Book Recommendation System using Content Based and Collaborative Filtering Technique

Himani Parikh[1], Jay Patel[2], Brian McCann[3] ,Yuva Jampana[4]
Department of Computer Science
New York Institute Of Technology

***Abstract* - Today, the amount of information on the internet is quickly increasing, and individuals want methods to identify and access useful information. One such technique is referred to as recommendation system. Recommendation systems make it easier to explore and find the information you need. In general, they are used to raise profits in online stores. Various recommendation systems, such as Collaborative Filtering, Content-based, and Demographic, have been used, however there are various disadvantages that cause these strategies to fail in offering useful suggestions. As a result, more differentiating traits must be identified in order to optimize these procedures. This problem can be overcome by combing the features of various recommendation techniques such as content based, collaborative and demographic. Thus, in this paper a hybrid recommendation system is suggested which satisfies a user by offering optimal and efficient book suggestions.**

***Index Term* – Book recommendation System, User's Interest, Hybrid Technique, Demographic Technique, Collaborative Technique, Content Based Technique, Recommendation Engine**

1. Introduction

A recommendation system is a type of information filtering system that predicts a user's preference or rating for a particular item. These systems typically generate recommendations in one of two ways: collaborative filtering or content-based filtering (also known as a personality-based approach). Collaborative filtering approaches build a model from a user's past behavior (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in. Content-based filtering approaches use specific characteristics of an item to recommend similar items. These approaches are often combined to create a hybrid recommender system.[1]. There are several approaches for making recommendations. These are categorized based on various knowledge sources. Knowledge sources include user characteristics such as age and gender, item/service characteristics such as keywords, genre, and user-item preferences data such as rating, purchase history, and so on. This user-item preference data is used to develop a user profile, which is essential in recommendation. There are two major issues in existing recommendation system one is cold start and another one is accuracy [3].

In the given model, we attempted to address the cold start problem by taking into account certain user input during the account creation process and improving accuracy through the use of various algorithms. However, if the recommendations are too far from the user's preferences, they may lose trust in the system and stop using it. To build trust, personalized recommendations are necessary. One effective way to do this is through the use of demographic suggestions. Additionally, using a collaborative filtering approach can improve the quality of the recommendations. Recommendations suited to the user's age, region and Interests can be made more personalized. The cold start problem is a major issue in many recommendation systems. In such a scenario, the system is unable to give appropriate predictions until it has a better idea about the user's preferences. Demographic recommendations could help alleviate this problem to some extent, if not entirely in case of a newly added user. A user always would like to stay abreast of their liked category or liked author's books. The traditional filtering techniques may not always be able to keep a user updated about the recent trends in books.[2]

To summarize the technique, we use a hybrid model that provides personalized recommendations to each user. This system combines a content-based and a collaborative recommender system method. This allows us to display more accurate and diverse recommendations, which will improve the user experience and increase the chances of purchasing books.

2. Algorithm

2.1. Collaborative Filtering

The collaborative filtering approach generates recommendations based on user-item preference data in the form of ratings. It predicts ratings by analyzing previous ratings given by individuals and other users recorded in a database. The Slope One algorithm is used to compute the average difference between items and their ratings, as well as the number of ratings assigned to each item. If a user has rated many items, the predictions are combined using a weighted average.

2.1.2 Slope one Algorithm

Input: User, Book and Rating information from database
Output: Recommended books
Step 1: Read user book id and user rating.
Step 2: Compute frequency of rating given by users per book.
Step 3: Compute average difference in ratings given by users with respect to current user.
Step 4: Compute predicted rating as sum of the average difference in rating and rating given by current user.
Step 5: Compute weighted average rating for books with respect to frequency and predicted rating.
Step 6: Arrange the weighted average ratings in descending order.
Step 7: Display recommendations.[6]

2.2 Content based Filtering

A content-based filtering system selects and recommends items based on the relationship between their content. In the case described in the paper, the content of a book and a user's purchase history are used to recommend other books with similar content. The system uses various characteristics of the book to make these recommendations and provides an overview of the book's content to help the user find what they are looking for. By using this approach, users can easily discover books with content that is similar to what they have enjoyed in the past. Content based recommendation system filter the entire set of books from the dataset based on the content of the book, where buyer is interested to buy. Recommendation system uses content-based filtering for doing the separation and filtering of books from other books which is having similar kind of content. Also, this helps to discover the content of purchased history from the browsing data. This leads to result in a good recommendation of books to the user based on their interest.[5]

2.2.2 LSH/MinHash Algorithm

Input: User profile (age, gender, book category) from ontology database
Output: Recommended books
Step 1: Read user profile of the current user.
Step 2: Read user profile of the other user for comparison.
Step 3: Initialize minHasharray length and i=0.
Step 4: Initialize minHash matrix to max integer value.
Step 5: Initialize minHasharray with randomly generated hash values.
Step 6: Build similarity matrix of users used for comparison.
Step 7: Calculating similarity between users
while similarity matrix has elements && i < minHasharray length
check
 if current user profile has element
 check
 if hashvalue< minHashmatrix[i][j]
 do
 minHashmatrix[i][j]= hashvalue
 endif
 endif
 if other user profile has element
 check
 if hashvalue< minHashmatrix[i][j]
 do
 minHashmatrix[i][j]= hashvalue
 endif
 endif
endwhile

while i< minHasharray length
check
If minHashmatrix[0][i]== minHashmatrix[1][j]
Identical minHash++
endwhile
Similarity= identical minHash / minHasharray length [6].

3. Problem in Existing Recommendation System

The following real-world challenges were discovered that need to be addressed in recommendation systems: [2]

A. Cold Start Problem: The cold start problem is a common issue in recommender systems, where there is a lack of data for a new user or a new item. In a content-based system, a new item cannot be recommended until it has been rated by users. This can be a challenge, as the system has no information about the item to base its recommendations on. To overcome this problem, other methods, such as demographic suggestions, can be used to provide personalized recommendations for new users and items. For instance, MovieLens (movielens.org) cannot recommend new movies until these have got some initial ratings. The new-user problem is bit hard to handle because it is not possible to find similar users or to create a CB profile without previous preferences of a user.

B. Scalability of The Approach: One of the major challenges of recommender systems today is the scalability of algorithms when dealing with large, real-world datasets. As user-item interactions, such as preferences, ratings, and reviews, generate increasingly large and dynamic datasets, it becomes difficult to effectively process and make use of this data. This issue is a crucial one, as the ability to handle and make use of these datasets is essential for the continued development and improvement of recommender systems.

C. Sparse, Missing, Erroneous and Malicious Data: A common issue in recommender systems is the sparsity of the ratings matrix, which occurs when the majority of users do not rate most of the items. This sparsity reduces the chances of finding groups of users with similar ratings, which can impact the accuracy of the recommendations. To address this problem, various techniques, such as collaborative filtering or demographic suggestions, can be used to provide personalized recommendations even when the data is sparse. This is the most eminent drawback of the CF technique. This concern can be alleviated by using some additional domain information.

D. Big data: Generally, a user can opt for an item of his interest from a recommendation list if the list reflects some diversity in the recommended items to some extent. Seamless recommendations for a restricted type of product have no value until or unless it is desired or explicitly described by the user with a narrow clique of preferences. In the initial stage, when the RS is used as a knowledge discovery tool, the users may wish to explore new and different options available.

3.1. Hybrid recommendation System [2]

Our proposed system is Hybrid Recommendation system designed to overcome cold start issue and reduces dependency of rating-based system. It starts with general page where different books are shown to user based on their categories. User can search any books by its title or author name. While signing up, user is been asked to fill certain information like their category preferences, liked authors, location and age for finding similar users. Based on this information, books are being recommended which in turn help to overcome cold start problem. After signup, user can see their liked category and liked author's books in different titled catalogues.

If existing user has bought book(s) but not rated them yet, they will be shown books based on their purchase history as well as the information they've provided while signing up (categories and authors). Along with it, user will see random recommendations and predictions using different algorithms like SVD, KNN, and Hybrid recommendations based on the books they've rated recently.

Furthermore, the system will track purchase history of users and that will reflect latest recommendations for book recommender system. So, with time user will be shown recommendation based in their most recent purchase history. User can see the details of book such as author, publication, rating, cover page, publication year. These books can be added to cart. The cart will show the books added to it and also other book

recommendations in the "you may also like section". This section is being populated using Cosine similarity algorithm. We are using same algorithm (Cosine similarity) for providing with the search results.

Once a book is ordered, User can rate book based on their experiences in "My Orders" section of the application. Thus, the proposed system will ensure personalized recommendations eliminating drawbacks of rating-based approach.

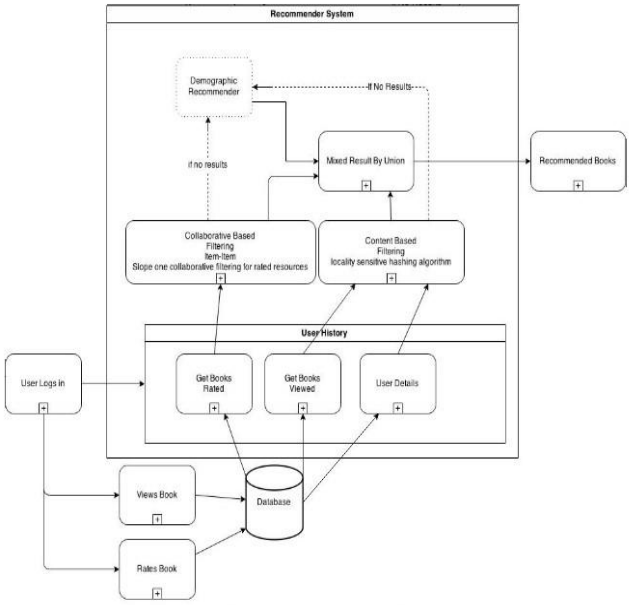


Figure 1: Hybrid Recommendation System

4. Process of Solving Cold Star Problem [5,6,2]

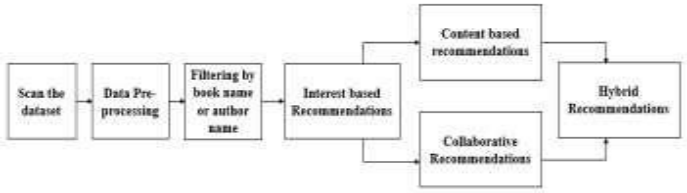


Figure 2: Steps involved for recommendations

Step 1: Scan the Books Dataset In this step, application is scanning the entire storage server and simultaneously performing the data cleaning, which include removal of irrelevant data and keeping the relevant data for recommendations. This process reduces the data sparsity by eliminating missing, erroneous and malicious data from the working data set.

Step 2: Data Pre-processing This step also works of the data correction part to ensure more accurate recommendations. According to our application, it includes the extraction of data that are needed for recommendations, which means extraction of only books having categories and users having demographic data.

Step 3: Filtering by book name or author name This step revolves around proving users with the best and relevant search results. Factors like authors and book name can be searched and the result will return books using cosine similarity algorithm.

Step 4: Perform Content based Filtering In this step we need to perform content-based filtering of books according to user preferences. For example, User1 clicked on book B1, assume that we have some related books B2, B3 and B4 in the dataset. Assume B2 is of different type, but B3 and B4 is of same type of book B1. Now we check the Meta data (category, author etc.) of the books B3 and B4, if it matches with book B1, then the system will recommend books B3 and B4 for the user. If user clicks on book B1, then the user will get books B3 and B4 as the recommended. Cosine similarity is used for finding similarity between two items while searching and showing similar items in cart. KNN is also used for finding similar users whose rating matched with User1's book rating history.

Step 5: Perform Collaborative Filtering Here we consider the quality of the book content. In our example, recommending the books B3 and B4. This will perform based on the registered user's interest and rating. SVD is used for finding user's having similar book interest based on their ratings. RBM is used for giving accurate result even for books which does not been yet rated.

Step 6: Perform Interest Based Filtering Here we consider interest of users, if user likes one specific genre or one author or any subject for example user likes some subjective books like on data science, so we recommend books according to their interest or author likes. It helps to overcome cold start problem.

Step 7: Final Recommendations In the final recommendation, based on type of user,

recommendations will differ like if user is new some interest-based result will be shown to user, if user don't like to rate then interest and similar books of past ordered books will be shown to user else rating-based hybrid recommendations will be shown to user.

5. Architecture [8]

The figure below illustrates the overall architecture of the book recommendation system. The focus of this paper is on the business logic layer on the server side, where the personalized recommendation system is implemented. This system takes into account the user's interests and feedback on book evaluations to provide personalized recommendations for items in the recommendation pool. By designing and implementing this system, we aim to improve the user experience and help users discover books that are tailored to their individual preferences.

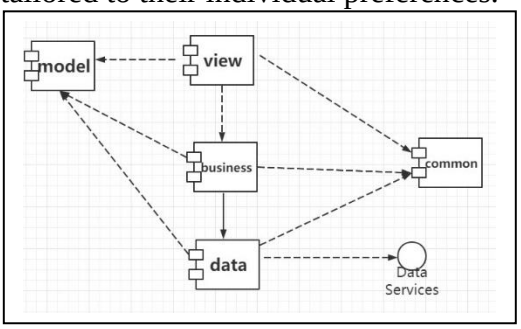


Figure 3: System Architecture

Figure identifies the hierarchical relationships among the modules in the system. The arrow represents the interaction among the various modules of the system, and users use the view to interact with the system.

In the following, we will discuss the details of each module and the interaction between them: **View layer**: This layer is mainly responsible for the page display in the system, which is used to visualize the system data from the user's perspective. It also controls login and user's authority. It is the specific implementation of the user view and controller in MVC model. In this part, we also implement a lightweight perturb function of a user's history data and privacy data according to the user's need to provide a safe and available data environment for the next book recommendation process. In addition, for friendly user interaction, the layer catches the image of a recommendation pool. In this way, users can get the book recommendation system in a very short time. At the same time, computer resources are used reasonably. **Business logic layer**: This layer is mainly responsible for the implementation and expression of specific business logic. It carries out the internal operation structure of the organization through the parameters transferred by the user's call in the presentation layer, realizes the program expression of the business logic that the user wants, calls the data layer to request the corresponding data, and returns the results to the presentation layer for visualization processing. Data mining, recommendation pool generation, and personalized implementation and expression of specific business logic. It carries out the internal operation structure of the organization through the parameters transferred by the user's call in the presentation layer, realizes the program expression of the business logic that the user wants, calls the data layer to request the corresponding data, and returns the results to the presentation layer for visualization processing. Data mining, recommendation pool generation, and personalized parameters are implemented in this layer. The recommendation module is executed periodically in the system to update the user recommendation pool and user parameters and user preference parameters. The data layer is mainly responsible for database connection. Through the business requirements of the business logic layer, it requests to connect to the database, carries out the corresponding data operation, and returns it to the calling business logic part for processing. This greatly improves the maintainability and scalability of the system. **Common layer**: This layer mainly places some common methods and properties. It mainly involves file reading and writing and realizes some specific requirements for system presentation layer and business logic layer. **Model layer:** This layer mainly places the data types used in the system, including the data request parameter form and data return parameter type among the presentation layer, business logic layer and data layer, as well as the storage of query related data. In fact, the scheduling layer is included in the system, which is mainly responsible for the operations that the

system needs to perform regularly, or activities carried out according to certain rules. The layer is used to optimize the processing of system data and regularly import data. This layer mainly involves the regular generation of recommendation pool and personalized parameters. Considering the current scale of the system, we use the business call of the presentation layer for implementation, so we do not propose it explicitly here. This kind of system design and construction greatly improves the performance of the system.

5.1 Data

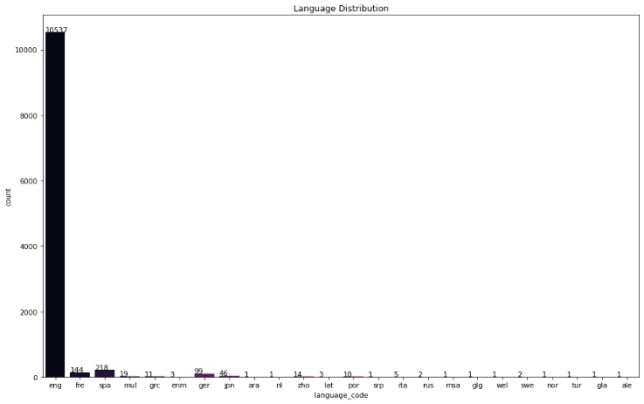
In this project we made Exploratory Data Analysis, Data Visualization and lastly Modelling. The dataset contains 11123 rows in csv file. Each example row represents a book with 12 different information. Before modelling part I have to check NaN values and make some small adjustment for easy to use of the dataset and merge couple of languages on 1 language (en-AUS,en-UK to eng). Later we made a couple of visualization to understand the dataset better. In the modelling part, we used unsupervised learning algorithm K-means which is grouping unlabeled data. For deciding number of cluster we used Elbow method and decided to do 5 clusters. Finally, we test the model with several books and add input function for searching easily.

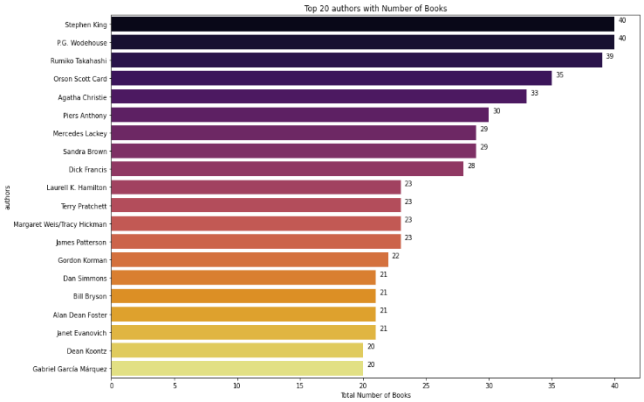
Dataset contains 12 columns and 11123 rows.
Columns Description:
bookID = contains the unique ID for each book/series
title = contains the titles of the books
authors = contains the author of the particular book
average_rating = the average rating of the books, as decided by the users
ISBN ISBN(10) = number, tells the information about a book - such as edition and publisher
ISBN 13 = the new format for ISBN, implemented in 2007. 13 digits
language_code = tells the language for the books
Num_pages = contains the number of pages for the book
Ratings_count = contains the number of ratings given for the book
text_reviews_count = has the count of reviews left by users
publication_date = date of publication
publisher = name of publisher

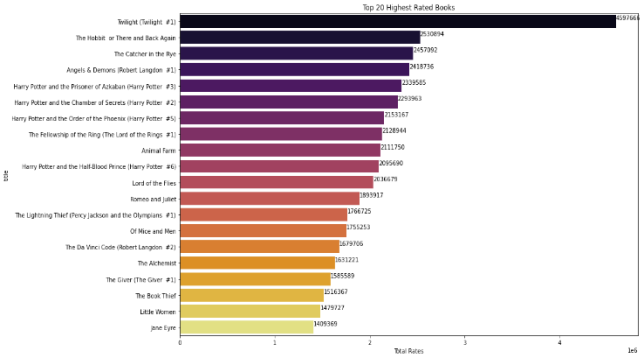
the Data Services

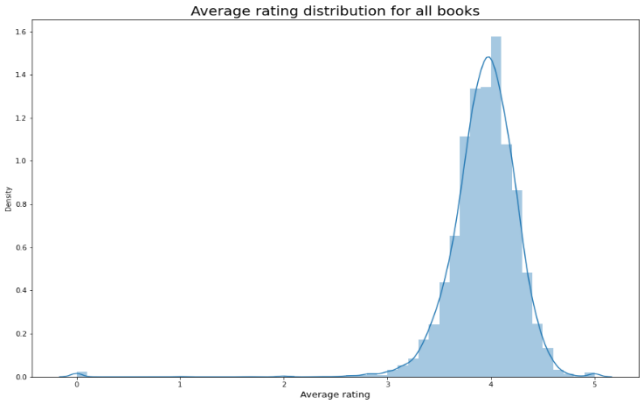
5.2 Exploratory Data Analysis [7]

In EDA we visualize language distribution, Top 20 authors with number of books, Top 20 highest rated books, and Average rating distribution for all books.

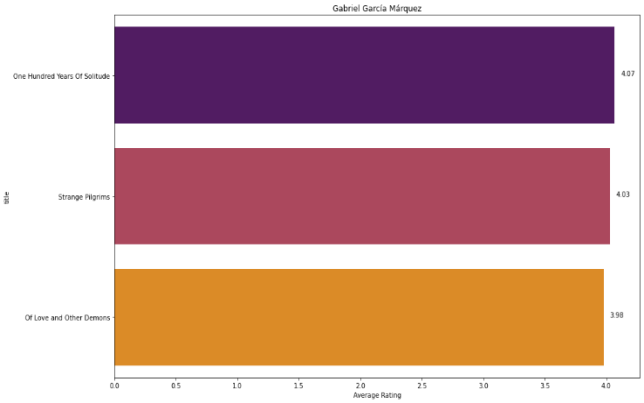


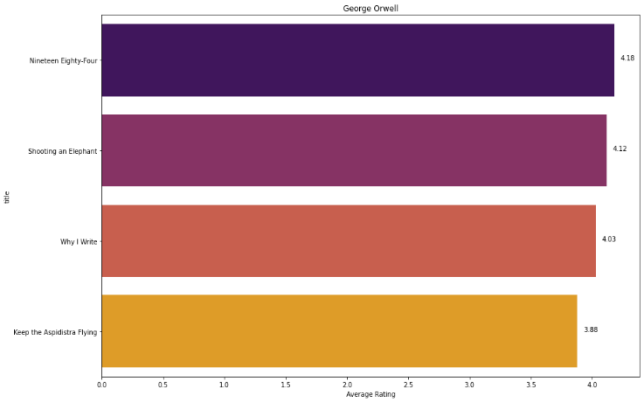


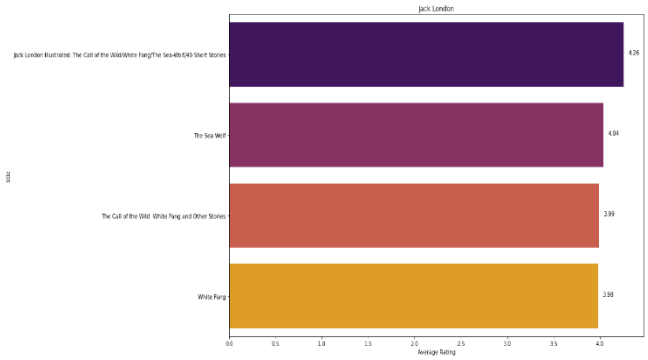


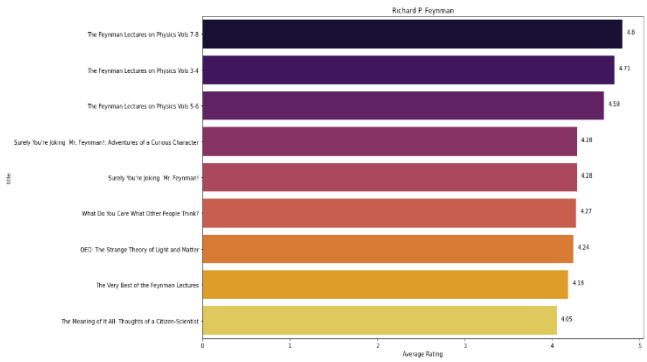


Secondly, we create a list for my favorite authors and visualize their books according to average rating of books.
authors = ['Gabriel García Márquez', 'Jack London', 'George Orwell', 'Jules Verne', 'Richard P. Feynman']

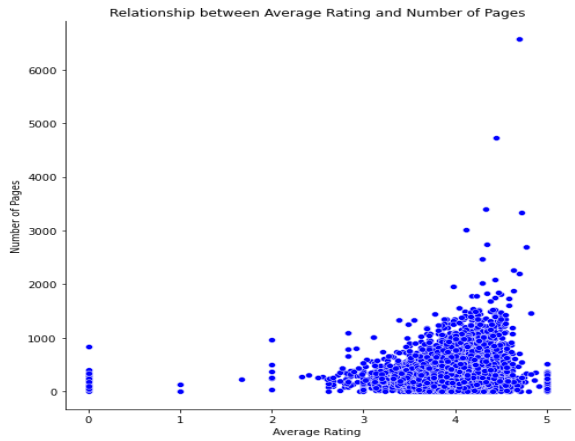


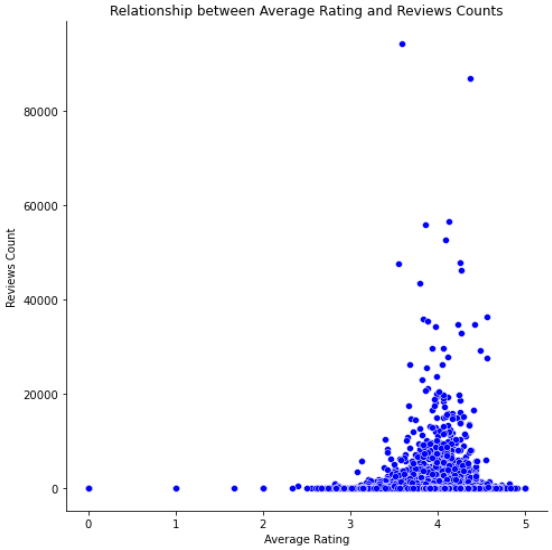


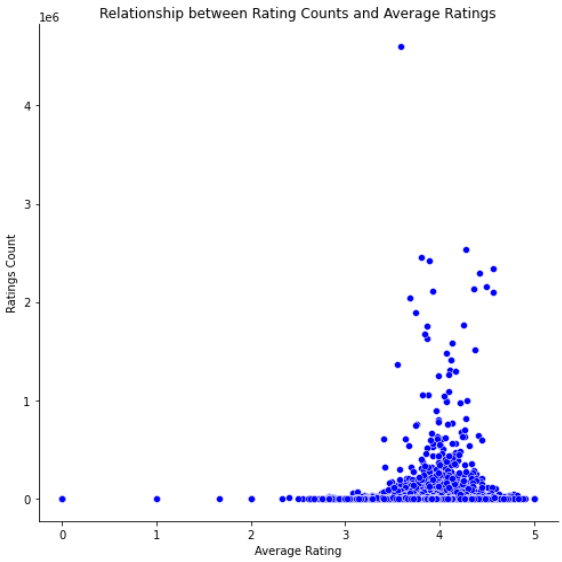




After all these steps, we wanted to investigate the relationship of columns. As you can see below, Average Rating and Number of Pages, Average Rating and Reviews Counts, Rating Counts and Average Ratings.

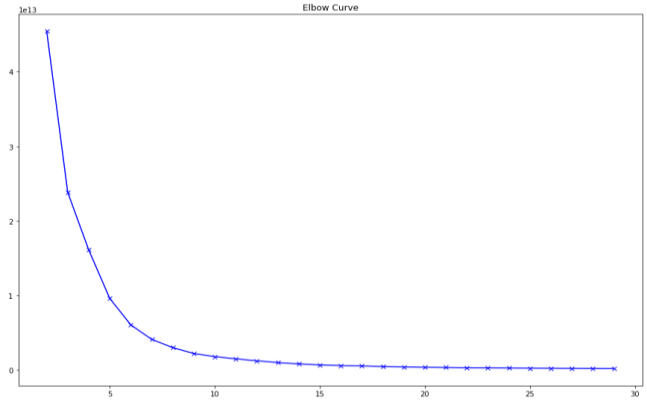




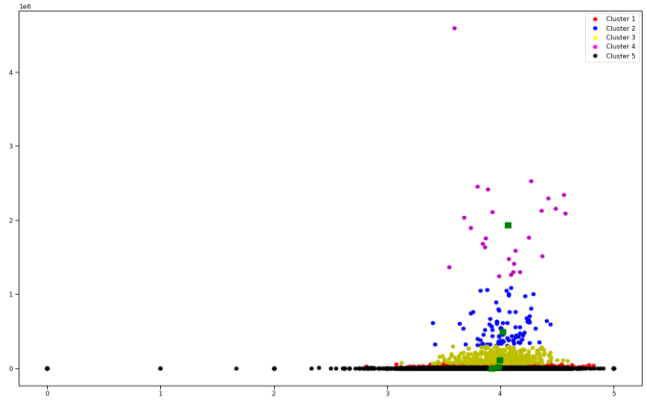


5.3 Modeling

In the modelling part, we have already decided to use K-Means Algorithm but we have to decide how many should we use. For deciding this we used Elbow Method which is giving very good assumption. In the figure below you can see the graph.



After deciding 5 clusters, we created plotting and expressing clusters.



Lastly, we implemented Min-max scaler, for reducing bias. Because some books have massive number of features and some of them very few. So, Min-Max scaler will find the median all books.

5.3 Result and Future Work

print_similar_books("Caesar (Masters of Rome #5)")

- The Metaphysical Club
- One Hundred Years of Solitude
- Alice's Adventures in Wonderland and Through the Looking-Glass (Alice's Adventures in Wonderland #1-2)
- In Cold Blood
- Desperation / The Regulators: Box Set

print_similar_books("Lord of the Flies")

- Introduction to the Philosophy of History with Selections from The Philosophy of Right

- Marie Dancing
- The Odyssey
- The Hour Before Dark
- A Philosophical Enquiry into the Origin of our Ideas of the Sublime and Beautiful

As a result, book recommender gives good results. But still there is more room for improvement. Such as, finding category of each book makes everything more effective. Or increasing size of data or information (more rows) can help more accurate recommendations

6. Conclusion

A book recommendation system could be evaluated based on several different factors, such as its accuracy, ability to handle different types of data, ability make personalized recommendations, scalability, and so on. In general, however, a good book recommendation system should be able to provide accurate and personalized recommendations to used based on their reading history and preferences and should be able to handle large amounts of data without sacrificing performance**.** Our research indicates that integrating Collaborative Filtering and Content-based techniques with demographic attributes to produce a Hybrid approach yields the best suggestions. The cold start problem for new users and new products has also been solved by leveraging demographic data to locate user co-relationships and receive suggestions.

7. References

[1] Robin Burke, ”Hybrid Web Recommender Systems”, January 2007.
[2] Anagha vaidya, Dr. Subhash Shinde, “Hybrid Book Recommendation System”, July 2009.
[3] Salil Kanetkar, Akshay Nayak, Sridhar Swamy, Gresha Bhatia, “Web-based Personalized Hybrid Book Recommendation System” IEEE International Conference on Advances in Engineering & Technology Research (ICAETR-2014), August 01-02, 2014, Dr. Virendra Swarup Group of Institutions, Unnao, India.
[4] Jihane KARIM, ”Hybrid System for Personalized Recommendations” 978-1-4799-2393-9/14 ©2014 IEEE
[5] Praveena Mathew, Ms. Bincy Kuriakose, Mr.Vinayak Hegde, Book Recommendation System through Content Based and Collaborative Filtering Method, **2012**.
[6] Manisha Chandaka, Sheetal Giraseb, Debajyoti Mukhopadhyayc, Introducing Hybrid Technique for Optimization of Book Recommender System, 2015.
[7]https://github.com/HalukSumen/Book_Recommender
[8] Yong Zhu, Combining user interest and feedback in an online book personalized recommendation system, 2020

Role of Group Members:-

1. **Brian McCann:** Found the source reference files for our project.
2. **Yuva:** Made thesis paper and PPT for presentation.
3. **Himani:** found the source and research regarding book recommendation system.
4. **Jay:** Code the program and did data visualization and provided related content for thesis paper.