Paper 123-2010

The Power of the Group Processing Facility in SAS® Enterprise Miner™

Sascha Schubert, SAS Institute Inc., Cary, NC

ABSTRACT

The group processing facility in SAS® Enterprise Miner™ is useful when your data can be segmented or grouped, and you want to process the grouped data in different ways. It uses BY-group processing to process observations from one or more data sources that are grouped or ordered by values of one or more common variables. For example, model projects that leverage the same input data space and multiple target variables can be easily and efficiently managed and processed by using the group processing facility.

INTRODUCTION

Figure 1 illustrates the general process flow of group processing in SAS Enterprise Miner. BY-group processing takes place between the Start Groups node and the End Groups node. The nature of the BY-group processing is defined by setting parameters in the dialog box of the Start Groups node.

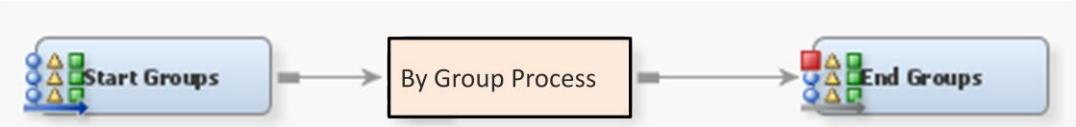


Figure 1: General Flow of Group Processing

Based on these settings, the group processing facility can be used to:

- analyze more than one target variable in the same process flow
- define group variables such as GENDER or JOB, in order to obtain separate analyses for each level of the group variable or variables.
- use cross validation techniques to test the stability of predictive models
- specify index looping, or how many times the flow following the node should loop
- resample the data set to create bagging and boosting models

This paper introduces examples for these different use cases of the group processing facility.

MULTIPLE TARGETS – COMBINE MODELS

Many business problems in different industries can gain from a unified data preparation process and group processing by using the multiple target approach. For example, in a customer intelligence context, often organizations are interested in modeling several dimensions of customer behavior based on the same or a very similar input data space. The input data for different models can then be collected, transformed, and prepared in an integrated process that creates a unified analytical base table (ABT), which contains all potential predictors for different customer behavior models, such as customer retention, next best offer (for example, best selection from portfolio of offerings), the response to different treatments in a campaign (for example, basic, premium, and platinum), up-selling, channel optimization in multi-channel campaigns, and so on.

The format of the ABT is illustrated in Figure 2. The input space consists of attributes that provide potential information for the prediction and/or classification of several targets. For each target variable that is identified in the input data based on the metadata of the Input Data Source node, the group processing facility creates a modeling loop. SAS Enterprise Miner allows the combination of

mixed or binary categorical and interval targets, and it chooses the appropriate modeling algorithms automatically as part of the group processing step.

The group processing facility can handle several nodes as part of the group processing step, which enables it to leverage stratum-dependent data partitioning and data transformations. Furthermore, parallel branches inside the group processing step are supported. Thus, in each loop, multiple modeling algorithms can run in parallel to identify the champion model.

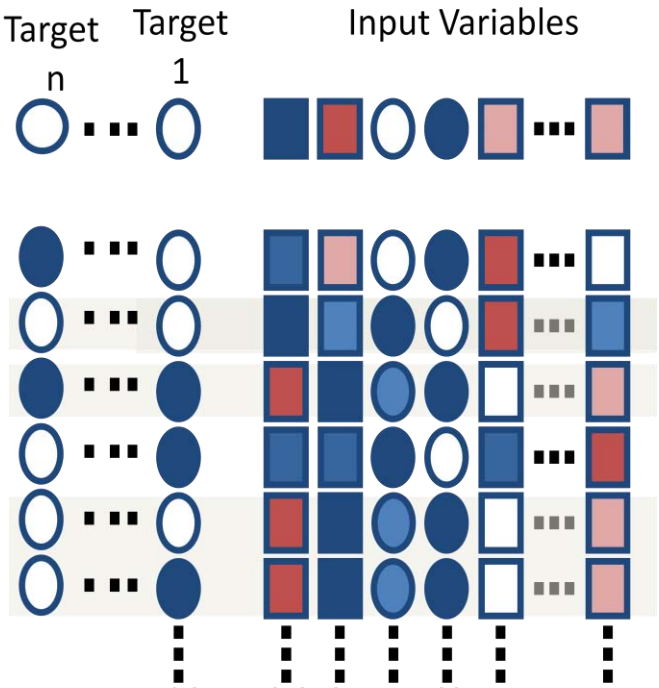
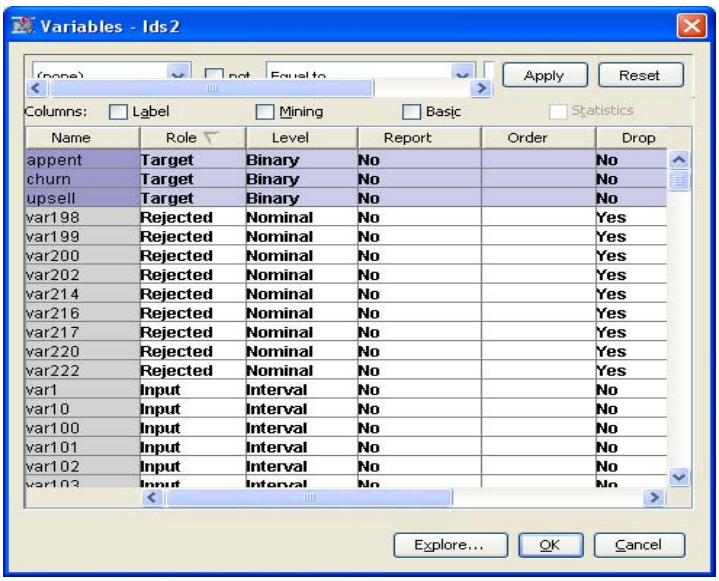


Figure 2: Format of the Analytical Base Table

Figure 3 lists the metadata for the variables of the ABT in a customer behavior modeling project. Based on approximately 200 input variables that describe numerous customer behavior dimensions, we want to create a classification model for three target variables: “appent,” “churn,” and “upsell.”



Name	Role	Level	Report	Order	Drop
appent	Target	Binary	No		No
churn	Target	Binary	No		No
upsell	Target	Binary	No		No
var198	Rejected	Nominal	No		Yes
var199	Rejected	Nominal	No		Yes
var200	Rejected	Nominal	No		Yes
var202	Rejected	Nominal	No		Yes
var214	Rejected	Nominal	No		Yes
var216	Rejected	Nominal	No		Yes
var217	Rejected	Nominal	No		Yes
var220	Rejected	Nominal	No		Yes
var222	Rejected	Nominal	No		Yes
var1	Input	Interval	No		No
var10	Input	Interval	No		No
var100	Input	Interval	No		No
var101	Input	Interval	No		No
var102	Input	Interval	No		No

Figure 3: Data Mining Metadata

Figure 4 shows the Group Processing Node properties for the mode “Target”. Except for this property adjustment, we use the default settings for this node, which means that all the other settings are set based on the characteristics of the data.

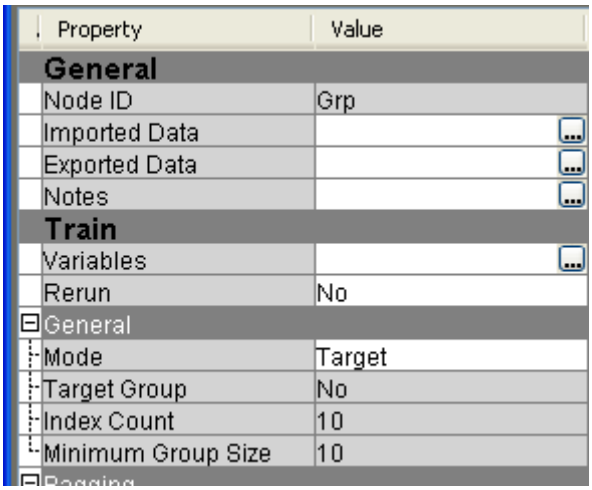


Figure 4: Property Settings for Target Group Processing

As outlined in Figure 5, we use group-based sampling, partitioning, and decision tree modeling inside the group processing facility. With this process design, the sample will be drawn separately within each BY-group segment. The partition data will contain data only from the respective group. Users are very flexible with their process design and can easily adjust to the requirements of their project.

Figure 5: Process Flow Example for Target Group Processing

The results of this process flow can be assessed by opening the result browser of the End Groups node as summarized in Figure 6. Three binary classification tree models have been created, and for each model several model performance statistics and graphics are displayed.

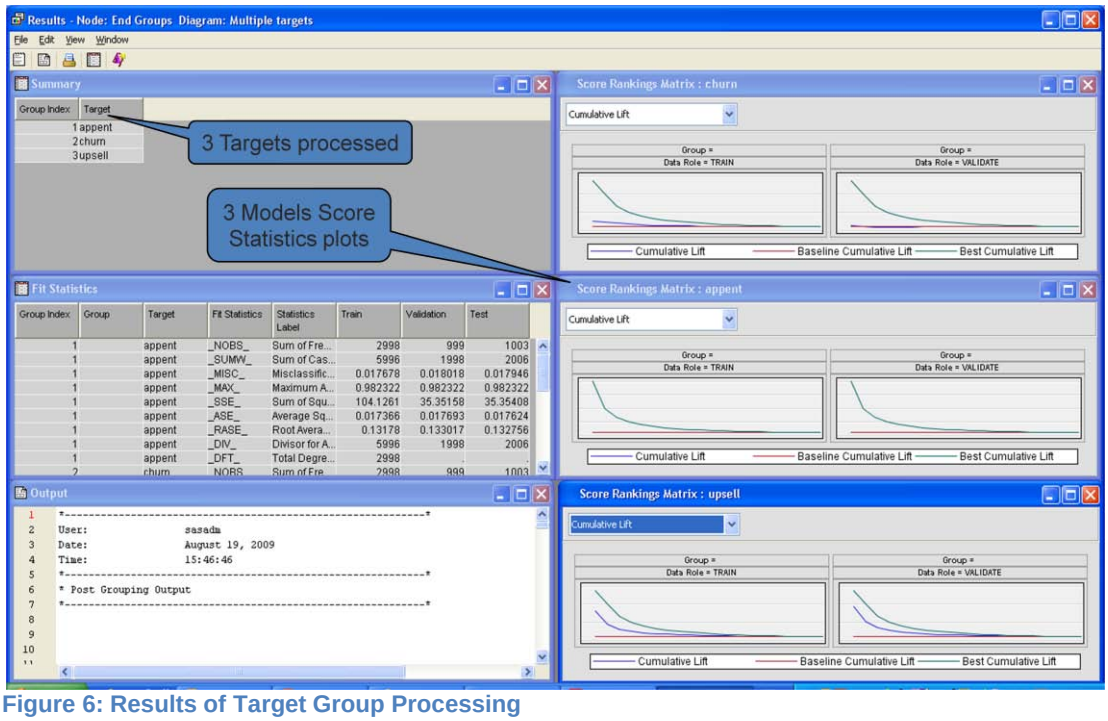


Figure 6: Results of Target Group Processing

This simple example demonstrates nicely how efficiently the processing facility can be used for modeling multiple targets based on a unified analytical base table, instead of creating different flows for the different models. This very compact flow can be exported as a single SAS batch program. The model training for these different targets can also be processed in a scheduled or triggered way without user intervention.

At the end of the process flow in Figure 5, we use a Score Code node to accumulate the score code for all models created in the group processing loop in one single program. For scoring, we need to provide only one input data table and can process all three models in a single run.

With SAS Enterprise Miner 6.1, the score code has been optimized to contain only variables required for the final model scoring. This optimization minimizes the data required for scoring to just the ones that are used in the scoring run, including variables required for the calculation of derived and transformed variables. Besides a slimmer scoring input table, this optimization also minimizes the scoring code by removing all calculations that are not mandatory for the scoring. Scoring performance, data preparation, and code maintenance will significantly benefit from this enhancement.

STRATIFIED, BY-GROUPS (SEGMENTS) – SEPARATE MODELS

While the target group processing helps to combine model training processes, the BY-group processing for the input data helps create separate models for separate segments in a single analytical base table. This can be required in different situations. For example, customer behavior model performance can benefit from segmenting the customer base into more homogenous groups before training predictive models. Marketing organizations segment their customers by using demographic characteristics, such as gender, regions, or family status. Another approach is to dynamically segment the customer base by using analytical segmentation techniques and training separate predictive models for each cluster.

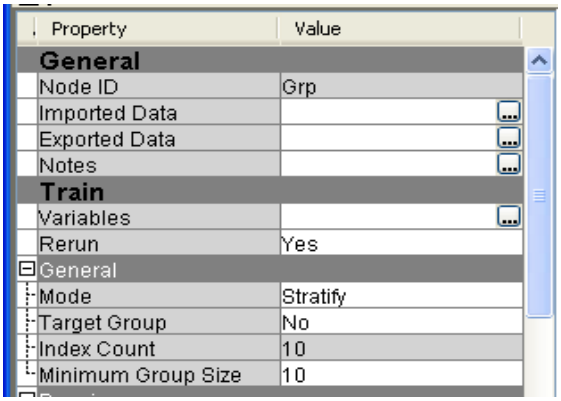
The group processing subsets the input data for each segment definition and applies the data mining process flow steps inside the grouping facility to the selected data segment. Segmentation on multiple attributes is supported. For example, an ABT can be segmented both on the attribute “Region” and the attribute “Store.” For each combination of the unique levels of the stratification variables “Region” and “Store,” a separate model is created.

It is also possible to combine target and segment grouping, which allows for the design of compact process flows for many segments and targets. For example, a marketing organization might want to create a next best offer model framework for different customer segments of their customer base. With a portfolio of five different products to offer and three customer segments, they would be able to build 15 different models, one for each product and for each segment, in a very simple process flow.

The model score is accumulated, as it is for the target group processing. That is, with one single score code, we are able to create a single score program that processes all models in one run.

An example process flow is shown in Figure 7. A decision tree model is trained for different segments in ABT. The segment variable is defined in the metadata variable window of the Start Groups node, as well as the process mode “Stratify.” (See Figure 8.) In our example, as shown in Figure 9, we use the variable “Gender” for segmenting the input data space.

Figure 7: Process Flow Example for Segment Group Processing



Property	Value
General	
Node ID	Grp
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Rerun	Yes
General	
Mode	Stratify
Target Group	No
Index Count	10
Minimum Group Size	10

Figure 8: Property Setting for Segment Grouping

Name	Use	Report	Grouping Role	Role	Level
ACCTNUM	**Default**	**No**	**Default**	ID	Nominal
AMOUNT	**Default**	**No**	**Default**	Target	Interval
APRTMNT	**Default**	**No**	**Default**	Input	Nominal
GENDER	**Default**	**No**	**Stratification**	Input	Nominal
NTITLE	**Default**	**No**	**Default**	Input	Nominal
PURCHASE	**Default**	**No**	**Default**	Target	Binary
STATECOD	**Default**	**No**	**Default**	ID	Nominal
TELIND	**Default**	**No**	**Default**	Input	Nominal

Figure 9: Definition of the Segmentation Variable Role

The result browser of the End Groups processing node summarizes the results of the stratified model training. (See Figure 10). In the case of two distinct values for the attribute “Gender”, two stratified models are trained, one for Gender=”Female” and one for Gender=”Male”. All the usual model assessment statistics are calculated and graphics are created for the stratified models as well as for the overall population, which makes it easy to spot differences and decide whether the model stratification leads to better model results.

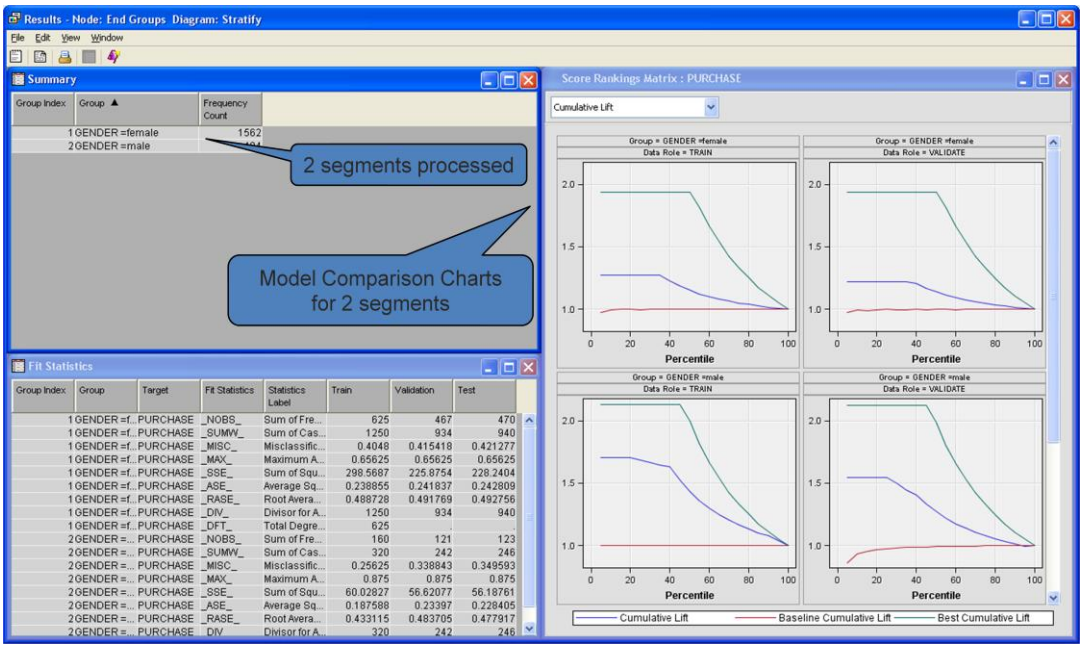


Figure 10: Model Assessment for Stratified Models

The design of the process steps within the group processing facility can be very flexible when using the stratification mode. Using additional model algorithms, we can test champion models against challenger algorithms running in a parallel processing environment. This provides the data miner with a high-performance model training facility.

The process flow in Figure 11 compares three modeling algorithms, Decision Tree, Regression, and Neural Network, for different segments of the customer base.

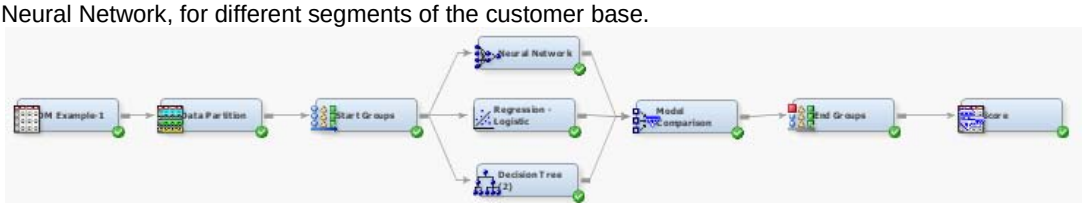


Figure 11: Flow for Champion-Challenger Comparison on Data Segments

Figure 12 provides an overview of all model results in a compact and easy-to-understand display. For each segment (Male and Female), each data partition role (Training, Validation, and Test) and each modeling algorithm (Tree, Regression, and Neural Network), the misclassification rate is compared. This chart shows remarkable differences in the model performance for the different strata. In general, it seems that males are more predictable than females, as the misclassification rate for the male segment is consistently lower. Looking at the model performance using the validation data, we see the regression model performs best for females, and the decision tree performs best for males. Armed with this information, the data miner can easily select the best model for each segment. However, SAS Enterprise Miner also selects the best model per strata automatically based on a pre-selected assessment statistic, which by default is the misclassification rate on the validation data.

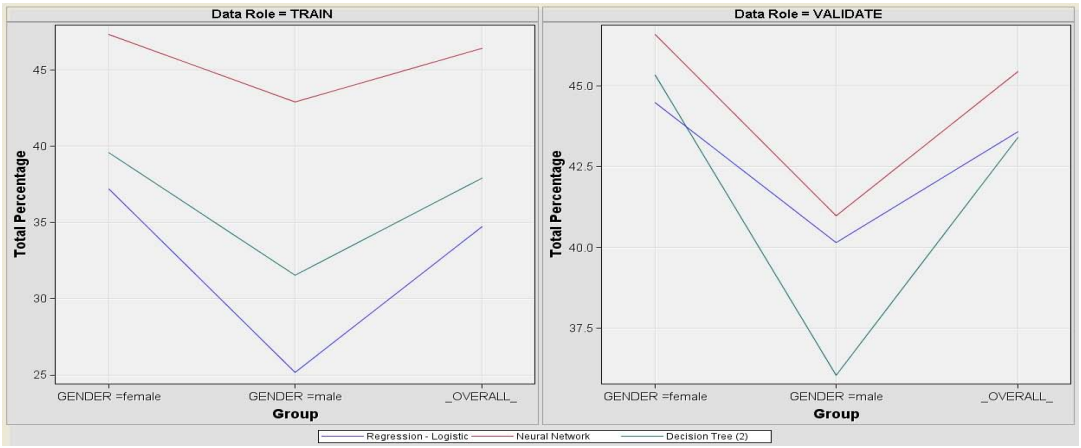


Figure 12: Stratified Model Assessment Chart

The optimized scoring code is compiled automatically for the best model selected for each segment including all necessary data transformations. This is a tremendous time-saver for the analyst, who would otherwise need to manually paste the code together from different sources.

Besides the business-driven segmentation of the input data space, the group processing facility can also handle dynamic data-driven segmentation. Using one of the segmentation algorithms provided, such as clustering, SOMs, or decision tree segmentation, the input data space can be segmented into homogenous groups by using statistical segmentation. Each of these segmentation techniques provides an integrated scoring algorithm, which assigns a segment identifier to each record. This identifier can then be used to define the loops in the group processing. Figure 13 shows an example of dynamic segmentation combined with group processing in SAS Enterprise Miner. The Cluster node has identified three groups in the input data, and for each of these segments a decision tree is created by using the group processing facility.

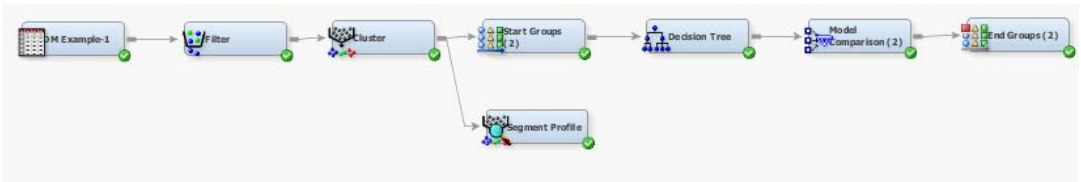


Figure 13: Example Process Flow for Dynamic Segmentation

As shown in Figure 14, the resulting performance of stratified models differs, suggesting that there are real differences between the segments. Dynamic segmentation provides the advantage of stratified modeling being applied to as many groups as are supported by the structure of the input data space.

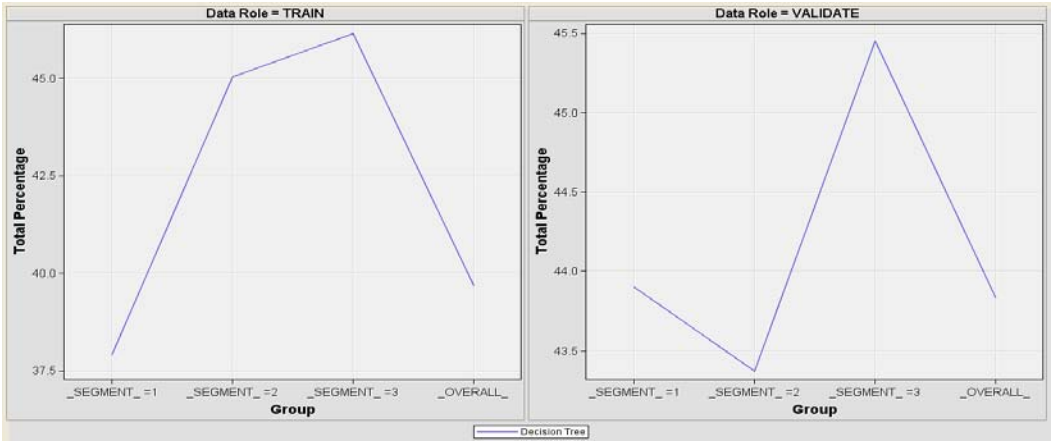


Figure 14: Model Assessment for Dynamic Segmentation

CROSS VALIDATION – TEST MODELS

The group processing facility also allows data miners to apply cross validation to test the model stability. This is especially useful when the input data is scarce and the usual data partitioning approach (training, validation, and test data) is not possible. The group processing facility in SAS Enterprise Miner supports two methodologies for cross validation, one direct and one indirect. The direct method is to select the model "Cross-Validation" in the dialog box of the Start Groups node. (See Figure 15.)

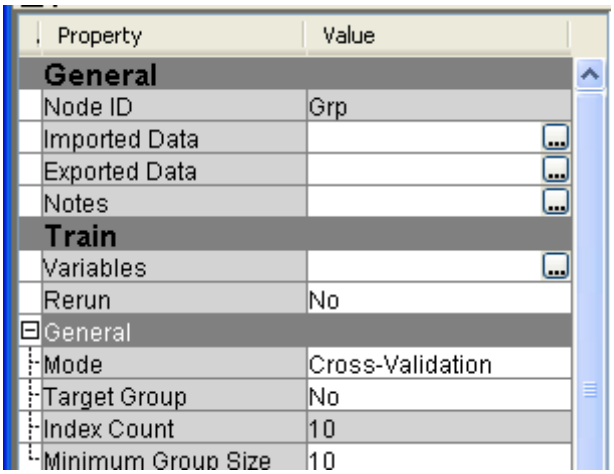


Figure 15: Settings for Cross Validation

This selection causes the subset of the input data to export the complement of groups specified as opposed to the groups themselves. For example, assume you have two group variables, GENDER [M, F] and REGION [N, S, E, W]. The following groups (Table 1) are passed when you use stratify-mode looping to perform standard group processing.

Table 1: Input Data Segmentation for Standard Group Processing

Loop Number	Segment Values Selected for Model Training
1	GENDER M AND REGION N
2	GENDER F AND REGION N
3	GENDER M AND REGION S
4	GENDER F AND REGION S
5	GENDER M AND REGION E
6	GENDER F AND REGION E
7	GENDER M AND REGION W
8	GENDER F AND REGION W

When we use cross-validation mode looping, all the data except the group is passed. Using cross-validation mode looping with our example group variables GENDER [M, F] and REGION [N, S, E, W], we get the following groups (Table 2).

Table 2: Input Data Segmentation for f-fold Cross-Validation

Loop Number	Segment Values Selected for Model Training
1	GENDER F AND REGIONS S,W,E
2	GENDER M AND REGIONS S,W,E
3	GENDER F AND REGIONS N,W,E
4	GENDER M AND REGIONS N,W,E
5	GENDER F AND REGIONS S,N,W
6	GENDER M AND REGIONS S,N,W
7	GENDER F AND REGIONS N,E,S
8	GENDER M AND REGIONS N,E,S

With this setting we build the model on the complement data of a group segment and provide N loops over different data segments that can be used to calculate model stability statistics.

One disadvantage with this approach might be that the data segments are not selected randomly and model instability effects could be systematic effects caused by differences in the groups.

The use of the group processing facility also allows a true f-fold cross validation by using the data transformation node. As shown in Figure 16, we use a transformation before entering the looping process.

Figure 16: Use of a Transformation Node for f-fold Cross Validation

In this node we create a random segmentation ID for the data for 10 groups to be used as cross validation indicators in the group processing loop. (See Figure 17.) The general formula for this segment ID is:

int((f*(ranuni(0)))+1),

where f is the number of segments (Folds) that should be used in the f-fold cross validation. The role of the new variable can be set to "Segment" in the transformation node, which allows for its use for cross validation segmentation immediately.

Again, we set the mode in the Start Groups node dialog box to cross validation, causing the selection of the complement of each segment during the loop. For example, with the setting Segment ID =1, all segments excluding ID =1 are used for the model training, providing a sufficiently large sample for stable model training.

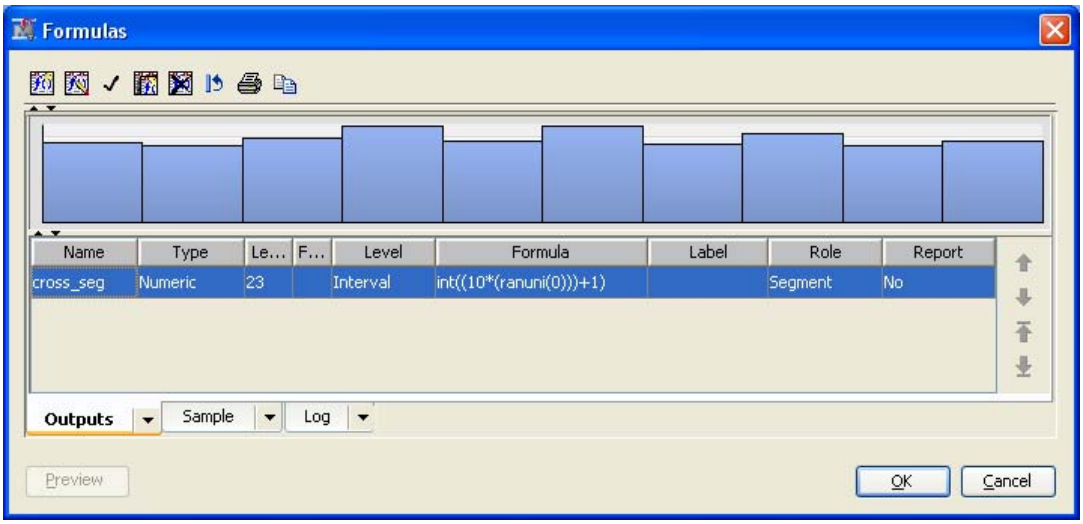


Figure 17: Cross Validation Segment ID Creation

Figure 18 depicts the results of the 10 runs for the cross validation runs. We see quite a bit of variance between the different loops, which points to instability in the models.

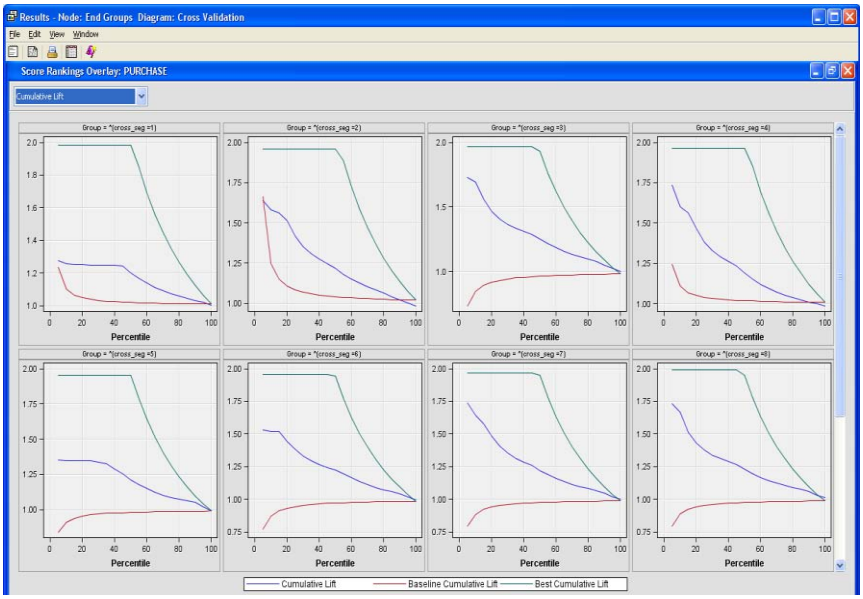


Figure 18: Results of 10-fold Cross Validation

The SAS Enterprise Miner interactive graph facility provides a large variety of graphs that can be created based on the rich set of performance statistics that are calculated by modeling nodes to visualization the variance in model performance. (see Figure 19.)

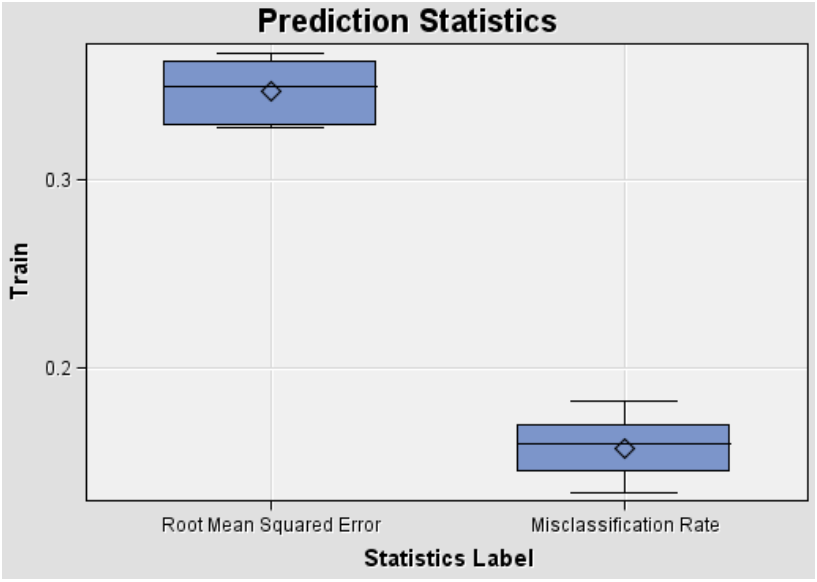


Figure 19: Box Plot of Variances in RMSE and Misclassification Rate

RESAMPLING MODEL ENSEMBLES – STABILIZE MODELS

Finally, the group processing facility in SAS Enterprise Miner provides two automated model ensemble algorithms, bagging and boosting. These are machine learning ensemble algorithms to improve accuracy and stability of classification and prediction models. See the references at the end of this paper for more information about the specific approaches to sampling for bagging and boosting,.

The main difference between bagging and boosting is the sampling approach for the subset that is selected for each model loop.

Bagging applies an unweighted resampling that uses random sampling with replacement to create the n samples. Each observation has the same chance to be drawn into the model training subset in each loop. Thus, each loop is independent of all the previous loops and can be run easily in parallel. The final model output is created by averaging the probabilities generated by each model iteration.

Boosting performs weighted resampling to boost the accuracy of the model by focusing on observations that are more difficult to classify or to predict. At the end of each iteration, the sampling weight is adjusted for each observation in relation to the accuracy of the model result. Correctly classified observations receive a lower sampling weight, and incorrectly classified observations receive a higher weight. Thus, the next iteration will draw a sample with more observations that have been misclassified before and will focus in on these cases. This leads to a dependency of the iterations and a sequential processing of the algorithm. Also, the scoring of the final model needs to be processed in the same sequence as the training algorithm. The final model outcome is the weighted majority vote of the sequence of classifiers.

SAS Enterprise Miner creates the final scoring code of the model ensembles automatically by accumulating the score code from each model iteration. Therefore, the entire score code is produced as the result of the group processing run, and ready to be deployed in production. One disadvantage of model ensemble algorithms is the loss of interpretability and transparency of the model results.

Figure 20 summarizes the differences between bagging and boosting.

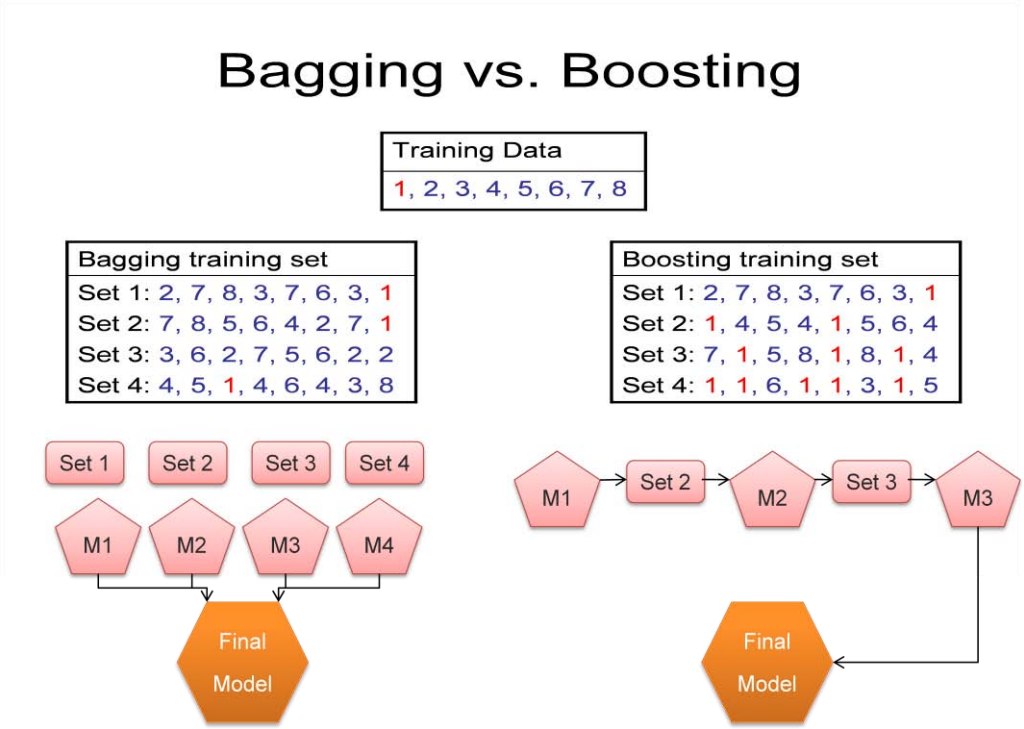


Figure 20: Main Characteristics of Bagging and Boosting

Bagging and boosting can be easily implemented in SAS Enterprise Miner with the group processing facility. The user chooses the approach by selecting the respective mode in the dialog box of the Start Groups node. (See Figure 21.) Once the sampling properties are defined for bagging, the algorithms run automatically and produce results that can be assessed in the result browser of the End Groups node. For boosting, the user just needs to define the number of iterations.

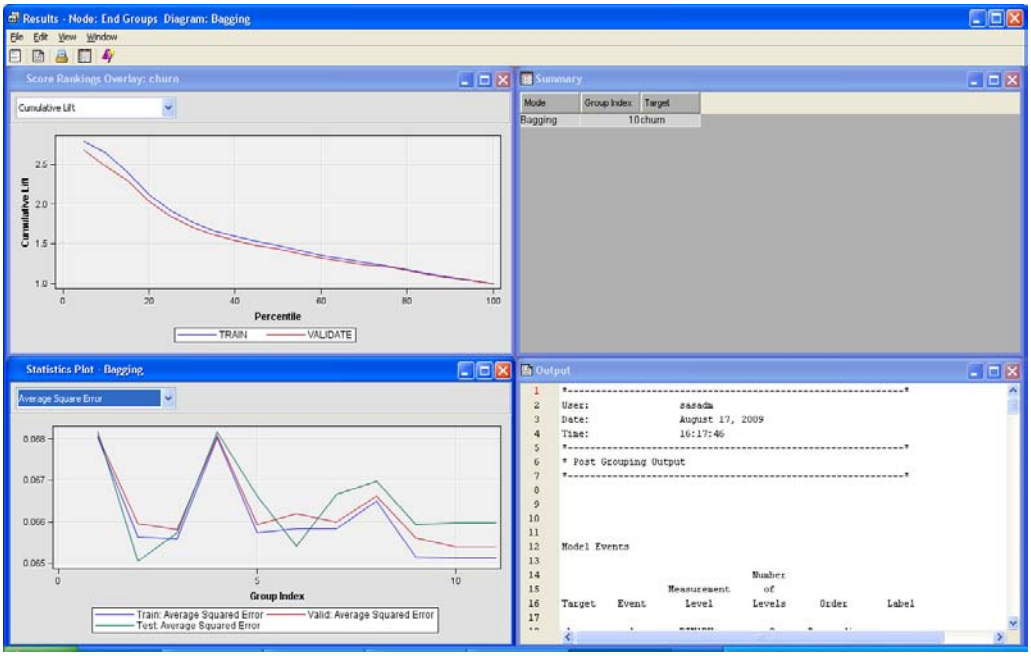


Figure 21: Bagging Results Graphics

The lift chart shows the model results for the final model classifications for training and validation data. The statistics plot displays different error statistics for each model iteration and provides insight into the model stability. We can see in our example that the models are stable over all iterations and all data partitions.

CONCLUSION

The group processing facility in SAS Enterprise Miner provides different approaches to segment or combine model training to make it easy for users to select the most efficient process.
Models can be trained in combination for several targets based on the same input data set. Models can be built easily on different segments of the input data if this data suggests segments either from a business or a dynamic data segmentation perspective.

Model stability can be easily tested with different cross validation algorithms, and model stability can be improved by applying resampling model ensemble techniques.

As with any other predictive model built in SAS Enterprise Miner, at the end of the process the scoring code required to deploy the optimal model in production is readily available. This can help to speed up productivity to build models on the right level quickly. Of course, the process flow can be easily exported to run in batch mode to leverage off-office hours and computing power.

REFERENCES

Breiman, L. 1996. “Bagging Predictors.” *Mathematical Reviews* 26: 123-140.

Freund, Y., and R. Schapire. 1996. “Game Theory, On-line Prediction and Boosting.” *Proceedings of the Ninth Annual Conference on Computational Learning Theory*. 325-332.

“SAS Enterprise Miner 6.1 Fact Sheet.” SAS Institute Inc., Cary, NC. Available at http://www.sas.com/resources/factsheet/sas-enterprise-miner-factsheet.pdf

SAS Enterprise Miner 6.1 Help. SAS Institute Inc., Cary, NC.

ACKNOWLEDGMENTS

The author would like to thank the following SAS employees for their valuable contributions to this paper: Wayne Thompson, David Duling, and Dominique Latour.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Sascha Schubert
SAS Institute Inc.
Domaine de Grégy
Grégy-sur-Yerres
77257 Brie Comte Robert Cedex
Sascha.Schubert@sas.com