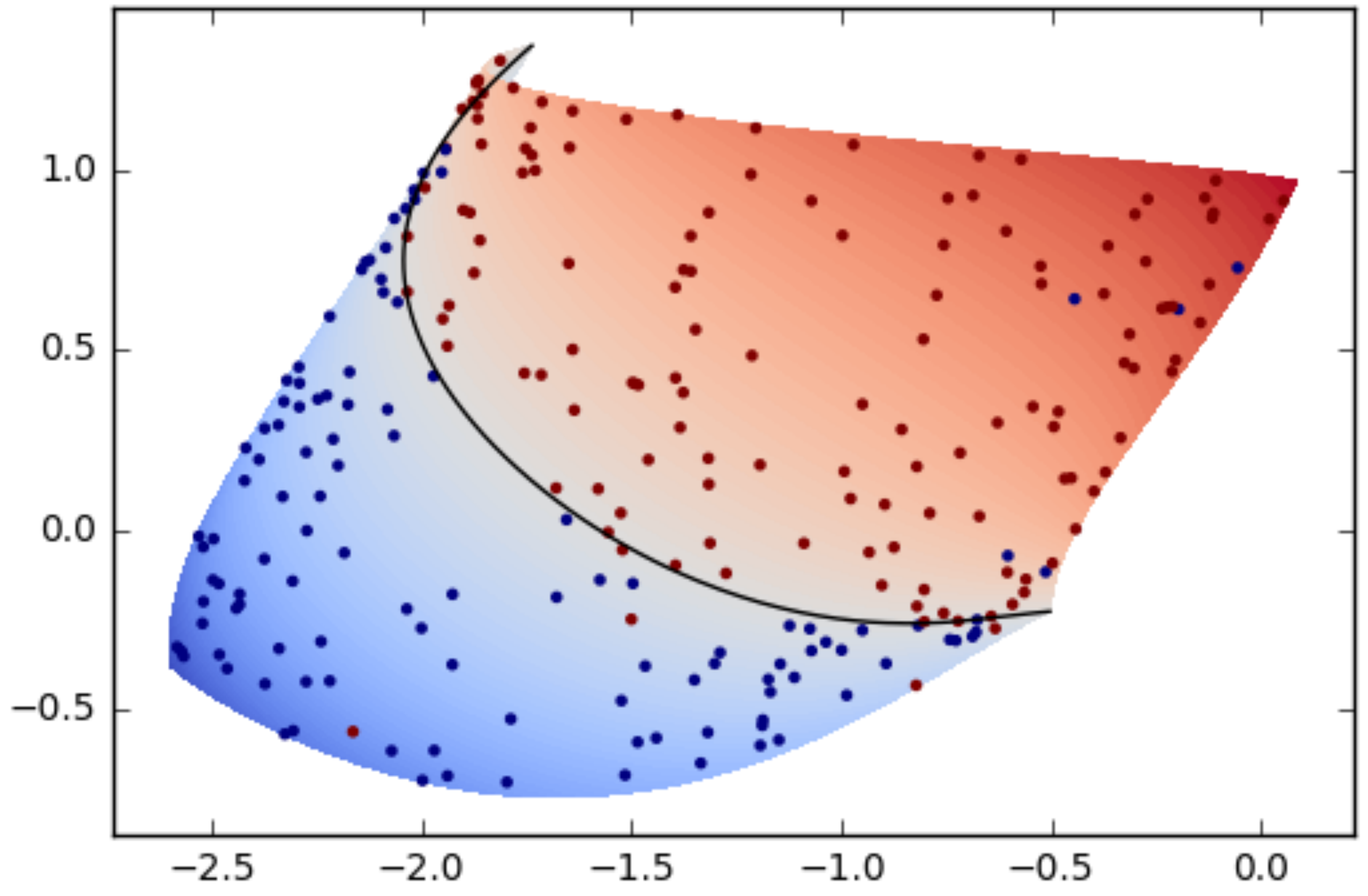
CS 273A: Machine Learning

Winter 2021

Lecture 1: Introduction

Roy Fox
Department of Computer Science
Bren School of Information and Computer Sciences
University of California, Irvine



All slides in this course adapted from Alex Ihler & Sameer Singh

Today's lecture

What is machine learning?

Course logistics

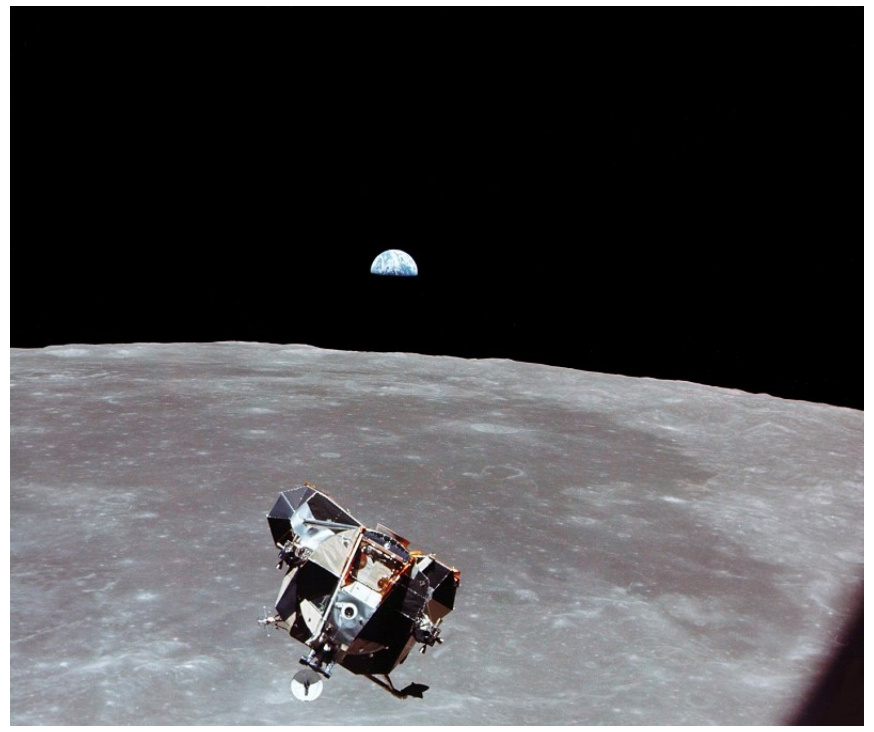
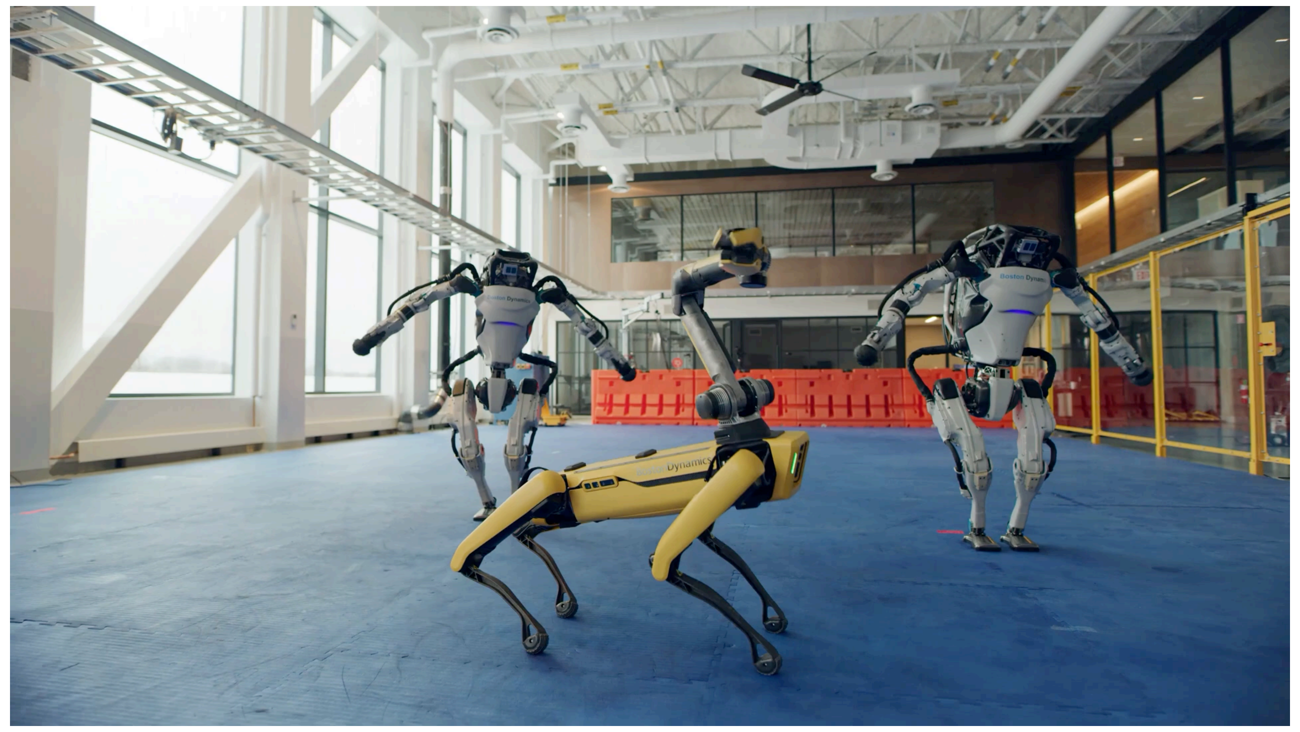
Data management and visualization

Supervised learning

Artificial intelligence (AI) beyond ML

- Machine learning (ML) is a way to get machines to be intelligent
- Not the only way:
    - Engineered solutions (expert systems)
    - Good old-fashioned AI (GOFAI)
        - Rule-based systems
        - Logic programming (e.g. Prolog)
        - Search algorithms
    - Model-based optimization



What is intelligence?

- Big question, beyond our scope...
- Behavioristic definition: intelligence = good decision making
  - Can all intelligent behavior be reduced to good decision making?
- Decision making: in situation x, do y (decision / prediction / action)
  - At the core of AI systems: a decision function $f : x \mapsto y$
  - Examples: visual classification, price prediction, medical diagnosis, motor control
- "Good" decision: assume a given score function $v : x, y \mapsto \mathbb{R}$, higher = better
  - Or loss function $\ell : x, y \mapsto \mathbb{R}$, lower = better

What is learning?

- Learning = taking in information to know more than you did before
  - But what is knowledge? Another big question...
- Machine learning = use data to make better decisions than before [Mitchell 1997]
- ML can help when other AI methods fail:
  - Experts are scarce
  - Rules / logic are hard to specify
  - Search space is too large
  - Models are unknown / hard to specify

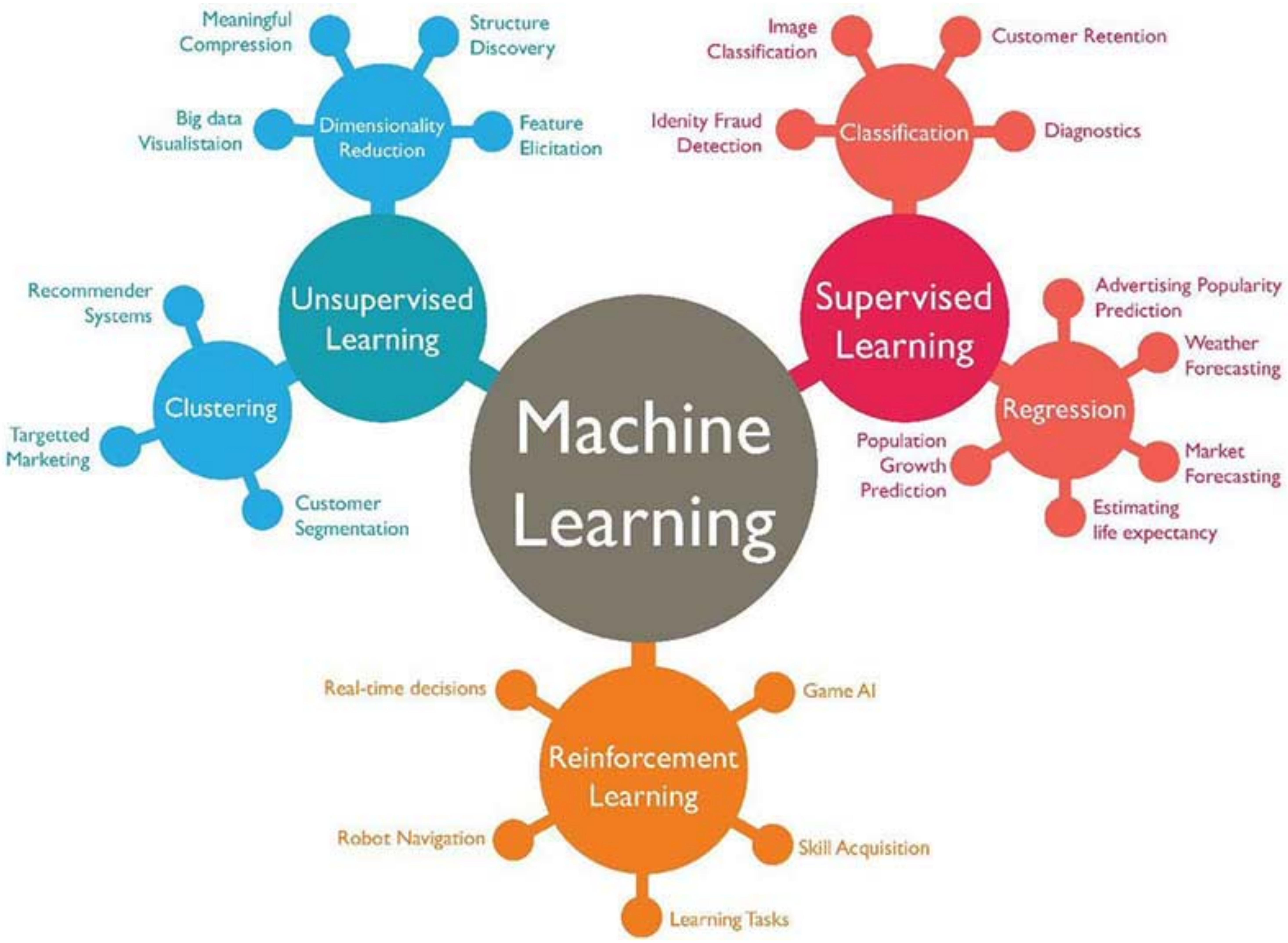
Statistics vs. ML

- Statistics = mathematical toolset for analyzing data
- ML = using data to build AI systems
- Successful ML draws on many disciplines
- The ML stack:
  - Math: probability theory, (linear) algebra, computational learning theory
  - Algorithms: ML algorithms, optimization, data structures
  - Software: ML frameworks, databases, testing, deployment
  - Hardware: cloud computing, distributed systems, cyber-physical systems

The ML stack:

math
algorithms
software
hardware

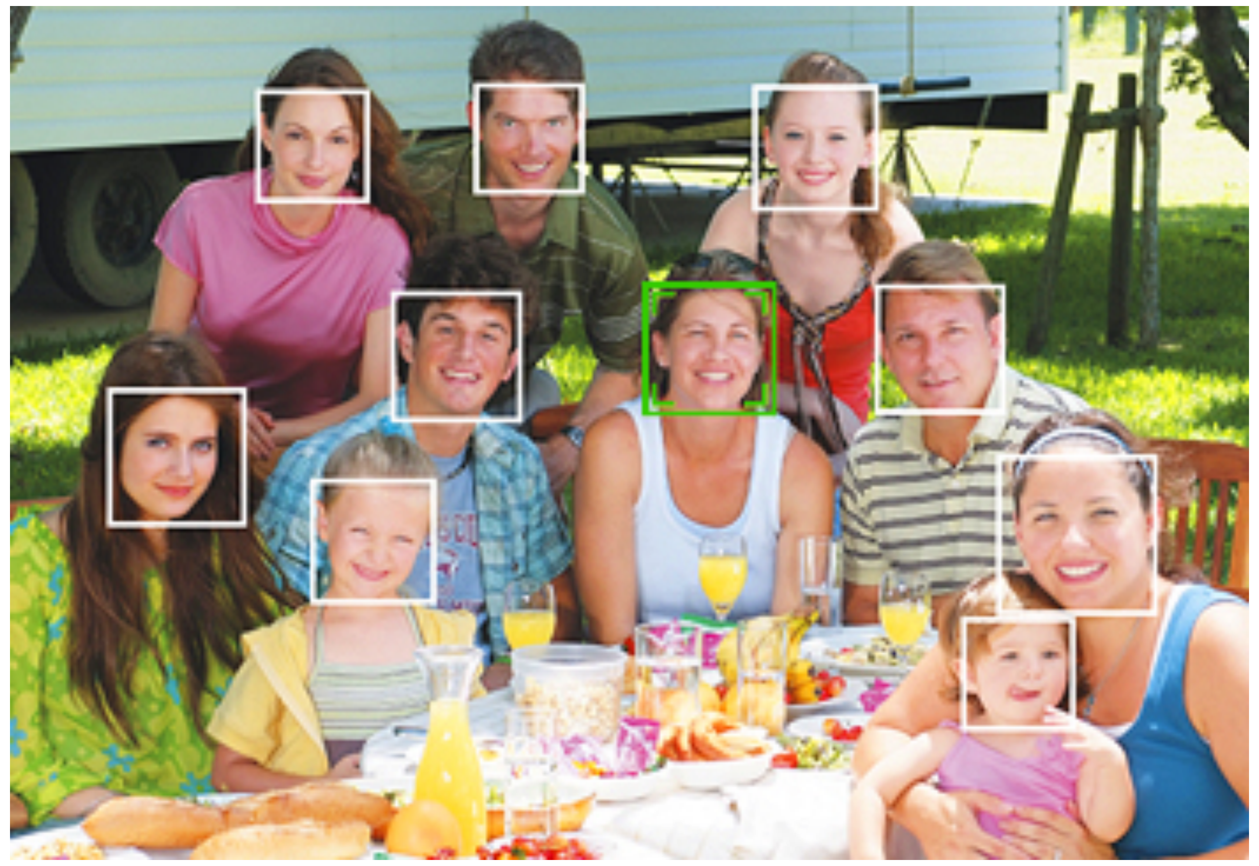
Taxonomy of ML

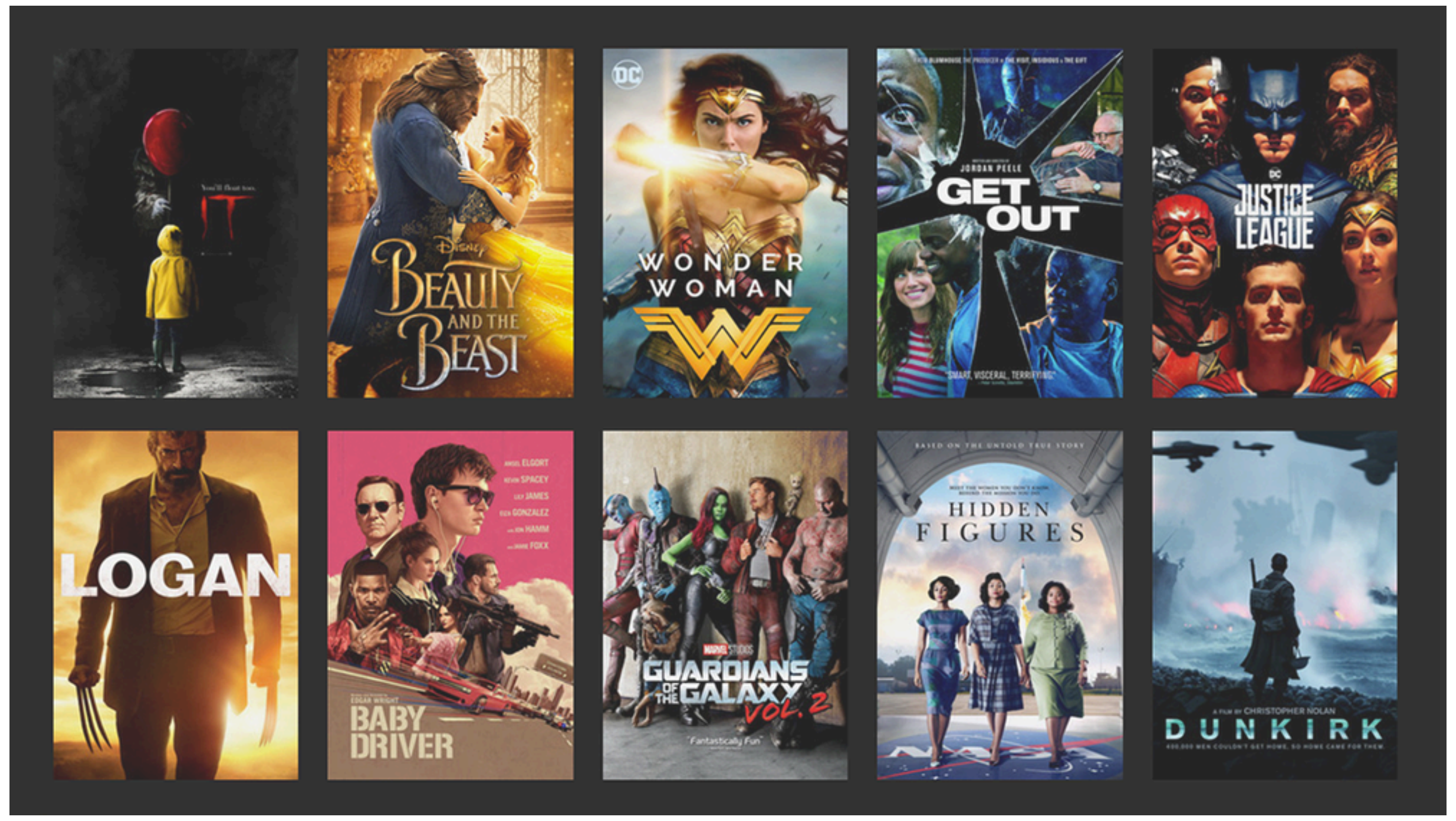


https://www.techleer.com/articles/203-machine-learning-algorithm-backbone-of-emerging-technologies/

Learning settings (1): supervised learning

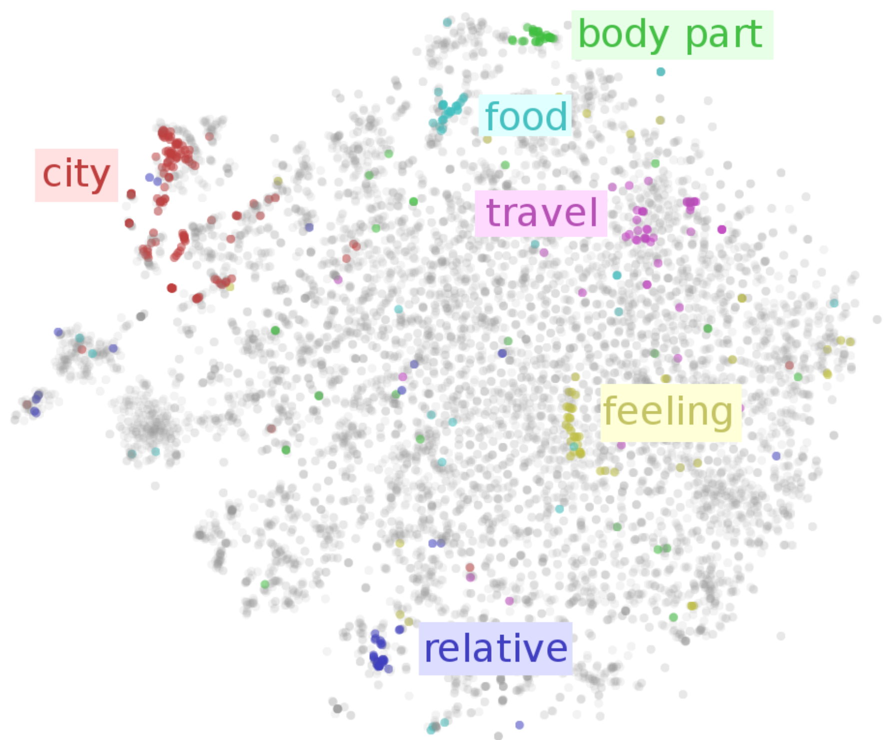
- How can we learn $f : x \mapsto y$ that achieves good performance $v(x, y)$?
- Supervised learning
  - Data: examples of instances x and good decisions y (labels)
  - Given a training dataset $\mathscr{D}$, find f that agrees with $\mathscr{D}$'s labels on its instances
  - Classification: y is a class in a small set
  - Regression: y is continuous



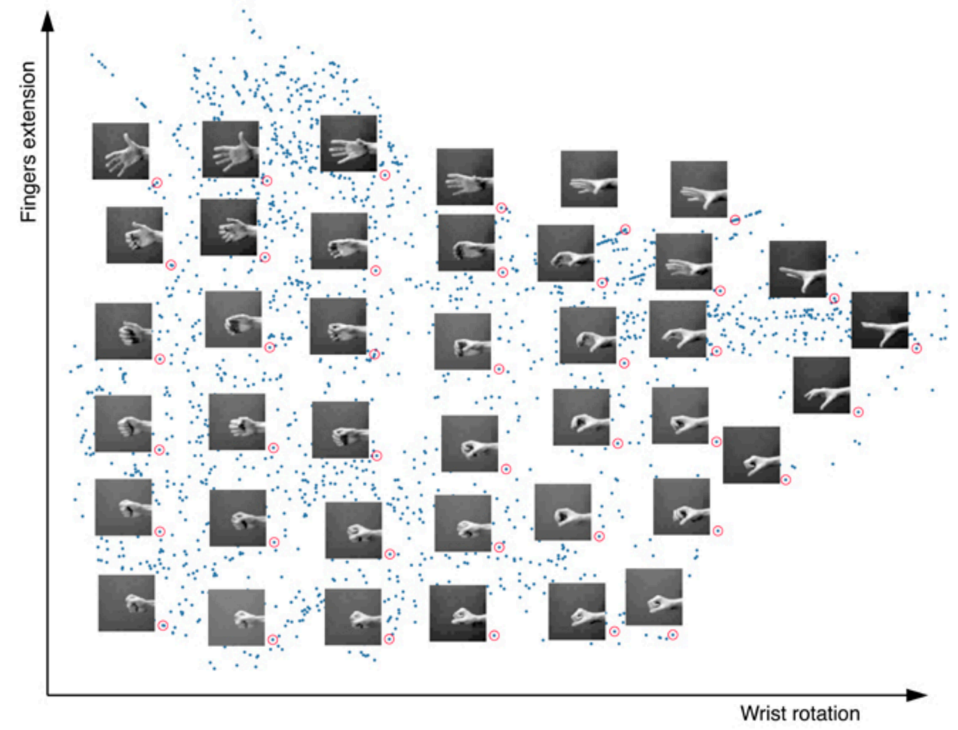


Learning settings (2): unsupervised learning

- How can we learn $f : x \mapsto y$ that achieves good performance $v(x, y)$?



- Unsupervised learning
  - Data: examples of instances x (no labels y)
  - What are we looking for? Some insight, discover pattern / structure of the data
  - Performance measure v / ℓ is often global rather than per-instance
  - Clustering: y is a cluster in a small set
  - Dimensionality reduction: y is a low-dimensional representation
  - Density estimation, anomaly detection, ...

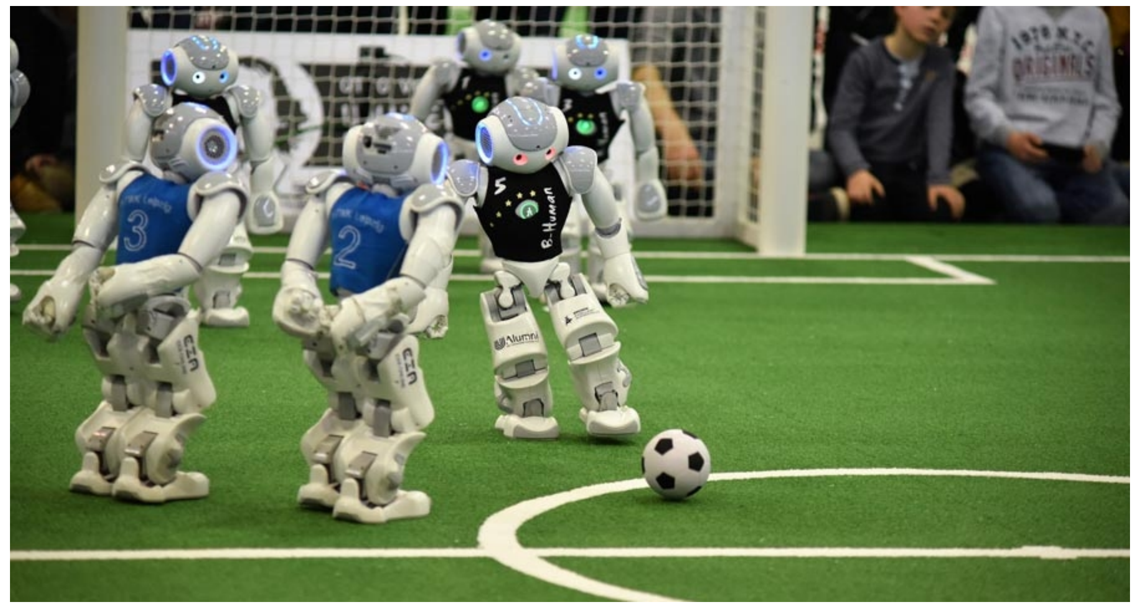


https://ruder.io/word-embeddings-1/
https://www.cs.cmu.edu/~efros/courses/AP06/presentations/melchior_isomap_demo.pdf

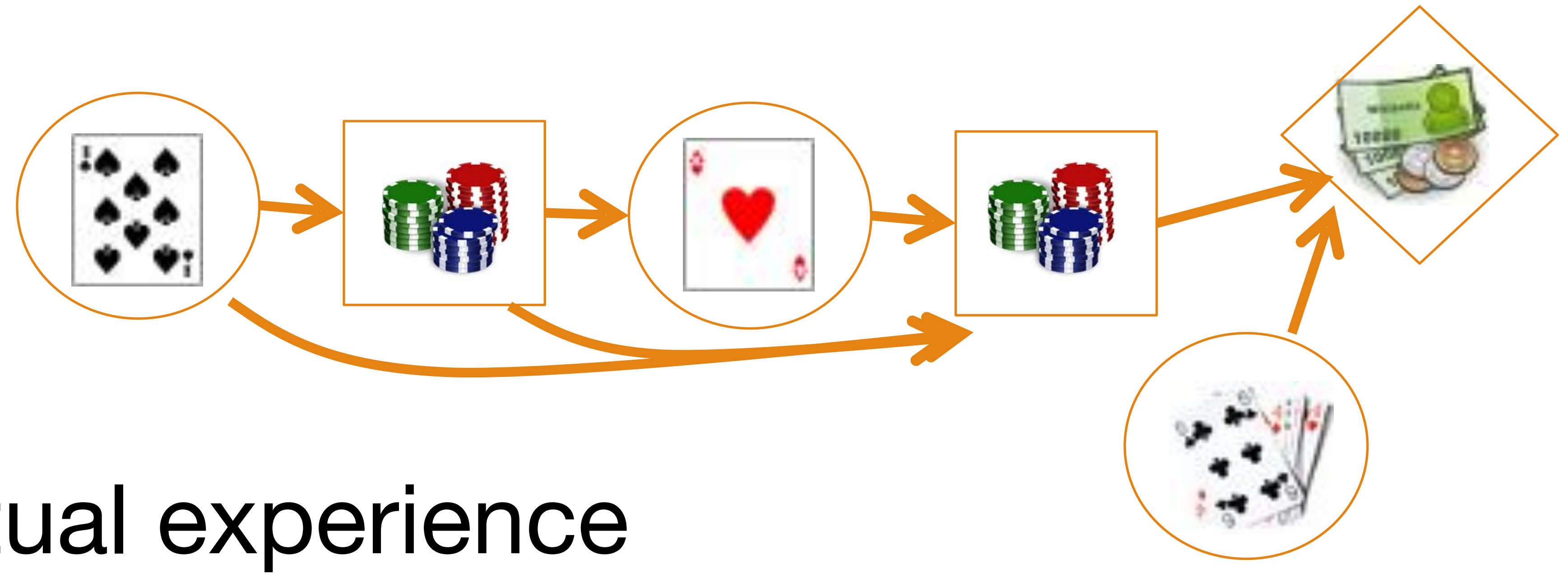
Learning settings (3): reinforcement learning

- How can we learn $f : x \mapsto y$ that achieves good performance $v(x, y)$?

- Reinforcement learning



  - Decisions are actions that the agent takes in the environment
  - No dataset, data is collected through this interaction
  - Several new challenges:



    - Online learning: score v (reward) is only revealed for actual experience
    - Active learning: the agent also decides on which instances x to visit
    - Sequential decisions: how to assign the credit for v to parts of the decision sequence?

Mixed supervision

- Learning settings can be mixed
- Semi-supervised learning:
  - Mixture of supervised and unsupervised learning
  - Benefit from seeing labels y on some instances x
  - Benefit from seeing a large set of (unlabeled) instances x
  - Examples: image tagging, document retrieval, medical diagnosis

Recap

- Machine learning: data-driven approach to building AI
  - Use experience to improve performance on decision / prediction task
- Common learning settings:
  - Supervised learning: $\mathscr{D} = \{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$
  - Unsupervised learning: $\mathscr{D} = \{x^{(1)}, \dots, x^{(m)}\}$
  - Semi-supervised learning: only some instances are labeled
  - Reinforcement learning: experience gathered by agent

Today's lecture

What is machine learning?

Course logistics

Data management and visualization

Supervised learning

Course logistics

- When: Tuesdays and Thursdays, 2–3:20pm
  - Lectures will be recorded and published afterwards
- Where: https://uci.zoom.us/j/94903054276
- Website: https://royf.org/crs/W21/CS273A/ ← Schedule!
- Forum: https://piazza.com/uci/winter2021/cs273a
  - For announcement and questions (no email please!)
- Assignments: https://www.gradescope.com/courses/220628
  - Published biweekly

Staff

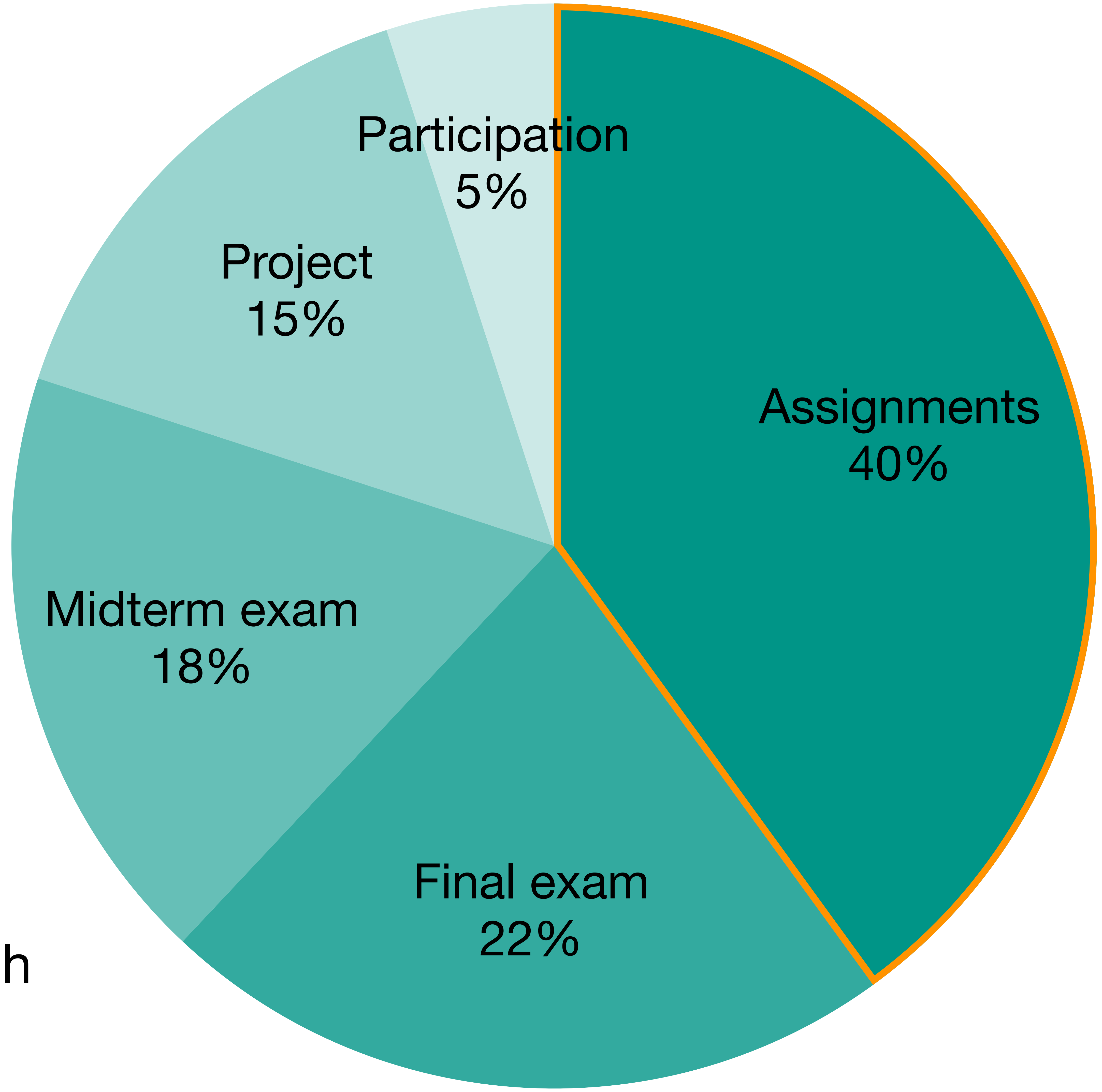
- Instructor: Prof. Roy Fox

- Teaching assistant: Hao Tang

- Contact us on piazza (publicly or privately)
    - We may be unreachable by email
- Office hours: https://calendly.com/royfox/office-hours
    - Welcome to schedule 15-min slots and invite friends; give 4 hour notice

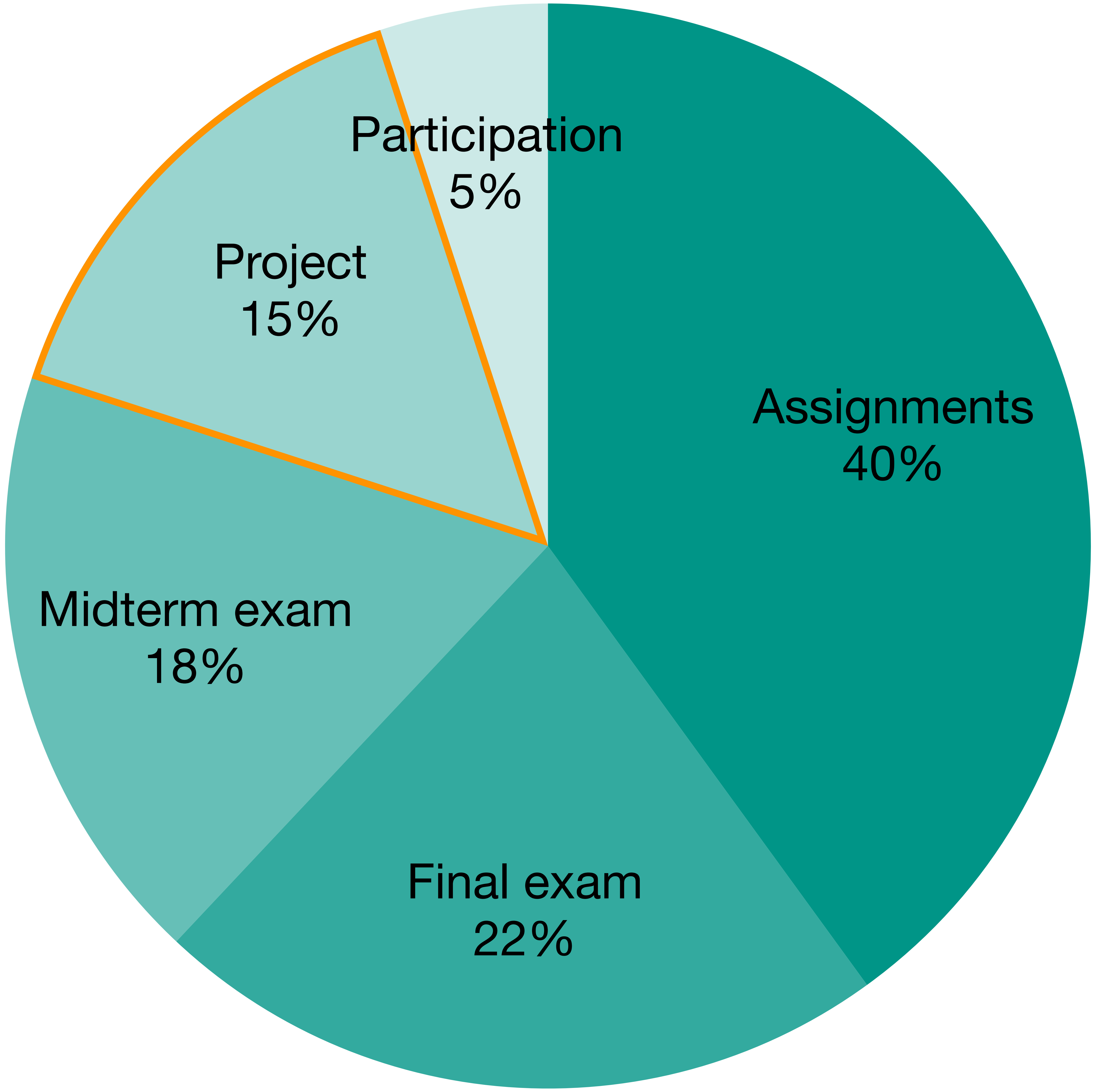
Grading policy: assignments

- 5 programming assignments
  - Apply ML techniques in Python
  - Show your code and results
    - We will read it, not run it
  - Must include statement of collaboration
- Grading:
  - 40% of final grade
  - Your 4 best assignments count for 10% each
  - But no late submission



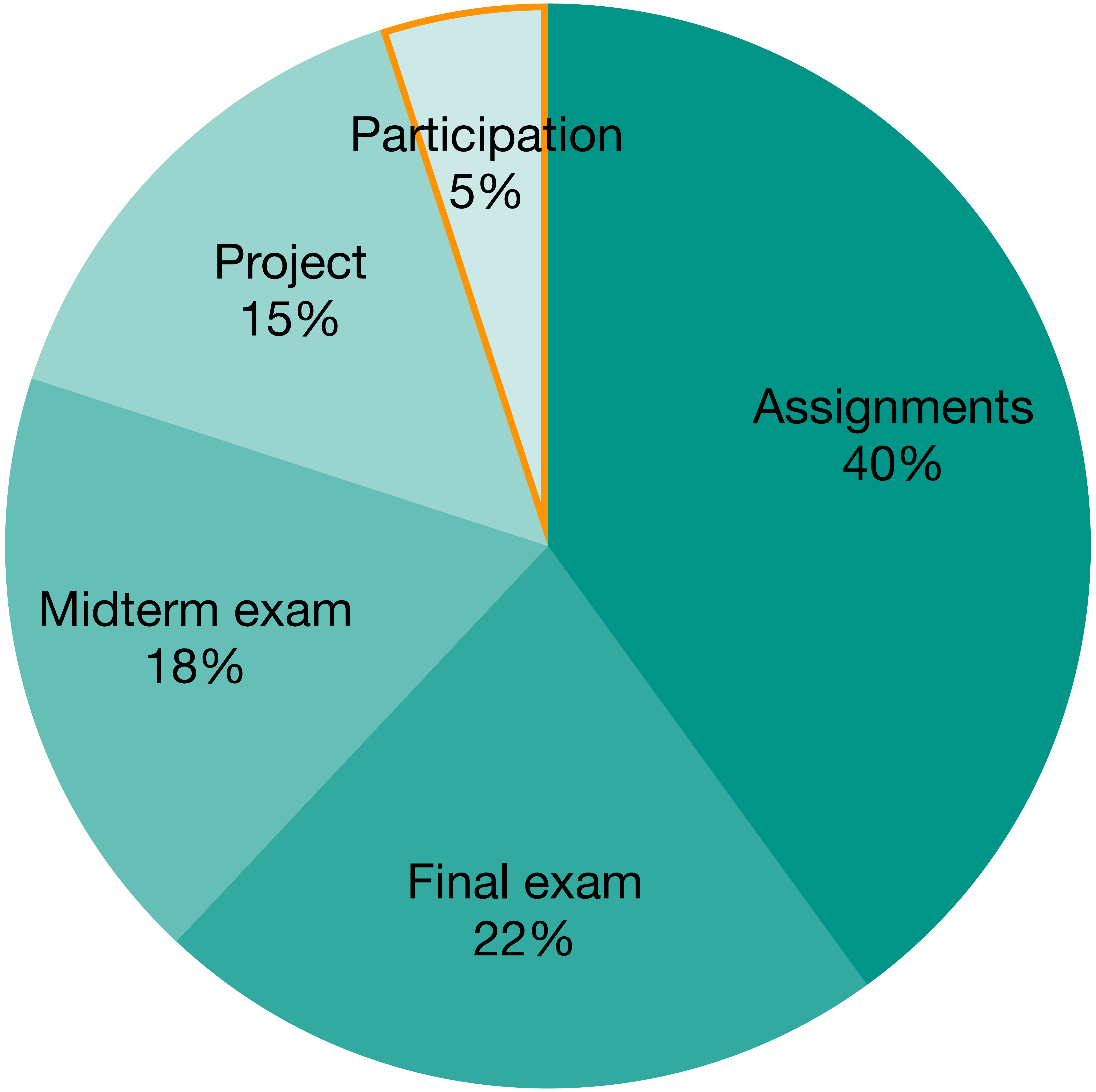
Grading policy: project

- Teams of 3
  - Start forming teams now
- Deadlines:
  - Team roster — week 4 (1% credit)
  - Abstract — week 7 (2% credit)
  - Report — week 10 (12% credit)



Grading policy: participation

- Forum participation
  - Ask questions if you have any
  - Answer questions if you can
  - Post relevant useful links
  - Upvote useful posts
  - Give private feedback to staff
- Quizzes, surveys, and evaluations
  - Answer polls published on the forum
  - Submit course evaluations



What will it take to do well?

- We'll rely heavily on math: probability theory, linear algebra, calculus
  - We're here to help, but solid background expected

- You'll need to code well in Python
- Some ideas are challenging — ask early what you don't fully understand
- Help your friends and get help — from us too! — but never cheat

Today's lecture

What is machine learning?

Course logistics

Data management and visualization

Supervised learning

Know thy data

- ML is a data science
  - Look at your data, know what it is, get a "feel" for it
- How many data points?
- What are the features of every data point? What are their data types?
  - Booleans (spam, inbound/outbound, control group)
  - Discrete categories (country/state, protocol, user ID)
  - Integers (1–5 stars, # of bedrooms, year of birth)
  - Reals — up to digital representation (pixel intensity, price, timestamp)
- Is there missing data? Unreasonable values? Surprisingly missing / repeated values?

Data wrangling

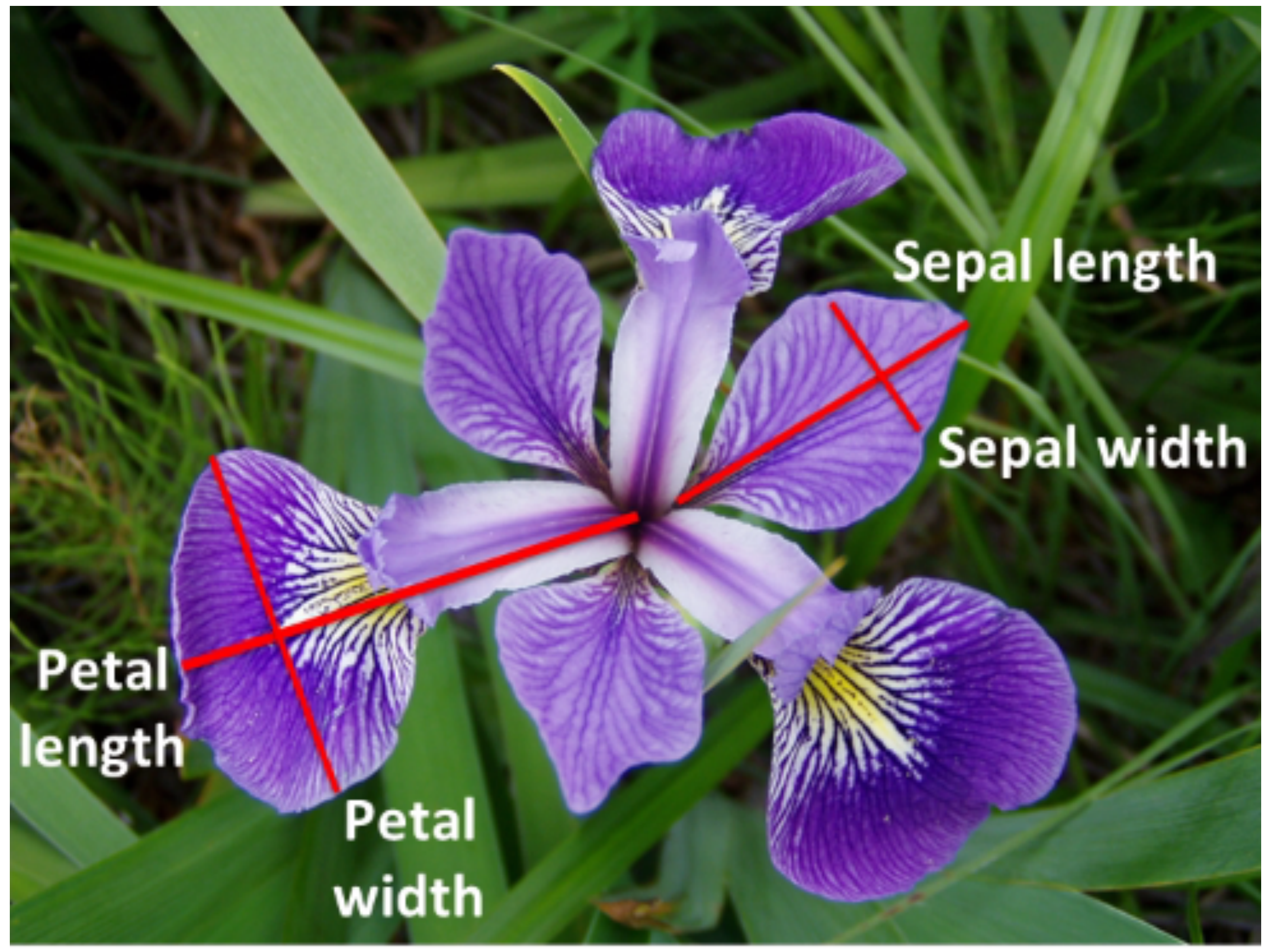
- Data wrangling: tools and practices for preparing data for usage
  - Discovering: explore the data to understand what it is
  - Structuring: organize into useful features; e.g. Jan 5, 2021 → (2021, 1, 5) or 18632
  - Cleaning: standardize values, remove errors, flag missing data; e.g. Calif. → CA
  - Validating: flag inconsistencies, surprising value distributions
  - Publishing: verify that the data format is readable in the intended way

Programming with data

- Python
  - numpy, matplotlib, scipy, pandas, scikit-learn, tensorflow / pytorch...
- Matlab / Octave: still popular in some engineering fields
- R: popular among statisticians
- C/C++: used for performance in production, not for research / prototyping
- Other niche languages and tools for visualization and modeling

Example: Iris flower dataset

- Dataset of 3 species of Iris, $y \in \{0,1,2\}$
- 150 data points, 50 of each class, $|\mathcal{D}| = 150$
- 4 features per data point: length & width of sepals and petals, x_1, x_2, x_3, x_4




Representing the data

- $m = 150$ data points, $\mathscr{D} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$
- Each instance is a vector of $n = 4$ features, $x^{(j)} = \begin{bmatrix} x_1^{(j)} & \cdots & x_n^{(j)} \end{bmatrix} \in \mathbb{R}^n$
- We can represent this as a data matrix $x = \begin{bmatrix} x_1^{(1)} & \cdots & x_n^{(1)} \\ \vdots & & \vdots \\ x_1^{(m)} & \cdots & x_n^{(m)} \end{bmatrix} \in \mathbb{R}^{m \times n}$

```
>>> from sklearn import datasets  # import scikit-learn
>>> iris = datasets.load_iris()  # load dataset
>>> X, y = iris.data, iris.target
>>> X.shape
(150, 4)
>>> y.shape
(150,)
```

Basic statistics

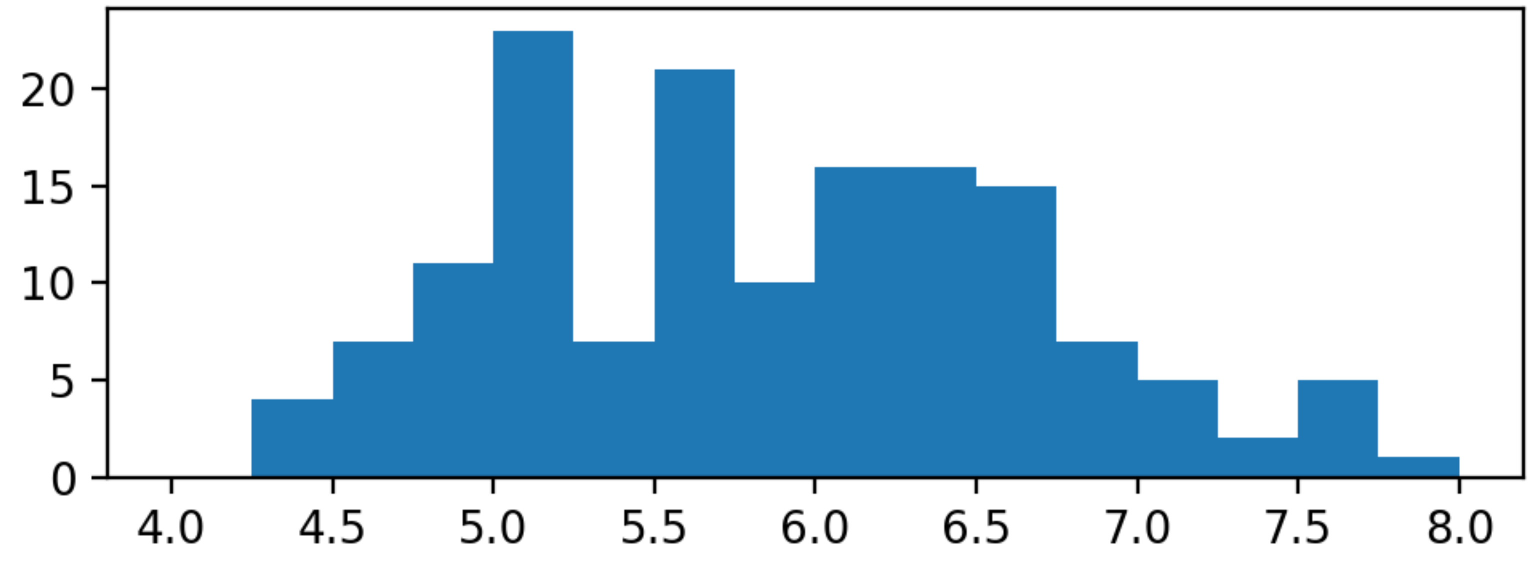
- Let's look at the basic statistics of the data
  - Location (mean value)
  - Scale (standard deviation)
  - Order statistics (minimum, maximum, median)

```
>>> import numpy as np
>>> X.mean(axis=0)
array([5.843, 3.057, 3.758, 1.199])
>>> X.std(axis=0)
array([0.825, 0.434, 1.759, 0.76 ])
>>> X.min(axis=0)
array([4.3, 2. , 1. , 0.1])
>>> np.median(X, axis=0)
array([5.8 , 3.  , 4.35, 1.3 ])
>>> X.max(axis=0)
array([7.9, 4.4, 6.9, 2.5])
```

Data visualization: histograms

- Count the data points falling in each of k equal bins
  - Summarize data as a length-k vector of counts
  - Bins too small → too little aggregation, lose "topology" of data point clusters
  - Bins too large → too much aggregation, lose information about cluster sizes
    - Bins should become smaller the denser the data

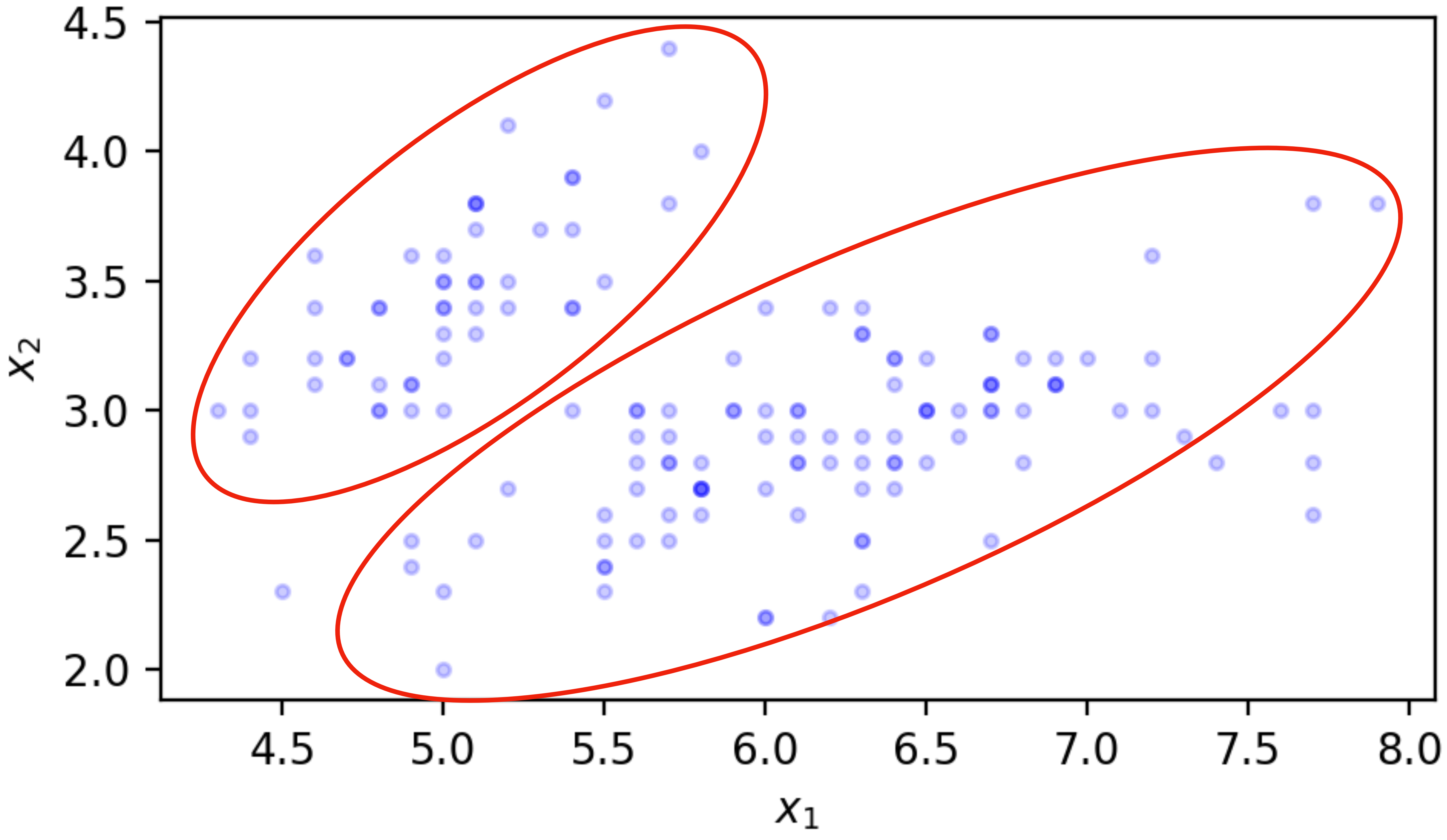
```
>>> import matplotlib.pyplot as plt
>>> plt.hist(X[:, 0], bins=np.linspace(4, 8, 17))
```



Data visualization: scatterplots

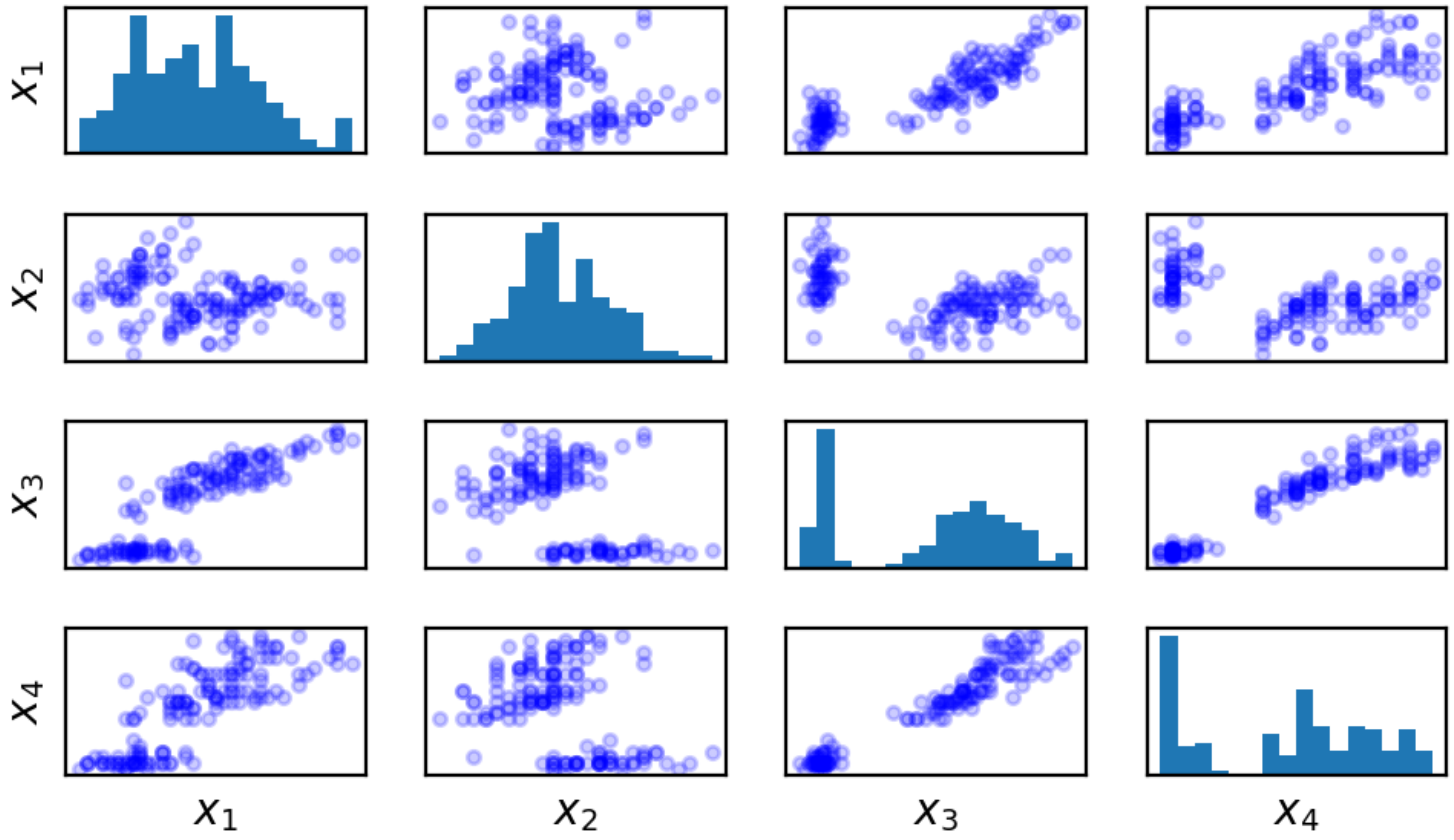
- Place data points on a 2D plane

```
>>> plt.plot(X[:, 0], X[:, 1], '.', color=[0., 0., 1., .2])
```



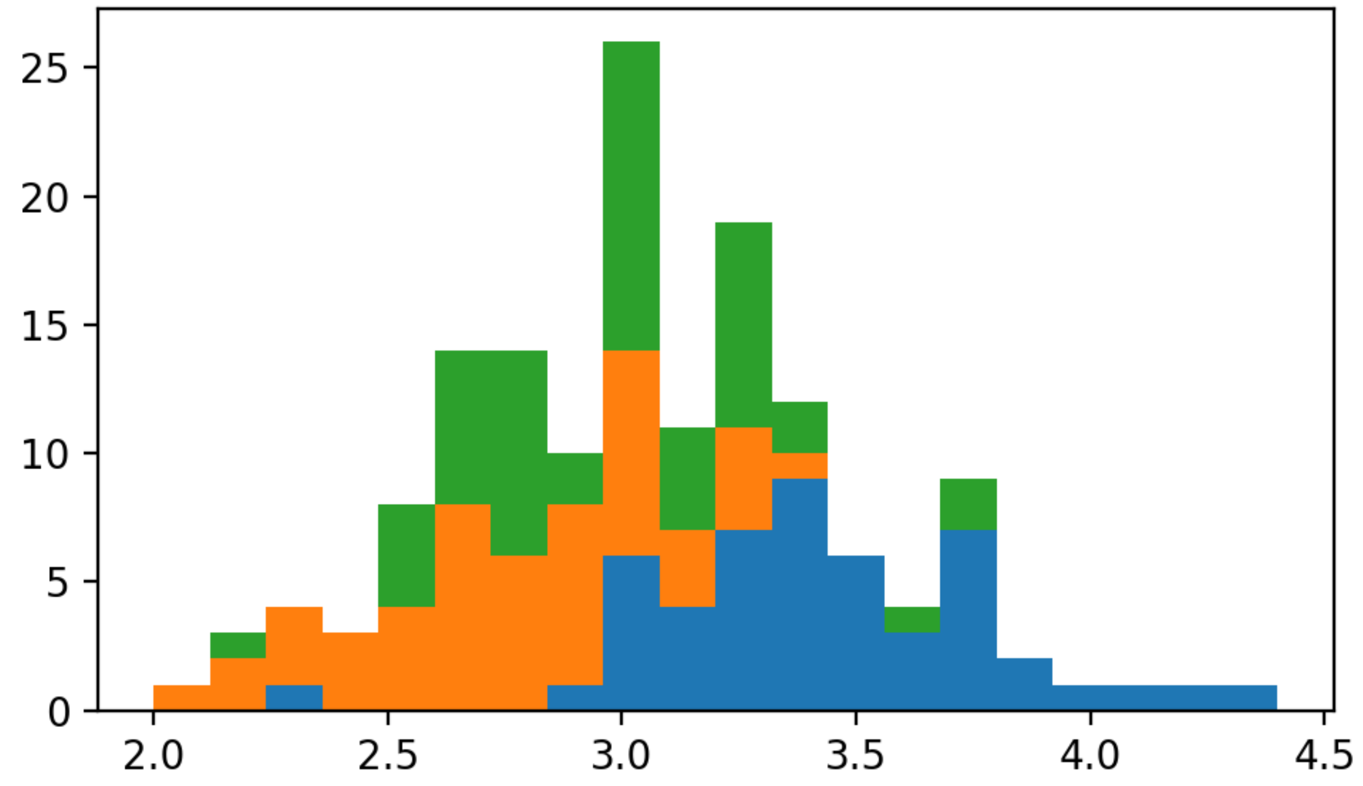
Data visualization: pair plot

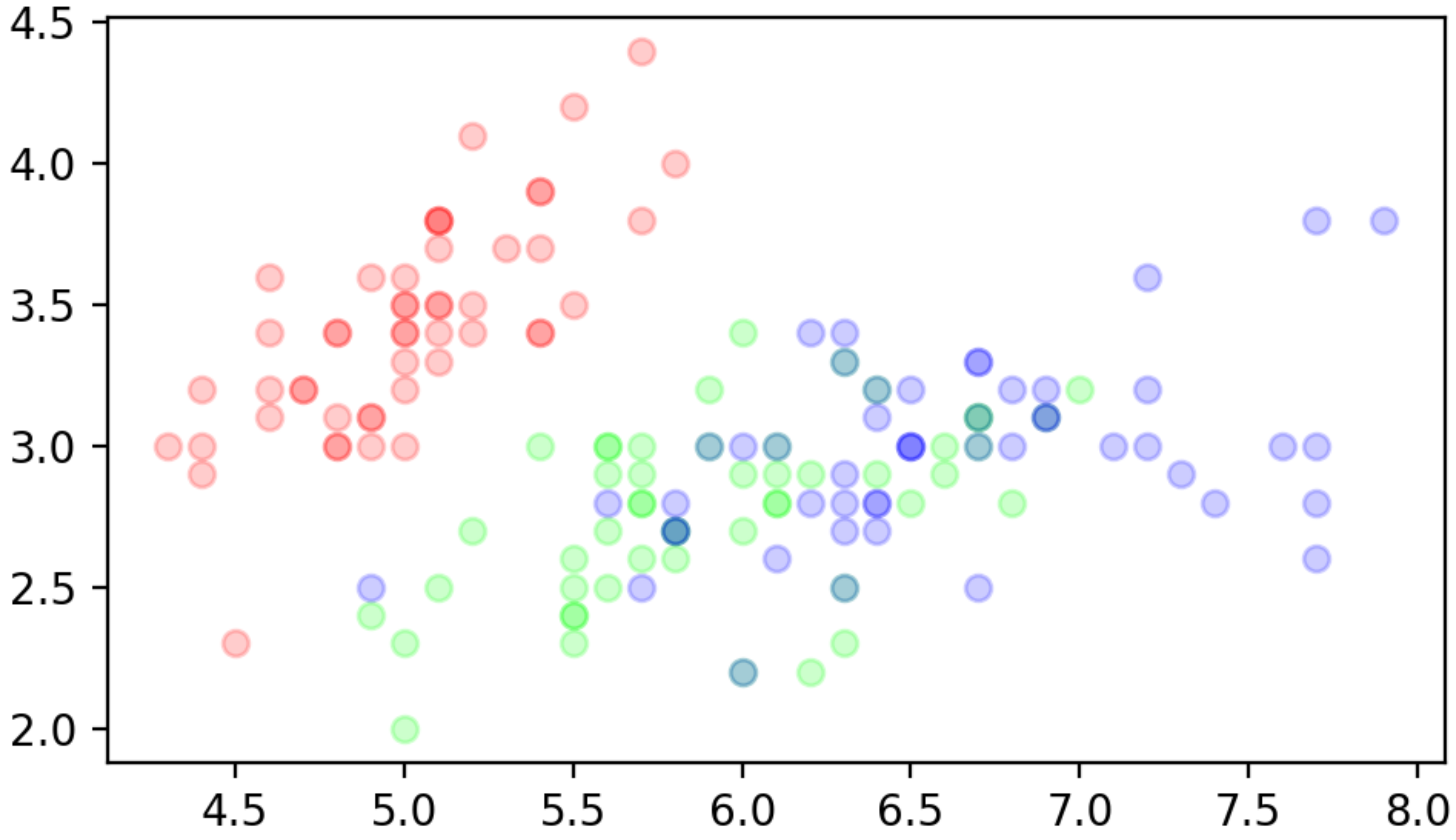
- With more than two features, plot all pairs
  - Histograms on the diagonal



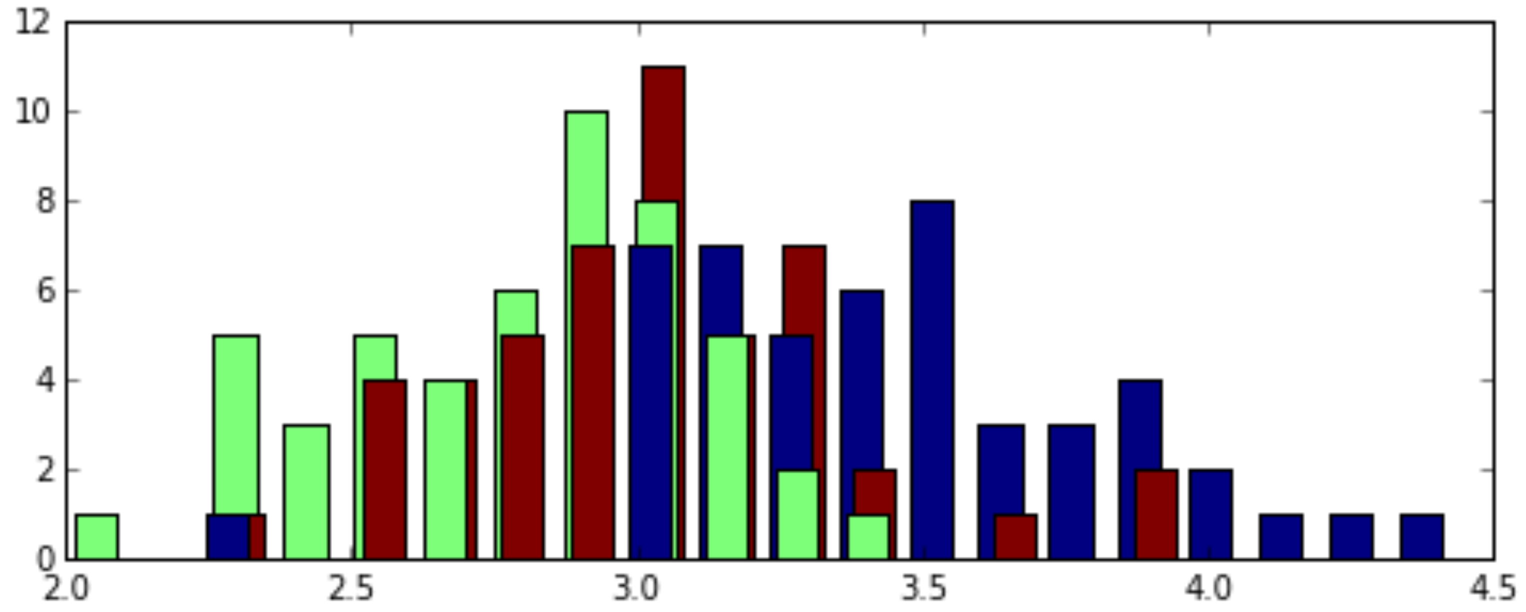
Visualizing labels

- How are different classes distributed?
  - Histograms can be stacked:





  - or side-by-side:



Today's lecture

What is machine learning?

Course logistics

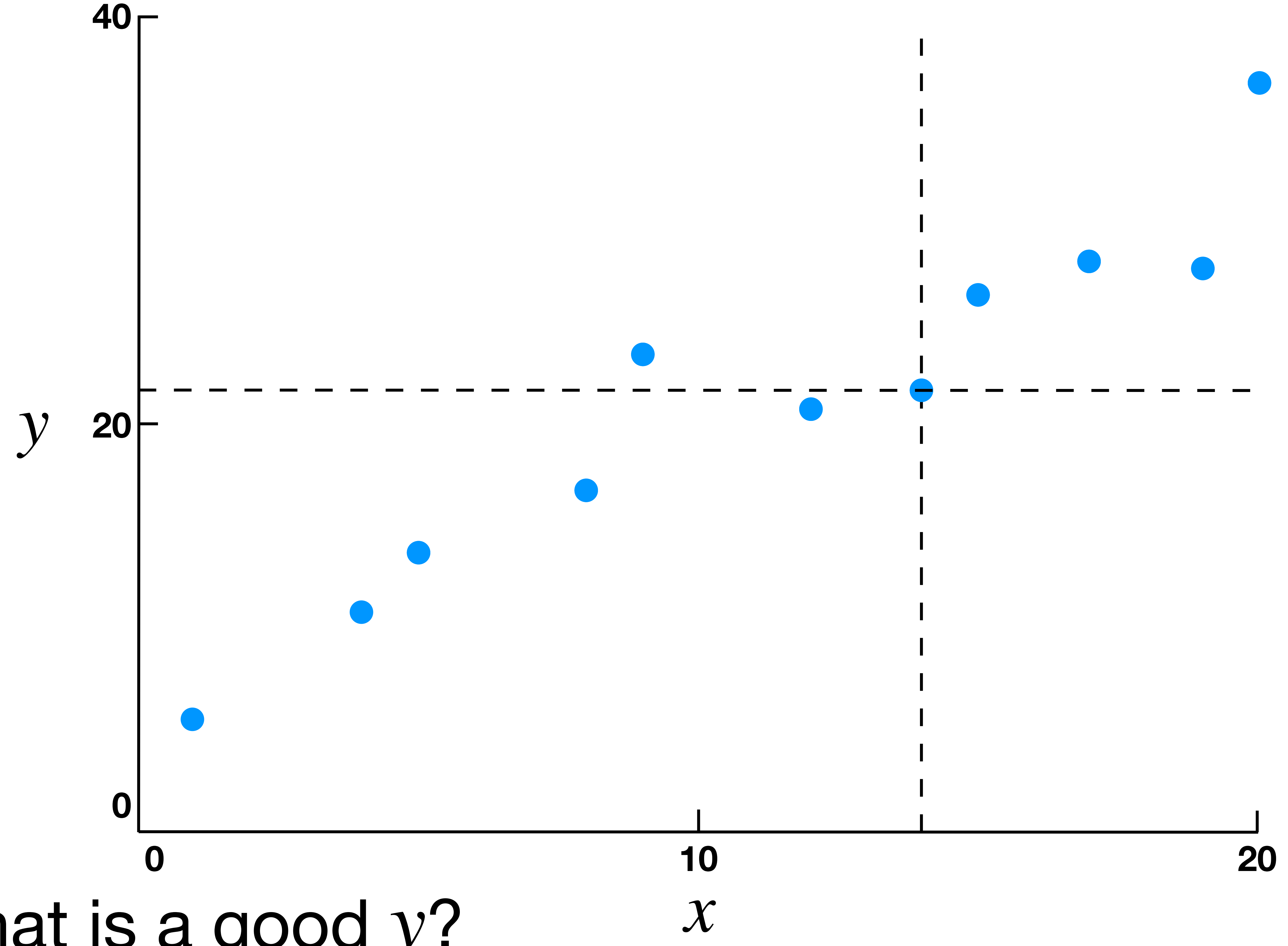
Data management and visualization

Supervised learning

Supervised learning

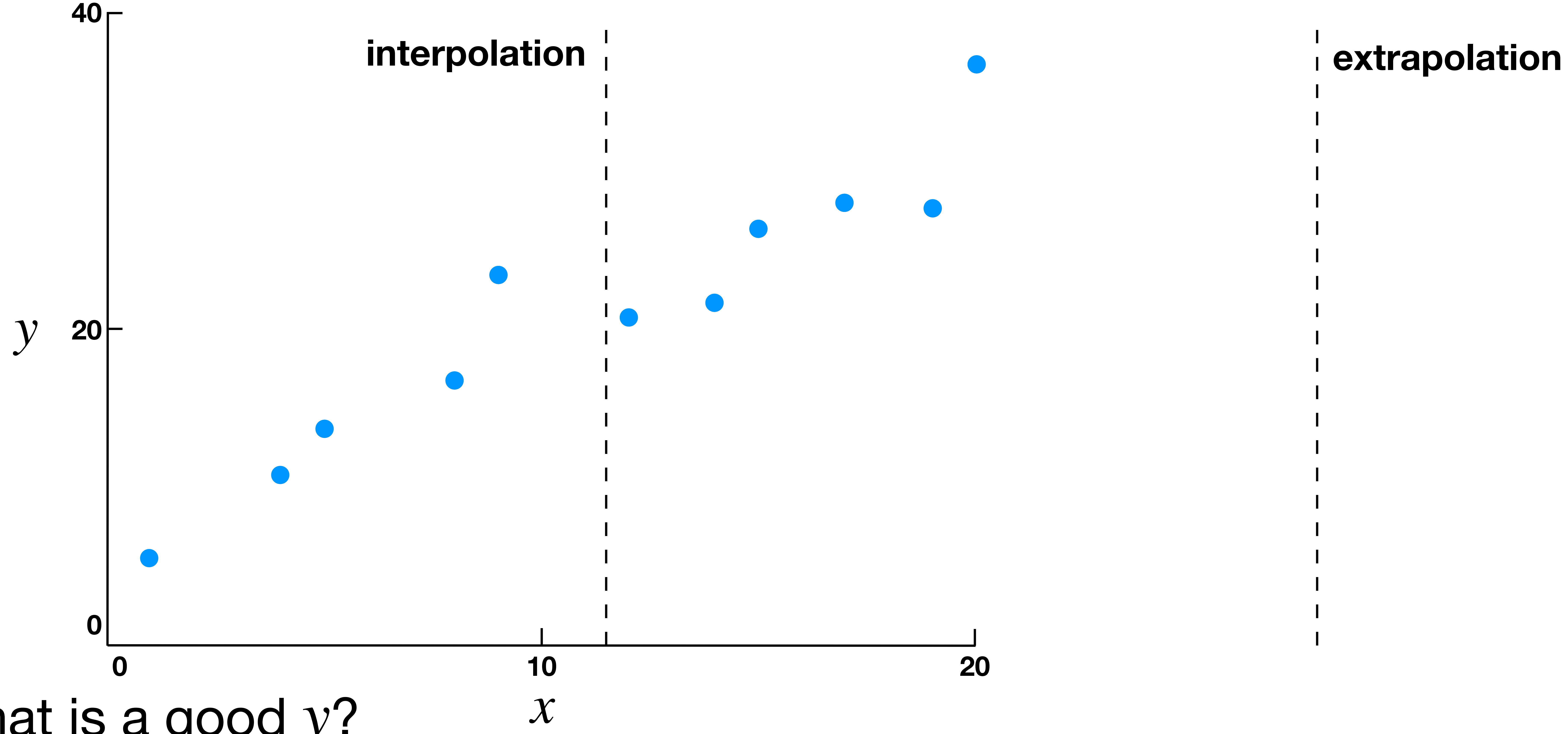
- Data shows trend

- But also "noise"



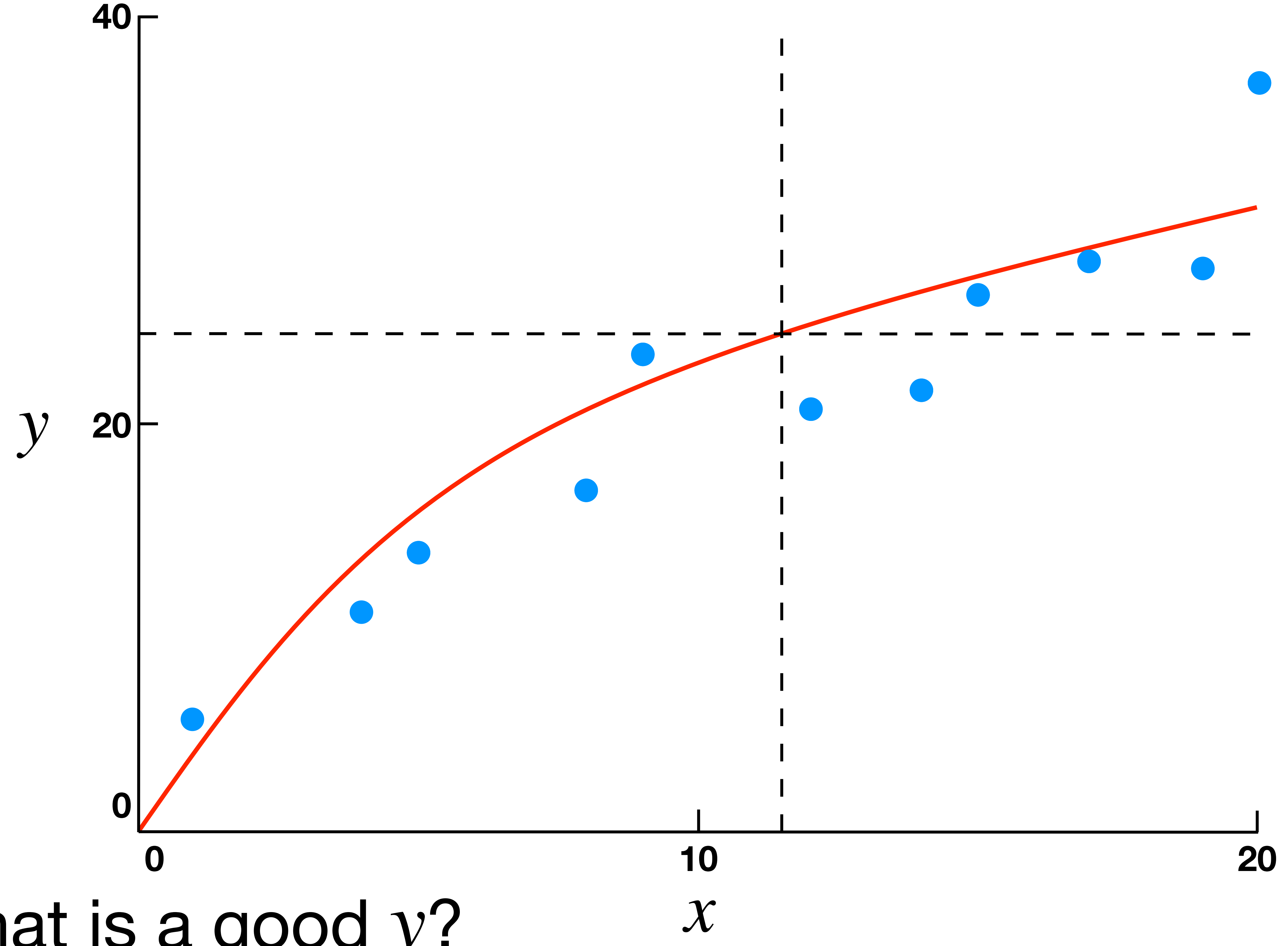
- Given some x, what is a good y?

Supervised learning



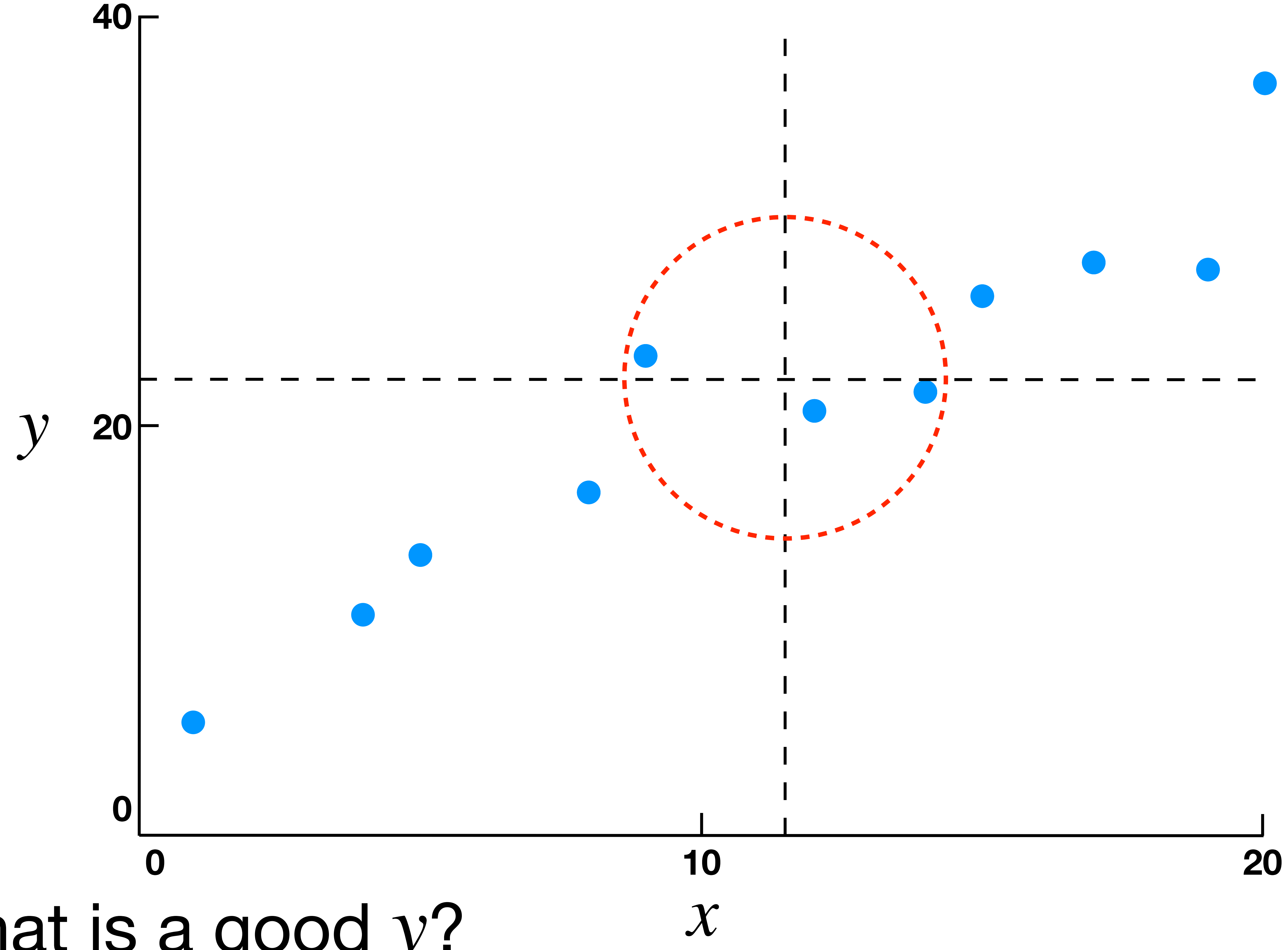
- Given some x, what is a good y?

Supervised learning



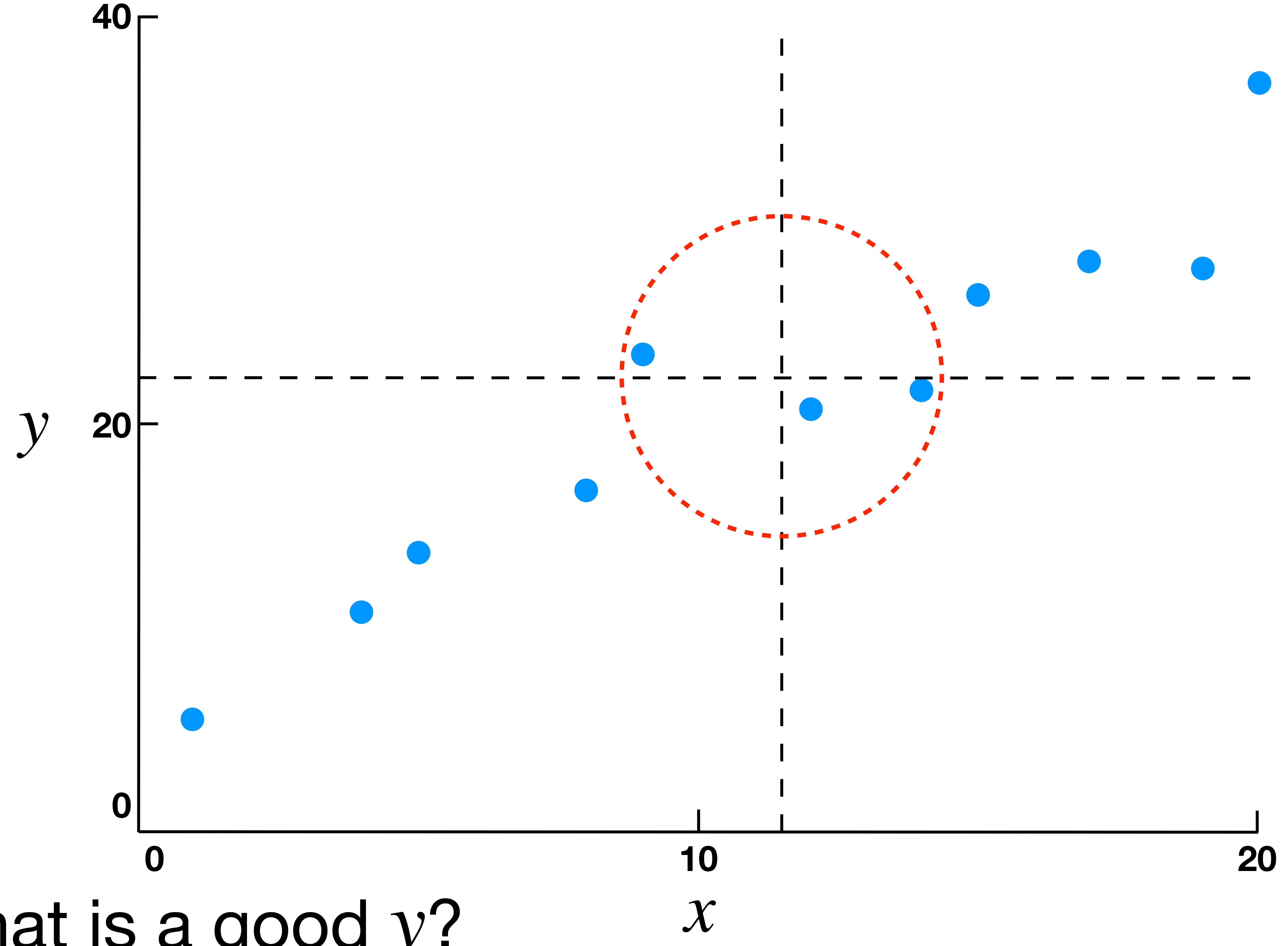
- Given some x, what is a good y?
  - Directly represent $f : x \mapsto y$

Supervised learning



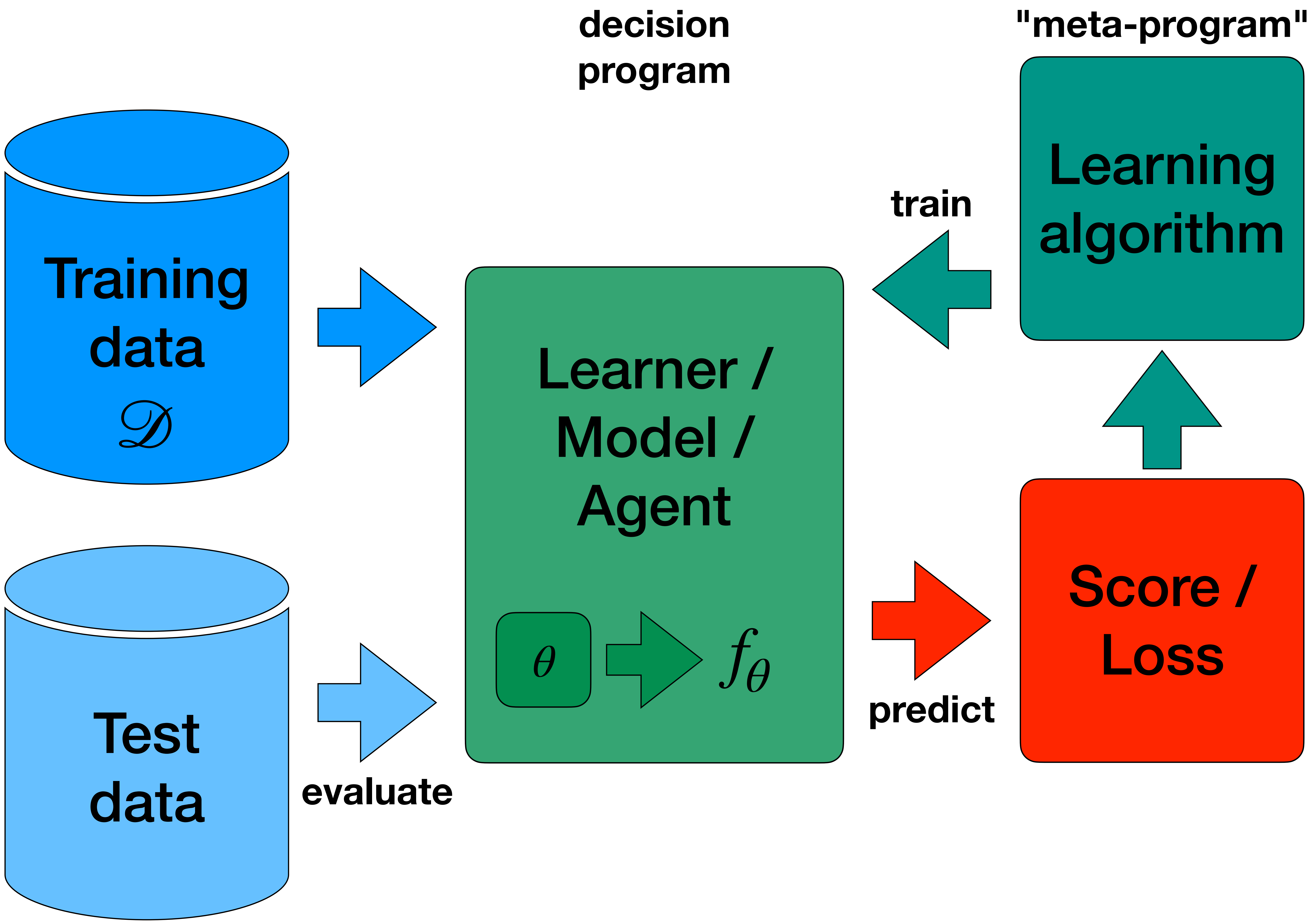
- Given some x, what is a good y?
  - Directly represent $f : x \mapsto y$
  - Average k nearest neighbors

Supervised learning



- Given some x, what is a good y?
  - Directly represent $f : x \mapsto y$
  - Average k nearest neighbors (k too large: missing trend; k too small: catching noise)

What is machine learning?



Upcoming...

logistics

- Join piazza for announcements and forum
- See website for est. schedule

assignments

- Assignment 1 to be published soon
- Meanwhile, get familiar with Python + numpy