Recommender Systems

Course Syllabus for DSCI 641

AT A GLANCE

Course

Course Title	DSCI 641: Recommender Systems for Data Science
Credits	3 (min. 10 hrs/week of class and work)
Schedule	Tue 6:00–8:50 PM
Office hours	By appointment (schedule here)
Readings	Posted to Blackboard (from library & online articles)
Prerequisite	DSCI 521 (*Data Analysis and Interpretation*) or CS 613 (*Machine Learning*)
Software	Python, PyData, and LensKit

REVISION LOG

Jan. 8	Clarified that there is no final exam Revised software notes

COURSE INFORMATION

Course Description

Recommender systems help users to discover new products and services. The goal is generating meaningful recommendation to a collection of users with items or products that might interest them. Recommender systems are encountered on multiple domains such as e-commerce, content and media distribution, social media, and more. The course will cover fundamental and practical aspects of Recommender systems focusing on the data science approach. The course includes topics and concepts for recommender systems: collaborative filtering, content-based recommendation, knowledge-based recommendation, hybrid recommendation, attack-resistance recommendation, and evaluation of the recommender systems. Students will gain hands-on experiences with assignments and a term project.

Course Purpose Within a Program of Study

This course is an elective available to graduate students, in particular students in the MS Data Science and MS Information Systems programs. Students will gain understanding on different approaches of recommender system development using content, knowledge, and collaboration.

Course Learning Outcomes

Upon successful completion of this course, a student will be able to:

1. Understand the basic concepts of recommender systems
2. Solve mathematical optimization problems for recommender systems
3. Carry out performance evaluation of recommender systems based on various metrics
4. Implement machine-learning / data-mining algorithms in recommender systems data sets.
5. Design and implement various types of recommender systems.
6. Learn about advanced topics and current applications of recommender systems.

COURSE MATERIALS

Recommender systems is a rapidly-evolving topic that has seen extensive change in the last half-decade or so, and many of the available textbooks are somewhat old. Therefore, we will be using a combination of textbook chapters and other resources as the readings for the course. Specific readings will be listed in Blackboard for each week. Some readings will be fundamental to learning the material, while others are additional perspectives or resources for further study on topics of interest; each reading will be labeled.

Textbooks

We will be working with the following textbooks; one is free on the author's web site, and the other three are available online through the university library:

Julian McAuley, *Personalized Machine Learning* (PML). Cambridge University Press, 2022. [author version]

Charu C. Aggarwal, *Recommender Systems: The Textbook* (RST), Springer, 2016. ISBN: 9783319296579 [library link]

Dietmar Jannach, Markus Zanker, Alexander Felfernig, & Gerhard Friedrich. *Recommender Systems: An Introduction* (RSI), Cambridge University Press, 2010. ISBN: 0521493366 [library link]

Kim Falk. *Practical Recommender Systems*. Manning, 2019. [library link]

Additional Books

These books are useful if you want to dive deeper into some of the topics we discuss:

Francesco Ricci, Lior Roach, and Bracha Shapira, eds. *Recommender Systems Handbook*, 3rd Edition. Springer-Verlag, 2022. ISBN: 978-1-0716-2196-7.

Deepak Agarwal and Bee-Chung Chen. *Statistical Methods for Recommender Systems.* Cambridge, 2015. ISBN: 978-1-13-956586-8.

You may also find this survey useful to read — it is older, but covers many of the fundamentals:

Collaborative Filtering Recommender Systems by Ekstrand, Riedl, and Konstan (now publishers; *Foundations and Trends in Human-Computer Interaction* **4**(2)). Available free from the author's web site.

Research Readings

Some of the readings will be research papers. This is because it takes time for knowledge to make its way from current research to textbooks, and recommender systems is advancing and evolving quickly as noted above. In the context of the class topic, therefore, Learning Outcome 5 involves being able to understand and apply new ideas in recommender systems from papers, talks, and other current resources to improve an application's recommendations.

Software Resources

This course contains several programming assignments in Python using the PyData stack (NumPy, Pandas, etc.) and the LensKit recommender systems library. For more information on programming with PyData, *Python for Data Analysis* by Wes Kinney is a good resource. Links to documentation for the different libraries we use will be in Blackboard.

There are several options for running your code for the assignments and project:

- Your own computer (recommend at least 16GB memory, but most assignments can be completed with 8; an Nvidia GPU with 6+GB memory is helpful for later assignments)
- Tux (CCI lab servers, ample RAM but no GPUs)
- Google Collab (provides limited GPU and TPU capacity)

I will provide project files with dependency lists to facilitate software installation with '`uv`' (recommended) or '`pip`'.

LECTURE LOGISTICS

In-person lecture is on Tuesday evenings from 6-8:50 PM. This class is recorded for students in the online cross-listed section.

Online students will work from lecture videos, which I will be posting each week by end of day on Wednesday. Online students are also welcome to watch live through Echo 360.

When I am traveling (see schedule), class will be online for all students.

ASSESSMENT PLAN

The work in this course falls into several categories:

- Short **quizzes** to check your learning along the way.
- **Assignments** to give you experience building, testing, and analyzing recommender systems.

- A **project** to building a recommender or carrying out a new experiment.
- A **midterm** essay exam.

Your final grade will be computed from these components as follows:

Category	%
Assignments	50
Project	30
Quizzes	10
Midterm	10

Grade Scale

The minimum points to achieve each letter grade are as follows:

Points	*Grade*	*Points*	*Grade*	*Points*	*Grade*
97	A+	82	B	70	C-
92	A	80	B-	67	D+
90	A-	77	C+	60	D
87	B+	72	C	0	F

Assignments

There will be **4 assignments** throughout the semester. Many of these assignments will require you to implement and/or test one or more recommender algorithms using the LensKit toolkit and Python data science / machine learning tools (the PyData stack). Some assignments will require a Jupyter notebook as part of the assignment to present your results. These assignments primarily assess learning outcomes 1–3.

As the course progresses, the assignments will begin to require **significant compute time**. It will likely take several hours to run the code for some of the assignments. I therefore strongly encourage you to begin early so that you have enough time to run your code and complete the analysis before the assignment is due.

Each assignment is due at **midnight on Monday** the week in which it is due.

You may work with a partner on the assignments, and I encourage you to do so. You and your partner may collaborate with other classmates on the **concepts** of the assignments and on debugging your code, but you **must submit your own work**. In each assignment submission, list your team and all classmates you collaborated with on that assignment. You may also borrow pieces of code, so long as (1) you clearly describe the origin of any code you did not write yourself in comments (including code written with the help of Copilot and similar tools); (2) borrowed

code does not comprise more than 45% of the total code in your submission; and (3) the code is borrowed in a manner compliant with its copyright and license.

Project

The final component of this class is the project, in groups of up to 3. Each group will prepare a written report and video presentation of their project. More details will be available in the project description released in the second week of class.

Exam

This course has a midterm essay exams. You will have 48 hours to complete the midterm this exam, which is designed to assess your ability to define and analyze a recommendation problem.

Quizzes

There will be 5 short quizzes throughout the term to check your understanding of the material as we go, one every other week.

COURSE POLICIES

Announcements

I will post course announcements to Blackboard Announcements and to Discord, including any changes to the syllabus or assignments. You are responsible for making sure that you receive course announcements in a timely fashion. If I need to change the syllabus or an assignment description after its initial publication, I will include a dated Revision Log in that document describing the modifications.

Late Work

You have a budget of 4 **late days** to use at your discretion throughout the semester (with a max of 3 for any individual deliverable). Each day extends a project, assignment, or exam deadline by 24 hours, no questions asked. For assignments with a partner, you only need to use one team member's late days. When using late days, mention in your assignment submission that you are using them. I appreciate advance notice that you plan to use them, but this is not required. When working with a partner, you can use either partner's late days.

Late days are intended to cover most ordinary need for extensions, including such things as illness or child care needs, and additional individual extensions will rarely be granted.

Conduct

I expect you to respect me and your fellow students in all class interactions, both in official meetings such as lectures and out-of-classroom activities such as project group meetings and study sessions, and to contribute to a constructive learning environment.

In addition to the Drexel Conduct and Community Standards, the Recurse Center Social Rules are a good source of guidance on how to maintain a constructive and educational environment in a computing learning context.

Academic Integrity

As an information professional or data scientist, you are expected to do your own work, attribute sources, and respect the legal and moral rights of others with respect to their work; as a student, you are also required to abide by the university policies regarding academic integrity. While I aim to allow you to make reasonable use of resources, cheating (including copying code, using unauthorized resources during tests, etc.) is not acceptable. If I find you to be cheating, the penalty may range from an F on the assignment to an F on the course, and will be reported to the university.

You may consult external resources such as other books and web sites for understanding how to solve assignments or portions of the project. In your assignment submissions, list all external resources you used; if they are available online, provide the URL. You do not need to cite the textbooks or the official documentation for the software we are using. You must specifically declare any use of generative AI (e.g. ChatGPT or GitHub Copilot); see the Artificial Intelligence policy below for further details on use of AI.

Besides the course forum, you may ask questions related to the course material and concepts required to complete the work on publicly accessible discussion forums such as Stack Overflow, newsgroups, or publicly archived mailing lists. To qualify as publicly accessible, a site must provide access to complete discussions without requiring payment or registration. Provide URLs to the forum discussion on the relevant web site or archive with your assignment submission. When you ask a question for one of the assignments, mention that it is for a course project and that your instructor permits you to make reasonable use of discussion forums.

A good question will ask about how to go about a particular sub-portion of the problem, how something works, why something you are trying doesn't work, or other specific difficulties. Do not ask "how do I solve <the problem description>?", or similarly direct translations of the project requirements, or for specific code. Questions should be written to fill in a gap in your understanding that will then enable you to continue your work, not to get a solution to the assignment.

This course is also subject to the Drexel Academic Integrity, Plagiarism, Dishonesty, and Cheating Policy; the above does not replace or modify that policy, but is intended to clarify how the university policy is applied to this course.

Permitted Use of Artificial Intelligence

You are allowed to use generative AI tools such as ChatGPT and GitHub Copilot in the assignments and project for this course, although I do not encourage you to do so. If you use generative AI, you must clearly state that you did so, and document which portions of your submitted work were produced, drafted, or substantially edited by AI. If you use a tool such as ChatGPT, save a record of your prompts and the system's responses, and provide it as an

appendix to your submission. The Drexel Policy on Academic Integrity Pertaining to Artificial Intelligence provides further details on university policies regarding AI.

Disability Accommodations

If you need particular accommodations to be able to fully participate in this course, please talk with me as soon as possible. If you have documentation from Disability Resources for particular accommodations, please bring it, but I am happy to discuss with you anything needed for you to fully participate in the class.

Office Hours

My office hours are by appointment, arranged through my Bookings page. There is a link to this page at the beginning of the syllabus and in Blackboard.

Course Changes

I may need to make changes to the course as the term progresses to better support your learning and the logistics of delivering the course. Such changes will be announced through Blackboard and Discord, as well as mentioned in lecture when the timing of the change permits.

SCHEDULE

The following is the planned schedule of topics and due dates for the term.

Week	Date	Topic	Due
1	1/7	Fundamental Elements of Recommendation	
2	1/14	Recommending from First Principles	
3	1/21	Evaluating Recommender Systems	A1
4	1/28	Matrices, Embeddings, and Optimization	*Proposal*
5	2/4	Content and Knowledge	A2
6	2/11	Sessions, Sequences, Contexts, and Time (*online*)	
7	2/18	Explanations, Interfaces, and Human Data	A3
8	2/25	Robustness, Trust, and Privacy	*Midterm*
9	3/4	Productionizing Recommender Systems	A4
10	3/11	Bias, Ethics, and Fairness; *Project Presentations*	*Presentation*
F	3/16	*Finals Week*	*Report*

Assignments (A*n*) and the midterm are due **on Monday at midnight** the week they are listed. The final will be released at noon on Monday and is due by noon on Thursday. Project due dates are specified in the project description, and quizzes will be listed in Blackboard.