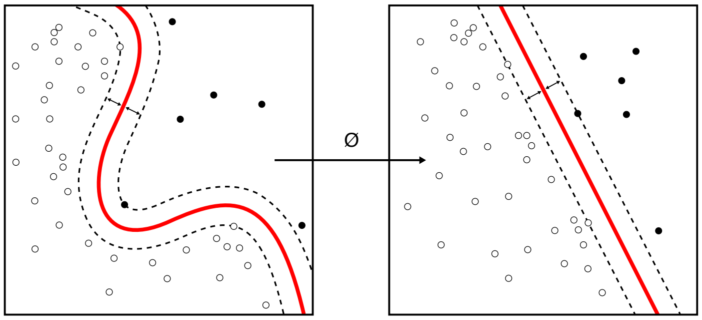
# COMP 432 Machine Learning Fall 2021




This course introduces conceptual and practical aspects of machine learning. Concepts include regression, classification, maximum likelihood estimation, discriminative vs generative modeling, generalization, supervised learning, unsupervised learning, semi-supervised learning. Methods include linear models, mixture models, nearest neighbours, support vector machines, random forests, boosting, ensembles, neural networks, convolutional networks, recurrent networks, and Gaussian processes.

## What you will learn

- The "big ideas" behind many famous learning algorithms, why they work, and why they sometimes don't.
- How to use popular packages like scikit-learn and PyTorch.
- How to train lots models on data, and how to measure success in a meaningful way.

By the end of the course, students should be conversant in machine learning concepts and terminology, capable of implementing basic learning algorithms from scratch, capable of using the scikit-learn and PyTorch libraries, and be effective at incorporating machine learning into their own work.

## Staff

*Instructor:*
Andrew Delong <andrew.delong@concordia.ca> (but please **use Moodle messages** to communicate)
Office: Zoom only until further notice (see course Moodle page)
Office hours: Moonday 8:30–10:15am (you *must* reserve a slot through the **Moodle scheduler**)

*Teaching assistants:*
Nada Abdelkhalek <shimashahfar@gmail.com>
Mahsa Mesgaran <mahsa.mesgaran@gmail.com>
Mohammad Shamshiri <ma.shamshiri@gmail.com>
Naghmeh Shafiee <naghmehshafiee@gmail.com>
Eniela Vela <eniela.vela@mail.concordia.ca>
Nima Sarang <nimasarang@gmail.com>

# Weekly Schedule

*Lecture:*
Monday 5:45–8:15pm with MB-S2.330 reserved, but held over Zoom; see Moodle for **lecture Zoom links**

*Labs sections:*
`DDDI` : Tuesdays @ 11:10pm–1:00pm in H823 (Nada Abdelkhalek)
`DDDJ` : Wednesdays @ 11:10pm–1:00pm in H823 (Mahsa Mesgaran)
`DDDK` : Thursdays @ 11:10pm–1:00pm in H823 (Mahsa Mesgaran)
`DDDL` : Thursdays @ 11:10pm–1:00pm in H827 (Mohammad Shamshiri)

# Tentative Term Schedule

There will be one practice quiz (Q0), five graded quizzes (Q1, Q2, Q3, Q4, Q5), one graded assignment (A1).

A tentative week-to-week schedule is given below, but may be subject to small changes.

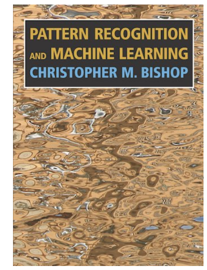
| Week | Lecture topics | Lab | Due |
|---|---|---|---|
| Sep 13 | Intro, Linear Regression | *Lab1* Python, Numpy, Plots | |
| Sep 20 | Maximum Likelihood, Logistic Regression, Clustering | *Lab2* Linear Models | Q0 |
| Sep 27 | K-Means, Gaussian Mixtures | *Lab3* Clustering | Q1 |
| Oct 4 | Kernel Density, Support Vector Machines | *Lab4* Support Vector Machines | |
| Oct 11 | None (Thanksgiving) | None | |
| Oct 18 | Multi-class, Decision Trees, Random Forests | *Lab5* Random Forests | Q2 |
| Oct 25 | Bootstrap, Bagging, Boosting, Ensembles | *Lab6* Boosting | |
| Nov 1 | Loss, Cross Validation, Hyperparameter Search | *Lab7* Hyperparameter Search | Q3 |
| Nov 8 | Neural Networks, Backpropagation | *Lab8* Neural Networks | |
| Nov 15 | Convolutional Networks, Recurrent Networks | *Lab9* Convolutional Networks | Q4 |
| Nov 22 | Dimensionality Reduction, PCA, Autoencoders | *Lab10* Dimensionality Reduction | A1 |
| Nov 29 | Generative Models, Naive Bayes, Gaussian Processes | None | Q5 |
| Dec 6 | Variational Autoencoders, Generative Adversarial Networks | *Lab11* Gaussian Processes | |
| Dec 7 | **Project presentations** | | |

December 7 is the make-up day for Oct 11.

The instructor may make small changes to these dates. Any changes will be announced.

# Textbook

The required textbook for this course is known as "the PRML book" or "the Bishop book":




- *Pattern Recognition and Machine Learning* by Christopher M. Bishop (2006)
- See also the *errata* for the book, as it contains some mistakes, for example in the derivation of the EM algorithm for GMMs.

Other recommended books include:

- *Introduction to Machine Learning with Python* by Andreas Mueller and Sarah Guido (2016)
- *Pattern Classification* by Richard O. Duda, Peter E. Hart, and David G. Stork (2001)
- *Deep Learning* by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (2016)
- *Applied Machine Learning* by David Forsyth (2019)

The Bishop book explains only a subset of topics covered in the course. For example, the *Pattern Classification* book does a much better job of explaining decision trees / CART, and the *Deep Learning* book does a much better job of explaining convolutional and recurrent networks. The *Applied Machine Learning* book provides an alternate presentation of SVMs and decision trees, and may make a good supplement. However, acquiring the "recommended" books is not required for the course. *The instructor will provide lecture slides, labs, and course notes for topics not covered by Bishop.*

The lectures and course notes may occasionally refer to an online reading via hyperlink. If any such reading is *required*, it will be indicated as such.

# Course evaluation

The following assessment is tentative, subject to approval by the department.

10% Labs (1% each, best 10 of 11)
24% Assignment
24% Quizzes (6% each, best 4 of 5)
42% Project

There will be no midterm or final exam for the online version of this course.
See "Projects" and "Policies" sections for more details on assessment.

# Prerequisites

This course requires the following skills:

- Strong Python programming skills (quality of Python code matters).
- Basic linear algebra, basic multivariable calculus, and basic probability.
- Basic understanding of numerical computing and/or optimization.

If you are weak in several of these areas, this course may not be right for you.
Use the first lab and first quiz for self-assessment.

# Projects

Projects must be done in groups of 3–4 students. The 42% project grade breaks down as follows:

- 3% **project proposal** explaining your project goals and plans (max 2 pages).
- 5% **project presentation** to be submitted with the report; the instructor will play these in final lecture.
- 4% **originality** to be assessed by the instructor, to reward groups that propose interesting projects.
- 20% **project report** describing your project data, methods, and conclusions (3 pages).
- 10% **code and data** to be submitted with the report; the instructor and TAs will inspect these files.

Project guidelines will be released later in the course. See "Policies" section for more details.

# Course Policies

**Academic integrity.** Your instructor takes academic integrity very, very seriously. Students who are suspected of violating the Code of Conduct in will be reported. This includes plagiarism of code and/or written text, attempted communication during a quiz or exam, and everything else in the list of offences. All submitted materials must be authored by you alone. At no time should you offer, ask for, or be in possession of, another student's solution to any graded component of this course. Please, if you are struggling, do not cheat yourself out of an opportunity to learn, or cheat your fellow students out of a fair assessment, or risk ruining your academic standing — instead, ask for help from TAs or the instructor right away, because we sincerely want you to learn and want you to succeed!

**Public Github repositories.** GitHub or GitLab repositories containing project, assignment, or other course material must be set to private. Do not share any solutions that you created for the purpose of this course. If you do, and someone else copies from them in the future, you will be included in any cheating incident reports that are subsequently filed, regardless of whether the copying was done with your permission.

**Course content.** Lectures, lab files, quizzes, and assignments will be hosted on Moodle and posted on a weekly basis.

**Zoom usage.** When you join a Zoom session, your Zoom profile must use your real name. The instructor may remove students whose names are not recognized.

**Office hours.** Instructor office hours are by appointment only. Each week students will be able to book up to one 15-minute time slot at a time.

**Communication.** Communicate with the instructor through **MOODLE ONLY**, except in urgent situations. The instructor is teaching many students, and communicating in Moodle is his only hope of staying organized!

- Questions regarding course material should be posted on the forum on Moodle. Students are encouraged to try to answer each others' questions if the instructor or TAs have not been able to answer immediately.
- Personal matters such as "I will miss lab 4 because I have the flu" should be sent to the instructor as a direct message on Moodle, NOT through e-mail unless it is very urgent.
- If the instructor is unresponsive then he is likely overwhelmed with emails and messages and waiting for a block of time to work through them all and reset the "queue"; in that case the best way to communicate may be booking an office hour slot.

**Lectures.** Lectures will be held on Zoom. The Zoom links for lecture can be found in the course Moodle page. The format of each lecture will be to alternate between a 20-30 min block of course material (recorded) followed by a 5-8 min break (not recorded). During each break, students can ask questions in the chat or by raising their hand (using Zoom's "raise hand" feature). Students should be respectful of other students and of the instructor. Recorded portions will be posted on Moodle.

**Quizzes.** Quizzes will be written as timed Moodle quizzes at the start of lecture. There is one practice quiz (Q0) and five graded quizzes (Q1,Q2,Q3,Q4,Q5). The practice quiz Q0, along with the "best 4 of 5 quizzes" policy, is meant to ensure technical glitches or personal emergencies do not ruin your grade. However, beyond that, no exceptions will be made, and it is your responsibility to have a stable internet connection and to successfully complete at least 4 of the 5 graded quizzes. Therefore, you are encouraged to attempt all quizzes just in case you have a personal matter later in the term.

**Assignment.** The assignment will involve programming in Python. Assignments will be collected via a Moodle "assignment". Assignment instructions will be posted partway through the term.

**Labs.** There are 11 labs worth 1% each up to a maximum of 10% of the total grade. Labs are a major learning opportunity in this course, in terms of practical skills and understanding how the algorithms work—please take advantage of labs. Lab sessions are primarily for you to get help from your lab supervisor or from peers. Solutions to early labs will be provided, but not for later labs. *For quizzes, students are responsible for understanding all completed lab material.*

- Lab files (`labX.ipynb`) will be collected as a Moodle assignment each week.
  - Submit labs with the code cells already run, so that the TA does not need to re-run your lab.
  - The TA for your section will give you lab grade of 0.5 if you completed approximately half the lab and 1.0 if you completed the entire lab.
  - Even if your lab appears complete, the TA will assign 0.0 if he/she determines that you have typed nonsense into the code cells, effectively wasting the TA's time. Any suspected plagiarism will be reported.
- You must attend your assigned lab section ONLY. Please respect this rule, it helps balance work among the TAs and to keep them organized.
- Students are encouraged to help each other with labs *verbally*, but copying code is forbidden. You must be the author of all your submitted answers. If you do not complete a lab, you risk struggling on quizzes.

**Projects.** The projects have specific requirements around group assignment, equal contribution, and code reproducibility.

- Students who form their own group must send the names to the instructor via Moodle direct message by the announced deadline. Otherwise they will be randomly assigned a group.
- Project proposals must be written using the $\LaTeX$ template provided on Moodle. Proposals will be collected as a Moodle "assignment" with up to 250 MB of combined code and data.
- Students are expected to contribute equitably. If an individual group member fails to contribute meaningfully, that person's individual project grade may be lowered.
- The "originality" grade is subjectively assessed by the instructor, with the goal of having students put time into designing their own project and not taking one "off the shelf". For example, if you simply attack a standard Kaggle competition, for which many project codes exist on the web, then you will receive an originality score of 0% even if you did a great job at implementing and writing the report.
- Presentations must be a 3 minute recorded video, as if you were submitted a "highlight video" your project to a virtual conference. Consider using Zoom local recording to record audio over slides. The recordings will not be posted online, but they will be presented in the final class.