

# BookAdda

# Book Recommendation System



By

**Foram Jivani(201711012)**  
**Smriti Changulani(201711019)**  
**Dhruva Shah(201711024)**  
**Mausamee Patel(201711056)**

**Guided By**  
**Prof. Saurish Das Gupta**

## INTRODUCTION

We have developed book recommendation system as part of our course project. Virtually everyone has had an online experience where a website makes personalized recommendations in hopes of future sales or ongoing traffic. Here we have tried to give recommendation of books based on users' past issues by using collaborative and content based filtering and as well as users' current action on particular book by using some aspects of reinforcement learning. Hence we have used hybrid model for our system to recommend books to users.

## Decision Parameters

For our current prototype we have considered following decision parameters:

- User Ratings
- User Likes/Dislikes
- Book Author
- Book Genre
- Book's Average Global Rating
- Current User Action

Recommender system is defined as a decision making strategy for users under complex information environments. These parameters are important part of different techniques for recommendation system.

## Model

We have built prototype based on hybrid model which includes combination of following approaches:

1. Popularity Based

2. Item - Item Collaborative Filtering
3. User Based filtering
4. Genre & Author based content based filtering

#### 1. Popularity Based Filtering

In this approach we have considered average global rating for book for recommending top 5 books. Here we are keeping our database updated.

#### 2. Item - Item Based Collaborative Filtering

Item-item collaborative filtering is a form of collaborative filtering for recommender systems based on the similarity between items calculated using people's ratings of those items. Item-item models resolve these problems in systems that have more users than items. Item-item models use rating distributions per item, not per user. This provides more stable rating distributions in the model, so the model doesn't have to be rebuilt as often. So in our model we calculate similarity based on item once for training then we use it over entire user session so time complexity is not much. When users consume and then rate an item, that item's similar items are picked from the existing system model and added to the user's recommendations.

#### 3. User Based Collaborative Filtering

1. Look for users who share the same rating patterns with the active user (the user whom the prediction is for).
2. Use the ratings from those like-minded users found in step 1 to calculate a prediction for the active user

#### 4. Genre and Author Content based Collaborative filtering

In this approach we are finding similarities between books by genre and author alone as well as in combination. We are first converting genre and authors of the books into tokens and then calculate similarities. Moreover, we have also tried to capture user's favourite genre and author for current session, that is what user wants or is searching during current session.

## Learning Algorithm

1. Existing User can Log in while new user can register into system
2. User can take following actions:
  - a. Give rating (0-5)
  - b. Like / Dislike item (like is considered as rating 5 while dislike is as 0)
  - c. View Item (that is considered as Click (not included in Prototype))
3. For existing user, apply SVD to show the top 10 items and top 5 popular items.
4. When user take any action, then apply following algorithms based on given ratings
  - i. Item-Item Based Similarity
  - ii. Genre-Author Based Similarity
  - iii. Popularity Based model.
5. Based on past and current ratings, show New Recommendation list of 10 items
6. New Dialog box will be prompted if want new recommendation or want to exit
7. If user takes any action repeat step (4) - (6) until logout. After Logout save the current session ratings into rating file.

## DATASET

- We have used following dataset files:
  - Ratings\_test\_900.csv
  - books\_test\_SVD1\_900.csv
- Which consists of
  - #of Users : 7000
  - #of books : 1000
  - #of Ratings : 11400 (updated at runtime)

## RESULTS

Results slides include snapshots of the prototype system that we have built by including each steps. When user will login at that time for first recommendation iteration, system will show items based on SVD. If user rating information was not present then it will

show items based on Popularity Model. After that user will perform actions on recommended items. Hence system will start to learn user's mood and will start to recommend items based on Item Item similarity, Genre and Author based similarity as well as user based similarity to incorporate serendipity. When user logs out all of his/her rating information will be stored in rating files so that when user again logs in its past data will be there.

What if New User:

If user is new to system then we are asking user to enter some of his/her favourite genre. So that when he/ she next time login, system can make use of that information to recommend items that are relevant to user. As well as Popularity based model will be used to recommend items.

What if New Item:

As we are using genre based content filtering, new item will be shown to user if it comes under user's favorite genres. If it is popular then it will be shown under most popular list.

## CONCLUSION

Hence, our system uses hybrid model with using some concepts of reinforcement learning. In sense that our system is trying to learn current user's mood with reinforcement learning.