Appendix ACase Studies

A.1 Banking Segmentation Case Study

Case Study Description

A consumer bank sought to segment its customers based on historic usage patterns. Segmentation was to be used for improving contact strategies in the Marketing Department.

A sample of 100,000 active consumer customers was selected. An *active consumer customer* was defined as an individual or household with at least one checking account and at least one transaction on the account during a three-month study period. All transactions during the three-month study period were recorded and classified into one of four activity categories:

- traditional banking methods (TBM)
- automatic teller machine (ATM)
- point of sale (POS)
- customer service (CSC)

A three-month activity profile for each customer was developed by combining historic activity averages with observed activity during the study period. Historically, for one CSC transaction, an average customer would conduct two POS transactions, three ATM transactions, and 10 TBM transactions. Each customer was assigned this initial profile at the beginning of the study period. The initial profile was updated by adding the total number of transactions in each activity category over the entire three-month study period.

The **PROFILE** data set contains all 100,000 three-month activity profiles. This case study describes the creation of customer activity segments based on the **PROFILE** data set.

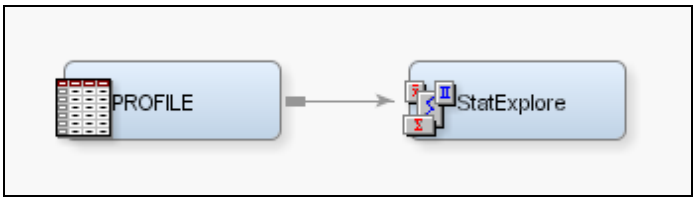
The diagram containing this analysis is stored as an XML file on the course data disk. You can open this file by right-clicking **Diagrams** and selecting **Import Diagram from XML** in SAS Enterprise Miner. All nodes in the opened file, except the data node, contain the property settings outlined in this case study. If you want to run the diagram, you need to re-create the case study data set using the metadata settings indicated below.

Case Study Data

Name	Model Role	Measurement Level	Description
ID	ID	Nominal	Customer ID
CNT_TBM	Input	Interval	Traditional bank method transaction count
CNT_ATM	Input	Interval	ATM transaction count
CNT_POS	Input	Interval	Point-of-sale transaction count
CNT_CSC	Input	Interval	Customer service transaction count
CNT_TOT	Input	Interval	Total transaction count

Accessing and Assaying the Data

A SAS Enterprise Miner data source was defined using the metadata settings indicated above. The StatExplore node was used to provide preliminary statistics on the input variables.



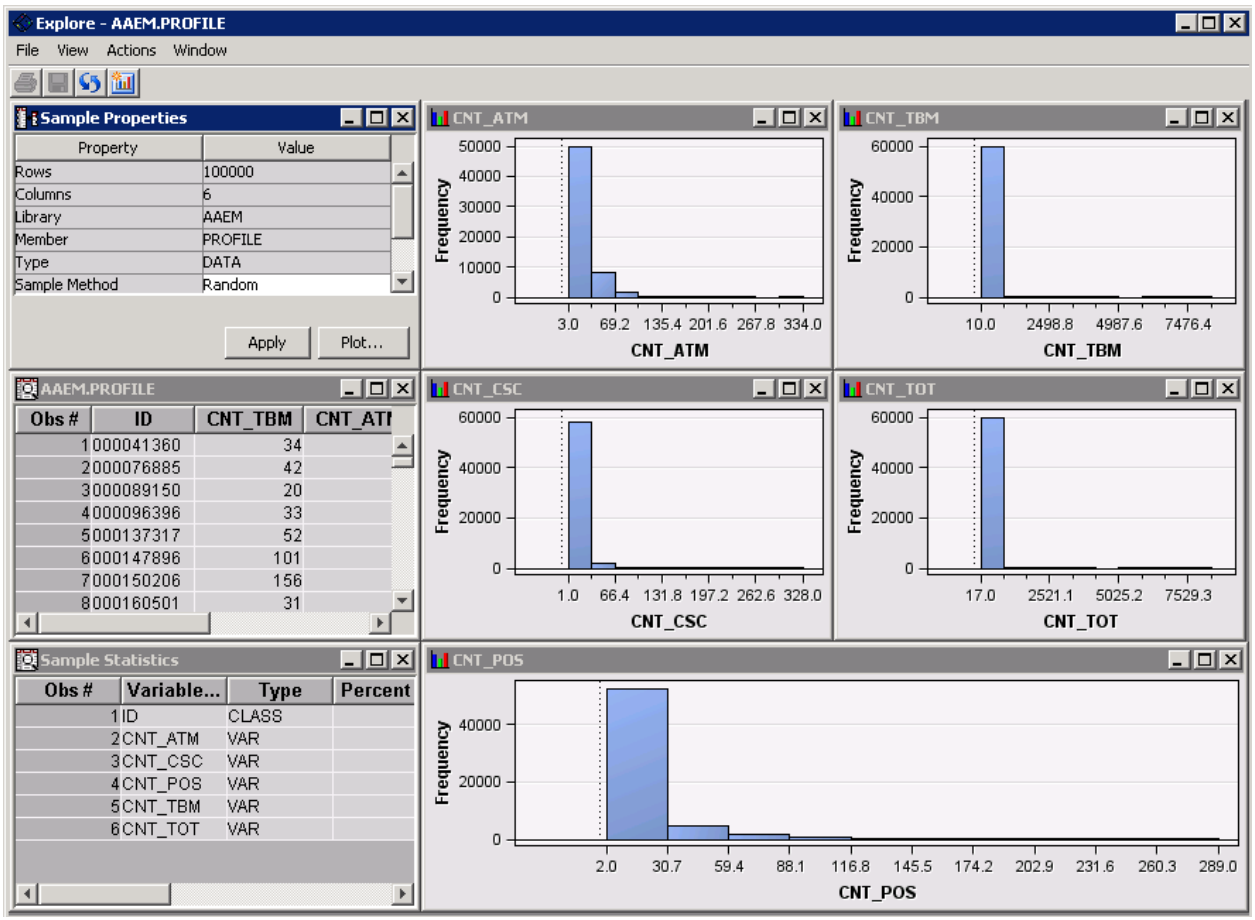
The Interval Variable Summary from the StatExplore node showed no missing values but did show a surprisingly large range on the transaction counts.

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

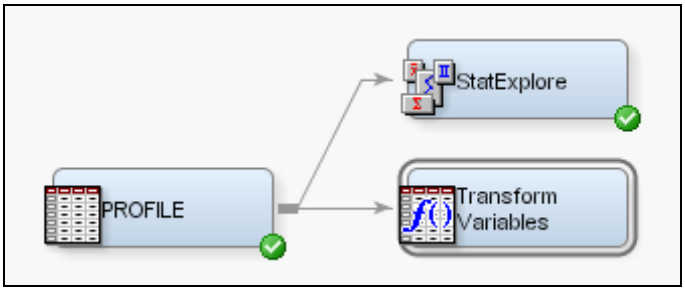
Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness
CNT_ATM	INPUT	19.49971	20.8561	100000	0	3	13	628	2.357293
CNT_CSC	INPUT	6.68411	12.12856	100000	0	1	2	607	6.236494
CNT_POS	INPUT	11.9233	20.73384	100000	0	2	2	345	3.343805
CNT_TBM	INPUT	68.13696	101.1542	100000	0	10	52	14934	53.05219
CNT_TOT	INPUT	106.2441	113.3704	100000	0	17	89	15225	39.2061

A plot of the input distributions showed highly skewed distributions for all inputs.



It would be difficult to develop meaningful segments from such highly skewed inputs. Instead of focusing on the transaction counts, it was decided to develop segments based on the relative proportions of transactions across the four categories. This required a transformation of the raw data.

A Transform Variables node was connected to the **PROFILE** node.

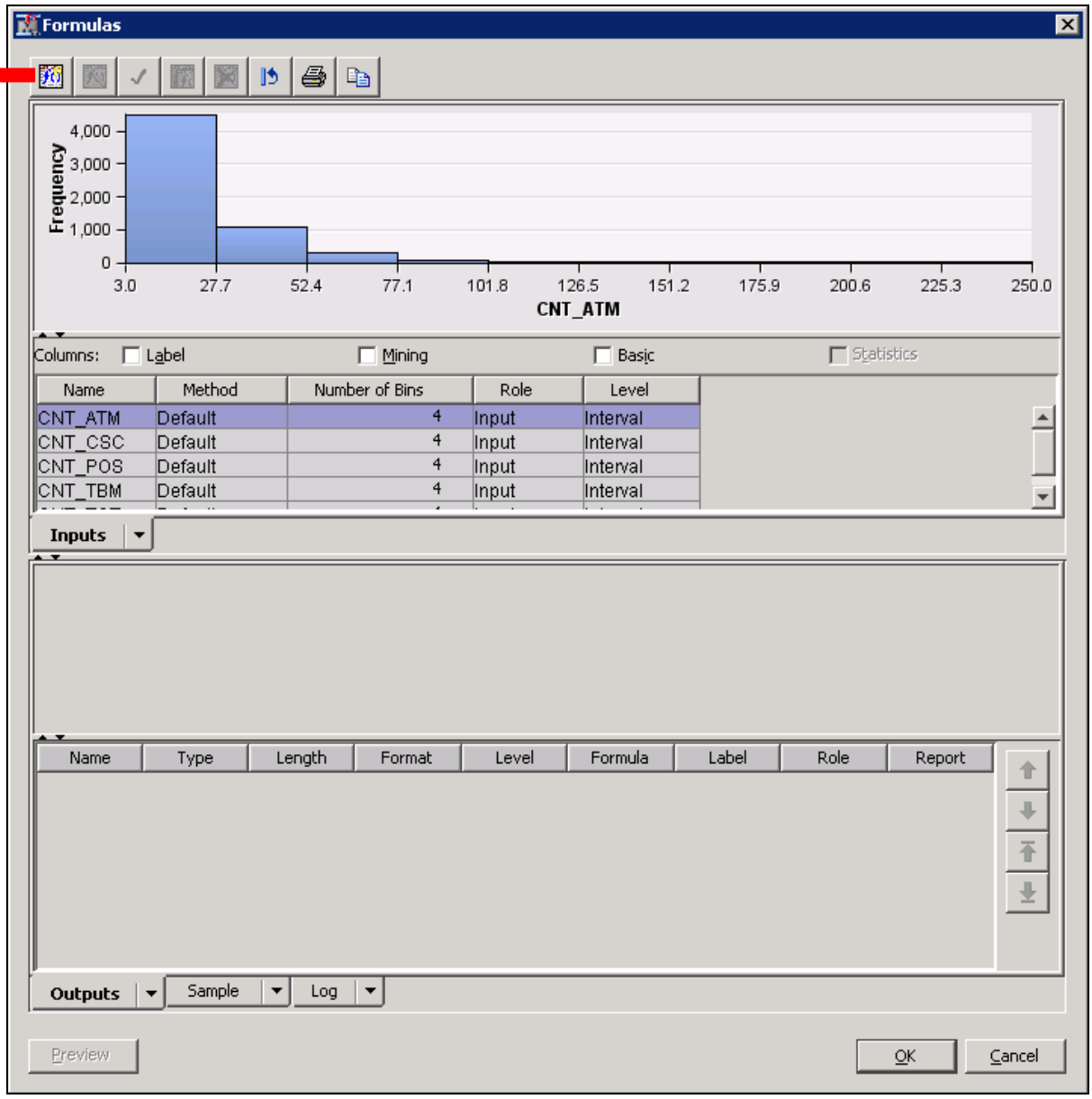


The Transform Variables node was used to create *category logit scores* for each transaction category.

$$\textit{category logit score} = \log(\textit{transaction count}_{\text{in category}} / \textit{transaction count}_{\text{out of category}})$$

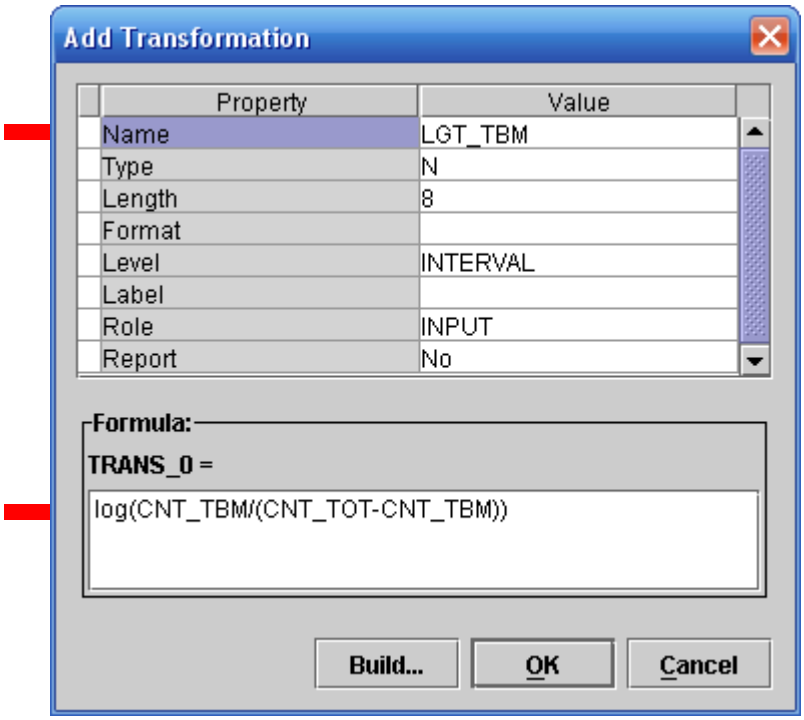
The transformations were created using these steps:

1. Select **Formulas** in the Transform Variable node's Properties panel. The Formulas window appears.



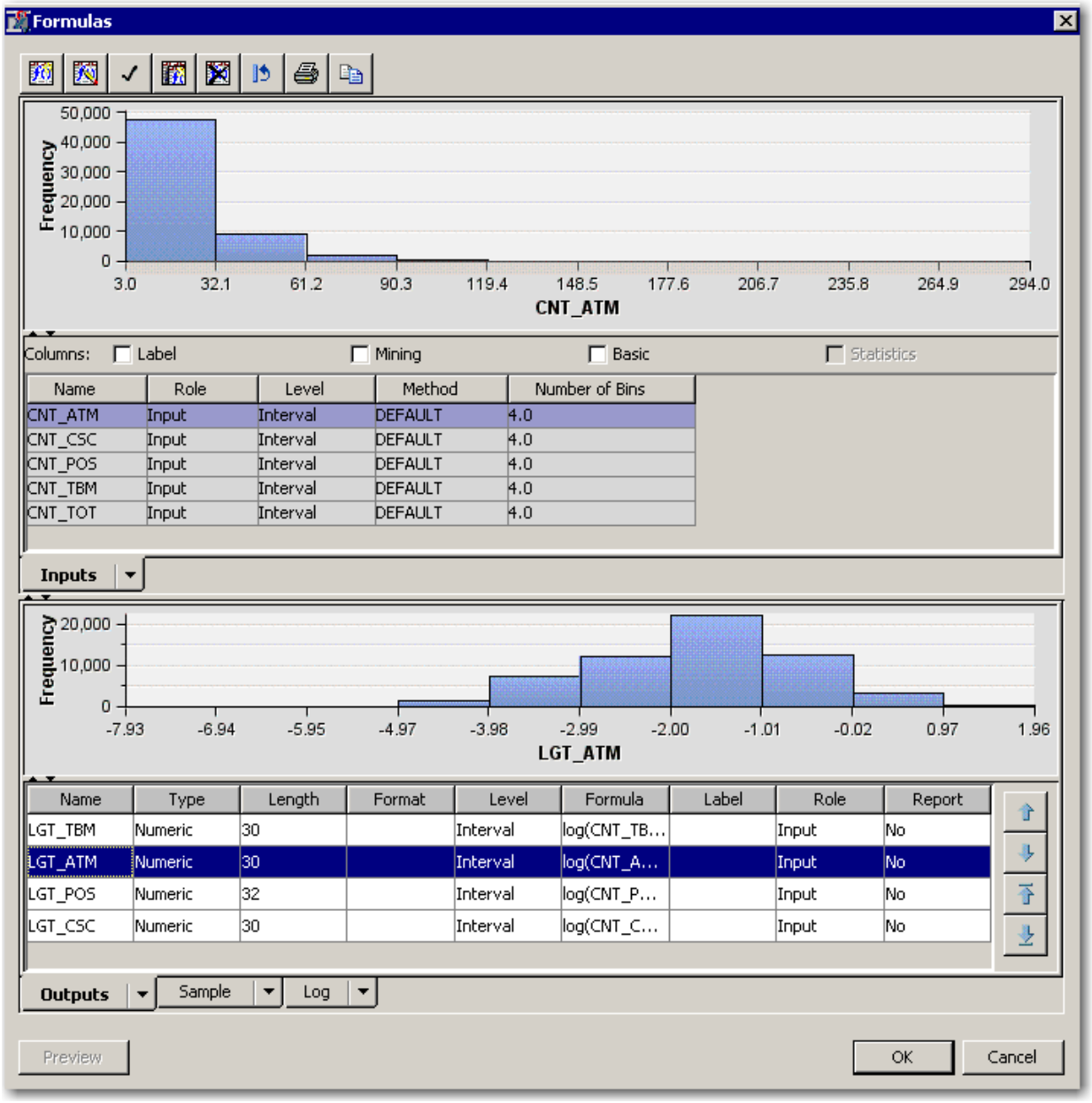
2. Select the Create icon as indicated above.

The Add Transformation dialog box appears.



3. For each transaction category, type the name and formula as indicated.
4. Select **OK** to add the transformation. The Add Transformation dialog box closes and you return to the Formula Builder window.

5. Select **Preview** to see the distribution of the newly created input.



6. Repeat Steps 1-5 for the other three transaction categories.

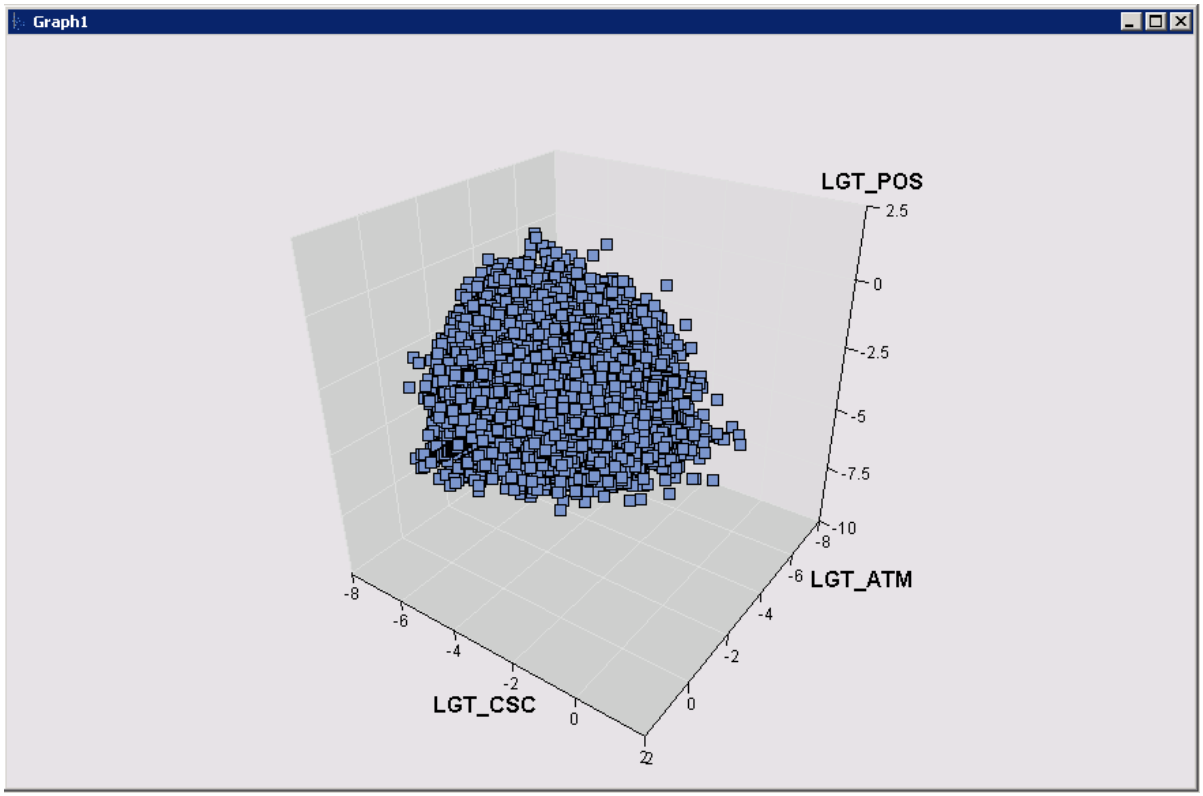
7. Select **OK** to close the Formula Builder window.

8. Run the Transform Variables node.

Segmentation was to be based on the newly created category logit scores. Before proceeding, it was deemed reasonable to examine the joint distribution of the cases using these derived inputs. A scatter plot using any three of the four derived inputs would represent the joint distribution without significant loss of information.

A three-dimensional scatter plot was produced using the following steps:

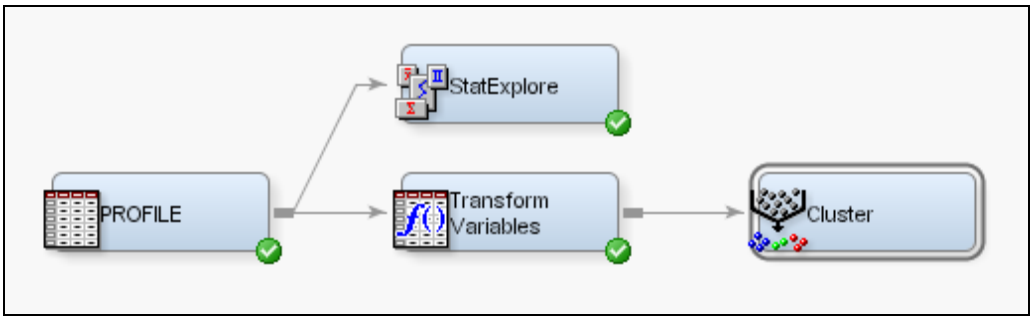
1. Select **Exported Data** from the Properties panel of the Transform Variables node. The Exported Data window appears.
2. Select the **TRAIN** data and select **Explore**. The Explore window appears.
3. Select **Actions** ⇨ **Plot…** or click  (the Plot Wizard icon). The Plot Wizard appears.
4. Select a three-dimensional scatter plot.
5. Select **Role** ⇨ **X**, **Y**, and **Z** for **LGT_ATM**, **LGT_CSC**, and **LGT_POS**, respectively.
6. Select **Finish** to generate the scatter plot.



The scatter plot showed a single clump of cases, making this analysis a segmentation (rather than a clustering) of the customers. There were a few outlying cases with apparently low proportions on the three plotted inputs. Given that the proportions in the four original categories must sum to 1, it followed that these outlying cases must have a high proportion of transactions in the non-plotted category, TBM.

Creating Segments

Transactions segments were created using the Cluster node.

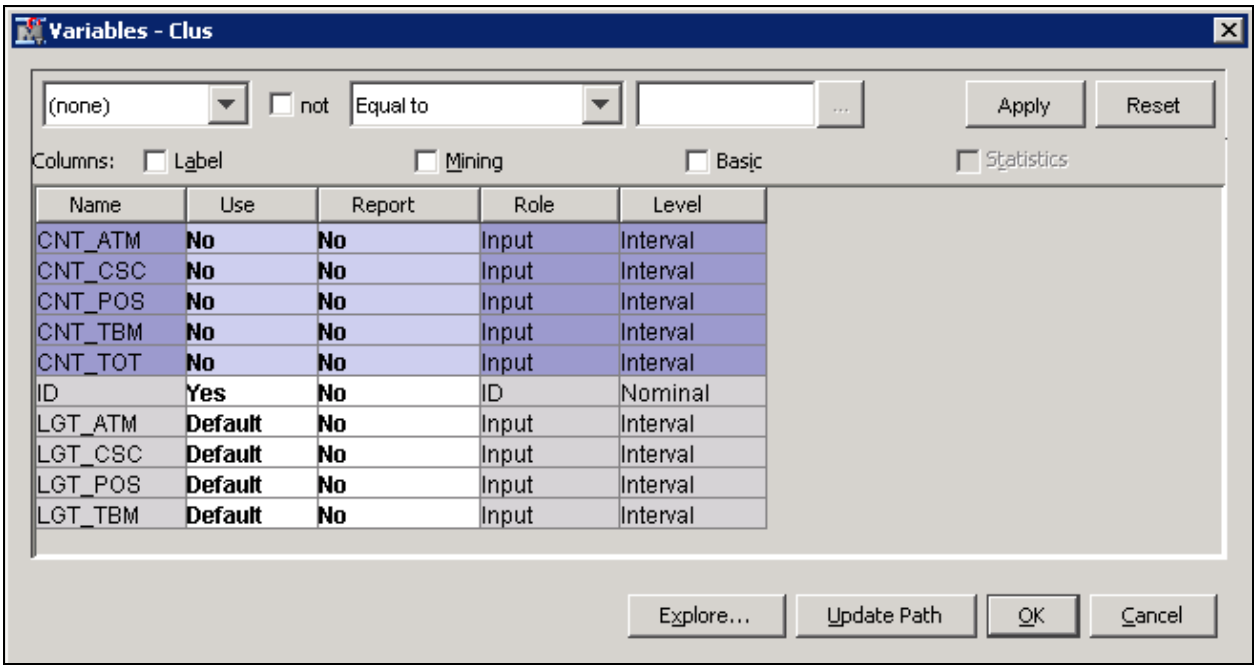


Two changes to the Cluster node default properties were made, as indicated below. Both were related to limiting the number of clusters created to 5.

Train	
Variables	
Cluster Variable Role	Segment
Internal Standardizatio	None
Number of Clusters	
Specification Method	User Specify
Maximum Number of C	5

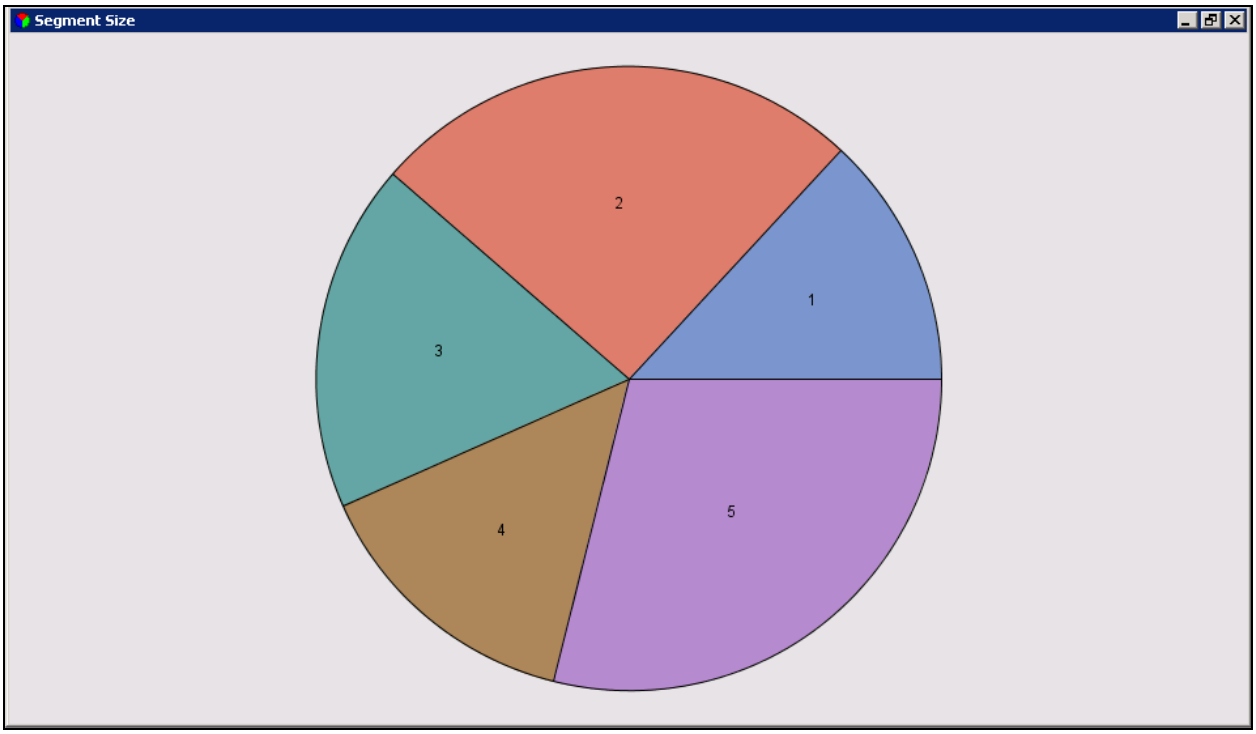
Because the inputs were all on the same measurement scale (category logit score), it was decided to ***not*** standardize the inputs.

Only the four LGT inputs defined in the Transform Variables node were set to **Default** in the Cluster node.



Name	Use	Report	Role	Level
CNT_ATM	**No**	**No**	Input	Interval
CNT_CSC	**No**	**No**	Input	Interval
CNT_POS	**No**	**No**	Input	Interval
CNT_TBM	**No**	**No**	Input	Interval
CNT_TOT	**No**	**No**	Input	Interval
ID	**Yes**	**No**	ID	Nominal
LGT_ATM	**Default**	**No**	Input	Interval
LGT_CSC	**Default**	**No**	Input	Interval
LGT_POS	**Default**	**No**	Input	Interval
LGT_TBM	**Default**	**No**	Input	Interval

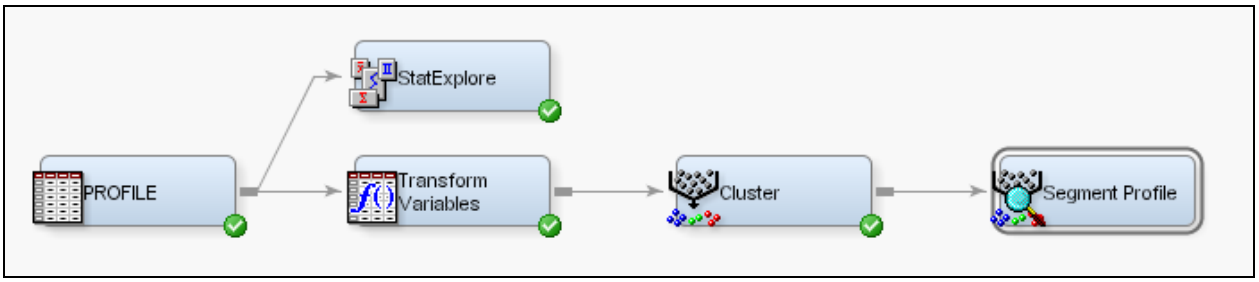
Running the Cluster node and viewing the Results window confirmed the creation of five nearly equally sized clusters.



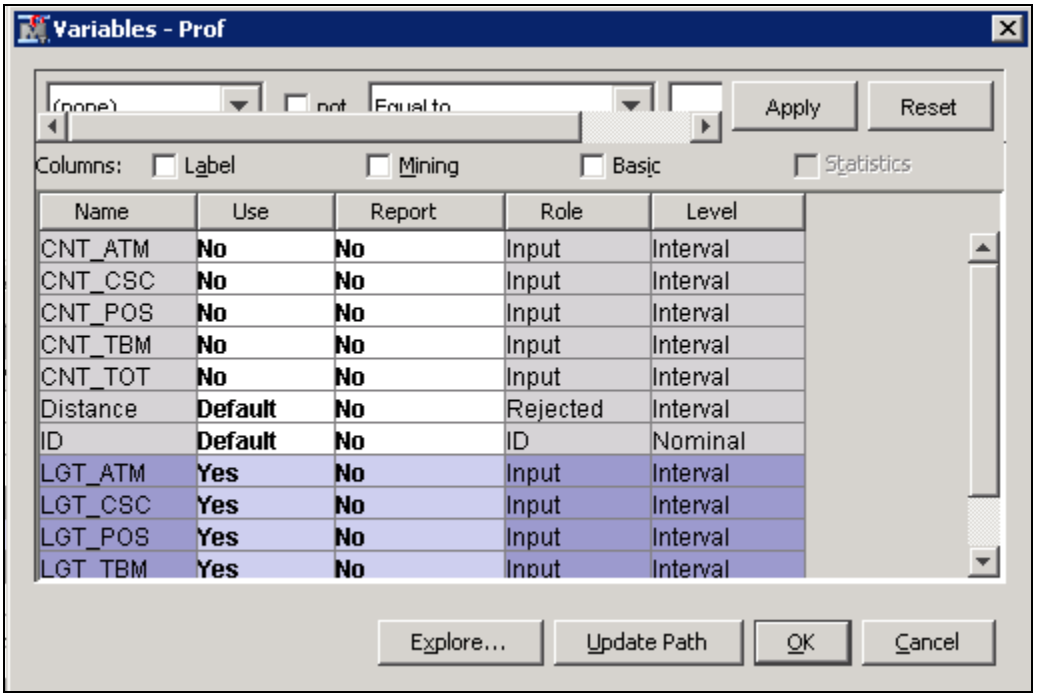
Additional cluster interpretations were made with the Segment Profile tool.

Interpreting Segments

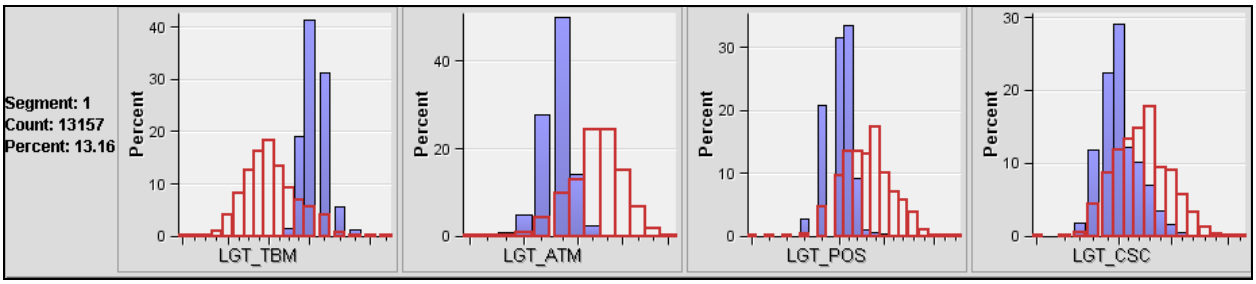
A Segment Profile node attached to the Cluster node helped to interpret the contents of the generated segments.



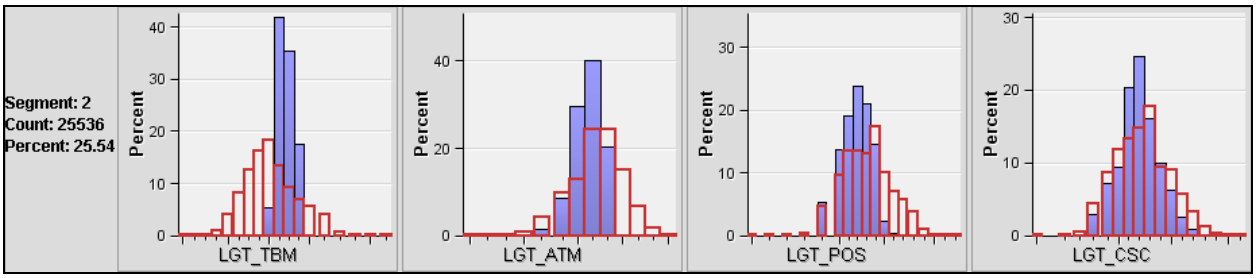
Only the **LGT** inputs were set to **Yes** in the Segment Profile node.



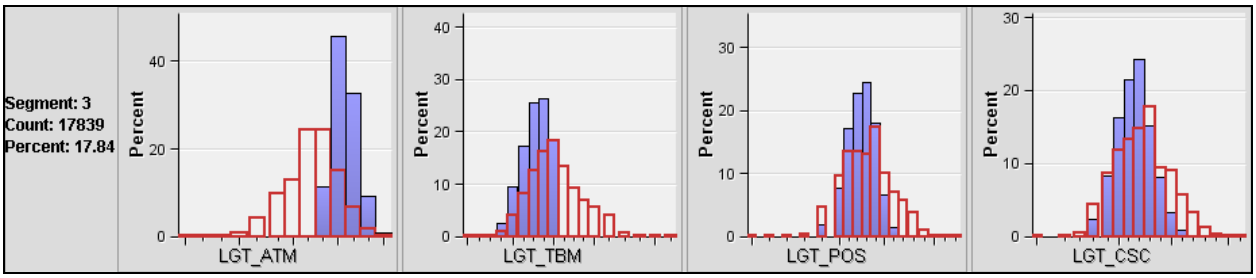
The following profiles were created for the generated segments:



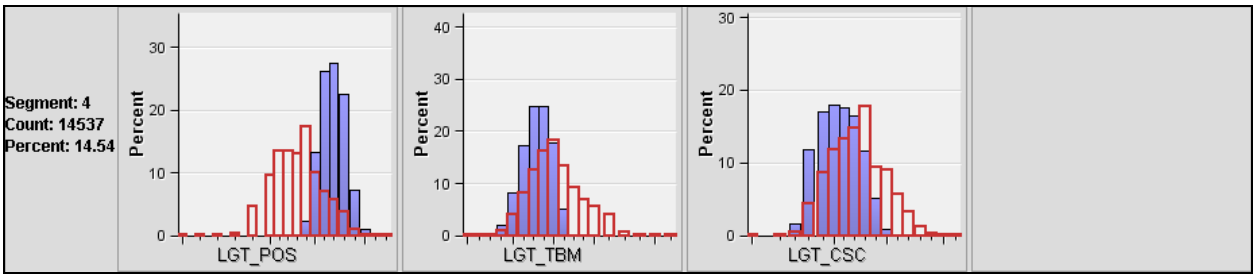
Segment 1 customers had a significantly higher than average use of traditional banking methods and lower than average use of all other transaction categories. This segment was labeled **Brick-and-Mortar**.



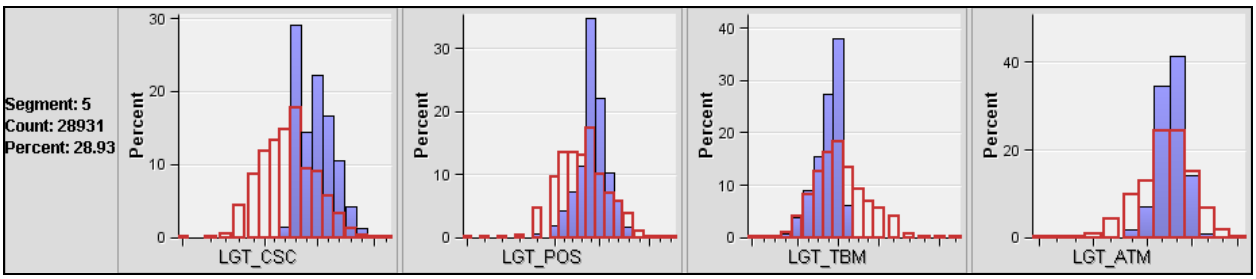
Segment 2 customers had a higher than average use of traditional banking methods but were close to the distribution centers on the other transaction categories. This segment was labeled **Transitionals** because they seem to be transitioning from brick-and-mortar to other usage patterns.



Segment 3 customers eschewed traditional banking methods in favor of ATMs. This segment was labeled **ATMs**.



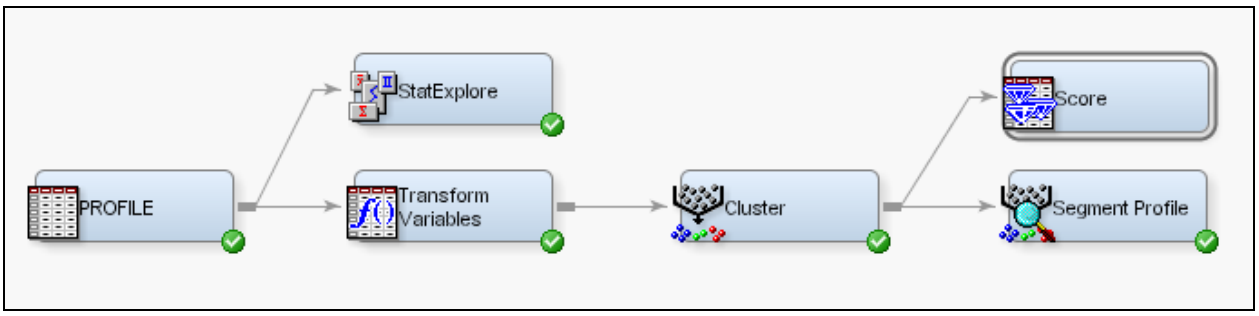
Segment 4 was characterized by a high prevalence of point-of-sale transactions and few traditional bank methods. This segment was labeled **Cashless**.



Segment 5 had a higher than average rate of customer service contacts and point-of-sale transactions. This segment was labeled **Service**.

Segment Deployment

Deployment of the transaction segmentation was facilitated by the Score node.



The Score node was attached to the Cluster node and run. The SAS Code window inside the Results window provided SAS code that was capable of transforming raw transaction counts to cluster assignments. The complete SAS scoring code is shown below.

```
*-------------------------------------------------------------*;
* Formula Code;
*-------------------------------------------------------------*;
LGT_TBM =log(CNT_TBM/(CNT_TOT-CNT_TBM)) ;
LGT_ATM =log(CNT_ATM/(CNT_TOT-CNT_ATM)) ;
LGT_POS =log(CNT_POS/(CNT_TOT - CNT_POS)) ;
LGT_CSC =log(CNT_CSC/(CNT_TOT-CNT_CSC)) ;
*-------------------------------------------------------------*;
* TOOL: Clustering;
* TYPE: EXPLORE;
* NODE: Clus;
*-------------------------------------------------------------*;
****************************************;
*** Begin Scoring Code from PROC DMVQ ***;
****************************************;


*** Begin Class Look-up, Standardization, Replacement ;
drop _dm_bad; _dm_bad = 0;

*** No transformation for LGT_ATM ;

*** No transformation for LGT_CSC ;

*** No transformation for LGT_POS ;

*** No transformation for LGT_TBM ;

*** End Class Look-up, Standardization, Replacement ;


*** Omitted Cases;
if _dm_bad then do;
   _SEGMENT_ = .; Distance = .;
   goto CLUSvlex ;
end; *** omitted;
```

```
* EM SCORE CODE;
* EM Version: 7.1;
* SAS Release: 9.03.01M0P060711;
* Host: SASBAP;
* Encoding: wlatin1;
* Locale: en_US;
* Project Path: D:\Workshop\winsas\EM_Projects;
* Project Name: apxa;
* Diagram Id: EMWS1;
* Diagram Name: case_study1;
* Generated by: sasdemo;
* Date: 09SEP2011:16:50:09;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
```

```
* TOOL: Input Data Source;
* TYPE: SAMPLE;
* NODE: Ids2;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* TOOL: Transform;
* TYPE: MODIFY;
* NODE: Trans;
*-------------------------------------------------------------*;
LGT_ATM = log(CNT_ATM/(CNT_TOT-CNT_ATM));
LGT_CSC = log(CNT_CSC/(CNT_TOT-CNT_CSC));
LGT_POS = log(CNT_POS/(CNT_TOT - CNT_POS));
LGT_TBM = log(CNT_TBM/(CNT_TOT-CNT_TBM));
*-------------------------------------------------------------*;
* TOOL: Clustering;
* TYPE: EXPLORE;
* NODE: Clus;
*-------------------------------------------------------------*;
****************************************;
*** Begin Scoring Code from PROC DMVQ ***;
****************************************;
*** Begin Class Look-up, Standardization, Replacement ;
drop _dm_bad; _dm_bad = 0;

*** No transformation for LGT_ATM ;
*** No transformation for LGT_CSC ;
*** No transformation for LGT_POS ;
*** No transformation for LGT_TBM ;
*** End Class Look-up, Standardization, Replacement ;
*** Omitted Cases;
if _dm_bad then do;
   _SEGMENT_ = .; Distance = .;
   goto CLUSvlex ;
end; *** omitted;
*** Compute Distances and Cluster Membership;
label _SEGMENT_ = 'Segment Id' ;
label Distance = 'Distance' ;
array CLUSvads [5] _temporary_;
drop _vqclus _vqmvar _vqnvar;
_vqmvar = 0;
do _vqclus = 1 to 5; CLUSvads [_vqclus] = 0; end;
if not missing( LGT_ATM ) then do;
   CLUSvads [1] + ( LGT_ATM - -3.54995114884545 )**2;
   CLUSvads [2] + ( LGT_ATM - -2.2003888516185 )**2;
   CLUSvads [3] + ( LGT_ATM - -0.23695023328541 )**2;
   CLUSvads [4] + ( LGT_ATM - -1.47814712774378 )**2;
   CLUSvads [5] + ( LGT_ATM - -1.49704375204907 )**2;
end;
else _vqmvar + 1.31533540479169;
if not missing( LGT_CSC ) then do;
   CLUSvads [1] + ( LGT_CSC - -4.16334022538952 )**2;
```

```
   CLUSvads [2] + ( LGT_CSC - -3.38356120535047 )**2;
   CLUSvads [3] + ( LGT_CSC - -3.55519058753002 )**2;
   CLUSvads [4] + ( LGT_CSC - -3.96526745641347 )**2;
   CLUSvads [5] + ( LGT_CSC - -2.08727391873096 )**2;
end;
else _vqmvar + 1.20270093291078;
if not missing( LGT_POS ) then do;
   CLUSvads [1] + ( LGT_POS - -4.08779761080977 )**2;
   CLUSvads [2] + ( LGT_POS - -3.27644694006697 )**2;
   CLUSvads [3] + ( LGT_POS - -3.02915771770446 )**2;
   CLUSvads [4] + ( LGT_POS - -0.9841959454775 )**2;
   CLUSvads [5] + ( LGT_POS - -2.21538937073223 )**2;
end;
else _vqmvar + 1.3094245726273;
if not missing( LGT_TBM ) then do;
   CLUSvads [1] + ( LGT_TBM - 2.62509260779666 )**2;
   CLUSvads [2] + ( LGT_TBM - 1.40885156098965 )**2;
   CLUSvads [3] + ( LGT_TBM - -0.15878507901546 )**2;
   CLUSvads [4] + ( LGT_TBM - -0.11252803970828 )**2;
   CLUSvads [5] + ( LGT_TBM - 0.22075831354075 )**2;
end;
else _vqmvar + 1.17502484629096;
_vqnvar = 5.00248575662075 - _vqmvar;
if _vqnvar <= 2.2748671456705E-12 then do;
   _SEGMENT_ = .; Distance = .;
end;
else do;
   _SEGMENT_ = 1; Distance = CLUSvads [1];
   _vqfzdst = Distance * 0.99999999999988; drop _vqfzdst;
   do _vqclus = 2 to 5;
      if CLUSvads [_vqclus] < _vqfzdst then do;
         _SEGMENT_ = _vqclus; Distance = CLUSvads [_vqclus];
         _vqfzdst = Distance * 0.99999999999988;
      end;
   end;
   Distance = sqrt(Distance * (5.00248575662075 / _vqnvar));
end;
CLUSvlex :;
*************************************;
*** End Scoring Code from PROC DMVQ ***;
*************************************;
*-------------------------------------------------------------*;
* Clus: Creating Segment Label;
*-------------------------------------------------------------*;
length _SEGMENT_LABEL_ $80;
label _SEGMENT_LABEL_='Segment Description';
if _SEGMENT_ = 1 then _SEGMENT_LABEL_="Cluster1";
else
if _SEGMENT_ = 2 then _SEGMENT_LABEL_="Cluster2";
else
if _SEGMENT_ = 3 then _SEGMENT_LABEL_="Cluster3";
```

```
else
if _SEGMENT_ = 4 then _SEGMENT_LABEL_="Cluster4";
else
if _SEGMENT_ = 5 then _SEGMENT_LABEL_="Cluster5";
*-------------------------------------------------------------*;
* TOOL: Score Node;
* TYPE: ASSESS;
* NODE: Score;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* Score: Creating Fixed Names;
*-------------------------------------------------------------*;
LABEL EM_SEGMENT = 'Segment Variable';
EM_SEGMENT = _SEGMENT_;
```

A.2 Web Site Usage Associations Case Study

Case Study Description

A radio station developed a Web site to broaden its audience appeal and its offerings. In addition to a simulcast of the station's primary broadcast, the Web site was designed to provide services to Web users, such as podcasts, news streams, music streams, archives, and live Web music performances. The station tracked usage of these services by URL. Analysts at the station wanted to see whether any unusual patterns existed in the combinations of services selected by its Web users.

The **WEBSTATION** data set contains services selected by more than 1.5 million unique Web users over a two-month period in 2006. For privacy reasons, the URLs are assigned anonymous ID numbers.

The diagram containing this analysis is stored as an XML file on the course data disk. You can open this file by right-clicking **Diagrams** ⇨ **Import Diagram from XML** in SAS Enterprise Miner. All nodes in the opened file, except the data node, contain the property settings outlined in this case study. If you want to run the diagram, you need to re-create the case study data set using the metadata settings indicated below.

Case Study Data

Name	Model Role	Measurement Level	Description
ID	ID	Nominal	URL (with anonymous ID numbers)
TARGET	Target	Nominal	Web service selected

The **WEBSTATION** data set should be assigned the role of **Transaction**. This role can be assigned either in the process of creating the data source or by changing the properties of the data source inside SAS Enterprise Miner.

Accessing and Assaying the Data

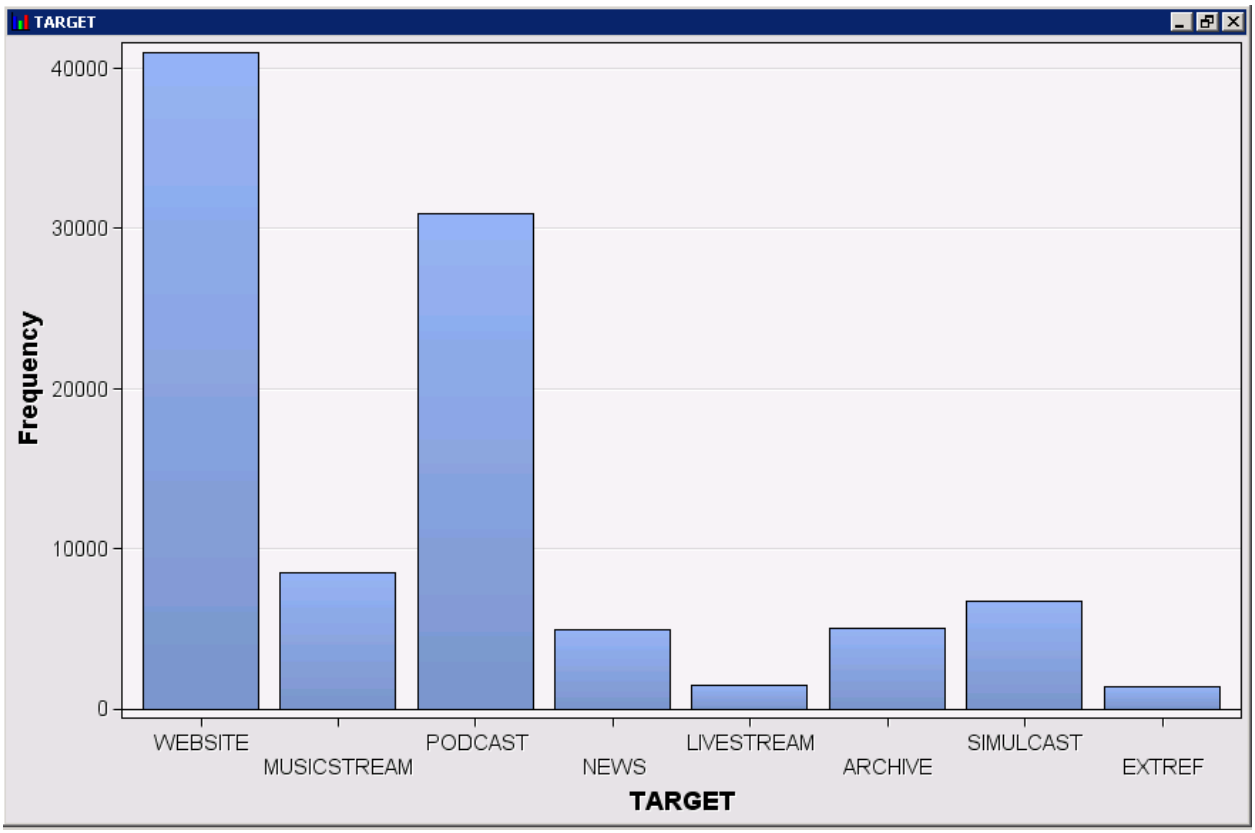
A SAS Enterprise Miner data source was defined for the **WEBSTATION** data set using the metadata settings indicated above. By right-clicking on the Data Source node in the diagram and selecting **Edit Variables**, the **TARGET** variable can be explored by highlighting the variable and then selecting **Explore**. (The following results are obtained by specifying **Random** and **Max** for the Sample Method and Fetch Size.)

The Sample Statistics window shows that there are over 128 unique URLs in the data set and 8 distinct services.

Sample Statistics

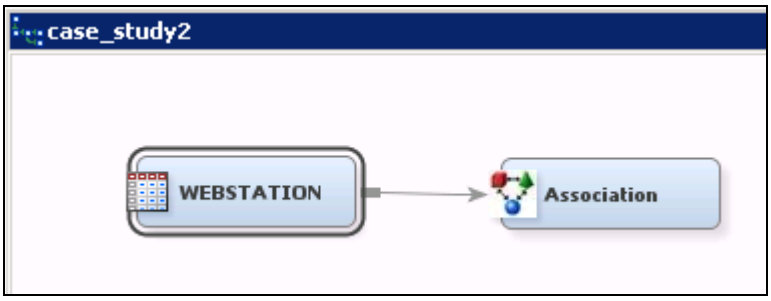
Obs #	Variable...	Type	Percent ...	Number ...	Mode Pe...	Mode
1	ID	CLASS	0	128+	1.481481	0000275
2	TARGET	CLASS	0	8	41.022	WEBSITE

A plot of target distribution (produced from the Explore window) identified the eight levels and displayed the relative frequency in a random sample of 100000 cases.

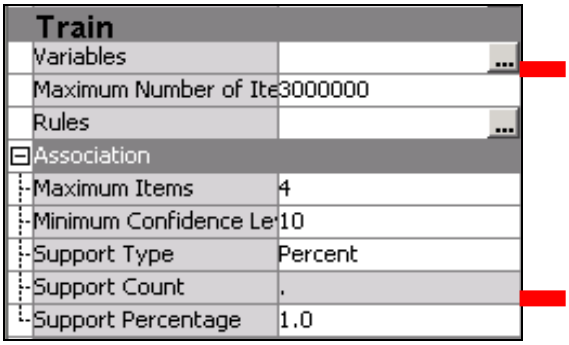


Generating Associations

An Association node was connected to the **WEBSTATION** node.



A preliminary run of the Association node yielded very few association rules. It was discovered that the default minimum Support Percentage setting was too large. (Many of the URLs selected only one service, diminishing the support of all association rules.) To obtain more association rules, the minimum Support Percentage setting was changed to **1.0**. In addition, the number of items to process was increased to **3000000** to account for the large training data set.



Using these changes, the analysis was rerun and yielded substantially more association rules.

The Rules Table was used to scrutinize the results.

Rules Table

Relations	Expected Confidence(%)	Confidence(%)	Support(%)	Lift	Transaction Count	Rule	Left Hand of Rule	Right Hand of Rule	Rule Item 1	Rule Item 2	Rule Item 3	Rule Item 4	Rule Item 5	Rule Index	Transpose Rule
3	7.32	98.32	1.69	13.42	26744	WEBSITE &...	WEBSITE &...	ARCHIVE	WEBSITE	EXTREF	========...	ARCHIVE		1	1
3	1.71	23.02	1.69	13.42	26744	ARCHIVE =...	ARCHIVE	WEBSITE &...	ARCHIVE	==========	========...	WEBSITE	EXTREF	2	1
2	7.32	98.07	1.92	13.39	30419	EXTREF ==...	EXTREF	ARCHIVE	EXTREF	==========	========...	ARCHIVE		3	1
2	1.96	26.19	1.92	13.39	30419	ARCHIVE =...	ARCHIVE	EXTREF	ARCHIVE	==========	========...	EXTREF		4	1
3	1.96	23.90	1.69	12.22	26744	WEBSITE &...	WEBSITE &...	EXTREF	WEBSITE	ARCHIVE	==========	========...	EXTREF	6	1
3	7.05	86.22	1.69	12.22	26744	EXTREF ==...	EXTREF	WEBSITE &...	EXTREF	==========	========...	WEBSITE	ARCHIVE	5	1
4	1.78	16.05	0.66	9.03	10424	WEBSITE &...	WEBSITE &...	PODCAST ...	WEBSITE	SIMULCAST	==========	========...	PODCAST	MUSICSTR...	7	1
4	4.10	36.97	0.66	9.03	10424	PODCAST ...	PODCAST ...	WEBSITE &...	PODCAST	MUSICSTR...	==========	========...	WEBSITE	SIMULCAST	8	1
4	1.58	12.29	0.66	7.80	10424	WEBSITE &...	WEBSITE &...	SIMULCAS...	WEBSITE	MUSICSTR...	==========	========...	SIMULCAST	PODCAST	9	1
4	5.35	41.71	0.66	7.80	10424	SIMULCAS...	SIMULCAS...	WEBSITE &...	SIMULCAST	PODCAST	==========	========...	WEBSITE	MUSICSTR...	10	1
3	9.47	64.45	0.90	6.81	14275	NEWS & M...	NEWS & M...	SIMULCAST	NEWS	MUSICSTR...	==========	========...	SIMULCAST		11	1
3	9.47	51.35	0.69	5.43	10944	WEBSITE &...	WEBSITE &...	SIMULCAST	WEBSITE	NEWS	==========	========...	SIMULCAST		12	1
4	9.47	44.86	0.66	4.74	10424	WEBSITE &...	WEBSITE &...	SIMULCAST	WEBSITE	PODCAST	MUSICSTR...	==========	========...	SIMULCAST	13	1
3	6.95	31.69	0.90	4.56	14275	SIMULCAS...	SIMULCAS...	NEWS	SIMULCAST	MUSICSTR...	==========	========...	NEWS		15	1
3	2.84	12.95	0.90	4.56	14275	NEWS ==> ...	NEWS	SIMULCAS...	NEWS	==========	========...	SIMULCAST	MUSICSTR...		14	1
3	9.47	41.55	0.74	4.39	11714	PODCAST ...	PODCAST ...	SIMULCAST	PODCAST	MUSICSTR...	==========	========...	SIMULCAST		16	1
4	11.83	51.44	0.66	4.35	10424	WEBSITE &...	WEBSITE &...	MUSICSTR...	WEBSITE	SIMULCAST	PODCAST	==========	========...	MUSICSTR...	17	1
3	11.83	46.87	0.74	3.96	11714	SIMULCAS...	SIMULCAS...	MUSICSTR...	SIMULCAST	PODCAST	==========	========...	MUSICSTR...		18	1
3	11.83	44.61	0.60	3.77	9506.0	WEBSITE &...	WEBSITE &...	MUSICSTR...	WEBSITE	NEWS	==========	========...	MUSICSTR...		19	1
3	11.83	44.00	0.90	3.72	14275	SIMULCAS...	SIMULCAS...	MUSICSTR...	SIMULCAST	NEWS	==========	========...	MUSICSTR...		20	1
3	11.83	38.17	1.56	3.23	24794	WEBSITE &...	WEBSITE &...	MUSICSTR...	WEBSITE	SIMULCAST	==========	========...	MUSICSTR...		22	1
3	4.10	13.21	1.56	3.23	24794	MUSICSTR...	MUSICSTR...	WEBSITE &...	MUSICSTR...	==========	========...	WEBSITE	SIMULCAST		21	1
2	9.47	29.43	2.05	3.11	32444	NEWS ==> ...	NEWS	SIMULCAST	NEWS	==========	========...	SIMULCAST			23	1
2	6.95	21.61	2.05	3.11	32444	SIMULCAS...	SIMULCAST	NEWS	SIMULCAST	==========	========...	NEWS			24	1
3	9.47	29.24	1.56	3.09	24794	WEBSITE &...	WEBSITE &...	SIMULCAST	WEBSITE	MUSICSTR...	==========	========...	SIMULCAST		25	1
3	5.35	16.51	1.56	3.09	24794	SIMULCAS...	SIMULCAST	WEBSITE &...	SIMULCAST	==========	========...	WEBSITE	MUSICSTR...		26	1
2	11.83	30.01	2.84	2.54	45051	SIMULCAS...	SIMULCAST	MUSICSTR...	SIMULCAST	==========	========...	MUSICSTR...			27	1
2	9.47	24.01	2.84	2.54	45051	MUSICSTR...	MUSICSTR...	SIMULCAST	MUSICSTR...	==========	========...	SIMULCAST			28	1
3	7.32	18.30	0.75	2.50	11890	WEBSITE &...	WEBSITE &...	ARCHIVE	WEBSITE	SIMULCAST	==========	========...	ARCHIVE		30	1
3	4.10	10.24	0.75	2.50	11890	ARCHIVE =...	ARCHIVE	WEBSITE &...	ARCHIVE	==========	========...	WEBSITE	SIMULCAST		29	1
3	6.95	16.85	0.69	2.42	10944	WEBSITE &...	WEBSITE &...	NEWS	WEBSITE	SIMULCAST	==========	========...	NEWS		31	1
3	7.32	17.53	0.94	2.39	14861	WEBSITE &...	WEBSITE &...	ARCHIVE	WEBSITE	MUSICSTR...	==========	========...	ARCHIVE		32	1
3	5.35	12.79	0.94	2.39	14861	ARCHIVE =	ARCHIVE	WEBSITE &	ARCHIVE	==========	WEBSITE	MUSICSTR			33	1

The following were among the interesting findings from this analysis:

- Most external referrers to the Web site pointed to the programming archive (98% confidence).
- Selecting the simulcast service tripled the chances of selecting the news service.
- Users who streamed music, downloaded podcasts, used the news service, or listened to the simulcast were less likely to go to the Web site.

A.3 Credit Risk Case Study

A bank sought to use performance on an in-house subprime credit product to create an updated risk model. The risk model was to be combined with other factors to make future credit decisions.

A sample of applicants for the original credit product was selected. Credit bureau data describing these individuals (at the time of application) was recorded. The ultimate disposition of the loan was determined (paid off or bad debt). For loans rejected at the time of application, a disposition was inferred from credit bureau records on loans obtained in a similar time frame.

The credit scoring models pursued in this case study were required to conform to the standard industry practice of transparency and interpretability. This eliminated certain modeling tools from consideration (for example, neural networks) except for comparison purposes. If a neural network significantly outperformed a regression, for example, it could be interpreted as a sign of lack of fit for the regression. Measures could then be taken to improve the regression model.

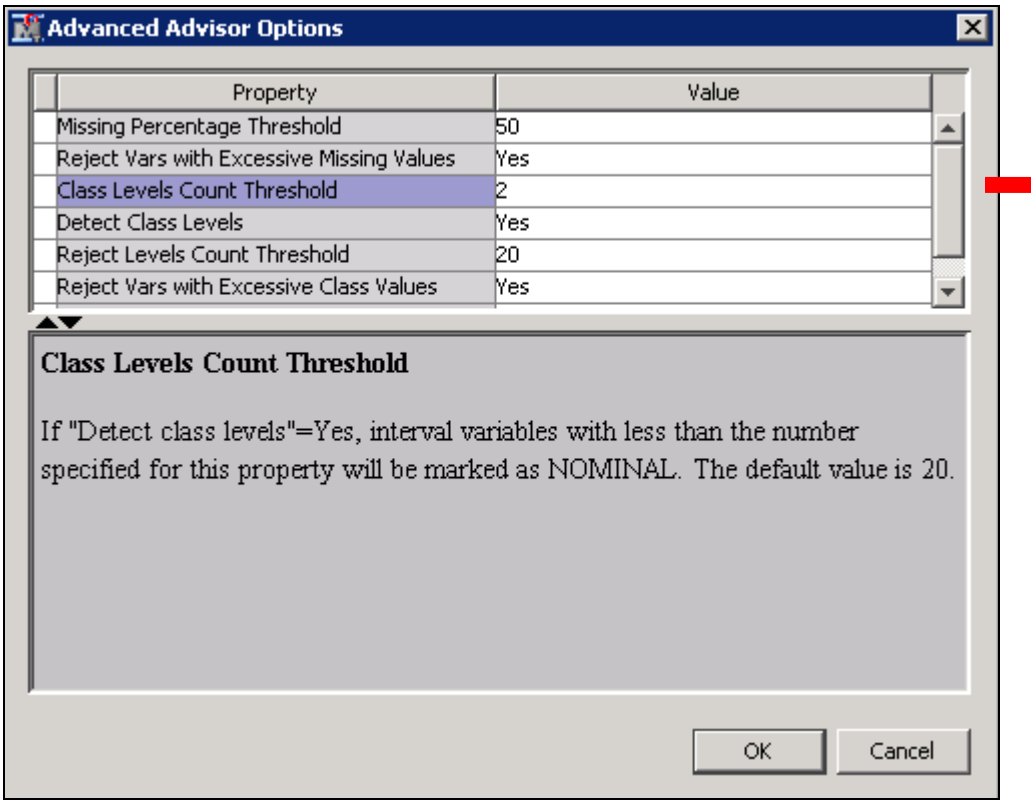
The diagram containing this analysis is stored as an XML file on the course data disk. You can open this file by right-clicking **Diagrams** ⇨ **Import Diagram from XML** in SAS Enterprise Miner. All nodes in the opened file, except the data node, contain the property settings outlined in this case study. If you want to run the diagram, you need to re-create the case study data set using the metadata settings indicated below.

Case Study Training Data

Name	Role	Level /	Label
TARGET	**Target**	**Binary**	
BanruptcyInd	**Input**	**Binary**	Bankruptcy Indicator
TLBadDerogCnt	**Input**	**Interval**	Number Bad Dept plus Public Derogatories
CollectCnt	**Input**	**Interval**	Number Collections
InqFinanceCnt24	**Input**	**Interval**	Number Finance Inquires 24 Months
InqCnt06	**Input**	**Interval**	Number Inquiries 6 Months
DerogCnt	**Input**	**Interval**	Number Public Derogatories
TLDel3060Cnt24	**Input**	**Interval**	Number Trade Lines 30 or 60 Days 24 Months
TL50UtilCnt	**Input**	**Interval**	Number Trade Lines 50 pct Utilized
TLDel60Cnt24	**Input**	**Interval**	Number Trade Lines 60 Days or Worse 24 Months
TLDel60CntAll	**Input**	**Interval**	Number Trade Lines 60 Days or Worse Ever
TL75UtilCnt	**Input**	**Interval**	Number Trade Lines 75 pct Utilized
TLDel90Cnt24	**Input**	**Interval**	Number Trade Lines 90+ 24 Months
TLBadCnt24	**Input**	**Interval**	Number Trade Lines Bad Debt 24 Months
TLDel60Cnt	**Input**	**Interval**	Number Trade Lines Currently 60 Days or Worse
TLSatCnt	**Input**	**Interval**	Number Trade Lines Currently Satisfactory
TLCnt12	**Input**	**Interval**	Number Trade Lines Opened 12 Months
TLCnt24	**Input**	**Interval**	Number Trade Lines Opened 24 Months
TLCnt03	**Input**	**Interval**	Number Trade Lines Opened 3 Months
TLSatPct	**Input**	**Interval**	Percent Satisfactory to Total Trade Lines
TLBalHCPct	**Input**	**Interval**	Percent Trade Line Balance to High Credit
TLOpenPct	**Input**	**Interval**	Percent Trade Lines Open
TLOpen24Pct	**Input**	**Interval**	Percent Trade Lines Open 24 Months
TLTimeFirst	**Input**	**Interval**	Time Since First Trade Line
InqTimeLast	**Input**	**Interval**	Time Since Last Inquiry
TLTimeLast	**Input**	**Interval**	Time Since Last Trade Line
TLSum	**Input**	**Interval**	Total Balance All Trade Lines
TLMaxSum	**Input**	**Interval**	Total High Credit All Trade Lines
TLCnt	**Input**	**Interval**	Total Open Trade Lines
ID	**ID**	**Nominal**	

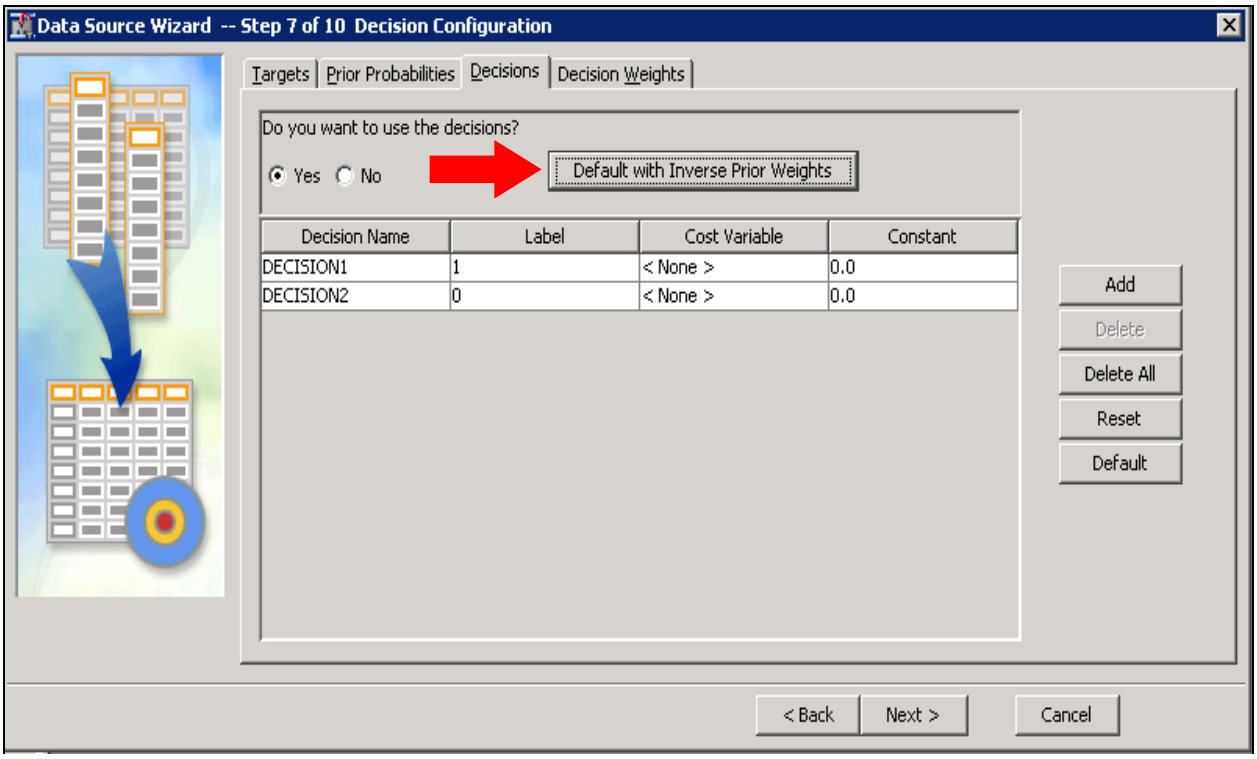
Accessing and Assaying the Data

A SAS Enterprise Miner data source was defined for the **CREDIT** data set using the metadata settings indicated above. The Data source definition was expedited by customizing the Advanced Metadata Advisor in the Data Source Wizard as indicated.

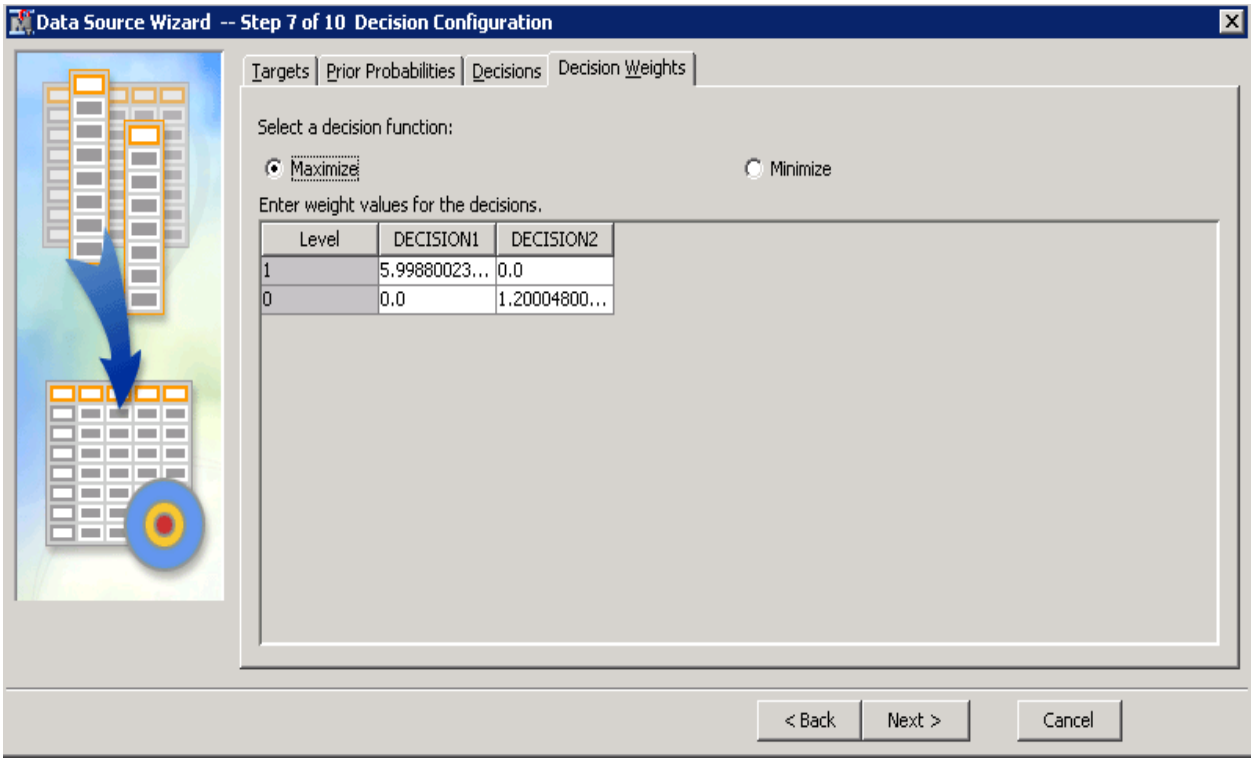


With this change, all metadata was set correctly by default.

Decision processing was selected in step 6 of the Data Source Wizard.

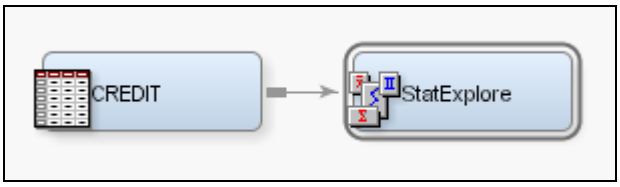


The Decisions option **Default with Inverse Prior Weights** was selected to provide the values in the Decision Weights tab.



It can be shown that, theoretically, the so-called central decision rule optimizes model performance based on the KS statistic.

The StatExplore node was used to provide preliminary statistics on the target variable.



BanruptcyInd and **TARGET** were the only two class variables in the **CREDIT** data set.

Class Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	BanruptcyInd	INPUT	2	0	0	84.67	1	15.33
TRAIN	TARGET	TARGET	2	0	0	83.33	1	16.67

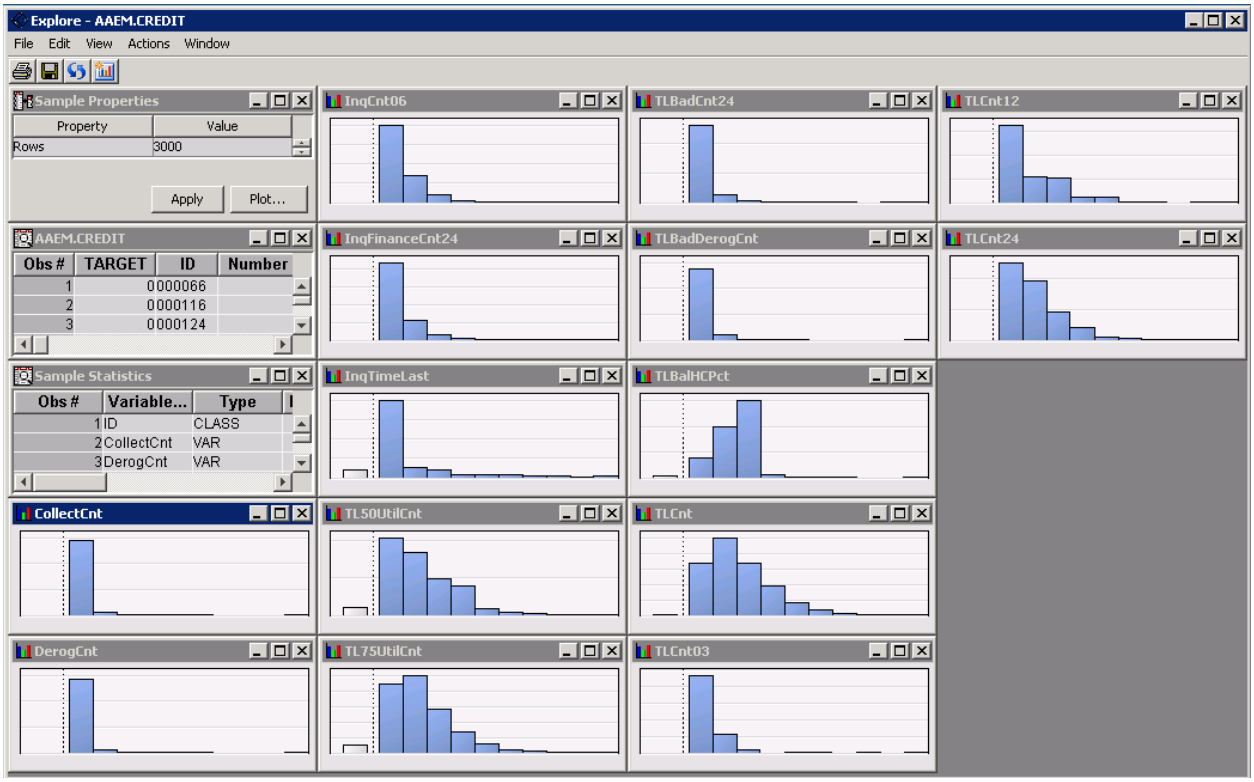
The Interval Variable Summary shows missing values on 11 of the 27 interval inputs.

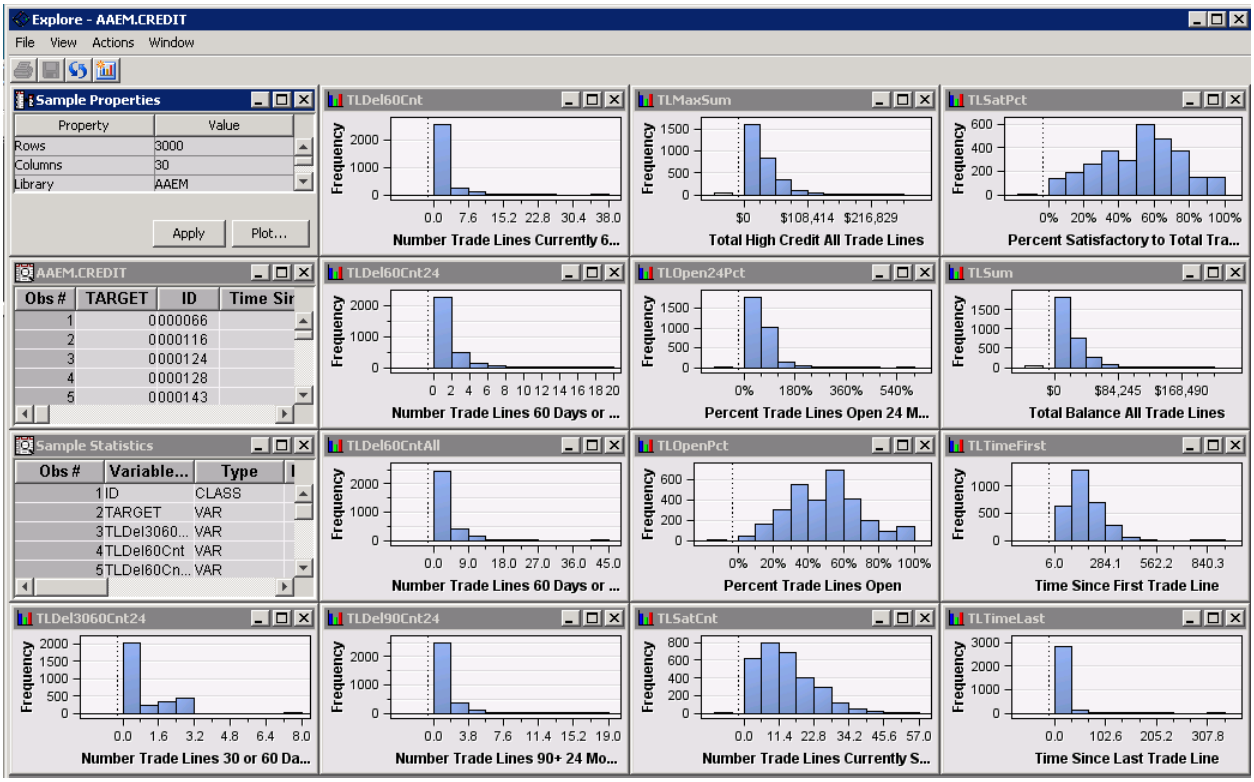
Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
CollectCnt	INPUT	0.857	2.161352	3000	0	0	0	50	7.556541	111.8365
DerogCnt	INPUT	1.43	2.731469	3000	0	0	0	51	5.045122	50.93801
InqCnt06	INPUT	3.108333	3.479171	3000	0	0	2	40	2.580016	12.82077
InqFinanceCnt24	INPUT	3.555	4.477536	3000	0	0	2	48	2.806893	13.05141
InqTimeLast	INPUT	3.108108	4.637831	2812	188	0	1	24	2.386563	5.626803
TL50UtilCnt	INPUT	4.077904	3.108076	2901	99	0	3	23	1.443077	3.350659
TL75UtilCnt	INPUT	3.121682	2.605435	2901	99	0	3	20	1.50789	3.686636
TLBadCnt24	INPUT	0.567	1.324423	3000	0	0	0	16	4.376858	28.58301
TLBadDerogCnt	INPUT	1.409	2.460434	3000	0	0	0	47	4.580204	48.24276
TLBalHCPct	INPUT	0.648178	0.266486	2959	41	0	0.6955	3.3613	-0.18073	4.015619
TLCnt	INPUT	7.879546	5.421595	2997	3	0	7	40	1.235579	2.195363
TLCnt03	INPUT	0.275	0.582084	3000	0	0	0	7	2.805575	12.66839
TLCnt12	INPUT	1.821333	1.925265	3000	0	0	1	15	1.623636	3.684793
TLCnt24	INPUT	3.882333	3.396714	3000	0	0	3	28	1.60771	4.379948
TLDel3060Cnt24	INPUT	0.726	1.163633	3000	0	0	0	8	1.381942	1.408509
TLDel60Cnt	INPUT	1.522	2.809653	3000	0	0	0	38	3.30846	17.76184
TLDel60Cnt24	INPUT	1.068333	1.806124	3000	0	0	0	20	3.080191	14.35044
TLDel60CntAll	INPUT	2.522	3.407255	3000	0	0	1	45	2.564126	12.70062
TLDel90Cnt24	INPUT	0.814667	1.609508	3000	0	0	0	19	3.623972	19.7006
TLMaxSum	INPUT	31205.9	29092.91	2960	40	0	24187	271036	2.061138	8.093434
TLOpen24Pct	INPUT	0.564219	0.480105	2997	3	0	0.5	6	2.779055	18.5329
TLOpenPct	INPUT	0.496168	0.206722	2997	3	0	0.5	1	0.379339	-0.01934
TLSatCnt	INPUT	13.51168	8.931769	2996	4	0	12	57	0.851193	0.690344
TLSatPct	INPUT	0.518331	0.234759	2996	4	0	0.5263	1	-0.12407	-0.48393
TLSum	INPUT	20151.1	19682.09	2960	40	0	15546	210612	2.276832	10.96413
TLTimeFirst	INPUT	170.1137	92.8137	3000	0	6	151	933	1.031307	2.860035
TLTimeLast	INPUT	11.87367	16.32141	3000	0	0	7	342	6.447907	80.31043

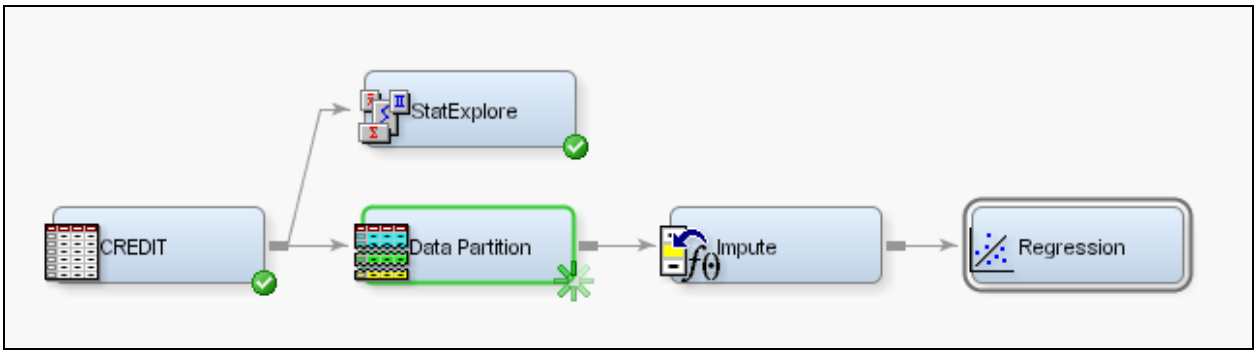
By creating plots using the Explore window, it was found that several of the interval inputs show somewhat skewed distributions. Transformation of the more severe cases was pursued in regression modeling.





Creating Prediction Models: Simple Stepwise Regression

Because it was the most likely model to be selected for deployment, a regression model was considered first.



- In the Data Partition node, 50% of the data was chosen for training and 50% for validation.
- The Impute node replaced missing values for the interval inputs with the input mean (the default for interval valued input variables), and added unique imputation indicators for each input with missing values.
- The Regression node used the stepwise method for input variable selection, and validation profit for complexity optimization.

The selected model included seven inputs. See line 1197 of the Output window.

Analysis of Maximum Likelihood Estimates

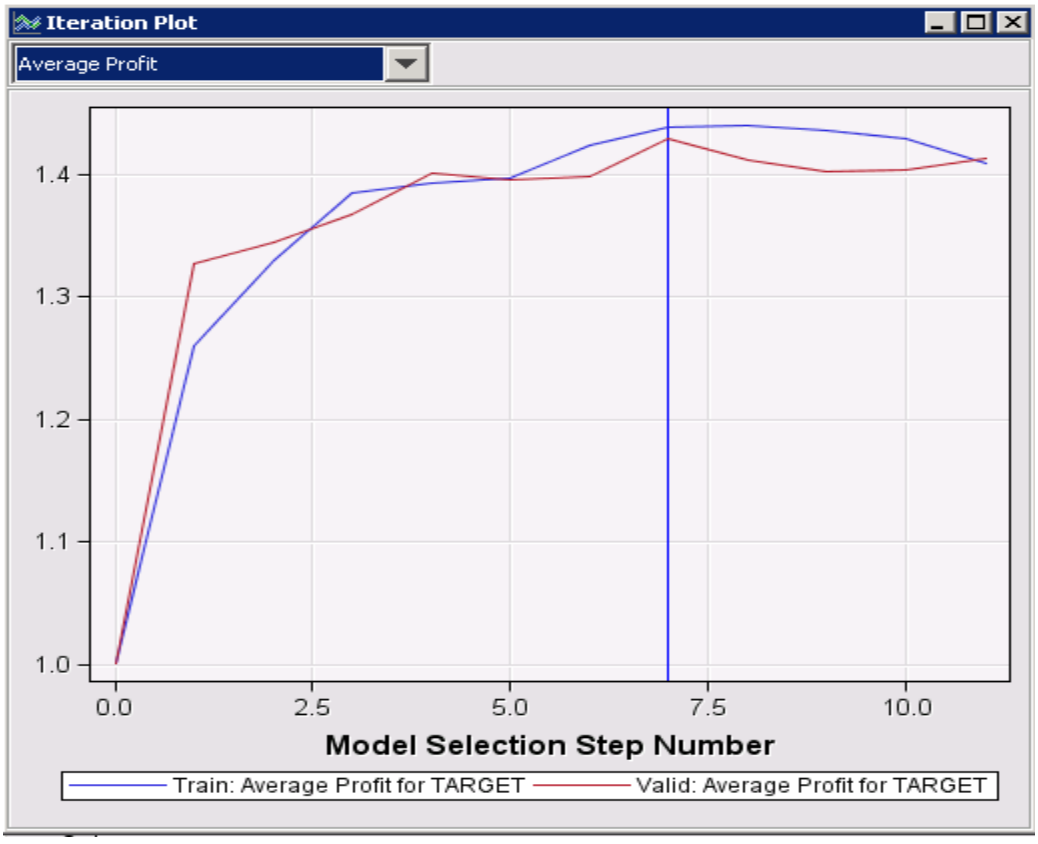
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq	Standardized Estimate	Exp(Est)
Intercept	1	-2.7602	0.4089	45.57	<.0001		0.063
IMP_TLBalHCPct	1	1.8759	0.3295	32.42	<.0001	0.2772	6.527
IMP_TLSatPct	1	-2.6095	0.4515	33.40	<.0001	-0.3363	0.074
InqFinanceCnt24	1	0.0610	0.0149	16.86	<.0001	0.1527	1.063
TLDel3060Cnt24	1	0.3359	0.0623	29.11	<.0001	0.2108	1.399
TLDel60Cnt24	1	0.1126	0.0408	7.62	0.0058	0.1102	1.119
TLOpenPct	1	1.5684	0.4633	11.46	0.0007	0.1792	4.799
TLTimeFirst	1	-0.00253	0.000923	7.50	0.0062	-0.1309	0.997

The odds ratio estimates facilitated model interpretation. Increasing risk was associated with increasing values of **IMP_TLBalHCPct**, **InqFinanceCnt24**, **TLDel3060Cnt24**, **TLDel60Cnt**, and **TLOpenPct**. Increasing risk was associated with decreasing values of **IMP_TLSatPct** and **TLTimeFirst**.

Odds Ratio Estimates

Effect	Point Estimate
IMP_TLBalHCPct	6.527
IMP_TLSatPct	0.074
InqFinanceCnt24	1.063
TLDel3060Cnt24	1.399
TLDel60Cnt24	1.119
TLOpenPct	4.799
TLTimeFirst	0.997

The iteration plot (found by selecting **View** ⇨ **Model** ⇨ **Iteration Plot** in the Results window) can be set to show average profit versus iteration.

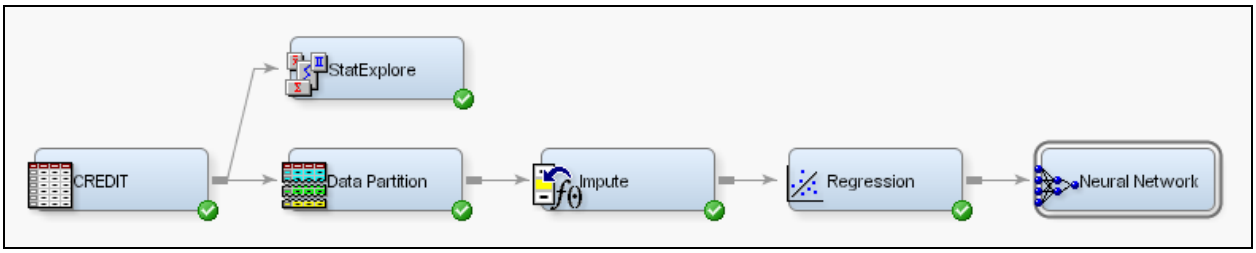


In theory, the average profit for a model using the defined profit matrix equals 1+KS statistic. Thus, the iteration plot (from the Regression node's Results window) showed how the profit (or, in turn, the KS statistic) varied with model complexity. From the plot, the maximum validation profit equaled 1.43, which implies that the maximum KS statistic equaled 0.43.

The actual calculated value of KS (as found using the Model Comparison node) was found to differ slightly from this value (see below).

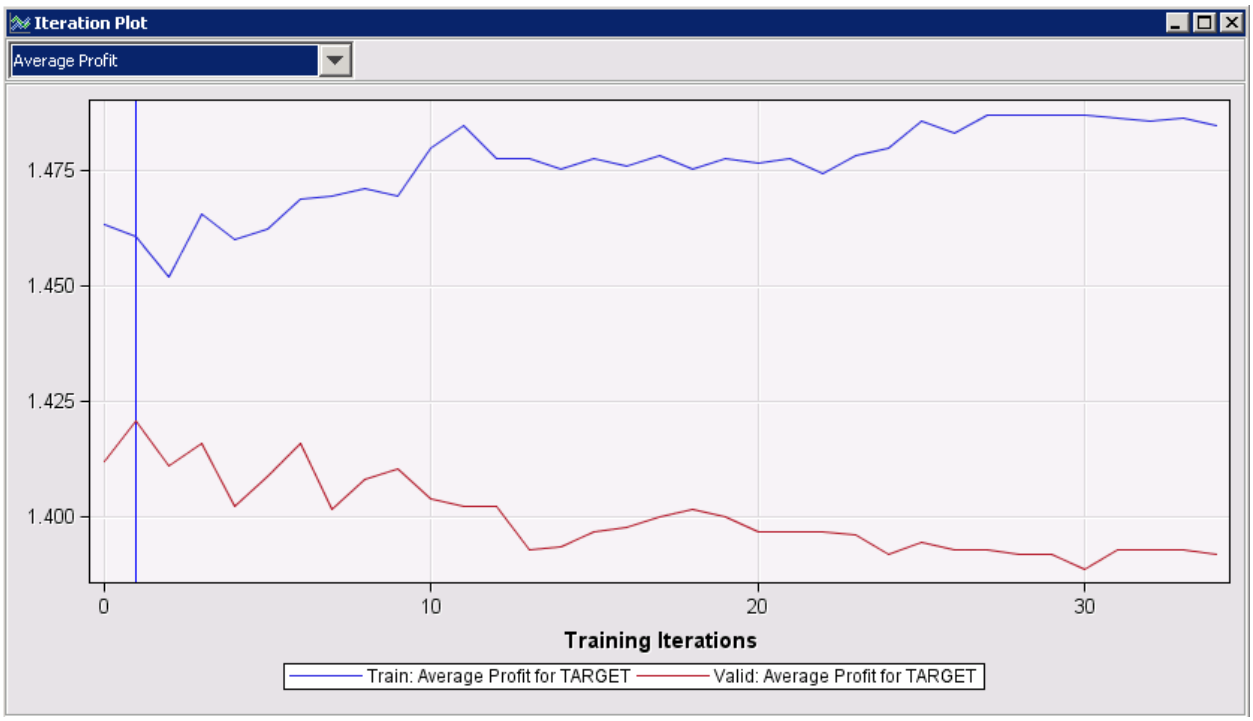
Creating Prediction Models: Neural Network

While it is not possible to deploy as the final prediction model, a neural network was used to investigate regression lack of fit.



The default settings of the Neural Network node were used in combination with inputs selected by the Stepwise Regression node.

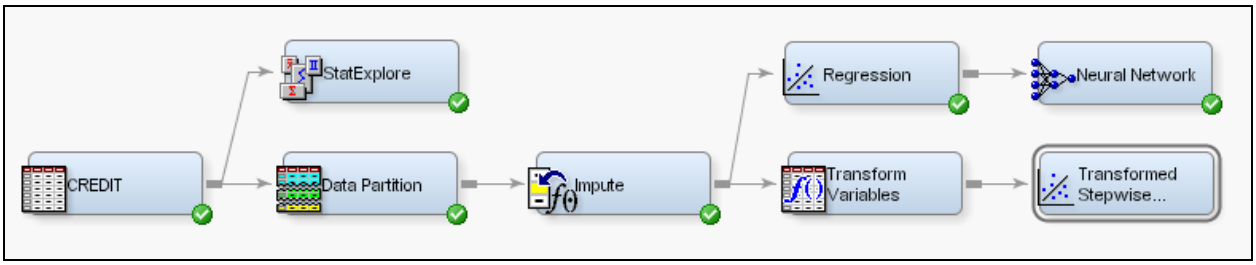
The iteration plot showed slightly higher validation average profit compared to the stepwise regression model.



It was possible (although not likely) that transformations to the regression inputs could improve regression prediction.

Creating Prediction Models: Transformed Stepwise Regression

In assaying the data, it was noted that some of the inputs had rather skewed distributions. Such distributions create high leverage points that can distort an input's association with the target. The Transform Variables node was used to regularize the distributions of the model inputs before fitting the stepwise regression.



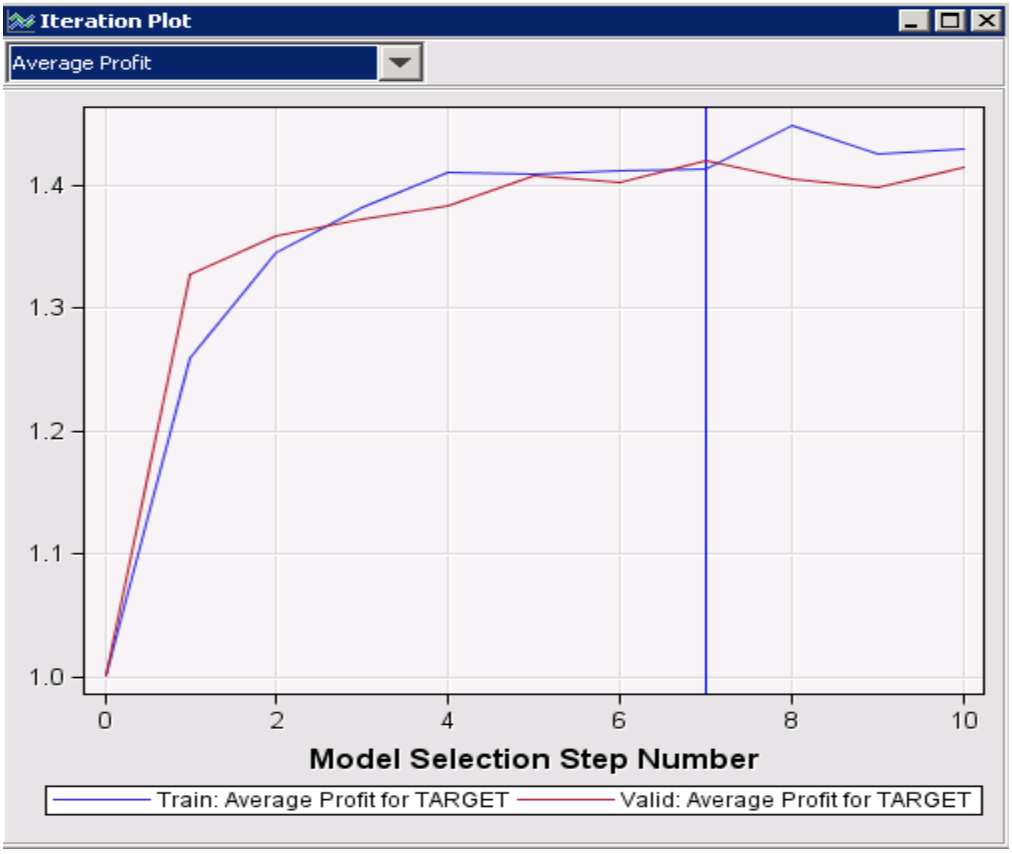
The Transform Variables node was set to maximize the normality of each interval input by selecting from one of several power and logarithmic transformations.

Property	Value
General	
Node ID	Trans
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Formulas	
Interactions	
SAS Code	
Default Methods	
Interval Inputs	Maximum Normal
Interval Targets	None
Class Inputs	None
Class Targets	None

The Transformed Stepwise Regression node performed stepwise selection from the transformed inputs. The selected model had many of the same inputs as the original stepwise regression model, but on a transformed (and difficult to interpret) scale.

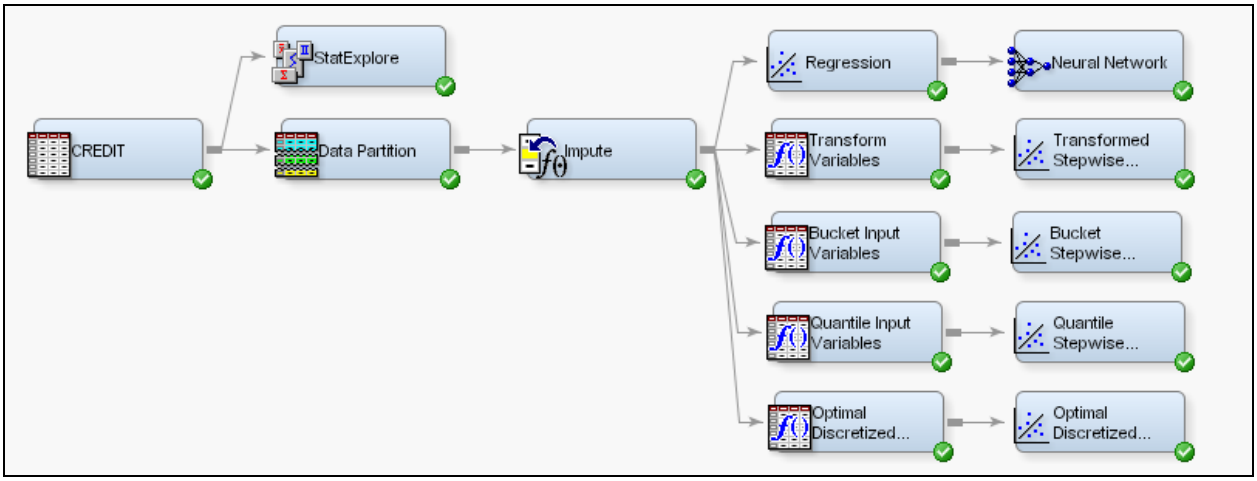
```
1105          Odds Ratio Estimates
1106
1107                                      Point
1108  Effect                           Estimate
1109
1110  IMP_TLBalHCPct                      4.237
1111  IMP_TLSatPct                        0.090
1112  LOG_InqFinanceCnt24                 9.648
1113  LOG_TLDel60Cnt24                    9.478
1114  SQRT_IMP_TL75UtilCnt                5.684
1115  SQRT_TLDel3060Cnt24                 4.708
1116  SQRT_TLTimeFirst                    0.050
```

The transformations would be justified (despite the increased difficulty in model interpretation) if they resulted in significant improvement in model fit. Based on the profit calculation, the transformed stepwise regression model showed only marginal performance improvement compared to the original stepwise regression model.



Creating Prediction Models: Discretized Stepwise Regression

Partitioning input variables into discrete ranges was another common risk-modeling method that was investigated.



Three discretization approaches were investigated. The Bucket Input Variables node partitioned each interval input into four bins with equal ***widths***. The Bin Input Variables node partitioned each interval input into four bins with equal ***sizes***. The Optimal Discrete Input Variables node found optimal partitions for each input variable using decision tree methods.

Bucket Transformation

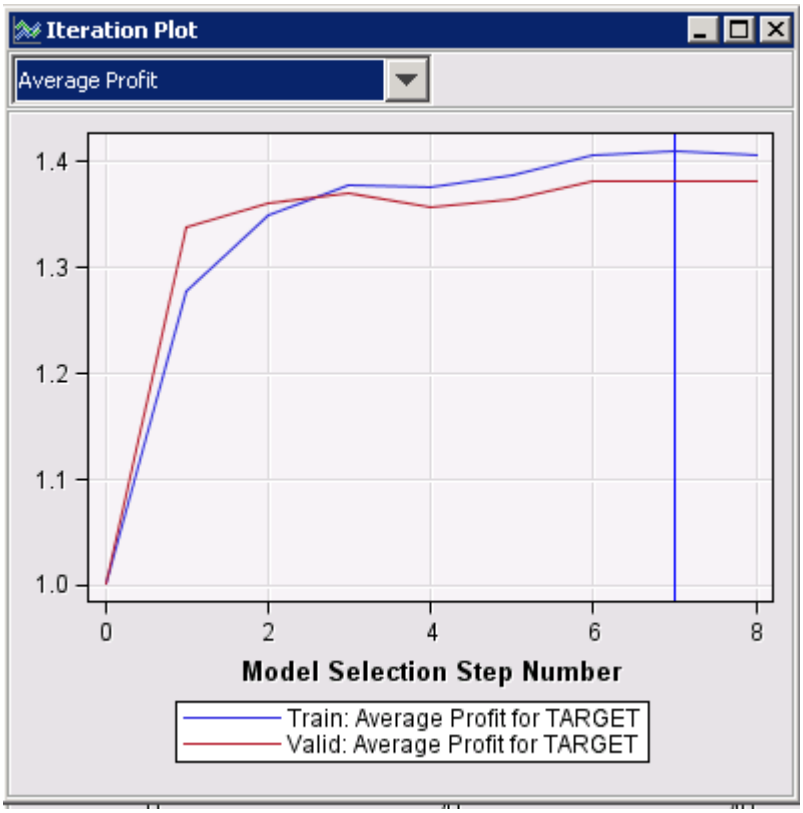
The relatively small size of the **CREDIT** data set resulted in problems for the bucket stepwise regression model. Many of the bins had a small number of observations, which resulted in quasi-complete separation problems for the regression model, as dramatically illustrated by the selected model's odds ratio report. Go to line 1059 of the Output window.

```
                          Odds Ratio Estimates


Point
          Effect
Estimate
          BIN_IMP_TL75UtilCnt 01:low -5 vs 04:15-high
999.000
          BIN_IMP_TL75UtilCnt 02:5-10 vs 04:15-high
999.000
          BIN_IMP_TL75UtilCnt 03:10-15 vs 04:15-high
999.000
          BIN_IMP_TLBalHCPct  01:low -0.840325 vs 04:2.520975-high
<0.001
          BIN_IMP_TLBalHCPct  02:0.840325-1.68065 vs 04:2.520975-high
<0.001
          BIN_IMP_TLBalHCPct  03:1.68065-2.520975 vs 04:2.520975-high
<0.001
          BIN_IMP_TLSatPct    01:low -0.25 vs 04:0.75-high
4.845
          BIN_IMP_TLSatPct    02:0.25-0.5 vs 04:0.75-high
1.819
          BIN_IMP_TLSatPct    03:0.5-0.75 vs 04:0.75-high
1.009
          BIN_InqFinanceCnt24 01:low -9.75 vs 04:29.25-high
0.173
          BIN_InqFinanceCnt24 02:9.75-19.5 vs 04:29.25-high
0.381
          BIN_InqFinanceCnt24 03:19.5-29.25 vs 04:29.25-high
0.640
          BIN_TLDel3060Cnt24  01:low -2 vs 04:6-high
999.000
          BIN_TLDel3060Cnt24  02:2-4 vs 04:6-high
999.000
          BIN_TLDel60CntAll   01:low -4.75 vs 04:14.25-high
0.171
```

```
          BIN_TLDel60CntAll   02:4.75-9.5 vs 04:14.25-high
0.138
          BIN_TLDel60CntAll   03:9.5-14.25 vs 04:14.25-high
0.166
          BIN_TLTimeFirst     01:low -198.75 vs 04:584.25-high
999.000
          BIN_TLTimeFirst     02:198.75-391.5 vs 04:584.25-high
999.000
          BIN_TLTimeFirst     03:391.5-584.25 vs 04:584.25-high
999.000
```

The iteration plot showed substantially worse performance compared to the other modeling efforts.



Bin (or Quantile) Transformation

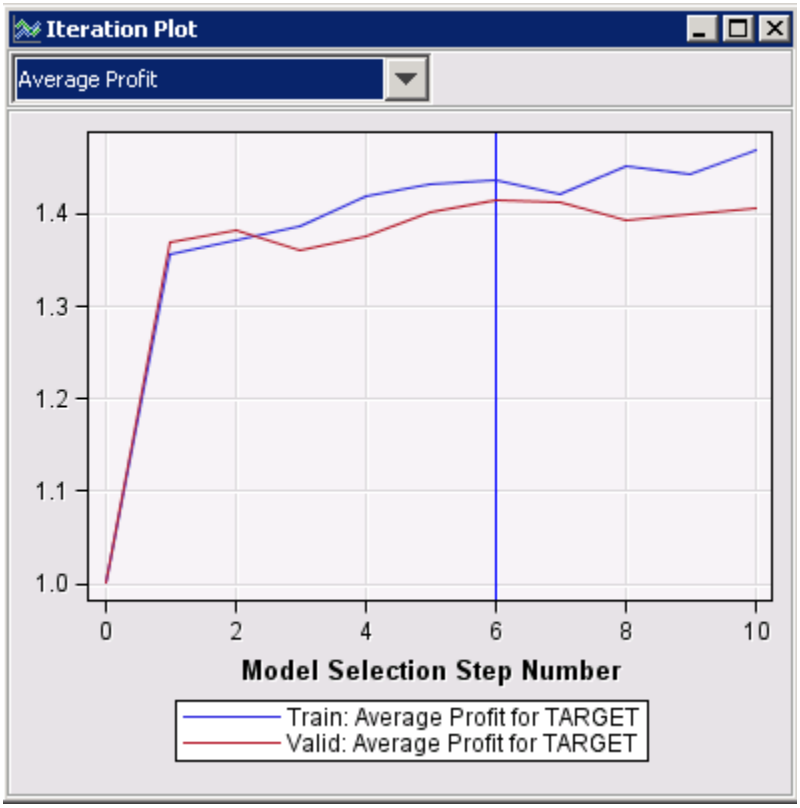
Somewhat better results were seen with the binned stepwise regression model. By ensuring that each bin included a reasonable number of cases, more stable model parameter estimates could be made. See line 1249 of the Output window.

```
                              Odds Ratio Estimates


Point
            Effect
Estimate
```

```
             PCTL_IMP_TLBalHCPct  01:low -0.513 vs 04:0.8389-high
0.272
             PCTL_IMP_TLBalHCPct  02:0.513-0.7041 vs 04:0.8389-high
0.452
             PCTL_IMP_TLBalHCPct  03:0.7041-0.8389 vs 04:0.8389-high
0.630
             PCTL_IMP_TLSatPct    01:low -0.3529 vs 04:0.6886-high
1.860
             PCTL_IMP_TLSatPct    02:0.3529-0.5333 vs 04:0.6886-high
1.130
             PCTL_IMP_TLSatPct    03:0.5333-0.6886 vs 04:0.6886-high
1.040
             PCTL_InqFinanceCnt24 01:low -1 vs 04:5-high
0.599
             PCTL_InqFinanceCnt24 02:1-2 vs 04:5-high
0.404
             PCTL_InqFinanceCnt24 03:2-5 vs 04:5-high
0.807
             PCTL_TLDel3060Cnt24  02:0-1 vs 03:1-high
0.453
             PCTL_TLDel60Cnt24    02:0-1 vs 03:1-high
0.357
             PCTL_TLTimeFirst     01:low -107 vs 04:230-high
1.688
             PCTL_TLTimeFirst     02:107-152 vs 04:230-high
1.477
             PCTL_TLTimeFirst     03:152-230 vs 04:230-high
0.837
```

The improved model fit was also seen in the iteration plot, although the average profit of the selected model was still not as large as the original stepwise regression model.



Optimal Transformation

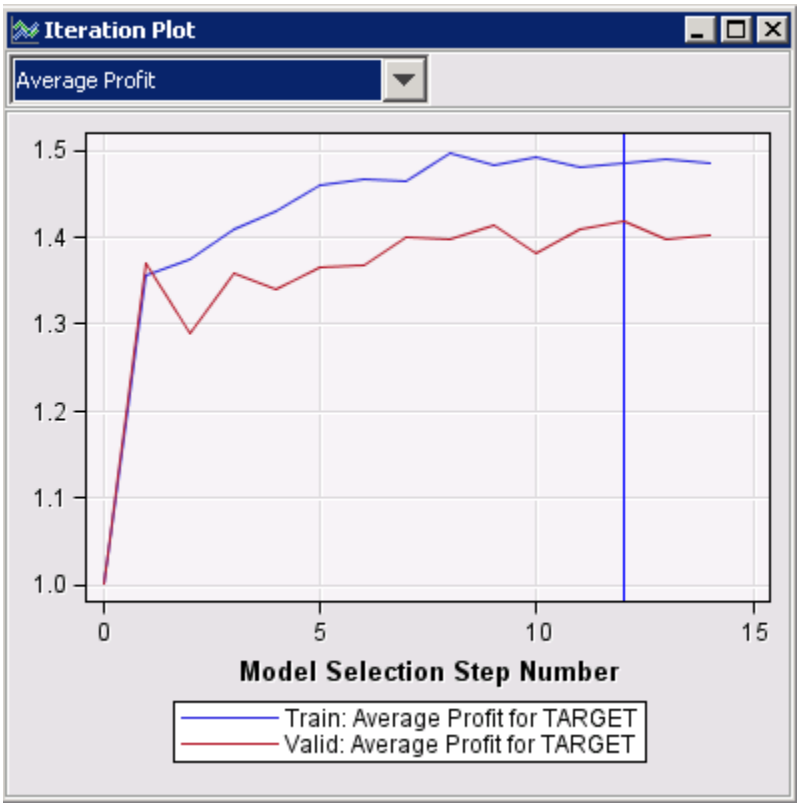
A final attempt on discretization was made using the optimistically named Optimal Discrete transformation. The final 18 degree-of-freedom model included 10 separate inputs (more than any other model). Contents of the Output window starting at line 1698 are shown below.

```
                              Odds Ratio Estimates


Point
       Effect
Estimate
       BanruptcyInd        0 vs 1
2.267
       OPT_IMP_TL75UtilCnt 01:low -1.5 vs 03:8.5-high
0.270
       OPT_IMP_TL75UtilCnt 02:1.5-8.5, MISSING vs 03:8.5-high
0.409
       OPT_IMP_TLBalHCPct  01:low -0.6706, MISSING vs 04:1.0213-high
0.090
       OPT_IMP_TLBalHCPct  02:0.6706-0.86785 vs 04:1.0213-high
0.155
       OPT_IMP_TLBalHCPct  03:0.86785-1.0213 vs 04:1.0213-high
0.250
```

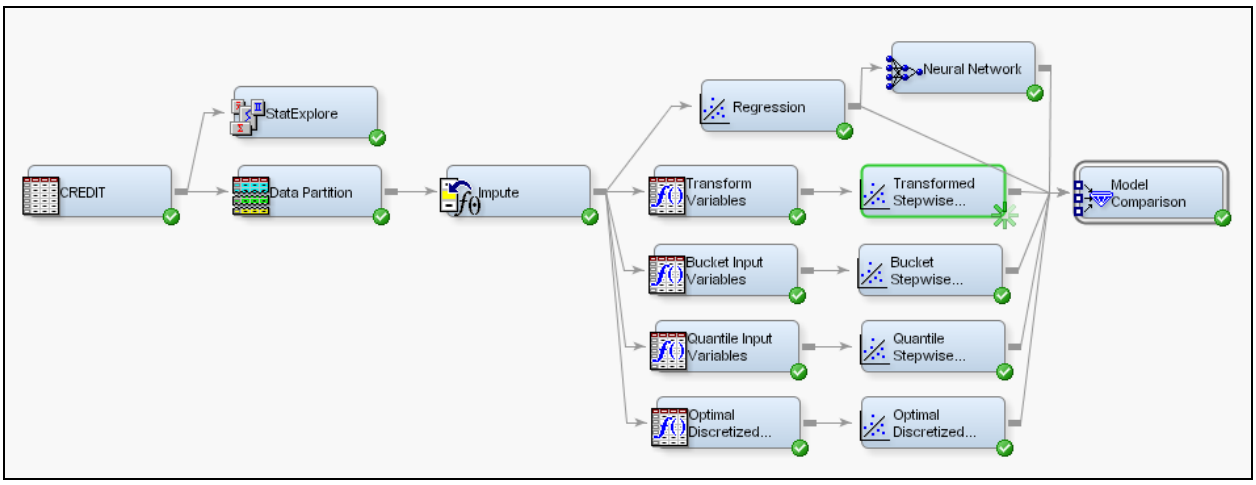
```
       OPT_IMP_TLSatPct    01:low -0.2094 vs 03:0.4655-high,
5.067
       OPT_IMP_TLSatPct    02:0.2094-0.4655 vs 03:0.4655-high,
1.970
       OPT_InqFinanceCnt24 01:low -2.5, MISSIN vs 03:7.5-high
0.353
       OPT_InqFinanceCnt24 02:2.5-7.5 vs 03:7.5-high
0.657
       OPT_TLDel3060Cnt24  01:low -1.5, MISSIN vs 02:1.5-high
0.499
       OPT_TLDel60Cnt      01:low -0.5, MISSIN vs 03:14.5-high
0.084
       OPT_TLDel60Cnt      02:0.5-14.5 vs 03:14.5-high
0.074
       OPT_TLDel60Cnt24    01:low -0.5, MISSIN vs 03:5.5-high
0.327
       OPT_TLDel60Cnt24    02:0.5-5.5 vs 03:5.5-high
0.882
       OPT_TLTimeFirst     01:low -154.5, MISSING vs 02:154.5-high
1.926
       TLOpenPct
3.337
```

The validation average profit was still slightly smaller than the original model. A substantial difference in profit between the training and validation data was also observed. Such a difference was suggestive of overfitting by the model.

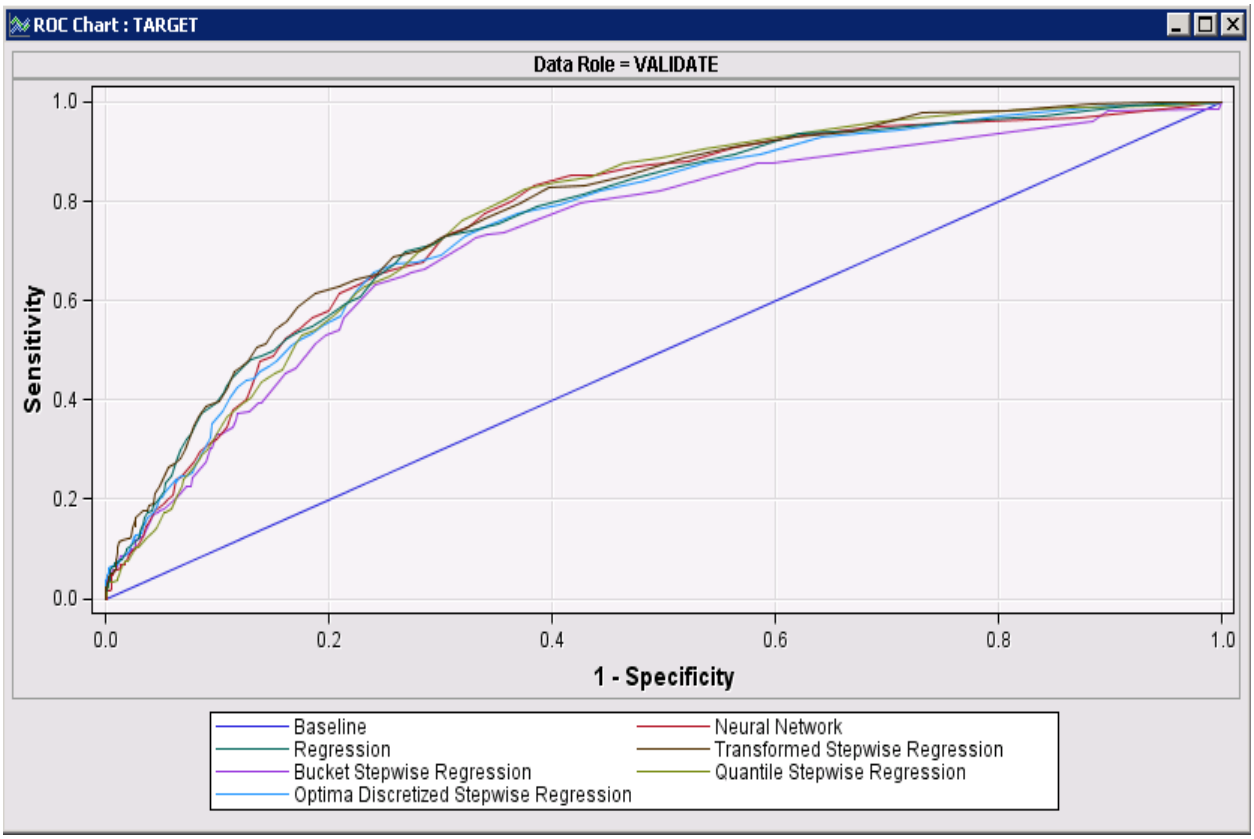


Assessing the Prediction Models

The collection of models was assessed using the Model Comparison node.



The ROC chart shows a jumble of models with no clear winner.



The Fit Statistics table from the Output window is shown below.

Data Role=Valid						
Statistics	Reg	Neural	Reg5	Reg2	Reg4	Reg3
Valid: Kolmogorov-Smirnov Statistic	0.43	0.46	0.42	0.44	0.45	0.39
Valid: Average Profit for TARGET	1.43	1.42	1.42	1.42	1.41	1.38
Valid: Average Squared Error	0.12	0.12	0.12	0.12	0.12	0.13
Valid: Roc Index	0.77	0.77	0.76	0.78	0.77	0.73
Valid: Average Error Function	0.38	0.39	0.40	0.38	0.39	0.43
Valid: Percent Capture Response	14.40	12.00	11.60	14.40	12.64	9.60
Valid: Divisor for VASE	3000.00	3000.00	3000.00	3000.00	3000.00	3000.00
Valid: Error Function	1152.26	1168.64	1186.46	1131.42	1158.23	1282.59
Valid: Gain	180.00	152.00	148.00	192.00	144.89	124.00
Valid: Gini Coefficient	0.54	0.54	0.53	0.56	0.54	0.47
Valid: Bin-Based Two-Way Kolmogorov-Smirnov Statistic	0.43	0.44	0.41	0.44	0.45	0.39
Valid: Lift	2.88	2.40	2.32	2.88	2.53	1.92
Valid: Maximum Absolute Error	0.97	0.99	1.00	0.98	0.99	1.00
Valid: Misclassification Rate	0.17	0.17	0.17	0.17	0.17	0.17
Valid: Mean Square Error	0.12	0.12	0.12	0.12	0.12	0.13
Valid: Sum of Frequencies	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00
Valid: Total Profit for TARGET	2143.03	2131.02	2127.45	2127.44	2121.42	2072.25
Valid: Root Average Squared Error	0.35	0.35	0.35	0.34	0.35	0.36
Valid: Percent Response	48.00	40.00	38.67	48.00	42.13	32.00
Valid: Root Mean Square Error	0.35	0.35	0.35	0.34	0.35	0.36
Valid: Sum of Square Errors	359.70	367.22	371.58	352.69	366.76	381.44
Valid: Sum of Case Weights Times Freq	3000.00	3000.00	3000.00	3000.00	3000.00	3000.00

The best model, as measured by average profit, was the original regression. The neural network had the highest KS statistic. The log-transformed regression, Reg2, had the highest ROC-index.

If the purpose of a credit risk model is to order the cases, then Reg2, the transformed regression, had the highest rank decision statistic, the ROC index.

In short, the best model for deployment was as much a matter of taste as of statistical performance. The relatively small validation data set used to compare the models did not produce a clear winner.

In the end, the model selected for deployment was the original stepwise regression, because it offered consistently good performance across multiple assessment measures.

A.4 Enrollment Management Case Study

Case Study Description

In the fall of 2004, the administration of a large private university requested that the Office of Enrollment Management and the Office of Institutional Research work together to identify prospective students who would most likely enroll as new freshmen in the Fall 2005 semester. The administration stated several goals for this project:

- increase new freshman enrollment
- increase diversity
- increase SAT scores of entering students

Historically, inquiries numbered about 90,000+ students, and the university enrolled from 2400 to 2800 new freshmen each Fall semester.

The diagram containing this analysis is stored as an XML file on the course data disk. You can open this file by right-clicking **Diagrams** ⇨ **Import Diagram from XML** in SAS Enterprise Miner. All nodes in the opened file, except the data node, contain the property settings outlined in this case study. If you want to run the diagram, you need to re-create the case study data set using the metadata settings indicated below.

Case Study Training Data

Name	Model Role	Measurement Level	Description
ACADEMIC_INTEREST_1	Rejected	Nominal	Primary academic interest code
ACADEMIC_INTEREST_2	Rejected	Nominal	Secondary academic interest code
CAMPUS_VISIT	Input	Nominal	Campus visit code
CONTACT_CODE1	Rejected	Nominal	First contact code
CONTACT_DATE1	Rejected	Nominal	First contact date
ETHNICITY	Rejected	Nominal	Ethnicity
ENROLL	Target	Binary	1=Enrolled F2004, 0=Not enrolled F2004
IRSCHOOL	Rejected	Nominal	High school code
INSTATE	Input	Binary	1=In state, 0=Out of state
LEVEL_YEAR	Rejected	Unary	Student academic level
REFERRAL_CNTCTS	Input	Ordinal	Referral contact count
SELF_INIT_CNTCTS	Input	Interval	Self initiated contact count
SOLICITED_CNTCTS	Input	Ordinal	Solicited contact count
TERRITORY	Input	Nominal	Recruitment area
TOTAL_CONTACTS	Input	Interval	Total contact count
TRAVEL_INIT_CNTCTS	Input	Ordinal	Travel initiated contact count
AVG_INCOME	Input	Interval	Commercial HH income estimate
DISTANCE	Input	Interval	Distance from university
HSCRAT	Input	Interval	5-year high school enrollment rate
INIT_SPAN	Input	Interval	Time from first contact to enrollment date
INT1RAT	Input	Interval	5-year primary interest code rate
INT2RAT	Input	Interval	5-year secondary interest code rate
INTEREST	Input	Ordinal	Number of indicated extracurricular interests
MAILQ	Input	Ordinal	Mail qualifying score (1=very interested)

(Continued on the next page.)

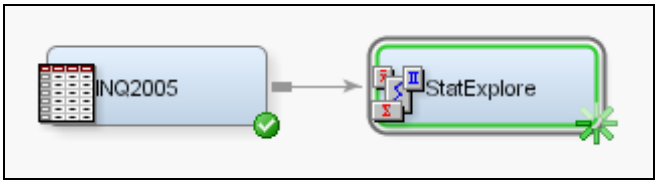
PREMIERE	Input	Binary	1=Attended campus recruitment event, 0=Did not
SATSCORE	Rejected	Interval	SAT (original) score
SEX	Rejected	Binary	Sex
STUEMAIL	Input	Binary	1=Have e-mail address, 0=Do not
TELECQ	Rejected	Ordinal	Telecounciling qualifying score (1=very interested)

The Office of Institutional Research assumed the task of building a predictive model, and the Office of Enrollment Management served as consultant to the project. The Office of Institutional Research built and maintained a data warehouse that contained information about enrollment for the past six years. It was decided that inquiries for Fall 2004 would be used to build the model to help shape the Fall 2005 freshman class. The data set **Inq2005** was built over a period of a several months in consultation with Enrollment Management. The data set included variables that could be classified as demographic, financial, number of correspondences, student interests, and campus visits. Many variables were created using historical data and trends. For example, high school code was replaced by the percentage of inquirers from that high school over the past five years who enrolled. The resulting data set included over 90,000 observations and over 50 variables. For this case study, the number of variables was reduced. The data set **Inq2005** is in the AAEM library, and the variables are described in the table above. Some of the variables were automatically rejected based on the number of missing values.

The nominal variables **ACADEMIC_INTEREST_1**, **ACADEMIC_INTEREST_2**, and **IRSCHOOL** were rejected because they were replaced by the interval variables **INT1RAT**, **INT2RAT**, and **HSCRAT**, respectively. For example, academic interest codes 1 and 2 were replaced by the percentage of inquirers over the past five years who indicated those interest codes and then enrolled. The variable **IRSCHOOL** is the high school code of the student, and it was replaced by the percentage of inquirers from that high school over the last five years who enrolled. The variables **ETHNICITY** and **SEX** were rejected because they cannot be used in admission decisions. Several variables count the various types of contacts the university has with the students.

Accessing and Assaying the Data

A SAS Enterprise Miner data source was defined using the metadata settings indicated above. The StatExplore node was used to provide preliminary statistics on the input variables.



The following is extracted from the StatExplore node's Results window:

Class Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	CAMPUS_VISIT	INPUT	3	0	0	96.61	1	3.31
TRAIN	Instate	INPUT	2	0	Y	62.04	N	37.96
TRAIN	REFERRAL_CNTCTS	INPUT	6	0	0	96.46	1	3.21
TRAIN	SOLICITED_CNTCTS	INPUT	8	0	0	52.45	1	41.60
TRAIN	TERRITORY	INPUT	12	1	2	15.98	5	15.34
TRAIN	TRAVEL_INIT_CNTCTS	INPUT	7	0	0	67.00	1	29.90
TRAIN	interest	INPUT	4	0	0	95.01	1	4.62
TRAIN	mailq	INPUT	5	0	5	69.33	2	12.80
TRAIN	premiere	INPUT	2	0	0	97.11	1	2.89
TRAIN	stuemail	INPUT	2	0	0	51.01	1	48.99
TRAIN	Enroll	TARGET	2	0	0	96.86	1	3.14

The class input variables are listed first. Notice that most of the count variables have a high percent of 0s.

Distribution of Class Target and Segment Variables
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Level	Frequency Count	Percent
TRAIN	Enroll	TARGET	0	88614	96.8650
TRAIN	Enroll	TARGET	1	2868	3.1350

Next is the target distribution. Only 3.1 % of the target values are 1s, making a 1 a rare event. Standard practice in this situation is to separately sample the 0s and 1s. The Sample tool, used below, enables you to create a stratified sample in SAS Enterprise Miner.

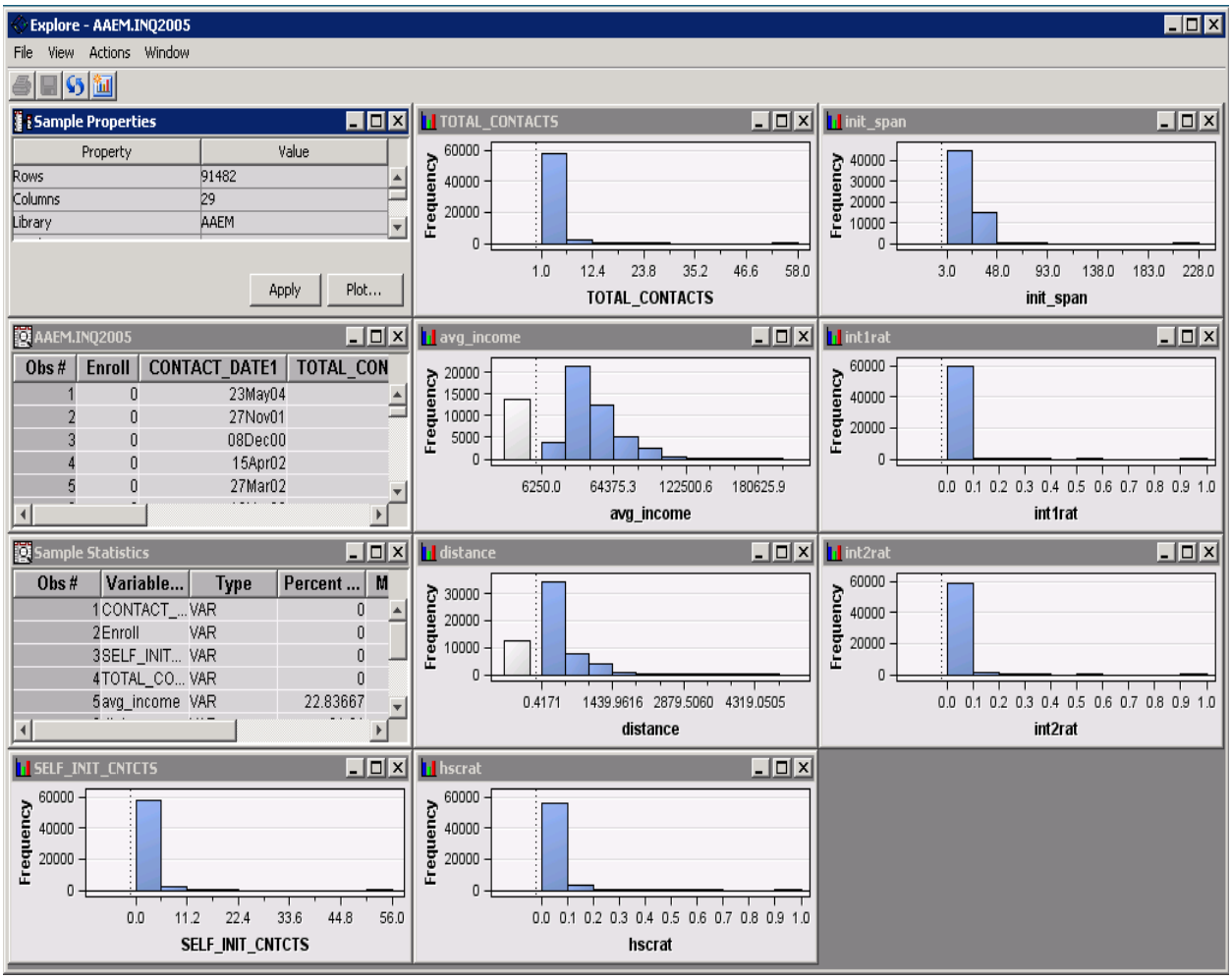
Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
SELF_INIT_CNTCTS	INPUT	1.214119	1.666529	91482	0	0	1	56	2.916263	21.50072
TOTAL_CONTACTS	INPUT	2.166098	1.852537	91482	0	1	2	58	3.062389	19.60427
avg_income	INPUT	47315.33	20608.89	70553	20929	4940	42324	200001	1.258231	1.874903
distance	INPUT	380.4276	397.9788	72014	19468	0.417124	183.5467	4798.899	2.276541	9.369703
hscrat	INPUT	0.037652	0.057399	91482	0	0	0.033333	1	7.021978	93.31547
init_span	INPUT	19.68616	8.722109	91482	0	-216	19	228	0.758461	10.43657
intlrat	INPUT	0.037091	0.024026	91482	0	0	0.042105	1	3.496845	74.08503
int2rat	INPUT	0.042896	0.025244	91482	0	0	0.05667	1	3.215683	56.32374

Finally, interval variable summary statistics are presented. Notice that **avg_income** and **distance** have missing values.

The Explore window was used to study the distribution of the interval variables.

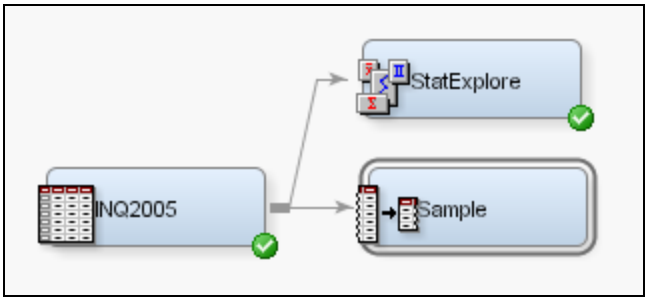


The apparent skewness of all inputs suggests that some transformations might be needed for regression models.

Creating a Training Sample

Cases from each target level were separately sampled. All cases with the primary outcome were selected. For each primary outcome case, seven secondary outcome cases were selected. This created a training sample with a 12.5% overall enrollment rate.

The Sample tool was used to create a training sample for subsequent modeling.



To create the sample as described, the following modifications were made to the Sample node's properties panel:

1. Type **100** as the Percentage value (in the Size property group).
2. Select **Criterion** ⇨ **Level Based** (in the Stratified property group).
3. Type **12.5** as the Sample Proportion value (in the Level Based Options property group).

Train	
Variables	...
Output Type	Data
Sample Method	Default
Random Seed	12345
Size	
Type	Percentage
Observations	.
Percentage	100.0
Alpha	0.01
PValue	0.01
Cluster Method	Random
Stratified	
Criterion	Level Based
Ignore Small Strata	No
Minimum Strata Size	5
Level Based Options	
Level Selection	Event
Level Proportion	100.0
Sample Proportion	12.5
Oversampling	
Adjust Frequency	No
Based on Count	No
Exclude Missing Levels	No

The Sample node Results window shows all primary outcome cases that are selected and a sufficient number of secondary outcome cases that are selected to achieve the 12.5% primary outcome proportion.

Summary Statistics for Class Targets
(maximum 500 observations printed)

Data=DATA

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
Enroll	0	0	88614	96.8650	
Enroll	1	1	2868	3.1350	

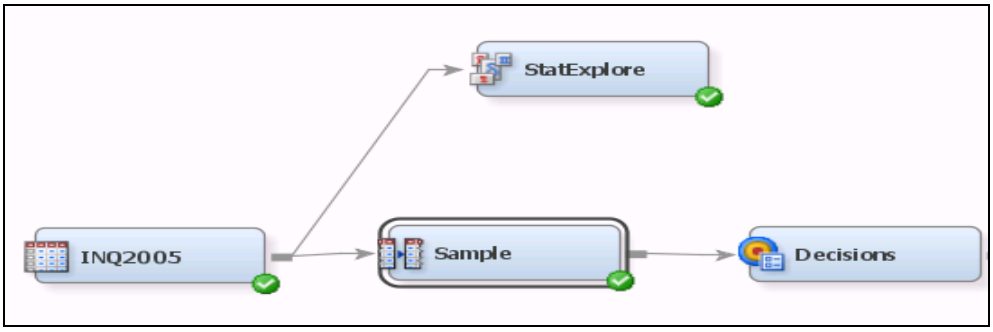
Data=SAMPLE

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
Enroll	0	0	20076	87.5	
Enroll	1	1	2868	12.5	

Configuring Decision Processing

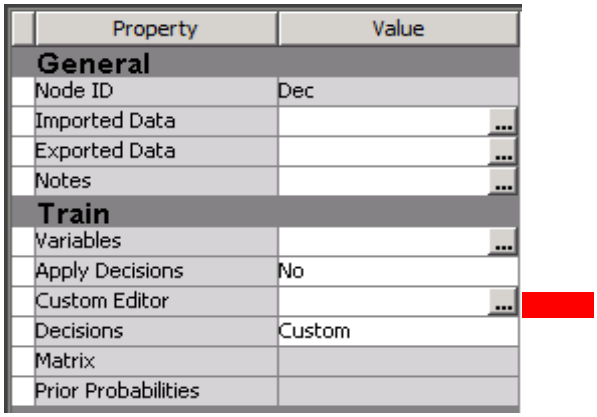
The primary purpose of the predictions was decision optimization and, secondarily, ranking. An applicant was considered a good candidate if his or her probability of enrollment was higher than average.

Because of the Sample node, decision information consistent with the above objectives could not be entered in the data source node. To incorporate decision information, the Decisions tool was incorporated in the analysis.

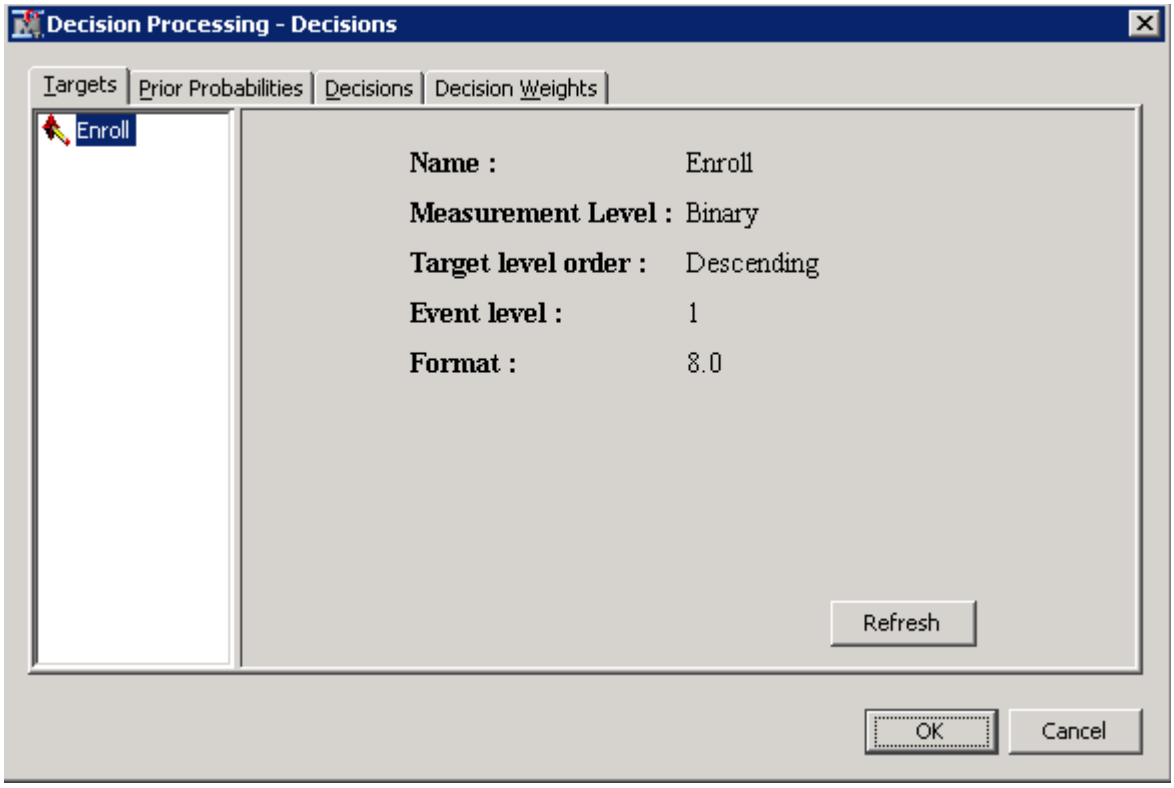


These steps were followed to configure the Decisions node:

1. In the Properties panel of the Decision node, set **Decisions** to Custom. Then select **Custom Editor** ⇨ [...].

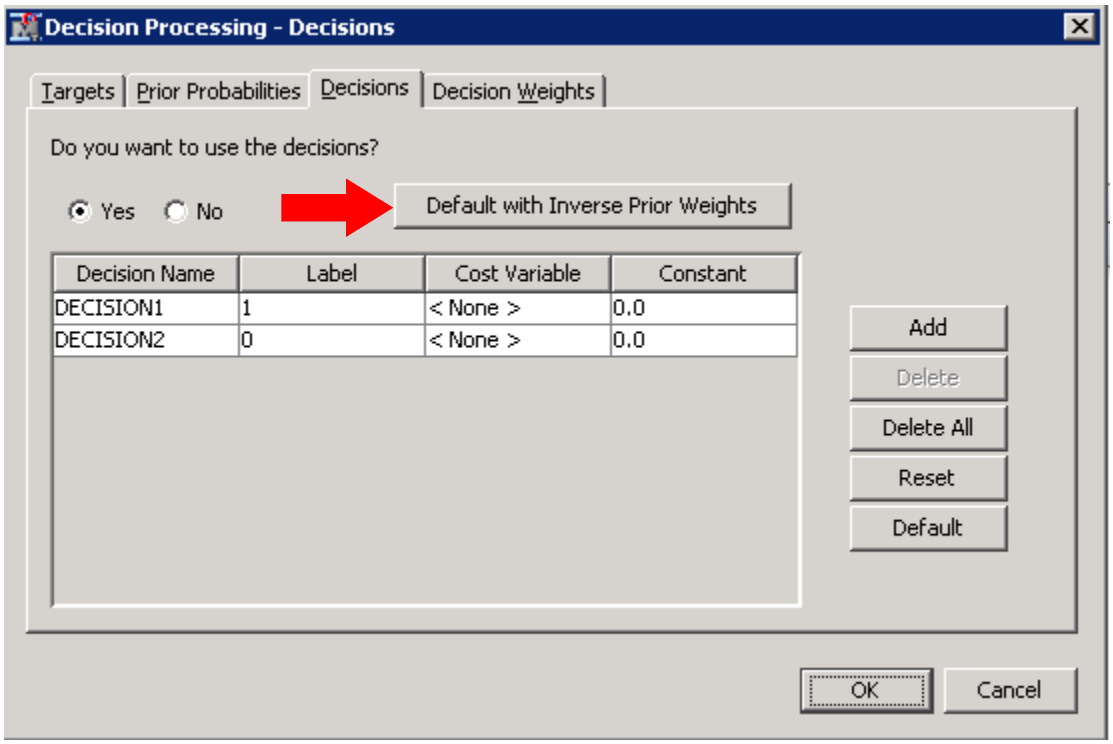


After the analysis path is updated, the Decision window appears.

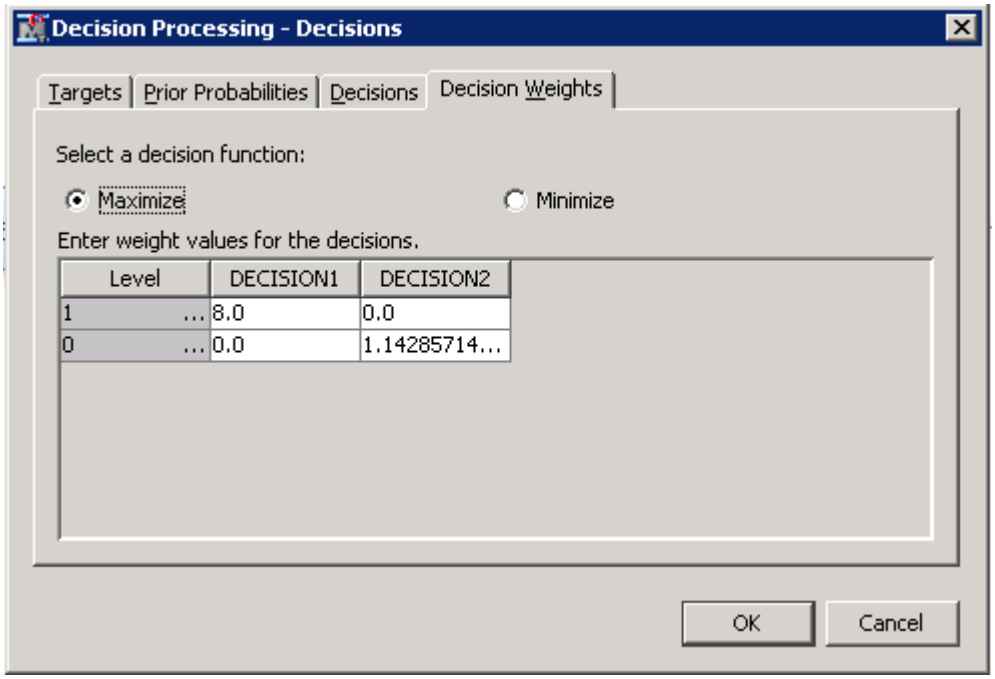


2. Select the **Decisions** tab.

3. Select **Default with Inverse Prior Weights**.



4. Select the **Decision Weights** tab.

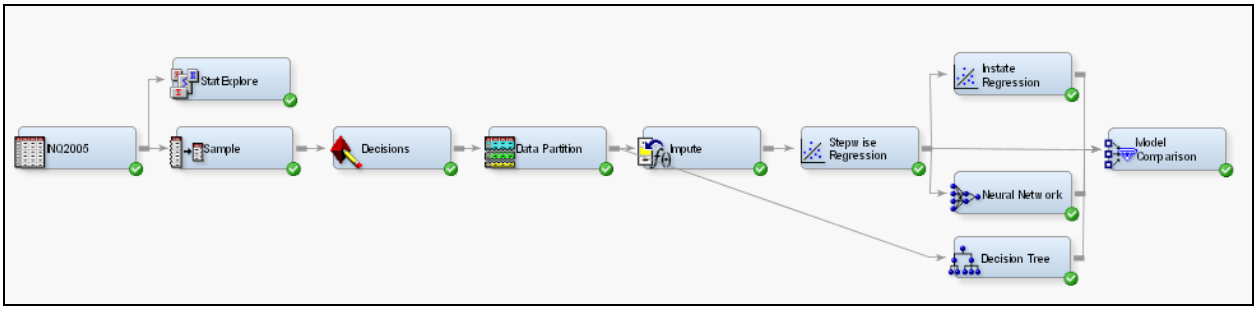


The nonzero values used in the decision matrix are the inverse of the prior probabilities (1/.125=8. and 1/0.875=1.142857). Such a decision matrix, sometimes referred to as the *central decision rule*, forces a primary decision when the estimated primary outcome probability for a case exceeds the primary outcome prior probability (0.125 in this case).

Creating Prediction Models (All Cases)

Two rounds of predictive modeling were performed. In the first round, all cases were considered for model building. From the Decision node, partitioning, imputation, modeling, and assessment were performed. The completed analysis appears as shown.

If the Stepwise Regression model is not connected to the Model Comparison node, you might have to first delete the ***connections*** for the Instate Regression and Neural Network nodes to the Model Comparison node. Then connect the Stepwise Regression node, Neural Network node, and Regression nodes – in that order – to the Model Comparison node.



- The Data Partition node used 60% for training and 40% for validation.
- The Impute node used the Tree method for both class and interval variables. Unique missing indicator variables were also selected and used as inputs.
- The stepwise regression model was used as a variable selection method for the Neural Network and second Regression nodes.
- The Regression node labeled **Instate Regression** included the variables from the Stepwise Regression node and the variable **Instate**. It was felt that prospective students behave differently based on whether they are in state or out of state.

In this implementation of the case study, the Stepwise Regression node selected three inputs: high school, self-initiated contact count, and student e-mail indicator. The model output is shown below.

Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq	Standardized Estimate	Exp(Est)
INTERCEPT		1	-12.1422	18.9832	0.41	0.5224		0.000
SELF_INIT_CNTCTS		1	0.6895	0.0203	1156.19	<.0001	0.8773	1.993
HSCRAT		1	16.4261	0.8108	410.46	<.0001	0.7506	999.000
STUEMAIL	0	1	-7.7776	18.9824	0.17	0.6820		0.000

Odds Ratio Estimates

Effect		Point Estimate
SELF_INIT_CNTCTS		1.993
HSCRAT		999.000
STUEMAIL	0 VS 1	<0.001

The unusual odds ratio estimates for **HSCRAT** and **STUEMAIL** result from an extremely strong association in those inputs. For example, certain high schools had all applicants or no applicants enroll. Likewise, very few students enrolled who did not provide an e-mail address.

Adding the **INSTATE** input in the Instate Regression model changed the significance of inputs selected by the stepwise regression model. The input **STUEMAIL** is no longer statistically significant after including the **INSTATE** input.

Analysis of Maximum Likelihood Estimates

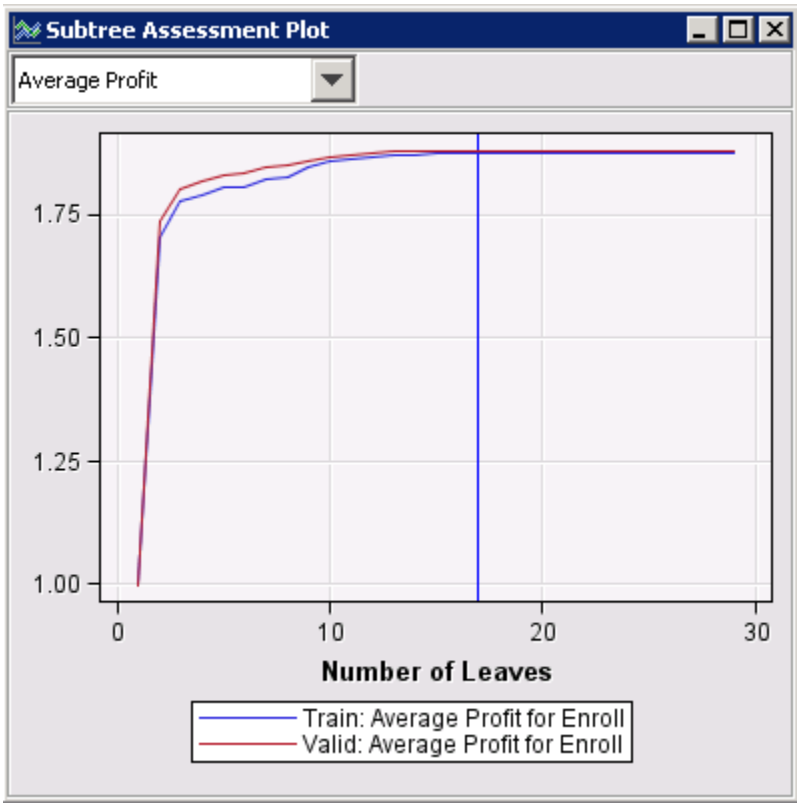
Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq	Standardized Estimate	Exp(Est)
INTERCEPT		1	-12.0541	16.7449	0.52	0.4716		0.000
INSTATE	N	1	-0.4145	0.0577	51.67	<.0001		0.661
SELF_INIT_CNTCTS		1	0.6889	0.0196	1233.22	<.0001	0.8231	1.992
HSCRAT		1	16.2327	0.7553	461.95	<.0001	0.7142	999.000
STUEMAIL	0	1	-7.3528	16.7443	0.19	0.6606		0.001

Odds Ratio Estimates

Effect		Point Estimate
INSTATE	N VS Y	0.437
SELF_INIT_CNTCTS		1.992
HSCRAT		999.000
STUEMAIL	0 VS 1	<0.001

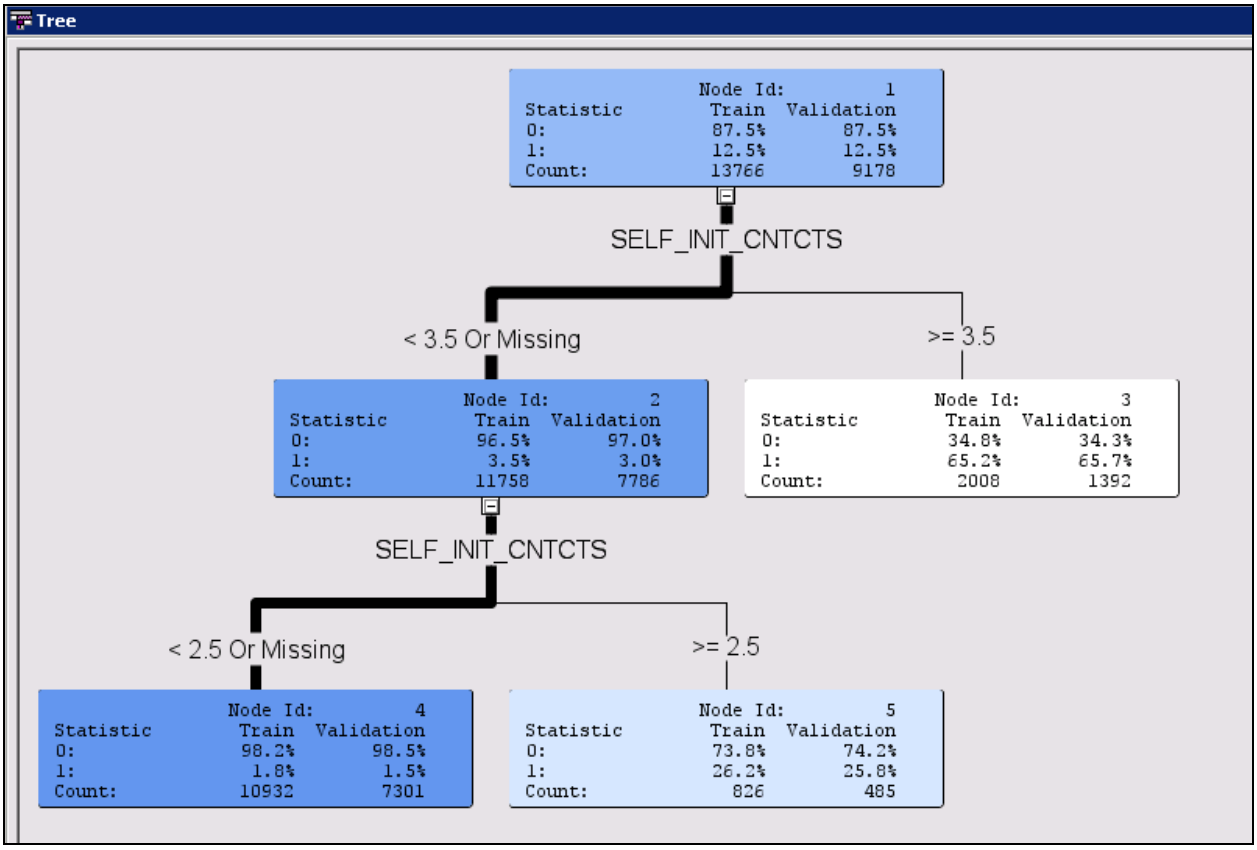
A slight increase in validation profit (the criterion used to tune models) was found using the neural network model.

The tree provides insight into the strength of model fit. The Subtree Assessment plot shows the highest profit having 17 leaves. Most of the predictive performance, however, is provided by the initial splits.



A simpler tree is scrutinized to aid in interpretation.

The tree model was rerun with properties changed as follows to produce a tree with three leaves: Method=**N**, Number of Leaves=**3**.



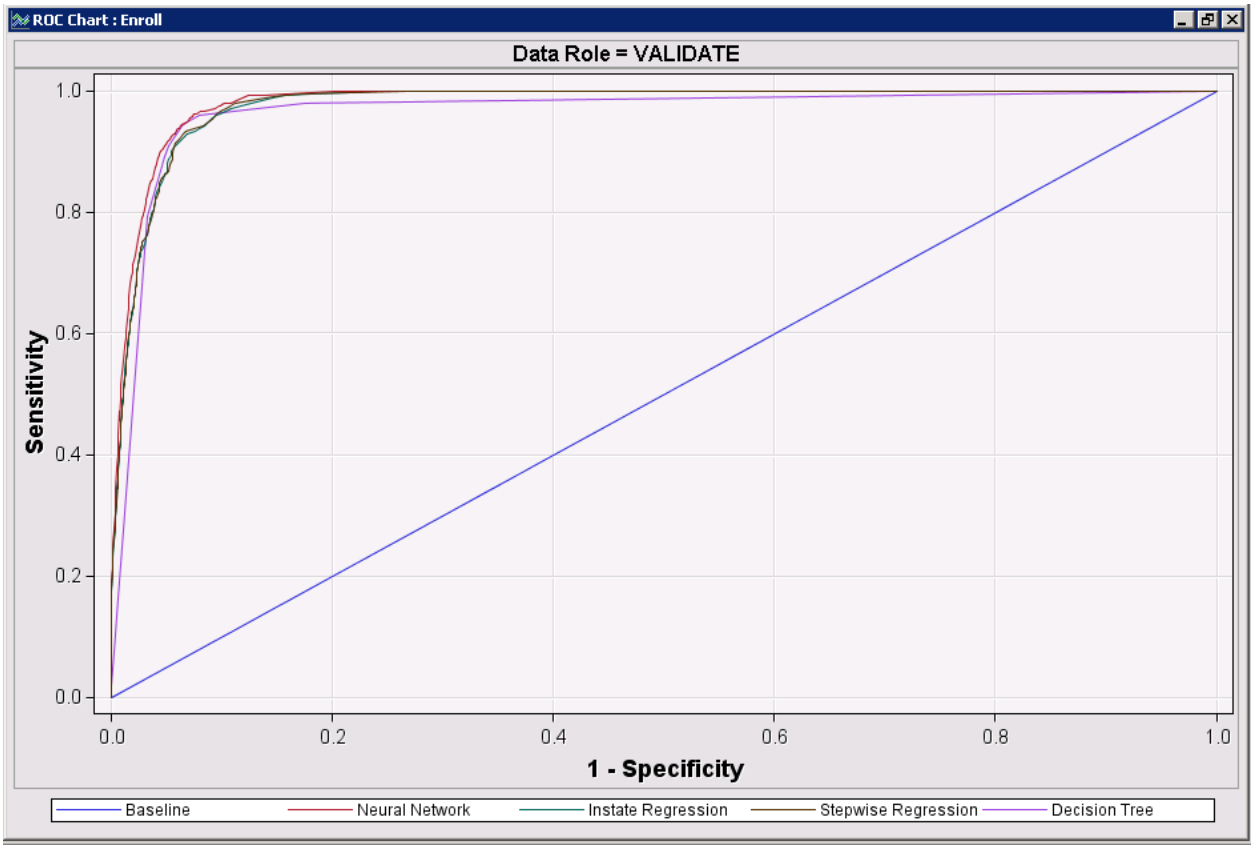
Students with three or fewer self-initiated contacts rarely enrolled (as seen in the left leaf of the first split). Enrollment was even rarer for students with two or fewer self-initiated contacts (as seen in the left leaf of the second split). Notice that the primary target percentage is rounded down. Also notice that most of the secondary target cases can be found in the lower left leaf.

The decision tree results shown in the rest of this case study are generated by the original, 17-leaf tree.

Assessing the Prediction Models

Model performance was compared in the Model Comparison node.

If the Stepwise Regression model does not appear in the ROC chart, it might not be connected to the Model Comparison node. You might have to first delete the ***connections*** for the Instate Regression and Neural Network nodes to the Model Comparison node. Connect the Stepwise Regression node, Neural Network node, and Regression nodes – in that order – to the Model Comparison node and re-run the Model Comparison node to make all models visible.



The validation ROC chart showed an extremely good performance for all models. The neural model seemed to have a slight edge over the other models. This was mirrored in the Fit Statistics table (abstracted below to show only the validation performance).

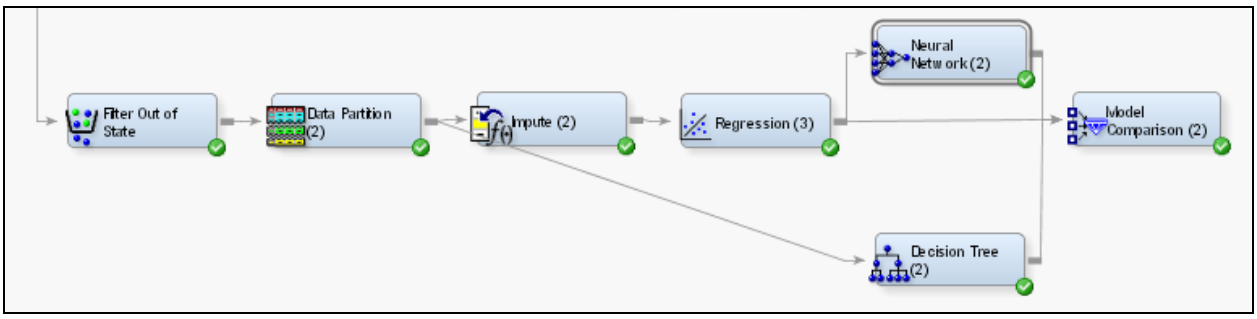
Data Role=Valid

Statistics	Neural	Tree	Reg	Reg2
Valid: Kolmogorov-Smirnov Statistic	0.89	0.88	0.87	0.86
Valid: Average Profit for Enroll	1.88	1.88	1.87	1.86
Valid: Average Squared Error	0.04	0.04	0.04	0.04
Valid: Roc Index	0.98	0.96	0.98	0.98
Valid: Average Error Function	0.11	.	0.14	0.14
Valid: Percent Capture Response	30.94	30.95	29.72	29.55
Valid: Divisor for VASE	18356.00	18356.00	18356.00	18356.00
Valid: Error Function	2097.03	.	2521.91	2486.75
Valid: Gain	576.37	519.90	552.83	552.83
Valid: Gini Coefficient	0.96	0.93	0.95	0.95
Valid: Bin-Based Two-Way Kolmogorov-Smirnov Statistic	0.88	0.87	0.86	0.86
Valid: Lift	6.19	6.19	5.94	5.91
Valid: Maximum Absolute Error	1.00	1.00	1.00	1.00
Valid: Misclassification Rate	0.05	0.05	0.06	0.06
Valid: Mean Square Error	0.04	.	0.04	0.04
Valid: Sum of Frequencies	9178.00	9178.00	9178.00	9178.00
Valid: Total Profit for Enroll	17285.71	17256.00	17122.29	17099.43
Valid: Root Average Squared Error	0.19	0.20	0.20	0.20
Valid: Percent Response	77.34	77.36	74.29	73.85
Valid: Root Mean Square Error	0.19	.	0.20	0.20
Valid: Sum of Square Errors	657.78	735.99	752.78	754.37
Valid: Sum of Case Weights Times Freq	18356.00	18356.00	18356.00	18356.00
Valid: Number of Wrong Classifications.	463.00	.	.	.

It should be noted that a ROC Index of 0.98 needed careful consideration because it suggested a near-perfect separation of the primary and secondary outcomes. The decision tree model provides some insight into this apparently outstanding model fit. Self-initiated contacts are critical to enrollment. Fewer than three self-initiated contacts almost guarantees non-enrollment.

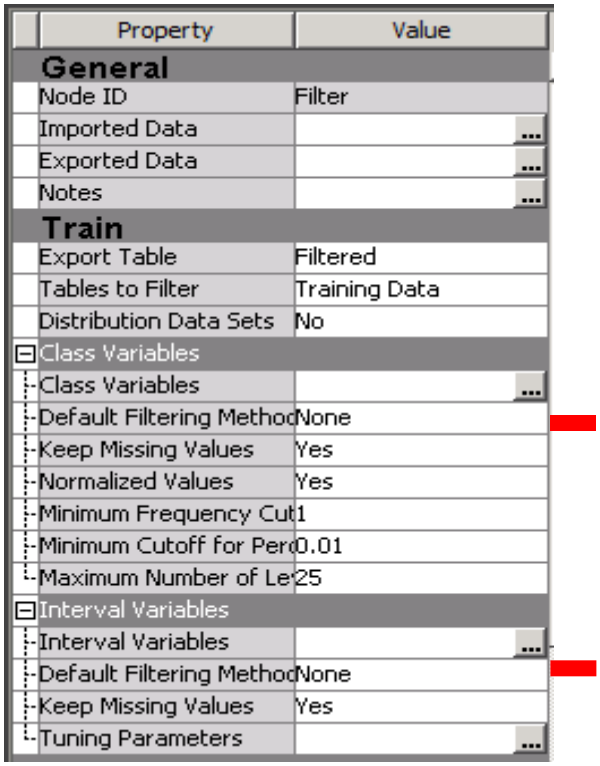
Creating Prediction Models (Instate-Only Cases)

A second round of analysis was performed on instate-only cases. The analysis sample was reduced using the Filter node. The Filter node was attached to the Decisions node, as shown below.



The following configuration steps were applied:

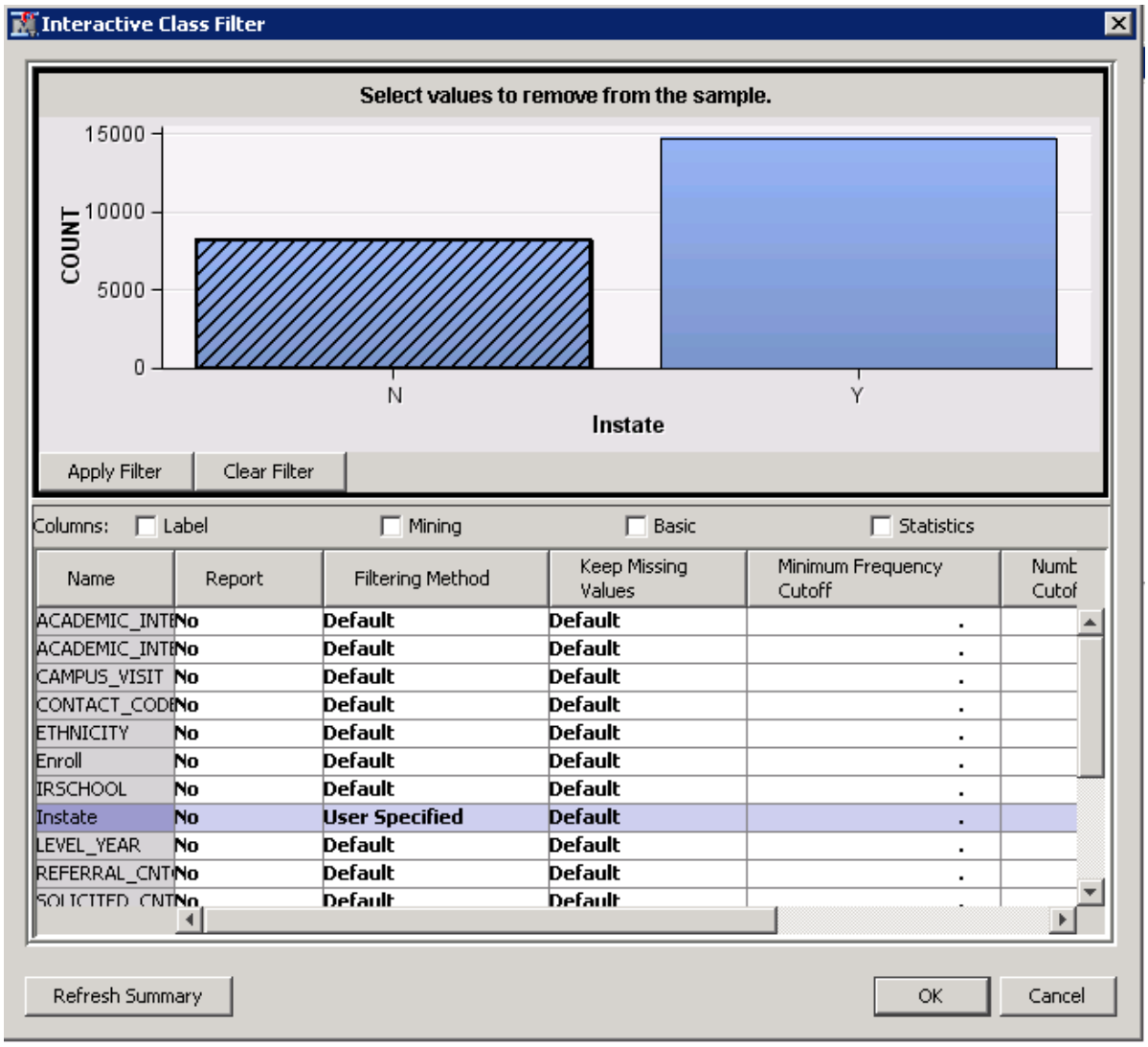
1. In the Filter Out of State node, select **Default Filtering Method** ⇨ **None** for both the class and interval variables.



2. Select **Class Variables** ⇨ ... . After the path is updated, the Interactive Class Filter window appears.

3. Select **Generate Summary** and then select **Yes** to generate summary statistics.

4. Select **Instate**. The Interactive Class Filter window is updated to show the distribution of the **Instate** input.



5. Select the **N** bar and select **Apply Filter**.

6. Select **OK** to close the Interactive Class Filter window.

7. Run the Filter node and view the results.

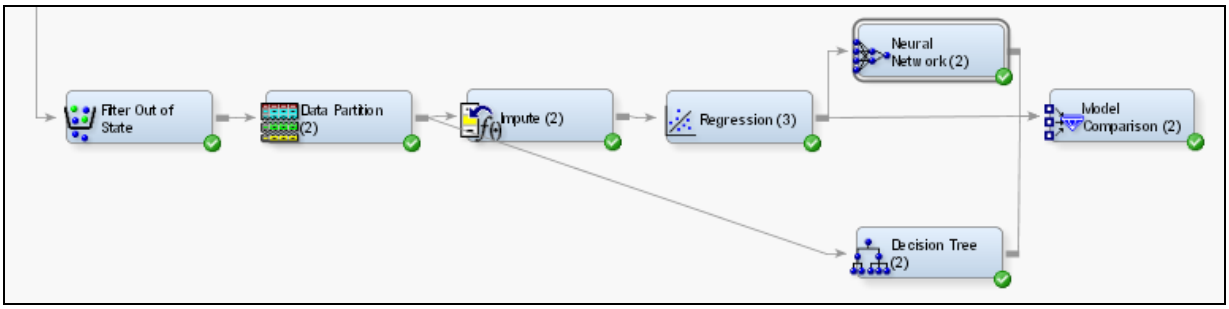
Excluded Class Values
(maximum 500 observations printed)

Variable	Role	Level	Train Count	Train Percent	Label	Filter Method	Keep Missing Values
Instate	INPUT	N	8200	35.7392		MANUAL	

All out-of-state cases were filtered from the analysis.

After filtering, an analysis similar to the above was conducted with stepwise regression, neural network, and decision tree models.

The partial diagram (after the Filter node) is shown below:

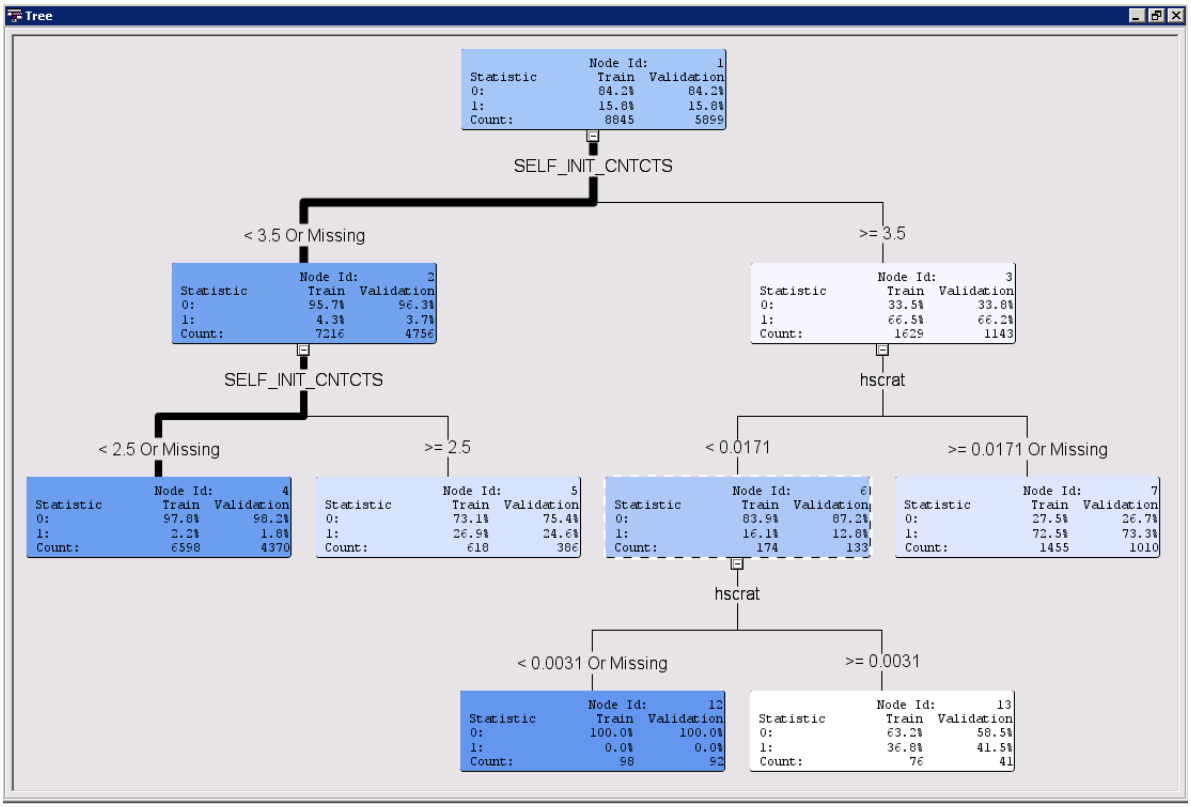


As for the models in this subset analysis, the **Instate** Stepwise Regression model selects two of the same inputs found in the first round of modeling, **SELF_INIT_CNTCTS** and **STUEMAIL**.

Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq	Standardized Estimate	Exp(Est)
Intercept		1	-10.1372	20.8246	0.24	0.6264		0.000
SELF_INIT_CNTCTS		1	0.7188	0.0210	1174.00	<.0001	0.9297	2.052
stuemail	0	1	-6.8602	20.8245	0.11	0.7418		0.001

The **Instate** decision tree showed a structure similar to the decision tree model from the first round. The tree with the highest validation profit possessed 20 leaves. The best five-leaf tree, whose validation profit is 97% of the selected tree, is shown below.

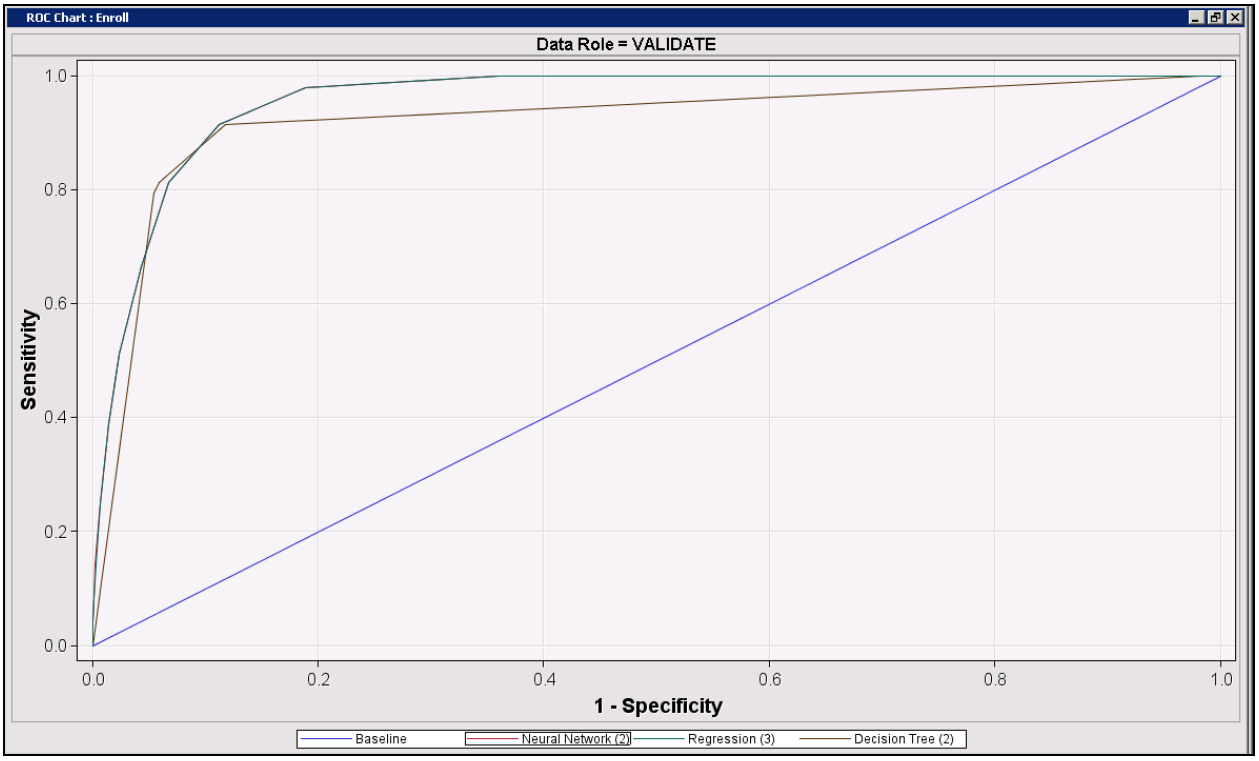


Again, much of the performance of the model is due to a low self-initiated contacts count.

Assessing Prediction Models (Instate-Only Cases)

As before, model performance was gauged in the Model Comparison node.

The ROC chart showed no clearly superior model, although all models had rather exceptional performance.



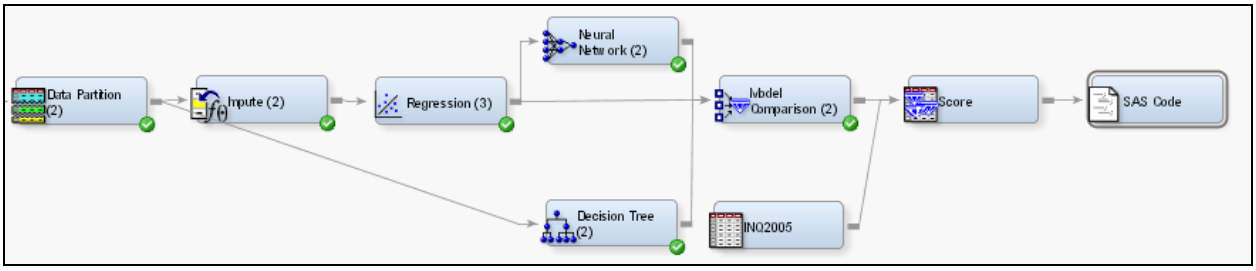
The Fit Statistics table of the Output window showed a slight edge over the tree model in misclassification rate. The validation ROC index and validation average profit favored the Stepwise Regression and Neural Network models. Again, it should be noted that these were unusually high model performance statistics.

Data Role=Valid

Statistics	Neural2	Reg3	Tree2
Valid: Kolmogorov-Smirnov Statistic	0.80	0.80	0.80
Valid: Average Profit	2.02	2.02	2.00
Valid: Average Squared Error	0.06	0.06	0.06
Valid: Roc Index	0.96	0.96	0.92
Valid: Average Error Function	0.19	0.20	0.21
Valid: Bin-Based Two-Way Kolmogorov-Smirnov Probability Cutoff	.	.	.
Valid: Cumulative Percent Captured Response	50.69	50.69	46.38
Valid: Percent Captured Response	23.41	23.41	23.19
Valid: Frequency of Classified Cases	5899.00	5899.00	5899.00
Valid: Divisor for ASE	11798.00	11798.00	11798.00
Valid: Error Function	2194.05	2330.78	2462.62
Valid: Gain	406.85	406.85	363.74
Valid: Gini Coefficient	0.91	0.91	0.84
Valid: Bin-Based Two-Way Kolmogorov-Smirnov Statistic	0.00	0.00	0.00
Valid: Kolmogorov-Smirnov Probability Cutoff	0.14	0.14	0.03
Valid: Cumulative Lift	5.07	5.07	4.64
Valid: Lift	4.68	4.68	4.64
Valid: Maximum Absolute Error	0.97	1.00	0.98
Valid: Misclassification Rate	0.09	0.09	0.08
Valid: Sum of Frequencies	5899.00	5899.00	5899.00
Valid: Total Profit	11901.71	11901.71	11824.00
Valid: Root Average Squared Error	0.24	0.25	0.25
Valid: Cumulative Percent Response	80.08	80.08	73.27
Valid: Percent Response	73.97	73.97	73.27
Valid: Sum of Squared Errors	705.43	725.26	716.66
Valid: Number of Wrong Classifications	510.00	527.00	462.00

Deploying the Prediction Model

The Score node facilitated deployment of the prediction model, as shown in the diagram's final form.



The best (instate) model was selected by the **Instate** Model Comparison node and passed on to the Score node. Another **INQ2005** data source was assigned a role of Score and attached to the Score node. Columns from the scored **INQ2005** were then passed into the Office of Enrollment Management's data management system by the final SAS Code node.