Spring 2025
24-787 Machine Learning and Artificial Intelligence for Engineers

Instructor	Levent Burak Kara (lkara@cmu.edu)
Class	M & W 12:00pm – 1:50pm SH 236
Recitation	F 4:00 - 4:50PM SH 234.
Teaching and Course Assistants	See "personnel.pdf"
Office Hours:	Thursday 8:00pm-9:00pm – TA1 – Zoom. See Canvas for zoom link. Friday 2:00pm-3:00pm – TA2 – Zoom. See Canvas for zoom link.
Recommended Text Books	Introduction to Statistical Machine Learning. Masashi Sugiyama, 2016. Pattern Recognition and Machine Learning. Christopher M. Bishop, 2006. Reinforcement Learning: An Introduction. R. S. Sutton, A. Barto, 2020. Deep Learning by Ian Goodfellow, 2016: Available free online (https://www.deeplearningbook.org) There is no single textbook that covers all of the topics we will discuss in this course. Hence, there will be no required textbook and instead you will be provided with copies of lecture notes. The recommended texts referenced above may be useful for a deeper understanding of the course subjects.
Website	https://canvas.cmu.edu This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com. Find our class signup link at this course's Canvas page.

Description:
This course provides an introduction to the fundamental methods and algorithms at the core of modern machine learning. It also covers theoretical foundations as well as essential algorithms and practical techniques for supervised and unsupervised learning.

Topics covered (tentative):
Module 1 - Introduction
Module 2 - Linear Least Squares Regression
Module 3 - Logistic Regression
Module 4 - Support Vector Machines
Module 5 - Decision Trees
Module 6- Feature Engineering
Module 7- Neural Networks I
Module 8 - Neural Networks II
Module 9 - Error Metrics
Module 10 – Model Training
Module 11 - Unsupervised Learning
Module 12 - Dimensionality Reduction
Module 13 - Ensemble Methods
Module 14 - Introduction to Deep Learning

Prerequisites

- Familiarity with basic linear algebra. Solving linear systems of equations. Matrix inversion.
- Familiarity with basic probability theory and statistics. Axioms of probability. Bayes Rule. MLE and MAP estimations. Properties of normal distributions, mean, standard deviation. Covariance.
- Python or basic computer programming skills in another language (Matlab, C/C++) and scientific computing.

Learning Outcomes

By the end of this course, students should be able to:

- Describe the fundamental problems addressed by machine learning and artificial intelligence.
- Design and implement various machine learning algorithms.
- Apply mathematical concepts (including linear algebra, probability, statistics, and optimization) in the context of machine learning algorithms.
- Formulate data-driven approaches to problems and communicate these solutions through implemented algorithms and write-ups.

Recitations

Recitation materials will be posted online, but will not be covered live. Instead, the scheduled recitation hours will be used as office hours where all TAs will be available to take your questions and help with the assignments.

Office Hours

The TAs will be available throughout the week with scheduled, remote office hours. See Canvas for times and Zoom links. In addition, students can use Canvas > Piazza to post questions.

Python Programming

We will use the Python programming language for all assignments in this course. Python is a great general-purpose programming language on its own, but with the help of a few popular libraries (numpy, scipy, matplotlib) it becomes a powerful environment for scientific computing.

Assignments

Most of the assignments in this course involve answering concept questions (multiple choice and/or text responses) and writing computer programs in Python Jupyter notebooks. In a typical assignment, you will implement a machine learning technique from the lecture and use it to solve a sample problem. There will be extra-credit problems with each homework. You will be graded on how well your computer program works. Therefore, you should carefully implement, test and debug each program. Remember, just because there are no error messages, does not mean the program works as it should. In addition to submitting your code, you will typically be required to annotate and comment on your program. To make programming, reporting, commenting, plotting and submitting your program easier, we will use **iPython notebooks (Jupyter Notebooks).**

Getting Help from Teaching Assistants

We encourage you to seek out help from Teaching Assistants. Here is a description of the type of help your TAs can offer you:

- Start your assignments early and seek input/support from TAs as soon as you can. TAs may not always be available to respond to emails right away, so please factor this into your planning.
- Your TA can help you figure out where you code went wrong through asking you strategic questions. While it can be frustrating to not get "the" answer or have TAs debug your code for you, the TAs are asking you questions to help you narrow down the issue and to help you learn the crucial skill of problem solving.
- Your TA will be able to answer specific questions rather than vague questions such as "how do I debug my code." It is important to come to office hours with some prepared questions.
- Your TA may be able to meet with you outside of normal office hours, depending on their availability. Don't hesitate to send an inquiring email.

Submission Procedures:
You will submit a **single PDF file** for each assignment. This will involve combining multiple PDF files on your part. For example, you will be writing/editing multiple Jupyter notebook solutions. Once the code is run and the results are generated (output values, plots, your comments all visible), you will print the Jupyter notebook files as a PDF. You will then combine several PDFs obtained in this way to create a single submission PDF for that week.

Once your single PDF is generated, you will have to assign the correct page numbers to the corresponding questions in your assignment using the submission form found at Canvas > Gradescope.

Jupyter Notebook export methods:
If you are having difficulty exporting your ipynb files, you may try these:

- Jupyter noteook -> export html > open html in browser > print > save as pdf from browser
- Use the print function on your browser cmd+P -> save as pdf
- Another way with more steps is to upload the notebook (.ipynb file) on Google Colab (if you have an andrew id you should have access to Colab) and again use the print function.
- If Colab trims off some cells, you can possibly change the print size of page. By default it might be A4 or Letter size, you can change it to one that works. In certain cases, we have had success with A2 or A3 size.
- One free online tool you can use is Vertopal, which allows you to convert .ipynb files to PDF without any code being cut off: https://www.vertopal.com/en/convert/ipynb-to-pdf

If you use Colab, here is a script you can add to the end of your notebook file to automatically create and download a pdf of your ipynb file. Change the notebook name in the first line as appropriate (also see: colab_print.pdf)

```
#################
NOTEBOOK_NAME = "Untitled0.ipynb"
!apt-get install -y pandoc > /dev/null 2>&1
!sudo apt-get install texlive-xetex texlive-fonts-recommended texlive-plain-generic > /dev/null 2>&1
!pip install nbconvert > /dev/null 2>&1

from google.colab import drive, files
drive.mount('/content/drive')
notebook_path = rf'/content/drive/MyDrive/Colab Notebooks/{NOTEBOOK_NAME}'

import os
if os.path.exists(notebook_path):
   !jupyter nbconvert --to pdf "{notebook_path}" > /dev/null 2>&1
   pdf_filename = notebook_path.replace('.ipynb', '.pdf')
   files.download(pdf_filename)
else:
   print("Notebook not found. Please check the path.")
#################
```

Grading:
Assignments: 100% Total = 100%

90%-100% = A
80%-90% = B
70%-80% = C
60%-70% = D
0%-60% = R

The grading of each assignment will be out of 126 points (which will equal to 100%). The teaching crew may deduct points if submissions are not aligned with the submission instructions.

Late Submission Policy:
You have the option to submit your assignment after the announced deadline (started late, had to travel, computer malfunction etc.). The following policy will apply:

- You have a total of 3 days to be late with each assignment. See "assessment_Schedule.pdf" for details.
- After the original submission deadline:
  - 0 hrs < Submission_time < 3 hrs: Grade = Nominal assignment grade * 0.95.
  - 3 hrs < Submission_time < 12 hrs: Grade = Nominal assignment grade * 0.80.
  - 12 hrs < Submission_time < 24 hrs: Grade = Nominal assignment grade * 0.70.
  - 24 hrs < Submission_time < 48 hrs: Grade = Nominal assignment grade * 0.60.
  - 48 hrs < Submission_time < 72 hrs: Grade = Nominal assignment grade * 0.50.
  - 72 hrs < Submission_time < ---- Grade = Nominal assignment grade * 0.00.
  - As seen, for submissions after the 72 hours passed the deadline; the multiplier is 0.00, hence you will not receive any credit for those submissions. This is to ensure grading and solution posting can follow the strict timelines of the semester.

You are asked to use your grace periods carefully. The teaching crew will not honor additional requests for postponement of a submission.

In any given assignment, if you fail to include the necessary outputs (text, graphs, or whatever is asked for), or the outputs are not clearly visible (blurry, cut off, not properly printed), you will not receive any credit for that particular problem. If you submit a regrade request with the required outputs now visible, you will only receive 60% of the nominal points for that problem. Example: For a problem that is worth 28 points, you originally received a 0. Say you update your solution to now properly display your outputs, submit a regrade request, and receive only 23 out of 28 as there were some mistakes. Your final score for that problem will be 0.6X23 = 13.8 out of 28. Please double check your submissions carefully.

A note on collaboration:
For your assignments, we encourage you to collaborate with each other. To avoid misunderstandings, we explain the parameters of acceptable collaboration below.

Here are some guidelines for you when it comes to collaboration:

1. Any text or graphics contained in your report should be solely your work. You should not use anything that was created by someone else.
2. Any code that you submit should be entirely written by you (excluding, of course, the framework we provide you).
3. To reiterate (1) and (2), you should not be cut-and-pasting from someone else, sending work via email to someone else, or having someone else type the work for you.

Here are some forms of good collaboration:

- Discussing the implications of results with your classmates. "Hey, my neural network has worse results than my decision trees, but that doesn't make intuitive sense to me. What do you think?" "Yeah, my NN is better than my DT. Perhaps your code is doing <insert statement about the code> incorrectly?"
- Asking for help conceptualizing something. "I don't get the purpose of the Kernel as it relates to SVM. Do you know what it means?" "Yes, it means..."
- Discussing theorems, principles, equations, etc. "Hey, in the proof on Problem 2a, I tried using the definition of the dot product; did you do something similar?" "Yes, I did, and I noticed that the fact that cosine is an even function make the two sides of the equation equivalent."
- Debugging code. "My code is giving me this error. Can you come take a look?" "Sure...oh yeah, I got that error as well. You can get rid of it by using this function instead."

If you are unsure of whether you are participating in inappropriate collaboration, talk to the TA. During

grading, we reserve the right to deduct points if cheating is suspected.

Here is a useful link to CMU's policy on the matter:
https://www.cmu.edu/policies/student-and-student-life/academic-integrity.html

A note on Web sites and tools providing solutions to homework problems such as Assignment Geek, Bartleby, Chegg, chatGPT, CoPilot, etc.]:
First, please note that the use of such solution services during exams or assignments is not allowed and is grounds for course failure and additional penalties (see https://www.cmu.edu/policies/student-and student-life/academic-integrity.html). Use of these services is also forbidden when preparing homework solutions; posting homework questions to such services or accessing solutions to these problems will result in zero credit for the entire homework assignment.

Second, there is a broader concern to be aware of. Students have greatly increased access to homework solutions and online services are even available to provide solutions. Great caution should be used in accessing and using these materials in preparing homework. Homework is an opportunity to challenge yourself to apply the course material in a low-stakes assessment. An over reliance on these resources to complete an assignment will likely impede your learning of the material. The most significant challenge students face in these courses is typically the selection of a method for a physical scenario (e.g., what assumptions to use). If a solution is reviewed prior to sufficiently attempting a problem independently, that aspect of problem solving is not practiced. If you choose to use these resources in completing your homework, it is recommended that you review them after completing your attempt. If you have challenges in completing the homework, you are strongly advised to seek help during office hours to assist you in identifying and addressing any possible knowledge gaps or for us to provide additional tips on the solution process.

Cheating:
In the event of a student cheating on any assignment / exam in this course, unless extraordinary circumstances prevail, I plan to impose a penalty of failure for the course and, as required, will report it to the university as a violation of academic integrity.

Accommodations for Students with Disabilities:
If you have a disability and have an accommodations letter from the Disability Resources office, I encourage you to discuss your accommodations and needs with me as early in the semester as possible. I will work with you to ensure that accommodations are provided as appropriate. If you suspect that you may have a disability and would benefit from accommodations but are not yet registered with the Office of Disability Resources, I encourage you to contact them at access@andrew.cmu.edu and/or to visit https://www.cmu.edu/disability-resources/.

Take care of yourself:
This course can be challenging. All of us benefit from support during times of struggle. Please reach out if you need any support with the course content or otherwise. There are also many helpful resources available on campus and an important part of the college experience is learning how to ask for help. Asking for support sooner rather than later is almost always helpful.

If you or anyone you know experiences any academic stress, difficult life events, or feelings like anxiety. or depression, we strongly encourage you to seek support. Counseling and Psychological Services (CaPS) is here to help: call 412-268-2922 and visit their website at http://www.cmu.edu/counseling/ Consider reaching out to a friend, faculty or family member you trust for help getting connected to the support that can help.

If you or someone you know is feeling suicidal or in danger of self-harm, call someone immediately, day or night:

CaPS: 412-268-2922
Re:solve Crisis Network: 888-796-8226
If the situation is life threatening, call the police
On campus: CMU Police: 412-268-2323

Off campus: 911

Thank you, and have a great semester.