**Assignment #6**

James Gray

**Introduction:**

This study will fit a multiple logistic regression model for a binary response variable to the **credit\_approval** data set and assess its predictive accuracy. We will then compare the predictive performance of our multiple logistic regression model (Model #1) to the predictive performance of a pre-specified model (Model #2). During the modeling process we will use a statistical methodology called *cross-validation* to train and test these models. This will provide a method to determine how well the model responds to new data. Finally, we use a set of statistics and diagnostics to evaluate the two models and select an optimal model.

**Results:**

### In-Sample Results

In this part of the study we will use the cross-validation methodology to split the sample data. 70% of the sample will be used for model training and 30% for testing the predictive accuracy of the model. In this section we will build the model using the in-sample data, and then use the out-of-sample data in a later part of the study to determine how well the model predicts “new” data. The credit\_approval data set includes 15 independent variables (6 continuous, 9 discrete) and a discrete dependent variable (A16) used to identify whether the credit request is approved or not (“+” is yes, “-“ is no). The first step of this process uses a uniform random variable to execute the 70/30 data split. A new variable, train, is appended to the data set to flag the observation as either part of the training sample (train=1) or the testing sample (train=0). The next step evaluates the observed response variable (A16) and sets a new response variable Y equal to ‘1’ for values ‘+’ and equal to ‘0’ for values ‘-‘. This step prepares the data set for logistic regression now that we have a binary response variable.

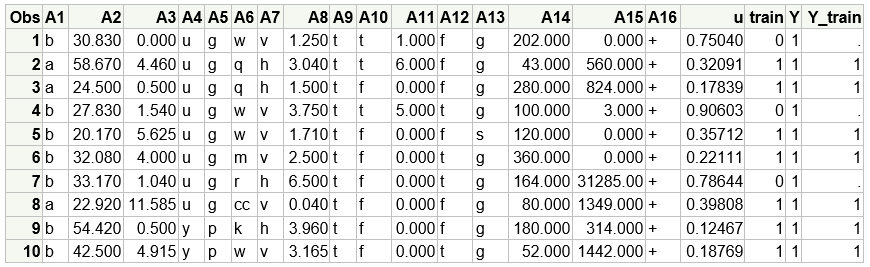


Figure - Credit\_Approval data set with cross-validation sample assignments

The next step of the data preparation process specifies design variables for the independent discrete variables given that non-numerical values cannot be combined directly into a model fitting process with continuous variables. Design variables were specified for each of the 9 categorical variables using k-1 variables for a categorical variable with k categories. For example, the A5 variable has 3 different categories shown in the dataset (g, gg, p) and therefore 2 design variables (A5\_gg, A5\_p) are required to represent the 3 states. The “base” category is generally the category value with the smallest number of observations. The results of an Exploratory Data Analysis (EDA) from a previous assignment are presented in Figure 2 that shows the frequency distribution for the 9 categorical predictor variables and response variable (A16).

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

Figure - Frequency distributions for categorical variables

The final step of this phase removes any observations that have missing values in any of the independent variables. Missing values are represented by ‘.’ and ‘?’ for continuous and categorical values respectively. The original data set contains 690 observations and 37 observations with a least one missing value leaving 653 for model fitting.

#### Model #1 – Backward Elimination

In this part of the study we will fit a logistic regression model using backward variable selection. Backward elimination is an iterative process that first considers all predictors and drops variables one at a time on the basis of their contribution to the reduction of error sum of squares (Chatterjee & Hadi, 2012). The process concludes when all variables are deleted or all variables with statistical significance are selected. The design variables A4\_y, A5\_p, A6\_j, A7\_n, A7\_dd, A7\_j and A7\_z were not included in the model fitting due to the very small number of occurrences when compared to the other variables.

Using backward selection option of the SAS PROC LOGISTIC procedure, the backward selection summary table and parameter estimates are generated (Figure 3). The logit for this model is thus

g(x) = -3.0087 + 0.2338\*A11 + 0.00056\*A15 – 2.2218\*A7\_ff + 3.5735\*A9\_t (Model #1)

The probability function is derived from the logit as follows:

|  |  |
| --- | --- |
|  |  |

Figure - Backward Elimination Summary and Parameter Estimates

Now that the model is fit we can begin the process of evaluating the variables in the model to determine whether the predictors are statistically significant to the response variable. The first step is to evaluate the overall significance of the model by interpreting the Likelihood Ratio. This statistic, also known as deviance, is the kin to sum-of-squares error (SSE) in linear regression. The Likelihood Ratio is significant at the 0.0001 level and therefore we can conclude one more of the predictors are non-zero. We can now move forward to evaluate the statistical significance of each predictor. The Wald Chi-Square statistics for the constant (intercept) and independent variables are statistically significant at different levels although all are acceptable. The constant and A9\_t are the most significant (p < 0.001), followed by A11 (p=0.001), and A15 and A7\_ff (p < 0.01). Although all logit dependent variables are statistically significant, the Odds Ratio (OR) for each variable provides the greatest insight into understanding model behavior. The odds of a positive credit approval (y=1) is 35.6 times greater when the value of A9\_t =1. The other variables have a minimal impact on a positive credit approval when compared to A9\_t.

We will now evaluate how accurately Model #1 reflects the true response outcome of the credit\_approval data set. When we assess goodness of fit we are comparing the fitted values to the observed values, where we can think of the observed values as being from the best possible (saturated) model (Hosmer, Lemeshow & Sturdivant, 2013). The goodness of fit statistics generated for Model #1 are shown below in Figure 4.

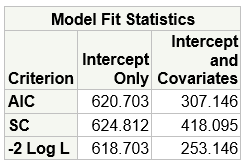


Figure - Model #1 Goodness of Fit Statistics

The goodness of fit statistics in Figure 4 includes statistics for a model with no predictors (intercept only) and the fitted model (Model #1). Higher values imply a worse fit so clearly the fitted variables have predictive power due to the significant reduction in the statistics for Model #1. There is not an absolution standard for what’s considered a good fit for the -2LogL statistic, so one can only use this statistic to compare different models fit to the same data set (Allison, 2012). The Akaike Information Criterion (AIC) and Bayesian Information Criteria (BIC) statistics penalize models with more predictors so these are effective measures to compare competing models. We will use these statistics as comparative measures to Model #2 in an upcoming step.

At this point of our analysis we have confirmed that the variables are significantly related to the response variable and the predicted response is accurate when compared to the actual data. Another step in evaluating model adequacy is to assess the predictive power of the model. Here we are determining the degree to how well the model can predict the actual response variable given the predictor values. It should be noted that we can have model that is considered a good fit, but does not accurately predict the response. We could also have the reverse where the model is not a good fit but does accurately predict the response. Four measures of association are shown in Figure 5 to support this analysis.

|  |  |
| --- | --- |
| Figure - Model #1 measures of association | C = concordant pairs, D = discordant pairs, T= Tied, N = # of pairs before elimination  Somers’s D = (C-D)/(C+D+T)  Gamma = (C-D)/(C+D)  Tau-a = (C-D) / N |

These values are derived through the use of a classification table where we can compare the predicted value to the actual response value. The analysis is reduced to evaluating pairs of observations where the response is 0 or 1 (50049). Pairs are considered *concordant* when the observation where the response is 1 has a higher predicted probability than the observation where the response is 0. Cases where this is not the case is considered *discordant*. We can conclude that Model #1 has a high predictive power given that 91.6% of the cases are classified properly, 5.4% are misclassified and 3.0% are tied. The four measures (Somers D, Gamma, Tau-a and c) vary between 0 and 1 with higher values implying a stronger association between predicted and observed values (Allison, 2012).

It is often useful to evaluate diagnostics for each observation to evaluate their impact on the model. First we will look for any outliers in the data by evaluating how much any of the predictor coefficients change when an observation is removed from the analysis. The SAS DFBETAS option generates the plot for the constant and the four predictors in Figure 6 below. The plots do not show any significant changes (>1) and therefore no outliers exist.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Figure - DFBETAS statistic for Model #1

Deviance statistics produced by SAS highlight the change in deviance with the deletion of an observation. This can identify changes in error and points that are poorly fit. Observations with high values of influence and also well separated from other points are candidates (Allison, 2012). The observations highlighted by the red boxes may be candidates to review in more detail.

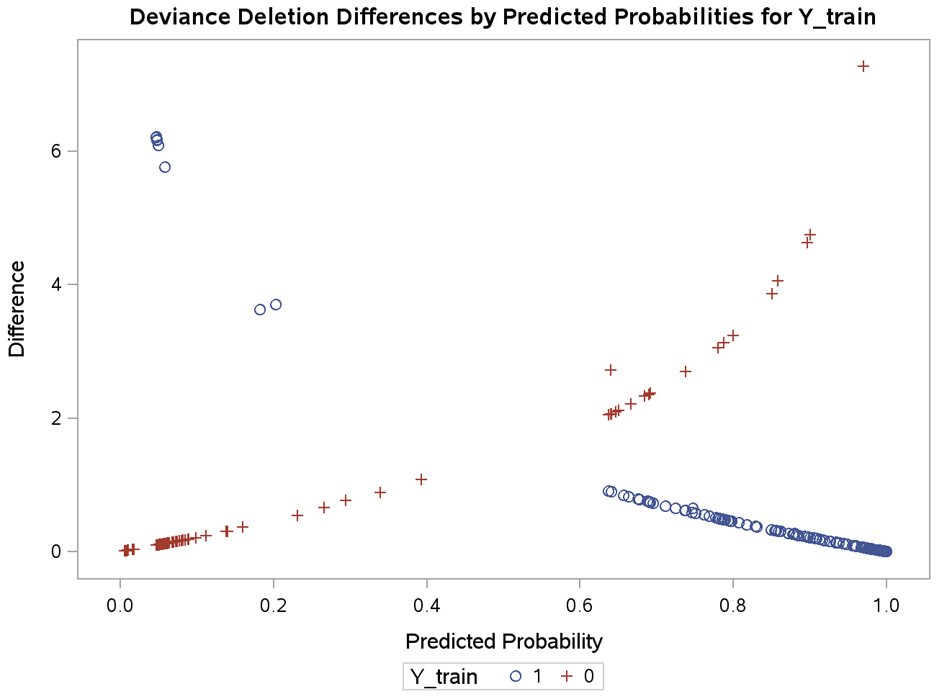


Figure - SAS DIFDEV for Model #1

#### Model #2 – Manager specified Model

In this section of the study another model is fit (Model #1) using a particular model provided by a manager. This reduced model includes three predictors A9\_t, A2 and A3. The model is fit and the parameter and odds ratios are shown below in Figure 8.

|  |  |
| --- | --- |
|  |  |

Figure - Model #2 Parameter and Odds Ratio Estimates

The logit for Model #2 is thus

g(x) = -3.6287 + 3.9836\*A9\_t + 0.0227\*A2 + 0.0527\*A3 (Model #2)

The probability function is derived from the logit as follows:

#### Model #1 and Model #2 Comparisons (In-Sample Data)

We will now compare Model #1 and Model #2 using the SAS parameter estimates and goodness-of-fit statistics.

| **Model #1 (Backward Elimination)** | **Model #2 (Manager)** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Figure - Model #1 and Model #2 Comparison

Both models are significant given that all three statistics in the Testing Global Null Hypothesis: Beta=0 confirm significance at the 0.0001 level. Analyzing in the individual parameters for significance confirms no changes for the constant and A9\_t but lower values of statistical significance in Model #2. Model #1 has a better goodness of fit compared to Model #2 due to the lower values of the AIC and -2LogL statistics. The measures of association are all consistently higher for Model #1 as well. At this part of the study, we can tentatively conclude that Model #1 is preferred to Model #2 based on the statistical significance of the parameters and goodness of fit. Additional considerations will be analyzed before we make a final conclusion.

A *lift chart* is another method to evaluate the predictive accuracy of a model. A lift chart graphically represents the improvement that a mining model provides when compared against a random guess, and measures the change in terms of a lift score (Microsoft). The insight derived from the visual can also help to properly apply the model predictions.

We will create a lift chart for Model #1 and Model #2 using the training data set. The lift chart for this analysis will plot the cumulative percentage of a positive credit approval (Y axis) by the cumulative percentage of all observations (X axis). This line is contrasted with a 45 degree line that represents a random outcome. The “lift” is calculated at any point along the X axis by dividing the predicted response by the random response. The lift represents the effect of the model compared to not using a model.

A lift chart and plot are generated from SAS for Model #1 and Model #2 as shown in Figure 10. By contrasting the lift charts we can compare the accuracy of the model at various points. As shown in the tables below the prediction rates for Model #1 are consistently higher as we approach maximum lift when compared to Model #2. Therefore Model #1 would be preferred to Model #2 on based on the accuracy delivered by the lift chart.

|  |  |
| --- | --- |
| **Model #1 (Backward Elimination)** | **Model #2 (Manager)** |
|  |  |
|  |  |

Figure - Figure 9 - Lift Charts for Model #1 and #2 using Training data set

At this point in the study we will now transition to evaluate the out-of-sample data and how well the models predict new data. It is assumed that the models will not predict new data better than the fitted data so it’s really a question of relativity.

### Out-of-Sample Results

In this part of the study we will conduct cross-validation using the out-of-sample test data to evaluate the potential for overfitting. We essentially will evaluate how well the model prediction performs using “new” data. An overfitted model is one that approaches reproducing the training data on which the model is build – by capitalizing on the idiosyncrasies of the training data (Ratner, 2012). An overfitted model generally has more predictors than required adding unneeded complexity. As the fit of the model increases by including more information, the predictive performance of the model on the validation data decreases (Ratner, 2012).

Again we will use lift charts to evaluate predictive accuracy. To create the lift charts for the out-of-sample data (train=0) we need to scale the data to the number of successes (95).

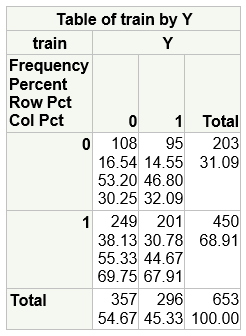


Figure -Frequency distribution for Cross-Validation data

Model #1 and Model #2 did not require to be fit specifically using the out-of-sample data set given that SAS automatically fits the model on the training data set when using the variable Y\_train. The lift charts for each model were calculated using SAS and displayed in Figure 12. Both models show consistently very minor decreases in predictive accuracy for the new data although no major discrepancies exist. Model #1 is consistently more accurate from 0% to 60% of the population and then both models taper off fairly consistently as they approach 100% of the population. Model #1 is thus the preferred model.

|  |  |
| --- | --- |
| **Model #1 – Test Data** | **Model #2 – Test Data** |
|  |  |
|  |  |

Figure - Lift Charts for Model #1 and #2 using Test data set

**Conclusions:**

In this study we prepared the credit\_approval data set for a multiple variable logistic regression and divided the data set into a training and testing samples for the purposes of conducting cross-validation. A model was fit to the training data set using the backward elimination method (Model #1). We concluded the model was adequate and a good fit after evaluating a series of parameter and goodness of fit statistics. A second pre-defined model (Model #2) was also fit to the training data set where we then compared the parameter statistics, goodness of fit and lift chart to Model #1 and concluded that Model #1 was preferred. In the last section of the study we evaluated the predictive accuracy of each model using the test data set to ensure overfitting did not exist. After reviewing the lift charts using the test data both models remained very accurate when compared to the lift charts based on training data. Model #1 also showed a slightly higher accuracy than Model #2 for the test data set. Overall Model #1 is the recommended model.

**Code:**

/\* James Gray

2013.07.31

graymatter@u.northwestern.edu

Assignment6\_JG.sas

\*/

/\* This code is for PREDICT 410 Assignment #6 - Multiple Logistic Regression Model.

This assignment will fit a multiple logistic regression model for a binary response

variable to the credit\_approval data set using PROC LOGISTIC and assess its predictive

accuracy. We will then compare the predictive performance of our multiple logistic

regression model to the predictive performance of a pre-specified model.

\*/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Get the data on the SAS server - mydata.credit\_approval

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**libname** mydata '/courses/u\_northwestern.edu1/i\_833463/c\_3505/SAS\_Data/' access=readonly**;**

**run;**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Review credit\_approval dataset metadata and 5 observations;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

title 'Credit\_Approval Data Set'**;**

**proc** **contents** **data**=mydata.credit\_approval**;** **run;** **quit;**

**proc** **print** **data**=mydata.credit\_approval**(**obs=**5);** **run;** **quit;**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Split the dataset into training and testing samples for cross-validation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**data** temp**;**

set mydata.credit\_approval**;**

/\* Assign each observation to either a training or testing dataset; The UNIFORM(SEED)

function generates values from a random uniform distribution between 0 and 1. The "seed"

is the initial starting point for the random numbers. \*/

u=uniform**(123);**

if **(**u<**0.7)** then train=**1;**

else train=**0;**

\* create output variable Y and set based on value in A16;

if **(**A16='+'**)** then Y=**1;**

else if **(**A16='-'**)** then Y=**0;**

else Y=**.;**

\* create a response indicator based on the training/testing split;

if **(**train=**1)** then Y\_train=Y**;**

else Y\_train=**.;**

\* create design vars for A1 with 2 categories (a,b) with 'a' as base (smallest);

if **(**A1='b'**)** then A1\_b=**1;** else A1\_b=**0;**

\* create design vars for A4 with 3 categories (l,u,y) with 'l' as base (smallest);

if **(**A4='u'**)** then A4\_u=**1;** else A4\_u=**0;**

if **(**A4='y'**)** then A4\_y=**1;** else A4\_y=**0;**

\* create design vars for A5 with 3 categories (g,gg,p) with 'gg' as base (smallest);

if **(**A5='g'**)** then A5\_g=**1;** else A5\_g=**0;**

if **(**A5='p'**)** then A5\_p=**1;** else A5\_p=**0;**

/\* create design vars for A6 with 13 categories (aa,c,cc,d,e,ff,i,j,k,m,q,r,w,x) with

'r' as base (smallest) \*/

if **(**A6='aa'**)** then A6\_aa=**1;** else A6\_aa=**0;**

if **(**A6='c'**)** then A6\_c=**1;** else A6\_c=**0;**

if **(**A6='cc'**)** then A6\_cc=**1;** else A6\_cc=**0;**

if **(**A6='d'**)** then A6\_d=**1;** else A6\_d=**0;**

if **(**A6='e'**)** then A6\_e=**1;** else A6\_e=**0;**

if **(**A6='ff'**)** then A6\_ff=**1;** else A6\_ff=**0;**

if **(**A6='i'**)** then A6\_i=**1;** else A6\_i=**0;**

if **(**A6='j'**)** then A6\_j=**1;** else A6\_j=**0;**

if **(**A6='k'**)** then A6\_k=**1;** else A6\_k=**0;**

if **(**A6='m'**)** then A6\_m=**1;** else A6\_m=**0;**

if **(**A6='q'**)** then A6\_q=**1;** else A6\_q=**0;**

if **(**A6='w'**)** then A6\_w=**1;** else A6\_w=**0;**

if **(**A6='x'**)** then A6\_x=**1;** else A6\_x=**0;**

\* create design vars for A7 with 9 categories (bb,dd,ff,h,j,n,o,v,z) with 'o' as base (smallest);

if **(**A7='bb'**)** then A7\_bb=**1;** else A7\_bb=**0;**

if **(**A7='dd'**)** then A7\_dd=**1;** else A7\_dd=**0;**

if **(**A7='ff'**)** then A7\_ff=**1;** else A7\_ff=**0;**

if **(**A7='h'**)** then A7\_h=**1;** else A7\_h=**0;**

if **(**A7='j'**)** then A7\_j=**1;** else A7\_j=**0;**

if **(**A7='n'**)** then A7\_n=**1;** else A7\_n=**0;**

if **(**A7='v'**)** then A7\_v=**1;** else A7\_v=**0;**

if **(**A7='z'**)** then A7\_z=**1;** else A7\_z=**0;**

\* create design vars for A9 with 2 categories (f,t) with 'f' as base (smallest);

if **(**A9='t'**)** then A9\_t=**1;** else A9\_t=**0;**

\* create design vars for A10 with 2 categories (f,t) with 't' as base;

if **(**A10='t'**)** then A10\_t=**1;** else A10\_t=**0;**

\* create design vars for A12 with 2 categories (f,t) with 'f' as base;

if **(**A12='t'**)** then A12\_t=**1;** else A12\_t=**0;**

\* create design vars for A13 with 3 categories (g,p,s) with 'p' as base (smallest);

if **(**A13='g'**)** then A13\_g=**1;** else A13\_g=**0;**

if **(**A13='s'**)** then A13\_s=**1;** else A13\_s=**0;**

\* delete missing values - categorical="?", continuous="-";

if **(**A1='?'**)** OR

**(**A2=**.)** OR

**(**A3=**.)** OR

**(**A4='?'**)** OR

**(**A5='?'**)** OR

**(**A6='?'**)** OR

**(**A7='?'**)** OR

**(**A8=**.)** OR

**(**A9='?'**)** OR

**(**A10='?'**)** OR

**(**A11=**.)** OR

**(**A12='?'**)** OR

**(**A13='?'**)** OR

**(**A14=**.)** OR

**(**A15=**.)**

then delete**;**

**run;**

\* review 10 observations of final data set;

title 'Credit\_Approval Data Set after data cleansing'**;**

**proc** **print** **data**=temp**(**obs=**10);** **run;** **quit;**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Fit the model using Backward Elimination variable selection (Model #1)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

/\* note that design vars A4\_y, A5\_p, A6\_j, A7\_n, A7\_dd, A7\_j, A7\_z were excluded due to very

small number of frequencies. The variable PHAT contains the predicted probabilities of the

dependent variable. The new variable model\_data created by the OUTPUT statement contains

all of the variables in the model including PHAT.

\*/

title 'Model #1 Logistic Regression Fitting'**;**

\* proc logistic data = temp descending plots=phat (UNPACK);

**proc** **logistic** **data** = temp descending plots**(**UNPACK**)**=**(**INFLUENCE DFBETAS PHAT**);**

model Y\_train = A2 A3 A8 A11 A14 A15

A1\_b A4\_u A5\_g

A6\_aa A6\_c A6\_cc A6\_ff A6\_i A6\_k A6\_m A6\_q A6\_w A6\_x

A7\_bb A7\_ff A7\_h A7\_v

A9\_t A10\_t A12\_t A13\_g / selection=backward AGGREGATE**;**

output out=model\_data pred=yhat**;**

**run;**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Fit the Model #2 assigned by manager

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

title 'Model #2 Logistic Regression Fitting'**;**

**proc** **logistic** **data**=temp descending**;**

model Y\_train = A9\_t A2 A3**;**

output out=model\_data2 pred=yhat**;**

**run;**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Fit the Model #2 assigned by manager and create a Lift Chart

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

title 'Model #2 Fitting for Lift Chart'**;**

**proc** **logistic** **data**=temp descending**;**

model Y\_train = A9\_t A2 A3**;**

output out=model\_data2 pred=yhat**;**

**run;**

\* The descending option assigns the highest model scores to the lowest score\_decile;

**proc** **rank** **data**=model\_data2 out=training\_scores descending groups=**10;**

var yhat**;** ranks score\_decile**;** where train=**1;** **run;**

\* To create the lift chart run this exact code;

**proc** **means** **data**=training\_scores sum**;**

class score\_decile**;** var Y**;**

output out=pm\_out sum**(**Y**)**=Y\_Sum**;** **run;**

**proc** **print** **data**=pm\_out**;** **run;**

**data** lift\_chart**;** set pm\_out

**(**where=**(**\_type\_=**1));**

by \_type\_**;**

Nobs=\_freq\_**;**

score\_decile = score\_decile+**1;**

if first.\_type\_ then do**;**

cum\_obs=Nobs**;**

model\_pred=Y\_Sum**;**

end**;**

else do**;** cum\_obs=cum\_obs+Nobs**;**

model\_pred=model\_pred+Y\_Sum**;**

end**;**

retain cum\_obs model\_pred**;**

\* 201 represents the number of successes;

\* This value will need to be changed with different samples;

pred\_rate=model\_pred/**201;**

base\_rate=score\_decile\***0.1;**

lift = pred\_rate-base\_rate**;**

drop \_freq\_ \_type\_ **;**

**run;**

**proc** **print** **data**=lift\_chart**;** **run;**

ods graphics on**;**

axis1 label=**(**angle=**90** '% Captured from Target Population'**);** axis2 label=**(**'Total Population'**);**

legend1 label=**(**color=black height=**1** ''**)** value=**(**color=black height=**1** 'Model #2' 'Random Guess'**);**

title 'Model #2: In-Sample Lift Chart'**;**

symbol1 color=green interpol=join w=**2** value=dot height=**1;**

symbol2 color=black interpol=join w=**2** value=dot height=**1;**

**proc** **gplot** **data**=lift\_chart**;** plot pred\_rate\*base\_rate base\_rate\*base\_rate / overlay legend=legend1

vaxis=axis1 haxis=axis2**;** **run;** **quit;**

ods graphics off**;**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Create a Lift Chart for Model #1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

title 'Model #1: Create In-Sample Lift Chart'**;**

\* The descending option assigns the highest model scores to the lowest score\_decile;

**proc** **rank** **data**=model\_data out=training\_scores descending groups=**10;**

var yhat**;** ranks score\_decile**;** where train=**1;** **run;**

\* To create the lift chart run this exact code;

**proc** **means** **data**=training\_scores sum**;**

class score\_decile**;** var Y**;**

output out=pm\_out sum**(**Y**)**=Y\_Sum**;** **run;**

**proc** **print** **data**=pm\_out**;** **run;**

**data** lift\_chart**;** set pm\_out

**(**where=**(**\_type\_=**1));**

by \_type\_**;**

Nobs=\_freq\_**;**

score\_decile = score\_decile+**1;**

if first.\_type\_ then do**;**

cum\_obs=Nobs**;**

model\_pred=Y\_Sum**;**

end**;**

else do**;** cum\_obs=cum\_obs+Nobs**;**

model\_pred=model\_pred+Y\_Sum**;**

end**;**

retain cum\_obs model\_pred**;**

\* 201 represents the number of successes;

\* This value will need to be changed with different samples;

pred\_rate=model\_pred/**201;**

base\_rate=score\_decile\***0.1;**

lift = pred\_rate-base\_rate**;**

drop \_freq\_ \_type\_ **;**

**run;**

**proc** **print** **data**=lift\_chart**;** **run;**

ods graphics on**;**

axis1 label=**(**angle=**90** '% Captured from Target Population'**);** axis2 label=**(**'Total Population'**);**

legend1 label=**(**color=black height=**1** ''**)** value=**(**color=black height=**1** 'Model #1' 'Random Guess'**);**

title 'Model #1: In-Sample Lift Chart'**;**

symbol1 color=green interpol=join w=**2** value=dot height=**1;**

symbol2 color=black interpol=join w=**2** value=dot height=**1;**

**proc** **gplot** **data**=lift\_chart**;** plot pred\_rate\*base\_rate base\_rate\*base\_rate / overlay legend=legend1

vaxis=axis1 haxis=axis2**;** **run;** **quit;**

ods graphics off**;**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Out-of-sample testing for Model #1 - Lift Chart

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

title 'Model #1: Out-of-Sample Lift Chart'**;**

\* confirm the quantity of observations in the training and testing samples;

**proc** **freq** **data**=temp**;** tables train\*Y**;** **run;**

/\* The Model #1 does not need to be refit using the test data given that when using the

variable 'Y\_train' as the response variable SAS will automatically fit the model to the

testing data set and score the data set with the out-of-sample predicted values

(yhat where (train=0).

\*/

\* The descending option assigns the highest model scores to the lowest score\_decile;

**proc** **rank** **data**=model\_data out=testing\_scores descending groups=**10;**

var yhat**;**

ranks score\_decile**;**

where train=**0;** \* train=0 is to rank the testing data set

run;

\* To create the lift chart run this exact code;

**proc** **means** **data**=testing\_scores sum**;**

class score\_decile**;** var Y**;**

output out=pm\_out sum**(**Y**)**=Y\_Sum**;** **run;**

**proc** **print** **data**=pm\_out**;** **run;**

**data** lift\_chart**;** set pm\_out

**(**where=**(**\_type\_=**1));**

by \_type\_**;**

Nobs=\_freq\_**;**

score\_decile = score\_decile+**1;**

if first.\_type\_ then do**;**

cum\_obs=Nobs**;**

model\_pred=Y\_Sum**;**

end**;**

else do**;** cum\_obs=cum\_obs+Nobs**;**

model\_pred=model\_pred+Y\_Sum**;**

end**;**

retain cum\_obs model\_pred**;**

\* 95 represents the number of successes - this scales the lift chart;

pred\_rate=model\_pred/**95;**

base\_rate=score\_decile\***0.1;**

lift = pred\_rate-base\_rate**;**

drop \_freq\_ \_type\_ **;**

**run;**

**proc** **print** **data**=lift\_chart**;** **run;**

ods graphics on**;**

axis1 label=**(**angle=**90** '% Captured from Target Population'**);** axis2 label=**(**'Total Population'**);**

legend1 label=**(**color=black height=**1** ''**)** value=**(**color=black height=**1** 'Model #1' 'Random Guess'**);**

title 'Model #1: Out-of-Sample Lift Chart'**;**

symbol1 color=green interpol=join w=**2** value=dot height=**1;**

symbol2 color=black interpol=join w=**2** value=dot height=**1;**

**proc** **gplot** **data**=lift\_chart**;** plot pred\_rate\*base\_rate base\_rate\*base\_rate / overlay legend=legend1

vaxis=axis1 haxis=axis2**;** **run;** **quit;**

ods graphics off**;**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Out-of-sample testing for Model #2 - Lift Chart

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

title 'Model #2: Out-of-Sample Lift Chart'**;**

/\* Model #2 does not need to be refit using the test data given that when using the

variable 'Y\_train' as the response variable SAS will automatically fit the model to the

testing data set and score the data set with the out-of-sample predicted values

(yhat where (train=0).

\*/

\* The descending option assigns the highest model scores to the lowest score\_decile;

**proc** **rank** **data**=model\_data2 out=testing\_scores descending groups=**10;**

var yhat**;**

ranks score\_decile**;**

where train=**0;** \* train=0 is to rank the testing data set

run;

\* To create the lift chart run this exact code;

**proc** **means** **data**=testing\_scores sum**;**

class score\_decile**;** var Y**;**

output out=pm\_out sum**(**Y**)**=Y\_Sum**;** **run;**

**proc** **print** **data**=pm\_out**;** **run;**

**data** lift\_chart**;** set pm\_out

**(**where=**(**\_type\_=**1));**

by \_type\_**;**

Nobs=\_freq\_**;**

score\_decile = score\_decile+**1;**

if first.\_type\_ then do**;**

cum\_obs=Nobs**;**

model\_pred=Y\_Sum**;**

end**;**

else do**;** cum\_obs=cum\_obs+Nobs**;**

model\_pred=model\_pred+Y\_Sum**;**

end**;**

retain cum\_obs model\_pred**;**

\* 95 represents the number of successes - this scales the lift chart;

pred\_rate=model\_pred/**95;**

base\_rate=score\_decile\***0.1;**

lift = pred\_rate-base\_rate**;**

drop \_freq\_ \_type\_ **;**

**run;**

**proc** **print** **data**=lift\_chart**;** **run;**

ods graphics on**;**

axis1 label=**(**angle=**90** '% Captured from Target Population'**);** axis2 label=**(**'Total Population'**);**

legend1 label=**(**color=black height=**1** ''**)** value=**(**color=black height=**1** 'Model #2' 'Random Guess'**);**

title 'Model #2: Out-of-Sample Lift Chart'**;**

symbol1 color=green interpol=join w=**2** value=dot height=**1;**

symbol2 color=black interpol=join w=**2** value=dot height=**1;**

**proc** **gplot** **data**=lift\_chart**;** plot pred\_rate\*base\_rate base\_rate\*base\_rate / overlay legend=legend1

vaxis=axis1 haxis=axis2**;** **run;** **quit;**

ods graphics off**;**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* END

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**References:**

Microsoft. (n.d.). *Lift chart (analysis services - data mining)*. Retrieved from <http://msdn.microsoft.com/en-us/library/ms175428.aspx>

Ratner, B. (2012). *Statistical and machine-learning data mining*. (2nd ed., p. 415). Boca Raton: CRC Press.

Chatterjee, S., & Hadi, A. (2012). *Regression analysis by example*. (5th ed., p. 308). Hoboken: John Wiley & Sons.

Allison, P. (2012). *Logistic regression using sas: theory and application*. (2nd ed., p. 67). Cary: SAS Publishing.

Hosmer, D., Lemeshow, S., & Sturdivant, R. (2013). *Applied logistic regression*. (3rd ed., p. 154). Hoboken: John Wiley & Sons.