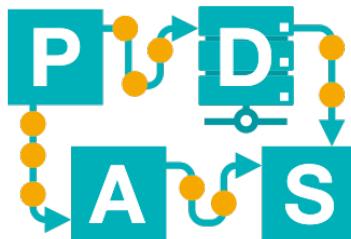


Decision Trees

Lecture 3

IDS-L3

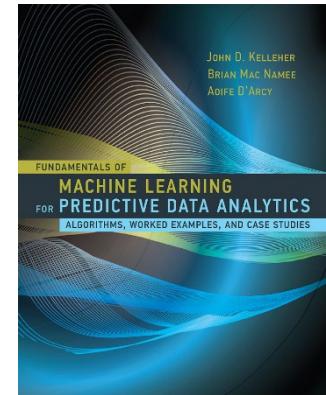


Chair of Process
and Data Science

RWTH AACHEN
UNIVERSITY

Outline of Today's Lecture

- Let the data speak
- Decision trees
- Entropy
- ID3 algorithm
- Variations of the ID3 algorithm
- Dealing with continuous variables



Based on Chapter 4 of
Fundamentals of Machine
Learning for Predictive Data
Analytics by J. Kelleher, B. Mac
Namee and A. D'Arcy.

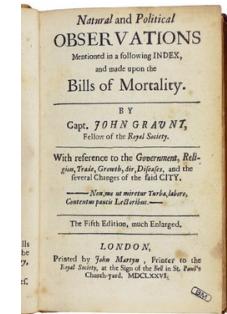
Let the data speak



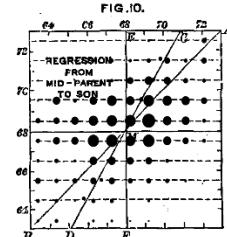
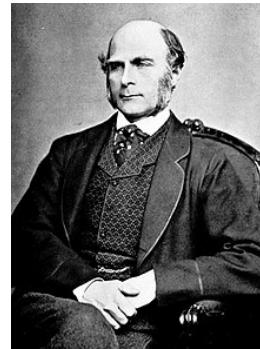
Statistics versus DM / ML (1/2)

- Statistics have been around for a while ...

John Graunt (1620-1674) studied London's death records around 1660. Based on this he was able to predict the life expectancy of a person at a particular age.



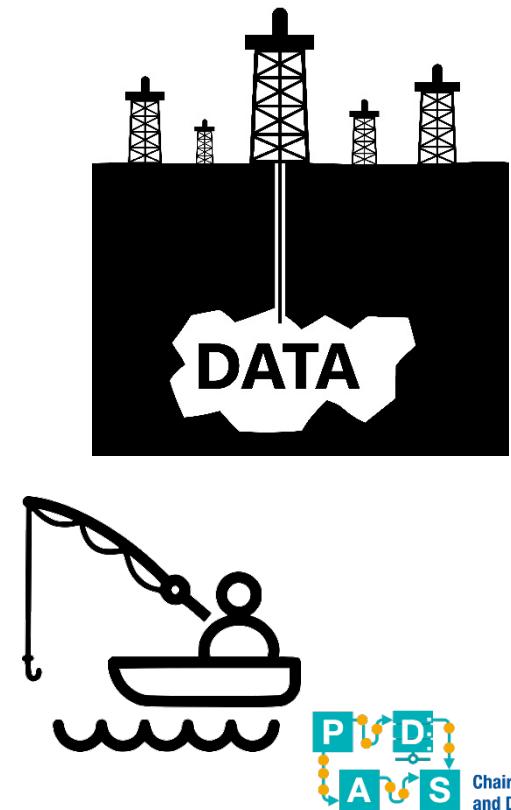
Francis Galton (1822-1911) introduced statistical concepts like regression and correlation at the end of the 19th century.



Chair of Process
and Data Science

Statistics versus DM / ML (2/2)

- Statistics have been around for a while ...
- However, progress in data science was fueled by more pragmatic approaches (handling large amounts of data), not statistics.
- Major breakthroughs in the discovery of patterns and relationships (e.g., efficiently learning decision trees and association rules), were described as “data fishing”, “data snooping”, and “data dredging” by traditional statisticians.
- Now some statisticians claim “owning” the data science field.



Interesting reads

Statistical Science
16(2), Vol. 16, No. 3, 199-231

Statistical Modeling: The Two Cultures

Leo Breiman

Abstract. There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats data as deterministic. The statistician's commitment to one has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statistics from modeling a large range of interesting scientific problems. Algorithmic modeling, however, in recent years, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from an exclusive dependence on data models and adopt a more diverse set of tools.

1. INTRODUCTION

Statistics starts with data. Think of the data as being generated by a black box in which a vector of input variables x and a response variable y come out, and on the other side there the response y come out. Inside the black box, nature functions to associate the predictor variables with the response variables, so the picture is like this:



There are two goals in analyzing the data:

Prediction. To be able to predict what the responses are going to be to future input variables;

Information. To extract some information about how nature is associating the response variables to the input variables.

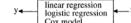
There are two different approaches toward these goals:

The Data Modeling Culture

The analysis in this culture starts with assuming a stochastic data model inside of the black box. For example, a common data model is that data are generated by independent draws from a response variable $y = f(x)$ + predictor variables, random noise, parameters)

Leo Breiman is Professor, Department of Statistics, University of California, Berkeley, California 94720-4735 (e-mail: leo@stat.berkeley.edu).

The values of the parameters are estimated from the data and the model then used for information and/or prediction. Thus the black box is filled in like this:



Model validation. Yes—no using goodness-of-fit tests and residual examination.

Estimated culture population. 98% of all statisticians.

The Algorithmic Modeling Culture

The analysis in this culture considers the inside of the box complex and unknown. Their approach is to find a function $f(x)$ —an algorithm that operates on x to predict the response y . Their black box looks like this:



Model validation. Measured by predictive accuracy.
Estimated culture population. 2% of statisticians, many in other fields.

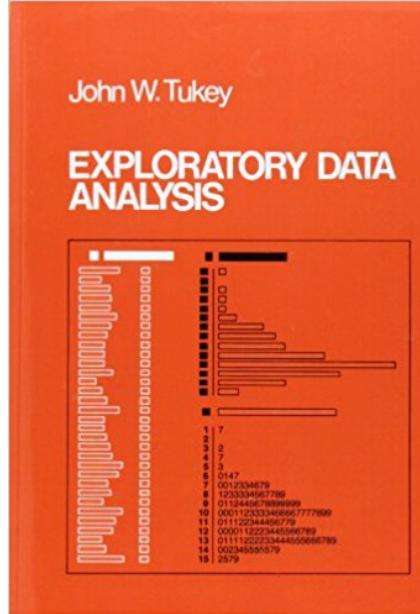
In this paper I will argue that the focus in the statistical community on data models has:

- Led to irrelevant theory and questionable scientific conclusions;

199

Statistical Modeling: The Two Cultures Statist. Sci. 16 (2001), no. 3, 199–231. doi:10.1214/ss/1009213726.

Leo Breiman (1928-2005)



Exploratory Data Analysis. Addison-Wesley Publishing Company, 1977

John W. Tukey (1915-2000)

JELLY BEANS
CAUSE ACNE!

SCIENTISTS!
INVESTIGATE!

BUT WE'RE
PLAYING
MINECRAFT!
... FINE.



WE FOUND NO
LINK BETWEEN
JELLY BEANS AND
ACNE ($P > 0.05$).



THAT SETTLES THAT.

I HEAR IT'S ONLY
A CERTAIN COLOR
THAT CAUSES IT.

SCIENTISTS!

BUT
MINECRAFT!



Statistics:
Don't torture
the data until
it confesses

WE FOUND NO
LINK BETWEEN
PURPLE JELLY
BEANS AND ACNE
($P > 0.05$).

WE FOUND NO
LINK BETWEEN
BROWN JELLY
BEANS AND ACNE
($P > 0.05$).

WE FOUND NO
LINK BETWEEN
PINK JELLY
BEANS AND ACNE
($P > 0.05$).

WE FOUND NO
LINK BETWEEN
BLUE JELLY
BEANS AND ACNE
($P > 0.05$).

WE FOUND NO
LINK BETWEEN
TEAL JELLY
BEANS AND ACNE
($P > 0.05$).

WE FOUND NO
LINK BETWEEN
PURPLE JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
BROWN JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
PINK JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
BLUE JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
TEAL JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
SALMON JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
RED JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
TURQUOISE JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
MAGENTA JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
YELLOW JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
GREY JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
TAN JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
CYAN JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND A
LINK BETWEEN
GREEN JELLY
BEANS AND ACNE
($P < 0.05$).



WE FOUND NO
LINK BETWEEN
YELLOW JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
BEIGE JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
LILAC JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
BLACK JELLY
BEANS AND ACNE
($P > 0.05$).

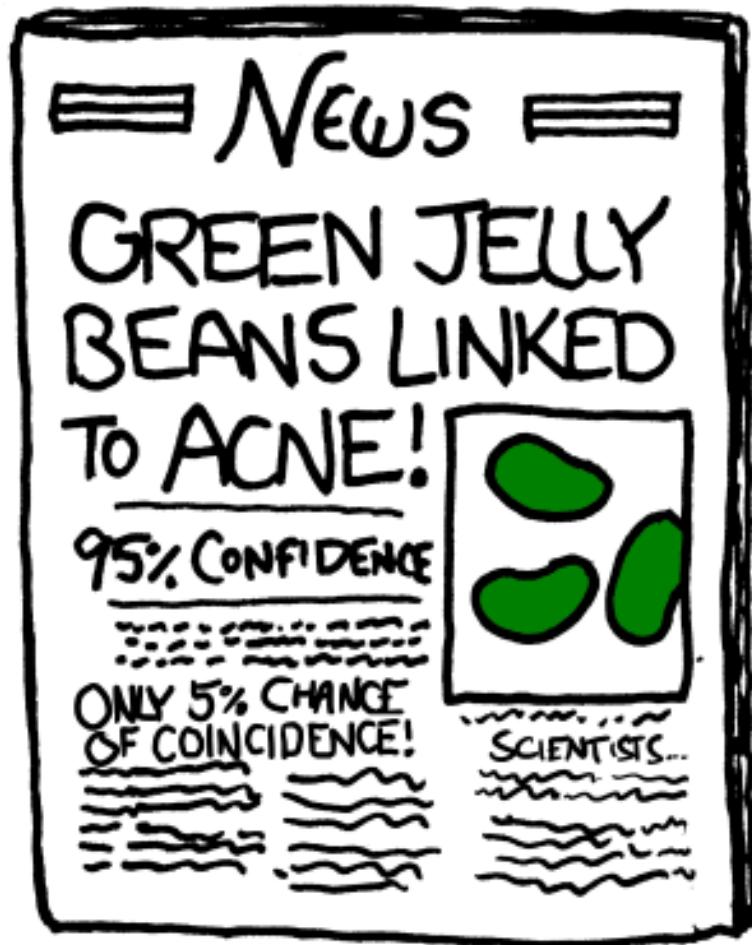


WE FOUND NO
LINK BETWEEN
PEACH JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
ORANGE JELLY
BEANS AND ACNE
($P > 0.05$).





Risks

- Testing many hypotheses.
- Overfitting the data.
- Underfitting the data.
- Bias in data.
- Bias in representation.

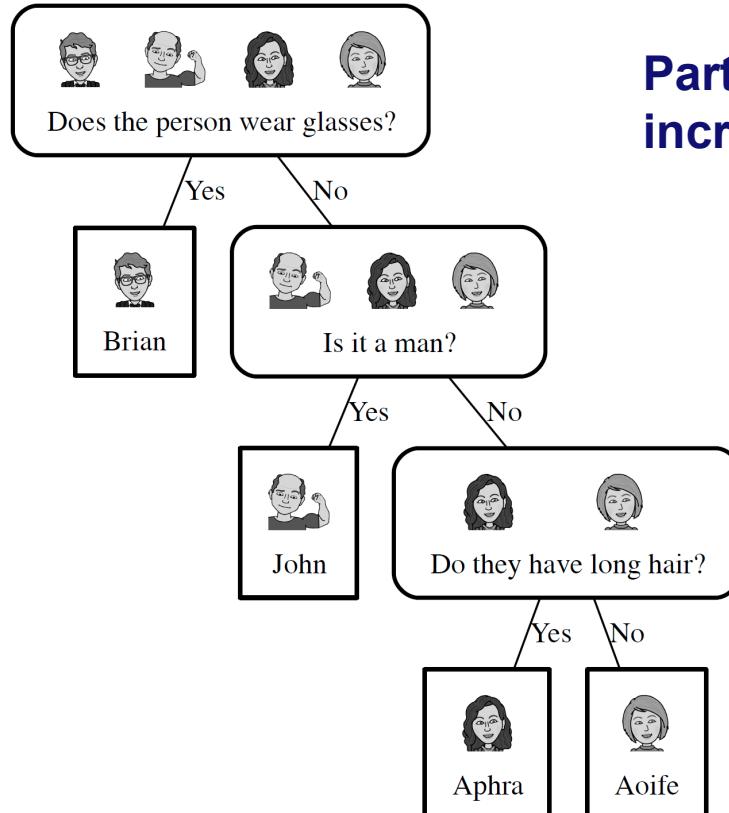
Careful, but pragmatic ...

- The data is there: *Let it speak!*
- Data will always be dirty, biased, etc.
- Simply summarizing the data can be surprisingly useful.
- Shift from (a) “N = small” or “N = sample” to (b) “N = Big” or “N = all” changes the perspective.
- We have the computing power, storage, and tools to do so.

Decision trees



Grouping instances in decision trees



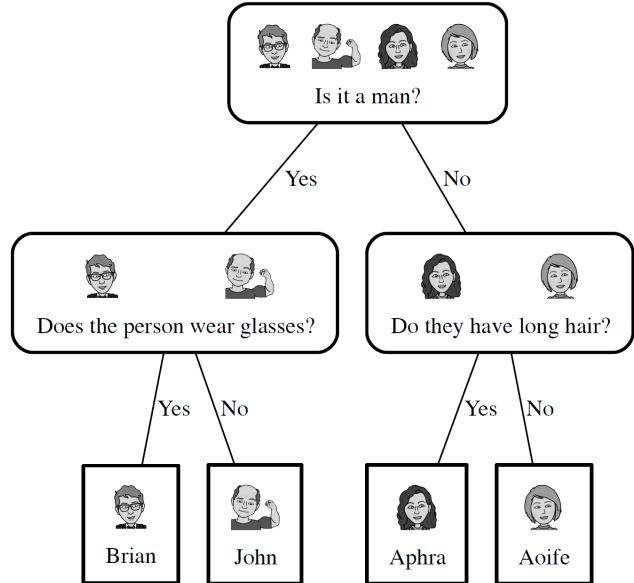
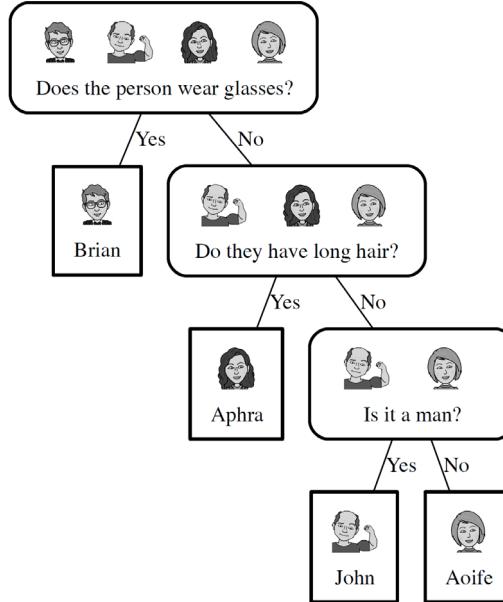
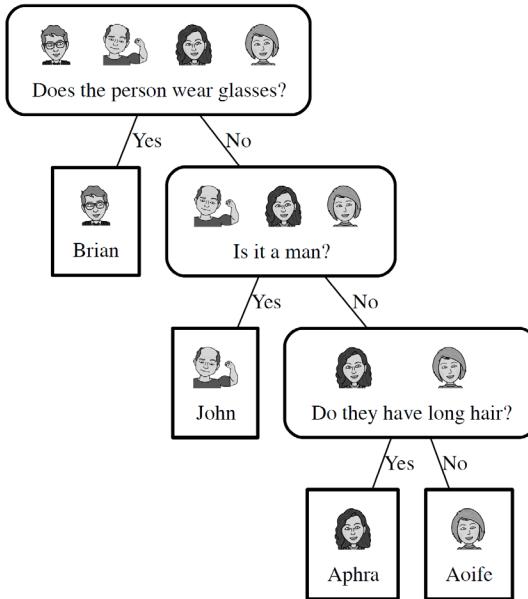
Partitioning instances in
increasingly smaller groups

Trees taken from **Fundamentals of Machine Learning for Predictive Data Analytics** by J. Kelleher, B. Mac Namee and A. D'Arcy.



Chair of Process
and Data Science

Grouping instances in decision trees



- Different trees possible.
- Two goals: (1) the tree is small and simple and (2) the leaves are homogeneous in terms of the target feature.

Supervised learning

features

instances

f1	f2	f3	f4	...	fn	class
						high
						high
						low
						medium
						high
						low

descriptive
features

target feature

Decision tree
aims to explain
the target feature
in terms of the
descriptive
features

Example: Life expectancy

target feature

drinker	smoker	weight	age
yes	yes	120	44
no	no	70	96
yes	no	72	88
yes	yes	55	52
no	yes	94	56
no	no	62	82
...	

descriptive
features

Class label:

- ≥ 70 = old
- < 70 = young



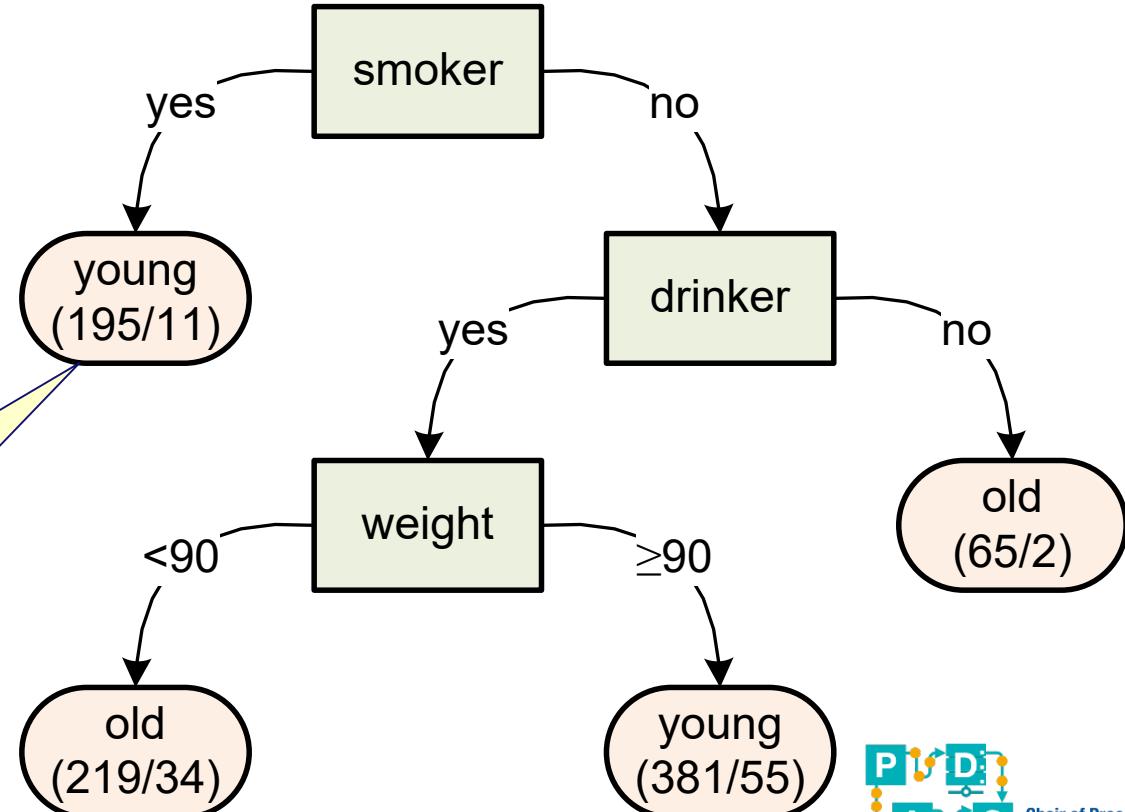
Chair of Process
and Data Science

Example: Life expectancy

drinker	smoker	weight	age
yes	yes	120	44
no	no	70	96
yes	no	72	88
yes	yes	55	52
no	yes	94	56
no	no	62	93
...

$\geq 70 = \text{old}$
 $< 70 = \text{young}$

all 195 smokers were classified as "young" but 11 were classified incorrectly



Example: Study results

target feature

linear algebra	logic	program-ming	operations research	workflow systems	...	duration	result
9	8	8	9	9	...	36	cum laude
7	6	-	8	8	...	42	passed
-	-	5	4	6	...	54	failed
8	6	6	6	5	...	38	passed
6	7	6	-	8	...	39	passed
9	9	9	9	8	...	29	cum laude
5	5	-	6	6	...		
			

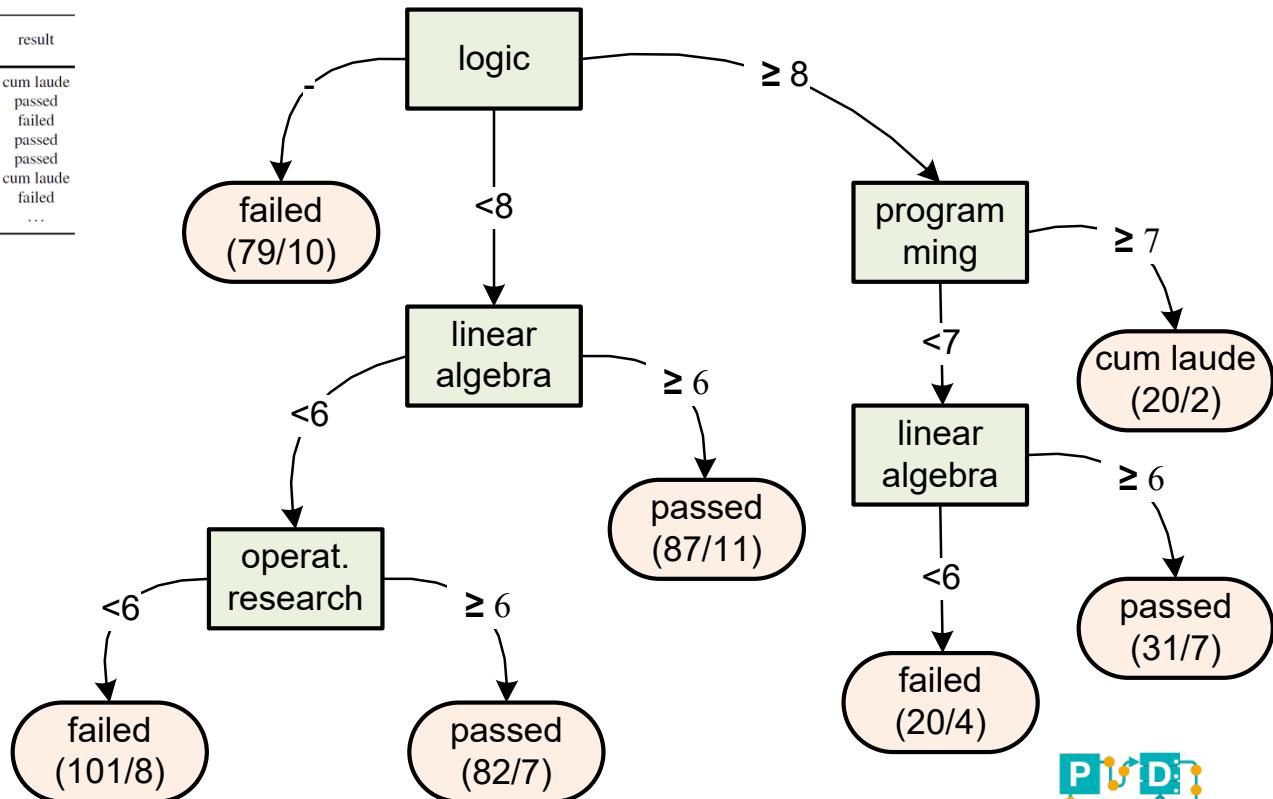
descriptive features

Class label:

- cum laude
- passed
- failed

Example: Study results

linear algebra	logic	program-ming	operations research	workflow systems	...	duration	result
9	8	8	9	9	...	36	cum laude
7	6	-	8	8	...	42	passed
-	-	5	4	6	...	54	failed
8	6	6	6	5	...	38	passed
6	7	6	-	8	...	39	passed
9	9	9	9	8	...	38	cum laude
5	5	-	6	6	...	52	failed
...



Decision tree

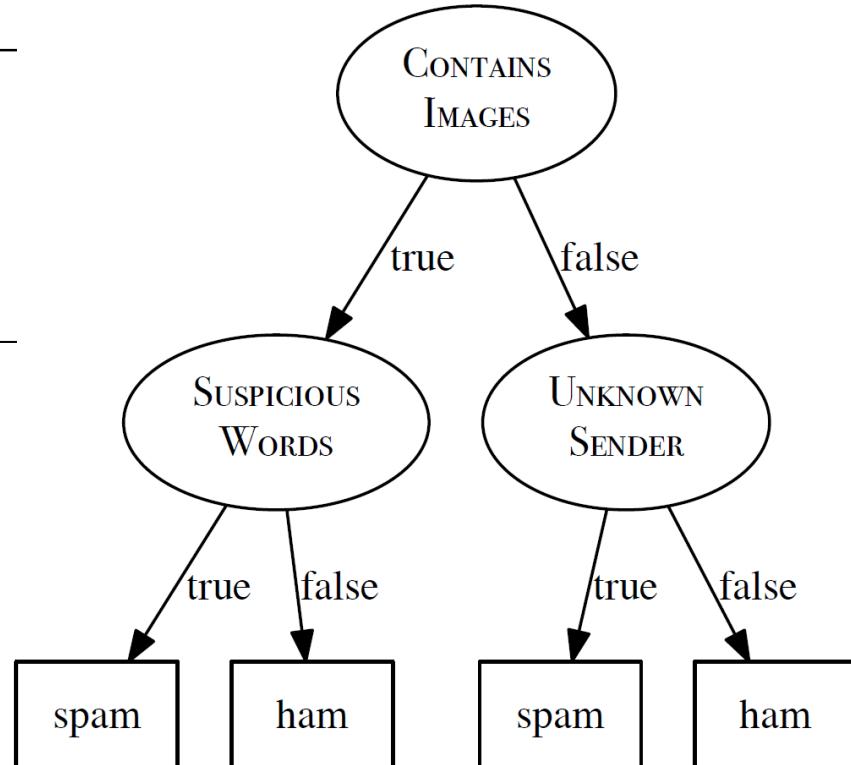
- Three types of nodes: root node, interior nodes and leaf node.
- Root node refers to all instances.
- Interior nodes partition the set of instances based on a descriptive feature.
- Leaf nodes have a label (target feature value) and hopefully correspond to a homogeneous group of instances having the same label.

Example from book

SUSPICIOUS		UNKNOWN	CONTAINS	
ID	WORDS	SENDER	IMAGES	CLASS
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

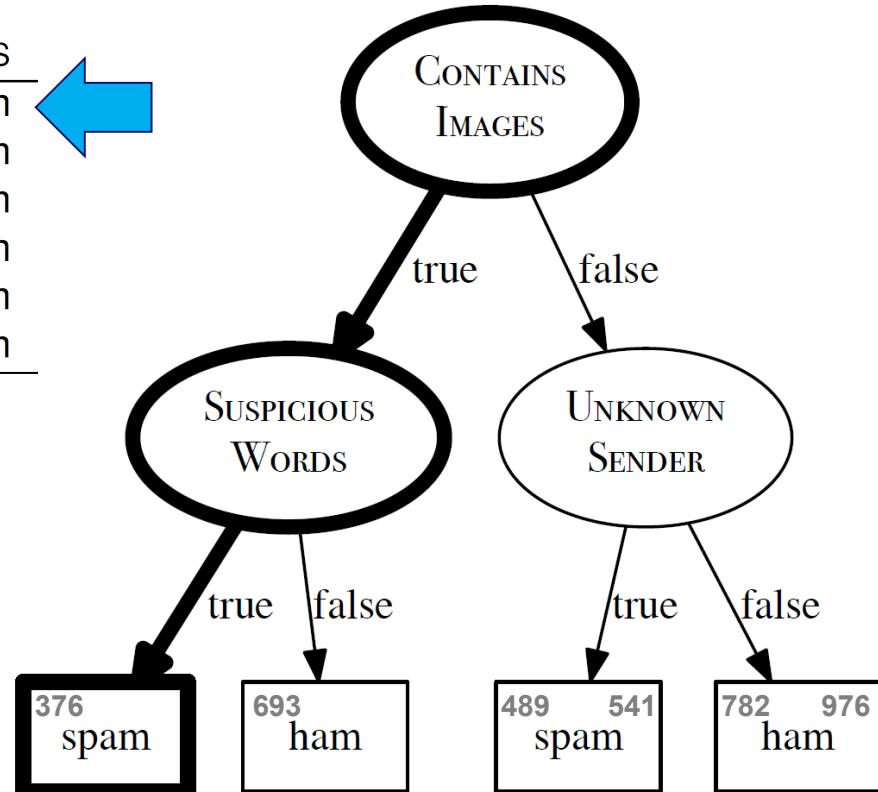
Example from book

ID	SUSPICIOUS WORDS	UNKNOWN SENDER	CONTAINS IMAGES	CLASS
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham



Example from book

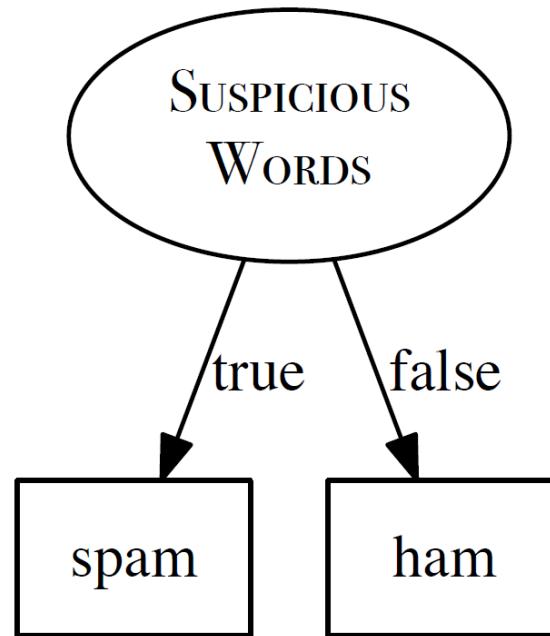
ID	SUSPICIOUS WORDS	UNKNOWN SENDER	CONTAINS IMAGES	CLASS
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham



Example from book

ID	SUSPICIOUS WORDS	UNKNOWN SENDER	CONTAINS IMAGES	CLASS
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

All correctly classified.



376 489 541 693 782 976

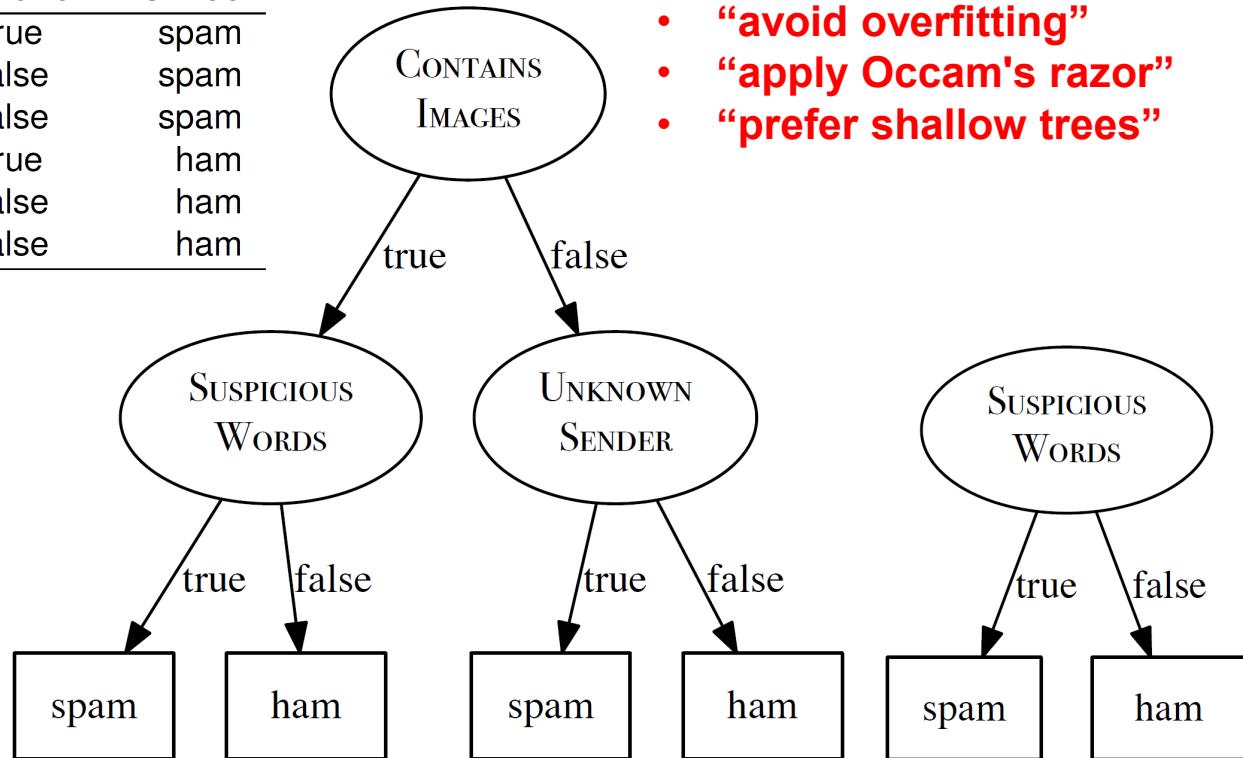


Chair of Process
and Data Science

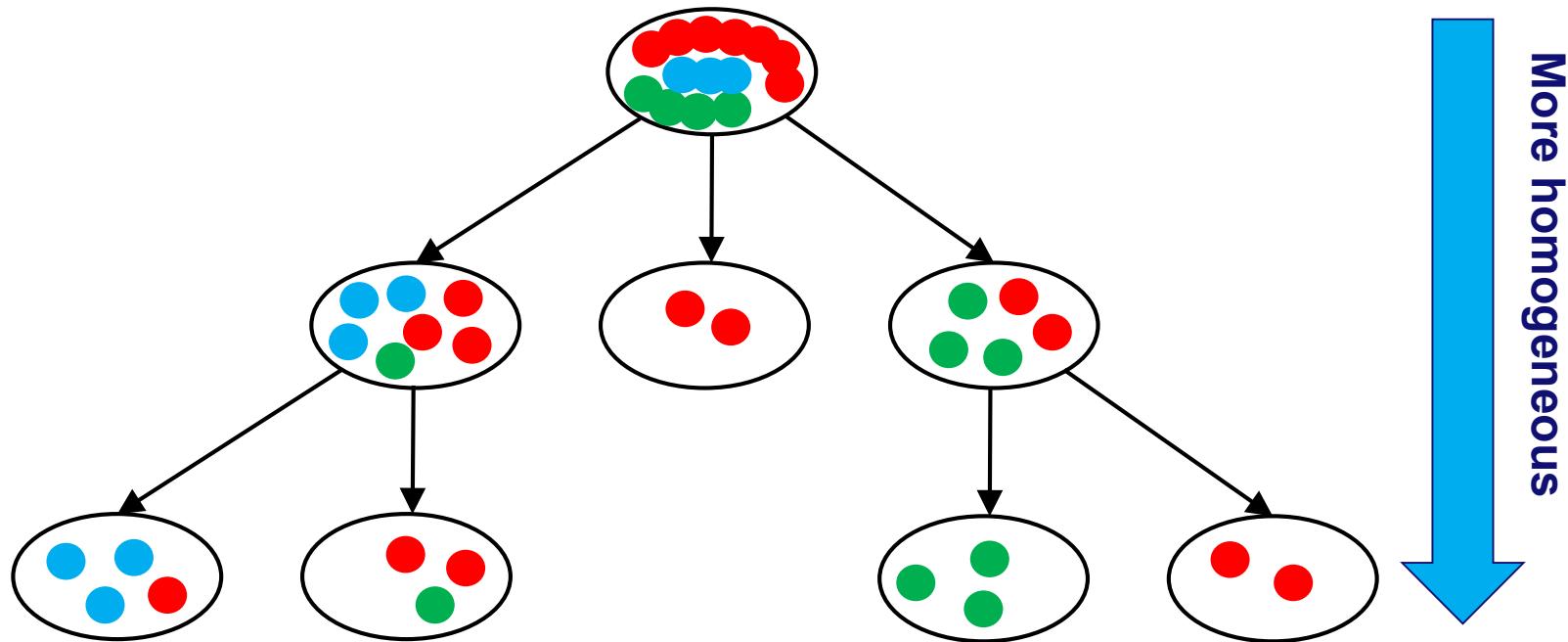
Example from book

ID	SUSPICIOUS WORDS	UNKNOWN SENDER	CONTAINS IMAGES	CLASS
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

Both correctly classify all observed instances, but the “simpler” one seems “better”.



Information gain idea



**Information gain = improvement in knowledge
(predictability of class label in nodes)**

Entropy

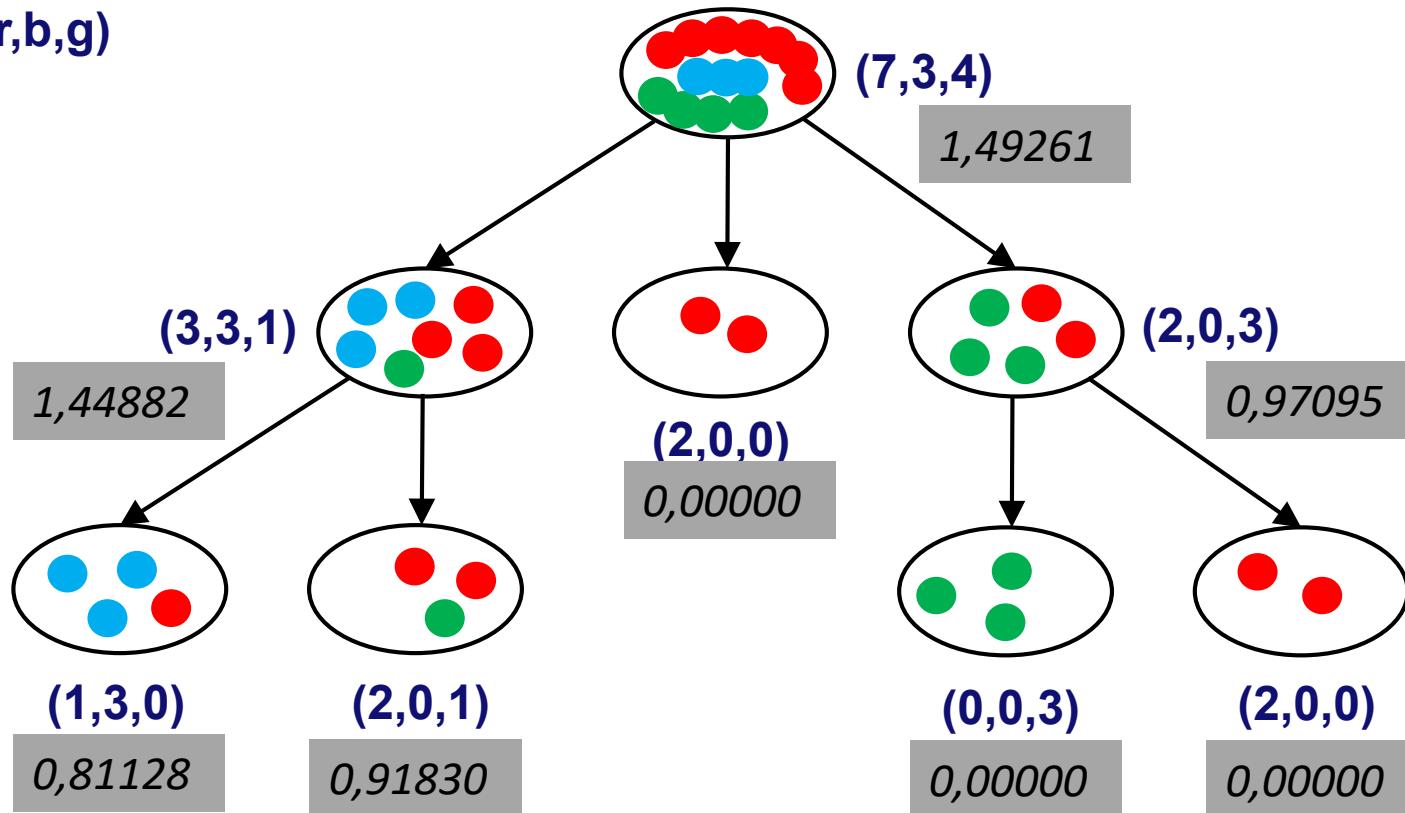


Entropy: Intuition

measure of impurity
uncertainty when guessing
incompressibility

worst case three values: $\log_2(3) \approx 1.58$

(r,b,g)



Formula

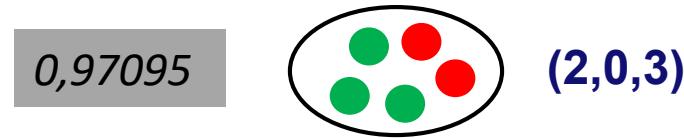


$$H(t) = - \sum_{i=1}^I (P(t = i) \times \log_s(P(t = i)))$$

$$- (7/14 \times \log_2(7/14) + 3/14 \times \log_2(3/14) + 4/14 \times \log_2(4/14)) = 1.49261$$

(we use as logarithm base s=2)

Formula

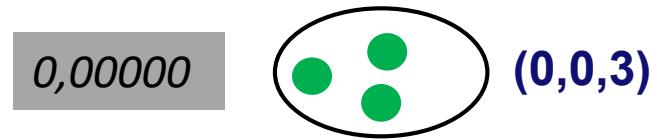


$$H(t) = - \sum_{i=1}^I (P(t = i) \times \log_s(P(t = i)))$$

$$- (2/5 \times \log_2(2/5) + 3/5 \times \log_2(3/5)) = 0.97095$$

(we use as logarithm base s=2)

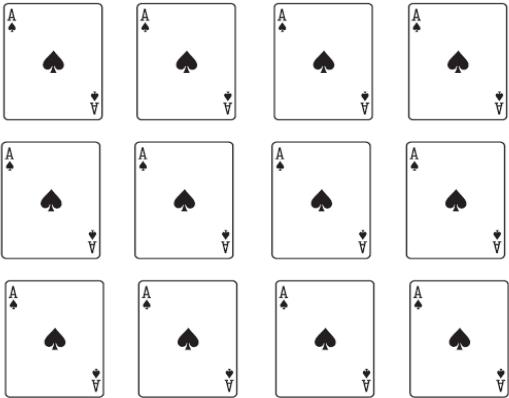
Formula



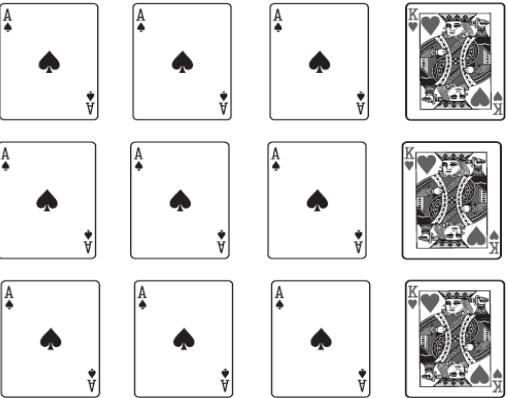
$$H(t) = - \sum_{i=1}^I (P(t = i) \times \log_s(P(t = i)))$$

$$- (3/3 \times \log_2(3/3)) = 0$$

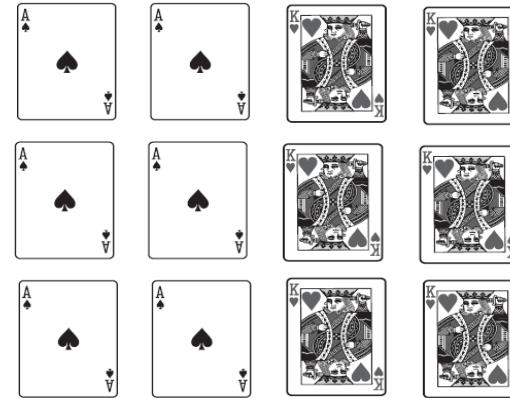
(we use as logarithm base s=2)



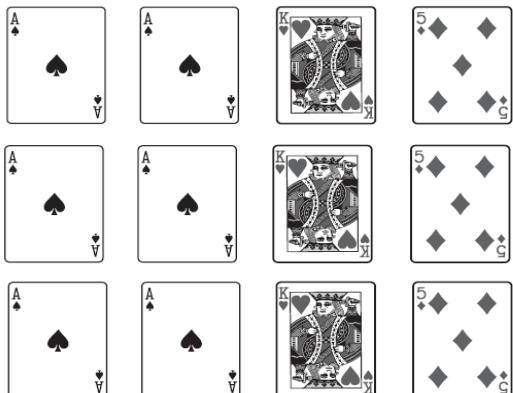
(a) $H(card) = 0.00$



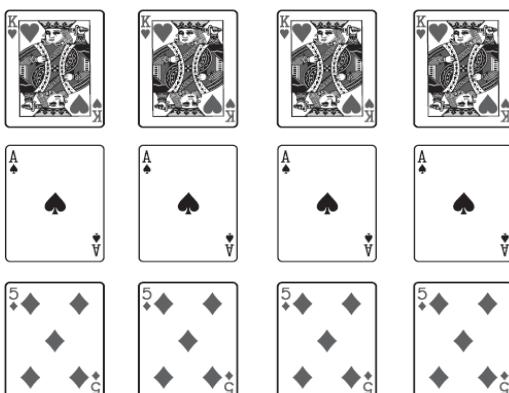
(b) $H(card) = 0.81$



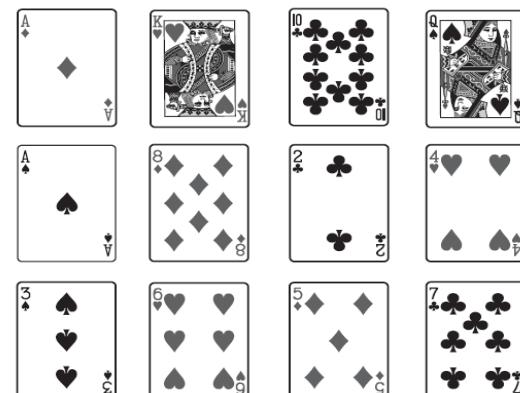
(c) $H(card) = 1.00$



(d) $H(card) = 1.50$



(e) $H(card) = 1.58$

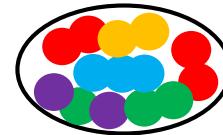


(f) $H(card) = 3.58$

Question

- Suppose that we have I possible values (colors) and n instances (balls).
- What distribution of the n instances over the I possible values yields the lowest entropy?
- What distribution of the n instances over the I possible values yields the highest entropy?

$$H(t) = - \sum_{i=1}^I (P(t = i) \times \log_s(P(t = i)))$$

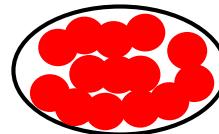


Lowest entropy

- What distribution of the n instances over the I possible values yields the lowest entropy?
- Answer: All instances have the same value!
- $H(t) = 0$

lowest impurity possible
no uncertainty when guessing
highly compressible

$$H(t) = - \sum_{i=1}^I (P(t = i) \times \log_s(P(t = i)))$$



Highest entropy

- What distribution of the n instances over the I possible values yields the highest entropy?
- Answer: Even distribution over all possible values!
- $H(t)$ is maximized

highest impurity possible
highest uncertainty when guessing
incompressible

$$H(t) = - \sum_{i=1}^I (P(t = i) \times \log_s(P(t = i)))$$



Highest entropy

- What distribution of the n instances over the l possible values yields the highest entropy?
- Answer: Even distribution over all possible values!
- Assume $(k, k, k, k, k, \dots, k)$ distribution with $n = l \times k$
- $H(t) = -l (1/l \times \log_2(1/l)) = -\log_2(1/l) = \log_2(l)$

unique values	1	2	3	4	5	6	7	8	9	10
maximal entropy	0,00000	1,00000	1,58496	2,00000	2,32193	2,58496	2,80735	3,00000	3,16993	3,32193

$$H(t) = - \sum_{i=1}^l (P(t = i) \times \log_s(P(t = i)))$$

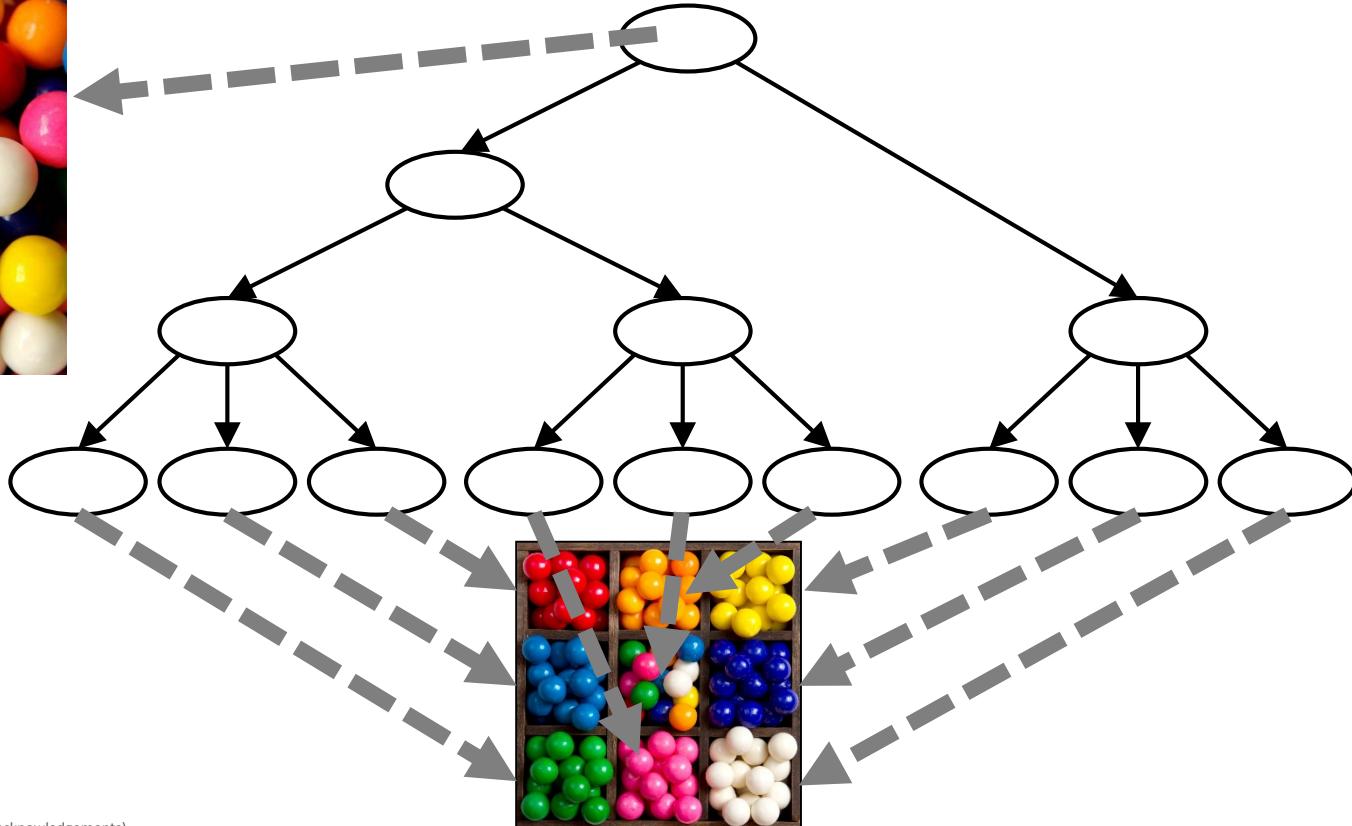


Interpretation

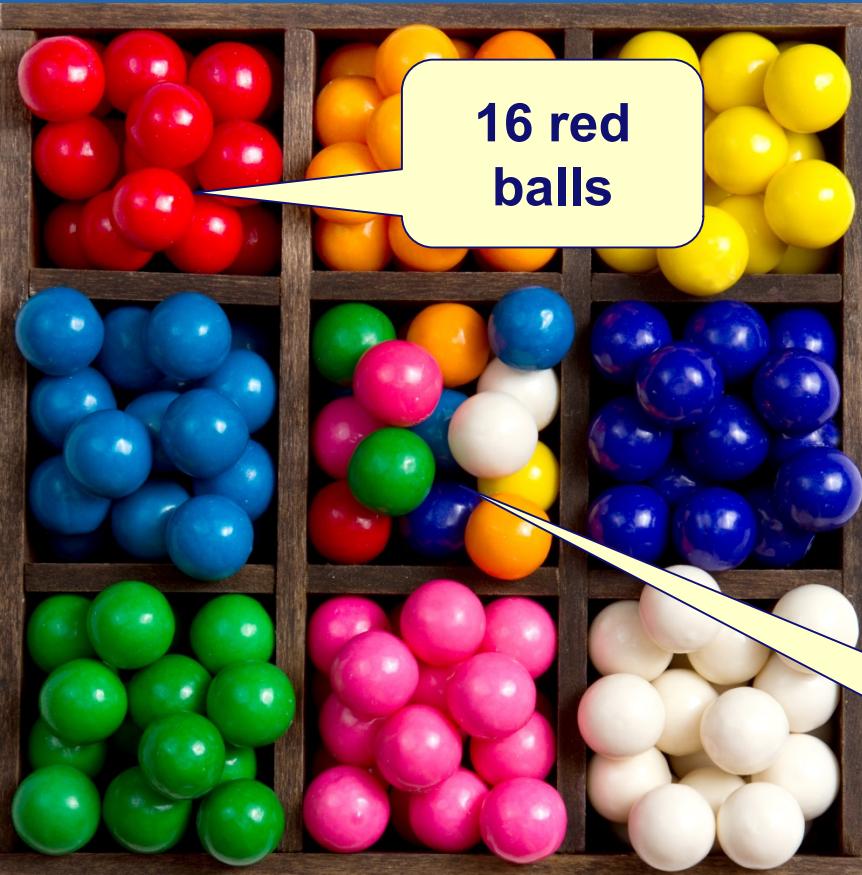
impurity = uncertainty = incompressible

**Number of bits needed to encode one instance
knowing the population it comes from**

Decision tree: Aim is to have “pure” leaves



Question: Compute entropy



- Compute the entropy of all individual cells.
- What is the overall entropy (weighted average)?
- What is the overall entropy if there is just one cell containing all 144 balls?

Answer: $E=3$ for cell in middle



- Cell in the middle:

$$2+2+2+2+2+2+2 = 8 \times 2 \text{ balls}$$

$$E = - \sum_{i=1}^k p_i \log_2(p_i)$$

$$= - \sum_{i=1}^8 \frac{2}{16} \log_2\left(\frac{2}{16}\right)$$

$$= -8 \times \frac{1}{8} \times -3 = 3$$

Answer: $E=0$ for other cells

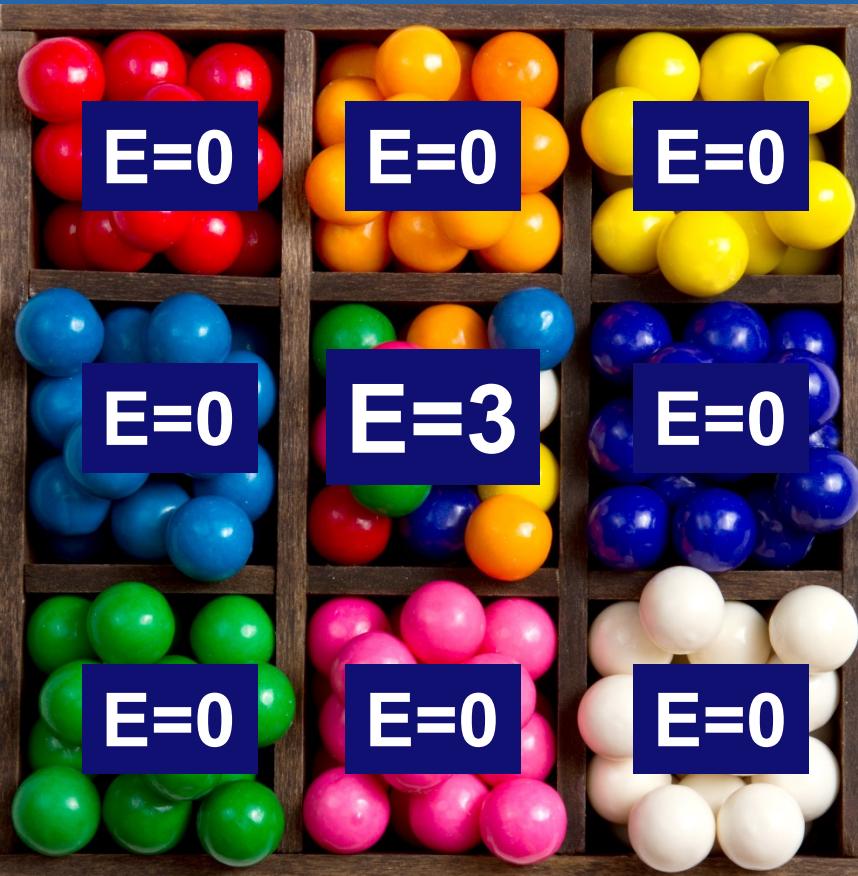


- Other cells:
16+0+0+0+0+0+0+0 balls

$$E = - \sum_{i=1}^k p_i \log_2(p_i)$$

$$\begin{aligned} &= - \sum_{i=1}^1 \frac{16}{16} \log_2\left(\frac{16}{16}\right) \\ &= -1 \times 0 = 0 \end{aligned}$$

Overall entropy (weighted average): $E=0.33$



$$\begin{aligned}E &= \frac{16}{144} \times 0 + \frac{16}{144} \times 0 + \dots + \frac{16}{144} \times 3 \\&= 8 \times \frac{16}{144} \times 0 + \frac{16}{144} \times 3 \\&= \frac{1}{9} \times 3 = \frac{1}{3}\end{aligned}$$

Entropy after mixing the 9 cells: $E=3$



- **144 balls having 8 different colors:
18:18:18:18:18:18:18:18**

$$\begin{aligned} E &= - \sum_{i=1}^k p_i \log_2(p_i) \\ &= - \sum_{i=1}^8 \frac{18}{144} \log_2\left(\frac{18}{144}\right) \\ &= -8 \times \frac{1}{8} \times -3 = 3 \end{aligned}$$



$E=0.3333$

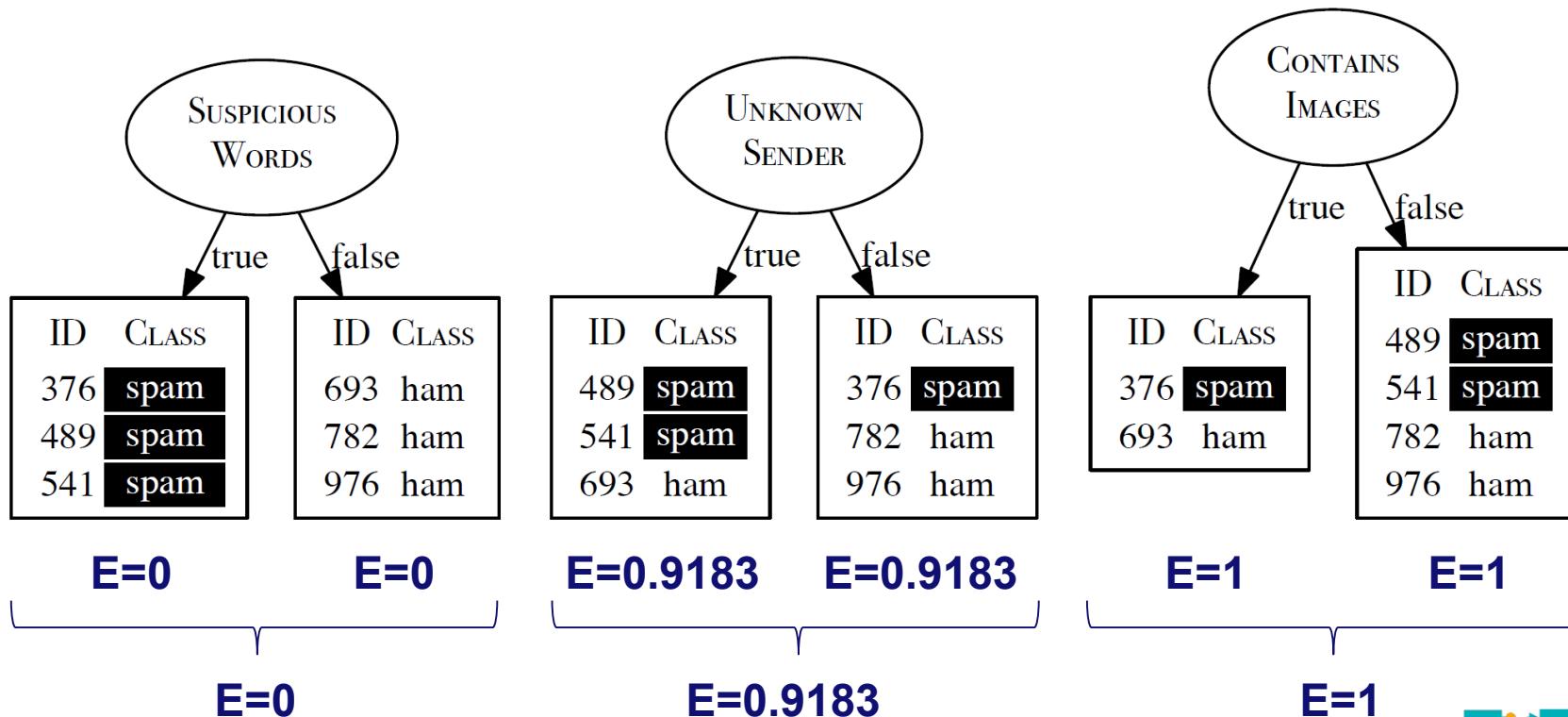
information gain = 2.6666



$E=3$



Example revisited



Building an entropy calculator in Excel

The screenshot shows an Excel spreadsheet titled "entropy-calculator2.xls [Compatibility Mode] - Excel". The table has columns labeled A through L. Row 1 contains labels: "value" under A, "val 1" under B, "val 2" under C, "val 3" under D, "val 4" under E, "val 5" under F, "val 6" under G, "val 7" under H, "val 8" under I, "val 9" under J, "val 10" under K, and "sum" under L. Row 2 contains values: 7, 3, 4, 0, 0, 0, 0, 0, 0, 0, 0, and 14,00000. Row 3 contains fractions: 0,50000, 0,21429, 0,28571, 0,00000, 0,00000, 0,00000, 0,00000, 0,00000, 0,00000, 0,00000, and 1,00000. Row 4 contains calculated values: 0,50000, 0,47623, 0,51639, 0,00000, 0,00000, 0,00000, 0,00000, 0,00000, 0,00000, 0,00000, and 1,49261. The formula bar shows the current cell is I10.

Annotations explain the formulas used:

- "frequencies" points to the row of values (B2:K2).
- "=SUM(B2:K2)" points to the "sum" value in row 2.
- "=F2/L2" points to the fraction in row 3.
- "=B2/L2" points to the fraction in row 4.
- "=IF(F3>0; -F3*LOG(F3;2);0)" points to the calculation in row 4.
- "=IF(B3>0; -B3*LOG(B3;2);0)" points to the calculation in row 4.
- "=SUM(B4:K4)" points to the sum in row 4.

	A	B	C	D	E	F	G	H	I	J	K	L
1		val 1	val2	val 3	val 4	val 5	val 6	val 7	val 8	val 9	val 10	sum
2	value	7	3	4	0	0	0	0	0	0	0	14,00000
3	fraction	0,50000	0,21429	0,28571	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	1,00000
4	-p*log(p)	0,50000	0,47623	0,51639	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	1,49261

$$H(t) = - \sum_{i=1}^I (P(t = i) \times \log_s(P(t = i)))$$



Building an entropy calculator in Excel

minimal entropy	val 1	val2	val 3	val 4	val 5	val 6	val 7	val 8	val 9	val 10	sum
value	10	0	0	0	0	0	0	0	0	0	10,00000
fraction	1,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	1,00000
-p*log(p)	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000

maximal entropy	val 1	val2	val 3	val 4	val 5	val 6	val 7	val 8	val 9	val 10	sum
value	1	1	1	1	1	1	1	1	1	1	10,00000
fraction	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	1,00000
-p*log(p)	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	3,32193

only relative frequencies matter	val 1	val2	val 3	val 4	val 5	val 6	val 7	val 8	val 9	val 10	sum
value	10	10	10	10	10	10	10	10	10	10	100,00000
fraction	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	1,00000
-p*log(p)	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	3,32193

Building an entropy calculator in Excel

as before	val 1	val2	val 3	val 4	val 5	val 6	val 7	val 8	val 9	val 10	sum
value	10	10	10	10	10	10	10	10	10	10	100,00000
fraction	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	0,10000	1,00000
-p*log(p)	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	0,33219	3,32193

concentrated	val 1	val2	val 3	val 4	val 5	val 6	val 7	val 8	val 9	val 10	sum
value	55	9	8	7	6	5	4	3	2	1	100,00000
fraction	0,55000	0,09000	0,08000	0,07000	0,06000	0,05000	0,04000	0,03000	0,02000	0,01000	1,00000
-p*log(p)	0,47437	0,31265	0,29151	0,26856	0,24353	0,21610	0,18575	0,15177	0,11288	0,06644	2,32356

even more concentrated	val 1	val2	val 3	val 4	val 5	val 6	val 7	val 8	val 9	val 10	sum
value	91	1	1	1	1	1	1	1	1	1	100,00000
fraction	0,91000	0,01000	0,01000	0,01000	0,01000	0,01000	0,01000	0,01000	0,01000	0,01000	1,00000
-p*log(p)	0,12382	0,06644	0,06644	0,06644	0,06644	0,06644	0,06644	0,06644	0,06644	0,06644	0,72176

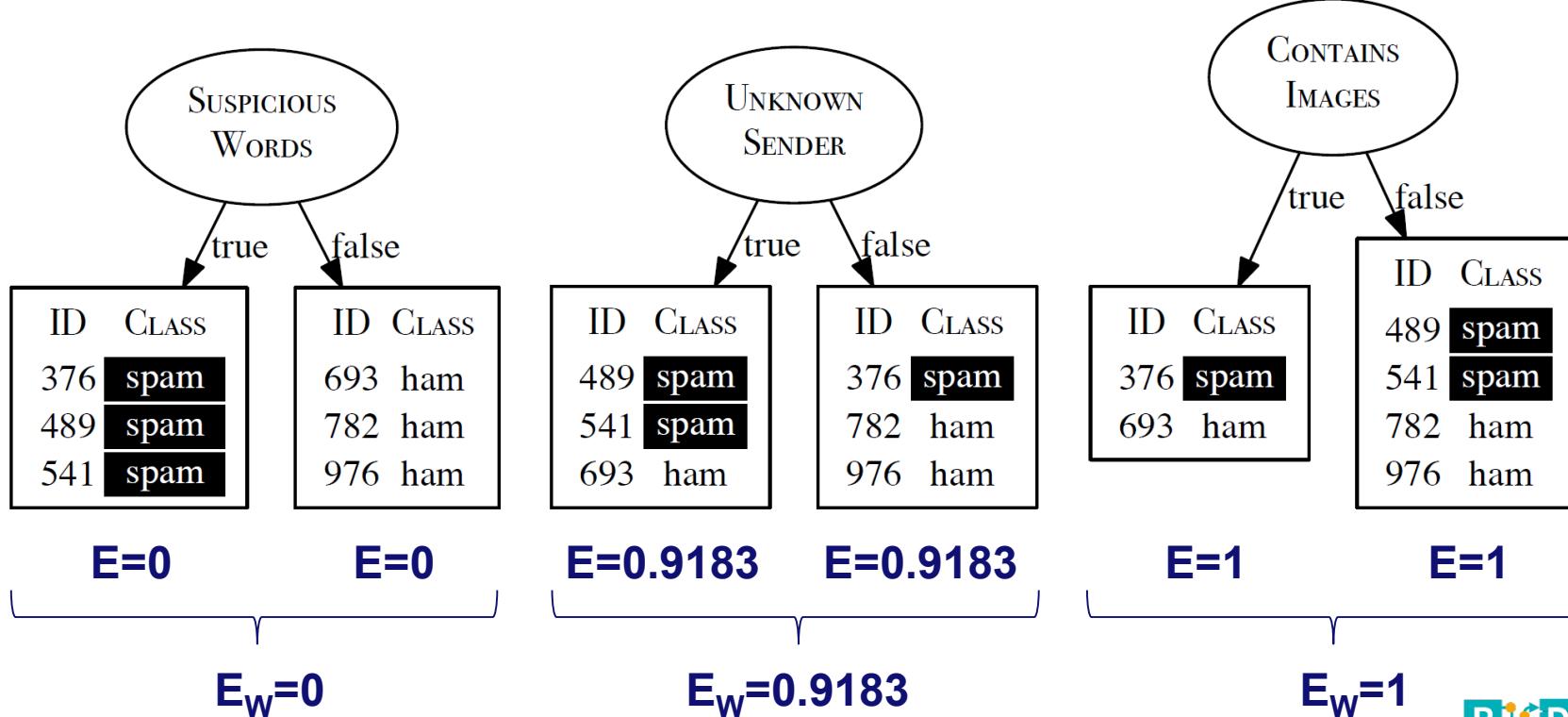
Example from book revisited

ID	SUSPICIOUS WORDS	UNKNOWN SENDER	CONTAINS IMAGES	CLASS
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

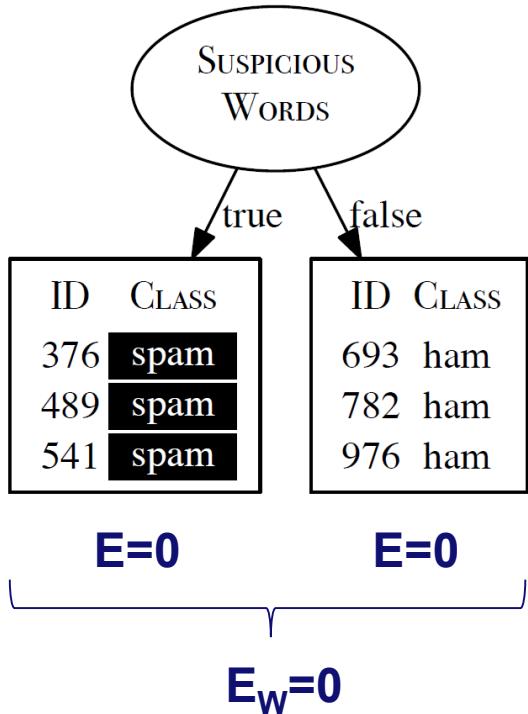
$$\begin{aligned} H(t, \mathcal{D}) &= - \sum_{l \in \{ \text{'spam'}, \text{'ham'} \}} (P(t = l) \times \log_2(P(t = l))) \\ &= - ((P(t = \text{'spam'}) \times \log_2(P(t = \text{'spam'}))) \\ &\quad + (P(t = \text{'ham'}) \times \log_2(P(t = \text{'ham'})))) \\ &= - \left(\left(\frac{3}{6} \times \log_2(\frac{3}{6}) \right) + \left(\frac{3}{6} \times \log_2(\frac{3}{6}) \right) \right) \\ &= 1 \text{ bit} \end{aligned}$$

Example taken from **Fundamentals of Machine Learning for Predictive Data Analytics** by J. Kelleher, B. Mac Namee and A. D'Arcy.

Example revisited



Example revisited

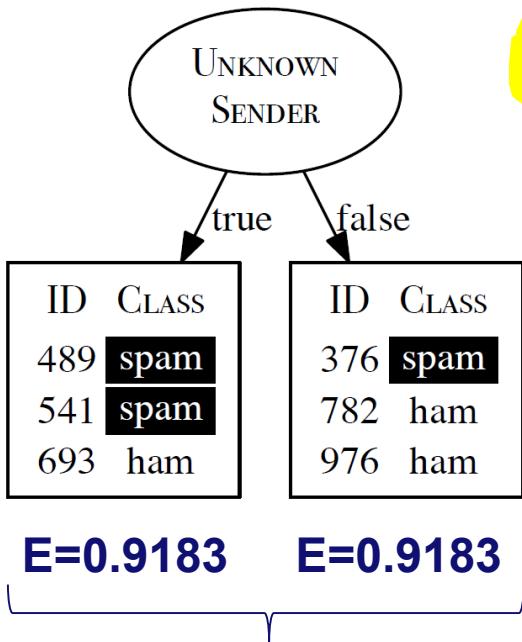


$rem(\text{WORDS}, \mathcal{D})$

$$\begin{aligned}
 &= \left(\frac{|\mathcal{D}_{\text{WORDS}=T}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{\text{WORDS}=T}) \right) + \left(\frac{|\mathcal{D}_{\text{WORDS}=F}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{\text{WORDS}=F}) \right) \\
 &= \left(\frac{3}{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t=l) \times \log_2(P(t=l)) \right) \right) \\
 &\quad + \left(\frac{3}{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t=l) \times \log_2(P(t=l)) \right) \right) \\
 &= \left(\frac{3}{6} \times \left(- \left(\left(\frac{3}{3} \times \log_2(\frac{3}{3}) \right) + \left(\frac{0}{3} \times \log_2(\frac{0}{3}) \right) \right) \right) \right) \\
 &\quad + \left(\frac{3}{6} \times \left(- \left(\left(\frac{0}{3} \times \log_2(\frac{0}{3}) \right) + \left(\frac{3}{3} \times \log_2(\frac{3}{3}) \right) \right) \right) \right) = 0 \text{ bits}
 \end{aligned}$$

rem = remaining entropy (weighed average)

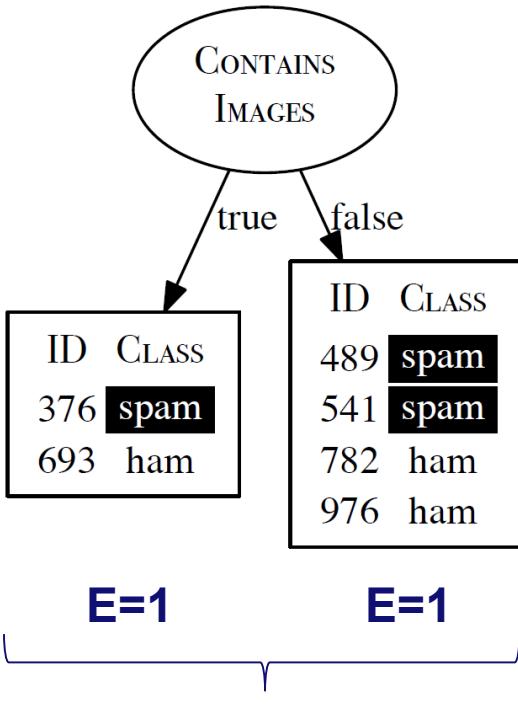
Example revisited



$\text{rem}(\text{SENDER}, \mathcal{D})$

$$= \left(\frac{|\mathcal{D}_{\text{SENDER}=T}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{\text{SENDER}=T}) \right) + \left(\frac{|\mathcal{D}_{\text{SENDER}=F}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{\text{SENDER}=F}) \right)$$
$$= \left(\frac{3}{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t = l) \times \log_2(P(t = l)) \right) \right)$$
$$+ \left(\frac{3}{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t = l) \times \log_2(P(t = l)) \right) \right)$$
$$= \left(\frac{3}{6} \times \left(- \left(\left(\frac{2}{3} \times \log_2(\frac{2}{3}) \right) + \left(\frac{1}{3} \times \log_2(\frac{1}{3}) \right) \right) \right) \right)$$
$$+ \left(\frac{3}{6} \times \left(- \left(\left(\frac{1}{3} \times \log_2(\frac{1}{3}) \right) + \left(\frac{2}{3} \times \log_2(\frac{2}{3}) \right) \right) \right) \right) = 0.9183 \text{ bits}$$

Example revisited



$rem(\text{IMAGES}, \mathcal{D})$

$$\begin{aligned}
 &= \left(\frac{|\mathcal{D}_{\text{IMAGES}=T}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{\text{IMAGES}=T}) \right) + \left(\frac{|\mathcal{D}_{\text{IMAGES}=F}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{\text{IMAGES}=F}) \right) \\
 &= \left(\frac{2}{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t = l) \times \log_2(P(t = l)) \right) \right) \\
 &\quad + \left(\frac{4}{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t = l) \times \log_2(P(t = l)) \right) \right) \\
 &= \left(\frac{2}{6} \times \left(- \left(\left(\frac{1}{2} \times \log_2(\frac{1}{2}) \right) + \left(\frac{1}{2} \times \log_2(\frac{1}{2}) \right) \right) \right) \right) \\
 &\quad + \left(\frac{4}{6} \times \left(- \left(\left(\frac{2}{4} \times \log_2(\frac{2}{4}) \right) + \left(\frac{2}{4} \times \log_2(\frac{2}{4}) \right) \right) \right) \right) = 1 \text{ bit}
 \end{aligned}$$

Information gain

$$\begin{aligned}IG(\text{SUSPICIOUS WORDS}, \mathcal{D}) &= H(\text{CLASS}, \mathcal{D}) - \text{rem}(\text{SUSPICIOUS WORDS}, \mathcal{D}) \\&= 1 - 0 = 1 \text{ bit}\end{aligned}$$

good

$$\begin{aligned}IG(\text{UNKNOWN SENDER}, \mathcal{D}) &= H(\text{CLASS}, \mathcal{D}) - \text{rem}(\text{UNKNOWN SENDER}, \mathcal{D}) \\&= 1 - 0.9183 = 0.0817 \text{ bits}\end{aligned}$$

not so good

$$\begin{aligned}IG(\text{CONTAINS IMAGES}, \mathcal{D}) &= H(\text{CLASS}, \mathcal{D}) - \text{rem}(\text{CONTAINS IMAGES}, \mathcal{D}) \\&= 1 - 1 = 0 \text{ bits}\end{aligned}$$

worst



Chair of Process
and Data Science

ID3 algorithm



ID3 (Iterative Dichotomiser 3)

- ID3 (Iterative Dichotomiser 3) was developed by Ross Quinlan (1986).
- ID3 is the predecessor of algorithms like C4.5.
- Key idea:
 1. Calculate the entropy of every attribute using the data set D.
 2. Split the set D into subsets using the attribute for which the resulting entropy (after splitting) is minimal (or, equivalently, information gain is maximum).
 3. Make a decision tree node containing that attribute.
 4. Recurse on subsets using remaining attributes.

ID3 Algorithm

Algorithm 4.1 Pseudocode description of the ID3 algorithm.

Require: set of descriptive features \mathbf{d}

Require: set of training instances \mathcal{D}

- 1: **if** all the instances in \mathcal{D} have the same target level C **then**
- 2: **return** a decision tree consisting of a leaf node with label C
- 3: **else if** \mathbf{d} is empty **then**
- 4: **return** a decision tree consisting of a leaf node with the label of the majority target level in \mathcal{D}
- 5: **else if** \mathcal{D} is empty **then**
- 6: **return** a decision tree consisting of a leaf node with the label of the majority target level of the dataset of the immediate parent node
- 7: **else**
- 8: $\mathbf{d}[\text{best}] \leftarrow \arg \max_{d \in \mathbf{d}} IG(d, \mathcal{D})$
- 9: make a new node, $Node_{\mathbf{d}[\text{best}]}$, and label it with $\mathbf{d}[\text{best}]$
- 10: partition \mathcal{D} using $\mathbf{d}[\text{best}]$
- 11: remove $\mathbf{d}[\text{best}]$ from \mathbf{d}
- 12: **for** each partition \mathcal{D}_i of \mathcal{D} **do**
- 13: grow a branch from $Node_{\mathbf{d}[\text{best}]}$ to the decision tree created by rerunning ID3 with $\mathcal{D} = \mathcal{D}_i$



Algorithm

Algorithm 4.1 Pseudocode description of the ID3 algorithm.

Require: set of descriptive features \mathbf{d}

Require: set of training instances \mathcal{D}

```
1: if all the instances in  $\mathcal{D}$  have the same target level  $C$  then
2:   return a decision tree consisting of a leaf node with label  $C$ 
3: else if  $\mathbf{d}$  is empty then
4:   return a decision tree consisting of a leaf node with the label of the
   majority target level in  $\mathcal{D}$ 
5: else if  $\mathcal{D}$  is empty then
6:   return a decision tree consisting of a leaf node with the label of the
   majority target level of the dataset of the immediate parent node
7: else
8:    $\mathbf{d}[\text{best}] \leftarrow \arg \max_{d \in \mathbf{d}} IG(d, \mathcal{D})$ 
9:   make a new node,  $Node_{\mathbf{d}[\text{best}]}$ , and label it with  $\mathbf{d}[\text{best}]$ 
10:  partition  $\mathcal{D}$  using  $\mathbf{d}[\text{best}]$ 
11:  remove  $\mathbf{d}[\text{best}]$  from  $\mathbf{d}$ 
12:  for each partition  $\mathcal{D}_i$  of  $\mathcal{D}$  do
13:    grow a branch from  $Node_{\mathbf{d}[\text{best}]}$  to the decision tree created by rerunning
       ID3 with  $\mathcal{D} = \mathcal{D}_i$ 
```

Three reasons to stop:

- All instances have the same classification
(label=consensus value)
- No features left
(label=majority value)
- Data set is empty
(label=majority parent)



Algorithm

Algorithm 4.1 Pseudocode description of the ID3 algorithm.

Require: set of descriptive features \mathbf{d}

Require: set of training instances \mathcal{D}

```
1: if all the instances in  $\mathcal{D}$  have the same target level  $C$  then
2:   return a decision tree consisting of a leaf node with label  $C$ 
3: else if  $\mathbf{d}$  is empty then
4:   return a decision tree consisting of a leaf node with the label of the
   majority target level in  $\mathcal{D}$ 
5: else if  $\mathcal{D}$  is empty then
6:   return a decision tree consisting of a leaf node with the label of the
   majority target level of the dataset of the immediate parent node
7: else
8:    $\mathbf{d}[\text{best}] \leftarrow \arg \max_{d \in \mathbf{d}} IG(d, \mathcal{D})$ 
9:   make a new node,  $Node_{\mathbf{d}[\text{best}]}$ , and label it with  $\mathbf{d}[\text{best}]$ 
10:  partition  $\mathcal{D}$  using  $\mathbf{d}[\text{best}]$ 
11:  remove  $\mathbf{d}[\text{best}]$  from  $\mathbf{d}$ 
12:  for each partition  $\mathcal{D}_i$  of  $\mathcal{D}$  do
13:    grow a branch from  $Node_{\mathbf{d}[\text{best}]}$  to the decision tree created by rerunning
       ID3 with  $\mathcal{D} = \mathcal{D}_i$ 
```

- **Pick a feature which maximizes information gain.**
- **Once a feature is picked along a path from the root, it cannot be used again.**

Algorithm

Algorithm 4.1 Pseudocode description of the ID3 algorithm.

Require: set of descriptive features \mathbf{d}

Require: set of training instances \mathcal{D}

```
1: if all the instances in  $\mathcal{D}$  have the same target level  $C$  then
2:   return a decision tree consisting of a leaf node with label  $C$ 
3: else if  $\mathbf{d}$  is empty then
4:   return a decision tree consisting of a leaf node with the label of the
   majority target level in  $\mathcal{D}$ 
5: else if  $\mathcal{D}$  is empty then
6:   return a decision tree consisting of a leaf node with the label of the
   majority target level of the dataset of the immediate parent node
7: else
8:    $\mathbf{d}[\text{best}] \leftarrow \arg \max_{d \in \mathbf{d}} IG(d, \mathcal{D})$ 
9:   make a new node,  $Node_{\mathbf{d}[\text{best}]}$ , and label it with  $\mathbf{d}[\text{best}]$ 
10:  partition  $\mathcal{D}$  using  $\mathbf{d}[\text{best}]$ 
11:  remove  $\mathbf{d}[\text{best}]$  from  $\mathbf{d}$ 
12:  for each partition  $\mathcal{D}_i$  of  $\mathcal{D}$  do
13:    grow a branch from  $Node_{\mathbf{d}[\text{best}]}$  to the decision tree created by rerunning
       ID3 with  $\mathcal{D} = \mathcal{D}_i$ 
```

- **Create subproblems based on the selected feature.**

Example from book

ID	STREAM	SLOPE	ELEVATION	VEGETATION
1	false	steep	high	chaparral
2	true	moderate	low	riparian
3	true	steep	medium	riparian
4	false	steep	medium	chaparral
5	false	flat	high	conifer
6	true	steep	highest	conifer
7	true	steep	high	chaparral

$$H(\text{VEGETATION}, \mathcal{D})$$

$$= - \sum_{I \in \{\text{'chaparral'}, \text{'riparian'}, \text{'conifer'}\}} P(\text{VEGETATION} = I) \times \log_2 (P(\text{VEGETATION} = I))$$

$$= - \left(\left(\frac{3}{7} \times \log_2 \left(\frac{3}{7} \right) \right) + \left(\frac{2}{7} \times \log_2 \left(\frac{2}{7} \right) \right) + \left(\frac{2}{7} \times \log_2 \left(\frac{2}{7} \right) \right) \right) \\ = 1.5567 \text{ bits}$$

Example taken from **Fundamentals of Machine Learning for Predictive Data Analytics** by J. Kelleher, B. Mac Namee and A. D'Arcy.

Example from book

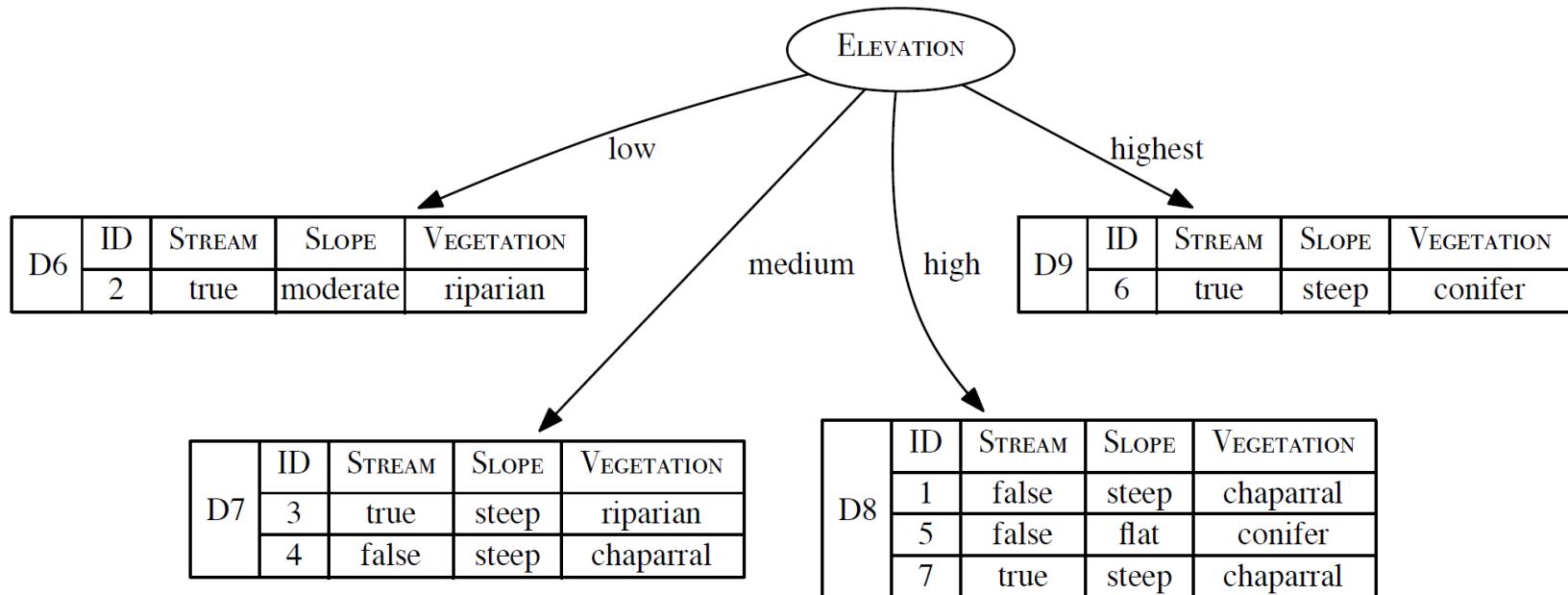
ID	STREAM	SLOPE	ELEVATION	VEGETATION
1	false	steep	high	chaparral
2	true	moderate	low	riparian
3	true	steep	medium	riparian
4	false	steep	medium	chaparral
5	false	flat	high	conifer
6	true	steep	highest	conifer
7	true	steep	high	chaparral

$$\begin{aligned}
 H(\text{VEGETATION}, \mathcal{D}) &= - \sum_{I \in \{\text{'chaparral'}, \text{'riparian'}, \text{'conifer'}\}} P(\text{VEGETATION} = I) \times \log_2(P(\text{VEGETATION} = I)) \\
 &= - \left(\left(\frac{3}{7} \times \log_2(\frac{3}{7}) \right) + \left(\frac{2}{7} \times \log_2(\frac{2}{7}) \right) + \left(\frac{2}{7} \times \log_2(\frac{2}{7}) \right) \right) \\
 &= 1.5567 \text{ bits}
 \end{aligned}$$

Split By Feature	Level	Part.	Instances	Partition Entropy	Rem.	Info. Gain
STREAM	'true'	\mathcal{D}_1	$\mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_6, \mathbf{d}_7$	1.5	1.2507	0.3060
	'false'	\mathcal{D}_2	$\mathbf{d}_1, \mathbf{d}_4, \mathbf{d}_5$	0.9183		
SLOPE	'flat'	\mathcal{D}_3	\mathbf{d}_5	0	0.9793	0.5774
	'moderate'	\mathcal{D}_4	\mathbf{d}_2	0		
	'steep'	\mathcal{D}_5	$\mathbf{d}_1, \mathbf{d}_3, \mathbf{d}_4, \mathbf{d}_6, \mathbf{d}_7$	1.3710		
ELEVATION	'low'	\mathcal{D}_6	\mathbf{d}_2	0	0.6793	0.8774
	'medium'	\mathcal{D}_7	$\mathbf{d}_3, \mathbf{d}_4$	1.0		
	'high'	\mathcal{D}_8	$\mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_7$	0.9183		
	'highest'	\mathcal{D}_9	\mathbf{d}_6	0		

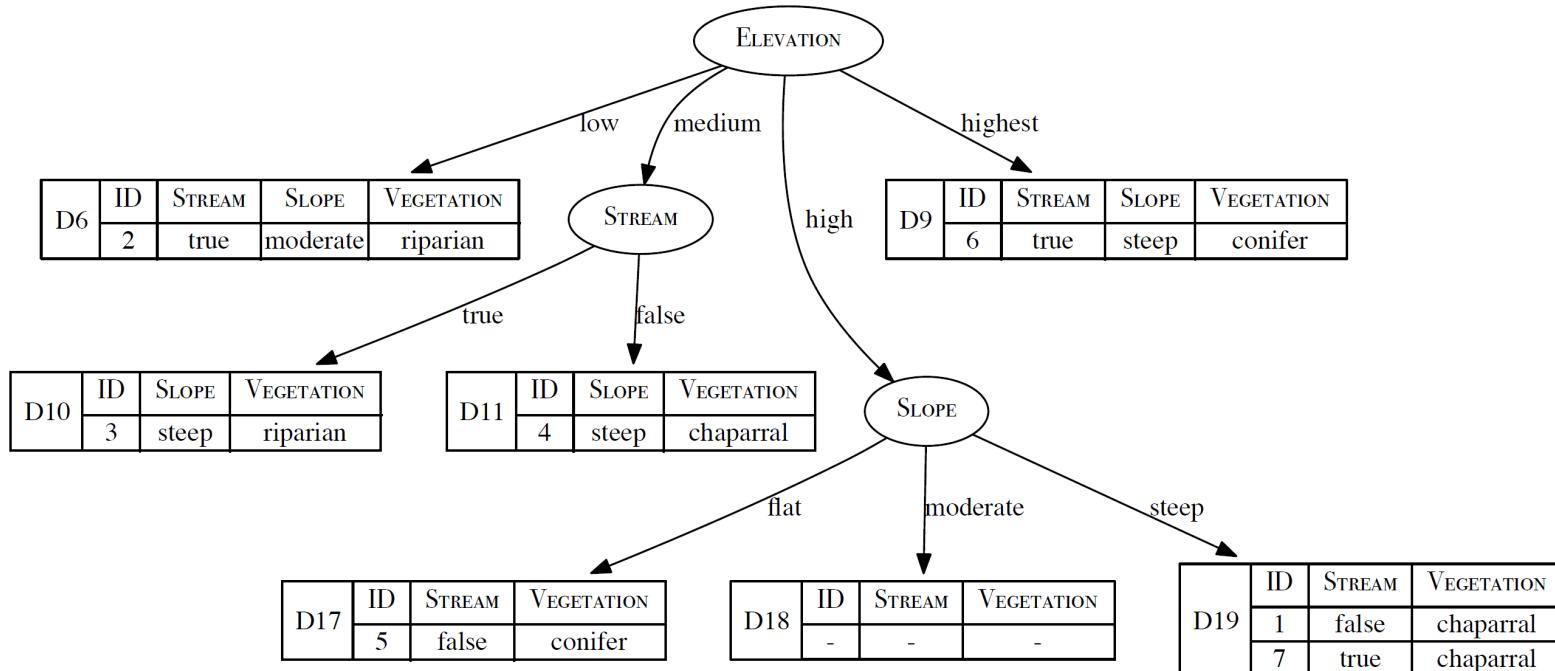


Example from book



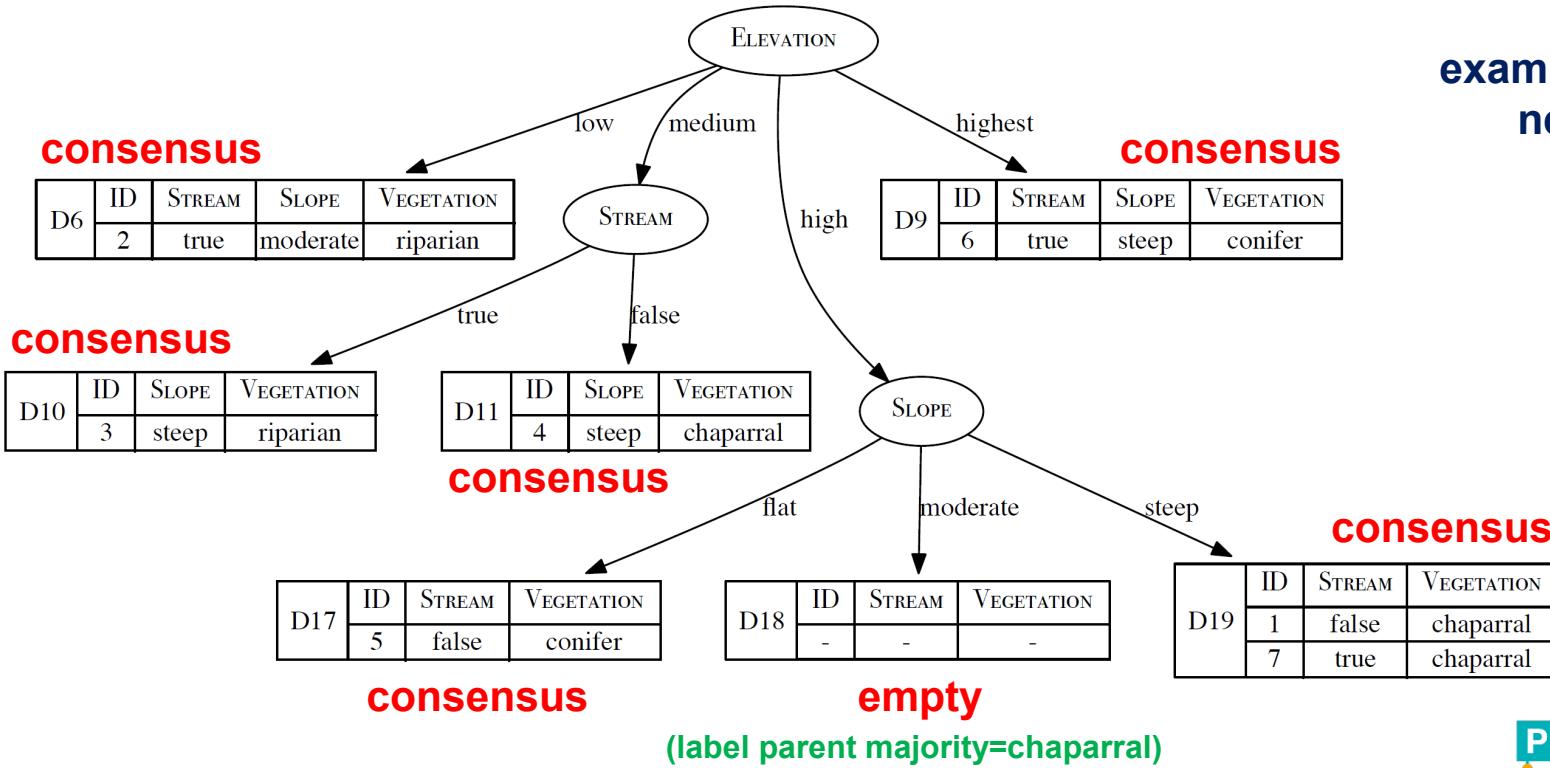
Four smaller data sets resulting from the split using the Elevation feature.

Example from book



Recursion until one of the three conditions holds.

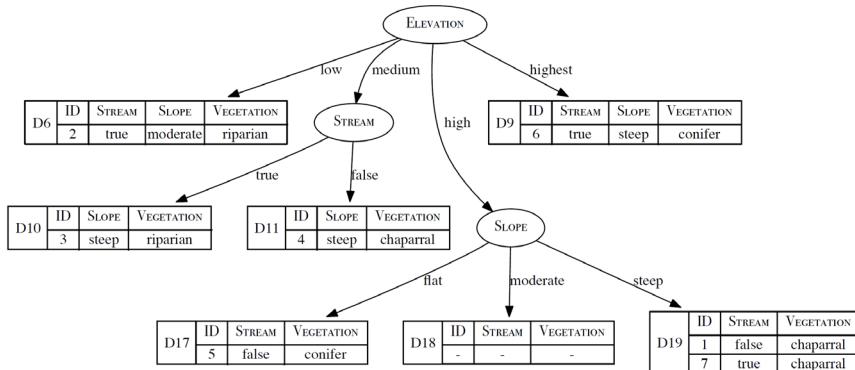
Example from book



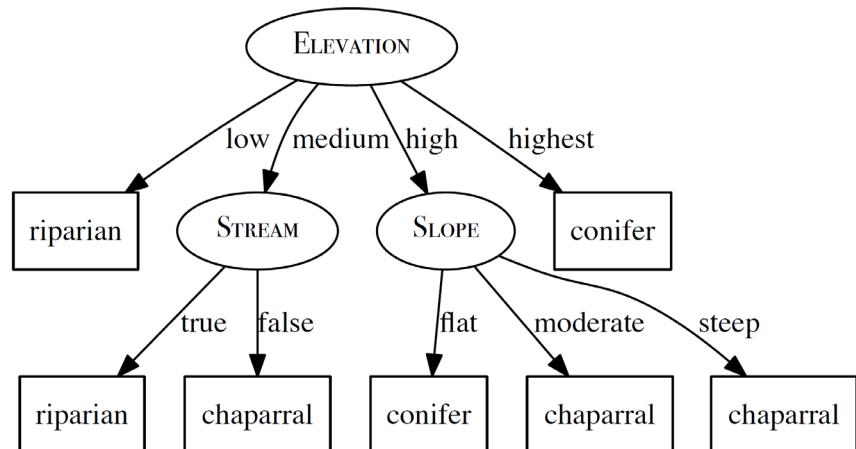
In this small example a branch never uses all features.

Example from book

ID	STREAM	SLOPE	ELEVATION	VEGETATION
1	false	steep	high	chaparral
2	true	moderate	low	riparian
3	true	steep	medium	riparian
4	false	steep	medium	chaparral
5	false	flat	high	conifer
6	true	steep	highest	conifer
7	true	steep	high	chaparral



Final result



Note: no instances here,
took parent label.



Variations of the ID3 algorithm



Alternative information gain notions

- Information gain aims to measure the improvement in “purity” / ”predictability” / ”compressibility”
- Example alternatives:
 - Information gain ratio (GR)
 - Gini index (Gini)

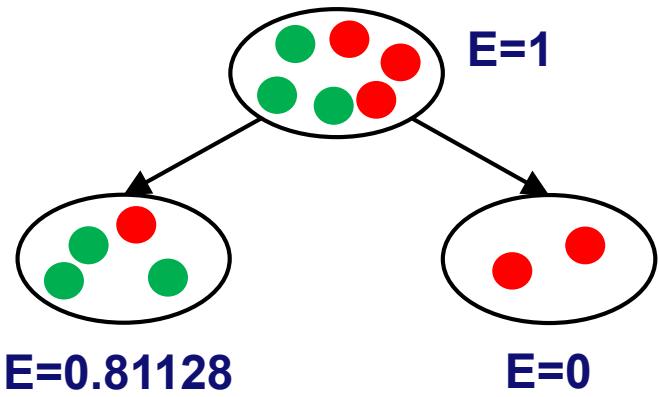
Information gain ratio

- The standard information gain notion favors features with many values (split in many subsets increases entropy).
- Information gain ratio addresses this:

$$GR(d, \mathcal{D}) = \frac{IG(d, \mathcal{D})}{-\sum_{l \in levels(d)} (P(d = l) \times \log_2(P(d = l)))}$$

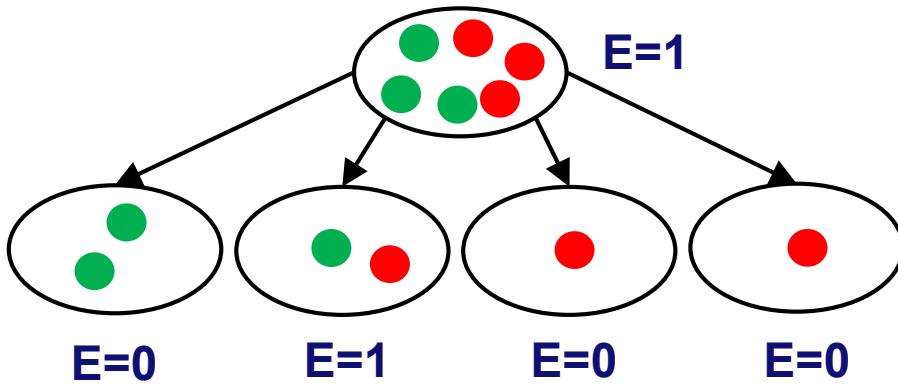
(like making an absolute value relative)

Information gain



$$E_w = 4/6 \times 0.81128 = 0.5408$$

$$IG = 0.4591$$

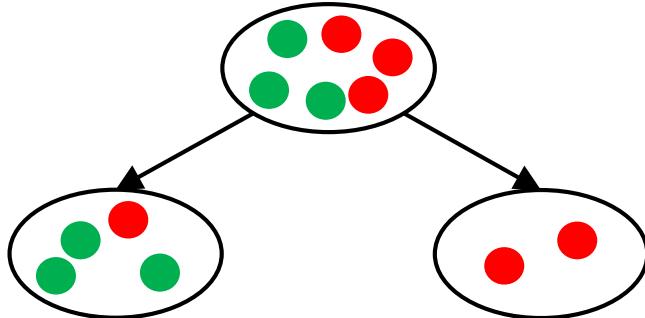


$$E_w = 2/6 \times 1 = 0.3333$$

$$IG = 0.6666$$

Information gain ratio

$$GR(d, \mathcal{D}) = \frac{IG(d, \mathcal{D})}{-\sum_{l \in levels(d)} (P(d = l) \times \log_2(P(d = l)))}$$

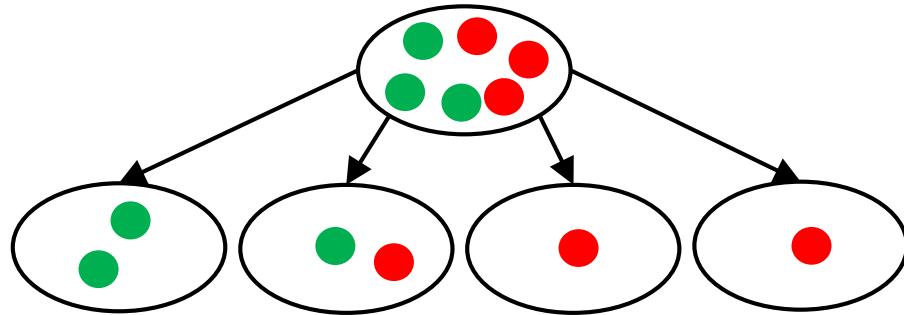


$$(4,2) \Rightarrow E' = 0,91830$$

$$IG = 0.4591$$



$$GR = 0.4591/E' = 0.50$$



$$(2,2,1,1) \Rightarrow E' = 1,91830$$

$$IG = 0.6666$$



$$GR = 0.6666/E' = 0.34$$



Gini index

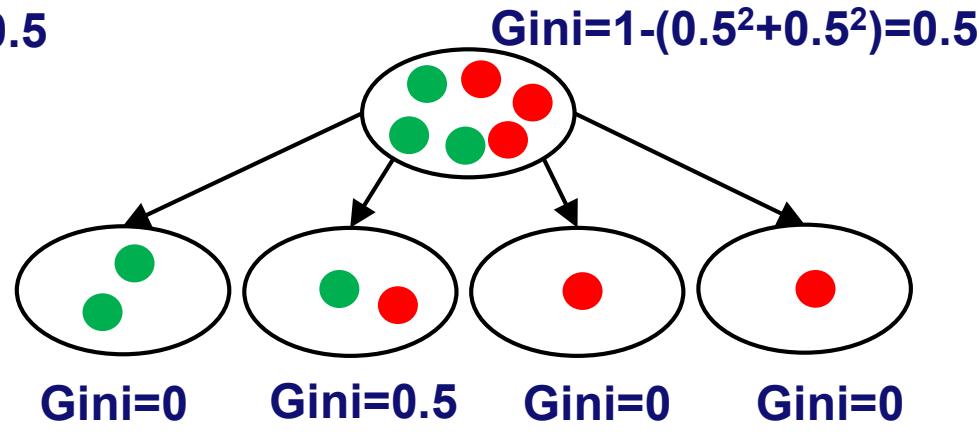
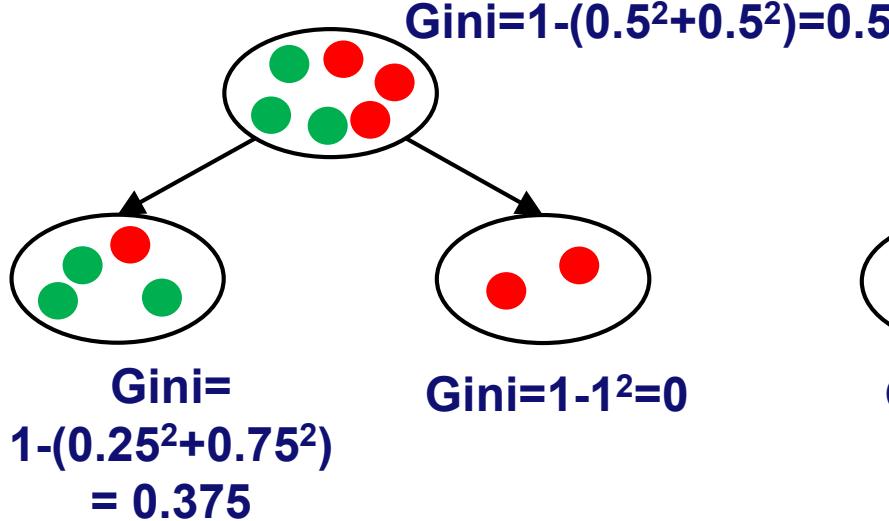
- Alternative measure of impurity
- Expected misclassification rate when guessing based on observed distribution

$$Gini(t, \mathcal{D}) = 1 - \sum_{l \in levels(t)} P(t = l)^2$$

- With probability $P(t=l)$ we guess $t=l$ and with probability $P(t=l)$ this is right.
- Gini index can be seen as the probability of guessing the wrong label.

Gini index

$$Gini(t, \mathcal{D}) = 1 - \sum_{l \in levels(t)} P(t = l)^2$$



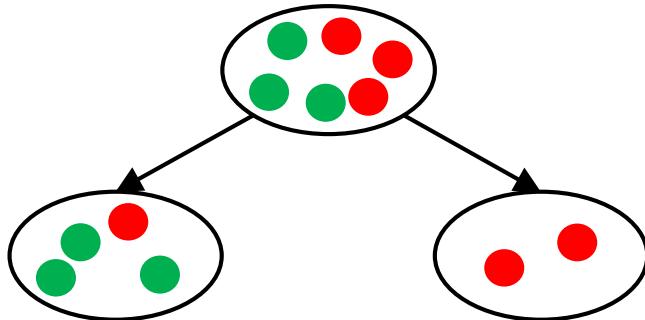
$$Gini_w = \frac{4}{6} \times 0.375 = 0.25$$

$$IG_{Gini} = 0.5 - 0.25 = 0.25$$

$$Gini_w = \frac{2}{6} \times 0.5 = 0.1666$$

$$IG_{Gini} = 0.5 - 0.166 = 0.33$$

Comparison



IG = 0.4591

normal information gain

GR = 0.50

information gain ratio

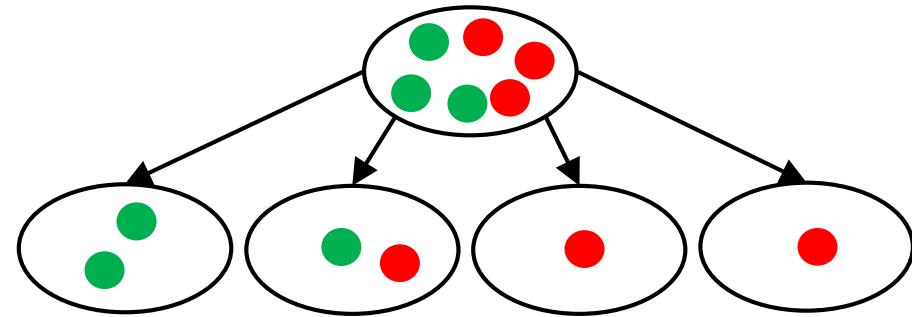
IG_{Gini} = 0.25

Gini-based information gain ratio

IG = 0.6666

GR = 0.34

IG_{Gini} = 0.33



There is not “a best one”: Compare and interpret results!

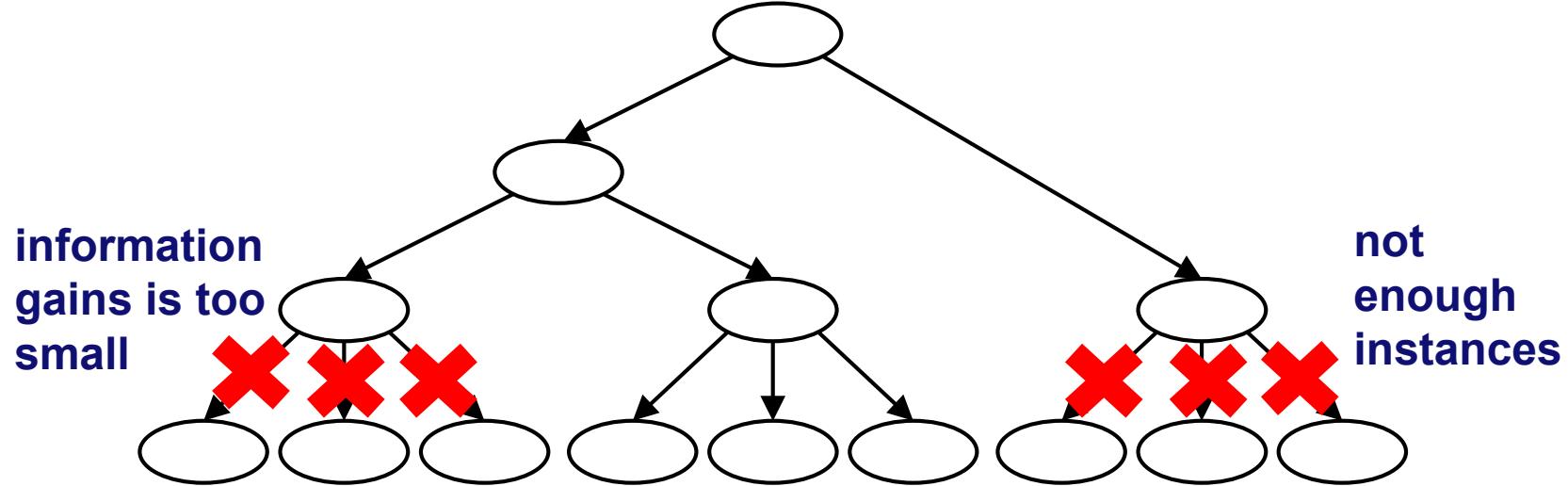
修剪 Pruning decision trees

- Possible problems:
 - Decision tree is overfitting the data.
 - Decision tree is too complex / deep.
- Two solution directions:
 - Pre-pruning (early stopping/forward)
 - Post-pruning (reduced error/backward)

Pre-pruning

- Stop creating subtrees and use a majority vote to determine label.
- Many possible stopping criteria:
 - Lower bound for number of instances.
 - Lower bound for information gain.
- May create trees that are not consistent with respect to the data.
- To generalize and avoid overfitting.

Pre-pruning



Efficient, but one may miss strong dependencies at lower levels of tree.

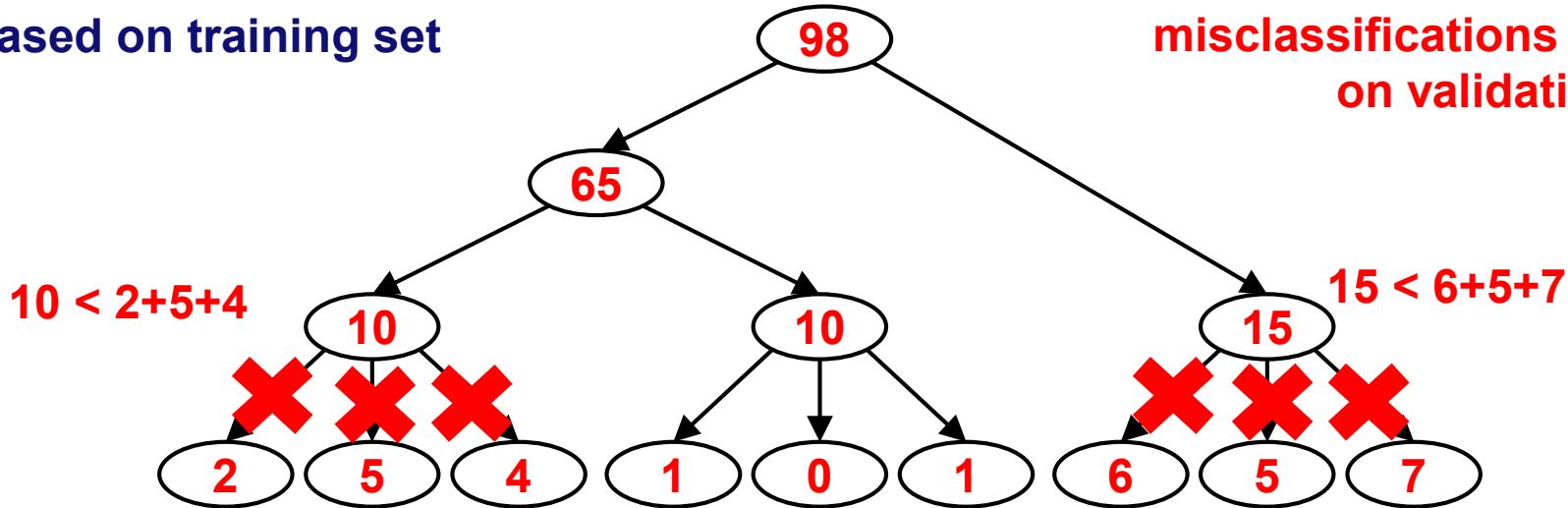
Post-pruning

- First build the whole decision tree and then cut-off branches that do not add much.
- Common approach: split data in training set and validation/test set.
- Measure “performance of splits” based on validation/test set.

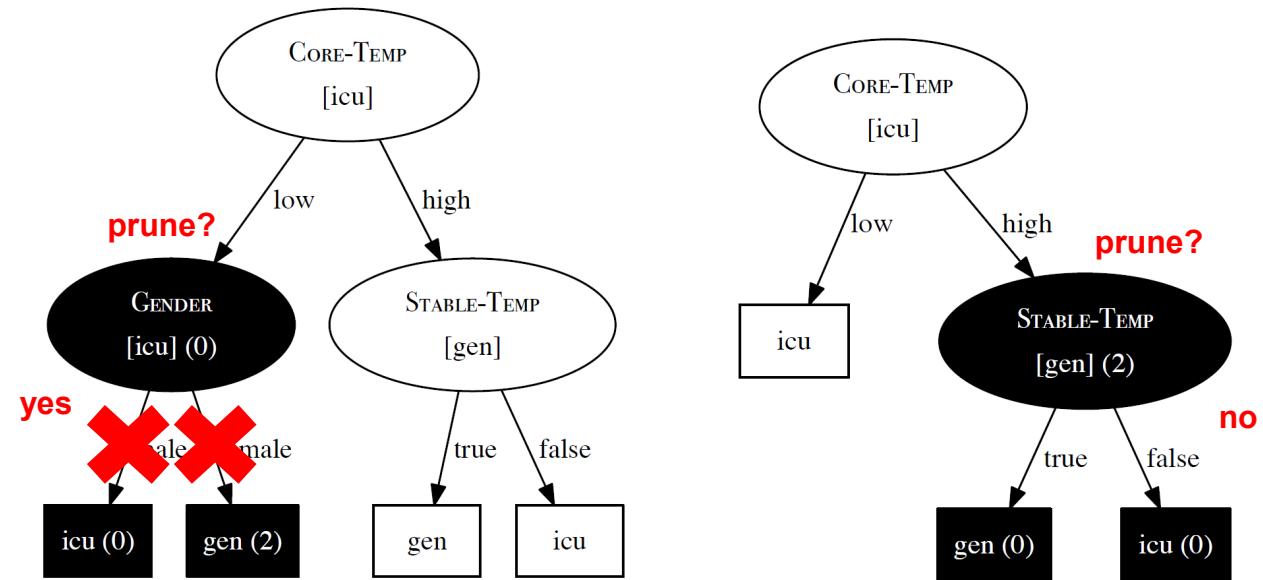
Post-pruning

Tree was learned
based on training set

Numbers indicate
misclassifications based
on validation set



Example in book



Numbers indicate misclassifications based on validation set

Ensembles

- Rather than creating a single decision tree, we aim to create a set of trees (called a model **ensemble**).
- Models should complement each other.
- Different models can “vote” on the label (votes may be weighted).

Ensembles: Main idea

- Related concepts “Wisdom of the crowds” and “Avoiding overfitting”.
- Multiple trees may give different answers (select the most frequent value or the average).
- Many variations of the same idea.

Ensemble: Boosting

(correct iteratively)

- Iteratively change the data set based on misclassifications.
- Instances that are wrongly classified get a higher weight when learning the next model.

Ensemble: Bagging

(split data upfront)

- **Each model is based on a random sample of the data set.**
- **Idea: Avoids models depending on the specific sample in the data set (learning decision trees may be very sensitive to small variations).**
- **Many variants (e.g., remove some instances and duplicate others)**

Ensemble: Bagging

f1	f2	f3	f4	...	fn	class
						high
			X			high
			X			low
						medium
						high
						low

f1	f2	f3	f4	...	fn	class
			X			high
						high
			X			low
						medium
						high
						low

f1	f2	f3	f4	...	fn	class
						high
						high
						low
			X			medium
			X			high
						low

f1	f2	f3	f4	...	fn	class
						high
						high
			X			low
						medium
						high
						low

Ensemble: Subspace sampling

- **Each model is based on a random set of descriptive features.**
- **Idea: Learning process is faster and less likely to be overfitting when focusing on just a few features.**

Ensemble: Subspace sampling

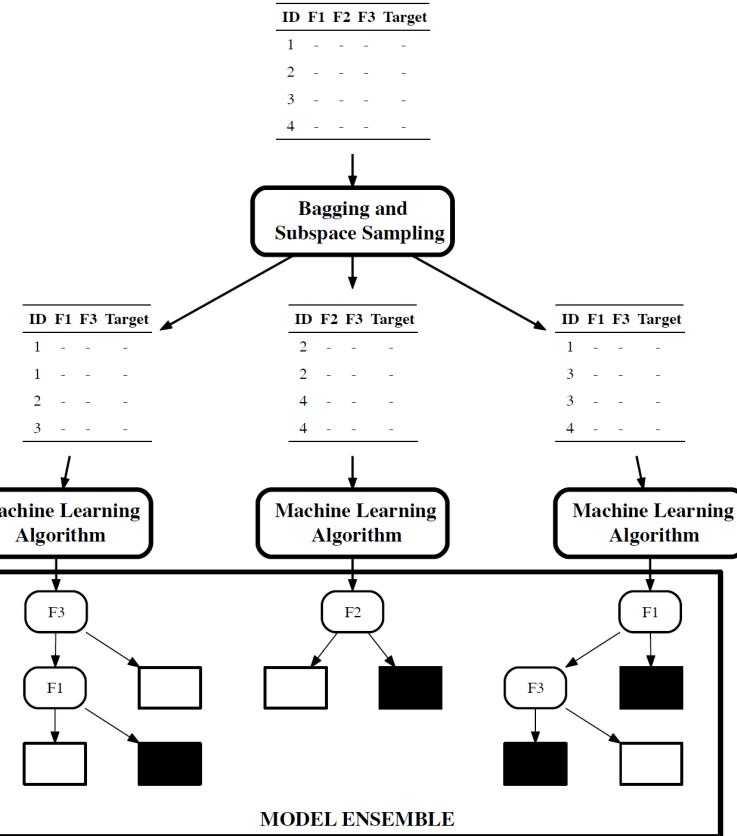
f1	✗	✗	f4	...	fn	class
						high
						high
						low
						medium
						high
						low

✗	f2	f3	✗	...	fn	class
						high
						high
						low
						medium
						high
						low

f1	f2	f3	✗	✗	fn	class
						high
						high
						low
						medium
						high
						low

f1	f2	✗	f4	...	✗	class
						high
						high
						low
						medium
						high
						low

Ensemble: Random forest



- Combination of bagging and subspace sampling.

Ensemble: Random forest

f1	f2	f3	f4	...	fn	class
						high
				X		high
						low
						medium
						high
						low

	f2	f3	f4	...	fn	class
X						high
						high
			X			low
						medium
						high
						low

f1	f2	f3	f4	...	fn	class
						high
						high
						low
			X			medium
			X			high
						low

f1	f2	f3	f4	...	fn	class
						high
						high
						low
			X			medium
						high
						low

Dealing with continuous variables



Dealing with continuous variables

instances

continuous descriptive features			continuous target feature		
f1	f2	f3	...	fn	class
high	true	gold	88	59.99	5043
high	false	gold	76	50.00	4598
low	false	silver	32	39.50	3248
low	true	silver	89	49.99	5466
high	true	gold	21	59.99	7682
low	true	gold	45	29.99	4325

Thus far we assumed features were categorical.

Recall that one can use binning to make continuous features categorical.



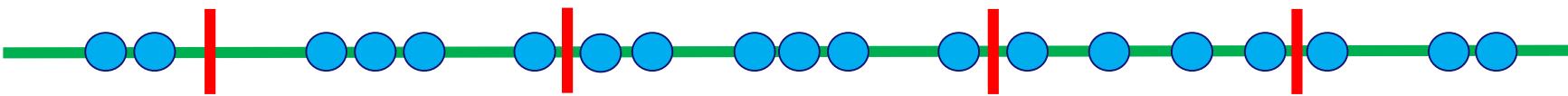
Chair of Process
and Data Science

Continuous descriptive features

f1	f2	f3	f4	...	fn	class
high	true	gold	88		59.99	yes
high	false	gold	76		50.00	no
low	false	silver	32		39.50	no
low	true	silver	89		49.99	yes
high	true	gold	21		59.99	no
		gold	45		29.99	yes

continuous
descriptive features

Continuous descriptive features



- Challenge: determine suitable boundaries.
- Infinite number of thresholds is possible.
- Idea: Sort instances based on the continuous feature and look for changes in class labels.
- Change points are candidate thresholds.
- Select threshold with the highest information gain.

Example from book

ID	STREAM	SLOPE	ELEVATION	VEGETATION
1	false	steep	3 900	chapparal
2	true	moderate	continuous descriptive feature	riparian
3	true	steep	1 500	riparian
4	false	steep	1 200	chapparal
5	false	flat	4 450	conifer
6	true	steep	5 000	conifer
7	true	steep	3 000	chapparal

sort 

Example taken from Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

Example from book

ID	STREAM	SLOPE	ELEVATION	VEGETATION
1	false	steep	3 900	chapparal
2	true	moderate	300	riparian
3	true	steep	1 500	riparian
4	false	steep	1 200	chapparal
5	false	flat	4 450	conifer
6	true	steep	5 000	conifer
7	true	steep	3 000	chapparal

Take middle values of continuous feature in between changed target features

ID	STREAM	SLOPE	ELEVATION	VEGETATION
2	true	moderate	300	riparian
4	false	steep	1 200	chapparal
3	true	steep	1 500	riparian
7	true	steep	3 000	chapparal
1	false	steep	3 900	chapparal
5	false	flat	4 450	conifer
6	true	steep	5 000	conifer

Example from book

ID	STREAM	SLOPE	ELEVATION	VEGETATION
2	true	moderate	300	riparian
4	false	steep	750	chapparal
3	true	steep	1350	riparian
7	true	steep	2250	chapparal
1	false	steep	3000	chapparal
5	false	flat	3900	chapparal
6	true	steep	4450	conifer
			4175	conifer

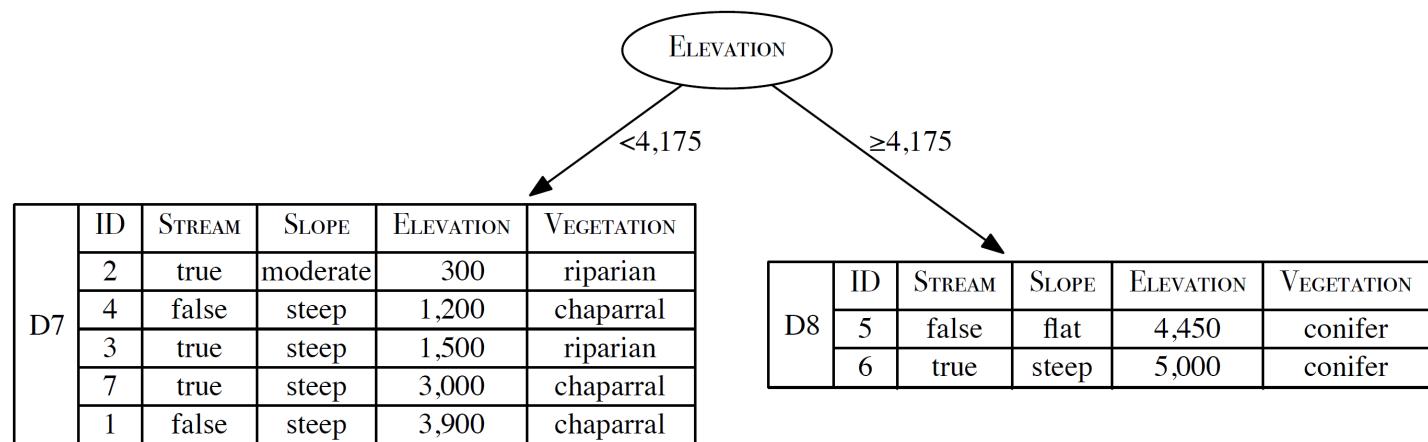
Compute information gain:
business as usual

Split by Threshold	Part.	Instances	Partition Entropy	Rem.	Info. Gain
≥ 750	\mathcal{D}_1	d_2	0.0	1.2507	0.3060
	\mathcal{D}_2	$d_4, d_3, d_7, d_1, d_5, d_6$	1.4591		
≥ 1350	\mathcal{D}_3	d_2, d_4	1.0	1.3728	0.1839
	\mathcal{D}_4	d_3, d_7, d_1, d_5, d_6	1.5219		
≥ 2250	\mathcal{D}_5	d_2, d_4, d_3	0.9183	0.9650	0.5917
	\mathcal{D}_6	d_7, d_1, d_5, d_6	1.0		
≥ 4175	\mathcal{D}_7	d_2, d_4, d_3, d_7, d_1	0.9710	0.6935	0.8631
	\mathcal{D}_8	d_5, d_6	0.0		



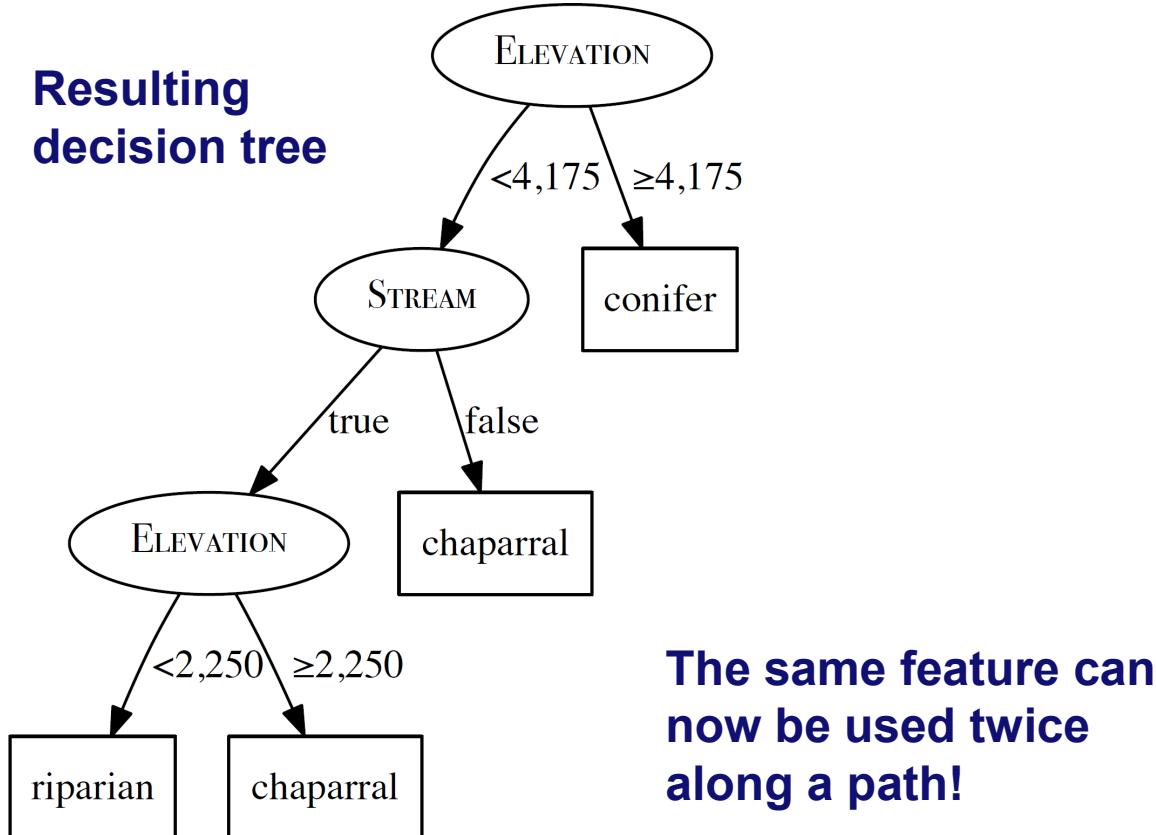
Example from book

ID	STREAM	SLOPE	ELEVATION	VEGETATION
2	true	moderate	300	riparian
4	false	steep	1,200	chaparral
3	true	steep	1,500	riparian
7	true	steep	3,000	chaparral
1	false	steep	3,900	chaparral
5	false	flat	4,450	conifer
6	true	steep	5,000	conifer



Example from book

Resulting
decision tree



The same feature can
now be used twice
along a path!

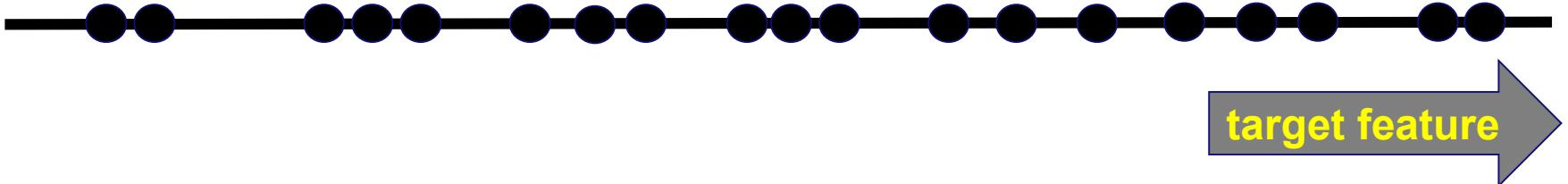
Continuous target features

f1	f2	f3	f4	...	fn	class
high	true	gold	blue		cloudy	5043
high	false	gold	red		rain	4598
low	false	silver	green		rain	3248
low	true	silver	green		sun	5466
high	true	gold	blue		cloudy	7682
low	true	gold	blue		sun	4325

continuous
target feature

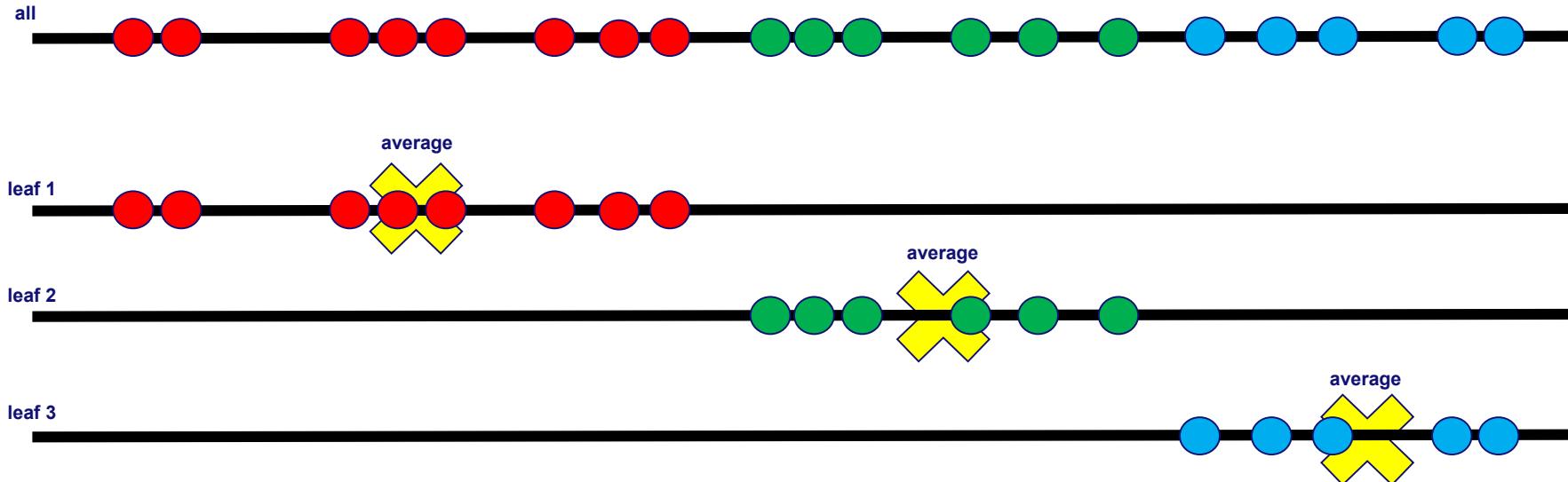


Continuous target features



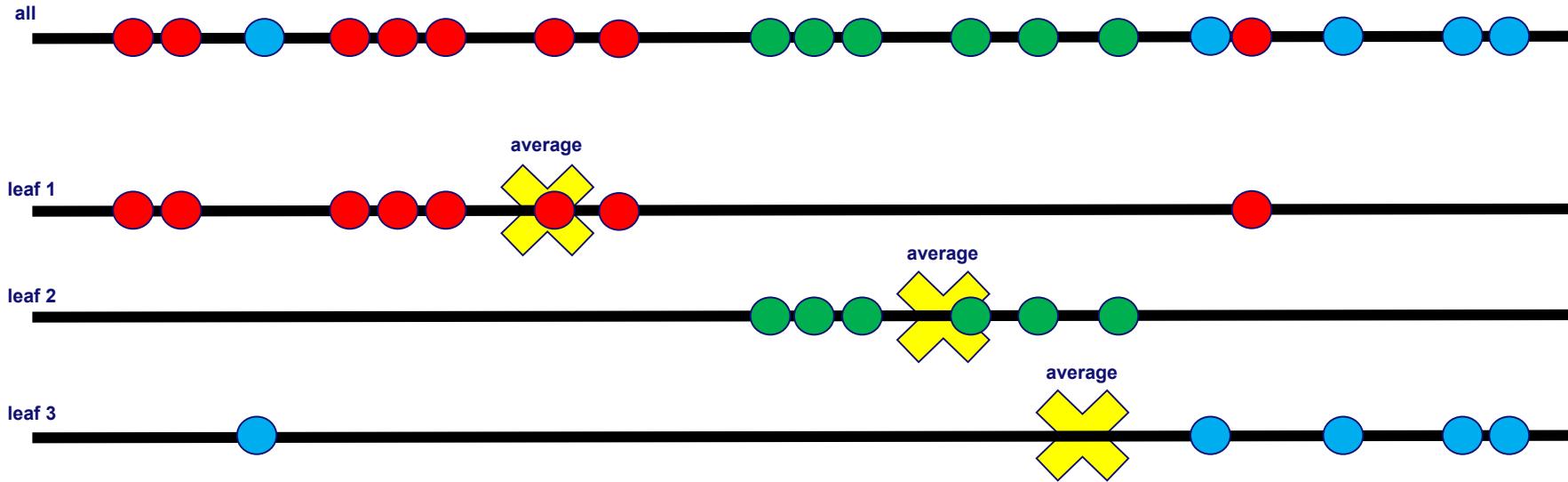
- We want to find descriptive features that “nicely” partition the target feature axis.
- Impurity = variance within a partition.
- We cannot use the target feature itself!
- Lets “color the dots” based on a selected descriptive feature.

Good classification



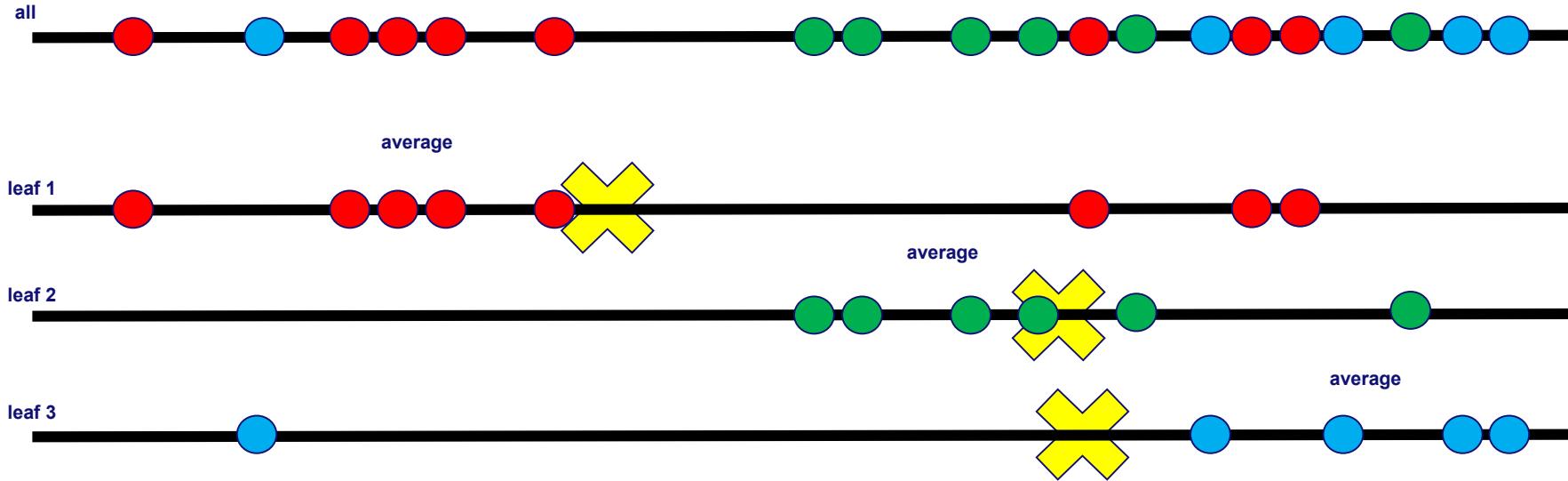
- Three leaves (red, green, blue show mapping).
- As before we need a measure of quality: impurity.
- Variance within a leaf of the decision tree.

Reasonable classification



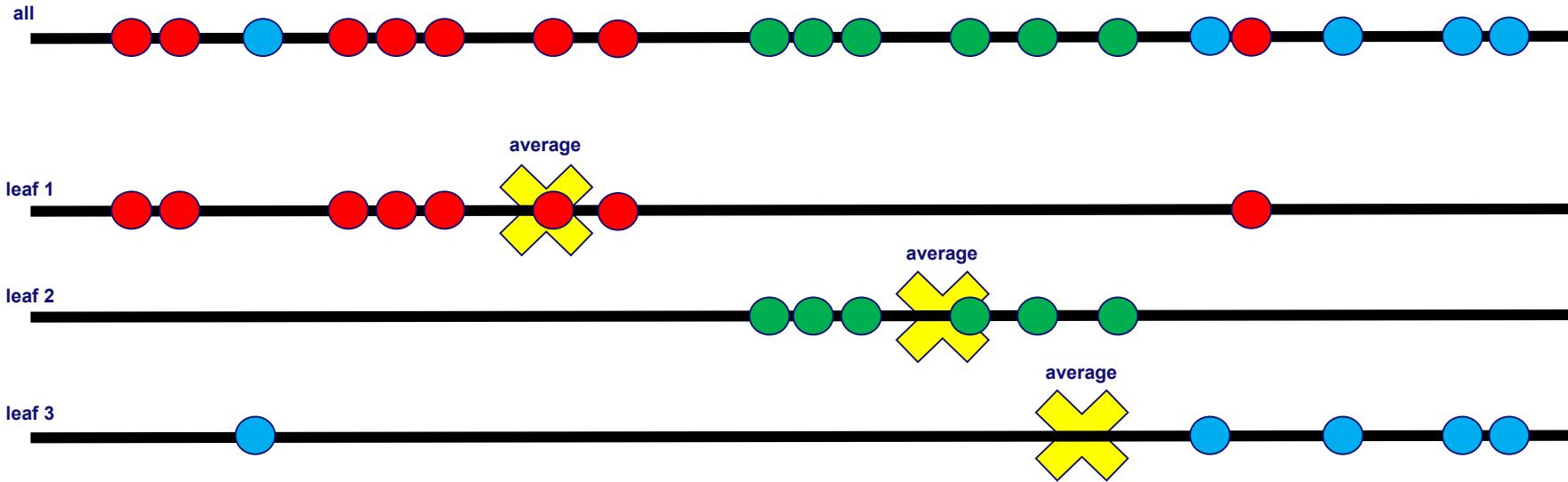
- Variance within leaf 1 and leaf 3 increased with respect to the “good classification”.

Poor classification



- Variance within all leaves is high compared to the “good classification”.

Impurity = variance in node/leaf



$$\text{var}(t, \mathcal{D}) = \frac{\sum_{i=1}^n (t_i - \bar{t})^2}{n - 1}$$

Adapting the ID3 algorithm: Basic idea

Algorithm 4.1 Pseudocode description of the ID3 algorithm.

Require: set of descriptive features \mathbf{d}

Require: set of training instances \mathcal{D}

1: **if** all the instances in \mathcal{D}

2: **return** a decision tree

3: **else if** \mathbf{d} is empty **then**

4: **return** a decision tree with the

majority target level

5: **else if** \mathcal{D} is empty **then**

6: **return** a decision tree consisting
 of a node with the label of the
 majority target level of the dataset

7: **else**

8: $\mathbf{d}[\text{best}] \leftarrow \arg \max_{d \in \mathbf{d}} IG(d, \mathcal{D})$

9: make a new node, $Node_{\mathbf{d}[\text{best}]}$, and label it with $\mathbf{d}[\text{best}]$

10: partition \mathcal{D} using $\mathbf{d}[\text{best}]$

11: remove $\mathbf{d}[\text{best}]$ from \mathbf{d}

12: **for** each partition \mathcal{D}_i of \mathcal{D} **do**

13: grow a branch from $Node_{\mathbf{d}[\text{best}]}$ to the decision tree created by rerunning ID3 with $\mathcal{D} = \mathcal{D}_i$

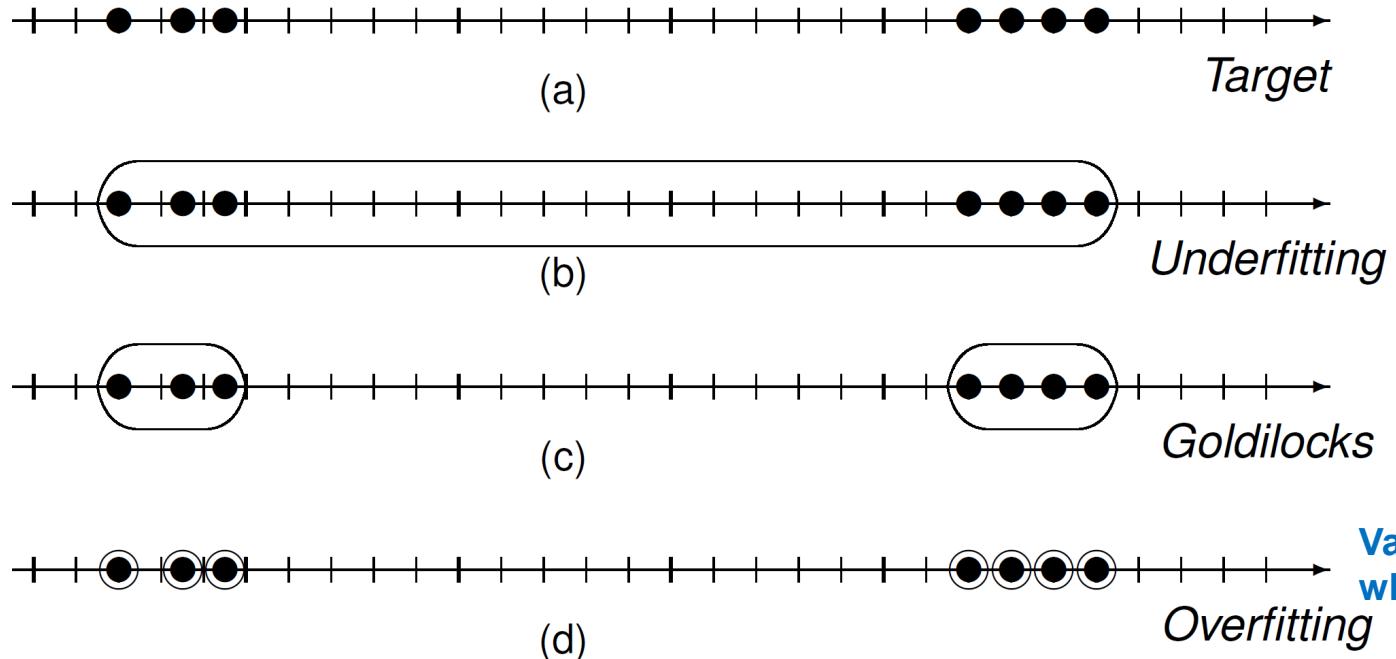
$$\mathbf{d}[\text{best}] = \operatorname{argmin}_{d \in \mathbf{d}} \sum_{l \in \text{levels}(d)} \frac{|\mathcal{D}_{d=l}|}{|\mathcal{D}|} \times \text{var}(t, \mathcal{D}_{d=l})$$



leaf node with the label of the
immediate parent node

**Split based on the feature that
lowers the weighted variance
within the subtrees as much
as possible.**

Overfitting and underfitting



Avoid overfitting by stopping early enough!

Example from book

ID	SEASON	WORK DAY	RENTALS	ID	SEASON	WORK DAY	RENTALS
1	winter	false	800	7	summer	false	3 000
2	winter	false	826	8	summer	true	5 800
3	winter	true	900	9	summer	true	6 200
4	spring	false	2 100	10	autumn	false	2 910
5	spring	true	4 740	11	autumn	false	2 880
6	spring	true	4 900	12	autumn	true	2 820

Target feature = bike rentals per day

Split by Feature	Level	Part.	Instances	$\frac{ \mathcal{D}_{d=1} }{ \mathcal{D} }$	$\text{var}(t, \mathcal{D})$	Weighted Variance
SEASON	'winter'	\mathcal{D}_1	$\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$	0.25	2 692	
	'spring'	\mathcal{D}_2	$\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$	0.25	$2\ 472\ 533\frac{1}{3}$	
	'summer'	\mathcal{D}_3	$\mathbf{d}_7, \mathbf{d}_8, \mathbf{d}_9$	0.25	3 040 000	$1\ 379\ 331\frac{1}{3}$
	'autumn'	\mathcal{D}_4	$\mathbf{d}_{10}, \mathbf{d}_{11}, \mathbf{d}_{12}$	0.25	2 100	
WORK DAY	'true'	\mathcal{D}_5	$\mathbf{d}_3, \mathbf{d}_5, \mathbf{d}_6, \mathbf{d}_8, \mathbf{d}_9, \mathbf{d}_{12}$	0.50	$4\ 026\ 346\frac{1}{3}$	
	'false'	\mathcal{D}_6	$\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_7, \mathbf{d}_{10}, \mathbf{d}_{11}$	0.50	1 077 280	$2\ 551\ 813\frac{1}{3}$

best split, lowers variance most



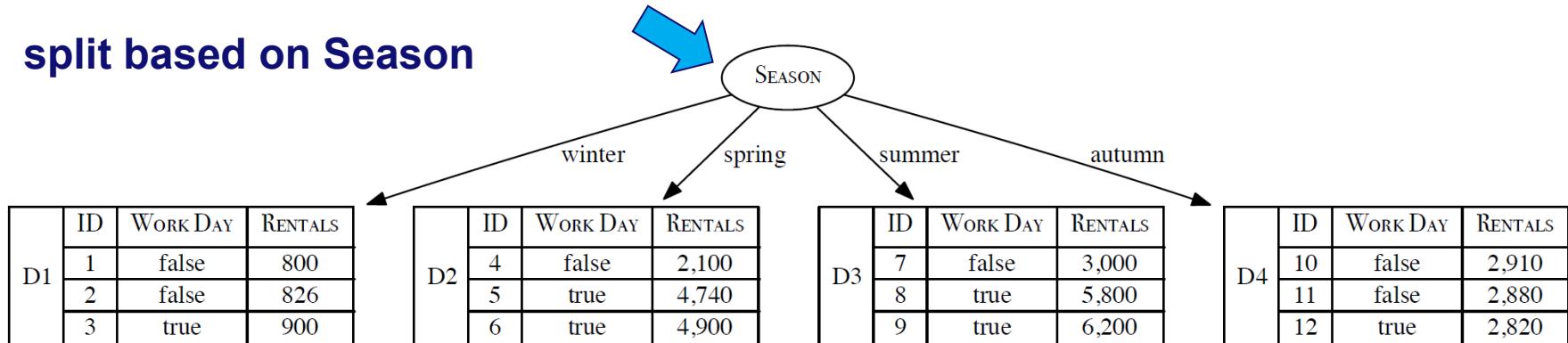

Example from book

ID	SEASON	WORK DAY	RENTALS
1	winter	false	800
2	winter	false	826
3	winter	true	900
4	spring	false	2 100
5	spring	true	4 740
6	spring	true	4 900

ID	SEASON	WORK DAY	RENTALS
7	summer	false	3 000
8	summer	true	5 800
9	summer	true	6 200
10	autumn	false	2 910
11	autumn	false	2 880
12	autumn	true	2 820

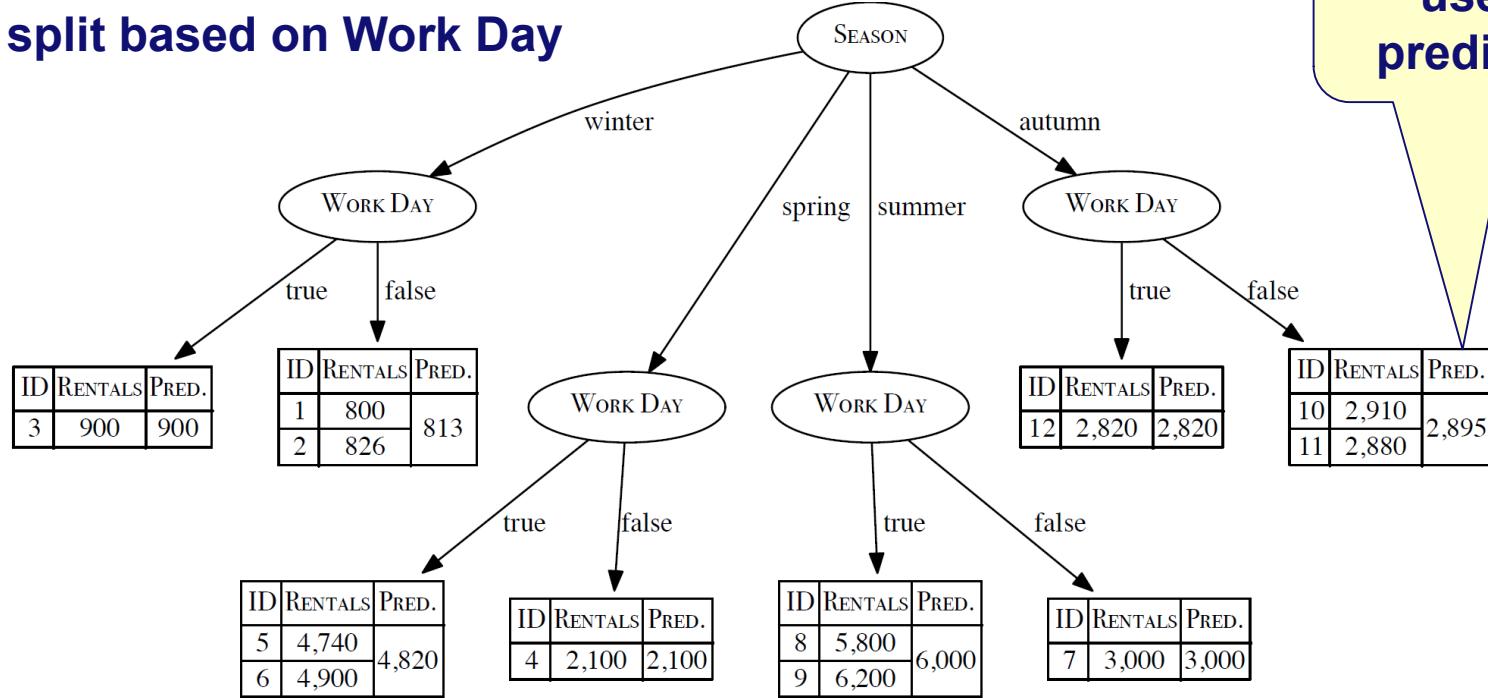
Split by Feature	Level	Part.	Instances	$\frac{ \mathcal{D}_{d=l} }{ \mathcal{D} }$	$\text{var}(t, \mathcal{D})$	Weighted Variance
				$ \mathcal{D} $		
SEASON	'winter'	\mathcal{D}_1	d_1, d_2, d_3	0.25	2 692	
	'spring'	\mathcal{D}_2	d_4, d_5, d_6	0.25	2 472 533 $\frac{1}{3}$	1 379 331 $\frac{1}{3}$
	'summer'	\mathcal{D}_3	d_7, d_8, d_9	0.25	3 040 000	
	'autumn'	\mathcal{D}_4	d_{10}, d_{11}, d_{12}	0.25	2 100	
WORK DAY	'true'	\mathcal{D}_5	$d_3, d_5, d_6, d_8, d_9, d_{12}$	0.50	4 026 346 $\frac{1}{3}$	
	'false'	\mathcal{D}_6	$d_1, d_2, d_4, d_7, d_{10}, d_{11}$	0.50	1 077 280	2 551 813 $\frac{1}{3}$

split based on Season



Example from book

split based on Work Day



average of leaf is
used as the
predicted value

Many combinations of ideas are possible!

- **ID3**: The original.
- **C4.5**: Dealing with continuous features, missing values, post-pruning, etc.
- **C5.0**: Further development of C4.5 by Ross Quinlan.
- **J48**: Open source implementation of C4.5 (used in WEKA).
- **CART (Classification and Regression Trees)**: Uses Gini index, handles continuous features, and can deal with missing value.
- **CAID (Chi-square Automatic Interaction Detector)**: Relies on Chi-square tests to determine the best next split at each step.

Conclusion

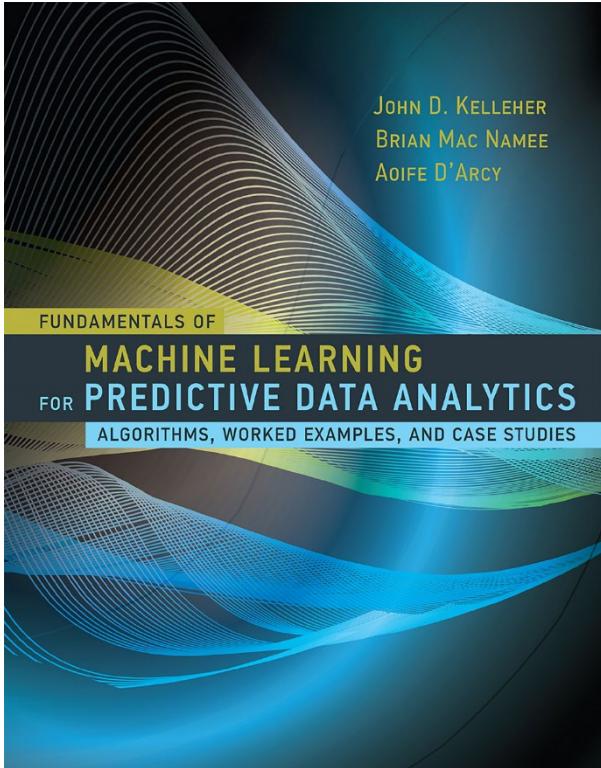


Short summary of lecture

- **Supervised learning:** Explain the target feature in terms of descriptive features.
- **Decision trees** are easy to understand.
- Focus on **categorical** variables, but extensions to **continuous** variables possible.
- **Many variations** are possible based on the basic ID3 algorithm (pruning, ensembles, different purity measures, etc.).



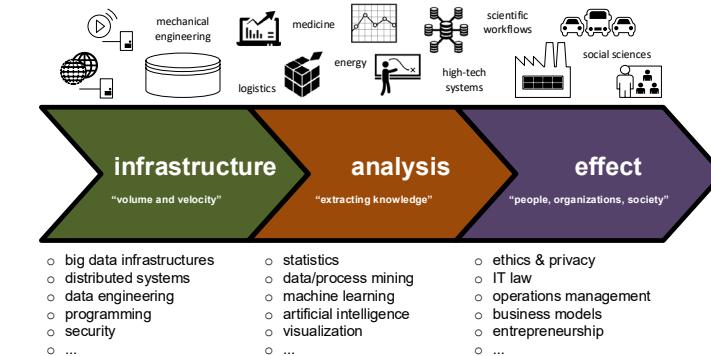
Relevant Literature



- **Chapter 4 of Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.**
- **Also see Chapter 4 of Process Mining - Data Science in Action by Wil van der Aalst, Springer Verlag, 2016.**

Lecture		date	day
Lecture 1	Introduction	13/10/2021	Wednesday
<i>Instruction 1</i>	Python	14/10/2021	Thursday
<i>Instruction 2</i>	Crash Course in Python	15/10/2021	Friday
Lecture 2	Basic data visualization/exploration	20/10/2021	Wednesday
Lecture 3	Decision trees	21/10/2021	Thursday
<i>Instruction 3</i>	Decision trees and data visualization/exploration	22/10/2021	Friday
Lecture 4	Regression	27/10/2021	Wednesday
Lecture 5	Support vector machines	28/10/2021	Thursday
<i>Instruction 4</i>	Regression and support vector machines	29/10/2021	Friday
Lecture 6	Neural networks (1/2)	03/11/2021	Wednesday
Lecture 7	Neural networks (2/2)	04/11/2021	Thursday
Lecture 8	Evaluation of supervised learning problems	10/11/2021	Wednesday
<i>Instruction 5</i>	Neural networks	11/11/2021	Thursday
<i>Instruction 6</i>	Neural networks and evaluation	12/11/2021	Friday
Lecture 9	Clustering	17/11/2021	Wednesday
Lecture 10	Frequent item sets	18/11/2021	Thursday
<i>Instruction 7</i>	Clustering and frequent item sets	19/11/2021	Friday
Lecture 11	Association rules	24/11/2021	Wednesday
Lecture 12	Sequence mining	25/11/2021	Thursday
<i>Instruction 8</i>	Association rules and sequence mining	26/11/2021	Friday
Lecture 13	Process mining (unsupervised)	01/12/2021	Wednesday
Lecture 14	Process mining (supervised)	02/12/2021	Thursday
<i>Instruction 9</i>	Process Mining	03/12/2021	Friday
Lecture 15	Text Mining (1/2)	08/12/2021	Wednesday
Lecture 16	Text Mining (2/2)	09/12/2021	Thursday
<i>Instruction 10</i>	Q&A Assignment 1	10/12/2021	Friday
Lecture 17	Data preprocessing, data quality, binning, etc.	15/12/2021	Wednesday
Lecture 18	Visual analytics & information visualization	16/12/2021	Thursday
<i>Instruction 11</i>	Text Mining	17/12/2021	Friday
Lecture 19	Responsible data science (1/2)	22/12/2021	Wednesday
Lecture 20	Responsible data science (2/2)	23/12/2021	Thursday
Lecture 21	Big data	12/01/2022	Wednesday
<i>Instruction 12</i>	Preprocessing and visualization	13/01/2022	Thursday
<i>Instruction 13</i>	Q&A Assignment 2	14/01/2022	Friday
Lecture 22	Closing	19/01/2022	Wednesday
<i>Instruction 14</i>	Big Data (1/2)	20/01/2022	Thursday
<i>Instruction 15</i>	Responsible data science	21/01/2022	Friday
<i>Instruction 16</i>	Big Data (2/2)	27/01/2022	Thursday
<i>Instruction 17</i>	Example Exam Questions	28/01/2022	Friday
<i>Instruction 18</i>	Questions	02/02/2022	Wednesday

Disclaimer: always check RWTH Moodle for last minute changes



Lecture		date	day
Lecture 1	Introduction	20/10/2021	Wednesday
Lecture 2	Basic data visualization/exploration		
Lecture 3	Decision trees	27/10/2021	Wednesday
Lecture 4	Regression		
Lecture 5	Support vector machines		
Lecture 6	Neural networks (1/2)		
Lecture 7	Neural networks (2/2)	10/11/2021	Wednesday
Lecture 8	Evaluation of supervised learning problems		
Lecture 9	Clustering	17/11/2021	Wednesday
Lecture 10	Frequent item sets		
Lecture 11	Association rules	24/11/2021	Wednesday
Lecture 12	Sequence mining		
Lecture 13	Process mining (unsupervised)		
Lecture 14	Process mining (supervised)	02/12/2021	Thursday
Lecture 15	Text Mining (1/2)		
Lecture 16	Text Mining (2/2)	09/12/2021	Thursday
Lecture 17	Data preprocessing, data quality, binning, etc.		
Lecture 18	Visual analytics & information visualization	16/12/2021	Thursday
Lecture 19	Responsible data science (1/2)		
Lecture 20	Responsible data science (2/2)	23/12/2021	Thursday
Lecture 21	Big data		
Lecture 22	Closing	19/01/2022	Wednesday