Laboratory Medicine & Pathobiology
UNIVERSITY OF TORONTO

Course Name	**Basic principles of machine learning in biomedical research**
Coordinator(s)	Rahul G. Krishnan: rahulgk@cs.toronto.edu Sana Tonekaboni: stonekaboni@cs.toronto.edu
Day and Time	**Thursday, 10:30a – 12:30p**
Location	Sydney Smith building Room 179
Prerequisites	Undergraduate level probability, statistics, multivariable calculus, linear algebra
Who can attend	You must be registered in a graduate program to attend this course. This course is open to all graduate students at the University of Toronto, provided you have pre-approval from your department and the course coordinators.
Course Description	This course is intended for graduate students in Health Sciences to learn the basic principles of machine learning in biomedical research and to build and strengthen their computational skills of medical research. The course aims to equip students with the fundamental knowledge of machine learning (ML). During the course, the students will acquire basic computational skills and hands-on experience to deploy ML algorithms using python. The students will learn the current practices and applications of ML in medicine, and understand what ML can and cannot do for medicine. The goal of this course to establish an essential foundation for graduate students to take the first steps in computational research in medicine. Introduction to basic principles and current practices of machine learning in biomedical research. Focus on the fundamental ML algorithms with applications in biomedical data; the application of unsupervised learning in genomic data; the application of supervised learning for medical images.
Evaluation Method	- Three assignments (45%) - Term project on machine learning algorithms in medicine (45%) [10% proposal, 35% final presentation] - Math diagnostic test (10%)
Submission Policy	Lateness: Assignments and Projects will be accepted up to 3 days late, but 10% will be deducted for each day late, rounded up to the nearest day. After that, submissions will not be accepted and will receive a score of 0.

Auditing	If you are not registered in the class, it is possible for you to audit it (sit in on the lectures). Here are the official university rules on auditors (taken from the Department of Computer Science instructor's advice page): To audit a course is to sit and listen to the lectures, and perhaps to the tutorials, without formally enrolling. Auditing is acceptable if the auditor is a student at U of T, and no University resources are to be committed to the auditor. The "must be a student" condition means that students of other universities, employees of outside organizations (or even of U of T itself!), or any other non-students, are not permitted to be auditors. (If we did not have this rule, the University would require us to collect auditing fees, and we are not willing to do that.) The "no resources used" condition means that auditors do not get computing accounts, cannot have term work marked, and cannot write exams. In other words, they cannot use instructors time, TA time, or administrative resources of any kind. An auditor may not attend class unless there is an empty seat after the last regularly-enrolled student has sat down. That sounds frivolous, but in fact it is an aspect of an important point: if enrollment in a course has been closed because the room size has been reached, then there may well be physical seats for auditors, because it is rare for every student to appear for a lecture, but auditors will not be allowed to enroll later on in the course, even if some students drop it. Neither instructors nor the department can waive this rule. Often these conditions are perfectly acceptable to auditors; we don't mean to ban the practice, but only to live within the University's rules.
Academic Honesty	You should already be familiar with the University of Toronto's academic honesty policy (the "Code of Behavior on Academic Matters") that deals with issues including plagiarism and cheating. Note that, as should be obvious, plagiarism on problem sets is plagiarism. Using other resources, such as getting your answers from another student or finding them online, rather than working them out yourself, is plagiarism. For a review of the policy please navigate to the following links: http://www.academicintegrity.utoronto.ca/ https://governingcouncil.utoronto.ca/secretariat/policies/code-behaviour-academic-matters-july-1-2019

ChatGPT-GPT4 and the use of generative AI tools for learning	There are several generative AI tools such as ChatGPT, GPT4, Bard, Claude available for public use as general purpose chatbots. These machine learning models are incredibly powerful and are capable of providing responses to arbitrarily complex text (and sometimes image) queries. You may use generative artificial intelligence (AI) tools as learning aids. Generative AI is not required to complete any aspect of this course, and we caution you to not rely entirely on these tools to complete your coursework. Instead, we recommend treating generative AI as a supplementary tool only for exploration or drafting content. Ultimately, you (and not any AI tool) are responsible for your own learning in this course, and for all the work you submit for credit. It is your responsibility to critically evaluate the content generated, and to regularly assess your own learning independent of generative AI tools. Over-reliance on generative AI may give you a false sense of how much you've actually learned, which will lead to poor performance on the midterm test and final exam, in later courses, or in future work or studies after graduation. You will not be permitted to use generative AI on the midterm test or final exam. This assignments and homeworks are designed to be doable without the use of generative AI, using only concepts and skills we have covered in lecture and in the course readings. While some generative AI tools are currently available for free in Canada, please be warned that these tools have not been vetted by the University of Toronto and might not meet University guidelines or requirements for privacy, intellectual property, security, accessibility, and records retention. Generative AI may produce content which is incorrect or misleading, or inconsistent with the expectations of this course. They may even provide citations to sources that don't exist, and submitting work with false citations is an academic offense. These tools may be subject to service interruptions, software modifications, and pricing changes during the semester. The final work you submit for assignments in this course must be your own, and may not include any verbatim content from generative artificial intelligence (AI) tools. You may, however, use generative AI to support your work on this assignment in the following ways: • To answer general questions about high-level concepts covered in this course or assignment • To provide examples on the usage of a library's API • To summarize information • To generate test cases for your code • To assist with understanding and debugging errors If you choose to use any generative AI tools while working on this assignment, you must acknowledge which generative AI tools you used and how you used them. It is an academic offence to not credit sources —including generative AI—in work that you submit. This acknowledgment should take the form of: • A statement at the beginning of the solution pdf highlighting which, tool was used and how. • You should submit a seperate txt file listing all raw ChatGPT transcripts that record your interactions with the tool while working on each assignment. • For code, you should use comments in your code acknowledging which functions, classes, or blocks of code were created in collaboration with generative AI. e.g. #genai - dataloader was created with chatgpt

Schedule		
Date	**Lecture**	**Note**
January 11, 2023	Intro to ML in medicine; nearest neighbor classifier	
January 18, 2023	Linear methods for regression and classification; tree-based classifier	Math diagnostics due
January 25, 2023	Introduction to Python; basic linear algebra; model evaluation methods	
February 1, 2023	ENSEMBLE-based methods; neural networks	Assignment #1 due
February 8, 2023	Supervised learning; Python tutorial for supervised learning practice	
February 15, 2023	Unsupervised learning for clustering: K-means, Gaussian mixture models	Assignment #2 due
February 22, 2023	Reading week, no class	
February 29, 2023	Unsupervised learning for clustering: auto-encoder, graph-based methods; Python tutorial for unsupervised learning practice	Project proposal due
March 7, 2023	Guest lecturer	Assignment #3 due
March 14, 2023	Guest lecturer	
March 21, 2023	Advanced deep learning methods for medical image analysis	
March 28, 2023	Term project in-class presentation	
April 4, 2023	Term project in-class presentation	