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.

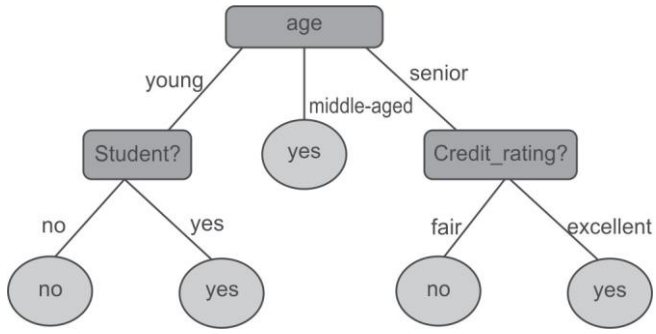

Figure 5.6 Decision tree to predict whether a customer will buy a laptop or not

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.

5.1.1 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.

5.1.2 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.

5.1.3 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.

5.1.4 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.

5.1.5 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.

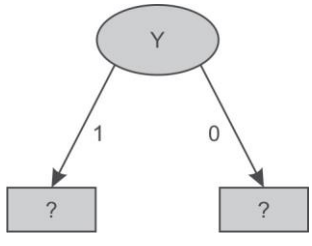

Figure 5.8 Data splitting based on Y attribute

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.

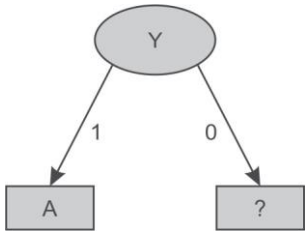

Figure 5.9 Decision tree after splitting of attribute Y having value '1'

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.

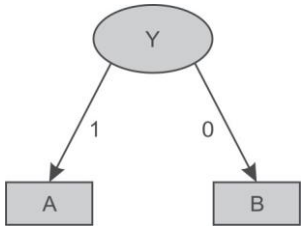

Figure 5.10 Decision tree after splitting of attribute Y value '0'

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,

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

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

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

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,

$$I(High) = I(H) = -(3/7)\log(3/7) - (4/7)\log(4/7) = 0.98861$$
$$I(Normal) = I(N) = -(6/7)\log(6/7) - (1/7)\log(1/7) = 0.593704$$

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.

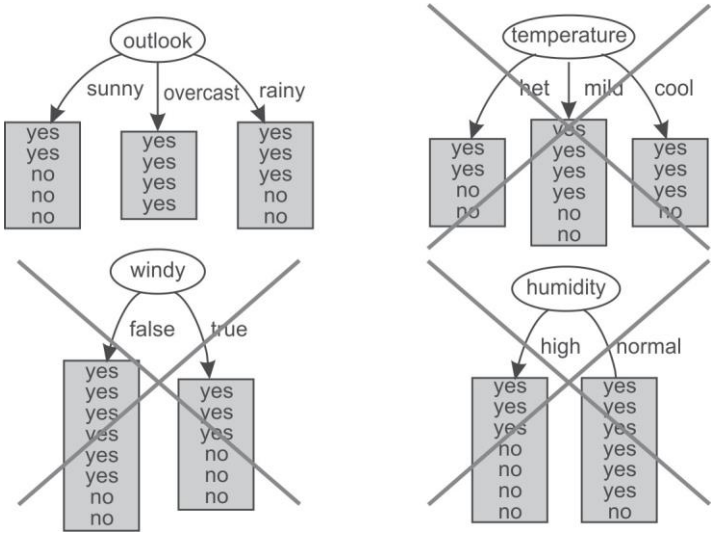

Figure 5.12 Selection of Outlook as root attribute

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.

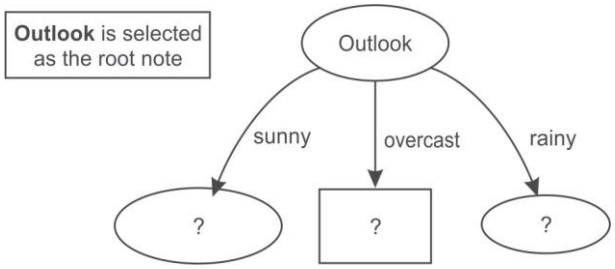

Figure 5.13 Data splitting based on the Outlook attribute

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,

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

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

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,

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

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

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.

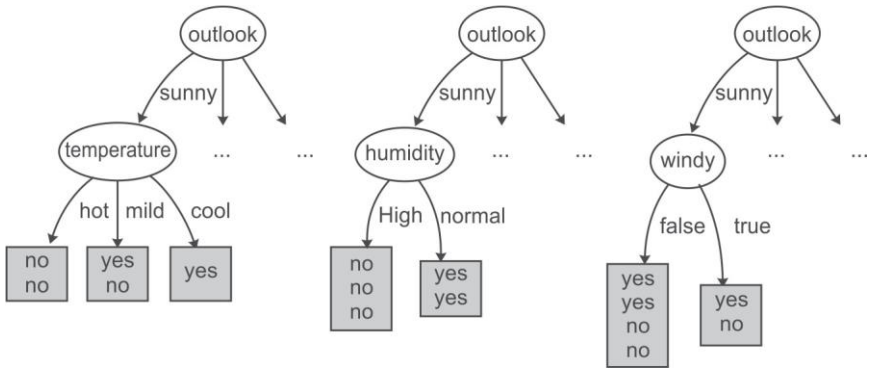

Figure 5.14 Humidity attribute is selected from dataset of Sunny instances

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.

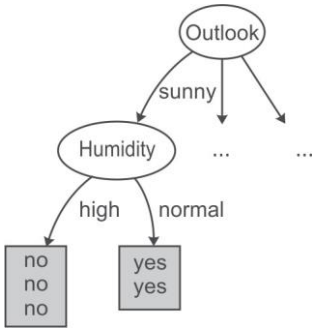

Figure 5.15 Decision tree after spitting of data on Humidity attribute

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.

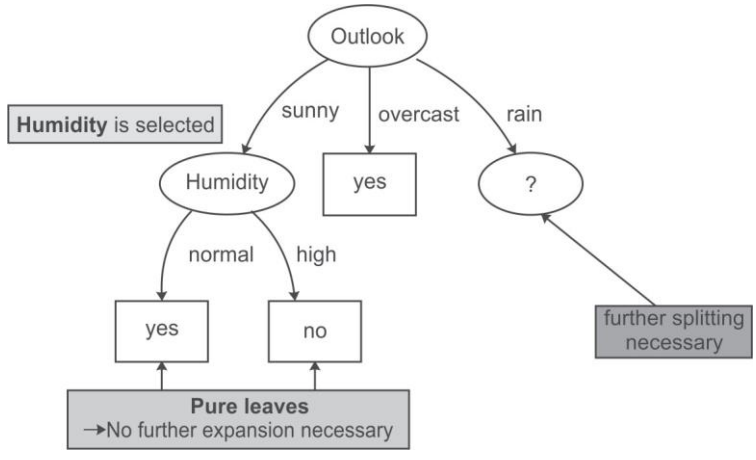

Figure 5.16 Decision tree after analysis of Sunny and Overcast dataset

Therefore, it has been concluded that if the outlook is 'Overcast' then play is always held. Now we have to select another split attribute for the outlook rainy and subset of the dataset for this is given below.

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,

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

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

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,

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

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

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,

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

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

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.

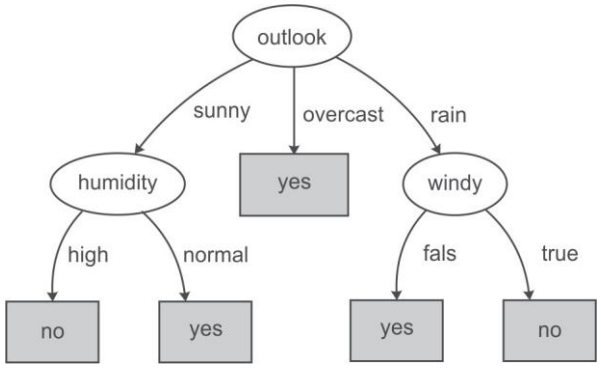

Figure 5.17 Decision tree after analysis of Sunny, Overcast and Rainy dataset

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.

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.

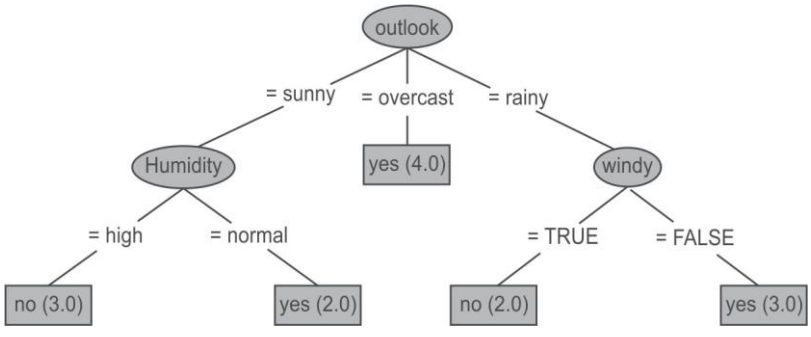

Figure 5.18 Final decision tree after analysis of Sunny, Overcast and Rainy dataset

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.

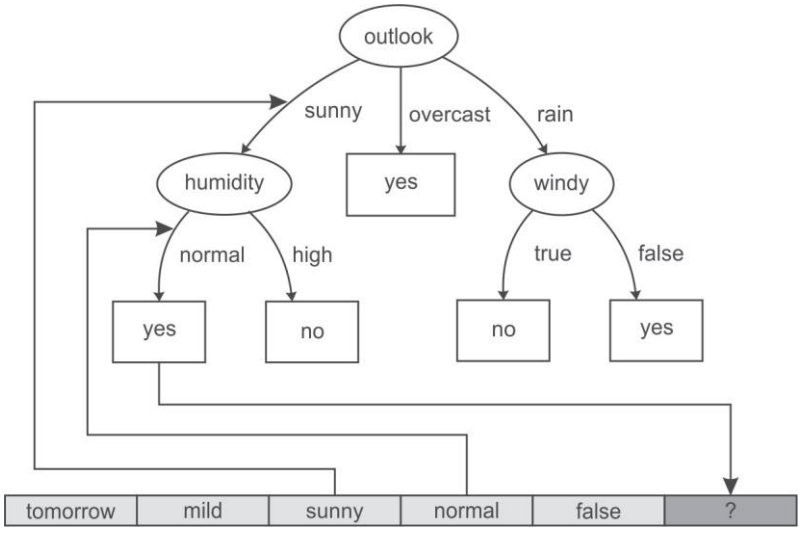

Figure 5.19 Prediction of Play for an unknown instance

5.1.6 Drawbacks of information gain theory

Though information gain is a good measure for determining the significance of an attribute, it has its own limitations. When it is applied to attributes with a large number of distinct values, a noticeable problem occurs. For example, while building a decision tree for a dataset consisting of the records of customers. In this case, information gain is used to determine the most significant attributes so that they can be tested for the root of the tree. Suppose, customers' credit card number is one of the input attributes in the customer dataset. As this input attribute uniquely identifies each customer, it has high mutual information. However, it cannot be included in the decision tree for making decisions because not all customers have credit cards and there is also a problem of how to treat them on the basis of their credit card number. This method will not work for all the customers.

5.1.7 Split algorithm based on Gini Index

The Gini Index is used to represent level of equality or inequality among objects. It can also be used to make decision trees. It was developed by Italian scientist Corrado Gini in 1912 and was used to analyze equality distribution of income among people. We will consider the example of wealth distribution in society in order to understand the concept of the Gini Index. The Gini Index always ranges between 0 and 1. It was designed to define the gap between the rich and the poor, with 0 signifying perfect equality where all people have the same income while 1 demonstrating perfect inequality where only one person gets everything in terms of income and rest of the others get nothing.

From this, it is evident that if Gini Index is very high, there will be huge inequality in income distribution. Therefore, we will be interested in knowing person's income. But in a society where everyone has same income then no one will be interested in knowing each other's income because

they know that everyone is at the same level. Thus, the attribute of interest can be decided on the basis that if attribute has a high value Gini Index then it carriers more information and if it has a low value Gini Index then its information content is low.

To define the index, a graph is plotted by considering the percentage of income of the population as the Y-axis and the percentage of the population as the X-axis as shown in Figure 5.20.

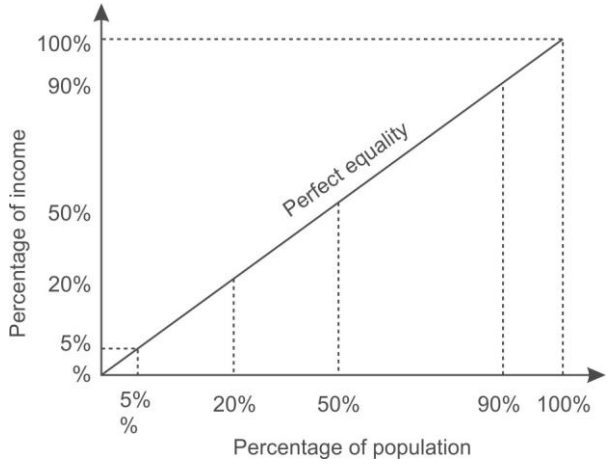

Figure 5.20 Gini Index representing perfect equality

In case of total equality in society, 5% of the people own 5% of the wealth, 20% of the people own 20% of the wealth similarly 90% of people own 90% of wealth. The line at 45 degrees thus represents perfect equal distribution of income in society.

The Gini Index is the ratio of the area between the Lorenz curve and the 45 degree line to the area under the 45 degree line given as follows.

$$\text{Gini Index (G)} = \frac{\text{area between the Lorenz curve and the 45 – degree line}}{\text{area under the 45 – degree line}}$$

As shown in Figure 5.21, the area that lies between the line of equality and the Lorenz curve is marked with *A* and the total area under the line of equality is represented by *(A + B)* in the figure. Therefore,

$$G = \frac{A}{(A + B)}$$

Smaller the ratio, lesser is the area between the two curves and more evenly distributed is the wealth in the society.

The most equal society will be the one in which every person has the same income, making $A = 0$, thus making $G = 0$.

However, the most unequal society will be the one in which a single person receives 100% of the total wealth thus making $B = 0$ and $G = A/A = 1$. From this, we can observe that G always lies between 0 and 1.

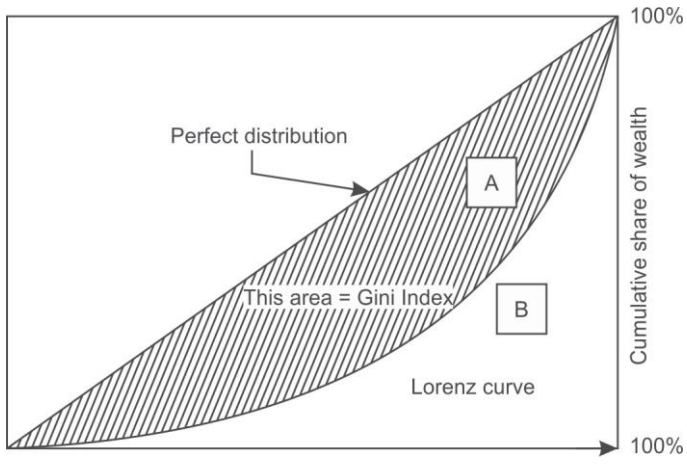

Figure 5.21 Lorenz curve

The more nearly equal a country's income distribution is, the closer is its Lorenz curve to the 45 degree line and the lower is its Gini Index. The more unequal a country's income distribution is, the farther is its Lorenz curve from the 45 degree line and the higher is its Gini Index. For example, as shown in Figure 5.22 (a), the bottom 10% of the people have 90% of total income, therefore Gini Index will be larger as the area between the equality line and the Lorenz curve is larger because there is large inequality in income distribution. Similarly, as shown in Figure 5.22 (b), the bottom 20% of the people have 80% of the total income and the bottom 30% of the people have 70% of the total income is shown in Figure 5.22 (c).

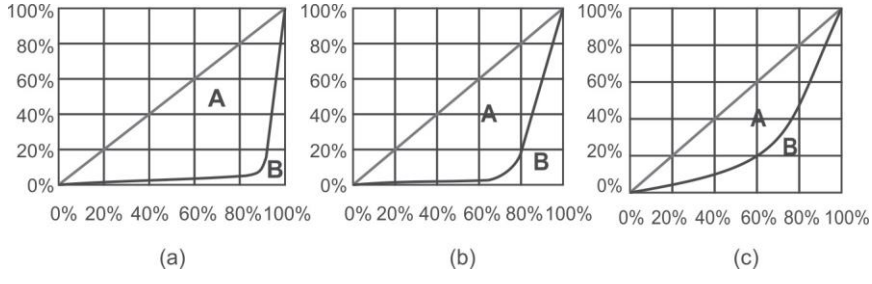

Figure 5.22 Lorenz curves with varying income distributions

From this, it can be concluded that, if income were distributed with perfect equality, the Lorenz curve would coincide with the 45 degree line and the Gini Index would be zero. However, if income were distributed with perfect inequality, the Lorenz curve would coincide with the horizontal axis and the right vertical axis and the index would be 1.

Gini Index can also be calculated in terms of probability and if a dataset D contains instances from n classes, the Gini Index, G(D), is defined as

$$G(D) = 1 - \sum(p_i)^{\text{No. of classes}}$$

Here, p_i is the relative frequency or probability of class *i* in *D. Let us calculate the Gini Index for tossing an unbiased coin.*

$$G = 1 - (\text{Probability of getting head})^{\text{No of classes}} - (\text{Probability of getting tail})^{\text{No of classes}}$$

$$G = 1 - (0.5)^2 - (0.5)^2 = 0.5$$

But if coin is biased having head on both sides, then there is 100% chance of a head and 0% chance of tail then the Gini Index is:

$$G = 1 - (1)^2 = 0$$

As there is no uncertainty about the coin in this case so the index is 0. It is maximum at a value of 0.5 in case of an unbiased coin. If the coin is biased, such that head occurs 60% of the times. then the Gini Index is reduced to 0.48.

Similarly we can calculate the Gini Index for a dice with six possible outcomes with equal probability as

$$G = 1 - 6(1/6)^2 = 5/6 = 0.833$$

If the dice is biased, let us say there is 50% chance of getting 6 and remaining 50% is being shared by other 5 numbers leaving only a 10% chance of getting each number other than 6 then the Gini Index is

$$G = 1 - 5(0.1)^2 - (0.5)^2 = 0.70$$

$$G = 1 - 5(0.05)^2 - (0.75)^2 = 0.425$$

Here, Gini Index has been reduced from 0.833 to 0.70. Clearly, the high value of index means high uncertainty.

It is important to observe that the Gini Index behaves in the same manner as the information gain discussed in Section 5.6.4. Table 5.1 clearly shows that same trend in both the cases.

Table 5.1 Information and Gini Index for a number of events

Event	*Information*	*Gini Index*
Toss of a unbiased coin	1.0	0.5
Toss of a biased coin (60% heads)	0.881	0.42
Throw of a unbiased dice	2.585	0.83
Throw of a biased die (50% chance of a 6)	2.16	0.7

5.1.8 Building a decision tree with Gini Index

Let us build the decision tree for the dataset given in Figure 5.23.

It has 3 input attributes X, Y, Z and one output attribute 'Class'. We have added a new column for instance numbers for easy explanation. The dataset contains four records and the output attribute or class can be either A or B.

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 5.23 Dataset for class C prediction based on given attribute condition

The frequencies of the two output classes are given as follows:

A = 2 (Instances 1,2)

B = 2 (Instances 3,4)

The Gini Index of the whole dataset is calculated as follows:

$$G = 1 - (\text{probability for Class A})^{\text{No of classes}} - (\text{probability for Class B})^{\text{No of classes}}$$

Here, probability for class A = (No of instances for class I/Total no of instances) = 2/4

And probability for class B = (No of instances for class II/Total no of instances) = 2/4

Therefore, $G = 1 - (2/4)^2 - (2/4)^2 = 0.5$

Let us consider each attribute one by one as split attribute and calculate Gini Index for each attribute.

Attribute 'X'

As given in dataset, there are two possible values of X, i.e., 1, 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 number 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 Gini Index by using this attribute. We divide the dataset into two subsets according to X being 1 or 0.

Computing the Gini Index for each case,

$$G(X1) = 1 - (2/3)^2 - (1/3)^2 = 0.44444$$

$$G(X0) = 1 - (0/1)^2 - (1/1)^2 = 0$$

Total Gini Index for the two sub-trees = probability for X having value 1 * G(X1) + probability for X having value 0 * G(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 Gini Index for two subtrees = (3/4) G(X1) + (1/4) G(X0)

= 0.333333+0

= 0.333333

Attribute 'Y'

There are two possible values of the 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 2 instances when Y = 0 both records belong to class B.

Given the above values, let us compute Gini Index by using this attribute. We divide the dataset into two subsets according to Y being 1 or 0.

Computing the Gini Index for each case,

$$G(Y1) = 1 - (2/2)^2 - (0/2)^2 = 0$$

$$G(Y0) = 1 - (0/2)^2 - (2/2)^2 = 0$$

Total Gini Index for two sub-trees = probability for Y having value 1* G(Y1) + probability for Y having value 0 * G(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, Total Gini Index for the two sub-trees = (2/4) G(Y1) + (2/4) G(Y0)

= 0 + 0 = 0

Attribute 'Z'

There are two possible values of the 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 either being 0 or 1.

Given the above values, let us compute the information by using this attribute. We divide the dataset into two subsets according to Z conditions, i.e., 1 and 0.

Computing the Gini Index for each case,

$$G(Z1) = 1 - (1/2)^2 - (1/2)^2 = 0.5$$

$$G(Z0) = 1 - (1/2)^2 - (1/2)^2 = 0.5$$

Total Gini Index for the two sub-trees = probability for Z having value 1 * G(Z1) + probability for Z having value 0 * G(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 the two subtrees = (2/4) G(Z1) + (2/4) G(Z0)

= 0.25 + 0.25 = 0.50

The Gain can now be computed:

Potential Split attribute	*Gini Index before split*	*Gini Index after split*	*Gain*
X	0.5	0.333333	0.166667
Y	0.5	0	0.5
Z	0.5	0.50	0

Since the largest Gain is provided by the attribute 'Y' it is used for the split as depicted in Figure 5.24.

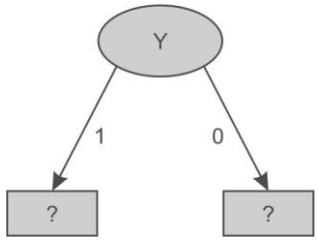

Figure 5.24 Data splitting based on Y attribute

There are two possible values for attribute Y, i.e., 1 or 0, and so the dataset will be split into two subsets based on distinct values of Y attribute as shown in Figure 5.24.

Dataset for Y '1'

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

There are 2 samples records both belonging to class A. Thus, frequencies of classes are as follows:
A = 2 (Instances 1, 2)
B = 0
Information provided by the whole dataset on the basis of class is given by

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

As irrespective of value of X and Z both the records belong to class 'A' hence for Y = 1 the tree will classify as class 'A' as shown in Figure 5.25.

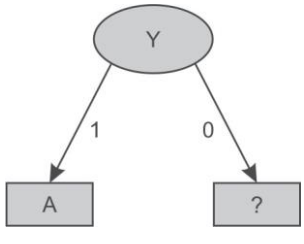

Figure 5.25 Decision tree after splitting of attribute Y having value '1'

Dataset for Y '0'

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

Here, for Y having value 0, both records are classified as class 'B' irrespective of value of X and Z. Thus, the decision tree will look like as shown in Figure 5.26 after analysis of the Y = 0 subset.

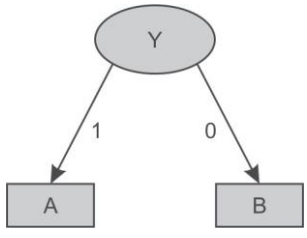

Figure 5.26 Decision tree after splitting of attribute Y value '0'

Let us consider another example to build a decision tree for the dataset given in Figure 5.27. It has four input attributes outlook, temperature, humidity and windy. Instance numbers have been added for easier explanation. Here, 'Play' is the class or output attributes and there are 14 records containing the information if play was held or not, based on weather conditions.

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	*Temperature*	*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 5.27 Dataset for play prediction based on given day weather conditions

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

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

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

The Gini Index of the whole dataset is given by

$$G = 1 - (\text{probability for Play Yes})^{\text{No of classes}} - (\text{probability for Play No})^{\text{No of classes}}$$

Here, probability for Play Yes = 9/14

And, probability for Play No = 5/14

Therefore, $G = 1-(9/14)^2 - (5/14)^2 = 0.45918$

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

Attribute 'Outlook'

As given in the 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 for outlook having value sunny. Play is held in case of 2 instances (9 and 11) and is not held in case of 3 instances (1, 2 and 8).

There are 4 instances for outlook having value overcast. Play is held in case of all the 4 instances (3, 7, 12 and 13).

There are 5 instances for outlook having value rainy. Play is held in case of 3 instances (4, 5 and 10) and is not held in case of 2 instances (6 and 14).

Given the above values, let us compute the Gini Index by using this 'Outlook' attribute. We divide the dataset into three subsets according to outlook being sunny, overcast or rainy. Computing Gini Index for each case,

$$G(\text{Sunny}) = G(S) = 1- (2/5)^2 - (3/5)^2 = 0.48$$

$$G(\text{Overcast}) = G(O) = 1 - (4/4)^2 - (0/4)^2 = 0$$

$$G(\text{Rainy}) = G(R) = 1 - (3/5)^2 - (2/5)^2 = 0.48$$

Total Gini Index for three sub-trees = probability for outlook Sunny * G(S) + probability for outlook Overcast * G(O) + probability for outlook Rainy * G (R)

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

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

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

Therefore, the total Gini Index for the three sub-trees = (5/14) G(S) + (4/14) G(O) + (5/14) G(R)
= 0.171428+ 0 + 0.171428
= 0.342857

Attribute 'Temperature'

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

There are 4 instances for temperature having value hot. Play is held in case of 2 instances (3 and 13) and is not held in case of other 2 instances (1 and 2).

There are 6 instances for temperature having value 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 having value 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 Gini Index by using this attribute. We divide the dataset into three subsets according to temperature conditions, i.e., hot, mild and cool. Computing the Gini Index for each case,

$$G(\text{Hot}) = G(H) = 1 - (2/4)^2 - (2/4)^2 = 0.50$$

$$G(\text{Mild}) = G(M) = 1 - (4/6)^2 - (2/6)^2 = 0.444$$

$$G(\text{Cool}) = G(C) = 1 - (3/4)^2 - (1/4)^2 = 0.375$$

Total Gini Index of 3 sub-trees = probability for temperature hot * G(H) + probability for temperature mild * G(M) + probability for temperature cool * G(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, the total Gini Index of the three sub-trees = (4/14) G(H) + (6/14) G(M) + (4/14) G(C)
= 0.44028

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 for humidity having high value. 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 for humidity having normal value. 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 Gini Index by using this attribute. We divide the dataset into two subsets according to humidity conditions, i.e., high and normal. Computing Gini Index for each case,

$$G\ (High) = G(H) = 1 - (3/7)^2 - (4/7)^2 = 0.48979$$

$$G\ (Normal) = G(N) = 1 - (6/7)^2 - (1/7)^2 = 0.24489$$

Total Gini Index for the two sub-trees = probability for humidity high* G(H) + probability for humidity normal * G(N)

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

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

Therefore, the total Gini Index for two sub-trees = (7/14) G(H) + (7/14) G(N)

= 0.3673439

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 for windy having value true. Play is held in case of 3 instances (7, 11 and 12) and is not held in case of another 3 instances (2, 6 and 14).

There are 8 instances for windy having value false. 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 Gini Index by using this attribute. We divide the dataset into two subsets according to windy being true or false. Computing Gini Index for each case,

$$G\ (True) = G(T) = 1 - (3/6)^2 - (3/6)^2 = 0.5$$

$$G\ (False) = G(F) = 1 - (6/8)^2 - (2/8)^2 = 0.375$$

Total Gini Index for the two sub-trees = probability for windy true * G(T) + probability for windy false * G(F)

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

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

Therefore, total Gini Index for the two sub-trees = (6/14) G(T) + (8/14) G(F)

= 0.42857

The gain can now be computed as follows:

Potential Split attribute	*Gini Index before split*	*Gini Index after split*	*Gain*
Outlook	*0.45918*	*0.342857*	*0.116323*
Temperature	0.45918	0.44028	0.0189
Humidity	0.45918	0.3673439	0.0918361
Windy	0.45918	0.42857	0.03061

Hence, the largest gain is provided by the attribute 'Outlook' and it is used for the split as depicted in Figure 5.28.

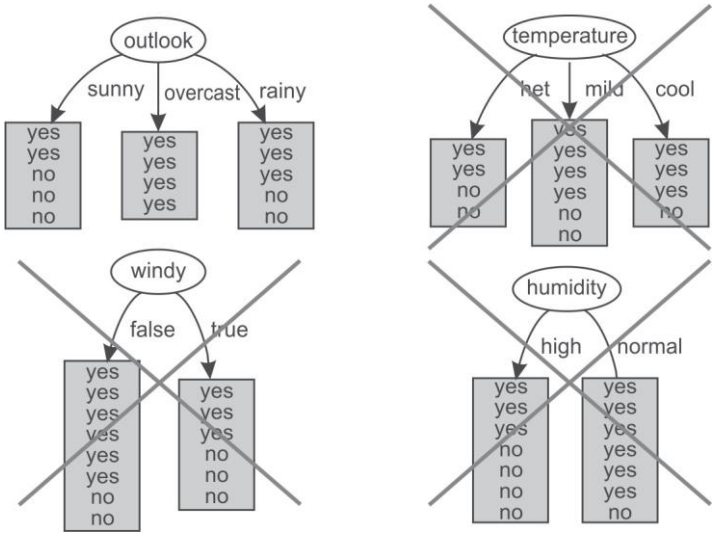

Figure 5.28 Selection of Outlook as root attribute

For Outlook, as there are three possible values, i.e., sunny, overcast and rainy, the dataset will be split into three subsets based on distinct values of the Outlook attribute as shown in Figure 5.29.

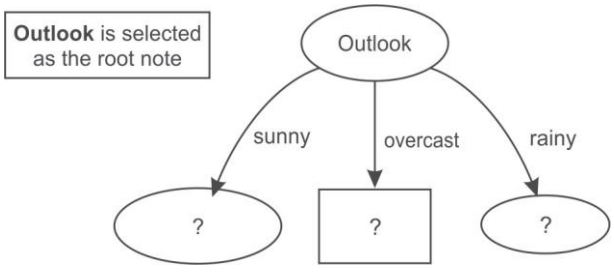

Figure 5.29 Data splitting based on Outlook attribute

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

In this case, we have three input attributes Temperature, Humidity and Windy. This dataset consists of 5 samples. The frequencies of the classes are as follows:

Yes = 2 (Instances 4, 5)

No = 3 (Instances 1,2,3)

The Gini Index of the whole dataset is given by

$$G = 1 - (\text{probability for Play Yes})^{\text{Number of classes}} - (\text{probability for Play No})^{\text{Number of classes}}$$

Here, probability for Play Yes = 2/5

And probability for Play No = 3/5

Therefore, $G = 1 - (2/5)^2 - (3/5)^2 = 0.48$

Let us consider each attribute one by one as split attributes and calculate the Gini Index 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 having value hot. Play is not held in both of 2 instances (1 and 2).

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

There is only 1 instance for temperature having value cool. Play is held in this single instance (4).

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

$$G(\text{Hot}) = G(H) = 1 - (0/2)^2 - (2/2)^2 = 0$$

$$G(\text{Mild}) = G(M) = 1 - (1/2)^2 - (1/2)^2 = 0.5$$

$$G(\text{Cool}) = G(C) = 1 - (1/1)^2 - (0/1)^2 = 0$$

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

Here, probability for temperature hot = (Number of instances for temperature hot/Total number 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 Gini Index for these three sub-trees = (2/5)G(H) + (1/5) G(M) + (2/5) G(C)

$$= 0 + 0.1 + 0 = 0.1$$

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 not held in any of these 3 instances (1, 2 and 3).

There are 2 instances when humidity is normal. Play is held in both of 2 instances (4 and 5).

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

$$G\ (High) = G(H) = 1 - (0/3)^2 - (3/3)^2 = 0$$

$$G\ (Normal) = G(N) = 1 - (2/2)^2 - (0/2)^2 = 0$$

Total Gini Index for the two sub-trees = probability for humidity high * G(H) + probability for humidity normal * G(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 Gini Index for the two sub-trees = (3/5) G(H) + (2/5) G(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 it is windy. Play is held in case of 1 instance (5) and is not held in case of another 1 instance (2).

There are 3 instances when it is not windy. 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 Gini Index by using this attribute. We divide the dataset into two subsets according to windy being true or false. Computing Gini Index for each case,

$$G\ (True) = G\ (T) = 1 - (1/2)^2 - (1/2)^2 = 0.5$$

$$G\ (False) = G\ (F) = 1 - (1/3)^2 - (2/3)^2 = 0.4444$$

Total Gini Index for the two sub-trees = probability for windy true * G(T) + probability for windy false * G(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 Gini Index for the two sub-trees = (2/5) G(T) + (3/5) G(F) = 0.2 + 0.2666 = 0.4666

The Gain can now be computed:

Potential Split attribute	*Gini Index before split*	*Gini Index after split*	*Gain*
Temperature	0.48	0.1	0.38
Humidity	0.48	0	0.48
Windy	0.48	0.4666	0.014

The largest gain is provided by the attribute 'Humidity' and so, it is used for the split. Here, the algorithm is to be tuned in such a way that it should stop when we get the 0 value of the Gini Index

to reduce the calculations for larger datasets. Now, the Humidity attribute is selected as the split attribute as depicted in Figure 5.30.

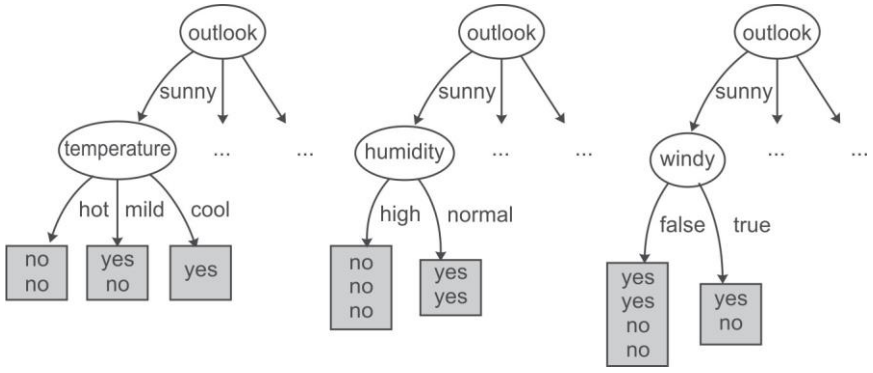

Figure 5.30 Humidity attribute is selected from dataset of Sunny instances

As the dataset consists of two possible values of humidity so data is split in 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

We can clearly see that all records with high humidity are classified as play having 'No' value. On the other hand, all records with normal humidity value are classified as play having 'Yes' value.

Dataset for Humidity 'Normal'

Instance Number	*Temperature*	*Windy*	*Play*
1	cool	false	Yes
2	mild	true	Yes

Thus, the decision tree will look like as shown in Figure 5.31 after analysis of the Humidity attribute.

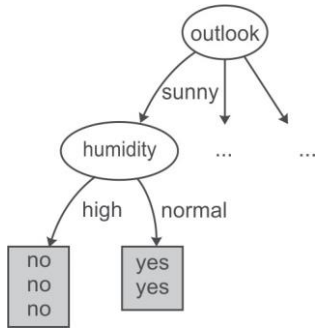

Figure 5.31 Decision tree after spitting data on the Humidity attribute

The analysis of the 'Sunny' dataset is now over and has allowed the the generation of the decision tree shown in Figure 5.31. It has been analyzed that if the outlook is 'Sunny' and humidity is 'Normal' then play will be held while on the other hand if the humidity is 'high' then play will not be held.

Now, let us take next dataset of where Outlook has value 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

This dataset consists of 3 records. For outlook overcast all records belong to the 'Yes' class only. Thus, the decision tree will look like Figure 5.32 after analysis of the overcast dataset.

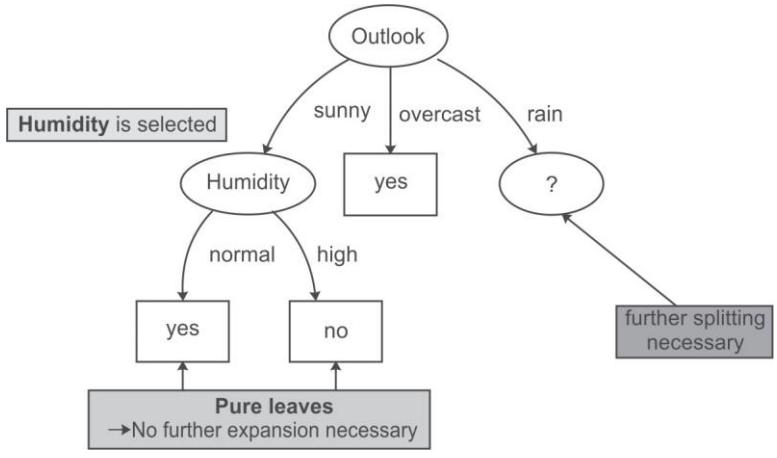

Figure 5.32 Decision tree after analysis of Sunny and Overcast datasets

Therefore, it may be concluded that if the outlook is 'Overcast' then play is held. Now we have to select another split attribute for the outlook rainy and dataset for this is given as follows.

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

This dataset consists of 5 records. The frequencies of the classes are as follows:
Yes= 3 (Instances 1,2 and 4)
No= 2 (Instances 3 and 5)
Gini Index of the whole dataset on the basis of whether play held or not is given by

$$G = 1 - (\text{probability for Play Yes})^{\text{Number of classes}} - (\text{probability for Play No})^{\text{Number of classes}}$$

Here, probability for Play Yes = 3/5
And probability for Play No = 2/5
Therefore, $G = 1 - (3/5)^2 - (2/5)^2 = 0.48$
Let us consider each attribute one by one as split attributes and calculate the Gini Index 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 for temperature having 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 for temperature having value cool. Play is held in case of 1 instance (2) and is not held in case of another 1 instance (3).

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

$$G(\text{Mild}) = G(M) = 1 - (2/3)^2 - (1/3)^2 = 0.444$$

$$G(\text{Cool}) = G(C) = 1 - (1/2)^2 - (1/2)^2 = 0.5$$

Total Gini Index for the two sub-trees = probability for temperature mild * G(M) + probability for temperature cool * G (C)

Here, 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 Gini Index for the two sub-trees = (3/5) G(M) + (2/5) G(C) = 0.2666 + 0.2 = 0.4666

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 for humidity having high value. Play is held in case of 1 instance (1) and is not held in case of 1 instance (5).

There are 3 instances for humidity having normal value. Play is held in case of 2 instances (2 and 4) and is not held in case of 1 instance (3).

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

$$G\ (High) = G(H) = 1 - (1/2)^2 - (1/2)^2 = 0.5$$

$$G\ (Normal) = G(N) = 1 - (2/3)^2 - (1/3)^2 = 0.4444$$

Total Gini Index for the two sub-trees = probability for humidity high* G(H) + probability for humidity normal * G(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 Gini Index for the two sub-trees = (2/5) G(H) + (3/5) G(N) = 0.2 + 0.2666 =0.4666

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 2 instances (3 and 5).

There are 3 instances where windy is false. Play is held in case of all 3 instances (1, 2 and 4).

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

$$G\ (True) = G(T) = 1 - (0/2)^2 - (2/2)^2 = 0$$

$$G\ (False) = G(F) = 1 - (3/3)^2 - (0/3)^2 = 0$$

Total Gini Index for the two sub-trees = probability for windy true * G(T) + probability for windy false * G(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 Gini Index for these two subtrees = (2/5) G(T) + (3/5) G(F) = 0
The Gain can now be computed:

Potential Split attribute	Gini Index before split	Gini Index after split	Gain
Temperature	0.48	0.4666	0.0134
Humidity	0.48	0.4666	0.0134
Windy	0.48	0	0.48

Hence, the largest gain is provided by the attribute 'Windy' and it is used for the split. Now, the Windy attribute is selected as split attribute.

Dataset for Windy 'TRUE'

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

It is clear from the data that whenever it is windy the play is not held. Play is held whenever conditions are otherwise.

Dataset for Windy 'FALSE'

Instance Number	*Temperature*	*Humidity*	*Play*
1	mild	high	Yes
2	cool	normal	Yes
3	mild	normal	Yes

Thus, the decision tree will look like Figure 5.33 after analysis of the 'Rainy' attribute.

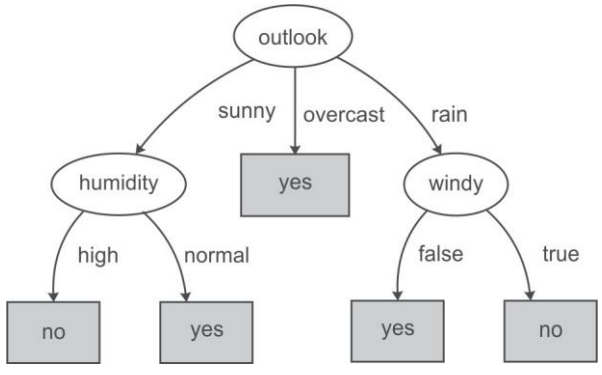

Figure 5.33 Decision tree after analysis of Sunny, Overcast and Rainy datasets

It has been concluded that if the outlook is 'Rainy' and value of windy is 'False' then play will be held and on the other hand, if value of windy is 'True' then the play will not be held.

Figure 5.34 represents the final tree view of all the 14 records of the dataset given in Figure 5.27 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.

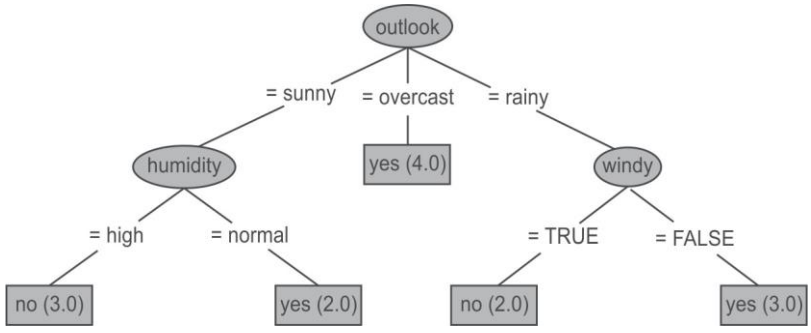

Figure 5.34 Final decision tree after analysis of Sunny, Overcast and Rainy datasets

If we have to predict whether 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 5.34. For the given unknown instance, play will be held as shown in Figure 5.35.

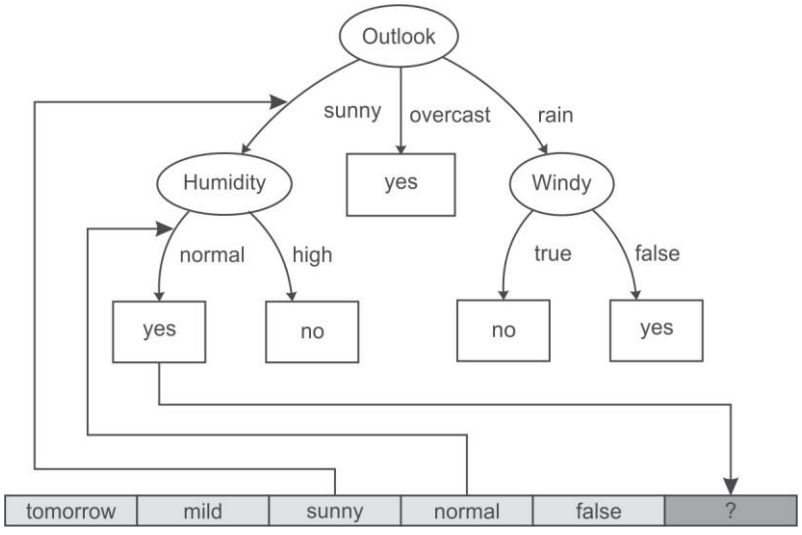

Figure 5.35 Prediction of play for unknown instance

5.1.9 Advantages of the decision tree method

The main advantages of a decision tree classifier are as follows:.

- The rules generated by a decision tree classifier are easy to understand and use.
- Domain knowledge is not required by the decision tree classifier.
- Learning and classification steps of the decision tree are simple and quick.

5.1.10 Disadvantages of the decision tree

The disadvantages of a decision tree classifier can be:

- Decision trees are easy to use compared to other decision-making models, but preparing decision trees, especially large ones with many branches, are complex and time-consuming affairs.
- They are unstable, meaning that a small change in the data can lead to a large change in the structure of the optimal decision tree.
- They are often relatively inaccurate. Many other predictors perform better with similar data.
- Decision trees, while providing easy to view illustrations, can also be unmanageable. Even data that is perfectly divided into classes and uses only simple threshold tests may require a large decision tree. Large trees are not intelligible, and pose presentation difficulties.

The other important classification technique is the Naïve Bayes Method which is based on Bayestheorem. The details of Naïve Bayes method have been discussed in next section.