

CHAPTER TEN: DECISION TREES

CONTEXT AND PERSPECTIVE

Richard works for a large online retailer. His company is launching a next-generation eReader soon, and they want to maximize the effectiveness of their marketing. They have many customers, some of whom purchased one of the company's previous generation digital readers. Richard has noticed that certain types of people were the most anxious to get the previous generation device, while other folks seemed to content to wait to buy the electronic gadget later. He's wondering what makes some people motivated to buy something as soon as it comes out, while others are less driven to have the product.

Richard's employer helps to drive the sales of its new eReader by offering specific products and services for the eReader through its massive web site—for example, eReader owners can use the company's web site to buy digital magazines, newspapers, books, music, and so forth. The company also sells thousands of other types of media, such as traditional printed books and electronics of every kind. Richard believes that by mining the customers' data regarding general consumer behaviors on the web site, he'll be able to figure out which customers will buy the new eReader early, which ones will buy next, and which ones will buy later on. He hopes that by predicting when a customer will be ready to buy the next-gen eReader, he'll be able to time his target marketing to the people most ready to respond to advertisements and promotions.

LEARNING OBJECTIVES

After completing the reading and exercises in this chapter, you should be able to:

- Explain what decision trees are, how they are used and the benefits of using them.
- Recognize the necessary format for data in order to perform predictive decision tree mining.

- Develop a decision tree data mining model in RapidMiner using a training data set.
- Interpret the visual tree's nodes and leaves, and apply them to a scoring data set in order to deploy the model.
- Use different tree algorithms in order to increase the granularity of the tree's detail.

ORGANIZATIONAL UNDERSTANDING

Richard wants to be able to predict the timing of buying behaviors, but he also wants to understand how his customers' behaviors on his company's web site indicate the timing of their purchase of the new eReader. Richard has studied the classic diffusion theories that noted scholar and sociologist Everett Rogers first published in the 1960s. Rogers surmised that the adoption of a new technology or innovation tends to follow an 'S' shaped curve, with a smaller group of the most enterprising and innovative customers adopting the technology first, followed by larger groups of middle majority adopters, followed by smaller groups of late adopters (Figure 10-1).

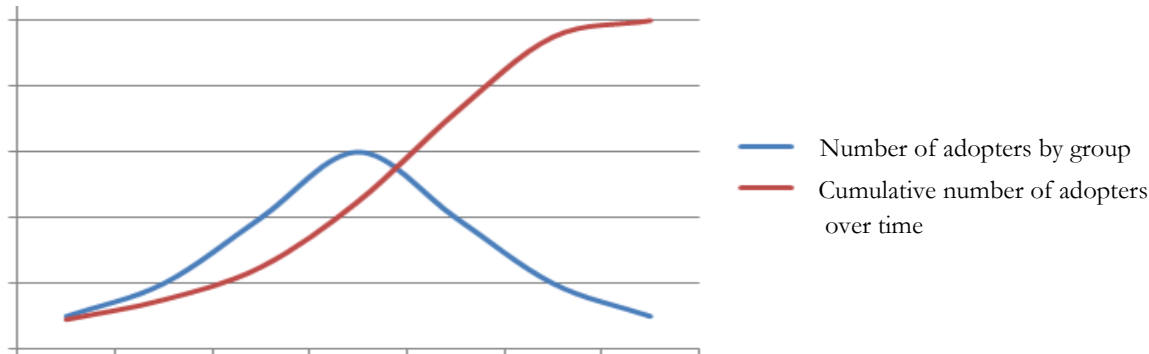


Figure 10-1. Everett Rogers' theory of adoption of new innovations.

Those at the front of the blue curve are the smaller group that are first to want and buy the technology. Most of us, the masses, fall within the middle 70-80% of people who eventually acquire the technology. The low end tail on the right side of the blue curve are the laggards, the ones who eventually adopt. Consider how DVD players and cell phones have followed this curve.

Understanding Rogers' theory, Richard believes that he can categorize his company's customers into one of four groups that will eventually buy the new eReader: Innovators, Early Adopters, Early Majority or Late Majority. These groups track with Rogers' social adoption theories on the diffusion of technological innovations, and also with Richard's informal observations about the speed of adoption of his company's previous generation product. He hopes that by watching the

customers' activity on the company's web site, he can anticipate approximately when each person will be most likely to buy an eReader. He feels like data mining can help him figure out which activities are the best predictors of which category a customer will fall into. Knowing this, he can time his marketing to each customer to coincide with their likelihood of buying.

DATA UNDERSTANDING

Richard has engaged us to help him with his project. We have decided to use a **decision tree** model in order to find good early predictors of buying behavior. Because Richard's company does all of its business through its web site, there is a rich data set of information for each customer, including items they have just browsed for, and those they have actually purchased. He has prepared two data sets for us to use. The training data set contains the web site activities of customers who bought the company's previous generation reader, and the timing with which they bought their reader. The second is comprised of attributes of current customers which Richard hopes will buy the new eReader. He hopes to figure out which category of adopter each person in the scoring data set will fall into based on the profiles and buying timing of those people in the training data set.

In analyzing his data set, Richard has found that customers' activity in the areas of digital media and books, and their general activity with electronics for sale on his company's site, seem to have a lot in common with when a person buys an eReader. With this in mind, we have worked with Richard to compile data sets comprised of the following attributes:

- **User_ID:** A numeric, unique identifier assigned to each person who has an account on the company's web site.
- **Gender:** The customer's gender, as identified in their customer account. In this data set, it is recorded a 'M' for male and 'F' for Female. The Decision Tree operator can handle non-numeric data types.
- **Age:** The person's age at the time the data were extracted from the web site's database. This is calculated to the nearest year by taking the difference between the system date and the person's birthdate as recorded in their account.
- **Marital_Status:** The person's marital status as recorded in their account. People who indicated on their account that they are married are entered in the data set as 'M'. Since the

web site does not distinguish single types of people, those who are divorced or widowed are included with those who have never been married (indicated in the data set as ‘S’).

- **Website_Activity:** This attribute is an indication of how active each customer is on the company’s web site. Working with Richard, we used the web site database’s information which records the duration of each customers visits to the web site to calculate how frequently, and for how long each time, the customers use the web site. This is then translated into one of three categories: Seldom, Regular, or Frequent.
- **Browsed_Electronics_12Mo:** This is simply a Yes/No column indicating whether or not the person browsed for electronic products on the company’s web site in the past year.
- **Bought_Electronics_12Mo:** Another Yes/No column indicating whether or not they purchased an electronic item through Richard’s company’s web site in the past year.
- **Bought_Digital_Media_18Mo:** This attribute is a Yes/No field indicating whether or not the person has purchased some form of digital media (such as MP3 music) in the past year and a half. This attribute does not include digital book purchases.
- **Bought_Digital_Books:** Richard believes that as an indicator of buying behavior relative to the company’s new eReader, this attribute will likely be the best indicator. Thus, this attribute has been set apart from the purchase of other types of digital media. Further, this attribute indicates whether or not the customer has *ever* bought a digital book, not just in the past year or so.
- **Payment_Method:** This attribute indicates how the person pays for their purchases. In cases where the person has paid in more than one way, the mode, or most frequent method of payment is used. There are four options:
 - Bank Transfer—payment via e-check or other form of wire transfer directly from the bank to the company.
 - Website Account—the customer has set up a credit card or permanent electronic funds transfer on their account so that purchases are directly charged through their account at the time of purchase.
 - Credit Card—the person enters a credit card number and authorization each time they purchase something through the site.
 - Monthly Billing—the person makes purchases periodically and receives a paper or electronic bill which they pay later either by mailing a check or through the company web site’s payment system.

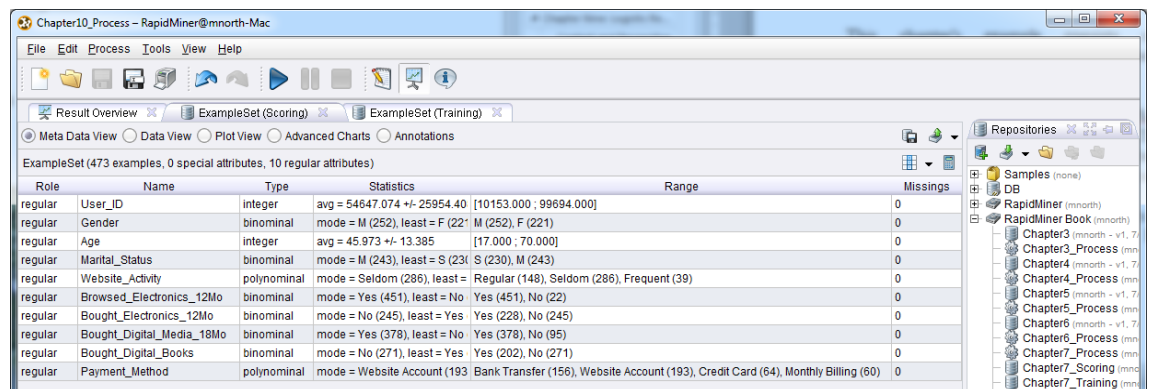
- **eReader_Adoption:** This attribute exists only in the training data set. It consists of data for customers who purchased the previous-gen eReader. Those who purchased within a week of the product's release are recorded in this attribute as 'Innovator'. Those who purchased after the first week but within the second or third weeks are entered as 'Early Adopter'. Those who purchased after three weeks but within the first two months are 'Early Majority'. Those who purchased after the first two months are 'Late Majority'. This attribute will serve as our label when we apply our training data to our scoring data.

With Richard's data and an understanding of what it means, we can now proceed to...

DATA PREPARATION

This chapter's example consists of two data sets: Chapter10DataSet_Training.csv and Chapter10DataSet_Scoring.csv. Download these from the companion web site now, then complete the following steps:

- 1) Import both data sets into your RapidMiner repository. You do not need to worry about attribute data types because the Decision Tree operator can handle all types of data. Be sure that you do designate the first row of each of the data sets as the attribute names as you import. Save them in the repository with descriptive names, so that you will be able to tell what they are.
- 2) Drag and drop both of the data sets into a new main process window. Rename the Retrieve objects as Training and Scoring respectively. Run your model to examine the data and familiarize yourself with the attributes.



Role	Name	Type	Statistics	Range	Missings
regular	User_ID	integer	avg = 54647.074 +/- 25954.40	[10153.000 ; 99694.000]	0
regular	Gender	binominal	mode = M (252), least = F (22)	M (252), F (221)	0
regular	Age	integer	avg = 45.973 +/- 13.385	[17.000 ; 70.000]	0
regular	Marital_Status	binominal	mode = M (243), least = S (23)	S (230), M (243)	0
regular	Website_Activity	polynomial	mode = Seldom (286), least = Regular (148)	Seldom (286), Frequent (39)	0
regular	Browsed_Electronics_12Mo	binominal	mode = Yes (451), least = No	Yes (451), No (22)	0
regular	Bought_Electronics_12Mo	binominal	mode = No (245), least = Yes	Yes (228), No (245)	0
regular	Bought_Digital_Media_18Mo	binominal	mode = Yes (378), least = No	Yes (378), No (95)	0
regular	Bought_Digital_Books	binominal	mode = No (271), least = Yes	Yes (202), No (271)	0
regular	Payment_Method	polynomial	mode = Website Account (193)	Bank Transfer (156), Website Account (193), Credit Card (64), Monthly Billing (60)	0

Figure 10-2. Meta data for the scoring data set.

- 3) Switch back to design perspective. While there are no missing or apparently inconsistent values in the data set, there is still some data preparation yet to do. First of all, the User_ID is an arbitrarily assigned value for each customer. The customer doesn't use this value for anything, it is simply a way to uniquely identify each customer in the data set. It is not something that relates to each person in any way that would correlate to, or be predictive of, their buying and technology adoption tendencies. As such, it should not be included in the model as an independent variable.

We can handle this attribute in one of two ways. First, we can remove the attribute using a Select Attributes operator, as was demonstrated back in Chapter 3. Alternatively, we can try a new way of handling a non-predictive attribute. This is accomplished using the Set Role operator. Using the search field in the Operators tab, find and add Set Role operators to both your training and scoring streams. In the Parameters area on the right hand side of the screen, set the role of the User_ID attribute to 'id'. This will leave the attribute in the data set throughout the model, but it won't consider the attribute as a predictor for the label attribute. Be sure to do this for both the training and scoring data sets, since the User_ID attribute is found in both of them (Figure 10-3).

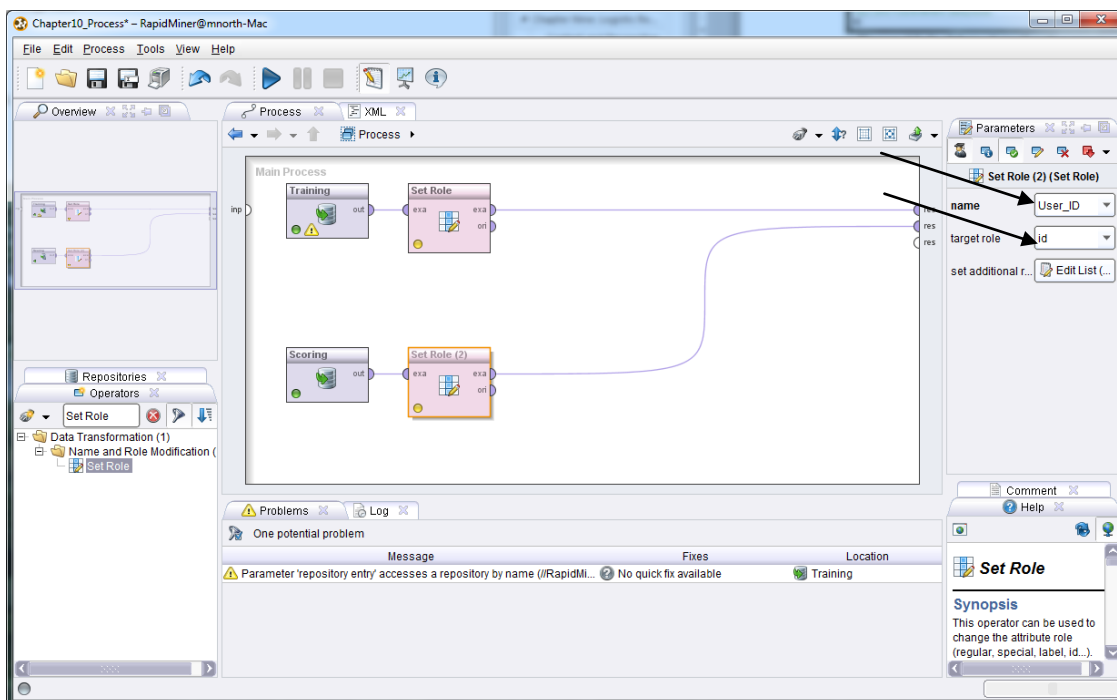


Figure 10-3. Setting the User_ID attribute to an 'id' role, so it won't be considered in the predictive model.

- 4) One of the nice side-effects of setting an attribute's role to 'id' rather than removing it using a Select Attributes operator is that it makes each record easier to match back to individual people later, when viewing predictions in results perspective. Thinking back to some of our other predictive models in previous chapters (e.g. Discriminant Analysis), you could use such an approach to leave in peoples' names or ID numbers so that you could easily know who to contact during the deployment phase of data mining projects.

Before adding a Decision Tree operator, we still need to do another data preparation step. The Decision Tree operator, as with other predictive model operators we've used to this point in the text, expects the training stream to supply a 'label' attribute. For this example, we want to predict which adopter group Richard's next-gen eReader customers are likely to be in. So our label will be eReader_Adoption (Figure 10-4).

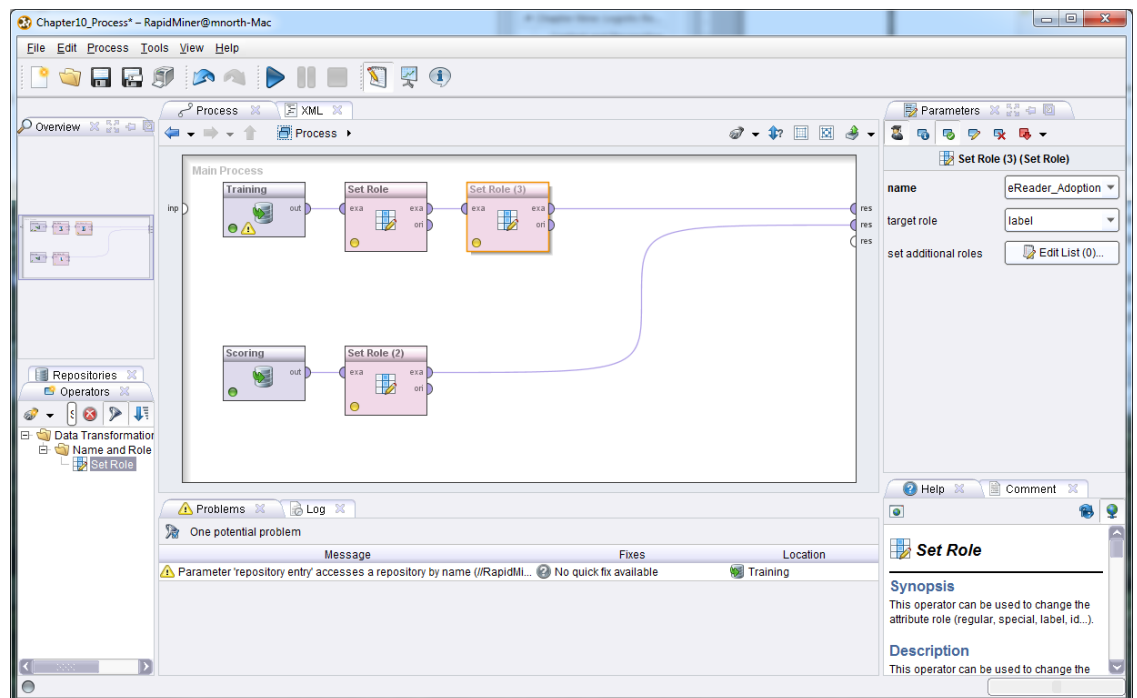


Figure 10-4. Setting the eReader_Adoption attribute as the label in our training stream.

- 5) Next, search in the Operators tab for 'Decision Tree'. Select the basic Decision Tree operator and add it to your training stream as it is in Figure 10-5.

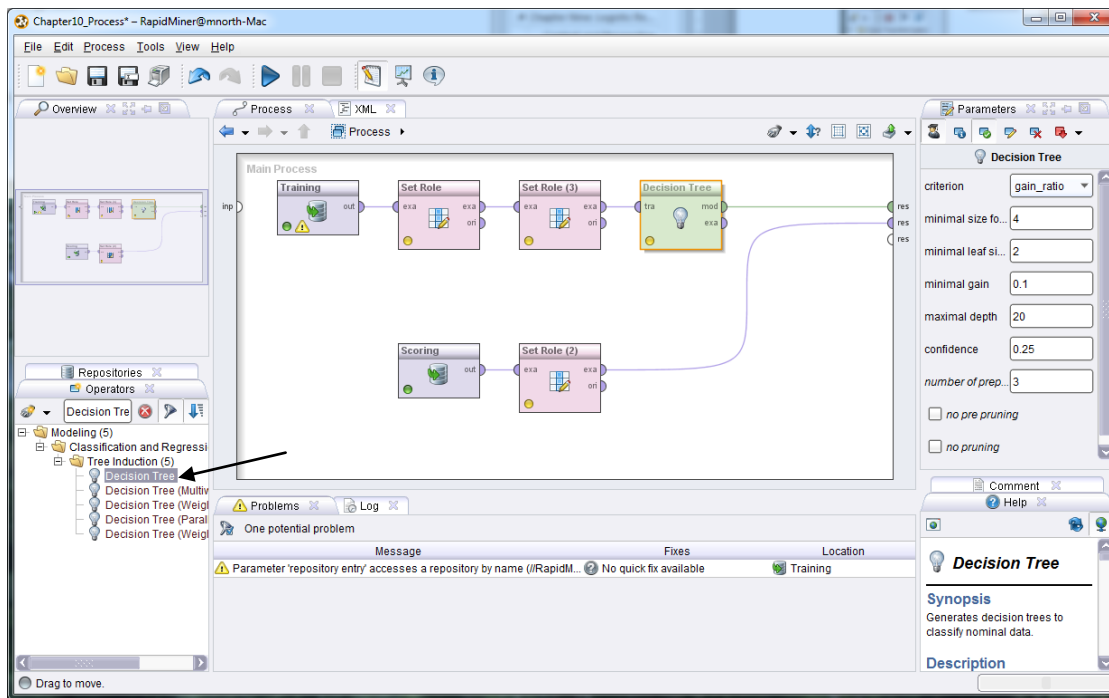


Figure 10-5. The Decision Tree operator added to our model.

- 6) Run the model and switch to the Tree (Decision Tree) tab in results perspective. You will see our preliminary tree (Figure 10-6).

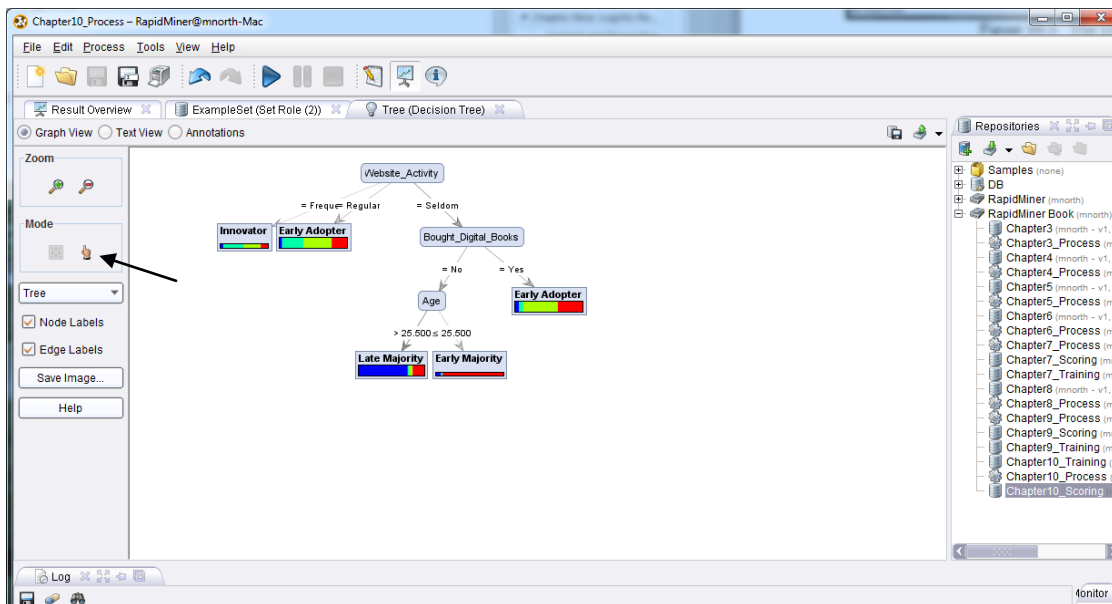


Figure 10-6. Decision tree results.

- 7) In Figure 10-6, we can see what are referred to as **nodes** and **leaves**. The nodes are the gray oval shapes. They are attributes which serve as good predictors for our label attribute. The leaves are the multicolored end points that show us the distribution of categories from

our label attribute that follow the branch of the tree to the point of that leaf. We can see in this tree that Website_Activity is our best predictor of whether or not a customer is going to adopt (buy) the company's new eReader. If the person's activity is frequent or regular, we see that they are likely to be an Innovator or Early Adopter, respectively. If however, they seldom use the web site, then whether or not they've bought digital books becomes the next best predictor of their eReader adoption category. If they have not bought digital books through the web site in the past, Age is another predictive attribute which forms a node, with younger folks adopting sooner than older ones. This is seen on the branches for the two leaves coming from the Age node in Figure 10-6. Those who seldom use the company's website, have never bought digital books on the site, and are older than 25 ½ are most likely to land in the Late Majority category, while those with the same profile but are under 25 ½ are bumped to the Early Majority prediction. In this example you can see how you read the nodes, leaves and branch labels as you move down through the tree.

Before returning to design perspective, take a minute to try some of the tools on the left hand side of the screen. The magnifying glasses can help you see your tree better, spreading out or compacting the nodes and leaves to enhance readability or to view more of a large tree at one time. Also, try using the 'hand' icon under Mode (see the arrow on Figure 10-6). This allows you to click and hold on individual leaves or nodes and drag them around to enhance your tree's readability. Finally, try hovering your mouse over one of the leaves in the tree. In Figure 10-7, we see a tool-tip hover box showing details of this leaf. Although our training data is going to predict that 'regular' web site users are going to be Early Adopters, the model is not 100% based on that prediction. In the hover, we read that in the training data set, 9 people who fit this profile are Late Adopters, 58 are Innovators, 75 are Early Adopters and 41 are Early Majority. When we get to Evaluation phase, we will see that this uncertainty in our data will translate into confidence percentages, similar to what we saw in Chapter 9 with logistic regression.

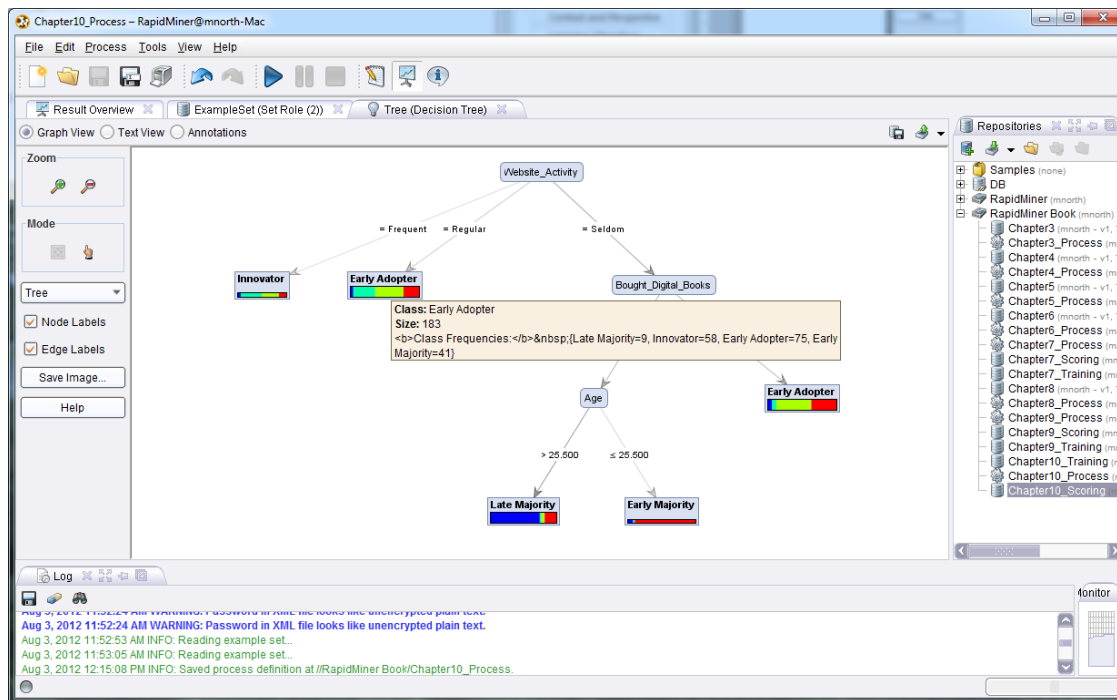


Figure 10-7. A tool-tip hover showing expanded leaf detail in our tree.

With our predictor attributes prepared, we are now ready to move on to...

MODELING

- 8) Return to design perspective. In the Operators tab search for and add an Apply Model operator, bringing your training and scoring streams together. Ensure that both the *lab* and *mod* ports are connected to *res* ports in order to generate our desired outputs (Figure 10-8).

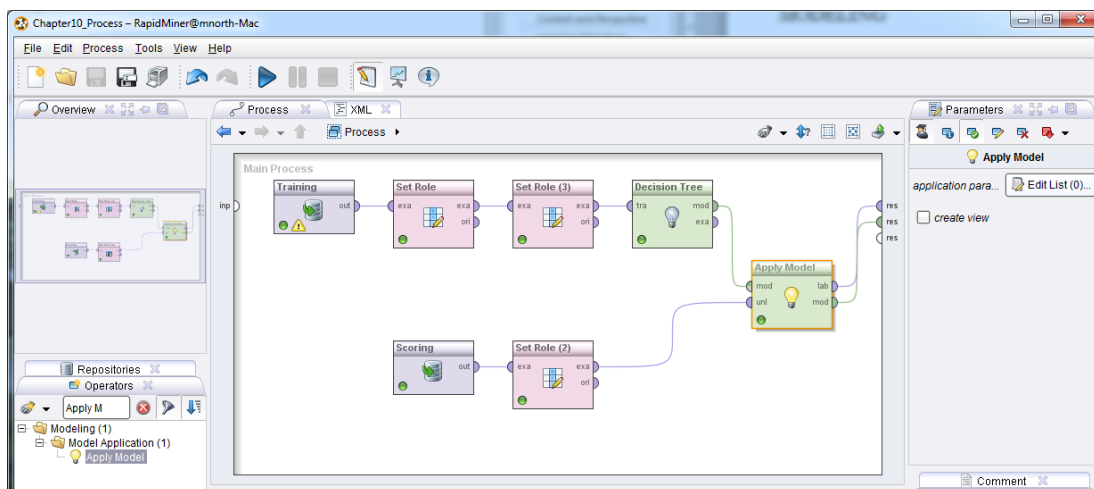


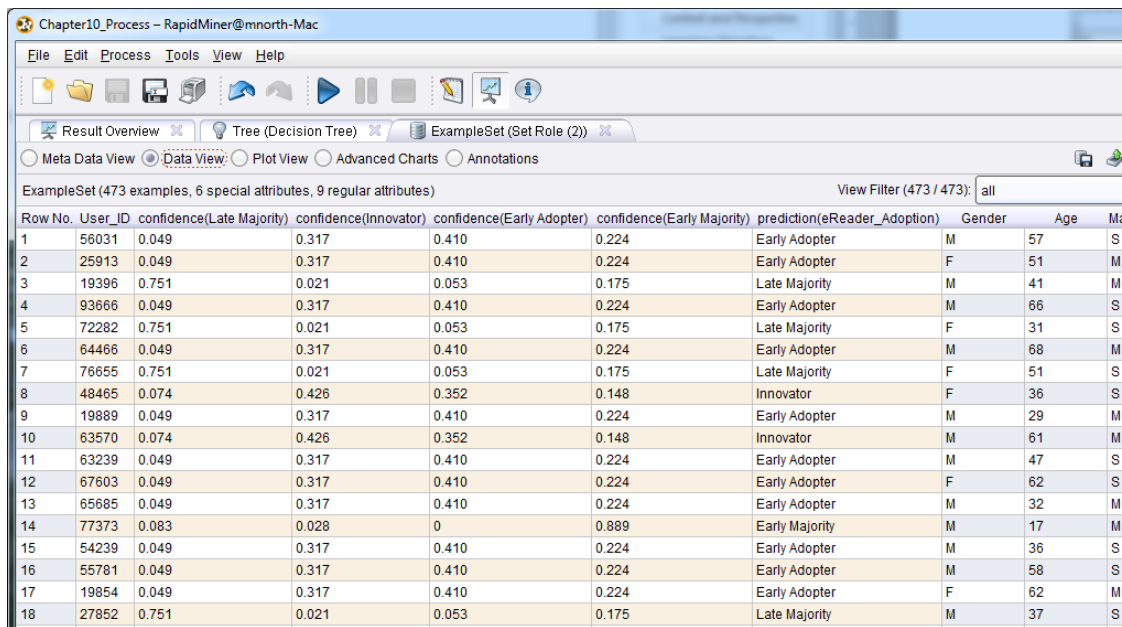
Figure 10-8. Applying the model to our scoring data, and outputting label predictions (*lab*) and a decision tree model (*mod*).

- 9) Run the model. You will see familiar results—the tree remains the same as it was in Figure 10-6, for now. Click on the ExampleSet tab next to the Tree tab. Our tree has been applied to our scoring data. As was the case with logistic regression, confidence attributes have been created by RapidMiner, along with a prediction attribute.

Role	Name	Type	Statistics	Range	Missings
	id	integer	avg = 54647.074 +/- 25954.408	[10153.000 ; 99694.000]	0
confidence_Late Majority	confidence(Late Majority)	real	avg = 0.262 +/- 0.313	[0.049 ; 0.751]	0
confidence_Innovator	confidence(Innovator)	real	avg = 0.158 +/- 0.150	[0.021 ; 0.426]	0
confidence_Early Adopter	confidence(Early Adopter)	real	avg = 0.314 +/- 0.192	[0.000 ; 0.508]	0
confidence_Early Majority	confidence(Early Majority)	real	avg = 0.266 +/- 0.144	[0.148 ; 0.889]	0
prediction	prediction(eReader_Adoption)	polynomial	mode = Early Adopter (280), least = Early Majority (17)	Late Majority (137), Innovator (39), Early (0)	0
regular	Gender	binominal	mode = M (252), least = F (221)	M (252), F (221)	0
regular	Age	integer	avg = 45.973 +/- 13.385	[17.000 ; 70.000]	0
regular	Marital_Status	binominal	mode = M (243), least = S (230)	S (230), M (243)	0
regular	Website_Activity	polynomial	mode = Seldom (286), least = Frequent (39)	Regular (148), Seldom (286), Frequent (0)	0
regular	Browsed_Electronics_12Mo	binominal	mode = Yes (451), least = No (22)	Yes (451), No (22)	0
regular	Bought_Electronics_12Mo	binominal	mode = No (245), least = Yes (228)	Yes (228), No (245)	0
regular	Bought_Digital_Media_18Mo	binominal	mode = Yes (378), least = No (95)	Yes (378), No (95)	0
regular	Bought_Digital_Books	binominal	mode = No (271), least = Yes (202)	Yes (202), No (271)	0
regular	Payment_Method	polynomial	mode = Website Account (193), least = Monthly Billing (60)	Bank Transfer (156), Website Account (1)	0

Figure 10-9. Meta data for scoring data set predictions.

- 10) Switch to Data View using the radio button. We see in Figure 10-10 the prediction for each customer's adoption group, along with confidence percentages for each prediction. Unlike the logistic regression example in the previous chapter, there are four confidence attributes, corresponding to the four possible values in the label (eReader_Adoption). We interpret these the same way that we did with the other models though—the percentages add to 100%, and the prediction is whichever category yielded the highest confidence percentage. RapidMiner is very (but not 100%) convinced that person 77373 (Row 14, Figure 10-10) is going to be a member of the early majority (88.9%). Despite some uncertainty, RapidMiner is completely sure that this person is *not* going to be an early adopter (0%).



Chapter10_Process - RapidMiner@mnorth-Mac

File Edit Process Tools View Help

Result Overview Tree (Decision Tree) ExampleSet (Set Role (2))

Meta Data View Data View Plot View Advanced Charts Annotations

ExampleSet (473 examples, 6 special attributes, 9 regular attributes) View Filter (473 / 473): all

Row No.	User_ID	confidence(Late Majority)	confidence(Innovator)	confidence(Early Adopter)	confidence(Early Majority)	prediction(eReader_Adoption)	Gender	Age	M:
1	56031	0.049	0.317	0.410	0.224	Early Adopter	M	57	S
2	25913	0.049	0.317	0.410	0.224	Early Adopter	F	51	M
3	19396	0.751	0.021	0.053	0.175	Late Majority	M	41	M
4	93666	0.049	0.317	0.410	0.224	Early Adopter	M	66	S
5	72282	0.751	0.021	0.053	0.175	Late Majority	F	31	S
6	64466	0.049	0.317	0.410	0.224	Early Adopter	M	68	M
7	76655	0.751	0.021	0.053	0.175	Late Majority	F	51	S
8	48465	0.074	0.426	0.352	0.148	Innovator	F	36	S
9	19889	0.049	0.317	0.410	0.224	Early Adopter	M	29	M
10	63570	0.074	0.426	0.352	0.148	Innovator	M	61	M
11	63239	0.049	0.317	0.410	0.224	Early Adopter	M	47	S
12	67603	0.049	0.317	0.410	0.224	Early Adopter	F	62	S
13	65685	0.049	0.317	0.410	0.224	Early Adopter	M	32	M
14	77373	0.083	0.028	0	0.889	Early Majority	M	17	M
15	54239	0.049	0.317	0.410	0.224	Early Adopter	M	36	S
16	55781	0.049	0.317	0.410	0.224	Early Adopter	M	58	S
17	19854	0.049	0.317	0.410	0.224	Early Adopter	F	62	M
18	27852	0.751	0.021	0.053	0.175	Late Majority	M	37	S

Figure 10-10. Predictions and their associated confidence percentages using our decision tree.

- 11) We've already begun to evaluate our model's results, but what if we feel like we'd like to see greater detail, or granularity in our model. Surely some of our other attributes are also predictive in nature. Remember that CRISP-DM is cyclical in nature, and that in some modeling techniques, especially those with less structured data, some back and forth trial-and-error can reveal more interesting patterns in data. Switch back to design perspective, click on the Decision Tree operator, and in the Parameters area, change the 'criterion' parameter to 'gini_index', as shown in Figure 10-11.

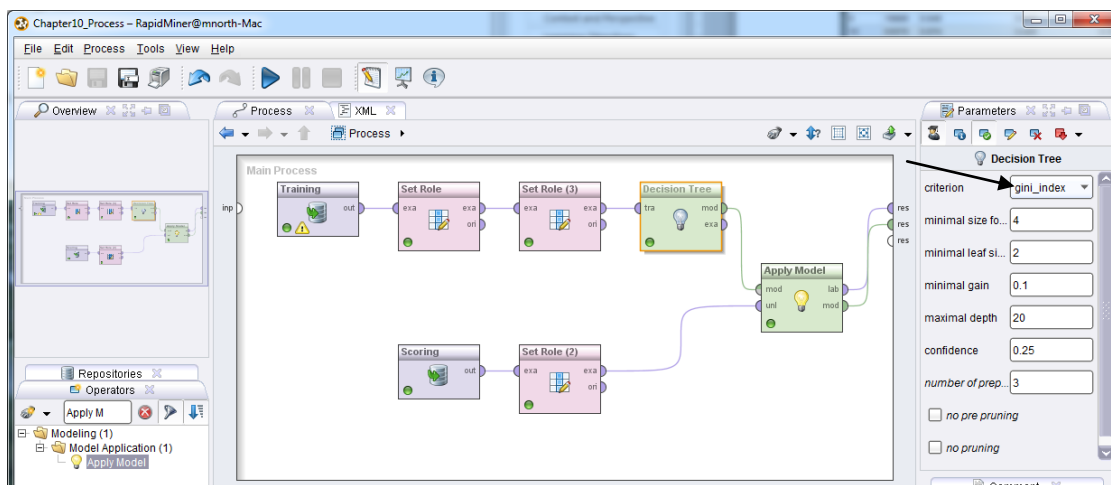


Figure 10-11. Constructing our decision tree model using the **gini_index** algorithm rather than the **gain_ratio** algorithm.

Now, re-run the model and we will move on to...

EVALUATION

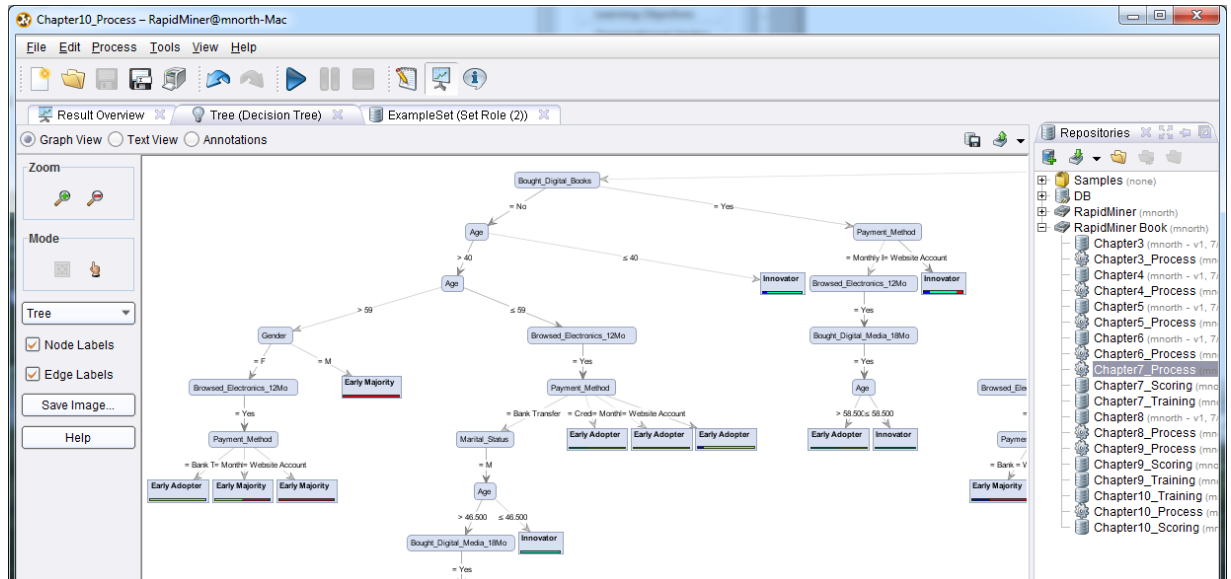
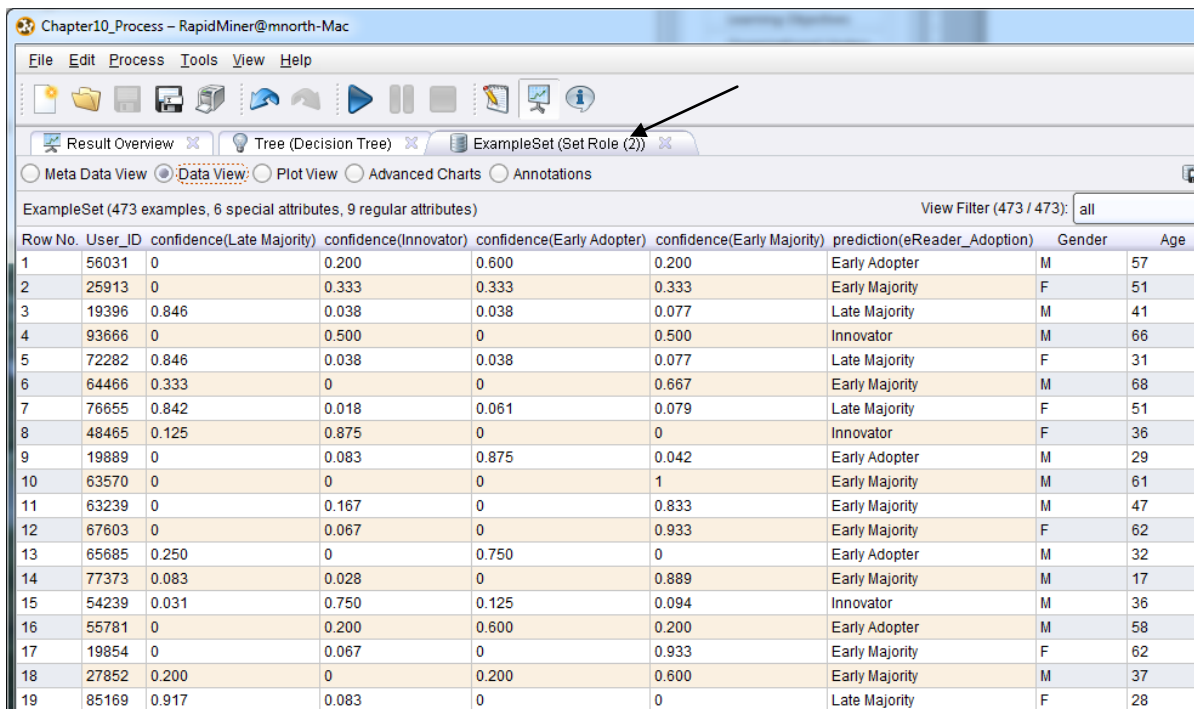


Figure 10-12. Tree resulting from a gini_index algorithm.

We see in this tree that there is much more detail, more granularity in using the Gini algorithm as our parameter for our decision tree. We could further modify the tree by going back to design view and changing the minimum number of items to form a node (size for split) or the minimum size for a leaf. Even accepting the defaults for those parameters though, we can see that the Gini algorithm alone is much more sensitive than is the Gain Ratio algorithm in identifying nodes and leaves. Take a minute to explore around this new tree model. You will find that it is extensive, and that you will to use both the Zoom and Mode tools to see it all. You should find that most of our other independent variables (predictor attributes) are now being used, and the granularity with which Richard can identify each customer's likely adoption category is much greater. How active the person is on Richard's employer's web site is still the single best predictor, but gender, and multiple levels of age have now also come into play. You will also find that a single attribute is sometimes used more than once in a single branch of the tree. Decision trees are a lot of fun to experiment with, and with a sensitive algorithm like Gini generating them, they can be tremendously interesting as well.

Switch to the ExampleSet tab in Data View. We see here (Figure 10-13) that changing our tree's underlying algorithm has, in some cases, also changed our confidence in the prediction.



Chapter10_Process – RapidMiner@mnorth-Mac

File Edit Process Tools View Help

Result Overview Tree (Decision Tree) ExampleSet (Set Role (2))

Meta Data View Data View Plot View Advanced Charts Annotations

ExampleSet (473 examples, 6 special attributes, 9 regular attributes) View Filter (473 / 473): all

Row No.	User_ID	confidence(Late Majority)	confidence(Innovator)	confidence(Early Adopter)	confidence(Early Majority)	prediction(eReader_Adoption)	Gender	Age
1	56031	0	0.200	0.600	0.200	Early Adopter	M	57
2	25913	0	0.333	0.333	0.333	Early Majority	F	51
3	19396	0.846	0.038	0.038	0.077	Late Majority	M	41
4	93666	0	0.500	0	0.500	Innovator	M	66
5	72282	0.846	0.038	0.038	0.077	Late Majority	F	31
6	64466	0.333	0	0	0.667	Early Majority	M	68
7	76655	0.842	0.018	0.061	0.079	Late Majority	F	51
8	48465	0.125	0.875	0	0	Innovator	F	36
9	19889	0	0.083	0.875	0.042	Early Adopter	M	29
10	63570	0	0	0	1	Early Majority	M	61
11	63239	0	0.167	0	0.833	Early Majority	M	47
12	67603	0	0.067	0	0.933	Early Majority	F	62
13	65685	0.250	0	0.750	0	Early Adopter	M	32
14	77373	0.083	0.028	0	0.889	Early Majority	M	17
15	54239	0.031	0.750	0.125	0.094	Innovator	M	36
16	55781	0	0.200	0.600	0.200	Early Adopter	M	58
17	19854	0	0.067	0	0.933	Early Majority	F	62
18	27852	0.200	0	0.200	0.600	Early Majority	M	37
19	85169	0.917	0.083	0	0	Late Majority	F	28

Figure 10-13. New predictions and confidence percentages using Gini.

Let's take the person on Row 1 (ID 56031) as an example. In Figure 10-10, this person was calculated as having at least some percentage chance of landing in any one of the four adopter categories. Under the Gain Ratio algorithm, we were 41% sure he'd be an early adopter, but almost 32% sure he might also turn out to be an innovator. In other words, we feel confident he'll buy the eReader early on, but we're not sure how early. Maybe that matters to Richard, maybe not. He'll have to decide during the deployment phase. But perhaps using Gini, we can help him decide. In Figure 10-13, this same man is now shown to have a 60% chance of being an early adopter and only a 20% chance of being an innovator. The odds of him becoming part of the late majority crowd under the Gini model have dropped to zero. We know he will adopt (or at least we are *predicting* with 100% confidence that he will adopt), and that he will adopt early. While he may not be at the top of Richard's list when deployment rolls around, he'll probably be higher than he otherwise would have been under *gain_ratio*. Note that while Gini has changed some of our predictions, it hasn't affected all of them. Re-check person ID 77373 briefly. There is no difference in this person's predictions under either algorithm—RapidMiner is quite certain in its predictions for this young man. Sometimes the level of confidence in a prediction through a

decision tree is so high that a more sensitive underlying algorithm won't alter an observation's prediction values at all.

DEPLOYMENT

Richard's original desire was to be able to figure out which customers he could expect to buy the new eReader and on what time schedule, based on the company's last release of a high-profile digital reader. The decision tree has enabled him to predict that and to determine how reliable the predictions are. He's also been able to determine which attributes are the most predictive of eReader adoption, and to find greater granularity in his model by using `gini_index` as his tree's underlying algorithm.

But how will he use this new found knowledge? The simplest and most direct answer is that he now has a list of customers and their probable adoption timings for the next-gen eReader. These customers are identifiable by the `User_ID` that was retained in the results perspective data but not used as a predictor in the model. He can segment these customers and begin a process of target marketing that is timely and relevant to each individual. Those who are most likely to purchase immediately (predicted innovators) can be contacted and encouraged to go ahead and buy as soon as the new product comes out. They may even want the option to pre-order the new device. Those who are less likely (predicted early majority) might need some persuasion, perhaps a free digital book or two with eReader purchase or a discount on digital music playable on the new eReader. The least likely (predicted late majority), can be marketed to passively, or perhaps not at all if marketing budgets are tight and those dollars need to be spent incentivizing the most likely customers to buy. On the other hand, perhaps very little marketing is needed to the predicted innovators, since they are predicted to be the most likely to buy the eReader in the first place.

Further though, Richard now has a tree that shows him which attributes matter most in determining the likelihood of buying for each group. New marketing campaigns can use this information to focus more on increasing web site activity level, or on connecting general electronics that are for sale on the company's web site with the eReaders and digital media more specifically. These types of cross-categorical promotions can be further honed to appeal to buyers of a specific gender or in a given age range. Richard has much that he can use in this rich data mining output as he works to promote the next-gen eReader.

CHAPTER SUMMARY

Decision trees are excellent predictive models when the target attribute is categorical in nature, and when the data set is of mixed types. Although this chapter's data sets did not contain any examples, decision trees are better than more statistics-based approaches at handling attributes that have missing or inconsistent values that are not handled—decision trees will work around such data and still generate usable results.

Decision trees are made of nodes and leaves (connected by labeled branch arrows), representing the best predictor attributes in a data set. These nodes and leaves lead to confidence percentages based on the actual attributes in the training data set, and can then be applied to similarly structured scoring data in order to generate predictions for the scoring observations. Decision trees tell us what is predicted, how confident we can be in the prediction, and *how we arrived at* the prediction. The 'how we arrived at' portion of a decision tree's output is shown in a graphical view of the tree.

REVIEW QUESTIONS

- 1) What characteristics of a data set's attributes might prompt you to choose a decision tree data mining methodology, rather than a logistic or linear regression approach? Why?
- 2) Run this chapter's model using the `gain_ratio` algorithm and make a note of three or four individuals' prediction and confidences. Then re-run the model under `gini_index`. Locate the people you noted. Did their prediction and/or confidences change? Look at their attribute values and compare them to the nodes and leaves in the decision tree. Explain why you think at least one person's prediction changed under Gini, based on that person's attributes and the tree's nodes.
- 3) What are confidence percentages used for, and why would they be important to consider, in addition to just considering the prediction attribute?
- 4) How do you keep an attribute, such as a person's name or ID number, that should not be considered predictive in a process's model, but is useful to have in the data mining results?

- 5) If your decision tree is large or hard to read, how can you adjust its visual layout to improve readability?

EXERCISE

For this chapter's exercise, you will make a decision tree to predict whether or not you, and others you know would have lived, died, or been lost if you had been on the Titanic. Complete the following steps.

- 1) Conduct an Internet search for passenger lists for the Titanic. The search term "Titanic passenger list" in your favorite search engine will yield a number of web sites containing lists of passengers.
- 2) Select from the sources you find a sample of passengers. You do not need to construct a training data set of every passenger on the Titanic (unless you want to), but get at least 30, and preferably more. The more robust your training data set is, the more interesting your results will be.
- 3) In a spreadsheet in OpenOffice Calc, enter these passengers' data.
 - a. Record attributes such as their name, age, gender, class of service they traveled in, race or nationality if known, or other attributes that may be available to you depending on the detail level of the data source you find.
 - b. Be sure to have at least four attributes, preferably more. Remember that the passengers' names or ID numbers won't be predictive, so that attribute shouldn't be counted as one of your predictor attributes.
 - c. Add to your data set whether the person lived (i.e. was rescued from a life boat or from the water), died (i.e. their body was recovered), or was lost (i.e. was on the Titanic's manifest but was never accounted for and therefore presumed dead after the ship's sinking). Call this attribute 'Survival_Result'.
 - d. Save this spreadsheet as a CSV file and then import it into your RapidMiner repository. Set the Survival_Result attribute's role to be your label. Set other

attributes which are not predictive, such as names, to not be considered in the decision tree model.

- e. Add a Decision Tree operator to your stream.
- 4) In a new, blank spreadsheet in OpenOffice Calc, duplicate the attribute names from your training data set, with the exception of Survival_Result. You will predict this attribute using your decision tree.
- 5) Enter data for yourself and people that you know into this spreadsheet.
 - a. For some attributes, you may have to decide what to put. For example, the author acknowledges that based on how relentlessly he searches for the absolutely cheapest ticket when shopping for airfare, he almost certainly would have been in 3rd class if he had been on the Titanic. He further knows some people who very likely would have been in 1st class.
 - b. If you want to include some people in your data set but you don't know every single attribute for them, remember, decision trees can handle some missing values.
 - c. Save this spreadsheet as a CSV file and import it into your RapidMiner repository.
 - d. Drag this data set into your process and ensure that attributes that are not predictive, such as names, will not be included as predictors in the model.
- 6) Apply your decision tree model to your scoring data set.
- 7) Run your model using gain_ratio. Report your tree nodes, and discuss whether you and the people you know would have lived, died or been lost.
- 8) Re-run your model using gini_index. Report differences in your tree's structure. Discuss whether your chances for survival increase under Gini.
- 9) Experiment with changing leaf and split sizes, and other decision tree algorithm criteria, such as information_gain. Analyze and report your results.