Paper SD01

Binary Logistic Regression Model Optimization

Jerry Musial, Cingular Wireless, Atlanta, GA

ABSTRACT

There are a number of techniques available to statisticians to measure predictiveness of a binary logistic regression model. The log likelihood ratio chi-square, Brier score and classification tables are some of these methods. Models that are very good at correctly predicting events may also have a high false positive rate. This paper will describe a set of macros which can be used to build, compare and select an optimized model from a series of competing models by using multiple measures of model predictiveness.

INTRODUCTION

SAS PROC LOGISTIC and other SAS/STAT procedures provide tables and statistics to help you analyze and evaluate the estimated model. The AIC (Akaike Information Criterion) and SC (Schwarz Criterion) are two criteria reported by PROC LOGISTIC that can be used to compare models. You may also calculate the Brier score, ROC (receiver operating characteristic) curves and classification tables.

This paper will describe a method of comparing, ranking and selecting the optimal model for predicting voluntary churn. Churn is a term used in the telecom industry to define a customer who voluntarily disconnects their service. Four user-defined criteria will be used to determine the optimal model. 1) Brier Score; 2) The number of records correctly predicted to be a churners and non-churners; 3) The percent of overall churn predicted; 4) Cost – benefit analysis. Each individual step will be described followed by a description of how to combine all these steps into one optimization process. Note: This paper assumes the modeler has gone through the requisites steps of variable selection and reduction. The process described here comes after reducing my modeling database of 300 variables to less than 30 using odds-ratios, stepwise regression and correlation checks.

GENDAT MACRO – GENERATE DATA

The first step in the process is a DATA step which randomly partitions the modeling data set into a modeling and validation records. The validation response is set equal to the response value and the response variable is set to missing for validation records. PROC LOGISTIC will discard these records while fitting the logistic regression model. Fortunately, the PROC LOGISTIC output data set will contain predicted probabilities for all of the observations in the data set. The observed response (the variable ‘pchurn’) for the validation records will be used to test the predictive accuracy of the model.

The data step also determines the number or percent of observations which will be in the model or validation set. One may also further define the percent of responder and non-responders included in the modeling set. The example code below shows a slight over sampling of the churners (40 percent) versus 30 percent of non-churners.

```
%macro gendat
 %global tot_val ;
 data analysis ;
   set churn.guard1_atl end=eof;
   retain tot_churn tot_val 0;
 /***********************************/
 /** Churners                      **/
 /***********************************/
   if churn = 1 then do;
      if uniform(0) < .40 then do;
      totchurn + 1;
      end;
    else  do;
      pchurn = 1;   /* Validation Churner */
      churn = .;
      tot_val + 1;
      end;
    end;
 /***********************************/
 /** Non-Churner                   **/
 /***********************************/
 else if churn = 0 then do;
      if uniform(0) < .3  then do;
        tot_churnx + 1;
       end;
      else do;
        pchurn = 0; /*Validation Non-Churn*/
        churn = .;  tot_val + 1;
        end;
      end;
 if eof then do;
    /* nbr of validation records */
    call symput('tot_val',tot_val);
    end;
 %mend gendat;
```

%LGSTIC MACRO – PROC LOGISTIC

PROC LOGISTIC is run to create a model using the data set generated by the GENDAT macro and outputs a data set with predicted probabilities. The following code shows the LGSTIC macro which creates a model and calls the optimization macros. The output data set PROBS3 contains the variable phat, the estimated probability that the mobile associated with this record will churn.

```
%macro lgstic;
%put &market  &run &a&b&c&d&e&f;
title  "Model: Market=&MARKET, Run=&run";
title2 "Varlist=&a.&b.&c.&d.&e.&f";
proc logistic data=analysis descending noprint;
     model churn = &modvar1 &modvar2 &modvar3 &modvar4
                   &var1 &var2 &var3 &var4 &var5 &var6
             / lackfit rl rsquare;
  output out=probs3(keep= mobile pchurn churn phat)
         predicted=phat;
  run;
/*proc univariate data=probs3; var phat; run; */
/*** Call model evaluation macros ***/
%brier;
%clss ;
%cost(10,400,.15);
%mend lgstic;
```

```
DATA PROBS3

Churn   Pchurn  Phat
0       .       0.001066
0       .       0.000930
.       0       0.000339
1       .       0.191215
.       0       0.000901
.       1       0.114509
.       0       0.040012
0       .       0.009543
.       1       0.762340
0       .       0.023344
```

BRIER MACRO – CALCULATE BRIER SCORE

The Brier Score is calculated on the output data set. Brier scores may range from 0 to 1, with the smaller the score the better the predictive ability of the model. Our target variable 'churn' is either 0 or 1. What we are hoping for is to build a model where records where churn=0 have a phat close to 0 and records where churn = 1 have a phat close to 1. The Brier score is stored in the Brier score data set.

```
%macro brier;
  data probs4;
       set probs3;
       brier = (phat-pchurn)**2;  /* valid brier score */
       run;

  proc means data=probs4 noprint;
       var brier ;
       output out=brier1 n=n  mean=;
       run;
```

```
data brier1 ;
     drop _type_;
     set brier1;
     run = &run;
     varlist=  &a.&b.&c.&d.&e.&f;
     run;

data churn.brieratl;
     update churn.brieratl brier1;
     by varlist run ;
     run;
%mend brier;.
```

CLSS MACRO – CLASSIFICATION TABLE

The following PROC FREQ output is an example of a classification table. The number of records correctly predicted to be a churner (Pchurn = 1 and Predict = 1) or a non-churner (Pchurn = 0 and Predict = 0) and the percent of total churn predicted in the top 10% of scored records are the two optimization criteria calculated by the CLSS macro. Only validation records are included in this step. The 5,579 records used by PROC LOGISTIC to estimate the model have a missing value for pchurn.

```
TABLE OF PCHURN BY PREDICT
PCHURN     PREDICT(P=.14)

FREQUENCY|
ROW PCT  |       0|       1|  TOTAL
---------+--------+--------+
       0 |   9350 |    590 |   9940
         |  94.06 |   5.94 |
---------+--------+--------+
       1 |    138 |    123 |    261
         |  52.87 |  47.13 |
---------+--------+--------+
TOTAL        9488      713    10201

    FREQUENCY MISSING = 5579
```

To compare the predicted churners to actual churners we need to define a new variable which represents a predicted churner. One needs a probability breakpoint to calculate the new variable. To determine the cutoff point, PROC UNIVARIATE is run on the variable phat and the cutoff point is set to the 90th percentile value. This means those records whose predicted probability score is in the top 10% of all probability scores will be defined as a 'Predicted Churner'. The bottom 90% will be defined to be a non-churner. Thus, data set PROBS5 has both the predicted response and the actual response. One can now use PROC FREQ to calculate a classification table. The values are stored in the Class data set.

```
%macro clss;
data probs5;
     set probs3;
     label predict ='P=.14';                    /*****  Set Probability Breakpoint ********/
     if phat >=.14  then predict =1;
       else predict  = 0;
     run;
title1 "Churn Modeling: Market=&market  ";
proc freq data=probs5;
     tables  pchurn * (predict) / out=probfreq nocol nopercent;
     title2 "Classification Table";
     run;
data probs6;
     varlist=  &a.&b.&c.&d.&e.&f;
     run = &run;
     set probfreq end=eof;
     retain pct0 pct1 pct2 pct3 totnpd 0;
     if pchurn = 1 and predict = 1 then pct1 = percent;         /* % churners predicted  */
        else if pchurn = 0 and predict = 0 then pct0 = percent; /* % non-churners predict*/
     pct2 = sum(pct0,pct1);                                     /* % total predicted     */
     if pchurn = 1 then totchurn + count;                       /* count churners        */
     if pchurn = 1 and predict = 1 then churn=count;            /* Correct pred churns   */
     if eof then do;
        pct3 = churn / totchurn;                                /* % total churn pred    */
        output;
        end;
     drop pchurn predict count percent;
     run;
data churn.classatl ;                                           /*update class eval data set*/
     update churn.classatl probs6;
     by varlist run ;
     run;

%mend clss;
```

COST MACRO – COST / BENEFIT OPTIMIZATION

The COST macro provides a method of optimizing a model based on the estimated cost of treating the top x percent of the cohort and estimated benefit from predicted response. For example, in the LGSTIC macro the call to the COST macro was %COST(10,400,.15). This means it will cost $10 to treat each record, there is a $400 benefit to each correct prediction and we want top treat the top 15% based on predicted score. The values are stored in the Cost evaluation data set.

```
%macro cost(cost,benefit,percent);                  /*****************************************/
proc sort data=probs3 out=probs7;                   /* Macro Parameters                      **/
     by descending phat;                            /* cost – cost per treatment             **/
     where pinvchrn ne .;                           /* benefit – value of correct prediction **/
     run;                                           /* Percent – percent of records treated  **/
data probs8;                                        /*****************************************/
     varlist=  &a.&b.&c.&d.&e.&f;
     run = &run;
     retain totcost totbenefit;
     set probs7 end=eof ;
     if _n_ / &tot_val le &percent then do;         /* tot_val value set in GENDAT macro     **/
        totcost + &cost;
        if pinvchrn = 1 then totbenefit + &benefit;
        end;
     if eof then do;
        profit = totbenefit - totcost;
        output; end;
     run;
data churn.costatl ;                                /* update cost eval data set             **/
     update churn.costatl  probs8;
     by varlist run ;
     drop pinvchrn phat ;
     run;
%mend cost;
```

SHUFFLE MACRO

The previous macros describe one run through the modeling and optimization process. The next step is to run the same process using the same model evaluation data set using different combinations of explanatory variables on the model statement. The SHUFFLE macro will produce 64 different models by setting the macro variables var1 – var6 listed on the model statement in the LGSTIC macro to either an explanatory variable or blank. All of the optimization macros will be run on each of the 64 models produced and the results stored in the evaluation data sets.

One call to the SHUFFLE macro provides the modeler with data to compare 64 different models on different evaluation criteria. The %do macro variables a, b, c, d, e and f resolve to either 1 or 2. These values also populate the varlist variable in the optimization macros. Varlist 111111 is the model with all six test explanatory variables in the model while varlist = 222222 is the model with none of the six test explanatory variables.

```
%macro shuffle(v1,v2,v3,v4,v5,v6);
%do a = 1 %to 2;
  %do b = 1 %to 2;
     %do c = 1 %to 2;     %do d = 1 %to 2;     %do e = 1 %to 2;     %do f = 1 %to 2;

%if &a = 1 %then %do;
         %let var1 = &v1;  %end;
      %else %do;
         %let var1 =  ;    %end;
     %if &b = 1 %then %do;
         %let var2 = &v2;  %end;
      %else %do;
         %let var2 = ;     %end;
     %if &c = 1 %then %do;
         %let var3 = &v3;  %end;
      %else %do;
         %let var3 = ;     %end;
     %if &d = 1 %then %do;
         %let var4 = &v4;  %end;      %else %do;         %let var4 = ;     %end;
     %if &e = 1 %then %do;
         %let var5 = &v5;  %end;      %else %do;         %let var5 = ;     %end;
     %if &f = 1 %then %do;
         %let var6 = &v6;  %end;      %else %do;         %let var6 = ;     %end;

      %lgstic;                                        /* Call LGSTIC macro */
     %end;
   %end;
  %end;  %end;%end;%end;
run;

%mend shuffle;
```

BOOT MACRO

Why stop at one run? The BOOT macro will run this process a selected number of times. Each run through the process uses a different modeling and validation data set. Obviously this process can be time and compute intensive. Twenty runs through the process generates 1,280 evaluation records.

```
   %macro boot;
   /*************************************************************************/
   /** create random sample, run logistic, analyze models,revclass=4       **/
   /*************************************************************************/
   %do run = 1 %to  10;
   %global modvar1 modvar2 modvar3 modvar4 modvar5;
   %gendat;
   %let modvar1 = %str( acc_negx airhighx chaffx chtelsalx clair_posx clairlenhighx  );
   %let modvar2 = %str(cltoldropx cont_newx f_rsx f_insx ftr_negx ftr_posx ftruserx );
   %let modvar3 = %str( irdbx7 irdbx13 mltyr48x mourom_posx old_contx peakuserx pk_negx ) ;
   %let modvar4 = %str( rev_negx revhighx romuserx tech_gsmx tech_tdmx ) ;
   %shuffle(acchighx, clairdropx, clairzerox, ftrlowx,pkmdropx,rev_posx  );
   %end;
   %mend boot;

   %boot;
```

EVALMOD MACRO

The final step is to evaluate the models. The EVALMOD macro provides this analysis. Each optimization statistic is averaged and ranked by model. The last data step calculates two additional rankings. The first ranking is a sum of the four optimization criteria rankings. The second ranking doubles the rank of the optimization criteria of the statistic for percent of total churn predicted.

```
%macro evalmod;
%let mkt=atl;
data brier; set mod.brier&mkt;
     run;
/* Avg brier score by model   */
proc summary data=brier nway;
     class varlist ;
     var brier ;
     output out=brier2
            mean=m_b ;
     run;
 /* Rank by valid brier score   */
proc rank data=brier2 out=brier3 ;
              var m_b ;
              ranks m_br;
              run;

proc sort data=brier3;
             by varlist;
             run;


….
/**** Similar code to process CLSS and
COST evaluation data sets **/
```

```
data rank3 ;
 merge brier3 class3 cost3;
  by varlist;
 rank = m_cr + m_cr1 + m_br + m_pr;
 mod_cr1 = (m_cr1 * 2);
 rank_cr1 = m_cr + mod_cr1 + m_br + m+pr;
 label rank= 'Overall Rank'
       rank_cr1 = 'Mod  Rank';
 run;
proc sort data=rank3 ;
        by rank_cr1;
        run;
proc print data=rank3 label;
     by revenue;
     var varlist _freq_ rank rank_npd
     m_br m_cr m_cr1 ;
     label m_b1 = 'Model'
     m_b = 'Mean Brier' _freq_ = 'Runs'
        m_br=’Rank of Brier Score’
        m_cr  = 'Rank Total Sample Class'
       m_cr1 = 'Rank of Churn Class';
       Run;
%mend evalmod;
%evalmod;
```

The following shows a sample model evaluation report. The varlist column defines the model. The top model according to both the Overall and Mod Rank is model 221112. This is the model which did not include the test explanatory variables a, b and f, but includes c, d and e. The runs column shows how many runs through the modeling process were completed. This was 10 runs for this output.

The Overall Rank is the sum of the 4 optimization rankings. Note that the top model is only ranked number 1 on the Rank of Churn Class, i.e. the percent of total churn correctly predicted. The top model is only the 5th best model for Profit, 4th for Brier Score and 3rd for correctly predicting churn and non-churners.

The model represented by varlist = 111111 uses all six explanatory variables. This model is essentially equivalent to the model produced by PROC LOGISTIC using a Stepwise option on the entire modeling database.

Obs	varlist	Runs	Overall Rank	Mod Rank	Rank of Brier Score	Rank Total Sample Class	Rank of Churn Class	Rank of Profit	Mean Brier
1	221112	10	13.0	14.0	4	3	1	5.0	0.022990
2	121112	10	15.0	18.0	1	4	3	7.0	0.022959
3	121111	10	26.0	28.0	3	13	2	8.0	0.022988
4	111112	10	23.0	32.0	7	6	9	1.0	0.023002
5	211112	10	28.0	32.0	13	9	4	2.0	0.023038
6	222112	10	29.0	34.0	6	5	5	13.0	0.022995
7	112112	10	28.0	41.0	8	1	13	6.0	0.023004
8	122112	10	30.5	42.5	2	2	12	14.5	0.022963
9	111111	10	33.0	44.0	11	8	11	3.0	0.023032
10	211111	10	38.0	45.0	15	12	7	4.0	0.023063
11	122111	10	37.5	45.5	5	15	8	9.5	0.022991
12	212112	10	41.0	47.0	14	10	6	11.0	0.023042
13	221111	10	44.5	54.5	9	16	10	9.5	0.023013
14	112111	10	45.0	59.0	12	7	14	12.0	0.023034
15	222111	10	54.5	70.5	10	14	16	14.5	0.023018
16	212111	10	58.0	73.0	16	11	15	16.0	0.023066

CONCLUSION

This paper has provided a method of incorporating user defined optimization criteria to the modeling process. This methodology is not limited to Logistic regression but can be applied to any modeling process where actual response and predicted probabilities are available to the modeler.

REFERENCES

SAS Institute Inc. *Logistic Regression Examples Using the SAS® System, Version 6, First Edition,* Cary, NC: SAS Institute Inc., 1995. 163 pp.

ACKNOWLEDGMENTS

Acknowledgments go after your references. This section is not required.

CONTACT INFORMATION

(In case a reader wants to get in touch with you, please put your contact information at the end of the paper.)
Your comments and questions are valued and encouraged. Contact the author at:

Jerry Musial
Cingular Wireless, Inc
5565 Glenridge Connector
Atlanta, GA 30342
404-236-6853
404-236-6832
jerry.musial@cingular.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 trademarks of their respective companies.