/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THIS CODE BUILDS THE STEPWISE , FORWARD & BACKWARD MODELS ON BOTH LINEAR AND LOGISTIC METHODOLOGIES FOR DIFFERENT RAW & TRANSFORMED VARIABLES POST APPLYING

THE RELEVANT FILTERS OF VIF VALUES , P VALUES AND SIGN CHECKS

The code also summarizes the results in terms of fits , summary , coefficient and variable contribution.

Developped By : Kinsuk Ghatak

Last Updated on : 13-05-2020

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

LIBNAME NEW\_2 SPDE "/sas/data/rmad/analytics/data\_output/model\_development/development/mep\_emrg\_mkt"

DATAPATH = ("/sas/data/rmad/analytics/data\_output/model\_development/development/mep\_emrg\_mkt/data")

INDEXPATH = ("/sas/data/rmad/analytics/data\_output/model\_development/development/mep\_emrg\_mkt/index")

HDFSHOST = DEFAULT PARTSIZE = 1048576;

LIBNAME APL SPDE "/sas/data/rmad/analytics/data\_output/model\_development/development/md024\_apl"

DATAPATH = ("/sas/data/rmad/analytics/data\_output/model\_development/development/md024\_apl/data")

INDEXPATH = ("/sas/data/rmad/analytics/data\_output/model\_development/development/md024\_apl/index")

HDFSHOST = DEFAULT PARTSIZE=1048576;

LIBNAME EM\_LIB SPDE '/sas/data/rmad/analytics/data\_output/model\_development/development/md026\_em\_pd\_2'

DATAPATH = ('/sas/data/rmad/analytics/data\_output/model\_development/development/md026\_em\_pd\_2/data')

INDEXPATH = ('/sas/data/rmad/analytics/data\_output/model\_development/development/md026\_em\_pd\_2/index')

HDFSHOST = DEFAULT PARTSIZE = 1048576;

data MFA\_Qual;

set em\_lib.EM\_Qual\_MVI\_FINAL\_WoE\_combined;

run;

/\*proc freq data=MFA\_Qual;table IND\_RSK\_FLG;run; \*/

data MFA\_Qual;

set MFA\_Qual;

where IND\_RSK\_FLG ne 'BLANK';

run;

data MFA\_Qual;

SET MFA\_Qual;

if IND\_RSK\_FLG = 'LOW' then IND\_RSK\_FLG\_f=0; else if IND\_RSK\_FLG='MEDIUM' then IND\_RSK\_FLG\_f=1 ; else if IND\_RSK\_FLG='HIGH' THEN IND\_RSK\_FLG\_f=2;

DROP IND\_RSK\_FLG;

RENAME IND\_RSK\_FLG\_f=IND\_RSK\_FLG;

RUN;

%let separator= %str( );

%let dummy\_list= %str();

%macro create\_dummy(variable\_list);

%do n=1 %to %sysfunc(countw(&variable\_list.));

%let variable\_raw=%scan(&variable\_list.,&n.);

proc sql;

select distinct &variable\_raw. into:var\_raw\_dist separated by ' ' from MFA\_qual;

quit;

%do p=1 %to %sysfunc(countw(&var\_raw\_dist))-1;

%let prefix=%scan(&var\_raw\_dist,&p.);

data MFA\_qual;

set MFA\_qual;

if &variable\_raw.= &prefix. then &variable\_raw.\_&prefix.=1;

else &variable\_raw.\_&prefix.=0 ;

run;

%let dummy\_list = &dummy\_list.&separator.&variable\_raw.\_&prefix.;

%put &dummy\_list.;

%end;

%end;

%mend;

options symbolgen mprint mlogic;

%create\_dummy(IND\_RSK\_FLG);

%let Ind\_vars=

/\*IND\_RSK\_FLG\_0\*/

/\*IND\_RSK\_FLG\_1\*/

;

%let vars\_raw =

AVAIL135

BARRI026

BUSIN452

CAPIT050

COMPE618

ENVIR960

FISCA988

INDUS038

INDUS404

INDUS712

INDUS844

INFLA869

LABOU080

PRODU265

REGUL789

TECHN323

GENER\_Combo

ACCES\_Combo

REFIN\_Combo

;

%let vars\_woe =

AVAIL135\_woe

BARRI026\_woe

BUSIN452\_woe

CAPIT050\_woe

COMPE618\_woe

ENVIR960\_woe

FISCA988\_woe

INDUS038\_woe

INDUS404\_woe

INDUS712\_woe

INDUS844\_woe

INFLA869\_woe

LABOU080\_woe

PRODU265\_woe

REGUL789\_woe

TECHN323\_woe

GENER\_Combo\_woe

ACCES\_Combo\_woe

REFIN\_Combo\_woe

;

%let vars\_cwoe =

AVAIL135\_cont\_woe

BARRI026\_cont\_woe

BUSIN452\_cont\_woe

CAPIT050\_cont\_woe

COMPE618\_cont\_woe

ENVIR960\_cont\_woe

FISCA988\_cont\_woe

INDUS038\_cont\_woe

INDUS404\_cont\_woe

INDUS712\_cont\_woe

INDUS844\_cont\_woe

INFLA869\_cont\_woe

LABOU080\_cont\_woe

PRODU265\_cont\_woe

REGUL789\_cont\_woe

TECHN323\_cont\_woe

GENER\_Combo\_cont\_woe

ACCES\_Combo\_cont\_woe

REFIN\_Combo\_cont\_woe

;

data var\_list\_sign;

length Effect $100.;

input Effect$ Exp\_Sign;

datalines;

AVAIL135 1

BARRI026 -1

BUSIN452 -1

CAPIT050 1

COMPE618 1

ENVIR960 -1

FISCA988 -1

INDUS038 1

INDUS404 -1

INDUS712 -1

INDUS844 1

INFLA869 1

LABOU080 1

PRODU265 -1

REGUL789 1

TECHN323 -1

GENER\_Combo -1

REFIN\_Combo 1

ACCES\_Combo -1

AVAIL135\_0 1

BARRI026\_0 1

BUSIN452\_0 -1

CAPIT050\_0 1

COMPE618\_0 1

ENVIR960\_0 -1

FISCA988\_0 -1

INDUS038\_0 1

INDUS404\_0 -1

INDUS712\_0 -1

INDUS844\_0 1

INFLA869\_0 1

LABOU080\_0 1

PRODU265\_0 -1

REGUL789\_0 1

TECHN323\_0 -1

GENER\_Combo\_0 -1

REFIN\_Combo\_0 1

ACCES\_Combo\_0 -1

AVAIL135\_1 1

BARRI026\_1 -1

BUSIN452\_1 -1

CAPIT050\_1 1

COMPE618\_1 1

ENVIR960\_1 -1

FISCA988\_1 -1

INDUS038\_1 1

INDUS404\_1 -1

INDUS712\_1 -1

INDUS844\_1 1

INFLA869\_1 1

LABOU080\_1 1

PRODU265\_1 -1

REGUL789\_1 1

TECHN323\_1 -1

GENER\_Combo\_1 -1

REFIN\_Combo\_1 1

ACCES\_Combo\_1 -1

AVAIL135\_2 1

BARRI026\_2 -1

BUSIN452\_2 -1

CAPIT050\_2 1

COMPE618\_2 1

ENVIR960\_2 -1

FISCA988\_2 -1

INDUS038\_2 1

INDUS404\_2 -1

INDUS712\_2 -1

INDUS844\_2 1

INFLA869\_2 1

LABOU080\_2 1

PRODU265\_2 -1

REGUL789\_2 1

TECHN323\_2 -1

GENER\_Combo\_2 -1

REFIN\_Combo\_2 1

ACCES\_Combo\_2 -1

AVAIL135\_3 1

BARRI026\_3 -1

BUSIN452\_3 -1

CAPIT050\_3 1

COMPE618\_3 1

ENVIR960\_3 -1

FISCA988\_3 -1

INDUS038\_3 1

INDUS404\_3 -1

INDUS712\_3 -1

INDUS844\_3 1

INFLA869\_3 1

LABOU080\_3 1

PRODU265\_3 -1

REGUL789\_3 1

TECHN323\_3 -1

GENER\_Combo\_3 -1

REFIN\_Combo\_3 1

ACCES\_Combo\_3 -1

IND\_RSK\_FLG\_0 -1

IND\_RSK\_FLG\_1 -1

AVAIL135\_woe 1

BARRI026\_woe 1

BUSIN452\_woe 1

CAPIT050\_woe 1

COMPE618\_woe 1

ENVIR960\_woe 1

FISCA988\_woe 1

INDUS038\_woe 1

INDUS404\_woe 1

INDUS712\_woe 1

INDUS844\_woe 1

INFLA869\_woe 1

LABOU080\_woe 1

PRODU265\_woe 1

REGUL789\_woe 1

TECHN323\_woe 1

GENER\_Combo\_woe 1

REFIN\_Combo\_woe 1

ACCES\_Combo\_woe 1

AVAIL135\_cont\_woe 1

BARRI026\_cont\_woe 1

BUSIN452\_cont\_woe 1

CAPIT050\_cont\_woe 1

COMPE618\_cont\_woe 1

ENVIR960\_cont\_woe 1

FISCA988\_cont\_woe 1

INDUS038\_cont\_woe 1

INDUS404\_cont\_woe 1

INDUS712\_cont\_woe 1

INDUS844\_cont\_woe 1

INFLA869\_cont\_woe 1

LABOU080\_cont\_woe 1

PRODU265\_cont\_woe 1

REGUL789\_cont\_woe 1

TECHN323\_cont\_woe 1

GENER\_Combo\_cont\_woe 1

REFIN\_Combo\_cont\_woe 1

ACCES\_Combo\_cont\_woe 1

;

proc print data=var\_list\_sign;

title2'Expected\_Signs';

run;

%let separator= %str( );

%let dummy\_list= %str();

%macro create\_dummy2(variable\_list);

%do n=1 %to %sysfunc(countw(&variable\_list.));

%let variable\_raw=%scan(&variable\_list.,&n.);

proc sql;

select distinct &variable\_raw. into:var\_raw\_dist separated by ' ' from MFA\_qual;

quit;

%do p=1 %to %sysfunc(countw(&var\_raw\_dist));

%let prefix=%scan(&var\_raw\_dist,&p.);

data MFA\_qual;

set MFA\_qual;

if &variable\_raw.= &prefix. then &variable\_raw.\_&prefix.=1;

else &variable\_raw.\_&prefix.=0 ;

run;

%let dummy\_list = &dummy\_list.&separator.&variable\_raw.\_&prefix.;

%put &dummy\_list.;

%end;

%end;

%mend;

options symbolgen mprint mlogic;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NOW WE START BUILDING THE LOGISTIC MODELS IN FWD , STEPWISE AND BACXKWARD METHODS .

THE FOLLOWING CODES & MACROS ARE USED TO BUILD THE MODELS , SUMMARISE THEM AND THEN REPORT THE SUMMARY , FITS AND COEFFICIENTS ACCORDINGLY

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MACRO TO BUILD STEPWISE LOGISTIC MODELS :

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%macro logstic\_stepwise(VARS,SLENTRY,SLSTAY,SELECTION);

%let temp= %str();

%let separator= %str( );

%let temp\_raw= %str();

%let var\_AR= %str();

%let temp\_vars=%str();

%let dummy\_list= %str();

%let temp\_ind = %str();

%let temp\_logistic = %str();

%let p\_val\_thres=0.05;

%let vif\_thres =4;

%let var\_flag =%str();

ods output association=association;

ods output FitStatistics=FitStatistics;

ods output ParameterEstimates=ParameterEstimates;

ods output ModelBuildingSummary=summary\_logistic;

PROC logistic data=MFA\_Qual;

class &vars\_raw./order=data;

model intodef(event='1')=&VARS. &Ind\_vars. /

stb

selection = &SELECTION.

slentry=&SLENTRY.

slstay=&SLSTAY.

details

lackfit;

output out=pred p=phat lower=lcl upper= ucl;

run;

%let model\_id=&SELECTION.\_&SLENTRY.\_.&SLSTAY.;

data Param\_logistic;

set ParameterEstimates;

rename Variable=Effect;

method="&SELECTION.";

slentry=&SLENTRY.;

slstay=&SLSTAY.;

model\_id="&model\_id.";

run;

proc sql;

create table Param\_logistic as select \* from Param\_logistic having step=max(step);

quit;

proc sql;

create table Param\_logistic2 as select a.\*,b.step, b.nValue2 as Somers\_D from Param\_logistic a

left join association b on a.step=b.step

where b.label2 like'%Somer%';

quit;

proc sql;

create table Param\_logistic\_FINAL as select a.\*,b.InterceptAndCovariates as AIC\_Logistic from Param\_logistic2 a

left join FitStatistics b on a.step=b.step

where b.criterion="AIC";

quit;

proc sql; select distinct Effect into : Final\_Logistic\_Vars separated by ' ' from Param\_Logistic\_Final ;quit;

%let find\_woe=%index(&Final\_Logistic\_Vars.,woe);

data Param\_logistic\_FINAL;

length Var\_Level $300.;

set Param\_logistic\_FINAL;

if &find\_woe. ne 0 then Var\_Level=Effect;

else Var\_Level=catx('\_',Effect,ClassVal0);

run;

data Param\_logistic\_FINAL;

set Param\_logistic\_FINAL;

Actual\_Sign=sign(Estimate);

run;

proc sql; create table Param\_logistic\_FINAL as select a.\*,b.Exp\_sign from Param\_logistic\_FINAL a left join var\_list\_sign b

on a.Effect=b.Effect ;

quit;

data Param\_logistic\_FINAL;

set Param\_logistic\_FINAL (keep= Step model\_id method Effect Var\_Level Estimate StandardizedEst StdErr ProbChisq Somers\_D AIC\_Logistic Actual\_Sign Exp\_Sign) ;

if Actual\_Sign=Exp\_Sign then Sign\_Match=1;

else Sign\_Match=0;

run;

proc sql; select distinct count(Effect) into : vars\_no from Param\_logistic\_FINAL ;quit;

data Coefficients\_final;

set Param\_logistic\_FINAL;

Total\_Variables=&vars\_no.;

rename ProbChisq=p\_value;

rename Somers\_D=AR\_Logistic;

rowID+1;

run;

proc sql;

select sum(case when Sign\_Match=1 then 1 else 0 end)/count(\*) into: Percnt\_Sign\_Match from Coefficients\_final;

quit;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*VIF CALCULATION STARTS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

proc sql;

select distinct Effect into : var\_AR separated by ' ' from Coefficients\_final where Effect ne 'Intercept';

quit;

data coefficients\_final1;

set coefficients\_final;

where Estimate ne 0 & Effect ne 'Intercept' & Effect ne 'IND\_RSK\_FLG\_0' & Effect ne 'IND\_RSK\_FLG\_1';

run;

proc sql;

select distinct Var\_Level into:Var\_Level\_Raw separated by ' ' from coefficients\_final1;

quit;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*changes and addtiions stsart \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%do p=1 %to %sysfunc(countw(&var\_AR.));

%let var=%scan(&var\_AR.,&p.);

%let find\_woe1=%index(&var.,woe);

%let find\_ind=%index(&var.,IND\_RSK);

%if %eval(&find\_woe1.)= 0 & %eval(&find\_ind.)= 0 %then

%do;

%let temp\_raw=&temp\_raw.&separator.&var.;

%end;

%if %eval(&find\_ind.) ne 0 %then

%do;

%let temp\_ind=&temp\_ind.&separator.&var.;

%end;

%end;

%if &temp\_raw. ne " " %then

%do;

%create\_dummy2(&temp\_raw.);

%let temp\_vars=&Var\_Level\_Raw.&separator.&temp\_ind.;

%end;

%else

%do;

%let temp\_vars=&var\_AR.;

%end;

ods output ParameterEstimates=Coefficients\_VIF;

proc reg data=MFA\_Qual;

model log\_pd=&temp\_vars./VIF;

run;

data VIF\_data (rename=(Variable=Var\_Level VarianceInflation=VIF));

set Coefficients\_VIF (keep= Variable VarianceInflation);

run;

proc sql;

create table Coefficients\_final as select a.\*,b.VarianceInflation as VIF , b.Probt as pval\_linear from Coefficients\_final a left join Coefficients\_VIF b

on a.Var\_Level=b.Variable;

quit;

/\* proc sql; create table Coefficients\_final as select a.\* ,b.VIF from Coefficients\_final a left join VIF\_data b \*/

/\* on a.Var\_Level= b.Variable;\*/

/\* quit;\*/

data Coefficients\_final;

set Coefficients\_final;

abs\_st\_estimate = abs(StandardizedEst);

run;

proc sql;

create table coefficients\_final as select distinct Effect,Var\_Level,rowID,method,model\_id,Estimate,StandardizedEst,abs\_st\_estimate, p\_value ,pval\_linear,VIF,

(abs\_st\_estimate/sum(abs\_st\_estimate)) as Percent\_Contribution,AR\_Logistic,AIC\_Logistic ,actual\_sign,Exp\_sign,Sign\_Match

from coefficients\_final where Var\_Level NE "Intercept";

quit;

proc sql;

create table coefficients\_final as select distinct Effect,Var\_Level,rowID,method,model\_id,Estimate,StandardizedEst,abs\_st\_estimate, p\_value ,pval\_linear,VIF,

Percent\_Contribution,sum(Percent\_Contribution) as Var\_total\_contri, AR\_Logistic,AIC\_Logistic ,actual\_sign,Exp\_sign,Sign\_Match

from coefficients\_final group by Effect;

quit;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

proc sql; select max(p\_value) into: max\_p\_value from Coefficients\_final;quit;

proc sql; select max(VIF) into: max\_VIF from Coefficients\_final;quit;

proc sql; create table logistic\_summary\_main as select model\_id,method,AR\_Logistic,AIC\_Logistic

from Coefficients\_final where rowID=1;

quit;

data logistic\_summary\_main;

set logistic\_summary\_main;

Max\_p\_Value=&max\_p\_value.;

Max\_VIF=&max\_VIF.;

percent\_sign\_match = &Percnt\_Sign\_Match.;

run;

data Coefficients\_final\_check\_main;

set Coefficients\_final;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Check and filtering of the models begin \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%let vars\_nxt1= %str( );

%let vars\_nxt2= %str( );

%let final\_filtd\_vars =%str( );

%do %while(%sysevalf(&max\_p\_value.>= &p\_val\_thres.) and (&max\_VIF.>=&vif\_thres.));

proc sql;

select distinct quote(trim(Var\_Level),"'") into: vars\_nxt1 separated by ','

from Coefficients\_final

where p\_value <=&p\_val\_thres. or vif <= &vif\_thres.;

quit;

proc sql;

select distinct Var\_Level into: vars\_nxt11 separated by ' '

from Coefficients\_final

where p\_value <=&p\_val\_thres. & vif <= &vif\_thres.;

quit;

proc sql;

create table coefficient\_final2 as select Var\_Level,Percent\_Contribution,Sign\_Match from Coefficients\_final

where Var\_Level in (&vars\_nxt1.);

quit;

data coefficient\_final2\_1;

set coefficient\_final2;

where Sign\_Match=1;

run;

proc sql; select distinct Var\_Level into: vars\_1 separated by ' ' from coefficient\_final2\_1 ;quit;

%let sign\_match\_find\_woe = %index(&vars\_nxt11.,woe);

%if %eval(&sign\_match\_find\_woe.=0) %then %do;

data coefficient\_final12\_2;

set coefficient\_final2;

where Sign\_Match=0;

run;

%end;

proc sql; select count(\*) into : count\_row\_coef\_final2\_2 from coefficient\_final12\_2;quit;

%if %sysevalf(&count\_row\_coef\_final2\_2.>1) %then %do ;

proc sql;

select distinct Var\_Level into: vars\_2 separated by ' '

from coefficient\_final12\_2

having Percent\_Contribution ne min(Percent\_Contribution);

quit;

%let vars\_nxt2 = &vars\_1. &vars\_2.;

%end;

%else %do;

%let vars\_nxt2 = &vars\_1.;

%end;

ods output ParameterEstimates=ParameterEstimates2;

ods output association=association2;

ods output FitStatistics=Fitstatistics\_logistic2;

PROC logistic data=MFA\_Qual;

model intodef(event='1')=&vars\_nxt2./stb;

output out=qual\_PD\_Pred\_log pred=pred;

run;

proc sql; select max(ProbChiSq) into :max\_p\_value from ParameterEstimates2 ;quit;

ods output ParameterEstimates=Coefficients\_reg2;

ods output FitStatistics=Fitstatistics2;

proc reg data=MFA\_Qual outest=Reg\_out;

model log\_pd=&vars\_nxt2./VIF STB AIC;

run;

proc sql; select max(VarianceInflation) into :max\_VIF from Coefficients\_reg2;quit;

data ParameterEstimates2;

set ParameterEstimates2;

rename Variable=Var\_Level;

rename ProbChiSq=p\_value;

max\_p= &max\_p\_value.;

max\_vif= &max\_VIF.;

actual\_sign=sign(Estimate);

abs\_st\_estimate=ABS(StandardizedEst);

run;

proc sql;

create table ParameterEstimates3 as select a.\*,b.Exp\_Sign from ParameterEstimates2 a left join var\_list\_sign b

on a.Var\_Level=b.Effect;

quit;

proc sql;

create table ParameterEstimates4 as select a.\*,b.VarianceInflation as VIF , b.Probt as pval\_linear from ParameterEstimates3 a left join Coefficients\_reg2 b

on a.Var\_Level=b.Variable;

quit;

data ParameterEstimates4;

set ParameterEstimates4;

Effect=scan(Var\_Level,1,"\_");

if Exp\_sign=actual\_sign then

Sign\_Match=1;

else Sign\_Match=0;

run;

data ParameterEstimates4;

set ParameterEstimates4;

rowID+1;

method="&SELECTION.";

model\_id="&model\_id.";

run;

proc sql;

create table Coefficients\_final as select distinct Effect,Var\_Level,rowID,method,model\_id,Estimate,StandardizedEst,abs\_st\_estimate, p\_value, pval\_linear,

(abs\_st\_estimate/sum(abs\_st\_estimate)) as Percent\_Contribution, VIF,actual\_sign,Exp\_sign,Sign\_Match

from ParameterEstimates4 where Var\_Level NE "Intercept";

quit;

proc sql;

create table coefficients\_final as select distinct Effect,Var\_Level,rowID,method,model\_id,Estimate,StandardizedEst,abs\_st\_estimate, p\_value ,pval\_linear,VIF,

Percent\_Contribution,sum(Percent\_Contribution) as Var\_total\_contri, actual\_sign,Exp\_sign,Sign\_Match

from coefficients\_final group by Effect;

quit;

proc sql; select nValue2 into : AR\_Logistic\_Final from Association2 where Label2 like'%Somers%';quit;

proc sql; select InterceptAndCovariates into : AIC\_Logistic\_Final from Fitstatistics\_logistic2 where Criterion like'%AIC%';quit;

data Coefficients\_final;

set Coefficients\_final;

AR\_Logistic= &AR\_Logistic\_Final.;

AIC\_Logistic = &AIC\_Logistic\_Final.;

rowID+1;

run;

%end;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Filtering ends final variables passed on to the final model \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%let var\_seq= Effect Var\_Level rowID method model\_id Estimate StandardizedEst abs\_st\_estimate p\_value VIF Percent\_Contribution Var\_total\_contri AR\_Logistic AIC\_Logistic Actual\_Sign Exp\_Sign Sign\_Match;

data Coefficients\_final;

retain &var\_seq;

set Coefficients\_final;

run;

proc sql; select max(p\_value) into: max\_p\_final from Coefficients\_final;quit;

proc sql; select max(VIF) into: max\_VIF\_final from Coefficients\_final;quit;

proc sql; select distinct Var\_Level into : final\_vars separated by ' ' from coefficients\_final; quit;

proc sql; create table logistic\_summary\_final as select model\_id,method,AR\_Logistic,AIC\_Logistic

from Coefficients\_final having rowId=min(rowId);

quit;

data logistic\_summary\_final;

set logistic\_summary\_final;

Max\_p\_Value=&max\_p\_final.;

Max\_VIF=&max\_VIF\_final.;

run;

data pred;

Rename Phat = Predicted\_PD;

set pred (keep= CARM\_INSTANCE RELN\_ID CUST\_ID APPROVAL\_DATE REP\_DATE INTODEF YR AUDITMETHOD Country\_Use CTRY\_RSK\_FLG IND\_RSK\_FLG PD Log\_PD Intodef Phat lcl ucl);

if Phat NE ' ' then

output;

run;

%mend;

options symbolgen mprint mlogic;

%logstic\_stepwise(&vars\_raw.,0.05,0.05,stepwise);

options symbolgen mprint mlogic;

%logstic\_stepwise(&vars\_woe.,0.05,0.05,stepwise);

%logstic\_stepwise(&vars\_woe.,0.05,0.05,forward);

%logstic\_stepwise(&vars\_woe.,0.05,0.05,backward);

%let separator= %str(\_);

%let methods=

STEPWISE

FORWARD

BACKWARD

;

%let final\_var\_list =

vars\_raw

vars\_woe

vars\_cwoe

;

%macro logistic\_final\_summary\_loop();

data summary\_all\_log\_models\_final;

set \_null\_;

run;

data coef\_summary\_all\_log\_final;

/\* length Var\_Level $300.;\*/

set \_NULL\_;

run;

data predicted\_val\_logistic\_summary;

set \_NULL\_;

run;

%do a=1 %to %sysfunc(countw(&final\_var\_list.));

%let vars=%scan(&final\_var\_list.,&a.);

%let temp= %str();

%let separator= %str( );

%let temp\_raw= %str();

%let var\_AR= %str();

%let total\_qual\_list= %str();

%let vars\_nxt1= %str( );

%let vars\_nxt2= %str( );

%let final\_filtd\_vars =%str( );

%let Max\_p\_Value= %str();

%let Max\_VIF = %str();

%let max\_VIF\_final = %str();

%let max\_p\_final = %str();

%let AR\_Logistic\_Final = %str();

%let AIC\_Logistic\_Final= %str();

%let vars\_nxt2= %str();

%let vars\_1= %str();

%let vars\_2 = %str();

%let vars\_nxt11=%str();

%do k=1 %to %sysfunc(countw(&methods.));

%let method=%scan(&methods.,&k.);

%let method\_short=%substr(&method.,1,4);

%do l=5 %to 5 %by 5;

%let SLENTRY= %sysevalf(&l./100);

%do m=5 %to 5 %by 5;

%let SLSTAY= %sysevalf(&m./100);

%let model\_id\_main=log\_&method\_short.\_&vars.;

%logstic\_stepwise(&&&VARS..,&SLENTRY.,&SLSTAY.,&method.);

data coef\_&model\_id\_main.;

set Coefficients\_final;

model\_id\_main= "&model\_id\_main.";

run;

data pred\_&model\_id\_main.;

set pred;

model\_id\_main= "&model\_id\_main.";

run;

data model\_sum\_&model\_id\_main.;

set logistic\_summary\_final;

model\_id\_main= "&model\_id\_main.";

run;

data coef\_summary\_all\_log\_final;

set coef\_summary\_all\_log\_final coef\_&model\_id\_main.;

run;

data summary\_all\_log\_models\_final;

set summary\_all\_log\_models\_final model\_sum\_&model\_id\_main.;

run;

data predicted\_val\_logistic\_summary;

set predicted\_val\_logistic\_summary pred\_&model\_id\_main.;

run;

%end;

%end;

%end;

%end;

%mend;

%logistic\_final\_summary\_loop();

data EM\_LIB.EM\_Qual\_LOG\_op\_IND\_FLG;

set predicted\_val\_logistic\_summary;

run;

data EM\_LIB.EM\_Qual\_LOG\_op\_Withoug\_IND\_FLG;

set predicted\_val\_logistic\_summary;

run;