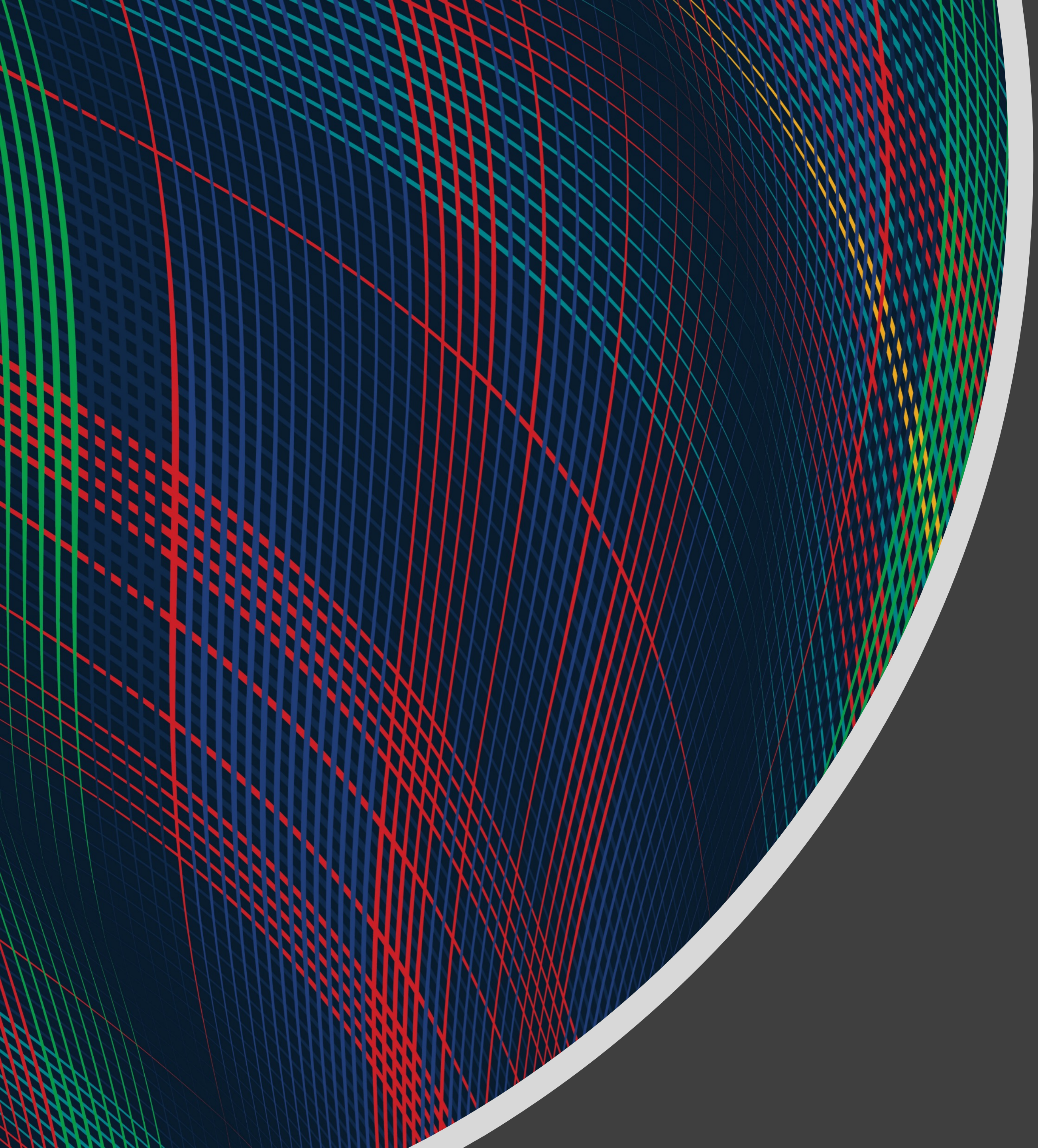


Introduction to Machine Learning

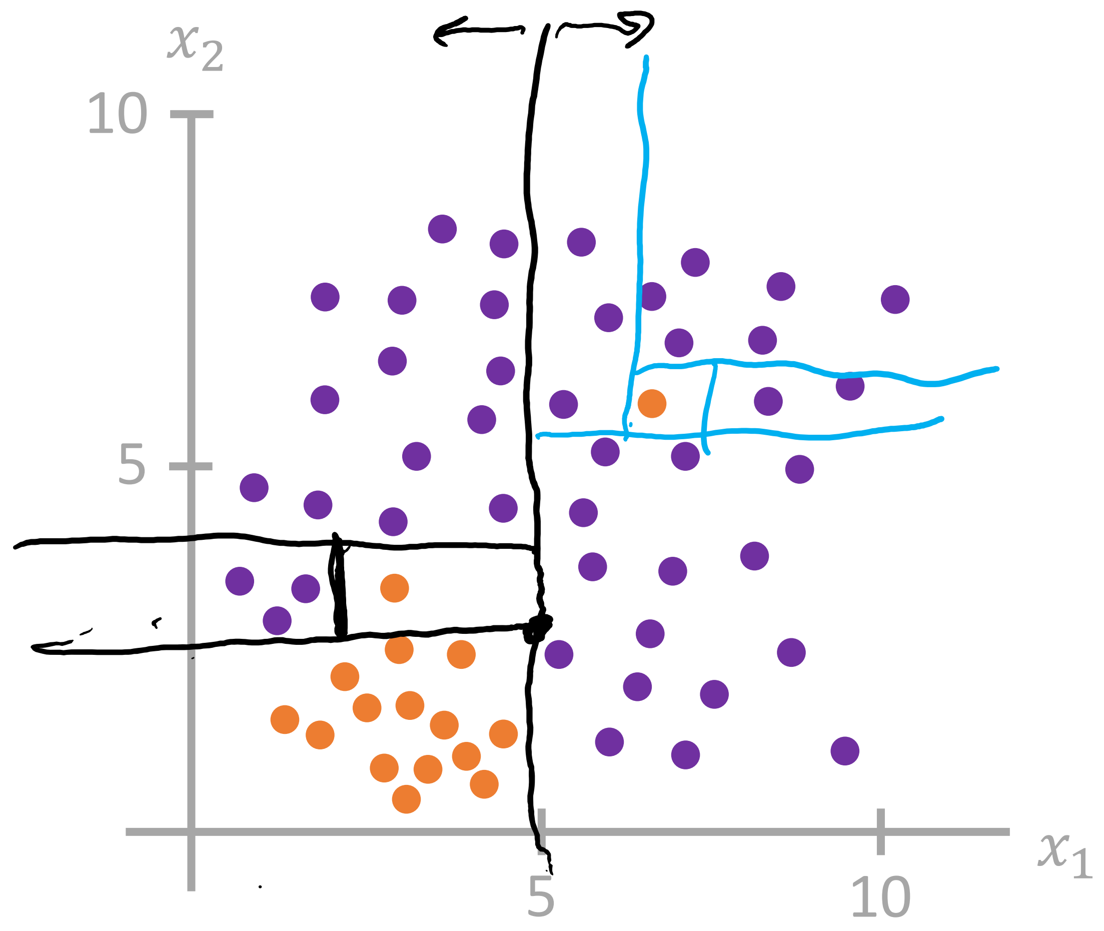
Nearest Neighbor and Model Selection

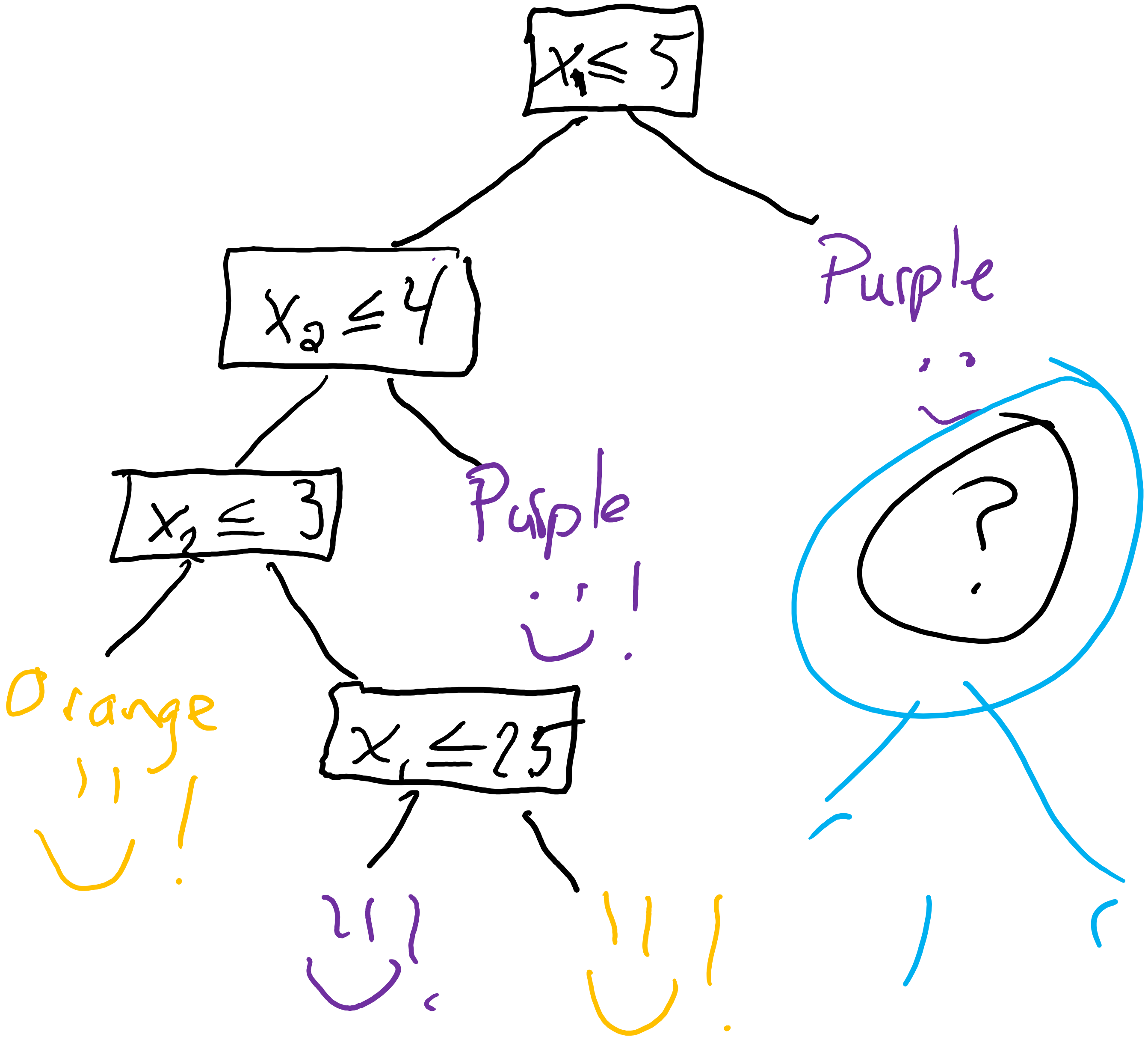
Instructor: Pat Virtue

Warm-up as you walk in

Consider input features $x \in \mathbb{R}^2$.

Draw a reasonable decision tree.





Poll 1

Decision tree generalization

Which of the following generalize best to unseen examples?

~~A.~~ Small tree with low training accuracy

~~B.~~ Large tree with low training accuracy

[✓ C.] Small tree with high training accuracy

D. Large tree with high training accuracy

Poll 1

Decision tree generalization

Which of the following generalize best to unseen examples?

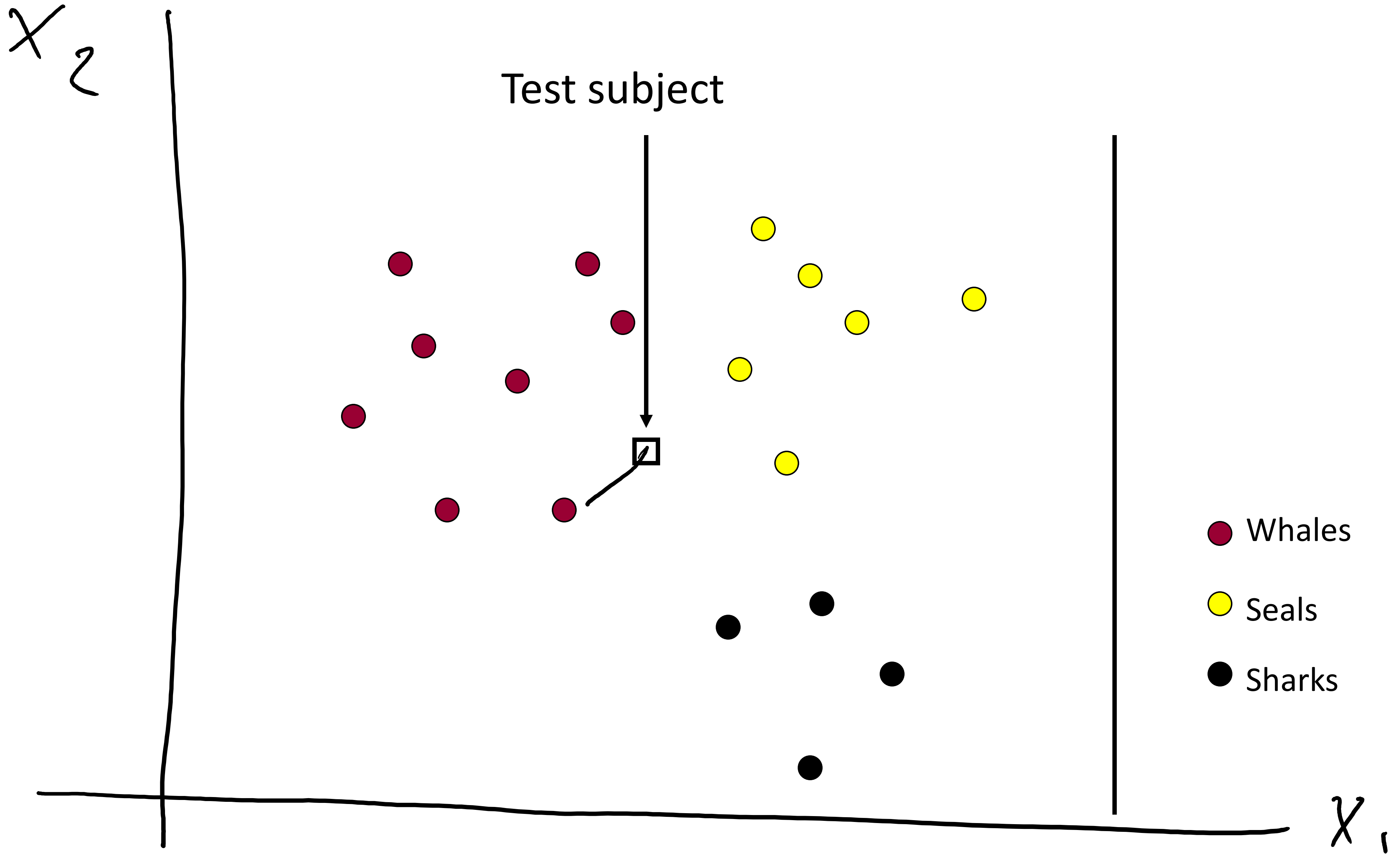
A. Small tree with low training accuracy
B. Large tree with low training accuracy
C. Small tree with high training accuracy
D. Large tree with high training accuracy

Poll 2

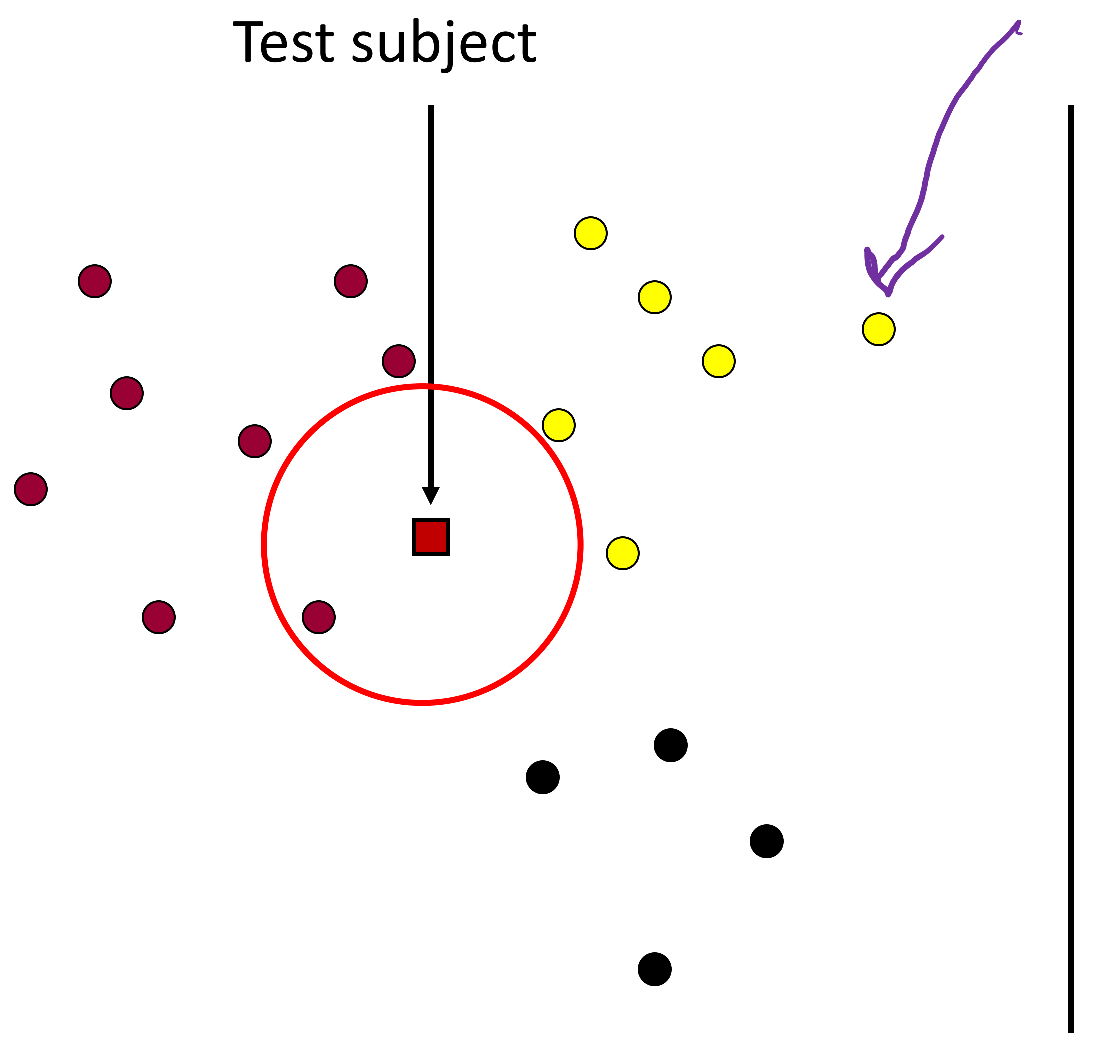
True or False:

For any dataset, you can find a decision tree that can perfectly classify the training data?

Nearest Neighbor Classifier



Nearest Neighbor Classifier



● Whales 1

● Seals 2

● Sharks 3

Nearest Neighbor Classification

Given a training dataset $\mathcal{D} = \{y^{(n)}, \boldsymbol{x}^{(n)}\}_{n=1}^{N}$, $y \in \{1, \dots, C\}$, $\boldsymbol{x} \in \mathbb{R}^M$

and a test input $\boldsymbol{x}_{test}$, predict the class label, $\hat{y}_{test}$:

1) Find the closest point in the training data to $\boldsymbol{x}_{test}$

$$n = \underset{n}{\operatorname{argmin}}\, \underline{d(\boldsymbol{x}_{test}, \boldsymbol{x}^{(n)})}$$

Could also use others, e.g. ℓ_1

$\downarrow$

Euclidean (ℓ_2) dist

2) Return the class label of that closest point

$$\hat{y}_{test} = \underline{y^{(n)}}$$

Need distance function! What should $d(\boldsymbol{x}, \boldsymbol{z})$ be?

2D

$$d(\vec{x}, \vec{z}) = \sqrt{\underline{(x_1 - z_1)^2} \overset{\uparrow}{+} \underline{(x_2 - z_2)^2}} = \left(\overset{\uparrow}{\sum_{j=1}^{M}} \underline{(x_j - z_j)^2}\right)^{1/2} = \|\vec{x} - \vec{z}\|_2$$

Fisher Iris Dataset

Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

	Species	Sepal Length	Sepal Width	Petal Length	Petal Width	
$n=1$	0	4.3	3.0	1.1	0.1	
$n=2$	0	4.9	3.6	1.4	0.1	
$n=3$	0	5.3	3.7	1.5	0.2	
	1	4.9	2.4	3.3	1.0	
$y^{(5)}$	$\boxed{1}$	5.7	2.8	4.1	1.3	$\vec{x}^{(5)}$
	1	6.3	3.3	4.7	1.6	
	1	6.7	3.0	5.0	1.7	

Full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set

Fisher Iris Dataset

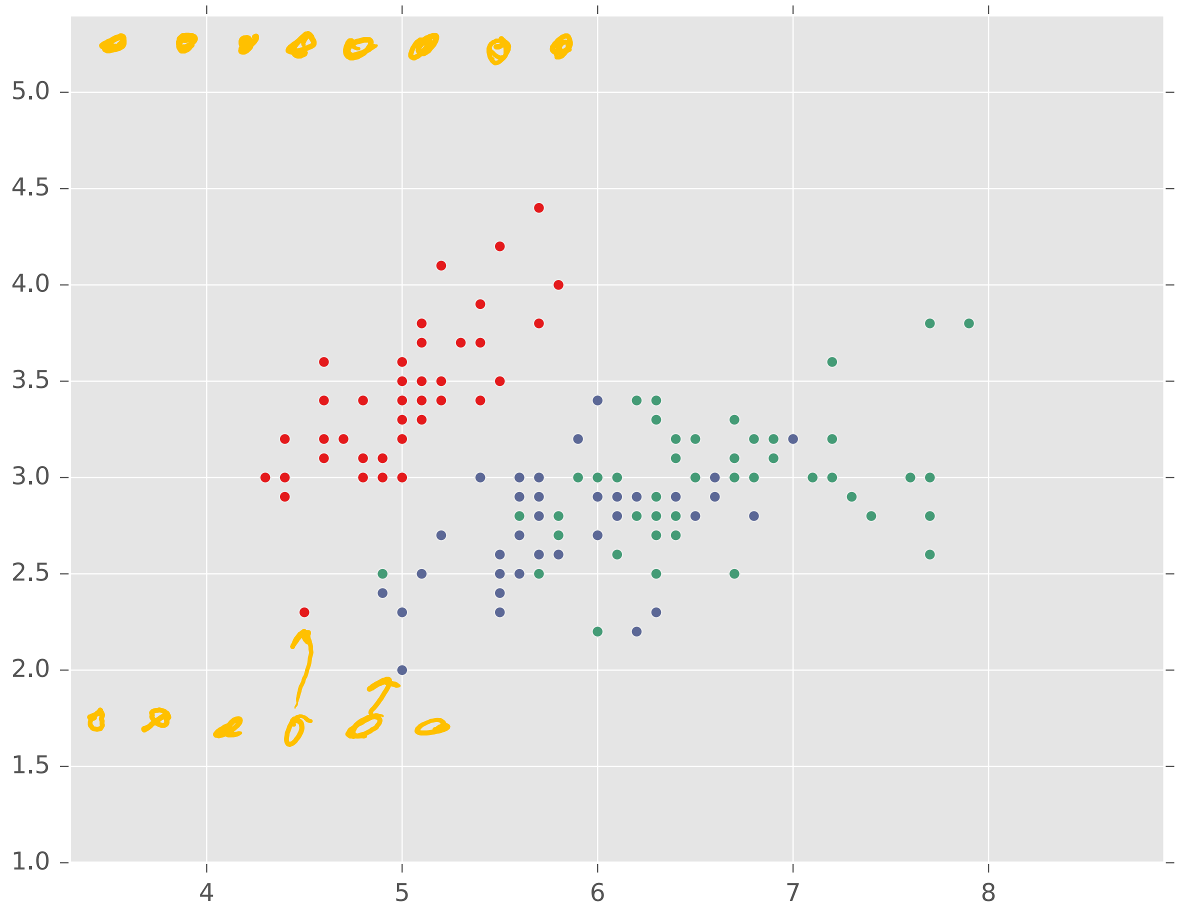
Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

Species	Sepal Length	Sepal Width
0	4.3	3.0
0	4.9	3.6
0	5.3	3.7
1	4.9	2.4
1	5.7	2.8
1	6.3	3.3
1	6.7	3.0

Deleted two of the four features, so that input space is 2D

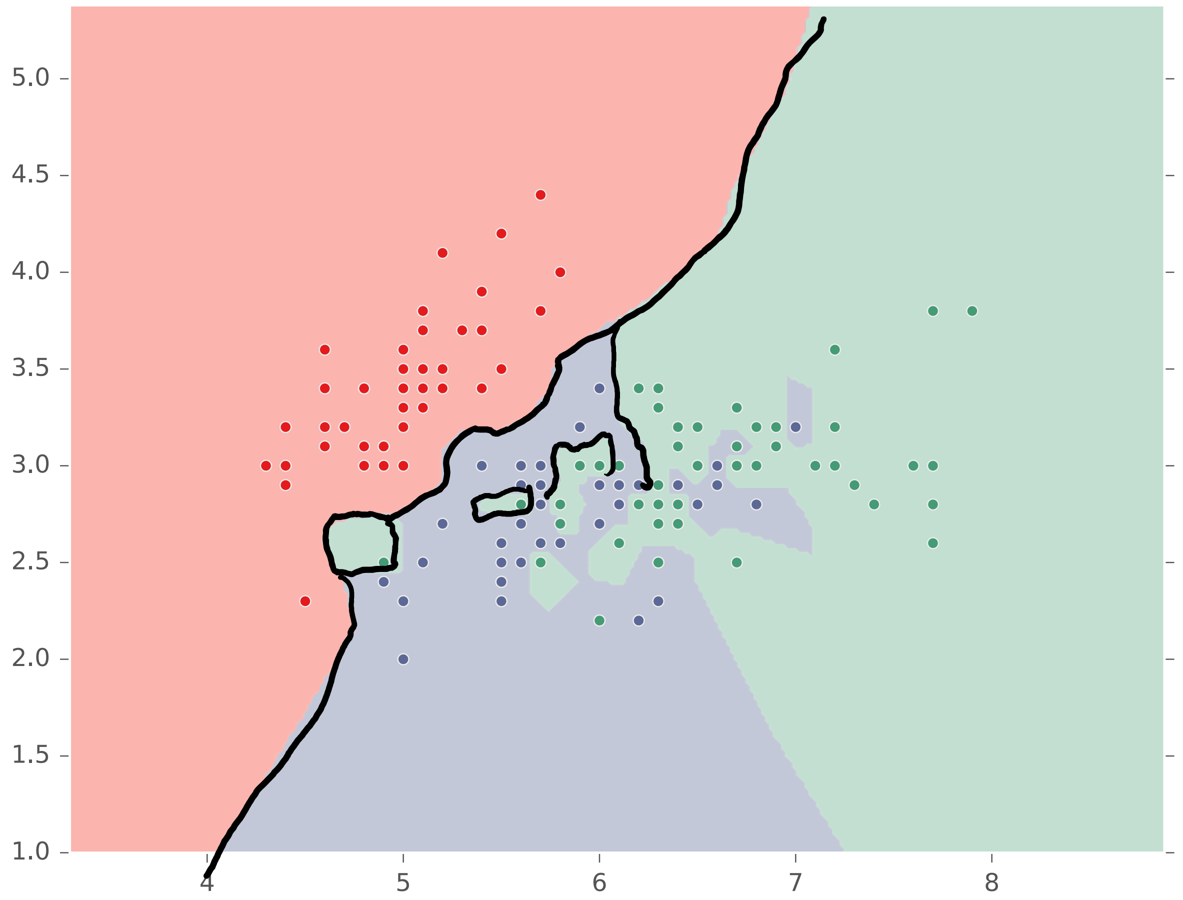
Full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set

Nearest Neighbor on Fisher Iris Data



Slide credit: CMU MLD Matt Gormley

Nearest Neighbor on Fisher Iris Data



Slide credit: CMU MLD Matt Gormley

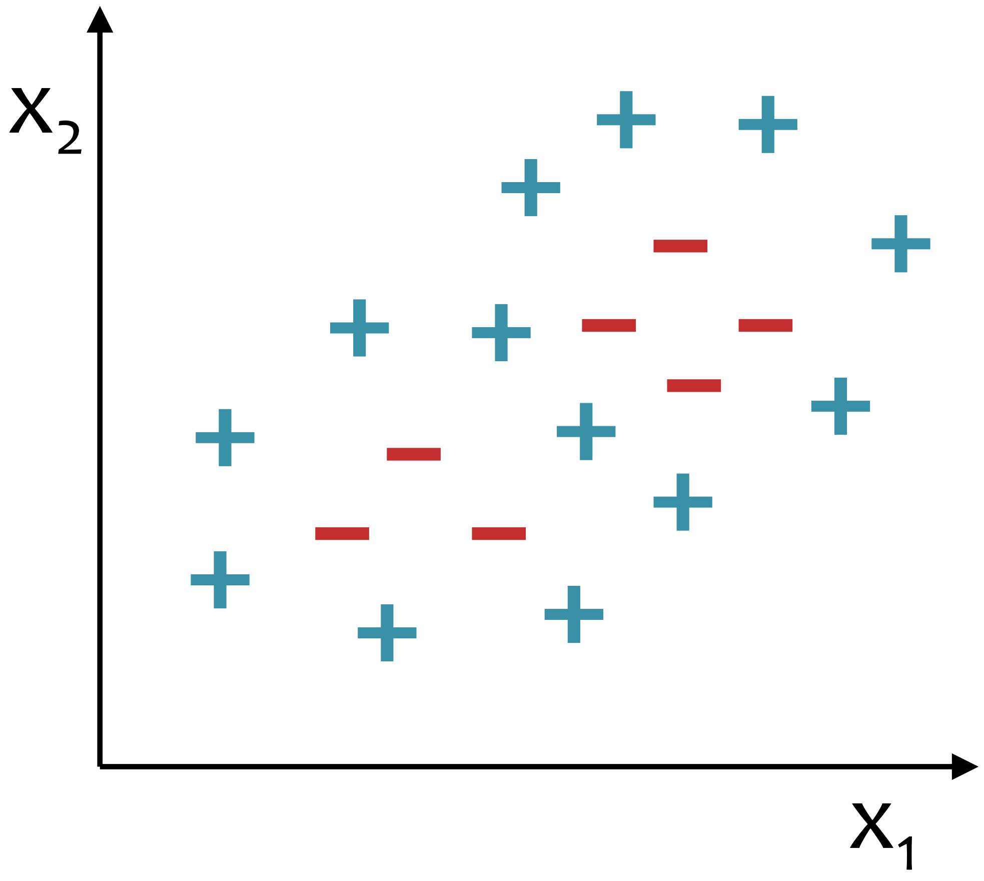
Poll 3

Which methods can achieve zero training error on this dataset?

- ✓ A. Decision trees
- ✓ B. 1-Nearest Neighbor
- ✓ C. Both
- D. Neither

If zero error, draw the decision boundary.

Otherwise, why not?



Poll 3

Which methods can achieve zero training error on this dataset?

A. Decision trees

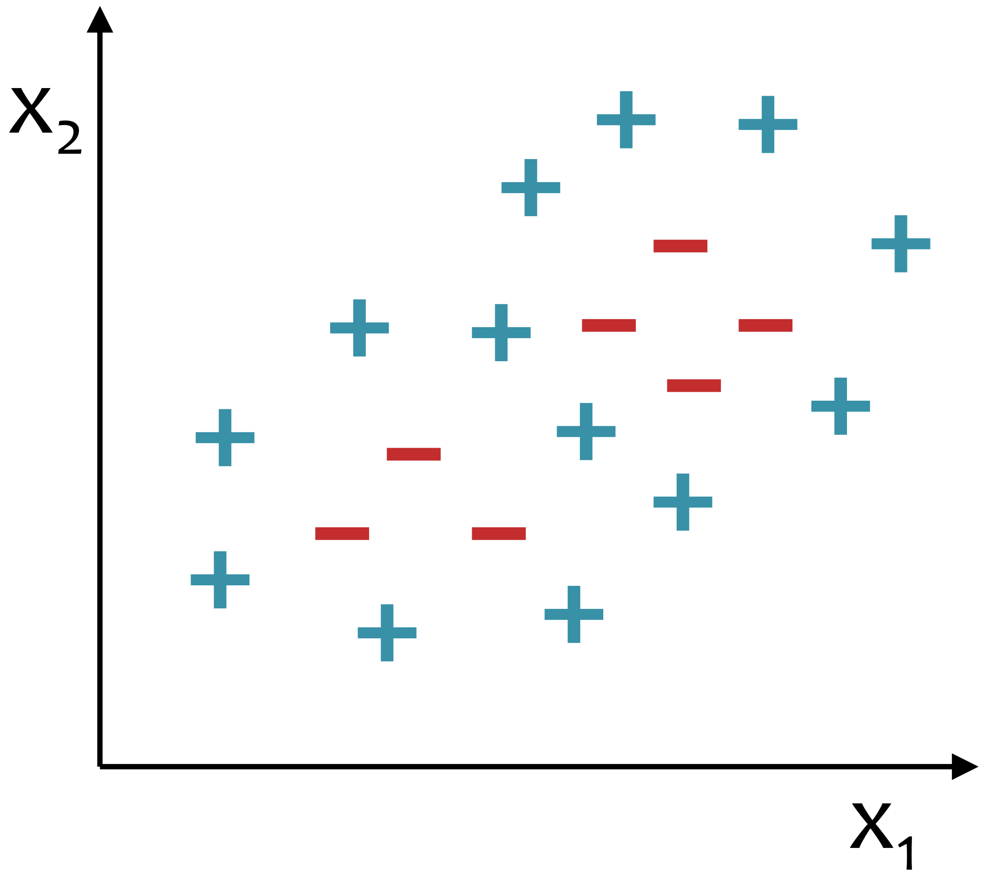
B. 1-Nearest Neighbor

C. Both

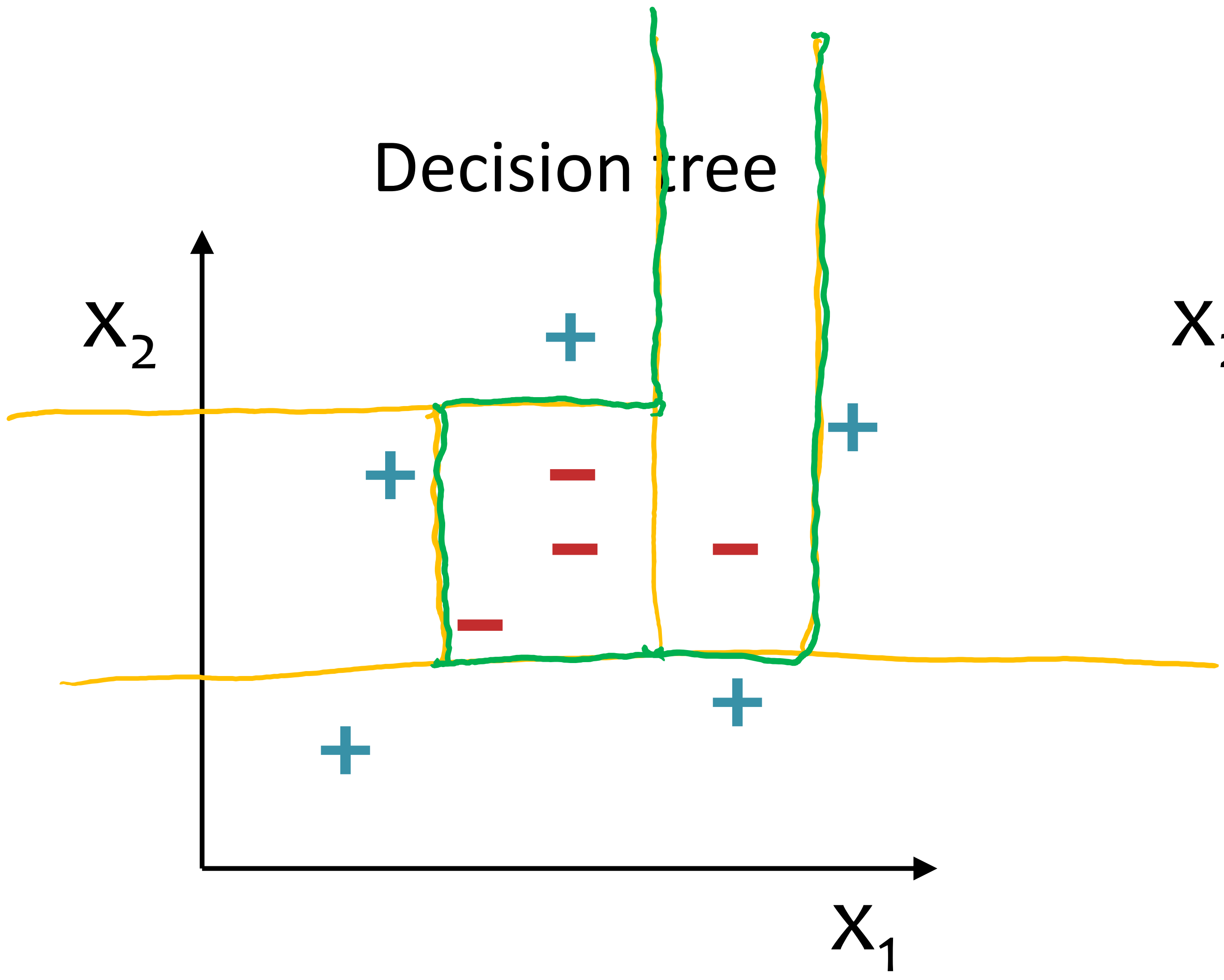
D. Neither

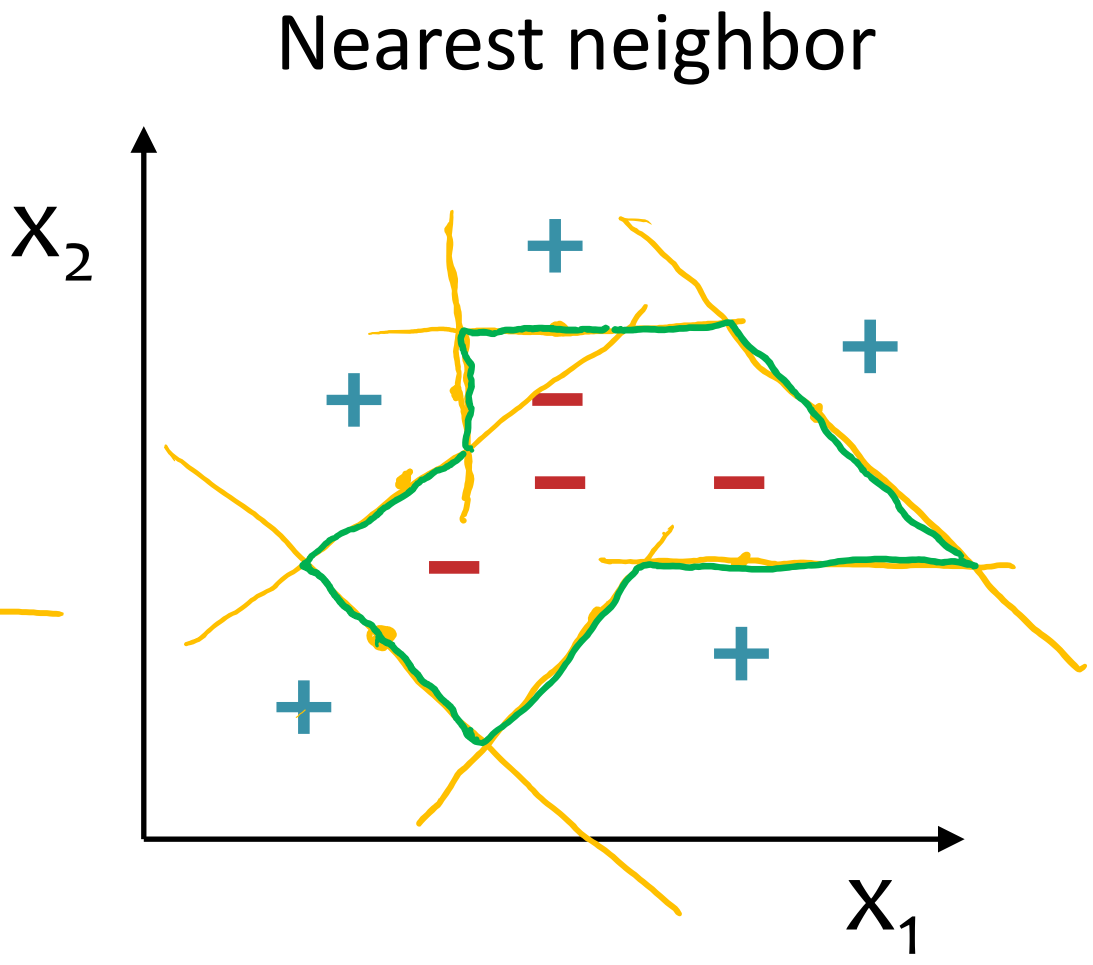
If zero error, draw the decision boundary.

Otherwise, why not?



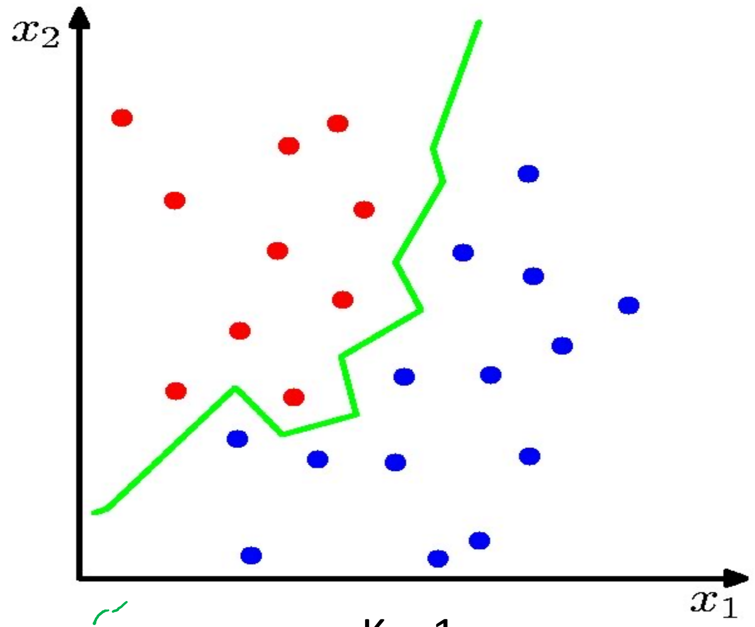
Decision Boundaries





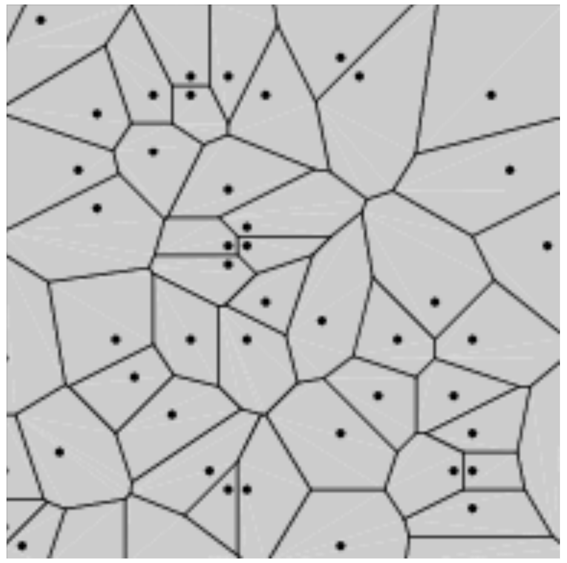
Nearest Neighbor Decision Boundary

1-nearest neighbor classifier decision boundary



K = 1

Voronoi Diagram



Poll 4

1-nearest neighbor will likely:

[✓ A. Overfit]

B. Underfit

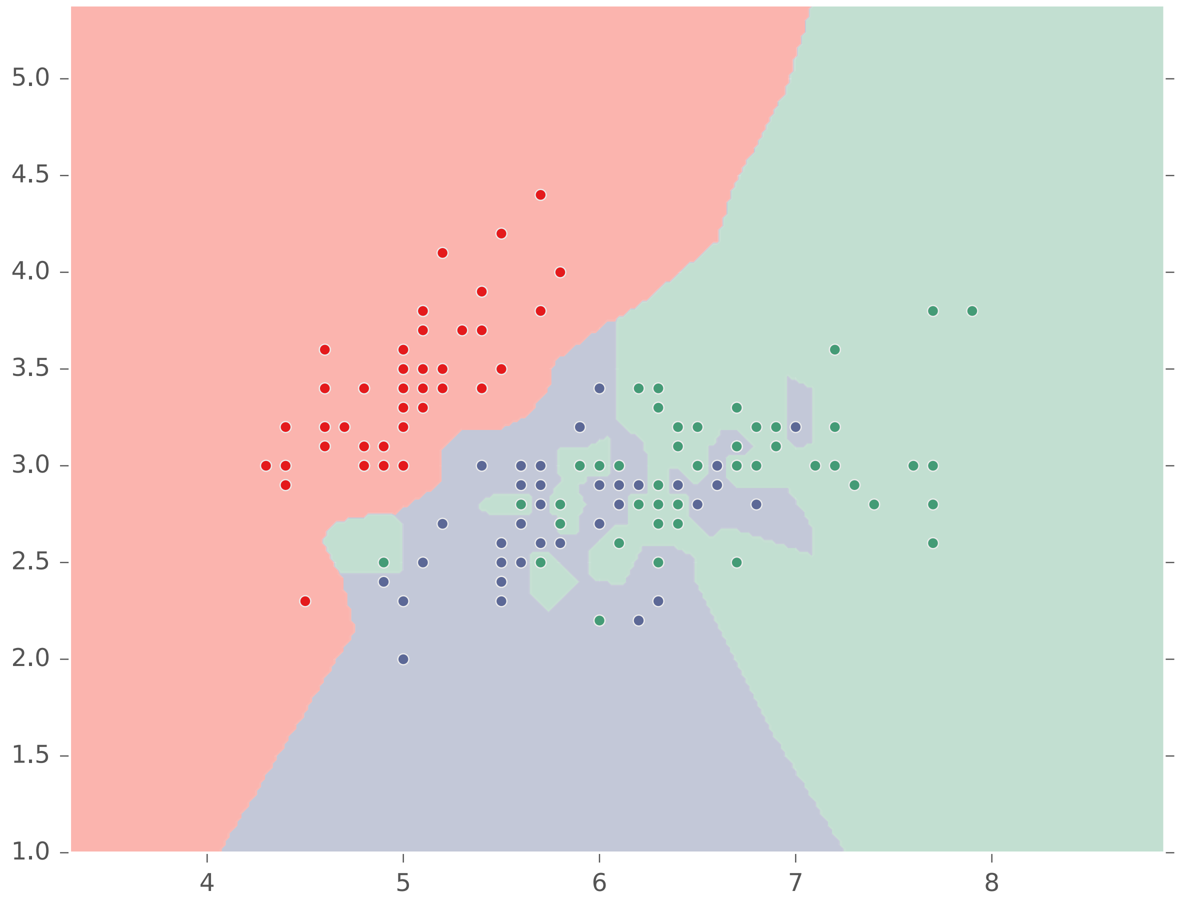
C. Neither (it's a great learner!)

Poll 4

1-Nearest neighbor will likely:

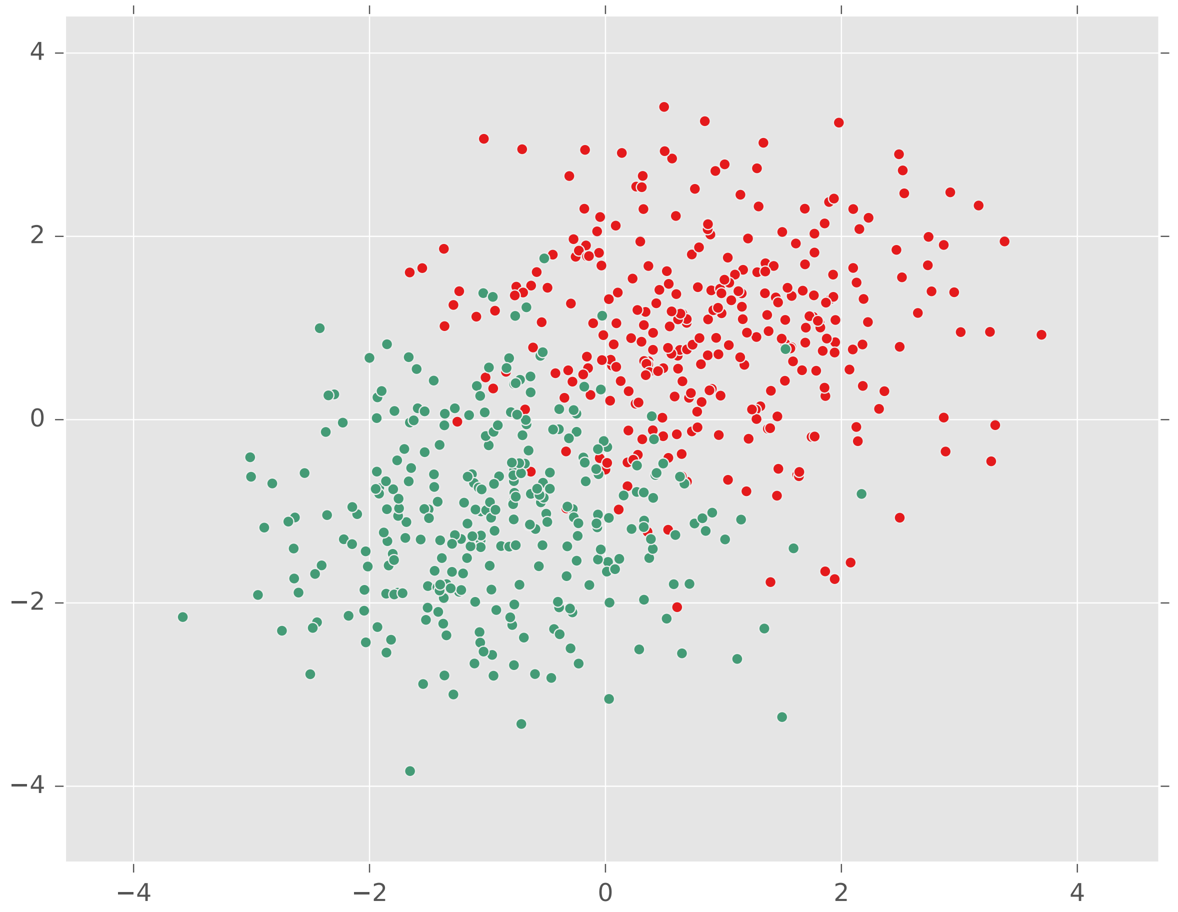
A. Overfit
B. Underfit
C. Neither (it’s a great learner!)

Nearest Neighbor on Fisher Iris Data



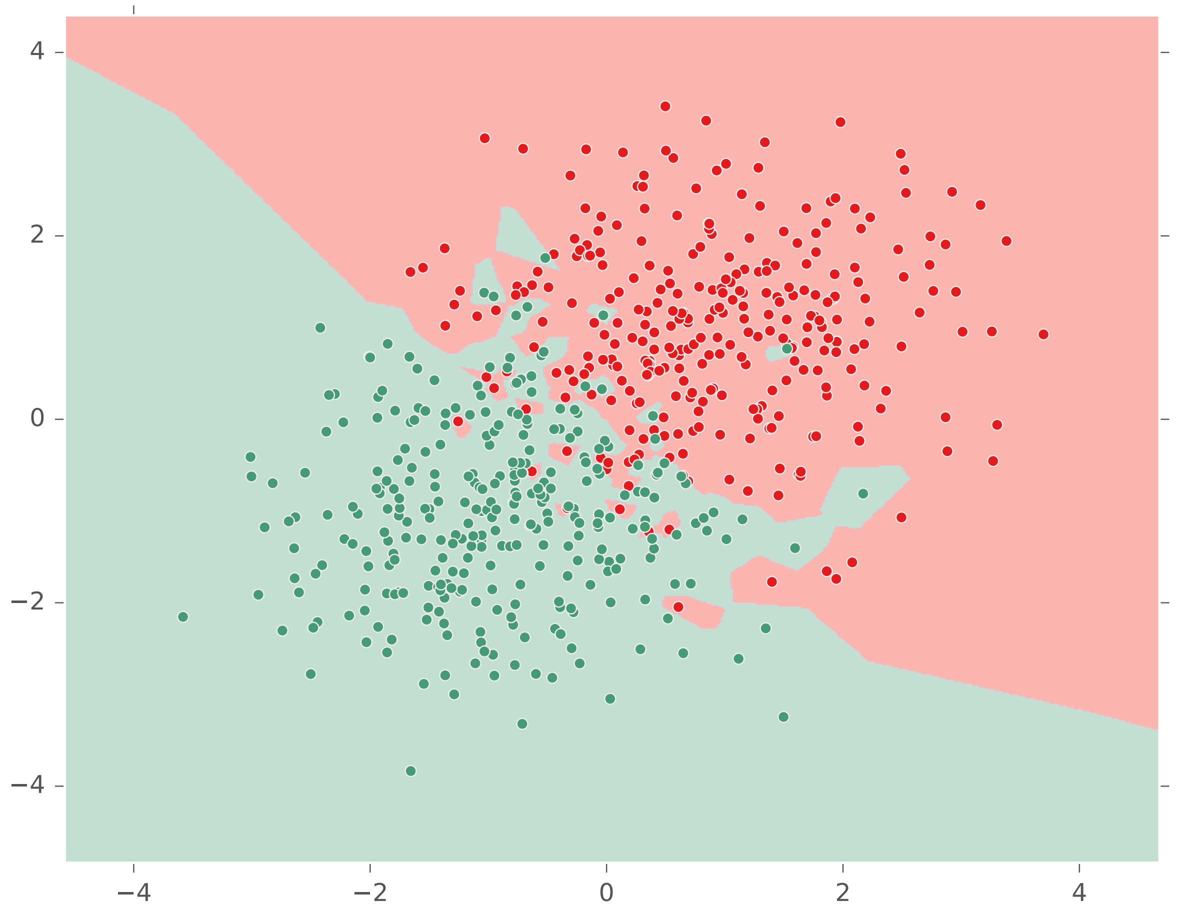
Slide credit: CMU MLD Matt Gormley

Nearest Neighbor on Gaussian Data



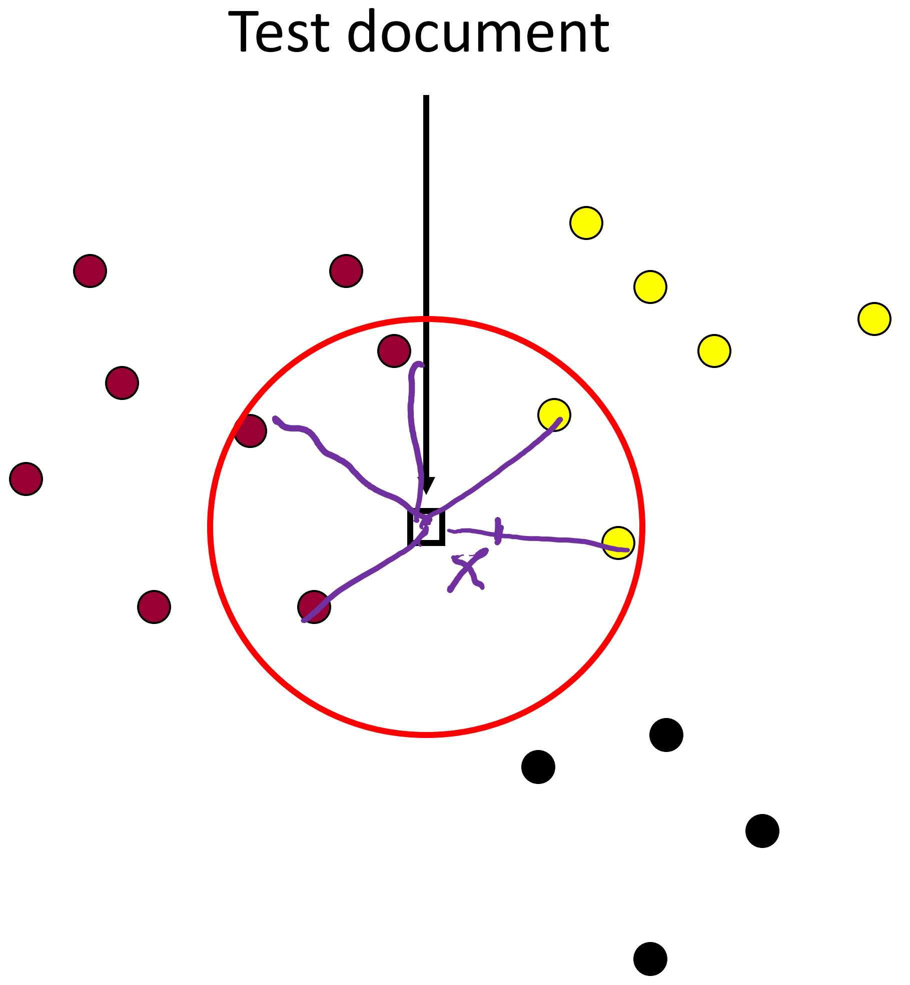
Slide credit: CMU MLD Matt Gormley

Nearest Neighbor on Gaussian Data



Slide credit: CMU MLD Matt Gormley

kNN classifier (k=5)



predict(x')
return red

0	●	Whales	3/5
1	●	Seals	2/5
2	●	Sharks	0/5

Nearest Neighbor Classification

Given a training dataset $\mathcal{D} = \{y^{(n)}, \boldsymbol{x}^{(n)}\}_{n=1}^{N}$, $y \in \{1, \ldots, C\}$, $\boldsymbol{x} \in \mathbb{R}^M$

and a test input $\boldsymbol{x}_{test}$, predict the class label, $\hat{y}_{test}$:

1) Find the closest point in the training data to $\boldsymbol{x}_{test}$

$$n = \underset{n}{\operatorname{argmin}}\, d(\boldsymbol{x}_{test}, \boldsymbol{x}^{(n)})$$

2) Return the class label of that closest point

$$\hat{y}_{test} = y^{(n)}$$

```
def train(D)
    data = D

def predict(x')
```

k-Nearest Neighbor Classification

Given a training dataset $\mathcal{D} = \{y^{(n)}, \boldsymbol{x}^{(n)}\}_{n=1}^{N}$, $y \in \{1, \dots, C\}$, $\boldsymbol{x} \in \mathbb{R}^M$

and a test input $\boldsymbol{x}_{test}$, predict the class label, $\hat{y}_{test}$:

1) Find the closest k points in the training data to $\boldsymbol{x}_{test}$

$$\mathcal{N}_k(\boldsymbol{x}_{test}, \mathcal{D}) \longleftarrow$$

2) Return the class label of that closest point

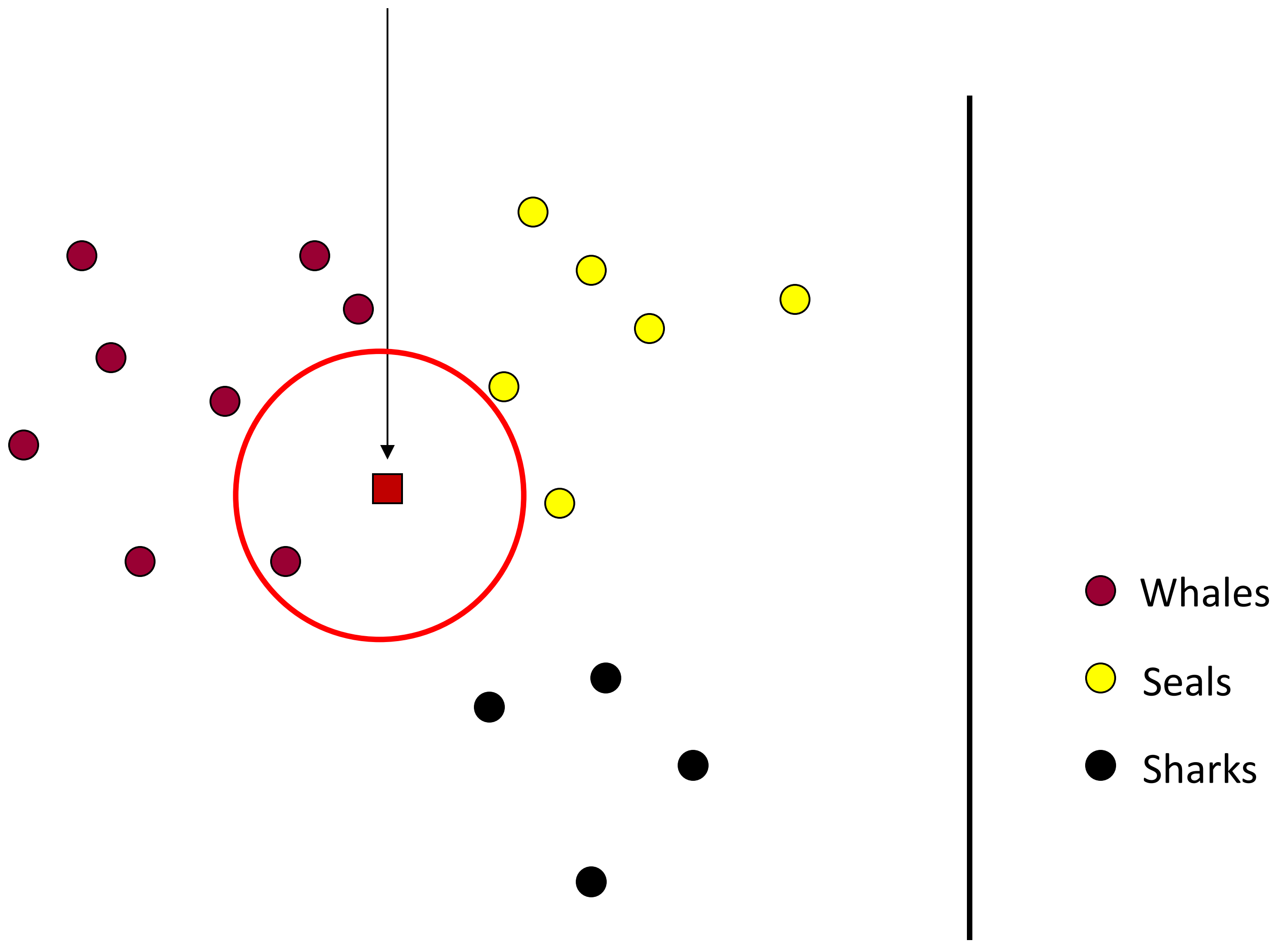
$$\hat{y}_{test} = \underset{c}{\operatorname{argmax}}\, p(Y = c \mid \boldsymbol{x}_{test}, \mathcal{D}, k) \longleftarrow$$

$$= \underset{c}{\operatorname{argmax}} \frac{1}{k} \sum_{i \in \mathcal{N}_k(\boldsymbol{x}_{test}, \mathcal{D})} \mathbb{1}(y^{(i)} = c)$$

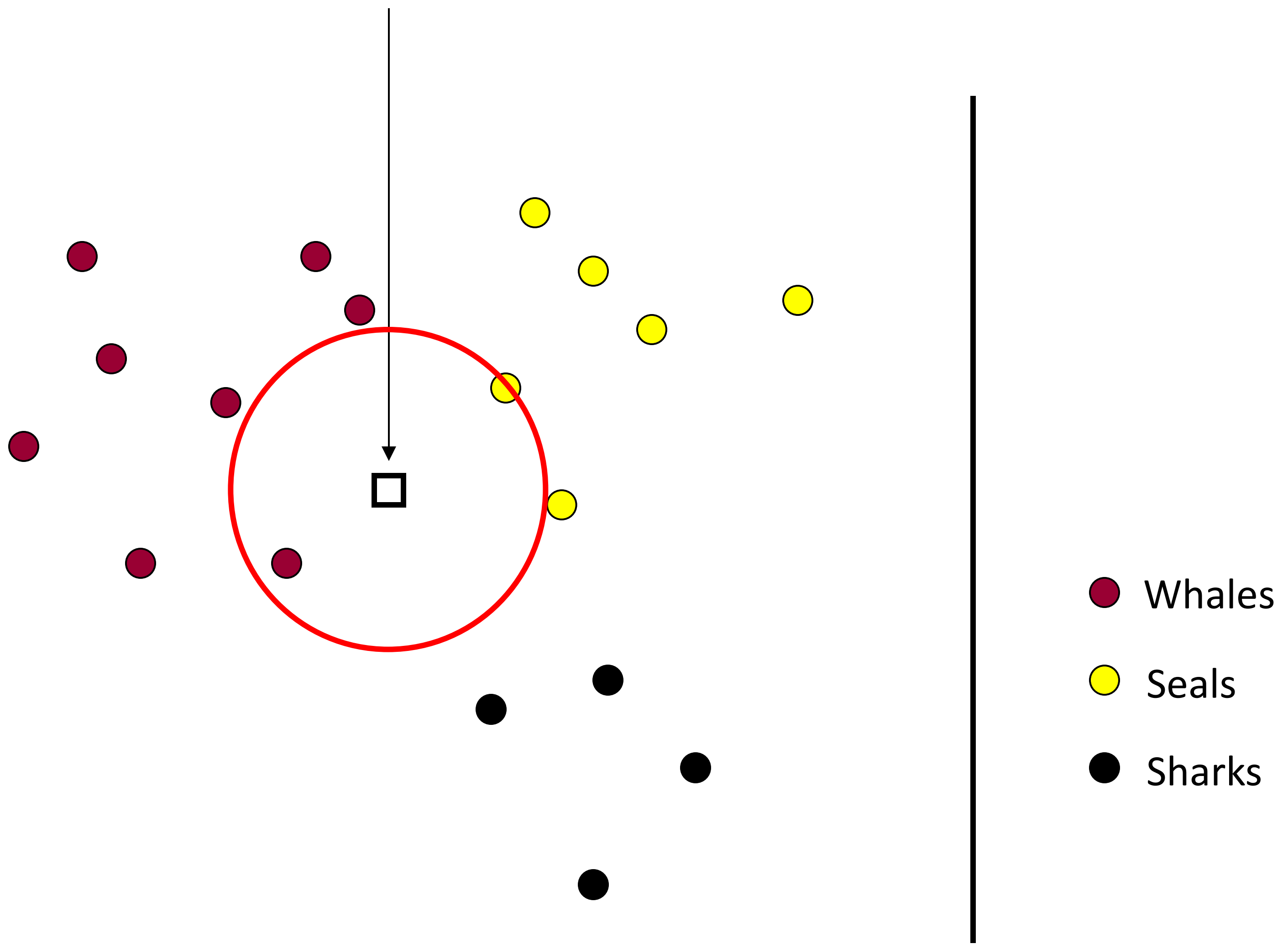
$$= \underset{c}{\operatorname{argmax}} \frac{k_c}{k},$$

where k_c is the number of the k-neighbors with class label c

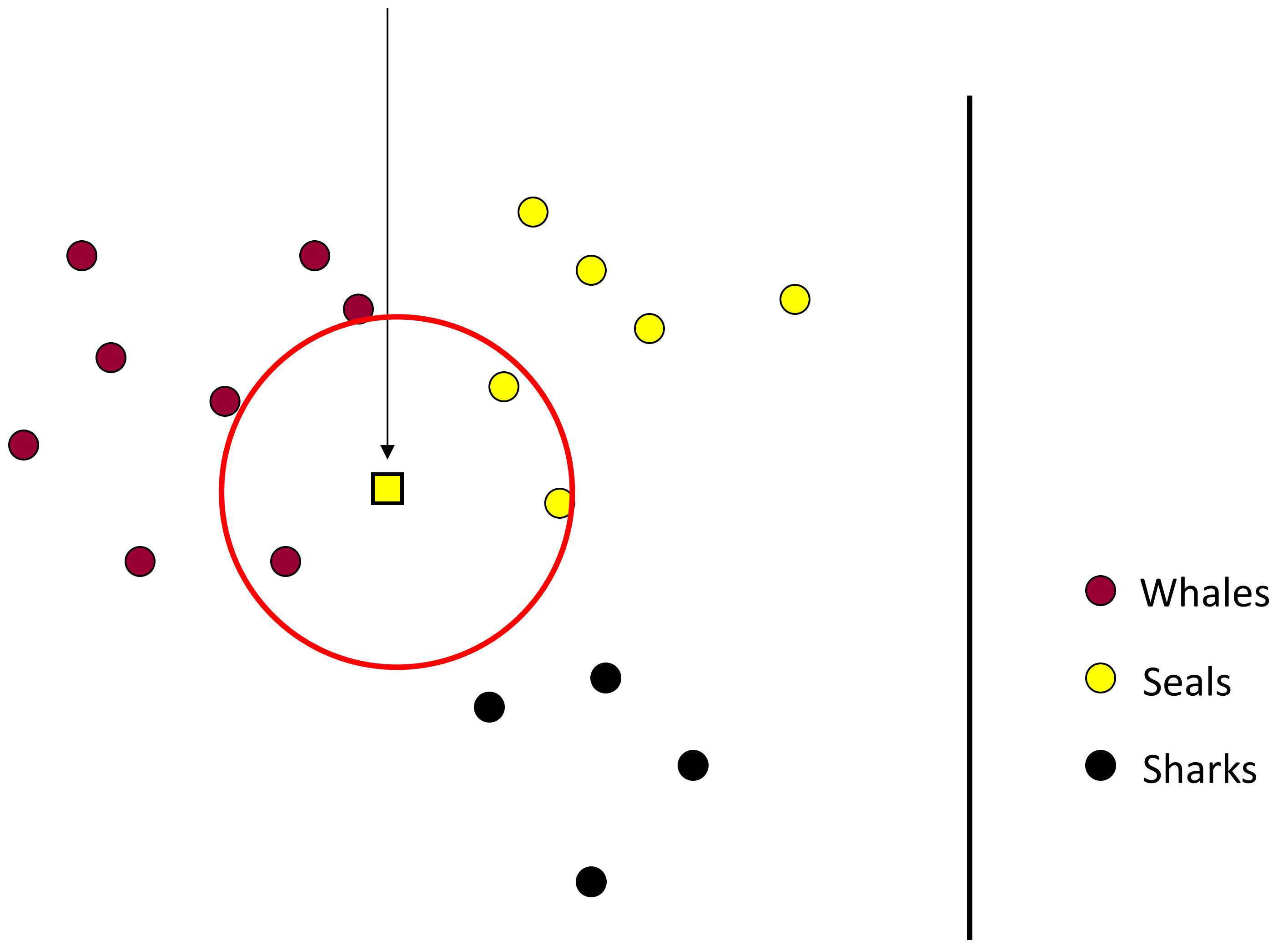
1-Nearest Neighbor (kNN) classifier



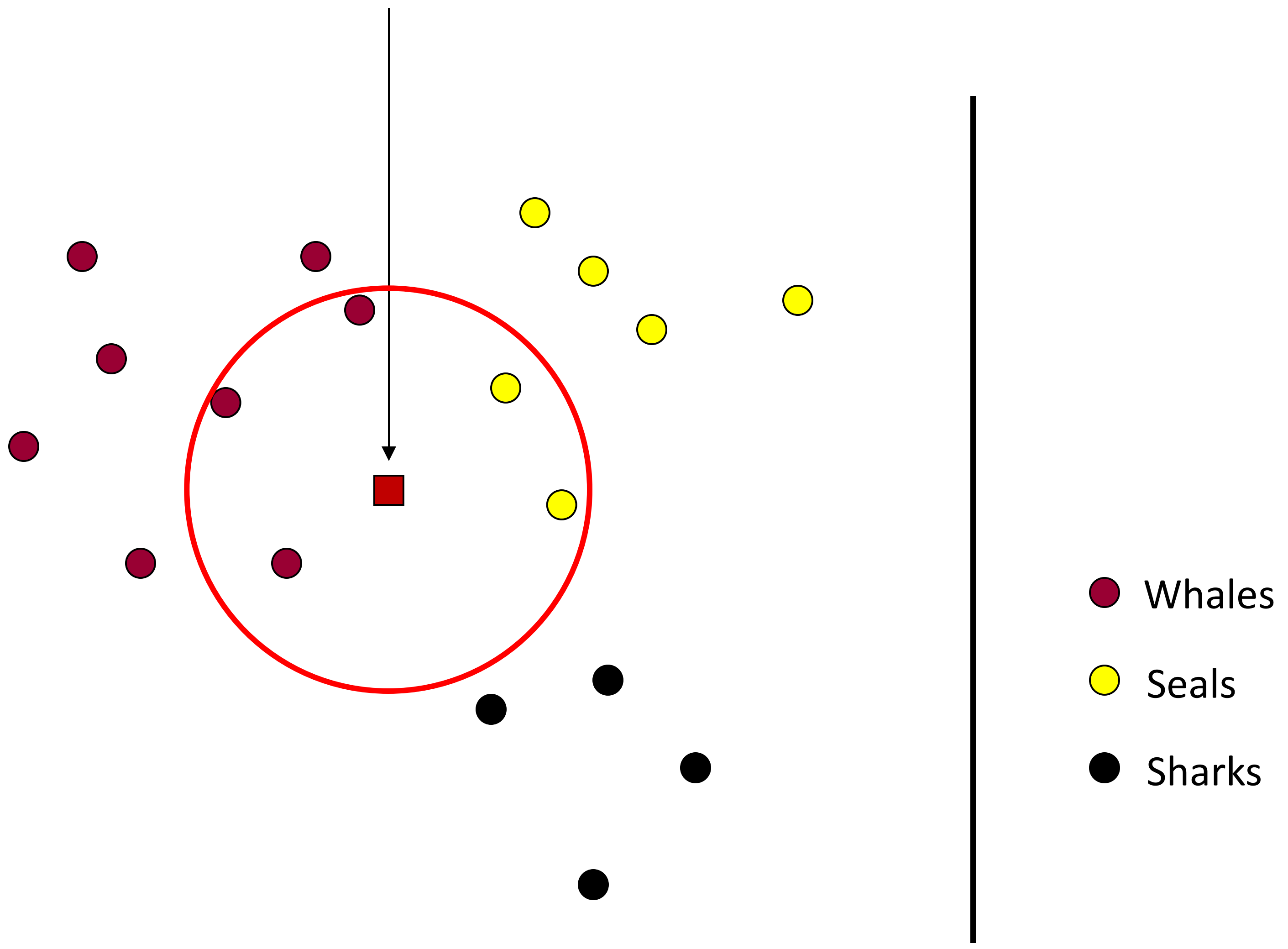
2-Nearest Neighbor (kNN) classifier



3-Nearest Neighbor (kNN) classifier



5-Nearest Neighbor (kNN) classifier



What is the best k?

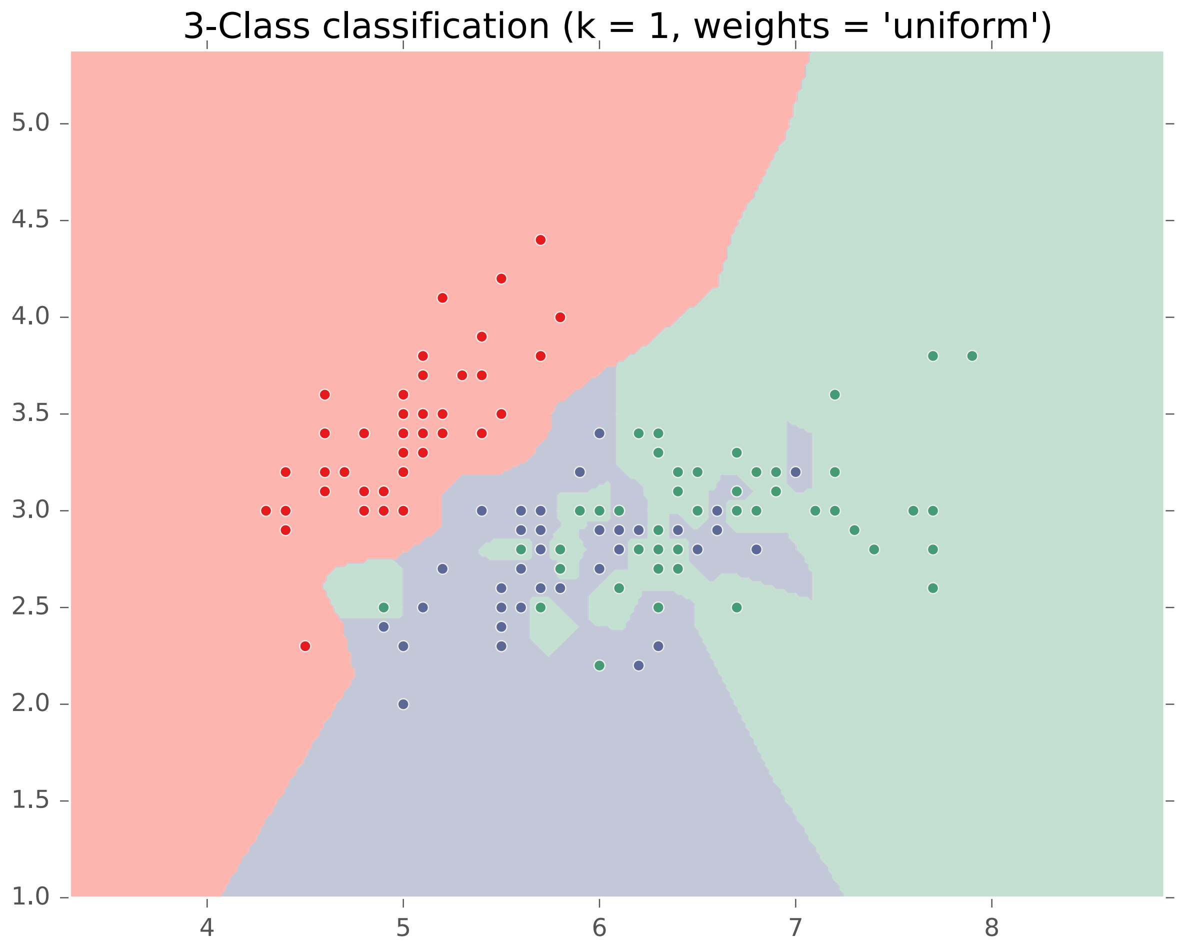
How do we choose a learner that is accurate and also generalizes to unseen data?

- Larger k → predicted label is more stable
- Smaller k → predicted label is more affected by individual training points

But how to choose k?

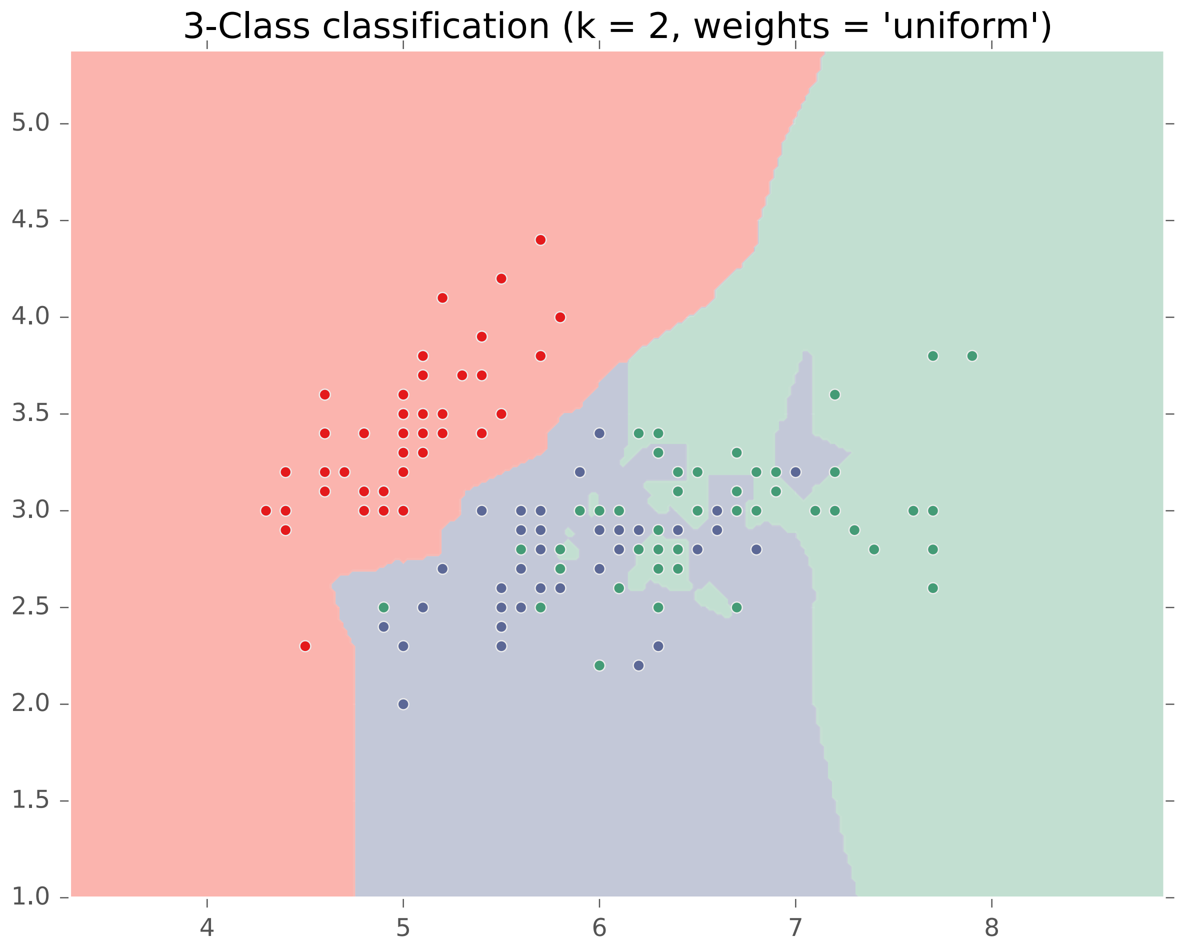
k-NN on Fisher Iris Data

Special Case: Nearest Neighbor



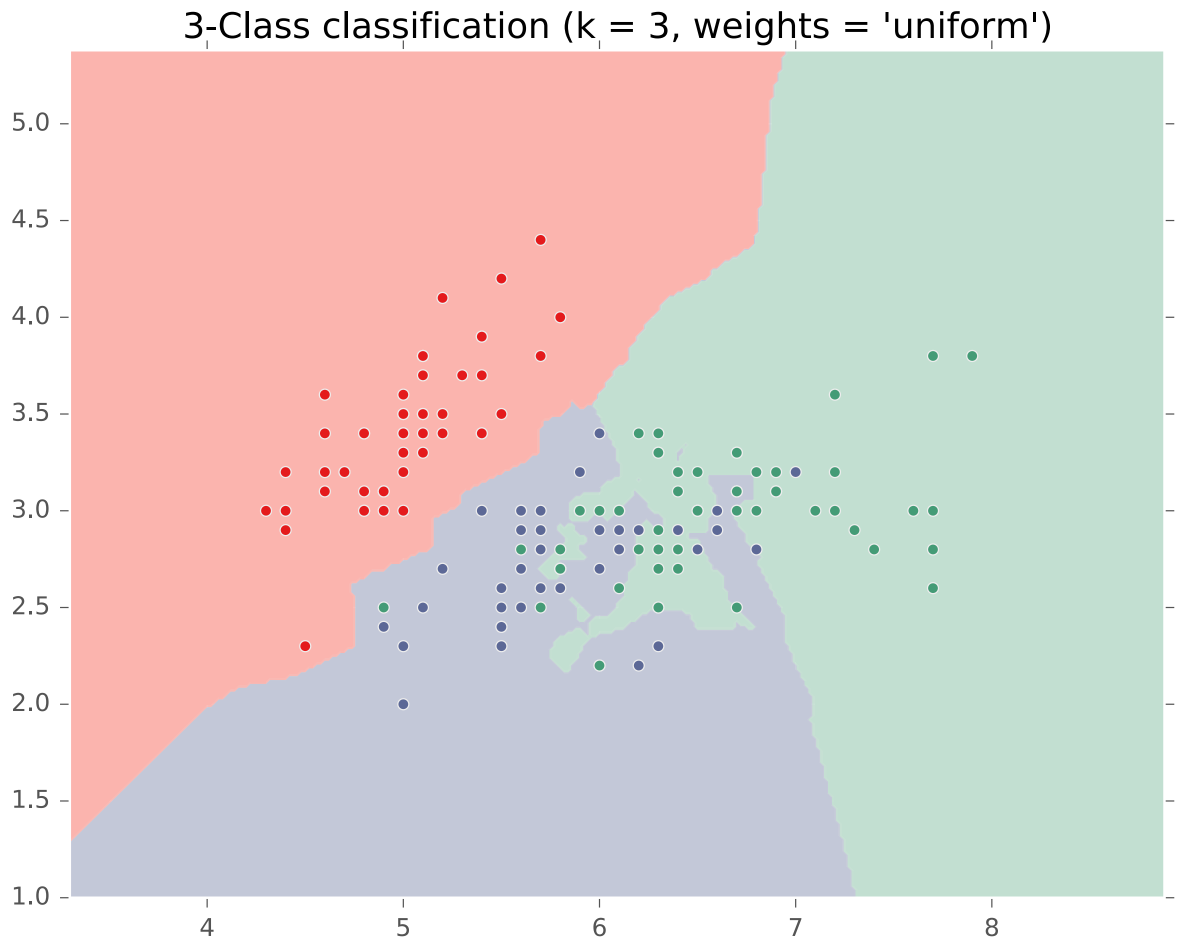
Slide credit: CMU MLD Matt Gormley

k-NN on Fisher Iris Data



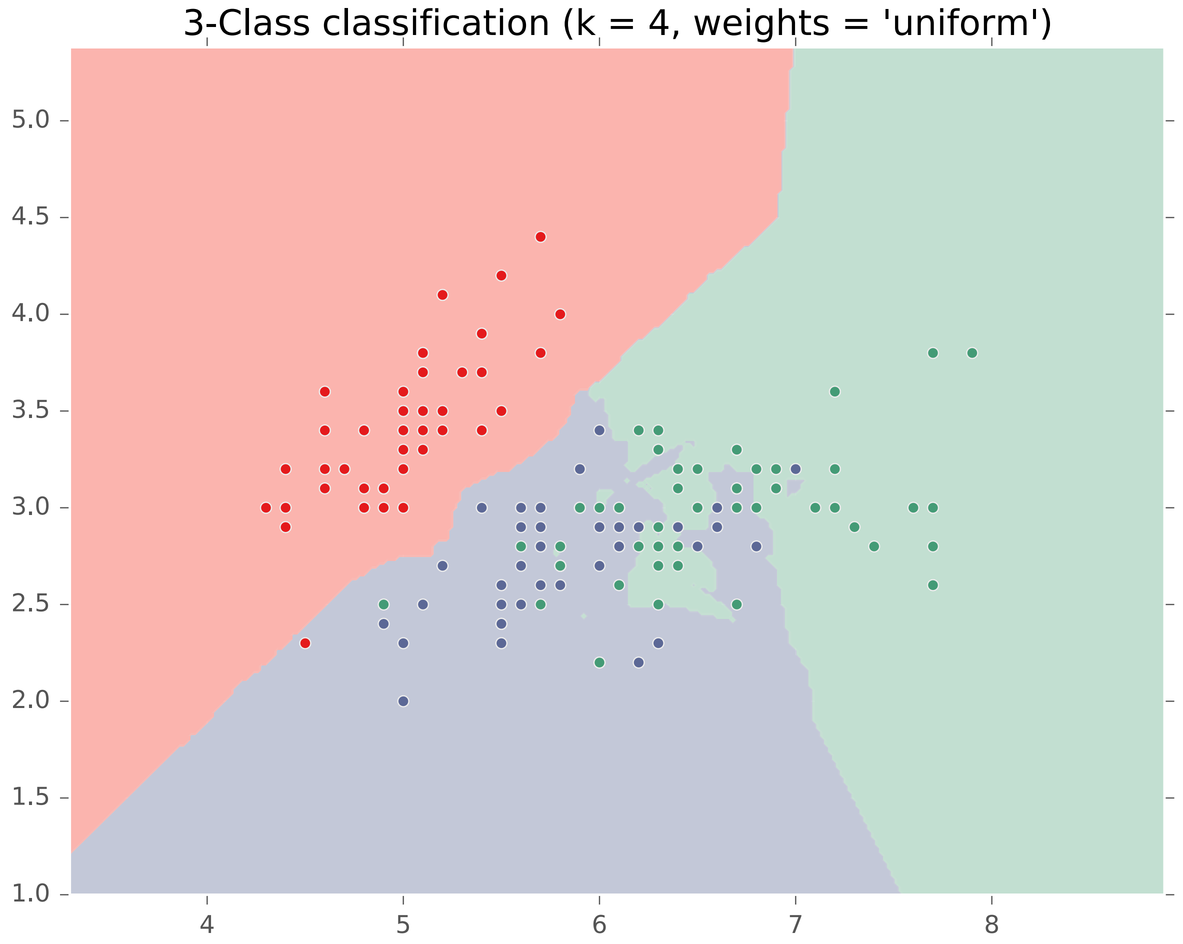
Slide credit: CMU MLD Matt Gormley

k-NN on Fisher Iris Data



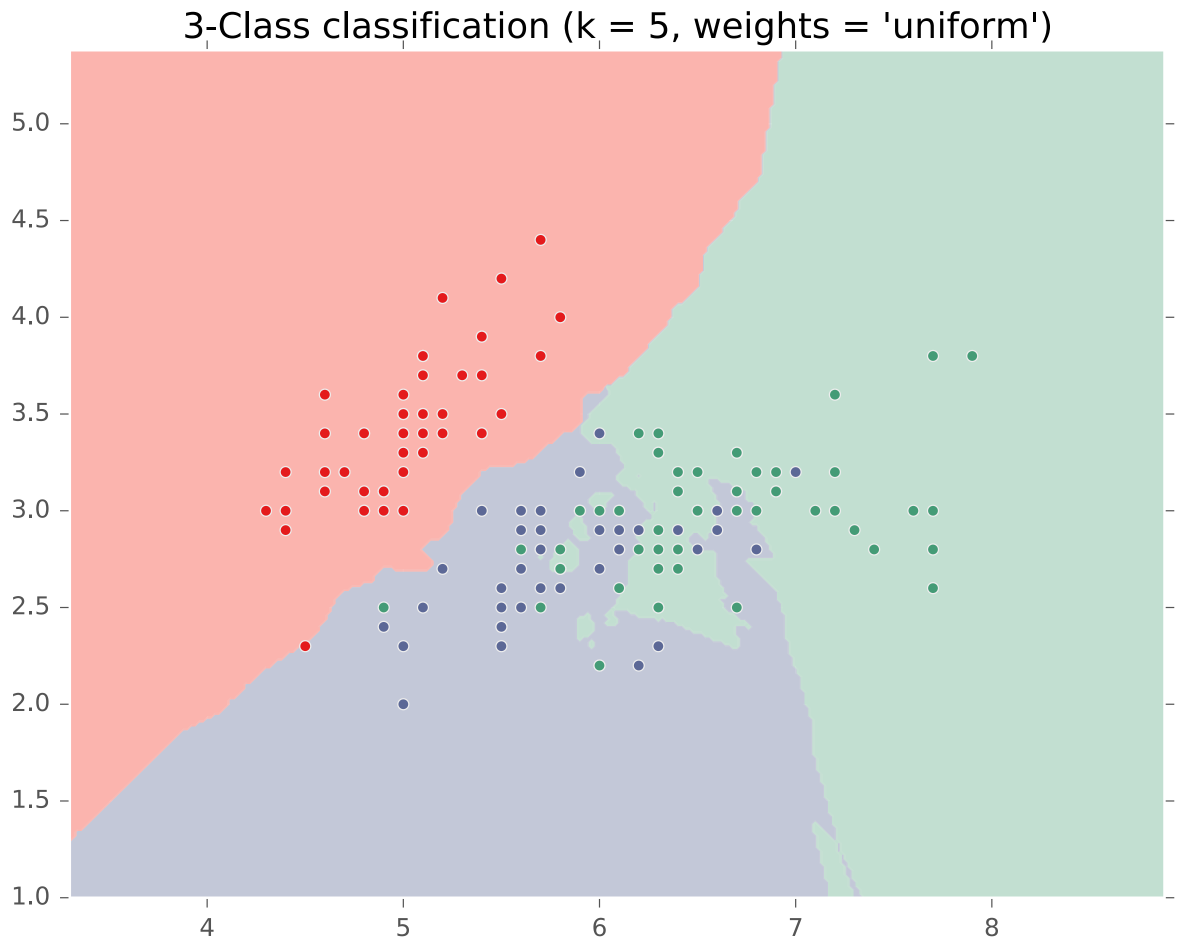
Slide credit: CMU MLD Matt Gormley

k-NN on Fisher Iris Data



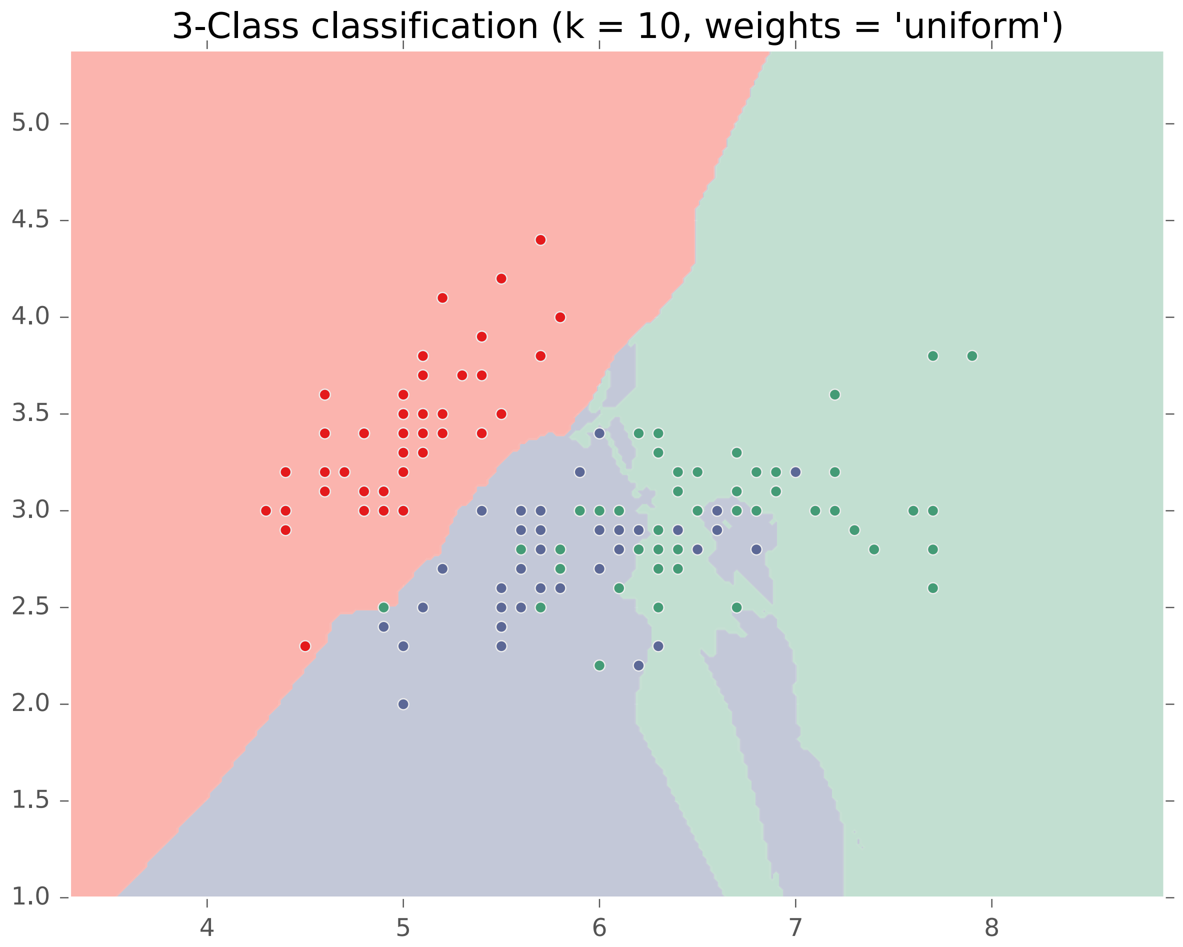
Slide credit: CMU MLD Matt Gormley

k-NN on Fisher Iris Data



Slide credit: CMU MLD Matt Gormley

k-NN on Fisher Iris Data

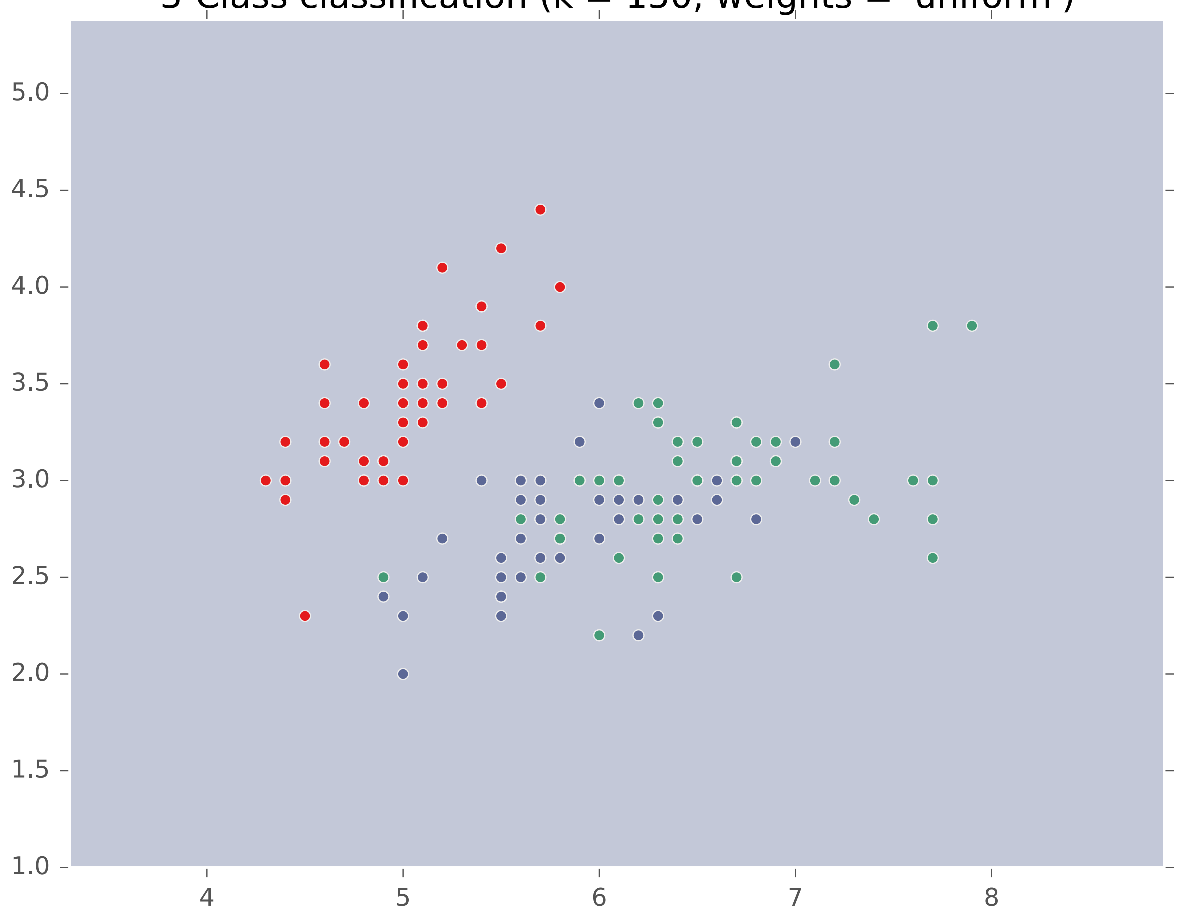


Slide credit: CMU MLD Matt Gormley

k-NN on Fisher Iris Data

Special Case: Majority Vote

3-Class classification (k = 150, weights = 'uniform')



Slide credit: CMU MLD Matt Gormley

k-NN: Remarks

Inductive Bias:

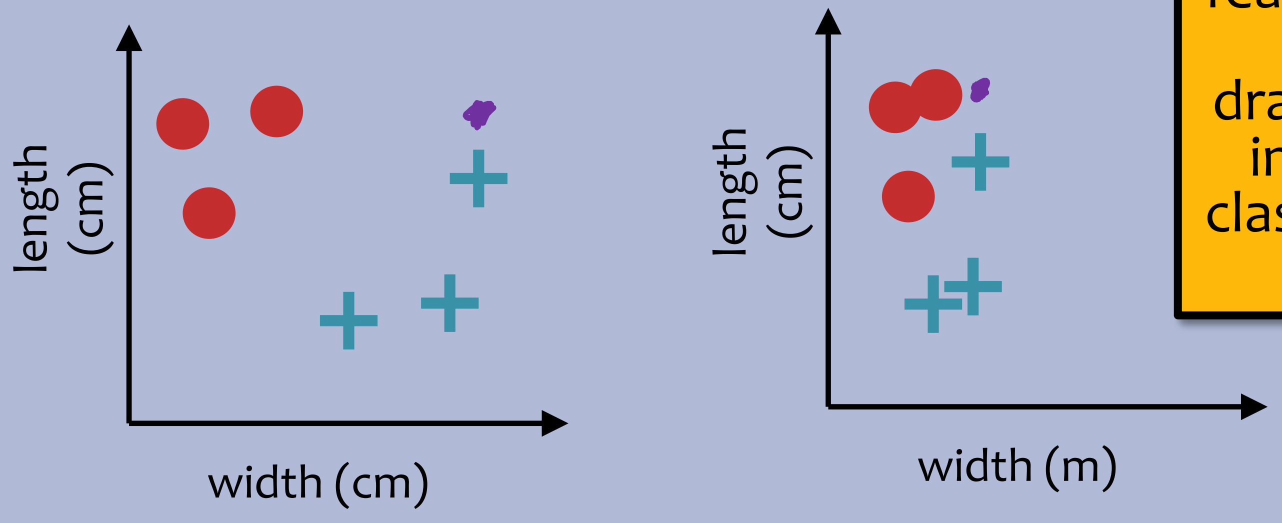
1. Close points should have similar labels
2. All dimensions are created equally!

k-NN: Remarks

Inductive Bias:

1. Close points should have similar labels
2. All dimensions are created equally!

Example: two features for k-NN



big problem: feature scale could dramatically influence classification results

Slide credit: CMU MLD Matt Gormley

k-NN: Remarks

Computational Efficiency:

- Suppose we have N training examples, and each one has M features
- Computational complexity for the special case where k=1:

def train(D)
 store D

Poll 5 (train) and Poll 6 (predict)

Suppose we have N training examples, and each one has M features
Computational complexity for the special case where k=1:

[✓ A.] O(1)

B. O(log N)

C. O(log M)

D. O(log NM)

E. O(N)

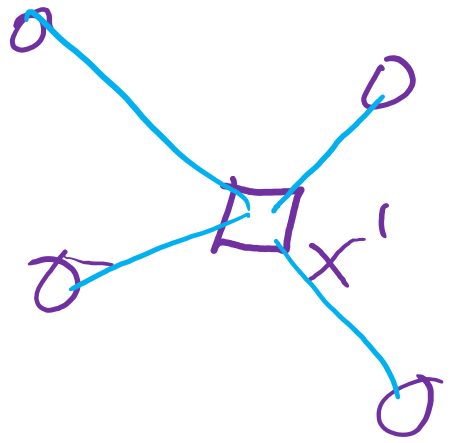
F. O(M)

[✓ G.] O(NM)

H. O(N^2)

I. O(N^2M)

```
def predict($\vec{x}'$)
    for i in 1..N
        $d(\vec{x}^{(i)}, \vec{x}')$
```



```
def $d(\vec{x}, \vec{z})$
    for j in 1..M
```

$$d(x,z) = \left(\sum_{j=1}^{M} (x_j - x_j)^2\right)^{1/2}$$

Poll 5 (train) and Poll 6 (predict)

Suppose we have N training examples, and each one has M features
Computational complexity for the special case where k=1:

A. O(1)
B. O(log N)
C. O(log M)
D. O(log NM)
E. O(N)
F. O(M)
G. O(NM)
H. O(N^2)
I. O(N^2M)

k-NN: Remarks

Computational Efficiency:

- Suppose we have N training examples, and each one has M features
- Computational complexity for the special case where k=1:

Task	Naive	k-d Tree
Train	O(1)	~ O(M N log N)
Predict (one test example)	O(MN)	~ O(2^M log N) on average

Problem: Very fast for small M, but very slow for large M

In practice: use stochastic approximations (very fast, and empirically often as good)

MODEL SELECTION

Model Selection

WARNING:

- In some sense, our discussion of model selection is premature.
- The models we have considered thus far are fairly simple.
- The models and the many decisions available to the data scientist wielding them will grow to be much more complex than what we've seen so far.

Model Selection

Statistics

- *Def*: a **model** defines the data generation process (i.e. a set or family of parametric probability distributions)
- *Def*: **model parameters** are the values that give rise to a particular probability distribution in the model family
- *Def*: **learning** (aka. estimation) is the process of finding the parameters that best fit the data
- *Def*: **hyperparameters** are the parameters of a prior distribution over parameters

Machine Learning

- *Def*: (loosely) a **model** defines the hypothesis space over which learning performs its search
- *Def*: **model parameters** are the numeric values or structure selected by the learning algorithm that give rise to a hypothesis
- *Def*: the **learning algorithm** defines the data-driven search over the hypothesis space (i.e. search for good parameters)
- *Def*: **hyperparameters** are the tunable aspects of the model, that the learning algorithm does *not* select

Slide credit: CMU MLD Matt Gormley

Model Selection

Example: Decision Tree

- model = set of all possible trees, possibly restricted by some hyperparameters (e.g. max depth)
- parameters = structure of a specific decision tree
- learning algorithm = ID3, CART, etc.
- hyperparameters = max-depth, threshold for splitting criterion, etc.

Machine Learning

- *Def*: (loosely) a **model** defines the hypothesis space over which learning performs its search human
- *Def*: **model parameters** are the numeric values or structure selected by the learning algorithm that give rise to a hypothesis human data ↳ alg
- *Def*: the **learning algorithm** defines the data-driven search over the hypothesis space (i.e. search for good parameters) alg
- *Def*: **hyperparameters** are the tunable aspects of the model, that the learning algorithm does *not* select ?

Model Selection

Example: k-Nearest Neighbors

- model = set of all possible nearest neighbors classifiers
- parameters = none
(KNN is an instance-based or non-parametric method)
- learning algorithm = for naïve setting, just storing the data
- hyperparameters = *k*, the number of neighbors to consider

Machine Learning

- *Def*: (loosely) a **model** defines the hypothesis space over which learning performs its search
- *Def*: **model parameters** are the numeric values or structure selected by the learning algorithm that give rise to a hypothesis
- *Def*: the **learning algorithm** defines the data-driven search over the hypothesis space (i.e. search for good parameters)
- *Def*: **hyperparameters** are the tunable aspects of the model, that the learning algorithm does *not* select

Model Selection

Statistics

- *Def*: a **model** defines the data generation process (i.e. a set or family probability distributions)
- *Def*: **model parameters** are give rise to a particular pro distribution in the model fa
- *Def*: **learning** (aka. estimation) is the process of finding the paramete hat best fit the data
- *Def*: **hyperparameters** are the parameters of a prior distribution over parameters

Machine Learning

- *Def*: (loosely) a **model** defines the hypothesis which learning performs its
- **parameters** are the numeric ructure selected by the learning hat give rise to a hypothesis
- *Def*: the **learning algorithm** defines the data-driven searc ver the hypothesis space (i.e. search for goo arameters)
- *Def*: **hyperparameters** are the tunable aspects of the model, that the learning algorithm does *not* select

If “learning” is all about picking the best **parameters** how do we pick the best **hyperparameters?**

Model Selection

- Two *very* similar definitions:
  - *Def*: **model selection** is the process by which we choose the "best" model from among a set of candidates
  - *Def*: **hyperparameter optimization** is the process by which we choose the "best" hyperparameters from among a set of candidates **(could be called a special case of model selection)**

- **Both** assume access to a function capable of measuring the quality of a model

- **Both** are typically done "outside" the main training algorithm --- typically training is treated as a black box

Slide credit: CMU MLD Matt Gormley

Experimental Design

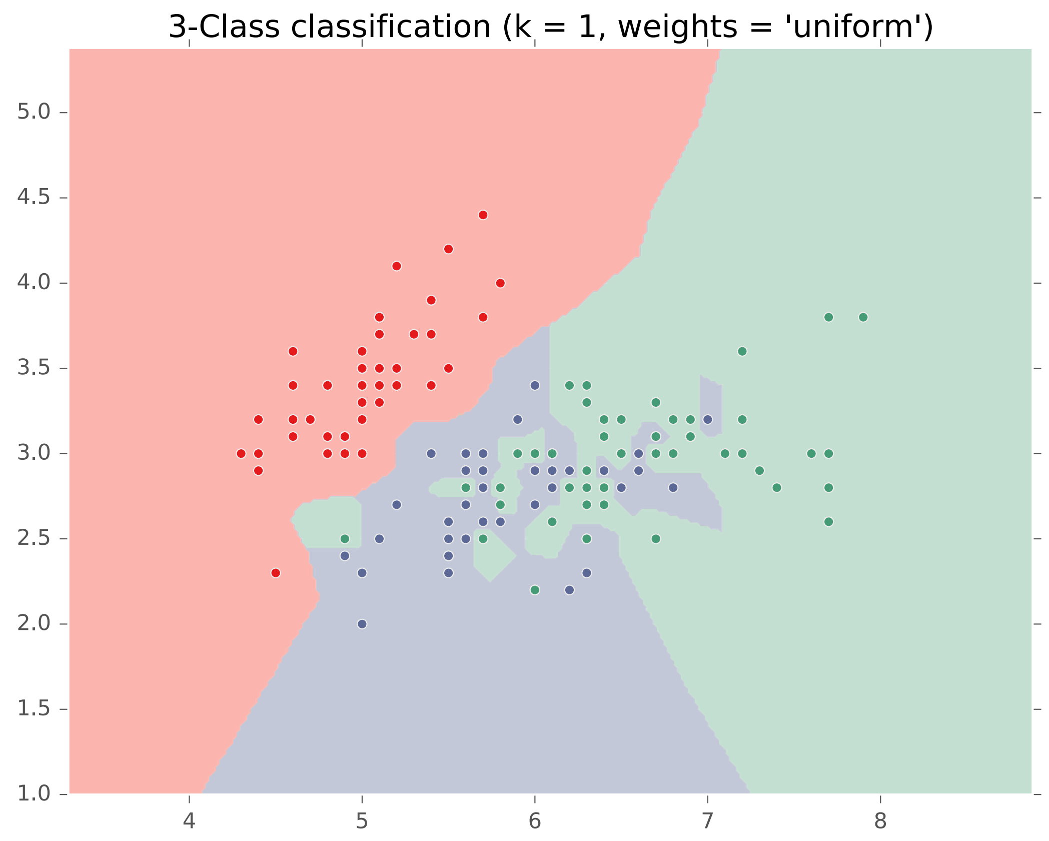
	Input	Output	Notes
Training	• training dataset • hyperparameters	• best model parameters	We pick the best model parameters by learning on the training dataset for a fixed set of hyperparameters
Hyperparameter Optimization	• training dataset • validation dataset	• best hyperparameters	We pick the best hyperparameters by learning on the training data and evaluating error on the validation error
Testing	• test dataset • hypothesis (i.e. fixed model parameters)	• test error	We evaluate a hypothesis corresponding to a decision rule with fixed model parameters on a test dataset to obtain test error

for

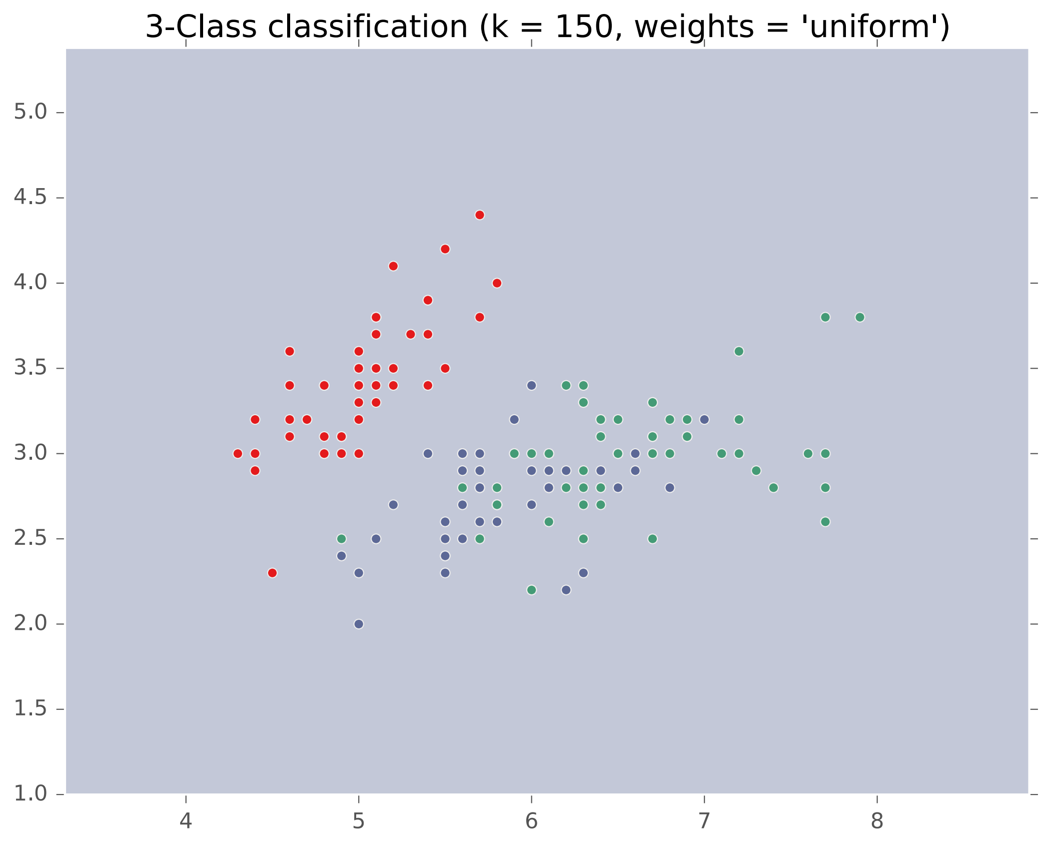
Slide credit: CMU MLD Matt Gormley

Special Cases of k-NN

k=1: Nearest Neighbor

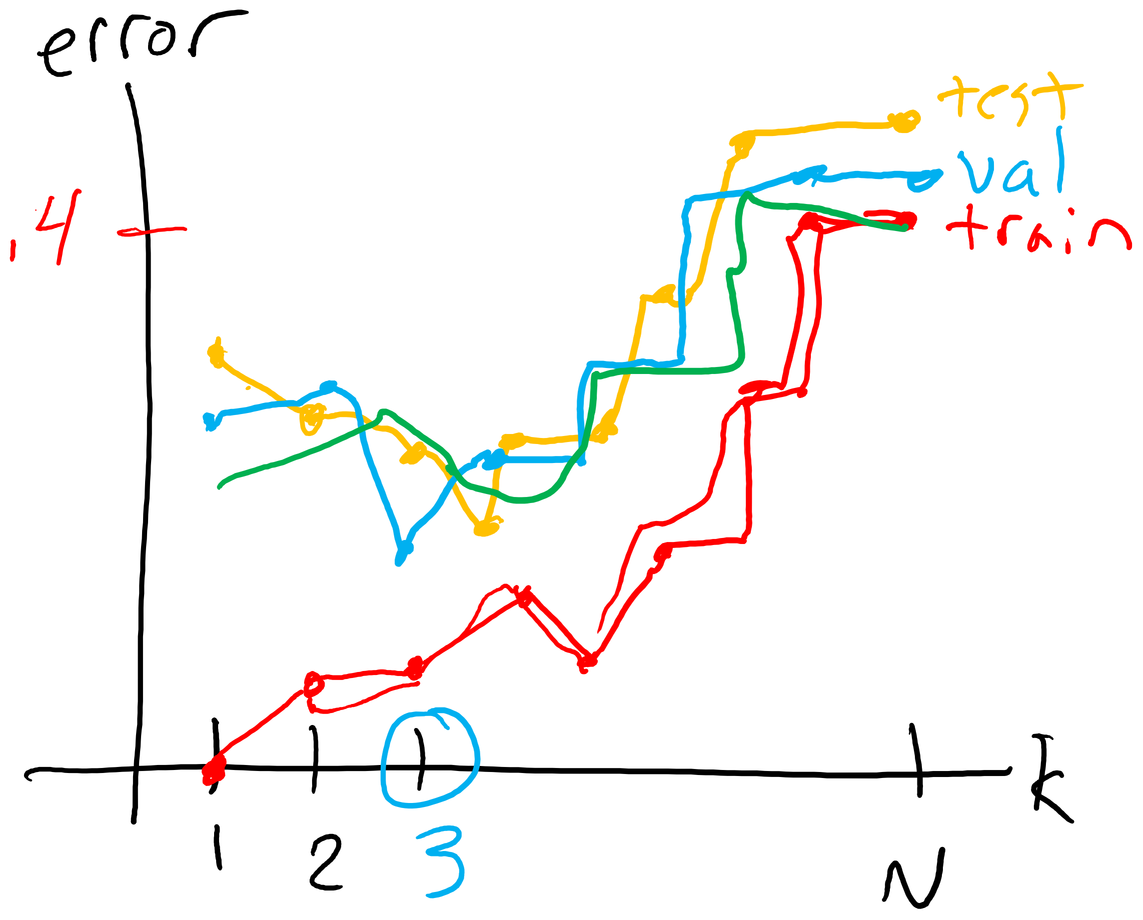


k=N: Majority Vote

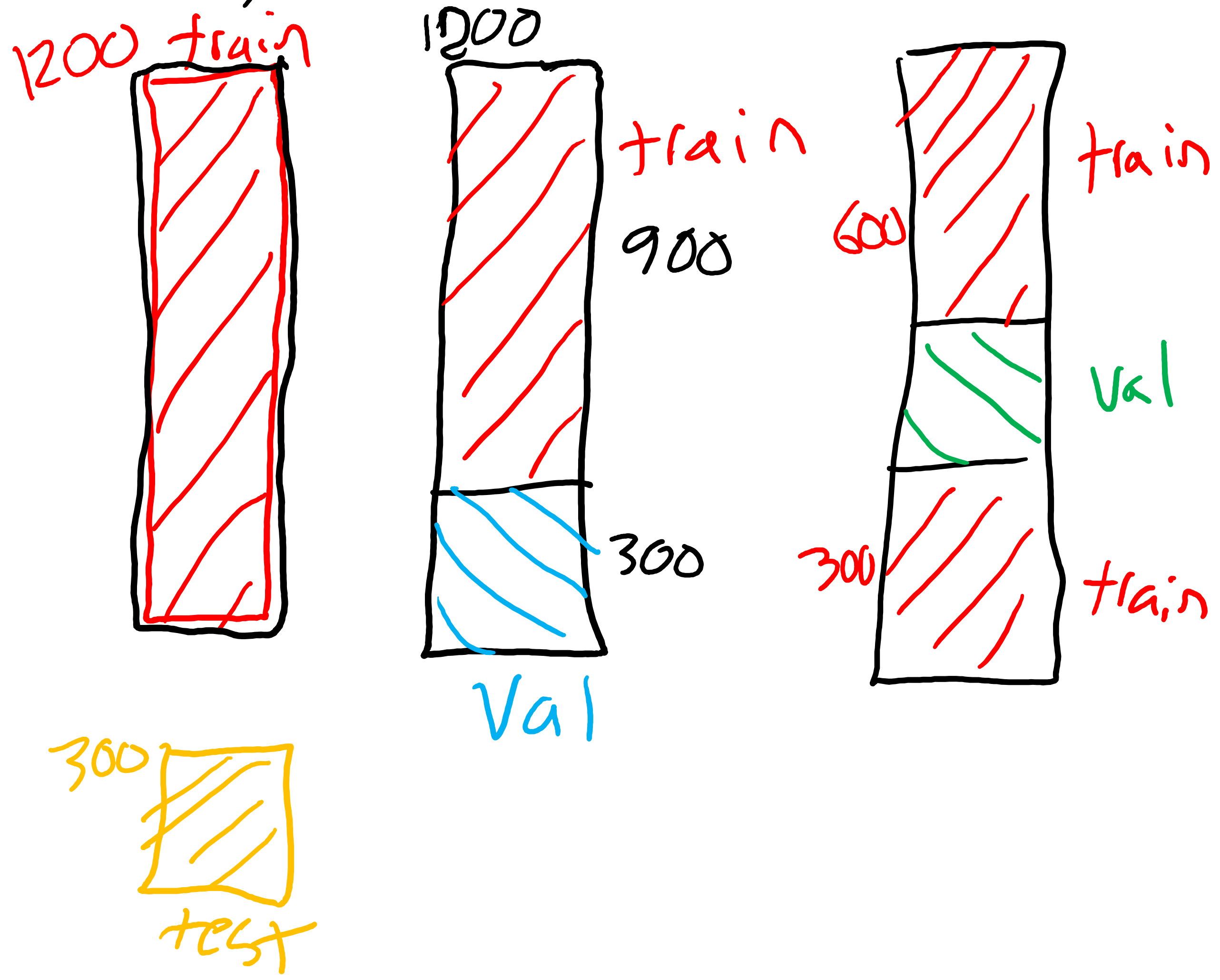


Example of Hyperparameter Optimization

Choosing k for k-NN

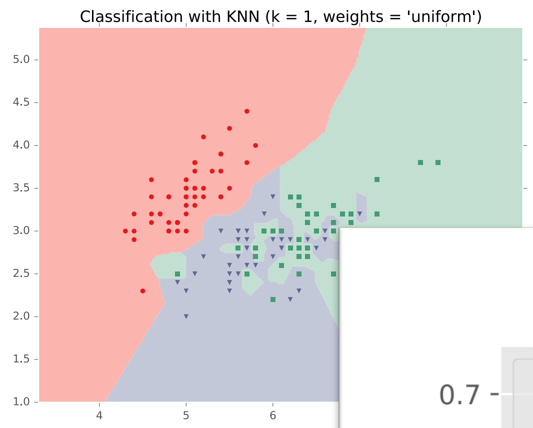


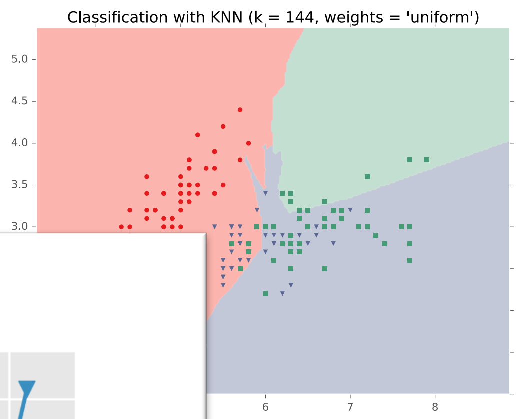
40% y=0   60% y=1

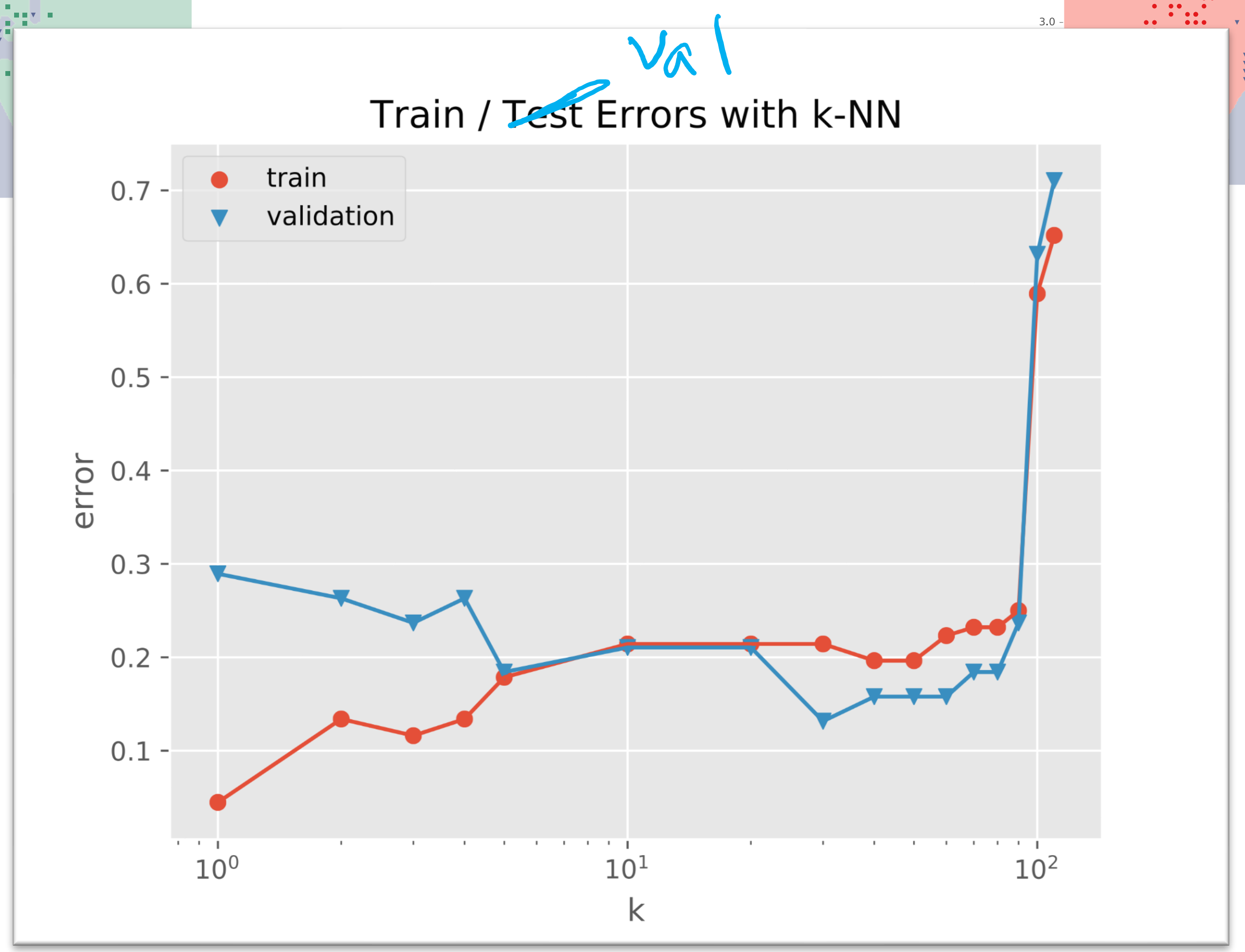


Slide credit: CMU MLD Matt Gormley

k-NN: Choosing k



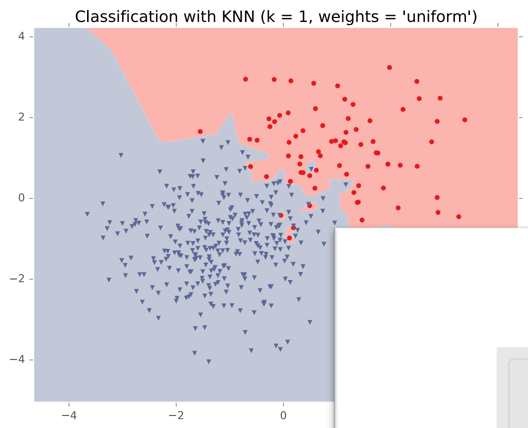


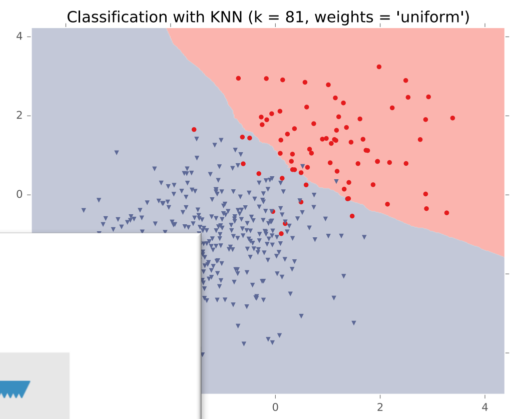


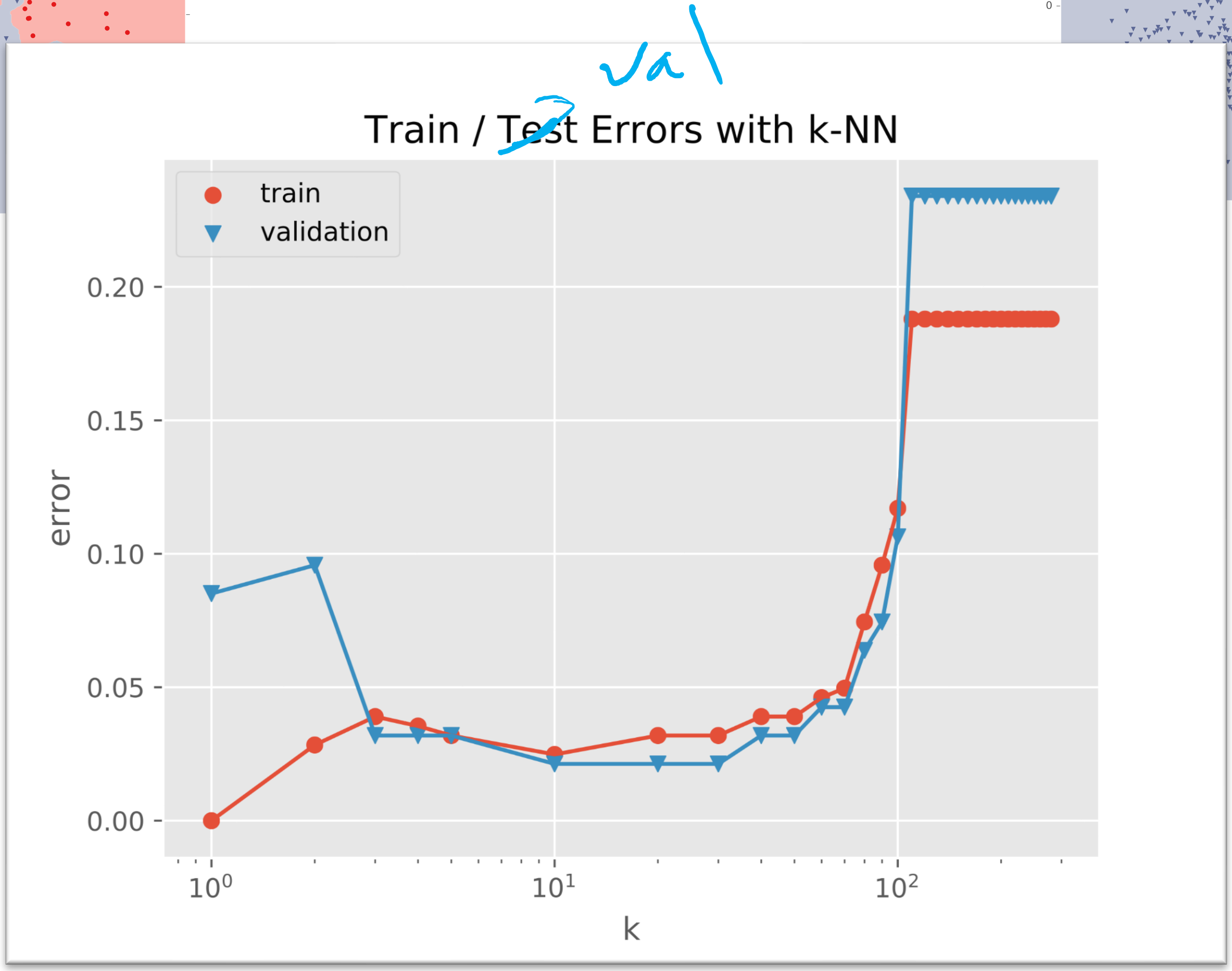
Fisher Iris Data: varying the value of k

Slide credit: CMU MLD Matt Gormley

k-NN: Choosing k







Gaussian Data: varying the value of k

Validation

Why do we need validation?

- Choose hyperparameters
- Choose technique
- Help make any choices beyond our parameters

But now, we have another choice to make!

- How do we split training and validation?

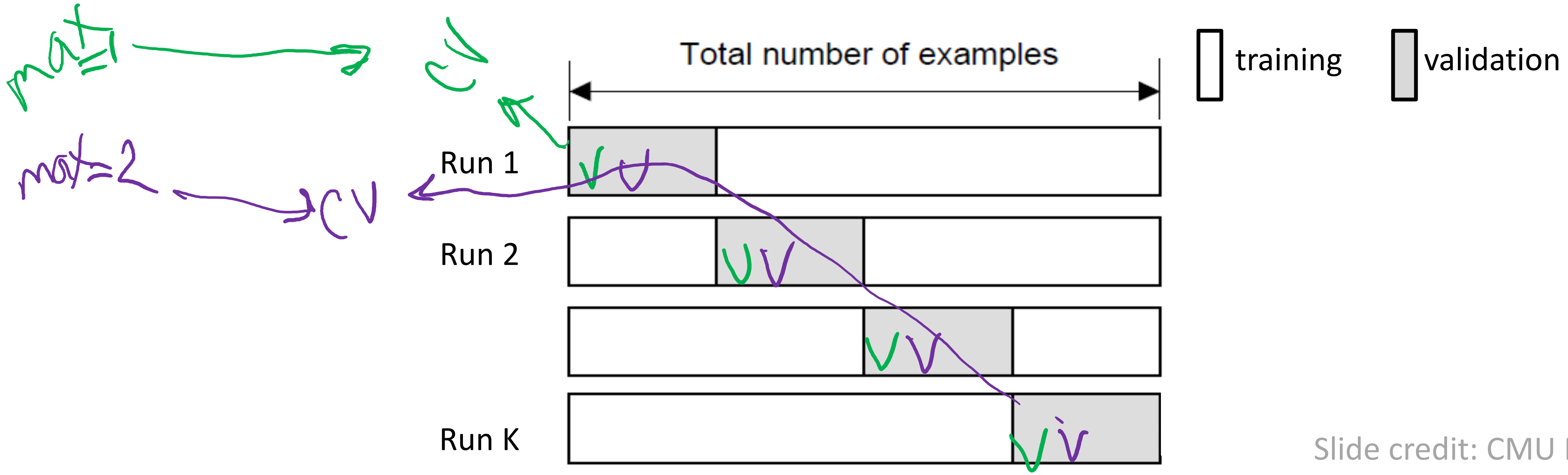
Trade-offs

- More held-out data, better meaning behind validation numbers
- More held-out data, less data to train on!

Cross-validation

K-fold cross-validation

Create K-fold partition of the dataset.
Do K runs: train using K-1 partitions and calculate validation error on remaining partition (rotating validation partition on each run).
Report average validation error

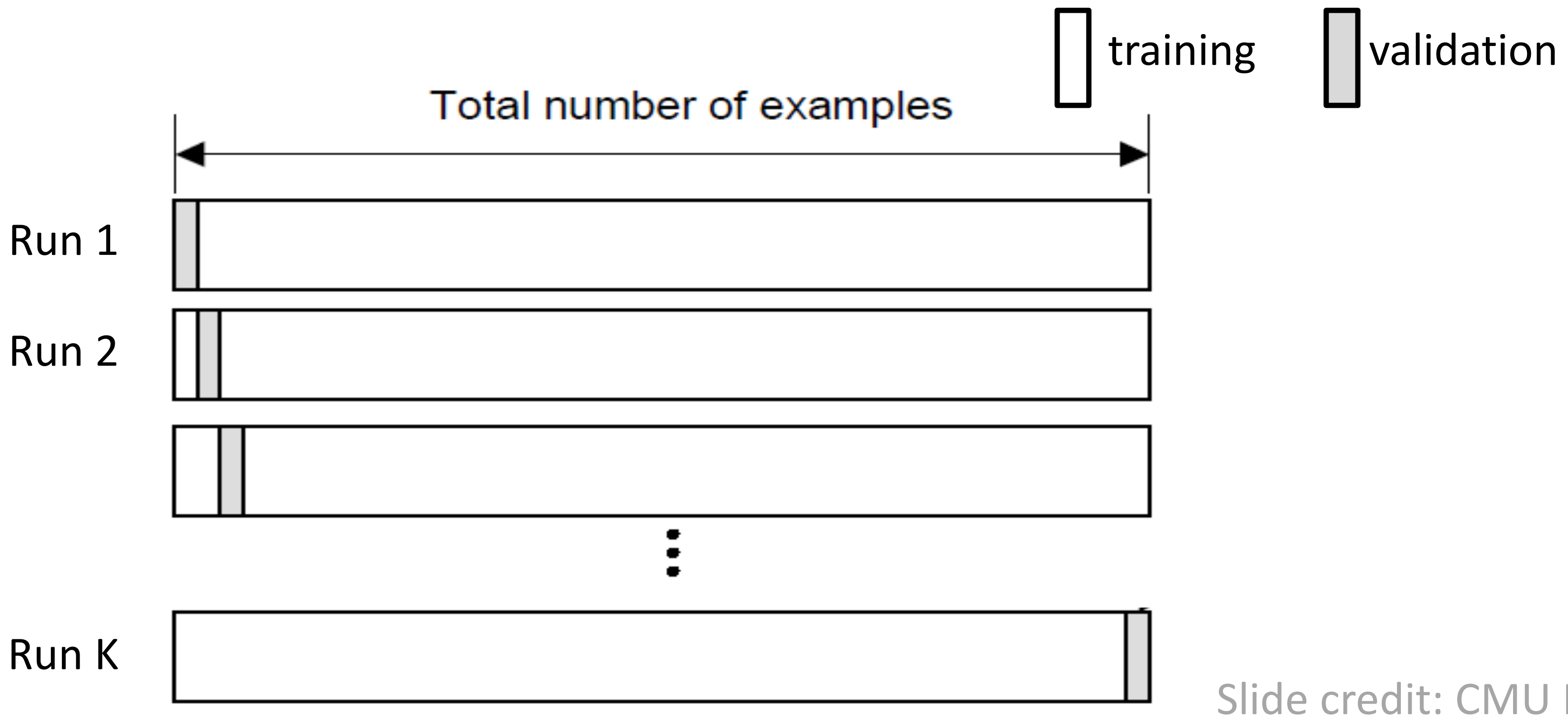


Slide credit: CMU MLD Aarti Singh

Cross-validation

Leave-one-out (LOO) cross-validation

Special case of K-fold with K=N partitions
Equivalently, train on N-1 samples and validate on only one sample per run for N runs



Slide credit: CMU MLD Aarti Singh

Cross-validation

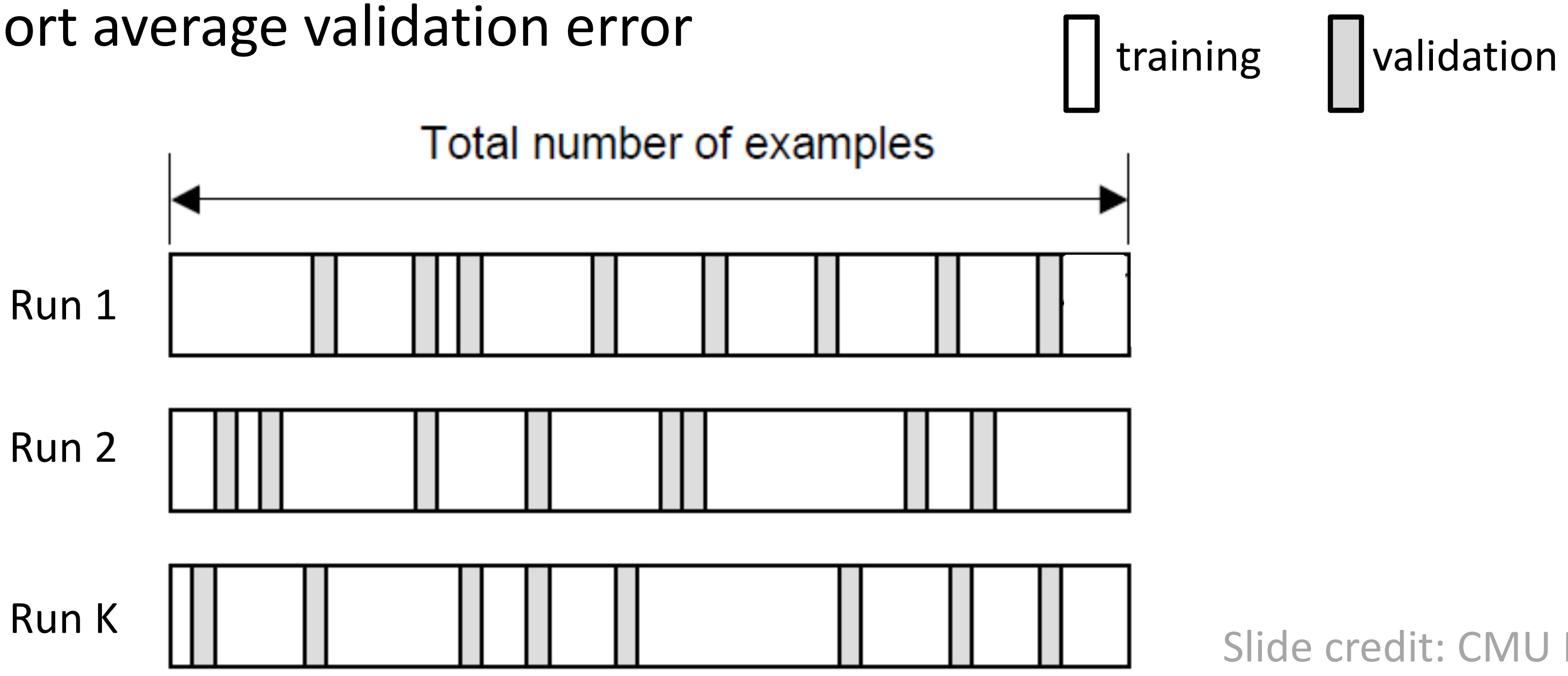
Random subsampling

Randomly subsample a fixed fraction αN ($0 < \alpha < 1$) of the dataset for validation.

Compute validation error with remaining data as training data.

Repeat K times

Report average validation error



Slide credit: CMU MLD Aarti Singh

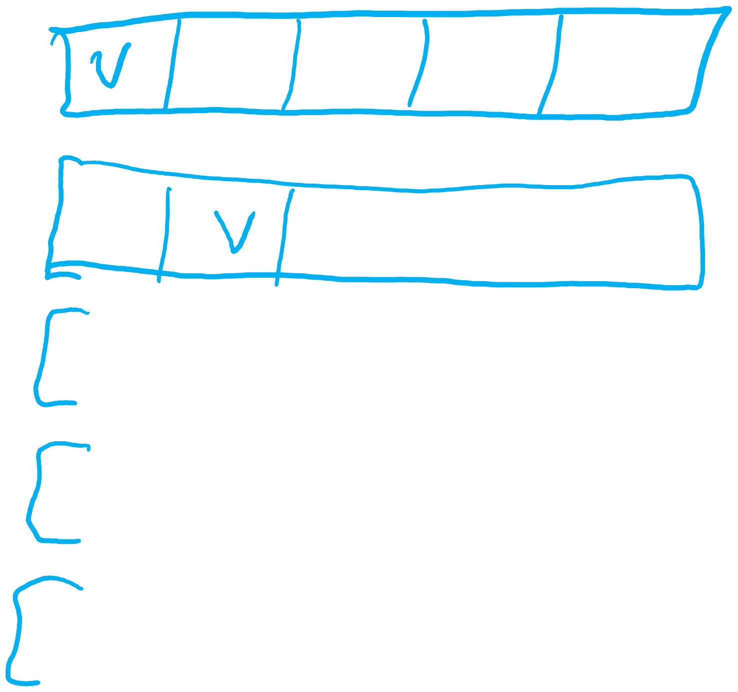
Poll 7

Say you are choosing amongst 7 discrete values of a decision tree *mutual information threshold*, and you want to do K=5-fold cross-validation.

How many times do I have to train my model?

A. 1
B. 5
C. 7
D. 12
E. 35
F. 5^7

```
for i in 1..7
    cv_err[i] = 0
    for j in 1..5
        train
        v_err = ____
        cv_err[i] += 
```



Poll 7

Say you are choosing amongst 7 discrete values of a decision tree *mutual information threshold*, and you want to do K=5-fold cross-validation.

How many times do I have to train my model?

A. 1
B. 5
C. 7
D. 12
E. 35
F. 5^7

Model Selection

WARNING (again):

- This section is only scratching the surface!
- Lots of methods for hyperparameter optimization: (to talk about later)
  - Grid search
  - Random search
  - Bayesian optimization
  - Graduate-student descent
  - …

Main Takeaway:

- Model selection / hyperparameter optimization is just another form of learning