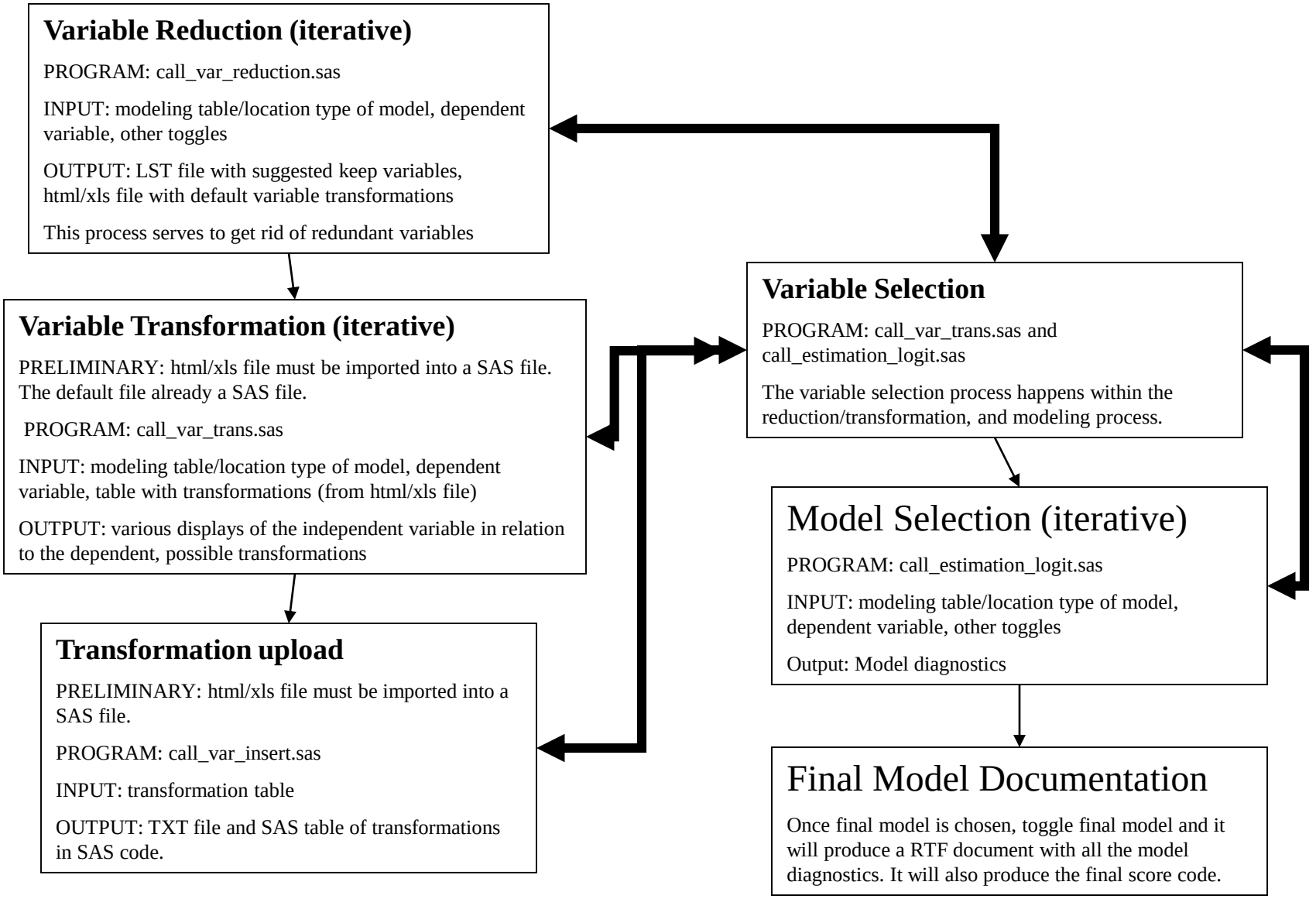
Customer Analytics

Modeling Process

Using the DePoint Modeling Macros and Procedures

Work Flow


Variable Reduction (iterative)
PROGRAM: call_var_reduction.sas
INPUT: modeling table/location type of model, dependent variable, other toggles
OUTPUT: LST file with suggested keep variables, html/xls file with default variable transformations
This process serves to get rid of redundant variables
Variable Transformation (iterative)
PRELIMINARY: html/xls file must be imported into a SAS file. The default file already a SAS file.
PROGRAM: call_var_trans.sas
INPUT: modeling table/location type of model, dependent variable, table with transformations (from html/xls file)
OUTPUT: various displays of the independent variable in relation to the dependent, possible transformations
Transformation upload
PRELIMINARY: html/xls file must be imported into a SAS file.
PROGRAM: call_var_insert.sas
INPUT: transformation table
OUTPUT: TXT file and SAS table of transformations in SAS code.
Variable Selection
PROGRAM: call_var_trans.sas and call_estimation_logit.sas
The variable selection process happens within the reduction/transformation, and modeling process.
Model Selection (iterative)
PROGRAM: call_estimation_logit.sas
INPUT: modeling table/location type of model, dependent variable, other toggles
Output: Model diagnostics
Final Model Documentation
Once final model is chosen, toggle final model and it will produce a RTF document with all the model diagnostics. It will also produce the final score code.

Call_var_reduction.sas

- This procedure performs a variable cluster routine, then a Pearson's correlation and a chi-squared test of significance
  - median value temporarily inserted for missing values to make the routine work
- It clusters the variables and then chooses the "best" variable in the cluster according to the
    - 1-rsq ratio = $(1 - rsq_{\text{own cluster}})/ (1 - rsq_{\text{next cluster}})$
    - the variables from cluster can be modified by user.
- A Pearson's correlation between dependent variable and independent variable is performed. This captures linearity.
- A chi-squared ratio test of a bucketed independent variable and the dependent variable (H_o: no difference across bucket) in order to capture any non-linearities. Missings are treated as their own bucket.
- The user chooses the top variables to keep from both the Pearson's and the Chi-Squared tests.
- A table that is exported to html/xls shows some default transformations, to be modified later.

Call_var_reduction.sas

- /* Call the variable reduction program*/
- /* DEPOINT oct 2002*/
- /* updated may 2003 April 2004*/
- /* supply a modeling data set and it will give you a table with a list of variables and their specific statistics*/
- /* it ranks variables by their pearson's correlation coefficient and their CHISQ value */
- /* needs an include for the modeling_macros.sas */

- %let model_table = ; /*table you will model on*/
- %let model_lib = ; /*model table location*/
- %let DEPVAR = ; /*name of the dependent variable*/
- %let ignore_list = ; /*variables you do wish to ingore as potential independent variables*/
- %let must_keep_list = ; /*variables you want to save through the varclus procedure */
- %let out_table = var_reduction&model_table; /*name to call the output table*/
- %let out_lib = ; /*output table library name*/
- %let keep_amt = 75; /*specify the number of variables you want to keep, as ranked by PCORR or CHISQ*/
- %let var_type_flag = ; /*LEAVE blank for now*/

- %include '/home/mca/opencode/modeling_var_reduction.sas';

Call_var_trans.sas

- This procedure performs bi-variate and uni-variate analysis on the data, given the transformation table supplied by the user.
- It produces a Pearson's Correlation coefficient between the dependent and each independent variable for multiple transformations (sqroot, x-squared, x-cubed, a dummy variable at the median value, and log(x + 1))
  - It chooses the "best" transformation based upon the highest absolute value of the Pearson's and produces more diagnostics on the raw and the "best" transformation.
- It produces a E-LOGIT plot for the binned (proc rank) independent variable
- It produces a Mean of dependent plot and table for a binned (proc rank) independent variable
- It produces a two-sample KS for the dependent against the independent.
  - Remember, it does this all for the raw independent variable and the "best" transformation.
- It produces UNIVARIATE diagnostics for the raw independent.
- NOTE: "raw" above refers to the definitions given to the procedure by the transformation table supplied by the user. This function allows you to test your transformations by running the procedure again.

Call_var_trans.sas

```
/* Call the variable transformation program*/
/* DEPOINT oct 2002*/
/* updated May 2003 April 2004*/
/* supply a modeling data set and a table with a variable list */
/* and it will give you an output list with multiple transformations, e-logit plots, and PROC UNIVARIATE results.*/

%let model_table    = ;     /*table you will model on*/
%let model_lib      = ;     /*model table location*/
%let DEPVAR         = ;     /*name of the dependent variable*/
%let var_table      = ;     /*name of the var list table*/
%let var_lib        = ;     /*library of the varlist table*/
%let Transf_test    = N;    /*if you are testing transformation you all ready choose, put Y. First time call*/
                            /* leave at N      */
%let key_var                = varname;
%let var_type_flag  = ;     /*LEAVE BLANK FOR NOW*/

%include '/home/mca/opencode/modeling_var_trans.sas';
```

Call_var_insert.sas

```
/* Call the transformation insertion program*/
/* DEPOINT oct 2002*/
/* updated may 2003 April 2004*/
/* supply a table with variable names*/

%let var_lib                    = ;             /*lib for var table*/
%let var_table                  = ;             /*var table name*/
%let key_var                    = varname;      /*name of variable with variable names*/
%let out_lib                    = &var_lib;
%let out_table                  = ;
%let export_type        = SAS;  /*either SAS*/
%let var_type_flag      = ;     /*LEAVE blank*/

%include '/home/mca/opencode/modeling_insert_trans.sas';
```

This just takes the final transformation definitions from the html/xls table imported into SAS and creates a transformation table in a SAS table and a txt table

Call_estimation_logit.sas

- Creates a Development and Validation dataset on the fly, either with a set random seed (for the same D and V) or a new random seed (boot strapping).
- Calculates dynamically the correct p-value for SLSSTAY based upon an approximation of the BAYESIAN-INFORMATION CRITERIA.
  - 1-probchi(2*log(sum(responders)),2)
- Performs a BACKWARD FAST selection criteria
- Creates diagnostics for Multicollineary (variance inflation/tolerance)
- Displays the bi-variate Pearson's correlation signs (a priori assumtion) and compares it to the sign from the model.
- Displays the standardized error, the wald chi-sqrd, the p-value, the adjusted r-squared (D and V) the c-value (area under the ROC curve) the two-sample KS (D and V), and the correlation matrix. The odds-ratio and the delta-p are available.
- Creates gains charts for the Validation data set using the estimates from the Development.
- Creates the ROC curve and the diagnostics available from that calculation.
- Creates the mean of XBETA to be used in the final score code to scale the score (if desired)

Call_estimation_logit.sas FINAL MODEL

- Outputs the above stated diagnostics to an RTF file
- Outputs the means of transformed table
- Outputs the Mean of dependent plots and table, the KS values, for each final variable.
- Outputs the raw score code with
  - transformations
  - equations
  - raw predicted values
  - the score (“flipped” for risk models, so larger values are always better)
  - canned exclusion scores with the option to add more
  - canned adverse action reason codes based upon the “difference from the mean” method
- It creates two copies of the score code, just in case.

Call_estimation_logit.sas

```
/* Call the model estimation logit program*/ /* DEPOINT oct 2002*/ /* updated may 2003 April 2004 */ /* supply a modeling data set */
%let model_table      = ;                    /*table you will model on*/
%let model_lib        = model;                                        /*model table location*/
%let DEPVAR           = ;                                             /*name of the dependent variable*/
%LET DEV_VAL_SPLIT    = .50;        /*how many do you want in developement and validation*/
%let RANDOM_SEED      = 0;/*do you want a fixed or random seed? if random, then value=0 */
%let var_table        = ;           /*a table that contains the potential independent variables*/
%let var_lib          = ;           /*the location of said table*/
%LET key_var          = varname;    /*the variable in the list that has the VARNAMES as values*/
%let trans_table      = ;           /*the test list of transformations*/
%let trans_lib        = &var_lib;   /*the location of said table in UNIX format. IF NO TRANS, then leave blank.*/
%let RISK_MODEL       = ;           /*if Risk model, then put Y, leave blank otherwise*/
%let ignore_vars      = ;           /*variables you want to remove as potential independent variables*/

/*************************************************************************/
/* FINAL ANALYSIS                                                        */
/*************************************************************************/
%let final_ind_var = ;
%let FINAL            = N; /*If Final Estimation then label (Y)es. Final Estimation will output parameter tables with diagnostics*/
                           /* to a RTF file*/
%let export_type      = SAS;        /*SAS*/
%let final_LOC                      = ;  /* in UNIX format*/
%let final_name                     = FINAL1&model_table;
%include '/home/mca/opencode/modeling_estimation_logit.sas';
```