Machine Learning

Rudolf Mayer

(mayer@ifs.tuwien.ac.at)

November 20th, 2019

Outline

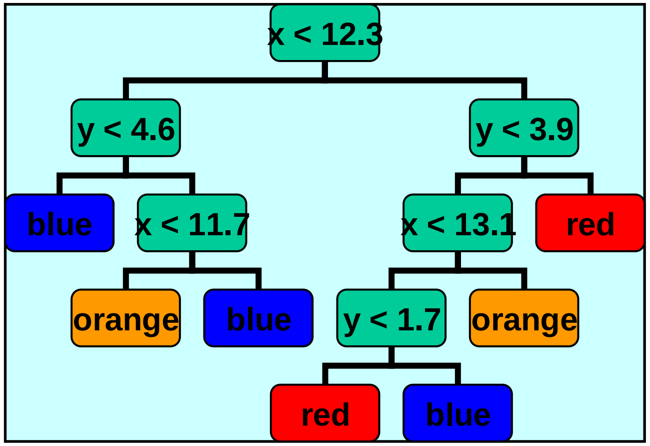
- Short recap
- Decision Trees (continued)
- Evaluation (continued)

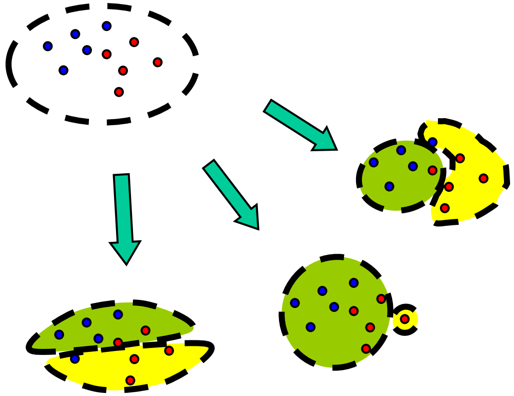
Outline

- Short recap
- Decision Trees (continued)
- Evaluation (continued)

Short Recap – Lecture October 30th

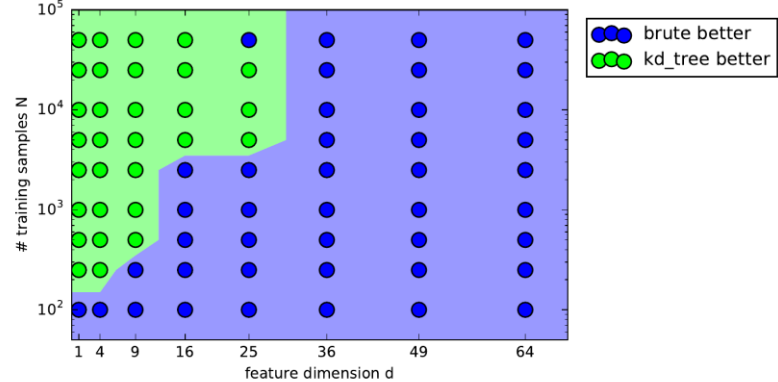
- Decision Tree Learning
  - Finding optimal split
    - Numerical attributes
  - Different criteria for optimality
    - Error Rate
    - Information Gain
    - (Gini Index)
  - Binary & multiple classes
  - Overfitting & (pre)pruning
  - Stability
  - Binary / n-ary trees
  - Categorical & numerical data





Short Recap

- K-nn continued
  - Weighting in neighbour search
  - Search optimisations



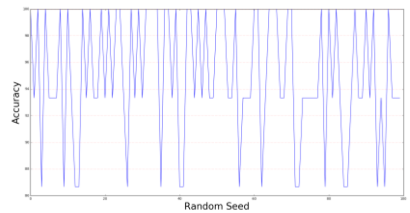
- Evaluation
  - Confusion matrix
  - Micro vs. Macro averaging

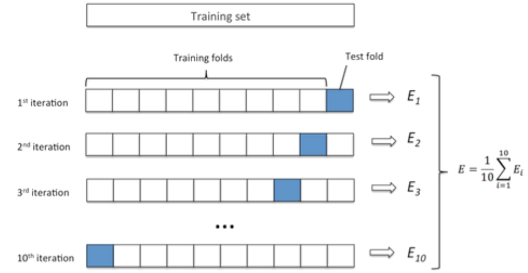
	true	false
true	True positive (TP)	False positive (FP, Type I error)
false	False negative (FN, Type II error)	True negative (TN)

classified as a	b	c	d	e	f	g	h	i	j	k	genre
34	3	0	0	2	8	0	0	2	10	1	a = Country
9	39	0	1	1	4	0	0	0	5	1	b = Folk
0	2	47	0	1	4	1	0	1	4	0	c = Grunge
0	2	0	39	0	3	1	6	8	0	1	d = Hip-Hop
2	3	3	0	34	4	10	0	0	4	0	e = Metal
10	3	9	4	4	11	3	2	1	11	2	f = Pop
5	2	5	0	10	2	36	0	0	0	0	g = Punk Rock
2	0	0	10	0	3	0	40	2	1	2	h = R&B
0	1	0	7	0	1	0	2	45	0	4	i = Reggae
8	1	8	1	3	5	1	1	1	27	4	j = Slow Rock
1	0	0	0	0	1	0	1	3	2	52	k = Children's
47	69	65	63	62	23	69	76	71	42	77	Precision
57	65	78	65	57	18	6	67	75	45	87	Recall

$$\frac{1}{|C|}\sum_{i=1}^{|C|}\frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i}$$

- Holdout vs. cross validation





Outline

- Short recap
- Decision Trees (continued)
- Evaluation (continued)

Decision Trees: Algorithm

- Popular measures to compute best split
  - Error rate
  - Information gain
  - *Gini impurity (Gini index)*

Information Theory & Entropy

- Introduced by Claude Shannon (1948)
  - Original for compression & reliable communication
  - Applications in statistical inference, NLP, cryptography,…
- Entropy: # of bits needed for communication
  - Absolute limit for best lossless compression
- Measure of uncertainty
  - High probability – low entropy
- Concerned with measuring actual information vs. redundancy

What is „Information Entropy“?

- Entropy – measure of uncertainty

- ML: measure for the “impurity” of a set
  - High Entropy ➔ bad for prediction
  - High Entropy ➔ needs to be reduced

$$H(X) = \mathrm{E}(\mathrm{I}(X)) = \sum_{i=1}^{n} p(x_i) I(x_i) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

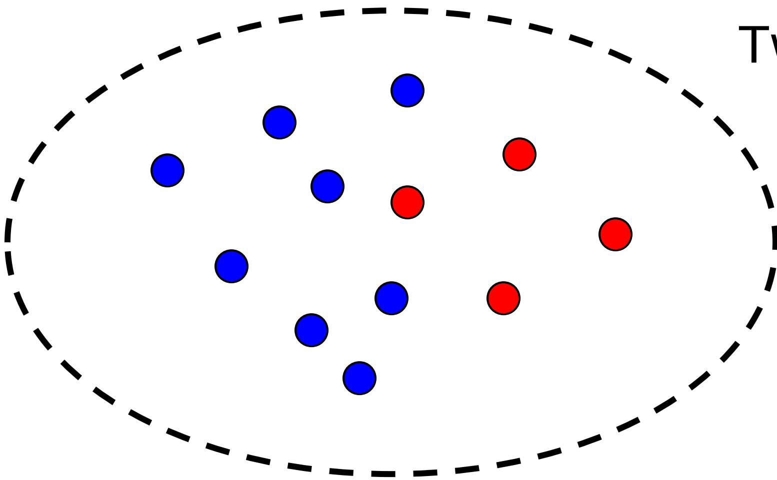
H … Entropy

E … Expected value

I(X) …information content of X

p(...) … probability function

Calculating *H(X)*: example



Two dimensional data, two classes

$$p(x_{\text{red}}) = \frac{4}{12} = 0.33$$

$$p(x_{\text{blue}}) = \frac{8}{12} = 0.67$$

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$$

$$H(X) = -p(x_{red})\log_2 p(x_{red}) - p(x_{blue})\log_2 p(x_{blue})$$

Remember:

$$\log_2(x) = \log(x)/\log(2)$$

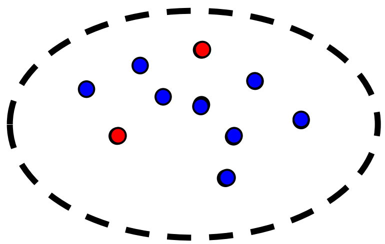
$$H(X) = -\frac{1}{3}\times\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\times\log_2\left(\frac{2}{3}\right)$$

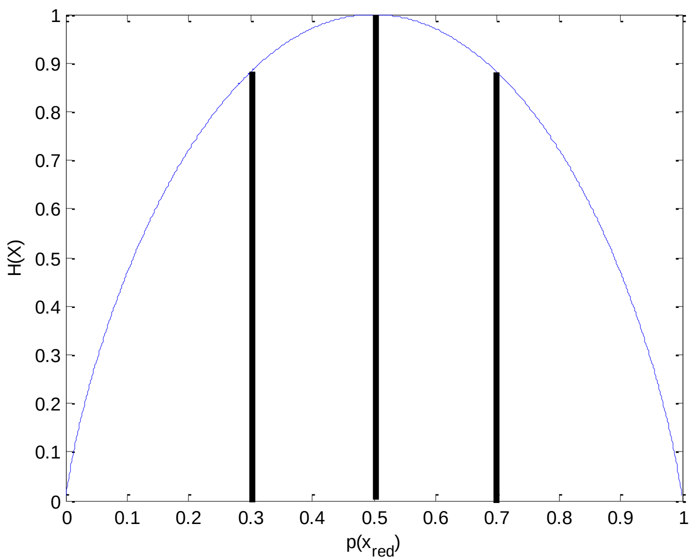
$$= -\frac{1}{3}\times -1.58 - \frac{2}{3}\times -0.58$$

$$= 0.53 + 0.39$$

$$= 0.92$$

$H(X)$: Example values



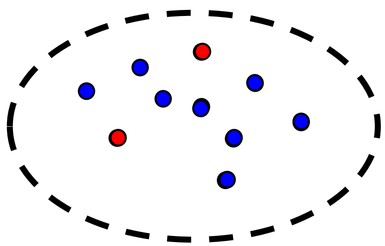


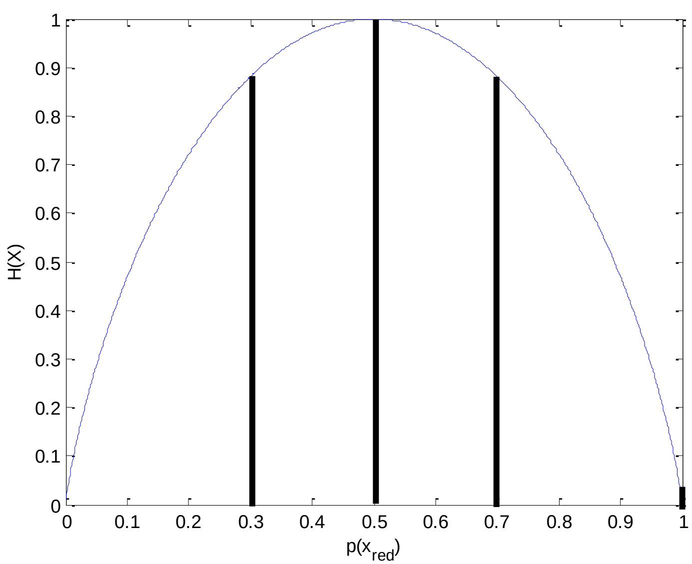
	$p(x_{red})$	$p(x_{blue})$	$H(X)$
I	0.5	0.5	?
II	0.3	0.7	?
III	0.7	0.3	?
IV	0	1	?

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

Any guesses?

$H(X)$: Example values



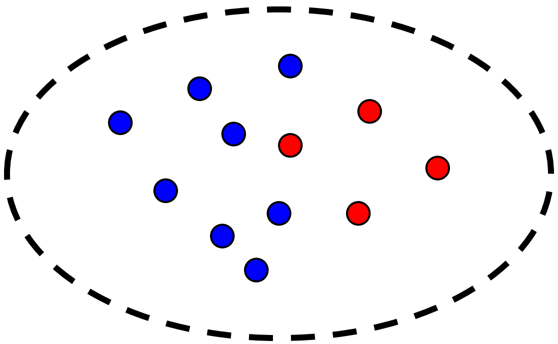


	$p(x_{red})$	$p(x_{blue})$	$H(X)$
I	0.5	0.5	?
II	0.3	0.7	?
III	0.7	0.3	?
IV	0	1	?

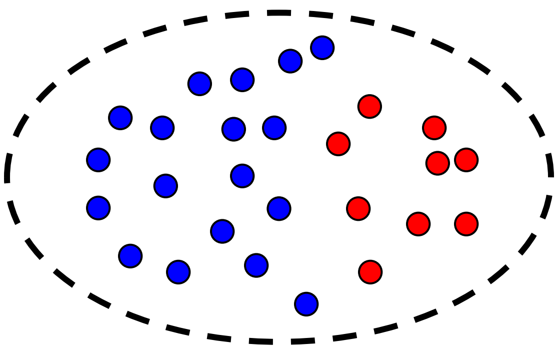
$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

$H(X)$: Relative vs. absolute frequencies

What are the entropies of I and II?



vs.



$$p(x_{\text{red, I}}) = \frac{4}{12} = 0.33\,;\; p(x_{\text{blue, I}}) = \frac{8}{12} = 0.67$$

$$p(x_{\text{red, II}}) = \frac{9}{27} = 0.33\,;\; p(x_{\text{blue, II}}) = \frac{18}{27} = 0.67$$

$\Rightarrow H(X_I) = H(X_{II})$

Dataset	red	blue
I	8	4
II	18	9

Only **relative** frequencies matter!

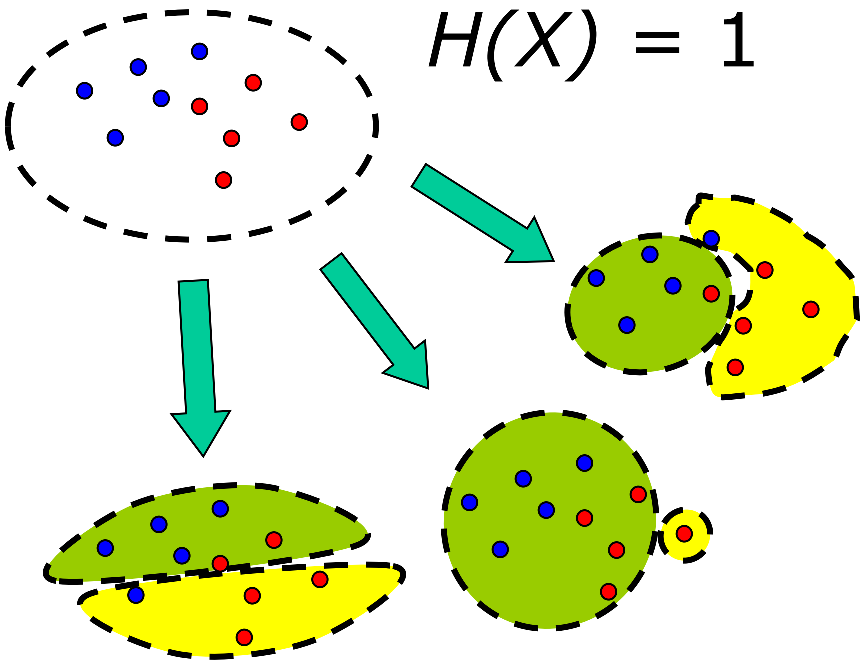
Information Gain

Given a set and a choice between possible subsets, which one is preferable?

Information Gain: Sets that minimize Entropy by largest amount

$$IG(X_A, X_B) = H(X) - p(x_A)H(X_A) - p(x_B)H(X_B)$$

$H(X) = 1$



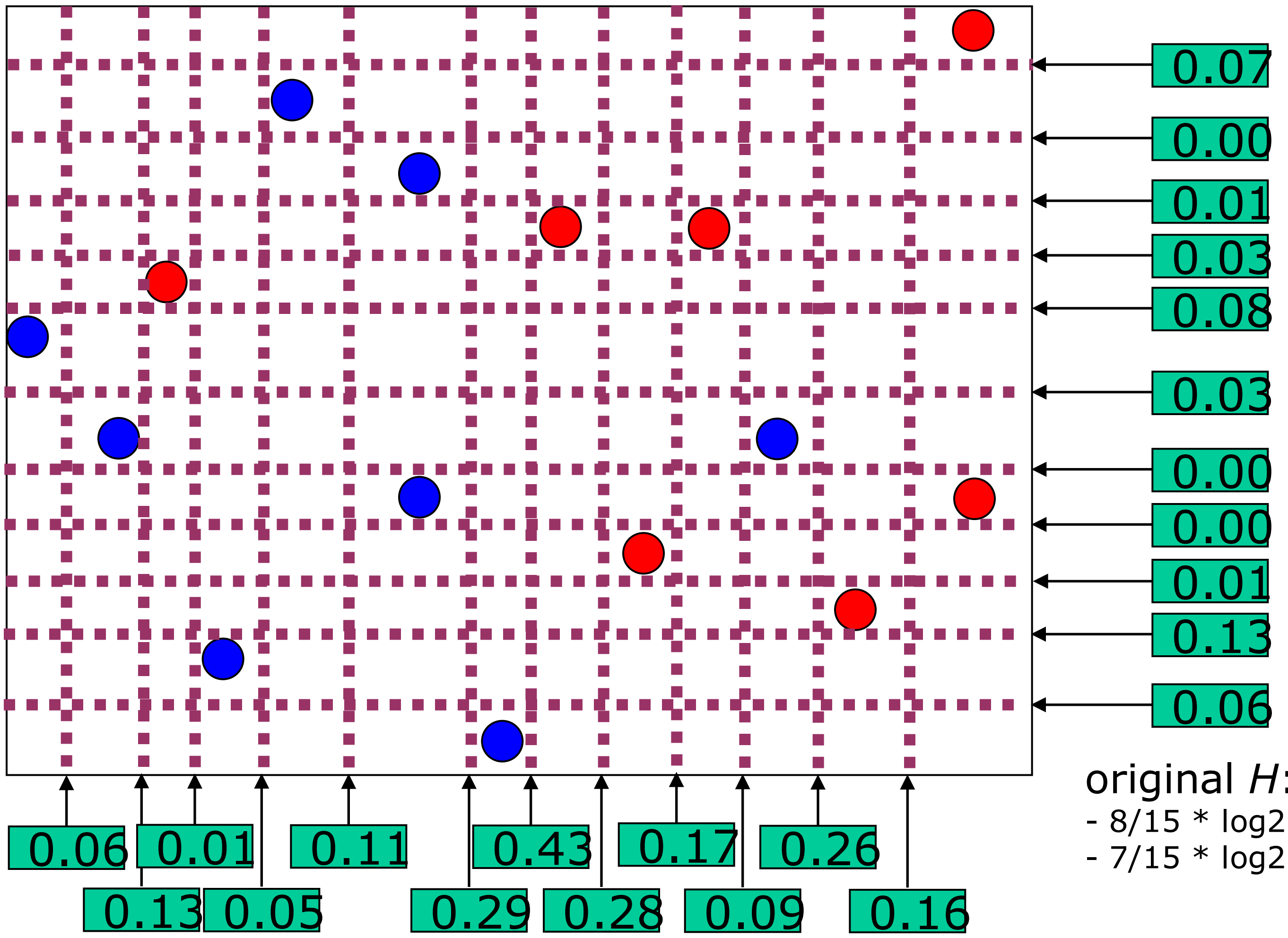
	A (green)	B (yellow)
Points	[illegible]	[illegible]
$p(X.)$	0.[illegible]	0.[illegible]
$p(x_{red})$	0.[illegible]	[illegible]
$p(x_{blue})$	0.[illegible]	0.25
$H(X.)$	0.[illegible]	0.[illegible]
IG	0.[illegible] [illegible]	

Information Gain (Properties)

- Information Gain is
  - the amount by which the original Entropy can be reduced by splitting into subsets
  - *Min/max bounds of Information gain?*
  - at most as large as the Entropy of the undivided set
  - at least zero (if Entropy is not reduced)
- $0 \leq IG \leq H(X)$

Decision Trees: information gain

- 2-dimensional data (x, y), numerical values, two classes



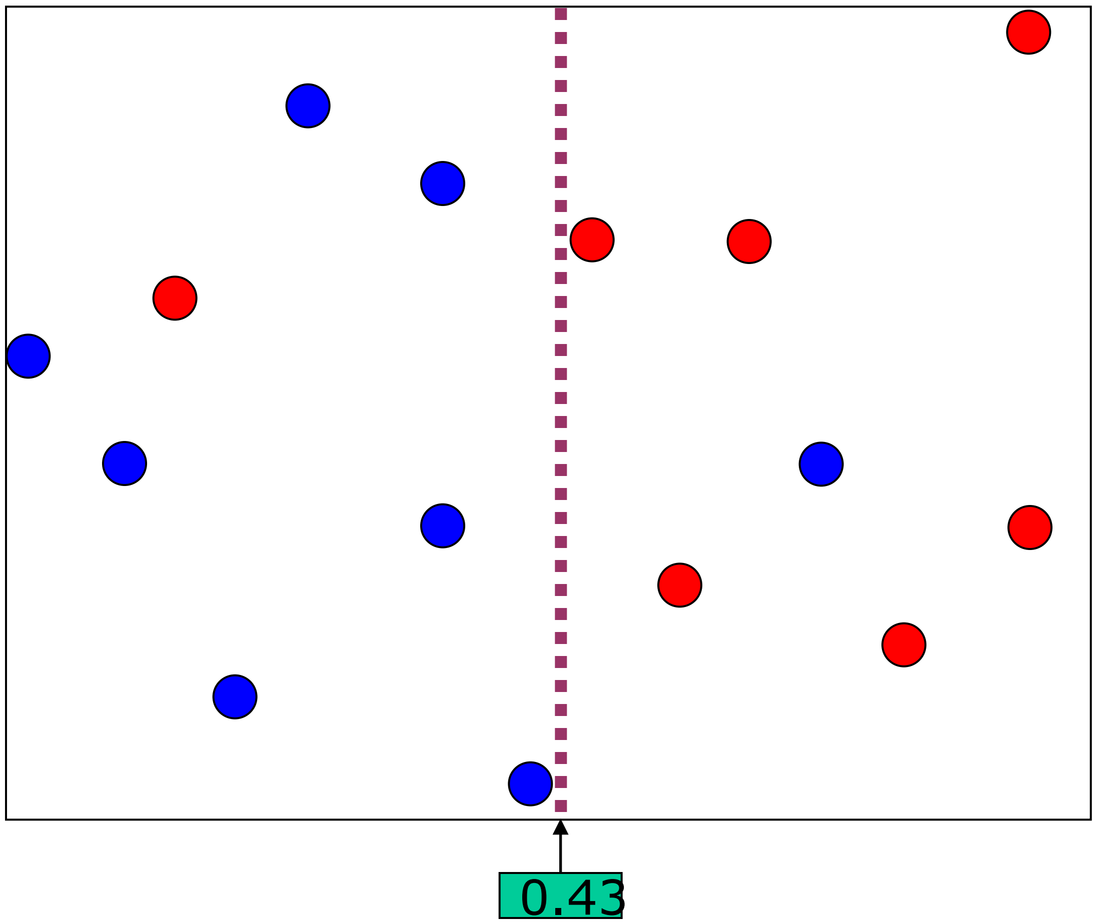
original H: 0.99679

- 8/15 * log2(8/15)
- 7/15 * log2(7/15)

Decision Trees: information gain

H(left) =
$-0.125\log_2 0.125 - 0.875\log_2 0.875 =$
$0.375 + 0.169 = 0.54356$

H(right) =
$-0.143\log_2 0.143 - 0.857\log_2 0.857 =$
$0.401+0.191 = 0.59167$



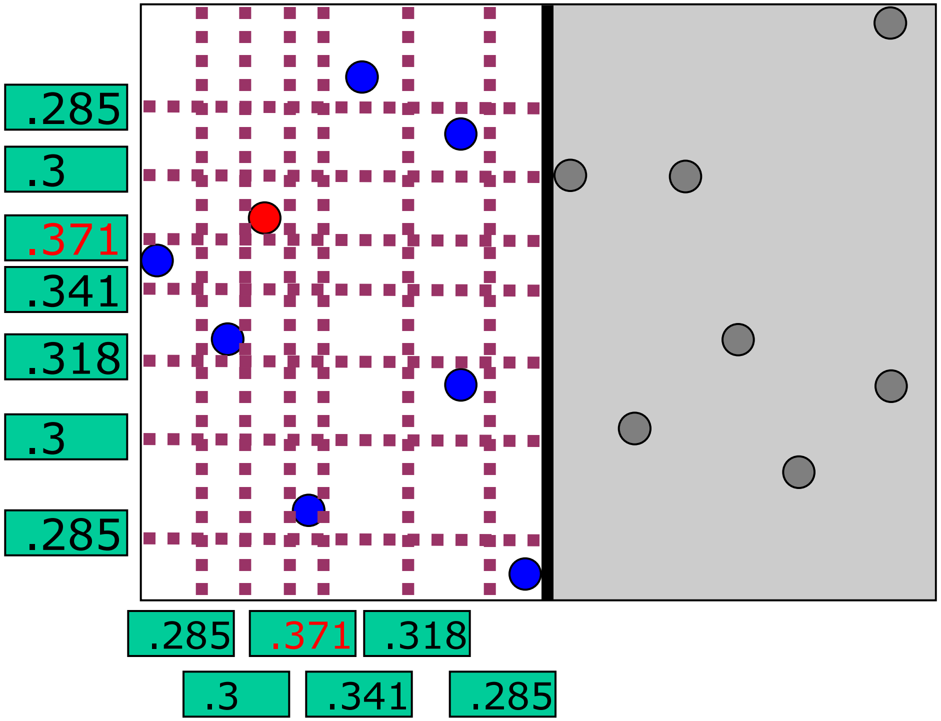
H(split) =
0.54356*8/15 +
0.59167*7/15 =
0.566011333

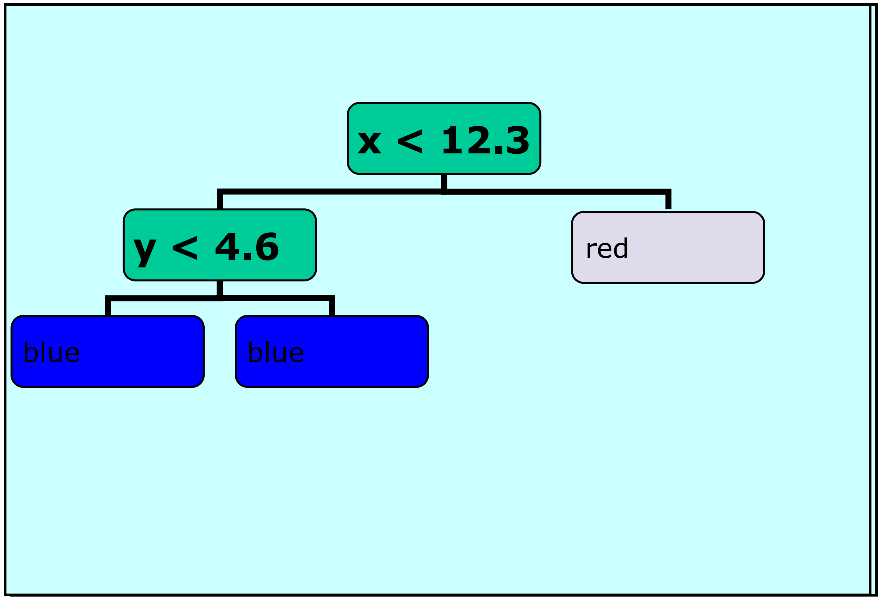
original *Entropy*:
H(x) = 0.99679

- 8/15 * log2(8/15)
- 7/15 * log2(7/15)

IG = 0.43078

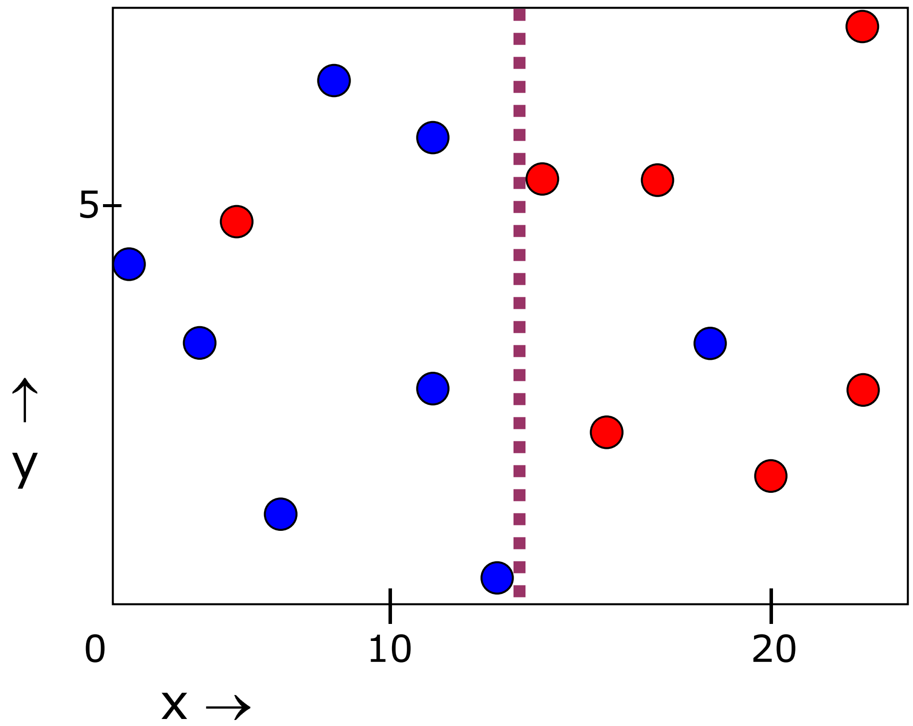
Decision Trees: information gain

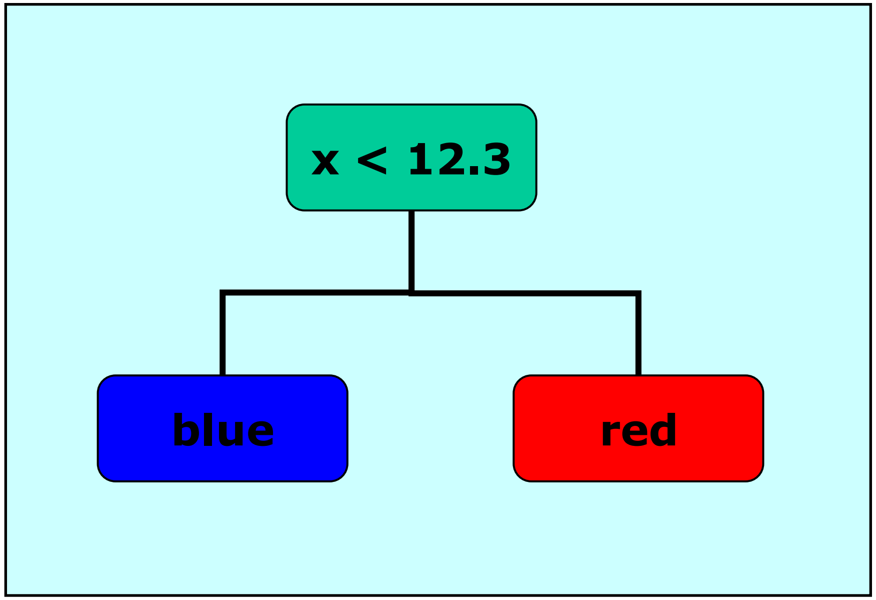




Decision Trees: information gain

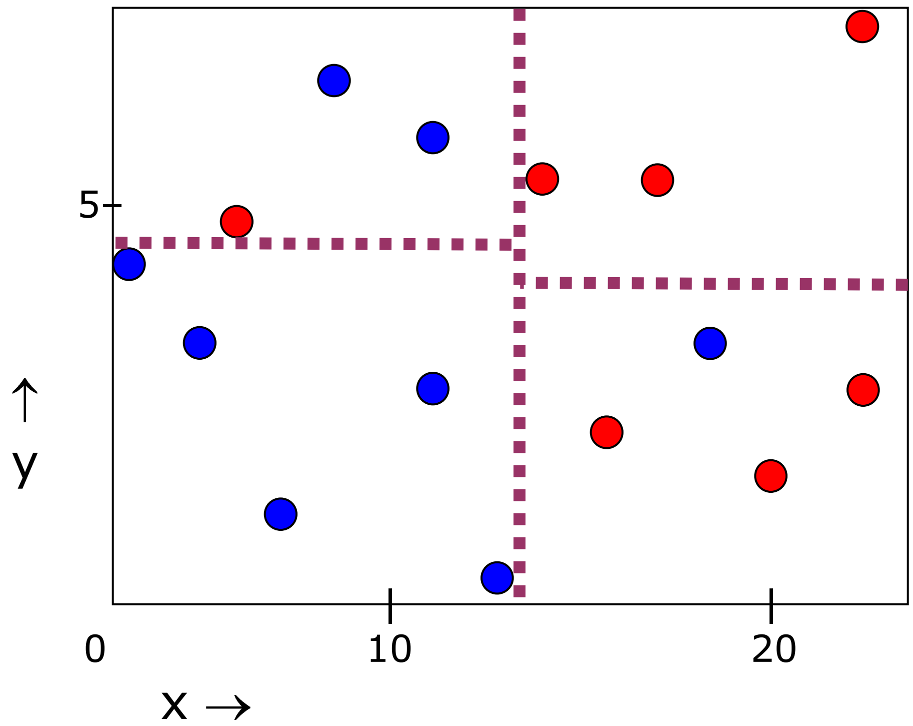
- Tree training, level 1

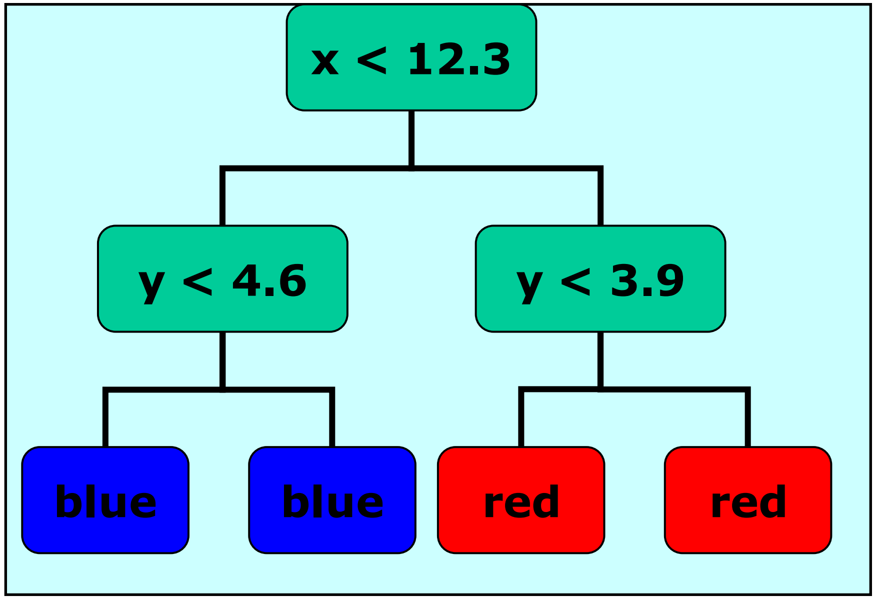




Decision Trees: information gain

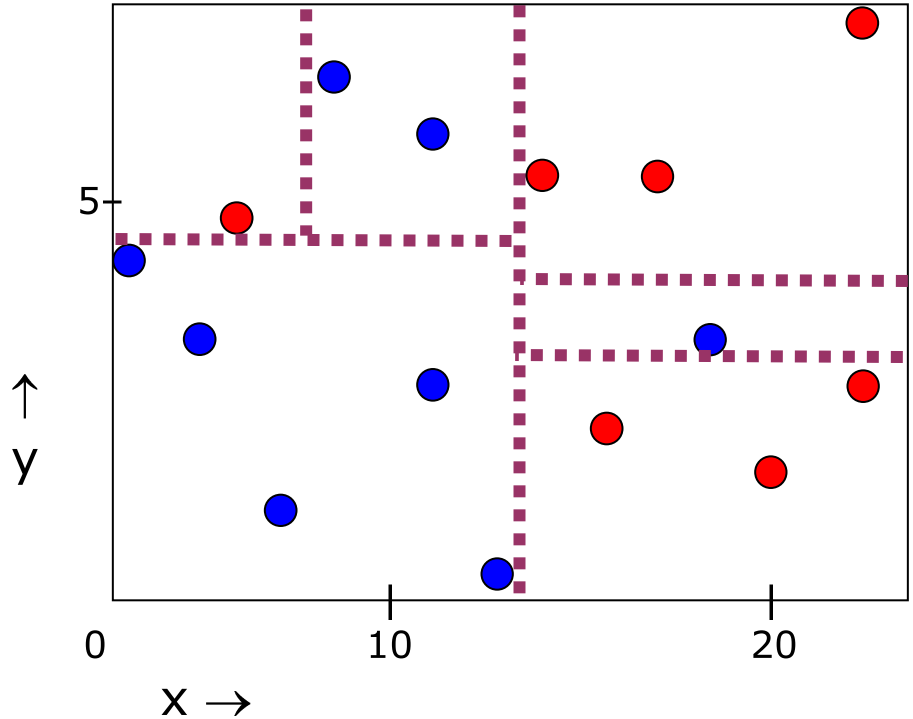
- Tree training, level 2

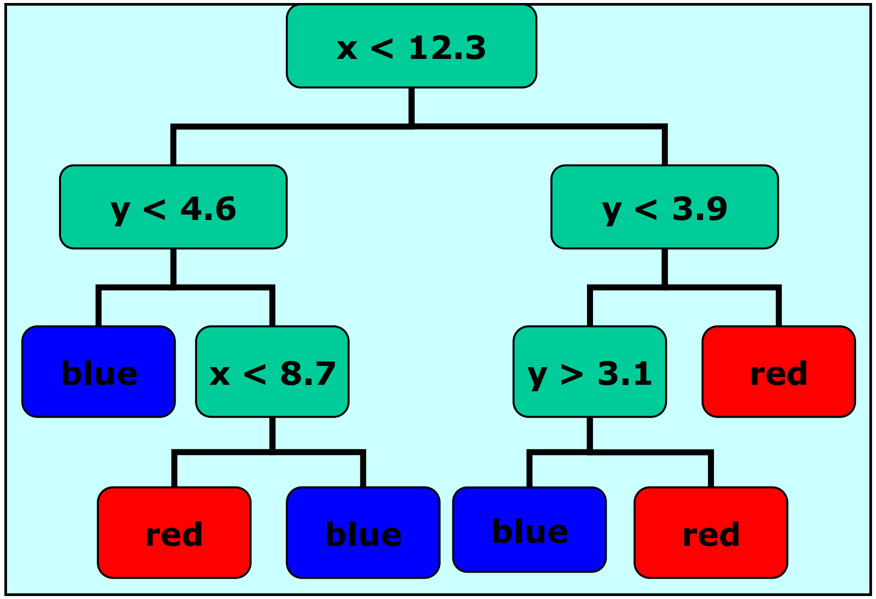




Decision Trees: information gain

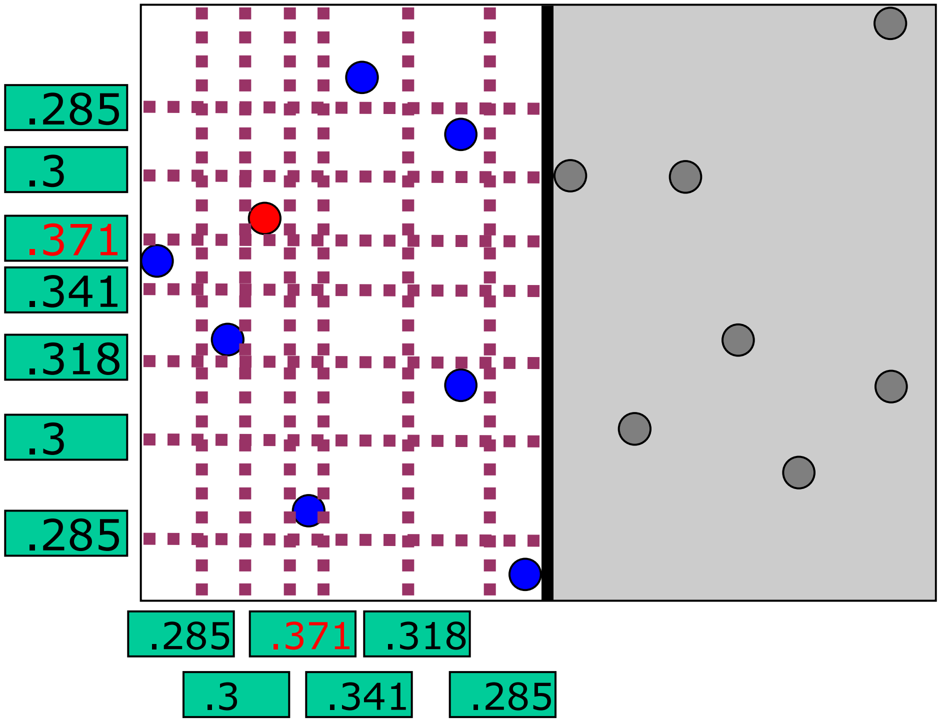
- Tree training, level 3: completely built tree

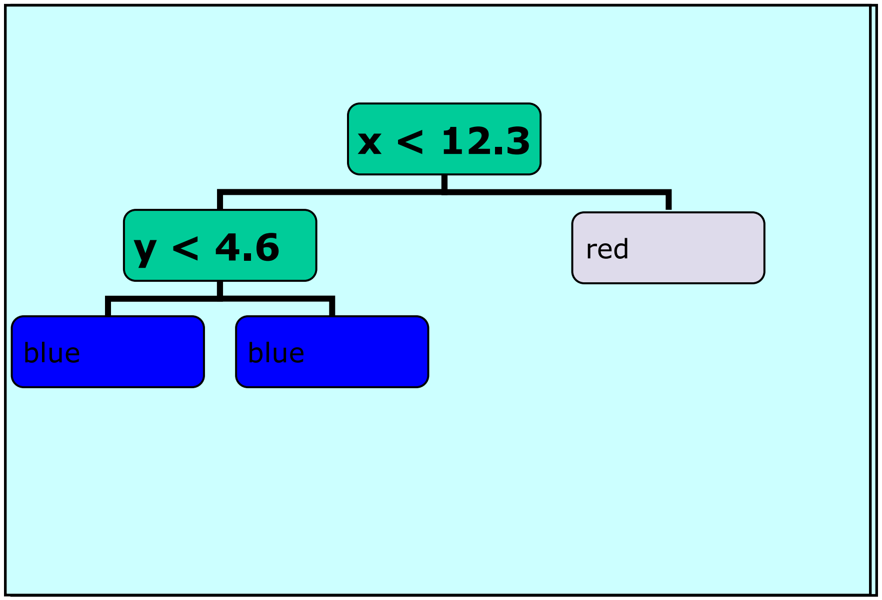




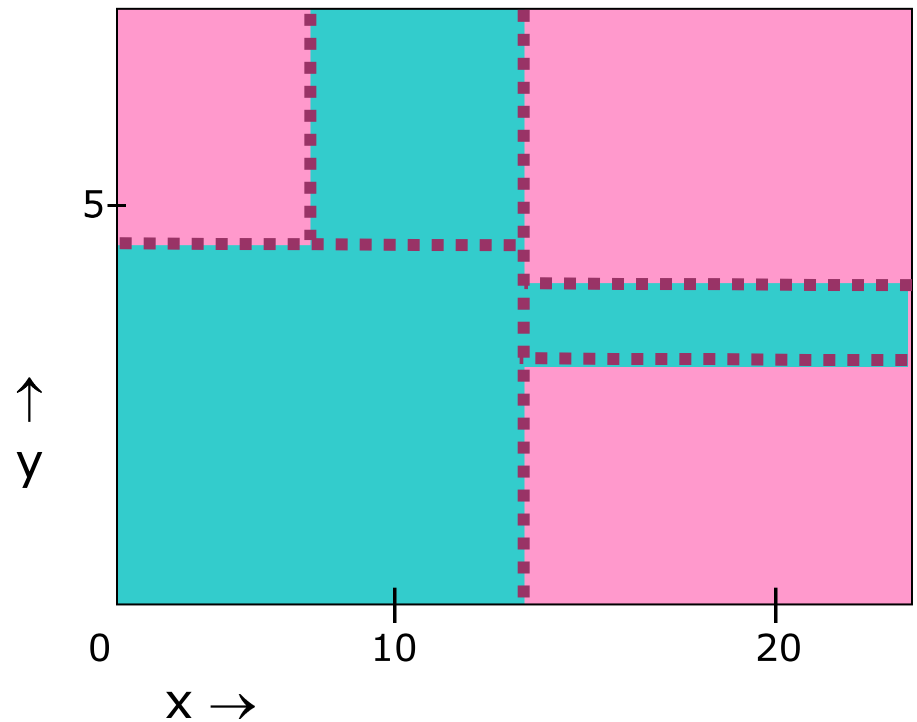
Decision Trees: information gain

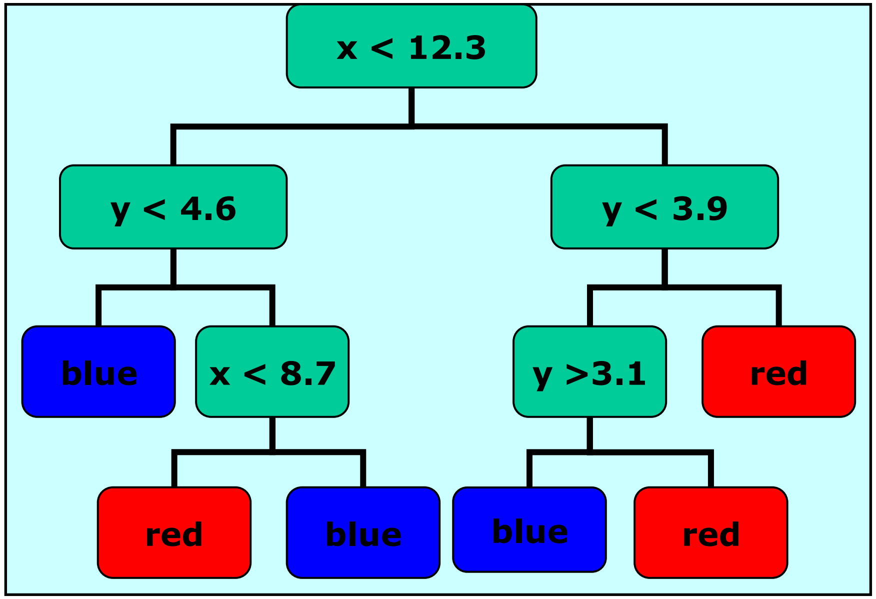
- Information gain: considers sizes of subsets
  - Preferred over error rate





Decision Trees: Classification

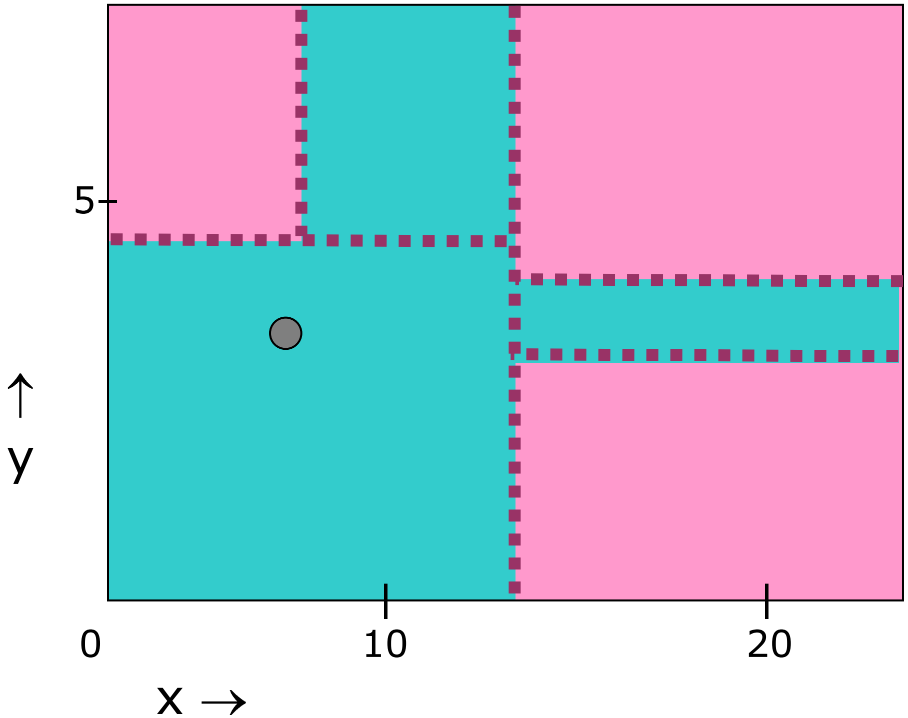


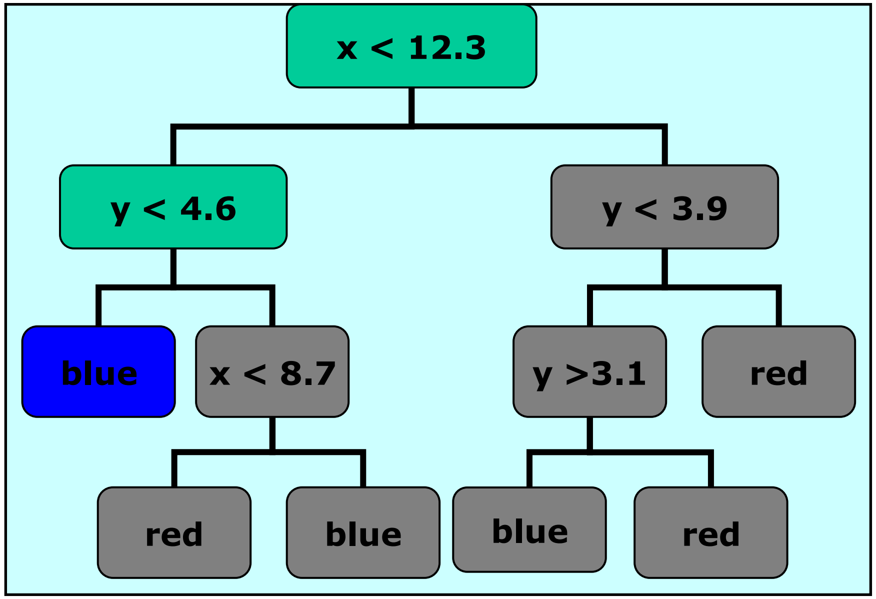


How to classify unknown items?

Same as for a tree built error rate!

Decision Trees: Classification

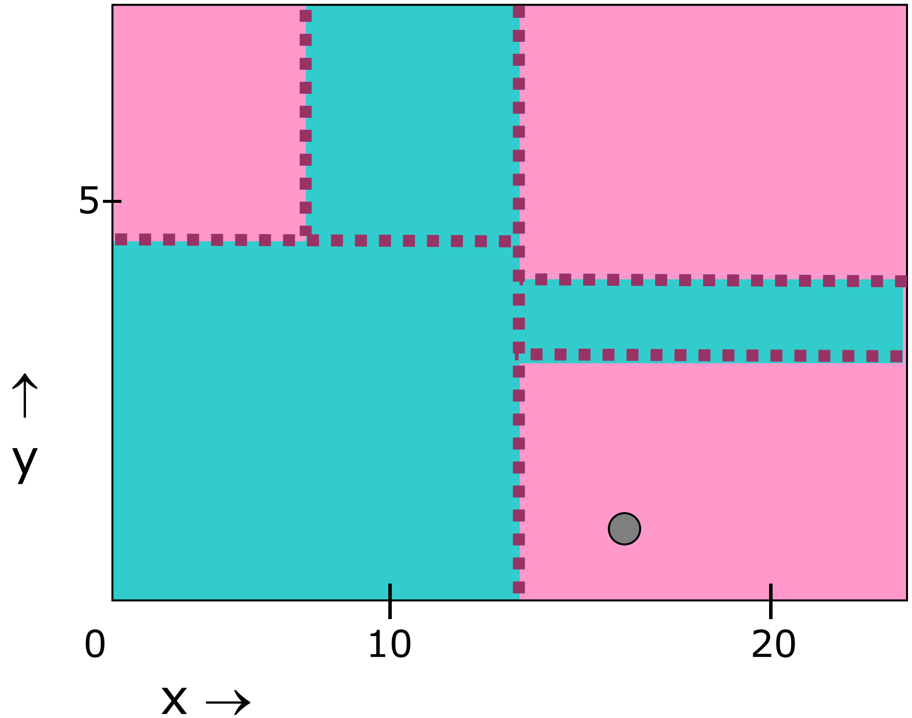


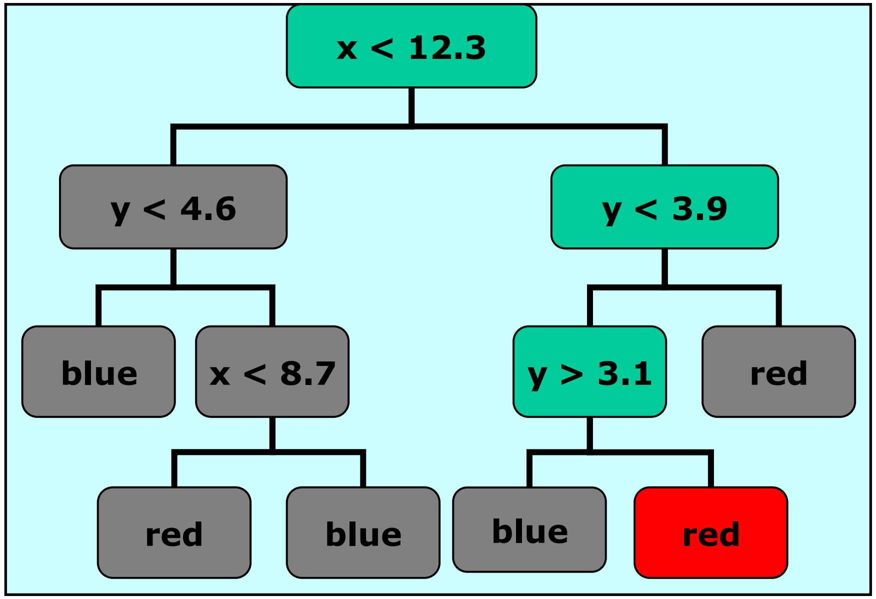


➔ Decend the tree until leaf-node

➔ Use majority of class in that leaf node

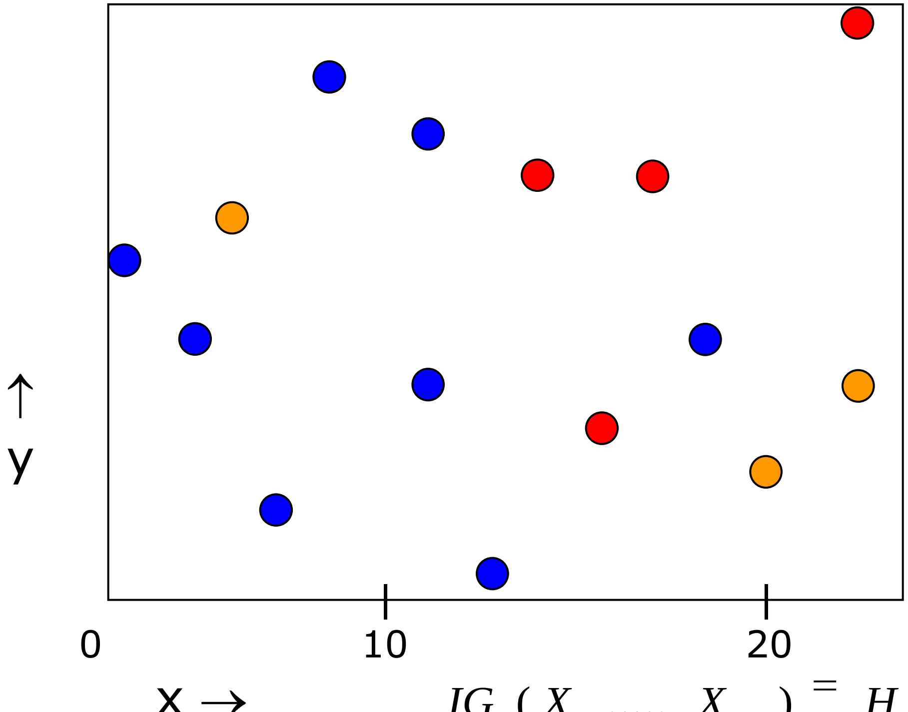
Decision Trees: Classification





- ➔ Decend the tree until leaf-node
- ➔ Use majority of class in that leaf node

Decision Trees: More than 2 classes



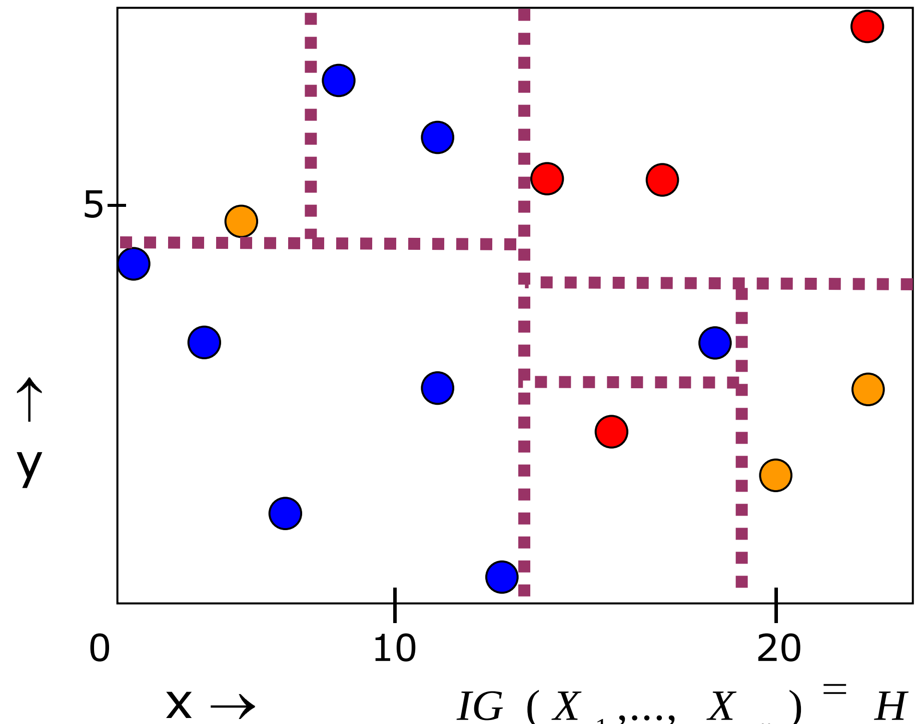
- *Conceptual changes?*

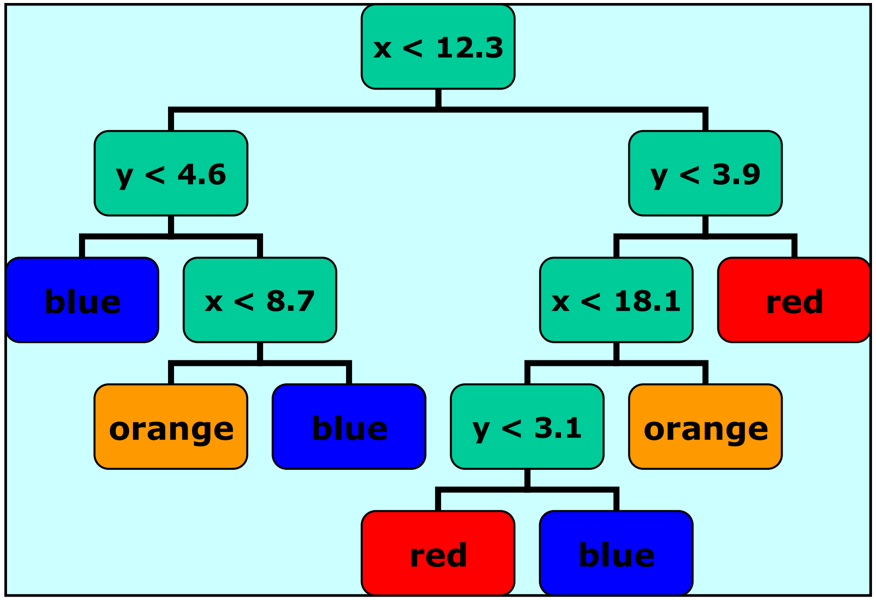
$$IG(X_1,\ldots,X_m) = H(X) - \sum_{j=1}^{m} p(x_j) H(X_j)$$

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

$$H(X) = -p(x_{red}) \log_2 p(x_{red}) - p(x_{blue}) \log_2 p(x_{blue}) - p(x_{yellow}) \log_2 p(x_{yellow})$$

Decision Trees: More than 2 classes



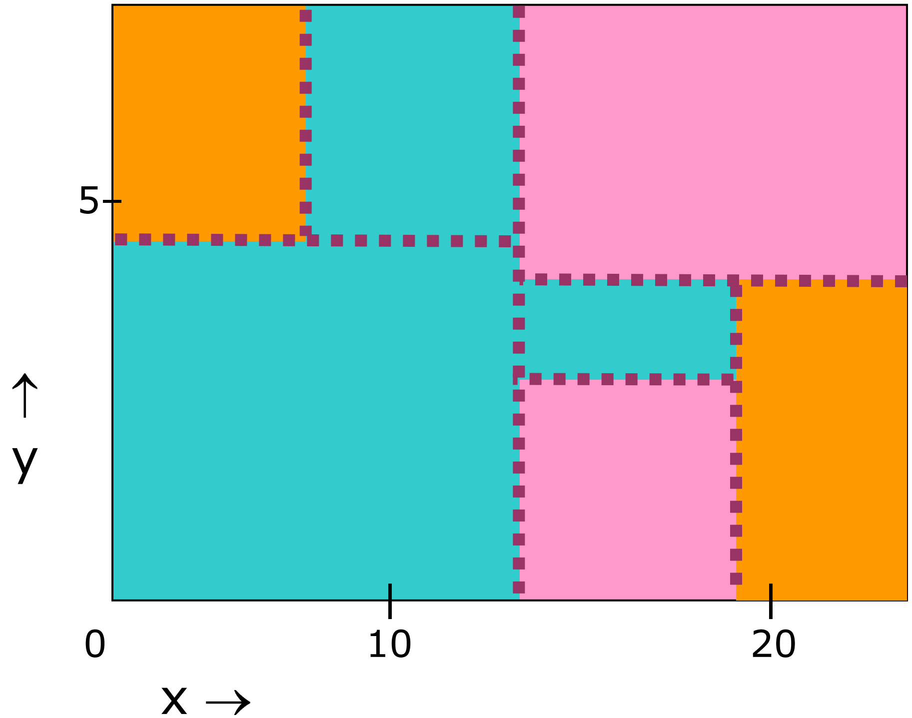


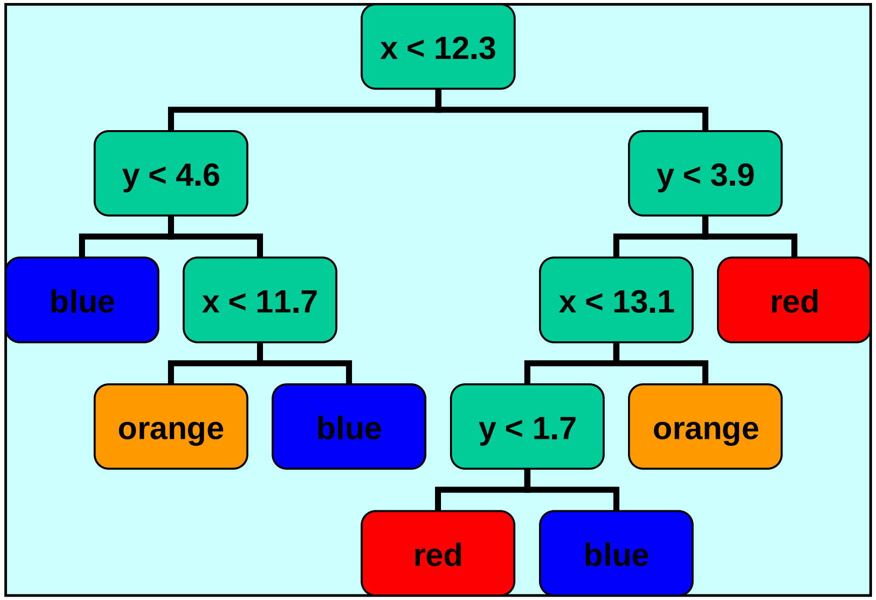
$$IG(X_1, \ldots, X_m) = H(X) - \sum_{j=1}^{m} p(x_j) H(X_j)$$

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

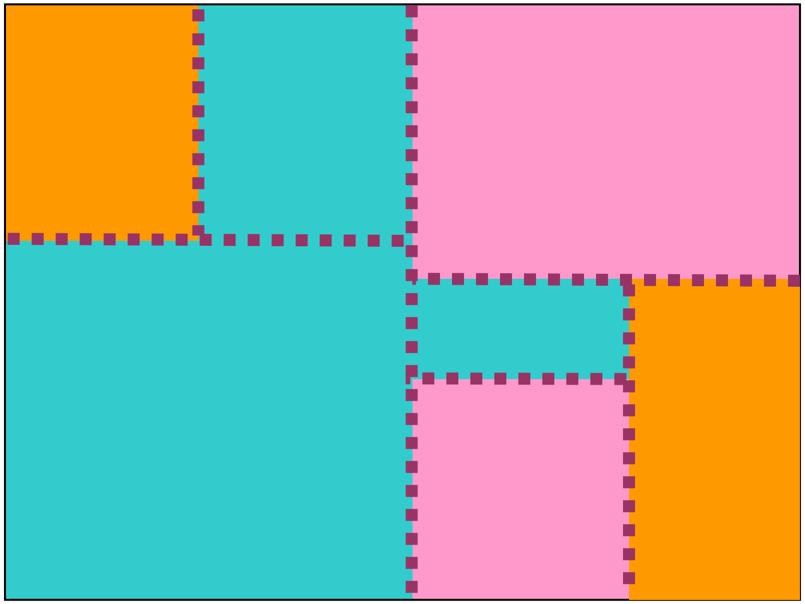
FACULTY OF !NFORMATICS

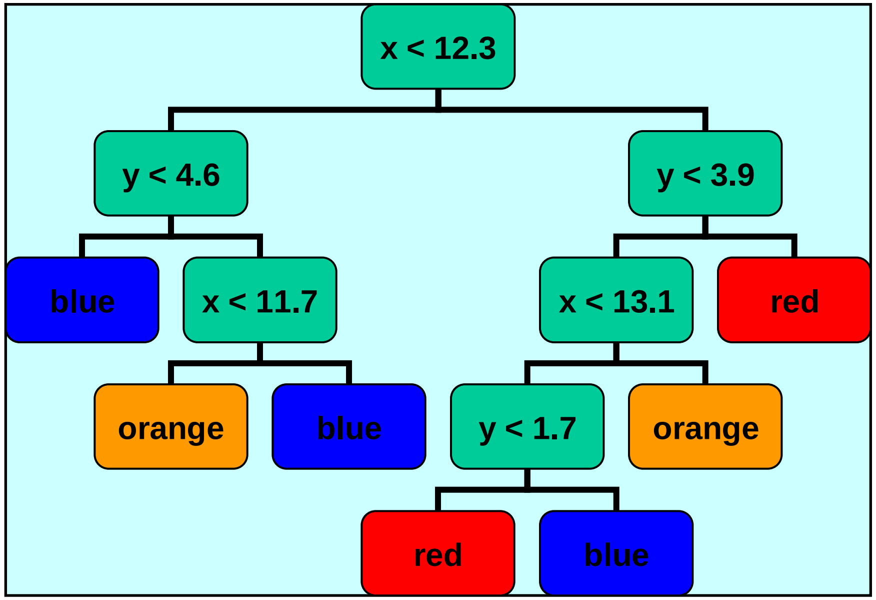
Decision Trees: More than 2 classes





Decision Trees: More than 2 classes



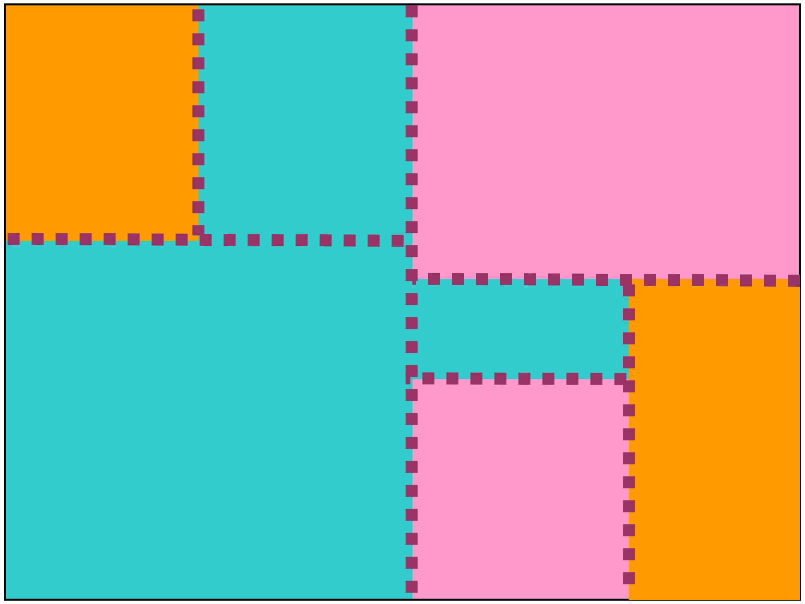


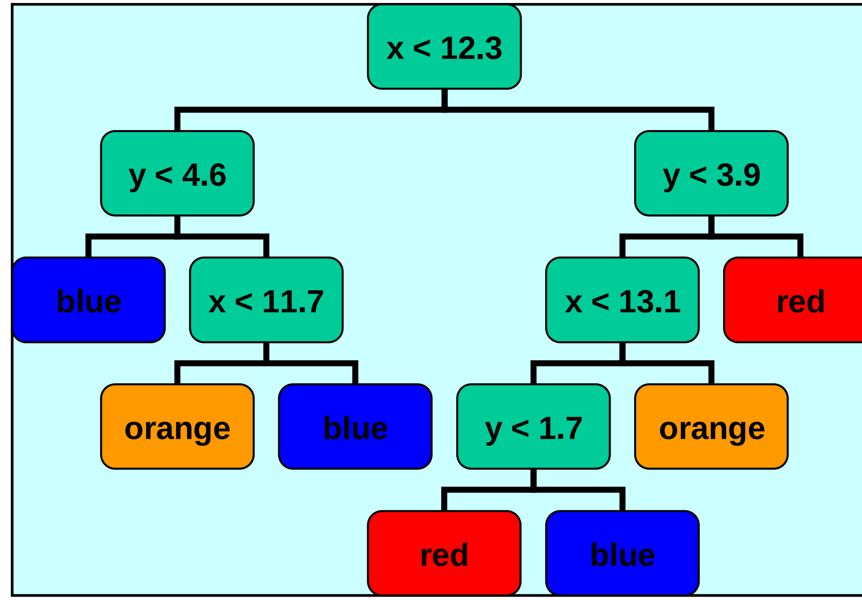
$$IG(X_1, \ldots, X_m) = H(X) - \sum_{j=1}^{m} p(x_j) H(X_j)$$

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

Maximum value of Entropy?

Decision Trees: More than 2 classes





$$IG(X_1,\ldots,X_m) = H(X) - \sum_{j=1}^{m} p(x_j) H(X_j)$$

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$$

$$H(X) = -p(x_{red})\log_2 p(x_{red}) - p(x_{blue})\log_2 p(x_{blue}) - p(x_{yellow})\log_2 p(x_{yellow})$$

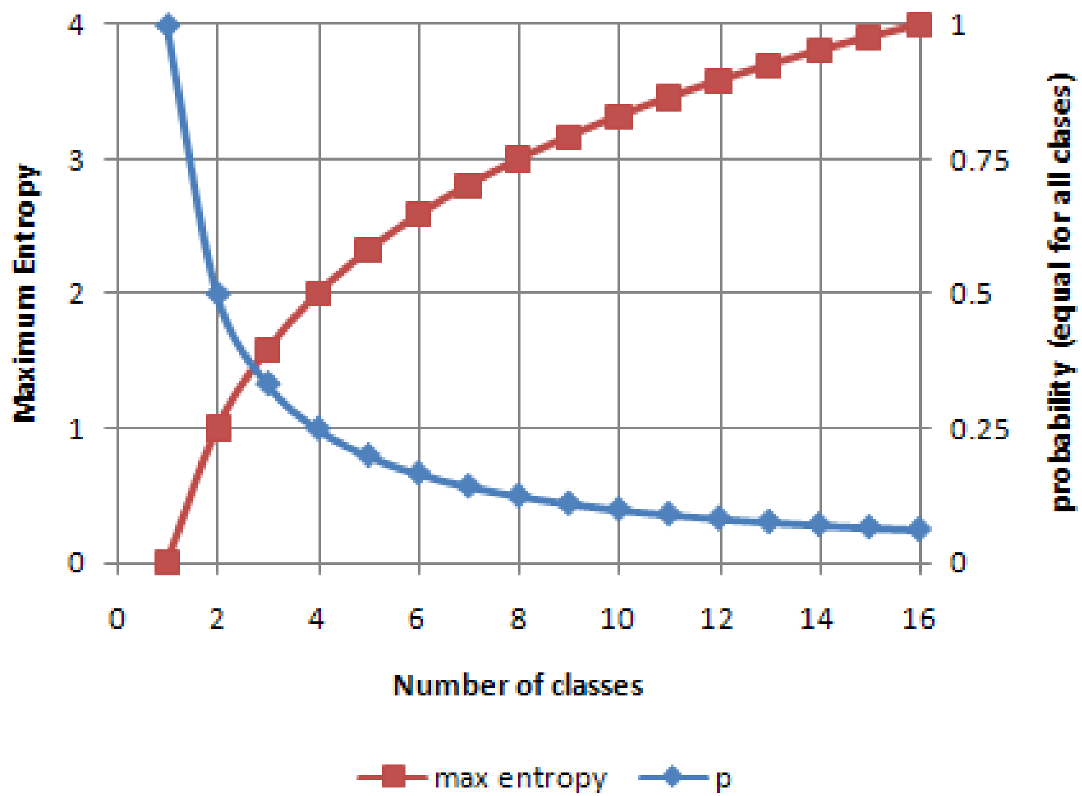
Maximum Entropy? $H(X) = \frac{1}{3} \times \log_2(\frac{1}{3}) - \frac{1}{3} \times \log_2(\frac{1}{3}) - \frac{1}{3} \times \log_2(\frac{1}{3}) = 1.5849$

Decision Trees: More than 2 classes

$$IG(X_1,\ldots,X_m) = H(X) - \sum_{j=1}^{m} p(x_j) H(X_j)$$

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

Maximum Entropy?



Decision Trees: Algorithm

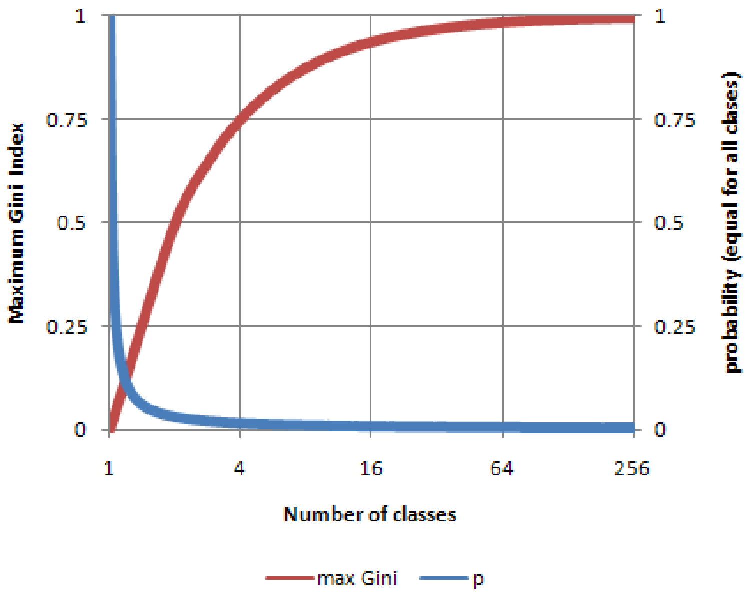
- Popular measures to compute best split
  - Error rate
  - Information gain
  - *Gini impurity (Gini index)*

Gini impurity (Gini index)

- Inequality among values of a distribution
  - Developed initial for income levels
  - How often a randomly chosen element from the set would be incorrectly labeled, if it was randomly labeled according to the distribution of labels in the subset

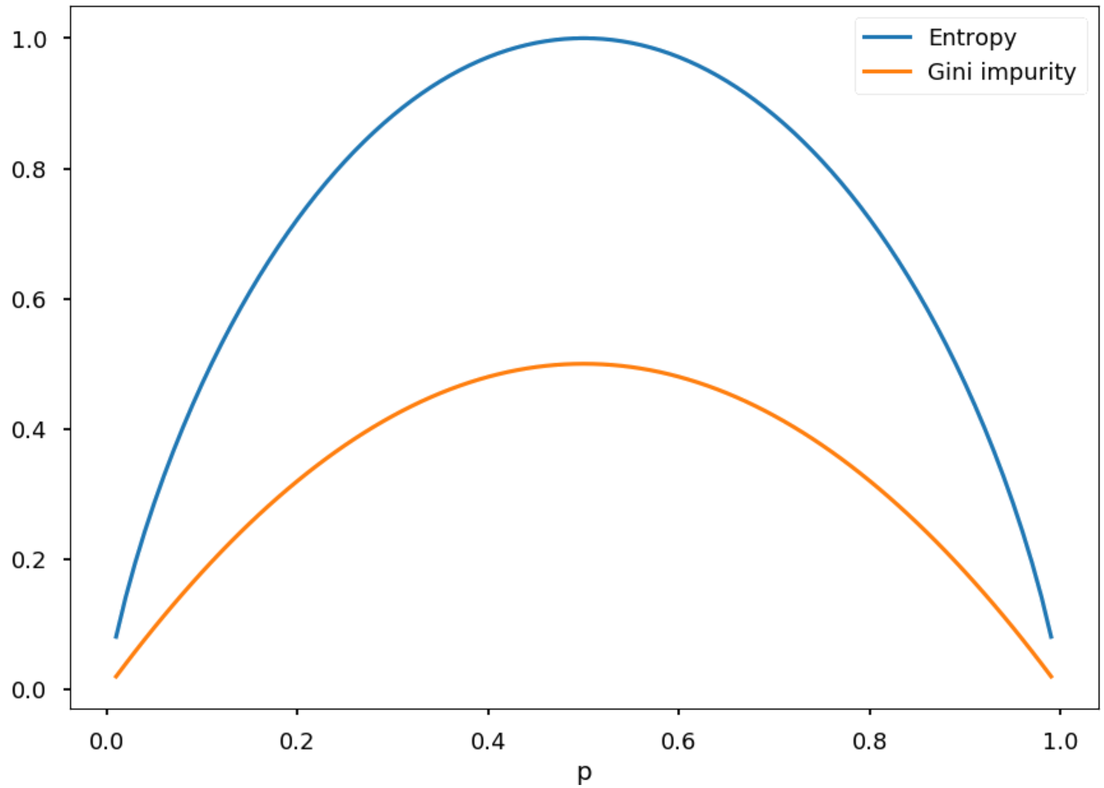
$$I_G(p) = \sum_{i=1}^{|C|} p_i(1-p_i) = \sum_{i=1}^{|C|} (p_i - p_i^2) = \sum_{i=1}^{|C|} p_i - \sum_{i=1}^{|C|} p_i^2 = 1 - \sum_{i=1}^{|C|} p_i^2$$

- Value range?
  - Between 0 (only one class) and 1 (total inequality)



Entropy vs. Gini Impurity

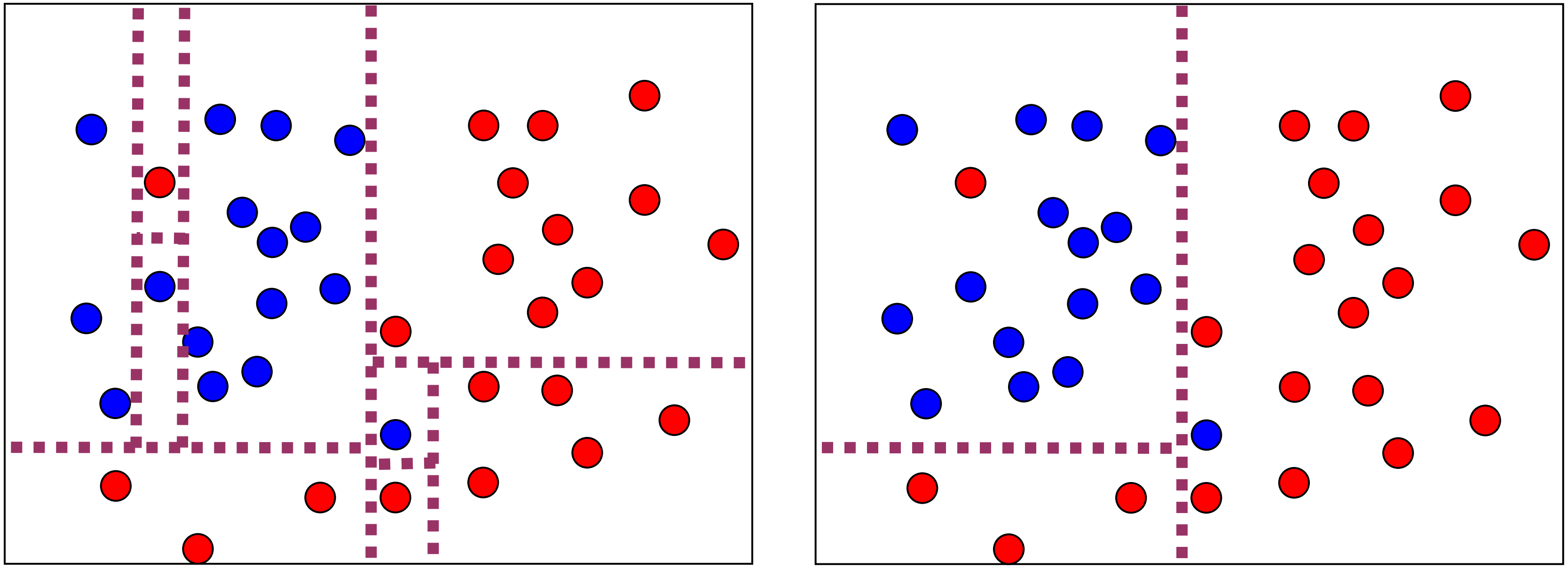
- Slightly different properties
  - Empirically studies have shown no clear evidence
  - Disagree only on 2% of potential cases [1]



[1] Laura Elena Raileanu & Kilian Stoffel. Theoretical comparison between the Gini Index and Information Gain criteria

Decision Trees: Overfitting

- Fully grown trees are usually too complicated
  - *Why is that an issue?*
    - Generalisation
    - Understanding
  - Especially useful when there is noisy/"useless" data



Fully grown

Simplified

Decision Trees: Stopping Criteria

- *How to achieve simplified trees?*
  - Avoid fully growing trees! *How?*
    ➔ Alternative stopping criteria
  - Stop splitting a node when
    - ~~*Data in each node from only one class*~~
    - Absolute number of samples is low (< threshold)
    - Entropy is already relatively low (< threshold)
    - Information Gain is low (< threshold)
    - Depth of tree has reached a max value (> threshold)
  - Threshold values depend on data set

Decision Trees: Pruning

- *How to achieve simplified trees?*
  - “Cut back” complicated trees

- "Pruning" means removing nodes from a tree after training has finished
  - Especially useful when there is noisy data
  - Stopping criteria sometimes referred to as "pre-pruning“
- Least contributing nodes are removed
  - sometimes tree is remodelled
- A set of candidate trees is generated
  - Best tree is selected
- Reduces complexity of tree

Decision Trees: Pruning

- Simple bottom-up approach: reduced error pruning

1. Starting from leaves, remove a sub-tree from the tree
   a. Replace it with the majority class
   b. Evaluate the performance without the pruned node
   c. Keep tree if effectiveness is not decrease

2. Repeat step 1
   - Until no improvement is obtained from pruning
   - (As long as ***effectiveness is still acceptable)***

Decision Trees: Pruning

- Other approaches

- Cost complexity pruning (bottom up)
  - Generate a list of candidate trees T_0, T_1, … T_m
    - T_m = root node only
    - Selecting subtrees to be replaced as the tree that minimises

$$\frac{\mathrm{err}(\mathrm{prune}(T,t),S) - \mathrm{err}(T,S)}{|\mathrm{leaves}(T)| - |\mathrm{leaves}(\mathrm{prune}(T,t))|}$$

  - Take the best performing tree as the final decision tree

- Pessimistic Error Pruning (top-down)

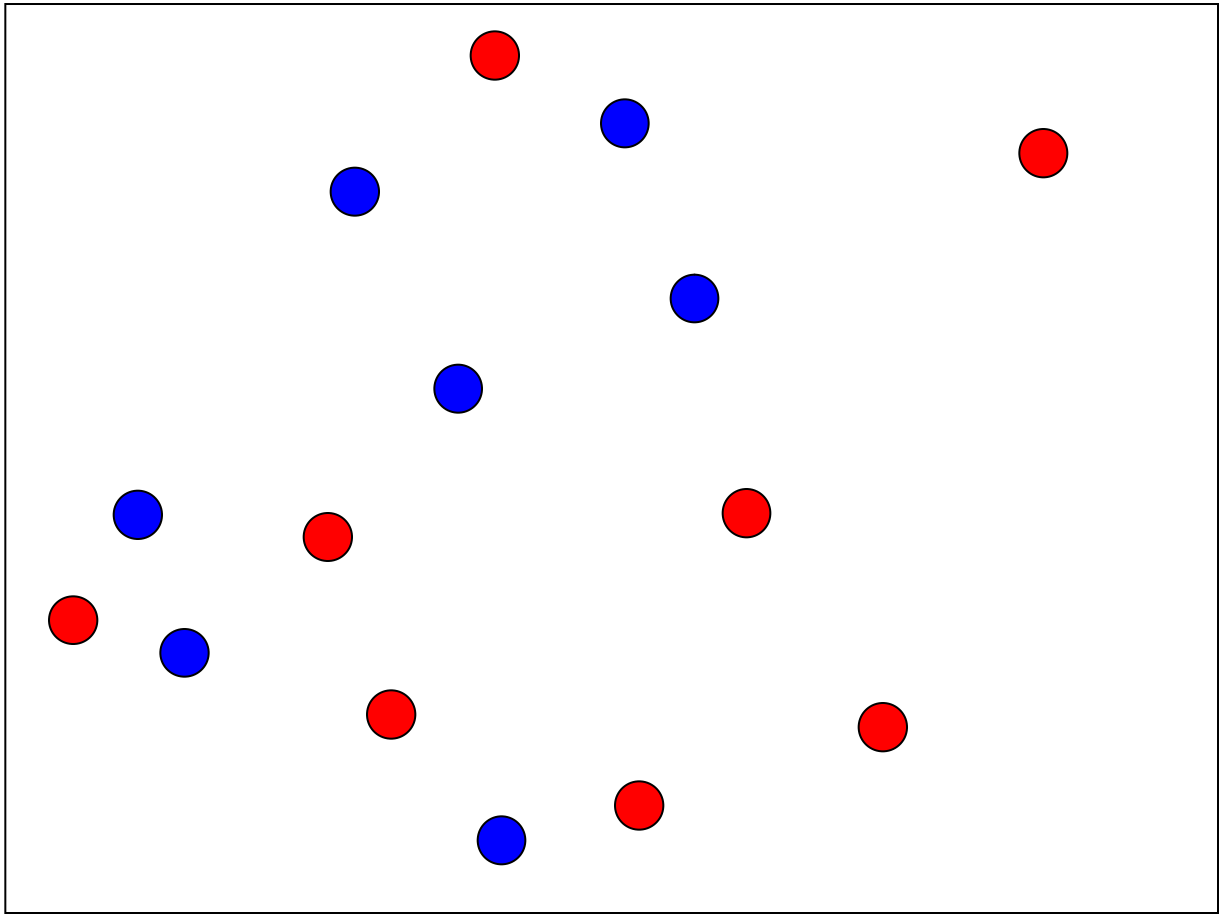
Decision Trees: Overfitting

- *What's the difference between pruning & pre-pruning?*

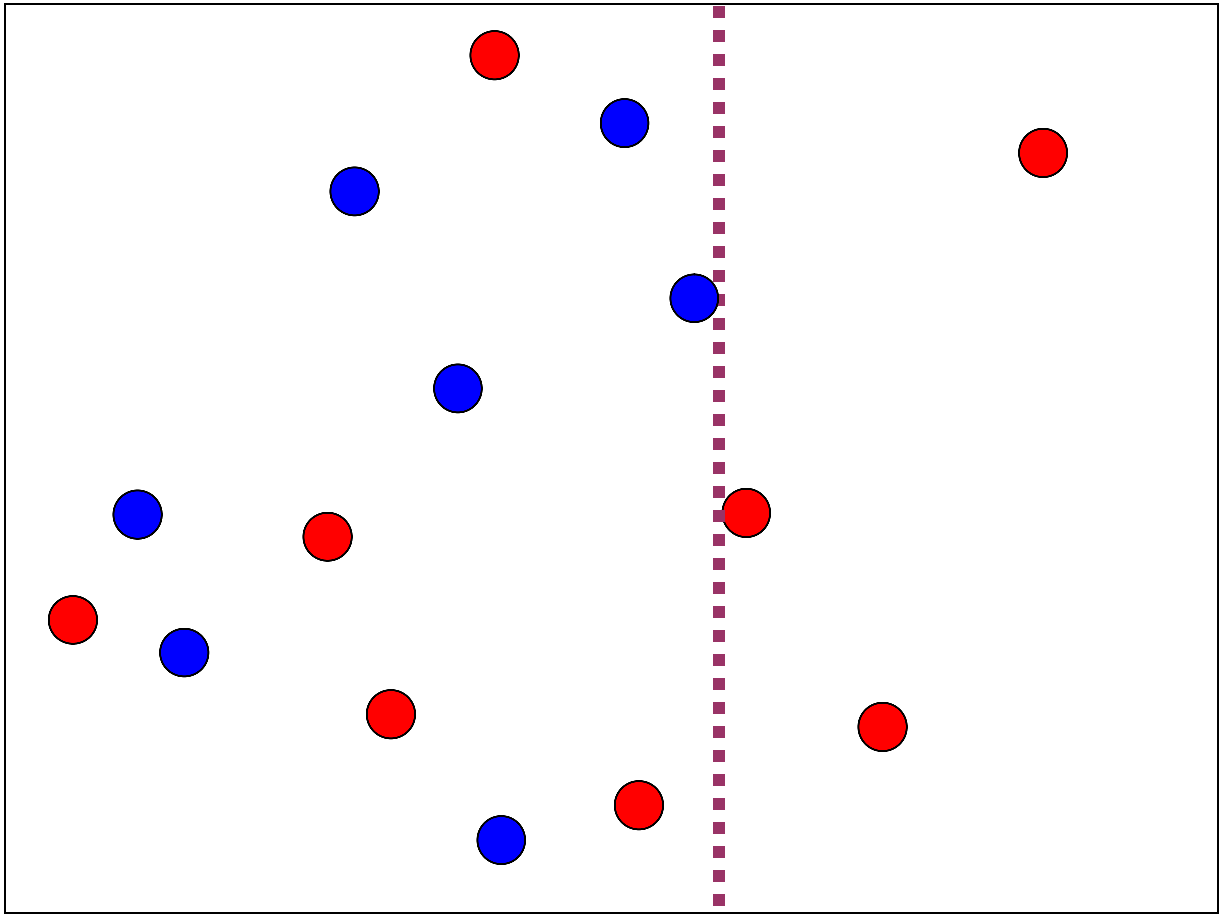
  – Effectiveness of final tree
    - Might be better for pruned trees
    - Or simply more fitted to training data?

  – Effort for pruning algorithm (runtime)

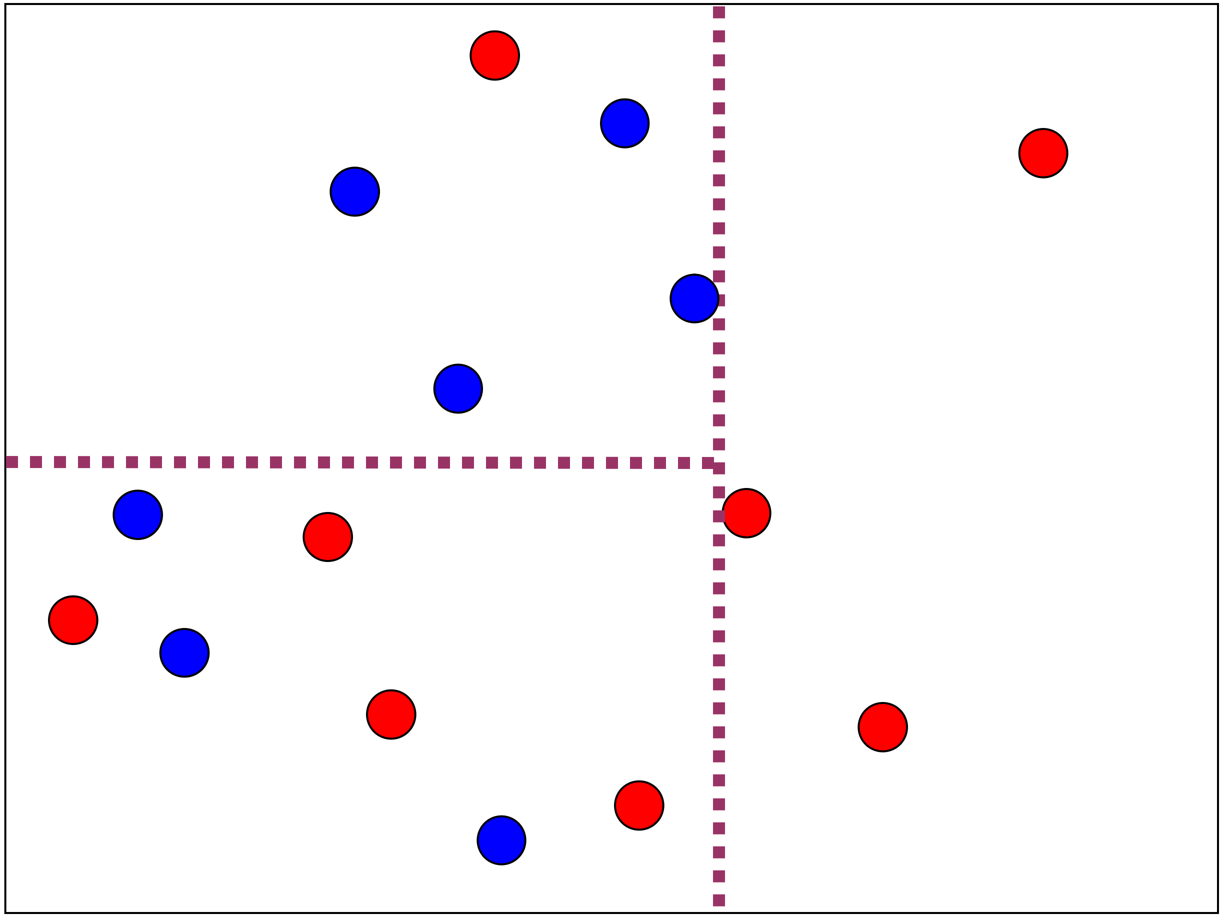
Decision Trees: Stability



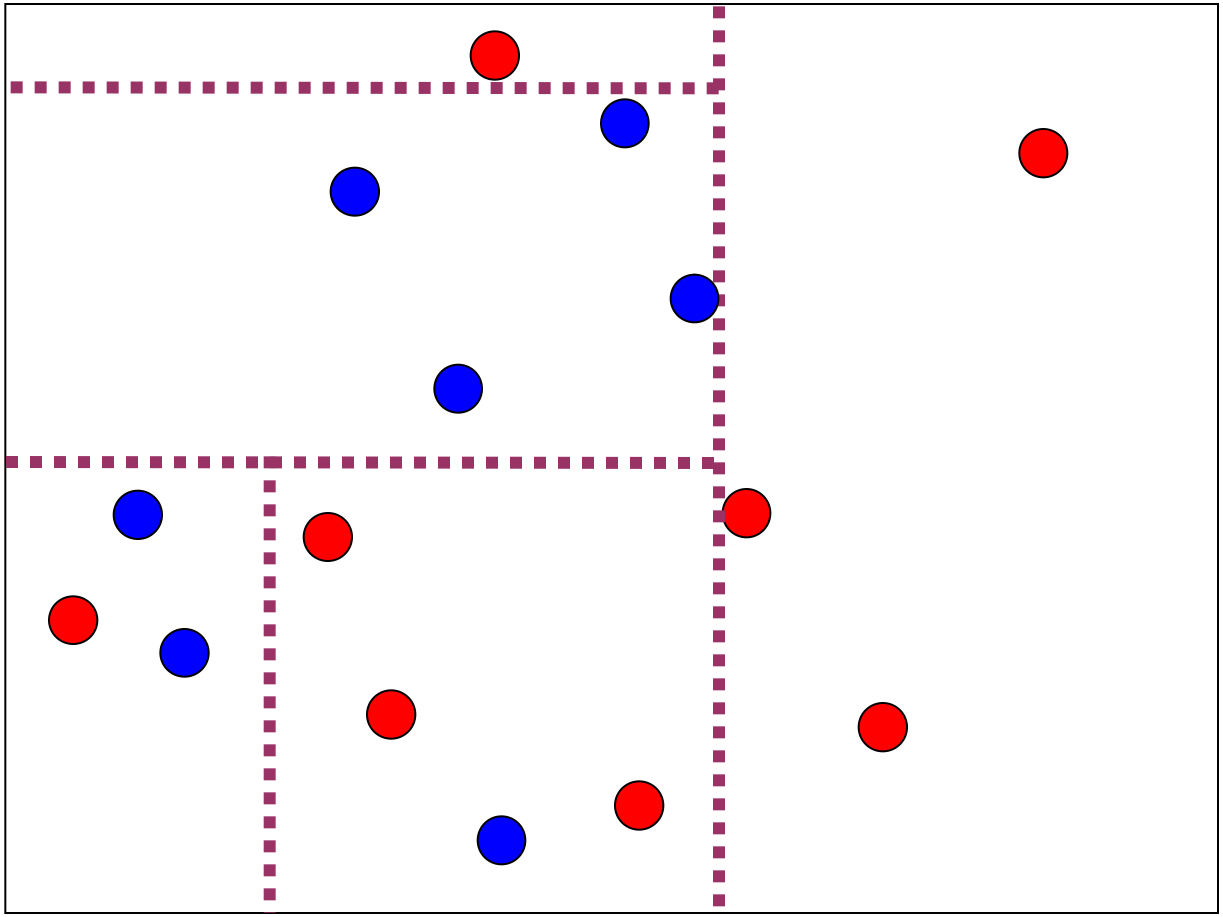
Decision Trees: Stability



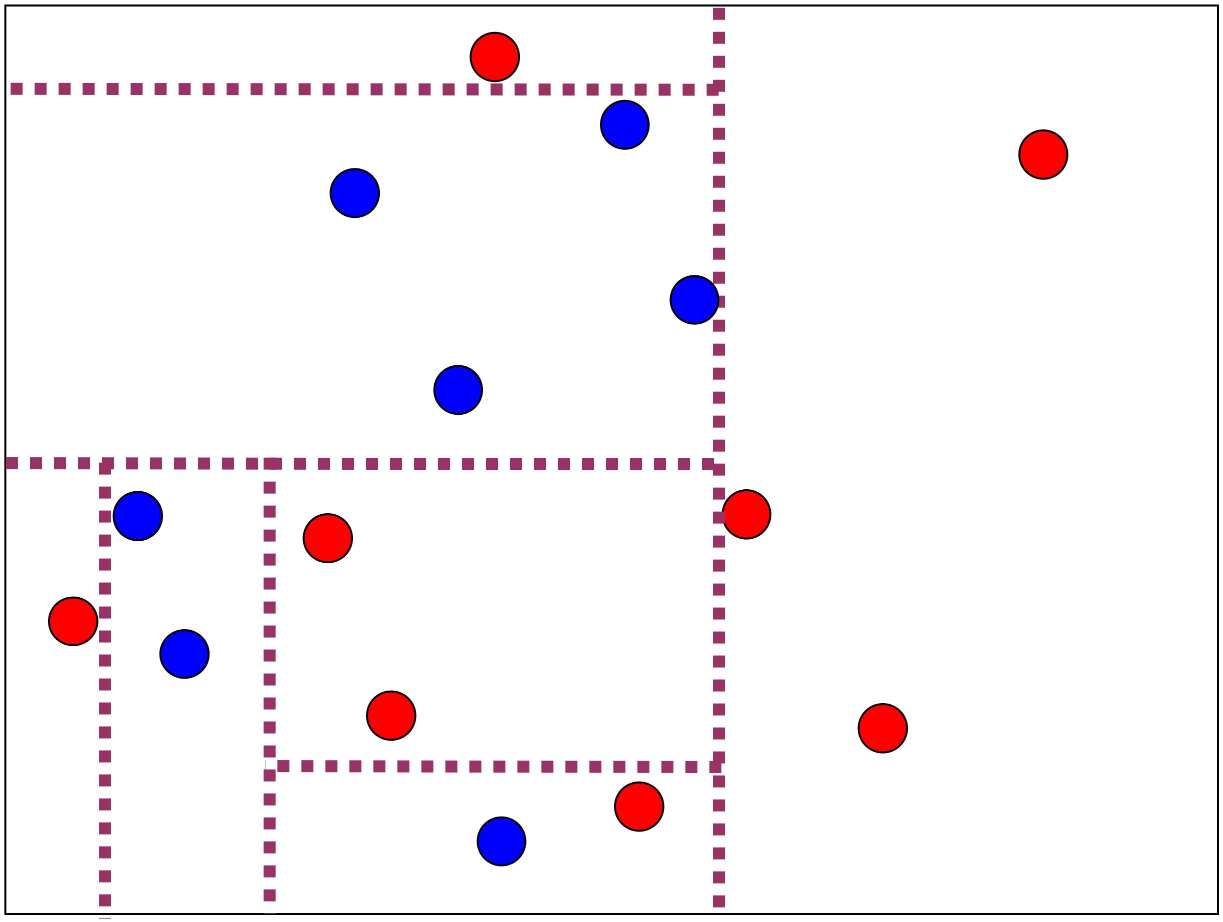
Decision Trees: Stability



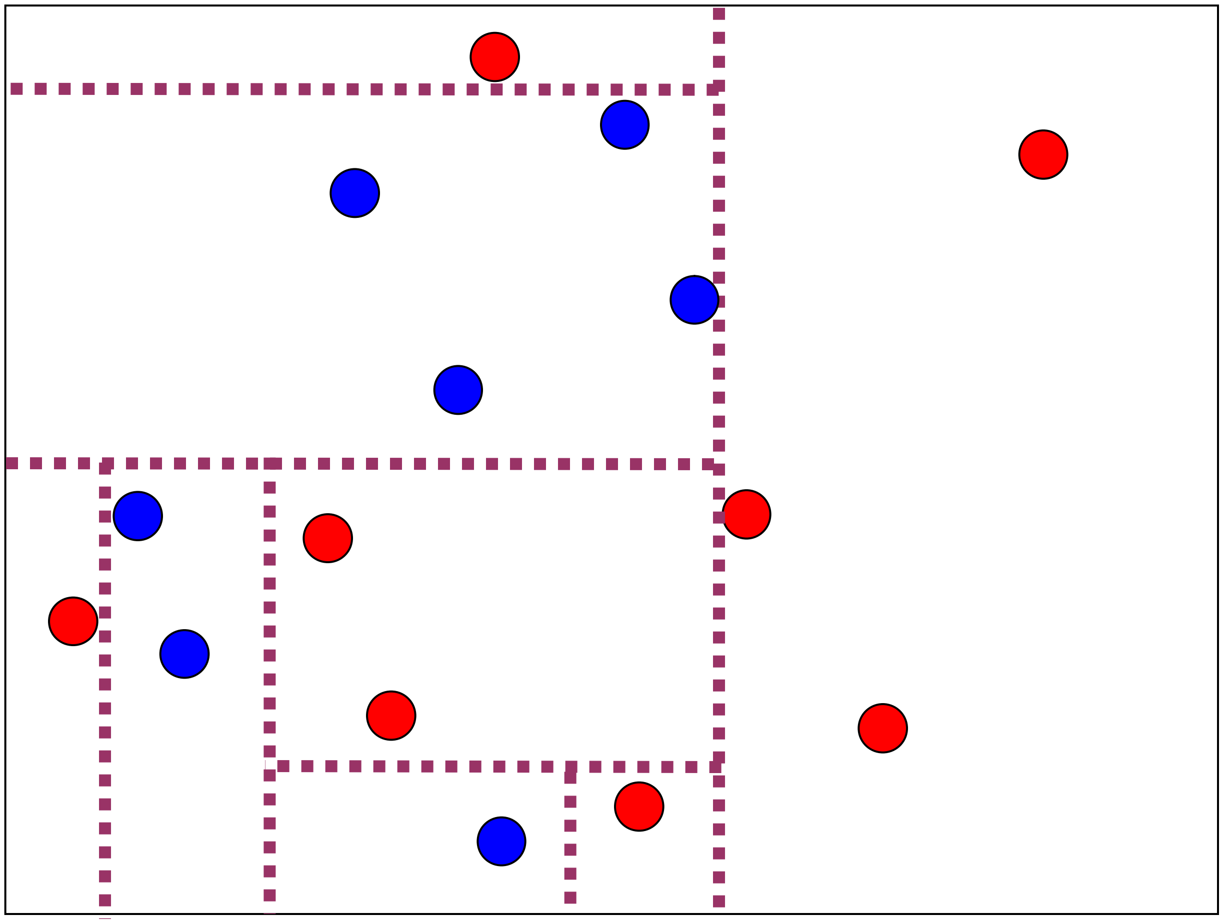
Decision Trees: Stability



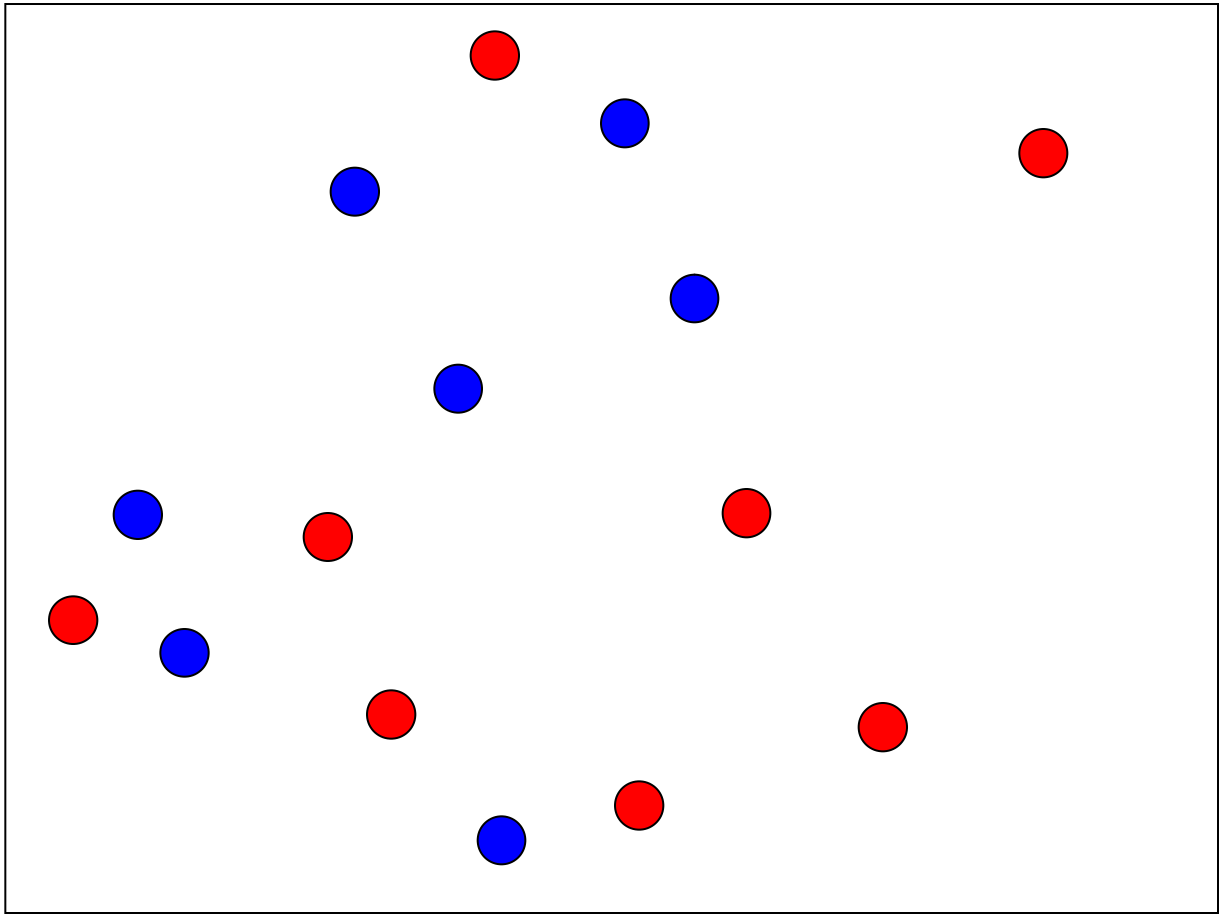
Decision Trees: Stability



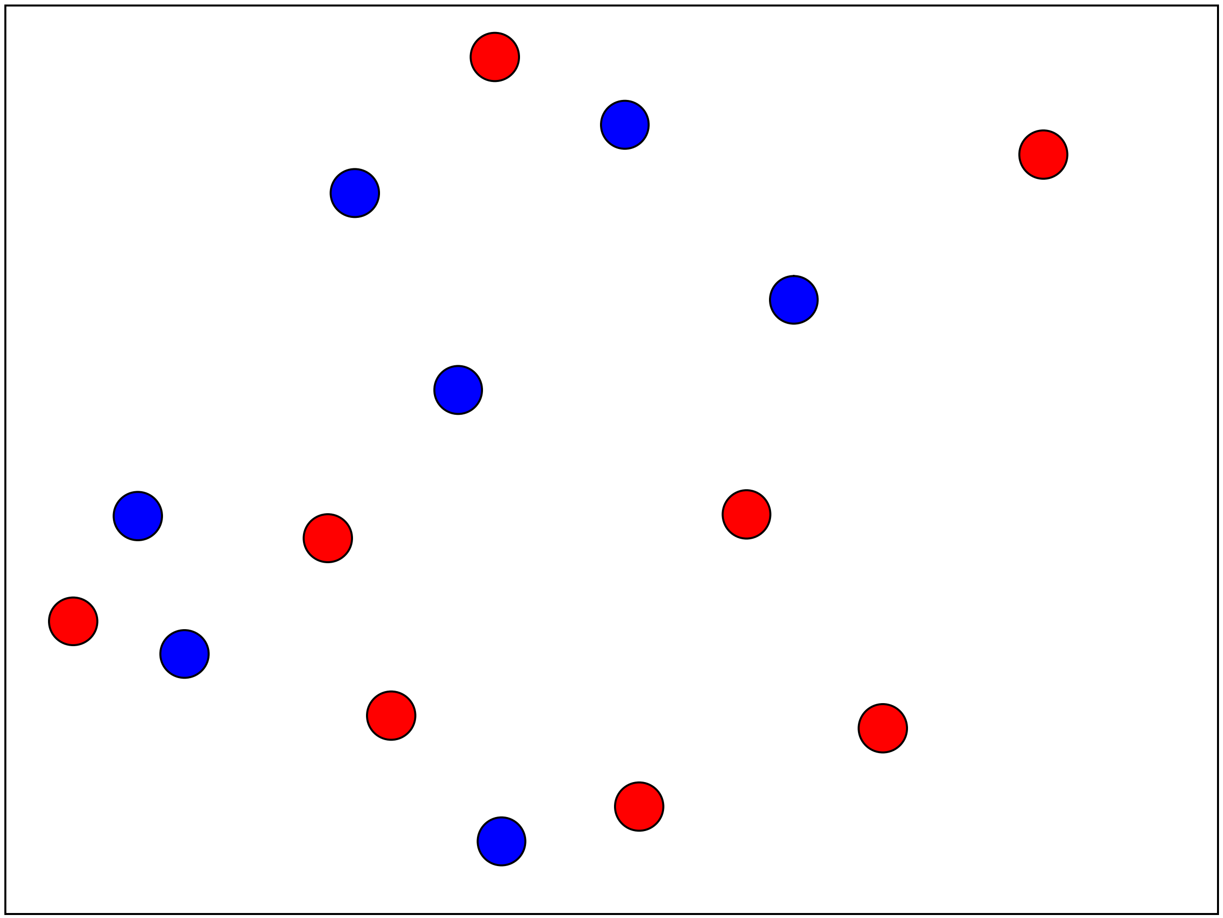
Decision Trees: Stability



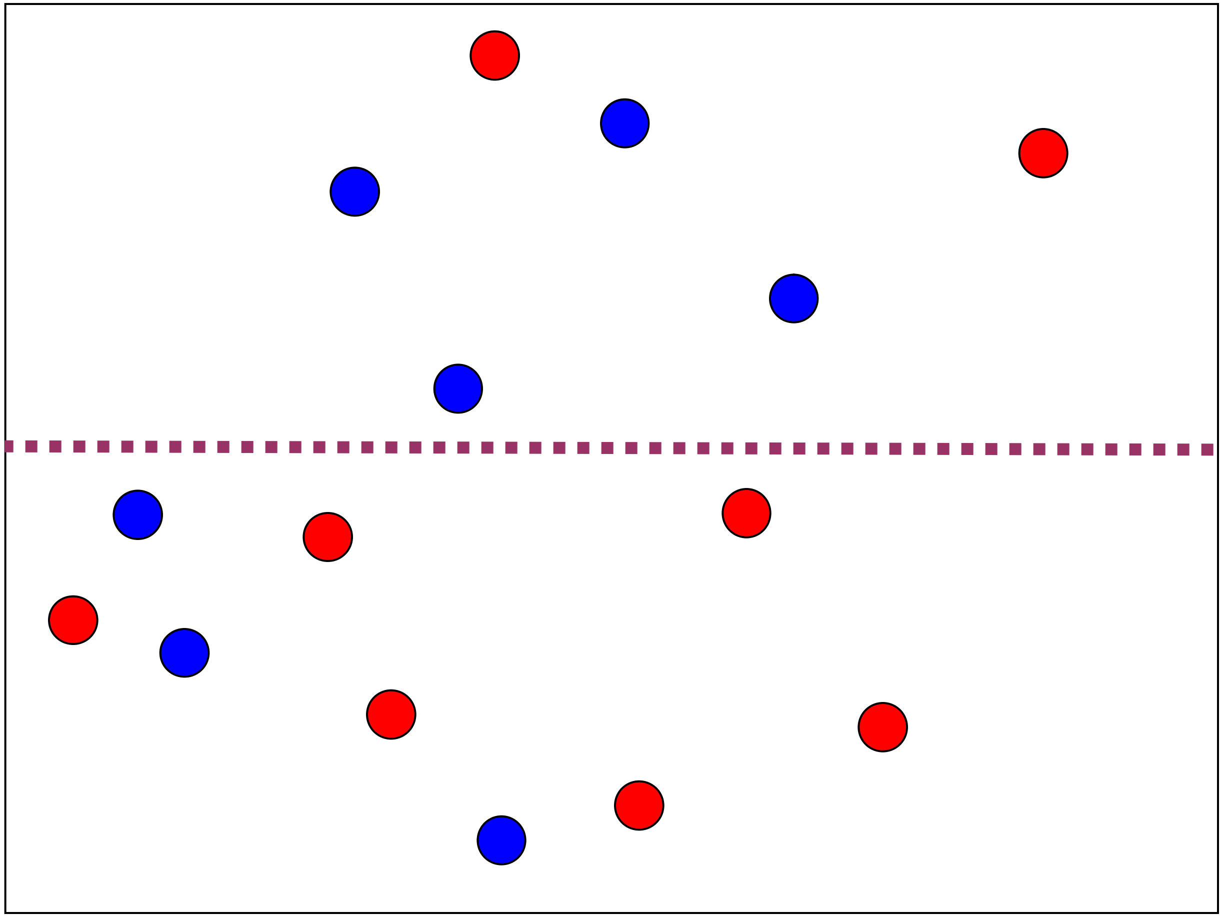
Decision Trees: Stability



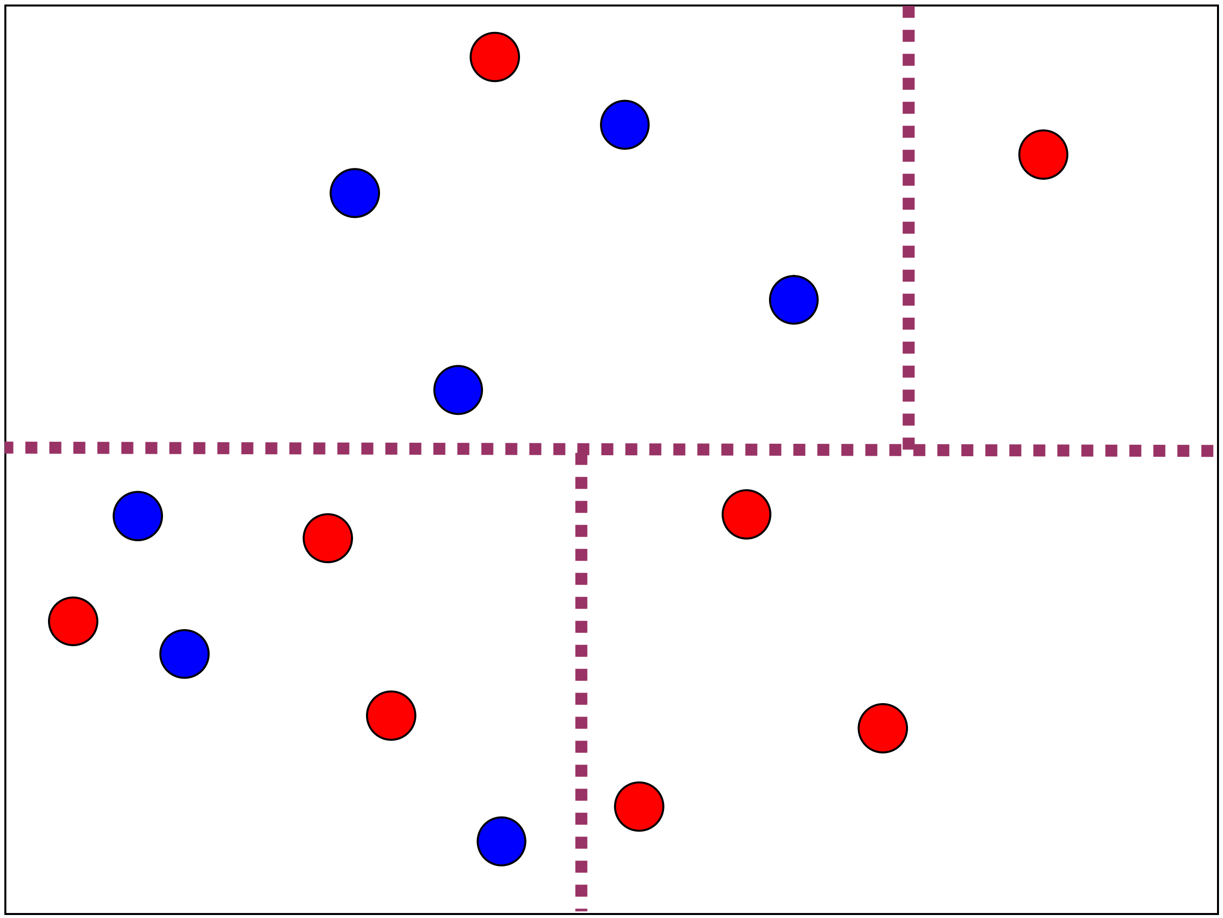
Decision Trees: Stability



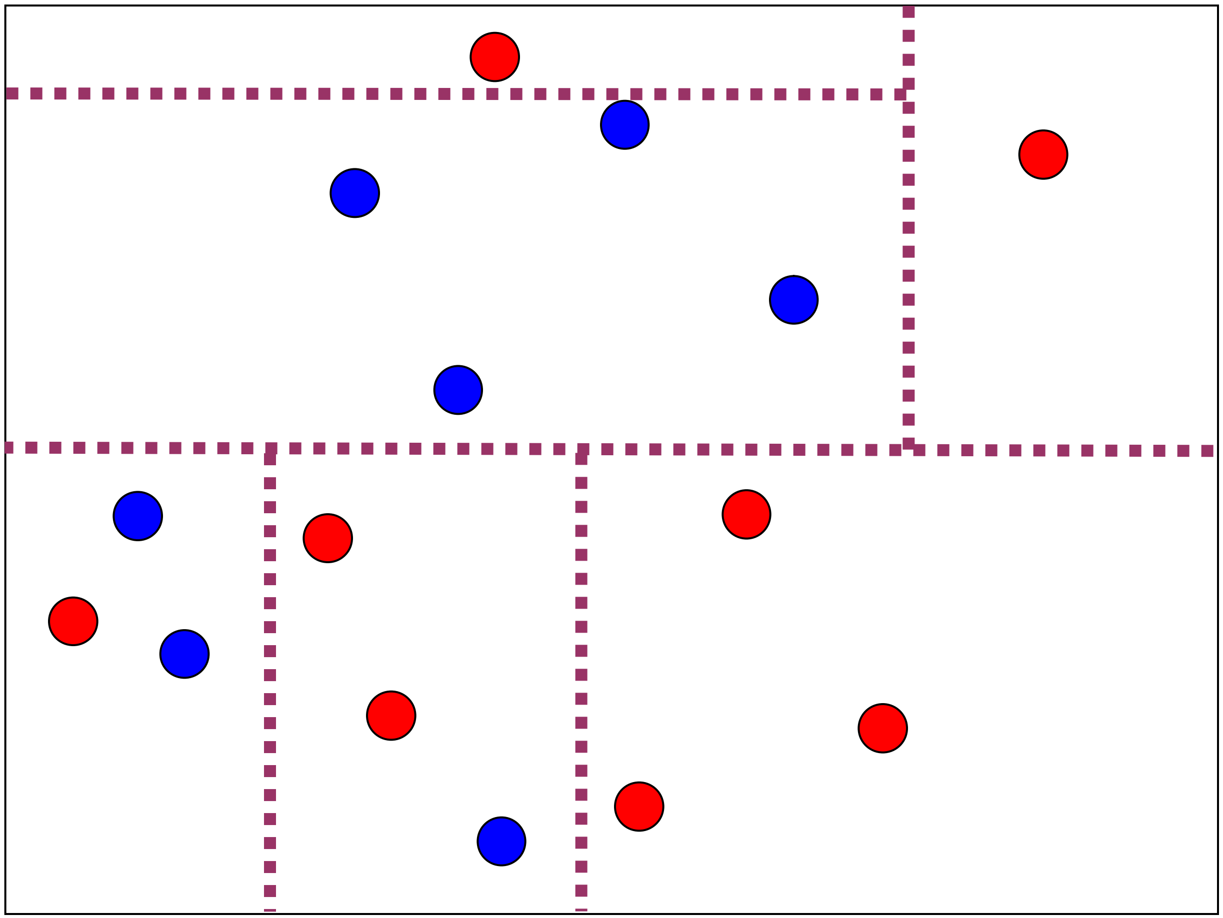
Decision Trees: Stability



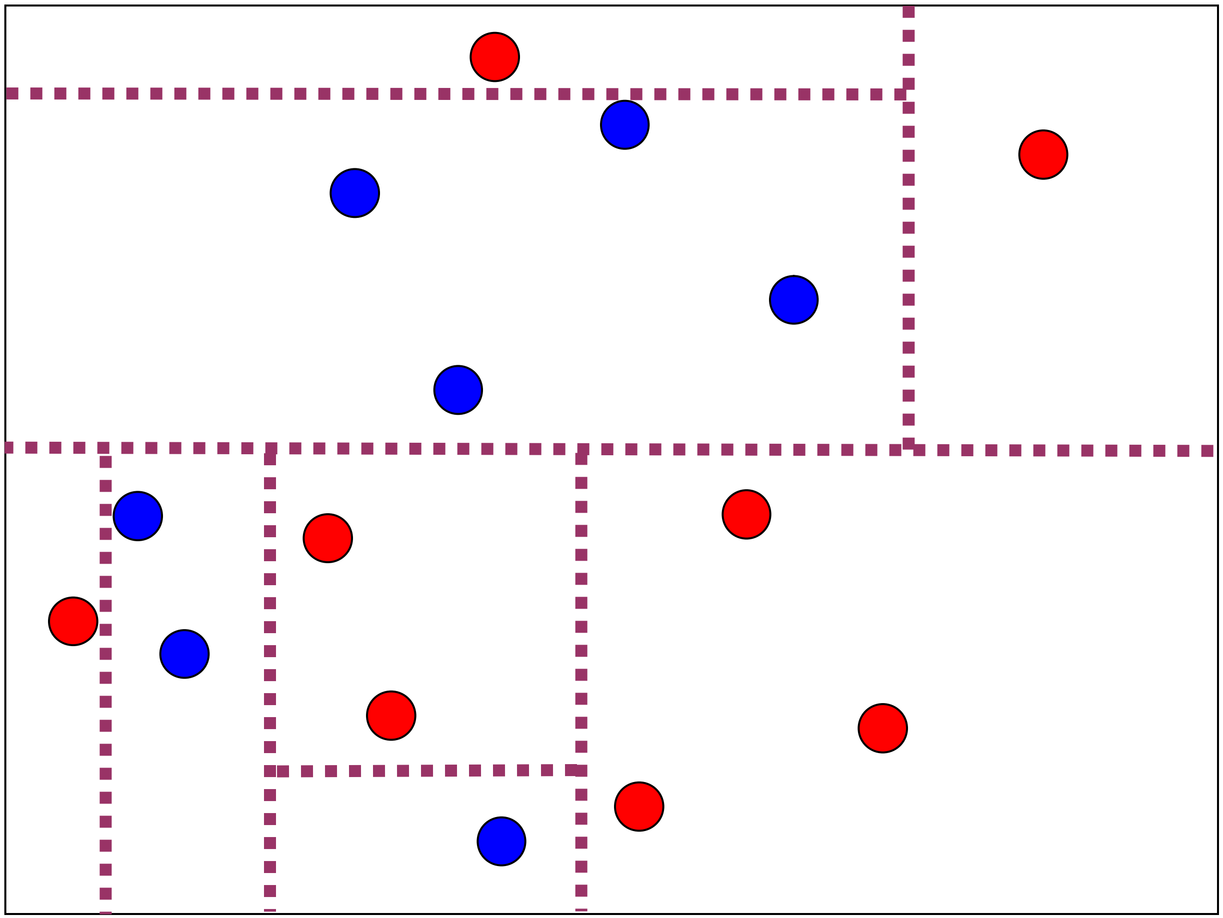
Decision Trees: Stability



Decision Trees: Stability

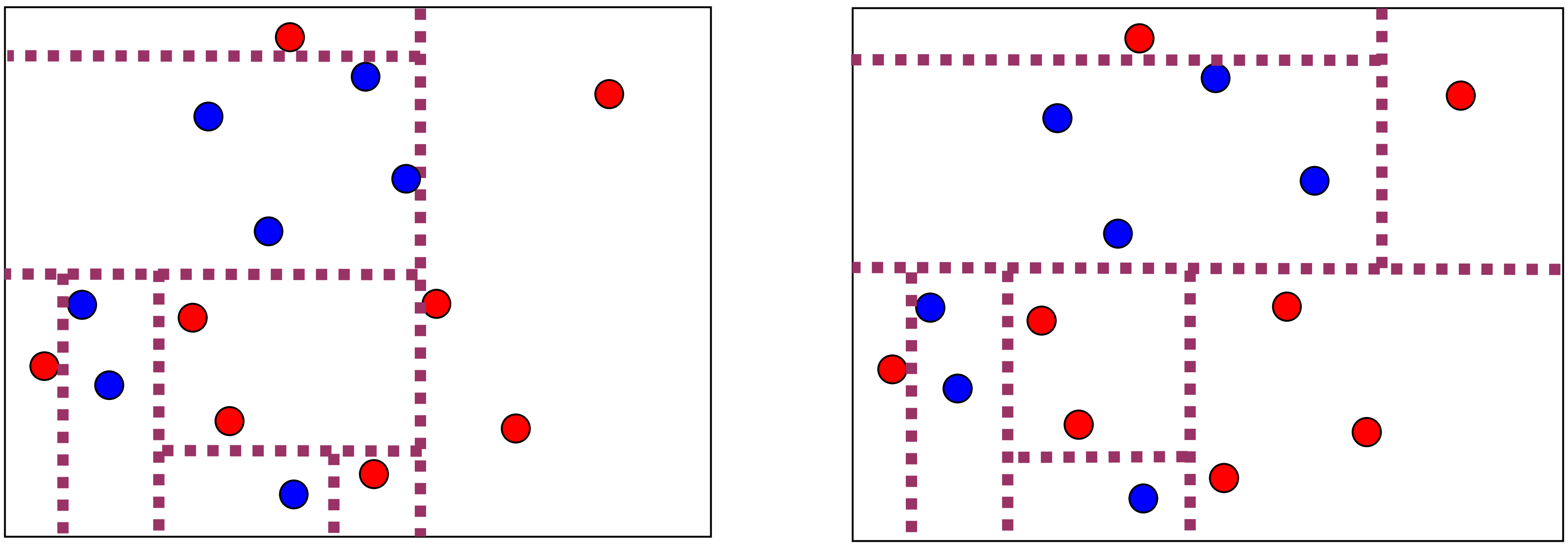


Decision Trees: Stability



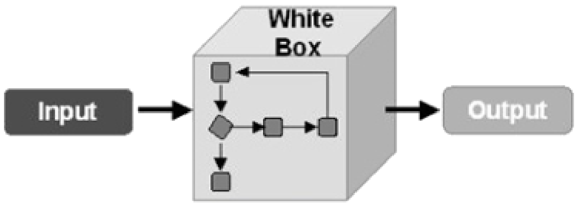
Decision Trees: Stability

Small changes in data
➔ potentially very different tree!



Decision Trees – properties

- Rather old model, well known
- Simple algorithm ➔ easy to understand
  - White box, rather than black-box
  - Used in many non-IT domains
  - Can be used to illustrate expert knowledge

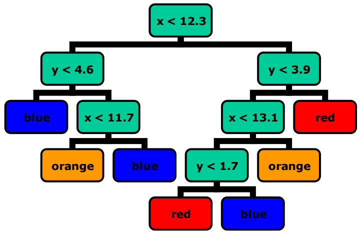


- Various split criteria

- Problems with overfitting – (pre)pruning helps
- Problems with stability ➔ can be exploited!

Decision Trees: algorithm properties

- Previous example: 2D data, x & y axis; BUT:
  - There can be more than two classes



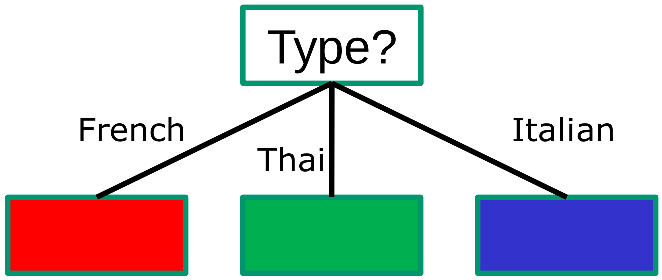
  - Input data can be of any dimensionality
    - E.g. in 3D space of numerical data:
      planes dividing the space along x, y or z axis
  - Input data does not have to be numerical

    ➔ decision trees also work on categorical data

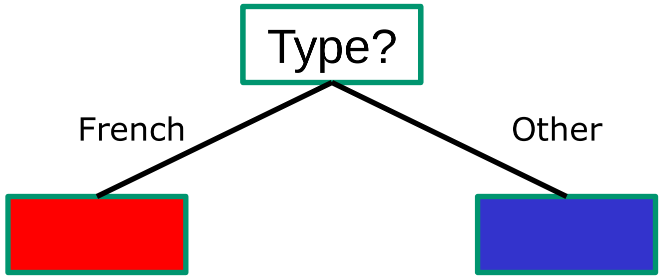
  - Splits not limited to binary (i.e. > two branches)

Decision Trees: categorical data

- *How do we split categorical data?*
  - Imagine an attribute "Type" (of food):
    - French, Italian, Thai
  - One approach:
    each value becomes
    a sub-branch

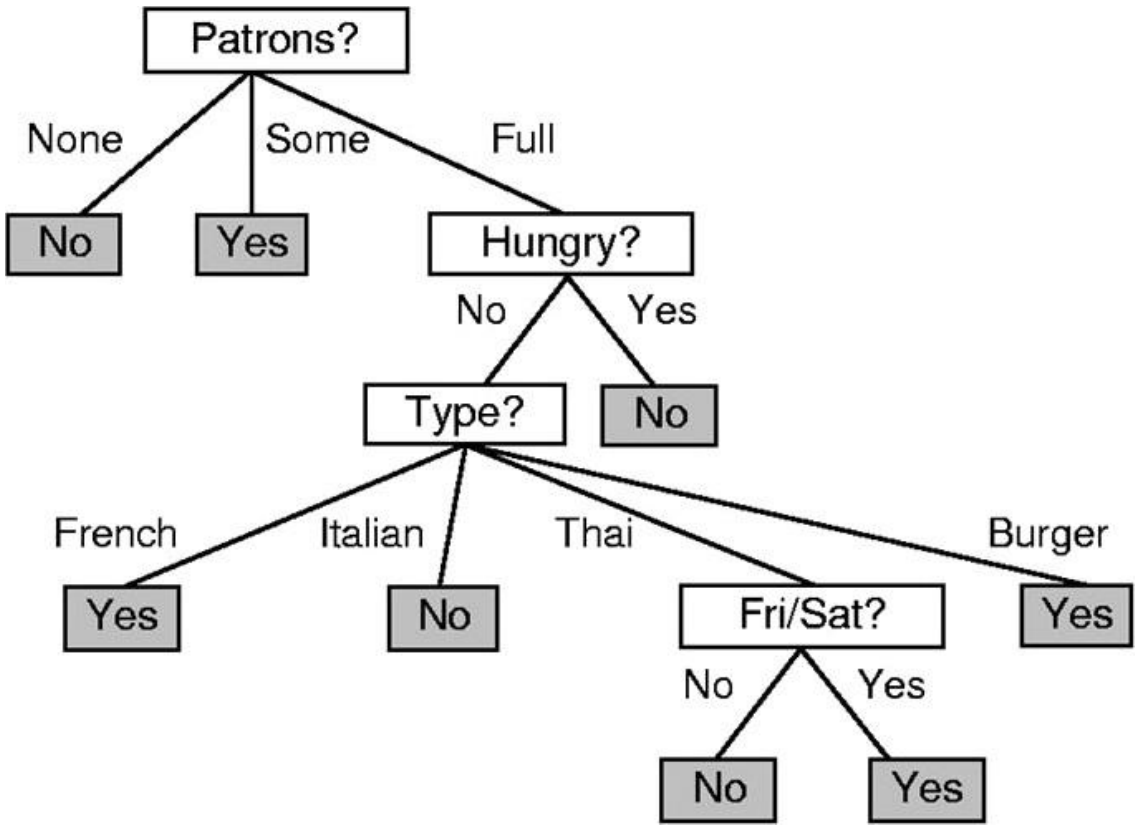


  - Another approach
    - "One vs. all" or other arbitrary combinations
    - Too minimise number of branches



Decision Trees: categorical data

- Example with n-ary splits:



Miles Per Gallon Data Set

cylinders	displacement	horse power	weight	accelleration	Model year	maker	MpG
4	low	low	low	high	75-78	Asia	**good**
6	medium	medium	medium	medium	70-74	America	**bad**
4	medium	medium	medium	low	75-78	Europe	**bad**
8	high	high	high	low	70-74	America	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
4	low	medium	low	medium	70-74	Asia	**bad**
4	low	medium	low	low	70-74	Asia	**bad**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	medium	high	high	79-83	America	**good**
8	high	high	high	low	75-78	America	**bad**
4	low	low	low	low	79-83	America	**good**
6	medium	medium	medium	high	75-78	America	**bad**
4	medium	low	low	low	79-83	America	**good**
4	low	low	medium	high	79-83	America	**good**
8	high	high	high	low	70-71	America	**bad**
4	low	medium	low	medium	75-78	Europe	**good**
5	medium	medium	medium	medium	75-78	Europe	**bad**

18/40 Records subsample

similar in UCI Machine learning repository: https://archive.ics.uci.edu/ml/datasets/auto+mpg

Miles Per Gallon Data Set

cylinders	displacement	horse power	weight	accelleration	Model year	maker	MpG
4	low	low	low	high	75-78	Asia	**good**
6	medium	medium	medium	medium	70-74	America	**bad**
4	medium	medium	medium	low	75-78	Europe	**bad**
8	high	high	high	low	70-74	America	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
4	low	medium	low	medium	70-74	Asia	**bad**
4	low	medium	low	low	70-74	Asia	**bad**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	medium	high	high	79-83	America	**good**
8	high	high	high	low	75-78	America	**bad**
4	low	low	low	low	79-83	America	**good**
6	medium	medium	medium	high	75-78	America	**bad**
4	medium	low	low	low	79-83	America	**good**
4	low	low	medium	high	79-83	America	**good**
8	high	high	high	low	70-71	America	**bad**
4	low	medium	low	medium	75-78	Europe	**good**
5	medium	medium	medium	medium	75-78	Europe	**bad**

Entropy of data set $H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$

12 samples class bad (2/3), 6 samples good (1/3)

Miles Per Gallon Data Set

cylinders	displacement	horse power	weight	accelleration	Model year	maker	MpG
4	low	low	low	high	75-78	Asia	**good**
6	medium	medium	medium	medium	70-74	America	**bad**
4	medium	medium	medium	low	75-78	Europe	**bad**
8	high	high	high	low	70-74	America	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
4	low	medium	low	medium	70-74	Asia	**bad**
4	low	medium	low	low	70-74	Asia	**bad**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	medium	high	high	79-83	America	**good**
8	high	high	high	low	75-78	America	**bad**
4	low	low	low	low	79-83	America	**good**
6	medium	medium	medium	high	75-78	America	**bad**
4	medium	low	low	low	79-83	America	**good**
4	low	low	medium	high	79-83	America	**good**
8	high	high	high	low	70-71	America	**bad**
4	low	medium	low	medium	75-78	Europe	**good**
5	medium	medium	medium	medium	75-78	Europe	**bad**

Entropy of data set:

$-1/3 \times \log_2 1/3 - 2/3 \times \log_2 2/3$ $= -1/3 \times \log(1/3)/\log(2) - 2/3 \times \log(2/3)/\log(2)$

$= -1/3 \times -1{,}59946 - 2/3 \times -0{,}58496$

$= 0{,}918295834$

Miles Per Gallon Data Set

cylinders	displacement	horse power	weight	accelleration	Model year	maker	MpG
4	low	low	low	high	75-78	Asia	**good**
4	low	low	low	low	79-83	America	**good**
4	low	medium	low	medium	75-78	Europe	**good**
4	medium	low	low	low	79-83	America	**good**
4	low	low	medium	high	79-83	America	**good**
4	medium	medium	medium	low	75-78	Europe	**bad**
4	low	medium	low	medium	70-74	Asia	**bad**
4	low	medium	low	low	70-74	Asia	**bad**
5	medium	medium	medium	medium	75-78	Europe	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	high	75-78	America	**bad**
8	high	medium	high	high	79-83	America	**good**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	70-71	America	**bad**

Split on first attribute – cylinders

- Sort data set by cylinders & MpG (output variable)

Miles Per Gallon Data Set

cylinders	displacement	horse power	weight	accelleration	Model year	maker	MpG
4	low	low	low	high	75-78	Asia	**good**
4	low	low	low	low	79-83	America	**good**
4	low	medium	low	medium	75-78	Europe	**good**
4	medium	low	low	low	79-83	America	**good**
4	low	low	medium	high	79-83	America	**good**
4	medium	medium	medium	low	75-78	Europe	**bad**
4	low	medium	low	medium	70-74	Asia	**bad**
4	low	medium	low	low	70-74	Asia	**bad**
5	medium	medium	medium	medium	75-78	Europe	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	high	75-78	America	**bad**
8	high	medium	high	high	79-83	America	**good**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	70-71	America	**bad**

Split on first attribute – cylinders

- Sort data set by cylinders & MpG (output variable)
- Identify subsets: 4 distinct values → 4 sets

Miles Per Gallon Data Set

cylinders	displacement	horse power	weight	accelleration	Model year	maker	MpG
4	low	low	low	high	75-78	Asia	**good**
4	low	low	low	low	79-83	America	**good**
4	low	medium	low	medium	75-78	Europe	**good**
4	medium	low	low	low	79-83	America	**good**
4	low	low	medium	high	79-83	America	**good**
4	medium	medium	medium	low	75-78	Europe	**bad**
4	low	medium	low	medium	70-74	Asia	**bad**
4	low	medium	low	low	70-74	Asia	**bad**
5	medium	medium	medium	medium	75-78	Europe	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	high	75-78	America	**bad**
8	high	medium	high	high	79-83	America	**good**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	70-71	America	**bad**

Split on first attribute – cylinders

- 4 distinct values – split in 4 sets
- Compute IG – compute entropy for each subset

Miles Per Gallon Data Set

cylinders	displacement	horse power	weight	accelleration	Model year	maker	MpG
4	low	low	low	high	75-78	Asia	**good**
4	low	low	low	low	79-83	America	**good**
4	low	medium	low	medium	75-78	Europe	**good**
4	medium	low	low	low	79-83	America	**good**
4	low	low	medium	high	79-83	America	**good**
4	medium	medium	medium	low	75-78	Europe	**bad**
4	low	medium	low	medium	70-74	Asia	**bad**
4	low	medium	low	low	70-74	Asia	**bad**

5 samples class good (5/8), 3 samples class bad (3/8)

$$H(X_{cylinders=4}) = -5/8 \text{ x } \log_2(5/8) - 3/8 \text{ x } \log_2(3/8)$$

$$= (-5/8 \ \log(5/8) \ \log(2)) + (-3/8 \ \log(3/8) \ \log(2))$$

$$= 0{,}954434003$$

Miles Per Gallon Data Set

cylinders	displacement	horse power	weight	accelleration	Model year	maker	MpG
5	medium	medium	medium	medium	75-78	Europe	**bad**

1 sample class bad

$H(X_{cylinders=5}) = 0$

Miles Per Gallon Data Set

cylinders	displacement	horse power	weight	accelleration	Model year	maker	MpG
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	high	75-78	America	**bad**

3 sample class bad

$H(X_{cylinders=6}) = 0$

Miles Per Gallon Data Set

cylinders	displacement	horse power	weight	accelleration	Model year	maker	MpG
8	high	medium	high	high	79-83	America	**good**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	70-71	America	**bad**

1 sample class good (1/6), 5 samples class bad (5/6),

$$H(X_{cylinders=8}) = -1/6 \times \log_2(1/6) - 5/6 \times \log_2(5/6)$$

$$= (-1/6 \ \log(1/6) \ \log(2)) + (-5/6 \ \log(5/6) \ \log(2))$$

$$= 0{,}650022422$$

Miles Per Gallon Data Set

cylinders	displacement	horse power	weight	accelleration	Model year	maker	MpG
4	low	low	low	high	75-78	Asia	**good**
4	low	low	low	low	79-83	America	**good**
4	low	medium	low	medium	75-78	Europe	**good**
4	medium	low	low	low	79-83	America	**good**
4	low	low	medium	high	79-83	America	**good**
4	medium	medium	medium	low	75-78	Europe	**bad**
4	low	medium	low	medium	70-74	Asia	**bad**
4	low	medium	low	low	70-74	Asia	**bad**
5	medium	medium	medium	medium	75-78	Europe	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	high	75-78	America	**bad**
8	high	medium	high	high	79-83	America	**good**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	70-71	America	**bad**

Entropy of split:

$$H(X_{cyl}) = p(x_{cyl=4})H(X_{cyl=4}) + p(x_{cyl=5})H(X_{cyl=5}) + p(x_{cyl=6})H(X_{cyl=6}) + p(x_{cyl=8})H(X_{cyl=8})$$

$$= \frac{8}{18} \times 0{,}95443 + \frac{1}{18} \times 0 + \frac{3}{18} \times 0 + \frac{6}{18} \times 0{,}6500$$

Miles Per Gallon Data Set

cylinders	displacement	horse power	weight	accelleration	Model year	maker	MpG
4	low	low	low	high	75-78	Asia	**good**
4	low	low	low	low	79-83	America	**good**
4	low	medium	low	medium	75-78	Europe	**good**
4	medium	low	low	low	79-83	America	**good**
4	low	low	medium	high	79-83	America	**good**
4	medium	medium	medium	low	75-78	Europe	**bad**
4	low	medium	low	medium	70-74	Asia	**bad**
4	low	medium	low	low	70-74	Asia	**bad**
5	medium	medium	medium	medium	75-78	Europe	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	high	75-78	America	**bad**
8	high	medium	high	high	79-83	America	**good**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	70-71	America	**bad**

Information Gain: $IG(X_A, X_B) = H(X) - p(x_A)H(X_A) - p(x_B)H(X_B)$

Miles Per Gallon Data Set

cylinders	displacement	horse power	weight	accelleration	Model year	maker	MpG
4	low	low	low	high	75-78	Asia	**good**
4	low	low	low	low	79-83	America	**good**
4	low	medium	low	medium	75-78	Europe	**good**
4	medium	low	low	low	79-83	America	**good**
4	low	low	medium	high	79-83	America	**good**
4	medium	medium	medium	low	75-78	Europe	**bad**
4	low	medium	low	medium	70-74	Asia	**bad**
4	low	medium	low	low	70-74	Asia	**bad**
5	medium	medium	medium	medium	75-78	Europe	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	high	75-78	America	**bad**
8	high	medium	high	high	79-83	America	**good**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	70-71	America	**bad**

Information Gain: $IG(X_A, X_B) = H(X) - p(x_A)H(X_A) - p(x_B)H(X_B)$

0,918295834 – 8/18 x 0,954434003 – 6/18 x 0,650022422

= 0,277428803

Miles Per Gallon Data Set

cylinders	displacement	horse power	weight	accelleration	Model year	maker	MpG
4	low	low	low	high	75-78	Asia	**good**
4	low	low	low	low	79-83	America	**good**
4	low	medium	low	medium	75-78	Europe	**good**
4	low	low	medium	high	79-83	America	**good**
4	low	medium	low	medium	70-74	Asia	**bad**
4	low	medium	low	low	70-74	Asia	**bad**
4	medium	low	low	low	79-83	America	**good**
5	medium	medium	medium	medium	75-78	Europe	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	medium	70-74	America	**bad**
6	medium	medium	medium	high	75-78	America	**bad**
4	medium	medium	medium	low	75-78	Europe	**bad**
8	high	medium	high	high	79-83	America	**good**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	75-78	America	**bad**
8	high	high	high	low	70-74	America	**bad**
8	high	high	high	low	70-71	America	**bad**

Split on second attribute – displacement

- 3 distinct values – split in 3 sets
- Compute IG – compute entropy for each subset
- *Finish it at home as exercise!*

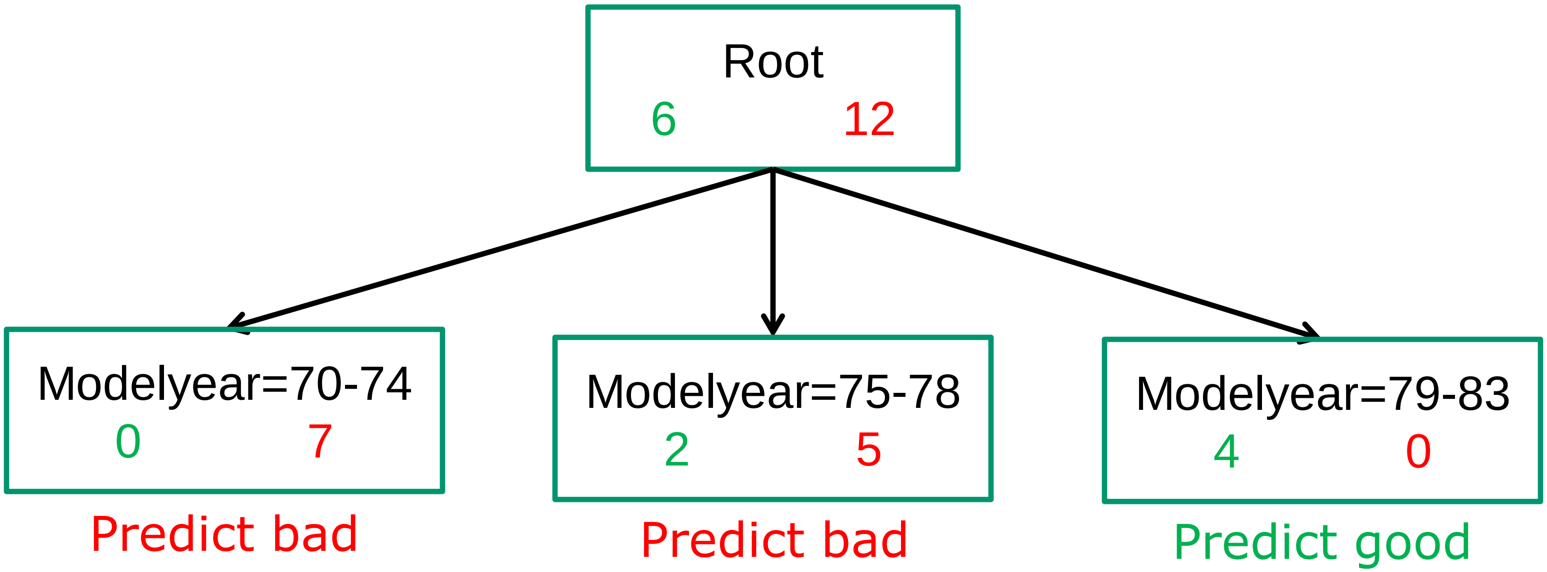
Training the tree

- Build a decision tree

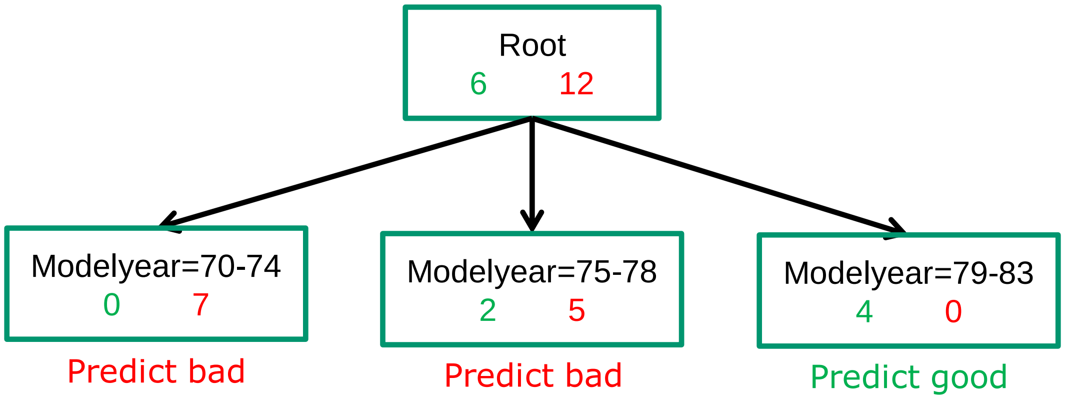
1. Identify splits
2. Compute IGs
3. Select attribute with highest IG
   ➔ Model Year

Attribute	Attr.Value	good	bad	Distribution	Entropy	Info Gain
Full Dataset		6	12		0.9183	
Attribute	**Attr.Value**	**good**	**bad**	**Distribution**	**Entropy**	**Info Gain**
cylinders	*4*	5	3		0.9544	
	5	0	3		0.	
	6	0	3		0.	
	8	1	3		0.8113	
				Split	***0.6045***	***0.3138***
displacement	*low*	4	2		0.9183	
	medium	1	5		0.65	
	high	1	5		0.65	
				Split	***0.7394***	***0.1788***
horse power	*low*	4	0		0.	
	medium	2	7		0.7642	
	high	0	5		0.	
				Split	***0.3821***	***0.5362***
weight	*low*	4	2		0.9183	
	medium	1	5		0.65	
	high	1	5		0.65	
				Split	***0.7394***	***0.1788***
accelleration	*low*	2	7		0.7642	
	medium	1	4		0.7219	
	high	3	1		0.8113	
				Split	***0.7629***	***0.1554***
Model year	*70-74*	0	7		0.	
	75-78	2	5		0.8631	
	79-83	4	0		0.	
				Split	***0.3357***	***0.5826***
maker	*Asia*	1	2		0.9183	
	America	4	8		0.9183	
	Europe	1	2		0.9183	
				Split	0.9183	0.

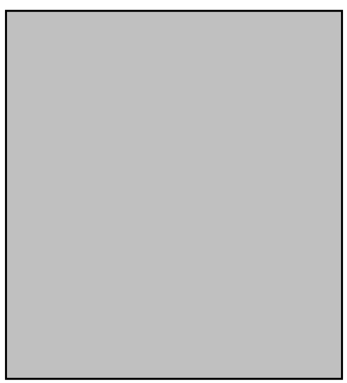
First level of Decision Tree



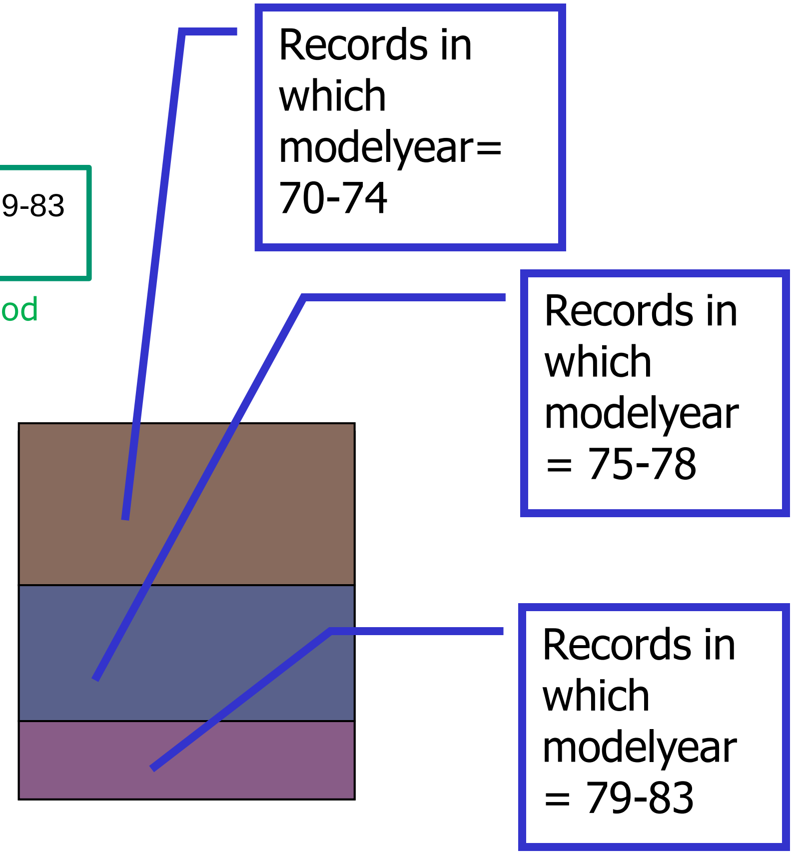
Recursion Step



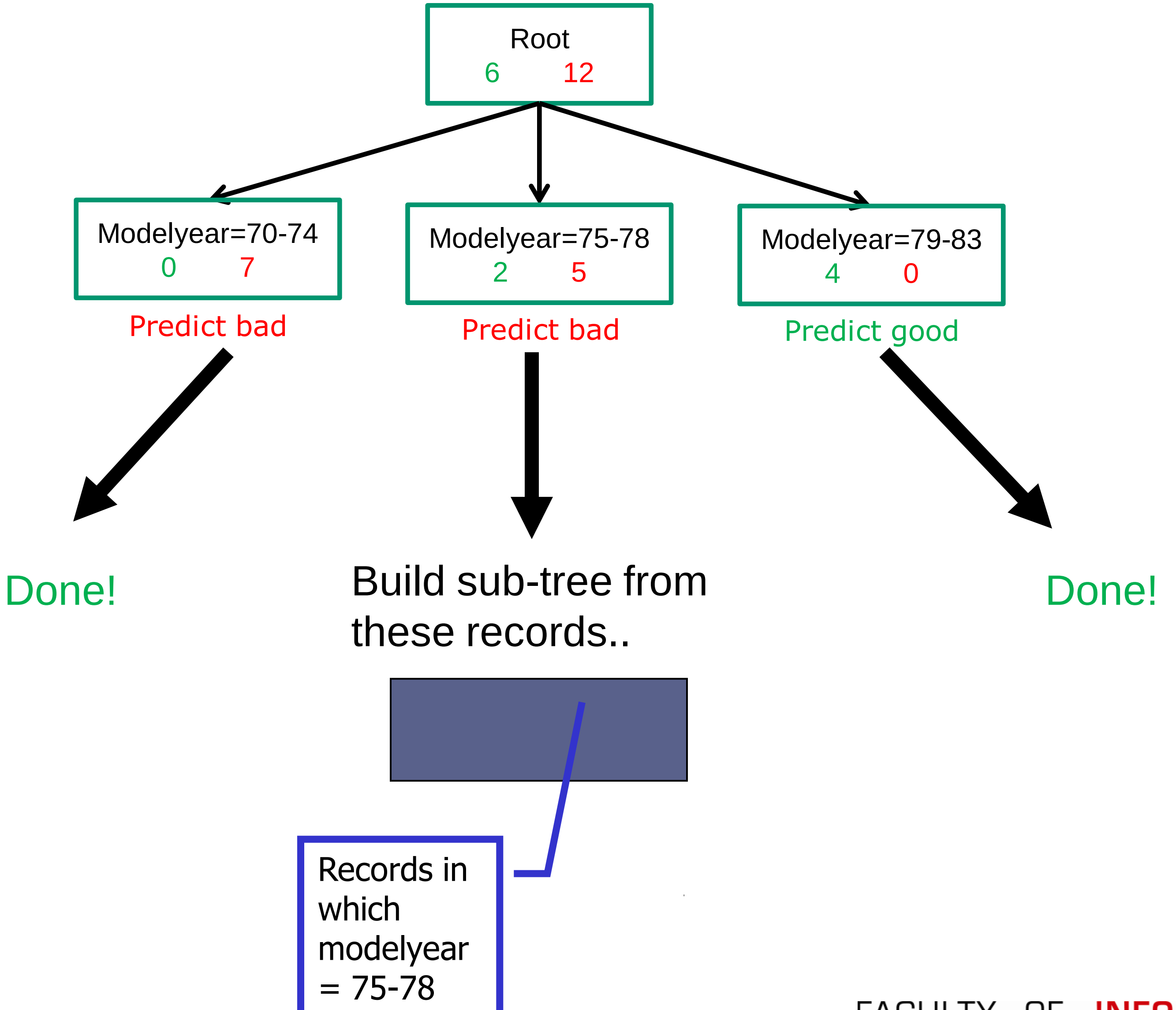
Take the Original Dataset..



And partition it according to the value of the attribute we split on



Recursion Step



Second level of tree

- Only one node from first level needs expansion

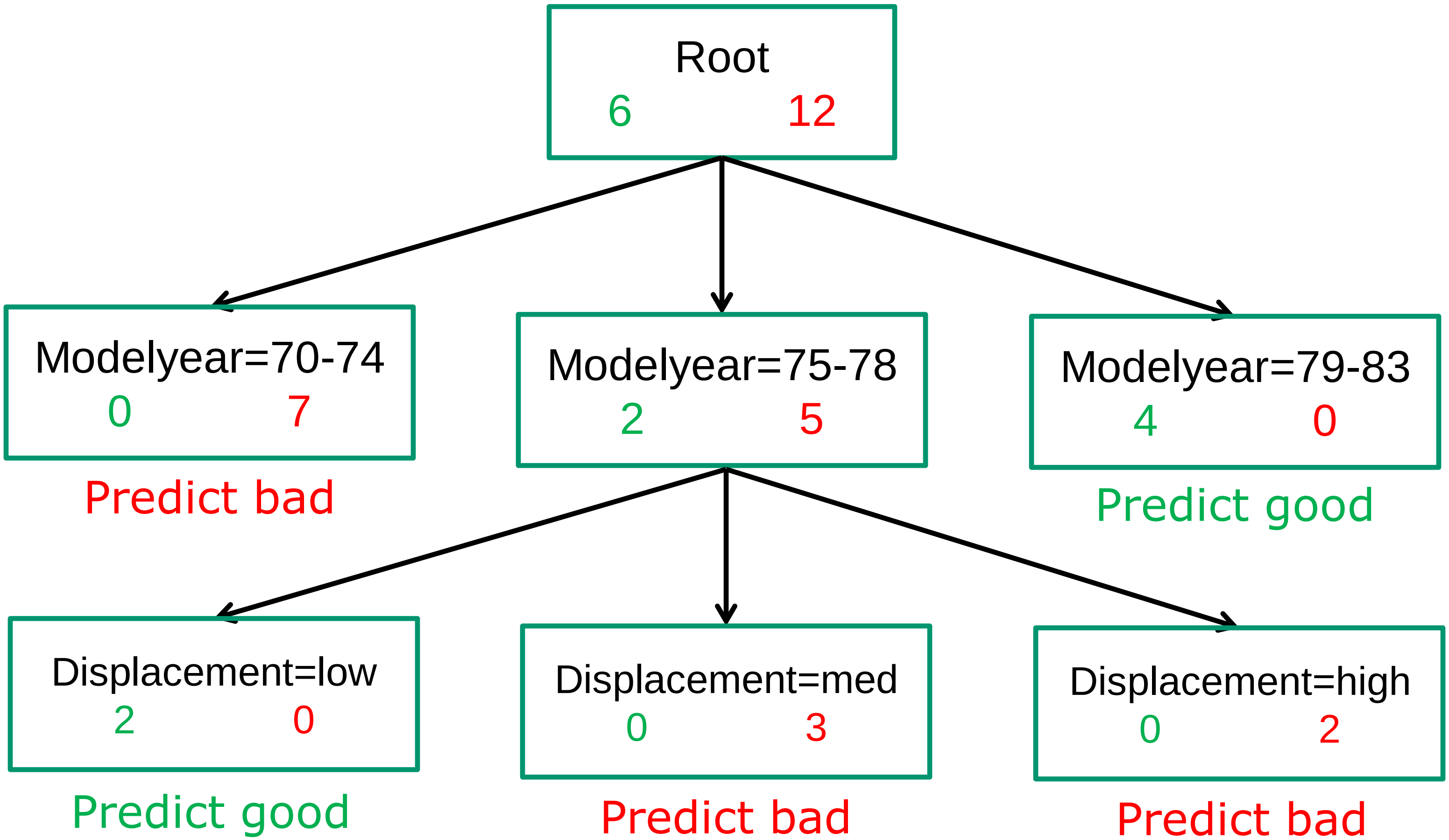
1. Identify splits (on all attributes except *modelyear*)

2. Compute IGs

3. Select attribute with highest IG
   ➔ displacement OR weight

Attribute	Attr.Value	good	bad	Distribution	Entropy	Info Gain
cylinders	*4*	2	1		0.9183	
	5	0	1		0	
	6	0	1		0	
	8	0	2		0	
				Split	***0.3936***	***0.5247***
displacement	*low*	2	0		0	
	medium	0	3		0	
	high	0	2		0	
				Split	***0***	***0.9183***
horse power	*low*	1	0		0	
	medium	1	3		0.8113	
	high	0	2		0	
				Split	***0.4636***	***0.4547***
weight	*low*	2	0		0	
	medium	0	3		0	
	high	0	2		0	
				Split	***0***	***0.9183***
accelleration	*low*	0	3		0	
	medium	1	1		1.	
	high	1	1		1.	
				Split	***0.5714***	***0.3469***
maker	*Asia*	1	0		0	
	America	0	3		0	
	Europe	1	2		0.9183	
				Split	0.3936	0.5247

Second level of tree



Decision Trees

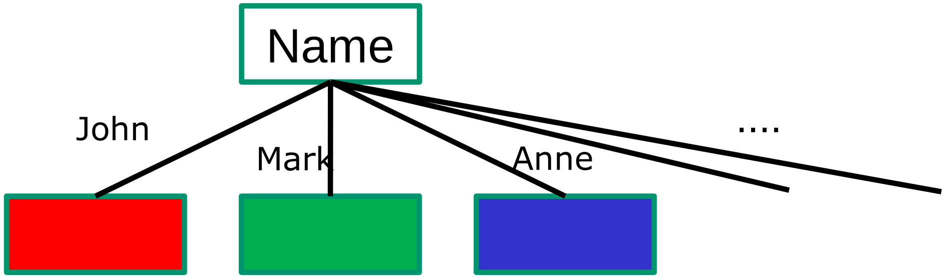
- *Differences between numerical and categorical variables for decision tree learning?*
  - Categorical variables define (one possible) split rather explicitly
    - Either splitting binary, or n-ary
    - In numerical data, we can have way more potential splits
  - *What if we split n-ary on all potential attribute values?*
    - Do not need to consider that attribute in future steps
    - Because it does not separate data in any new way

Information Gain Ratio

- Let‘s assume name is also an attribute (not an ID)

name	sex	age	Play games
John	M	old	N
Mark	M	young	Y
Anne	F	old	Y
Adam	M	young	Y
John	M	young	Y
Alex	F	young	N
Alex	M	old	N
Xena	F	old	N
Tina	F	young	Y
Lucy	F	young	Y

- How will that tree look like?



Information Gain Ratio

- Information gain favours features with many possible outcomes
- Information Gain **Ratio** accounts for that
  - Computes "Value" of an attribute, based on number of different values :

$$V(X) = -\sum_{i=1}^{N} \frac{|T_i|}{|T|} \cdot \log\left(\frac{|T_i|}{|T|}\right)$$

  - "Normalises" Information gain V

$$R(X) = \frac{G(X)}{V(X)}$$

Play-golf decision tree

'Play golf/tennis' data set

Outlook	Temperature	Humidity	Windy	Play?
sunny	85	85	false	Don't Play
sunny	80	90	true	Don't Play
overcast	83	78	false	Play
rain	70	96	false	Play
rain	68	80	false	Play
rain	65	70	true	Don't Play
overcast	64	65	true	Play
sunny	72	95	false	Don't Play
sunny	69	70	false	Play
rain	75	80	false	Play
sunny	75	70	true	Play
overcast	72	90	true	Play
overcast	81	75	false	Play
rain	71	80	true	Don't Play

- *Solve it at home as an exercise !*
- Discussion next lecture

Decision Trees: Algorithm in detail

- For each leaf node
  - If not all data from the same class *(or other stopping criterion)*
    - For ***each*** attribute
      - Identify possible splits of samples into (two or more) subspaces
    - Compute ***best*** split (over all attributes!)
      - Based on a split goodness measure/criterion
  - Until data in all leaf nodes is *pure* (same class)
    - Or cannot be distinguished (*When can this happen?)*
    - *Or other stopping criterion fullfilled (e.g. maximum depth)*

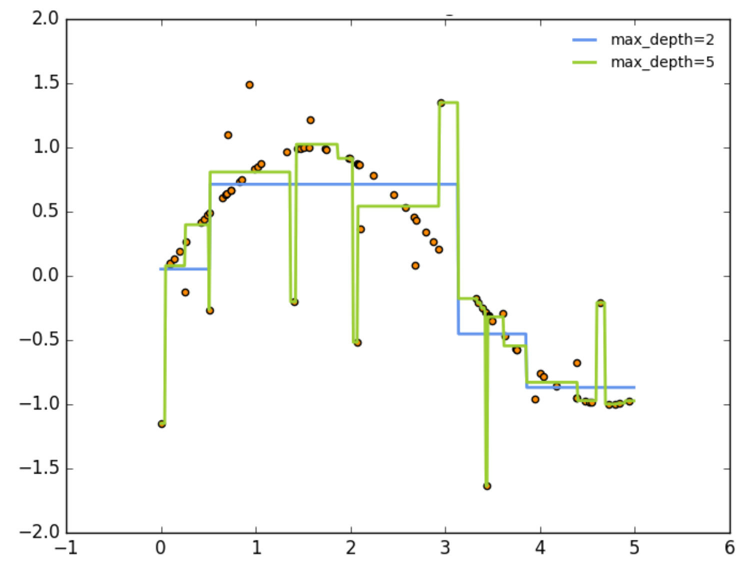
Decision Trees: Algorithm in detail

- For ***each*** attribute
  - Identify possible splits of samples into (two or more) subspaces
    - categorical variables? (e.g. size with values “small” / “medium” / “large”)
      - *By each variable value, i.e. split into 3 sub-branches*
      - *Or one value vs. other values: small vs. rest, medium vs rest, large vs. rest (split into 2 sub-branches)*
      - *Difference?*
    - numerical variables? (e.g. size in centimeters)
      - *sort values & split between each pair of values*
      - *➔ How many candidate splits?*

Decision Trees – Regression

- *Can decision trees also work for regression tasks?*

- *How to compute output?*
  - *Average of all values in the leaf node*



- *Any changes in learning?*
  - *Split goodness computation*
    - *Means squared error (MSE), mean absolute error (MAE), ...*

Outline

- Short recap
- Decision Trees (continued)
- Evaluation (continued)

Confusion Matrix

- Matrix of classification results per class
  - Size (# classes) x (# classes)
- For each actual class plot the predicted classes
- Shows accuracy for single classes
- Indicates which classes are confused

Confusion Matrix: Example

	Grey	Black	Red	*Accuracy*
Grey	5	3	0	*0.625*
Black	2	3	1	*0.500*
Red	0	1	12	*0.920*
				0.740

- *How does the ideal matrix look like?*
  - Numbers only in the diagonal
    - In other cells: indicates misclassification

Confusion Matrix

- Important to analyse mistake patterns
  - *Which classes get mixed up?*

classified as											
a	b	c	d	e	f	g	h	i	j	k	genre
34	3	0	0	2	8	0	0	2	10	1	a = Country
9	39	0	1	1	4	0	0	0	5	1	b = Folk
0	2	47	0	1	4	1	0	1	4	0	c = Grunge
0	2	0	39	0	3	1	6	8	0	1	d = Hip-Hop
2	3	3	0	34	4	10	0	0	4	0	e = Metal
10	3	9	4	4	11	3	2	1	11	2	f = Pop
5	2	5	0	10	2	36	0	0	0	0	g = Punk Rock
2	0	0	10	0	3	0	40	2	1	2	h = R&B
0	1	0	7	0	1	0	2	45	0	4	i = Reggae
8	1	8	1	3	5	1	1	1	27	4	j = Slow Rock
1	0	0	0	0	1	0	1	3	2	52	k = Children's

Confusion Matrix

- Important to analyse mistake patterns
  - Which classes get mixed up

classified as											
a	b	c	d	e	f	g	h	i	j	k	genre
34	3	0	0	2	8	0	0	2	10	1	a = Country
9	39	0	1	1	4	0	0	0	5	1	b = Folk
0	2	47	0	1	4	1	0	1	4	0	c = Grunge
0	2	0	39	0	3	1	6	8	0	1	d = Hip-Hop
2	3	3	0	34	4	10	0	0	4	0	e = Metal
10	3	9	4	4	11	3	2	1	11	2	f = Pop
5	2	5	0	10	2	36	0	0	0	0	g = Punk Rock
2	0	0	10	0	3	0	40	2	1	2	h = R&B
0	1	0	7	0	1	0	2	45	0	4	i = Reggae
8	1	8	1	3	5	1	1	1	27	4	j = Slow Rock
1	0	0	0	0	1	0	1	3	2	52	k = Children's
47	69	65	63	62	23	69	76	71	42	77	Precision
57	65	78	65	57	18	6	67	75	45	87	Recall

Confusion Matrix: Example

	BigClass	SmallClass	*Accuracy*
BigClass	490	0	*100*
SmallClass	10	0	*0*
			0.98

Evaluation measures – averages

- Previous measures are ***micro-averaged***
- Do not indicate issues with imbalanced classes
- Alternative: macro-averaged measures
  – Compute precision, recall, ... ***per class***
  – Average class-results

Evaluation – micro average

	BigClass	SmallClass	*Accuracy*
BigClass	490	0	*100*
SmallClass	10	0	*0*
			0.98

- Accuracy: $$\frac{TP + TN}{TP + FP + TN + FN} = \frac{TP + TN}{\#\ samples}$$

Evaluation – macro average

	BigClass	**SmallClass**	***Accuracy***
BigClass	490	0	*100*
SmallClass	10	0	*0*
			0.5

- Accuracy: $$\frac{1}{|C|}\sum_{i=1}^{|C|}\frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i}$$

Performance per class

- Important to consider when
  - imbalanced classes
  - Performance of a particular class is more important

- *Examples ?*
  - Health prediction
  - Classify sensitive documents, …
  - Spam filter
  - Identify malicious software

Costs of misclassification

- Cost / loss functions
  - Measures per class with weighted averages
  - Higher weight to classes where errors are more severe

  ➔ Requires expert knowledge to identify weights

Effectiveness & Efficiency

- Effectiveness: quality of classification
  - Accuracy, precision, recall, F1, …

- Efficiency: computational efficiency (speed, runtime) of a classification

- Performance: often used as *synonym* for either *effectiveness OR efficiency* !

Effectiveness & Efficiency

- What is more important?

- Trade-off between effectiveness & efficiency
- Differentiate between efficiency on
  - Training (learning) a model
  - Classification

- Efficiency is more relevant if model needs to be (re-)trained frequently

Leave-*p*-out Cross validation

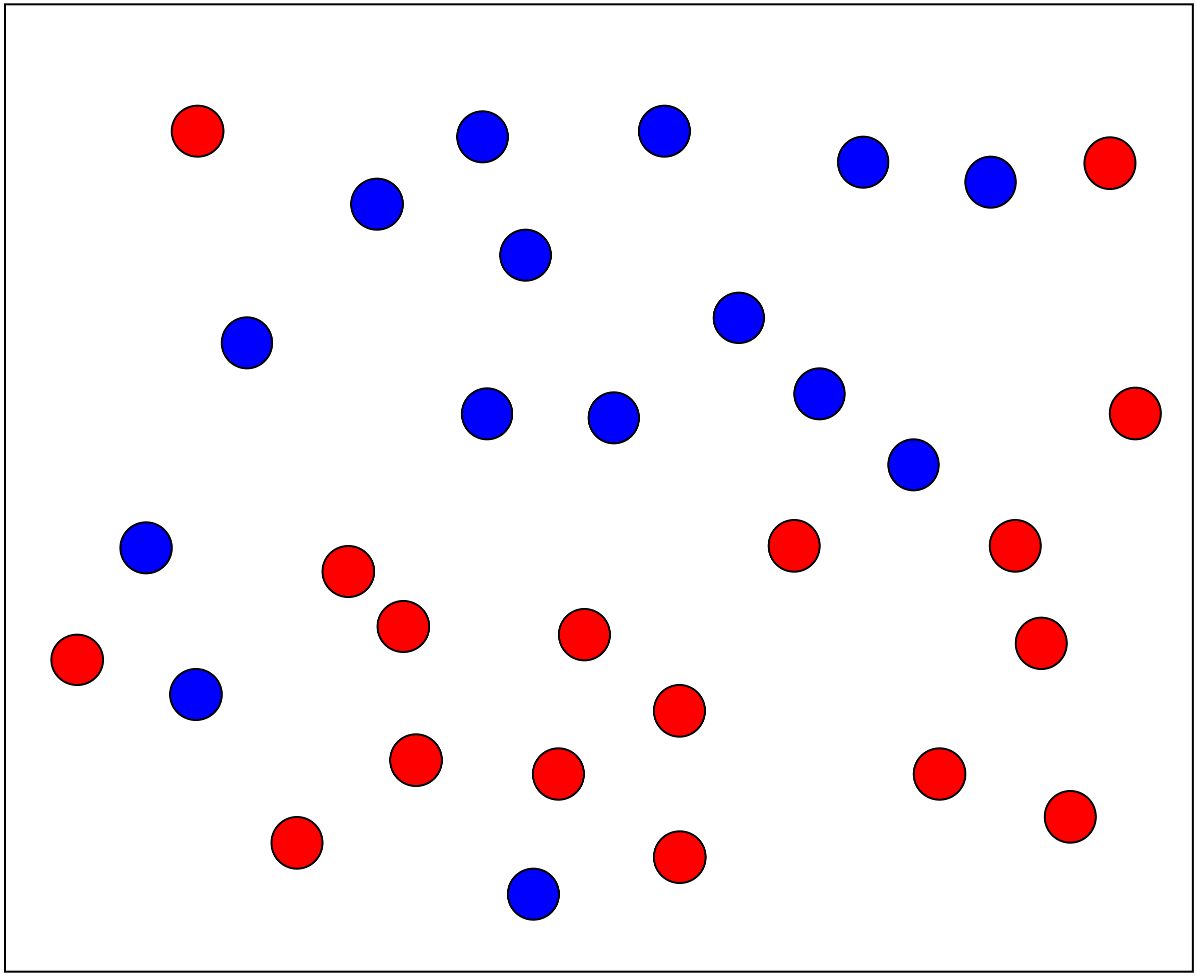
- A type of ***exhaustive*** cross-validation
  - Use *p* observations in test (validation) set
    - Remaining samples are in training set
  - Repeated **for all combinations** to cut *p* samples
  - Quickly becomes computationally infeasible
    - 100 samples, p=30
      - **3×10^{25}** combinations!

- Special case: p=1, leave-one-out cross validation
  - Test/validation set contains one sample
  - Number of combinations?
    - **n**

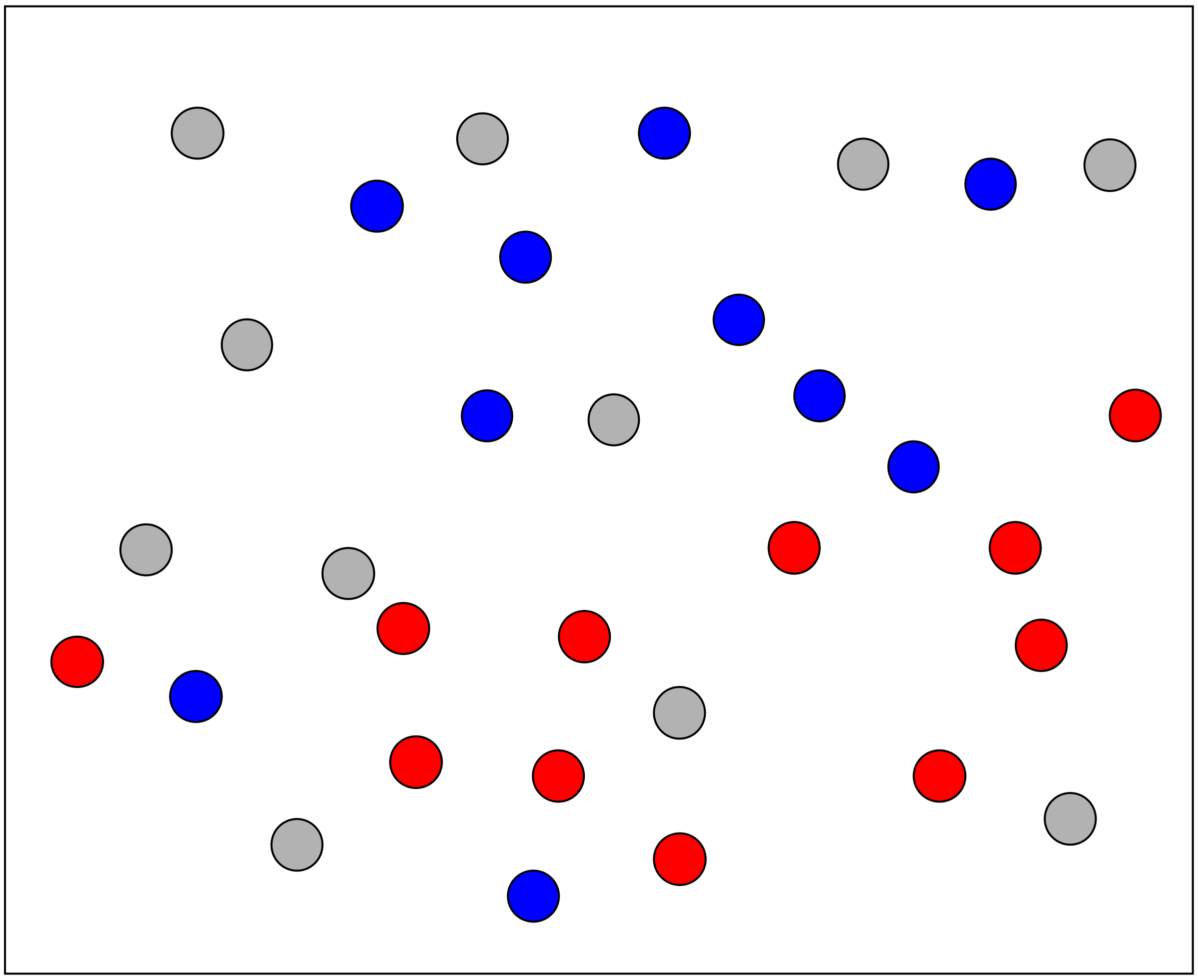
Bootstrapping

- A bootstrap sample is a **random subset** of the data sample
- Data points may be selected repeatedly
  - i.e. selection with replacement
- An arbitrary number of bootstrap samples may be used
- Bootstrapping is an alternative to cross validation and holdout method (training-test split)
- Testing often on “out-of-bag samples”

Example: Bootstrapping



Example: Bootstrapping



Overfitting & Generalisation

- Overfitting: model is trained too specific to learning examples
  - Examples of classifiers?

- Generalisation: ability of model to perform well on the general problem
  - i.e. the real distribution that generated the training data

Questions?