SAS® Model Manager 14.2: Feature Contribution Index Macros

SAS® Documentation
April 29, 2018

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *SAS® Model Manager 14.2: Feature Contribution Index Macros*. Cary, NC: SAS Institute Inc.

SAS® Model Manager 14.2: Feature Contribution Index Macros

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

April 2018

Contents

Chapter 1

Introduction to FCI Macros

Overview

What Are FCI Macros?

The feature contribution index (FCI) macros enable you to compute the feature contribution indices for interval and nominal predictors, and create an ad hoc report. Before you use these macros, the model outcome must already be available. That is, the input data set must be scored using the model first, and then the model outcomes must be saved in the scoring data set.

This document contains the syntax and argument descriptions, as well as examples for the macros. The macros and sample data are available for download from the SAS Model Manager Downloads page on support.sas.com.

Measuring Predictor Influence

When you train a model, you can evaluate the importance of predictors within that model. Some training algorithms (for example, decision trees) provide variable importance indices. Alternatively, they can be calculated using statistics on predictors (for example, multiplying –1 by the logarithm of a predictor's significance value from a regression model). When you deploy a model, you might want to determine how much influence a predictor has on model outcomes. Because the distributions of predictors in a scoring data set might differ from those found in the training data set, the original variable importance metrics obtained from training the model might no longer be relevant.

Scoring Details

Therefore, the procedures used to compute the original variable importance metrics might not work for scoring because the observed target variables that are required by the procedures are not available in the scoring data set. If you face these constraints on the scoring data set, here are possible solutions to help you meet your users' needs. When you follow these procedures, you do not need to wait for the observed target values to be available. Instead, you can use the predicted values for the interval target variable or the predicted probabilities for the nominal target variable. Instead of customizing procedures for different models, compute the FCI, a model-neutral procedure.

How the Model Outcome Is Determined

The contribution of a predictor or a feature of a model is defined as the *aggregated influence* of that predictor's values on the spread of the model outcome. For classification models (nominal target variables), the model outcome consists of the predicted probabilities. For regression models (interval target variables), the model outcome is the predicted value. In both types of models, the model outcome consists of one or more numeric values. To measure the contribution, use the following procedure:

1. For each numeric value in the model outcome, build the main effect analysis of variance with each individual predictor.
2. Measure the contribution of a predictor by the R-squared statistic. (For nominal predictors, this is the full eta-squared statistic. For interval predictors, this is the squared Pearson correlation coefficient.)

Note: For nominal targets, the contribution indices are aggregated for each individual predicted probability.

The index is a numeric value between 0 and 1, inclusive. A value of 1 indicates that the variable contributes the most to the model and is most likely the only variable needed for the model. A value of 0 indicates that the variable contributes the least to the model, and its absence would have little or no impact on the model.

Aggregation for Nominal Targets

There is one FCI for each predicted probability. In order to provide a single index to users, calculate a weighted sum of the individual FCIs.

Here are two common options for weights:

- Each weight is equivalent to the reciprocal of the number of predicted probabilities (uninformative).
- Weights are equivalent to the observed relative frequencies (proportions) of the target categories in the training data set.

For a binary target variable, the choice of weight should not matter because the contribution of a predictor to either predicted probability is the same.

Chapter 2

Macro Reference

Dictionary

%Compute_FCI Macro

Calls the %Compute_FCI_NomPred and the %Compute_FCI_IntPred macros to compute the FCIs' given input specifications. It overwrites the output FCI data set with the FCIs. Besides using a DATA step statement, this macro calls these procedures: CONTENTS, DATASETS, PRINT, and SORT.

Syntax

%Compute_FCI (
InData=*scoring-dataset-name*,
TargetSpec=*target-specification-dataset*,
PredictorSpec=*predictor-specification-dataset*,
OutFCIData=*output-FCI-dataset*,
<NameFCI=*variable-name-FCI-index*>,
<WorkLib=*work-library-reference*>,
<Debug=Y | N>,
);

Required Arguments

InData=*scoring-dataset-name*
Specifies the name of the scoring data set.

TargetSpec=*target-specification-dataset*
Specifies the name of the data set that contains the target specifications.

PredictorSpec=*predictor-specification-dataset*
Specifies the name of the data set that contains the predictor specifications.

OutFCIData=*output-FCI-dataset*
Specifies the name of the output FCI data set.

Optional Arguments

NameFCI=*variable-name-FCI-index*
Specifies the name of the variable that contains the aggregated FCI. The default name is _FCI_.

WorkLib=*work-library-reference*
Specifies the name of the working library reference. The default library is WORK.

Debug
Indicates whether to display debugging information. The default value is **N**.

Details

The following data sets are associated with the %Compute_FCI macro: target specification, predictor specification, and output FCI.

The target specification data set has as many rows as the number of model outcomes. The %Compute_FCI macro looks for the following variables in the data set.

Table 2.1 *Target Specification Data Set*

Variable	Type	Valid Values	Description
NAME	String; maximum of 32 characters	SAS Name	Specifies the name of a model outcome.
PRIOR	Numeric	Number	Specifies the weight for calculating the weighted sum of the individual FCIs.

The predictor specification data set has as many rows as the number of predictors. The %Compute_FCI macro looks for the following variables in the data set.

Table 2.2 *Predictor Specification Data Set*

Variable	Type	Valid Values	Description
NAME	String; maximum of 32 characters	SAS Name	Specifies the name of a predictor.
LEVEL	String; maximum of 8 characters	INTERVAL or NOMINAL	Specifies the level for the predictor.

Variable	Type	Valid Values	Description
QMISSNOM	String; maximum of 1 character	N or Y	Specifies whether to include missing values in nominal predictors. This value is ignored if LEVEL is INTERVAL.

The output FCI data set has as many rows as the number of predictors. The following variables are in this data set.

Table 2.3 *Output FCI Data Set*

Variable	Type	Valid Values	Description
VARNAME	String; maximum of 32 characters	SAS Name	Specifies the name of a predictor.
User-supplied name	Numeric	Between 0 and 1, inclusive	Users supply a name for the aggregated (weighted sum) FCI.

Example

Code fragment:

```
%Compute_FCI (
   InData = TESTLIB.ScoreData,
   TargetSpec = TESTLIB.TargetSpec,
   PredictorSpec = TESTLIB.PredictorSpec,
   OutFCIData = TESTLIB.FCIData,
   NameFCI = _FCI_BAD_,
   WorkLib = WORK,
   Debug = N);
```

For a full example, see “Example 1: Running FCI Macros Using Base SAS or SAS Studio” on page 11.

%Compute_FCI_NomPred Macro

Computes FCIs for a list of nominal predictors. It appends FCIs to the output FCI data set. Besides using a DATA step statement, this macro calls these procedures: APPEND, DATASETS, PRINT, and SUMMARY.

Syntax

%Compute_FCI_NomPred (
  InData=*scoring-dataset-name*,
  DepVarList=*list-of-variable-names*,
  NomPredList=*list-of-nominal-predictors*,
  OutFCIData=*output-FCI-dataset*,

<qMissNom=Y | N>,
<WorkLib=*work-library-reference*>,
<Debug=Y | N>,
);

Required Arguments

InData=*scoring-dataset-name*
Specifies the name of the scoring data set.

DepVarList=*list-of-variable-names*
Lists the names of the numeric model output, separated by blanks.

NomPredList=*list-of-nominal-predictors*
Lists the names of nominal predictors, separated by blanks.

OutFCIData=*output-FCI-dataset*
Specifies the name of the output FCI data set.

Optional Arguments

qMissNom
Indicates whether to include missing values in nominal predictors. The default value is **N**.

WorkLib=*work-library-reference*
Specifies the name of the working library reference. The default library is WORK.

Debug
Indicates whether to display debugging information. The default value is **N**.

Example

To determine the FCIs of the nominal predictors, call the %Compute_FCI_NomPred macro directly. Here is the SAS code for the badloans_monitor_4 data set.

```
data badloans_score;
   set MMData.badloans_monitor_4;
   %include _LGRSCR_;
run;

%Compute_FCI_NomPred
(
   InData = badloans_score,
   DepVarList = %str(P_BAD0 P_BAD1),
   IntPredList = &NomPred.,
   qMissNom = N,
   OutFCIData = OutFCIData_4_Nom,
   WorkLib = WORK,
   Debug = &Debug.
);

proc print data = OutFCIData_4_Nom;
run;
```

For a full example that includes output results, see “Example 1: Running FCI Macros Using Base SAS or SAS Studio” on page 11.

%Compute_FCI_IntPred Macro

Computes FCIs for a list of interval predictors. It appends FCIs to the output FCI data set. Besides using a DATA step statement, this macro calls the APPEND procedure.

Syntax

%Compute_FCI_IntPred (
  InData=*scoring-dataset-name*,
  DepVarList=*list-of-variable-names*,
  IntPredList=*list-of-interval-predictors*,
  OutFCIData=*output-FCI-dataset*,
  <WorkLib=*work-library-reference*>,
  <Debug=Y | N>,
);

Required Arguments

InData=*scoring-dataset-name*
  Specifies the name of the scoring data set.

DepVarList=*list-of-variable-names*
  Contains a list of names of the numeric model output, separated by blanks.

IntPredList=*list-of-interval-predictors*
  Contains a list of names of interval predictors, separated by blanks.

OutFCIData=*output-FCI-dataset*
  Specifies the name of the output FCI data set.

Optional Arguments

WorkLib=*work-library-reference*
  Specifies the name of the working library reference. The default library is WORK.

Debug
  Indicates whether to display debugging information. The default value is `N`.

Example

To determine the FCIs of the interval predictors, call the %Compute_FCI_IntPred macro directly. Here is the SAS code for the badloans_monitor_4 data set.

```
data badloans_score;
   set MMData.badloans_monitor_4;
   %include _LGRSCR_;
run;

%Compute_FCI_IntPred
(
   InData = badloans_score,
   DepVarList = %str(P_BAD0 P_BAD1),
   IntPredList = &IntPred.,
```

```
      OutFCIData = OutFCIData_4_Int,
      WorkLib = WORK,
      Debug = &Debug.
   );

   proc print data = OutFCIData_4_Int;
   run;
```

For a full example that includes output results, see “Example 1: Running FCI Macros Using Base SAS or SAS Studio” on page 11.

%MM_AdHocReport_FCI Macro

Generates an ad hoc report in SAS Model Manager. It calls the %Compute_FCI macro to compute the FCIs for a list of scoring data sets. Besides using a DATA step statement and procedures that are used in the macros called, this macro also calls these procedures: APPEND, DELETE, PRINT, SGPANEL, SGPLOT, SQL, and TABULATE.

Syntax

%MM_AdHocReport_FCI (
    ScoreDataPrefix=*scoring-dataset-prefix-name*,
    P_VarList=*list-of-predicted-outcome-variables*,
    <ScoreDataCount=*scoring-dataset-count*>,
    <QMissNomPred=Y | N>,
    <Debug=Y | N>,
);

Required Arguments

ScoreDataPrefix=*scoring-dataset-prefix-name*
Specifies the name of the common prefix (including the library name) of scoring data set names.

P_VarList=*list-of-predicted-outcome-variables*
Contains a list of model outcome variables.

Optional Arguments

ScoreDataCount=*scoring-dataset-count*
Specifies the number of scoring data sets. A positive numeric value is expected. The default value is 1.

QMissNomPred
Indicates whether to include missing values in nominal predictors. The default value is **N**.

Debug
Indicates whether to display debugging information. The default value is **N**.

Details

This macro assumes that the scoring data sets are named according to the convention specified by &ScoreDataPrefix.k, where k is an integer from 1 to &ScoreDataCount

without any gaps. It reads the following XML files whose actual locations are pointed to by the SAS Model Manager macro variables.

Table 2.4 XML Files Associated with MM_AdHocReport_FCI Macro

XML File	Macro Variable	Description
InputVar.xml	&_MM_Input.	Constructs the predictor specification data set.
OutputVar.xml	&_MM_Output.	Constructs the target specification data set. The uninformative prior is used.

Copy the following macro files to a folder on your server in order to use them for generation of the ad hoc report. Note that full permissions, including execution, must exist for that folder:

- Compute_FCI_NomPred.sas
- Compute_FCI_IntPred.sas
- Compute_FCI.sas
- MM_AdHocReport_FCI.sas

Example: %MM_AdHocReport_FCI Macro Code Example

```
/* Load the custom SAS macro code for calculating the feature contribution indices. */
%let MyLib = %str(c:\FCI);

%include "&MyLib.\Macros\MM_AdHocReport_FCI.sas";
%include "&MyLib.\Macros\Compute_FCI.sas";
%include "&MyLib.\Macros\Compute_FCI_IntPred.sas";
%include "&MyLib.\Macros\Compute_FCI_NomPred.sas";

libname TESTLIB "&MyLib.\Data";

/* Specify the folders that the tables and the charts are to be written to. */
ods html path = "&MyLib." gpath = "&MyLib.";

/* Load the macros inside this catalog for displaying the report. */
filename mmreport catalog "sashelp.modelmgr.reportexportmacros.source";
%include mmreport;
%mm_exportReportsBegin (
   fileName = BadLoan_FCI_Report,
   reportFormat = HTML,
   reportstyle = seaside);

/* Create reports for the four data sets with scores. */
/* Include missing values for nominal predictors in the calculations. */
%MM_AdHocReport_FCI (
   ScoreDataPrefix = %str(TESTLIB.BadLoans_Monitor_Scoring_),
   ScoreDataCount = 4,
   P_VarList = %str(P_BAD0 P_BAD1),
   QMissNomPred = N,
```

```
    Debug = N);

%mm_exportReportsEnd(reportFormat = HTML);
/* End of the ad hoc report. */
```

The MM_AdHocReport_FCI macro generates one table and two charts:

- The table has two variables in the row dimension and one variable in the column dimension:
  - The two variables in the row dimension are names of predictors and their measurement levels.
  - The variable in the column dimension is the scoring data set sequential number (the integer **k**).
  - The cell contents are the FCI displayed in percentage format.
- The first chart is a panel series chart:
  - Each predictor name constitutes one panel.
  - Each series plots the FCIs versus the scoring data set sequential number.
- The second chart is an overlay series chart:
  - Each scoring data set sequential number constitutes one series.
  - Each series plots the FCIs versus names of the predictors.

For a full example that includes output results, see "Example 2: Running FCI Macros Using SAS Model Manager" on page 24.

Chapter 3
Examples

Example 1: Running FCI Macros Using Base SAS or SAS Studio

Overview

The %Compute_FCI macro assists in calculating FCIs. In addition to specifying the data set on which a model is deployed, this macro requires a specification data set for the target variable and another for the predictors. The target specification data set should have two columns (NAME and PRIOR), and as many rows as the number of levels of the target variable (it is assumed that an interval target variable has a single level). The predictor specification data set should have three columns (NAME, LEVEL, and QMISSNOM), and as many rows as the number of predictors. For more information, see "%Compute_FCI Macro" on page 3.

Note: The paths that you specify in your example code for the MMLib macro variable, as well as the MMData and MMPub libraries, should be the same as where the macros are installed. For example, in the code examples below, the Windows directory path for the data tables is `c:\FCI\Data`.

To run this example:

1. Specify the macro variables Debug and MMLib.
2. Set the Debug argument to Y, to request the debugging information from the macros.
3. Set the MMLib macro variable to the upper-level folder where you want to store your work.

4. Create a library named MMData and set the directory path to where the FCI data tables are located (for example, **c:\FCI\Data**).
5. Create a folder named **Publication** within the FCI upper-level folder, and then create a library named MMPub that points to the folder that you created.

Because the ad hoc report contains charts, the PATH and GPATH folders must be defined using the ODS HTML command, and Write permissions must exist for these two folders. It is recommended that the ODS GRAPHICS command be used to specify the WIDTH and the HEIGHT to be larger values because the panel chart can have as many panels as the number of predictors.

SAS Code Examples

Example Code 1 *Compute FCIs*

```
%let Debug = N;
%let MMLib = %str(c:\FCI);

libname MMData "&MMLib.\Data";
libname MMPub "&MMLib.\Publication";

/* Here you specify the rows in the predictor specification data set that are for the interval predictors.
   Only the names are specified. The other two columns are specified later. */

data MMPub.IntPredSpec;
   length NAME $ 32;
   input NAME $;
   datalines;
CLAGE
CLNO
DELINQ
DEROG
MORTDUE
NINQ
YOJ
;

/* Here you specify the rows in the predictor specification data set that are for the nominal predictors.
   Only the names are specified. The other two columns are specified later. */

data MMPub.NomPredSpec;
   length NAME $ 32;
   input NAME $;
   datalines;
JOB
REASON
;

/* The names are extracted into the respective macro variables. These macro variables help you
   write a more elegant syntax for building the logistic model. */

proc sql noprint;
   select NAME into :IntPred separated by ' ' from MMPub.IntPredSpec;
   select NAME into :NomPred separated by ' ' from MMPub.NomPredSpec;
quit;
```

```
/* The logistic model is built using the HPLOGISTIC procedure, which is a high-performance
   statistical procedure. It requires a license for SAS High-Performance Statistics. The procedure
   writes a SAS program code file that is used for scoring the incoming data sets. */

filename _LGRSCR_ "&MMLib.\Publication\Badloans_Logistic_Score.sas";
proc hplogistic data = MMData.badloans_train
                maxiter = 100
                technique = newrap
                namelen = 128;
   class &NomPred. / param = glm order = freq descending;
   model BAD (event = '1') = &IntPred. &NomPred. / link = logit rsquare association;
   selection method = stepwise;
   code file = _LGRSCR_;
run;

/* Three SAS macro code files must be loaded for calculating the FCIs. */

%include "&MMLib.\Macros\Compute_FCI.sas";
%include "&MMLib.\Macros\Compute_FCI_IntPred.sas";
%include "&MMLib.\Macros\Compute_FCI_NomPred.sas";

/* The predictor specification data set is generated using the individual parts. At the same time,
   the other two columns,  LEVEL and QMISSNOM, are populated with the appropriate values. */

data MMPub.PredictorSpec;
   set MMPub.IntPredSpec (in = in1)
       MMPub.NomPredSpec (in = in2);

   length LEVEL $ 16;
   length QMISSNOM $ 1;

   if (in1) then
   do;
      LEVEL = 'INTERVAL';
      QMISSNOM = ' ';
   end;
   else if (in2) then
   do;
      LEVEL = 'NOMINAL';
      QMISSNOM = 'N';
   end;
run;

/* The predictor specification data set is printed so that you can see its contents. */

proc print data = MMPub.PredictorSpec;
run;

/* The target specification data set is generated from the output of the FREQ procedure because
   you want the weights to be proportional to the frequencies of the target values. The NAME column
   contains P_BAD0 and P_BAD1, which are the predicted probabilities of BAD = 0 and BAD = 1,
   respectively. These two probabilities are the model outcomes. */

proc freq data = MMData.badloans_train;
   table BAD / noprint out = BAD_Percent;
```

```
run;

data MMPub.TargetSpec;
   set BAD_Percent;

   length NAME $ 32;
   length PRIOR 8;

   NAME = cat('P_BAD', strip(vvalue(BAD)));
   PRIOR = PERCENT / 100;

   keep NAME PRIOR;
run;

/* The target specification data set is printed so that you can see its contents. */

proc print data = MMPub.TargetSpec;
run;

/* The data set is first scored by including the SAS program code from the HPLOGISTIC procedure.
   Next, the %Compute_FCI macro is executed by specifying the appropriate arguments.  The FCIs are
   calculated for the badloans_train data set as the benchmark. */

data badloans_score;
   set MMData.badloans_train;
   %include _LGRSCR_;
run;

%Compute_FCI
(
   InData = badloans_score,
   TargetSpec = MMPub.TargetSpec,
   PredictorSpec = MMPub.PredictorSpec,
   OutFCIData = OutFCIData_0,
   NameFCI = %str(_FCI_),
   WorkLib = WORK,
   Debug = &Debug.
);

/* Calculate the FCIs for the badloans_monitor_1 data set. */

data badloans_score;
   set MMData.badloans_monitor_1;
   %include _LGRSCR_;
run;

%Compute_FCI
(
   InData = badloans_score,
   TargetSpec = MMPub.TargetSpec,
   PredictorSpec = MMPub.PredictorSpec,
   OutFCIData = OutFCIData_1,
   NameFCI = %str(_FCI_),
   WorkLib = WORK,
   Debug = &Debug.
);
```

```
/* Calculate the FCIs for the badloans_monitor_2 data set. */

data badloans_score;
   set MMData.badloans_monitor_2;
   %include _LGRSCR_;
run;

%Compute_FCI
(
   InData = badloans_score,
   TargetSpec = MMPub.TargetSpec,
   PredictorSpec = MMPub.PredictorSpec,
   OutFCIData = OutFCIData_2,
   NameFCI = %str(_FCI_),
   WorkLib = WORK,
   Debug = &Debug.
);

/* Calculate the FCIs for the badloans_monitor_3 data set. */

data badloans_score;
   set MMData.badloans_monitor_3;
   %include _LGRSCR_;
run;

%Compute_FCI
(
   InData = badloans_score,
   TargetSpec = MMPub.TargetSpec,
   PredictorSpec = MMPub.PredictorSpec,
   OutFCIData = OutFCIData_3,
   NameFCI = %str(_FCI_),
   WorkLib = WORK,
   Debug = &Debug.
);

/* Calculate the FCIs for the badloans_monitor_4 data set. */

data badloans_score;
   set MMData.badloans_monitor_4;
   %include _LGRSCR_;
run;

%Compute_FCI
(
   InData = badloans_score,
   TargetSpec = MMPub.TargetSpec,
   PredictorSpec = MMPub.PredictorSpec,
   OutFCIData = OutFCIData_4,
   NameFCI = %str(_FCI_),
   WorkLib = WORK,
   Debug = &Debug.
);

/* Combine the five data sets of FCIs and create the _TIME_ variable. */
```

```
/* Combine all the results into one data set indexed by their times */
data MMPub.badloans_FCI_Result;
   set OutFCIData_0 (in = in0)
       OutFCIData_1 (in = in1)
       OutFCIData_2 (in = in2)
       OutFCIData_3 (in = in3)
       OutFCIData_4 (in = in4);

        if (in0) then _TIME_ = 0;
   else if (in1) then _TIME_ = 1;
   else if (in2) then _TIME_ = 2;
   else if (in3) then _TIME_ = 3;
   else if (in4) then _TIME_ = 4;
run;

/* Bring the measurement levels of the predictors in from the predictor specification data set. */

proc sort data = MMPub.badloans_FCI_Result;
   by _VARNAME_ _TIME_;
run;

proc sort data = MMPub.PredictorSpec;
   by NAME;
run;

data MMPub.badloans_FCI_Result;
   merge MMPub.badloans_FCI_Result (in = in0 rename = (_VARNAME_ = NAME))
         MMPub.PredictorSpec (in = in1);
   by NAME;

   keep NAME LEVEL _FCI_ _TIME_;
run;

/* Review and compare the FCIs. */

title2 "Feature Contribution Indices";
ods graphics / reset height = 7in;

proc tabulate data = MMPub.badloans_FCI_Result;
   class NAME LEVEL _TIME_;
   var _FCI_;
   table LEVEL='Measurement Level' * NAME='Predictor',
         _TIME_='Monitor Time' * (sum = '' * f = percent10.1) * _FCI_ = '';
   format _FCI_ 10.4;
run;

proc sgpanel data = MMPub.badloans_FCI_Result;
   panelby NAME / onepanel novarname;
   series y = _FCI_ x = _TIME_ / markers;
   rowaxis grid label = 'Feature Contribution Index';
   colaxis grid integer label = 'Monitor Time';
run;

proc sgplot data = MMPub.badloans_FCI_Result;
   series y = _FCI_ x = NAME / markers group = _TIME_ name = 'series';
```

```
   keylegend 'series' / location = outside position = right title = 'Monitor Time';
   yaxis grid label = 'Feature Contribution Index';
   xaxis grid label = 'Predictor';
run;

/* The results indicate that the predicted probabilities are collectively correlated with the DELINQ
   variable.  However, it substantially drops its contribution to the predicted probabilities in the
   badloans_monitor_3.  You then can use the histograms to visually compare the distribution of DELINQ. */

proc sgplot data = MMData.badloans_train;
   histogram DELINQ / binwidth = 1 scale = percent;
   xaxis values = (0 to 20 by 1) offsetmin = 0.05 offsetmax = 0.05;
   yaxis values = (0 to 100 by 10) grid;
   footnote MMData.badloans_train;
run;

proc sgplot data = MMData.badloans_monitor_1;
   histogram DELINQ / binwidth = 1 scale = percent;
   xaxis values = (0 to 20 by 1) offsetmin = 0.05 offsetmax = 0.05;
   yaxis values = (0 to 100 by 10) grid;
   footnote MMData.badloans_monitor_1;
run;

proc sgplot data = MMData.badloans_monitor_2;
   histogram DELINQ / binwidth = 1 scale = percent;
   xaxis values = (0 to 20 by 1) offsetmin = 0.05 offsetmax = 0.05;
   yaxis values = (0 to 100 by 10) grid;
   footnote MMData.badloans_monitor_2;
run;

proc sgplot data = MMData.badloans_monitor_3;
   histogram DELINQ / binwidth = 1 scale = percent;
   xaxis values = (0 to 20 by 1) offsetmin = 0.05 offsetmax = 0.05;
   yaxis values = (0 to 100 by 10) grid;
   footnote MMData.badloans_monitor_3;
run;

proc sgplot data = MMData.badloans_monitor_4;
   histogram DELINQ / binwidth = 1 scale = percent;
   xaxis values = (0 to 20 by 1) offsetmin = 0.05 offsetmax = 0.05;
   yaxis values = (0 to 100 by 10) grid;
   footnote MMData.badloans_monitor_4;
run;
```

Example Code 2 *Compute the FCI for Interval Predictors and Nominal Predictors*

```
/* Suppose you want to see the unweighted indices for the badloans_monitor_4 data set,
   or suppose that you want to skip the steps in creating the specification data sets. */

data badloans_score;
   set MMData.badloans_monitor_4;
   %include _LGRSCR_;
run;

%Compute_FCI_IntPred
(
```

```
   InData = badloans_score,
   DepVarList = %str(P_BAD0 P_BAD1),
   IntPredList = &IntPred.,
   OutFCIData = OutFCIData_4_Int,
   WorkLib = WORK,
   Debug = &Debug.
);

%Compute_FCI_NomPred
(
   InData = badloans_score,
   DepVarList = %str(P_BAD0 P_BAD1),
   NomPredList = &NomPred.,
   qMissNom = N,
   OutFCIData = OutFCIData_4_Nom,
   WorkLib = WORK,
   Debug = &Debug.
);

/* The output FCI data sets are printed so that you can see the output results. */

proc print data = OutFCIData_4_Int;
run;

proc print data = OutFCIData_4_Nom;
run;
```

Output Results

Here are the output results that include the observations for the predictor specification data set, the observations for the target specification data set, the measurement level for the interval predictors and nominal predictors, and the calculated FCIs.

Figure 3.1 *Predictor Specification Data Set*

Obs	NAME	LEVEL	QMISSNOM
1	CLAGE	INTERVAL	
2	CLNO	INTERVAL	
3	DELINQ	INTERVAL	
4	DEROG	INTERVAL	
5	MORTDUE	INTERVAL	
6	NINQ	INTERVAL	
7	YOJ	INTERVAL	
8	JOB	NOMINAL	N
9	REASON	NOMINAL	N

Figure 3.2 *Target Specification Data Set*

Obs	NAME	PRIOR
1	P_BAD0	0.79788
2	P_BAD1	0.20212

Figure 3.3 *FCI Measurement Levels for Internal and Nominal Predictors*

Feature Contribution Indices

		Monitor Time				
		0	1	2	3	4
Measurement Level	Predictor					
INTERVAL	CLAGE	9.9%	0.9%	10.7%	16.6%	17.1%
	CLNO	0.0%	0.7%	0.0%	5.6%	0.4%
	DELINQ	49.6%	47.5%	35.9%	8.6%	33.7%
	DEROG	33.8%	31.3%	28.6%	30.8%	40.2%
	MORTDUE	2.9%	2.6%	3.1%	5.2%	3.0%
	NINQ	16.1%	13.8%	10.4%	23.7%	24.0%
	YOJ	1.9%	0.1%	1.0%	5.8%	4.2%
NOMINAL	JOB	4.9%	4.0%	5.5%	9.7%	6.5%
	REASON	0.0%	0.3%	0.3%	0.0%	0.6%

Figure 3.4 *FCI Plotted Against the Monitoring Time for Each Predictor*

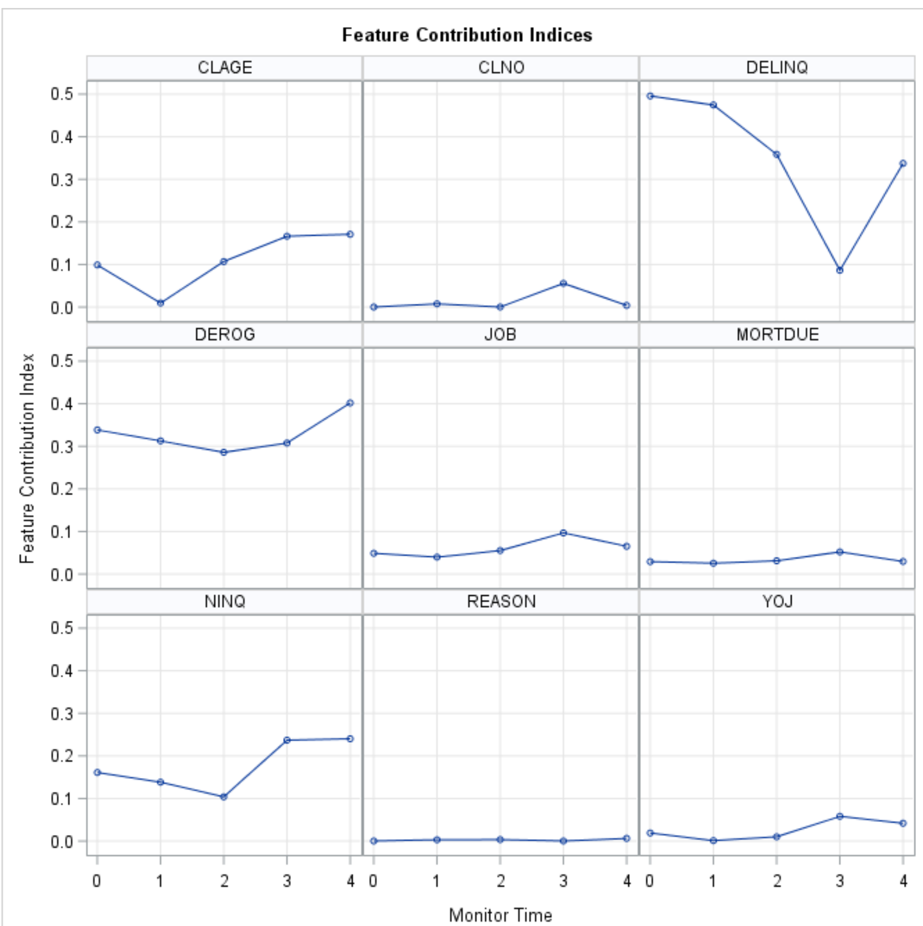
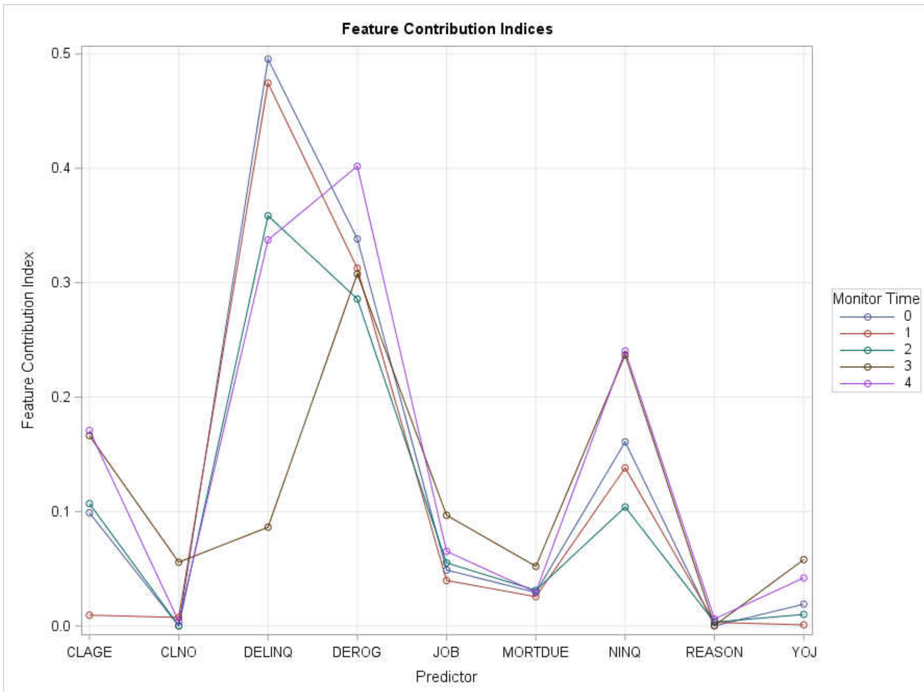


Figure 3.5 *FCI Plotted Against Predictors for Each Monitoring Time*



For the histograms that are shown below, the horizontal axes are the values of DELINQ. The vertical axes are the percent of observations. The histograms correspond to the _TIME_ variable in the row-major order. The first histogram is for _TIME_ = 0 (in other words, the benchmark). The other histograms are for _TIME_ = 1, 2, 3, and 4.

The histogram that corresponds to _TIME_ = 3 shows that DELINQ = 0 for almost all of the observations. In other words, the DELINQ variable seems to be constant in the badloans_monitor_3 data set. Therefore, do not expect its correlations with the two predicted probabilities in the badloans_monitor_3 data set to be as high as those in other data sets. If you do not have the badloans_monitor_4 data set, then you can conclude that the model is stale and a refresh is necessary. Because the badloans_monitor_4 data set is present and the FCI goes up again, you can safely conclude that the findings in the badloans_monitor_3 data set are due to spurious fluctuation and the time to rebuild the model has not yet come.

Figure 3.6 *FCI of the DELINQ Predictor Versus Percent of Observations for the badloans_train Data Set*

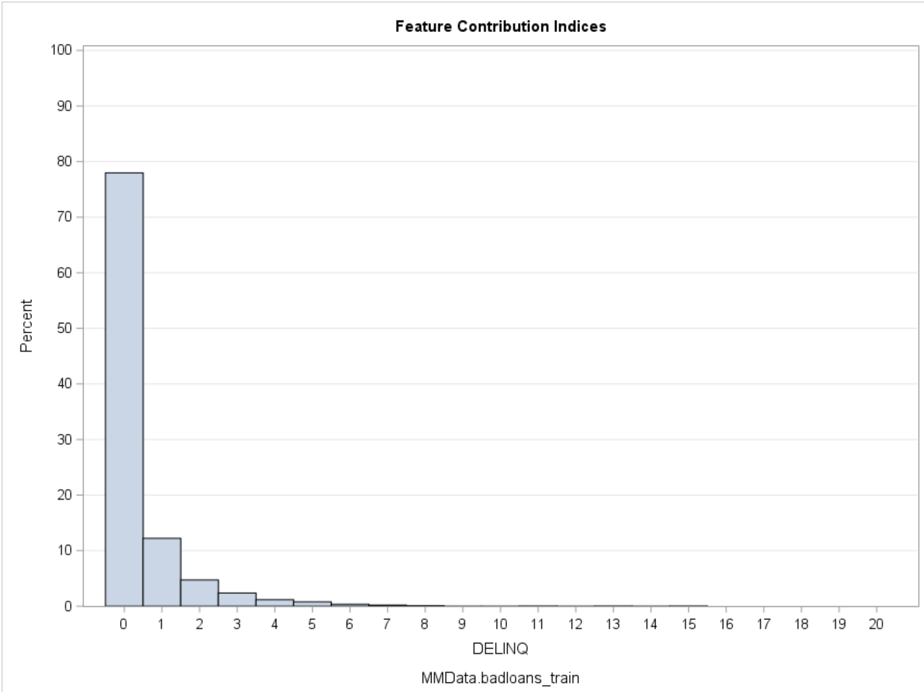


Figure 3.7 *FCI of the DELINQ Predictor Versus Percent of Observations for the badloans_monitor_1 Data Set*

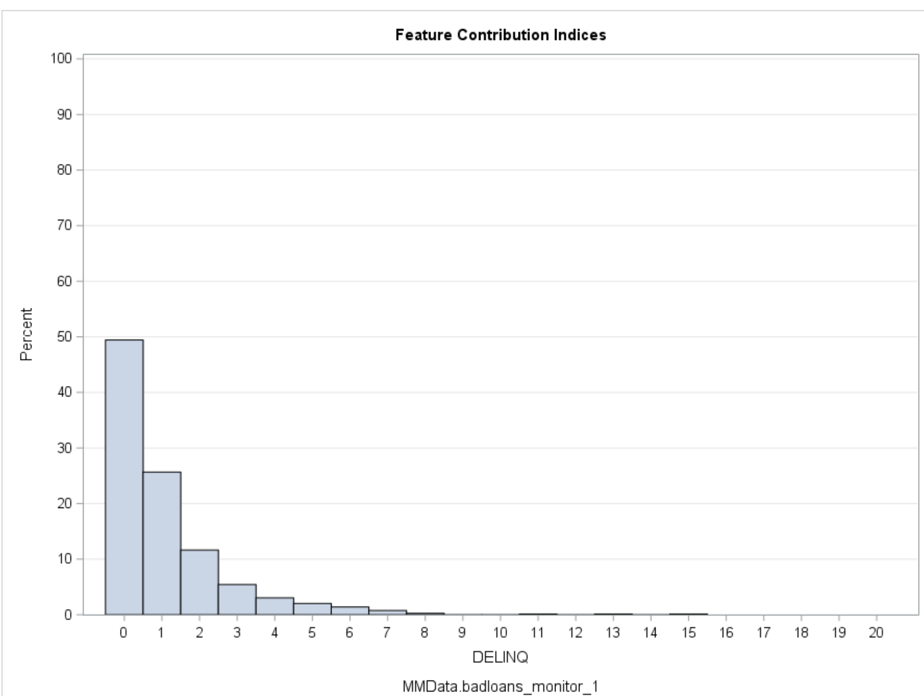


Figure 3.8 *FCI of the DELINQ Predictor Versus Percent of Observations for the badloans_monitor_2 Data Set*

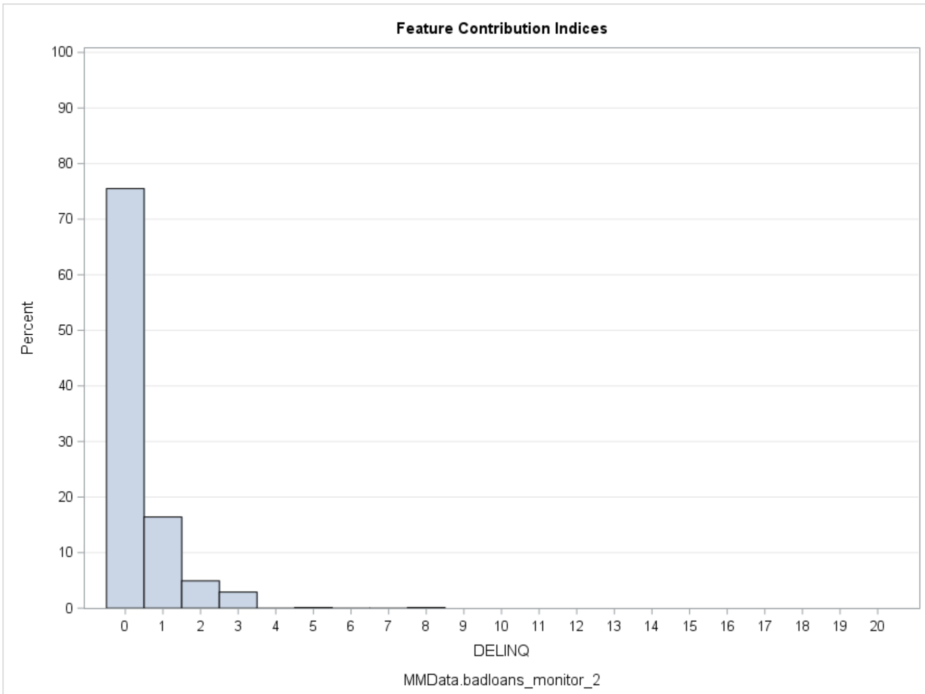


Figure 3.9 *FCI of the DELINQ Predictor Versus Percent of Observations for the badloans_monitor_3 Data Set*

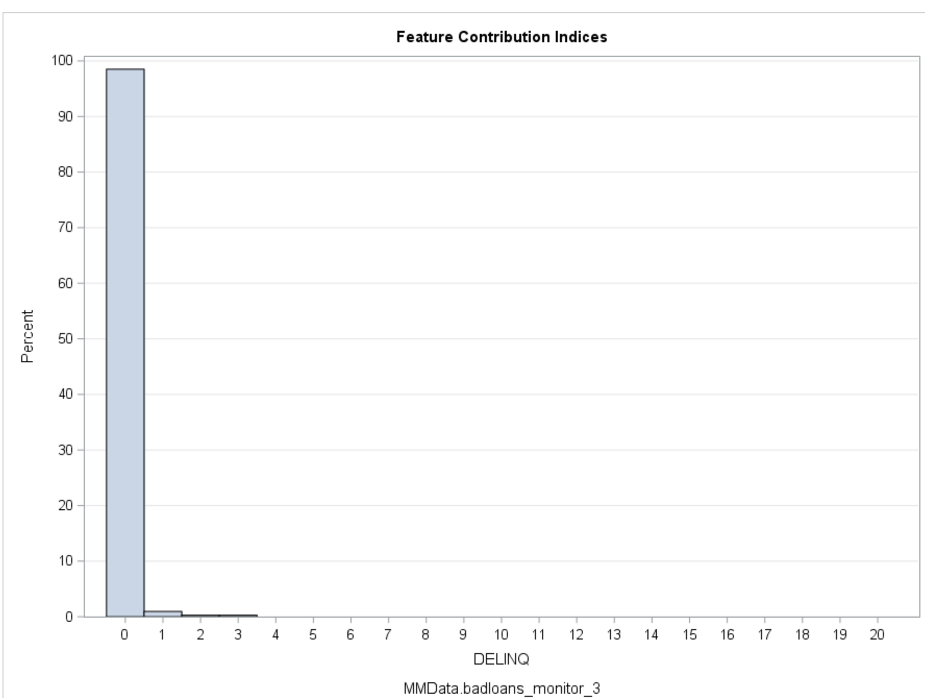


Figure 3.10 *FCI of the DELINQ Predictor Verse Percent of Observations for the badloans_monitor_4 Data Set*

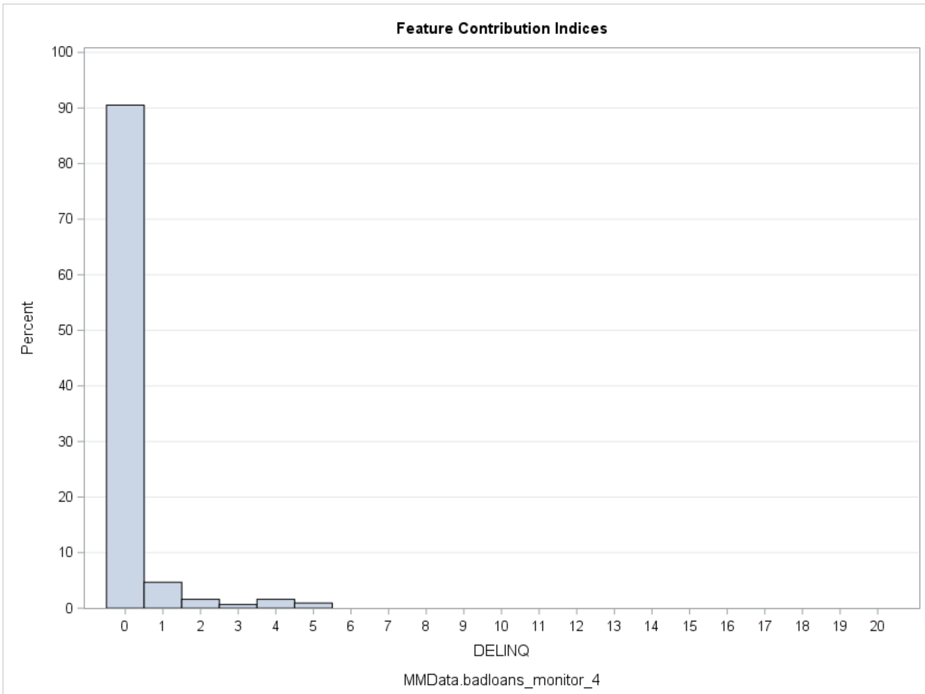


Figure 3.11 *FCI for Interval Predictors*

Feature Contribution Indices

Obs	_VARNAME_	P_BAD1	P_BAD0
1	CLAGE	0.17102	0.17102
2	CLNO	0.00387	0.00387
3	DELINQ	0.33742	0.33742
4	DEROG	0.40186	0.40186
5	MORTDUE	0.02962	0.02962
6	NINQ	0.24030	0.24030
7	YOJ	0.04201	0.04201

MMData.badloans_monitor_4

Figure 3.12 *FCI for Nominal Predictors*

Feature Contribution Indices

Obs	_VARNAME_	P_BAD0	P_BAD1
1	REASON	0.006160	0.006160
2	JOB	0.065304	0.065304

MMData.badloans_monitor_4

Example 2: Running FCI Macros Using SAS Model Manager

Overview

Classification models can be used to predict the likelihood of default of loan applications. These models implement the logistic regression algorithm that computes this probability. In this example, the algorithm is trained using the badloans_train data set. The target variable is BAD and has two values (0 and 1). If a loan application results in default, BAD = 1 (otherwise, BAD = 0). After assessing business value and satisfying legal requirements, consider the following predictors:

- Interval predictors:
  - CLAGE
  - CLNO
  - DELINQ
  - DEROG
  - MORTDUE
  - NINQ
  - YOJ
- Nominal predictors:
  - JOB
  - REASON

The final logistic model includes all of these predictors with the exception of REASON. This model is then used to score (that is, calculate the probability of) incoming observations regarding loan default according to the deployment schedule (for example, monthly, weekly, daily, or hourly).

The incoming observations are collected in these data sets:

- badloans_monitor_1
- badloans_monitor_2
- badloans_monitor_3
- badloans_monitor_4

Because the financial profiles of loan applicants change constantly, the classification must be regularly maintained in order to ensure prediction accuracy. On the other hand, the model should not capture any spurious signals that are due to temporary fluctuations of the financial market. Thus, after each data set is scored, the FCIs of the predictors are calculated and the indices are compared visually across the four data sets. The indices of the training data set are also calculated as a benchmark. If the index of a predictor substantially increases or decreases at a particular time, this indicates that the predictor might contribute to the model more or less than the normal level. This change might be due to an unexpected variation of the predictor's distribution at that time. If the model is refreshed, the findings can help support a particular decision. For more information, see "%MM_AdHocReport_FCI Macro" on page 8.

Import and Score Models

After you have used your preferred tool to build the logistic regression model, you must import and score your model using the SAS Model Manager web application before you can run the %MM_AdHocReport_FCI macro. Sample scoring data sets are included with the macros in the archive file, which is available on the SAS Model Manager Downloads page on support.sas.com.

For more information, see "Importing Models" in *SAS Model Manager: User's Guide* and "Scoring Models" in *SAS Model Manager: User's Guide*.

Create an Ad Hoc Report

1. On the **Reports** page of your project, create an ad hoc report, and name it **BadLoan_FCI_Report**.
2. Select the model that you previously imported.
3. After copying the following SAS code into the SAS Editor, click **Run**.

```
/* Load the custom SAS macro code for calculating the FCIs. */
%let MyLib = %str(c:\FCI);

%include "&MyLib.\Macros\MM_AdHocReport_FCI.sas";
%include "&MyLib.\Macros\Compute_FCI.sas";
%include "&MyLib.\Macros\Compute_FCI_IntPred.sas";
%include "&MyLib.\Macros\Compute_FCI_NomPred.sas";

libname TESTLIB "&MyLib.\Data";

/* Specify the folders that the tables and the charts are to be written to. */
ods html path = "&MyLib." gpath = "&MyLib.";

/* Load the macros inside this catalog in order to display the report */
filename mmreport catalog "sashelp.modelmgr.reportexportmacros.source";
%include mmreport;
%mm_exportReportsBegin (
   fileName = BadLoan_FCI_Report,
   reportFormat = HTML,
   reportstyle = seaside);

/* Create reports for the four data sets with scores. */
/* Include missing values for nominal predictors in the calculations. */
%MM_AdHocReport_FCI (
   ScoreDataPrefix = %str(TESTLIB.BadLoans_Monitor_Scoring_),
   ScoreDataCount = 4,
   P_VarList = %str(P_BAD0 P_BAD1),
   QMissNomPred = N,
   Debug = N);

%mm_exportReportsEnd(reportFormat = HTML);
/* End of the ad hoc report. */
```

For more information, see "Ad Hoc Reports" in *SAS Model Manager: User's Guide*.

Output Results

After you have successfully run the ad hoc report code, open the report to review the results. The target specification data set is printed below. The %MM_AdHocReport_FCI macro assigns equal priors for the values that are the reciprocal of the number of model outcome variables (which is 2 in this example).

Figure 3.13 *Target Specification Data Set and FCI Measurement Levels for Internal and Nominal Predictors*

Model Outcome Variables and Their Priors

Obs	NAME	PRIOR
1	P_BAD0	0.5
2	P_BAD1	0.5

Feature Contribution Indices

		Monitor Time			
		1	2	3	4
Measurement Level	Predictor				
INTERVAL	CLAGE	0.9%	10.7%	16.6%	17.1%
	CLNO	0.7%	0.0%	5.6%	0.4%
	DEBTINC	0.0%	0.3%	1.1%	0.0%
	DELINQ	47.5%	35.9%	8.6%	33.7%
	DEROG	31.3%	28.6%	30.8%	40.2%
	LOAN	0.5%	1.3%	0.8%	0.0%
	MORTDUE	2.6%	3.1%	5.2%	3.0%
	NINQ	13.8%	10.4%	23.7%	24.0%
	VALUE	0.3%	4.3%	7.2%	4.3%
	YOJ	0.1%	1.0%	5.8%	4.2%
NOMINAL	JOB	4.0%	5.5%	9.7%	6.5%
	REASON	0.3%	0.3%	0.0%	0.6%

Figure 3.14 *FCI Plotted Against the Monitoring Time for Each Predictor*

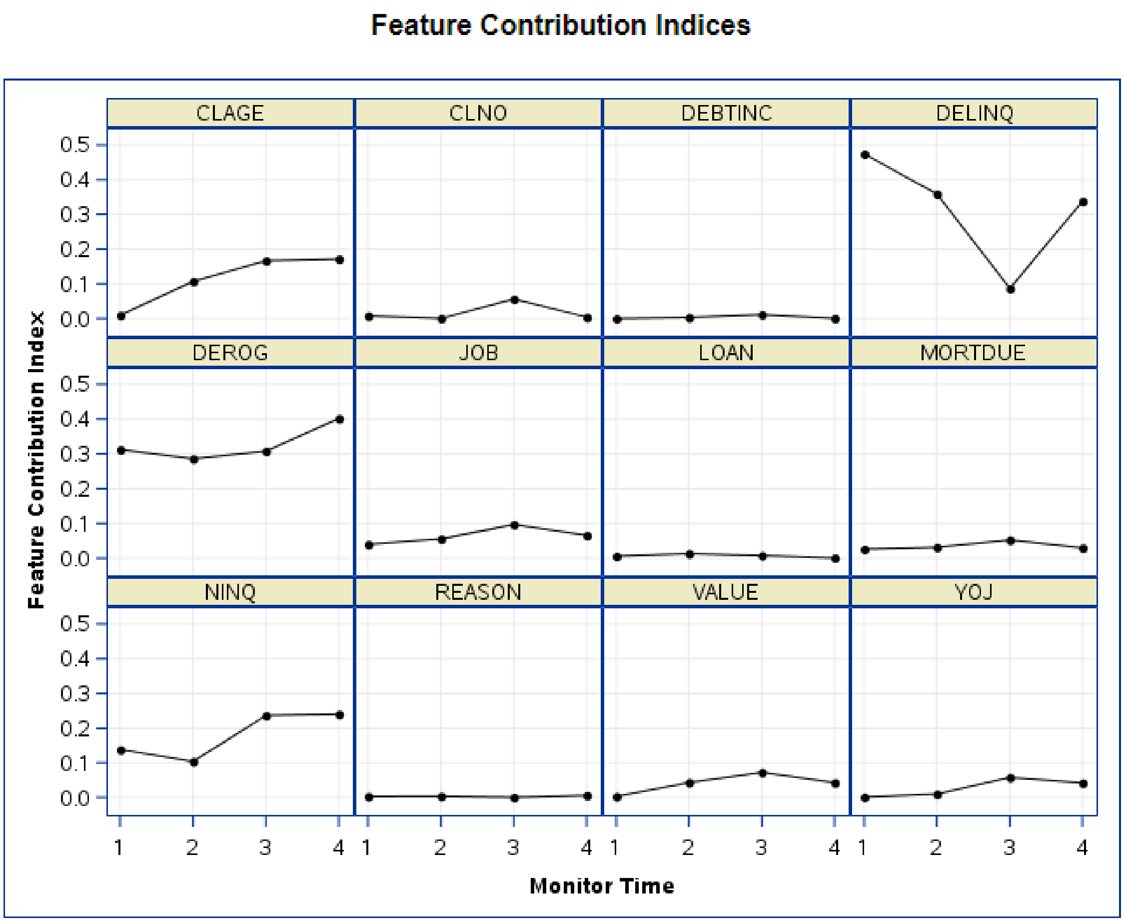
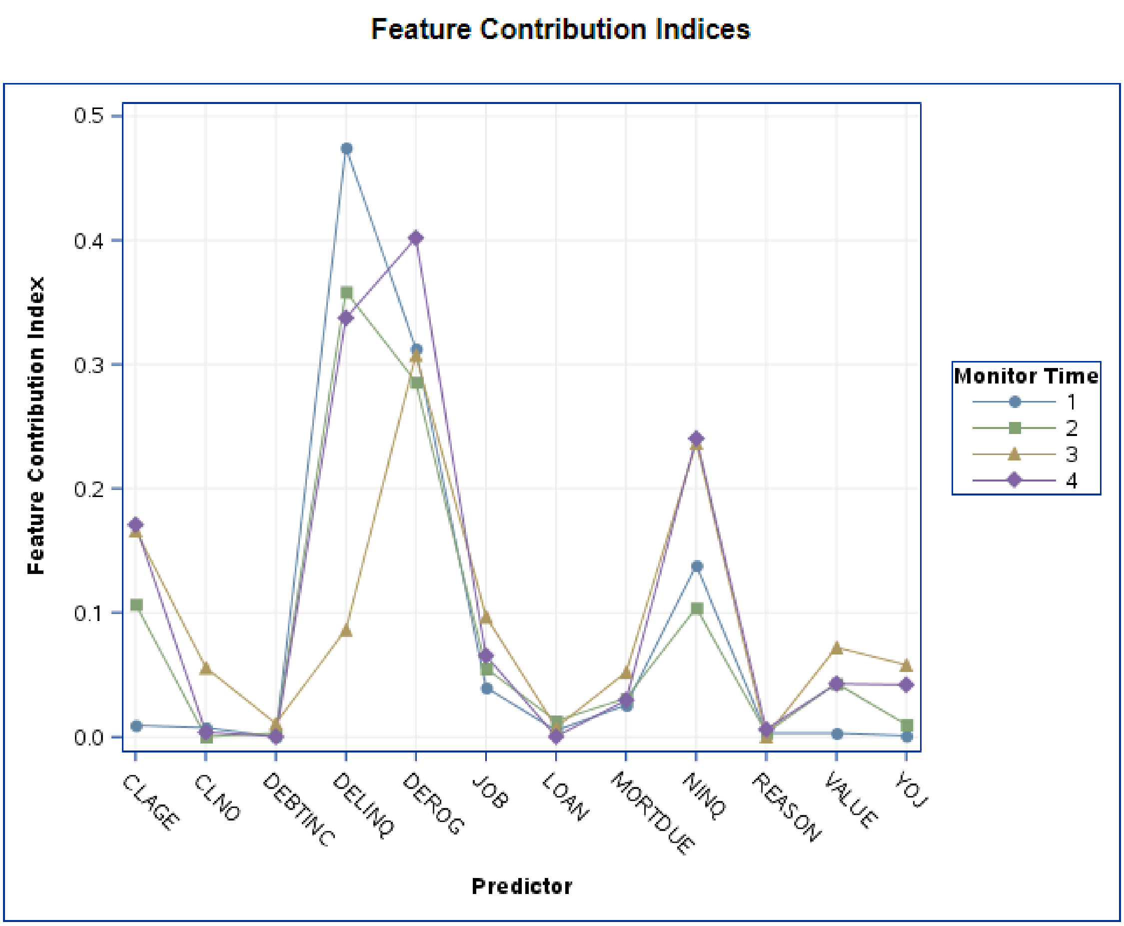


Figure 3.15 *FCI Plotted against Predictors for Each Monitoring Time*



The report indicates that the variable DELINQ contributes relatively less at Monitor Time 3 than at other times. You should act on this finding to compare the distributions of DELINQ across the four times. Your goal is to determine whether the finding is due to spurious fluctuation.