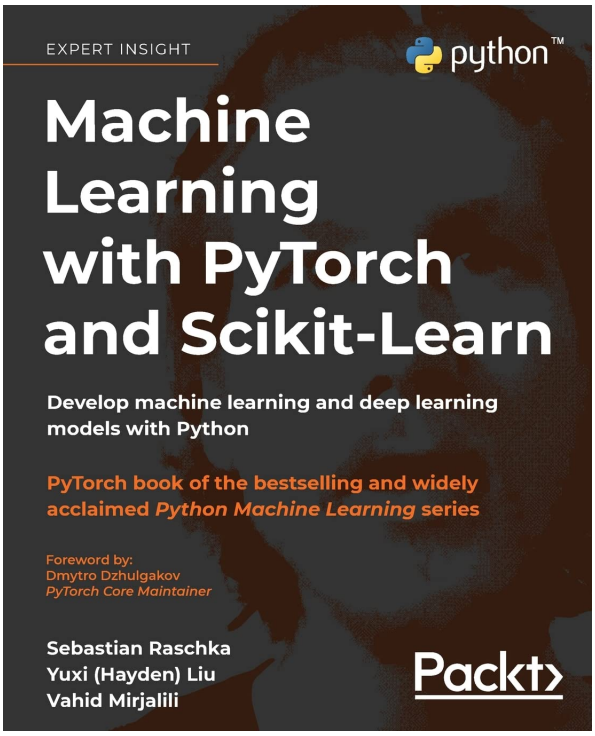
CS 429/529: Introduction to Machine Learning

Spring 2026 Syllabus

Xin Chen

Department of Computer Science
University of New Mexico

Time:	9:30 am - 10:45 am (TR)
Place:	Dane Smith Hall 223
Textbook:	Sebastian Raschka, Yuxi (Hayden) Liu, Vahid Mirjalili. *Machine Learning with PyTorch and Scikit-Learn:* *Develop machine learning and deep learning models with Python.* Packt Publishing. 2022. ISBN-13: 978-1801819312. 
Other references:	Tom Mitchell. *Machine Learning.* McGraw-Hill. 1997. ISBN-13: 978-0071154673. Daniel Liang. *Introduction to Python Programming and Data Structures (3rd Edition).* Pearson. 2022. ISBN-13: 978-1292424125. Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar. *Foundations of Machine Learning (2nd Edition).* The MIT Press. 2018. ISBN-13: 978-0262039406.
Office hours:	My Office (FEC 3190): 2:00 pm - 3:00 pm (TR)

Course Description

Introduction

Machine Learning (ML) techniques are often used to "understand" data and build mathematical models for further interpretations. An example of a learning problem is how to use a finite sample of randomly selected documents, each labeled with a topic, to accurately predict the topic of unseen documents. Therefore, the purpose of most ML problems is to build prediction models based on given data.

ML consists of designing efficient and accurate prediction models and algorithms. In this course, we are going to study the basics of the most popular ML algorithms as well as their models, such as perceptron, support vector machine, decision tree, and artificial neural networks. We do not only implement and use those ML methods but also analyze their performance. This course teaches students the basics of general ML knowledge and helps to prepare for advanced courses.

The course content mainly consists of six parts: (1) *supervised learning*, (2) *traditional ML models*, (3) *ML model evaluation and tuning*, (4) *basic unsupervised learning*, (5) *artificial neural networks*, (6) *basic reinforcement learning*. In addition, the course also teaches basic AI/ML applications in embedded systems.

Prerequisites

This course requires you to already know

- data structures and algorithms (CS 361L);
- basic Python programming including the NumPy and SciPy libraries;
- matrix computation.

Basic Axioms for This Class

Past experience suggests that the study of algorithms can lead to **unnecessary** panic leading to frustration amongst student. There are some basic expectations that every student of this subject should have.

- **No Mysteries.** Ideally, there should be nothing mysterious or ambiguous about any aspect of a data structure or algorithm. Mysteries and ambiguities are bad. They confuse not just the dilettante but also the "seasoned practitioner". Everything is built on simple underlying principles that should be understood, nay grokked. The light of this understanding shall magically transform the firebreathing, human crunching and gargantuan monsters into tiny and unthreatening minnows. To this end, the student should always seek help from the cognoscenti and remember not to panic.
- **Seek and Read Documentation.** You should get into the habit of reading the textbook, looking up the references and online documents. Once again, the instructor

and course staff will point you out to helpful sources, upon request and do so with a friendly smile on their faces.

- **Knowing is Doing.** Like every class, it is one thing to watch your instructor and course staff program, walk through concepts and do proofs. However, until your fingers try out the code, play with it, write new code, reason about it in your own words and write them down, you will not **own** the knowledge. In the process you will make mistakes and may lose your way but once again, the instructor and the course staff will support you in this endeavor.

- **Taking Class Notes.** Note-taking is one of the most effective ways to learn and practice algorithms design and problem solving. You should not only attend every class but also take notes during the classes. Besides, you should spend at least 1 or 2 hours after every class to review and digest the taught content.

Grade

The course grade will be determined by the grades of the following parts:

Assignments: 50 %, Attendance: 10%, Midterm Project: 20%, Final Project: 20%

Attendance will be taken randomly by in-class exercises. We will also have reading assignments which will not be graded but are parts of projects.

Course Policy

- Attending every class. Taking notes. **Please do not use anything (cell phones, laptops, etc.) which distracts you from the class.**

- Students are not allowed to enter the classroom 10 minutes after the start of the class unless the instructor is informed in advance.

- Reading the textbook carefully and arrange regular meetings with your teammates. **Please be respectful to your teammates.**

- Slides, homework assignments and project descriptions will be available on UNM Canvas, and they are not allowed to be shared with anyone outside the class.

- Announcements will be sent via UNM Canvas.

Students with Disabilities: If you anticipate or experience physical or academic barriers based on disability, please let the instructor know immediately so that we can discuss options.

Schedule of the Lectures

I am going to keep the following 22 core lectures and add others related to robotics control.

Lecture	Topic
1	Introduction
2	Supervised Learning
3	Perceptron Learning
4	Adaline, Logistic Regression
5	Support Vector Machine
6	Support Vector Machine
7	Decision and Regression Tree
8	K-Nearest Neighbors
9	Data Preprocessing
10	Dimensionality Reduction
11	Dimensionality Reduction
12	Model Evaluation and Ensemble Learning
13	Clustering Analysis
14	Bayesian Learning
15	Introduction to Artificial Neural Network
16	Feedforward Neural Network
17	Feedforward Neural Network
18	Feedforward Neural Network
19	Convolutional Neural Network
20	Recurrent Neural Network
21	Reinforcement Learning
22	Reinforcement Learning