

Paper 095-213

Variable Reduction in SAS by Using Weight of Evidence and Information Value

Alec Zhixiao Lin, PayPal, a division of eBay, Timonium, MD

ABSTRACT

Variable reduction is a crucial step for accelerating model building without losing potential predictive power of the data. This paper provides a SAS macro that uses Weight of Evidence and Information Value to screen continuous, ordinal and categorical variables based on their predictive power. The results can lend useful insights to variable reduction. The reduced list of variables enables statisticians to quickly identify the most informative predictors for building logistic regression models.

INTRODUCTION

The growing plethora of consumer data is a promise and a challenge for statisticians. Data can be mined to extract valuable business intelligence, but sifting through hundreds – and sometimes thousands - of variables could be a daunting task.

Various methods have been developed for reducing the number of variables to a manageable size. Factor Analysis and Principal Component Analysis are two commonly used tools. Many statisticians also employ such options as STEPWISE, FORWARD and BACKWARD in PROC LOGISTIC to screen variables before deciding which ones should enter a final regression.

In this paper, the author provides a SAS macro that will rank order the predictive power of all variables by their Information Value (IV) and Weight of Evidence (WOE). At the end of the program, several summary reports and graphs will be generated to offer useful suggestions for variable reduction. One advantage of this program is that continuous, ordinal and categorical variables can be assessed together. Moreover, nonlinear behavior of some variables in relation to the targeted outcome is also to be revealed.

CORE CONCEPTS

In recent years WOE and IV have been receiving increasing attention from various sectors beyond scorecard development for credit risk. The author has found both measures to be extremely useful in reducing variables when building such models as response to marketing campaigns, recovery rate of delinquent accounts, etc..

The following table illustrates how Weight of Evidence is computed. The targeted outcome in this example is the likelihood of incoming calls to customer service, based on customers' historical transaction frequency.

Purchase Count In Past 12 Months	# Customers	% Customers	# Caller	# nonCaller	% Caller	% nonCaller	WOE	% Customers who have called in
0	20,070	20.1%	438	19,632	9%	20.6%	-82.67	2.2%
1-3	21,260	21.3%	596	20,664	12%	21.7%	-56.98	2.8%
4-10	12,394	12.4%	457	11,937	9%	12.5%	-28.66	3.7%
11-25	8,744	8.7%	350	8,393	7%	8.8%	-20.01	4.0%
26-40	6,920	6.9%	305	6,615	6%	7.0%	-10.18	4.4%
41-60	5,613	5.6%	314	5,299	6%	5.6%	15.08	5.6%
61-100	9,886	9.9%	651	9,235	13%	9.7%	32.42	6.6%
100-200	7,234	7.2%	575	6,659	12%	7.0%	52.69	7.9%
>200	7,879	7.9%	1,165	6,714	24%	7.1%	122.49	14.8%
Total	100,000	100.0%	4,852	95,148	100%	100.0%		4.9%

% Caller: number of callers of the tier/number of all callers
 % nonCaller: number of noncallers of the tier/all noncallers

The formula of Weight of Evidence is as the following:

$$WOE = \left[\ln \left(\frac{\%Caller_i}{\%nonCaller_i} \right) \right] \times 100$$

As seen from the formula, Weight of Evidence takes into consideration not only how the call rate (% of customers who have called) trends with the purchase count, but also how purchase count is distributed across the entire spectrum of the customer portfolio.

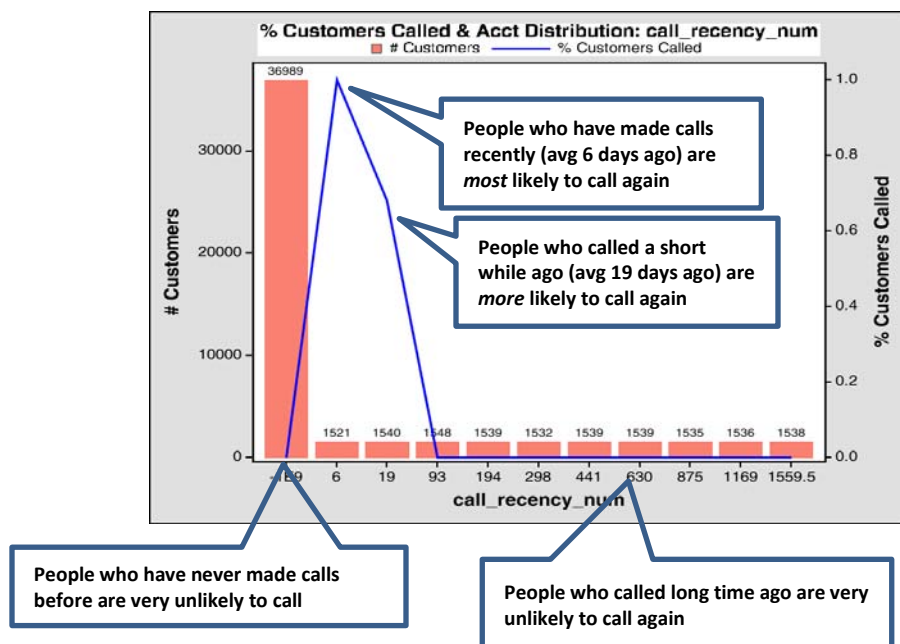
While Weight of Evidence analyzes the predictive power of a variable in relation to the targeted outcome, Information Value assesses the overall predictive power of the variable being considered, and therefore can be used for comparing the predictive power among competing variables. The following is how Information Value is calculated:

$$IV = \sum_{i=1}^{\infty} \left((\%Caller_i - \%nonCaller_i) \times \ln\left(\frac{\%Caller_i}{\%nonCaller_i}\right) \right)$$

In our example, the IV for the variable of Purchase Count in Past 12 Months is 0.41.

Compared with other methods of variable reduction, Weight of Evidence and Information IV have two advantages:

- They can be used to assess the predictive power of categorical, ordinal and continuous together;
- The SAS macro provided in this paper includes the calculation of Weight of Evidence for the missing values of each variable. The result can lend useful insights to variable imputation in subsequent regression analysis;
- The program also captures a nonlinear relationship between a predictor and a target outcome, albeit short of providing optimal divides for binning these variables. The following figure illustrated how call recency impacts call rate.



THE SAS PROGRAM

Before running the program, do the following to the data set:

- Run PROC MEANS to screen out variables with zero coverage;
- Categorical and ordinal variables expressed in character should be converted to numeric indicators first. For example, assign 1, 2, 3 to nominal values of A, B and C, and so on;
- Make sure that the list of variables contains no duplicates.

The following is the beginning part of the program. One can make needed changes to underlined libname, file names, variable names and variable list.

```
libname dataloc 'the designated folder where the SAS data set is stored';
%let inset=model_sample;                               /* data set name */
%let target=called;                                     /* target variable (y) */
%let libout=C:/output folder;                          /* designated folder for exporting outputs */
%let varall= var1 var2 var3 ..... varn;                /* list of variables to be assessed.
                                                         Duplicates are not allowed here. */
%let tiermax=10;                                         /* max number of bins to assign to variables */
```

```

%let ivthresh=0.1; /* set to 0 if you want to see output graphs for all variables */
%let outgraph=iv_woe_graph; /* pdf graph for top predictors */
%let ivout=iv_ranked; /* output file in txt for Information Value */
%let woeout=woe_ranked; /* output file in txt for Weight of Evidence */
%let libdata=dataloc; /* name of library where the data set is stored. */
%let outcome=pct_cust_called; /* name of target for summary tables */
%let outname=% Customers Called; /* label of target for summary tables and graphs */
(See subsequent part of the program in Appendix)

```

THE OUTPUT

The program generates three output summaries in a designated folder: a flow table of Information Values for all variables, a table of Weight of Evidence for all variables, and a PDF collage of graphs for variables whose Information Values are above a specified threshold (0.1 in the SAS example above).

- 1) A flow table of Information Value for all variables, sorted in a descending order.

Variable	Information Value	% Records with missing value	IV Rank
Sales Count In Most Recent Months	0.6442	91%	1
Average Volume Per Sale In 6 Months	0.5952	81%	2
% Of Purchase Count Over Total Transaction Count In 3 Months	0.5538	36%	3
Purchase Volume Over Total Transaction Volume In 12 Months	0.5171	1%	4
Number Of Negative Balances Apparing On Sellers Acct In 12 Month:	0.5041	82%	5
Purchase Count In 3 Months	0.4339	36%	6
Purchase Count In 6 Months	0.4085	20%	7
Number Of Canceled Purchase In 12 Months	0.4044	72%	8
Average Monthly Purchase Count In 12 Months	0.3930	0%	9
Customer Type (Personal Or Business)	0.3712	0%	10
Average Purchase Volume In Most Recent Month	0.3683	57%	11
Average Monthly Purchase Amount In 12 Months	0.3571	0%	12
Number Of Payment Withdrawl In 6 Months	0.3497	88%	13
% Of Purchase Count In Most Recent Month Over 12 Months	0.3000	57%	14
Average Volume Per Purchase In 12 Months	0.1402	1%	15
Number Of Checking Account Added In 12 Months	0.1368	86%	16
Purchase Count By Check In Most Recent Month	0.1128	82%	17
% Of Denials Of Purchase In Most Recent Months Over 12 Months	0.0706	98%	18
Number Of Credit Cards Added In 12 Months	0.0689	49%	19

This table provides a very straightforward view for comparing Information Values across all variables. Variable of the same Information Value will be tied in IV Rank.

Variables of similar natures usually behave similarly and have very high correlation with each other. The flow table of Information Value exhibits how these variables cluster together. In the current example, Purchase Count In Past 6 Months offers a very similar Information Value as Purchase Count In Past 3 Months. As a regression prefers variables of lower dependence from one another, only one variable from each cluster needs to be chosen to enter the regression.

Siddiqi suggests that variables with extremely high IVs invite suspicion. He provides the following rule of thumb for approaching variables based on their Information Value :

< 0.02: unproductive
 0.02 to 0.1: weak
 0.1 to 0.3: medium
 0.3 to 0.5: strong
 > 0.5: suspicious

In the current example, Sales Count In Most Recent Months has an Information Value of 0.6442, which means customers have made online sales very recently are more likely to call service representative. This observation speaks for common sense, but if the score to be developed is intended for credit offering, one might ask the following question: should credit-worthiness be associated with sales recency of a customer?

- 2) A flow table of Weight of Evidence for all variables, ranked in descending order by Information Value and then by variable tiers from lowest to highest.

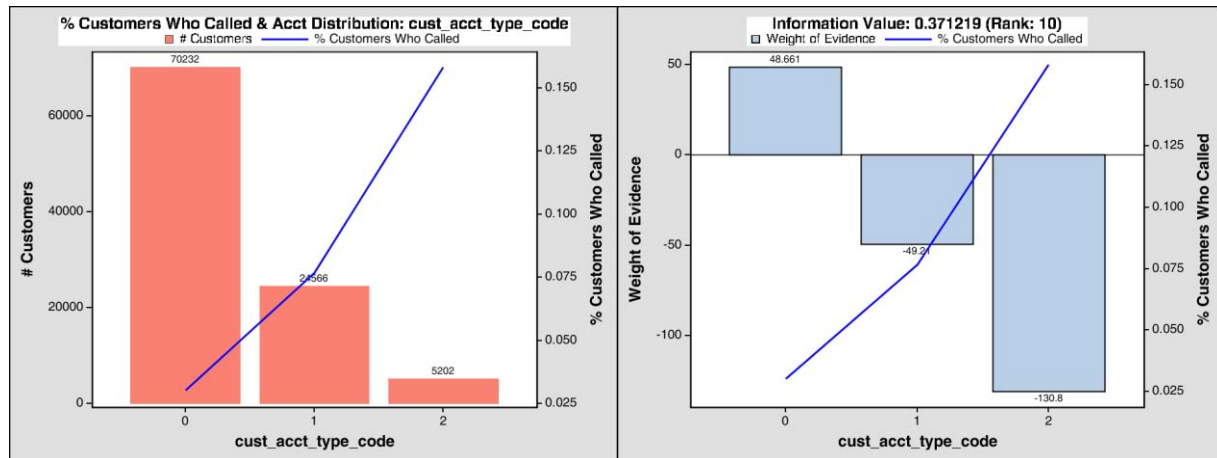
While preserving all contents in the flow table of Information Value, the flow table of Weight of Evidence expands to tiers within each variable, with WOE provided for each tier.

Variable	IV	IV Rank	Median Value for the bin	# Customers	% Customers	# Callers	# nonCallers	% Callers	% nonCallers	Weight of Evidence	% Customers who called	% Records with missing value
sales count in most recent months	0.6442	1	.	91111	91.1%	88055	3056	92.5%	63.2%	38.11	3.4%	91.1%
sales count in most recent months	0.6442	1	1	3846	3.8%	3356	490	3.5%	10.1%	-105.56	12.7%	91.1%
sales count in most recent months	0.6442	1	2	1419	1.4%	1157	262	1.2%	5.4%	-149.45	18.5%	91.1%
sales count in most recent months	0.6442	1	3	837	0.8%	673	164	0.7%	3.4%	-156.79	19.6%	91.1%
sales count in most recent months	0.6442	1	4	1117	1.1%	849	268	0.9%	5.5%	-182.67	24.0%	91.1%
sales count in most recent months	0.6442	1	6	681	0.7%	494	187	0.5%	3.9%	-200.83	27.5%	91.1%
sales count in most recent months	0.6442	1	10	989	1.0%	581	408	0.6%	8.4%	-262.62	41.3%	91.1%
average volume per sale in 6 months	0.5952	2	.	81025	81.0%	78674	2351	82.7%	48.6%	53.07	2.9%	81.0%
average volume per sale in 6 months	0.5952	2	8.33	1897	1.9%	1749	148	1.8%	3.1%	-51.01	7.8%	81.0%
average volume per sale in 6 months	0.5952	2	22.82	1898	1.9%	1728	170	1.8%	3.5%	-66.08	9.0%	81.0%
average volume per sale in 6 months	0.5952	2	37.50	1897	1.9%	1721	176	1.8%	3.6%	-69.96	9.3%	81.0%
average volume per sale in 6 months	0.5952	2	55.00	1898	1.9%	1703	195	1.8%	4.0%	-81.26	10.3%	81.0%
average volume per sale in 6 months	0.5952	2	78.13	1897	1.9%	1667	230	1.8%	4.8%	-99.90	12.1%	81.0%
average volume per sale in 6 months	0.5952	2	107.47	1898	1.9%	1663	235	1.7%	4.9%	-102.29	12.4%	81.0%
average volume per sale in 6 months	0.5952	2	152.19	1897	1.9%	1607	290	1.7%	6.0%	-126.75	15.3%	81.0%
average volume per sale in 6 months	0.5952	2	220.00	1898	1.9%	1599	299	1.7%	6.2%	-130.30	15.8%	81.0%
average volume per sale in 6 months	0.5952	2	353.34	1921	1.9%	1582	339	1.7%	7.0%	-143.93	17.6%	81.0%
average volume per sale in 6 months	0.5952	2	872.34	1874	1.9%	1472	402	1.5%	8.3%	-168.18	21.5%	81.0%

(Only partial table is pasted above)

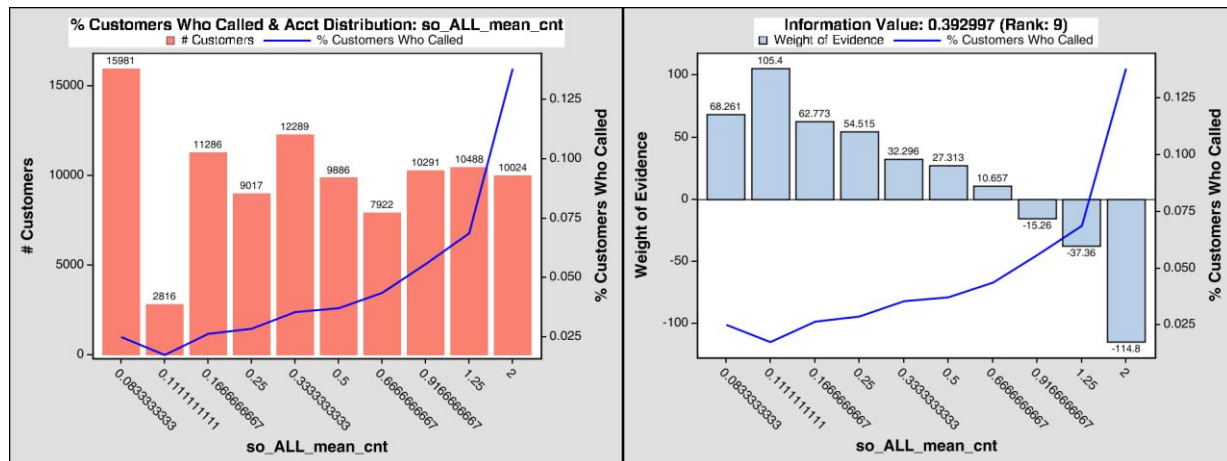
3) The bar chart for Weight of Evidence in the above table has been hand-coded, but the SAS program generates a PDF collage of Weight of Evidence for all variables above a specified threshold of Information Value (0.1 in this example). If one wants to see the charts for all variables, simply set the macro variable `ivthresh=0`. Besides providing a good visual aid for variable reduction, the graphs also offers useful suggestions for imputing missing values for variables of interest.

The following two charts illustrate how frequency of incoming calls is associated with account type (0=personal, 1=premier, and 2=business):



The left chart presents account distribution by account type and the call rate. The right chart gives the Weight of Evidence for each account type (call rate is also preserved here).

For a continuous variable, x-axis indicates the median value for each tier. The following two charts show how payment count trends with the call rate:



For the subsequent logistic regression, the author considers the following criteria for selecting potential predictors of similar Information Value to enter the final logistic regression:

- Magnitude of Information Value. Variables with higher Information Values are preferred;
- Linearity. Variables exhibiting a strong linearity are preferred;
- Coverage. Variables with fewer missing values are preferred;
- Distribution. Variables with less uneven distributions are preferred;
- Interpretation. Variables that make more sense in business are preferred.

Processing a large number of variables usually runs into the problem of insufficient system memory. To partially counter this problem, this SAS program automatically divides the data set into ten partitions for separate processing and then sums the results into an output. If the issue of insufficient memory is still unresolved, the following tips will help:

- Run the SAS program on a smaller sample. The order of Information Value will change slightly, but it does not disrupt the general pattern by all variables;
- Delete those variables with extremely low coverage;
- Run separate programs, each time for a subset of all variables. For example, if 1000 variables need to be assessed, you can process 300, 300 and 400 variables in three separate jobs. After all outputs have been generated, you can consolidate a list of top 200 variables with the highest Information Values and rerun the program for these variables only.

CONCLUSION

Information Value is a useful tool to compare the predictive power across different variables. Weight of Evidence paints furthermore how each variable behaves. The SAS program included in this paper has been tested repeatedly by multiple statisticians and proven to work well. It lends an additional tool to the SAS inventory of methods for variable reduction.

ACKNOWLEDGEMENTS

I would like to thank David Zhu, Ren Li, Fan Hu and Shawn Benner for their inspiration and for providing valuable feedbacks, which have helped me to improve the designed process.

REFERENCES

Siddiqi, Naeem (2006). *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. SAS Institute, pp 79-83.

CONTACT INFORMATION

Alec Zhixiao Lin
Principal Statistical Analyst
PayPal, a division of eBay
9690 Deereco Road

Timonium, MD 21093
 Phone: 703-593-4290
 Fax: 443-921-1985
 Email: alecindc@gmail.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® Indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.

APPENDIX

The following is the entire SAS program:

```
libname dataloc 'the designated folder where the SAS data set is stored';
%let inset=model sample;                                /* data set name */
%let target=called;                                        /* target variable (y) */
%let libout=C:/output folder;                             /* folder for export outputs */
%let varall= var1 var2 var3 ..... varn;                  /* list of variables to be assessed */
%let tiermax=10;                                           /* max number of bins to assign to variables */
%let ivthresh=0.1;    /* set to 0 if you want to see output graphs for all variables */
%let outgraph=iv_woe_graph;                               /* pdf graph for top predictors */
%let ivout=iv_ranked;                                     /* output file in txt for Information Value */
%let woeout=woe_ranked;                                   /* output file in txt for Weight of Evidence */
%let libdata=dataloc;                                     /* name of library where the data set is stored. */
%let outcome=pct cust called;                             /* name of target for summary tables */
%let outcome=% Customers Called;                          /* label of target for summary tables and graphs */
***** Changes are needed for underlined codes above only;
```

*****There is no need to change the following part of the program;

```
ods output nlevels=checkfreq;
proc freq data=&libdata.&inset nlevels;
tables &varall/noprint;
run;
ods output close;

data varcnt;
set checkfreq;
varcnt+1; run;

proc univariate data=varcnt;
var varcnt;
output out=pctscore pctlpts=0 10 20 30 40 50 60 70 80 90 100
pctlpre=pct_;
run;

data _null_;
set pctscore;
call symputx('start1', 1);
call symputx('end1', int(pct_10)-1);
call symputx('start2', int(pct_10));
call symputx('end2', int(pct_20)-1);
call symputx('start3', int(pct_20));
call symputx('end3', int(pct_30)-1);
call symputx('start4', int(pct_30));
call symputx('end4', int(pct_40)-1);
call symputx('start5', int(pct_40));
call symputx('end5', int(pct_50)-1);
call symputx('start6', int(pct_50));
call symputx('end6', int(pct_60)-1);
call symputx('start7', int(pct_60));
```

```

call symputx('end7', int(pct_70)-1);
call symputx('start8', int(pct_70));
call symputx('end8', int(pct_80)-1);
call symputx('start9', int(pct_80));
call symputx('end9', int(pct_90)-1);
call symputx('start10', int(pct_90));
call symputx('end10', pct_100);
run;

** get some important macro values;
** rename the variables;

** select variables with less than needed number of tiers such as 10 in this example;
proc sql;
select tablevar
into :varmore separated by ' '
from varcnt
where nlevels > &tiermax; quit;

proc sql;
create table vcnt as select count(*) as vcnt
from varcnt where nlevels > &tiermax; quit;

data _null_;
set vcnt;
call symputx('vmcnt', vcnt); run;

proc sql;
select tablevar
into :v1-:v&vmcnt
from varcnt
where nlevels > &tiermax; quit;

proc sql;
select max(varcnt), compress('&x'||put(varcnt, 10.))
into :varcount, :tempvar separated by ' '
from varcnt
order by varcnt;
quit;

proc sql;
select tablevar
into :x1-:x&end10
from varcnt; quit;

proc sql;
select count(*)
into :obsCnt
from &libdata..&inset; quit;

%macro stkorig;
%do i=1 %to &vmcnt;
data v&i;
length tablevar $32.;
set &libdata..&inset(keep=&v&i rename=(&v&i=origvalue));
tablevar="&v&i";
format tablevar $32.;

attrib _all_ label='';

```



```

run;

proc rank data=v&i groups=&tiermax out=v&i;
by tablevar;
var origvalue;
ranks rankvmore;
run;

proc means data=v&i median mean min max nway noprint;
class tablevar rankvmore;
var origvalue;
output out=vmoreranked&i(drop=_type_ _freq_)
      median=med_origv
      mean=mean_origv
      min=min_origv
      max=max_origv;
run;
%end;
%mend;
%stkorig;

data stackorig;
set vmoreranked1-vmoreranked&vmcnt; run;

** make a permanent dataset just in case;
data &libdata..stackorig;
set stackorig; run;

** only rank these variables with more than 10 values;
** the following dataset is for later aggregation in a sas macro;
proc rank data=&libdata..&inset groups=&tiermax out=try_model(keep=&tempvar &target);
var &varmore;
ranks &varmore;
run;

** generate Information Value and Weight of Evidence;
%macro outshell;
%do i=1 %to &varcount;
** count good and bad;

data try_model;
set try_model;
if &&x&i=. then &&x&i=-1000000000;
run;

proc sql;
select sum(case when &target=1 then 1 else 0 end), sum(case when &target=0 then 1 else
0 end), count(*)
      into :tot_bad, :tot_good, :tot_both
from try_model;
quit;

proc sql;
select count(*) into :nonmiss
from try_model
where &&x&i ne -1000000000;
quit;

** compute Weight of Evidence (WoE);

```



```

proc sql;
create table woe&i as
(select "&x&i" as tablevar,
      &x&i as tier,
      count(*) as cnt,
      count(*)/&tot_both as cnt_pct,
      sum(case when &target=0 then 1 else 0 end) as sum_good,
      sum(case when &target=0 then 1 else 0 end)/&tot_good as dist_good,
      sum(case when &target=1 then 1 else 0 end) as sum_bad,
      sum(case when &target=1 then 1 else 0 end)/&tot_bad as dist_bad,
      log((sum(case when &target=0 then 1 else 0 end)/&tot_good)/(sum(case when
&target=1 then 1 else 0 end)/&tot_bad))*100 as woe,
      ((sum(case when &target=0 then 1 else 0 end)/&tot_good)-(sum(case when
&target=1 then 1 else 0 end)/&tot_bad))
      *log((sum(case when &target=0 then 1 else 0
end)/&tot_good)/(sum(case when &target=1 then 1 else 0 end)/&tot_bad)) as pre_iv,
      sum(case when &target=1 then 1 else 0 end)/count(*) as &outcome

      from try_model
      group by "&x&i", &x&i
)
order by &x&i;
quit;

```

```

** compute Information Value (IV);
proc sql;
create table iv&i as select "&x&i" as tablevar,
                          sum(pre_iv) as iv,
                          (1-&nonmiss/&obsct) as pct_missing

from woe&i; quit;
&end;
&mend outshell;

%outshell;

```

```

%macro stackset;
%do j=1 %to 10;
data tempiv&j;
length tablevar $32.;
set iv&&start&j-iv&&end&j;
format tablevar $32.;
run;

data tempwoe&j;
length tablevar $32.;
set woe&&start&j-woe&&end&j;
format tablevar $32.;
run;
&end;
&mend;

%stackset;

data &libdata..ivall; set tempiv1-tempiv10; run;
data &libdata..woeall; set tempwoe1-tempwoe10; run;

proc sort data=&libdata..ivall; by descending iv; run;

data &libdata..ivall; set &libdata..ivall; ivrank+1; run;

```

```

proc sort data=&libdata..ivall nodupkey out=ivtemp(keep=iv); by descending iv; run;

data ivtemp; set ivtemp; ivtier+1; run;

proc sort data=ivtemp; by iv; run;
proc sort data=&libdata..ivall; by iv; run;

data &ivout;
merge &libdata..ivall ivtemp; by iv; run;

proc sort data=&ivout; by tablevar; run;
proc sort data=&libdata..woeall; by tablevar; run;

data &libdata..iv_woe_all;
merge &ivout &libdata..woeall;
by tablevar; run;

proc sort data=&libdata..iv_woe_all; by tablevar tier; run;
proc sort data=&libdata..stackorig; by tablevar rankvmore; run;

data &libdata..iv_woe_all2;
merge &libdata..iv_woe_all(in=t) &libdata..stackorig(in=s rename=(rankvmore=tier));
by tablevar tier;
if t;
if s then tier=med_origv;
run;

proc sort data=&libdata..iv_woe_all2; by ivrank tier; run;

%let retvar=tablevar iv ivrank ivtier tier cnt cnt_pct dist_good dist_bad woe
    &outcome pct_missing;

data &libdata..&woeout(keep=&retvar);
retain &retvar;
set &libdata..iv_woe_all2;

label tablevar="Variable";
label iv="Information Value";
label ivrank="IV Rank";
label tier="Tier/Bin";
label cnt="# Customers";
label cnt_pct="% Custoemrs";
label dist_good="% Good";
label dist_bad="% Bad";
label woe="Weight of Evidence";
label &outcome="&outname";
label pct_missing="% Missing Values";
run;

** examine KS;
proc nparlway data=&libdata..&inset /* specify the input dataset */
    edf noprint;
    var &varall; /* type your list of predictors(x) here */
    class &target; /* target variable such as BAD */
    output out=ks101(keep= _var_ _D_ rename=(_var_=tablevar _D_=ks));
run;

proc sort data=ks101; by tablevar; run;
proc sort data=&ivout; by tablevar; run;

```

```

data &libdata.&ivout;
retain tablevar iv ivrank ivtier ks pct_missing;
merge ks101 &ivout;
by tablevar;
keep tablevar iv ivrank ivtier ks pct_missing;
run;

proc contents data=&libdata.&woeout varnum; run;
proc contents data=&libdata.&ivout varnum; run;

proc sort data=&libdata.&woeout out=&woeout(drop=ivrank rename=(ivtier=iv_rank)); by
ivtier tablevar; run;
proc sort data=&libdata.&ivout out=&ivout(drop=ivrank rename=(ivtier=iv_rank)); by
ivtier; run;

%macro to_excel(data_sum);
PROC EXPORT DATA=&data_sum
          OUTFILE="&libout/&data_sum"
          DBMS=tab REPLACE;
run;
%mend;
%to_excel(&ivout);
%to_excel(&woeout);

proc sql;
select count(distinct ivrank)
into :cntgraph
from &libdata.&ivout
where iv > &ivthresh; quit;

data _null_;
call symputx('endlabel', &cntgraph); run;

** add tier label;
proc sql;
select tablevar, iv
into :tl1-:tl&endlabel, :ivr1-:ivr&endlabel
from &libdata.&ivout
where ivrank le &cntgraph
order by ivrank; quit;

proc template;
define style myfont;
parent=styles.default;
style GraphFonts /
'GraphDataFont'=("Helvetica",8pt)
'GraphUnicodeFont'=("Helvetica",6pt)
'GraphValueFont'=("Helvetica",9pt)
'GraphLabelFont'=("Helvetica",12pt,bold)
'GraphFootnoteFont' = ("Helvetica",6pt,bold)
'GraphTitleFont'=("Helvetica",10pt,bold)
'GraphAnnoFont' = ("Helvetica",6pt)
;
end;
run;

```

```

ods pdf file="%libout/&outgraph..pdf" style=myfont;
%macro drgraph;
%do j=1 %to &cntgraph;
proc sgplot data=&libdata..&woeout(where=(ivrank=&j));
vbar tier / response=cnt nostatlabel nooutline fillattrs=(color="salmon");
vline tier / response=&outcome datalabel y2axis lineattrs=(color="blue" thickness=2)
nostatlabel;
label cnt="# Customers";
label &outcome="&outname";
label tier="&&tl&j";
keylegend / location = outside
position = top
noborder
title = "&outname & Acct Distribution: &&tl&j";
format cnt_pct percent7.4;
format &outcome percent7.3;
run;

proc sgplot data=&libdata..&woeout(where=(ivrank=&j));
vbar tier / response=woe datalabel nostatlabel;
vline tier / response=cnt_pct y2axis lineattrs=(color="salmon" thickness=2)
nostatlabel;
label woe="Weight of Evidence";
label cnt_pct="% Customers";
label &outcome="&outname";
label tier="&&tl&j";
keylegend / location = outside
position = top
noborder
title = "Information Value: &&ivr&j (Rank: &j)";
format cnt_pct percent7.4;
run;
%end;
%mend;
%drgraph;
ods pdf close;

```