For my 15-112 term project, I designed a book recommender system called “Radvice”. A recommender system is essentially an information filtering system. Recommender systems seek to predict the user’s rating for a given item, given information such as his/her past rating, and the ratings of other individuals.

There are two main types of recommender systems: Collaborative and content-based. In my project, I have used **collaborative filtering** in order to determine which books a reader is likely to enjoy reading.

**Collaborative filtering** **Content-based filtering**

-Uses past behavior of user and -Uses characteristics of an object: for

similar behavior of other users. Instance, if like fantasy books, more

fantasy books will be recommended to me.

1 2 3 4



User 1

User 2

User 3

User 4

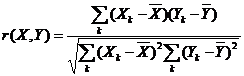
User 5

Consider the above image taken from <https://en.wikipedia.org/wiki/Collaborative_filtering>.

Here, a table has been constructed, with the users and the respective object. Entries in the table include whether or not the person likes the item with a question mark, based on his ratings of other items, and based on other users’ ratings.

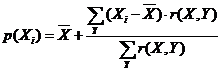
* Here, we see that user 5 and user 3 seem to have similar ratings; they both items 1 and 2. So, their correlation is high.
* User 2 and 5 also seem to have similar ratings. So, they also have a high correlation.
* On the other hand, user 1 and user 5 have a low correlation, because their ratings are seemingly dissimilar; user 1 dislikes 2 and likes 4, but user 5 likes 2 and dislikes 4.
* Based on this observation, we see that user 5’s choices are likely to be the opposite of user 1’s, and similar to User 2 and 3.
* The cumulative effects of these correlations leads to the prediction that user 5 will dislike item 3.

For my project, I have used the following correlation formula, from the following website (<http://recommender-systems.org/collaborative-filtering/>)



The LHS depicts the correlation between two users, X and Y. The numerator on the right hand side is the summation of the product of the difference between user1’s rating and his average rating and user 2’s current and average rating, for all items that both of them have rated.

The denominator is the square root of the product of the summations of the square of the difference between user 1’s rating and his average rating, and the summation of the square of the difference between user 1’s rating and his average rating



To get person X’s particular rating for object i (Xi), we add X’s mean rating to normalize the predicted rating.

I modified this formula by substituting Yi – Yaverage instead of Xi – Xaverage, as it doesn’t make sense to predict X’s rating if we already have his rating

After obtaining the correlation of every user with every other user, I constructed a prediction 2D dictionary, which, for every user, contained their predicted rating for every single book they had not read. My book list contains about 100,000 books.

All of the data I analyzed for this project was obtained from <http://www2.informatik.uni-freiburg.de/~cziegler/BX/>

More specifically, I used the “CSV Dump” which includes three excel sheets, one mapping ISBN number to books, and the other mapping users to their ratings ISBNs.

This is a free-use, open database, and I obtained permission from Professor Cai-Nicolas Ziegler, and informed him about my intent to use the source.

Advice given to me by my peers during the user study on Tuesday (features that I will try to implement in the future, and would have tried to implement had time permitted it):

* My peers were mostly happy with the design (when I showed it to my friends the previous week, they said it was unsightly, but I took their advice and improved the UI).
* One of my group members suggested that I should have an age filter. While using Radvice, she searched “Pokemon” and rated a bunch of different books. Among the ratings she received several animal-related texts, and one inappropriate title. However, for me to implement this, I would have to use not only collaborative but also content based filtering, and I did not have access to this information in the current data set. In the future, I will try to implement a feature that allows users to tag books and sort them categorically, so that they only receive appropriate recommendations.

Discussion of my user interface:

* I have two main modes; one that allows a person to look at and rate the most popular books, and another that lets a user search for the book they want.
* While designing my UI, I studied the following websites:

Amazon, Goodreads, and What Should I Read Next. What Should I read next only allowed a user to input one book at a time, which, according to me, provided a very limited spectrum of recommendations based solely on that one title. So, I ask the user for at least ten books because a) my program requires a significant amount of data to make meaningful recommendations and b) also to provide the user with new and exciting recommendations that may be unexpected. Because just because a user liked Harry Potter does not mean they want to read <insert popular fantasy book here>.

* An issue that I found with Amazon was that it is hard to tell on what basis books show up in your feed (whether based on your past behavior, etc.) and there is no way to view the most popular books of all time.
* Goodreads was a perfect combination because it allows you to see both the most popular books and search. So, I tried to make Radvice follow the Goodreads model.
* I added a cool graph feature which plots your ratings against the book averages. While all these websites allow you to look at average ratings, mine is more explicit because it shows you what you rated the book at the same time.