

Predicting Your Ratings to Unveil Your Next Best Read

Group W04G9

Emily Joseph

1410988

ejjose@student.unimelb.edu.au

Erich Wiguna

1389444

ewiguna@student.unimelb.edu.au

Raphael Renaldo

1389446

rrenaldo@student.unimelb.edu.au

Executive Summary

Our main objective was to implement a recommendation system based on how we think users from an online bookstore would rate a batch of new books with only information of their past ratings. We utilized data pre-processing techniques including scaling, data imputation, discretizing and data manipulation to enhance our data integrity and accuracy. In utilizing Information Gain to assess correlations between one attribute to another, it was revealed that the User ID exhibits the highest correlation with book ratings. Due to the fact we only had limited attributes about each book, we were forced to base the similarities between books by using the TF-IDF of their titles. We then developed a similarity matrix using these TF-IDF values and fed them, as well as User ID data, into a K-Means clustering algorithm.

Despite our skewed title cluster, our recommendation system achieved an accuracy of 62.4%, surpassing the baseline accuracy of 30%. With this recommendation system, we aim to provide personalized book recommendations that can enhance user satisfaction and engagement, whilst also benefiting authors and publishers with their discoverability and sales.

Introduction

With the increasing number of books in the market, finding the right one may be tough. Book lovers might spend ages scrolling through endless titles and genres, hoping to stumble upon something they'll love. That's when a recommendation system for new books would come in handy. By recommending new books that align with a customer's interest, we can ensure they never face a boring day not knowing what to read next. In order to know which book a customer would like, we would need to predict how much that customer would enjoy that particular book, ideally estimating the rating they would assign to it. But what information is required to make such predictions? We would need data describing both the book and the customer. This inquiry forms the basis of our research question: "to what extent can we predict book ratings using the attributes of a book and a user?"

The attributes of books in question comes from a dataset named "BX-Books.csv" which contains information on 18,185 distinct books including their International Standard Book Number (ISBN), Title, Author name, Year of publication and Publisher name. Using the ISBN provided for each book, we are able to merge this dataset with contents of "BX-Ratings.csv" which also includes an ISBN for each book, as well as ratings provided by users who have read those books alongside their respective User IDs. Finally the attributes of each user can be found in "BX-User.csv", which comprises information on 48,299 users of an online bookstore, including their User ID, City State, Country and Age.

In joining these three datasets, it allowed us to better see all the available data, enabling us to question any potential correlations between user attributes and the way they rate books. For example, users from the same location might have similar book preferences, thus resulting in similar ratings for the same books they've read. Once we have understood these correlations, we can then proceed to predict ratings for books they haven't read yet using these attributes. The data containing books the users have yet to explore can be found in "BX-NewBooks.csv" which holds information on 8,924 books, including their ISBN, Title, Author name, Year of publication and Publisher name. This dataset follows a format identical to "BX-Books.csv", but for different books. With a sample of users who have previously rated books from the online bookstore, we create a new dataset "BX-NewBooks-Users.csv" which contains information about 8,520 users from the original dataset "BX-User.csv".

Consequently, this would mean our recommendation system would exclusively cater towards users with a prior rating history.

After we predict how each user in “BX-NewBooks-Users.csv” would rate the books assigned to them from “BX-NewBooks.csv”, using our choice of collaborative filtering, it is essential that we compare these predicted ratings to the actual ratings of these books. These real ratings can be found in “BX-NewBooks-Ratings.csv”. This assessment of prediction accuracy enables us to determine patterns or trends that may signify weakness in our item-based collaborative filtering model, prompting us to strategize on how to improve it.

Methodology

Scaling

In every dataset, it's expected that we encounter wrong data inputs or outliers. To ensure data integrity, we addressed potential outlier scenarios by thoroughly checking each column that contained continuous numerical data. This involved checking columns such as:

- Ratings: We checked for values outside the range of 0 to 10, including negative ratings, but none were found.
- Ages: Ages needed to be within reasonable ranges. We discovered age inputs below 6 years old, which we removed since children are known to start reading after this age. Similarly, for ages above 100, we considered them unrealistic thus we removed them as well.
- Years of publishing: Any future publication years, which are years beyond 2024, were found and removed from the dataset.

Data imputation

After ignoring the unrealistic outliers through rescaling, it became evident that the age column contained a noticeable number of missing values (NaN). We then proceeded to analyse the number of missing age values compared to the total 204,164 rows of data. To our surprise, there was a significant portion of missing age values, 30% to be exact shown in Figure 1.

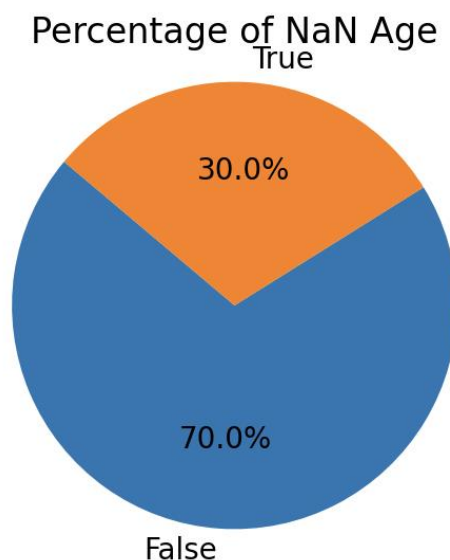


Figure 1: Pie chart showing the percentage of NaN User Age in the dataset

Due to the significant number of missing age values, removing all these rows from our dataset could potentially introduce inaccuracies into our data analysis. Our initial strategy to predict the missing age values involved filling them with the median age from all user ages, which was 34. However, this approach led to an incredibly skewed distribution, heavily favouring the age of 34, as shown in Figure 2, due to all 61,309 rows of NaN age being replaced with this value.

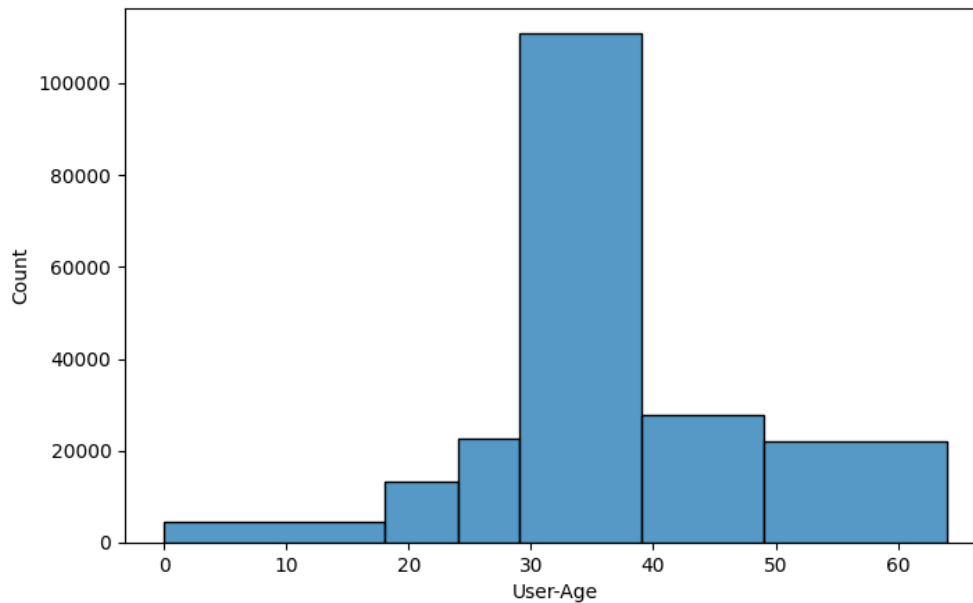


Figure 2: Histogram of the User Age frequency in bins after filling the NaN values with median

We concluded that filling all the NaN values with a single age was not the most effective method. Thus we opted to employ K Nearest Neighbour clustering to better predict the missing user ages for every user using the average age of 5 nearest neighbours of that specific user.

Given that the only information we had about our users, aside from their ratings, is their locations (city, state, and country), it was clear that the only logical data we could utilize to predict their age is their location. However, due to how KNN clustering is unable to process string data, such as the countries' names, it meant it was necessary for us to somehow convert the string name of countries into discrete numerical values.

Data manipulation

Whilst in the process of filling in our missing age using the discrete numerical values of each user's country, we realized that columns of state and city were not necessary to use as we are provided with the country feature which captures the information that is given by the two features before. Although using state and city would give us more detailed data on the location of the user, having the country as the main feature in denoting the user location provided us with a middle ground in detail and simplicity. The other reason for ignoring those two features is because countries universally have unique names, unlike states and cities which could distract the machine learning models that are implemented in this research.

With the use of GitHub, we were able to find a table of country codes, region codes and subregion codes for each country in the 'all.csv' (Duncalfe, 2017). However, after assessing the country codes of two countries close to each other, it became clear that these codes did not accurately reflect the proximity between each country, which is essential for KNN clustering. Consequently, we opted to find a table consisting of longitude and latitude coordinates for each country. This would provide a more accurate basis for geographical correlation to better predict the missing values of age. ▼

Before connecting the country names to those in the GitHub csv 'location.csv' (Wang, 2012), we had to ensure the country names from both datasets had to precisely match each other. Thus, we removed the extra strings (such as " after each country name) in the given dataset and ensured that all letters of the country names in location.csv were lowercase. After merging these datasets, a new feature of distance is added where the longitude and latitude are computed with the centre points of 0, 0. This Distance feature is then used as the similarity measurement between users in the KNN clustering which then we gain the distribution of binned age as Figure 3.

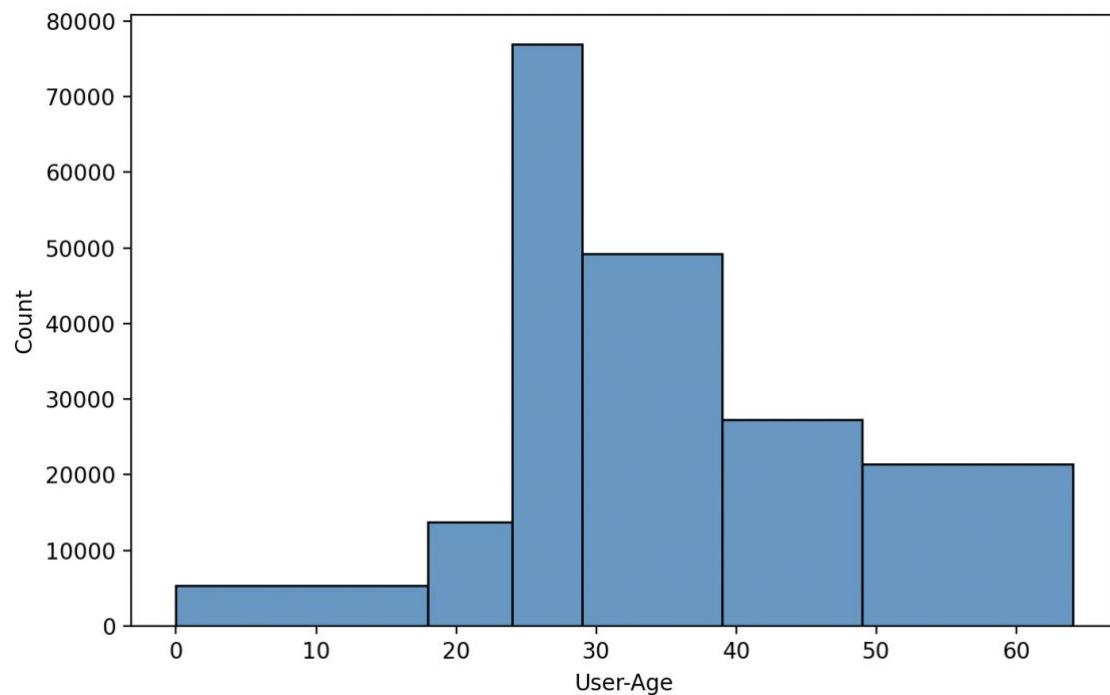


Figure 3: Histogram of User Age frequency in bins after filling the NaN values using KNN methods

However after analysing this produced binned ages in data exploration, we opted to ignore this feature as it provides little information regarding the main goal of predicting new users' rating. This leads us to the pre-processing of the next accessible feature of book title. The main reason why is because the KNN method is biased towards the massive number of users that are located in the United States as seen in Figure 4.

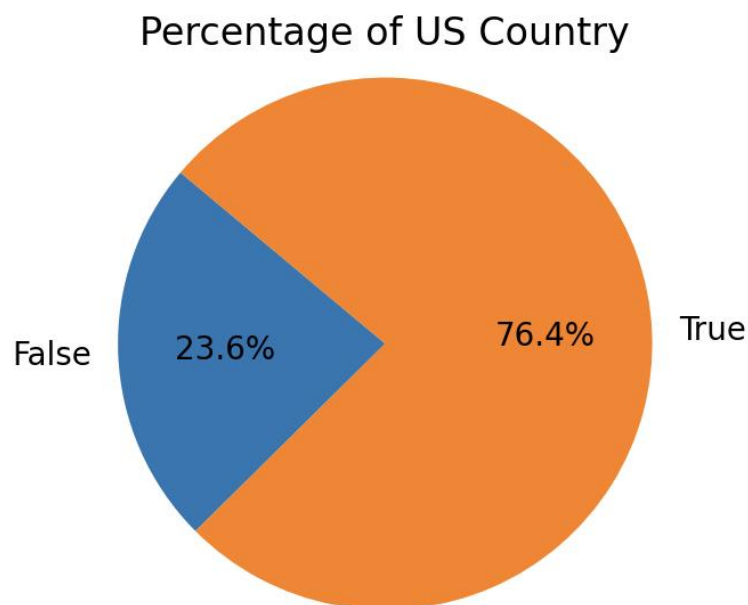


Figure 4: Pie Chart showing the percentage of United States as User Country


TF-IDF

In developing our recommendation system, it's essential to figure out how similar a book is to another book. However, since we lack attributes indicating their similarities between the list of books, we were forced to calculate the TF-IDF vectors for each book title in the dataset. Our aim was to use the textual content of the titles, under the assumption that books with similar names might share similar content.

However, it is important to pre-process the book title so that the TF-IDF matrix could capture relevant information and similar trends. The steps that were used to pre-process the book title are lowering all the letters, removing the punctuation, tokenizing the title, removing the stopwords, using a stemmer to break down words into their base form, and finally combining the tokens.

Clustering

After transforming the book title into a TF-IDF matrix by way of vectorization, this model proceeds to cluster them into a feature called “Title-Cluster” through K-means. K-means works by placing random points called centroids and calculates the distance between said points and data points, in this case, the data provided by the TF-IDF matrix to measure the similarity between book titles. Then, it iterates through every single data point to find the closest centroid, creating another centroid in the centre of all data points that have the same centroid. This cycle repeats until it stabilizes, meaning that no data points change their closest centroid. Thus, creating a title cluster containing similar book titles.

After obtaining the title cluster, this model calculates the mean book rating of each user for every title cluster that they had read. This model intended to predict the new book rating by using the mean rating of the title cluster that the new books fall into for every user. However, this proves to be another obstacle, as the mean amount of books each user reads is around four while the median is one, this means that the majority of users had not read every single title cluster, with the cluster being greater than one. Thus, there is a high likelihood that the new books that the user reads are not within the same title cluster as the old ones, meaning that there is no mean rating that can be used to predict the new book rating. Therefore, changes to the model are implemented, if such a case were to happen, the predicted rating would use that specific user overall mean rating without accounting for its individual title cluster mean rating. 

Data Exploration and Analysis

NMI and IG Checking

Both Normalised Mutual Information (NMI) and Information Gain (IG) are heavily used in figuring out which features should be retained and would be the best to help the machine learning model in predicting the rating of the new books. Unlike Mutual Information (MI) which has values up until infinity, the main advantage of NMI is that the result is normalized so that the values are in the interval [0,1] providing us with a more standardized scale for comparison. Formulas that are used are shown in Figure 5.

$$H(X) = -\sum P(X=xi) \log_2(P(X=xi))$$



$$MI(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

$$IG(X,y) = H(X) - H(X|y) = H(y) - H(y|X)$$

Figure 5: Formulas of entropy (H), mutual information (MI), and information gain (IG)

After computing the NMI of binned rating given the year of publication, book author, ISBN, and book publisher we gained the values of 0.001, 0.016, 0.026, and 0.006 denoting that there is little information that is shared between the binned rating and four of these features. This led to the decision of ignoring those four features as it is highly unlikely for those features to provide relevant information to our model.

Determining Features

The main reason that led our model us to use the User-ID with the help of the clustered Book-Title is because of the high information gain of binned rating given the User-ID which sits at 0.451. As the highest information gains value out of other features, the User-ID feature is used as the main feature that would best portray the characteristics between users.  Therefore, to capture the characteristics of the book, we opted to use the K-means clustering algorithm that clusters the TF-IDF matrix of the Book-Title. As seen in Figure 6, a k value of 6 is chosen for the K-means as it is located at the ‘elbow’ 

of the graph where we could promote a minimum cluster while maintaining a low error (in this case Sum of Squared Error).

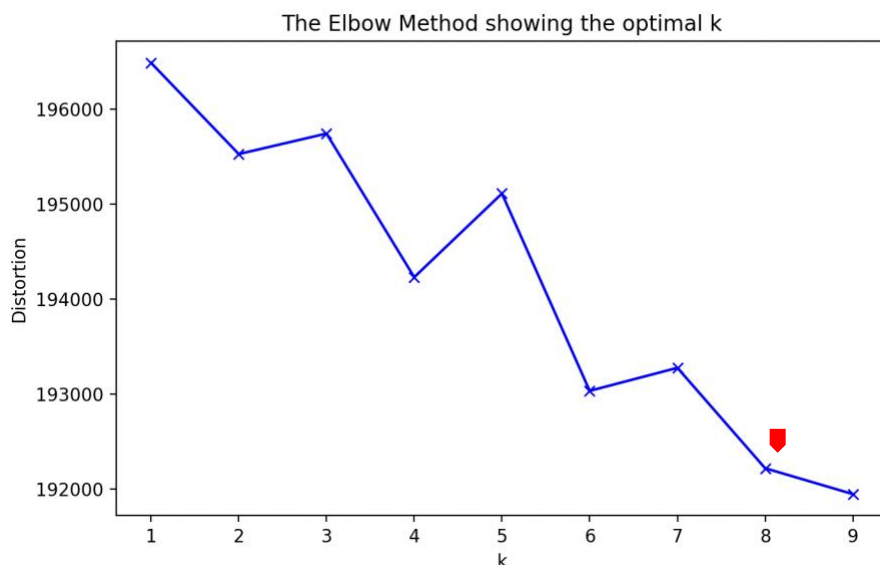


Figure 6: The Distortion Plot of K-Means for given k

Results

This model predicts new book ratings based on the users' ratings of similar previous books with the final aim of recommending books that will be rated highly based on each user's history with books. The methods are separated into steps:

1. Transform the book titles into a TF-IDF matrix using vectorization
2. Cluster the book titles based on the similarity given by TF-IDF
3. Use the title cluster in conjunction with the user ID to create a user-cluster mean rating and user mean rating
4. Predict the new book rating based on the user-cluster mean rating if the new and previous books share the same title cluster
5. Use user mean rating to fill in any gaps left by users reading a new book that does not share any title cluster with their previous books

The accuracy of this model is around 62.4%, this accuracy score is calculated by comparing the actual binned rating and the predicted binned rating. The visualisations of this result can be better seen in Figure 7 and 8.

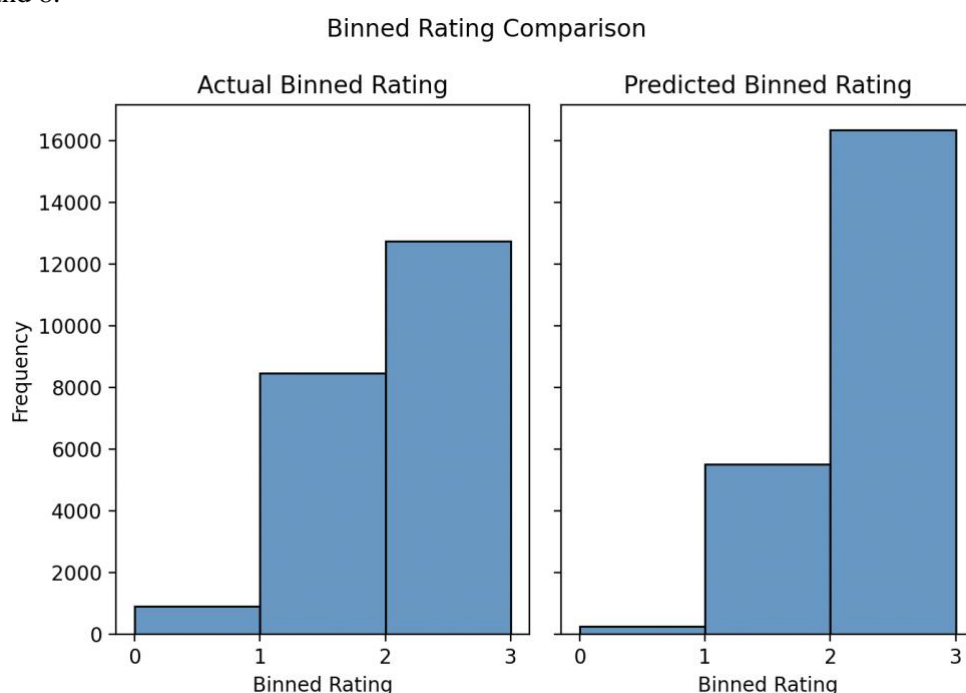


Figure 7: Histogram Comparison of Binned Rating

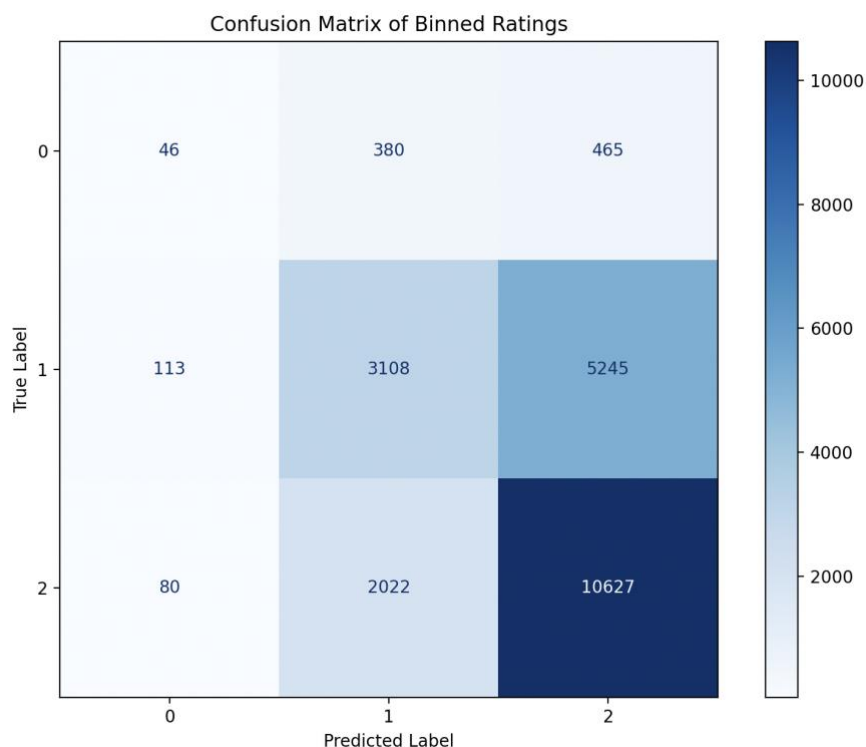


Figure 8: Confusion Matrix Of Predicted Binned Rating and the Actual Binned Rating

The baseline accuracy for this model is around 30%, this is obtained by letting a random number generator predict the rating, this is expected as the binned rating consists of three ranges. While there is a significant increase in accuracy when compared to the baseline, this model's predictive power could not be considered sufficient as a recommender system if it were to be implemented in real-world scenarios. This is evident from Figure 8 above as it can be seen that the model tends to predict the rating to be high and is able to predict those with high actual ratings accurately. However, it fails to accurately predict books with actual ratings in medium and low rating bins, with the correct low rating prediction only amounting to 46 cases. This trend of high ratings being the dominant bin can also be seen in Figure 9.

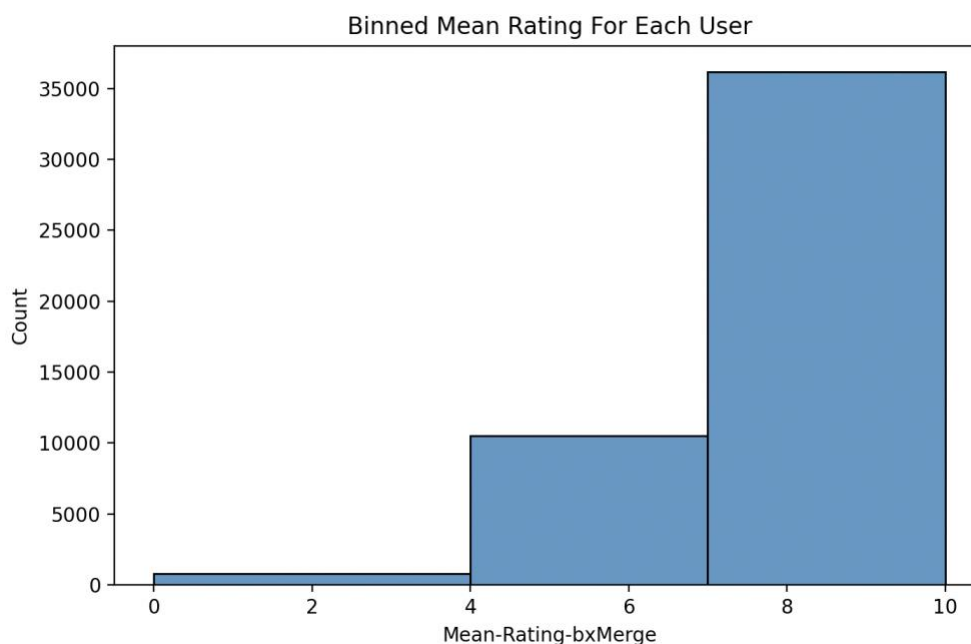


Figure 9: Histogram of binned mean rating for each user

This disparity could be due to the imbalance in the distribution of title clusters, further analysis is present in the limitations section.

Limitations and Improvement Opportunities

In the process of predicting the new book rating, many limiting factors have risen. One such limitation is the model's way of analyzing the book title. The distribution of the title cluster is imbalanced as seen in Figure 10, with one cluster consisting around 84.9% of the whole book title entries, while the rest five clusters combined consist only around 15%.

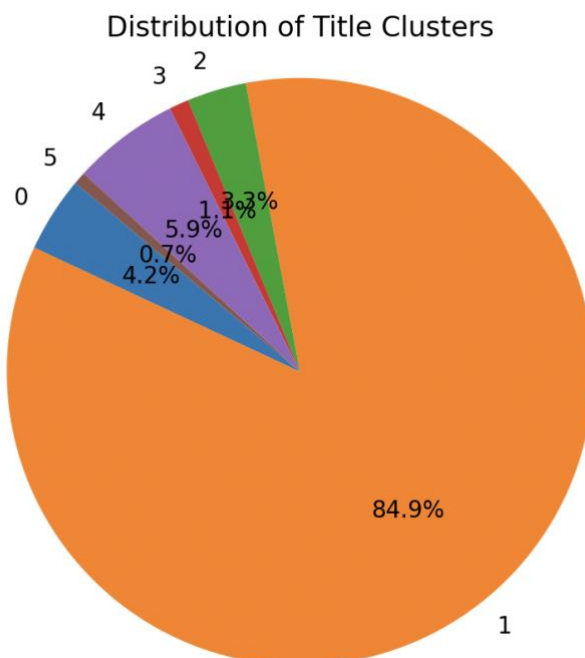


Figure 10: Pie chart showing the Distribution of Title Cluster

This skewed distribution may have been caused by poor pre-processing of book titles, the text pre-processing and the transformation into a TF-IDF matrix could not capture the trend and genres of the books and its correlation with their ratings, or the data itself may already have been skewed from the start.

Another limitation came from the miniscule data each specific user has. While the mean amount of books each user reads is around four books, the median is only one book. This means that the majority of users have only read one book while a minority of users reads a lot of books raising the mean into four books. Therefore, it is hard to predict for a majority of users if the new books are not similar to their previous one. This may be an opportunity to use a User-Based recommender system to find similar users and predict using that similar user-cluster mean rating if the new books happen to fall within the same title cluster read by the similar user but not the original user.

Coming from experience, one limitation comes from our personal desktops, to be more specific the processing power and memory allocation it has. This problem occurs on many occasions when trying to incorporate PCA into our book title processing, and this serves as the reason on why it is not implemented.

The absence of certain data plays a big part in the accuracy of our rating predictions. Additional attributes such as genre would have provided a better description of each book, this would help us identify similarities between each book beyond just the title. Since two or more books may share similar titles but have entirely different content, having the genre of each book would have allowed us to form more precise clusters based on each genre. Moreover, if we had a list of genres each user was interested in, we could simply recommend books with high ratings within the user's preferred genres. This would make our recommendation system simpler yet more accurate.

Discussion and Interpretation

Now that we have the predicted ratings for new books according to a certain user, it provides us with an estimation of how much they would enjoy a book they've never read before. These predicted ratings play a crucial role in creating a recommendation system because it is personalized to each user's preference and taste. If we simply recommended every highly rated book to a user, they would end up overwhelmed with the amount of suggestions and be less likely to explore the books we recommended. This would lead to the decrease of effectiveness of our recommendation system as users could feel that it's useless. Therefore, it's essential to selectively recommend books that are likely to be enjoyed with each user, to maximize positive engagement and encourage them to keep using our recommendations in the future.

However, our predicted ratings, if not accurate, would potentially lead to a lack of trust users have in our recommendations. If a user reads a recommended book and dislikes it, they may lose confidence in our suggestions altogether. Therefore, a crucial part to the success of our recommendation would be the accuracy. With precise predicted ratings, users are more inclined to explore and enjoy the books we recommend, resulting in an increase of satisfaction and engagement.

While our recommendation system currently focuses on recommending books solely from the new books dataset, there's an opportunity to provide a greater variety of book recommendations by incorporating data from the regular books dataset into our system. In books.csv, not every user has read every book, this allows us to recommend books they have yet to read from that dataset. This approach not only provides users with additional recommendations to choose from, but also ensures that the users who are not included in the new books dataset would also receive book recommendations.

As our recommendation system gains more popularity, it will have a positive impact on sales and engagement of all the books in our system. Users are more inclined to purchase the book we recommend to them, which will lead to an increase in the revenue and visibility for both authors and publishers. This is particularly beneficial for less popular authors who often struggle marketing their new releases. Through our recommendation system, we offer these authors a chance to gain recognition and increase the discoverability of their books.

Conclusion

In conclusion, our effort to create a recommendation for books that have no ratings using collaborative filtering showed promising results, through meticulous data pre-processing and descriptive statistics, we successfully predicted book ratings with an accuracy of 62.4% surpassing the baseline accuracy. Despite this success, there's still room for improvement in our recommendation system.

To improve our system, additional data such as genres for each book and each user's genres of interest would be most helpful. By enriching our dataset, we aim to improve recommendation accuracy and provide reasons with even better suggestions. This not only helps readers find books they'll love but also support authors and publishers by making their work more visible. Our goal now is to keep refining our system and make a positive impact on how people discover and enjoy books worldwide.

References

- Wang, A., 2012, *avenews*,
<https://github.com/albertyw/avenews/blob/master/old/data/average-latitude-longitude-countries.csv>
- Duncalfe, L., 2019, *ISO-3166-Countries-with-Regional-Codes*,
<https://github.com/luke/ISO-3166-Countries-with-Regional-Codes/blob/master/all/all.csv>