Classification

Chapter Objectives

- ✓ To comprehend the concept, types and working of classification
- ✓ To identify the major differences between classification and regression problems
- ✓ To become familiar about the working of classification
- ✓ To introduce the decision tree classification system with concepts of information gain and Gini Index
- ✓ To understand the workings of the Naïve Bayes method

5.1 Introduction to Classification

Nowadays databases are used for making intelligent decisions. Two forms of data analysis namely classification and regression are used for predicting future trends by analyzing existing data. Classification models predict discrete value or class, while Regression models predict a continuous value. For example, a classification model can be built to predict whether India will win a cricket match or not, while regression can be used to predict the runs that will be scored by India in a forthcoming cricket match.

Classification is a classical method which is used by machine learning researchers and statisticians for predicting the outcome of unknown samples. It is used for categorization of objects (or things) into given discrete number of classes. Classification problems can be of two types, either binary or multiclass. In binary classification the target attribute can only have two possible values. For example, a tumor is either cancerous or not, a team will either win or lose, a sentiment of a sentence is either positive or negative and so on. In multiclass classification, the target attribute can have more than two values. For example, a tumor can be of type 1, type 2 or type 3 cancer; the sentiment of a sentence can be happy, sad, angry or of love; news stories can be classified as weather, finance, entertainment or sports news.

Some examples of business situations where the classification technique is applied are:

- To analyze the credit history of bank customers to identify if it would be risky or safe to grant them loans.
- To analyze the purchase history of a shopping mall's customers to predict whether they will buy a certain product or not.

In first example, the system will predict a discrete value representing either risky or safe, while in second example, the system will predict yes or no.

Some more examples to distinguish the concept of regression from classification are:

- To predict how much a given customer will spend during a sale.
- To predict the salary-package of a student that he/she may get during his/her placement.

In these two examples, there is a prediction of continuous numeric value. Therefore, both are regression problems.

In this chapter, we will discuss the basic approaches to perform the classification of data.

5.2 Types of Classification

Classification is defined as two types. These are:

- Posteriori classification
- Priori classification

5.2.1 Posteriori classification

The word '*Posteriori*' means something derived by reasoning from the observed facts. It is a supervised machine learning approach, where the target classes are already known, i.e., training data is already labeled with actual answers.

5.2.2 Priori classification

The word '*Priori*' means something derived by reasoning from self-evident propositions. It is an unsupervised machine learning approach, where the target classes are not given. The question is 'Is it possible to make predictions, if labeled data is not available?'

The answer is yes. If data is not labeled, then we can use clustering (unsupervised technique) to divide unlabeled data into clusters. Then these clusters can be assigned some names or labels and can be further used to apply classification to make predictions based on this dataset. Thus, although data is not labeled, we can still make predictions based on data by first applying clustering followed by classification. This approach is known as Priori classification.

In this chapter, we will learn posteriori classification, a supervised machine learning approach to predict the outcome when training dataset is labeled.

5.3 Input and Output Attributes

In any machine learning approach the input data is very important. Data contains two types of attributes, namely, input attributes and output attributes. The class attribute that represents the output of all other attributes is known as an output attribute or dependent attribute, while all other attributes are known as input attributes or independent attributes. For example, the dataset for iris plant's flowers given in Figure 5.1 has Sepal length, Sepal width, Petal length and Petal width as input attributes and species as an output attribute.

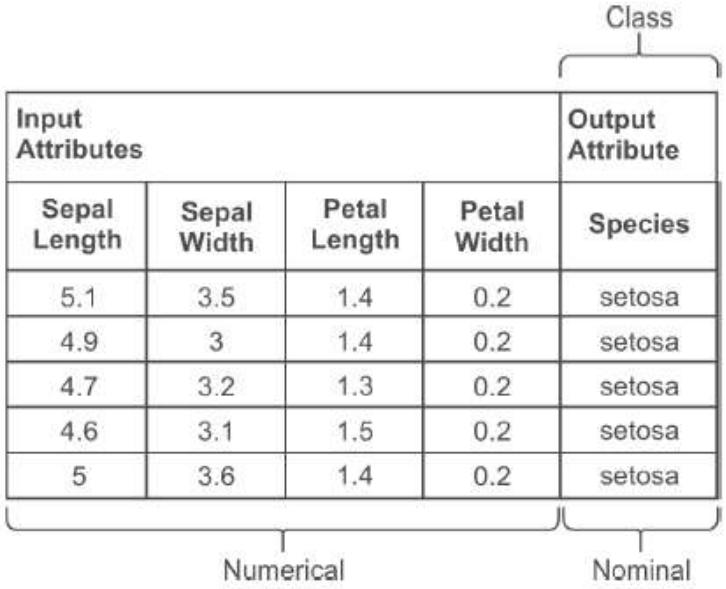

Figure 5.1 Input and output attributes

The attributes can be of different types. The attributes having numbers are called numerical attributes while attributes whose domain is not numerical are known as nominal or categorical attributes. Here, input attributes are of numerical type while species, i.e., the output attribute is of nominal type (as shown in Figure 5.1).

5.4 Working of Classification

Classification is a two-step process. The first step is training the model and the second step is testing the model for accuracy. In the first step, a classifier is built based on the training data obtained by analyzing database tuples and their associated class labels. By analyzing training data, the system learns and creates some rules for prediction. In the second step, these prediction rules are tested on some unknown instances, i.e., test data. In this step, rules are used to make the predictions about the output attribute or class. In this step, the predictive accuracy of the classifier is calculated. The system performs in an iterative manner to improve its accuracy, i.e., if accuracy is not good on test data, the system will reframe its prediction rules until it gets optimized accuracy on test data as shown in Figure 5.2.

The test data is randomly selected from the full dataset. The tuples of the test data are independent of the training tuples. This means that the system has not been exposed to testing data during the training phase. The accuracy of a classifier on a given test data is defined as the percentage of test data tuples that are correctly classified by the classifier. The associated class label of each test tuple is compared with the class prediction made by the classifier for that particular tuple. If the accuracy of the classifier is satisfactory then it can be used to classify future data tuples with unknown class labels.

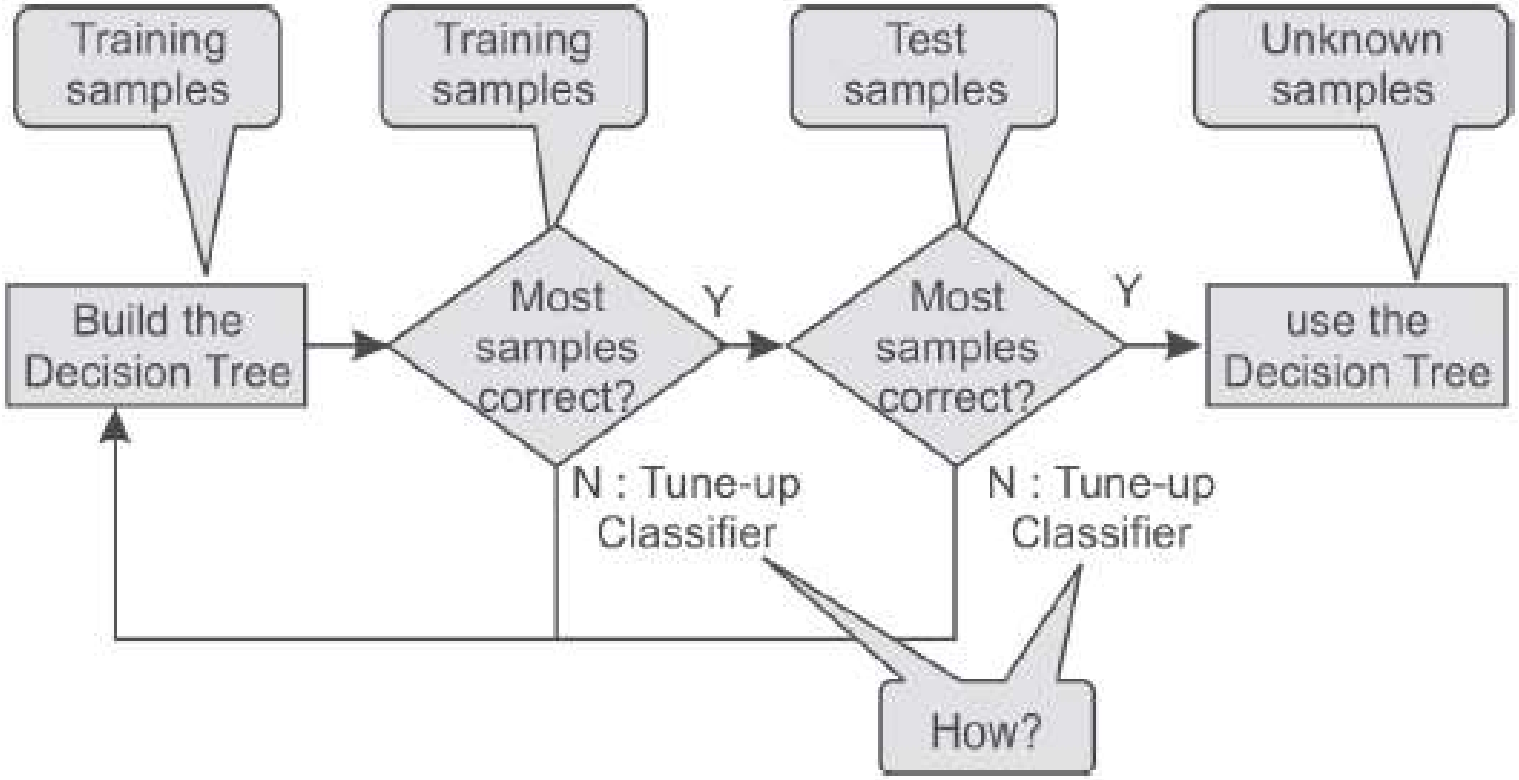

Figure 5.2 Training and testing of the classifier

For example, by analyzing the data of previous loan applications as shown in Figure 5.3, the classification rules obtained can be used to approve or reject the new or future loan applicants as shown in Figure 5.4.

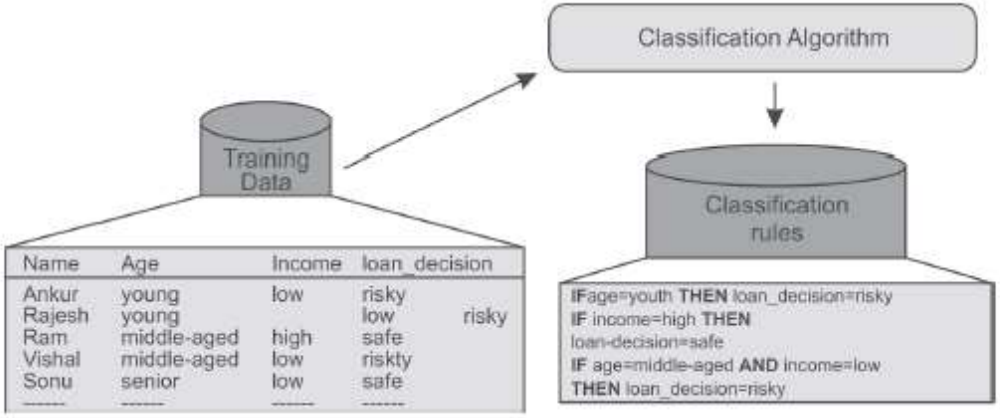

Figure 5.3 Building a classifier to approve or reject loan applications

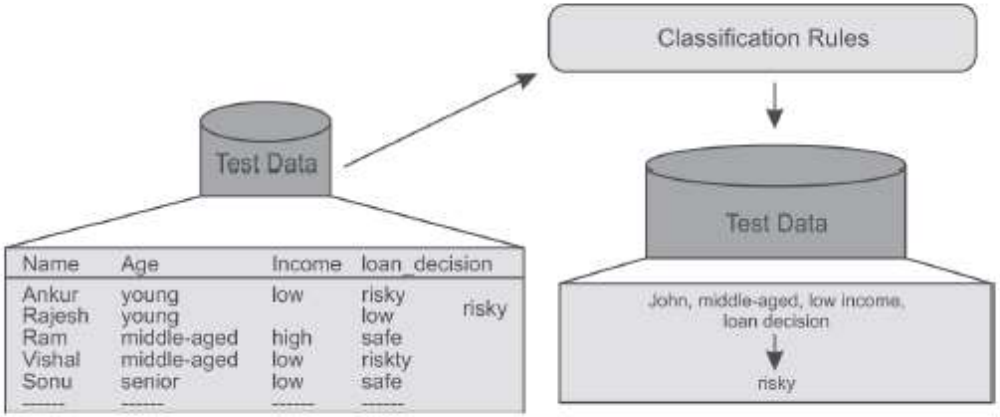

Figure 5.4 Predicting the type of customer based on trained classifier

The same process of training and testing the classifier has also been illustrated in Figure 5.5.

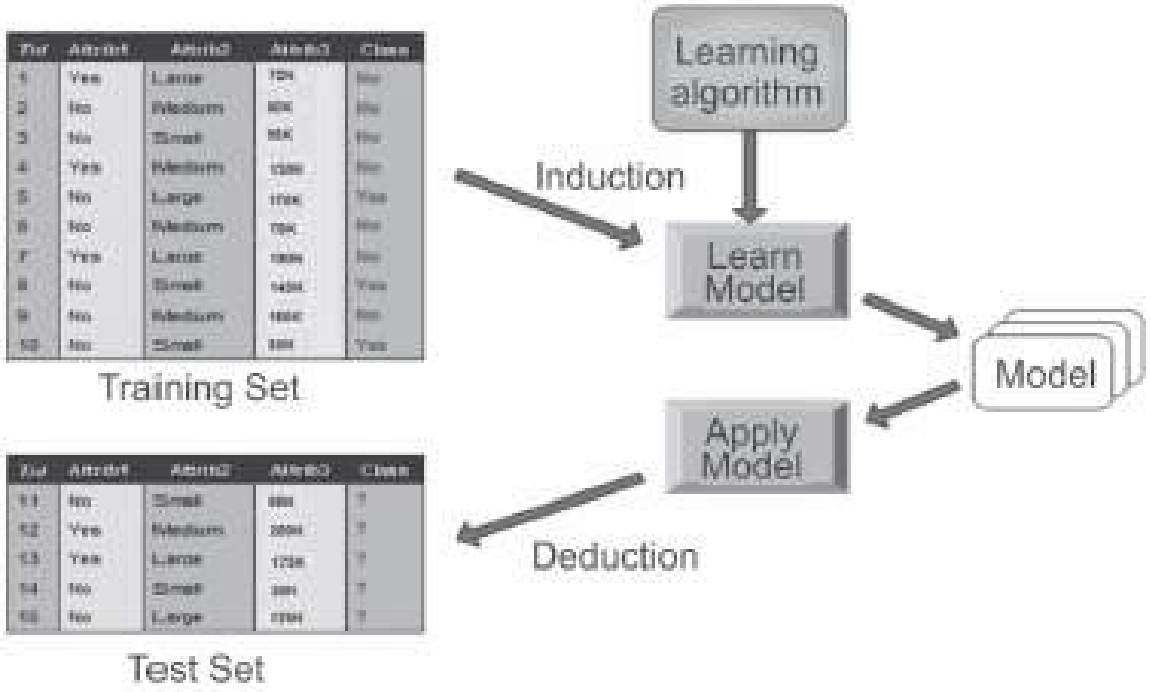

Figure 5.5 Training and testing of the classifier

5.5 Guidelines for Size and Quality of the Training Dataset

There should be a balance between the number of training samples and independent attributes. It has been observed that generally, the number of training samples required is likely to be relatively small if the number of independent or input attributes is small and similarly, number of training samples required is likely to be relatively large if the number of independent or input attributes is large. The quality of the classifier depends upon the quality of the training data. If there are two or more than two classes, then sufficient training data should be available belonging to each of these classes.

Researchers have developed a number of classifiers that include: Decision Tree, Naïve Bayes, Support Vector Machine and Neural Networks. In this chapter, two important classification methods, namely, Decision Tree and Naive Bayes are discussed in detail, as these are widely used by scientists and organizations for classification of data.

5.6 Introduction to the Decision Tree Classifier

In the decision tree classifier, predictions are made by using multiple 'if…then...' conditions which are similar to the control statements in different programming languages, that you might have learnt. The decision tree structure consists of a root node, branches and leaf nodes. Each internal node represents a condition on some input attribute, each branch specifies the outcome of the condition and each leaf node holds a class label. The root node is the topmost node in the tree.

The decision tree shown in Figure 5.6 represents a classifier tasked for predicting whether a customer will buy a laptop or not. Here, each internal node denotes a condition on the input attributes and each leaf node denotes the predicted outcome (class). By traversing the decision tree, one can analyze that if a customer is middle aged then he will probably buy a laptop, if a customer is young and a student then he will probably not buy a laptop. If a customer is a senior citizen and has an excellent credit rating then he can probably buy a laptop. The system makes these predictions with a certain level of probability.

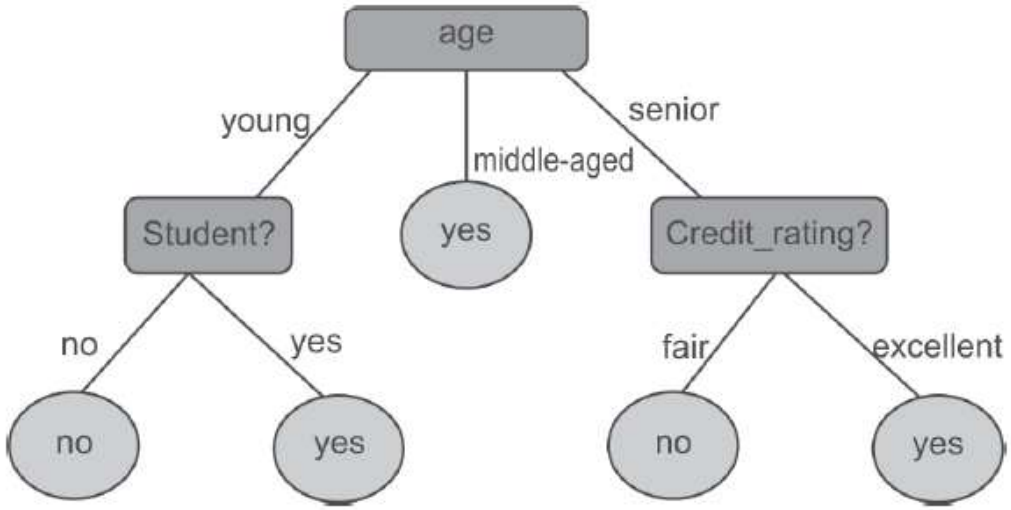

Figure 5.6 Decision tree to predict whether a customer will buy a laptop or not

Decision trees can easily be converted to classification rules in the form of if-then statements. Decision tree based classification is very similar to a '20 questions game'. In this game, one player writes something on a page and other player has to find what was written by asking at most 20 questions; the answers to which can only be yes or no. Here, each node of the decision tree denotes a choice between numbers of alternatives and the choices are binary. Each leaf node specifies a decision or prediction. The training process that produces this tree is known as induction.

5.6.1 Building decision tree

J. Ross Quinlan, a researcher in machine learning, developed a decision tree algorithm known as **ID3 (Iterative Dichotomiser)** during the late 1970s and early 1980s. Quinlan later proposed C4.5 (a successor of ID3), which became a benchmark to which newer supervised learning algorithms are often compared. Decision tree is a common machine learning technique which has been implemented in many machine learning tools like Weka, R, Matlab as well as some programming languages such as Python, Java, *etc.*

These algorithms are based on the concept of Information Gain and Gini Index. So, let us first understand the role of information gain in building the decision tree.

5.6.2 Concept of information theory

Decision tree algorithm works on the basis of information theory. It has been observed that information is directly related with uncertainty. If there is uncertainty then there is information and if there is no uncertainty then there is no information. For example, if a coin is biased having a head on both sides, then the result of tossing it does not give any information but if a coin is unbiased having a head and a tail then the result of the toss provides some information.

Usually the newspaper carries the news that provides maximum information. For example, consider the case of an India-UAE world cup cricket match. It appears certain that India will beat UAE, so this news will not appear on front page as main headlines, but if UAE beats India in a world cup cricket match then this news being very unexpected (uncertain) will appear on the first page as headlines.

Let us consider another example, if in your university or college, there is holiday on Sunday then a notice regarding the same will not carry any information (because it is certain) but if some particular Sunday becomes a working day then it will be information and henceforth becomes a news.

From these examples we can observe that information is related to the probability of occurence of an event. Another important question to consider is, whether the probability of occurrence of an event is more. Then, the information gain will be more frequent or less frequent?

It is certain from above examples that 'more certain' events such as India defeating UAE in cricket or Sunday being a holiday carry very little information. But if UAE beats India or Sunday is working, then even though the probability of these events is lesser than the previous event, it will carry more information. Hence, less probability means more information.

5.6.3 Defining information in terms of probability

Information theory was developed by Claude Shannon. Information theory defines entropy which is average amount of information given by a source of data. Entropy is measured as follows.

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

Therefore, the total information for an event is calculated by the following equation:

$$I = \sum_i \left(-p_i \log p_i\right)$$

In this, information is defined as $-p_i \log p_i$ where p_i is the probability of some event. Since, probability p_i is always less than 1, $\log p_i$ is always negative; thus negating $\log pi$ we get the overall information gain ($-pi \log pi$) as positive.

It is important to remember that the logarithm of any number greater than 1 is always positive and the logarithm of any number smaller than 1 is always negative. Logarithm of 1 is always zero, no matter what the base of logarithm is. In case of log with base 2, following are some examples.

$$\log_2(2) = 1$$

$$\log_2(2^n) = n$$

$$\log_2(1/2) = -1$$

$$\log_2(1/2^n) = -n$$

Let us calculate the information for the event of throwing a coin. It has two possible values, i.e., head (p_1) or tail (p_2). In case of unbiased coin, the probability of head and tail is 0.5 respectively. Thus, the information is

$$I = -0.5 \log(0.5) - 0.5 \log(0.5)$$

$$= -(0.5) * (-1) - (0.5) * (-1) \qquad [\text{As, } \log_2(0.5) = -1]$$

$$= 0.5 + 0.5 = 1$$

The result is 1.0 (using log base 2) and it is the maximum information that we can have for an event with two possible outcomes. This is also known as entropy.

But if the coin is biased and has heads on both the sides, then probability for head is 1 while the probability of tails will be 0. Thus, total information in tossing this coin will be as follows.

$$I = -1 \log(1) - 0 \log(0) = 0 \qquad [\text{As, } \log_2(1) = 0]$$

You can clearly observe that tossing of biased coin carries no information while tossing of unbiased coin carries information of 1.

Suppose, an unbiased dice is thrown which has six possible outcomes with equal probability, then the information is given by:

$$I = 6(-1/6)\log(1/6) = 2.585$$

[the probability of each possible outcome is 1/6 and there are in total six possible outcomes from 1 to 6]

But, if dice is biased such that there is a 50% chance of getting a 6, then the information content of rolling the die would be lower as given below.

$$I = 5(-0.1)\log(0.1) - 0.5\log(0.5) = 2.16$$

[One event has a probability of 0.5 while 5 other events has probability of 0.5, which makes 0.5/5 = 0.1 as the probability of each of remaining 5 events.]

And if the dice is further biased such that there is a 75% chance of getting a 6, then the information content of rolling the die would be further low as given below.

$$I = 5(-0.05)\log(0.05) - 0.75\log(0.75) = 1.39$$

[One event has a probability of 0.75 while 5 other events has probability of 0.25, which makes 0.25/5 = 0.05 as probability of each of remaining 5 events.]

We can observe that as the certainty of an event goes up, the total information goes down.

Information plays a key role in selecting the root node or attribute for building a decision tree. In other words, selection of a *split attribute* plays an important role. Split attribute is an attribute that reduces the uncertainty by largest amount, and is always accredited as a root node. So, the attribute must distribute the objects such that each attribute value results in objects that have as little uncertainty as possible. Ideally, each attribute value should provide us with objects that belong to only one class and therefore have zero information.

5.6.4 Information gain

Information gain specifies the amount of information that is gained by knowing the value of the attribute. It measures the 'goodness' of an input attribute for predicting the target attribute. The attribute with the highest information gain is selected as the next split attribute.

Mathematically, it is defined as the entropy of the distribution before the split minus the entropy of the distribution after split.

Information gain = (Entropy of distribution before the split)
– (Entropy of distribution after the split)

The largest information gain is equivalent to the smallest entropy or minimum information. It means that if the result of an event is certain, i.e., the probability of an event is 1 then information provided by it is zero while the information gain will be the largest, thus it should be selected as a split attribute.

Assume that there are two classes, P and N, and let the set of training data S (with a total number of records s) contain p records of class P and n records of class N. The amount of information is defined as

$$I = -(p/s)\log(p/s) - (n/s)\log(n/s)$$

Obviously if $p = n$, i.e., the probability is equally distributed then I is equal to 1 and if $p = s$ *or* $n = 0$, i.e., training data S contains all the elements of a single class only, then $I = 0$. Therefore if there was an attribute for which all the records had the same value (for example, consider the attribute gender, when all people are male), using this attribute would lead to no information gain that is, no reduction in uncertainty. On the other hand, if an attribute divides the training sample such that all female records belong to Class A, and male records belong to Class B, then uncertainty has been reduced to zero and we have a large information gain.

Thus after computing the information gain for every attribute, the attribute with the highest information gain is selected as split attribute.

5.6.5 Building a decision tree for the example dataset

Let us build decision tree for the dataset given in Figure 5.7.

Instance Number	*X*	*Y*	*Z*	*Class*
1	1	1	1	A
2	1	1	0	A
3	0	0	1	B
4	1	0	0	B

X	*Y*	*Z*	*Class*
1 = 3	1 = 2	1 = 2	A = 2
0 = 1	0 = 2	0 = 2	B = 2

Figure 5.7 Dataset for class C prediction based on given attribute condition

The given dataset has three input attributes X, Y, Z and one output attribute Class. The instance number has been given to show that the dataset contains four records (basically for convenience while making references). The output attribute or class can be either A or B.

There are two instances for each class so the frequencies of these two classes are given as follows:

A = 2 (Instances 1, 2)

B = 2(Instances 3, 4)

The amount of information contained in the whole dataset is calculated as follows:

I = - probability for Class A * log (probability for class A)
- probability for class B * log (probability for class N)

Here, probability for class A = (Number of instances for class A/Total number of instances) = 2/4

And probability for class B = (Number of instances for class B/Total number of instances) = 2/4

Therefore, I = (-2/4) log (2/4) – (2/4) log (2/4) = 1

Let us consider each attribute one by one as a split attribute and calculate the information for each attribute.

Attribute 'X'

As given in the dataset, there are two possible values of X, i.e., 1 or 0. Let us analyze each case one by one.

For X= 1, there are 3 instances namely instance 1, 2 and 4. The first two instances are labeled as class A and the third instance, i.e, record 4 is labeled as class B.

For X = 0, there is only 1 instance, i.e, instance number 3 which is labeled as class B.

Given the above values, let us compute the information given by this attribute. We divide the dataset into two subsets according to X either being 1 or 0. Computing information for each case,

$$I(\text{for } X = 1) = I(X1) = -(2/3)\log(2/3) - (1/3)\log(1/3) = 0.92333$$

$$I(\text{for } X = 0) = I(X0) = -(0/1)\log(0/1) - (1/1)\log(1/1) = 0$$

Total information for above two sub-trees = probability for X having value 1 * I(X1) + probability for X having value 0 * I(X0)

Here, probability for X having value 1 = (Number of instances for X having value 1/Total number of instances) = 3/4

And probability for X having value 0 = (Number of instances for X having value 0/Total number of instances) = 1/4

Therefore, total information for the two sub-trees = (3/4) I(X1) + (1/4) I(X0)

= 0.6925 + 0

= 0.6925

Attribute 'Y'

There are two possible values of Y attribute, i.e., 1 or 0. Let us analyze each case one by one.

There are 2 instances where Y has value 1. In both cases when Y=1 the record belongs to class A and, in the 2 instances when Y = 0 both records belong to class B.

Given the above values, let us compute the information provided by Y attribute. We divide the dataset into two subsets according to Y either being 1 and 0. Computing information for each case,

$$I(\text{For } Y = 1) = I(Y1) = -(2/2)\log(2/2) - (0/2)\log(0/2) = 0$$

$$I(\text{For } Y = 0) = I(Y0) = -(0/2)\log(0/2) - (2/2)\log(2/2) = 0$$

Total information for the two sub-trees = probability for Y having value 1* I(Y1) + probability for Y having value 0 * I(Y0)

Here, probability for Y in 1 = (Number of instances for Y having 1/Total number of instances) = 2/4

And probability for Y in 0 = (Number of instances for Y having 0/Total number of instances) = 2/4

Therefore, the total information for the two sub-trees = (2/4) I(Y1) + (2/4) I(Y0)

$$= 0 + 0$$

$$= 0$$

Attribute 'Z'

There are two possible values of Z attribute, i.e., 1 or 0. Let us analyze each case one by one.

There are 2 instances where Z has value 1 and 2 instances where Z has value 0. In both cases, there exists a record belonging to class A and class B with Z is either 0 or 1.

Given the above values, let us compute the information provided by the Z attribute. We divide the dataset into two subsets according to Z either being 1 or 0. Computing information for each case,

$$I(\text{For } Z = 1) = I(Z1) = -(1/2)\log(1/2) - (1/2)\log(1/2) = 1.0$$

$$I(\text{For } Z = 0) = I(Z0) = -(1/2)\log(1/2) - (1/2)\log(1/2) = 1.0$$

Total information for the two sub-trees = probability for Z having value 1 * I(Z1) + probability for Z having value 0 * I(Z0)

Here, probability for Z having value 1 = (Number of instances for Z having value 1/Total number of instances) = 2/4

And probability for Z having value 0 = (Number of instances for Z having value 0/Total number of instances) = 2/4

Therefore, total information for two sub-trees = (2/4) I(Z1) + (2/4) I(Z0)

$$= 0.5 + 0.5$$

$$= 1.0$$

The Information gain can now be computed:

Potential Split attribute	*Information before split*	*Information after split*	*Information gain*
X	1.0	0.6925	0.3075
Y	**1.0**	**0**	**1.0**
Z	1.0	1.0	0

Hence, the largest information gain is provided by the attribute 'Y' thus it is used for the split as depicted in Figure 5.8.

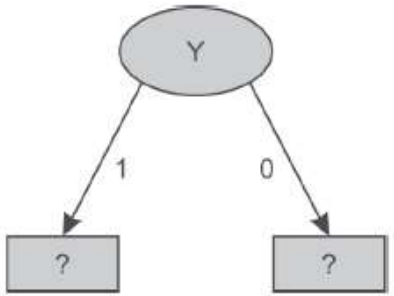

Figure 5.8 Data splitting based on Y attribute

For Y, as there are two possible values, i.e., 1 and 0, therefore the dataset will be split into two subsets based on distinct values of the Y attribute as shown in Figure 5.8.

Dataset for Y = '1'

Instance Number	X	Z	Class
1	1	1	A
2	1	0	A

There are 2 samples and the frequency of each class is as follows.

A = 2 (Instances 1, 2)
B = 0 Instances

Information of the whole dataset on the basis of class is given by

$$I = (-2/2)\log(2/2) - (0/2)\log(0/2) = 0$$

As it represents the same class 'A' for all recorded combinations of X and Z, therefore, it represents class 'A' as shown in Figure 5.9.

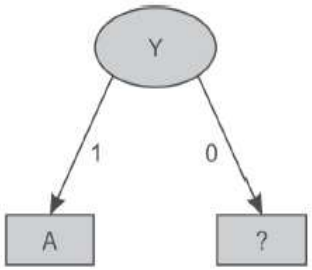

Figure 5.9 Decision tree after splitting of attribute Y having value '1'

Dataset for Y = '0'

Instance Number	X	Z	Class
3	0	1	B
4	1	0	B

For Y having value 0, it represents the same class 'B' for all the records. Thus, the decision tree will look like as shown in Figure 5.10 after analysis of Y dataset.

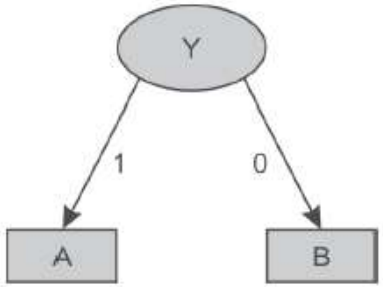

Figure 5.10 Decision tree after splitting of attribute Y value '0'

Let us consider another example and build a decision tree for the dataset given in Figure 5.11. It has 4 input attributes outlook, temperature, humidity and windy. As before we have added instance number for explanation purposes. Here, 'play' is the output attribute and these 14 records contain the information about weather conditions based on which it was decided if a play took place or not.

Instance Number	*Outlook*	*Temperature*	*Humidity*	*Windy*	*Play*
1	sunny	hot	high	false	No
2	sunny	hot	high	true	No
3	overcast	hot	high	false	Yes
4	rainy	mild	high	false	Yes
5	rainy	cool	normal	false	Yes
6	rainy	cool	normal	true	No
7	overcast	cool	normal	true	Yes
8	sunny	mild	high	false	No
9	sunny	cool	normal	false	Yes
10	rainy	mild	normal	false	Yes
11	sunny	mild	normal	true	Yes
12	overcast	mild	high	true	Yes
13	overcast	hot	normal	false	Yes
14	rainy	mild	high	true	No

Attribute values and counts

Outlook	Temp.	Humidity	Windy	Play
sunny = 5	hot = 4	high = 7	true = 6	yes = 9
overcast = 4	mild = 6	normal = 7	false = 8	no = 5
rainy = 5	cool = 4			

Figure 5.11 Dataset for play prediction based on given day weather conditions

In the dataset, there are 14 samples and two classes for target attribute 'Play', i.e., Yes or No. The frequencies of these two classes are given as follows:

Yes = 9 (Instance number 3,4,5,7,9,10,11,12,13 and 14)

No = 5 (Instance number 1,2,6,8 and 15)

Information of the whole dataset on the basis of whether play is held or not is given by

I = - probability for play being held * log (probability for play being held) - probability for play not being held * log (probability for play not being held)

Here, probability for play having value Yes = (Number of instances for Play is Yes/Total number of instances) = 9/14

And probability for play having value No = (Number of instances for Play is No/Total number of instances) = 5/14

Therefore, I = - (9/14) log (9/14) – (5/14) log (5/14) = 0.9435142

Let us consider each attribute one by one as split attributes and calculate the information for each attribute.

Attribute 'Outlook'

As given in dataset, there are three possible values of outlook, i.e., sunny, overcast and rainy. Let us analyze each case one by one.

There are 5 instances where outlook is sunny. Out of these 5 instances, in 2 instances (9 and 11) the play is held and in remaining 3 instances (1, 2 and 8) the play is not held.

There are 4 instances where outlook is overcast and in all these instances the play always takes place.

There are 5 instances where outlook is rainy. Out of these 5 instances, in 3 instances (4, 5 and 10) the play is held whereas in remaining 2 instances (6 and 14) the play is not held.

Given the above values, let us compute the information provided by the outlook attribute. We divide the dataset into three subsets according to outlook conditions being sunny, overcast or rainy. Computing information for each case,

$$I(\text{Sunny}) = I(S) = -(2/5)\log(2/5) - (3/5)\log(3/5) = 0.97428$$

$$I(\text{Overcast}) = I(O) = -(4/4)\log(4/4) - (0/4)\log(0/4) = 0$$

$$I(\text{Rainy}) = I(R) = -(3/5)\log(3/5) - (2/5)\log(2/5) = 0.97428$$

Total information for these three sub-trees = probability for outlook Sunny * I(S) + probability for outlook Overcast * I(O) + probability for outlook Rainy * I(R)

Here, probability for outlook Sunny = (Number of instances for outlook Sunny/Total number of instances) = 5/14

And probability for outlook Overcast = (Number of instances for outlook Overcast/Total number of instances) = 4/14

And probability for outlook Rainy = (Number of instances for outlook Rainy/Total number of instances) = 5/14

Therefore, total information for three sub-trees = (5/14) I(S) + (4/14) I(O) + (5/14) I(R)

= 0.3479586 + 0.3479586

= 0.695917

Attribute 'Temperature'

There are three possible values of the Temperature attribute, i.e., Hot, Mild and Cool. Let us analyze each case one by one.

There are 4 instances for Temperature hot. Play is held in case of 2 of these instances (3 and 13) and is not held in case of the other 2 instances (1 and 2).

There are 6 instances for Temperature mild. Play is held in case of 4 instances (4, 10, 11 and 12) and is not held in case of 2 instances (8 and 14).

There are 4 instances for Temperature cool. Play is held in case of 3 instances (5, 7 and 9) and is not held in case of 1 instance (6).

Given the above values, let us compute the information provided by Temperature attribute. We divide the dataset into three subsets according to temperature conditions being Hot, Mild or Cool. Computing information for each case,

$$I(\text{Hot}) = I(H) = -(2/4)\log(2/4) - (2/4)\log(2/4) = 1.003433$$

$$I(\text{Mild}) = I(M) = -(4/6)\log(4/6) - (2/6)\log(2/6) = 0.9214486$$

$$I(\text{Cool}) = I(C) = -(3/4)\log(3/4) - (1/4)\log(1/4) = 0.814063501$$

Total information for the three sub-trees = probability for temperature hot * I(H) + probability for temperature mild * I(M) + probability for temperature cool * I(C)

Here, probability for temperature hot = (Number of instances for temperature hot/Total number of instances) = 4/14

And probability for temperature mild = (Number of instances for temperature mild/Total number of instances) = 6/14

And probability for temperature cool = (Number of instances for temperature cool/Total number of instances) = 4/14

Therefore, total information for these three sub-trees

$$= (4/14)\,I(H) + (6/14)\,I(M) + (4/14)\,I(C)$$
$$= 0.2866951429 + 0.3949065429 + 0.23258957$$
$$= 0.9141912558$$

Attribute 'Humidity'

There are two possible values of the Humidity attribute, i.e., High and Normal. Let us analyze each case one by one.

There are 7 instances where humidity is high. Play is held in case of 3 instances (3, 4 and 12) and is not held in case of 4 instances (1, 2, 8 and 14).

There are 7 instances where humidity is normal. Play is held in case of 6 instances (5, 7, 9, 10, 11 and 13) and is not held in case of 1 instance (6).

Given the above values, let us compute the information provided by the humidity attribute. We divide the dataset into two subsets according to humidity conditions being high or normal. Computing information for each case,

$$I(\text{High}) = I(H) = -(3/7)\log(3/7) - (4/7)\log(4/7) = 0.98861$$

$$I(\text{Normal}) = I(N) = -(6/7)\log(6/7) - (1/7)\log(1/7) = 0.593704$$

Total information for the two sub-trees = probability for humidity high * I(H) + probability for humidity normal * I(N)

Here, probability for humidity high = (Number of instances for humidity high/Total number of instances) = 7/14

And probability for humidity normal = (Number of instances for humidity normal/Total number of instances) = 7/14

Therefore, Total information for the two sub-trees = (7/14) I(H) + (7/14) I(N)

= 0.494305 + 0.29685

= 0.791157

Attribute 'Windy'

There are two possible values for this attribute, i.e., true and false. Let us analyze each case one by one.

There are 6 instances when it is windy. On windy days, play is held in case of 3 instances (7, 11 and 12) and is not held in case of remaining 3 instances (2, 6 and 14).

For non-windy days, there are 8 instances. On non-windy days, the play is held in case of 6 instances (3, 4, 5, 9, 10 and 13) and is not held in case of 2 instances (1 and 8).

Given the above values, let us compute the information provided by the windy attribute. We divide the dataset into two subsets according to windy being true or false. Computing information for each case,

$$I(\text{True}) = I(T) = -(3/6)\log(3/6) - (3/6)\log(3/6) = 1.003433$$

$$I(\text{False}) = I(F) = -(6/8)\log(6/8) - (2/8)\log(2/8) = 0.81406$$

Total information for the two sub-trees = probability for windy true * I(T) + probability for windy true * I(F)

Here, The probability for windy being True = (Number of instances for windy true/Total number of instances) = 6/14

And probability for windy being False = (Number of instances for windy false/Total number of instances) = 8/14

Therefore, Total information for the two sub-trees = (6/14) I(T) + (8/14) I(F)

$$= 0.4300427 + 0.465179$$

$$= 0.89522$$

The information gain can now be computed:

Potential Split attribute	*Information before split*	*Information after split*	*Information gain*
Outlook	0.9435	0.6959	0.2476
Temperature	0.9435	0.9142	0.0293
Humidity	0.9435	0.7912	0.15234
Windy	0.9435	0.8952	0.0483

From the above table, it is clear that the largest information gain is provided by the attribute 'Outlook' so it is used for the split as depicted in Figure 5.12.

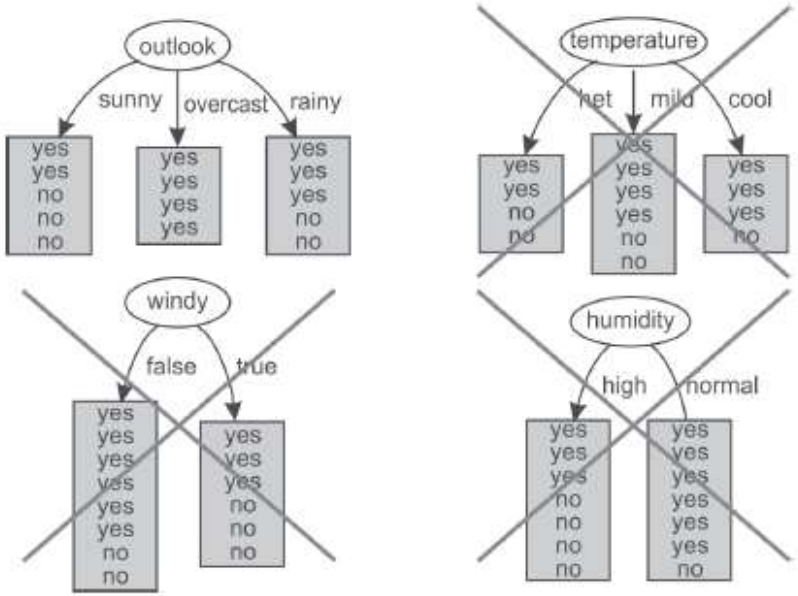

Figure 5.12 Selection of Outlook as root attribute

For Outlook, as there are three possible values, i.e., Sunny, Overcast and Rain, the dataset will be split into three subsets based on distinct values of the Outlook attribute as shown in Figure 5.13.

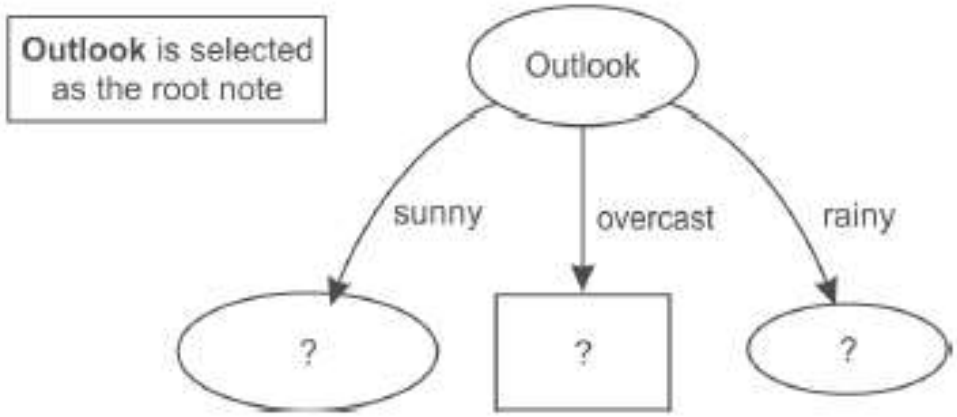

Figure 5.13 Data splitting based on the Outlook attribute

Dataset for Outlook 'Sunny'

Instance Number	Temperature	Humidity	Windy	Play
1	Hot	high	false	No
2	Hot	high	true	No
3	Mild	high	false	No
4	Cool	normal	false	Yes
5	Mild	normal	true	Yes

Again, in this dataset, a new column instance number is added to the dataset for making explanation easier. In this case, we have three input attributes Temperature, Humidity and Windy. This dataset consists of 5 samples and two classes, i.e., Yes and No for the Play attribute. The frequencies of classes are as follows:

Yes = 2 (Instances 4, 5)

No = 3 (Instances 1, 2, 3)

Information of the whole dataset on the basis of whether play is held or not is given by

I = - probability for Play Yes * log (probability for Play Yes)
- probability for Play No * log (probability for Play No)

Here, probability for play being Yes = (Number of instances for Play Yes/Total number of instances) = 2/5

And probability for Play being No = (Number of instances for Play No/Total number of instances) = 3/5

Therefore, I = (-2/5) log (2/5) – (3/5) log(3/5) = 0.97

Let us consider each attribute one by one as a split attribute and calculate the information for each attribute.

Attribute 'Temperature'

There are three possible values of Temperature attribute, i.e., hot, mild and cool. Let us analyze each case one by one.

There are 2 instances for temperature hot. Play is never held when the temperature is hot as shown in 2 instances (1 and 2).

There are 2 instances when temperature is mild. Play is held in case of 1 instance (5) and is not held in case of 1 instance (3).

There is 1 instance when temperature is cool. Play is held in this case as shown in instance 4.

Given the above values, let us compute the information provided by this attribute. We divide the dataset into three subsets according to temperature conditions being hot, mild or cool. Computing information for each case,

$$I(\text{Hot}) = I(H) = -(0/2)\log(0/2) - (2/2)\log(2/2) = 0$$

$$I(\text{Mild}) = I(M) = -(1/2)\log(1/2) - (1/2)\log(1/2) = 1.003433$$

$$I(\text{Cool}) = I(C) = -(1/1)\log(1/1) - (0/1)\log(1/1) = 0$$

Total information for these three sub-trees = probability for temperature hot * I(H) + probability for temperature mild * I(M) + probability for temperature cool * I(C)

Here, probability for temperature hot = (No of instances for temperature hot/Total no of instances) = 2/5

And probability for temperature mild = (Number of instances for temperature mild/Total number of instances) = 2/5

And probability for temperature cool = (Number of instances for temperature cool/Total number of instances) = 1/5

Therefore, total information for three subtrees = (2/5) I(H) + (2/5) I(M) + (1/5) I(C)

$$= 0 + (0.4) * (1.003433) + 0$$

$$= 0.40137$$

Attribute 'Humidity'

There are two possible values of Humidity attribute, i.e., High and Normal. Let us analyze each case one by one.

There are 3 instances when humidity is high. Play is never held when humidity is high as shown in case of 3 instances (1, 2 and 3).

There are 2 instances when humidity is normal. Play is always held as shown in case of 2 instances (4 and 5).

Given the above values, let us compute the information provided by this attribute. We divide the dataset into two subsets according to humidity conditions being high or normal. Computing information for each case,

$$I\ (High) = I\ (H) = -(0/3)\ \log(0/3)-(3/3)\ \log(3/3) = 0$$

$$I\ (Normal) = I\ (N) = -(2/2)\ \log(2/2)-(0/2)\ \log(0/2) = 0$$

Total information for these two sub-trees = probability for humidity high * I(H) + probability for humidity normal * I(N)

Here, probability for humidity high = (Number of instances for humidity high/Total number of instances) = 3/5

And probability for humidity normal = (Number of instances for humidity normal/Total number of instances) = 2/5

Therefore, total information for these two sub-trees = (3/5) I(H) + (2/5) I(N) = 0

Attribute 'Windy'

There are two possible values for this attribute, i.e., true and false. Let us analyze each case one by one.

There are 2 instances when windy is true. On windy days play is held in case of 1 instance (5) and it is not held in case of another 1 instance (2).

There are 3 instances when windy is false. The play is held in case of 1 instance (4) and is not held in case of 2 instances (1 and 3).

Given the above values, let us compute the information by using this attribute. We divide the dataset into two subsets according to windy being true or false. Computing information for each case,

$$I(\text{True}) = I(T) = -(1/2)\log(1/2) - (1/2)\log(1/2) = 1.003433$$

$$I(\text{False}) = I(F) = -(1/3)\log(1/3) - (2/3)\log(2/3) = 0.9214486$$

Total information for these two sub-trees = probability for windy true * I(T) + probability for windy true * I(F)

Here, the probability for windy true = (Number of instances for windy true/Total number of instances) = 2/5

And probability for windy false = (Number of instances for windy false/Total number of instances) = 3/5

Therefore, total information for two sub-trees = (2/5) I(T) + (3/5) I(F)

= 0.40137 + 0.55286818

= 0.954239

The Information gain can now be computed:

Potential Split attribute	*Information before split*	*Information after split*	*Information gain*
Temperature	0.97	0.40137	0.56863
Humidity	0.97	0	0.97
Windy	0.97	0.954239	0.015761

Thus, the largest information gain is provided by the attribute 'Humidity' and it is used for the split. This algorithm can be tuned by stopping when we get the 0 value of information as in this case to reduce the number of calculations for big datasets. Now, the Humidity attribute is selected as split attribute as depicted in Figure 5.14.

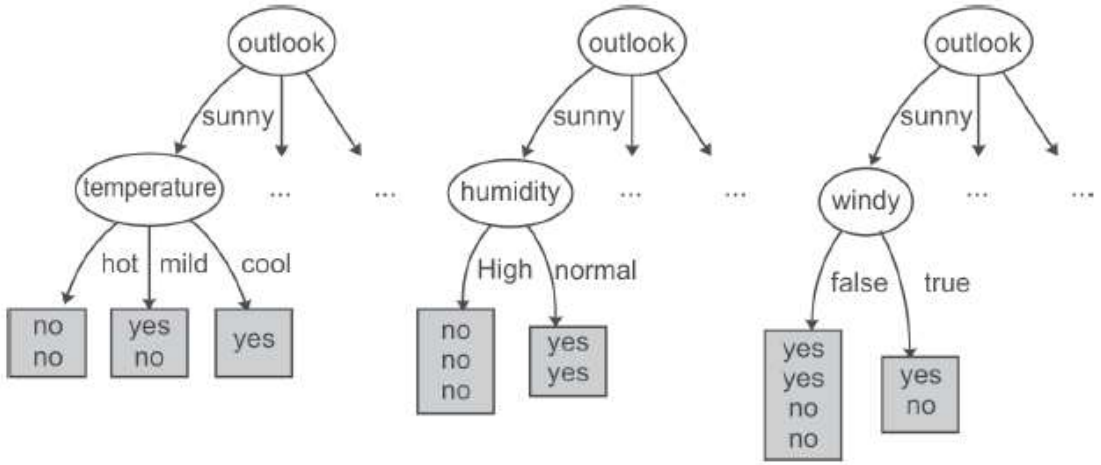

Figure 5.14 Humidity attribute is selected from dataset of Sunny instances

There are two possible values of humidity so data is split into two parts, i.e. humidity 'high' and humidity 'low' as shown below.

Dataset for Humidity 'High'

Instance Number	Temperature	Windy	Play
1	hot	false	No
2	hot	true	No
3	mild	false	No

Again, in this dataset a new column instance number has been introduced for explanation purposes. For this dataset, we have two input attributes Temperature and Windy. As the dataset represents the same class 'No' for all the records, therefore for Humidity value 'High' the output class is always 'No'.

Dataset for Humidity 'Normal'

Instance No	Temperature	Windy	Play
1	cool	false	Yes
2	mild	true	Yes

When humidity's value is 'Normal' the output class is always 'Yes'. Thus, decision tree will look like as shown in Figure 5.15 after analysis of the Humidity dataset.

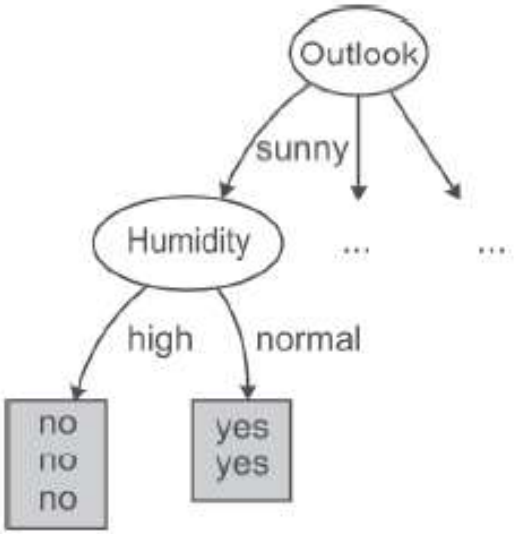

Figure 5.15 Decision tree after spitting of data on Humidity attribute

Now, the analysis of Sunny dataset is over. From the decision tree shown in Figure 5.15, it has been analyzed that if the outlook is 'Sunny' and humidity is 'normal' then play is always held while on the other hand if the humidity is 'high' then play is not held.

Let us take next subset which has Outlook as 'Overcast' for further analysis.

Dataset for Outlook 'Overcast'

Instance Number	*Temperature*	*Humidity*	*Windy*	*Play*
1	hot	high	false	Yes
2	cool	normal	true	Yes
3	mild	high	true	Yes

For outlook Overcast, the output class is always 'Yes'. Thus, decision tree will look like Figure 5.16 after analysis of the overcast dataset.

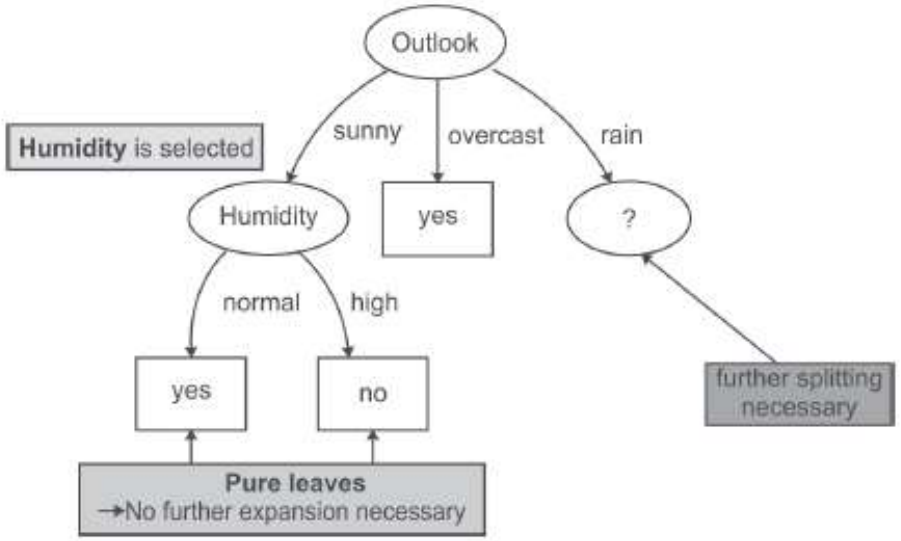

Figure 5.16 Decision tree after analysis of Sunny and Overcast dataset

Dataset for Outlook 'Rainy'

Instance Number	Temperature	Humidity	Windy	Play
1	Mild	High	False	Yes
2	Cool	Normal	False	Yes
3	Cool	Normal	True	No
4	Mild	Normal	False	Yes
5	Mild	High	True	No

In the above dataset, a new column for instance numbers has again been added for ease of explanation. This dataset consists of 5 samples having input attributes Temperature, Humidity and Windy; and the single output attribute Play has two classes, i.e., Yes and No. The frequencies of these classes are as follows:

Yes = 3 (Instances 1, 2, 4)

No = 2 (Instances 3, 5)

Information of the whole dataset on the basis of whether play is held or not is given by

I = - probability for Play Yes * log (probability for Play Yes)
- probability for Play No * log (probability for Play No)

Here, probability for Play Yes = (Number of instances for Play Yes/Total number of instances) = 3/5

And probability for Play No = (Number of instances for Play No/Total number of instances) = 2/5

Therefore, I = (-3/5) log (3/5) – (2/5) log (2/5) = 0.97428

Let us consider each attribute one by one as split attributes and calculate the information provided for each attribute.

Attribute 'Temperature'

There are two possible values of Temperature attribute, i.e., mild and cool. Let us analyze each case one by one.

There are 3 instances with temperature value mild. Play is held in case of 2 instances (1 and 4) and is not held in case of 1 instance (5).

There are 2 instances with temperature value cool. Play is held in case of 1 instance (2) and is not held in case of other instance (3).

Given the above values, let us compute the information provided by this attribute. We divide the dataset into two subsets according to temperature being mild and cool. Computing information for each case,

$$I\ (\text{Mild}) = I(M) = -\ (2/3)\ \log(2/3) - (1/3)\ \log(1/3) = 0.921545$$

$$I\ (\text{Cool}) = I(C) = -\ (1/2)\ \log(1/2) - (1/2)\ \log(1/2) = 1.003433$$

Total information for the two sub-trees = probability for temperature mild * I(M) + probability for temperature cool * I(C)

And probability for temperature mild = (Number of instances for temperature mild/Total number of instances) = 3/5

And probability for temperature cool = (Number of instances for temperature cool/Total number of instances) = 2/5

Therefore, total information for three subtrees = (3/5) I(M) + (2/5) I(C)

= 0.552927 + 0.40137

= 0.954297

Attribute 'Humidity'

There are two possible values of Humidity attribute, i.e., High and Normal. Let us analyze each case one by one.

There are 2 instances with high humidity. Play is held in case of 1 instance (1) and is not held in case of another instance (5).

There are 3 instances with normal humidity. The play is held in case of 2 instances (2 and 4) and is not held in case of single instance (3).

Given the above values, let us compute the information provided by this attribute. We divide the dataset into two subsets according to humidity being high or normal. Computing information for each case,

$$I(\text{High}) = I(H) = -(1/2)\log(1/2) - (1/2)\log(1/2) = 1.0$$

$$I(\text{Normal}) = I(N) = -(2/3)\log(2/3) - (1/3)\log(1/3) = 0.9187$$

Total information for the two sub-trees = probability for humidity high * I(H) + probability for humidity normal * I(N)

Here, probability for humidity high = (Number of instances for humidity high/Total number of instances) = 2/5

And probability for humidity normal = (Number of instances for humidity normal/Total number of instances) = 3/5

Therefore, total information for the two sub-trees = (2/5) I(H) + (3/5) I(N)

= 0.4 + 0.5512 = 0.9512

Attribute 'Windy'

There are two possible values for this attribute, i.e., true and false. Let us analyze each case one by one.

There are 2 instances where windy is true. Play is not held in case of both of the 2 instances (3 and 5).

For non-windy days, there are 3 instances. Play is held in all of the 3 instances (1, 2 and 4).

Given the above values, let us compute the information provided by this attribute. We divide the dataset into two subsets according to windy being true or false. Computing information for each case,

$$I(\text{True}) = I(T) = -(0/2)\log(0/2) - (2/2)\log(2/2) = 0$$

$$I(\text{False}) = I(F) = -(3/3)\log(3/3) - (0/3)\log(0/3) = 0$$

Total information for the two sub-trees = probability for windy true * I(T) + probability for windy false * I(F)

Here, probability for windy true = (Number of instances for windy true/Total number of instances) = 2/5

And, probability for windy false = (Number of instances for windy false/Total number of instances) = 3/5

Therefore, total information for the two sub-trees = (2/5) I(T) + (3/5) I(F) = 0

The Information gain can now be computed:

Potential Split attribute	*Information before split*	*Information after split*	*Information gain*
Temperature	0.97428	0.954297	0.019987
Humidity	0.97428	0.9512	0.02308
Windy	0.97428	0	0.97428

Hence, the largest information gain is provided by the attribute 'Windy' and this attribute is used for the split.

Dataset for Windy 'TRUE'

Instance Number	*Temperature*	*Humidity*	*Play*
1	cool	normal	No
2	mild	high	No

From above table it is clear that for Windy value 'TRUE', the output class is always 'No'.

Dataset for Windy 'FALSE'

Instance Number	*Temperature*	*Humidity*	*Play*
1	Mild	High	Yes
2	Cool	Normal	Yes
3	Mild	Normal	Yes

Also for Windy value 'FALSE', the output class is always 'Yes'. Thus, the decision tree will look like Figure 5.17 after analysis of the rainy attribute.

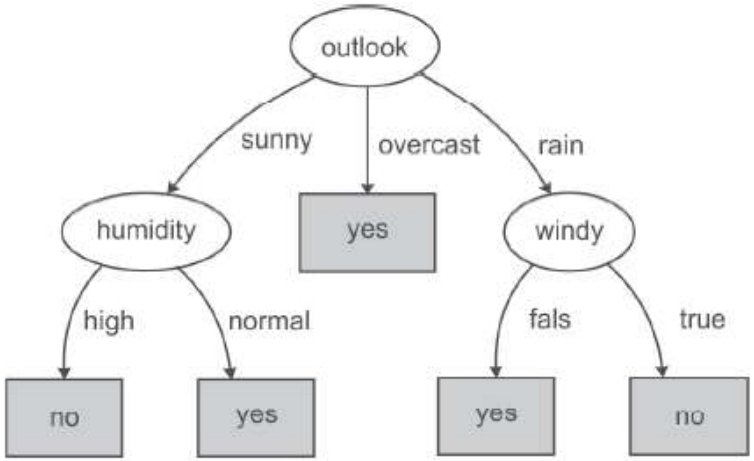

Figure 5.17 Decision tree after analysis of Sunny, Overcast and Rainy dataset

We have concluded that if the outlook is 'Rainy' and value of windy is 'False' then play is held and on the other hand, if value of windy is 'True' then it means that play is not held in that case.

We have concluded that if the outlook is 'Rainy' and value of windy is 'False' then play is held and on the other hand, if value of windy is 'True' then it means that play is not held in that case.

Figure 5.18 represents the final tree view of all the 14 records of the dataset given in Figure 5.11 when it is generated using Weka for the prediction of play on the basis of given weather conditions. The numbers shown along with the classes in the tree represent the number of instances that are classified under that node. For example, for outlook 'overcast', play is always held and there are total 4 instances in the dataset which agree with this rule. Similarly, there are 2 instances in the dataset for which play is not held if outlook is rainy and windy is true.

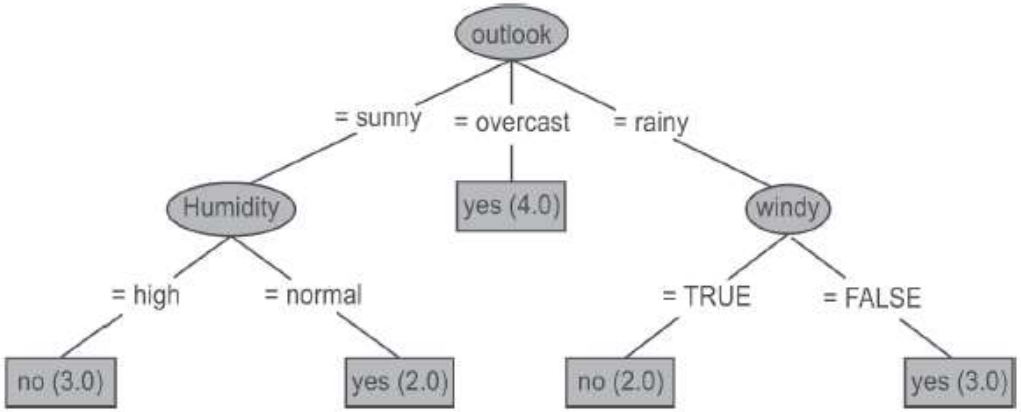

Figure 5.18 Final decision tree after analysis of Sunny, Overcast and Rainy dataset

Suppose, we have to predict whether the play will be held or not for an unknown instance having Temperature 'mild', Outlook 'sunny', Humidity 'normal' and windy 'false', it can be easily predicted on the basis of decision tree shown in Figure 5.18. For the given unknown instance, the play will be held based on conditional checks as shown in Figure 5.19.

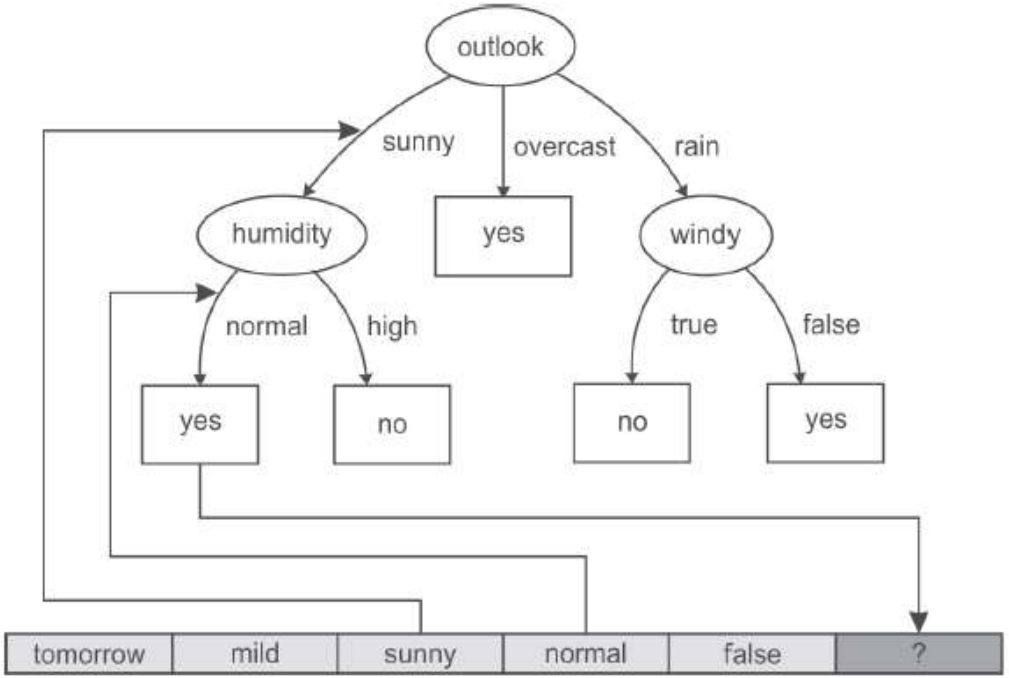

Figure 5.19 Prediction of Play for an unknown instance