Information Based Learning

Decision Trees

Sarah Jane Delany

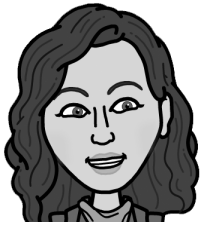
Slides adapted from ML for PDA book

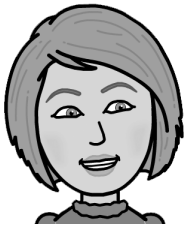
Big Idea

- Guess Who?


Brian


John


Aphra

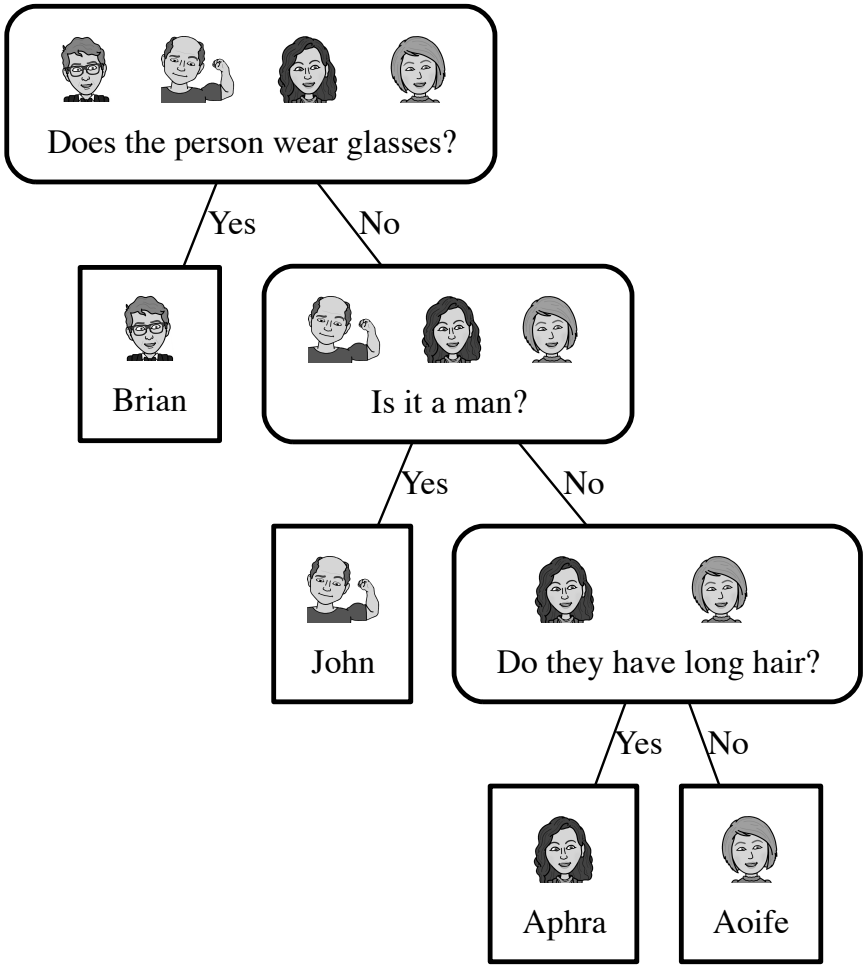

Aoife

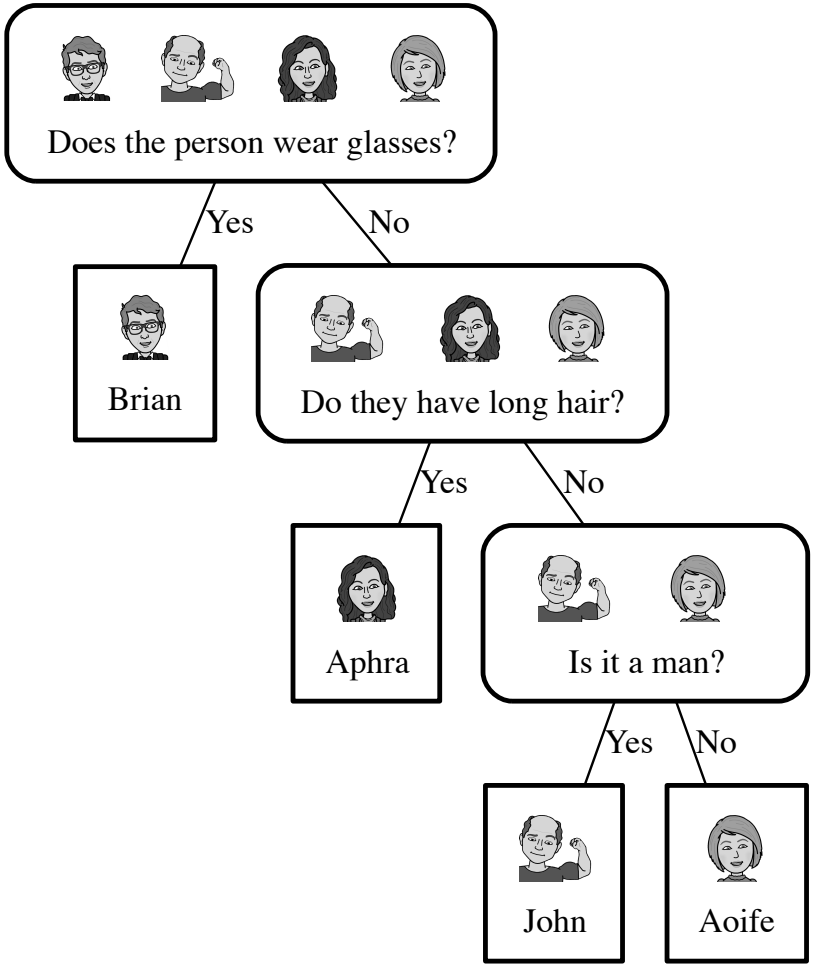
Q1: Does the person wear glasses?

Q2: Is the person a man?

Q3: Does the person have long hair?

Big Idea

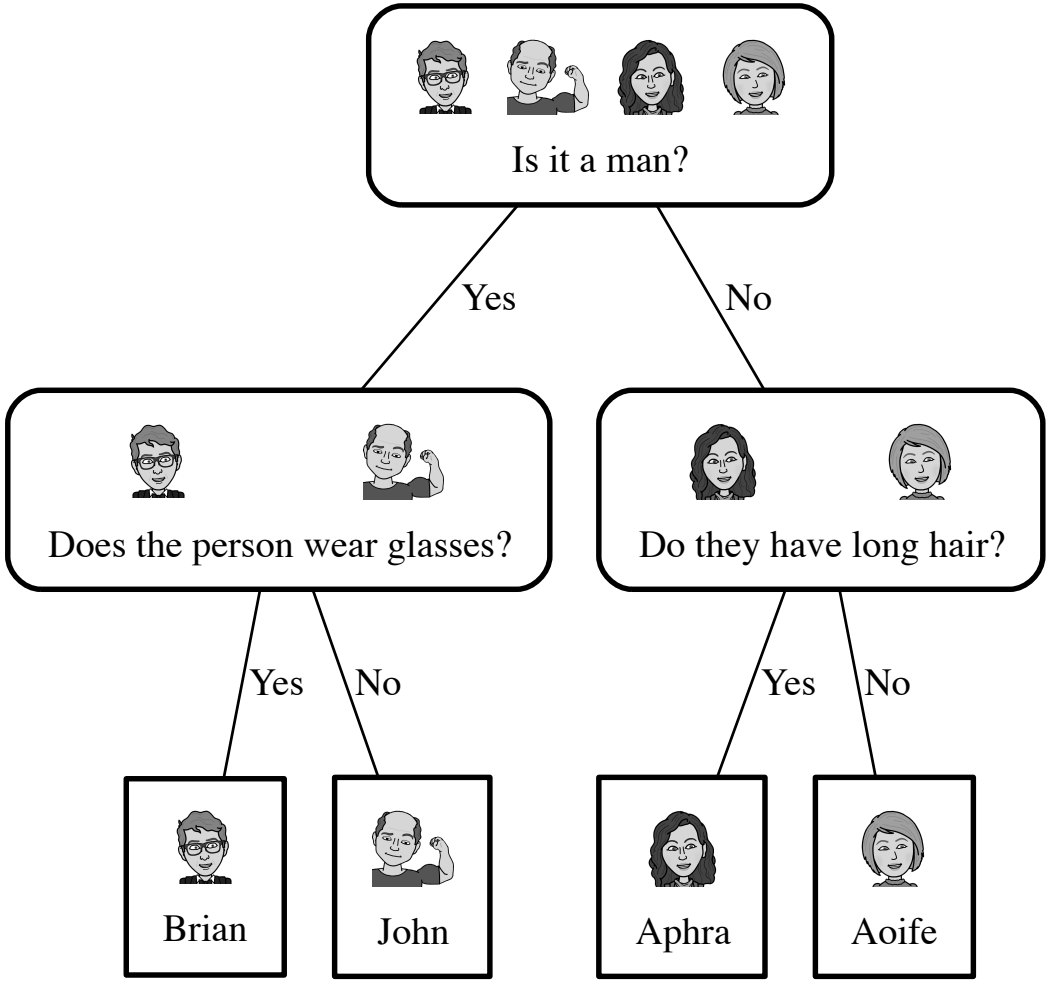




- Average number of questions you have to ask:

$$\frac{1+2+3+3}{4} = 2.25$$

Big Idea



- Average number of questions you have to ask:

$$\frac{2+2+2+2}{4} = 2$$

Big Idea

- Getting an answer to Q2 (Is it a man?) gives more information than an answer to other questions
- Information given is about how the domain is split up after the answer is received and the likelihood of each answer
- Information-based learning uses this idea…

- Algorithms determine which descriptive features provide most info about the target feature

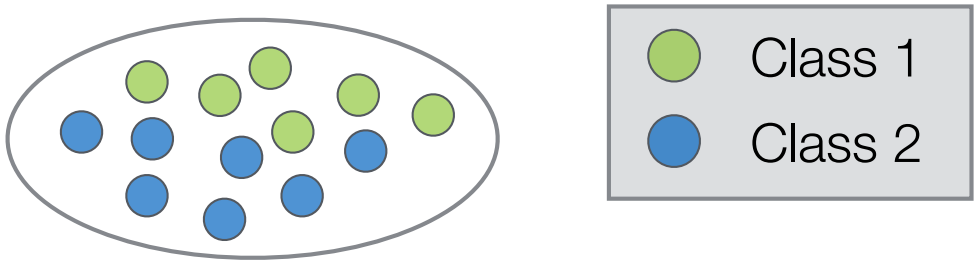
Man	Long Hair	Glasses	Name
Yes	No	Yes	Brian
Yes	No	No	John
No	Yes	No	Aphra
No	No	No	Aoife

- Predictions are made by sequentially testing the features in order of *informativeness*

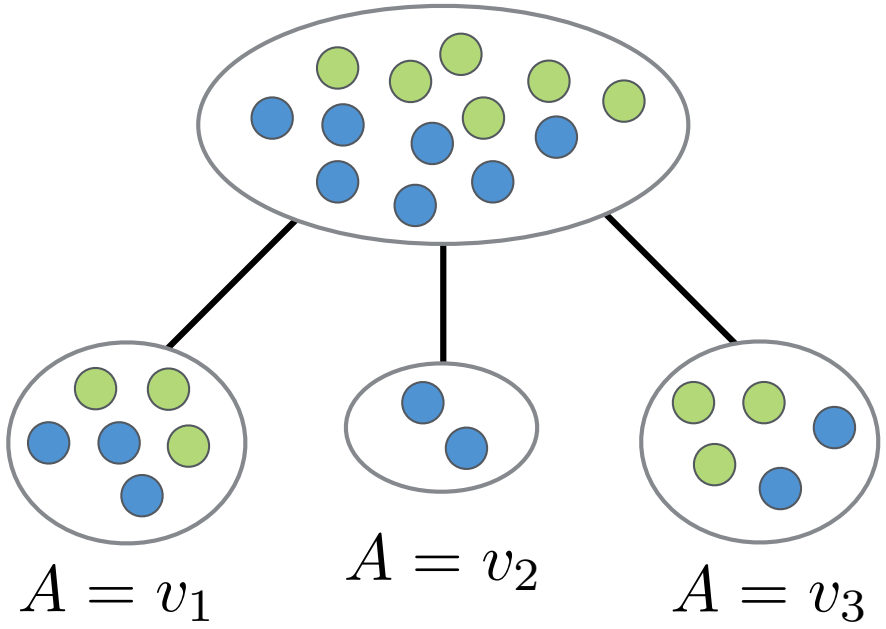
How do we measure this?

Decision Tree Training

1. Initially all examples in the training set are placed at the root node of the tree.



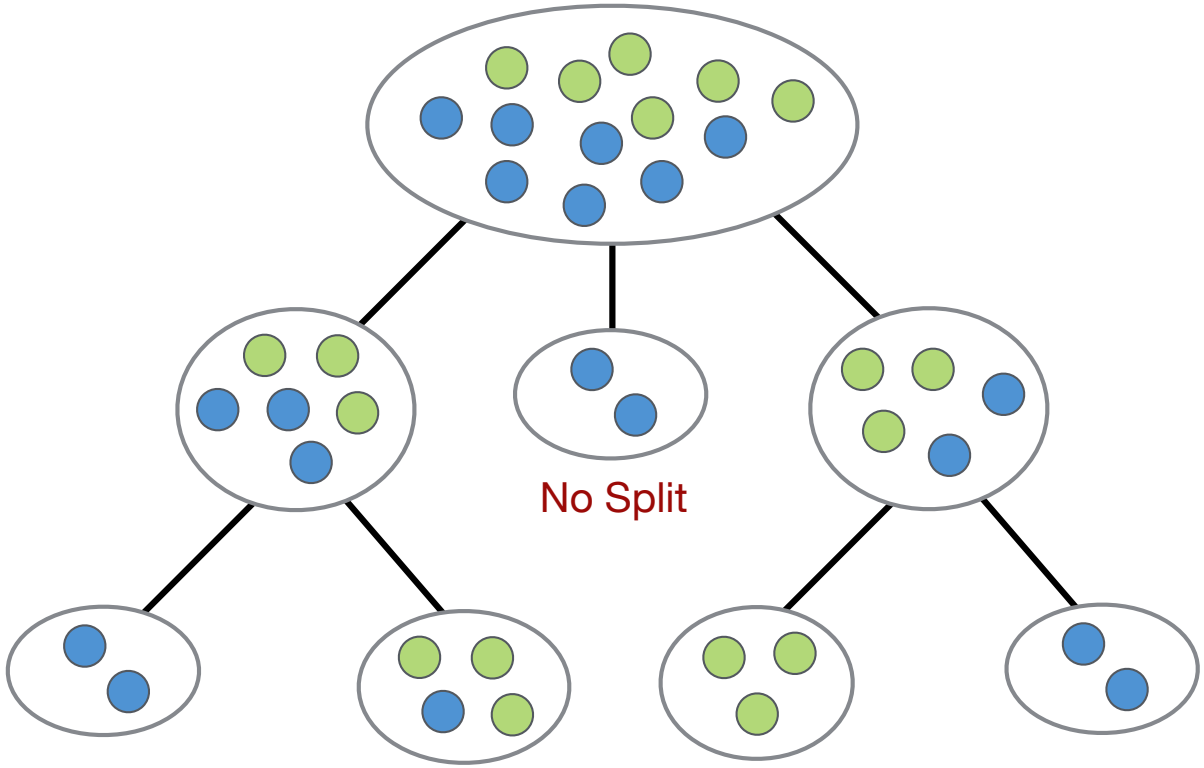
2. A test is performed on a features (*A*) to split the examples at the root node into two or more subsets of examples at interior nodes.



Decision Rules

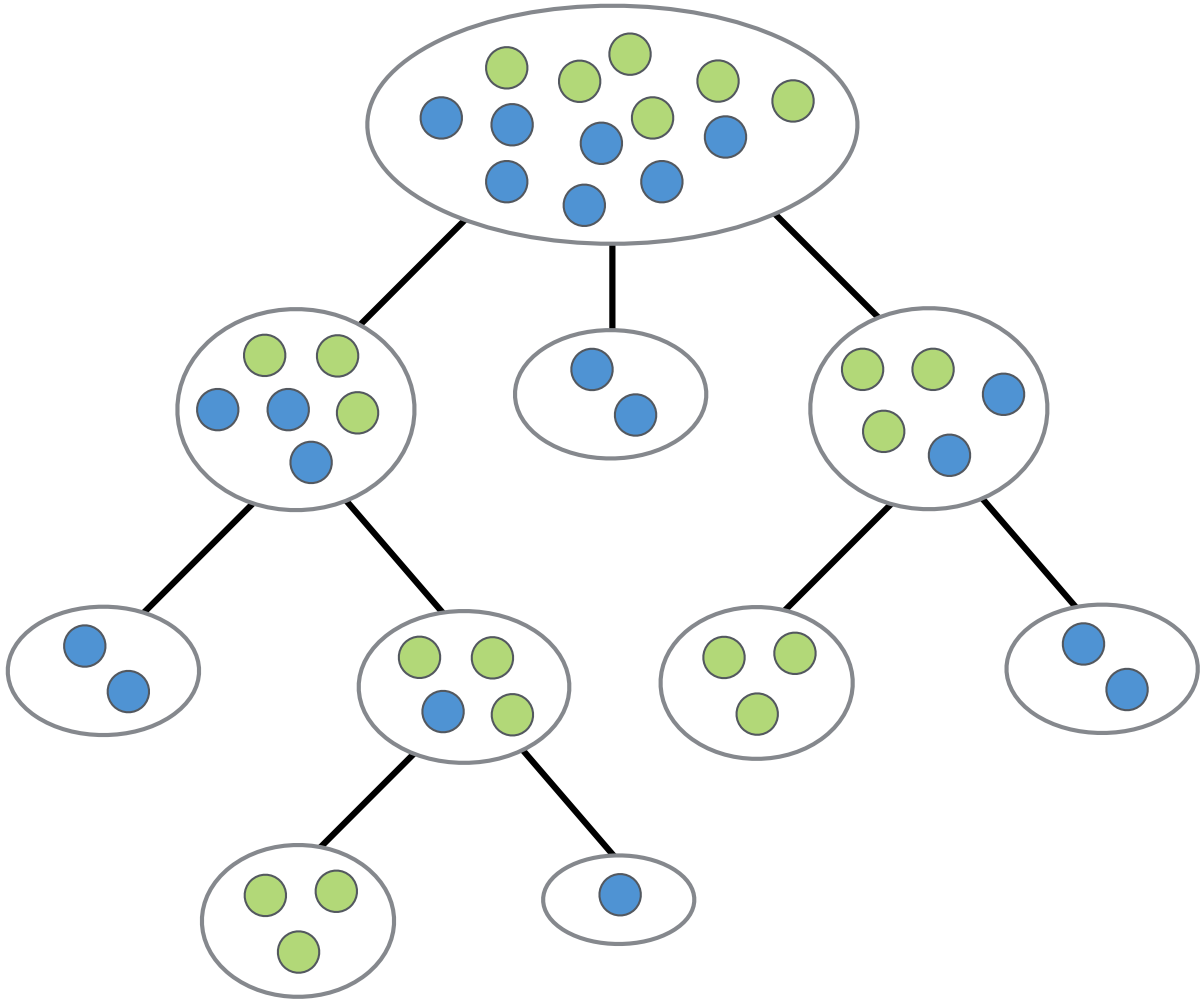
Decision Tree Training

3. The same process is now applied to each interior node, except at leaf nodes where all examples have the same class.



Decision Tree Training

4. Repeat until all leaf nodes in the tree have examples with the same class.

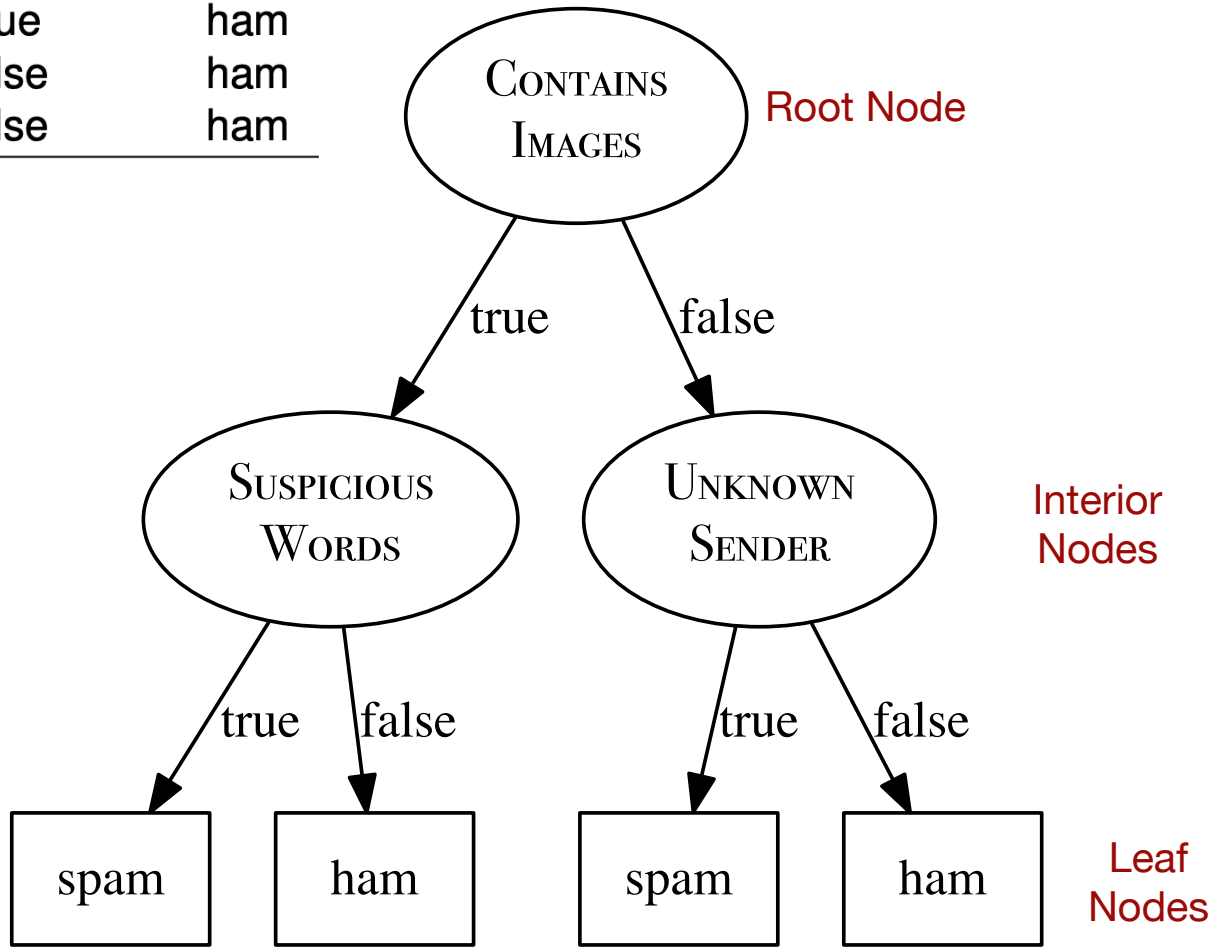


Decision Tree Classification

ID	Suspicious Words	Unknown Sender	Contains Images	Class
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

To classify a query example:

Test the value of the feature at the node and follow the relevant branch until a leaf node is reached



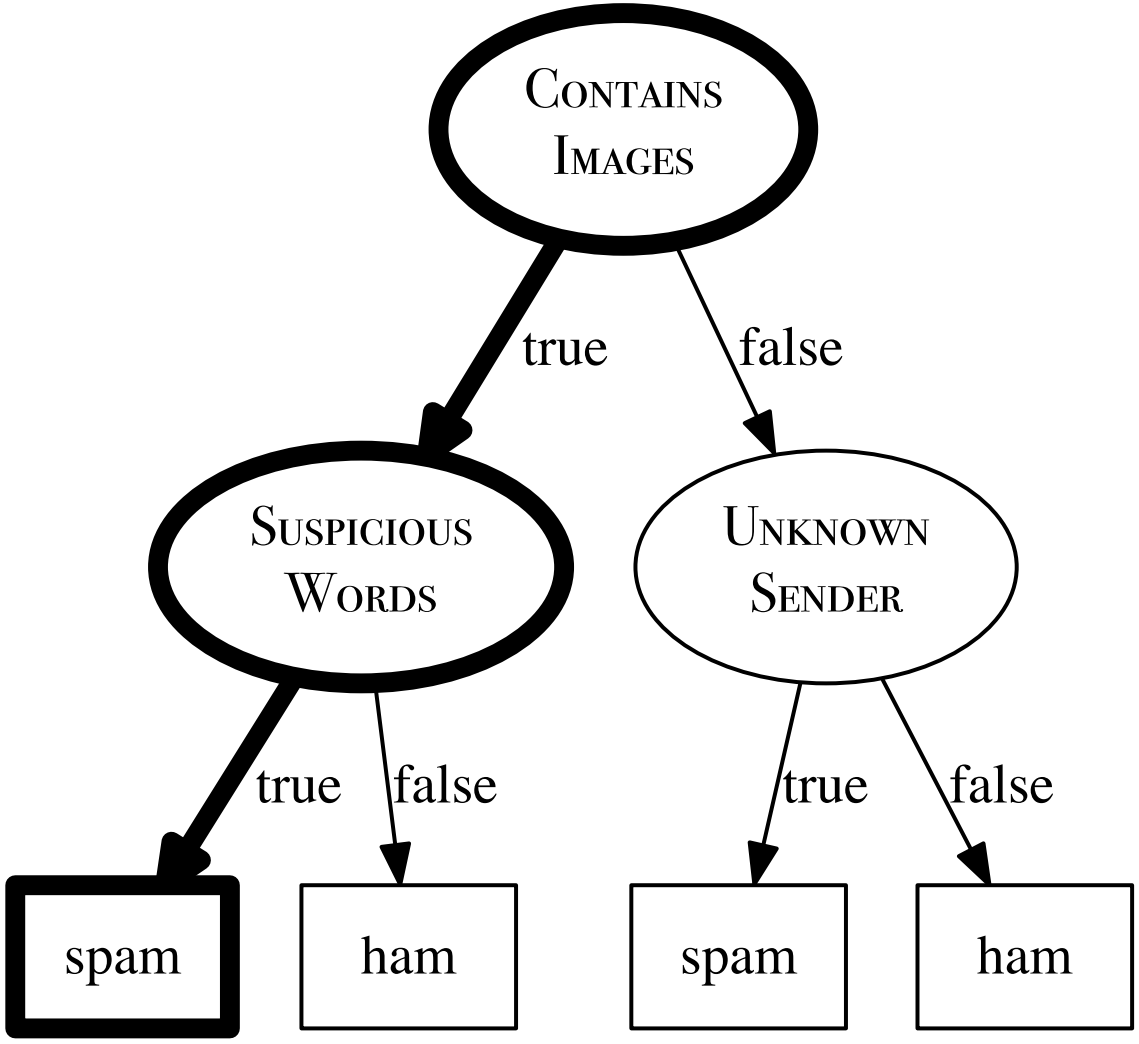
Decision Tree Classification

Query example:

Suspicious Words: true

Unknown Sender: true

Contains Images: true

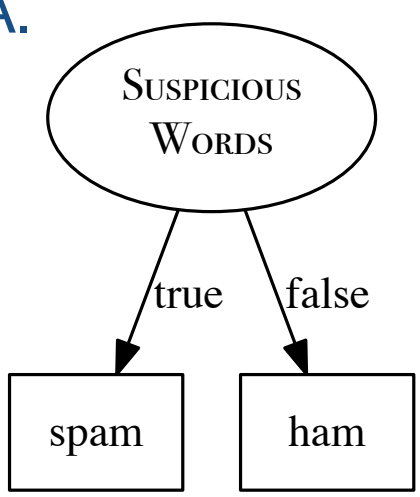


Decision Tree Classification

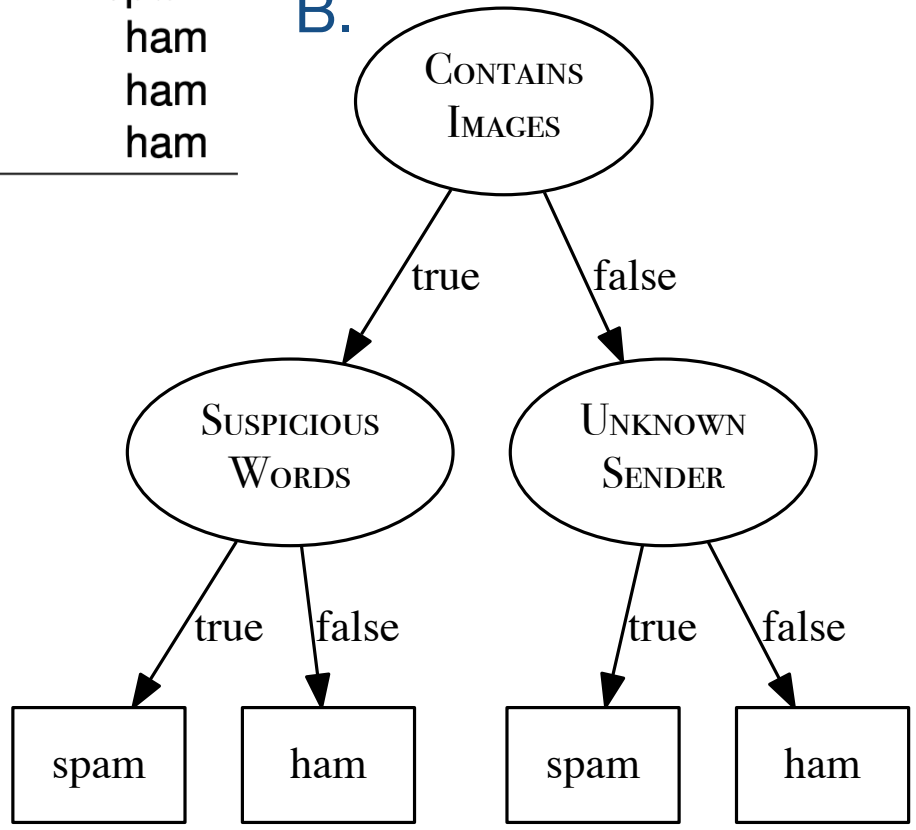
ID	Suspicious Words	Unknown Sender	Contains Images	Class
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

A or B?

A.



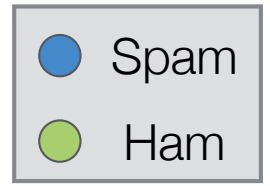
B.



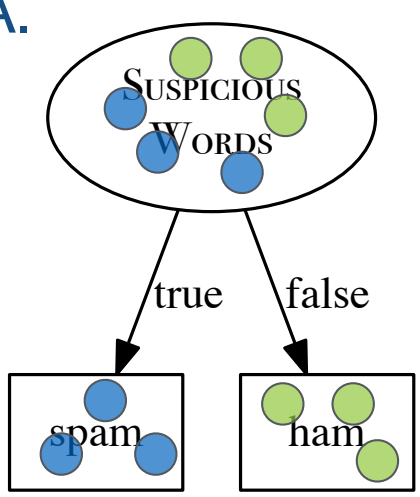
Both trees are consistent with the examples in the training data

Decision Tree Classification

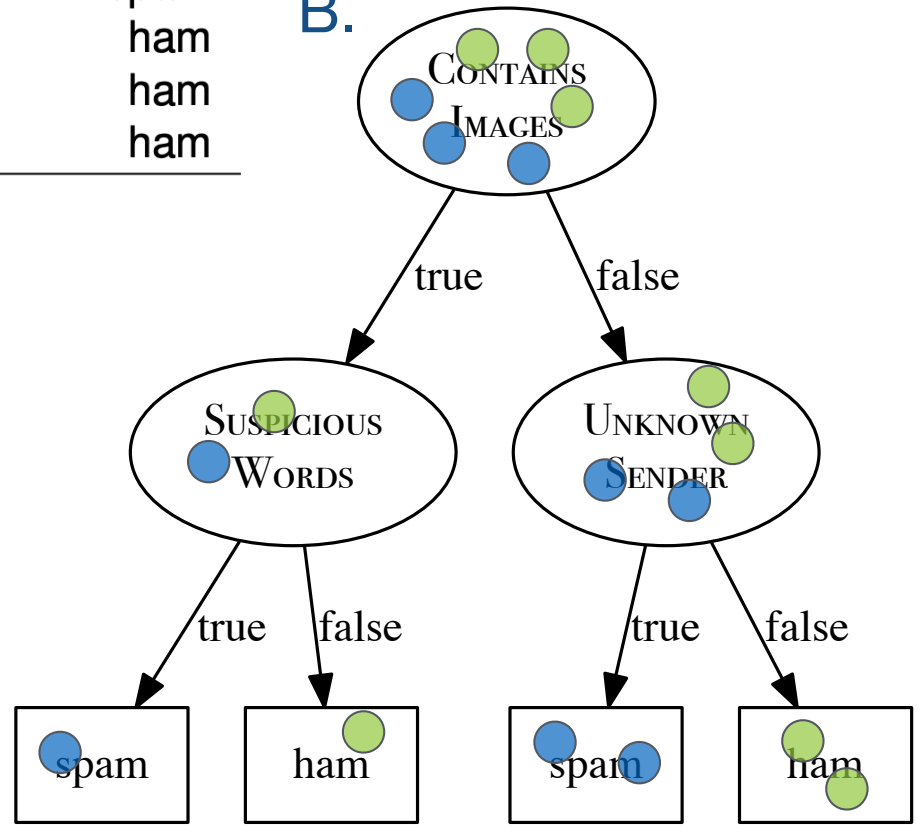
ID	Suspicious Words	Unknown Sender	Contains Images	Class
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham



A.



B.



A node is pure if all examples at that node have the same label

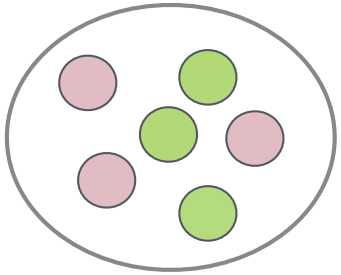
Decision Trees Inductive Bias

- Preference Bias: Choose decision trees that have fewer tests, i.e. shallower trees
- Pure nodes provide more information about the value of the of the target feature for a query
- Descriptive features that split the dataset into pure sets provide information about the target feature and are considered more informative
- Testing the informative features early on in the tree can result in shallower trees.
- Claude Shannon's entropy model is a computational metric of the purity of a set

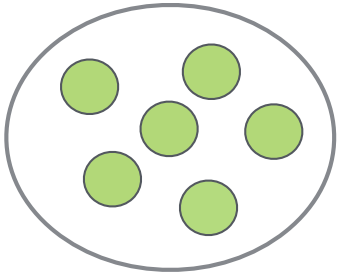
Entropy ~ the uncertainty associated with guessing the result if you were to make a random selection from a set

Entropy

Entropy ~ the uncertainty associated with guessing the result if you were to make a random selection from a set



Node has high uncertainty
→ High entropy



Node has low uncertainty
→ Low entropy

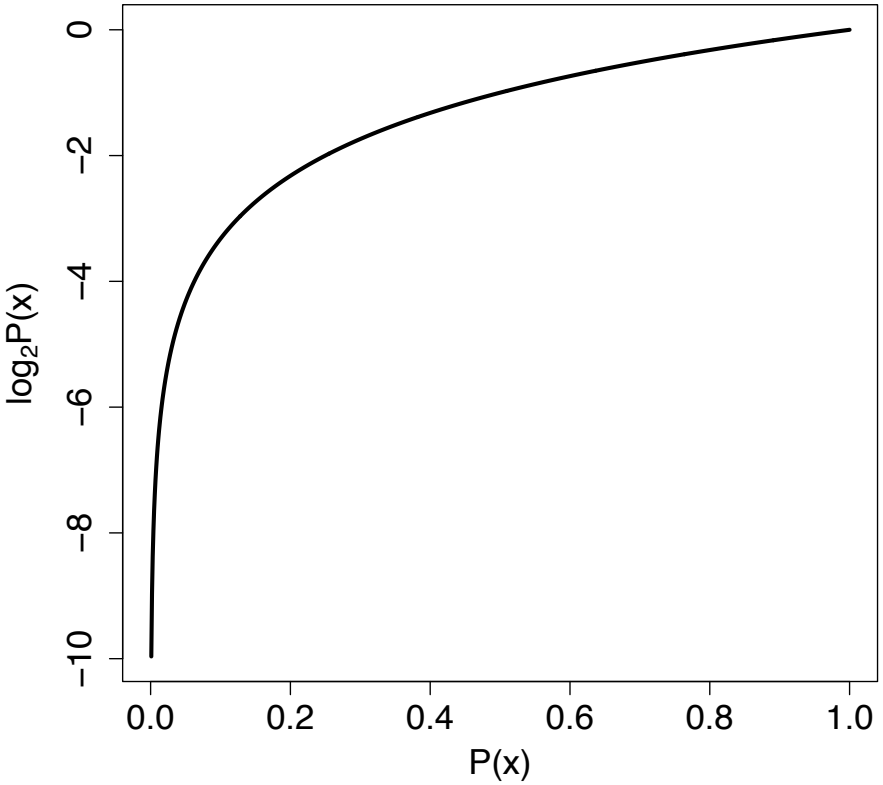
- Entropy is related to the probability of an outcome
  - High probability—> Low entropy
  - Low probability —> High entropy

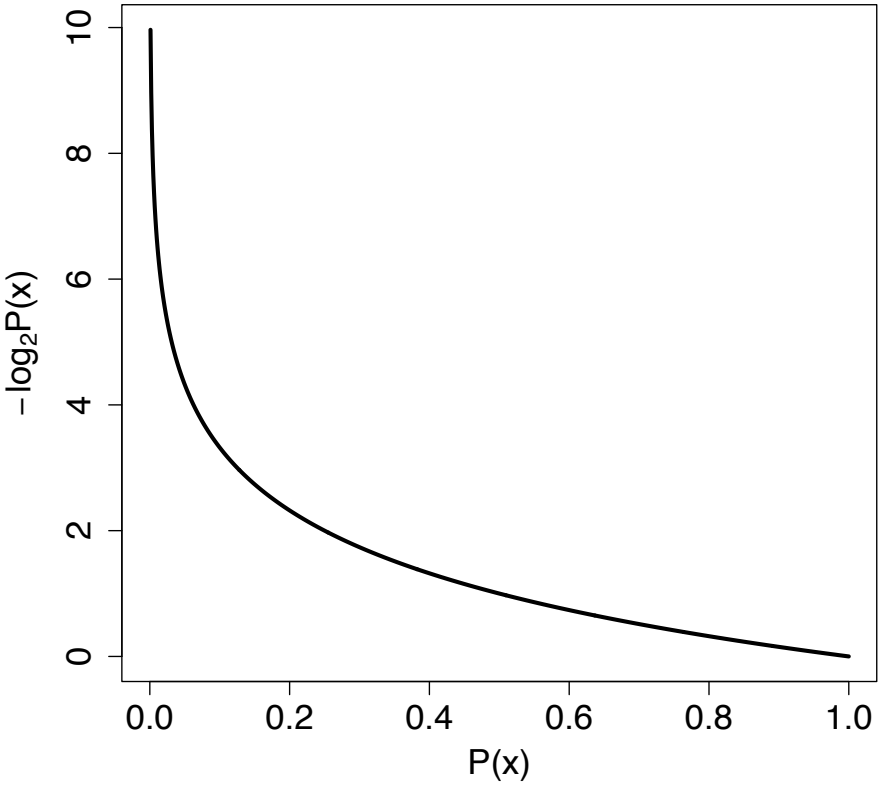
Entropy

- The log of a probability multiplied by -1 gives the mapping
  - High probability —> low entropy
  - Low probability —> high entropy

$$log_b(a) = x \text{ where } b^x = a$$

$$log_2(0.5) = -1 \text{ because } 2^{-1} = 0.5$$
$$log_2(1) = 0 \text{ because } 2^0 = 1$$
$$log_2(8) = 3 \text{ because } 2^3 = 8$$
$$log_5(25) = 2 \text{ because } 5^2 = 25$$



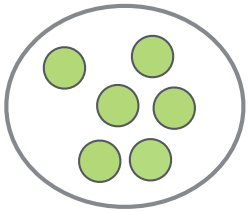


Entropy

- Entropy of a dataset of examples D with labels $\{t_1, t_2, \ldots, t_l\}$

$$H(D) = -\sum_{i=1}^{l} (P(t_i) \times log_2(P(t_i)))$$

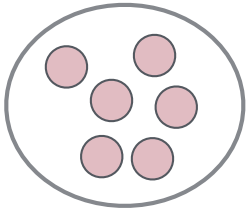
where $P(t_i)$ is the probability of randomly selecting example with label t_i



$p(t_1) = 6/6 = 1.0 \quad p(t_2) = 0/6 = 0$

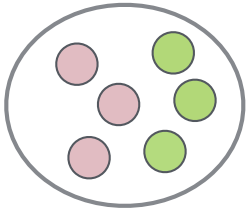
$H(D) = -((1 \times \log_2(1)) + (0 \times \log_2(0))) = -(0 + 0) = 0$

NB: Define $\log_2(0)=0$



$p(t_1) = 0/6 = 0 \quad p(t_2) = 6/6 = 1.0$

$H(D) = -((0 \times \log_2(0)) + (1 \times \log_2(1))) = -(0 + 0) = 0$



$p(t_1) = 3/6 = 0.5 \quad p(t_2) = 3/6 = 0.5$

$H(D) = -((0.5 \times \log_2(0.5)) + (0.5 \times \log_2(0.5))) = -(-0.5 - 0.5) = 1$

Entropy examples

- The entropy of a set of 52 playing cards:

$$H(D) = -\sum_{i=1}^{52} P(card = i) \times log_2(P(card = i))$$

$$= -\sum_{i=1}^{52} \frac{1}{52} \times log_2(\frac{1}{52}) = \ 5.7$$

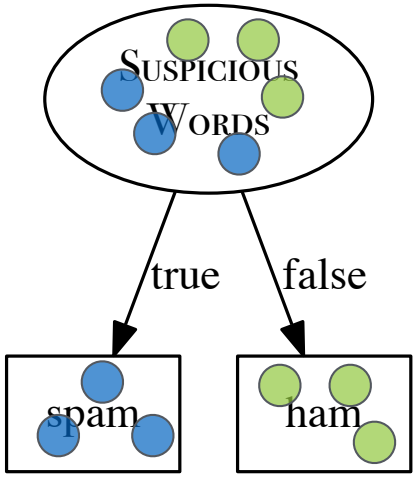
- The entropy of a set of 52 playing cards distinguishing cards only by suit:

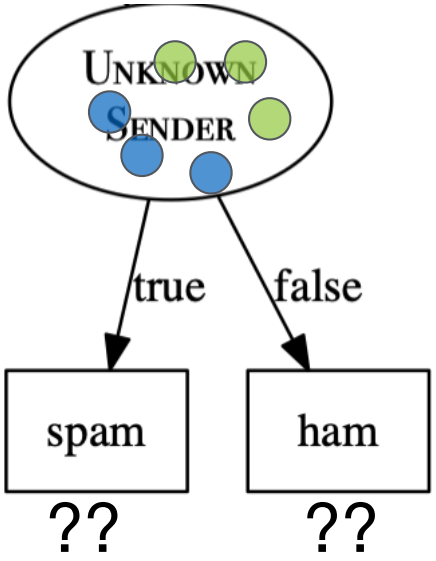
$$H(D) = -\sum_{i=1}^{4} P(suit = i) \times log_2(P(suit = i))$$

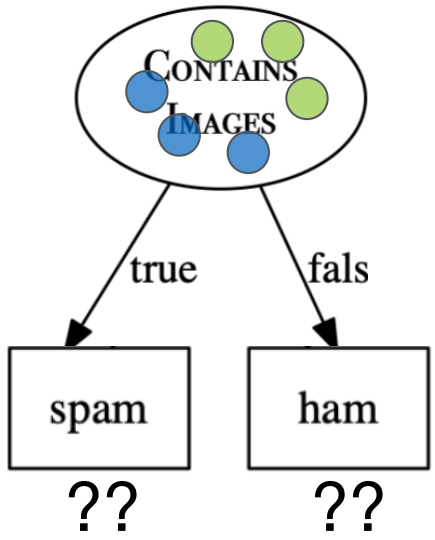
$$= -\sum_{i=1}^{4} \frac{13}{52} \times log_2(\frac{13}{52}) = \ 2$$

Entropy

ID	Suspicious Words	Unknown Sender	Contains Images	Class
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

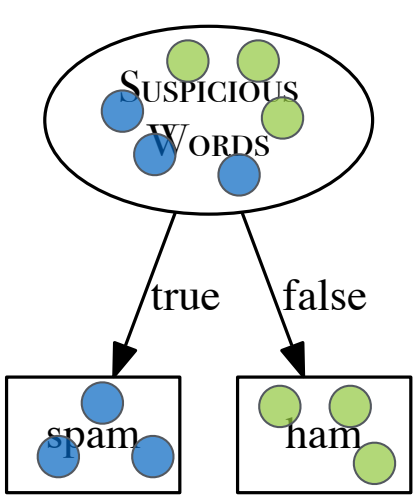


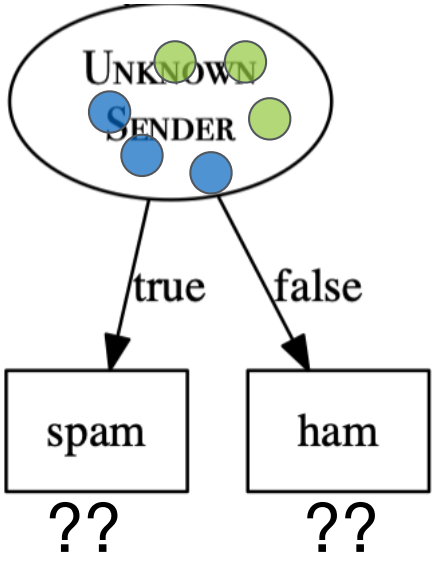


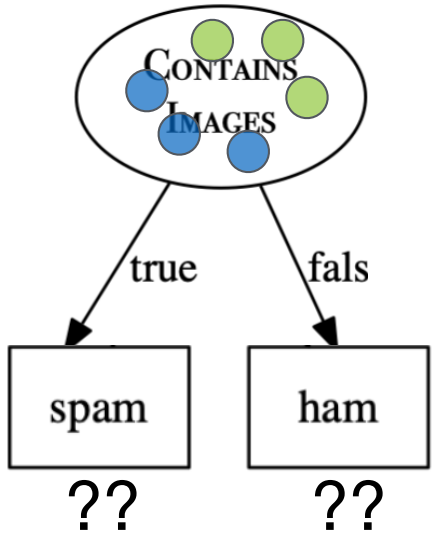


Entropy

ID	Suspicious Words	Unknown Sender	Contains Images	Class
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham







Information Gain ~ a measure of the reduction in the overall entropy of a set that is achieved by testing on a feature

Information Gain

- IG for descriptive feature d that splits a dataset D of examples into subsets or partitions $\{D_1, D_2, \ldots, D_k\}$

$$IG(d, D) = (\text{original entropy}) - (\text{entropy after split})$$

The entropy remaining after the dataset is split using descriptive feature *d*

$$IG(d, D) = H(D) - rem(d, D)$$

$$H(D) = -\sum_{i=1}^{l} (P(t_i) \times log_s(P(t_i)))$$

The entropy on the full dataset wrt the target feature *t*

$$rem\,(d, \mathcal{D}) = \sum_{i}^{k} \underbrace{\frac{|\mathcal{D}_i|}{|\mathcal{D}|}}_{\text{weighting}} \times \underbrace{H\,(\mathcal{D}_i)}_{\substack{\text{entropy of} \\ \text{partition } \mathcal{D}_i}}$$

Each partition is weighted in proportion to its size

Information Gain - example

Calculate $H(D)$

Entropy dataset wrt target feature

ID	Suspicious Words	Unknown Sender	Contains Images	Class
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

$$H(D) = -\sum_{l \in \{\text{spam},\text{ham}\}} (P(t_l) \times log_2(P(t_l)))$$

$$= -((P(t = \text{spam}) \times log_2(P(t = \text{spam}))$$

$$+ (P(t = \text{ham}) \times log_2(P(t = \text{ham})))$$

$$= -\left(\left({}^3/_6 \times log_2({}^3/_6)\right) + \left({}^3/_6 \times log_2({}^3/_6)\right)\right) = 1$$

Information Gain - example

Calculate $rem(SW, D)$

Entropy remaining after splitting on SW descriptive feature

ID	Suspicious Words	Unknown Sender	Contains Images	Class
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

$$rem(\mathrm{SW}, D) = \sum_{i \in \{true, false\}} \left(\frac{|D_i|}{|D|}\right) \times H(D_i)$$

$$rem\left(\mathrm{SW}, D\right) = \left(\frac{|D_{true}|}{|D|} \times H\left(D_{true}\right)\right) + \left(\frac{|D_{false}|}{|D|} \times H\left(D_{false}\right)\right)$$

$$= ({}^{3}/_{6} \times (- \sum_{c \in \{spam, ham\}} P(t_c) \times log_2(P(t_c))) + ({}^{3}/_{6} \times (- \sum_{c \in \{spam, ham\}} P(t_c) \times log_2(P(t_c)))$$

$$= ({}^{3}/_{6} \times (-(({}^{3}/_{3} \times log_2({}^{3}/_{3})) + ({}^{0}/_{3} \times log_2({}^{0}/_{3})))))$$
$$+ ({}^{3}/_{6} \times (-(({}^{0}/_{3} \times log_2({}^{0}/_{3})) + ({}^{3}/_{3} \times log_2({}^{3}/_{3})))))$$

$$= 0$$

Information Gain - example

Calculate $rem(\mathrm{US}, D)$

Entropy remaining after splitting on SW descriptive feature

ID	Suspicious Words	Unknown Sender	Contains Images	Class
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

$$rem(\mathrm{US}, D) = \sum_{i \in \{true, false\}} \left(\frac{|D_i|}{|D|}\right) \times H(D_i)$$

$$rem\left(\mathrm{US}, D\right) = \left(\frac{|D_{true}|}{|D|} \times H\left(D_{true}\right)\right) + \left(\frac{|D_{false}|}{|D|} \times H\left(D_{false}\right)\right)$$

$$= ({}^{3}/_{6} \times (- \sum_{c \in \{spam, ham\}} P(t_c) \times log_2(P(t_c))) + ({}^{3}/_{6} \times (- \sum_{c \in \{spam, ham\}} P(t_c) \times log_2(P(t_c)))$$

$$= ({}^{3}/_{6} \times (-(({}^{2}/_{3} \times log_2({}^{2}/_{3})) + ({}^{1}/_{3} \times log_2({}^{1}/_{3})))))$$
$$+ ({}^{3}/_{6} \times (-(({}^{1}/_{3} \times log_2({}^{1}/_{3})) + ({}^{2}/_{3} \times log_2({}^{2}/_{3})))))$$

$$= 0.9183$$

Information Gain - example

Calculate $rem(\mathrm{CI}, D)$

Entropy remaining after splitting on SW descriptive feature

ID	Suspicious Words	Unknown Sender	Contains Images	Class
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

$$rem(\mathrm{CI}, D) = \sum_{i \in \{true, false\}} \left(\frac{|D_i|}{|D|}\right) \times H(D_i)$$

$$rem\left(\mathrm{CI}, D\right) = \left(\frac{|D_{true}|}{|D|} \times H\left(D_{true}\right)\right) + \left(\frac{|D_{false}|}{|D|} \times H\left(D_{false}\right)\right)$$

$$= ({}^{2}/_{6} \times (- \sum_{c \in \{spam, ham\}} P(t_c) \times log_2(P(t_c))) + ({}^{4}/_{6} \times (- \sum_{c \in \{spam, ham\}} P(t_c) \times log_2(P(t_c)))$$

$$= ({}^{2}/_{6} \times (-(({}^{1}/_{2} \times log_2({}^{1}/_{2})) + ({}^{1}/_{2} \times log_2({}^{1}/_{2})))))$$

$$+ ({}^{4}/_{6} \times (-(({}^{2}/_{4} \times log_2({}^{2}/_{4})) + ({}^{2}/_{4} \times log_2({}^{2}/_{4})))))$$

$$= 1$$

Information Gain - example

$$IG(d, D) = H(D) - rem(d, D)$$

$$\begin{aligned} IG(\text{SW}, D) &= H(D) - rem(\text{SW}, D) \\ &= 1 - 0 = 1 \\ IG(\text{US}, D) &= H(D) - rem(\text{US}, D) \\ &= 1 - 0.9183 = 0.0817 \\ IG(\text{CI}, D) &= H(D) - rem(\text{CI}, D) \\ &= 1 - 1 = 0 \end{aligned}$$

- This result matches our intuitions - Suspicious Words is the best feature to split on

ID3 Algorithm

- ID3 (Iterative Dichotomizer 3)
- Attempts to create the shallowest tree that is consistent with the dataset
- Builds the tree in a recursive, depth first manner, beginning at the root node and working down to the leaf nodes.

ID3 Algorithm

Set of training examples D
Set of descriptive features $\mathbf{d}$

- IF all examples in D belong to the same class C THEN
  - Return a leaf node and label it with class C
- IF no features left in D THEN
  - Return a leaf node and label it with majority class C of D
- IF no examples left in D THEN
  - Return a leaf node and label it with majority class C of examples at the immediate parent node
- ELSE
  - Select a feature d_i from $\mathbf{d}$ based on some feature selection criterion
  - Generate a tree node with d_i as the test feature
  - FOR EACH value v_j of d_i
    - Let $D_j \subset D$ contain all examples with $d_i = v_j$
    - Build a subtree by applying $\mathrm{ID3}(D_j)$

ID3 Algorithm

Set of training examples D
Set of descriptive features $\mathbf{d}$

- IF all examples in D belong to the same class C THEN
  - Return a leaf node and label it with class C
- IF no features left in $\mathbf{d}$ THEN
  - Return a leaf node and label it with majority class C of D
- IF no examples left in D THEN
  - Return a leaf node and label it with majority class C of examples at the immediate parent node

Stop growing the current path by adding a leaf node

- ELSE
  - Select a feature d_i from $\mathbf{d}$ based on some feature selection criterion
  - Generate a tree node with d_i as the test feature
  - FOR EACH value v_j of d_i
    - Let $D_j \subset D$ contain all examples with $d_i = v_j$
    - Build a subtree by applying $\mathrm{ID3}(D_j)$

ID3 Algorithm

Set of training examples D
Set of descriptive features $\mathbf{d}$

- IF all examples in D belong to the same class C THEN
  - Return a leaf node and label it with class C
- IF no features left in D THEN
  - Return a leaf node and label it with majority class C of D
- IF no examples left in D THEN
  - Return a leaf node and label it with majority class C of examples at the immediate parent node

Stop growing the current path by adding a leaf node

- ELSE
  - Select a feature d_i from $\mathbf{d}$ based on some feature selection criterion
  - Generate a tree node with d_i as the test feature
  - FOR EACH value v_j of d_i
    - Let $D_j \subset D$ contain all examples with $d_i = v_j$
    - Build a subtree by applying $\mathrm{ID3}(D_j)$

Extend the current path by adding an interior node and growing its branches

ID3 Example

ID	Stream	Slope	Elevation	Vegetation
1	false	steep	high	chaparral
2	true	moderate	low	riparian
3	true	steep	medium	riparian
4	false	steep	medium	chaparral
5	false	flat	high	conifer
6	true	steep	highest	conifer
7	true	steep	high	chaparral

Ecological modelling: Predicting vegetation based on features from aerial maps, which inputs to animal management

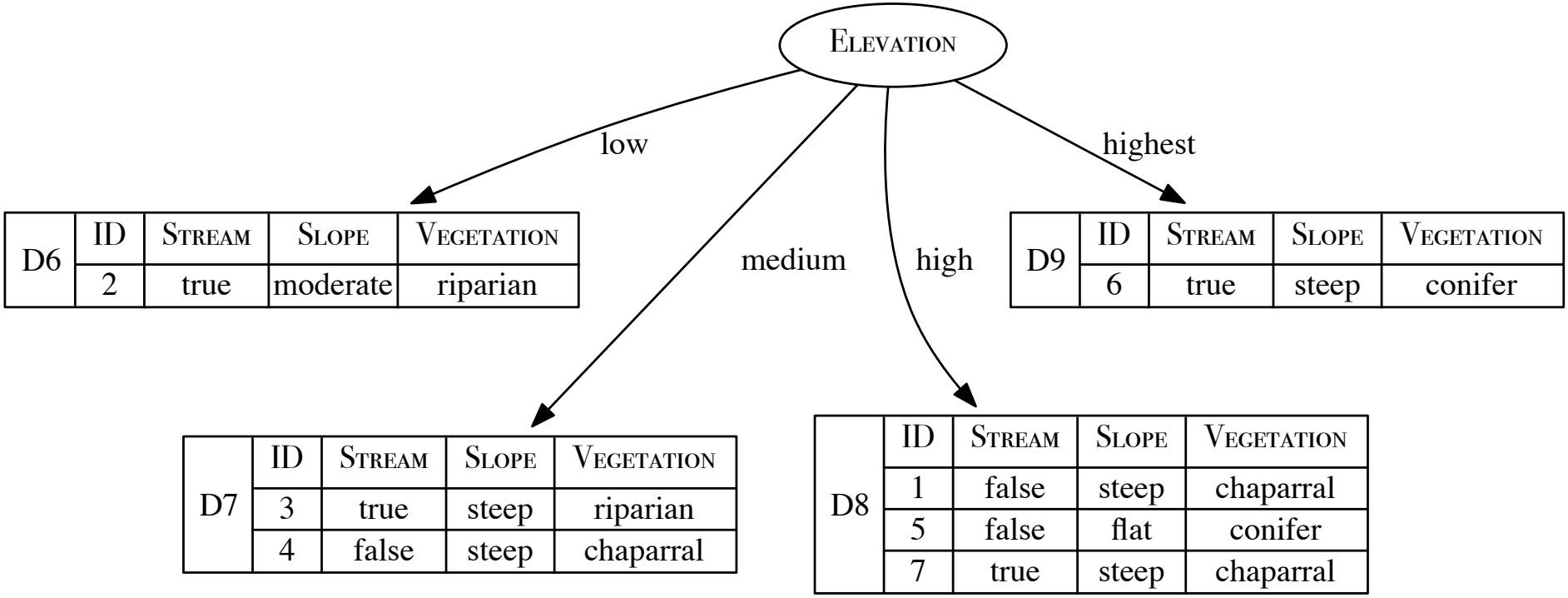
$$
\begin{aligned}
H(\mathcal{D}) &= - \sum_{l \in \left\{ \begin{smallmatrix} \text{chaparral}, \\ \text{riparian}, \\ \text{conifer} \end{smallmatrix} \right\}} P(\text{Vegetation} = l) \times log_2 \left(P(\text{Vegetation} = l) \right) \\
&= - \left(\left({}^3/_7 \times log_2({}^3/_7) \right) + \left({}^2/_7 \times log_2({}^2/_7) \right) + \left({}^2/_7 \times log_2({}^2/_7) \right) \right) \\
&= 1.5567
\end{aligned}
$$

ID	STREAM	SLOPE	ELEVATION	VEGETATION
1	false	steep	high	chaparral
2	true	moderate	low	riparian
3	true	steep	medium	riparian
4	false	steep	medium	chaparral
5	false	flat	high	conifer
6	true	steep	highest	conifer
7	true	steep	high	chaparral

$$H(\mathcal{D}) = 1.5567$$

Split By Feature	Label	Part.	Instances	Partition Entropy	Rem.	Info. Gain
STREAM	*'true'*	$\mathcal{D}_1$	$\mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_6, \mathbf{d}_7$	1.5	1.2507	0.3060
	'false'	$\mathcal{D}_2$	$\mathbf{d}_1, \mathbf{d}_4, \mathbf{d}_5$	0.9183		
SLOPE	*'flat'*	$\mathcal{D}_3$	$\mathbf{d}_5$	0	0.9793	0.5774
	'moderate'	$\mathcal{D}_4$	$\mathbf{d}_2$	0		
	'steep'	$\mathcal{D}_5$	$\mathbf{d}_1, \mathbf{d}_3, \mathbf{d}_4, \mathbf{d}_6, \mathbf{d}_7$	1.3710		
ELEVATION	*'low'*	$\mathcal{D}_6$	$\mathbf{d}_2$	0	0.6793	0.8774
	'medium'	$\mathcal{D}_7$	$\mathbf{d}_3, \mathbf{d}_4$	1.0		
	'high'	$\mathcal{D}_8$	$\mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_7$	0.9183		
	'highest'	$\mathcal{D}_9$	$\mathbf{d}_6$	0		

What feature should be at the root of the tree?



D6	ID	Stream	Slope	Vegetation
	2	true	moderate	riparian

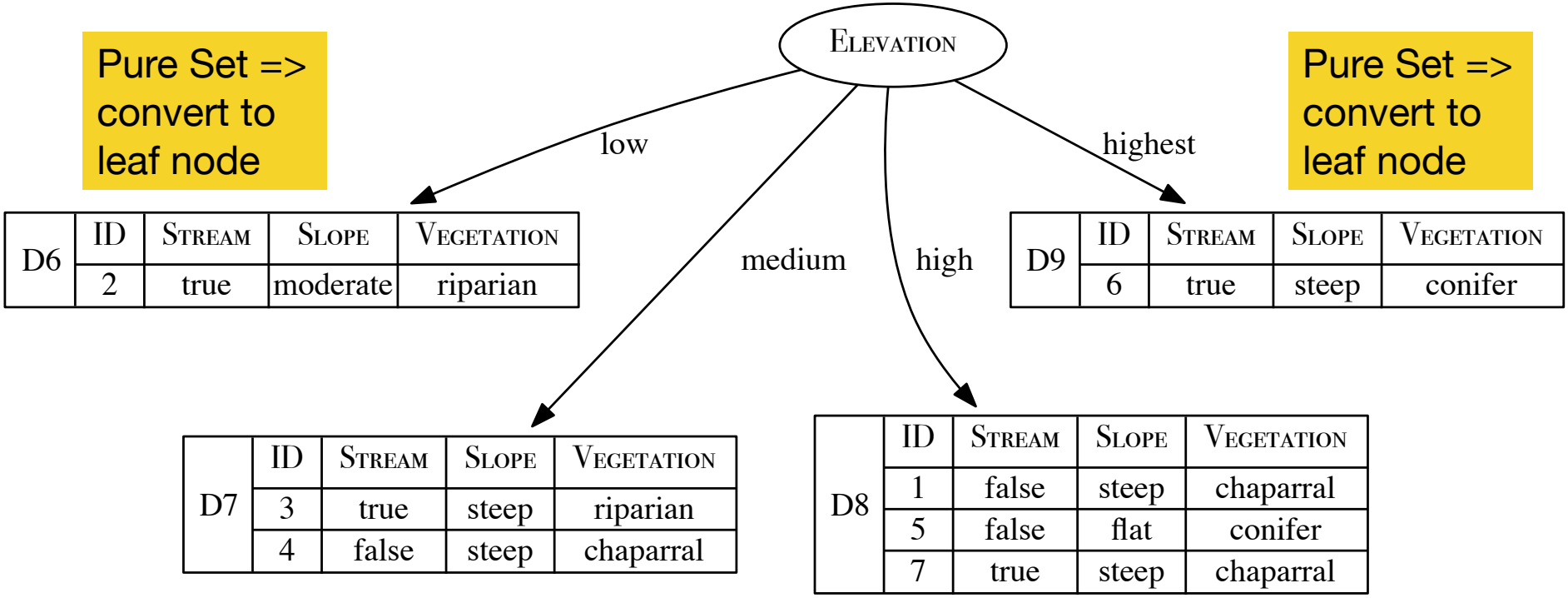
D7	ID	Stream	Slope	Vegetation
	3	true	steep	riparian
	4	false	steep	chaparral

D8	ID	Stream	Slope	Vegetation
	1	false	steep	chaparral
	5	false	flat	conifer
	7	true	steep	chaparral

D9	ID	Stream	Slope	Vegetation
	6	true	steep	conifer

ID	Stream	Slope	Elevation	Vegetation
1	false	steep	high	chaparral
2	true	moderate	low	riparian
3	true	steep	medium	riparian
4	false	steep	medium	chaparral
5	false	flat	high	conifer
6	true	steep	highest	conifer
7	true	steep	high	chaparral

Pure Set => convert to leaf node



Pure Set => convert to leaf node

D6	ID	Stream	Slope	Vegetation
	2	true	moderate	riparian

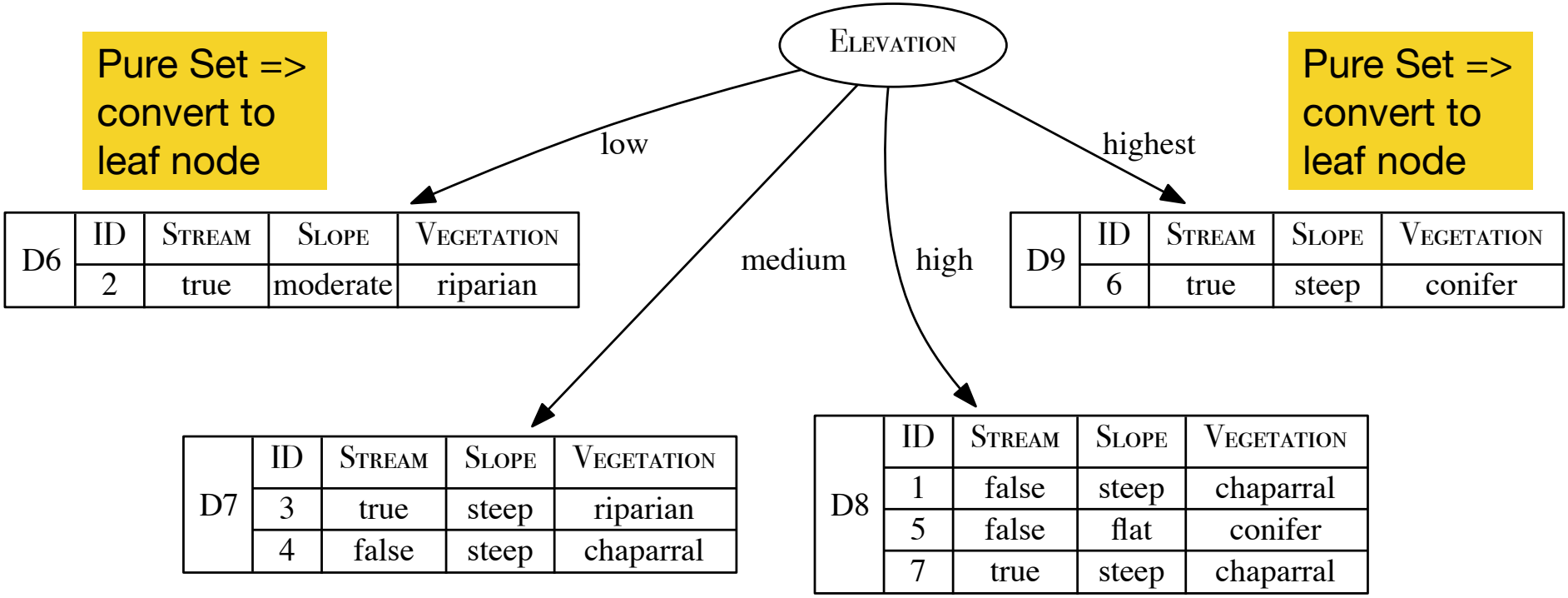
D9	ID	Stream	Slope	Vegetation
	6	true	steep	conifer

D7	ID	Stream	Slope	Vegetation
	3	true	steep	riparian
	4	false	steep	chaparral

D8	ID	Stream	Slope	Vegetation
	1	false	steep	chaparral
	5	false	flat	conifer
	7	true	steep	chaparral

ID	Stream	Slope	Elevation	Vegetation
1	false	steep	high	chaparral
2	true	moderate	low	riparian
3	true	steep	medium	riparian
4	false	steep	medium	chaparral
5	false	flat	high	conifer
6	true	steep	highest	conifer
7	true	steep	high	chaparral

Pure Set => convert to leaf node



D6	ID	Stream	Slope	Vegetation
	2	true	moderate	riparian

D7	ID	Stream	Slope	Vegetation
	3	true	steep	riparian
	4	false	steep	chaparral

D8	ID	Stream	Slope	Vegetation
	1	false	steep	chaparral
	5	false	flat	conifer
	7	true	steep	chaparral

D9	ID	Stream	Slope	Vegetation
	6	true	steep	conifer

Pure Set => convert to leaf node

ID	Stream	Slope	Elevation	Vegetation
1	false	steep	high	chaparral
2	true	moderate	low	riparian
3	true	steep	medium	riparian
4	false	steep	medium	chaparral
5	false	flat	high	conifer
6	true	steep	highest	conifer
7	true	steep	high	chaparral

$$
\begin{aligned}
&H\left(\mathcal{D}_7\right) \\
&= - \sum_{l \in \left\{ \begin{array}{l} \text{chaparral}, \\ \text{riparian}, \\ \text{conifer} \end{array} \right\}} P(\text{Veg} = l) \times log_2\left(P(\text{Veg} = l)\right) \\
&= -\left(\left({}^1/_2 \times log_2({}^1/_2)\right) + \left({}^1/_2 \times log_2({}^1/_2)\right) + \left({}^0/_2 \times log_2({}^0/_2)\right)\right) \\
&= 1.0
\end{aligned}
$$

	ID	STREAM	SLOPE	VEGETATION
D7	3	true	steep	riparian
	4	false	steep	chaparral

$$H\left(\mathcal{D}_7\right) = 1.0$$

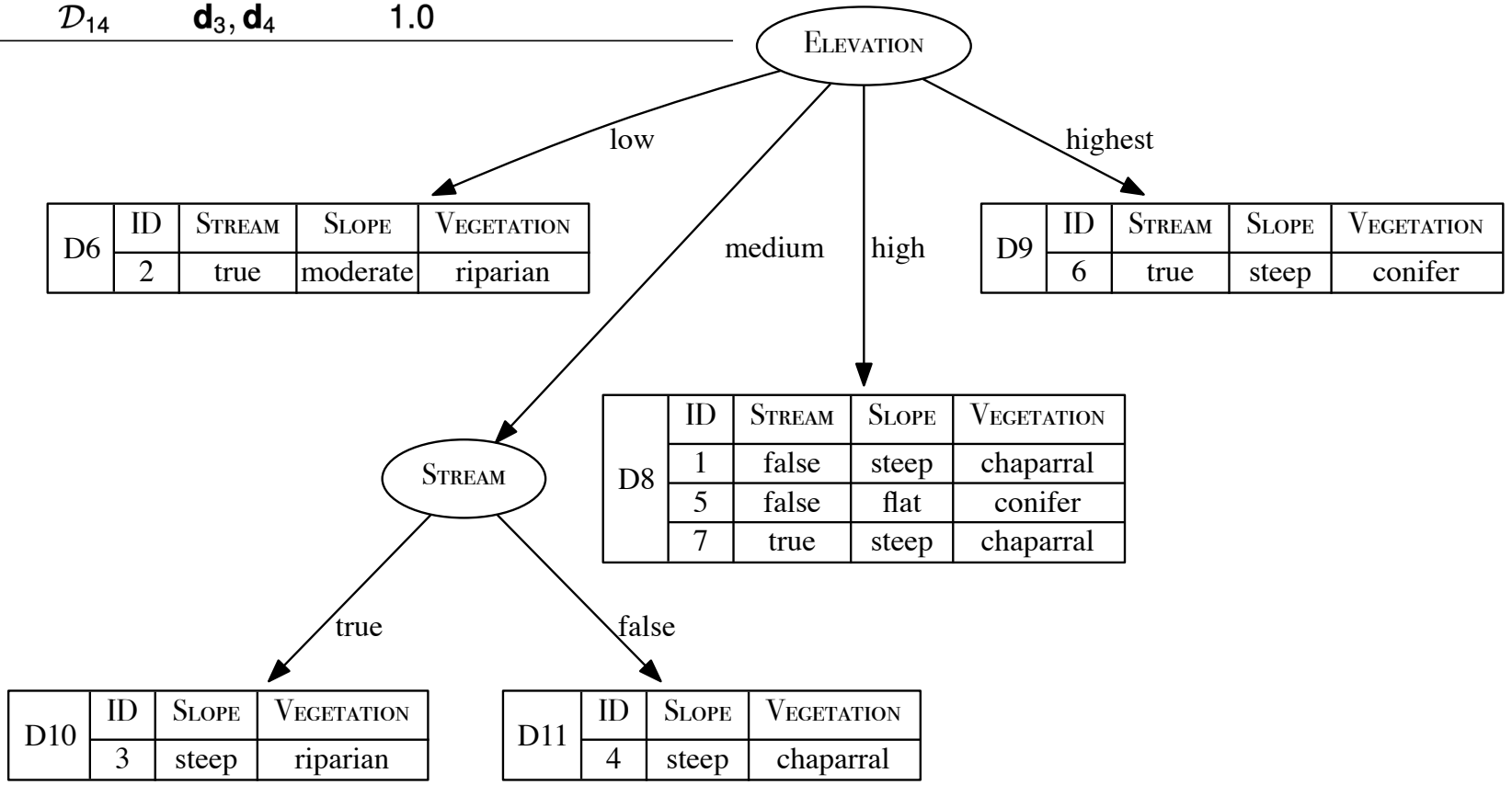
Split By Feature	Level	Part.	Instances	Partition Entropy	Rem.	Info. Gain
STREAM	*'true'*	$\mathcal{D}_{10}$	$\mathbf{d}_3$	0	0	1.0
	'false'	$\mathcal{D}_{11}$	$\mathbf{d}_4$	0		
SLOPE	*'flat'*	$\mathcal{D}_{12}$		0	1.0	0
	'moderate'	$\mathcal{D}_{13}$		0		
	'steep'	$\mathcal{D}_{14}$	$\mathbf{d}_3, \mathbf{d}_4$	1.0		

What feature do we split D7 on?

D7	ID	Stream	Slope	Vegetation
	3	true	steep	riparian
	4	false	steep	chaparral

$$H\left(\mathcal{D}_7\right) = 1.0$$

Split By Feature	Level	Part.	Instances	Partition Entropy	Rem.	Info. Gain
Stream	*'true'*	$\mathcal{D}_{10}$	$\mathbf{d}_3$	0	0	1.0
	'false'	$\mathcal{D}_{11}$	$\mathbf{d}_4$	0		
Slope	*'flat'*	$\mathcal{D}_{12}$		0	1.0	0
	'moderate'	$\mathcal{D}_{13}$		0		
	'steep'	$\mathcal{D}_{14}$	$\mathbf{d}_3, \mathbf{d}_4$	1.0		



D6	ID	Stream	Slope	Vegetation
	2	true	moderate	riparian

D9	ID	Stream	Slope	Vegetation
	6	true	steep	conifer

D8	ID	Stream	Slope	Vegetation
	1	false	steep	chaparral
	5	false	flat	conifer
	7	true	steep	chaparral

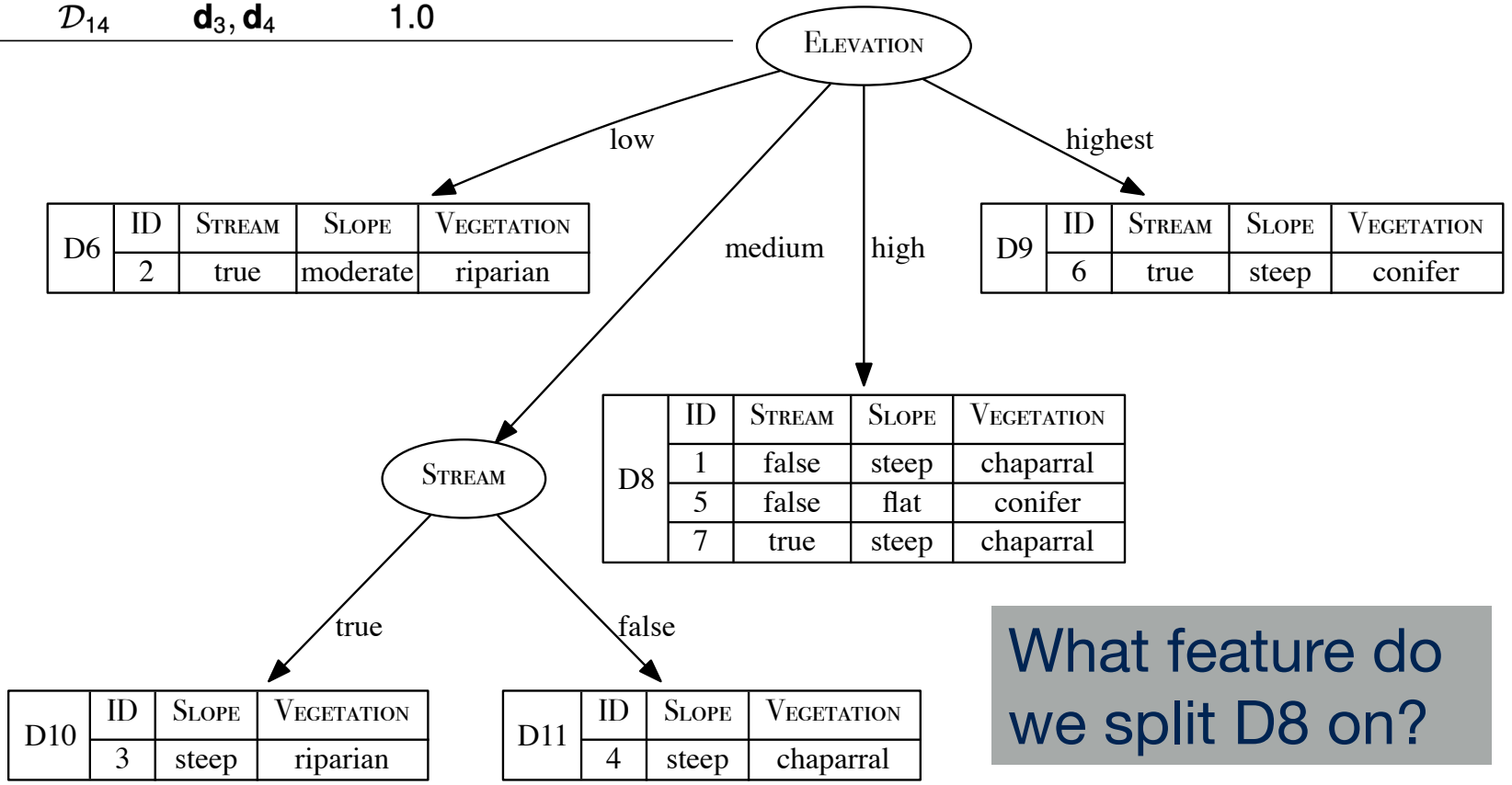
D10	ID	Slope	Vegetation
	3	steep	riparian

D11	ID	Slope	Vegetation
	4	steep	chaparral

D7	ID	Stream	Slope	Vegetation
	3	true	steep	riparian
	4	false	steep	chaparral

$$H\left(\mathcal{D}_7\right) = 1.0$$

Split By Feature	Level	Part.	Instances	Partition Entropy	Rem.	Info. Gain
Stream	*'true'*	$\mathcal{D}_{10}$	$\mathbf{d}_3$	0	0	1.0
	'false'	$\mathcal{D}_{11}$	$\mathbf{d}_4$	0		
Slope	*'flat'*	$\mathcal{D}_{12}$		0	1.0	0
	'moderate'	$\mathcal{D}_{13}$		0		
	'steep'	$\mathcal{D}_{14}$	$\mathbf{d}_3, \mathbf{d}_4$	1.0		



D6	ID	Stream	Slope	Vegetation
	2	true	moderate	riparian

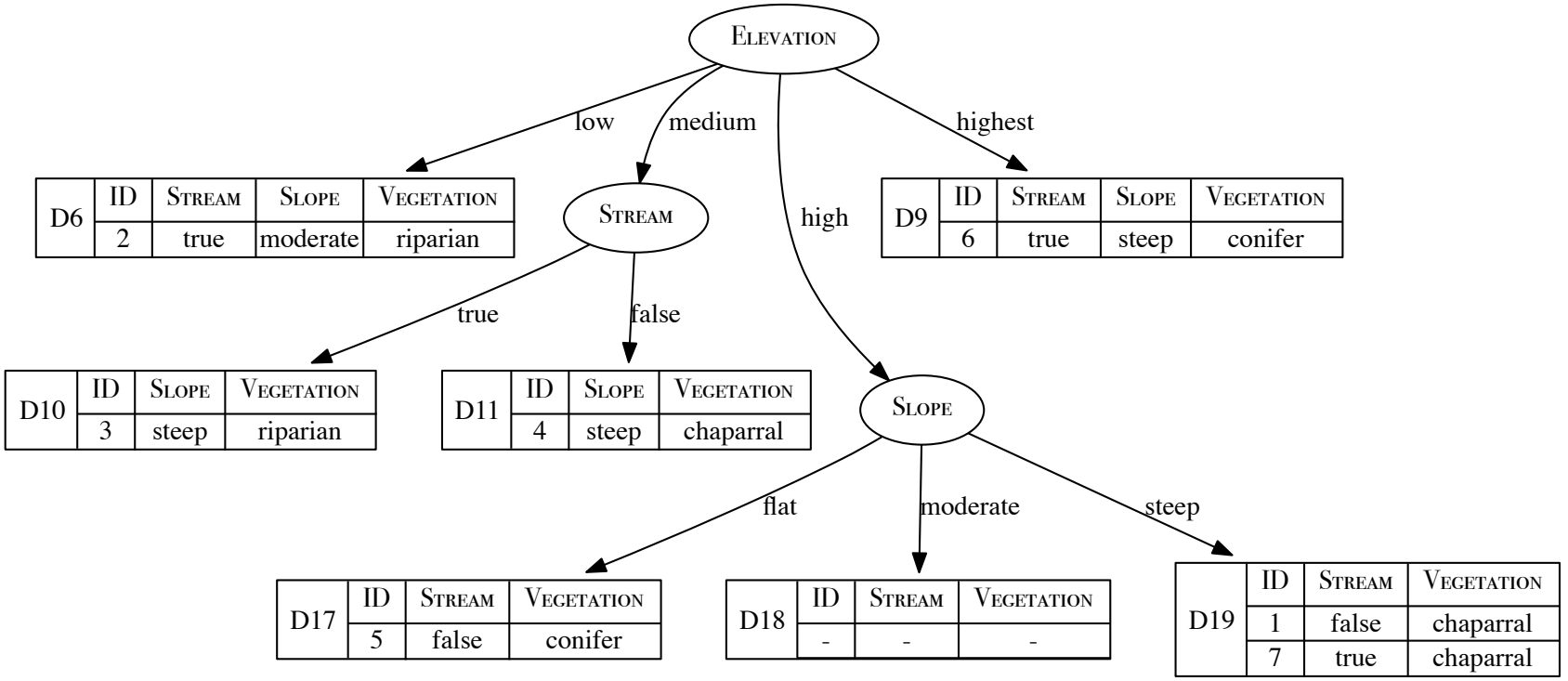
D9	ID	Stream	Slope	Vegetation
	6	true	steep	conifer

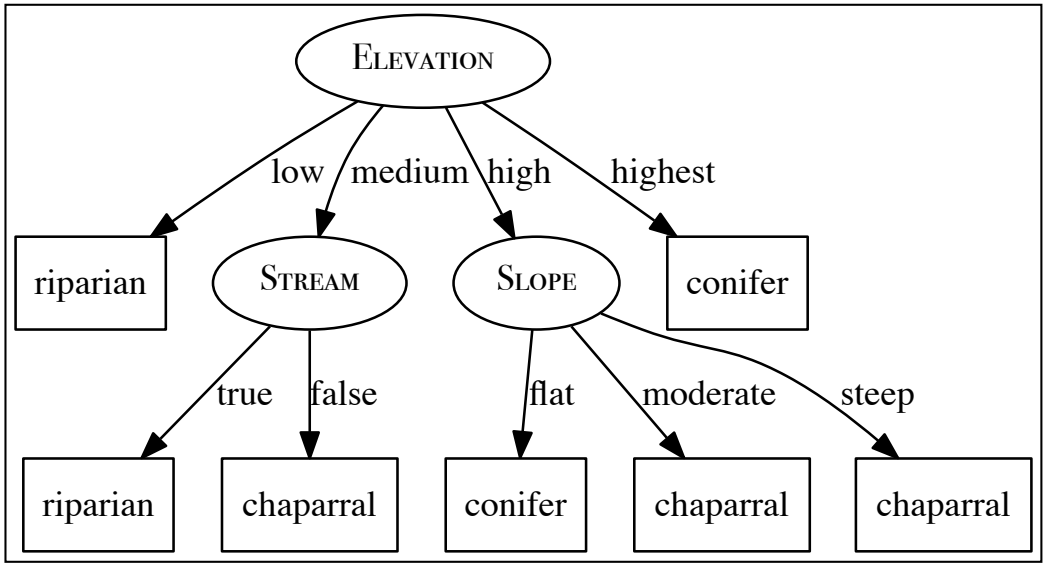
D8	ID	Stream	Slope	Vegetation
	1	false	steep	chaparral
	5	false	flat	conifer
	7	true	steep	chaparral

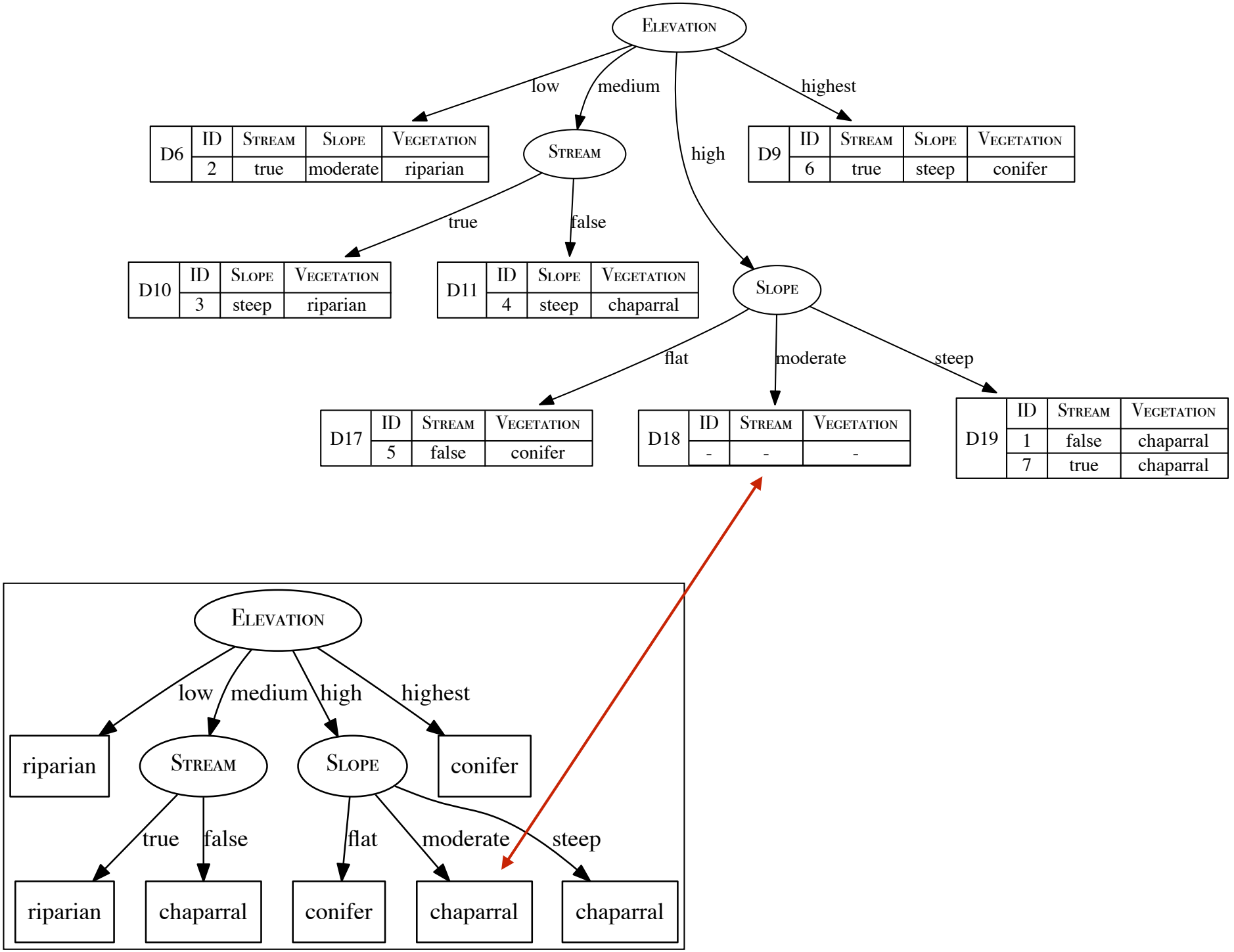
D10	ID	Slope	Vegetation
	3	steep	riparian

D11	ID	Slope	Vegetation
	4	steep	chaparral

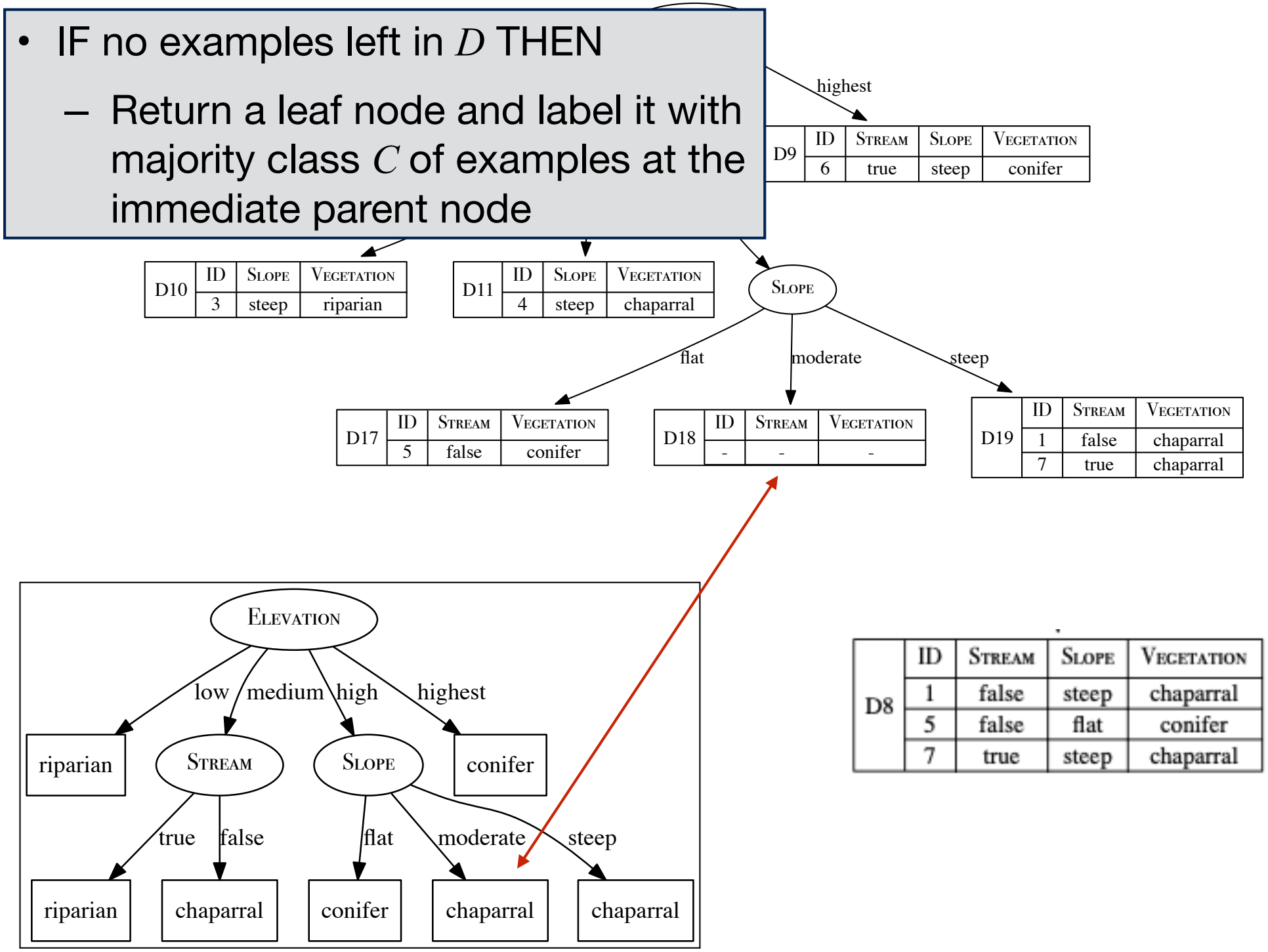
What feature do we split D8 on?


Elevation
low
medium
highest
high
Stream
true
false
Slope
flat
moderate
steep
D6
ID Stream Slope Vegetation
2 true moderate riparian
D9
ID Stream Slope Vegetation
6 true steep conifer
D10
ID Slope Vegetation
3 steep riparian
D11
ID Slope Vegetation
4 steep chaparral
D17
ID Stream Vegetation
5 false conifer
D18
ID Stream Vegetation
- - -
D19
ID Stream Vegetation
1 false chaparral
7 true chaparral

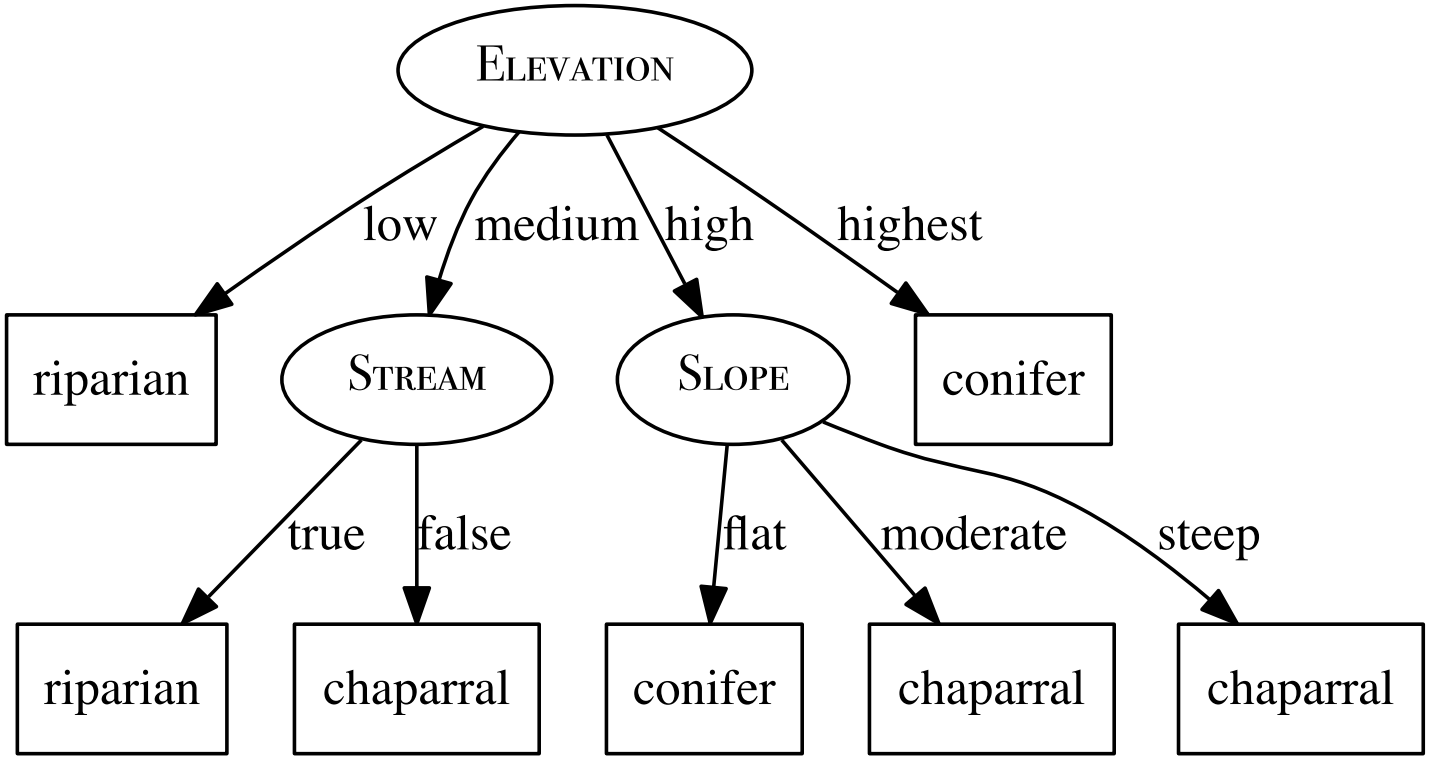

Elevation
low
medium
high
highest
riparian
Stream
Slope
conifer
true
false
flat
moderate
steep
riparian
chaparral
conifer
chaparral
chaparral


Elevation
low
medium
high
highest
Stream
true
false
Slope
flat
moderate
steep
D6
ID Stream Slope Vegetation
2 true moderate riparian
D9
ID Stream Slope Vegetation
6 true steep conifer
D10
ID Slope Vegetation
3 steep riparian
D11
ID Slope Vegetation
4 steep chaparral
D17
ID Stream Vegetation
5 false conifer
D18
ID Stream Vegetation
- - -
D19
ID Stream Vegetation
1 false chaparral
7 true chaparral
Elevation
low
medium
high
highest
riparian
Stream
Slope
conifer
true
false
flat
moderate
steep
riparian
chaparral
conifer
chaparral
chaparral

- IF no examples left in D THEN
  - Return a leaf node and label it with majority class C of examples at the immediate parent node



Using the tree for Prediction



- What prediction would the tree return for the query?

  Stream = ‘true’, Slope = ‘moderate’, Elevation = ‘high’

ID3 Algorithm

Set of training examples D
Set of descriptive features $\mathbf{d}$

- IF all examples in D belong to the same class C THEN
  - Return a leaf node and label it with class C
- IF no features left in D THEN
  - Return a leaf node and label it with majority class C of D
- IF no examples left in D THEN
  - Return a leaf node and label it with majority class C of examples at the immediate parent node
- ELSE
  - Select a feature d_i from $\mathbf{d}$ based on some feature selection criterion
  - Generate a tree node with d_i as the test feature
  - FOR EACH value v_j of d_i
    - Let $D_j \subset D$ contain all examples with $d_i = v_j$
    - Build a subtree by applying $\mathrm{ID3}(D_j)$

Different feature selection criteria

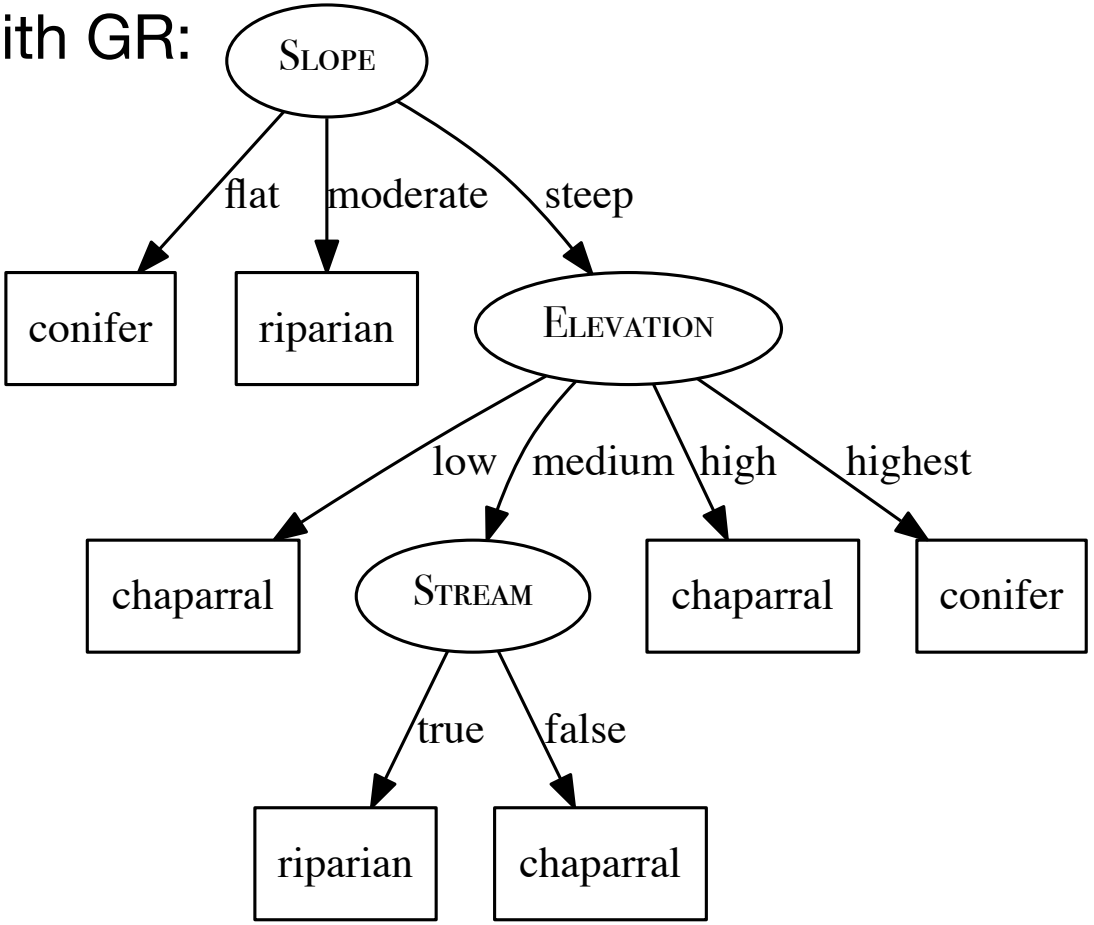
- Information Gain prefers features with many labels as it will split the data into small subsets, which will tend to be pure, irrespective of any correlation between the feature and the target
- Information Gain Ratio:
  - Divides the IG of a feature d by the amount of information used to determine the value of the feature (i.e. the entropy of the dataset wrt the feature d)

$$GR(d, \mathcal{D}) = \frac{IG(d, \mathcal{D})}{-\sum_{l \in labels(d)} (P(d = l) \times log_2(P(d = l)))}$$

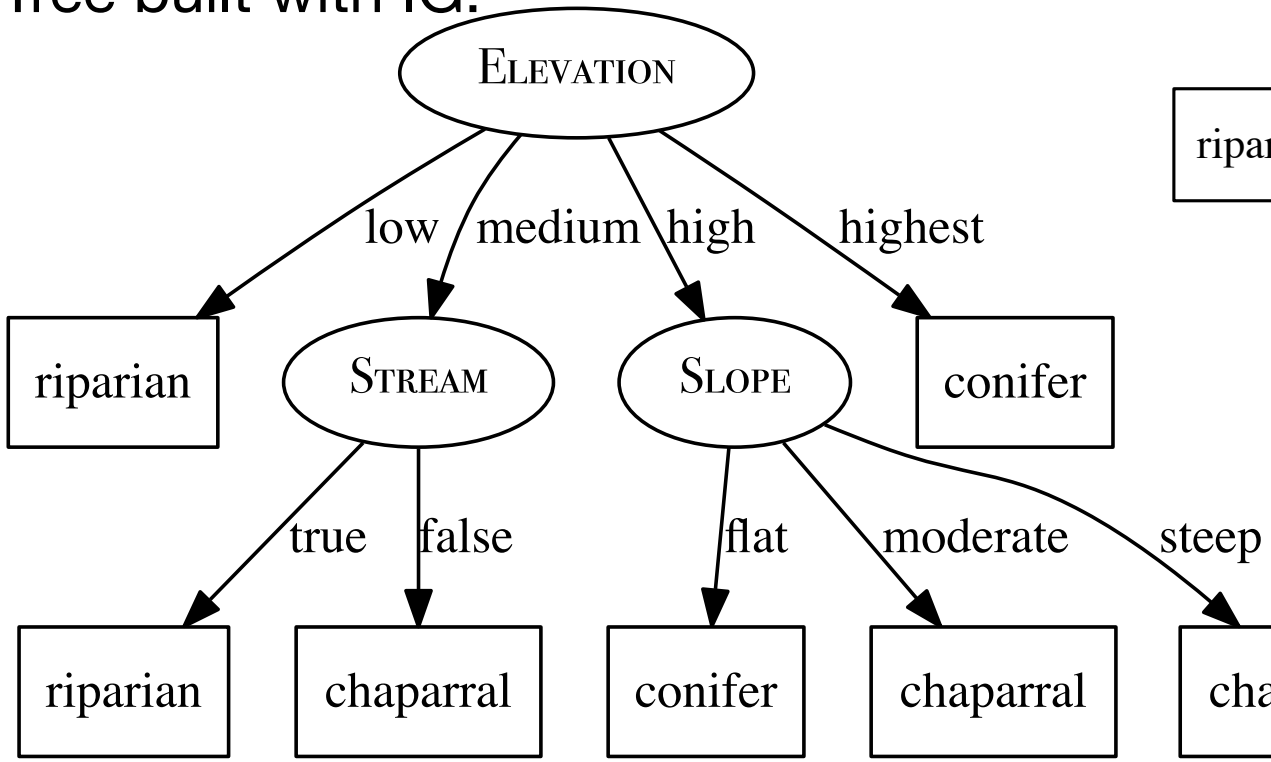
- GR addresses the bias IG has towards features with large numbers of values as the divisor biases away from these types of features

Tree built with GR:

$$GR(\text{STREAM}, \mathcal{D}) = \frac{0.3060}{0.9852} = 0.3106$$
$$GR(\text{SLOPE}, \mathcal{D}) = \frac{0.5774}{1.1488} = 0.5026$$
$$GR(\text{ELEVATION}, \mathcal{D}) = \frac{0.8774}{1.8424} = 0.4762$$



Tree built with IG:



$$IG(\text{STREAM}, \mathcal{D}) = 0.3060$$
$$IG(\text{SLOPE}, \mathcal{D}) = 0.5774$$
$$IG(\text{ELEVATION}, \mathcal{D}) = 0.8774$$

Different feature selection criteria

- Gini Index

$$Gini\,(t, \mathcal{D}) = 1 - \sum_{l \in labels(t)} P(t = l)^2$$

where $P(t=l)$ is prob of an instance having target label l

- Gini index can be thought of as calculating how often you would misclassify an instance in a dataset if you classified it based on the distribution of target labels in the dataset
- IG can be calculated by replacing entropy with the Gini index
- CART algorithm (variant of ID3) uses the Gini index

Handling Continuous Descriptive Features

- Turn them into boolean features based on a threshold

- To find the threshold:
  1. Sort according to feature values
  2. Adjacent instances that have different classifications are potential thresholds
  3. Compute IG for each potential threshold
  4. Select one with highest IG as actual threshold

- New dynamically created boolean feature competes with other features for selection as the splitting feature for a node
- Repeat as needed as the tree is built

Example: Continuous Descriptive Features

ID	Stream	Slope	Elevation	Vegetation
1	false	steep	3 900	chapparal
2	true	moderate	300	riparian
3	true	steep	1 500	riparian
4	false	steep	1 200	chapparal
5	false	flat	4 450	conifer
6	true	steep	5 000	conifer
7	true	steep	3 000	chapparal

1

ID	Stream	Slope	Elevation	Vegetation
2	true	moderate	300	riparian
4	false	steep	1 200	chapparal
3	true	steep	1 500	riparian
7	true	steep	3 000	chapparal
1	false	steep	3 900	chapparal
5	false	flat	4 450	conifer
6	true	steep	5 000	conifer

Example: Continuous Descriptive Features

ID	Stream	Slope	Elevation	Vegetation
1	false	steep	3 900	chapparal
2	true	moderate	300	riparian
3	true	steep	1 500	riparian
4	false	steep	1 200	chapparal
5	false	flat	4 450	conifer
6	true	steep	5 000	conifer
7	true	steep	3 000	chapparal

1

ID	Stream	Slope	Elevation	Vegetation
2	true	moderate	300	riparian
4	false	steep	1 200	chapparal
3	true	steep	1 500	riparian
7	true	steep	3 000	chapparal
1	false	steep	3 900	chapparal
5	false	flat	4 450	conifer
6	true	steep	5 000	conifer

750
1350
2250
4175

2

Example: Continuous Descriptive Features

ID	Stream	Slope	Elevation	Vegetation
1	false	steep	3 900	chapparal
2	true	moderate	300	riparian
3	true	steep	1 500	riparian
4	false	steep	1 200	chapparal
5	false	flat	4 450	conifer
6	true	steep	5 000	conifer
7	true	steep	3 000	chapparal

1

ID	Stream	Slope	Elevation	Vegetation
2	true	moderate	300	riparian
4	false	steep	1 200	chapparal
3	true	steep	1 500	riparian
7	true	steep	3 000	chapparal
1	false	steep	3 900	chapparal
5	false	flat	4 450	conifer
6	true	steep	5 000	conifer

2

750
1350
2250
4175

3

Split by Threshold	Part.	Instances	Partition Entropy	Rem.	Info. Gain
≥750	$\mathcal{D}_1$	$\mathbf{d}_2$	0.0	1.2507	0.3060
	$\mathcal{D}_2$	$\mathbf{d}_4, \mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.4591		
≥1 350	$\mathcal{D}_3$	$\mathbf{d}_2, \mathbf{d}_4$	1.0	1.3728	0.1839
	$\mathcal{D}_4$	$\mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.5219		
≥2 250	$\mathcal{D}_5$	$\mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_3$	0.9183	0.9650	0.5917
	$\mathcal{D}_6$	$\mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.0		
≥4 175	$\mathcal{D}_7$	$\mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1$	0.9710	0.6935	0.8631
	$\mathcal{D}_8$	$\mathbf{d}_5, \mathbf{d}_6$	0.0		

Example: Continuous Descriptive Features

ID	Stream	Slope	Elevation	Vegetation
1	false	steep	3 900	chapparal
2	true	moderate	300	riparian
3	true	steep	1 500	riparian
4	false	steep	1 200	chapparal
5	false	flat	4 450	conifer
6	true	steep	5 000	conifer
7	true	steep	3 000	chapparal

1

ID	Stream	Slope	Elevation	Vegetation
2	true	moderate	300	riparian
4	false	steep	1 200	chapparal
3	true	steep	1 500	riparian
7	true	steep	3 000	chapparal
1	false	steep	3 900	chapparal
5	false	flat	4 450	conifer
6	true	steep	5 000	conifer

750
1350
2250
4175

2

3

Split by Threshold	Part.	Instances	Partition Entropy	Rem.	Info. Gain
≥750	$\mathcal{D}_1$	$\mathbf{d}_2$	0.0	1.2507	0.3060
	$\mathcal{D}_2$	$\mathbf{d}_4, \mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.4591		
≥1 350	$\mathcal{D}_3$	$\mathbf{d}_2, \mathbf{d}_4$	1.0	1.3728	0.1839
	$\mathcal{D}_4$	$\mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.5219		
≥2 250	$\mathcal{D}_5$	$\mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_3$	0.9183	0.9650	0.5917
	$\mathcal{D}_6$	$\mathbf{d}_7, \mathbf{d}_1, \mathbf{d}_5, \mathbf{d}_6$	1.0		
≥4 175	$\mathcal{D}_7$	$\mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_3, \mathbf{d}_7, \mathbf{d}_1$	0.9710	0.6935	0.8631
	$\mathcal{D}_8$	$\mathbf{d}_5, \mathbf{d}_6$	0.0		

Selected Threshold

4

Predicting Continuous Targets

Regression Trees

- Output value is typically the mean of the target feature values of examples in the leaf node
  => error = predicted value - actual target value
- Tree should be built so that the variance of the target feature values at leaf node is minimised
- Measure of impurity at a node is variance

$$var\,(t, \mathcal{D}) = \frac{\sum_{i=1}^{n} (t_i - \bar{t})^2}{n - 1}$$

where n training examples at the node, t_i is the target feature value of example i, and $\bar{t}$ is the mean of target values of n examples

ID3 Algorithm for Continuous Targets

Set of training examples D
Set of descriptive features $\mathbf{d}$

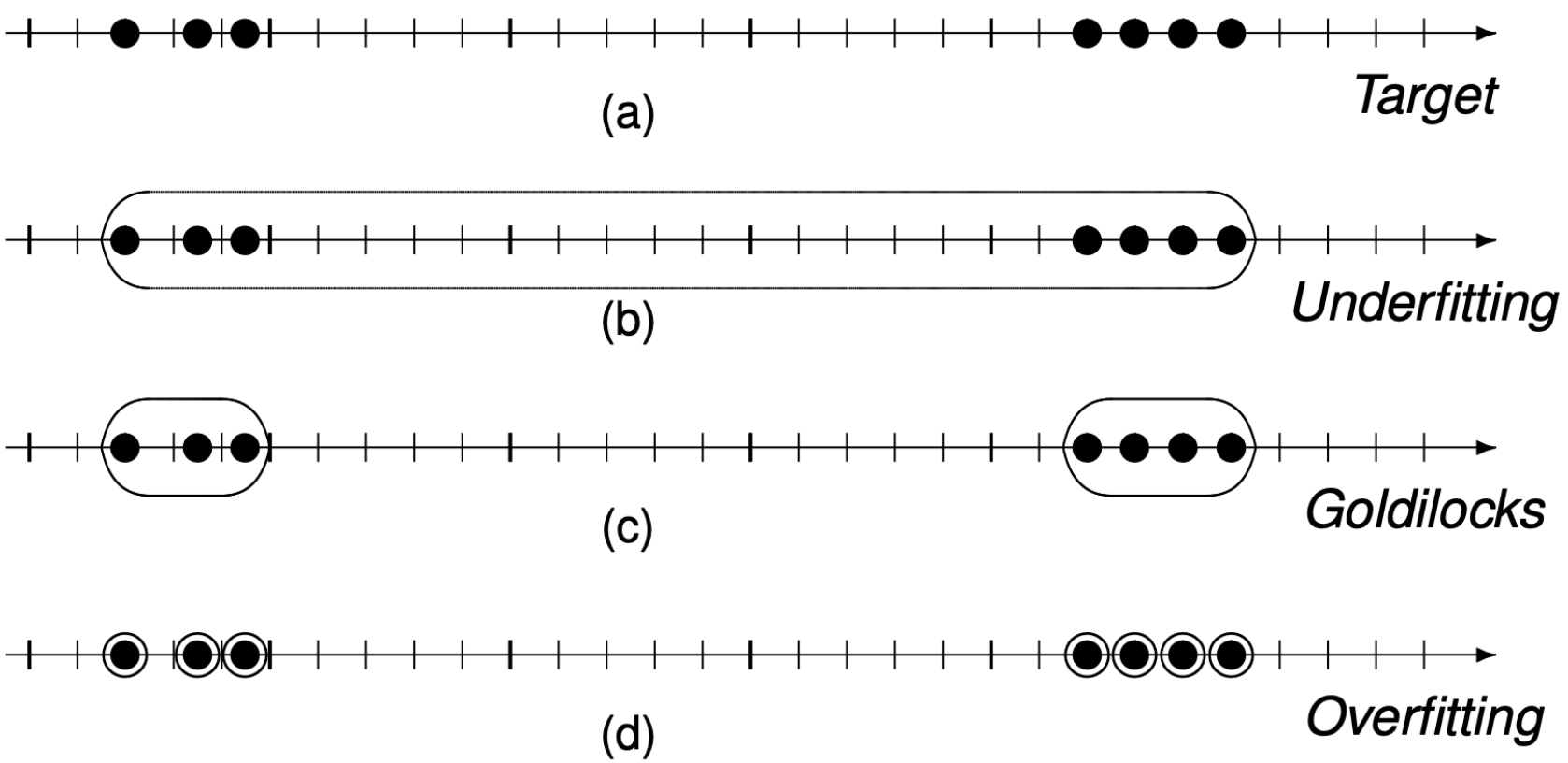
- IF all examples in D belong to the same class C THEN
  - Return a leaf node and label it with class C

??

- IF no features left in D THEN
  - Return a leaf node and label it with average target value of D
- IF no examples left in D THEN
  - Return a leaf node and label it with average target value of examples at the immediate parent node
- ELSE
  - Select a feature d_i from $\mathbf{d}$ based on some feature selection criterion
  - Generate a tree node with d_i as the test feature
  - FOR EACH value v_j of d_i
    - Let $D_j \subset D$ contain all examples with $d_i = v_j$
    - Build a subtree by applying $\mathrm{ID3}(D_j)$

$var\,(d, \mathcal{D})$

Partitioning using variance



- To prevent (d) use an early stopping criterion
  - Stop partitioning if n < some threshold, usually 5% of overall dataset size

Tree Pruning

- Decision trees have a natural tendency to segregate noisy data and create leaf nodes around these instances
- Over-fitting in a decision tree involves splitting data at an irrelevant feature
- Likelihood of over-fitting occurring increases as a tree gets deeper as predictions are based on smaller and smaller subsets
- Pruning the tree identifies and removes sub-trees that are likely to be due to noise and sample variance
  - replace subtree with leaf node covering data partition at that point
  - may result in tree not being consistent with training data but will promote generalisation

Pruning

- Pre-pruning involve Early Stopping Criteria
  - simple approaches e.g. n < some threshold; IG < some threshold (critical value pruning); tree depth > some threshold;
  - statistical significance tests, e.g. χ^2 pruning
  - Computationally efficient but can miss interactions between features than emerge within subtrees
- Post-pruning involves growing tree to completion and then checking each branch for tuning
  - recommended approach is to compare the error rate when subtree is included and excluded on an independent validation set

Summary

- A decision tree is an eager learning algorithm where the model is induced from data in the form of decision rules
- A decision tree model makes predictions based on a sequence of tests on the descriptive features of a query
- Advantages:
  - interpretable
  - can handle both categorical and continuous descriptive features (C4.6 algorithm)
  - relatively robust to noise if pruning is used
- Disadvantages:
  - can become large when dealing with continuous features
  - can overfit if there is a lot of features (high dimensionality)
  - require retraining when modelling concepts that change over time, concept drift