Introduction to Machine Learning

Rasika Bhalerao
Northeastern University
Climate Change AI Summer School 2023

Agenda

- **What is machine learning?**
- Supervised learning
- Unsupervised learning
- Reinforcement learning
- Generative models

Machine Learning:

“The science of getting computers to act without being explicitly programmed.”

- Andrew Ng

When should we use machine learning?

- Machine learning is not appropriate for every task!
  - If you can solve it analytically, that's better! (More explainable)
- When you have access to **data which "matches" the task**
- When **errors are allowable**
- When it's **cost effective**
  - Machine learning costs include (i) **dataset creation and processing** and (ii) **model development, deployment, and maintenance**

Machine learning strengths and limitations

Strengths

- Performing tasks at scale
- Modeling complex systems
- Generating derived data
- Integrating with other methods, e.g. domain and physical models

Limitations

- “Garbage in, garbage out”
- Inherits biases in data + human design/use
- Assumes patterns are persistent
- Finds correlation, not causation

Types of learning

- **Supervised learning**
  Learning to predict or classify labels based on labeled input data
  Performance feedback

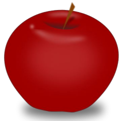
- **Unsupervised learning**
  Finding patterns in unlabeled data
  No performance feedback

- **Reinforcement learning**
  Learning well-performing behavior from state observations and rewards
  Performance feedback

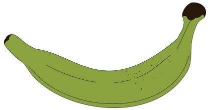
Supervised vs. Unsupervised learning

Supervised

Apple



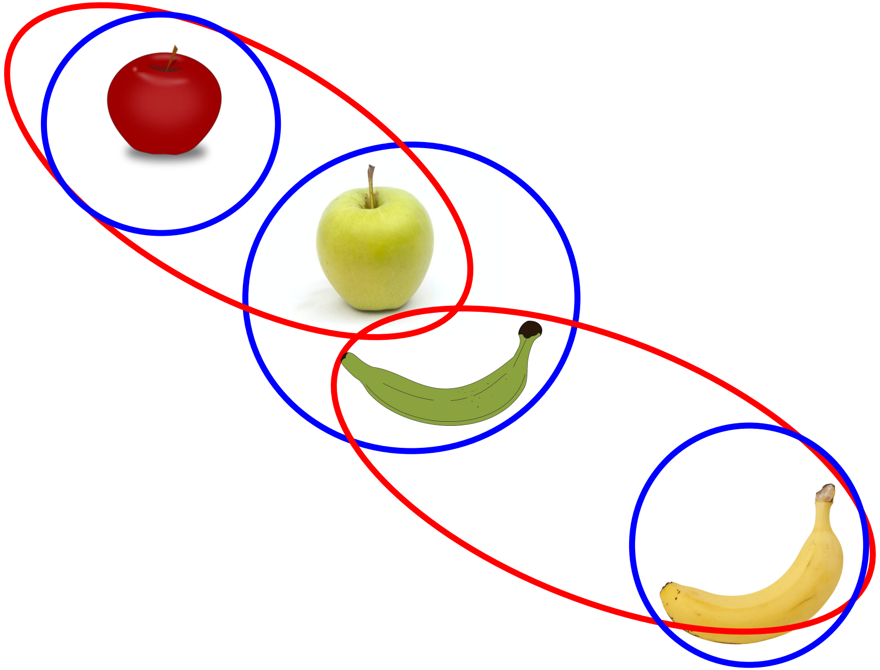
Apple



Banana

Banana

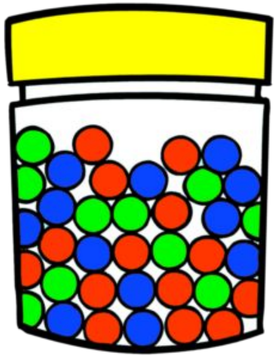
Unsupervised



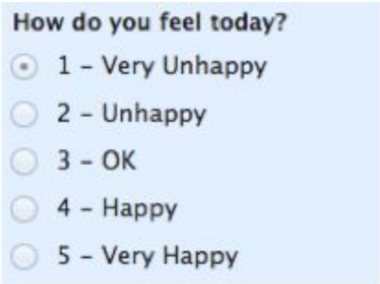
Data Types

- **Continuous**

- **Discrete**
- **Categorical**



- **Binary**
  - Special case of categorical
- **Ordinal**



How do you feel today?

1 – Very Unhappy
2 – Unhappy
3 – OK
4 – Happy
5 – Very Happy

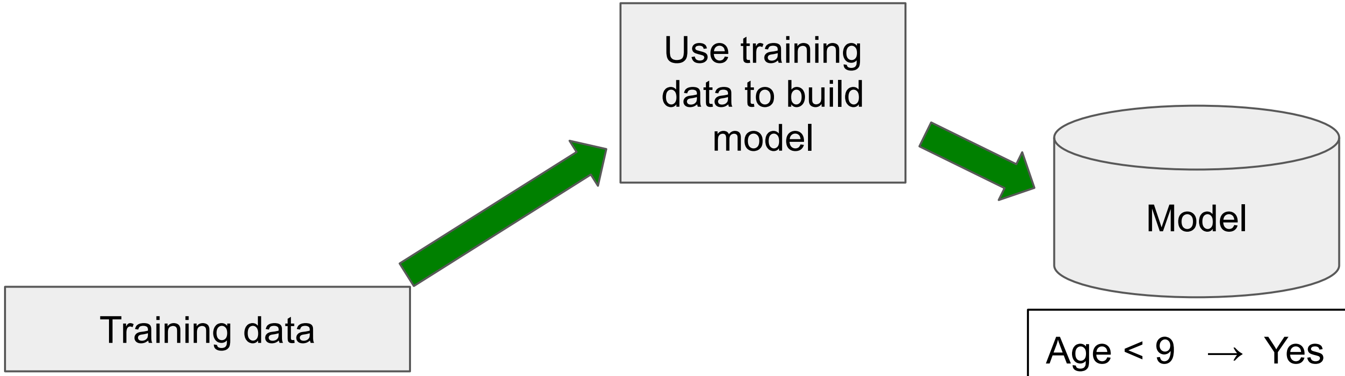
How satisfied are you with our service?

1 – Very Unsatisfied
2 – Somewhat Unsatisfied
3 – Neutral
4 – Somewhat Satisfied
5 – Very Satisfied

Agenda

- What is machine learning?
- **Supervised learning**
- Unsupervised learning
- Reinforcement learning
- Generative models

Supervised learning: training



Age < 9 → Yes
Age >= 9 → No

Person	**Age**	**Favorite food**	**Customer?**
Cat	8	Sushi	Yes
Elephant	13	Salad	No
Dog	7	Sandwich	Yes
Turtle	15	Sandwich	No

Supervised learning: test / prediction

Model

Age < 9 → Yes
Age >= 9 → No

Evaluate performance using test data

Person	**Age**	**Favorite food**	**Customer?**
Penguin	6	Sushi	Yes
Snake	15	Soup	No

Use on unseen real-world data

Person	**Age**	**Favorite food**
Chicken	8	Salad

Customer? Yes

Supervised learning: categorical versus continuous labels

- Classification: **categorical labels**
  - Examples: pregnant or not, from which country, which type of road sign
- Regression: **continuous labels**
  - Examples: future stock price, life expectancy, distance to obstacle

Example: predicting bicycle counts

https://www.climatechange.ai/papers/iclr2023/15

Given: historical data of the number of bicycles in certain locations per hour

Want to predict: number of bicycles in future times at those locations

Which type are the labels? Categorical or continuous?

Supervised models

Model	When to use it?
KNN	➢ Little / no training time, large prediction time ➢ Given small / medium dataset size
Linear / polynomial regression	➢ Linear / polynomial relationship between input and output ➢ Small training time ➢ Given small / medium dataset size
Logistic regression, SVM, decision tree	➢ Categorical output ➢ Given small / medium training time
Neural network	➢ Large training time, large computer ➢ Given large dataset size

k-Nearest Neighbors Algorithm

Training set: n instances, each with a feature vector and an output category

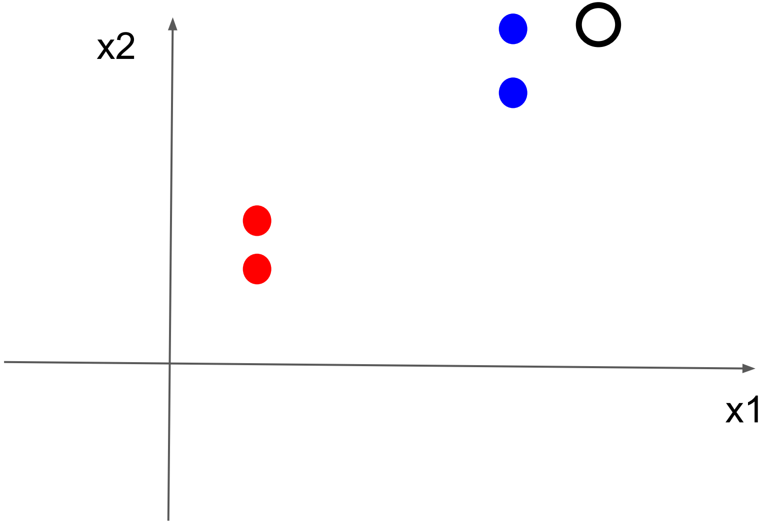
Now, given another (unseen) instance, we want to determine its category

Check the k instances in the training data that are closest to your new instance

- ➢ Categorical: choose the majority of those values
- ➢ Continuous: choose the mean/median of those values

Training set:
(1,2) → red
(1,3) → red
(5,5) → blue
(5,6) → blue

New instance:
(6,6) → blue



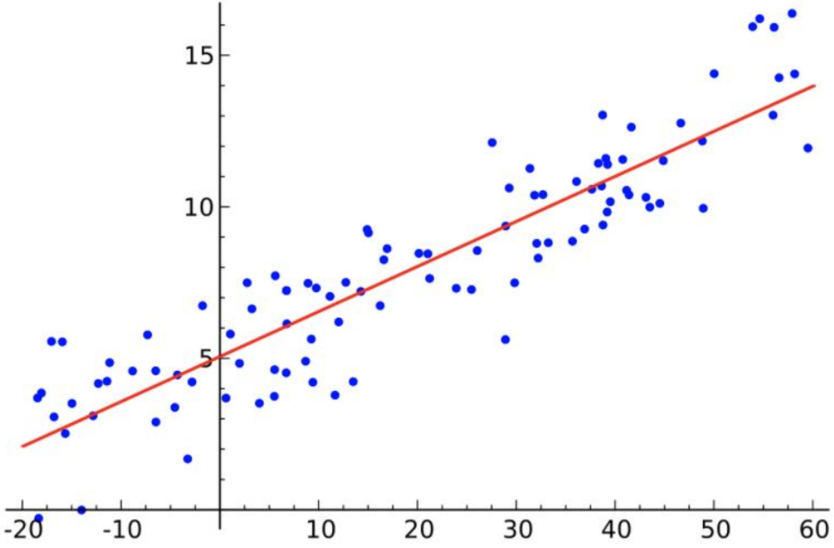
Linear / polynomial regression

Given x ϵ$\mathbb{R}$ and y $\mathbb{R}$

Find a function f: x → y

How?

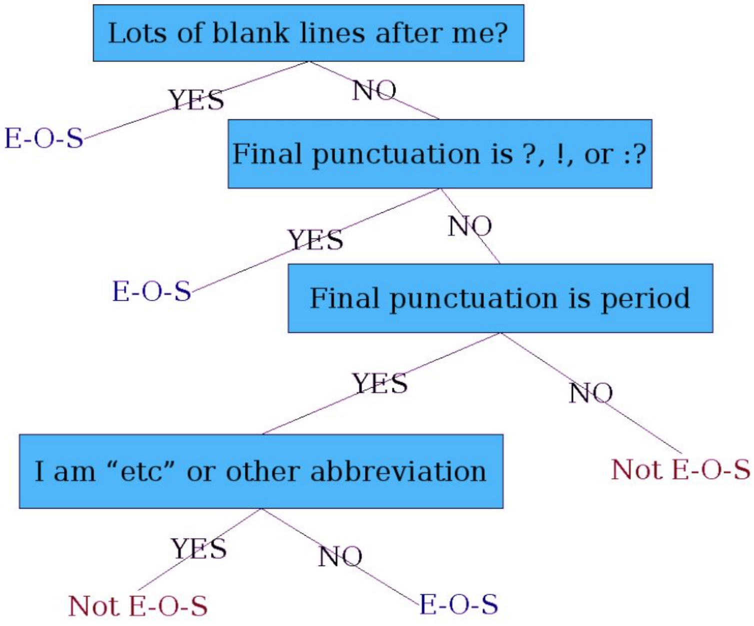
Define a **loss function** (“error”) and minimize it!



Other supervised classifiers

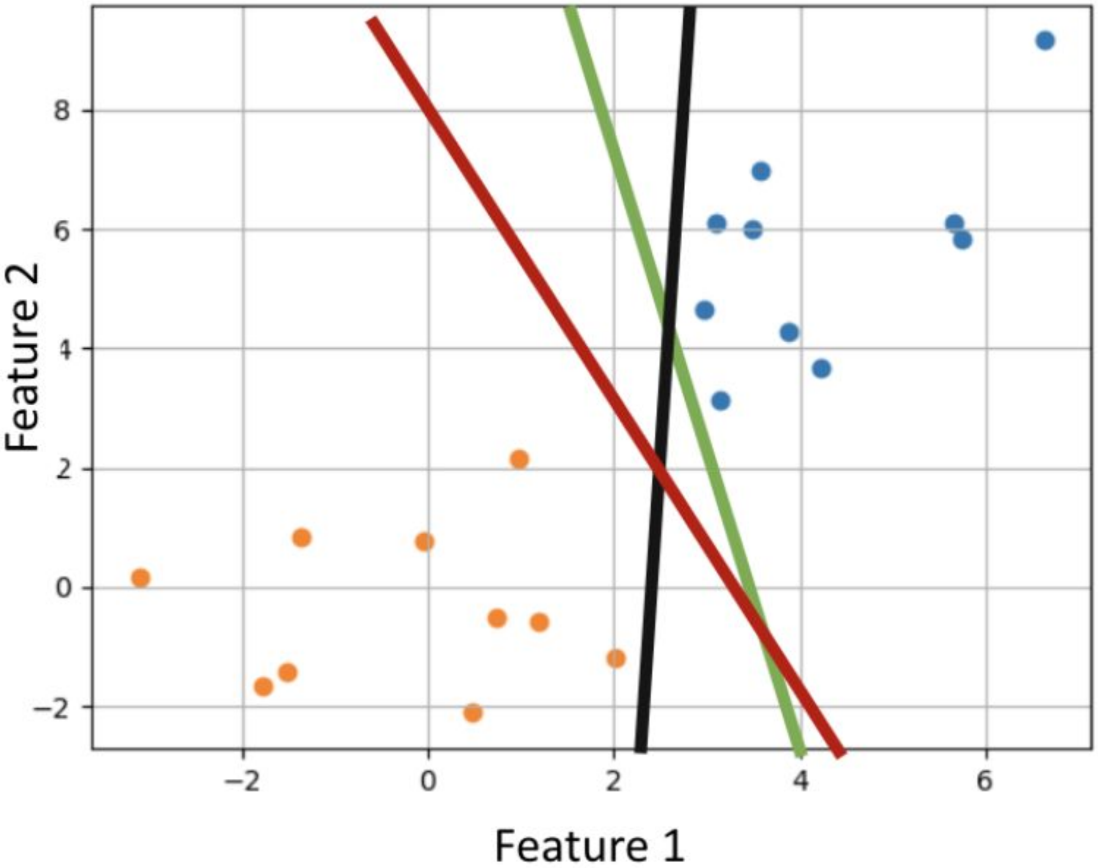
- **Decision tree**

Determining if a word is end-of-sentence: a Decision Tree



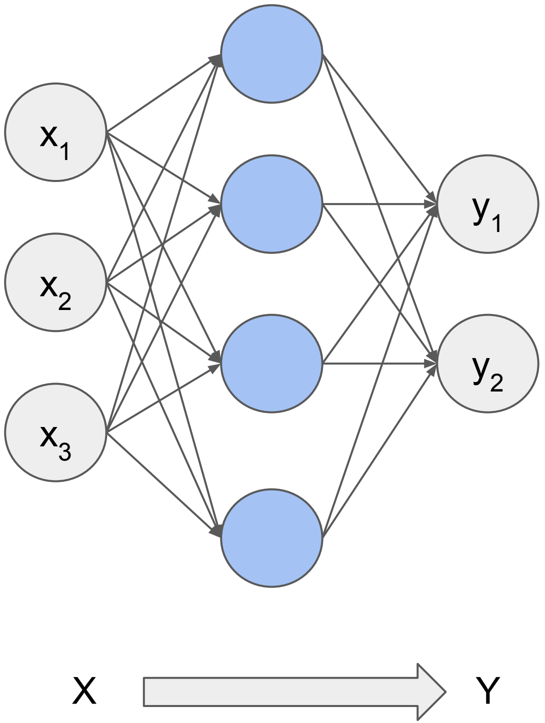
Other supervised classifiers

- **Decision tree**
- **Support Vector Machine**



Other supervised classifiers

- **Decision tree**
- **Support Vector Machine**
- **Neural Network**

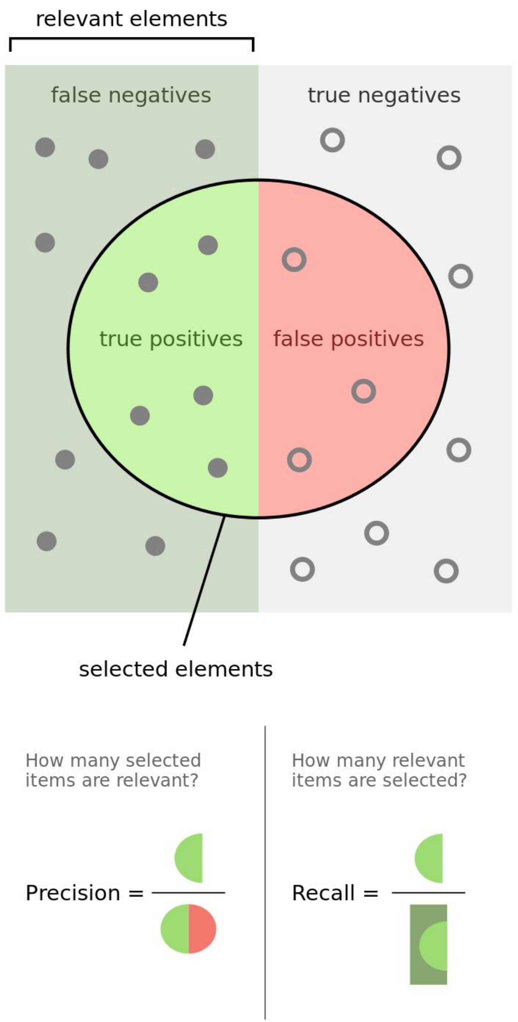


How good is the model?

We define a **metric** to measure and compare accuracy.

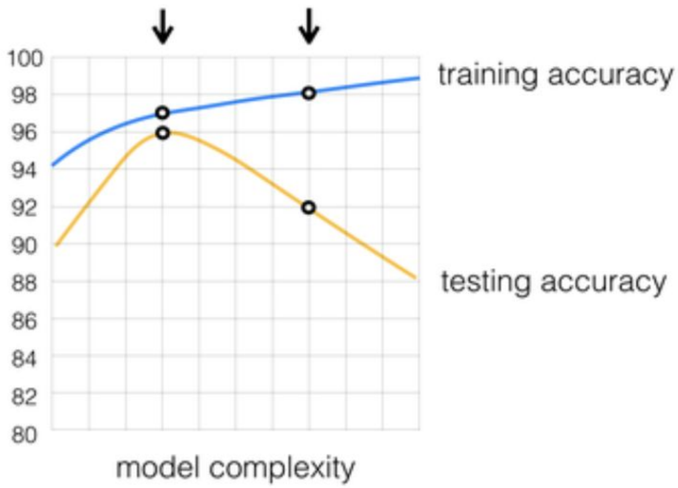
- Precision
  - Out of those tested positive, how many are truly positive?
  - TP / (TP + FP)
- Recall
  - Out of those truly positive, how many tested positive?
  - TP / (TP + FN)
- F1

$$\frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}}$$



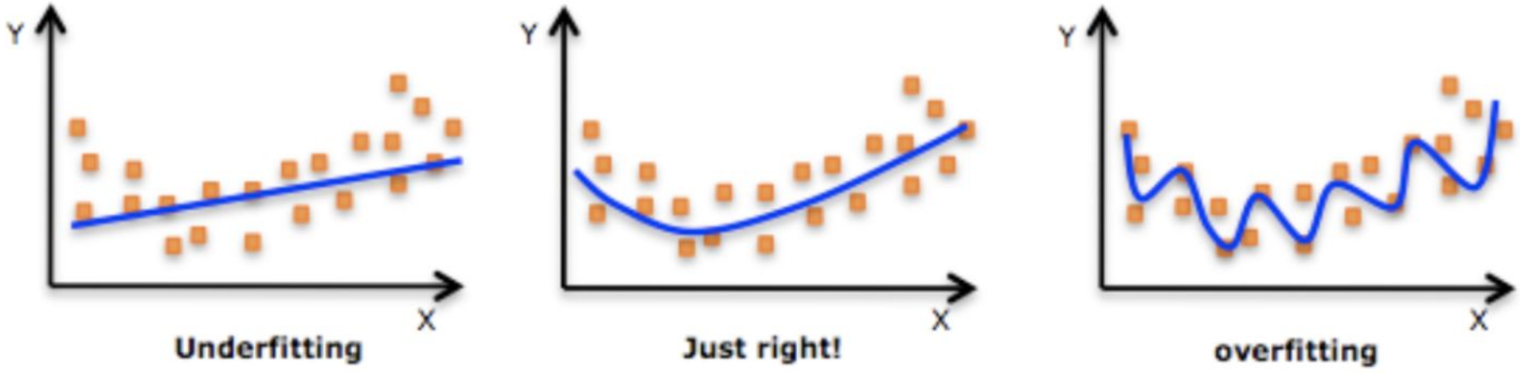
https://en.wikipedia.org/wiki/F1_score

Overfitting



Solution: **Cross-Validation**

Split the training data into two non-overlapping sets. Train on one set, and measure performance on the other. Pick the model that does well on the data that you *didn't* train on.



Supervised models

Model	When to use it?
KNN	➢ Little / no training time, large prediction time ➢ Given small / medium dataset size
Linear / polynomial regression	➢ Linear / polynomial relationship between input and output ➢ Small training time ➢ Given small / medium dataset size
Logistic regression, SVM, decision tree	➢ Categorical output ➢ Given small / medium training time
Neural network	➢ Large training time, large computer ➢ Given large dataset size

Note / life tip

- **Don’t re-implement it yourself!**
  - Unless you are doing research on the method itself, you are trying to learn how it works, or you are coding in an obscure language where it isn’t already implemented
  - The already implemented versions are widely used and tested

Note / life tip

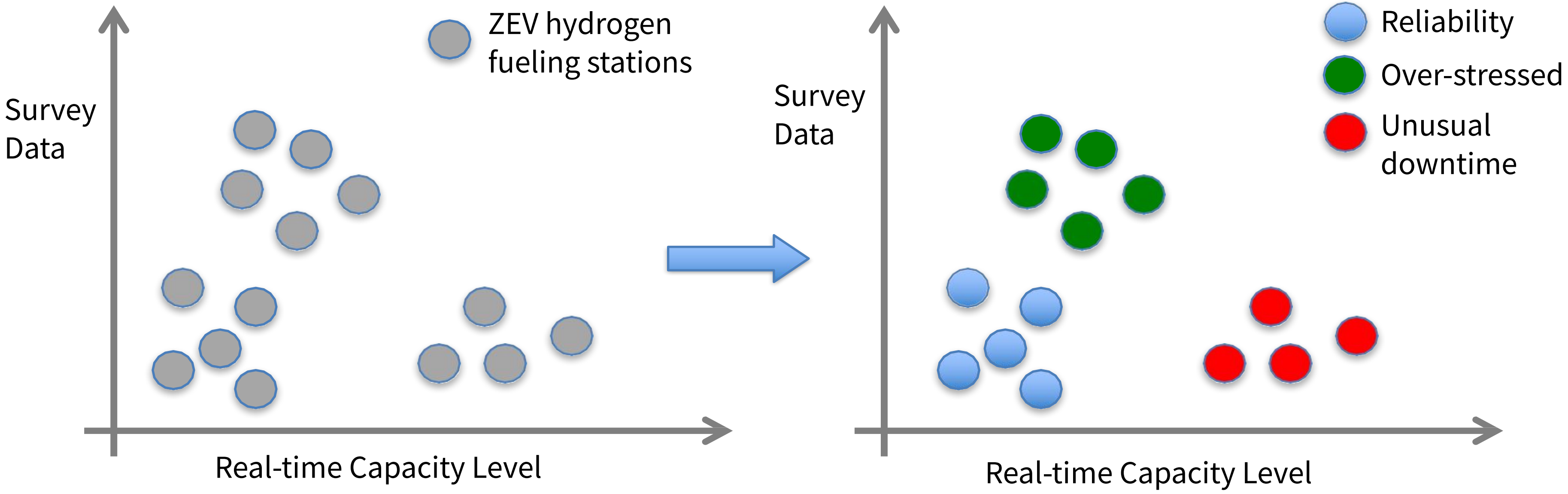
- **Don’t re-implement it yourself!**
  - Unless you are doing research on the method itself, you are trying to learn how it works, or you are coding in an obscure language where it isn’t already implemented
  - The already implemented versions are widely used and tested
- **Use these common tools:**
  - Scikit-learn has most supervised and unsupervised methods you might need
  - If you want to build a custom neural network, try using Pytorch or Tensorflow
  - There are many task-specific libraries

Agenda

- What is machine learning?
- Supervised learning
- **Unsupervised learning**
- Reinforcement learning
- Generative models

Unsupervised Learning

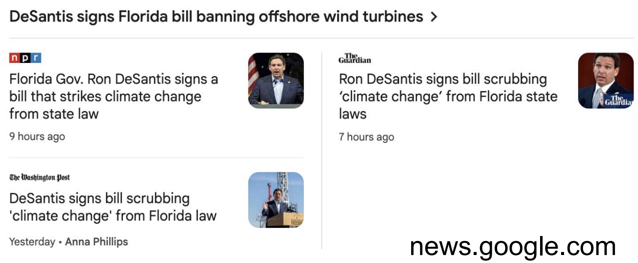
Finding patterns in unlabeled data



Unsupervised Temporal Clustering to Monitor the Performance of Alternative Fueling Infrastructure. Ramea 2020.

Unsupervised Learning

Finding patterns in unlabeled data



news.google.com

Examples:

- ➢ Finding clusters
  - ○ Customer segmentation (group customers so you can target advertising)
  - ○ Finding user accounts that are all suspiciously similar
  - ○ Group search results (or news / trending topics)
- ➢ Topic modeling (LDA)
- ➢ Figure out important features to use for supervised learning
- ➢ Learn vector representations for words / documents

Clustering

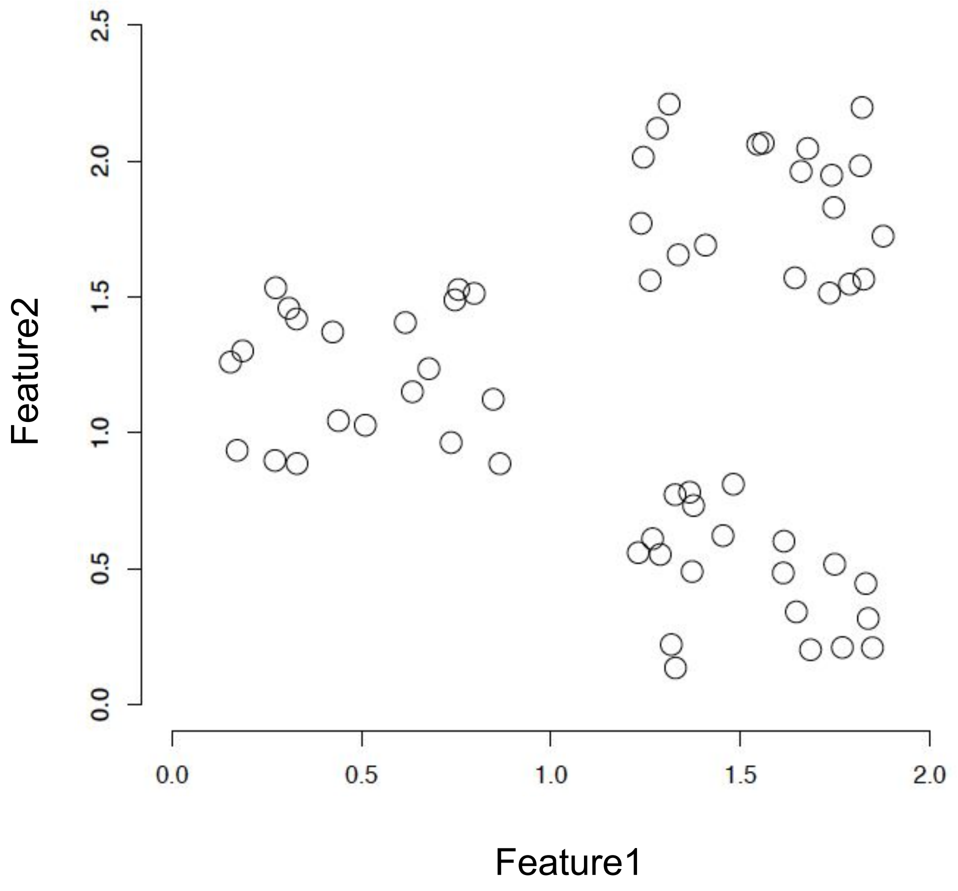
1. Extract features from raw data

Raw Data Item	Feature 1	Feature 2
Apple1	0.4	0.2
Apple2	0.5	0.1
Banana1	1.3	2.1
. . .	. . .	. . .

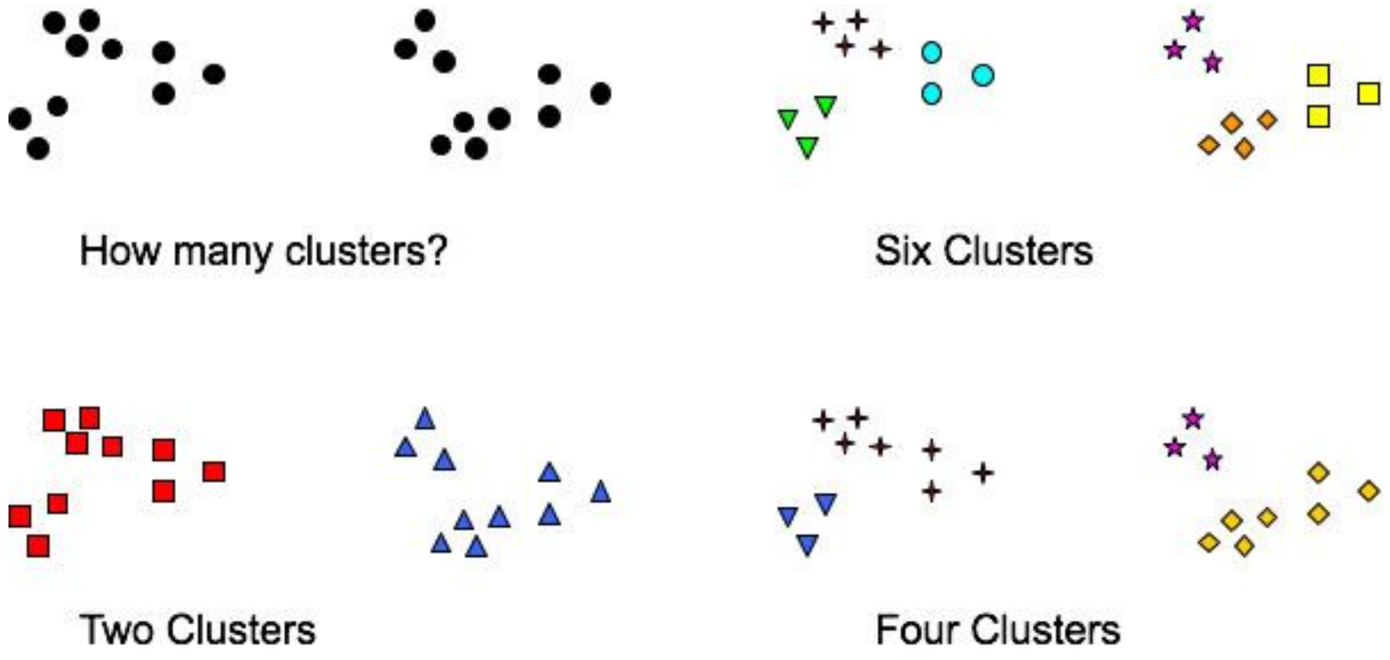
Clustering

1. Extract features from raw data
2. **Find natural groupings**

Raw Data Item	**Feature 1**	**Feature 2**
Apple1	0.4	0.2
Apple2	0.5	0.1
Banana1	1.3	2.1
. . .	. . .	. . .



Clusters are ambiguous

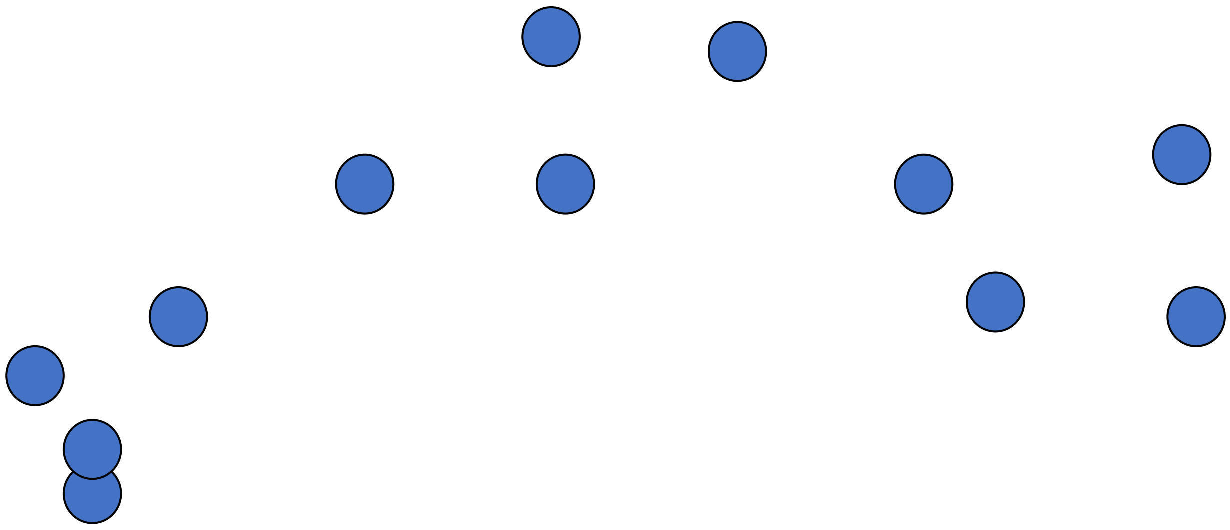


http://jcsites.juniata.edu/faculty/rhodes/ml/clusterAn.htm

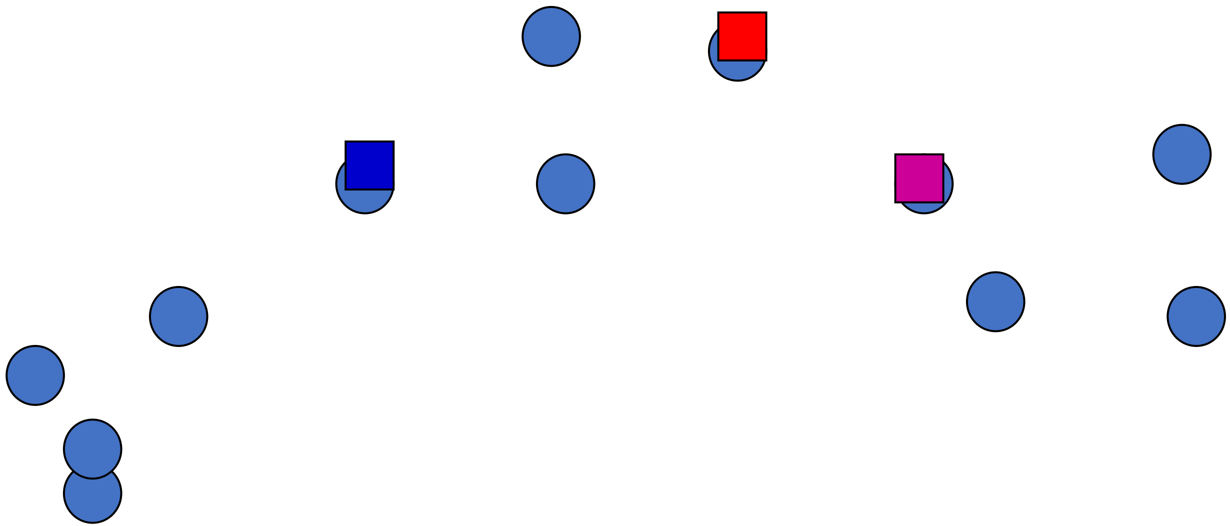
K-means

- Most well-known popular clustering algorithm
- Usually a baseline
- The algorithm:
  - Iterate until the clusters stop changing:
    - Assign / cluster each example to the closest center
    - Recalculate the centers as the mean of the points in their cluster

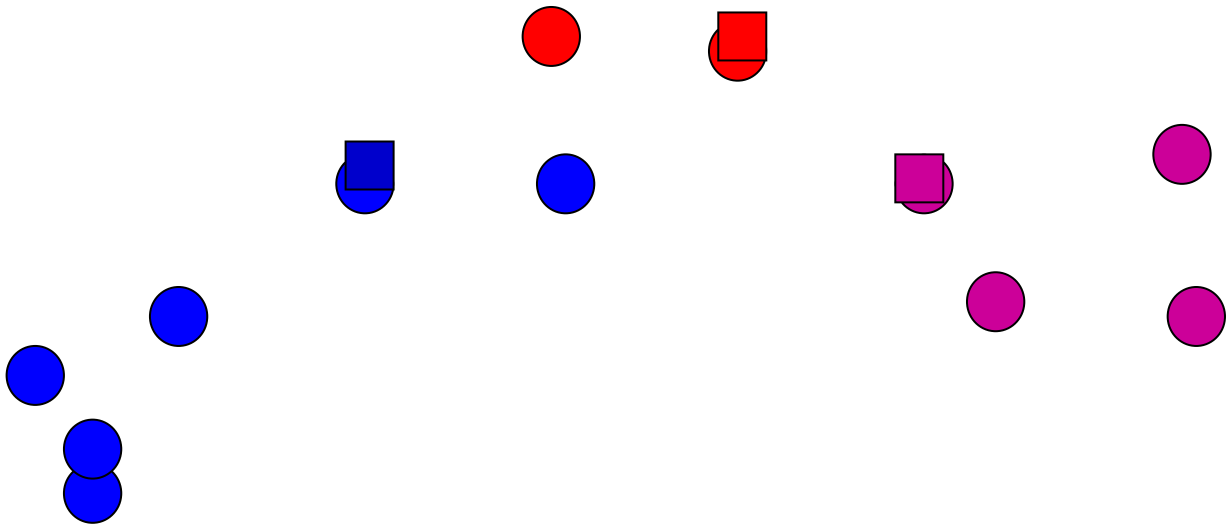
K-means example



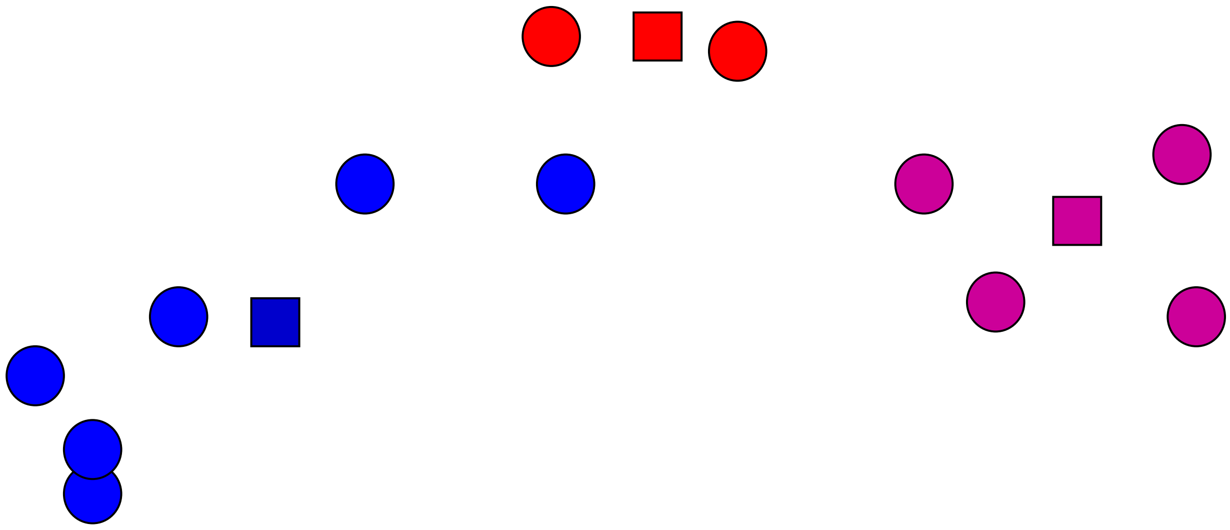
K-means example: initialize centers randomly



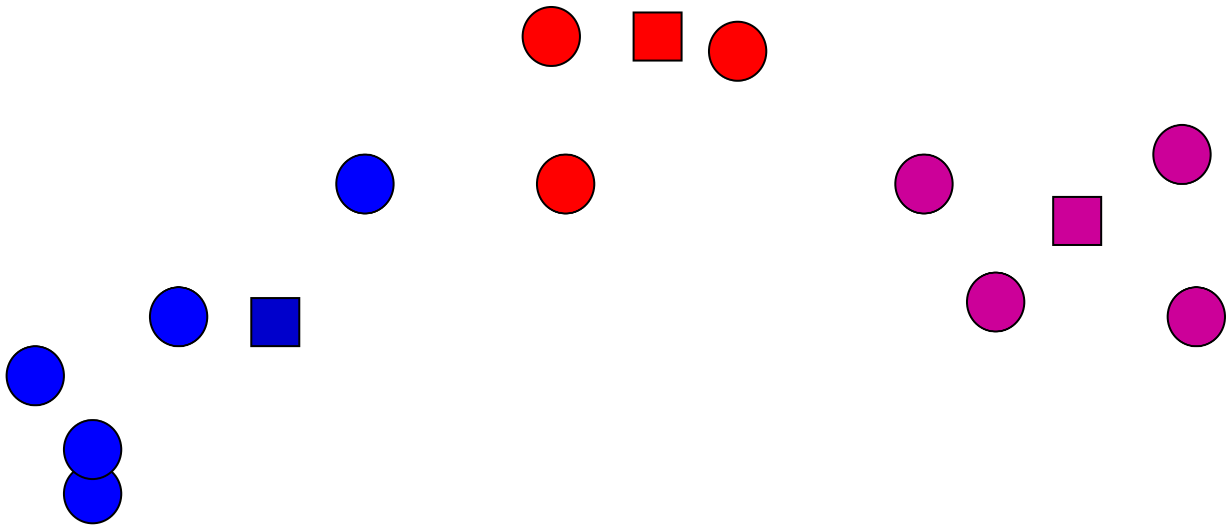
K-means example: assign points to nearest center



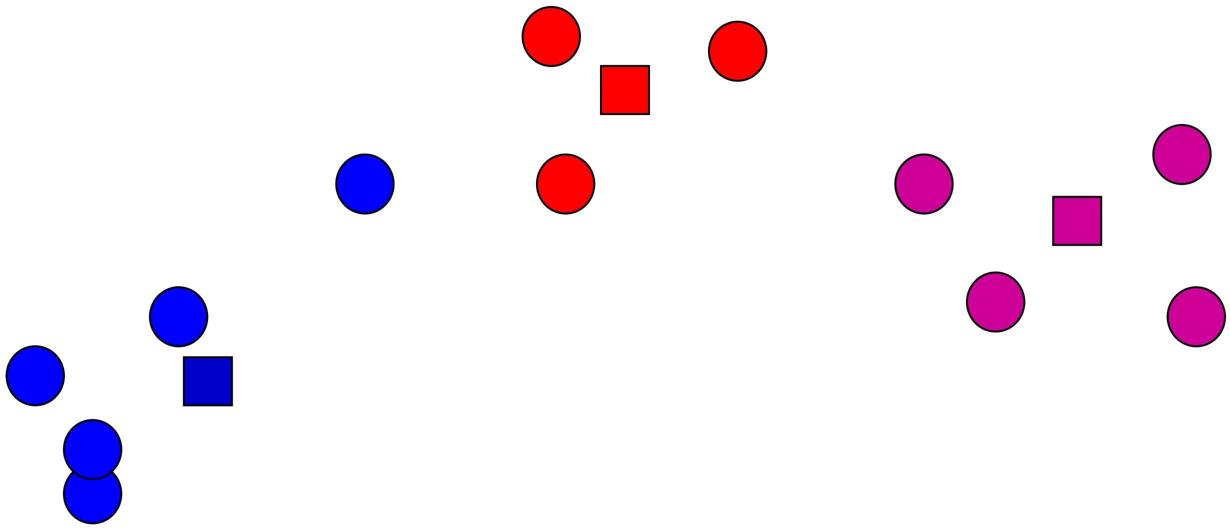
K-means example: recalculate centers



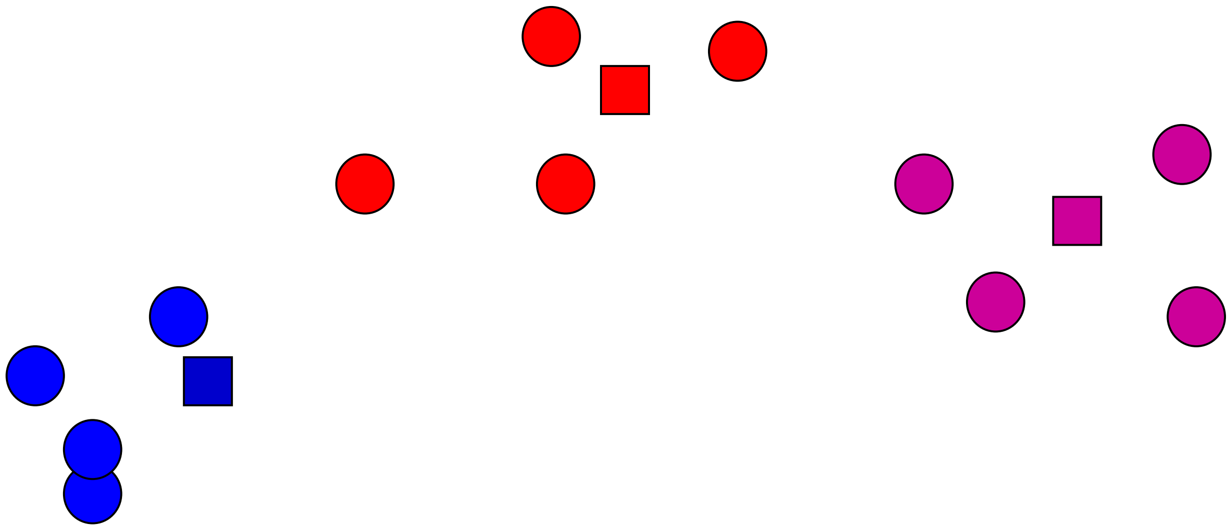
K-means example: assign points to nearest center



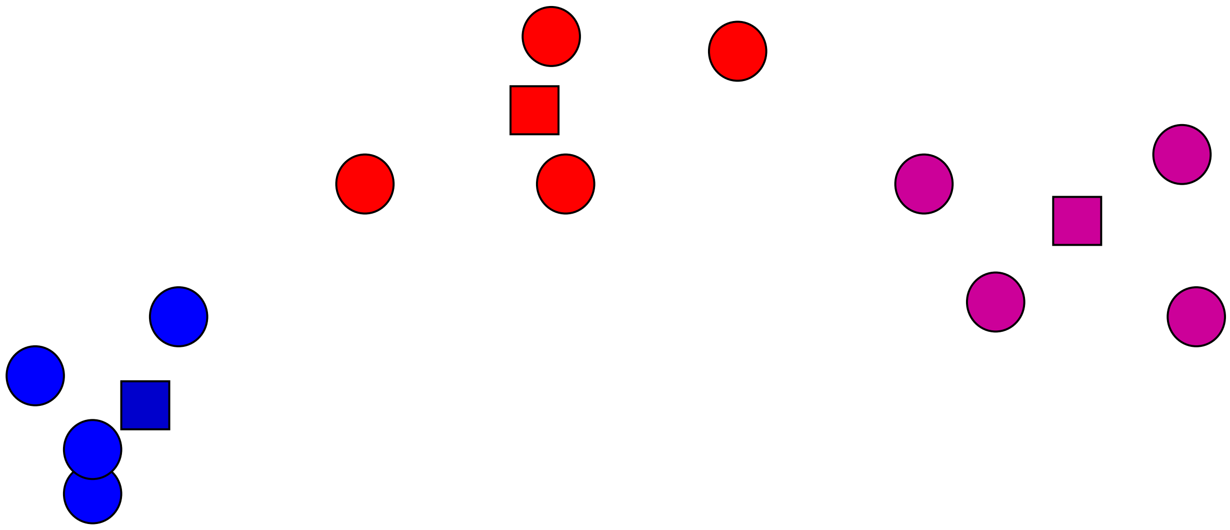
K-means example: recalculate centers



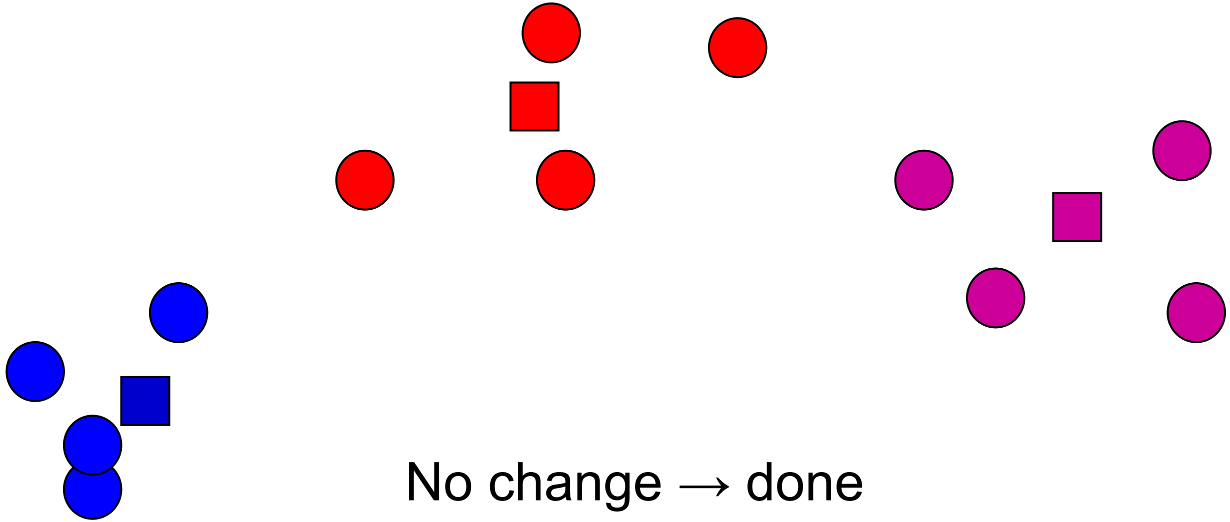
K-means example: assign points to nearest center



K-means example: recalculate centers



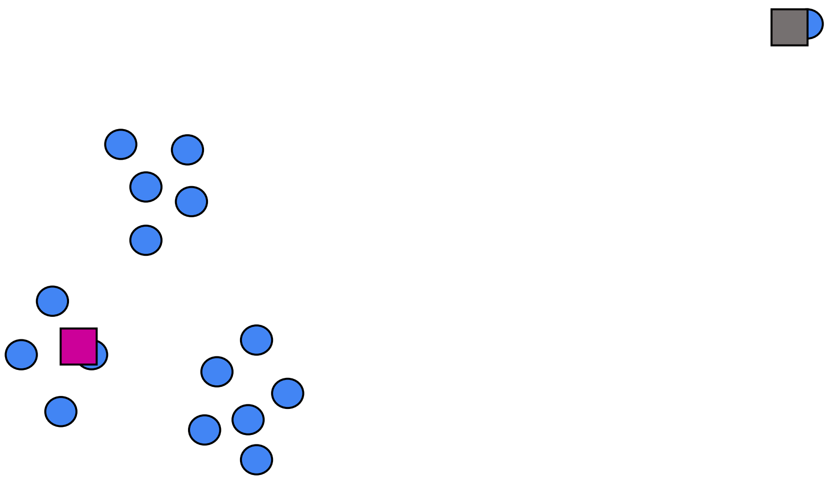
K-means example: assign points to nearest center



No change → done

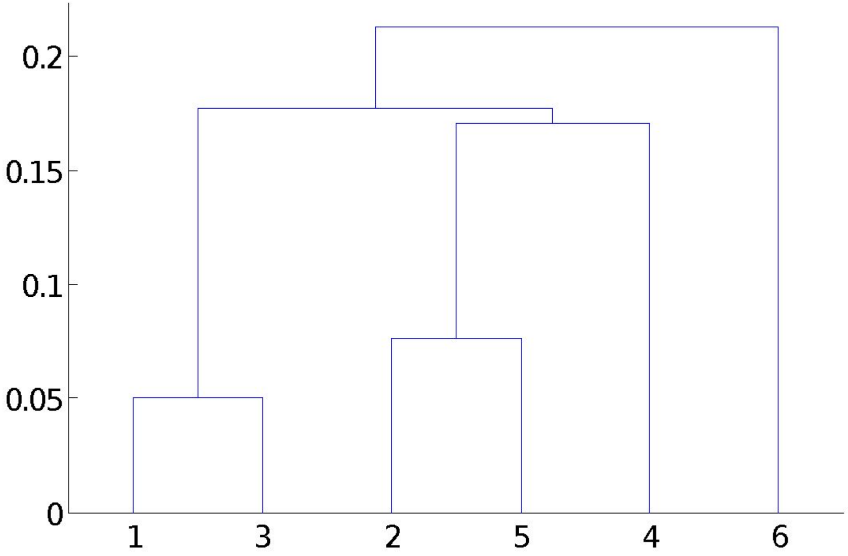
A Problem with K-Means: **Outliers**

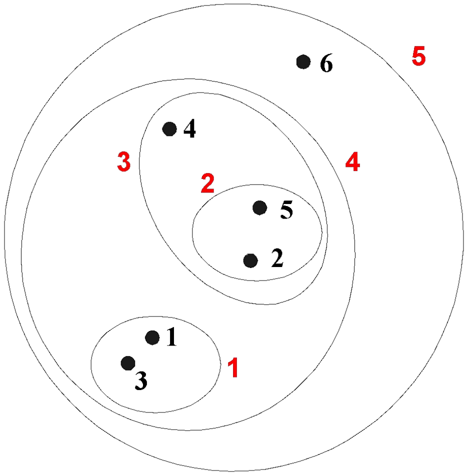
- Centroid has to move all the way to the outlier
- Each outlier takes up an entire cluster



Hierarchical Clustering

- ➢ Produces a set of nested clusters organized as a hierarchical tree
- ➢ Can be visualized as a **dendrogram**
  - ○ A tree like diagram that records the sequences of merges or splits





Clustering in Scikit-Learn

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters or distance threshold	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters or distance threshold, linkage type, distance	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
OPTICS	minimum cluster membership	Very large `n_samples`, large `n_clusters`	Non-flat geometry, uneven cluster sizes, variable cluster density	Distances between points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large `n_clusters` and `n_samples`	Large dataset, outlier removal, data reduction.	Euclidean distance between points

https://scikit-learn.org/stable/modules/clustering.html

Example: climate policy documents

https://www.climatechange.ai/papers/neurips2022/59

Given: Many companies' climate policy documents

Want to know: What is in these documents? Understand vague general categories

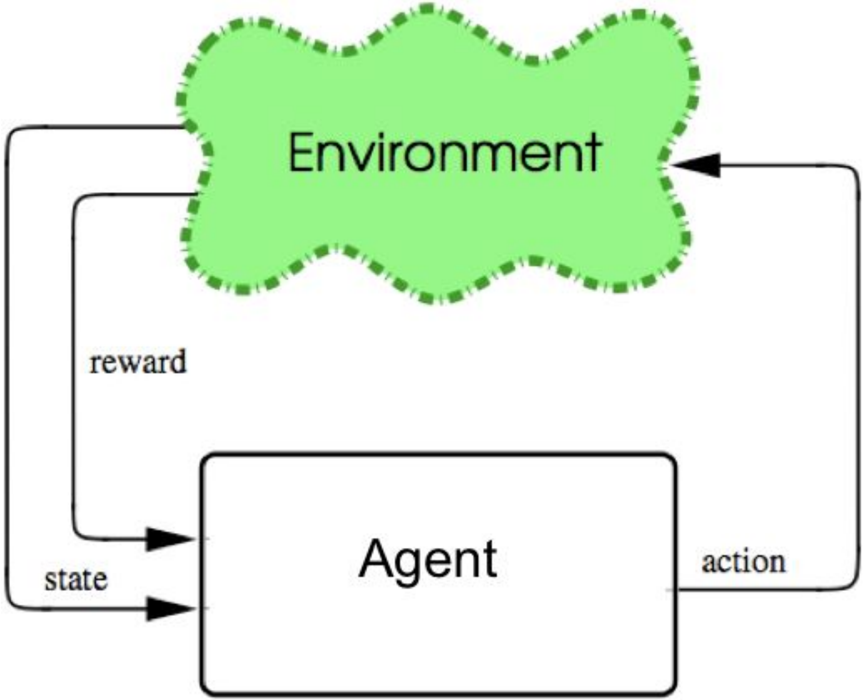
It's not labeled. **Which unsupervised algorithm might work best?**

A. Flat clustering (KMeans)
B. Hierarchical clustering
C. Topic modeling (LDA)

Agenda

- What is machine learning?
- Supervised learning
- Unsupervised learning
- **Reinforcement learning**
- Generative models

Reinforcement Learning



$$s_t \xrightarrow{a_t} \underset{r_{t+1}}{\overset{\uparrow}{}} s_{t+1} \xrightarrow{a_{t+1}} s_{t+2} \xrightarrow{a_{t+2}} s_{t+3} \xrightarrow{a_{t+3}} \ldots$$

Example rewards: PacMan

- One example:
  - 1 if you eat a pill
  - -10 if you get caught by a ghost
  - 2 if you eat a power pill or eat a ghost
  - 0 otherwise
- Another example:
  - -1 at every time step
  - 1,000,000 if you win the level

Exploration vs. Exploitation

- **Exploitation:** take good actions in each state already taken before to maximize reward
- **Exploration:** take a chance on actions that may have lower value in order to learn more, and maybe find true best action to later exploit

Need to balance the two!

Example: battery charging / discharging policy

https://www.climatechange.ai/papers/iclr2024/16

Given: batteries which can charge and discharge in a complex power grid

Want to know: an optimal charging / discharging policy

Which type(s) of learning could we use?

A. Supervised learning
B. Unsupervised learning
C. Reinforcement learning

Agenda

- What is machine learning?
- Supervised learning
- Unsupervised learning
- Reinforcement learning
- **Generative models**

Generative language models predict the next word.

Please turn your homework…

A. in
B. over
C. the
D. agriculture

Large language models (LLMs)

- **Huge** and trained on **large amounts of text** (the internet)
    - Large emissions to train (https://www.jmlr.org/papers/v24/23-0069.html)

- Can “hallucinate” facts
- Reproduces the (social) bias from its training set (the internet)

When do we use generative models like GPT?

For each: yes or no?

⁇ To make decisions which we use without a human in the loop

⁇ To look up facts and information which we verify afterwards

⁇ To write code which we run without verifying

⁇ To help write text or code faster while verifying everything it writes

Key take-aways

- Use ML for tasks which:
  - Ask the computer to find and use patterns in data (with errors)
  - Are more cost-effective with ML
  - Have appropriate data
- **Supervised** vs. **unsupervised** vs. **reinforcement** learning
  - vs. **generative models**
- **Categorical** vs. **continuous** data
  - Images: each pixel is 3 continuous features (RGB)
  - Text: each word is a categorical feature
- Most things can be done in a couple lines of code using Scikit-learn
  - Make use of their code examples