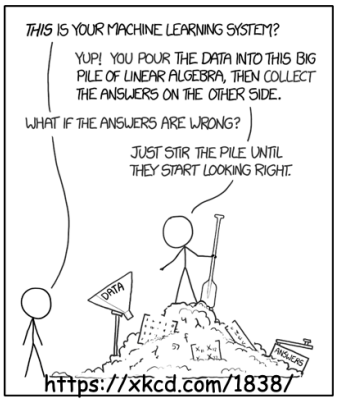
Course Information

Contents



DSCI 563 Unsupervised Learning

UBC Master of Data Science program, 2024-25

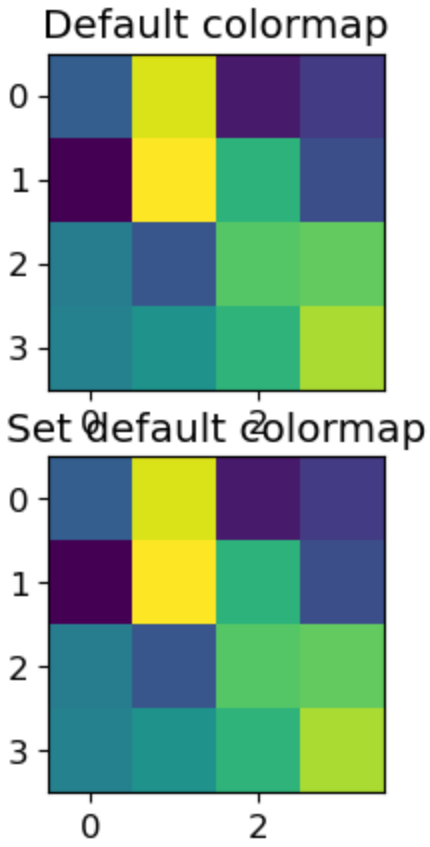
Imports

```
import sys

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

import sys, os
sys.path.append(os.path.join(os.path.abspath(".."), "code"))

from plotting_functions import *
from sklearn import datasets
```



Learning outcomes

From this lecture, students are expected to be able to:

- Explain what is unsupervised learning.
- Explain the difference between supervised and unsupervised learning.
- Name some example applications of unsupervised learning.

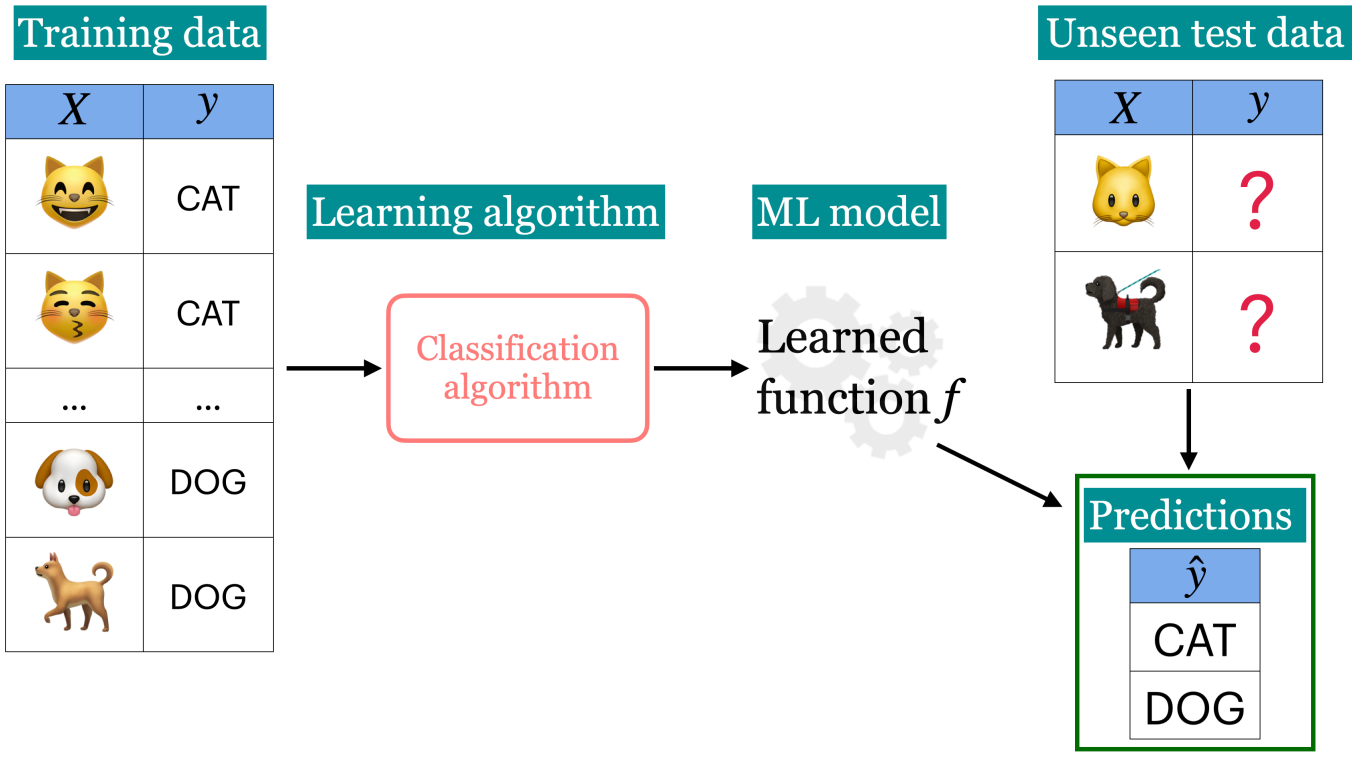
Types of machine learning

Recall the typical learning problems we discussed in 571.

- Supervised learning (Gmail spam filtering)
  - Training a model from input data and its corresponding targets to predict targets for new examples. (571, 572, 573)
- **Unsupervised learning** (this course) (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.

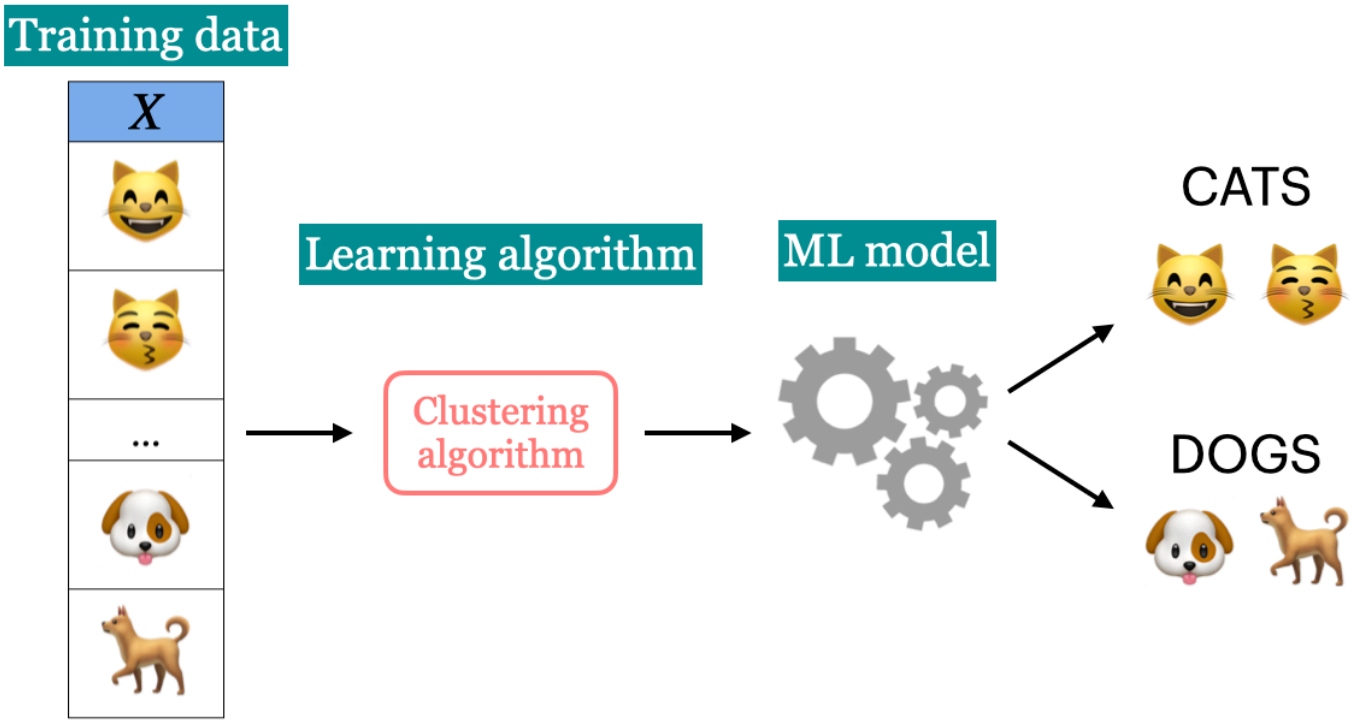
Supervised learning

- 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.
- Then use that model function to predict the targets of new examples.
- We have been working with this set up so far.



Unsupervised learning

- Training data consists of observations (X) without any corresponding targets.
- Unsupervised learning could be used to group similar things together in X or to find underlying structure in the data.



Can we learn without targets?

- Yes, but the learning will be focused on finding the underlying structures of the inputs themselves (rather than finding the function f between input and output like we did in supervised learning models).
- Examples:
  - Clustering
  - Dimensionality reduction

Labeled vs. Unlabeled data

- If you have access to labeled training data, you're in the "supervised" setting.
- You know what to do in that case from 571, 572, 573.
- Unfortunately, getting large amount of labeled training data is often time consuming and expensive.
- Annotated data can become "stale" after a while in cases such as fraud detection.
- Can you still make sense of the data even though you do not have the labels?
- Yes! At least to a certain extent!

Example: Supervised vs unsupervised learning

- In supervised learning, we are given features X and target y.

Dataset 1		**Dataset2**		
x_1	y	x_1	x_2	y
101.0	Sick	-2.68	0.32	class 1
98.5	Not Sick	-2.71	-0.18	class 1
93.8	Sick	1.28	0.69	class 2
104.3	Sick	0.93	0.32	class 2
98.6	Not Sick	1.39	-0.28	class 3

- In unsupervised learning, we are only given features X.

Dataset 1	Dataset 2	
x_1	x_1	x_2
101.0	-2.68	0.32
98.5	-2.71	-0.18
93.8	1.28	0.69
104.3	0.93	0.32
98.6	1.39	-0.28

An example with `sklearn` toy dataset

```
## Iris dataset
iris = datasets.load_iris()  # loading the iris dataset
features = iris.data[:, 2:4]  # only consider two features for visualization
labels = iris.target_names[
    iris.target
]  # get the targets, in this case the types of the Iris flower

iris_df = pd.DataFrame(features, columns=iris.feature_names[2:])
iris_df
```

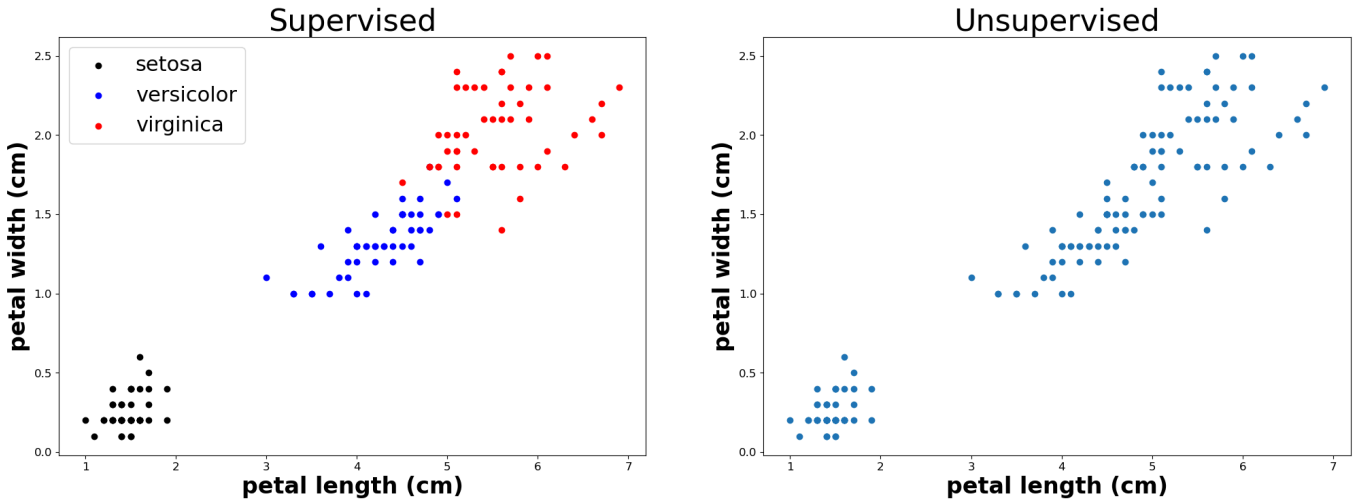
	petal length (cm)	petal width (cm)
0	1.4	0.2
1	1.4	0.2
2	1.3	0.2
3	1.5	0.2
4	1.4	0.2
...	...	...
145	5.2	2.3
146	5.0	1.9
147	5.2	2.0
148	5.4	2.3
149	5.1	1.8

150 rows × 2 columns

```
np.unique(labels)
```

```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

```
iris_df["target"] = labels
plot_sup_x_unsup(iris_df, 8, 8)
```

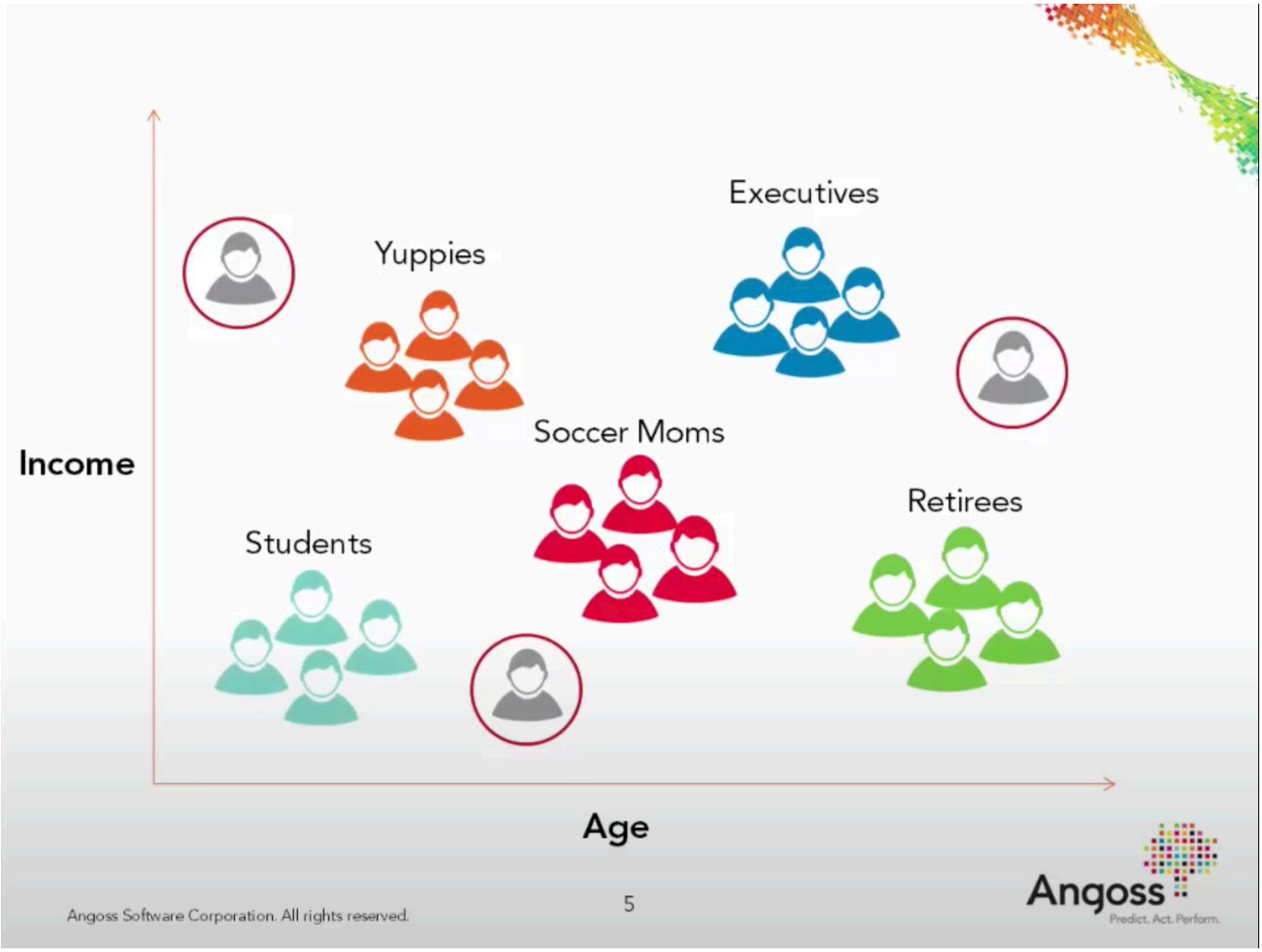


- In case of supervised learning, we're given X and y (showed with different colours in the plot above).
- In case of unsupervised learning, we're only given X and the goal is to identify the underlying structure in data.

Unsupervised learning applications

Example 1: Customer segmentation

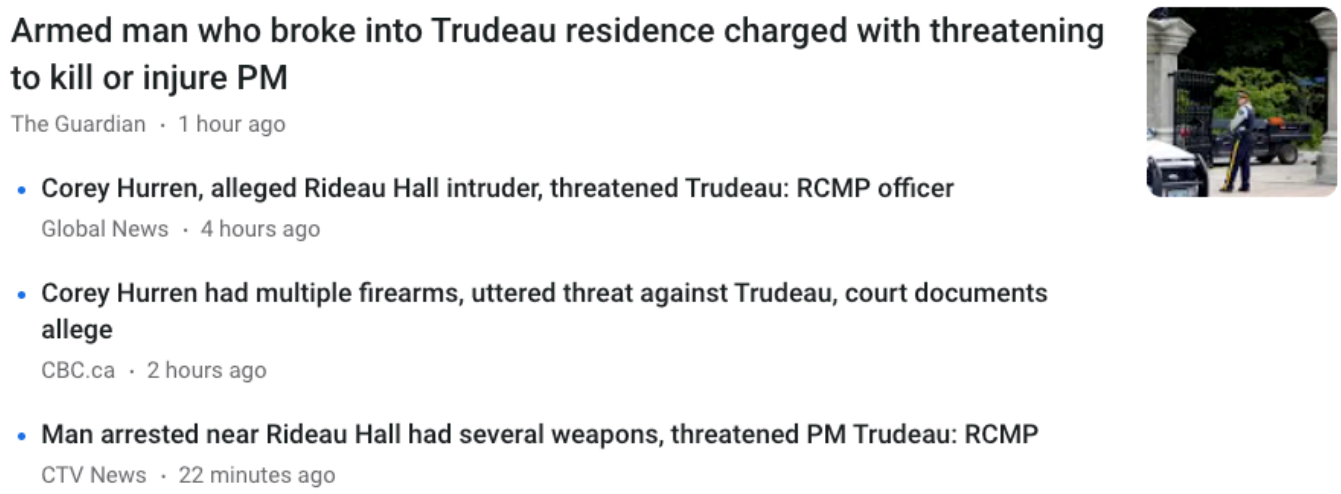
- Understand landscape of the market.



source

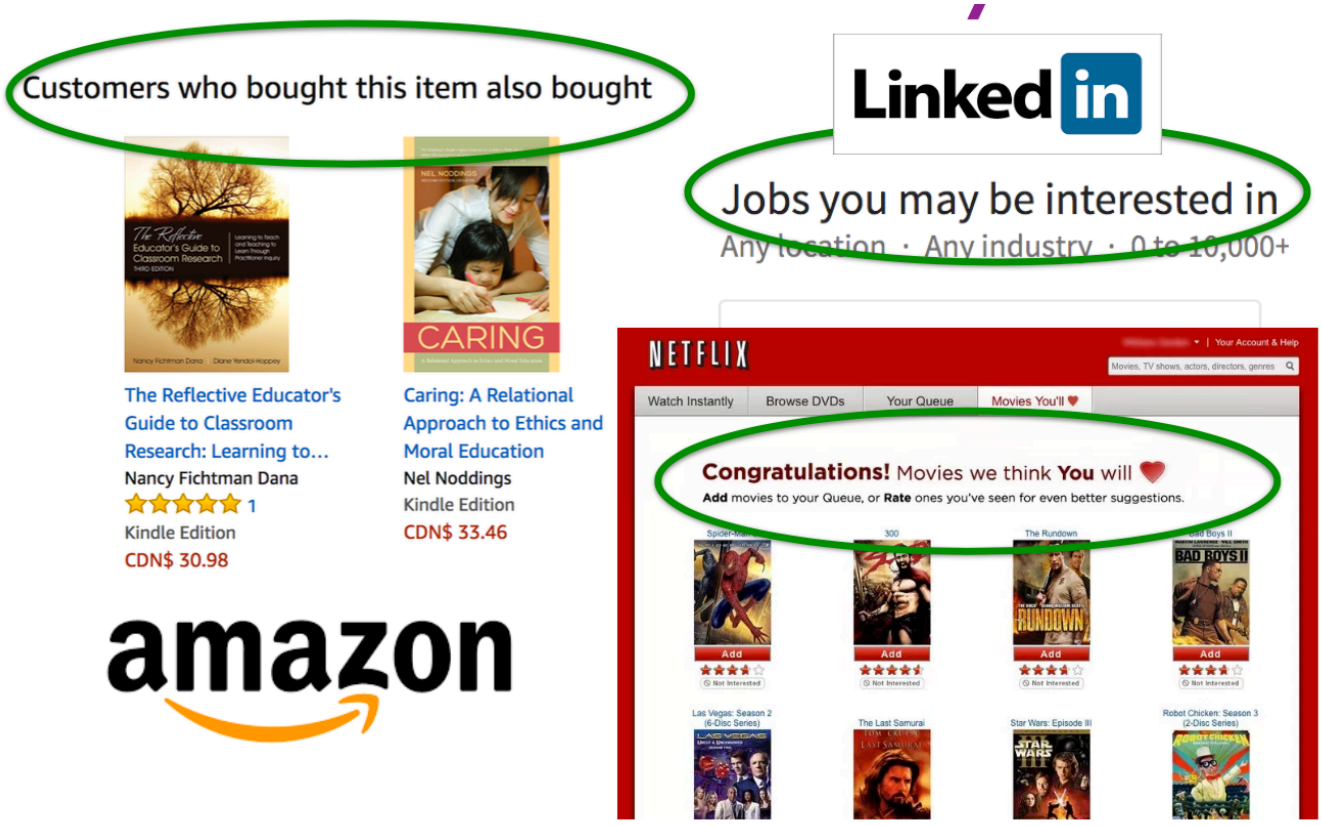
Example 2: Document clustering

Grouping articles on different topics from different news sources. For example, Google News.



Example 3: Recommender systems

- What should we recommend to the user, so that they buy another product?



Example 4: Examples projects from Capstone proposals

Here are some projects from Capstone proposals which would involve unsupervised learning.

- Creating (figurative) ecommerce shopping aisles with ML
- Life Decision Support: Choose your best career path
- ...

- If you want to build a machine learning model to cluster such images how would you represent such images?
- Imagine that we also have ratings data of food items and users for a large number of users. Can you exploit the power of community to recommend certain food items to a given user they are likely to consume?

Course roadmap

In this course we'll try to answer these questions. In particular, here is the roadmap of the course.

- Week 1
  - Clustering (How to group unlabeled data?)
- Week 2 and week 3
  - Dimensionality reduction (How to represent the data?)
  - Word embeddings
- Week 4
  - Recommender systems (How to exploit the power of community to recommend relevant products, services, items to customers they are likely to consume?)

Framework and tools

- `sklearn`
- `PyTorch`

I will be using `matplotlib`, `plotly`, `seaborn` for plotting simply because I am not very comfortable with `Altair`. The plotting is mainly for the demonstration of the concepts and you are not expected to learn these libraries for this class. You are free to use the libraries of your choice in labs but encouraged to use `Altair`.

Attributions

The material of this course is built on the material developed by amazing instructors who have taught this course before. In particular, many thanks to Mike Gelbart, Rodolfo Lourenzutti, and Giulio Valentino Dalla Riva.