CHAPTER EIGHT:
LINEAR REGRESSION

CONTEXT AND PERSPECTIVE

Sarah, the regional sales manager from the Chapter 4 example, is back for more help. Business is booming, her sales team is signing up thousands of new clients, and she wants to be sure the company will be able to meet this new level of demand. She was so pleased with our assistance in finding correlations in her data, she now is hoping we can help her do some prediction as well. She knows that there is some correlation between the attributes in her data set (things like temperature, insulation, and occupant ages), and she's now wondering if she can use the data set from Chapter 4 to predict heating oil usage for new customers. You see, these new customers haven't begun consuming heating oil yet, there are a lot of them (42,650 to be exact), and she wants to know how much oil she needs to expect to keep in stock in order to meet these new customers' demand. Can she use data mining to examine household attributes and known past consumption quantities to anticipate and meet her new customers' needs?

LEARNING OBJECTIVES

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

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

ORGANIZATIONAL UNDERSTANDING

Sarah's new data mining objective is pretty clear: she wants to anticipate demand for a consumable product. We will use a **linear regression** model to help her with her desired predictions. She has data, 1,218 observations from the Chapter 4 data set that give an attribute profile for each home, along with those homes' annual heating oil consumption. She wants to use this data set as training data to predict the usage that 42,650 new clients will bring to her company. She knows that these new clients' homes are similar in nature to her existing client base, so the existing customers' usage behavior should serve as a solid gauge for predicting future usage by new customers.

DATA UNDERSTANDING

As a review, our data set from Chapter 4 contains the following attributes:

- **Insulation**: This is a density rating, ranging from one to ten, indicating the thickness of each home's insulation. A home with a density rating of one is poorly insulated, while a home with a density of ten has excellent insulation.
- **Temperature**: This is the average outdoor ambient temperature at each home for the most recent year, measure in degree Fahrenheit.
- **Heating_Oil**: This is the total number of units of heating oil purchased by the owner of each home in the most recent year.
- **Num_Occupants**: This is the total number of occupants living in each home.
- **Avg_Age**: This is the average age of those occupants.
- **Home_Size**: This is a rating, on a scale of one to eight, of the home's overall size. The higher the number, the larger the home.

We will use the Chapter 4 data set as our training data set in this chapter. Sarah has assembled a separate Comma Separated Values file containing all of these same attributes, except of course for Heating_Oil, for her 42,650 new clients. She has provided this data set to us to use as the scoring data set in our model.

DATA PREPARATION

You should already have downloaded and imported the Chapter 4 data set, but if not, you can get it from the book's companion web site (https://sites.google.com/site/dataminingforthemasses/). Download and import the Chapter 8 data set from the companion web site as well. Once you have both the Chapter 4 and Chapter 8 data sets imported into your RapidMiner data repository, complete the following steps:

1) Drag and drop both data sets into a new process window in RapidMiner. Rename the Chapter 4 data set to 'Training (CH4), and the Chapter 8 data set to 'Scoring (CH8)'. Connect both *out* ports to *res* ports, as shown in Figure 8-1, and then run your model.

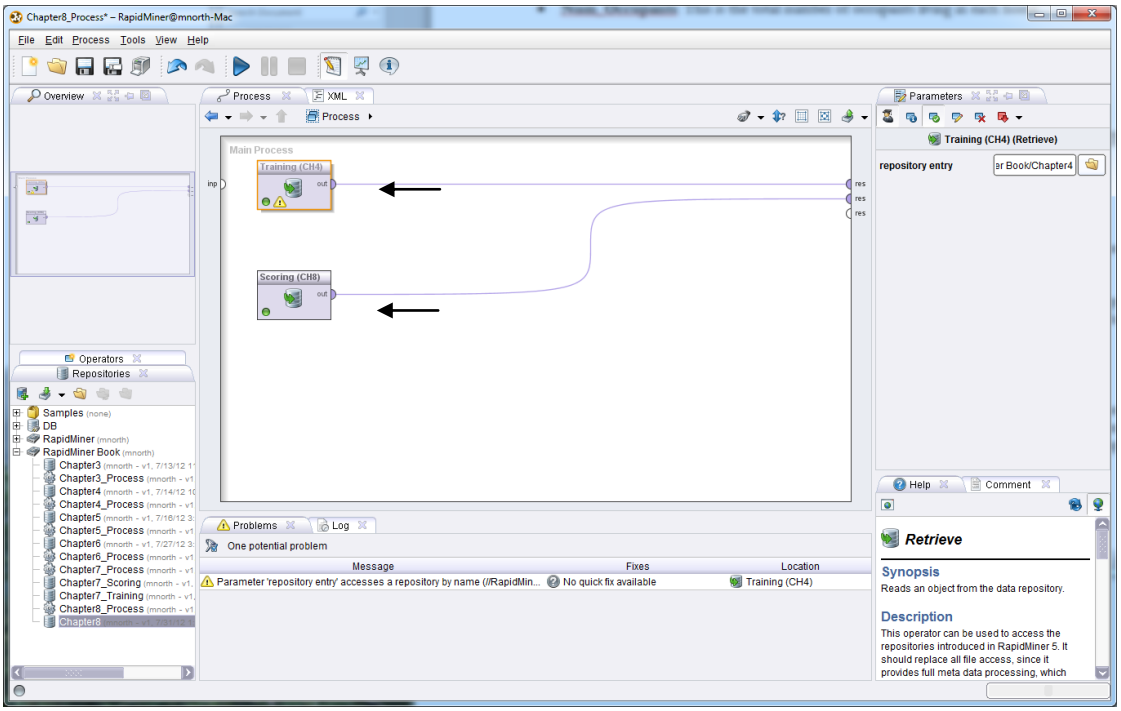


Figure 8-1. Using both Chapter 4 and 8 data sets to set up a linear regression model.

2) Figures 8-2 and 8-3 show side-by-side comparisons of the training and scoring data sets. When using linear regression as a predictive model, it is extremely important to remember that the ranges for all attributes in the scoring data must be within the ranges for the corresponding attributes in the training data. This is because a training data set cannot be relied upon to predict a target attrtibute for observations whose values fall outside the training data set's values.

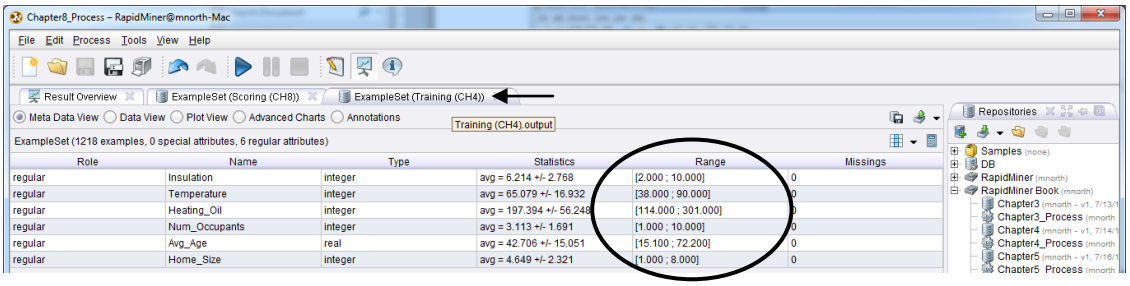


Figure 8-2. Value ranges for the training data set's attributes.

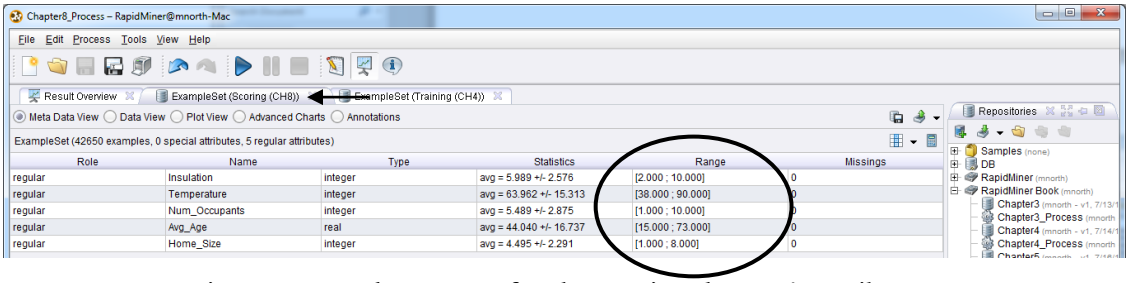


Figure 8-3. Value ranges for the scoring data set's attributes.

3) We can see that in comparing Figures 8-2 and 8-3, the ranges are the same for all attributes except Avg_Age. In the scoring data set, we have some observations where the Avg_Age is slightly below the training data set's lower bound of 15.1, and some observations where the scoring Avg_Age is slightly above the training set's upper bound of 72.2. You might think that these values are so close to the training data set's values that it would not matter if we used our training data set to predict heating oil usage for the homes represented by these observations. While it is likely that such a slight deviation from the range on this attribute would not yield wildly inaccurate results, we cannot use linear regression prediction values as evidence to support such an assumption. Thus, we will need to remove these observations from our data set. Add two Filter Examples operators with the parameters attribute_value_filter and Avg_Age>=15.1 | Avg_Age <=72.2. When you run your model now, you should have 42,042 observations remaining. Check the ranges again to ensure that none of the scoring attributes now have ranges outside those of the training attributes. Then return to design perspective.

4) As was the case with discriminant analysis, linear regression is a predictive model, and thus will need an attribute to be designated as the label—this is the target, the thing we want to predict. Search for the Set Role operator in the Operators tab and drag it into your training

stream. Change the parameters to designate Heating_Oil as the label for this model (Figure 8-4).

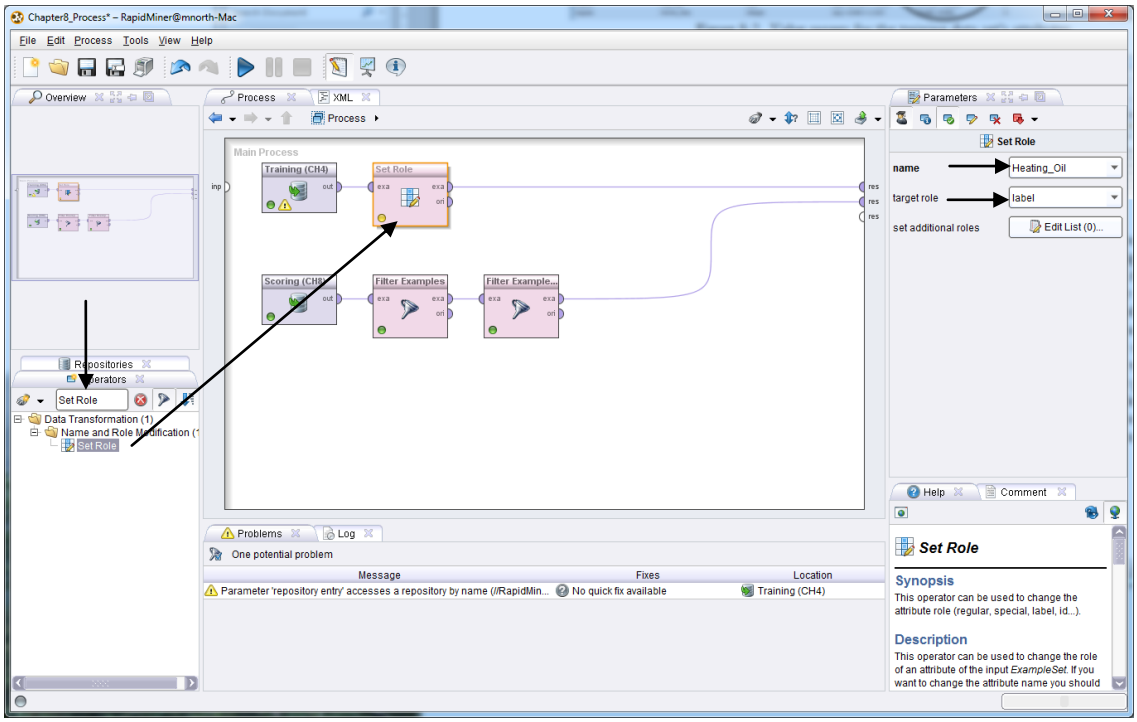


Figure 8-4. Adding an operator to designate Heating_Oil as our label.

With this step complete our data sets are now prepared for…

MODELING

5) Using the search field in the Operators tab again, locate the Linear Regression operator and drag and drop it into your training data set's stream (Figure 8-5).

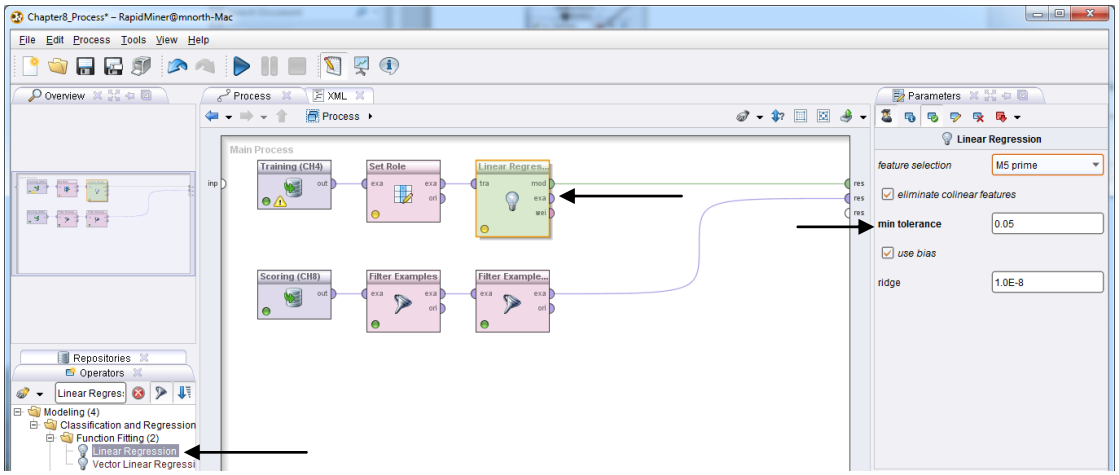


Figure 8-5. Adding the Linear Regression model operator to our stream.

6) Note that the Linear Regression operator uses a default tolerance of .05 (also known in statistical language as the **confidence level** or **alpha level**). This value of .05 is very common in statistical analysis of this type, so we will accept this default. The final step to complete our model is to use an Apply Model operator to connect our training stream to our scoring stream. Be sure to connect both the *lab* and *mod* ports coming from the Apply Model operator to *res* ports. This is illustrated in Figure 8-6.

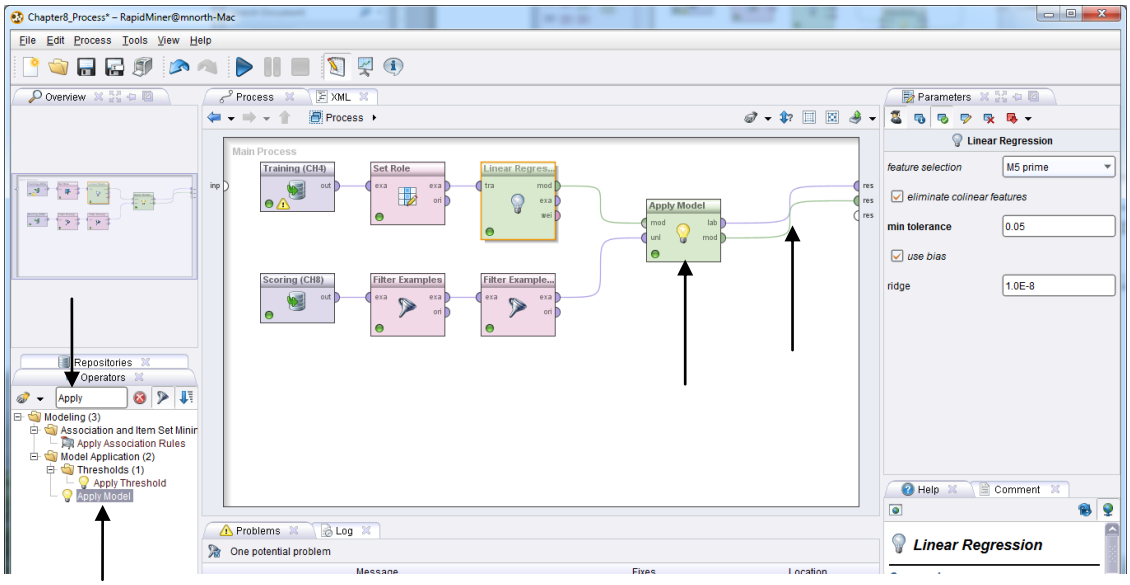
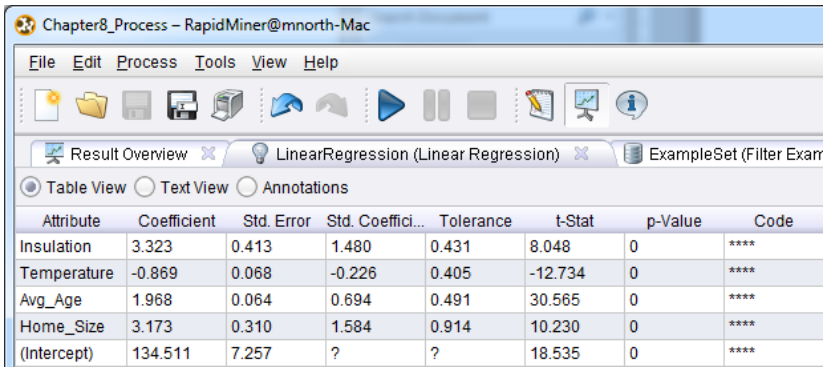


Figure 8-6. Applying the model to the scoring data set.

7) Run the model. Having two splines coming from the Apply Model operator and connecting to *res* ports will result in two tabs in results perspective. Let's examine the LinearRegression tab first, as we begin our…

EVALUATION



Attribute	Coefficient	Std. Error	Std. Coeffici...	Tolerance	t-Stat	p-Value	Code
Insulation	3.323	0.413	1.480	0.431	8.048	0	****
Temperature	-0.869	0.068	-0.226	0.405	-12.734	0	****
Avg_Age	1.968	0.064	0.694	0.491	30.565	0	****
Home_Size	3.173	0.310	1.584	0.914	10.230	0	****
(Intercept)	134.511	7.257	?	?	18.535	0	****

Figure 8-7. Linear regression coefficients.

Linear regression modeling is all about determing how close a given observation is to an imaginary line representing the average, or center of all points in the data set. That imaginary line gives us the first part of the term "linear regression". The formula for calculating a prediction using linear regression is $y=mx+b$. You may recognize this from a former algebra class as the formula for calculating the slope of a line. In this formula, the variable y, is the target, the label, the thing we want to predict. So in this chapter's example, y is the amount of Heating_Oil we expect each home to consume. But how will we predict y? We need to know what m, x, and b are. The variable m is the value for a given predictor attribute, or what is sometimes referred to as an **independent variable**. Insulation, for example, is a predictor of heating oil usage, so Insulation is a predictor attribute. The variable x is that attribute's coefficient, shown in the second column of Figure 8-7. The coefficient is the amount of weight the attribute is given in the formula. Insulation, with a coefficient of 3.323, is weighted heavier than any of the other predictor attributes in this data set. Each observation will have its Insulation value multipled by the Insulation coefficient to properly weight that attribute when calculating y (heating oil usage). The variable b is a constant that is added to all linear regression calculations. It is represented by the Intercept, shown in figure 8-7 as 134.511. So suppose we had a house with insulation density of 5; our formula using these Insulation values would be $y=(5*3.323)+134.511$.

But wait! We had more than one predictor attribute. We started out using a combination of five attributes to try to predict heating oil usage. The formula described in the previous paragraph only uses one. Furthermore, our LinearRegression result set tab pictured in Figure 8-7 only has four predictor variables. What happened to Num_Occupants?

The answer to the latter question is that Num_Occupants was not a **statistically significant** predictor of heating oil usage in this data set, and therefore, RapidMiner removed it as a predictor. In other words, when RapidMiner evaluated the amount of influence each attribute in the data set had on heating oil usage for each home represented in the training data set, the number of occupants was so non-influential that its weight in the formula was set to zero. An example of why this might occur could be that two older people living in a house may use the same amount of heating oil as a young family of five in the house. The older couple might take longer showers, and prefer to keep their house much warmer in the winter time than would the young family. The variability in the number of occupants in the house doesn't help to explain each home's heating oil usage very well, and so it was removed as a predictor in our model.

But what about the former question, the one about having multiple independent variables in this model? How can we set up our linear formula when we have multiple predictors? This is done by using the formula: $y=mx+mx+mx\ldots+b$. Let's take an example. Suppose we wanted to predict heating oil usage, using our model, for a home with the following attributes:

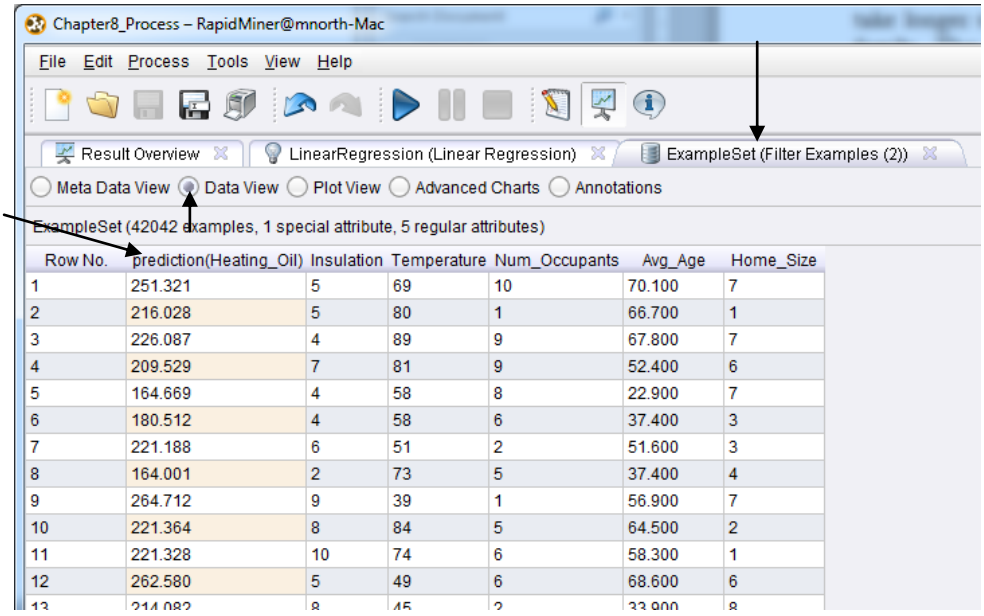
- Insulation: 6
- Temperature: 67
- Avg_Age: 35.4
- Home_Size: 5

Our formula for this home would be: $y=(6*3.323)+(67*-0.869)+(35.4*1.968)+(5*3.173)+134.511$

Our prediction for this home's annual number of heating oil units ordered (y) is 181.758, or basically 182 units. Let's check our model's predictions as we discuss possibilities for…

DEPLOYMENT

While still in results perspective, switch to the ExampleSet tab, and select the Data View radio button. We can see in this view (Figure 8-8) that RapidMiner has quickly and efficiently predicted the number of units of heating oil each of Sarah's company's new customers will likely use in their first year. This is seen in the prediction(Heating_Oil) attribute.



Row No.	prediction(Heating_Oil)	Insulation	Temperature	Num_Occupants	Avg_Age	Home_Size
1	251.321	5	69	10	70.100	7
2	216.028	5	80	1	66.700	1
3	226.087	4	89	9	67.800	7
4	209.529	7	81	9	52.400	6
5	164.669	4	58	8	22.900	7
6	180.512	4	58	6	37.400	3
7	221.188	6	51	2	51.600	3
8	164.001	2	73	5	37.400	4
9	264.712	9	39	1	56.900	7
10	221.364	8	84	5	64.500	2
11	221.328	10	74	6	58.300	1
12	262.580	5	49	6	68.600	6
13	214.082	8	45	2	33.900	8

Figure 8-8. Heating oil predictions for 42,042 new clients.

Let's check the first of our 42,042 households by running the linear regression formula for row 1:

$$(5*3.323)+(69*-0.869)+(70.1*1.968)+(7*3.173)+134.511 = 251.321$$

Note that in this formula we skipped the Num_Occupants attribute because it is not predictive. The formula's result does indeed match RapidMiner's prediction for this home. Sarah now has a prediction for each of the new clients' homes, with the exception of those that had Avg_Age values that were out of range. How might Sarah use this data? She could start by summing the prediction attribute. This will tell her the total new units of heating oil her company is going to need to be able to provide in the coming year. This can be accomplished by exporting her data to a spreasheet and summing the column, or it can even be done within RapidMiner using an Aggregate operator. We will demonstrate this briefly.

1) Switch back to design perspective.

2) Search for the Aggreate operator in the Operators tab and add it between the *lab* and *res* ports, as shown in Figure 8-9. It is not depicted in Figure 8-9, but if you wish to generate a tab in results perspective that shows all of your obsevations and their predictions, you can connect the *ori* port on the Aggregate operator to a *res* port.

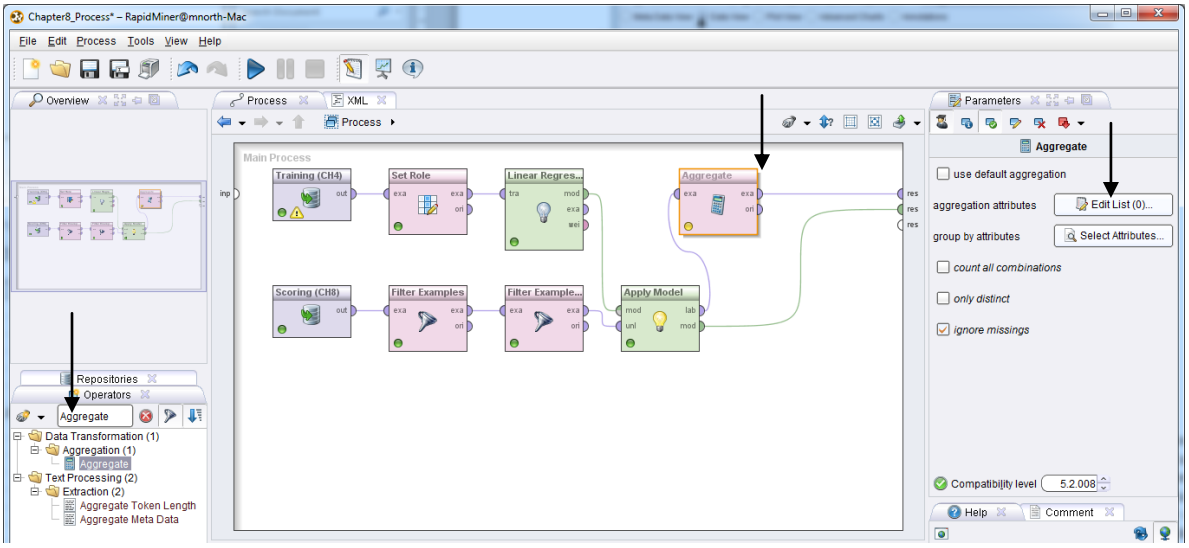


Figure 8-9. Adding an Aggregate operator to our linear regression model.

3) Click on the Edit List button. A window similar to Figure 8-10 will appear. Set the prediction(Heating_Oil) attribute as the aggregation attribute, and the aggregation function to 'sum'. If you would like you can add other aggretations. In the Figure 8-10 example, we have added an average for prediction(Heating_Oil) as well.

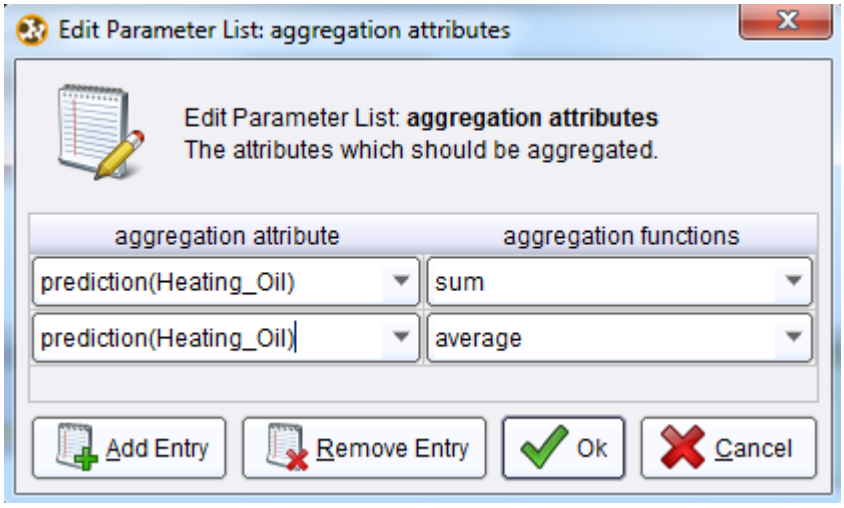


Figure 8-10. Configuring aggregations in RapidMiner.

4) When you are satisfied with your aggregations, click OK to return to your main process window, then run the model. In results perspective, select the ExampleSet(Aggregate) tab, then select the Data View radio button. The sum and average for the prediction attribute will be shown, as depicted in Figure 8-11.

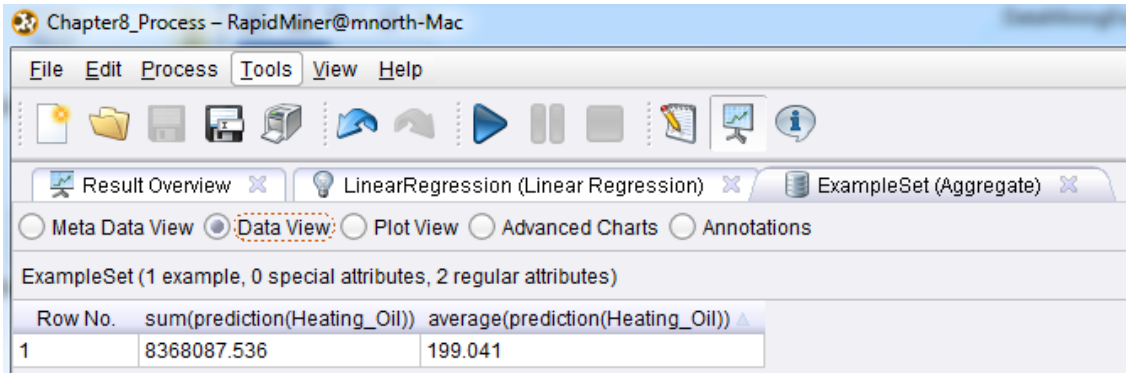


Figure 8-11. Aggregate descriptive statistics for our predicted attribute.

From this image, we can see that Sarah's company is likely to sell some 8,368,088 units of heating oil to these new customers. The company can expect that on average, their new customers will order about 200 units each. These figures are for all 42,042 clients together, but Sarah is probably going to be more interested in regional trends. In order to deploy this model to help her more specifically address her new customers' needs, she should probably extract the predictions, match

them back to their source records which might contain the new clients' addresses, enabling her to break the predictions down by city, county, or region of the country. Sarah could then work with her colleagues in Operations and Order Fulfillment to ensure that regional heating oil distribution centers around the country have appropriate amounts of stock on hand to meet anticipated need. If Sarah wanted to get even more granular in her analysis of these data, she could break her training and scoring datas set down into months using a month attribute, and then run the predictions again to reveal fluctuations in usuage throughout the course of the year.

CHAPTER SUMMARY

Linear regression is a predictive model that uses training and scoring data sets to generate numeric predictions in data. It is important to remember that linear regression uses numeric data types for all of its attributes. It uses the algebraic formula for calculating the slope of a line to determine where an observation would fall along an imaginary line through the scoring data. Each attribute in the data set is evaluated statistically for its ability to predict the target attribute. Attributes that are not strong predictors are removed from the model. Those attributes that are good predictors are assigned coefficients which give them weight in the prediction formula. Any observations whose attribute values fall in the range of corresponding training attribute values can be plugged into the formula in order to predict the target.

Once linear regression predictions are calculated, the results can be summarized in order to determine if there are differences in the predictions in subsets of the scoring data. As more data are collected, they can be added into the training data set in order to create a more robust training data set, or to expand the ranges of some attributes to include even more values. It is very important to remember that the ranges for the scoring attributes must fall within the ranges for the training attributes in order to ensure valid predictions.

REVIEW QUESTIONS

1) What data type does linear regression expect for all attributes? What data type will the predicted attribute be when it is calculated?

2) Why are the attribute ranges so important when doing linear regression data mining?

3) What are linear regression coefficients? What does 'weight' mean?

4) What is the linear regression mathematical formula, and how is it arranged?

5) How are linear regression results interpreted?

Extra thought question:

6) If you have an attribute that you want to use in a linear regression model, but it contains text data, such as the make or model of a car, what could you do in order to be able to use that attribute in your model?

EXERCISE

In the Chapter 4 exercise, you compiled your own data set about professional athletes. For this exercise, we will enhance this data set and then build a linear regression model on it. Complete the following steps:

1) Open the data set you compiled for the Chapter 4 exercise. If you did not do that exercise, please turn back to Chapter 4 and complete steps 1 – 4.

2) Split your data set's observations in two: a training portion and a scoring portion. Be sure that you have at least 20 observations in your training data set, and at least 10 in your scoring data set. More would be better, so if you only have 30 observations total, perhaps it would be good to take some time to look up ten or so more athletes to add to your scoring data set. Also, we are going to try to predict each athlete's salary, so if Salary is not one of your attributes, look it up for each athlete in your training data set (don't look it up for the scoring data set athletes, we're going to try to predict these). Also, if there are other attributes that you don't have, but that you think would be great predictors of salary, look these up, and add them to both your training and scoring data sets. These might be things like points per game, defensive statistics, etc. Be sure your attributes are numeric.

3) Import both of your data sets into your RapidMiner repository. Be sure to give them descriptive names. Drag and drop them into a new process, and rename them as Training and Scoring so that you can tell them apart.

4) Use a Set Role operator to designate the Salary attribute as the label for the training data.

5) Add a linear regression operator and apply your model to your scoring data set.

6) Run your model. In results perspective, examine your attribute coefficients and the predictions for the athletes' salaries in your scoring data set.

7) Report your results:
    a. Which attributes have the greatest weight?
    b. Were any attributes dropped from the data set as non-predictors? If so, which ones and why do you think they weren't effective predictors?
    c. Look up a few of the salaries for some of your scoring data athletes and compare their actual salary to the predicted salary. Is it very close? Why or why not, do you think?
    d. What other attributes do you think would help your model better predict professional athletes' salaries?