POLITECNICO DI MILANO

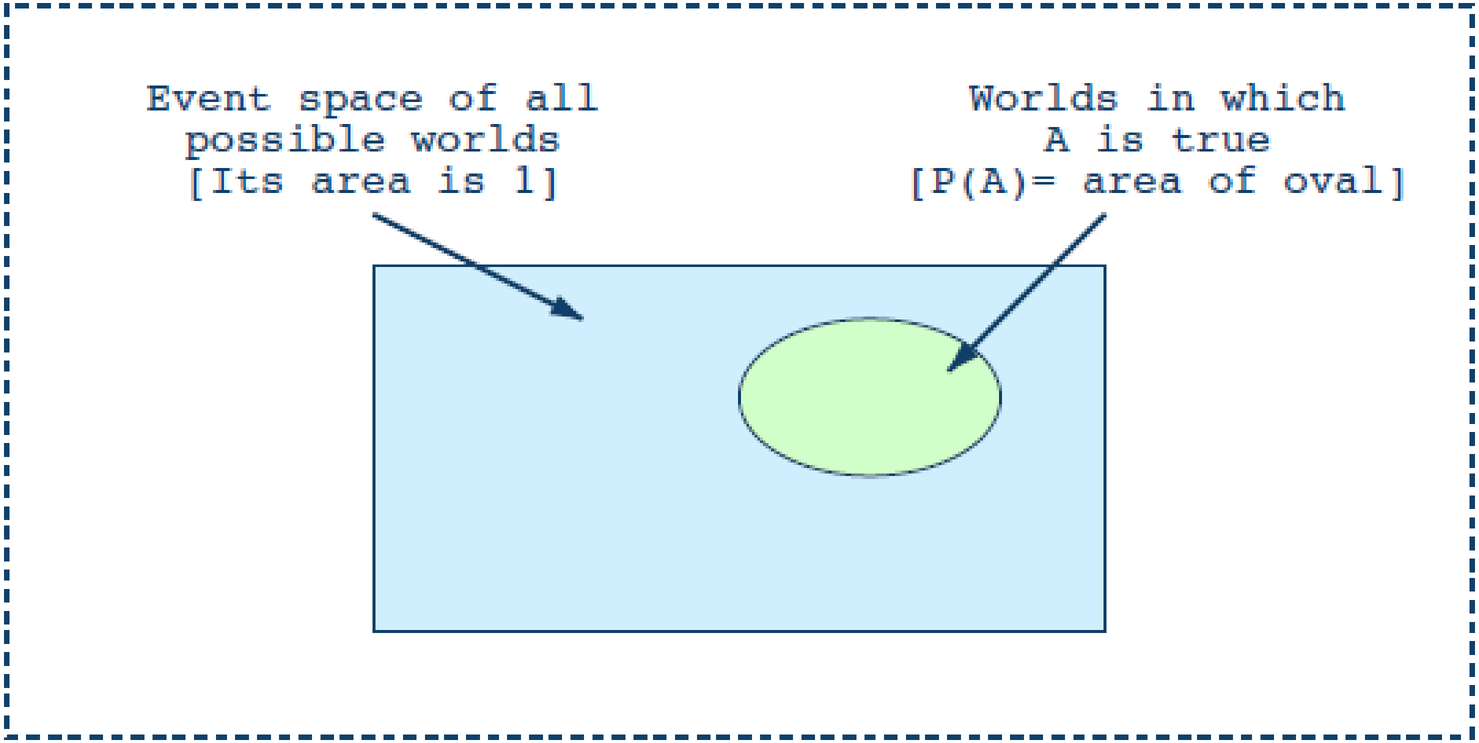
Decision Trees

Information Retrieval and Data Mining

Prof. Matteo Matteucci

Probability for Data Miners

Boolean-valued random variable A is a Boolean-valued random variable if A denotes an event, and there is some degree of uncertainty as to whether A occurs.

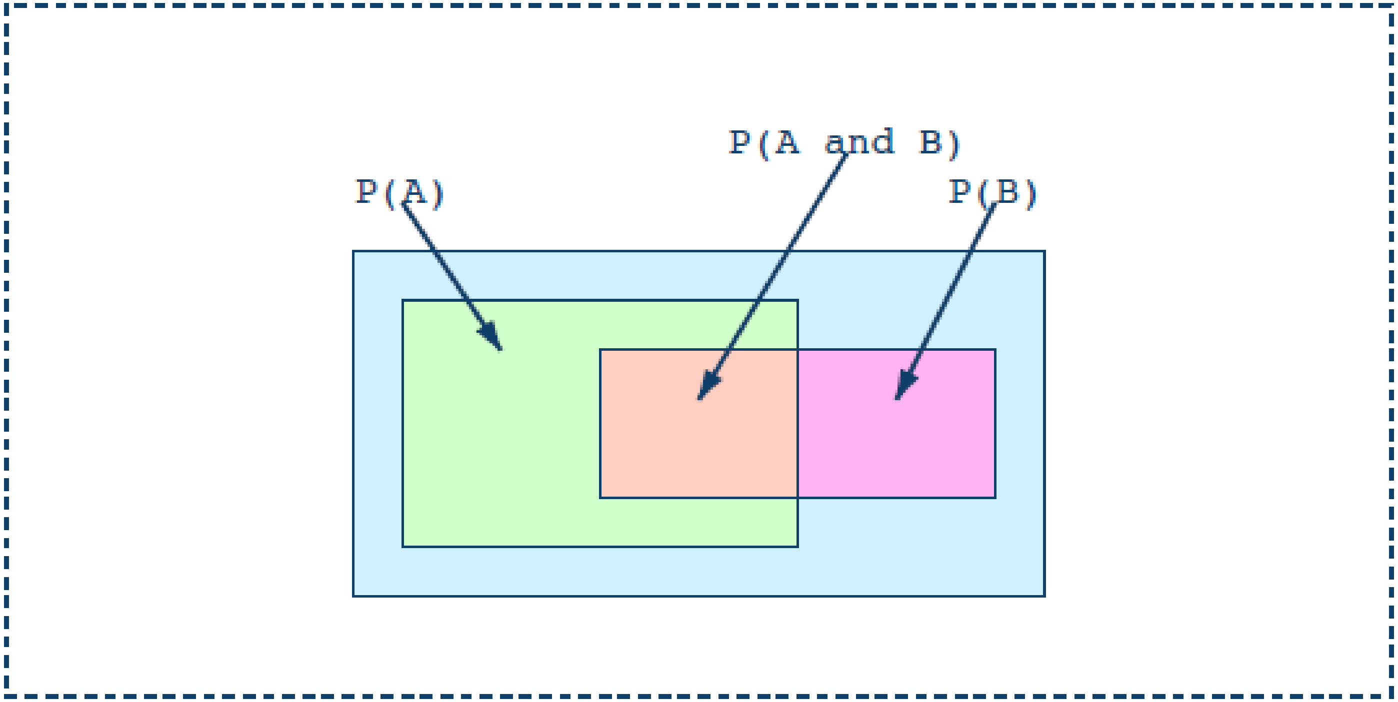
- 

Probability of A "the fraction of possible worlds in which A is true"

Probability Axioms

Define the whole set of possible worlds with the label TRUE and the empty set with FALSE:

- $0 \leq P(A) \leq 1$
- $P(TRUE) = 1; P(FALSE) = 0$
- $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$



Theorems from the Axioms (I)

- Using the axioms:
  - $P(TRUE) = 1$; $P(FALSE) = 0$
  - $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$

- Prove: $P(\sim A) = 1 - P(A)$

$$
\begin{aligned}
TRUE &= A \vee \sim A \\
P(TRUE) &= P(A \vee \sim A) \\
&= P(A) + P(\sim A) - P(A \wedge \sim A) \\
&= P(A) + P(\sim A) - P(FALSE) \\
1 &= P(A) + P(\sim A) - 0 \\
1 - P(A) &= P(\sim A)
\end{aligned}
$$

Theorems from the Axioms (II)

- Using the axioms:
  - $P(A = TRUE) = 1$; $P(A = FALSE) = 0$
  - $P(A \lor B) = P(A) + P(B) - P(A \land B)$

- Prove: $P(A) = P(A \land B) + P(A \land \sim B)$

$$
\begin{aligned}
A &= A \land TRUE \\
&= A \land (B \lor \sim B) \\
&= (A \land B) \lor (A \land \sim B) \\
P(A) &= P((A \land B) \lor (A \land \sim B)) \\
&= P(A \land B) + P(A \land \sim B) - P((A \land B) \land (A \land \sim B)) \\
&= P(A \land B) + P(A \land \sim B) - P(FALSE) \\
&= P(A \land B) + P(A \land \sim B)
\end{aligned}
$$

Multivalued Random Variables

Multivalued random variable A is a *random variable of arity* k if it can take on exactly one values out of $\{v_1, v_2, \ldots, v_k\}$.

We still have the probability axioms plus

- $P(A = v_i \wedge A = v_j) = 0$ if $i \neq j$
- $P(A = v_i \vee A = v_2 \vee \ldots \vee A = v_k) = 1$

Using those you can prove:

$$P(A = v_1 \vee A = v_2 \vee \ldots \vee A = v_i) = \sum_{j=1}^{i} P(A = v_j)$$

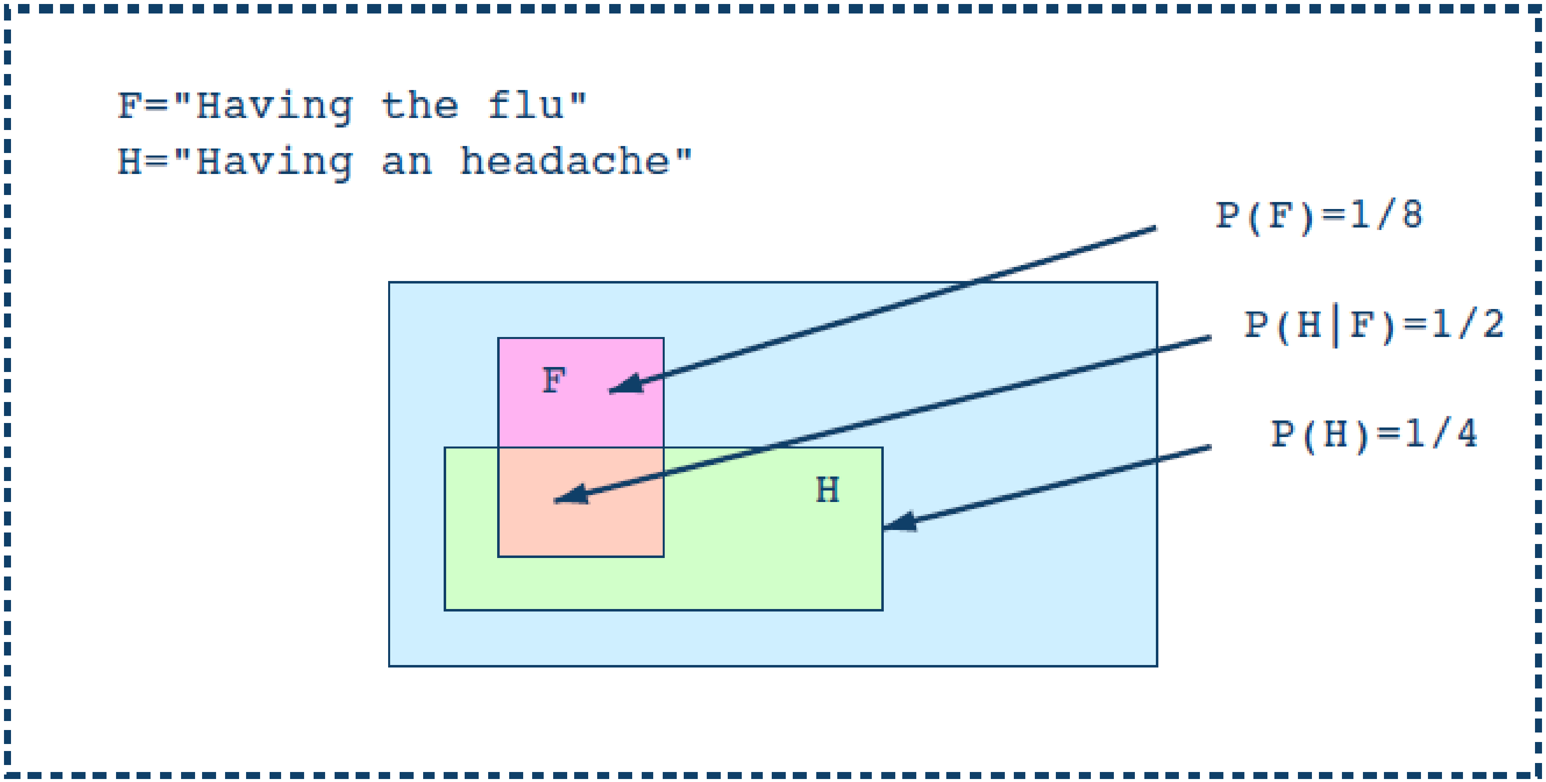
$$\sum_{j=1}^{k} P(A = v_j) = 1$$

$$P(B \wedge [A = v_1 \vee A = v_2 \vee \ldots \vee A = v_i]) = \sum_{j=1}^{i} P(B \wedge A = v_j)$$

$$P(B) = \sum_{j=1}^{k} P(B \wedge A = v_j)$$

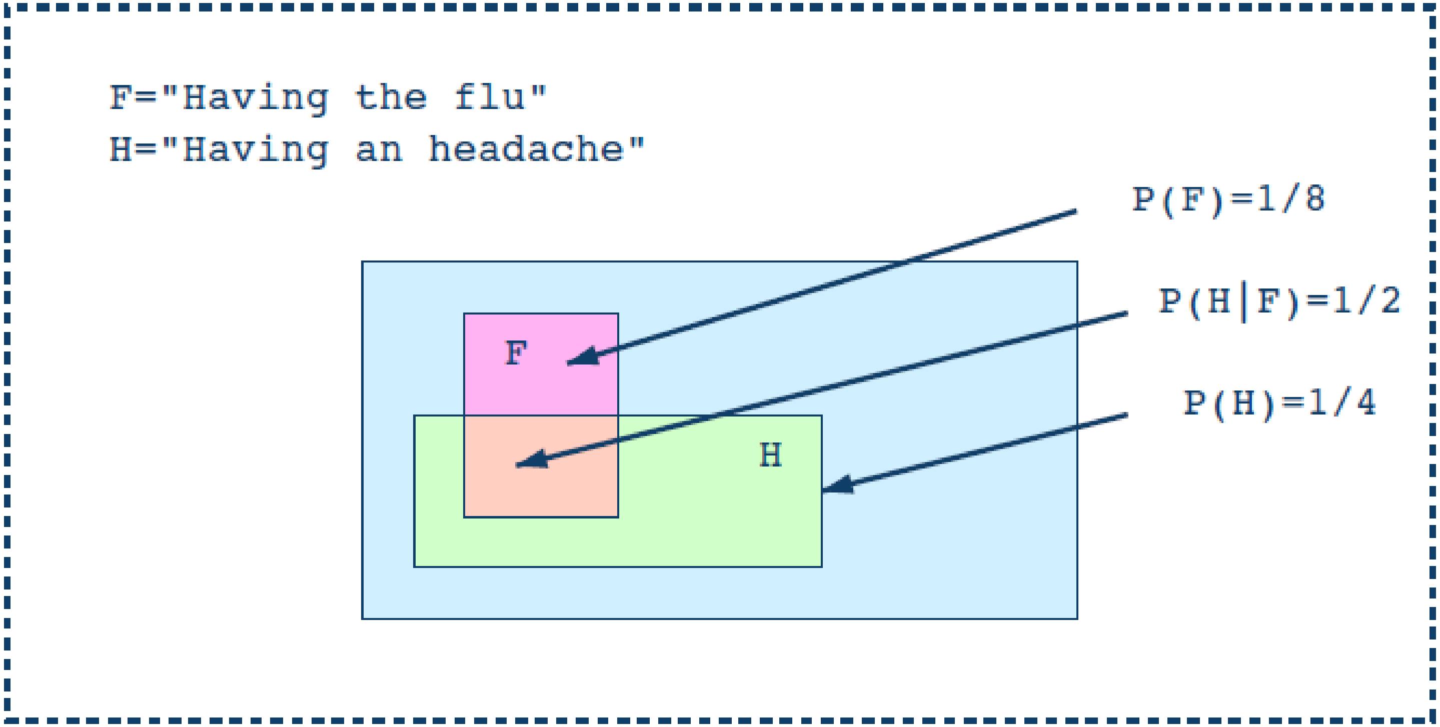
Conditional Probability

Probability of A **given** B**:** "the fraction of possible worlds in which B is true that also have A true"



$$P(H|F) = \frac{\text{Num. of worlds with F and H}}{\text{Num. worlds with F}} = \frac{P(H \wedge F)}{P(F)}$$

"Half of the flus are associated with headaches so I must have 50% chance of getting the flu".



- Is this reasoning correct?

$$P(F|H) = \frac{P(F \wedge H)}{P(H)} = \frac{P(H \wedge F)}{P(H)} = \frac{P(H|F) * P(F)}{P(H)} = \frac{1/2 * 1/8}{1/4} = 1/4$$

Theorems We Will Use

To make inference we can use

- *Chain Rule* $$P(A \wedge B) = P(A|B)P(B)$$
- *Bayes Theorem* $$P(A|B) = \frac{P(A \wedge B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

And several Bayes Theorem Generalizations

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

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

$$P(A = v_i|B) = \frac{P(B|A=v_i)P(A=v_i)}{\sum_{k=1}^{n_A} P(B|A=v_k)P(A=v_k)}$$

Independent Variables (I)

Independent variables: Assume A and B are boolean random variables; A and B are independent (denote it with $A \perp B$) if and only if:

$$P(A|B) = P(A)$$

- Using the definition:
  - $P(A|B) = P(A)$
- Proove: $P(A \wedge B) = P(A)P(B)$

$$\begin{aligned} P(A \wedge B) &= P(A|B)P(B) \\ &= P(A)P(B) \end{aligned}$$

Independent Variables (I)

Independent variables: Assume A and B are boolean random variables; A and B are independent (denote it with $A \perp B$) if and only if:

$$P(A|B) = P(A)$$

- Using the definition:
  - $P(A|B) = P(A)$
- Proove: $P(B|A) = P(B)$

$$\begin{aligned} P(B|A) &= P(A|B)P(B) / P(A) \\ &= P(A)P(B) / P(A) \\ &= P(B) \end{aligned}$$

Something for Computer Scientists!

Information and bits …

Your mission, if you decide to accept it:

"Transmit a set of independent random samples of X over a binary serial link."

1. Starring at X for a while, you notice that it has only four possible values: A, B, C, D
2. You decide to transmit the data encoding each reading with two bits:

A = 00, B = 01, C = 10, D = 11.

Mission Accomplished!

Information and fewer bits …

Your mission, if you decide to accept it:

"The previous code uses 2 bits for symbol. Knowing that the probabilities are not equal: P(X=A)=1/2, P(X=B)=1/4, P(X=C)=1/8, P(X=D)=1/8, invent a coding for your transmission that only uses 1.75 bits on average per symbol."

You decide to transmit the data encoding each reading with a different number of bits:

A = 0,B = 10,C = 110,D = 111.

Mission Accomplished!

Suppose X can have one of m values with probability

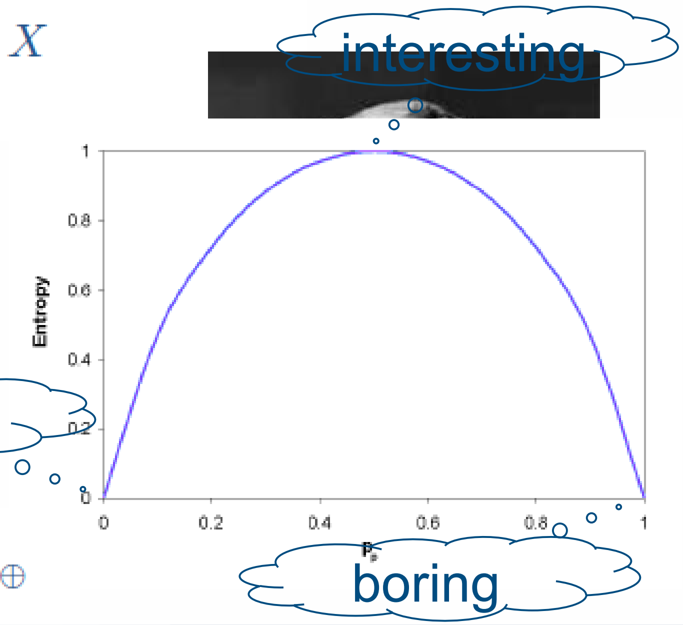
$$P(X = V_1) = p_1, \ldots, P(X = V_m) = p_m.$$

What's the smallest possible number of bits, on average, per symbol, needed to transmit a stream of symbols drawn from X's distribution?

$$\begin{aligned} H(X) &= -p_1 \log_2 p_1 - p_2 \log_2 p_2 - \ldots - p_m \log_2 p_m \\ &= -\sum_{j=1}^{m} p_j \log_2 p_j = \textit{Entropy of } X \end{aligned}$$

Simple binary example:

- X has 2 values $\oplus$ and $\ominus$
- $p_{\oplus}$ probability of $\oplus$
- $p_{\ominus} = 1 - p_{\oplus}$ probability of $\ominus$



$$H(X) = -p_{\ominus} \log_2 p_{\ominus} - p_{\oplus} \log_2 p_{\oplus}$$

Useful Facts on Logarithms

Just for you to know it might be useful to review a couple of formulas to be used in calculation with logarithms:

- $\ln x \times y = \ln x + \ln y$
- $\ln \frac{x}{y} = \ln x - \ln y$
- $\ln x^y = y \times \ln x$
- $\log_2 x = \frac{\ln x}{\ln 2} = \frac{\log_{10} x}{\log_{10} 2}$
- $\log_a x = \frac{}{\log_b a}$
- $\log_2 0 = -\infty$ (the formula is no good for a probability of 0)

Now we can practice with a simple example!

Specific Conditional Entropy

Suppose we are interested in predicting output Y from input X where

X = University subject

Y = Likes the movie "Gladiator"

From this data we can estimate

P(Y = Yes) = 0.5

P(X = Math) = 0.5

P(Y = Yes | X = History) = 0

We define Specific Conditional Entropy as

H(Y|X=v)

For instance in our case

H(Y|X=Math) = 1

H(Y|X=History) = 0

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
Hystory	No
Math	Yes

Conditional Entropy

Definition of Conditional Entropy H(Y|X):

$$\sum_j P(X=v_j)H(Y|X=v_j)$$

- *The average Y specific conditional entropy*
- *Expected number of bits to transmit Y if both sides will know the value of X*

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
Hystory	No
Math	Yes

v_j	$P(X=v_j)$	$H(Y\|X=v_j)$
Math	0.5	1
Hystory	0.25	0
CS	0.25	0

H(Y|X) = ?

$$H(Y|X) = 0.5 \times 1 + 0.25 \times 0 + 0.25 \times 0 = 0.5$$

Information Gain

"I must transmit Y on a binary serial line. How many bits on average would it save me if both ends of the line knew X?"

The answer is ***Information Gain***

$$
\begin{aligned}
IG(Y|X) &= H(Y) - H(Y|X) \\
&= 1 - 0.5 = 0.5
\end{aligned}
$$

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
Hystory	No
Math	Yes

Information Gain measures "information" provided by X to predict Y

Relative Information Gain

"I must transmit Y on a binary serial line. What fraction of bits on average would it save me if both ends of the line knew X?"

The answer is ***Relative Information Gain***

$$
\begin{aligned}
RIG(Y|X) &= (H(Y) - H(Y|X))/H(Y) \\
&= (1 - 0.5)/1 = 0.5
\end{aligned}
$$

What this all has to do with data mining?

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
Hystory	No
Math	Yes

The Shortest Decision Tree

Your mission, if you decide to accept it:

"Predict whether or not someone is going to live past 80 years."

From historical data you might find:

- IG(LongLife | HairColor) = 0.01
- IG(LongLife | Smoker) = 0.2
- IG(LongLife | Gender) = 0.25
- IG(LongLife | LastDigitOfSSN) = 0.00001

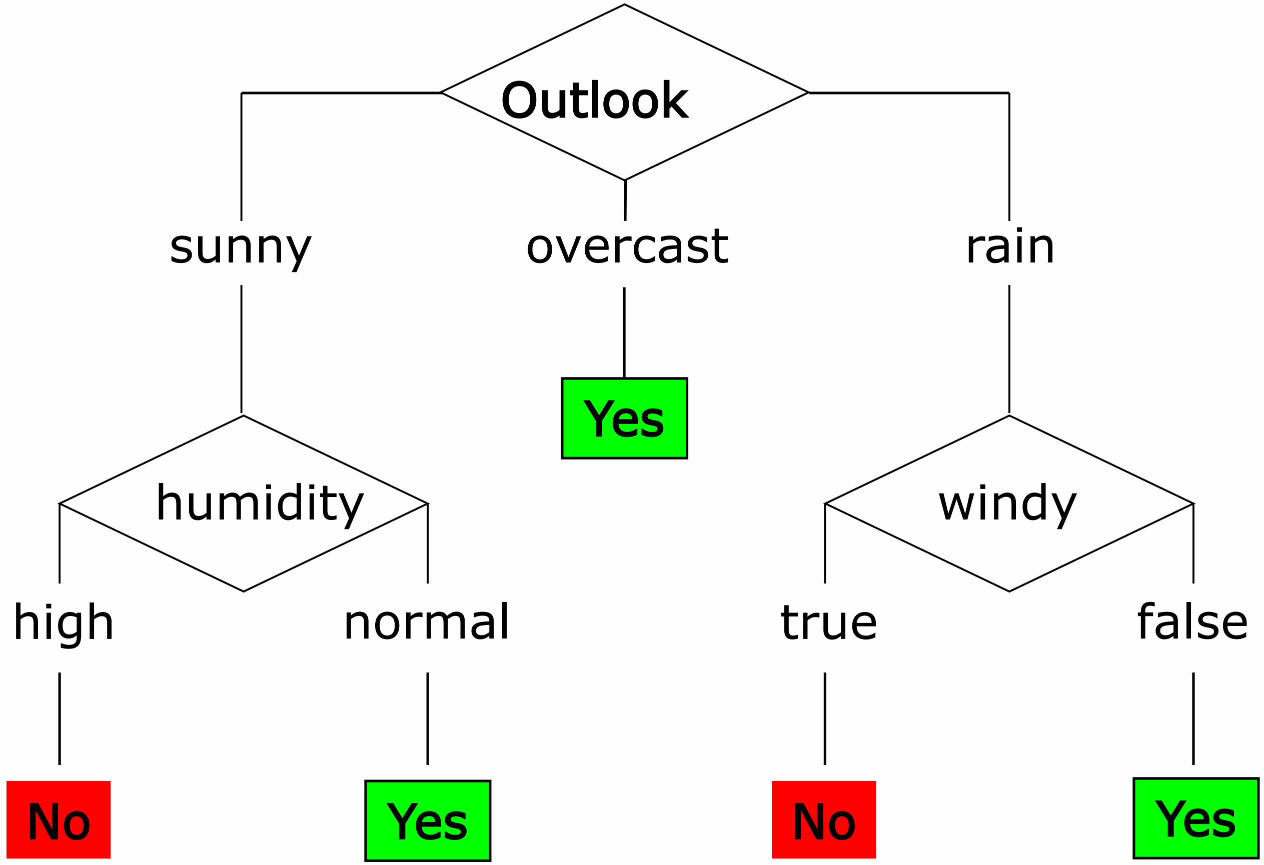
What you should ask having one shot option?

What is a Decision Tree?

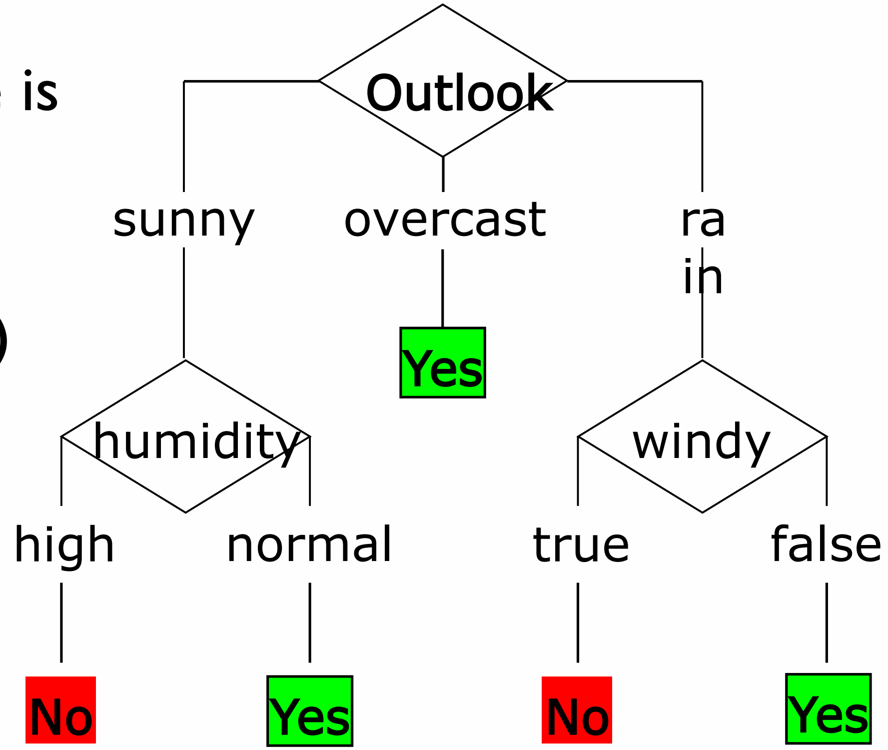
The Weather (or Golf) Dataset

Outlook	Temp	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

A Decision Tree for Playing Golf …



Decision Trees in a Nutshell



- An internal (i.e., not leaf) node is a test on an attribute
- A branch represents the test outcome (e.g., outlook=windy)
- A leaf node represents a class label or class label distribution

Noteworthy facts

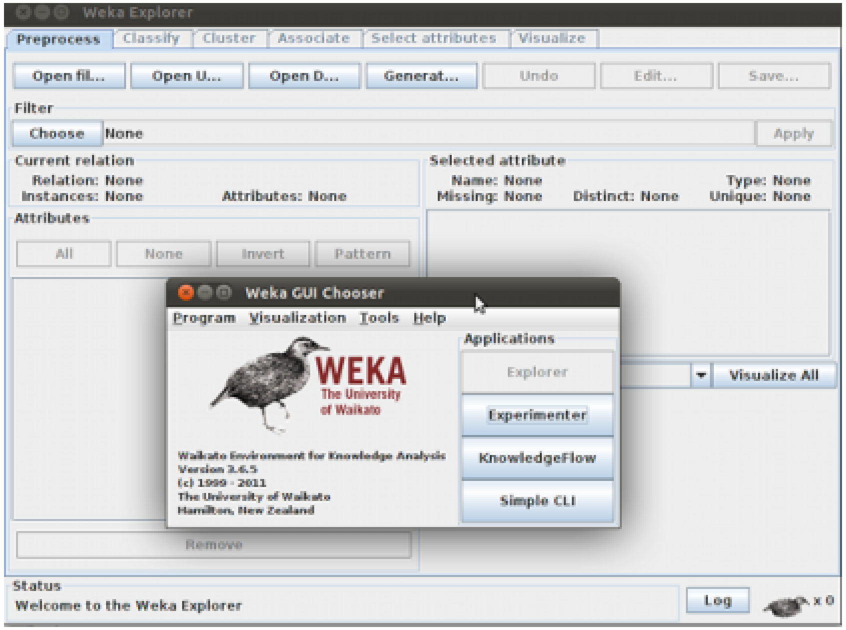
- At each node, one attribute is chosen to split training examples into classes as much distinct as possible
- Once an attribute has been used for splitting it is not reused
- New cases are classified following a paths from root to leaves

Building the Tree with Weka

What is Weka?

Weka: a collection of machine learning algorithms for data mining:

- Algorithms can either be applied directly or called from Java
- Weka contains tools for:
  - Data pre-processing
  - Classification
  - Regression
  - Clustering
  - Association rules
  - Visualization



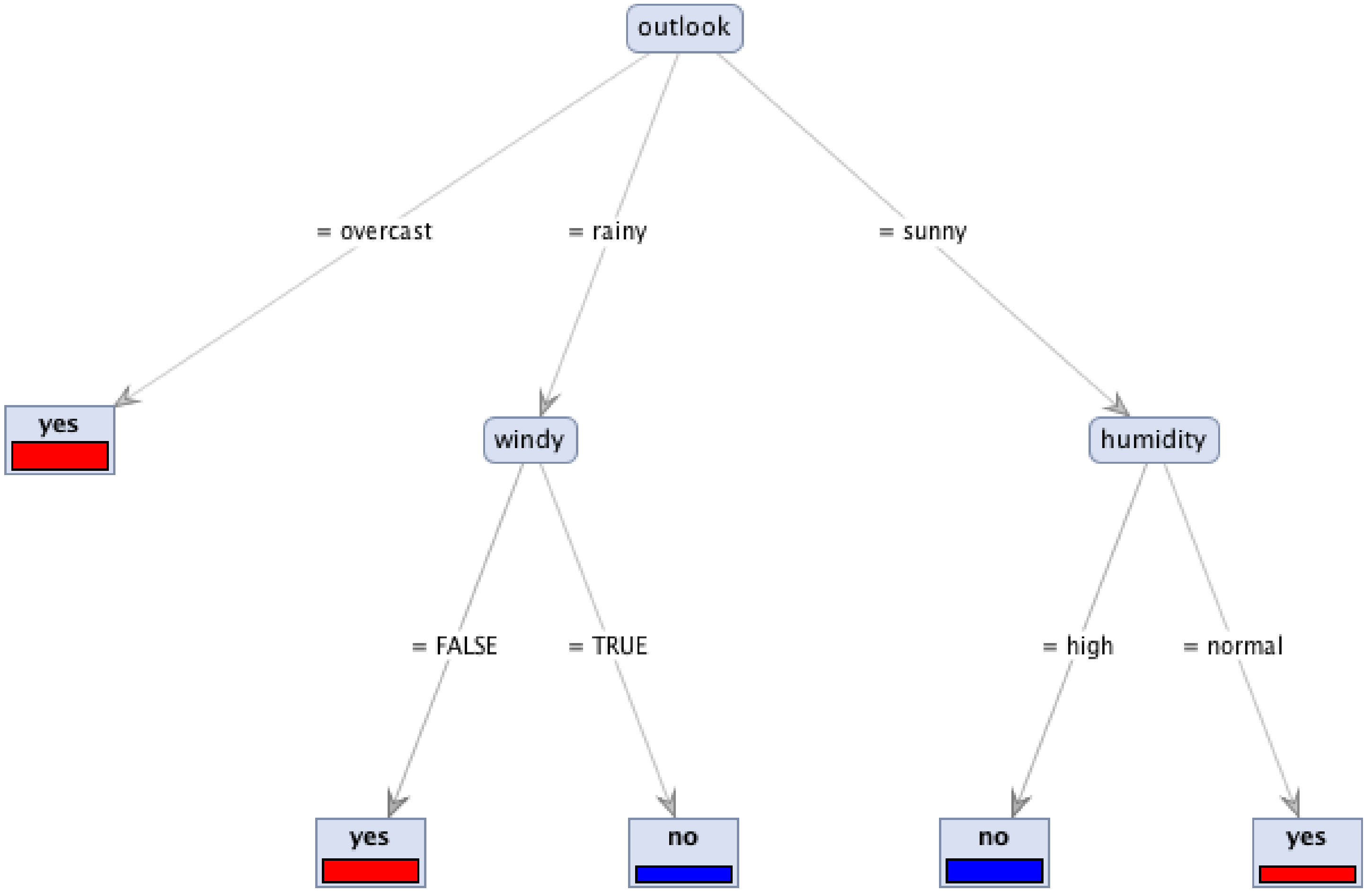
- Weka is open source software:

http://weka.waikato.ac.nz

Decision Tree Representations (Text)


```
outlook = overcast: yes {no=0, yes=4}
outlook = rainy
|   windy = FALSE: yes {no=0, yes=3}
|   windy = TRUE: no {no=2, yes=0}
outlook = sunny
|   humidity = high: no {no=3, yes=0}
|   humidity = normal: yes {no=0, yes=2}
```

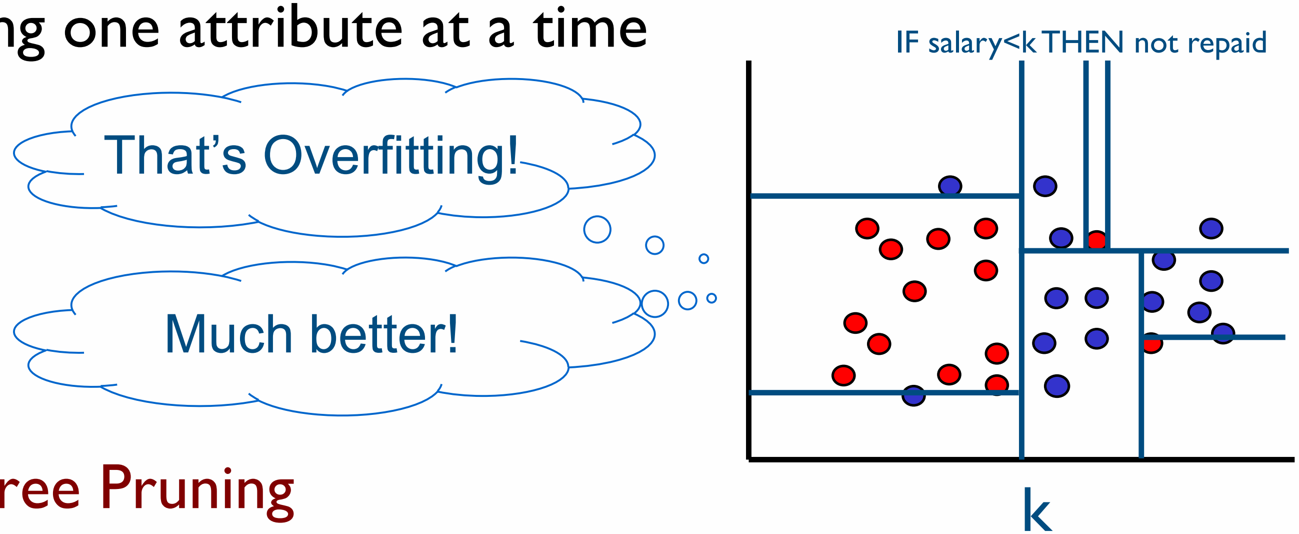
Decision Tree Representations (Graphical)



Building Decision Trees

Building Decision Trees

- Top-down Tree Construction
  - Initially, all the training examples are at the root
  - Then, the examples are recursively partitioned, by choosing one attribute at a time



- Bottom-up Tree Pruning
  - Remove subtrees or branches, in a bottom-up manner, to improve the estimated accuracy on new cases.

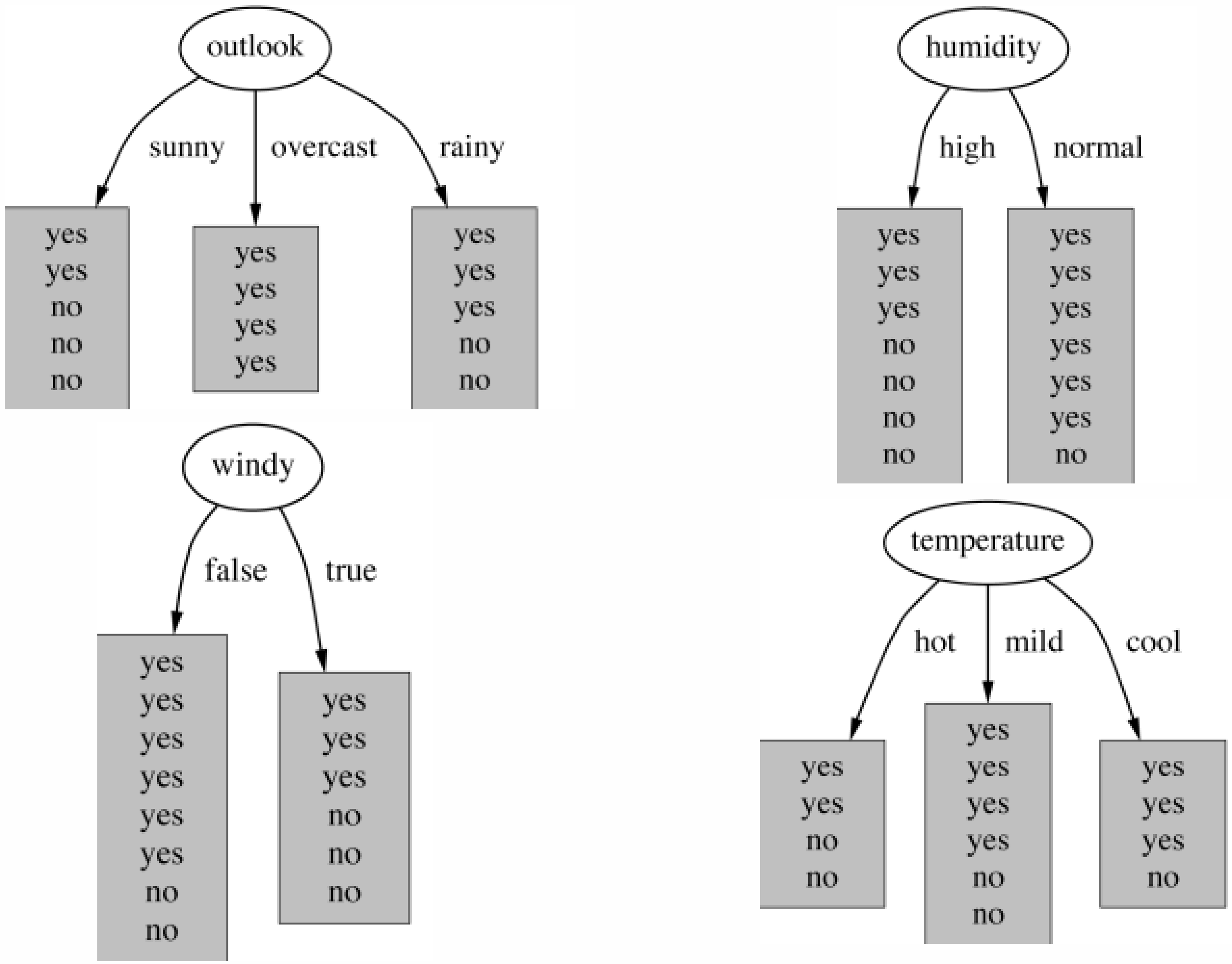
How is the Splitting Attribute Determined?

Which Attribute for Splitting?

- At each node, available attributes are evaluated on the basis of separating the classes of the training examples
- A purity or impurity measure is used for this purpose
- Typical goodness functions:
  - information gain (ID3)
  - information gain ratio (C4.5)
  - gini index (CART)

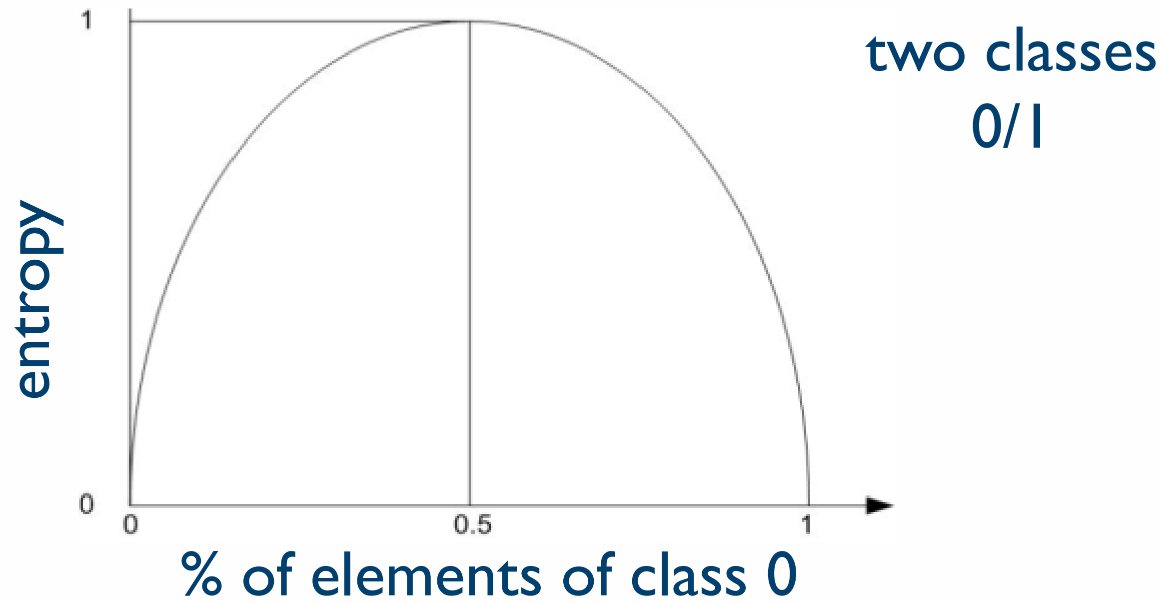
- Information Gain: increases with the average purity of the subsets that an attribute produces
- Splitting Strategy: choose the attribute that results in greatest information gain

Which Attribute Should We Select?



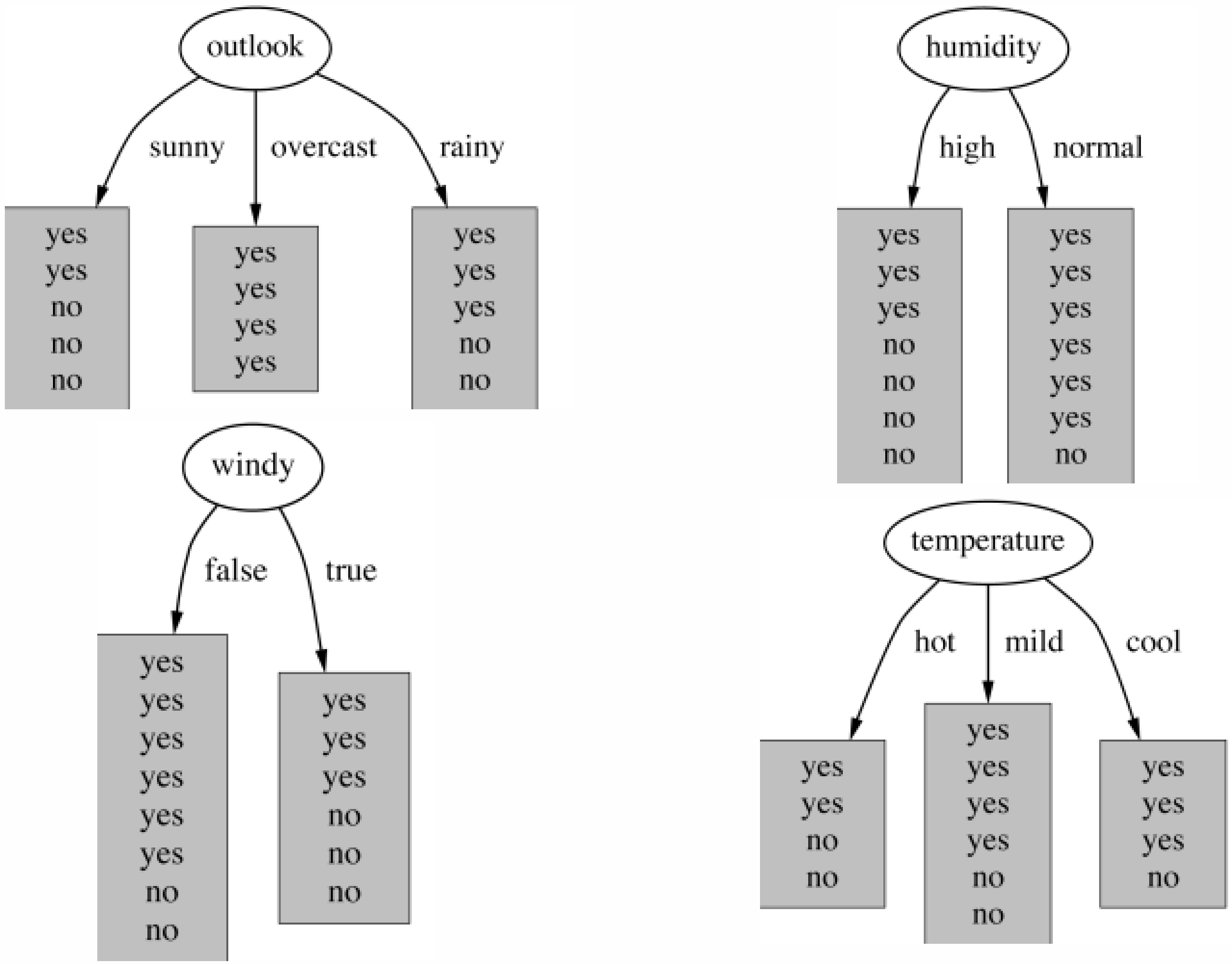
Computing Information Reminder

- Given a probability distribution, the info required to predict an event is the distribution's entropy
- Entropy gives the information required in bits (this can involve fractions of bits!)



$$entropy(p_1, p_2, \ldots, p_n) = -p_1 \log p_1 - p_2 \log p_2 \ldots - p_n \log p_n$$

Which Attribute Should We Select?



The Attribute "outlook"

- "outlook" = "sunny"

$$info([2,3]) = entropy(2/5, 3/5) = 0.971$$

- "outlook" = "overcast"

$$info([4,0]) = entropy(1,0) = 0.000$$

- "outlook" = "rainy"

$$info([3,2]) = entropy(3/5, 2/5) = 0.971$$

- Expected information for attribute "outlook"

$$\begin{aligned} info([2,3][4,0][3,2]) &= 5/14 \times 0.971 + \\ & \quad 4/14 \times 0 + 5/14 \times 0.971 \end{aligned}$$

Information Gain

- Difference between the information before split and the information after split

$$gain(A) = info(D) - info_A(D)$$

- The information before the split, info(D), is the entropy,

$$info(D) = -p_1 \log p_1 - \ldots - p_n \log p_n$$

- The information after the split using attribute A is computed as the weighted sum of the entropies on each split, given n splits,

$$info_A(D) = \frac{|D_1|}{|D|} info(D_1) + \ldots + \frac{|D_n|}{|D|} info(D_n)$$

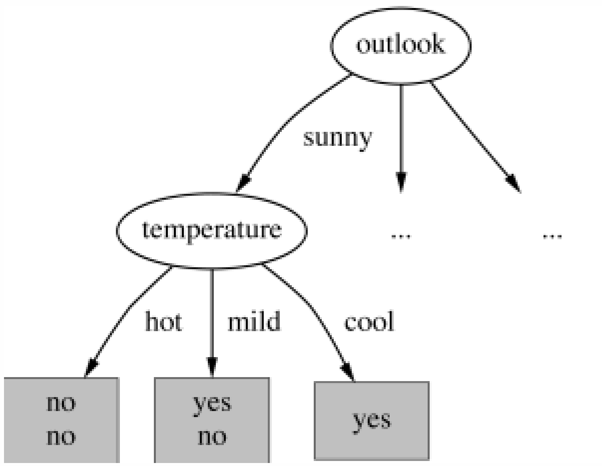
Information Gain

- Difference between the information before split and the information after split

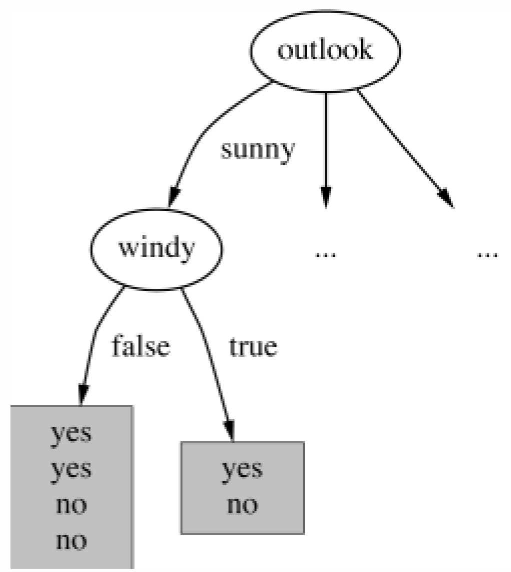
$$\begin{aligned} gain(outlook) &= info([9,5]) - info([2,3][4,0][3,2]) \\ &= 0.940 - 0.693 \\ &= 0.247 \end{aligned}$$

- Information gain for the attributes from the weather data:
  - gain(“outlook”)=0.247 bits
  - gain(“temperature”)=0.029 bits
  - gain(“humidity”)=0.152 bits
  - gain(“windy”)=0.048 bits

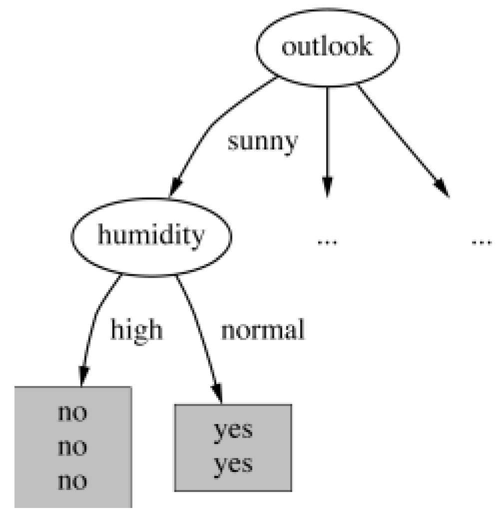
Going Further



$$gain(temperature) = 0.571$$



$$gain(windy) = 0.020$$

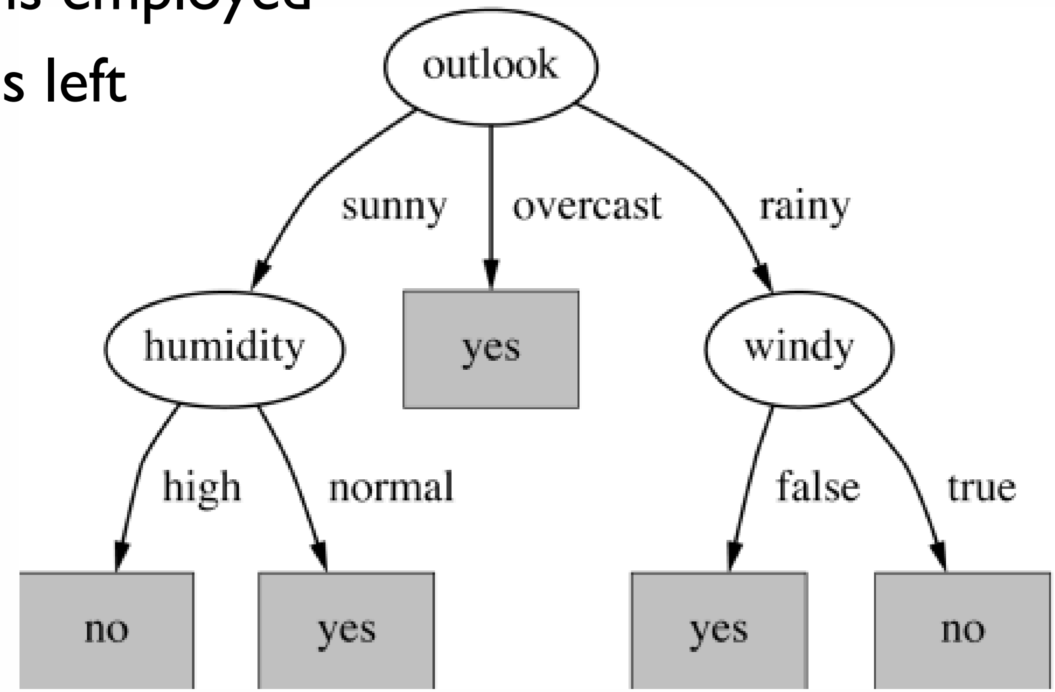


$$gain(humidity) = 0.971$$

When should stop splitting?

When to Stop Splitting?

- All the leaves samples belong to the same class
- Splitting stops when data can not be split any further
  - There are no remaining attributes for further partitioning, then majority voting is employed
  - There are no samples left



What if Info Gain is Zero?

Consider the following example:

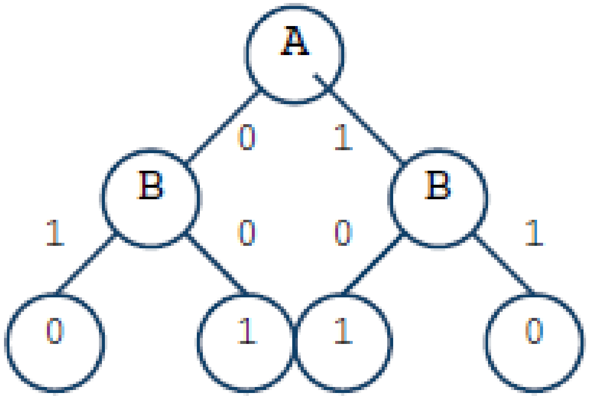
Y = A xor B

$$
\begin{aligned}
H(Y) &= 1 \\
H(Y|A) &= P(\bar{A})H(Y|\bar{A})P(A)H(Y|A) \\
&= 1/2 \times 1 + 1/2 \times 1 = 1 \\
H(Y|B) &= P(\bar{B})H(Y|\bar{B})P(B)H(Y|B) \\
&= 1/2 \times 1 + 1/2 \times 1 = 1
\end{aligned}
$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Should recursion be stopped?

- if I stop recursion
  - randomly predict one of the output
  - 50% Error Rate
- If random split when info gain is zero
  - Then we get 0% error rate



Top Down Induction of Decision Trees


```
function TDIDT(S)     // S, a set of labeled examples

Tree = new empty node
if  (samples have same class c) OR (no further possible splitting)
then                       // new leaf labeled with majority class c
     Label(Tree) = c
else                       // new decision node
    (A,T) = FindBestSplit(S)
    foreach test t in T do
       S_t = all examples that satisfy t
       Node_t = TDIDT(S_t)
       AddEdge(Tree -> Node_t)
    endfor
endif
return Tree
```

What if Attributes are Numerical?

The Weather Dataset (Numerical)

Outlook	Temp	Humidity	Windy	Play
Sunny	85	85	False	No
Sunny	80	90	True	No
Overcast	83	78	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	80	True	No

The Temperature Attribute

- First, sort the temperature values, including the class labels
- Then, check all the cut points and choose the one with the best information gain

64	65	68	69	70	71	72	72	75	75	80	81	83	85
Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No

E.g. temperature $<$ 71.5: yes/4, no/2
temperature $\geq$ 71.5: yes/5, no/3

Info([4,2],[5,3]) = 6/14 info([4,2]) + 8/14 info([5,3])
= 0.939

- Place split points halfway between values

Can evaluate all split points in one pass!

Information Gain for Humidity

Humidity	Play	# of Yes	% of Yes	# of No	% of No	Weight	Entropy Left	# of Yes	% of Yes	# of No	% of No	Weight	Entropy Right	Information Gain
65	Yes	1	100.00%	0	0.00%	7.14%	0.00	8.00	0.62	5.00	0.38	92.86%	0.96	0.0477
70	No	1	50.00%	1	50.00%	14.29%	1.00	8.00	0.67	4.00	0.33	85.71%	0.92	0.0103
70	Yes	2	66.67%	1	33.33%	21.43%	0.92	7.00	0.64	4.00	0.36	78.57%	0.95	0.0005
70	Yes	3	75.00%	1	25.00%	28.57%	0.81	6.00	0.60	4.00	0.40	71.43%	0.97	0.0150
75	Yes	4	80.00%	1	20.00%	35.71%	0.72	5.00	0.56	4.00	0.44	64.29%	0.99	0.0453
78	Yes	5	83.33%	1	16.67%	42.86%	0.65	4.00	0.50	4.00	0.50	57.14%	1.00	0.0903
80	Yes	6	85.71%	1	14.29%	50.00%	0.59	3.00	0.43	4.00	0.57	50.00%	0.99	0.1518
80	Yes	7	87.50%	1	12.50%	57.14%	0.54	2.00	0.33	4.00	0.67	42.86%	0.92	0.2361
80	No	7	77.78%	2	22.22%	64.29%	0.76	2.00	0.40	3.00	0.60	35.71%	0.97	0.1022
85	No	7	70.00%	3	30.00%	71.43%	0.88	2.00	0.50	2.00	0.50	28.57%	1.00	0.0251
90	No	7	63.64%	4	36.36%	78.57%	0.95	2.00	0.67	1.00	0.33	21.43%	0.92	0.0005
90	Yes	8	66.67%	4	33.33%	85.71%	0.92	1.00	0.50	1.00	0.50	14.29%	1.00	0.0103
95	No	8	61.54%	5	38.46%	92.86%	0.96	1.00	1.00	0.00	0.00	7.14%	0.00	0.0477
96	Yes	9	64.29%	5	35.71%	100.00%	0.94	0.00	0.00	0.00	0.00	0.00%	0.00	0.0000

$$IG(Y|X:t) = H(Y) - H(Y|X:t)$$

$$H(Y|H:t) \quad = \quad H(Y|X \leq t)P(X \leq t) + H(Y|X > t)P(X > t)$$

$$IG^*(Y|X) = \max_t IG(Y|X:t)$$

The Information Gain for Humidity

Humidity	Play
65	Yes
70	No
70	Yes
70	Yes
75	Yes
78	Yes
80	Yes
80	Yes
80	No
85	No
90	No
90	Yes
95	No
96	Yes

sort the attribute values

Humidity	Play
65	Yes
70	No
70	Yes
70	Yes
75	Yes
78	Yes
80	Yes
80	Yes
80	No
85	No
90	No
90	Yes
95	No
96	Yes

compute the gain for every possible split

what is the information gain if we split here?

What if Attributes are Missing?

Missing Values …

- Discarding examples with missing values
  - Simplest approach
  - Allows the use of unmodified data mining methods
  - Only practical if there are few examples with missing values. Otherwise, it can introduce bias
- Convert the missing values into a new value
  - Use a special value for it
  - Add an attribute that indicates if value is missing or not
  - Greatly increases the difficulty of the data minig process
- Imputation methods
  - Assign a value to the missing one, based on the dataset.
  - Use the unmodified data mining methods.

Other purity measures?

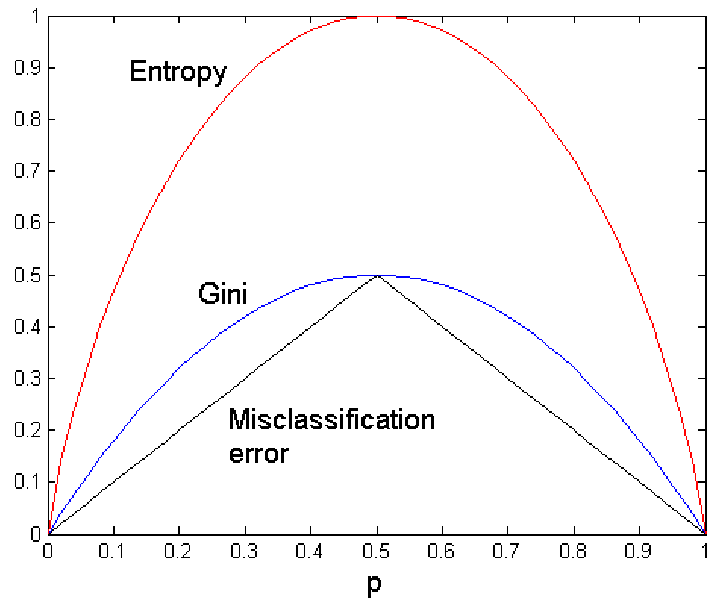
The Gini Index

- The gini index, for a data set T contains examples from n classes, is defined as

$$gini(T) = 1 - \sum_{j=1}^{n} p_j^2$$

where p_j is the relative frequency of class j in T

- gini(T) is minimized if the classes in T are skewed.



The Gini Index

- If a data set D is split on A into two subsets D_1 and D_2, then,

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- The reduction of impurity is defined as,

$$\Delta gini(A) = gini(D) - gini_A(D)$$

- The attribute provides the smallest gini splitting D over A (or the largest reduction in impurity) is chosen to split the node (need to enumerate all the possible splitting points for each attribute)

The Gini Index: Example

- D has 9 tuples labeled "yes" and 5 labeled "no"

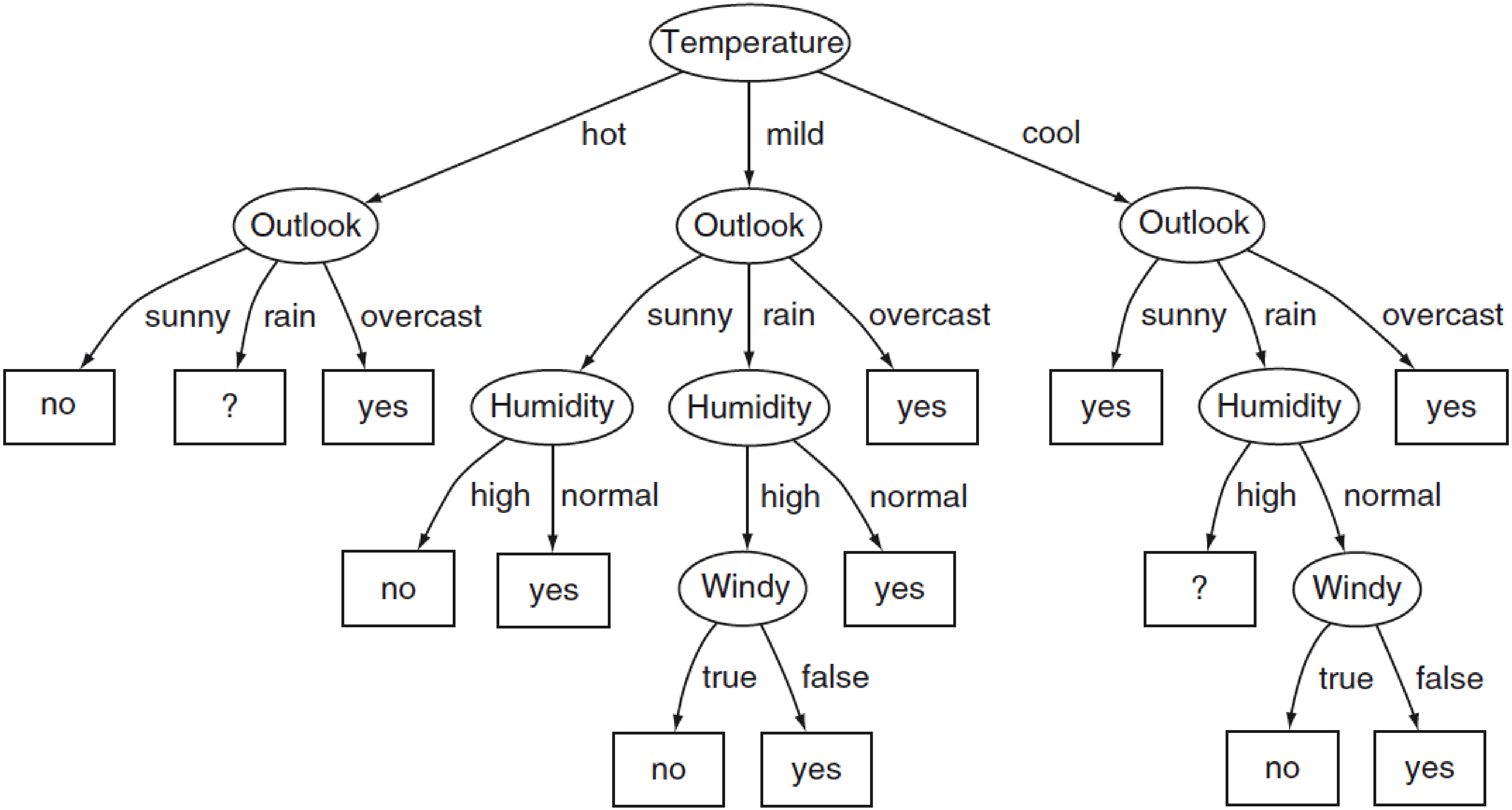
$$gini(D) = 1 - (\frac{9}{14})^2 - (\frac{5}{14})^2 = 0.459$$

- Suppose the attribute income partitions D into 10 in D1 branching on low and medium and 4 in D2

$$\begin{aligned} gini(D)_{\{l,m\}} &= \left(\frac{10}{14}\right) gini(D_1) + \left(\frac{4}{14}\right) gini(D_2) \\ &= \left(\frac{10}{14}\right) \left(1 - (\frac{6}{10})^2 - (\frac{4}{10})^2\right) + \\ &+ \left(\frac{4}{14}\right) \left(1 - (\frac{1}{4})^2 - (\frac{3}{4})^2\right) = 0.450 \end{aligned}$$

We Can Always Build a 100% Accurate Tree…

The Perfect Fit …

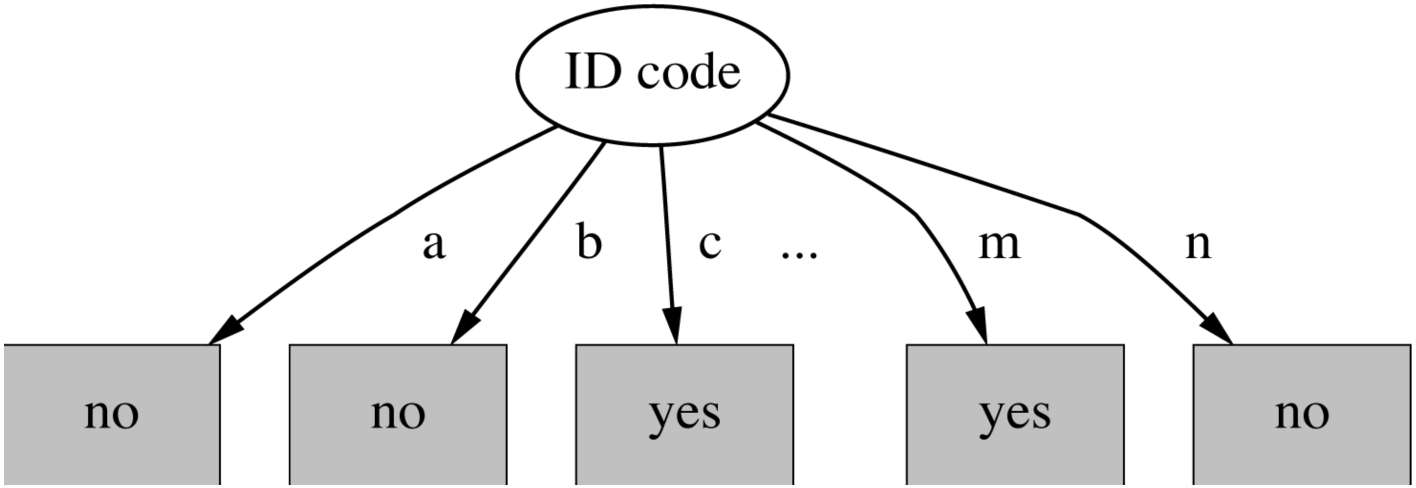


Another Version of the Weather Dataset

ID Code	Outlook	Temp	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

Decision Tree for the New Dataset

- Entropy for splitting using “ID Code” is zero, since each leaf node is “pure”



- Information Gain is thus maximal for ID code

Highly-Branching Attributes

- Attributes with a large number of values are problematic

Examples:

id, primary keys, or almost primary key attributes

- Subsets are likely to be pure if there is a large number of values
- Information Gain is biased towards choosing attributes with a large number of values
- This may result in overfitting (selection of an attribute that is non-optimal for prediction)

Information Gain Ratio
(Different from Relative Information Gain)

Information Gain Ratio

- Modification of the Information Gain that reduces the bias toward highly-branching attributes

- **Information Gain Ratio** should be
  - Large when data is evenly spread
  - Small when all data belong to one branch

- **Information Gain Ratio**:
  - takes number and size of branches into account when choosing an attribute
  - corrects *Information Gain* by taking the *Intrinsic Information* of a split into account

Information Gain Ratio & Intrinsic Information

- Intrinsic information (i.e., entropy)

$$IntrinsicInfo(S, A) = -\sum \frac{|S_i|}{|S|} \log \frac{|S_i|}{|S|}$$

computes the entropy of distribution of instances into branches

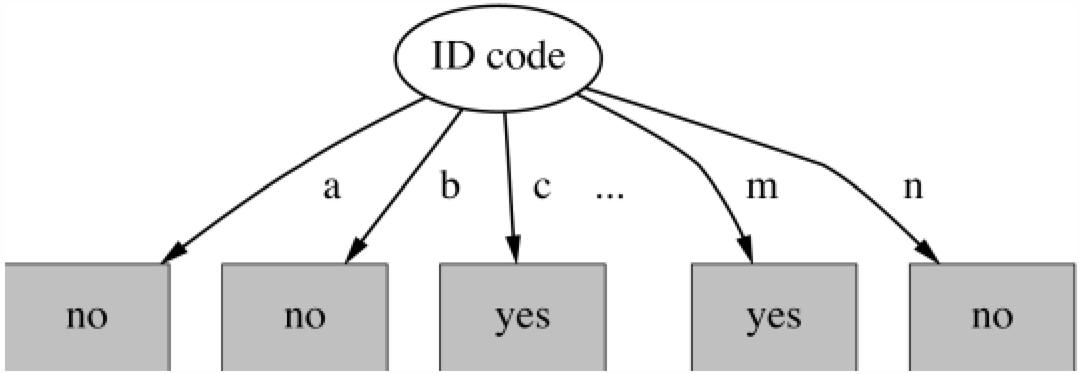
- Information Gain Ratio normalizes Information Gain by entropy

$$GainRatio(S, A) = \frac{Gain(S, A)}{IntrinsicInfo(S, A)}$$

Computing the Information Gain Ratio

- The intrinsic information for ID code is

$$info([1,1,\ldots 1]) = 14 \times (-1/14 \times \log 1/14) = 3.807$$



$$GainRatio(ID\_code) = \frac{0.940}{3.807} = 0.246$$

- Importance of attribute decreases as intrinsic information gets larger

Information Gain Ratio for Weather Data

Outlook		**Temperature**	
Info:	0.693	Info:	0.911
Gain: 0.940-0.693	0.247	Gain: 0.940-0.911	0.029
Split info: info([5,4,5])	1.577	Split info: info([4,6,4])	1.362
Gain ratio: 0.247/1.577	0.156	Gain ratio: 0.029/1.362	0.021

Humidity		**Windy**	
Info:	0.788	Info:	0.892
Gain: 0.940-0.788	0.152	Gain: 0.940-0.892	0.048
Split info: info([7,7])	1.000	Split info: info([8,6])	0.985
Gain ratio: 0.152/1	0.152	Gain ratio: 0.048/0.985	0.049

More on Information Gain Ratio

- "Outlook" still comes out top, however "ID code" has greater Information Gain Ratio ☹

  - The standard fix is an ad-hoc test to prevent splitting on that type of attribute

- First, consider attributes with greater than average Information Gain; then, compare them using the Information Gain Ratio

  - Information Gain Ratio overcompensates and may choose an attribute because its intrinsic information is very low

No free lunch!!