Statistisches Data Mining (StDM)
Woche 11

Oliver Dürr

Institut für Datenanalyse und Prozessdesign

Zürcher Hochschule für Angewandte Wissenschaften

oliver.duerr@zhaw.ch

Winterthur, 29 November 2016

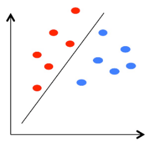
No laptops,
no phones, no problems

Multitasking senkt Lerneffizienz:

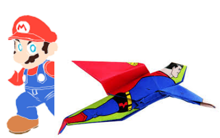
- **Keine Laptops im Theorie-Unterricht Deckel zu oder fast zu (Sleep modus)**

Overview of classification (until the end to the semester)

Classifiers



K-Nearest-Neighbors (KNN)
Logistic Regression
Linear discriminant analysis
Support Vector Machine (SVM)
Classification Trees
Neural networks NN
Deep Neural Networks (e.g. CNN, RNN)
...



Combining classifiers

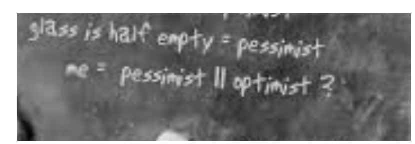
Bagging
Boosting
Random Forest

Evaluation

Cross validation
Performance measures
ROC Analysis / Lift Charts

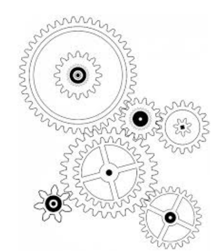
Theoretical Guidance / General Ideas

Bayes Classifier
Bias Variance Trade off (Overfitting)



Feature Engineering

Feature Extraction
Feature Selection



Decision Trees
Chapter 8.1 in ILSR

Note on ISLR

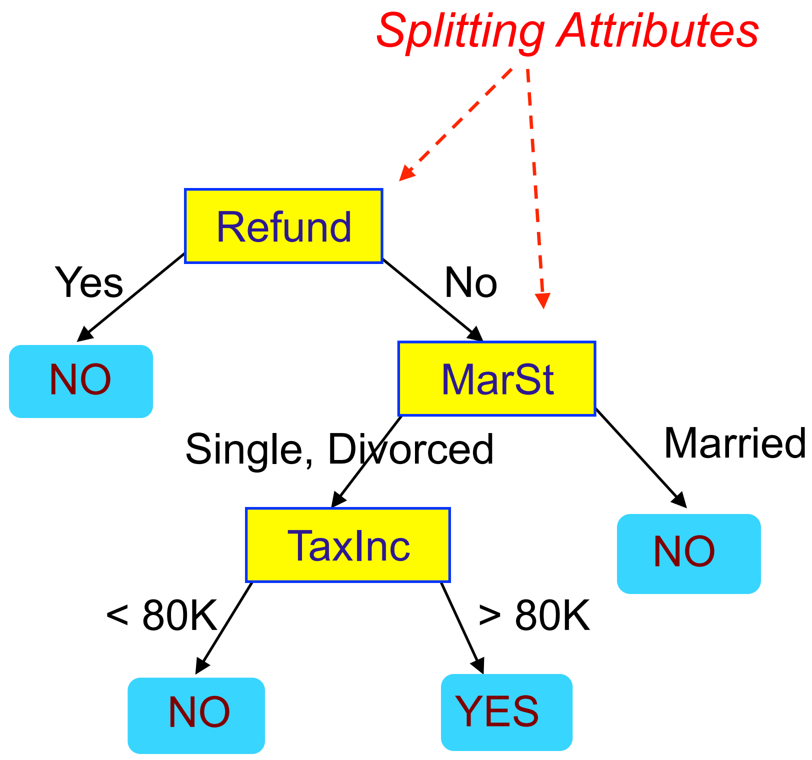
- In ISLR they also include trees for regression. Here we focus on trees for classification

Example of a Decision Tree for classification

categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



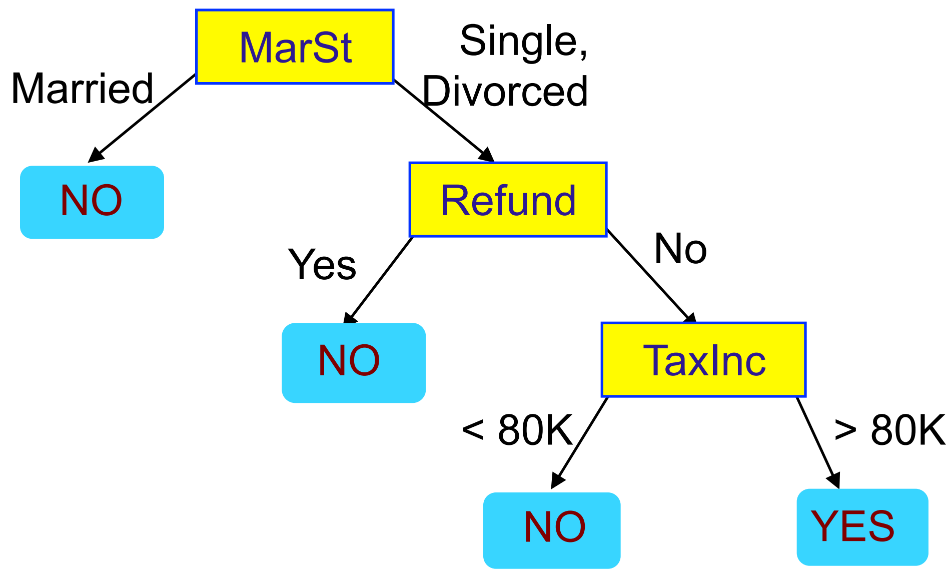
Model: Decision Tree

Source: Tan, Steinbach, Kumar

Another Example of Decision Tree

categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

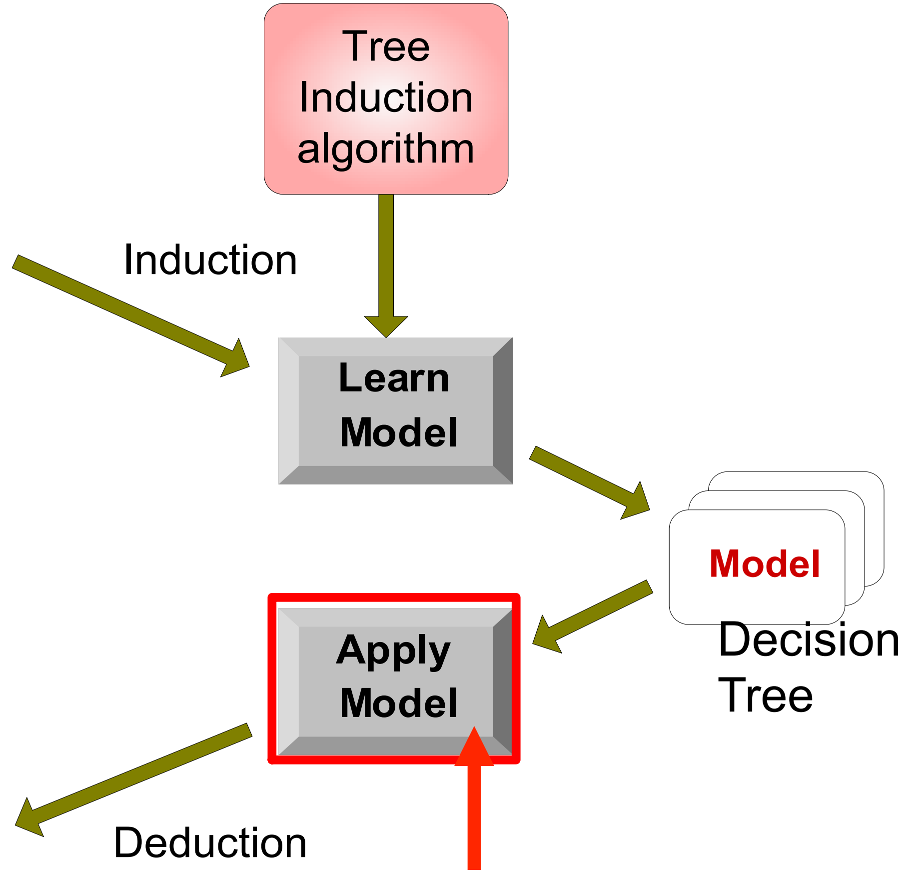


There could be more than one tree that fits the same data!

Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

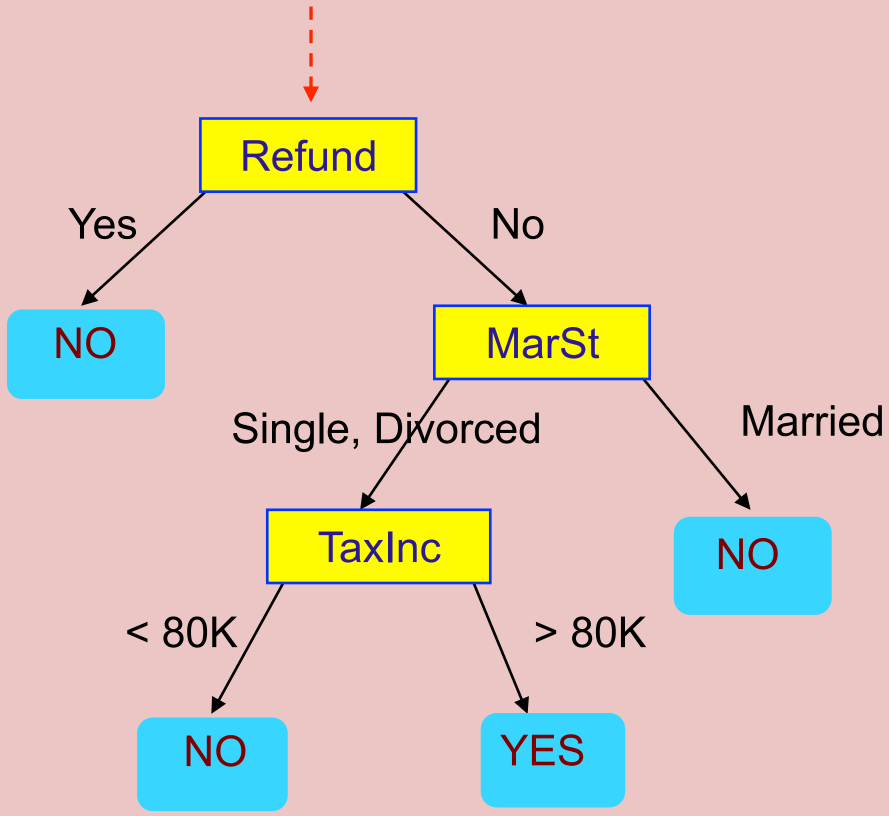


Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Apply Model to Test Data

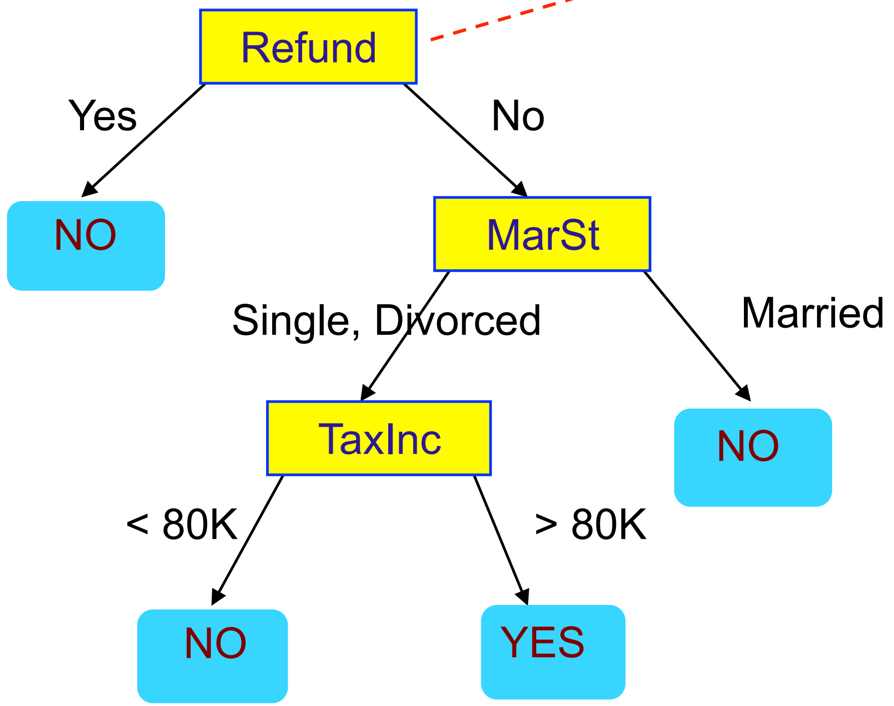
Start from the root of tree.



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

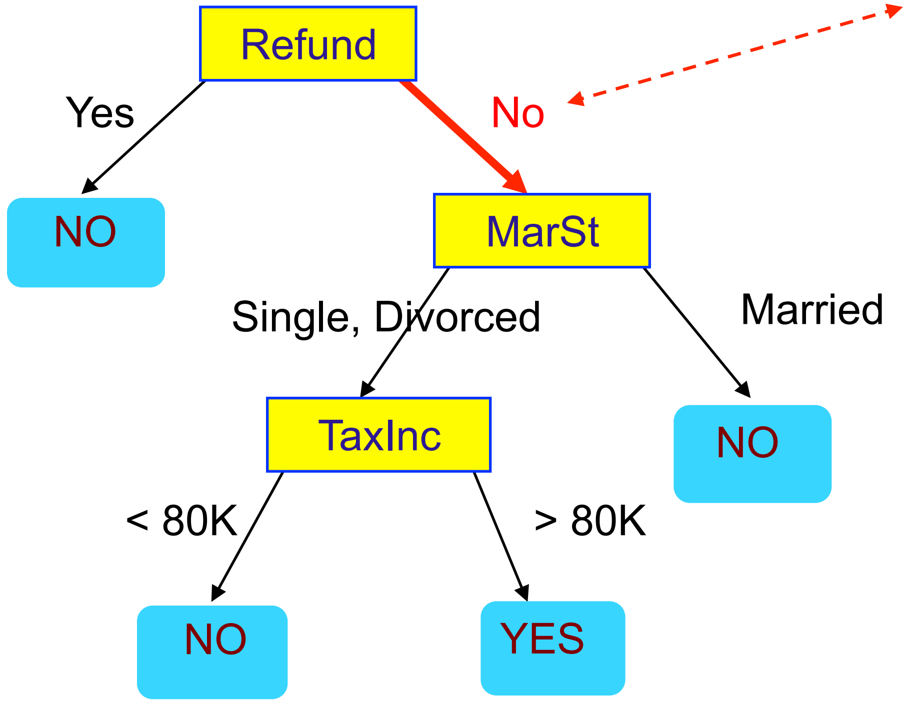


Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Source: Tan, Steinbach, Kumar

Apply Model to Test Data



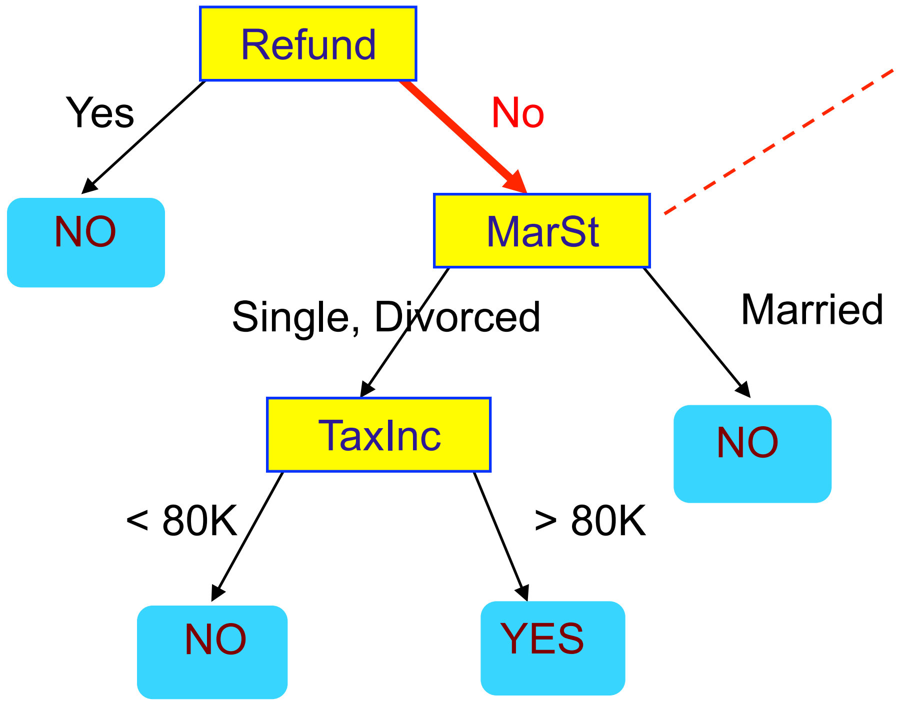
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

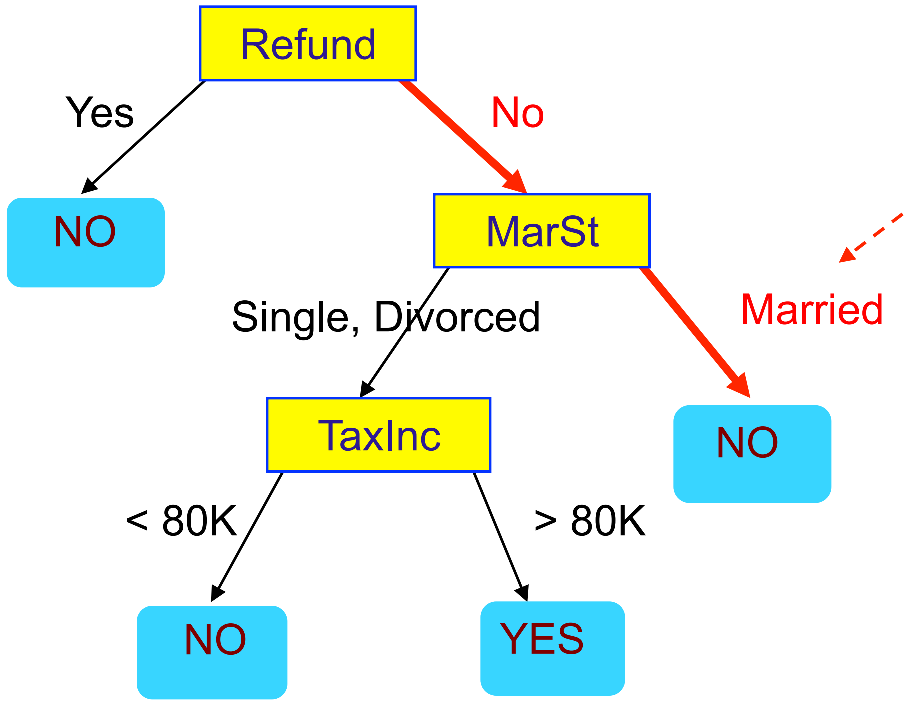


Source: Tan, Steinbach, Kumar

Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

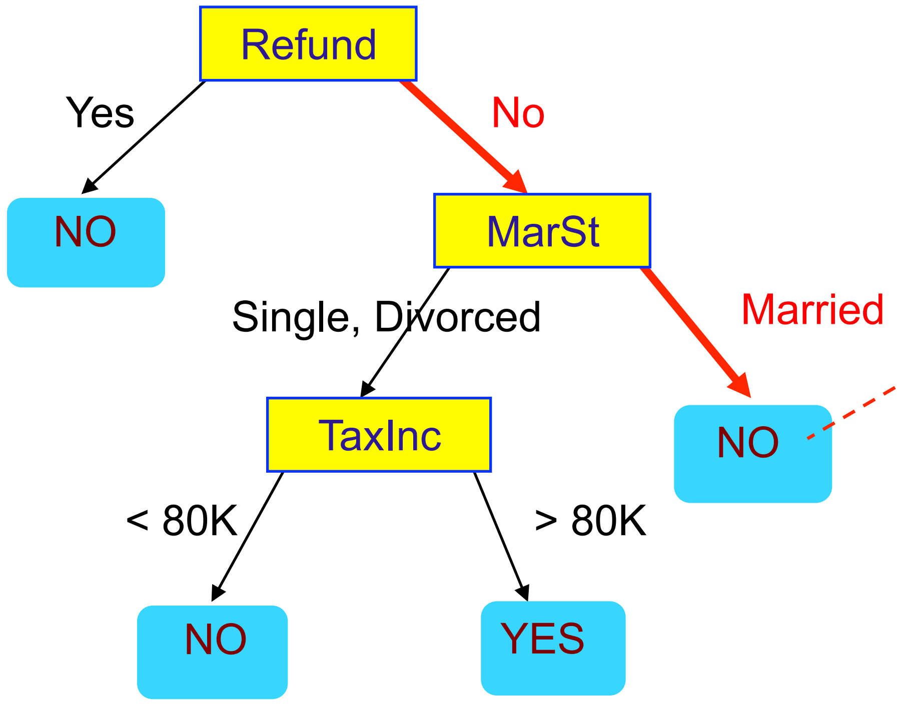


Source: Tan, Steinbach, Kumar

Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



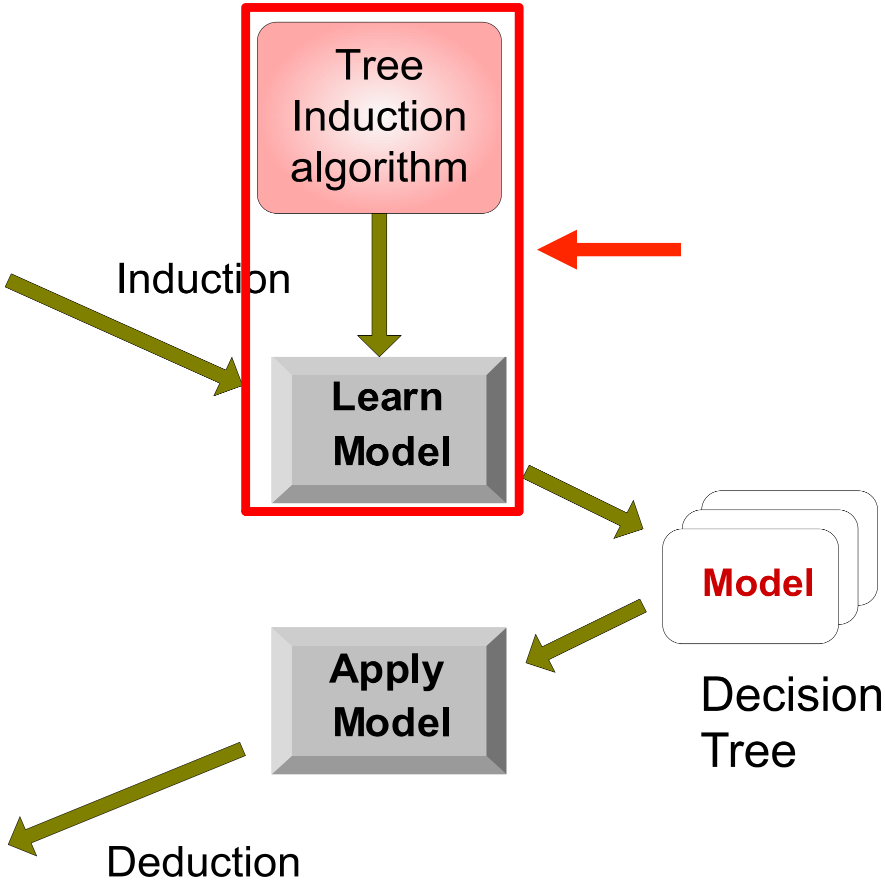
Assign Cheat to "No"

Source: Tan, Steinbach, Kumar

Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



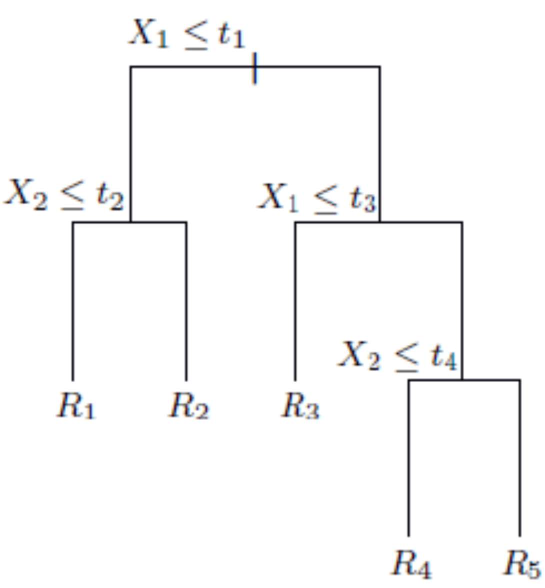
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

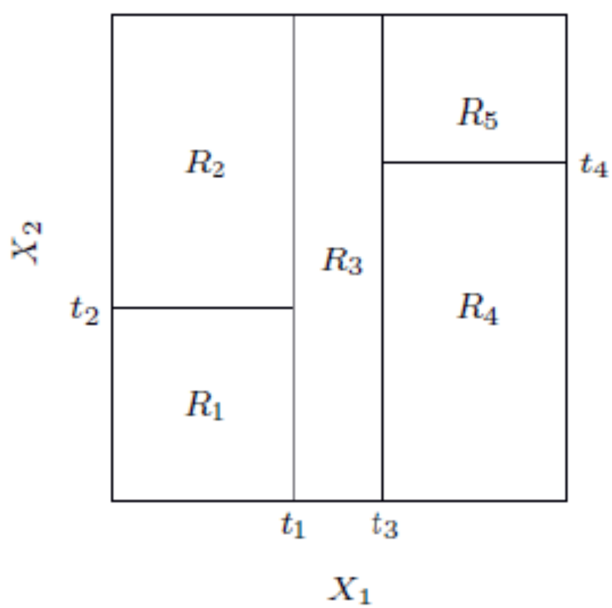
Test Set

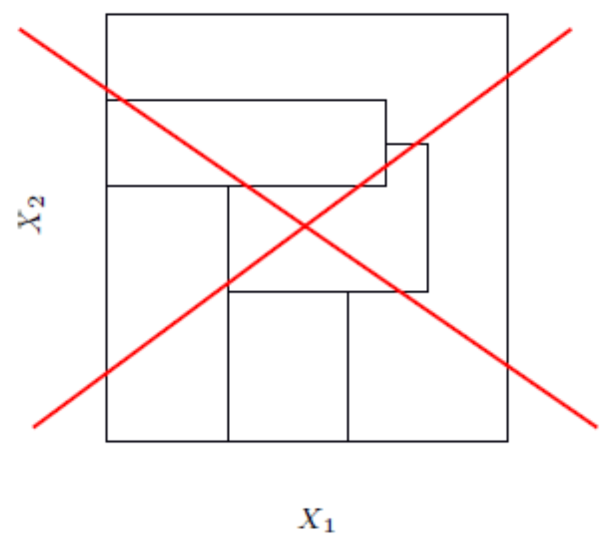
Restrict to recursive Partitioning by binary splits

- There many approaches
  - C 4.5 / C5 Algorithms, SPRINT

- Here top down splitting until no further split possible (or other criteria)







How to train a classification tree

– Starting with a single region -- i.e., all given data

– At the m-th iteration:

for each region R

for each attribute x_j in R

for each possible split s_j of x_j

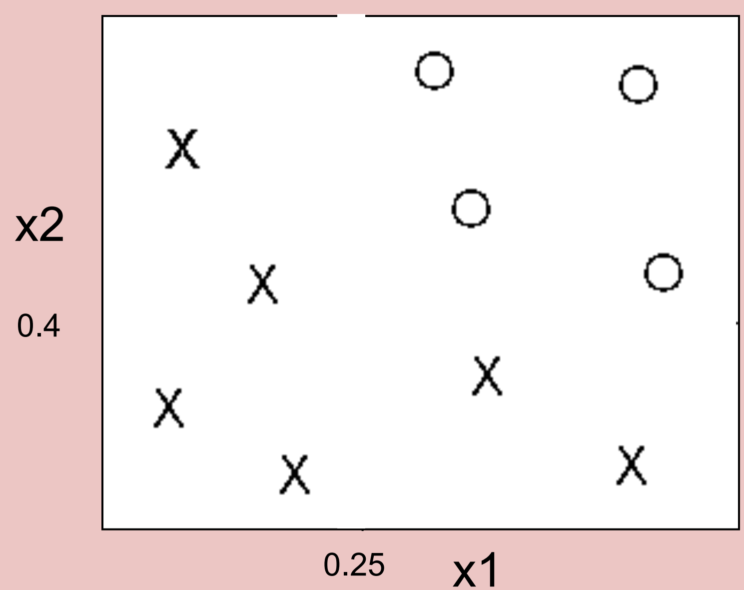
record change in <u>score</u> when we partition R into R^l and R^r

Choose $(x_j\ , s_j\ )$ giving maximum improvement to fit

Replace R with R^l; add R^r

Score: Node impurity (next slides)

- Draw 3 splits to separate the data.
- Draw the corresponding tree



Construction of a classification tree: Minimize the impurity in each node

Parent Node p is split into 2 partitions

Maximize Gain over possible splits:

$$GAIN_{split} = IMPURITY(p) - \left(\sum_{i=1}^{2} \frac{n_i}{n} IMPURITY(i) \right)$$

n_i is number of records in partition i

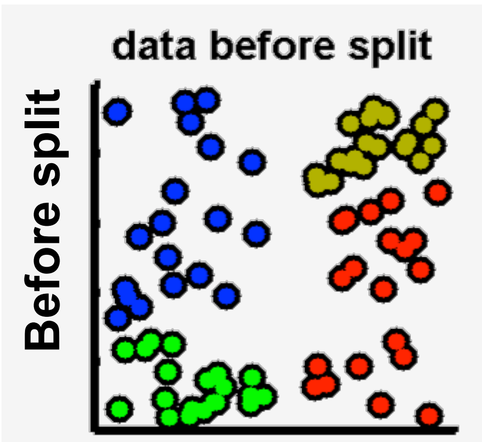
Possible Impurity Measures:

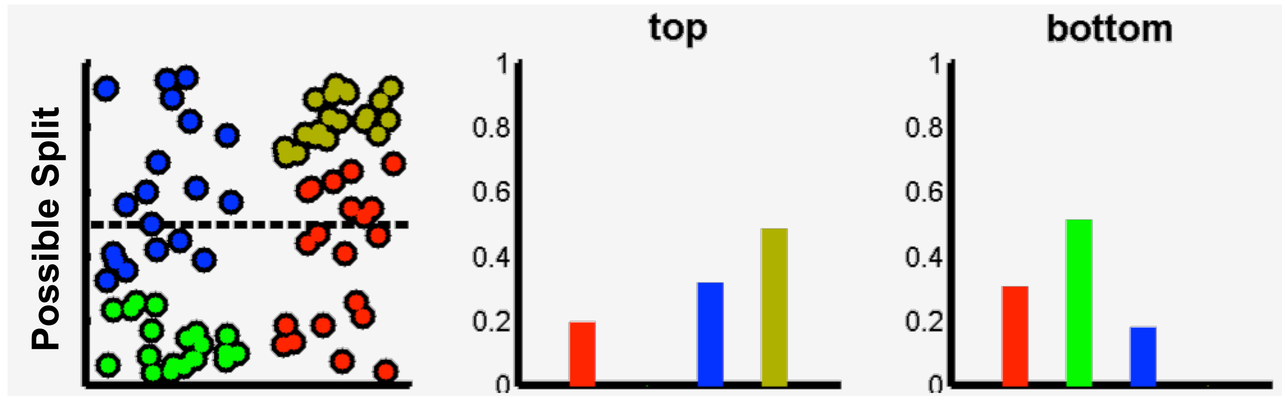
- Gini index
- entropy
- misclassification error

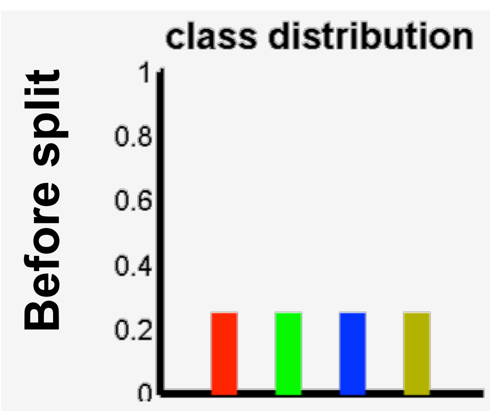
Source: Tan, Steinbach, Kumar

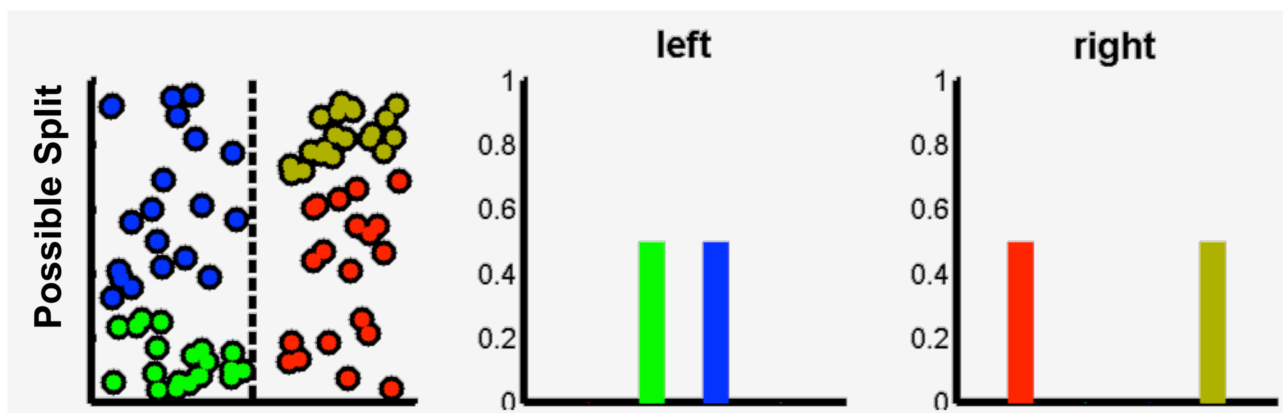
Construction of a classification tree: Minimize the impurity in each node

We have n_c=4 different classes: red, green, blue, olive









The three most common impurity measures

$p(j \mid t)$ is the relative frequency of class j at node t

n_c is the number of different classes

Gini Index:

$$GINI(t) = 1 - \sum_{j=1}^{n_c} [p(j \mid t)]^2$$

Entropy:

$$Entropy(t) = -\sum_{j=1}^{n_C} p(j \mid t) \log_2 \left(p(j \mid t) \right)$$

Classification error:

$$Classification - Error(t) = 1 - \max_{j \in \{1,\ldots,n_c)} p(j \mid t)$$

Computing the Gini Index for three 2-class examples

Gini Index at a given node t (in case of n_c=2 different classes):

$$GINI(t) = 1 - \sum_{j=1}^{2} [p(j \mid t)]^2 = 1 - \left(p(class.1 \mid t)^2 + p(class.2 \mid t)^2 \right)$$

Distribution of class 1 and class2 in node t:

C1	**0**
C2	**6**

$P(C1) = 0/6 = 0 \qquad P(C2) = 6/6 = 1$

$Gini = 1 – P(C1)^2 – P(C2)^2 = 1 – 0 – 1 = 0$

C1	**1**
C2	**5**

$P(C1) = 1/6 \qquad P(C2) = 5/6$

$Gini = 1 – (1/6)^2 – (5/6)^2 = 0.278$

C1	**2**
C2	**4**

$P(C1) = 2/6 \qquad P(C2) = 4/6$

$Gini = 1 – (2/6)^2 – (4/6)^2 = 0.444$

Source: Tan, Steinbach, Kumar

Computing the Entropy for three examples

Entropy at a given node t (in case of 2 different classes):

$$Entropy(t) = -\left(p(class.1) \cdot \log_2(\mathrm{p}(class.1)) + p(class.2) \cdot \log_2(\mathrm{p}(class.2)) \right)$$

C1	**0**
C2	**6**

p(C1) = 0/6 = 0 p(C2) = 6/6 = 1

Entropy = – 0 log(0) – 1 log(1) = – 0 – 0 = 0

C1	**1**
C2	**5**

p(C1) = 1/6 p(C2) = 5/6

Entropy = – (1/6) $\log_2$(1/6) – (5/6) $\log_2$(1/6) = 0.65

C1	**2**
C2	**4**

p(C1) = 2/6 p(C2) = 4/6

Entropy = – (2/6) $\log_2$(2/6) – (4/6) $\log_2$(4/6) = 0.92

Source: Tan, Steinbach, Kumar

Computing the miss-classification error for three examples

Classification error at a node t (in case of 2 different classes) :

$$Classification - Error(t) = 1 - \max\left(p(class.1)\ ,\ \mathrm{p(class.2)} \right)$$

C1	**0**
C2	**6**

p(C1) = 0/6 = 0   p(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0

C1	**1**
C2	**5**

p(C1) = 1/6   p(C2) = 5/6

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

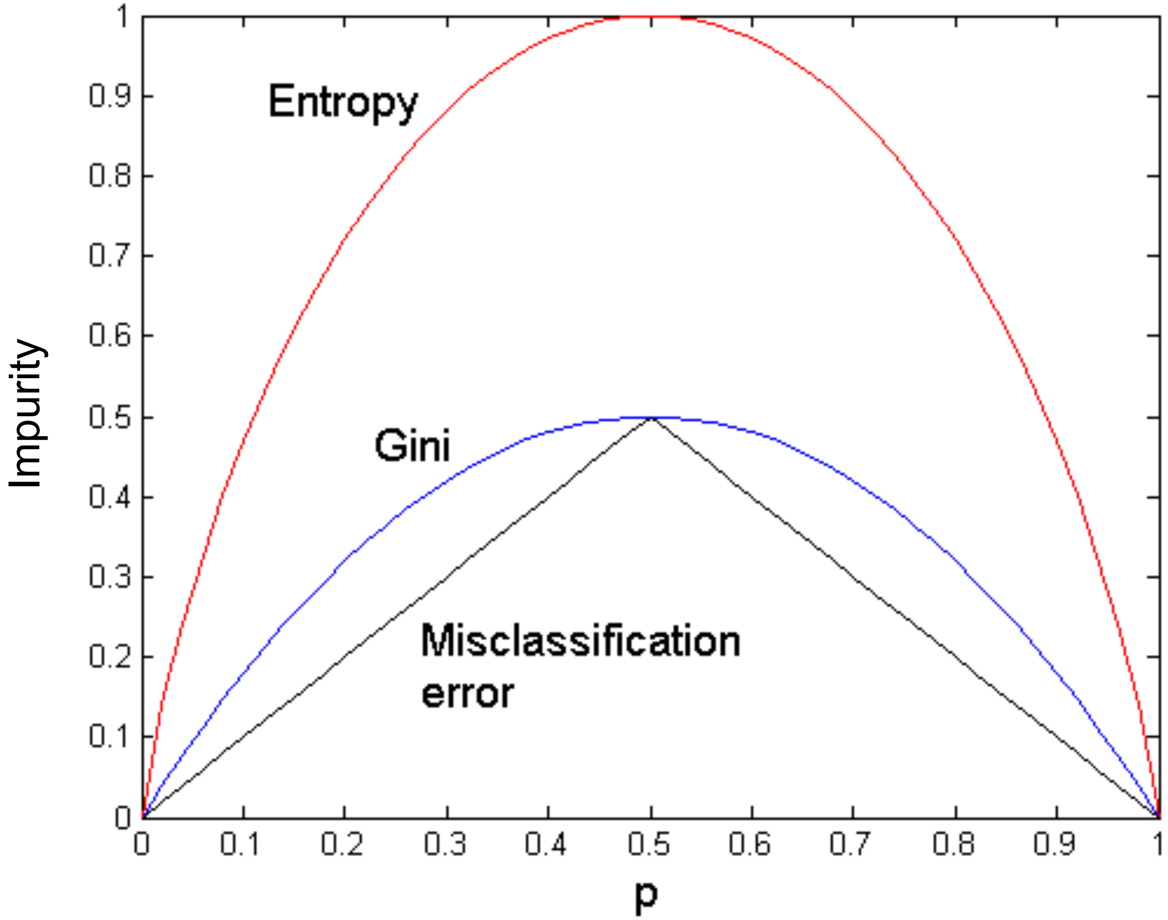
C1	**2**
C2	**4**

p(C1) = 2/6   p(C2) = 4/6

Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

Source: Tan, Steinbach, Kumar

Compare the three impurity measures for a 2-class problem



p: proportion of class 1

Source: Tan, Steinbach, Kumar

Using a tree for classification problems

– Starting with a single region -- i.e., all given data
– At the m-th iteration:

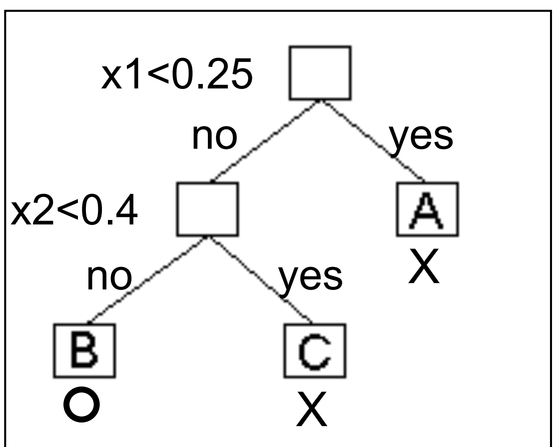
for each region R
  for each attribute x_j in R
    for each possible split s_j of x_j
      record change in score when we partition R into R^l and R^r

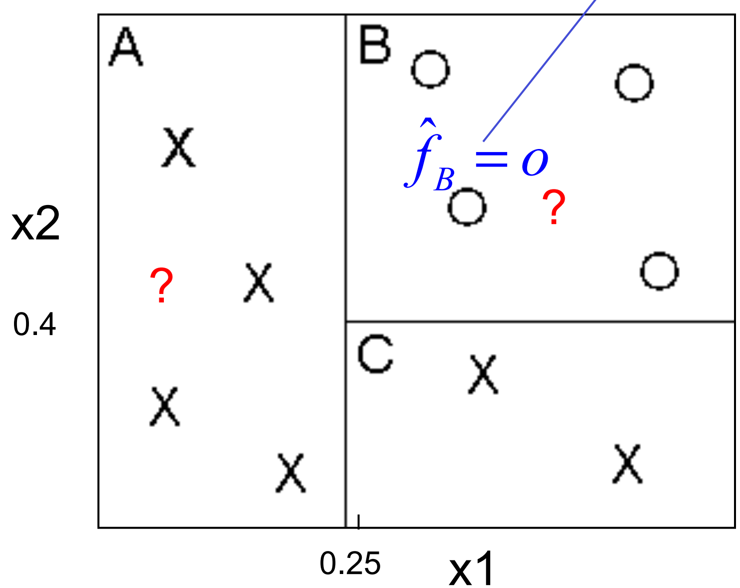
Choose (x_j, s_j) giving maximum improvement to fit

Replace R with R^l; add R^r

Score: Node impurity

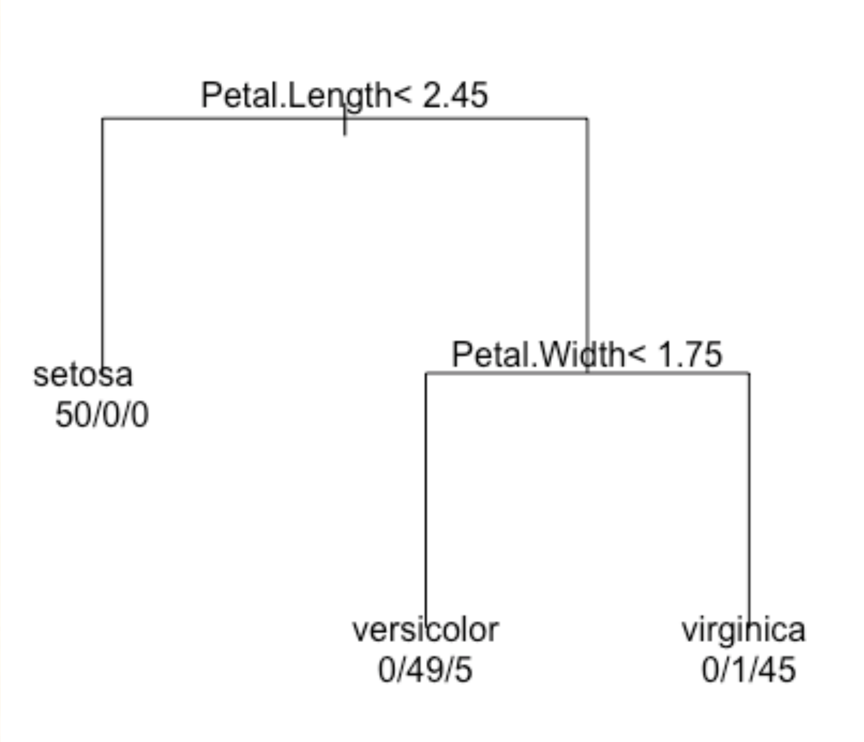
Model class label is given by the majority of all observation labels in the region



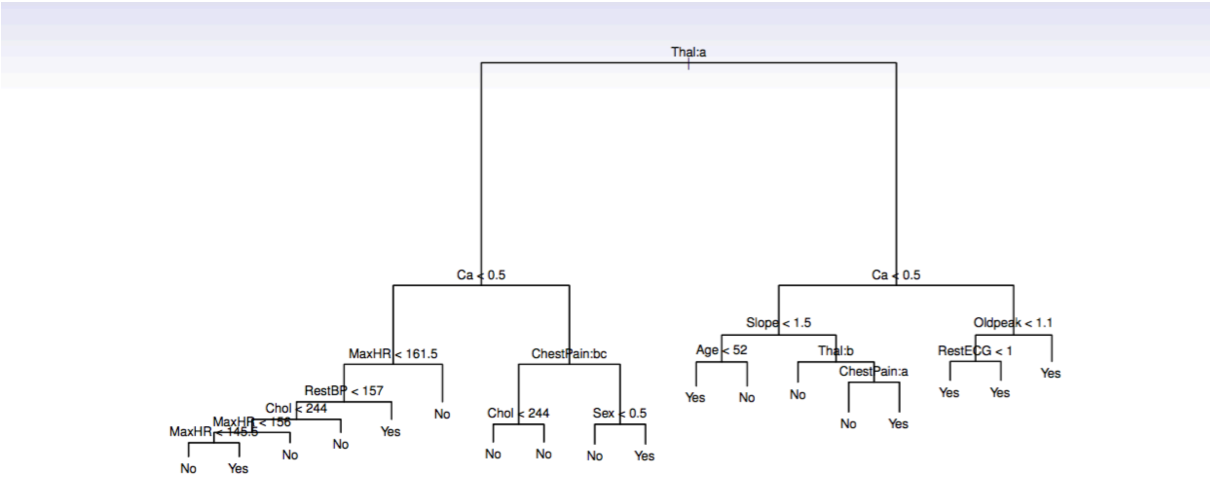


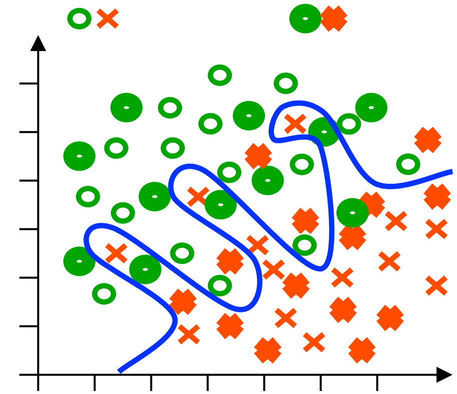
Decision Tree in R

```
# rpart for recursive partitioning

library(rpart)
d = rpart(Species ~., data=iris)
plot(d,margin=0.2, uniform=TRUE)
text(d, use.n = TRUE)
```

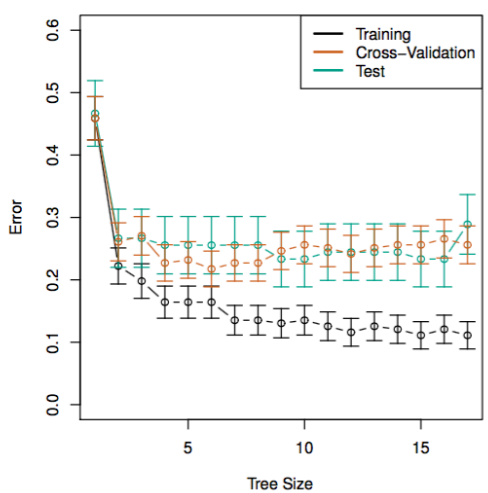


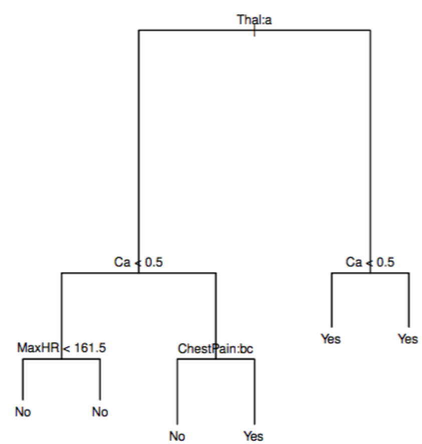
Overfitting





Overfitting





The larger the tree, the lower the error on the training set. “low bias”

However, training error can get worse. “high variance”

Taken from ISLR

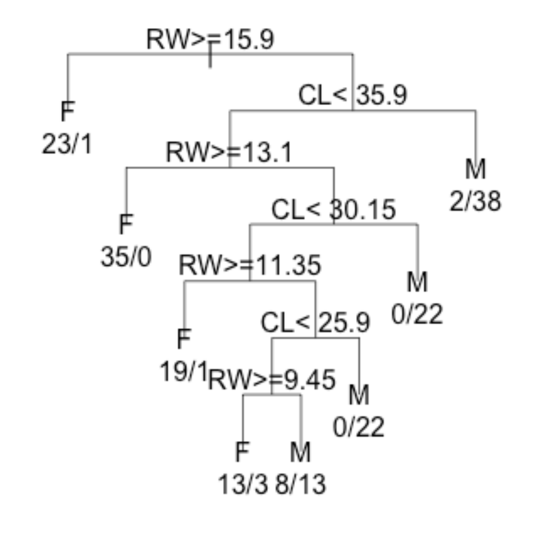
Pruning

- First Strategy: Stop growing the tree if impurity measure gain is small.
- To short-sighted: a seemingly worthless split early on in the tree might be followed by a very good split
- A better strategy is to grow a very large tree to the end, and then prune it back in order to obtain a subtree

- In rpart the pruning controlled with a complexity parameter cp
- cp is a parameter in regressions trees which can be optimized (like k in kNN)
- Later: Crossvalidation a way to systematically find the best meta parameter

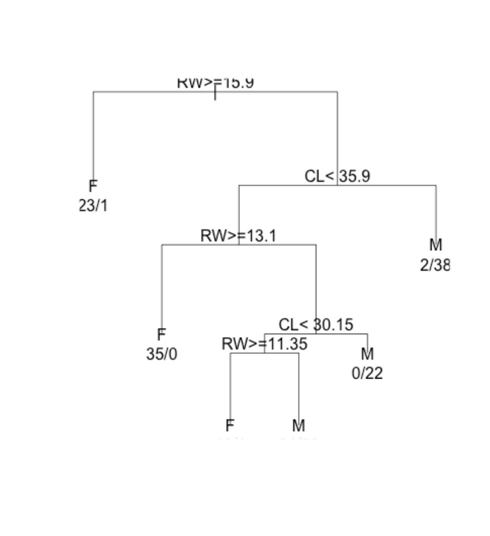
Pruning

Original Tree

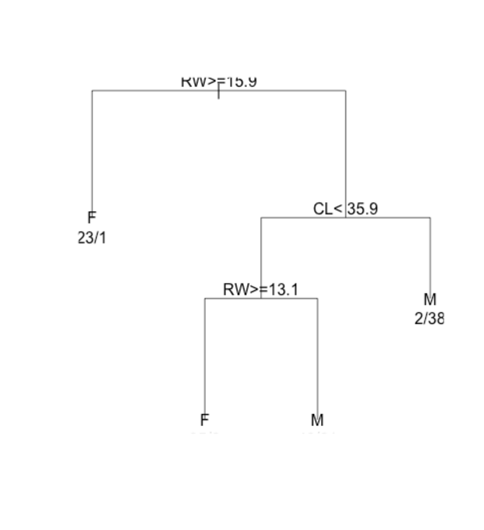



```
t1 <- rpart(sex ~., data=crabs)
plot(t1, margin=0.2, uniform=TRUE);text(t1,use.n=TRUE)

t2 = prune(t1, cp=0.05)
plot(t2);text(t2,use.n=TRUE)

t3 = prune(t1, cp=0.1)
plot(t3);text(t3,use.n=TRUE)


t4 = prune(t1, cp=0.2)
plot(t4);text(t4,use.n=TRUE)
```

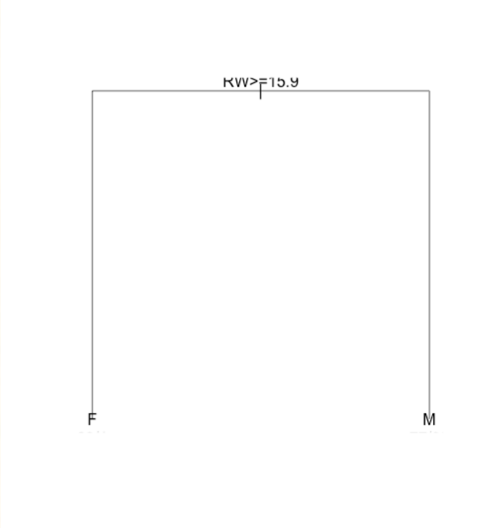
cp = 0.05



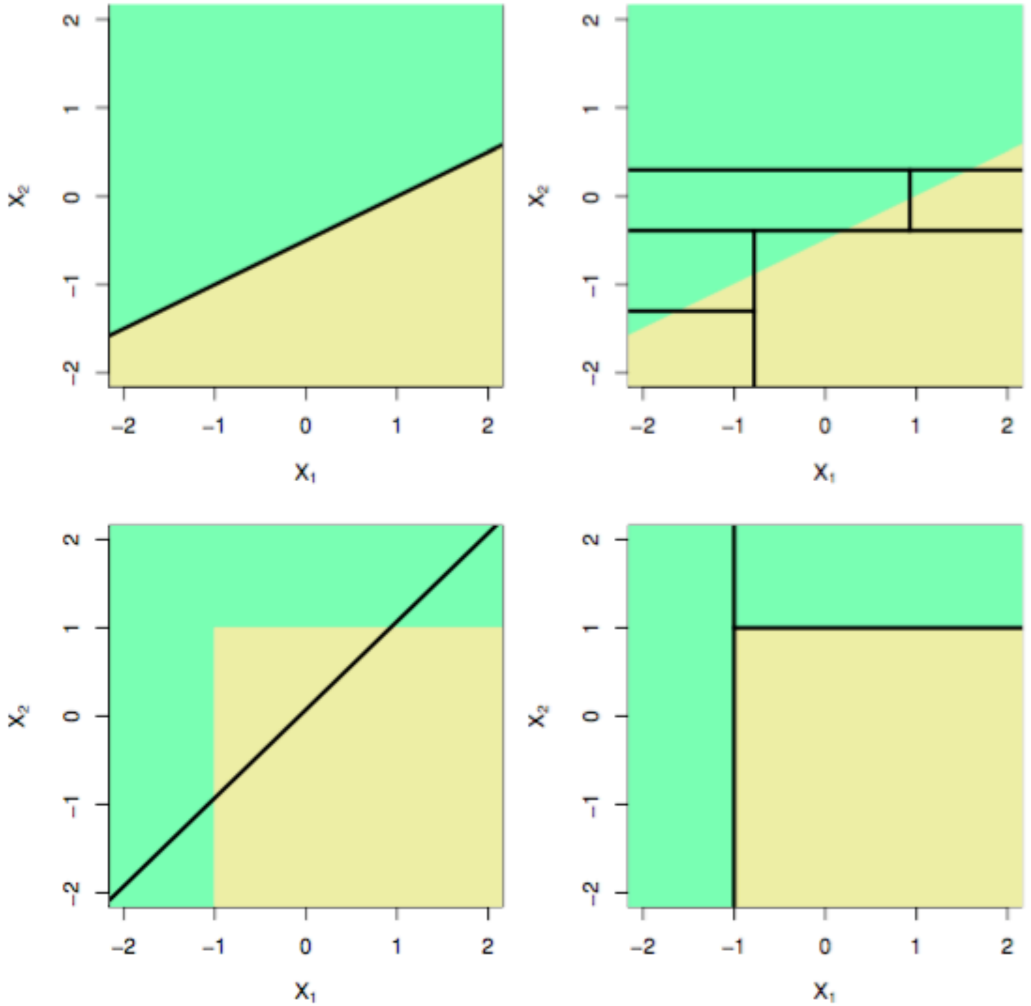
cp = 0.1



cp = 0.2



Trees vs Linear Models



Slide Credit: ISLR

Pros and cons of tree models

- Pros
  - Interpretability
  - Robust to outliers in the input variables
  - Can capture non-linear structures
  - Can capture local interactions very well
  - Low bias if appropriate input variables are available and tree has sufficient depth.
- Cons
  - High variation (instability of trees)
  - Tend to overfit
  - Needs big datasets to capture additive structures
  - Inefficient for capturing linear structures

Joining Forces

Which supervised learning method is best?

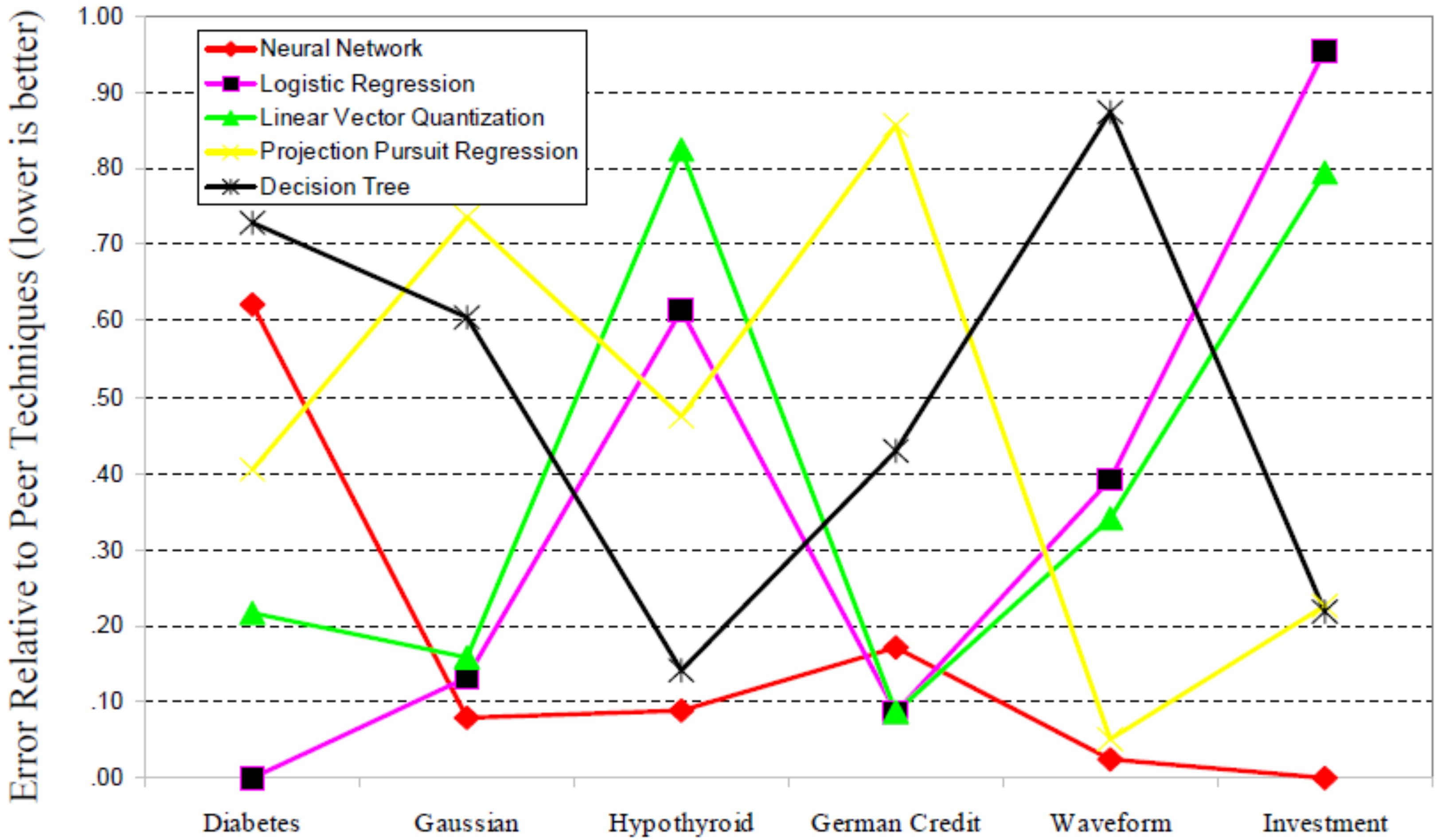


Bild: Seni & Elder

Bundling improves performance

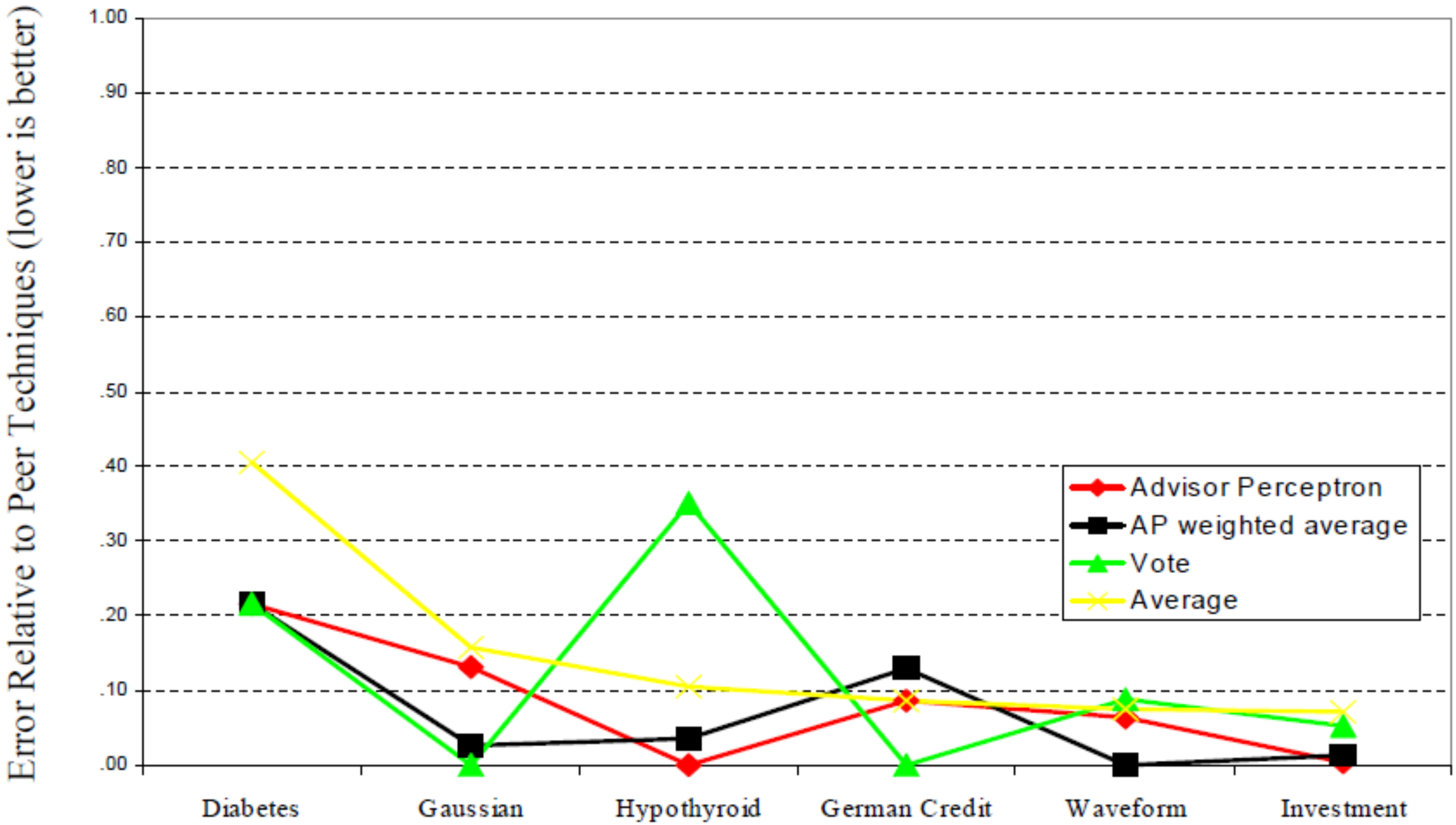
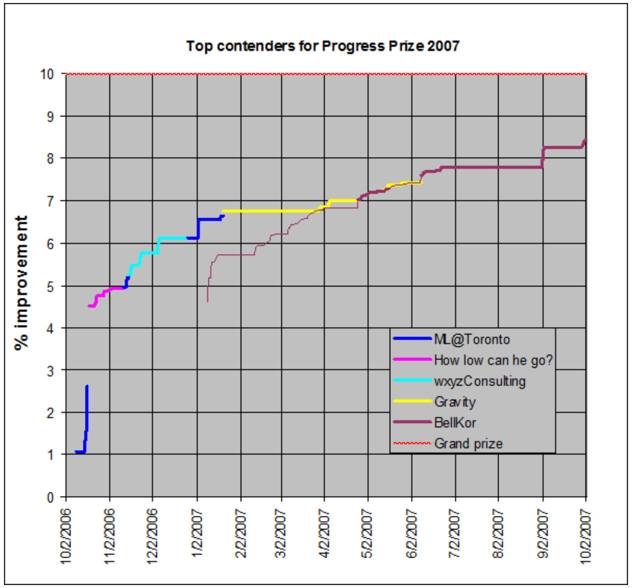


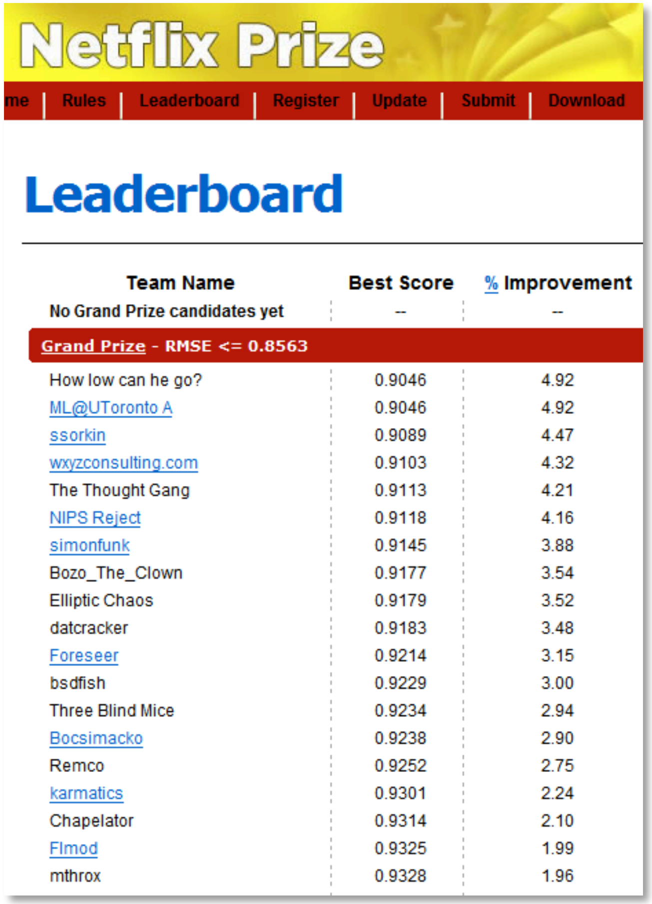
Bild: Seni & Elder

Evolving results I
Low hanging fruits and slowed down progress

- **After 3 weeks**, at least **40 teams** had **improved** the Netflix classifier
- Top teams showed about 5% improvement
- However, **improvement slowed**:



from http://www.research.att.com/~volinsky/netflix/



Netflix Prize

me | Rules | Leaderboard | Register | Update | Submit | Download

Leaderboard

Team Name	Best Score	% Improvement
No Grand Prize candidates yet	--	--
Grand Prize - RMSE <= 0.8563		
How low can he go?	0.9046	4.92
ML@UToronto A	0.9046	4.92
ssorkin	0.9089	4.47
wxyzconsulting.com	0.9103	4.32
The Thought Gang	0.9113	4.21
NIPS Reject	0.9118	4.16
simonfunk	0.9145	3.88
Bozo_The_Clown	0.9177	3.54
Elliptic Chaos	0.9179	3.52
datcracker	0.9183	3.48
Foreseer	0.9214	3.15
bsdfish	0.9229	3.00
Three Blind Mice	0.9234	2.94
Bocsimacko	0.9238	2.90
Remco	0.9252	2.75
karmatics	0.9301	2.24
Chapelator	0.9314	2.10
Flmod	0.9325	1.99
mthrox	0.9328	1.96

Details: Gravity

- Quote
- Table 5: Best results of single approaches and their combinations

Method/Combination	RMSE
MF	0.9190
NB	0.9313
CL	0.9606
NB + CL	0.9275
MF + CL	0.9137
MF + NB	0.9089
MF + NB + CL	0.9089

[home.mit.bme.hu/~gtakacs/download/gravity.pdf]

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

Details: When Gravity and Dinosaurs Unite

- Quote
- *"Our common team **blends the result** of team Gravity and team Dinosaur Planet."*

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

Details: BellKor / KorBell

- Quote
- *"Our final solution (RMSE=0.8712) consists of **blending 107 individual results**."*

--	No Progress Prize candidates yet	--	--
	Progress Prize - RMSE <= 0.8625		
1	BellKor	0.8705	8.50
	Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell		
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

Evolving results III
Final results

- The winner was an **ensemble of ensembles** (including BellKor)
- Gradient boosted decision trees [http://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf]

Leaderboard

Showing Test Score. Click here to show quiz score

Display top 20 leaders.

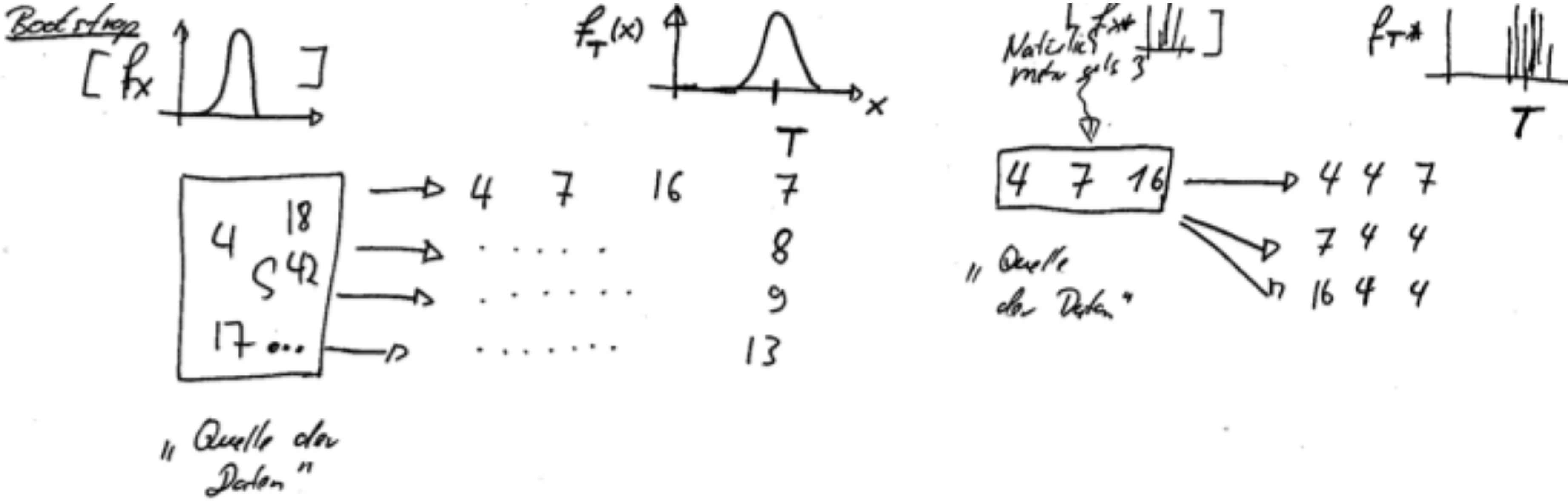
Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace_	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11
Progress Prize 2008 - RMSE = 0.8627 - Winning Team: BellKor in BigChaos				

Overview of Ensemble Methods

- Many instances of the same classifier
  - Bagging (bootstrapping & aggregating)
    - Create “new” data using bootstrap
    - Train classifiers on new data
    - Average over all predictions (e.g. take majority vote)
  - Bagged Trees
    - Use bagging with decision trees
  - Random Forest
    - Bagged trees with special trick
  - Boosting
    - An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records.
- Combining classifiers
  - Weighted averaging over predictions
  - Stacking classifiers
    - Use output of classifiers as input for a new classifier

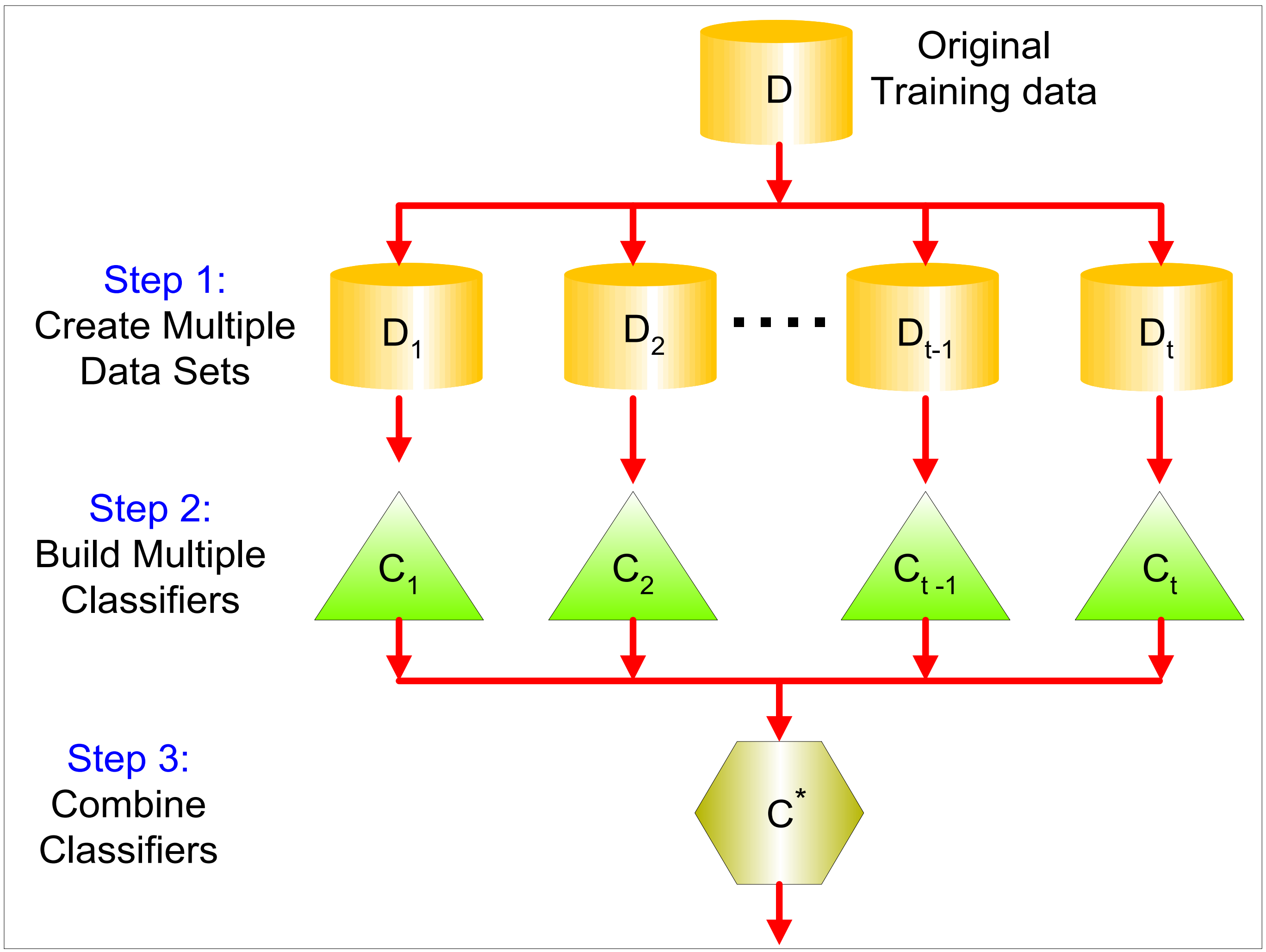
Bagging

Idee des Bootstraps



Idee: Die Sampling Verteilung liegt nahegenug an der 'wahren'

Bagging: Bootstrap Aggregating



Bootstrap

Aggregating:
mean, majority vote

Source: Tan, Steinbach, Kumar

Why does it work?

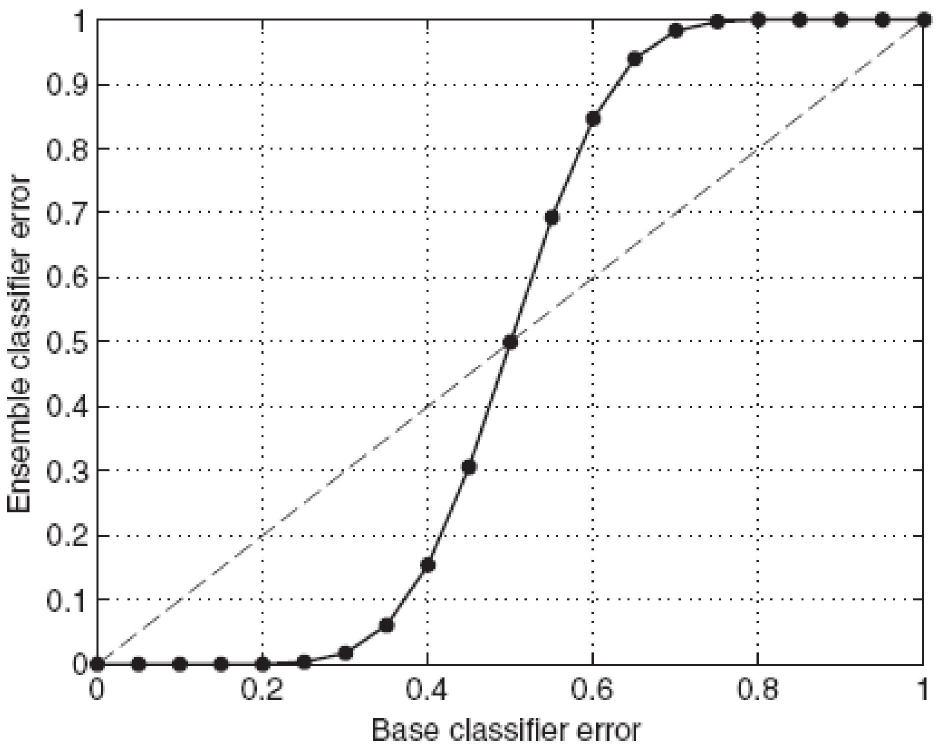
- Suppose there are 25 base classifiers
- Each classifier has error rate, $\varepsilon = 0.35$
- Assume classifiers are independent (that's the hardest assumption)
- Take **majority vote**
- Majority voter is wrong if 13 or more are wrong
- Number of wrong predictors $X \sim \text{Bin}(\text{size}=25, p_0=0.35)$

```
•> 1 - pbinom(12, size=25, prob = 0.35)
•[1] 0.06044491
```

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

Why does it work?

- 25 Base Classifiers



Ensembles are only better than one classifier, if each classifier is better than random guessing!