Accuracy, Cross-Validation, Overfitting, and ROC

Slides adopted from Data Mining for Business Analytics

Stern School of Business
New York University
Spring 2014

Evaluation

How do we measure generalization performance?

Evaluating Classifiers: Plain Accuracy

$$\text{Accuracy} = \frac{\text{Number of correct decisions made}}{\text{Total number of decisions made}}$$

$$= 1 - \text{error rate}$$

- *Too* simplistic..

Evaluating Classifiers: The Confusion Matrix

- A confusion matrix for a problem involving n classes is an $n \times n$ matrix,
  - with the columns labeled with actual classes and the rows labeled with predicted classes

- It separates out the decisions made by the classifier,
  - making explicit how one class is being confused for another

	p	n
Y	True Positives	False Positives
N	False Negatives	True Negatives

- The errors of the classifier are the false positives and false negatives

Building a Confusion Matrix

Default Truth	Model Prediction
0	0
1	1
0	1
0	1
0	0
1	1
0	0
0	0
1	1
1	0

Predicted class Actual class	Default	No Default	Total
Default	3	1	4
No Default	2	4	6
Total	5	5	10

Other Evaluation Metrics

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

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

- $\text{F}-\text{measure} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

Over-fitting the data

- Finding chance occurrences in data that look like interesting patterns, but which do not generalize, is called over-fitting the data

- We want models to apply not just to the exact training set but to the general population from which the training data came
  - Generalization

Over-fitting

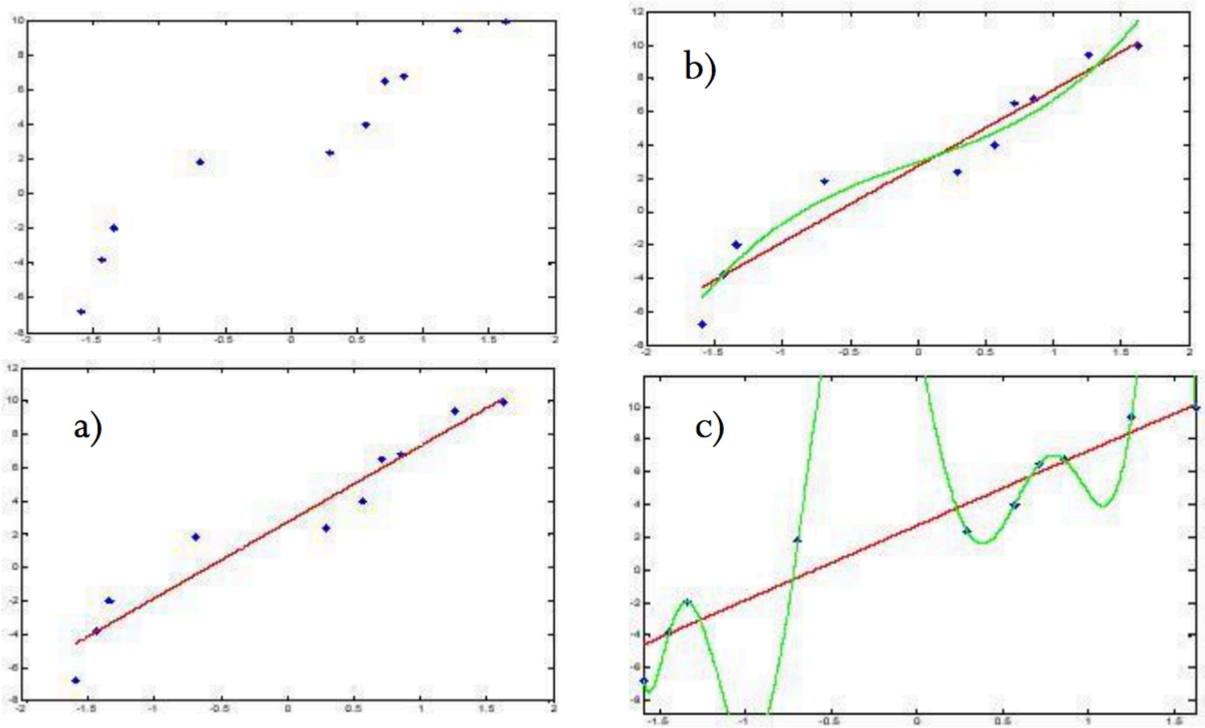
- The tendency of DM procedures to tailor models to the training data, *at the expense of generalization* to previously unseen data points.

- All data mining procedures have the tendency to over-fit to some extent
  - Some more than others.

- "If you torture the data long enough, it will confess"

- There is no single choice or procedure that will eliminate over-fitting
  - recognize over-fitting and manage complexity in a principled way.

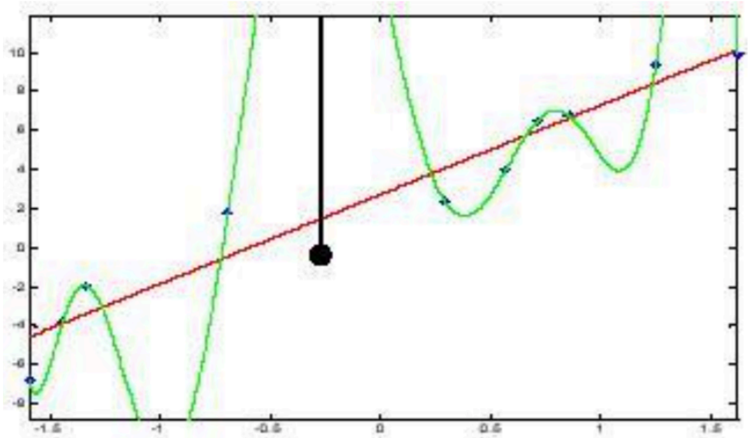
What's over-fitting ?



42

Introduction to Machine Learning: Decision Tree Learning

What's over-fitting ?



- $h \in H$ overfits training data if there's an alternative $h' \in H$ such that:

  $err_{train}(h) < err_{train}(h')$

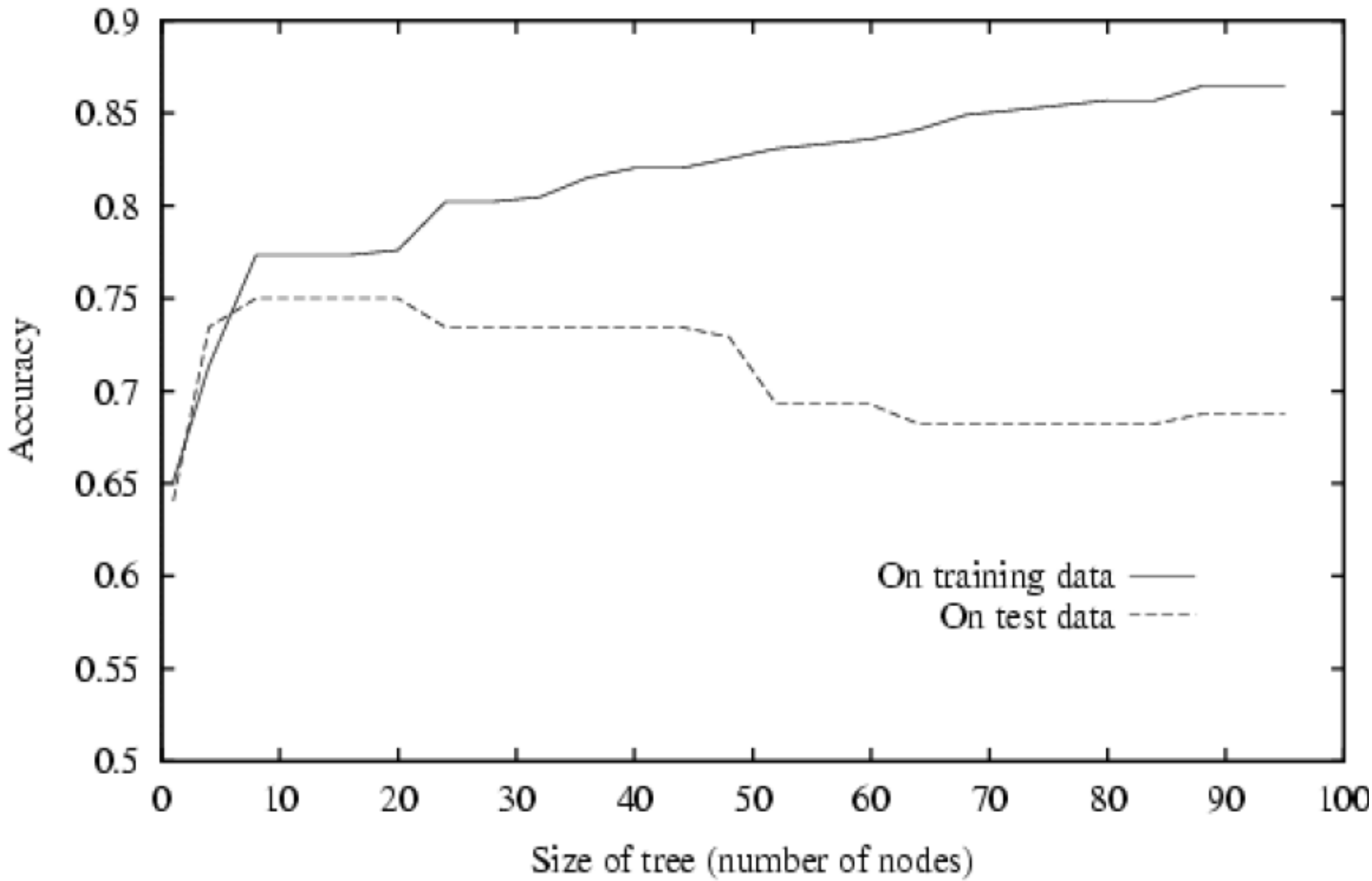
  AND

  $err_{test}(h) > err_{test}(h')$

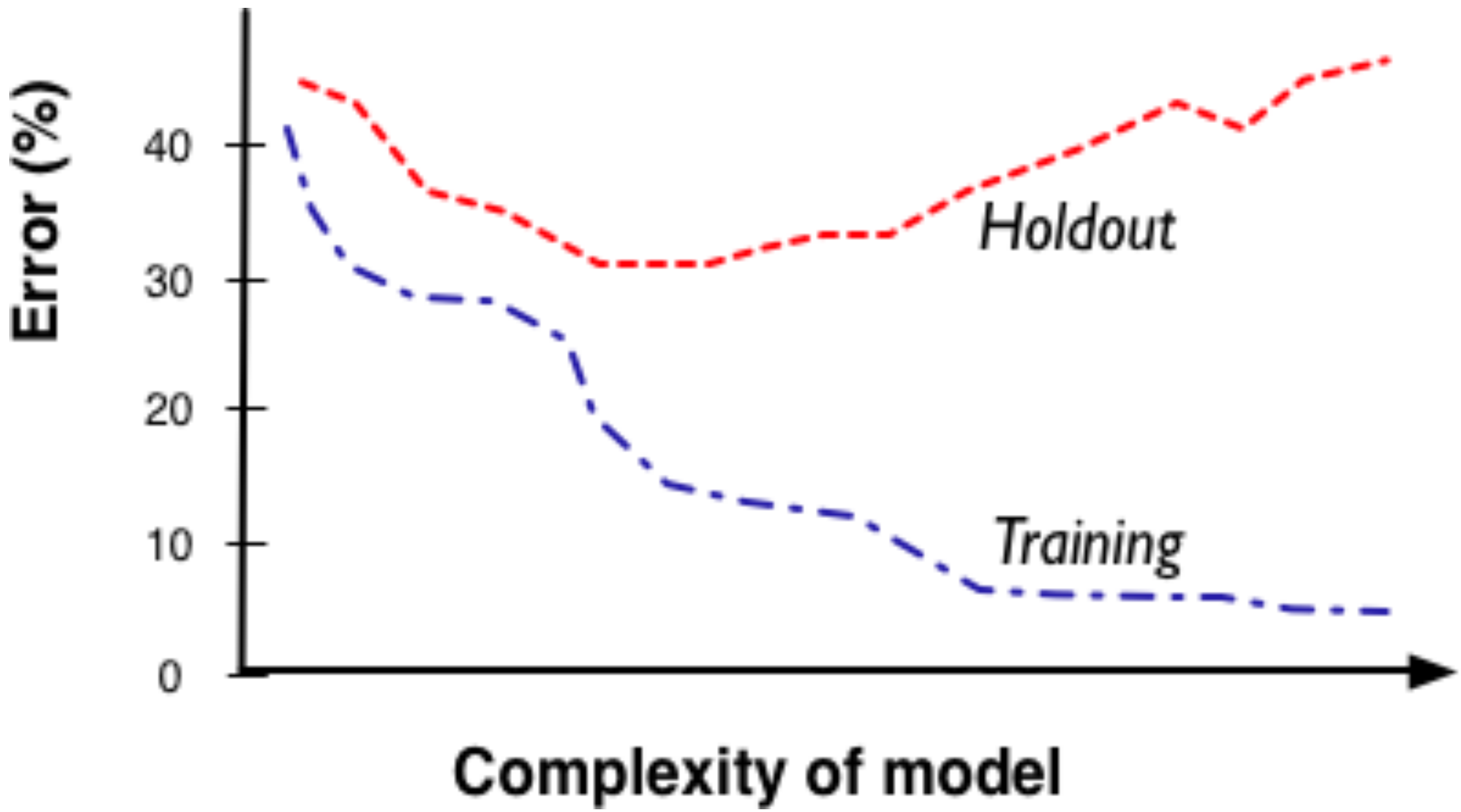
An example of over-fitting in DTree

- Each leaf corresponds to a single training point and the full tree is merely a convenient implementation of a lookup table

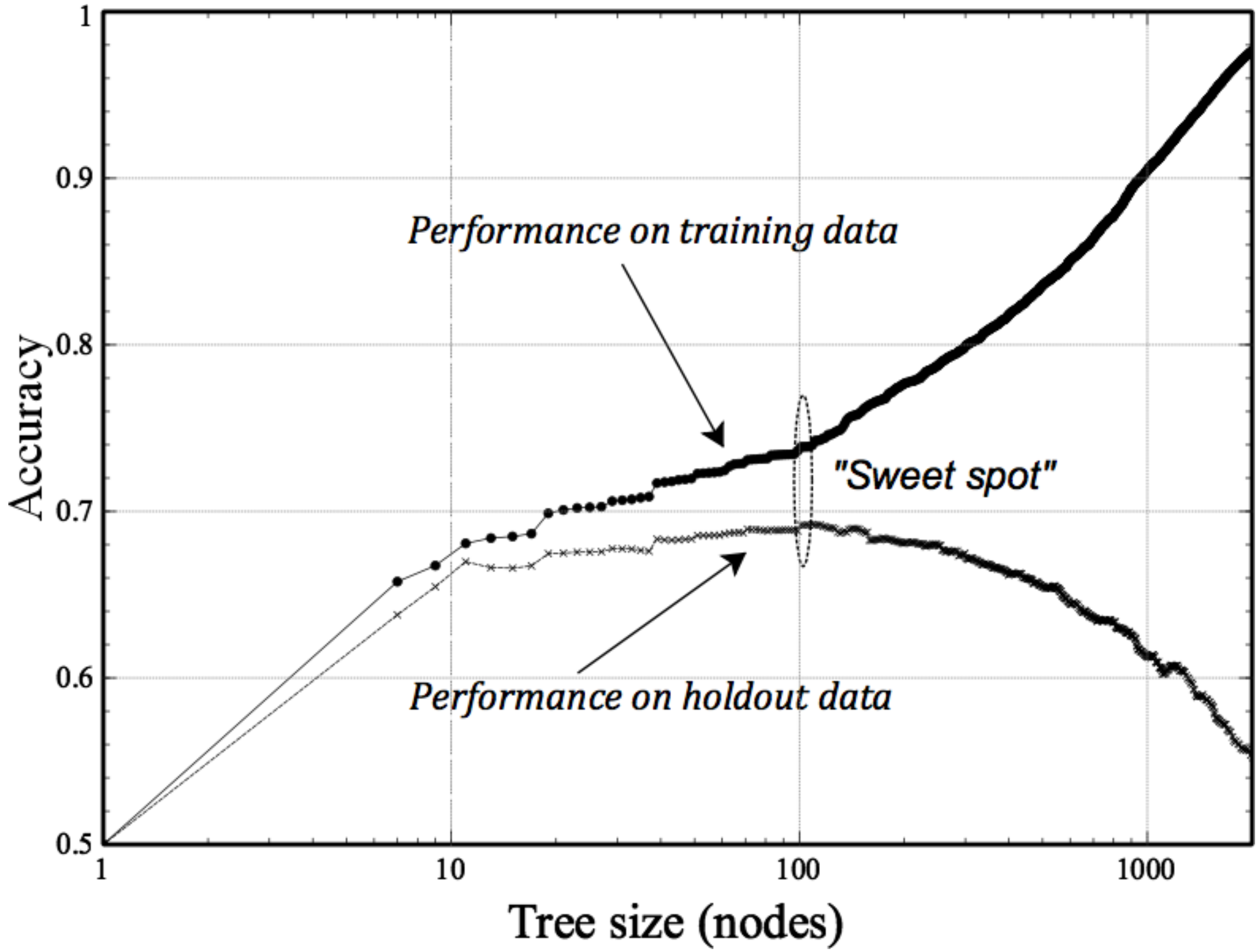
Tree Complexity and Over-fitting



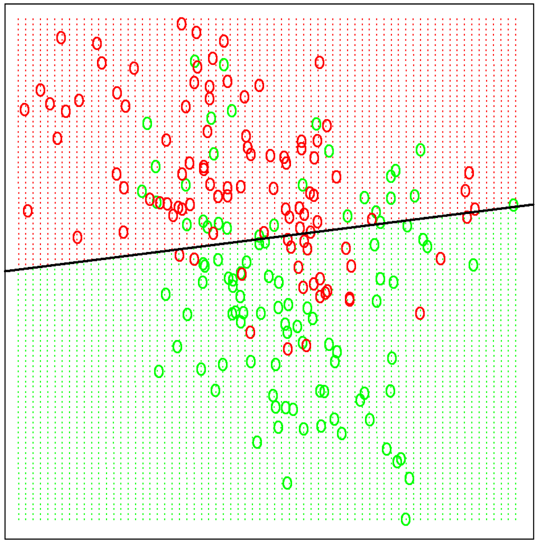
Fitting Graph



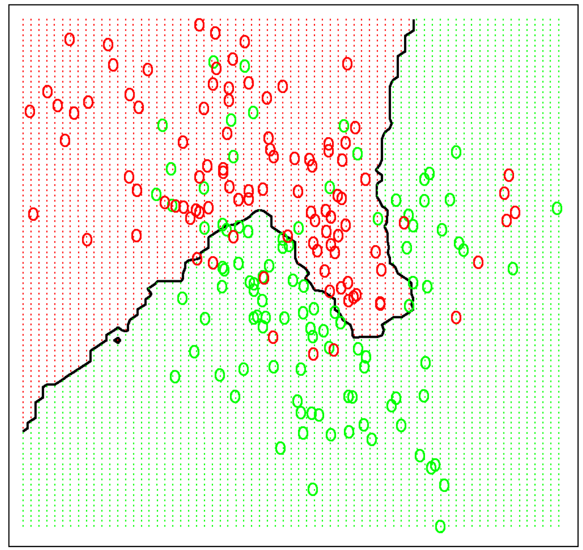
Over-fitting in tree induction



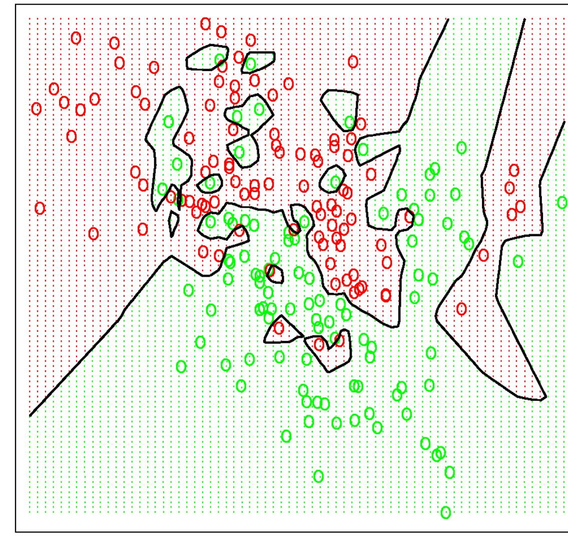
Need for holdout evaluation



Under-fitting



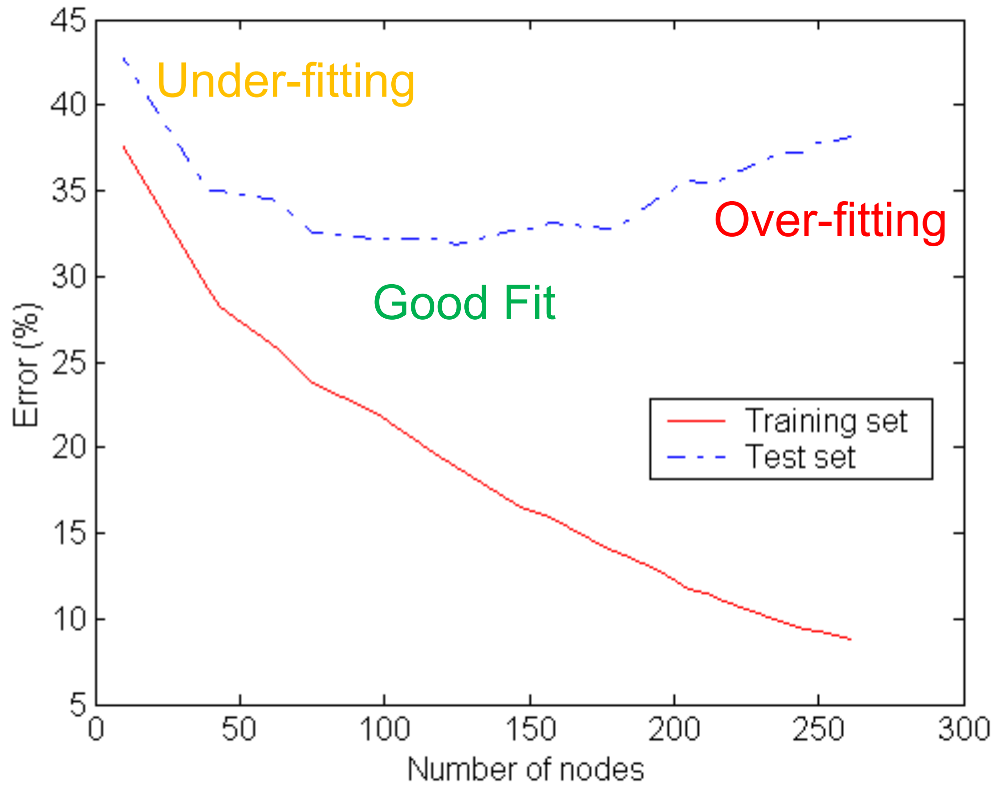
Good



Over-fitting

- In sample evaluation is in favor or “memorizing”
- On the *training data* the right model would be best
- But on *new data* it would be bad

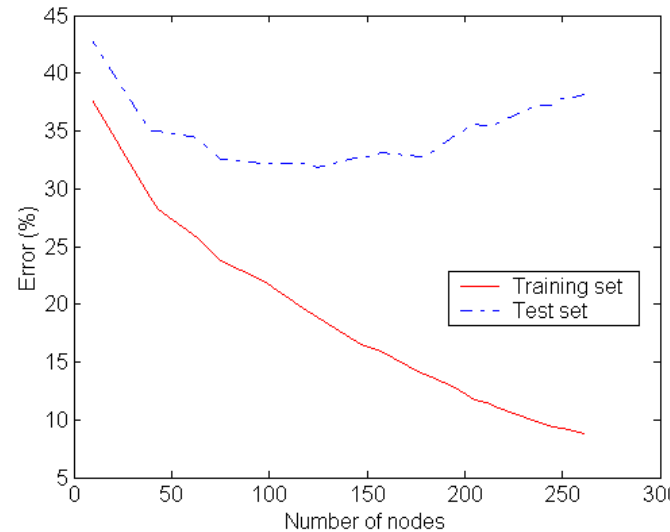
Over-fitting



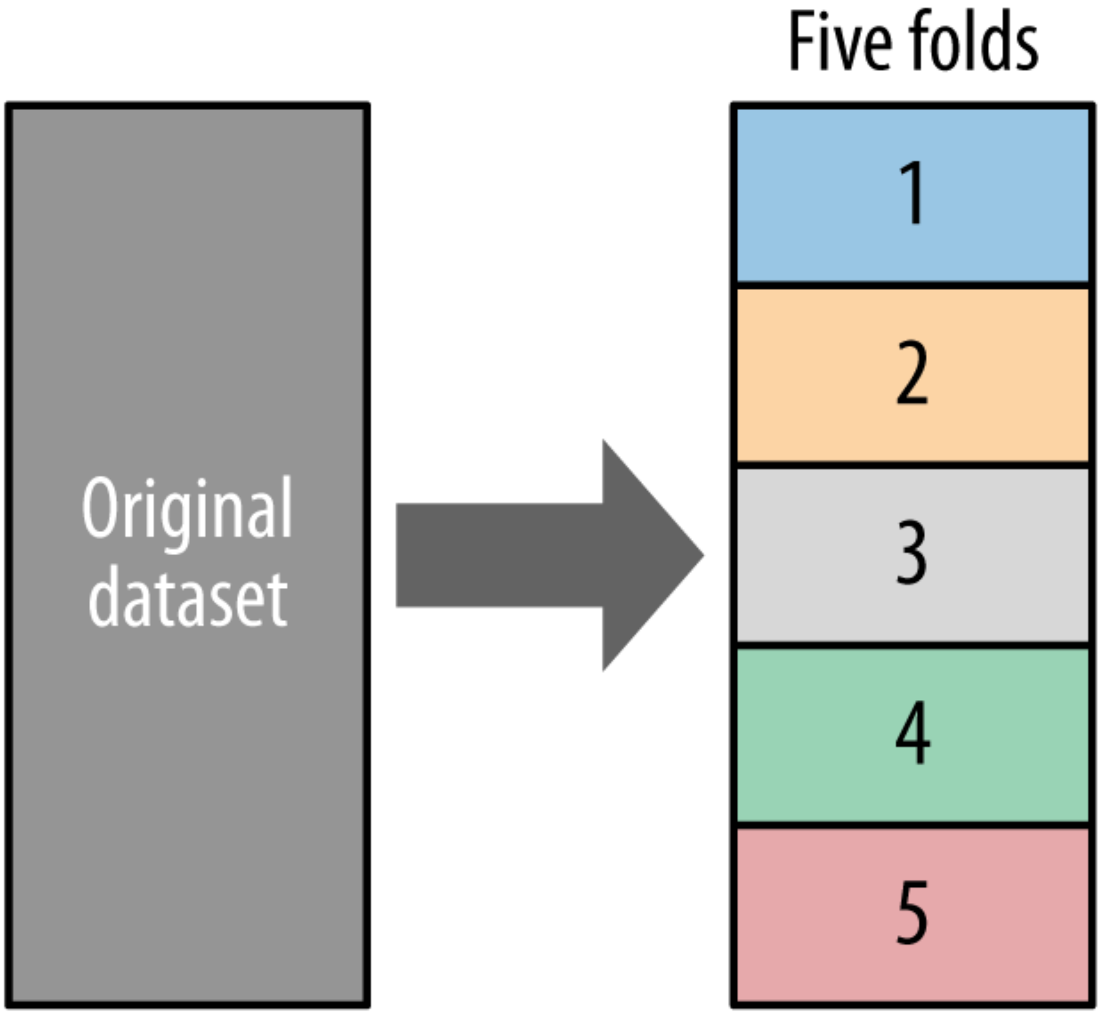
- Over-fitting: Model "memorizes" the properties of the particular training set rather than learning the underlying concept or phenomenon

Holdout validation

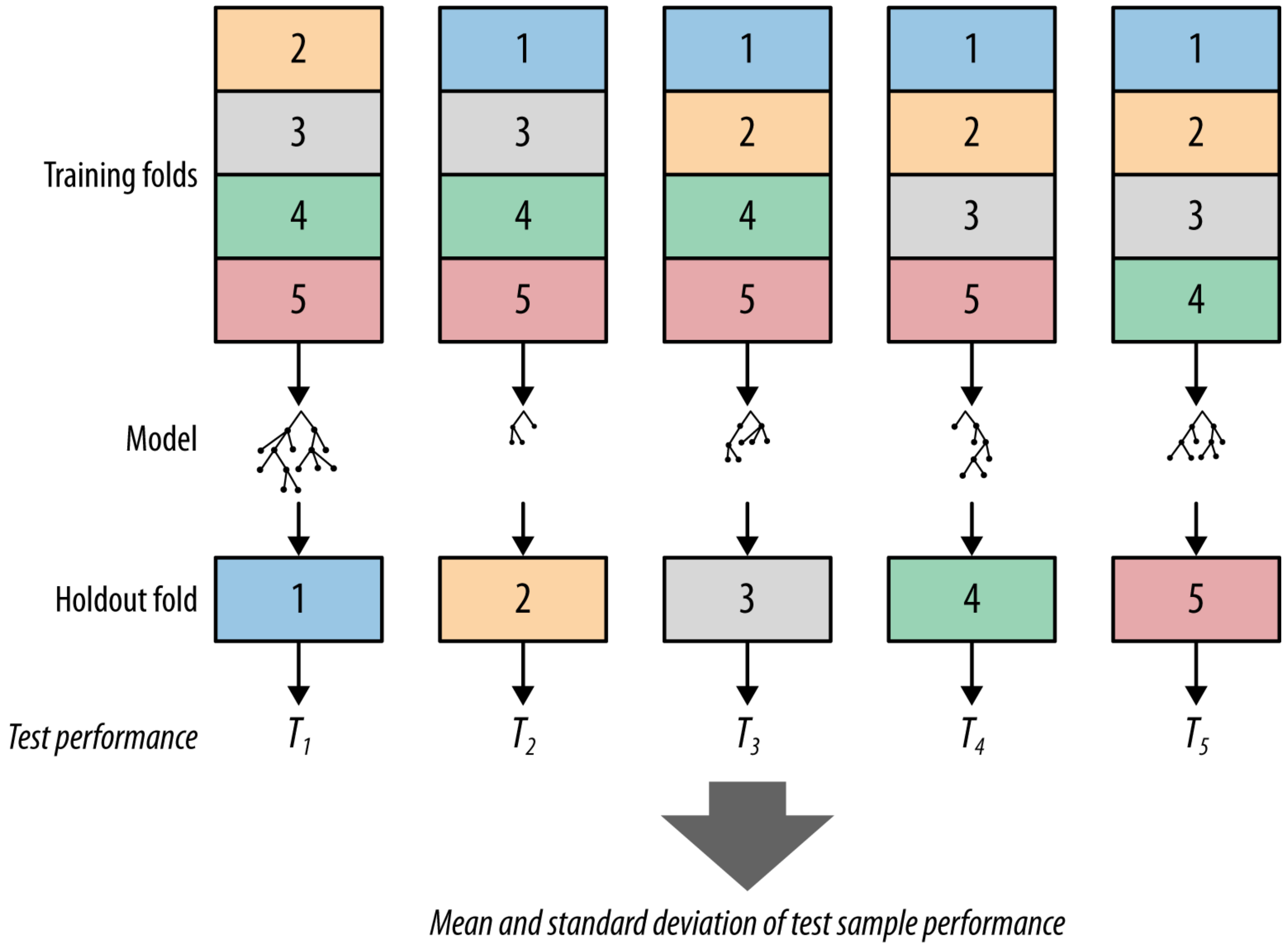
- We are interested in generalization
  - The performance on data not used for training
- Given only one data set, we hold out some data for evaluation
  - Holdout set for final evaluation is called the test set
- Accuracy on training data is sometimes called **"in-sample" accuracy**, vs. **"out-of-sample" accuracy** on test data



Cross-Validation



Cross-Validation



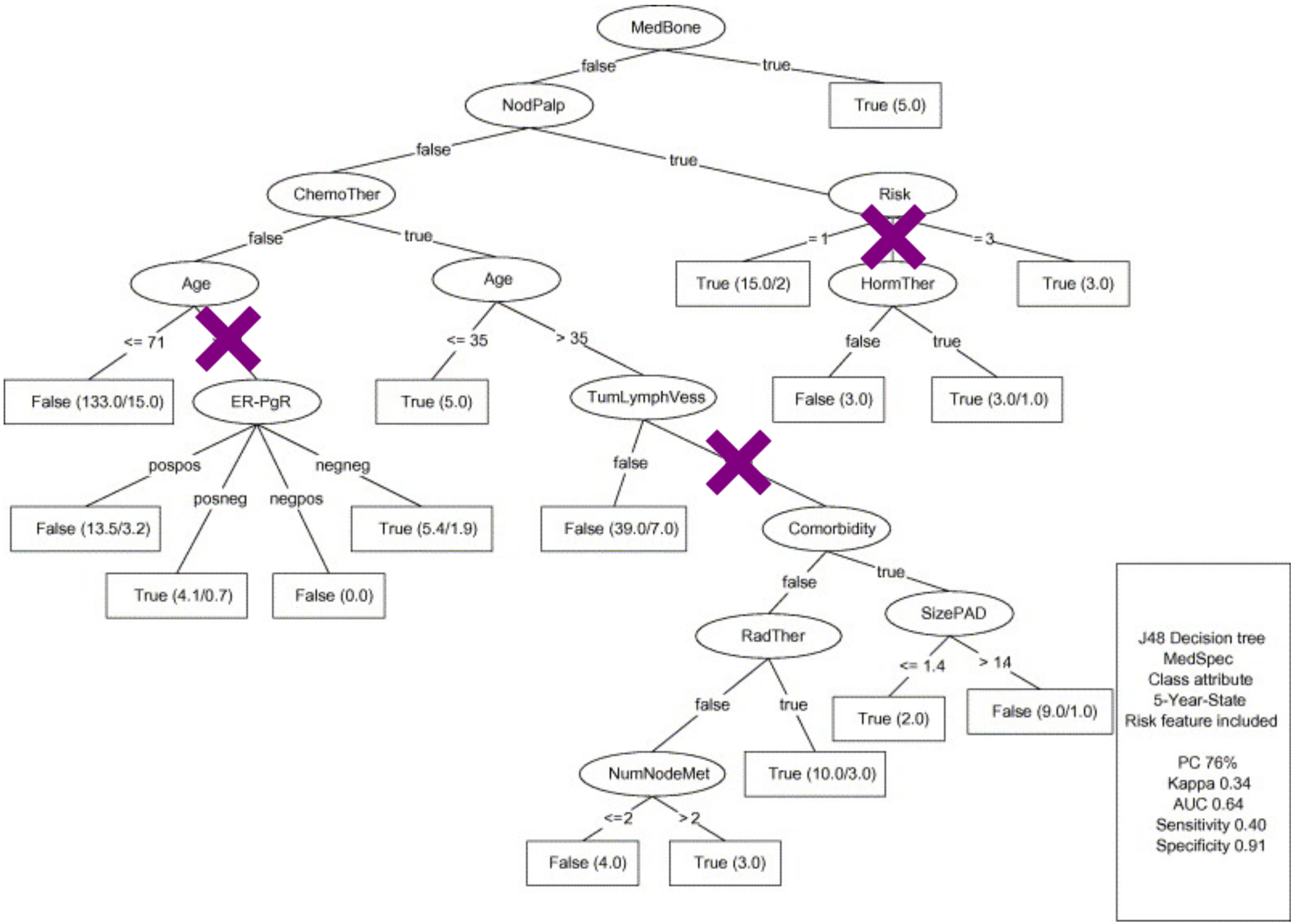
From Holdout Evaluation to Cross-Validation

- Not only a simple estimate of the generalization performance, but also some statistics on the estimated performance,
  - such as the mean and variance
- Better use of a limited dataset
  - Cross-validation computes its estimates over *all* the data
- Used for comparing different learning procedure
  - e.g. Decision Trees vs Logistic Regression
- Used for comparing hyper-parameters in a specific procedure
  - e.g. the maximum depth (minimum amount of data in the leaf node) of the decision tree.

Pruning

- Pruning simplifies a decision tree to prevent over-fitting to noise in the data
- **Post-pruning**:
  - takes a fully-grown decision tree and discards unreliable parts
- **Pre-pruning**:
  - stops growing a branch when information becomes unreliable
    - too few data points (e.g. <5% of the total) after splitting
    - Trees are too deep
- Post-pruning preferred in practice

Post-pruning a tree



Avoid over-fitting

- Two ways of avoid over-fitting for DTree
  I. Stop growing when data split not statistically significant (pre-pruning)
  II. Grow full tree, then post-pruning

For Option I:

Hard to estimate the
size of the tree

Pre-Pruning: When to stop splitting (I) Number of instances

- Frequently, a node is not split further if
  - The number of training instances reaching a node is smaller than a certain percentage of the training set
    - (e.g. 5%)
  - Regardless the impurity or error.
  - Any decision based on too few instances causes variance and thus generalization error.

Pre-Pruning: When to stop splitting
(2) Threshold of information gain value

- Set a small threshold value, splitting is stopped if

  $\Delta i(s) \leq \beta$

- Benefits: Use all the training data. Leaf nodes can lie in different levels of the tree.
- Drawback: Difficult to set a good threshold

Avoid over-fitting

- Two ways of avoid over-fitting for D-Tree

  I. Stop growing when data split not statistically significant (pre-pruning)

  II. Grow full tree, then post-pruning

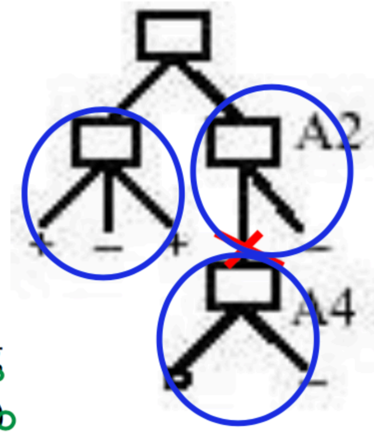
For option II:

- How to select "best" tree?
  - Measure performance over training data (statistical pruning)
    - Confidence level (will be introduced later)
  - Measure performance over separate validation data set
- MDL (Minimize Description Length 最小描述长度):

  minimize (*size*(tree) + *size*(misclassifications(tree)))

Post-pruning (1). Reduced-Error pruning

- Split data into training set and validation set
  - Validation set:
    - Known label
    - Test performance
    - No model updates during this test!
- Do until further pruning is harmful:
  - Evaluate impact on validation set of pruning each possible node (plus the subtree it roots)
  - Greedily remove the one that most improves validation set accuracy

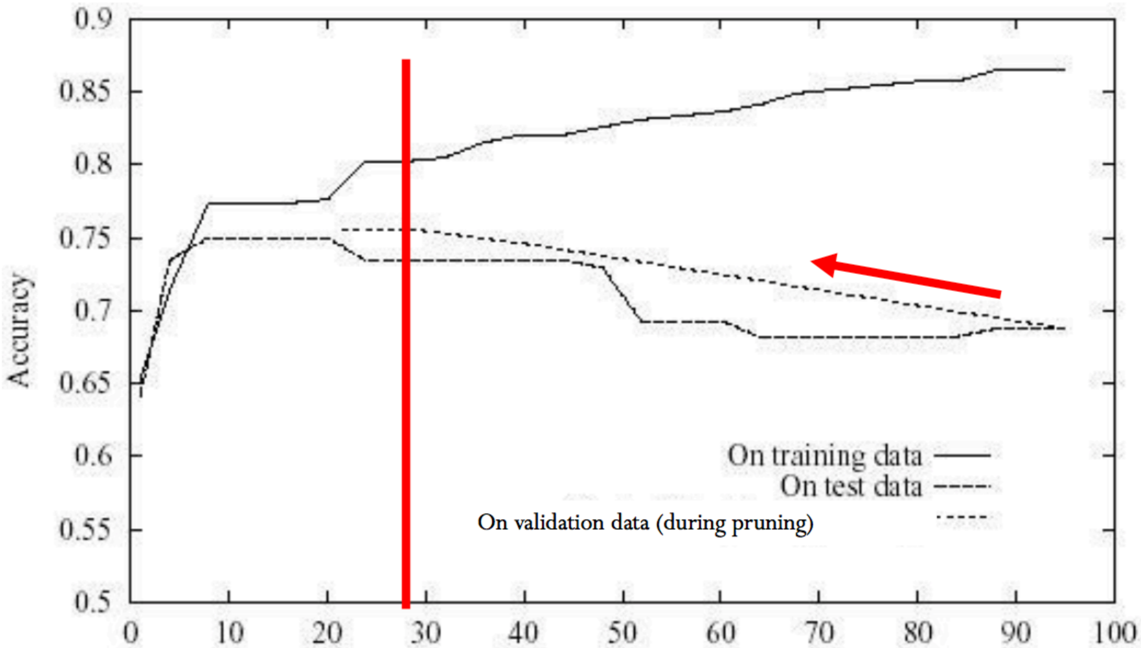


How to assign the label of the new leaf node?

Supplement: strategies of the new leaf node label after pruning

- Assign the most common class.
- Give the node multiple-class labels
  - Each class has a support degree (based on the number of the training data with each label)
  - On test: select one class with probability, or select multiple classes
- If it is the regression tree (numeric labels), can be averaged, or weighted average.
- ……

Effect of Reduced-Error pruning



Introduction to Machine Learning: Decision Tree Learning

Post-pruning (2). Rule Post-pruning

1, Convert tree to equivalent set of rules

- e.g. if (outlook=sunny)$\wedge$(humidity=high) then playTennis = no

2, Prune each rule by removing any preconditions that result in improving its estimated accuracy

- i.e. (outlook=sunny), (humidity=high)

3, Sort rules into desired sequence (by their estimated accuracy) .

4, Use the final rules in the same sequence when classifying instances.

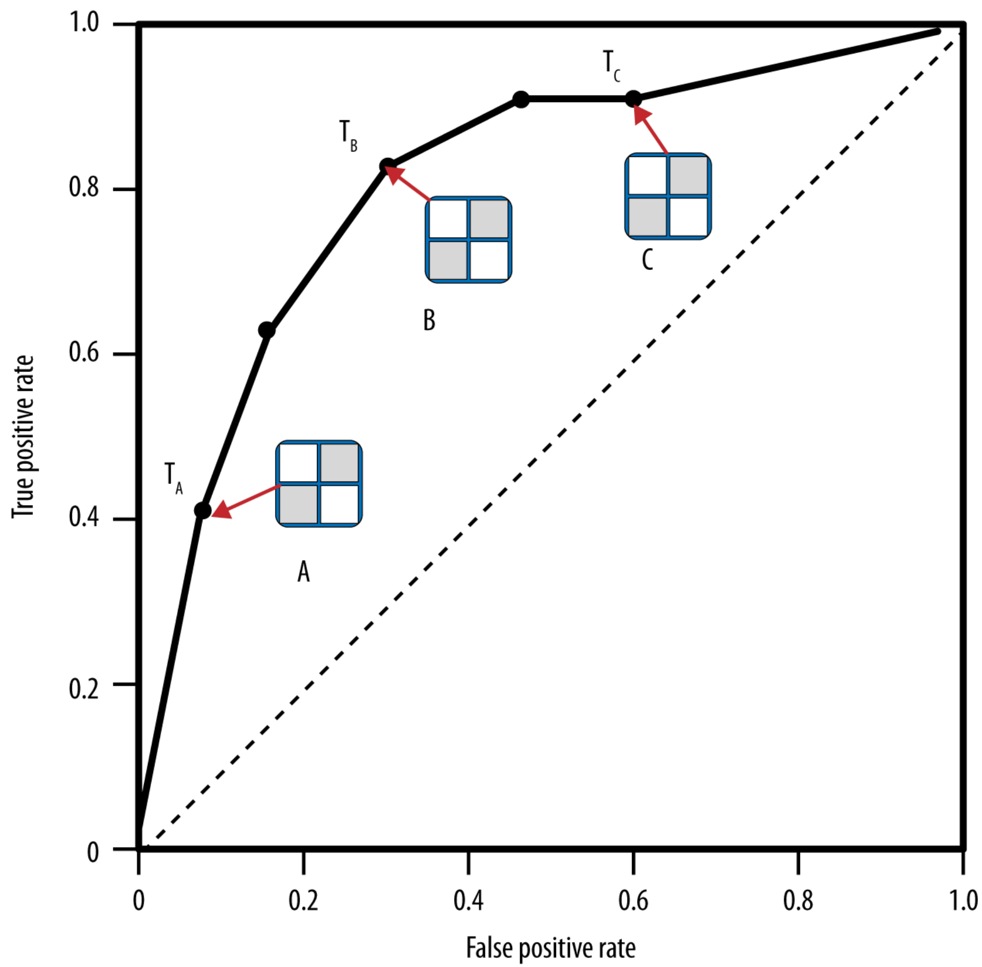
(after the rules are pruned, it may not be possible to write them back as a tree anymore.)

One of the most frequently used methods, e.g. in C4.5.

Why convert the decision tree to rule before pruning?

- Independent to contexts.
  - Otherwise, if the tree were pruned, two choices:
    - Remove the node completely, or
    - Retain it there.
- No difference between root node and leaf nodes.
- Improve readability

ROC Graphs and Curves



Generating ROC curve: Algorithm

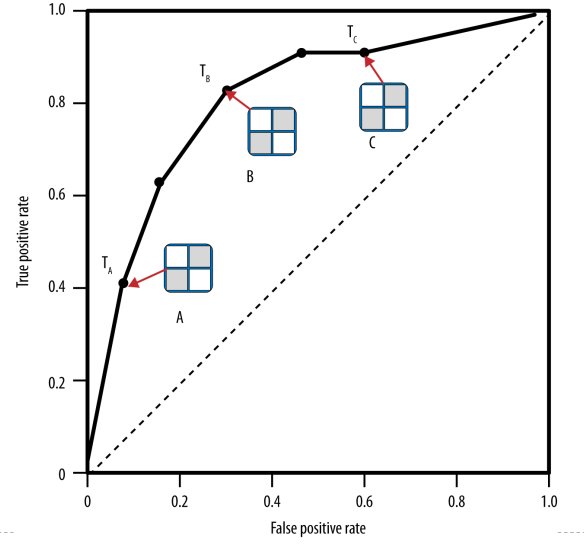
- For each test, count the number of true positives TP (positives with prediction above the cutoff) and false positives FP (negatives above the cutoff)
- Calculate TP rate (TP/P) and FP (FP/N) rate
- Plot current number of TP/P as a function of current FP/N

ROC Graphs and Curves

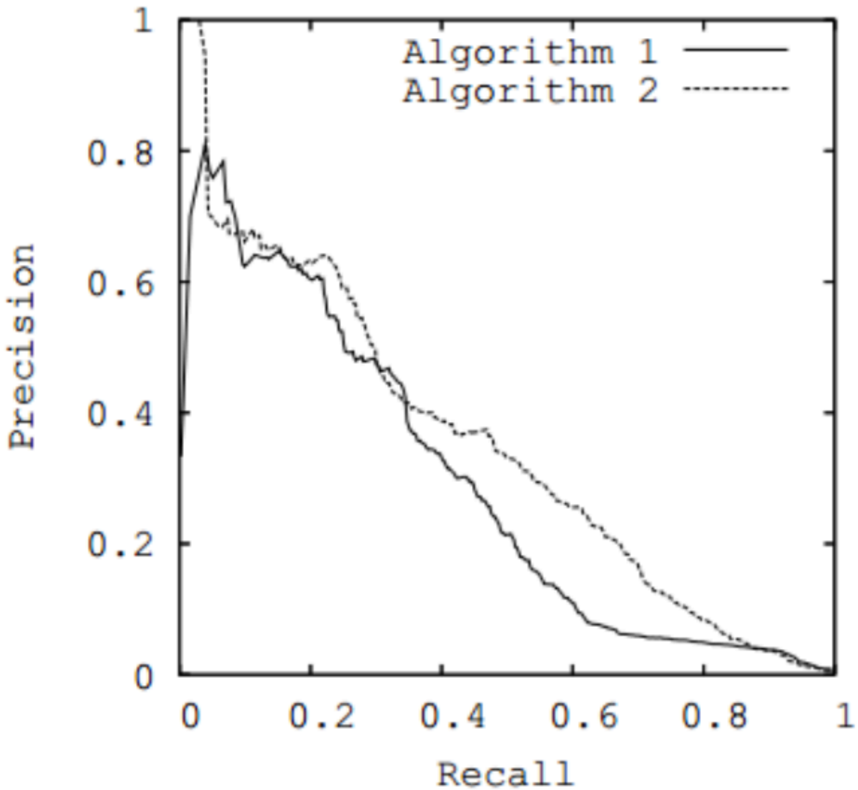
- ROC graphs decouple classifier performance from the conditions under which the classifiers will be used
- ROC graphs are independent of the class proportions as well as the costs and benefits
- Not the most intuitive visualization for many business stakeholders

Area Under the ROC Curve (AUC)

- The area under a classifier's curve expressed as a fraction of the unit square
  - Its value ranges from zero to one
- The AUC is useful when a single number is needed to summarize performance, or when nothing is known about the operating conditions
  - A ROC curve provides more information than its area



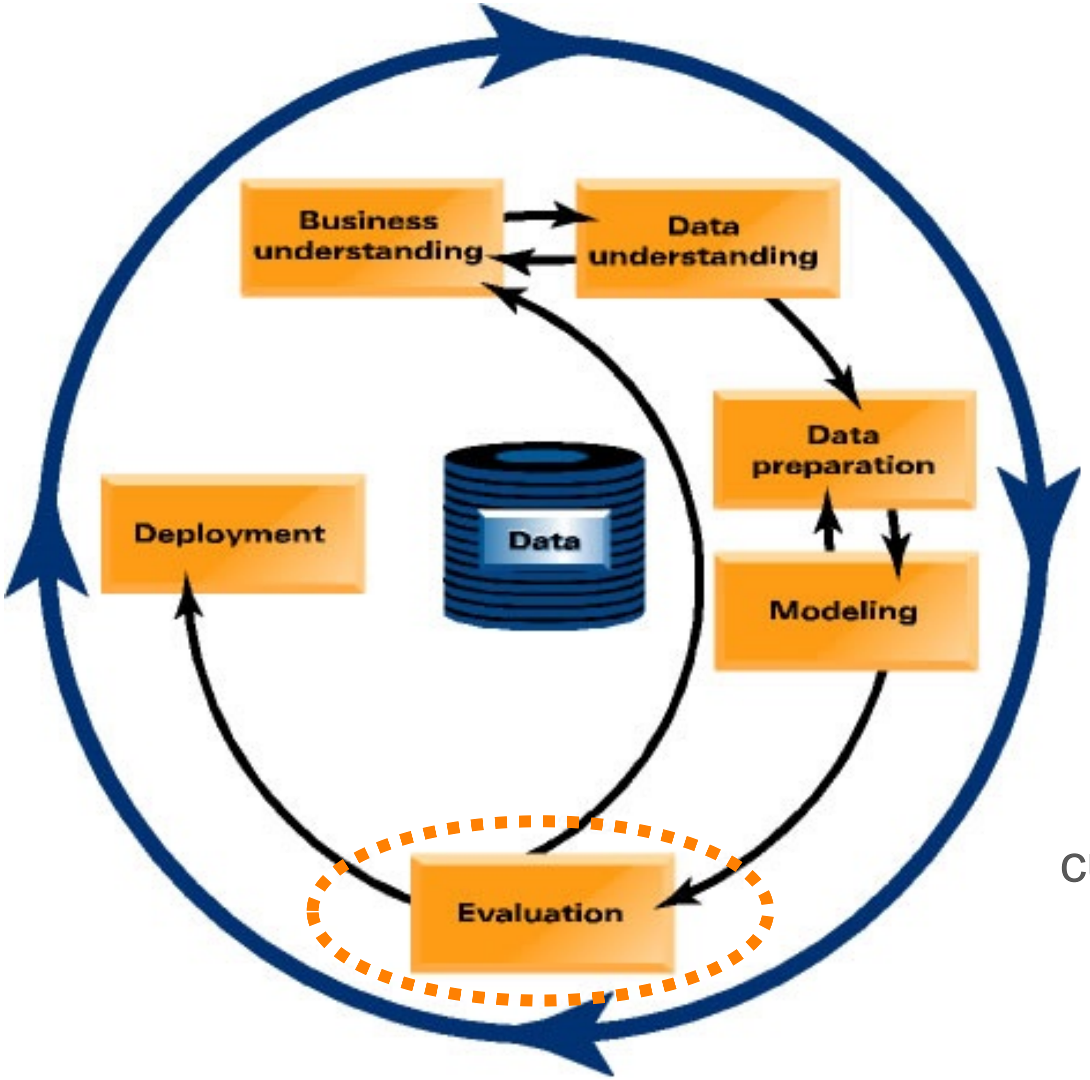
P-R Curve: Tradeoff between Precision and Recall



- Precision $= \frac{TP}{TP+FP}$
- Recall $= \frac{TP}{TP+FN}$

AUPRC: Area under P-R Curve

Let’s focus back in on actually mining the data..



Which model should TelCo select in order to target customers with a special offer, prior to contract expiration?

Performance Evaluation

Training Set:

Model	Accuracy
Classification Tree	95%
Logistic Regression	93%
k-Nearest Neighbors	100%
Naïve Bayes	76%

Test Set:

Model	Accuracy	AUC
Classification Tree	91.8%$\pm$0.0	0.614$\pm$0.014
Logistic Regression	93.0%$\pm$0.1	0.574$\pm$0.023
k-Nearest Neighbors	93.0%$\pm$0.0	0.537$\pm$0.015
Naïve Bayes	76.5%$\pm$0.6	0.632$\pm$0.019

Performance Evaluation

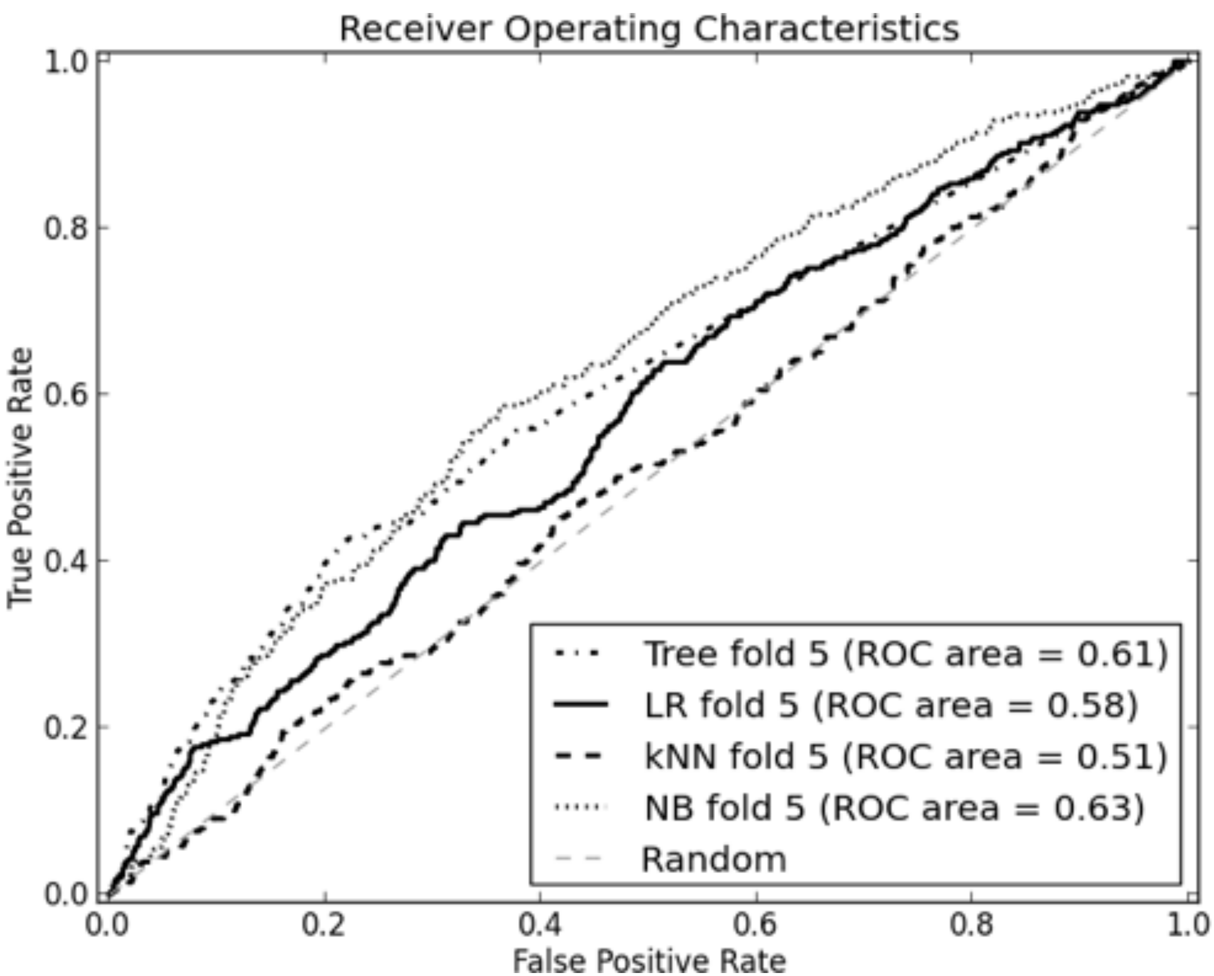
Naïve Bayes confusion matrix:

	p	n
Y	127 (3%)	848 (18%)
N	200 (4%)	3518 (75%)

k-Nearest Neighbors confusion matrix:

	p	n
Y	3 (0%)	15 (0%)
N	324 (7%)	4351 (93%)

ROC Curve


Receiver Operating Characteristics
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Tree fold 5 (ROC area = 0.61)
LR fold 5 (ROC area = 0.58)
kNN fold 5 (ROC area = 0.51)
NB fold 5 (ROC area = 0.63)
Random

Thanks!

Questions?