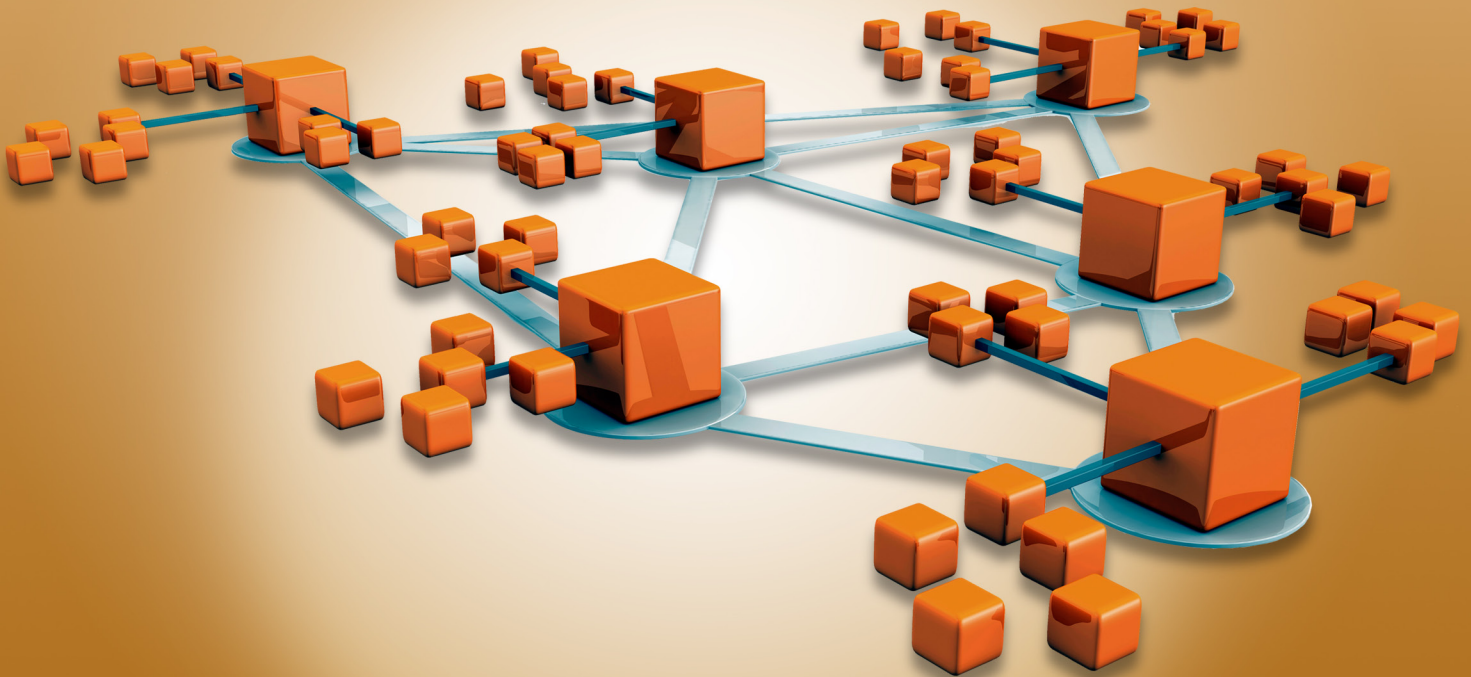
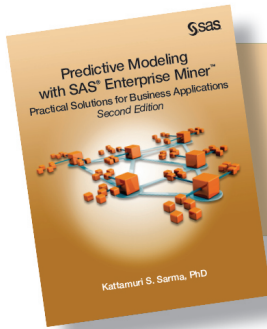


Predictive Modeling with SAS[®] Enterprise Miner[™]

Practical Solutions for Business Applications
Second Edition



Kattamuri S. Sarma, PhD



From *Predictive Modeling with SAS[®] Enterprise Miner[™]*,
Second Edition. Full book available for purchase [here](#).

Contents

Preface	xi
About This Book	xv
About The Author	xix
Acknowledgments	xxi
Chapter 1: Research Strategy	1
1.1 Introduction	1
1.2 Measurement Scales for Variables	1
1.3 Defining the Target	2
1.3.1 Predicting Response to Direct Mail	2
1.3.2 Predicting Risk in the Auto Insurance Industry	3
1.3.3 Predicting Rate Sensitivity of Bank Deposit Products	4
1.3.4 Predicting Customer Attrition	6
1.3.5 Predicting a Nominal Categorical (Unordered Polychotomous) Target	7
1.4 Sources of Modeling Data	8
1.4.1 Comparability between the Sample and Target Universe	8
1.4.2 Observation Weights	8
1.5 Pre-Processing the Data	8
1.5.1 Data Cleaning Before Launching SAS Enterprise Miner	9
1.5.2 Data Cleaning After Launching SAS Enterprise Miner	9
1.6 Alternative Modeling Strategies	10
1.6.1 Regression with a Moderate Number of Input Variables	10
1.6.2 Regression with a Large Number of Input Variables	11
1.7 Notes	11
Chapter 2: Getting Started with Predictive Modeling	13
2.1 Introduction	14
2.2 Opening SAS Enterprise Miner 12.1	14
2.3 Creating a New Project in SAS Enterprise Miner 12.1	14
2.4 The SAS Enterprise Miner Window	15
2.5 Creating a SAS Data Source	16
2.6 Creating a Process Flow Diagram	25
2.7 Sample Nodes	26
2.7.1 Input Data Node	26
2.7.2 Data Partition Node	27
2.7.3 Filter Node	28

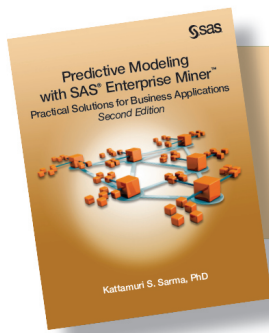
2.7.4 File Import Node	32
2.7.5 Time Series Node	35
2.7.6 Merge Node.....	45
2.7.7 Append Node	48
2.8 Tools for Initial Data Exploration.....	50
2.8.1 Stat Explore Node.....	51
2.8.2 MultiPlot Node	56
2.8.3 Graph Explore Node.....	58
2.8.4 Variable Clustering Node.....	61
2.8.5 Cluster Node	69
2.8.6 Variable Selection Node.....	72
2.9 Tools for Data Modification	79
2.9.1 Drop Node	79
2.9.2 Replacement Node.....	80
2.9.3 Impute Node.....	83
2.9.4 Interactive Binning Node	83
2.9.5 Principal Components Node	90
2.9.6 Transform Variables Node.....	95
2.10 Utility Nodes	101
2.10.1 SAS Code Node	101
2.11 Appendix to Chapter 2.....	107
2.11.1 The Type, the Measurement Scale, and the Number of Levels of a Variable	107
2.11.2 Eigenvalues, Eigenvectors, and Principal Components.....	110
2.11.3 Cramer's V.....	113
2.11.4 Calculation of Chi-Square Statistic and Cramer's V for a Continuous Input.....	113
2.12 Exercises.....	115
Chapter 3: Variable Selection and Transformation of Variables.....	117
3.1 Introduction	117
3.2 Variable Selection	118
3.2.1 Continuous Target with Numeric Interval-scaled Inputs (Case 1)	119
3.2.2 Continuous Target with Nominal-Categorical Inputs (Case 2).....	124
3.2.3 Binary Target with Numeric Interval-scaled Inputs (Case 3)	129
3.2.4 Binary Target with Nominal-scaled Categorical Inputs (Case 4)	135
3.3 Variable Selection Using the Variable Clustering Node.....	138
3.3.1 Selection of the Best Variable from Each Cluster.....	140
3.3.2 Selecting the Cluster Components.....	148
3.4 Variable Selection Using the Decision Tree Node.....	150
3.5 Transformation of Variables	153
3.5.1 Transform Variables Node.....	153
3.5.2 Transformation before Variable Selection	155
3.5.3 Transformation after Variable Selection	157
3.5.4 Passing More Than One Type of Transformation for Each Interval Input to the Next Node.....	159
3.5.5 Saving and Exporting the Code Generated by the Transform Variables Node.....	163

3.6 Summary	163
3.7 Appendix to Chapter 3.....	164
3.7.1 Changing the Measurement Scale of a Variable in a Data Source	164
3.7.2 SAS Code for Comparing Grouped Categorical Variables with the Ungrouped Variables	165
Exercises.....	166
Note	167
Chapter 4: Building Decision Tree Models to Predict Response and Risk.....	169
4.1 Introduction	170
4.2 An Overview of the Tree Methodology in SAS Enterprise Miner	170
4.2.1 Decision Trees	170
4.2.2 Decision Tree Models	170
4.2.3 Decision Tree Models vs. Logistic Regression Models.....	172
4.2.4 Applying the Decision Tree Model to Prospect Data.....	173
4.2.5 Calculation of the Worth of a Tree.....	173
4.2.6 Roles of the Training and Validation Data in the Development of a Decision Tree.....	175
4.2.7 Regression Tree	176
4.3 Development of the Tree in SAS Enterprise Miner.....	176
4.3.1 Growing an Initial Tree	176
4.3.2 P-value Adjustment Options	183
4.3.3 Controlling Tree Growth: Stopping Rules	185
4.3.4 Pruning: Selecting the Right-Sized Tree Using Validation Data.....	185
4.3.5 Step-by-Step Illustration of Growing and Pruning a Tree.....	188
4.3.6 Average Profit vs. Total Profit for Comparing Trees of Different Sizes.....	192
4.3.7 Accuracy /Misclassification Criterion in Selecting the Right-sized Tree (Classification of Records and Nodes by Maximizing Accuracy).....	193
4.3.8 Assessment of a Tree or Sub-tree Using Average Square Error.....	194
4.3.9 Selection of the Right-sized Tree	194
4.4 A Decision Tree Model to Predict Response to Direct Marketing.....	195
4.4.1 Testing Model Performance with a Test Data Set	204
4.4.2 Applying the Decision Tree Model to Score a Data Set	205
4.5 Developing a Regression Tree Model to Predict Risk	208
4.5.1 Summary of the Regression Tree Model to Predict Risk.....	214
4.6 Developing Decision Trees Interactively	215
4.6.1 Interactively Modifying an Existing Decision Tree	215
4.6.2 Growing a Tree Interactively Starting from the Root Node	225
4.6.3 Developing the Maximal Tree in Interactive Mode	231
4.7 Summary	233
4.8 Appendix to Chapter 4.....	234
4.8.1 Pearson's Chi-Square Test.....	234
4.8.2 Adjusting the Predicted Probabilities for Over-sampling	235
4.8.3 Expected Profits Using Unadjusted Probabilities	236
4.8.4 Expected Profits Using Adjusted Probabilities	236
4.9 Exercises.....	236

Chapter 5: Neural Network Models to Predict Response and Risk	239
5.1 Introduction	240
5.1.1 Target Variables for the Models.....	240
5.1.2 Neural Network Node Details.....	240
5.2 A General Example of a Neural Network Model	241
5.2.1 Input Layer	242
5.2.2 Hidden Layers	242
5.2.3 Output Layer or Target Layer	246
5.2.4 Activation Function of the Output Layer	247
5.3 Estimation of Weights in a Neural Network Model.....	247
5.4 A Neural Network Model to Predict Response	249
5.4.1 Setting the Neural Network Node Properties	250
5.4.2 Assessing the Predictive Performance of the Estimated Model	254
5.4.3 Receiver Operating Characteristic (ROC) Charts	258
5.4.4 How Did the Neural Network Node Pick the Optimum Weights for This Model?.....	261
5.4.5 Scoring a Data Set Using the Neural Network Model	263
5.4.6 Score Code.....	266
5.5 A Neural Network Model to Predict Loss Frequency in Auto Insurance.....	266
5.5.1 Loss Frequency as an Ordinal Target	267
5.5.3 Classification of Risks for Rate Setting in Auto Insurance with Predicted Probabilities	279
5.6 Alternative Specifications of the Neural Networks	279
5.6.1 A Multilayer Perceptron (MLP) Neural Network	279
5.6.2 A Radial Basis Function (RBF) Neural Network	281
5.7 Comparison of Alternative Built-in Architectures of the Neural Network Node	286
5.7.1 Multilayer Perceptron (MLP) Network.....	287
5.7.2 Ordinary Radial Basis Function with Equal Heights and Widths (ORBFEQ)	288
5.7.3 Ordinary Radial Basis Function with Equal Heights and Unequal Widths (ORBFUN).....	291
5.7.4 Normalized Radial Basis Function with Equal Widths and Heights (NRBFEQ).....	292
5.7.5 Normalized Radial Basis Function with Equal Heights and Unequal Widths (NRBFEH). 295	
5.7.6 Normalized Radial Basis Function with Equal Widths and Unequal Heights (NRBFEW) 297	
5.7.7 Normalized Radial Basis Function with Equal Volumes (NRBFEV)	300
5.7.8 Normalized Radial Basis Function with Unequal Widths and Heights (NRBFUN).....	302
5.7.9 User-Specified Architectures	305
5.8 AutoNeural Node.....	307
5.9 DMNeural Node	309
5.10 Dmine Regression Node	312
5.11 Comparing the Models Generated by DMNeural, AutoNeural, and Dmine Regression Nodes	314
5.12 Summary	316
5.13 Appendix to Chapter 5.....	317
5.14 Exercises.....	318
Chapter 6: Regression Models	321
6.1 Introduction	321

6.2 What Types of Models Can Be Developed Using the Regression Node?	321
6.2.1 Models with a Binary Target	321
6.2.2 Models with an Ordinal Target	324
6.2.3 Models with a Nominal (Unordered) Target	329
6.2.4 Models with Continuous Targets	333
6.3 An Overview of Some Properties of the Regression Node	333
6.3.1 Regression Type Property	333
6.3.2 Link Function Property	333
6.3.3 Selection Model Property	335
6.3.4 Selection Criterion Property	348
6.4 Business Applications	358
6.4.1 Logistic Regression for Predicting Response to a Mail Campaign	359
6.4.2 Regression for a Continuous Target	371
6.5 Summary	379
6.6 Appendix to Chapter 6	380
6.6 Exercises	382
Chapter 7: Comparison and Combination of Different Models	383
7.1 Introduction	383
7.2 Models for Binary Targets: An Example of Predicting Attrition	384
7.2.1 Logistic Regression for Predicting Attrition	386
7.2.2 Decision Tree Model for Predicting Attrition	387
7.2.3 A Neural Network Model for Predicting Attrition	389
7.3 Models for Ordinal Targets: An Example of Predicting the Risk of Accident Risk	392
7.3.1 Lift Charts and Capture Rates for Models with Ordinal Targets	393
7.3.2 Logistic Regression with Proportional Odds for Predicting Risk in Auto Insurance	394
7.3.3 Decision Tree Model for Predicting Risk in Auto Insurance	396
7.3.4 Neural Network Model for Predicting Risk in Auto Insurance	400
7.4 Comparison of All Three Accident Risk Models	401
7.5 Boosting and Combining Predictive Models	402
7.5.1 Gradient Boosting	402
7.5.2 Stochastic Gradient Boosting	404
7.5.3 An Illustration of Boosting Using the Gradient Boosting Node	404
7.5.4 The Ensemble Node	407
7.5.5 Comparing the Gradient Boosting and Ensemble Methods of Combining Models	410
7.6 Appendix to Chapter 7	411
7.6.1 Least Squares Loss	411
7.6.2 Least Absolute Deviation Loss	411
7.6.3 Huber-M Loss	411
7.6.4 Logit Loss	412
7.7 Exercises	412
Chapter 8: Customer Profitability	415
8.1 Introduction	415
8.2 Acquisition Cost	417
8.3 Cost of Default	418

8.4 Revenue	419
8.5 Profit	419
8.6 The Optimum Cut-off Point.....	421
8.7 Alternative Scenarios of Response and Risk.....	422
8.8 Customer Lifetime Value	422
8.9 Suggestions for Extending Results	423
Chapter 9: Introduction to Predictive Modeling with Textual Data	425
9.1 Introduction	425
9.1.1 Quantifying Textual Data: A Simplified Example.....	426
9.1.2 Dimension Reduction and Latent Semantic Indexing	429
9.1.3 Summary of the Steps in Quantifying Textual Information	431
9.2 Retrieving Documents from the World Wide Web.....	432
9.2.1 The %TMFILTER Macro.....	432
9.3 Creating a SAS Data Set from Text Files.....	433
9.4 The Text Import Node.....	436
9.5 Creating a Data Source for Text Mining	436
9.6 Text Parsing Node.....	436
9.7 Text Filter Node.....	440
9.7.1 Frequency Weighting	440
9.7.2 Term Weighting.....	440
9.7.3 Adjusted Frequencies	441
9.7.4 Frequency Weighting Methods	441
9.7.5 Term Weighting Methods	441
9.8 Text Topic Node	445
9.8.1 Developing a Predictive Equation Using the Output Data Set Created by the Text Topic Node.....	449
9.9 Text Cluster Node	450
9.9.1 Hierarchical Clustering	451
9.9.2 Expectation-Maximization (EM) Clustering	452
9.9.3 Using the Text Cluster Node	458
9.10 Exercises.....	461
Index.....	463



From *Predictive Modeling with SAS[®] Enterprise Miner[™]*,
Second Edition. Full book available for purchase [here](#).

Note: This short excerpt was taken from a chapter in the published book.

5.4 A Neural Network Model to Predict Response

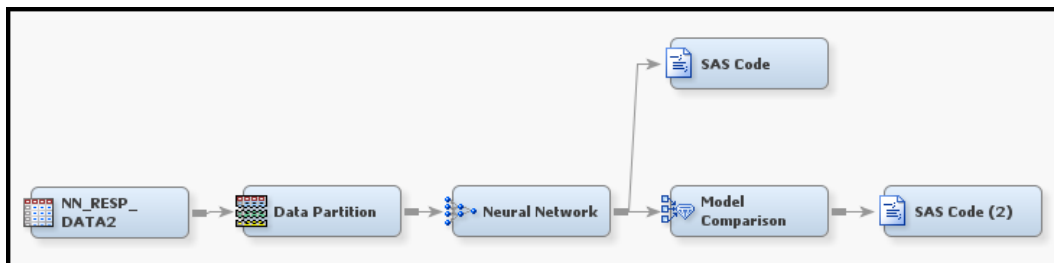
This section discusses the neural network model developed to predict the response to a planned direct mail campaign. The campaign's purpose is to solicit customers for a hypothetical insurance company. A two-layered network with one hidden layer was chosen. Three units are included in the hidden layer. In the hidden layer, the combination function chosen is linear, and the activation function is hyperbolic tangent. In the output layer, a logistic activation function and Bernoulli error function are used. The logistic activation function results in a logistic regression type model with non-linear transformation of the inputs, as shown in Equation 5.14 in Section 5.2.4. Models of this type are in general estimated by minimizing the Bernoulli error functions shown in Equation 5.16. Minimization of the Bernoulli error function is equivalent to maximizing the likelihood function.

Display 5.1 shows the process flow for the response model. The first node in the process flow diagram is the **Input Data** node, which makes the SAS data set available for modeling. The next node is **Data Partition**, which creates the Training, Validation, and Test data sets. The Training data set is used for preliminary model fitting. The Validation data set is used for selecting the optimum weights. The **Model Selection Criterion** property is set to Average Error.

As pointed out earlier, the estimation of the weights is done by minimizing the error function. This minimization is done by an iterative procedure. Each iteration yields a set of weights. Each set of weights defines a model. If I set the **Model Selection Criterion** property to Average Error, the algorithm selects the set of weights that results in the smallest error, where the error is calculated from the Validation data set.

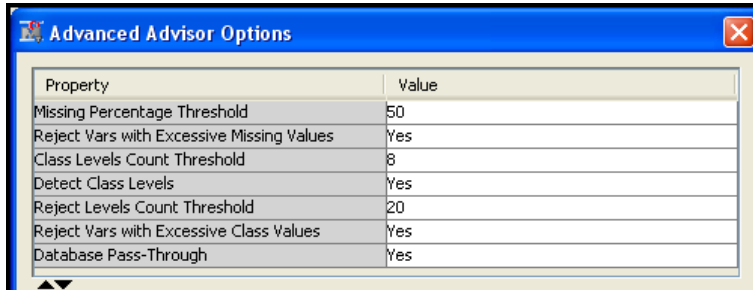
Since both the Training and Validation data sets are used for parameter estimation and parameter selection, respectively, an additional holdout data set is required for an independent assessment of the model. The Test data set is set aside for this purpose.

Display 5.1



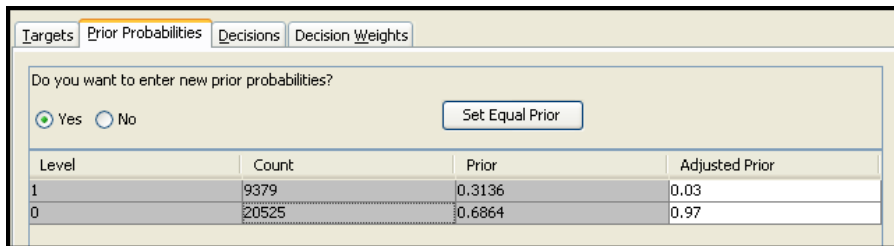
Input Data Node

I create the data source for the **Input Data** node from the data set NN_RESP_DATA2. I create the metadata using the Advanced Advisor Options, and I customize it by setting the **Class Levels Count Threshold** property to 8, as shown in Display 5.2

Display 5.2


Property	Value
Missing Percentage Threshold	50
Reject Vars with Excessive Missing Values	Yes
Class Levels Count Threshold	8
Detect Class Levels	Yes
Reject Levels Count Threshold	20
Reject Vars with Excessive Class Values	Yes
Database Pass-Through	Yes

I set adjusted prior probabilities to 0.03 for response and 0.97 for non-response, as shown in Display 5.3.

Display 5.3


Level	Count	Prior	Adjusted Prior
1	9379	0.3136	0.03
0	20525	0.6864	0.97

Data Partition Node

The input data is partitioned such that 60% of the observations are allocated for training, 30% for validation, and 10% for Test, as shown in Display 5.4.

Display 5.4

Data Set Allocations	
Training	60.0
Validation	30.0
Test	10.0

5.4.1 Setting the Neural Network Node Properties

Here is a summary of the neural network specifications for this application:


- One hidden layer with three neurons
- Linear combination functions for both the hidden and output layers
- Hyperbolic tangent activation functions for the hidden units
- Logistic activation functions for the output units
- The Bernoulli error function
- The **Model Selection Criterion** is Average Error

These settings are shown in Displays 5.5–5.7.

Display 5.5 shows the Properties panel for the **Neural Network** node.

Display 5.5

Property	Value
General	
Node ID	Neural
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Continue Training	No
Network	
Optimization	
Initialization Seed	12345
Model Selection Criterion	Average Error
Suppress Output	No
Score	
Hidden Units	No
Residuals	Yes
Standardization	No
Status	
Create Time	1/8/13 7:59 AM

To define the network architecture, click  located to the right of the **Network** property. The Network Properties panel opens, as shown in Display 5.6.


Display 5.6

Property	Value
Architecture	User
Direct Connection	No
Number of Hidden Units	3
Randomization Distribution	Normal
Randomization Center	0.0
Randomization Scale	0.1
Input Standardization	Standard Deviation
Hidden Layer Combination Function	Linear
Hidden Layer Activation Function	Hyperbolic Tangent
Hidden Bias	Yes
Target Layer Combination Function	Linear
Target Layer Activation Function	Logistic
Target Layer Error Function	Bernoulli
Target Bias	Yes
Weight Decay	0.0

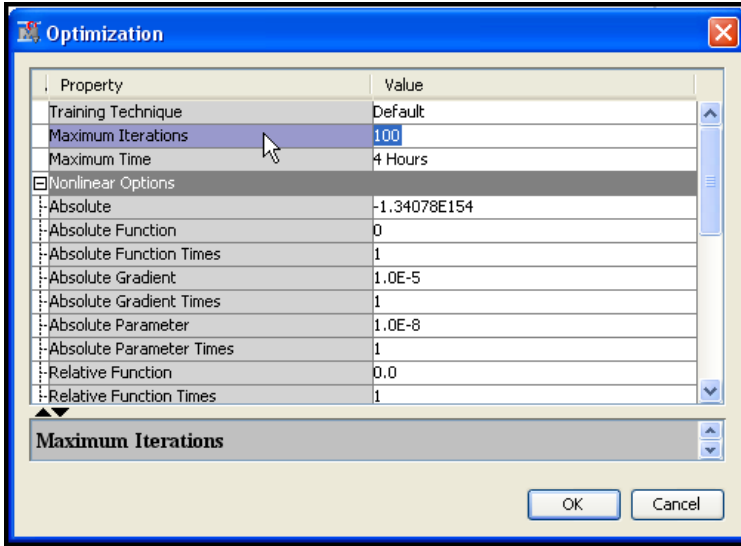
Architecture

OK Cancel

Set the properties as shown in Display 5.6 and click **OK**.

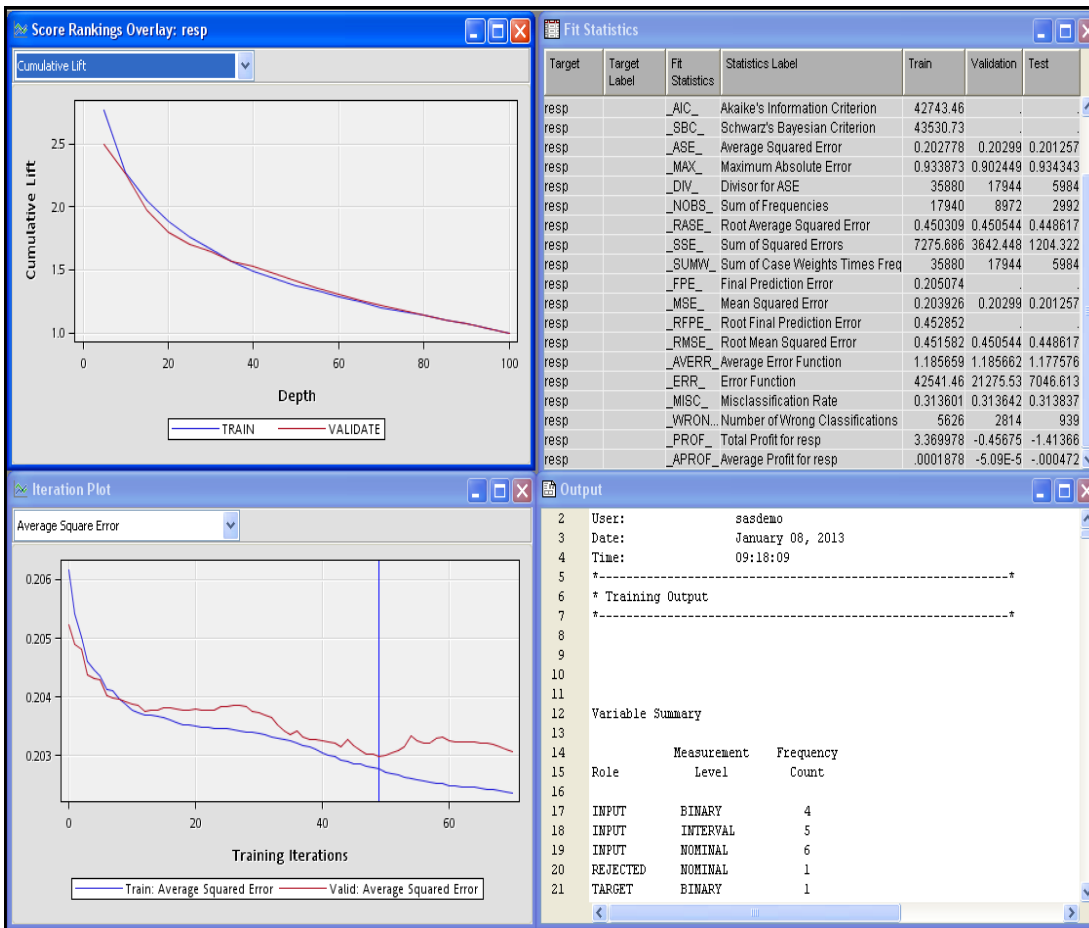
To set the iteration limit, click  located to the right of the **Optimization** property. The Optimization Properties panel opens, as shown in Display 5.7. Set **Maximum Iterations** to 100.

Display 5.7



After running the **Neural Network** node, you can open the Results window, shown in Display 5.8. The window contains four windows: Score Rankings Overlay, Iteration Plot, Fit Statistics, and Output.

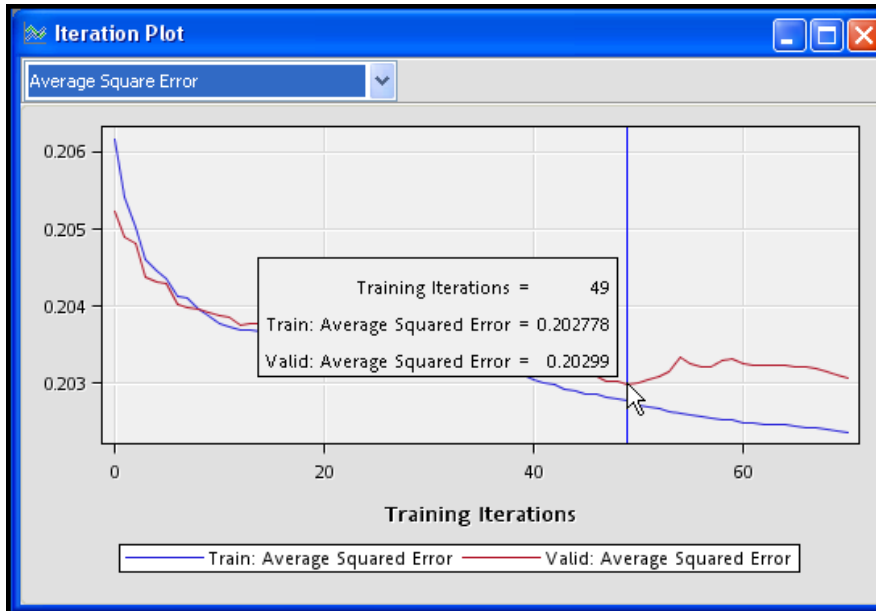
Display 5.8



The Score Rankings Overlay window in Display 5.8 shows the cumulative lift for the Training and Validation data sets. Click the down arrow next to the text box to see a list of available charts that can be displayed in this window.

Display 5.9 shows the iteration plot with Average Squared Error at each iteration for the Training and Validation data sets. The estimation process required 70 iterations. The weights from the 49th iteration were selected. After the 49th iteration, the Average Squared Error started to increase in the Validation data set, although it continued to decline in the Training data set.

Display 5.9



You can save the table corresponding to the plot shown in Display 5.9 by clicking the **Tables** icon and then selecting **File**→**Save As**. Table 5.1 shows the three variables `_ITER_` (iteration number), `_ASE_` (Average Squared Error for the Training data), and `_VASE_` (Average Squared Error from the Validation data) at iterations 41-60.

Table 5.1

Training Iterations	Train: Average Squared Error	Valid: Average Squared Error
41	0.20300	0.20324
42	0.20298	0.20321
43	0.20293	0.20314
44	0.20291	0.20328
45	0.20287	0.20316
46	0.20285	0.20308
47	0.20282	0.20303
48	0.20280	0.20303
49	0.20278	0.20299
50	0.20272	0.20301
51	0.20268	0.20304
52	0.20267	0.20308
53	0.20263	0.20314
54	0.20260	0.20334
55	0.20258	0.20325
56	0.20258	0.20320
57	0.20255	0.20322
58	0.20253	0.20330
59	0.20252	0.20332
60	0.20248	0.20326

You can print the variables `_ITER_`, `_ASE_`, and `_VASE_` by using the SAS code shown in Display 5.10.

Display 5.10

```

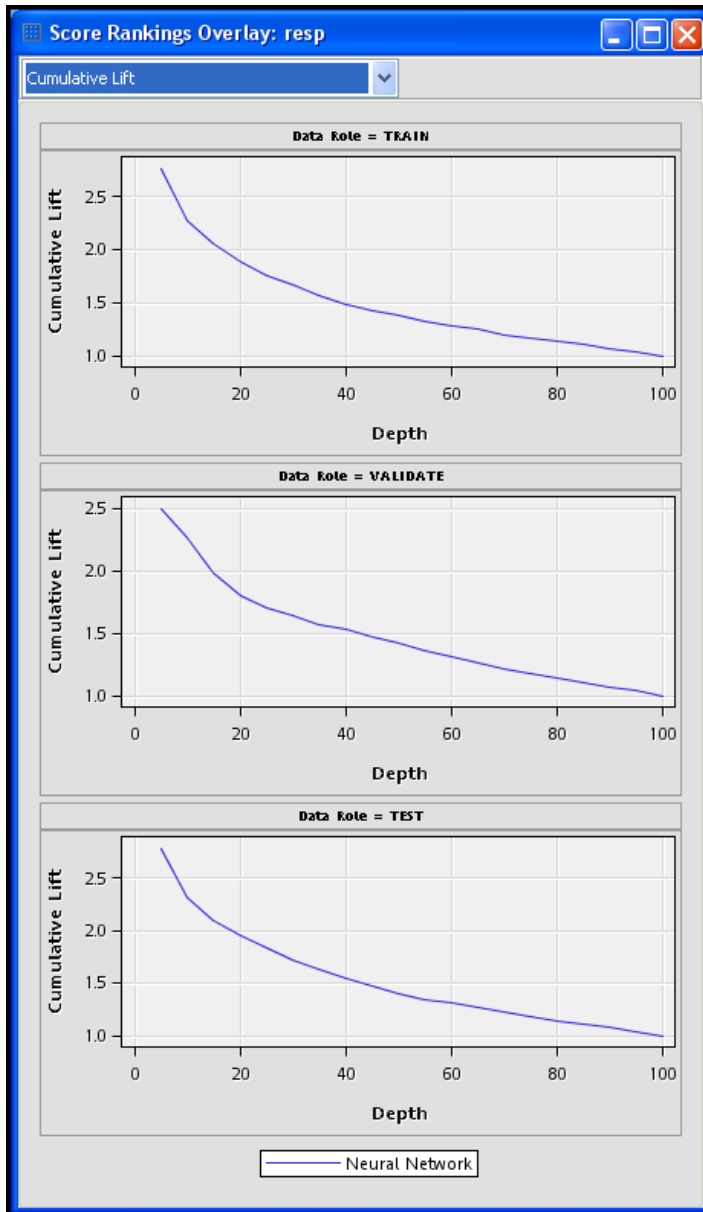
Training Code
proc print data=&em_lib..neural_plotds noobs label;
  var _ITER_ _ASE_ _VASE_ ;
  where 41 le _ITER_ le 60;
run;

```

5.4.2 Assessing the Predictive Performance of the Estimated Model

In order to assess the predictive performance of the neural network model, run the **Model Comparison** node and open the Results window. In the Results window, the Score Rankings Overlay shows the Lift charts for the Training, Validation, and Test data sets. These are shown in Display 5.11.

Display 5.11



Click the arrow in the box at the top left corner of the Score Ranking Overlay window to see a list of available charts.

SAS Enterprise Miner saves the Score Rankings table as EMWS.MdlComp_EMRANK. Tables 5.2, 5.3, and 5.4 are created from the saved data set using the simple SAS code shown in Display 5.12.

Display 5.12

```

Training Code
options center ;
proc print data=&EM_LIB..MdlComp_EMRank label noobs ;
  where upcase(datarole) = "TRAIN" and bin ne . ;
  var bin decile resp respc lift liftc cap capc ;
  title "Lift and Capture Rates: Training Data set";
run;
proc print data=&EM_LIB..MdlComp_EMRank label noobs;
  where upcase(datarole) = "VALIDATE" and bin ne . ;
  var bin decile resp respc lift liftc cap capc ;
  title "Lift and Capture Rates: Validation Data set";
run;
proc print data=&EM_LIB..MdlComp_EMRank label noobs;
  where upcase(datarole) = "TEST" and bin ne . ;
  var bin decile resp respc lift liftc cap capc ;
  title "Lift and Capture Rates: Test Data set";
run;
    
```

Table 5.2

Lift and Capture Rates: Training Data set							
Bin	Depth	% Response	Cumulative % Response	Lift	Cumulative Lift	% Captured Response	Cumulative % Captured Response
1	5	8.31775	8.31775	2.77258	2.77258	13.8642	13.8642
2	10	5.31131	6.81464	1.77044	2.27155	8.8518	22.7160
3	15	4.84188	6.15707	1.61396	2.05236	8.0697	30.7856
4	20	4.12431	5.64861	1.37477	1.88287	6.8788	37.6644
5	25	3.83131	5.28540	1.27710	1.76180	6.3811	44.0455
6	30	3.58335	5.00173	1.19445	1.66724	5.9723	50.0178
7	35	2.91134	4.70309	0.97045	1.56770	4.8525	54.8702
8	40	2.78073	4.46259	0.92691	1.48753	4.6392	59.5094
9	45	2.91268	4.29046	0.97089	1.43015	4.8525	64.3619
10	50	2.64467	4.12588	0.88156	1.37529	4.4081	68.7700
11	55	2.61352	3.98844	0.87117	1.32948	4.3548	73.1248
12	60	2.46397	3.86142	0.82132	1.28714	4.1059	77.2307
13	65	2.34569	3.74481	0.78190	1.24827	3.9104	81.1411
14	70	1.71661	3.59991	0.57220	1.19997	2.8617	84.0028
15	75	2.15451	3.50356	0.71817	1.16785	3.5905	87.5933
16	80	1.97366	3.40798	0.65789	1.13599	3.2883	90.8816
17	85	1.67467	3.30604	0.55822	1.10201	2.7906	93.6722
18	90	1.58819	3.21056	0.52940	1.07019	2.6484	96.3207
19	95	1.34421	3.11236	0.44807	1.03745	2.2396	98.5603
20	100	0.86420	3.00000	0.28807	1.00000	1.4397	100.000

Table 5.3

Lift and Capture Rates: Validation Data set							
Bin	Depth	% Response	Cumulative % Response	Lift	Cumulative Lift	% Captured Response	Cumulative % Captured Response
1	5	7.49586	7.49586	2.49862	2.49862	12.5089	12.5089
2	10	6.05928	6.77825	2.01976	2.25942	10.0924	22.6013
3	15	4.24246	5.93308	1.41415	1.97769	7.0718	29.6731
4	20	3.84086	5.41043	1.28029	1.80348	6.3966	36.0697
5	25	3.85782	5.09981	1.28594	1.69994	6.4321	42.5018
6	30	4.00842	4.91792	1.33614	1.63931	6.6809	49.1827
7	35	3.42772	4.70477	1.14257	1.56826	5.7214	54.9041
8	40	3.65246	4.57347	1.21749	1.52449	6.0768	60.9808
9	45	3.11138	4.41094	1.03713	1.47031	5.1883	66.1692
10	50	2.78928	4.24858	0.92976	1.41619	4.6553	70.8244
11	55	2.26190	4.06814	0.75397	1.35605	3.7669	74.5913
12	60	2.28275	3.91945	0.76092	1.30648	3.8024	78.3937
13	65	2.08817	3.77850	0.69606	1.25950	3.4826	81.8763
14	70	1.90004	3.64450	0.63335	1.21483	3.1628	85.0391
15	75	2.04643	3.53794	0.68214	1.17931	3.4115	88.4506
16	80	1.59942	3.41680	0.53314	1.13893	2.6652	91.1158
17	85	1.61828	3.31087	0.53943	1.10362	2.7008	93.8166
18	90	1.57989	3.21483	0.52663	1.07161	2.6297	96.4463
19	95	1.32169	3.11518	0.44056	1.03839	2.2033	98.6496
20	100	0.81060	3.00000	0.27020	1.00000	1.3504	100.000

Table 5.4

Lift and Capture Rates: Test Data set							
Bin	Depth	% Response	Cumulative % Response	Lift	Cumulative Lift	% Captured Response	Cumulative % Captured Response
1	5	8.36796	8.36796	2.78932	2.78932	13.9510	13.9510
2	10	5.49570	6.93212	1.83190	2.31071	9.1587	23.1097
3	15	5.02337	6.29385	1.67446	2.09795	8.4132	31.5229
4	20	4.47671	5.84044	1.49224	1.94681	7.4547	38.9776
5	25	4.17028	5.50778	1.39009	1.83593	6.9223	45.8999
6	30	3.50818	5.17403	1.16939	1.72468	5.8573	51.7572
7	35	3.12520	4.88096	1.04173	1.62699	5.2183	56.9755
8	40	3.01327	4.64829	1.00442	1.54943	5.0053	61.9808
9	45	2.54180	4.41310	0.84727	1.47103	4.2599	66.2407
10	50	2.30884	4.20349	0.76961	1.40116	3.8339	70.0745
11	55	2.17732	4.01970	0.72577	1.33990	3.6209	73.6954
12	60	3.00147	3.93481	1.00049	1.31160	5.0053	78.7007
13	65	2.03820	3.78849	0.67940	1.26283	3.4079	82.1086
14	70	2.17732	3.67367	0.72577	1.22456	3.6209	85.7295
15	75	1.52626	3.52989	0.50875	1.17663	2.5559	88.2854
16	80	1.66239	3.41329	0.55413	1.13776	2.7689	91.0543
17	85	1.86309	3.32260	0.62103	1.10753	3.0884	94.1427
18	90	1.85065	3.24072	0.61688	1.08024	3.0884	97.2311
19	95	1.14908	3.13055	0.38303	1.04352	1.9169	99.1480
20	100	0.51256	3.00000	0.17085	1.00000	0.8520	100.000

The lift and capture rates calculated from the Test data set (shown in Table 5.4) should be used for evaluating the models or comparing the models because the Test data set is not used in training or fine-tuning the model.

To calculate the lift and capture rates, SAS Enterprise Miner first calculates the predicted probability of response for each record in the Test data. Then it sorts the records in descending order of the predicted probabilities (also called the scores) and divides the data set into 20 groups of equal size. In Table 5.4, the column Bin shows the ranking of these groups. If the model is accurate, the table should show the highest actual

response rate in the first bin, the second highest in the next bin, and so on. From the column %Response, it is clear that the average response rate for observations in the first bin is 8.36796%. The average response rate for the entire test data set is 3%. Hence the lift for Bin 1, which is the ratio of the response rate in Bin 1 to the overall response rate, is 2.7893. The lift for each bin is calculated in the same way. The first row of the column Cumulative %Response shows the response rate for the first bin. The second row shows the response rate for bins 1 and 2 combined, and so on.

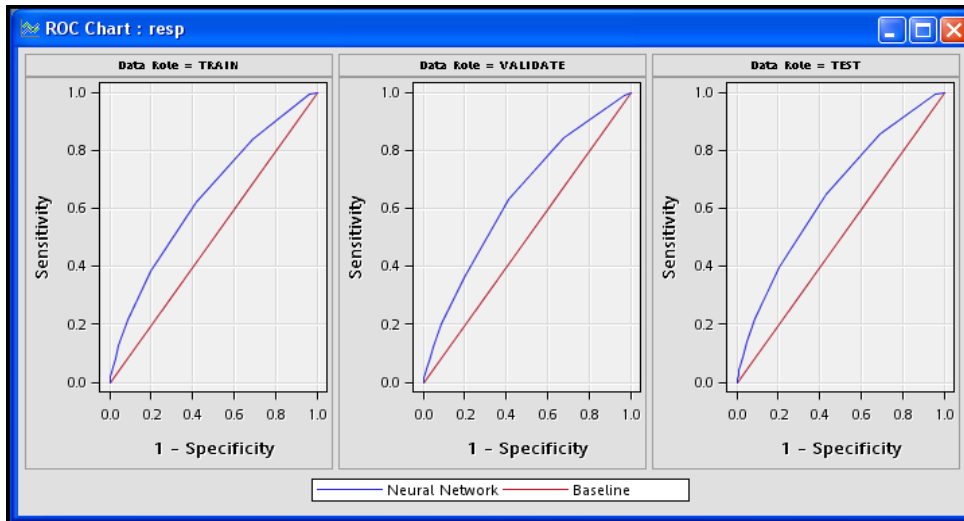
The capture rate of a bin shows the percentage of likely responders that it is reasonable to expect to be captured in the bin. From the column Captured Response, you can see that 13.951% of all responders are in Bin 1.

From the Cumulative % Captured Response column of Table 5.3, you can be seen that, by sending mail to customers in the first four bins, or the top 20% of the target population, it is reasonable to expect to capture 39% of all potential responders from the target population. This assumes that the modeling sample represents the target population.

5.4.3 Receiver Operating Characteristic (ROC) Charts

Display 5.13, taken from the Results window of the **Model Comparison** node, displays ROC curves for the Training, Validation, and Test data sets. An ROC curve shows the values of the *true positive fraction* and the *false positive fraction* at different *cut-off values*, which can be denoted by P_c . In the case of the response model, if the estimated probability of response for a customer record were above a cut-off value P_c , then you would classify the customer as a responder; otherwise, you would classify the customer as a non-responder.

Display 5.13



In the ROC chart, the true positive fraction is shown on the vertical axis, and the false positive fraction is on the horizontal axis for each cut-off value (P_c).

If the calculated probability of response (P_{resp1}) is greater than equal to the cut-off value, then the customer (observation) is classified as a responder. Otherwise, the customer is classified as non-responder.

True positive fraction is the proportion of responders correctly classified as responders. The false positive fraction is the proportion of non-responders incorrectly classified as responders. The true positive fraction is also called *sensitivity*, and *specificity* is the proportion of non-responders correctly classified as non-responders. Hence, the false positive fraction is 1-specificity. An ROC curve reflects the tradeoff between sensitivity and specificity.

The straight diagonal lines in Display 5.13 that are labeled Baseline are the ROC charts of a model that assigns customers at random to the responder group and the non-responder group, and hence has no predictive power. On these lines, sensitivity = 1 - specificity at all cut-off points. The larger the area between the ROC curve of the model being evaluated and the diagonal line, the better the model. The area under the ROC curve is a measure of the predictive accuracy of the model and can be used for comparing different models.

Table 5.5 shows sensitivity and 1-specificity at various cut-off points in the validation data.

Table 5.5

ROC Table: Validation Data		
Cutoff	Sensitivity	1-Specificity
1.00000	0.00000	0.00000
0.26583	0.00036	0.00000
0.22075	0.00071	0.00000
0.15334	0.00071	0.00016
0.14645	0.00142	0.00016
0.13954	0.00284	0.00049
0.12783	0.00391	0.00097
0.11956	0.00640	0.00162
0.10987	0.00995	0.00325
0.09991	0.01741	0.00503
0.08999	0.02665	0.01007
0.07991	0.04655	0.01689
0.06999	0.07285	0.02598
0.05995	0.12296	0.04644
0.04999	0.20220	0.08834
0.03998	0.36283	0.19584
0.03000	0.63184	0.41410
0.02000	0.84115	0.68139
0.01000	0.98969	0.96963
0.00000	1.00000	1.00000

From Table 5.5, you can see that at a cut-off probability (P_c) of 0.02, for example, the sensitivity is 0.84115. That is, at this cut-off point, you will correctly classify 84.1% of responders as responders, but you will also *incorrectly* classify 68.1% of non-responders as responders, since 1-specificity at this point is 0.68139. If instead you chose a much higher cut-off point of $P_c = 0.13954$, you would classify 0.284% of true responders as responders and 0.049% of non-responders as responders. In this case, by increasing the cut-off probability beyond which you would classify an individual as a responder, you would be reducing the fraction of false positive decisions made, while, at the same time, also reducing the fraction of true positive decisions made. These pairings of a true positive fraction with a false positive fraction are plotted as the ROC curve for the VALIDATE case in Display 5.13.

The SAS macro in Display 5.14 demonstrates the calculation of the true positive rate (TPR) and the false positive rate (FPR) at the cut-off probability of 0.02.

Display 5.14

```

%macro roccalc(PC=);
ods html file = "C:/TheBook/EM12.1/Reports/Chapter5/RespRateV.html";
title "Validation Data Set";
proc freq data=&EM_LIB..MdlComp_Validate;
  table resp / noperc nocumperc out=tab1(keep= resp count rename=(count=N));
run;
ods html close ;
ods html file = "C:/TheBook/EM12.1/Reports/Chapter5/RespRateV_cutoff.html";
Title "Cases with Predicted Probability (P_respl) GE &PC";
title2 "Validation Data Set";
proc freq data=&EM_LIB..MdlComp_Validate;
  table resp / noperc nocumperc out=tab2(keep= resp count rename=(count=NC));
  where P_respl ge &PC;
run;
ods html close;
data temp;
merge tab1 tab2 ;
by resp ;
if resp=0 then TYPE='FPR' ; else if resp=1 then TYPE='TPR'; Rate= NC/N;
cutoff=&pc;
run;
ods html file = "C:/TheBook/EM12.1/Reports/Chapter5/TPR_FPR_Cutoff.html";
Title "True Positive Rate (TPR) and False Positive Rate (FPR)";
title2 "at cut-off = &PC";
proc print data=temp label noobs;
var RESP N NC TYPE RATE ;
label N = "Number of Observations in the sample";
label NC = "Number of Observations classified as responders ";
label RESP = "Actual RESPONSE ";
label TYPE = "ROC Coordinate ";
label Rate = "ROC Coordinate Value";
run;
ods html close;
%mend roccalc;
%roccalc(PC=0.02);

```

Tables 5.6, 5.7, and 5.8, generated by the macro shown in Display 5.14, show the sequence of steps for calculating TPR and FPR for a given cut-off probability.

Table 5.6

Validation Data Set	
<i>The FREQ Procedure</i>	
resp	Frequency
0	6158
1	2814

Table 5.7

Cases with Predicted Probability (P_{resp1}) GE 0.02
Validation Data Set

The FREQ Procedure

resp	Frequency
0	4196
1	2367

Table 5.8

True Positive Rate (TPR) and False Positive Rate (FPR)
at cut-off = 0.02

Actual RESPONSE	Number of Observations in the sample	Number of Observations classified as responders	ROC Coordinate	ROC Coordinate Value
0	6158	4196	FPR	0.68139
1	2814	2367	TPR	0.84115

For more information about ROC curves, see the textbook by A. A. Afifi and Virginia Clark (2004).⁴

Display 5.15 shows the SAS code that generated Table 5.5.

Display 5.15

```
proc print data=&EM_LIB..mdlcomp_emroc noobs label;
var cutoff sensitivity oneminusspecificity ;
where upcase(datarole) = 'VALIDATE' and upcase(model)="NEURAL" and cutoff ne .;
Title "ROC Table: Validation Data" ;
label cutoff = "Cutoff" sensitivity = "Sensitivity"
oneminusspecificity="1-Specificity";
run;
```

5.4.4 How Did the Neural Network Node Pick the Optimum Weights for This Model?

In Section 5.3, I described how the optimum weights are found in a neural network model. I described the two-step procedure of estimating and selecting the weights. In this section, I show the results of these two steps with reference to the neural network model discussed in Sections 5.4.1 and 5.4.2.

The weights such as those shown in Equations 5.1, 5.3, 5.5, 5.7, 5.9, 5.11, and 5.13 are shown in the Results window of the **Neural Network** node. You can see the estimated weights created at each iteration by opening the results window and selecting **View**→**Model**→**Weights-History**. Display 5.16 shows a partial view of the Weights-History window.

Display 5.16

Weights - History											
ITER	AGE -> HL1	CRED -> HL1	DELINQ -> HL1	MILEAGE -> HL1	NUMTR -> HL1	AGE -> HL2	CRED -> HL2	DELINQ -> HL2	MILEAGE -> HL2	NUMTR -> HL2	AGE -> HL3
30	0.051634	0.013581	-0.04942	0.037994	-0.09108	0.021394	0.015272	-0.01862	0.032955	0.072774	0.03261
31	0.05182	0.014726	-0.05917	0.039808	-0.09145	0.021723	0.013705	-0.02283	0.035434	0.080229	0.033779
32	0.04926	0.014514	-0.06559	0.035954	-0.09289	0.020158	0.013317	-0.01744	0.035331	0.083941	0.035385
33	0.049163	0.015336	-0.07253	0.036421	-0.09141	0.019859	0.012407	-0.01959	0.036852	0.086744	0.036088
34	0.049216	0.015535	-0.07133	0.035055	-0.08937	0.019039	0.012593	-0.01825	0.036613	0.084751	0.036105
35	0.047965	0.016614	-0.08615	0.035396	-0.09089	0.019267	0.010752	-0.02111	0.039862	0.09669	0.037988
36	0.046969	0.015786	-0.08104	0.033397	-0.09163	0.019507	0.011716	-0.01786	0.038681	0.093822	0.036828
37	0.045379	0.017111	-0.09285	0.031359	-0.09064	0.019137	0.010233	-0.01878	0.041725	0.101743	0.03723
38	0.043214	0.017856	-0.10395	0.029164	-0.09141	0.01911	0.009036	-0.01862	0.044226	0.110682	0.037984
39	0.041534	0.018032	-0.10769	0.026898	-0.09245	0.019452	0.008694	-0.01763	0.045423	0.115703	0.037584
40	0.039495	0.019087	-0.11634	0.023946	-0.09319	0.019732	0.007452	-0.01835	0.048368	0.124959	0.036921
41	0.041232	0.019089	-0.10932	0.025544	-0.09307	0.020193	0.007861	-0.02079	0.047792	0.12189	0.035542
42	0.042101	0.019884	-0.11178	0.026301	-0.09267	0.019756	0.007088	-0.02322	0.048779	0.12454	0.0359
43	0.044041	0.020461	-0.10627	0.025526	-0.09514	0.016812	0.006401	-0.02138	0.047231	0.122084	0.036323
44	0.043902	0.022069	-0.1106	0.022244	-0.09944	0.014736	0.004598	-0.02197	0.049496	0.131921	0.035225
45	0.044047	0.021174	-0.1079	0.023537	-0.09872	0.015295	0.005568	-0.02131	0.048206	0.127659	0.035611
46	0.044491	0.021269	-0.10816	0.023826	-0.09965	0.01462	0.005453	-0.02128	0.047965	0.127914	0.036005
47	0.045297	0.021834	-0.10742	0.022637	-0.10376	0.01277	0.004953	-0.0206	0.048183	0.130482	0.03537
48	0.048549	0.022946	-0.10315	0.023189	-0.11333	0.010181	0.004433	-0.02328	0.049119	0.135763	0.032797
49	0.046004	0.022182	-0.10612	0.022121	-0.10898	0.011822	0.004946	-0.02065	0.048699	0.13283	0.033759
50	0.047727	0.022403	-0.10271	0.023855	-0.11699	0.01153	0.005256	-0.02343	0.049256	0.135583	0.031076
51	0.048555	0.022612	-0.09836	0.023541	-0.12846	0.010614	0.005728	-0.02371	0.049642	0.1387	0.027236

The second column in Display 5.16 shows the weight of the variable AGE in hidden unit 1 at each iteration. The seventh column shows the weight of AGE in hidden unit 2 at each iteration. The twelfth column shows the weight of AGE in the third hidden unit. Similarly, you can trace through the weights of other variables. You can save the Weights-History table as a SAS data set.

To see the final weights, open the Results window. Select **View**→**Model**→**Weights_Final**. Then, click the **Table** icon. Selected rows of the final weights_table are shown in Display 5.17.

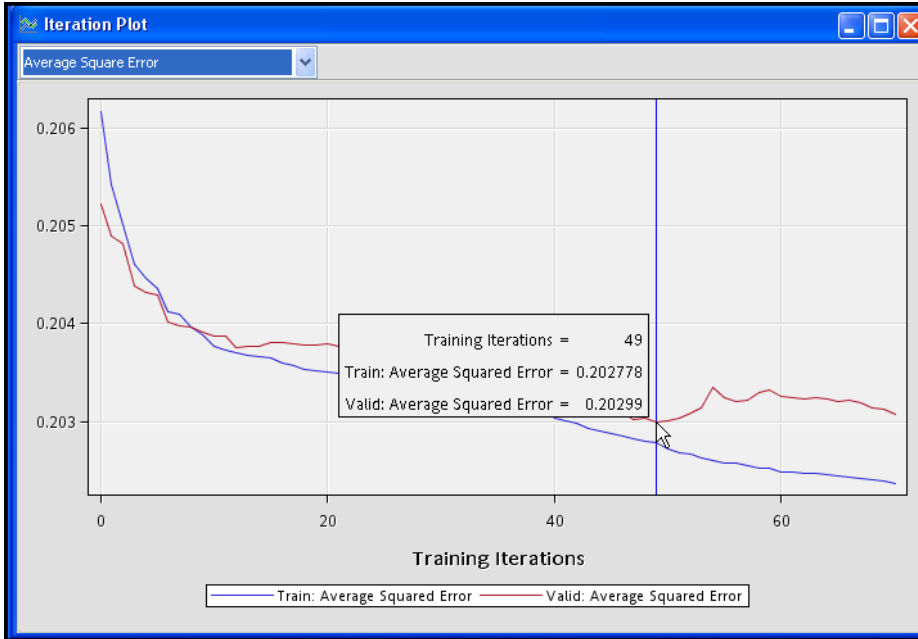
Display 5.17

LABEL	FROM	TO	WEIGHT
AGE -> HL1	AGE	HL1	0.04600
AGE -> HL2	AGE	HL2	0.01182
AGE -> HL3	AGE	HL3	0.03376
BIAS -> HL1	BIAS	HL1	0.21515
BIAS -> HL2	BIAS	HL2	-0.03256
BIAS -> HL3	BIAS	HL3	-0.09328
BIAS -> resp0	BIAS	resp0	1.02894
BIAS -> resp1	BIAS	resp1	-1.02894
CRED -> HL1	CRED	HL1	0.02218
CRED -> HL2	CRED	HL2	0.00495
CRED -> HL3	CRED	HL3	0.03979
HL1 -> resp0	HL1	resp0	1.29172
HL1 -> resp1	HL1	resp1	-1.29172
HL2 -> resp0	HL2	resp0	-1.59773
HL2 -> resp1	HL2	resp1	1.59773
HL3 -> resp0	HL3	resp0	1.56982
HL3 -> resp1	HL3	resp1	-1.56982

Outputs of the hidden units become inputs to the target layer. In the target layer, these inputs are combined using the weights estimated by the **Neural Network** node.

In the model I have developed, the weights generated at the 49th iteration are the optimal weights, because the Average Squared Error computed from the Validation data set reaches its minimum at the 49th iteration. This is shown in Display 5.18.

Display 5.18

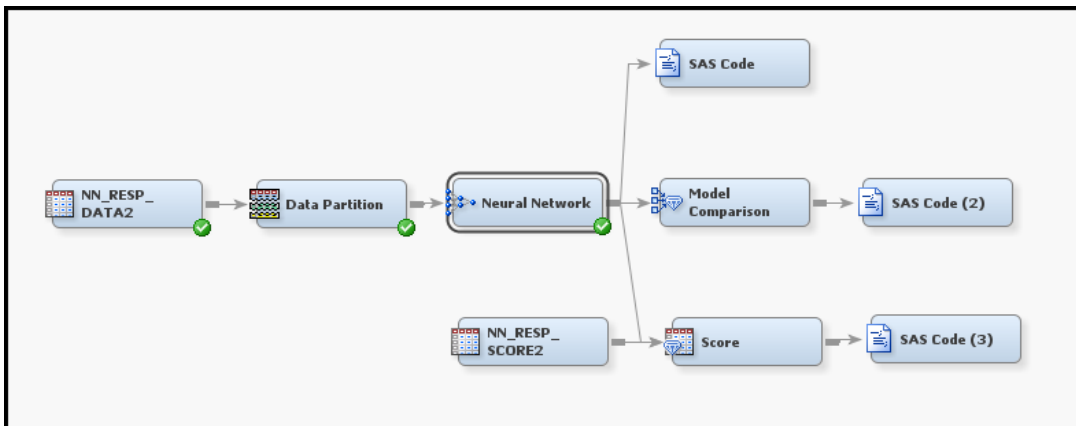


5.4.5 Scoring a Data Set Using the Neural Network Model

You can use the SAS code generated by the **Neural Network** node to score a data set within SAS Enterprise Miner or outside. This example scores a data set inside SAS Enterprise Miner.

The process flow diagram with a scoring data set is shown in Display 5.19.

Display 5.19



Set the **Role** property of the data set to be scored to Score, as shown in Display 5.20.

Display 5.20

Train	
Output Type	View
Role	Score
Rerun	No
Summarize	No
Drop Map Variables	No

The Score Node applies the SAS code generated by the **Neural Network** node to the Score data set NN_RESP_SCORE2, shown in Display 5.19.

For each record, the probability of response, the probability of non-response, and the expected profit of each record is calculated and appended to the scored data set.

Display 5.21 shows the segment of the score code where the probabilities of response and non-response are calculated. The coefficients of HL1, HL2, and HL3 in Display 5.21 are the weights in the final output layer. These are same as the coefficients shown in Display 5.17.

Display 5.21

```

P_resp1 = -1.29172482403562 * HL1 + 1.5977312209735 * HL2
+ -1.56981973510786 * HL3 ;
P_resp0 = 1.29172482403562 * HL1 + -1.5977312209735 * HL2
+ 1.56981973510786 * HL3 ;
P_resp1 = -1.0289412689995 + P_resp1 ;
P_resp0 = 1.0289412689995 + P_resp0 ;
DROP _EXP_BAR;
_EXP_BAR=50;
P_resp1 = 1.0 / (1.0 + EXP(MIN( - P_resp1 , _EXP_BAR)));
P_resp0 = 1.0 / (1.0 + EXP(MIN( - P_resp0 , _EXP_BAR)));

```

The code segment given in Display 5.21 calculates the probability of response using the

$$\text{formula } P_{_resp1}_i = \frac{1}{1 + \exp(-\eta_{i21})},$$

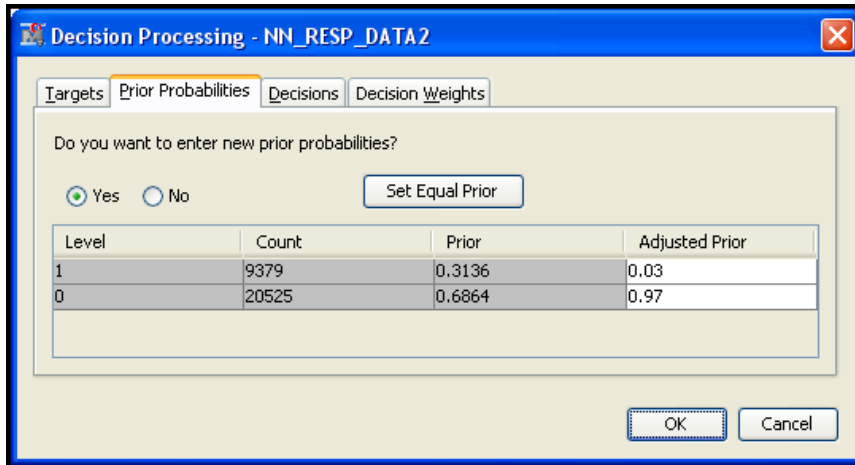
where $\mu_{i21} = -1.29172481842873 * HL1 + 1.59773122184585 * HL2$
 $+ -1.56981973539319 * HL3 - 1.0289412689995;$

This formula is the same as Equation 5.14 in Section 5.2.4. The subscript i is added to emphasize that this is a record-level calculation. In the code shown in Display 5.21, the probability of non-response is calculated

$$\text{as } p_{_resp0} = \frac{1}{1 + \exp(\eta_{i21})}.$$

The probabilities calculated above are modified by the prior probabilities I entered prior to running the **Neural Network** node. These probabilities are shown in Display 5.22.

Display 5.22



You can enter the prior probabilities when you create the **Data Source**. Prior probabilities are entered because the responders are overrepresented in the modeling sample, which is extracted from a larger sample. In the larger sample, the proportion of responders is only 3%. In the modeling sample, the proportion of responders is 31.36%. Hence, the probabilities should be adjusted before expected profits are computed. The SAS code generated by the **Neural Network** node and passed on to the **Score** node includes statements for making this adjustment. Display 5.23 shows these statements.

Display 5.23

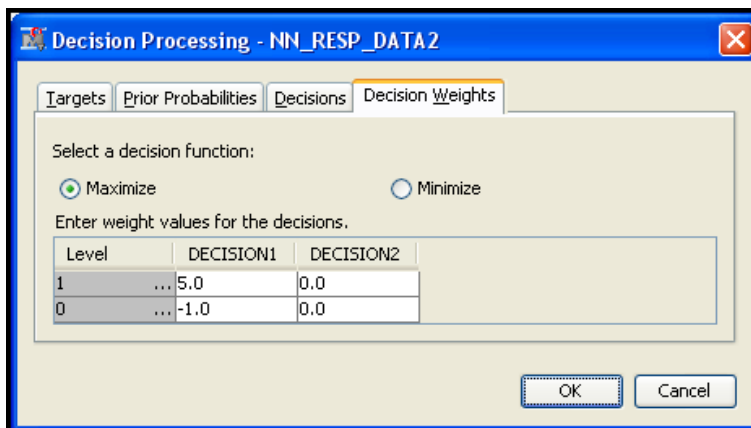
```

*** Update Posterior Probabilities;
P_respl = P_respl * 0.03 / 0.31368937998772;
P_resp0 = P_resp0 * 0.97 / 0.68631062001227;
drop _sum; _sum = P_respl + P_resp0 ;
if _sum > 4.135903E-25 then do;
  P_respl = P_respl / _sum;
  P_resp0 = P_resp0 / _sum;
end;

```

Display 5.24 shows the profit matrix used in the decision-making process.

Display 5.24



Given the above profit matrix, calculation of expected profit under the alternative decisions of classifying an individual as responder or non-responder proceeds as follows. Using the neural network model, the scoring algorithm first calculates the individual's probability of response and non-response. Suppose the calculated probability of response for an individual is 0.3, and probability of non-response is 0.7. The expected profit if the individual is classified as responder is $0.3 \times \$5 + 0.7 \times (-\$1.0) = \$0.8$. The expected profit if the individual is

classified as non-responder is $0.3x(\$0) + 0.7x(\$0) = \$0$. Hence classifying the individual as responder (Decision1) yields a higher profit than if the individual is classified as non-responder (Decision2). An additional field is added to the record in the scored data set indicating the decision to classify the individual as a responder.

These calculations are shown in the score code segment shown in Display 5.25.

Display 5.25

```

*** Decision Processing;
label D_RESP = 'Decision: resp' ;
label EP_RESP = 'Expected Profit: resp' ;

length D_RESP $ 9;

D_RESP = ' ';
EP_RESP = .;

*** Compute Expected Consequences and Choose Decision;
_decnum = 1; drop _decnum;

D_RESP = '1' ;
EP_RESP = P_respl * 5 + P_resp0 * -1;
drop _sum;
_sum = P_respl * 0 + P_resp0 * 0;
if _sum > EP_RESP + 2.273737E-12 then do;
    EP_RESP = _sum; _decnum = 2;
    D_RESP = '0' ;
end;

*** End Decision Processing ;

```

5.4.6 Score Code

The score code is automatically saved by the **Score** node in the sub-directory \Workspaces\EMWSn\Score within the project directory.

For example, in my computer, the Score code is saved by the **Score** node in the folder C:\TheBook\EM12.1\EMProjects\Chapter5\Workspaces\EMWS3\Score. Alternatively, you can save the score code in some other directory. To do so, run the Score node, and then click Results. Select either the Optimized SAS Code window or the SAS Code window. Click **File**→**Save As**, and enter the directory and name for saving the score code.

Note: This short excerpt was taken from a chapter in the published book.



From *Predictive Modeling with SAS[®] Enterprise Miner[™]*,
Second Edition. Full book available for purchase [here](#).

Index

A

accuracy criterion 193–194
acquisition cost 417–418
activation functions
 about 243, 322
 output layer 247
 target layer 270–272
Add value 306
adjusted frequencies 441
adjusted probabilities, expected profits using
 236
AIC (Akaike Information Criterion) 350–352
Append node 48–50, 116
Arc Tanget function 243–244
Architecture property
 about 316
 MLP setting 247
 Neural Network node 281, 283, 284, 293,
 295–297, 298–300, 305
 NRBFUN network 303–304
 Regression node 389
architectures
 alternative built-in 286–307
 of neural networks 316
 user-specified 305–307
Assessment Measure property 174, 187, 193,
 198, 387–389, 396–399
attrition, predicting 384–392
auto insurance industry, predicting risk in 3–4
AutoNeural node 307–309, 314–315, 316
Average method 408
average profit, vs. total profit for comparing tree
 size 192–193
average squared error 174, 194

B

β , as vector of coefficients 322

Backward Elimination method
 about 335
 when target is binary 335–337
 when target is continuous 338–340
bank deposit products, predicting rate sensitivity
 of 4–5
bin
 See groups
binary split search, splitting nodes using 176–
 177
binary targets
 Backward Elimination method with 335–
 337
 Forward Selection method with 340–342
 models for 384–392
 with nominal-scaled categorical inputs 135–
 138
 with numeric interval-scaled inputs 129–
 135
 regression models with 321–324
 stepwise selection method with 344–345
binning transformations 96–97
Bonferroni Adjustment property 183–184
Boolean retrieval method 427
branch 170
bucket 96
 See also groups
business applications
 logistic regression for predicting mail
 campaign response 359–371
 of regression models 358–379

C

calculating
 Chi-Square statistic for continuous input
 113–115
 cluster components 64
 Cramer's V for continuous input 113–115
 (continued)

- eigenvectors 111
- misclassification rate/accuracy rate 193–194
- principal components 112
- residuals 403–404
- validation profits 190–192
- worth of a tree 173–175
- worth of splits 177–182
- categorical variables 1–2, 165
- child nodes 170
- Chi-Square
 - calculating for continuous input 113–115
 - criterion for 130–135, 137–138
 - selection method 73
 - statistic 52, 53
 - test for 234
- Chi-Square property, StatExplore node 114
- class inputs, transformations of 98
- Class Inputs property, Transform Variables node 98, 99, 162, 376
- class interval
 - See* groups
- Class Levels Count Threshold property 19, 20, 108, 109, 164, 250, 267
- Cloglog 335
- Cluster Algorithm property 451–458
- Cluster node 50, 69–72, 451
- Cluster Variable Role property, Cluster node 70, 72
- Clustering Source property, Variable Clustering node 63
- clusters and clustering
 - assigning variables to 64–65
 - EM (Expectation-Maximization) 452–458, 460–461
 - hierarchical 451–459
 - selecting components 148–150
 - selecting variables for 140–148
- Code Editor property, SAS Code node 103–104
- combination functions 243, 270–272
- combining
 - groups 88–90
 - models 383–413
 - predictive models 402–411
- comparing
 - alternative built-in architectures of neural networks 286–307
 - categorical variables with ungrouped variables 165
 - gradient boosting and ensemble methods 410–411
 - models 383–413
 - models generated by DMNeural, AutoNeural, and Dmine Regression nodes 314–315
 - samples and targets 8
- Complementary Log-Log link (Cloglog) 335
- continuous input, calculating Chi-Square and Cramer's V for 113–115
- continuous targets
 - Backward Elimination method with 338–340
 - with Forward Selection method 342–343
 - with nominal-categorical inputs 124–129
 - with numeric interval-scaled inputs 119–124
 - regression for 371–379
 - regression models with 333
 - stepwise selection method with 345–347
- Correlations property, StatExplore node 55
- Cosine function 243
- cost of default 418–419
- Cramer's V 53–54, 113–115
- Cross Validation Error 355
- Cross Validation Misclassification rate 355
- Cross Validation Profit/Loss criterion 357–358
- customer attrition, predicting 6
- customer lifetime value 422
- customer profitability
 - about 415–417
 - acquisition cost 417–418
 - alternative scenarios of response and risk 422
 - cost of default 418–419

(continued)

- customer lifetime value 422
- extending results 423
- optimum cut-off point 421–422
- profit 419–421
- revenue 419
- Cutoff Cumulative property, Principal Components node 92–93
- Cutoff Value property
 - Replacement node 81
 - Transform Variables node 98
- cut-off values 258

D

- data
 - applying decision tree models to prospect 173
 - pre-processing 8–10
- data cleaning 9
- data matrix 427–428, 430–431
- Data Mining the Web* (Markov and Larose) 458
- data modification, nodes for
 - Drop 10, 79–80
 - Impute 10, 83, 153–154, 360, 386
 - Interactive Binning 83–90
 - Principal Components 90–95
 - Replacement 80–83
 - Transform Variables (*See* Transform Variables node)
- Data Options dialog box 58
- Data Partition node
 - about 27–28, 29, 249, 250
 - loss frequency as an ordinal target 269
 - Partitioning Method property 28, 386
 - property settings 197
 - Regression node 360, 372
 - variable selection 139, 145
 - variable transformation 153–154
- Data Set Allocations property, Data Partition node 28

- data sets
 - applying decision tree models to score 205–208
 - creating from text files 433–435
 - scoring using Neural Network models 263–266
 - scoring with models 277–279
- Data Source property, Input Data node 26–27
- data sources
 - changing measurement scale of variables in 164–165
 - creating 16–25, 37–40, 436
 - creating for text mining 436
 - creating for transaction data 37–40
- decision 171, 174
- decision tree models
 - about 170–172
 - accuracy/misclassification criterion 193–194
 - adjusting predicted possibilities for over-sampling 235–236
 - applying to prospect data 173
 - assessing using Average Square Error 194
 - average profit vs. total profit 192–193
 - binary split searches 176–177
 - calculating worth of trees 173–175
 - compared with logistic regression models 172
 - controlling growth of trees 185
 - developing interactively 215–233
 - developing regression tree model to predict risk 208–215
 - exercises 236–237
 - impurity reduction 182–183
 - measuring worth of splits 177–182
 - Pearson's Chi-square test 234
 - for predicting attrition 387–389
 - predicting response to direct marketing with 195–208
 - for predicting risk in auto insurance 396–399
 - pruning trees 185–192

(continued)

- p*-value adjustment options 183–185
 - regression tree 176
 - roles of training and validation data in 175
 - in SAS Enterprise Miner 176–195
 - selecting size of trees 194–195
 - Decision Tree node
 - See also* decision trees
 - about 117–118, 163
 - bins in 97
 - building decision tree models 187, 195
 - Interactive property 222, 225, 231–233
 - Leaf Role property 151
 - logistic regression 366, 367, 368, 377
 - in process flow 134
 - regression models 387–389, 392–401
 - Regression node 359
 - variable selection in 121
 - variable selection using 150–153
 - decision trees
 - developing interactively 215–233
 - growing 233
 - Decision Weights tab 22–23
 - Decisions property 187
 - Decisions tab 23
 - Default Filtering Method property, Filter node 29
 - Default Input Method property, Impute node 83
 - Default Limits Method property, Replacement node 81
 - default methods 98–100
 - degree of separation 178–179
 - depth adjustment 184
 - depth multiplier 184
 - Diagram Workspace 15, 16
 - dimension reduction 429–431
 - direct mail, predicting response to 2–3
 - direct marketing, predicting response to 195–208
 - DMDB procedure 78–79
 - Dmine Regression node 312–313, 314–315, 316
 - DMNeural node 309–312, 314–315, 316
 - documents, retrieving from World Wide Web 432–433
 - document-term matrix 427–428
 - Drop from Tables property, Drop node 80
 - Drop node 10, 79–80
- E**
- EHRadial value 307
 - Eigenvalue Source property 91
 - eigenvalues 64, 110–115
 - eigenvectors 110–115
 - Elliot function 243–244
 - EM (Expectation-Maximization) clustering 452–458, 460–461
 - Ensemble node 384, 402, 407–409, 410–411
 - Entropy 180–181
 - Entry Significance Level property, Regression node
 - Forward Selection method 340–342, 342–343
 - regression models 372, 386
 - Stepwise Selection method 345–347
 - EQRadial value 307
 - EQSlopes value 307
 - error function 248
 - EVRadial value 307
 - EWRadial value 307
 - exercises
 - decision tree models 236–237
 - models, combining 412–413
 - neural network models 318–319
 - predictive modeling 115–116
 - regression models 382
 - textual data, predictive modeling with 461
 - variable selection 166–167
 - Expectation-Maximization (EM) clustering 452–458, 460–461
 - expected losses 423
 - expected lossfrq 394
 - explanatory variables 170, 241
 - Exported Data property
 - Input Data node 102
 - Time Series node 43–44

F

false positive fraction 258
 File Import node 32–35
 Filter node 10, 28–32, 445
 Filter Viewer property 445
 fine tuning 27
 Forward Selection method
 about 340
 when target is binary 340–342
 when target is continuous 342–343
 frequency
 about 8
 adjusted 441
 FREQUENCY procedure 54
 frequency weighting 440, 441
 Frequency Weighting property 444

G

Gauss function 243
 Gini Cutoff property, Interactive Binning node
 84–85
 Gini Impurity Index 180
 gradient boosting 402–404
 Gradient Boosting node 402, 404–406, 410–411
 GraphExplore node 50, 51, 58–61
 groups
 See also leaf nodes
 combining 88–90
 splitting 85–88

H

Help Panel 15
 Hidden Layer Activation Function property 241,
 281, 283, 287–288, 305
 Hidden Layer Combination Function property
 241, 281, 283, 287–288, 305–306
 hidden layers 242–246
 Hide property
 Regression node 363

Transform Variables node 101, 156, 159,
 162
 transforming variables 162
 Hide Rejected Variables property, Variable
 Selection node 120, 122
 hierarchical clustering 451–459
 Huber-M Loss 411
 Hyperbolic Tangent function 243–244

I

Identity link 335
 Import File property, File Import node 33, 34
 Imported Data property, SAS Code node 102–
 103
 impurity reduction
 about 53
 as measure of goodness of splits 179–180
 when target is continuous 182–183
 Impute node 10, 83, 153–154, 360, 386
 Include Class Variables property, Variable
 Clustering node 139–140
 initial data exploration, nodes for
 about 50–51
 Cluster 50, 69–72, 451
 Graph Explore 50, 51, 58–61
 MultiPlot 50, 51, 56–58, 358
 Stat Explore 10, 50, 51–56, 79, 114, 358
 Variable Clustering 50, 61–69, 117–118,
 138–150, 163, 352
 Variable Selection 50, 72–79, 153–154,
 155–157, 163, 164, 359
 input 111, 170
 Input Data node
 about 10, 249, 250
 building decision tree models 195–196, 205
 Data Source property 26–27
 Exported Data property 102
 loss frequency as an ordinal target 267
 in process flow 91, 101
 regression models 385–386, 407
 scoring datasets 277
 transforming variables 153–154

Input Data Source node 333, 360, 372
 input layer 242
 Input Standardization property 305
 input variables, regression with large number of
 11
 inputs window 6
 Interactive Binning node 83–90
 Interactive Binning property, Interactive Binning
 node 85, 88–89
 Interactive property 216
 Interactive property, Decision Tree node 222,
 225, 231–233
 Interactive Selection property, Principal
 Components node 91, 93
 intermediate nodes 130
 Interval Criterion property 177–178, 182, 183
 interval inputs, transformations for 95–98
 Interval Inputs property
 Merge node 45–46, 47
 Regression node 364
 Transform Variables node 95, 96, 98, 155,
 159, 162
 interval variables 2
 Interval Variables property
 Filter node 30
 StatExplore node 52, 55, 114
 inverse link function 322

K

KeepHierarchies property, Variable Clustering
 node 65

L

Larose, D.T.
 Data Mining the Web 458
 latent semantic indexing 429–431
 leaf nodes 170, 233
 See also terminal nodes
 Leaf Role property
 Decision Tree node 151

 Regression node 377
 Leaf Rule property 389–390
 Leaf Size property 185, 215, 367
 Leaf Variable property 389–390
 Least Absolute Deviation Loss 411
 Least Squares Loss 411
 lift 174–175
 lift charts 393–394
 Linear Combination function 305–306
 Linear Regression 333
 Linear value 306
 link function 321
 Link Function property, Regression node 324,
 333–335, 348
 Logistic function 244
 logistic regression
 about 333
 for predicting attrition 386–387
 for predicting mail campaign response 359–
 371
 with proportional odds 394–396
 logistic regression models, vs. decision tree
 models 172
 Logit link 322, 334
 Logit Loss 412
 logworth 178–179
 loss frequency 208, 240, 266–279

M

marginal profit 420
 marginal revenue 420
 Markov, Z.
 Data Mining the Web 458
 maximal tree 175, 191–192
 Maximum Clusters property, Variable Clustering
 node 62
 Maximum Depth property 185, 215, 367
 Maximum Eigenvalue property, Variable
 Clustering node 62, 64
 Maximum method 408
 Maximum Number of Steps property 345–347

- maximum posterior probability/accuracy,
 - classifying nodes by 193
 - measurement scale 1–2, 107–109
 - measurement scale, of variables 164–165
 - Menu Bar 15
 - Merge node 45–47, 48, 159–162
 - Merging property, Transform Variables node 161
 - Metadata Advisor Options window 19
 - Method property 84, 187, 367
 - methods
 - Average 408
 - Backward Elimination 335–340
 - Boolean retrieval 427
 - Chi-Square selection 73
 - default 98–100
 - frequency weighting 441
 - Maximum 408
 - R-Square selection 72–73
 - term weighting 441–445
 - Minimum Chi-Square property, Variable Selection node 73
 - Minimum property, Cluster node 70
 - Minimum R-Square property, Variable Selection node 73, 74, 120–121, 157
 - misclassification criterion 174, 193–194
 - MLP (Multilayer Perception) neural network 279–281, 287–288
 - Model Comparison node
 - assessing predictive performance of
 - estimated models 254–258
 - building decision tree models 204–205, 211, 233
 - comparing alternative built-in architectures 286
 - in process flow 139, 149
 - regression models 391
 - Regression node 360, 368
 - variable selection 151
 - Model Selection Criterion property 248, 249, 250, 273, 389
 - Model Selection property 386
 - modeling data, sources of 8
 - modeling strategies, alternative 10–11
 - models
 - See also* neural network models
 - for binary targets 384–392
 - combining 412–413
 - comparing and combining 383–413
 - for ordinal targets 392–401
 - Multilayer Perception (MLP) neural network 279–281, 287–288
 - Multiple Method property, Transform Variable node 162
 - MultiPlot node 50, 51, 56–58, 358
- N**
- Network property, Neural Network node 251, 269, 283
 - neural network models
 - about 240–241
 - alternative specifications of 279–286
 - AutoNeural node 314–315
 - comparing alternative built-in architectures
 - of Neural Network node 286–309
 - Dmine Regression node 312–313, 314–315
 - DMNeural node 309–312, 314–315
 - estimating weights in 247–249
 - exercises 318–319
 - general example of 241–247
 - nodes for 240–241
 - for predicting attrition 389–392
 - predicting loss frequency in auto insurance 266–279
 - for predicting risk in auto insurance 400–401
 - scoring data sets using 263–266
 - target variables for 240
 - Neural Network node
 - about 240–241, 316
 - Architecture property 281, 283, 284, 293, 295–297, 298–300, 305
 - loss frequency as an ordinal target 268–269

(continued)

- Model Selection Criterion property 273
- Multilayer Perceptron (MLP) neural networks 278–281
- Normalized Radial Basis Function with Equal Heights and Unequal Widths (NRBFEH) 295–297
- Normalized Radial Basis Function with Equal Volumes (NRBFEV) 301–302
- Normalized Radial Basis Function with Equal Widths and Heights (NRBFEQ) 292–294
- Ordinary Radial Basis Function with Equal Heights and Unequal Widths (ORBFEQ) 291–292
- Radial Basis Function neural networks in 282–286
 - regression models 389–390, 392–401
 - score ranks in Results window 275
 - scoring datasets 277
 - selecting optimal weights 261–263
 - setting properties of 250–254
 - target layer combination and activation functions 270–272
- neural networks
 - about 316
 - alternative specifications of 279–286
 - comparing alternative built-in architectures in 286–307
- node definitions 175
- Node (Tool) group tabs 15
- Node ID property, Transform Variables node 159, 161
- nodes
 - See also* Data Partition node
 - See also* Decision Tree node
 - See also* Input Data node
 - See also* Model Comparison node
 - See also* Neural Network node
 - See also* Regression node
 - See also* SAS Code node
 - See also* Transform Variables node
 - See also* Variable Clustering node
 - See also* Variable Selection node
- Append 48–50, 116
- AutoNeural node 307–309, 314–315, 316
- child 170
- classifying by maximum posterior probability/accuracy 193
- Cluster 50, 69–72, 451
- for data modification 79–101
- Dmine Regression 312–313, 314–315, 316
- DMNeural 309–312, 314–315, 316
- Drop 10, 79–80
- Ensemble 384, 402, 407–409, 410–411
- File Import 32–35
- Filter 10, 28–32, 445
- Gradient Boosting 402, 404–406, 410–411
- GraphExplore 50, 51, 58–61
- Impute 10, 83, 153–154, 360, 386
- for initial data exploration 50–79
- Input Data Source 333, 360, 372
- Interactive Binning 83–90
- intermediate 130
- leaf 170, 233 (*See also* terminal nodes)
- Merge 45–47, 48, 159–162
- MultiPlot 50, 51, 56–58, 358
- for neural network models 240–241
- parent 170
- Principal Components 90–95
- Replacement 80–83
- responder 192
- Root 130, 170, 225–231
- sample 26–50
- Score 205–207, 265, 266, 277
- splitting using binary split search 176–177
- StatExplore 10, 50, 51–56, 79, 114, 358
- Stochastic Boosting 384
- terminal 130, 170
- Text Cluster 450, 451–458, 458–461
- Text Filtering 432, 440–450, 442
- Text Import 431, 436
- Text Parsing 431, 436–440, 442, 445–450
- Text Topic 445–450
- Time Series 35–44
- Transformation 376
- utility 101–107

- nominal categorical (unordered polychotomous) target, predicting 7
 - Nominal Criterion property 177, 180–181
 - nominal (unordered) target, regression models with 329–332
 - nominal-categorical inputs, continuous target with 124–129
 - nominal-scaled categorical inputs, binary target with 135–138
 - non-responders 234
 - NRBFEH (Normalized Radial Basis Function with Equal Heights and Unequal Widths) 295–297
 - NRBFEQ (Normalized Radial Basis Function with Equal Widths and Heights) 292–294
 - NRBFEV (Normalized Radial Basis Function with Equal Volumes) 300–302
 - NRBFEW (Normalized Radial Basis Function with Equal Widths and Unequal Heights) 297–300
 - NRBFUN (Normalized Radial Basis Function with Unequal Widths and Heights) 302–304
 - Number of Bins property
 - about 131
 - StatExplore node 52
 - Variable Selection node 73
 - Number of Hidden Units property 279–281, 286, 389, 400–401
 - number of levels, of variables 107–109
 - numeric interval-scaled inputs
 - binary target with 129–135
 - continuous target with 119–124
- O**
- observation weights 8
 - observed proportions 170
 - Offset Value property, Transform Variables node 96
 - opening SAS Enterprise Miner 12.1 14
 - operational lag 6
 - optimal binning 45, 96–97
 - optimal tree 175
 - Optimization property 250
 - optimum cut-off point 421–422
 - ORBFEQ (Ordinary Radial Basis Function with Equal Heights and Widths) 288–290
 - ORBFUN (Ordinary Radial Basis Function with Equal heights and Unequal Widths) 291–292
 - ORBFUN (Ordinary Radial with Unequal Widths) 283
 - ordered polychotomous targets
 - See ordinal targets
 - Ordinal Criterion property 177, 180–181
 - ordinal targets
 - loss frequency as 267–279
 - models for 392–401
 - regression models with 324–329
 - original segment 170
 - output data sets
 - created by Time Series node 43–44
 - developing predictive equations created by Text Topic node 449–450
 - output layer 246–247
 - overriding default methods 99–100
 - over-sampling, adjusting predicted probabilities for 235–236
- P**
- p weights 284
 - parent nodes 170
 - Partitioning Method property, Data Partition node 28, 386
 - Pearson Correlations property, StatExplore node 55
 - Pearson's Chi-square test 234
 - percentage of ranked data (n%) 175
 - performance window 6, 7
 - posterior probability
 - about 170
 - for leaf nodes from training data 189

(continued)

- of non-response 203
 - of response 203
 - Posterior Probability property 408
 - Predicted Values property 408
 - predicting
 - attrition 384–392
 - customer attrition 6
 - loss frequency in auto insurance with Neural Network model 266–279
 - nominal categorical (unordered polychotomous) target 7
 - rate sensitivity of bank deposit products 4–5
 - response (*See* neural network models)
 - response to direct mail 2–3
 - response to direct marketing 195–208
 - risk (*See* neural network models)
 - risk in auto insurance industry 3–4
 - risk of accident risk 392–401
 - risk with regression tree models 208–215
 - predictive equations, developing using output data set created by Text Topic node 449–450
 - predictive modeling
 - See also* textual data, predictive modeling with
 - about 14
 - boosting 402–411
 - combining 402–411
 - creating new projects in SAS Enterprise Miner 12.1 14–15
 - creating process flow diagrams 25–26
 - creating SAS data sources 16–25
 - eigenvalues 64, 110–115
 - eigenvectors 110–115
 - exercises 115–116
 - measurement scale 107–109
 - nodes for data modification 79–101
 - nodes for initial data exploration 50–79
 - number of levels of variable 107–109
 - opening SAS Enterprise Miner 12.1 14
 - principal components 110–115
 - sample nodes 26–50
 - SAS Enterprise Miner window 15–16
 - type of variable 107–109
 - utility nodes 101–107
 - Preliminary Maximum property, Cluster node 70
 - pre-processing data 8–10
 - principal components 110–115
 - Principal Components node 90–95
 - Prior Probabilities tab 22
 - probabilities, adjusted 236
 - Probit link 334
 - process flow diagrams 25–26, 40
 - profit 419–421
 - See also* customer profitability
 - See also* validation profit
 - average vs. total 192–193
 - marginal 420
 - Profit/Loss criterion 357
 - Project Panel 15
 - projects, creating in SAS Enterprise Miner 12.1 14–15
 - promotion window 4
 - properties
 - See also specific properties*
 - of Neural Network node 250–254
 - of Regression node 333–358
 - Properties Panel 15
 - Proportional Odds model 394–396
 - pruning trees 175, 185
 - p-value 52
 - P*-value adjustment options
 - Bonferroni Adjustment property 183–184
 - depth adjustment 184
 - Leaf Size property 185
 - Split Adjustment property 184
 - Threshold Significance Level property 184
 - Time of Kass Adjustment property 184
- Q**
- quantifying textual data 426–428, 431–432
 - quantile 96

R

- rate sensitivity, predicting of bank deposit products 4–5
- RBF (Radial Basis Function) neural network 281–286
- Receiver Operating Characteristic (ROC) charts 258–261
- recursive partitioning 130, 170
- regression
 - for continuous targets 371–379
 - with large number of input variables 11
- regression models
 - about 321
 - with binary targets 321–324
 - business applications 358–379
 - exercises 382
 - Regression node properties 333–358
 - types of models developed using 321–333
- Regression node
 - See also* regression models
 - about 10, 11
 - Architecture property 389
 - Chi-Square criterion 138
 - Data Partition node 360, 372
 - Decision Tree node 359
 - Entry Significance Level property 340–342, 342–343, 345–347, 372, 386
 - Hide property 363
 - Interval Inputs property 364
 - Leaf Role property 377
 - Link Function property 324, 333–335, 348
 - predictive modeling 449–450
 - in process flow 76, 90, 95, 96, 101, 123–124, 129, 133, 139, 143–144
 - properties of 333–358
 - regression models 386, 387–389, 392–401
 - Regression Type property 324, 333, 348
 - Reject property 363
 - R-Square criterion 130, 136–137
 - Selection Model property 144, 335–347, 340, 342–343, 348, 367, 394, 449–450
 - testing significance of dummy variables 98
 - testing variables and transformations 45, 47, 48
 - Transform Variables node 359
 - transforming variables 157, 159, 161–162
 - Variable Clustering node 352
 - variable selection 145, 146, 148–149, 150, 151, 152
 - variable selection in 121
 - Variable Selection property 377, 389–390
 - Variables property 95, 123–124, 133–134
- regression tree 176, 208–215
- Regression Type property, Regression node 324, 333, 348
- Reject property
 - Regression node 363
 - Transform Variables node 101, 156, 159, 162
 - transforming variables 162
- Replacement Editor property, Replacement node 81–82
- Replacement node 80–83
- research strategy
 - about 1
 - alternative modeling strategies 10–11
 - defining targets 2–8
 - measurement scales for variables 1–2
 - pre-processing data 8–10
- residuals, calculating 403–404
- responder node 192
- responders 234
- response
 - alternative scenarios of 422
 - predicting (*See* neural network models)
 - predicting to direct mail 2–3
- revenue 419
- risk
 - See also* neural network models
 - alternative scenarios of 422
 - classifying for rate setting 279
 - predicting in auto insurance industry 3–4
 - predicting with regression tree models 208–215
- risk rate 415–416

ROC (Receiver Operating Characteristic) charts
258–261
Role property 263
Root node 130, 170, 225–231
R-Square criterion 130, 136–137
R-Square selection method 72–73

S

sample nodes

Append 48–50, 116
Data Partition 27–28, 29, 139, 145, 153–
154, 198, 249, 250, 269, 360, 372
File Import 32–35
Filter 10, 28–32, 445
Input Data 10, 26–27, 91, 101, 102, 153–
154, 195–196, 205, 249, 250, 267, 277,
385–386, 407
Merge 45–47, 48, 159–162
Time Series 35–44

samples, compared with targets 8

SAS Code node

about 10, 101–107
building decision tree models 207–208
logistic regression 374
predictive modeling 438, 444
score ranks in Results window 275

SAS Enterprise Miner

creating projects in 14–15
data cleaning after launching 9
data cleaning before launching 9
developing decision trees in 176–195
opening 14
window 15–16

SAS Enterprise Miner: Reference Help 284, 437

SBC (Schwarz Bayesian Criterion) 352–353

Score node 205–207, 265, 266, 277

scoring

data sets using Neural Network models
263–266
datasets with models 277–279
showing ranks in Results window 273–276

segments 170

See also leaf nodes

Select an Analysis property, Time Series node
41

Selection Criterion property, Regression node
about 348–350

Akaike Information Criteria (AIC) 350–352

Backward Elimination method 335

cross validation error 355

cross validation misclassification rate 355

Cross Validation Profit/Loss Criterion 357–
358

Forward Selection method 341–342, 342–
343

logistic regression 394

predictive modeling with textual data 449–
450

Profit/Loss Criterion 357

regression models 365, 367, 376, 386

Schwarz Bayesian Criterion (SBC) 352–
353

validation error 353–354

validation misclassification 354

Validation Profit/Loss Criterion 355–356

variable selection 144

Selection Default property 343

Selection Model property, Regression node 144,
335–347, 340, 342–343, 348, 367,
394, 449–450

Selection Options property 336, 344–345

sensitivity

See true positive fraction

separation, degree of 178–179

Significance Level property 184, 215, 348, 367

simple transformation 96

Sine function 243

Singular Value Decomposition (SVD) 429, 431

sources, of modeling data 8

Spearman Correlations property, StatExplore
node 55

specificity

See true positive fraction

Split Adjustment property 184, 215

split point, changing of nominal variables 222–225

Split Size property 185

splits, measuring worth of 177–181, 181–182

splitting

- groups 85–88
- nodes using binary split search 176–177
- process of 62

Splitting Rule Criterion property 233, 387–389

Splitting Rule Interval Criterion property 209

splitting value 176

StatExplore node 10, 50, 51–56, 79, 114, 358

Status Bar 15, 16

Stay Significance Level property 335, 336, 337, 343, 345–347, 372

stepwise selection method

- about 343
- when target is binary 344–345
- when target is continuous 345–347

Stochastic Boosting node 384

stochastic gradient boosting 404

Stop R-Square property, Variable Selection node 73, 74, 121, 157

Sub Tree Method property 198

sub-segments 170

Subtree Assessment Measure property 233

Subtree Method property 377, 387–389, 396–399

SVD (Singular Value Decomposition) 429, 431

SVD Revolution property 451

synthetic variables 246–247

T

Tables to Filter property, Filter node 29

Target Activation Function property 393

target layer 246–247, 270–272

Target Layer Activation Function property 241, 281, 283, 305, 307

Target Layer Combination Function property 241, 270, 281, 283, 305, 307

Target Layer Error Function property 272–273, 281, 283

Target Model property, Variable Selection node 72, 77–78, 78–79, 121–122, 136, 137

target variables, for neural network models 240

targets

- See also* binary targets
- See also* continuous targets
- See also* ordinal targets
- compared with samples 8
- defining 2–8
- maximizing relationship to 96–98
- transformations of 98

Targets tab 21

Term Weight property 444

term weighting 440–441, 441–445

term-document matrix 426–427

terminal nodes 130, 170

test data

- roles of in development of decision trees 175
- testing model performance with 204–205

Test property, Data Partition node 28, 360

Text Cluster node 450, 451–458, 458–461

text files, creating SAS data sets from 433–435

Text Filter node 432, 440–450

Text Filtering node 442

Text Import node 431, 436

text mining, creating data sources for 436

Text Parsing node 431, 436–440, 442, 445–450

Text Topic node 445–450

textual data, predictive modeling with

- about 425–426
- creating data sources for text mining 436
- creating SAS data sets from text files 433–435
- dimension reduction 429–431
- exercises 461
- latent semantic indexing 429–431
- quantifying textual data 426–428
- retrieving documents from World Wide Web 432–433

Text Cluster node 450, 451–458, 458–461

Text Filter node 432, 440–450

(continued)

- Text Import node 431, 436
 - Text Parsing node 431, 436–440, 442, 445–450
 - Text Topic node 445–450
 - Threshold Significance Level property 184
 - Time of Kass Adjustment property 184
 - Time Series node 35–44
 - %TMFILTER macro 432, 435
 - Toolbar Shortcut Buttons 15, 16
 - tools
 - See* nodes
 - Tools Bar 15
 - total profit, vs. average profit for comparing tree size 192–193
 - training, of trees 172
 - training data
 - developing trees using 188–189
 - roles of in development of decision trees 175
 - training data set 233
 - Training property, Data Partition node 28, 360
 - transaction data
 - converting to time series 35–37
 - creating data sources for 37–40
 - transform variables, saving code generated by 163
 - Transform Variables node
 - See also* variable selection
 - about 95–101, 117–118, 163, 164
 - Class Inputs property 376
 - Hide property 101, 156, 159, 162
 - Interval Inputs property 95, 96, 98, 155, 159, 162
 - Merging property 161
 - Multiple Method property 162
 - Node ID property 159, 161
 - Offset Value property 96
 - in process flow 101, 102, 116
 - Regression node 359
 - Reject property 101, 156, 159, 162
 - testing variables and transformations 45, 46–47, 48
 - transforming variables 155–157, 158, 159, 160–162
 - transforming variables with 153–155
 - Variables property 99
 - Transformation node 376
 - transformations
 - after variable selection 157–159
 - binning 96
 - of class inputs 98
 - for interval inputs 95–98
 - multiple using Multiple Method property 162
 - passing more than one for each interval input 159–162
 - passing two types using Merge node 159–162
 - simple 96
 - of targets 98
 - before variable selection 155–157
 - of variables 153–163
 - TRANSPOSE procedure 439
 - Treat Missing as Level property
 - Interactive Binning node 84
 - Regression node 376
 - trees
 - about 170
 - assessing using Average Square Error 194
 - true positive fraction 258
- U**
- unadjusted probabilities, expected profits using 236
 - ungrouped variables, compared with categorical variables 165
 - unordered (nominal) target, regression models with 329–332
 - Use AOV16 Variables property
 - Dmine Regression node 313
 - Variable Selection node 73, 74, 78, 120, 122, 157
 - Use Group Variables property, Variable Selection node 74–75, 122, 124–125

Use Selection Defaults property 335, 336, 340, 342–343, 386
 user-defined networks 307
 user-specified architectures 305–307
 utility nodes 101–107

V

validation accuracy 193
 validation data
 pruning trees using 185–187
 roles of in development of decision trees 175
 validation error 353–354
 Validation Error criterion 386
 validation misclassification 354
 validation profit 175, 190–192
 Validation Profit/Loss criterion 355–356
 Validation property, Data Partition node 28, 360
 variable clustering, using example data set 65–69
 Variable Clustering node
 about 50, 61–69, 117–118, 163
 Include Class Variables property 139–140
 Maximum Clusters property 62
 Maximum Eigenvalue property 62, 64
 Regression node 352
 Variable Selection property 139
 variable selection using 138–150
 variable selection
 See also Transform Variables node
 about 117–119
 binary target with nominal-scaled categorical inputs 135–138
 binary target with numeric interval-scaled inputs 129–135
 continuous target with nominal-categorical inputs 124–129
 continuous target with numeric interval-scaled inputs 119–124
 exercises 166–167
 transformation after 157–159
 transformation before 155–157

 using Decision Tree node 150–153
 using Variable Clustering node 138–150
 Variable Selection node
 about 50, 72–79, 163, 164
 Hide Rejected Variables property 120, 122
 Minimum R-Square property 73, 74, 120–121, 157
 regression models 359
 Stop R-Square property 73, 74, 121, 157
 transforming variables 153–154, 155–157
 Variable Selection property
 Regression node 377, 389–390
 Variable Clustering node 139
 variables
 assigning to clusters 64–65
 categorical 1–2, 165
 changing measurement scale of in data sources 164–165
 explanatory 241
 interval 2
 measurement scale of 1–2, 107–109
 number of levels of 107–109
 selecting for clusters 140–148
 synthetic 246–247
 transformation of 153–163
 types of 107–109
 Variables property
 about 164
 Drop node 80
 File Import node 34
 Impute node 83
 Regression node 95, 123–124, 133–134
 Transform Variables node 99
 viewing properties 25
 variance
 of inputs 111
 proportion explained by cluster component 65
 proportion of explained by principal components 112–113
 Variation Proportion property, Variable Clustering node 62
 Voting Posterior Probabilities property 408–409

Voting...Average method 408–409
Voting..Proportion method 409

W

weights
 estimating in neural network models 247–
 249
 selecting for Neural Network node 261–263
windows
 Metadata Advisor Options 19
 SAS Enterprise Miner 15–16
World Wide Web, retrieving documents from
 432–433

X

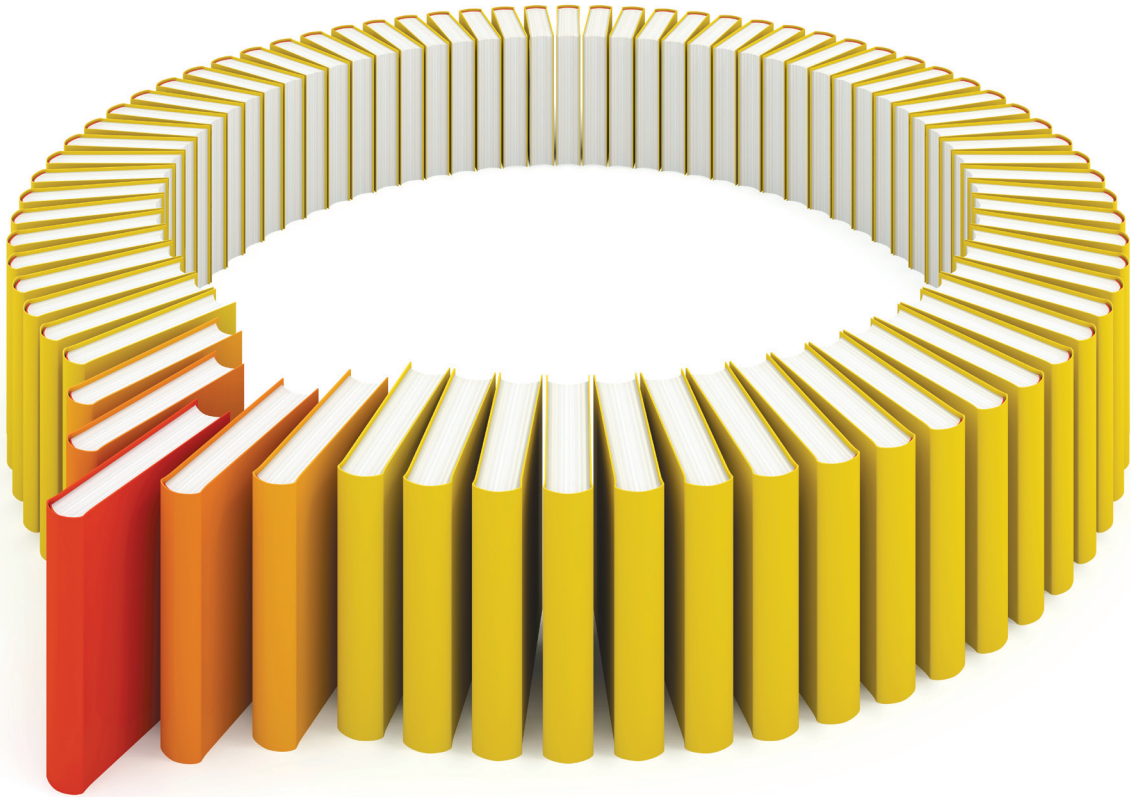
XRadial value 307

About The Author



Kattamuri S. Sarma, PhD, is an economist and statistician with 30 years of experience in American business, including stints with IBM and AT&T. He is the founder and president of Ecostat Research Corp., a consulting firm specializing in predictive modeling and forecasting. Over the years, Dr. Sarma has developed predictive models for the banking, insurance, telecommunication, and technology industries. He has been a SAS user since 1992, and he has extensive experience with multivariate statistical methods, econometrics, decision trees, and data mining with neural networks. The author of numerous professional papers and publications, Dr. Sarma is a SAS Certified Professional and a SAS Alliance Partner. He received his bachelor's degree in mathematics and his master's degree in economic statistics from universities in India. Dr. Sarma received his PhD in economics from the University of Pennsylvania, where he worked under the supervision of Nobel Laureate Lawrence R. Klein.

Learn more about this author by visiting his author page at <http://support.sas.com/publishing/authors/sarma.html>. There you can download free book excerpts, access example code and data, read the latest reviews, get updates, and more.



Gain Greater Insight into Your SAS[®] Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

 support.sas.com/bookstore
for additional books and resources.


THE POWER TO KNOW[®]