CAPSTONE PROJECT - IV

# Book Recommendation System

By:

Shrinidhi Choragi

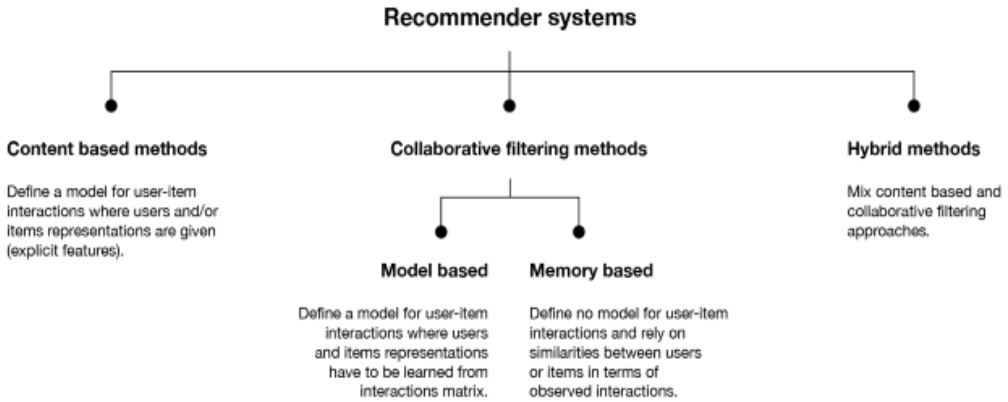
Data Science Trainee

Almabetter

# Contents

# Introduction

- Recommendation System is an information filtering technology used in a wide range of platforms as per the interest of users and is implemented in applications like movies, music, venue, books, research articles, tourism, and social media in general.




# Problem Statement

The main objective is to *create a machine learning model to recommend relevant books to users while exploring different algorithms for the same.*

The following points are investigated:

- What is a recommendation system and how does it work?
- What hypothesis can be made from the data analysis?
- Explore/understand the  different algorithms that recommend books.
- What are the pros and cons of different approaches and what solutions are suggested?

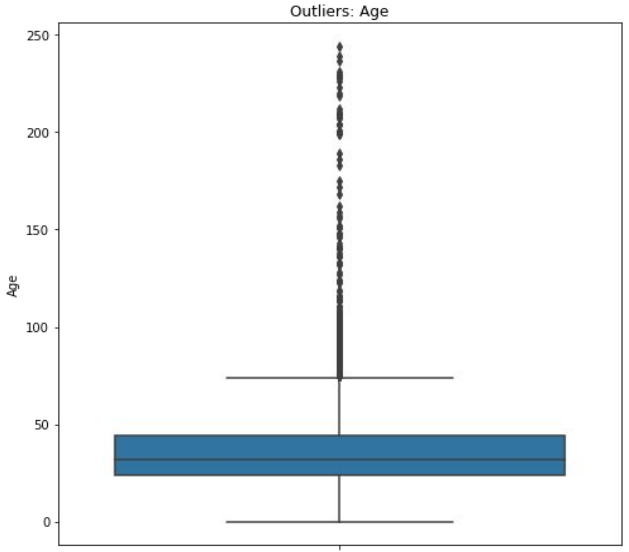
# Dataset

The ***Book-Crossing*** dataset comprises of the following three files.

- ***Users***
  - *UserID*
  - Demographic Data : *Location*, *Age*
- ***Books***
  - *ISBN*
  - *Book-Title*, *Book-Author*, *Year-Of-Publication*, *Publisher*
  - Image URL ( *Image-URL-S*,*Image-URL-M*, *Image-URL*-L)
- ***Ratings***
  - *Book-Rating*
    - Contains explicit rating expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit,expressed by 0.

# Data Preprocessing

- The anomalous entries in feature *Year-Of-Publication* are replaced with the median value.
  Anomalies:
  - Entries for that are greater than 2020
  - The entries of year as '0'.
- The missing values in the feature variable "*Age*" are imputed with the median value.
- The anomalous values/missing values in *Book-Author*, *Publisher* are correctly matched/imputed.
- The values of "*Age*" greater than 80, and lesser than 10 are replaced with the median value.
- The observations of ratings corresponding to the books that aren't present in "Books" dataset are excluded.
- The implicit ratings provide no information regarding user interaction. Hence dropped.




***Outliers: "Age"***

# Feature Engineering

## ➢ Feature Imputation

- New features the "*Average-Rating*" and the "*Book-Rating-Count*" are formed using the "*Book-Rating*" feature information.
    - The data is grouped based on "*ISBN*" to obtain the average ratings per book using mean transformation.

***Average-Rating= Mean of ratings given for a book***

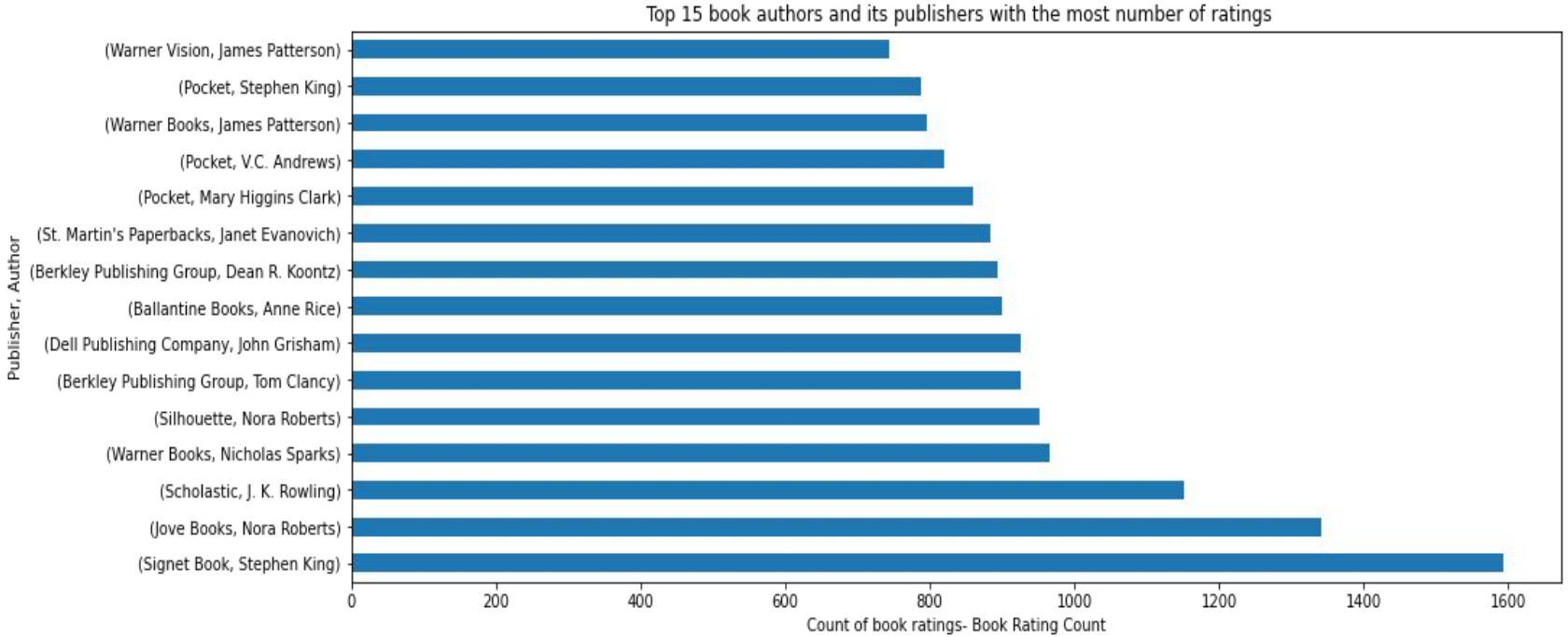
    - Similarly, grouping based on "*ISBN*" in order to obtain the count of ratings gives the "*Book-Rating-Count*" for each book.

***Book-Rating-Count = Number of ratings for a book***

- Information regarding country and state names is extracted from the existing column "*Location*", and the column is dropped later.

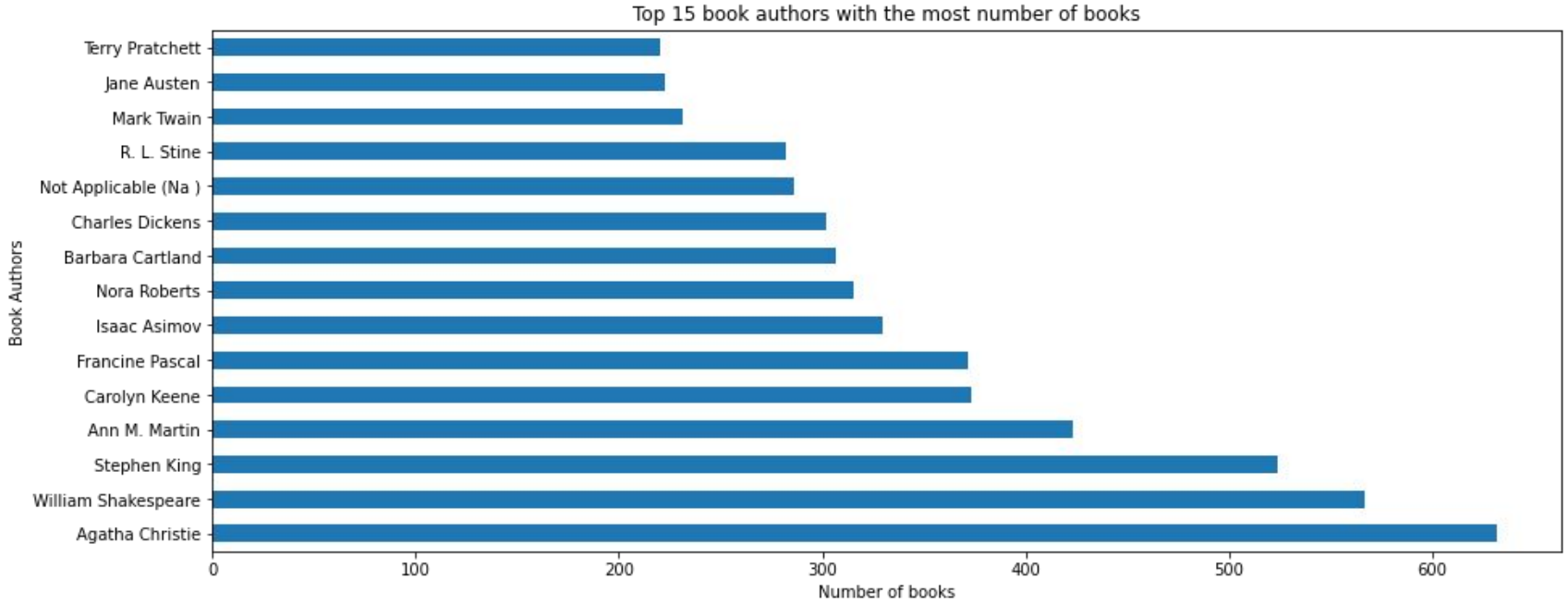
# Exploratory Data Analysis

***"Publishers" and "Book-Authors" with the most number of ratings***



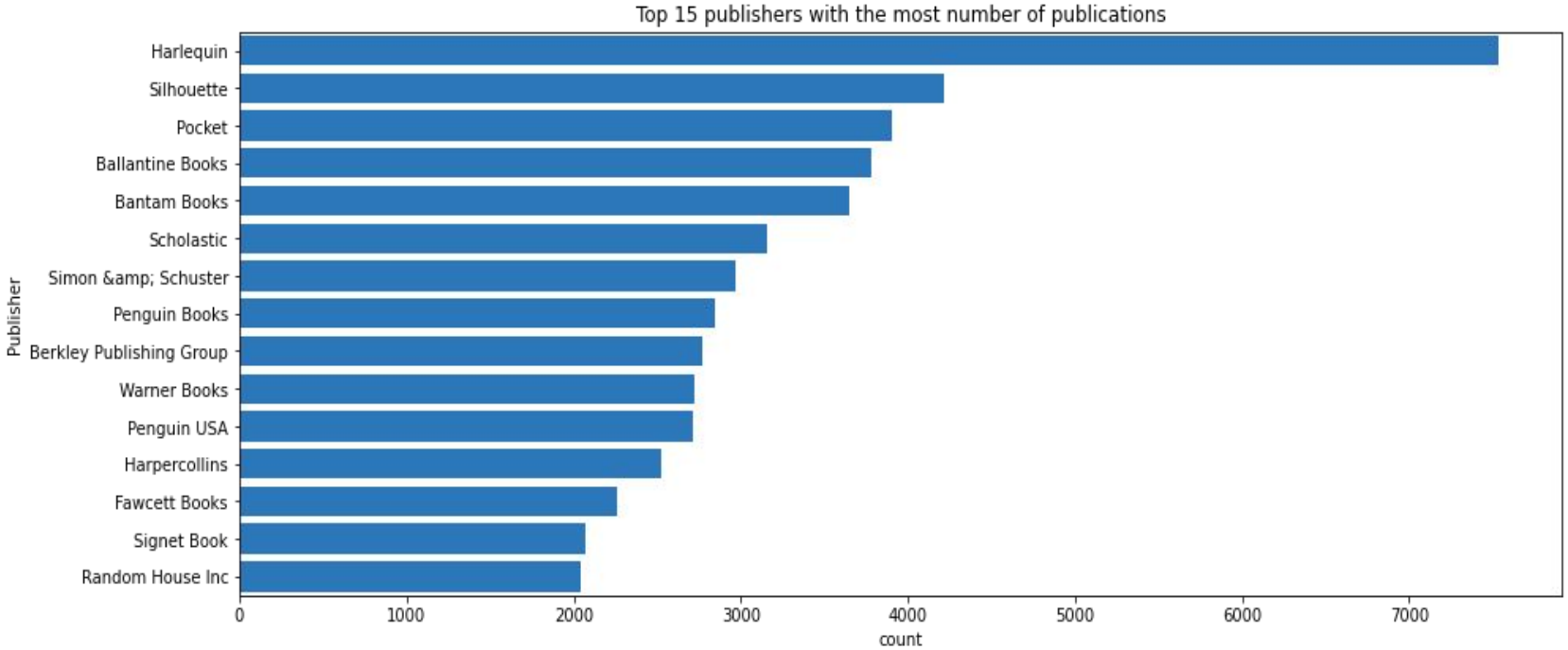
# Exploratory Data Analysis

***"Book-Authors" with the most number of books***



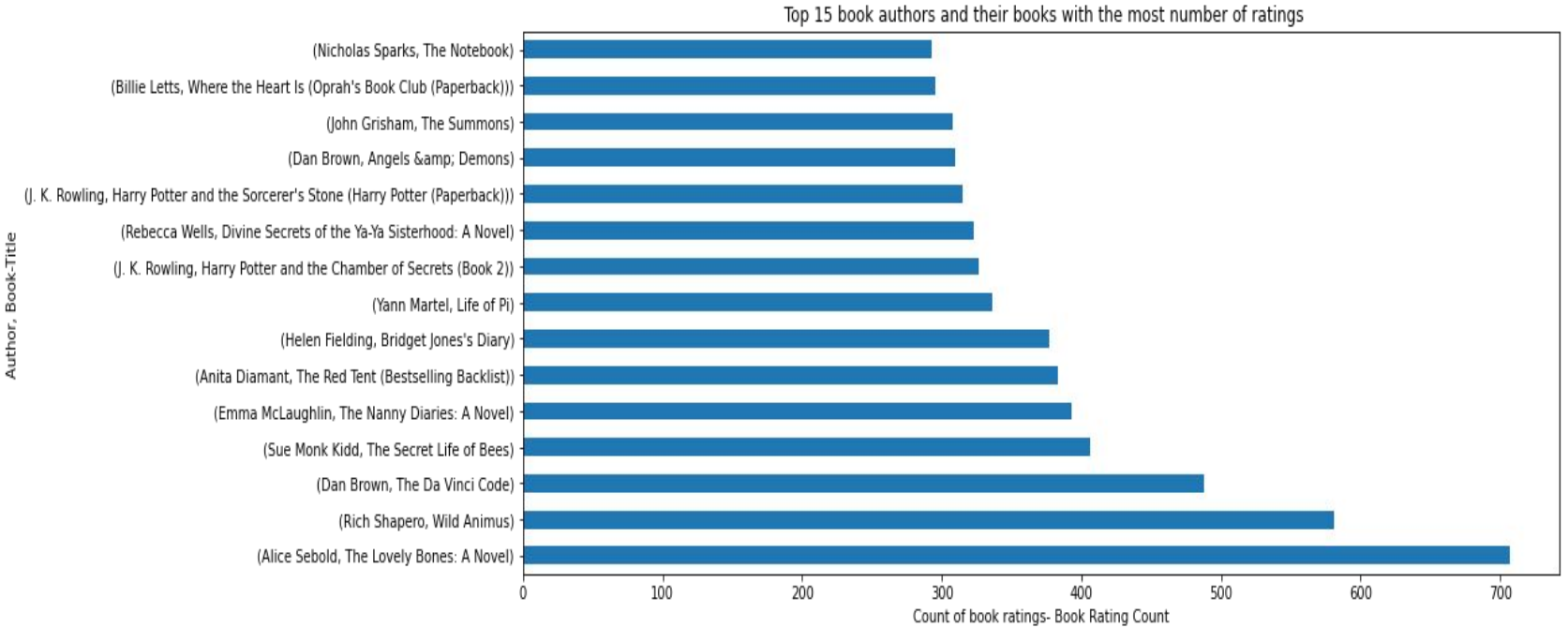
# Exploratory Data Analysis

***"Publishers" with the most number of book publications***



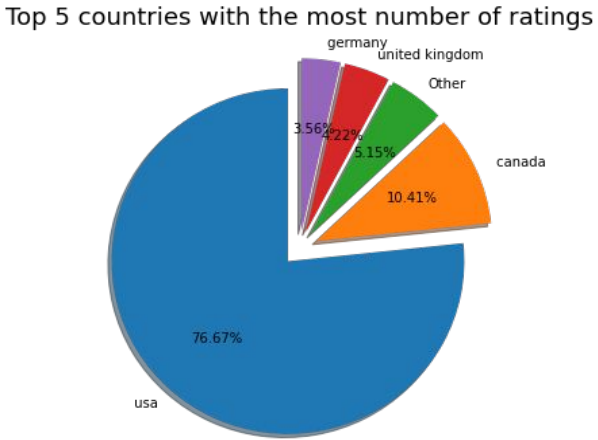
# Exploratory Data Analysis

***"Book-Authors" and "Book-Names" with the most number of ratings***

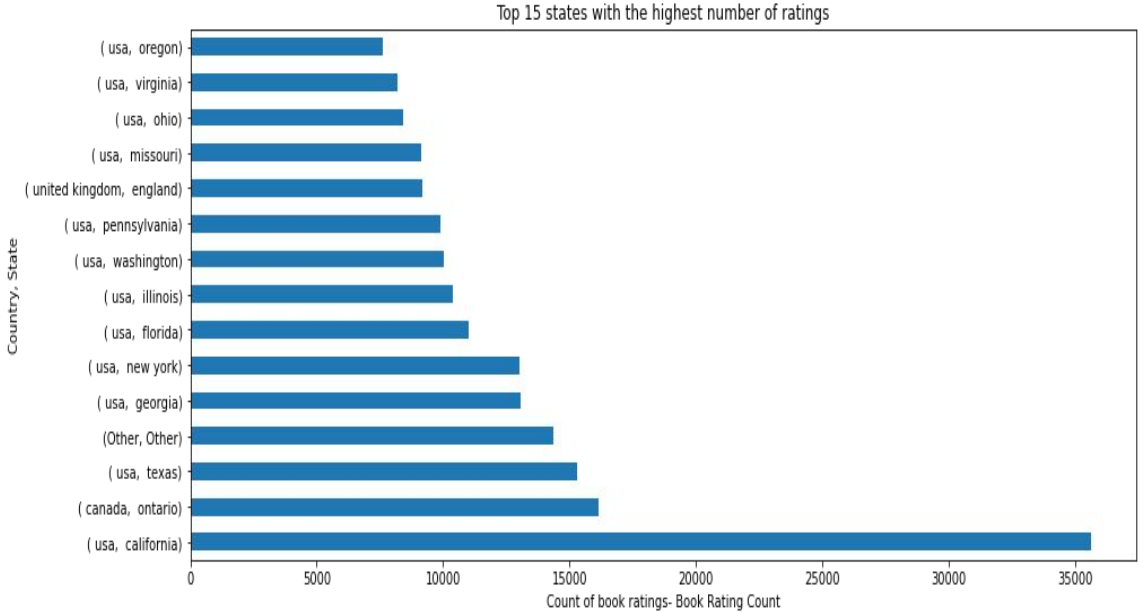


# Exploratory Data Analysis

## Countries with the most ratings

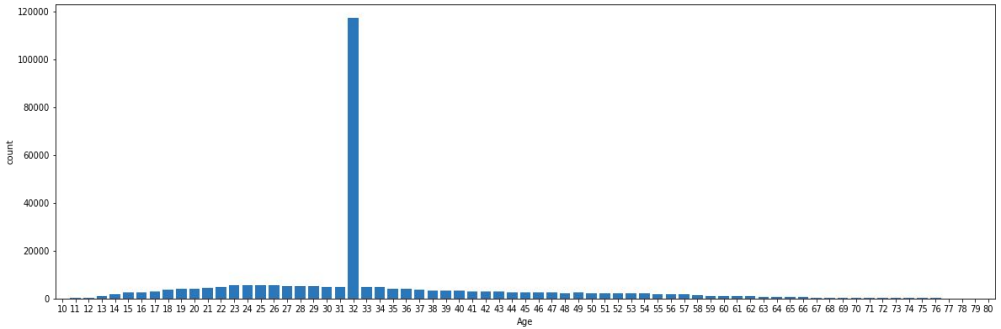



## Countries, States with the most number of ratings

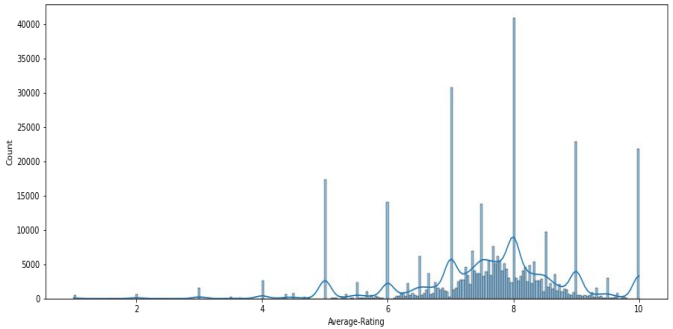


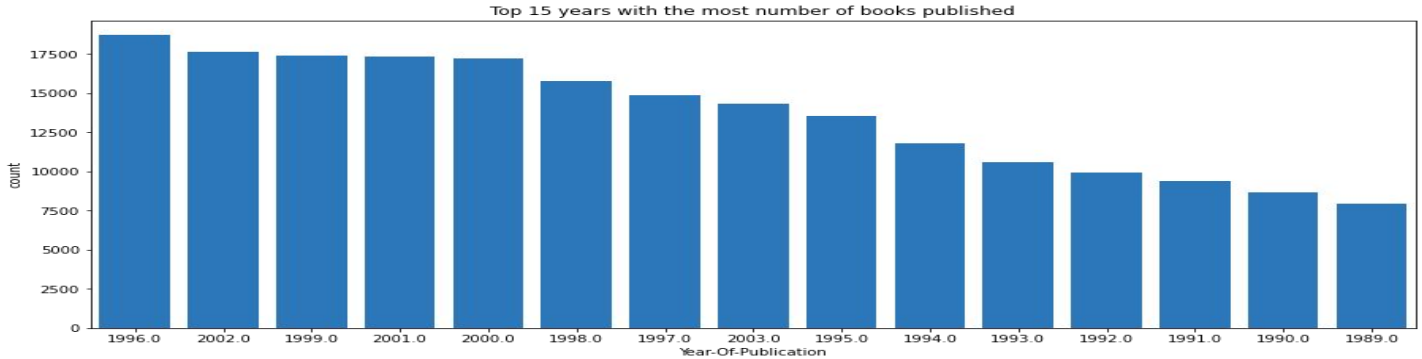
# Exploratory Data Analysis

### *Age distribution*




### *"Average Rating" Distribution*




***Years with the most number of books published***

# Recommenders and Evaluation Metrics

The following recommenders have been studied and implemented on the given dataset:

- *Recommender System- Popularity*
- *Recommender System- Weighted Average*
- *Recommender System- Collaborative Filtering*
  - *Memory-Based Collaborative Filtering*
  - *Model-Based Collaborative Filtering*

Evaluation methods for recommender systems can mainly be divided into two sets:

- The evaluation is based on well-defined metrics
  - If the recommender system is based on a model that outputs numeric values such as rating predictions: ***RMSE, MAE etc***
- The evaluation is mainly based on human judgment and satisfaction estimation.
  - If the recommender system is not based on numeric values and only returns a list of recommendations

# Recommender System - *Data Filtering*

- The data is considered for filtering in order to:
  - Reduce the dimensionality of the dataset and avoid running into memory error.
  - To bring statistical significance.
  - To access relevant information required to make recommendation.

- Data filtering algorithms define a threshold/cutoff on one or more characteristics.
  Ex:
  - Define a popularity threshold/ percentile cutoff
  - Define threshold on multiple features: considering the users with at least "*user_threshold*" ratings and books with at least "*ratings_threshold*" ratings.

    ***"user_threshold" = 60 ; "ratings_threshold" =10***

# Recommender System - Popularity

- It's a kind of recommendation system that works on the principle of popularity and/or whatever is trending.
- Example: The most viewed article for a website, the most popular movie for Netflix, the most sold items for Amazon, the most trending videos for YouTube, etc.
- Youtube studio also uses this system to show the most popular videos in the last 28 days.

Advantages: *Simplicity, less computational usage and easy to keep updated.*
Disadvantages: *Lack of personalization, and poor accuracy in recommendations, implying fewer profits generated.*

# Recommender System - Popularity

- The recommendation based on popularity is done by considering the ***Average Rating*** of the book.
- Initially, data is filtered by considering the ***Popularity Threshold***- Minimum number of ratings for a book to be considered for the recommendation.
- The resulting list of books is sorted in decreasing fashion based on the ***Average Rating***.


| | Book-Title | Book-Author | Publisher | Average-Rating | Book-Rating-Count |
|---|---|---|---|---|---|
| 0 | Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback)) | J. K. Rowling | Arthur A. Levine Books | 8.939297 | 313 |
| 1 | The Secret Life of Bees | Sue Monk Kidd | Penguin Books | 8.452769 | 307 |
| 2 | The Da Vinci Code | Dan Brown | Doubleday | 8.435318 | 487 |
| 3 | The Lovely Bones: A Novel | Alice Sebold | Little, Brown | 8.185290 | 707 |
| 4 | The Red Tent (Bestselling Backlist) | Anita Diamant | Picador USA | 8.182768 | 383 |

***Popularity Threshold: 300***

# Recommender System - Weighted Average

- The recommendation index used for our books dataset is ***Weighted Average Rating***.
  It is one of the types of popularity recommendation.

- For the calculation of threshold(m), the *90th percentile* is used as the cutoff.
- The weighted average score is calculated for each book and recommendations are done based on the highest scores.
- This approach is not sensitive to the interests and tastes of a particular user.

$$W = \frac{Rv + Cm}{v + m}$$

*Where,*

*W is Weighted Average*

*v is the number of ratings for the book*

*m is the threshold- the minimum number of ratings required to be listed in the chart*

*R is the average rating of the book*

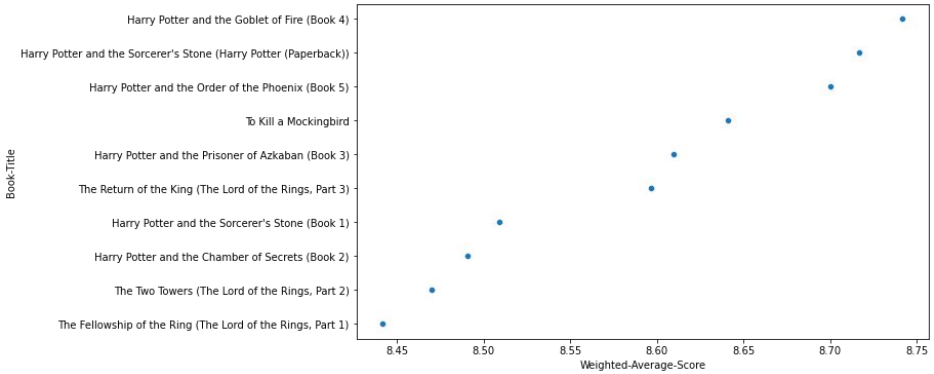
*C is the mean average ratings across the whole dataset- mean(R)*

# Recommender System - Weighted Average

### *Recommendations based on "Weighted Average Book Rating"*

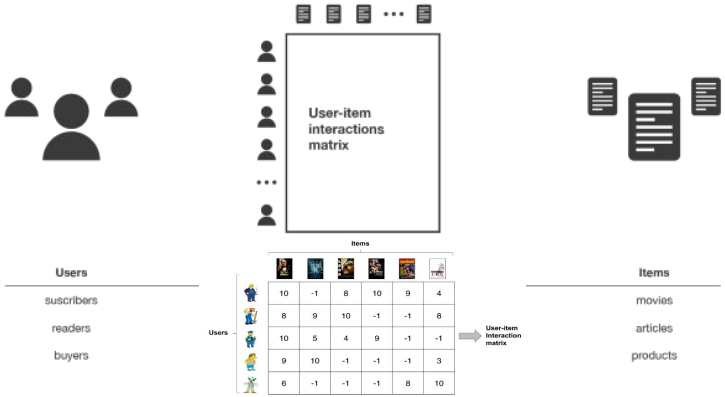
| | Book-Title | Book-Author | Publisher | Average-Rating | Book-Rating-Count | Weighted-Average-Score |
|---|---|---|---|---|---|---|
| 0 | Harry Potter and the Goblet of Fire (Book 4) | J. K. Rowling | Scholastic | 9.262774 | 137 | 8.741835 |
| 1 | Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback)) | J. K. Rowling | Arthur A. Levine Books | 8.939297 | 313 | 8.716469 |
| 2 | Harry Potter and the Order of the Phoenix (Book 5) | J. K. Rowling | Scholastic | 9.033981 | 206 | 8.700403 |
| 3 | To Kill a Mockingbird | Harper Lee | Little Brown & Company | 8.943925 | 214 | 8.640679 |
| 4 | Harry Potter and the Prisoner of Azkaban (Book 3) | J. K. Rowling | Scholastic | 9.082707 | 133 | 8.609690 |

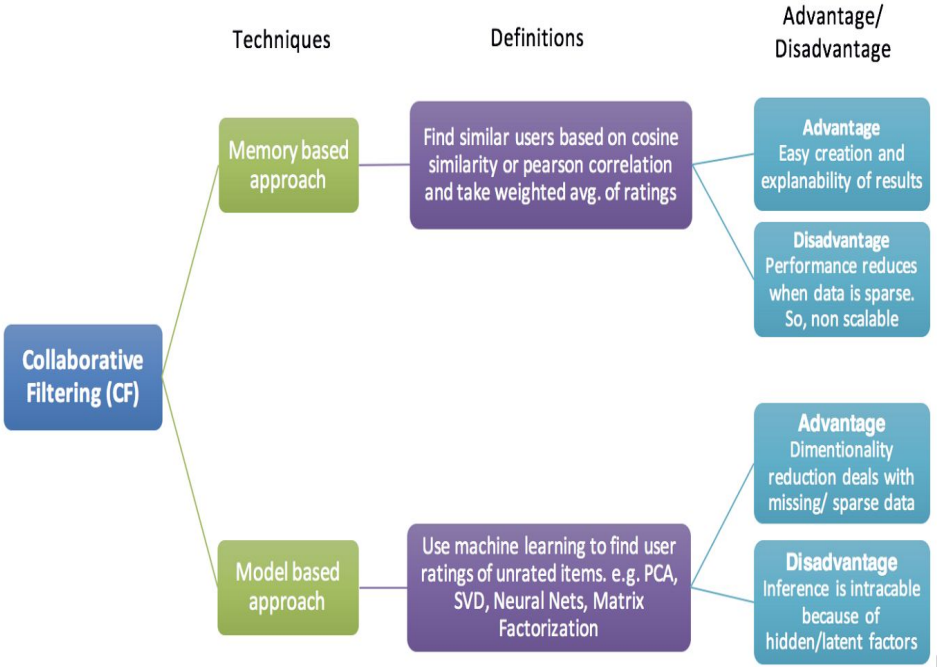

### *Top 10 recommendations and their scores*



# Recommender System - Collaborative Filtering

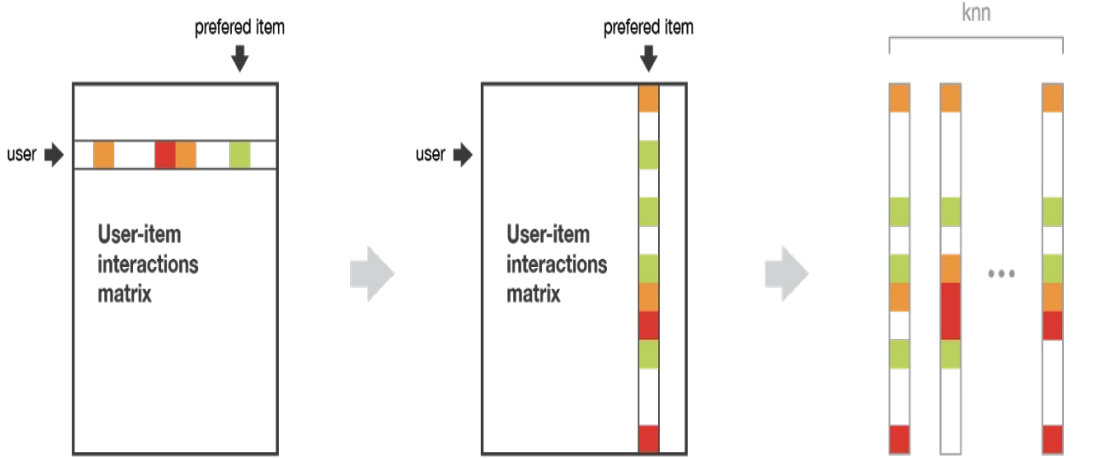
- Collaborative methods for recommender systems are based on the past interactions recorded between users and items in order to produce new recommendations.
- These interactions are stored in the so-called "***user-item interactions matrix***".



# Recommender System - Collaborative Filtering

## *Item-Item Memory Based Collaborative Filtering*

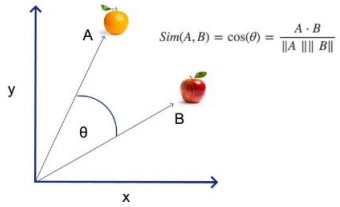



We identify the prefered item of user we want to make recommendation for.

The prefered item is represented by its column in the matrix.

We can search and recommend the K nearest items to this "prefered item"

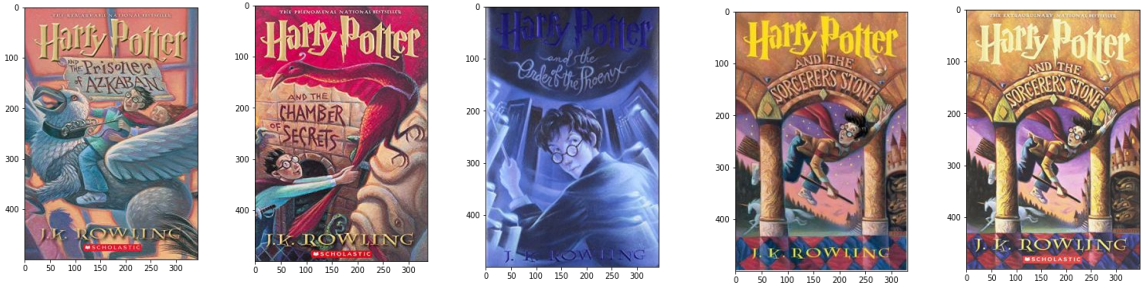
**Cosine Similarity**




# Collaborative Filtering- *Item-Item Memory Based Collaborative Filtering*

**Top 5 recommended books to user who prefers "*Harry Potter and The Goblet Of Fire*"**

| | Book-Title | Book-Author | Publisher | Average-Rating | Book-Rating-Count | Similarity-Score |
|---|---|---|---|---|---|---|
| 0 | Harry Potter and the Prisoner of Azkaban (Book 3) | J. K. Rowling | Scholastic | 9.082707 | 133 | 0.365923 |
| 1 | Harry Potter and the Chamber of Secrets (Book 2) | J. K. Rowling | Scholastic | 8.783069 | 189 | 0.365923 |
| 2 | Harry Potter and the Order of the Phoenix (Book 5) | J. K. Rowling | Scholastic | 9.033981 | 206 | 0.365923 |
| 3 | Harry Potter and the Sorcerer's Stone (Book 1) | J. K. Rowling | Scholastic | 8.983193 | 119 | 0.365923 |
| 4 | Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback)) | J. K. Rowling | Arthur A. Levine Books | 8.939297 | 313 | 0.365923 |



# Recommender System - Collaborative Filtering

## Item-Item Memory Based Collaborative Filtering

Memory-based collaborative filtering approaches that compute distance relationships between items or users have these two major issues:

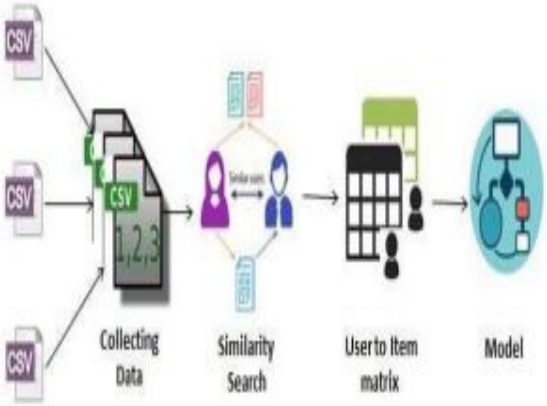
- It doesn't scale particularly well to massive datasets, especially for real-time recommendations based on user behavior similarities which take a lot of computations.
- Rating matrices may be overfitting to noisy representations of user tastes and preferences.

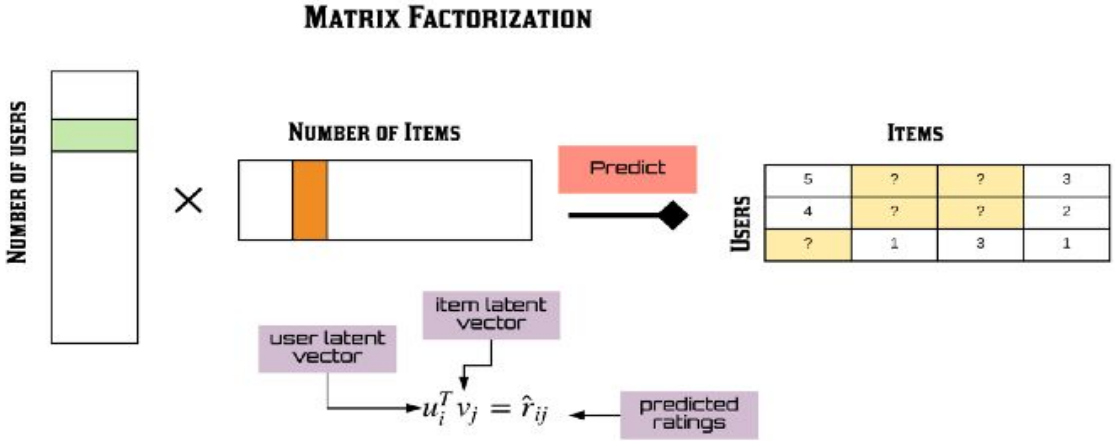
Therefore to look for potential benefits of both *speed* and *scalability*, ***model-based collaborative filtering*** is investigated.

# Recommender System - Collaborative Filtering

## Model Based Collaborative Filtering

- Involves building a model based on the dataset of ratings.
- *Latent factor methods* explain the ratings by characterizing both items and users on many factors inferred from the rating pattern.



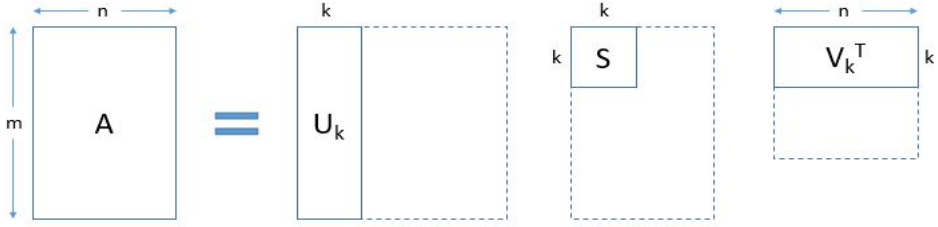



# Collaborative Filtering - Model Based Collaborative Filtering

## Singular Value Decomposition (SVD)

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 finds factors of matrices from the factorization of a high-level (user-item-rating) matrix.




$$A = USV^T$$

Where,

*A* is a *m x n* utility matrix,

*U* is a *m x k* orthogonal left singular matrix, which represents the relationship between users and latent factors,

*S* is a *k x k* diagonal matrix, which describes the strength of each latent factor.

*V* is a *k x n* diagonal right singular matrix, which indicates the similarity between items and latent factors.

# Collaborative Filtering- Model Based Collaborative Filtering- Predictions

***Books that have been rated by user-1424***

| | Book-Title | Book-Author | Publisher | Book-Rating | Average-Rating | Book-Rating-Count |
|---|---|---|---|---|---|---|
| 0 | A Walk in the Woods: Rediscovering America on the Appalachian Trail (Official Guides to the Appalachian Trail) | Bill Bryson | Broadway Books | 8 | 8.207547 | 106 |
| 1 | Prey | Michael Crichton | Avon Books | 8 | 7.571429 | 77 |
| 2 | A Map of the World | Jane Hamilton | Anchor Books/Doubleday | 7 | 7.000000 | 111 |
| 3 | The Poisonwood Bible | Barbara Kingsolver | HarperTorch | 7 | 8.264706 | 34 |
| 4 | The Name of the Rose: including Postscript to the Name of the Rose | Umberto Eco | Harvest Books | 8 | 8.523810 | 21 |
| 5 | The Joy Luck Club | Amy Tan | Prentice Hall (K-12) | 6 | 8.195876 | 194 |
| 6 | Plain Truth | Jodi Picoult | Washington Square Press | 8 | 8.148936 | 47 |
| 7 | The Poisonwood Bible: A Novel | Barbara Kingsolver | Perennial | 7 | 8.178899 | 218 |
| 8 | The Bean Trees | Barbara Kingsolver | HarperTorch | 5 | 7.861111 | 72 |
| 9 | Memoirs of a Geisha Uk | Arthur Golden | Trafalgar Square | 8 | 8.174419 | 86 |
| 10 | Year of Wonders | Geraldine Brooks | Penguin Books | 7 | 8.318182 | 88 |


***Recommendations to user 1424***

| | Book-Title | Book-Author | Publisher | Average-Rating | Book-Rating-Count |
|---|---|---|---|---|---|
| 0 | 1984 | George Orwell | Signet Book | 8.772277 | 101 |
| 1 | 1st to Die: A Novel | James Patterson | Little Brown and Company | 7.661017 | 59 |
| 2 | 2010: Odyssey Two | Arthur C. Clarke | Del Rey Books | 7.413793 | 29 |
| 3 | 2061: Odyssey Three | Arthur C. Clarke | Del Rey Books | 7.666667 | 18 |
| 4 | 2nd Chance | James Patterson | Warner Vision | 7.722222 | 90 |

***USER-ID: 1424***

# Recommender System - Collaborative Filtering

## Model Based Collaborative Filtering - Results

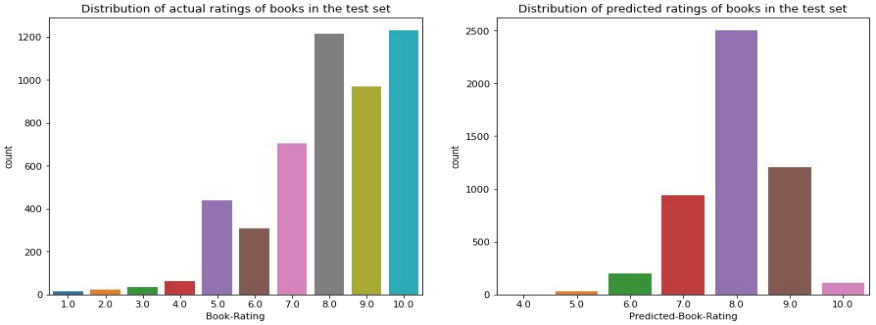
The predictions are based on the ***SVD*** model used for evaluating the model.

- The SVD model is hyperparameter tuned to obtain the ***RMSE value of 1.50***

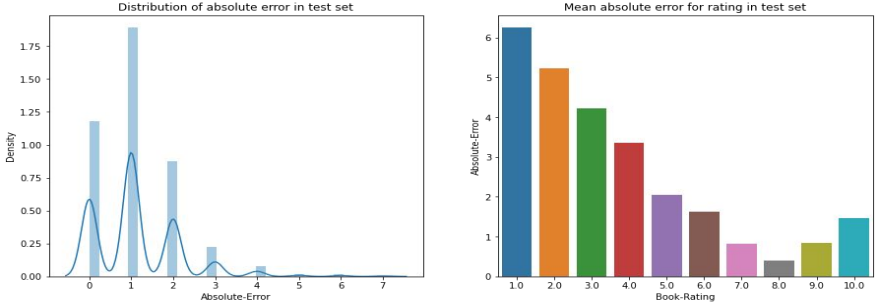
*"Book-Rating" predictions for **User 1424**:*

- *'The Poisonwood Bible' ( Already Rated)*
  - Estimated rating: **7.38**
  - Actual rating: **7**
- *'1st to Die: A Novel' ( Unseen Book)*
  - Estimated rating: **7.67**

### *Actual and predicted "Book-Rating" Plot*




### *"Absolute Error" distribution*



# Cold-Start Problem

***Cold start*** is a potential problem in computer-based information systems which involves a degree of automated data modeling. There are three instances of a cold start:

1. *New community* 2. *New item* 3. *New user*

***collaborative filtering suffers from the "cold start problem".***

*Possible solutions :*

- Random strategy
- Maximum expectation strategy
- Exploratory strategy
- Hybrid Strategy

# Conclusion

- Different types of recommenders are explored in order to recommend books based on the "*Book- Crossing*" dataset.
- The exploratory data analysis gave a basis for hypothesis before going for any data modelling.
- Popularity-based recommendation is simple to implement but suffers from lack of personalization, and doesn't read the user interests.
- Recommendation system based on weighted average is yet another variety of popularity-based recommenders that considers score based on average rating.
- Collaborative filtering recommenders provide larger flexibility, scalability in terms of personalization of recommendations.
- The limitations in memory-based approach like, overfitting, not scalable to real-life large data etc. are overcome by the model-based collaborative filtering resulting in ***RMSE: 1.50***
- The hybrid of popularity-based recommenders with collaborative filtering recommenders can be considered for ultimate performance.

# Thank you