CSCI 347, Data Mining

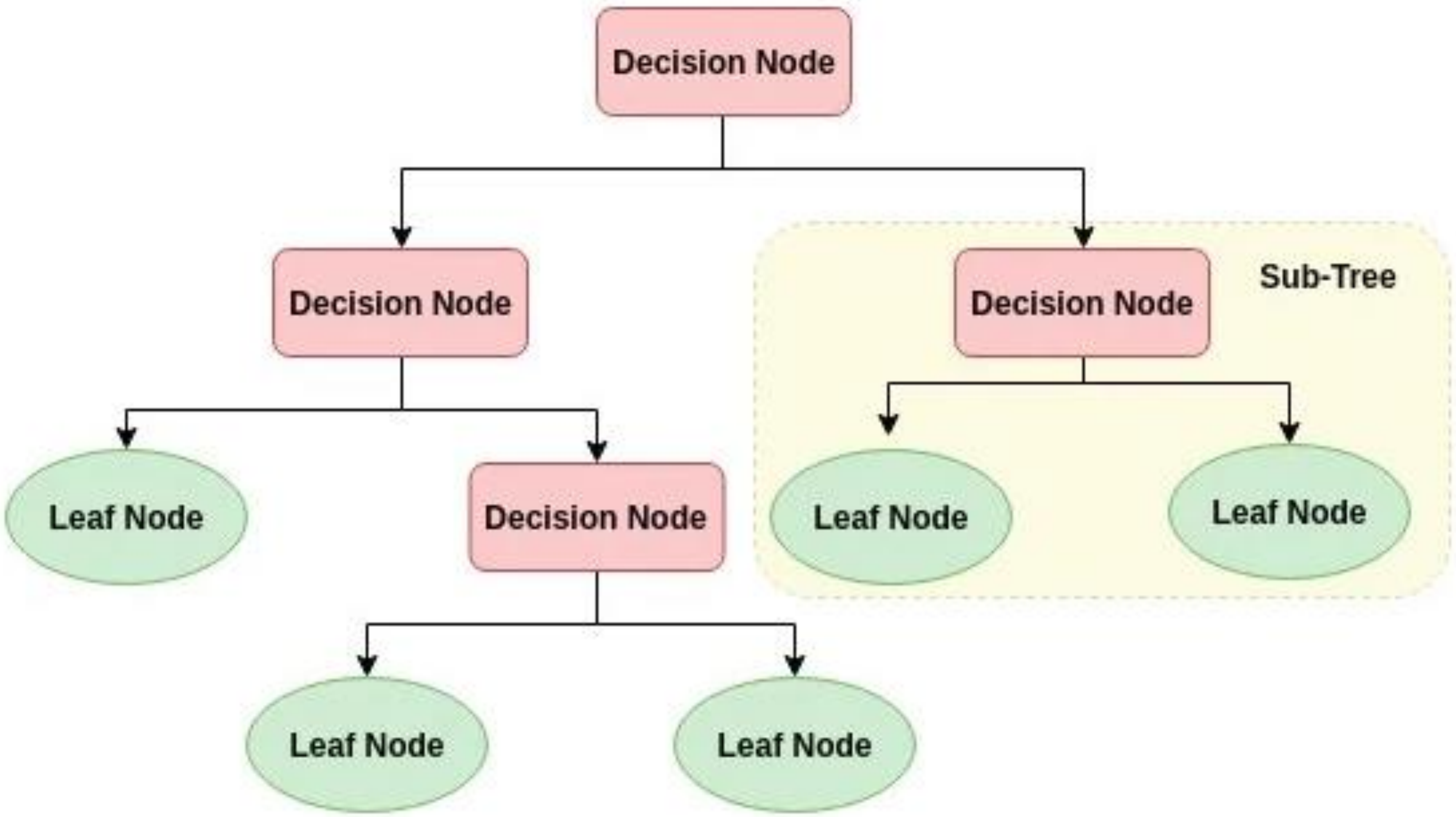
Decision Trees

Decision Trees

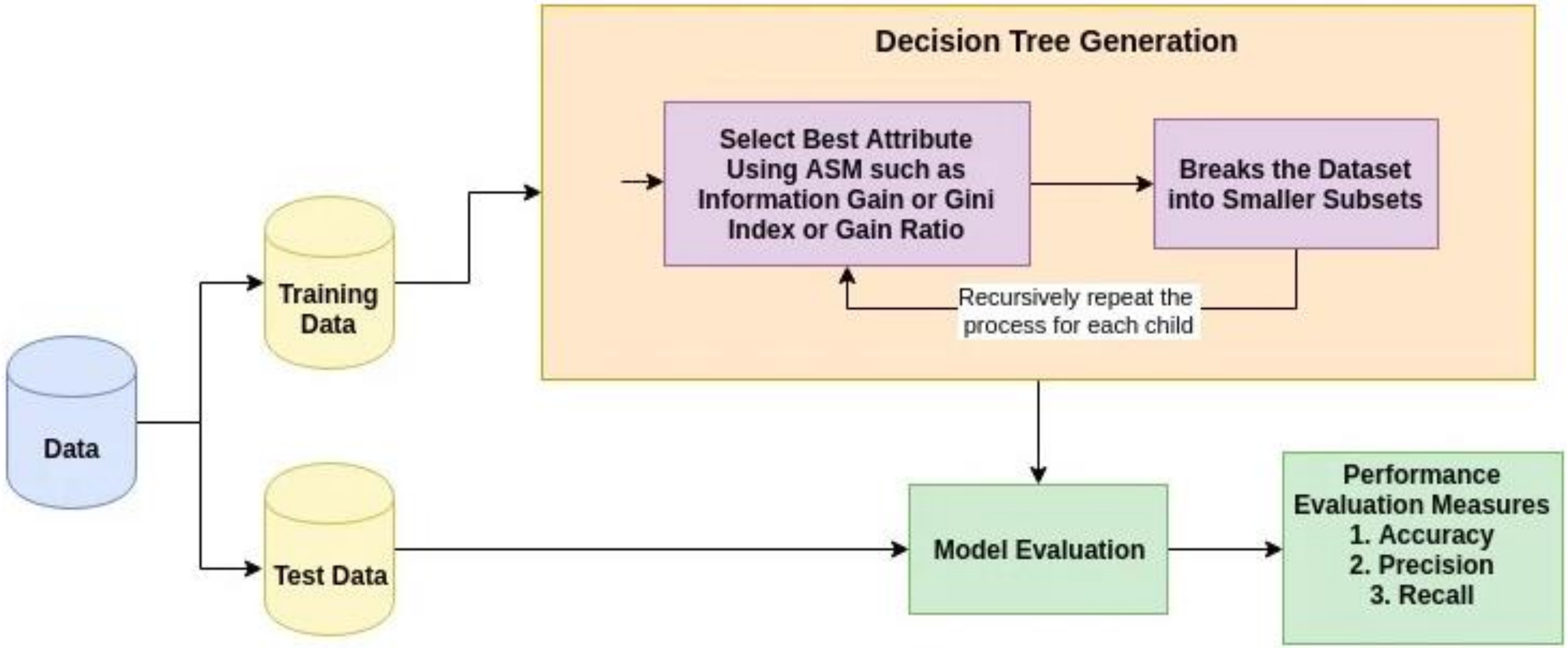
Decision Trees:

- Supervised learning
- Tree built by splitting on attributes, where the number of children is usually equal to the number of values for the attributes
- The leaves tell the classification
- Attributes typically only get tested once
- Divide and conquer approach, recursive – determine best attribute to serve as root node, split the records based on the attribute values, apply the algorithm to each subset of the records
- Attributes can be numeric or nominal. Class value is typically nominal.

Decision Trees

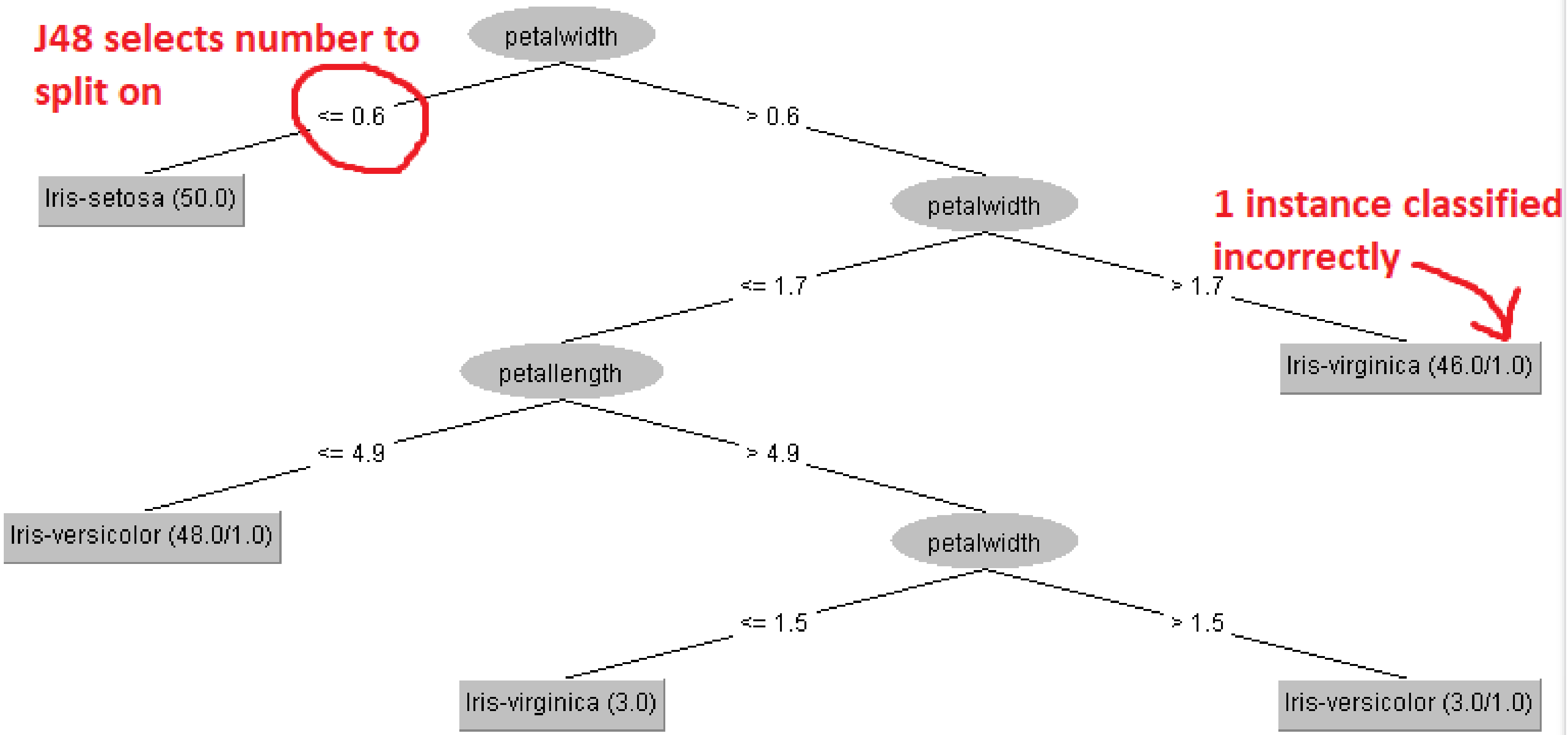


Process

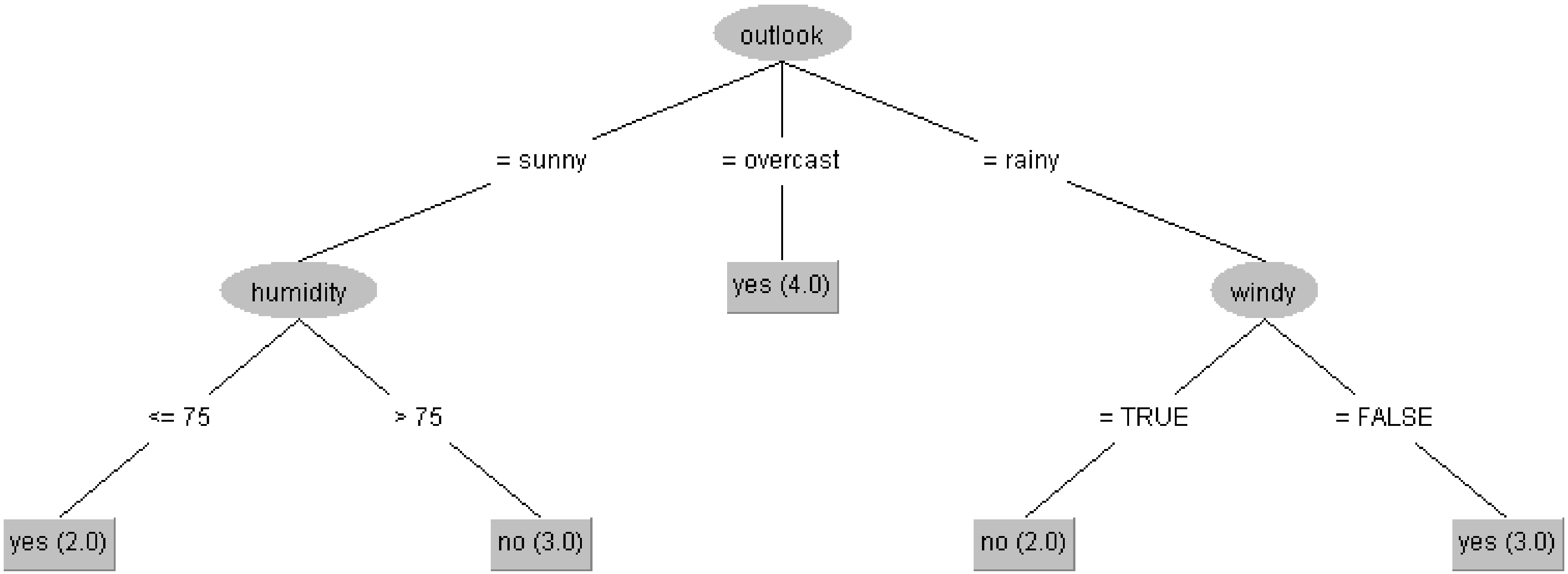


Decision Tree Generation
Select Best Attribute Using ASM such as Information Gain or Gini Index or Gain Ratio
Breaks the Dataset into Smaller Subsets
Recursively repeat the process for each child
Data
Training Data
Test Data
Model Evaluation
Performance Evaluation Measures
1. Accuracy
2. Precision
3. Recall

Iris Dataset Weka's J48 Decision Tree



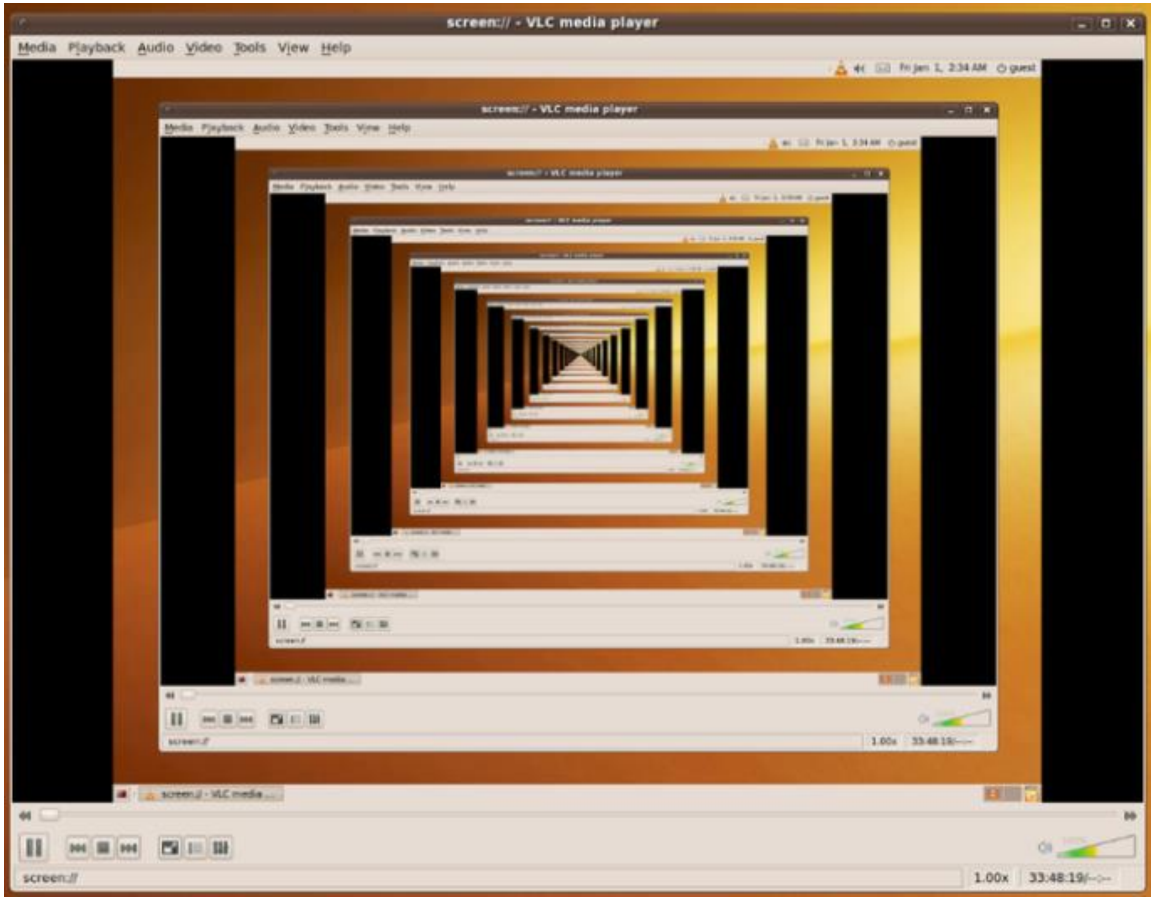
Weather Dataset Weka's J48 Decision Tree



Decision Trees Advantages/Disadvantages

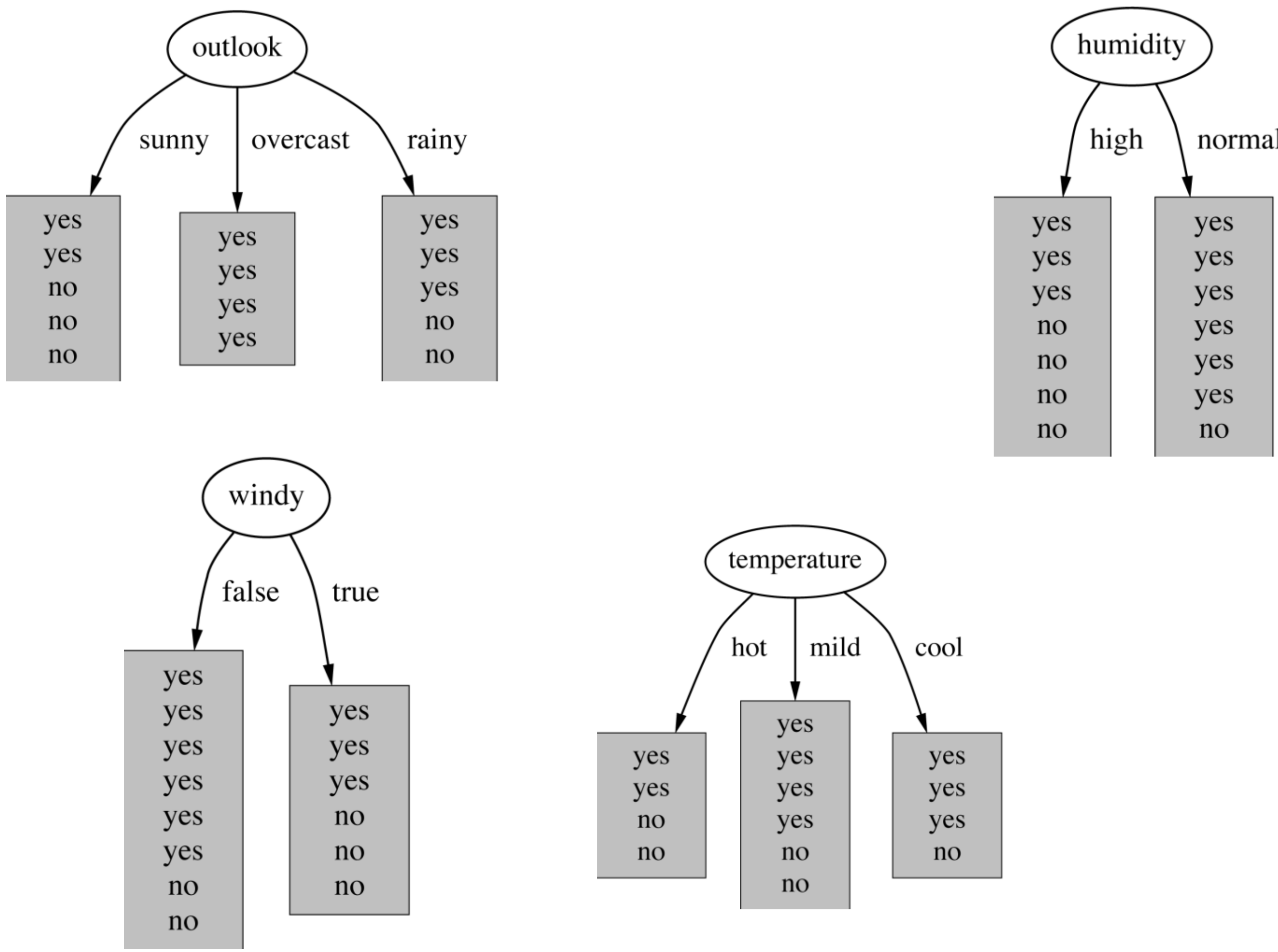
Pros	Cons
Simple to understand, visual	Trees can be overly complex, overfitting the data
Requires little data preparation, although trees perform best if dataset is balanced with class values	Small variations in the data might result in completely different trees being generated
Cost of predicting data is logarithmic in the number of nodes in the tree	Learning an optimal decision tree is NP-complete, even for simple concepts
Can handle both numeric and nominal data	Less expressive than rules.
Not limited to binary class values. Can handle multi-output problems	If dataset is not balanced with class values trees can be biased.
Model can be validated via statistical tests	

Recursion

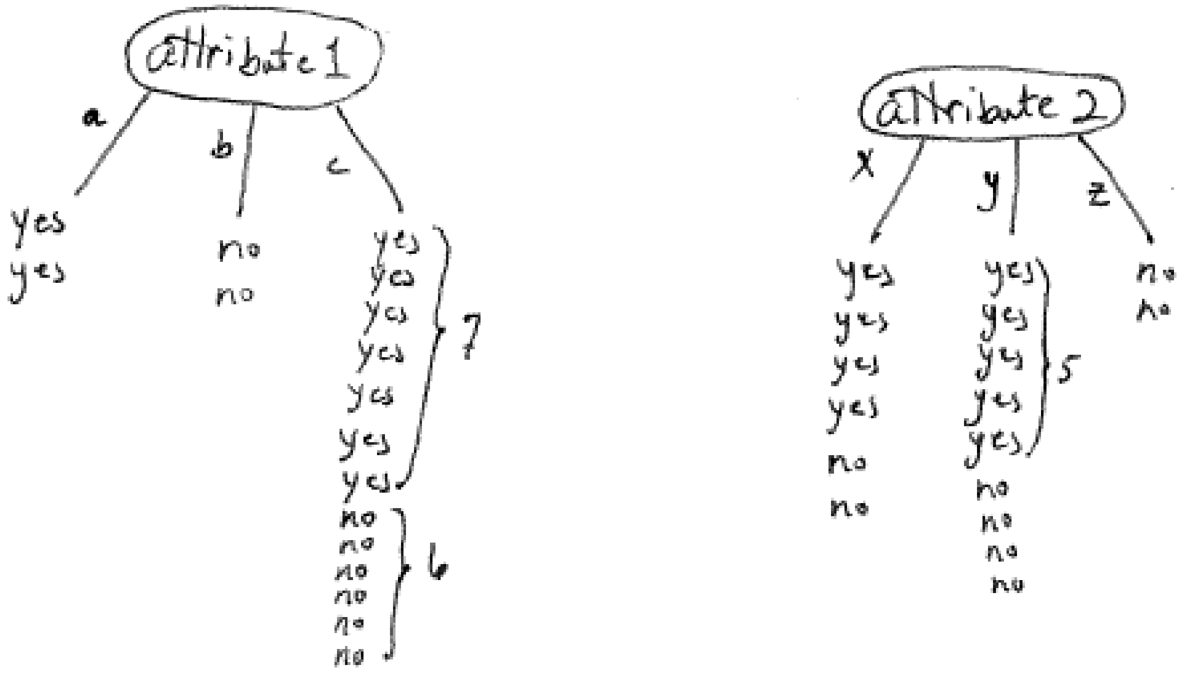


screen:// - VLC media player
Media Playback Audio Video Tools View Help
screen://
1.00x
33:48:19/--:--

Which Attribute to Select?



Same % Correct, Left Seems Better



Both predict correctly 11/17 times

Information Theory uses Entropy

Entropy:

- Stands for "disorder" or a measure of uncertainty
- entropy(p_1, p_2, …, p_n) =

  $-p_1 \log p_1 - p_2 \log p_2 \ldots - p_n \log p_n$ bits

Information Theory & Entropy

Which is has more entropy?

- flip of a coin
- roll of a dice

Information Theory

Measure purity in bits:

- info for one 2-way split:

  info([a,b]) = entropy(a/(a+b), b/(a+b))

- info for one 3-way split:

  info([a,b,c]) = entropy(a/(a+b+c), b/(a+b+c), c/(a+b+c))

Computing Information Gain

Information gain =

info. before splitting – info. after splitting

gain(*Outlook*) = info([9,5]) – info([2,3],[4,0],[3,2])
= 0.940 – 0.694
= 0.246 bits

Information gain for attributes from weather data:

gain(*Outlook*) = 0.246 bits
gain(*Temperature*) = 0.029 bits
gain(*Humidity*) = 0.152 bits
gain(*Windy*) = 0.048 bits

Review of Logarithms

$\log_b x = y$ when $b^y = x$

e.g. $4 = \log_2 16 = 4$ because $2^4 = 16$,

Logarithms are exponents

Changes multiplication to addition

$\log(xy) = \log x + \log y$ since $a^n a^m = a^{n+m}$

Changes division to subtraction:

$\log(x/y) = \log x - \log y$ since $a^n/a^m = a^{n-m}$

Changes raising to a power to multiplication

- $\log(x^y) = y*\log x$ since $(a^n)^m = a^{nm}$

Review of Logarithms

To change to a different base:

$\log_b x = \log_{10} x / \log_{10} b$

e.g.

$\log_2 2 = \log_{10} 2 / \log_{10} 2 = 0.301 / 0.301 = 1$

$\log_2 4 = \log_{10} 4 / \log_{10} 2 = 0.602 / 0.301 = 2$

$\log_2 8 = \log_{10} 8 / \log_{10} 2 = 0.9031 / 0.301 = 3$

Highly Branching Attributes

- If an attribute has many possible values, the split will cause each node to come out very pure, so this method would chose to branch on that attribute
- Often we want to avoid highly branching attributes
- Consider the example on the next slide

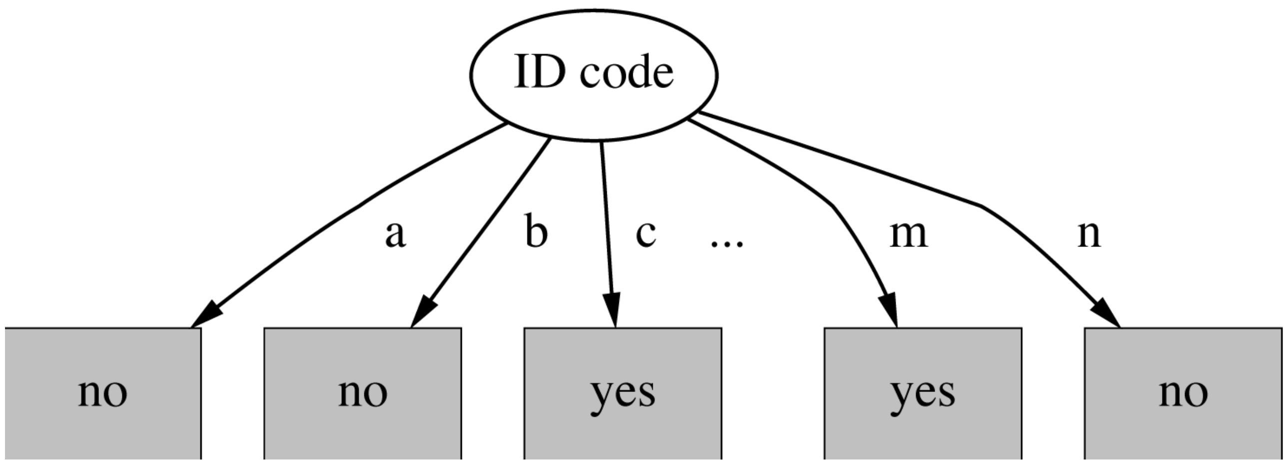
Weather Data with *ID Code*

ID code	Outlook	Temp.	Humidity	Windy	Play
A	Sunny	Hot	High	False	No
B	Sunny	Hot	High	True	No
C	Overcast	ot	High	False	Yes
D	Rainy	Mild	High	False	Yes
E	Rainy	Cool	Normal	False	Yes
F	Rainy	Cool	Normal	True	No
G	Overcast	ool	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	lild	High	True	Yes
M	Overcast	ot	Normal	False	Yes
N	Rainy	Mild	High	True	No

Tree Stump for *ID Code* Attribute

Entropy of split:

⇒Information gain is maximal for ID code (namely 0.940 bits)



$$\text{info}(ID\ code) = info([0,1]) + info([0,1]) + \cdots + info([0,1]) = 0 bits$$

Work Around for Highly Branching Attributes

- To avoid selecting highly branching attribute, use the "gain ratio" rather than the "information gain"
- Calculate the gain ratio by taking into account the number and size of daughter nodes, disregarding any information about the class
- This is called the "intrinsic" information of the split

Gain Ratios for Weather Data

Outlook		Temperature	
Info:	0.693	Info:	0.911
Gain: 0.940-0.693	0.246	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: 0.029/1.557	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