UNIVERSITY OF CAPE TOWN
DEPARTMENT OF STATISTICAL SCIENCES
STA5073Z – Data Science for Industry 2025

Assignment 2
Recommender systems

This project must be done individually
Due date 13 October 2025 12 pm (noon)

Project objective

The aim of this project is to build an ensemble recommender system that recommends to users books they might enjoy, based on their past book evaluations. The data you will be working with is a modified version of the "Book-Crossing" dataset. This subset, which has been made available in the file `book_ratings.RData`, contains ratings on a scale of 0 - 10 of 150 books from 10,000 users. The following 3 objects are available:

- `book_ratings`: Data frame containing a unique ID variable for identifying users (User.ID), a unique ID variable for identifying books (ISBN), and the book ratings (Book.Rating). Explicit ratings are gvien as integers (1 – 10). Zero ratings are implicit, meaning that the book was read, but not rated.
- `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 do not need this data to build the recommender but if you want to go a bit further you can try including this information.

The recommender system should be able to provide recommendations both for existing users and new users. For a new user, you can assume that they will provide (explicit) ratings for a small number of books – say, 5 or fewer – when joining the platform, thereby avoiding the cold-start problem in this exercise.

Some EDA is expected – you must always explore the structure and scope of your data. This will also help inform the splitting of the data into training and testing sets.

Assignment objectives and deliverables

1. To build recommender systems that predict the rating a user will give to a book based on each of item-based, user-based, matrix factorization-based, and neural network-based collaborative filtering algorithms.
2. To compare the accuracy of these kinds of recommender systems using cross-validation.

3. To provide some evidence-based guidance on the relationship between the number of titles a book recommender system needs to contain and predictive accuracy. For example, if you build a recommender system based on just 5 titles, how different is the accuracy to a case where you have data on 50 titles, or 100? Is there a point after which adding further titles does not improve accuracy?

4. To write a short scientific report (word limit: 4500) on this work. This can be, be does not need to be, in the form of a quarto markdown document. There is no website or version control required for this assignment.

Submission guidelines

On Amathuba, please submit the following two (possibly more) files, using your **STUDENT ID as the files' names** –

1. `ABCXYZ001.xxx` is your scientific report where `ABCXYZ001` is replaced by your student number. This can be in whatever format you like e.g. a word document, typeset latex pdf, html, etc.

2. `ABCXYZ001.yyy` is the code used to generate your results. This might be the .qmd file used to render your report, or a standalone .R/.py script. It is vital that I can run these files myself i.e. your results in your report should be fully reproducible (you can assume I have the `book_ratings.RData` file). If you've used multiple scripts, then upload these and provide information on how to use them.

Use of AI

Use of an LLM is encouraged and is an important part of the assignment. See the separate document on the course policy on LLM use for assignments for further details.

Notes

1. You should code your own item-based and user-based collaborative filtering recommender systems without the use of special purpose recommender systems packages.

2. The factorization of large matrices is computationally intensive, and the methods we used during lectures may take a very long time to run and/or run into memory problems. They are not efficient; you will need some way of making the code provided more efficient. You may use the `recosystem` package for this step, or write your own custom gradient descent function.

3. 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 rows/users will probably increase prediction error (as you have less data) but does not fundamentally alter the task. This is not a recommended strategy, but an option if you are getting stuck.