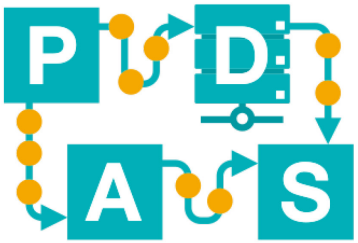
Decision Trees

Lecture 4

IDS-L4

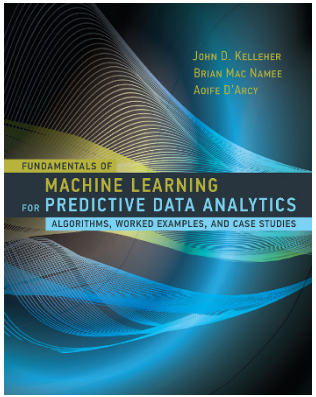


Chair of Process and Data Science

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



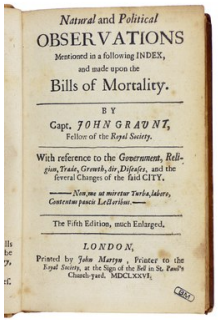
Based on Chapter 4 of Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

Let the data speak

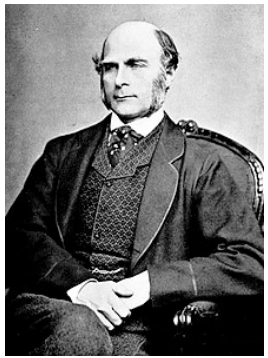
Statistics versus DM / ML (1/2)

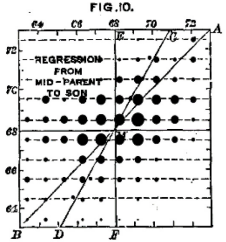
- **Statistics have been around for a while …**

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.

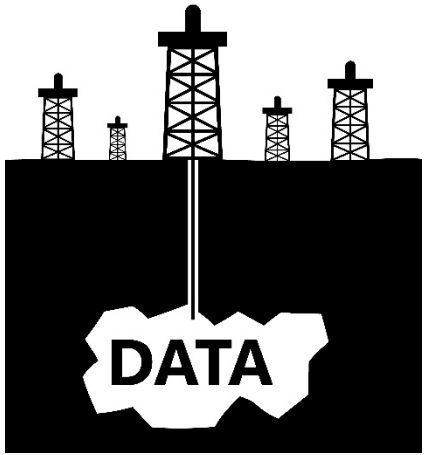


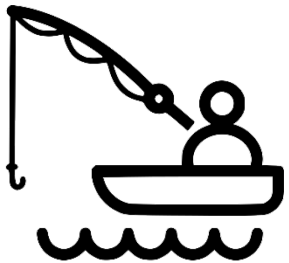


Chair of Process and Data Science

Statistics versus DM / ML (2/2)

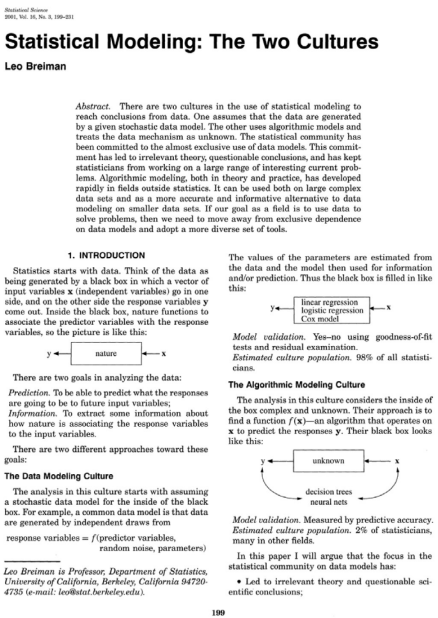
- Statistics have been around for a while …
- However, progress in data science was fueled by more pragmatic approaches (handling large amounts of data), not statistics.
- Major breakthroughs in the discovery of patterns and relationships (e.g., efficiently learning decision trees and association rules),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





Chair of Process and Data Science

Interesting reads



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 the data mechanism as unknown. The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, 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 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** (independent variables) go in one side, and on the other side the response variables **y** come out. Inside the black box, nature functions to associate the predictor variables with the response variables, so the picture is like this:

y ← nature ← x

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 for the inside of the black box. For example, a common data model is that data are generated by independent draws from

response variables = f(predictor variables, random noise, parameters)

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:

y ← linear regression / logistic regression / Cox model ← x

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 $f(\mathbf{x})$—an algorithm that operates on **x** to predict the responses **y**. Their black box looks like this:

y ← unknown ← x
decision trees / neural nets

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:

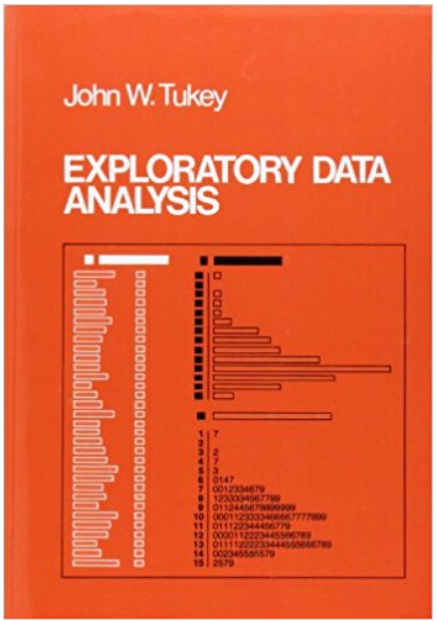
- Led to irrelevant theory and questionable scientific conclusions;

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

199

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

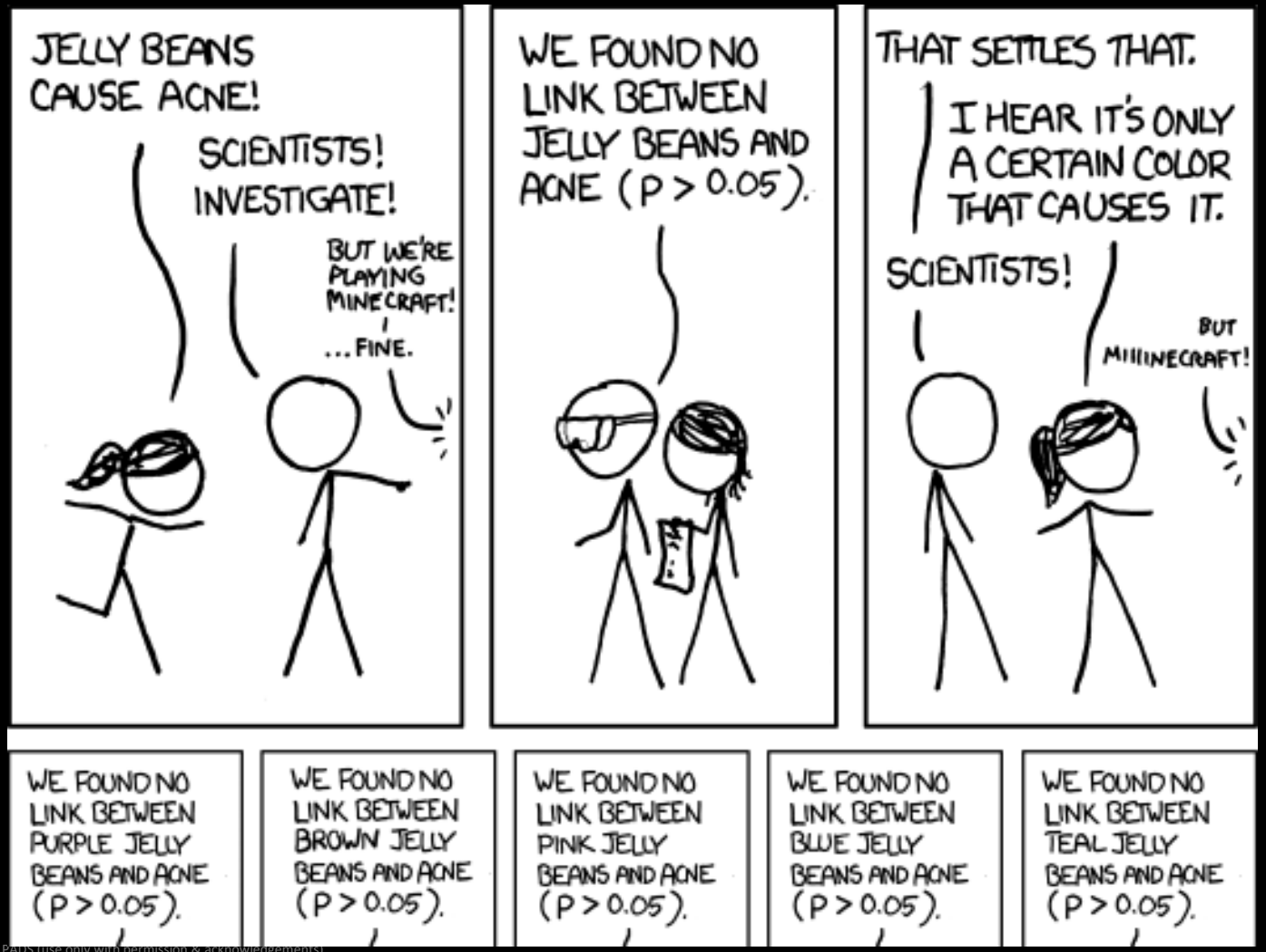
Leo Breiman (1928-2005)



Exploratory Data Analysis. Addison-Wesley Publishing Company, 1977

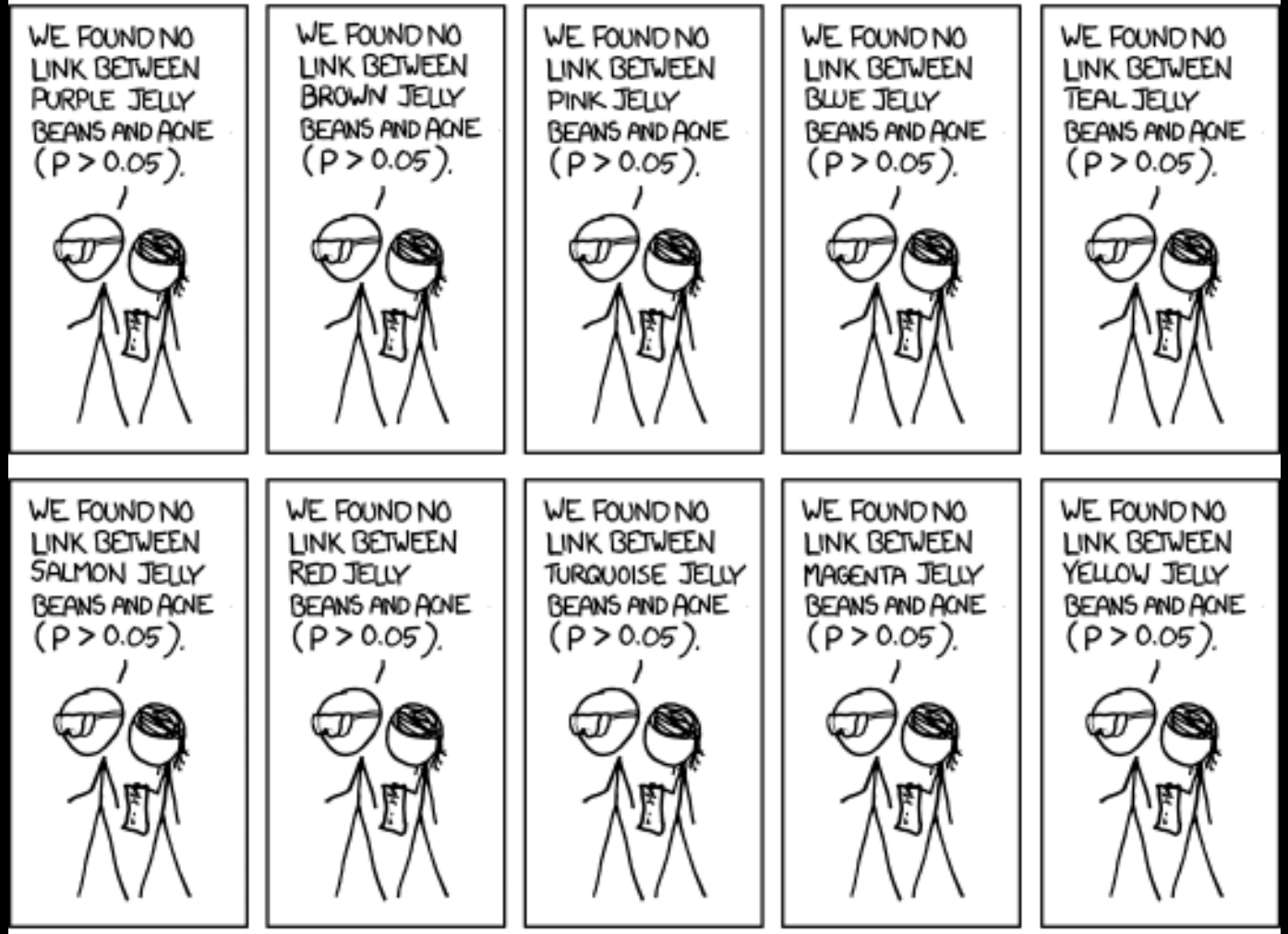
John W. Tukey (1915-2000)

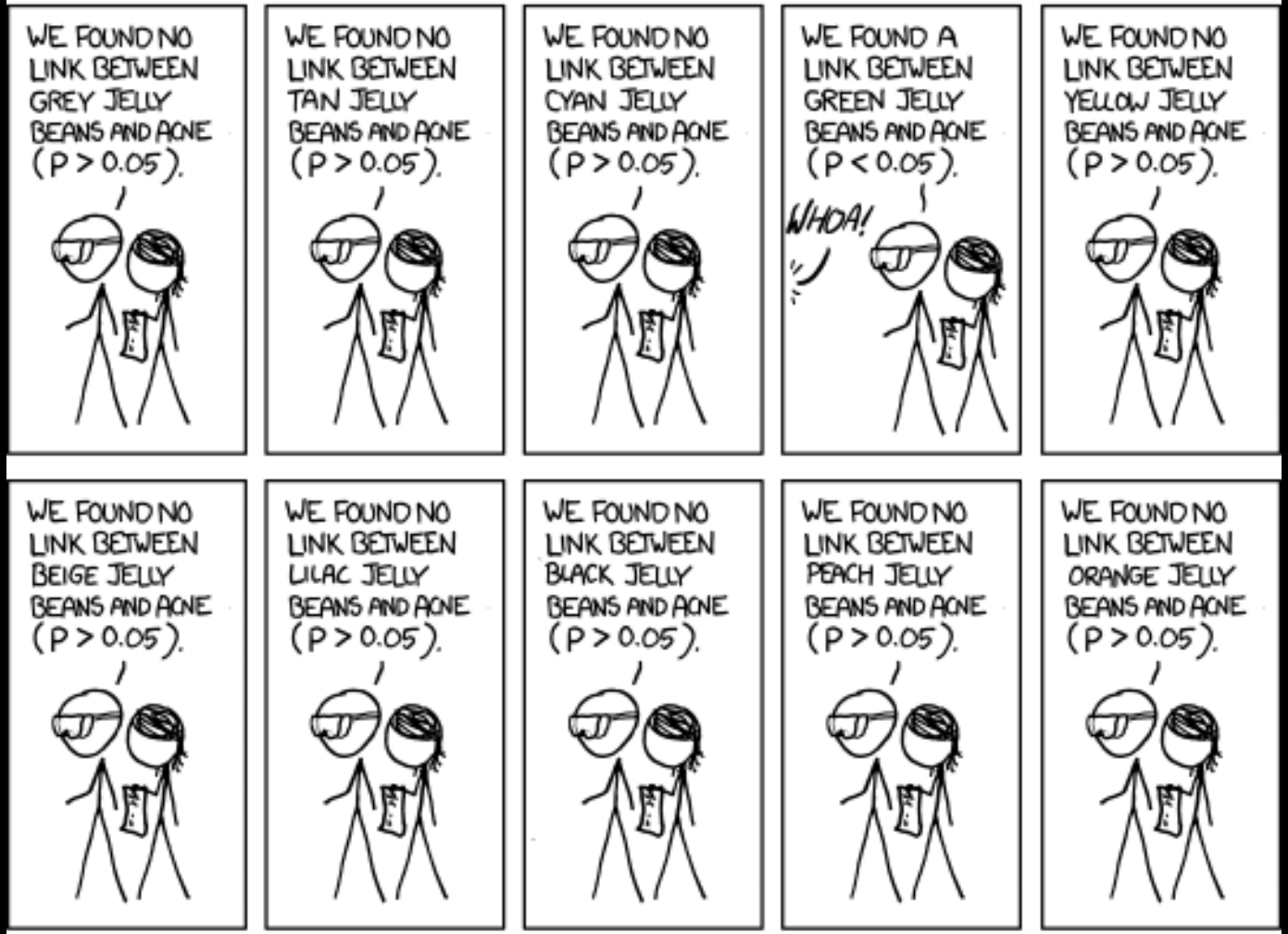
Chair of Process and Data Science

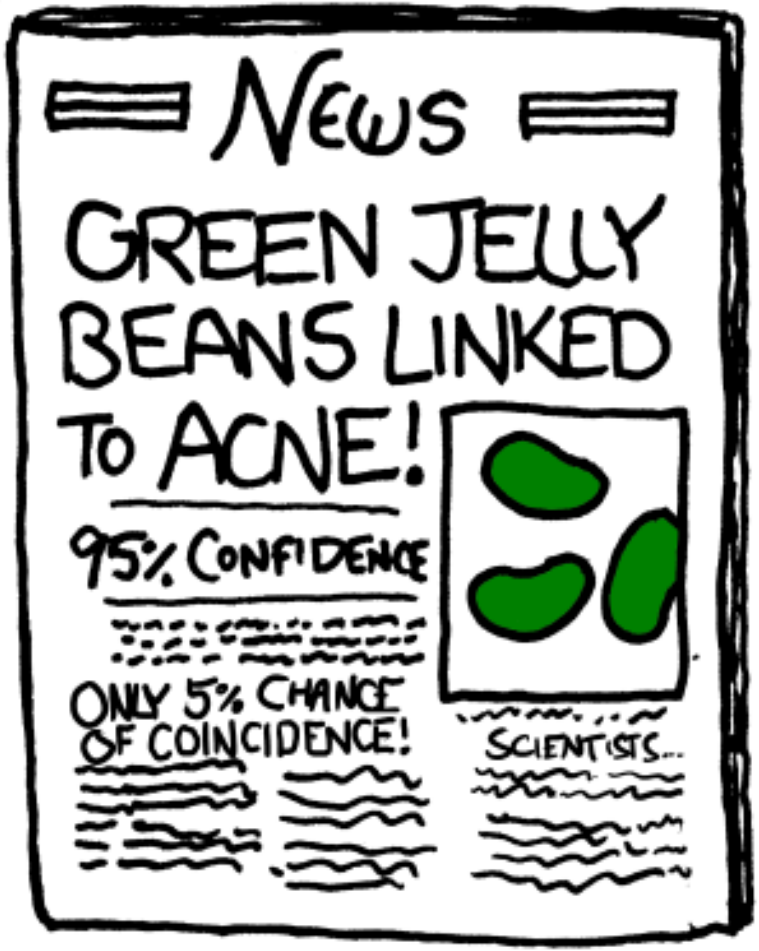


Statistics: Don't torture the data until it confesses

Chair of Process and Data Science







Chair of Process and Data Science

Risks

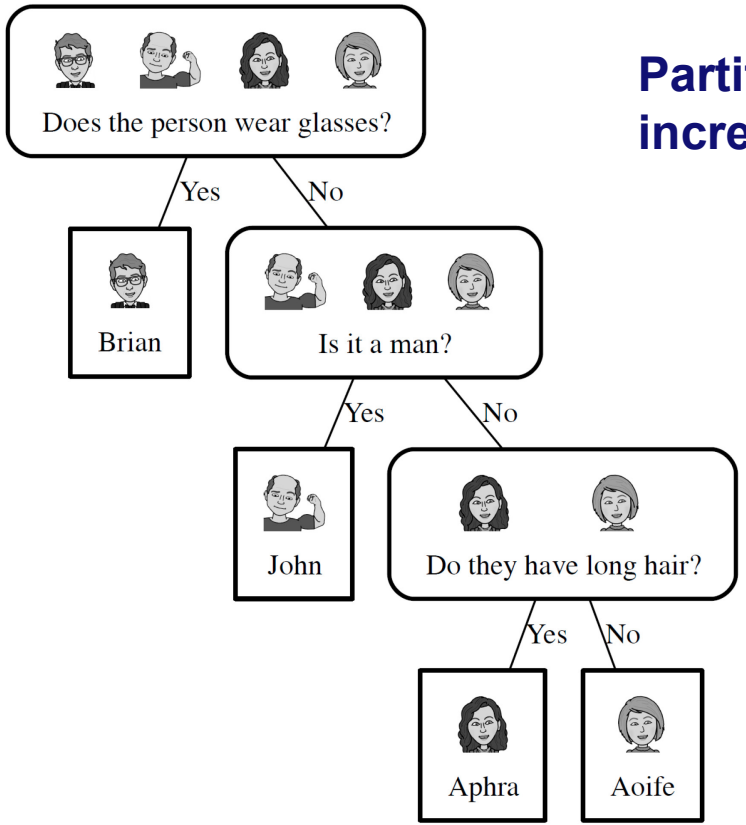
- Testing many hypotheses.
- Overfitting the data.
- Underfitting the data.
- Bias in data.
- Bias in representation.

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

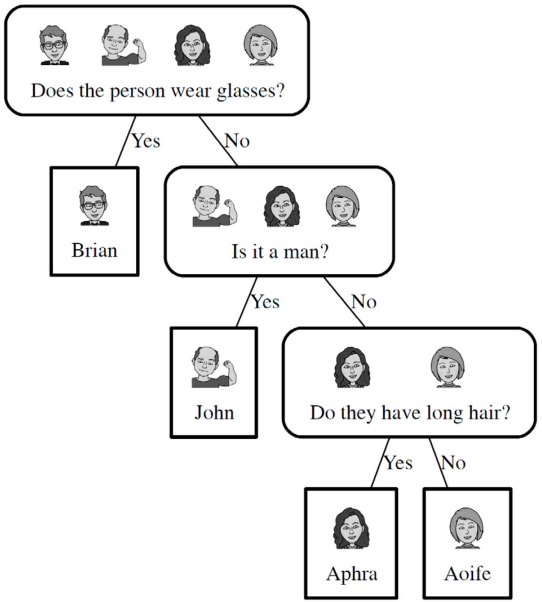


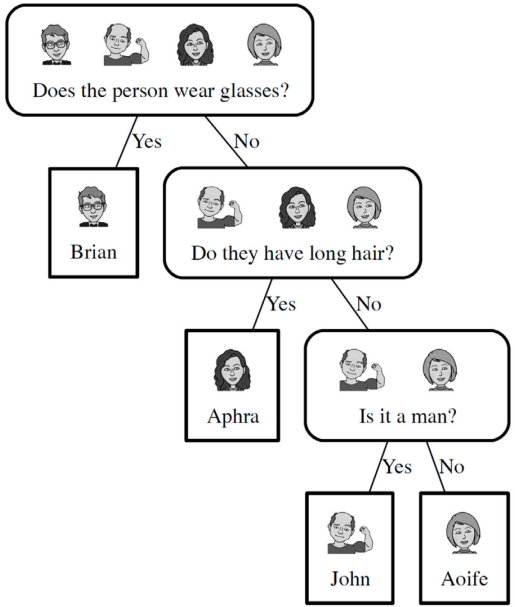
Partitioning instances in increasingly smaller groups

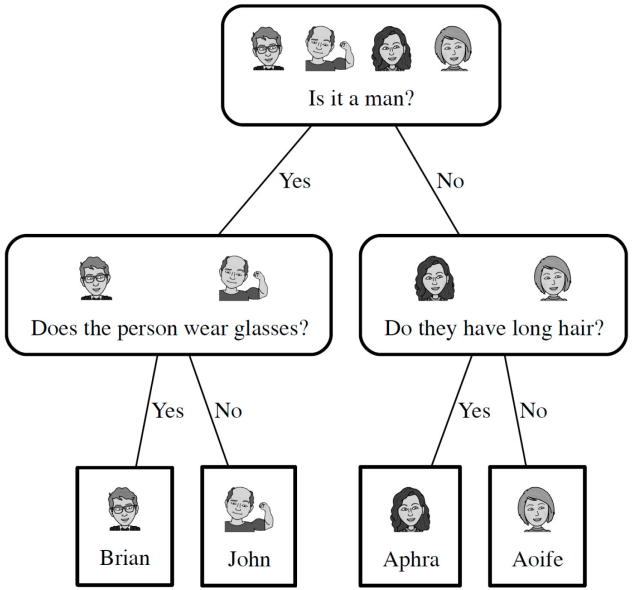
Trees taken from Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

Chair of Process and Data Science

Grouping instances in decision trees



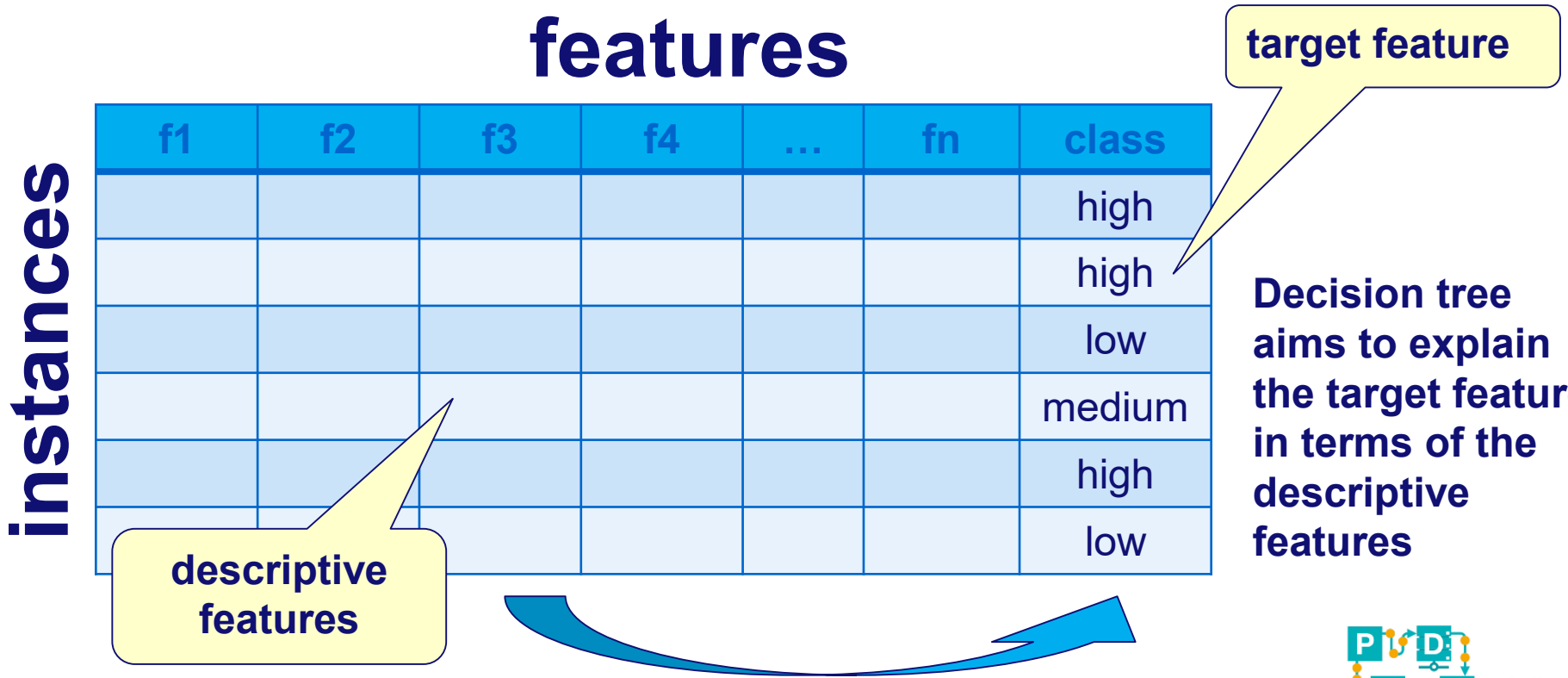




Different trees possible.
Two goals: simple and homogeneous leaves.

Chair of Process and Data Science

Supervised learning



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

Chair of Process and Data Science

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	
…	…	…	

descriptive features

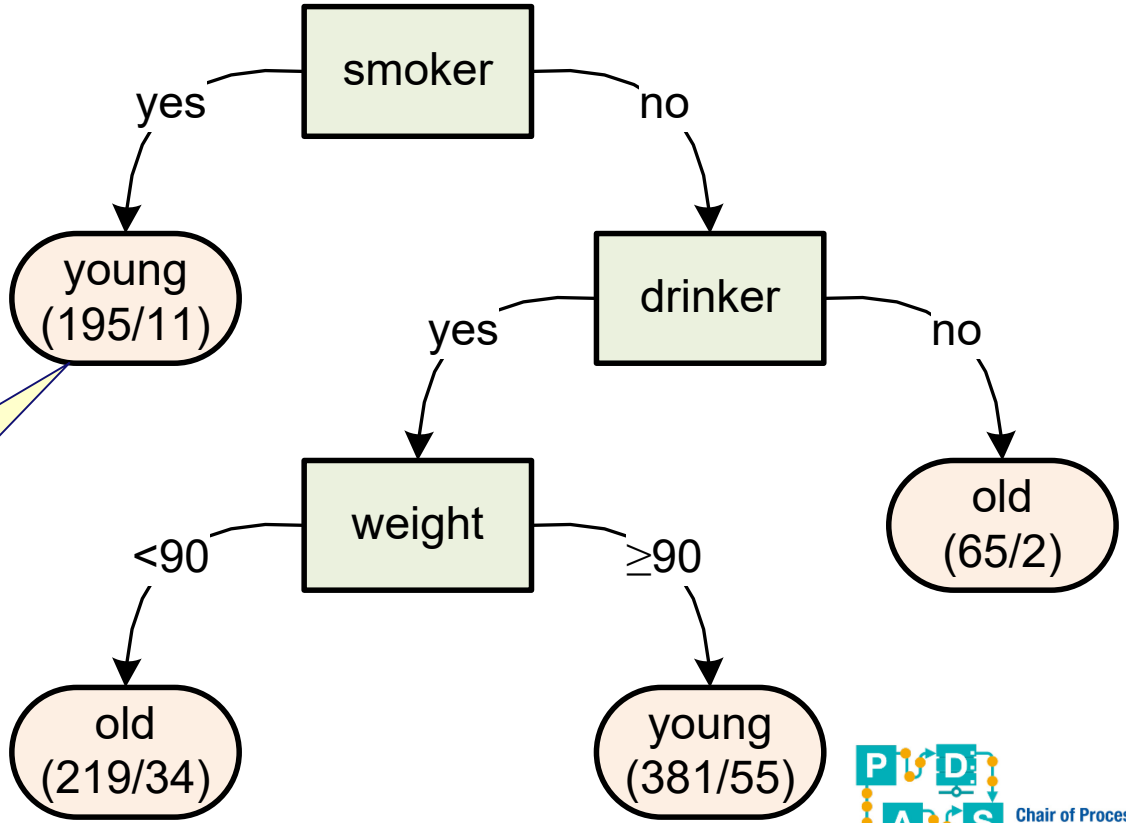
Class label:
- ≥70 = old
- <70 = young

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
…	…	…	…

⩾70 = old
<70 = young

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



Example: Study results

target feature

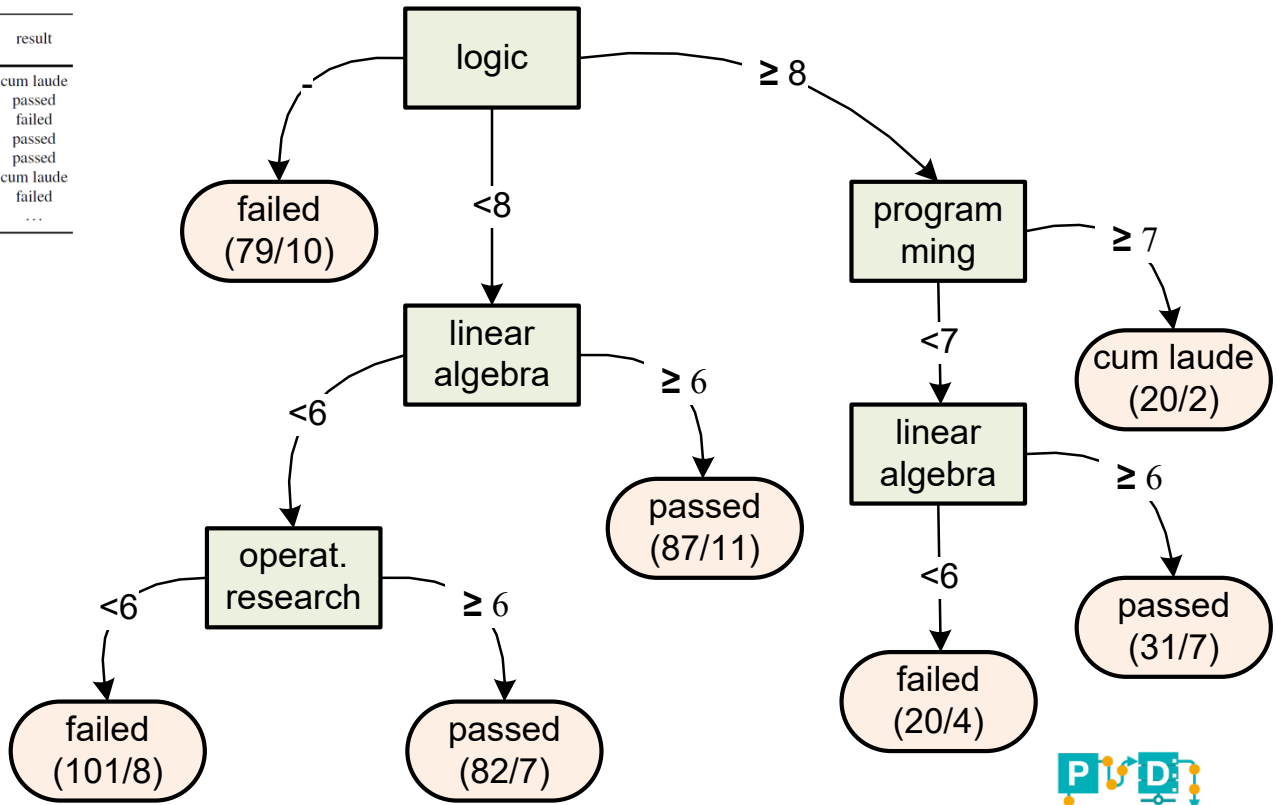
linear algebra	logic	program-ming	operations research	workflow systems	...	duration	result
9	8	8	9	9	...	36	cum laude
7	6	-	8	8	...	42	passed
-	-	5	4	6	...	54	failed
8	6	6	6	5	...	38	passed
6	7	6	-	8	...	39	passed
9	9	9	9	8	...	38	cum laude
5	5	-	6	6	...		
		...	...	...	...		

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
…	…	…	…	…	…	…	…



Chair of Process and Data Science

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.

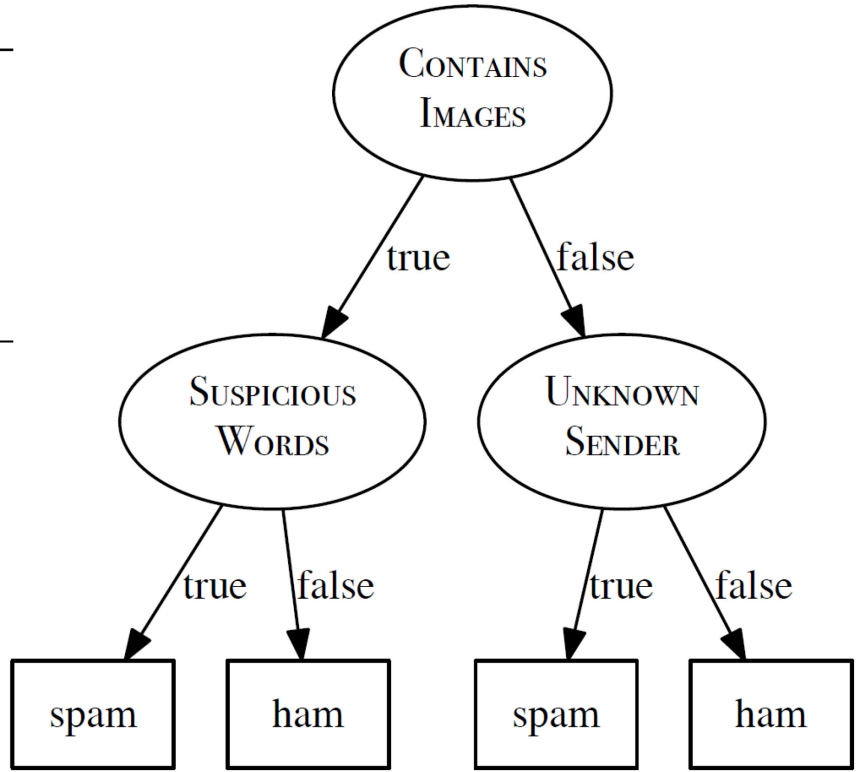
Example from book

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

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

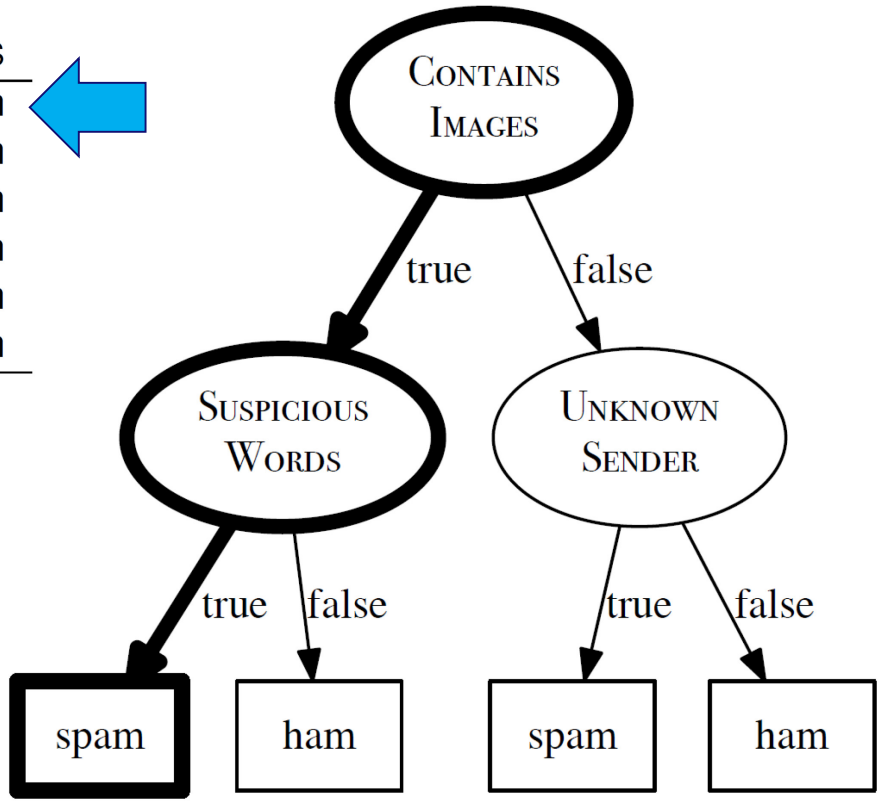
Example from book

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

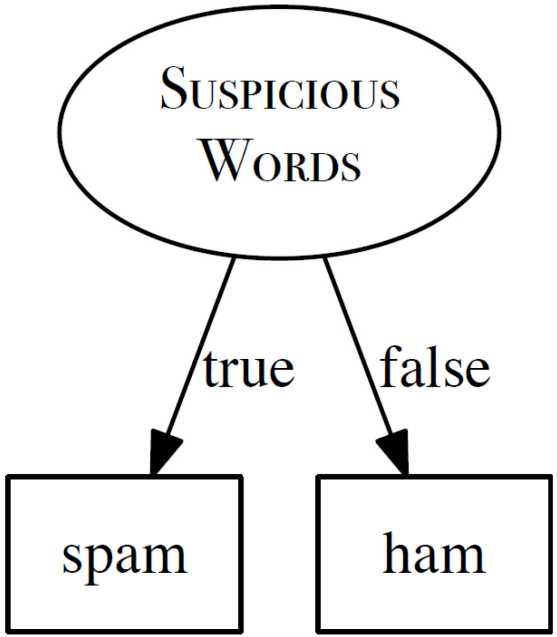
ID	Suspicious Words	Unknown Sender	Contains Images	Class
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham



All correctly classified.

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.

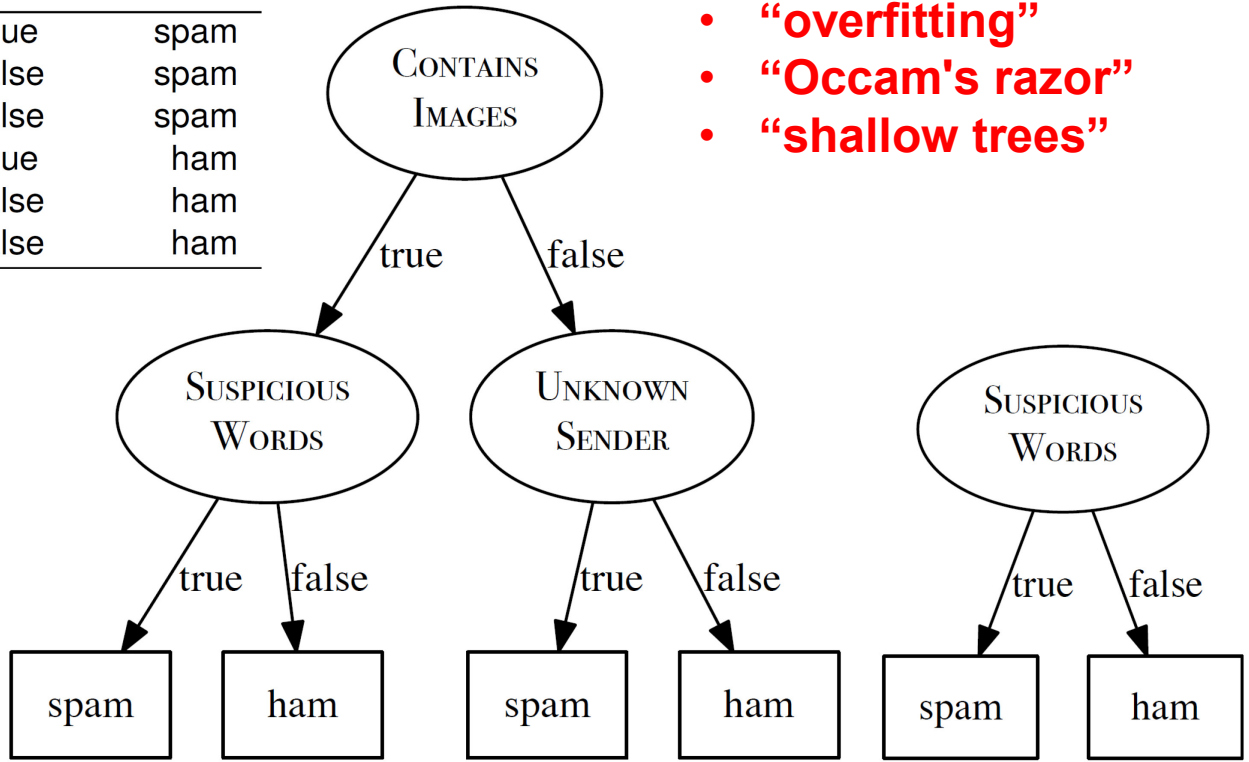
Example from book

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

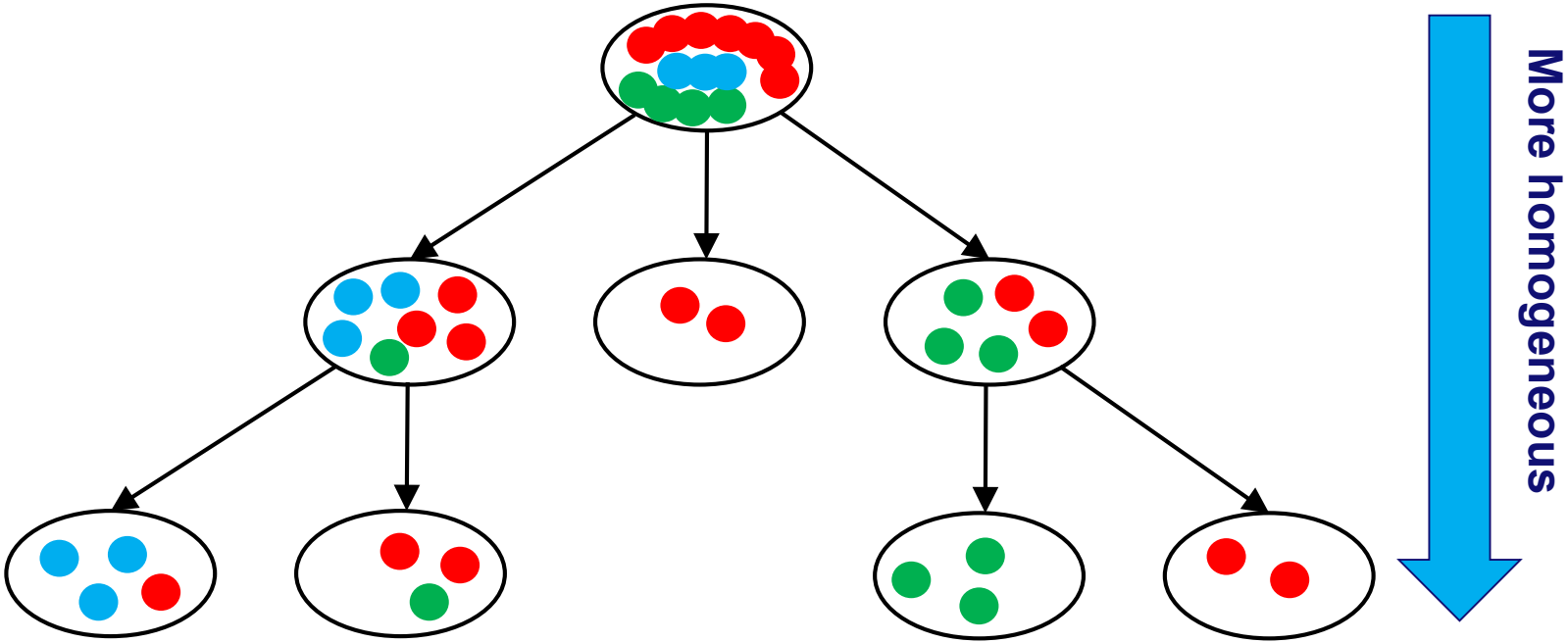
Keywords:

- **“overfitting”**
- **“Occam's razor”**
- **“shallow trees”**

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



Information gain idea



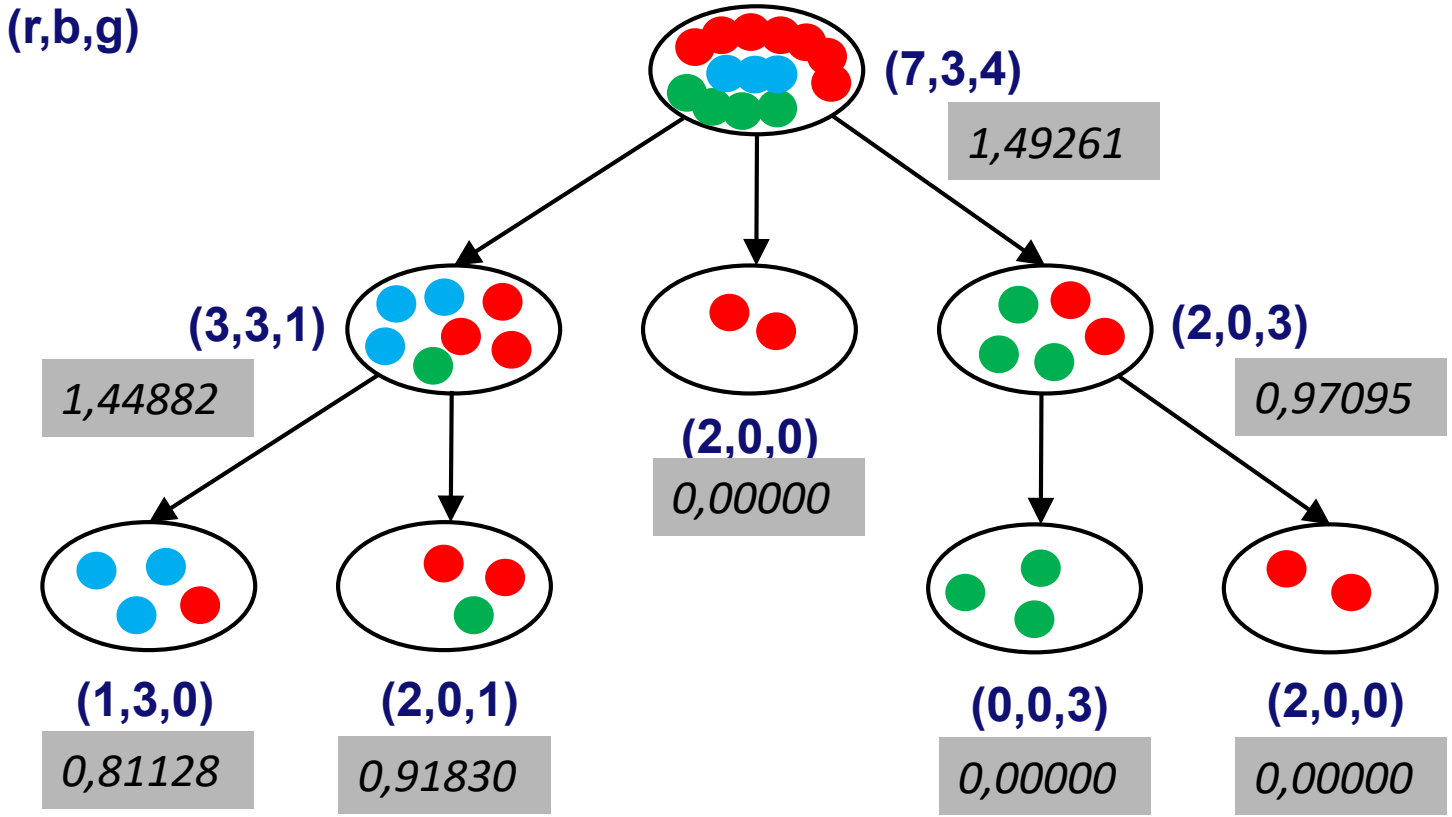
Information gain = improvement in knowledge (predictability of class label in nodes)

Entropy

Entropy: Intuition

measure of impurity
uncertainty when guessing
incompressibility

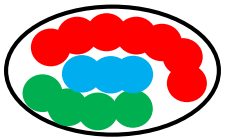
worst case three values: $\log_2(3) \approx 1.58$



Chair of Process and Data Science

Formula

1,49261



(7,3,4)

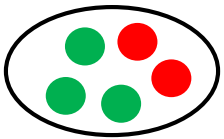
$$H(t) = -\sum_{i=1}^{I} (P(t=i) \times log_s(P(t=i)))$$

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

(we use as logarithm base s=2)

Formula

0,97095



(2,0,3)

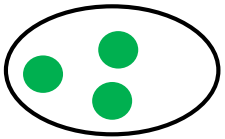
$$H(t) = -\sum_{i=1}^{I} (P(t=i) \times log_s(P(t=i)))$$

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

(we use as logarithm base s=2)

Formula

0,00000

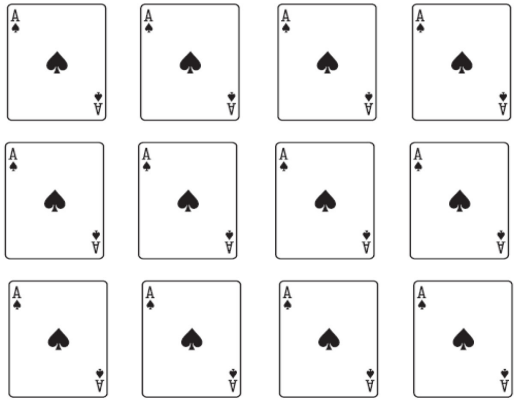


(0,0,3)

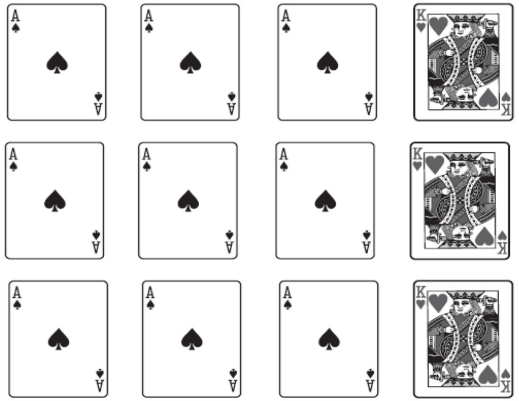
$$H(t) = -\sum_{i=1}^{I} (P(t=i) \times log_s(P(t=i)))$$

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

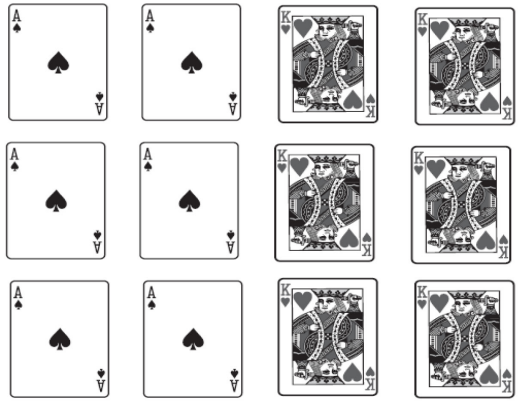
(we use as logarithm base s=2)



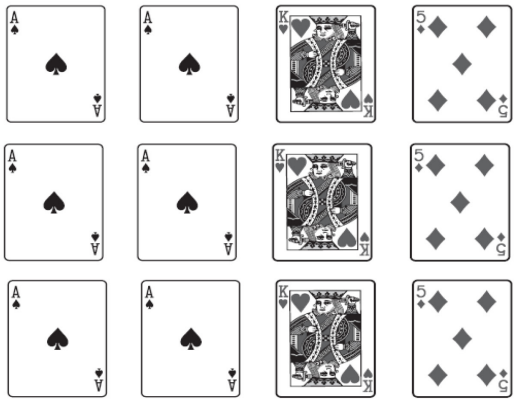
(a) $H(card) = 0.00$



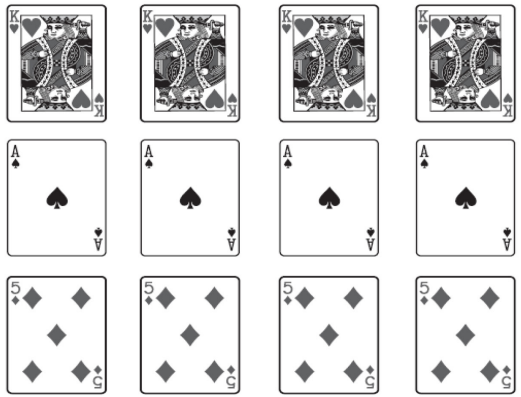
(b) $H(card) = 0.81$



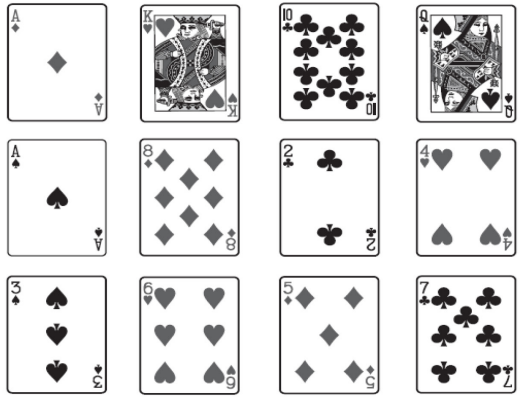
(c) $H(card) = 1.00$



(d) $H(card) = 1.50$



(e) $H(card) = 1.58$



(f) $H(card) = 3.58$

Question

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

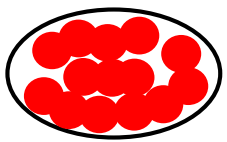
$$H(t) = -\sum_{i=1}^{l} (P(t=i) \times log_s(P(t=i)))$$

Lowest entropy

- **What distribution of the *n* instances over the *l* 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

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



Chair of Process and Data Science

Highest entropy

- **What distribution of the *n* instances over the *l* 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

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

Chair of Process and Data Science

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,…,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

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

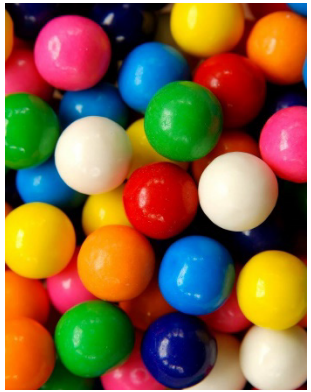
Chair of Process and Data Science

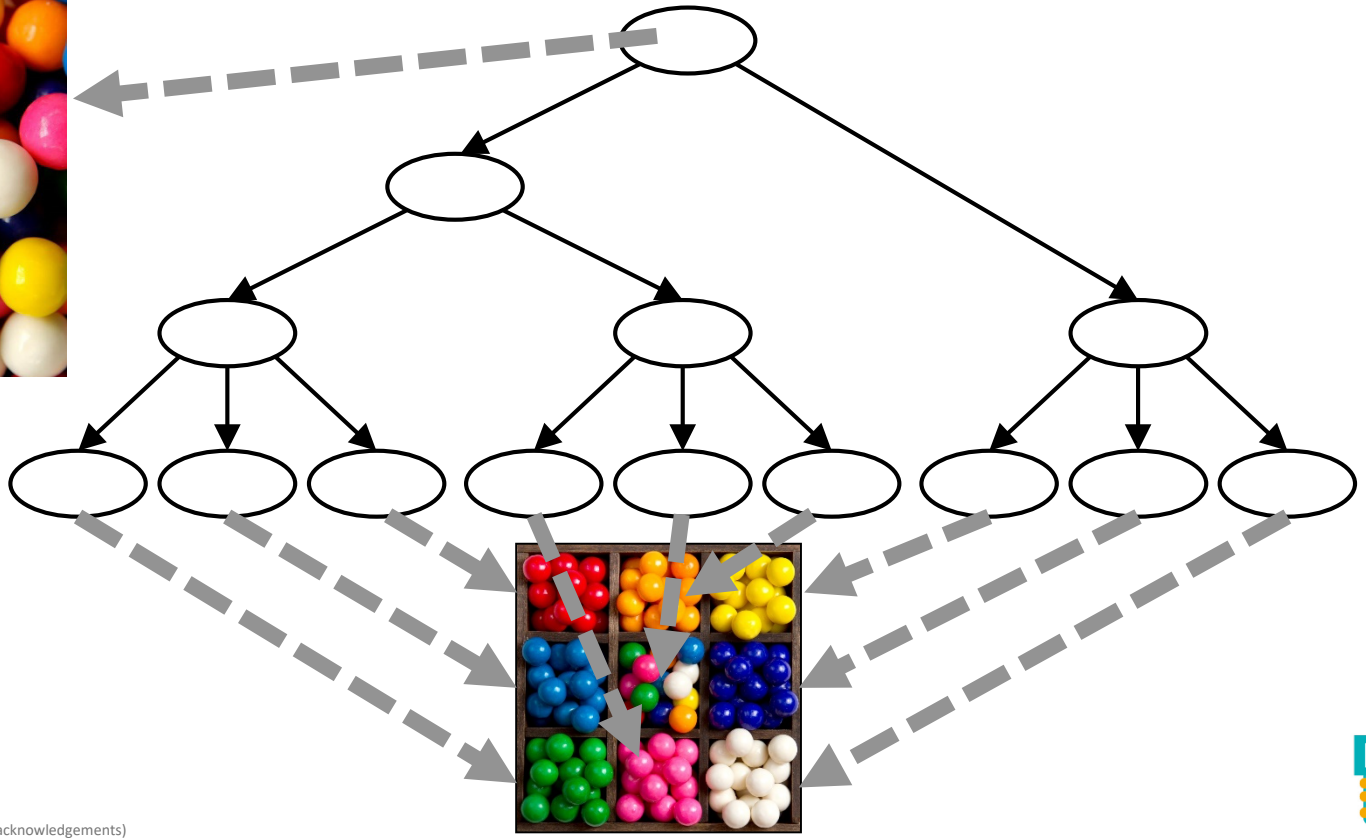
Interpretation

impurity = uncertainty = incompressible

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

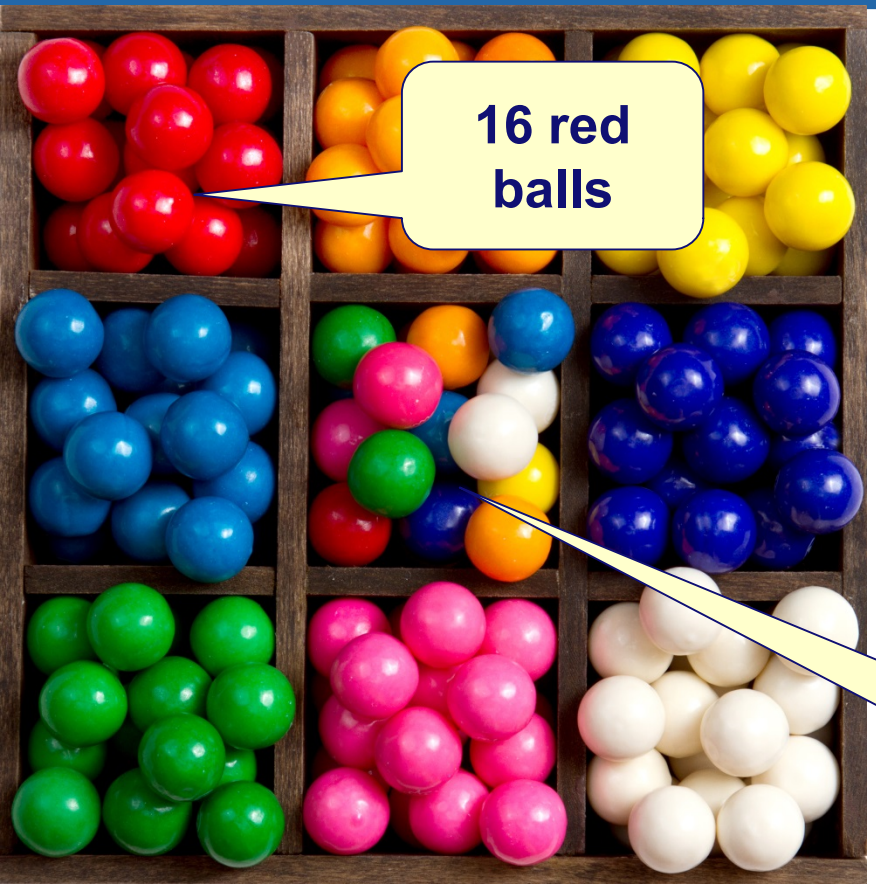
Decision tree: Aim is to have “pure” leaves





Chair of Process and Data Science

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+2 = 8x2 balls**

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

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

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

Chair of Process and Data Science

Answer: *E=0* for other cells

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

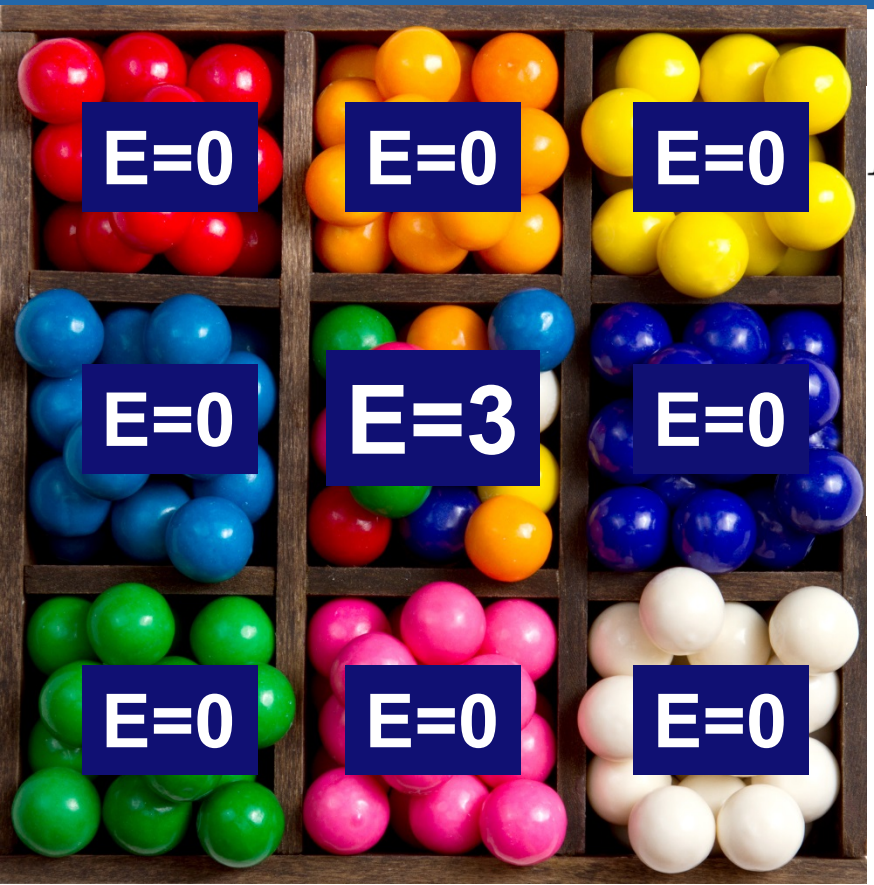
$$E = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

$$= -\sum_{i=1}^{1} \frac{16}{16} \log_2(\frac{16}{16})$$

$$= -1 \times 0 = 0$$

Chair of Process and Data Science

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

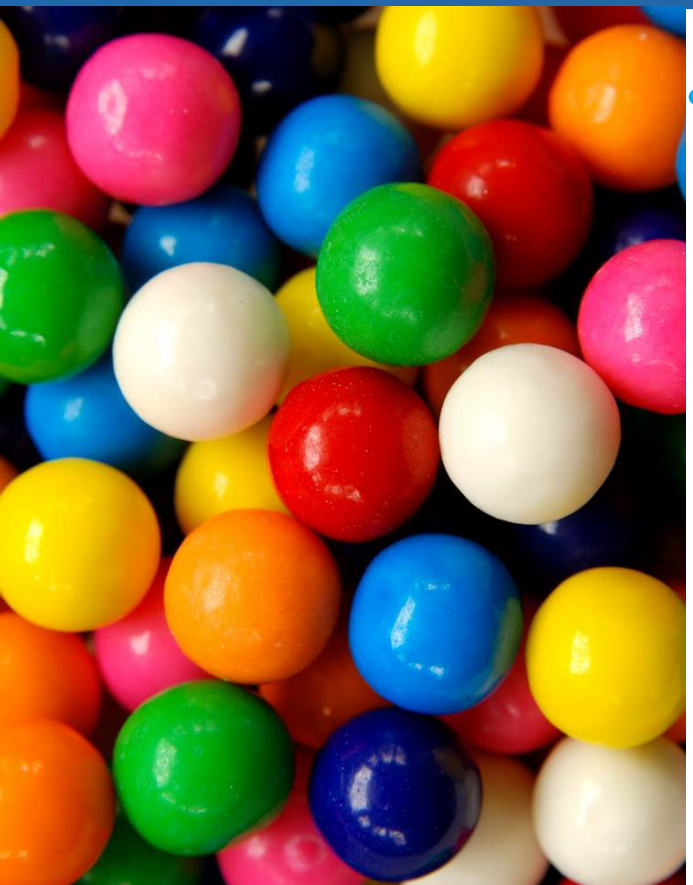


$$E = \frac{16}{144} \times 0 + \frac{16}{144} \times 0 + \ldots + \frac{16}{144} \times 3$$

$$= 8 \times \frac{16}{144} \times 0 + \frac{16}{144} \times 3$$

$$= \frac{1}{9} \times 3 = \frac{1}{3}$$

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



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

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

$$= -\sum_{i=1}^{8} \frac{18}{144} \log_2(\frac{18}{144})$$

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

E=0.3333

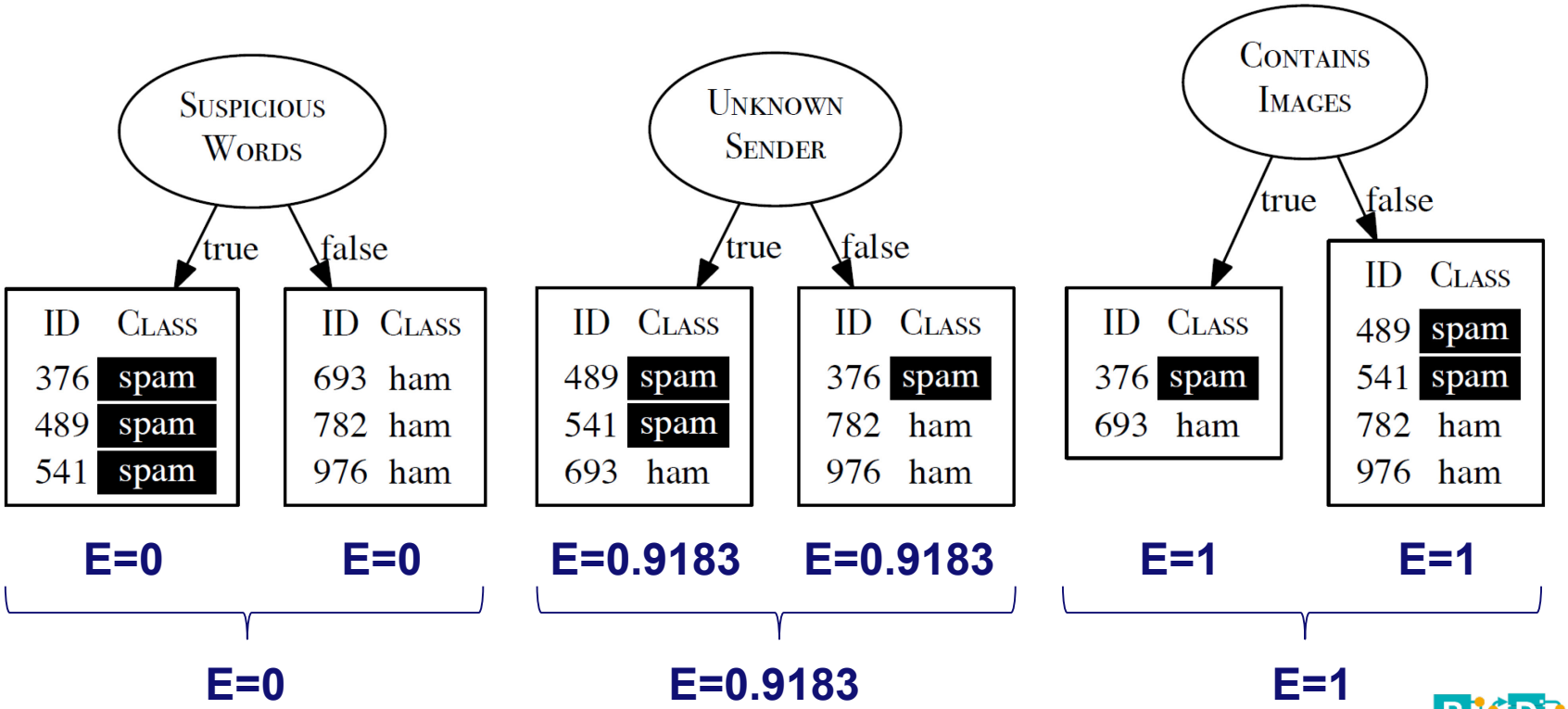
information loss = 2.6666

E=3

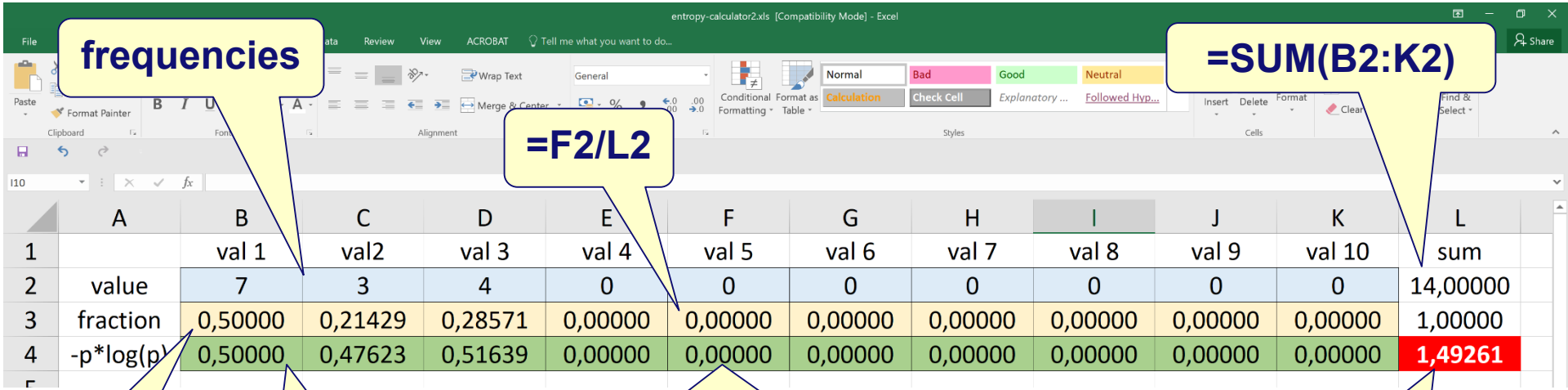
information gain = 2.6666

Chair of Process and Data Science

Example revisited



Building an entropy calculator in Excel



	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

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

Chair of Process and Data Science

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**

Chair of Process and Data Science

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**

Chair of Process and Data Science

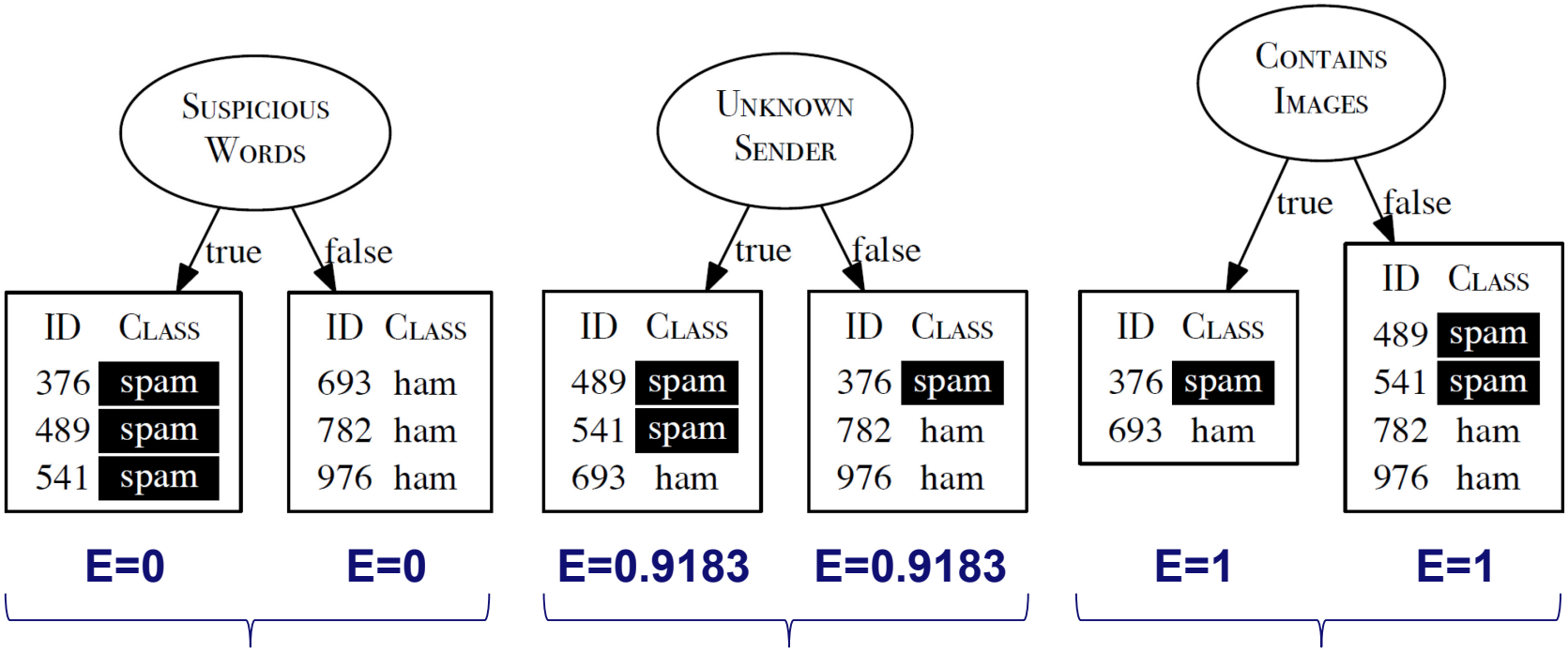
Example from book revisited

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

$$
\begin{aligned}
H(t, \mathcal{D}) &= -\sum_{l \in \{\text{'spam'}, \text{'ham'}\}} (P(t = l) \times log_2(P(t = l))) \\
&= -\ ((P(t = \text{'spam'}) \times log_2(P(t = \text{'spam'})) \\
&\quad + (P(t = \text{'ham'}) \times log_2(P(t = \text{'ham'}))) \\
&= -\left(\left({}^3/_6 \times log_2({}^3/_6)\right) + \left({}^3/_6 \times log_2({}^3/_6)\right)\right) \\
&= 1\ bit
\end{aligned}
$$

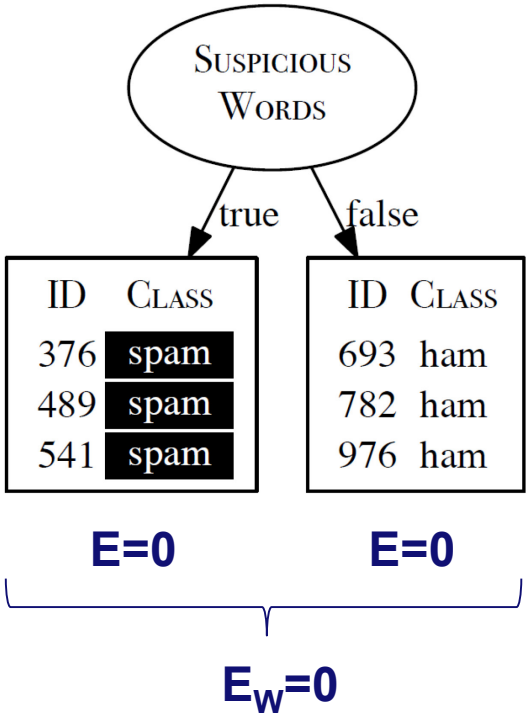
Example taken from Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

Example revisited



Chair of Process and Data Science

Example revisited



$$rem(\text{WORDS}, \mathcal{D}) = \left(\frac{|\mathcal{D}_{\text{WORDS}=T}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{\text{WORDS}=T})\right) + \left(\frac{|\mathcal{D}_{\text{WORDS}=F}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{\text{WORDS}=F})\right)$$

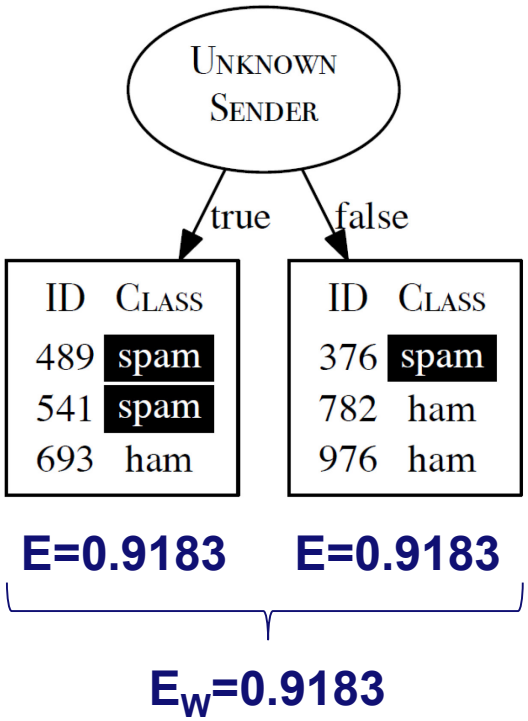
$$= \left({}^{3}/_{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t=l) \times log_2(P(t=l))\right)\right)$$

$$+ \left({}^{3}/_{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t=l) \times log_2(P(t=l))\right)\right)$$

$$= \left({}^{3}/_{6} \times \left(-\left(\left({}^{3}/_{3} \times log_2({}^{3}/_{3})\right) + \left({}^{0}/_{3} \times log_2({}^{0}/_{3})\right)\right)\right)\right)$$

$$+ \left({}^{3}/_{6} \times \left(-\left(\left({}^{0}/_{3} \times log_2({}^{0}/_{3})\right) + \left({}^{3}/_{3} \times log_2({}^{3}/_{3})\right)\right)\right)\right) = 0 \; bits$$

Example revisited



$$rem(\text{SENDER}, \mathcal{D}) = \left(\frac{|\mathcal{D}_{\text{SENDER}=T}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{\text{SENDER}=T})\right) + \left(\frac{|\mathcal{D}_{\text{SENDER}=F}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{\text{SENDER}=F})\right)$$

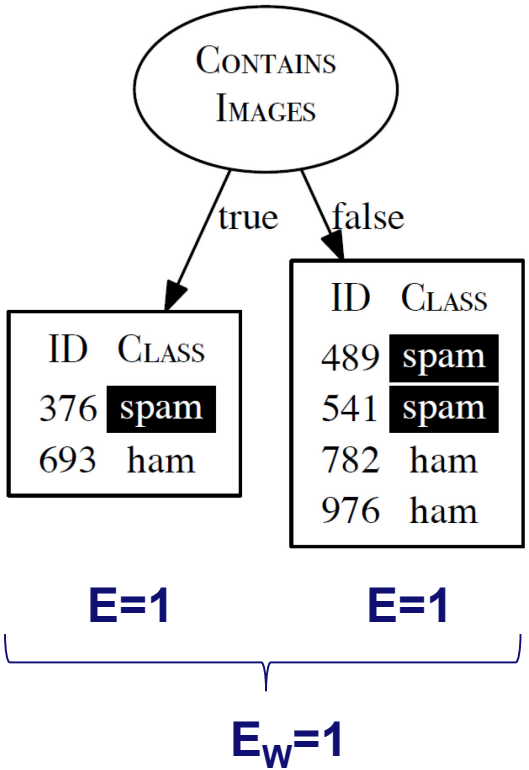
$$= \left({}^{3}/_{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t=l) \times log_2(P(t=l))\right)\right)$$

$$+ \left({}^{3}/_{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t=l) \times log_2(P(t=l))\right)\right)$$

$$= \left({}^{3}/_{6} \times \left(- \left(\left({}^{2}/_{3} \times log_2({}^{2}/_{3})\right) + \left({}^{1}/_{3} \times log_2({}^{1}/_{3})\right)\right)\right)\right)$$

$$+ \left({}^{3}/_{6} \times \left(- \left(\left({}^{1}/_{3} \times log_2({}^{1}/_{3})\right) + \left({}^{2}/_{3} \times log_2({}^{2}/_{3})\right)\right)\right)\right) = 0.9183 \; bits$$

Example revisited



$$rem(\text{IMAGES}, \mathcal{D}) = \left(\frac{|\mathcal{D}_{\text{IMAGES}=T}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{\text{IMAGES}=T})\right) + \left(\frac{|\mathcal{D}_{\text{IMAGES}=F}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{\text{IMAGES}=F})\right)$$

$$= \left({}^{2}/_{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t=l) \times log_2(P(t=l))\right)\right)$$

$$+ \left({}^{4}/_{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t=l) \times log_2(P(t=l))\right)\right)$$

$$= \left({}^{2}/_{6} \times \left(- \left(\left({}^{1}/_{2} \times log_2({}^{1}/_{2})\right) + \left({}^{1}/_{2} \times log_2({}^{1}/_{2})\right)\right)\right)\right)$$

$$+ \left({}^{4}/_{6} \times \left(- \left(\left({}^{2}/_{4} \times log_2({}^{2}/_{4})\right) + \left({}^{2}/_{4} \times log_2({}^{2}/_{4})\right)\right)\right)\right) = 1 \; bit$$

Information gain

$$IG(\text{SUSPICIOUS WORDS}, \mathcal{D}) = H(\text{CLASS}, \mathcal{D}) - rem(\text{SUSPICIOUS WORDS}, \mathcal{D})$$
$$= 1 - 0 = 1 \; bit \qquad \textbf{good}$$

$$IG(\text{UNKNOWN SENDER}, \mathcal{D}) = H(\text{CLASS}, \mathcal{D}) - rem(\text{UNKNOWN SENDER}, \mathcal{D})$$
$$= 1 - 0.9183 = 0.0817 \; bits \qquad \textbf{not so good}$$

$$IG(\text{CONTAINS IMAGES}, \mathcal{D}) = H(\text{CLASS}, \mathcal{D}) - rem(\text{CONTAINS IMAGES}, \mathcal{D})$$
$$= 1 - 1 = 0 \; bits \qquad \textbf{worst}$$

ID3 algorithm

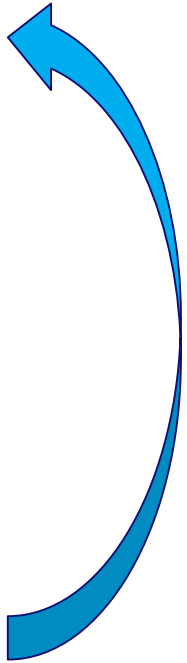
ID3 (Iterative Dichotomiser 3)

- 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 minimum (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: $\mathbf{d}[best] \leftarrow \underset{d \in \mathbf{d}}{\arg\max}\, IG(d, \mathcal{D})$
9: make a new node, $Node_{\mathbf{d}[best]}$, and label it with $\mathbf{d}[best]$
10: partition $\mathcal{D}$ using $\mathbf{d}[best]$
11: remove $\mathbf{d}[best]$ from $\mathbf{d}$
12: **for** each partition $\mathcal{D}_i$ of $\mathcal{D}$ **do**
13: grow a branch from $Node_{\mathbf{d}[best]}$ 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 **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 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: $\mathbf{d}[best] \leftarrow \arg\max_{d \in \mathbf{d}} IG(d, \mathcal{D})$
9: make a new node, $Node_{\mathbf{d}[best]}$, and label it with $\mathbf{d}[best]$
10: partition $\mathcal{D}$ using $\mathbf{d}[best]$
11: remove $\mathbf{d}[best]$ from **d**
12: **for** each partition $\mathcal{D}_i$ of $\mathcal{D}$ **do**
13: grow a branch from $Node_{\mathbf{d}[best]}$ 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: $\mathbf{d}[best] \leftarrow \arg\max_{d \in \mathbf{d}} IG(d, \mathcal{D})$
9: make a new node, $Node_{\mathbf{d}[best]}$, and label it with $\mathbf{d}[best]$
10: partition $\mathcal{D}$ using $\mathbf{d}[best]$
11: remove $\mathbf{d}[best]$ from $\mathbf{d}$
12: **for** each partition $\mathcal{D}_i$ of $\mathcal{D}$ **do**
13: grow a branch from $Node_{\mathbf{d}[best]}$ 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 **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 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: $\mathbf{d}[best] \leftarrow \arg\max_{d \in \mathbf{d}} IG(d, \mathcal{D})$
9: make a new node, $Node_{\mathbf{d}[best]}$, and label it with $\mathbf{d}[best]$
10: partition $\mathcal{D}$ using $\mathbf{d}[best]$
11: remove $\mathbf{d}[best]$ from **d**
12: **for** each partition $\mathcal{D}_i$ of $\mathcal{D}$ **do**
13: grow a branch from $Node_{\mathbf{d}[best]}$ 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

$$
\begin{aligned}
&H(\text{VEGETATION}, \mathcal{D}) \\
&= - \sum_{l \in \left\{ \begin{array}{l} \text{'chaparral'}, \\ \text{'riparian'}, \\ \text{'conifer'} \end{array} \right\}} P(\text{VEGETATION} = l) \times log_2 \left(P(\text{VEGETATION} = l) \right) \\
&= - \left(\left({}^3/_7 \times log_2({}^3/_7) \right) + \left({}^2/_7 \times log_2({}^2/_7) \right) + \left({}^2/_7 \times log_2({}^2/_7) \right) \right) \\
&= 1.5567 \; bits
\end{aligned}
$$

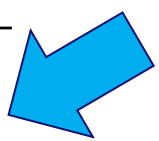
Example taken from Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

Example from book

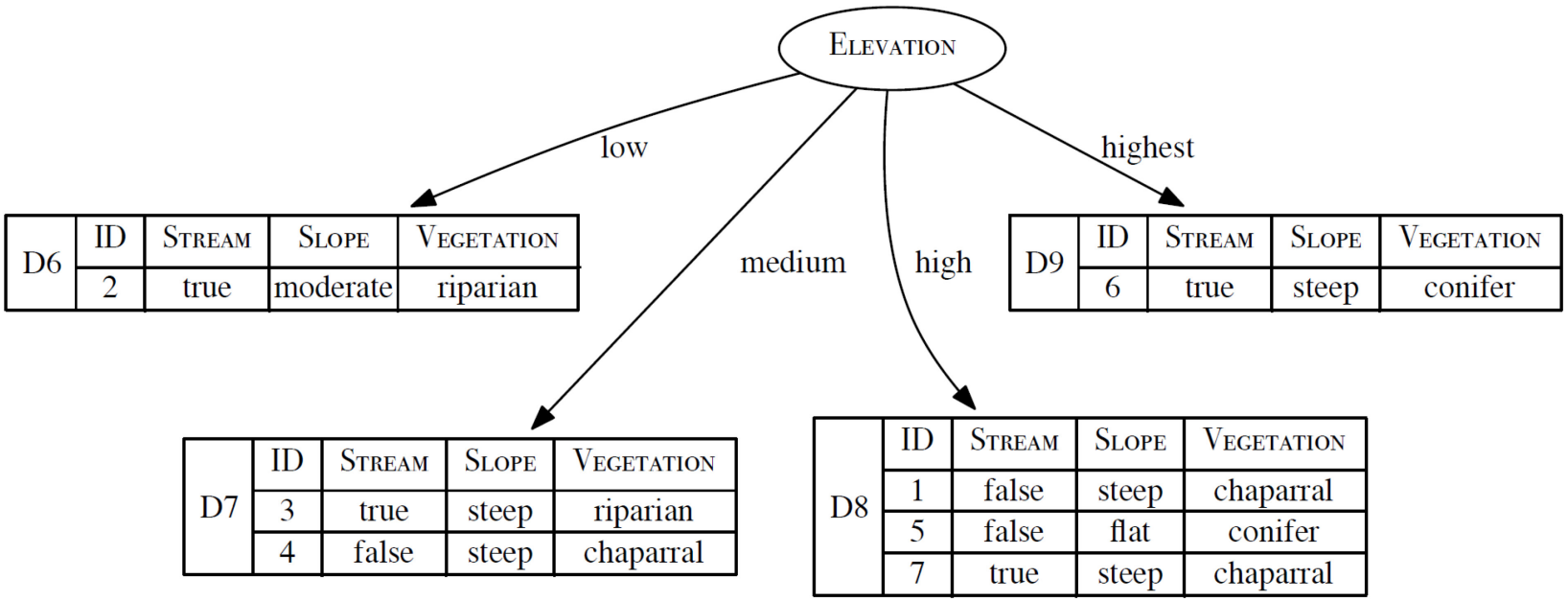
ID	Stream	Slope	Elevation	Vegetation
1	false	steep	high	chaparral
2	true	moderate	low	riparian
3	true	steep	medium	riparian
4	false	steep	medium	chaparral
5	false	flat	high	conifer
6	true	steep	highest	conifer
7	true	steep	high	chaparral

$$
\begin{aligned}
&H(\text{Vegetation}, \mathcal{D}) \\
&= - \sum_{l \in \left\{ \begin{smallmatrix} \text{'chaparral'}, \\ \text{'riparian'}, \\ \text{'conifer'} \end{smallmatrix} \right\}} P(\text{Vegetation} = l) \times log_2 \left(P(\text{Vegetation} = l) \right) \\
&= - \left(\left({}^3/_7 \times log_2({}^3/_7) \right) + \left({}^2/_7 \times log_2({}^2/_7) \right) + \left({}^2/_7 \times log_2({}^2/_7) \right) \right) \\
&= 1.5567 \; bits
\end{aligned}
$$

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



D6

ID	Stream	Slope	Vegetation
2	true	moderate	riparian

D7

ID	Stream	Slope	Vegetation
3	true	steep	riparian
4	false	steep	chaparral

D8

ID	Stream	Slope	Vegetation
1	false	steep	chaparral
5	false	flat	conifer
7	true	steep	chaparral

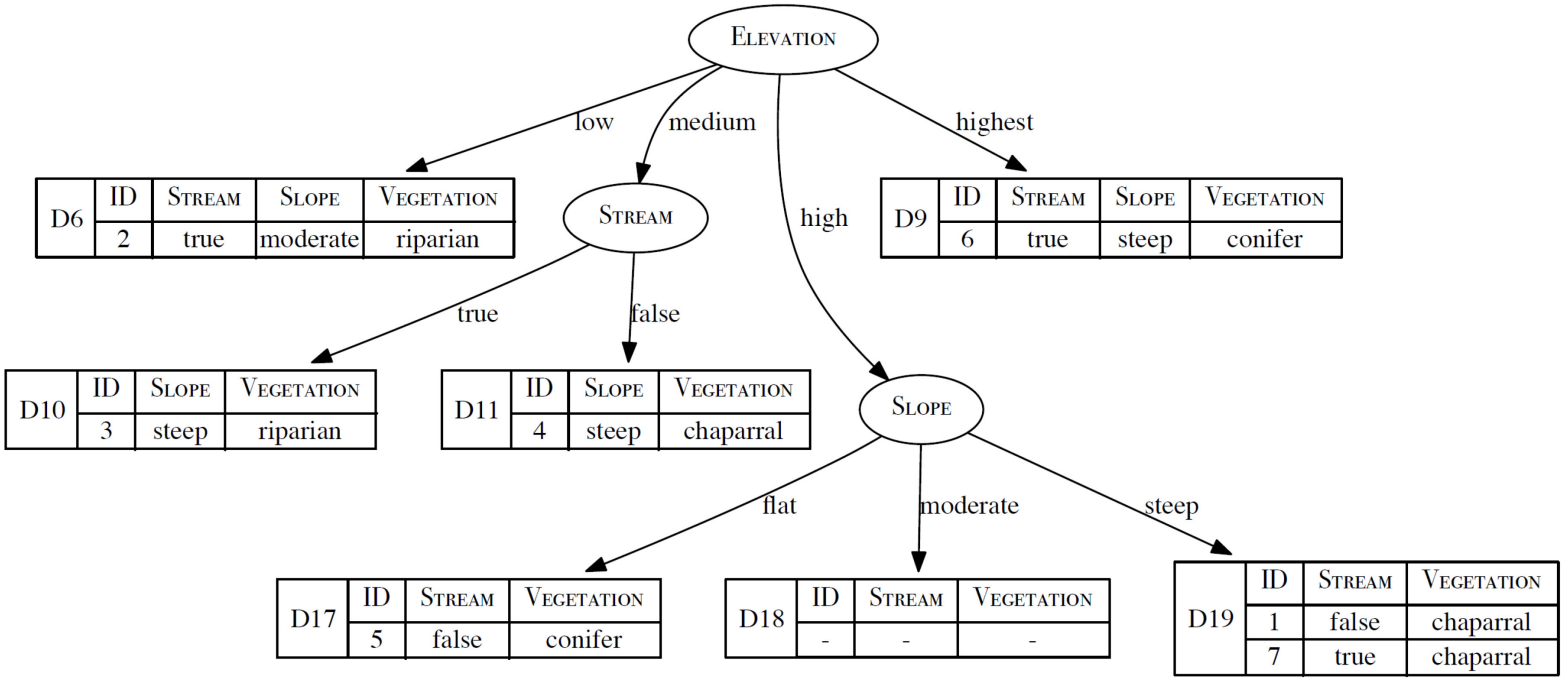
D9

ID	Stream	Slope	Vegetation
6	true	steep	conifer

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

Chair of Process and Data Science

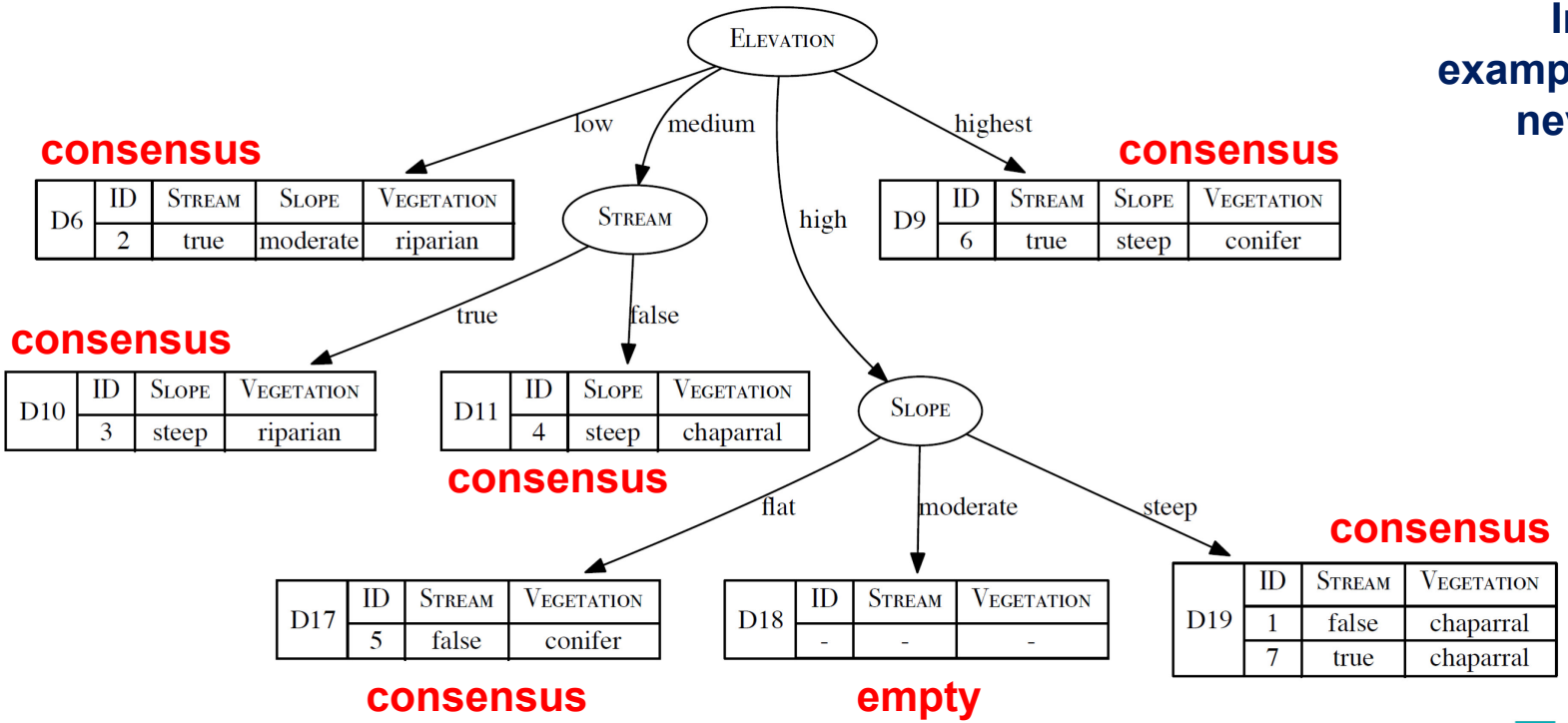
Example from book



Recursion until one of the three conditions holds.

Chair of Process and Data Science

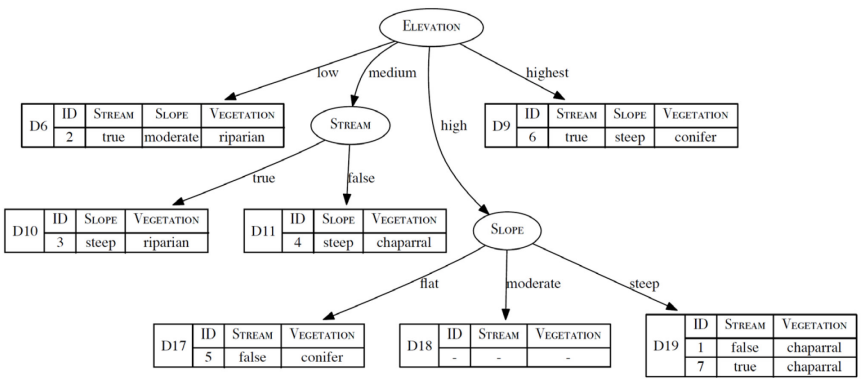
Example from book



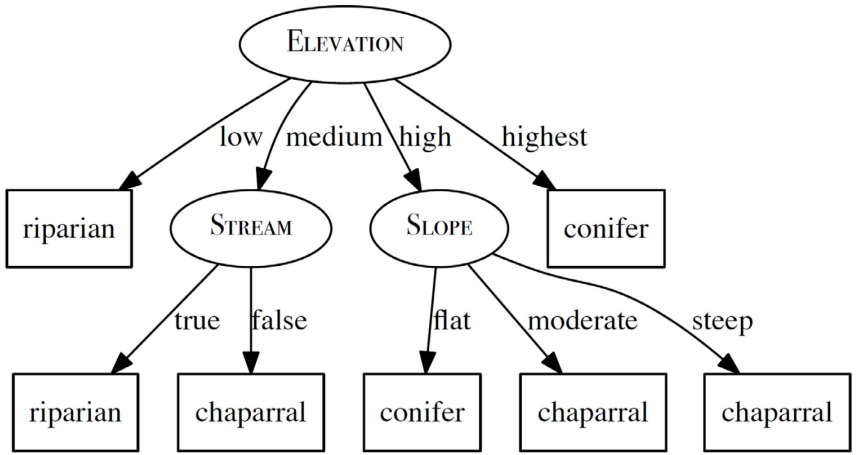
In this small example a branch never uses all features.

Example from book

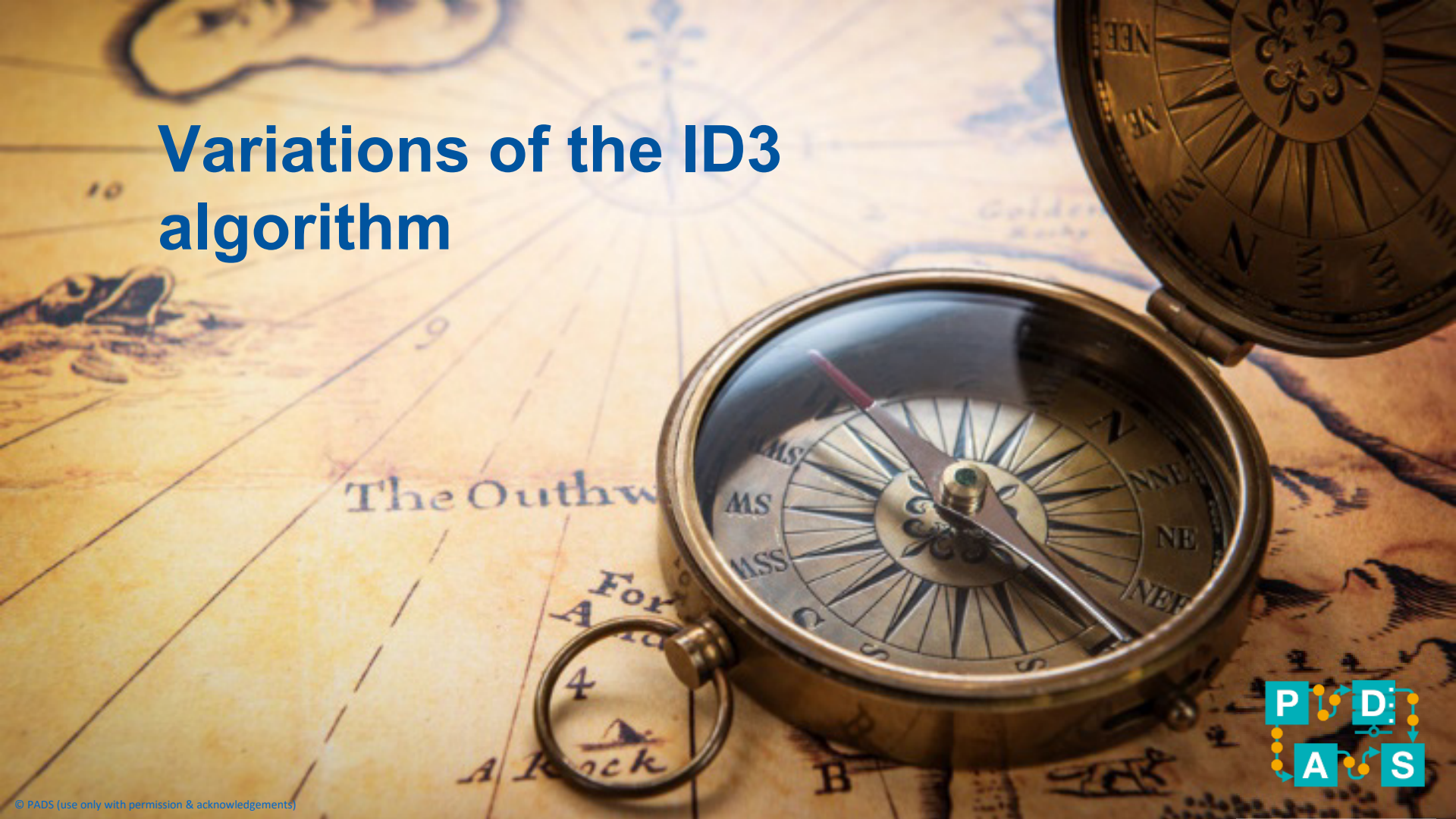
ID	Stream	Slope	Elevation	Vegetation
1	false	steep	high	chaparral
2	true	moderate	low	riparian
3	true	steep	medium	riparian
4	false	steep	medium	chaparral
5	false	flat	high	conifer
6	true	steep	highest	conifer
7	true	steep	high	chaparral



Final result



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)

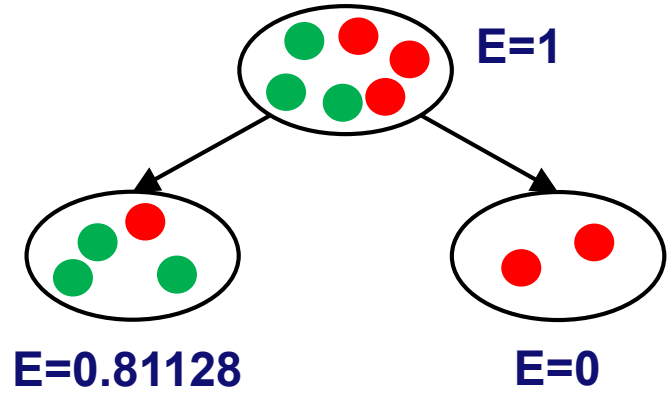
Information gain ratio

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

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

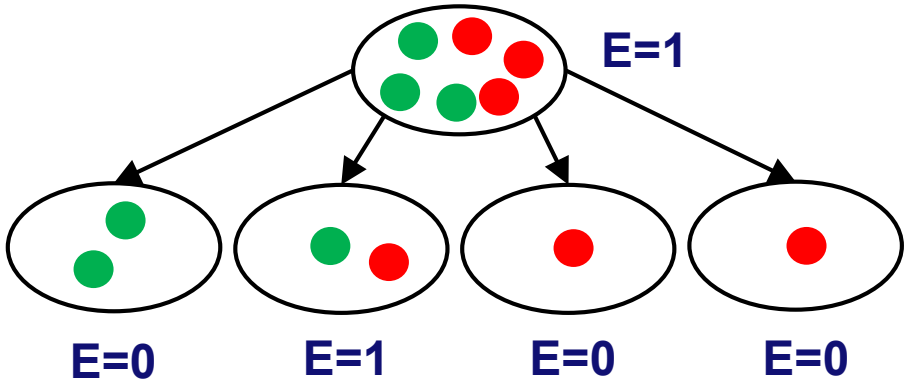
(like making an absolute value relative)

Information gain



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

IG = 0.4591

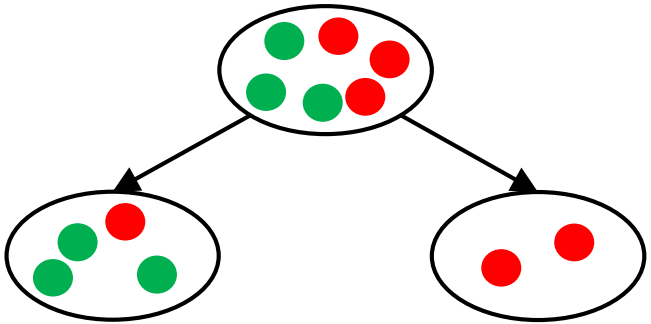


$E_W = 2/6 \times 1 = 0.3333$

IG = 0.6666

Information gain ratio

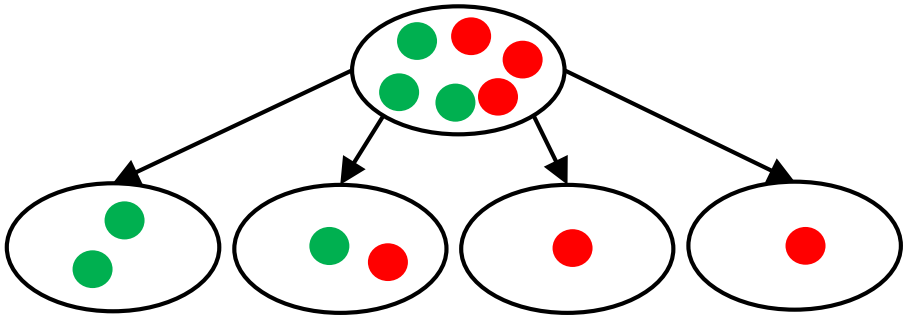
$$GR(d, \mathcal{D}) = \frac{IG(d, \mathcal{D})}{-\sum_{l \in levels(d)} (P(d = l) \times log_2(P(d = l)))}$$



(4,2) ⇒ E'=0,91830

IG = 0.4591

GR = 0.4591/E' = 0.50



(2,2,1,1) ⇒ E'=1,91830

IG = 0.6666

GR = 0.6666/E' = 0.34

Gini index

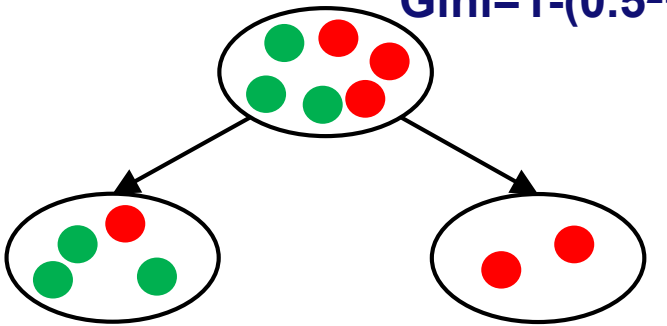
- **Alternative measure of impurity**
- **Expected misclassification rate when guessing based on observed distribution**

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

With probability P(t=l) we guess t=l and with probability P(t=l) this is right.

Gini index

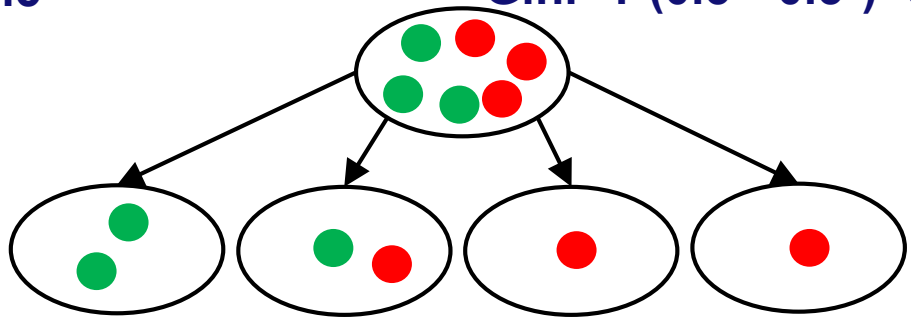
$$Gini\,(t, \mathcal{D}) = 1 - \sum_{l \in levels(t)} P(t = l)^2$$



Gini=1-(0.5²+0.5²)=0.5

Gini= 1-(0.25²+0.75²) = 0.375

Gini=1-1²=0



Gini=1-(0.5²+0.5²)=0.5

Gini=0 Gini=0.5 Gini=0 Gini=0

$Gini_w$= 4/6 x 0.375 = 0.25

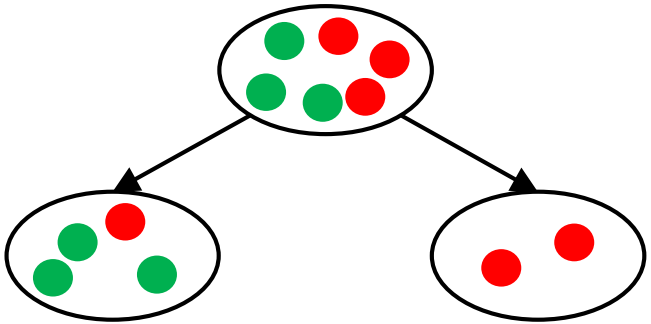
IG_{Gini}= 0.5 - 0.25 = 0.25

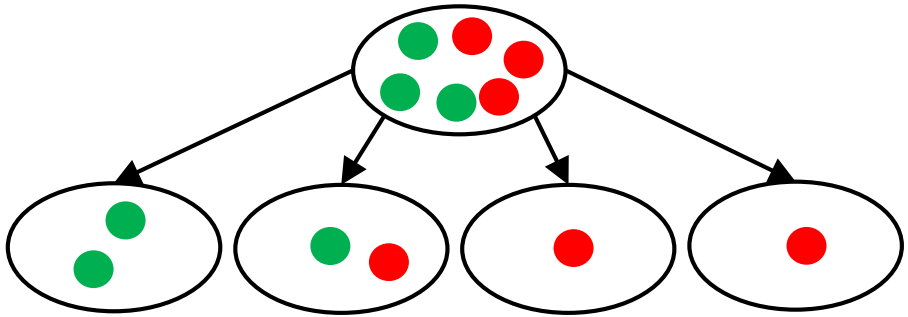
$Gini_w$= 2/6 x 0.5 = 0.1666

IG_{Gini}= 0.5 – 0.166 = 0.33

Chair of Process and Data Science

Comparison





IG = 0.4591	normal information gain	IG = 0.6666
GR = 0.50	information gain ratio	GR = 0.34
IG_{Gini} = 0.25	Gini-based information gain ratio	IG_{Gini} = 0.33

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

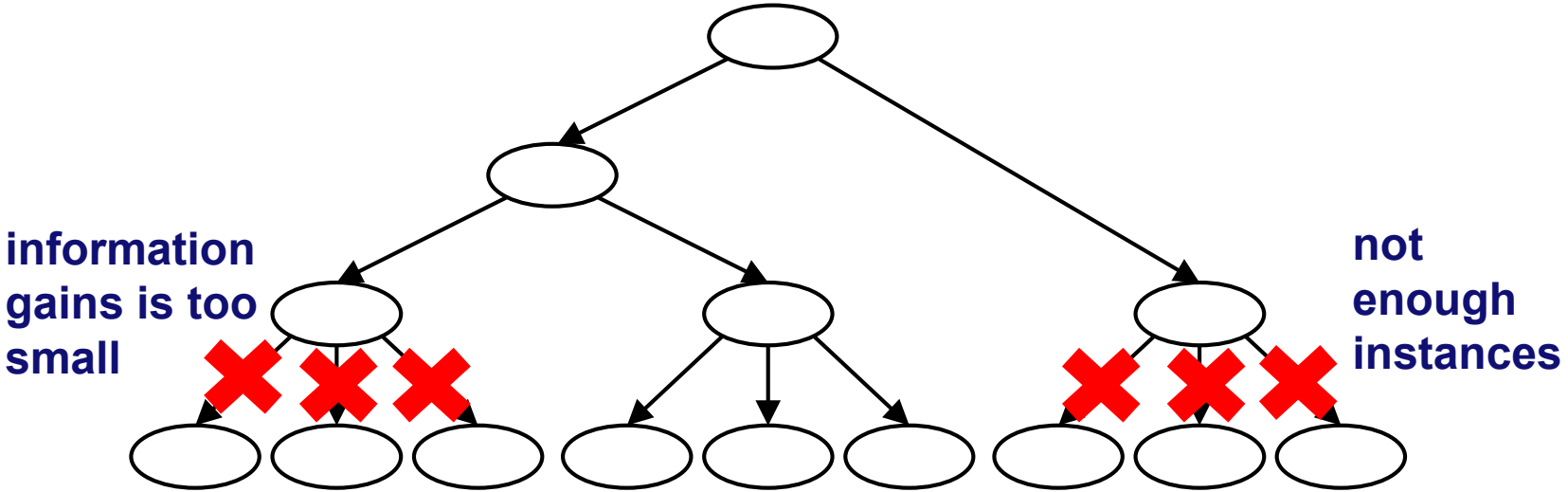
Pruning decision trees

- **Possible problems:**
  - Decision tree is overfitting the data.
  - Decision tree is too complex / deep.
- **To 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.

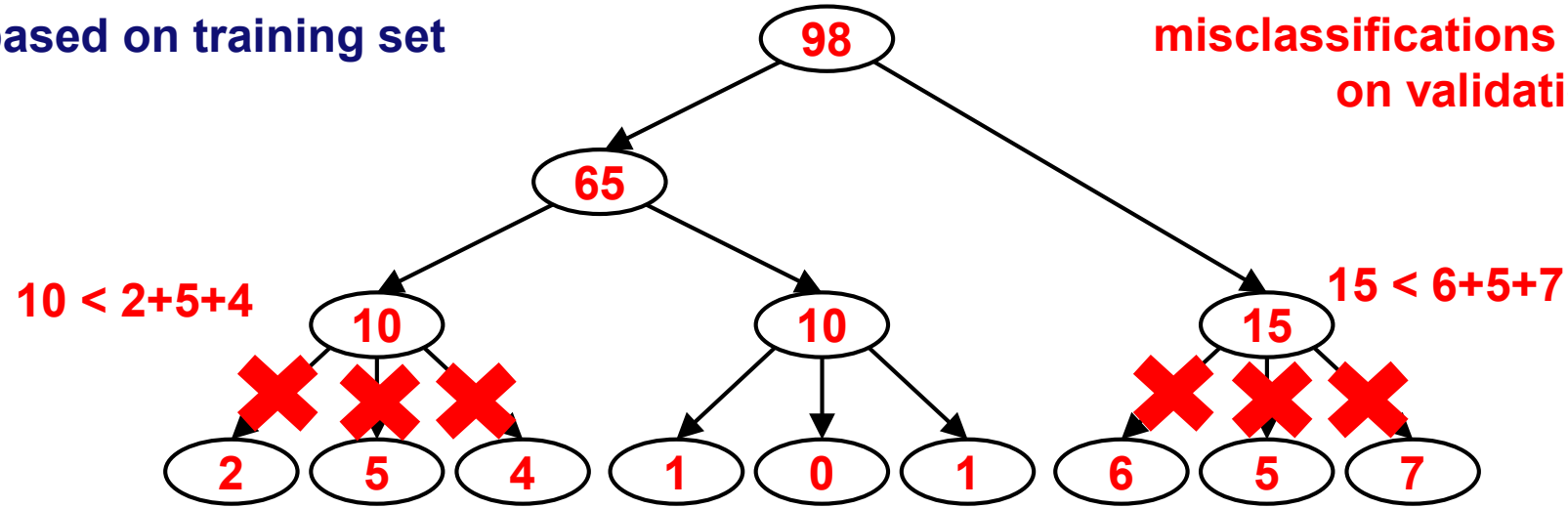
Post-pruning

- 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

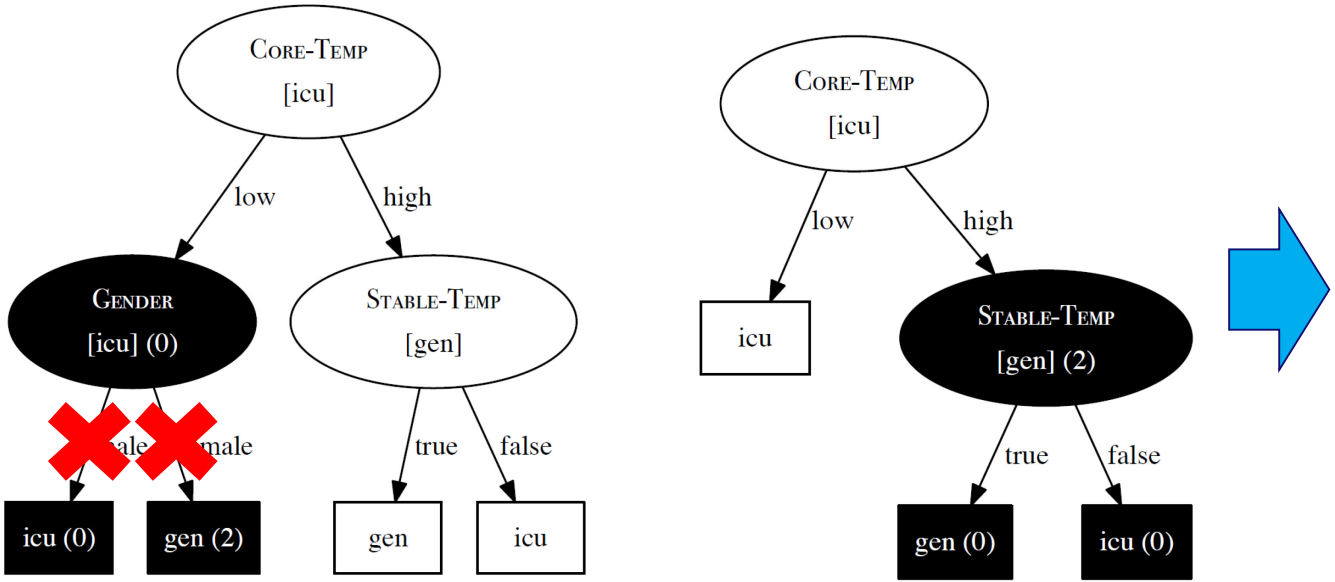
Tree was learned based on training set

Numbers indicate misclassifications based on validation set



Chair of Process and Data Science

Example in book



Chair of Process and Data Science

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).

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).

Ensemble: Bagging

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

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

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

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

Chair of Process and Data Science

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

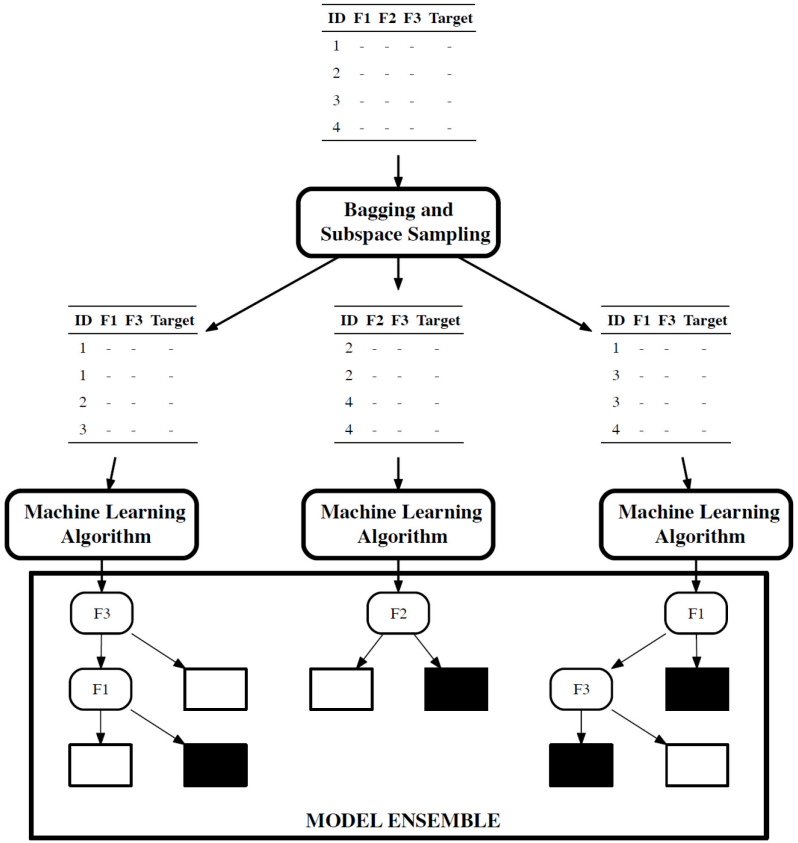
	f2	f3		…	fn	class
						high
						high
						low
						medium
						high
						low

f1	f2	f3			fn	class
						high
						high
						low
						medium
						high
						low

f1	f2		f4	…		class
						high
						high
						low
						medium
						high
						low

Chair of Process and Data Science

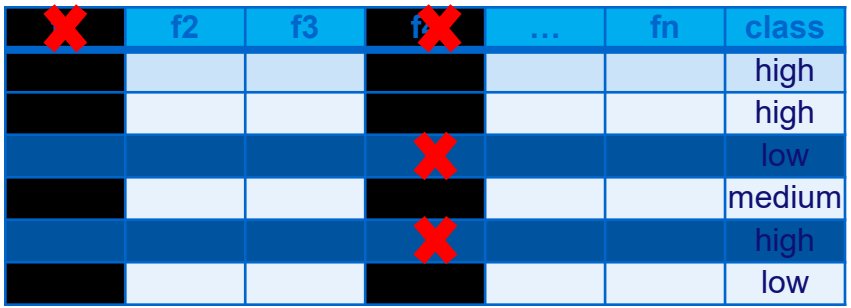
Ensemble: Random forest

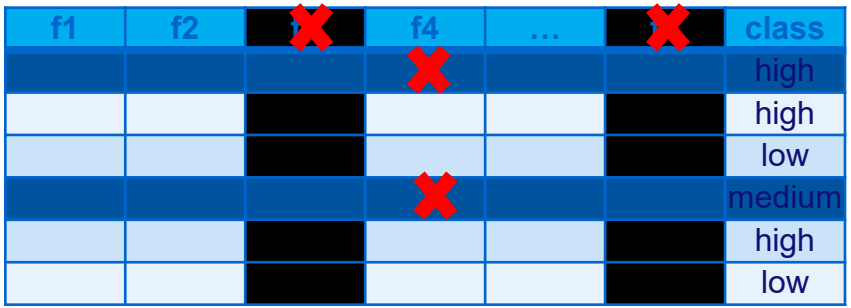


- **Combination of bagging and subspace sampling.**

Chair of Process and Data Science

Ensemble: Random forest





Chair of Process and Data Science

Dealing with continuous variables

Dealing with continuous variables

continuous descriptive features

continuous target feature

instances

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.

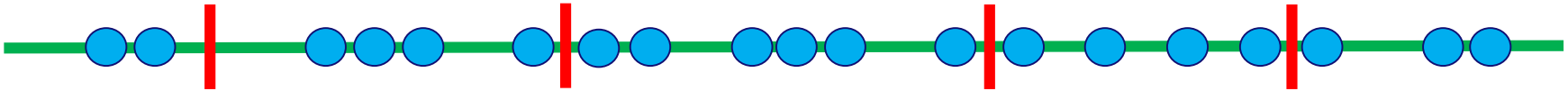
Recall that one can use binning to make continuous features categorical.

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
		old	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 highest information gain.

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

continuous descriptive feature

class

Example 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

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
			750	
4	false	steep	1 200	chapparal
			1350	
3	true	steep	1 500	riparian
			2250	
7	true	steep	3 000	chapparal
1	false	steep	3 900	chapparal
			4175	
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
			750	
4	false	steep	1 200	chapparal
			1350	
3	true	steep	1 500	riparian
			2250	
7	true	steep	3 000	chapparal
1	false	steep	3 900	chapparal
			4175	
5	false	flat	4 450	conifer
6	true	steep	5 000	conifer

Compute information gain: business as usual

Split by Threshold	Part.	Instances	Partition Entropy	Rem.	Info. Gain
≥750	$\mathcal{D}_1$	$\mathbf{d}_2$	0.0	1.2507	0.3060
	$\mathcal{D}_2$	$\mathbf{d}_4, \mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.4591		
≥1 350	$\mathcal{D}_3$	$\mathbf{d}_2, \mathbf{d}_4$	1.0	1.3728	0.1839
	$\mathcal{D}_4$	$\mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.5219		
≥2 250	$\mathcal{D}_5$	$\mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_3$	0.9183	0.9650	0.5917
	$\mathcal{D}_6$	$\mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.0		
≥4 175	$\mathcal{D}_7$	$\mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1$	0.9710	0.6935	0.8631
	$\mathcal{D}_8$	$\mathbf{d}_5, \mathbf{d}_6$	0.0		

best

Example from book

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

4175

<4,175 ≥4,175

D7

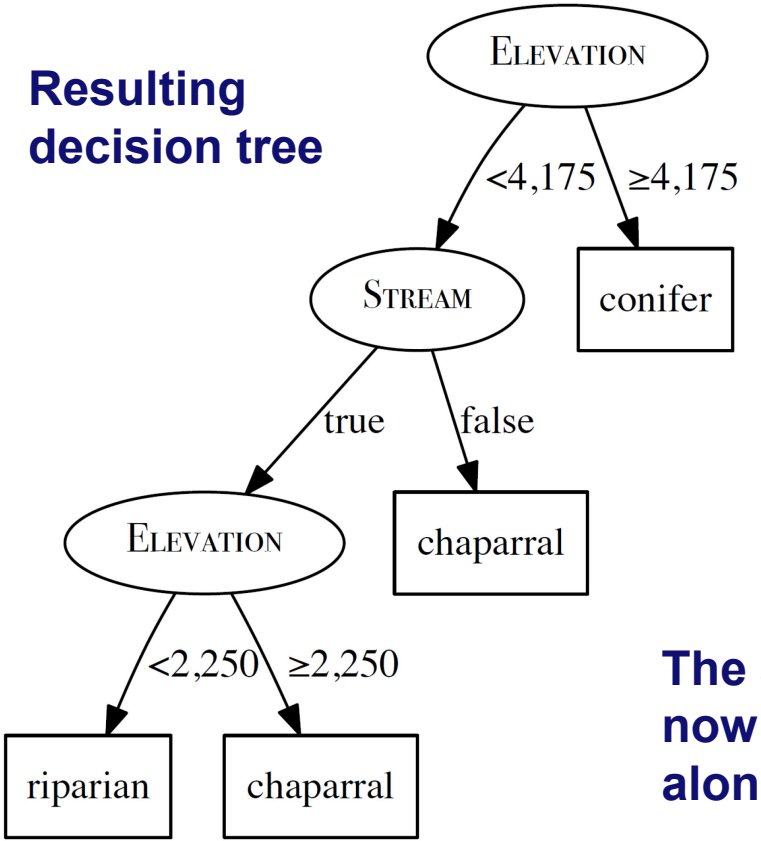
ID	Stream	Slope	Elevation	Vegetation
2	true	moderate	300	riparian
4	false	steep	1,200	chaparral
3	true	steep	1,500	riparian
7	true	steep	3,000	chaparral
1	false	steep	3,900	chaparral

D8

ID	Stream	Slope	Elevation	Vegetation
5	false	flat	4,450	conifer
6	true	steep	5,000	conifer

Example from book

Resulting decision tree



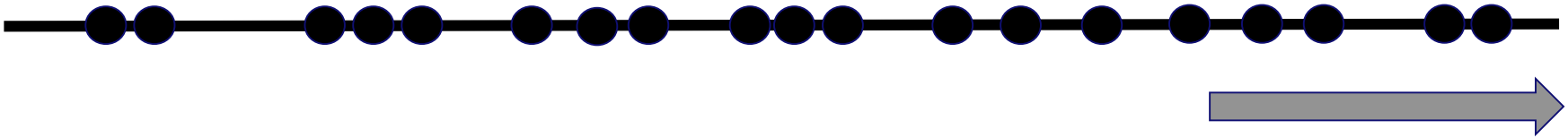
The same feature can now be used twice along a path!

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

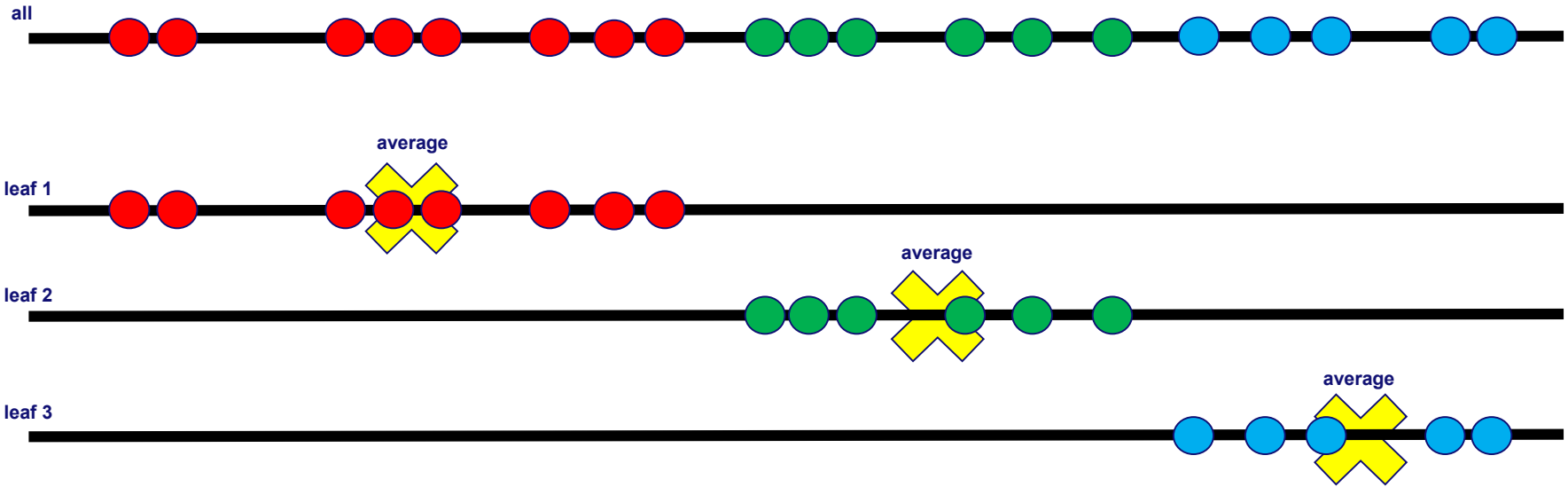
continuous target feature

Continuous target features



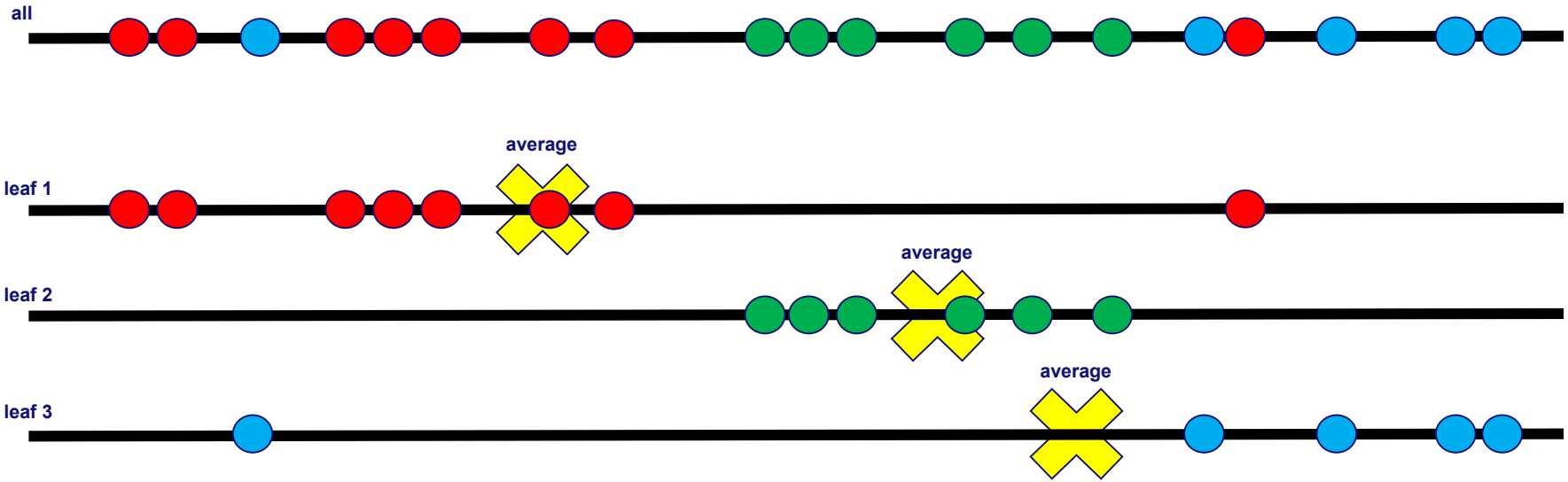
- We want to find descriptive features that “nicely” partition the target feature axis.
- Impurity = variance within a partition.
- We cannot use the target feature itself!
- Lets “color the dots” based on a selected descriptive feature.

Good classification



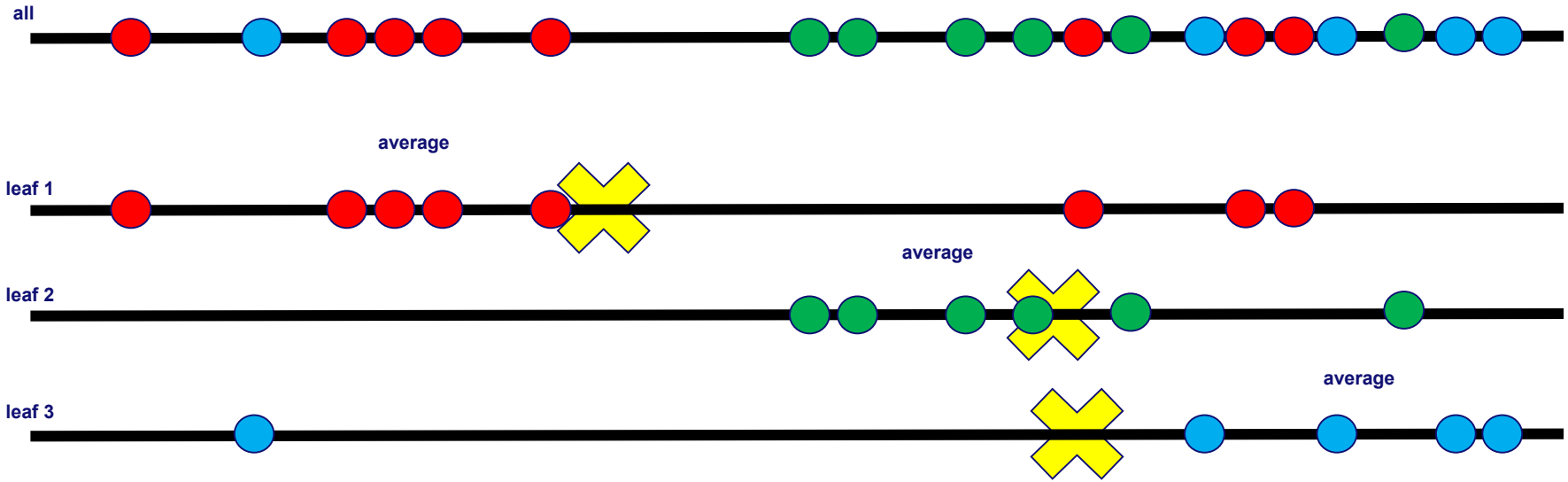
- Three leaves (red, green, blue show mapping).
- As before we need a measure of quality: impurity.
- Variance within a leaf of the decision tree.

Reasonable classification



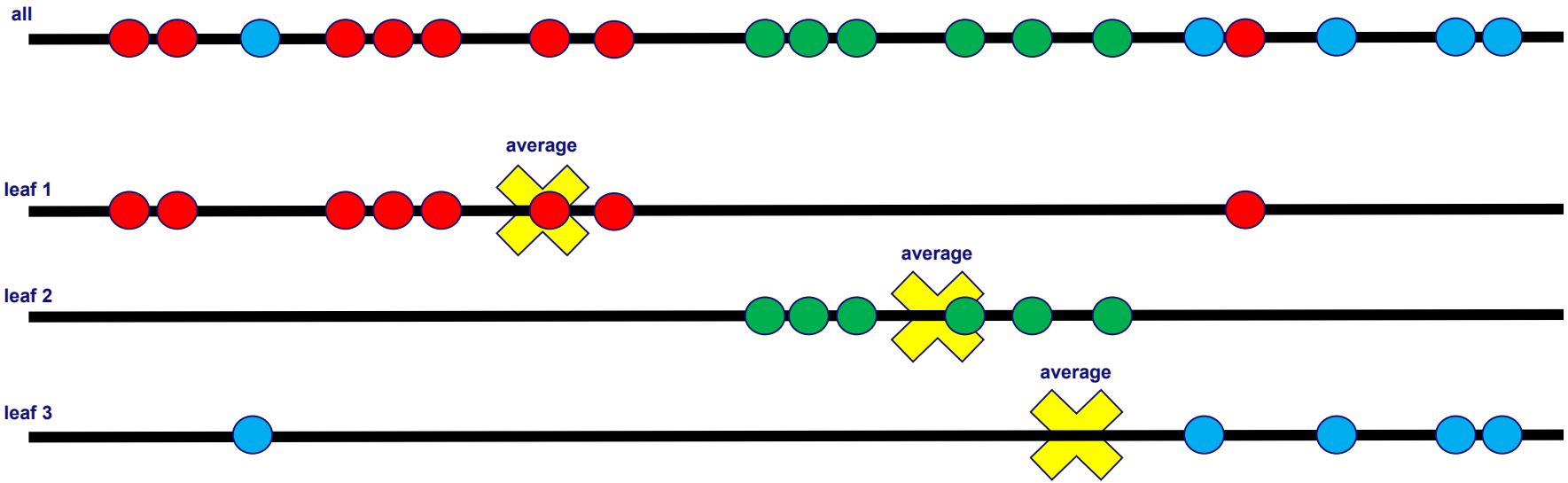
- **Variance within leaf 1 and leaf 3 increased with respect to the “good classification”.**

Poor classification



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

Impurity = variance in node/leaf



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

Adapting the ID3 algorithm: Basic idea

Algorithm 4.1 Pseudocode description of the ID3 algorithm.

Require: set of descriptive features **d**

Require: set of training ins

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

2: **return** a decision tr

3: **else if d** is empty **then**

4: **return** a decision t

majority target level

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

6: **return** a decision tree consisting af node with the label of the majority target level of the datas mmediate parent node

7: **else**

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

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

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

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

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

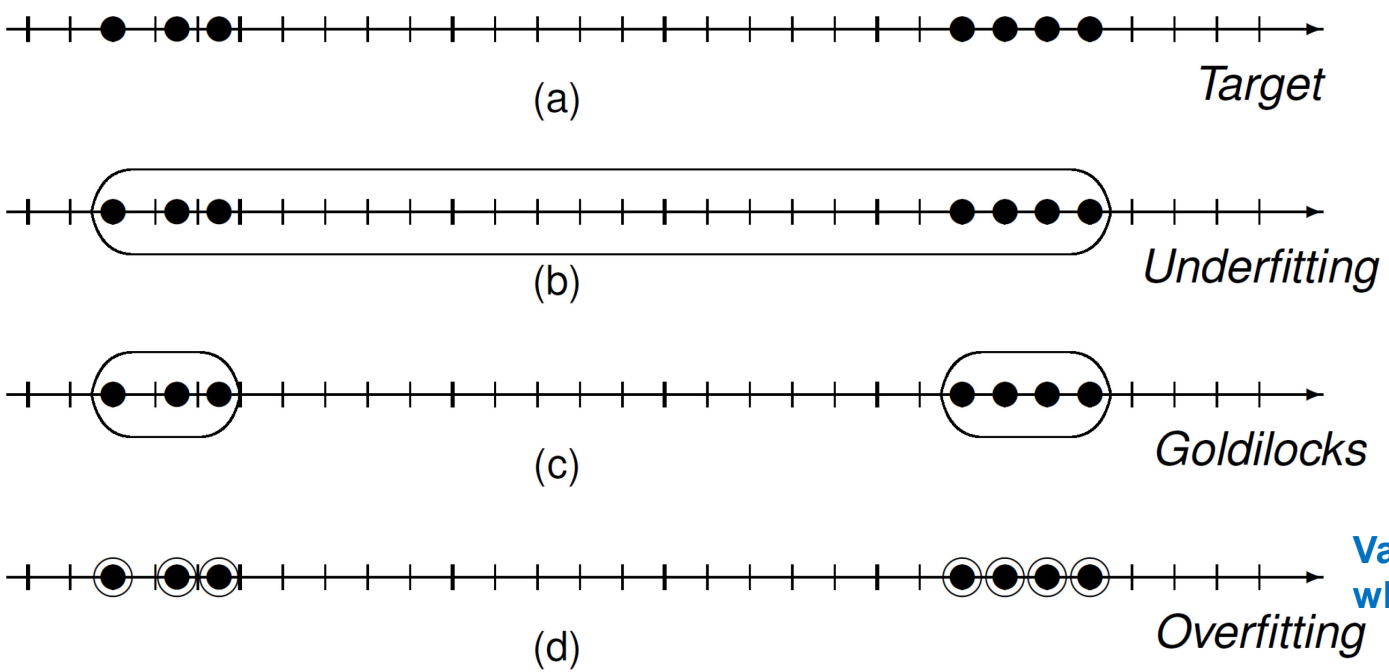
13: grow a branch from $Node_{\mathbf{d}[best]}$ to the decision tree created by rerunning *ID3* with $\mathcal{D} = \mathcal{D}_i$

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

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

Chair of Process and Data Science

Overfitting and underfitting



Variance gets smaller when sets get smaller

Avoid overfitting by stopping early enough!

Chair of Process and Data Science

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

Target feature = bike rentals per day

Split by Feature	Level	Part.	Instances	$\frac{\lvert\mathcal{D}_{d=l}\rvert}{\lvert\mathcal{D}\rvert}$	$var\,(t,\mathcal{D})$	Weighted Variance
Season	*'winter'*	$\mathcal{D}_1$	$\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$	0.25	2 692	$1\,379\,331\frac{1}{3}$
	'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	
	'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	$4\,026\,346\frac{1}{3}$	$2\,551\,813\frac{1}{3}$
	'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	

best split, lowers variance most

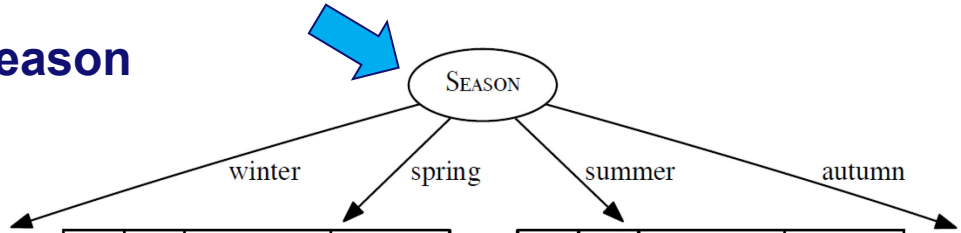
Example from book

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	$\frac{\lvert\mathcal{D}_{d=l}\rvert}{\lvert\mathcal{D}\rvert}$	$var(t,\mathcal{D})$	Weighted Variance
Season	*'winter'*	$\mathcal{D}_1$	$\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$	0.25	2 692	1 379 331 $\frac{1}{3}$
	'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	
	'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 $\frac{1}{3}$	2 551 813 $\frac{1}{3}$
	'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	

split based on Season



	ID	Work Day	Rentals
D1	1	false	800
	2	false	826
	3	true	900

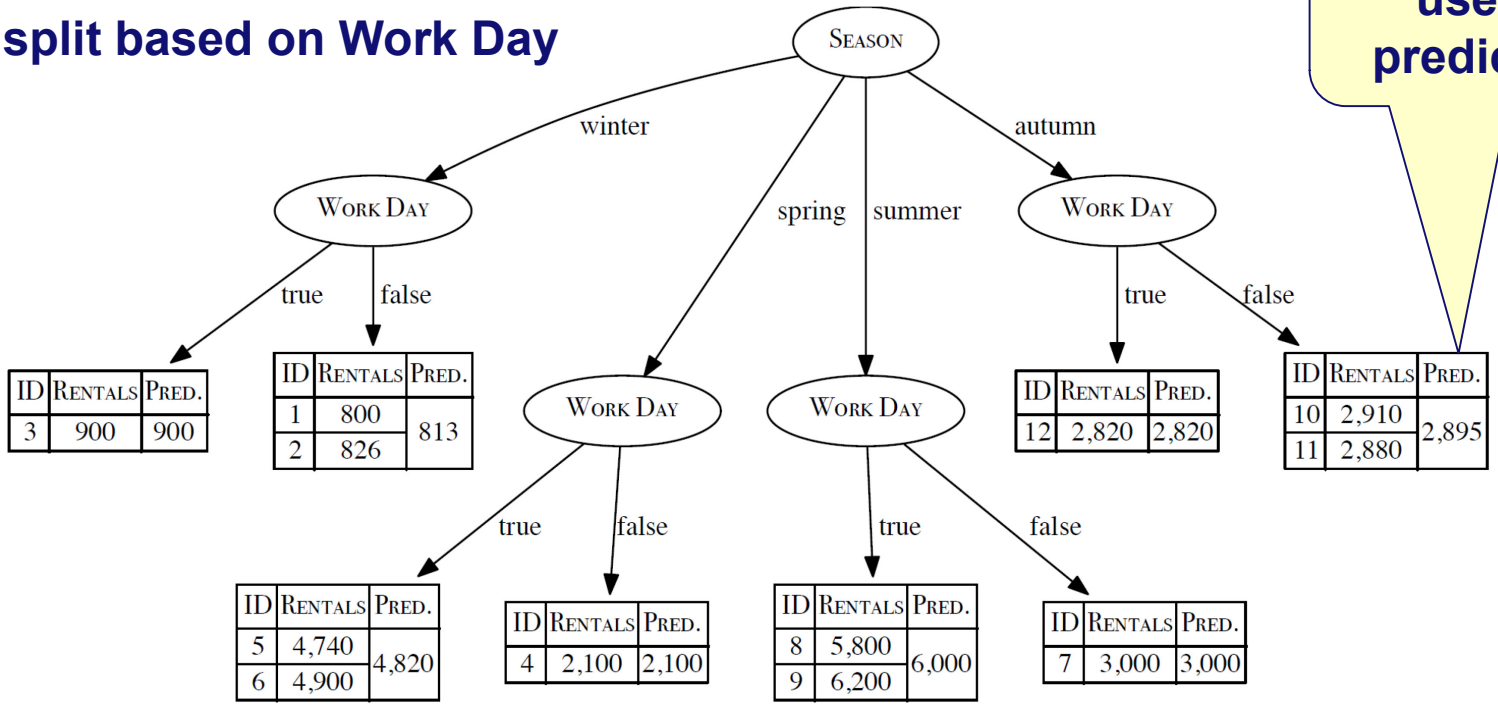
	ID	Work Day	Rentals
D2	4	false	2,100
	5	true	4,740
	6	true	4,900

	ID	Work Day	Rentals
D3	7	false	3,000
	8	true	5,800
	9	true	6,200

	ID	Work Day	Rentals
D4	10	false	2,910
	11	false	2,880
	12	true	2,820

Example from book

split based on Work Day



ID	Rentals	Pred.
3	900	900

ID	Rentals	Pred.
1	800	813
2	826	

ID	Rentals	Pred.
5	4,740	4,820
6	4,900	

ID	Rentals	Pred.
4	2,100	2,100

ID	Rentals	Pred.
8	5,800	6,000
9	6,200	

ID	Rentals	Pred.
7	3,000	3,000

ID	Rentals	Pred.
12	2,820	2,820

ID	Rentals	Pred.
10	2,910	2,895
11	2,880	

average of leaf is used as the predicted value

Many combinations of ideas are possible!

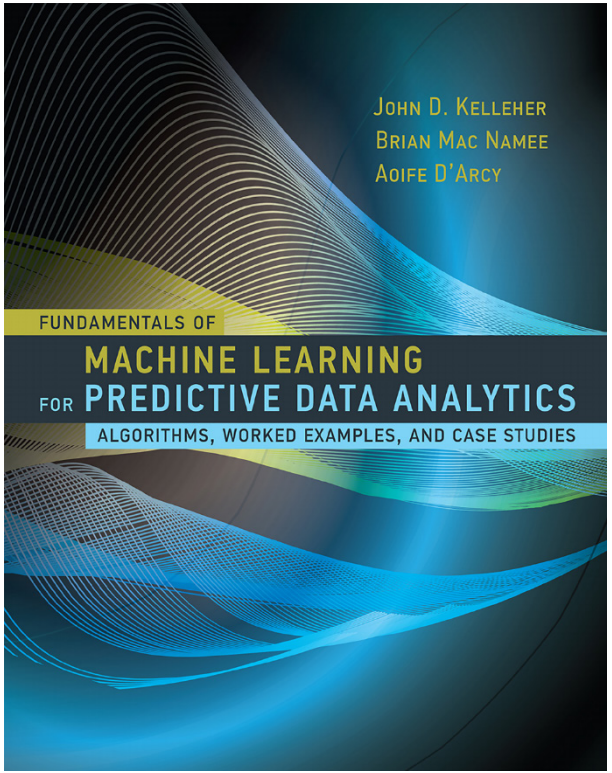
- **ID3**: The original.
- **C4.5**: Dealing with continuous features, missing values, post-pruning, etc.
- **C5.0**: Further development of C4.5 by Ross Quinlan.
- **J48**: Open source implementation of C4.5 (used in WEKA).
- **CART** (Classification and Regression Trees): Uses Gini index, handles continuous 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 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	10/10/2018	Wednesday
Lecture 2	Crash Course in Python	11/10/2018	Thursday
Instruction 1	*Python*	*12/10/2018*	*Friday*
Lecture 3	Basic data visualisation/exploration	17/10/2018	Wednesday
Lecture 4	Decision trees	18/10/2018	Thursday
Instruction 2	*Decision trees and data visualization/explora*	*19/10/2018*	*Friday*
Lecture 5	Regression	24/10/2018	Wednesday
Lecture 6	Support vector machines	25/10/2018	Thursday
Instruction 3	*Regression and support vector machines*	*26/10/2018*	*Friday*
Lecture 7	Neural networks (1/2)	31/10/2018	Wednesday
Instruction 4	*Neural networks and supervised learning*	*02/11/2018*	*Friday*
Lecture 8	Neural networks (2/2)	07/11/2018	Wednesday
Lecture 9	Evaluation of supervised learning problems	08/11/2018	Thursday
Instruction 5	*Neural networks and supervised learning*	*09/11/2018*	*Friday*
Lecture 10	Clustering	14/11/2018	Wednesday
Lecture 11	Frequent items sets	15/11/2018	Thursday
Lecture 12	Association rules	21/11/2018	Wednesday
Lecture 13	Sequence mining	22/11/2018	Thursday
Instruction 6	*Clustering, frequent items sets, association ru*	*23/11/2018*	*Friday*
Lecture 14	Process mining (unsupervised)	28/11/2018	Wednesday
Lecture 15	Process mining (supervised)	29/11/2018	Thursday
Instruction 7	*Process mining and sequence mining*	*30/11/2018*	*Friday*
Lecture			
Instruction 8			
Lecture			
Lecture			
Lecture			
backup			
Instruction 9			
Lecture			
Lecture			
Instruction 10			
Lecture			
Lecture			
Instruction 11			
Lecture			
backup			
Instruction 12			
backup			
backup			
extra	*Question hour*	*01/02/2019*	*Friday*

Lecture 4	Decision trees	18/10/2018	Thursday
Instruction 2	*Decision trees and data visualization/explora*	*19/10/2018*	*Friday*
Lecture 5	Regression	24/10/2018	Wednesday
Lecture 6	Support vector machines	25/10/2018	Thursday
Instruction 3	*Regression and support vector machines*	*26/10/2018*	*Friday*
Lecture 7	Neural networks (1/2)	31/10/2018	Wednesday
Instruction 4	*Neural networks and supervised learning*	*02/11/2018*	*Friday*
Lecture 8	Neural networks (2/2)	07/11/2018	Wednesday
Lecture 9	Evaluation of supervised learning problems	08/11/2018	Thursday