CSCI-567: Machine Learning

Summer 2025

Prof. V. Adamchik

Course Description:

Machine learning (ML) is a set of algorithms that allow machines to learn (the way humans do) from experience, by extracting useful information and taking the decision based upon the data analysis. The essence of ML is ability to learn from data, identify patterns and build successful predictive models for the unknown datasets. Over the past two decades, machine learning has become increasingly central both in artificial intelligence (AI) as an academic field, and in the technology industry. ML is generally considered separate from AI which is more about building systems to do intelligent things. Machine learning is a specific subset of AI which primarily deals with creation of algorithms which learn with experience and improve themselves over time by feeding on data.

This course provides students with an in-depth introduction to the theory and practical algorithms for machine learning from a variety of perspectives. It covers some of the main models and algorithms for regression, classification, clustering and Markov decision processes. Topics includes linear and logistic regression, regularization, probabilistic (Bayesian) inference, SVMs and kernel methods, ANNs, deep learning, clustering, dimensionality reduction, and generative learning. The course uses the Python programming language and assumes in addition familiarity with linear algebra, probability theory, and multivariate calculus. This course is designed to give graduate-level students a thorough grounding in the methodologies, technologies, mathematics, and algorithms currently needed by people who apply machine learning to a whole host of applications.

Learning Objectives:

- Understanding a wide variety of learning algorithms.
- Develop skills to apply learning algorithms to solving practical problems.
- Understanding how to perform evaluation of learning algorithms and model selection.
- Implement in code common ML algorithms (as assessed by the homeworks).
- Learn how to do a research project, present it and write a report.

Prerequisites:

Students in the class are expected to have a reasonable degree of mathematical sophistication, and to be familiar with the basic knowledge of linear/matrix algebra, multivariate calculus, probability and statistics. Undergraduate classes in these subjects should be sufficient. Students are also expected to have knowledge of basic algorithm design techniques (greedy, dynamic programming, randomized algorithms, linear programming, approximation algorithms) and basic data structures. Programming in Python is required.

Recommended Textbooks:

Bishop = "*Pattern Recognition and Machine Learning*", by C.M. Bishop, Springer, 2006

ESL = "*The Elements of Statistical Learning*", by T. Hastie, R. Tibshirani, and J. Friedman, Springer, 2008

Reading:

The purpose of the readings is to provide a broader and deeper foundation than just the lectures and assessments. The readings for this course are required. We recommend you read them after the lecture. Sometimes the readings include extra topics that are not covered in lecture.

Review Materials:

Linear Algebra: http://viterbi-web.usc.edu/~adamchik/567/review-linalg.pdf

Probability: http://viterbi-web.usc.edu/~adamchik/567/review-prob.pdf

Python Tutorial: http://cs231n.github.io/python-numpy-tutorial/

Google Colab: https://colab.research.google.com/notebooks/intro.ipynb

Theory Written Assignments:

- There will be three written theory assignments.
- The assignments should be submitted electronically via Brightspace.
- Theory assignments must be neatly written or typed, for example in MS Word, and then converted to pdf.
- You may work in groups of 2-3. However, each person should hand-in their own writeup.
- Collaboration should be limited to talking about the problems, so that your writeup is written entirely by you and not copied from your partner.

- There are NO late days for assignments, and we will not accept late submissions.
- We won't regrade assignments.

Programming Assignments:

- There will be three programming (in Python) assignments.
- We advise you to use Google Colab interactive environment.
- Programming assignments should be submitted electronically to Brightspace.
- Collaboration should be limited to talking about the problems.
- Each assignment will be checked for code plagiarism.
- There are NO late days for assignments, and we will not accept late submissions.

Exam:

- There will be a midterm exam. The exam time is limited to the lecture time.
- The practice exams will be posted.
- Before each exam we will schedule a TA review session.
- No makeup exams will be provided.
- If you skip the second exam, you may be eligible for an IN grade for the course. The incomplete grade has to be completed within one year. However, in order to get an IN you have to have a valid cause. Please read the University policy on IN grade for more details.
- The exam solutions and grading rubric will always be posted.
- There will be a regrading session for each exam where you can discuss grading errors. A regrade is allowed only when there are clear and obvious grading errors. Grading errors are simple mistakes made on the part of the graders, and not differences in interpretation of a question or answer.

Final Project:

All students will be required to work in a team (at most three students) on a research project of their choice or from the list of projects proposed by the professor and the TAs. Each project is expected to cover one or more core ML algorithms. The goal of the research projects is to allow the students to go deep into a single topic area and gain experience with research in ML and computer science in general. Some projects may be accompanied by implementation in Python. In those projects you will use UCI or Kaggle classification datasets and test predicting the corresponding outcomes using the machine learning technique. The project consists of 3 parts: a project proposal, a short project presentation, and a project report. The expectations for each part will be discussed in the following sections. Students can use this project to enhance their portfolio or resume for applying to the undergraduate or graduate study.

Project Proposal. The main purpose of the project proposal is to receive feedback from the TA and the instructor regarding whether your project is feasible and whether it is within the scope of this class. The project proposal also offers a chance to receive useful feedback and suggestions on your project. The project proposal template will be provided by the instructor.

Project Presentation. During the last lecture, you will be presenting your project to the class. Each member of the group shall present his or her part of the project. The presentation should cover the following:

- introduce the topic to your class (why is it important, what is the goal of it, what you are planning to achieve)
- summarize the main approach or method (mention models you used, describe your data set, selecting features, data augmentation, tuning hyperparameters, checking for overfitting)
- highlight the outcomes of your project.

Project Report. The written project report (one for all team members) is expected to be 5 pages long and should contain the following sections:

1. Introduction
2. Research Methods (ML Algorithms)
3. Evaluation (Programming Experiments)
4. Conclusions
5. References

Piazza & Emails:

If you have a question about the material or logistics of the class, please do not use e-mail but instead post it on the Piazza at http://piazza.com/usc/summer2025/csci567

You may post it publicly to the whole class or privately to the instructors. Often times, if one student has a question/comment, other also have a similar question/comment. Please DO NOT send emails to the course staff unless your issue is private and/or a private post on Piazza is unsuitable.

Grading:

Theory assignments	**24%**
Programming assignments	**24%**
Midterm exam	**22%**
Final project	**30%**

Letter Grade Distribution:

$\geq$ 90	A	60 – 65	C+
85 – 90	A-	55 – 60	C
80 – 85	B+	50 – 55	C-
73 – 80	B	45 – 50	D+
65 – 73	B-	$\leq$ 45	D

Office Hours:

TBA

Schedule:

This schedule is meant as an outline. Depending on progress, material may be added or removed. Lectures include a problem-solving (discussion) session.

Day	Date	Topics Covered
Wed	May 21	Lecture 1: Course Overview
Thu	May 22	Lecture 2: kNN model, hyperparameters tuning
Mon	May 26	**Memorial Day (no class)**
Tue	May 27	Lecture 3: Decision Tree, Entropy and Gini impurity, Reduced-Error Pruning
Wed	May 28	Lecture 4: Ensemble Learning, Boosting, AdaBoost
Thu	May 29	Lecture 5: Linear Regression, Ridge and Lasso Regularizations
Mon	June 2	Lecture 6: Kernel Methods
Tue	June 3	Lecture 7: Support Vector Machines, Dual SVM
Wed	June 4	Lecture 8: Perceptron, Logistic Regression, Surrogate Losses
Thu	June 5	Lecture 9: Multiclass Classification, Gradient Descent
Mon	June 9	Lecture 10: Deep Learning
Tue	June 10	Lecture 11: Backpropagation
Wed	June 11	Lecture 12: Convolutional Neural Networks, R-CNN
Thu	June 12	Review for exam
Mon	June 16	Exam – I
Tue	June 17	Lecture 13: Generative Learning, Naïve Bayes
Wed	June 18	Lecture 14: Autoencoder, Final Project Discussion
Thu	June 19	**Juneteenth** (no class)
Mon	June 23	Lecture 15: Dimensionality Reduction, Principal Component Analysis
Tue	June 24	Lecture 16: K-means clustering, Kernel Density Estimation
Wed	June 25	Lecture 17: Gaussian Mixture Models, EM algorithm
Thu	June 26	Lecture 18: Markov Models
Mon	June 30	Lecture 19: HMM, Viterbi algorithm, Baum-Welch algorithm
Tue	July 1	Final Project Presentations

Programming Assignments:

Assignment	Content	Out	Due
PA1	Decision Trees	May 26	June 4
PA2	Regressions, SVM	June 4	June 11
PA3	CNN	June 11	June 23

Theory Assignments:

Assignment	Content	Out	Due
HW1	DT, Regression, Boosting (lectures 3 - 6)	May 27	June 3
HW2	SVM, Backpropagation, CNN (lectures 7 - 12)	June 3	June 12
HW3	NB, PCA, Clustering, GMM (lectures 13 - 17)	June 18	June 26

Disclaimer:

Although the instructor does not expect this syllabus to drastically change, he reserves every right to change this syllabus any time in the semester.

Academic Integrity:

The USC Student Conduct Code prohibits plagiarism. All USC students are responsible for reading and following the Student Conduct Code, which appears on https://policy.usc.edu/files/2018/07/SCampus-2018-19.pdf.

In this course we encourage students to study together. This includes discussing general strategies to be used on individual assignments. However, all work submitted for the class is to be done individually. Some examples of what is not allowed by the conduct code: copying all or part of someone else's work (by hand or by looking at others' files, either secretly or if shown), and submitting it as your own; giving another student in the class a copy of your assignment solution; consulting with another student during an exam. If you have questions about what is allowed, please discuss it with the instructor.

For Students with Disabilities:

Any student requesting academic accommodations based on a disability is required to register with Office of Student Accessibility Services (OSAS) each semester. A letter of verification for approved accommodations can be obtained from OSAS. Please be sure the letter is delivered to me (or to TA) as early in the semester as possible. OSAS is located in GFS 120 and is open 8:30 a.m.- 5:00 p.m., Monday through Friday.