Assignment #6

Introduction:

In this exercise, we will fit two multiple logistic regression models to a binary response variable for credit card application approval in the credit_approval data set. The first model will be fit using the backward selection automated variable selection technique. This model will then be compared to a pre-determined model that includes the A9, A2 and A3 variables.

To assess the predictive accuracy of both models, the data will be split into training and test sets to perform cross-validation. The models will be fitted on the training data, and evaluated on in-sample and out-of sample goodness-of-fit statistics and predictive accuracy. Predictive accuracy will be assessed by creating lift charts for both in-sample and out-of-sample data and comparing the results.

Results:

Before fitting any regression models, the credit_approval data was split into a training and test set. Both sets were created through a random selection of observations, with 70 percent of the observations assigned to the training data and 30 percent of the observations assigned to the test data set. The models discussed below were fit on the training data. In-sample and out-of-sample results follow.

In-Sample Results:

Model #1

Model #1 was fit using the backward elimination automated variable selection technique. Each of the nine categorical variables (A1, A4, A5, A6, A7, A9, A10, A12 and A13) was transformed into a set of dummy/design variables prior to performing the backward selection. All six of the continuous variables (A2, A3, A8, A11, A14 and A15) in the data set were also included in the procedure for consideration.

The selection procedure was completed in 22 steps. The summary table for the backward elimination procedure is included below.

Summary of Backward Elimination					
Step	Effect Removed	DF	Number In	Wald Chi-Square	Pr > ChiSq
1	A6_aa	1	25	0.0000	0.9977
2	A6_m	1	24	0.0008	0.9777
3	A3	1	23	0.0571	0.8111
4	A6_ff	1	22	0.1409	0.7074
5	A6_k	1	21	0.2492	0.6177
6	A6_q	1	20	0.4946	0.4819
7	A6_c	1	19	0.2848	0.5935
8	A13_g	1	18	0.5308	0.4663
9	A10_t	1	17	0.6713	0.4126
10	A2	1	16	0.7937	0.3730
11	A1_b	1	15	0.9181	0.3380
12	A7_bb	1	14	0.9882	0.3202
13	A7_h	1	13	0.5178	0.4718
14	A12_t	1	12	1.0196	0.3126
15	A7_v	1	11	1.2996	0.2543
16	A14	1	10	1.7137	0.1905
17	A6_i	1	9	1.7887	0.1811
18	A8	1	8	2.2819	0.1309
19	A6_w	1	7	2.6159	0.1058
20	A6_cc	1	6	3.0036	0.0831
21	A4_u	1	5	3.2175	0.0729
22	A6_x	1	4	3.8322	0.0503

The resulting multiple logistic regression model includes four predictor variables, A11, A15, A7_ff and A9_t. The maximum likelihood estimates of the coefficients and intercept of the model are included below:

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-3.0087	0.3228	86.8612	<.0001
A11	1	0.2338	0.0608	14.7699	0.0001
A15	1	0.000561	0.000206	7.3932	0.0065
A7_ff	1	-2.2218	0.8556	6.7427	0.0094
A9_t	1	3.5735	0.3587	99.2458	<.0001

The output shows that each of the four predictor variables is significant at or beyond the $p<0.01$ level using the Wald Chi-Square test. Both A11 and A15 are continuous, and A9_t is a design

variable for the A9 categorical variable that takes the values ‘t’ or ‘f’. A7 is a categorical variable with nine categories, one of which is “ff.” The other categories for A7 considered in the backward elimination procedure were dropped in steps 12, 13 and 15, indicating that these categories were not seen as statistically significantly different from the categories that make up the base of the set of A7 dummy variables. In other words, through selecting A7_ff in this particular model the backward elimination procedure is indicating that this particular category of the A7 variable was the only one seen as a significant predictor of the response variable.

The odds ratio estimates for a one-unit increase in each predictor variable in Model #1 are:

Odds Ratio Estimates			
Effect	**Point Estimate**	**95% Wald Confidence Limits**	
A11	1.263	1.121	1.423
A15	1.001	1.000	1.001
A7_ff	0.108	0.020	0.580
A9_t	35.640	17.645	71.989

These values are difficult to compare given the scale of each variable. Through previous exploratory data analysis, we know that the 5th to 95th percentile values for the continuous variables range from 0 to 14 for A11 to 0 to 8000 for A15. The odds ratio for A11 indicates that the response variable is 1.263 times more likely to equal 1 for every one-unit increase in A11. The odds ratio for A15 indicates that the odds of Y=1 are 1.001 higher for every one-unit increase in A15. Despite a lower odds ratio, it can’t be said automatically that A15 is less of an influence on the value of Y. Given its large range of values, it is plausible that changes in A15 could and would come in unit increases many times greater than one.

The odds ratios of the design variables are interpreted in a similar fashion. When an observation falls into the A7 ‘ff’ category, the odds of Y=1 are multiplied by 0.108 (which is actually a reduction in the odds of Y=1). When A9 is equal to ‘t’ rather than ‘f’, the odds of Y=1 are 35.64 times greater.

The goodness of fit of Model #1 is measured in the following table:

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	89.1	**Somers' D**	0.787
Percent Discordant	10.5	**Gamma**	0.790
Percent Tied	0.4	**Tau-a**	0.390
Pairs	50049	**c**	0.893

Each of these measures is based on the pairwise comparisons of observations with Y=0 and observations with Y=1. For each pair, we compare the predicted (Y-hat) values of Y=1 and Y=0. If the Y=1 half of the pair has a higher score (Y-hat value), the pair is listed as “concordant;” if

not, the value is “discordant” or “tied” (for equal scores). Each of the measures on the right side of the table are based off of the number of concordant (C), discordant (D), tied (T) values or number of observations (N). The relationships are:

Tau-a = (C-D) / N
Gamma = (C-D) / (C+D)
Somer’s D = (C-D) / (C+D+T)
c = 0.5*(1-Somer’s D)

Each of the measures takes a value between 0 and 1, with a value nearer to 1 indicating a better goodness-of-fit. The statistic c also corresponds to the area under the model’s ROC curve, which can be used to assess predictive strength of the model. All of the four measures indicate a strong fit for the model. This is backed up by additional model fit statistics and global tests of significance for the model.

Model Fit Statistics		
Criterion	**Intercept Only**	**Intercept and Covariates**
AIC	620.703	291.046
SC	624.812	311.592
-2 Log L	618.703	281.046

Testing Global Null Hypothesis: BETA=0			
Test	**Chi-Square**	**DF**	**Pr > ChiSq**
Likelihood Ratio	337.6572	4	<.0001
Score	265.6694	4	<.0001
Wald	132.2097	4	<.0001

The statistics show a clearly significant model. The intercept and covariates values for the AIC, SC and deviance (-2LogL) decrease dramatically from the intercept-only values, indicating a stronger and better fitting model. Additionally, the Likelihood Ratio, Score and Wald tests for the null hypothesis that none of the coefficients are significant are all statistically significant at the $p<0.0001$ level, strongly indicating the model has statistically significant explanatory power.

In addition to the goodness-of-fit statistics, a lift chart table and lift chart plot were also generated for each model on both the training and test data. The lift chart is helpful in understanding the predictive power of a model (and how to use it). To build the chart, the data is sorted so that the observations are listed from highest predicted probability of an event (Y=1) to lowest. This can be done with or without a “target value,” which means that the chart can be generated comparing predictions for Y=1 *and* Y=0 (without a target value) or just one or the other (with a target value). In this case, we are interested in determining how well our models can predict credit card application approval (coded as Y=1), so a target value will be used.

After the observations were sorted, they were combined into ten groups (score deciles). The first decile is made up of the 10 percent of observations determined to have the highest probability of Y=1. The next decile is made up of the next 10 percent of observations rated on probability of Y=1, and continuing until all observations have been assigned to a decile. We then compare the actual number of observations where Y=1 with the predicted number of observations where Y=1. The lift chart table from the training data for Model #1 shows more below.

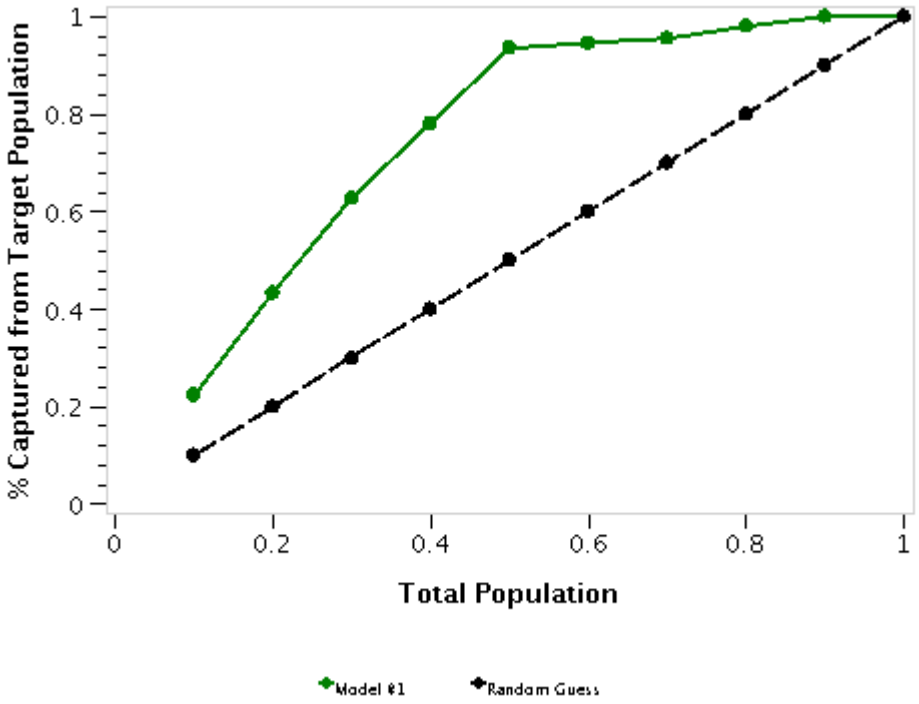
Obs	score_decile	Y_Sum	Nobs	cum_obs	model_pred	pred_rate	base_rate	lift
1	1	45	45	45	45	0.22388	0.1	0.12388
2	2	42	45	90	87	0.43284	0.2	0.23284
3	3	39	45	135	126	0.62687	0.3	0.32687
4	4	31	43	178	157	0.78109	0.4	0.38109
5	5	31	55	233	188	0.93532	0.5	0.43532
6	6	2	37	270	190	0.94527	0.6	0.34527
7	7	2	45	315	192	0.95522	0.7	0.25522
8	8	5	34	349	197	0.98010	0.8	0.18010
9	9	4	74	423	201	1.00000	0.9	0.10000
10	10	0	27	450	201	1.00000	1.0	0.00000

The deciles are labeled 1 through 10. Y_Sum is equal to the number of *actual* observations of Y=1. The Nobs column represents the number of *predicted* observations of Y=1 for each decile. It can easily be seen that the best predictions are made in the top level deciles due to sorting the observations based on probability of Y=1. The cum_obs column lists the cumulative number of predicted Y=1 observations as we progress down through the deciles and model_pred does the same for the number of actual Y=1 observations.

The heart of lift chart table is found in the three rightmost columns. Pred_rate is a cumulative measure that tracks the percent of actual observations correctly predicted. For example, the first pred_rate of 0.22388 indicates that 22.388 percent of the actual observations of Y=1 were predicted in the first decile of observations. Looking further down the table in row 5, we can see that the first 50 percent of the observations (deciles 1 through 5) when sorted for probability predict more than 93 percent of the actual observations of Y=1. This has significant practical applications. Marketers for example use lift charts to target audiences most likely to respond (saving money by sending their message to those most likely to respond).

The base_rate in the lift chart corresponds to the cumulative percentage of Y=1 observations we would expect to have predicted correctly through random guessing. With a target value, this amounts to guessing each Y=1 for each observation. After going through 100 percent of the sample, we would expect to have correctly guessed 100 percent of the actual Y=1 observations (although at a poor rate). The “lift” score, also called the Kolmogorov-Smirnov test statistic, indicates how much better the model classification rate (pred_rate) is than random guessing (base_rate).

Lift peaks at the inflection point before diminishing marginal returns of classification set in. In other words, lift is highest when many observations have been correctly classified yet relatively few have been examined. The lift values for the deciles in Model #1 are highest at decile 5, where 50 percent of the observations have been examined yet more than 93 percent of actual observations have been found. A visual presentation of the lift chart table can be found in the lift chart plot below.



The cumulative percentage of actual observations found (pred_rate) is on the Y-axis of the plot and the percentage of observations examined (base_rate) is on the X-axis. The number of actual observations found at any point of examination can be found by drawing a vertical line up from the X-axis to the curve in green and then over the Y-axis. These X and Y values represent the base_rate and pred_rate at the level chosen (for example, 0.5 on the X-axis corresponds with about .93 on the Y-axis, as we saw previously in the table).

The straight-dotted line on the graph indicates the results of random guessing (1*base_rate). It is the baseline that is used to evaluate model fit. A perfect model would start at 0 in the lower left-hand corner of the graph and rise sharply with each additional observation until it hits the upper-bound of 1.00, where it would switch to a straight line parallel to the X-axis continuing across the top of the graph. The intuitive explanation for this line is that each observation of Y would yield a correct prediction (moving up the Y-axis) until all actual observations of Y=1 had been correctly predicted without any errors (hitting the upper-bound of 1). The graph would then continue across where Y=1.00 due to all actual observations having been already found. The closer the lift chart plot is to this perfect model, the better the model.

MODEL #2

A second model, Model #2, was fit using the predictor variables A9, A2 and A3. This model was pre-determined because of its particular interest to management. The goal of the exercise is to compare the automated selection Model #1 to Model #2 to determine which model has the greater predictive power. The coefficient estimates and statistical significance tests for each predictor variable for Model #2 are displayed below.

Analysis of Maximum Likelihood Estimates					
Parameter	**DF**	**Estimate**	**Standard Error**	**Wald Chi-Square**	**Pr > ChiSq**
Intercept	1	-3.6287	0.5051	51.6051	<.0001
A9_t	1	3.9836	0.3302	145.5842	<.0001
A2	1	0.0227	0.0127	3.1641	0.0753
A3	1	0.0527	0.0314	2.8241	0.0929

The A9 variable, here present once again in the design variable form A9_t, is statistically significant at the p<0.0001 level. The continuous A2 and A3 variables are both significant at the p<0.10 level but are *not* significant at the p<0.05 level. This explains why both variables were dropped by the backward elimination procedure, which used p<0.05 as the cutoff level for inclusion in the final model. The variables were removed in steps 2 and 16 of the backward elimination method.

The odds ratios for the three predictor variables are:

Odds Ratio Estimates			
Effect	**Point Estimate**	**95% Wald Confidence Limits**	
A9_t	53.712	28.122	102.590
A2	1.023	0.998	1.049
A3	1.054	0.991	1.121

Logistic regression does not have an error term like ordinary least squares regression, but instead shifts the Y=1 probability-influencing effects of the model between the variables included. This explains the change in the odds ratio for the A9_t variable, which is now listed as 53.712 times greater odds for Y=1 when A9 is ‘t’ (compared to 35.640 in Model #1).

The 5th to 95th percentile values for the continuous variables are 18.83 to 58.42 for A2 and 0.17 to 15.00 for A3 when Y=1. Once again, the odds ratios give the increase in odds for Y=1 for each one-unit increase in the predictor variable. Interestingly, the 95% Wald Confidence Limits for the two continuous variables gives a range that includes odds ratios that improve (values greater than 1) or worsen (values less than 1) the odds of Y=1, meaning that the relationship between these two variables and the outcome of Y is less certain.

The concordant and discordant measures of association are close, but inferior to the values for the same measurements in Model #1. The percent concordant is 89.1 in Model #2 compared to

91.6 in Model #1. The percent discordant is nearly double (10.5 to 5.4) in Model #2, which

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	89.1	**Somers' D**	0.787
Percent Discordant	10.5	**Gamma**	0.790
Percent Tied	0.4	**Tau-a**	0.390
Pairs	50049	**c**	0.893

explains the lower Somer's D, Gamma, Tau-a and c values for Model #2, which are overall still relatively high values.

Model Fit Statistics		
Criterion	**Intercept Only**	**Intercept and Covariates**
AIC	620.703	340.739
SC	624.812	357.176
-2 Log L	618.703	332.739

Testing Global Null Hypothesis: BETA=0			
Test	**Chi-Square**	**DF**	**Pr > ChiSq**
Likelihood Ratio	285.9640	3	<.0001
Score	246.5494	3	<.0001
Wald	151.7473	3	<.0001

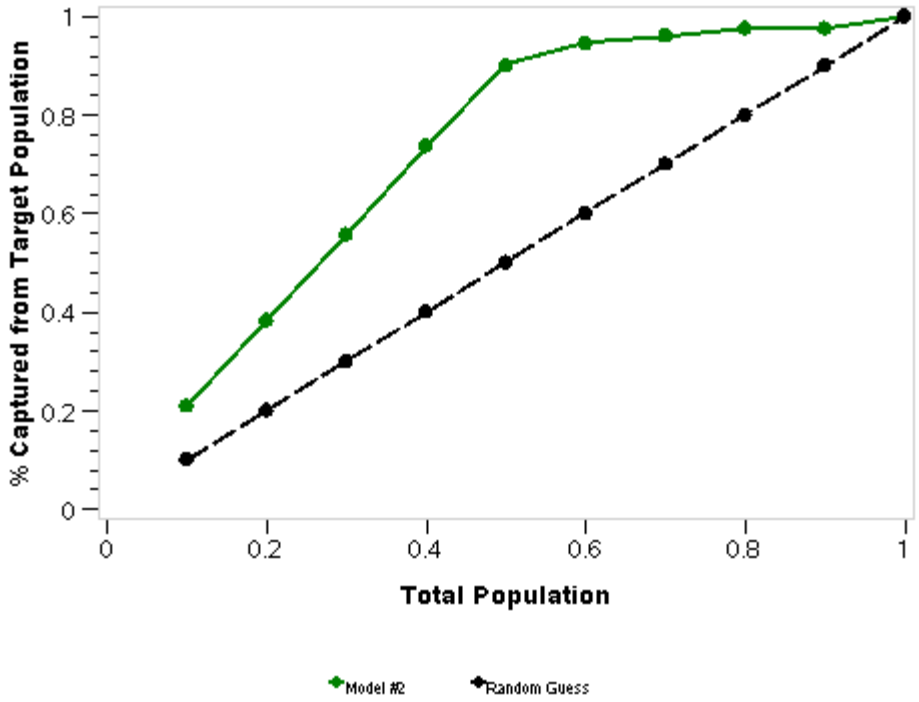
AIC, SC, -2LogL and the global tests for significance all also suggest that Model #2 has significant explanatory power. Additionally, these measures can be used to compare the two models based on goodness-of-fit. Lower AIC, SC, and -2LogL values for Model #1 indicate Model #1 has a better fit than Model #2. Lastly, all three Chi-Square significant tests are significant at the $p<0.0001$ level in Model #2.

The lift chart for Model #2 show fairly good lift values, although once again, Model #1 appears

Obs	score_decile	Y_Sum	Nobs	cum_obs	model_pred	pred_rate	base_rate	lift
1	1	42	45	45	42	0.20896	0.1	0.10896
2	2	35	45	90	77	0.38308	0.2	0.18308
3	3	35	45	135	112	0.55721	0.3	0.25721
4	4	36	45	180	148	0.73632	0.4	0.33632
5	5	33	45	225	181	0.90050	0.5	0.40050
6	6	9	45	270	190	0.94527	0.6	0.34527
7	7	3	45	315	193	0.96020	0.7	0.26020

Obs	score_decile	Y_Sum	Nobs	cum_obs	model_pred	pred_rate	base_rate	lift
8	8	3	45	360	196	0.97512	0.8	0.17512
9	9	0	45	405	196	0.97512	0.9	0.07512
10	10	5	45	450	201	1.00000	1.0	0.00000

to do a better job. The lift values for each of the first 5 deciles are higher in Model #1 than in Model #2. Model #1 also has a higher peak lift value (0.43532 vs. 0.40050). The lift values after the fifth decile are higher in Model #2 than in Model #1, but that is due to the lift curve leveling off faster in Model #1 as all or nearly all Y=1 values are already predicted by higher probability observations of Y.



All else considered, both models appear to be close in fit and predictive accuracy, with Model #1 holding an advantage. To determine which model is to be preferred for prediction, however, both models were analyzed on out-of-sample test data to see which yielded the best predictions.

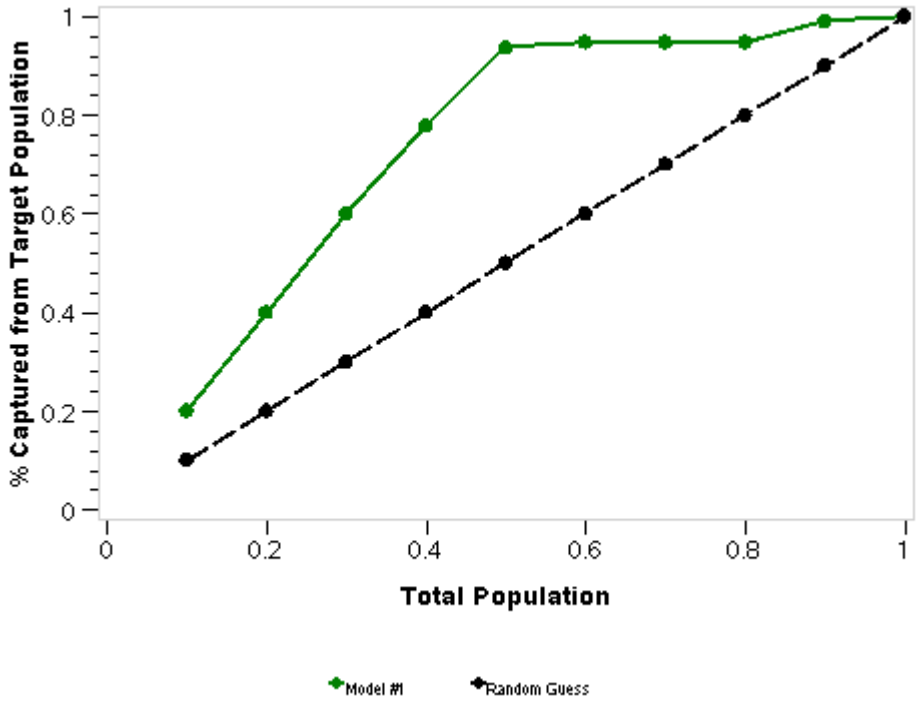
Out-of-Sample Results:

MODEL #1

Lift charts were created for both models using out-of-sample data. The out-of-sample or “test” data was originally split off of the original data set. It represents the 30 percent of original observations not used to train either of the models.

The lift chart for Model #1 is below:

Obs	score_decile	Y_Sum	Nobs	cum_obs	model_pred	pred_rate	base_rate	lift
1	1	19	20	20	19	0.20000	0.1	0.10000
2	2	19	20	40	38	0.40000	0.2	0.20000
3	3	19	21	61	57	0.60000	0.3	0.30000
4	4	17	20	81	74	0.77895	0.4	0.37895
5	5	15	29	110	89	0.93684	0.5	0.43684
6	6	1	12	122	90	0.94737	0.6	0.34737
7	7	0	20	142	90	0.94737	0.7	0.24737
8	8	0	7	149	90	0.94737	0.8	0.14737
9	9	4	42	191	94	0.98947	0.9	0.08947
10	10	1	12	203	95	1.00000	1.0	0.00000



The Model #1 out-of-sample lift chart shows that the model continues to predict Y at nearly the same rate as it did on the in-sample data. Here, the peak lift value is almost identical (0.43684 out-of-sample vs. 0.43532 in-sample). Once again, the predictions for Y in the top five deciles is very strong, with more than 93 percent of all “Y=1” events predicted among the top half of observed values of Y, sorted by probability of an event. It takes longer to reach 100 percent of events in the out-of-sample data. The in-sample chart showed that all events had been predicted by the ninth decile, but the out-of-sample data shows that all events are predicted in decile 10.

The prediction rates for the out-of-sample data are also quite good. The out-of-sample lift chart shows lower pred_rate values for the first few deciles, but by decile 4, the model predicted

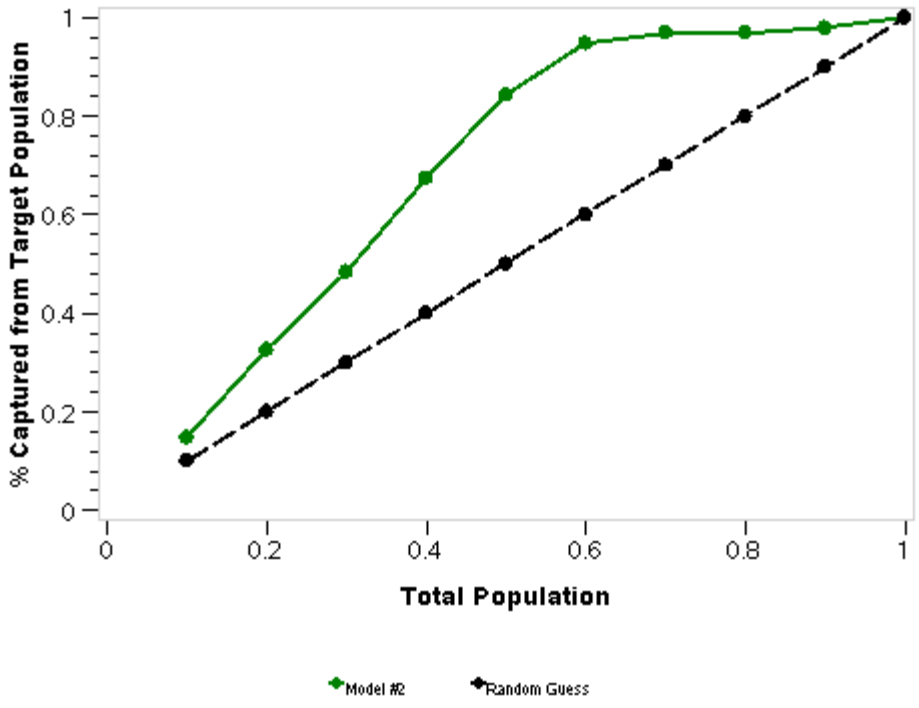
about as many events as it did on the in-sample data.

These are impressive results for the out-of-sample predictive accuracy of Model #1. Despite the fact that none of the observations of Y in the out-of-sample data were used to train the model, the model retains much of its predictive accuracy. This indicates that the model would do well in predicting future values of Y.

MODEL #2

The lift chart for out-of-sample data for Model #2 is below:

Obs	score_decile	Y_Sum	Nobs	cum_obs	model_pred	pred_rate	base_rate	lift
1	1	14	20	20	14	0.14737	0.1	0.04737
2	2	17	20	40	31	0.32632	0.2	0.12632
3	3	15	21	61	46	0.48421	0.3	0.18421
4	4	18	20	81	64	0.67368	0.4	0.27368
5	5	16	20	101	80	0.84211	0.5	0.34211
6	6	10	21	122	90	0.94737	0.6	0.34737
7	7	2	20	142	92	0.96842	0.7	0.26842
8	8	0	21	163	92	0.96842	0.8	0.16842
9	9	1	20	183	93	0.97895	0.9	0.07895
10	10	2	20	203	95	1.00000	1.0	0.00000



Model #2's predictive accuracy erodes to a much greater degree on the out-of-sample data than

Model #1's predictive accuracy did. Here we see that only about 84 percent of events are predicted in the top 5 deciles (compared to 90 percent using in-sample data). The lift scores for each of the top 5 deciles are lower on the out-of-sample lift chart for Model #2, indicating a poorer performance of predicting events using the highest probability predictions for Y=1. Peak lift is also down on the out-of-sample lift chart for Model #2 (0.34737 vs. 0.40050).

The prediction rate scores for Model #2 on out-of-sample data are also much lower overall. The top decile, which is made up of the top 10 percent of observations that the model assigns the highest probability of finding Y=1, predicts only about 14 percent of the overall events (Y=1 observations). That is compared with a prediction rate of nearly 21 percent on the in-sample data. The result is a much poorer lift score for decile 1 (0.04737 vs. 0.10896). Similar poor results can be found in each of the other deciles.

COMPARISON

When comparing Model #1 and Model #2, we see that Model #1's predictive accuracy, judged by lift scores, is higher for both in-sample and out-of-sample data. Additionally, Model #1 retains a higher degree of its predictive accuracy on out-of-sample data than does Model #2. This is an important piece of information to consider when evaluating the two models. The model that does better on out-of-sample data can be trusted to do a better job of predicting future observations of Y, which will also be out-of-sample by definition.

Model #1 has a higher peak lift in all cases than Model #2. It also predicts all events (Y=1) higher up on the lift chart than Model #2. This last point is important because of the way lift charts are typically used. Sometimes, a company will want to predict a certain number of events, and they will use a lift chart to determine which observations in their data give them the best chance of meeting their prediction goal while using the fewest amount of resources.

For example, in credit card application approval rating, if the company wants to identify the 100 credit card applications most likely to be improved. A model that has good ratings toward the top of the lift chart will help the credit card company identify the 100 credit card applications that will be approved while looking at the fewest number of applications. They don't want to have to sort through every single application to find potentially strong ones. The lift chart helps them narrow down their choice.

In this case, Model #1 does this job better than Model #2 for all of the reason suggested.

Conclusions:

Two multiple logistic regression models were fit to a binary response variable for credit card application approval. The model chosen through automated variable selection (Model #1) was found to have a better fit on in-sample data than Model #2, which was predetermined to include three predictor variables (A9, A2, A3).

Lift charts were used to compare both models on in-sample and out-of-sample data. Model #1

did a superior job of predicting Y than Model #2 on both in-sample and out-of-sample data. Model #1 also retained its predictive accuracy better on out-of-sample data than did Model #2.

For these reasons, it was determined that Model #1 is the superior model and should be preferred over Model #2.

Code:

```
/*****************************************/
/*PREDICT 401                            **/
/*Assignment #6                          **/
/*****************************************/

/*****************************************/
/** Data Information                      **/
/*****************************************/

libname mydata
'/courses/u_northwestern.edu1/i_833463/c_3505/SAS_Data/'
access=readonly;

title "Data Information";
proc contents data=mydata.credit_approval; run; quit;
proc print data=mydata.credit_approval(obs=5); run; quit;

* Turn off ods graphics;
ods graphics off;

/*****************************************/
/** Data Formatting                       **/
/*****************************************/

title;

data temp;
     set mydata.credit_approval;

     *Split data into training/test sets;
     u=uniform(123);
     if (u<0.7) then train=1; else train=0;

     *Create response variable Y from 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=.;

     /****************************************/

     /*Define dummy variables                        */

     * A1: Base category is: A1='a';
     if (A1='b') then A1_b=1; else A1_b=0;

     * A4: Base category is: l,y;
     if (A4='u') then A4_u=1; else A4_u=0;

     * A5: Base category is: gg,p;
     if (A5='g') then A5_g=1; else A5_g=0;

     * A6: Base category is: d,e,j,r;
     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='ff') then A6_ff=1; else A6_ff=0;
     if (A6='i')  then A6_i=1; else A6_i=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;

     * A7: Base category is: dd,j,n,o,z;
     if (A7='bb') then A7_bb=1; else A7_bb=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='v')  then A7_v=1; else A7_v=0;

     *A9-A12: Base category is: f;
     if (A9='t')  then A9_t=1; else A9_t=0;
     if (A10='t') then A10_t=1; else A10_t=0;
     if (A12='t') then A12_t=1; else A12_t=0;

     * A13: Base category is: p,s;
     if (A13='g') then A13_g=1; else A13_g=0;

     /****************************************/

     *Delete observations with missing values;
     if (A1='?') or (A4='?') or (A5='?') or (A6='?') or (A7='?')
     or (A2=.) or (A3=.) or (A8=.) or (A11=.) or (A14=.) or
(A15=.)
```

```
     then delete;

run;

/*****************************************/
/** Model Fitting                       **/
/*****************************************/

*Fit Model #1 with automated variable selection;
title "Model #1";
proc logistic data=temp descending;
     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_q A6_x A7_bb
          A7_ff A7_h A7_v A9_t A10_t
          A12_t A13_g / selection=backward;
          output out=model_data pred=yhat;
run;

*Fit Model #2;
title "Model #2";
proc logistic data=temp descending;
     model Y_train = A9_t A2 A3;
     output out=model_data2 pred=yhat;
run;

/*****************************************/
/** Assessing Predictive Accuracy       **/
/*****************************************/

title;

/*****************************************/
/** Model #1: In-Sample Lift Chart      **/
/*****************************************/

*Rank model scores;
proc rank data=model_data out=training_scores descending
groups=10;
     var yhat;
     ranks score_decile;
     where train=1;
run;

*Identify scaling factor;
title "Scale Factor for Lift Charts";
proc freq data=temp;
     tables train*Y;
```

```
run;

*Create lift chart;
title "Model #1: In-Sample Lift Chart Tables";
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;

     pred_rate=model_pred/201; *201 is scaling factor for in-
sample lift chart;
     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;

/*****************************************/



/*****************************************/
/** Model #2: In-Sample Lift Chart      **/
/*****************************************/

*Rank model scores;
proc rank data=model_data2 out=training_scores descending
groups=10;
     var yhat;
     ranks score_decile;
     where train=1;
run;

*Create lift chart;
title "Model #2: In-Sample Lift Chart Tables";
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;
```

```
     pred_rate=model_pred/201; *201 is scaling factor for in-
sample lift chart;
     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;

/*****************************************/



/*****************************************/
/** Model #1: Out-of-Sample Lift Chart  **/
/*****************************************/

*Rank model scores;
proc rank data=model_data out=test_scores descending groups=10;
     var yhat;
     ranks score_decile;
     where train=0;
run;

*Create lift chart;
title "Model #1: Out-of-Sample Lift Chart Tables";
proc means data=test_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;

     pred_rate=model_pred/95; *95 is scaling factor for out-of-
sample lift chart;
     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;

/*****************************************/

/*****************************************/
```

```
/** Model #2: Out-of-Sample Lift Chart  **/
/*****************************************/

*Rank model scores;
proc rank data=model_data2 out=test_scores descending groups=10;
     var yhat;
     ranks score_decile;
     where train=0;
run;

*Create lift chart;
title "Model #2: Out-of-Sample Lift Chart Tables";
proc means data=test_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;

     pred_rate=model_pred/95; *95 is scaling factor for out-of-
sample lift chart;
     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 CODE                            **/
/*****************************************/
```