Announcements

- pset1 out today, due in two weeks.
    - Submit on Gradescope

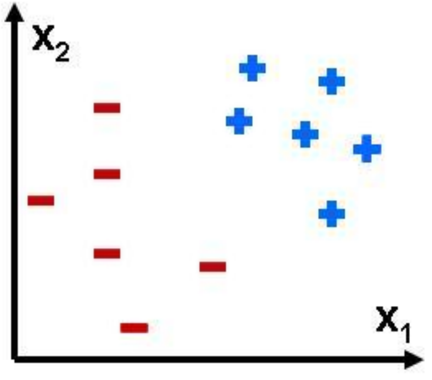
- No screens (laptops, tablets, phones) during the class.

Last time: Supervised Learning

Output

Categorical → **Classification**

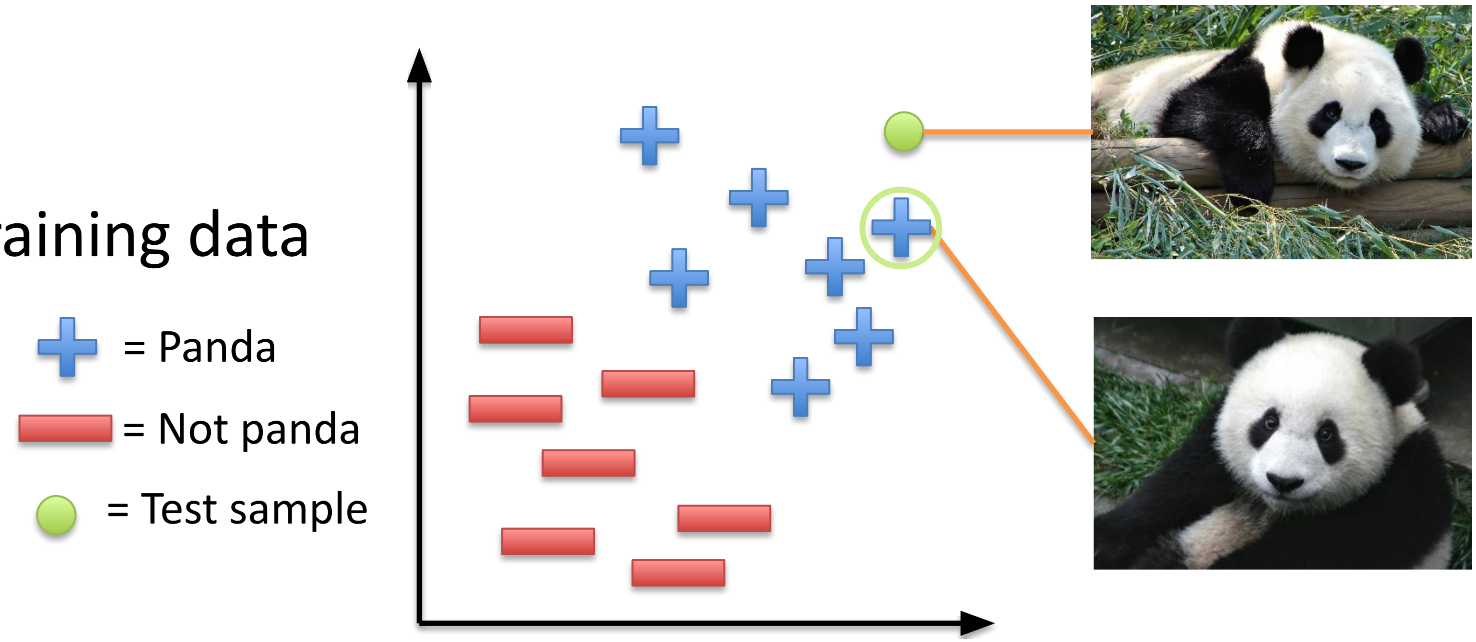
Continuous → **Regression**



$y \in \{0,1\}$ 0: Negative Class
1: Positive Class

Nearest Neighbor Classifier

Training data



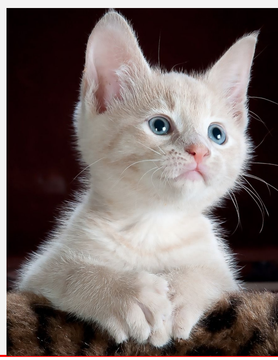
Variants of this algorithm:

- K-nearest neighbor classifier (k-NN)
- Approx nearest neighbor

K-Nearest Neighbor Classifier

Training Data

Feature vector



Feature vector

$$\sum_{i=1}^{d}(x_i - y_i)^2$$

L2 distance

Feature vector

Diving deeper into model analysis

	Predicted "1"	Predicted "0"
GT Label "1"	**True Positive (TP)**	False Negative (FN)
GT Label "0"	False Positive (FP)	True Negative (TN)

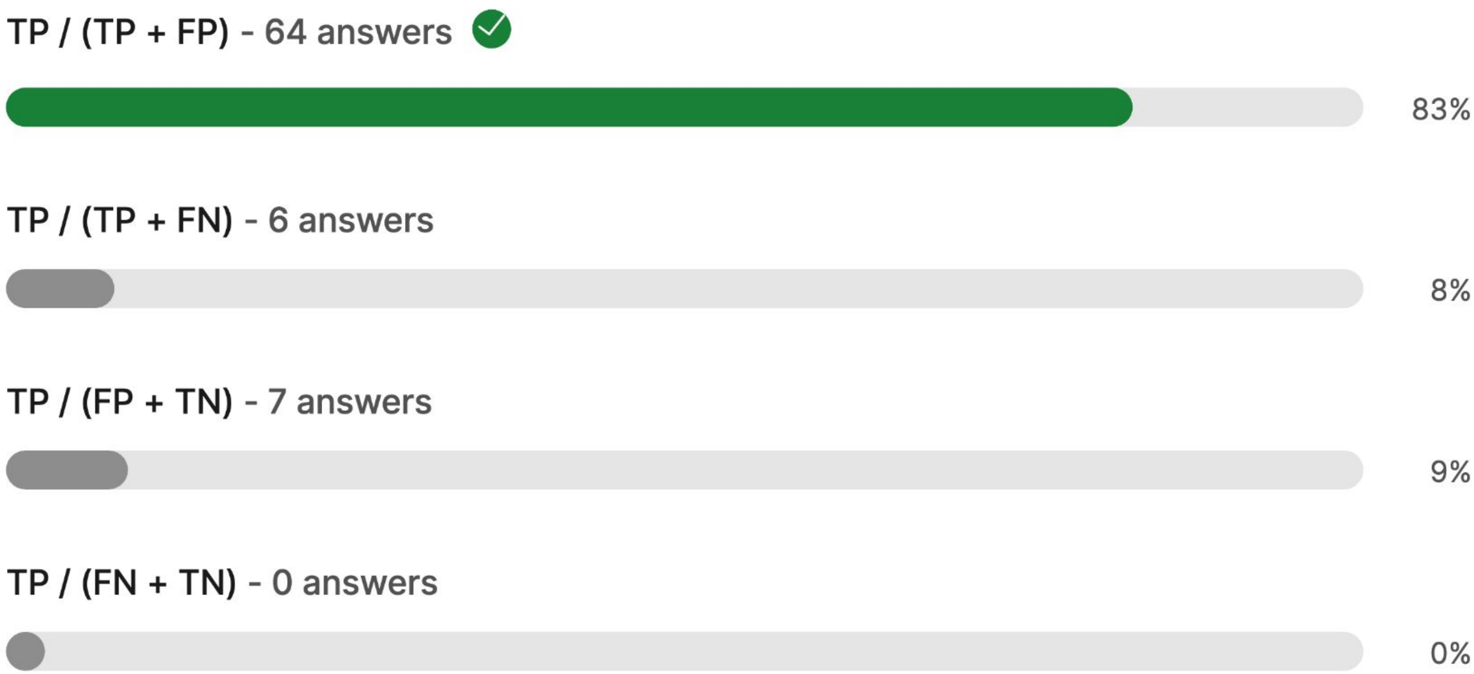
slido

What is precision?

Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

What is precision?

Quiz question 77 answers 77 participants



slido

Precision and Recall

- **Precision:**
  a. The percentage of predictions that are correct
  b. The ability to identify **only** relevant data points

$$Precision = \frac{TP}{TP + FP}$$

- **Recall:**
  a. The percentage of relevant data points that are correctly identified
  b. The ability to identify **all** relevant data points

$$Recall = \frac{TP}{TP + FN}$$

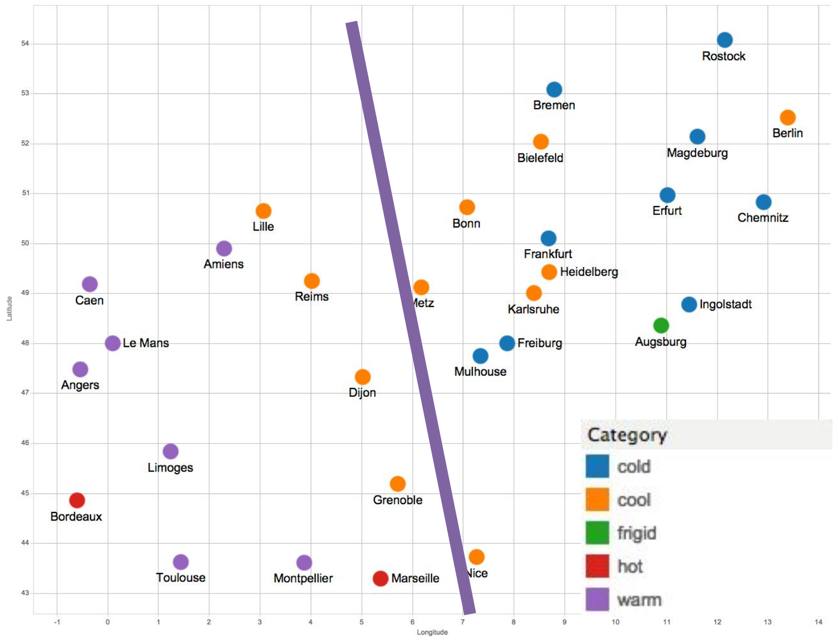
Hypothesis γ

γ : function parametrized by θ, e.g.,

$$\gamma(x) = \mathrm{sign}(\underbrace{\boxed{a}}_{\theta_{0,1}}x + \underbrace{\boxed{b}}_{\theta_2})$$

Goal: input x $\longrightarrow$ $\boxed{\gamma_{a,b}{}^*}$ $\longrightarrow$ output y

Linear Regression



Linear hypothesis:

$$\gamma_{a,b}(x) = \text{sign}(ax + b)$$

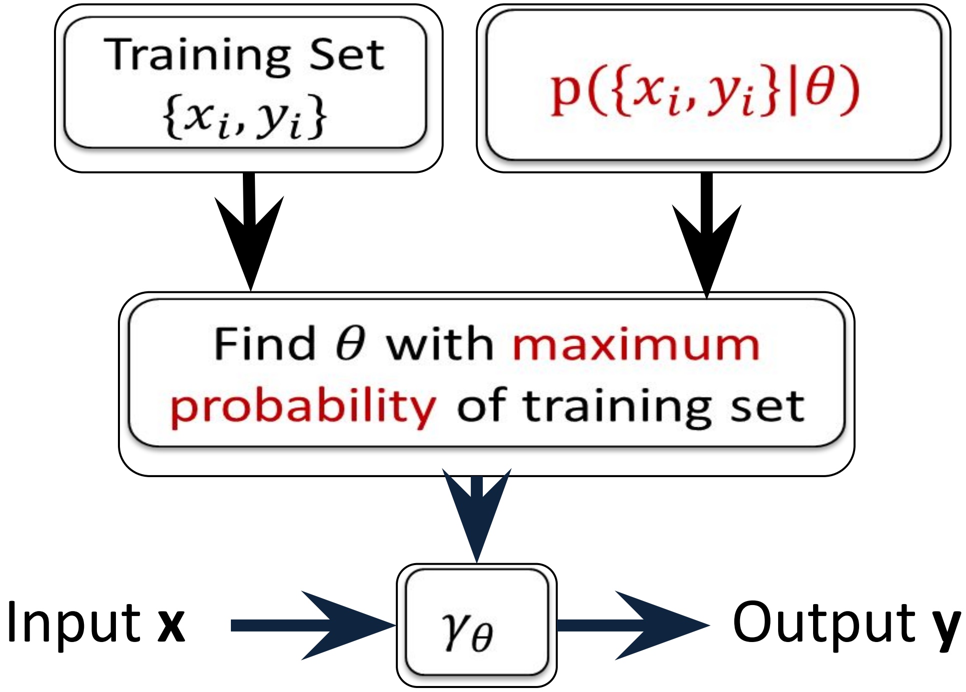
Error Function:

Portion of incorrect predictions

$$Error(\gamma_{a,b}, D\{x, y\}) = \frac{1}{N} \sum_{i=1}^{N} \gamma_{a,b}(x_i) \neq y_i$$

Goal: minimize $Error(\gamma_{a,b}, D\{x, y\})$

Alternative View:
"Maximum Likelihood"



Maximum likelihood way of estimating model parameters θ

In general, assume data is generated by some distribution

$$U \sim p(U|\theta)$$

Observations (i.i.d.)

$$\mathrm{D} = \{u^{(1)}, u^{(2)}, \ldots, u^{(m)}\}$$

Maximum likelihood estimate

Likelihood

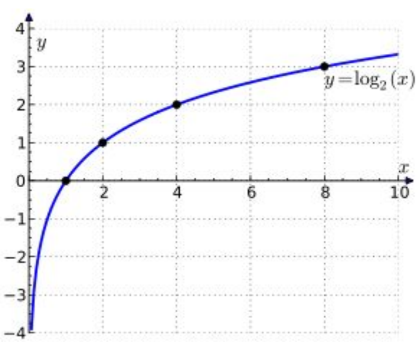
$$\mathcal{L}(\mathrm{D}) = \boxed{\prod_{i=1}^{N} p(u^{(i)}|\theta)}$$

$$\theta_{ML} = \underset{\theta}{\operatorname{argmax}}\, \mathcal{L}(\mathrm{D})$$

Log likelihood

$$= \underset{\theta}{\operatorname{argmax}} \boxed{\sum_{i=1}^{N} \log p(u^{(i)}|\theta)}$$

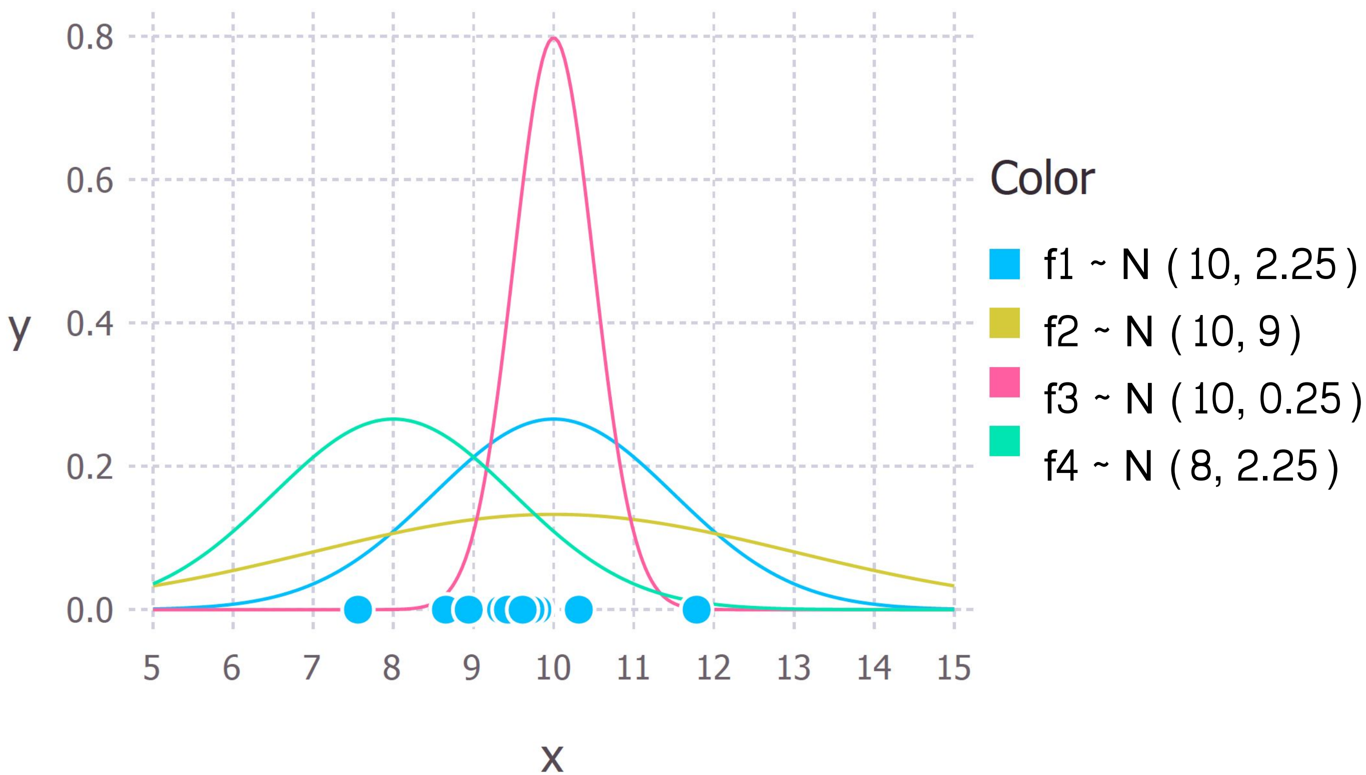
Note: *p* replaces γ, and max replaces min



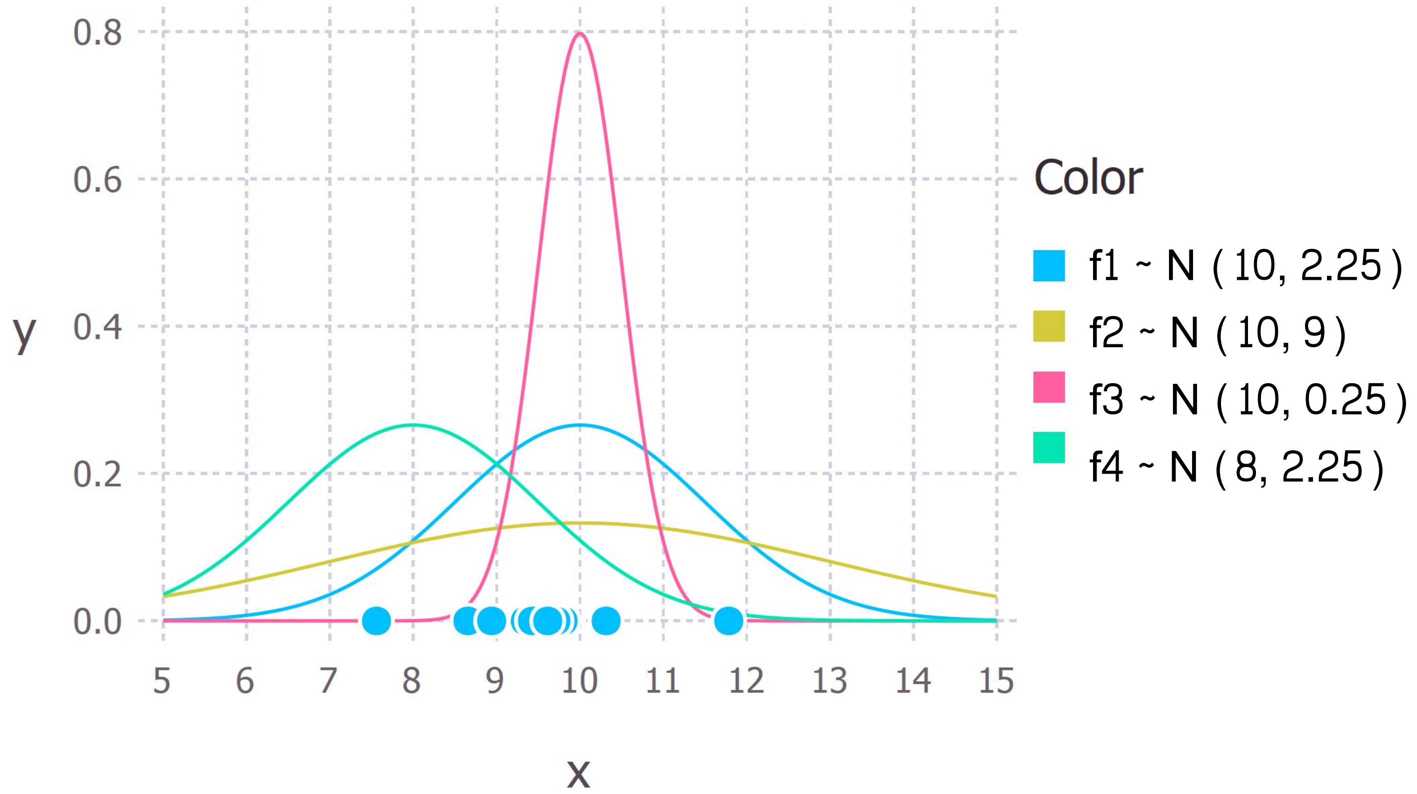
log(f(x)) is monotonic/increasing, same argmax as *f(x)*

ML: Another example

- Observe a dataset of points $D = \{x^i\}_{i=1:10}$
- Assume x is generated by Normal distribution, $x \sim N(x|\mu, \sigma)$
- Find parameters $\theta_{ML} = [\mu, \sigma]$ that maximize $\prod_{i=1}^{10} N(x^i|\mu, \sigma)$



0.8
0.6
0.4
0.2
0.0
y
5
6
7
8
9
10
11
12
13
14
15
x
Color
f1 ~ N (10, 2.25)
f2 ~ N (10, 9)
f3 ~ N (10, 0.25)
f4 ~ N (8, 2.25)



What is the right distribution that fits the given datapoints?

Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

What is the right distribution that fits?

Multiple Choice Poll   79 votes   79 participants

Blue N(10,2.25) - 56 votes — 71%

Yellow-green: N(10,9) - 12 votes — 15%

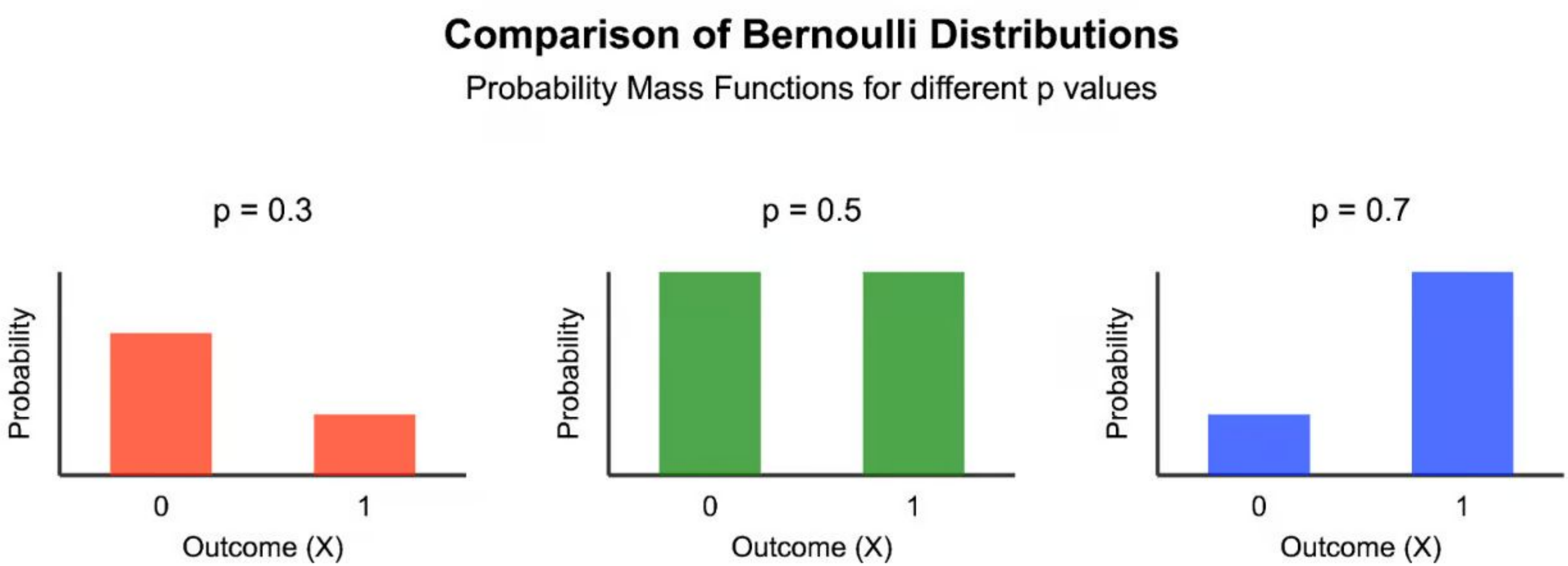
Magenta: N(10,0.25) - 11 votes — 14%

Green: N(8,2.25) - 0 votes — 0%

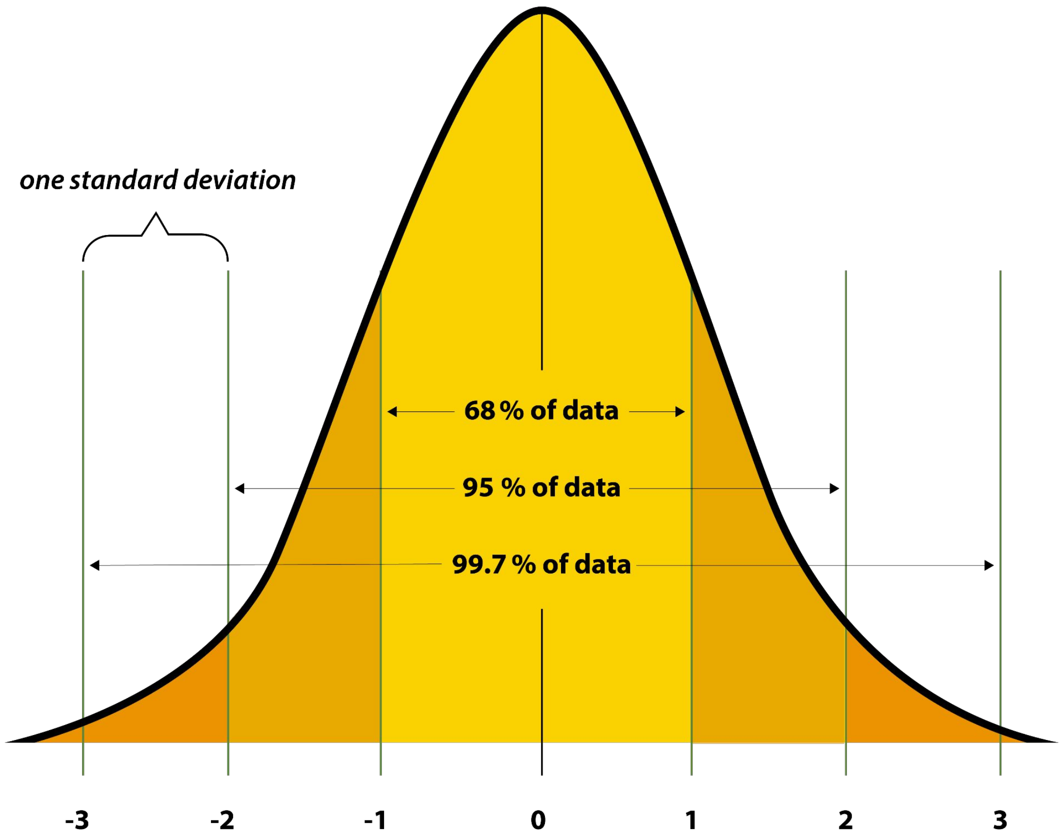
slido

Types of data distribution: Bernoulli

$$P(X = 1) = p, \quad P(X = 0) = 1 - p$$



Types of data distribution: Normal



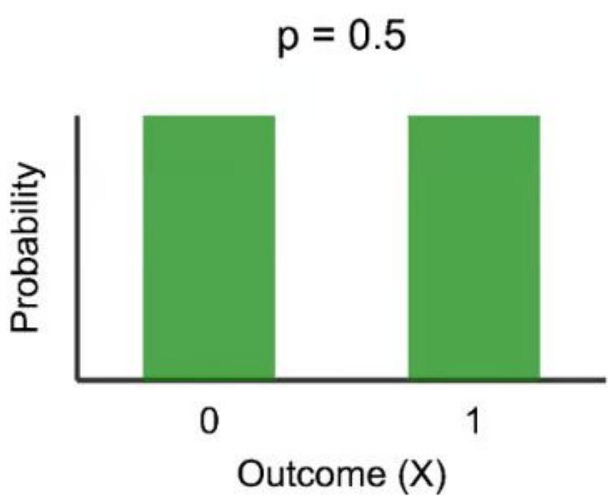
$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

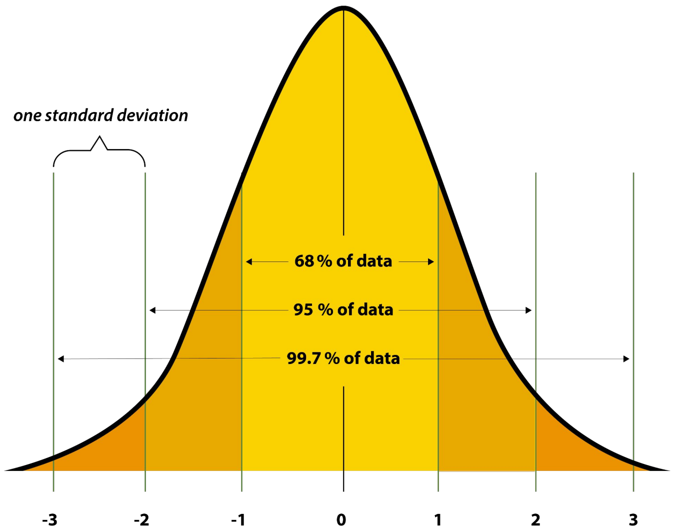
Parameters:

- Mean (**μ**)
- Deviation (***σ***)

How to choose a particular distribution?

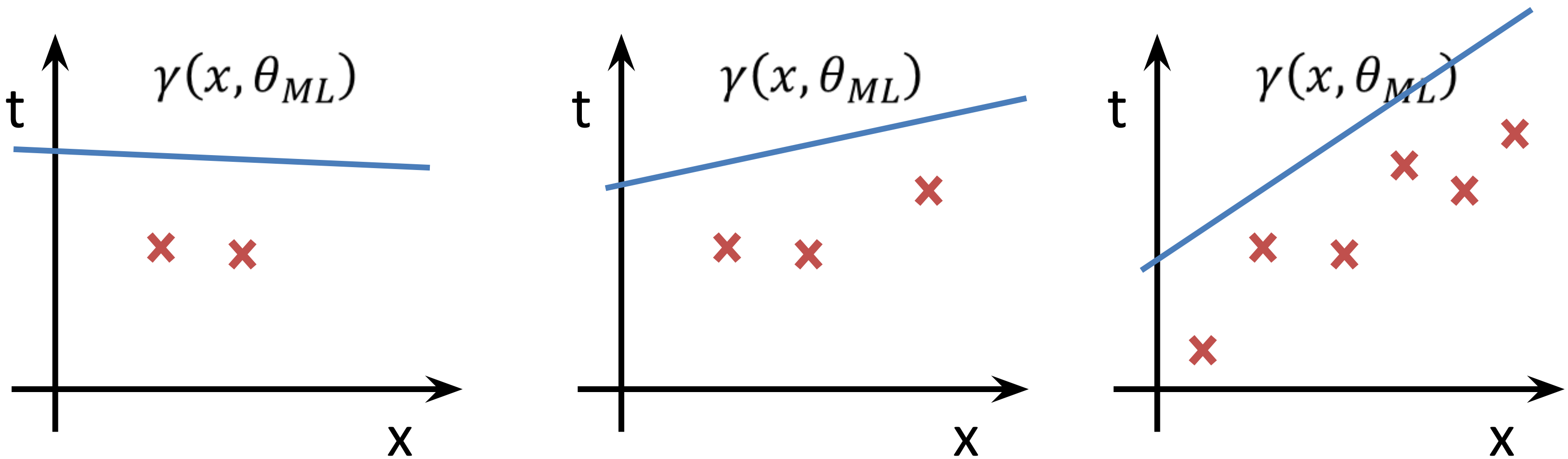
Distribution	
Bernoulli	● Discrete data samples, with binary outcomes. ○ Eg: Coin tosses
Normal	● Data is primarily centered around a value, eg: mean ● Most other data points fall very close to the mean. ● Eg: Weights of a mouse





Problem with Maximum Likelihood: Bias

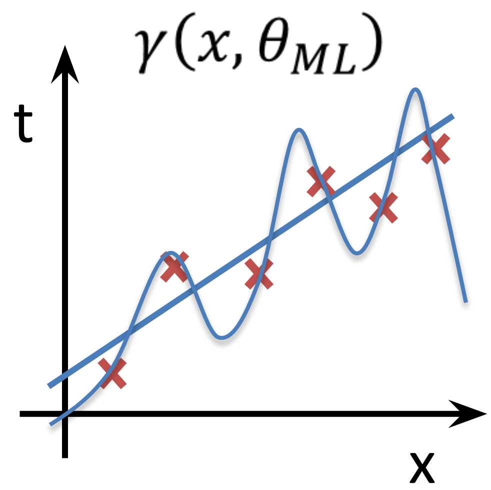
$$\text{MLE} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log p(u^{(i)}|\theta)$$



- Depends on the number of datapoints (N)
- When N is small, the estimate may not be an accurate reflection of the true model

One issue: Overfitting

MLE = $\underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log p(u^{(i)}|\theta)$



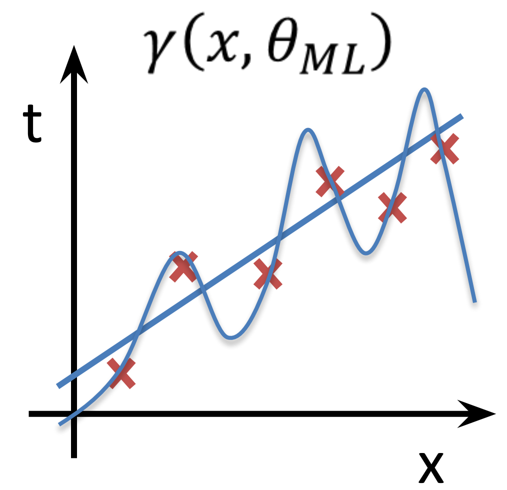
- Let K denote the complexity of the model estimator: **$\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K\}$**
  - K = 1 denotes a line.
  - K = 15 denotes a 15-degree polynomial function.

- How to control the value of K?

- MLE does not offer a way to directly control model complexity.
  - It will always choose the solution with K = 15.

One issue: Overfitting

$$\text{MLE} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log p(u^{(i)}|\theta)$$



- Let K denote the complexity of the model estimator: $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K\}$
  - K = 1 denotes a line.
  - K = 15 denotes a 15-degree polynomial function.

- How to control the value of K?

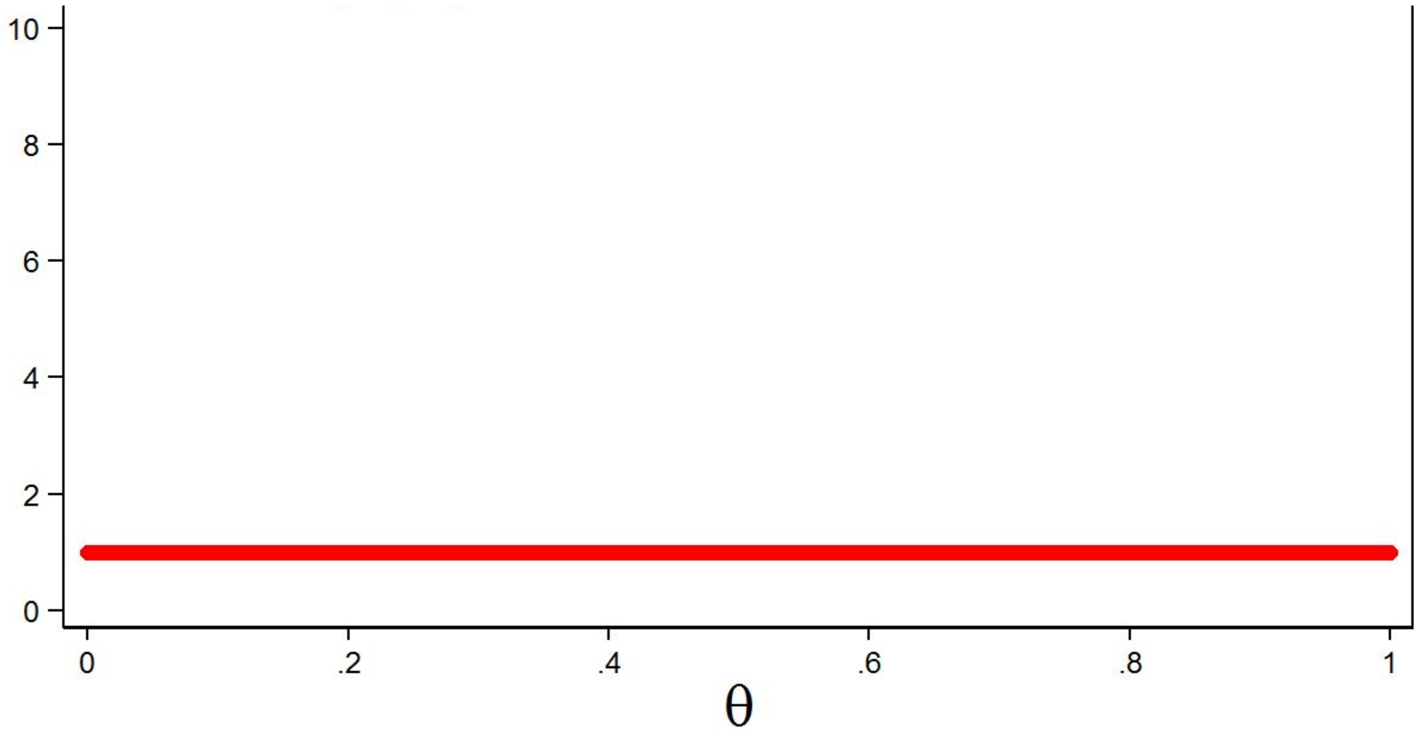
- **Solution:** use a Bayesian method--define a prior distribution over the parameters (results in regularization)

Bayesian Modeling

Prior distribution

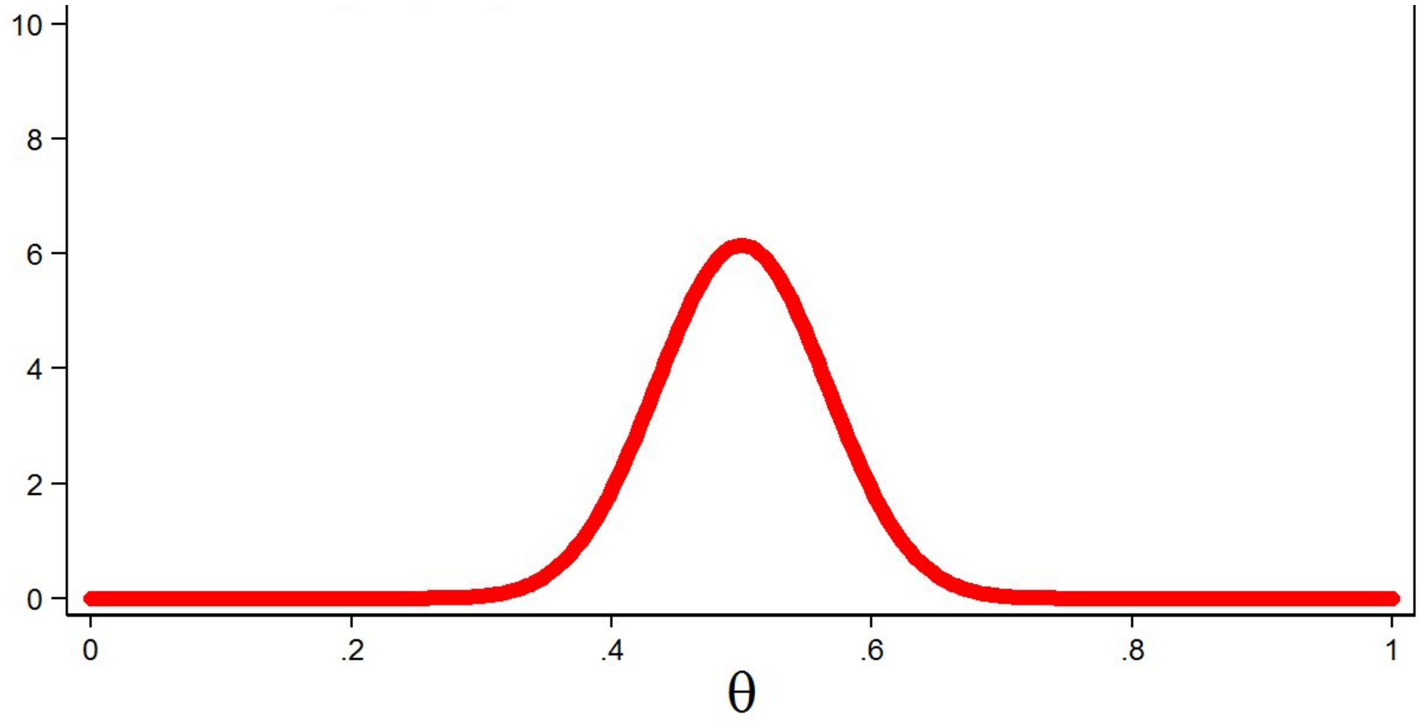
- Prior distributions p(θ) are probability distributions of model parameters based on some a **priori knowledge** about the problem at hand.
- Prior distributions are assumed **before** any data is observed.
- **Examples:**
  - A coin toss is random (50% heads, 50% tails).
  - Assumed prior : **Binomial distribution**
    - bernoulli is a special case of binomial.

What information is this prior (θ) is giving?



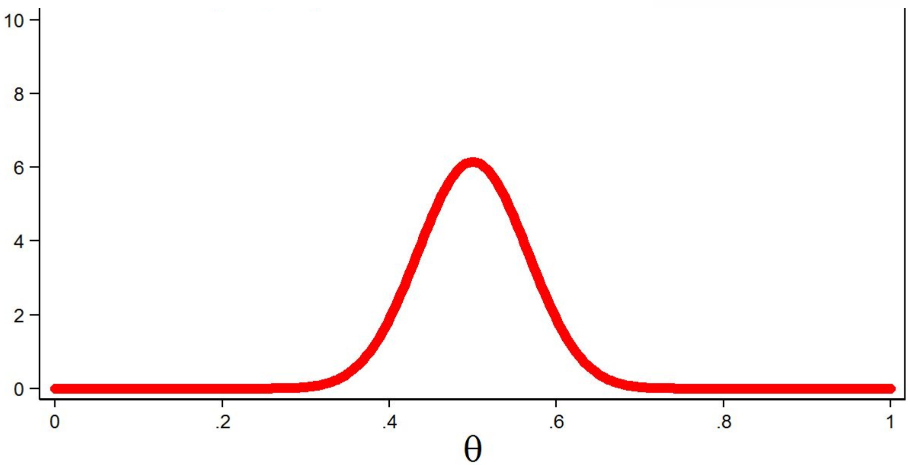
The prior takes a uniform probability distribution.

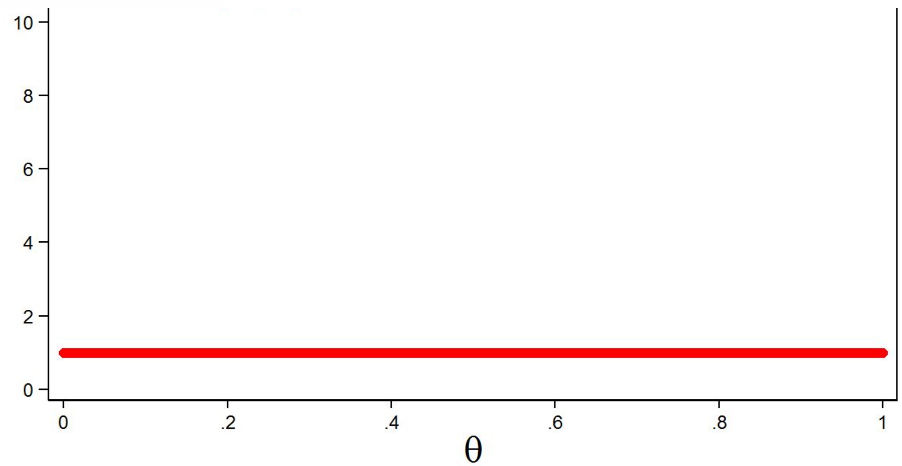
What information is this prior (θ) is giving?



- The coin is likely to have a outcome of heads with 50% probability

Informative v/s uninformative priors





- Important to choose a prior that is is more likely to be aligned with the expected outcomes.

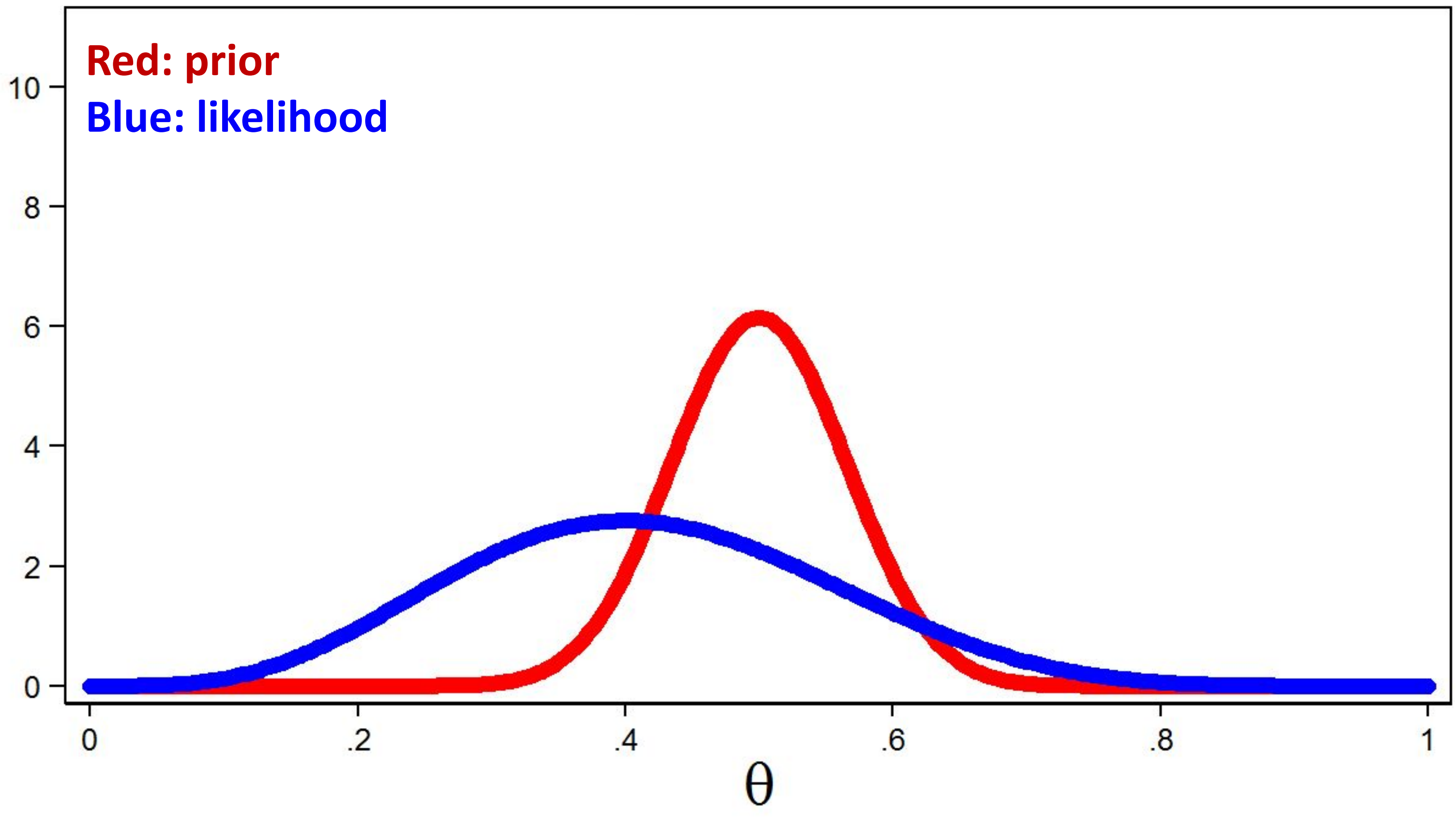
Reminder: Coin Toss Experiment

- n=10 coin tosses
- y (target) =4 number of heads

- P(head) = 4/10 = 0.4
- P(tail) = 1- 0.4 = 0.6

Likelihood
(estimated purely from data)

Prior and Likelihood



Posterior Distribution

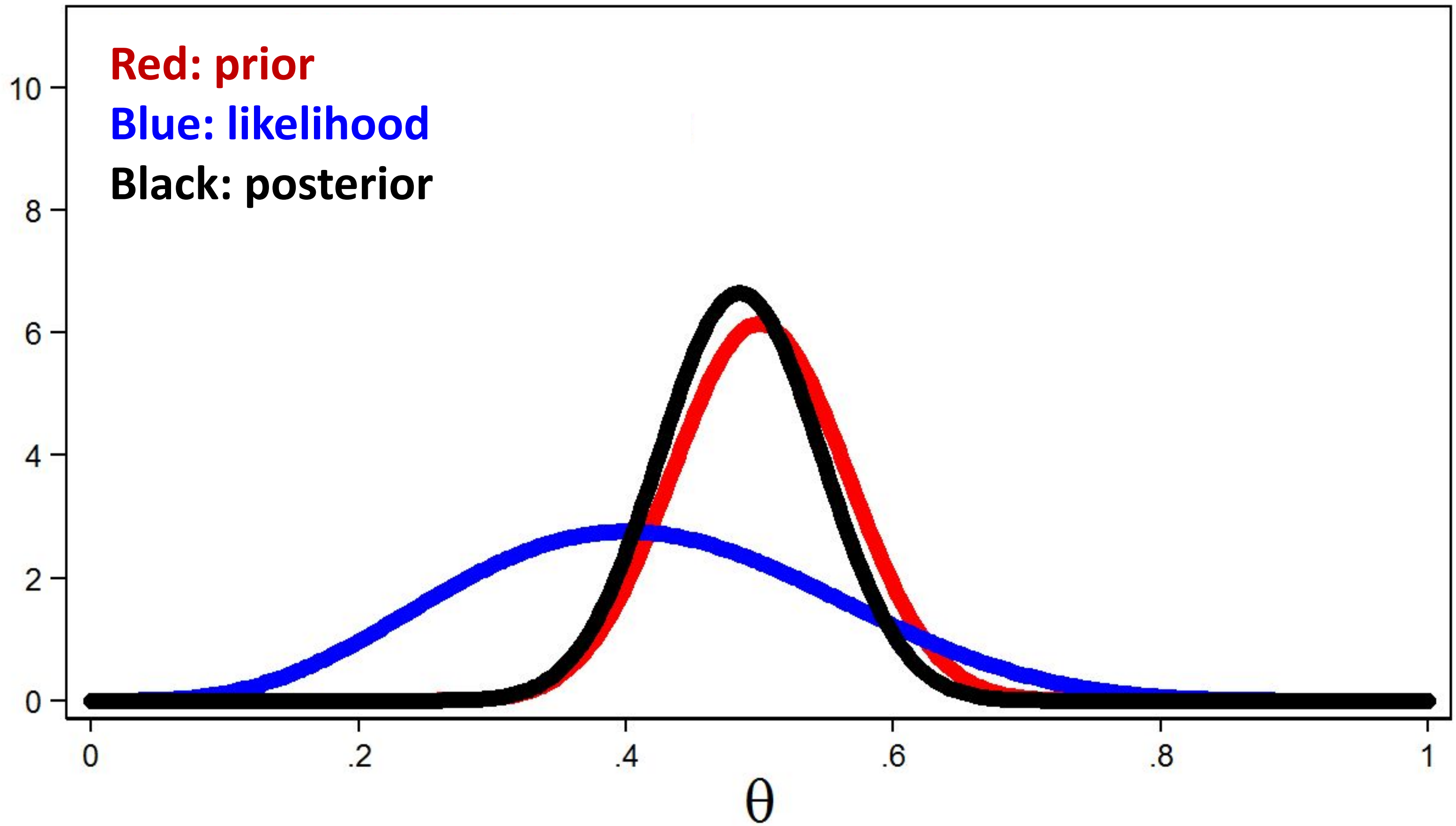
$$Posterior = \color{red}{Prior} \times \color{blue}{Likelihood}$$

Your mental model of the target distribution

Your estimate from the observed data

$$P(\theta|y) = \color{red}{P(\theta)}\color{blue}{P(y|\theta)}$$

Posterior Distribution



Evaluating your model

Setting up an experiment

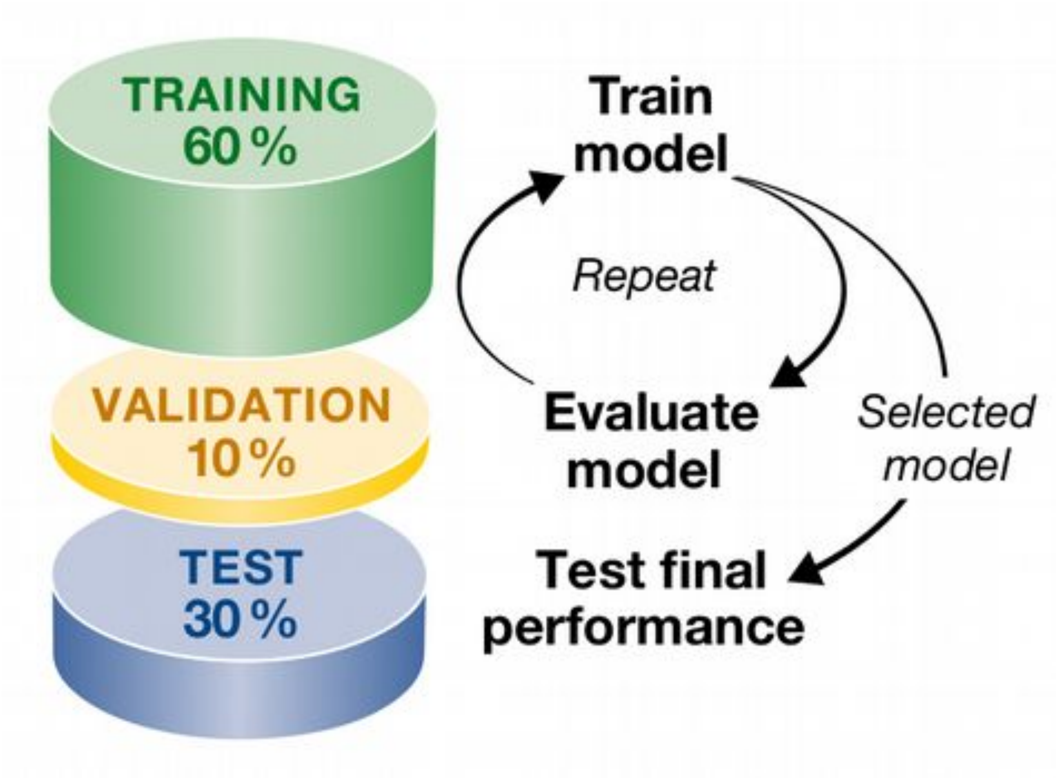
	Size	Price
train	2104	400
	1600	330
	2400	369
	1416	232
	3000	540
validation	1985	300
	1534	315
	1427	199
test	1380	212
	1494	243

1. **For each value of a hyperparameter**
   - train on the train set
   - evaluate learned parameters on the validation set.

2. Pick the model (hyper parameters) that achieved the **lowest validation error.**

3. Report this model's test set error.

Train-test-val framework



Pro: Computationally efficient

Con: This split of data may introduce a bias

slido

How to avoid bias due to the way training data was split?

ⓘ Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

How to avoid bias due to the way training data was split?

Quiz question | 81 answers | 81 participants

Do not split the data, use all of it! - 2 answers

2%

Split the data randomly many times find hyper parameter that works well most times - 59 answers

73%

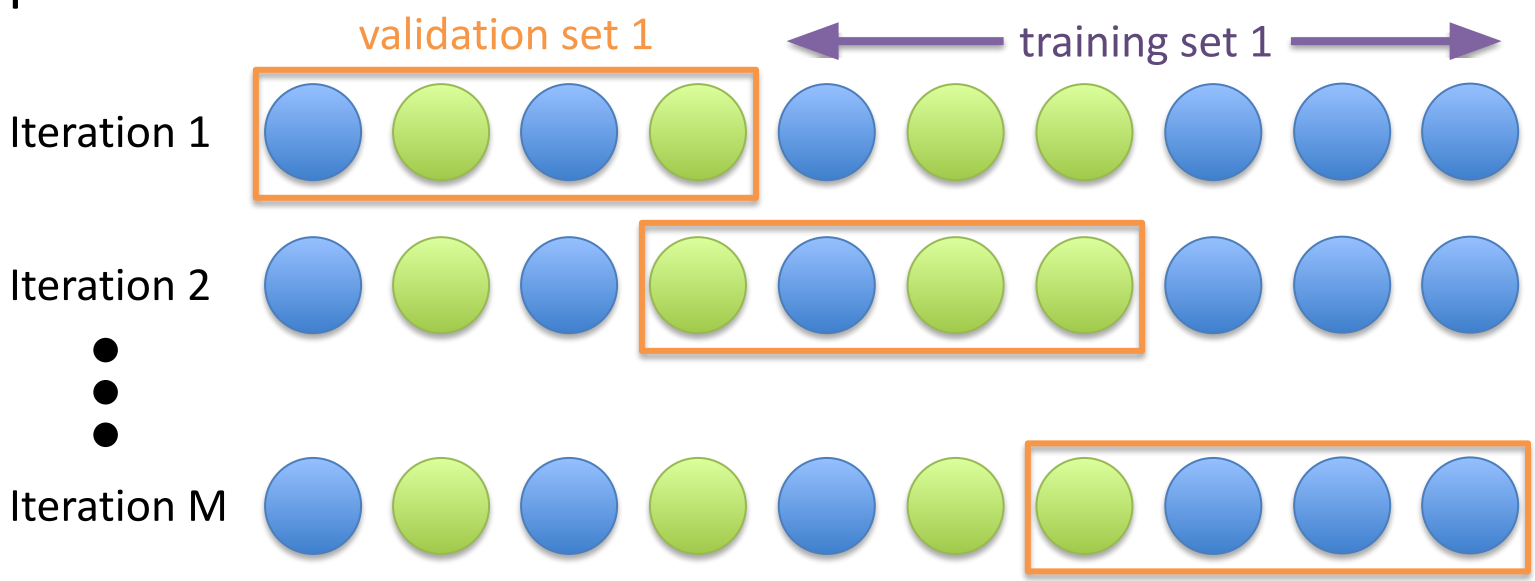
Split the data randomly many times and report the average test metric - 69 answers

85%

slido

What is the right way to split the data?

- **Solution:** use M-fold cross validation.
- Split training set into train/validation sets M times
- **Option-1:** Report average predictions
- **Option-2:** Pick hyper-parameters that work well across most of M splits



Pro: More robust to split bias

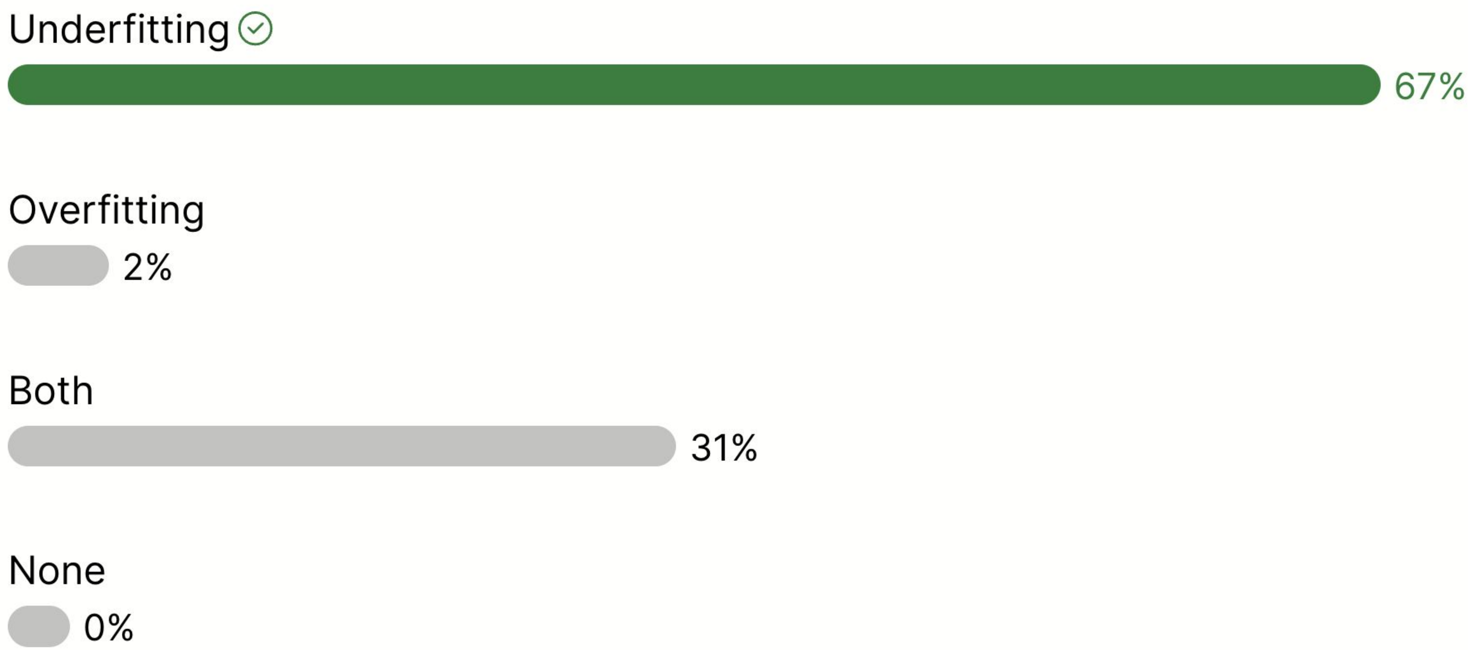
Con: Requires training M different models

slido

What do you call the scenario when a classifier is performing poorly on training data?

Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

What do you call the scenario when a classifier is performing poorly on training data?



slido

What do we call a classifier performing well on training data but poorly on test data?

What do we call a classifier performing well on training data but poorly on test data?

Overfitting
100%

Underfitting
0%

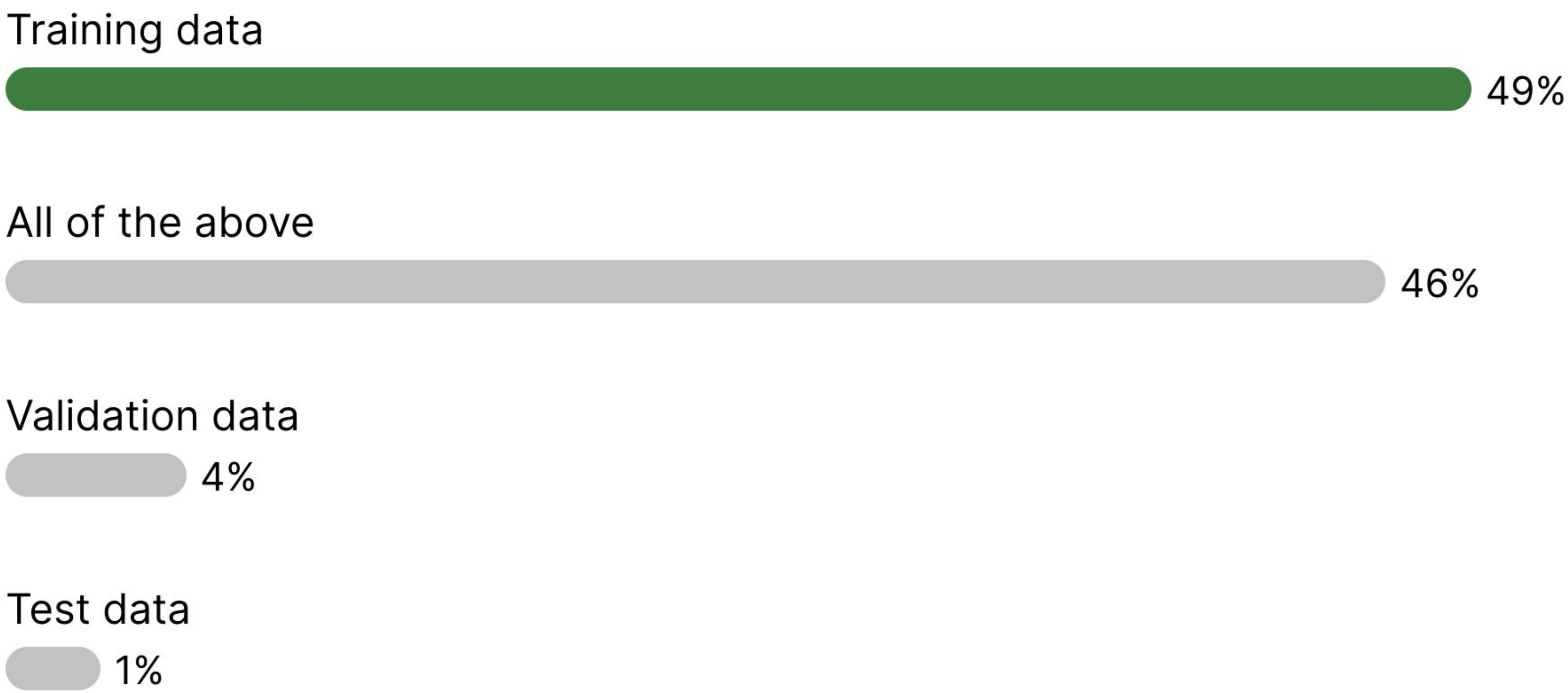
Both
0%

None
0%

What do classifiers work best on?

Click **Present with Slido** or install our [Chrome extension] to activate this poll while presenting.

What do classifiers work best on?



Reasons for underfitting

- Not enough training samples.
- Model too simple.
- Training samples too diverse.

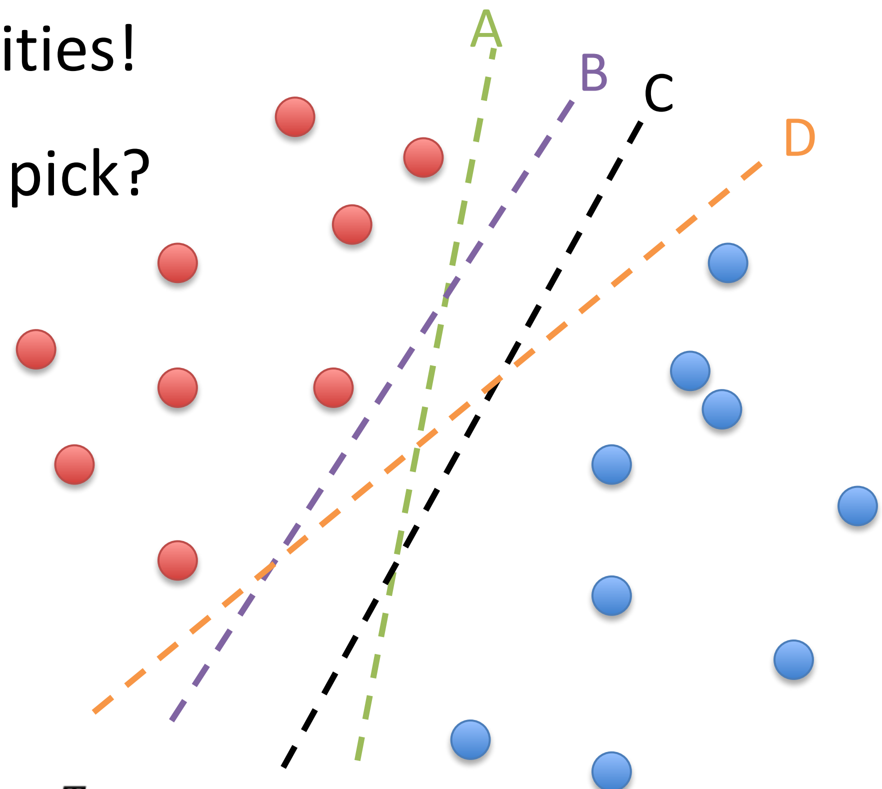
Reasons for overfitting

- Model overparameterized.
- Not correctly cross-validated.
- Training data from a very different distribution than test data.

Learning the right decision boundary

Separating two classes

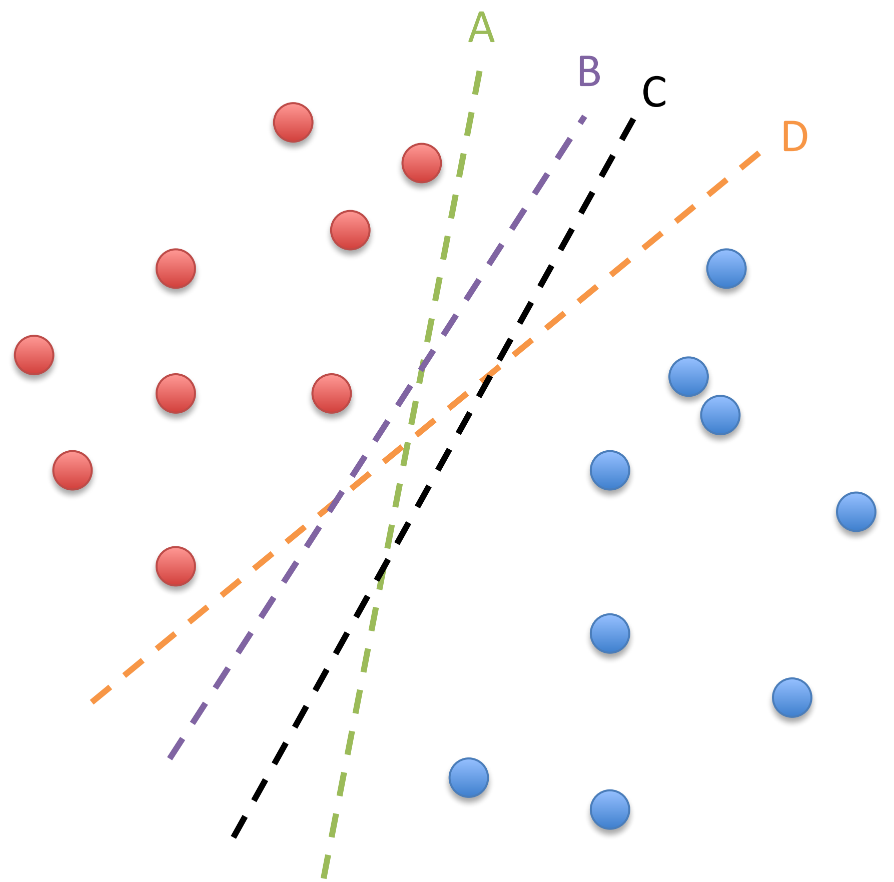
- Many possibilities!
- Which one to pick?



Decision boundary: $w^T x + b = 0$

$$y = \begin{cases} +1 \text{ [red] if } \operatorname{sign}(a^T x + b \geq 0) \\ -1 \text{ [blue] if } \operatorname{sign}(a^T x + b < 0) \end{cases}$$

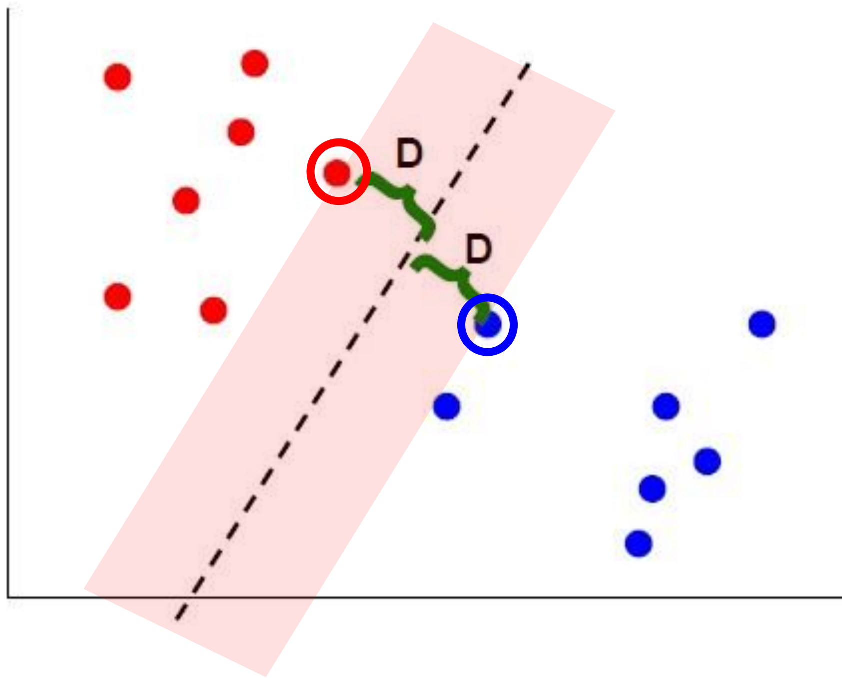
Separating two classes



Goal: Avoid misclassifying new test points generated from the same distribution as the training points

Maximum margin classification

Intuition: Instead of fitting all the points, focus on boundary points



Aim: learn a boundary that leads to the largest margin (buffer) from points on both sides

Why: make the decision boundary robust to small perturbations

The subset of vectors that support (determine boundary) are called the support vectors (circled)

Next Class

Classification III:

Regularization, stochastic gradient descent, multiclass SVM

Reading: Forsyth Ch 2.1.3-2.1.7