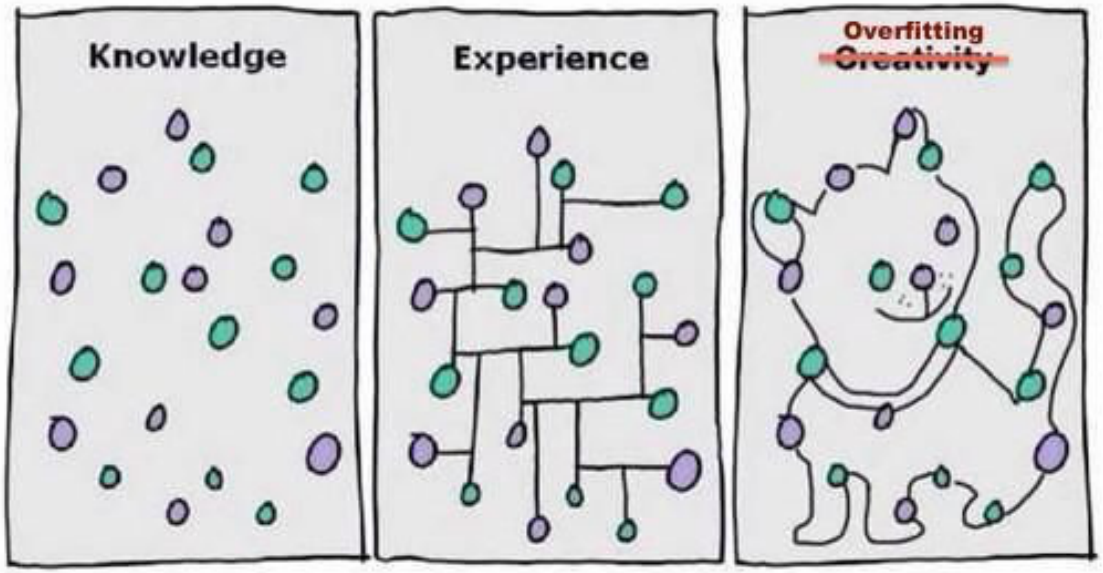
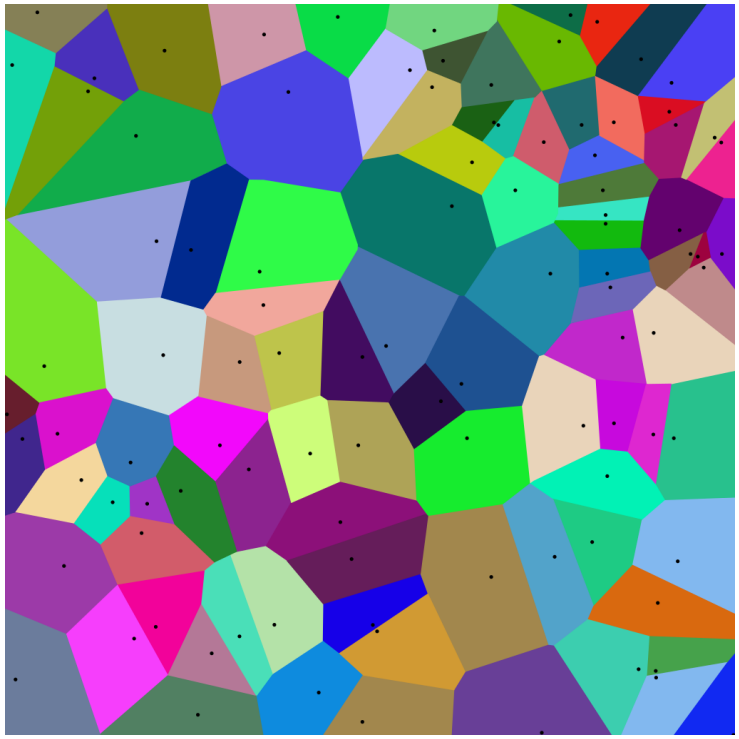

Knowledge
Experience
Overfitting
Creativity

Nearest Neighbor Classifiers



Types of Learning Algorithms

- **Eager Learners:** model input attributes to a class label as soon as training data is available (e.g. Decision Trees)

- **Lazy Learners:** delay the process of generalizing the training data until it is needed to classify test examples (e.g. Nearest Neighbor)

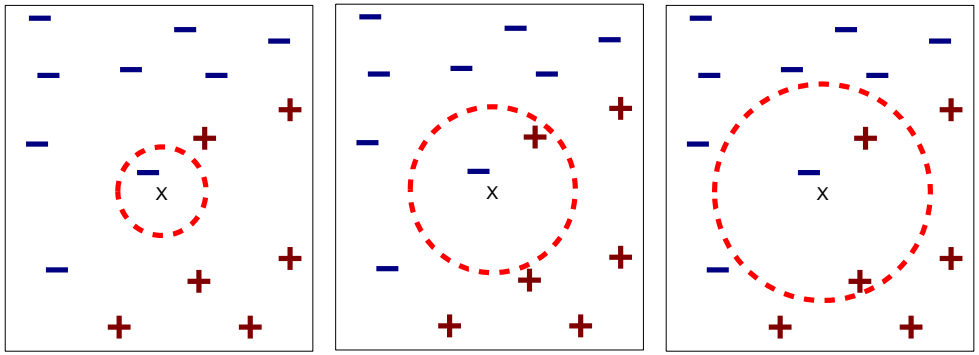
Nearest Neighbor

- Find all the training examples that are relatively similar to the new record. These examples are the **nearest neighbors** and are used to classify the new record.

"If it walks like a duck, quacks like a duck, and looks like a duck, then it's probably a duck."

k-Nearest Neighbors (KNN)

- Represent each training example as a data point in a d-dimensional space (where d is the number of attributes)
- Given a test record, compute it's distance to all of the training points
- The test record is classified based on the majority class labels of it's k nearest neighbors



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

Computing Distance

- Euclidian distance:

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

Where i is the number of dimensions

- For nominal attributes, Hamming distance (or Overlap):

  distance = 0, if they are same
  distance = 1, if they are different

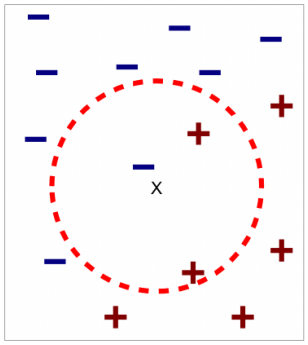
IMPORTANT: Data must be scaled so that all features are on the same scale!

Voting Schemes

- In majority voting, every neighbor has same impact on the classification

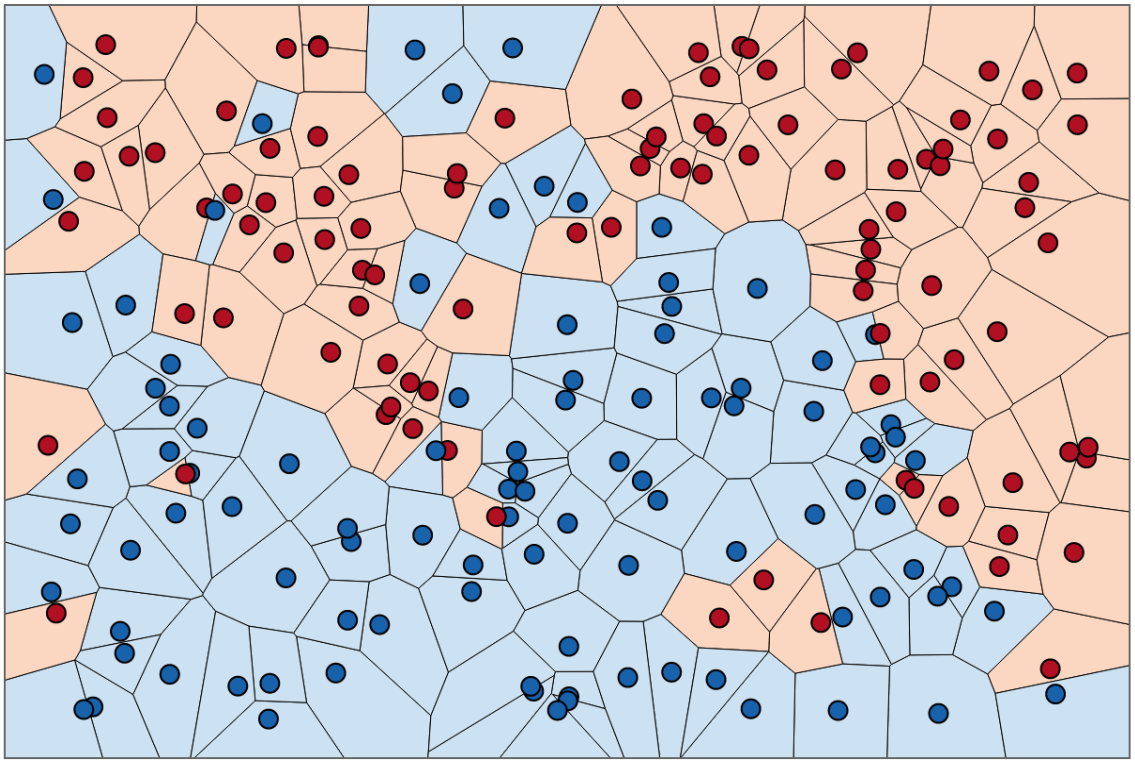
- Or, can weigh each vote according to distance

  weight factor, $w = 1/d^2$



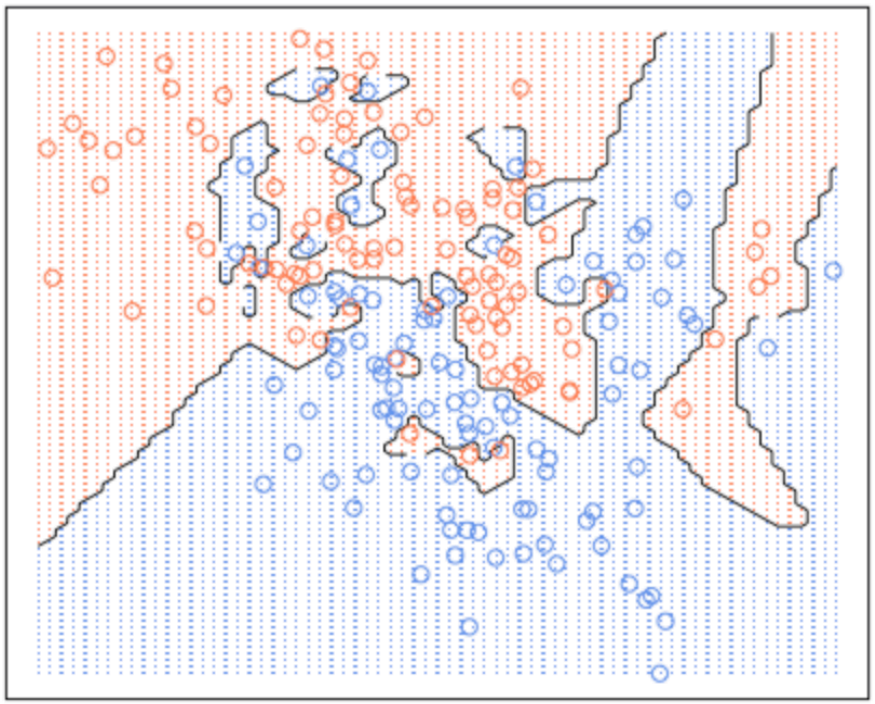
(c) 3-nearest neighbor

Decision Boundaries

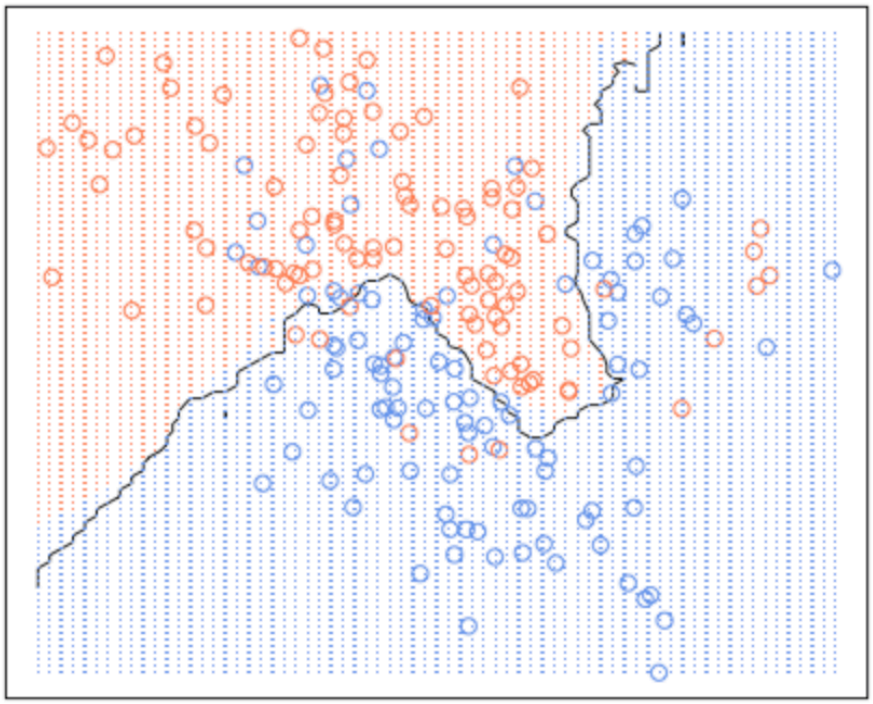


k = 1

Decision Boundaries



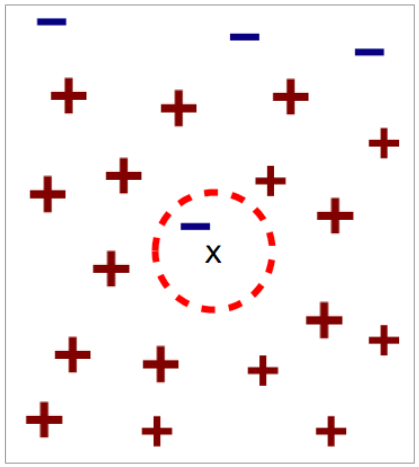
k = 1

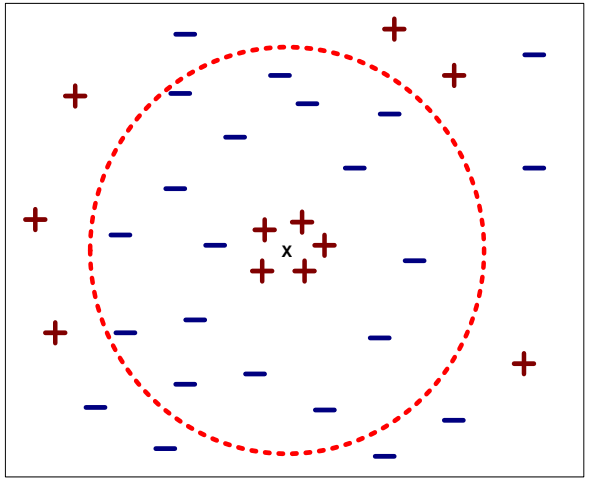


k = 20

Choosing the right k

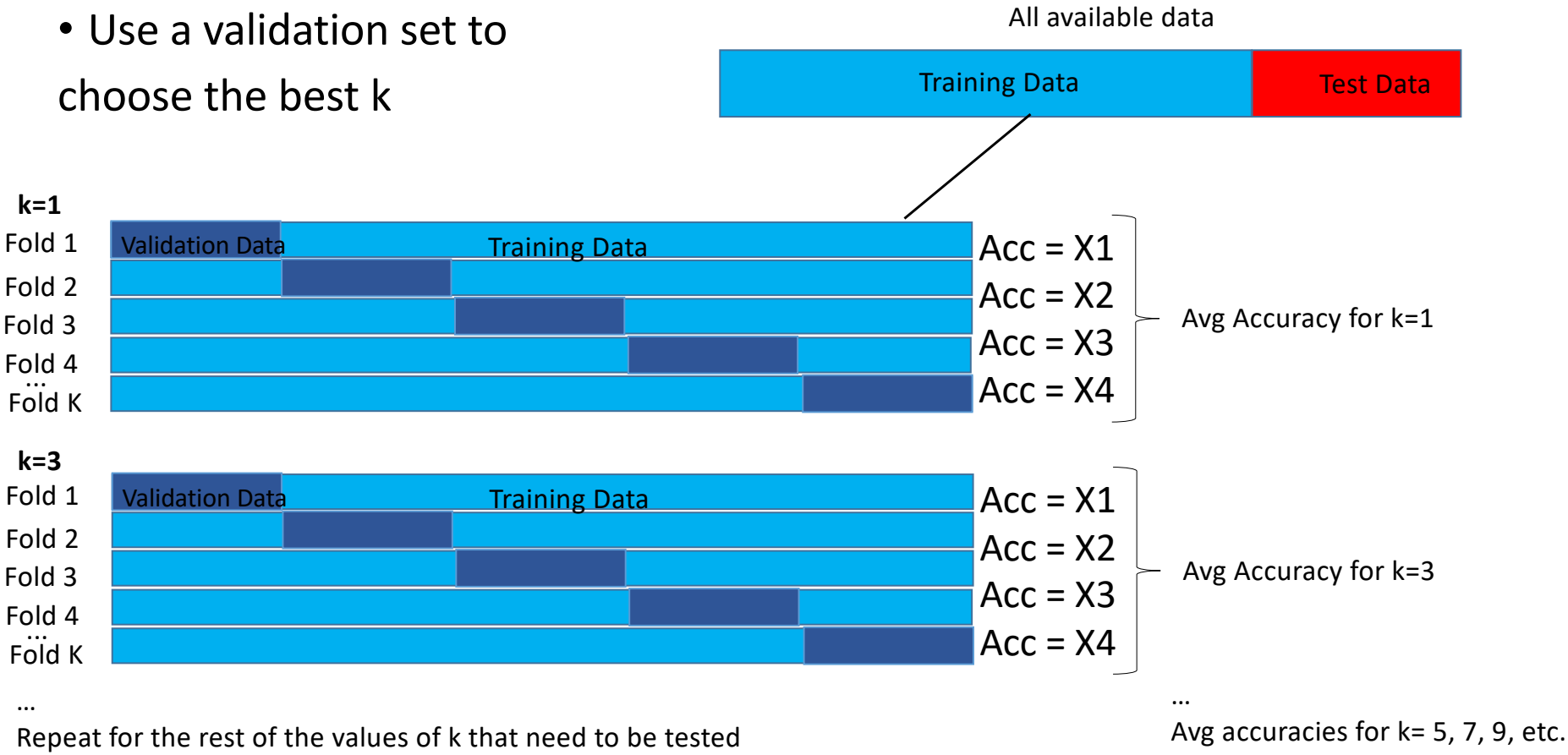
- If k is too small, sensitive to noise
- If k is too large, may include points of other classes



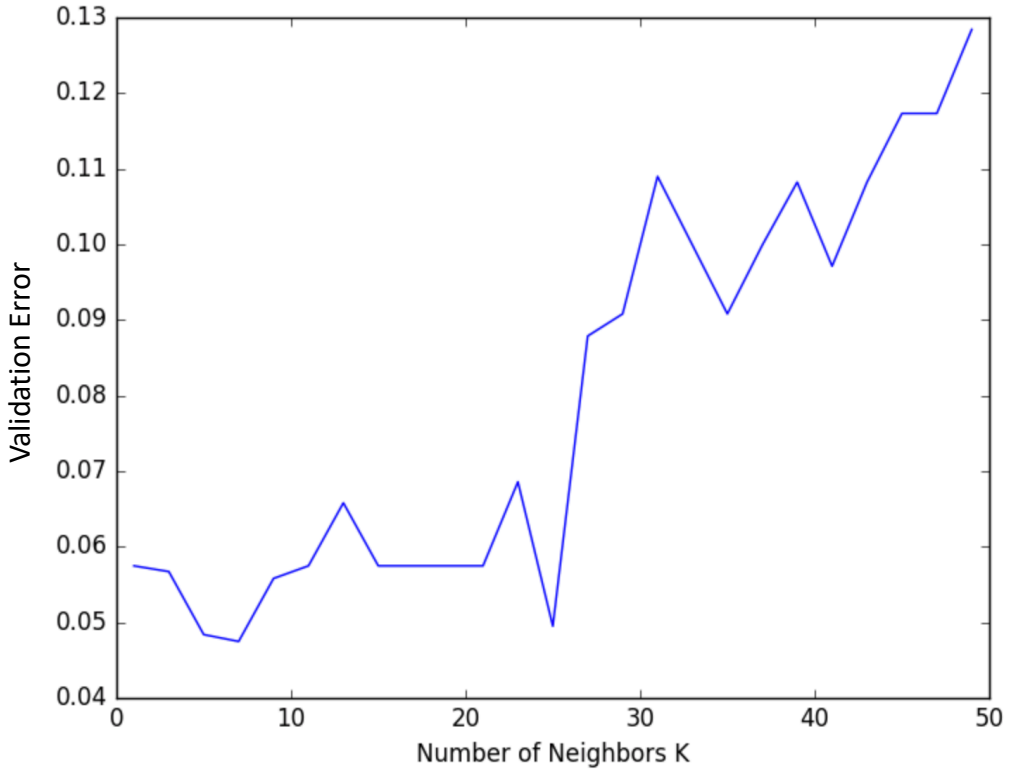


Choosing k: Nested CV

- Use a validation set to choose the best k

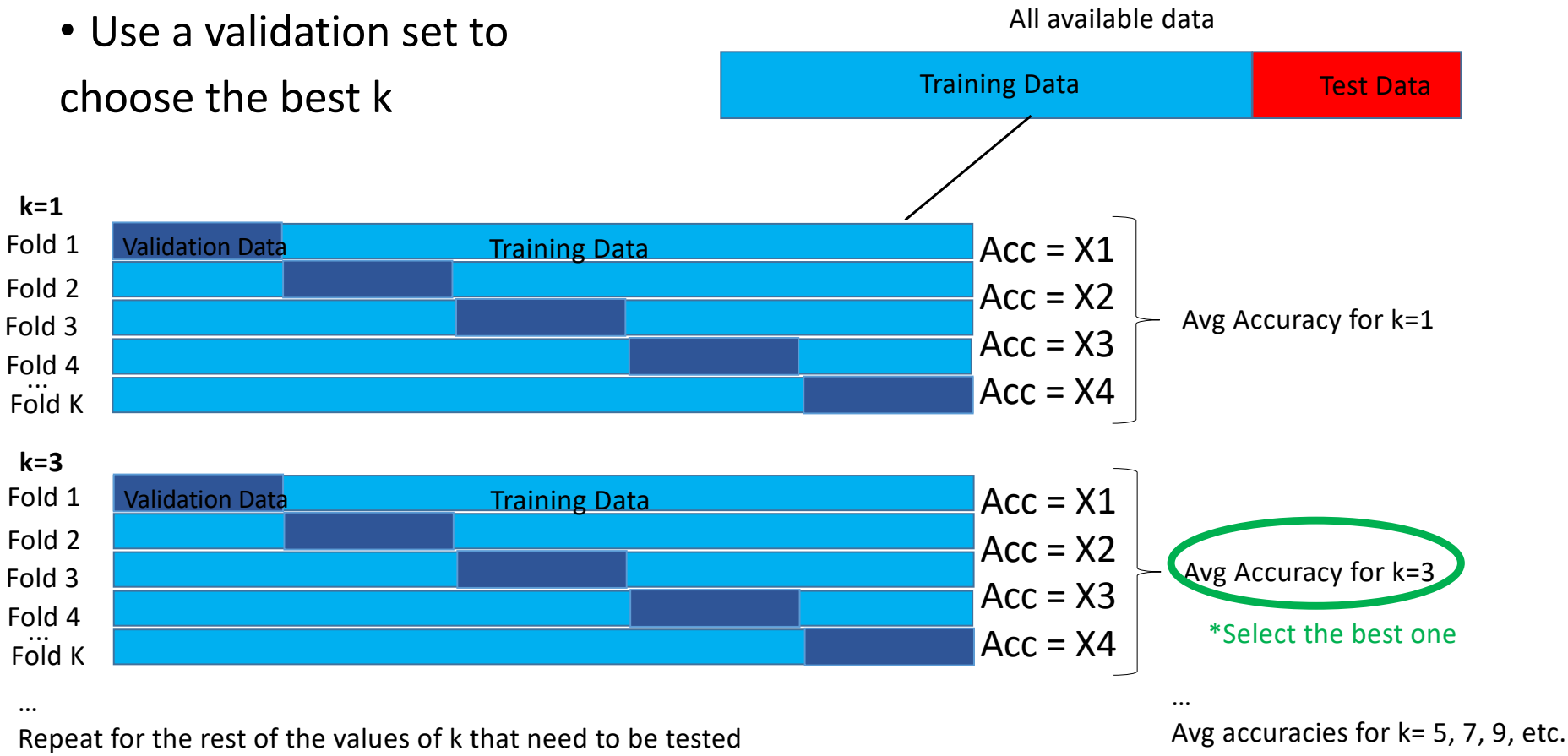


Use Cross-Validation to Find Best k



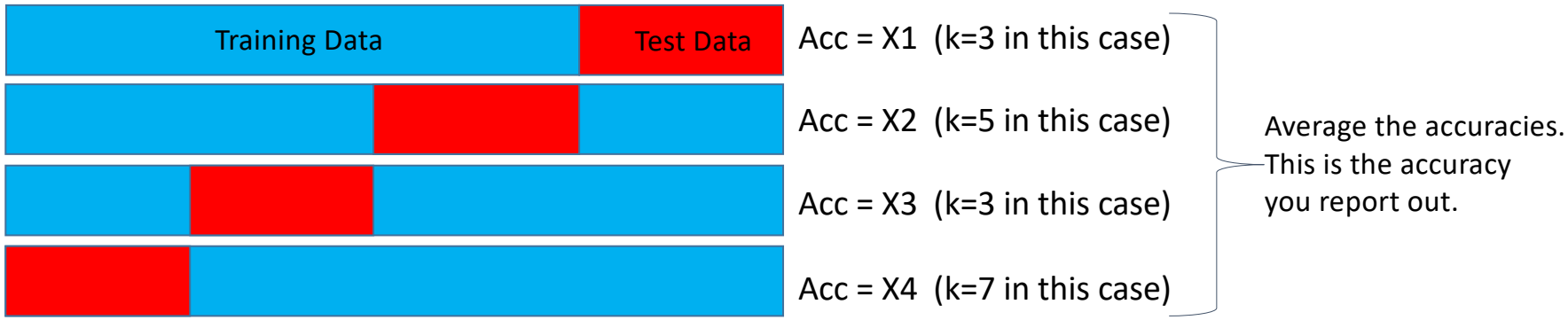
Choosing k: Nested CV

- Use a validation set to choose the best k



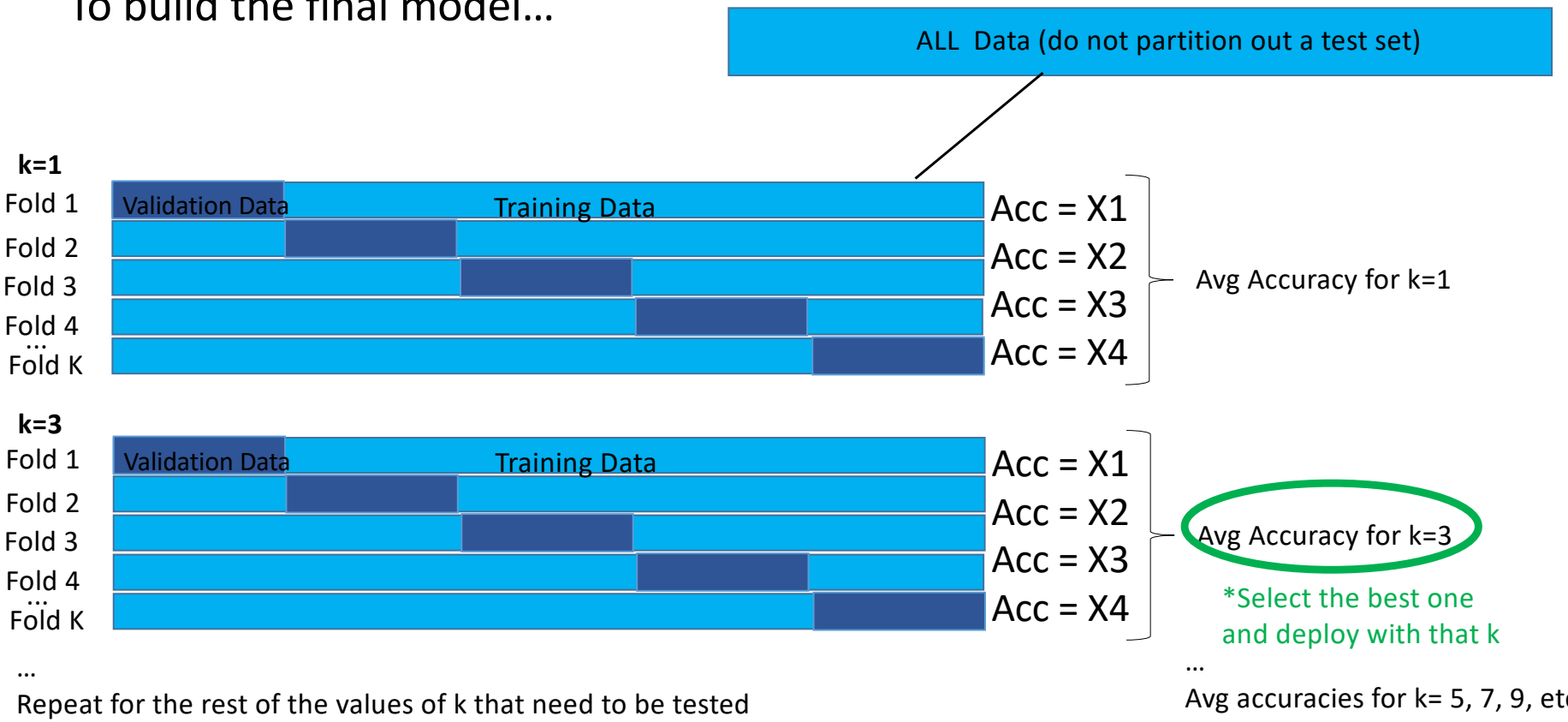
Nested CV

- Classify the test set using the best k found in the inner CV loop. This gets you an accuracy for one fold of the outer CV loop.
- Repeat for all folds of the outer CV loop.
- Average accuracies from the outer CV loop to get overall estimate of generalization accuracy/error.



Choosing k

To build the final model…

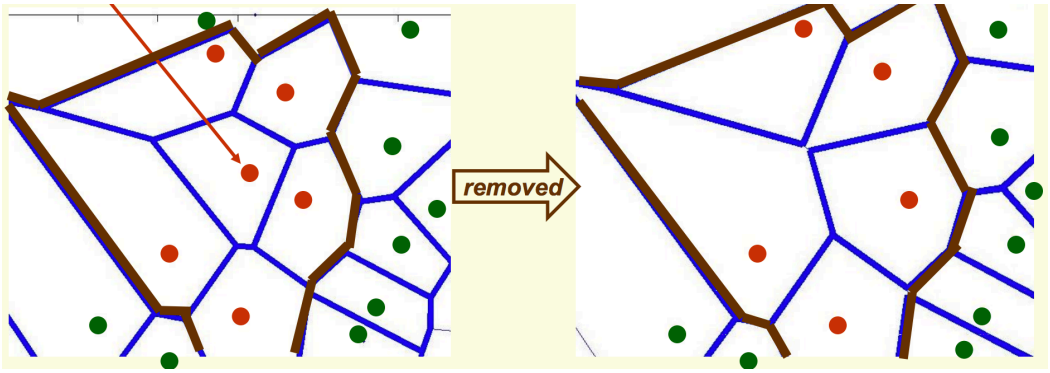


Instance Reduction Algorithms

- Select a representative subset of the training data to use (eliminate the rest)
- Can help with classification times
- Can help with memory/storage issues of storing all training records
- Can help eliminate noise points and reduce overfitting

Reduced Nearest Neighbor (RNN)

- Create S, a subset of T (the training set). Start with S = T.
- If removing record r from S does not cause any other record in T to be misclassified using S, permanently remove record r
- Typically use k=1, or k=3



Characteristics of KNN classifiers

- KNN classification uses a general technique called **instance-based learning**, where specific training instances are used to make predictions, rather than maintaining an abstraction (model)
  - No need to re-train a model as data changes/updates
- Simple to understand, easy to implement
- Classifying a test record can be computationally expensive
- Data needs to be scaled to avoid one attribute dominating the decision. Ex: height can vary from 5ft – 7ft, but weight can vary from 100lb – 400lb
- Can suffer if there is a class imbalance
  - Weighting votes can help
- **Curse of Dimensionality:** KNN breaks down in high dimensions
- Feature Selection is critical: irrelevant features can dominate a decision

Consider the one-dimensional data set

x	0.5	3.0	4.5	4.6	4.9	5.2	5.3	5.5	7.0	9.5
y	−	−	+	+	+	−	−	+	−	−

(1) Classify the data point x = 5.0 according to its 1-, 3-, 5-, and 9-nearest neighbors (using majority vote)

(2) Repeat the previous analysis using the distance-weighted voting approach