Introduction to Machine Learning

University of Toronto, Summer 2024 Course Information

Course Meetings

CSC311H1Y

Section	Day & Time	Delivery Mode & Location
LEC5101	Thursday, 6:00 PM - 9:00 PM	In Person: BA 1130

Refer to ACORN for the most up-to-date information about the location of the course meetings.

Tutorials will start in the first week and will be run from 8-9pm, right after the lecture.

Course Contacts

Course Website: https://q.utoronto.ca/courses/345748

Instructor: Amanjit Singh Kainth

Email: csc311-2024-05@cs.toronto.edu

Office Hours and Location: Thursdays 4-5pm in BA 5287

Important Links

- **Piazza**: https://piazza.com/utoronto.ca/summer2024/csc311h1y/home
- **MarkUs**: https://markus.teach.cs.toronto.edu/2024-05

Course Overview

An introduction to methods for automated learning of relationships on the basis of empirical data. Classification and regression using nearest neighbour methods, decision trees, linear models, and neural networks. Clustering algorithms. Problems of overfitting and of assessing accuracy. Basics of reinforcement learning.

Machine learning (ML) is a set of techniques that allow computers to learn from data and experience rather than requiring humans to specify the desired behaviour by hand. ML has become increasingly central both in AI as an academic field and industry. This course provides a broad introduction to some of the most commonly used ML algorithms. It also introduces vital algorithmic principles that will serve as a foundation for more advanced courses, such as CSC412/2506 (Probabilistic Learning and Reasoning) and CSC413/2516 (Neural Networks and Deep Learning).

We start with nearest neighbours, the canonical non-parametric model. We then turn to parametric models: linear regression, logistic regression, soft max regression, and neural networks. We then move on to unsupervised learning, focusing in particular on probabilistic models, but also principal components analysis and K-means. Finally, we cover the basics of reinforcement learning.

This course will delve deep into the mathematical foundations of machine learning and will focus on implementing, understanding and debugging software programs for building machine learning models from data. This class will make heavy use of the mathematical and statistical pre-requisites for this class. Students are expected to be comfortable with technical material from multivariate probability theory, multivariable calculus and linear algebra enough to manipulate them towards the description, design and implementation of learning algorithms for machine learning models.

Course Learning Outcomes

By the end of the course, students will be able to apply supervised and unsupervised learning models to solve machine learning problems. Models covered typically include linear regression, logistic regression, probabilistic models (Naive Bayes), decision trees, neural networks, k-means clustering, expectation-maximization, and principal component analysis. In particular, students will:

- Understand and apply the mathematical techniques used in machine learning models, particularly how to turn a learning problem into an optimization problem and solve that optimization problem (e.g., via gradient descent or other methods)
- Use numerical computing libraries (e.g., NumPy) to build and analyze models; analyze and prepare data for modelling.
- Apply hyperparameter tuning and choose models by evaluating model performance considering the bias-variance tradeoff.
- Evaluate model results on real-world data; communicate the performance and limitations of a model.
- Understand and communicate ethical considerations in deploying a model, including the concerns related to algorithmic fairness.

Prerequisites:

- Programming basics: CSC207H1/ APS105H1/ APS106H1/ ESC180H1/ CSC180H1;
- Multivariate Calculus: MAT235Y1/ MAT237Y1/ MAT257Y1/ (minimum of 77% in MAT135H1 and MAT136H1)/ (minimum of 73% in MAT137Y1)/ (minimum of 67% in MAT157Y1)/ MAT291H1/ MAT294H1/ (minimum of 77% in MAT186H1, MAT187H1)/ (minimum of 73% in MAT194H1, MAT195H1)/ (minimum of 73% in ESC194H1, ESC195H1);
- Linear Algebra: MAT223H1/ MAT240H1/ MAT185H1/ MAT188H1;
- Probability: STA237H1/ STA247H1/ STA255H1/ STA257H1/ STA286H1/ CHE223H1/ CME263H1/ MIE231H1/ MIE236H1/ MSE238H1/ ECE286H1

Corequisites: None

Exclusions: CSC411H1, STA314H1, ECE421H1, CSC311H5, CSC411H5, CSCC11H3. NOTE: Students not enrolled in the Computer Science Major or Specialist program at A&S, UTM, or UTSC, or the Data Science Specialist at A&S, are limited to a maximum of 1.5 credits in 300-/400-level CSC/ECE courses.

Recommended Preparation: MAT235Y1/ MAT237Y1/ MAT257Y1

Credit Value: 0.5

Course Materials

There is no required textbook, but we may assign optional readings from classical textbooks in Machine learning. The following reading books may be of further interest, and are freely available online:

- Bishop, Christopher M. *Pattern recognition and machine learning*. *Springer google schola* 2 (2006): 645-678.
- Hastie, Trevor, et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. New York: springer, 2009.
- MacKay, David JC. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.
- Barber, David. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.
- Sutton, Richard S., and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

- Deisenroth, Marc Peter, A. Aldo Faisal, and Cheng Soon Ong. *Mathematics for machine learning*. Cambridge University Press, 2020.
- Shalev-Shwartz, Shai, and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- Murphy, Kevin P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Course Evaluation

The course will be evaluated as follows:

- Diagnostic Quiz (3%) - for the required math, stats and programming background
- 3 homework assignments (27%, weighted equally)
- Minor assignments (reflection and survey) for the embedded ethics unit (5%). A good faith effort results in full credit.
- Project (10%)
- Midterm Test (20%)
- Final Exam (30%)

Marking Scheme

Assessment	Percent	Details	Due Date
Diagnostic quiz	3%		May 17
Assignments	27%	3 assignments: these will be a mix of mathematical and programming exercises.	May 31, June 14 and July 19 (Tentative)
Ethics Module	5%	Pre-module Survey, Class participation, Post-module Reflection and Survey	June 14 and August 2
Final Project	10%	To be done in groups of 2-3	August 9
Midterm Test	20%		To be held sometime between June 19-24, scheduled by FAS
In-Person Final Exam	35%		Final Exam Period

Late Assessment Submissions Policy

Every student will receive 3 grace days that can be used at any point during the semester on the three homework assignments. After these 3 days have been exhausted, no late submissions will be accepted. Consequently, no credit will be given for assignments submitted after 3 days. The final project will have a separate policy: each group will have 10% of their marks deducted for each late day, up to a maximum of 3 days. After that, no late submissions will be accepted.

Policies & Statements

Late Submission Policy: See above.

Quercus Info

This Course uses the University's learning management system, Quercus, to post information about the course. This includes posting readings and other materials required to complete class activities and course assignments, as well as sharing important announcements and updates. New information and resources will be posted regularly as we move through the term. To access the course website, go to the U of T Quercus log-in page at https://q.utoronto.ca. SPECIAL NOTE ABOUT GRADES POSTED ONLINE: Please also note that any grades posted are for your information only, so you can view and track your progress through the course. No grades are considered official, including any posted in Quercus at any point in the term, until they have been formally approved and posted on ACORN at the end of the course. Please contact me as soon as possible if you think there is an error in any grade posted on Quercus.

The course quercus can found at https://q.utoronto.ca/courses/345748.

Academic Integrity

All suspected cases of academic dishonesty will be investigated following procedures outlined in the Code of Behaviour on Academic Matters (https://governingcouncil.utoronto.ca/secretariat/policies/code-behaviour-academic-matters-july-1-2019). If you have questions or concerns about what constitutes appropriate academic behaviour or appropriate research and citation methods, please reach out to me. Note that you are expected to seek out additional information on academic integrity from me or from other institutional resources. For example, to learn more about how to cite and use source material appropriately and for other writing support, see the U of T writing support website at http://www.writing.utoronto.ca. Consult the Code of Behaviour on Academic Matters for a complete outline of the University's policy and expectations. For more information, please see A&S Student Academic Integrity (https://www.artsci.utoronto.ca/current/academic-advising-and-support/student-academic-integrity) and the University of Toronto Website on Academic Integrity (https://www.academicintegrity.utoronto.ca).

All work you submit must be your own. It is an academic offense to copy the work of someone else *unless you explicitly and clearly attribute the work to its source.* This includes words, sentences, entire documents, and even ideas. Whether you copy or let someone else copy, it is an offense. Academic offenses are taken very seriously and can have correspondingly serious consequences.

At the same time, we want you to benefit from working with other students. For the programming assignments in this course, you cannot submit the same code as another student. However, you can discuss how to solve the problems with anyone you wish. The purpose of the assignments is to allow you to practice implementing an algorithm to solve a real problem. Even if you did not figure out all the implementation yourself, you could potentially still receive full credit for writing up a program *with a list of ALL sources you consulted*: textbooks, web pages, students with whom you discussed the problem, etc. Include all the citations in the Python files that you submit.

You are also welcome to discuss course material and technology related to assignments with each other, and we encourage you to do so. For example, you may work through examples that help you understand course material or new technology or help each other configure your system to run a supporting piece of software.

Please take a few minutes to consult the Academic Integrity at U of T website: it contains good information and concrete strategies to help support your learning in ways that follow the principles of academic integrity, in addition to references to formal policies and procedures.

Collaboration Policy: Collaboration on assignments is not allowed. Each student is responsible for his/her own work. Discussion of assignments should be limited to clarification of the handout itself, and should not involve any sharing of pseudocode or code or simulation results. Violation of this policy is grounds for a semester grade of F, in accordance with university regulations.

Students with Disabilities or Accommodation Requirements

Students with diverse learning styles and needs are welcome in this course. If you have an acute or ongoing disability issue or accommodation need, you should register with Accessibility Services (AS) at the beginning of the academic year by visiting https://studentlife.utoronto.ca/department/accessibility-services/. Without registration, you will not be able to verify your situation with your instructors, and instructors will not be advised about your accommodation needs. AS will assess your situation, develop an accommodation plan with you, and support you in requesting accommodation for your course work. Remember that the process of accommodation is private: AS will not share details of your needs or condition with any instructor, and your instructors will not reveal that you are registered with AS.

Special Consideration Policies if you are registered with Accessibility Services: Your accommodation letter will allow for an extension of up to 7 full days. However, due to the incremental nature of CS courses, granting such a long extension from the onset may cause you to fall behind and be disadvantaged. As such, we will start by suggesting an initial 3-day extension. We will grant the 7-day extension later if necessary.