Book Recommendation System

Gayathri Sundareshwar, Keerthana Gopikrishnan, & Deepasha Jenamani,
Department of Applied Data Science, San Jose State University
DATA 240 – Data Mining

May 17th, 2023

1. Introduction:

As technology occasionally evolves, the times have changed from relying on hardcovered books to e-books. Recommendation systems can assist such users with personalized options. The proposed study aims to create a system that helps suggest books related to the title search based on the user's history. The motivation behind choosing the study is that many prevalent works recommended books either based on title or through user history but not both inclusively. The background research suggests that most works use algorithms such as the traditional association rule. This does not include an extensive data-cleaning process which will be addressed as one of the milestones of the study. The project demonstrates how the proposed solution is better than the traditional approaches. The goal is also to develop a system that is unbiased and caters specifically to the users' needs. The study also aims to develop a User Interface which can help improve the interaction between the user and the system. The study aims to improve the performance by inducing hybrid methodologies, which also experiment with traditional ones.

2. Literature Review

An intensive literature review can help establish context, identify gaps in existing studies, understand the significance, and develop a theoretical framework for the current project. Thus, it was the first stage performed. Desai et al. (2016) suggested a book recommendation system using the k-means clustering method and item category weights[1]. The proposed model calculates the weight of each category, measuring the number of users who viewed or purchased items and the average rating of the items in that category. K-means clustering groups the users based on their reading preferences using the item category weights and the users' reading history, and the books are recommended using cosine similarity. This model helps handle large datasets in real time by using a computationally efficient clustering method. The drawback is that the system is prone to bias if some categories are underrepresented.

Adak et al. (2021) proposed a method where the data is first clustered utilizing an unsupervised learning method, followed by the design of a decision tree model with the C4.5 algorithm, which determines the rules in the fuzzy model [2]. The recommendation based on the user's preference happens using this fuzzy model. The model has enhanced accuracy by using a decision tree to determine the fuzzy rules. The use of rating parameters leads to the generation of less effective rules. Jomsri et al. (2014) recommended using an Apriori algorithm for association rule mining to identify the frequent item sets [3]. The rules are generated from the preprocessed data, and different metrics, such as support, confidence, and lift, are used for the evaluation, which provides personalized recommendations based on their profiles. The methodology is applied to a real-world dataset in a digital library, proving that the approach is effective in real-world scenarios. The drawback is that the system fails to introduce them to new authors or genres to the user. Xin et al. (2013) proposed using accurate Collaborative Filtering (CF) algorithms based on user borrowing records with time stamps. Traditional CF algorithms are linearly blended using different methods without considering the feedback ratings [4]. Different measures such as Pearson correlation coefficient, Jaccard, and cosine similarity are used for calculating the similarity, and books are recommended. High accuracy is achieved by blending different CF algorithms, which is an additional benefit. However, as other influencing factors like personal preference and detailed feedback ratings are not considered, this may not generate diverse recommendations.

The review of the existing studies shows that clustering and rule-based methods are mainly used in most cases. Clustering follows an unsupervised learning approach, whereas a rule-based system uses a semi-supervised or supervised approach. The methodologies suggested in this project are content-based, collaborative filtering(CF), and weighted average. The use of CF ensures providing a personalized recommendation to users. Additionally, the system uses a content-based filtering approach that improves accuracy despite less user rating information.

3. Methodology

The methodology for a data mining project involves a systematic approach to each of these stages to identify patterns and trends that can be used to make informed business decisions. By following a structured methodology, projects can be more efficient and effective in achieving their objectives. **Figure 1** depicts the architecture of the book recommendation system.

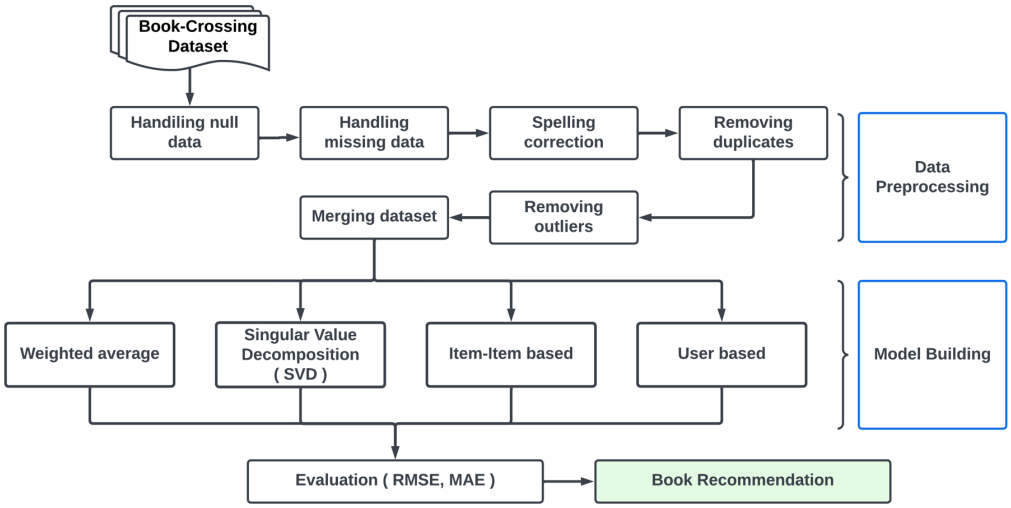

Figure 1: Architecture of Book Recommendation System

3.1 Data Collection

The data collected for this project was Book-Crossing Dataset mined by Cai-Nicolas Ziegler and DBIS Freiburg. The dataset was crawled for four weeks from August to September 2004 from the Book-Crossing Community with permission from Ron Hornbaker. The dataset consists of three tables, BX-Users consists of User-Id, Age, and demographic information of 278,858 users; the second table is BX-Books which details the Book Title, Author, Year of publication, and publisher of 271,379 books. The BX-Book-Ratings table consists of book information and the book's rating from 1-10, which has 1,149,780 records. The data was cleaned and preprocessed before being used for recommendations to address the issue of inconsistency due to varied sources and other errors.

3.2 Data Preprocessing

To make raw data ready for analysis, pre-processing entails cleaning, converting, and other preparations. The data's accuracy, consistency, and error-freeness are all ensured through pre-processing, which also helps ensure that the analysis is of the highest caliber possible. By pre-processing the data, analysts can guarantee that the data unbiased and appropriate. Some of the pre-processing steps that were carried out on the data are:

1. **Handling Nulls:** The null values in the dataset were addressed using techniques such as imputation (replacing with mean, median or mode or deletion (removing entirely).
2. **Handling missing data:** The missing data can be imputed using methods such as mean or regression. If more than 80% of records were missing, removal is done.
3. **Correcting spellings:** The records were checked for spelling and corrected to ensure no errors. The records were cleaned and replaced without abbreviations.
4. **Removing duplicates:** The dataset may contain duplicate entries, which can skew the analysis or model. Removing these duplicates can improve the accuracy of the analysis.
5. **Removing outliers**: The dataset may contain data points that are significantly different from the other. Such records were removed or imputed with mean or median.
6. **Merging all datasets:** The final dataset is obtained by merging all the cleaned datasets into one that can be used for the recommendation system.

3.3 Modeling and model details and training:

A well-designed model for the proposed recommendation system assists in improving user satisfaction by providing a relevant and personalized recommendation. A diverse set of modeling techniques depending on the available data are proposed as part of this project which is explained in this section.

Weighted Average :The first model is based on a weighted average rating method where the overall rating of a book is calculated by considering both the average rating and the number of users who have rated it. All those books are considered where the user rating exceeds the minimum threshold value of 64.

Collaborative Filtering - Singular Value Decomposition : Collaborative filtering (CF) is a technique used in recommendation systems to make predictions about the interests or preferences of a user by analyzing the patterns of ratings or interactions between users and items. Based on this, a second methodology, model-based collaborative filtering, is used, which predicts the user preference where a model is created based on a user-item rating matrix, and a rating is predicted for an item that the user has not interacted with. Two models Non-negative Matrix Factorization (NMF) and Singular Value Decomposition (SVD) are chosen to perform the recommendation. A user rating of a threshold value of three is considered while selecting the top items to be considered for the modeling. Root Mean Square Error (RMSE) calculation shows that SVD is most suited for the task rather than NMF. The model is trained using a 3-fold grid search cross-validation strategy, GridSearchCV, from the Surprise library with four hyperparameter settings: number of factors(n_factors), number of iteration(n_epochs), parameter learning rate(lr_all) and parameter regularization term(reg_all). Prediction ratings and the actual ratings are compared using the absolute error approach. Finally, the validation of the model proves a higher accuracy of the proposed methodology.

Collaborative Filtering - Item-Item-Based : The next model is an item-item-based CF recommendation using the K-nearest neighbors (KNN) algorithm. Initially, a user-item pivot matrix is created where the rows represent the user IDs; the columns represent the ISBN of the books. Only the popular books are filtered out and used where a minimum threshold value of 50 ratings is considered. A final matrix is created representing the book titles row-wise, user IDS in columns, and the value in each cell shows the corresponding user rating. After handling the null values, a sparse matrix is generated from the mentioned final matrix, which is used as input to build the item-item similarity matrix using KNN, where distances between points are measured using cosine similarity and six neighbors are taken into account. A pivot matrix generated after training the KNN shows the user IDs row-wise and books in columns, transposed further. SVD is applied to reduce the dimensionality to 12 components. For the recommendation, a correlation matrix is obtained using the reduced matrix for retrieving similar books having a higher positive correlation based on a book title.

Collaborative Filtering - User-Based : Finally, a personalized recommendation model is developed using CF methodology. The number of interactions between a user and books is calculated, and only those set of users who have rated at least 100 books are filtered and used. The book ratings are transformed using a logarithmic function for each user and book pair. For training the model, 80% of the data is used, where the splitting of the data follows the stratification approach ensuring the balance of users in both the training and testing set. A pivot matrix generated using the training dataset shows the corresponding book ratings by each user. For obtaining the prediction result, the SVD algorithm is used with an appropriate number of factors which is 15, in this case, ensuring capturing of finer distinctions in user preferences and item characteristics, avoiding the risk of overfitting. A CF-based recommendation system is then developed using the mentioned user-item rating matrix. For any specific user, the predicted ratings are sorted, excluding the items that the user has already interacted with, and the top recommendations are generated.

4. UI Development:

A well-designed user interface(UI) can enhance the user experience, making it more intuitive, efficient, and accessible. As part of this project, a UI is developed to provide a platform for the recommendations. There are two ways in which recommendations are provided using the designed UI. The UI is designed to represent both users with archived history recorded as well as first-come users. This has been incorporated as an attempt to tackle the cold start problem. The UI consists of two options that can be used to log in; the first one is by using a valid user ID. In this case, an existing member can log in using a user ID along with the valid associated password, and personalized books are suggested to the user based on the preference, including book ratings and other details such as publication and author. The second way to get suggestions is by a guest search, which provides recommendations based on a book title. Additionally, the developed UI shows the current statistics, and for new users, a registration portal is provided to get personalized recommendations. Once the book details or the user credentials are entered, the UI traverses to a different page where the recommendations will be displayed. **Figure 2 and 3** depicts member-based search and title based search with results.

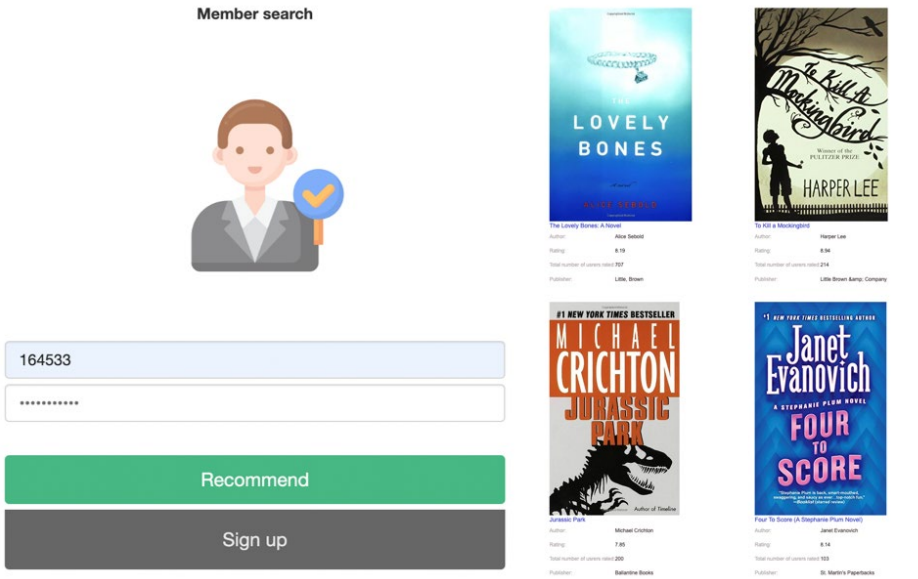

Figure 2 : Member search and sample of resultant recommendation

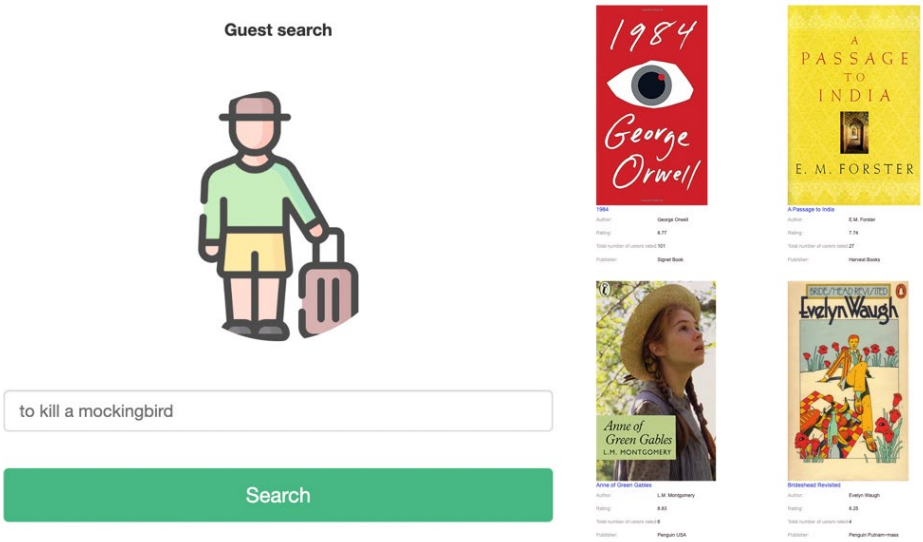

Figure 3 : Title search and sample of resultant recommendation

5. Evaluation:

In order to understand how optimal the model is, it is necessary to validate it through a diverse range of experimental setups followed by the evaluation phase. It is vital to choose the correct set of experimental setups and tune the hyperparameters as such so that the precision of the model is improved significantly. For the filtering method discussed before, namely the item-item-based and content based, the optimal set of hyperparameters was identified through a thorough grid search. The finalized model is trained and evaluated by assigning the hyperparameters to the best-case setup identified through the grid search. **Figure 4** represents the tune hyperparameters for the respective models.

Hyperparameters -KNN	
n_neighbors	5
algorithm	brute
metric	cosine

a) Hyperparameters for Item-Item Based

Hyperparameters - SVD	
n_factor	50
n_epochs	20
lr_all	0.005
reg_all	0.1

b) Hyperparameters for Content-Based (SVD)

Hyperparameters -NNF	
n_components	2
init	random
l1_ratio	1
random_state	42

c) Hyperparameters for Content-Based (NNF)

Figure 4 : Hyperparameters for Experimental Setup

Following the tuning, the evaluation phase begins. Evaluation is an essential part of model development which helps determine the performance and generalization ability of the ML model on new, unseen data. It is essential to choose the appropriate evaluation metric based on the problem at hand and to ensure that the evaluation is conducted on a representative sample of the data. By conducting a thorough evaluation of the model, one can make informed decisions about the performance and suitability of the model for the intended application. RMSE and MAE are commonly used evaluation metrics to help determine the model's accuracy. RMSE measures the difference between a continuous variable's predicted and actual values. In contrast, MAE measures the absolute difference between the predicted and actual values and provides a measure of the average magnitude of errors.

This project uses SVD and NMF models to predict book ratings for collaborative filtering. RMSE and MAE were used to evaluate the model, and the model with the lower RMSE or MAE value is generally considered to perform better and is considered for deployment. However, it is essential to consider other factors, such as computational complexity and model interpretability, when selecting a model based on the project requirements. Cross-validation is significant for measuring RMSE and MAE as it assists in mitigating the risk of overfitting the model to the training data. A three-fold cross-validation methodology is followed to obtain a more accurate estimate of model performance on new data, which gives potential information to select the best-fit model or tune the hyperparameters. **Figure 5** represents the evaluation results of SVD and NMF while content filtering. SVD is identified to be more optimal since it has lower scores and is hence considered suitable for the recommendation task.

	RMSE	MAE	Fit Time (s)	Test Time (s)
SVD	1.601957	1.223955	3.387059	0.947547
NMF	2.612234	2.230801	7.868145	0.861659

Figure 5 : Content Filtering Results

For the comprehensive evaluation of the recommendation system, there are two commonly used metrics which are recall@k and hitcount@k. Recall@k gauges the proportion of relevant items that are recommended in the top k items, whereas hitcount@k calculates the number of relevant items that are recommended in the top k items. A higher value of these metrics indicates that the algorithm is better at recommending books that are relevant to the user. As part of this project, recall@5, recall@10, hitcount@5, and hitcount@10 are used. **Figure 6** portrays the evaluation result of the model using these four metrics.

Models	hits@5_count	hits@_10count	recall@5
Collaborative Filtering	259	336	0.704
Content-Based Filtering	231	293	0.632
Item-Item Based Filtering	197	249	0.557

Figure 6 : Model evaluation result

6. Discussion and future improvements

In recent years, the book recommendation system has become a critical component of many online platforms, such as Amazon, Goodreads, and Google Books. With the influx of data available, it is becoming more challenging for users to discover relevant and interesting books. Collaborative filtering, on the other hand, recommends books to users based on their similarity to other users. It considers the preferences of other users with similar interests and recommends books they have enjoyed. The two collaborative filtering techniques experimented with were Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF).

The results show that SVD was more accurate in predicting the user's preferences as it had a lower Root Mean Squared Error (RMSE). However, it was also observed that collaborative filtering often suffered from the cold-start problem, where it fails to recommend books for new users or books with few ratings handled in this project. In conclusion, using a hybrid approach that combines content-based filtering, collaborative filtering, and the weighted average method is recommended. This approach can overcome the limitations of each technique and provide personalized and diverse book recommendations.

Furthermore, future research can explore deep learning techniques, such as neural networks, to improve the accuracy and robustness of book recommendation systems. Future work can also include implementing voice recognition for searching instead of typing. Voice recognition can be further improved to adapt to multiple languages.

Reference

[1] T. Desai, S. Gandhi, P. Murlidhar, S. Gupta, M. Vijayalakshmi and G. P. Bhole, "An enterprise-friendly book recommendation system for very sparse data," *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, Pune, India, 2016, pp. 211-215, doi: 10.1109/CAST.2016.7914968.[1]

[2] M. F. Adak and M. Uçar, "A Book Recommendation System Using Decision Tree-based Fuzzy Logic for E-Commerce Sites," *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Ankara, Turkey, 2021, pp. 1-5, doi: 10.1109/HORA52670.2021.9461319.[2]

[3] P. Jomsri, "Book recommendation system for digital library based on user profiles by using association rule," *Fourth edition of the International Conference on the Innovative Computing Technology (INTECH 2014)*, Luton, UK, 2014, pp. 130-134, doi: 10.1109/INTECH.2014.6927766.[3]

[4] L. Xin, E. Haihong, S. Junde, S. Meina and T. Junjie, "Collaborative Book Recommendation Based on Readers' Borrowing Records," *2013 International Conference on Advanced Cloud and Big Data*, Nanjing, China, 2013, pp. 159-163, doi: 10.1109/CBD.2013.14.[4]

[5] Dataset : http://www2.informatik.uni-freiburg.de/~cziegler/BX/