## Book Recommendation System

## Team Mate - Chetan Patil, Mrunal Badgujar,

## Sachin Chaudhari, Rajesh Patil,

**Alma Better Capstone Project**

**Abstract:**

A recommendation engine is a class of machine learning which offers relevant suggestions to the customer. A recommendation system is one of the top applications of data science. Every consumer Internet company requires a recommendation system like Netflix, YouTube, a news feed, etc. What you want to show out of a huge range of items is a recommendation system. 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, and YouTube knows what type of videos you will watch based on your search history, watch history, or purchase history. We were provided with a ‘Book Recommendation System’ dataset to perform machine learning tasks, where to get insights from user-items interactions and provide the best recommendation to users.

Our experiment can help understand what could be the reason for unsupervised problems of such labels by feature engineering, data analysis, and prediction with machine learning models taking into account previous trends to determine unsupervised problems.

***Keywords: machine learning, unsupervised, collaborative filtering, recommendation system***

# 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.

Content

The Book-Crossing dataset comprises 3 files.

● Users

Contains the users. Note that user IDs (User-ID) have been anonymized and mapped 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 the case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavors (Image-URL-S, Image-URL-M, Image-URL-L), i.e., small, medium, and large. These URLs point to the Amazon website.

● 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.

# Introduction

A book recommendation system is a type of recommendation system where we have to recommend similar books to the reader based on his interest. The books recommendation system is used by online websites which provide eBooks like google play books, open library, good Read, Amazon Kindle, etc. We’ve to use the Collaborative based filtering method to build a book recommender system. The major concern is about providing the best recommendation to users. Looking towards various factors/features we can take leverage data to make better predictions on test sets. To let proceed with the recommendation it’s important to have every user’s interactions with items.

Generally, the models can predict the best recommendation for the data according to its RMSE score. As for Collaborative Filtering, we’ll be focusing on RMSE. This recommendation system data can show how it varies from every machine learning approach. So, generating different RMSE scores from different methods can reveal the best recommendation.

Our goal is to build a predictive recommendation model, which could help a company in predicting get insights from user-items interactions and provide the best recommendation to users.

# Users-Items Description of Data

The user’s interaction is a very vital role in a recommendation. To successfully build a collaborative filtering model in a recommender system, data preparation is important. Beginning with book data–dropping URL features (i.e. 'Image-URL-S', 'Image-URL-M', 'Image-URL-L'). We have some extra columns which are not required for our task like image URLs. And we rename the columns of each file as the name of the column contains space and lowercase letters so we will correct as to make it easy to use. The features depict great analysis by feature engineering.

# Observation from features and hypothetical assumption

The dataset is reliable and can consider a large dataset. We have 271360 books data and the total registered users on the website are approximately 278000 and they have given near about 11 lakh ratings. hence, we can say that the dataset we have is nice and reliable.

Agatha Christie is leading at the top with more than 600 counts, followed by William Shakespeare. It can happen in some possible cases that Agatha Christie is not the best Author, though Agatha Christie has the greatest number of books as compared to others. William Shakespeare is one of the most popular Authors in the world. Still, he doesn't have the highest number of books. Among all other Authors, a few of the Authors might have some of the best seller books that have millions of copies sold in the world.

Harlequin has the greatest number of books published, followed by Silhouette. Hypothetical assumptions - Some of the top authors had published their books from Harlequin. We can observe Harlequin publisher's marking better performance than any other publishers. Penguin Books, Warner Books, Penguin USA, Berkely Publishing Group, and many more are among popular publishers remarking competition with Harlequin. Though Penguin Books Publisher has less number of books published it might happen that only top Authors are approaching Penguin Books Publisher.

There are 4618 entries as ‘0’ and 0 NaN entries in the Year of Publication field. Publication years are somewhat between 1950 - 2005. The publication of books got vital when it starts emerging in 1950. It might happen people start to understand the importance of books and gradually got productivity habits in their life. Every user has their taste to read books based on what particular subject Author uses. The subject of writing books got emerged in late 1940 slowly. Till 1970 it has got the opportunity to recommend books to people or users that they love to read. The highest peak we can observe is between 1995-2001 year. The user understands what they like to read. Looking towards the raise the recommendation is also increased to understand their interest.

Looking towards the users aged between 30- 40 prefer more and somewhat we can also view between 20-30. Most of the user books are from Ages 30 to 40. The users might be more interested in the subject of what Authors are publishing in the market. The age group between 20-30 are immensely attracted to reading books published by Author. We can observe the same pitch for the Age group between 10-20 and 50-60. There can be a lot of different reasons.

As per ratings "Selected Poems" has been rated most followed by "Little Women". Selected Poems are most favorable to users as per ratings. Three of the books 'The Secret Garden', 'Dracula', and 'Adventures of Huckleberry Finn' are struggling to compete with each other. Similarly, we can observe this in 'Masquerade', 'Black Beauty', and 'Frankenstein'. Firstly the above ratings are unique ratings from the 'ratings\_data' and 'books\_data' datasets.

We have separated the explicit ratings represented by 1–10 and the implicit ratings represented by 0. Let's make some hypothesis assumptions - Mostly the users have rated 8 ratings out of 10 as per books. (i.e best books ever). Now, this count plot of book ratings indicates that higher ratings are more common amongst users and rating 8 has been rated the highest number of times. There can be many assumptions based on ratings of users - taking rating groups from 1-4. This can be a negative impact on books being published if they have ratings from 1 to 4. It can be issues related to language, offended by any chapter's incident/paragraph/Author, they've read the worst book ever. For 5 ratings the users might not sure about book ratings and whether it's a positive or negative impact. take the rating group from 6-10. This is positive feedback

It can happen that not every book is perfect in all desires. So, the users have decided to rate it 8. Since 6 ratings are very low among other ratings. As we can aspects 7 and 8 are average and have more ratings from users. 9 and 10 ratings are the top best ratings based on Author, Publisher, and a book published.

# Top-rated books as per ratings

Building a recommendation system based on popularity (i.e ratings). These recommendations are usually given to every user irrespective of personal characterization. Merged book\_data dataset and ratings\_explicit. Considering ISBNs that were explicitly rated for this recommendation system. So we achieved the top ten books as per ratings –

|  |  |
| --- | --- |
| book\_title | book\_rating |
| The Lovely Bones: A Novel | 707 |

It might happen that the feedback is positive but not extremely positive as 10 ratings

|  |  |
| --- | --- |
| Wild Animus | 581 |
| The Da Vinci Code | 494 |
| The Secret Life of Bees | 406 |
| The Nanny Diaries: A Novel | 393 |
| The Red Tent (Bestselling Backlist) | 383 |
| Bridget Jones’s Diary | 377 |
| A Painted House | 366 |
| Life of Pie | 336 |
| Harry Potter and the Chamber of Secrets (Book 2) | 326 |

The above are the top 10 books recommendation as per ratings. But this is not based on some recommendation system. They are top 10 books as per ratings.

# Steps Involved

## Exploratory Data Analysis

Analytics for every dataset (i.e book, users, ratings) has helped to understand user-item interactions for a book recommendation. Viewed top books as per ratings. Analysis based on top authors with the highest number of books, top publishers with highest number of books, number of books published yearly, users’ age distributions, top books as per ratings, different various user ratings.

## Null values treatment

We served with three datasets (i.e book\_dataset, users\_dataset, ratings\_dataset). We got null values in book dataset (features are book-author, publisher, Image-URL-L), users dataset in age. Ratings dataset doesn’t contain any null values.

## Dropping and replacing data

Proceeding with data cleaning and feature selection is a crucial step – we dropped features like image URLs. Replaced feature with lowercase and ‘-‘ to build a space between words. Some of the null values were present in feature data, which we replaced with mean of that particular feature. Deal with mismatch features like book\_title, book\_author, year\_of\_publication, publisher. Only considering ages between 5-

90 we took users’ data to analyze and perform recommendation on it.

## Univariate Analysis

Data Visualization is a major part of a project to understand each perspective view. We perform analysis on single feature. It could be a simple and easy stage but more effective to focus on. Both numeric and categorical feature has helped with analysis and got insight. Each feature like book\_title, ISBN, book\_author, year\_of\_publication, publisher, user, age, book\_rating etc. shows a great impact on the analysis and training the recommendation model.

## Collaborative Filtering in memory-based and model based

Collaborative Filtering (CF) techniques make collaborative research and process over user or item ratings to deduce new recommendations for users. This collaborative research includes finding similarities between users and items to make assumptions for missing rating values and deducing new recommendations.

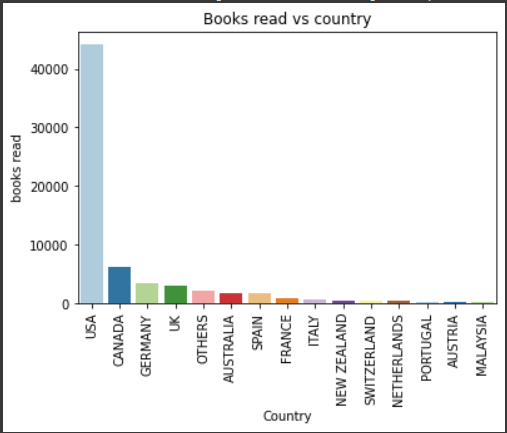
Memory-based approach was our first trial on the train and test dataset which uses the memory of previous users interactions to compute users similarities based on items they’ve interacted (i.e user-based approach) or compute items similarities based on the users that have interacted with them (i.e item-based approach). Applying cosine similarity to make item-item similarity need to take transpose of matrix. This matrix would help in manage train-test matrix. After all views predictions based on similarity, we find recommendation on it based on score.

Model-based collaborative filtering algorithms provide item recommendation by first developing a model of user ratings. Algorithms in this category take a probabilistic approach and envision the collaborative filtering process as computing the expected value of a user prediction, given his/her ratings on other items. We use Latent Factor Model called Singular Value Decomposition (SVD). SVD made dimensionality reduction technique in machine learning. SVD is a matrix factorization technique, which reduces the number of features of a dataset by reducing the space dimension from N- dimension to K-dimension (where K<N). It uses a matrix structure where each row represents a user, and each column represents an item. The elements of this matrix are the ratings that are given to items by users.

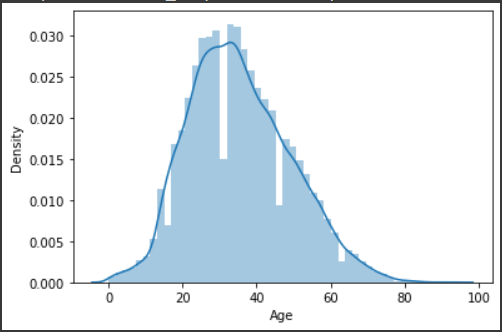
## Model Evaluation Metrics

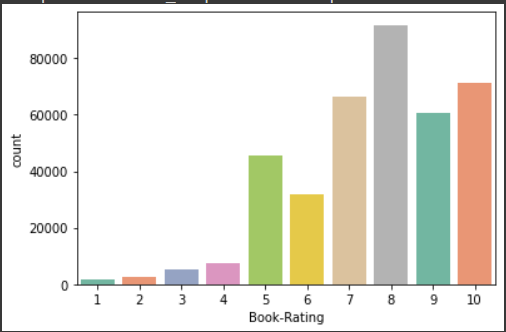
Model evaluation metrics is important to distinguish the best collaborative filtering – either by memory based or model based approach. The memory based approach – Cosine Similarity shows RMSE score for item based CF is 8.00 and for user based CF it shows 8.00. The score is slightly similar. Model based collaborative filtering made it better score with Latent Factor Model called SVD. The score improved to 1.63 for both SVD RMSE and accuracy score.

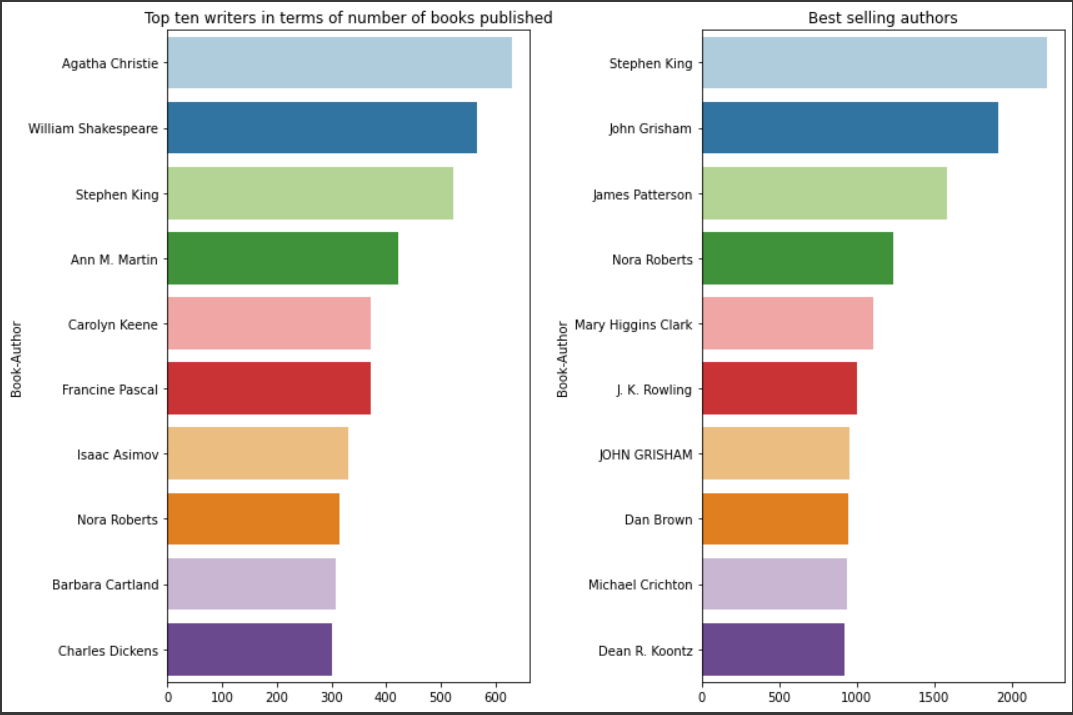
Countries with maximum number of users



Age distribution of users

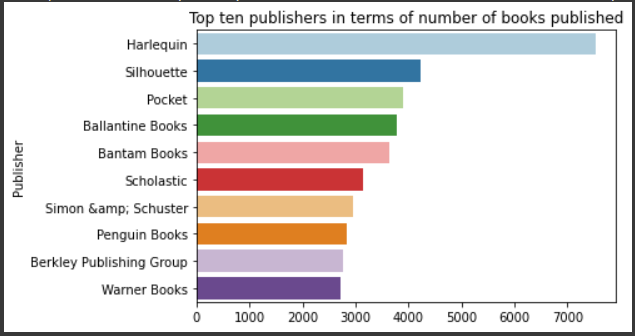
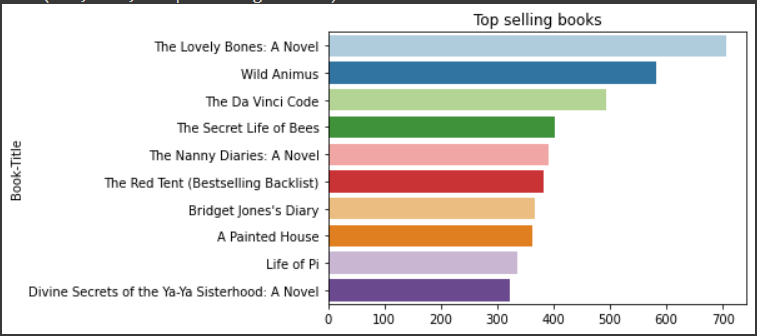


Rating distribution for explicit rating



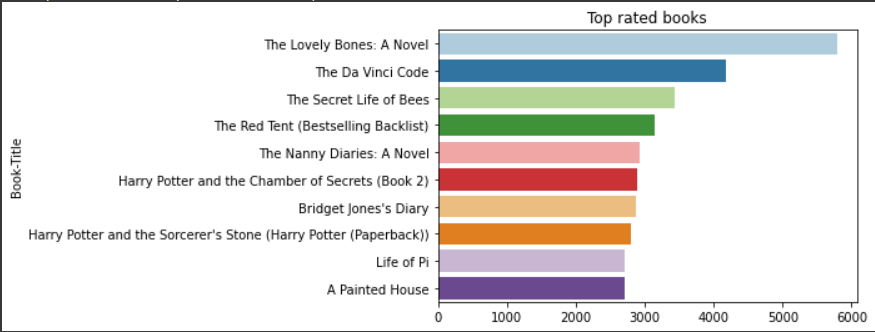
Agatha Christie has wrote and published most number of books

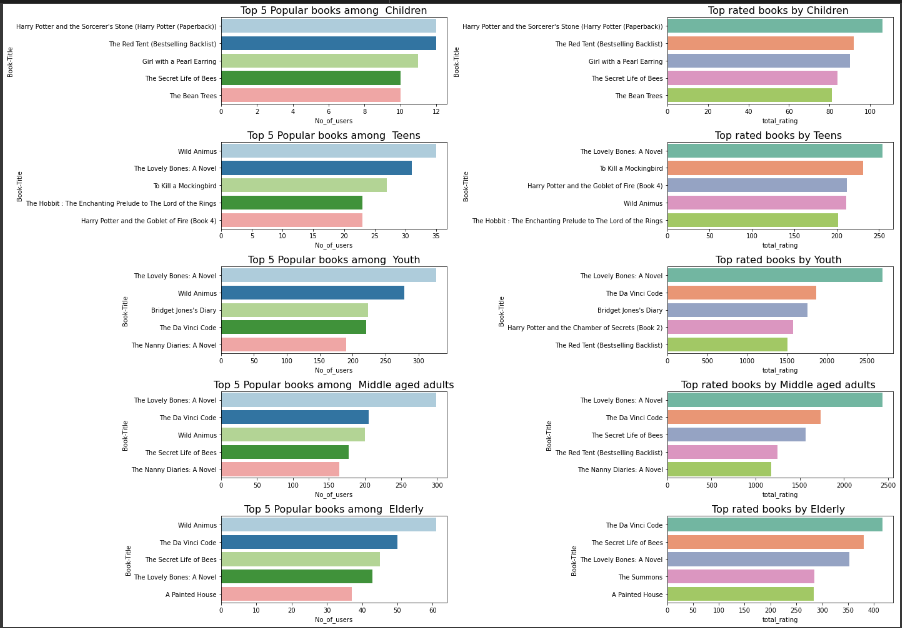
Stephen king is the Best selling author



Companies with the most number of books published Top selling books

Top-rated books





1. **Collaborative Filtering**

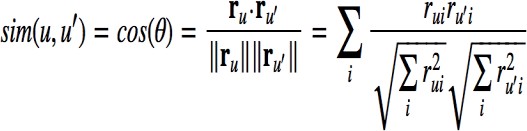
Collaborative Filtering (CF) techniques make collaborative research and process over user or item ratings to deduce new recommendations for users. This collaborative research includes finding similarities between users and items to make assumptions for missing rating values and deducing new recommendations. CF techniques are grouped in two methods: Memory-based and Model-based methods.

* + **Memory based CF** 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 between the memory-based approach from the model-based techniques 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.

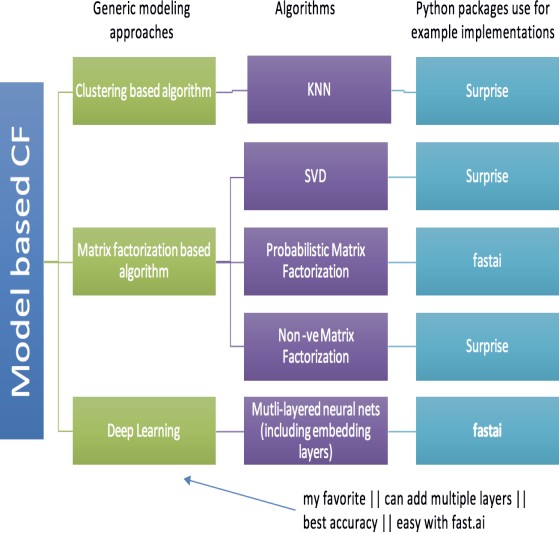
A common distance metric is cosine similarity. For user-based collaborative filtering, two users’ similarity is measured as the cosine of the angle between the two users’ vectors. For users u and u′, the cosine similarity is:



As no training or optimization is involved, it is an easy-to-use approach. But its performance decreases when we have sparse data which hinders the scalability of this approach for most of the real-world problems.

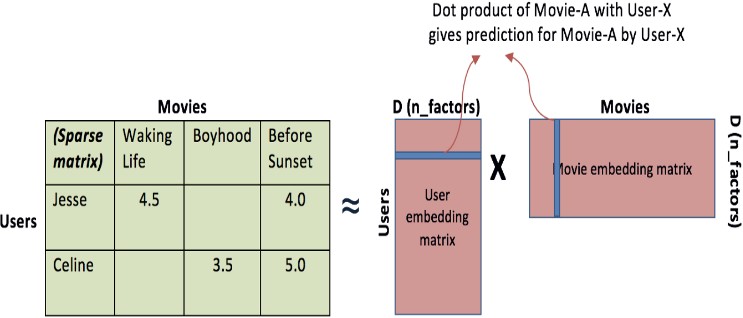
## Model-based CF

In this approach, CF models are developed using machine learning algorithms to



## Matrix Factorization (MF):

The idea behind such models is that the attitudes or preferences of a user can be determined by a small number of hidden factors. We can call these factors Embeddings.

Intuitively, we can understand embeddings as low-dimensional hidden factors for items and users**.**

predict user’s rating of unrated items. As per my understanding, the algorithms in this approach can further be broken down into three sub-types.

In the SVD model, an estimated rating of user *u* on item *i* is calculated as:

https://miro.medium.com/max/426/1*r6lwnjiy11x13gYZXUzV7w.png

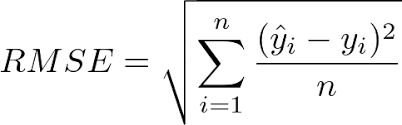
where *μ* is the overall average rating, and every other parameter is calculated from the model with a gradient descent method. So the model will try to fit this estimated rating on all the known ratings, minimize the MSE, and return the closest fit. *bᵤ* and *bᵢ* are scalars, they represent the biases of the user *u* or item *i.*

*pᵤ* and *qᵢ* are vectors, and their length is a hyperparameter of the model, *n*. They are the actual matrix-factorization part of the model, that is where the magic happens. Each user and item will be represented by their vector, which tries to capture their essence in *n* numbers. And we get the rating by multiplying the item—user pairs (and adding averages and biases of course.

Training the SVD model like other models and testing the model performance using RMSE score (which stands for Root Mean Squared Error, the lower the better).

## Model Evaluation metrics

RMSE score is the best way to model performance for recommendation systems. Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data.



# Conclusion

That’s it! I reached the end of the exercise. Starting with loading the data so far I have done data cleaning and feature engineering, null values treatment, and some univariate analysis. Collaborative Filtering was among the best methods to approach the recommendation system for this project. A model-based approach like the Latent Factor Model called SVD and Memory based approach with cosine similarity was model building approach. The comparison of the RMSE score between the model and the memory-based approach was quite different. Model Evaluation metrics show better recommendations with model-based CF. The RMSE score varies in both the model and the optimal model we can find in SVD. So Model evaluation metrics are important to distinguish the best collaborative filtering – either by memory-based or model-based approach. The memory-based approach – Cosine Similarity shows RMSE score for item-based CF is 8.00 and for user-based CF it shows 8.00. The score is slightly similar. Model-based collaborative filtering made it a better score with the Latent Factor Model called SVD. The score improved to 1.63 for both SVD RMSE and accuracy scores.

SVD with RMSE score is the best model with 1.63 for this dataset. This performance could be due to various reasons: pattern of data, different models giving different accuracy scores, business understanding, machine learning approach, etc. Finally, Singular Value Decomposition (SVD) is an optimal model for the book recommendation system of this dataset.

## References: -

1. Google
2. Alma Better
3. Kaggle