Lecture 1: Introduction to CPSC 330

Firas Moosvi (Slides adapted from Varada Kolhatkar)

🤝 Introductions ! 🤝

About your instructor

Firas Moosvi

Lecturer

University of British Columbia

Biography

I am a Lecturer in the Computer Science department at the University of British Columbia. Though I mainly teach computer science now, I am a multidisciplinary educator with a PhD in Physics and is also interested in data science and education in general. I strongly believe in computational literacy for all and aims to make STEM courses accessible through Active Learning techniques and open education resources. My two main research umbrellas are the scholarship of teaching and learning (SoTL), and Learning Analytics. I am looking at how learning analytics data can provide insight to surface and ultimately reduce inequities in STEM programs. I am also heavily invested in promoting and implementing alternative grading systems in large classes, at scale. I am always happy to collaborate on teaching and learning projects, drop me a note here!

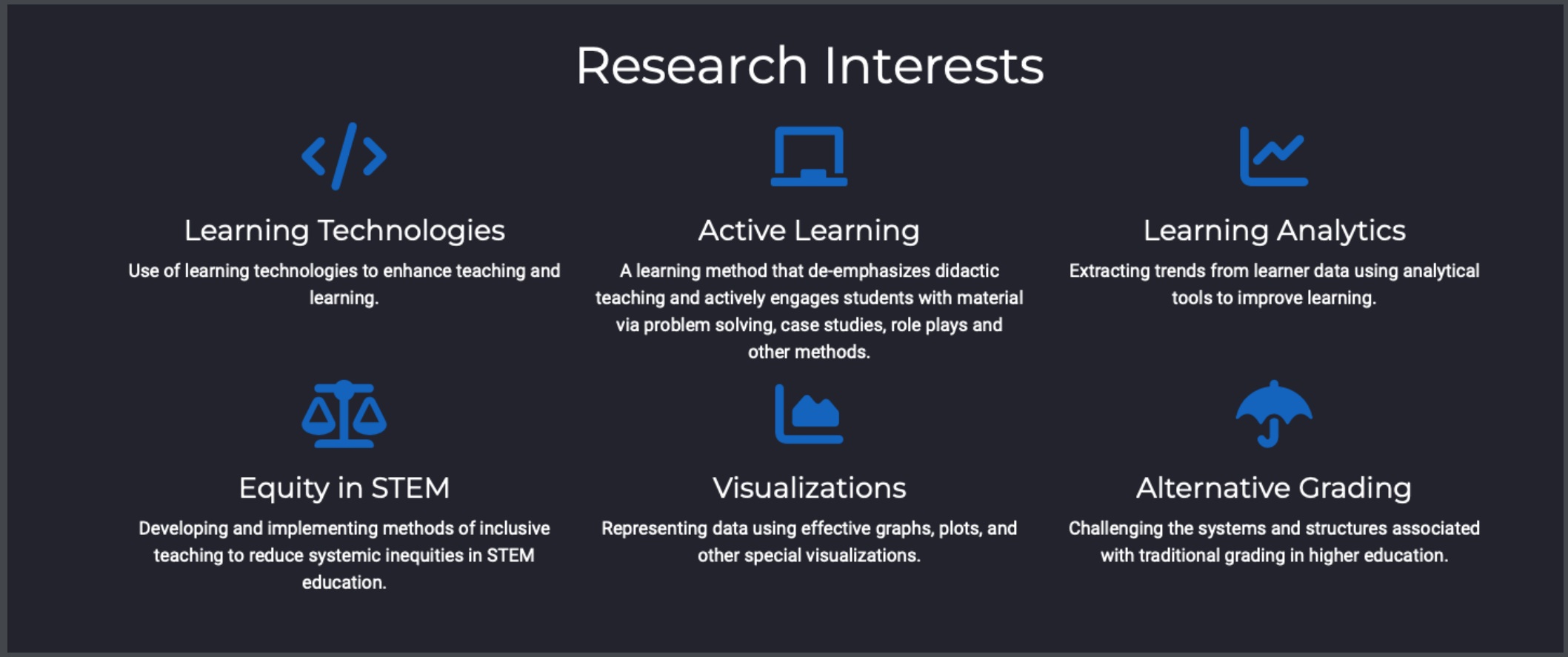
Interests

- Scholarship of teaching and learning
- Authentic assessments
- Alternative grading paradigms
- Learning analytics
- Data visualization and science communication

Education

- PhD in Medical Physics, 2019
  University of British Columbia
- MSc in Medical Biophysics, 2012
  University of Toronto
- BSc in Biophysics, 2009
  University of British Columbia

About my research interests



Group work in this class

This term we will try to work in “Pods” of 3-5 …

Research shows that there is tremendous benefits in students working (and struggling) together!

Students ask better and more insightful questions, engage more deeply with the work, and it adds a social element to class.

We will try this in CPSC 330 this term!

Group work in this class

Understandably, not everyone is a fan of group work - I understand that!

So you will never be forced to work in groups If you would like to opt-out, move to the far left and far right sides of the room so we know you prefer to work individually.

If everyone moves to the side of the room, we will re-evaluate this approach 😂

There are no marks or points associated with these groups, and everyone should work on their own laptops as well

Group work: Pods

Form a Pod of 3-5 people sitting close to you.

Each person should answer the following questions:

- Preferred Name,
- Year,
- (intended) Major
- Why are you taking CPSC 330?

Then, as a group, answer the following question:

What is the most interesting (good or bad) example of Machine Learning in society?

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!

🎯 Learning Outcomes

By the end of this lesson, you will be able to:

- Explain the difference between AI, ML, and DL
- Describe what machine learning is and when it is appropriate to use ML-based solutions.
- Briefly describe supervised learning.
- Differentiate between traditional programming and machine learning.
- Evaluate whether a machine learning solution is suitable for your problem or whether a rule-based or human-expert solution is more appropriate.
- Navigate the course materials and get familiar with the course syllabus and policies.

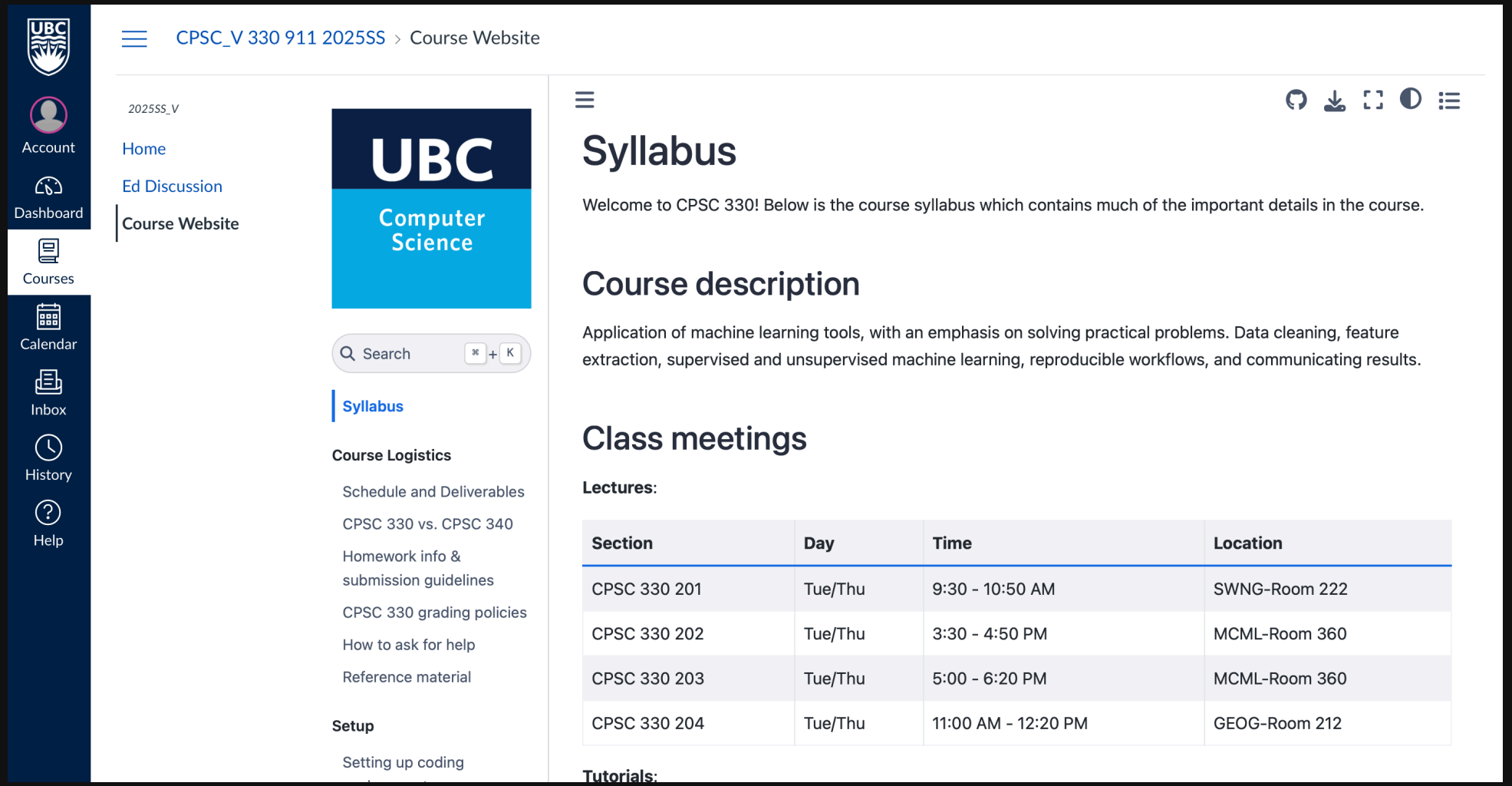
About this course

CPSC 330 website

- Course Jupyter book: https://ubc-cs.github.io/cpsc330-2026S1
- Course GitHub repository: https://github.com/UBC-CS/cpsc330-2026S1

Important

Course website: https://ubc-cs.github.io/cpsc330-2026S1 is the most important link. You can access the course website from Canvas.



Please read everything on there!

You can find the source code for everything we do here: https://ubc-cs.github.io/cpsc330-2026S1.

Important

Make sure you go through the syllabus thoroughly and complete the syllabus quiz before next class!

Asking questions during class

You are welcome to ask questions by raising your hand!

If you would prefer to write notes and ask questions later, you are more than welcome to do that also! Use Ed Discussion.

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.

- We are expecting that all students registered on the waitlist have already, or will soon get a notification to join the course!
- The waitlist will close on Friday at 3 PM and no more students will be able to register after that!
- It is your responsibility to catch up on any missed work.

Lecture format

- In person lectures Monday, Wednesday, and Friday from 10:00 AM to 12:20 PM
- **WARNING**: This is Summer, things will be fast!!
- There will be videos to watch before almost every 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.
- All materials will be posted in this GitHub repository.
  - You may attend any tutorials or office hours your want, regardless of in which/whether you're registered.

Home work assignments

- First homework assignment is due soon, it will be released to you today.
- 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.
- You must do the first two homework assignments on your own.

Exams

- We'll have three self-scheduled midterms over a few days and one final exam.
- All exams will be held in Computer-based Testing Facility (CBTF).

Course structure

- Part I: Introduction, ML fundamentals, preprocessing, midterm 1
- Part II: Unsupervised learning, transfer learning, common special cases, midterm 1
- Part III: Communication and ethics
  - ML skills are not beneficial if you can't use them responsibly and communicate your results. In this module we'll talk about these aspects.
  - **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 nice.

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`
- **Recommended Browsers:** Microsoft Edge, or Mozilla Firefoxor Safari. Google Chrome is *not* recommended for reasons...

Course conda environment

- Follow the setup instructions here to create a course `conda` environment 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:** Interactive Lectures with questions, activities, and live demos
- **Support:** Weekly tutorials and office hours
- **Practice:** Weekly assignments (x2)

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

Homework format: Jupyter lab notebooks

- 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.

Jupyter lab notebooks

- Notebooks contain a mix of code, code output, markdown-formatted text (including LaTeX equations), and more.
- When you open a Jupyter notebook in one of these apps, the document is "live", meaning you can run the code.

For example:

```
1  1 + 1
```

```
2
```

```
1  x = [1, 2, 3]
2  x[0] = 9999
3  x
```

```
[9999, 2, 3]
```

More about Jupyter lab

- By default, Jupyter prints out the result of the last line of code, so you don't need as many `print` statements.
- In addition to the "live" notebooks, Jupyter notebooks can be statically rendered in the web browser, e.g. this.
  - This can be convenient for quick read-only access, without needing to launch the Jupyter notebook/lab application.
  - But you need to launch the app properly to interact with the notebooks.

Lecture notes

- All the lectures from last year are available here.
- We cannot promise anything will stay the same from last year to this year, so read them in advance at your own risk.
- A “finalized” version will be pushed to GitHub and the Jupyter book right before each class.
- Each instructor will have slightly adapted versions of notes to present slides during lectures.
- You will find the link to these slides in our repository: https://github.com/UBC-CS/cpsc330-2026S1/tree/main/lectures/103-Firas-lectures

Grades

- The grading breakdown is here.
- The policy on challenging grades is here.

Setting up your computer for the course

Recommended browser and tools

- You can install Firefox here.
- You can install Edge here.

In this course, we will primarily be using `Python` , `git`, `GitHub`, `Canvas`, `Gradescope`, `Ed DIscussion`, and `PrairieLearn`.

Course conda environment

- Follow the setup instructions here to create a course `conda` environment 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
- Sparse matrices

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.

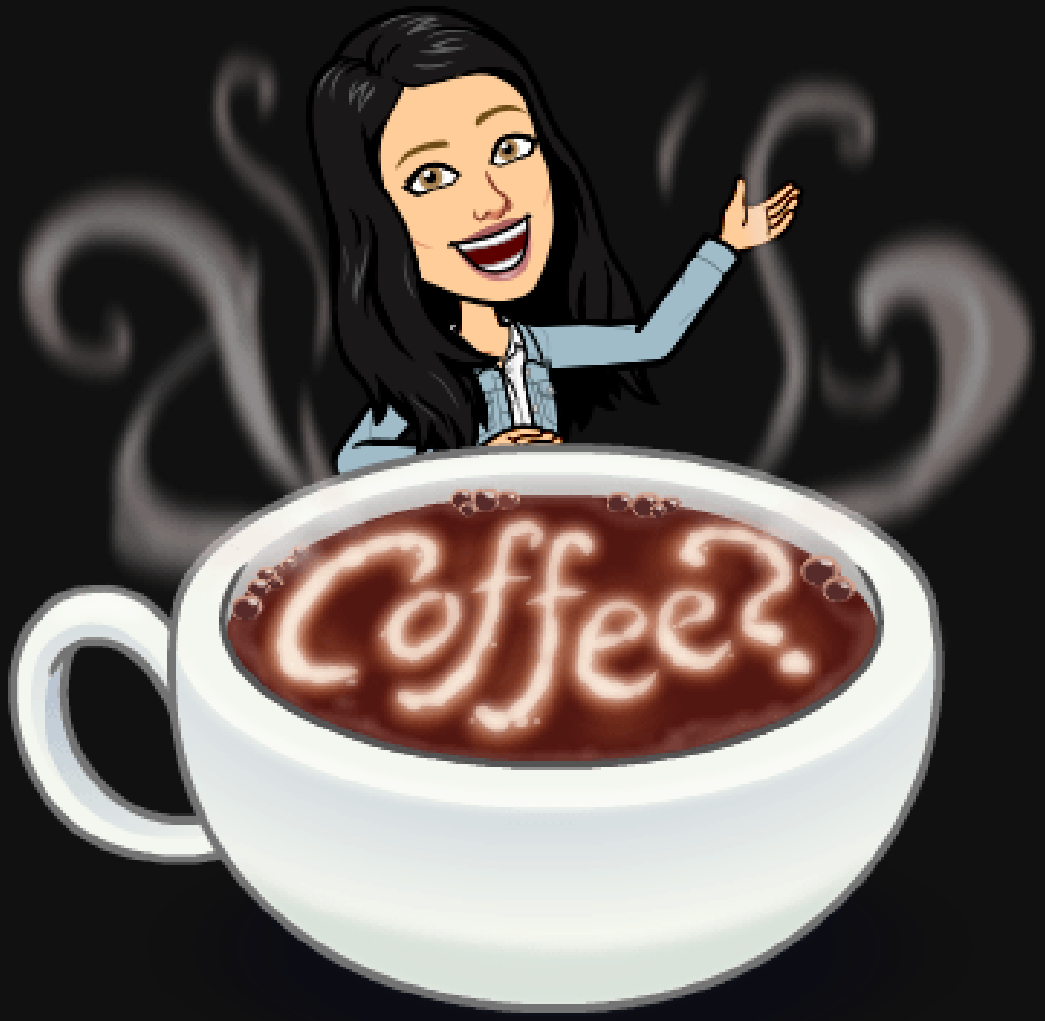
CPSC 330 vs. 340

Read https://ubc-cs.github.io/cpsc330-2026S1/docs/330_vs_340.html which explains the difference between two courses.

TLDR:

- 340: how do ML models work?
- 330: how do I use ML models?
- 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.

Break



Activity 1

Discuss with you neighbour

- What do you know about machine learning?
- What would you like to get out this course?
- Are there any particular topics or aspects of this course that you are especially excited or anxious about? Why?

What is machine learning?

Which cat do you think is AI-generated?

- A
- B
- Both
- None

Source

- What clues did you use to decide?

What are AI, ML, DL?

- **Artificial Intelligence (AI)**: Making computers act smart
  - Examples: Deep Blue, early spell checkers
- **Machine Learning (ML)**: Learning patterns from data
  - Example: Spam filtering in Gmail
- **Deep Learning (DL)**: Using neural networks to learn complex patterns
  - Examples: Face recognition in your phone, voice assistants

ARTIFICIAL INTELLIGENCE (AI)

Techniques that enable machines to perform tasks that typically require human intelligence.

MACHINE LEARNING (ML)

Machines learn patterns from data without being explicitly programmed with rules.

DEEP LEARNING (DL)

Uses neural networks to automatically extract patterns and features from raw data

Let’s walk through an example

- Have you used search in Google Photos? You can search for “cat” and it will retrieve photos from your libraries containing cats.
- This can be done using **image classification**.

Image classification

- Imagine we want a system that can tell cats and foxes apart.
- How might we do this with traditional programming? With ML?

Image ID	Whiskers Present	Ear Size	Face Shape	Fur Color	E S
1	Yes	Large	Round	Mixed	R
2	Yes	Medium	Round	Brown	A
3	Yes	Large	Pointed	Red	N
4	Yes	Large	Pointed	Red	N
5	Yes	Small	Round	Mixed	R
6	Yes	Large	Pointed	Red	N
7	Yes	Small	Round	Grey	R
8	Yes	Small	Round	Black	R
9	Yes	Large	Pointed	Red	N

Traditional programming: example

- You hard-code rules. If all of the following satisfy, it's a fox.
  - pointed face ✅
  - red fur ✅
  - narrow eyes ✅
- This works for normal cases, but what if there are exceptions

ML approach: example

- We don’t tell the model the exact rule. Instead, we give it many labeled images, and it learns probabilistic patterns across multiple features, not rigid rules.
  - If fur is red ⟶ 90% chance of Fox.

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.

What is ML?

- ML uses algorithms to learn patterns from data and build models.
- These models can:
  - Make predictions on new data
  - Support complex decisions
  - Generate new content
- ML systems can improve when trained on more data.
- There is no one-size-fits-all model. The right choice depends on the problem.

When to use ML?

- When the problem can’t be solved with a fixed set of rules
- When you have **lots of data** and **complex relationships**
- When human decision-making is too slow or inconsistent

Approach	Best for
Traditional Programming	Rules are known, data is clean/predictable
Machine Learning	Rules are complex/unknown, data is noisy

When to use Machine Learning (ML) solutions?

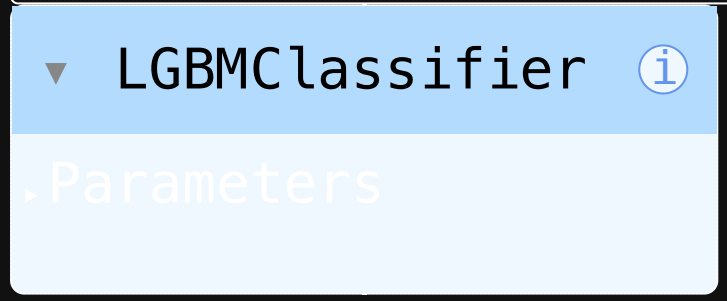
Example: Supervised classification

- We want to predict liver disease from tabular features:

Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_A
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 model = LGBMClassifier(random_state=123, verbose=-1)
3 model.fit(X_train, y_train)
```



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_Phosphotase	Alamine_A
19	1.4	0.8	178	13
12	1.0	0.2	719	157
60	5.7	2.8	214	412
42	0.5	0.1	162	155

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_Phospl
No Disease	19	1.4	0.8	178
Disease	12	1.0	0.2	719
Disease	60	5.7	2.8	214
Disease	42	0.5	0.1	162

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	waterfro
509000.0	2	1.50	1930	3521	2.0	0
675000.0	5	2.75	2570	12906	2.0	0
420000.0	3	1.00	1150	5120	1.0	0
680000.0	8	2.75	2530	4800	2.0	0
357823.0	3	1.50	1240	9196	1.0	0

Building a regression model

```
from lightgbm.sklearn import LGBMRegressor

X_train, y_train = train_df.drop(columns= ["target"]), train_df["target"]
X_test, y_test = test_df.drop(columns= ["target"]), train_df["target"]

model = LGBMRegressor()
model.fit(X_train, y_train);
```

Predicting prices of unseen houses

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

Predicted_target	bedrooms	bathrooms	sqft_living	sqft_lot	floors	w
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 apposed to discrete values in `disease` vs. `no disease` example.

Text data

Example: Text classification

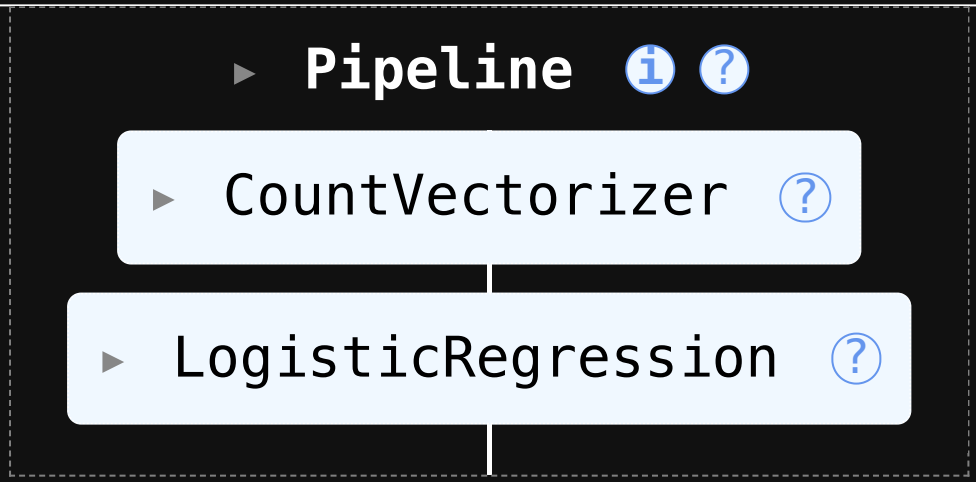
- Suppose you are given some data with labeled spam and non-spam messages and you want to predict whether a new message is spam or not spam.

Code | Output

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

Let's train a model

```
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(max_iter=5000))
4 clf.fit(X_train, y_train) # Training the model
```



Unseen messages

- Now use the trained model to predict targets of unseen messages:

	sms
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
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 m...
1044	We know someone who you know that fancies you. Call 09058097218 to find out who. POBox 6, LS15HB 150p
2484	Only if you promise your getting out as SOON as you can. And you'll text me in the morning to let me know you made it in ok.

Predicting on unseen data

The model is accurately predicting labels for the unseen text messages above!

	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

Example1: Text classification with LLMs

- LLMs = Large Language Models

```
from transformers import pipeline, AutoModelForTokenClassification, AutoTokenizer
# Sentiment analysis pipeline
analyzer = pipeline("sentiment-analysis", model='distilbert-base-uncased-finetuned-sst-2-english')
analyzer(["I asked my model to predict my future, and it said '404: Life not found.'",
          '''Machine learning is just like cooking—sometimes you follow the recipe,
            and other times you just hope for the best!.'''])
```

```
config.json: 100%  629/629 [00:00<00:00, 91.3kB/s]
model.safetensors: 100%  268M/268M [00:01<00:00, 288MB/s]
Loading weights: 100%  104/104 [00:00<00:00, 3210.22it/s]
tokenizer_config.json: 100%  48.0/48.0 [00:00<00:00, 8.17kB/s]
vocab.txt: 232k/? [00:00<00:00, 4.95MB/s]
```

```
[{'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.

README.md: 9.05k/? [00:00<00:00, 1.09MB/s]

split/train-00000-of-00001.parquet: 100% 1.03M/1.03M [00:00<00:00, 1.87MB/s]

split/validation-00000-of-00001.parquet: 100% 127k/127k [00:00<00:00, 335kB/s]

split/test-00000-of-00001.parquet: 100% 129k/129k [00:00<00:00, 898kB/s]

Generating train split: 100% 16000/16000 [00:00<00:00, 756028.43 examples/s]

Generating validation split: 100% 2000/2000 [00:00<00:00, 214339.58 examples/s]

Zero-shot learning for emotion detection

```python
from transformers import AutoTokenizer
from transformers import pipeline
import torch

#Load the pretrained model
model_name = "facebook/bart-large-mnli"
classifier = pipeline('zero-shot-classification', model=model_name)
exs = dataset["test"]["text"][10:20]
candidate_labels = ["sadness", "joy", "love","anger", "fear", "surprise"]
outputs = classifier(exs, candidate_labels)
```

```
config.json:          1.15k/?  [00:00<00:00,  155kB/s]

model.safetensors:  100%                                  1.63G/1.63G  [00:05<00:00,  701MB/s]

Loading weights:  100%                                    515/515  [00:00<00:00,  3387.07it/s]

tokenizer_config.json:  100%                              26.0/26.0  [00:00<00:00,  4.50kB/s]

vocab.json:           899k/?  [00:00<00:00,  65.2MB/s]
```

merges.txt: 456k/? [00:00<00:00, 41.6MB/s]

tokenizer.json: 1.36M/? [00:00<00:00, 87.5MB/s]

Zero-shot learning for emotion detection

	sequence	labels	scores
0	i don t feel particularly agitated	[surprise, anger, joy, sadness, fear, love]	[0.36008885502815247, 0.30189985036849976, 0.11901344358921051, 0.11381492018699646, 0.06039227545261383, 0.04479063674807548 5]
	i feel beautifully emotional knowing that these women of	[joy, love, surprise, fear,	[0.36994174122810364, 0.28871557116508484, 0.2560800909996032 7,

Image data

Example: Predicting labels of a given image

- Suppose you have a bunch of animal images. You do not have any labels associated with them and you want to predict labels of these images.
- We can use machine learning to predict labels of these images using a technique called **transfer learning**.

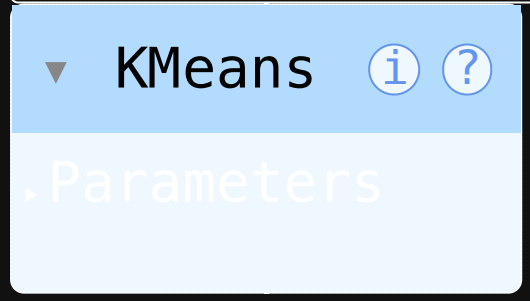
Clustering images

Finding groups in food images

K-Means on food dataset

```
densenet = models.densenet121(weights="DenseNet121_Weights.IMAGENET1K_V1")
densenet.classifier = torch.nn.Identity()  # remove that last "classification" layer
```

```
Z_food = get_features_unsup(densenet, food_inputs)
k = 5
km = KMeans(n_clusters=k, n_init='auto', random_state=123)
km.fit(Z_food)
```



Examining food clusters

```
1 for cluster in range(k):
2     get_cluster_images(km, Z_food, X_food, cluster, n_img=6)
```

```
39
Image indices:  [ 39 197  12  14 138 181]
```

Cluster center 0

```
228
Image indices:  [228  65 128  54 175 260]
```

Cluster center 1

Interactive: Is ML appropriate?

? ? Questions for you

Select all that apply: Which problems are suitable for ML?

- a. Checking if a UBC email address ends with @student.ubc.ca before allowing login
- b. Deciding which students should be awarded a scholarship based on their personal essays
- c. Predicting which songs you'll like based on your Spotify listening history
- d. Detecting plagiarism by checking if two essays are exactly identical
- e. Automatically tagging photos of your friends on Instagram

Summary: When is ML suitable?

Approach	Best Used When…
Machine Learning	The dataset is **large and complex**, and the decision rules are **unknown, fuzzy, or too complex to define explicitly**
Rule-based System	The logic is **clear and deterministic**, and the rules or thresholds are **known and stable**
Human Expert	The problem involves **ethics, creativity, emotion, or ambiguity** that can't be formalized easily

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?

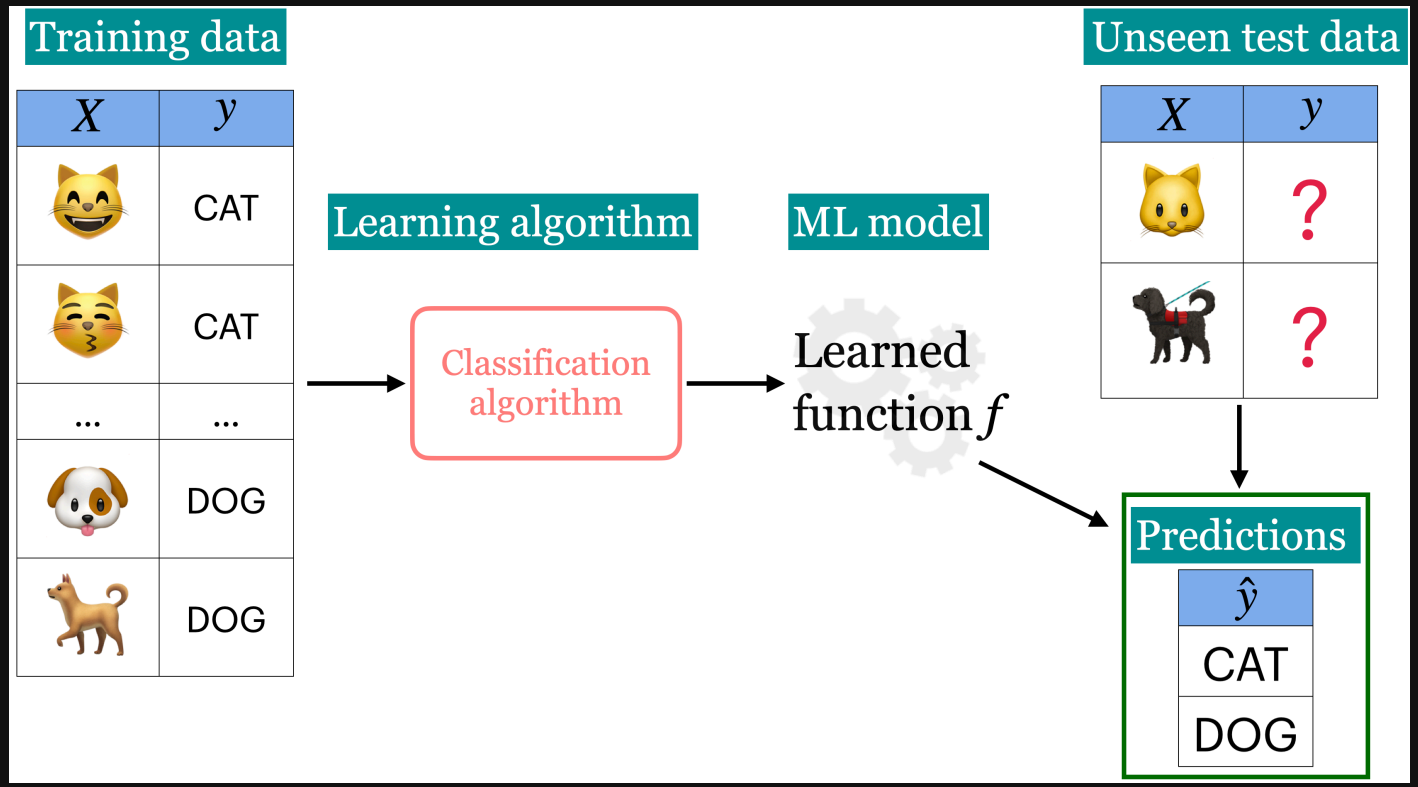
Types of machine learning

Here are some typical learning problems.

- **Supervised learning** (Gmail spam filtering)
- **Unsupervised learning** (Google News)
- Reinforcement learning (AlphaGo)
- Generative AI (ChatGPT)
- **Recommendation systems** (Amazon item recommendation system)

What is supervised learning?

- Training data comprises a set of observations (X) and their corresponding targets (y).
- We wish to find a model function f that relates X to y.
- We use the model function to predict targets of new examples.



🤔 Eva's questions

At this point, Eva is wondering about many questions.

- How are we exactly "learning" whether a message is spam and ham?
- Are we expected to get correct predictions for all possible messages? How does it predict the label for a message it has not seen before?
- What if the model mis-labels an unseen example? For instance, what if the model incorrectly predicts a non-spam as a spam? What would be the consequences?
- How do we measure the success or failure of spam identification?
- If you want to use this model in the wild, how do you know how reliable it is?
- Would it be useful to know how confident the model is about the predictions rather than just a yes or a no?

It's great to think about these questions right now. But Eva has to be patient. By the end of this course you'll know answers to many of these questions!

What is Machine Learning (ML)?

Spam prediction

- Suppose you are given some data with labelled spam and non-spam messages

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)
```

Traditional programming vs. ML

- Imagine writing a Python program for spam identification, i.e., whether a text message or an email is spam or non-spam.
- Traditional programming
  - Come up with rules using human understanding of spam messages.
  - Time consuming and hard to come up with robust set of rules.
- Machine learning
  - Collect large amount of data of spam and non-spam emails and let the machine learning algorithm figure out rules.

Let's train a model

- There are several packages that help us perform machine learning.

```
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(max_iter=5000))
4 clf.fit(X_train, y_train); # Training the model
```

Unseen messages

- Now use the trained model to predict targets of unseen messages:

	sms
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
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 m...
1044	We know someone who you know that fancies you. Call 09058097218 to find out who. POBox 6, LS15HB 150p
2484	Only if you promise your getting out as SOON as you can. And you'll text me in the morning to let me know you made it in ok.

Predicting on unseen data

The model is accurately predicting labels for the unseen text messages above!

	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

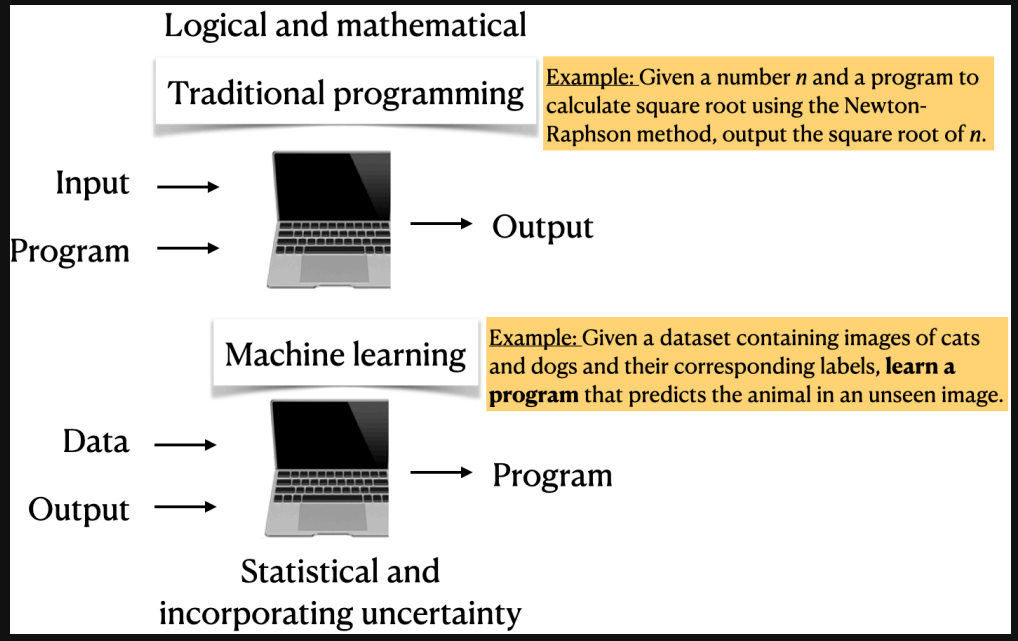
A different way to solve problems

Machine learning uses computer programs to model data. It can be used to extract hidden patterns, make predictions in new situation, or generate novel content.

> A field of study that gives computers the ability to learn without being explicitly programmed.
> – Arthur Samuel (1959)

ML vs. traditional programming

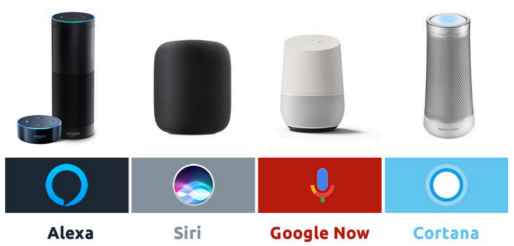
- With machine learning, you're likely to
  - Save time
  - Customize and scale products



Prevalence of ML

Let's look at some examples.

Voice assistants

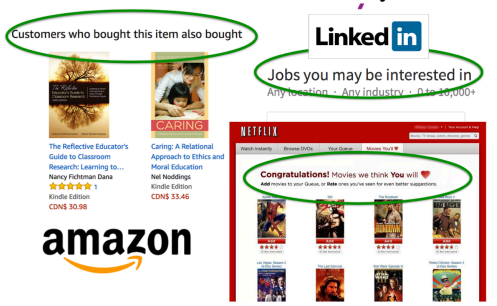


Google news

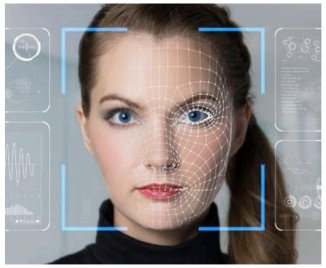
Armed man who broke into Trudeau residence charged with threatening to kill or injure PM
The Guardian · 1 hour ago

- Corey Hurren, alleged Rideau Hall intruder, threatened Trudeau: RCMP officer
  Global News · 4 hours ago
- Corey Hurren had multiple firearms, uttered threat against Trudeau, court documents allege
  CBC.ca · 2 hours ago
- Man arrested near Rideau Hall had several weapons, threatened PM Trudeau: RCMP
  CTV News · 22 minutes ago

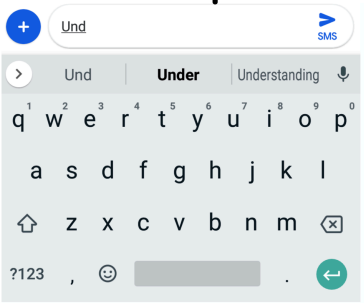
Recommendation systems



Face recognition

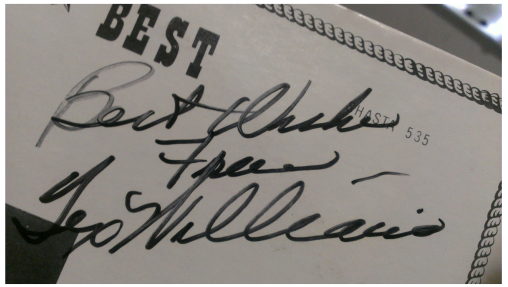


Auto-completion



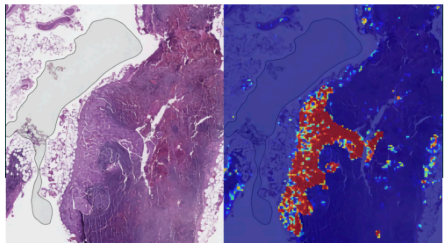
Stock market prediction

Character recognition



Self-driving car

Cancer diagnosis



Drug discovery

AlphaGo

Activity: For what type of problems ML is appropriate? (~5 mins)

Discuss with your neighbour for which of the following problems you would use machine learning

- Finding a list of prime numbers up to a limit
- Given an image, automatically identifying and labeling objects in the image
- Finding the distance between two nodes in a graph

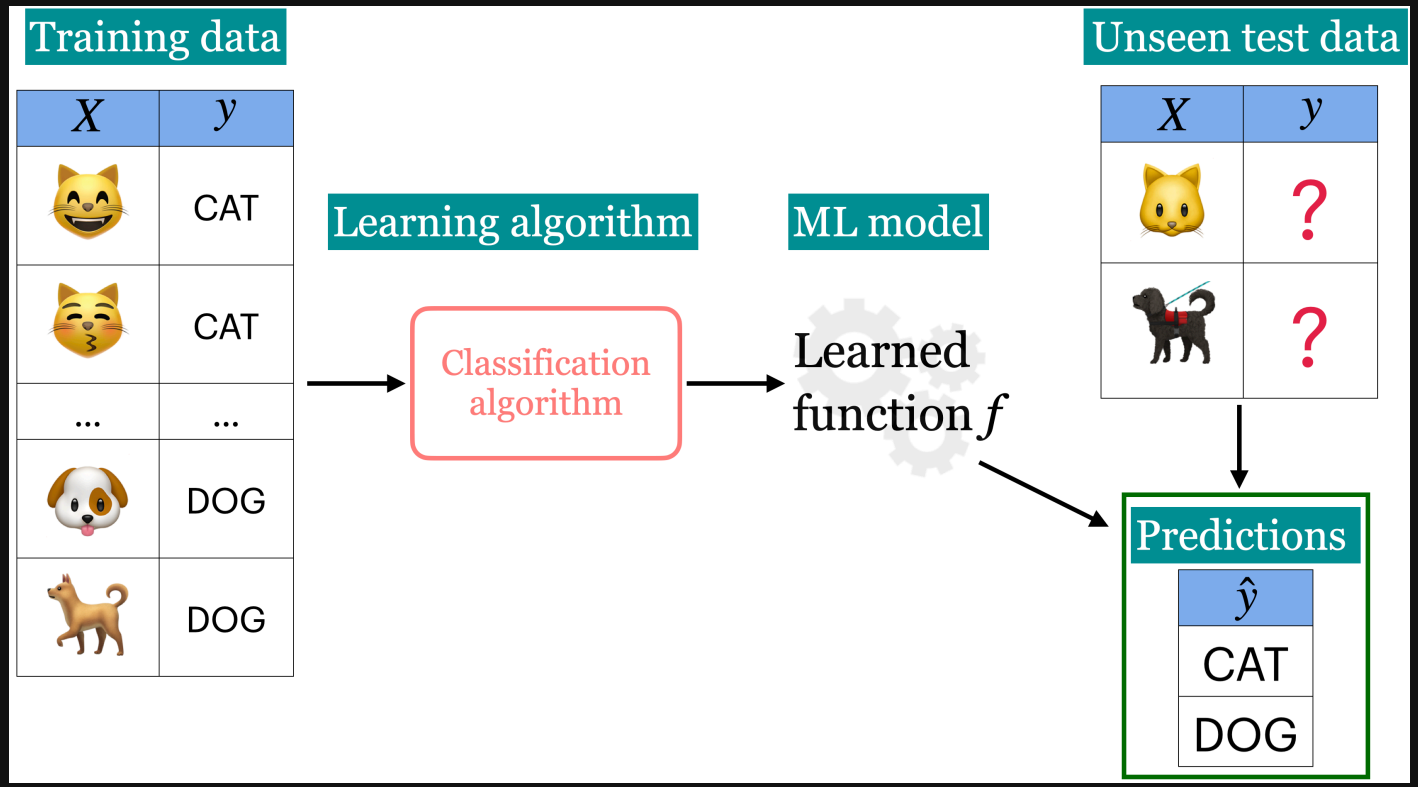
Types of machine learning

Here are some typical learning problems.

- **Supervised learning** (Gmail spam filtering)
  - Training a model from input data and its corresponding targets to predict targets for new examples.
- Unsupervised learning (Google News)
  - Training a model to find patterns in a dataset, typically an unlabeled dataset.
- Reinforcement learning (AlphaGo)
  - A family of algorithms for finding suitable actions to take in a given situation in order to maximize a reward.
- Recommendation systems (Amazon item recommendation system)
  - Predict the “rating” or “preference” a user would give to an item.

What is supervised learning?

- Training data comprises a set of observations (X) and their corresponding targets (y).
- We wish to find a model function f that relates X to y.
- We use the model function to predict targets of new examples.



🤔 Eva's questions

At this point, Eva is wondering about many questions.

- How are we exactly "learning" whether a message is spam and ham?
- Are we expected to get correct predictions for all possible messages? How does it predict the label for a message it has not seen before?
- What if the model mis-labels an unseen example? For instance, what if the model incorrectly predicts a non-spam as a spam? What would be the consequences?
- How do we measure the success or failure of spam identification?
- If you want to use this model in the wild, how do you know how reliable it is?
- Would it be useful to know how confident the model is about the predictions rather than just a yes or a no?

It's great to think about these questions right now. But Eva has to be patient. By the end of this course you'll know answers to many of these questions!

Looking ahead to next class

It is *very* important that you watch the assigned pre-lecture videos before class!