

Course Introduction

Contents

- Imports
 - Learning outcomes
 - Characters in this course?
 - Why machine learning (ML)? [video]
 - Supervised machine learning
 - (Optional) Surveys
 - About this course
 - Summary

DSCI 571 Supervised Learning

UBC Master of Data Science program, 2024-25

Imports

```
import os
import sys

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

sys.path.append(os.path.join(os.path.abspath("."), "code"))
from IPython.display import HTML, display
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline

plt.rcParams["font.size"] = 16
pd.set_option("display.max_colwidth", 200)

%matplotlib inline

DATA_DIR = DATA_DIR = os.path.join(os.path.abspath("."), "data/")
```

Learning outcomes

From this lecture, you will be able to

- Explain the motivation behind study machine learning.
- Briefly describe supervised learning.
- Differentiate between traditional programming and machine learning.
- Assess whether a given problem is suitable for a machine learning solution.

Characters in this course?

- DSCI 571 teaching staff (instructors, TAs, and our co-ordinators)
- Eva (a fictitious enthusiastic student)
- And you all, of course 😊 !

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!

You all

- Since we're going to spend the next 10 months with each other, I would like to know you a bit better.
- Please fill out [Getting to know you survey](#) when you get a chance.

Why machine learning (ML)? [[video](#)]

See also

Check out [the accompanying video](#) on this material.

Prevalence of ML

Let's look at some examples.

<p>Voice assistants</p> <p>Alexa Siri Google Now Cortana</p>	<p>Google news</p> <p>Armed man who broke into Trudeau residence charged with threatening to kill or injure PM The Guardian - 1 hour ago</p> <ul style="list-style-type: none"> 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 	<p>Recommendation systems</p> <p>Customers who bought this item also bought LinkedIn</p> <p>Jobs you may be interested in Amazon</p> <p>Congratulations! Movie was Watched Netflix</p>	
<p>Face recognition</p>	<p>Auto-completion</p>	<p>Stock market prediction</p>	<p>Character recognition</p>
<p>Self-driving car</p>	<p>Cancer diagnosis</p>	<p>Drug discovery</p>	<p>AlphaGo</p>

- Image sources
 - [Voice assistants](#)
 - [Google News](#)
 - [Recommendation systems](#)
 - [Face Recognition source](#)
 - [Auto-completion](#)
 - [Stock market prediction](#)
 - [Character recognition](#)
 - [AlphaGo](#)
 - [Self-driving cars](#)
 - [Drug discovery](#)
 - [Cancer detection](#)

Saving time and scaling products

- Imagine writing a program for spam identification, i.e., whether 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.
- With machine learning, you're likely to
 - Save time
 - Customize and scale products

Supervised machine learning

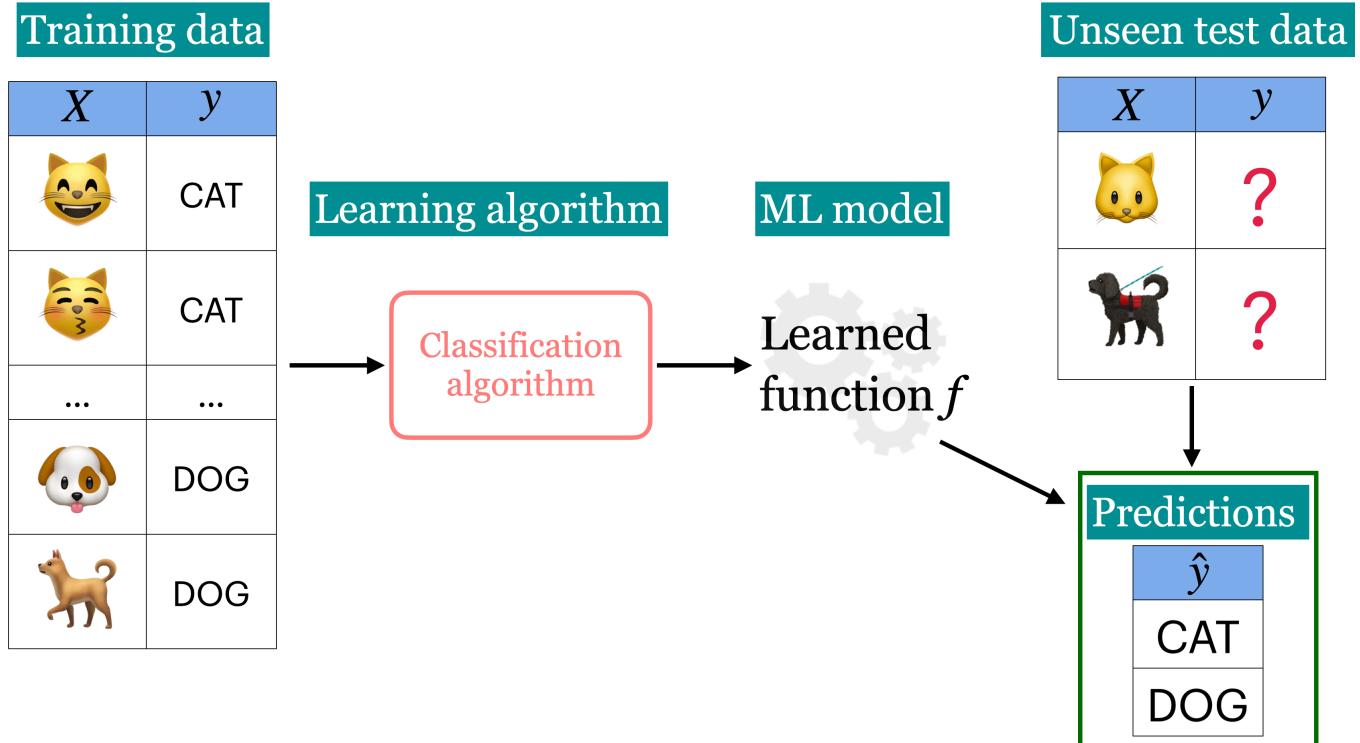
Types of machine learning

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.
- Generative AI
 - Create new content, such as text, images, and music, based on patterns and data it has learned from existing materials.

What is supervised machine learning (ML)?

- 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.



Example: Predict whether a message is spam or not

Input features X and target y

Note

Do not worry about the code and syntax for now.

Note

Download SMS Spam Collection Dataset from [here](#).

Training a supervised machine learning model with X and y

▶ Show code cell source

target	sms
spam	LookAtMe!: Thanks for your purchase of a video clip from LookAtMe!, you've been charged 35p. Think you can do better? Why not send a video in a MMSto 32323.
ham	Aight, I'll hit you up when I get some cash
ham	Don no da:)whats you plan?
ham	Going to take your babe out ?
ham	No need lar. Jus testing e phone card. Dunno network not gd i thk. Me waiting 4 my sis 2 finish bathing so i can bathe. Dun disturb u liao u cleaning ur room.

```
X_train, y_train = train_df["sms"], train_df["target"]
X_test, y_test = test_df["sms"], test_df["target"]
clf = make_pipeline(
    CountVectorizer(max_features=5000), LogisticRegression(max_iter=5000)
)
clf.fit(X_train, y_train);
```

Predicting on unseen data using the trained model

```
pd.DataFrame(X_test[0:4])
```

	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.

Note

Do not worry about the code and syntax for now.

```
pred_dict = {
    "sms": X_test[0:4],
    "spam_predictions": clf.predict(X_test[0:4]),
}
pred_df = pd.DataFrame(pred_dict)
pred_df.style.set_properties(**{"text-align": "left"})
```

		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	
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.	ham	

We have accurately predicted labels for the unseen text messages above!

(Supervised) machine learning: popular definition

A field of study that gives computers the ability to learn without being explicitly programmed.

-- Arthur Samuel (1959)

ML is a different way to think about problem solving.

Logical and mathematical

Traditional programming



Example: Given a number n and a program to calculate square root using the Newton-Raphson method, output the square root of n .

Machine learning



Example: Given a dataset containing images of cats and dogs and their corresponding labels, **learn a program** that predicts the animal in an unseen image.

Statistical and incorporating uncertainty

Examples

Let's look at some concrete examples of supervised machine learning.

Note

Do not worry about the code at this point. Just focus on the input and output in each example.

Example 1: Predicting whether a patient has a liver disease or not

Input data

Suppose we are interested in predicting whether a patient has the disease or not. We are given some tabular data with inputs and outputs of liver patients, as shown below. The data contains a number of input features and a special column called "Target" which is the output we are interested in predicting.

i Note

Download the data from [here](#).

i Note

Download the data from [here](#).

▶ Show code cell source

Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransf
40	14.5	6.4	358	
33	0.7	0.2	256	
24	0.7	0.2	188	
60	0.7	0.2	171	
18	0.8	0.2	199	

Building a supervised machine learning model

Let's train a supervised machine learning model with the input and output above.

```
from lightgbm.sklearn import LGBMClassifier

X_train = train_df.drop(columns=["Target"])
y_train = train_df["Target"]
X_test = test_df.drop(columns=["Target"])
y_test = test_df["Target"]
model = LGBMClassifier(random_state=123, verbose=-1)
model.fit(X_train, y_train)
```



LGBMClassifier



```
LGBMClassifier(random_state=123, verbose=-1)
```

Model predictions on unseen data

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

```
HTML(X_test.reset_index(drop=True).to_html(index=False))
```

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

```
pred_df = pd.DataFrame({"Predicted_target": model.predict(X_test).tolist()})

df_concat = pd.concat([pred_df, X_test.reset_index(drop=True)], axis=1)
HTML(df_concat.to_html(index=False))
```

Predicted_target	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransf
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 2: Predicting the label of a given image

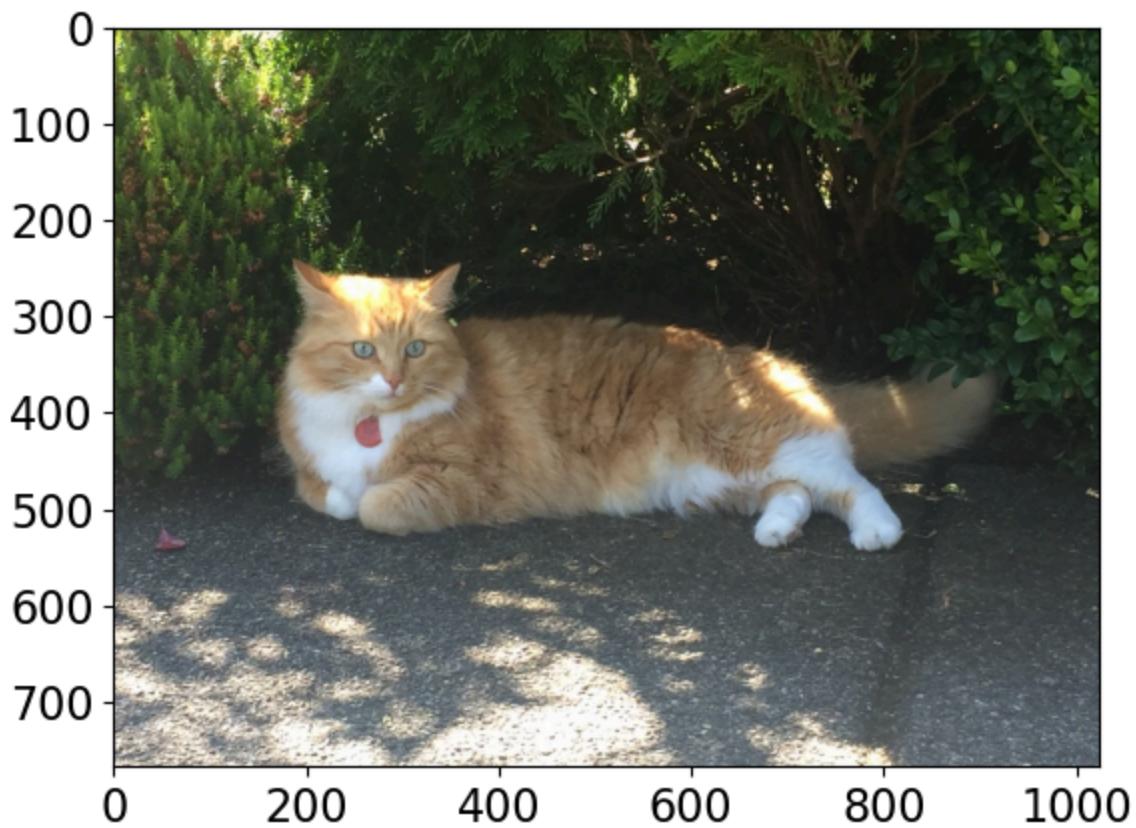
Suppose you want to predict the label of a given image using supervised machine learning. We are using a pre-trained model here to predict labels of new unseen images.

i Note

Assuming that you have successfully created 571 `conda` environment on your computer, you'll have to install `torchvision` in 571 `conda` environment to run the following code. If you are unable to install `torchvision` on your laptop, please don't worry at this point. It's not crucial at this point.

```
conda activate 571 conda install -c pytorch torchvision
```

```
import glob  
  
import img_classify  
import matplotlib.pyplot as plt  
from PIL import Image  
  
# Predict topn labels and their associated probabilities for unseen images  
images = glob.glob(DATA_DIR + "test_images/*.*")  
class_labels_file = DATA_DIR + "/imagenet_classes.txt"  
for img_path in images:  
    img = Image.open(img_path).convert("RGB")  
    img.load()  
    plt.imshow(img)  
    plt.show()  
    df = img_classify.classify_image(img_path, class_labels_file)  
    print(df.to_string(index=False))  
    print("-----")
```



Example 3: Predicting sentiment expressed in a movie review

Suppose you are interested in predicting whether a given movie review is positive or negative. You can do it using supervised machine learning.

Note

Download the data from [here](#). You'll have to rename it to `imdb_master.csv` and move it under `../data/` directory.```

▶ Show code cell source

```
sentiment
positive    25000
negative    25000
Name: count, dtype: int64
```

```
imdb_df = pd.read_csv(DATA_DIR + "imdb_master.csv", encoding="ISO-8859-1")
train_df, test_df = train_test_split(imdb_df, test_size=0.10, random_state=123)
HTML(train_df.head().to_html(index=False))
```

review	sentiment
<p>Often tagged as a comedy, The Man In The White Suit is laying out far more than a chuckle here and there.

Sidney Stratton is an eccentric inventor who isn't getting the chances to flourish his inventions on the world because nobody pays him notice, he merely is the odd ball odd job man about the place as it were. After bluffing his way into Birnley's textile mill, he uses their laboratory to achieve his goal of inventing a fabric that not only never wears out, but also never needs to be cleaned!. He is at first proclaimed a genius and those who ignored him at first suddenly want a big piece of him, but then the doom portents of an industry going bust rears its head and acclaim quickly turns to something far more scary.

Yes the film is very funny, in fact some scenes are dam hilarious, but it's the satirical edge to the film that lifts it way above the ordinary to me. The contradictions about the advent of technology is a crucial theme here, do we want inventions that save us fortunes whilst closing down industries ?, you only have to see what happened to the coal industry in Britain to know what I'm on about. The decade the film was made is a crucial point to note, the making of nuclear weapons became more than just hearsay, science was advancing to frighteningly new proportions. You watch this film and see the quick turnaround of events for the main protagonist Stanley, from hero to enemy in one foul swoop, a victim of his own pursuit to better mankind !, it's so dark the film should of been called The Man In The Black Suit.

I honestly can't find anything wrong in this film, the script from Roger MacDougall, John Dighton, and director Alex Mackendrick could be filmed today and it wouldn't be out of place such is the sharpness and thought of mind it has. The sound and setting is tremendous, the direction is seamless, with the tonal shift adroitly handled by Mackendrick. Some of the scenes are just wonderful, one in particular tugs on the heart strings and brings one to think of a certain scene in David Lynch's Elephant Man some 29 years later, and yet after such a downturn of events the film still manages to take a wink as the genius that is Alec Guinness gets to close out the film to keep the viewers pondering not only the future of Stanley, but also the rest of us in this rapidly advancing world.

A timeless masterpiece, thematically and as a piece of art, 10/10.</p>	positive
<p>Dirty Harry goes to Atlanta is what Burt called this fantastic, first-rate detective thriller that borrows some of its plot from the venerable Dana Andrews movie "Laura." Not only does Burt Reynolds star in this superb saga but he also helmed it and he doesn't make a single mistake either staging the action or with his casting of characters. Not a bad performance in the movie and Reynolds does an outstanding job of directing it. Henry Silva is truly icy as a hit-man.

Detective Tom Sharky (Burt Reynolds) is on a narcotics case in underground Atlanta when everything goes wrong. He winds up chasing a suspect and shooting it out with the gunman on a bus. During the melee, an innocent bystander dies. John Woo's "The Killer" replicates this scene. Anyway, the Atlanta Police Department busts Burt down to Vice and he takes orders from a new boss, Frisco (Charles Durning of "Oh, Brother, Where Art Thou?") in the basement. Sharky winds up in a real cesspool of crime. Sharky and his fellow detectives Arch (Bernie Casey) and Papa (Brian Keith) set up surveillance on a high-priced call girl Dominoe (Rachel Ward of "After Dark, My Sweet") who has a luxurious apartment that she shares with another girl.

Dominoe is seeing a local politician Hotchkins (Earl Holliman of "Police Woman") on the side who is campaigning for governor but the chief</p>	positive

	review	sentiment
villain, Victor (Vittorio Gassman of "The Dirty Game") wants him to end the affair. Hotchkins is reluctant to accommodate Victor, so Victor has cocaine snorting Billy Score (Henry Silva of "Wipeout") terminate Dominoe. Billy blasts a hole the size of a twelve inch pizza in the door of Dominoe's apartment and kills her. Sharky has done the unthinkable. During the surveillance, he has grown fond of Dominoe to the point that he becomes hopelessly infatuated with her. Sharky's mission in life now is to bust Victor, but he learns that Victor has an informant inside the Atlanta Police Department. The plot really heats up when Sharky discovers later that Billy shot the wrong girl and that Dominoe is still alive! Sharky takes her into protective custody and things grow even more complicated. He assembles his "Machine" of the title to deal with Victor and his hoods. William Fraker's widescreen lensing of the action is immaculate. Unfortunately, this vastly underrated classic is available only as a full-frame film. Fraker definitely contributes to the atmosphere of the picture, especially during the mutilation scene on the boat when the villain's cut off one of Sharky's fingers. This is a rather gruesome scene. Burt never made a movie that surpassed "Sharky's Machine."		
Yah. I know. It has the name "Sinatra" in the title, so how bad can it be? Well, it's bad, trust me! I rented this thinking it was some movie I missed in the theaters. It's not. It's some garbage "movie" made by the folks at Showtime (cable station). Geez, these cable stations make a few bucks they think they can make whatever garbage movies they want! It's not good. I am as big a Sinatra fan as any sane man, but this movie was just dumb. Boring. Dull. Unfunny. Uninteresting. The only redeeming quality is that (assuming they did stick to the facts) you do learn about what happened to the captors of Frank Jr. Otherwise it's just a stupid film.	negative	
The other lowest-rating reviewers have summed up this sewage so perfectly there seems little to add. I must stress that I've only had the Cockney Filth imposed on me during visits from my children, who insist on watching the Sunday omnibus. My god, it's depressing! Like all soaps, it consists entirely of totally unlikeable characters being unpleasant to each other, but it's ten times as bad as the next worst one could be. The reviewer who mocked the 'true to life' bilge spouted by its defenders was spot-on. If anyone lived in a social environment like this, they'd slash their wrists within days. And I can assure anyone not familiar with the real East End that it's rather more 'ethnically enriched' than you'll ever see here. Take my advice - avoid this nadir of the British TV industry. It is EVIL.	negative	
Made the unfortunate mistake of seeing this film in the Edinburgh film festival. It was well shot from the outset, but that's the last positive comment I have about the film. The acting was awful, I wonder if actual gogo girls were hired? But it was the plot that was truly laughable, in fact that it was laughable and not boring is the only reason I gave this 3/10. ** Spoilers below **. I just want to mention a few of the scenes that really got the audience laughing: Shoving the girl in the field: who would have thought that a kid shoving another kid could be acted so badly. A real eye-opener. The getting on the bus scene: the girl is getting on the bus. But, according to the music, the world is ending. The rolling under the clothes line: Wow, this one really demonstrates the plot writer's	negative	

review	sentiment
<p>skills. In the room, followed by raw meat and skill selling. Why not just get her to perform all three 'sins' at once? At least then the film might have been slightly shorter.

The running down the stairs of the mall: watch as one of the girls takes to flight down the stairs pursued by a flesh eating Dau, no wait .. she *is* just walking quickly trying not to break her nails.

The running covered in blood: this is definitely my favourite scene, and a fitting end to the movie. A half marathon in red paint, completed by vaulting up stairs and over the bridge, only to be sent flying most unrealistically by a passing car. Not only this, but this suicide is undertaken by the most self obsessed girl in the film, now that's sticking to character for you.

I'd like to think that this film was created by a 16 year old and their mates. Sadly, having met the director at the presentation, this is not the case.

But, if you're in a sarcastic mood, and fancy a laugh with a few mates.. then still don't even think about it.</p>	

```
# Build an ML model
X_train, y_train = train_df["review"], train_df["sentiment"]
X_test, y_test = test_df["review"], test_df["sentiment"]

clf = make_pipeline(
    CountVectorizer(max_features=5000), LogisticRegression(max_iter=5000)
)
clf.fit(X_train, y_train);
```

```
# Predict on unseen data using the built model
pred_dict = {
    "reviews": X_test[0:4],
    "sentiment_predictions": clf.predict(X_test[0:4]),
}
pred_df = pd.DataFrame(pred_dict)
pred_df.style.set_properties(**{"text-align": "left"})
```

	This movie was beyond awful, it was a pimple on the a*s of the movie industry. I know that every movie can't be a hit or for that matter even average, but the responsible parties that got together for this epic dud, should have been able to see that they had a ticking time bomb on their hands. I can't help but think that the cast would get together in between scenes and console each other for being in such a massive heap of dung. I can hear it now, "You getting' paid?" "Nope, you?" I understand that this flick was more than likely made on a shoe string budget but even with that taken into account, it still could've been better. You wait for the appearance of a monster/creature and when you finally see it, it's a big yawn.I'm so mad at myself for spending a 1.07 on this stinker!!!	negative
40828	As of this writing John Carpenter's 'Halloween' is nearing it's 30th anniversary. It has since spawned 7 sequels, a remake, a whole mess of imitations and every year around Halloween when they do those 'Top 10 Scariest Movies' lists it's always on there. That's quite amazing for a film that was made on a budget of around \$300,000 and featured a then almost completely unknown cast of up and coming young talent. I could go on and on, but the big question here is: How does the film hold up today? And all I can say to that is, fantastically!	positive

Pros: A simple, but spooky opening credits sequence that really sets the mood. An unforgettable and goosebump-inducing score by director/co-writer John Carpenter and Alan Howarth. Great cinematography. Stellar direction by Carpenter who keeps the suspense high, gets some great shots, and is careful not to show too much of his villain. Good performances from the then mostly unknown cast. A good sense of humor. Michael Myers is one scary, evil guy. A lot of eerie moments that'll stay with you. The pace is slow, but steady and never drags. Unlike most other slasher films, this one is more about suspense and terror than blood and a big body count.

Cons: Probably not nearly as scary now as it was then. Many of the goofs really stand out.

Final thoughts: I want to start out this section by saying this is not my favorite film in the series. I know that's not a popular opinion, but it's really how I feel. Despite that it truly is an important film that keeps reaching new generations of film buffs. And just because it's been remade for a new generation doesn't mean it'll be forgotten. No way, no how.

My rating: 5/5

- 36400** I must admit a slight disappointment with this film; I had read a lot about how spectacular it was, yet the actual futuristic sequences, the Age of Science, take up a very small amount of the film. The sets and are excellent when we get to them, and there are some startling images, but this final sequence is lacking in too many other regards... positive

Much the best drama of the piece is in the mid-section, and then it plays as melodrama, arising from the 'high concept' science-fiction nature of it all, and insufficiently robust dialogue. There is far more human life in this part though, with the great Ralph Richardson sailing gloriously over-the-top as the small dictator, the "Boss" of the Everytown. I loved Richardson's mannerisms and curt delivery of lines, dismissing the presence and ideas of Raymond Massey's aloof, confident visitor. This Boss is a posturing, convincingly deluded figure, unable to realise the small-fry nature of his kingdom... It's not a great role, yet Richardson makes a lot of it.

Everytown itself is presumably meant to be England, or at least an English town fairly representative of England. Interesting was the complete avoidance of any religious side to things; the 'things to come' seem to revolve around a conflict between warlike barbarism and a faith in science that seems to have little ultimate goal, but to just go on and on. There is a belated attempt to raise some arguments and tensions in the last section, concerning more personal 'life', yet one is left quite unsatisfied. The film hasn't got much interest in subtle complexities; it goes for barnstorming spectacle and unsubtle, blunt moralism, every time. And, of course, recall the hedged-bet finale: Raymond Massey waxing lyrical about how uncertain things are!

Concerning the question of the film being a prediction: I must say it's not at all bad as such, considering that one obviously allows that it is impossible to get the details of life anything like right. The grander conceptions have something to them; a war in 1940, well that was perhaps predictable... Lasting nearly 30 years, mind!? A nuclear bomb - the "super gun" or some such contraption - in 2036... A technocratic socialist "we don't believe in independent nation states"-type government, in Britain, after 1970... Hmm, sadly nowhere near on that one, chaps! ;-) No real politics are gone into here which is a shame; all

that surfaces is a very laudable anti-war sentiment. Generally, it is assumed that dictatorship - whether boneheaded-luddite-fascist, as under the Boss, or all-hands-to-the-pump scientific socialism - will *be the deal*, and these implications are not broached... While we must remember that in 1936, there was no knowledge at all of how Nazism and Communism would turn out - or even how they were turning out - the lack of consideration of this seems meek beside the scope of the filmmakers' vision on other matters.

Much of the earlier stuff should - and could - have been cut in my opinion; only the briefest stuff from '1940' would have been necessary, yet this segment tends to get rather ponderous, and it is ages before we get to the Richardson-Massey parts. I would have liked to have seen more done with Margareta Scott; who is just a trifle sceptical, cutting a flashing-eyed Mediterranean figure to negligible purpose. The character is not explored, or frankly explained or exploited, except for one scene which I shall not spoil, and her relationship with the Boss isn't explored; but then this was the 1930s, and there was such a thing as widespread institutional censorship back then. Edward Chapman is mildly amusing in his two roles; more so in the first as a hapless chap, praying for war, only to be bluntly put down by another Massey character. Massey himself helps things a lot, playing his parts with a mixture of restraint and sombre gusto, contrasting well with a largely diffident cast, save for Richardson, and Scott and Chapman, slightly.

I would say that "Things to Come" is undoubtedly a very extraordinary film to have been made in Britain in 1936; one of the few serious British science fiction films to date, indeed! Its set (piece) design and harnessing of resources are ravenous, marvellous.

Yet, the script is ultimately over-earnest and, at times, all over the place. The direction is prone to a flatness, though it does step up a scenic gear or two upon occasion. The cinematographer and Mr Richardson really do salvage things however; respectively creating an awed sense of wonder at technology, and an engaging, jerky performance that consistently beguiles. Such a shame there is so little substance or real filmic conception to the whole thing; Powell and Pressburger would have been the perfect directors to take on such a task as this - they are without peer among British directors as daring visual storytellers, great helmsmen of characters and dealers in dialogue of the first rate.

"Things to Come", as it stands, is an intriguing oddity, well worth perusing, yet far short of a "Metropolis"... 'Tis much as "silly", in Wells' words, as that Lang film, yet with nothing like the astonishing force of it.

Oh dear! The BBC is not about to be knocked off its pedestal for absorbing period dramas by this one. I agree this novel of Jane Austens is the difficult to portray particularly to a modern audience, the heroine is hardly a Elizabeth Bennet, even Edmund is not calculated to cause female hearts to skip a beat. However I must say I was hoping for an improvement on the last and was sadly disappointed. The basic story was preserved, but the dialogue was so altered that all

- 5166 that was Jane Austen's tone, manner, feeling, wit, negative
depth, was diluted if not lost. If some past adaptions may be seen as dated the weakness of this one must be that it is too modern ('his life is one long party'?????) The cast was generally adequate, but I think Billie Piper was the wrong choice, it needed someone more restrained, I gained no impression of hidden depths beneath a submissive exterior, she was more like a frolicking child. I see I must wait for the BBC to weave its magic once again.

Example 4: Predicting housing prices

Suppose we want to predict housing prices given a number of attributes associated with houses.

Note

Download the data from [here](#).

► Show code cell source

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

```
# Build 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(verbose=-1)
model.fit(X_train, y_train);
```

```
# Predict on unseen examples using the built model
pred_df = pd.DataFrame(
    # {"Predicted target": model.predict(X_test[0:4]).tolist(), "Actual price": X_test[0:4].values[0][0]}
    {"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	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

To summarize, supervised machine learning can be used on a variety of problems and different kinds of data.

🤔 Eva's questions

At this point, Eva is wondering about many questions.

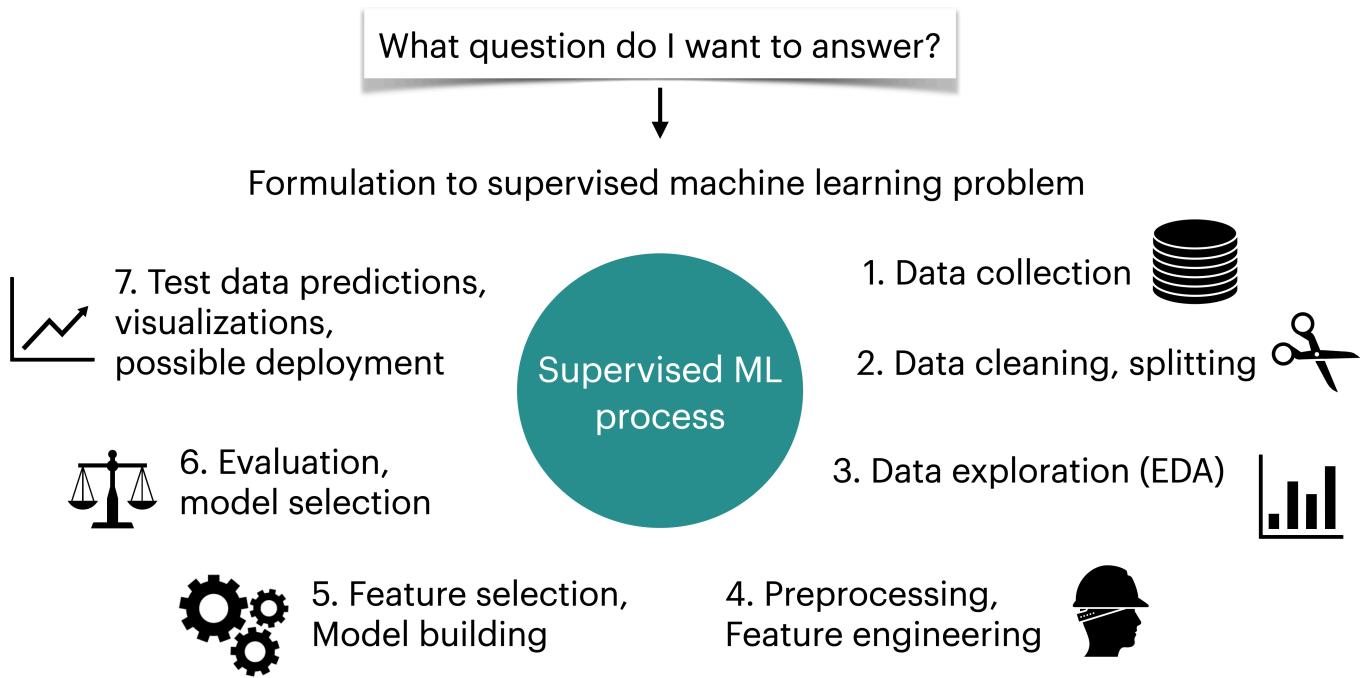
- How are we exactly “learning” whether a message is spam and ham?
- What do you mean by “learn without being explicitly programmed”? The code has to be somewhere ...
- Are we expected to get correct predictions for all possible messages? How does it predict the label for a message it has not seen before?
- 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!



Machine learning workflow

Supervised machine learning is quite flexible; it can be used on a variety of problems and different kinds of data. Here is a typical workflow of a supervised machine learning systems.



We will build machine learning pipelines in this course, focusing on some of the steps above.

(Optional) Surveys

- Please complete the "Getting to know you" survey on [Canvas](#).
- Also, please complete the anonymous restaurant survey on Qualtrics [here](#).
 - We will analyze this data set in the coming weeks.

About this course

Course learning outcomes

By the end of this course you'll be able to answer most of the above questions.

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

- Describe supervised learning and its suitability for various tasks.
- Explain key machine learning concepts such as classification, regression, overfitting, and the trade-off in model complexity.
- Identify appropriate data preprocessing techniques for specific scenarios, provide reasons for their selection, and integrate them into machine learning pipelines.
- Develop an intuitive understanding of common machine learning algorithms.
- Build end-to-end supervised machine learning pipelines using Python and scikit-learn on real-world datasets.

What we won't cover

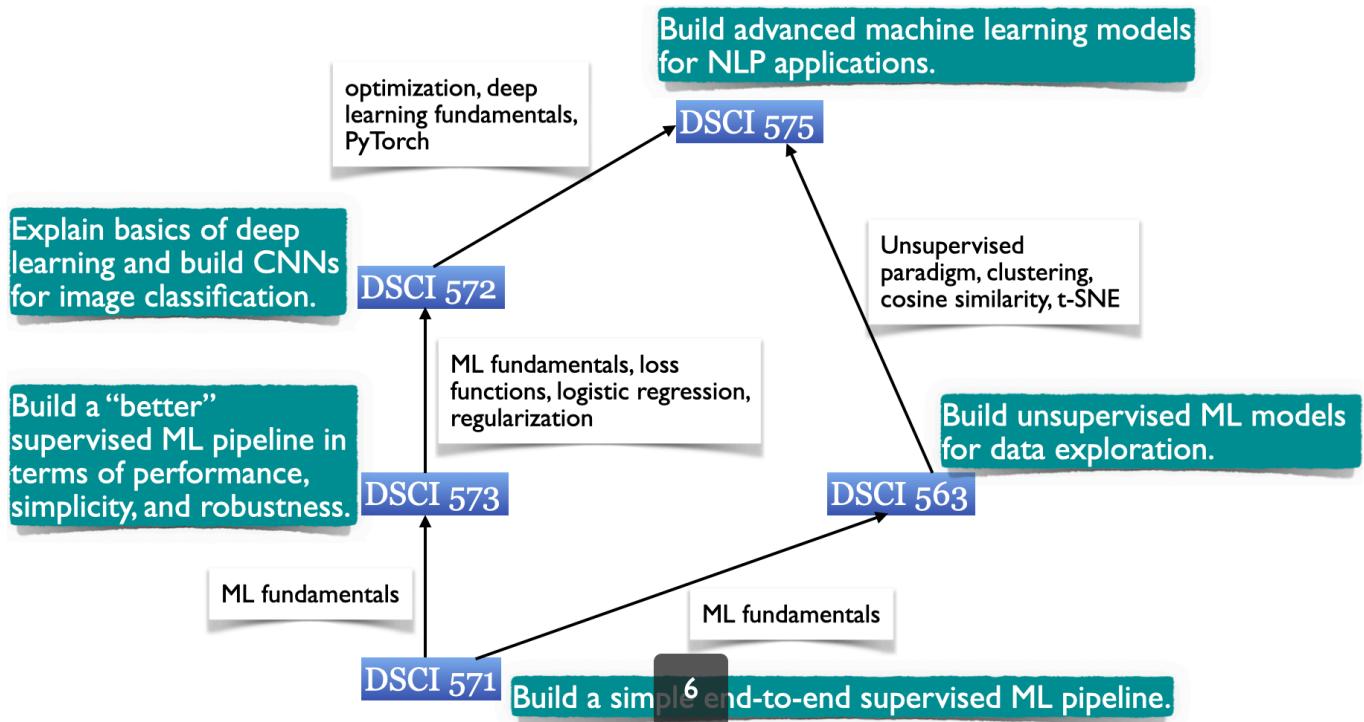
- I'll give you intuition of different ML algorithms but you won't be implementing the algorithms on your own.
- Instead, we will focus on machine learning fundamentals, preprocessing, and building machine learning pipelines.

Course roadmap (tentative)

- Week 1
 - ML terminology, decision trees
 - ML fundamentals
- Week 2
 - KNNs, SVMs with RBF Kernels
 - Preprocessing
- Week 3
 - More preprocessing: text features

- Hyperparameter optimization, optimization bias
- Week 4
 - Naive Bayes
 - Linear models

Map of ML courses in MDS



Framework we will be using in this class

- A number machine learning libraries are available.
- We will be using `sklearn` library, which is a popular ([more than 59k stars on Github](#)) Machine Learning library for Python.

```

import IPython

url = "https://scikit-learn.org"
IPython.display.IFrame(width=1000, height=650, src=url)
  
```

Checklist for you before the first class

- [] Have you created [course](#) [conda](#) [environment](#) on your computer?
- [] Did you watch the [course information video](#)?
- [] Did you watch the [assigned pre-recorded lecture videos](#) for lecture 1?
- [] Have you read the [course README](#) carefully, in particular the [course communication](#) section?
- [] Are you on course Slack channel of your section?
- [] Are you able to access course [Canvas](#) shell?

- [] Are you enrolled in the [iClicker Cloud](#) for this course?
- [] Are you able to access [Gradescope](#)? (If not, refer to the [Gradescope Student Guide](#).)
- [] (Optional) Did you complete the anonymous [restaurant survey on Qualtrics](#)?
- [] (Optional) Did you complete the [Getting to know you](#) survey?

! Attention

Note that the first couple of lectures are kind of fundamental and the videos are longish. We are asking you to pre-watch all recorded videos for the first lecture before class. If this is too much workload for you please reach out to us and we'll reassess.

Summary

- Machine learning is a different paradigm for problem solving.
- Very often it reduces the time you spend programming and helps customizing and scaling your products.
- In supervised learning we are given a set of observations (X) and their corresponding targets (y) and we wish to find a model function f that relates X to y .
- Carefully read the course website. Make sure to complete the surveys.
- **The teaching team is dedicated to facilitating your learning and ensuring your success in the course.**
- If you have any concerns regarding the format, teaching approach, or any other aspect, please don't hesitate to get in touch. I will do my best to be accommodating and understanding.
- Let's have fun learning this material together!

