e-PG Pathshala

Subject : Computer Science

Paper: Machine Learning

Module: Decision Tree Algorithm ID3

Module No: CS/ML/14

Quadrant I – e-text

Welcome to the e-PG Pathshala Lecture Series on Machine Learning. In this module we will be discussing the ID3 heuristic for choosing the attributes of a Decision Tree.

Learning Objectives:

The learning objectives of this module are as follows:

- To explain greedy algorithm for Decision tree induction
- To outline the ID3 heuristic for choosing attributes
- To explain the concepts of entropy, impurity and information gain
- To illustrate with an example the building of a Decision tree using ID3

14.1 Introduction

The decision tree can be defined as a tree with decision nodes which partitions the examples into 2 subsets based on the value of the attribute representing the decision node. The leaves of this indicates classification of an example. The class of a new sample can be determined by starting at the root and choosing alternatives of the decision node according to the values of the attributes until a leaf node indicating the value of the target variable is reached.

14.2 Decision Tree Algorithms

The basic idea behind any decision tree algorithm is choosing the *best* attribute(s) to split the remaining instances and make that attribute a decision node. We repeat this process recursively for each child. The stopping criterion is generally one of the following, either all the instances have the same target attribute value, or there are no more attributes or there are no more instances to handle.

14.2.1 Basic Algorithm - Decision Tree Induction

The basic algorithm for construction of a decision tree is greedy in nature. In this method the decision tree is constructed in a top-down recursive divide-and-

conquer manner. At start, all the training examples are at the root. Attributes are categorical and the examples are partitioned recursively based on selected attributes. The test attributes are selected on the basis of a heuristic or statistical measure (e.g., **information gain**). Let us understand this algorithm using the example training set given in Table 14.1. The decision attributes age, income, whether student or not, credit rating are used to classify people based on whether they would buy a computer or not. Figure 14.1 shows one sample decision tree for the table. Remember that we can construct more than one decision tree for the table based on the order in which the decision attributes are chosen. In figure 14.1, the first decision attribute chosen is age.

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

Table 14.1 Decision Tree Induction: Training Dataset

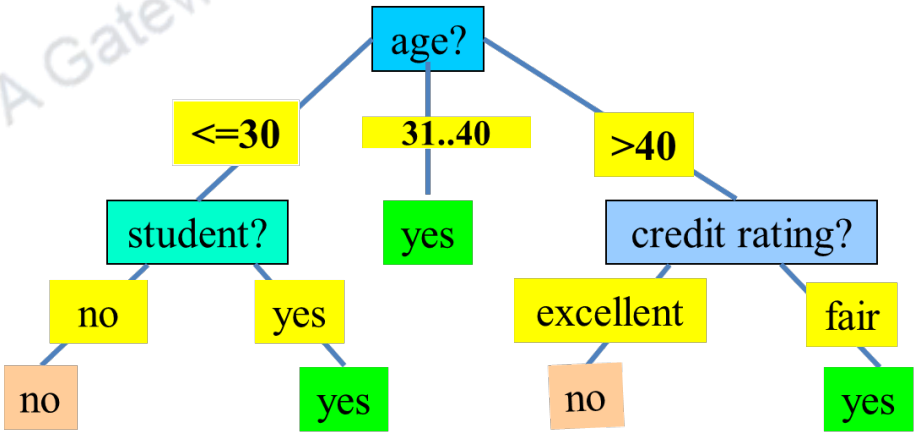


Figure 14.1 Decision Tree for "*buys_computer*"

14.3 Criterion for Attribute Selection

Now the question is how to choose the best attribute. The best choice of attribute will result in the smallest decision tree. Now the basic heuristic is to choose the attribute that produces the "purest". We will discuss what purity is later on. One popular *purity criterion* is ***information gain.*** As the average purity of the partitions obtained based on the chosen attribute increases, the information gain increases. Therefore the strategy to be adopted is to choose attribute that results in greatest information gain. Now we need a good measure of purity – we need a way of knowing when the purity is maximal and when the purity is minimal.

14.3.1 Choosing Attributes

The basic methodology of creating a decision tree is the same for most of the decision tree algorithms. The crucial difference lies in how we select the attributes for the tree, or the order in which we select the attributes for the construction of the decision tree.

We will first focus on the ID3 algorithm developed by Ross Quinlan (Figure 14.2) in 1975. **ID3** (**Iterative Dichotomiser 3**) is an algorithm invented by Ross Quinlan. The algorithm is used to generate a decision tree from a dataset using Shannon Entropy.

Figure 14.2 Ross Quinlan

14.3.2 Identifying the Best Attributes

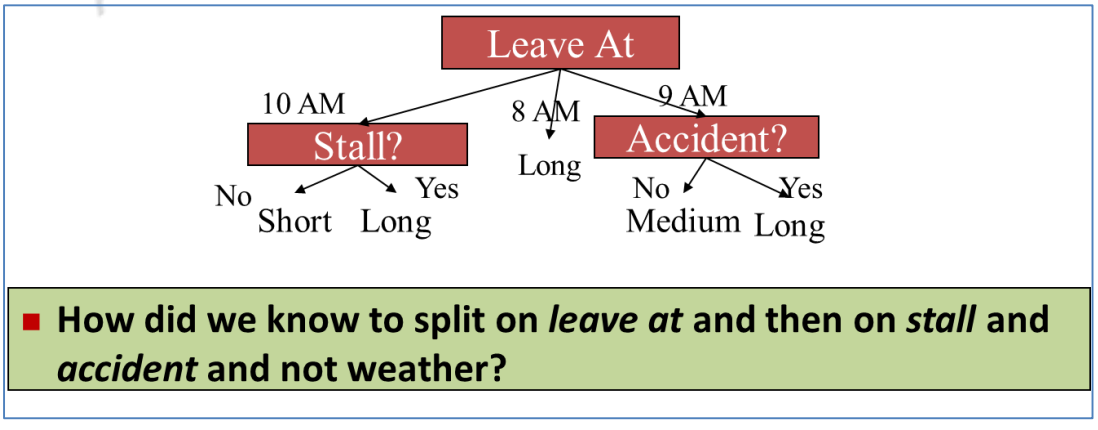


Figure 14.3 Example of Decision Tree

In the decision tree given in Figure 14.3, we need to classify the length of the travel as short, medium or long based on when we leave, whether there is an accident on the way and whether the traffic was stalled. How did we know that we first need to split with ***Leave At*** attribute and then go on to split with ***Stall*** on left side and ***Accident*** on the right? This is the question we need to answer.

14.3.3 Construction of Decision Trees

The algorithm works in a top down recursive divide-and-conquer fashion where the first attribute is selected as root node and branch is created for each possible attribute value. Then the instances are split into disjoint subsets (one for each branch extending from the decision node). This partitioning of the subsets of the instances is repeated recursively for each branch of the decision node. The recursive process stops when all instances that belong to a subset have the same class.

14.3.4 ID3 Heuristic

ID3 employs top-down induction of decision tree. Attribute selection is the fundamental step to construct a decision tree. ID3 employs a top-down greedy search through the space of possible decision trees. The algorithm is called greedy because the highest values are always picked first and there is no backtracking. The idea is to select the attribute that is at that point most useful for classifying examples.

To determine the best attribute, we look at the ID3 heuristic. ID3 splits attributes based on their ***entropy.*** Entropy is the measure of disinformation. It comes from information theory. Higher the entropy, higher is the information content. In other words we select the attribute that has the highest information gain. We will go into the details of entropy and information gain a little later.

14.3.4 Steps of the ID3 Algorithm

1. The first step of the algorithm is the selection of the attributes that will become nodes of the decision tree. As already discussed there are two terms entropy and information gain that are used as the basis for attribute selection.

2. Once the attribute is selected for the current node, the child nodes are generated, one for each possible value of the selected attribute.

3. The examples are then partitioned using the possible values of this attribute and these partitioned subsets of the examples are assigned to the appropriate child node

4. The steps 1 to 3 are repeated for each child node until all examples associated with a node are either all positive or all negative

14.3.5 ID3 in Gaming

Black & White, a game developed by Lionhead Studios, released in 2001 used ID3. The algorithm was used to predict a player's reaction to a certain creature's action. In this model, a greater feedback value means the creature should attack.

14.4 Information Gain

As we have discussed we need to determine which attribute best classifies the data. Figure 14.4 shows an example of classifying whether a particular person is likely to pay the insurance or not. The figure shows splits based on two attributes, ***Balance*** and whether the applicant is ***Employed***. The split over ***Balance*** ≤ 50K and >50K shows that the examples are mixed and not clearly demarcated. However the split over ***Employed*** shows that the split of examples is not mixed. This shows that the test based on ***Employed*** is more informative***.***

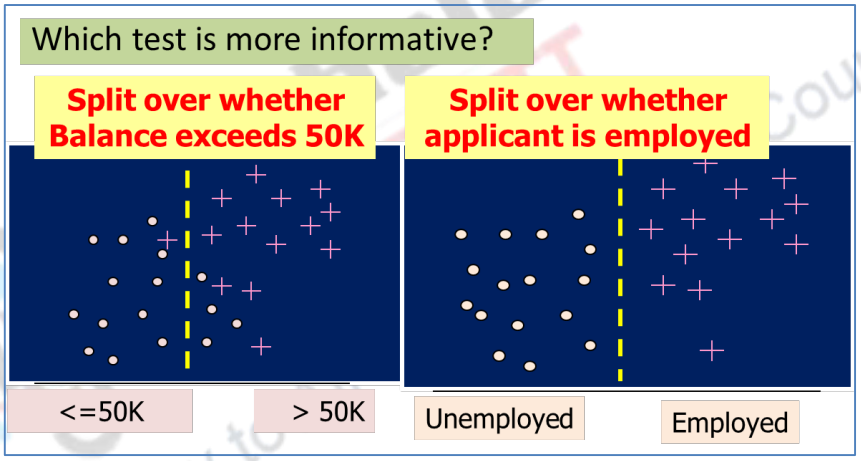


Figure 14.4 Example of Split based on Attributes

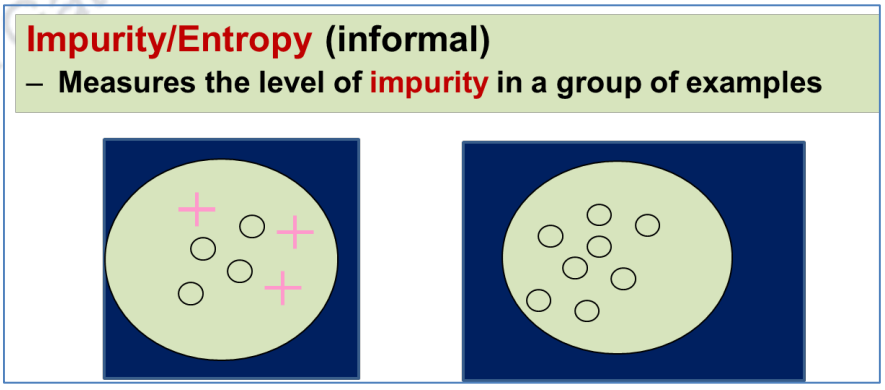


Figure 14.5 (a) Examples Levels of Impurity

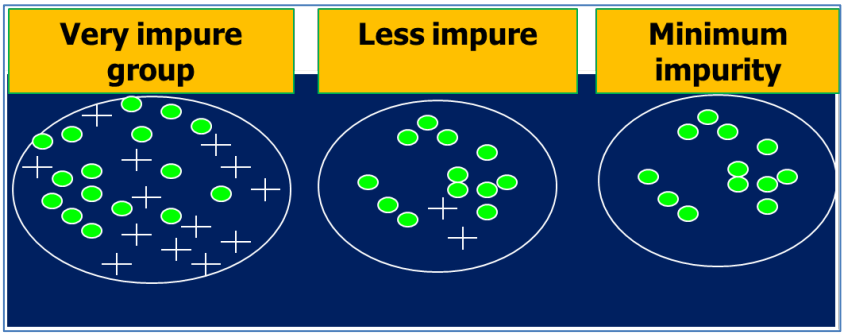


Figure 14.5 (b) Examples Levels of Impurity

Figure 14.5 (a) shows partitions of samples, one which have both types of samples and is therefore impure, while the other has only one type of samples and is therefore pure. Figure 14.5 (b) similarly shows partitions that are very impure, less impure and one with minimum impurity.

Information gain is the statistical quantity measuring how well an attribute classifies the data. In other words in order to select the best attribute we need to first calculate the information gain for each attribute and then choose the attribute with the greatest information gain.

14.4.1 Entropy

One common way to measure impurity was given by Claude Shannon. He defined a mathematical function, ***Entropy***, which measures information content of a *random process*. Entropy has the largest value when events are equiprobable and the smallest value when only one event has non-zero probability. Entropy comes from information theory. The higher the value of entropy the more is the information content. Entropy can be defined as given below:

Entropy = $\sum_{i} -p_i \log_2 p_i$

where p_i is the probability of class I and is computed as the proportion of class i

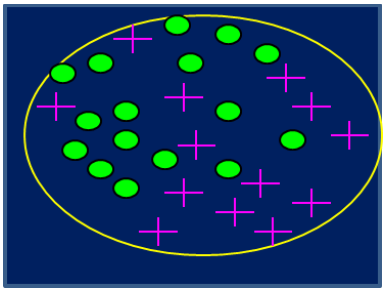


Fig 14.6 Example Set

in the set. Now let us understand entropy and learning from examples. Entropy is minimized when all values of the target attribute are the same. For example if

we know that an attribute ***commute time*** will always be *short*, then it's entropy is zero. Entropy is maximized when there is an equal chance of all values for the target attribute (i.e. the result is random). For example if commute time is short in 3 instances, medium in 3 instances and long in 3 instances, entropy is maximized.

In other words Entropy **H(S)** is a measure of the amount of uncertainty in the (data) set S. Essentially Entropy measures the impurity of an arbitrary collection S of examples. Therefore, more the entropy, measures information content of random process (Figure 14.7).

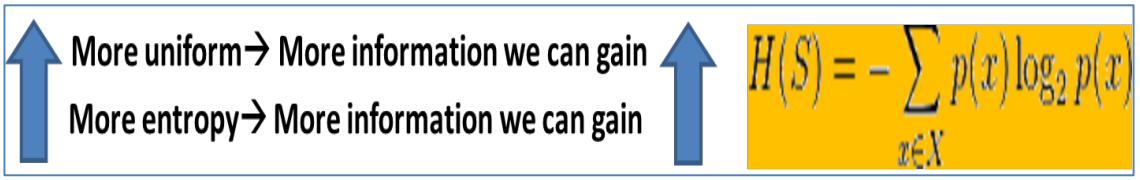


Figure 14.7 Entropy

For a collection S having positive and negative examples (Figure 14.8) , the entropy is given as:

Entropy(S) = $-p_+ \log_2 p_+ - p_- \log_2 p_-$

- where p_+ is the proportion of positive examples &
- p_- is the proportion of negative examples

Entropy(S) = 0 if all members of S belong to the same class and Entropy(S) = 1 (maximum) when all members are split equally.

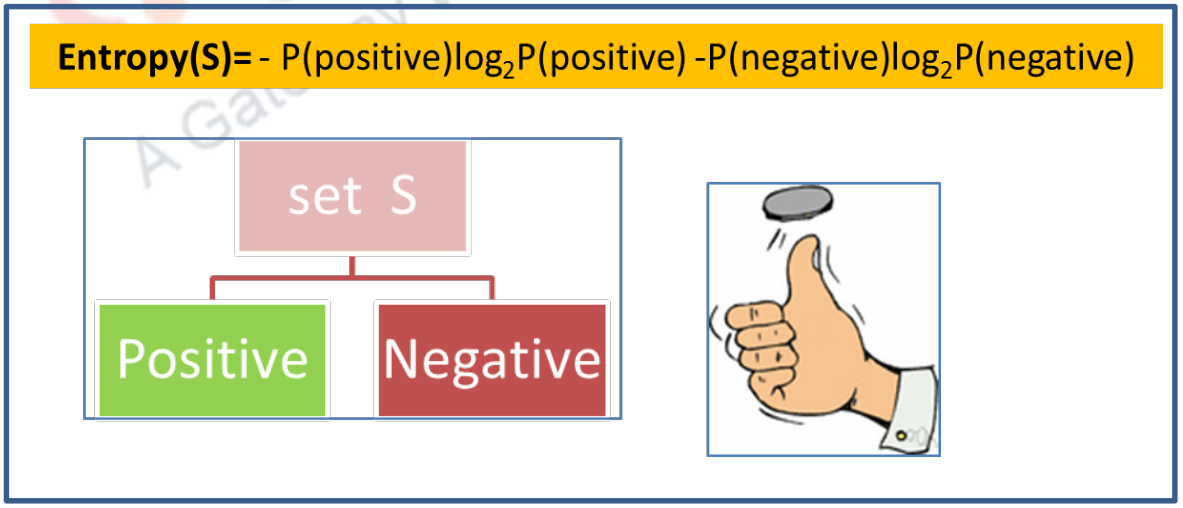


Figure 14.8 Entropy of Set having Positive and Negative Samples

14.4.2 Two-Class Cases

Let us consider the example given in Figure 14.9. What is the entropy of a group in which all examples belong to the same class?

- **entropy = - 1 $\log_2 1 = 0$**

However such a group is not a good training set for learning.

What is the entropy of a group with 50% in either class?

- **entropy = -0.5 $\log_2 0.5$ – 0.5 $\log_2 0.5$ =1**

This group is good training set for learning.

14.4.3 Entropy and Information Gain

As we have discussed before our objective is to determine which attribute in a given set of training feature vectors is most useful for discriminating between the classes to be learned. Information gain tells us how important a given attribute of the feature vector is. We will use this information to decide the ordering of attributes in the nodes of a decision tree. Now information gain of an attribute can be described in terms on entropy and determines the information gained by partitioning the original data set into subsets. With this basis information gain is defined as the measure of the difference in entropy from before to after the set is split on an attribute.

$$IG(A) = H(S) - \sum_{t \in T} p(t) H(t)$$

Here H(S) is the entropy of set S before splitting. T contains the subsets created after splitting S by attribute A such that S = is split into subsets t ε T. Here *p(t)* is the proportion of number of elements in t to the number of elements in the whole set S . H(t) is the entropy of each subset t.

We can calculate information gain as a measure in the expected reduction in entropy. The higher the information gain, more is the expected reduction in entropy.

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

here values(A) is the set of all possible values for attribute A, S_v is the subset of S for which attribute A has value v.

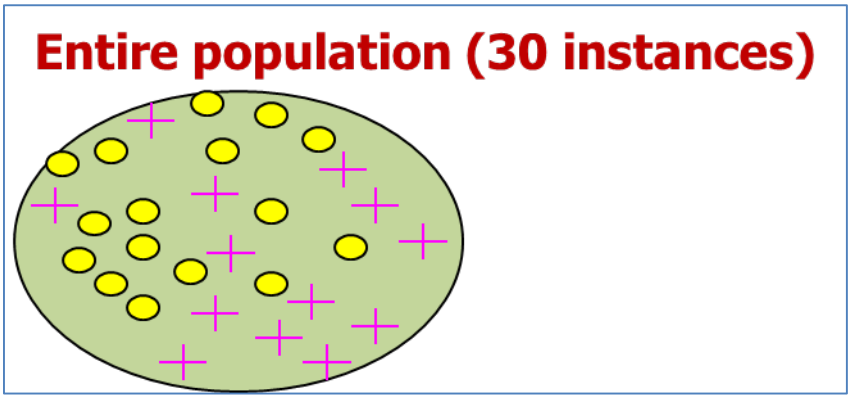


Figure 14.9 Example for Entropy Calculation

Figure 14.9 shows an example with 30 instances or samples of which 14 belongs to one class and the rest (16) belong to the second class. The overall information gain is the entropy(parent)-average entropy of it's children. Parent entropy is given below:

$$-\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right)=0.996$$

14.5 Illustrative Example for ID3 Algorithm

Let us consider the sample training data given in Table 14.2 to determine whether an animal lays eggs. The training data set consists of 6 samples having four attributes namely Warm-blooded, Feathers, Fur and Swims and we need to find out whether an animal lays eggs.

Independent/Condition attributes					Dependent /Decision attributes
Animal	Warm-blooded	Feathers	Fur	Swims	Lays Eggs
Ostrich	Yes	Yes	No	No	Yes
Crocodile	No	No	No	Yes	Yes
Raven	Yes	Yes	No	No	Yes
Albatross	Yes	Yes	No	No	Yes
Dolphin	Yes	No	No	Yes	No
Koala	Yes	No	Yes	No	No

Table 14.2 Sample Training Data - Whether an Animal Lays Eggs

$$Entropy(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Entropy(4Y,2N)=
$(4/6)\log_2(4/6) - (2/6)\log_2(2/6)$
$= 0.91829$

Figure 14.10 Entropy of Set S

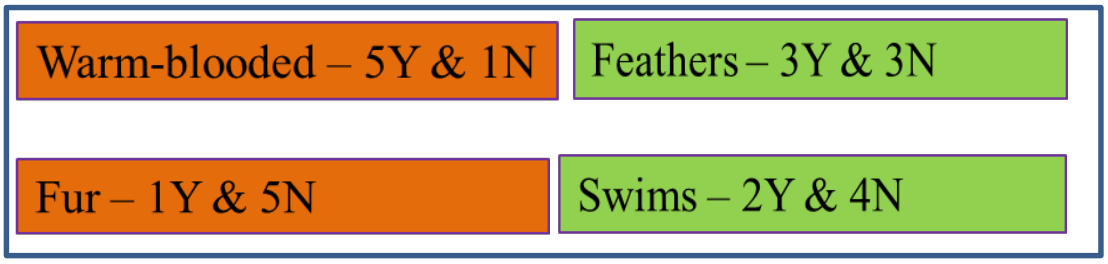


Figure 14.11 Analysis of the Decision Attributes

The first step in the algorithm is to examine the decision values in the sample set S and calculate the Entropy of S. We see that of the 6 samples, 4 samples are for Yes and 2 for No. Hence the Entropy(S) is calculated as given in Figure 14.10. Now we need to first find the decision values associated with each of the four attributes (Figure 14.11) in order to find the entropy corresponding to the subset for that attribute. From this we can find the Gain obtained by using the attribute for each decision value as given below in Figure 14.12:

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Figure 14.12 Gain(S,A)

Now, we have to find the Information Gain (IG) for all four attributes Warm-blooded, Feathers, Fur & Swims.

For attribute 'Warm-blooded':
Values(Warm-blooded) : [Yes,No] S = [4Y,2N]
S_{Yes} = [3Y,2N] $E(S_{Yes}) = 0.97095$
S_{No} = [1Y,0N] $E(S_{No}) = 0$ (all members belong to same class)
Gain(S,Warm-blooded) = 0.91829 –
[(5/6)*0.97095 + (1/6)*0]= 0.10916

Obtained for Lays Eggs -E(S)

Warm blooded	Lays Eggs
Yes	Yes
No	Yes
Yes	Yes
Yes	Yes
Yes	No
Yes	No

For attribute 'Feathers':
Values(Feathers) : [Yes,No] S = [4Y,2N]
S_{Yes} = [3Y,0N] $E(S_{Yes}) = 0$
S_{No} = [1Y,2N] $E(S_{No}) = 0.91829$
Gain(S,Feathers) = 0.91829 – [(3/6)*0 + (3/6)*0.91829]
= 0.45914

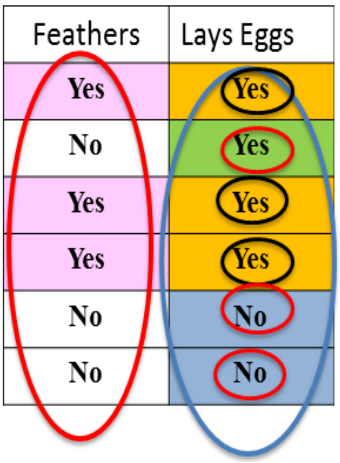
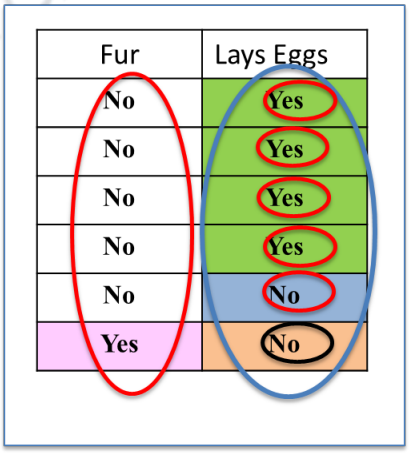


Figure 14.13 Gain for attributes Warm-blooded and Feathers

Figure 14.13 shows the calculation of Gain(S,Warm-Blooded) and Gain(S,Feathers). From the example we know that the set S has 4 Y and 2 N. Considering the 5 Y values of warm-blooded, 3 are Yes for Lays Eggs and 2N. Therefore Entropy(S_{Yes}) can be calculated. Similarly we can calculate Entropy(S_{No}). From these calculations and Entropy(S) (already found) we can find Gain(S,Warm-Blooded) = 0.10916 using Equation given in Figure 14.12. Similarly we can find Gain(S,Feathers) = 0.45914. Similarly as shown in Figure 14.4 we can find Gain(S,Fur) = 0.3167 and Gain(S,Swims) = 0.04411

For attribute 'Fur':
Values(Fur) : [Yes,No] S = [4Y,2N]
S_{Yes} = [0Y,1N] $E(S_{Yes}) = 0$
S_{No} = [4Y,1N] $E(S_{No}) = 0.7219$
Gain(S,Fur) = 0.91829 – [(1/6)*0 + (5/6)*0.7219]= 0.3167



For attribute 'Swims':

Values(Swims): [Yes,No] S = [4Y,2N]

S_{Yes} = [1Y,1N] $E(S_{Yes})$ = 1 (equal members in both classes)

S_{No} = [3Y,1N] $E(S_{No})$ = 0.81127

Gain(S,Swims)=0.91829–(2/6) *1+ [4/6)*0.81127]

=0.04411

Swims	Lays Eggs
No	Yes
Yes	Yes
No	Yes
No	Yes
Yes	No
No	No

Figure 14.14 Gain for attributes Fur and Swims

Gain(S,Warm-blooded) = 0.10916

Gain(S,Feathers) = 0.45914

Gain(S,Fur) = 0.31670

Gain(S,Swims) = 0.0441

Gain(S,Feathers) is maximum, so it is considered as the root node

Figure 14.15 Comparisons of Gains of the Four attributes

By comparing the Gains of the four attributes we see that Gain(S,Feathers) is the maximum and this is chosen as the root node of the decision tree.

Animal	Feathers	Lays Eggs
Ostrich	Yes	Yes
Crocodile	No	Yes
Raven	Yes	Yes
Albatross	Yes	Yes
Dolphin	No	No
Koala	No	No

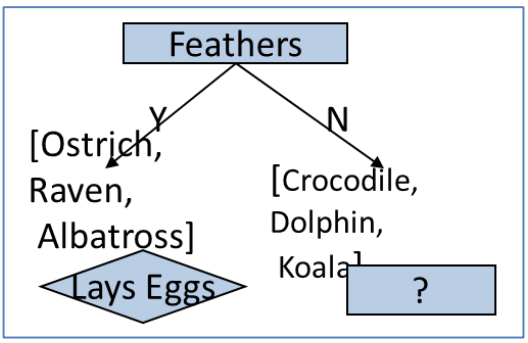


Figure 14.16 Decision Tree with Feathers as Root Node

On studying the part of the table with Feathers we see that knowing that Features as Y we see that the animals Ostrich, Raven and Albatross Lays Eggs without checking any other attribute. However for the case of Feathers (N), we are not able to unambiguously determine whether the animals Lays Eggs. For this we need to consider the reduced table and calculate the corresponding Entropy of new set S (Figure 14.17). Now we calculate the gain of the three attributes Warm-Blooded, Fur and Swims as shown in Figure 14.18.

Animal	Warm-blooded	Feathers	Fur	Swims	Lays Eggs
Crocodile	No	No	No	Yes	Yes
Dolphin	Yes	No	No	Yes	No
Koala	Yes	No	Yes	No	No

We now repeat the procedure,
S: [Crocodile, Dolphin, Koala]
S: [1Y,2N]
Entropy(S) = $-(1/3)\log_2(1/3) - (2/3)\log_2(2/3) = 0.91829$

Figure 14.17 Reduced Decision Tree

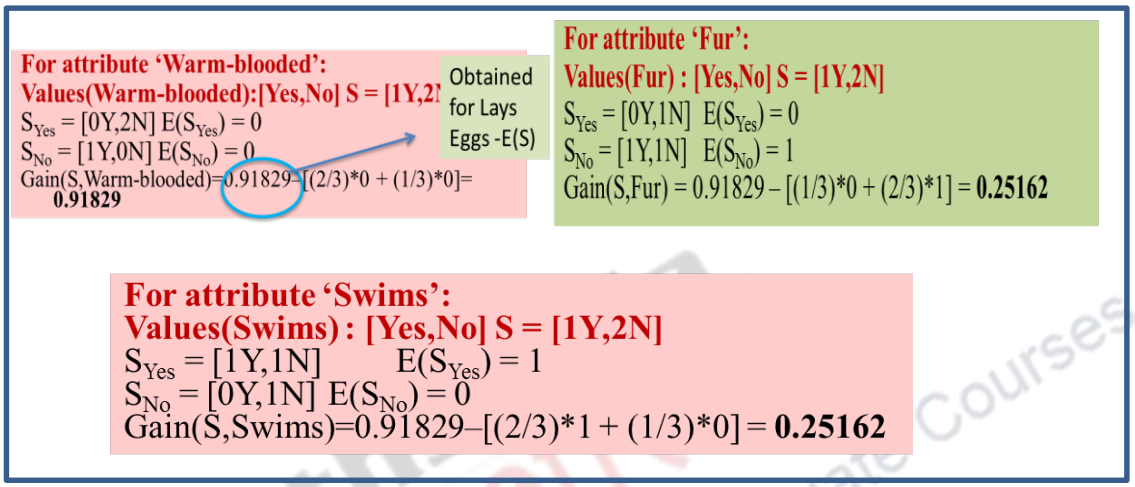


Figure 14.18 Gain for attributes Warm-Blooded, Fur and Swims

Now we find that the Gain(S,Warm-Blooded) is the highest and that is chosen as the next attribute. Now we continue constructing the tree (Figure 14.19). At this stage all the sample are taken care of with single class in each subset. Thus we have obtained the final tree (Figure 14.19)

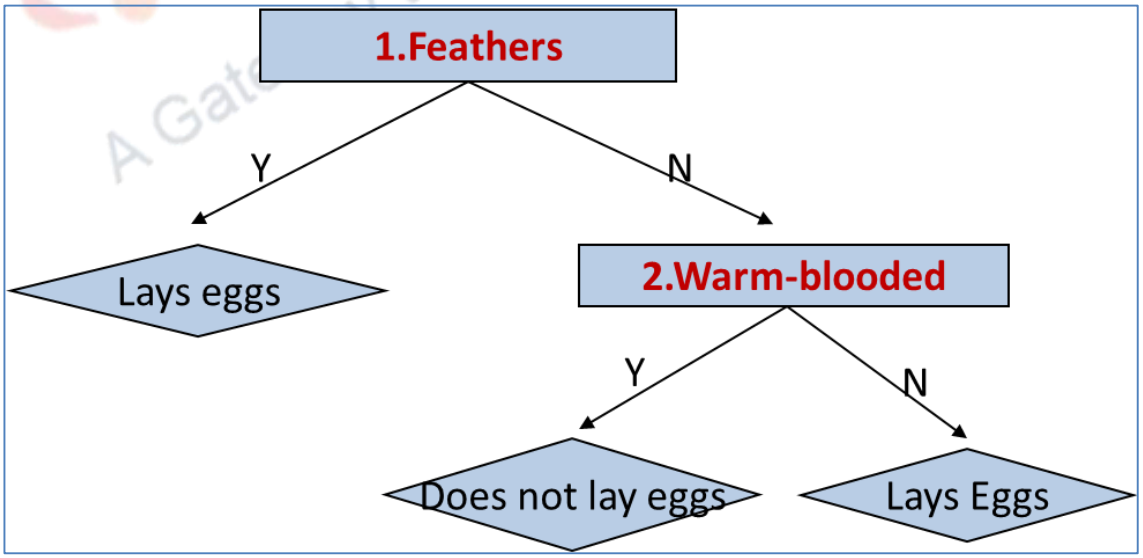


Figure 14.19 Final Decision Tree

14.6 Try Out Example - Factors affecting sunburn

Table 14.3 shows an example that can be tried out by you.

Name	Hair	Height	Weight	Lotion	Sunburned
Sarah	Blonde	Average	Light	No	Yes
Dana	Blonde	Tall	Average	Yes	No
Alex	Brown	Short	Average	Yes	No
Annie	Blonde	Short	Average	No	Yes
Emily	Red	Average	Heavy	No	Yes
Pete	Brown	Tall	Heavy	No	No
John	Brown	Average	Heavy	No	No
Katie	Blonde	Short	Light	Yes	No

Table 14.3 Try out Example

Summary

- Explained the greedy algorithm for Decision tree induction
- Outlined the ID3 heuristic for choosing attributes
- Explained the concepts of entropy, impurity and information gain
- Used an example to illustrate the building of a Decision tree using ID3