**Data Science for Industry Project 1**

**Due date: 29August 2018**

The Rdata file “book\_ratings.Rdata” (on Vula) contains ratings (on a scale of 0 to 10) of 150 books from 10,000 users. This is a modified version of the “Book-Crossings” dataset (http://www2.informatik.uni-freiburg.de/~cziegler/BX/). The Rdata file contains the following three objects:

* *book\_ratings*: data frame containing unique ID variable for identifying users (User.ID), a unique ID variable for identifying books (ISBN), and the book ratings (Book.Rating). Ratings are either integers (0 – 10) or NA. You can assume an NA means the book has not been read.
* *book\_info*: data frame containing the title (Book.Title) and author (Book.Author) for each ISBN.
* *user\_info*: data frame containing additional demographic information (Age) for some users. You don’t need this data to build the recommender but if you want to go a bit further you can try including this information.

The aim of this project is to build a recommender system based on the subset of the Book-Crossings dataset provided to you. The function of the recommender system is to recommend books to a user that, on the basis of his or her past evaluations, they might enjoy.

The recommender system should be able to provide recommendations both for existing users, and for a new user. For a new user, you can assume that they evaluate a small number of books (less than 5, say), and present their evaluations to you in the form of a data frame. For example, the new user may see three books (ISBN 0440234743, 0971880107, and 0345417623) and rate these as 2, 5, and 3 respectively. The associated data frame would be something like:

new\_user <- data.frame(ISBN= c(0440234743, 0971880107, 0345417623), Book.Rating = c(2, 5, 3))

You then need to recommend a book that you think they will like.

Specific objectives are:

* To build recommender systems based on each of item-based collaborative filtering, user-based collaborative filtering, and matrix factorization;
* To assess the accuracy of the matrix factorization recommender system using cross-validation;
* To use cross-validated performance of the recommender system above to assess the influence of adding L2 regularization and bias (i.e. assess accuracy with and without these features);
* To write R functions to do each of the above, and to **build an R package** containing and documenting your functions.

Write up your work in the form of a report **written in R Markdown**. The report should contain a description of the problem, the approach you took, and your results. Your code should be integrated into the document, and this code should be clearly described and commented (see the class .Rmd notebooks for examples). This can be added as a vignette in your R package or submitted as a separate document.

Notes and hints:

1. This project must be done in R, since one of the objective are building an R package.
2. You should not use any special purpose recommender systems R packages (e.g. recommenderlab)
3. The factorization of a large matrix such as this one (10 000 x 150) is computationally intensive, and the methods we used in class may take a very long time to run and/or run into memory problems: they are not efficient. You will probably need some way of making the code provided in class more efficient.
4. If you are struggling to make progress with the full dataset, try using fewer users. Taking a subset of a few dozen users turns the data into something similar to what we used in class. Reducing the number of columns isn’t a great idea (as this means some books can never be recommended). Reducing rows/users will probably increase prediction error (as you have less data) but does not fundamentally alter the task. Note that this is not a recommended strategy, but an option if you are unfamiliar with R and feel a bit stuck.
5. Anyone else should be able to run the code in the R markdown document to completion – if necessary use *set.seed()* to set a random seed so that your final results don't change.

The submission deadline is on or before 29 August 2018. Details on how/where to submit to follow.