**Book Recommendation System**

**Gaurav Gade, Hitesh Verma, Anand Gend**

**Data science trainees,**

**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 fulfil 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 to give more accurate recommendations.

**Keywords:** Recommendation System, Collaborative Filtering, Matrix Factorization, Pandas, SVD (Singular Value Decomposition)

**Introduction**

Recommendation system is an information filtering system proposed to reduce the additional cost of users in the search process. The design of Recommendation system is mainly based on the concepts of association rules, content filtering and collaborative filtering . According to the preferences of groups with the same hobbies and interests, users can recommend the information they are interested in. Through cooperative mechanism, users can respond to the information to a certain extent (such as rating, borrowing times)and record it to filter it, so as to help others choose the information. Collaborative filtering can be divided into group filtering and evaluation .

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, You tube knows what type of videos you will watch based on your search history, watch history, or purchase history.

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.

Recommendation systems are used in hundreds of different services - everywhere from online shopping to music to movies. For instance, the online retailer Amazon had a heavy hand in developing collaborative filtering algorithms that recommend items to users. Music services like Pandora identify up to 450 uniquely identifying characteristics of songs to find music similar to that of their users’ preferences. Other music streaming services, such as Spotify, heavily rely upon the music selections of similar users to make weekly song recommendations and personalized radio stations. Netflix, a popular television and movie streaming service, uses these systems to recommend movies that viewers may enjoy. We can see how recommendation systems have a surprisingly large impact on the materials consumers engage with over the course of their daily lives. For our senior comprehensive project, we analyzed three systems that predict how users will rate specific books. Our system that we created makes these predictions based on data gathered from the Amazon Book Reviews dataset, Book Crossing dataset, Google Books API, and Good Reads API. To accurately predict users’ reactions to books, we’ve integrated several strategies in the field of recommendation systems.

**Steps involved:**

* **Exploratory Data Analysis**

Exploratory data analysis is an statistical way of understanding the data which is usually done in a visual way. The graphs plotted in exploratory data analysis are for better understanding of data to the analyst.

After loading the dataset we performed this method by comparing our target variable with other independent variables. This process helped us figuring out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

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

* **Null values Treatment**

After the data is loaded , The missing data is checked using is.na() or isnul() function . The output depicted that there was many missing values in our dataset.

* **Feature Engineering**

To make the data tenable for understanding and further analysis , the data set was analyzed for identifiable statistical trends and patterns.

* **Standardization of features**

Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it.

The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.

* **Fitting different models**

For modelling we tried various algorithms like:

1.Popularity Based Recommendation

2.Model based collaborative filtering

3.Collaborative Filtering --(Item Item based)

4.Collaborative Filtering --(User Item based)

* **Algorithms:**

**1.** **Popularity-Based Recommendation System:**

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 product or movie which are in trend or are most popular among the users and directly recommend those .

For example, if a product is often purchased by most people then the system will get to know that that product is most popular so for every new user who just signed it, the system will recommend that product to that user also and chances becomes high that the new user will also purchase that.

Merits of popularity based recommendation system

It 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.

Demerits of popularity based recommendation system

Not personalized

The system would recommend the same sort of products/movies which are solely based upon popularity to every other user.

Example

Google News: News filtered by trending and most popular news.

YouTube: Trending videos.

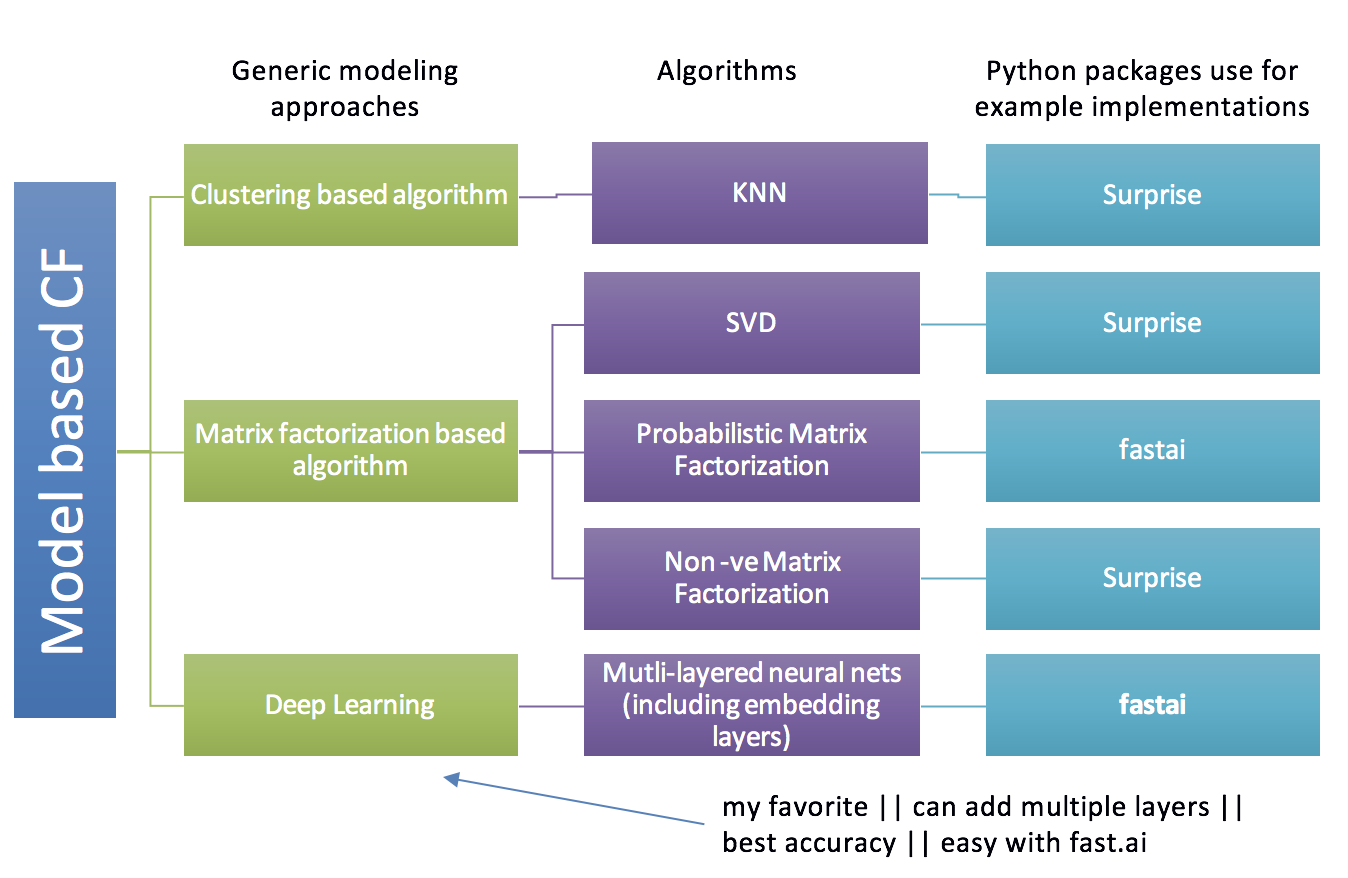
**2.** **Model based collaborative filtering:**

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 techniques (hang on, will be discussed in next paragraph) 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.

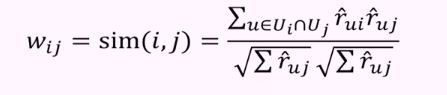


**2.Collaborative Filtering - (Item-Item based):**

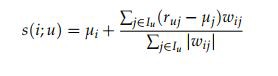
ITEM-ITEM collaborative filtering look for items that are similar to the articles that user has already rated and recommend most similar articles. But what does that mean when we say item-item similarity? In this case we don’t mean whether two items are the same by attribute like Fountain pen and pilot pen are similar because both are pen. Instead, what similarity means is how people treat two items the same in terms of like and dislike.

This method is quite stable in itself as compared to User based collaborative filtering because the average item has a lot more ratings than the average user. So an individual rating doesn’t impact as much.

To calculate similarity between two items, we looks into the set of items the target user has rated and computes how similar they are to the target item i and then selects k most similar items. Similarity between two items is calculated by taking the ratings of the users who have rated both the items and thereafter using the cosine similarity function mentioned below:



Once we have the similarity between the items, the prediction is then computed by taking a weighted average of the target user’s ratings on these similar items. The formula to calculate rating is very similar to the user based collaborative filtering except the weights are between items instead of between users. And we use the current users rating for the item or for other items, instead of other users rating for the current items.



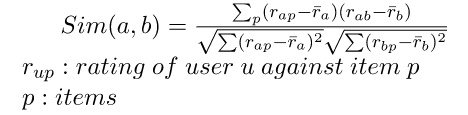
**2.Collaborative Filtering - (User-Item based):**

User-Based Collaborative Filtering is a technique used to predict the items that a user might like on the basis of ratings given to that item by the other users who have similar taste with that of the target user.  
Many websites use collaborative filtering for building their recommendation system.

**Steps for User-Based Collaborative Filtering:**

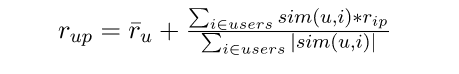
**Step 1: Finding the similarity of users to the target user U.**

Similarity for any two users *‘a’* and *‘b’* can be calculated from the given formula,



**Step 2: Prediction of missing rating of an item**  
Now, the target user might be very similar to some users and may not be much similar to the others. Hence, the ratings given to a particular item by the more similar users should be given more weightage than those given by less similar users and so on. This problem can be solved by using a weighted average approach. In this approach, you multiply the rating of each user with a similarity factor calculated using the above mention formula.

The missing rating can be calculated as,



**Model Evaluation :**

In Recommender Systems, there are a set metrics commonly used for evaluation. We choose to work with TopN accuracy metrics, which evaluates the accuracy of the top recommendations provided to a user, comparing to the items the user has actually interacted in test set.

Thisevaluation method works as:

1.For each user

2.For each item the user has interacted in test set

3.Sample 100 other items the user has never interacted.

4.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

5.Compute the TopN accuracy metrics for this user and interacted item from the re

commendations ranked list

6.Aggregate the global Top N accuracy metrics

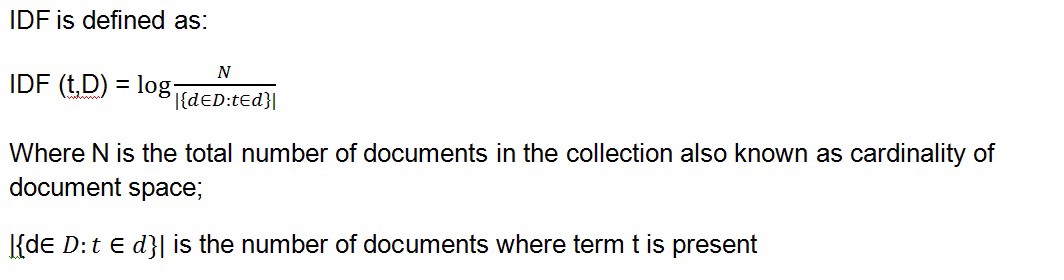
Similarly **some other metrics** used are:

1.**Recall@K**: Shows proportionof relevant items found in the top-k recommendations.

**2.Hit@K**: Proportion of the seen data being present in the top-k recommendations.

**TF — Term Frequency -** how frequent the term/word appears in the document

**TF- IDF -**  stands for Term Frequency and Inverse Document Frequency .TF-IDF helps in evaluating importance of a word in a document.



**Cosine Similarity:**

Well cosine similarity is a measure of similarity between two non zero vectors. One of the beautiful thing about vector representation is we can now see how closely related two sentence are based on what angles their respective vectors make.

Cosine value ranges from -1 to 1.

So if two vectors make an angle 0, then cosine value would be 1, which in turn would mean that the sentences are closely related to each other.

If the two vectors are orthogonal, i.e. cos 90 then it would mean that the sentences are almost unrelated

**Conclusion:**

1.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

2.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

3.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

4.Author with the most books was Agatha Christie, William Shakespeare and Stephen King

5.We can conclude that item item based collaborative filtering performed better than user user based collaborative filtering because of lower computation among the memory based approach

6.For modelling, it was observed that for model based collaborative filtering SVD technique worked way better than NMF with lower Mean Absolute Error (MAE)