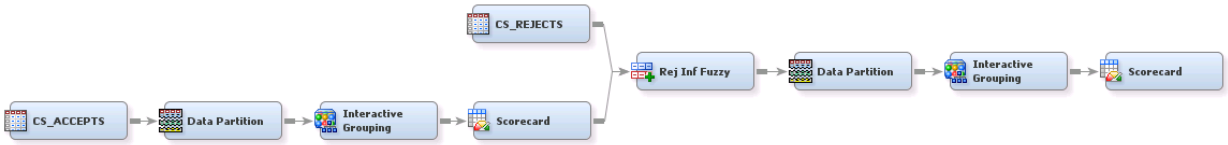
Reject Inference Using Credit Scoring for SAS® Enterprise Miner™



Data:
This flow requires two data sets. The first one is SAMPSIO.CS_ACCEPTS, the same data set that is used on the Credit Scoring Scorecard and Reverse Scorecard example; it contains 3,000 observations with a binary target variable, a frequency variable, and demographic and institution-specific input variables. The second data set is SAMPSIO.CS_REJECTS, which is called the Rejects data set in this example. The rejects data set is a sample of 1,500 customers whose credit application was rejected; it contains the same input variables but has no defined target variable.

Goal:
A scorecard that is developed using only the accepted applicants may incur sample bias. The goal of a reject inference diagram flow is to solve the bias by calibrating the scorecard in context with a population that includes both accepted and rejected observations. This population is usually known as the through-the-door population.

Flow:
After you develop a scorecard following the steps outlined in the Credit Scoring Scorecard and Reverse Scorecard example, add the Rejects Data node (titled CS_REJECTS in this example) to the diagram with a role of score. To do that, select the **score** role at the last step of the data source creation wizard.

Reject Inference Node
Connect both the Scorecard node and the CS_REJECTS node to the Reject Inference node. The Reject Inference node obtains an augmented training data set that is more similar to the through-the-door population by scoring the rejected data based on the training model, inferring the target for the Rejects dataset through any of the three available methods and then appending it to the Accepts data set. By default, the Fuzzy Criterion is used for reject inference; other available methods are described in the section “Reject Inference” in the Reference Help.

After the Reject Inference node has created the augmented data set, the basic Scorecard flow is repeated—the three best practice nodes for credit scoring are connected: a Data Partition node to avoid overtraining, an Interactive Grouping node to visually inspect all trends of weight of evidence across groupings (override with a manual WOE if necessary), and a Scorecard node to train a logistic regression and calculate score points for each level. The final scorecard of this flow now accounts for any bias, because it is trained with a population that is more similar to the through-the-door population.