TITLE: PROJECT #3 – RECOMMENDER SYSTEMS

AUTHORS: KASSANDRA SELLERS, CLARK NECCIAI

GROUP 46 – PREDICTIVE ANALYTICS

DATE: JUNE 25TH, 2024

**Executive Summary**

For this analysis, we were tasked with evaluating the capabilities of various Recommendation Systems such as user-based collaborative filtering (UCBF), item-based collaborative filtering (IBCF), and association analysis. We evaluated a variety of UCBF and IBCF models in recommending book suggestions from data found in the Goodreads Dataset. Ultimately, the IBCF model which used pearson correlation and the twenty-five (25) most similar items yielded the best overall performance. Utilizing association analysis, we detail three rules describing book sets which tend to be frequently purchased together. We conclude with a brief discussion on UBCF, IBCF, and association rules where we detail their differences, applications, implementation challenges, and our team’s preferred approach for this dataset.

**Problem Statement and Approach**

Our primary tasks in this analysis were fourfold:

* Perform exploratory analysis and provide notable insights on users, reviews and books
* Preprocess our Goodreads dataset for running the recommender system models
* Determine the best UBCF and IBCF models and compare their prediction overlap
* Generate three association rules detailing subsets of frequently read books

We began our approach by first evaluating for data missingness and duplicated observations. After we performed data preprocessing over our original dataset, we explored our remaining data and discovered notable aspects of the Goodreads datasets. We then appropriately formatted and evaluated our data for Recommender System modeling. After assessing multiple UBCF and IBCF models, we compared our best performing model’s top five predicted books. Lastly, we generated association rules describing groups of books that tend to be read together.

**Data Preprocessing**

The Goodreads dataset consists of two subsets of data containing information on 10,000 books (Books dataset) and their corresponding 981,756 ratings (Ratings dataset). The variables and their respective percentage of missing values are as follows: *isbn (7.0%), original\_publication\_year (0.21%), original\_title (5.85%), language\_code (10.84%).* Given that these variables will not be used in modeling, nor a reliable means to infer an imputable value readily available, we reasonably concluded that it would be safe to ignore these missing values.

We then proceeded to remove duplicate books from the Books dataset based on their title, finding that some observations had been duplicated two, three, and even four times over. In order to ensure no duplicates remained in the Books dataset, a total of 36 repeat observations had to be removed. Similarly, we removed 2,278 duplicate ratings from the same users within the Ratings dataset which was done by grouping *user\_id* and *book\_id* and then performing duplication removal. Within the Ratings dataset, a total of 3,511 observations which had no reference within the Books dataset were also removed. Lastly, as a means of attaining computational feasibility, we retained only those observations which belonged to users with at least 100 ratings, bringing our total number of ratings down to 164,733. For the remaining observations in both datasets, we examined summary statistics, finding no variables which contained illogical values.

**Exploratory Data Analysis (EDA)**

Given the unsupervised nature of our analysis, it is paramount to engage in a rigorous discovery phrase. Doing so grants us insight into deeper understanding of the inherent relationships in our dataset which may contain useful, explanatory information. During our results generation in the post-modeling phase, we may be able to draw from these insights and reason as to how we arrived at these outcomes. We began our exploratory analysis after taking the aforementioned preprocessing steps.

***Oldest Recorded Books***

In an effort to determine the validity of our negative values of *original\_publication\_year,* we recognized these values as representing texts written before the common era (B.C.E) due to prior knowledge. A few of these books include *The Epic of Gilgamesh, The Illiad, The I Ching or Book of Changes* and *The Odyssey*. Of all books written during this era, the author who had written the most books was Plato with a total of five books.

***Distribution of Ratings***

Our team believed there may be a relationship between the number of ratings a book receives and the quality of that book. We examined the distributions of *ratings\_1, ratings\_2, ratings\_3, ratings\_4,* and *ratings\_5*, finding that indeed, we see from our variables’ distributions that books that had ratings of 1 or 2 had very few ratings. We speculated this to mean that as individuals spread word that a book is poorly written or low quality, that other people tend to not want to read and rate these books, leading to the distribution we see in the chart. For books that had ratings of 3, 4, or 5, their distributions appeared similarly flat but spanning into higher values of the number of ratings. We take this to mean the antithesis; as books are found to be of higher quality, more and more individuals become aware and wish to read and provide ratings.

***Highest Rated Authors and Books***

When evaluating which authors had the highest average ratings, we wanted to include only those authors who have written and published at least five books. We wanted to take a wholistic approach in evaluating an author’s ability to consistently write highly rated books. Some immediately recognizable names which appeared on our list of top five authors were J.R.R Tolkien and J.K. Rowling, who wrote *The Lord of the Rings* and the *Harry Potter* series, respectively. However, the author found to have the highest average rating at 4.71, was Bill Watterson who authored the *Calvin and Hobbes* comic strip. We set no limitations when it came to evaluating which books had the highest ratings, finding that the highest rated book at 4.82 was *The Complete Calvin and Hobbes* written by the previously mentioned Bill Watterson.

**Recommender Systems Modeling**

Our Recommender Systems models will require that our data be in a matrix format with rows and columns pertaining to the *user\_id* and *book\_id* variables, respectively. We mapped our titles to book ids to make interpretation of our predicted values easier. Our matrix will consist of 1192 rows of users and 9230 columns of books. Given that not every user has read a majority of the books made available by Goodreads, it is likely the case that our utility matrix will be incredibly sparse. When we calculate the percentage of actual rating values by dividing the number of ratings present in the utility matrix by the product of our matrix dimensions, we find that a mere 1.5% of the matrix has ratings. Visualizing this sparsity, we see that the *book\_id* value increases, the density of the number of reviews becomes sparse. From this, we might intuitively infer that books added later to Goodreads database result in having fewer reviews, given that they have not be available to have reviews made on them.

With our ratings and books preprocessed and in an appropriate rating matrix format, we examined a few key aspects of our data. Our team began this post-processing exploration by examining the distribution of the count of ratings by users who have at least 100 ratings. This distribution is a gradually-sloping, right-skewed tail. One notable aspect is that there is a sudden bump in the number of reviews from approximately 180 to 195. We had no immediate reasoning as to why this might be the case. We then visualized the distributions of the average ratings made by users and average ratings of the books. Both distributions were both right skewed and had median ratings similar to one another at 3.8 and 3.88, respectively.

**KEY QUESTIONS AND ANSWERS SECTION**

**UBCF and IBCF Modeling**

UBCF and IBCF models, while mostly similar, differ in how they calculate similarity. UBCF calculates the similarity of behavior amongst users while IBCF calculates similarity amongst items (books). During the similarity calculations, both UBCF and IBCF models have the option of using either correlation or cosine similarity as distance functions. They both also use implementation defined parameters representing how many items or users to consider when calculating predictions. In the case of UBCF, the nearest-neighbors parameter (nn) determines the number of considered similar users while in IBCF, the *k* parameter denotes how many similar items to consider.

**Comparison and Evaluation of IBCF and UBCF Models**

We can evaluate the predictive performance of our models by bucketing our recommended items into two categories which compare whether these items were actually rated highly by the user. This approach carries some implicit subjectivity when it comes to defining what a “good” rating might be. Therefore, it is left to us to define what a high rating is by tuning the *good* parameter to a value we feel is appropriate. In our approach, we believed a rating of 4 (*goodRating=4)* would be an appropriate subjective match to what a good book might be rated. We can aggregate the predicted top-N lists and the withheld items liked by the user into a confusion matrix from which we can compare model performance via ROC and Precision vs. Recall plots. Our team built our User and Item Based Collaborative Filtering models by first specifying an overall evaluation scheme consisting of the following parameters:

* method = “split”
  + Specifies the use of a single train-test split during model training
* train = .90
  + 90% of the data will be used in training the model with 10% for testing
* given = -10
  + All but 10 items are given to the recommender for each user in the test set (items held out for testing with the rest being used for training)
* goodRating = 4
  + Ratings of 4 and above are considered good. Used when evaluating performance by bucketing our predictions into a binary ‘*good’* or ‘*not good’* based on the predicted rating value to determine metrics such as precision and recall.

Below we specify our considered UBCF and IBCF models and their corresponding parameters.

|  |  |  |
| --- | --- | --- |
| Model Type | Number of Considered Similar Items(k)/Users(nn) | Similarity Metric |
| UBCF | nn = 10 | Cosine Similarity |
| UBCF | nn = 10 | Pearson Correlation |
| UBCF | nn = 25 | Cosine Similarity |
| UBCF | nn = 25 | Pearson Correlation |
| IBCF | k = 10 | Cosine Similarity |
| IBCF | k = 10 | Pearson Correlation |
| IBCF | k = 25 | Cosine Similarity |
| IBCF | k = 25 | Pearson Correlation |

With our models and evaluation schema specified, we then evaluated our models’ performance over the top-N (N = {5, 10, 15, 20} predictions and examined their results via ROC and Precision vs. Recall plots. We found that the best UBCF and IBCF models were those that had the following specifications:

* UBCF with nn = 10 and uses Pearson Correlation
* IBCF with k = 25 and uses Pearson Correlation

The IBCF model was determined to be the best overall performing model with the highest AUC (based on ROC) and better precision vs recall tradeoff at values of 0.005 and 0.002, respectively.

**Comparison of Best UBCF and IBCF Predicted Top Five Books**

With each of our team’s best UBCF and IBCF models, we made five book recommendations to the first user in our utility matrix. Here we detail our best UBCF and IBCF models’ predicted set of top five book recommendations and corresponding predicted values:

* UBCF
  + Winter’s Tale | Rating: 5.0
  + Green Rider (Green Rider, #1) | Rating: 5.0
  + Midnight Crossroad (Midnight, Texas, #1) | Rating: 5.0
  + Birdman (Jack Caffery, #1) | Rating: 5.0
  + The Man With a Load of Mischief (Richard Jury, #1) | Rating 4.5
* IBCF
  + A Short History of Nearly Everything | Rating 4.55
  + The Drawing of the Three (The Dark Tower, #2) | Rating 4.55
  + Blindness | Rating 4.55
  + The Waste Lands (The Dark Tower, #3) | Rating 4.55
  + Different Seasons | Rating 4.55

As we can see, there is no overlap between our models’ predictions. The UBCF model tends to give higher ratings to its books, while the best performing IBCF model gives a rating of 4.55 to each of its predictions. This behavior of predicting different sets of recommended books can be expected to some extent, given that the way in which these differing models form their recommendations via user-to-user (UBCF) or item-to-item (IBCF) similarities.

**Association Rule Modeling**

In addition to Collaborative Filtering, our team repurposed the preexisting utility matrix in an effort to generate association rules with Association Analysis. Our original utility matrix had ratings which were not only 1’s and 0’s, which is the required for this type of analysis. We revised our matrix into an appropriate format where NA values were changed to 0’s and any ratings were changed to 1’s. This changes the inherent interpretation of the matrix: whether or not a user has read a particular book. Utilizing the apriori algorithm to determine association rules, we specified a minimum support level of 0.046 and a confidence of 1 finding many generated rules with high values of lift. We ordered our rules by lift values, of which high values suggest that the presence of one book set (A) increases the probability of another book set (B) more than chance. Below we show the top three rules generated by the apriori algorithm and support and confidence levels where interestingly enough, each of the sets of books came from the same series of books (*Wheel of Time* and *Sookie Stackhouse)*:

**Association Rule #1 with Lift Value: 17.27536**

Set A: {A Crown of Swords (Wheel of Time, #7), Winter's Heart (Wheel of Time, #9)} =>

Set B: {The Path of Daggers (Wheel of Time, #8)}

**Association Rule #2 with Lift Value: 17.27536**

Set A: {Lord of Chaos (Wheel of Time, #6), Winter's Heart (Wheel of Time, #9)} =>

Set B: {The Path of Daggers (Wheel of Time, #8)}

**Association Rule #3 with Lift Value: 17.** **02857**

Set A: {Living Dead in Dallas (Sookie Stackhouse, #2),

From Dead to Worse (Sookie Stackhouse, #8)} =>

Set B: {Dead as a Doornail (Sookie Stackhouse, #5)}

**Discussion on Recommender Systems**

At a high level, UBCF and IBCF models differ from association analysis, in that they identify similar users or items based on metrics such as cosine similarity or pearson correlation in order to make personalized recommendations. UBCF and IBCF differ from one another in that the former calculates and makes predictions based on similarity between user-to-user and the later does the same but for item-to-item. Association Analysis merely identifies frequent item sets of frequently occurring items in the form of rules based on support and confidence levels.

Business Application

Challenges when implementing?

Recommended Approach?

**Conclusion**

**Appendix**

**Figure #1: Distributions of Different Rating Scores**

**A graph showing a number of ratings

Description automatically generated**

**Figure #2: Top Rated Authors**A graph with blue and black bars

Description automatically generated

**Figure #3: Top Rated Books**

A graph of a number of books

Description automatically generated with medium confidence

**Figure #4: Final Rating Matrix with High Sparsity**

A close-up of a graph

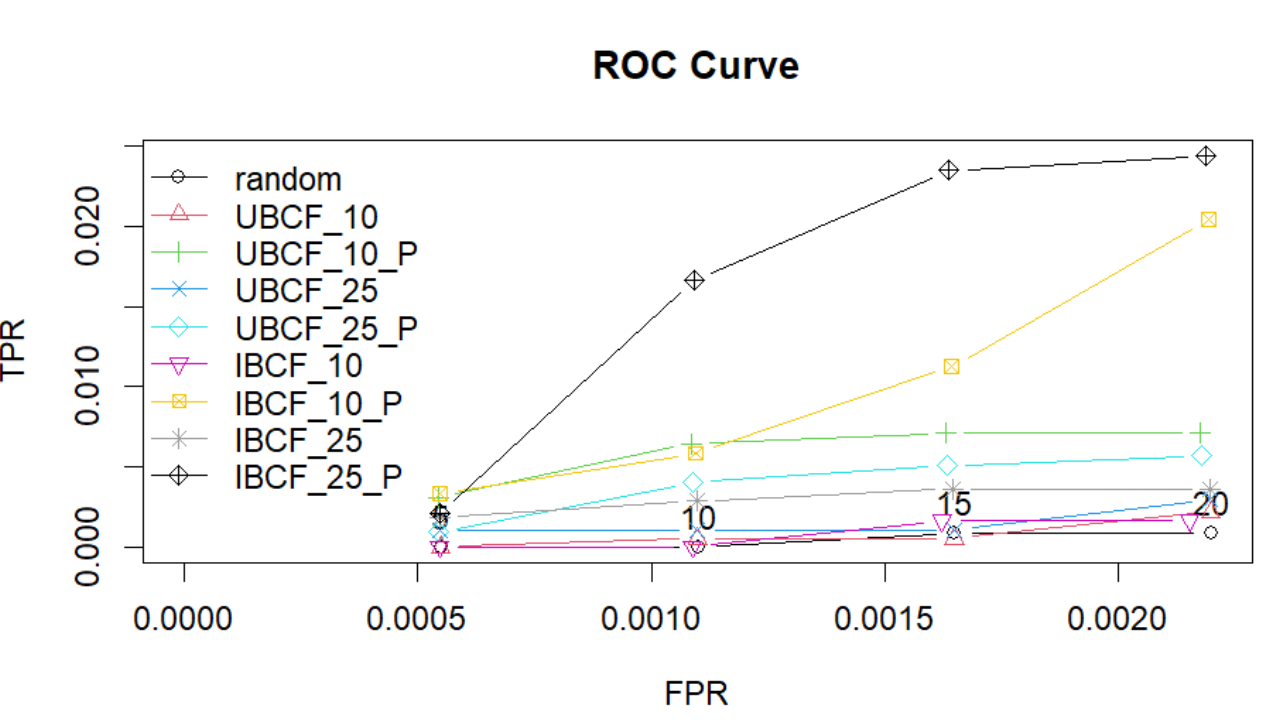
Description automatically generated

**Figure #5: Distribution of Submitted Ratings by Users with at least 100 Ratings**

**A graph with numbers and a red line

Description automatically generated**

**Figure #6: ROC Comparison – IBCF\_25\_P and UBCF\_10\_P Optimal Recommender Models**

****

**Figure #6: Precision vs. Recall – Recommender Models**

**A graph of different colored lines

Description automatically generated**