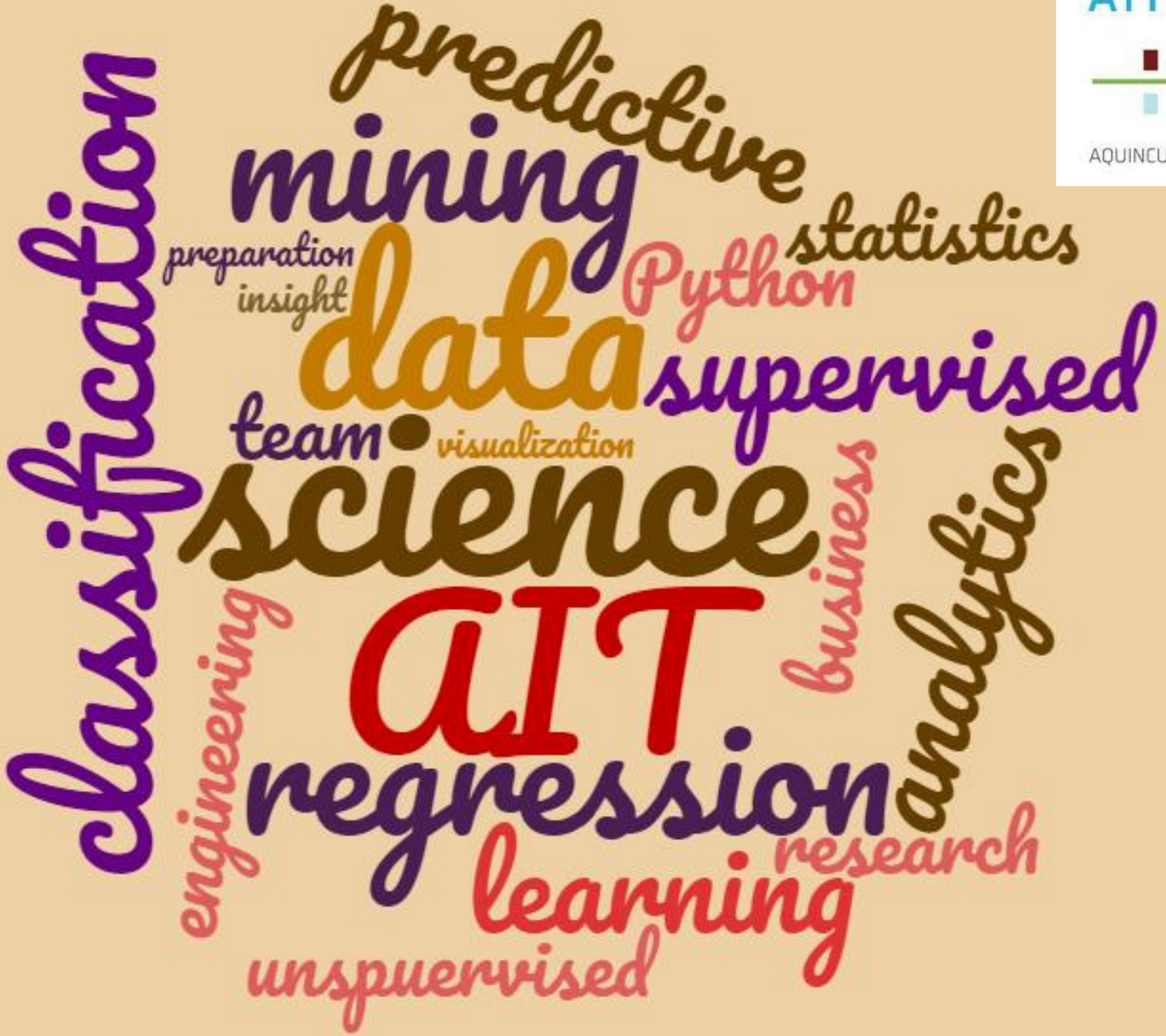
Data Science



AIT-BUDAPEST
AQUINCUM INSTITUTE OF TECHNOLOGY

Roland Molontay

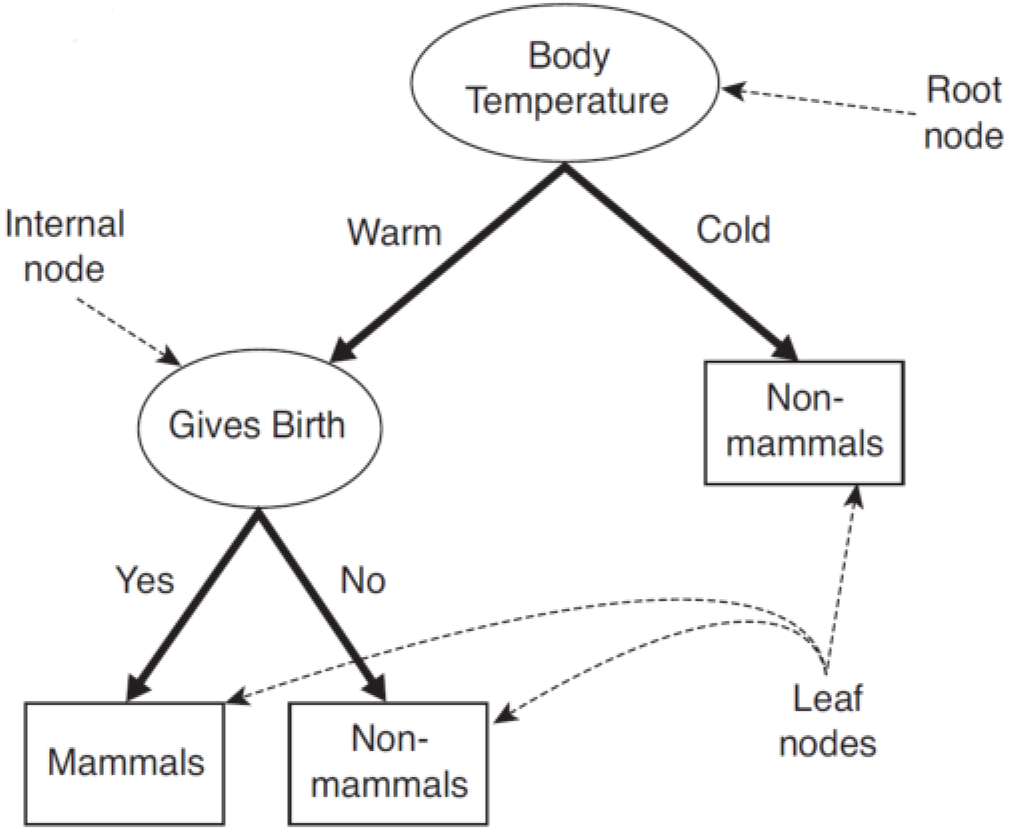
February 27, 2023.
Decision tree

Schedule of the semester

	Monday midnight	*Tuesday class*	*Friday class*
W1 (02/06)			
W2 (02/13)		HW1 out	
W3 (02/20)			
W4 (02/27)	HW1 deadline + TEAMS	HW2 out	
W5 (03/06)	PROJECT PLAN		
W6 (03/13)	HW2 deadline	HW3 out	
W7 (03/20)			MIDTERM
SPRING BREAK		SPRING BREAK	SPRING BREAK
W8 (04/03)	HW3 deadline		GOOD FRIDAY
W9 (04/10)	MILESTONE 1	HW4 out	
W10 (04/17)			
W11 (04/24)	HW4 deadline		
W12 (05/01)	MILESTONE 2		
W13 (05/08)			
W14 (05/15)		FINAL	PROJECT presentations
W15 (05/22)		PROJECT presentations	

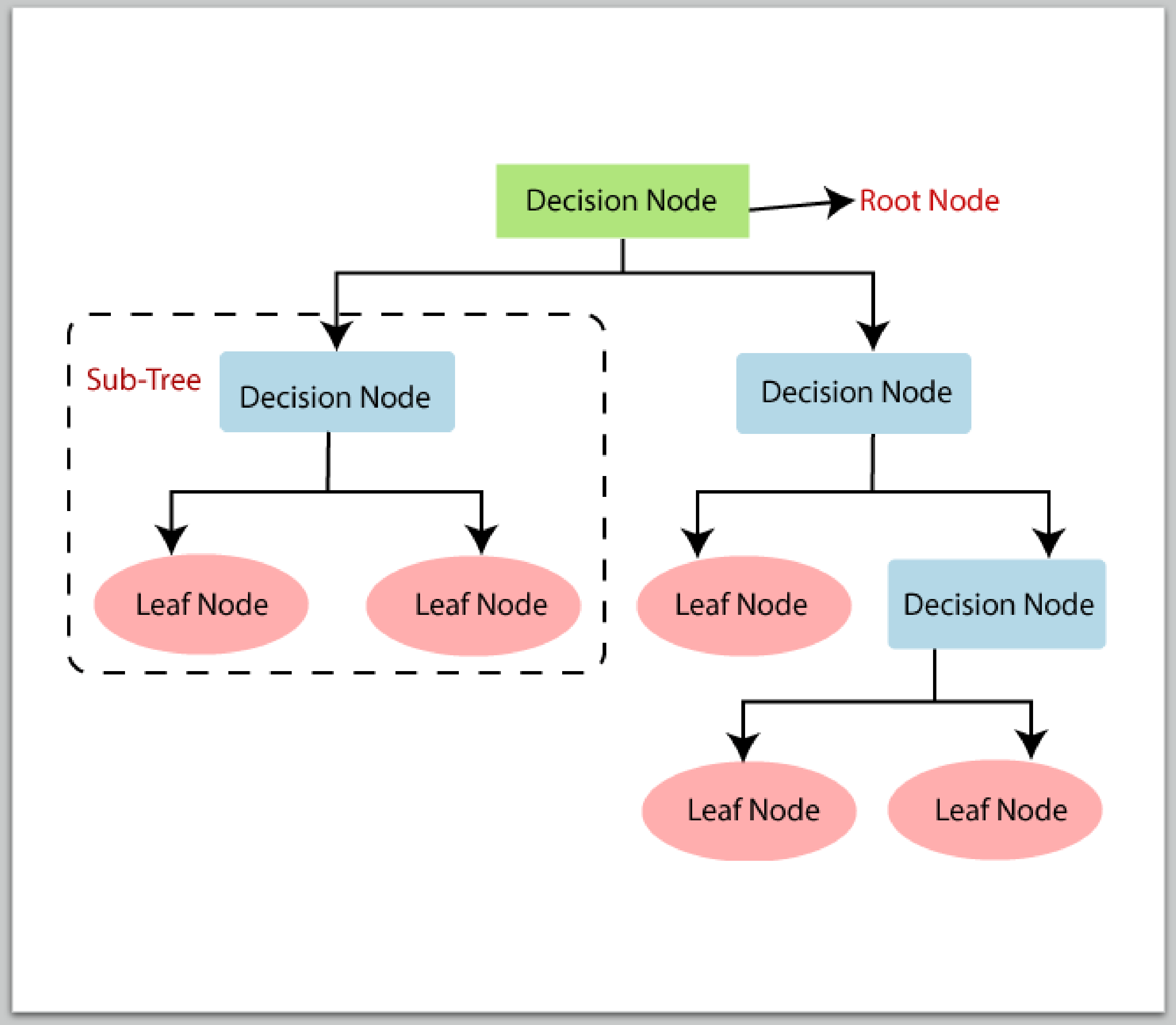
Decision tree

- Rooted, directed (sometimes binary) tree
- Each inner node has a decision rule that assigns instances uniquely to child nodes of the actual node
- Each leaf node has a class label



Prediction with decision tree

- Decision tree is one of the oldest predictive algorithm
    - We start at root node
    - At each interior node we evaluate the decision rule, branch to the child node picked by the decision rule
    - Once a leaf node is reached, predict the label assigned to that node
- It is mainly used for classification, but also suitable for regression problems

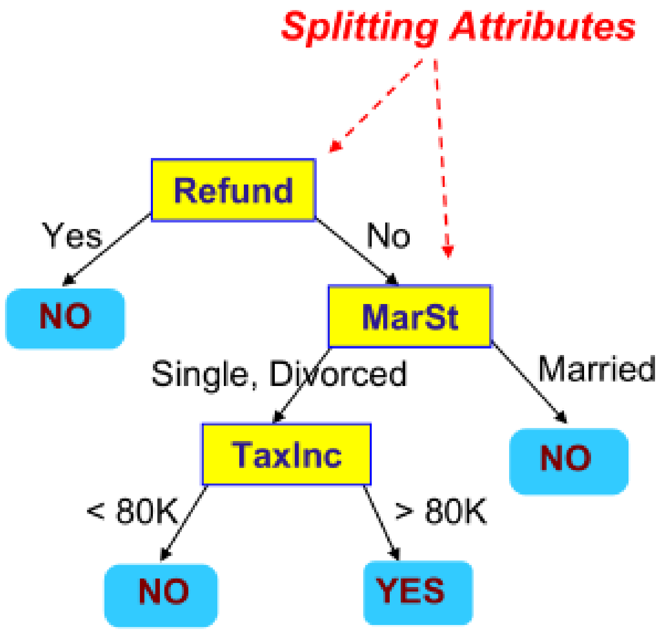


Decision tree - learning phase

It works with continuous and categorical attributes

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

Training Data

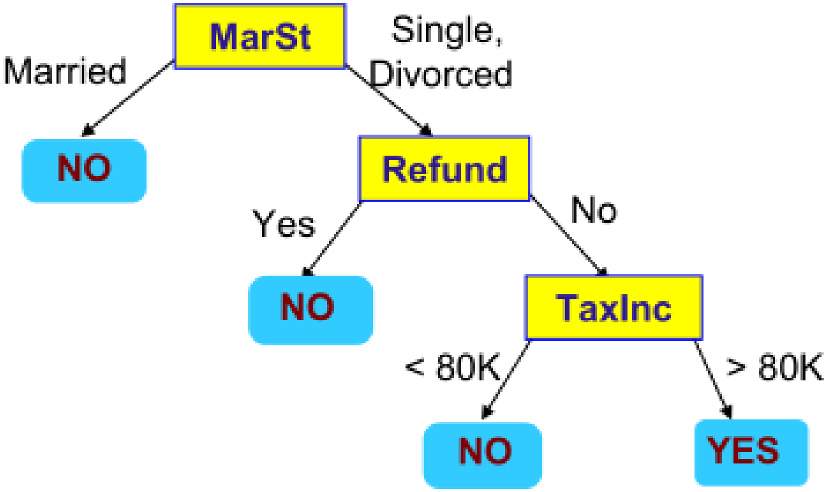


Model: Decision Tree

Another tree for the same data

categorical categorical continuous class

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

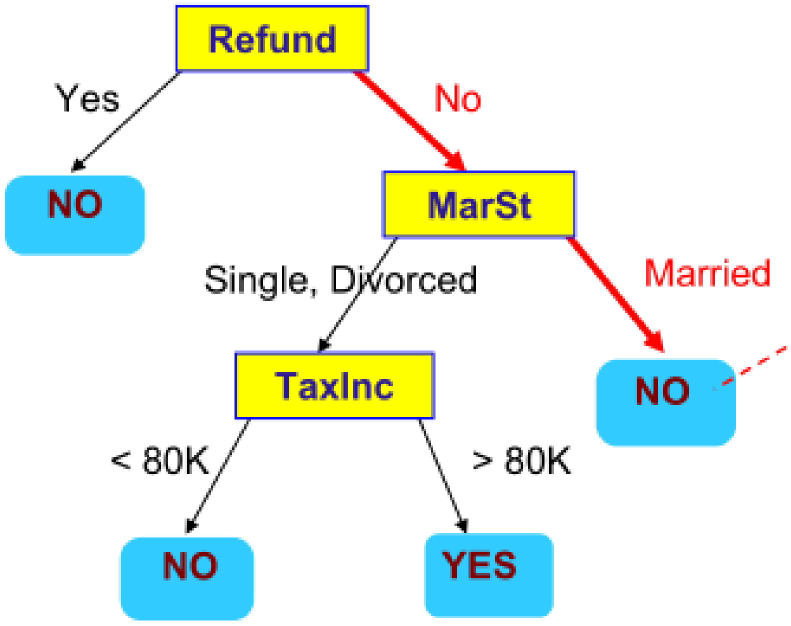


There could be more than one tree that fits the same data!

Applying the model to new data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to “No”

Problems

Ex 1

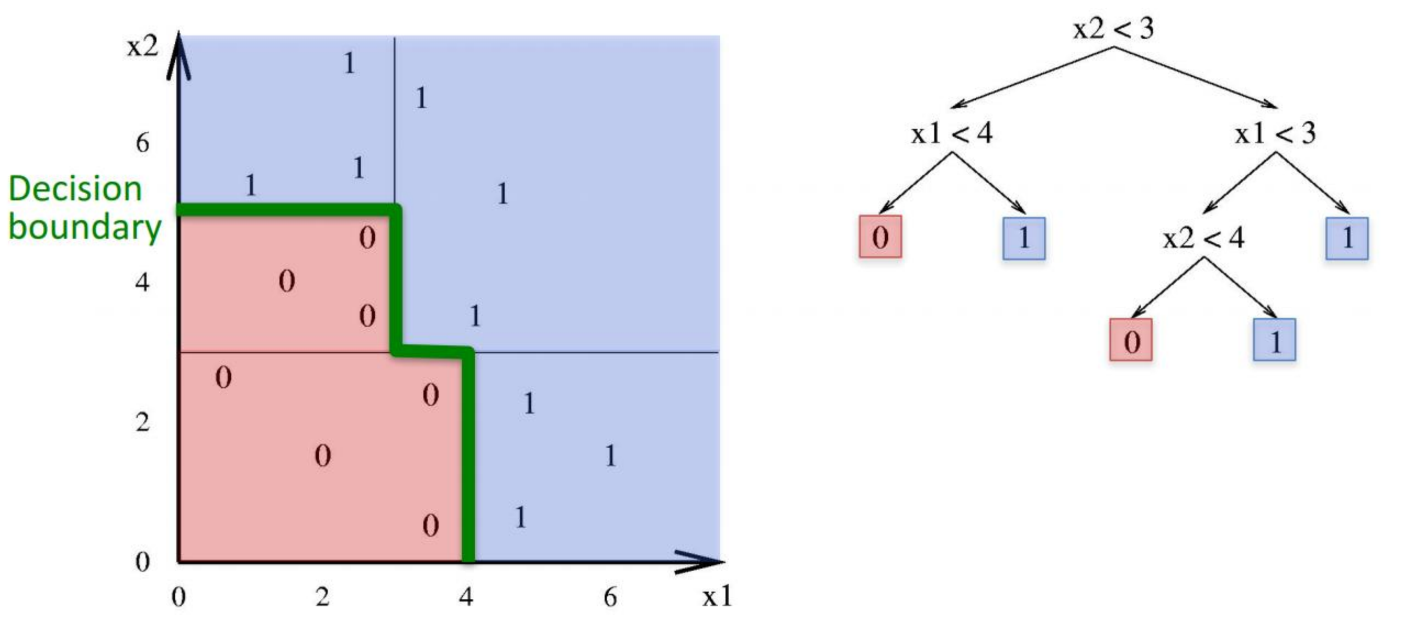
How many logical (Boolean, $f : \{0,1\}^N \to \{0,1\}$)) functions can be generated on N binary attributes? What are the possible functions for $N = 2$?

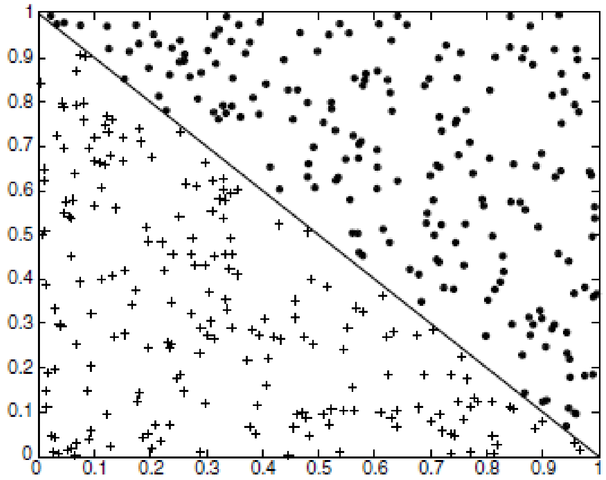
Ex 2

Can decision trees learn logical (Boolean) functions? How to represent the following functions with a decision tree: A **OR** B, A **AND** B, A **XOR** B, where A and B are logical variables.

Partitioning the feature space

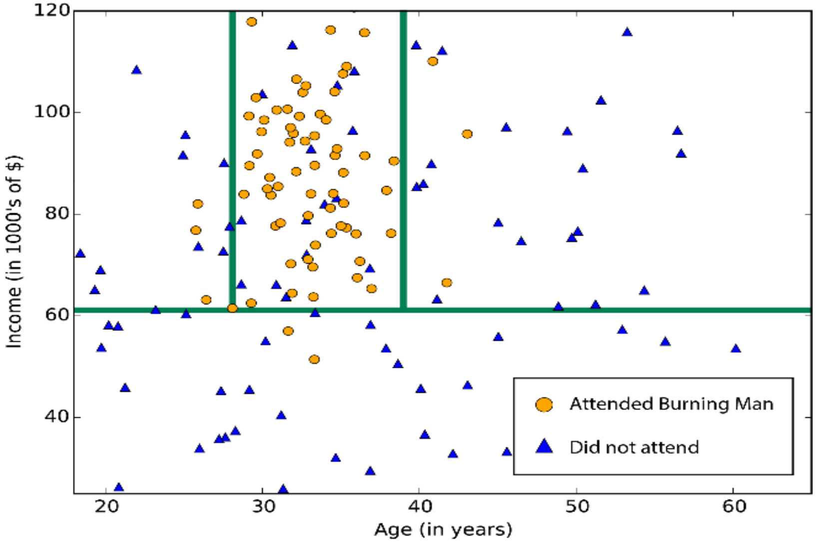
The boundaries are always parallel to the axes ➔ the decision regions are always unions of (hyper-)rectangles

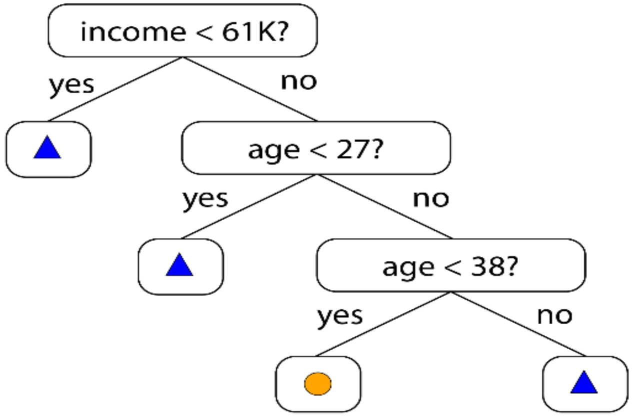


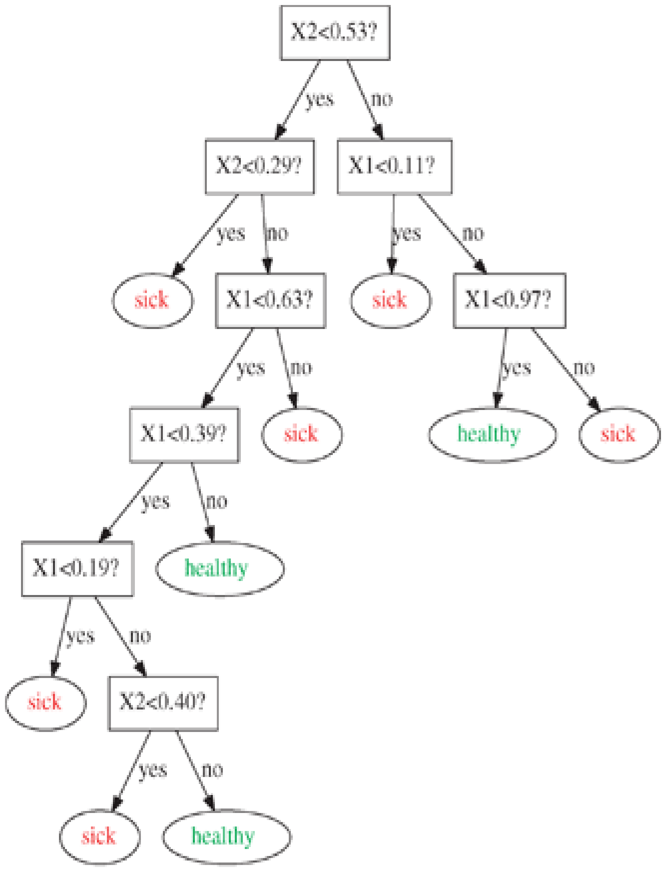


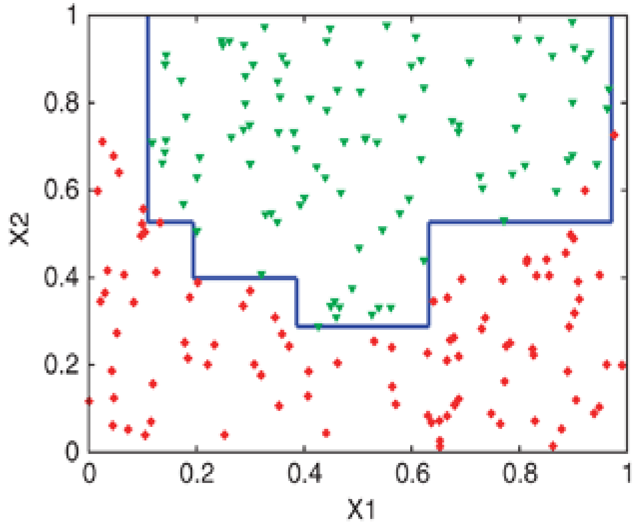
How could a decision tree represent a data set on the left?

Partitioning the feature space - examples









Algorithm for building decision trees

- We consider a general algorithm
  - Special versions of the algorithm are implemented in various programs
- We do not aim to find the best tree (it is underdefined what is best) but a good enough tree
  - There are exponentially many possible trees
- With a greedy method, deciding locally, quickly
- There are various algorithms (and their variants)
  - Hunt algorithm
  - CART
  - ID3, C4.5 (J48)
  - SLIQ, SPRINT

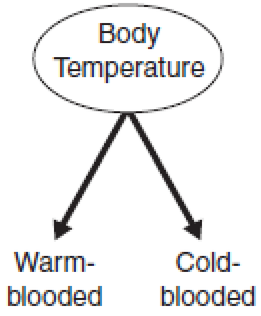
Sketch of Hunt algorithm

- We start with one node (the root), all the records are there
    - Its label is the majority label
- Further steps: we choose a node that is worth splitting on
- End: until there is no node worth splitting on
    - In each node all the records have the same label
    - There are no good splits, e.g. in all nodes, all the records agree in each attribute (aside from the label)

Hunt algorithm - questions

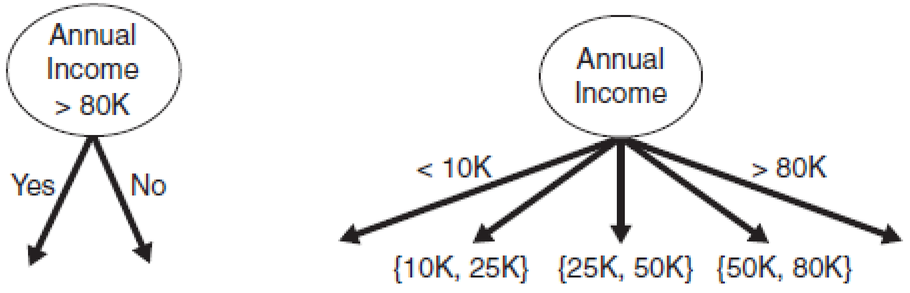
- When does it terminate?
  - When no more nodes left that are worth splitting on
- When is it not worth splitting on?
  - In each node all the records have the same label
  - There are no good splits, e.g. in all nodes, all the records agree in each attribute (aside from the label)
  - If we want to avoid to have a too deep tree (more details later)
- Which node to split on if there are more possibilities?
  - E.g. traversing nodes according to BFS (breadth-first search), DFS (depth-first search)
- How to split?
  - Splitting a node should increase the homogeneity (with respect to the label) of resultant sub-nodes
- How to decide on the label?
  - Using majority voting

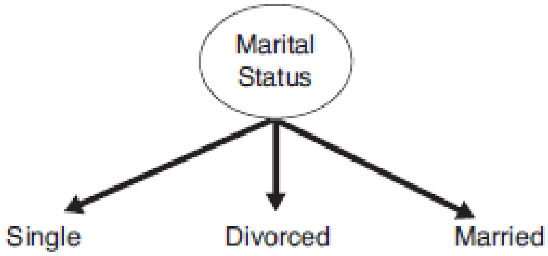
Possible splits



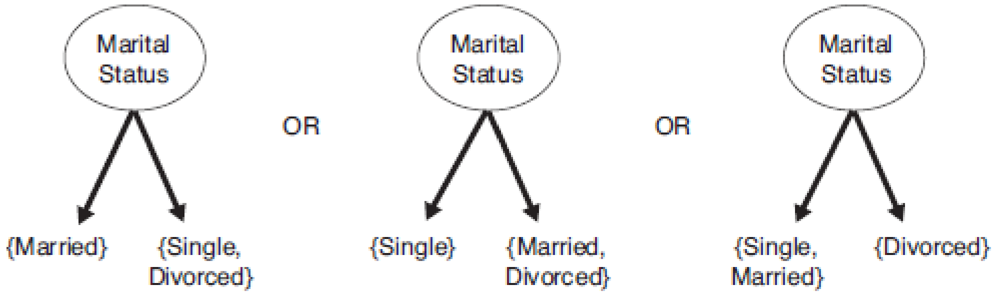
Binary

Numerical

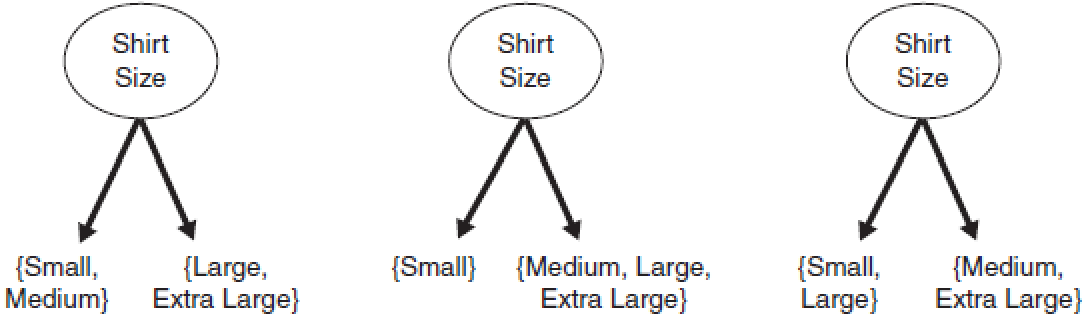




Nominal

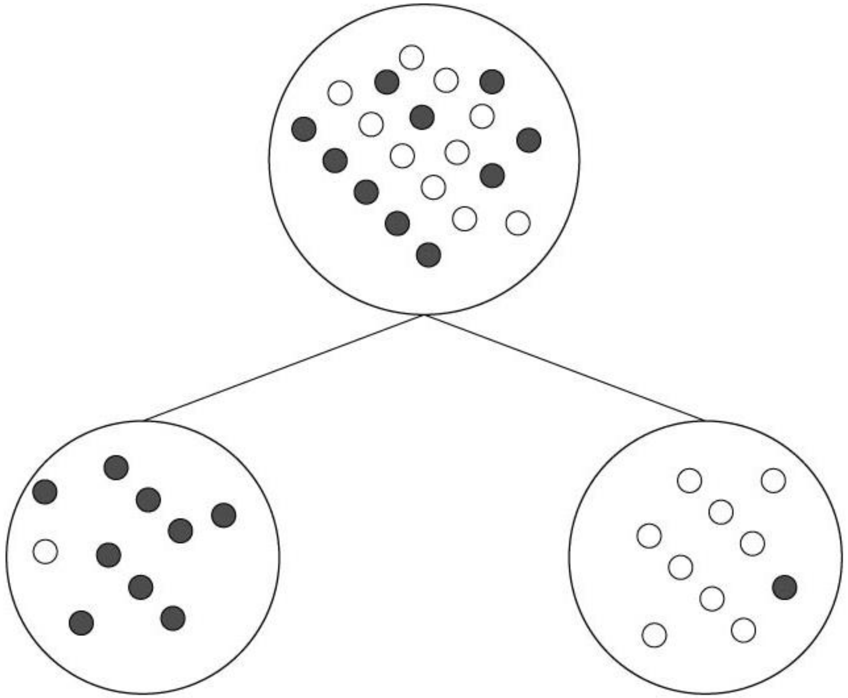


Ordinal



What defines a good split?

- It increases the homogeneity („purity”) of the target variable in the emerging child nodes compared to the parent node
- A good split creates child nodes of similar size (or at least does not create very small nodes)



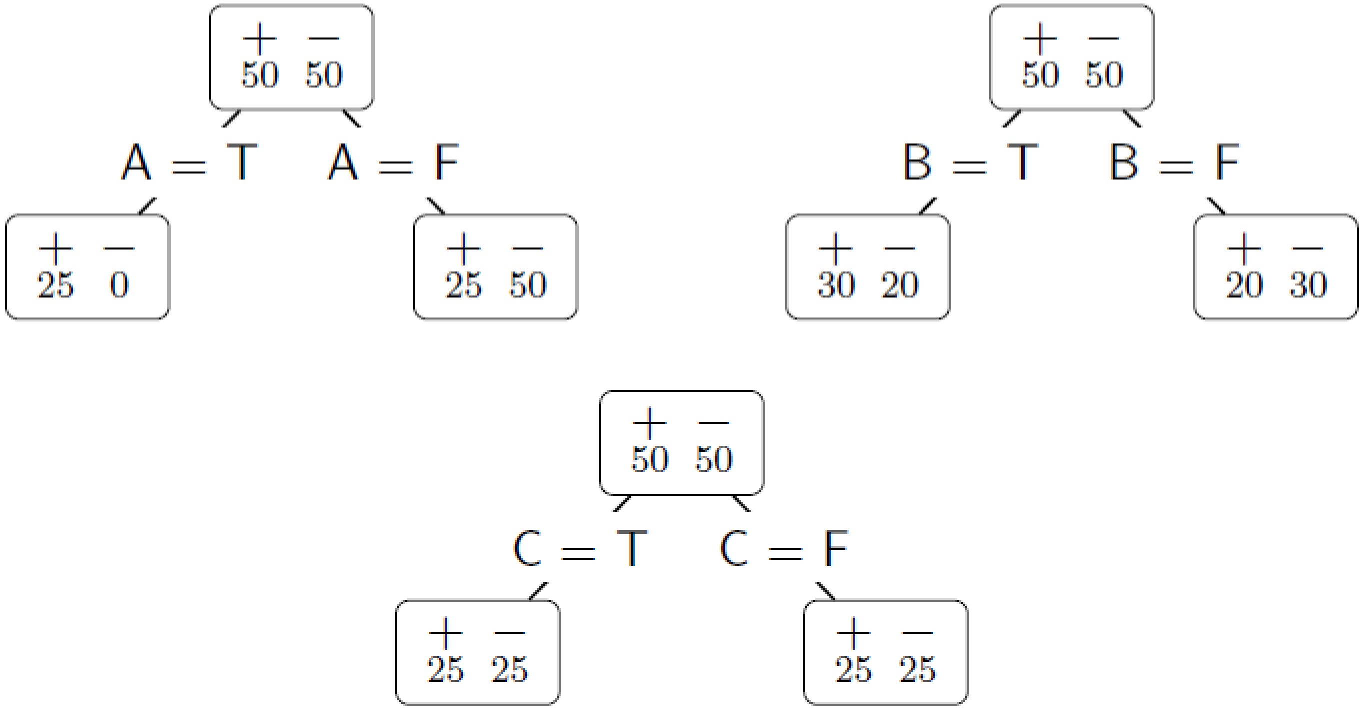
Example

- The following table summarizes a data set with three binary attributes (A, B, C) and two class labels (+, -).
- Which is the best split? What attribute would you split on?

A	B	C	Number of instances class: +	Number of instances class: −
T	T	T	5	0
F	T	T	0	20
T	F	T	20	0
F	F	T	0	5
T	T	F	0	0
F	T	F	25	0
T	F	F	0	0
F	F	F	0	25

Possible splits

A	B	C	Number of instances	
			class: +	class: −
T	T	T	5	0
F	T	T	0	20
T	F	T	20	0
F	F	T	0	5
T	T	F	0	0
F	T	F	25	0
T	F	F	0	0
F	F	F	0	25

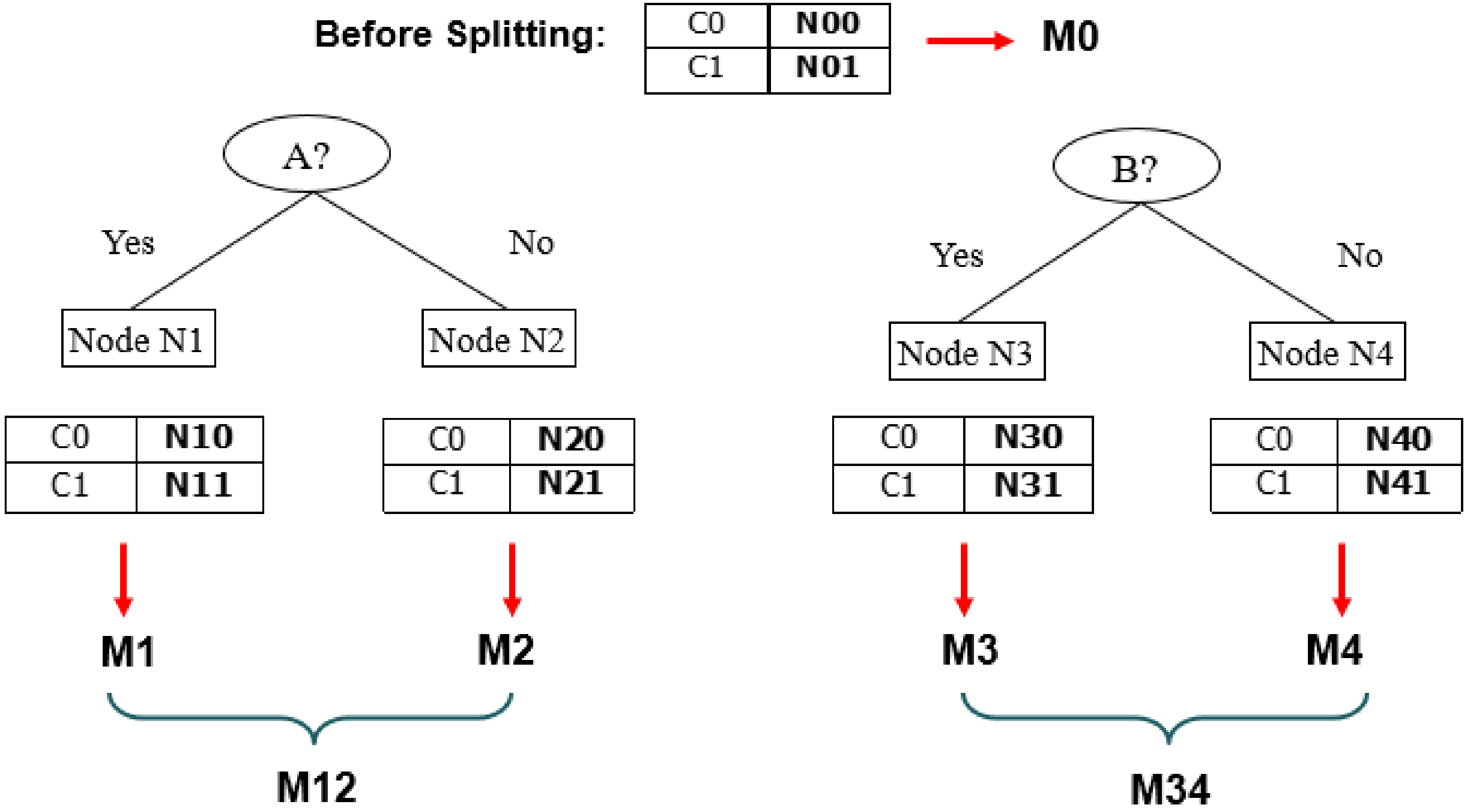


A is the best split!

How to measure the goodness of a split?

- We consider three possible metrics
- The main principle is the same for all:
  - We define a metric for a set of records that measures the degree of homogeneity (purity) of the target variable within that set
    - We consider three metrics: Gini coefficient, entropy, misclassification error
  - The goodness of a split is measured by the difference between the impurity of the parent node and the emergent child nodes
    - How much do we gain if we split the node? What is the degree of homogeneity increase?
    - We can also consider the size of the emerging child nodes, punishing the too small splits

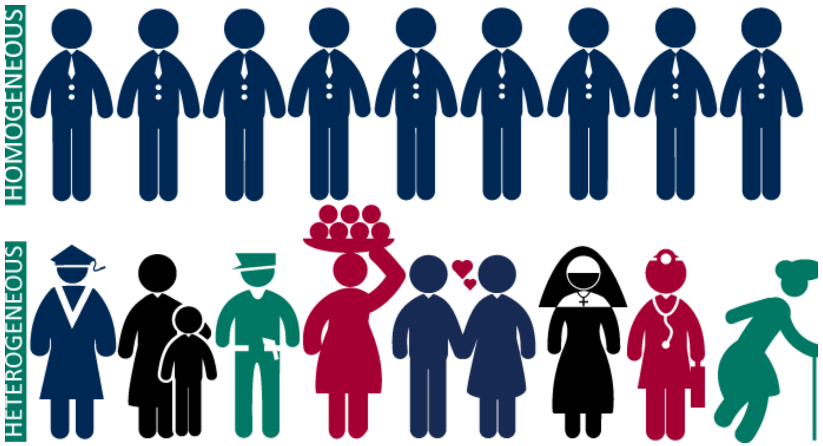
Which split is better?



Gain = M0 – M12 vs M0 – M34

Measuring inhomogeneity

- Let t be the node of a decision tree (i.e. a set of records), we aim to measure its homogeneity with respect to a target variable that has c possible values
- Let $p(i|t)$ denote the relative frequency of records with label i in node t



Misclassification error

- Misclassification error or classification error:

$$\text{Classification error }(t) = 1 - \max_i[p(i|t)]$$

- Its maximal value is: 1-1/c, when the records are distributed equally in all classes
- Its minimal value is 0, when all the records have the same label

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

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

Error = 1 – max (0, 1) = 1 – 1 = 0

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

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

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

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

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

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

Gini coefficient

- Gini coefficient

$$\text{Gini}(t) = 1 - \sum_{i=1}^{c} p(i|t)^2$$

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

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

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

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

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

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

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

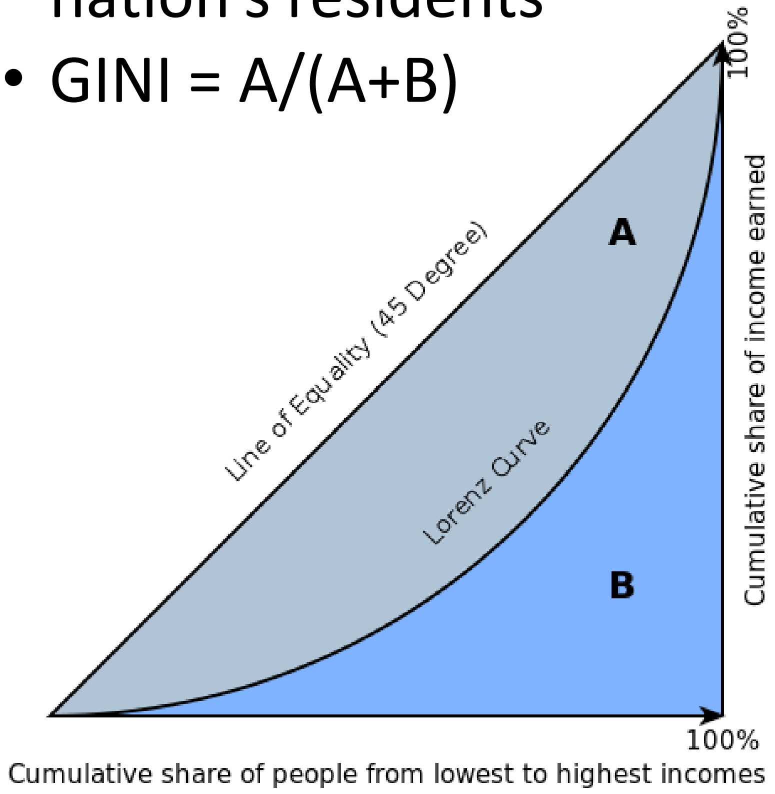
P(C1) = 2/6   P(C2) = 4/6

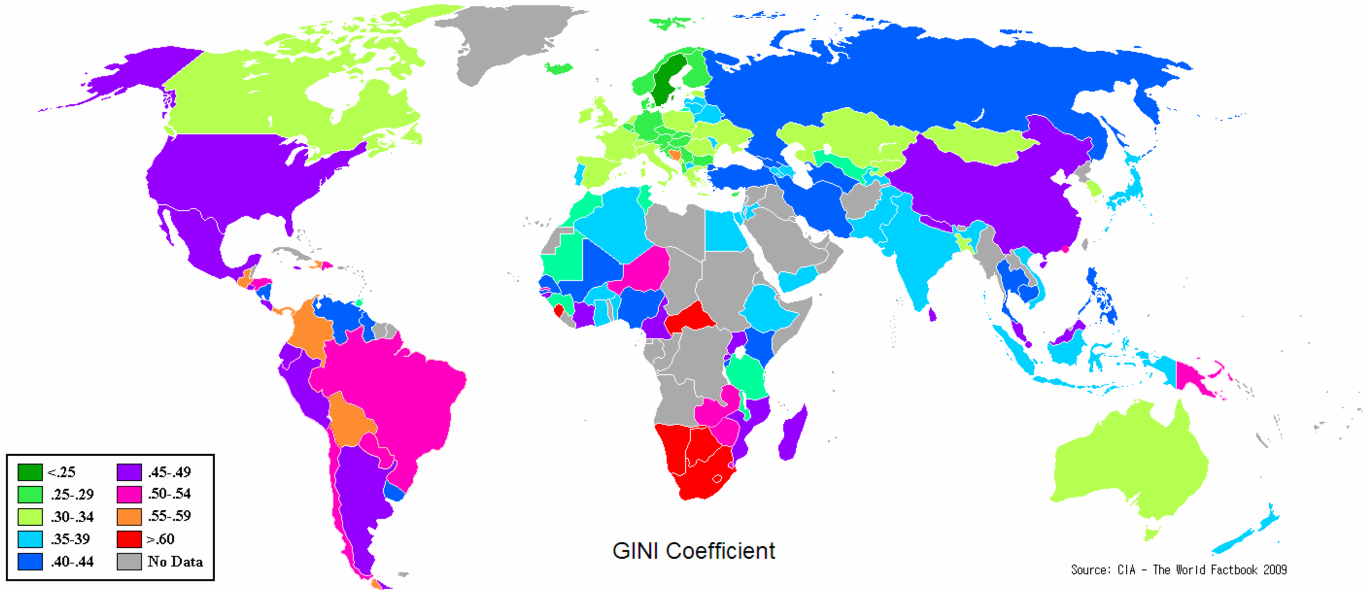
Gini = 1 – $(2/6)^2$ – $(4/6)^2$ = 0.444

- Its value ranges between 0 and 1 - 1/*c*
- Usually Gini is default setting for splitting criterion

Outlook – Gini coefficient in economics

- A similar concept but not the same, do not mix them
- It is used to measure the inequality in income or wealth distribution of a nation’s residents
- GINI = A/(A+B)





Entropy

$$\text{Entropy}(t) = -\sum_{i=1}^{c} p(i|t) \log_2 p(i|t)$$

- Its value ranges between 0 and $log_2 c$

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

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

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

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

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

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

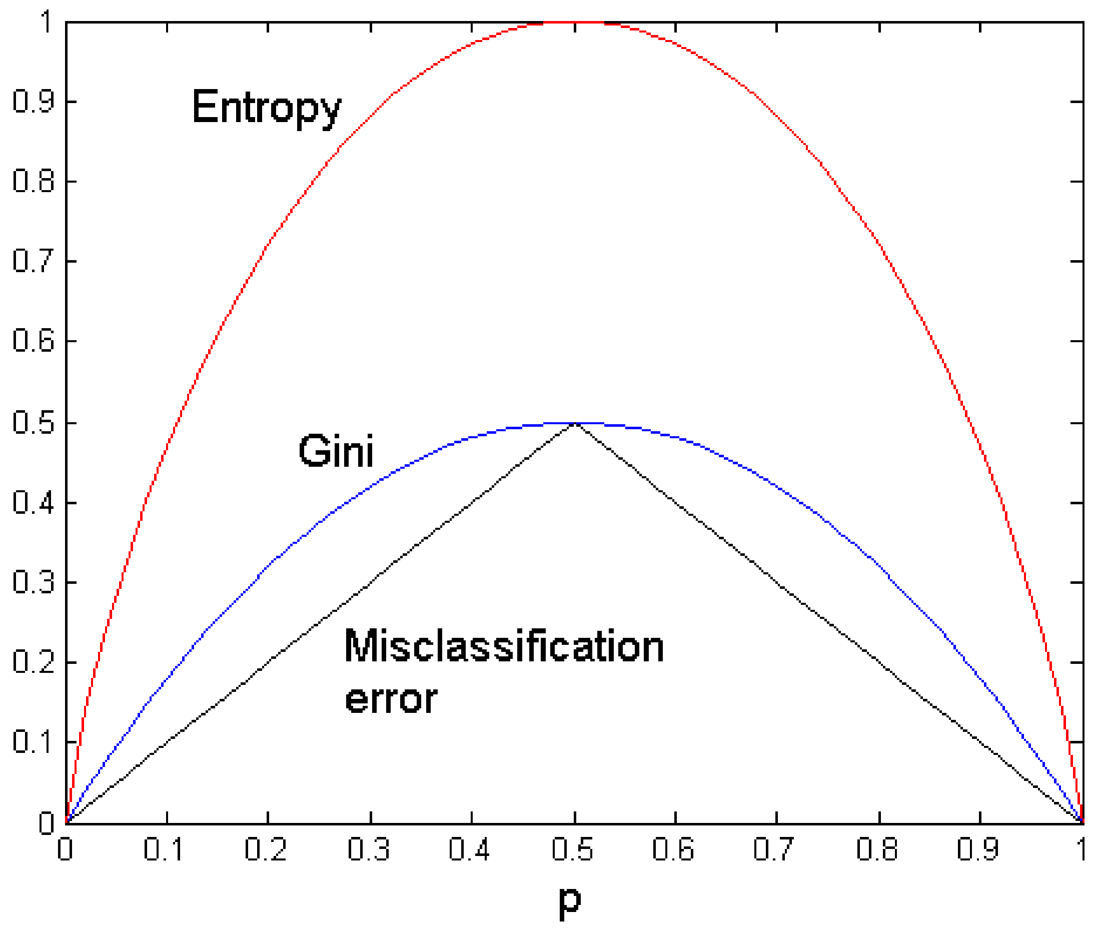
C1	**2**
C2	**4**

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

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

Comparing homogeneity metrics

For binary target variable (c=2)



The goodness of a split – the gain

- How much do we gain by the split?
  - Gain (Δ)

$$\Delta = I(parent) - \sum_{i=1}^{k} \frac{n_i}{n} I(child_i)$$

  - *I():* one of the three inhomogeneity metrics
  - n_i: number of records in child node *i*, *n*: number of records in the parent node
    - We weight the inhomogeneity of the child nodes by their relative size

Problem

- The following table summarizes a data set with three binary attributes (A, B, C) and two class labels (+, -).
- Using misclassication error as inhomogenity measure, calculate the gains for splitting on each attribute. Which attribute gives the best split?

A	B	C	Number of instances	
			class: +	class: −
T	T	T	5	0
F	T	T	0	20
T	F	T	20	0
F	F	T	0	5
T	T	F	0	0
F	T	F	25	0
T	F	F	0	0
F	F	F	0	25

Splitting by a continuous attribute

- Where to split?
- Sort the records according to the attribute values
- Calculate the inhomogeneity metric one by one increasing the value of the cut point
  - Which cut-point corresponds to the highest gain?

Class	No	No	No	Yes	Yes	Yes	No	No	No	No
	Annual Income									
Sorted Values →	60	70	75	85	90	95	100	120	125	220

Split Positions →	55		65		72		80		87		92		97		110		122		172		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		*0.300*		0.343		0.375		0.400		0.420	

Should we split on ID?

- Δ gain would the highest if we split on ID
- The method thus prefers to split the parent node to many small nodes
- It is usually not favorable
  - E.g. if we split on ID, that is useless
  - If the emergent child nodes are too small the model is more likely to have a worse generalization ability
- Possible solutions
  - Only allow for binary splits
  - Filter out those attributes that are not reasonable to split on
  - Instead of Δ gain use an other method to measure the goodness of split: gain ratio

Building a decision tree

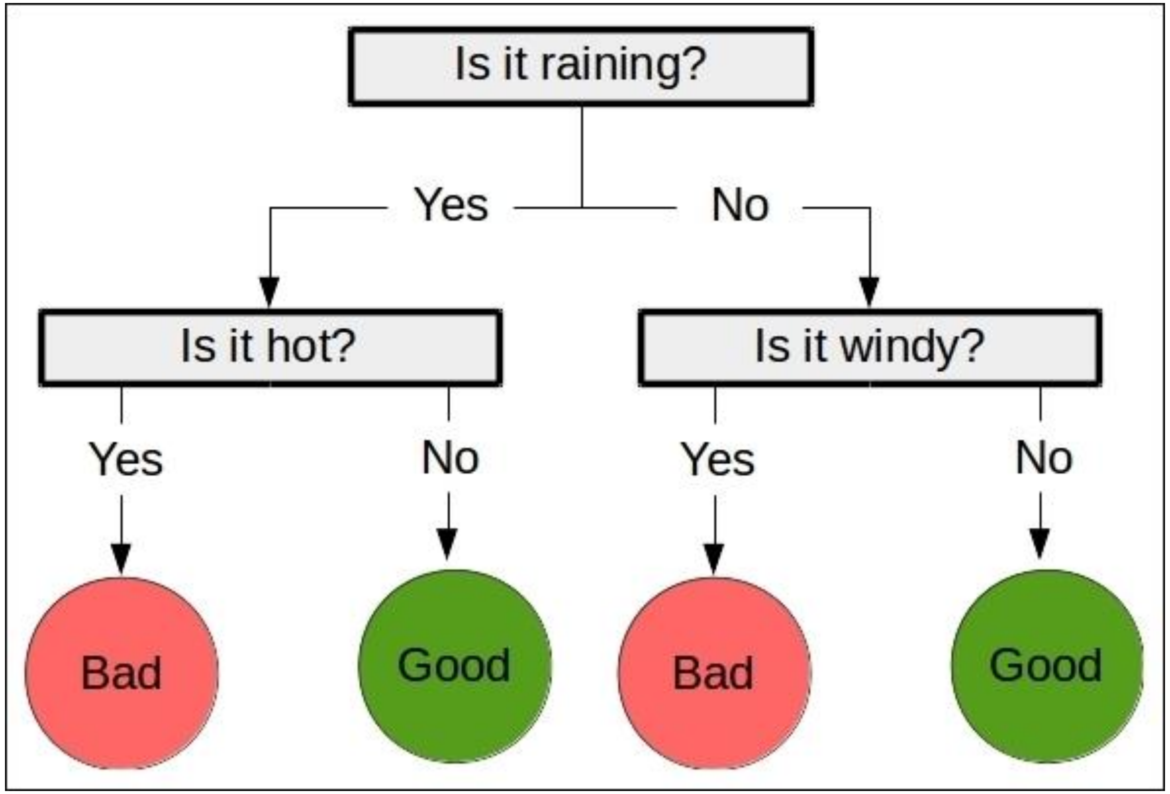
- Relatively fast, easy to interpret
- After building the decision tree, it is fast to predict new labels for unseen data points
- There are not many hyperparameter to set (but there are some)
  - What is the measure of inhomogeneity?
  - Do we allow for multi-split or just for binary?
  - How to traverse nodes?

Termination criteria

- If every node is homogeneous
  - Or every node is quasi-homogeneous (almost homogeneous)
- All the records agree in each attribute (aside from the label)
- Global termination criteria:
  - Bound for the number of leaf nodes
  - Bound for the number of levels (depth of the tree)
- We do not search for the „best” tree
  - It is not practical since it is just the „best” on the training data set

Advantages of decision tree

- Easy to interpret
- It works with a similar approach as human decision process
- Easy to visualize
- Insensitive to irrelevant attributes
- It can be used for a wide variety of problems (for classification/regression, with numerical/categorical attributes)



Acknowledgement

- András Benczúr, Róbert Pálovics, SZTAKI-AIT, DM1-2
- Krisztián Buza, MTA-BME, VISZJV68
- Bálint Daróczy, SZTAKI-BME, VISZAMA01
- Judit Csima, BME, VISZM185
- Gábor Horváth, Péter Antal, BME, VIMMD294, VIMIA313
- Lukács András, ELTE, MM1C1AB6E
- Tim Kraska, Brown University, CS195
- Dan Potter, Carsten Binnig, Eli Upfal, Brown University, CS1951A
- Erik Sudderth, Brown University, CS142
- Joe Blitzstein, Hanspeter Pfister, Verena Kaynig-Fittkau, Harvard University, CS109
- Rajan Patel, Stanford University, STAT202
- Andrew Ng, John Duchi, Stanford University, CS229