes or preferences of a user which can be found even with a few number of hidden attributes.

In our project, we apply Matrix factorization as python implementation for orthogonal factorization **(SVD**) and Non-negative factorization (**NMF**).

* The goal of CF-based recommendation algorithms is to suggest new products or to predict the utility of a product for a customer, based on the customer’s previous behavior and other similar customers ’opinions.
* However, these systems have some problems like sparsity and scalability. The weakness of CF algorithms for large, sparse databases led the researchers to alternative ways. In order to remove noise data from a large and sparse database, some **dimensionality reduction techniques** are proposed.
* SVD is a dimensionality reduction technique which captures the similarity between users and items.



* X denotes the utility matrix
* U is a left singular matrix which represents the relationship between users and **latent factors**
* S is a diagonal matrix describing the strength of each latent factor,
* V transpose is a right singular matrix, indicating the similarity between items and latent factors.

**NMF**

* Non-negative Matrix Factorization algorithm (NMF) is a matrix factorization algorithm which finds the positive factorization of a given positive matrix.
* It can eliminate invalid and redundant features in user-rating matrix

For this approach, we use the dataset containing just the ratings, user\_id and unique\_isbn. We initially, run our dataset on two algorithmic models:

SVD - Singular Value Decomposition

NMF - Non-negative Matrix Factorization

In order to implement these algorithms there is a package which is available to build recommender systems, it is ‘surprise’ package. [1]Surprise package: This package has been specially developed to make recommendation based on collaborative filtering easy. It has default implementation for a variety of CF algorithms. It uses all the functions from sklearn to build the models and makes our task easier. Please refer to the [link](https://surprise.readthedocs.io/en/stable/prediction_algorithms.html) for more information.