Paper 288-2010

The Applied Use of Population Stability Index (PSI) in SAS® Enterprise Miner™

Rex Pruitt, PREMIER Bankcard, LLC, Sioux Falls, SD

ABSTRACT

In this paper I will describe how to develop the technical components necessary to calculate Population Stability Index (PSI), implement PSI into a SAS® Enterprise Miner™ extension node, and interpret the results of applied PSI analytics as an industry solution "Best Practice".

Why is this of any value? A profound reality exists in the universe: "Change is absolute"!

PREMIER Bankcard's use of predictive modeling has resulted in the need for PSI utilization due to CHANGE experienced in the following areas:

1. Changes in business operations due to internal & external influences

2. Detection of data integrity and/or metadata issues caused by programmatic changes

3. Compliance with Regulatory review requirements

As companies continue to amass large amounts of data and use it to develop statistical models, the PSI measure helps monitor data and scorecard integrity. This is especially important since statistical models are being used to make strategic decisions worth millions of dollars.

INTRODUCTION

In this paper I will describe how to develop the technical components necessary to calculate Population Stability Index (PSI), implement PSI into a SAS® Enterprise Miner™ extension node, and interpret the results of applied PSI analytics as an industry solution "Best Practice".

Why is this of any value? A profound reality exists in the universe: "Change is absolute"!

PREMIER Bankcard's use of predictive modeling has resulted in the need for PSI utilization due to CHANGE experienced in the following areas:

1. Changes in business operations due to internal & external influences

2. Detection of data integrity and/or metadata issues caused by programmatic changes

3. Compliance with Regulatory review requirements

As companies continue to amass large amounts of data and use it to develop statistical models, the PSI measure helps monitor data and scorecard integrity. This is especially important since statistical models are being used to make strategic decisions worth millions of dollars.

HOW PSI WAS INTRODUCED TO PREMIER

The use of Population Stability Index (PSI) was presented to me initially through an inquiry by a co-worker. Apparently, during a review of our internal statistical modeling “best practices”, the Federal Reserve (FED) auditors inquired as to how we validate the continued stability of the components that are used in our models.

Come to find out, this FED inquiry had occurred during their visit in 2009. Subsequently, my peer had attended a SAS statistical training course at M2009. During that course he had asked the instructor to provide some supplemental instruction for the calculation of PSI. Originally, It was hoped that this could be accomplished using an option within Enterprise Miner (EM) v5.3. However, this option is not currently available.

In response to the PSI inquiry, the instructor provided a generic sample set of Base SAS code that could be tailored for our use at PREMIER. Ultimately, I created an abstract version used in the development of a customized extension node called PSI that has been deployed into PREMIER’s EM installation.

WHAT ARE THE TECHNICAL COMPONENTS NECESSARY TO CALCULATE PSI?

The basic premise of PSI is to measure the stability of a specific score and/or variable by comparing a data sample from one time period (Base Data) to another (Target Data). For example, a sample of customer records scored in January as compared to a corresponding representative sample in August.

The idea is to calculate the average percentage of change that has occurred in the sample population when comparing the Base Data distributions to the Target Data distributions. To do so the data must be sorted by the subject Score or Variable of interest. Then the data sample needs to be portioned in appropriate quantile distributions (I.e., Decile=10 Bins; Demi-Decile=20 Bins).

The following “source code” is what was used to create a Decile PSI analysis:

```
/*************************************************************************/
/* This program calculates PSI (Population Stability Index) Statistic    */
/* It was originally sent to PREMIER (Jay Kosters) per his request on    */
/* 4/2/2009.                                                             */
/* Dan Kelly, SAS Institute, provided an example of code with various    */
/* SAS Code options.                                                     */
/* Jay asked Rex to translate the SAS Code and refine it for use by      */
/* PREMIER                                                               */
/* Programming was completed between 4/10 & 4/15, 2009                   */
/*************************************************************************/
/* Dan Kelly's ancillary instructions:                                   */
/* So a few obvious questions that come up are "how do you define the    */
/* buckets" and "how many buckets do I need"?  And "what are sample 1    */
/* and sample 2"?                                                        */
/* If sample 1 and sample 2 are different months (as you have) then you  */
/* just need the bucket definition.                                      */
/*                                                                       */
/* Most of the time I think people use this on the scores, not the       */
/* individual attributes that comprise the score.  There's nothing       */
/* to stop you from testing whether x1 drifts from month to month,       */
/* or x2, or x3, ...                                                     */
/*                                                                       */
/* For the most part when I see people use this they are just looking at */
/* whether the distribution of the score is fairly stable.               */
/*                                                                       */
/* I used 10 buckets just because I like the word "decile";              */
/* often people use "demidecile" for 20 5% buckets.                      */
/*                                                                       */
/* Finally, your cutoffs (.1, .25...) sound like what I usually hear.    */
```

```
/* This statistic is basically (I think) a divergence type statistic,    */
/* like the Information value.  So any cutoff that seems reasonable for  */
/* those types of stats is probably reasonable here as well.             */
/*                                                                       */
/* You can change the distribution of MODELVAR in one of the data sets   */
/* and see what that does to the PSI in the last printout to get a feel  */
/* for what kind of differences in the distribution make what kind of    */
/* difference in the work.                                               */
/*************************************************************************/
/* Per Jay Kosters' research, a score of <= 0.1 indicates little change, */
/* 0.1 - 0.25 is little change but to small to determine and > 0.25 is   */
/* a significant shift.                                                  */
/*************************************************************************/

/*************************************************************************/
/* These Macro variables must be changed to represent the PSI Variable   */
/* (MODELVAR), PSI Output Library (PSILibrary) for storage of the ODS    */
/* Output, Source Data representing the original data file name of the   */
/* population being measured for stability (SourceData1), and the        */
/* current population file name being used to identify possible          */
/* divergence (SourceData2).                                             */
/*************************************************************************/

/*insert the model variable (Interval ONLY) on this line*/
%Let MODELVAR=Receivables;

/*insert the PSI Output Data Library on this line*/
%Let PSILibrary=\\pbidelprd042\DM_Inputs\rpruitt\PSIResults;

/*insert the original population File Name on this line*/
%Let SourceData1=EMWS.Ids_DATA;

/*insert the current population File Name on this line*/
%Let SourceData2=EMWS.Ids4_DATA;

/*************************************************************************/
/* BEGIN Steps to get the data samples for the periods being compared */

LIBNAME PSI "&PSILibrary";

DATA PSI.PSISample1;
SET &SourceData1
             (Keep=&MODELVAR)
             ;
             Format &MODELVAR 12.2;

/*************************************************************/
/* This is where you can place more SAS statements to modify your */
/* PSI Variable so it accurately represents the format and value  */
/* in your model.                                                 */
/*************************************************************/

RUN;

DATA PSI.PSISample2;
SET &SourceData2
             (Keep=&MODELVAR)
             ;
             Format &MODELVAR 12.2;

/*************************************************************/
/* This is where you can place more SAS statements to modify your */
/* PSI Variable so it accurately represents the format and value  */
```

```
/* in your model.                                                    */
/*******************************************************************/

RUN;

/* END Steps to get the data samples for the periods being compared */
/*******************************************************************/

/*********************************/
/*BEGIN establish ODS Output File */

ODS Listing Close;
ODS HTML
       Style=default
       File="&PSILibrary\PSICode&MODELVAR..htm"
       ;
       Title2 "PSI (Population Stability Index) Calculations for &MODELVAR";

/*************************/
/* BEGIN PSI Calculations */

/**********************************/
/* BEGIN break Sample1 into bins    */
/* BEGIN Sorting & Ranking process  */

Proc Means Noprint Data=PSI.PSISample1  ;
       Output
              Out=PSI.RankedTotal (rename=(_freq_=RankedTotal))
              ;
              run;
Data _Null_;
       Set PSI.RankedTotal  (Where=(_Type_=0));
       Call Symput('RankedTotal',RankedTotal);
       run;

Proc Means Noprint Data=PSI.PSISample2;
       Output
              Out=PSI.RankedTotal2 (rename=(_freq_=RankedTotal2))
              ;
              run;
Data _Null_;
       Set PSI.RankedTotal2  (Where=(_Type_=0));
       Call Symput('RankedTotal2',RankedTotal2);
       run;

Proc Sort
       Data=PSI.PSISample1;
       By &MODELVAR;
       run;

Proc Sort
       Data=PSI.PSISample2;
       By &MODELVAR;
       run;

/*********************************************************************/
/*BEGIN Use the Program Data Vector to override the binning of Zero's*/

Data PSI.PSISample1 (Keep=BinVar);
       Set PSI.PSISample1;
       BinVar=Sum(&MODELVAR,(_n_/&RankedTotal));
       run;
```

```
Data PSI.PSISample2 (Keep=BinVar);
       Set PSI.PSISample2;
       BinVar=Sum(&MODELVAR,(_n_/&RankedTotal2));
       run;

/*END Use the Program Data Vector to override the binning of Zero's*/
/*****************************************************************/

Proc Sort
       Data=PSI.PSISample1;
       By BinVar;
       run;

Proc Sort
       Data=PSI.PSISample2;
       By BinVar;
       run;

Proc Format;
       Value DecileF
              Low-0='00'
               0-.1='01'
              .1-.2='02'
              .2-.3='03'
              .3-.4='04'
              .4-.5='05'
              .5-.6='06'
              .6-.7='07'
              .7-.8='08'
              .8-.9='09'
              .9-1='10'
              .='11'
              ;
       Value DemiDecileF
              Low-0='00'
               0-.05='01'
              .05-.1='02'
              .1-.15='03'
              .15-.2='04'
              .2-.25='05'
              .25-.3='06'
              .3-.35='07'
              .35-.4='08'
              .4-.45='09'
              .45-.5='10'
              .5-.55='11'
              .55-.6='12'
              .6-.65='13'
              .65-.7='14'
              .7-.75='15'
              .75-.8='16'
              .8-.85='17'
              .85-.9='18'
              .9-.95='19'
              .95-1='20'
              .='21'
              ;
       Value ZeroMiss
              0='Zero'
              11='Missing'
              21='Missing'
              ;
              run;
```

```
Data PSI.PSISample1;
       Length decile 8.;
       Set PSI.PSISample1;
              Rank=_n_/&RankedTotal;
              Decile=Put(Rank,DecileF.);
       run;

/* END Sorting & Ranking process  */
/* END break Sample1 into 10 bins */
/**********************************/

/*********************************************************************/
/* BEGIN you can see they are 10 equally sized bins with no ties in  */
/* the output of this step.                                          */

proc freq data=PSI.PSISample1;
   tables decile / out=PSI.out1;
Title3 'Base-Line Sample Frequency By Decile Bin (Data=PSISample1)';
run;

/* END you can see they are 10 equally sized bins with no ties in    */
/* the output of this step.                                          */
/*********************************************************************/

/*****************************************************/
/* BEGIN Calculate how the deciles are defined on the */
/*   Supplied Variable (MODELVAR) scale               */
/*   so I want MAX(MODELVAR) in each decile           */

proc means data=PSI.PSISample1 nway;
   class decile;
   var BinVar;
   output out=PSI.endpoints max=maxVar;
Title3 'Base-Line Sample Mean, Max & Min Values (Data=PSISample1)';
run;

/* END   Calculate how the deciles are defined on the */
/*   Supplied Variable (MODELVAR) scale               */
/*   so I want MAX(MODELVAR) in each decile           */
/*****************************************************/

/*************************************************************************/
/* BEGIN Data Step to write code that applies the above decile definition to */
/* the data set with MODELVAR on it                                      */

data _NULL_;
   set PSI.endpoints end=last;
   file "&PSILibrary\decileSample1.sas";
   if _N_ = 1 then put "   select;";
   put "      when (BinVar le " maxVar ") decile = " decile ";" ;
   if last then do ;
      put "      otherwise decile = " decile ";" ;
         put "end;";
       call symput('maxbin',decile);
   end;
run;

data PSI.PSISample2;
   set PSI.PSISample2;
   %inc "&PSILibrary\decileSample1.sas" / source;
       If BinVar=. Then decile=&maxbin;
run;
```

```
/* END Data Step to write code that applies the above decile definition to   */
/* the data set with MODELVAR on it                                          */
/*****************************************************************/

/*****************************************************************/
/* BEGIN Use the same definition for the buckets to establish how    */
/* much data falls in each group for the sample 2                    */

proc freq data=PSI.PSISample2;
   tables decile / out=PSI.out2;
Title3 'Current Sample Frequency By Decile Bin (Data=PSISample2)';
run;

/* END Use the same definition for the buckets to establish how      */
/* much data falls in each group for the sample 2                    */
/*****************************************************************/

/*************************************************************************************/
/* BEGIN put the % fields on the same file and calculate the terms that make up PSI */

data PSI.PSICompare;
   merge PSI.out1 PSI.out2(rename=(percent=percent2));
   by decile;
   psi = log(percent/percent2)*(percent-percent2)/100;
run;

proc print data=PSI.PSICompare noobs;
   var dec: per:;
   Format decile ZeroMiss.;
   sum psi;
       Title3 "NOTE: PSI Calc Accomodates the Binning of Zero And Missing";
run;

/* END put the % fields on the same file and calculate the terms that make up PSI */
/*************************************************************************************/

/* END PSI Calculations */
/************************/

ODS _ALL_ Close;
ODS Listing;

/*END establish ODS Output File */
/*******************************/
```

The following, Figures 1-4) are examples of the output generated from the above code. This output is sent to the designated output library of your choice. The output library is a value to be supplied when completing the Enterprise Miner node properties.

The FREQ Procedure

decile	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	308889	10.00	308889	10.00
2	308889	10.00	617778	20.00
3	308889	10.00	926667	30.00
4	308890	10.00	1235557	40.00
5	308889	10.00	1544446	50.00
6	308889	10.00	1853335	60.00
7	308890	10.00	2162225	70.00
8	308889	10.00	2471114	80.00
9	308889	10.00	2780003	90.00
10	308890	10.00	3088893	100.00

Figure 1. PSI Data Sample 1 (Base Data Population)

The MEANS Procedure

decile	N Obs	N	Mean	Std Dev	Minimum	Maximum
1	**308889**	308889	347.8874547	19.1377974	214.0000000	369.0000000
2	**308889**	308889	475.4599646	76.3783618	369.0000000	577.0000000
3	**308889**	308889	608.6659900	14.8718501	577.0000000	632.0000000
4	**308890**	308890	651.3014892	11.1360143	632.0000000	670.0000000
5	**308889**	308889	687.4058837	9.4970027	670.0000000	703.0000000
6	**308889**	308889	715.0972226	6.9341163	703.0000000	727.0000000
7	**308890**	308890	741.6995791	8.8087846	727.0000000	758.0000000
8	**308889**	308889	781.7008408	15.3237272	758.0000000	812.0000000
9	**308889**	308889	860.9405935	31.1496111	812.0000000	917.0000000
10	**308890**	308890	965.0268736	25.0260635	917.0000000	1000.00

Figure 2. Proc Means output used to determine the Min & Max ranges for the Base Data Population

The FREQ Procedure

decile	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	337109	10.97	337109	10.97
2	243722	7.93	580831	18.89
3	244567	7.96	825398	26.85
4	305093	9.92	1130491	36.78
5	334140	10.87	1464631	47.65
6	344115	11.19	1808746	58.84
7	329769	10.73	2138515	69.57
8	321560	10.46	2460075	80.03
9	306128	9.96	2766203	89.99
10	307817	10.01	3074020	100.00

Figure 3. PSI Data Sample 2 (Current Data Population)

PSI (Population Stability Index)
The PRINT Procedure

Obs	decile	PERCENT	percent2	psi
1	1	10.0000	10.9664	0.000892
2	2	10.0000	7.9284	0.004809
3	3	10.0000	7.9559	0.004674
4	4	10.0000	9.9249	0.000006
5	5	10.0000	10.8698	0.000725
6	6	10.0000	11.1943	0.001347
7	7	10.0000	10.7276	0.000511
8	8	10.0000	10.4606	0.000207
9	9	10.0000	9.9586	0.000002
10	10	10.0000	10.0135	0.000000
			PSI =	**0.013173**

Figure 4. PSI Calculation Result

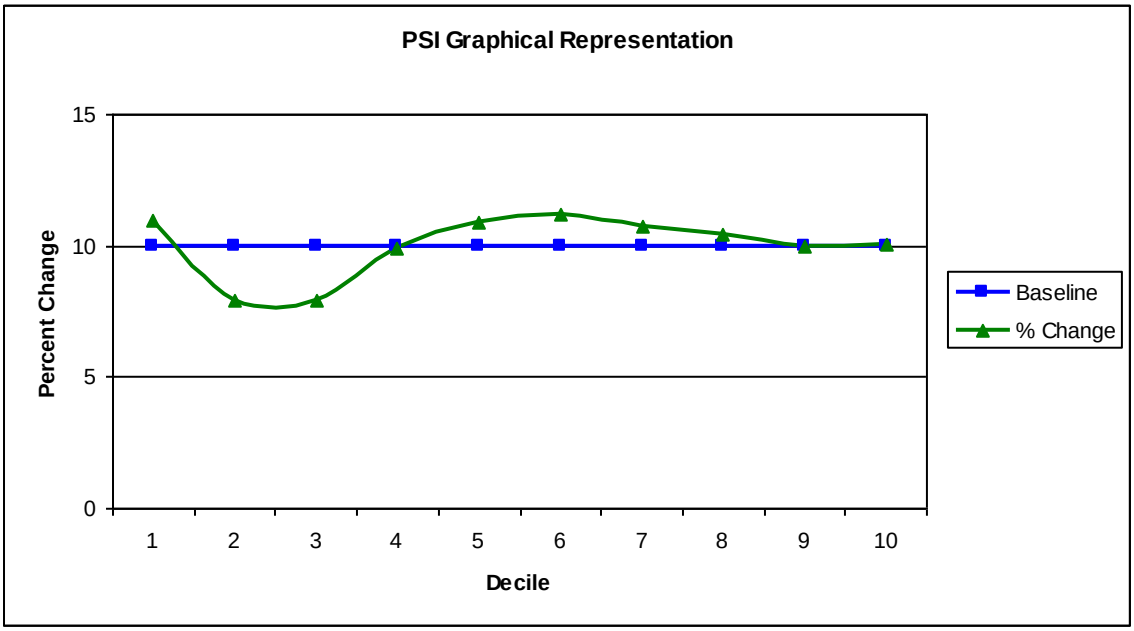

Figure 5. PSI Results Plot

IMPLEMENTING PSI INTO A SAS® ENTERPRISE MINER™ EXTENSION NODE

Once the PSI Base SAS code had been developed, I was able to deploy it into a SAS® Enterprise Miner™ extension node. To do so, I followed the instructions found at:

http://support.sas.com/documentation/onlinedoc/miner/developguide53.pdf.

Please note, the instructions for deployment are extremely installation specific. I will not be elaborating on the details of this procedure. However, in the interest of incorporating the KISS principle for our installation, I did not follow the exact procedure. Specifically, I chose to deploy the SAS code to a library that I am authorized to access edit. For server deployment, this will allow me to revise the code without involving our IT support area. Conversely, the recommended server installation requires intervention by the IT Server Administrators.

INTERPRETING THE RESULTS OF APPLIED PSI ANALYTICS AS AN INDUSTRY SOLUTION "BEST PRACTICE"

As noted within the context of the source code (earlier), the interpretation of the results of applied PSI analytics is simply the assessment of the resulting PSI calculation in 3 category ranges.

<= 0.1 indicates little change [*no action required*]

0.1 - 0.25 is little change but to small to determine [*still no action required*]

> 0.25 is a significant shift [***action required*** *- merits further investigation*]

Depending upon the result, there are various actions and treatments that may be chosen. If the result is acceptable, there is obviously no need for further specific action. However, if the result is not acceptable, there will be a need for more detailed analysis depending on whether a score or individual score component was being measured for PSI.

If the target variable is a score and it needs further investigation, it will likely be necessary to perform the PSI measurement on each individual variable component used in the score development. Naturally, if this were not an automated process, calculating the PSI on several individual variable components would be very time consuming. With the PSI node, it is as simple as changing the variable or sample datasets and executing the node again.

In instances where there are sever score variable components, there may be only one variable that is causing the problem. For example, there may have been a data problem that has gone undetected during the period of time being measured. Another possibility is the advent of an unexpected macro-economic event that may have distorted the population distribution. Or, there may have been a change to an existing business process that directly impacted the variable values. An example of this might be a change in how data is collected. Maybe household income is now collected as a range value instead of a literal value. Metadata changes could create population instability as well.

Regardless of the cause, the PSI calculation serves as a "Best Practice" for catching deteriorating population stability. This is very important as more companies integrate scoring models into their decision management processes.

CONCLUSION

PREMIER Bankcard's use of predictive modeling has resulted in the need for PSI utilization due to CHANGE experienced in the following areas:

1. Changes in business operations due to internal & external influences

2. Detection of data integrity and/or metadata issues caused by programmatic changes

3. Compliance with Regulatory review requirements

As companies continue to amass large amounts of data and use it to develop statistical models, the PSI measure helps monitor data and scorecard integrity. This is especially important since statistical models are being used to make strategic decisions worth millions of dollars.

REFERENCES

http://support.sas.com/documentation/onlinedoc/miner/developguide53.pdf

ACKNOWLEDGMENTS

Jay Kosters, PREMIER Bankcard – For asking me to help him with calculating PSI on his scoring model variables
Dan Kelly, SAS Institute – For providing the sample code used to begin the PSI development at PREMIER

RECOMMENDED READING

http://support.sas.com/documentation/onlinedoc/miner/developguide53.pdf

Contact Information

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

Rex Pruitt
PREMIER Bankcard
PO Box 5114; Mail Drop #113
3820 N. Louise Ave.
Sioux Falls, SD 57117-5114
(605) 575-9810 - Office
(605) 575-9866 - Fax
rpruitt@premierbankcard.com