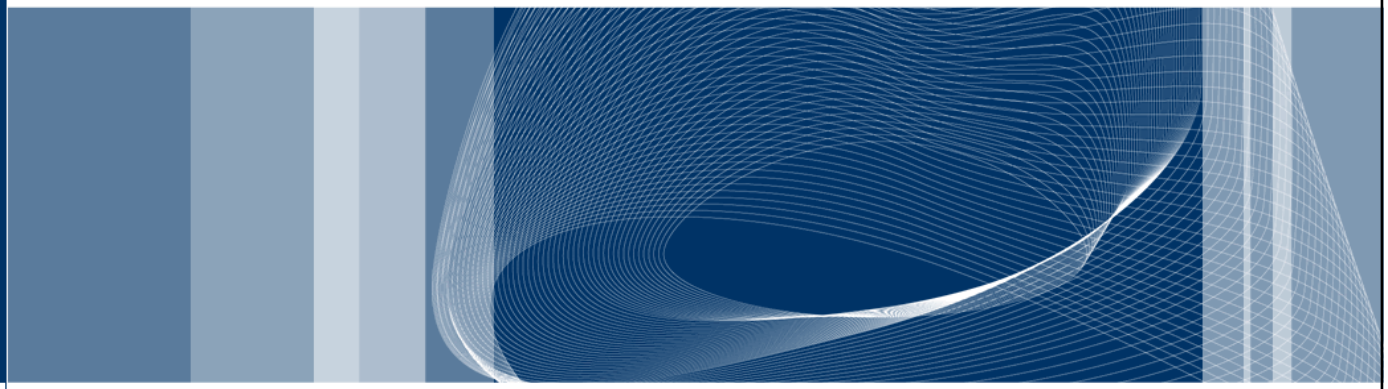
POLITECNICO DI MILANO



k-Nearest Neighbors

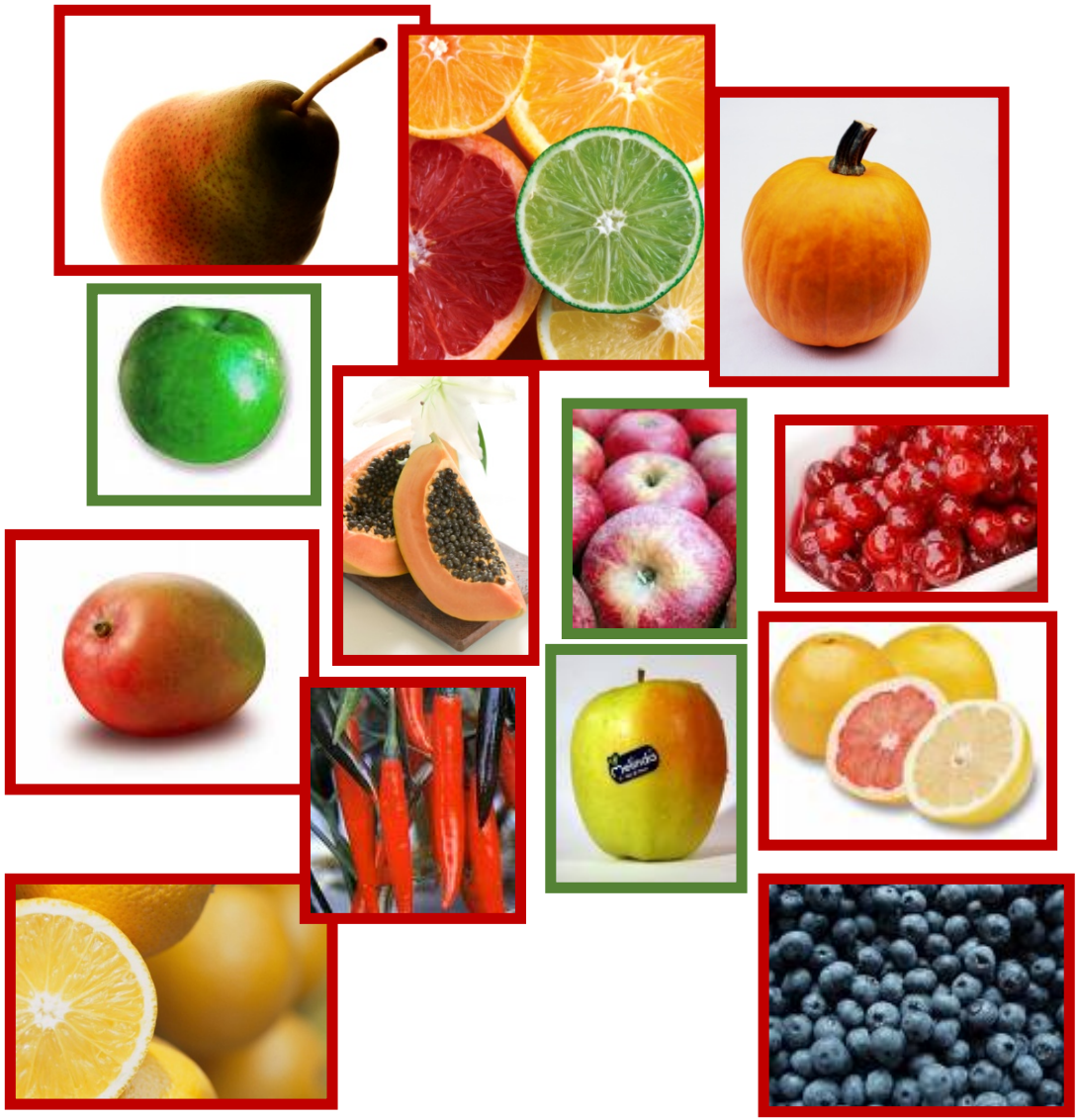
Data Mining and Text Mining

Prof. Pierluca Lanzi

k Nearest-Neighbors Classification

To classify an example x, select the k data points in the training set that are most similar to x

Assign the most frequent class among the k selected



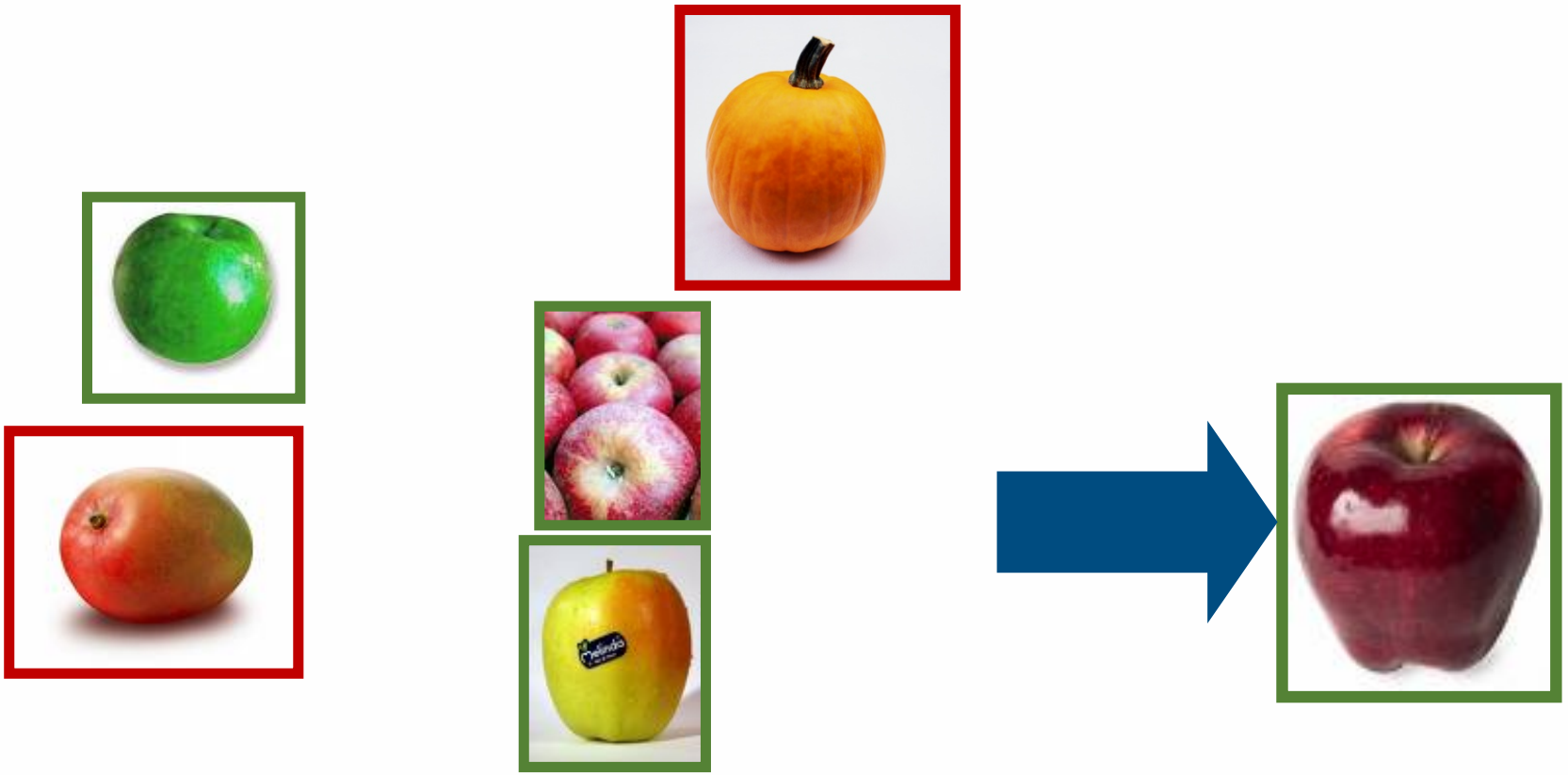
is this an apple?

To decide the label for an unseen example, takes the most similar k examples from the training data

Classify the unknown example
using the most frequent class

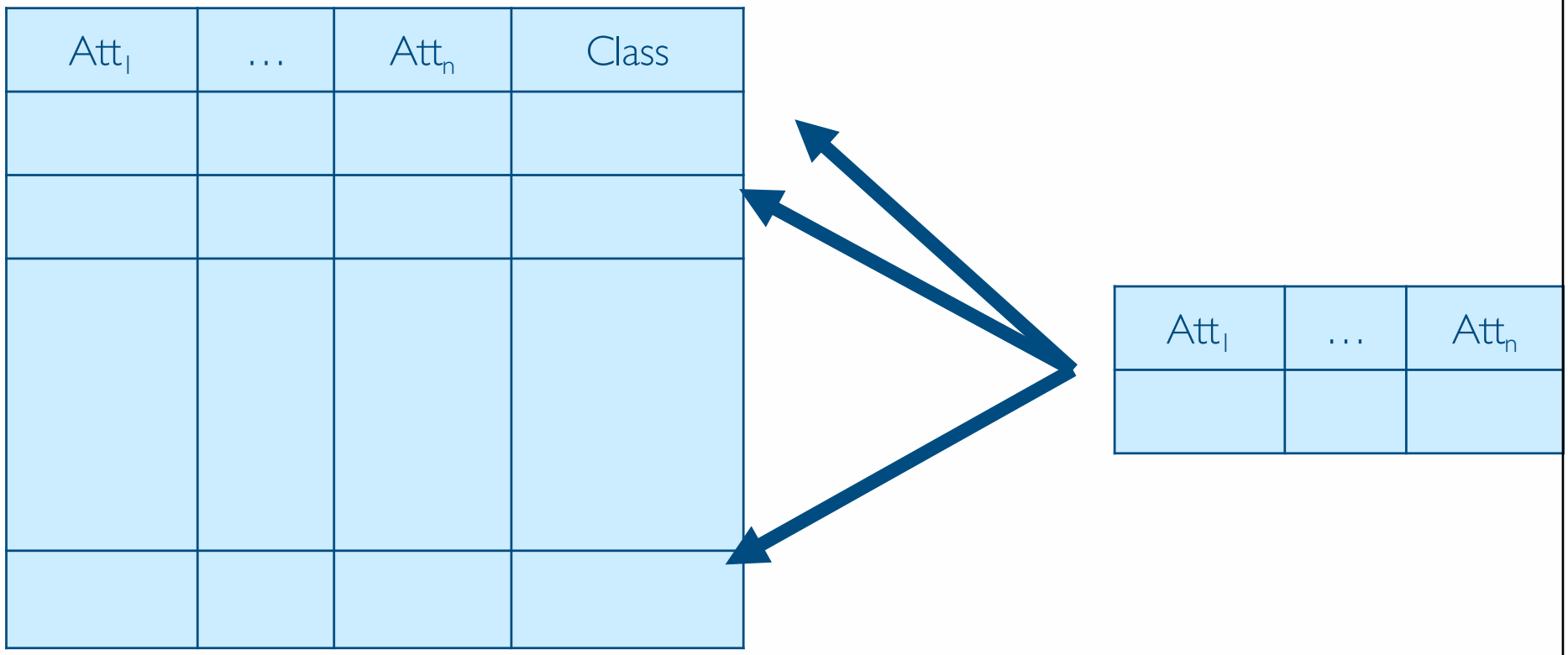
Nearest Neighbor in Action

- If k=5, these 5 fruits are the most similar ones to the unclassified example
- Since, the majority of the fruits are labeled as positive examples, we decide that the unknown fruit is an apple



Instance-Based Learning in General

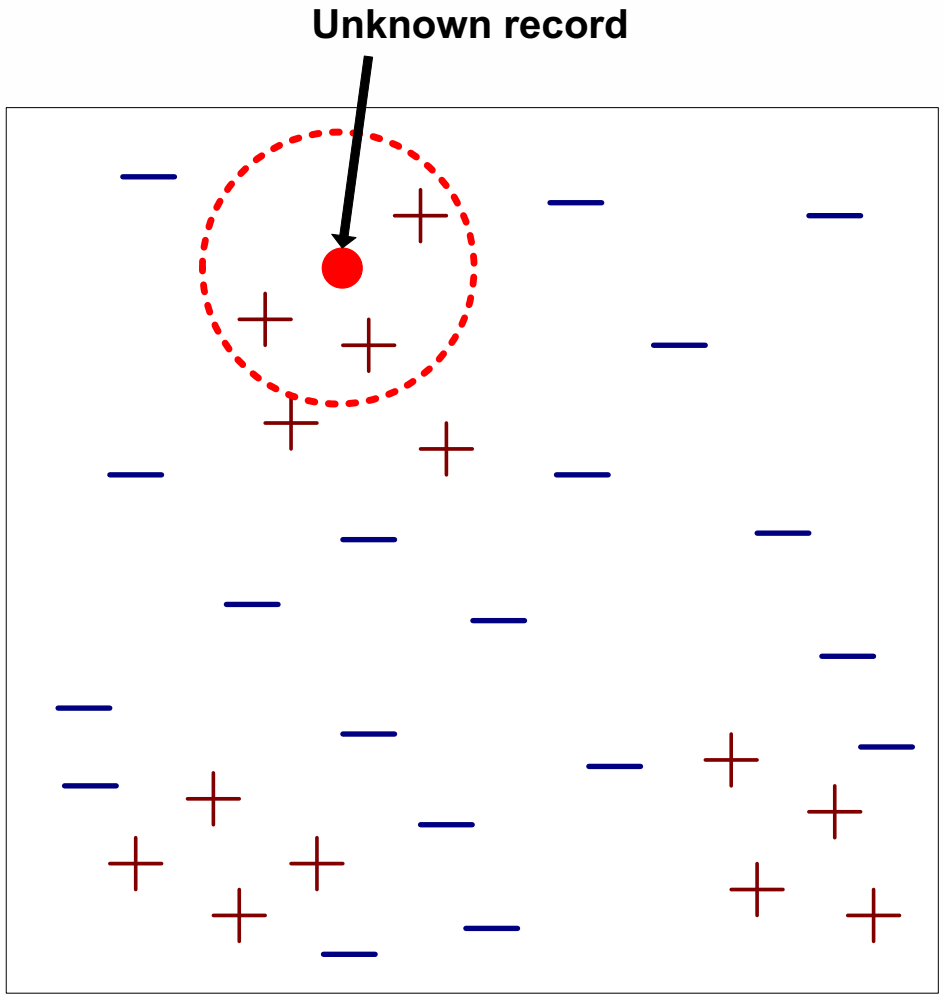
- Store the training records only, no model is computed
- Use the training records to predict an unknown class label



Instance-Based Methods

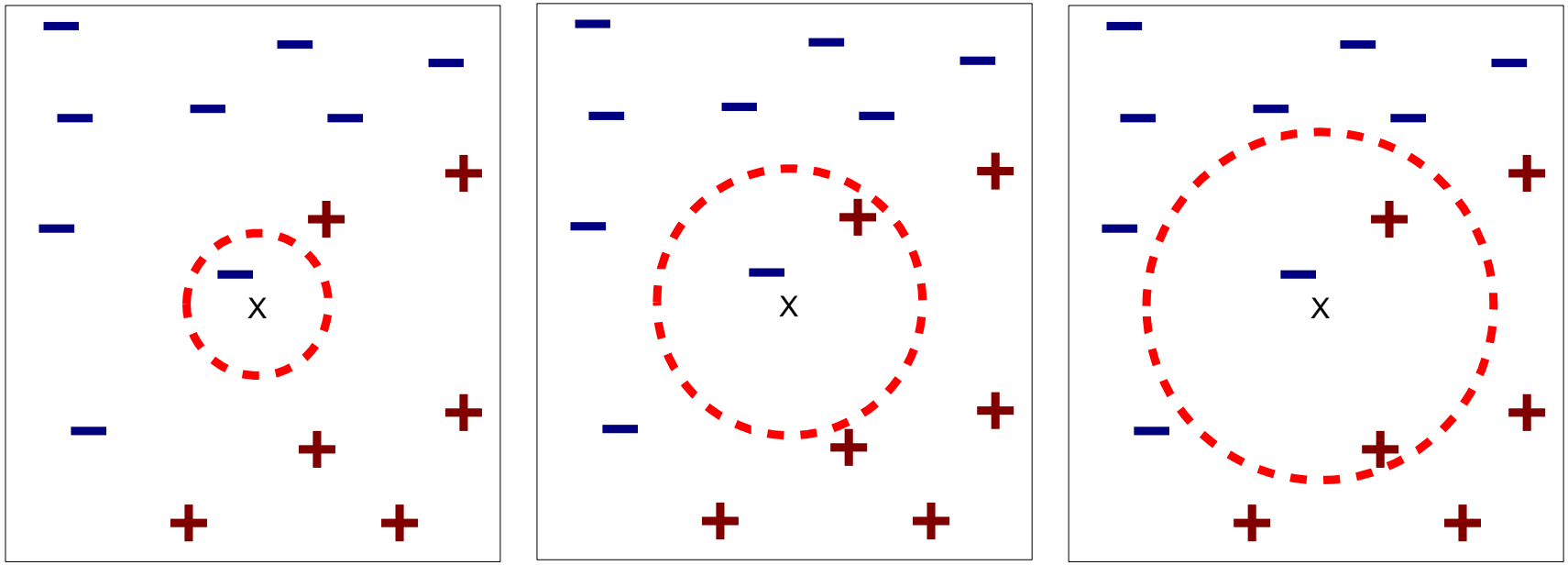
- They are the simplest form of learning
- The training dataset is searched for the instances that are more similar to the unlabeled instance
- The training dataset is the model itself, it is the knowledge
- The similarity function defines what's "learned"
- They implement a "lazy evaluation" (or lazy learning) scheme, nothing happens until a new unlabeled instance must be classified
- Known methods include "Rote Learning", "Case Base Reasoning", "k-Nearest Neighbors"

K-Nearest-Neighbors Classifier



- Three elements
  - The training dataset
  - Similarity function (or distance metric)
  - The value of k, the number of nearest neighbors to retrieve
- Classification
  - Compute distance to other training records
  - Identify the k nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

How Many Neighbors?



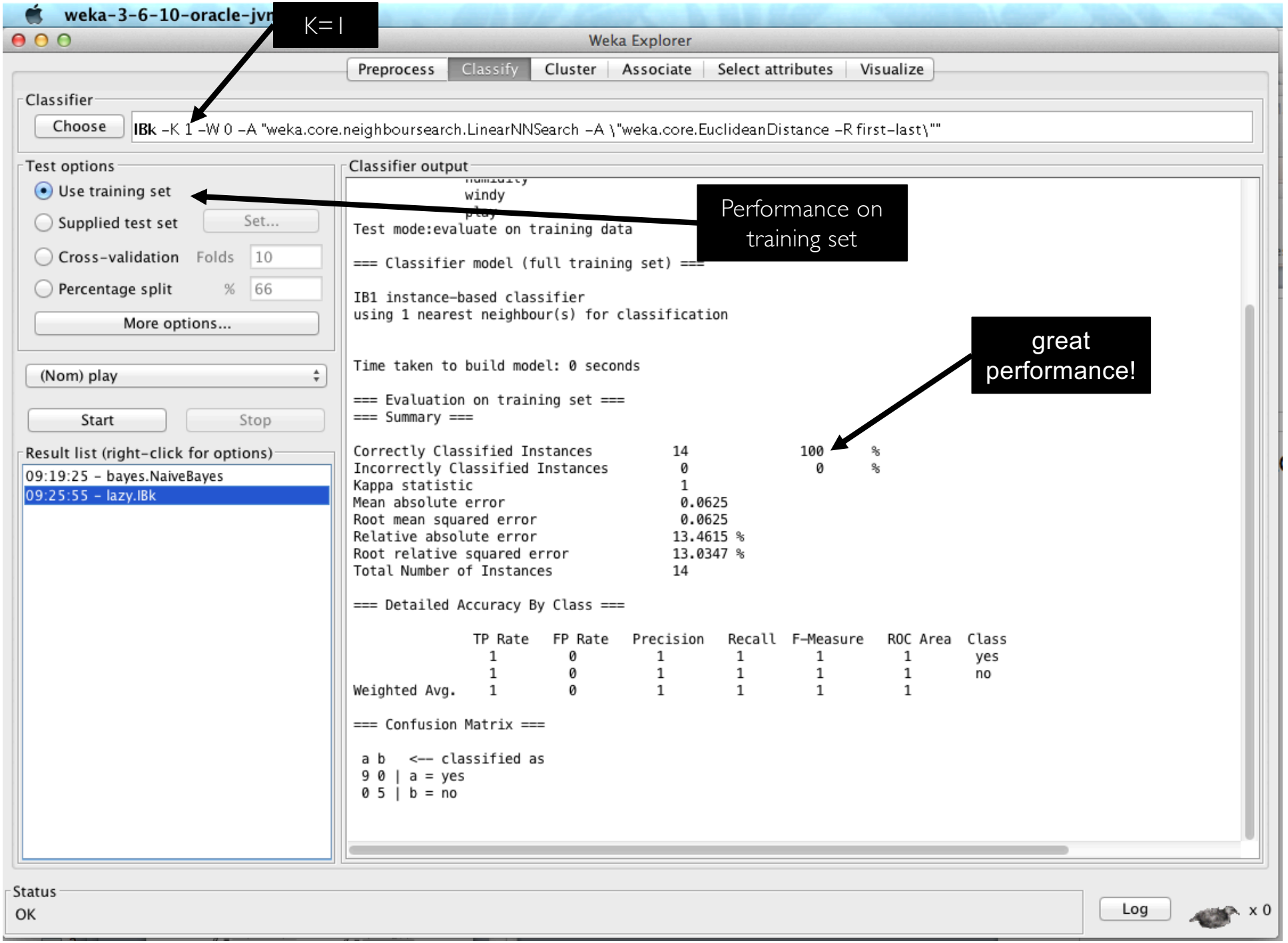
(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

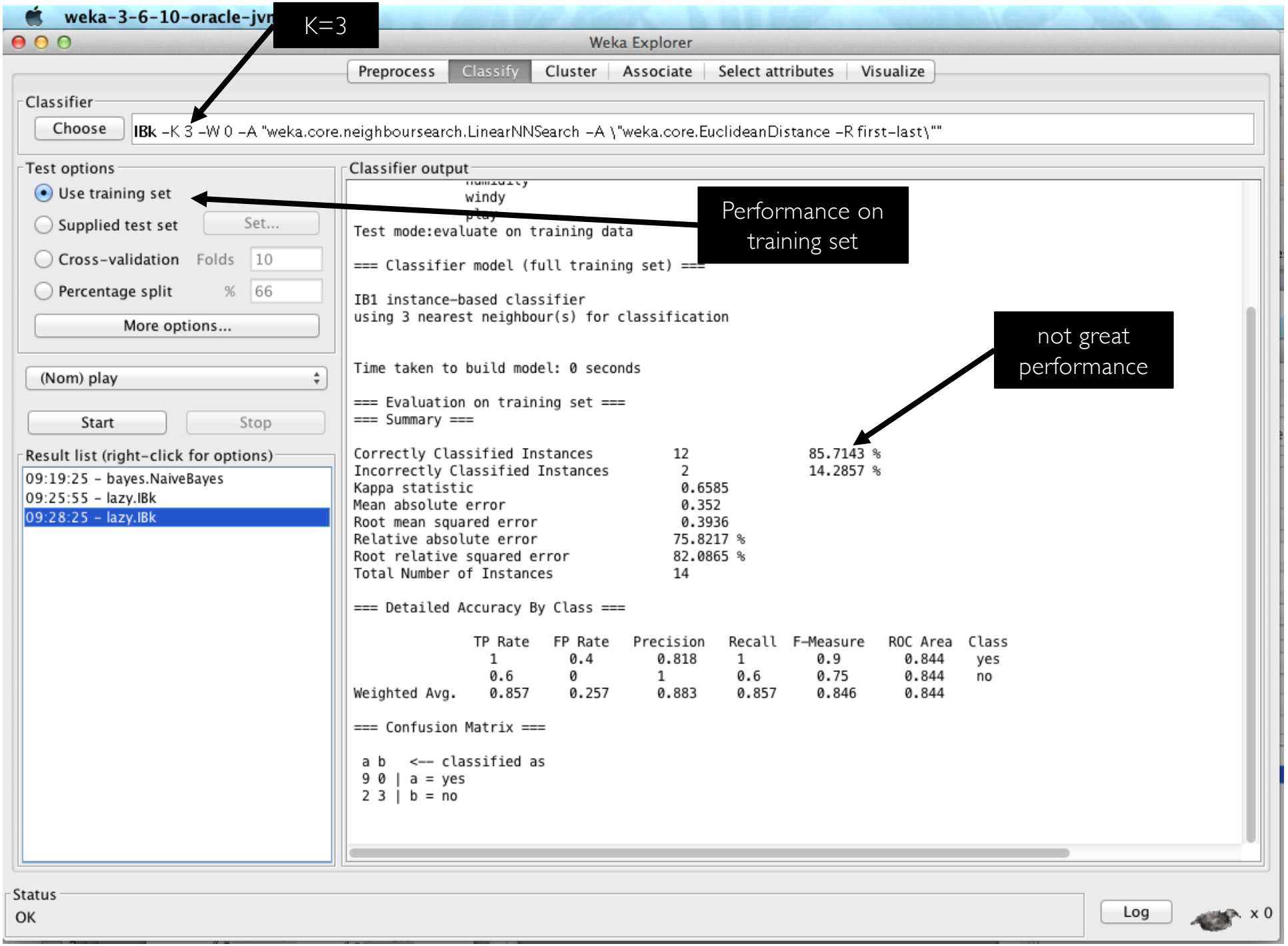
K-nearest neighbors of a record x are data points that have the k smallest distance to x

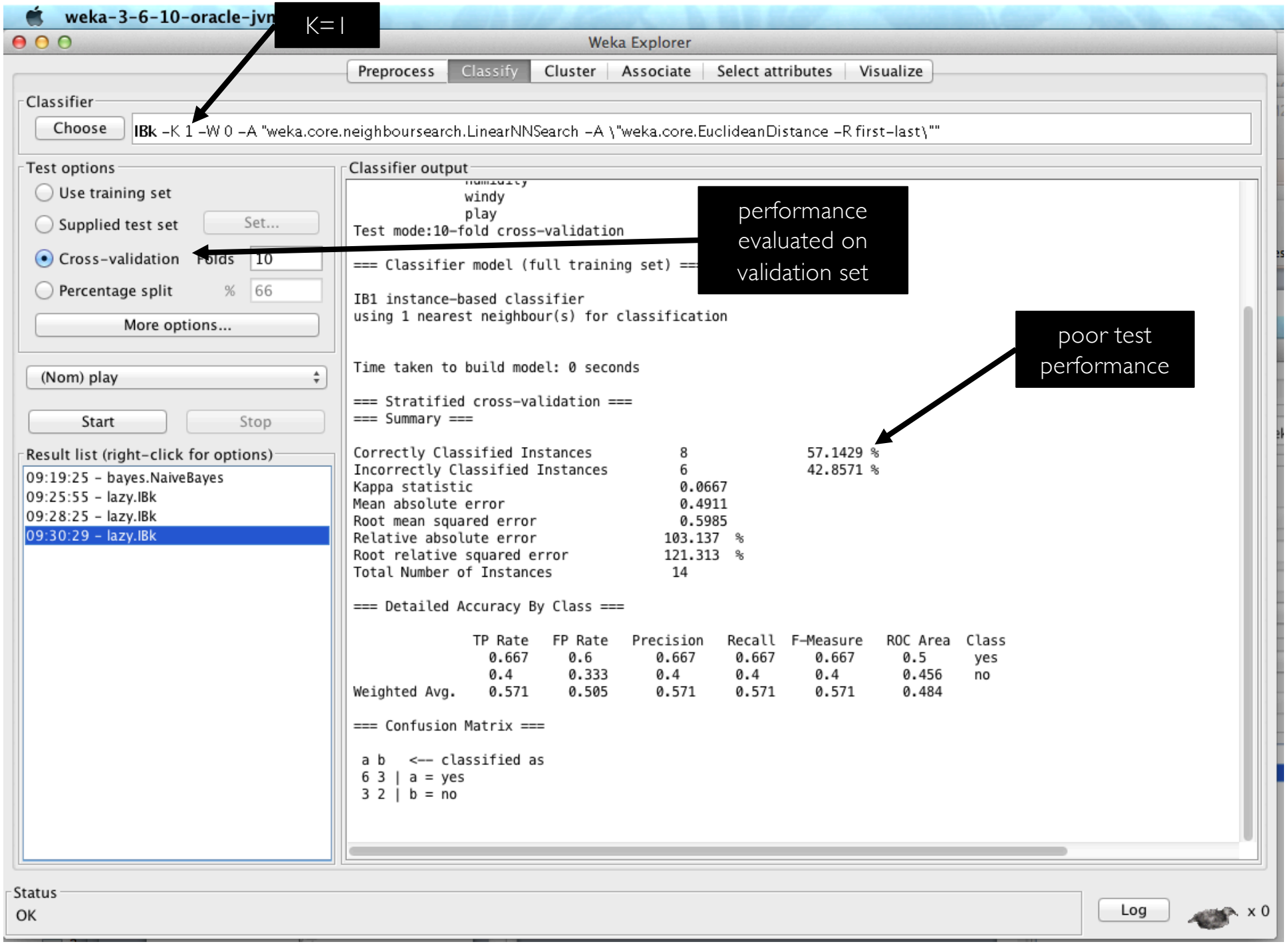
How many neighbors?

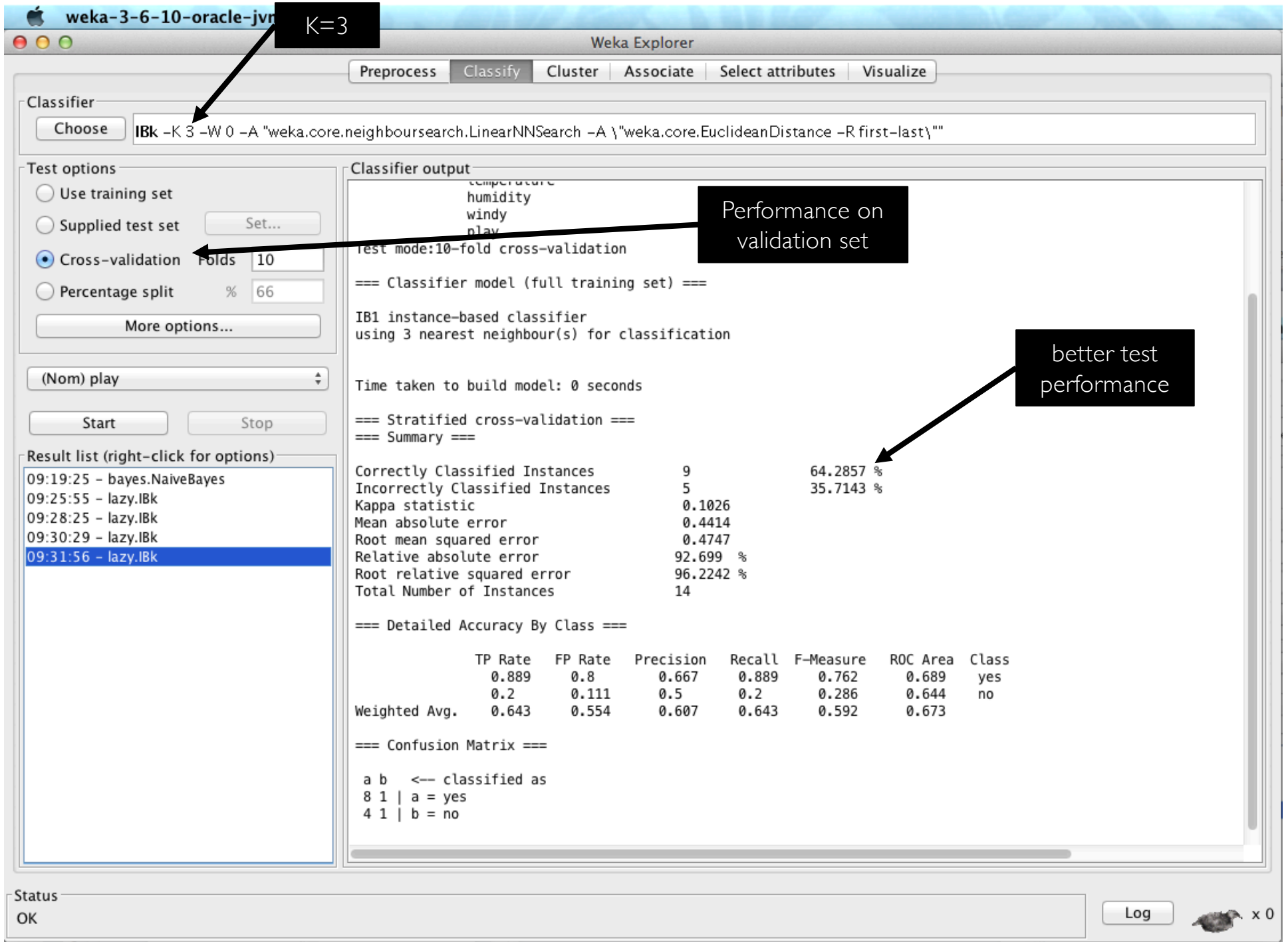
If k is too small, classification might be sensitive to noise points

If k is too large, neighborhood may include quite dissimilar examples


weka-3-6-10-oracle-jvm
K=1
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
windy
play
Test mode:evaluate on training data
Performance on training set
=== Classifier model (full training set) ===
IB1 instance-based classifier
using 1 nearest neighbour(s) for classification
Time taken to build model: 0 seconds
great performance!
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances 14 100 %
Incorrectly Classified Instances 0 0 %
Kappa statistic 1
Mean absolute error 0.0625
Root mean squared error 0.0625
Relative absolute error 13.4615 %
Root relative squared error 13.0347 %
Total Number of Instances 14
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure ROC Area Class
1 0 1 1 1 1 yes
1 0 1 1 1 1 no
Weighted Avg. 1 0 1 1 1 1
=== Confusion Matrix ===
a b <-- classified as
9 0 | a = yes
0 5 | b = no
(Nom) play
Start
Stop
Result list (right-click for options)
09:19:25 - bayes.NaiveBayes
09:25:55 - lazy.IBk
Status
OK
Log
x 0


weka-3-6-10-oracle-jvm
K=3
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
IBk -K 3 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) play
Start
Stop
Result list (right-click for options)
09:19:25 - bayes.NaiveBayes
09:25:55 - lazy.IBk
09:28:25 - lazy.IBk
Classifier output
windy
play
Test mode:evaluate on training data
Performance on training set
=== Classifier model (full training set) ===
IB1 instance-based classifier
using 3 nearest neighbour(s) for classification
Time taken to build model: 0 seconds
not great performance
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances 12 85.7143 %
Incorrectly Classified Instances 2 14.2857 %
Kappa statistic 0.6585
Mean absolute error 0.352
Root mean squared error 0.3936
Relative absolute error 75.8217 %
Root relative squared error 82.0865 %
Total Number of Instances 14
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure ROC Area Class
1 0.4 0.818 1 0.9 0.844 yes
0.6 0 1 0.6 0.75 0.844 no
Weighted Avg. 0.857 0.257 0.883 0.857 0.846 0.844
=== Confusion Matrix ===
a b <-- classified as
9 0 | a = yes
2 3 | b = no
Status
OK
Log
x 0


weka-3-6-10-oracle-jvm
K=1
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
windy
play
Test mode:10-fold cross-validation
performance evaluated on validation set
=== Classifier model (full training set) ===
IB1 instance-based classifier
using 1 nearest neighbour(s) for classification
Time taken to build model: 0 seconds
poor test performance
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 8 57.1429 %
Incorrectly Classified Instances 6 42.8571 %
Kappa statistic 0.0667
Mean absolute error 0.4911
Root mean squared error 0.5985
Relative absolute error 103.137 %
Root relative squared error 121.313 %
Total Number of Instances 14
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure ROC Area Class
0.667 0.6 0.667 0.667 0.667 0.5 yes
0.4 0.333 0.4 0.4 0.4 0.456 no
Weighted Avg. 0.571 0.505 0.571 0.571 0.571 0.484
=== Confusion Matrix ===
a b <-- classified as
6 3 | a = yes
3 2 | b = no
(Nom) play
Start
Stop
Result list (right-click for options)
09:19:25 - bayes.NaiveBayes
09:25:55 - lazy.IBk
09:28:25 - lazy.IBk
09:30:29 - lazy.IBk
Status
OK
Log
x 0


weka-3-6-10-oracle-jvm
K=3
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
IBk -K 3 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
humidity
windy
play
Test mode:10-fold cross-validation
Performance on validation set
=== Classifier model (full training set) ===
IB1 instance-based classifier
using 3 nearest neighbour(s) for classification
Time taken to build model: 0 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 9 64.2857 %
Incorrectly Classified Instances 5 35.7143 %
Kappa statistic 0.1026
Mean absolute error 0.4414
Root mean squared error 0.4747
Relative absolute error 92.699 %
Root relative squared error 96.2242 %
Total Number of Instances 14
better test performance
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure ROC Area Class
0.889 0.8 0.667 0.889 0.762 0.689 yes
0.2 0.111 0.5 0.2 0.286 0.644 no
Weighted Avg. 0.643 0.554 0.607 0.643 0.592 0.673
=== Confusion Matrix ===
a b <-- classified as
8 1 | a = yes
4 1 | b = no
(Nom) play
Start
Stop
Result list (right-click for options)
09:19:25 - bayes.NaiveBayes
09:25:55 - lazy.IBk
09:28:25 - lazy.IBk
09:30:29 - lazy.IBk
09:31:56 - lazy.IBk
Status
OK
Log
x 0

What similarity measures?

The same ones we applied for clustering!

Similarly to clustering, we must apply normalization when needed!

Finding Nearest Neighbors Efficiently

- Basic Approach
  - Linear scan of the data
  - Classification time for a single distance depends on the number of data points and the number of variables O(nd)
  - This can be prohibitive when the training set is large
- Nearest-neighbor search can be speeded up by using
  - KD-Trees
  - Ball-Trees
  - …

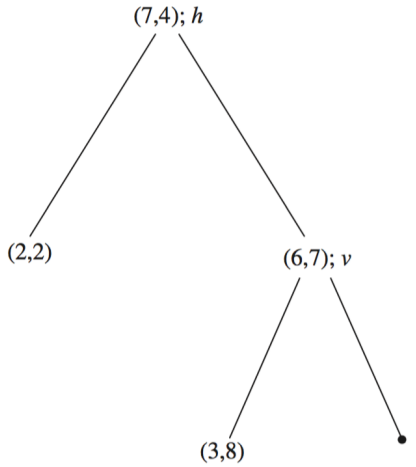
KD-Trees

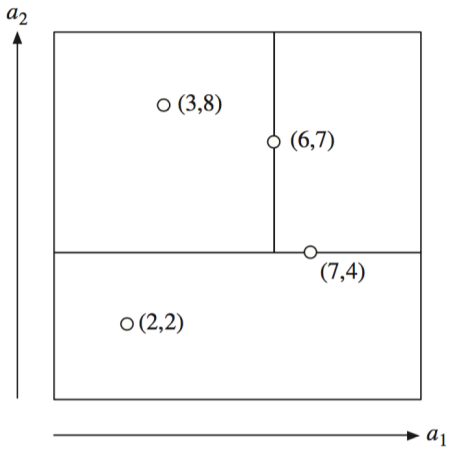
Split the space hierarchically using
a tree generated from the data

To find the neighbor of a specific example,
navigate the tree using the example

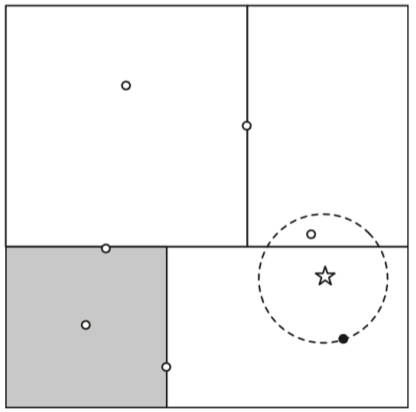
KD-Tree construction and search

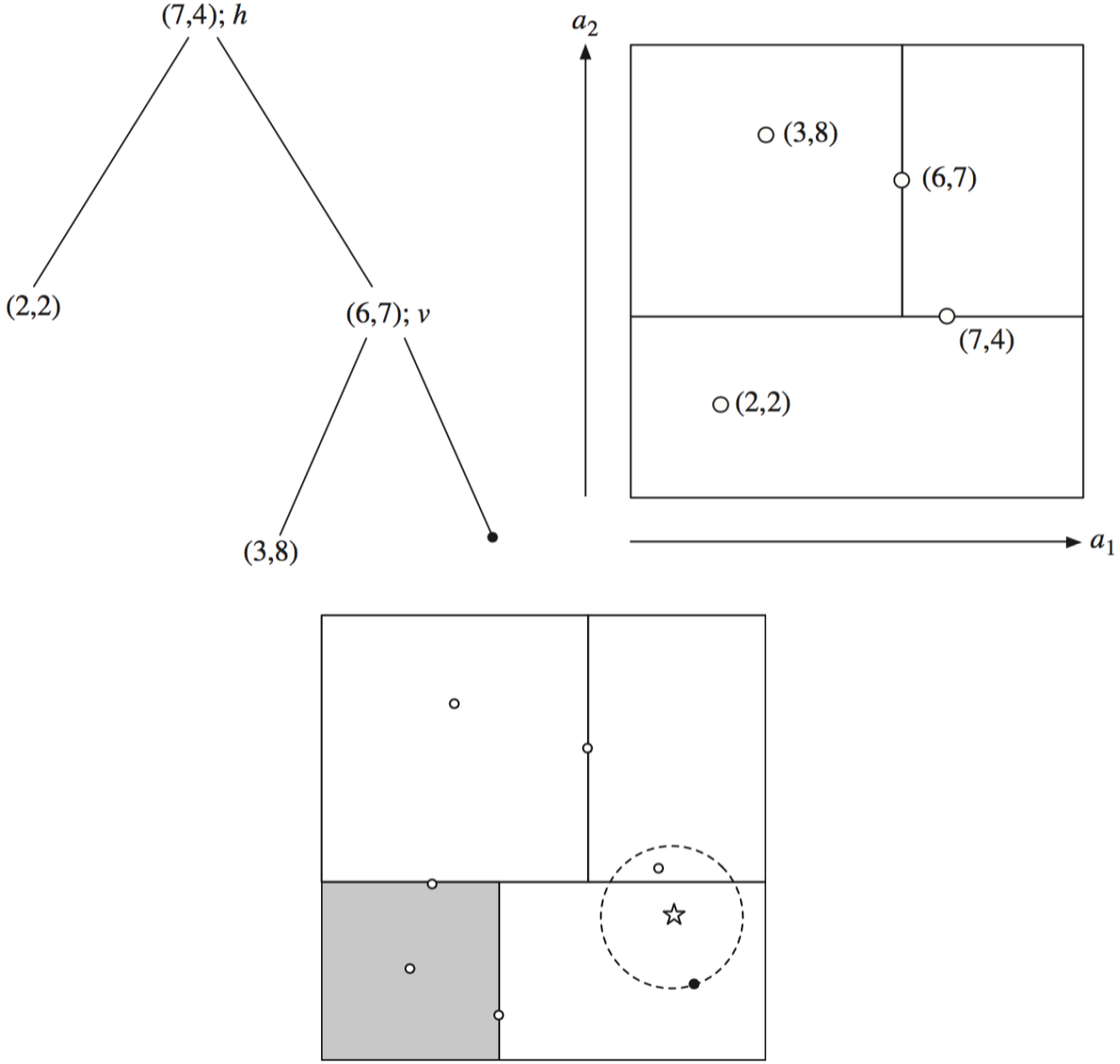
- Tree construction
  - Add points iteratively to tree
  - Each new point falls in a leaf and splits region around it based on value of one of its attributes





- Search for nearest neighbors
  - Navigate tree to reach leaf and check
  - Backtrack up tree to check nearby regions
  - Until all k-nearest neighbors found
    i.e. stop when closest region is further than k-th closest point found so far





Example of KD-tree. The tree works like a decision tree and splits the space in a hierarchy. When the class of one example is needed, the tree is navigated until a leaf is reached and only the nearby areas are actually checked by backtracking on the tree.

Effectiveness of KD-trees

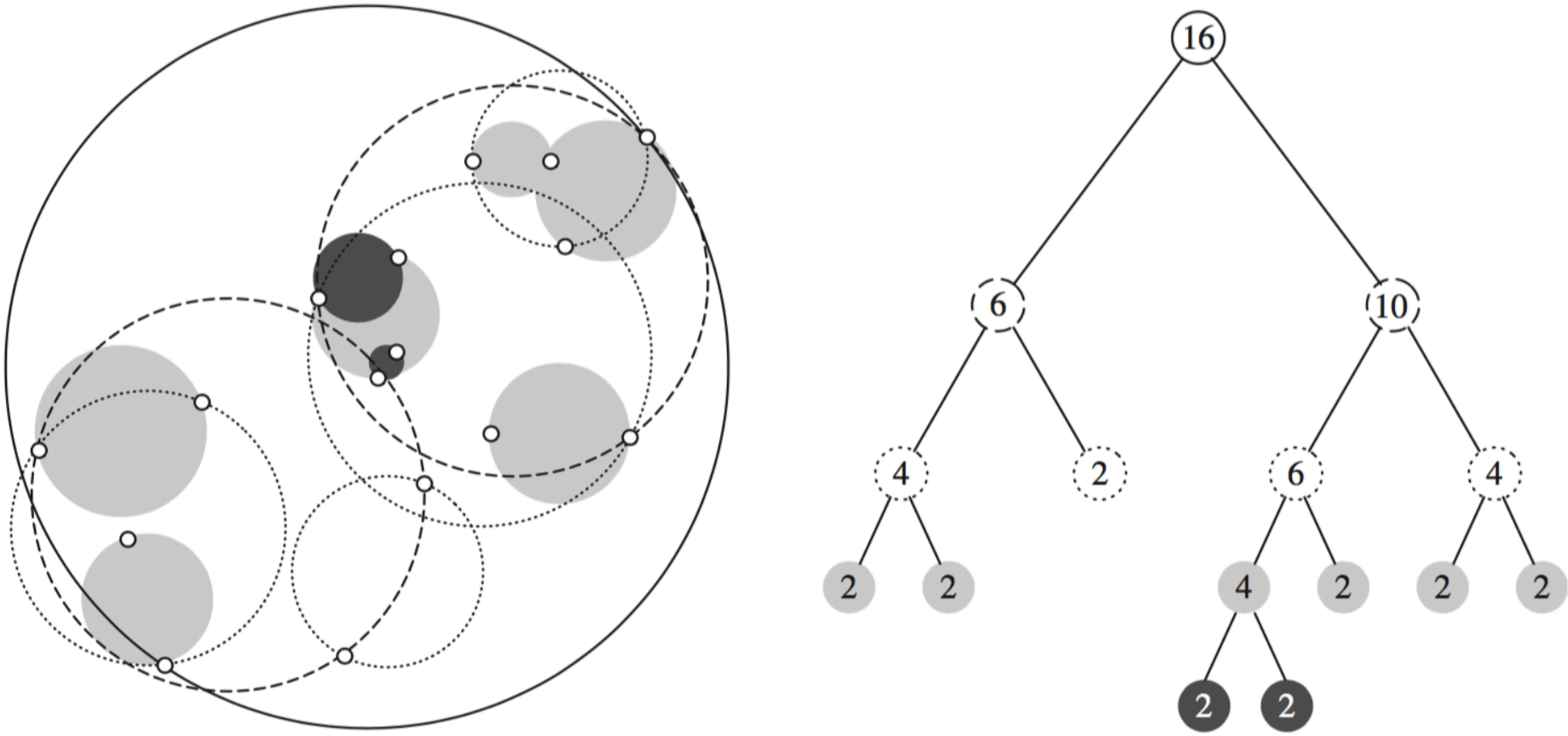
- Search complexity depends on depth of tree. It is the logarithm of number of nodes for balanced tree O(log(n))
- Occasional rebalancing of tree may be needed randomizing order of data is another option
- But amount of backtracking required depends on quality of tree some nodes are square (good) while others are skinny (bad)

More on kD-trees

- Complexity depends on depth of the tree, given by the logarithm of number of nodes for a balanced tree
- Amount of backtracking required depends on quality of tree ("square" vs. "skinny" nodes)
- How to build a good tree?
  - Need to find good split point and split direction
  - Possible split direction: direction with greatest variance
  - Possible split point: median value along that direction
- Using value closest to mean (rather than median) can be better if data is skewed
- Can apply this split selection strategy recursively just like in the case of decision tree learning

Ball Trees

- Corners in high-dimensional space may mean query ball intersects with many regions and this is a potential limitation for KD-Tree
- Note that there is no need to make sure that regions do not overlap, so they do not heed to be hyperrectangles
- Can use hyperspheres (balls) instead of hyperrectangles
- A ball tree organizes the data into a tree of k-dimensional hyperspheres
- Balls may allow for a better fit to the data and thus more efficient search



Example of Ball Tree.

Example using the loan dataset

Evaluation using 10-fold crossvalidation

kd-tree

t=11.163 Accuracy=0.806 Std=0.00

Brute force

t=567.895 Accuracy=0.806 Std=0.001

k-Nearest Neighbor Regression

- The regression methods we previously discussed were parametric
  - They assume an approximation function parametrized by a weight vector
  - The regression algorithm search for the best weight vector
  - E.g., linear regression assumes that the target is computed as a linear combination of the weight vector w and the feature vector h(x)
- Nearest Neighbor regression fits each point locally
- Basic Algorithm
  - Given a dataset $(x_1,y_1) \ldots (x_N,y_N)$
  - Given a query point x_q
  - The value y_q associated to x_q is computed as a local interpolation of the targets associated to neighbor points
- Prediction can use the plain average of the k nearest targets or a weighted average of the same targets
- Prediction can use kernel functions that take the distance as input and return a weight (e.g., a Gaussian function of the distance)

Discussion on k-Nearest Neighbor

- K-Nearest Neighbor is often very accurate but slow, since simple version scans entire training data to derive a prediction
- Assumes all attributes are equally important, thus may need attribute selection or weights
- Possible remedies against noisy instances:
  - Take a majority vote over the k nearest neighbors
  - Remove noisy instances from dataset (difficult!)
- Statisticians have used k-NN since early 1950s,
  If $n \rightarrow \infty$ and $k/n \rightarrow 0$, error approaches minimum
- Data Structures
  - kD-trees can become inefficient when the number of attributes is too large
  - Ball trees are may help

Study Material

- Mining of Massive Datasets Section 12.3, 12.4
- Data Mining and Analysis: Fundamental Concepts and Algorithms - Chapter 18 & 19
- Chapter 4 of Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques - Ian H. Witten et al.