
28

Net T-C Lift Model: Assessing the Net Effects of Test and Control Campaigns

Appendix 28.A TEST Logistic with Xs

```
libname upl 'c://0-upl';
options pageno=1;

%let depvar=TEST;
%let indvars= X14 X23 X25 X26 X36 X41 X42 X49;
PROC LOGISTIC data= upl.upl_datanumkpX nosimple des outest=coef;
model &depvar = &indvars;
run;

PROC SCORE data=upl.upl_datanumkpX predict type=parms score=coef out=score;
var &indvars;
run;

data score;
set score;
logit=&depvar.2;
prob_TEST=exp(logit)/(1+ exp(logit));

data score;
set score;
```

```
estimate=&depvar.2;
```

```
run;
```

```
data notdot;
```

```
set score;
```

```
if estimate ne .;
```

```
run;
```

```
PROC MEANS data=notdot sum noprint; var wt;
```

```
output out=samsize (keep=samsize) sum=samsize;
```

```
run;
```

```
data scoresam (drop=samsize);
```

```
set samsize score;
```

```
retain n;
```

```
if _n_=1 then n=samsize;
```

```
if _n_=1 then delete;
```

```
run;
```

```
PROC SORT data=scoresam; by descending estimate;
```

```
run;
```

```
data score;
```

```
set scoresam;
```

```
if estimate ne . then cum_n+wt;
```

```
if estimate = . then dec=.
```

```
else dec=floor(cum_n*10/(n+1));
```

```
run;
```

```
PROC SUMMARY data=score missing;
```

```
class dec;
```

```
var &depvar wt;
```

```
output out=sum_dec sum=sum_can sum_wt;
```

```
data sum_dec;
```

```
set sum_dec;
```

```
avg_can=sum_can/sum_wt;
```

```
run;
```

```
data avg_rr;
set sum_dec;
if dec=.;
keep avg_can;
run;
```

```
data sum_dec1;
set sum_dec;
if dec=. or dec=10 then delete;
cum_n +sum_wt;
r =sum_can;
cum_r +sum_can;
cum_rr=(cum_r/cum_n)*100;
avg_cann=avg_can*100;
run;
```

```
data avg_rr;
set sum_dec1;
if dec=9;
keep avg_can;
avg_can=cum_rr/100;
run;
```

```
data scoresam;
set avg_rr sum_dec1;
retain n;
if _n_=1 then n=avg_can;
if _n_=1 then delete;
lift=(cum_rr/n);
if dec=0 then decc=' top ';
if dec=1 then decc=' 2  ';
if dec=2 then decc=' 3  ';
if dec=3 then decc=' 4  ';
if dec=4 then decc=' 5  ';
if dec=5 then decc=' 6  ';
if dec=6 then decc=' 7  ';
if dec=7 then decc=' 8  ';
```

```

if dec=8 then decc=' 9  ';
if dec=9 then decc='bottom';
if dec ne .;
run;

title2' Decile Analysis based on ' ;
title3" &depvar Regressed on &indvars " ;

PROC PRINT data=scoresam d split='*' noobs;
var decc sum_wt r avg_cann cum_rr lift;
label decc='DECILE'
sum_wt ='NUMBER OF*CUSTOMERS'
r ='NUMBER OF*RESPONSES'
cum_r ='CUM No. CUSTOMERS w/* RESPONSES'
avg_cann ='TEST*RATE (%)'
cum_rr ='CUM TEST* RATE (%)'
lift =' C U M *LIFT (%)';
sum sum_wt r;
format sum_wt r cum_n cum_r comma10.;
format avg_cann cum_rr 5.2;
format lift 3.0;
run;

```

Appendix 28.B CONTROL Logistic with Xs

```

libname upl 'c://0-upl';
options pageno=1 ;
title ' ';
title2 'CONTROL1 ' ;
%let depvar=CONTROL;
%let indvars= X17 X23 X29 X37 X42 ;
title3 "adjust_n = &Control_n";
title4 "adjust_dot = &Contrl_dot";

```

```
PROC LOGISTIC data= upl.CONTROL1 nosimple des outest=coef;
model &depvar = &indvars;
run;
```

```
PROC SCORE data=upl.CONTROL1 predict type=parms score=coef out=score;
var &indvars;
run;
```

```
data score;
set score;
logit=&depvar.2;
prob_CONTROL=exp(logit)/(1+ exp(logit));
data score;
set score;
estimate=&depvar.2;
run;
```

```
data notdot;
set score;
if estimate ne .;
```

```
PROC MEANS data=notdot sum noprint; var wt;
output out=samsize (keep=samsize) sum=samsize;
run;
```

```
data scoresam (drop=samsize);
set samsize score;
retain n;
if _n_=1 then n=samsize;
if _n_=1 then delete;
run;
```

```
PROC SORT data=scoresam; by descending estimate;
run;
```

```
data score;
set scoresam;
if estimate ne . then cum_n+wt;
if estimate = . then dec=.;
else dec=floor(cum_n*10/(n+1));
run;
```

```
PROC SUMMARY data=score missing;
class dec;
var &depvar wt;
output out=sum_dec sum=sum_can sum_wt;
```

```
data sum_dec;
set sum_dec;
avg_can=sum_can/sum_wt;
run;
```

```
data avg_rr;
set sum_dec;
if dec=.;
keep avg_can;
run;
```

```
data sum_dec1;
set sum_dec;
if dec= . or dec=10 then delete;
cum_n +sum_wt;
r =sum_can;
cum_r +sum_can;
cum_rr=(cum_r/cum_n)*100;
avg_cann=avg_can*100;
run;
```

```
data avg_rr;
set sum_dec1;
if dec=9;
keep avg_can;
```

```

avg_cann=cum_rr/100;
run;

data scoresam;
set avg_rr sum_dec1;
retain n;
if _n_=1 then n=avg_cann;
if _n_=1 then delete;
lift=(cum_rr/n);
if dec=0 then decc=' top ';
if dec=1 then decc=' 2  ';
if dec=2 then decc=' 3  ';
if dec=3 then decc=' 4  ';
if dec=4 then decc=' 5  ';
if dec=5 then decc=' 6  ';
if dec=6 then decc=' 7  ';
if dec=7 then decc=' 8  ';
if dec=8 then decc=' 9  ';
if dec=9 then decc='bottom';
if dec ne .; run;

```

```

title5" &depvar Regressed on &indvars ";
PROC PRINT data=scoresam d split='*' noobs;
var decc sum_wt r avg_cann cum_rr lift;
label decc='DECILE'
sum_wt ='NUMBER OF*CUSTOMERS'
r ='NUMBER OF*RESPONSES'
cum_r ='CUM No. CUSTOMERS w/* CONTROLS'
avg_cann ='CONTROL *RATE (%)'
cum_rr ='CUM CONTROL * RATE (%)'
lift =' C U M *LIFT (%)';
sum sum_wt r;
format sum_wt r cum_n cum_r comma10.;
format avg_cann cum_rr 5.2;
format lift 3.0;
run;
footnote;

```

Appendix 28.C Merge Score

```
libname upl 'c://0-upl';
```

```
data score_RESP_uni;  
set upl.score_RESP;  
uni=uniform(12345);
```

```
data score_CNTRL_uni;  
set upl.score_CNTRL;  
uni=uniform(12345);
```

```
PROC SORT data =score_RESP_uni; by uni;  
PROC SORT data=score_CNTRL_uni; by uni;  
run;
```

```
data RESP_CNTRL_scores_uni;  
merge  
score_RESP_uni (in=r)  
score_CNTRL_uni (in=c); by uni;  
if r=1 then wtT=1; else wtT=0;  
if c=1 then wtC=1; else  
wtC=0;  
run;
```

```
data upl.diff_probs_uni;  
set RESP_CNTRL_scores_uni;  
diff_probs=prob_RESPONSE-prob_CONTROL;  
run;
```

```
PROC MEANS data=upl.diff_probs_uni n nmiss min max mean;  
var diff_probs prob_RESPONSE prob_CONTROL;  
run;
```

Appendix 28.D NET T-C Decile Analysis

```
libname upl 'c://0-upl';
options pageno=1 ps=33;

data score;
set upl.diff_probs_uniBS;
estimate=diff_probs;
do until (-0.435 < uni < 0.12345);
uni=uniform(12345);
end;
if estimate=. then estimate=uni;
TEST=RESPONSE;
keep _n_ wt wtC TEST CONTROL estimate;
run;

data notdot;
set score;
PROC MEANS data=notdot sum noprint; var wt;
output out=samsize (keep=samsize) sum=samsize;
run;

data scoresam (drop=samsize);
set samsize score;
retain n;
if _n_=1 then n=samsize;
if _n_=1 then delete;
run;

PROC SORT data=scoresam; by descending estimate;
run;

data score;
set scoresam;
if estimate ne . then cum_n+wt;
if estimate = . then dec=.;
```

```
else dec=floor(cum_n*10/(n+1));
if dec=. then delete;
run;
```

```
PROC SUMMARY data=score missing;
class dec;
var TEST wt;
output out=sum_decT sum=sum_canT sum_wtT;
run;
```

```
data sum_decT;
set sum_decT;
avg_canT=sum_canT/sum_wtT;
run;
```

```
data avg_rrT;
set sum_decT;
if dec=.;
keep avg_canT;
run;
```

```
data sum_dec1T;
set sum_decT;
if dec=. or dec=10 then delete;
cum_nT +sum_wtT;
rT =sum_canT;
cum_rT +sum_canT;
cum_rrT=(cum_rT/cum_nT)*100;
avg_cannT=avg_canT*100;
run;
```

```
data scoresamT;
set avg_rrT sum_dec1T;
retain n;
if _n_=1 then n=avg_canT;
if _n_=1 then delete;
liftT=(cum_rrT/n);
```

```
run;
```

```
PROC SUMMARY data=score missing;  
class dec;  
var CONTROL wtC;  
output out=sum_decC sum=sum_canC sum_wtC;  
run;
```

```
data sum_decC;  
set sum_decC;  
avg_canC=sum_canC/sum_wtC;  
run;
```

```
data avg_rrC;  
set sum_decC;  
if dec=.;  
keep avg_canC;  
run;
```

```
data sum_dec1C;  
set sum_decC;  
if dec=. or dec=10 then delete;  
cum_nC +sum_wtC;  
rC =sum_canC;  
cum_rC +sum_canC;  
cum_rrC=(cum_rC/cum_nC)*100;  
avg_cannC=avg_canC*100;  
run;
```

```
data scoresamC ;  
set avg_rrC sum_dec1C;  
retain n;  
if _n_=1 then n=avg_canC;  
if _n_=1 then delete;  
liftC=(cum_rrC/n);  
run;
```

```
PROC SORT data=scoresamC (drop= _FREQ_ _type_ n liftC); by dec;
PROC SORT data=scoresamT (drop= _FREQ_ _type_ n liftT); by dec;
```

```
data scoresam_TAC;
merge scoresamC scoresamT; by dec;
run;
```

```
data scoresam_TAC;
set scoresam_TAC;
CNTRL_SIZE =sum_wtC;
TEST_SIZE =sum_wtT;
TEST_CUM =cum_rrT;
CNTRL_CUM =cum_rrC;
CNTRL_RESP =rC;
TEST_RESP =rT;
TEST_MEAN =avg_cannT;
CNTRL_MEAN =avg_cannC;
NET_TEST_MEAN = (TEST_MEAN-CNTRL_MEAN);
CUM_TAC+NET_TEST_MEAN;
NET_TAC=TEST_RESP-CNTRL_RESP;
CUM_TAC1+TEST_RESP-CNTRL_RESP;
m=1;
run;
```

```
data lift_base;
set scoresam_TAC;
if dec=9 ;
lift_base=CUM_TAC1;
m=1;
keep lift_base m;
```

```
PROC SORT data=scoresam_TAC; by m;
PROC SORT data= lift_base; by m;
data scoresam_TAC_LIFT;
merge scoresam_TAC lift_base; by m;
drop m;
CUM_LIFT=(CUM_TAC/lift_base)*100;
```

```
CUM_LIFT1=(CUM_TAC1/lift_base)*100;
```

```
data LIFT;  
set scoresam_TAC_LIFT;  
if dec=0 then decc=' top  ' ;  
if dec=1 then decc=' 2  ' ;  
if dec=2 then decc=' 3  ' ;  
if dec=3 then decc=' 4  ' ;  
if dec=4 then decc=' 5  ' ;  
if dec=5 then decc=' 6  ' ;  
if dec=6 then decc=' 7  ' ;  
if dec=7 then decc=' 8  ' ;  
if dec=8 then decc=' 9  ' ;  
if dec=9 then decc='bottom';  
if dec ne . ;  
run;
```

```
PROC SORT data=scoresam_TAC_LIFT; by dec;  
PROC SORT data=LIFT; by dec;
```

```
data upl.final_NET_TAC;  
merge LIFT scoresam_TAC_LIFT; by dec;  
overall_net=(test_cum-cntrl_cum)/100;  
call symputx('overall',put(overall_net,percent8.1));  
NET_IMPROV=(NET_TAC/TEST_SIZE)*100;  
run;  
footnote " Overall NET T-C: &overall";
```

```
PROC PRINT data=upl.final_NET_TAC d split='*' noobs;  
var decc  
TEST_SIZE CNTRL_SIZE  
TEST_RESP CNTRL_RESP  
NET_TAC NET_IMPROV CUM_TAC1 CUM_LIFT1;
```

```
label  
decc='DECILE'  
TEST_SIZE ='TEST*Group*(n)'
```

```

CNTRL_SIZE ='CONTROL*Group*(n)'

TEST_RESP =' TEST*Count*(n)'
CNTRL_RESP='CONTROL*Count*(n)'

TEST_MEAN ='TEST*Rate*(%)'
CNTRL_MEAN='CNTRL*Rate*(%)'

TEST_CUM  ='CUM*TEST*Rate*(%)'
CNTRL_CUM ='CUM*CONTROL*Rate*(%)'

NET_TEST_MEAN ='NET T-C*Rate*(%)'
NET_TAC       ='NET T-C*Count*(n)'
NET_IMPROV    ='NET T-C*IMPROV*(%)'
CUM_TAC1     ='CUM*NET T-C*Count*(n)'
CUM_LIFT1    ='CUM*NET T-C*LIFT*(%)';
sum
TEST_SIZE CNTRL_SIZE TEST_RESP CNTRL_RESP;
format TEST_SIZE CNTRL_SIZE TEST_RESP CNTRL_RESP comma6.;
format NET_TEST_MEAN 6.1;
format TEST_CUM  CNTRL_CUM  4.1;
format TEST_MEAN CNTRL_MEAN  4.1;
format NET_TAC 3.0;
format CUM_TAC1 comma6.0;
format CUM_LIFT1 5.1;
format NET_IMPROV 4.1;
run;
footnote;

```