

CHAPTER NINE: LOGISTIC REGRESSION

CONTEXT AND PERSPECTIVE

Remember Sonia, the health insurance program director from Chapter 6? Well, she's back for more help too! Her k-means clustering project was so helpful in finding groups of folks who could benefit from her programs, that she wants to do more. This time around, she is concerned with helping those who have suffered heart attacks. She wants to help them improve lifestyle choices, including management of weight and stress, in order to improve their chances of *not* suffering a second heart attack. Sonia is wondering if, with the right training data, we can predict the chances of her company's policy holders suffering second heart attacks. She feels like she could really help some of her policy holders who have suffered heart attacks by offering weight, cholesterol and stress management classes or support groups. By lowering these key heart attack risk factors, her employer's clients will live healthier lives, and her employer's risk at having to pay costs associated with treatment of second heart attacks will also go down. Sonia thinks she might even be able to educate the insured individuals about ways to save money in other aspects of their lives, such as their life insurance premiums, by being able to demonstrate that they are now a lower risk policy holder.

LEARNING OBJECTIVES

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

- Explain what logistic regression is, how it is used and the benefits of using it.
- Recognize the necessary format for data in order to perform predictive logistic regression.
- Develop a logistic regression data mining model in RapidMiner using a training data set.
- Interpret the model's outputs and apply them to a scoring data set in order to deploy the model.

ORGANIZATIONAL UNDERSTANDING

Sonia's desire is to expand her data mining activities to determine what kinds of programs she should develop to help victims of heart attacks avoid suffering a recurrence. She knows that several risk factors such as weight, high cholesterol and stress contribute to heart attacks, particularly in those who have already suffered one. She also knows that the cost of providing programs developed to help mitigate these risks is a fraction of the cost of providing medical care for a patient who has suffered multiple heart attacks. Getting her employer on board with funding the programs is the easy part. Figuring out which patients will benefit from which programs is trickier. She is looking to us to provide some guidance, based on data mining, to figure out which patients are good candidates for which programs. Sonia's bottom line is that she wants to know whether or not something (a second heart attack) is likely to happen, and if so, how likely it is that it will or will not happen. **Logistic regression** is an excellent tool for predicting the likelihood of something happening or not.

DATA UNDERSTANDING

Sonia has access to the company's medical claims database. With this access, she is able to generate two data sets for us. This first is a list of people who have suffered heart attacks, with an attribute indicating whether or not they have had more than one; and the second is a list of those who have had a first heart attack, but not a second. The former data set, comprised of 138 observations, will serve as our training data; while the latter, comprised of 690 peoples' data, will be for scoring. Sonia's hope is to help this latter group of people avoid becoming second heart attack victims. In compiling the two data sets we have defined the following attributes:

- **Age:** The age in years of the person, rounded to the nearest whole year.
- **Marital_Status:** The person's current marital status, indicated by a coded number: 0—Single, never married; 1—Married; 2—Divorced; 3—Widowed.
- **Gender:** The person's gender: 0 for female; 1 for male.
- **Weight_Category:** The person's weight categorized into one of three levels: 0 for normal weight range; 1 for overweight; and 2 for obese.
- **Cholesterol:** The person's cholesterol level, as recorded at the time of their treatment for their most recent heart attack (their *only* heart attack, in the case of those individuals in the scoring data set.)

- **Stress_Management:** A binary attribute indicating whether or not the person has previously attended a stress management course: 0 for no; 1 for yes.
- **Trait_Anxiety:** A score on a scale of 0 to 100 measuring the level of each person's natural stress levels and abilities to cope with stress. A short time after each person in each of the two data sets had recovered from their first heart attack, they were administered a standard test of natural anxiety. Their scores are tabulated and recorded in this attribute along five point increments. A score of 0 would indicate that the person never feels anxiety, pressure or stress in any situation, while a score of 100 would indicate that the person lives in a constant state of being overwhelmed and unable to deal with his or her circumstances.
- **2nd_Heart_Attack:** This attribute exists only in the training data set. It will be our label, the prediction or target attribute. In the training data set, the attribute is set to 'yes' for individuals who have suffered second heart attacks, and 'no' for those who have not.

DATA PREPARATION

Two data sets have been prepared and are available for you to download from the companion web site. These are labeled Chapter09DataSet_Training.csv, and Chapter09DataSet_Scoring.csv. If you would like to follow along with this chapter's example, download these two datasets now, and complete the following steps:

- 1) Begin the process of importing the training data set first. For the most part, the process will be the same as what you have done in past chapters, but for logistic regression, there are a few subtle differences. Be sure to set the first row as the attribute names. On the fourth step, when setting data types and attribute roles, you will need to make at least one change. Be sure to set the 2nd Heart Attack data type to 'nominal', rather than binominal. Even though it is a yes/no field, and RapidMiner will default it to binominal because of that, the Logistic Regression operator we'll be using in our modeling phase expects the label to be nominal. RapidMiner does not offer binominal-to-nominal or integer-to-nominal operators, so we need to be sure to set this target attribute to the needed data type of 'nominal' as we import it. This is shown in Figure 9-1:

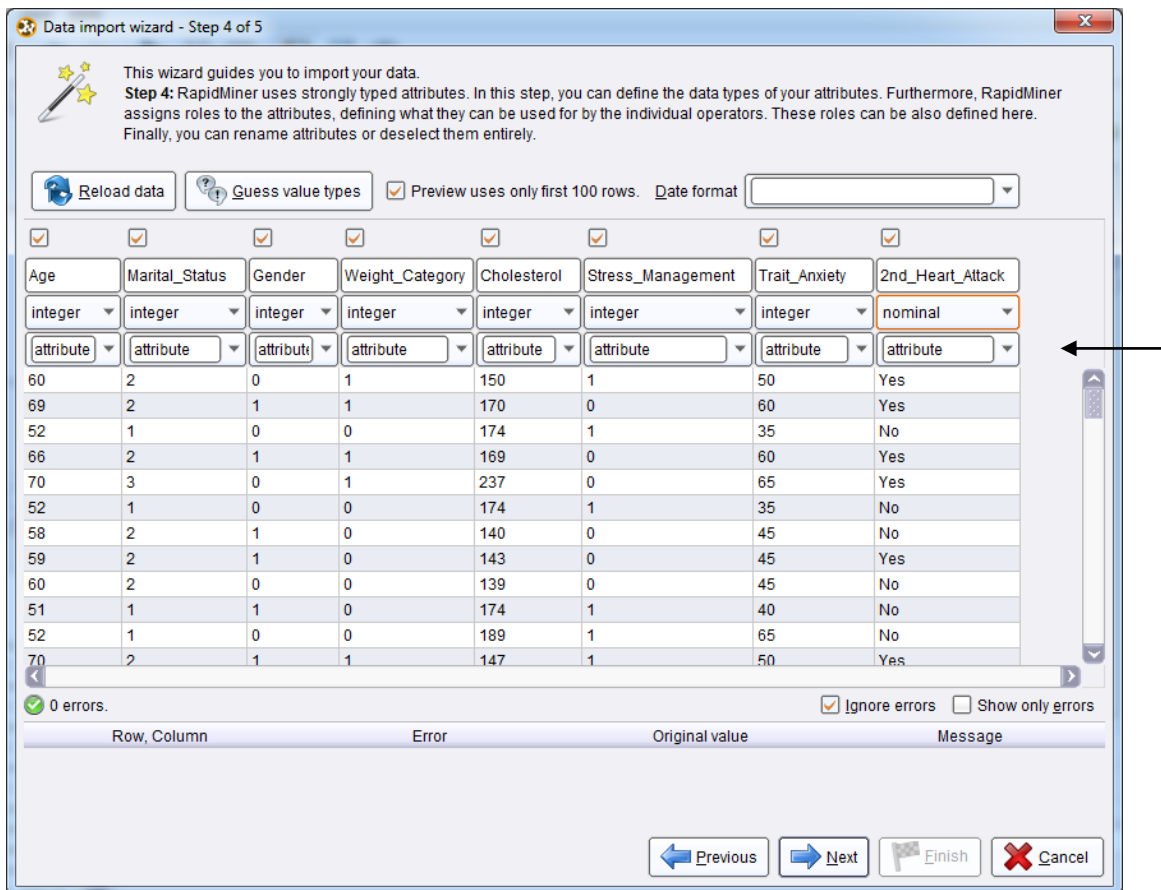


Figure 9-1. Setting the 2nd_Heart_Attack attribute's data type to 'nominal' during import.

- 2) At this time you can also change the 2nd_Heart_Attack attribute's role to 'label', if you wish. We have not done this in Figure 9-1, and subsequently we will be adding a Set Role operator to our stream as we continue our data preparation.
- 3) Complete the data import process for the training data, then drag and drop the data set into a new, blank main process. Rename the data set's Retrieve operator as 'Training'.
- 4) Import the scoring data set now. Be sure the data type for all attributes is 'integer'. This should be the default, but may not be, so double check. Since the 2nd_Heart_Attack attribute is not included in the scoring data set, you don't need to worry about changing it as you did in step 1. Complete the import process, drag and drop the scoring data set into your main process and rename this data set's Retrieve operator to be Scoring. Your model should now appear similar to Figure 9-2.

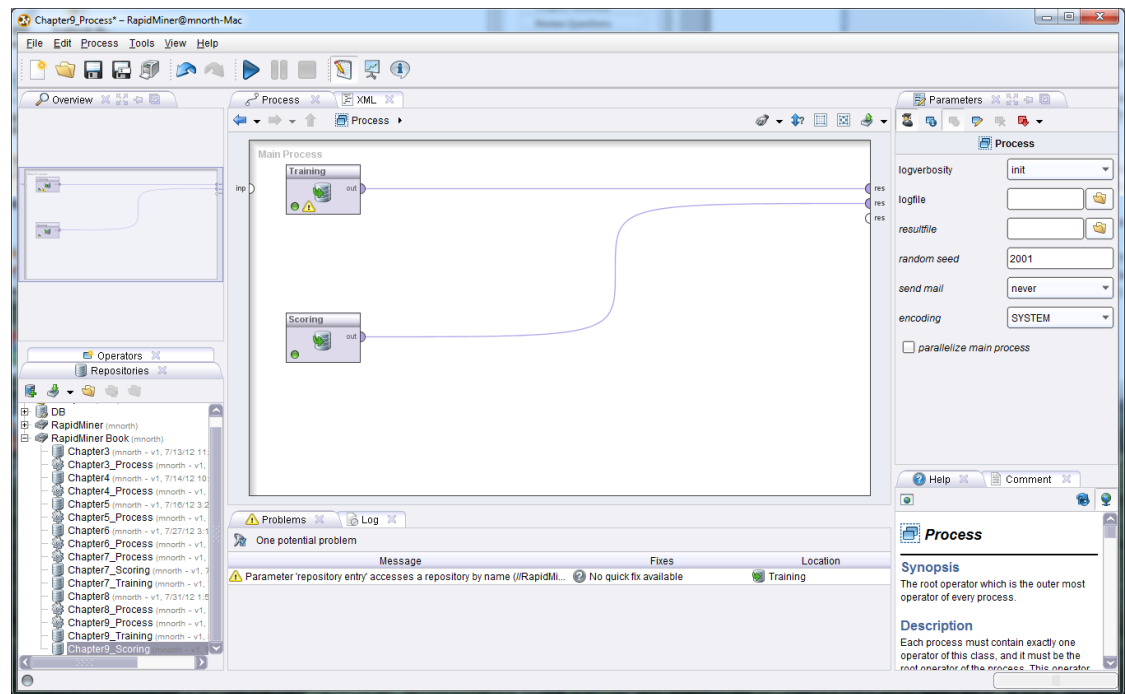


Figure 9-2. The training and scoring data sets in a new main process window in RapidMiner.

- 5) Run the model and compare the ranges for all attributes between the scoring and training result set tabs (Figures 9-3 and 9-4, respectively). You should find that the ranges are the same. As was the case with Linear Regression, the scoring values must all fall within the lower and upper bounds set by the corresponding values in the training data set. We can see in Figures 9-3 and 9-4 that this is the case, so our data are very clean, they were prepared during extraction from Sonia's source database, and we will not need to do further data preparation in order to filter out observations with inconsistent values or modify missing values.

Role	Name	Type	Statistics	Range	Missings
regular	Age	integer	avg = 62.932 +/- 7.899	[42.000 ; 81.000]	0
regular	Marital_Status	integer	avg = 1.696 +/- 0.822	[0.000 ; 3.000]	0
regular	Gender	integer	avg = 0.623 +/- 0.485	[0.000 ; 1.000]	0
regular	Weight_Category	integer	avg = 0.920 +/- 0.763	[0.000 ; 2.000]	0
regular	Cholesterol	integer	avg = 178.265 +/- 32.289	[122.000 ; 239.000]	0
regular	Stress_Management	integer	avg = 0.457 +/- 0.498	[0.000 ; 1.000]	0
regular	Trait_Anxiety	integer	avg = 55.435 +/- 12.337	[35.000 ; 80.000]	0

Figure 9-3. Meta data for the scoring data set (note absence of 2nd_Heart_Attack attribute).

Chapter9_Process - RapidMiner@mnorth-Mac

File Edit Process Tools View Help

Result Overview ExampleSet (Scoring) ExampleSet (Training)

Meta Data View Data View Plot View Advanced Charts Annotations

ExampleSet (138 examples, 0 special attributes, 8 regular attributes)

Role	Name	Type	Statistics	Range	Missings
regular	Age	integer	avg = 62.978 +/- 7.853	[42.000 ; 81.000]	0
regular	Marital_Status	integer	avg = 1.696 +/- 0.825	[0.000 ; 3.000]	0
regular	Gender	integer	avg = 0.623 +/- 0.486	[0.000 ; 1.000]	0
regular	Weight_Category	integer	avg = 0.920 +/- 0.765	[0.000 ; 2.000]	0
regular	Cholesterol	integer	avg = 177.391 +/- 32.263	[122.000 ; 239.000]	0
regular	Stress_Management	integer	avg = 0.442 +/- 0.498	[0.000 ; 1.000]	0
regular	Trait_Anxiety	integer	avg = 55.435 +/- 12.373	[35.000 ; 80.000]	0
regular	2nd_Heart_Attack	nominal	mode = No (70), least = Yes (68)	Yes (68), No (70)	0

Log

Repositories

- Samples (none)
- DB
- RapidMiner (mnorth)
- RapidMiner Book (mnorth)
 - Chapter3 (mnorth - v1, 7/13/12 11)
 - Chapter3_Process (mnorth - v1)
 - Chapter4 (mnorth - v1, 7/14/12 10)
 - Chapter4_Process (mnorth - v1)
 - Chapter5 (mnorth - v1, 7/16/12 3:2)
 - Chapter5_Process (mnorth - v1)
 - Chapter6 (mnorth - v1, 7/27/12 3:4)
 - Chapter6_Process (mnorth - v1)
 - Chapter7_Process (mnorth - v1)
 - Chapter7_Scoring (mnorth - v1, 7/31/12 1:5)
 - Chapter7_Training (mnorth - v1)
 - Chapter8 (mnorth - v1, 7/31/12 1:5)
 - Chapter8_Process (mnorth - v1)
 - Chapter9_Process (mnorth - v1)
 - Chapter9_Scoring (mnorth - v1)
 - Chapter9_Training (mnorth - v1)

Figure 9-4. Meta data for the training data set (2nd_Heart_Attack attribute is present with 'nominal' data type.) Note that all scoring ranges fall within all training ranges.

- 6) Switch back to design perspective and add a Set Role operator to your training stream. Remember that if you designated 2nd_Heart_Attack to have a 'label' role during the import process, you won't need to add a Set Role operator at this time. We did not do this in the book example, so we need the operator to designate 2nd_Heart_Attack as our label, our target attribute:

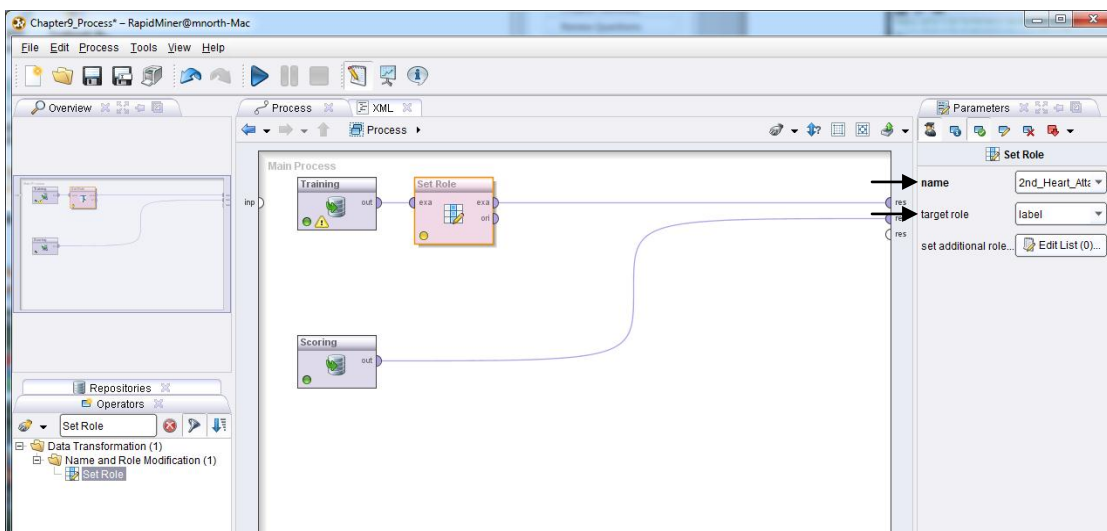


Figure 9-5. Configuring the 2nd_Heart_Attack attribute's role in preparation for logistic regression mining.

With the label attribute set, we are now prepared to begin...

MODELING

- 7) Using the search field in the Operators tab, locate the Logistic Regression operator. You will see that if you just search for the word 'logistic' (as has been done in Figure 9-6), there are several different logistic, and logistic regression operators available to you in RapidMiner. We will use the first one in this example, however, you are certainly encouraged to experiment with the others as you would like. Drag the Logistic Regression operator into your training stream.

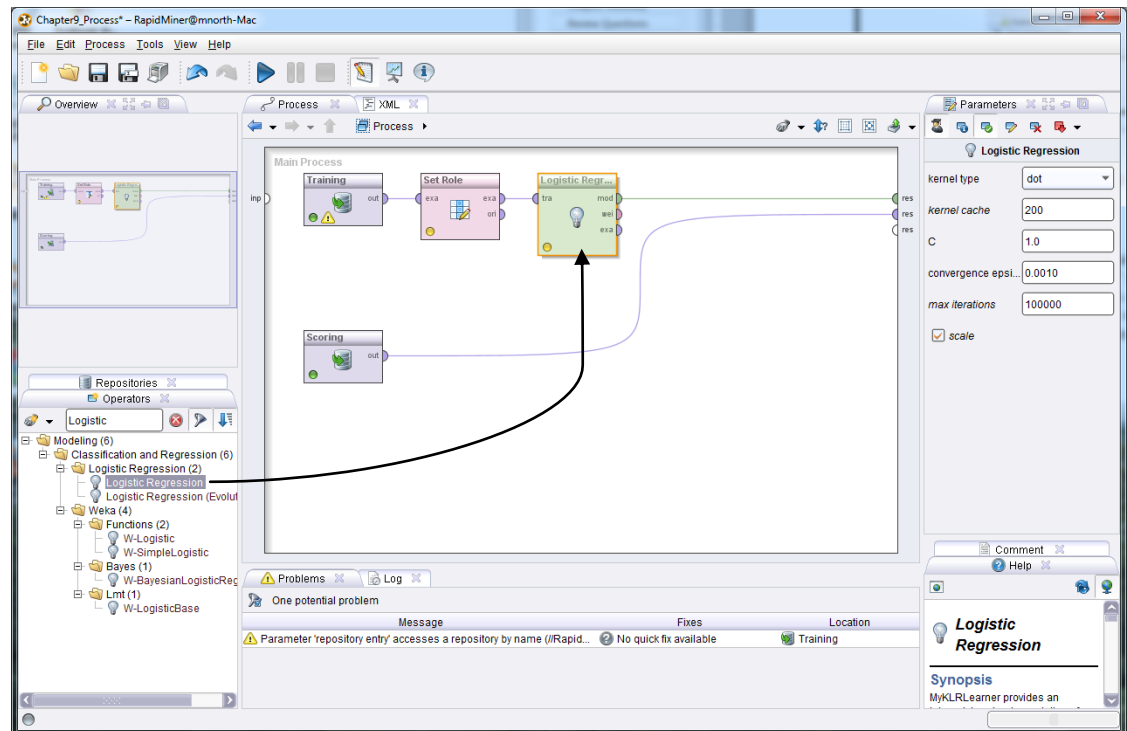


Figure 9-6. The Logistic Regression operator in our training stream.

- 8) The Logistic Regression operator will generate coefficients for each of our predictor attributes, in much the same way that the linear regression operator did. If you would like to see these, you can run your model now. The algebraic formula for logistic regression is different and a bit more complicated than the one for linear regression. We are no longer calculating the slope of a straight line, but rather, we are trying to determine the likelihood of an observation falling at a given point along a curvy and less well-defined imaginary line through a data set. The coefficients for logistic regression are used in that formula.

- 9) If you ran your model to see your coefficients, return now to design perspective. As you have done in our most recent few chapter examples, add an Apply Model operator to your stream, to bring the training and scoring data sets together. Remember that you may need to disconnect and reconnect some ports, as we did in Chapter 7 (step 13), in order to merge your two streams together. Be sure your *lab* and *mod* ports are both connected to *res* ports.

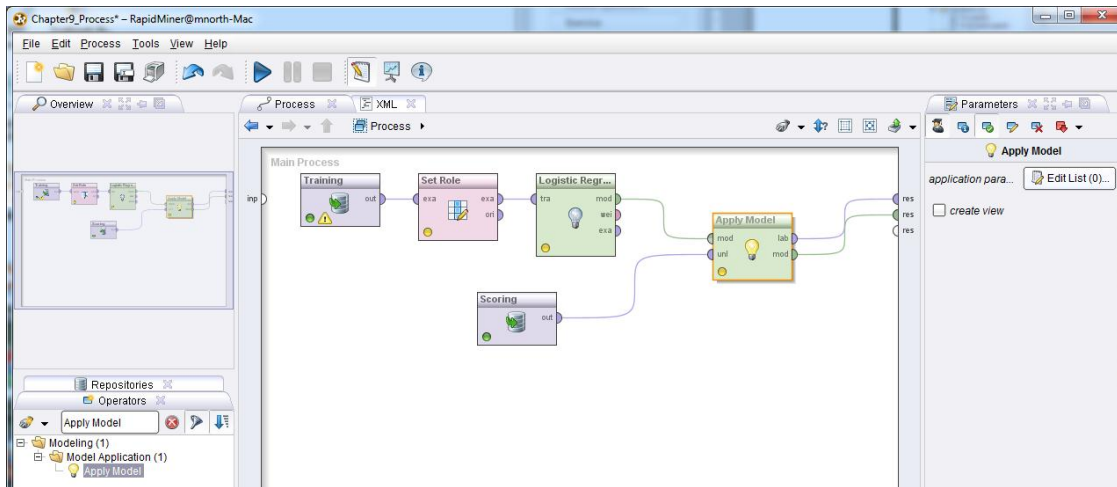


Figure 9-7. Applying the model to the scoring data set.

We are finished building the model. Run it now, and we will proceed to...

EVALUATION

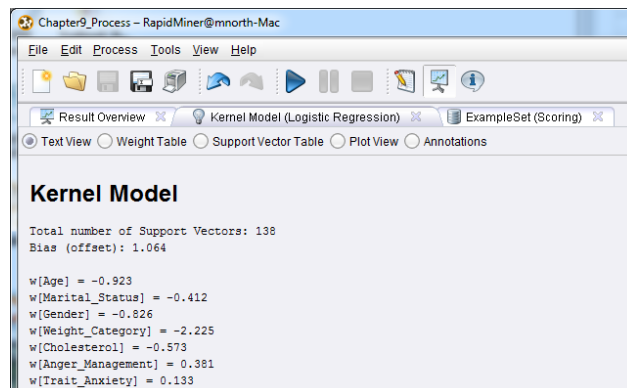


Figure 9-8. Coefficients for each predictor attribute.

The initial tab shown in results perspective is a list of our coefficients. These coefficients are used in the logistic regression algorithm to predict whether or not each person in our scoring data set

will suffer a second heart attack, and if so, how confident we are that the prediction will come true. Switch to the Scoring results tab. We will look first at the meta data (Figure 9-9).

Role	Name	Type	Statistics	Range	Missings
confidence_Yes	confidence(Yes)	real	avg = 0.473 +/- 0.420	[0.000 ; 0.996]	0
confidence_No	confidence(No)	real	avg = 0.527 +/- 0.420	[0.004 ; 1.000]	0
prediction	prediction(2nd_Heart_Attack)	nominal	mode = No (357), least = Yes (333)	Yes (333), No (357)	0
regular	Age	integer	avg = 62.932 +/- 7.899	[42.000 ; 81.000]	0
regular	Marital_Status	integer	avg = 1.696 +/- 0.822	[0.000 ; 3.000]	0
regular	Gender	integer	avg = 0.623 +/- 0.485	[0.000 ; 1.000]	0
regular	Weight_Category	integer	avg = 0.920 +/- 0.763	[0.000 ; 2.000]	0
regular	Cholesterol	integer	avg = 178.265 +/- 32.289	[122.000 ; 239.000]	0
regular	Stress_Management	integer	avg = 0.457 +/- 0.498	[0.000 ; 1.000]	0
regular	Trait_Anxiety	integer	avg = 55.435 +/- 12.337	[35.000 ; 80.000]	0

Figure 9-9. Meta data for our scoring predictions.

We can see in this figure that RapidMiner has generated three new attributes for us: confidence(Yes), confidence(No), and prediction(2nd_Heart_Attack). In our Statistics column, we find that out of the 690 people represented, we're predicting that 357 *will not* suffer second heart attacks, and that 333 will. Sonia's hope is that she can engage these 333, and perhaps some of the 357 with low confidence levels on their 'No' prediction, in programs to improve their health, and thus their chances of avoiding another heart attack. Let's switch to Data View.

Row No.	confidence(Yes)	confidence(No)	prediction(2nd_Heart_Attack)	Age	Marital_Status	Gender	Weight_Category	Cholesterol	Stress_Management	Trait_Anxiety
1	0.139	0.861	No	61	0	1	1	139	1	50
2	0.937	0.063	Yes	55	2	1	2	163	0	40
3	0.275	0.725	No	53	1	1	1	172	0	55
4	0.952	0.048	Yes	58	1	1	2	206	0	70
5	0.367	0.633	No	62	2	1	1	148	1	50
6	0.026	0.974	No	70	1	0	0	172	0	60
7	0.002	0.998	No	52	1	0	0	171	1	35
8	0.211	0.789	No	50	1	1	1	172	0	55
9	0.756	0.244	Yes	67	2	1	1	172	0	60
10	0.326	0.674	No	62	1	1	1	166	1	50
11	0.992	0.008	Yes	66	2	1	2	220	0	60
12	0.030	0.970	No	56	2	1	0	141	0	45
13	0.988	0.012	Yes	77	2	1	2	181	1	80
14	0.693	0.307	Yes	64	2	1	1	174	0	60
15	0.502	0.498	Yes	67	2	1	1	146	1	50

Figure 9-10. Predictions for our 690 patients who have suffered a first heart attack.

In Figure 9-10, we can see that each person has been given a predication of ‘No’ (they won’t suffer a second heart attack), or ‘Yes’ (they will). It is critically important to remember at this point of our evaluation that if this were real, and not a textbook example, these would be real people, with names, families and lives. Yes, we are using data to evaluate their health, but we shouldn’t treat these people like numbers. Hopefully our work and analysis will help our imaginary client Sonia in her efforts to serve these people better. When data mining, we should always keep the human element in mind, and we’ll talk more about this in Chapter 14.

So we have these predictions that some people in our scoring data set are on the path to a second heart attack and others are not, but how confident are we in these predictions? The confidence(Yes) and confidence(No) attributes can help us answer that question. To start, let’s just consider the person represented on Row 1. This is a single (never been married) 61 year old man. He has been classified as overweight, but has lower than average cholesterol (the mean shown in our meta data in Figure 9-9 is just over 178). He scored right in the middle on our trait anxiety test at 50, and has attended stress management class. With these personal attributes, compared with those in our training data, our model offers us an 86.1% level of confidence that the ‘No’ prediction is correct. This leaves us with 13.9% worth of doubt in our prediction. The ‘No’ and ‘Yes’ values will always total to 1, or in other words, 100%. For each person in the data set, their attributes are fed into the logistic regression model, and a prediction with confidence percentages is calculated.

Let’s consider one other person as an example in Figure 9-10. Look at Row 11. This is a 66 year old man who’s been divorced. He’s above the average values in every attribute. While he’s not as old as some in our data set, he is getting older, and he’s obese. His cholesterol is among the highest in our data set, he scored higher than average on the trait anxiety test and hasn’t been to a stress management class. We’re predicting, with 99.2% confidence, that this man will suffer a second heart attack. The warning signs are all there, and Sonia can now see them fairly easily. With an understanding of how to read the output, Sonia can now proceed to...

DEPLOYMENT

In the context of the person represented on Row 11, it seems pretty obvious that Sonia should try to reach out to this gentleman right away, offering help in every aspect. She may want to help him find a weight loss support group, such as Overeaters Anonymous, provide information about dealing with divorce and/or stress, and encourage the person to work with his doctor to better regulate his cholesterol through diet and perhaps medication as well. There may be a number of the 690 individuals who fairly clearly need specific help. Click twice on the attribute name confidence(Yes). Clicking on a column heading (the attribute name) in RapidMiner results perspective will sort the data set by that attribute. Click it once to sort in ascending order, twice to re-sort in descending order, and a third time to return the data set to its original state. Figure 9-11 shows our results sorted in descending order on the confidence(Yes) attribute.

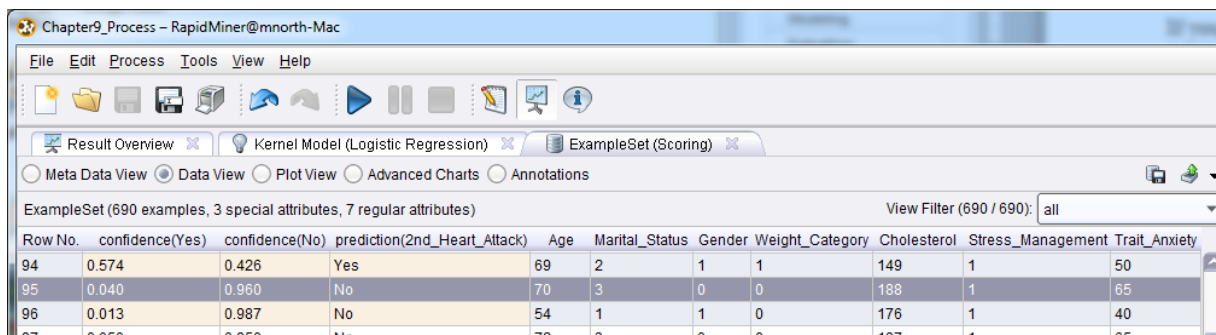
Row No.	confidence(Yes)	confidence(No)	prediction(2nd_Heart_Attack)	Age	Marital_Status	Gender	Weight_Category	Cholesterol	Stress_Management	Trait_Anxiety
677	0.996	0.004	Yes	76	3	1	2	174	0	75
433	0.995	0.005	Yes	74	3	1	2	177	0	75
605	0.995	0.005	Yes	74	3	1	2	175	0	75
474	0.995	0.005	Yes	73	3	1	2	181	0	75
687	0.994	0.006	Yes	73	3	1	2	177	0	75
633	0.994	0.006	Yes	69	2	1	2	221	0	60
436	0.993	0.007	Yes	67	2	1	2	223	0	60
408	0.993	0.007	Yes	67	2	1	2	222	0	60
141	0.992	0.008	Yes	66	2	1	2	224	0	60
170	0.992	0.008	Yes	66	2	1	2	223	0	60
308	0.992	0.008	Yes	66	2	1	2	223	0	60
483	0.992	0.008	Yes	66	2	1	2	223	0	60
189	0.992	0.008	Yes	66	2	1	2	222	0	60
262	0.992	0.008	Yes	66	2	1	2	222	0	60
671	0.992	0.008	Yes	66	2	1	2	222	0	60
213	0.992	0.008	Yes	66	2	1	2	221	0	60
128	0.992	0.008	Yes	73	2	1	2	162	0	40
11	0.992	0.008	Yes	66	2	1	2	220	0	60
335	0.992	0.008	Yes	66	2	1	2	220	0	60
318	0.992	0.008	Yes	73	2	1	2	161	0	40

Figure 9-11. Results sorted by confidence(Yes) in descending order (two clicks on the attribute name).

If you were to count down from the first record (Row 667) to the point at which our confidence(Yes) value is 0.950, you would find that there are 140 individuals in the data set for whom we have a 95% or better confidence that they are at risk for heart attack recurrence (and that's not rounding up those who have a 0.949 in the 'Yes' column). So there are some who are

fairly easy to spot. You might notice that many are divorced, but several are also widowed. Loss of a spouse by any means is difficult, so perhaps Sonia can begin by offering more programs to support those who fit this description. Most of these individuals are obese and have cholesterol levels over 200, and none have participated in stress management classes. Sonia has several opportunities to help these individuals, and she would probably offer these folks opportunities to participate in *several* programs, or create one program that offers a holistic approach to physical and mental well-being. Because there are a good number of these individuals who share so many high risk traits, this may be an excellent way to create support groups for them.

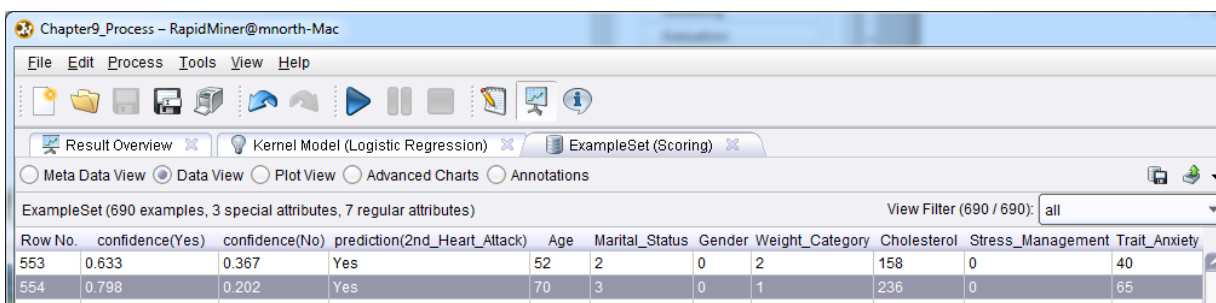
But there are also those individuals in the data set who maybe need help, but aren't quite as obvious, and perhaps only need help in one or two areas. Click confidence(yes) a third time to return the results data to its original state (sorted by Row No.). Now, scroll down until you find Row 95 (highlighted in Figure 9-12). Make a note of this person's attributes.



Row No.	confidence(Yes)	confidence(No)	prediction(2nd_Heart_Attack)	Age	Marital_Status	Gender	Weight_Category	Cholesterol	Stress_Management	Trait_Anxiety
94	0.574	0.426	Yes	69	2	1	1	149	1	50
95	0.040	0.960	No	70	3	0	0	188	1	65
96	0.013	0.987	No	54	1	1	0	176	1	40

Figure 9-12. Examining the first of two similar individuals with different risk levels.

Next locate Row 554 (Figure 9-13).



Row No.	confidence(Yes)	confidence(No)	prediction(2nd_Heart_Attack)	Age	Marital_Status	Gender	Weight_Category	Cholesterol	Stress_Management	Trait_Anxiety
553	0.633	0.367	Yes	52	2	0	2	158	0	40
554	0.798	0.202	Yes	70	3	0	1	236	0	65

Figure 9-13. The second of two similar individuals with different risk levels.

The two people represented on rows 95 and 554 have a lot in common. First of all, they're both in this data set because they've suffered heart attacks. They are both 70 year old women who's husbands have died. Both have trait anxiety of 65 points. And yet we are predicting with 96%

certainty that the first will *not* suffer another heart attack, while predicting with almost 80% that the other *will*. Even their weight categories are similar, though being overweight certainly plays into the second woman's risk. But what is really evident in comparing these two women is that the second woman has a cholesterol level that nearly touches the top of our range in this data set (the upper bound shown in Figure 9-9 is 239), and she hasn't been to stress management classes. Perhaps Sonia can use such comparisons to help this woman understand just how dramatically she can improve her chances of avoiding another heart attack. In essence, Sonia could say: "There are women who are a lot like you who have almost zero chance of suffering another heart attack. By lowering your cholesterol, learning to manage your stress, and perhaps getting your weight down closer to a normal level, you can almost eliminate your risk for another heart attack." Sonia could follow up by offering specific programs for this woman targeted specifically at cholesterol, weight or stress management.

CHAPTER SUMMARY

Logistic regression is an excellent way to predict whether or not something will happen, and how confident we are in such predictions. It takes a number of numeric attributes into account and then uses those through a training data set to predict the probable outcomes in a comparable scoring data set. Logistic regression uses a nominal target attribute (or label, in RapidMiner) to categorize observations in a scoring data set into their probable outcomes.

As with linear regression, the scoring data must have ranges that fall within their corresponding training data ranges. Without such bounds, it is unsafe and unwise to draw assumptions about observations in the scoring data set, since there are no comparable observations in the training data upon which to base your scoring assumptions. When used within these bounds however, logistic regression can help us quickly and easily predict the outcome of some phenomenon in a data set, and to determine how confident we can be in the accuracy of that prediction.

REVIEW QUESTIONS

- 1) What is the appropriate data type for independent variables (predictor attributes) in logistic regression? What about for the dependent variable (target or label attribute)?
- 2) Compare the predictions for Row 15 and 669 in the chapter's example model.
 - a. What is the single difference between these two people, and how does it affect their predicted 2nd_Heart_Attack risk?
 - b. Locate other 67 year old men in the results and compare them to the men on rows 15 and 669. How do they compare?
 - c. Can you spot areas when the men represented on rows 15 and 669 could improve their chances of not suffering a second heart attack?
- 3) What is the difference between confidence(Yes) and confidence(No) in this chapter's example?
- 4) How can you set an attribute's role to be 'label' in RapidMiner without using the Set Role operator? What is one drawback to doing it that way?

EXERCISE

For this chapter's exercise, you will use logistic regression to try to predict whether or not young people you know will eventually graduate from college. Complete the following steps:

- 1) Open a new blank spreadsheet in OpenOffice Calc. At the bottom of the spreadsheet there will be three default tabs labeled Sheet1, Sheet2, Sheet3. Rename the first one Training and the second one Scoring. You can rename the tabs by double clicking on their labels. You can delete or ignore the third default sheet.
- 2) On the training sheet, starting in cell A1 and going across, create attribute labels for five attributes: Parent_Grad, Gender, Income_Level, Num_Siblings, and Graduated.
- 3) Copy each of these attribute names except Graduated into the Scoring sheet.

- 4) On the Training sheet, enter values for each of these attributes for several adults that you know who are at the age that they could have graduated from college by now. These could be family members, friends and neighbors, coworkers or fellow students, etc. Try to do at least 20 observations; 30 or more would be better. Enter husband and wife couples as two separate observations. Use the following to guide your data entry:
 - a. For Parent_Grad, enter a 0 if neither of the person's parents graduated from college, a 1 if one parent did, and a 2 if both parents did. If the person's parents went on to earn graduate degrees, you could experiment with making this attribute even more interesting by using it to hold the total number of college degrees by the person's parents. For example, if the person represented in the observation had a mother who earned a bachelor's, master's and doctorate, and a father who earned a bachelor's and a master's, you could enter a 5 in this attribute for that person.
 - b. For Gender, enter 0 for female and 1 for male.
 - c. For Income_Level, enter a 0 if the person lives in a household with an income level below what you would consider to be below average, a 1 for average, and a 2 for above average. You can estimate or generalize. Be sensitive to others when gathering your data—don't snoop too much or risk offending your data subjects.
 - d. For Num_Siblings, enter the number of siblings the person has.
 - e. For Graduated, put 'Yes' if the person has graduated from college and 'No' if they have not.
- 5) Once you've compiled your Training data set, switch to the Scoring sheet in OpenOffice Calc. Repeat the data entry process for at least 20 (more is better) young people between the ages of 0 and 18 that you know. You will use the training set to try to predict whether or not these young people will graduate from college, and if so, how confident you are in your prediction. Remember this is your scoring data, so you won't provide the Graduated attribute, you'll predict it shortly.
- 6) Use the File > Save As menu option in OpenOffice Calc to save your Training and Scoring sheets as CSV files.
- 7) Import your two CSV files into your RapidMiner repository. Be sure to give them descriptive names.

- 8) Drag your two data sets into a new process window. If you have prepared your data well in OpenOffice Calc, you shouldn't have any missing or inconsistent data to contend with, so data preparation should be minimal. Rename the two retrieve operators so you can tell the difference between your training and scoring data sets.
- 9) One necessary data preparation step is to add a Set Role operator and define the Graduated attribute as your label in your training data. Alternatively, you can set your Graduated attribute as the label during data import.
- 10) Add a Logistic Regression operator to your Training stream.
- 11) Apply your Logistic Regression model to your scoring data and run your model. Evaluate and report your results. Are your confidence percentages interesting? Surprising? Do the predicted Graduation values seem reasonable and consistent with your training data? Does any one independent variable (predictor attribute) seem to be a particularly good predictor of the dependent variable (label or prediction attribute)? If so, why do you think so?

Challenge Step!

- 12) Change your Logistic Regression operator to a different type of Logistic operator (for example, maybe try the Weka W-Logistic operator). Re-run your model. Consider doing some research to learn about the difference between algorithms underlying different logistic approaches. Compare your new results to the original Logistic Regression results and report any interesting findings or differences.