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
- Examples

Machine Learning:

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

- Andrew Ng

Types of learning

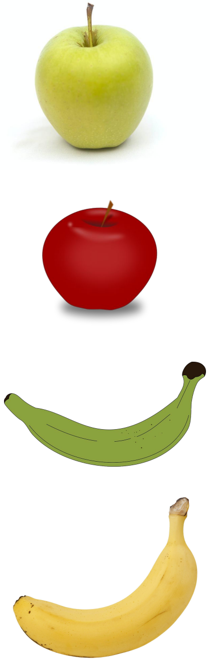
- **Supervised learning**
  Learning to predict or classify labels based on labeled input data

- **Unsupervised learning**
  Finding patterns in unlabeled data

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

Supervised vs. Unsupervised learning

Supervised



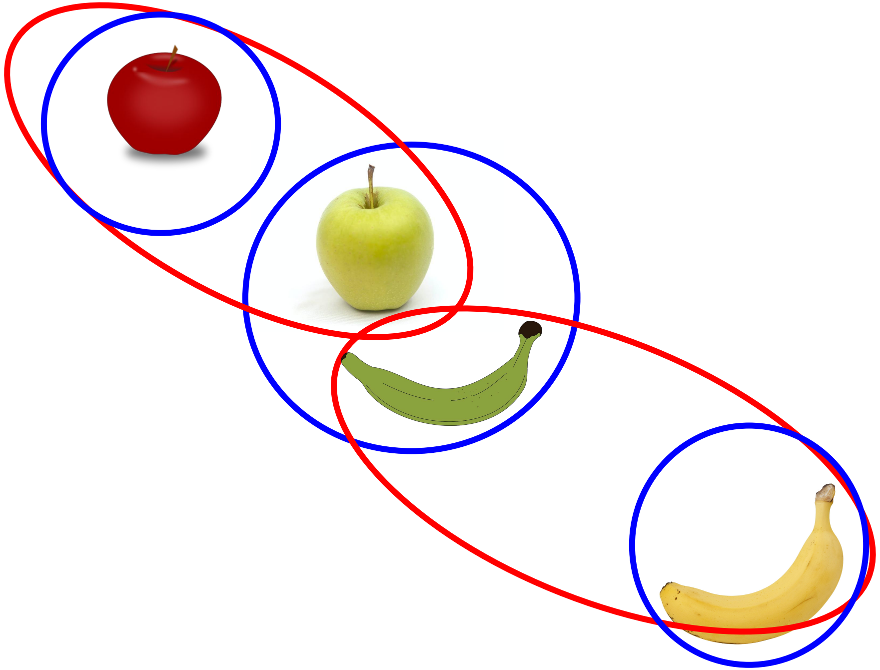
Apple

Apple

Banana

Banana

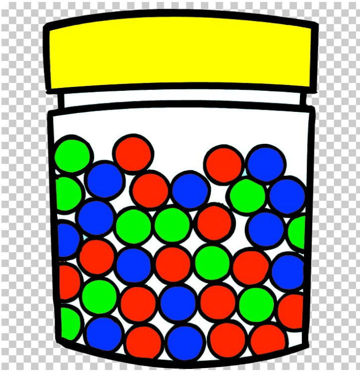
Unsupervised



Data Types

- **Continuous**

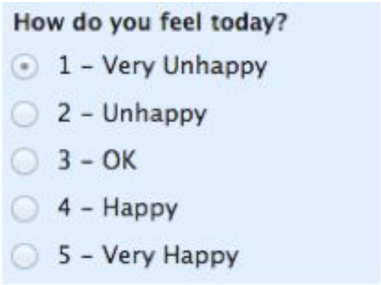
- **Discrete**



- **Categorical**

- **Binary**
  - Special case of categorical

- **Ordinal**



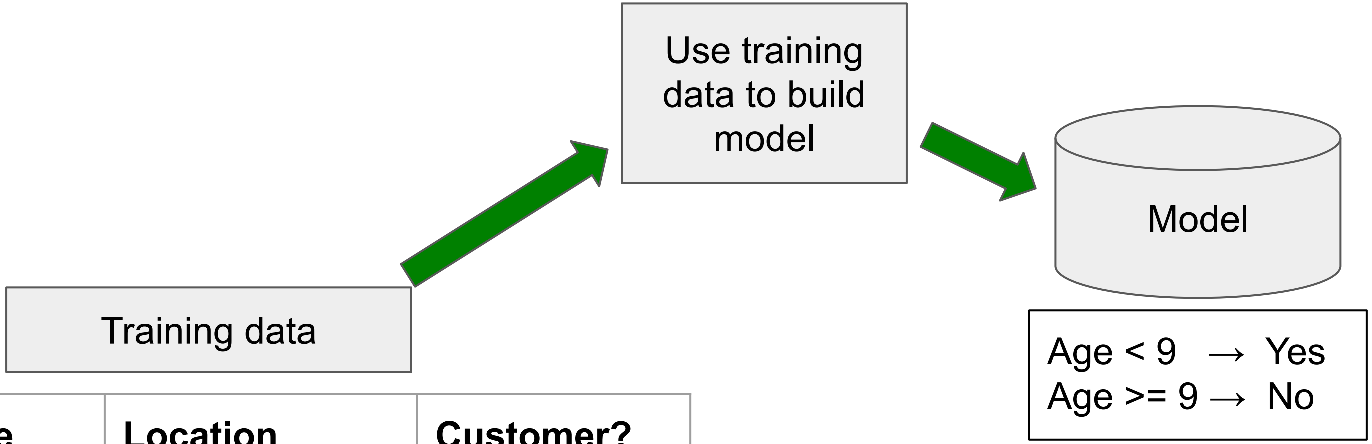
Data types you might use

- **Tabular**
  - Each item is a row in a table, and the columns are features
- **Time series**
  - Time / order of the data is part of the input
- **Graph / network**
  - Examples: social media friends graph, tweets / retweets
- **Images**
  - Each pixel is 3 continuous features (RGB)
- **Language / text**
  - Each word is a categorical feature

Agenda

- What is machine learning?
- **Supervised learning**
- Unsupervised learning
- Reinforcement learning
- Examples

Supervised learning: training



Person	**Age**	**Location**	**Customer?**
Cat	8	San Francisco	Yes
Elephant	13	Berlin	No
Dog	7	Delhi	Yes
Turtle	15	Bogotá	No

Supervised learning: test / prediction

Model

Age < 9 → Yes
Age >= 9 → No

Evaluate performance using test data

Person	**Age**	**Location**	**Customer?**
Penguin	6	Antarctica	Yes
Snake	15	Durban	No

Use on unseen real-world data

Person	**Age**	**Location**
Chicken	8	Addis Ababa

Customer? Yes

Supervised learning: categorical versus continuous labels

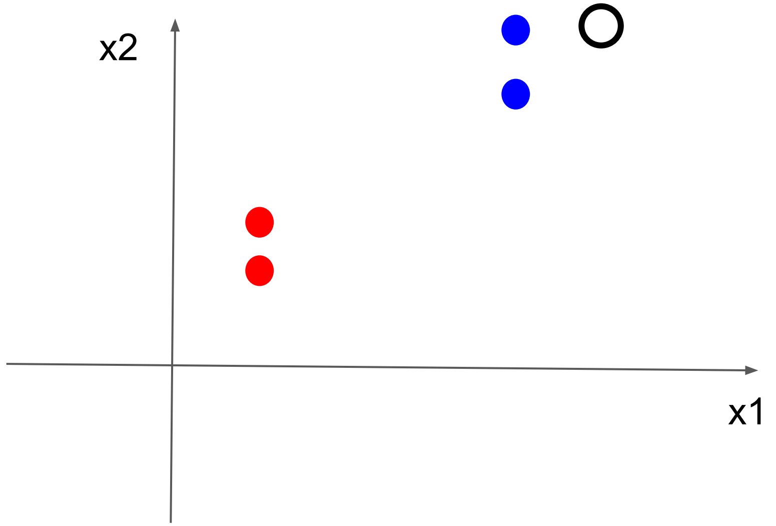
- Classification: **categorical labels**
  - Examples: pregnant or not, from which country, which type of road sign
- Regression: **continuous labels**
  - Examples: future stock price, life expectancy, distance to obstacle

What's the simplest imaginable working classifier?

- **Training set:** n instances, each with a feature vector and an output category
- Now, given another (unseen) instance, we want to determine its category

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

New instance:
(6,6)



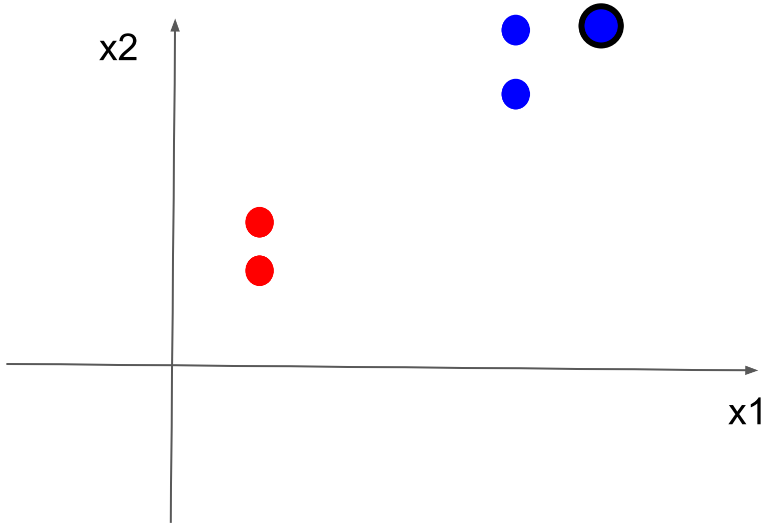
k-Nearest Neighbors Algorithm

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

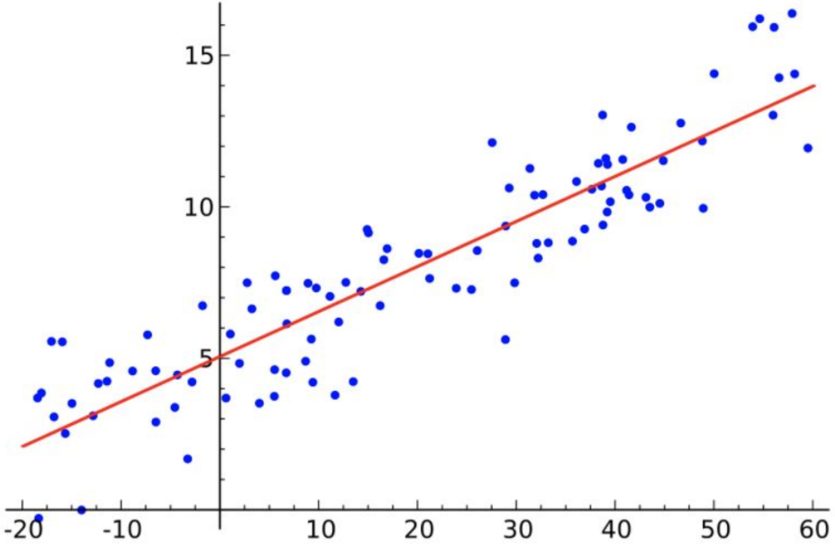


Linear regression

Given x ∈ $\mathbb{R}$ and y $\mathbb{R}$ find a linear function f: x → y

Performance measure: Least Squares

➔ Minimize mean square error between prediction and ground truth

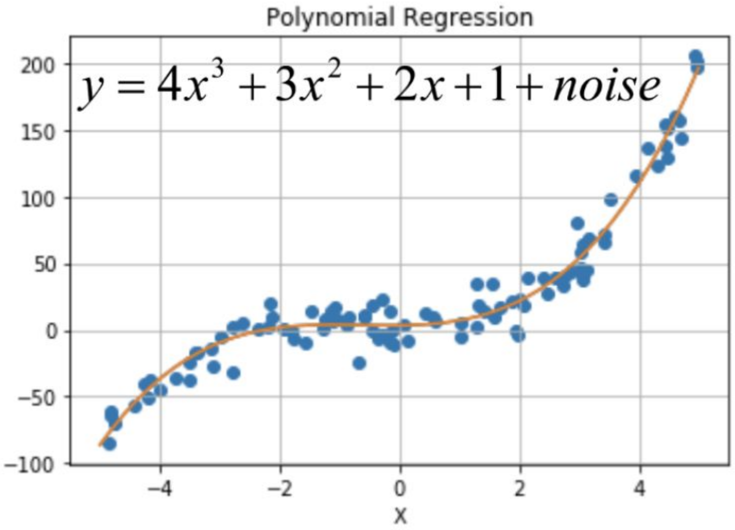


Polynomial fitting

Goal: find values for β in

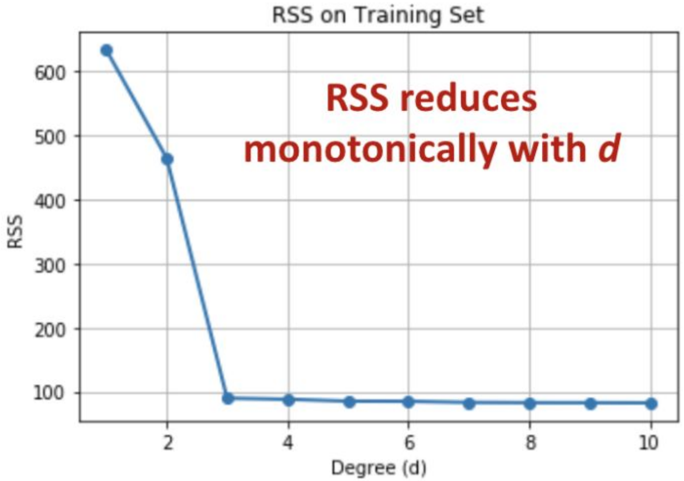
$$y = f(x) = \beta_d x^d \quad + \ldots \beta_2 x^2 \ + \beta_1 x \ + \beta_0$$

This turns into the same process as linear regression, if we know the value of d



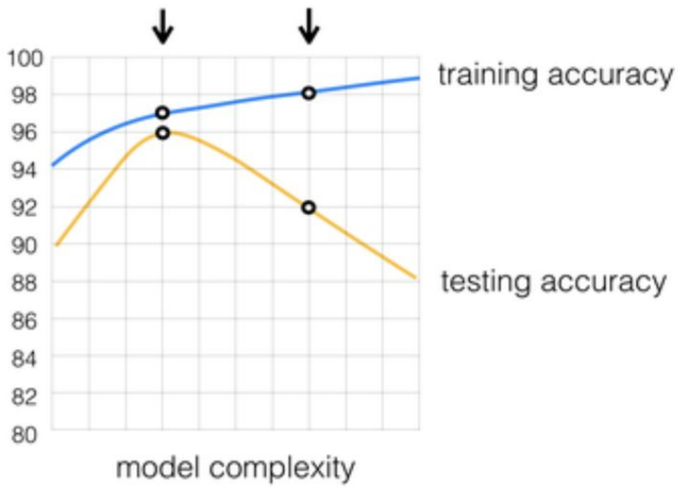
Polynomial fitting: what if you don't know d?

Compute RSS(**d**): squared error as a function of **d** on the training dataset



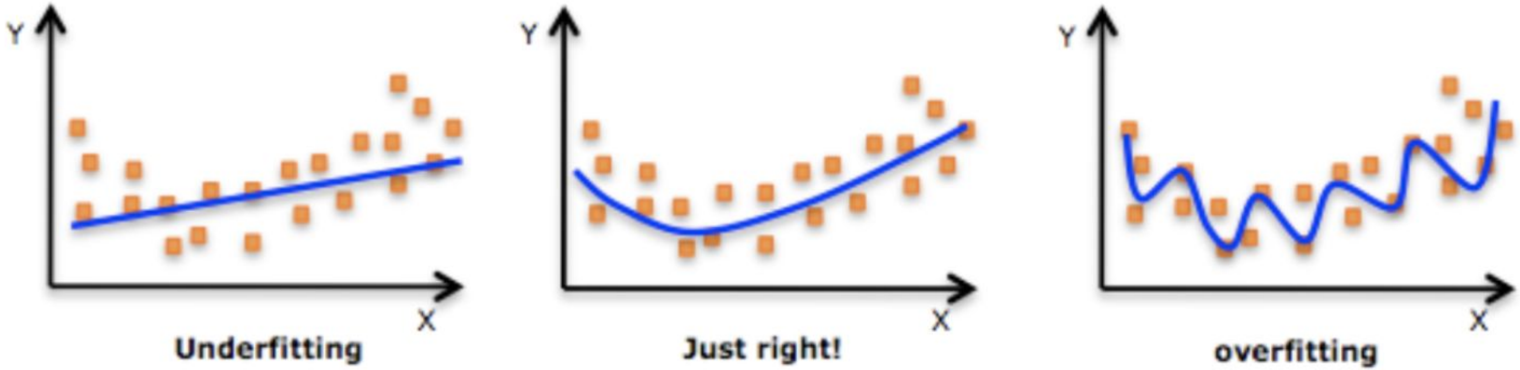
What is the problem with choosing the degree that has the lowest squared error?

Overfitting

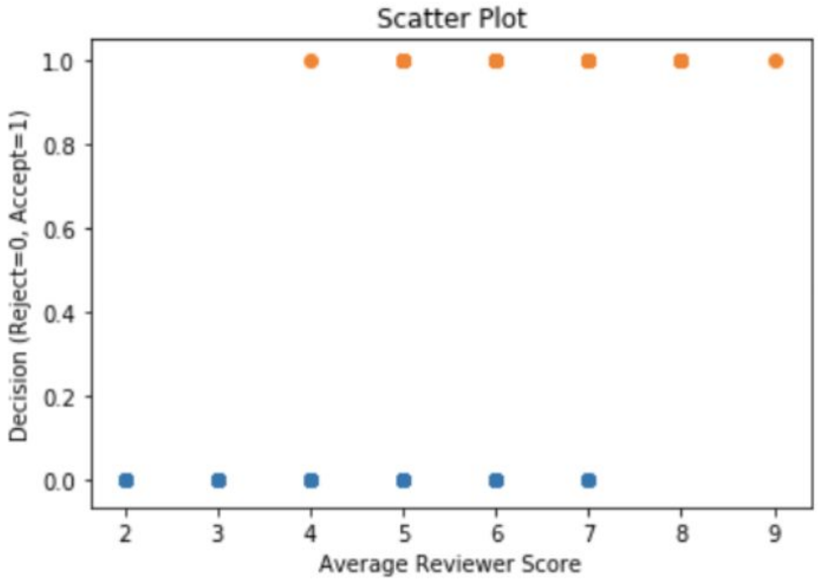


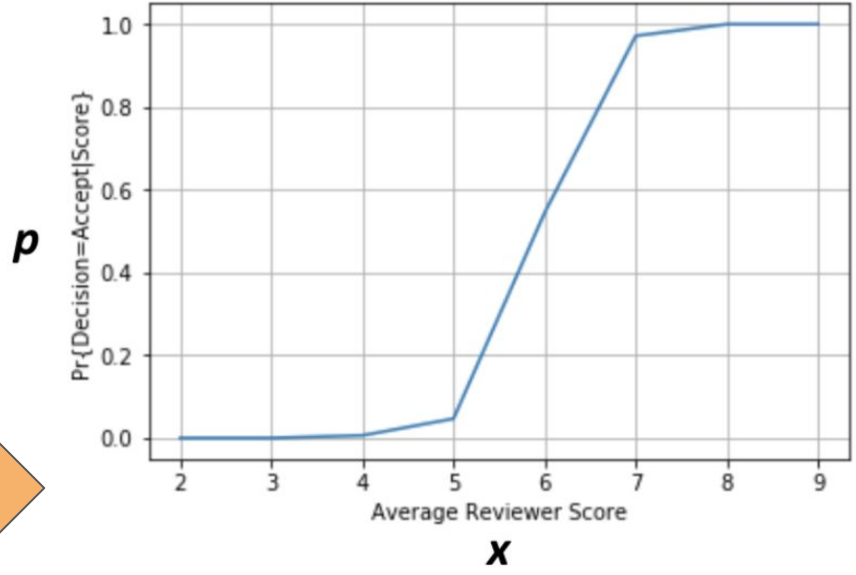
Solution: **Cross-Validation**

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



Classification: logistic regression



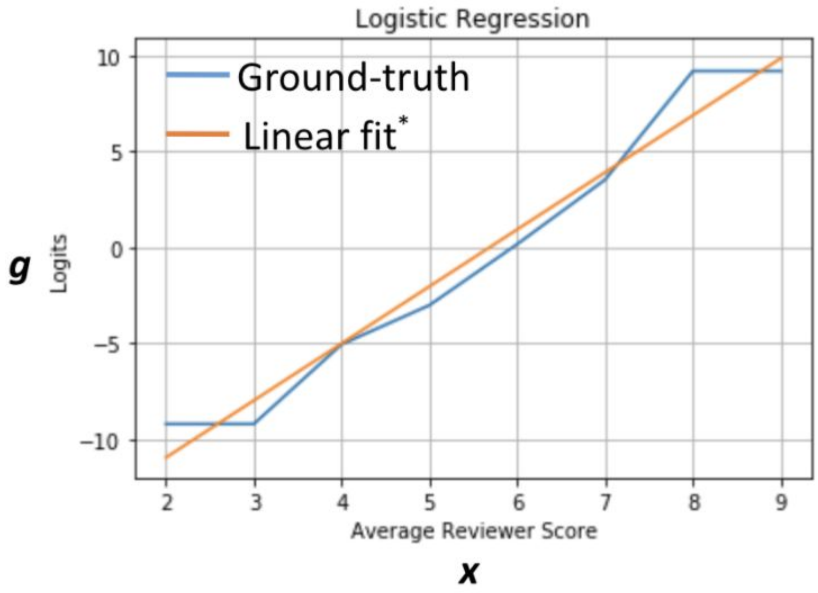


What's the problem with fitting a linear regression to the graph on the left?

Idea: p = probability of *accept* given score.
Now fit p as a function of x.
Why is this still not great?

Classification: logistic regression

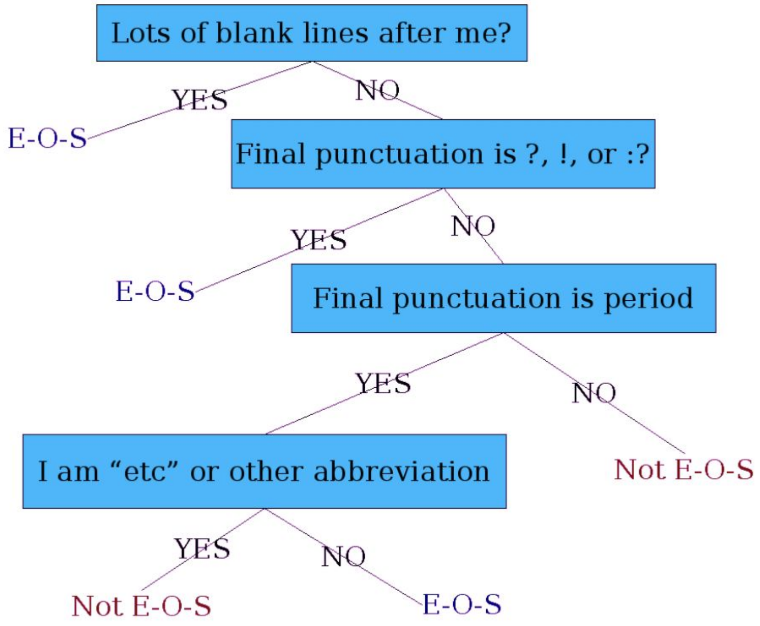
- Probabilities are in range [0,1]
- Map probability into range (-∞,∞) using $g = \log(\frac{p}{1-p})$
- Then do linear regression like before



Other supervised classifiers

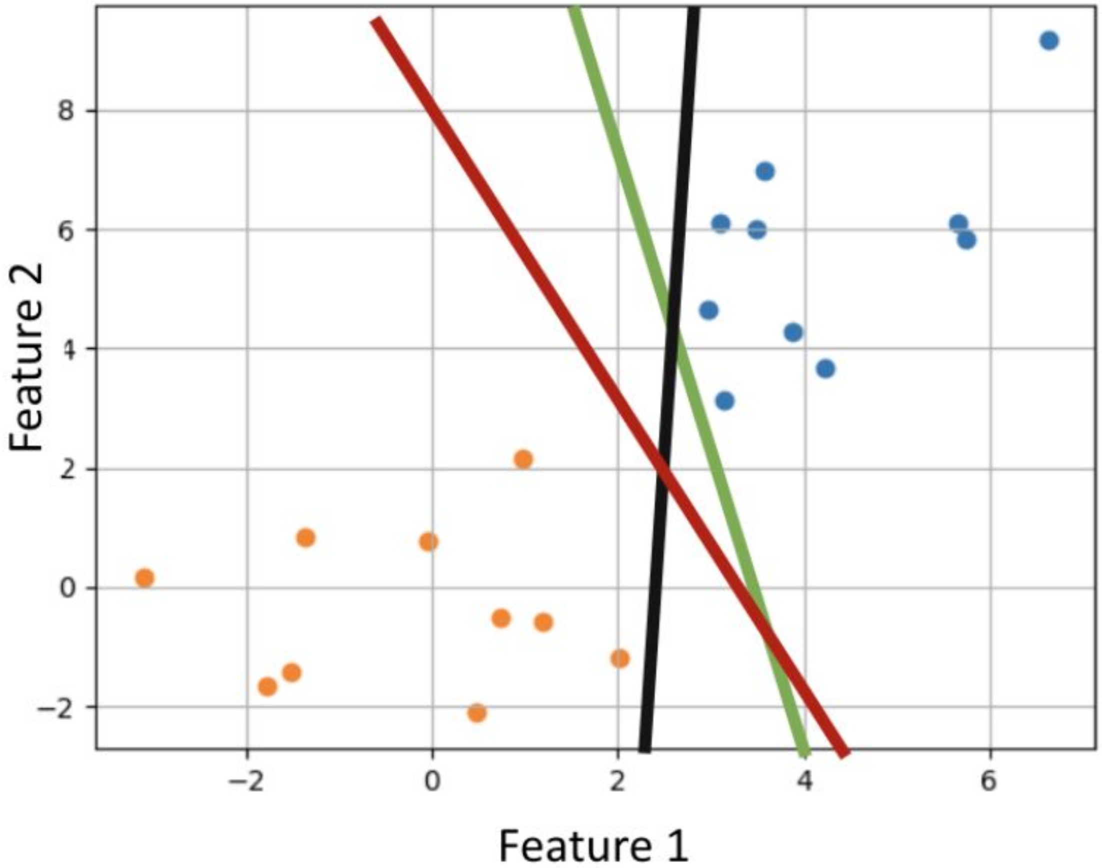
- **Decision tree**

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



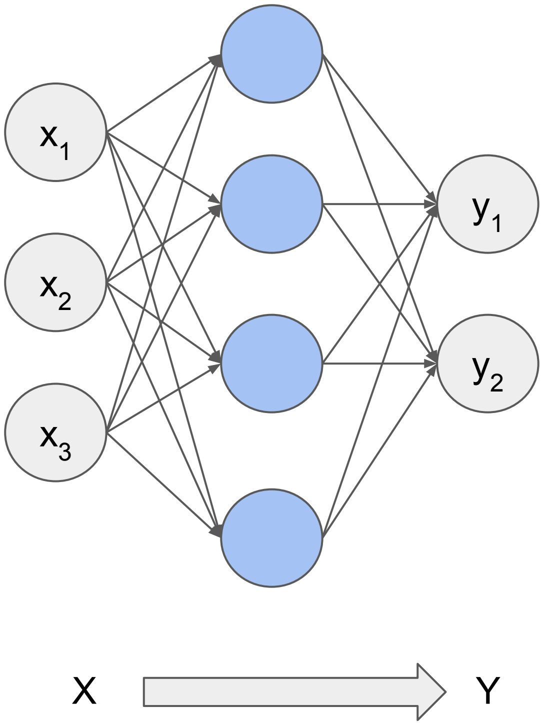
Other supervised classifiers

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



Other supervised classifiers

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



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
- Examples

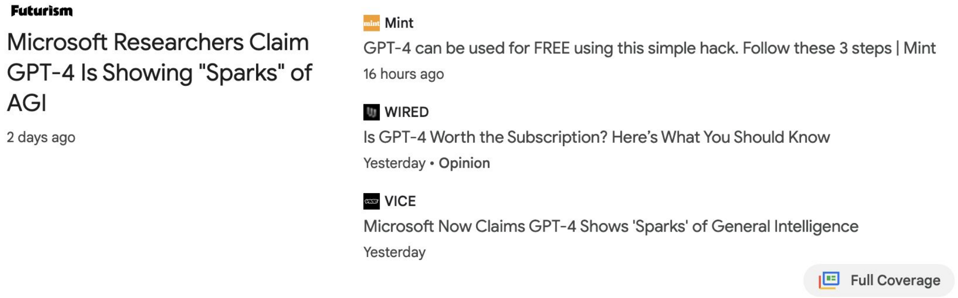
Unsupervised Learning

Learn clusters / groups without labels

Find patterns in unlabeled data

Applications:

- 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
- Language modeling
- TextRank (part of text summarization)



news.google.com

Clustering

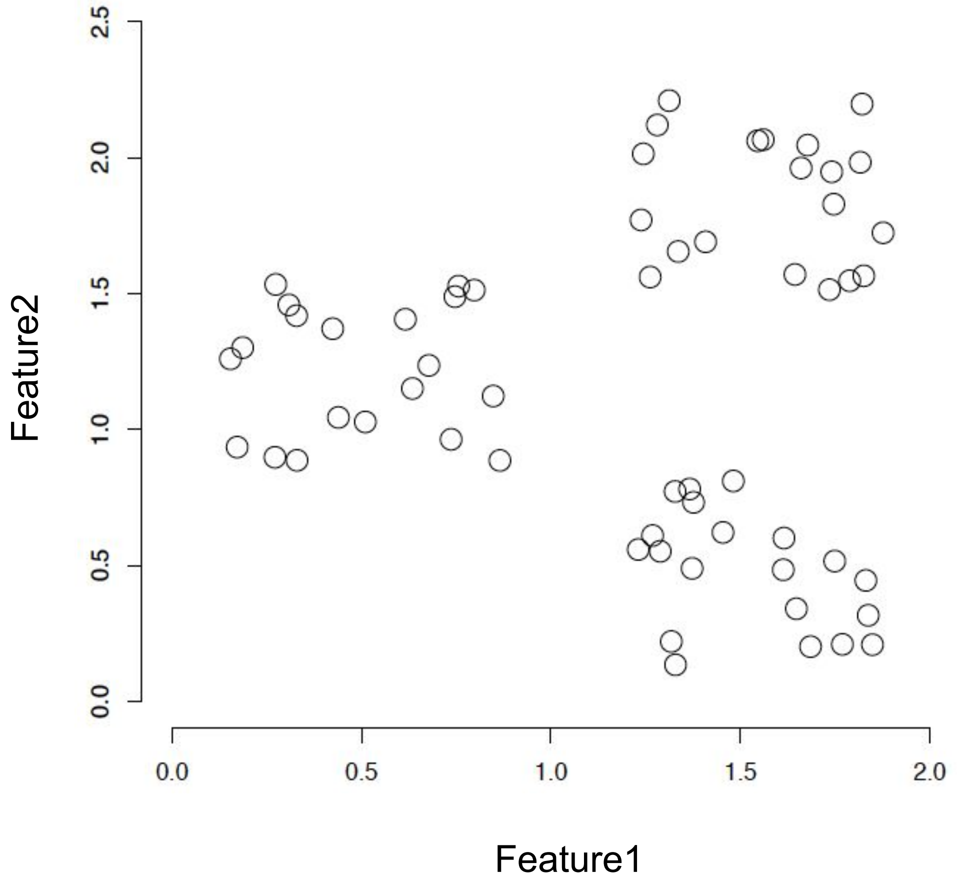
1. Extract features from raw data

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

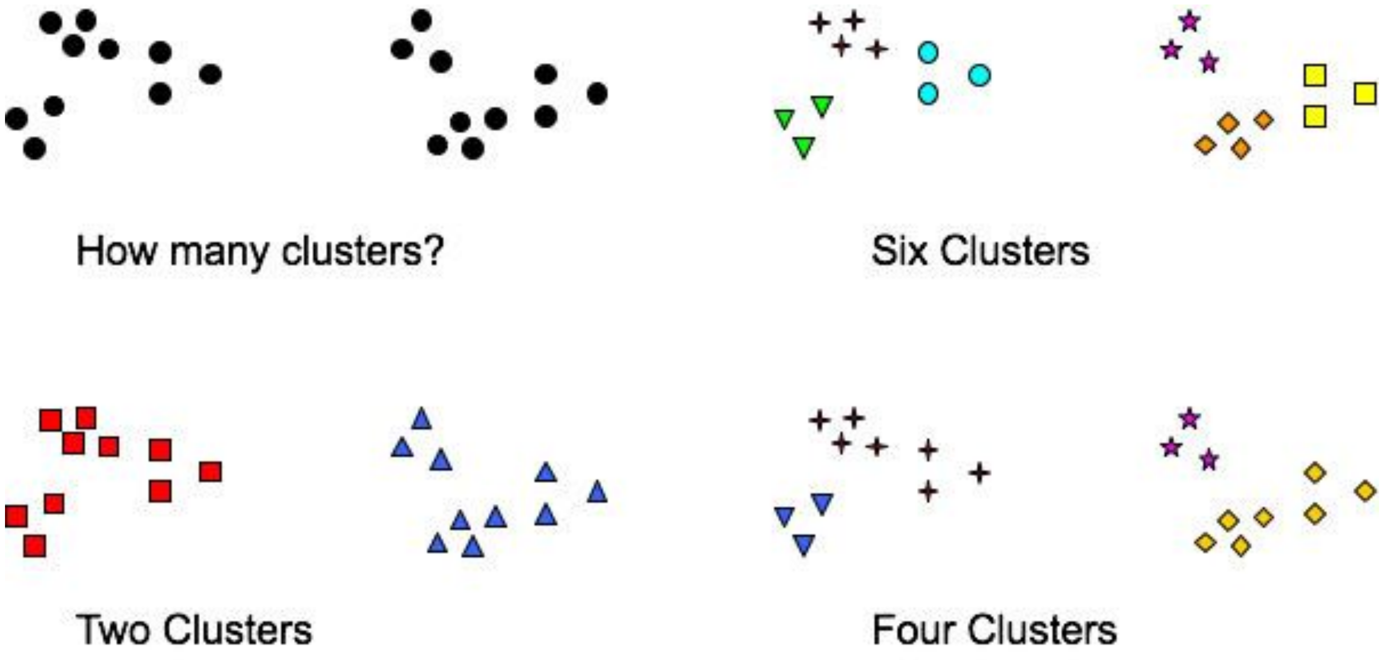
Clustering

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
. . .	. . .	. . .



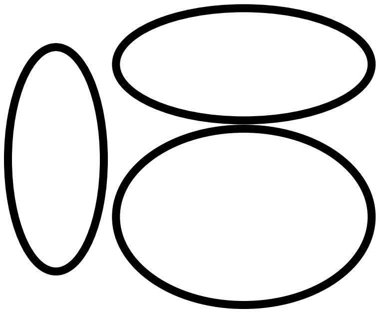
Clusters are ambiguous

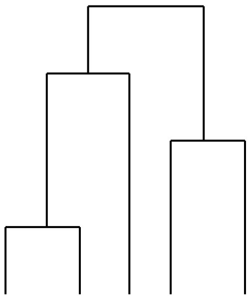


http://jcsites.juniata.edu/faculty/rhodes/ml/clusterAn.htm

Clustering: Flat vs. Hierarchical

- Flat
  - Usually start with a random clustering
  - Iteratively refine the clustering
- Hierarchical
  - Agglomerative (bottom - up)
  - Divisive (top - down)

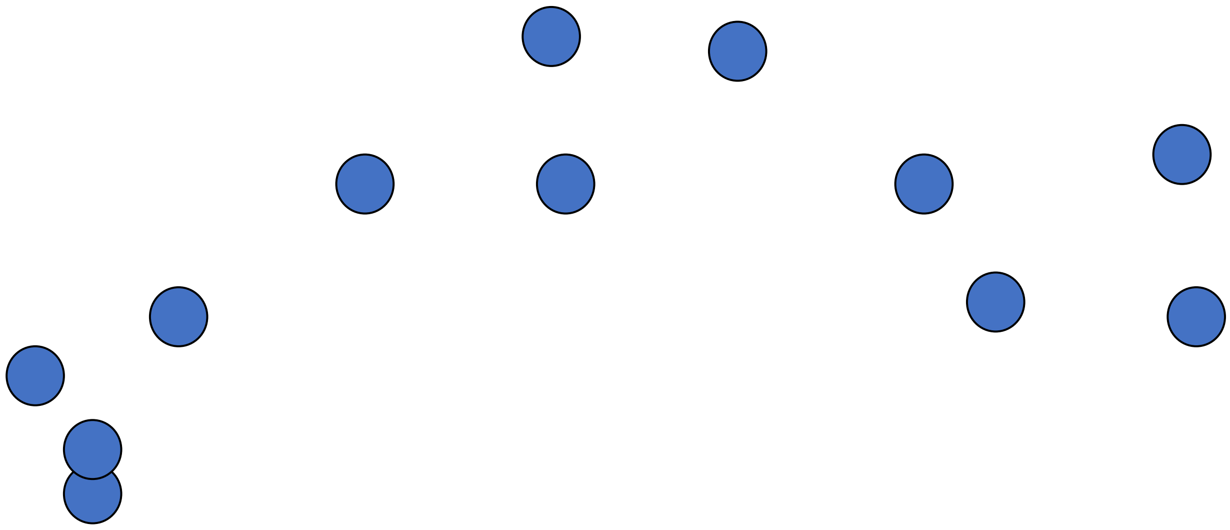




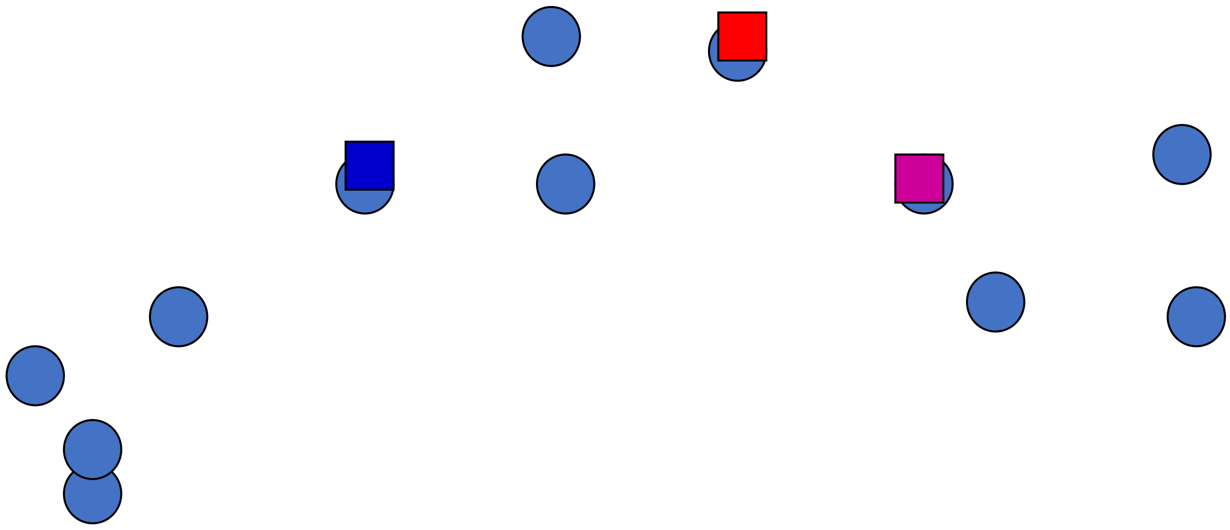
K-means

- 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

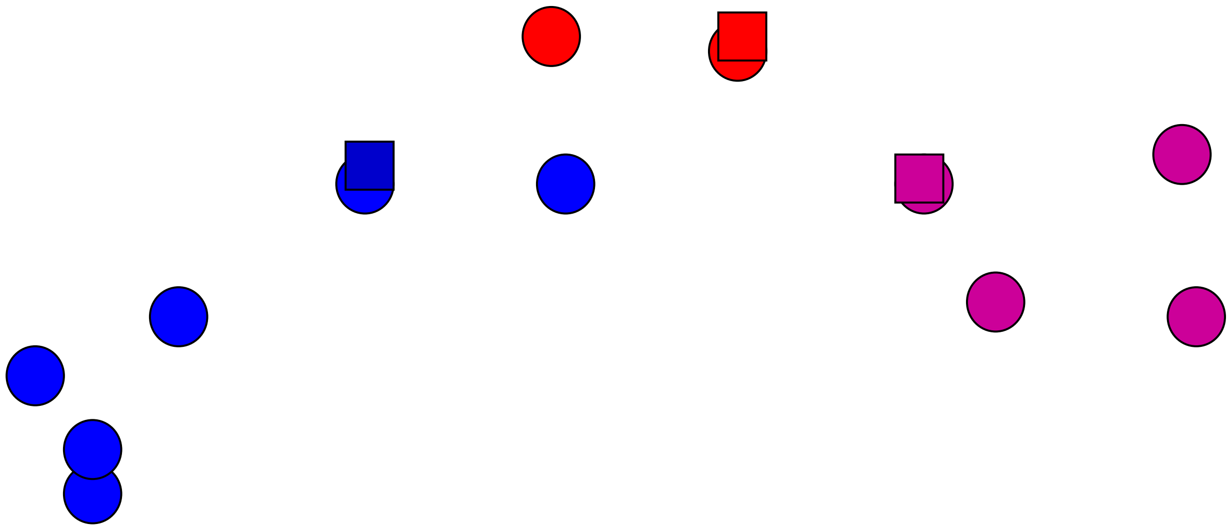
K-means example



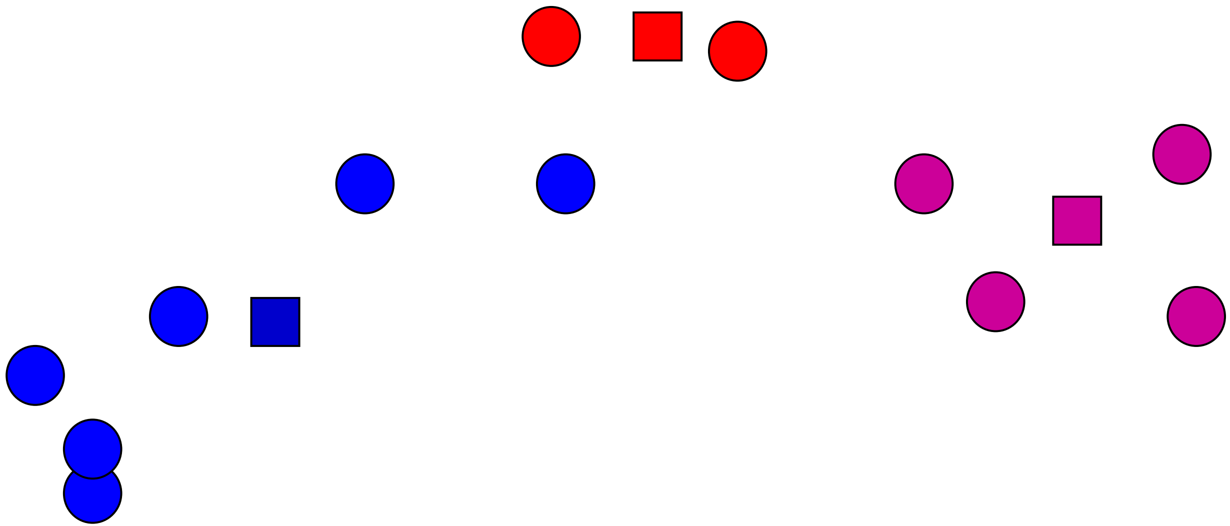
K-means example: initialize centers randomly



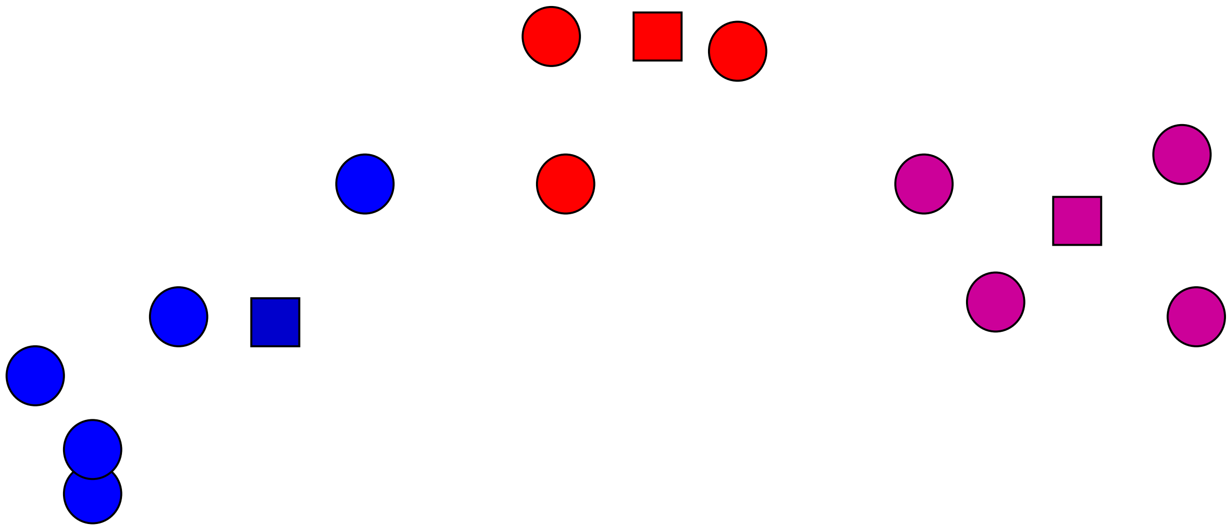
K-means example: assign points to nearest center



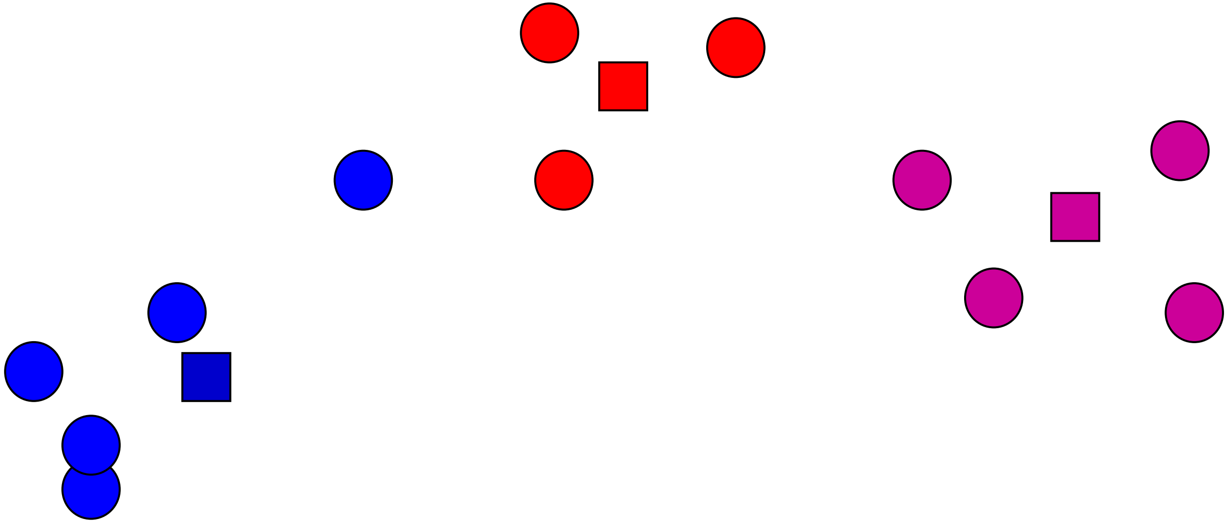
K-means example: recalculate centers



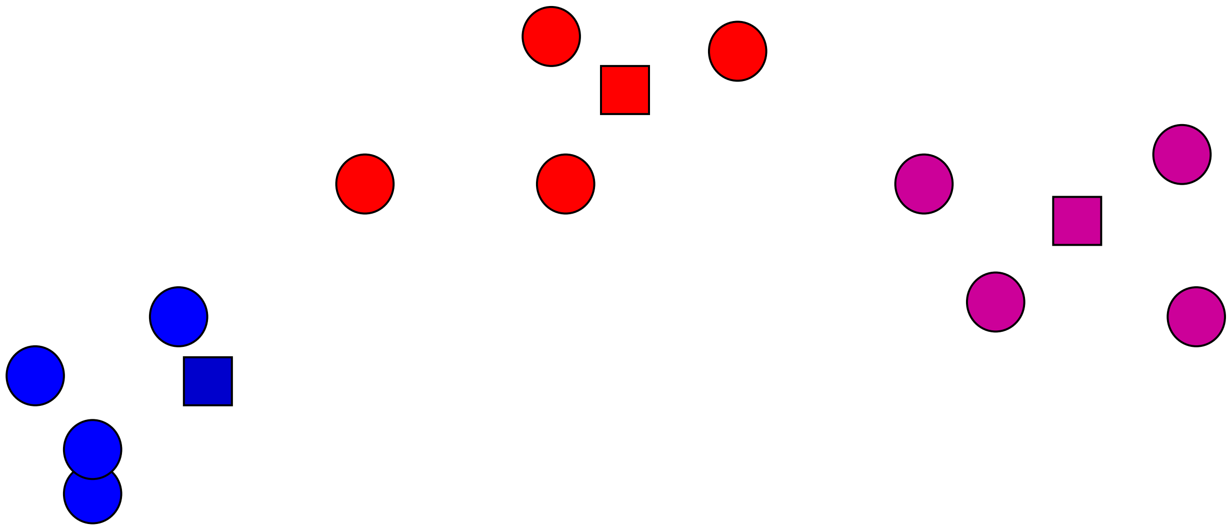
K-means example: assign points to nearest center



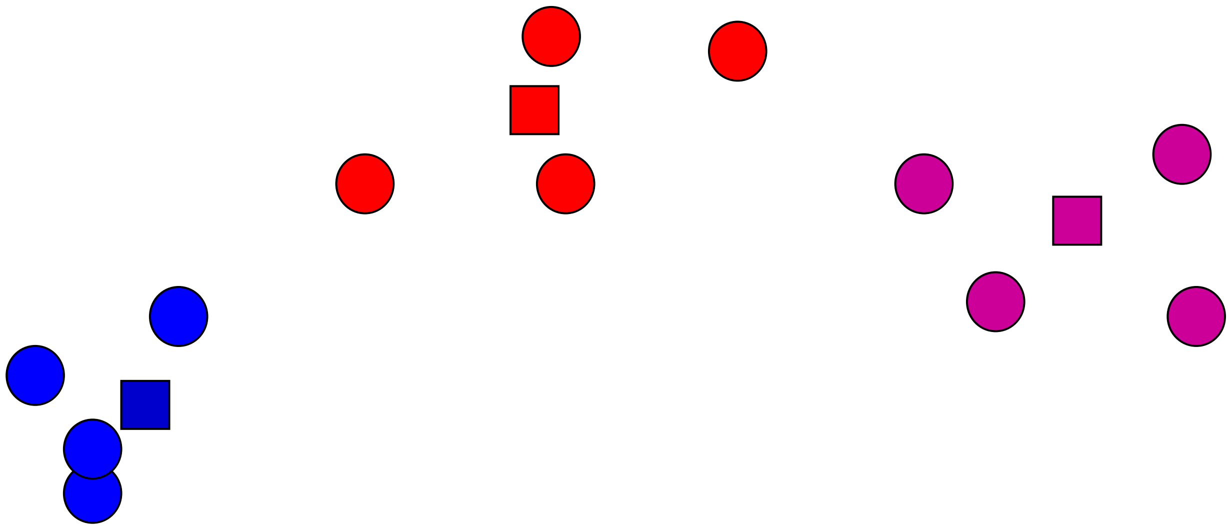
K-means example: recalculate centers



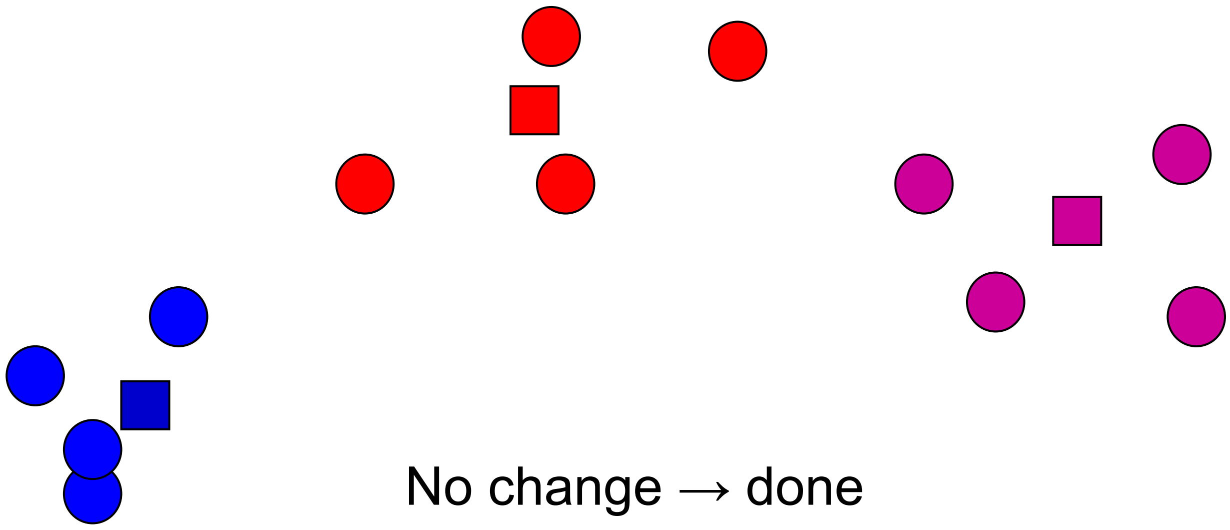
K-means example: assign points to nearest center



K-means example: recalculate centers



K-means example: assign points to nearest center



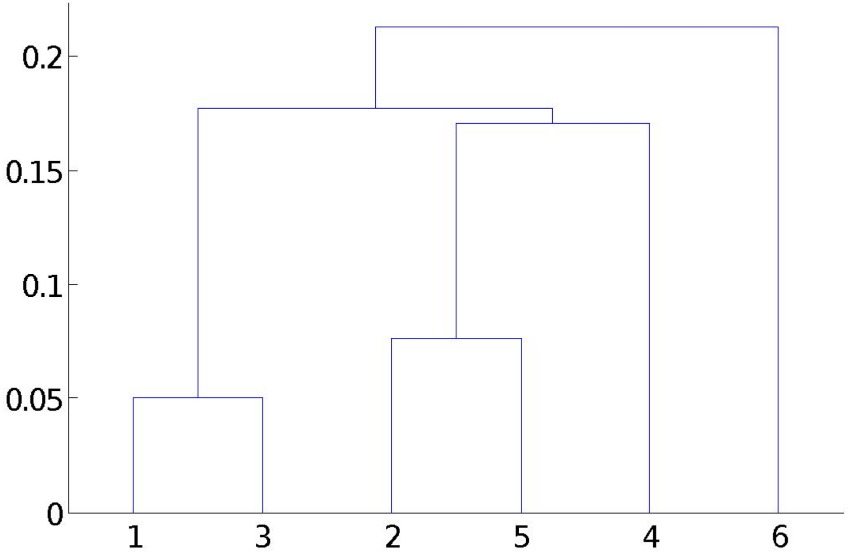
No change → done

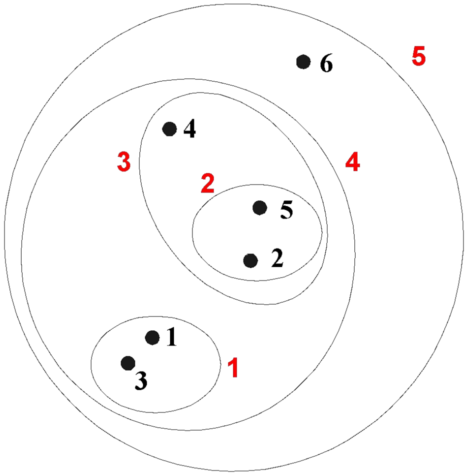
A Problem with K-Means: **Outliers**

- 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

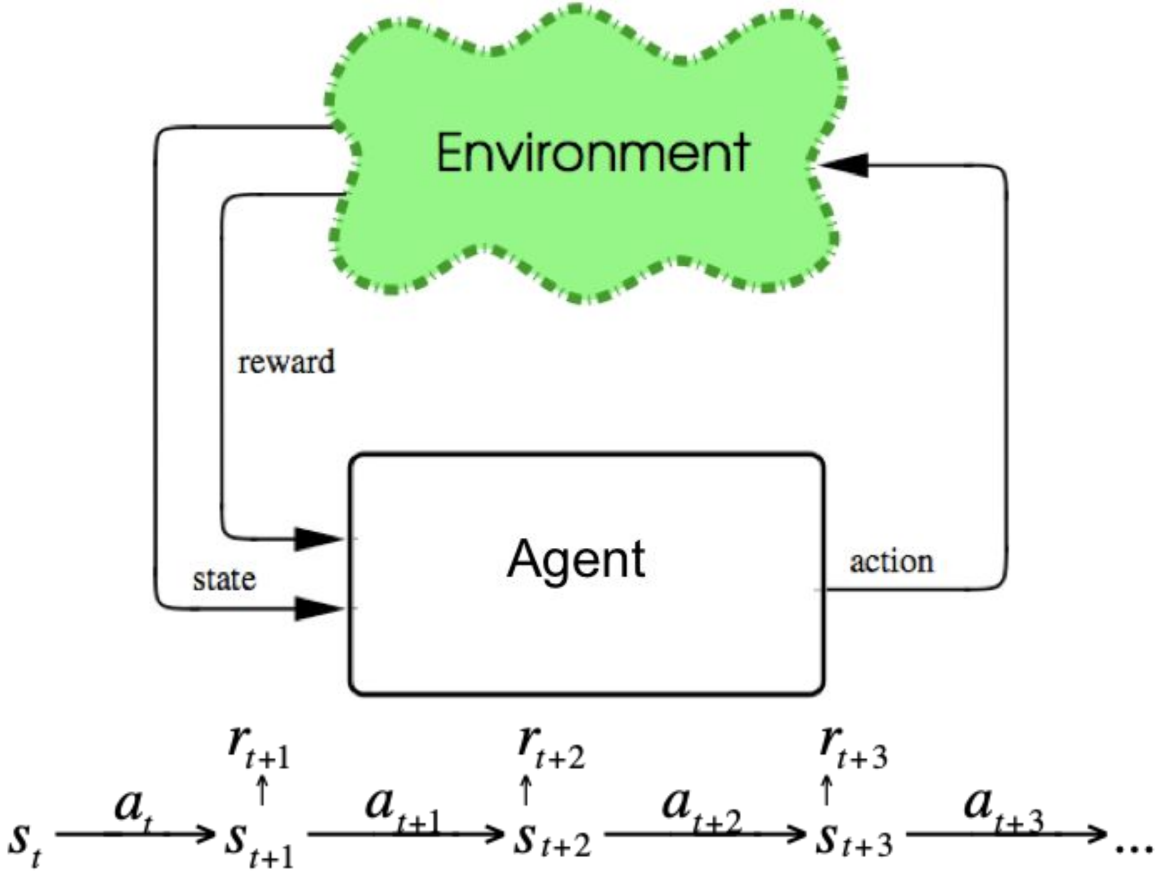
Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large n_samples, medium n_clusters with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not 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

Agenda

- What is machine learning?
- Supervised learning
- Unsupervised learning
- **Reinforcement learning**
- Examples

Reinforcement Learning



Example rewards: PacMan

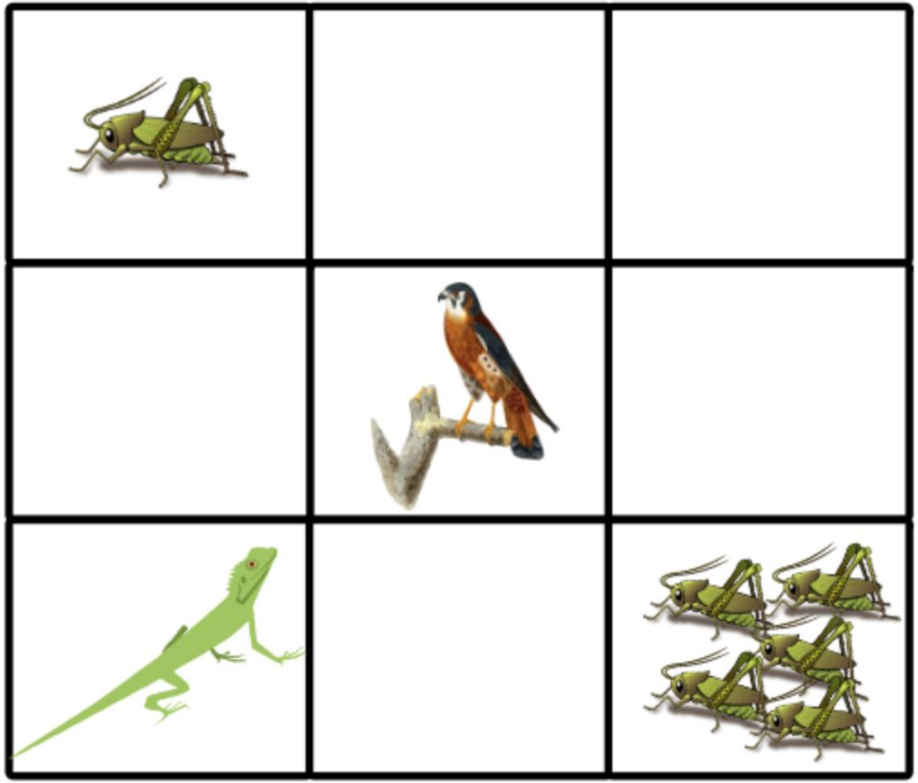
- One example:
  - 1 if you eat a pill
  - -10 if you get caught by a ghost
  - 2 if you eat a power pill or eat a ghost
  - 0 otherwise
- Another example:
  - -1 at every time step
  - 1,000,000 if you win the level

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!

Q-learning example: The Lizard Game



Agent: lizard

Goal: Eat as many crickets as possible as fast as possible without meeting a bird

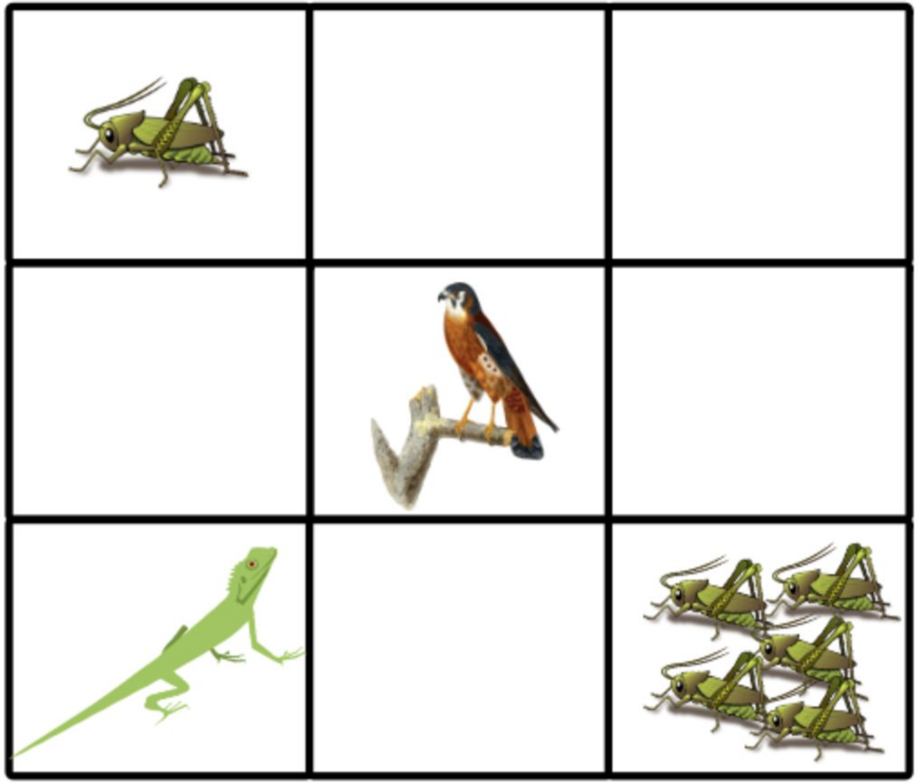
Actions: up, down, left, right

States: tiles

Rewards:

State	Reward	Game over?
1 cricket	1	No
Empty	-1	No
5 crickets	10	Yes
Bird	-10	Yes

Q-learning example: The Lizard Game



What would happen if we only did exploitation?

What would happen if we only did exploration?

State	Reward	Game over?
1 cricket	1	No
Empty	-1	No
5 crickets	10	Yes
Bird	-10	Yes

Agenda

- What is machine learning?
- Supervised learning
- Unsupervised learning
- Reinforcement learning
- **Examples**

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

What is the best option? (poll)

1. Linear regression
2. Overfitting
3. Clustering
4. Reinforcement learning

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

What is the best option? (poll)

1. Linear regression
2. Overfitting
3. Clustering
4. Reinforcement learning

A note on **GPT**

- GPT = **Generative Pretrained Transformer** language model
- It is **huge** and trained on **large amounts of text** (the internet)

- **GPT is a ML model that predicts the next word.**

  Input: a sequence of words (or just the "start of sequence" token)
  Output: the next word

A note on **GPT**

- GPT = **Generative Pretrained Transformer** language model
- It is **huge** and trained on **large amounts of text** (the internet)

- **GPT is a ML model that predicts the next word.**

  Input: a sequence of words (or just the “start of sequence” token)
  Output: the next word

- GPT can “hallucinate” facts
- GPT reproduces the social bias it learned from its training set (the internet)

Key take-aways

- **Supervised** vs. **unsupervised** vs. **reinforcement** learning
- **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