CSP 571: Project Proposal and Outline

4 October, 2018

1 Members

- Eshna Sengupta (A20422016)
- Pooja Mehta (A20399008)
- Sushmitha Sekuboyina (A20365741)
- Vivekanandan Sakthivel (A20432987)

2 Project Proposal

2.1 Formal description and stated research goal

Don't you hate it when you are looking
for a good book, but can't find one to read?

In order to address the aforementioned problem, we decided to build a recommendation system for books using R. A recommendation system, in essence, is an automated decision engine that predicts the *rating* or *preference* a user would give to a book. Our goal is to create such a product using certain specialized algorithms.

2.2 Proposed methodology/approach

Our first step is to perform *Data Exploration and Data Visualization* to discover the nature of the data and gain meaningful insights from it, such as: most popular books, most active readers, average ratings of each book, etc. *Ggplot2* is used to visualize the data.

After gaining knowledge of the data we proceed with *Data Preparation and Cleaning.* To do this, we select the most relevant data. For example, we may choose to ignore books that have been viewed only a few times, since their ratings might be biased because of lack of data. We then normalize the data, in order to reduce the bias of the ratings.

Once the data is in the correct format, we are ready to *Build the Recommendation Model* using the Collaborative Filtering algorithms for recommender systems.

Our final step is to *Evaluate the Recommender System* by evaluating how well it predicts ratings, or how good its recommendations are. We also plan on comparing the different models and identifying the most suitable model. Optionally, we may attempt to *Optimize the Model*.

Our finished product will be a fully functioning book recommender which is *Deployed using RShiny*.

2.3 Set of metrics which will measure analysis results

Our dataset contains books that are marked by the users as “to read”. Once we train our recommendation system, we will compare the top recommendations suggested by the model to the users’ actual “to read” list. The weighted average percentage of match will be the accuracy measure to evaluate the performance of the model.

3 Project Outline

3.1 Data

Data Source: goodbooks-10k.
Dataset Type: There are five comma separated value (CSV) files as follows:

Document Name	Description	No. of Attributes
books	Includes metadata used to describe the books and ratings received	23
books_tag	Contains tags assigned by users to the books	3
tags	Contains information needed to translate tagIDs to names	2
to_read	Contains IDs of the books marked as “to read” by each user as (user_id, book_id) pairs	2
ratings	Contains ratings given by each user to the books	3

Dataset size: There are two important data sets – **books** and **ratings**. The books dataset comprises of records pertaining to 10000 books and ratings consists of approximately six million records. 53424 users have provided ratings for books, with each user providing at least two ratings. Each book has at most 100 reviews with ratings ranging from one to five.

Feature Description: The features of the *Books* dataset are as follows:

Feature	**Data Type**	**Description**
book_id	Character	Continuous value assigned to each book
Authors	Character	Authors of the book
original_publication_year	Numeric	Year in which the book was published
ratings_count	Numeric	No. of ratings received for the book
Title	Character	Title of the book
average_rating	Numeric	Average rating of the book
work_text_reviews_count	Numeric	No. of text reviews received for the book
ratings_1	Numeric	No. of 1-star ratings
ratings_2	Numeric	No. of 2-star ratings
ratings_3	Numeric	No. of 3-star ratings
ratings_4	Numeric	No. of 4-star ratings
ratings_5	Numeric	No. of 5-star ratings

The features in the *ratings* dataset are:

Feature	**Data Type**	**Description**
book_id	Character	Book id assigned to book
user_id	Character	User id assigned to each user
Rating	Numeric	Rating given to that book by user ranges from 1 to 5

There are no missing values in the dataset. However, some pre-processing will be required since we noticed that in the *tags* document, the tag_name has numbers as well as symbols.

3.2 Data Processing

According to Figure 1, there are users who have rated fewer books than the average. We assume that better model results will be achieved by ignoring the outliers below the interquartile range in the distribution.

After removing the outliers, we prepare a *Ratings Matrix*, where the users form the row space and books form the column space of the matrix. The values in the matrix are the ratings given by the users for the books. We use row means and column means imputation to fill the NA's. Another solution is to create an *Affinity Matrix* to capture the users' particular tastes as compared to the general taste of all book readers.

For each of the possible matrices, a preliminary clustering is performed to get a rough idea of the distribution of data. The matrix that gives best results is then used for training the recommendation system using the **R** library *recommenderlab* .

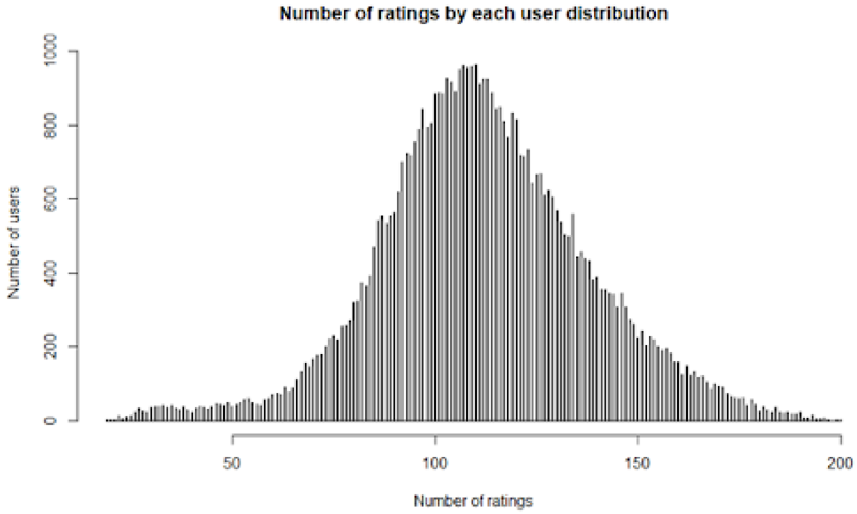

Figure 1: No. of ratings by each user

3.3 Model Selection

Although there are several models to chose from, we selected the Collaborative Filtering technique to build the recommendation system. There are two types of Collaborative filtering algorithms: *Item-based* and *User-Based*. Item-based algorithms recommend items that are similar to items the user has purchased in the past, whereas user-based algorithms recommend to user, items that are most liked by similar users. We plan to implement both these algorithms, and choose the one that performs best.

3.4 Tools

- Software packages: *RStudio, Tableau*
- Applications: *RShiny*
- Libraries: *ggplot2, recommenderlab, dplyr, reshape2*
- Development : *GitHub*

3.5 Related Work

Recommendation systems have become an important research area due to its abundance of practical applications that help users deal with overload of information. Few of the applications of recommendation systems include: movies, books, Amazon products, etc.

The commonly adopted formulation of the recommendation system was to utilize the similarity between users and/or items. However, after years of extensive studies, the commonly used methods have been classified into the following categories: collaborative filtering method and content based method. In contrast to collaborative filtering, content based filtering sorts items based on the users past preferences by analyzing the item description.

Here are some reference materials that we found useful:

- Relevant Papers: [1], [2], [3], [4], [5], [6], [7].
- Books: [8], [9].
- Tutorials: RecommenderLab Tutorial, How to build a recommendation engine in R, Movie Recommender.

References

[1] C. C. Aggarwal, J. L. Wolf, K.-L. Wu, and P. S. Yu, "Horting hatches an egg," *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 99*, 1999.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, p. 734–749, 2005.

[3] F. Ricci, "Recommender systems: Models and techniques," *Encyclopedia of Social Network Analysis and Mining*, p. 1–12, 2017.

[4] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, "Content-based recommendation," *Recommender Systems*, p. 51–80.

[5] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," *ACM SIGIR Forum*, vol. 51, p. 227–234, Feb 2017.

[6] M. Balabanovic and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, p. 66–72, Jan 1997.

[7] D. Cosley, S. K. Lam, I. Albert, J. A. Konstan, and J. Riedl, "Is seeing believing?," *Proceedings of the conference on Human factors in computing systems - CHI 03*, 2003.

[8] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, *Collaborative Filtering Recommender Systems.* Now Publishers, 2014.

[9] S. K. Gorakala and M. Usuelli, *Building a recommendation system with R: learn the art of building robust and powerful recommendation engines using R.* Packt Publishing, 2015.