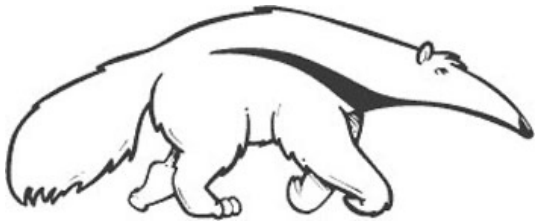
+

# Machine Learning and Data Mining

## Introduction

Prof. Alexander Ihler



# Artificial Intelligence (AI)

- Building “intelligent systems”
- Lots of parts to intelligent behavior

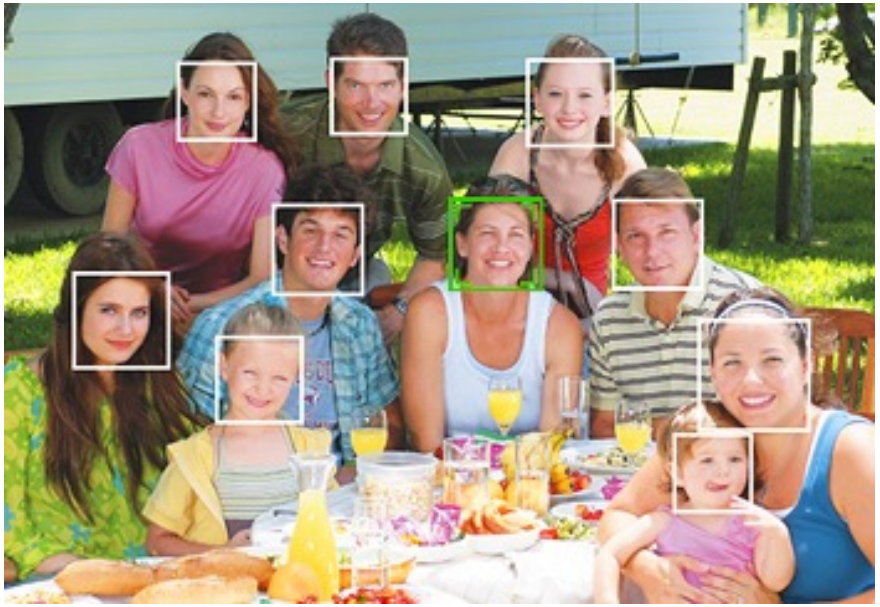

Darpa GC  (Stanley)

RoboCup

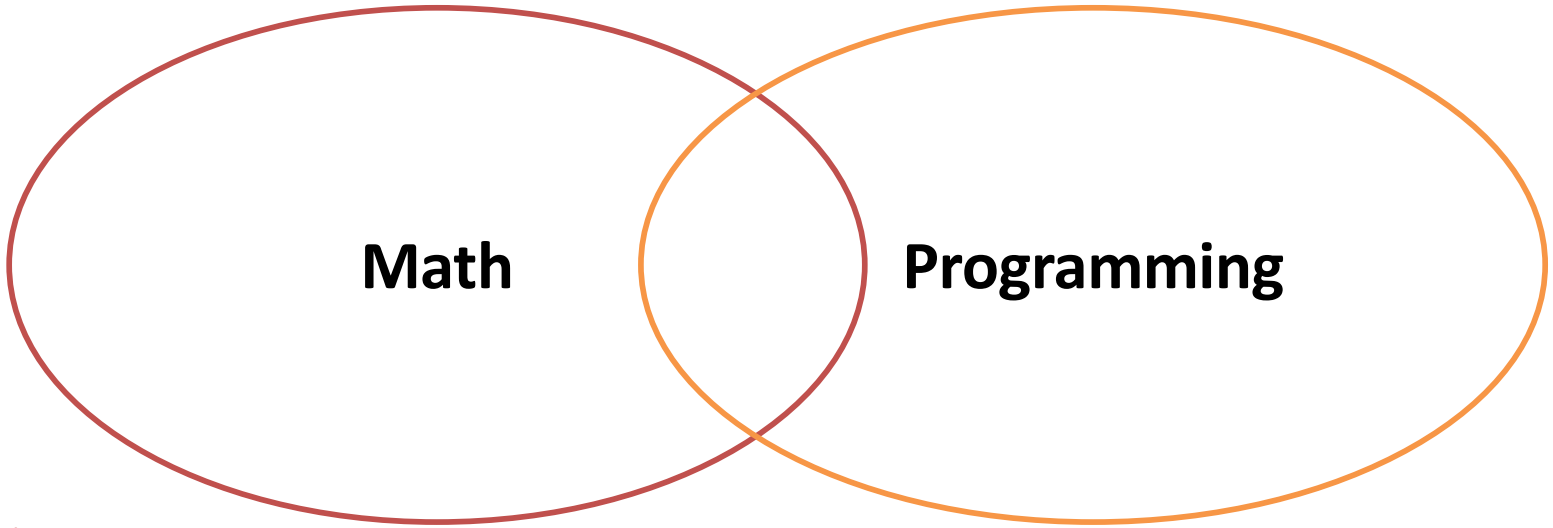

Chess (Deep Blue v. Kasparov)

# Machine learning (ML)

- One (important) part of AI
- Making predictions (or decisions)
- Getting better with experience (data)
- Problems whose solutions are “hard to describe”



# CS178: Machine Learning & Data Mining

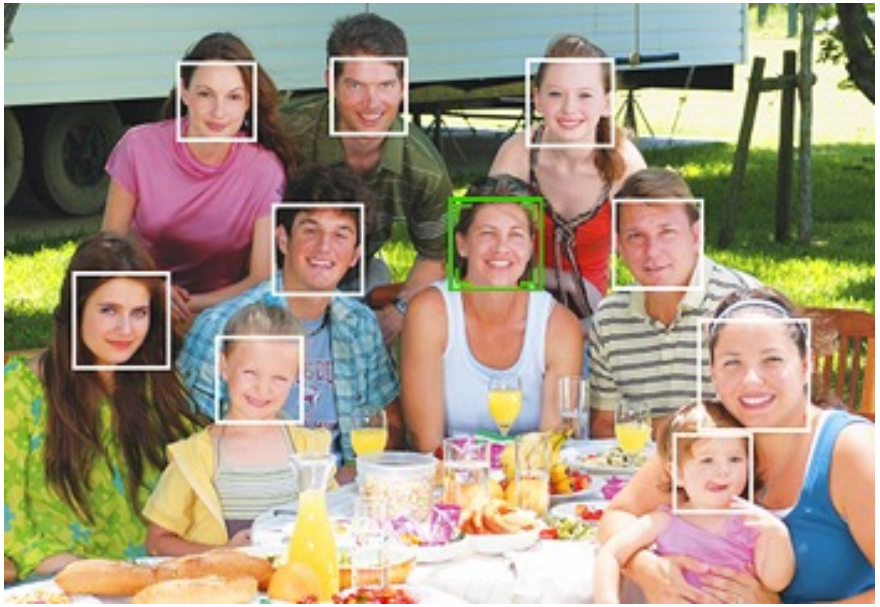



**Statistics,**
**Probability,**
**Linear Algebra,**
**Optimization**

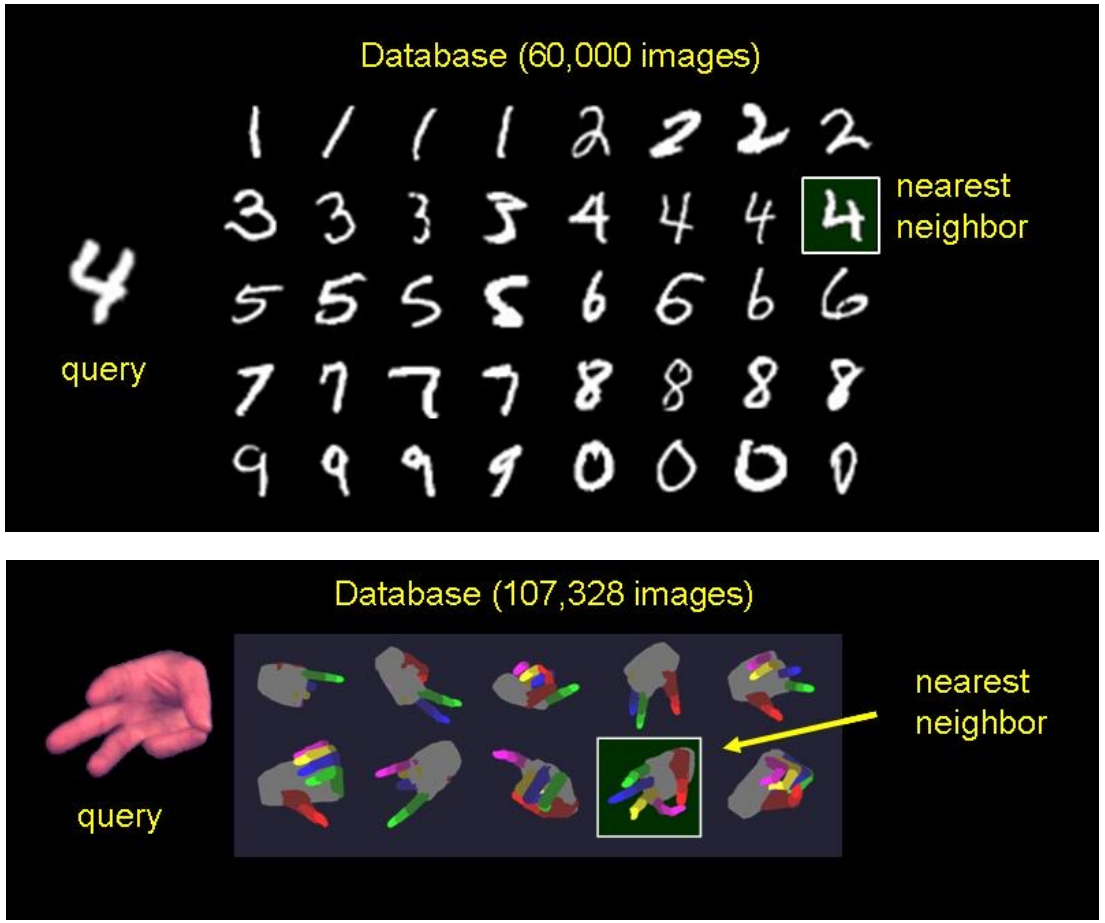
**Data Structures,**
**Algorithms,**
**Computational Complexity,**
**Data Management**

# Types of prediction problems

- Supervised learning
  - "Labeled" training data
  - Every example has a desired target value (a "best answer")
  - Reward prediction being close to target
  - **Classification:** a discrete-valued prediction (often: action / decision)
  - **Regression:** a continuous-valued prediction



# Digit & Hand Gesture Recognition

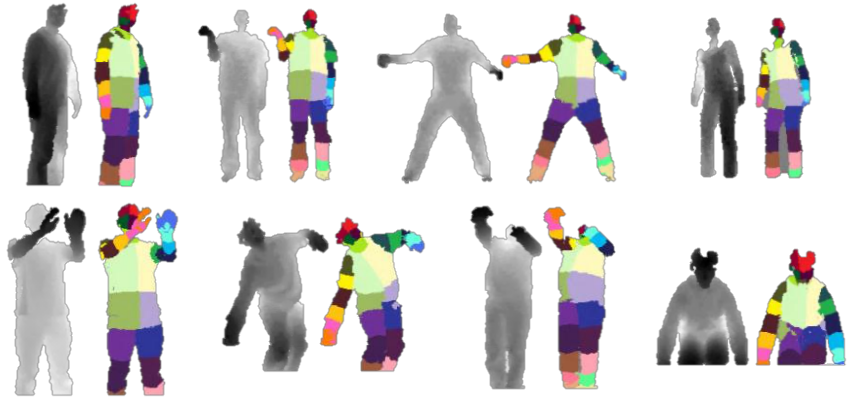



*Athitsos et al., CVPR 2004 & PAMI 2008*

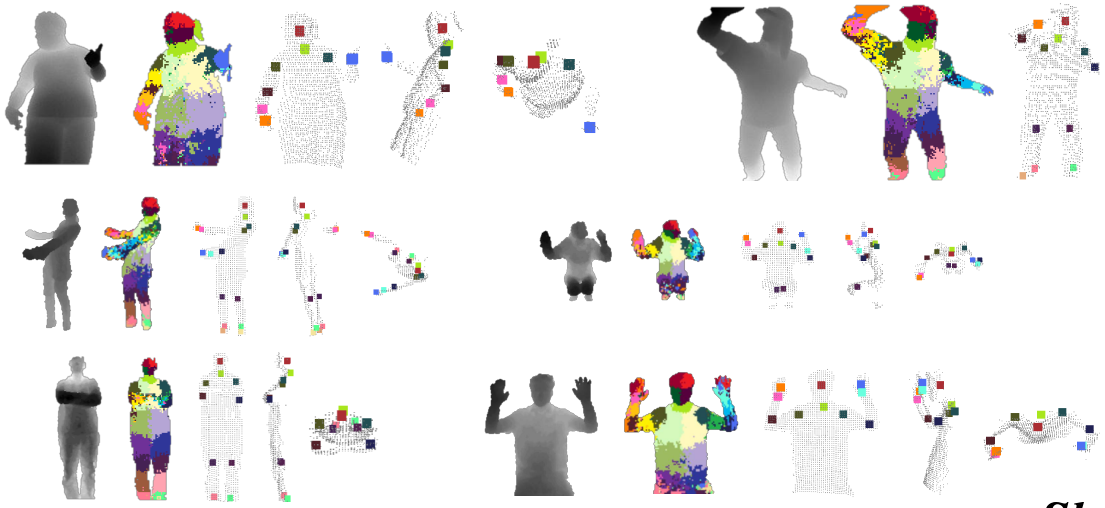
# Microsoft Kinect Pose Estimation

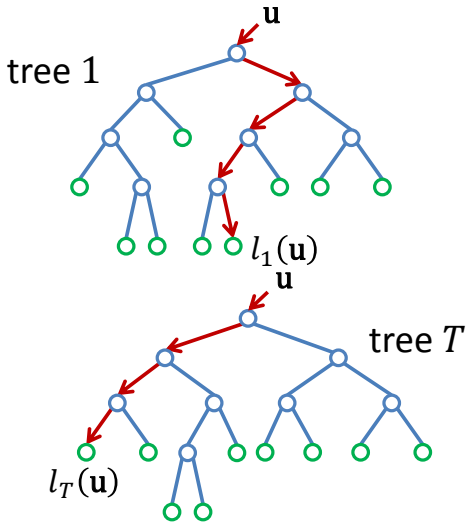
synthetic (train & test)

real (test)



body part classification



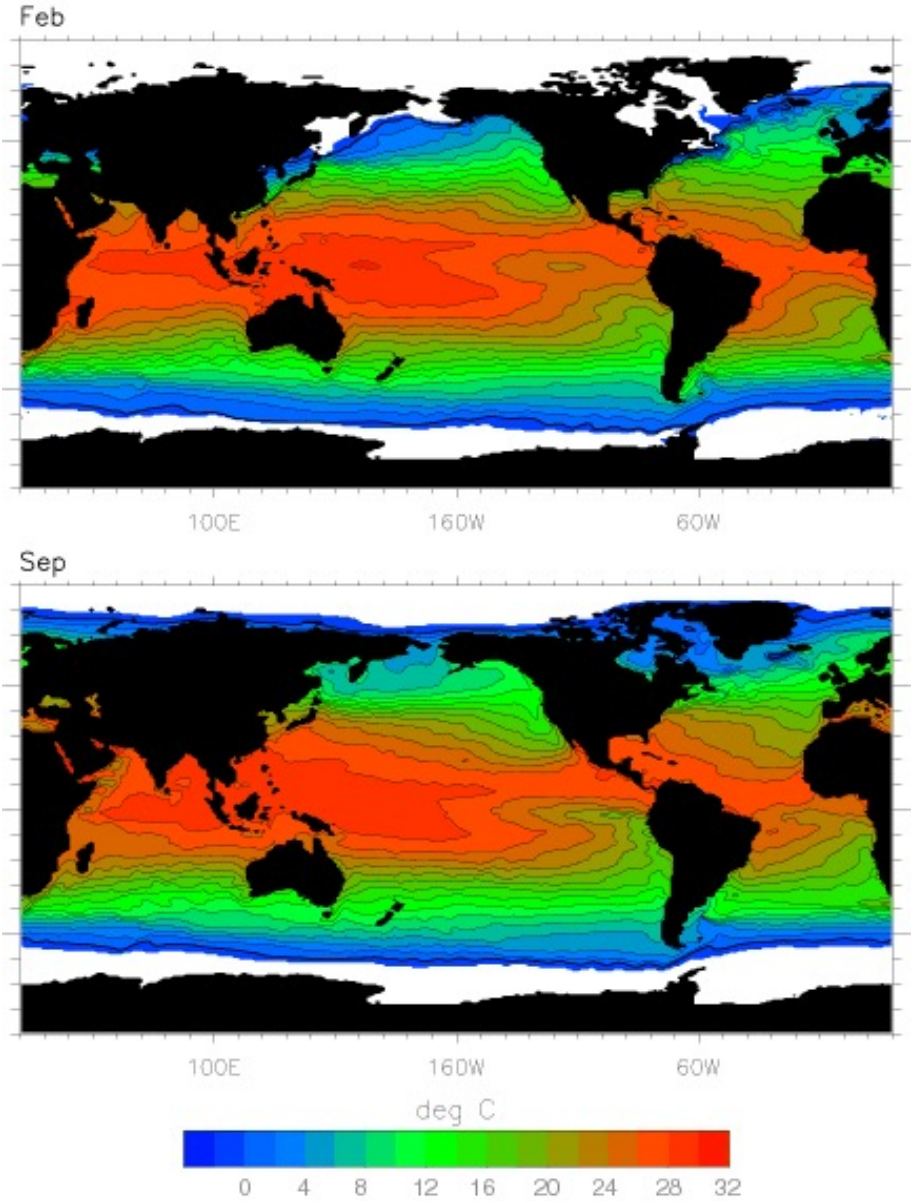




*Shotton et al., PAMI 2012*

# Climate Modeling

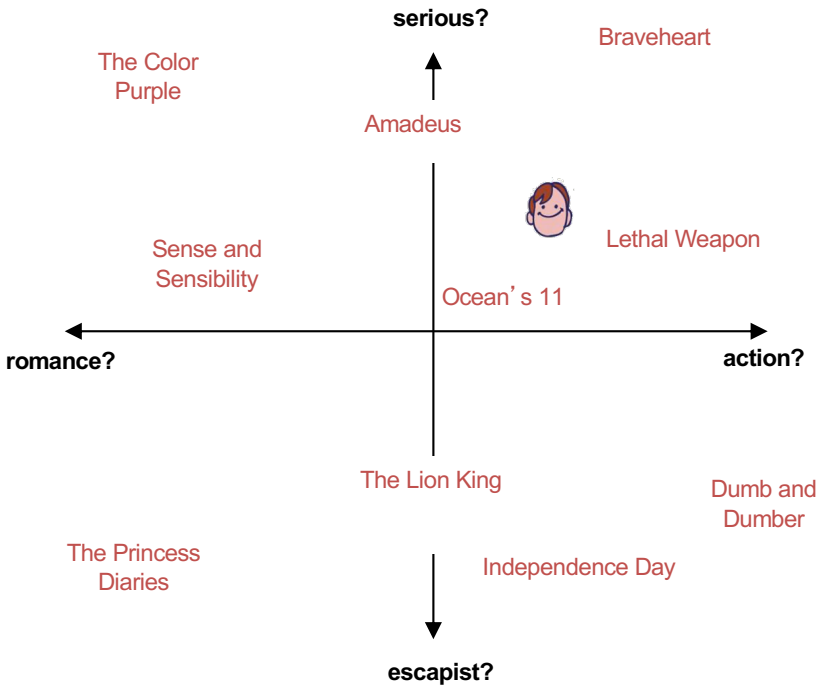
- Satellites measure sea-surface temperature at sparse locations
  - Noisy (atmosphere, sensors)
  - Partial coverage (satellite tracks, clouds)
- Want to infer a dense temperature field, and track its evolution over time

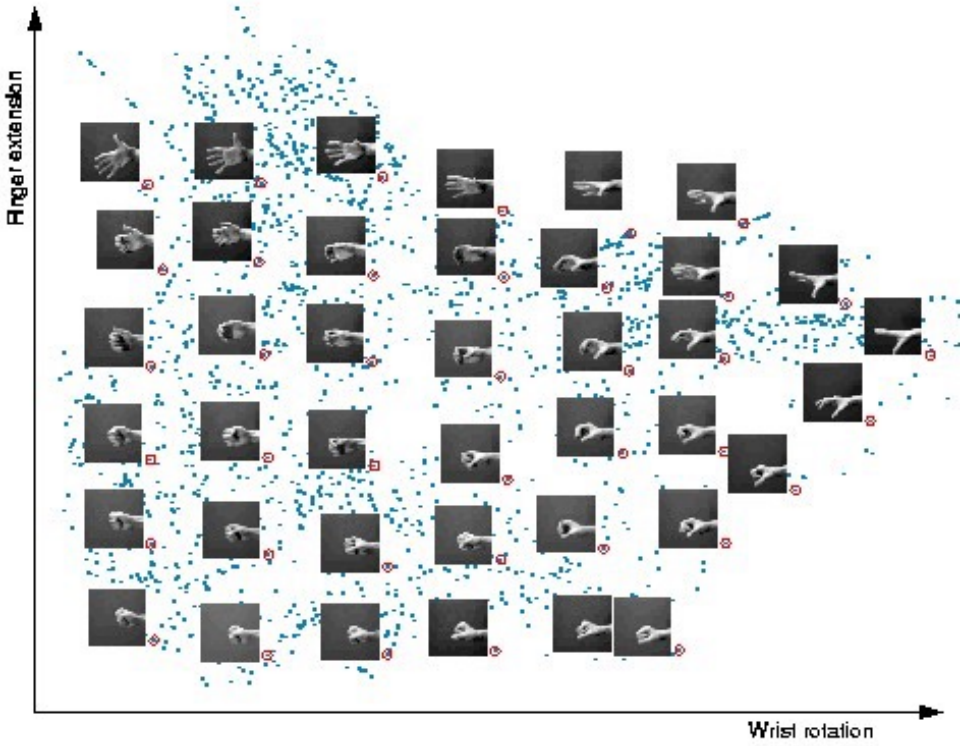



***NASA Seasonal to Interannual Prediction Project***
***http://ct.gsfc.nasa.gov/annual.reports/ess98/nsipp.html***

# Types of prediction problems

- Supervised learning
- Unsupervised learning
  - No known target values
  - No targets = nothing to predict?
  - Reward “patterns” or “explaining features”
  - Often, data mining



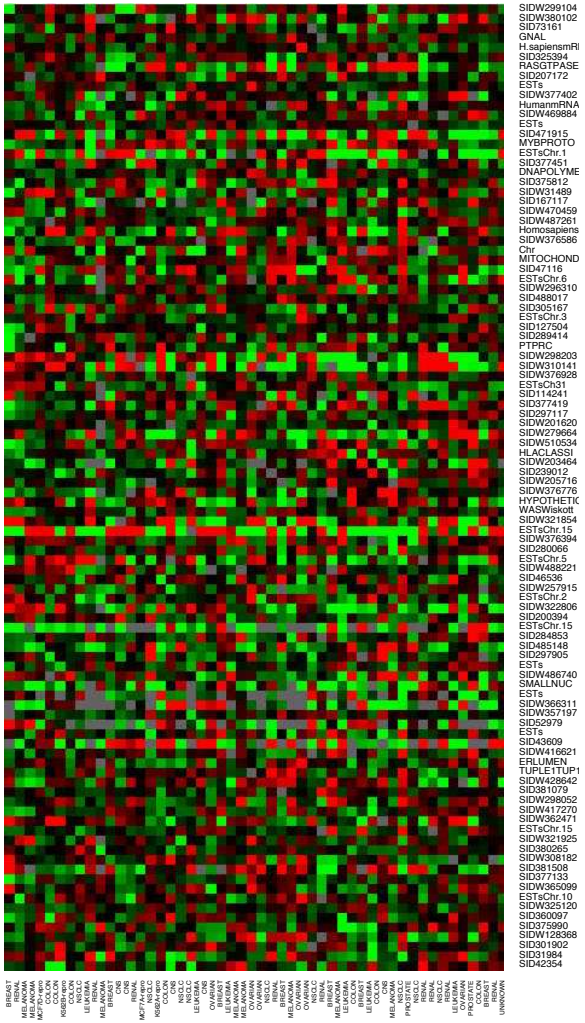
# Human Tumor Microarray Data

- 6830 x 64 matrix of real numbers
- Rows = genes
- Columns = tissue samples

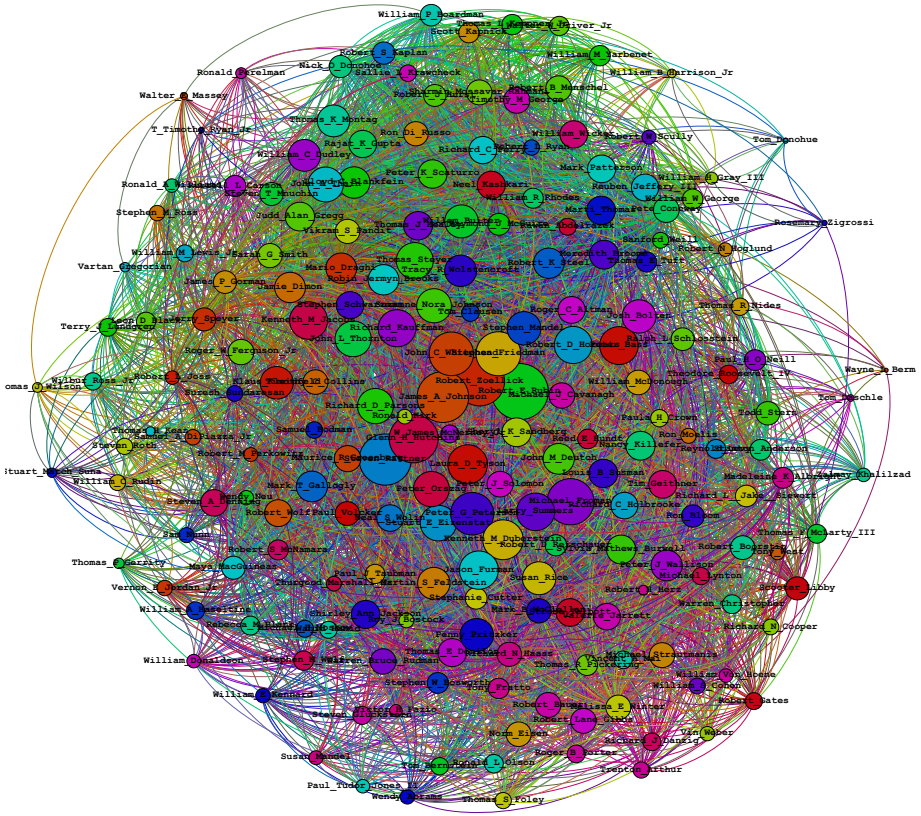
- Cluster rows (genes) to deduce function of unknown genes from experimentally known genes with similar profiles

- Cluster columns (samples) to hypothesize disease profiles

***Hastie, Tibshirani, & Friedman 2009***



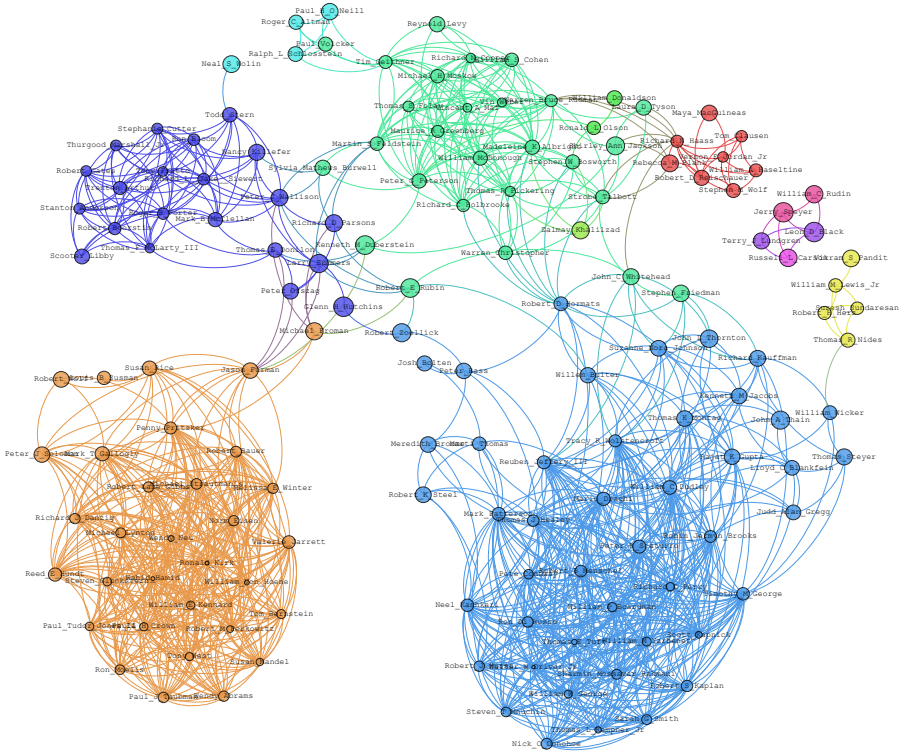
# Social Networks & Relationships



*Kim et al., NIPS 2013*

**LittleSis* is a free database of who-knows-who at the heights of business and government.**

* opposite of Big Brother



# Types of prediction problems

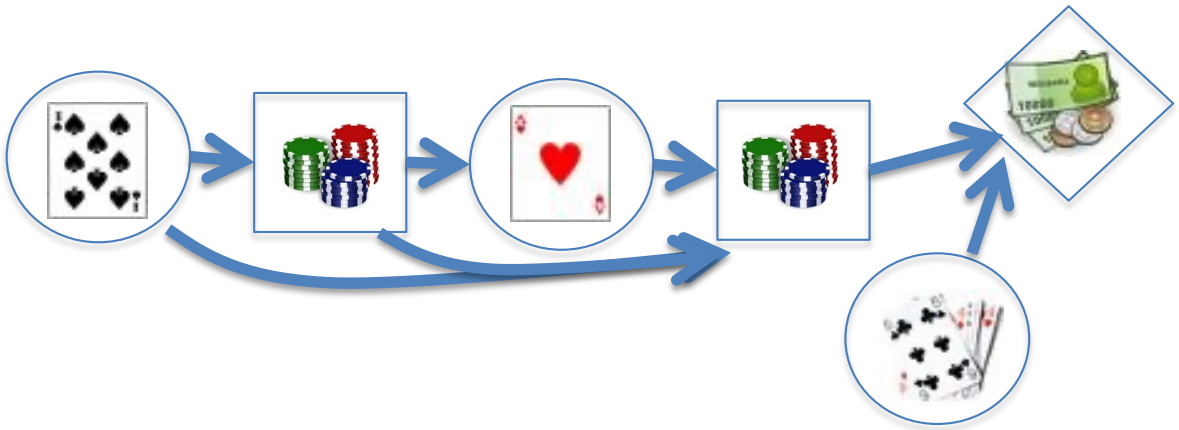
- Supervised learning
- Unsupervised learning
- Semi-supervised learning
  - Similar to supervised
  - some data have unknown target values

- Ex: medical data
  - Lots of patient data, few known outcomes
- Ex: image tagging
  - Lots of images on Flikr, but only some of them tagged

# Types of prediction problems

- Supervised learning
- Unsupervised learning
- Semi-supervised learning
- Reinforcement learning

- "Indirect" feedback on quality
  - No answers, just "better" or "worse"
  - Feedback may be delayed



# Issues to Understand

- Given two candidate models, which is better?
  - Accuracy at predicting training data?
  - Complexity of classification or regression function?
  - Are all mistakes equally bad?

- Given a family of classifiers with free parameters, which member of that family is best?
  - Are there general design principles?
  - What happens as I get more data?
  - Can I test all possible classifiers?
  - What if there are lots of parameters?

***Probability & Statistics***

***Algorithms & Linear Algebra***

# Machine Learning

Introduction to Machine Learning

**Course Logistics**

Data and Visualization

Supervised Learning

# Logistics

**Canvas page: https://canvas.eee.uci.edu/courses/39958**
*Course information, homework submission, grading.*

**Ed Discussion: https://edstem.org/us/courses/14090**
*For all questions and discussion of course material!*

**No required textbook:**

- All necessary information covered in lectures.
- Supplemental notes available for some topics.
- Recommended references on Canvas.

# Instruction

| Course Staff | |
|---|---|
| Instructor | Prof. Alex Ihler |
| TAs | Tiancheng Xu |
| | Shanlin Sun |
| Readers | Kyungmin Kim |
| | Ziheng Duan |
| | Junchen Zhao |

| Teaching | | |
|---|---|---|
| Lecture | M/W/F 10-11am | SSLH 100 + Zoom |
| Discussions | Fri 11am & 12pm<br>Fri 1pm & 2pm | ICS 174<br>SSL 228 |
| Office Hours | See Canvas | |

# Programming Assignments

*Homework 1 due Oct 3, released today.*

**5 Programming Assignments**

- We will drop lowest grade

**Objective**

- Learn to apply ML techniques
- Submission is a “report”

**Source Code (Python)**

- Submit relevant code snippets
  - We will not run it, but will read it
- Statement of collaboration, if any
  - Only limited discussions allowed

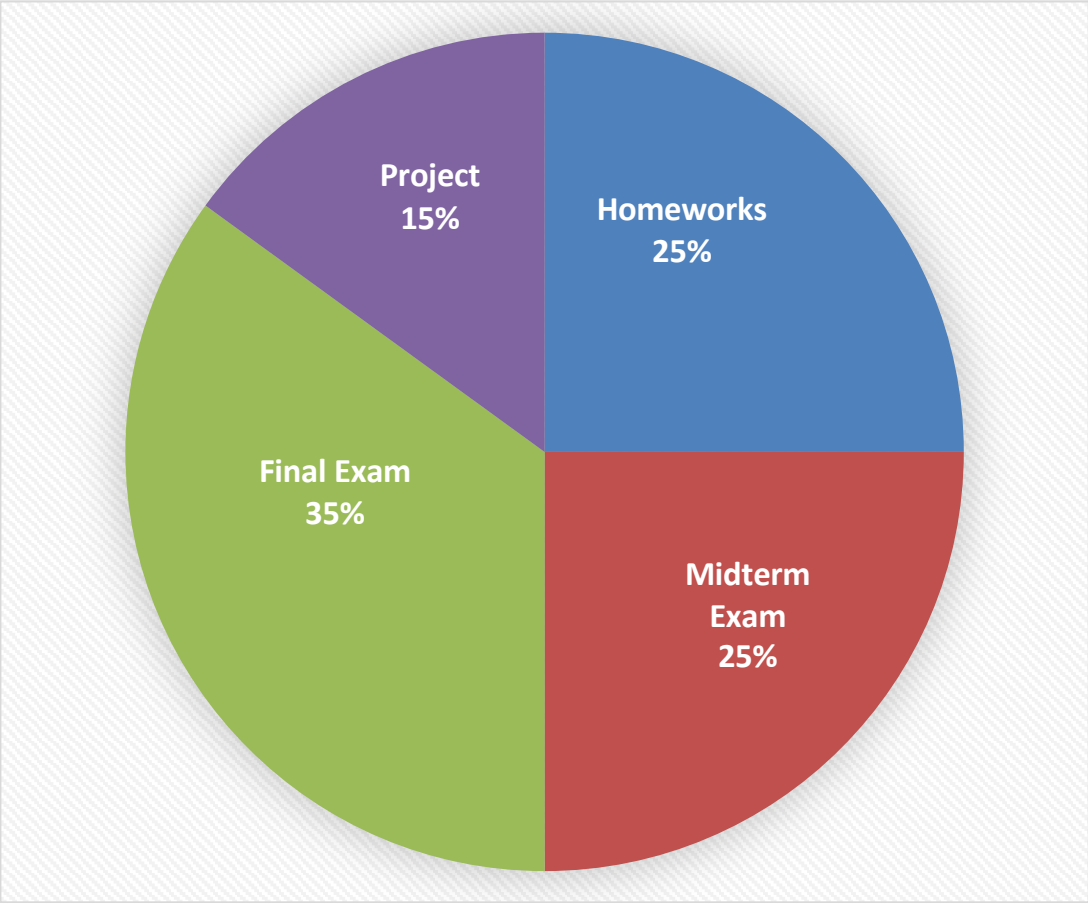
# Project


**Groups for the Project**

- **Team size up to 3 members**
  - **Larger teams not allowed**
  - **Smaller in special cases, with instructor permission**
- **More details coming later (see also Canvas)**
- **Report (two pages) due at the end of the quarter**

# Grading

**(Note: may be subject to change, i.e., due to COVID.)**




- **Midterm: During normal lecture time on Wed Nov 3 (tentative!)**
- **Final: Mon Dec 6, from 10:30am-12:30pm.**
- ***No rescheduling except in extraordinary, unexpected circumstances!***
- **Final lecture will be on Fri December 3.**

# Machine Learning

Introduction to Machine Learning

Course Logistics

**Data and Visualization**

Supervised Learning

# Data exploration

- Machine learning is a data science
  - Look at the data; get a “feel” for what might work
- What types of data do we have?
  - Binary values? (spam; gender; …)
  - Categories? (home state; labels; …)
  - Integer values? (1..5 stars; age brackets; …)
  - (nearly) real values? (pixel intensity; prices; …)
- Are there missing data?
- “Shape” of the data? Outliers?

# Scientific software

- **Python**
  - **Numpy, MatPlotLib, SciPy…**
- Matlab
  - Octave (free)
- R
  - Used mainly in statistics
- C++
  - For performance, not prototyping
- And other, more specialized languages for modeling…

# Representing data

- Example: Fisher's "Iris" data

  http://en.wikipedia.org/wiki/Iris_flower_data_set

- Three different types of iris
  - "Class", y
- Four "features", $x_1,\ldots,x_4$
  - Length & width of sepals & petals
- 150 examples (data points)

# Representing the data in Python

- Have m observations (data points)

$$\left\{x^{(1)} \ldots, x^{(m)}\right\}$$

- Each observation is a vector consisting of n features

$$x^{(j)} = [x_1^{(j)} x_2^{(j)} \ldots x_n^{(j)}]$$

- Often, represent this as a "data matrix"

$$\underline{X} = \begin{bmatrix} x_1^{(1)} & \ldots & x_n^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(m)} & \ldots & x_n^{(m)} \end{bmatrix}$$

```
import numpy as np    # import numpy
iris = np.genfromtxt("data/iris.txt",delimiter=None)
X = iris[:,0:4]       # load data and split into features, targets
Y = iris[:,4]
print X.shape         # 150 data points; 4 features each
   (150, 4)
```

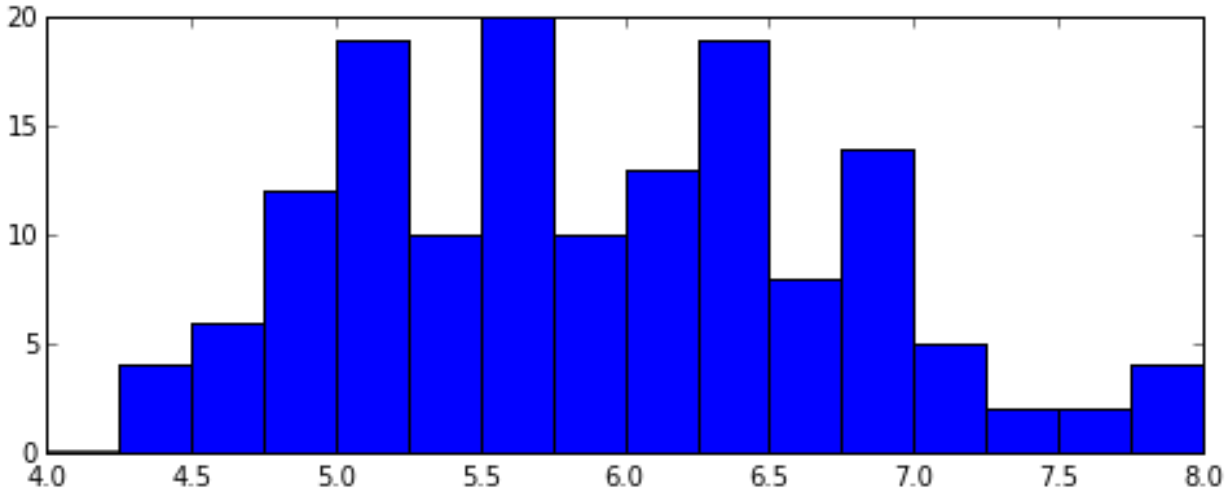
# Basic statistics

- Look at basic information about features
  - Average value? (mean, median, etc.)
  - "Spread"? (standard deviation, etc.)
  - Maximum / Minimum values?

```
print np.mean(X, axis=0)          # compute mean of each feature
  [ 5.8433    3.0573    3.7580    1.1993 ]
print np.std(X, axis=0)           #compute standard deviation of each feature
  [ 0.8281    0.4359    1.7653    0.7622 ]
print np.max(X, axis=0)           # largest value per feature
  [  7.9411    4.3632    6.8606    2.5236 ]
print np.min(X, axis=0)           # smallest value per feature
  [  4.2985    1.9708    1.0331    0.0536  ]
```

# Histograms

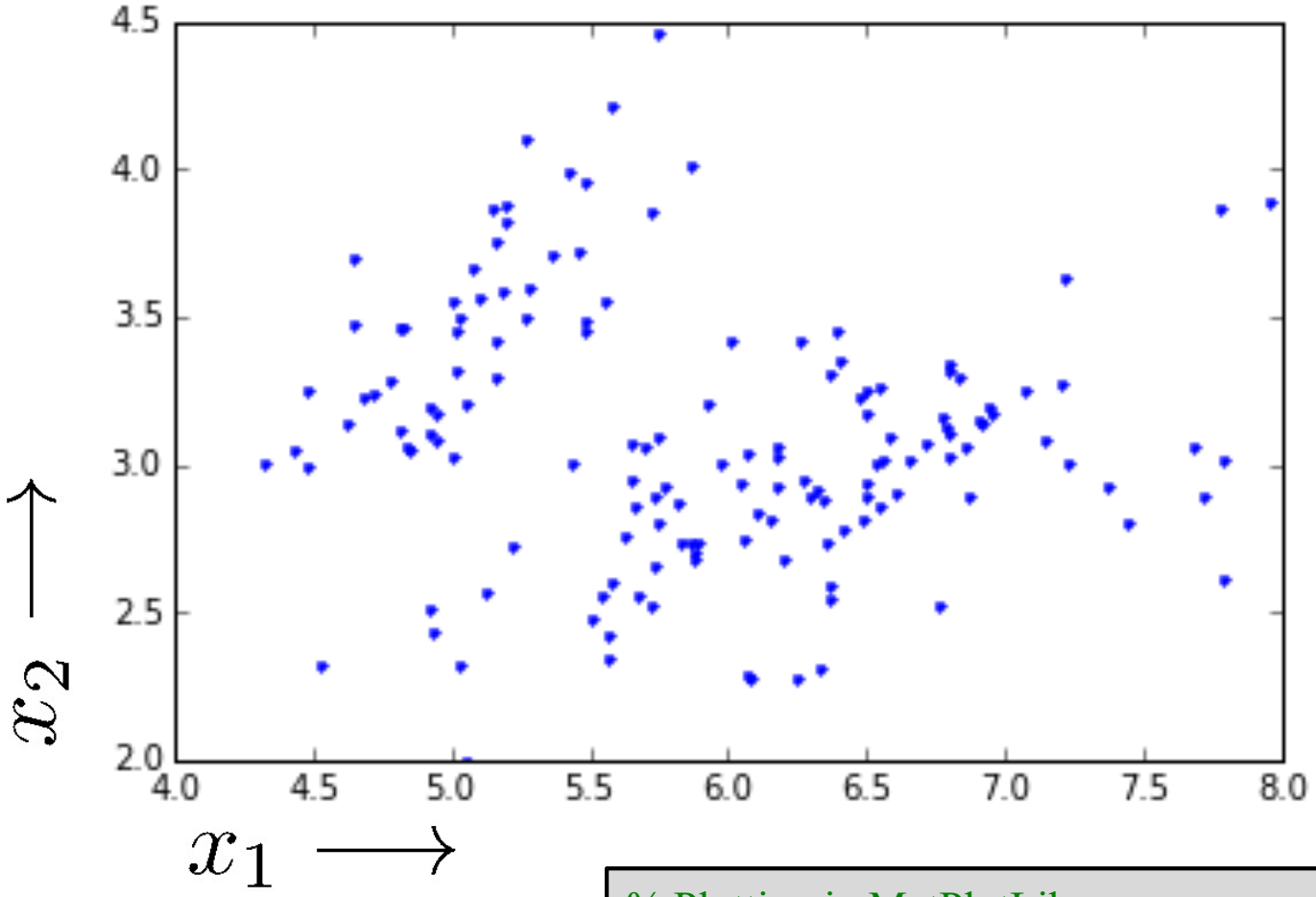
- Count the data falling in each of K bins
  - "Summarize" data as a length-K vector of counts (& plot)
  - Value of K determines "summarization"; depends on # of data
    - K too big: every data point falls in its own bin; just "memorizes"
    - K too small: all data in one or two bins; oversimplifies




```
% Histograms in MatPlotLib
import matplotlib.pyplot as plt
X1 = X[:,0]                    # extract first feature
Bins = np.linspace(4,8,17)     # use explicit bin locations
plt.hist( X1, bins=Bins )      # generate the plot
```

# Scatterplots

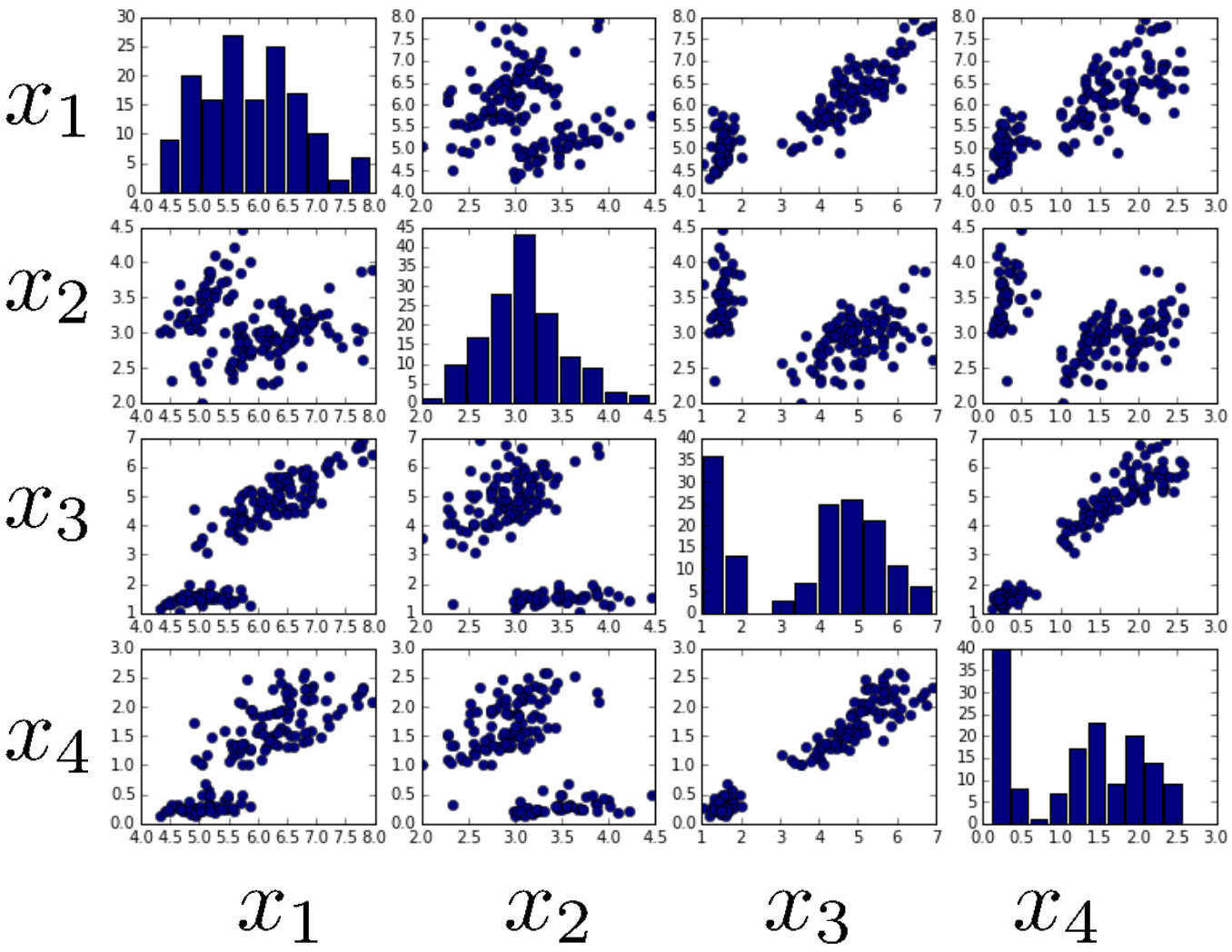
- Illustrate the relationship between two features




```
% Plotting in MatPlotLib
plt.plot(X[:,0], X[:,1], 'b.');   % plot data points as blue dots
```

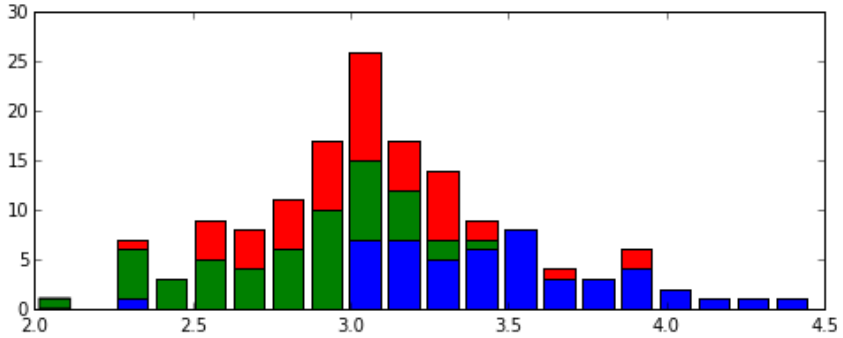
# Scatterplots

- For more than two features we can use a pair plot:

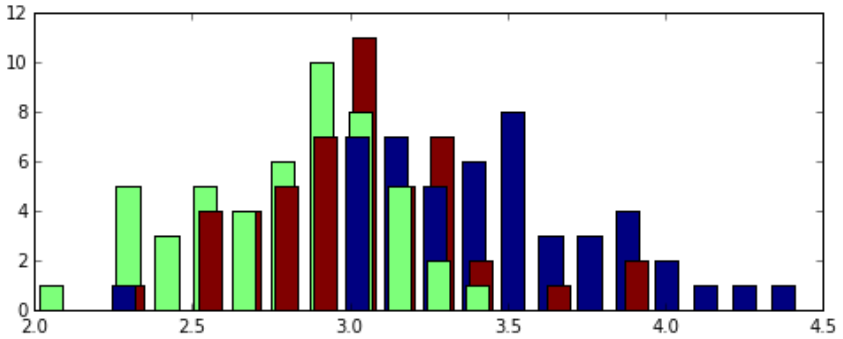


# Supervised learning and targets

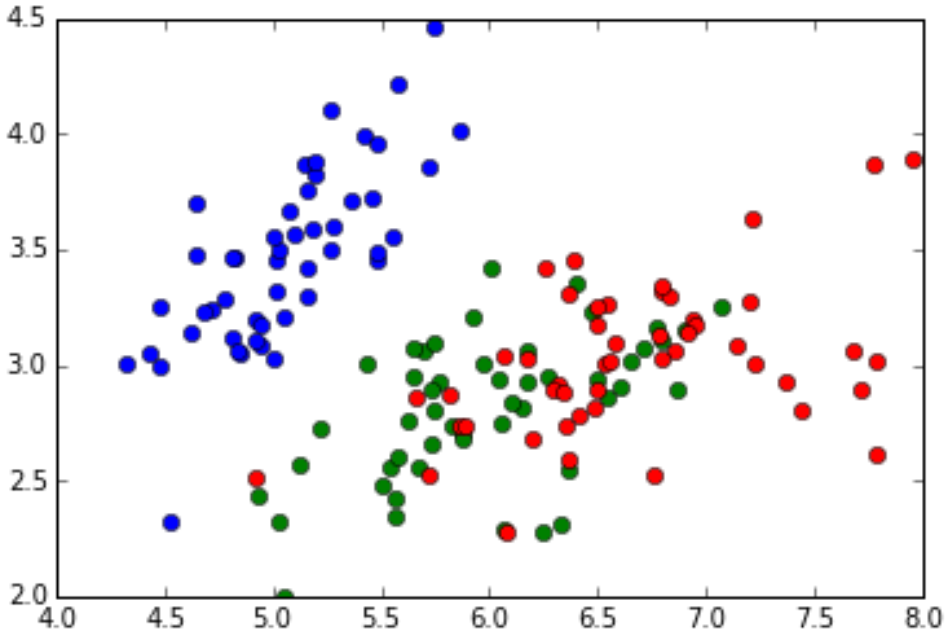
- Supervised learning: predict target values
- For discrete targets, often visualize with color




```
plt.hist( [X[Y==c,1] for c in np.unique(Y)] ,
          bins=20, histtype='barstacked')
```




```
ml.histy(X[:,1], Y, bins=20)
```




```
colors = ['b','g','r']
for c in np.unique(Y):
        plt.plot( X[Y==c,0], X[Y==c,1], 'o',
                  color=colors[int(c)] )
```

# Machine Learning

Introduction to Machine Learning

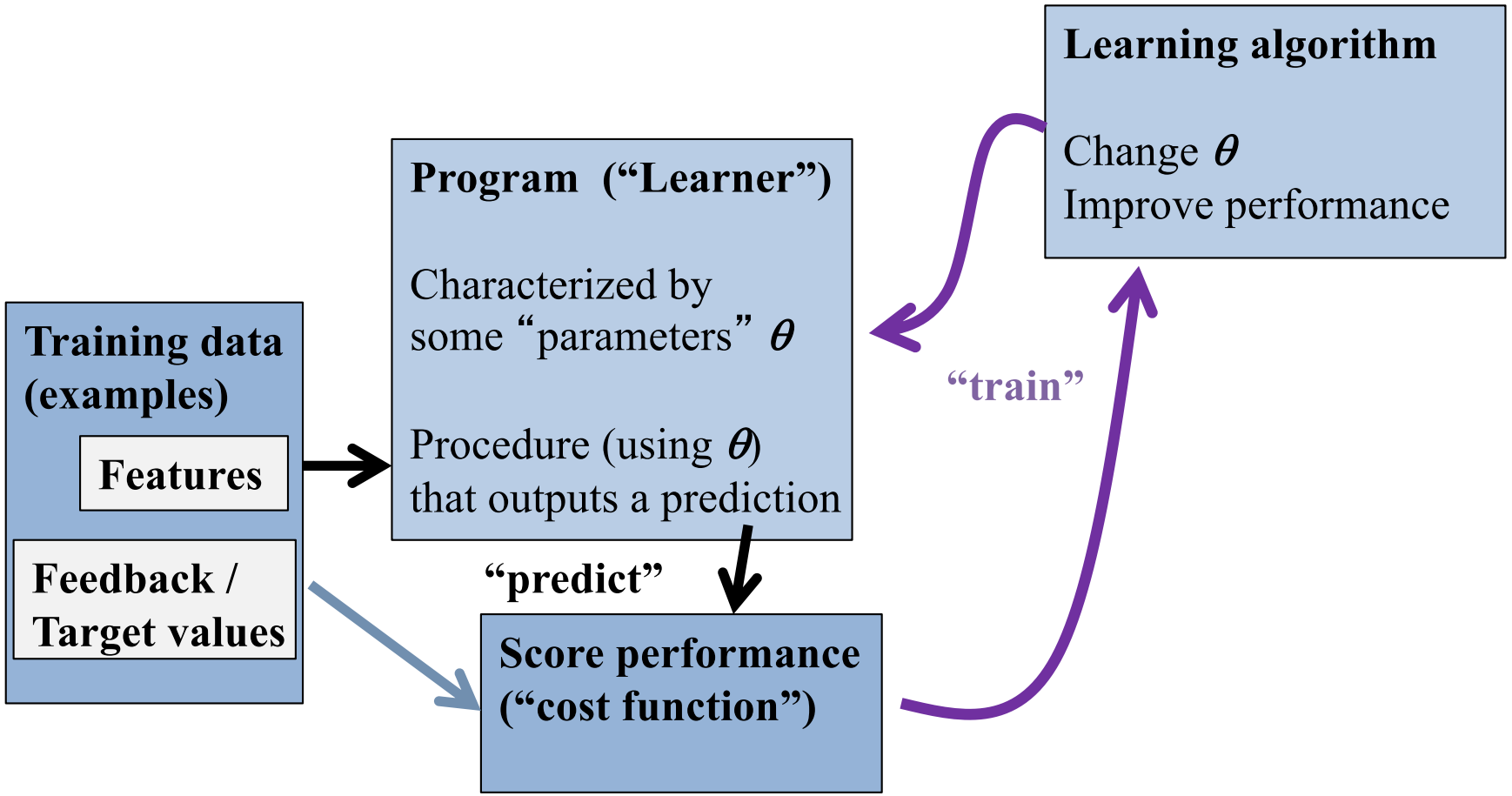
Course Logistics

Data and Visualization

**Supervised Learning**

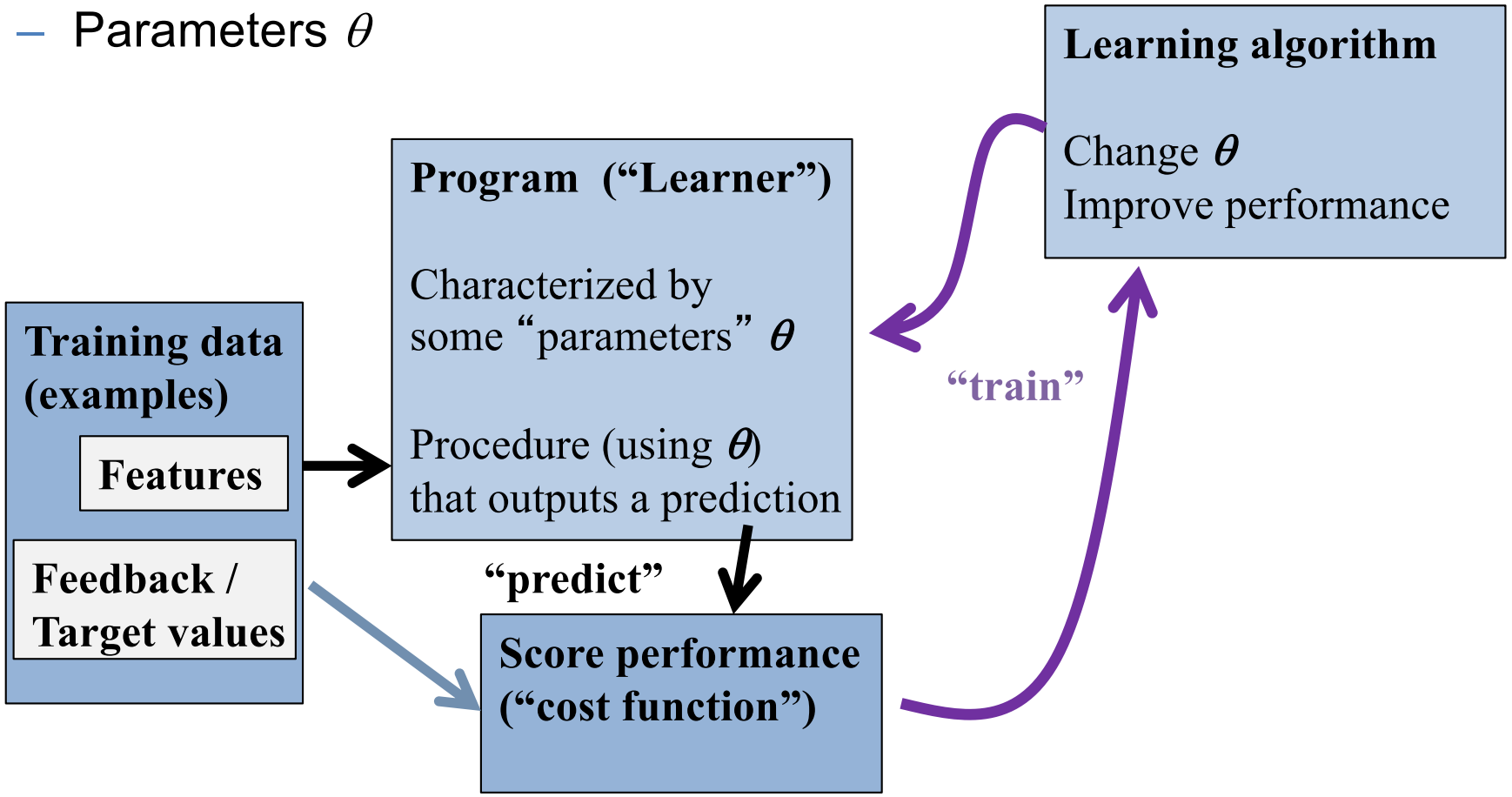
# How does machine learning work?

- "Meta-programming"
  - Predict – apply rules to examples
  - Score – get feedback on performance
  - Learn – change predictor to do better

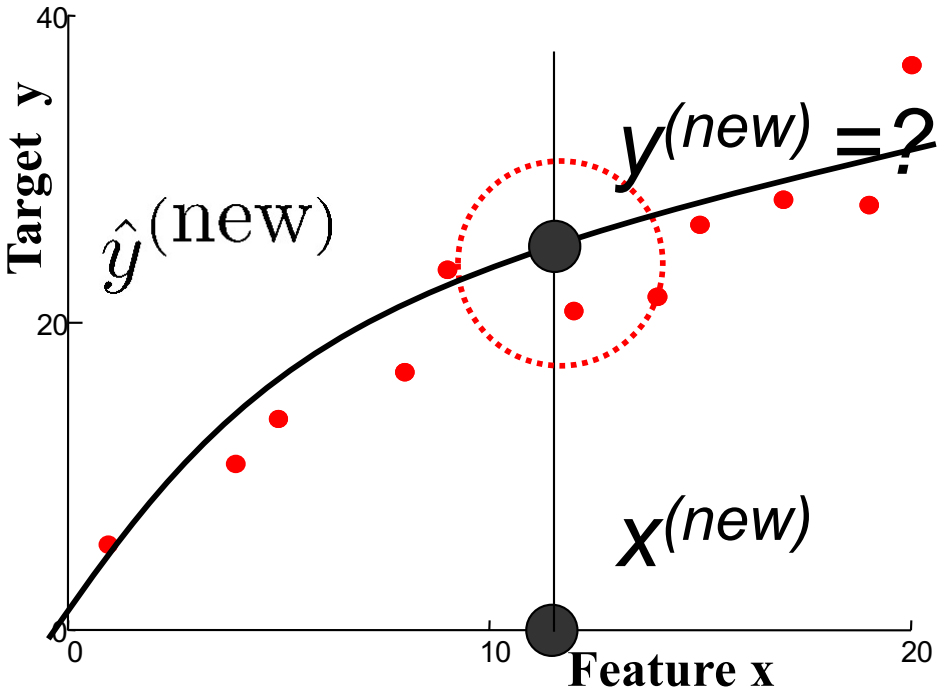


# Supervised learning

- Notation
  - Features $x$
  - Targets $y$
  - Predictions $\hat{y} = f(x\ ;\ \theta)$
  - Parameters $\theta$

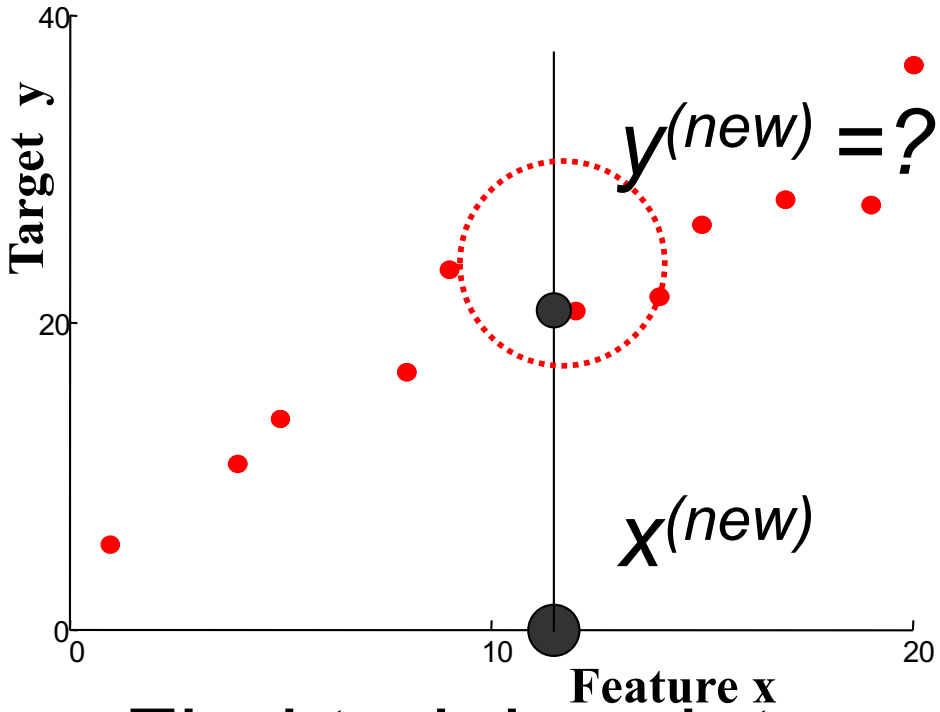


# Regression; Scatter plots



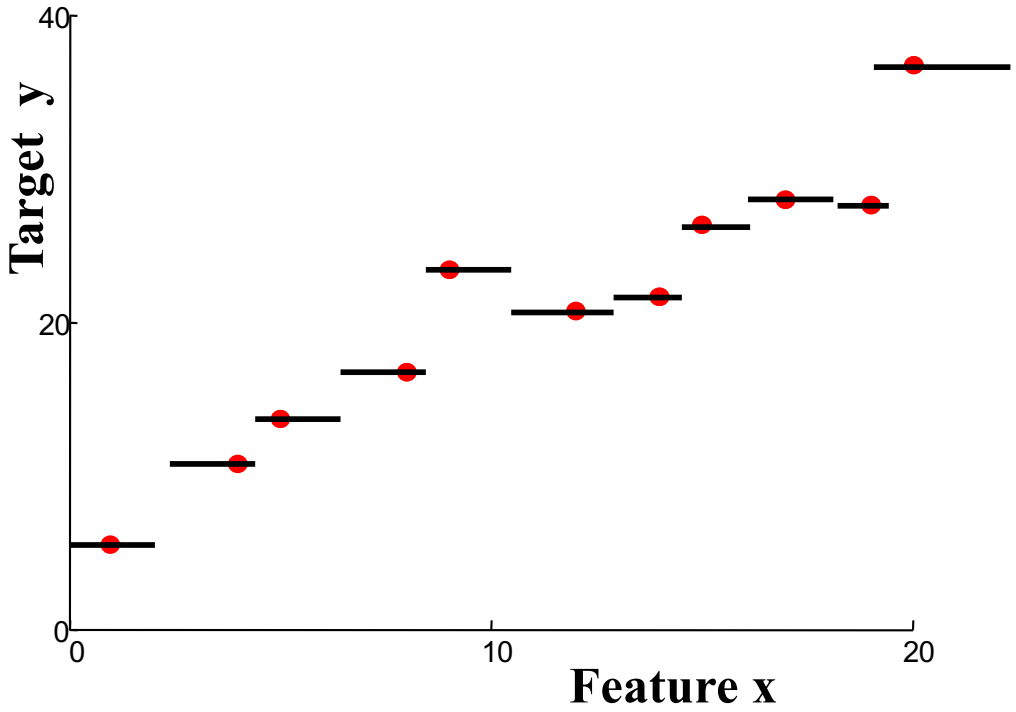

- Suggests a relationship between x and y
- *Prediction*: new x, what is y?

# Nearest neighbor regression




- Find training datum $x^{(i)}$ closest to $x^{(new)}$
  Predict $y^{(i)}$

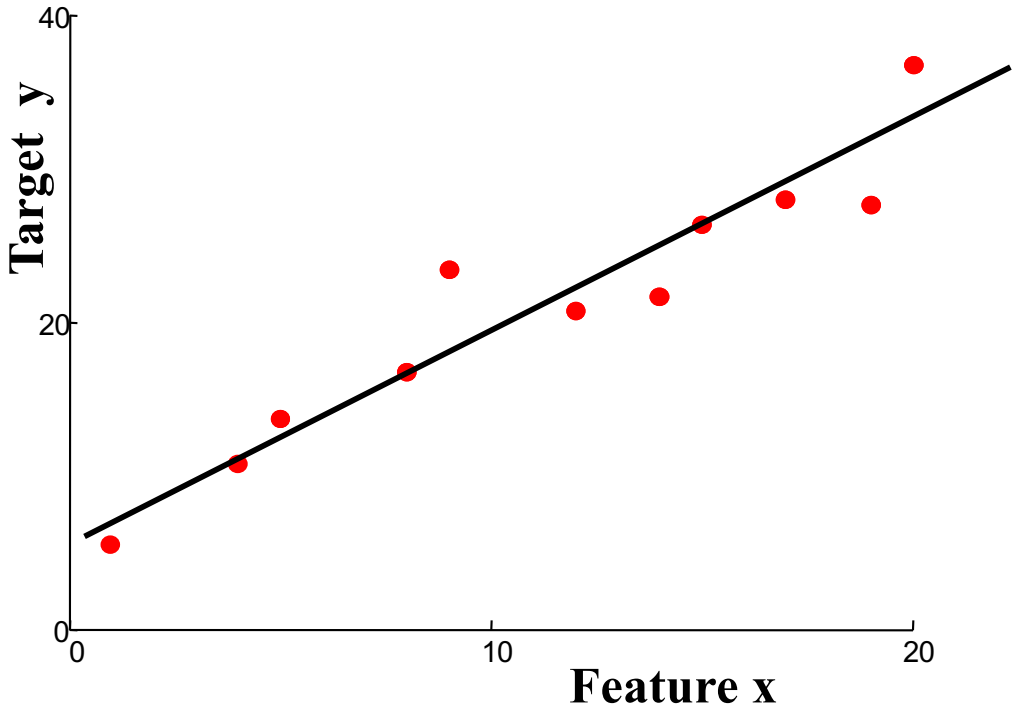
# Nearest neighbor regression




**"Predictor":**
Given new features:
Find nearest example
Return its value

- Defines a function f(x) implicitly
- "Form" is piecewise constant

# Linear regression



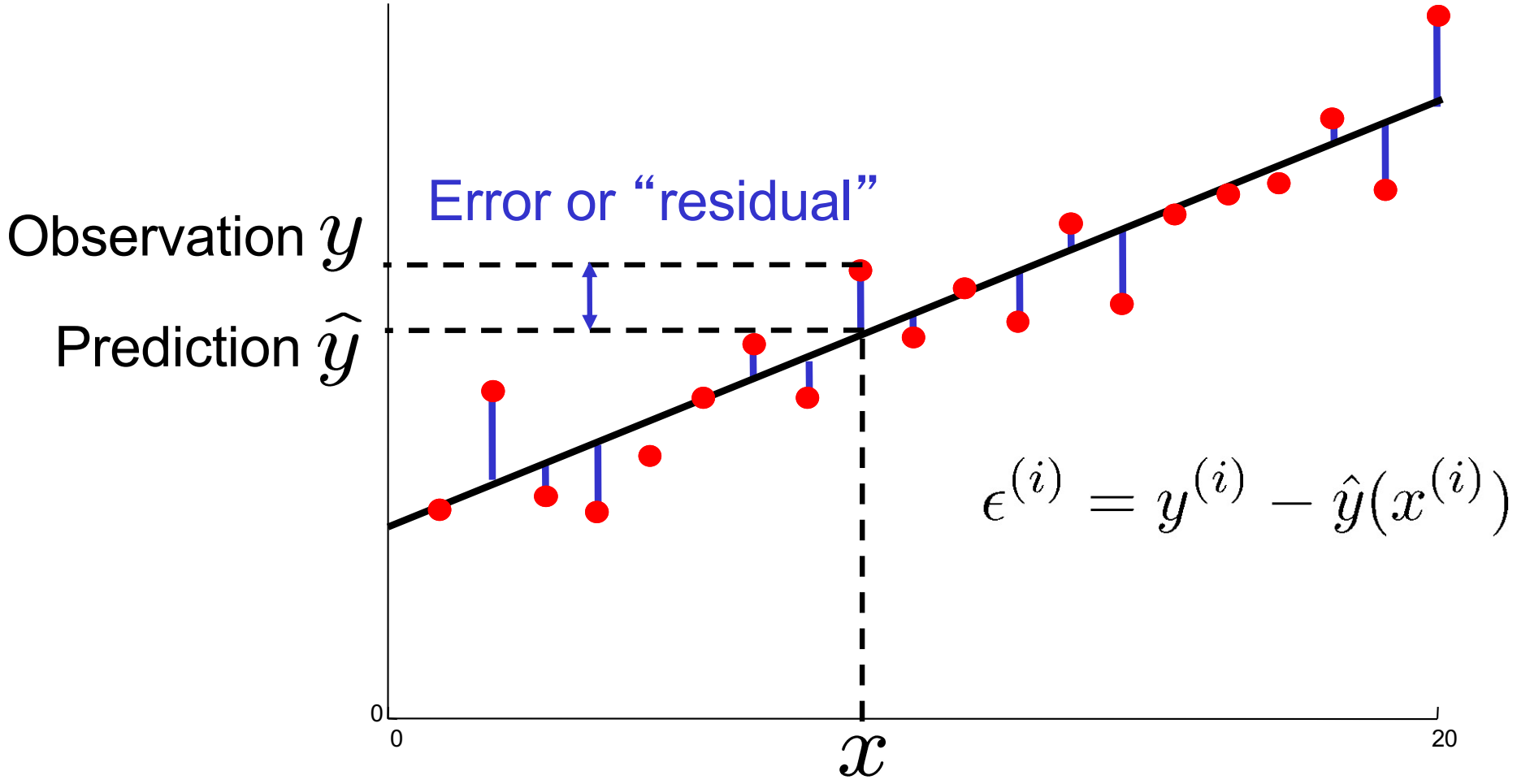

**"Predictor":**
Evaluate line:

$$r = \theta_0 + \theta_1 x_1$$

return r

- Define form of function f(x) explicitly
- Find a good f(x) within that family

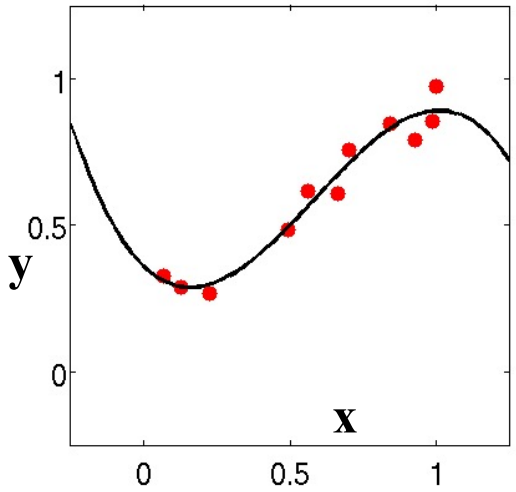
# Measuring error




$$\text{MSE} = \frac{1}{m} \sum_i \left( y^{(i)} - \hat{y}(x^{(i)}) \right)^2$$

# Regression vs. Classification

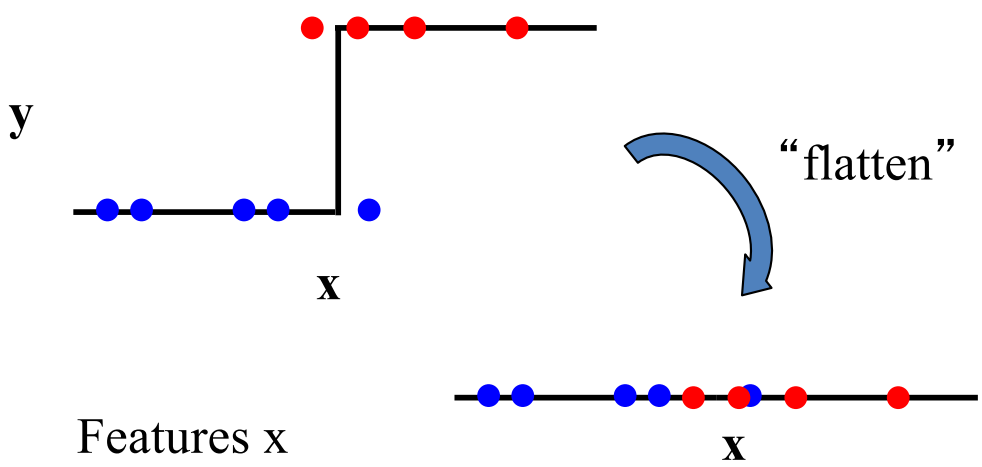
**Regression**




Features x
Real-valued target y

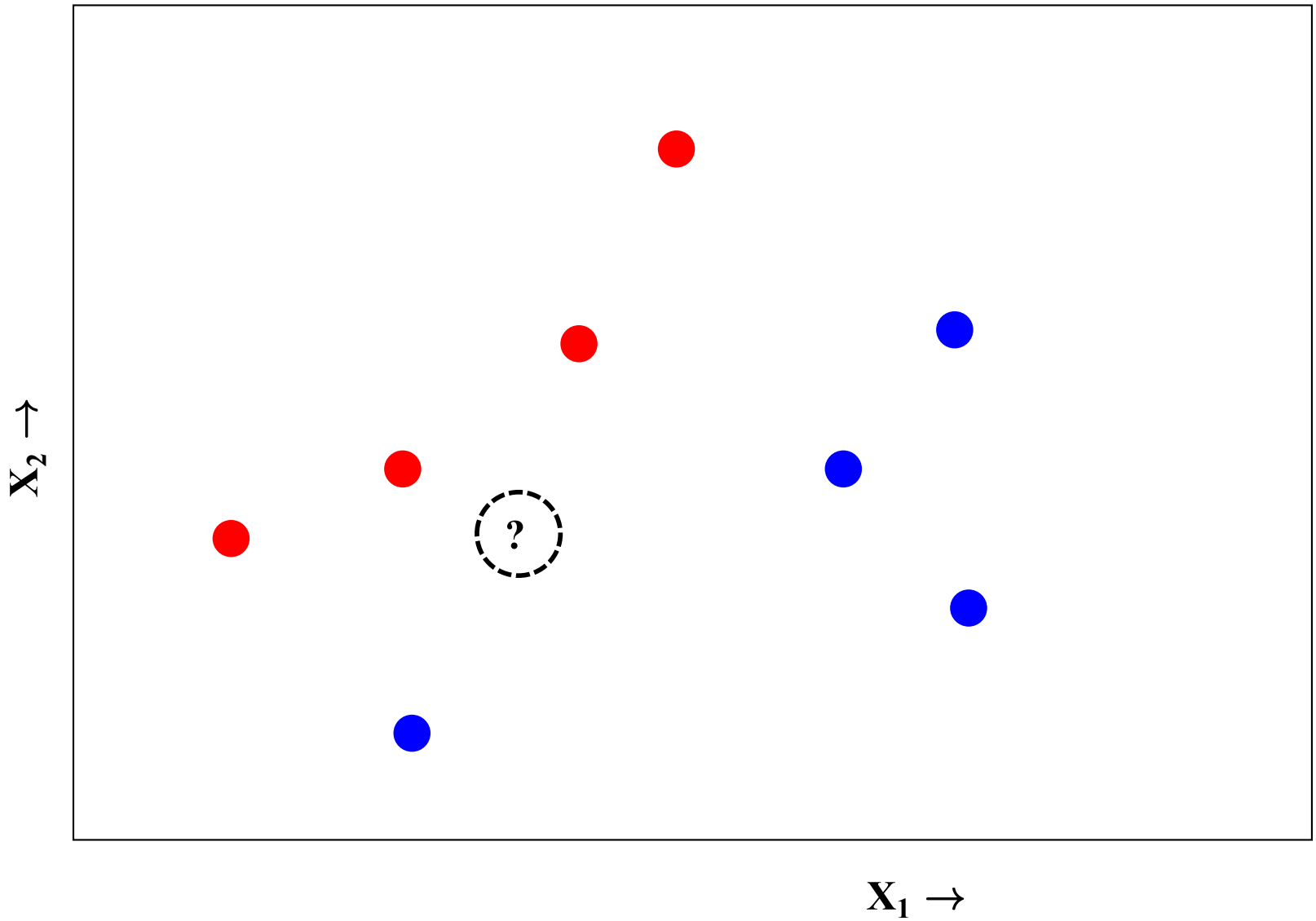
Predict continuous function $\hat{y}(x)$

**Classification**



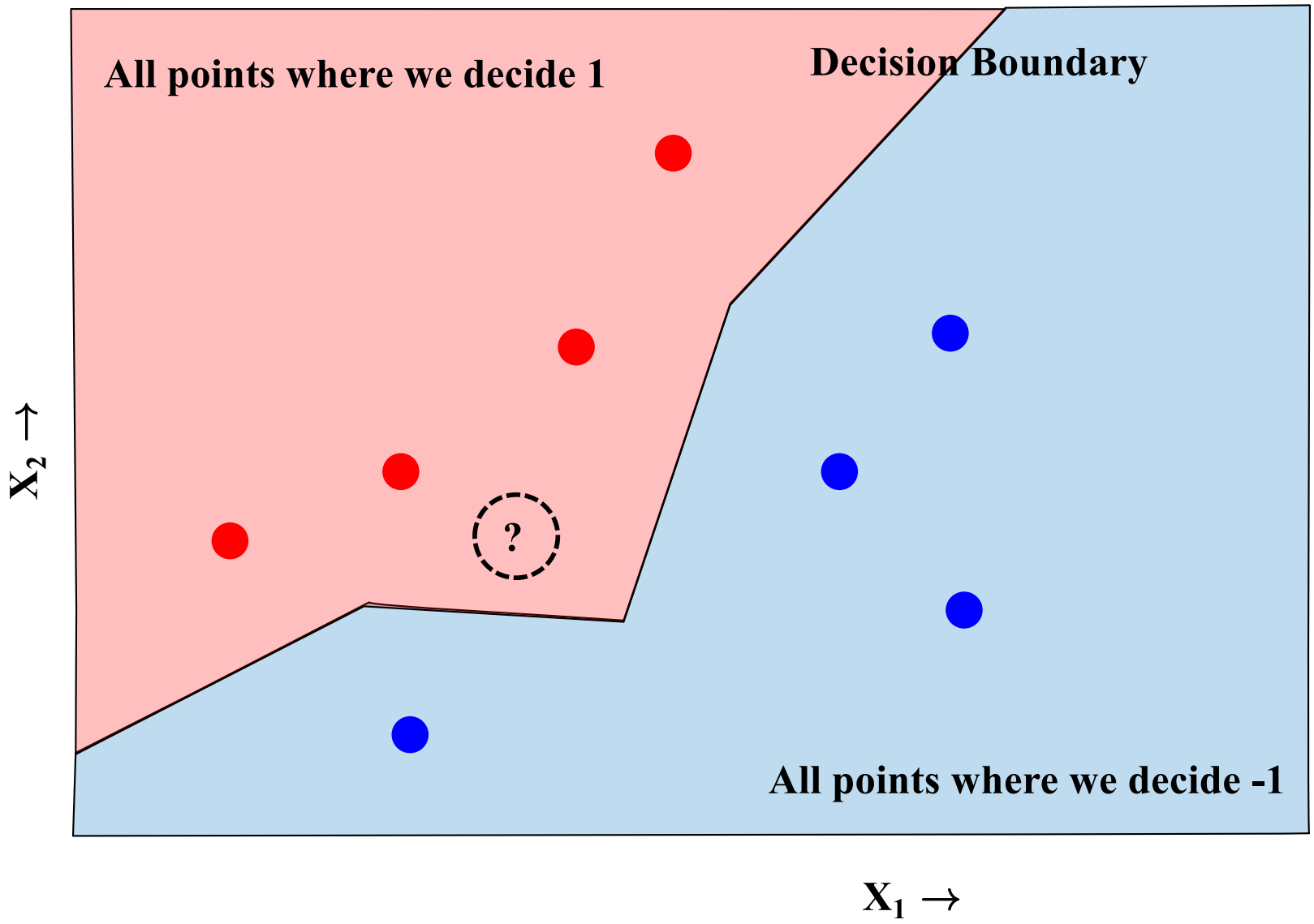

Features x
Discrete class c
(usually 0/1 or +1/-1 )
Predict discrete function $\hat{y}(x)$

# Classification



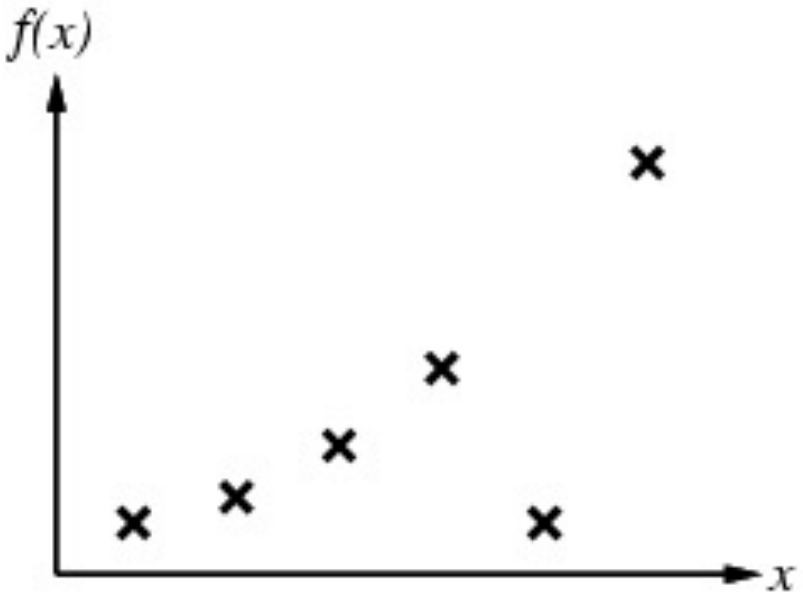
# Classification

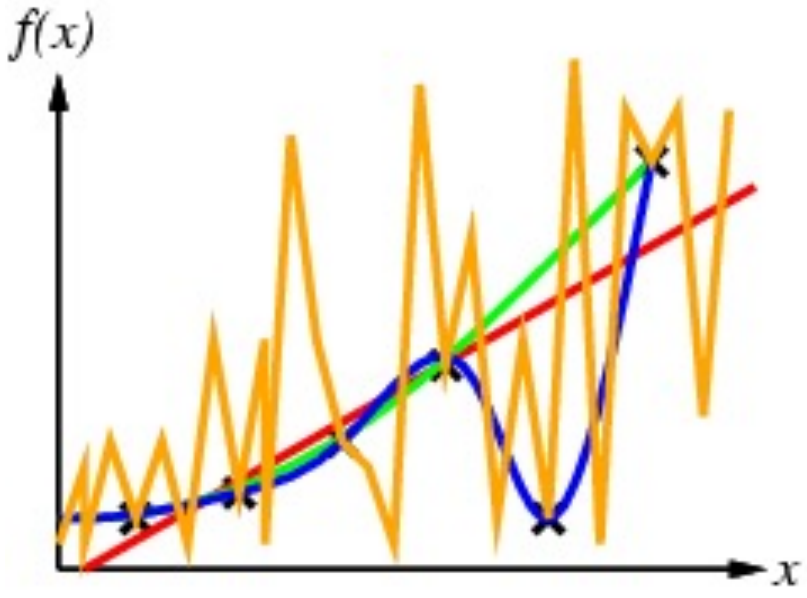
$$\mathrm{ERR} = \frac{1}{m} \sum_{i} \left[ y^{(i)} \neq \hat{y}(x^{(i)}) \right]$$



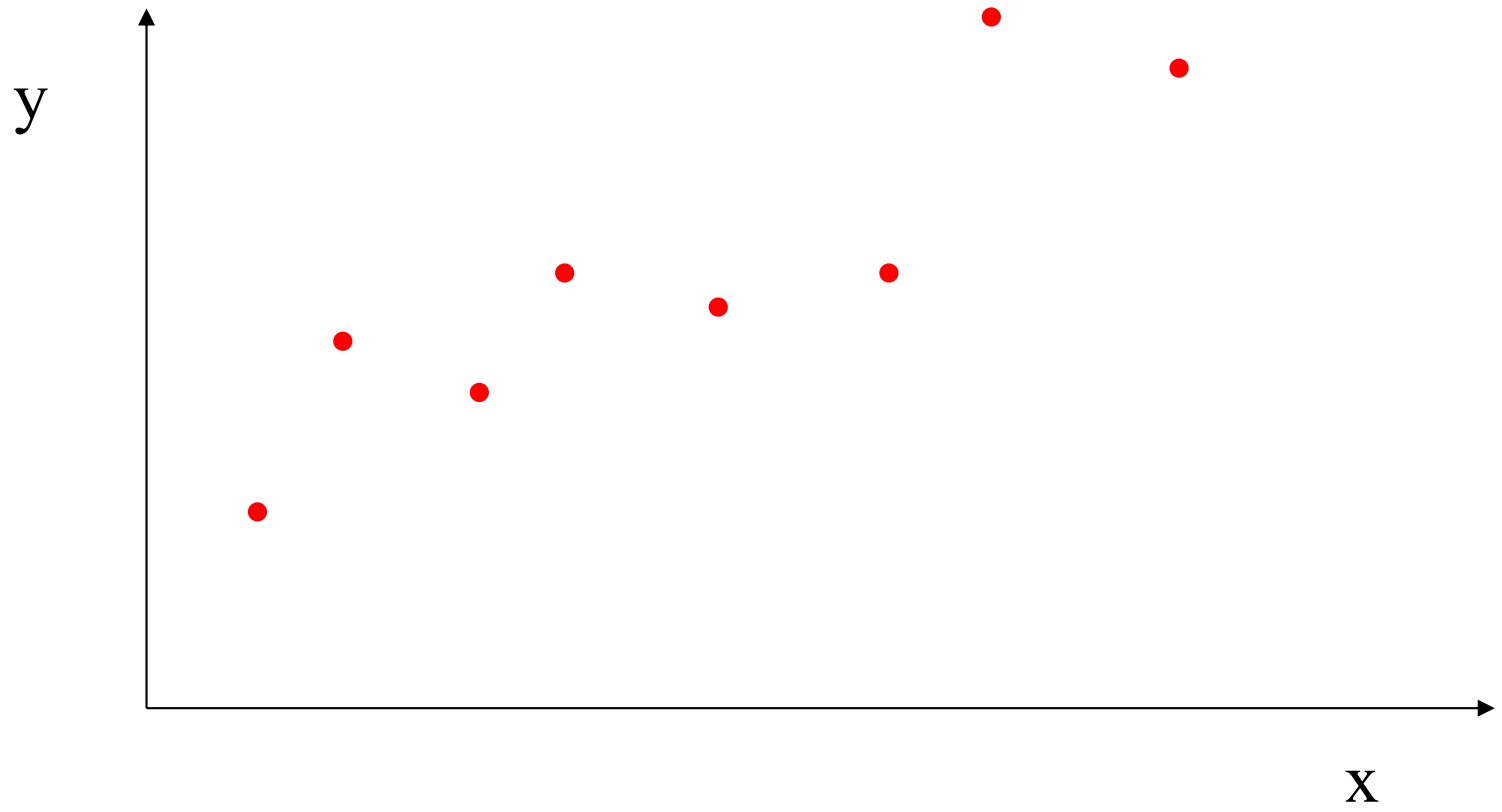
# Inductive bias

- "Extend" observed data to unobserved examples
  - "Interpolate" / "extrapolate"
- What kinds of functions to expect? Prefer these ("bias")
  - Usually, let data pull us away from assumptions only with evidence!



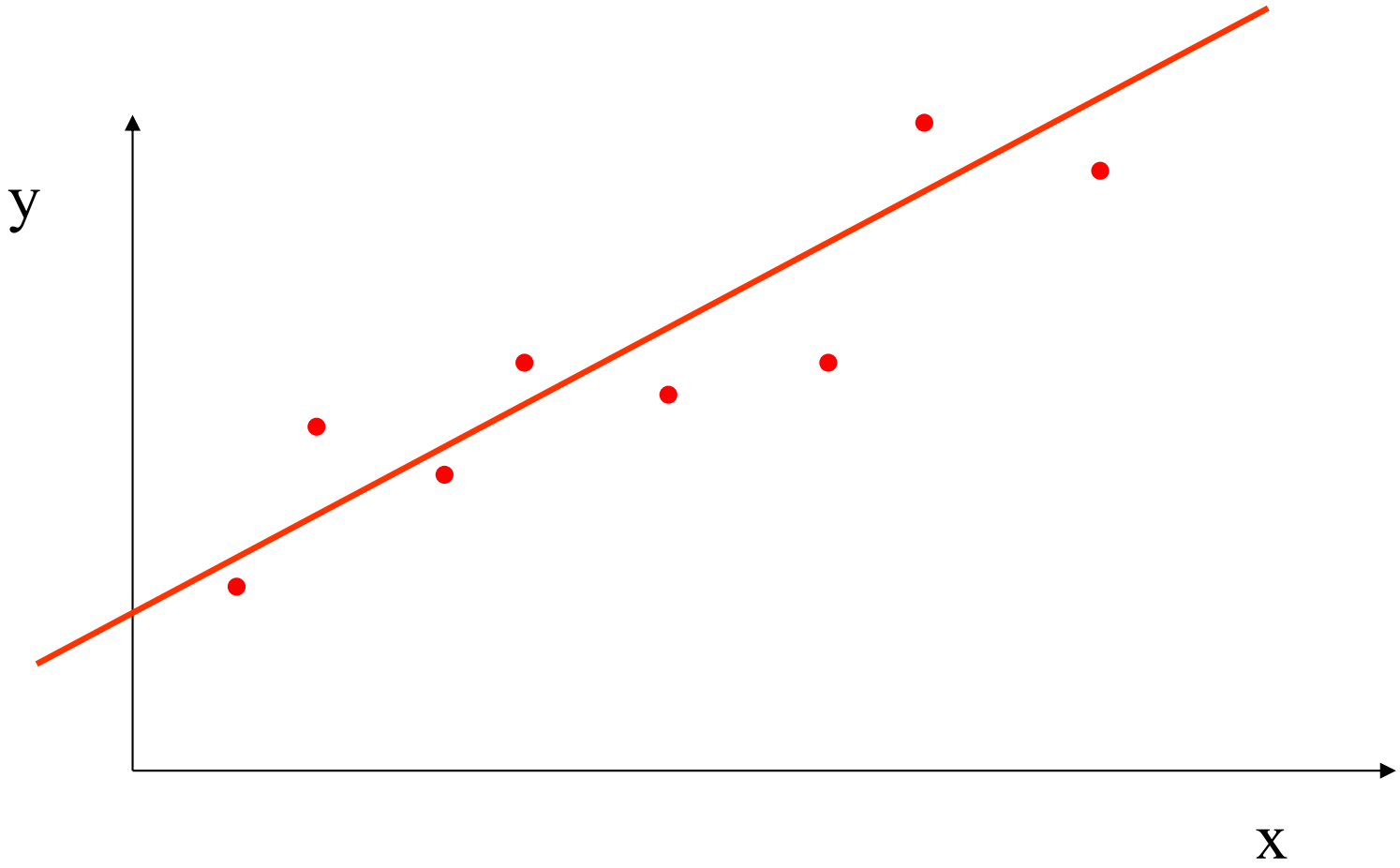



# Overfitting and complexity

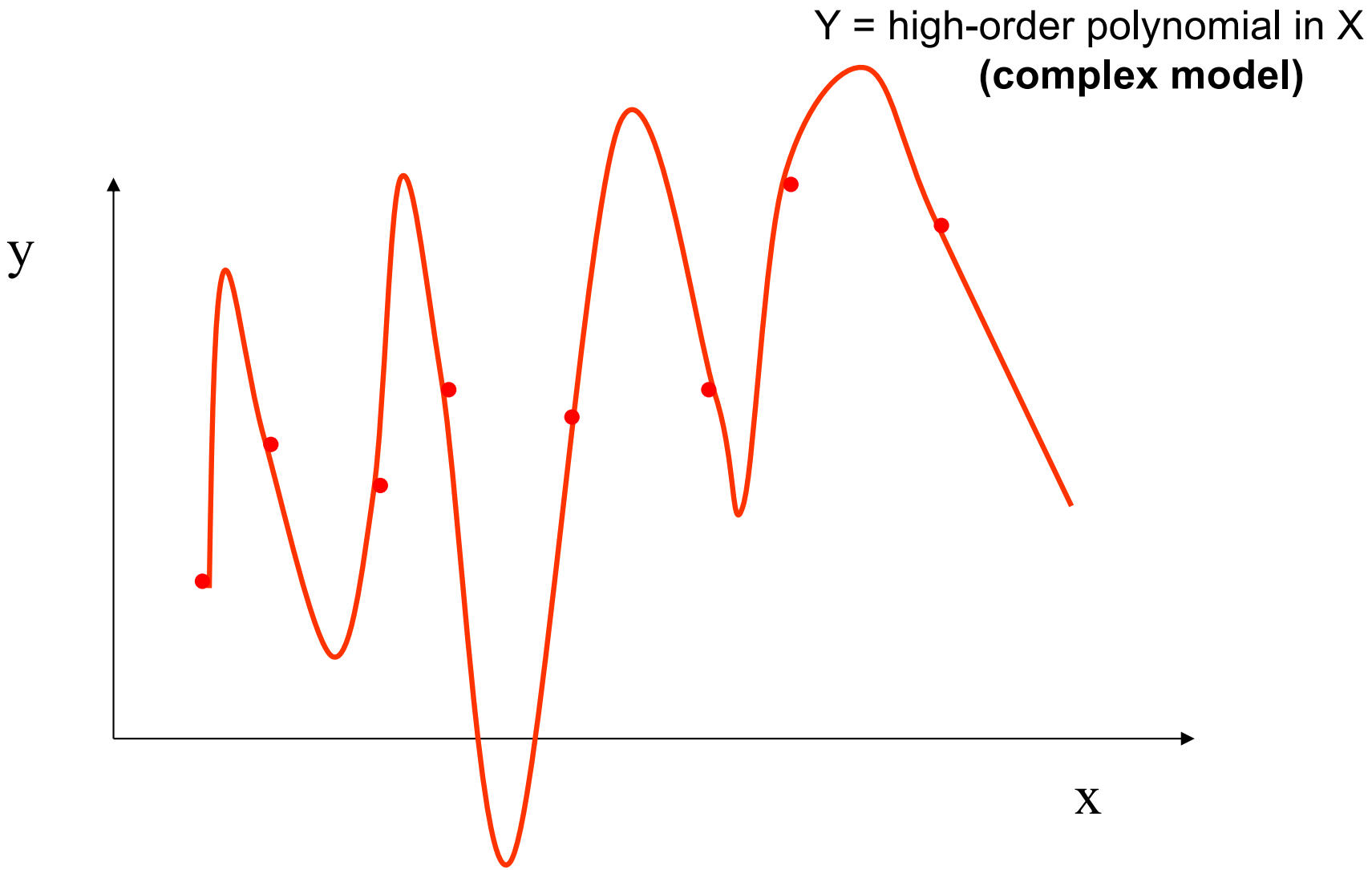


# Overfitting and complexity

**Simple** model: Y= aX + b + e

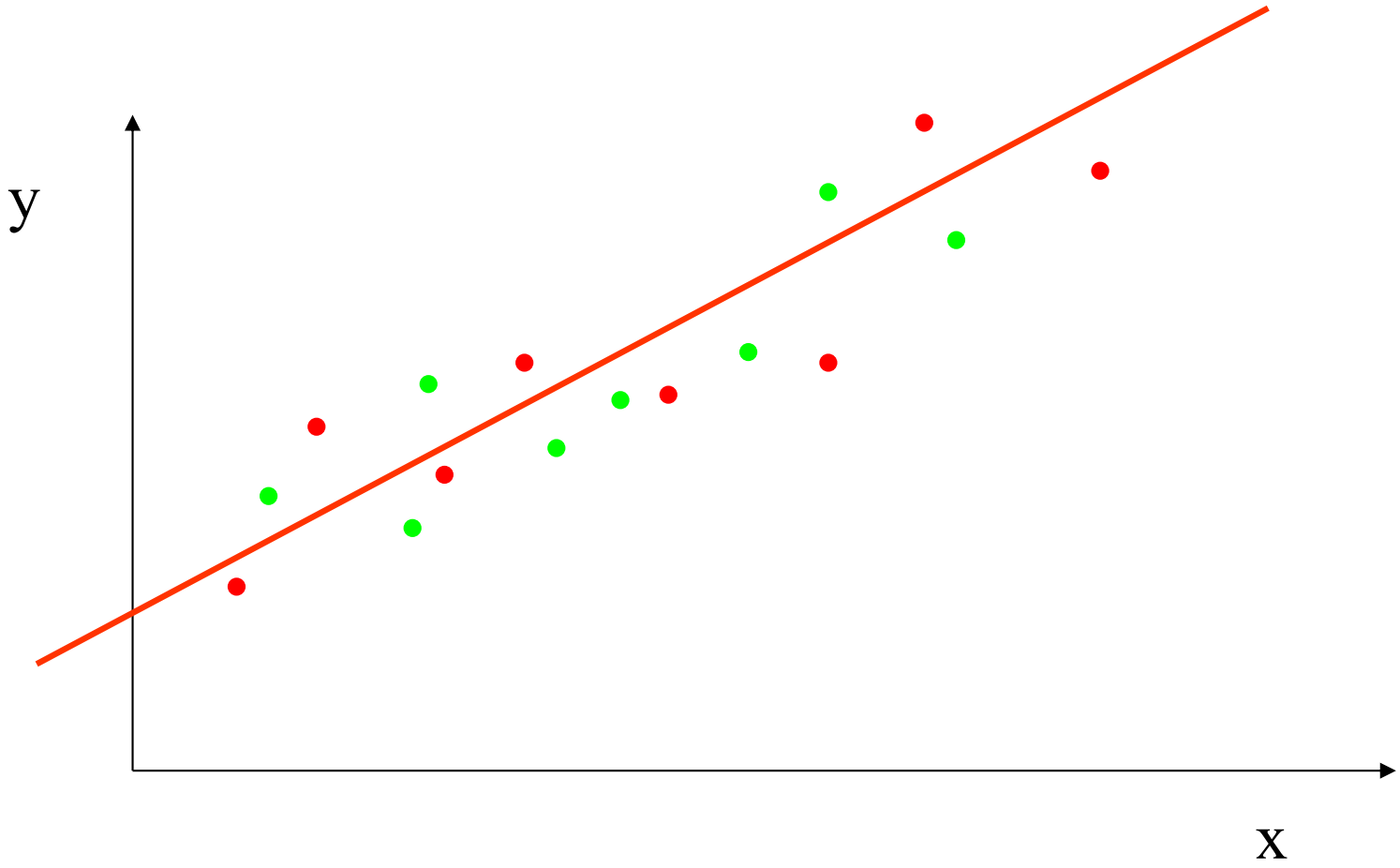


# Overfitting and complexity

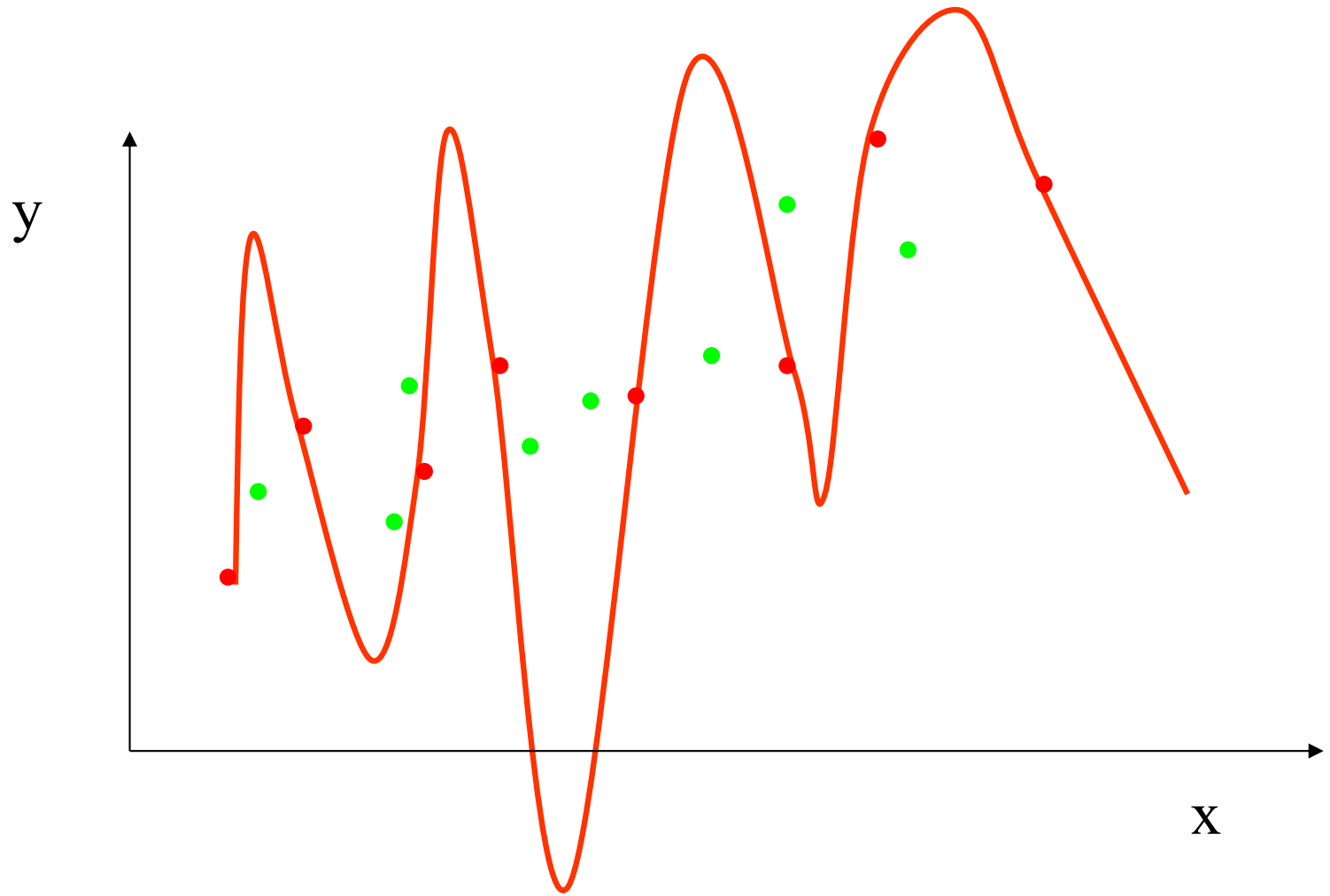


# Overfitting and complexity

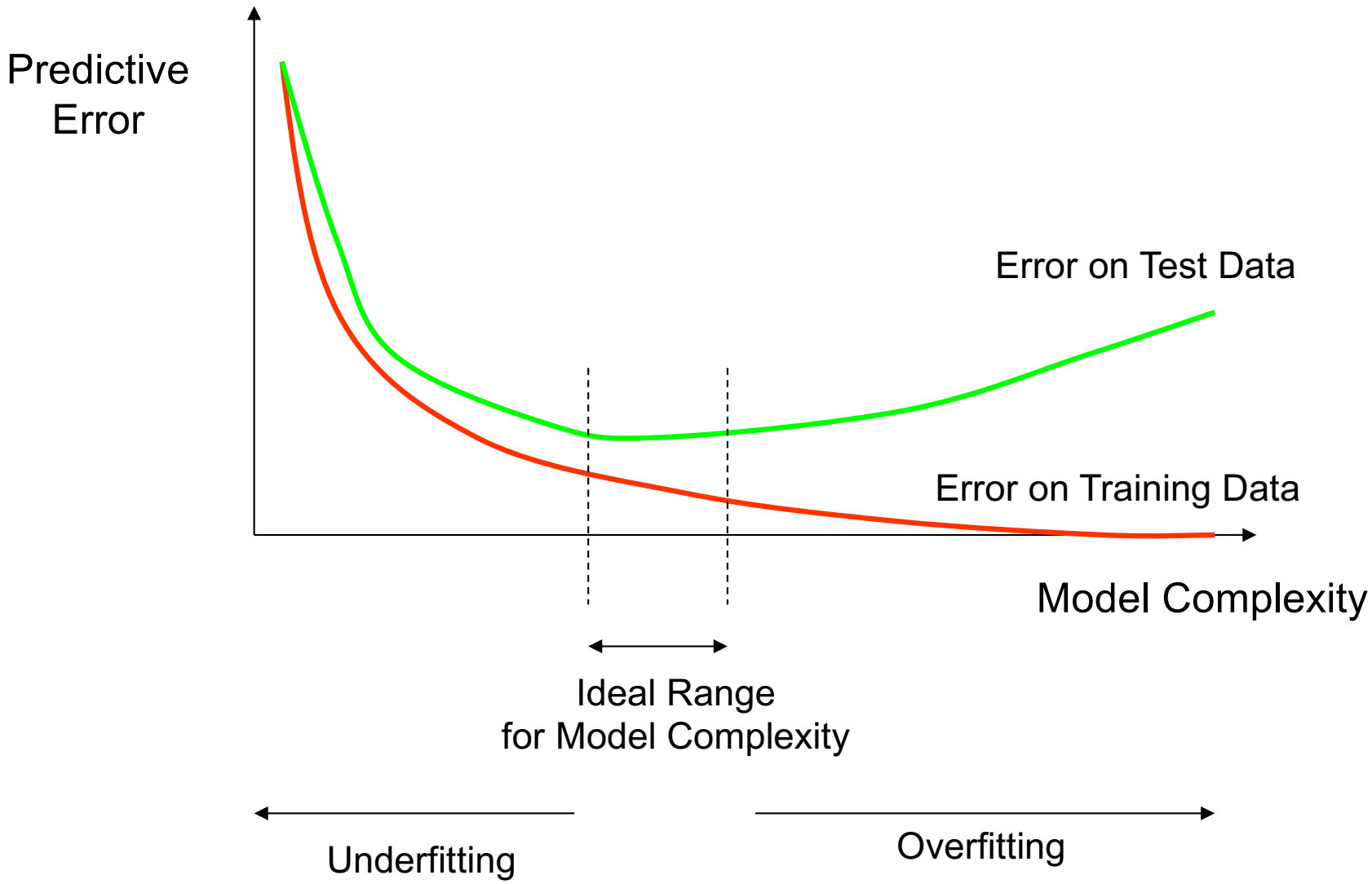
**Simple** model: Y= aX + b + e



# Overfitting and complexity



# How Overfitting affects Prediction



# Summary

- What is machine learning?
  - Types of machine learning
  - How machine learning works
- Supervised learning
  - Training data: features x, targets y
- Regression
  - (x,y) scatterplots; predictor outputs f(x)
- Classification
  - (x,x) scatterplots
  - Decision boundaries, colors & symbols
- Complexity, Bias and Overfitting