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.

Obs	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	u	train	Y	Y_train
1	b	30.830	0.000	u	g	w	v	1.250	t	t	1.000	f	g	202.000	0.000	+	0.75040	0	1	.
2	a	58.670	4.460	u	g	q	h	3.040	t	t	6.000	f	g	43.000	560.000	+	0.32091	1	1	1
3	a	24.500	0.500	u	g	q	h	1.500	t	f	0.000	f	g	280.000	824.000	+	0.17839	1	1	1
4	b	27.830	1.540	u	g	w	v	3.750	t	t	5.000	t	g	100.000	3.000	+	0.90603	0	1	.
5	b	20.170	5.625	u	g	w	v	1.710	t	f	0.000	f	s	120.000	0.000	+	0.35712	1	1	1
6	b	32.080	4.000	u	g	m	v	2.500	t	f	0.000	t	g	360.000	0.000	+	0.22111	1	1	1
7	b	33.170	1.040	u	g	r	h	6.500	t	f	0.000	t	g	164.000	31285.00	+	0.78644	0	1	.
8	a	22.920	11.585	u	g	cc	v	0.040	t	f	0.000	f	g	80.000	1349.000	+	0.39808	1	1	1
9	b	54.420	0.500	y	p	k	h	3.960	t	f	0.000	f	g	180.000	314.000	+	0.12467	1	1	1
10	b	42.500	4.915	y	p	w	v	3.165	t	f	0.000	t	g	52.000	1442.000	+	0.18769	1	1	1

Figure 1 - 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).

A1	Frequency	Percent	Cumulative Frequency	Cumulative Percent
?	12	1.74	12	1.74
a	210	30.43	222	32.17
b	468	67.83	690	100.00

A4	Frequency	Percent	Cumulative Frequency	Cumulative Percent
?	6	0.87	6	0.87
l	2	0.29	8	1.16
u	519	75.22	527	76.38
y	163	23.62	690	100.00

A5	Frequency	Percent	Cumulative Frequency	Cumulative Percent
?	6	0.87	6	0.87
g	519	75.22	525	76.09
gg	2	0.29	527	76.38
p	163	23.62	690	100.00

A6	Frequency	Percent	Cumulative Frequency	Cumulative Percent
?	9	1.30	9	1.30
aa	54	7.83	63	9.13
c	137	19.86	200	28.99
cc	41	5.94	241	34.93
d	30	4.35	271	39.28
e	25	3.62	296	42.90
ff	53	7.68	349	50.58
i	59	8.55	408	59.13
j	10	1.45	418	60.58
k	51	7.39	469	67.97
m	38	5.51	507	73.48
q	78	11.30	585	84.78
r	3	0.43	588	85.22
w	64	9.28	652	94.49
x	38	5.51	690	100.00

A7	Frequency	Percent	Cumulative Frequency	Cumulative Percent
?	9	1.30	9	1.30
bb	59	8.55	68	9.86
dd	6	0.87	74	10.72
ff	57	8.26	131	18.99
h	138	20.00	269	38.99
j	8	1.16	277	40.14
n	4	0.58	281	40.72
o	2	0.29	283	41.01
v	399	57.83	682	98.84
z	8	1.16	690	100.00

A9	Frequency	Percent	Cumulative Frequency	Cumulative Percent
f	329	47.68	329	47.68
t	361	52.32	690	100.00

A10	Frequency	Percent	Cumulative Frequency	Cumulative Percent
f	395	57.25	395	57.25
t	295	42.75	690	100.00

A12	Frequency	Percent	Cumulative Frequency	Cumulative Percent
f	374	54.20	374	54.20
t	316	45.80	690	100.00

A13	Frequency	Percent	Cumulative Frequency	Cumulative Percent
g	625	90.58	625	90.58
p	8	1.16	633	91.74
s	57	8.26	690	100.00

A16	Frequency	Percent	Cumulative Frequency	Cumulative Percent
+	307	44.49	307	44.49
-	383	55.51	690	100.00

Figure 2 - 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:

$$\Pi(x) = \frac{e^{g(x)}}{1+e^{g(x)}} = \frac{e^{(-3.0087 + 0.2338*A11 + 0.00056*A15 - 2.2218*A7_{ff} + 3.5735*A9_t)}}{1+e^{(-3.0087 + 0.2338*A11 + 0.00056*A15 - 2.218*A7\_ff + 3.5735*A9\_t)}}$$

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

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

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

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	365.5570	26	<.0001
Score	280.7880	26	<.0001
Wald	123.2131	26	<.0001

Figure 3 - 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.

Model Fit Statistics		
Criterion	Intercept Only	Intercept and Covariates
AIC	620.703	307.146
SC	624.812	418.095
-2 Log L	618.703	253.146

Figure 4 - 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.

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	91.6	Somers' D	0.863
Percent Discordant	5.4	Gamma	0.890
Percent Tied	3.0	Tau-a	0.427
Pairs	50049	c	0.931

Figure 5 - 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.

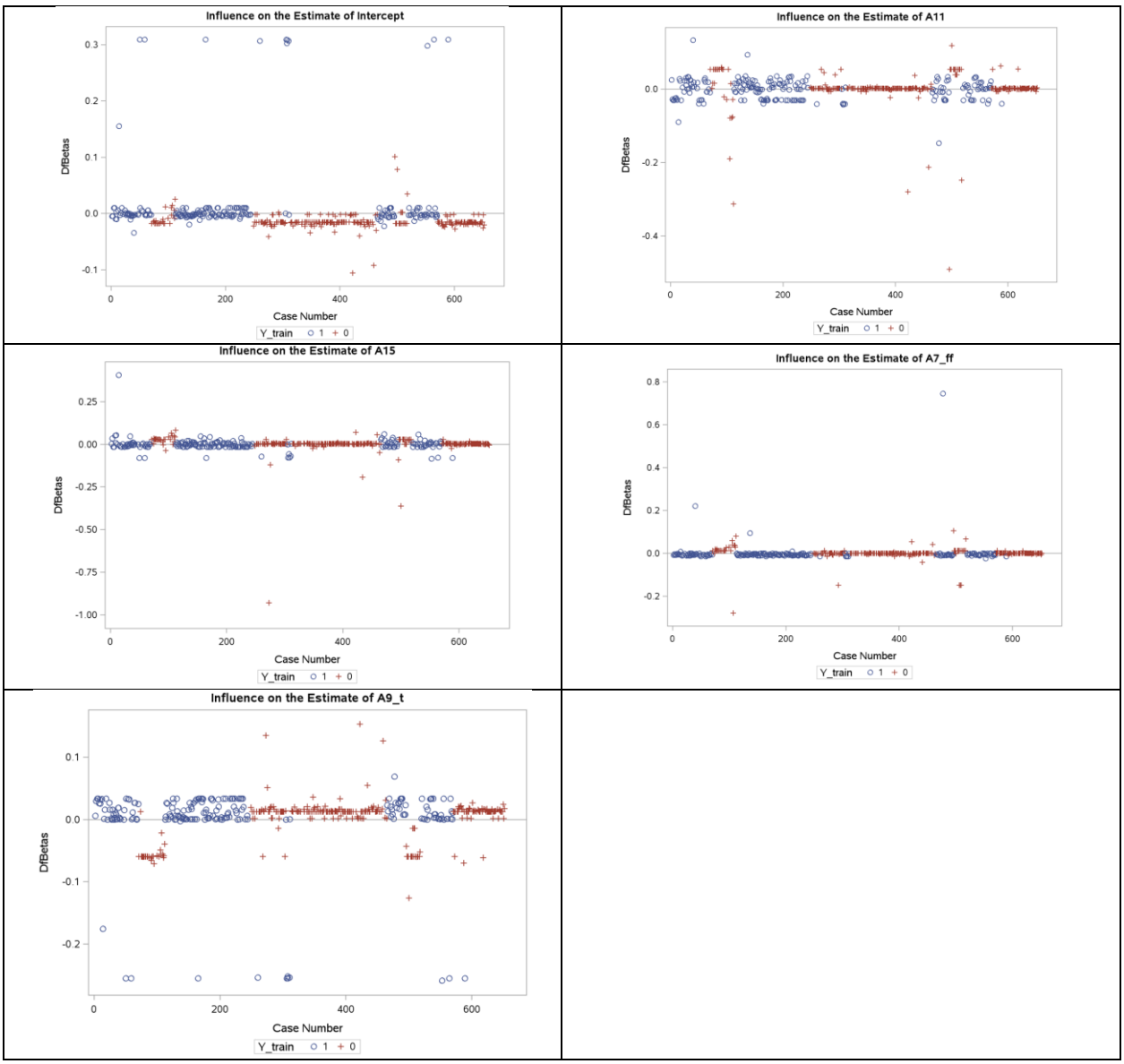


Figure 6 - DFBETAS statistic for Model #1

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.

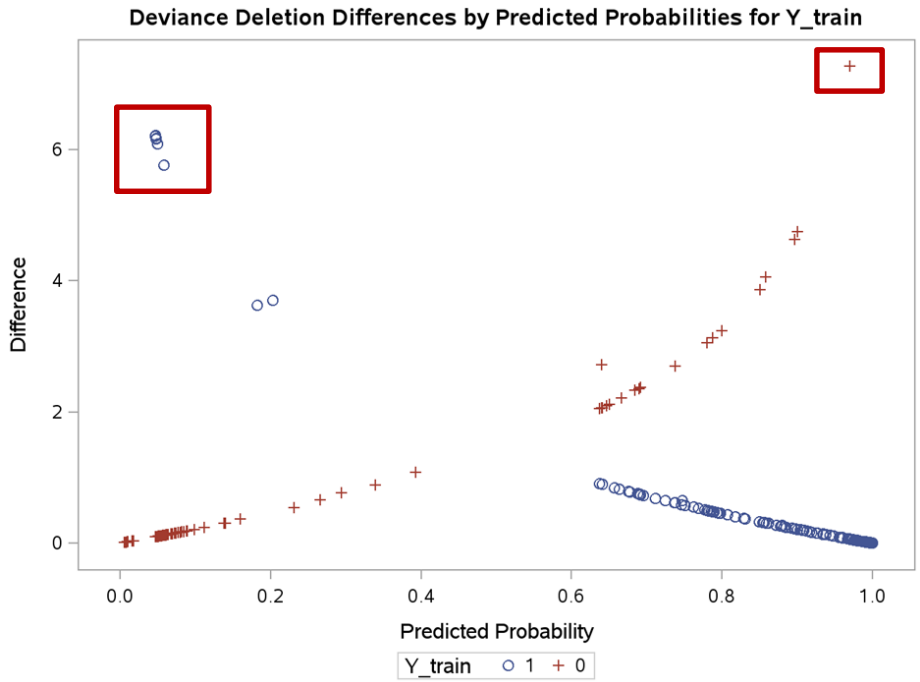


Figure 7 - SAS DIFDEV for Model #1

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.

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

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

Figure 8 - Model #2 Parameter and Odds Ratio Estimates

The logit for Model #2 is thus

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

The probability function is derived from the logit as follows:

$$\Pi(x) = \frac{e^{g(x)}}{1+e^{g(x)}} = \frac{e^{(-3.6287+\,3.9836*A9\_t\,+\,0.0227*A2\,+\,0.0527*A3)}}{1+e^{(-3.6287+\,3.9836*A9\_t\,+\,0.0227*A2\,+\,0.0527*A3)}}$$

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)

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

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

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	365.5570	26	<.0001
Score	280.7880	26	<.0001
Wald	123.2131	26	<.0001

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	620.703	307.146
SC	624.812	418.095
-2 Log L	618.703	253.146

Association of Predicted Probabilities and Observed Responses

Percent Concordant	91.6	Somers' D	0.863
Percent Discordant	5.4	Gamma	0.890
Percent Tied	3.0	Tau-a	0.427
Pairs	50049	c	0.931

Model #2 (Manager)

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

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

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

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

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

Figure 9 - 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.

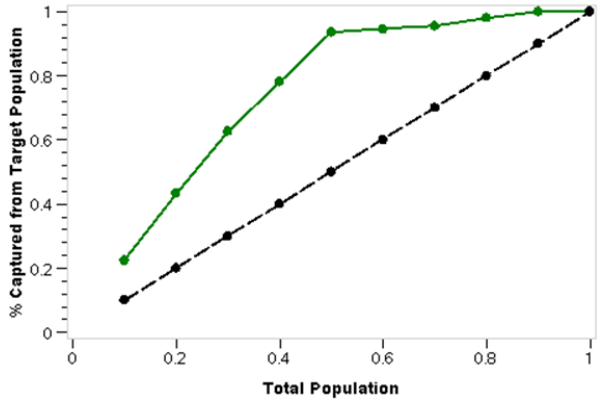
A lift chart and plot are generated from SAS for Model #1 and Model #2 as shown in Figure 10. By contrasting the lift charts we can compare the accuracy of the model at various points. As shown in the tables below the prediction rates for Model #1 are consistently higher as we approach maximum lift when 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)

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

Model #2 (Manager)

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
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



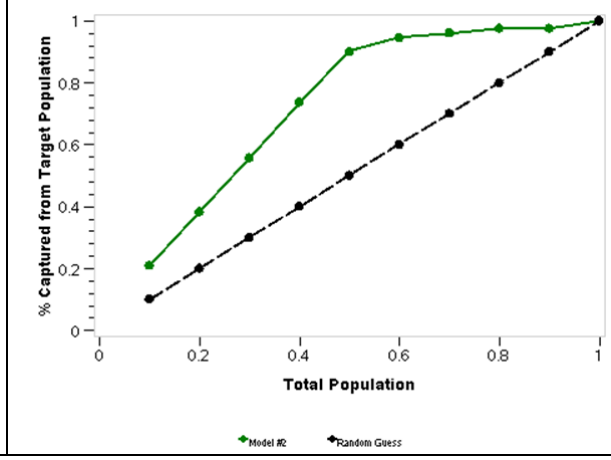


Figure 10 - Figure 9 - Lift Charts for Model #1 and #2 using Training data set

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).

Table of train by Y

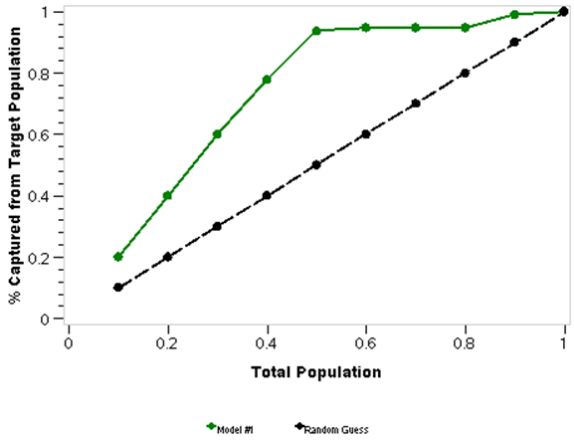
train	Y		
Frequency Percent Row Pct Col Pct	0	1	Total
0	108 16.54 53.20 30.25	95 14.55 46.80 32.09	203 31.09
1	249 38.13 55.33 69.75	201 30.78 44.67 67.91	450 68.91
Total	357 54.67	296 45.33	653 100.00

Figure 11 -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

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



Model #2 – Test Data

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

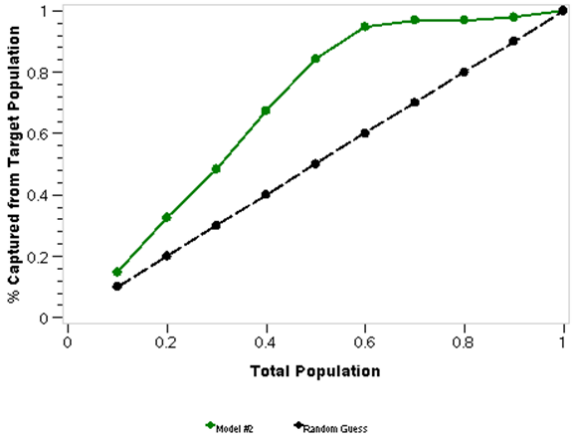


Figure 12 - Lift Charts for Model #1 and #2 using Test data set

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.