Project Proposal

We will use the Hybrid Collaborative-Content models as our recommendation system method. This method combines the three filtering techniques which are content-based, collaborative, and knowledge-based. By combining all methods, a hybrid model will be able to provide more accurate and diverse recommendations. This system will recommend books that are either close or similar to the users.

In our project, we will use the **Book Recommendation dataset** sourced from Kaggle, consisting of two files containing Ratings and Books. For the initial step, we will preprocess the data that includes Tokenization, stop words removal, lemmatization and stemming and vectorization. After that, we will conduct training and testing the data using our Content-based and Collaborative based filtering methods.

Content-based filtering recommends books to the user based on the features such as Publication Year, Author, Book Title. For the first time users, the system will recommend books based on the ratings or the user can search a book using the books' features such as Publication Year, Author, Book Title. For the returning users, the system uses their previous history to recommend books and additionally, they can get the same privileges as the first time users.
In the Content-based implementation, we will use the TF-IDF algorithm for text analysis on textual features like Book Titles get the similarities among books in the dataset to recommend to the user using the TF-IDF scores. The system also uses the weight score factor to recommend books to the user if they prefer recent books.

Collaborative based filtering is based solely on user behavior only, and recommends products to a user based on the preferences of other users. We will utilize the 'Books' and 'Ratings' file for this approach, as the 'Users' file only contains user_id, location and age, which are not useful for collaborative filtering.To implement this approach, we will use Matrix Factorization to create a utility matrix where columns represent users and rows represent rated items.

However, before proceeding, we need to preprocess the data, and clean up the rating file to only include user IDs with more than 150 ratings. Then, we will merge this table with the 'Books' file, and only retain the records of books that have received more than 50 ratings. Once this is done, we can finally create our utility matrix and start modeling. For this particular dataset, we will use KNearestNeighbors, which identifies clusters of similar users based on book ratings.

We will find the average of the content-based and collaborative filtering scores for each item to generate a final recommendation score. This score determines the rank of items in the recommendation list. This guarantees that the resulting recommendations achieve a balance between personalized content relevance and collaborative discovery. The Hybrid model provides more detailed and individualized recommendations. For example, suppose a user has previously rated highly a novel with a specific author, published in a particular year. The

content-based part of the model might identify other novels with similar authors and publication years. Meanwhile, the collaborative filtering part would recommend books highly rated by users who enjoyed similar novels.

For the **evaluation performance**, we will use the root mean-square error (RMSE) to measure the performance that compares our model predictions with the known ratings from the 'Ratings' file, precision at top 10 and rank correlation that spearman's correlation between system's and user's complete rankings.

Content-Based filtering - Melissa Pinto and Hoi Hin Ng
Collaborative-Based filtering - Yvonne Itangishaka and Mariam Lo
Documentation - All

Flow Diagrams