# Information Retrieval

# Book Recommendation System

**Group 56**

Anupam Garg(2020555)
Saharsh Dev(2020572)
Vanisha Singh(2020347)
Vanshika Goel(2020413)
Vishal Kumar(2020154)
Saksham Singhi(2020463)

# PROBLEM STATEMENT

In the current digital era, readers are faced with an immense number of books to choose from, making it difficult for them to navigate and identify books that fit their interests, preferences, and past reading experiences.

A book recommendation system will personalize, increase efficiency and engagement and diversify the range of books.

# LITERATURE SURVEY

- In 2010 Choi et.al. proposed RS based on HYRED, a hybrid algorithm using both content and collaborative filtering they used altered Pearson Coefficient based Collaborative filtering and distance-to-boundary (DTB) Content filtering.
- Kurmashov et al. in their paper tried to propose a book recommendation service by taking inputs of the preference like the most favorite genre and ratings on different books to further narrow down the space of search for recommendations when a user logs into the system.
- In 2016 Mathew et.al. proposed a system that saves details of books purchased by the user. From these Book contents and ratings, a hybrid algorithm using collaborative filtering, content-based filtering, and association rule generates book recommendations.
- In 2021, Sarma, Mittra, Hossain proposed a clustering-based book recommendation system that uses different approaches, including collaborative, hybrid, content-based, knowledge-based, and utility-based filtering.

# DATABASE

We have used two datasets in our project.

## Goodreads 2M Dataset

It is used for content based filtering. It contains various information such as book name, authors, rating, publish year, publisher, counts of reviews, language, pages number, and description. It has books in different languages, including English, Spanish, French, etc.

## Final Users Dataset

It is used for collaborative based filtering. It contains information on users' ratings of books. It has columns for the user ID, book name, book ID, rating and contains 11,057 unique users and 3,534 unique book IDs. This dataset is generated by the group members for this project.

# PROPOSED SOLUTION

We propose a recommendation engine that tries to alleviate some of the problems that each system encounters and it combines the user's choice with the wider community to generate relevant outcomes.

Our system gives the most suitable recommendations based on the estimated ratings that the user might give to that book calculated using collaborative filtering, and use these readings as a measure to filter the most similar books as per the user preferences in the past history.

# NOVELTY IN OUR PROPOSED SOLUTION

The novelty lies in the evaluation of content-based filtering, which is done using the estimated ratings generated from collaborative filtering. Using these calculated ratings (a user might give to each book), we defined our true positives of CB recommendations as those books which have their rating greater than the average rating that the user has given.

We are also integrating popularity-based approach with the hybrid model which will solve the cold start problem for the new users.

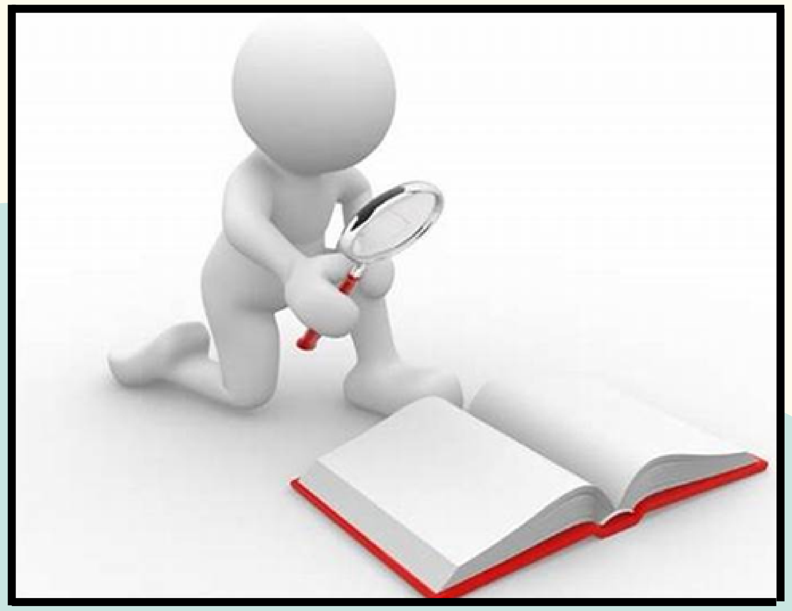
# INTEGRATING HYBRID MODEL WITH POPULARITY BASED APPROACH

Popularity-based approach checks those books which are the most popular among the users and could be recommended.

Though it might not be sensitive to the interests and tastes of a particular user, this model doesn't suffer with the cold start problems. It can recommend the products on various different filters and there is absolutely no need for the user’s historical data.

# FEATURES IN OUR SOLUTION

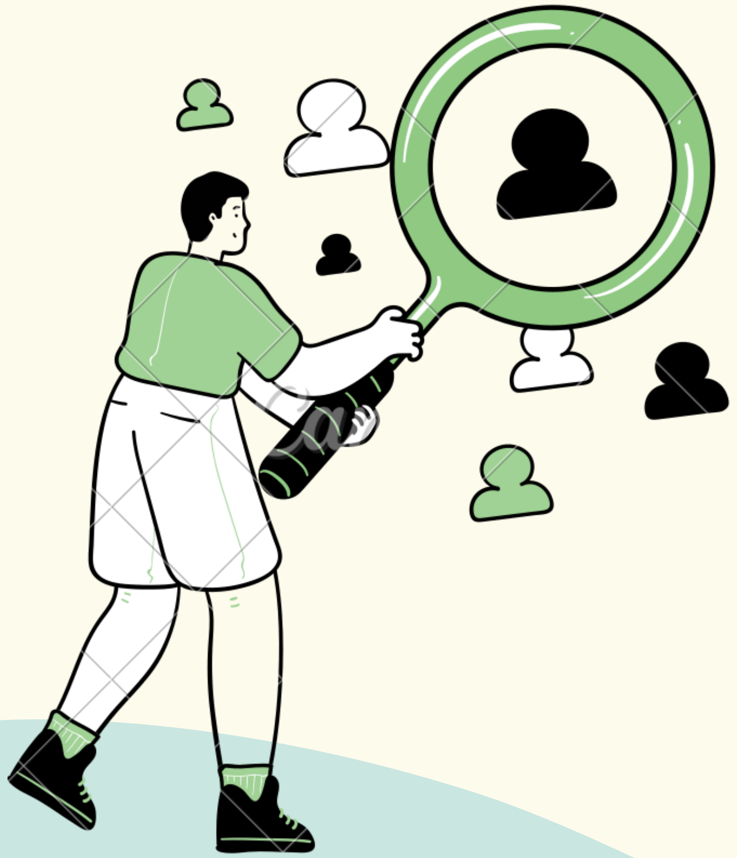
Problems of Sparsity and cold start can be resolved with these methods.



Based on a user's query, the system can recommend using the most relevant similar entities from a large dataset. This enhances the scalability of the system.

Provides qualitative recommendations to users automatically based on their past preferences without asking for more information.

# FEATURES IN OUR SOLUTION

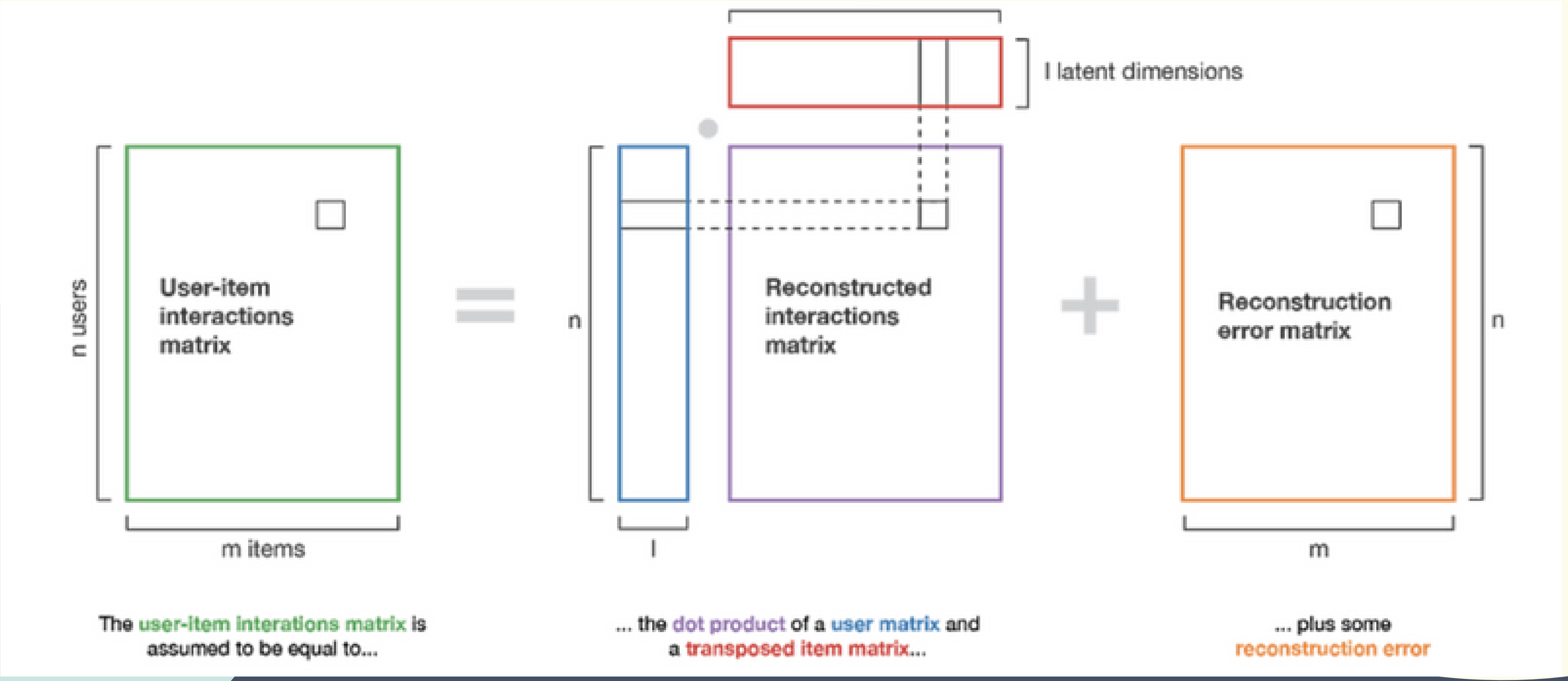


We are providing a section for books that are newly released so that the users can explore them. Also, using this we are solving the problem of recommendation of books that are not been read and rated by people.

For new users, we are recommending the most popular books by using the popularity based approach from the database so that user can find a suitable book.

# METHODS

- **Collaborative Filtering**: It suggests books to users based on the similarity between the user's preferences and the preferences of other similar users. By evaluating the user similarities based on their ratings and the inter-user comparisons, books are recommended to the user.



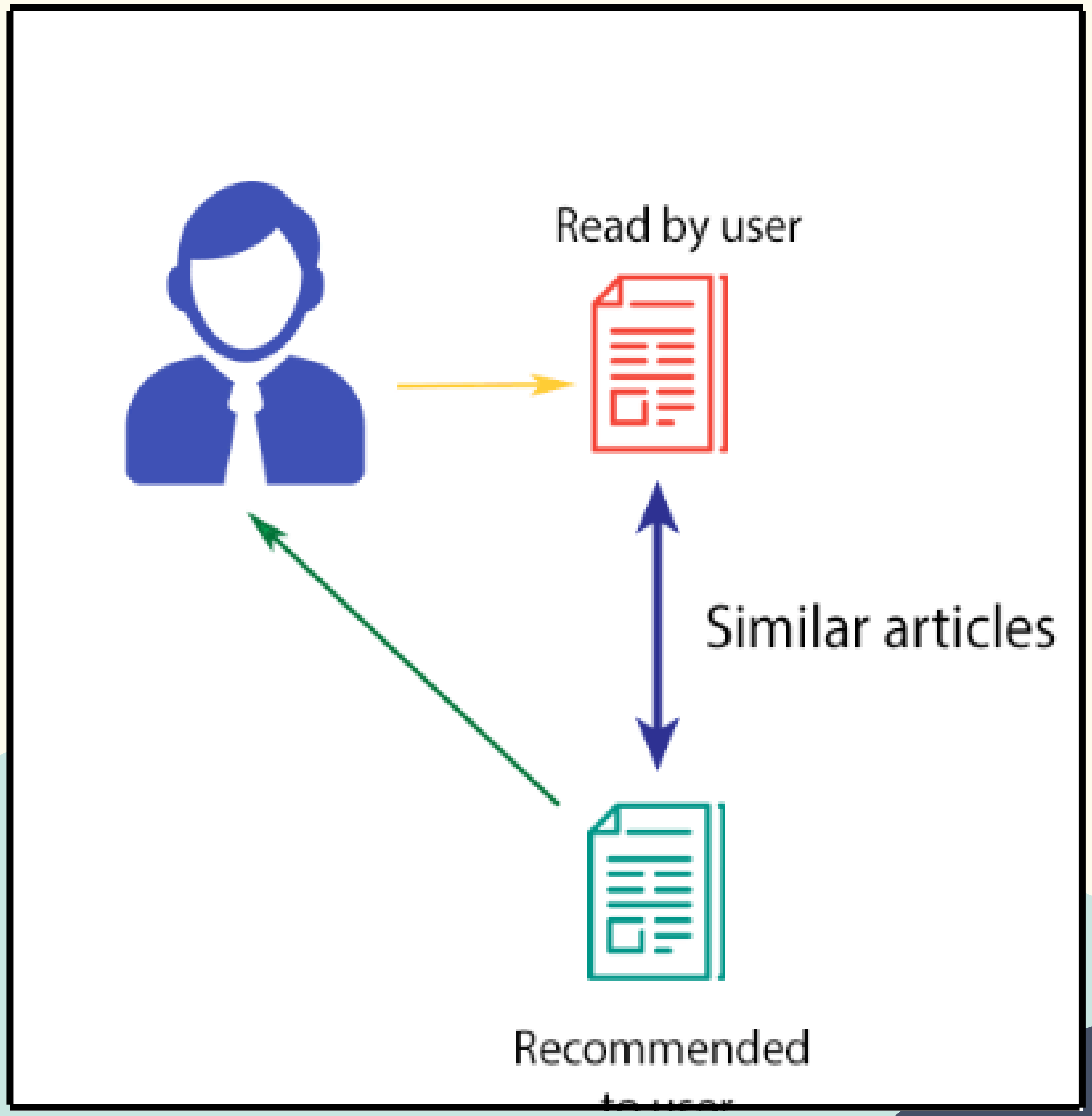
l latent dimensions
n users
User-item interactions matrix
m items
n
Reconstructed interactions matrix
l
Reconstruction error matrix
n
m

The user-item interations matrix is assumed to be equal to...

... the dot product of a user matrix and a transposed item matrix...

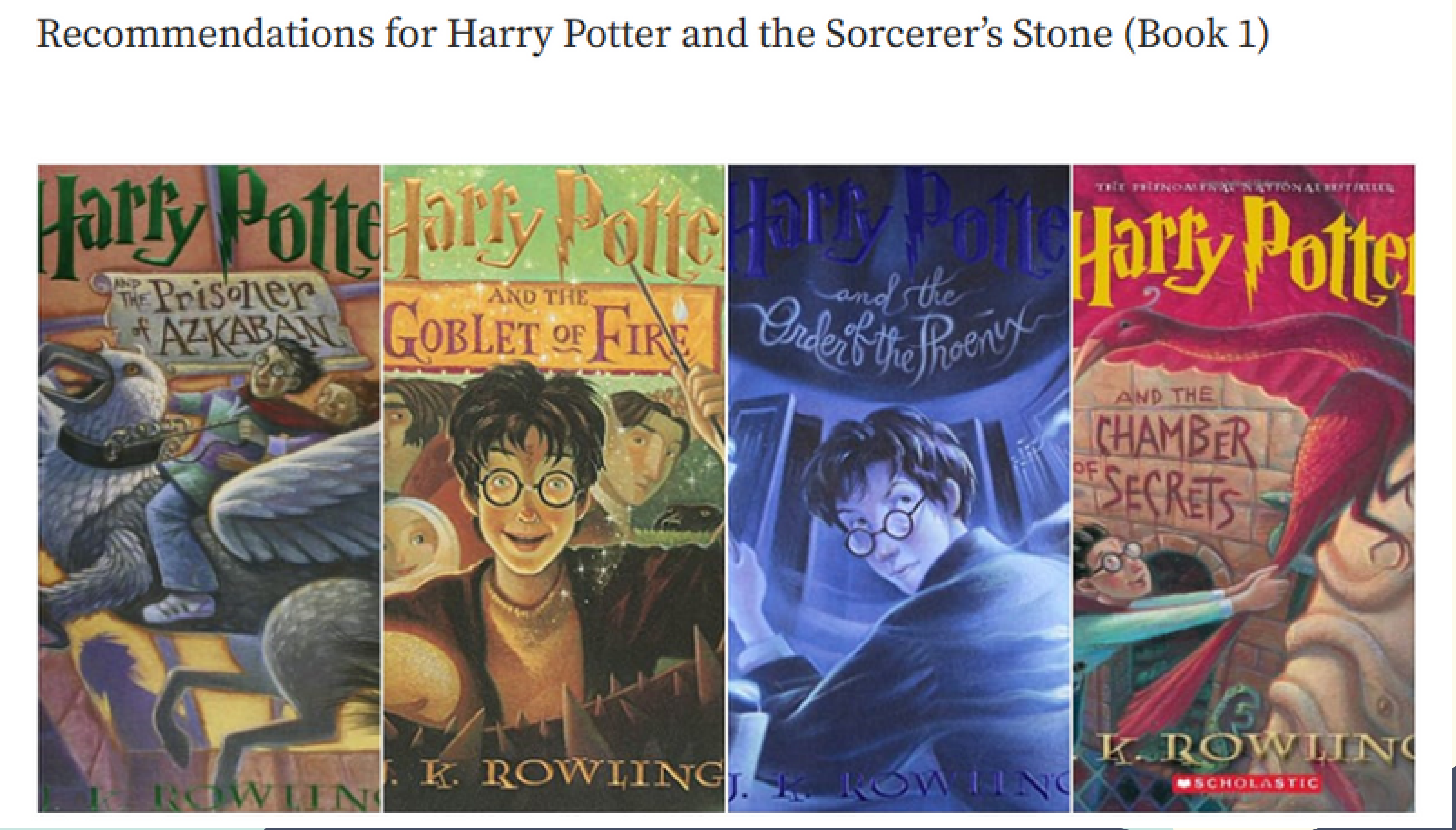
... plus some reconstruction error

- **Content-based filtering**: It helps to suggest books similar to the books which the user prefers, and the characteristics of these books. By calculating a similarity score (using cosine similarity) between the user's books and the other books in the database, the most similar books are filtered and suggested to the user.



- It computes similarities between the books liked by the user with all the other books and creates a TF-IDF vector of each Book Description.  This vector is used to calculate the cosine similarity between the books. The higher value of cosine similarity between the two vectors will determine the most similar books.

# Model – testing using KNN Algorithm



Recommendations for Harry Potter and the Sorcerer's Stone (Book 1)

Harry Potter and the Prisoner of Azkaban
J. K. Rowling
Harry Potter and the Goblet of Fire
J. K. Rowling
Harry Potter and the Order of the Phoenix
J. K. Rowling
The phenomenal national bestseller
Harry Potter and the Chamber of Secrets
J. K. Rowling
Scholastic

# LIMITATIONS

Some of the limitations of using these filtering techniques is that it is difficult to create item characteristics in some areas.

CB method recommends the same types of items because of that it suffers from an overspecialization problem.

The CF method can become computationally expensive as the number of users and items in the system grows.

CF systems often require a huge amount of existing data on the user, to make exact recommendations which lead to the Cold Start problem.

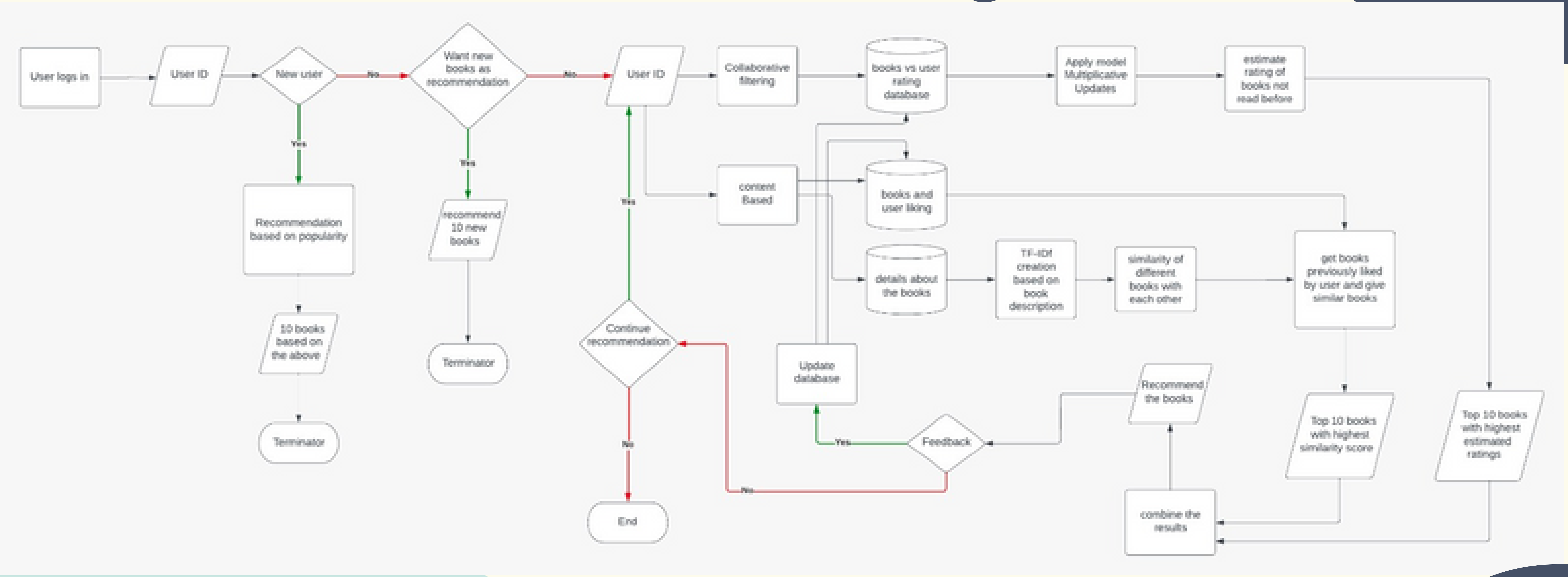
# HYBRID FILTERING

Implements content and collaborative-based methods separately and aggregates their prediction.

Uses different techniques to combine the results like weighted averaging, cascading, or switching to deal with the sparsity and cold start problem.

We used the weighted averaging technique, in which 1/3rd of the recommended books were from CF approach and rest were from CB approach.

# WORK FLOW


User logs in
User ID
New user
No
Want new books as recommendation
No
User ID
Collaborative filtering
books vs user rating database
Apply model Multiplicative Updates
estimate rating of books not read before
Yes
Recommendation based on popularity
10 books based on the above
Terminator
Yes
recommend 10 new books
Terminator
Yes
content Based
books and user liking
details about the books
TF-IDf creation based on book description
similarity of different books with each other
get books previously liked by user and give similar books
Continue recommendation
Update database
Recommend the books
Top 10 books with highest similarity score
Top 10 books with highest estimated ratings
Yes
Feedback
No
No
End
combine the results

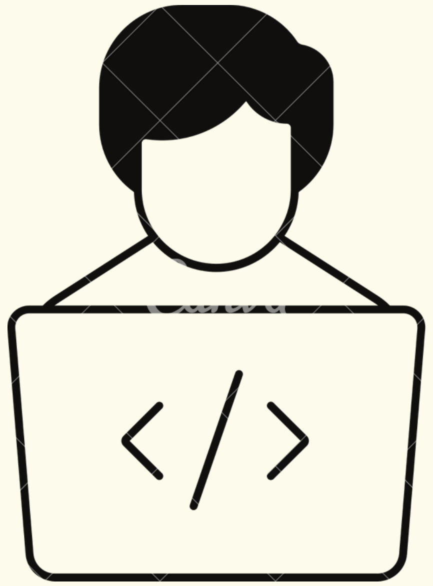
# CODE

For the content-based filtering, we first did the preprocessing of the dataset and then made the TF-IDF vector of the book descriptions, and then using those vectors we calculated the cosine similarity of each book description to another. Then taking the previously read books of the user which they liked, we recommended the books in order of highest cosine similarity values with those of previously read books.

# CODE

For the collaborative-based filtering, we first created the user vs books matrix of book ratings using the dataset and assigned zero to the missing values. Then we divided it into a ratio of 85:15 of train & test data. Then we integrated the Multiplicative Updates algorithm for matrix factorization. We used latent factors as 70 and calculated the normalised mean absolute error (NMAE).

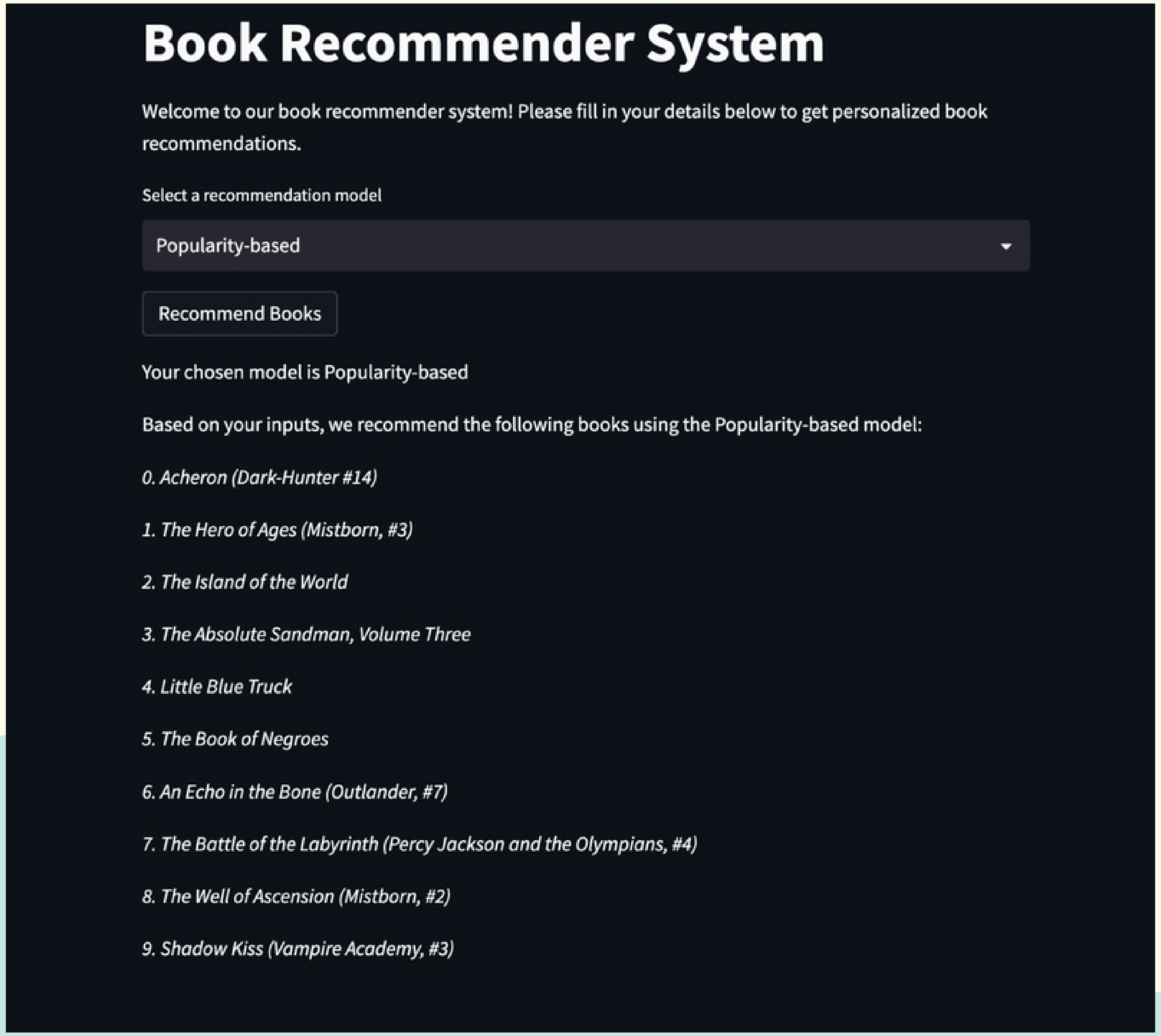
# CODE



To create a hybrid model, we integrated content and collaborative-based filtering. In our hybrid filtering, we first give the highest priority to books (x) that are common in both content and collaborative-based filtering. Then, as we are proposing a total of 10 books, we divided 10-x into 2:1 ratios of content and collaborative recommendations respectively to finally show the hybrid recommendations.

In the popularity based approach, we used Bayesian average ranking system. In this we used no of reviews of each book and the ranking of the book into account.

# WEBSITE



## Book Recommender System

Welcome to our book recommender system! Please fill in your details below to get personalized book recommendations.

Select a recommendation model

Popularity-based

Your chosen model is Popularity-based

Based on your inputs, we recommend the following books using the Popularity-based model:

*0. Acheron (Dark-Hunter #14)*

*1. The Hero of Ages (Mistborn, #3)*

*2. The Island of the World*

*3. The Absolute Sandman, Volume Three*

*4. Little Blue Truck*

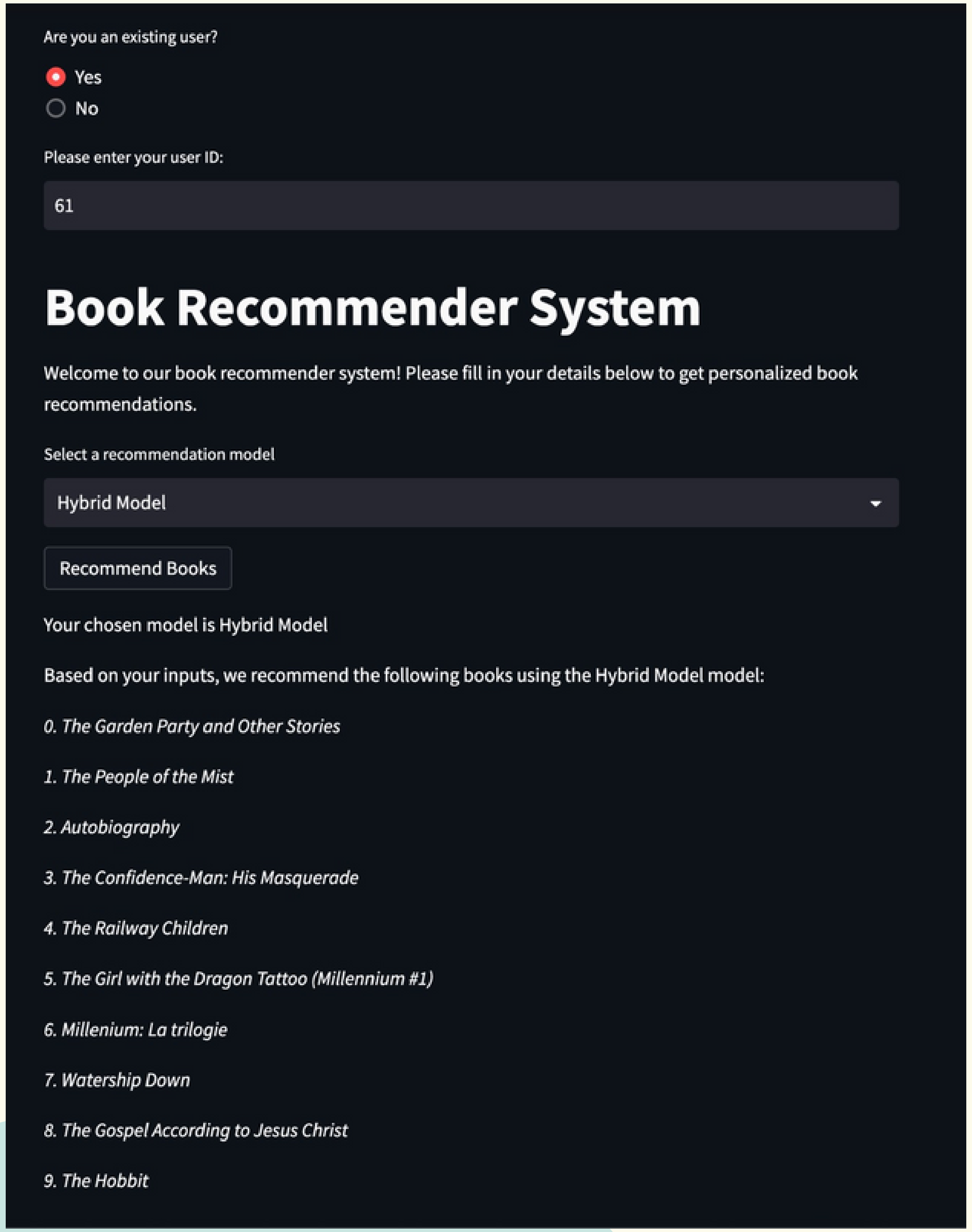
*5. The Book of Negroes*

*6. An Echo in the Bone (Outlander, #7)*

*7. The Battle of the Labyrinth (Percy Jackson and the Olympians, #4)*

*8. The Well of Ascension (Mistborn, #2)*

*9. Shadow Kiss (Vampire Academy, #3)*



Are you an existing user?

● Yes
○ No

Please enter your user ID:

61

## Book Recommender System

Welcome to our book recommender system! Please fill in your details below to get personalized book recommendations.

Select a recommendation model

Hybrid Model

Your chosen model is Hybrid Model

Based on your inputs, we recommend the following books using the Hybrid Model model:

*0. The Garden Party and Other Stories*

*1. The People of the Mist*

*2. Autobiography*

*3. The Confidence-Man: His Masquerade*

*4. The Railway Children*

*5. The Girl with the Dragon Tattoo (Millennium #1)*

*6. Millenium: La trilogie*

*7. Watership Down*

*8. The Gospel According to Jesus Christ*

*9. The Hobbit*

# EVALUATION

## Comparison with baselines and the system's performance on existing data

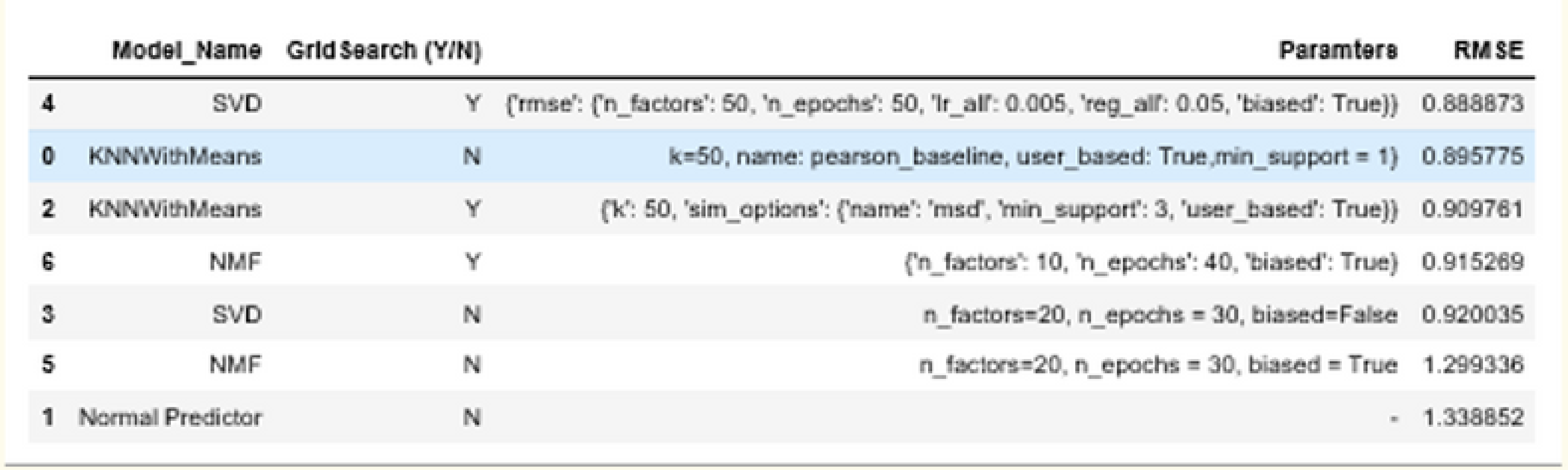
For the baseline, we implemented the Collaborative filtering approach, particularly a Model-based approach where we used a kNN model and cosine similarity matrix, to generate the results. We returned the top 6 nearest books for the recommendations. We used the Normalised Mean Absolute Error as a metric for testing our model performance, where we found that the error value was quite large for the kNN model (NMAE = 0.763) as compared to our own implemented collaborative technique, where we have used deep latent factor models like Matrix Factorization and Multiplicative Updates to obtain the predicted values for the item ratings.

Using this model, our loss value has been significantly reduced (NMAE = 0.204). The system only displayed the books which were liked by the other users who were similar. The system was not able to provide more personalized recommendations to the user as there were no criteria to compare the books which the user has already reviewed.

# EVALUATION

## SOTA on different Evaluation Metrics

The RMSE value achieved from our collaborative filtering model is 1.04044, which beats 2 state of the art models as shown in the picture below.



|  | Model_Name | Grid Search (Y/N) | Paramters | RMSE |
|---|---|---|---|---|
| 4 | SVD | Y | {'rmse': {'n_factors': 50, 'n_epochs': 50, 'lr_all': 0.005, 'reg_all': 0.05, 'biased': True}} | 0.888873 |
| 0 | KNNWithMeans | N | k=50, name: pearson_baseline, user_based: True,min_support = 1} | 0.895775 |
| 2 | KNNWithMeans | Y | {'k': 50, 'sim_options': {'name': 'msd', 'min_support': 3, 'user_based': True}} | 0.909761 |
| 6 | NMF | Y | {'n_factors': 10, 'n_epochs': 40, 'biased': True} | 0.915269 |
| 3 | SVD | N | n_factors=20, n_epochs = 30, biased=False | 0.920035 |
| 5 | NMF | N | n_factors=20, n_epochs = 30, biased = True | 1.299336 |
| 1 | Normal Predictor | N | - | 1.338852 |

The results are from the paper published on Sep 1 2019.
The NMAE value of the collaborative filtering model came out to be 0.207 and the accuracy of our content-based model using the values from the collaborative filtering model came out to be 96%.

# EVALUATION

**How the system performs on new data / handles different cases**

For new users who have not yet established their preferences, we are using popularity-based system that recommends books based on their popularity, which provides a starting point for users to explore different types of books before our system begins to personalize recommendations based on their individual preferences.

For new books, we are providing a section for books that are newly released so that the users can explore them. Also, using this we are solving the problem of recommendation of books that are not been read and rated by people.

# FUTURE WORK

We can suggest books on the basis of the language

We can modify our dataset in order to integrate the genre of the book and then recommend based on genre

We can also use the data about average number of pages user reads and suggest them books that are close to amount of pages he usually reads

# THANKS!

DO YOU HAVE ANY QUESTIONS?