CHAPTER SIX:
K-MEANS CLUSTERING

CONTEXT AND PERSPECTIVE

Sonia is a program director for a major health insurance provider. Recently she has been reading in medical journals and other articles, and found a strong emphasis on the influence of weight, gender and cholesterol on the development of coronary heart disease. The research she's read confirms time after time that there is a connection between these three variables, and while there is little that can be done about one's gender, there are certainly life choices that can be made to alter one's cholesterol and weight. She begins brainstorming ideas for her company to offer weight and cholesterol management programs to individuals who receive health insurance through her employer. As she considers where her efforts might be most effective, she finds herself wondering if there are natural groups of individuals who are most at risk for high weight and high cholesterol, and if there are such groups, where the natural dividing lines between the groups occur.

LEARNING OBJECTIVES

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

- Explain what k-means clusters are, how they are found and the benefits of using them.
- Recognize the necessary format for data in order to create k-means clusters.
- Develop a k-means cluster data mining model in RapidMiner.
- Interpret the clusters generated by a k-means model and explain their significance, if any.

ORGANIZATIONAL UNDERSTANDING

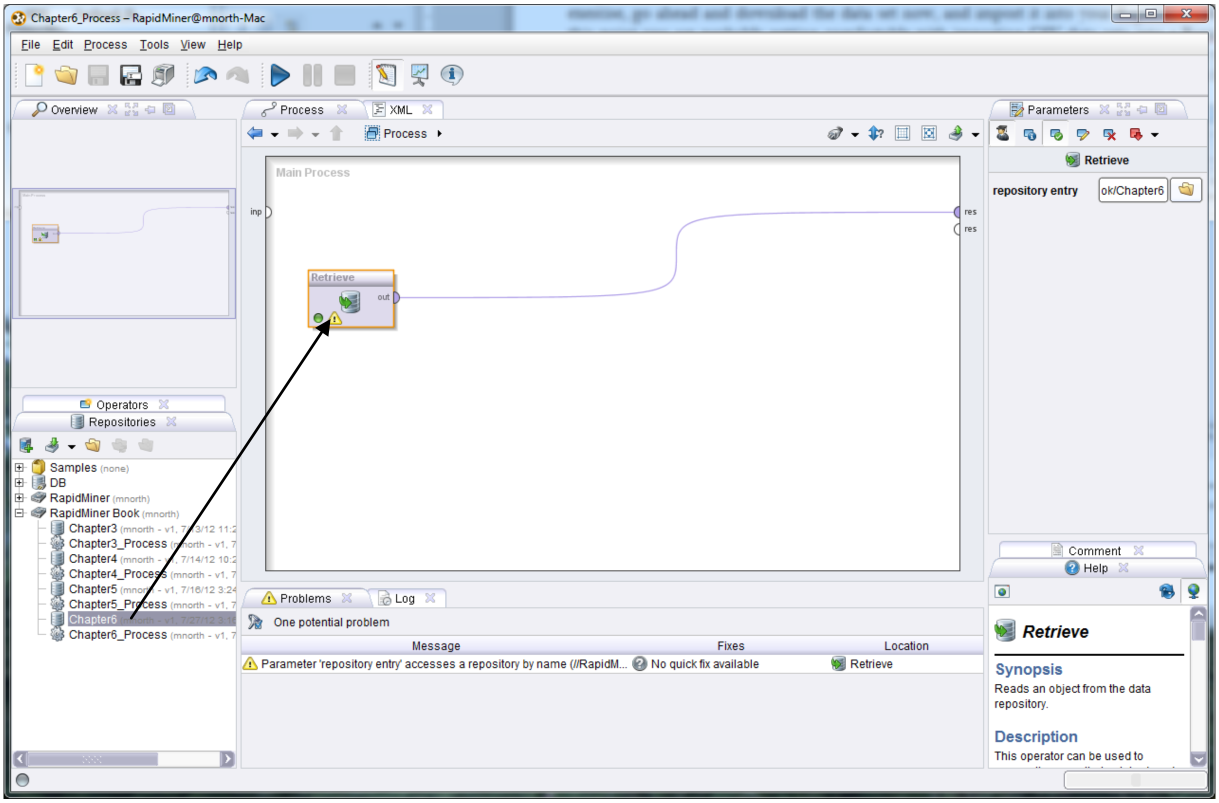
Sonia's goal is to identify and then try to reach out to individuals insured by her employer who are at high risk for coronary heart disease because of their weight and/or high cholesterol. She understands that those at low risk, that is, those with low weight and cholesterol, are unlikely to participate in the programs she will offer. She also understands that there are probably policy holders with high weight and low cholesterol, those with high weight *and* high cholesterol, and those with low weight and high cholesterol. She further recognizes there are likely to be a lot of people somewhere in between. In order to accomplish her goal, she needs to search among the thousands of policy holders to find groups of people with similar characteristics and craft programs and communications that will be relevant and appealing to people in these different groups.

DATA UNDERSTANDING

Using the insurance company's claims database, Sonia extracts three attributes for 547 randomly selected individuals. The three attributes are the insured's weight in pounds as recorded on the person's most recent medical examination, their last cholesterol level determined by blood work in their doctor's lab, and their gender. As is typical in many data sets, the gender attribute uses 0 to indicate Female and 1 to indicate Male. We will use this sample data from Sonia's employer's database to build a cluster model to help Sonia understand how her company's clients, the health insurance policy holders, appear to group together on the basis of their weights, genders and cholesterol levels. We should remember as we do this that means are particularly susceptible to undue influence by extreme outliers, so watching for inconsistent data when using the **k-Means clustering** data mining methodology is very important.

DATA PREPARATION

As with previous chapters, a data set has been prepared for this chapter's example, and is available as Chapter06DataSet.csv on the book's companion web site. If you would like to follow along with this example exercise, go ahead and download the data set now, and import it into your RapidMiner data repository. At this point you are probably getting comfortable with importing CSV data sets into a RapidMiner repository, but remember that the steps are outlined in Chapter 3 if you need to review them. Be sure to designate the attribute names correctly and to check your data types as you import. Once you have imported the data set, drag it into a new, blank process window so that you can begin to set up your k-means clustering data mining model. Your process should look like Figure 6-1.



One potential problem

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

Retrieve

Synopsis

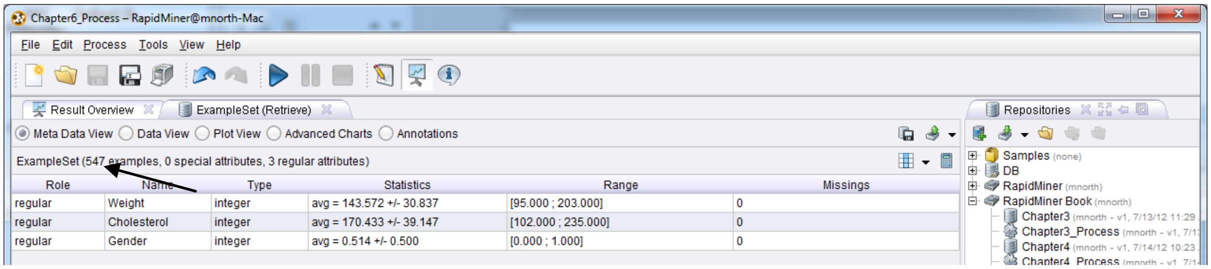
Reads an object from the data repository.

Description

This operator can be used to

Figure 6-1. Cholesterol, Weight and Gender data set added to a new process.

Go ahead and click the play button to run your model and examine the data set. In Figure 6-2 we can see that we have 547 observations across our three previously defined attributes. We can see the averages for each of the three attributes, along with their accompanying standard deviations and ranges. None of these values appear to be inconsistent (remember the earlier comments about using standard deviations to find statistical outliers). We have no missing values to handle, so our data appear to be very clean and ready to be mined.



ExampleSet (547 examples, 0 special attributes, 3 regular attributes)

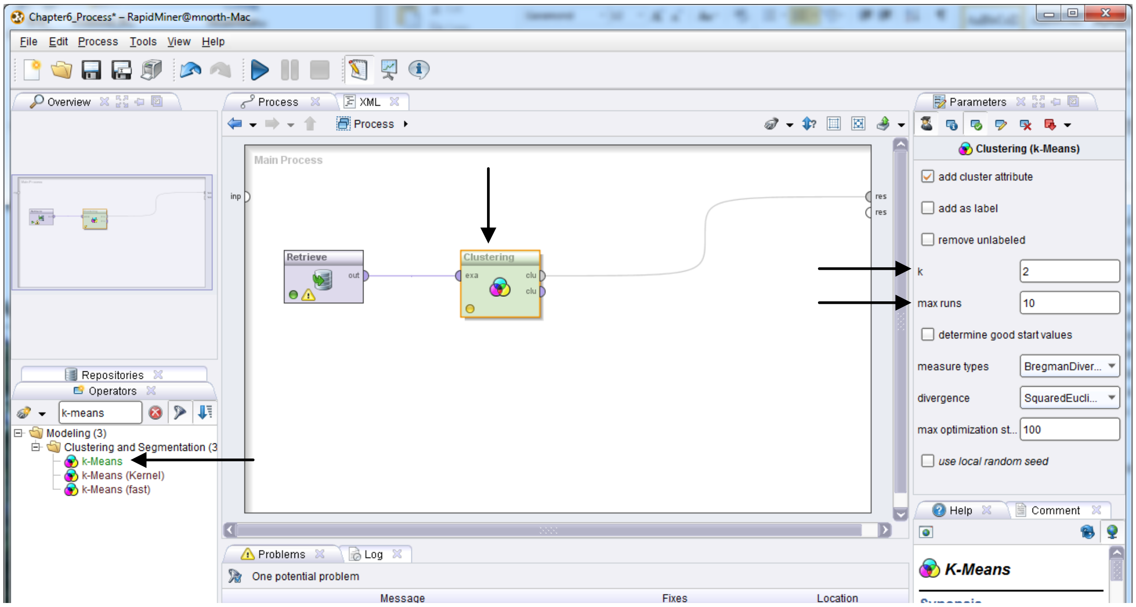
Role	Name	Type	Statistics	Range	Missings
regular	Weight	integer	avg = 143.572 +/- 30.837	[95.000 ; 203.000]	0
regular	Cholesterol	integer	avg = 170.433 +/- 39.147	[102.000 ; 235.000]	0
regular	Gender	integer	avg = 0.514 +/- 0.500	[0.000 ; 1.000]	0

Figure 6-2. A view of our data set's meta data.

MODELING

The '*k*' in k-means clustering stands for some number of groups, or clusters. The aim of this data mining methodology is to look at each observation's individual attribute values and compare them to the means, or in other words averages, of potential groups of other observations in order to find natural groups that are similar to one another. The k-means algorithm accomplishes this by sampling some set of observations in the data set, calculating the averages, or means, for each attribute for the observations in that sample, and then comparing the other attributes in the data set to that sample's means. The system does this repetitively in order to 'circle-in' on the best matches and then to formulate groups of observations which become the clusters. As the means calculated become more and more similar, clusters are formed, and each observation whose attributes values are most like the means of a cluster become members of that cluster. Using this process, k-means clustering models can sometimes take a long time to run, especially if you indicate a large number of "max runs" through the data, or if you seek for a large number of clusters (*k*). To build your k-means cluster model, complete the following steps:

1) Return to design view in RapidMiner if you have not done so already. In the operators search box, type k-means (be sure to include the hyphen). There are three operators that conduct k-means clustering work in RapidMiner. For this exercise, we will choose the first, which is simply named "k-Means". Drag this operator into your stream, and shown in Figure 6-3.



Operators search: k-means

- Modeling (3)
  - Clustering and Segmentation (3
    - k-Means
    - k-Means (Kernel)
    - k-Means (fast)

Clustering (k-Means)

☑ add cluster attribute

☐ add as label

☐ remove unlabeled

k: 2

max runs: 10

☐ determine good start values

measure types: BregmanDiver...

divergence: SquaredEucli...

max optimization st...: 100

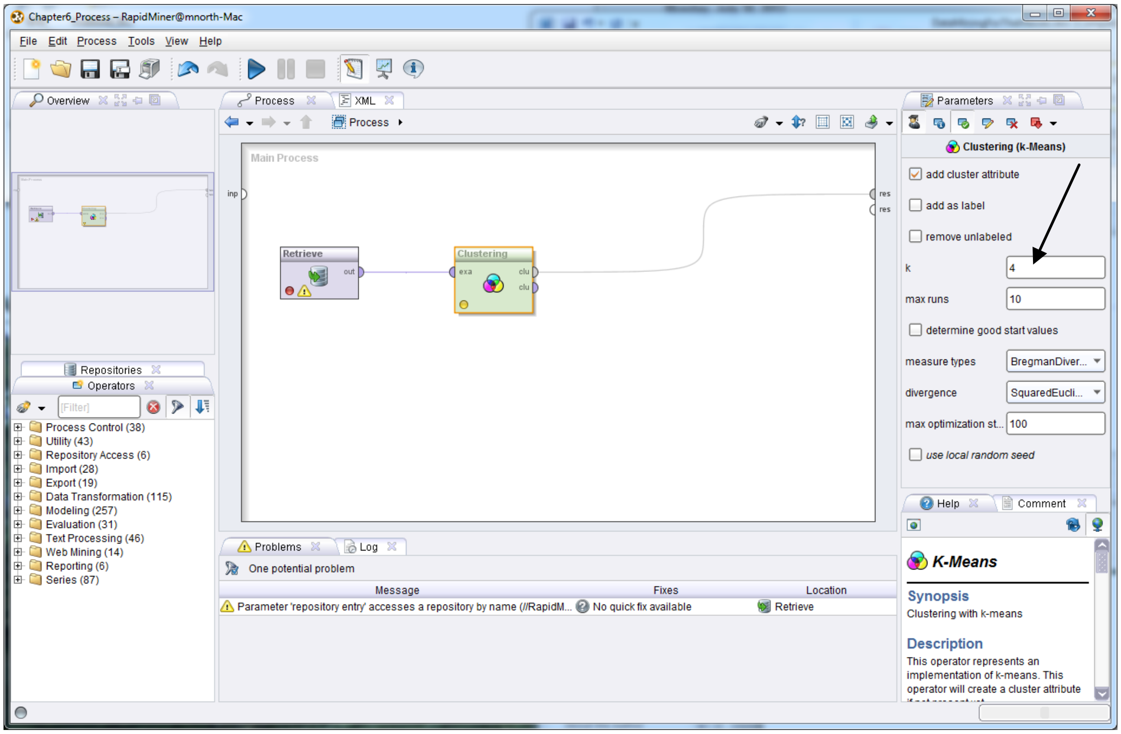
☐ *use local random seed*

One potential problem

K-Means

Figure 6-3. Adding the k-Means operator to our model.

2) Because we did not need to add any other operators in order to prepare our data for mining, our model in this exercise is very simple. We could, at this point, run our model and begin to interpret the results. This would not be very interesting however. This is because the default for our *k*, or our number of clusters, is 2, as indicated by the black arrow on the right hand side of Figure 6-3. This means we are asking RapidMiner to find only two clusters in our data. If we only wanted to find those with high and low levels of risk for coronary heart disease, two clusters would work. But as discussed in the Organizational Understanding section earlier in the chapter, Sonia has already recognized that there are likely a number of different types of groups to be considered. Simply splitting the data set into two clusters is probably not going to give Sonia the level of detail she seeks. Because Sonia felt that there were probably at least 4 potentially different groups, let's change the *k* value to four, as depicted in Figure 6-4. We could also increase of number of 'max runs', but for now, let's accept the default and run the model.



Clustering (k-Means)

☑ add cluster attribute

□ add as label

□ remove unlabeled

k: 4

max runs: 10

□ determine good start values

measure types: BregmanDiver...

divergence: SquaredEucli...

max optimization st...: 100

□ *use local random seed*

One potential problem

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

K-Means

Synopsis

Clustering with k-means

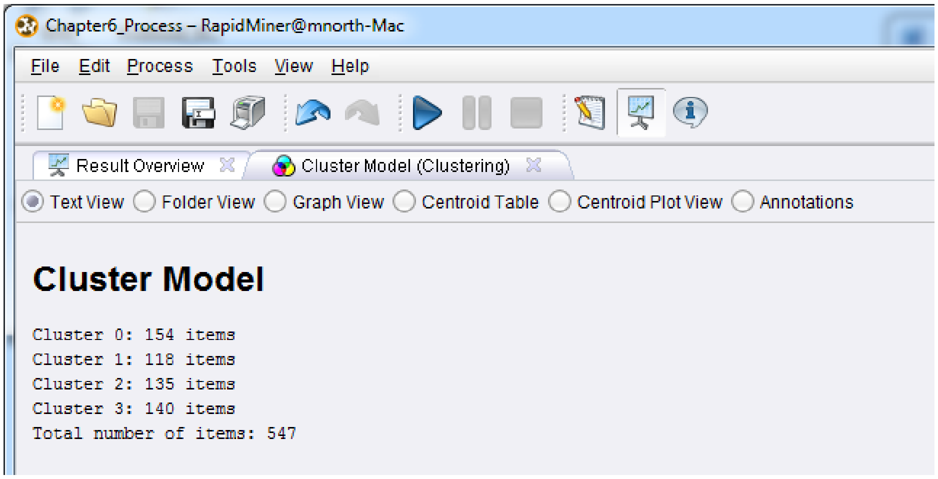
Description

This operator represents an implementation of k-means. This operator will create a cluster attribute

Figure 6-4. Setting the desired number of clusters for our model.

3) When the model is run, we find an initial report of the number of items that fell into each of our four clusters. (Note that the clustered are numbered starting from 0, a result of RapidMiner being written in the Java programming language.) In this particular model, our

clusters are fairly well balanced. While Cluster 1, with only 118 observations (Figure 6-5), is smaller than the other clusters, it is not unreasonably so.


Chapter6_Process – RapidMiner@mnorth-Mac
File Edit Process Tools View Help
Result Overview
Cluster Model (Clustering)
● Text View ○ Folder View ○ Graph View ○ Centroid Table ○ Centroid Plot View ○ Annotations

Cluster Model

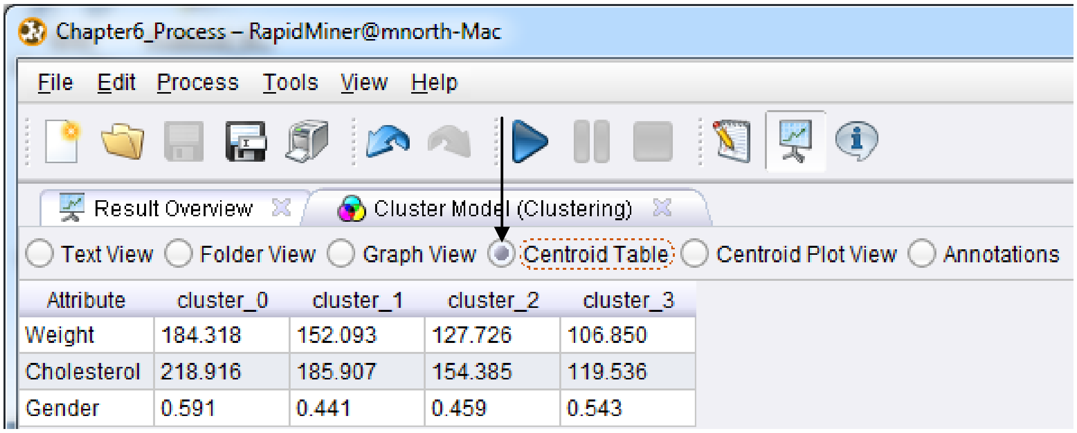
```
Cluster 0: 154 items
Cluster 1: 118 items
Cluster 2: 135 items
Cluster 3: 140 items
Total number of items: 547
```

Figure 6-5. The distribution of observations across our four clusters.

We could go back at this point and adjust our number of clusters, our number of 'max runs', or even experiment with the other parameters offered by the k-Means operator. There are other options for measurement type or divergence algorithms. Feel free to try out some of these options if you wish. As was the case with Association Rules, there may be some back and forth trial-and-error as you test different parameters to generate model output. When you are satisfied with your model parameters, you can proceed to…

EVALUATION

Recall that Sonia's major objective in the hypothetical scenario posed at the beginning of the chapter was to try to find natural breaks between different types of heart disease risk groups. Using the k-Means operator in RapidMiner, we have identified four clusters for Sonia, and we can now evaluate their usefulness in addressing Sonia's question. Refer back to Figure 6-5. There are a number of radio buttons which allow us to select options for analyzing our clusters. We will start by looking at our Centroid Table. This view of our results, shown in Figure 6-6, give the means for each attribute in each of the four clusters we created.

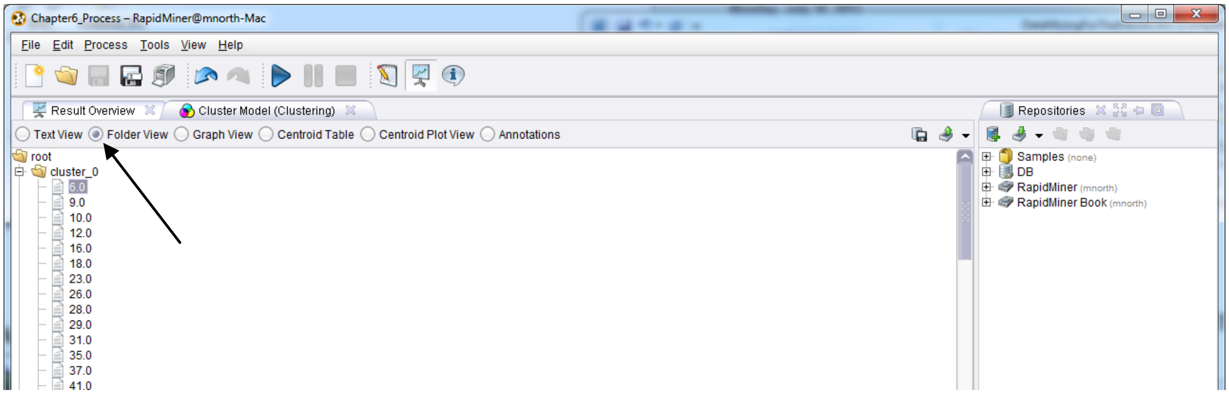

Chapter6_Process – RapidMiner@mnorth-Mac
File Edit Process Tools View Help
Result Overview
Cluster Model (Clustering)
○ Text View ○ Folder View ○ Graph View ● Centroid Table ○ Centroid Plot View ○ Annotations

Attribute	cluster_0	cluster_1	cluster_2	cluster_3
Weight	184.318	152.093	127.726	106.850
Cholesterol	218.916	185.907	154.385	119.536
Gender	0.591	0.441	0.459	0.543

Figure 6-6. The means for each attribute in our four (*k*) clusters.

We see in this view that cluster 0 has the highest average weight and cholesterol. With 0 representing Female and 1 representing Male, a mean of 0.591 indicates that we have more men than women represented in this cluster. Knowing that high cholesterol and weight are two key indicators of heart disease risk that policy holders can do something about, Sonia would likely want to start with the members of cluster 0 when promoting her new programs. She could then extend her programming to include the people in clusters 1 and 2, which have the next incrementally lower means for these two key risk factor attributes. You should note that in this chapter's example, the clusters' numeric order (0, 1, 2, 3) corresponds to decreasing means for each cluster. This is coincidental. Sometimes, depending on your data set, cluster 0 might have the highest means, but cluster 2 might have then next highest, so it's important to pay close attention to your centroid values whenever you generate clusters.

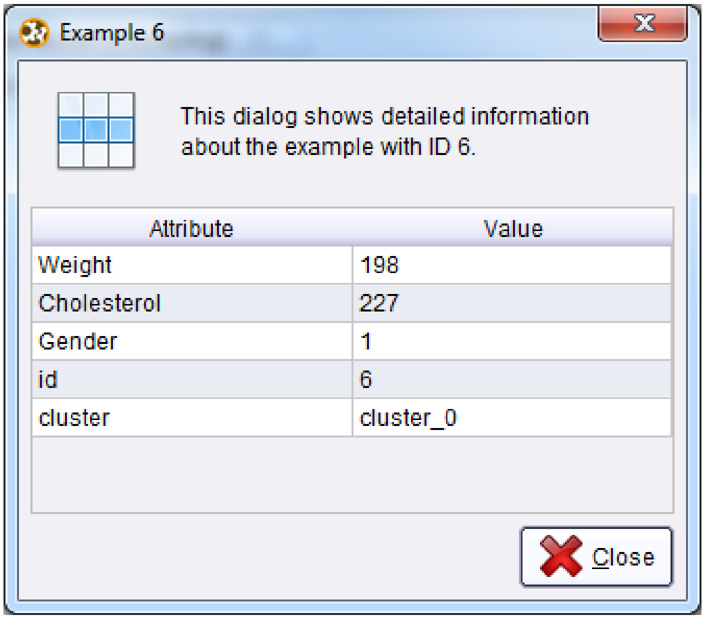
So we know that cluster 0 is where Sonia will likely focus her early efforts, but how does she know who to try to contact? Who are the members of this highest risk cluster? We can find this information by selecting the Folder View radio button. Folder View is depicted in Figure 6-7.


Chapter6_Process – RapidMiner@mnorth-Mac
File Edit Process Tools View Help
Result Overview
Cluster Model (Clustering)
○ Text View ● Folder View ○ Graph View ○ Centroid Table ○ Centroid Plot View ○ Annotations
Repositories
Samples (none)
DB
RapidMiner (mnorth)
RapidMiner Book (mnorth)

```
root
cluster_0
    6.0
    9.0
    10.0
    12.0
    16.0
    18.0
    23.0
    26.0
    28.0
    29.0
    31.0
    35.0
    37.0
    41.0
```

Figure 6-7. Folder view showing the observations included in Cluster 0.

By clicking the small + sign next to cluster 0 in Folder View, we can see all of the observations that have means which are similar to the mean for this cluster. Remember that these means are calculated for each attribute. You can see the details for any observation in the cluster by clicking on it. Figure 6-8 shows the results of clicking on observation 6 (6.0):



This dialog shows detailed information about the example with ID 6.

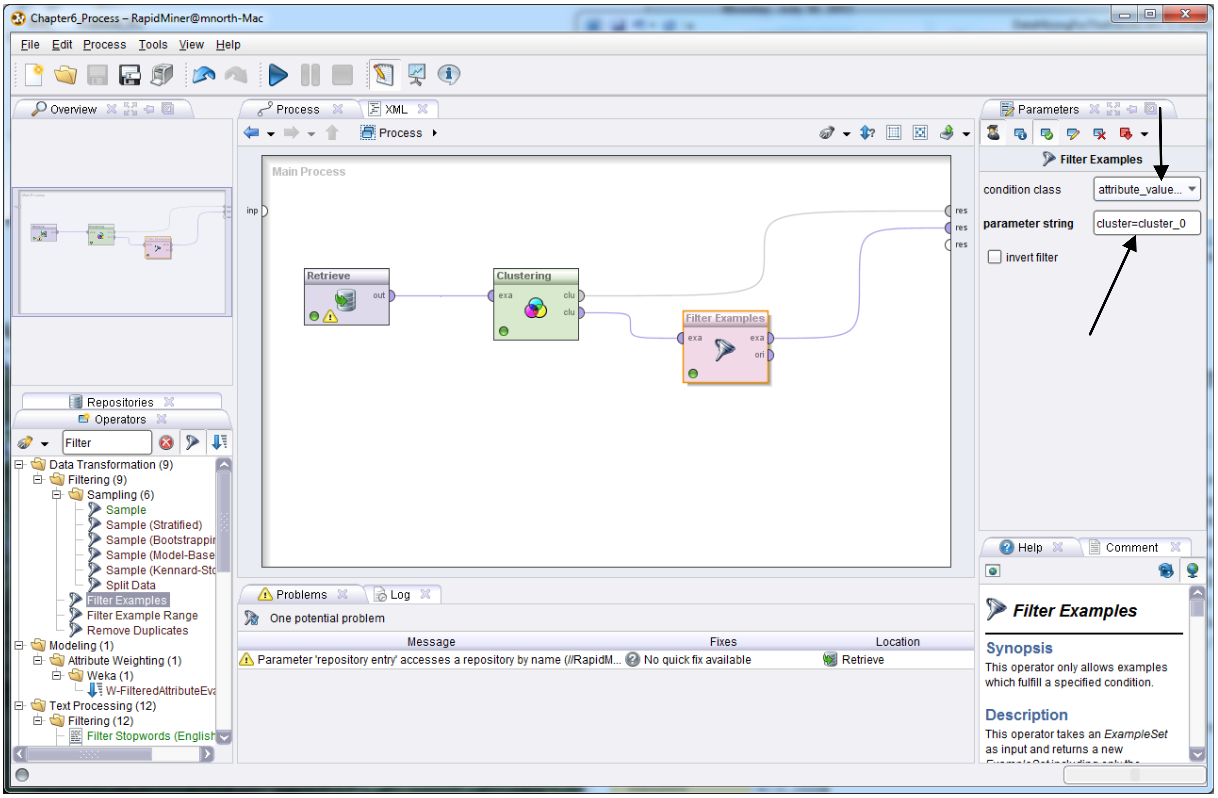
Attribute	Value
Weight	198
Cholesterol	227
Gender	1
id	6
cluster	cluster_0

Figure 6-8. The details of an observation within cluster 0.

The means for cluster 0 were just over 184 pounds for weight and just under 219 for cholesterol. The person represented in observation 6 is heavier and has higher cholesterol than the average for this highest risk group. Thus, this is a person Sonia is really hoping to help with her outreach program. But we know from the Centroid Table that there are 154 individuals in the data set who fall into this cluster. Clicking on each one of them in Folder View probably isn't the most efficient use of Sonia's time. Furthermore, we know from our Data Understanding paragraph earlier in this chapter that this model is built on only a sample data set of policy holders. Sonia might want to extract these attributes for all policy holders from the company's database and run the model again on that data set. Or, if she is satisfied that the sample has given her what she wants in terms of finding the breaks between the groups, she can move forward with…

DEPLOYMENT

We can help Sonia extract the observations from cluster 0 fairly quickly and easily. Return to design perspective in RapidMiner. Recall from Chapter 3 that we can filter out observations in our data set. In that chapter, we discussed filtering out observations as a Data Preparation step, but we can use the same operator in our Deployment as well. Using the search field in the Operators tab, locate the Filter Examples operator and connect it to your k-Means Clustering operator, as is depicted in Figure 6-9. Note that we have not disconnected the *clu* (cluster) port from the 'res' (result set) port, but rather, we have connected a second *clu* port to our *exa* port on the Filter Examples operator, and connected the *exa* port from Filter Examples to its own *res* port.



Filter Examples

condition class: attribute_value...

parameter string: cluster=cluster_0

☐ invert filter

One potential problem

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

Filter Examples

Synopsis

This operator only allows examples which fulfill a specified condition.

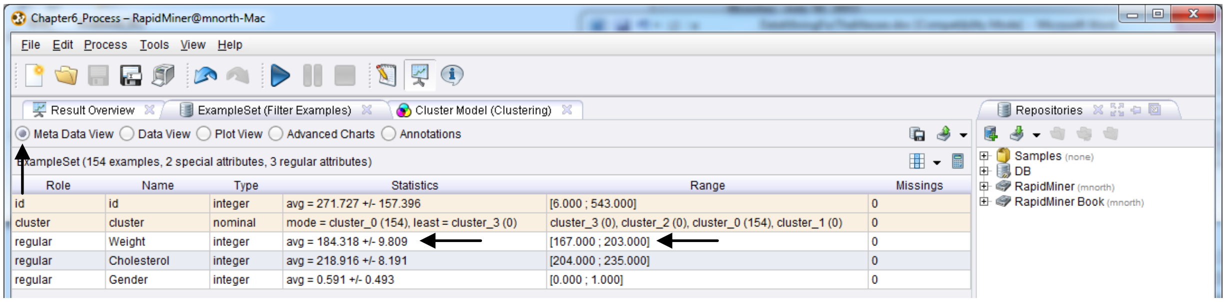
Description

This operator takes an *ExampleSet* as input and returns a new

Figure 6-9. Filtering our cluster model's output for only observations in cluster 0.

As indicated by the black arrows in Figure 6-9, we are filtering out our observations based on an attribute filter, using the parameter string cluster=cluster_0. This means that only those observations in the data set that are classified in the cluster_0 group will be retained. Go ahead and click the play button to run the model again.

You will see that we have not lost our Cluster Model tab. It is still available to us, but now we have added an ExampleSet tab, which contains only those 154 observations which fell into cluster 0. As with the result of previous models we've created, we have descriptive statistics for the various attributes in the data set.


Chapter6_Process – RapidMiner@mnorth-Mac
File Edit Process Tools View Help
Result Overview
ExampleSet (Filter Examples)
Cluster Model (Clustering)
Repositories
● Meta Data View ○ Data View ○ Plot View ○ Advanced Charts ○ Annotations
Samples (none)
DB
RapidMiner (mnorth)
RapidMiner Book (mnorth)

ExampleSet (154 examples, 2 special attributes, 3 regular attributes)

Role	Name	Type	Statistics	Range	Missings
id	id	integer	avg = 271.727 +/- 157.396	[6.000 ; 543.000]	0
cluster	cluster	nominal	mode = cluster_0 (154), least = cluster_3 (0)	cluster_3 (0), cluster_2 (0), cluster_0 (154), cluster_1 (0)	0
regular	Weight	integer	avg = 184.318 +/- 9.809	[167.000 ; 203.000]	0
regular	Cholesterol	integer	avg = 218.916 +/- 8.191	[204.000 ; 235.000]	0
regular	Gender	integer	avg = 0.591 +/- 0.493	[0.000 ; 1.000]	0

Figure 6-10. Filtered results for only cluster 0 observations.

Sonia could use these figures to begin contacting potential participants in her programs. With the high risk group having weights between 167 and 203 pounds, and cholesterol levels between 204 and 235 (these are taken from the Range statistics in Figure 6-10), she could return to her company's database and issue a SQL query like this one:

```
SELECT First_Name, Last_Name, Policy_Num, Address, Phone_Num
FROM PolicyHolders_view
WHERE Weight >= 167
AND Cholesterol >= 204;
```

This would give her the contact list for every person, male or female, insured by her employer who would fall into the higher risk group (cluster 0) in our data mining model. She could change the parameter criteria in our Filter Examples operator to be cluster=cluster_1 and re-run the model to get the descriptive statistics for those in the next highest risk group, and modify her SQL statement to get the contact list for that group from her organizational database; something akin to this query:

```
SELECT First_Name, Last_Name, Policy_Num, Address, Phone_Num
FROM PolicyHolders_view
WHERE (Weight >= 140 AND Weight <= 169)
AND (Cholesterol >= 168 AND Cholesterol <= 204);
```

If she wishes to also separate her groups by gender, she could add that criteria as well, such as "AND Gender = 1" in the WHERE clause of the SQL statement. As she continues to develop her health improvement programs, Sonia would have the lists of individuals that she most wants to target in the hopes of raising awareness, educating policy holders, and modifying behaviors that will lead to lower incidence of heart disease among her employer's clients.

CHAPTER SUMMARY

k-Means clustering is a data mining model that falls primarily on the side of Classification when referring to the Venn diagram from Chapter 1 (Figure 1-2). For this chapter's example, it does not necessarily predict which insurance policy holders *will* or *will not* develop heart disease. It simply takes known indicators from the attributes in a data set, and groups them together based on those attributes' similarity to group averages. Because any attributes that can be quantified can also have means calculated, k-means clustering provides an effective way of grouping observations together based on what is typical or normal for that group. It also helps us understand where one group begins and the other ends, or in other words, where the natural breaks occur between groups in a data set.

k-Means clustering is very flexible in its ability to group observations together. The k-Means operator in RapidMiner allows data miners to set the number of clusters they wish to generate, to dictate the number of sample means used to determine the clusters, and to use a number of different algorithms to evaluate means. While fairly simple in its set-up and definition, k-Means clustering is a powerful method for finding natural groups of observations in a data set.

REVIEW QUESTIONS

1) What does the *k* in k-Means clustering stand for?

2) How are clusters identified? What process does RapidMiner use to define clusters and place observations in a given cluster?

3) What does the Centroid Table tell the data miner? How do you interpret the values in a Centroid Table?

4) How do descriptive statistics aid in the process of evaluating and deploying a k-Means clustering model?

5) How might the presence of outliers in the attributes of a data set influence the usefulness of a k-Means clustering model? What could be done to address the problem?

EXERCISE

Think of an example of a problem that could be at least partially addressed by being able to group observations in a data set into clusters. Some examples might be grouping kids who might be at risk for delinquency, grouping product sale volumes, grouping workers by productivity and effectiveness, etc. Search the Internet or other resources available to you for a data set that would allow you to investigate your question using a k-means model. As with all exercises in this text, please ensure that you have permission to use any data set that might belong to your employer or another entity. When you have secured your data set, complete the following steps:

1) Ensure that your data set is saved as a CSV file. Import your data set into your RapidMiner repository and save it with a meaningful name. Drag it into a new process window in RapidMiner.

2) Conduct any data preparation that you need for your data set. This may include handling inconsistent data, dealing with missing values, or changing data types. Remember that in order to calculate means, each attribute in your data set will need to be numeric. If, for example, one of your attributes contains the values 'yes' and 'no', you may need to change these to be 1 and 0 respectively, in order for the k-Means operator to work.

3) Connect a k-Means operator to your data set, configure your parameters (especially set your *k* to something meaningful for your question) and then run your model.

4) Investigate your Centroid Table, Folder View, and the other evaluation tools.

5) Report your findings for your clusters. Discuss what is interesting about them and describe what iterations of modeling you went through, such as experimentation with different parameter values, to generate the clusters. Explain how your findings are relevant to your original question.

Challenge Step!

6) Experiment with the other k-Means operators in RapidMiner, such as Kernel or Fast. How are they different from your original model. Did the use of these operators change your clusters, and if so, how?