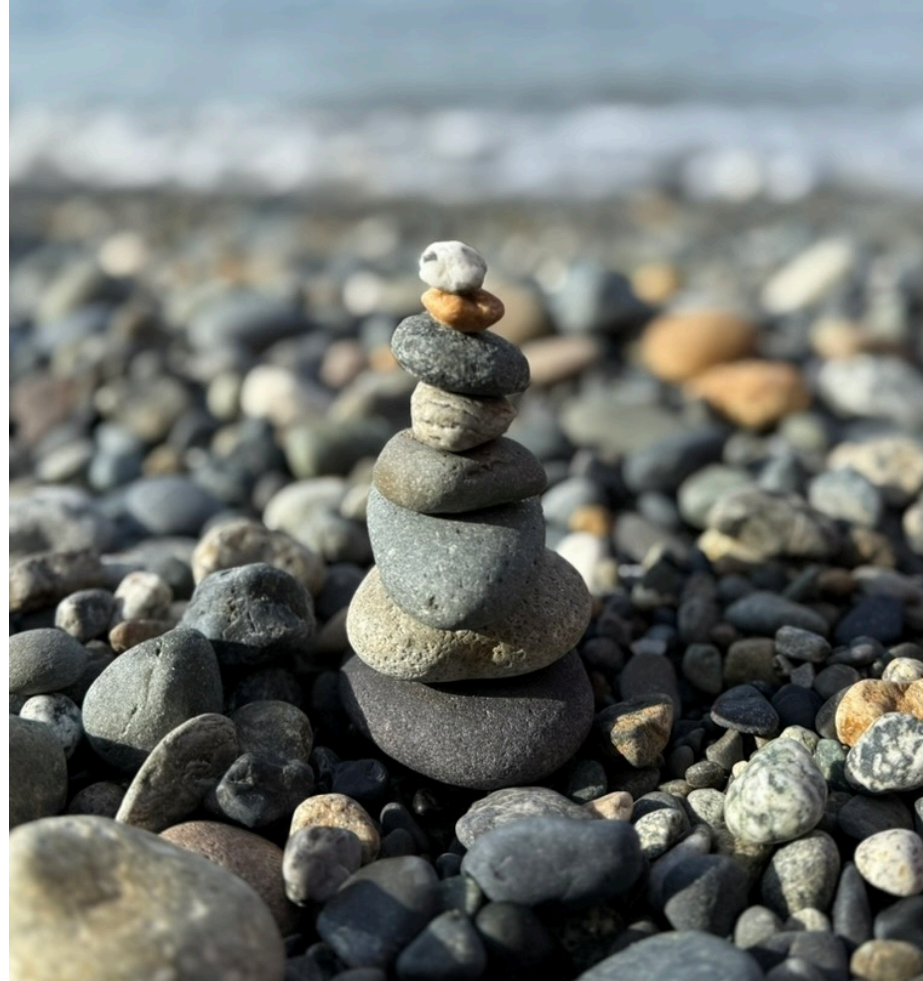


Lecture 1: Introduction to CPSC 330

2026–27 · Winter Term 1

Varada Kolhatkar

Focus on the breath!



Today's big questions

- **AI, ML, DL:** What's the difference?
- **Learning from examples:** How does it work?
- **Rules, ML, or people:** Which should we choose?
- **Getting started in CPSC 330:** What do you need?

CPSC 330 course materials

- [Course website](#): schedules, syllabus, and policies
- [Course book](#): shared course notes
- Today's reading: [Chapter 1: Course Introduction](#)
- [Course GitHub repository](#)

Introductions

Meet your instructor



- Varada Kolhatkar [[vəˈrəda kɔːlhətˈkər](#)]
- You can call me Varada, **V**, or **Ada**.
- Associate Professor of Teaching in the Department of Computer Science.
- Ph.D. in Computational Linguistics at the University of Toronto.
- I primarily teach machine learning courses in the [Master of Data Science \(MDS\) program](#).
- Contact information
 - Email: kvarada@cs.ubc.ca
 - Office: ICCS 237

Meet Eva (a fictitious persona)!



Eva is among one of you. She has some experience in Python programming. She knows machine learning as a buzz word. During her recent internship, she has developed some interest and curiosity in the field. She wants to learn what is it and how to use it. She is a curious person and usually has a lot of questions!

Activity 1

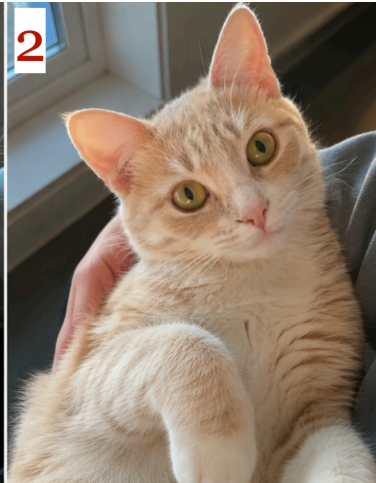
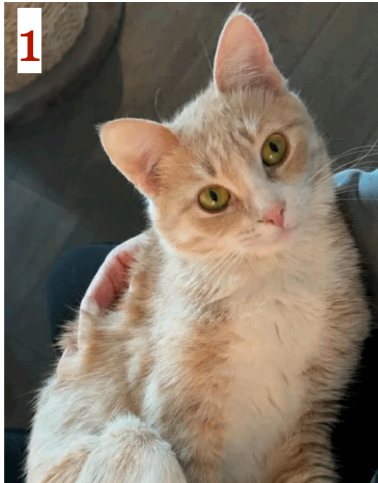
Discuss the following questions with your neighbour.

- What's your name and major?
- Where have you encountered machine learning?
- What would you like to get out of this course?
- What's one thing you're excited or unsure about?

AI in everyday life

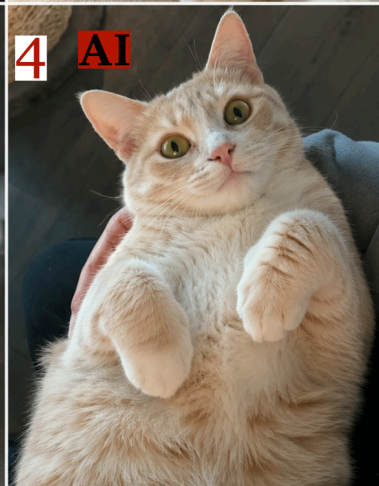
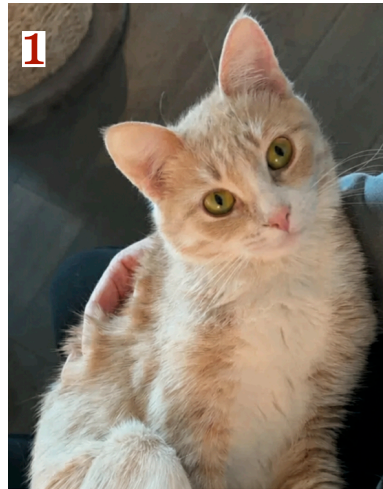
Original vs AI?

Which cats do you think are AI-generated?



Original vs AI?

How did you decide?



AI in the news

Mathematical discovery:

OpenAI reports an AI-generated solution to a major problem about fluid motion.

Emotional support:

About one in five students in an Ontario study reported seeking emotional support from AI. The association with emotional problems does not establish causation.



OpenAI says it cracked 90-year-old maths problem in 88 hours

2 days ago

Kali Hays
Technology reporter

Share  Save  [Add as preferred on Google](#)

Health

As more teens turn to AI for emotional support, experts say it's no substitute for human connection

Roughly one fifth of teens in University of Ottawa study used AI for questions about life, mental health

 [John MacFarlane](#) · CBC News · Posted: Sep 01, 2026 1:00 AM PDT | Last Updated: September 1

What would you want to know before trusting an AI system in either setting?

Which of these do you think use AI?

- A spam detector 
- Autocomplete on your phone 
- A basic calculator 
- Finding the closest holiday destination by straight-line distance 
- Predicting traffic conditions tomorrow
- Recommending a holiday destination you would enjoy 
- A chatbot planning your holiday 

What makes you call something AI?

AI, ML, and DL

What are AI, ML, and DL?

Artificial Intelligence (AI)

Building systems that perceive, reason, predict, or make decisions, often under uncertainty.

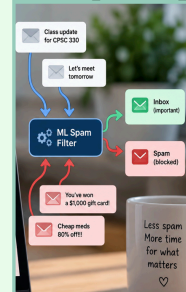
Examples: Deep Blue, rule-based expert systems



Machine Learning (ML)

Learning patterns from data to make predictions or decisions.

Examples: Spam filtering



Deep Learning (DL)

Using deep neural networks to learn patterns from data.

Examples: Face recognition in your phone, ChatGPT



Our focus: learning from examples and applying what we learn to new cases.

How ML works

Image classification

Searching photos for animals involves **image classification**. Imagine telling cats and foxes apart. How might we do this with traditional programming? With ML?



Image ID	Whiskers Present	Ear Size	Face Shape	Fur Color	Eye Shape	Label
1	Yes	Large	Round	Mixed	Round	Cat
2	Yes	Medium	Round	Brown	Almond	Cat
3	Yes	Large	Pointed	Red	Narrow	Fox
4	Yes	Large	Pointed	Red	Narrow	Fox
5	Yes	Small	Round	Mixed	Round	Cat
6	Yes	Large	Pointed	Red	Narrow	Fox
7	Yes	Small	Round	Grey	Round	Cat
8	Yes	Small	Round	Black	Round	Cat
9	Yes	Large	Pointed	Red	Narrow	Fox

Who writes the rules?

Traditional programming

We write the rules.

**Pointed face + red fur +
narrow eyes**



Predict **fox**

Machine learning

We provide labeled
examples.

Features + cat/fox labels



Learn a **prediction rule**

**What would happen with either approach if we encountered
a fox with white fur?**

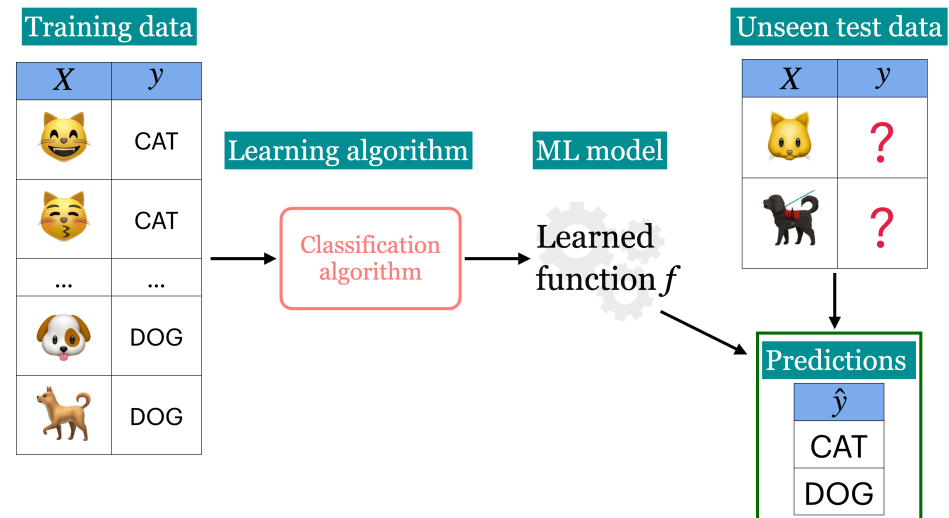
DL approach: example



- A neural network automatically learns which features to look at (edges → textures → objects).
- No need to even specify face shape or fur colour. It learns relevant features on its own.

Supervised learning: inputs, targets, and predictions

- **Inputs (X):** observations described by features, such as fur colour and ear size.
- **Targets (y):** labels or values we want to predict, such as cat or fox.
- **Training:** learn a model f relating inputs to targets.
- **Prediction:** use the model to predict targets for unseen examples.



Applications and coding examples

What to look for in the demos

For each example, identify the inputs, the desired output, and where learning happens.

- **Spam:** learn to predict a category from labeled messages.
- **House prices:** predict a continuous value.
- **Images:** reuse a model trained elsewhere.
- **Clustering and recommendations:** discover groups or patterns in preferences.

Focus on the inputs and outputs. You do not need to understand all the code yet.

Spam classification: labeled examples

- Suppose you are given some data with labeled spam and non-spam messages and you want to predict whether a new message is spam or not spam.

Code

Output

```
1 sms_df = pd.read_csv(DATA_DIR + "spam.csv", encoding="latin-1")
2 sms_df = sms_df.drop(columns = ["Unnamed: 2", "Unnamed: 3", "Unnamed: 4"])
3 sms_df = sms_df.rename(columns={"v1": "target", "v2": "sms"})
4 train_df, test_df = train_test_split(sms_df, test_size=0.10, random_state=42)
```

Spam classification: training

```
1 X_train, y_train = train_df["sms"], train_df["target"]
2 X_test, y_test = test_df["sms"], test_df["target"]
3 clf = make_pipeline(CountVectorizer(max_features=5000), LogisticRegression())
4 clf.fit(X_train, y_train) # Training the model
```

► Pipeline ⓘ ?

► CountVectorizer ?

► 5000 features 📄

► LogisticRegression ?

Spam classification: predictions on unseen messages

These are predictions for a few unseen messages. We need a larger evaluation to judge how reliable the model is.

sms	spam_predictions
3245 Funny fact Nobody teaches volcanoes 2 erupt, tsunamis 2 arise, hurricanes 2 sway aroundn no 1 teaches hw 2 choose a wife Natural disasters just happens	ham
944 I sent my scores to sophas and i had to do secondary application for a few schools. I think if you are thinking of applying, do a research on cost also. Contact joke ogunrinde, her school is one me the less expensive ones	ham
1044 We know someone who you know that fancies you. Call 09058097218 to find out who. POBox 6, LS15HB 150p	spam

Example: Supervised regression

Suppose we want to predict housing prices given a number of attributes associated with houses. The target here is **continuous** and not **discrete**.

target	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view
509000.0	2	1.50	1930	3521	2.0	0	0
675000.0	5	2.75	2570	12906	2.0	0	0
420000.0	3	1.00	1150	5120	1.0	0	0
680000.0	8	2.75	2530	4800	2.0	0	0
357823.0	3	1.50	1240	9196	1.0	0	0

Predicting prices of unseen houses

```

1 pred_df = pd.DataFrame(
2     {"Predicted_target": model.predict(X_test[0:4]).tolist()}
3 )
4 df_concat = pd.concat([pred_df, X_test[0:4].reset_index(drop=True)], axis=1)
5 HTML(df_concat.to_html(index=False))

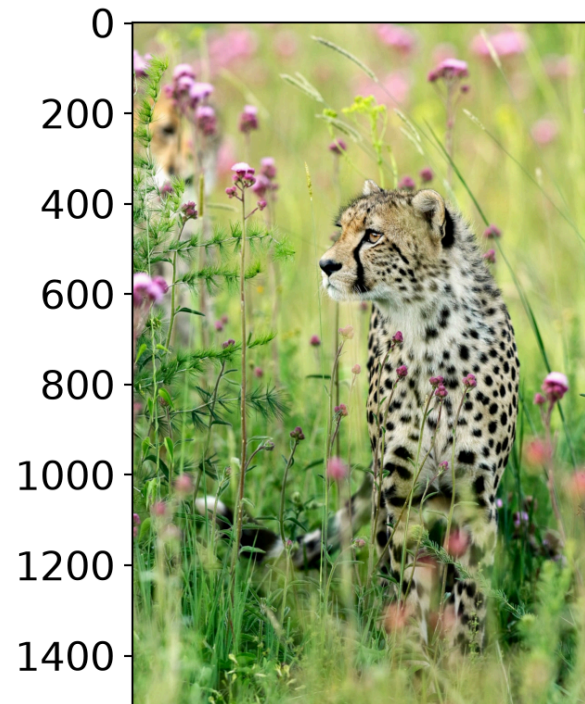
```

Predicted_target	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront
345831.740542	4	2.25	2130	8078	1.0	0
601042.018745	3	2.50	2210	7620	2.0	0
311310.186024	4	1.50	1800	9576	1.0	0
597555.592401	3	2.50	1580	1321	2.0	0

We are predicting **continuous values** here as opposed to discrete values in [spam](#) vs. [ham](#) example.

Example: Predicting image labels

- We reuse an image classifier trained elsewhere to predict labels for new photos.
- The model learned from labeled images during its original training; we do not train it here.



Finding groups in food images

- **Clustering** groups similar examples without supplied target labels.
- The image representation affects what counts as similar.
- People must interpret the groups: do they reflect food types, plate colour, or backgrounds?

Sample Training Images



Clustering images

```
1 import torch
2 densenet = models.densenet121(weights="DenseNet121_Weights.IMAGENET1K_V1")
3 densenet.classifier = torch.nn.Identity() # remove that last "classification" layer
4
5 Z_food = get_features_unsup(densenet, food_inputs)
6 k = 5
7 km = KMeans(n_clusters=k, n_init='auto', random_state=123)
8 km.fit(Z_food);
```

Examining food clusters

199

Image indices: [199 197 223 259 265 169]

Cluster center 0



224

Image indices: [224 93 126 104 7 134]

Cluster center 1

Learning patterns in preferences

							
Eva	?	?	?	5.0	?	5.0	1.0
Jim	?	?	4.0	5.0	?	5.0	?
Pat	5.0	5.0	?	?	4.0	?	?
Sam	5.0	5.0	?	?	5.0	1.0	4.0

- Sam and Pat like many of the same movies. What should we recommend to Pat next?

When should we use ML?

Rules, ML, or human judgment?

Starting point	When it is appropriate
Explicit rules	The logic is known, stable, and can be stated precisely.
Machine learning	Useful patterns exist in representative data, hand-written rules are difficult, and we can evaluate errors.
Human judgment	Context, empathy, values, or accountability are central to the decision.

Real systems can combine all three. Making a prediction does not automatically justify delegating the decision to a model.

? ? Rules, ML, or human judgment?

Choose a starting point for each problem and explain why. Combinations are possible.

- (A) Checking whether an email address ends with `@student.ubc.ca`
- (B) Awarding a scholarship based on a student's personal essay
- (C) Predicting which songs a listener might enjoy
- (D) Checking whether two essays are exactly identical
- (E) Tagging photos of friends

Where could mistakes cause harm, and who should be responsible?

Eva's questions

- How can a model predict a label for an example it has never seen before?
- What happens when it makes a mistake, such as marking a genuine message as spam?
- How do we measure whether it is reliable enough to use?

These questions will guide our work throughout the course.

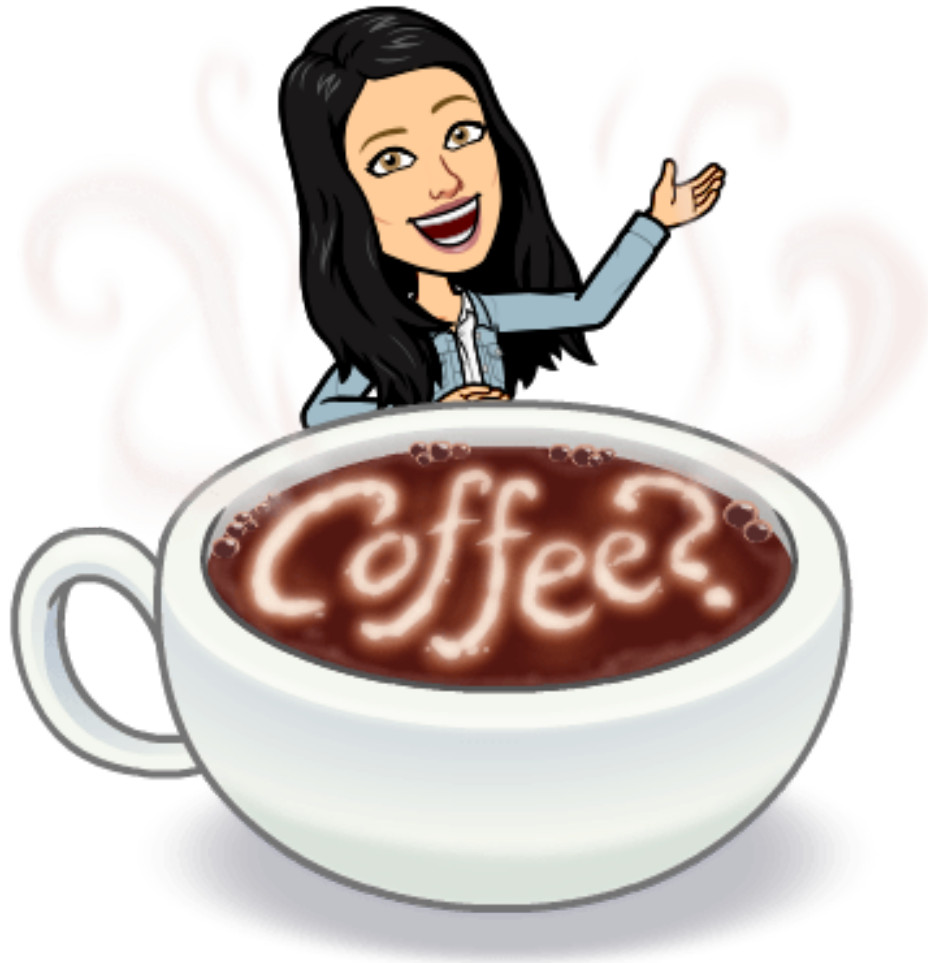
Developing ML solutions

In this course, we will learn to:

- Frame problems and decide what role ML should play.
- Prepare data and engineer and select useful features.
- Choose suitable methods and evaluate their results.
- Study errors, limitations, and possible harms.
- Communicate findings and explore deployment.

A trained model is one component of a solution.

Break



Course overview

Course website

Important

Use the [2026W1 course website](#) for schedules, announcements, and policies. Read the syllabus carefully. The [separate course book](#) contains our shared notes.

Important

Make sure you go through the syllabus thoroughly and complete the syllabus quiz before Sept 19th at 11:59pm.

What do we cover

- Designed for a **diverse group of students** (CS, Statistics, and beyond)
- **Gentle introduction** to machine learning, but also valuable for those with prior experience
- Covers **foundational concepts** in ML and data science:
 - Data preprocessing, supervised learning, clustering
 - Recommendation systems, text processing
 - Intro to neural networks, time series, survival analysis
- Emphasis on **hands-on skills**:
 - Model development, evaluation, interpretation
 - Ethical considerations and clear communication

Course structure

- Part I: Foundations and Supervised Learning
 - Weeks 2, 3, 4, 5, 6, 7
- Unsupervised Learning and Recommendation
 - Weeks 8, 9
- Working with Different Data Types
 - Weeks 9, 11, 12
- Responsible Machine Learning in Practice
 - Weeks 12, 13

CPSC 330 vs. 340

Read [330_vs_340](#) which explains the difference between two courses.

TLDR:

- 340: how do ML models work?
- 330: how do I use ML models and develop ML solutions?
- CPSC 340 has many prerequisites.
- CPSC 340 goes deeper but has a more narrow scope.
- I think CPSC 330 will be more useful if you just plan to apply basic ML.

Lecture format

- In person lectures T/Th.
- Sometimes there will be videos to watch before lecture. You will find the list of pre-watch videos in the schedule on the course webpage.
- We will also try to work on some questions and exercises together during the class.
- Use the course website for the schedule and slides, and the course book for shared notes.

Tutorials

- Weekly tutorials will be run by the TAs.
- Participation is worth **5%** of the course grade and is planned to be assessed during tutorials starting next week. Details will be announced.
- Make use of this helpful resource.

Lecture notes and slides

- Read the shared notes in the [CPSC 330 course book](#).
- For this lecture, read [Chapter 1: Course Introduction](#).
- Each instructor prepares slides and demos based on the book.
- Use the [2026W1 course website](#) for deadlines and policies.

Registration, waitlist and prerequisites

! Important

Please go through [this document](#) carefully before contacting your instructors about these issues. Even then, we are very unlikely to be able to help with registration, waitlist or prerequisite issues.

- See the current syllabus for registration and waitlist deadlines.
- If you are on the waitlist and would like to try your chances, you should already have access to Ed Discussion and PrairieLearn.
- **Please note that it is your responsibility to complete and submit all assessments while you are on the waitlist. No concessions will be made for students who are waitlisted.**
- Instructors cannot change the waitlist order or help you bypass it.

Setting up your computer for the course

Tools used in this course

We will use the following tools throughout the course:

- **Coding:** [Python](#), with either [Jupyter Lab](#) or [VS Code](#)
- **Version Control:** [git](#) and [GitHub](#)
- **Assignment Submission:** [PrairieLearn](#)
- **Discussion Forum:** [Ed Discussion](#)
- **Exams and Final Grades:** [PrairieLearn](#) and [Canvas](#)
- **Recommended Browsers:** [Google Chrome](#) or [Mozilla Firefox](#)

Course Python environment

- Follow the setup instructions [here](#) to set up the course Python environment with [uv](#) on your computer.
- If you do not have your computer with you, you can partner up with someone and set up your own computer later.

Python requirements/resources

We will primarily use Python in this course.

Here is the basic Python knowledge you'll need for the course:

- Basic Python programming
- Numpy
- Pandas
- Basic matplotlib

Homework 1 is all about Python.



Note

We do not have time to teach all the Python we need but you can find some useful Python resources [here](#).

Workload

What does a typical week look like?

- **Before class:** Watch pre-lecture videos or preview notes
- **In class:** Two 80-minute lectures with iClicker questions, activities, and live demos
- **Support:** Weekly tutorials and office hours
- **Practice:** Weekly assignments (except exam weeks)

Tips for success:

- Attend lectures regularly and ask questions
- Start homework early. Hands-on practice is essential
- Use Generative AI tools **responsibly**. No blind copy-pasting
- Always question your data, methods, and results. Justify your choices.

Course policies

Attendance

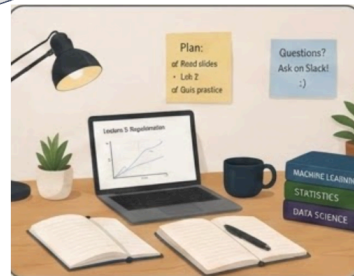
Scenario: I can learn just as well from home

I just learn better on my own. I can read the slides at home and ask questions on Slack if I get stuck. I've been keeping up with the material and doing well on assignments and quizzes. I don't think I need to attend every class.



Maya

Images generated
with GPT-5.6 Sol



What's your reaction?

A

That's fine as long as Maya is keeping up and getting good grades.

B

Attendance and engagement are important part of the learning experience, even if Maya can learn material independently.

C

Maya should receive an academic concession so she can continue learning from home.

Your presence and engagement matters!! Together we create energy that makes lectures valuable!

Grading scheme

- The grading breakdown is [here](#).
- The policy on challenging grades is [here](#).

Exams

- Two midterms, conducted in ORCA by self-reservation over a multi-day period.
- A comprehensive final exam during the exam period.
- Check the [course website](#) for dates and assessment instructions.

Homework assignments

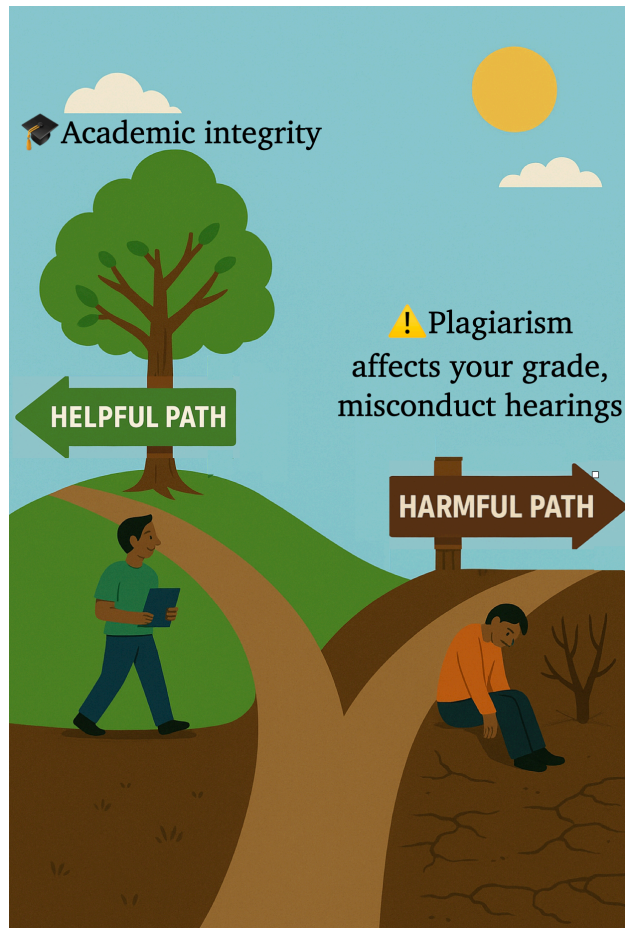
- Our notes are created in a [Jupyter notebook](#), with file extension [.ipynb](#).
- Also, you will complete your homework assignments using Jupyter notebooks.
- Confusingly, “Jupyter notebook” is also the original application that opens [.ipynb](#) files - but has since been replaced by **Jupyter Lab**.
 - I am using Jupyter Lab, some things might not work with the Jupyter notebook application.
 - You can also open these files in Visual Studio Code.

Important note

- Note that your first homework assignment is due **Monday, September 14, 11:59 PM** (tentative; check the [course schedule](#)). This is a relatively straightforward assignment on Python.
- If you struggle with this assignment then that could be a sign that you will struggle later on in the course.
- Write your own answers and code unless the assignment explicitly permits group work.

Plagiarism

Raise your hand if you've ever copied code from StackOverflow.



- Copying isn't always wrong but not acknowledging it is.
- Plagiarism may lead to serious consequences

Using generative AI in this course

Please read our [full Generative AI usage policy](#).

TL;DR: use AI to support, not substitute, your work. If you use a tool:

- Name the tool and briefly explain how you used it.
- Be able to explain and reproduce your work without the tool.
- Do not share instructor-provided course materials or sensitive information without permission.
- GenAI is not permitted during exams or timed assessments unless explicitly allowed.
- Follow group-work rules and be extra careful when collaborating.
- You are responsible for any errors (“hallucinations”) the tool produces.

Who to contact: Grading concerns?

- Start by opening a regrade request.
- If not resolved in two weeks, reach out to your section instructor on Ed Discussion

Who to contact: admin stuff/concessions?

- Read [Frequently Asked Questions](#) before contacting.
- Contact our [course co-ordinator](#)

Who to contact: Questions on the content?

- Make sure to read [our guide on asking for help](#) before reaching out.
- Post your question on Ed Discussion.
- Make use of instructor and TA office hours and tutorials
- I am open to answering questions after class.

Code of conduct

- Our main forum for getting help will be [Ed Discussion](#).

Important

Please read [this entire document about asking for help](#). **TLDR:** Be respectful.

Asking questions during class

- You are encouraged to ask questions by raising your hand.
- No question is a stupid question.
- Recommended reading as you begin your learning journey: [The Fear of Publicly Not Knowing](#)

What I quickly came to realize was that publicly not knowing wasn't an indicator of stupidity, it was an indicator of understanding. And from what I've seen, it is one of the clearest indicators of success in people — more than school prestige, more than GPA.

Checklist for you before the next class

- ☐ Have you read [the syllabus](#) and course policies carefully?
- ☐ Are you able to access course [Canvas](#) shell?
- ☐ Are you able to access [course Ed Discussion](#)?
- ☐ Are you able to access [iClicker Cloud for Section 102](#) for this course?
- ☐ Did you follow the setup instructions [here](#) to set up the course Python environment with [uv](#) on your computer?
- ☐ Have you read [Chapter 1 in the course book](#)?
- ☐ Are you almost finished or at least started with homework 1?

Optional demos and activities

These examples extend the main lecture and can be explored as time permits.

Example: Supervised classification

- We want to predict liver disease from tabular features:

Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotrans
40	14.5	6.4	358	50
33	0.7	0.2	256	21
24	0.7	0.2	188	11
60	0.7	0.2	171	31
18	0.8	0.2	199	34

Model training

```
1 from lightgbm.sklearn import LGBMClassifier
2 # Use one worker to avoid a native multithreading crash on macOS.
3 model = LGBMClassifier(random_state=123, verbosity=-1, n_jobs=1)
4 model.fit(X_train, y_train)
```

▼ LGBMClassifier ⓘ

.Parameters

.Fitted attributes

New examples

- Given features of new patients below we'll use this model to predict whether these patients have the liver disease or not.

Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphot
19	1.4	0.8	178
12	1.0	0.2	719
60	5.7	2.8	214
42	0.5	0.1	162

Model predictions on new examples

- Let's examine predictions

```
1 pred_df = pd.DataFrame({"Predicted_target": model.predict(X_test).tolist()})
2 df_concat = pd.concat([pred_df, X_test.reset_index(drop=True)], axis=1)
3 HTML(df_concat.to_html(index=False))
```

Predicted_target	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	A
No Disease	19	1.4	0.8	178	1
Disease	12	1.0	0.2	719	1
Disease	60	5.7	2.8	214	4
Disease	42	0.5	0.1	162	1

Text classification with a pretrained language model

- We reuse a language model that has already been fine-tuned for positive/negative sentiment classification.
- We do not train it on a new sentiment dataset here.

```
1 from transformers import pipeline, AutoModelForTokenClassification, AutoTok
2 # Sentiment analysis pipeline
3 analyzer = pipeline("sentiment-analysis", model='distilbert-base-uncased-fi
4 analyzer(["I asked my model to predict my future, and it said '404: Life no
5         "'Machine learning is just like cooking—sometimes you follow the
6         and other times you just hope for the best!.'"'])
[{'label': 'NEGATIVE', 'score': 0.995707631111145},
 {'label': 'POSITIVE', 'score': 0.9994770884513855}]
```

Zero-shot learning

- Now suppose you want to identify the emotion expressed in the text rather than just positive or negative.
- **Zero-shot** means no labeled training examples for this particular task; the model was still trained on other data.
- Candidate labels and wording affect predictions. We still need to evaluate them.

```
['im feeling rather rotten so im not very ambitious right now',  
'im updating my blog because i feel shitty',  
'i never make her separate from me because i don t ever want her to feel like i m ashamed with her',  
'i left with my bouquet of red and yellow tulips under my arm feeling slightly more optimistic than when i arrived',  
'i was feeling a little vain when i did this one',  
'i cant walk into a shop anywhere where i do not feel uncomfortable',  
'i felt anger when at the end of a telephone call',  
'i explain why i clung to a relationship with a boy who was in many ways immature and uncommitted despite the excitement',  
'i like to have the same breathless feeling as a reader eager to see what will happen next',  
'i jest i feel grumpy tired and pre menstrual which i probably am but then again its only been a week and im about to have my period']
```

Zero-shot learning for emotion detection

```
1 from transformers import AutoTokenizer
2 from transformers import pipeline
3 import torch
4
5 #Load the pretrained model
6 model_name = "facebook/bart-large-mnli"
7 classifier = pipeline('zero-shot-classification', model=model_name)
8 exs = dataset["test"]["text"][10:20]
9 candidate_labels = ["sadness", "joy", "love", "anger", "fear", "surprise"]
10 outputs = classifier(exs, candidate_labels)
```

Zero-shot learning for emotion detection

	sequence	labels	scores
0	i don t feel particularly agitated	[surprise, anger, joy, sadness, fear, love]	[0.36008691787719727, 0.3019044101238251, 0.11901266872882843, 0.11381471157073975, 0.060391418635845184, 0.04478989169001579]
1	i feel beautifully emotional knowing that these women of whom i	[joy, love, surprise, fear,	[0.3699439465999603, 0.2887146472930908, 0.2560795247554779,

Optional: How the food clusters were computed

The main demo groups neural-network representations with K-Means.

```
1 import torch
2
3 densenet = models.densenet121(weights="DenseNet121_Weights.IMAGENET1K_V1")
4 densenet.classifier = torch.nn.Identity()
5 Z_food = get_features_unsup(densenet, food_inputs)
6 km = KMeans(n_clusters=5, n_init="auto", random_state=123)
7 km.fit(Z_food)
```

The representation determines which similarities the clustering method can find.

Activity 2

Think of a problem you have come across in the past which could be solved using machine learning.

- What would be the input and output?
- How do humans solve this now? Are there heuristics or rules?
- What kind of data do you have or could you collect?
- What mistakes could the system make, and who would be affected?
- Should ML support a person or make the final decision?

Ways of learning from data

- **Supervised learning:** predict targets from labeled examples
- **Unsupervised learning:** discover structure without supplied targets
- **Reinforcement learning:** learn actions through interaction and rewards
- **Generative AI:** generate content; models can use several learning approaches
- **Recommendation systems:** learn preferences, often combining multiple approaches

These categories are not mutually exclusive.