CHAPTER SEVEN:
DISCRIMINANT ANALYSIS

CONTEXT AND PERSPECTIVE

Gill runs a sports academy designed to help high school aged athletes achieve their maximum athletic potential. On the boys side of his academy, he focuses on four major sports: Football, Basketball, Baseball and Hockey. He has found that while many high school athletes enjoy participating in a number of sports in high school, as they begin to consider playing a sport at the college level, they would prefer to specialize in one sport. As he's worked with athletes over the years, Gill has developed an extensive data set, and he now is wondering if he can use past performance from some of his previous clients to predict prime sports for up-and-coming high school athletes. Ultimately, he hopes he can make a recommendation to each athlete as to the sport in which they should most likely choose to specialize. By evaluating each athlete's performance across a battery of test, Gill hopes we can help him figure out for which sport each athlete has the highest aptitude.

LEARNING OBJECTIVES

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

- Explain what discriminant analysis is, how it is used and the benefits of using it.
- Recognize the necessary format for data in order to perform discriminant analysis.
- Explain the differences and similarities between k-Means clustering and discriminant analysis.
- Develop a discriminant analysis data mining model in RapidMiner using a training data set.
- Interpret the model output and apply it to a scoring data set in order to deploy the model.

ORGANIZATIONAL UNDERSTANDING

Gill's objective is to examine young athletes and, based upon their performance across a number of metrics, help them decide which sport is the most prime for their specialized success. Gill recognizes that all of his clients possess some measure of athleticism, and that they enjoy participating in a number of sports. Being young, athletic, and adaptive, most of his clients are quite good at a number of sports, and he has seen over the years that some people are so naturally gifted that they would excel in any sport they choose for specialization. Thus, he recognizes, as a limitation of this data mining exercise, that he may not be able to use data to determine an athlete's "best" sport. Still, he has seen metrics and evaluations work in the past, and has seen that some of his previous athletes really were pre-disposed to a certain sport, and that they were successful as they went on to specialize in that sport. Based on his industry experience, he has decided to go ahead with an experiment in mining data for athletic aptitude, and has enlisted our help.

DATA UNDERSTANDING

In order to begin to formulate a plan, we sit down with Gill to review his data assets. Every athlete that has enrolled at Gill's academy over the past several years has taken a battery test, which tested for a number of athletic and personal traits. The battery has been administered to both boys and girls participating in a number of different sports, but for this preliminary study we have decided with Gill that we will look at data only for boys. Because the academy has been operating for some time, Gill has the benefit of knowing which of his former pupils have gone on to specialize in a single sport, and which sport it was for each of them. Working with Gill, we gather the results of the batteries for all former clients who have gone on to specialize, Gill adds the sport each person specialized in, and we have a data set comprised of 493 observations containing the following attributes:

- **Age**: This is the age in years (one decimal precision for the part of the year since the client's last birthday) at the time that the athletic and personality trait battery test was administered. Participants ranged in age from 13-19 years old at the time they took the battery.
- **Strength**: This is the participant's strength measured through a series of weight lifting exercises and recorded on a scale of 0-10, with 0 being limited strength and 10 being

sufficient strength to perform all lifts without any difficulty. No participant scored 8, 9 or 10, but some participants did score 0.

- **Quickness**: This is the participant's performance on a series of responsiveness tests. Participants were timed on how quickly they were able to press buttons when they were illuminated or to jump when a buzzer sounded. Their response times were tabulated on a scale of 0-6, with 6 being extremely quick response and 0 being very slow. Participants scored all along the spectrum for this attribute.
- **Injury**: This is a simple yes (1) / no (0) column indicating whether or not the young athlete had already suffered an athletic-related injury that was severe enough to require surgery or other major medical intervention. Common injuries treated with ice, rest, stretching, etc. were entered as 0. Injuries that took more than three week to heal, that required physical therapy or surgery were flagged as 1.
- **Vision**: Athletes were not only tested on the usual 20/20 vision scale using an eye chart, but were also tested using eye-tracking technology to see how well they were able to pick up objects visually. This test challenged participants to identify items that moved quickly across their field of vision, and to estimate speed and direction of moving objects. Their scores were recorded on a 0 to 4 scale with 4 being perfect vision and identification of moving objects. No participant scored a perfect 4, but the scores did range from 0 to 3.
- **Endurance**: Participants were subjected to an array of physical fitness tests including running, calisthenics, aerobic and cardiovascular exercise, and distance swimming. Their performance was rated on a scale of 0-10, with 10 representing the ability to perform all tasks without fatigue of any kind. Scores ranged from 0 to 6 on this attribute. Gill has acknowledged to us that even finely tuned professional athletes would not be able to score a 10 on this portion of the battery, as it is specifically designed to test the limits of human endurance.
- **Agility**: This is the participant's score on a series of tests of their ability to move, twist, turn, jump, change direction, etc. The test checked the athlete's ability to move nimbly, precisely, and powerfully in a full range of directions. This metric is comprehensive in nature, and is influenced by some of the other metrics, as agility is often dictated by one's strength, quickness, etc. Participants were scored between 0 and 100 on this attribute, and in our data set from Gill, we have found performance between 13 and 80.
- **Decision_Making**: This portion of the battery tests the athlete's process of deciding what to do in athletic situations. Athlete's participated in simulations that tested their choices of

whether or not to swing a bat, pass a ball, move to a potentially advantageous location of a playing surface, etc. Their scores were to have been recorded on a scale of 0 to 100, though Gill has indicated that no one who completed the test should have been able to score lower than a 3, as three points are awarded simply for successfully entering and exiting the decision making part of the battery. Gill knows that all 493 of his former athletes represented in this data set successfully entered and exited this portion, but there are a few scores lower than 3, and also a few over 100 in the data set, so we know we have some data preparation in our future.

- **Prime_Sport**: This attribute is the sport each of the 453 athletes went on to specialize in after they left Gill's academy. This is the attribute Gill is hoping to be able to predict for his current clients. For the boys in this study, this attribute will be one of four sports: football (American, not soccer; sorry soccer fans), Basketball, Baseball, or Hockey.

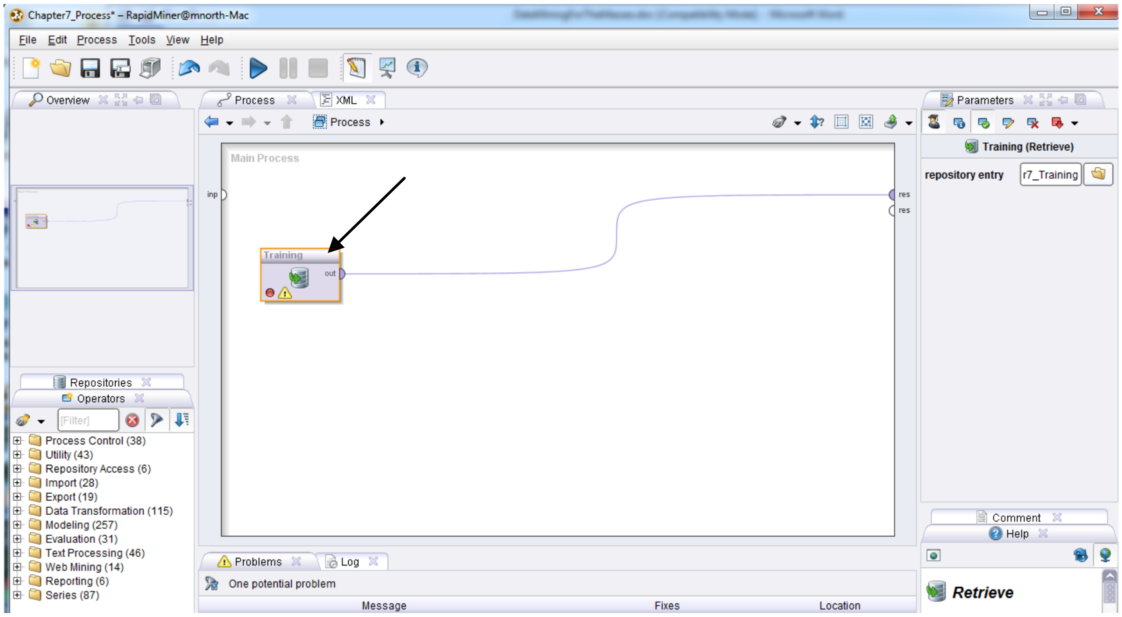
As we analyze and familiarize ourselves with these data, we realize that all of the attributes with the exception of Prime_Sport are numeric, and as such, we could exclude Prime_Sport and conduct a k-means clustering data mining exercise on the data set. Doing this, we might be able group individuals into one sport cluster or another based on the means for each of the attributes in the data set. However, having the Prime_Sport attribute gives us the ability to use a different type of data mining model: **Discriminant Analysis**. Discriminant analysis is a lot like k-means clustering, in that it groups observations together into like-types of values, but it also gives us something more, and that is the ability to *predict*. Discriminant analysis then helps us cross that intersection seen in the Venn diagram in Chapter 1 (Figure 1-2). It is still a data mining methodology for classifying observations, but it classifies them *in a predictive way*. When we have a data set that contains an attribute that we know is useful in predicting the same value for other observations that do not yet have that attribute, then we can use **training data** and **scoring data** to mine predictively. Training data are simply data sets that have that known prediction attribute. For the observations in the training data set, the outcome of the prediction attribute is already known. The prediction attribute is also sometimes referred to as the **dependent attribute (or variable)** or the **target attribute**. It is the thing you are trying to predict. RapidMiner will ask us to set this attribute to be the **label** when we build our model. Scoring data are the observations which have all of the same attributes as the training data set, with the exception of the prediction attribute. We can use the training data set to allow RapidMiner to evaluate the values of all our attributes in the context of the resulting prediction variable (in this case, Prime_Sport), and then compare those values to the scoring data set and predict the Prime_Sport for each observation in the scoring data set. That may seem a little confusing, but our chapter example should help clarify it, so let's move on to the next CRISP-DM step.

DATA PREPARATION

This chapter's example will be a slight divergence from other chapters. Instead of there being a single example data set in CSV format for you to download, there are two this time. You can access the Chapter 7 data sets on the book's companion web site (https://sites.google.com/site/dataminingforthemasses/).

They are labeled Chapter07DataSet_Scoring.csv and Chapter07DataSet_Training.csv. Go ahead and download those now, and import both of them into your RapidMiner repository as you have in past chapters. Be sure to designate the attribute names in the first row of the data sets as you import them. Be sure you give each of the two data sets descriptive names, so that you can tell they are for Chapter 7, and also so that you can tell the difference between the training data set and the scoring data set. After importing them, **drag only the training data set into a new process window**, and then follow the steps below to prepare for and create a discriminant analysis data mining model.

1) Thus far, when we have added data to a new process, we have allowed the operator to simply be labeled 'Retrieve', which is done by RapidMiner by default. For the first time, we will have more than one Retrieve operator in our model, because we have a training data set and a scoring data set. In order to easily differentiate between the two, let's start by renaming the Retrieve operator for the training data set that you've dragged and dropped into your main process window. Right click on this operator and select Rename. You will then be able to type in a new name for this operator. For this example, we will name the operator 'Training', as is depicted in Figure 7-1.



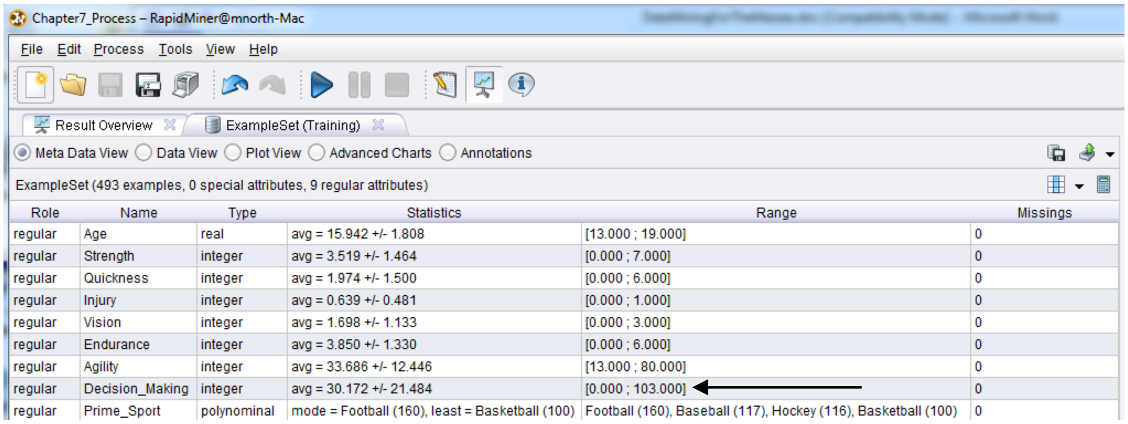
Training (Retrieve)

repository entry: r7_Training

One potential problem

Figure 7-1. Our Retrieve operator renamed as 'Training'.

2) We know from our Data Preparation phase that we have some data that need to be fixed before we can mine this data set. Specifically, Gill noticed some inconsistencies in the Decision_Making attribute. Run your model and let's examine the meta data, as seen in Figure 7-2.

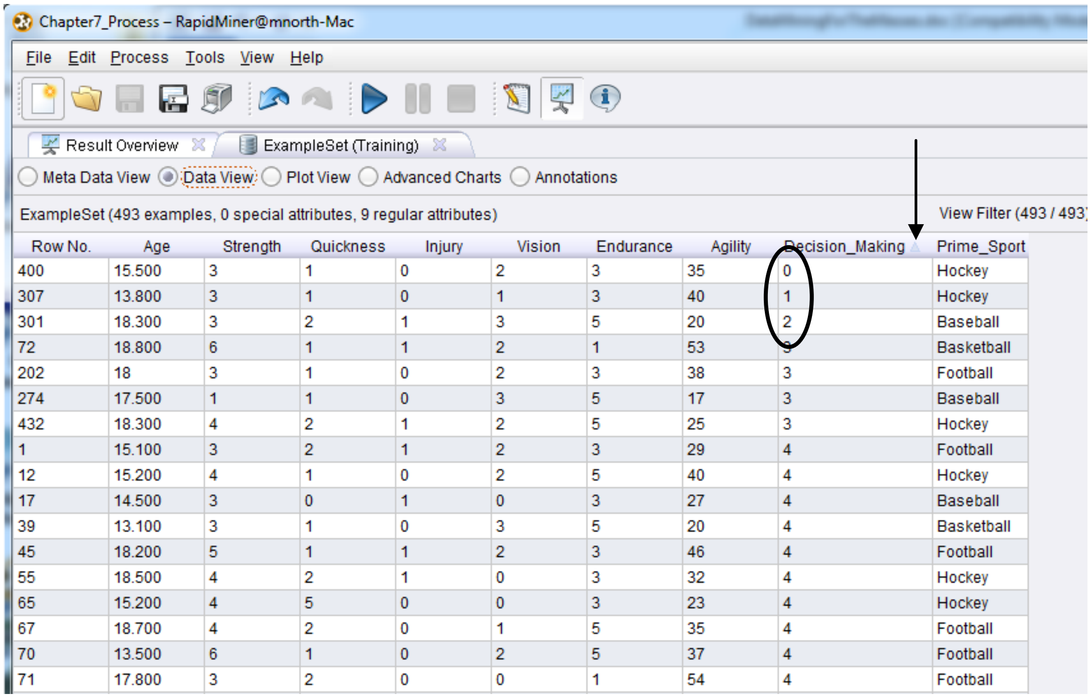

Chapter7_Process – RapidMiner@mnorth-Mac
File Edit Process Tools View Help
Result Overview
ExampleSet (Training)
● Meta Data View ○ Data View ○ Plot View ○ Advanced Charts ○ Annotations

ExampleSet (493 examples, 0 special attributes, 9 regular attributes)

Role	Name	Type	Statistics	Range	Missings
regular	Age	real	avg = 15.942 +/- 1.808	[13.000 ; 19.000]	0
regular	Strength	integer	avg = 3.519 +/- 1.464	[0.000 ; 7.000]	0
regular	Quickness	integer	avg = 1.974 +/- 1.500	[0.000 ; 6.000]	0
regular	Injury	integer	avg = 0.639 +/- 0.481	[0.000 ; 1.000]	0
regular	Vision	integer	avg = 1.698 +/- 1.133	[0.000 ; 3.000]	0
regular	Endurance	integer	avg = 3.850 +/- 1.330	[0.000 ; 6.000]	0
regular	Agility	integer	avg = 33.686 +/- 12.446	[13.000 ; 80.000]	0
regular	Decision_Making	integer	avg = 30.172 +/- 21.484	[0.000 ; 103.000]	0
regular	Prime_Sport	polynominal	mode = Football (160), least = Basketball (100)	Football (160), Baseball (117), Hockey (116), Basketball (100)	0

Figure 7-2. Identifying inconsistent data in the Decision_Making attribute.

3) While still in results perspective, switch to the Data View radio button. Click on the column heading for the Decision_Making attribute. This will sort the attribute from smallest to largest (note the small triangle indicating that the data are sorted in ascending order using this attribute). In this view (Figure 7-3) we see that we have three observations with scores smaller than three. We will need to handle these observations.



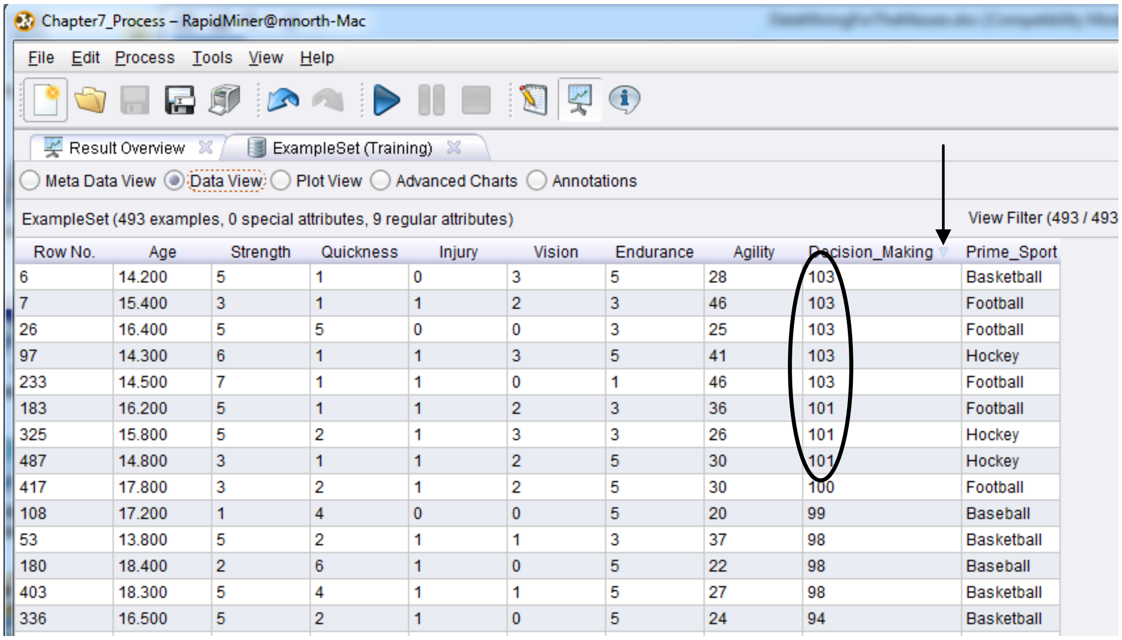
ExampleSet (493 examples, 0 special attributes, 9 regular attributes)

View Filter (493 / 493)

Row No.	Age	Strength	Quickness	Injury	Vision	Endurance	Agility	Decision_Making	Prime_Sport
400	15.500	3	1	0	2	3	35	0	Hockey
307	13.800	3	1	0	1	3	40	1	Hockey
301	18.300	3	2	1	3	5	20	2	Baseball
72	18.800	6	1	1	2	1	53	3	Basketball
202	18	3	1	0	2	3	38	3	Football
274	17.500	1	1	0	3	5	17	3	Baseball
432	18.300	4	2	1	2	5	25	3	Hockey
1	15.100	3	2	1	2	3	29	4	Football
12	15.200	4	1	0	2	5	40	4	Hockey
17	14.500	3	0	1	0	3	27	4	Baseball
39	13.100	3	1	0	3	5	20	4	Basketball
45	18.200	5	1	1	2	3	46	4	Football
55	18.500	4	2	1	0	3	32	4	Hockey
65	15.200	4	5	0	0	3	23	4	Hockey
67	18.700	4	2	0	1	5	35	4	Football
70	13.500	6	1	0	2	5	37	4	Football
71	17.800	3	2	0	0	1	54	4	Football

Figure 7-3. The data set sorted in ascending order by the Decision_Making attribute.

4) Click on the Decision_Making attribute again. This will re-sort the attribute in descending order. Again, we have some values that need to be addressed (Figure 7-4).



ExampleSet (493 examples, 0 special attributes, 9 regular attributes)

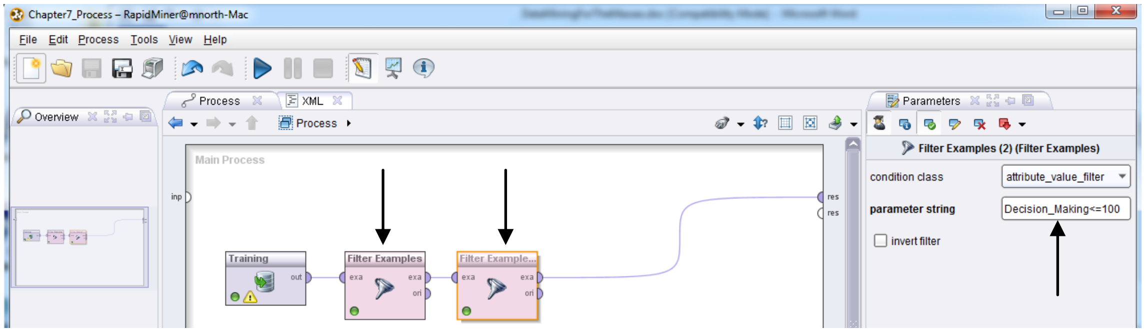
View Filter (493 / 493

Row No.	Age	Strength	Quickness	Injury	Vision	Endurance	Agility	Decision_Making	Prime_Sport
6	14.200	5	1	0	3	5	28	103	Basketball
7	15.400	3	1	1	2	3	46	103	Football
26	16.400	5	5	0	0	3	25	103	Football
97	14.300	6	1	1	3	5	41	103	Hockey
233	14.500	7	1	1	0	1	46	103	Football
183	16.200	5	1	1	2	3	36	101	Football
325	15.800	5	2	1	3	3	26	101	Hockey
487	14.800	3	1	1	2	5	30	101	Hockey
417	17.800	3	2	1	2	5	30	100	Football
108	17.200	1	4	0	0	5	20	99	Baseball
53	13.800	5	2	1	1	3	37	98	Basketball
180	18.400	2	6	1	0	5	22	98	Baseball
403	18.300	5	4	1	1	5	27	98	Basketball
336	16.500	5	2	1	0	5	24	94	Basketball

Figure 7-4. The Decision_Making variable, re-sorted in descending order.

5) Switch back to design perspective. Let's address these inconsistent data by removing them from our training data set. We could set these inconsistent values to missing then set missing values to another value, such as the mean, but in this instance we don't really know

what *should have* been in this variable, so changing these to the mean seems a bit arbitrary. Removing this inconsistencies means only removing 11 of our 493 observations, so rather than risk using bad data, we will simply remove them. To do this, add two Filter Examples operators in a row to your stream. For each of these, set the condition class to attribute_value_filter, and for the parameter strings, enter 'Decision_Making>=3' (without single quotes) for the first one, and 'Decision_Making<=100' for the second one. This will reduce our training data set down to 482 observations. The set-up described in this step is shown in Figure 7-5.



Filter Examples (2) (Filter Examples)

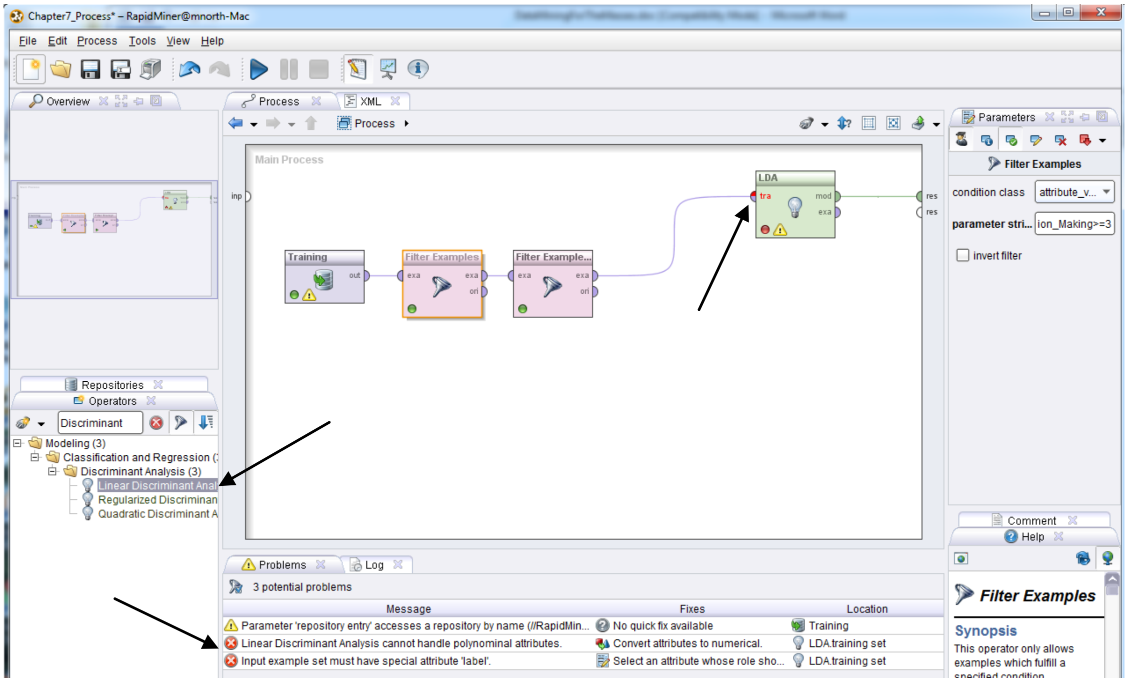
condition class: attribute_value_filter

parameter string: Decision_Making<=100

☐ invert filter

Figure 7-5. Filtering out observations with inconsistent data.

6) If you would like, you can run the model to confirm that your number of observations (examples) has been reduced to 482. Then, in design perspective, use the search field in the Operators tab to look for 'Discriminant' and locate the operator for Linear Discriminant Analysis. Add this operator to your stream, as shown in Figure 7-6.



Filter Examples

condition class: attribute_v...

parameter stri...: ion_Making>=3

☐ invert filter

Filter Examples

Synopsis

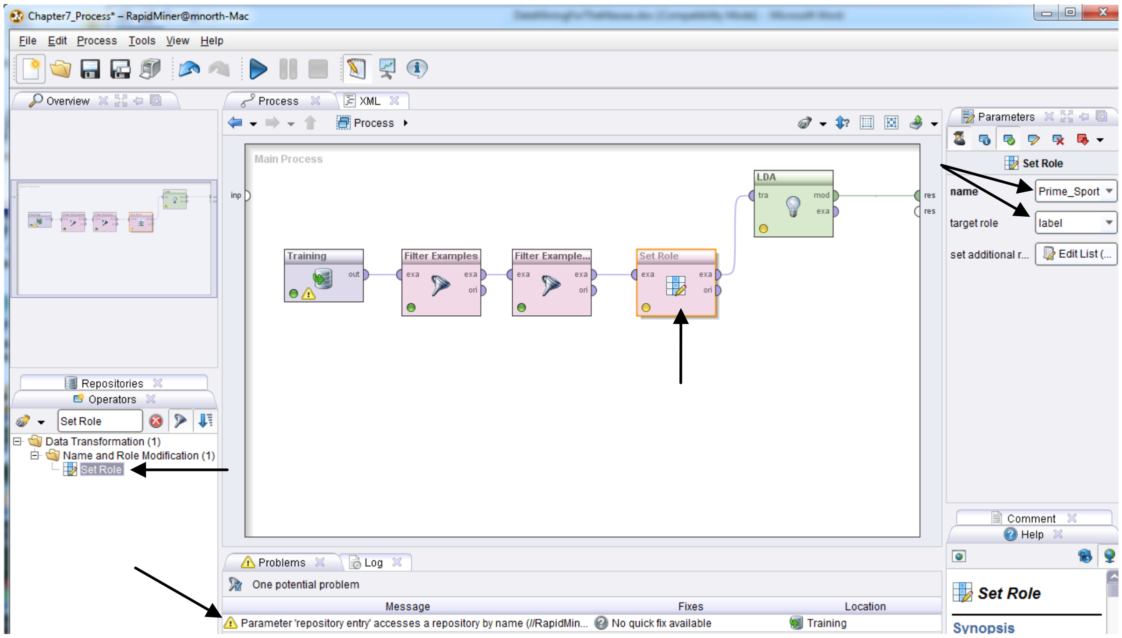
This operator only allows examples which fulfill a specified condition

3 potential problems

Message	Fixes	Location
Parameter 'repository entry' accesses a repository by name (//RapidMin...	No quick fix available	Training
Linear Discriminant Analysis cannot handle polynominal attributes.	Convert attributes to numerical.	LDA.training set
Input example set must have special attribute 'label'.	Select an attribute whose role sho...	LDA.training set

Figure 7-6. Addition of the Linear Discriminant Analysis operator to the model.

7) The *tra* port on the LDA (or Linear Discriminant Analysis) operator indicates that this tool does expect to receive input from a training data set like the one we've provided, but despite this, we still have received two errors, as indicated by the black arrow at the bottom of the Figure 7-6 image. The first error is because of our Prime_Sport attribute. It is data typed as polynominal, and LDA likes attributes that are numeric. This is OK, because the predictor attribute can have a polynominal data type, and the Prime_Sport attribute is the one we want to predict, so this error will be resolved shortly. This is because it is related to the second error, which tells us that the LDA operator wants one of our attributes to be designated as a 'label'. In RapidMiner, the label is the attribute that you want to predict. At the time that we imported our data set, we could have designated the Prime_Sport attribute as a label, rather than as a normal attribute, but it is very simple to change an attribute's role right in your stream. Using the search field in the Operators tab, search for an operator called Set Role. Add this to your stream and then in the parameters area on the right side of the window, select Prime_Sport in the name field, and in target role, select label. We still have a warning (which does not prevent us from continuing), but you will see the errors have now disappeared at the bottom of the RapidMiner window (Figure 7-7).



Set Role

name	Prime_Sport
target role	label
set additional r...	Edit List (...

Operators search: Set Role

Problems | Log

One potential problem

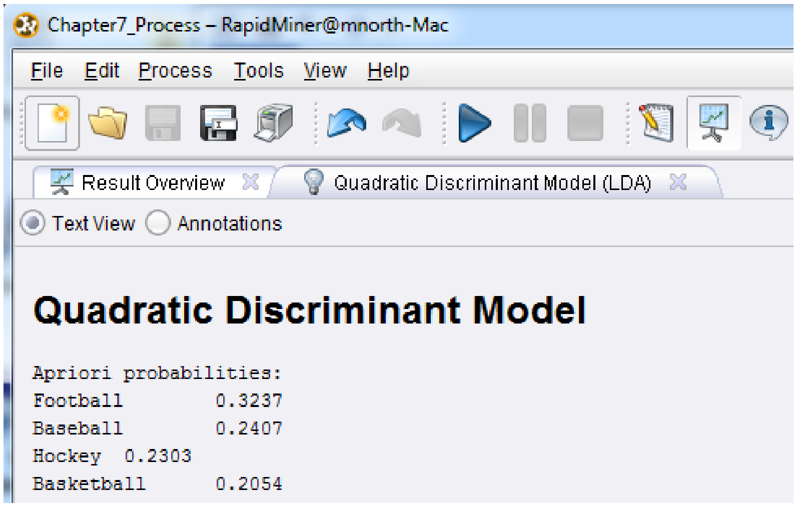
Message	Fixes	Location
Parameter 'repository entry' accesses a repository by name (//RapidMin...	No quick fix available	Training

Figure 7-7. Setting an attribute's role in RapidMiner.

With our inconsistent data removed and our errors resolved, we are now prepared to move on to…

MODELING

8) We now have a functional stream. Go ahead and run the model as it is now. With the *mod* port connected to the *res* port, RapidMiner will generate Discriminant Analysis output for us.


Chapter7_Process – RapidMiner@mnorth-Mac
File Edit Process Tools View Help
Result Overview
Quadratic Discriminant Model (LDA)
Text View
Annotations

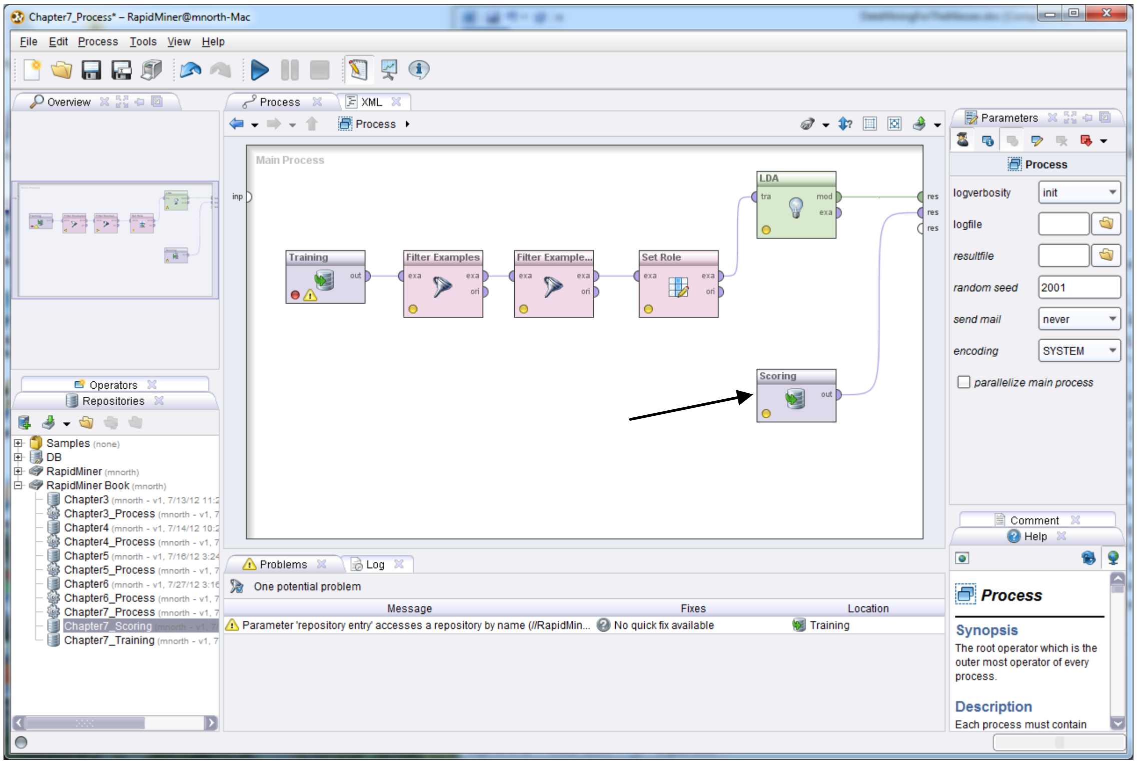
Quadratic Discriminant Model

```
Apriori probabilities:
Football        0.3237
Baseball        0.2407
Hockey  0.2303
Basketball      0.2054
```

Figure 7-8. The results of discriminant analysis on our training data set.

9) The probabilities given in the results will total to 1. This is because at this stage of our Discriminant Analysis model, all that has been calculated is the likelihood of an observation landing in one of the four categories in our target attribute of Prime_Sport. Because this is our training data set, RapidMiner can calculate theses probabilities easily—every observation is already classified. Football has a probability of 0.3237. If you refer back to Figure 7-2, you will see that Football as Prime_Sport comprised 160 of our 493 observations. Thus, the probability of an observation having Football is 160/493, or 0.3245. But in steps 3 and 4 (Figures 7-3 and 7-4), we removed 11 observations that had inconsistent data in their Decision_Making attribute. Four of these were Football observations (Figure 7-4), so our Football count dropped to 156 and our total count dropped to 482: 156/482 = 0.3237. Since we have no observations where the value for Prime_Sport is missing, each possible value in Prime_Sport will have some portion of the total count, and the sum of these portions will equal 1, as is the case in Figure 7-8. These probabilities, coupled with the values for each attribute, will be used to predict the Prime_Sport classification for each of Gill's current clients represented in our scoring data set. Return now to design perspective and in the Repositories tab, drag the Chapter 7 scoring data set over and drop it in the main process window. Do not connect it to your

existing stream, but rather, allow it to connect directly to a *res* port. Right click the operator and rename it to 'Scoring'. These steps are illustrated in Figure7-9.



Process

logverbosity: init

logfile:

resultfile:

random seed: 2001

send mail: never

encoding: SYSTEM

□ parallelize main process

One potential problem

Message	Fixes	Location
Parameter 'repository entry' accesses a repository by name (//RapidMin...	No quick fix available	Training

Process

Synopsis

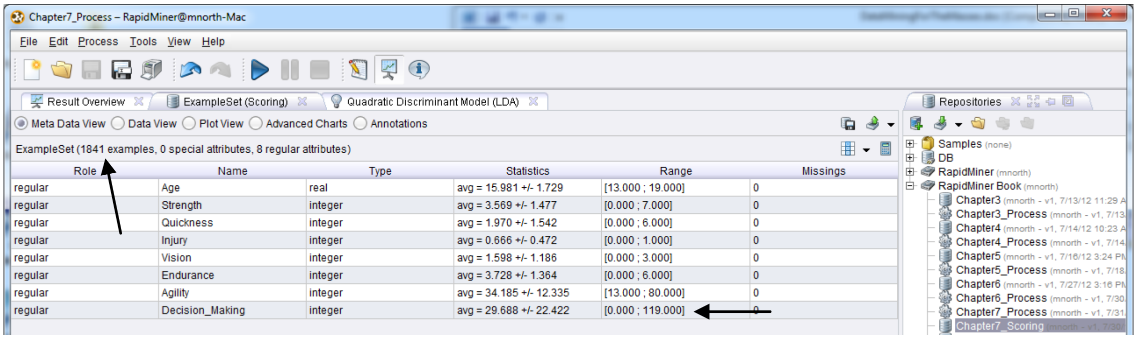
The root operator which is the outer most operator of every process.

Description

Each process must contain

Figure 7-9. Adding the scoring data set to our model.

10) Run the model again. RapidMiner will give you an additional tab in results perspective this time which will show the meta data for the scoring data set (Figure 7-10).



● Meta Data View ○ Data View ○ Plot View ○ Advanced Charts ○ Annotations

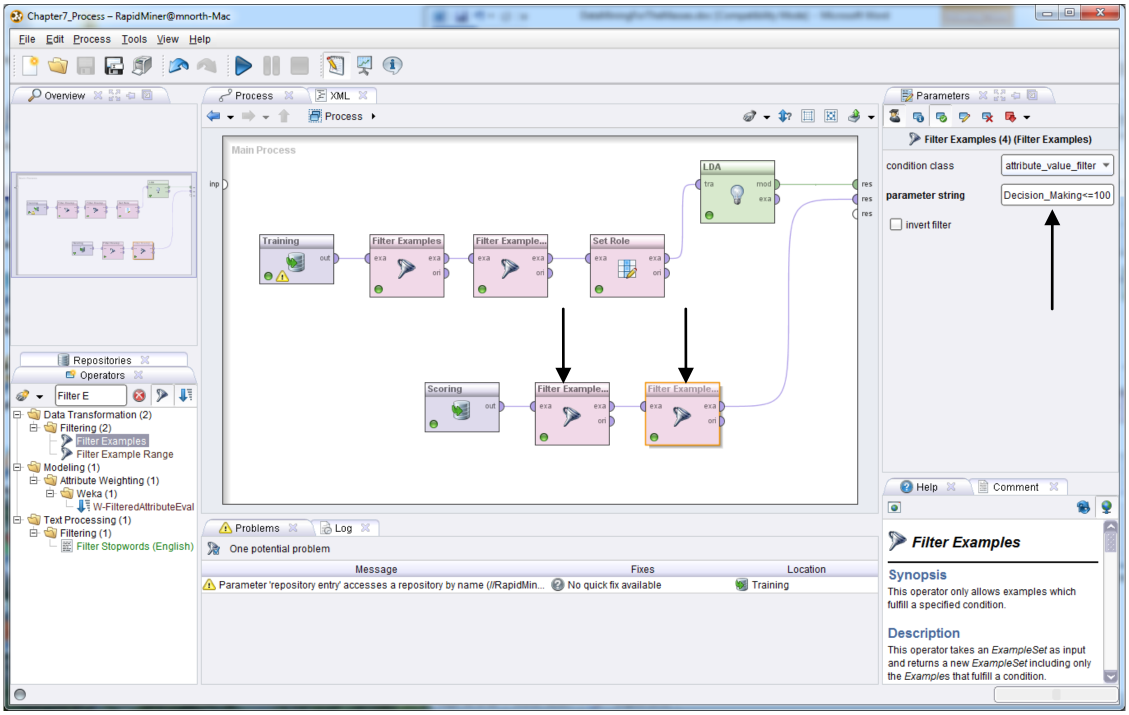
ExampleSet (1841 examples, 0 special attributes, 8 regular attributes)

Role	Name	Type	Statistics	Range	Missings
regular	Age	real	avg = 15.981 +/- 1.729	[13.000 ; 19.000]	0
regular	Strength	integer	avg = 3.569 +/- 1.477	[0.000 ; 7.000]	0
regular	Quickness	integer	avg = 1.970 +/- 1.542	[0.000 ; 6.000]	0
regular	Injury	integer	avg = 0.666 +/- 0.472	[0.000 ; 1.000]	0
regular	Vision	integer	avg = 1.598 +/- 1.186	[0.000 ; 3.000]	0
regular	Endurance	integer	avg = 3.728 +/- 1.364	[0.000 ; 6.000]	0
regular	Agility	integer	avg = 34.185 +/- 12.335	[13.000 ; 80.000]	0
regular	Decision_Making	integer	avg = 29.688 +/- 22.422	[0.000 ; 119.000]	0

Figure 7-10. Results perspective meta data for our scoring data set.

11) The scoring data set contains 1,841, however, as indicated by the black arrow in the Range column of Figure 7-10, the Decision_Making attribute has some inconsistent data again. Repeating the process previously outlined in steps 3 and 4, return to design perspective and use two consecutive Filter Examples operators to remove any observations that have values below 3 or above 100 in the Decision_Making attribute (Figure 7-11). This will

leave us with 1,767 observations, and you can check this by running the model again (Figure 7-12).



Filter Examples (4) (Filter Examples)

condition class: attribute_value_filter

parameter string: Decision_Making<=100

□ invert filter

One potential problem

Message	Fixes	Location
Parameter 'repository entry' accesses a repository by name (//RapidMin...	No quick fix available	Training

Filter Examples

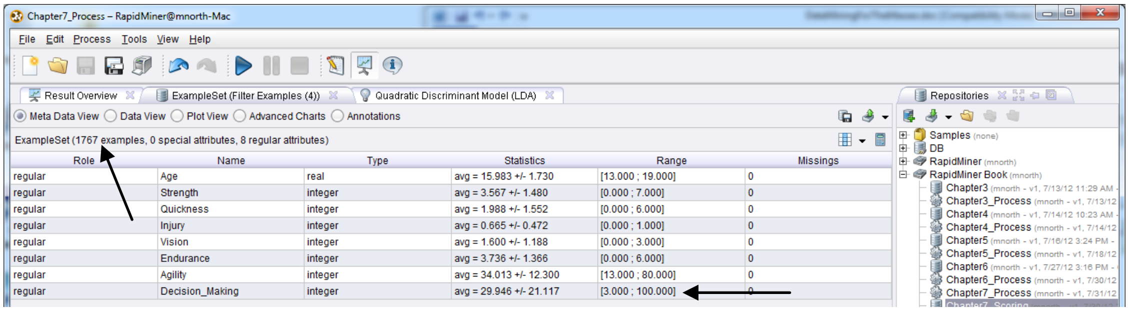
Synopsis

This operator only allows examples which fulfill a specified condition.

Description

This operator takes an *ExampleSet* as input and returns a new *ExampleSet* including only the *Examples* that fulfill a condition.

Figure 7-11. Filtering out observations containing inconsistent Decision_Making values.



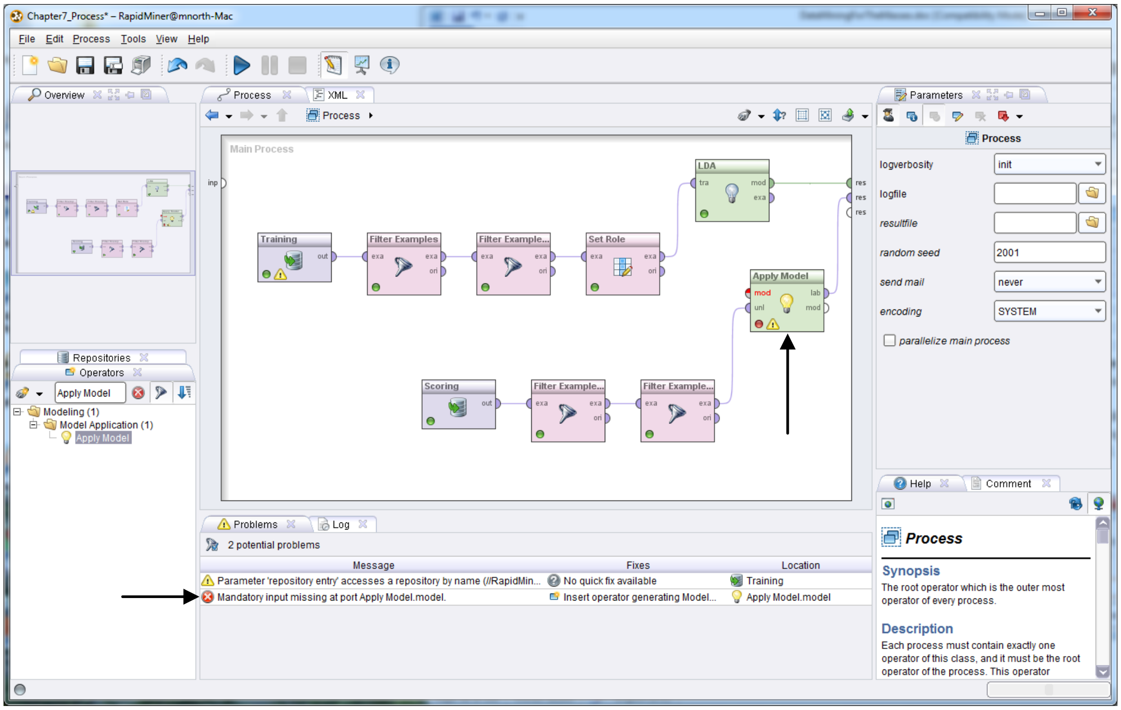
Chapter7_Process – RapidMiner@mnorth-Mac
File Edit Process Tools View Help
Result Overview
ExampleSet (Filter Examples (4))
Quadratic Discriminant Model (LDA)
● Meta Data View ○ Data View ○ Plot View ○ Advanced Charts ○ Annotations
Repositories
Samples
DB
RapidMiner
RapidMiner Book
Chapter3
Chapter3_Process
Chapter4
Chapter4_Process
Chapter5
Chapter5_Process
Chapter6
Chapter6_Process
Chapter7_Process

ExampleSet (1767 examples, 0 special attributes, 8 regular attributes)

Role	Name	Type	Statistics	Range	Missings
regular	Age	real	avg = 15.983 +/- 1.730	[13.000 ; 19.000]	0
regular	Strength	integer	avg = 3.567 +/- 1.480	[0.000 ; 7.000]	0
regular	Quickness	integer	avg = 1.988 +/- 1.552	[0.000 ; 6.000]	0
regular	Injury	integer	avg = 0.665 +/- 0.472	[0.000 ; 1.000]	0
regular	Vision	integer	avg = 1.600 +/- 1.188	[0.000 ; 3.000]	0
regular	Endurance	integer	avg = 3.736 +/- 1.366	[0.000 ; 6.000]	0
regular	Agility	integer	avg = 34.013 +/- 12.300	[13.000 ; 80.000]	0
regular	Decision_Making	integer	avg = 29.946 +/- 21.117	[3.000 ; 100.000]	0

Figure 7-12. Verification that observations with inconsistent values have been removed.

12) We now have just one step remaining to complete our model and predict the Prime_Sport for the 1,767 boys represented in our scoring data set. Return to design perspective, and use the search field in the Operators tab to locate an operator called Apply Model. Drag this operator over and place it in the Scoring data set's stream, as is shown in Figure 7-13.



2 potential problems

Message	Fixes	Location
Parameter 'repository entry' accesses a repository by name (//RapidMin...	No quick fix available	Training
Mandatory input missing at port Apply Model.model.	Insert operator generating Model...	Apply Model.model

Process

logverbosity: init
logfile:
resultfile:
random seed: 2001
send mail: never
encoding: SYSTEM
□ parallelize main process

Process

Synopsis

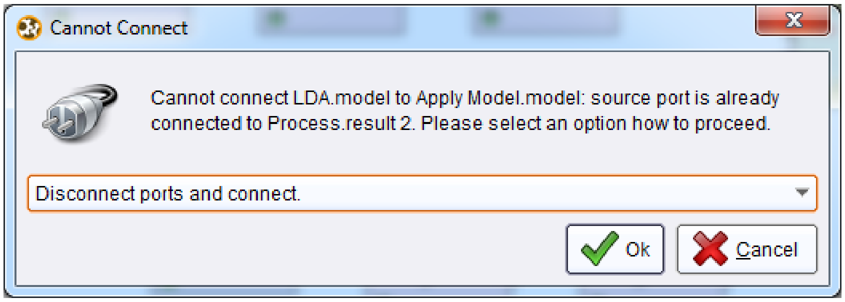
The root operator which is the outer most operator of every process.

Description

Each process must contain exactly one operator of this class, and it must be the root operator of the process. This operator

Figure 7-13. Adding the Apply Model operator to our Discriminant Analysis model.

13) As you can see in Figure 7-13, the Apply Model operator has given us an error. This is because the Apply Model operator expects the output of a model generation operator as its input. This is an easy fix, because our LDA operator (which generated a model for us) has a *mod* port for its output. We simply need to disconnect the LDA's *mod* port from the *res* port it's currently connected to, and connect it instead to the Apply Model operator's *mod* input port. To do this, click on the *mod* port for the LDA operator, and then click on the *mod* port for the Apply Model operator. When you do this, the following warning will pop up:

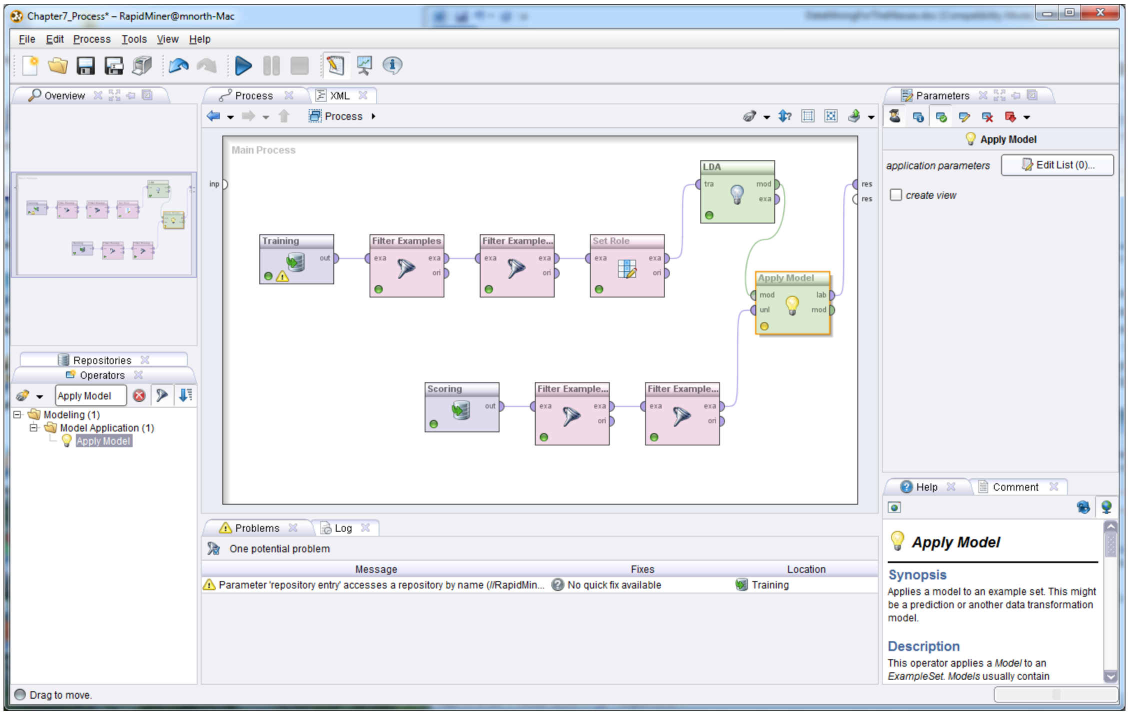


Cannot connect LDA.model to Apply Model.model: source port is already connected to Process.result 2. Please select an option how to proceed.

Disconnect ports and connect.

Figure 7-14. The port reconnection warning in RapidMiner.

14) Click OK to indicate to RapidMiner that you do in fact wish to reconfigure the spline to connect *mod* port to *mod* port. The error message will disappear and your scoring model will be ready for prediction (Figure 7-15).



Problems | Log

One potential problem

Message	Fixes	Location
Parameter 'repository entry' accesses a repository by name (//RapidMin...	No quick fix available	Training

Apply Model

Synopsis

Applies a model to an example set. This might be a prediction or another data transformation model.

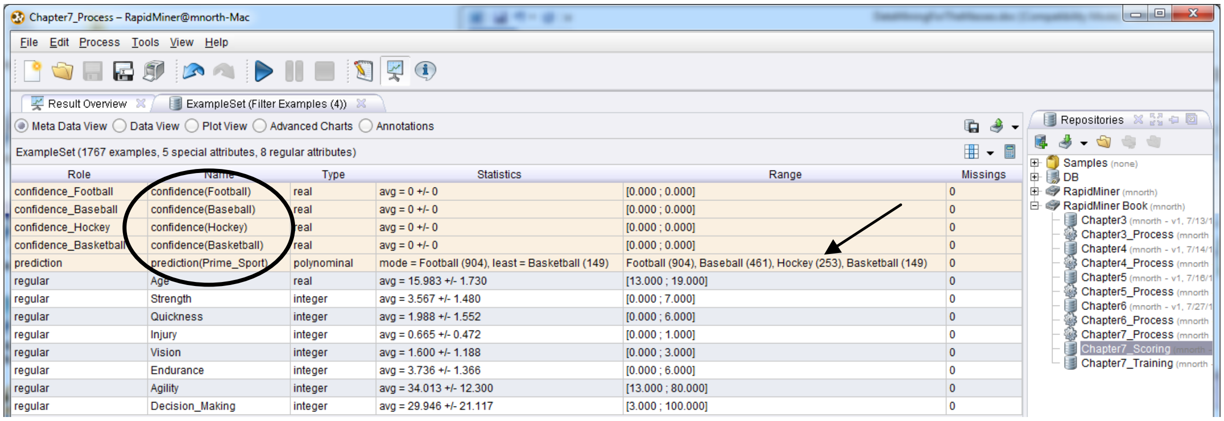
Description

This operator applies a *Model* to an *ExampleSet. Models* usually contain

Figure 7-15. Discriminant analysis model with training and scoring data streams.

15) Run the model by clicking the play button. RapidMiner will generate five new attributes and add them to our results perspective (Figure 7-16), preparing us for…

EVALUATION


Chapter7_Process – RapidMiner@mnorth-Mac
File Edit Process Tools View Help
Result Overview
ExampleSet (Filter Examples (4))
Repositories
Samples (none)
DB
RapidMiner (mnorth)
RapidMiner Book (mnorth)
Chapter3 (mnorth - v1, 7/13/1
Chapter3_Process (mnorth
Chapter4 (mnorth - v1, 7/14/1
Chapter4_Process (mnorth
Chapter5 (mnorth - v1, 7/16/1
Chapter5_Process (mnorth
Chapter6 (mnorth - v1, 7/27/1
Chapter6_Process (mnorth
Chapter7_Process (mnorth
Chapter7_Scoring (mnorth
Chapter7_Training (mnorth

● Meta Data View ○ Data View ○ Plot View ○ Advanced Charts ○ Annotations

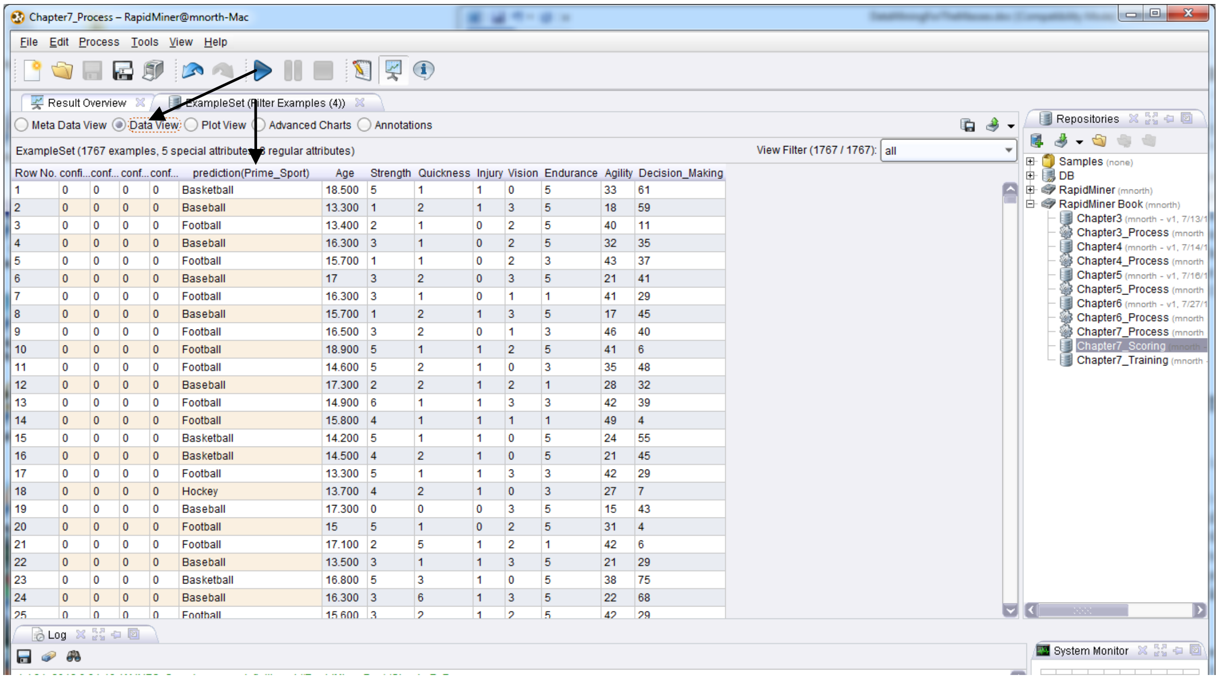
ExampleSet (1767 examples, 5 special attributes, 8 regular attributes)

Role	Name	Type	Statistics	Range	Missings
confidence_Football	confidence(Football)	real	avg = 0 +/- 0	[0.000 ; 0.000]	0
confidence_Baseball	confidence(Baseball)	real	avg = 0 +/- 0	[0.000 ; 0.000]	0
confidence_Hockey	confidence(Hockey)	real	avg = 0 +/- 0	[0.000 ; 0.000]	0
confidence_Basketball	confidence(Basketball)	real	avg = 0 +/- 0	[0.000 ; 0.000]	0
prediction	prediction(Prime_Sport)	polynominal	mode = Football (904), least = Basketball (149)	Football (904), Baseball (461), Hockey (253), Basketball (149)	0
regular	Age	real	avg = 15.983 +/- 1.730	[13.000 ; 19.000]	0
regular	Strength	integer	avg = 3.567 +/- 1.480	[0.000 ; 7.000]	0
regular	Quickness	integer	avg = 1.988 +/- 1.552	[0.000 ; 6.000]	0
regular	Injury	integer	avg = 0.665 +/- 0.472	[0.000 ; 1.000]	0
regular	Vision	integer	avg = 1.600 +/- 1.188	[0.000 ; 3.000]	0
regular	Endurance	integer	avg = 3.736 +/- 1.366	[0.000 ; 6.000]	0
regular	Agility	integer	avg = 34.013 +/- 12.300	[13.000 ; 80.000]	0
regular	Decision_Making	integer	avg = 29.946 +/- 21.117	[3.000 ; 100.000]	0

Figure 7-16. Prediction attributes generated by RapidMiner.

The first four attributes created by RapidMiner are confidence percentages, which indicate the relative strength of RapidMiner's prediction when compared to the other values the software might have predicted for each observation. In this example data set, RapidMiner has not generated confidence percentages for each of our four target sports. If RapidMiner had found some significant possibility that an observation might have more than one possible Prime_Sport, it would have calculated the percent probability that the person represented by an observation would succeed in one sport and in the others. For example, if an observation yielded a statistical possibility that the Prime_Sport for a person could have been any of the four, but Baseball was the strongest statistically, the confidence attributes on that observation might be: confidence(Football): 8%; confidence(Baseball): 69%; confidence(Hockey): 12%; confidence(Basketball): 11%. In some predictive data mining models (including some later in this text), your data *will* yield partial confidence percentages such as this. This phenomenon did not occur however in the data sets we used for this chapter's example. This is most likely explained by the fact discussed earlier in the chapter: all athletes will display some measure of aptitude in many sports, and so their battery test scores will likely be varied across the specializations. In statistical language, this is often referred to as **heterogeneity**.

Not finding confidence percentages does not mean that our experiment has been a failure however. The fifth new attribute, generated by RapidMiner when we applied our LDA model to our scoring data, is the prediction of Prime_Sport for each of our 1,767 boys. Click on the Data View radio button, and you will see that RapidMiner has applied our discriminant analysis model to our scoring data, resulting in a predicted Prime_Sport for each boy based on the specialization sport of previous academy attendees (Figure 7-17).



○ Meta Data View ● Data View ○ Plot View ○ Advanced Charts ○ Annotations

ExampleSet (1767 examples, 5 special attributes, 8 regular attributes)

View Filter (1767 / 1767): all

Row No.	confi...	conf...	conf...	conf...	prediction(Prime_Sport)	Age	Strength	Quickness	Injury	Vision	Endurance	Agility	Decision_Making
1	0	0	0	0	Basketball	18.500	5	1	1	0	5	33	61
2	0	0	0	0	Baseball	13.300	1	2	1	3	5	18	59
3	0	0	0	0	Football	13.400	2	1	0	2	5	40	11
4	0	0	0	0	Baseball	16.300	3	1	0	2	5	32	35
5	0	0	0	0	Football	15.700	1	1	0	2	3	43	37
6	0	0	0	0	Baseball	17	3	2	0	3	5	21	41
7	0	0	0	0	Football	16.300	3	1	0	1	1	41	29
8	0	0	0	0	Baseball	15.700	1	2	1	3	5	17	45
9	0	0	0	0	Football	16.500	3	2	0	1	3	46	40
10	0	0	0	0	Football	18.900	5	1	1	2	5	41	6
11	0	0	0	0	Football	14.600	5	2	1	0	3	35	48
12	0	0	0	0	Baseball	17.300	2	2	1	2	1	28	32
13	0	0	0	0	Football	14.900	6	1	1	3	3	42	39
14	0	0	0	0	Football	15.800	4	1	1	1	1	49	4
15	0	0	0	0	Basketball	14.200	5	1	1	0	5	24	55
16	0	0	0	0	Basketball	14.500	4	2	1	0	5	21	45
17	0	0	0	0	Football	13.300	5	1	1	3	3	42	29
18	0	0	0	0	Hockey	13.700	4	2	1	0	3	27	7
19	0	0	0	0	Baseball	17.300	0	0	0	3	5	15	43
20	0	0	0	0	Football	15	5	1	0	2	5	31	4
21	0	0	0	0	Football	17.100	2	5	1	2	1	42	6
22	0	0	0	0	Baseball	13.500	3	1	1	3	5	21	29
23	0	0	0	0	Basketball	16.800	5	3	1	0	5	38	75
24	0	0	0	0	Baseball	16.300	3	6	1	3	5	22	68
25	0	0	0	0	Football	15.600	3	2	1	2	5	42	29

Figure 7-17. Prime_Sport predictions for each boy in the scoring data set.

DEPLOYMENT

Gill now has a data set with a prediction for each boy that has been tested using the athletic battery at his academy. What to do with these predictions will be a matter of some thought and discussion. Gill can extract these data from RapidMiner and relate them back to each boy individually. For relatively small data sets, such as this one, we could move the results into a spreadsheet by simply copying and pasting them. Just as a quick exercise in moving results to other formats, try this:

1) Open a blank OpenOffice Calc spreadsheet.

2) In RapidMiner, click on the 1 under Row No. in Data View of results perspective (the cell will turn gray).

3) Press Ctrl+A (the keyboard command for 'select all' in Windows; you can use equivalent keyboard command for Mac or Linux as well). All cells in Data View will turn gray.

4) Press Ctrl+C (or the equivalent keyboard command for 'copy' if not using Windows).

5) In your blank OpenOffice Calc spreadsheet, right click in cell A1 and choose Paste Special… from the context menu.

6) In the pop up dialog box, select Unformatted Text, then click OK.

7) A Text Import pop up dialog box will appear with a preview of the RapidMiner data. Accept the defaults by clicking OK. The data will be pasted into the spreadsheet. The attribute names will have to be transcribed and added to the top row of the spreadsheet, but the data are now available outside of RapidMiner. Gill can match each prediction back to each boy in the scoring data set. The data are still in order, but remember that a few were removed because on inconsistent data, so care should be exercised when matching the predictions back to the boys represented by each observation. Bringing a unique identifying number into the training and scoring data sets might aid the matching once

predictions have been generated. This will be demonstrated in an upcoming chapter's example.

Chapter 14 of this book will spend some time talking about ethics in data mining. As previously mentioned, Gill's use of these predictions is going to require some thought and discussion. Is it ethical to push one of his young clients in the direction of one specific sport based on our model's prediction that that activity as a good match for the boy? Simply because previous academy attendees went on to specialize in one sport or another, can we assume that current clients would follow the same path? The final chapter will offer some suggestions for ways to answer such questions, but it is wise for us to at least consider them now in the context of the chapter examples.

It is likely that Gill, being experienced at working with young athletes and recognizing their strengths and weaknesses, will be able to use our predictions in an ethical way. Perhaps he can begin by grouping his clients by their predicted Prime_Sports and administering more 'sport-specific' drills—say, jumping tests for basketball, skating for hockey, throwing and catching for baseball, etc. This may allow him to capture more specific data on each athlete, or even to simply observe whether or not the predictions based on the data are in fact consistent with observable performance on the field, court, or ice. This is an excellent example of why the CRISP-DM approach is *cyclical*: the predictions we've generated for Gill are a starting point for a new round of assessment and evaluation, not the ending or culminating point. Discriminant analysis has given Gill some idea about where his young proteges may have strengths, and this can point him in certain directions when working with each of them, but he will inevitably gather more data and learn whether or not the use of this data mining methodology and approach is helpful in guiding his clients to a sport in which they might choose to specialize as they mature.

CHAPTER SUMMARY

Discriminant analysis helps us to cross the threshold between Classification and Prediction in data mining. Prior to Chapter 7, our data mining models and methodologies focused primarily on categorization of data. With Discriminant Analysis, we can take a process that is very similar in nature to k-means clustering, and with the right target attribute in a training data set, generate predictions for a scoring data set. This can become a powerful addition to k-means models, giving us the ability to apply our clusters to other data sets that haven't yet been classified.

Discriminant analysis can be useful where the classification for some observations is known and is not known for others. Some classic applications of discriminant analysis are in the fields of biology and organizational behavior. In biology, for example, discriminant analysis has been successfully applied to the classification of plant and animal species based on the traits of those living things. In organizational behavior, this type of data modeling has been used to help workers identify potentially successful career paths based on personality traits, preferences and aptitudes. By coupling known past performance with unknown but similarly structured data, we can use discriminant analysis to effectively train a model that can then score the unknown records for us, giving us a picture of what categories the unknown observations would likely be in.

REVIEW QUESTIONS

1) What type of attribute does a data set need in order to conduct discriminant analysis instead of k-means clustering?

2) What is a 'label' role in RapidMiner and why do you need an attribute with this role in order to conduct discriminant analysis?

3) What is the difference between a training data set and a scoring data set?

4) What is the purpose of the Apply Model operator in RapidMiner?

5) What are confidence percent attributes used for in RapidMiner? What was the likely reason that did we not find any in this chapter's example? Are there attributes about young athletes that you can think of that were not included in our data sets that might have helped up find some confidence percents? (Hint: think of things that are fairly specific to only one or two sports.)

6) What would be problematic about including both male and female athletes in this chapter's example data?

EXERCISE

For this chapter's exercise, you will compile your own data set based on people you know and the cars they drive, and then create a linear discriminant analysis of your data in order to predict categories for a scoring data set. 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 six attributes: Age, Gender, Marital_Status, Employment, Housing, and Car_Type.

3) Copy each of these attribute names except Car_Type into the Scoring sheet.

4) On the Training sheet, enter values for each of these attributes for several people that you know who have a car. 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, so long as each spouse has a different vehicle. Use the following to guide your data entry:
    a. For Age, you could put the person's actual age in years, or you could put them in buckets. For example, you could put 10 for people aged 10-19; 20 for people aged 20-29; etc.
    b. For Gender, enter 0 for female and 1 for male.
    c. For Marital_Status, use 0 for single, 1 for married, 2 for divorced, and 3 for widowed.
    d. For Employment, enter 0 for student, 1 for full-time, 2 for part-time, and 3 for retired.
    e. For Housing, use 0 for lives rent-free with someone else, 1 for rents housing, and 2 for owns housing.
    f. For Car_Type, you can record data in a number of ways. This will be your label, or the attribute you wish to predict. You could record each person's car by make (e.g.

Toyota, Honda, Ford, etc.), or you could record it by body style (e.g. Car, Truck, SUV, etc.). Be consistent in assigning classifications, and note that depending on the size of the data set you create, you won't want to have too many possible classificatons, or your predictions in the scoring data set will be spread out too much. With small data sets containing only 20-30 observations, the number of categories should be limited to three or four. You might even consider using Japanese, American, European as your Car_Types values.

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 people (more is better) that you know who *do not* have a car. You will use the training set to try to predict the type of car each of these people would drive if they had one.

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 respository. 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 Car_Type attribute as your label.

10) Add a Linear Discriminant Analysis operator to your Training stream.

11) Apply your LDA model to your scoring data and run your model. Evaluate and report your results. Did you get any confidence percentages? Do the predicted Car_Types seem reasonable and consistent with your training data? Why or why not?

Challenge Step!

12) Change your LDA operator to a different type of discriminant analysis (e.g. Quadratic) operator. Re-run your model. Consider doing some research to learn about the difference between linear and quadratic discriminant analysis. Compare your new results to the LDA results and report any interesting findings or differences.