Statistical Machine Learning (BE4M33SSU)
Lecture 12: Random Forests

Jan Drchal

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science

Overview

Topics covered in the lecture:

- Decision trees for classification and regression
- Combining models by means of bagging
- Random forests

Resources

- Hastie, Tibshirani and Friedman: *The Elements of Statistical Learning*, 2009
- Duda, Hart and Stork: *Pattern Classification*, 2000
- Criminisi et al.: *Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning* , 2011
- Gilles Louppe: *Understanding Random Forests: From Theory to Practice*, 2014

Decision Tree

- Supervised machine learning model
- Interpretable
- Supports both classification and regression (regression trees)
- Binary/multi-valued/continuous inputs
- Can deal with missing values
- Fast training and prediction

Decision Tree Example

- Will John play tennis?

Training examples: **9 yes / 5 no**

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

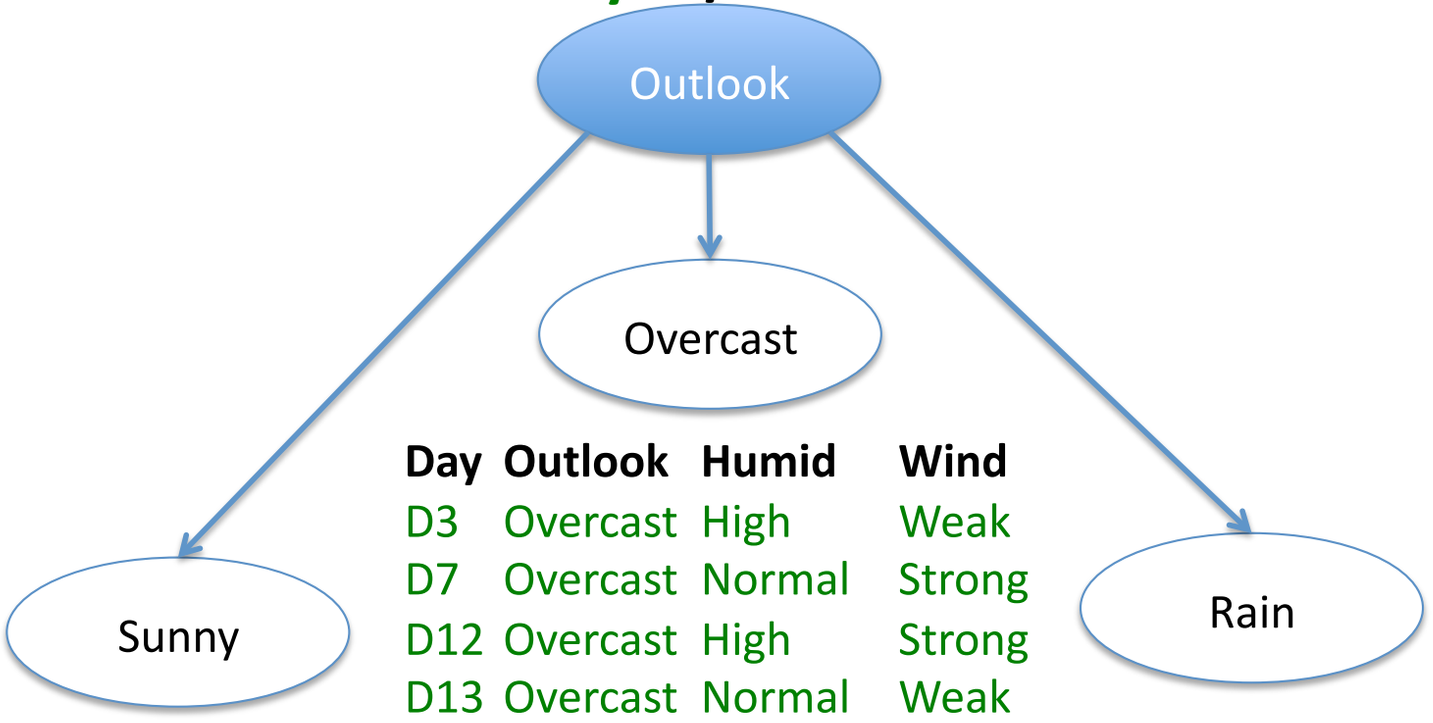
New data:

D15	Rain	High	Weak	?

Example and figures by Victor Lavrenko

Decision Tree Example (2)

9 yes / 5 no



Day	Outlook	Humid	Wind
D3	Overcast	High	Weak
D7	Overcast	Normal	Strong
D12	Overcast	High	Strong
D13	Overcast	Normal	Weak

4 yes / 0 no

pure subset

Day	Outlook	Humid	Wind
D1	Sunny	High	Weak
D2	Sunny	High	Strong
D8	Sunny	High	Weak
D9	Sunny	Normal	Weak
D11	Sunny	Normal	Strong

2 yes / 3 no

split further

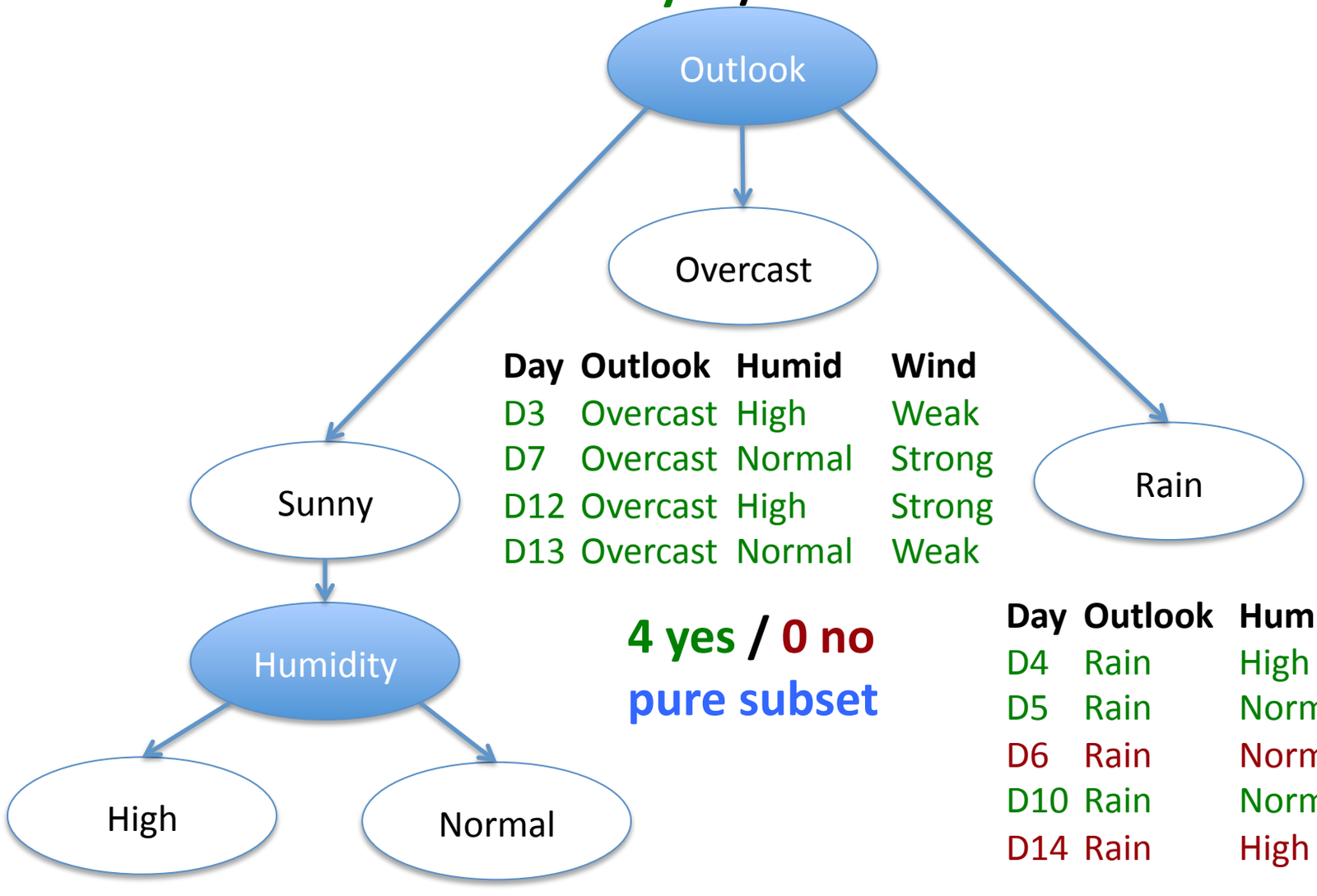
Day	Outlook	Humid	Wind
D4	Rain	High	Weak
D5	Rain	Normal	Weak
D6	Rain	Normal	Strong
D10	Rain	Normal	Weak
D14	Rain	High	Strong

3 yes / 2 no

split further

Decision Tree Example (3)

9 yes / 5 no



Day	Outlook	Humid	Wind
D3	Overcast	High	Weak
D7	Overcast	Normal	Strong
D12	Overcast	High	Strong
D13	Overcast	Normal	Weak

4 yes / 0 no

pure subset

Day	Outlook	Humid	Wind
D4	Rain	High	Weak
D5	Rain	Normal	Weak
D6	Rain	Normal	Strong
D10	Rain	Normal	Weak
D14	Rain	High	Strong

3 yes / 2 no

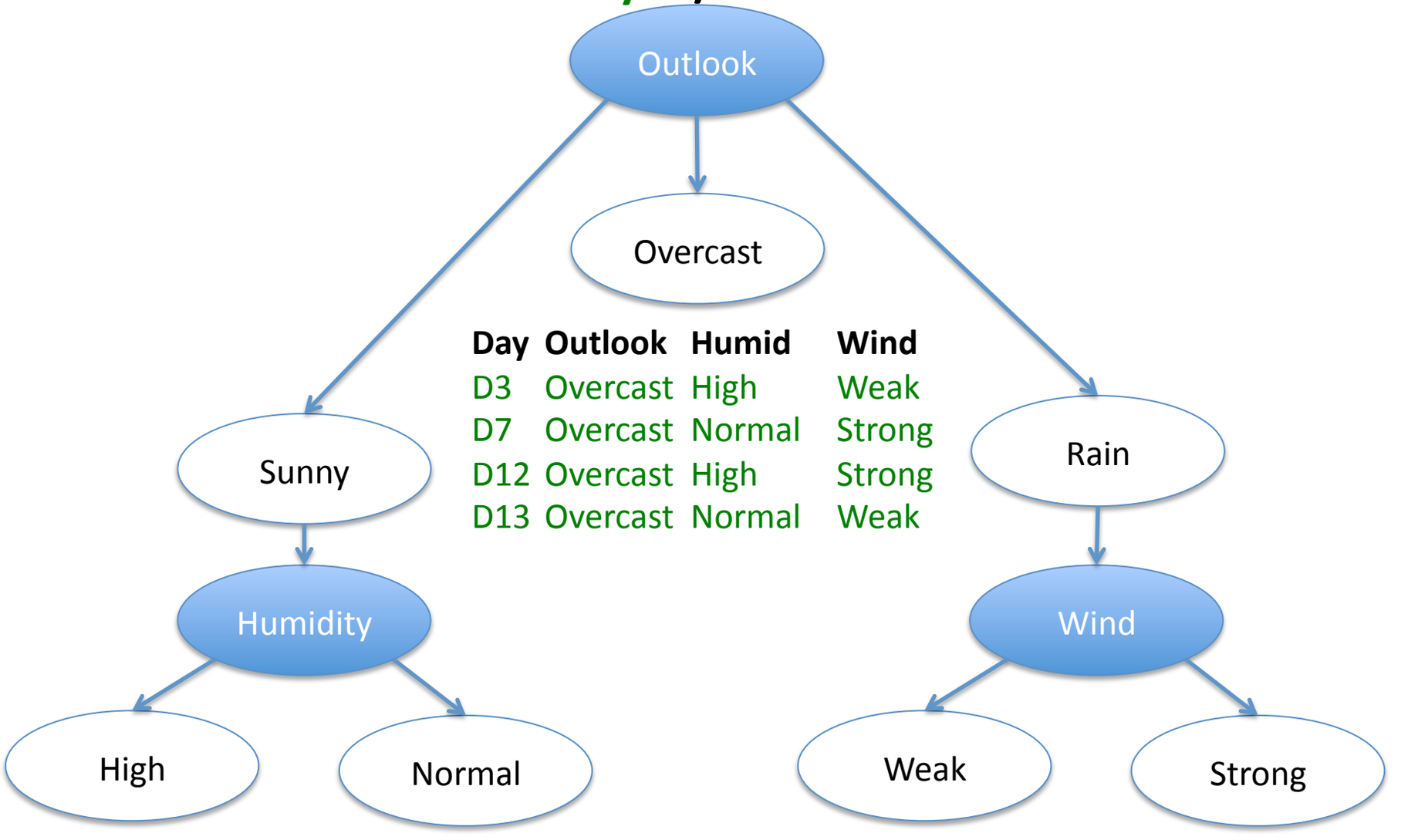
split further

Day	Humid	Wind
D1	High	Weak
D2	High	Strong
D8	High	Weak

Day	Humid	Wind
D9	Normal	Weak
D11	Normal	Strong

Decision Tree Example (4)

9 yes / 5 no



Day	Outlook	Humid	Wind
D3	Overcast	High	Weak
D7	Overcast	Normal	Strong
D12	Overcast	High	Strong
D13	Overcast	Normal	Weak

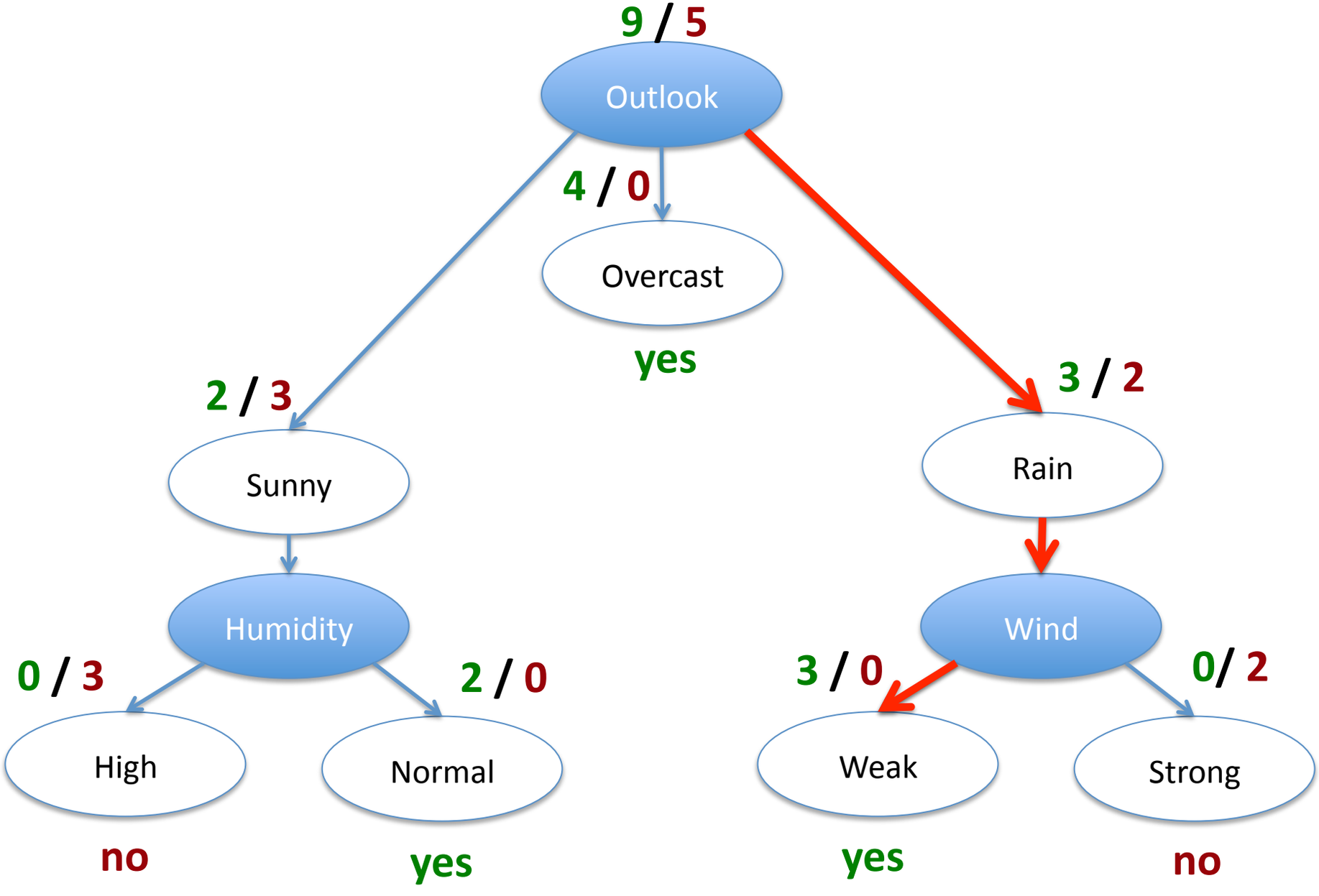
Day	Humid	Wind
D1	High	Weak
D2	High	Strong
D8	High	Weak

Day	Humid	Wind
D9	Normal	Weak
D11	Normal	Strong

Day	Humid	Wind
D4	High	Weak
D5	Normal	Weak
D10	Normal	Weak

Day	Humid	Wind
D6	Normal	Strong
D14	High	Strong

Decision Tree Example (5)



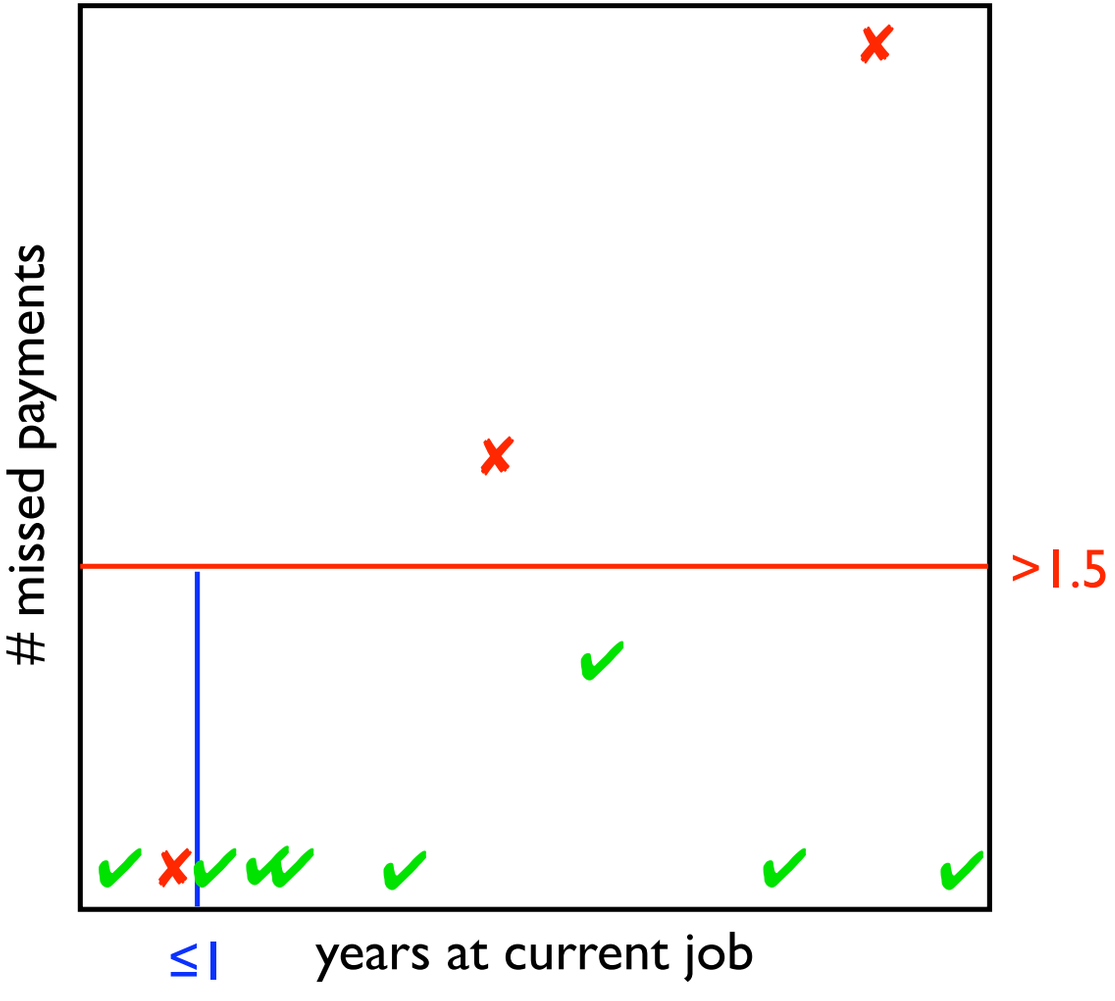
New data:

Day	Outlook	Humid	Wind	
D15	Rain	High	Weak	→ Yes

Continuous Inputs

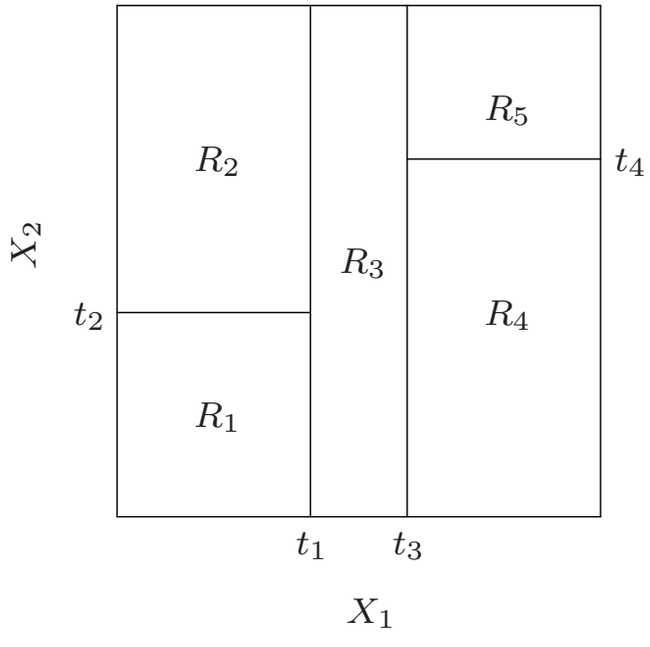
Predicting credit risk

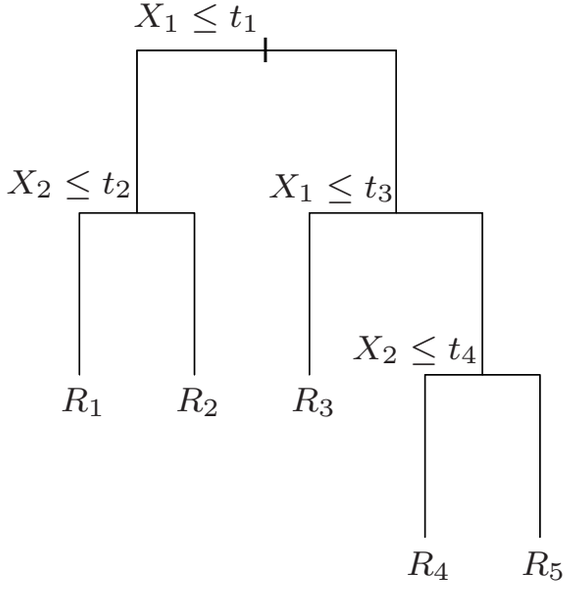
years at current job	# missed payments	defaulted?
7	0	N
0.75	0	Y
3	0	N
9	0	N
4	2	Y
0.25	0	N
5	1	N
8	4	Y
1.0	0	N
1.75	0	N

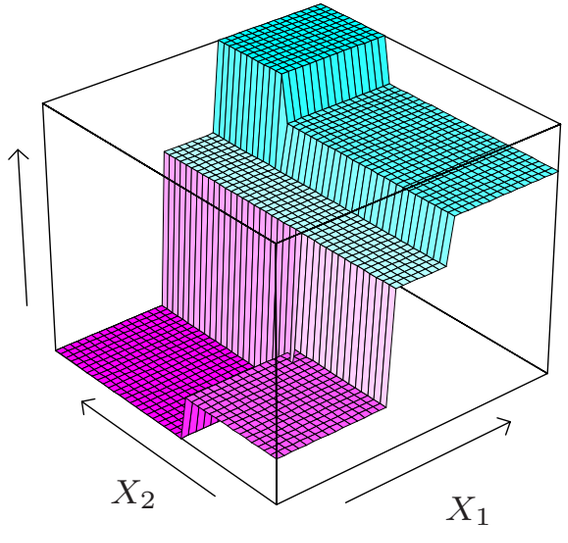


Example and figure by Michael S. Lewicki

Regression Trees







Hastie et al.: *The Elements of Statistical Learning*, 2009

Regression Trees (contd.)

- Training set: $\mathcal{T} = \{(\boldsymbol{x}_i, y_i) \mid i = 1, \ldots, m\}$, $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$
- Input space split into regions defined in leaves: $R_r,\ r \in \{1, \ldots, M\}$
- We can model *region responses* by constants $c_r,\ r \in \{1, \ldots, M\}$ but other possibilities, e.g., linear regression are possible
- Prediction:

$$h(\boldsymbol{x}) = \sum_{r=1}^{M} c_r \mathbb{I}\{\boldsymbol{x} \in R_r\}$$

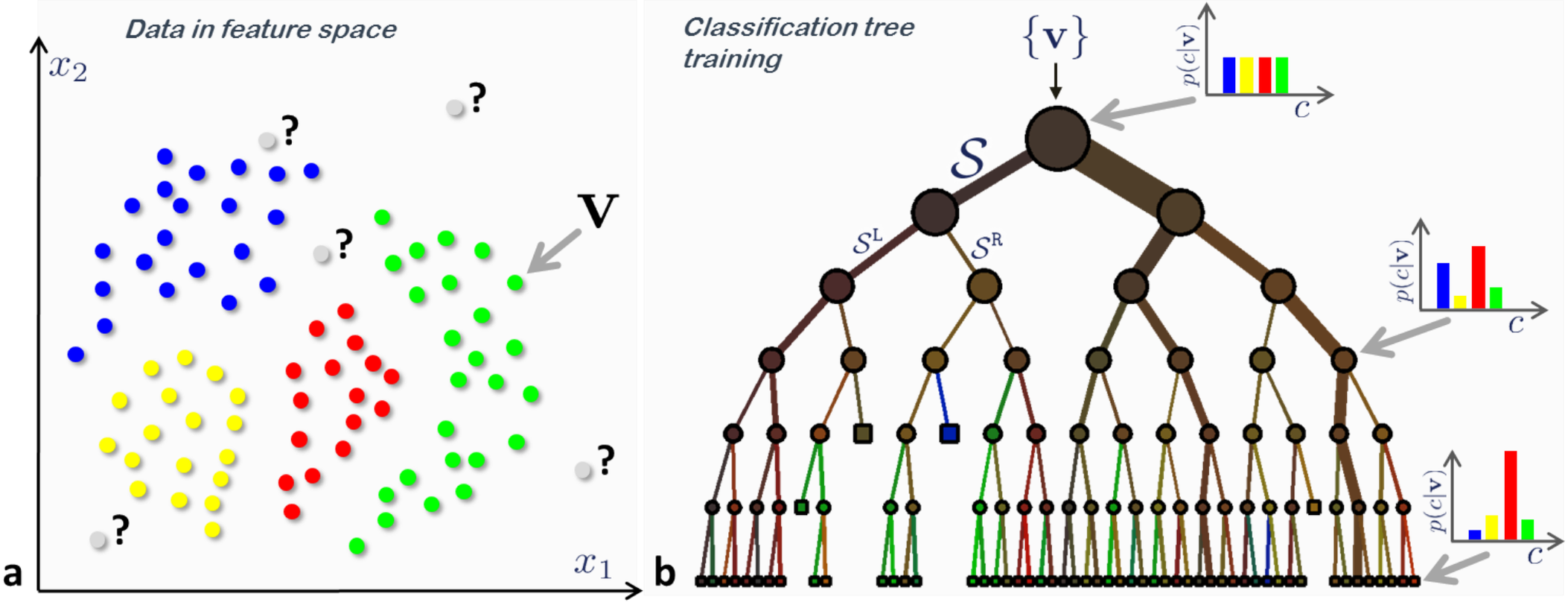
- For sum of squares *loss function* $\sum_{i=1}^{m}(y_i - h(\boldsymbol{x}_i))^2$ we set the responses to be the averages over regions:

$$\hat{c}_r = \frac{1}{|S_r|} \sum_{\boldsymbol{x}_i \in R_r} y_i \qquad \text{(see seminar)}$$

where we define samples per region sets :
$S_r = \{(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{T} \wedge \boldsymbol{x}_i \in R_r\}$

Output Empirical Distribution

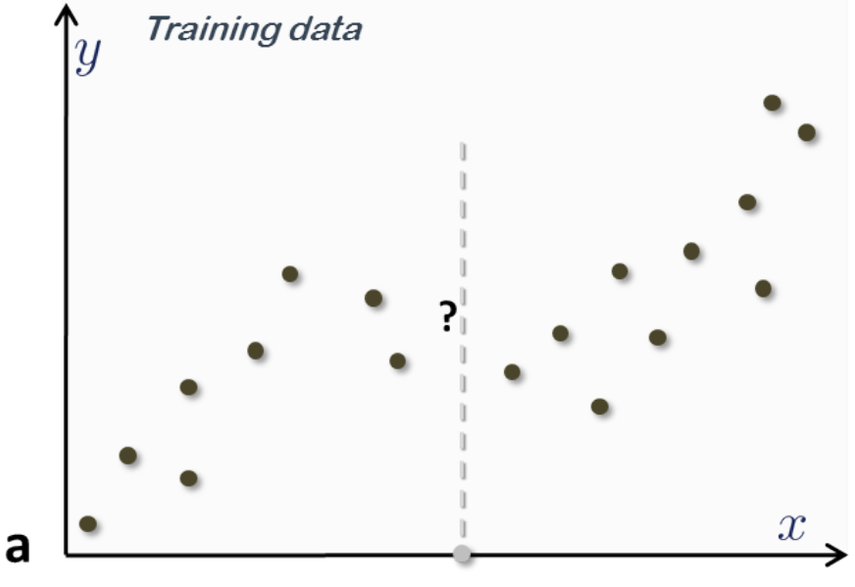
- We can output whole distribution instead of just the prevalent class

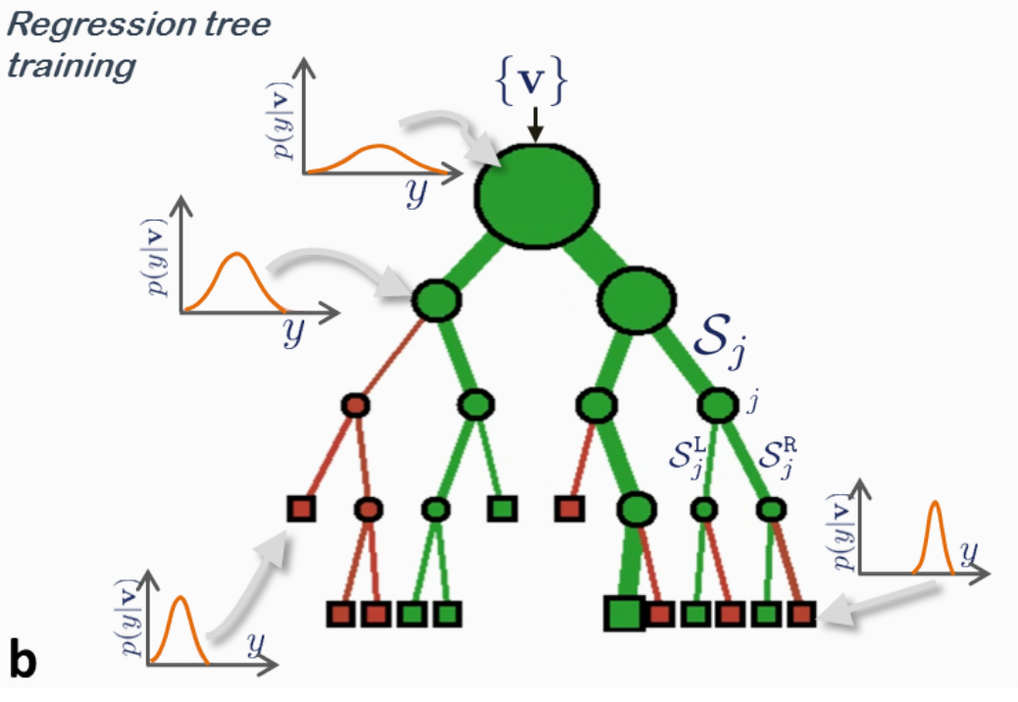


Criminisi et al.: *Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning* , 2011

Continuous Output Empirical Distribution

- Same can be applied for continuous outputs





Criminisi et al.: *Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning* , 2011

Number of Splits

- Number of splits $=$ branching factor B
- Many decision tree training algorithms use binary trees ($B = 2$ for all internal nodes)
  - any tree using $B > 2$ can be transformed into a binary tree
  - easier decision of what to split (see in a moment)
  - multiway splits may fragment data too early leaving insufficient data at the next level
- $\Rightarrow$ we consider binary trees in the following

Why Greedy Learning?

- How many distinct decision trees with n Boolean attributes for binary classification?
  - at least as many as boolean functions of n attributes
  - $=$ number of distinct truth tables with 2^n rows: 2^{2^n}
  - For 6 Boolean attributes at least 18,446,744,073,709,551,616 trees!
- Learning is NP-complete: [Hyafil and Rivest 1976]
- $\Rightarrow$ we need heuristics $\Rightarrow$ **greedy approach**
- Recursively choose the "most important" attribute to find a small tree consistent with the training data
- Split points:
  - **nominal attribute**: try all possibilities
  - **ordinal/continuous attribute**: try attribute values based on all training data samples or their subset

Regression Trees: Which Attribute to Split?

- The "most important" attribute for regression trees would be the one which will reduce the loss (sum of squared errors) by the greatest amount
- We have:

$$h(\boldsymbol{x}) = \sum_{r=1}^{R} c_r \mathbb{I}\{\boldsymbol{x} \in R_r\}$$

- Consider splitting attribute j and split point s, we split an original region R into a pair of half-planes for an ordinal (e.g., continuous) attribute:

$$R_L(j,s) = \{\boldsymbol{x} | \boldsymbol{x} \in R \wedge x_j \leq s\} \text{ and } R_R(j,s) = \{\boldsymbol{x} | \boldsymbol{x} \in R \wedge x_j > s\}$$

similarly for a nominal attribute:

$$R_L(j,s) = \{\boldsymbol{x} | \boldsymbol{x} \in R \wedge x_j = s\} \text{ and } R_R(j,s) = \{\boldsymbol{x} | \boldsymbol{x} \in R \wedge x_j \neq s\}$$

Regression Trees: Which Attribute to Split? (contd.)

- We seek for an attribute j and a split point s which minimize:

$$\min_{c_L} \sum_{\boldsymbol{x}_i \in R_L(j,s)} (y_i - c_L)^2 + \min_{c_R} \sum_{\boldsymbol{x}_i \in R_R(j,s)} (y_i - c_R)^2$$

for $(\boldsymbol{x}_i, y_i) \in S \subseteq \mathcal{T}$ ($S = \mathcal{T}$ for the root node)

- Inner minimizations (region response values) are solved by averaging tree outputs per region:

$$\hat{c}_L = \frac{1}{|S_L(j,s)|} \sum_{\boldsymbol{x}_i \in R_L(j,s)} y_i \qquad \text{and} \qquad \hat{c}_R = \frac{1}{|S_R(j,s)|} \sum_{\boldsymbol{x}_i \in R_R(j,s)} y_i$$

where $S_k(j,s) = \{(\boldsymbol{x}_i, y_i) \mid (\boldsymbol{x}_i, y_i) \in \mathcal{T} \wedge \boldsymbol{x}_i \in R_k(j,s)\}$

Entropy

- Measure of unpredictability used by information theory
- Lossless compression $\Rightarrow$ compressed information has more entropy per character
- Entropy of a random variable Y with possible values $\{y_1, y_2, \ldots, y_n\}$:

$$H(Y) = -\sum_{i=1}^{n} \mathbb{P}(Y = y_i) \log_2 \mathbb{P}(Y = y_i)$$

- Tossing a fair coin:

$$\begin{aligned} H(Y) &= -\mathbb{P}(\text{head}) \log_2 \mathbb{P}(\text{head}) - \mathbb{P}(\text{tail}) \log_2 \mathbb{P}(\text{tail}) \\ &= -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1 \text{ bit} \end{aligned}$$

- Two-heads coin: $H(Y) = 0$ bits

Classification Entropy

- We can use entropy as an **impurity measure**
- $\mathbb{P}(\text{def} = \text{Y}) = \frac{3}{10}$
- $\mathbb{P}(\text{def} = \text{N}) = \frac{7}{10}$
- Entropy:

$$H(\text{def}) = -\sum_{y \in \{\text{Y},\text{N}\}}^{n} \mathbb{P}(\text{def} = y) \log_2 \mathbb{P}(\text{def} = y) =$$

$$= -\frac{3}{10} \log_2 \frac{3}{10} - \frac{7}{10} \log_2 \frac{7}{10} \approx 0.8813$$

- We get zero entropy for a pure dataset

Predicting credit risk

<2 years at current job?	missed payments?	defaulted?
N	N	N
Y	N	Y
N	N	N
N	N	N
N	Y	Y
Y	N	N
N	Y	N
N	Y	Y
Y	N	N
Y	N	N

Example and figure by Michael S. Lewicki

Conditional Entropy

- Conditional entropy is the amount of uncertainty remaining about Y after X is known
- We first define the *specific conditional entropy*:

$$H(Y|X=x) = -\sum_y \mathbb{P}(Y=y|X=x) \log \mathbb{P}(Y=y|X=x)$$

- The *conditional entropy* is then:

$$H(Y|X) = \mathbb{E}_x(H(Y|X=x)) = \sum_x \mathbb{P}(X=x)\ H(Y|X=x)$$

Mutual Information (Information Gain)

- Mutual information is a symmetric measure:

$$I(Y;X) = \sum_y \sum_x \mathbb{P}(X=x, Y=y) \log \left(\frac{\mathbb{P}(X=x, Y=y)}{\mathbb{P}(X=x)\mathbb{P}(Y=y)} \right)$$
$$= H(X) - H(X|Y) = H(Y) - H(Y|X)$$

- It quantifies an information gain for a random variable when other random variable gets involved

Maximizing Information Gain

- Consider the splitting attribute j and the split point s, we get a pair of half-planes $R_L(j,s)$ and $R_R(j,s)$
- We seek for j and s maximizing the information gain:

$$I_s(Y;X_j) = H(Y) - H_s(Y|X_j)$$

where for ordinal attributes we have:

$$H_s(Y|X_j) = \mathbb{P}(X_j \leq s)\ H(Y|X_j \leq s) + \mathbb{P}(X_j > s)\ H(Y|X_j > s)$$

while for the nominal attributes:

$$H_s(Y|X_j) = \mathbb{P}(X_j = s)\ H(Y|X_j = s) + \mathbb{P}(X_j \neq s)\ H(Y|X_j \neq s)$$

Decision Tree Learning Algorithm

BUILD-TREE(S)

1 $i = \text{IMPURITY}(S)$ // $H(Y)$ on S
2 $\hat{g}, \hat{j}, \hat{s}, \hat{S}_L, \hat{S}_R = 0, 0, 0, \emptyset, \emptyset$ // current best kept in these
3 **for** $j \in \{1, \ldots, p\}$ // iterate over attributes $X_1, X_2, \ldots, X_p$
4 **for** $s \in \text{SPLIT-POINTS}(S, j)$ // iterate over all split points of X_j in S
5 $S_L, S_R = \text{SPLIT}(S, j, s)$ // $S_L = \{(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in S \wedge x_{ij} = s\}$
6 $i_L = \text{IMPURITY}(S_L)$ // $H(Y|X_j = s)$
7 $i_R = \text{IMPURITY}(S_R)$ // $H(Y|X_j \neq s)$
8 $g = i - \frac{|S_L|}{|S|} i_L - \frac{|S_R|}{|S|} i_R$ // $I_s(Y; X_j)$
9 **if** $g > \hat{g}$ and $|S_L| > 0$ and $|S_R| > 0$
10 $\hat{g}, \hat{j}, \hat{s}, \hat{S}_L, \hat{S}_R = g, j, s, S_L, S_R$
11 **if** $\hat{g} > 0$
12 $N_L = \text{BUILD-TREE}(\hat{S}_L)$
13 $N_R = \text{BUILD-TREE}(\hat{S}_R)$
14 **return** $\text{DECISION-NODE}(\hat{j}, \hat{s}, N_L, N_R)$
15 **else return** $\text{LEAF-NODE}(S)$

Maximizing Information Gain Example

- $H(\text{def}|\text{<2yrs} = \text{Y}) = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} \approx 0.8113$
- $H(\text{def}|\text{<2yrs} = \text{N}) = -\frac{2}{6}\log_2\frac{2}{6} - \frac{4}{6}\log_2\frac{4}{6} \approx 0.9183$
- $H(\text{def}|\text{<2yrs}) \approx \frac{4}{10} \times 0.8113 + \frac{6}{10} \times 0.9183 \approx 0.8755$

- $H(\text{def}|\text{miss} = \text{Y}) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} \approx 0.9183$
- $H(\text{def}|\text{miss} = \text{N}) = -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7} \approx 0.5917$
- $H(\text{def}|\text{miss}) \approx \frac{3}{10} \times 0.9183 + \frac{7}{10} \times 0.5917 \approx 0.69$

- $H(\text{def}) - H(\text{def}|\text{<2yrs}) \approx 0.8813 - 0.8755 = 0.0058$
- $H(\text{def}) - H(\text{def}|\text{miss}) \approx 0.8813 - 0.69 = \mathbf{0.1913}$

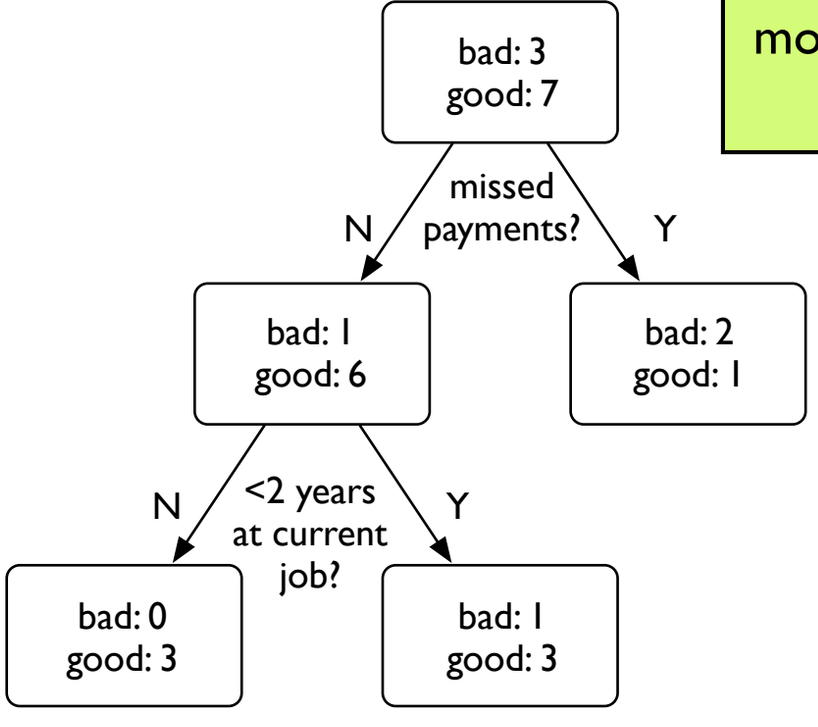
Predicting credit risk

<2 years at current job?	missed payments?	defaulted?
N	N	N
Y	N	Y
N	N	N
N	N	N
N	Y	Y
Y	N	N
N	Y	N
N	Y	Y
Y	N	N
Y	N	N

Example and figure by Michael S. Lewicki

Maximizing Information Gain Example (contd.)

- $I(\text{def}; <2\text{yrs}) = H(\text{def}) - H(\text{def}|<2\text{yrs}) \approx 0.0058$
- $I(\text{def}; \text{miss}) = H(\text{def}) - H(\text{def}|\text{miss}) \approx \mathbf{0.1913}$



Missed payments are the most informative attribute about defaulting.

Predicting credit risk

<2 years at current job?	missed payments?	defaulted?
N	N	N
Y	N	Y
N	N	N
N	N	N
N	Y	Y
Y	N	N
N	Y	N
N	Y	Y
Y	N	N
Y	N	N

Example and figures by Michael S. Lewicki

Information Gain for Multiway Splits

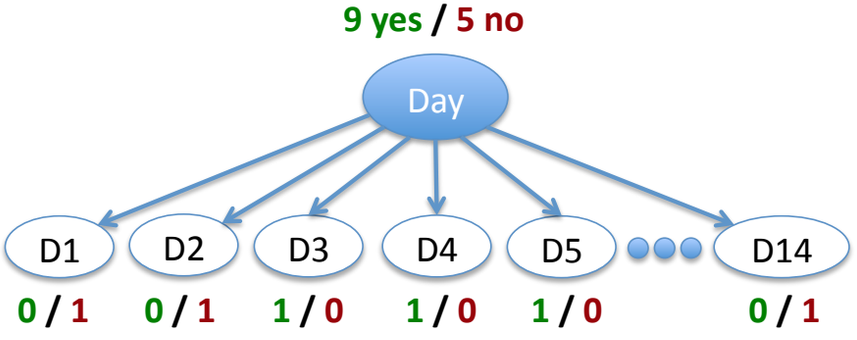
- For multiway splits we have:

$$H(Y|X_j) = \sum_{s \in \text{Values}(X_j)} \mathbb{P}(X_j = s)\, H(Y|X_j = s)$$

- Biases towards attributes with many values!
- Extreme case: sample ID (day)

  $H(\text{play}|\text{day}) = 0$
- Maximizes information gain:

  $I(\text{play};\text{day}) = H(\text{play}) - H(\text{play}|\text{day}) = H(\text{play})$



all subsets perfectly pure => optimal split

Training examples: **9 yes / 5 no**

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

New data:

D15	Rain	High	Weak	?

Example and figures by Victor Lavrenko

Multiway Splits: Information Gain Ratio

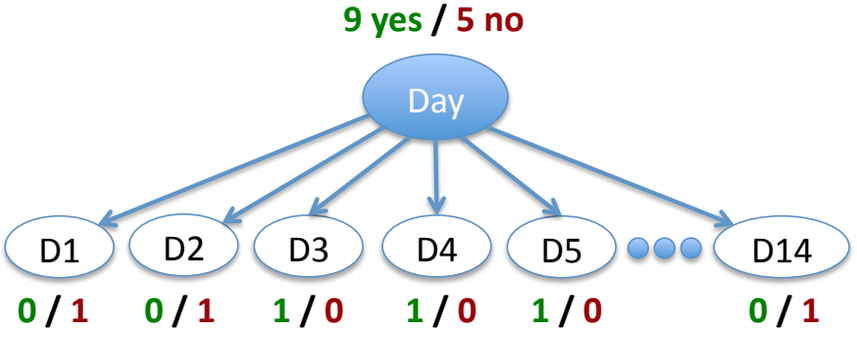
- Use *information gain ratio* instead:

$$GainRatio(Y; X_j) = \frac{I(Y; X_j)}{SplitEntropy(Y; X_j)}$$

where

$$SplitEntropy(Y; X_j) = \sum_{s \in \text{Values}(X_j)} \frac{|S_s|}{|S|} \log \frac{|S_s|}{|S|}$$

- High $SplitEntropy$: partitions have more or less the same size (uniform)
- Low $SplitEntropy$: few partitions hold most of the tuples (peaks)



all subsets perfectly pure => optimal split

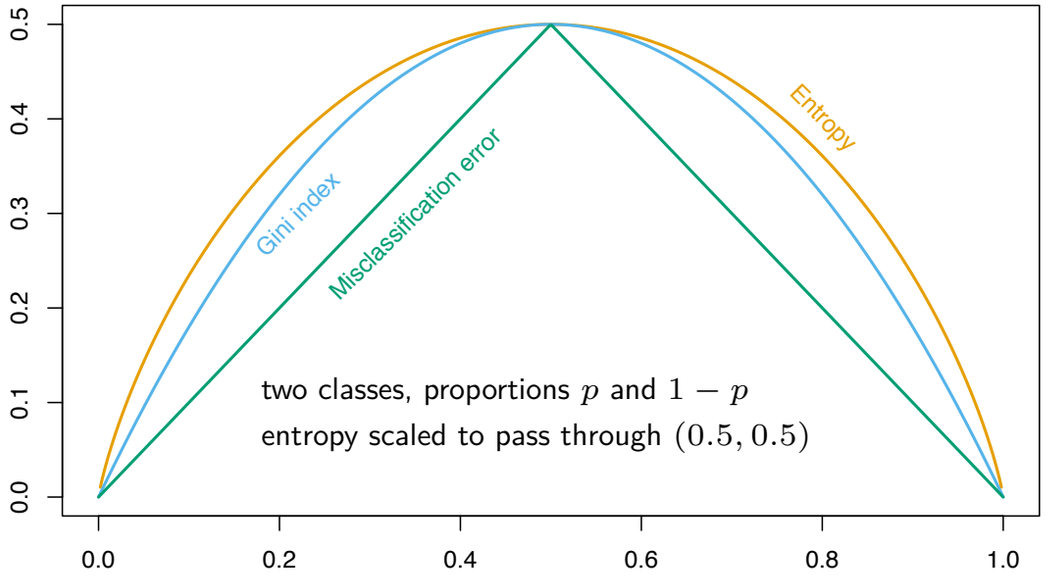
Other Impurity Measures

- *Gini impurity* (expected error rate if the classification is picked according to the class distribution):

$$Gini(Y) = \sum_{i \neq j} \mathbb{P}(Y = y_i)\, \mathbb{P}(Y = y_j) = 1 - \sum_{i} \mathbb{P}(Y = y_i)^2$$

- *Misclassification measure* (minimum probability of misclassification):

$$Mis(Y) = 1 - \max_i \mathbb{P}(Y = y_i)$$



Hastie et al.: *The Elements of Statistical Learning*, 2009

When to Stop Splitting?

- Split while impurity decreases
  - no assurance of zero impurity at leafs (e.g., for two samples $\boldsymbol{x}_i = \boldsymbol{x}_j$, $y_i \neq y_j$)
  - when all leaves are pure then tree becomes a lookup table $\Rightarrow$ **overfitting!**
- Check generalization error using validation set, stop when validation error starts to increase
- Use threshold β: stop splitting when maximum possible gain drops below β
  - uses all training data unlike the previous approach
  - leaves at different depths: adapts to complexity in input distribution
  - drawback: hard to set β

When to Stop Splitting? (contd.)

- Stop when the node represents less than n (e.g., 10) samples or less then a percentage of total samples (e.g., 5%)
- Trade complexity for test accuracy, minimize:

$$\alpha \cdot size + \sum_{l \in leaves} \text{IMPURITY}(S_l)$$

  where $\alpha > 0$ and $size$ can be the number of tree nodes or links

- Check statistical significance of the impurity reduction, e.g. using chi-squared test:
  - when a candidate split does not reduce the impurity *significantly*, splitting is stopped
  - does a candidate split significantly differ from a random split?

Pruning

- The previously stopping methods may stop tree growth prematurely due to the greedy approach
- Pruning: reduce a fully grown tree starting at leaves
- All pairs of sibling leaf nodes are considered for *merge*
- Any pair whose elimination yields a satisfactory (small) increase in impurity is eliminated
- Computationally costly but preferred for smaller problems
- Rule pruning: simplifying rules defined by conjunction of tests on a way from the root to leaves $\Rightarrow$ better interpretability

Decision Tree Methods

- CART (Classification And Regression Trees): described in previous slides (some extensions beyond were shown)
- ID3 (Interactive Dichotomizer 3)
  - Quinlan: *Induction of Decision Trees*, 1986
  - nominal (unordered inputs), uses binning for continuous variables
  - multiway
  - depth $\leq$ number of input variables
  - no pruning originally
- C4.5 (Quinlan)
  - multiway for nominal data
  - pruning based on statistical significance tests
  - missing features different that CART p. 412
  - rule-based prunning
- C5.0 (Quinlan): patented, faster, less memory, boosting support
- CHAID (CHi-square Automatic Interaction Detector)

Bias-Variance Decomposition

- Consider a regression problem with data generated as follows:

$$y = f(x) + \epsilon$$

where ϵ is noise: $\mathbb{E}(\epsilon) = 0$ and $\mathrm{Var}(\epsilon) = \sigma^2$

- Let $h(x; \mathcal{T}, \theta)$ be a model trained on data generated from $p(x, y)$ using algorithm with hyper-parameters θ (e.g., a random seed)
- Such definition allows randomized training algorithms generating different models for the same $\mathcal{T}$
- To assess performance of the particular learning algorithm on $\mathcal{T}$ and θ we can evaluate expected value of a square loss for samples from $p(x, y)$:

$$\mathrm{Err}_{\mathcal{T},\theta}(x) = \mathbb{E}_{y|x}\left([y - h(x; \mathcal{T}, \theta)]^2 \mid \mathcal{T}, \theta\right)$$

- Expected test error at x is then:

$$\mathrm{Err(x)} = \mathbb{E}_{\mathcal{T},\theta}(\mathrm{Err}_{\mathcal{T},\theta}(x)) = \mathbb{E}\left([y - h(x)]^2\right)$$

Bias-Variance Decomposition (contd.)

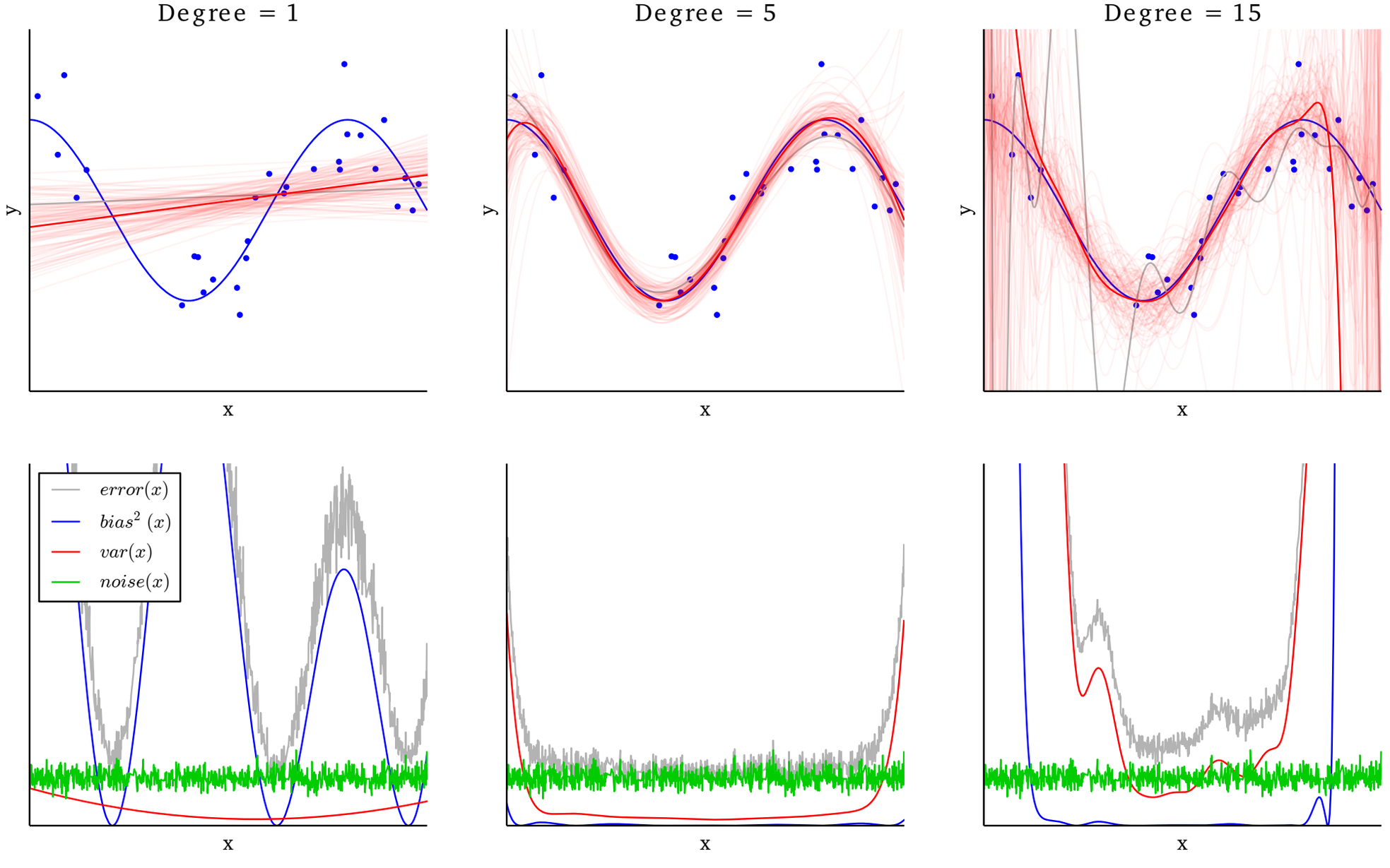
- Use $\mathbb{E}(y) = \mathbb{E}(f(x)) = f(x)$, $\mathrm{Var}(y) = \sigma^2$ and $\mathbb{E}(yh(x)) = \mathbb{E}((f(x) + \epsilon)h(x)) = f(x)\mathbb{E}(h(x)) = \mathbb{E}(y)\mathbb{E}(h(x))$:

$$
\begin{aligned}
\mathrm{Err}(x) &= \mathbb{E}\left([y - h(x)]^2\right) = \mathbb{E}(y^2) - 2\mathbb{E}(yh(x)) + \mathbb{E}(h(x)^2) = \\
&= \mathrm{Var}(y) + \mathbb{E}(y)^2 - 2\mathbb{E}(y)\mathbb{E}(h(x)) + \mathrm{Var}(h(x)) + \mathbb{E}(h(x))^2 = \\
&= \sigma^2 + \left[\mathbb{E}(y) - \mathbb{E}(h(x))\right]^2 + \mathrm{Var}(h(x)) = \\
&= \underbrace{\sigma^2}_{\mathrm{noise}(x)} + \underbrace{\mathbb{E}(f(x) - h(x))^2}_{\mathrm{bias}^2(x)} + \underbrace{\mathrm{Var}(h(x))}_{\mathrm{var(x)}}
\end{aligned}
$$

- The error splits into three terms
  - **noise**(x): irreducible determined by data,
  - **bias**$^2(x)$: error of approximation
  - **var**(x): measures sensitivity to particular dataset
- We have to find the right balance to minimize the loss
  $\Rightarrow$ **bias-variance tradeoff**

Bias-Variance: Model Complexity

- Polynomial regression with a varying degree of polynomial



Gilles Louppe: *Understanding Random Forests: From Theory to Practice*, 2014

Bias and Variance of Decision Trees

- Small changes of training data lead to big differences in final trees
- Decision trees grown deep enough have typically:
  - low bias
  - high variance

  $\Rightarrow$ **overfitting**
- Idea: *average multiple models* to reduce variance while (happily) not increasing bias much

Averaging Models

- Define model b as an average of M models:

$$b(x) = \frac{1}{M} \sum_{i=1}^{M} h_i(x)$$

- Noise is given by data and does not change
- Bias remains unchanged when compared to a single model:

$$\text{bias}(x) = f(x) - \mathbb{E}(b(x)) = f(x) - \mathbb{E}\left(\frac{1}{M} \sum_{i=1}^{M} h_i(x)\right)$$

$$= f(x) - \frac{1}{M} \sum_{i=1}^{M} \mathbb{E}\left(h_i(x)\right) = f(x) - \mathbb{E}(h(x))$$

- The last step is due to $\mathcal{T}_i$ and θ_i used to train $h_i(x)$ are i.i.d.

Averaging Models: Variance

- For uncorrelated component models $h_i(x)$:

$$\text{var}(x) = \text{Var}(b(x)) = \text{Var}\left(\frac{1}{M}\sum_{i=1}^{M} h_i(x)\right) =$$

$$= \frac{1}{M^2}\sum_{i=1}^{M} \text{Var}(h_i(x)) = \frac{1}{M}\text{Var}(h(x))$$

  which is a great improvement based on the **strong** assumption

- There is no improvement for maximum correlation, i.e., all component models are same ($h_m(x) = h(x)$ for $m = 1, \ldots, M$) we get:

$$\text{var}(x) = \text{Var}\left(\frac{1}{M}\sum_{i=1}^{M} h(x)\right) = \text{Var}(h(x))$$

  $\Rightarrow$ we need to train **uncorrelated** (diverse) component models while **keeping their bias reasonably low**

Bootstrapping

- In practice we have only a single training dataset $\mathcal{T}$
- Bootstrapping is a method producing datasets $\mathcal{T}_i$ for $i = 1, \ldots M$ by sampling $\mathcal{T}$ uniformly with *replacement*
- Bootstrap datasets have the same size as the original dataset $|\mathcal{T}_i| = |\mathcal{T}|$
- $\mathcal{T}_i$ is expected to have the fraction $1 - \frac{1}{e} \approx 63.2\%$ of unique samples from $\mathcal{T}$, others are duplicates (see seminar)

Bagging

- Bagging = Bootstrap AGGregating [Breiman 1994]:
  1. Use bootstrapping to generate M datasets
  2. Train a model h_i on each dataset $\mathcal{T}_i$
  3. Average the models
- When decision trees are used as the models $\Rightarrow$ **random forests**
- Low bias is achieved by growing the trees to maximal depth
- Trees are decorrelated by:
  - training each tree on a different bootstrap dataset
  - randomization of split attribute selection

Random Forest Algorithm

1. For $i = 1 \ldots M$:
   (a) draw a bootstrap dataset $\mathcal{T}_i$ from $\mathcal{T}$, $|\mathcal{T}_i| = |\mathcal{T}|$
   (b) grow a tree $h_i(x)$ using $\mathcal{T}_i$ by recursively repeating the following until the minimum node size n_{min} is reached:
      i. select k attributes at random from the p attributes
      ii. pick the best variable and split-point among the k
      iii. split the node into two daughter nodes

2. Output ensemble of trees $b(x)$ averaging $h_i(x)$ (regression) or selecting a majority vote (classification)

- Node size $=$ the number of dataset samples associated with the node

Out-of-Bag (OOB) Error

- "Cheap" way of generalization error assessment for bagging
- Bagging produces bootstrapped sets $\mathcal{T}_1, \mathcal{T}_2, \ldots \mathcal{T}_B$
- For each $(x_i, y_i) \in \mathcal{T}$ select only trees which were not trained on this sample: $H_i = \{h_j \mid (x_i, y_i) \notin \mathcal{T}_j\}$
- Average only the OOB trees in H_i when evaluating error for (x_i, y_i)
- Replacement for K-fold cross-validation

Feature Importance

- Random forests allow easy evaluation of feature importances
- Mean Decrease Impurity (MDI):
  - set $f_j = 0$ for all attributes $j = 1, \ldots, p$
  - traverse all trees processing all internal nodes
  - for each node having a split attribute j add its *impurity decrease* multiplied by the proportion of the *node size* to f_j
- Mean Decrease Accuracy (MDA), permutaion importance:
  - evaluate the forest using OOB data
  - do the same with permuted values of an attribute j
  - watch decrease in accuracy: low decrease means unimportant feature

Random Forest Summary

- Easy to use method: robust w.r.t. parameter settings (M, node size)
- While *consistency* is proven for decision trees (both regression and classification) we have only proofs for simplified versions of random forests [Breiman, 1984]
- Related methods: boosted trees

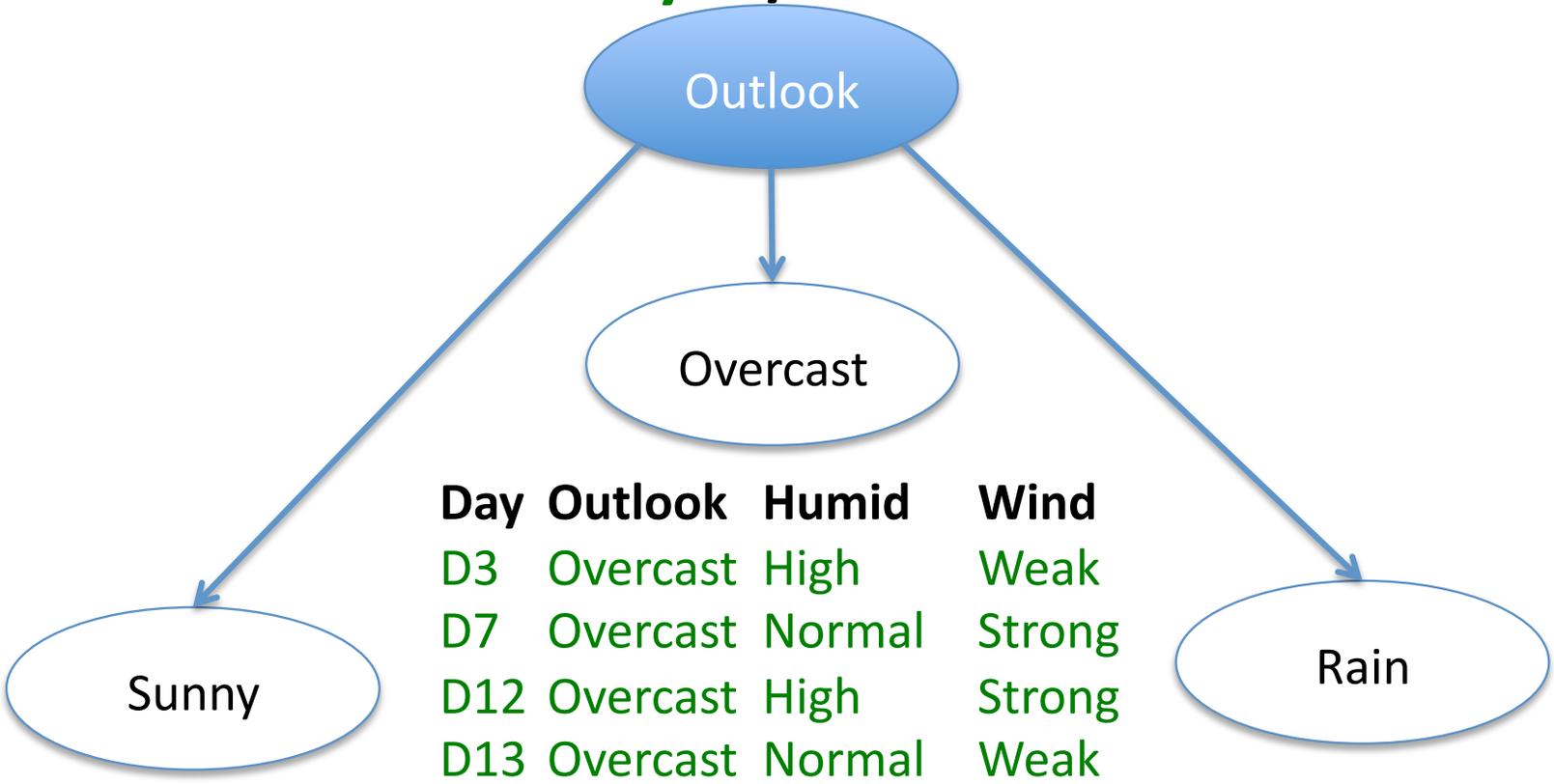
Training examples: **9 yes / 5 no**

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

New data:

D15	Rain	High	Weak	?

9 yes / 5 no



Day	Outlook	Humid	Wind
D3	Overcast	High	Weak
D7	Overcast	Normal	Strong
D12	Overcast	High	Strong
D13	Overcast	Normal	Weak

4 yes / 0 no

pure subset

Day	Outlook	Humid	Wind
D1	Sunny	High	Weak
D2	Sunny	High	Strong
D8	Sunny	High	Weak
D9	Sunny	Normal	Weak
D11	Sunny	Normal	Strong

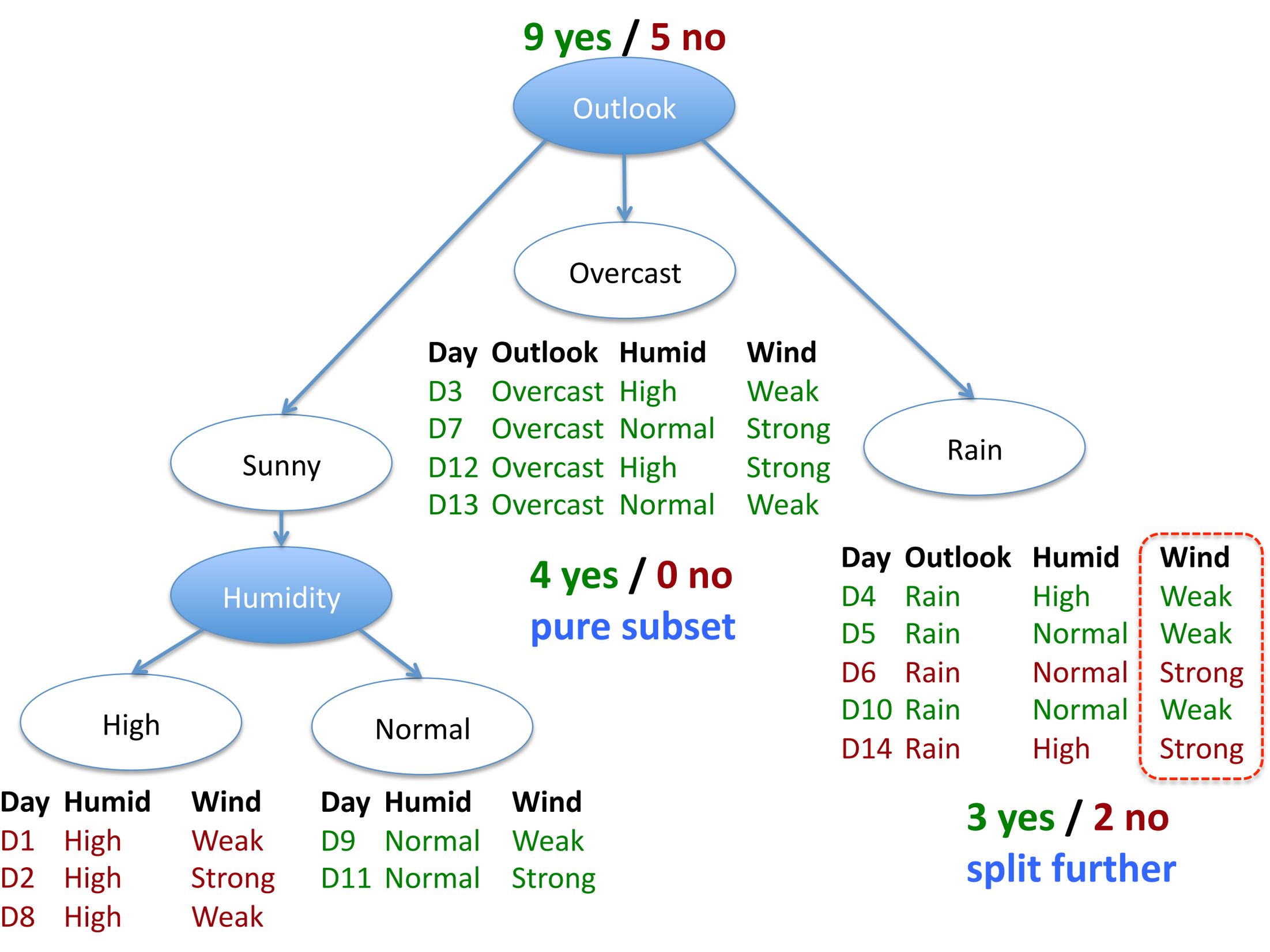
2 yes / 3 no

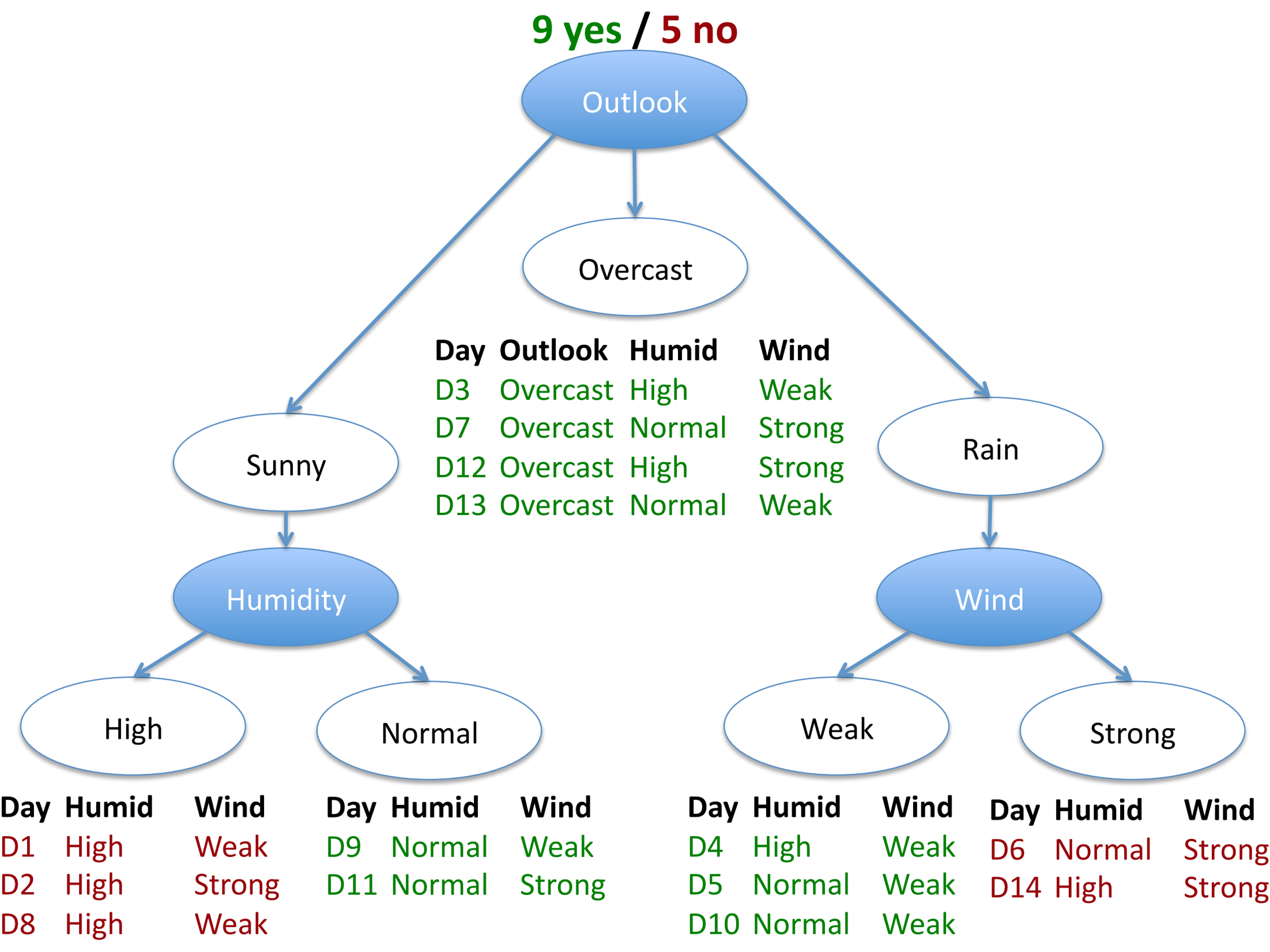
split further

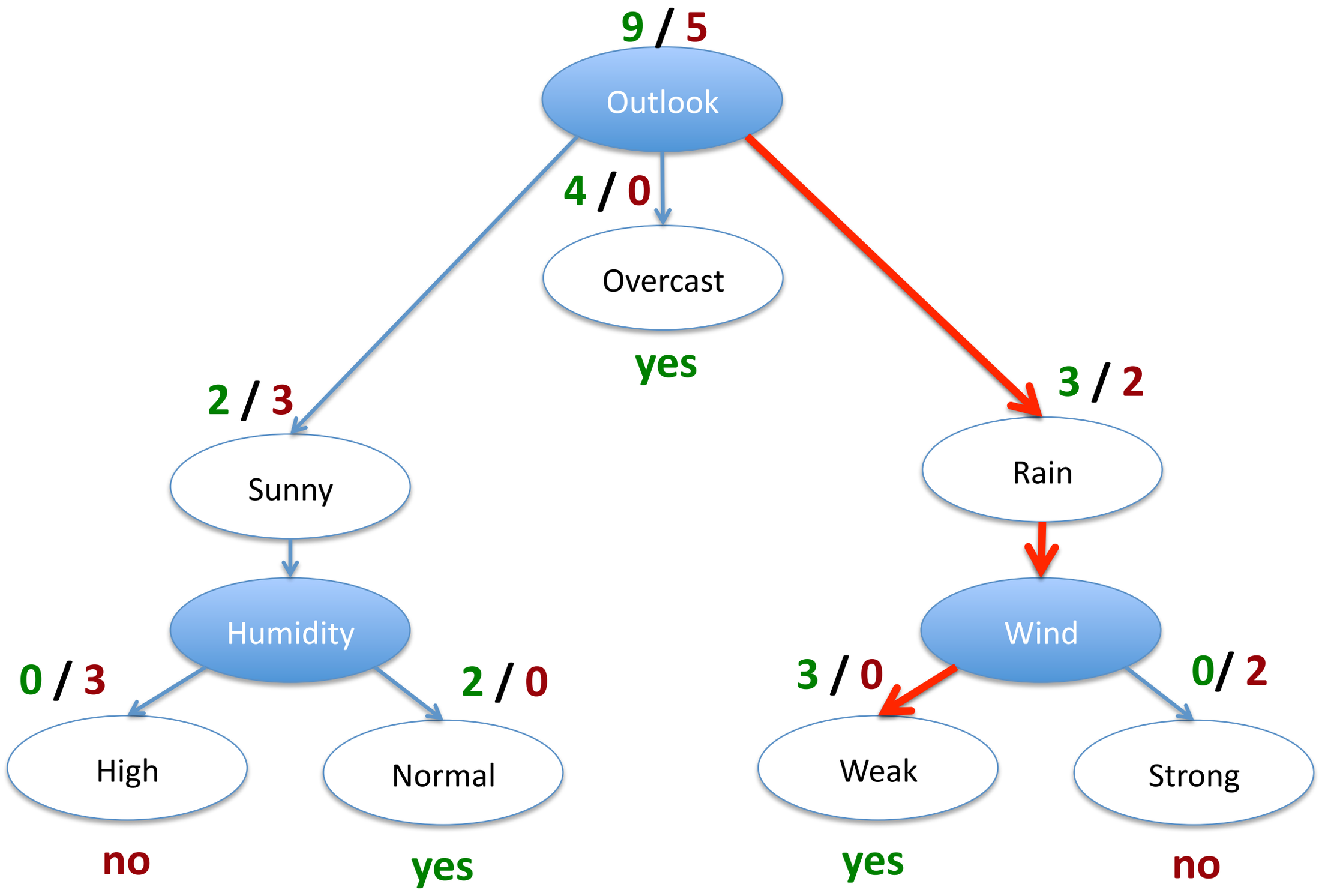
Day	Outlook	Humid	Wind
D4	Rain	High	Weak
D5	Rain	Normal	Weak
D6	Rain	Normal	Strong
D10	Rain	Normal	Weak
D14	Rain	High	Strong

3 yes / 2 no

split further


9 yes / 5 no
Outlook
Overcast
Sunny
Rain
Humidity
High
Normal
Day Outlook Humid Wind
D3 Overcast High Weak
D7 Overcast Normal Strong
D12 Overcast High Strong
D13 Overcast Normal Weak
4 yes / 0 no
pure subset
Day Outlook Humid Wind
D4 Rain High Weak
D5 Rain Normal Weak
D6 Rain Normal Strong
D10 Rain Normal Weak
D14 Rain High Strong
3 yes / 2 no
split further
Day Humid Wind
D1 High Weak
D2 High Strong
D8 High Weak
Day Humid Wind
D9 Normal Weak
D11 Normal Strong


9 yes / 5 no
Outlook
Overcast
Sunny
Rain
Humidity
Wind
High
Normal
Weak
Strong
Day Outlook Humid Wind
D3 Overcast High Weak
D7 Overcast Normal Strong
D12 Overcast High Strong
D13 Overcast Normal Weak
Day Humid Wind
D1 High Weak
D2 High Strong
D8 High Weak
Day Humid Wind
D9 Normal Weak
D11 Normal Strong
Day Humid Wind
D4 High Weak
D5 Normal Weak
D10 Normal Weak
Day Humid Wind
D6 Normal Strong
D14 High Strong

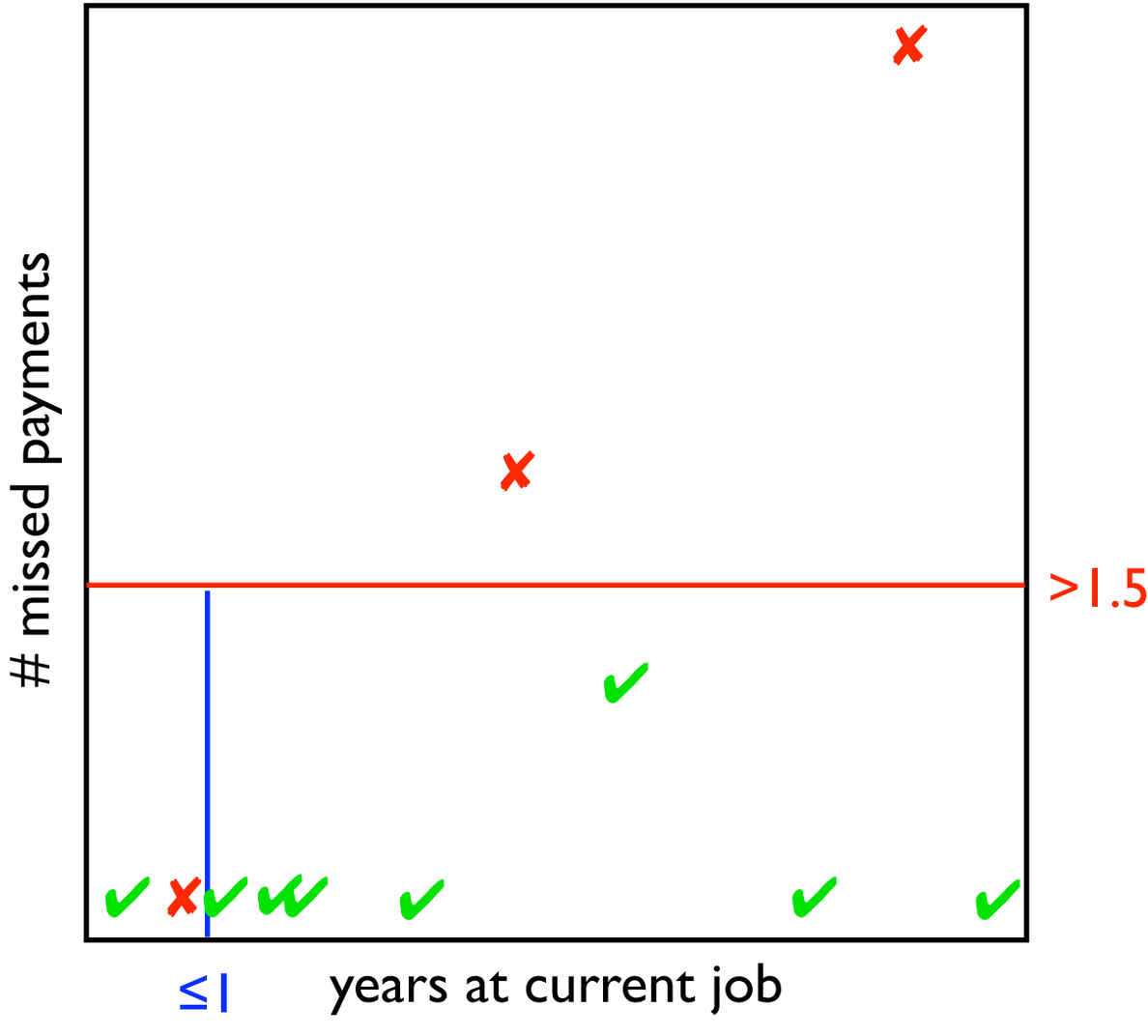


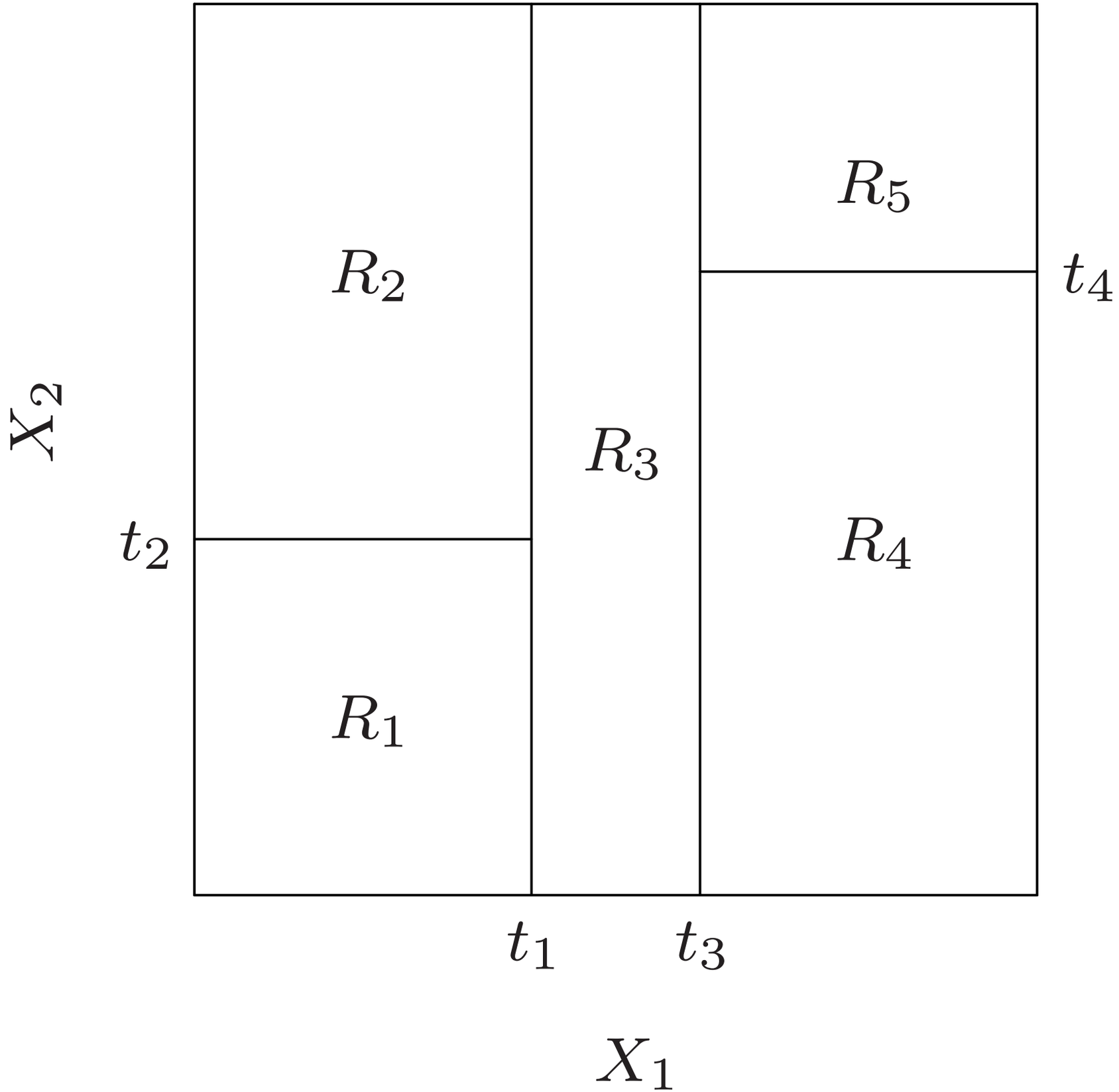
New data:

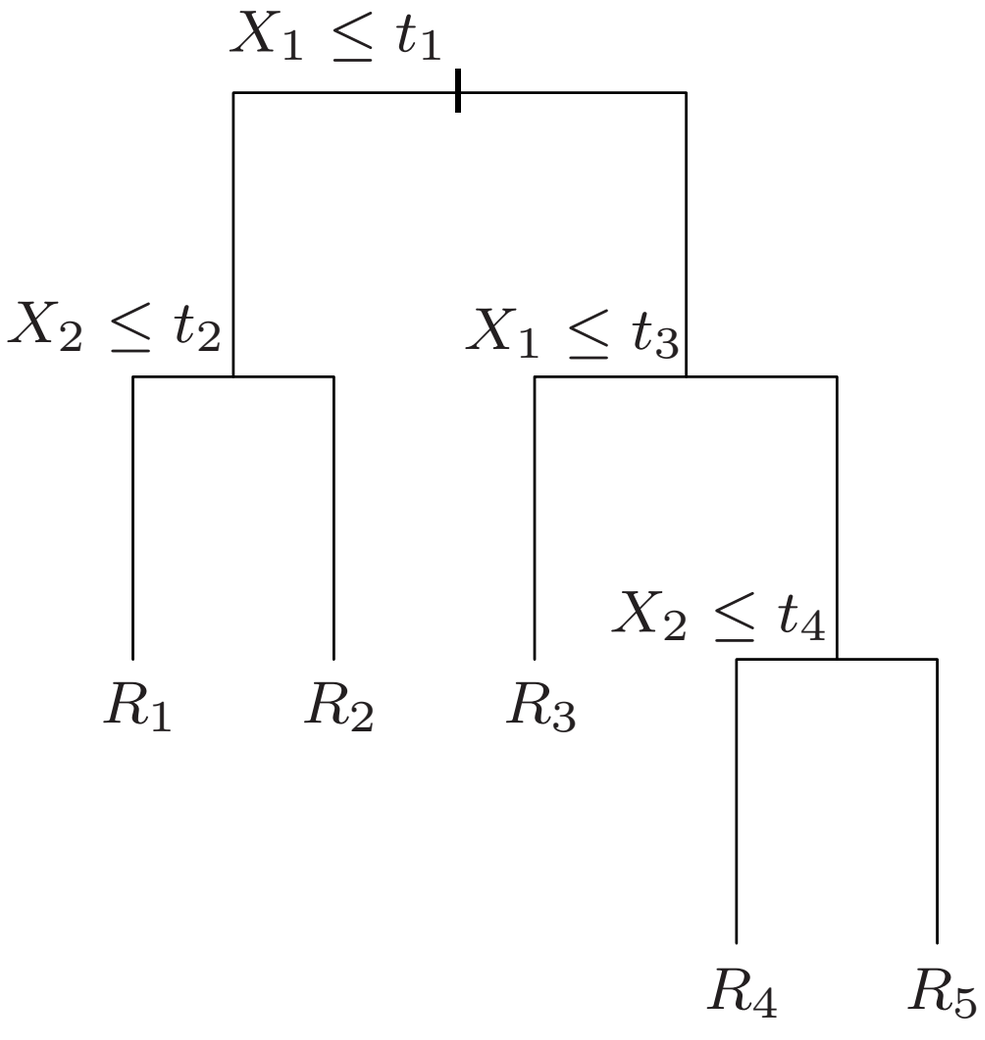
Day	Outlook	Humid	Wind	
D15	Rain	High	Weak	→ Yes

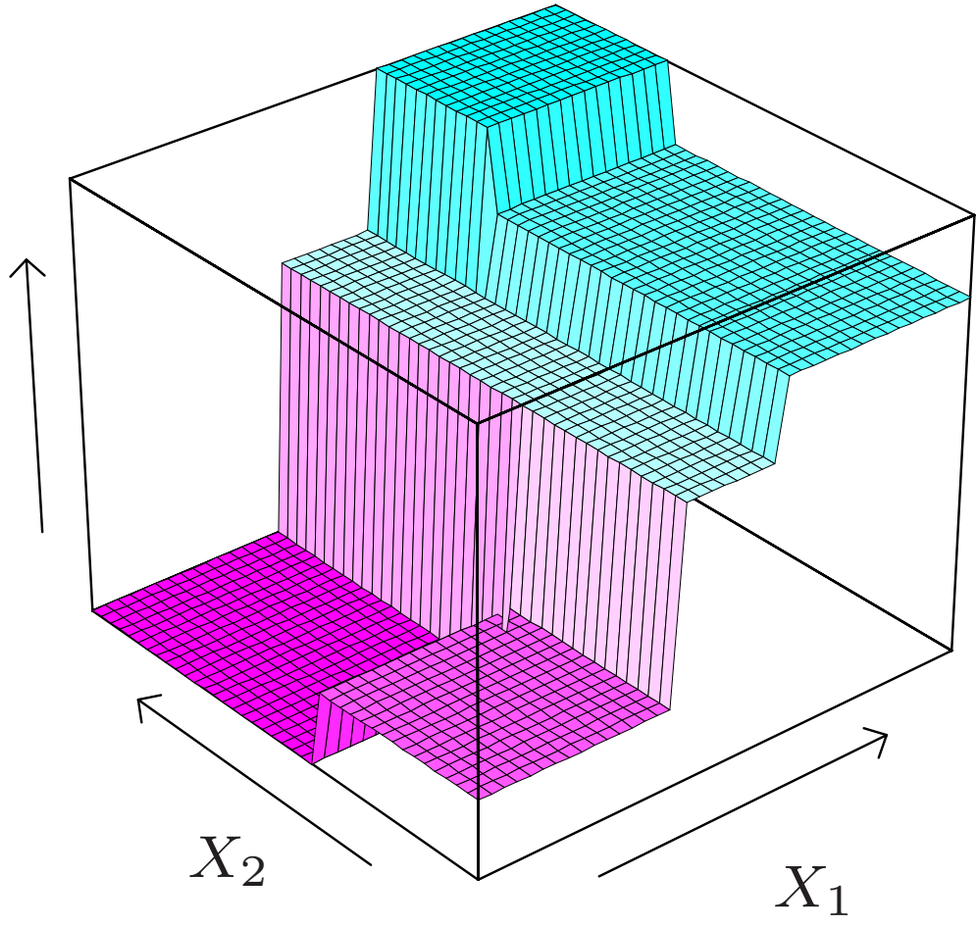
Predicting credit risk

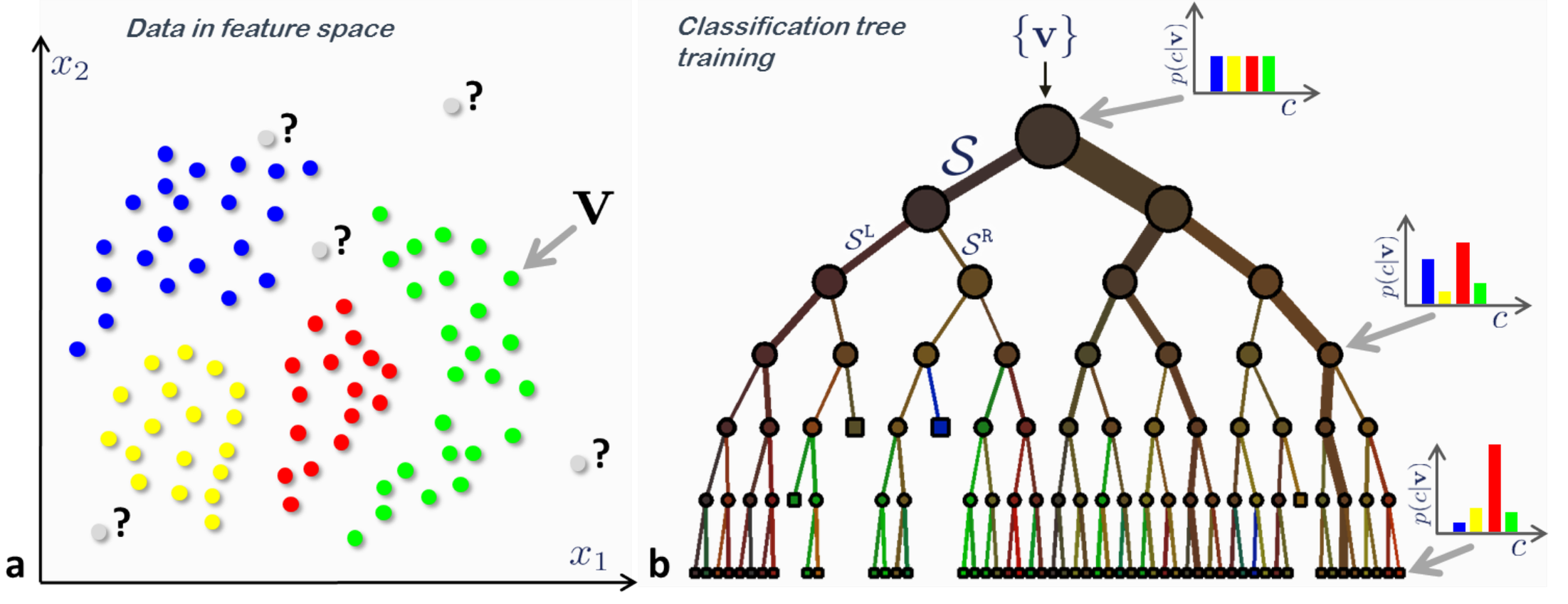
years at current job	# missed payments	defaulted?
7	0	N
0.75	0	Y
3	0	N
9	0	N
4	2	Y
0.25	0	N
5	1	N
8	4	Y
1.0	0	N
1.75	0	N

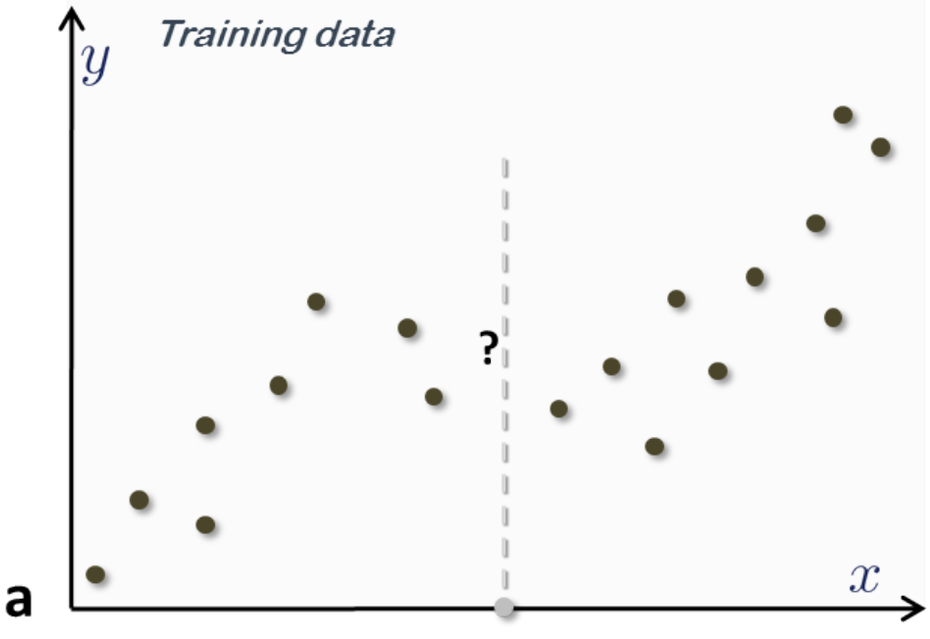



R_5
R_2
t_4
X_2
R_3
t_2
R_4
R_1
t_1
t_3
X_1

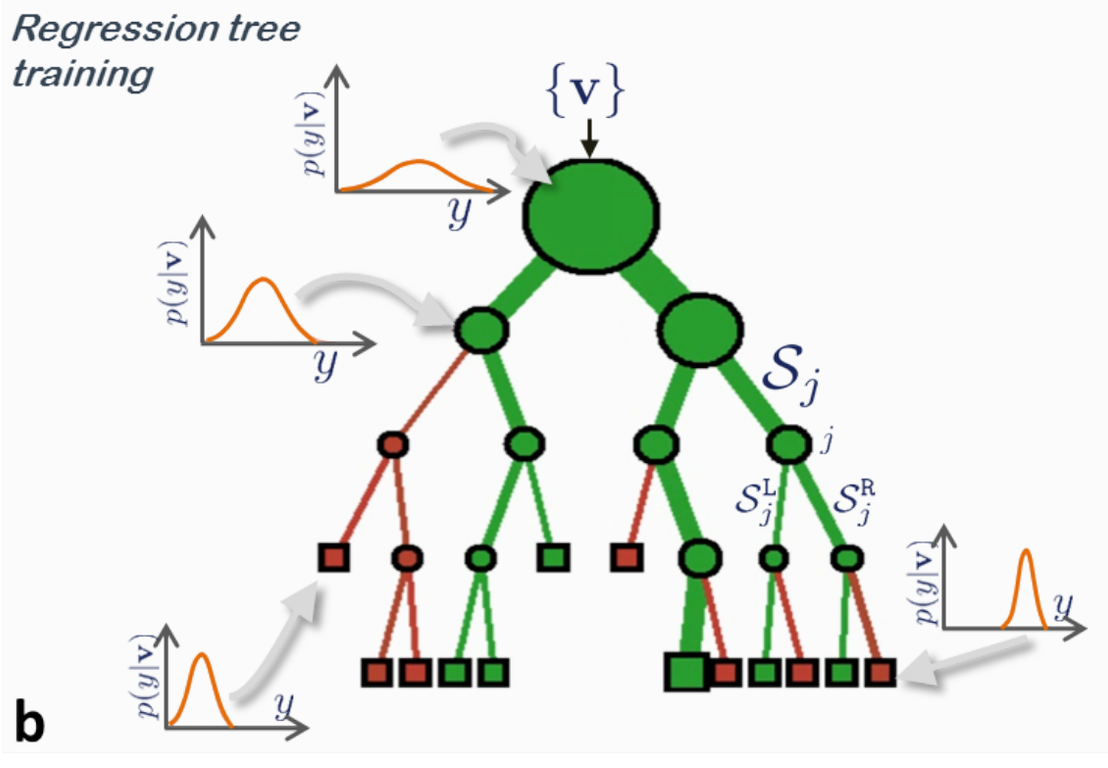

$X_1 \leq t_1$
$X_2 \leq t_2$
$X_1 \leq t_3$
$X_2 \leq t_4$
R_1
R_2
R_3
R_4
R_5


X_2
X_1


Data in feature space
x_2
x_1
?
?
?
?
?
V
a
Classification tree training
$\{\mathbf{v}\}$
$\mathcal{S}$
$\mathcal{S}^{\mathrm{L}}$
$\mathcal{S}^{\mathrm{R}}$
$p(c|\mathbf{v})$
c
b


Training data
y
x
?

a


Regression tree training
{v}
p(y|v)
y
S_j
j
S_j^L
S_j^R

b

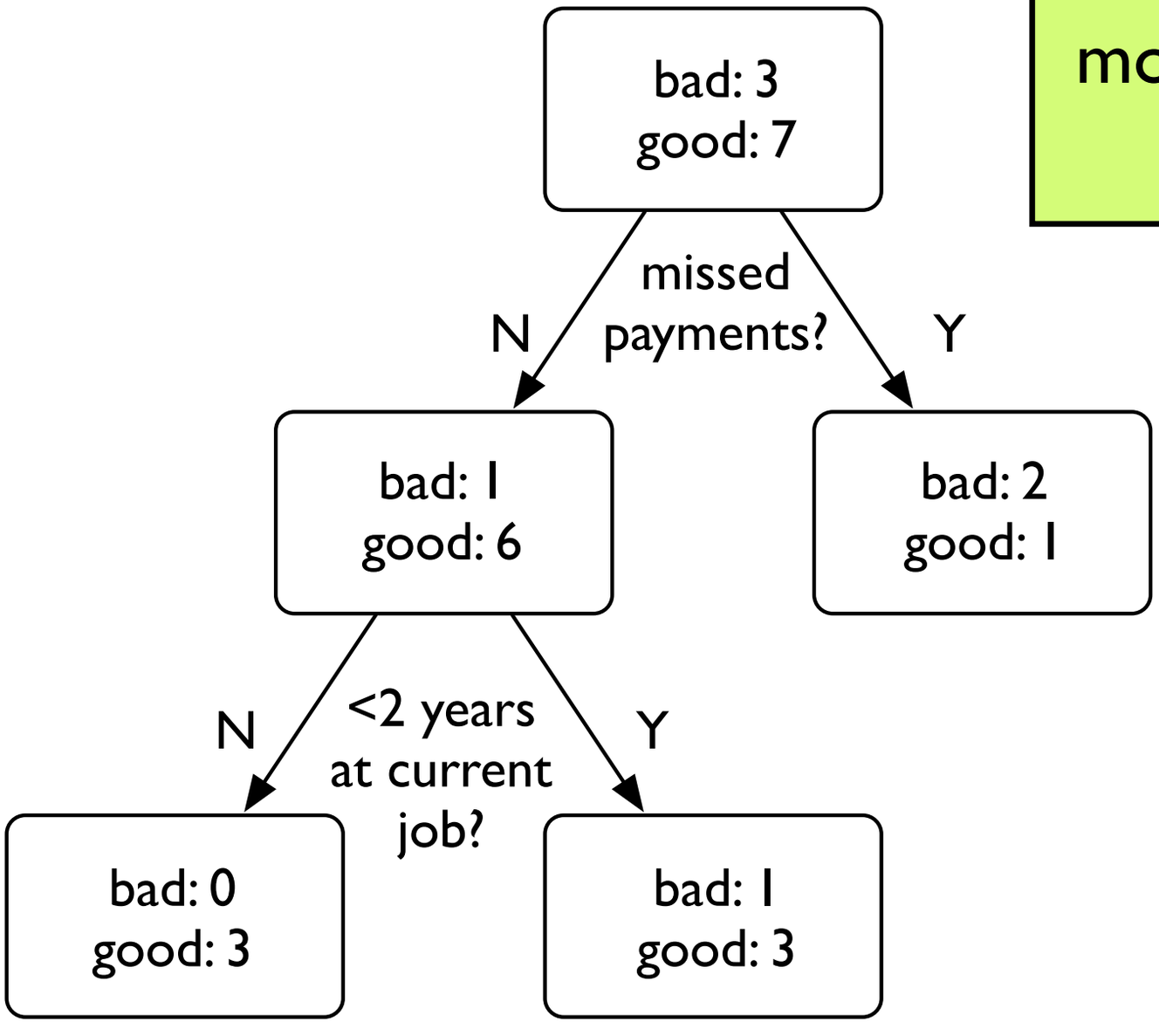
Predicting credit risk

<2 years at current job?	missed payments?	defaulted?
N	N	N
Y	N	Y
N	N	N
N	N	N
N	Y	Y
Y	N	N
N	Y	N
N	Y	Y
Y	N	N
Y	N	N

Predicting credit risk

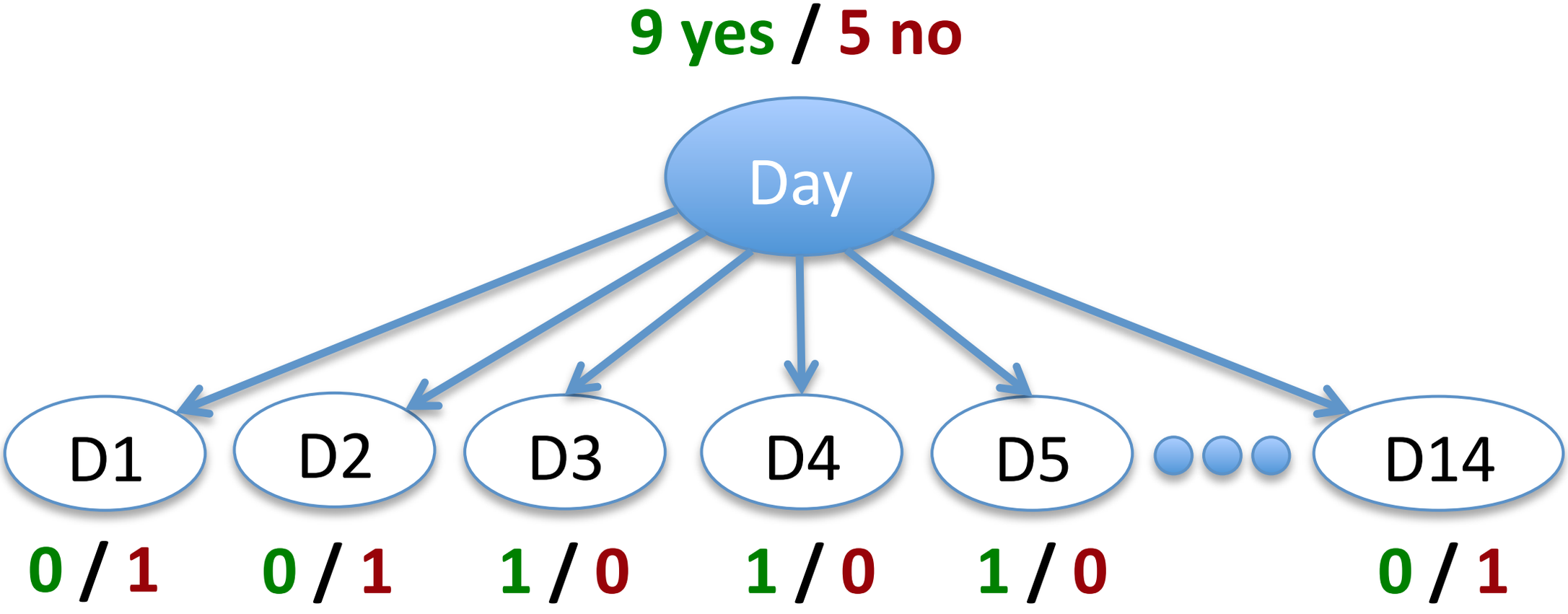
<2 years at current job?	missed payments?	defaulted?
N	N	N
Y	N	Y
N	N	N
N	N	N
N	Y	Y
Y	N	N
N	Y	N
N	Y	Y
Y	N	N
Y	N	N

Missed payments are the most informative attribute about defaulting.



Predicting credit risk

<2 years at current job?	missed payments?	defaulted?
N	N	N
Y	N	Y
N	N	N
N	N	N
N	Y	Y
Y	N	N
N	Y	N
N	Y	Y
Y	N	N
Y	N	N



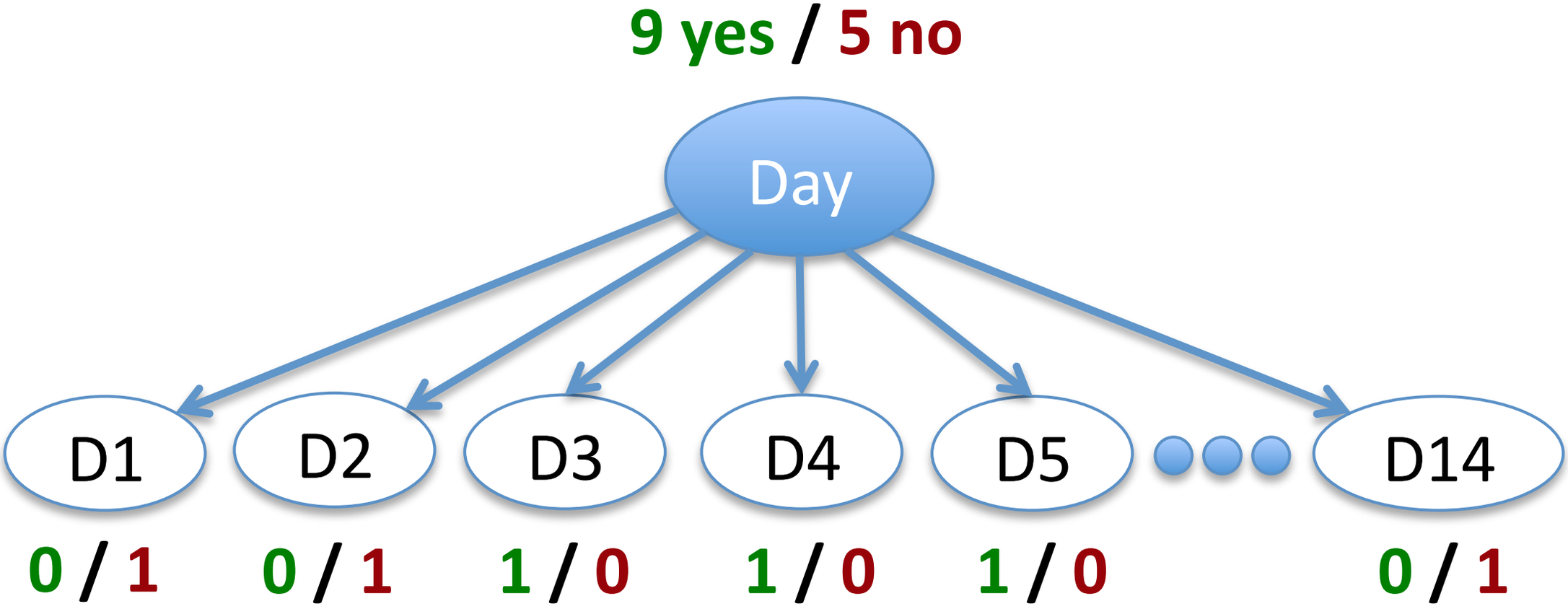
all subsets perfectly pure => optimal split

Training examples: **9 yes / 5 no**

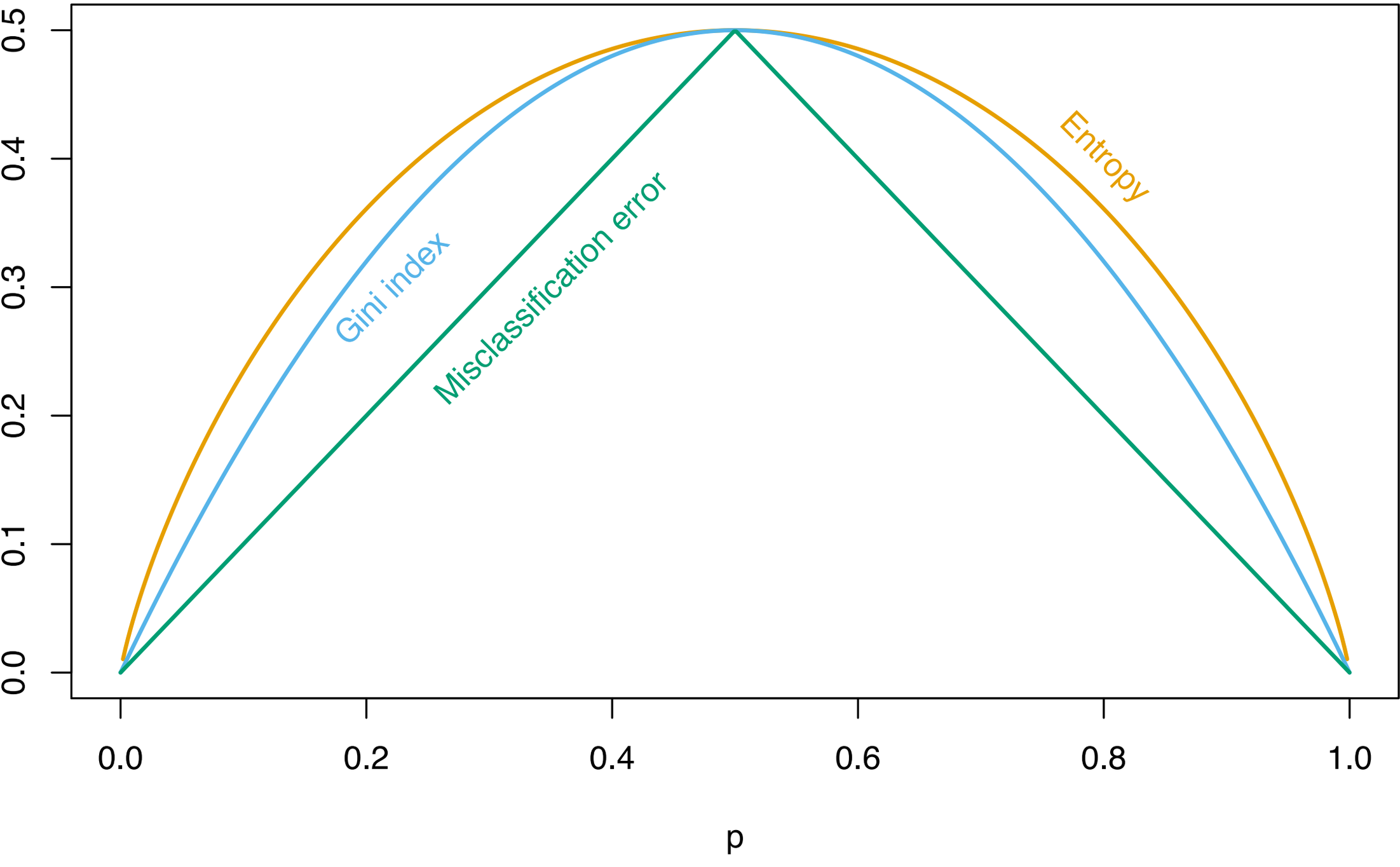
Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

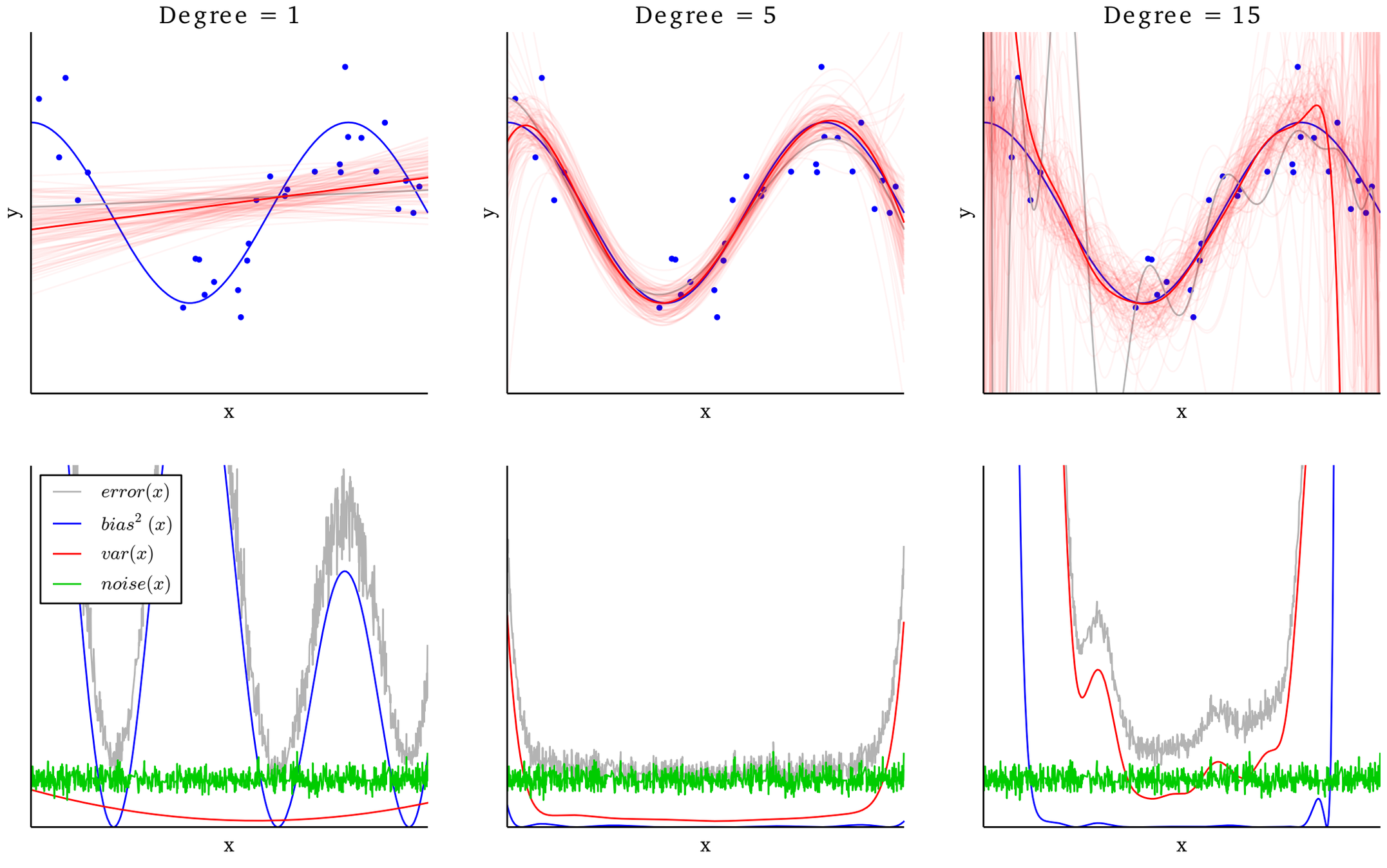
New data:

D15	Rain	High	Weak	?



all subsets perfectly pure => optimal split


0.0
0.1
0.2
0.3
0.4
0.5
0.0
0.2
0.4
0.6
0.8
1.0
p
Gini index
Misclassification error
Entropy


Degree = 1
Degree = 5
Degree = 15
y
x
error(x)
bias² (x)
var(x)
noise(x)