Creating a SAS Score Code Module (Self Study)

The SAS Score Code module is opened by default when you open the Score node.

1. Open the Score node Results window.
2. Maximize the SAS Code window.

The SAS Code window shows the SAS DATA step code that is necessary to append predictions from the selected model (in this case, the regression model) to a Score data set. Each node in the process flow can contribute to the DATA step code. The following list describes some highlights of the generated SAS code:

- Go to line **20**. This code removes the spurious zero from the median income input.

```
20  *-------------------------------------------------------------*;
21  * TOOL: Extension Class;
22  * TYPE: MODIFY;
23  * NODE: Repl;
24  *-------------------------------------------------------------*;
25  * ;
26  * Variable: DemMedIncome ;
27  * ;
28  Label REP_DemMedIncome='Replacement: Median Income Region';
29  REP_DemMedIncome =DemMedIncome ;
30  if DemMedIncome  eq . then REP_DemMedIncome = . ;
31  else
32  if DemMedIncome <1  then REP_DemMedIncome  = . ;
```

- Go to line **46**. This code takes the log transformation of selected inputs. (Only a portion of this part of the Score code is shown below.)

```
46  *------------------------------------------------------------*;
47  * TRANSFORM: GiftAvg36 , log(GiftAvg36 + 1);
48  *------------------------------------------------------------*;
49  label LOG_GiftAvg36 = 'Transformed: Gift Amount Average 36 Months';
50  if GiftAvg36 eq . then LOG_GiftAvg36 = .;
51  else do;
52  if GiftAvg36 + 1 > 0 then LOG_GiftAvg36 = log(GiftAvg36 + 1);
53  else LOG_GiftAvg36 = .;
54  end;
55  *------------------------------------------------------------*;
56  * TRANSFORM: GiftAvgAll , log(GiftAvgAll + 1);
57  *------------------------------------------------------------*;
58  label LOG_GiftAvgAll = 'Transformed: Gift Amount Average All Months';
59  if GiftAvgAll eq . then LOG_GiftAvgAll = .;
60  else do;
61  if GiftAvgAll + 1 > 0 then LOG_GiftAvgAll = log(GiftAvgAll + 1);
62  else LOG_GiftAvgAll = .;
63  end;
64  *------------------------------------------------------------*;
65  * TRANSFORM: GiftAvgCard36 , log(GiftAvgCard36 + 1);
66  *------------------------------------------------------------*;
67  label LOG_GiftAvgCard36 = 'Transformed: Gift Amount Average Card 36 Months';
68  if GiftAvgCard36 eq . then LOG_GiftAvgCard36 = .;
69  else do;
70  if GiftAvgCard36 + 1 > 0 then LOG_GiftAvgCard36 = log(GiftAvgCard36 + 1);
71  else LOG_GiftAvgCard36 = .;
72  end;
73  *------------------------------------------------------------*;
74  * TRANSFORM: GiftAvgLast , log(GiftAvgLast + 1);
75  *------------------------------------------------------------*;
76  label LOG_GiftAvgLast = 'Transformed: Gift Amount Last';
77  if GiftAvgLast eq . then LOG_GiftAvgLast = .;
78  else do;
79  if GiftAvgLast + 1 > 0 then LOG_GiftAvgLast = log(GiftAvgLast + 1);
80  else LOG GiftAvgLast = .;
.
.
.
```

- Go to line **118**. This code replaces the levels of the **StatusCat96NK** input.

```
118  *------------------------------------------------------------*;
119  * TOOL: Extension Class;
120  * TYPE: MODIFY;
121  * NODE: Repl2;
122  *------------------------------------------------------------*;
123
124  * ;
125  * Defining New Variables;
126  * ;
127  Length REP_StatusCat96NK $5;
128  Label REP_StatusCat96NK='Replacement: Status Category 96NK';
129  REP_StatusCat96NK= StatusCat96NK;
130
131  * ;
132  * Replace Specific Class Levels ;
133  * ;
134  length _UFormat200 $200;
135  drop   _UFORMAT200;
136  _UFORMAT200 = " ";
137  * ;
138  * Variable: StatusCat96NK;
139  * ;
140  _UFORMAT200 = strip(StatusCat96NK);
141  if _UFORMAT200 =  "A" then
142  REP_StatusCat96NK="A";
143  else
144  if _UFORMAT200 =  "S" then
145  REP_StatusCat96NK="A";
146  else
147  if _UFORMAT200 =  "F" then
148  REP_StatusCat96NK="N";
149  else
150  if _UFORMAT200 =  "N" then
151  REP_StatusCat96NK="N";
152  else
153  if _UFORMAT200 =  "E" then
154  REP_StatusCat96NK="L";
155  else
156  if _UFORMAT200 =  "L" then
157  REP_StatusCat96NK="L";
```

- Go to line **158**. This code replaces missing values and creates missing value indicators.

```
158  *-------------------------------------------------------------*;
159  * TOOL: Imputation;
160  * TYPE: MODIFY;
161  * NODE: Impt;
162  *-------------------------------------------------------------*;
163  *;
164  *MEAN-MAX-MIN-MEDIAN-MIDRANGE AND ROBUST ESTIMATES;
165  *;
166  label IMP_DemAge = 'Imputed: Age';
167  IMP_DemAge = DemAge;
168  if missing(DemAge) then IMP_DemAge = 59.262912088;
169  label IMP_LOG_GiftAvgCard36 = 'Imputed: Transformed: Gift Amount Average Card 36 Months';
170  IMP_LOG_GiftAvgCard36 = LOG_GiftAvgCard36;
171  if missing(LOG_GiftAvgCard36) then IMP_LOG_GiftAvgCard36 = 2.5855317177;
172  label IMP_REP_DemMedIncome = 'Imputed: Replacement: Median Income Region';
173  IMP_REP_DemMedIncome = REP_DemMedIncome;
174  if missing(REP_DemMedIncome) then IMP_REP_DemMedIncome = 53570.850493;
175  *;
176  *INDICATOR VARIABLES;
177  *;
178  label M_DemAge = "Imputation Indicator for DemAge";
179  if missing(DemAge) then M_DemAge = 1;
180  else M_DemAge= 0;
181  label M_LOG_GiftAvgCard36 = "Imputation Indicator for LOG_GiftAvgCard36";
182  if missing(LOG_GiftAvgCard36) then M_LOG_GiftAvgCard36 = 1;
183  else M_LOG_GiftAvgCard36= 0;
184  label M_REP_DemMedIncome = "Imputation Indicator for REP_DemMedIncome";
185  if missing(REP_DemMedIncome) then M_REP_DemMedIncome = 1;
186  else M_REP_DemMedIncome= 0;
```

- Go to line 187. This code comes from the Regression node. It is this code that actually adds the predictions to a Score data set. (Only a portion of this part of the Score code is shown below.)

```
187  *-------------------------------------------------------------*;
188  * TOOL: Regression;
189  * TYPE: MODEL;
190  * NODE: Reg;
191  *-------------------------------------------------------------*;
192  ************************************;
193  *** begin scoring code for regression;
194  ************************************;
195
196  length _WARN_ $4;
197  label _WARN_ = 'Warnings' ;
198
199  length I_TARGET_B $ 12;
200  label I_TARGET_B = 'Into: TARGET_B' ;
201  *** Target Values;
202  array REGDRF [2] $12 _temporary_ ('1' '0' );
203  label U_TARGET_B = 'Unnormalized Into: TARGET_B' ;
204  *** Unnormalized target values;
205  ARRAY REGDRU[2]  _TEMPORARY_ (1 0);
206
207  drop _DM_BAD;
208  _DM_BAD=0;
209
210  *** Check DemMedHomeValue for missing values ;
211  if missing( DemMedHomeValue ) then do;
212     substr(_warn_,1,1) = 'M';
213     _DM_BAD = 1;
214  end;
215
216  *** Check GiftTimeLast for missing values ;
217  if missing( GiftTimeLast ) then do;
218     substr(_warn_,1,1) = 'M';
219     _DM_BAD = 1;
220  end;
221
222  *** Check LOG_GiftAvgAll for missing values ;
223  if missing( LOG_GiftAvgAll ) then do;
224     substr(_warn_,1,1) = 'M';
225     _DM_BAD = 1;
.
.
.
```

- Go to line **325**. This block of code comes from the Model Comparison node. It adds demi-decile bin numbers to the scored output. For example, bin 1 corresponds to the top 5% of the data as scored by the Regression model, bin 2 corresponds to the next 5%, and so on. (Only a portion of this part of the Score code is shown below.)

```
325  *-------------------------------------------------------------*;
326  * TOOL: Model Compare Class;
327  * TYPE: ASSESS;
328  * NODE: MdlComp;
329  *-------------------------------------------------------------*;
330  if (P_TARGET_B1 ge 0.08433506126363) then do;
331  b_TARGET_B = 1;
332  end;
333  else
334  if (P_TARGET_B1 ge 0.07379667308709) then do;
335  b_TARGET_B = 2;
336  end;
337  else
338  if (P_TARGET_B1 ge 0.0675579857262) then do;
339  b_TARGET_B = 3;
340  end;
341  else
342  if (P_TARGET_B1 ge 0.06293120025423) then do;
343  b_TARGET_B = 4;
344  end;
345  else
346  if (P_TARGET_B1 ge 0.0592652370746) then do;
347  b_TARGET_B = 5;
348  end;
349  else
350  if (P_TARGET_B1 ge 0.05627609817451) then do;
351  b_TARGET_B = 6;
352  end;
353  else
354  if (P_TARGET_B1 ge 0.05380116377971) then do;
355  b_TARGET_B = 7;
356  end;
357  else
.
.
.
```

- Go to line **409**. This block of code comes from the Score node. It adds the following standardized variables to the scored data set:

EM_CLASSIFICATION	Prediction for **TARGET_B**
EM_DECISION	Recommended Decision for **TARGET_B**
EM_EVENTPROBABILITY	Probability for Level 1 of **Target**
EM_PROBABILITY	Probability of Classification
EM_PROFIT	Expected Profit for **TARGET_B**
EM_SEGMENT	Segment

```
409  *-------------------------------------------------------------*;
410  * TOOL: Score Node;
411  * TYPE: ASSESS;
412  * NODE: Score;
413  *-------------------------------------------------------------*;
414  *-------------------------------------------------------------*;
415  * Score: Creating Fixed Names;
416  *-------------------------------------------------------------*;
417  LABEL EM_SEGMENT = 'Segment';
418  EM_SEGMENT = b_TARGET_B;
419  LABEL EM_EVENTPROBABILITY = 'Probability for level 1 of TARGET_B';
420  EM_EVENTPROBABILITY = P_TARGET_B1;
421  LABEL EM_PROBABILITY = 'Probability of Classification';
422  EM_PROBABILITY =
423  max(
424  P_TARGET_B1
425  ,
426  P_TARGET_B0
427  );
428  LENGTH EM_CLASSIFICATION $%dmnorlen;
429  LABEL EM_CLASSIFICATION = "Prediction for TARGET_B";
430  EM_CLASSIFICATION = I_TARGET_B;
431  LENGTH EM_DECISION $%dmnorlen;
432  LABEL EM_DECISION= "Recommended Decision for TARGET_B";
433  EM_DECISION = D_TARGET_B;
434  LABEL EM_PROFIT= "Expected Profit for TARGET_B";
435  EM_PROFIT = EP_TARGET_B;
```

To use this code, you must embed it in a DATA step. The easiest way to do this is by saving it to a SAS code file and including it in your DATA step.

3. Select **File** ⇨ **Save As** to save this code to a location of your choice.