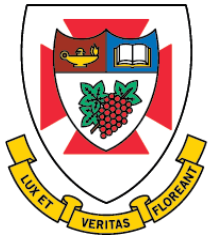
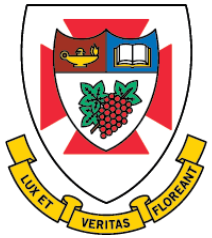


THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

INTRODUCTION TO MACHINE LEARNING

DIT 45100

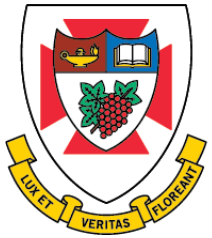


THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

Decision Trees

Agenda

- **Big Idea**
- **Fundamentals**
  - Decision Trees
  - Shannon's Entropy Model
  - Information Gain
- **Standard Approach: The ID3 Algorithm**
  - A Worked Example: Predicting Vegetation Distributions
- **Extensions**
  - Different impurity measures
  - Handling continuous features
  - Predicting continuous targets
  - Guarding against over-fitting
- **Summary**



THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

Big Idea

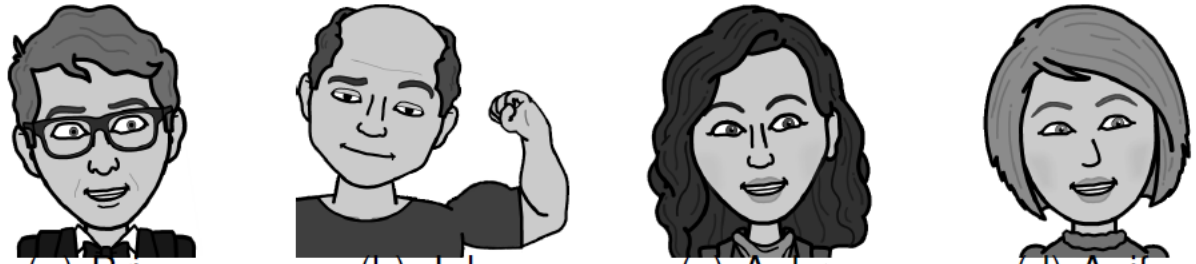


Figure: Cards showing character faces and names for the *Guess-Who* game

Based on these cards, what are the questions with yes/no answers one should ask?

Table: A dataset that represents the characters in the game.

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

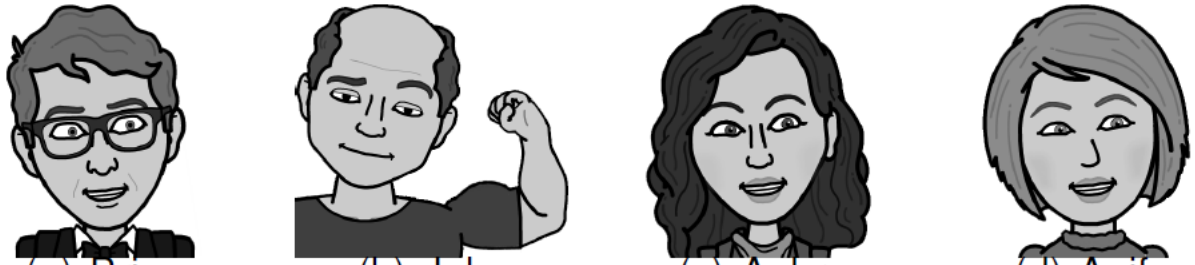


Figure: Cards showing character faces and names for the *Guess-Who* game

.

Which question would you ask first:

1. Is it a man?
2. Does the person wear glasses?

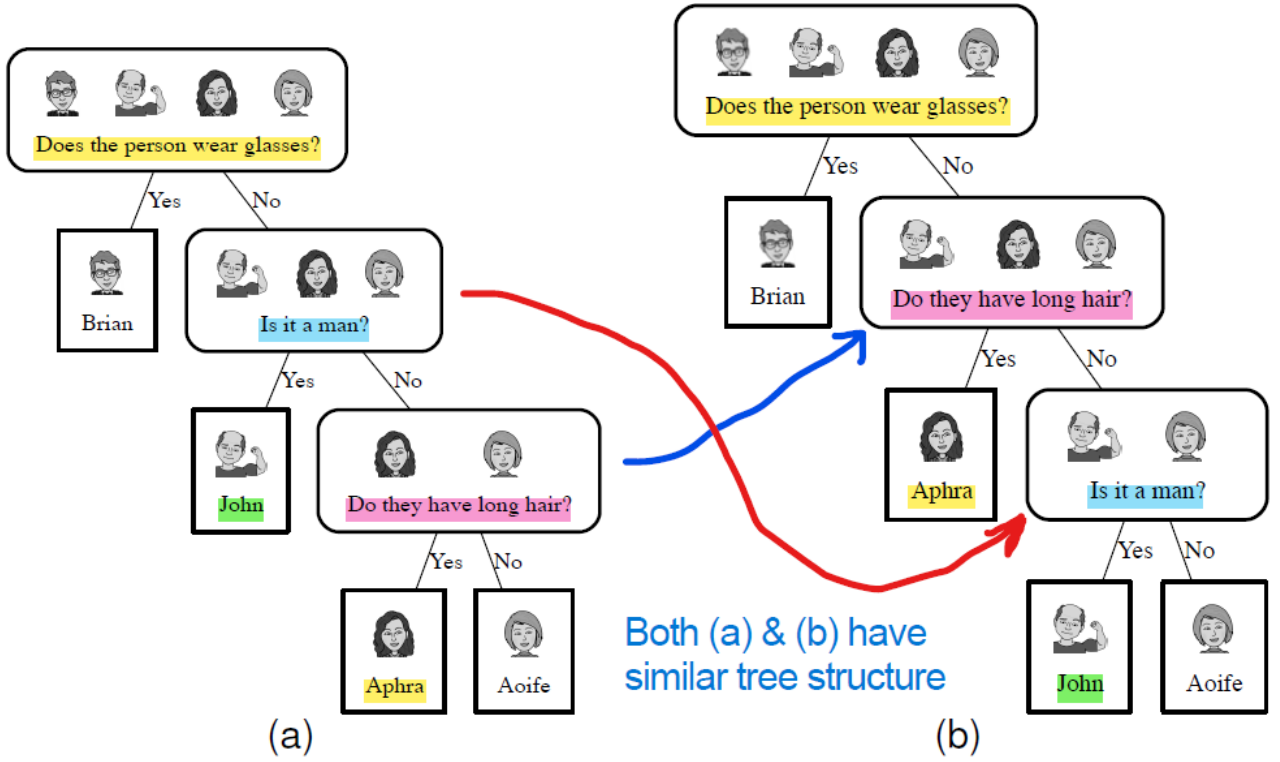


Figure: The different question sequences that can follow in a game of *Guess-Who* beginning with the question Does the person wear glasses?

- In both of the diagrams:
  - one path is 1 question long,
  - one path is 2 questions long,
  - and two paths are 3 questions long.
- Consequently, if you ask Question (2) first the average number of questions you have to ask per game is:

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

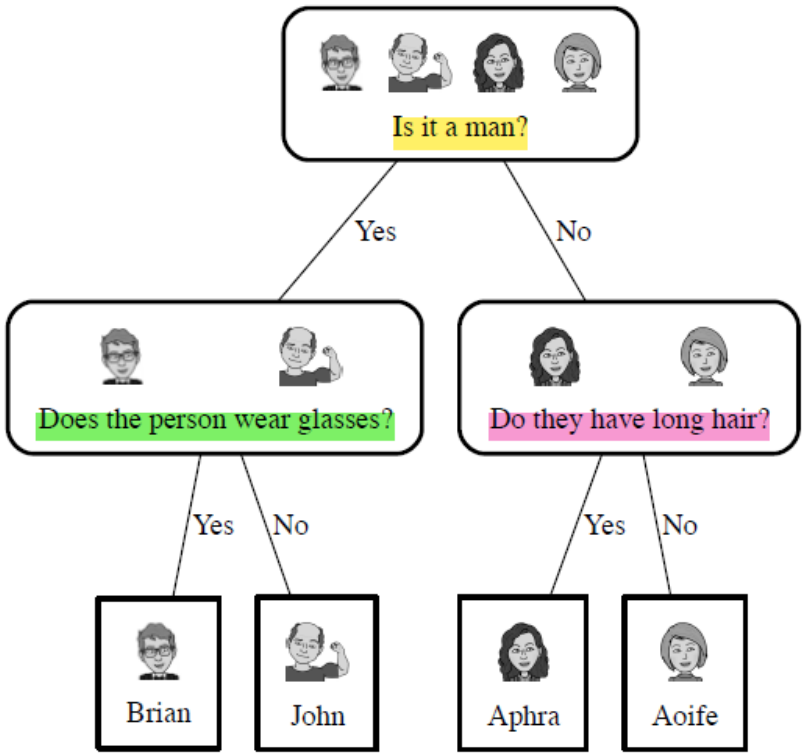


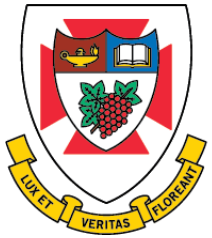
Figure: The different question sequences that can follow in a game of *Guess-Who* beginning with the question Is it a man?

- All the paths in this diagram are two questions long.
- So, on average if you ask Question (1) first the average number of questions you have to ask per game is:

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

Big Idea

- So the big idea here is to figure out which features are the most informative ones to ask questions about by considering the effects of the different answers to the questions, in terms of:
  - how the domain is split up after the answer is received
  - and the likelihood of each of the answers



THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

Fundamentals

Decision Trees

- A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks.
- A decision tree consists of:
  - a **root node** (or starting node),
  - **interior nodes**
  - and **leaf nodes** (or terminating nodes).
- Each of the non-leaf nodes (root and interior) in the tree specifies a test to be carried out on one of the query's descriptive features.
- Each of the leaf nodes specifies a prediction for the query.

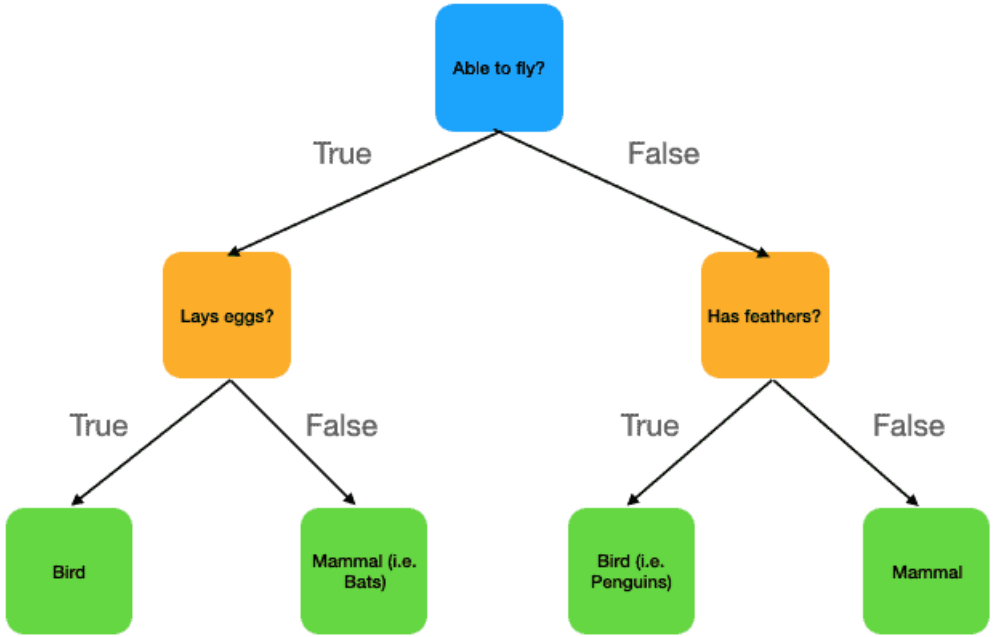


Table: An email spam prediction dataset.

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

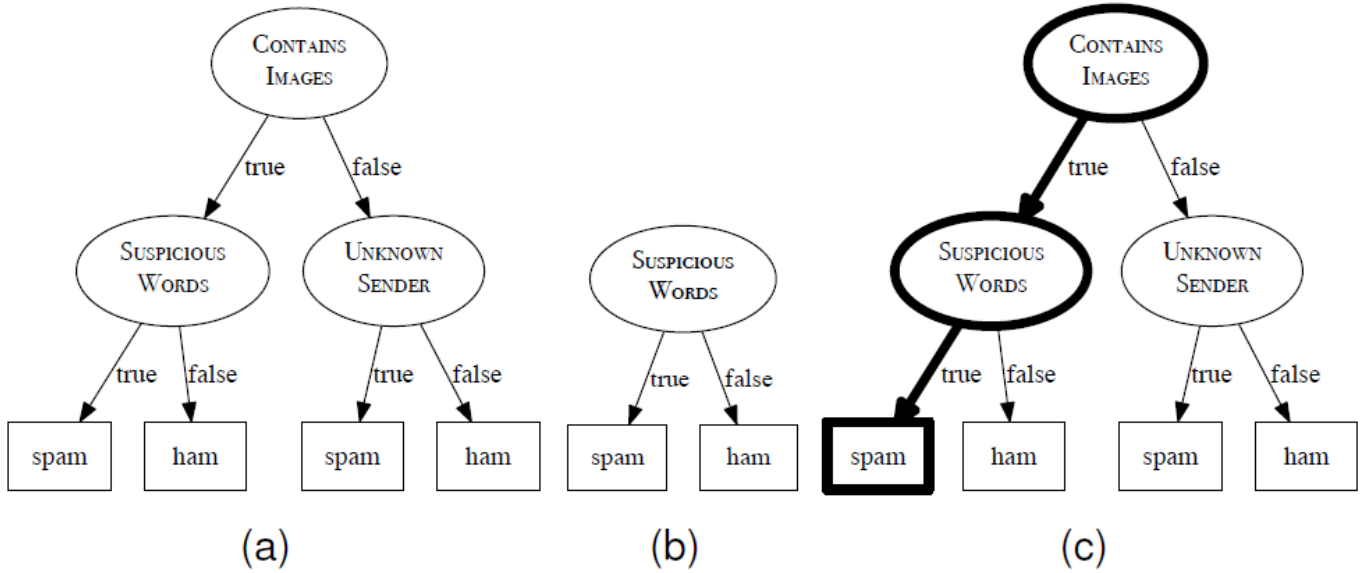


Figure: (a) and (b) show two decision trees that are consistent with the instances in the spam dataset. (c) shows the path taken through the tree shown in (a) to make a prediction for the query instance: SUSPICIOUS WORDS = *'true'*, UNKNOWN SENDER = *'true'*, CONTAINS IMAGES = *'true'*.

How do we create shallow trees?

- The tree that tests SUSPICIOUS WORDS at the root is very shallow because the SUSPICIOUS WORDS feature perfectly splits the data into pure groups of ’spam’ and ’ham’.
- Descriptive features that split the dataset into pure sets with respect to the target feature provide information about the target feature.
- So we can make shallow trees by testing the informative features early on in the tree.
- All we need to do that is a computational metric of the purity of a set: entropy

Shannon's Entropy Model

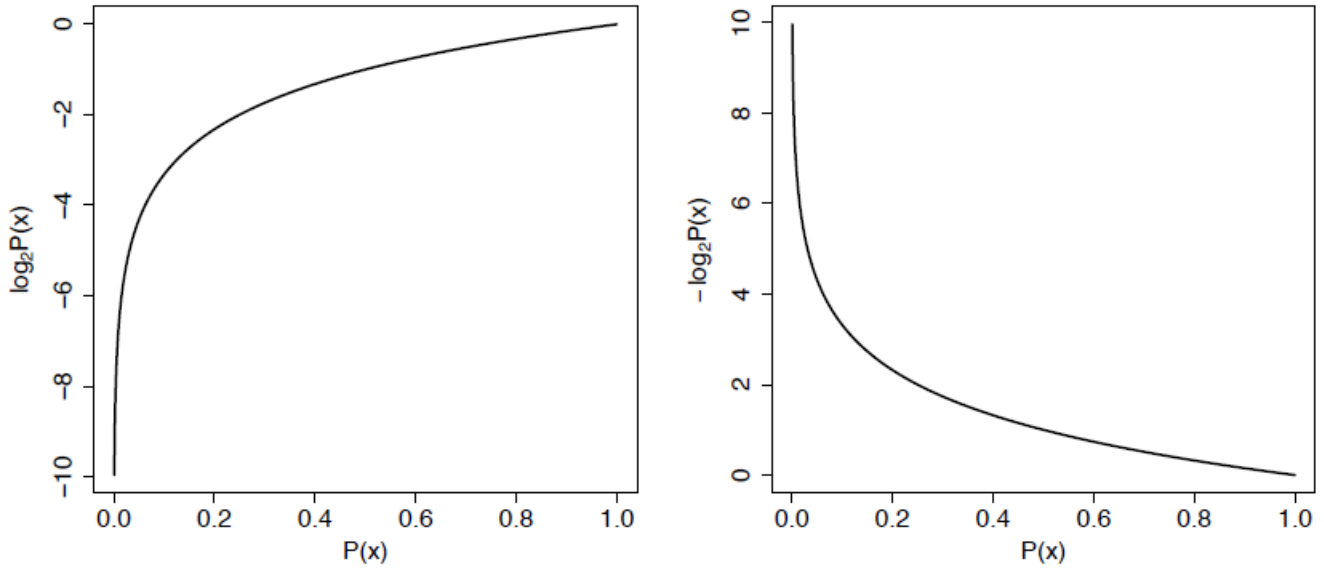


Figure: (a) A graph illustrating how the value of a binary log (the log to the base 2) of a probability changes across the range of probability values. (b) the impact of multiplying these values by −1.

Shannon's Entropy Model

- Shannon's model of entropy is a weighted sum of the logs of the probabilities of each of the possible outcomes when we make a random selection from a set.

$$H(t) = -\sum_{i=1}^{I} (P(t = i) \times log_s(P(t = i))) \qquad (1)$$

P(t = i) is the probability that the outcome of randomly selecting an element t is the type i,
I is the number of different types of things in the set, and
s is an arbitrary logarithmic base.

Entropy

- What is the entropy of a set of 52 different playing cards?

$$
\begin{aligned}
H(card) &= -\sum_{i=1}^{52} P(card = i) \times log_2(P(card = i)) \\
&= -\sum_{i=1}^{52} 0.019 \times log_2(0.019) = -\sum_{i=1}^{52} -0.1096 \\
&= 5.700 \; bits
\end{aligned}
$$

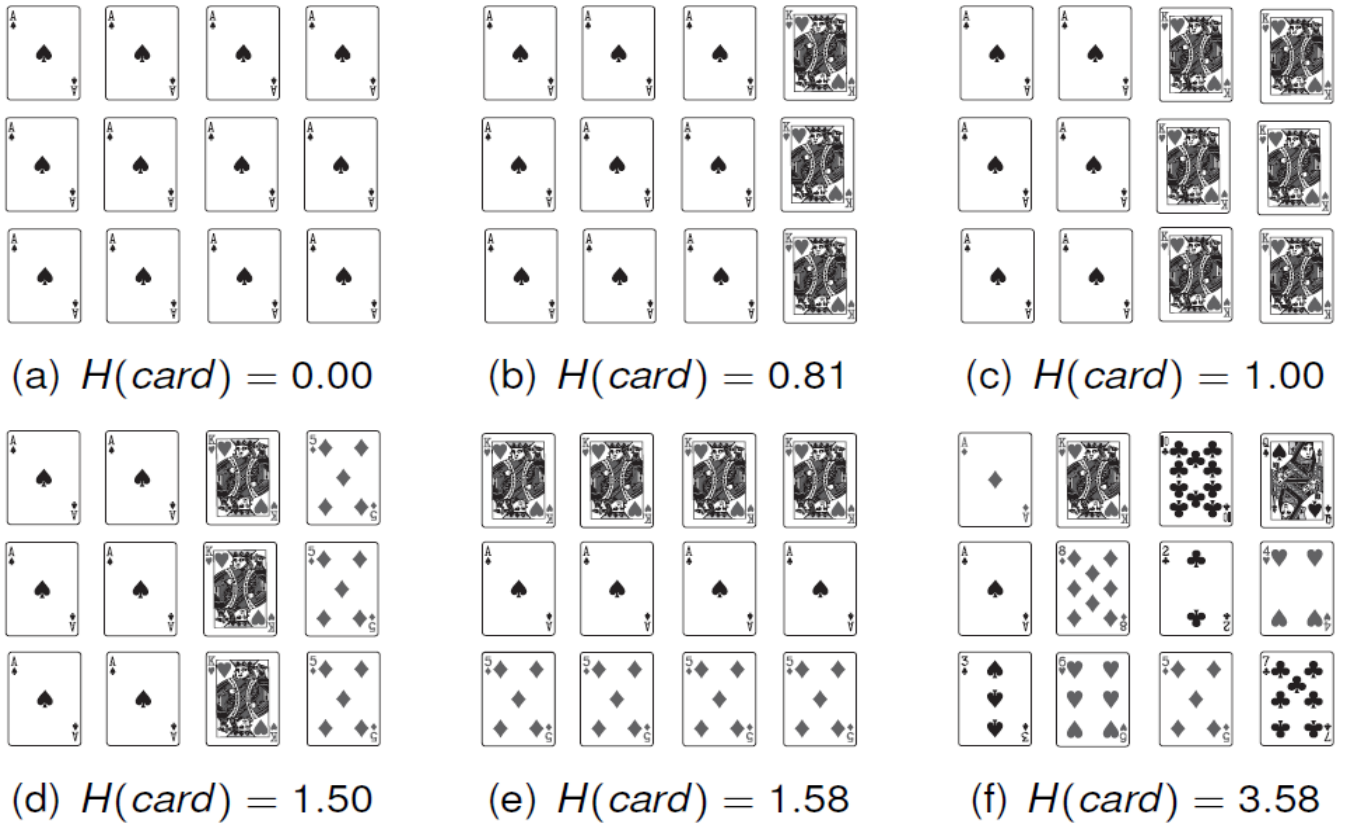
Entropy

- What is the entropy of a set of 52 playing cards if we only distinguish between the cards based on their suit {♥, ♣, ♦, ♠}?

Entropy

$$
\begin{aligned}
H(suit) &= -\sum_{l \in \{\heartsuit,\clubsuit,\diamondsuit,\spadesuit\}} P(suit = l) \times log_2(P(suit = l)) \\
&= -\big((P(\heartsuit) \times log_2(P(\heartsuit))) + (P(\clubsuit) \times log_2(P(\clubsuit))) \\
&\qquad + (P(\diamondsuit) \times log_2(P(\diamondsuit))) + (P(\spadesuit) \times log_2(P(\spadesuit))) \big) \\
&= -\big(\left({}^{13}/_{52} \times log_2({}^{13}/_{52})\right) + \left({}^{13}/_{52} \times log_2({}^{13}/_{52})\right) \\
&\qquad + \left({}^{13}/_{52} \times log_2({}^{13}/_{52})\right) + \left({}^{13}/_{52} \times log_2({}^{13}/_{52})\right) \big) \\
&= -\big((0.25 \times -2) + (0.25 \times -2) \\
&\qquad + (0.25 \times -2) + (0.25 \times -2) \big) \\
&= 2\ bits
\end{aligned}
$$

Entropy



(a) $H(card) = 0.00$ (b) $H(card) = 0.81$ (c) $H(card) = 1.00$

(d) $H(card) = 1.50$ (e) $H(card) = 1.58$ (f) $H(card) = 3.58$

Figure: The entropy of different sets of playing cards measured in bits

Entropy

Table: The relationship between the entropy of a message and the set it was selected from.

Entropy of a Message	Properties of the Message Set
High	A large set of equally likely messages.
Medium	A large set of messages, some more likely than others.
Medium	A small set of equally likely messages.
Low	A small set of messages with one very likely message.

Information Gain

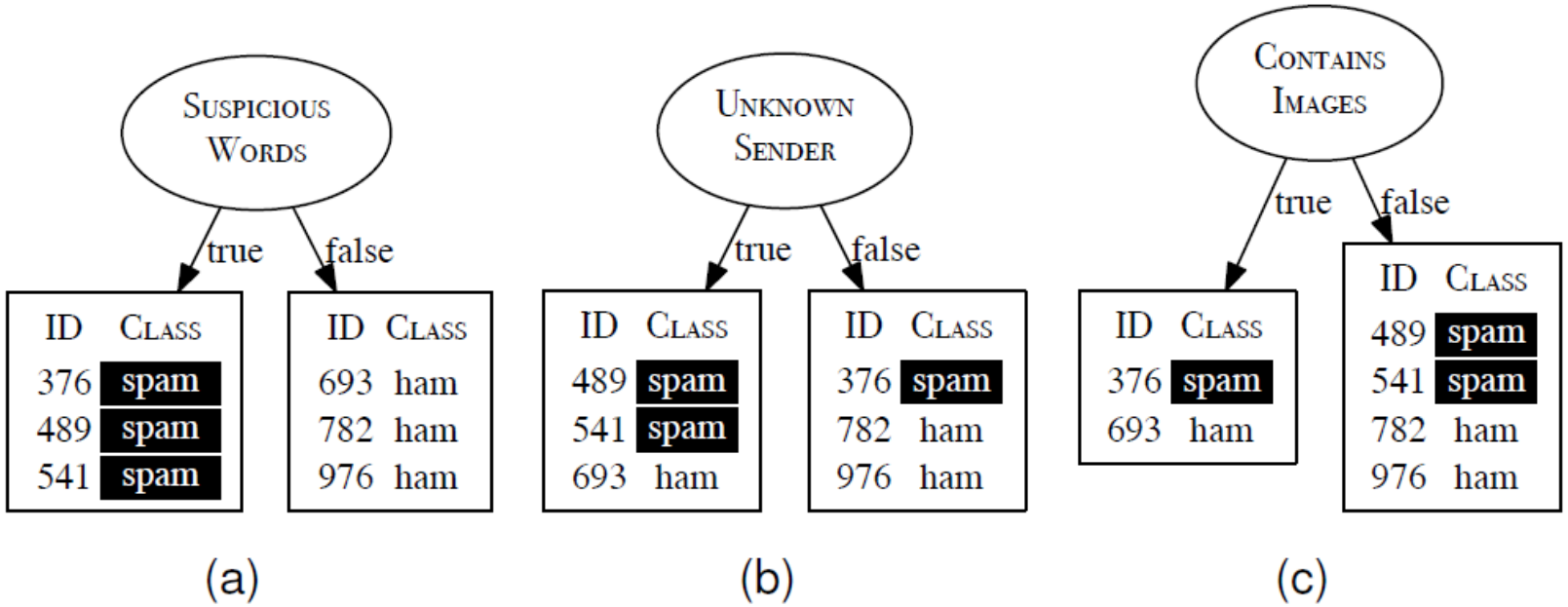


Figure: How the instances in the spam dataset split when we partition using each of the different descriptive features from the spam dataset

Information Gain

- Our intuition is that the ideal discriminatory feature will partition the data into **pure** subsets where all the instances in each subset have the same classification.
  - SUSPICIOUS WORDS perfect split.
  - UNKNOWN SENDER mixture but some information (when ’true’ most instances are ’spam’).
  - CONTAINS IMAGES no information.
- One way to implement this idea is to use a metric called **information gain**.

Information Gain

Information Gain

- The information gain of a descriptive feature can be understood as a measure of the reduction in the overall entropy of a prediction task by testing on that feature.

Information Gain

Computing information gain involves the following 3 equations:

$$H(t, \mathcal{D}) = -\sum_{l \in levels(t)} (P(t = l) \times log_2(P(t = l))) \quad (2)$$

$$rem(d, \mathcal{D}) = \sum_{l \in levels(d)} \underbrace{\frac{|\mathcal{D}_{d=l}|}{|\mathcal{D}|}}_{\text{weighting}} \times \underbrace{H(t, \mathcal{D}_{d=l})}_{\substack{\text{entropy of} \\ \text{partition } \mathcal{D}_{d=l}}} \quad (3)$$

$$IG(d, \mathcal{D}) = H(t, \mathcal{D}) - rem(d, \mathcal{D}) \quad (4)$$

Information Gain

- As an example we will calculate the information gain for each of the descriptive features in the spam email dataset.

Information Gain

- Calculate the entropy for the target feature in the dataset.

$$H(t, \mathcal{D}) = - \sum_{l \in levels(t)} (P(t = l) \times log_2(P(t = l)))$$

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

$$
\begin{aligned}
H(t, \mathcal{D}) &= -\sum_{l \in \{\text{'spam'}, \text{'ham'}\}} (P(t = l) \times log_2(P(t = l))) \\
&= -\ ((P(t = \text{'spam'}) \times log_2(P(t = \text{'spam'})) \\
&\quad + (P(t = \text{'ham'}) \times log_2(P(t = \text{'ham'}))) \\
&= -\left(\left({}^3/_6 \times log_2({}^3/_6)\right) + \left({}^3/_6 \times log_2({}^3/_6)\right)\right) \\
&= 1\ bit
\end{aligned}
$$

Information Gain

- Calculate the remainder for the SUSPICIOUS WORDS feature in the dataset.

$$rem(d, \mathcal{D}) = \sum_{l \in levels(d)} \underbrace{\frac{|\mathcal{D}_{d=l}|}{|\mathcal{D}|}}_{\text{weighting}} \times \underbrace{H(t, \mathcal{D}_{d=l})}_{\substack{\text{entropy of} \\ \text{partition } \mathcal{D}_{d=l}}}$$

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

$$rem(\text{WORDS}, \mathcal{D})$$

$$= \left(\frac{|\mathcal{D}_{\text{WORDS}=T}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{\text{WORDS}=T}) \right) + \left(\frac{|\mathcal{D}_{\text{WORDS}=F}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{\text{WORDS}=F}) \right)$$

$$= \left({}^{3}/_{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t = l) \times log_2(P(t = l)) \right) \right)$$

$$+ \left({}^{3}/_{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t = l) \times log_2(P(t = l)) \right) \right)$$

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

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

Information Gain

- Calculate the remainder for the UNKNOWN SENDER feature in the dataset.

$$rem(d, \mathcal{D}) = \sum_{l \in levels(d)} \underbrace{\frac{|\mathcal{D}_{d=l}|}{|\mathcal{D}|}}_{\text{weighting}} \times \underbrace{H(t, \mathcal{D}_{d=l})}_{\substack{\text{entropy of} \\ \text{partition } \mathcal{D}_{d=l}}}$$

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

$$
\begin{aligned}
&rem\,(\text{SENDER}, \mathcal{D}) \\
&\quad = \left(\frac{|\mathcal{D}_{\text{SENDER}=T}|}{|\mathcal{D}|} \times H\,(t, \mathcal{D}_{\text{SENDER}=T})\right) + \left(\frac{|\mathcal{D}_{\text{SENDER}=F}|}{|\mathcal{D}|} \times H\,(t, \mathcal{D}_{\text{SENDER}=F})\right) \\
&\quad = \left({}^{3}/_{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t = l) \times log_2(P(t = l))\right)\right) \\
&\quad + \left({}^{3}/_{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t = l) \times log_2(P(t = l))\right)\right) \\
&\quad = \left({}^{3}/_{6} \times \left(- \left(\left({}^{2}/_{3} \times log_2({}^{2}/_{3})\right) + \left({}^{1}/_{3} \times log_2({}^{1}/_{3})\right)\right)\right)\right) \\
&\quad + \left({}^{3}/_{6} \times \left(- \left(\left({}^{1}/_{3} \times log_2({}^{1}/_{3})\right) + \left({}^{2}/_{3} \times log_2({}^{2}/_{3})\right)\right)\right)\right) = 0.9183\ bits
\end{aligned}
$$

Information Gain

- Calculate the remainder for the CONTAINS IMAGES feature in the dataset.

$$rem(d, \mathcal{D}) = \sum_{l \in levels(d)} \underbrace{\frac{|\mathcal{D}_{d=l}|}{|\mathcal{D}|}}_{\text{weighting}} \times \underbrace{H(t, \mathcal{D}_{d=l})}_{\substack{\text{entropy of} \\ \text{partition } \mathcal{D}_{d=l}}}$$

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

$$
\begin{aligned}
&rem\,(\text{IMAGES}, \mathcal{D}) \\
&\quad = \left(\frac{|\mathcal{D}_{\text{IMAGES}=T}|}{|\mathcal{D}|} \times H\,(t, \mathcal{D}_{\text{IMAGES}=T})\right) + \left(\frac{|\mathcal{D}_{\text{IMAGES}=F}|}{|\mathcal{D}|} \times H\,(t, \mathcal{D}_{\text{IMAGES}=F})\right) \\
&\quad = \left({}^{2}/_{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t = l) \times log_2(P(t = l))\right)\right) \\
&\quad + \left({}^{4}/_{6} \times \left(- \sum_{l \in \{\text{'spam'}, \text{'ham'}\}} P(t = l) \times log_2(P(t = l))\right)\right) \\
&\quad = \left({}^{2}/_{6} \times \left(- \left(\left({}^{1}/_{2} \times log_2({}^{1}/_{2})\right) + \left({}^{1}/_{2} \times log_2({}^{1}/_{2})\right)\right)\right)\right) \\
&\quad + \left({}^{4}/_{6} \times \left(- \left(\left({}^{2}/_{4} \times log_2({}^{2}/_{4})\right) + \left({}^{2}/_{4} \times log_2({}^{2}/_{4})\right)\right)\right)\right) = 1\ bit
\end{aligned}
$$

Information Gain

- Calculate the information gain for the three descriptive feature in the dataset.

$$IG(d, \mathcal{D}) = H(t, \mathcal{D}) - rem(d, \mathcal{D})$$

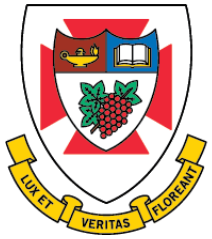
Information Gain

$$IG(\text{SUSPICIOUS WORDS}, \mathcal{D}) = H(\text{CLASS}, \mathcal{D}) - rem(\text{SUSPICIOUS WORDS}, \mathcal{D})$$
$$= 1 - 0 = 1 \; bit$$

$$IG(\text{UNKNOWN SENDER}, \mathcal{D}) = H(\text{CLASS}, \mathcal{D}) - rem(\text{UNKNOWN SENDER}, \mathcal{D})$$
$$= 1 - 0.9183 = 0.0817 \; bits$$

$$IG(\text{CONTAINS IMAGES}, \mathcal{D}) = H(\text{CLASS}, \mathcal{D}) - rem(\text{CONTAINS IMAGES}, \mathcal{D})$$
$$= 1 - 1 = 0 \; bits$$

- The results of these calculations match our intuitions.



THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

Standard Approach: The ID3 Algorithm

ID3 Algorithm

- The ID3 algorithm attempts to create the shallowest tree that is consistent with the data that it is given.
- The ID3 algorithm builds the tree in a recursive, depth-first manner, beginning at the root node and working down to the leaf nodes.

ID3 Algorithm

1. The algorithm begins by choosing the best descriptive feature to test (i.e., the best question to ask first) using **information gain**.
2. A root node is then added to the tree and labelled with the selected test feature.
3. The training dataset is then partitioned using the test.
4. For each partition a branch is grown from the node.
5. The process is then repeated for each of these branches using the relevant partition of the training set in place of the full training set and with the selected test feature excluded from further testing.

ID3 Algorithm

The algorithm defines three situations where the recursion stops and a leaf node is constructed:

1. All of the instances in the dataset have the same classification (target feature value) then return a leaf node tree with that classification as its label.
2. The set of features left to test is empty then return a leaf node tree with the majority class of the dataset as its classification.
3. The dataset is empty return a leaf node tree with the majority class of the dataset at the parent node that made the recursive call.

A Worked Example

Table: The vegetation classification dataset.

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

A Worked Example

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

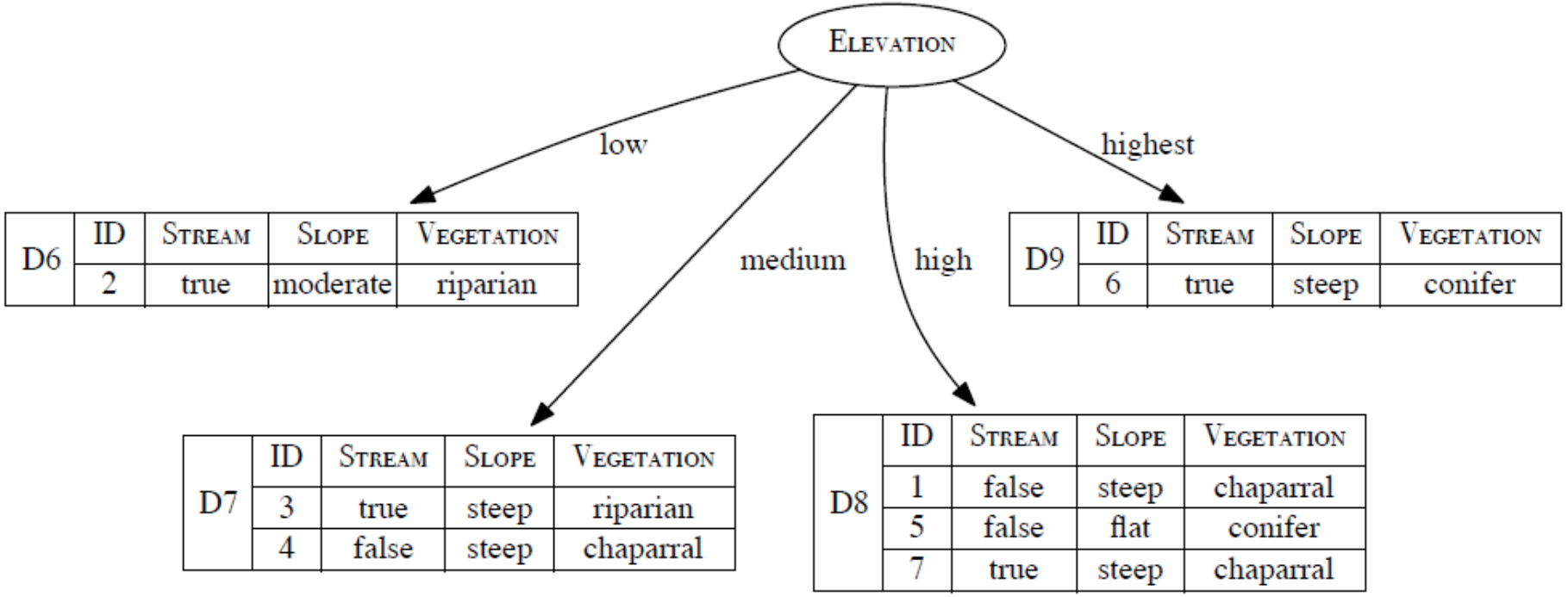
A Worked Example

Table: Partition sets (Part.), entropy, remainder (Rem.) and information gain (Info. Gain) by feature for the dataset

H (VEGETATION, D) = 1.5567 bits

Split By Feature	Level	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		

A Worked Example



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

Figure: The decision tree after the data has been split using ELEVATION.

A Worked Example

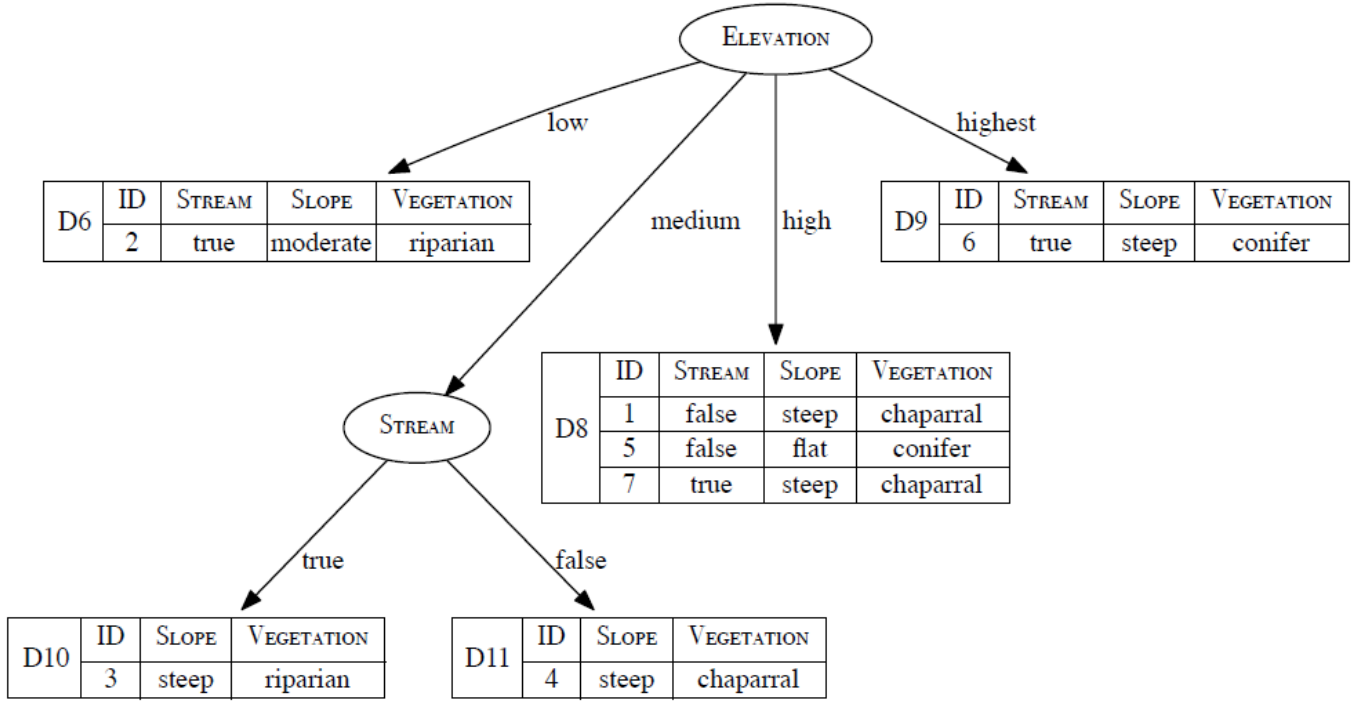
$$
\begin{aligned}
& H(\text{VEGETATION}, \mathcal{D}_7) \\
&= - \sum_{l \in \left\{ \begin{array}{l} \text{'chaparral'}, \\ \text{'riparian'}, \\ \text{'conifer'} \end{array} \right\}} P(\text{VEGETATION} = l) \times log_2 \left(P(\text{VEGETATION} = l) \right) \\
&= - \left(\left({}^1/_2 \times log_2({}^1/_2)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

A Worked Example

Table: Partition sets (Part.), entropy, remainder (Rem.) and information gain (Info. Gain) by feature for the dataset $\mathcal{D}_7$

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		

A Worked Example



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

Figure: The state of the decision tree after the $\mathcal{D}_7$ partition has been split using STREAM.

A Worked Example

$$
\begin{aligned}
& H(\text{VEGETATION}, \mathcal{D}_8) \\
&= - \sum_{I \in \left\{ \begin{array}{l} \text{'chaparral'}, \\ \text{'riparian'}, \\ \text{'conifer'} \end{array} \right\}} P(\text{VEGETATION} = I) \times log_2 \left(P(\text{VEGETATION} = I) \right) \\
&= - \left(\left({}^2/_3 \times log_2({}^2/_3) \right) + \left({}^0/_3 \times log_2({}^0/_3) \right) + \left({}^1/_3 \times log_2({}^1/_3) \right) \right) \\
&= 0.9183 \; bits
\end{aligned}
$$

A Worked Example

Table: Partition sets (Part.), entropy, remainder (Rem.) and information gain (Info. Gain) by by feature for the dataset $\mathcal{D}_8$

Split By Feature	Level	Part.	Instances	Partition Entropy	Rem.	Info. Gain
STREAM	*'true'*	$\mathcal{D}_{15}$	$\mathbf{d}_7$	0	0.6666	0.2517
	'false'	$\mathcal{D}_{16}$	$\mathbf{d}_1, \mathbf{d}_5$	1.0		
SLOPE	*'flat'*	$\mathcal{D}_{17}$	$\mathbf{d}_5$	0	0	0.9183
	'moderate'	$\mathcal{D}_{18}$		0		
	'steep'	$\mathcal{D}_{19}$	$\mathbf{d}_1, \mathbf{d}_7$	0		

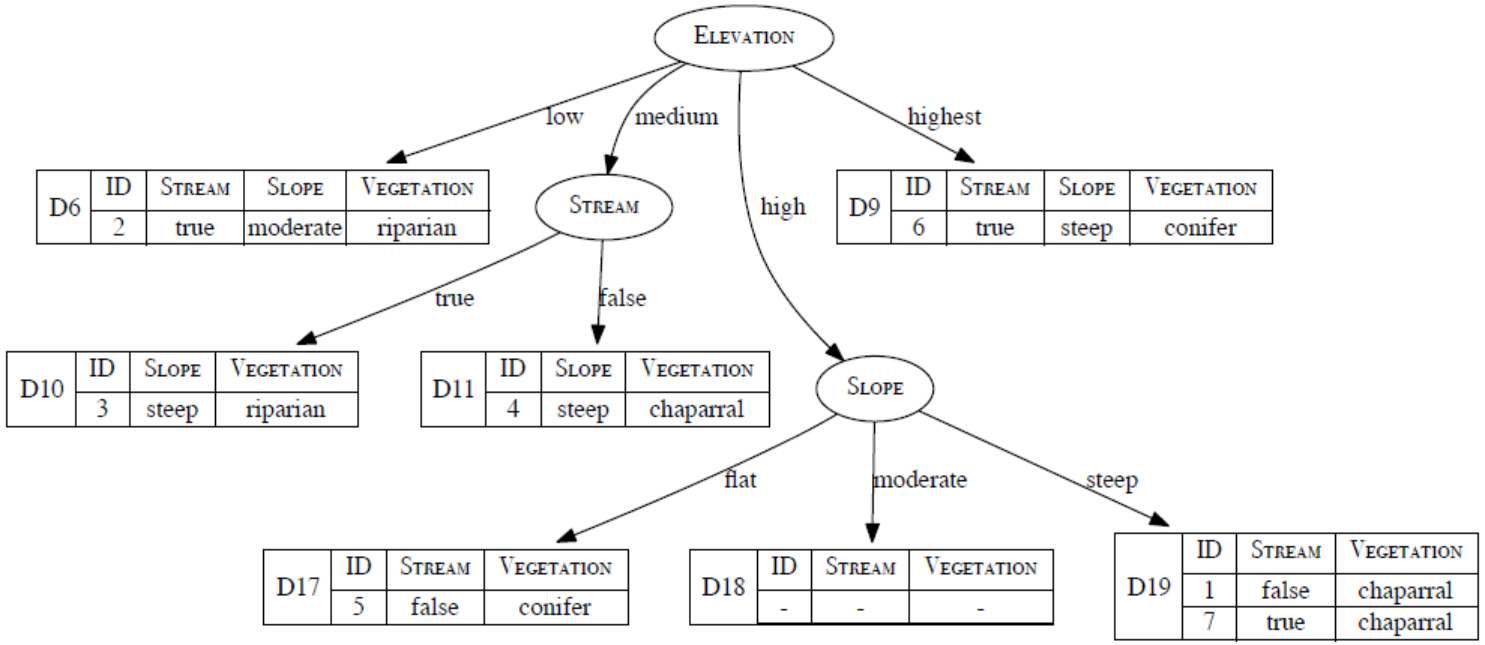
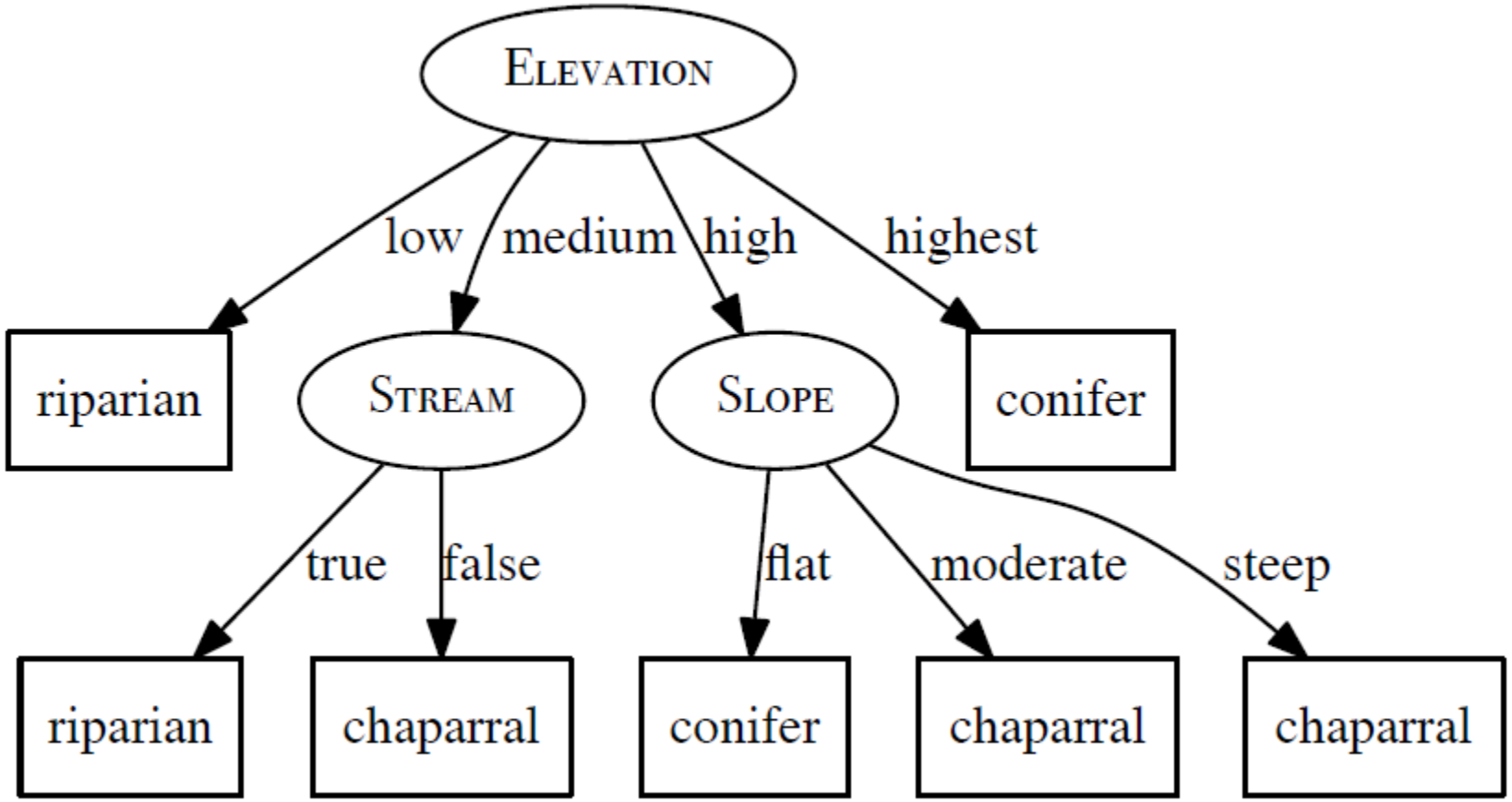
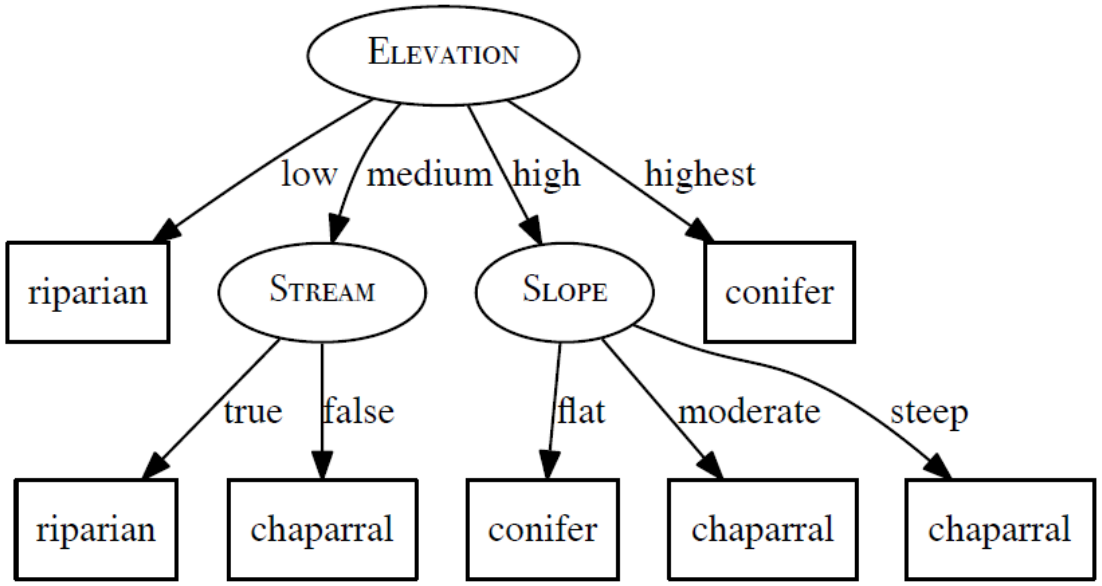


Figure: The state of the decision tree after the $\mathcal{D}_8$ partition has been split using SLOPE.

Vegetation Classification Decision Tree



Vegetation Prediction



- What prediction will this decision tree model return for the following query?

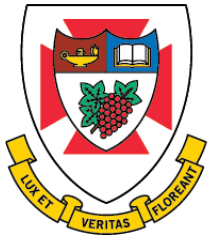
STREAM = *'true'*, SLOPE=*'Moderate'*, ELEVATION=*'High'*

Vegetation Prediction

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

STREAM = *'true'*, SLOPE=*'Moderate'*, ELEVATION=*'High'*

VEGETATION = *'Chaparral'*



THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

Alternative Feature Selection Metrics

Information Gain Ratio

- Entropy based information gain, preferences features with many values.
- One way of addressing this issue is to use **information gain ratio** which is computed by dividing the information gain of a feature by the amount of information used to determine the value of the feature:

$$GR(d, \mathcal{D}) = \frac{IG(d, \mathcal{D})}{-\sum_{l \in levels(d)} (P(d = l) \times log_2(P(d = l)))} \quad (5)$$

Gini Index

- Another commonly used measure of impurity is the **Gini index**:

$$Gini\,(t, \mathcal{D}) = 1 - \sum_{l \in levels(t)} P(t = l)^2 \tag{6}$$

- The Gini index can be thought of as calculating how often you would misclassify an instance in the dataset if you classified it based on the distribution of classifications in the dataset.
- Information gain can be calculated using the Gini index by replacing the entropy measure with the Gini index.

Handling Continuous Descriptive Features

Table 2: Dataset for predicting the vegetation in an area with a continuous ELEVATION feature (measured in feet).

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

Handling Continuous Descriptive Features

Table 3: Dataset for predicting the vegetation in an area sorted by the continuous ELEVATION feature.

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

Handling Continuous Descriptive Features

Table 4: Partition sets (Part.), entropy, remainder (Rem.), and information gain (Info. Gain) for the candidate ELEVATION thresholds: $\geq$750, $\geq$1 350, $\geq$2 250 and $\geq$4 175.

Split by Threshold	Part.	Instances	Partition Entropy	Rem.	Info. Gain
$\geq$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		
$\geq$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		
$\geq$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		
$\geq$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		

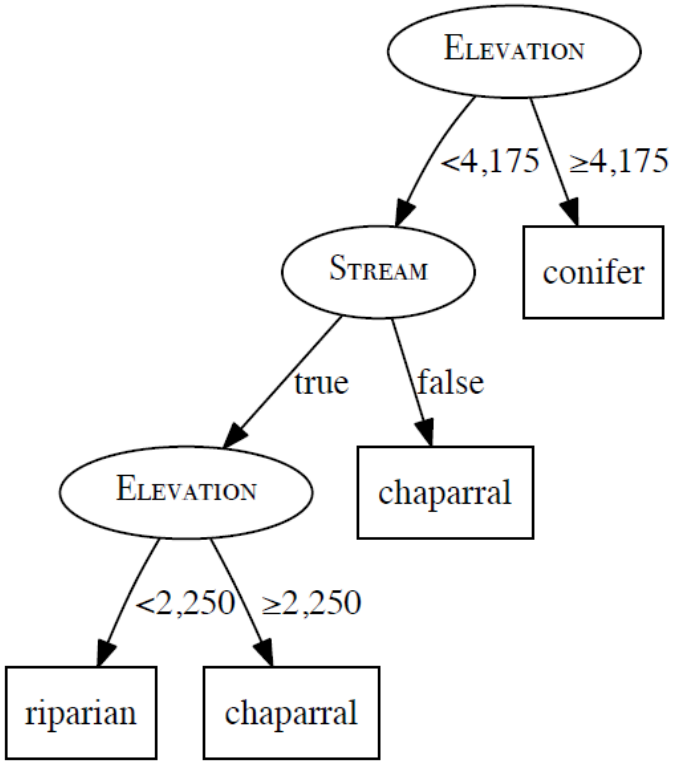
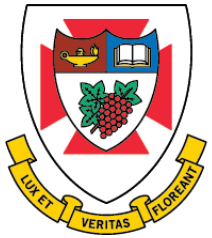


Figure 3: The decision tree that would be generated for the vegetation classification dataset



THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

Predicting Continuous Targets

Regression Trees

- Regression trees are constructed so as to reduce the variance in the set of training examples at each of the leaf nodes in the tree
- We can do this by adapting the ID3 algorithm to use a measure of variance rather than a measure of classification impurity (entropy) when selecting the best attribute

Regression Trees

- The impurity (variance) at a node can be calculated using the following equation:

$$var\,(t, \mathcal{D}) = \frac{\sum_{i=1}^{n} (t_i - \bar{t})^2}{n - 1} \tag{7}$$

- We select the feature to split on at a node by selecting the feature that minimizes the weighted variance across the resulting partitions:

$$\mathbf{d}[best] = \arg\min_{d \in \mathbf{d}} \sum_{l \in levels(d)} \frac{|\mathcal{D}_{d=l}|}{|\mathcal{D}|} \times var(t, \mathcal{D}_{d=l}) \tag{8}$$

Regression Trees

Table 5: A dataset listing the number of bike rentals per day.

ID	Season	Work Day	Rentals
1	winter	false	800
2	winter	false	826
3	winter	true	900
4	spring	false	2 100
5	spring	true	4 740
6	spring	true	4 900
7	summer	false	3 000
8	summer	true	5 800
9	summer	true	6 200
10	autumn	false	2 910
11	autumn	false	2 880
12	autumn	true	2 820

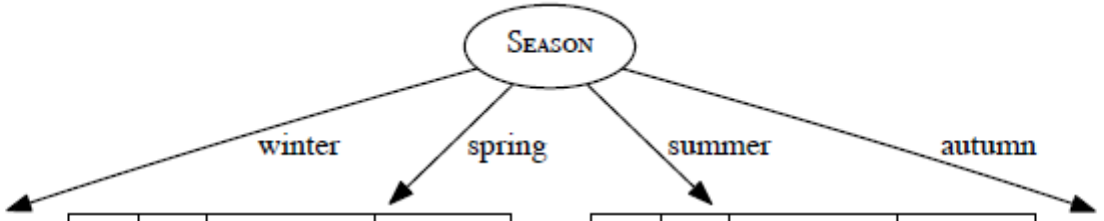
Bike Rentals Data Split

ID	Season	Work Day	Rentals
1	winter	false	800
2	winter	false	826
3	winter	true	900
4	spring	false	2 100
5	spring	true	4 740
6	spring	true	4 900

ID	Season	Work Day	Rentals
7	summer	false	3 000
8	summer	true	5 800
9	summer	true	6 200
10	autumn	false	2 910
11	autumn	false	2 880
12	autumn	true	2 820

Split by Feature	Level	Part.	Instances	$\frac{\lvert\mathcal{D}_{d=l}\rvert}{\lvert\mathcal{D}\rvert}$	$var\,(t,\mathcal{D})$	Weighted Variance
Season	*winter*	$\mathcal{D}_1$	$\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$	0.25	2 692	$1\,379\,331\frac{1}{3}$
	spring	$\mathcal{D}_2$	$\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$	0.25	$2\,472\,533\frac{1}{3}$	
	summer	$\mathcal{D}_3$	$\mathbf{d}_7, \mathbf{d}_8, \mathbf{d}_9$	0.25	3 040 000	
	autumn	$\mathcal{D}_4$	$\mathbf{d}_{10}, \mathbf{d}_{11}, \mathbf{d}_{12}$	0.25	2 100	
Work Day	*true*	$\mathcal{D}_5$	$\mathbf{d}_3, \mathbf{d}_5, \mathbf{d}_6, \mathbf{d}_8, \mathbf{d}_9, \mathbf{d}_{12}$	0.50	$4\,026\,346\frac{1}{3}$	$2\,551\,813\frac{1}{3}$
	false	$\mathcal{D}_6$	$\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_4, \mathbf{d}_7, \mathbf{d}_{10}, \mathbf{d}_{11}$	0.50	1 077 280	

Bike Rentals Data Split



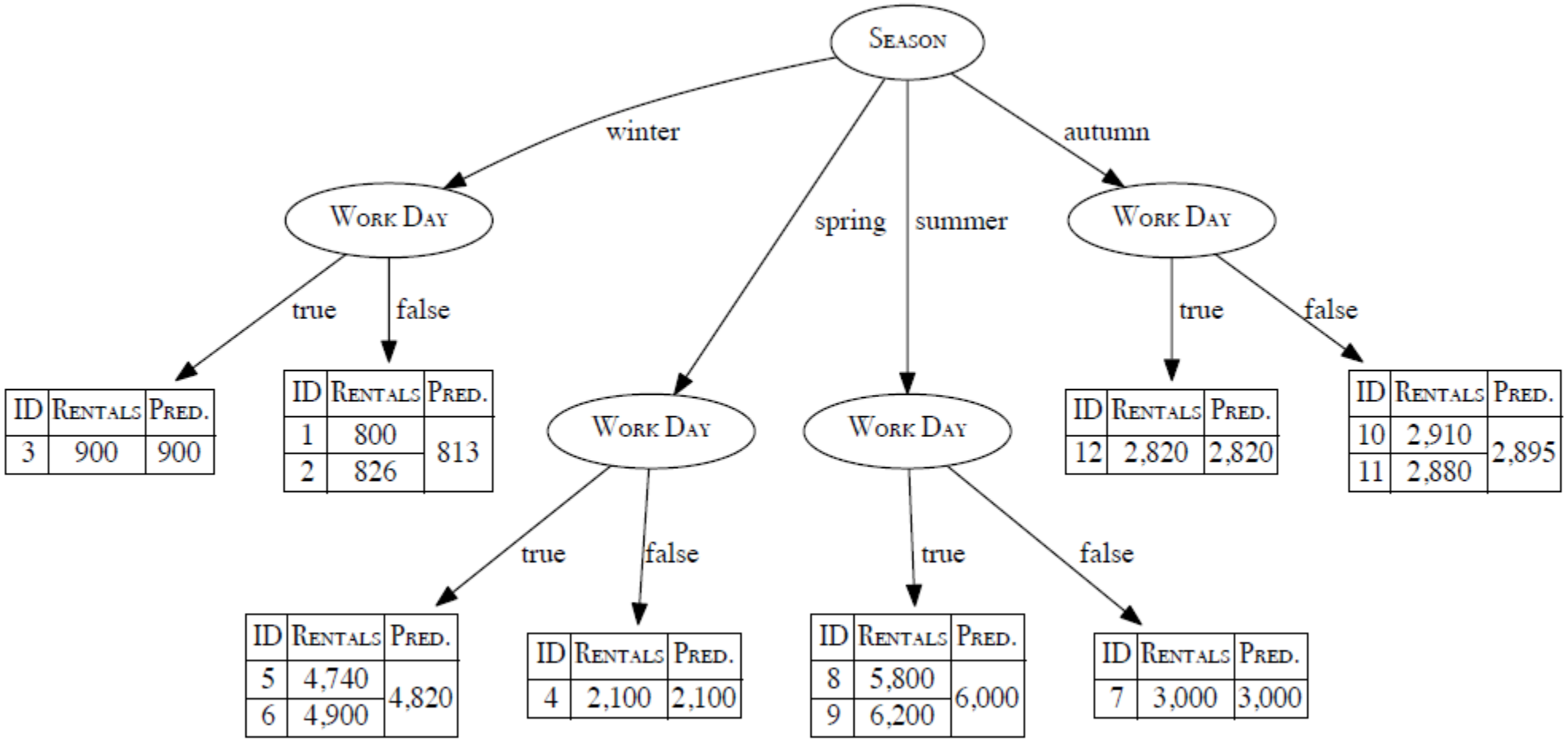
	ID	Work Day	Rentals
D1	1	false	800
	2	false	826
	3	true	900

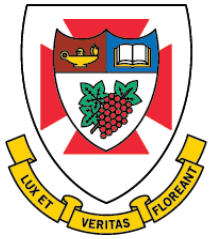
	ID	Work Day	Rentals
D2	4	false	2,100
	5	true	4,740
	6	true	4,900

	ID	Work Day	Rentals
D3	7	false	3,000
	8	true	5,800
	9	true	6,200

	ID	Work Day	Rentals
D4	10	false	2,910
	11	false	2,880
	12	true	2,820

Bike Rentals Regression Tree





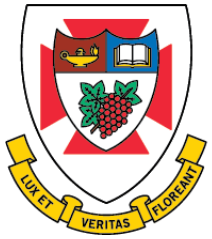
THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

Noisy Data, Overfitting and Tree Pruning

Overfitting

- In the case of a decision tree, over-fitting involves splitting the data on an irrelevant feature.

The likelihood of over-fitting occurring increases as a tree gets deeper because the resulting classifications are based on smaller and smaller subsets as the dataset is partitioned after each feature test in the path.



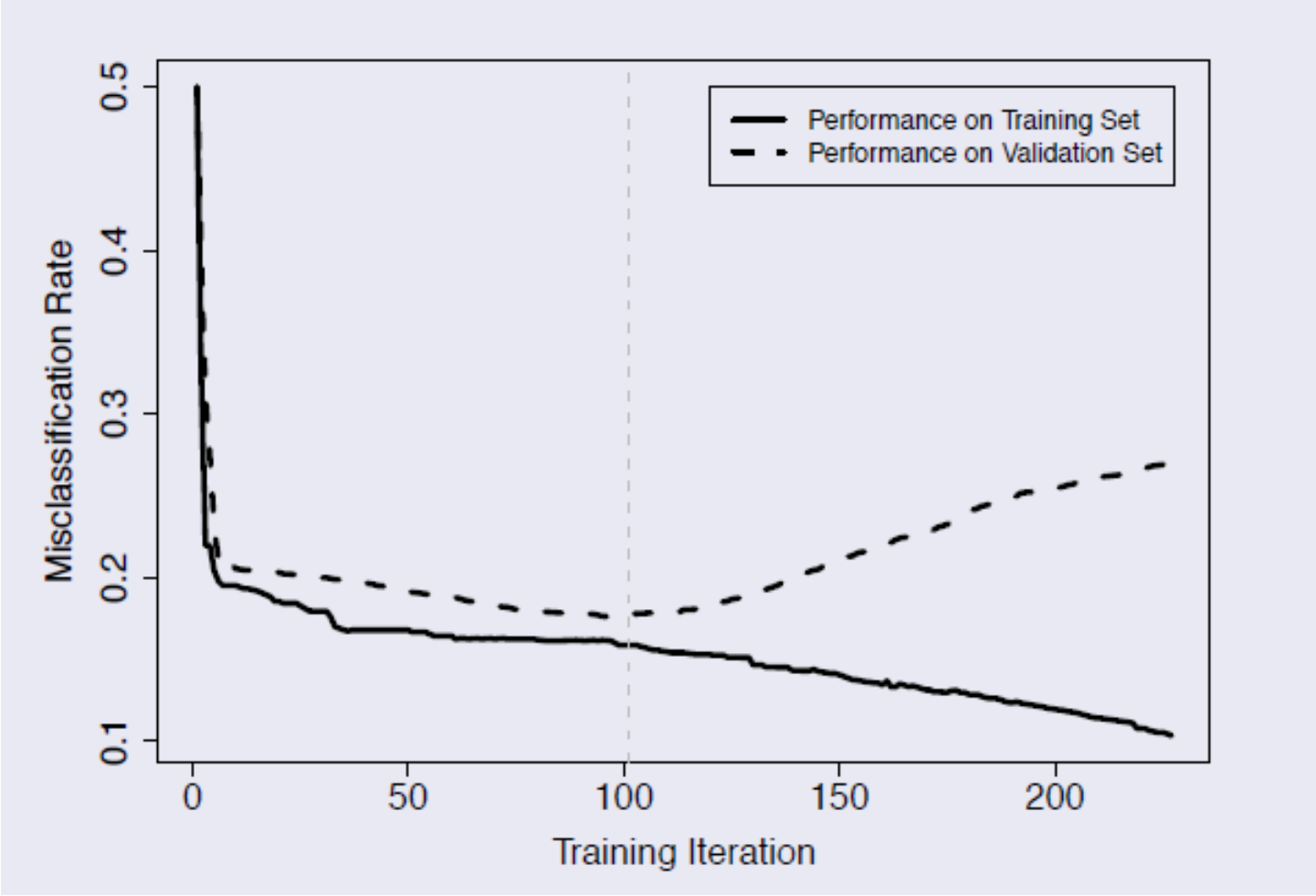
THE UNIVERSITY OF WINNIPEG | Professional, Applied and Continuing Education

How Do We Guard Against Over-Fitting?

Tree-Pruning

- **Pre-pruning**
  - Early stopping
  - X^2 pruning
- **Post-pruning**
  - allow the algorithm to grow the tree as much as it likes and then prune the tree branches that cause over-fitting
  - using the validation set evaluate the prediction accuracy achieved by both the fully grown tree and the pruned copy of the tree. If the pruned copy of the tree performs no worse than the fully grown tree the node is a candidate for pruning.

Pre-Pruning: Early Stopping Approach



Post-Pruning Approach

Common Post-pruning Approach

- Using the validation set evaluate the prediction accuracy achieved by both the fully grown tree and the pruned copy of the tree. If the pruned copy of the tree performs no worse than the fully grown tree the node is a candidate for pruning.

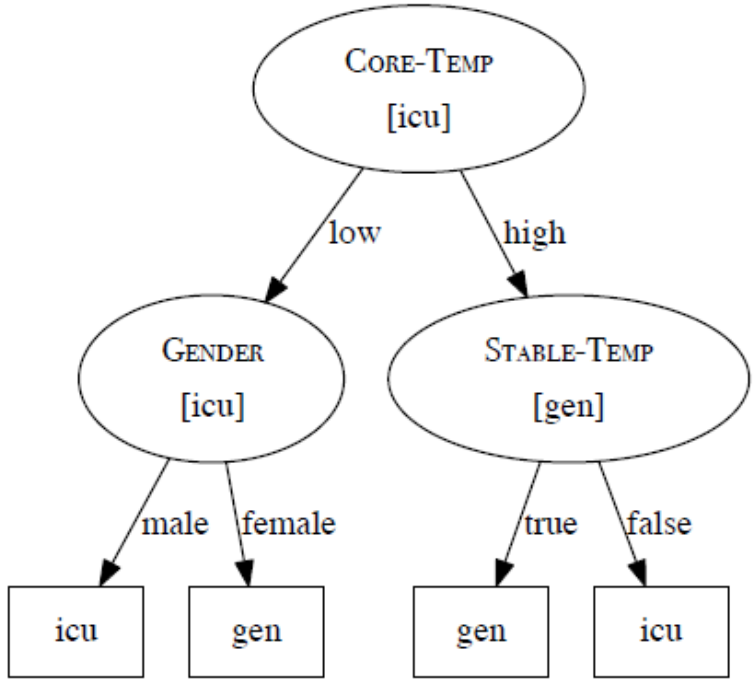
Tree Pruning

Table: An example validation set for the post-operative patient routing task.

ID	CORE-TEMP	STABLE-TEMP	GENDER	DECISION
1	high	true	male	gen
2	low	true	female	icu
3	high	false	female	icu
4	high	false	male	icu
5	low	false	female	icu
6	low	true	male	icu

Hypothermia is a major concern for post-operative patients. Hence, most descriptive features relevant to this domain relate to patient's body temperature,

Tree Pruning



The decision tree for the post-operative patient routing task.

Advantages of Tree Pruning

- Smaller trees are easier to interpret
- Increased generalization accuracy when there is noise in the training data (noise dampening).

Tree Pruning

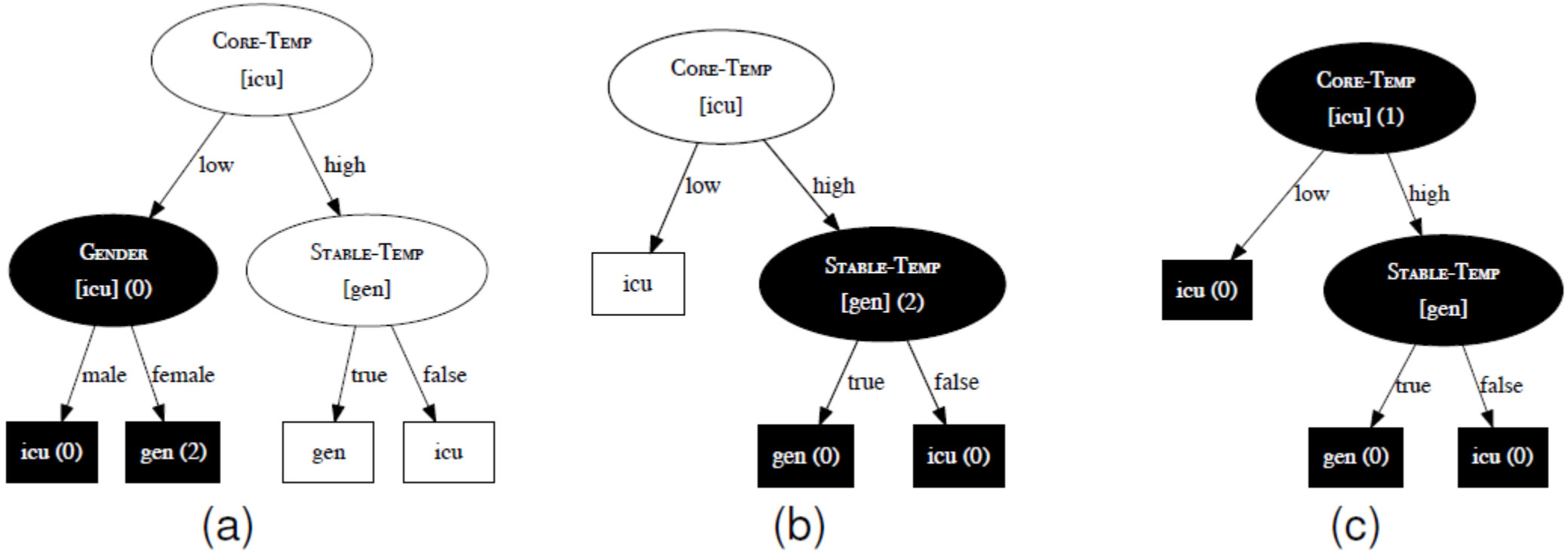


Figure: The iterations of reduced error pruning for the decision tree using the validation dataset. The subtree that is being considered for pruning in each iteration is highlighted in black. The prediction returned by each non-leaf node is listed in square brackets. The error rate for each node is given in round brackets.

Summary

- The ID3 algorithm works the same way for larger more complicated datasets.
- There have been many extensions and variations proposed for the ID3 algorithm:
  - using different impurity measures (Gini, Gain Ratio)
  - handling continuous descriptive features
  - to handle continuous targets
  - pruning to guard against over-fitting
  - using decision trees as part of an ensemble (Random Forests)