Thursday, November 24, 2016
12:20 PM

Chapter 13. Telecommunications Churn (Binomial Logistic Regression)

Logistic regression is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric one.

This example uses the stream named *telco_churn.str*, which references the data file named *telco.sav*. These files are available from the *Demos* directory of any IBM SPSS Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The *telco_churn.str* file is in the *streams* directory.

For example, suppose a telecommunications provider is concerned about the number of customers it is losing to competitors. If service usage data can be used to predict which customers are liable to transfer to another provider, offers can be customized to retain as many customers as possible.

This example focuses on using usage data to predict customer loss (churn). Because the target has two distinct categories, a binomial model is used. In the case of a target with multiple categories, a multinomial model could be created instead. See the topic Chapter 12, "Classifying Telecommunications Customers (Multinomial Logistic Regression)," on page 129 for more information.

Building the Stream

1. Add a Statistics File source node pointing to *telco.sav* in the *Demos* folder.

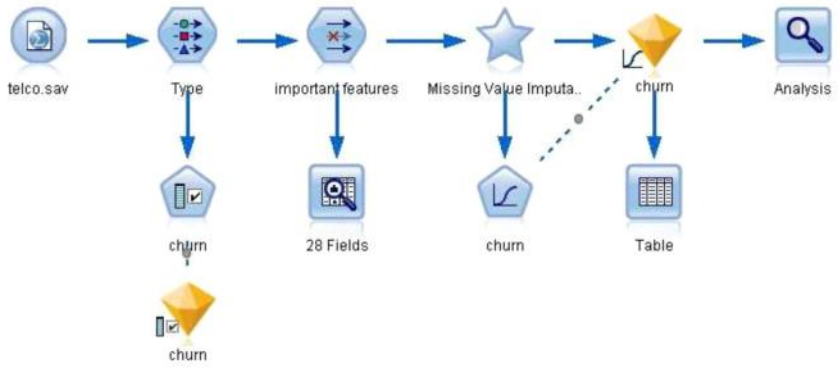

Figure 154. Sample stream to classify customers using binomial logistic regression

2. Add a Type node to define fields, making sure that all measurement levels are set correctly. For example, most fields with values 0 and 1 can be regarded as flags, but certain fields, such as gender, are more accurately viewed as a nominal field with two values.

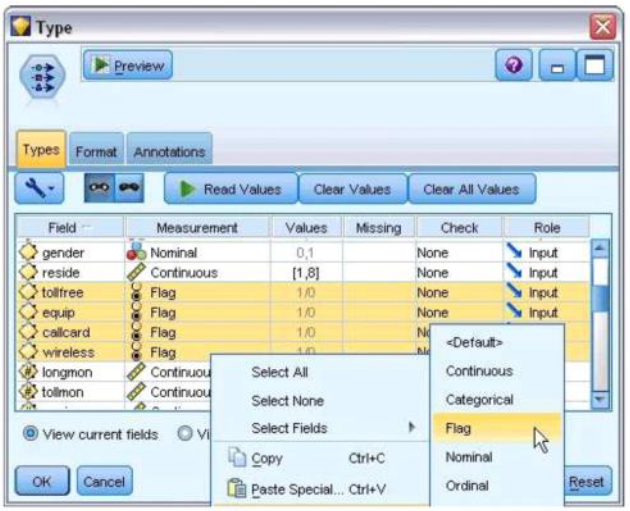

Figure 155. Setting the measurement level for multiple fields

Tip: To change properties for multiple fields with similar values (such as 0/1), click the *Values* column header to sort fields by value, and then hold down the Shift key while using the mouse or arrow keys to select all of the fields that you want to change. You can then right-click on the selection to change the measurement level or other attributes of the selected fields.

3. Set the measurement level for the *churn* field to **Flag**, and set the role to **Target**. All other fields should have their role set to **Input**.

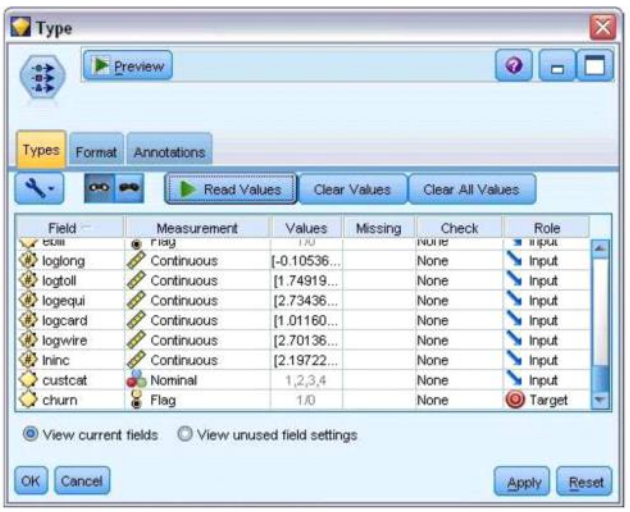

Figure 156. Setting the measurement level and role for the churn field

4. Add a Feature Selection modeling node to the Type node.

   Using a Feature Selection node enables you to remove predictors or data that do not add any useful information with respect to the predictor/target relationship.

5. Run the stream.

6. Open the resulting model nugget, and from the **Generate** menu, choose **Filter** to create a Filter node.

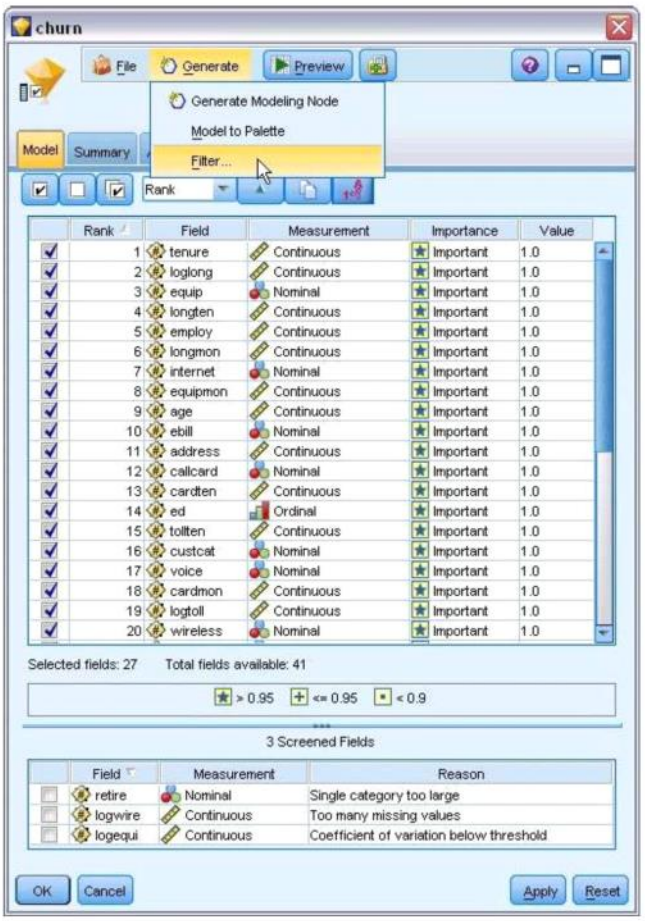

Figure 157. Generating a Filter node from a Feature Selection node

Not all of the data in the *telco.sav* file will be useful in predicting churn. You can use the filter to only select data considered to be important for use as a predictor.

7. In the Generate Filter dialog box, select **All fields marked: Important** and click **OK**.
8. Attach the generated Filter node to the Type node.

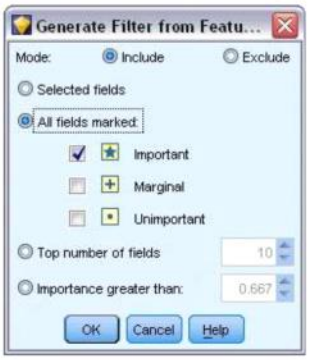

Figure 158. Selecting important fields

9. Attach a Data Audit node to the generated Filter node.
   Open the Data Audit node and click **Run**.
10. On the Quality tab of the Data Audit browser, click the *% Complete* column to sort the column by ascending numerical order. This lets you identify any fields with large amounts of missing data; in this case the only field you need to amend is *logtoll*, which is less than 50% complete.
11. In the *Impute Missing* column for *logtoll*, click **Specify**.

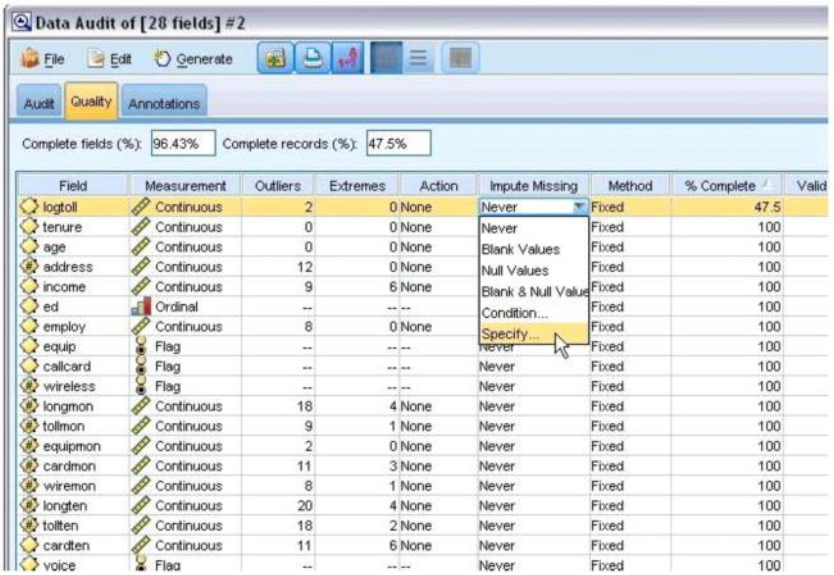

Figure 159. Imputing missing values for logtoll

12. For **Impute when**, select **Blank and Null values**. For **Fixed As**, select **Mean** and click **OK**.
    Selecting **Mean** ensures that the imputed values do not adversely affect the mean of all values in the overall data.

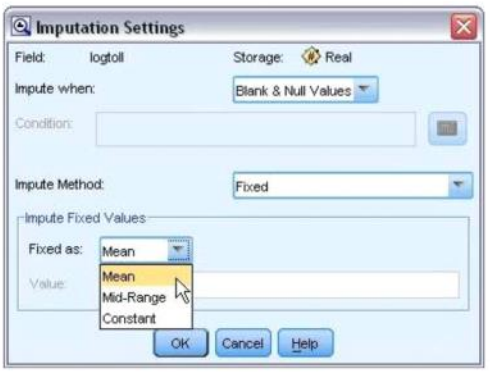

Figure 160. Selecting imputation settings

13. On the Data Audit browser Quality tab, generate the Missing Values SuperNode. To do this, from the menus choose:

    Generate > Missing Values SuperNode

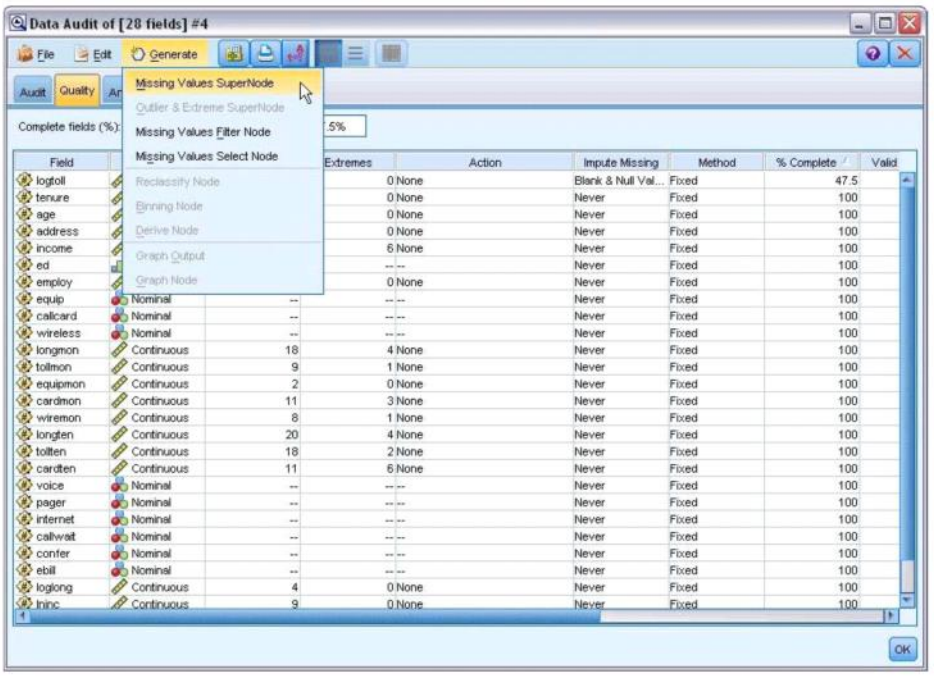

Figure 161. Generating a missing values SuperNode

In the Missing Values SuperNode dialog box, increase the **Sample Size** to 50% and click **OK**. The SuperNode is displayed on the stream canvas, with the title: *Missing Value Imputation*.

14. Attach the SuperNode to the Filter node.

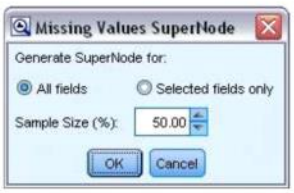

Figure 162. Specifying sample size

15. Add a Logistic node to the SuperNode.
16. In the Logistic node, click the Model tab and select the **Binomial** procedure. In the *Binomial Procedure* area, select the **Forwards** method.

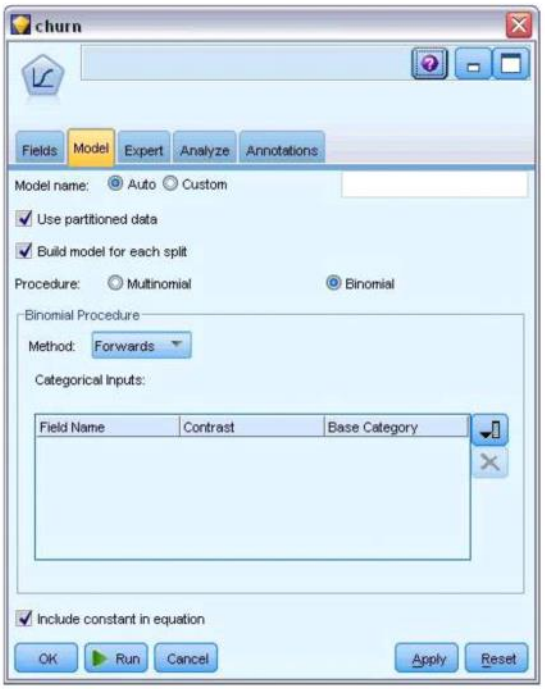

Figure 163. Choosing model options

17. On the Expert tab, select the **Expert** mode and then click **Output**. The Advanced Output dialog box is displayed.
18. In the Advanced Output dialog, select **At each step** as the *Display* type. Select **Iteration history** and **Parameter estimates** and click **OK**.

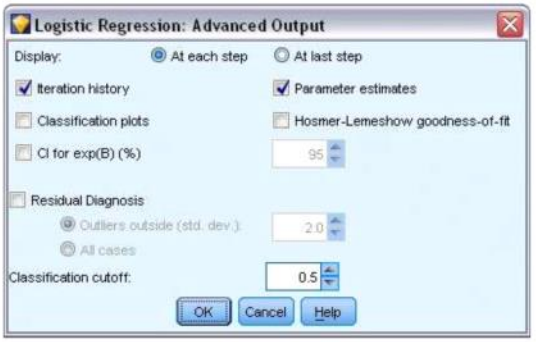

Figure 164. Choosing output options

Browsing the Model

1. On the Logistic node, click **Run** to create the model.

   The model nugget is added to the stream canvas, and also to the Models palette in the upper-right corner. To view its details, right-click on the model nugget and select **Edit** or **Browse**.

   The Summary tab shows (among other things) the target and inputs (predictor fields) used by the model. Note that these are the fields that were actually chosen based on the Forwards method, not the complete list submitted for consideration.

Figure 165. Model summary showing target and input fields

The items shown on the Advanced tab depend on the options selected on the Advanced Output dialog box in the Logistic node. One item that is always shown is the Case Processing Summary, which shows the number and percentage of records included in the analysis. In addition, it lists the number of missing cases (if any) where one or more of the input fields are unavailable and any cases that were not selected.

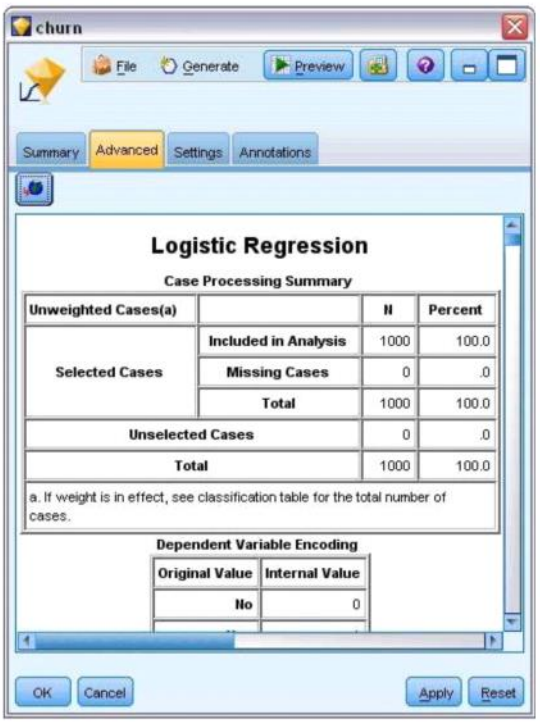

Figure 166. Case processing summary

2. Scroll down from the Case Processing Summary to display the Classification Table under Block 0: Beginning Block.

   The Forward Stepwise method starts with a null model - that is, a model with no predictors - that can be used as a basis for comparison with the final built model. The null model, by convention, predicts everything as a 0, so the null model is 72.6% accurate simply because the 726 customers who didn't churn are predicted correctly. However, the customers who did churn aren't predicted correctly at all.

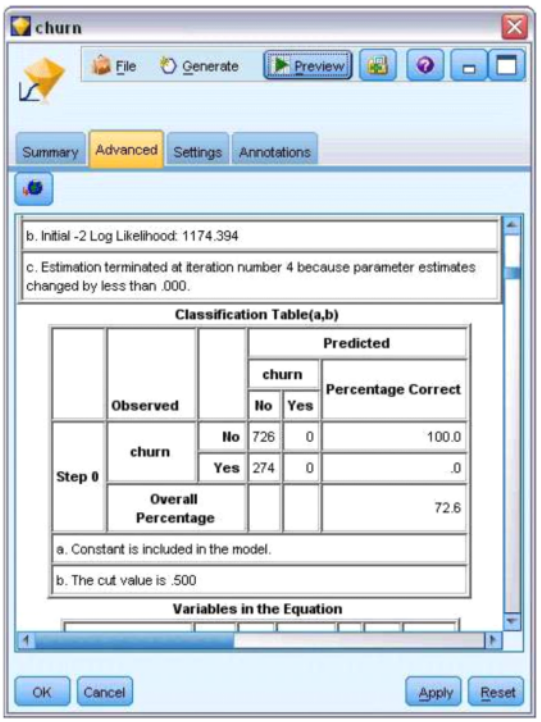

Figure 167. Starting classification table- Block 0

3. Now scroll down to display the Classification Table under Block 1: Method = Forward Stepwise.
   This Classification Table shows the results for your model as a predictor is added in at each of the steps. Already, in the first step - after just one predictor has been used - the model has increased the accuracy of the churn prediction from 0.0% to 29.9%

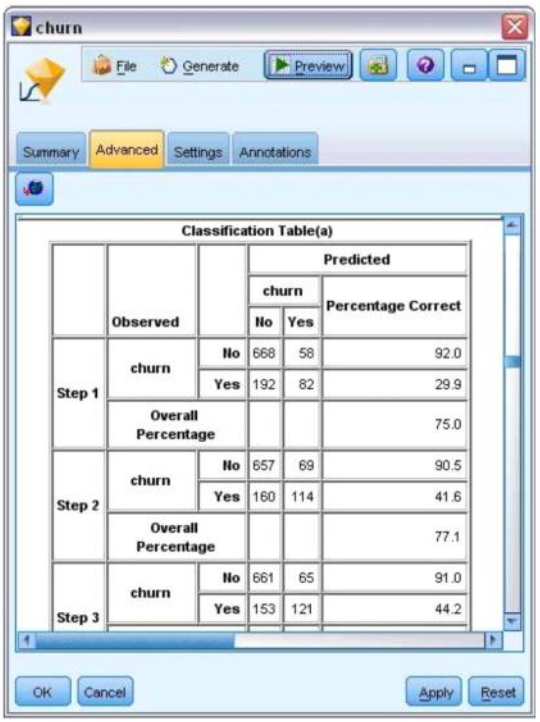

Figure 168. Classification table - Block 1

4. Scroll down to the bottom of this Classification Table.

The Classification Table shows that the last step is step 8. At this stage the algorithm has decided that it no longer needs to add any further predictors into the model. Although the accuracy of the non-churning customers has decreased a little to 91.2%, the accuracy of the prediction for those who did churn has risen from the original 0% to 47.1%. This is a significant improvement over the original null model that used no predictors.

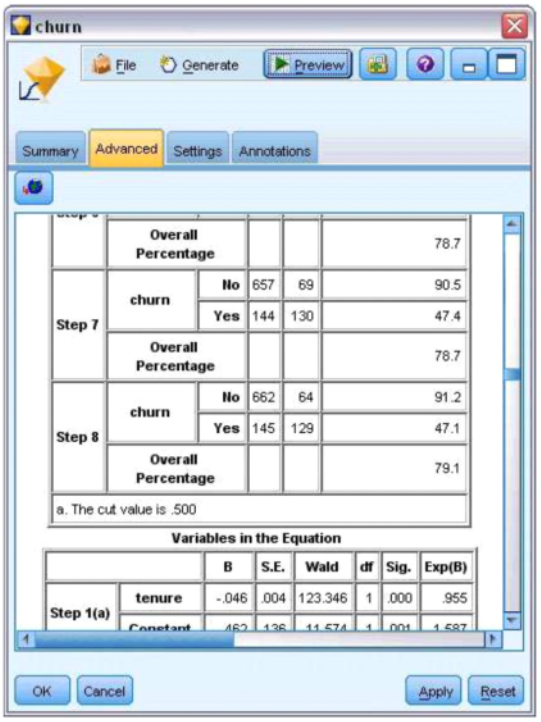

Figure 169. Classification table - Block 1

For a customer who wants to reduce churn, being able to reduce it by nearly half would be a major step in protecting their income streams.

Note: This example also shows how taking the Overall Percentage as a guide to a model's accuracy may, in some cases, be misleading. The original null model was 72.6% accurate overall, whereas the final predicted model has an overall accuracy of 79.1%; however, as we have seen, the accuracy of the actual individual category predictions were vastly different.

To assess how well the model actually fits the data, a number of diagnostics are available in the Advanced Output dialog box when you are building the model. Explanations of the mathematical foundations of the modeling methods used in IBM SPSS Modeler are listed in the *IBM SPSS Modeler Algorithms Guide*, available from the *\Documentation* directory of the installation disk.

Note also that these results are based on the training data only. To assess how well the model generalizes to other data in the real world, you would use a Partition node to hold out a subset of records for purposes of testing and validation.