连续变量的粗分类代码（宏）

\*Train 训练数据集;

\*Target 目标变量名称;

\*Leafsize 叶子节点最小规模;

\*Bin\_code\_interval 分组代码输出数据集;

\*Bin\_drop\_var 无法分组的变量数据集;

\*Split\_code\_pth 分组代码输出路径;

%macro split(train,target,var\_numeric,minleafsize,bin\_code\_interval,bin\_drop\_var\_numeric,split\_code\_pth);

data \_null\_;

set dm\_in.var\_numeric NOBS=last;

call symput('N\_VAR',last);

stop;

run;

proc sql;

drop table &bin\_code\_interval;

drop table &bin\_drop\_var\_numeric;

quit;

data \_null\_;

set &var\_numeric;

%do i=1 %to &N\_VAR;

if \_N\_=&i then do;

call symput("TRAIN\_VAR\_&i",compress('T\_'||name)||'(keep=&target '||compress(name)||')');

end;

%end;

run;

%macro train\_var;

%do i=1 %to &N\_VAR;

&&TRAIN\_VAR\_&i

%end;

%mend;

data %train\_var;

set &train;

run;

%do i=1 %to &N\_VAR;

data \_null\_;

set &var\_numeric;

if \_N\_=&i then do;

call symput('SPLIT\_VAR',name);

end;

run;

proc split data=T\_&split\_var outtree=tre\_&split\_var leafsize=&minleafsize criterion=chisq excludemiss;

input &SPLIT\_VAR/level=interval;

target &target/level=binary;

run;

proc sort data=tre\_&split\_var(where=(label='<')) out=grp\_&split\_var;

by y;

run;

proc sql noprint;

select count(\*) into:group

from grp\_&split\_var;

quit;

%if &group>0 %then %do;

data cde\_&split\_var;

set grp\_&split\_var end=last;

length code $1000;

retain tmp;

if \_n\_=1 then do;

code="if "||compress("&split\_var.")||"=. then "||compress("grp\_&split\_var=-1;");

grp\_var=-1;

var="&split\_var";

output;

tmp=compress(put(y,24.8));

code="else if .<"||compress("&split\_var.")||"<"||compress(tmp)||" then "||compress("grp\_&split\_var=")||put(\_N\_,5.)||";";

grp\_var=\_N\_;

var="&split\_var";

output;

end;

if \_n\_>1 then do;

code="else if "||compress(tmp)||"<="||compress("&split\_var.")||"<"||compress(put(y,24.8))||" then "||compress("grp\_&split\_var=")||put(\_N\_,5.)||";";

grp\_var=\_N\_;

var="&split\_var";

tmp=compress(put(y,24.8));

output;

end;

if last then do;

code="else if "||compress(tmp)||"<="||compress("&split\_var.")||" then "||compress("grp\_&split\_var=")||put(\_N\_+1,5.)||";";

grp\_var=\_N\_+1;

var="&split\_var";

tmp=compress(put(y,24.8));

output;

code="else "||compress("grp\_&split\_var=")||put(\_N\_+2,5.)||";";

grp\_var=\_N\_+2;

var="&split\_var";

output;

end;

run;

proc append base=&bin\_code\_interval data=cde\_&split\_var;

run;

%end;

%else %do;

data drop\_var\_numeric;

var="&split\_var.";

run;

proc append base=&bin\_drop\_var\_numeric data=drop\_var\_numeric;

run;

%end;

%end;

%let dsid=%sysfunc(open(&bin\_code\_interval));

%if &dsid>0 %then %do;

data \_null\_;

set &bin\_code\_interval;

file &split\_code\_pth lrecl=32767;

put code;

run;

%end;

%let dsid=%sysfunc(close(&dsid));

%mend;

WOE和IV值的计算代码（宏）

%MACRO WOE\_IV\_LST(DEV,WOE\_TABLE,target);

data &WOE\_TABLE;

length Var $40. VAR\_LST $40. Var\_GRP $40. TOT\_IV sum\_resp sum\_nresp \_freq\_ badrate TOTALGOOD TOTALBAD PERCGOOD PERCBAD ODDS WOE IV 8.;

stop;

run;

PROC CONTENTS DATA=&DEV

OUT=CONTENT\_tmp(KEEP=NAME) NOPRINT;

RUN;

data CONTENT;

set CONTENT\_tmp;

if upcase(substr(name,1,3))^='GRP' then delete;

run;

proc sql noprint;

select count(\*) into:n\_var

from content

;

quit;

data \_null\_;

set CONTENT;

%do i=1 %to &N\_VAR;

if \_N\_=&i then do;

call symput("FCT\_VAR\_&i",compress('F\_'||name)||'(keep=&target nresp resp '||compress(name)||')');

end;

%end;

run;

%macro FCT\_var;

%do i=1 %to &N\_VAR;

&&FCT\_VAR\_&i

%end;

%mend;

data %FCT\_var;

set &DEV;

if &target=1 then do;

nresp = 1;

resp = 0;

END;

ELSE DO;

nresp = 0;

resp = 1;

END;

run;

%do j=1 %to &n\_var;

data \_null\_;

set CONTENT;

if \_N\_=&j then do;

call symput('FCT\_VAR',name);

stop;

end;

run;

proc summary noprint data=f\_&FCT\_VAR missing;

class &FCT\_VAR;

var resp nresp;

output out=fct2 sum=sum\_resp sum\_nresp;

run;

data fct3;

retain &FCT\_VAR;

label sum\_resp = "GOOD" sum\_nresp = "BAD" \_freq\_ = "G+B" badrate = "BAD RATE(%)";

if \_n\_=1 then set fct2(keep=sum\_resp sum\_nresp rename=(sum\_resp=TOTALGOOD sum\_nresp=TOTALBAD));

set fct2(where = (\_type\_ = 1));

if sum\_resp = 0 then PERCGOOD=0.5/TOTALGOOD;

else PERCGOOD=sum\_resp/TOTALGOOD;

if sum\_nresp = 0 then PERCBAD=0.5/TOTALBAD;

else PERCBAD=sum\_nresp/TOTALBAD;

BADRATE = sum\_nresp / \_freq\_ ;

ODDS=PERCGOOD/PERCBAD;

WOE = log(ODDS);

IV= (PERCGOOD-PERCBAD)\*WOE;

format badrate 10.4;

drop \_TYPE\_;

run;

proc sql noprint;

select sum(IV) into: TOT\_IV

FROM FCT3;

QUIT;

DATA FCT4(DROP=&FCT\_VAR);

LENGTH Var VAR\_LST Var\_GRP $40 TOT\_IV 8;

SET FCT3;

VAR="&FCT\_VAR";

Var\_GRP=COMPRESS("&FCT\_VAR")||COMPRESS(&FCT\_VAR);

TOT\_IV=&TOT\_IV;

VAR\_LST=&FCT\_VAR;

RUN;

PROC APPEND BASE=&WOE\_TABLE DATA=FCT4;RUN;

%end;

%MEND;

共线性诊断（方差膨胀因子计算）代码

Proc Reg Data=Train\_Woe;

Model Evg\_Flg=Woe\_Age Woe\_Gen …

/ VIF COLLIN COLLINOINT;

Run;

显著性检验代码

Proc Logistic data=train\_woe out=train\_stat;

/\*调用SAS的Proc Logistic过程步，并指定数据集train\_woe，变量已经过粗分类转化为woe变量，模型参数输出到train\_stat表\*/

Model Evt\_Flg (Event='1') = &Var\_Woe. / selection=stepwise

sle=0.05 sls=0.05;

/\*显著性检验使用逐步筛选法，设定sle和sls两个检验指标，sle<0.05时变量进入模型，sls<0.05时变量留在模型中\*/

run;

数据打分代码

Proc Logistic data= train\_woe2 out=train\_stat;

Model Evt\_Flg (Event='1') = &Var\_Woe. / selection=none;

Score data= train\_woe2 out= score\_p;

run;

If then打分规则示例（模型）

lnp=0;

…

if .<Total\_call\_nbr<2.5 then lnp=lnp+-7.9874;

else if 2.5<=Total\_call\_nbr<3.5 then lnp=lnp+-1.0315;

else if 3.5<=Total\_call\_nbr<4.5 then lnp=lnp+0.0587;

else if 4.5<=Total\_call\_nbr<5.5 then lnp=lnp+0.3713;

else if 5.5<=Total\_call\_nbr<6.5 then lnp=lnp+1.1691;

else if 6.5<=Total\_call\_nbr<7.5 then lnp=lnp+1.2862;

else if 7.5<=Total\_call\_nbr then lnp=lnp+2.2826;

…

lnp=lnp+1.1598;

p1=1-exp(lnp)/(1+exp(lnp));

模型评估指标的计算代码

data valdt;

set score\_p;

where sta\_dte='31jan2015'd;

run;

%macro Fit(in,out,grp\_cnt,pred\_var,act\_var);

data work.tt1;

set &in;

run;

data \_null\_;

set work.tt1 nobs=obs ;

call symput("Base",obs/&grp\_cnt);

stop;

run;

proc sort data=work.tt1;

by descending &pred\_var;

run;

data work.tt1;

N=\_N\_;

set work.tt1;

format Grp2 4.0;

Grp2=INT((N-1)/&base);

run;

proc means data=work.tt1 nway noprint;

class Grp2 ;

output out=&out mean(&pred\_var &act\_var)=pred\_evt actual\_evt;

run;

%mend;

%Fit(valdt,fit\_v,20,p\_1,evt\_flg);

\*indicators;

proc sql;

select sum(\_freq\_\*actual\_evt) into:var1 from fit\_v;

select sum(\_freq\_-\_freq\_\*actual\_evt) into:var2 from fit\_v;

quit;

data tmp1;

set fit\_v;

resp=\_freq\_\*actual\_evt;

nonresp=\_freq\_-resp;

retain cul\_resp 0;

cul\_resp=cul\_resp+resp/&var1.;

retain cul\_nonresp 0;

cul\_nonresp=cul\_nonresp+nonresp/&var2.;

ks=cul\_resp-cul\_nonresp;

retain base 0;

base=base+0.05;

lift=cul\_resp/base;

run;

线性回归代码

proc reg data=train\_value outest=para ;

/\*指定建模数据集train\_value，将通过显著性检验的变量和参数估计值输出到指定的数据集para中\* /

model Value=&var. Ph/ vif collin selection= stepwise sle=0.05 sls=0.05;

/\*指定目标变量Value和预测变量x，这里用宏var表示；显著性检验采用逐步回归法，同时检验共线性，参数含义同上节介绍\*/

output out=pred\_value p=Pred\_v;

/\*对数据集打分，并输出到指定数据集pred\_value中，打分变量命名为Pred\_v \*/

run;

数据标准化/快速聚类代码

proc standard data=pred\_dat out=tmp1 mean=0 std=1;

var p\_1 pred\_v;

run;

proc fastclus data=tmp1 summary maxc=4 maxiter=99

outseed=seed replace=random random=1

out=tmp2 outstat=stat;

var p\_1 pred\_v;

run;