Continuing with the example in the box, if there are T events with equal probability of occurrence P, then T = 1/P. Claude Shannon, who developed the mathematical underpinnings for information theory (Shannon, 1948), defined entropy as log(1/P) or –log P where P is the probability of an event occurring. If the probability for all events is not identical, we need a weighted expression and thus entropy, H, is adjusted as follows:

$$H = -\sum p_k \log_2 (p_k) \tag{4.1}$$

where k = 1, 2, 3, …, m represent the m classes of the target variable. The p_k represent the proportion of samples that belong to class k. For our gym membership example from earlier, there are two classes: member or nonmember. If our data set had 100 samples with 50% of each, then the entropy of the dataset is given by $H = -[(0.5 \log_2 0.5) + (0.5 \log_2 0.5)] = -\log_2 0.5 = -(-1) = 1$. On the other hand, if we can partition the data into two sets of 50 samples each that contain all members and all nonmembers, the entropy of either of these two partitioned sets is given by $H = -1 \log_2 1 = 0$. Any other proportion of samples within a data set will yield entropy values between 0 and 1.0 (which is the maximum). The Gini index (G) is similar to the entropy measure in its characteristics and is defined as

$$G = \sum (1 - p_k^2) \tag{4.2}$$

The value of G ranges between 0 and a maximum value of 0.5, but otherwise has properties identical to H, and either of these formulations can be used to create partitions in the data (Cover, 1991).

Let us go back to the example of the golf data set introduced earlier, to fully understand the application of entropy concepts for creating a decision tree. This was the same dataset used by J. Ross Quinlan to introduce one of the original decision tree algorithms, the *Iterative Dichotomizer* 3, or ID3 (Quinlan, 1986). The full data is shown in Table 4.1.

There are essentially two questions we need to answer at each step of the tree building process: *where to split the data* and *when to stop splitting*.

Classic Golf Example and How It Is Used to Build a Decision Tree

Where to split data?

There are 14 examples, with four attributes—Temperature, Humidity, Wind, and Outlook. The target attribute that needs to be predicted is Play with two classes: Yes and No. We want to understand how to build a decision tree using this simple data set.

Table 4.1 The Classic Golf Data Set

Outlook	Temperature	Humidity	Windy	Play
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
overcast	83	78	FALSE	yes
rain	70	96	FALSE	yes
rain	68	80	FALSE	yes
rain	65	70	TRUE	no
overcast	64	65	TRUE	yes
sunny	72	95	FALSE	no
sunny	69	70	FALSE	yes
rain	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes
rain	71	80	TRUE	no

Start by partitioning the data on each of the four regular attributes. Let us start with Outlook. There are three categories for this variable: sunny, overcast, and rain. We see that when it is overcast, there are four examples where the outcome was Play = yes for all four cases (see Figure 4.2) and so the proportion of examples in this case is 100% or 1.0. Thus if we split the data set here, the resulting four sample partition will be 100% pure for Play = yes. Mathematically for this partition, the entropy can be calculated using Eq. 4.1 as

$$H_{\text{outlook:overcast}} = -(0/4)\log_2(0/4) - (4/4)\log_2(4/4) = 0.0$$

Similarly, we can calculate the entropy in the other two situations for Outlook:

$$H_{\text{outlook:sunny}} = -(2/5)\log_2(2/5) - (3/5)\log_2(3/5) = 0.971$$

$$H_{\text{outlook:rain}} = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.971$$

For the attribute on the whole, the total "information" is calculated as the weighted sum of these component entropies. There are four instances of Outlook = overcast, thus the proportion for overcast is given by $p_{\text{outlook:overcast}} = 4/14$. The other proportions (for Outlook = sunny and rain) are 5/14 each:

$$I_{\text{outlook}} = p_{\text{outlook:overcast}} * H_{\text{outlook:overcast}} + p_{\text{outlook:sunny}} * H_{\text{outlook:sunny}} + p_{\text{outlook:rain}} * H_{\text{outlook:rain}}$$

Row No.	Play	Outlook
3	yes	overcast
7	yes	overcast
12	yes	overcast
13	yes	overcast
4	yes	rain
5	yes	rain
6	no	rain
10	yes	rain
14	no	rain
1	no	sunny
2	no	sunny
8	no	sunny
9	yes	sunny
11	yes	sunny

FIGURE 4.2
Splitting the data on the Outlook attribute.

$$I_{outlook} = (4/14) * 0 + (5/14) * 0.971 + (5/14) * 0.971 = 0.693$$

Had we *not* partitioned the data along the three values for Outlook, the total information would have been simply the weighted average of the respective entropies for the two classes whose overall proportions were 5/14 (Play = no) and 9/14 (Play = yes):

$$I_{outlook,no\ partition} = -(5/14)\log_2(5/14) - (9/14)\log_2(9/14) = 0.940$$

By creating these splits or partitions, we have reduced some entropy (and thus gained some information). This is called, aptly enough, *information gain*. In the case of Outlook, this is given simply by

$$I_{outlook,\ no\ partition} - I_{outlook} = 0.940 - 0.693 = 0.247$$

We can now compute similar information gain values for the other three attributes, as shown in Table 4.2.

Table 4.2 Computing the Information Gain for All Attributes

Attribute	Information Gain
Temperature	0.029
Humidity	0.102
Wind	0.048
Outlook	0.247

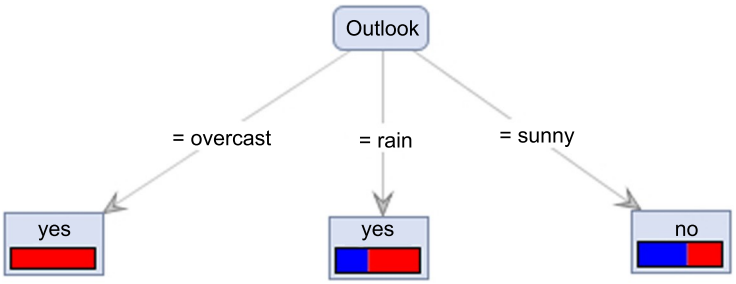

FIGURE 4.3
Splitting the golf data on the Outlook attribute yields three subsets or branches. The middle and right branches may be split further.

For numeric variables, possible split points to examine are essentially averages of available values. For example, the first potential split point for Humidity could be Average [65,70], which is 67.5, the next potential split point could be Average [70,75], which is 72.5, and so on. We use similar logic for the other numeric attribute, Temperature. The algorithm computes the information gain at each of these potential split points and chooses the one which maximizes it. Another way to approach this would be to discretize the numerical ranges, for example, Temperature >=80 could be considered "Hot," between 70 to 79 "Mild," and less than 70 "Cool."

From Table 4.2, it is clear that if we partition the data set into three sets along the three values of Outlook, we will experience the largest information gain. This gives the first node of the decision tree as shown in Figure 4.3. As noted earlier, the terminal node for the Outlook = overcast branch consists of four samples, all of which belong to the class Play = yes. The other two branches contain a mix of classes. The Outlook = rain branch has three yes results and the Outlook = sunny branch has three no results.

Thus not all the final partitions are 100% homogenous. This means that we could apply the same process for each of these subsets till we get "purer" results. So we revert back to the first question once again—where to split the data? Fortunately this was already answered for us when we computed the