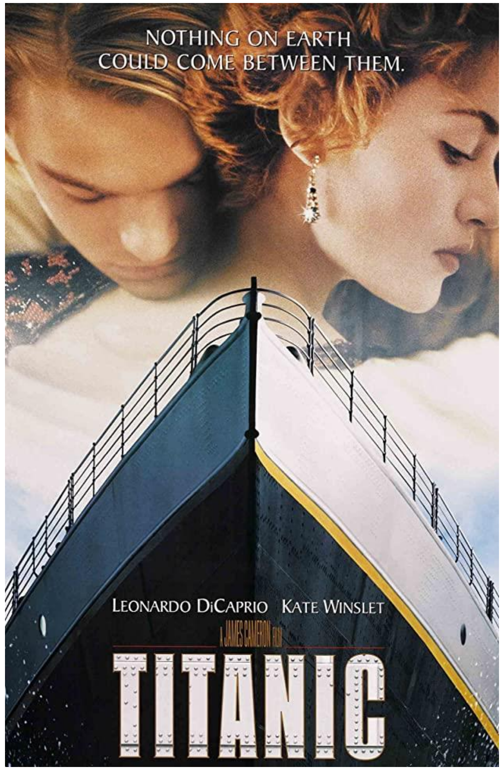
Data Mining

Prof. Kuan-Ting Lai

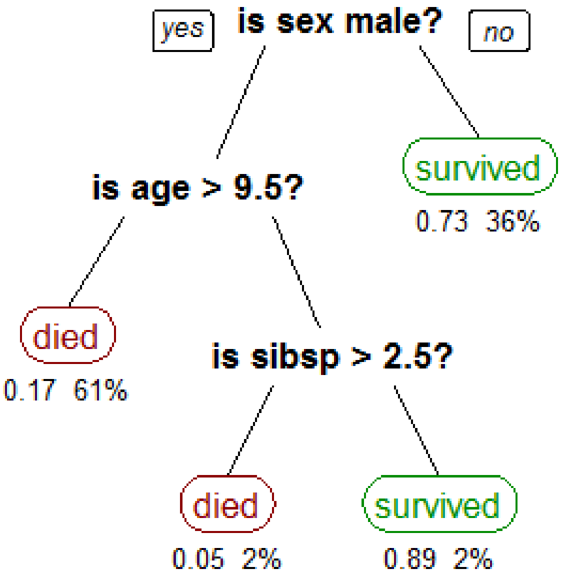
2023/10/25

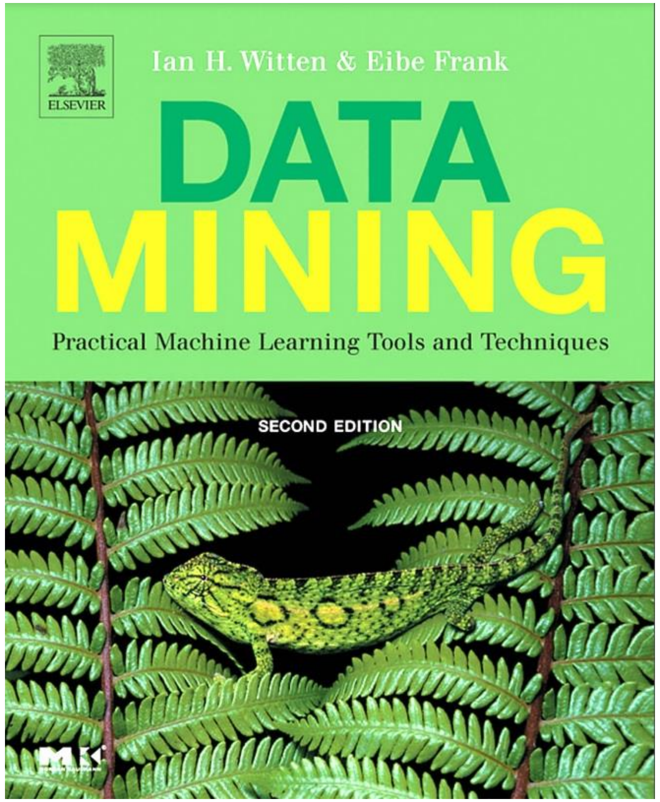
Mining the Rules (Symbolist)

- Decision Tree, expert system, rule-based system, …



Survival rate of Passengers on Titanic





https://www.academia.dk/BiologiskAntropologi/Epidemiologi/DataMining/Witten_and_Frank_DataMining_Weka_2nd_Ed_2005.pdf

Example: the Weather Problem

- Conditions for playing an unspecified game.

Table 1.2 The weather data.

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

ARFF File Format

- A block defining the attributes (`outlook`, `temperature`, `humidity`, `windy`, `play?`).
- Nominal attributes are followed by the set of values they can take on
- Numeric values are followed by the keyword `numeric`.

```
% ARFF file for the weather data with some numeric features
%
@relation weather

@attribute outlook { sunny, overcast, rainy }
@attribute temperature numeric
@attribute humidity numeric
@attribute windy { true, false }
@attribute play? { yes, no }

@data
%
% 14 instances
%
sunny, 85, 85, false, no
sunny, 80, 90, true, no
overcast, 83, 86, false, yes
rainy, 70, 96, false, yes
rainy, 68, 80, false, yes
rainy, 65, 70, true, no
overcast, 64, 65, true, yes
sunny, 72, 95, false, no
sunny, 69, 70, false, yes
rainy, 75, 80, false, yes
sunny, 75, 70, true, yes
overcast, 72, 90, true, yes
overcast, 81, 75, false, yes
rainy, 71, 91, true, no
```

Rules of Playing

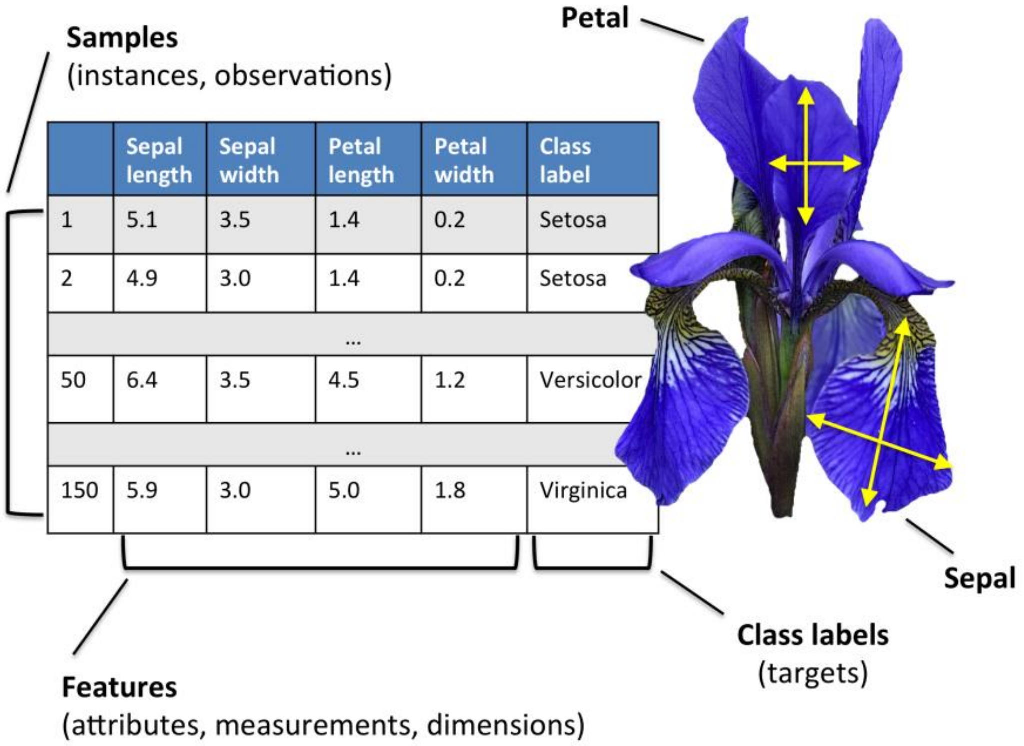
- If outlook = sunny and humidity = high then play = no
- If outlook = rainy and windy = true then play = no
- If outlook = overcast then play = yes
- If humidity = normal then play = yes
- If none of the above then play = yes

Table 1.2 The weather data.

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

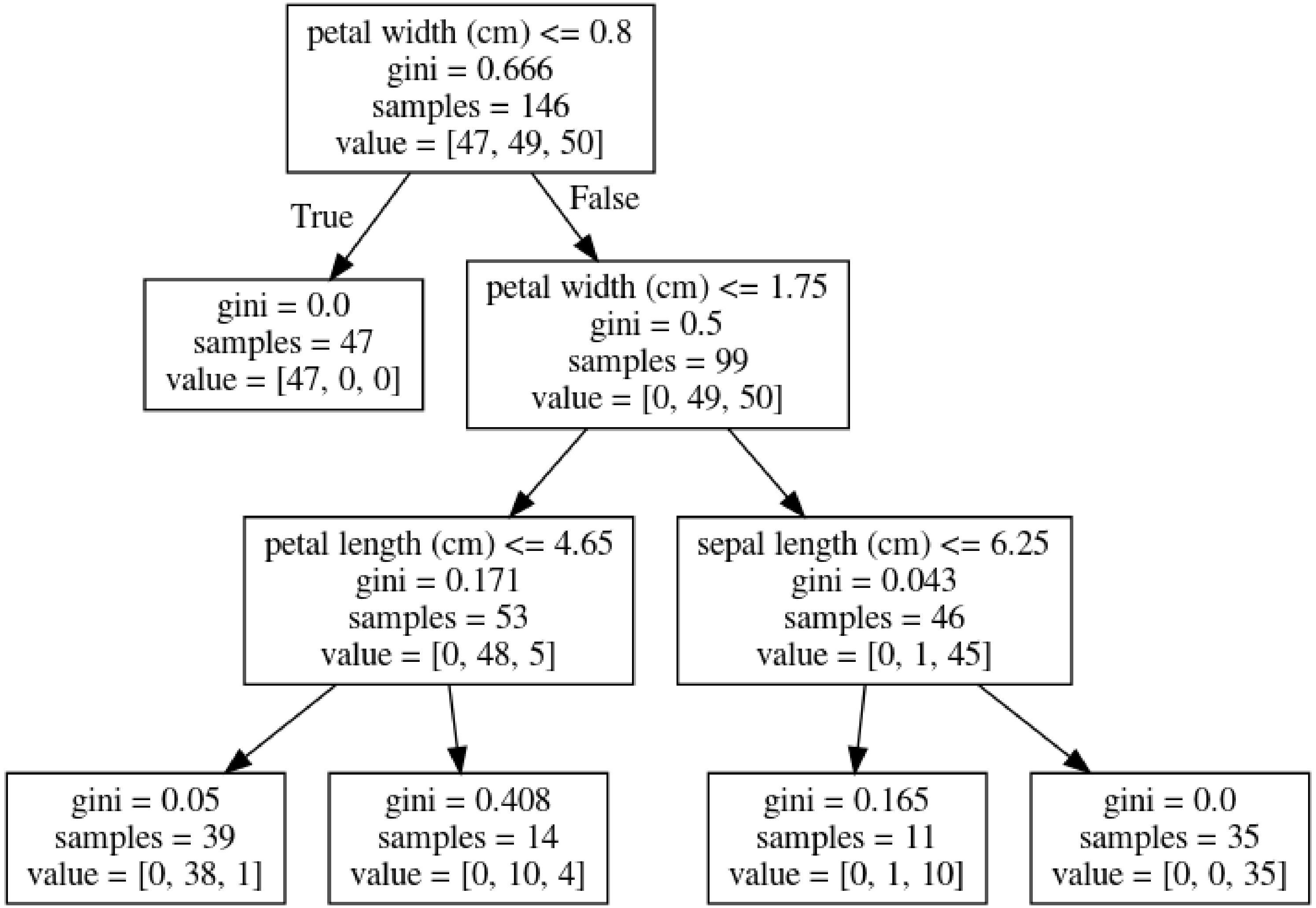
Rules of Classifying Iris Flowers

If sepal width < 2.55 and petal length < 4.95 and petal width < 1.55 then Iris versicolor
If petal length ≥ 2.45 and petal length < 4.95 and petal width < 1.55 then Iris versicolor
If sepal length ≥ 6.55 and petal length < 5.05 then Iris versicolor
If sepal width < 2.75 and petal width < 1.65 and sepal length < 6.05 then Iris versicolor
If sepal length ≥ 5.85 and sepal length < 5.95 and petal length < 4.85 then Iris versicolor
If petal length ≥ 5.15 then Iris virginica
If petal width ≥ 1.85 then Iris virginica
If petal width ≥ 1.75 and sepal width < 3.05 then Iris virginica
If petal length ≥ 4.95 and petal width < 1.55 then Iris virginica



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
…					
50	6.4	3.5	4.5	1.2	Versicolor
…					
150	5.9	3.0	5.0	1.8	Virginica

Decision Tree for Iris Flower Dataset



https://www.kaggle.com/code/shikhnu/decision-tree-iris-dataset

Decision Tree vs. Rule Set

- Both are based on classification rules, but different in the representation.
- Rule sets can retain most important information from a full decision tree but with a less complex model
- Rules can be derived from a Decision Tree

Tree for Numeric Prediction

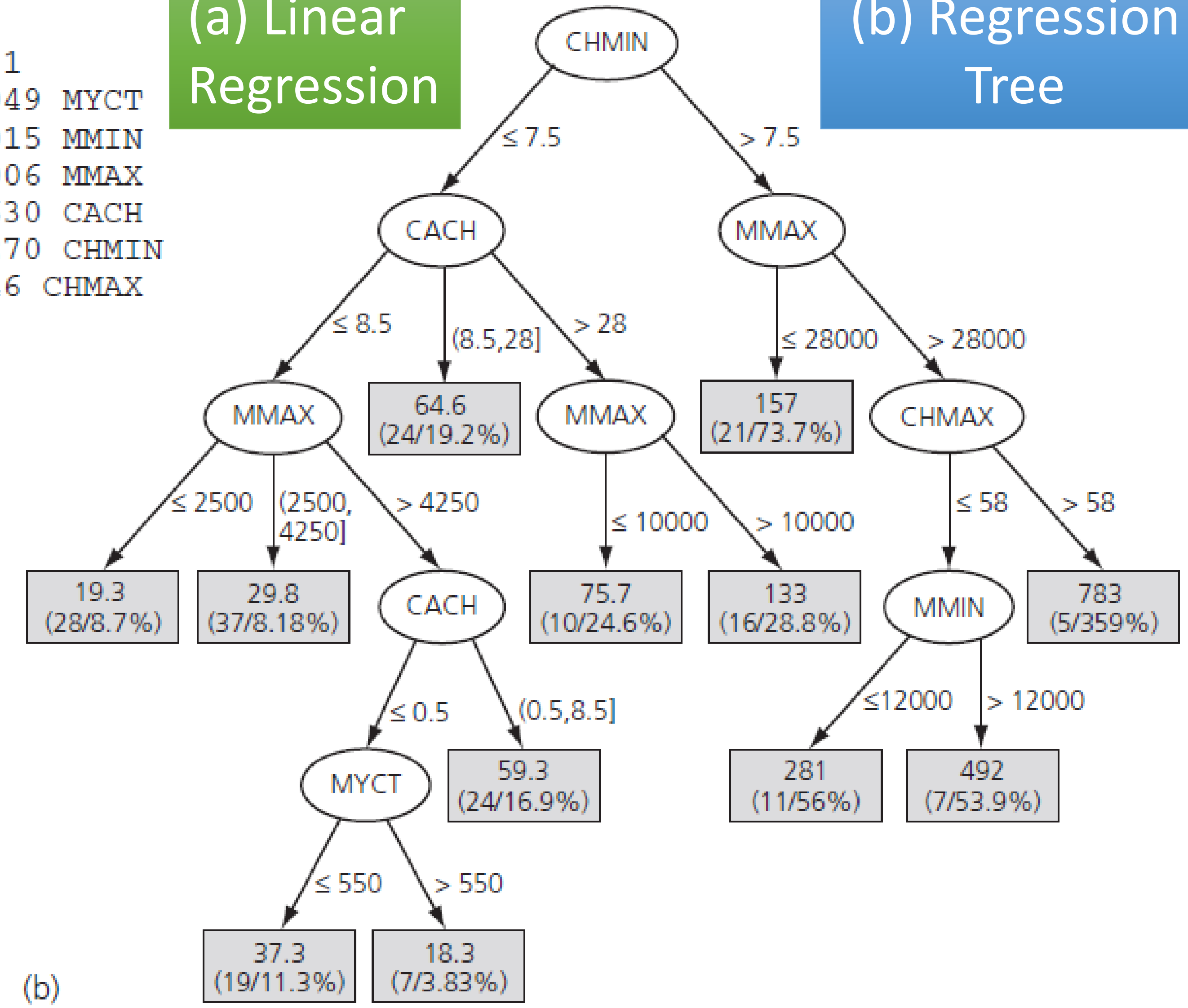
- CPU Performance Dataset
  - vendor: vendor name
  - myct: machine cycle time in nanoseconds (integer)
  - mmin: minimum main memory in kilobytes (integer)
  - mmax: maximum main memory in kilobytes (integer)
  - cach: cache memory in kilobytes (integer)
  - chmin: minimum channels in units (integer)
  - chmax: maximum channels in units (integer)

(a) Linear Regression

```
PRP =
  -56.1
  +0.049 MYCT
  +0.015 MMIN
  +0.006 MMAX
  +0.630 CACH
  -0.270 CHMIN
  +1.46 CHMAX
```

(a)

(b) Regression Tree

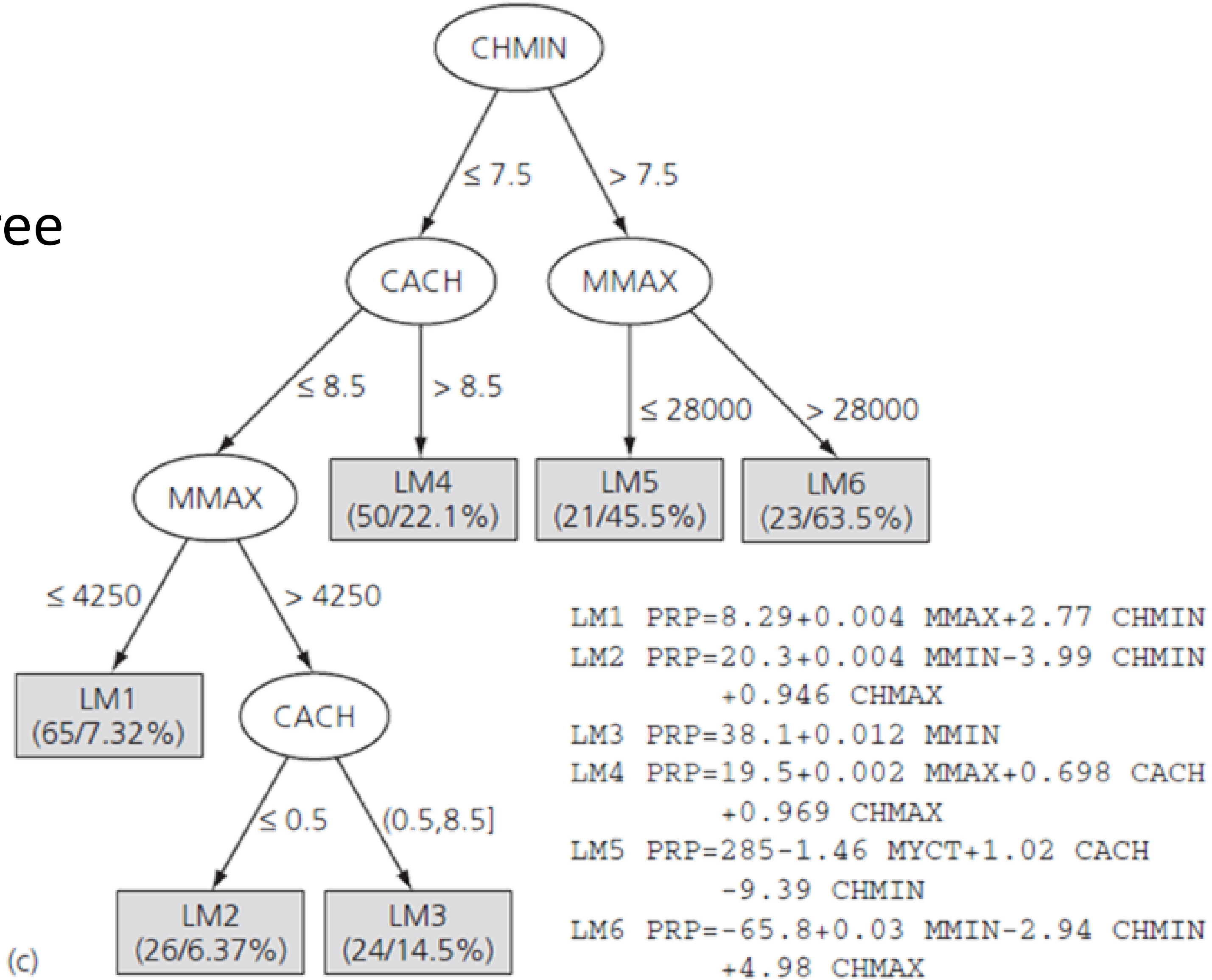


(b)

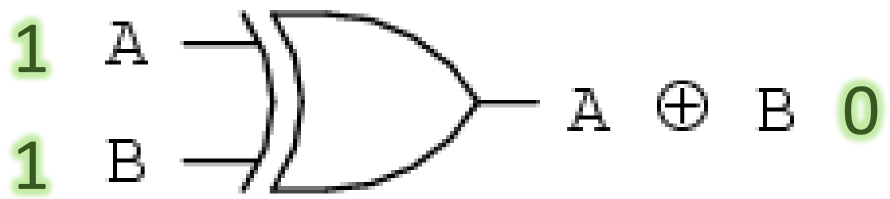
https://rstudio-pubs-static.s3.amazonaws.com/398165_1d19db3fb4c042e1a9938f81724521b9.html

Tree for Numeric Prediction (Model Tree)

- Model Tree = Linear regression + regression tree



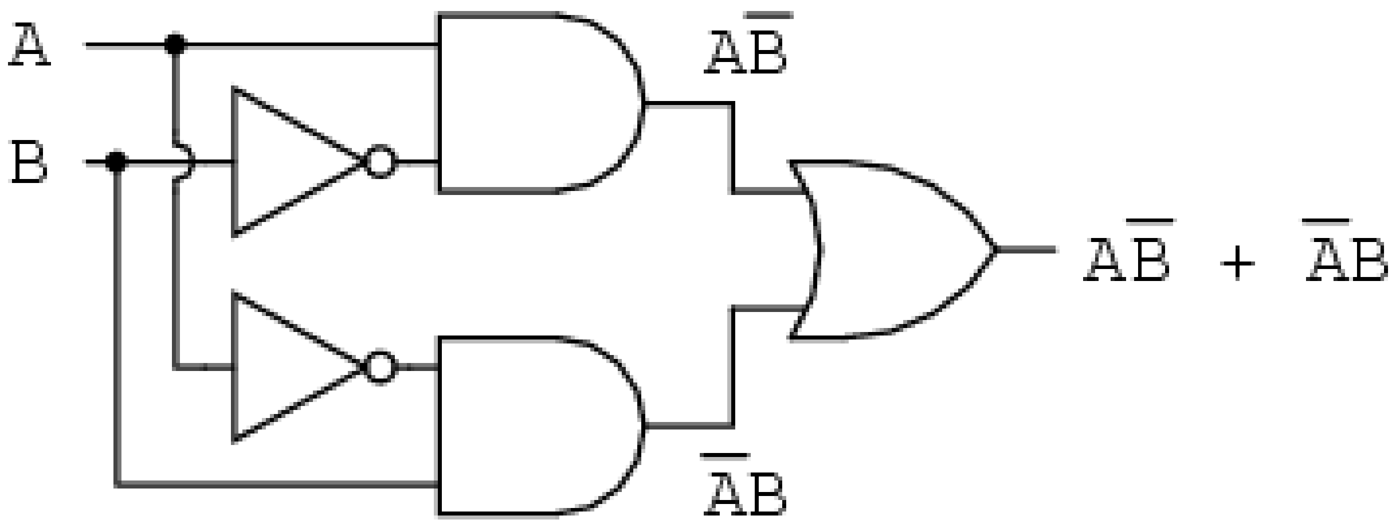
XOR Problem



- Exclusive OR

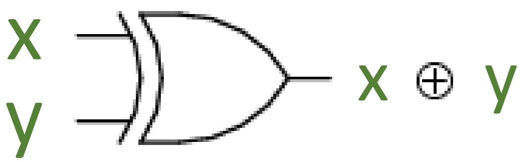
. . . is equivalent to . . .

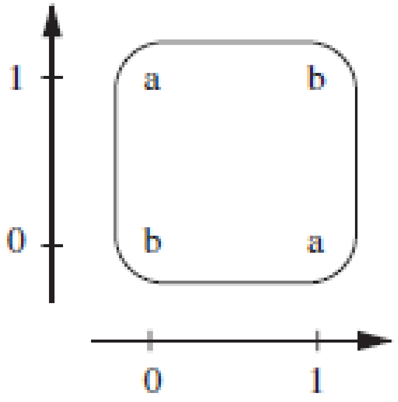
Input		Output
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

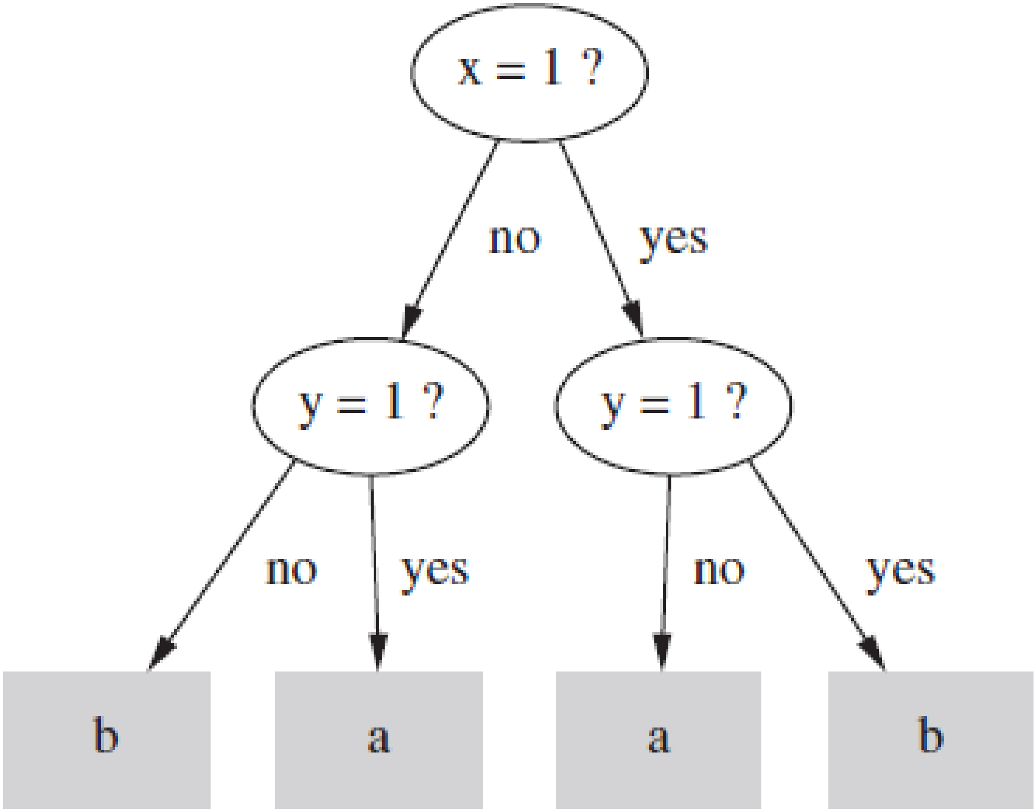


$$A \oplus B = A\overline{B} + \overline{A}B$$

XOR Decision Tree







Input		Output
x	y	Q
0	0	b
0	1	a
1	0	a
1	1	b

If x=1 and y=0 then class = a

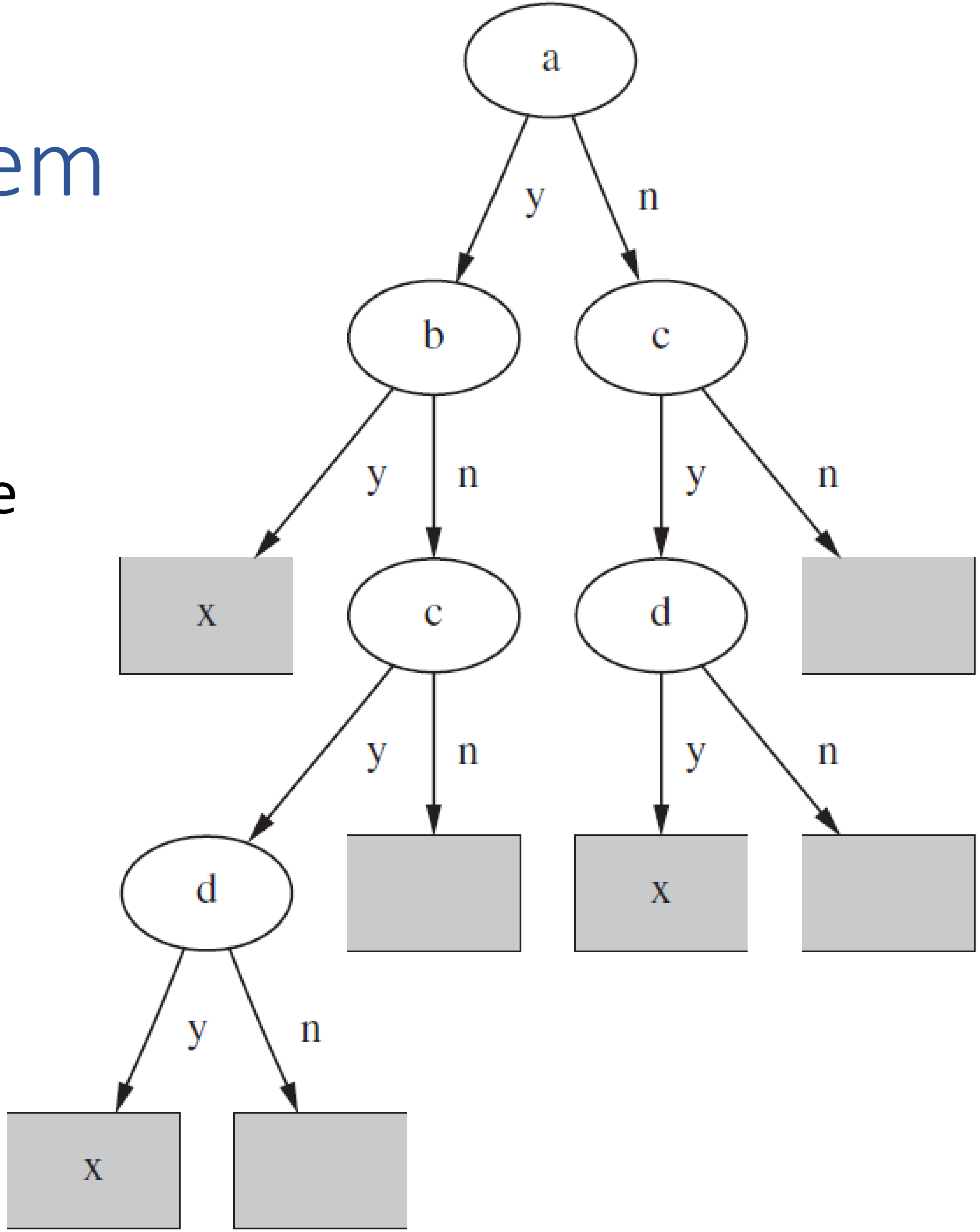
If x=0 and y=1 then class = a

If x=0 and y=0 then class = b

If x=1 and y=1 then class = b

Replicated Subtree Problem

- `If a and b then x`
- `If c and d then x`
- If a is chosen, the second rule must be repeated twice in the tree



1-Rule (1R) Method

- Choose 1 attribute and create a rule
- Example: Weather Problem

Table 1.2 The weather data.

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Table 4.1 Evaluating the attributes in the weather data.

	Attribute	Rules	Errors	Total errors
1	outlook	sunny → no	2/5	4/14
		overcast → yes	0/4	
		rainy → yes	2/5	
2	temperature	hot → no*	2/4	5/14
		mild → yes	2/6	
		cool → yes	1/4	
3	humidity	high → no	3/7	4/14
		normal → yes	1/7	
4	windy	false → yes	2/8	5/14
		true → no*	3/6	

Statistical Modeling

Table 4.2 The weather data with counts and probabilities.

Outlook			Temperature			Humidity			Windy			Play	
	yes	*no*		*yes*	*no*		*yes*	*no*		*yes*	*no*	*yes*	*no*
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

- Predict if to play (yes/no) for the new day
  - likelihood of *yes* = 2/9 * 3/9 * 3/9 * 3/9 * 9/14 = 0.0053
  - likelihood of *no* = 3/5 * 1/5 * 4/5 * 3/5 * 5/14 = 0.0206

Table 4.3 A new day.

Outlook	Temperature	Humidity	Windy	Play
sunny	cool	high	true	?

Normalize Probability of Yes / No

- Predict if to play (yes/no) for the new day
  - likelihood of yes = 2/9 * 3/9 * 3/9 * 3/9 * 9/14 = 0.0053
  - likelihood of no = 3/5 * 1/5 * 4/5 * 3/5 * 5/14 = 0.0206

$$\text{Probability of } yes = \frac{0.0053}{0.0053+0.0206} = 20.5\%,$$

$$\text{Probability of } no = \frac{0.0206}{0.0053+0.0206} = 79.5\%.$$

Bayes Rule

- Naïve Bayes: assume attributes are independent

Training Data

likelihood

Prior

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

class

Evidence

features

$$\Pr[yes|E] = \frac{\Pr[E_1|yes] \times \Pr[E_2|yes] \times \Pr[E_3|yes] \times \Pr[E_4|yes] \times \Pr[yes]}{\Pr[E]}$$

$$\Downarrow$$

$$\Pr[yes|E] = \frac{2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14}{\Pr[E]}$$

Numeric Attributes

- Assume normal distribution and calculate mean and variance

$$f(temperature = 66 \mid yes) = \frac{1}{\sqrt{2\pi} \cdot 6.2} e^{\frac{(66-73)^2}{2 \cdot 6.2^2}} = 0.0340.$$

$$f(humidity = 90 \mid yes) = 0.0221$$

Table 4.4 The numeric weather data with summary statistics.

Outlook			Temperature			Humidity			Windy			Play	
	yes	*no*		*yes*	*no*		*yes*	*no*		*yes*	*no*	*yes*	*no*
sunny	2	3		83	85		86	85	false	6	2	9	5
overcast	4	0		70	80		96	90	true	3	3		
rainy	3	2		68	65		80	70					
				64	72		65	95					
				69	71		70	91					
				75			80						
				75			70						
				72			90						
				81			75						
sunny	2/9	3/5	*mean*	73	74.6	*mean*	79.1	86.2	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	*std. dev.*	6.2	7.9	*std. dev.*	10.2	9.7	true	3/9	3/5		
rainy	3/9	2/5											

Yes/No Probability with Numeric Values

$$\text{likelihood of } yes = 2/9 \times 0.0340 \times 0.0221 \times 3/9 \times 9/14 = 0.000036,$$
$$\text{likelihood of } no = 3/5 \times 0.0221 \times 0.0381 \times 3/5 \times 5/14 = 0.000108;$$

$$\text{Probability of } yes = \frac{0.000036}{0.000036 + 0.000108} = 25.0\%,$$

$$\text{Probability of } no = \frac{0.000108}{0.000036 + 0.000108} = 75.0\%.$$

Divide & Conquer: Building Decision Trees

- Choose the most informative attribute to split
- How to measure the amount of information?

Entropy: $H(x) = E[I(x)] = -E[\log P(x)]$

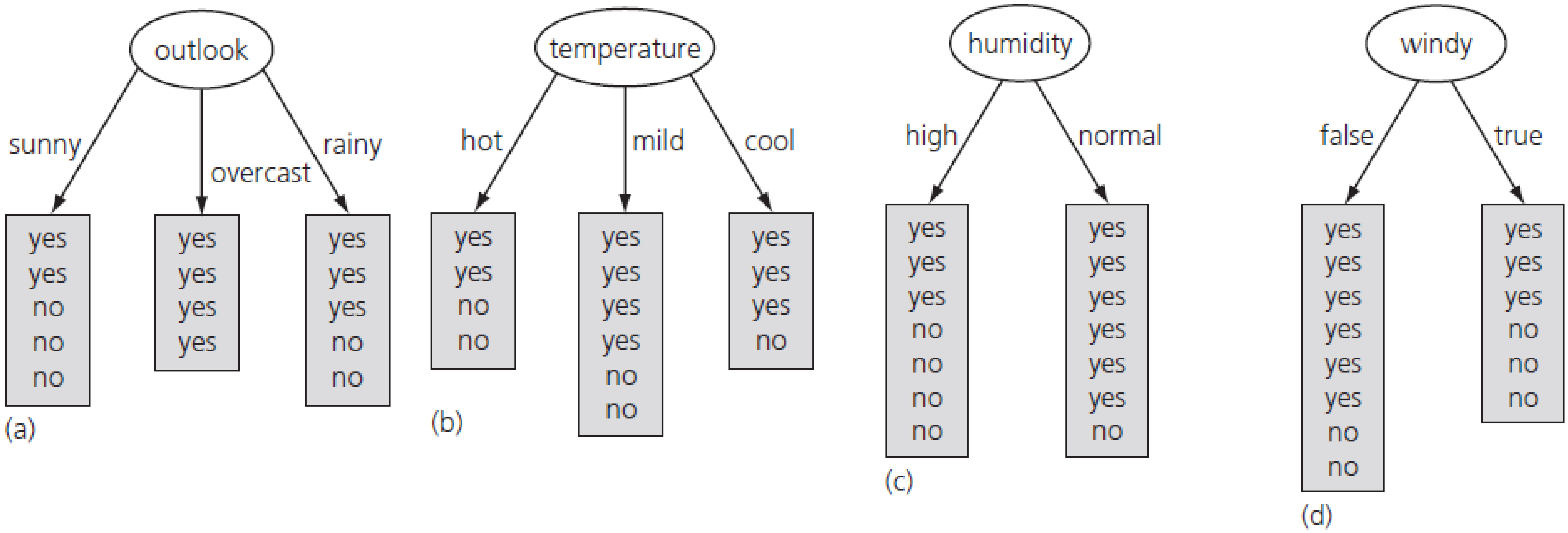
- Information Gain

Kullback-Leibler (KL) Divergence

$$D_{KL}(p||q) = E[\log P(X) - \log Q(X)] = E\left[\log \frac{P(x)}{Q(x)}\right]$$

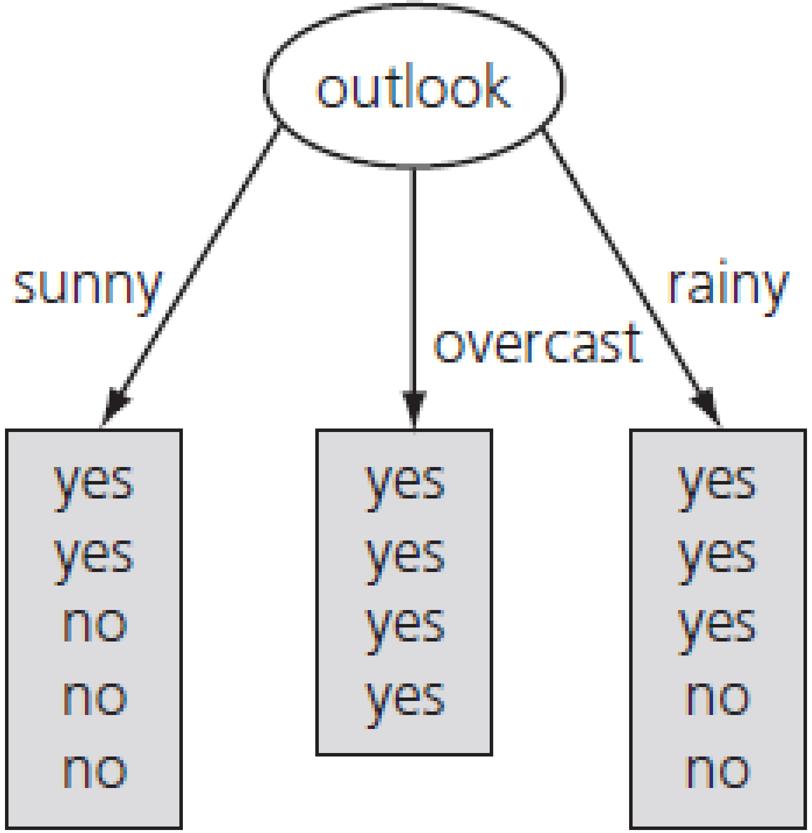
Building Decision Tree for Weather Dataset

- Tree stumps for the weather data



Entropy of an Attribute (Outlook)

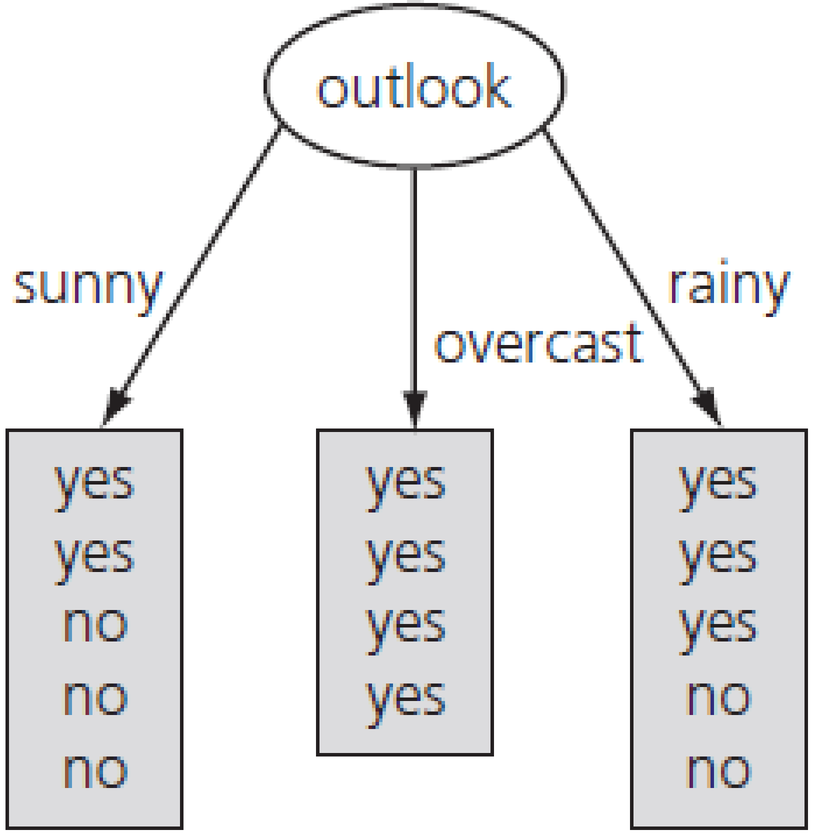
- Sunny predictions: yes*2, no*3
- info(sunny) = info([2,3])
  - $-\frac{2}{5}\log_2\left(\frac{2}{5}\right) + -\frac{3}{5}\log_2\left(\frac{3}{5}\right) = 0.971\text{bits}$
- info(overcast) = info([4,0]) = 0 bits
- info(rainy) = info([3,2]) = 0.971 bits



Information Gain of an Attribute (Outlook)

- Info(outlook) =
  $-\frac{9}{14}\log_2\left(\frac{9}{14}\right) + -\frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits}$

- info(sunny, overcast, rainy) = $\frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971 = 0.693$
- gain(outlook) = Info(outlook) - info(sunny, overcast, rainy) = 0.940 - 0.693 = 0.247 bits



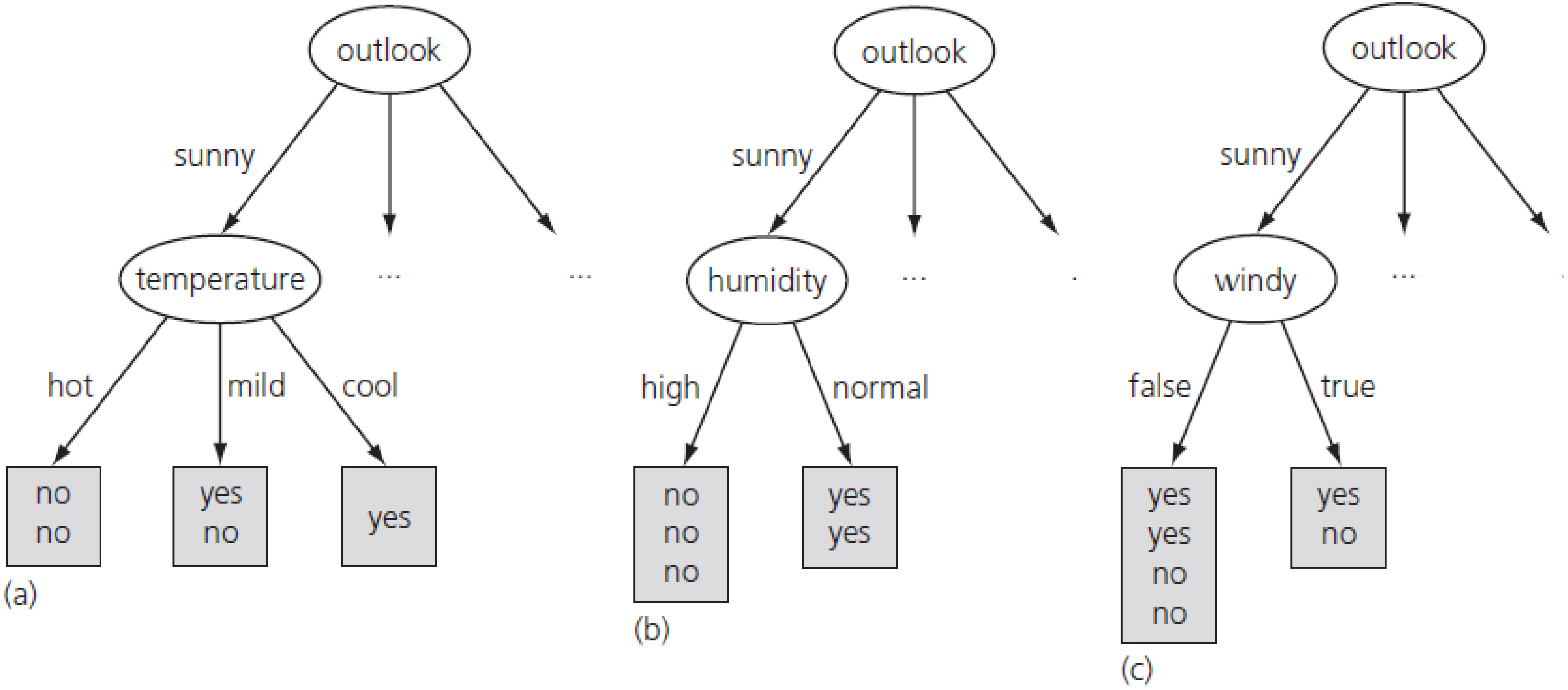
Select the Attribute with Max Information Gain

Select the attribute with max gain (outlook)

- gain(outlook) = 0.247 bits
- gain(temperature) = 0.029 bits
- gain(humidity) = 0.152 bits
- gain(windy) = 0.048 bits

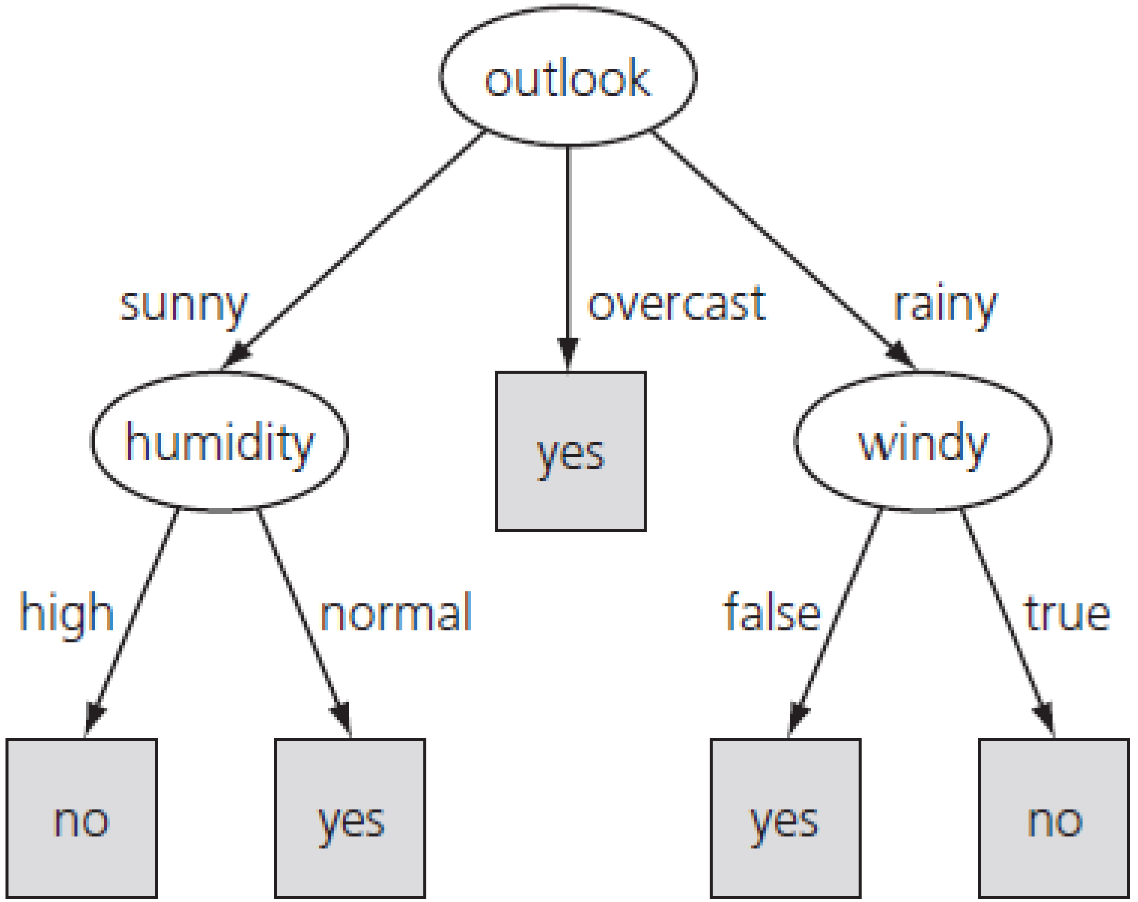
Select “Outlook, Sunny” and Keep Splitting

- gain(temperature) = 0.571 bits
- gain(humidity) = 0.971 bits
- gain(windy) = 0.02 bits



Final Decision Tree for the Weather Dataset

- Continue splitting until all leaf nodes are pure predictions



Decision Trees

- ID3
- C4.5
- CART

Iterative Dichotomiser 3 (ID3)

- Ross Quinlan, “Induction of Decision Trees.” Mach. Learn. 1, 1 (Mar. 1986), 81–106
- Core idea: Use information gain to select attributes
- Dataset Entropy
  - $\mathrm{H}(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{|D|}$,

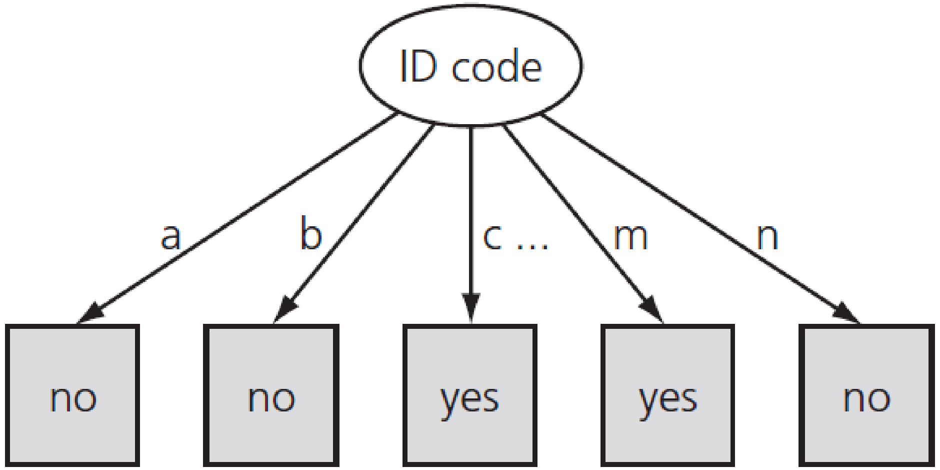
  where D is training data, C_k is the samples of class k
- Attribute Entropy
  - $\mathrm{H}(D|A) = \sum_{i=1}^{N} \frac{|D_i|}{|D|} H(D_i) = -\sum_{i=1}^{N} \frac{|D_i|}{|D|} \left(\sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} \log_2 \frac{|D_{ik}|}{|D_i|}\right)$
- $Gain(D, A) = \mathrm{H}(D) - H(D|A)$

https://zhuanlan.zhihu.com/p/85731206

Problems of ID3

- No pruning strategy and easy to overfitting
- Can handle only discrete data
- Prefer attributes with more features, such as “ID”

ID code	Outlook	Temperature	Humidity	Windy	Play
a	sunny	hot	high	false	no
b	sunny	hot	high	true	no
c	overcast	hot	high	false	yes
d	rainy	mild	high	false	yes
e	rainy	cool	normal	false	yes
f	rainy	cool	normal	true	no
g	overcast	cool	normal	true	yes
h	sunny	mild	high	false	no
i	sunny	cool	normal	false	yes
j	rainy	mild	normal	false	yes
k	sunny	mild	normal	true	yes
l	overcast	mild	high	true	yes
m	overcast	hot	normal	false	yes
n	rainy	mild	high	true	no



C4.5

- Quinlan, J. R. C4.5: *Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.
- Improvements from ID3
  - Handling both continuous and discrete attributes - For continuous attributes, C4.5 creates a threshold and then splits the list
  - Handling training data with missing attribute values - Missing attribute values are simply not used in gain and entropy calculations.
  - Handling attributes with differing costs.
  - Pruning trees after creation - C4.5 goes back through the tree once it's been created and attempts to remove branchesx that do not help by replacing them with leaf nodes.

C4.5 Pseudocode

1. Check for the above base cases.
2. For each attribute *a*, find the normalized information gain ratio from splitting on *a*.
3. Let *a_best* be the attribute with the highest normalized information gain.
4. Create a decision *node* that splits on *a_best*.
5. Recurse on the sublists obtained by splitting on *a_best*, and add those nodes as children of *node*.

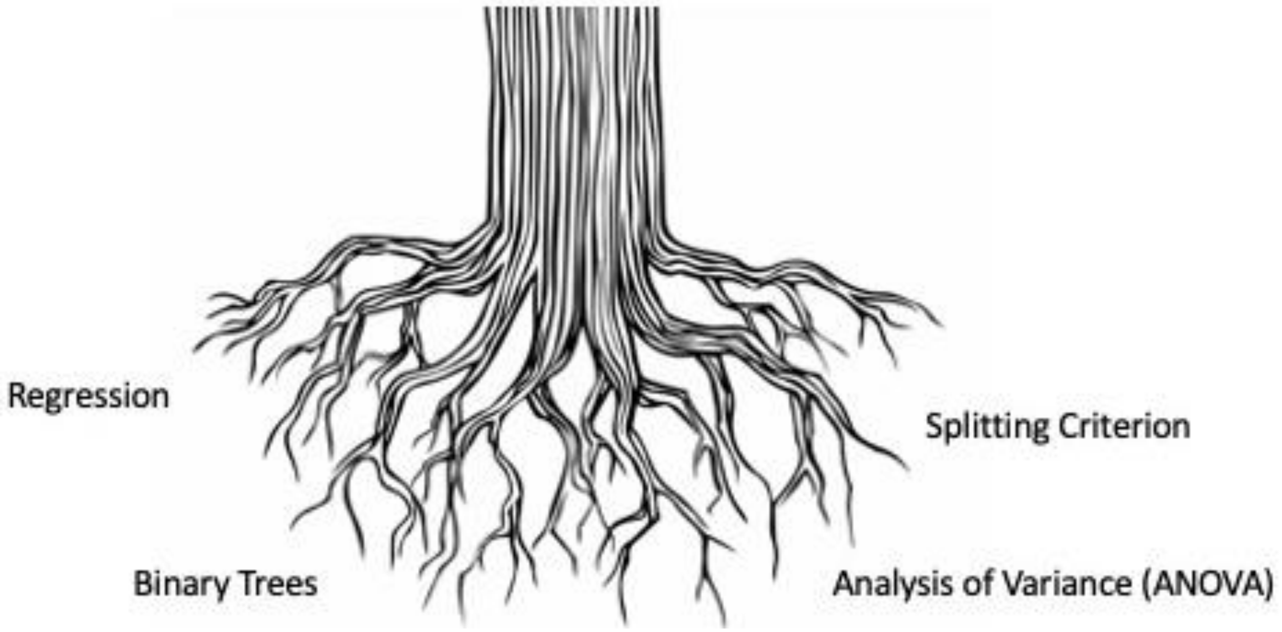
https://en.wikipedia.org/wiki/C4.5_algorithm

C5.0

- Commercial Software by Quinlan (1996)
- Faster and more memory efficient than C4.5
- Smaller decision trees
- Support for boosting and weighting
- Winnowing - a C5.0 option automatically winnows the attributes to remove those that may be unhelpful.

Classification And Regression Tree (CART)

- Sometimes CART is used as an umbrella term
- The CART introduced here was proposed by Leo Breiman and Charles Joel Stone, along with Jerome H. Friedman and Richard Olshen in 1984



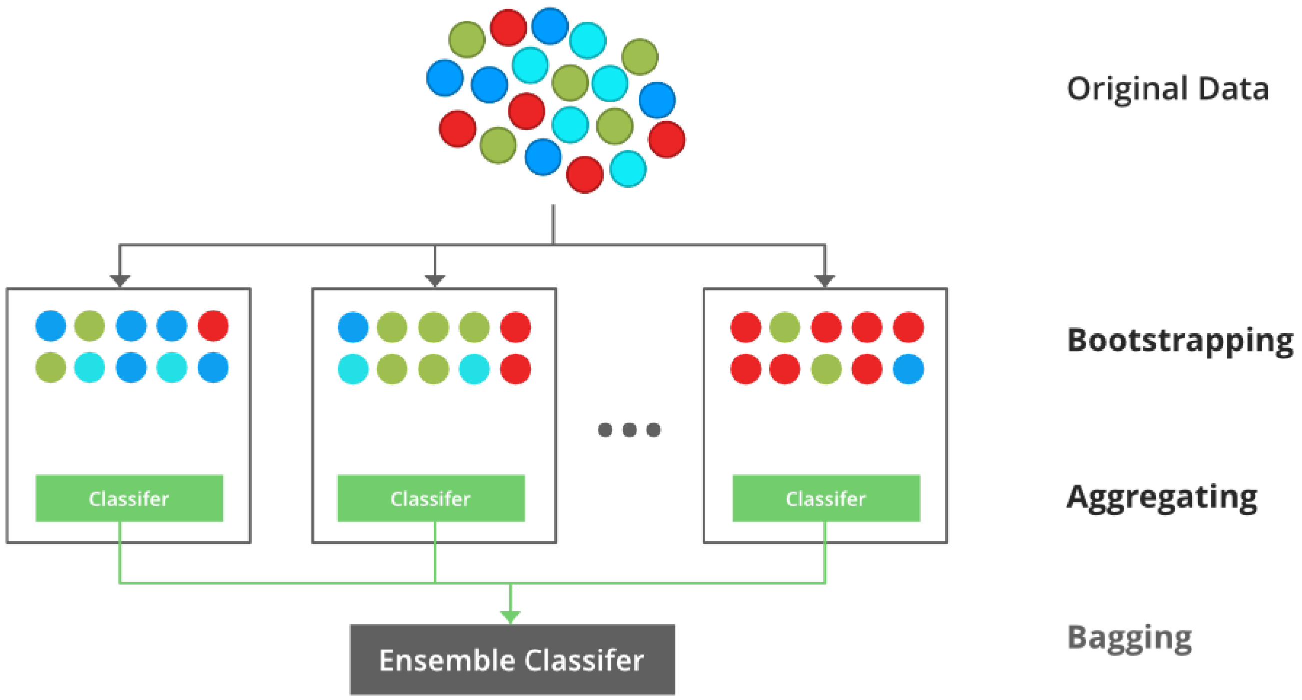
https://lnob.unescap.org/roots-our-lnob-trees

Ensemble Method: Bagging vs. Boosting

- Ensemble methods are made up of a set of classifiers
- A group of weak classifiers can be integrated to be a strong classifier
- Two learning strategies
  - Bagging
    - Learn weak classifiers in parallel and combine them later
  - Boosting
    - Learn weak classifiers sequentially and adaptively to improve model predictions

Bagging

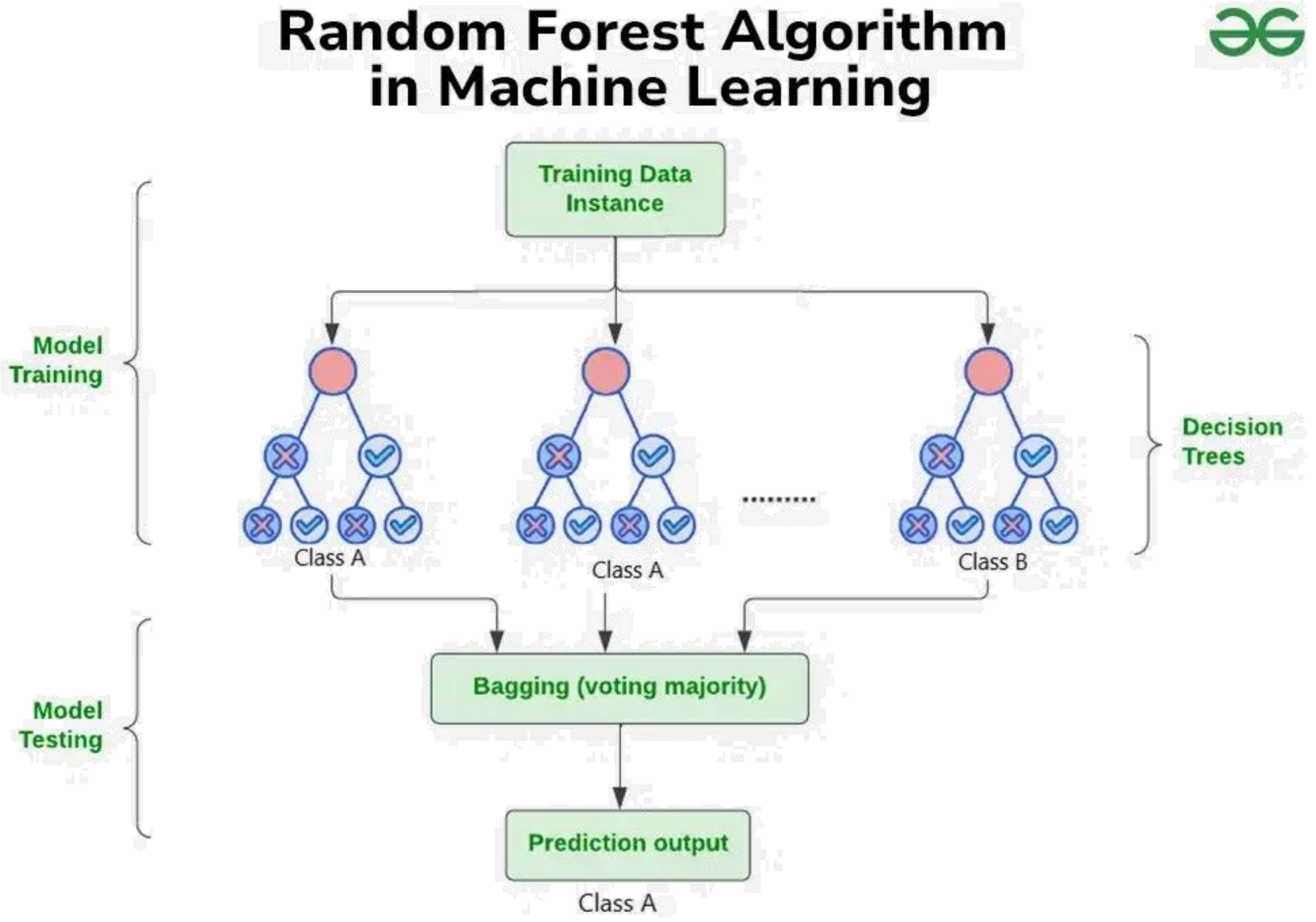
- Multiple subsets are created from the original data
- A base model is created on each of these subsets.
- Each model is learned in parallel with each training set and independent of each other.
- Combine the predictions of all the models to make final prediction



https://www.geeksforgeeks.org/bagging-vs-boosting-in-machine-learning/

Random Forest

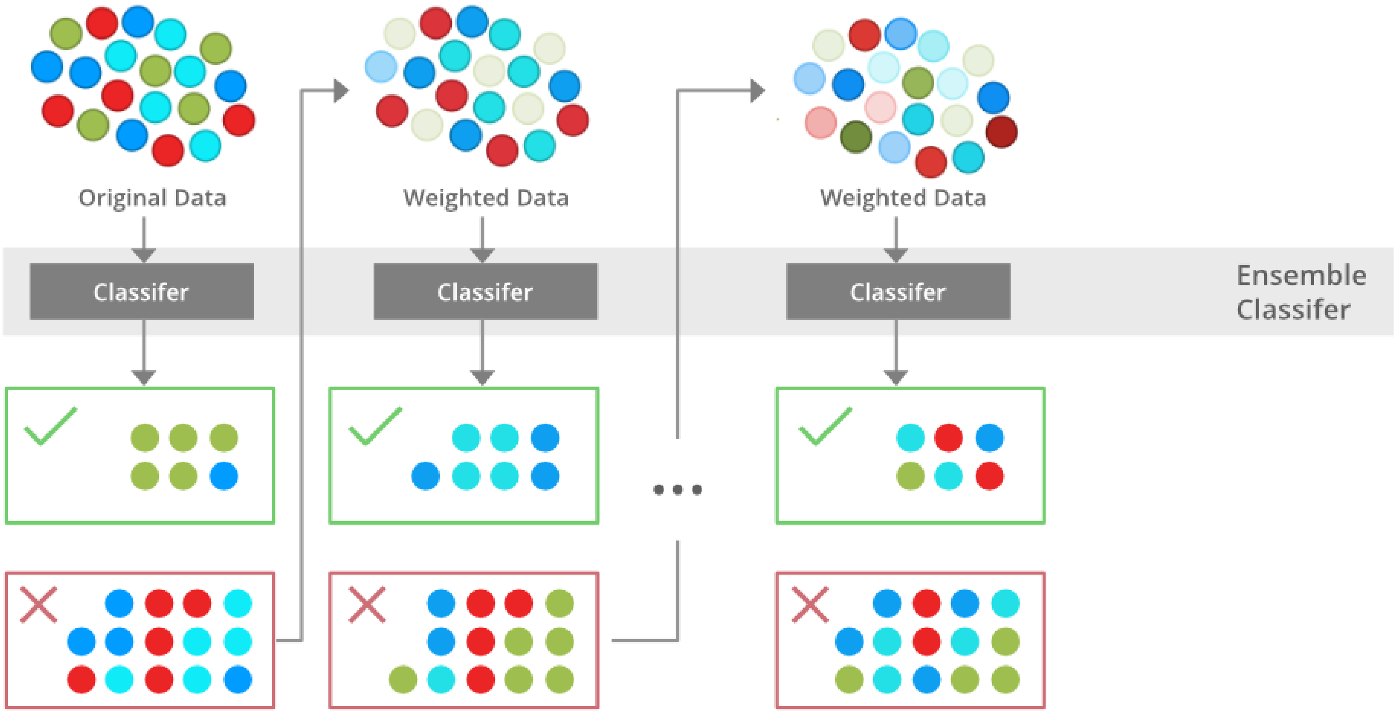
1. Select random K data points from the training set.
2. Build the decision trees associated with the selected data points(Subsets).
3. Choose the number N for decision trees that you want to build.
4. Repeat Step 1 and 2.
5. For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.



https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/

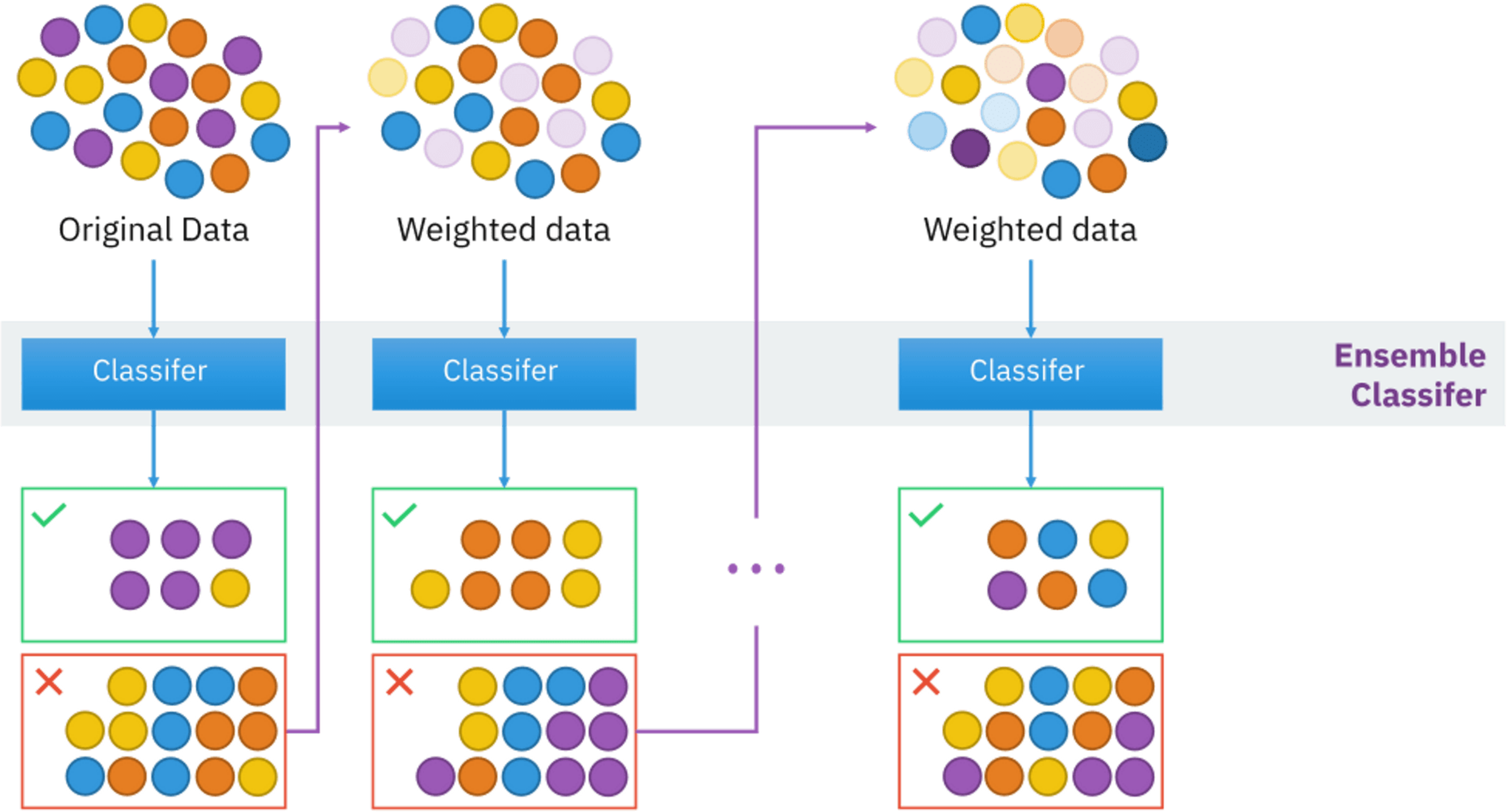
Boosting

1. *Assign equal weight to each of the data point.*
2. *Train a model and identify the wrongly classified data points.*
3. *Increase the weight of the wrongly classified data points and decrease the weights of correctly classified data points.*
4. *if (got required results)*
   Goto step 5
   else
   Goto step 2
5. *End*



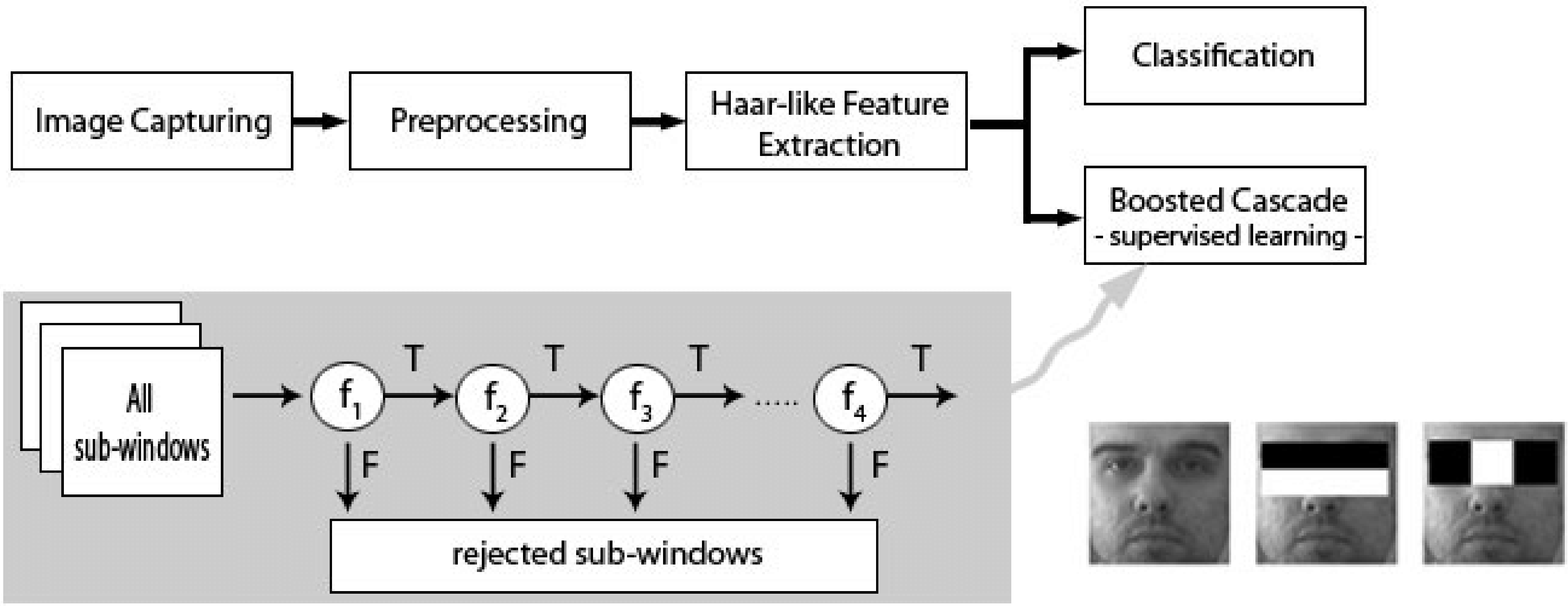
https://www.geeksforgeeks.org/bagging-vs-boosting-in-machine-learning/

AdaBoost (Adaptive Boosting)



https://www.almabetter.com/bytes/tutorials/data-science/adaboost-algorithm

AdaBoost for Face Detection



Bagging vs. Boosting

.NO	Bagging	Boosting
1.	**The simplest way of combining predictions that belong to the same type.**	**A way of combining predictions that belong to the different types.**
2.	**Aim to decrease variance, not bias.**	**Aim to decrease bias, not variance.**
3.	**Each model receives equal weight.**	**Models are weighted according to their performance.**
4.	**Each model is built independently.**	**New models are influenced by the performance of previously built models.**
5.	**Different training data subsets are selected using row sampling with replacement and random sampling methods from the entire training dataset.**	**Iteratively train models, with each new model focusing on correcting the errors (misclassifications or high residuals) of the previous models**
6.	**Bagging tries to solve the over-fitting problem.**	**Boosting tries to reduce bias.**
7.	**If the classifier is unstable (high variance), then apply bagging.**	**If the classifier is stable and simple (high bias) the apply boosting.**
8.	**In this base classifiers are trained parallelly.**	**In this base classifiers are trained sequentially.**

https://www.geeksforgeeks.org/bagging-vs-boosting-in-machine-learning/