Derek Hughes
Assignment #6
Predict 410 – Sec 57

INTRODUCTION:
Here we are dividing our data set into two section – an in-sample training sample and an out-sample testing sample. We are selecting the best variables for Model 1 using the backward selection process and then comparing our model to Model 2 which a predefined model provided by the "manager." Both models are fitted to the training sample and compared using goodness-of-fit statistics and lift values. Next, both models are compared to the testing sample data to check for robust characteristics and the absence of overfitting. Again using the lift and KS values, we compare the performances of each model to the other and finally select the model that is most effective at predicting the probability of the response or target variable being equal to 1 or, in this case, approved credit.

RESULTS:

------ MODEL 1 vs. TRAINING DATA ---------

Below is the backward summary chart. The backward selection process begins with every variable included in the model. It systematically removes the variable with the lowest Wald Chi-square and highest probability value from the model one variable at a time. A threshold value for the probability value is used to stop the iterative process. This table shows the order in which each variable was removed from the backward selection process. Thus we can get a rough idea about which variables were considered the most irrelevant for this selection process. The variables that were accepted into the model should *not* be in this table and are shown in the Maximum Likelihood Estimates table shown later.

Summary of Backward Elimination					
Step	Effect Removed	DF	Number In	Wald Chi-Square	Pr > ChiSq
1	A13_s	0	26	.	.
2	A6_aa	1	25	0.0000	0.9977
3	A6_m	1	24	0.0008	0.9777
4	A3	1	23	0.0571	0.8111
5	A6_ff	1	22	0.1409	0.7074
6	A6_k	1	21	0.2492	0.6177
7	A6_q	1	20	0.4946	0.4819
8	A6_c	1	19	0.2848	0.5935
9	A13_g	1	18	0.5308	0.4663
10	A10_f	1	17	0.6713	0.4126
11	A2	1	16	0.7937	0.3730
12	A1_b	1	15	0.9181	0.3380
13	A7_bb	1	14	0.9882	0.3202
14	A7_h	1	13	0.5178	0.4718
15	A12_f	1	12	1.0196	0.3126
16	A7_v	1	11	1.2996	0.2543
17	A14	1	10	1.7137	0.1905
18	A6_i	1	9	1.7887	0.1811
19	A8	1	8	2.2819	0.1309
20	A6_w	1	7	2.6159	0.1058
21	A6_cc	1	6	3.0036	0.0831
22	A4_u	1	5	3.2175	0.0729
23	A6_x	1	4	3.8322	0.0503

The model fit statistics can be used to assess how well the model fits. In this case we are looking for the lowest value of the three, which is -2LogL. -2LogL is also called the Deviance of the model and is used in the next table to calculate the Likelihood Ratio. AIC is the same metric we described before in previous assignments, and we already know that SC is a form of AIC but has a higher penalty for more parameters added to 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

When testing if the model is significant, we must check if each of the three values (Likelihood ratio, score, and Wald) is significant. If so, then we can say that the model has more statistically significant predictive power with the variable(s) than without the variables. The Likelihood Ratio is especially interesting because it is used to compare the Deviance of the reduced model to the Deviance of the full model. As we can see, each metric has a probability of less than .0001 which says that the model is statistically significant.

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 maximum likelihood estimates (below) are used to determine the coefficients/estimates, the odd-ratio, probability or fitted values, and the test statistics to assess each parameter and the model. We can test if individual variables have significant predictive power. Here we can see than A11, A15, A7_ff, and A9_t are all significant at the $p<.05$ level. This is determined by comparing the Wald Chi-Square value for the parameter to the critical Chi-Square value for the relative degrees of freedom (1 for each variable). The Wald Chi-Square value is found by dividing the Estimate by the Standard Error of the parameter and squaring that result.

The Estimate in the table is the coefficient (*Bi*) of the parameter or of the intercept (constant). The estimates can be inserted to find the g(x) or logit or log-odds of the logistic regression equation and is as follows:
g(x) = -3.0087 + .2338*A11 + .000561*A15 -2.2218*A7_ff + 3.5735*A9_t

These numeric coefficients can be interpreted as the expected change in the logit for every unit change of the parameter with the other parameters held constant. For example, the expected change in the logit is 0.2338 for every one unit change in the A11 variable when all other variables are held fixed.

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

Interestingly, by taking the *e* of each parameter's estimate (or coefficient) we can calculate the odds-ratio, which is generally much easier for readers to understand and interpret. For example, for A9_t the odds-ratio is exp(3.5735) = 35.6411. This is easier to interpret as it means that the probability that Y=1 is 35.640 times more likely for every unit change of A9_t when A9_t is 1 instead of 0. Furthermore, by observing the confidence limits, if the confidence interval does NOT contain the value 1, the variable has a significant effect on the odds ratio. If the interval is below 1 the variable significantly lowers the odds ratio and vice versa if the interval is above 1.

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

Below, the Association of Predicted Probabilities and Observed Responses table values are used to evaluate the association between the predicted values versus the observed values. These measures rely on concordant and discordant pairs. Concordant pairs are those pairs where the lower ordered response value (often 0) has a lower predicted mean score than the observation with the higher ordered response value. In other words, it is the percent of correctly classified pairs. This is desirable, while discordant pairs have a higher predicted mean score for lower order response values (less desirable).

Somers' D is used to determine the strength and direction of relation between pairs of variables. It has a value of -1 to +1 with +1 meaning that all pairs agree or are concordant. A Somers' D value of .863 shows strong concordance with between the predicted and observed responses.

Gamma is similar to Somers' D except that it does not penalize for ties and therefore (using the same scale of -1 to +1) is usually higher value than Somers' D, which is what we see here as well (0.890 vs. 0.863).

Tau-a is similar to a generalized value of R-square that is derived from the likelihood ratio. It is defined to be the ratio of the difference between the number of concordant pairs minus the discordant pairs divided by the total number of possible pairs.

C is used to determine how well the model can discriminate the response. It's value ranges from 0.5 to 1, where 0.5 is randomly guessing (no predictive power). Thus we want a higher number and our number of .931 shows us that our model is very strong at discriminating the response value. C is also equivalent to the area under the ROC curve and can be used to compare models.

Thus, we can see that, taken together, based on our concordant/discordant values, Somers' D, Gamma, Tau-a, and c values that we have a strong model for predicting the response variable values correctly.

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

Next we create a lift chart table and lift chart plot where we can see how well the model predicts values when the observations are divided into ten "buckets." It is essentially a test of the effectiveness of the model by comparing the results with and without the model. For example, if the model had zero prediction value then we would expect the number of positively predicted responses to be evenly divided into each bucket so that, using ten evenly divided "buckets," twenty of the positively predicted responses (201/10 = 20 rounded) are found within the first bucket, twenty more of the positively predicted responses are found in the second bucket, etc.

However, if we find that we have a higher number of positively predicted responses for a given bucket than what we would expect on average by randomly guessing, that indicates that our model has a higher prediction power than randomly guessing. The "lift" is this difference in each bucket compared to what would be expected on average. It is represented by dividing the predicted amount (rate) by the theoretical amount (theory) for that bucket. The larger the "lift" the more predictive power the model has for that bucket compared to the others. For Model 1 against the training data, we can see that the largest lift is 2.2388.

Furthermore, to obtain the KS value we determine the percentage difference in each bucket of the proportion of positive predicted responses compared to the *cumulative* theoretical responses for that bucket. For example, for bucket three, the cumulative percentage of positive responses is .62687 (126/201) and the base rate for this bucket is .30 (.10 + .10 + .10 = .30). The difference is .62687 - .30 = .32687. The maximum difference found in any of the ten buckets is considered to be the Kolmogorov-Smirnov (KS) value. Thus, in this case, the maximum difference or KS value for Model 1 against the training data (in-sample) is .43532 found in bucket 5, which means the model is predicting at a 43.5% better percentage than randomly guessing in this bucket.

Finally, we can see that throughout the entire range of buckets this model has a higher percentage difference or predictive ability in each bucket compared to randomly guessing (all of the buckets have a difference (the "lift" column in the SAS table) higher than 10%). This is visually obvious by observing the lift chart as well.

NOTE: I completed this assignment a week ago (August 3, 2014) because I was going to be out of town from Thursday to late Sunday night when the assignment is due. I am now scrambling to update it with the limited number of hours I have available this morning (Thursday) before I have to leave out of town from the SAS data to the Excel data sheets because of the recent confusion posted early Thursday morning (about 1 am EST) about how to do the lift/KS tables and charts and that we should use these Excel sheets instead of the SAS output for this section. My terminology in the conclusions and descriptions may now vary as well because I initially used the term "lift" to describe the percentage differences (as shown on the SAS table under the column "lift") and am now trying to convert that terminology to "percentage differences" that's used in the Excel charts instead. I'm utterly confused as to what charts we are supposed to use for this section now so I'm simply including everything.

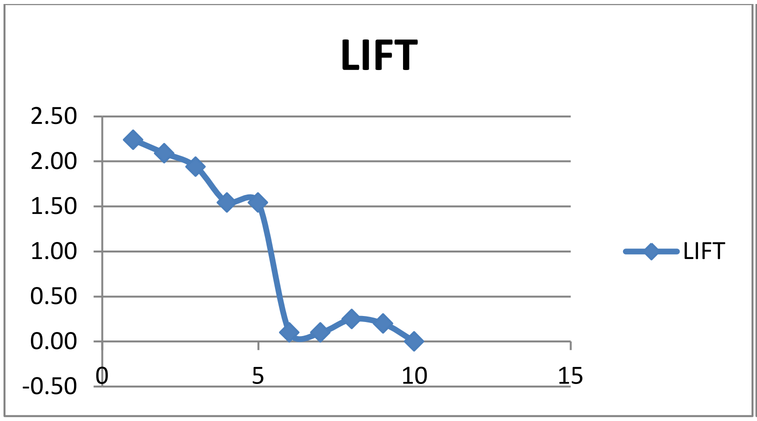
Model #1: In-Sample Training Data Lift Table

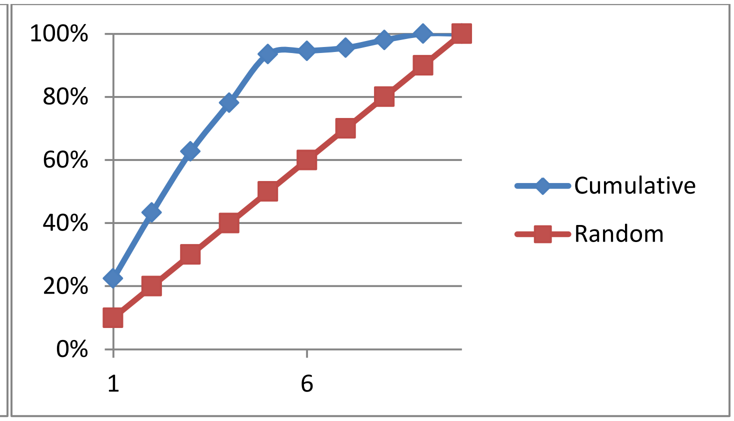
Obs	score_decile	Y_Sum	Nobs	cum_obs	model_pred	pred_rate	base_rate	lift
1	1	45	45	45	45	0.22388	0.1	0.12388
2	2	42	45	90	87	0.43284	0.2	0.23284
3	3	39	45	135	126	0.62687	0.3	0.32687
4	4	31	43	178	157	0.78109	0.4	0.38109
5	5	31	55	233	188	0.93532	0.5	0.43532
6	6	2	37	270	190	0.94527	0.6	0.34527
7	7	2	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

GROUP	Contacts	Responses	Rate	Theory	LIFT	Cumulative	RANDOM	Diff
1	45	45	45	20	2.24	22%	10%	12%
2	90	87	42	20	2.09	43%	20%	23%
3	135	126	39	20	1.94	63%	30%	33%
4	178	157	31	20	1.54	78%	40%	38%
5	233	188	31	20	1.54	94%	50%	44%
6	270	190	2	20	0.10	95%	60%	35%
7	315	192	2	20	0.10	96%	70%	26%
8	349	197	5	20	0.25	98%	80%	18%
9	423	201	4	20	0.20	100%	90%	10%
10	450	201	0	20	0.00	100%	100%	0%

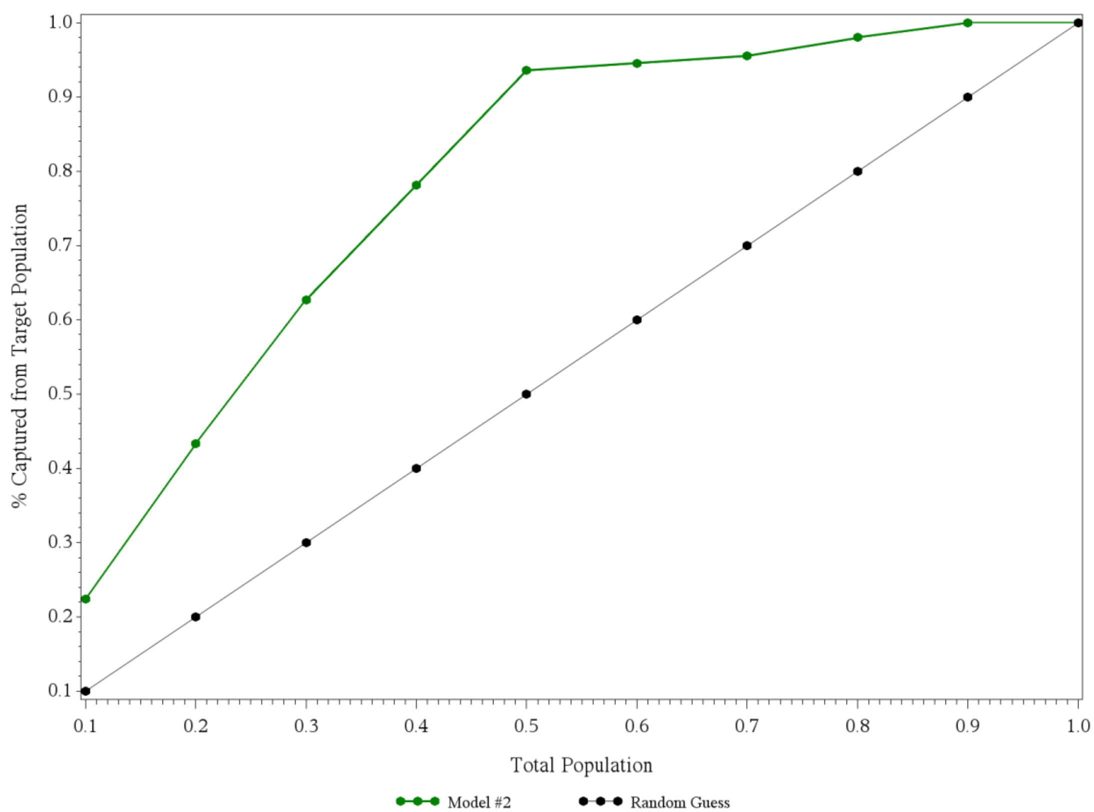
2.238806

KS= 44%





Model #1: In-Sample Training Data Lift Chart



------ MODEL 2 vs. TRAINING DATA ---------

The following tables and graphs are created from fitting the manager's model (Model 2) to the "in-sample" training data.

As we have already described each of the relevant metrics for each table/graph, we will discuss how the metrics in Model 2 compare to Model 1 in this section. There is no backward selection process here because the variables used in this model were based on the domain knowledge of the manager.

The model fit statistics for Model 2 shows a Deviance (-2LogL) of 332.739. Compared to Model 1's Deviance score of 281.046, we can say that Model 2 is not as good of a fit because it has more deviance from the observed values than Model 1.

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

Like Model 1, Model 2 is statistically significant in all three test metrics.

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

Using the Maximum Likelihood Estimates table we can derive the logit for the model and its coefficients for each parameter: $g(x) = -3.6287 + 3.9836*A9\_t + 0.0227*A2 + 0.0527*A3$.

One interesting point of note here is that the parameters A2 and A3 are not statistically significant at alpha .05, while every parameter in Model 1 was significant. Even though collectively the variables produce a model that is significant, some of the individual parameters are not.

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

Continuing with the results from the Analysis of Maximum Likelihood Estimates, the odds ratio estimates for A2 and A3 are not very strong. This means that the probability that Y=1 is only 1.023 and 1.054 times more likely for every unit change of A2 and A3 respectively. Interestingly, compared to Model 1, the parameter A9_t has a much higher odds ratio in Model 2 (53.712 vs. 35.640). This may be due to it having fewer parameters and because the parameters, A2 and A3, are individually insignificant; hence, this may allow A9_t to have a stronger overall influence on the model than it does in Model 1.

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

Like Model 1, Model 2 shows a strong concordant percentage at 89.1. Somer's D, Gamma, and Tau-a are also fairly decent values but, again, not as strong as those in Model 1 which were .863, .890, .427 respectively. Furthermore, Model 2's C value of .893 shows us that our model is strong at discriminating the response value.

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

Continuing with the theme that Model 2 isn't quite up to par with Model 1, when we examine the lift table and chart we can see that the KS value for Model 2 is lower than Model 1 (.40050 vs .43532). Also the percentage difference values (the "lift" column in the SAS table and which we want to be higher) are lower than those in Model 1 in every bucket except one bucket (bucket 7). The "lift" (from Excel chart) is lower than Model 1 as well (2.23 vs 2.08). This, again, shows us that Model 1 is superior to Model 2. Visually this is apparent as well in the plotted lift chart graph.

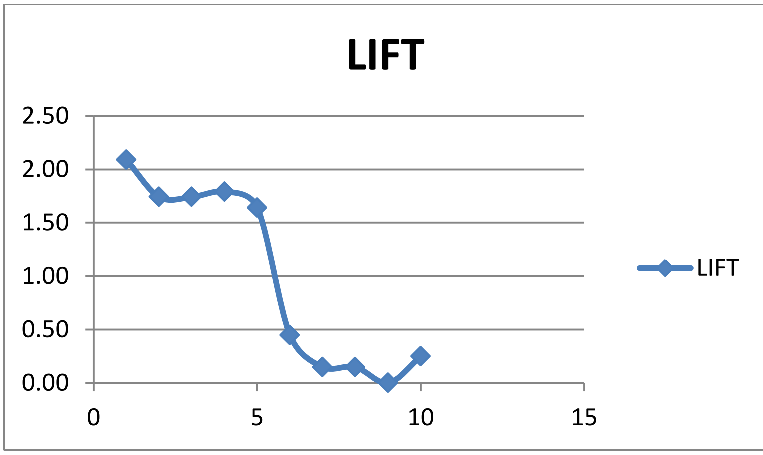
Model #2: In-Sample Lift Table

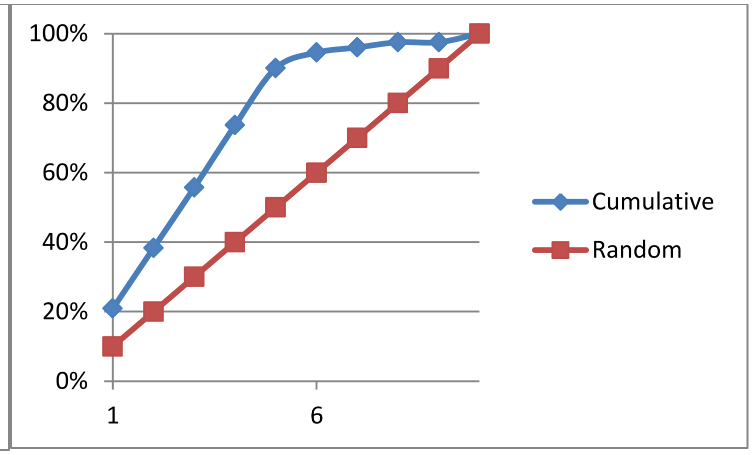
Obs	score_decile	Y_Sum	Nobs	cum_obs	model_pred	pred_rate	base_rate	lift
1	1	42	45	45	42	0.20896	0.1	0.10896
2	2	35	45	90	77	0.38308	0.2	0.18308
3	3	35	45	135	112	0.55721	0.3	0.25721
4	4	36	45	180	148	0.73632	0.4	0.33632
5	5	33	45	225	181	0.90050	0.5	0.40050
6	6	9	45	270	190	0.94527	0.6	0.34527
7	7	3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

GROUP	Contacts	Responses	Rate	Theory	LIFT	Cumulative	RANDOM	Diff
1	45	42	42	20	2.09	21%	10%	11%
2	90	77	35	20	1.74	38%	20%	18%
3	135	112	35	20	1.74	56%	30%	26%
4	180	148	36	20	1.79	74%	40%	34%
5	225	181	33	20	1.64	90%	50%	40%
6	270	190	9	20	0.45	95%	60%	35%
7	315	193	3	20	0.15	96%	70%	26%
8	360	196	3	20	0.15	98%	80%	18%
9	405	196	0	20	0.00	98%	90%	8%
10	450	201	5	20	0.25	100%	100%	0%

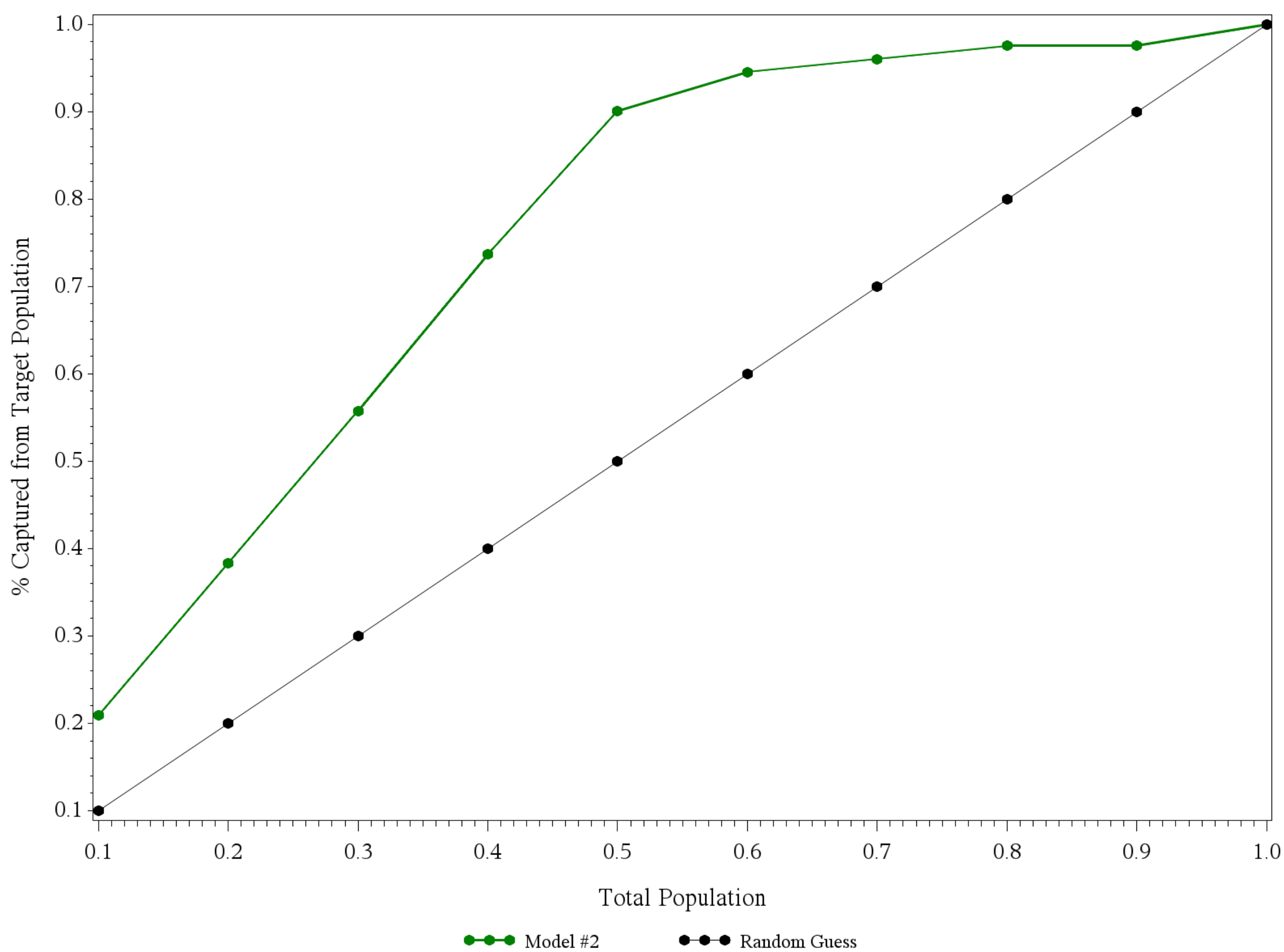
2.089552

KS= 40%





Model #2: In-Sample Lift Chart



------ MODEL 1 vs. TESTING DATA (OUT-SAMPLE) ---------

Compared to the training data, Model 1 does a very good job of replicating the same percentage difference (the "lift" column in the SAS table) values using the testing set (out-sample). Every bucket within the testing set has a percentage difference value (the "lift" column in the SAS table) that is within approximately .03 of the percentage difference value in the same bucket in the training set. However, the lift value (per Excel sheet) is lower in Model 1 vs testing sample compared to training sample (2.03 vs. 2.23). This indicates that Model 1 vs the testing data does not perform quite as well against the testing data and may indicate some slight overfitting, but I'd say the differences are not extraordinary. On the contrary, the KS value is almost identical (even better) in the testing set than the training set (.4368 vs. .4353). Overall, this tells us that Model 1 is fairly robust and probably does not suffer much overfitting.

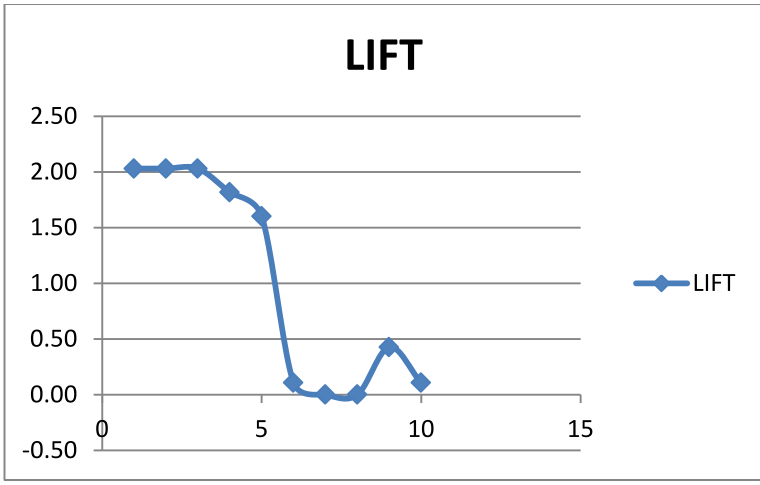
Model 1 vs TESTING SET

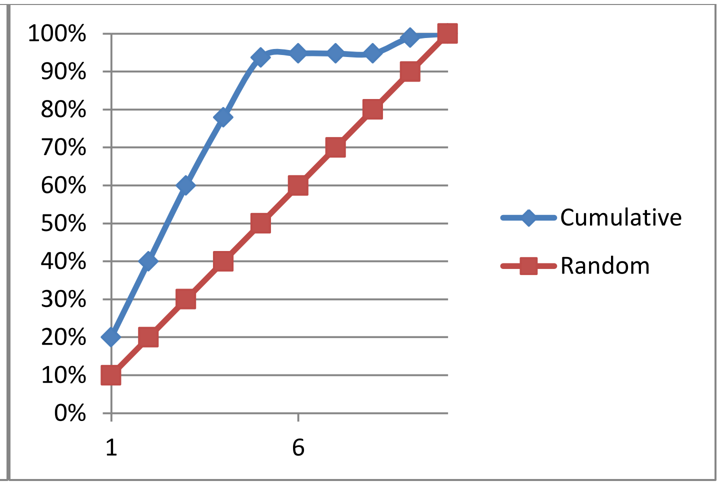
Obs	score_decile	Y_Sum	Nobs	cum_obs	model_pred	pred_rate	base_rate	lift
1	1	19	20	20	19	0.20000	0.1	0.10000
2	2	19	20	40	38	0.40000	0.2	0.20000
3	3	19	21	61	57	0.60000	0.3	0.30000
4	4	17	20	81	74	0.77895	0.4	0.37895
5	5	15	29	110	89	0.93684	0.5	0.43684
6	6	1	12	122	90	0.94737	0.6	0.34737
7	7	0	20	142	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

GROUP	Contacts	Responses	Rate	Theory	LIFT	Cumulative	RANDOM	Diff
1	20	19	19	9	2.03	20%	10%	10%
2	40	38	19	9	2.03	40%	20%	20%
3	61	57	19	9	2.03	60%	30%	30%
4	81	74	17	9	1.82	78%	40%	38%
5	110	89	15	9	1.60	94%	50%	44%
6	122	90	1	9	0.11	95%	60%	35%
7	142	90	0	9	0.00	95%	70%	25%
8	149	90	0	9	0.00	95%	80%	15%
9	191	94	4	9	0.43	99%	90%	9%
10	203	95	1	9	0.11	100%	100%	0%

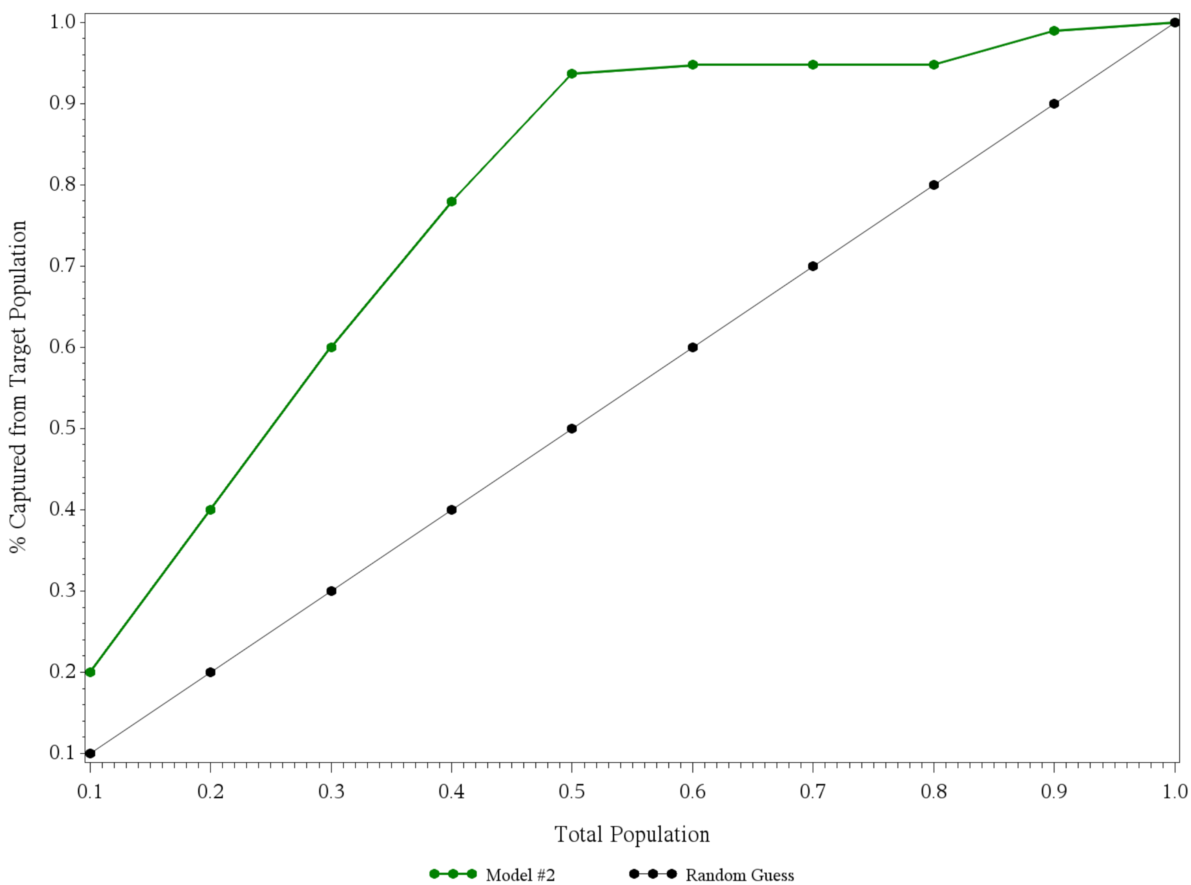
2.03

KS= 44%





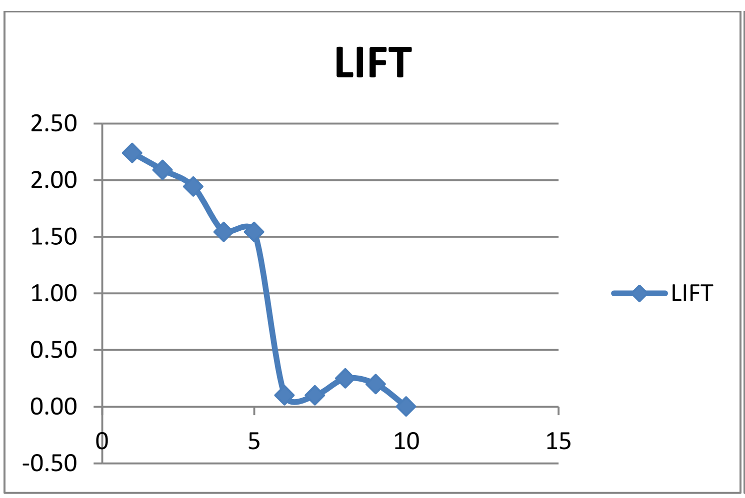
Model #1: Out-Sample Testing Data Lift Chart

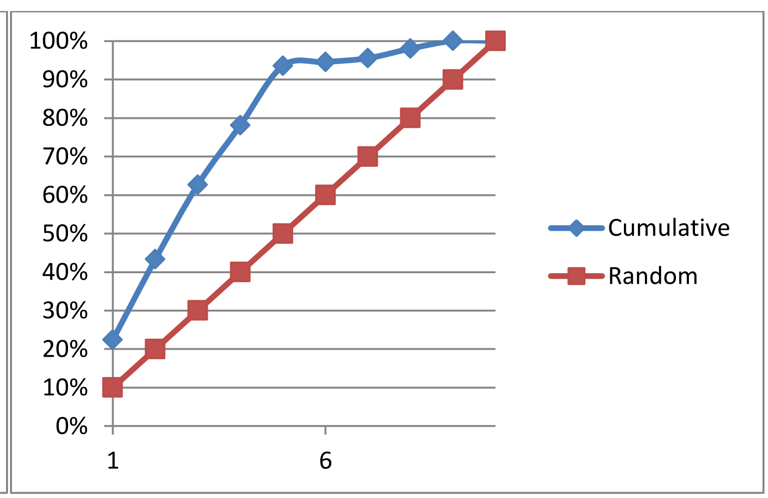


Model 1 vs TRAINNG SET

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

GROUP	Contacts	Responses	Rate	Theory	LIFT	Cumulative	RANDOM	Diff
1	45	45	45	20	2.24	22%	10%	12%
2	90	87	42	20	2.09	43%	20%	23%
3	135	126	39	20	1.94	63%	30%	33%
4	178	157	31	20	1.54	78%	40%	38%
5	233	188	31	20	1.54	94%	50%	44%
6	270	190	2	20	0.10	95%	60%	35%
7	315	192	2	20	0.10	96%	70%	26%
8	349	197	5	20	0.25	98%	80%	18%
9	423	201	4	20	0.20	100%	90%	10%
10	450	201	0	20	0.00	100%	100%	0%
					2.238806		KS=	44%





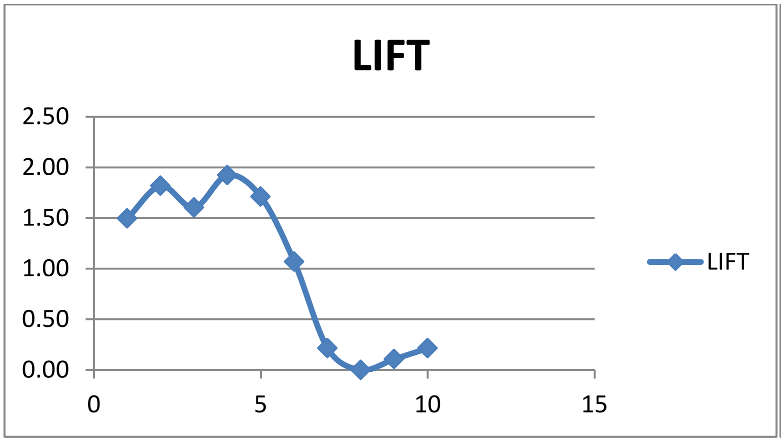
------ MODEL 2 vs. TESTING DATA (OUT-SAMPLE) ---------

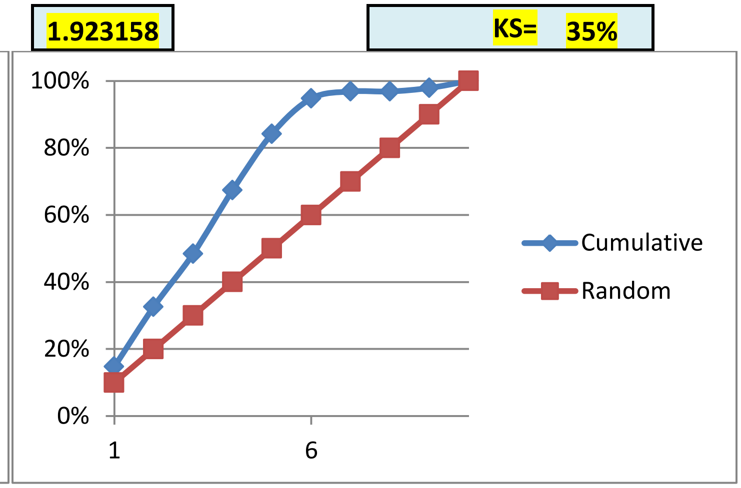
Compared to the training data, Model 2 does a fair job of replicating the same percentage difference values (the "lift" column in the SAS table) on the testing set (out-sample), but not as well as Model 1. Every bucket within the testing set has a percentage difference value (the "lift" column in the SAS table) that is within approximately .07 (this was .03 for Model 1) of the percentage difference value in the same bucket in the training set. Actually, for buckets 5-10, Model 2 has almost identical percentage difference values (the "lift" column in the SAS table). Its larger percentage differences are mainly found in the first four buckets. Overall, the differences in percentage difference values between the training and testing sets aren't dramatic, but compared to the variances Model 1 showed when comparing its fit over the two data sets Model 2 isn't quite as good. Furthermore, the KS value is lower for the testing set vs the training set (.3474 vs. .4005) and the maximum lift value (per Excel sheet) is lower as well (1.9232 vs. 2.0896). This indicates that Model 2 may suffer some overfitting issues which probably are not dramatic but certainly worth investigating. Visually these results can be seen in the lift chart graph shown below.

Model 2 vs TESTING SET

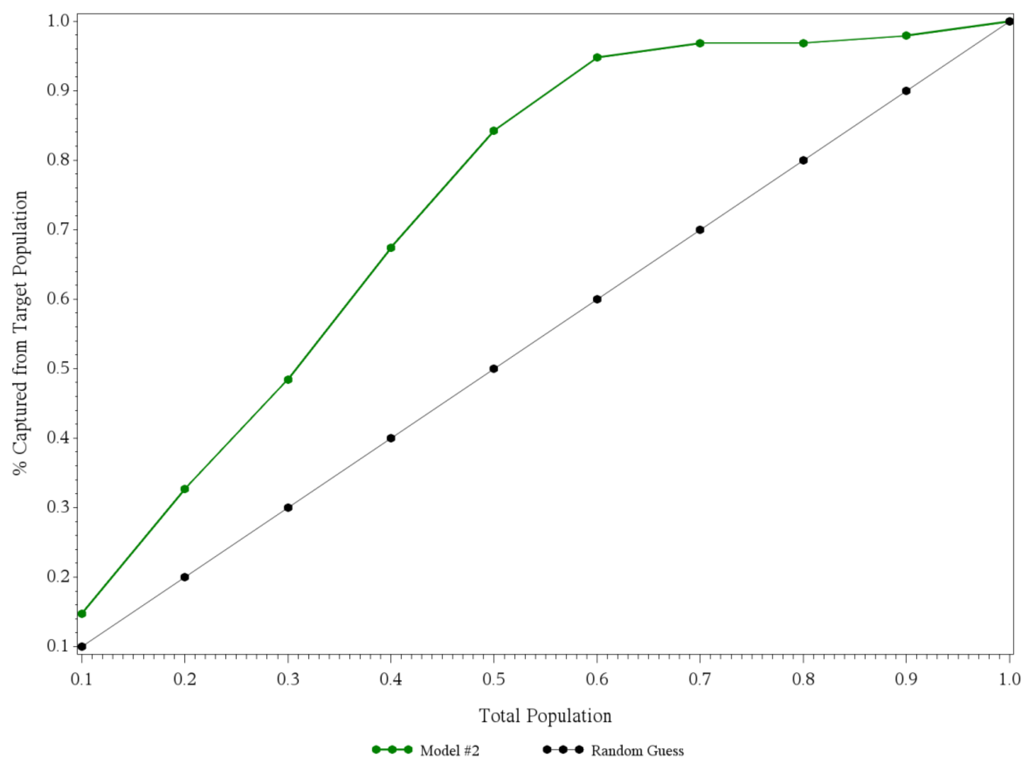
Obs	score_decile	Y_Sum	Nobs	cum_obs	model_pred	pred_rate	base_rate	lift
1	1	14	20	20	14	0.14737	0.1	0.04737
2	2	17	20	40	31	0.32632	0.2	0.12632
3	3	15	21	61	46	0.48421	0.3	0.18421
4	4	18	20	81	64	0.67368	0.4	0.27368
5	5	16	20	101	80	0.84211	0.5	0.34211
6	6	10	21	122	90	0.94737	0.6	0.34737
7	7	2	20	142	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

GROUP	Contacts	Responses	Rate	Theory	LIFT	Cumulative	RANDOM	Diff
1	20	14	14	9	1.50	15%	10%	5%
2	40	31	17	9	1.82	33%	20%	13%
3	61	46	15	9	1.60	48%	30%	18%
4	81	64	18	9	1.92	67%	40%	27%
5	101	80	16	9	1.71	84%	50%	34%
6	122	90	10	9	1.07	95%	60%	35%
7	142	92	2	9	0.21	97%	70%	27%
8	163	92	0	9	0.00	97%	80%	17%
9	183	93	1	9	0.11	98%	90%	8%
10	203	95	2	9	0.21	100%	100%	0%





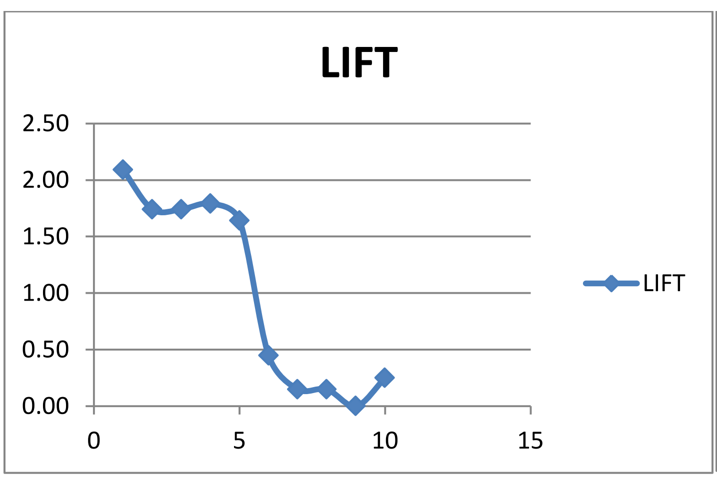
Model #2: Out-Sample Testing Data Lift Chart

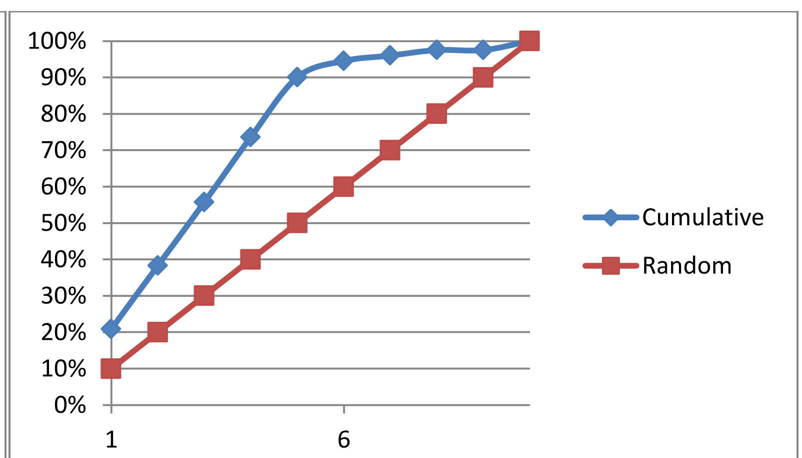


Model 2 vs. TRAINING DATA

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

GROUP	Contacts	Responses	Rate	Theory	LIFT	Cumulative	RANDOM	Diff
1	45	42	42	20	2.09	21%	10%	11%
2	90	77	35	20	1.74	38%	20%	18%
3	135	112	35	20	1.74	56%	30%	26%
4	180	148	36	20	1.79	74%	40%	34%
5	225	181	33	20	1.64	90%	50%	40%
6	270	190	9	20	0.45	95%	60%	35%
7	315	193	3	20	0.15	96%	70%	26%
8	360	196	3	20	0.15	98%	80%	18%
9	405	196	0	20	0.00	98%	90%	8%
10	450	201	5	20	0.25	100%	100%	0%
					2.089552		KS=	40%





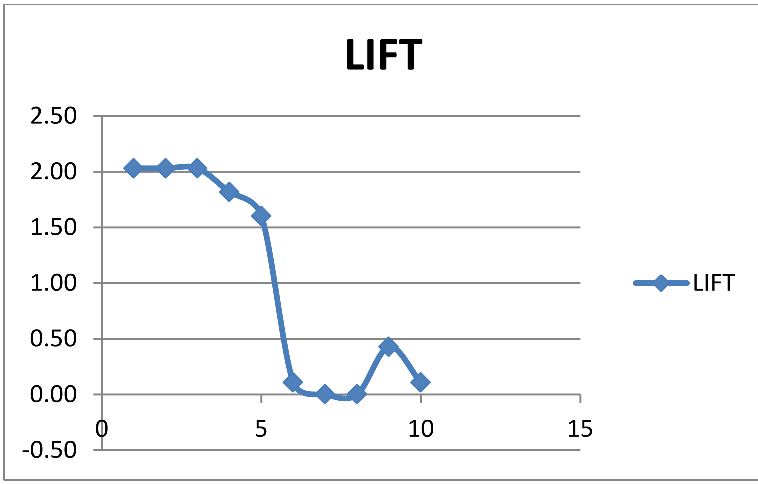
---------- MODEL 1 VS MODEL 2 (OUT-SAMPLE TESTING) -------------

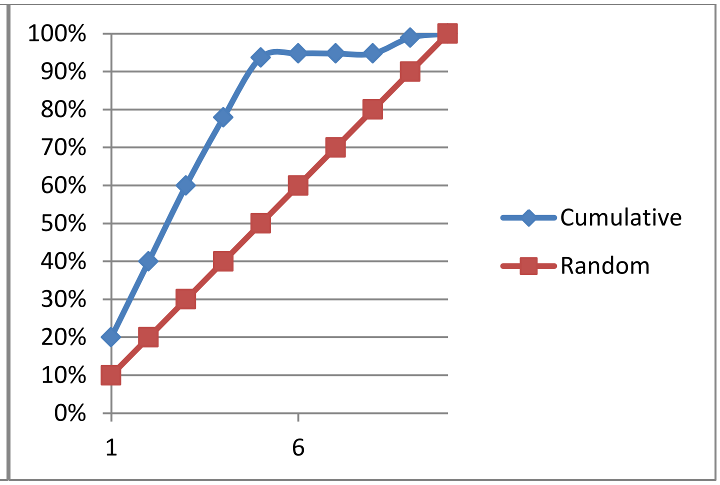
Comparing percentage differences values (the "lift" column in the SAS table) of Model 1 with the out-sample against the percentage differences values (the "lift" column in the SAS table) from Model 2 with the out-sample shows that Model 1 is clearly the better model. In every single bucket (except bucket 6 where both models have the same percentage differences values (the "lift" column in the SAS table)), Model 1 has a higher percentage difference value. This means that, comparatively speaking, the predictive power of Model 1 is better than Model 2 in every bucket (bucket 6 notwithstanding). Furthermore, the KS value of Model 1 is almost .10 higher than that Model 2 (.43684 vs. .34737). The maximum lift from Model 1 is also larger than Model 2 (2.03 vs. 1.9232). This makes it easy to determine that Model 1 is the model to use to achieve the best predictive probability results of the response variable.

Model 1 vs TESTING SET

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

GROUP	Contacts	Responses	Rate	Theory	LIFT	Cumulative	RANDOM	Diff
1	20	19	19	9	2.03	20%	10%	10%
2	40	38	19	9	2.03	40%	20%	20%
3	61	57	19	9	2.03	60%	30%	30%
4	81	74	17	9	1.82	78%	40%	38%
5	110	89	15	9	1.60	94%	50%	44%
6	122	90	1	9	0.11	95%	60%	35%
7	142	90	0	9	0.00	95%	70%	25%
8	149	90	0	9	0.00	95%	80%	15%
9	191	94	4	9	0.43	99%	90%	9%
10	203	95	1	9	0.11	100%	100%	0%
					2.03		**KS=**	**44%**

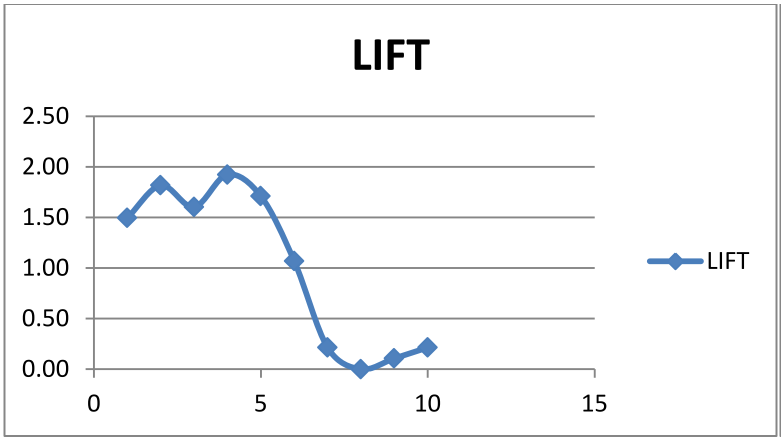


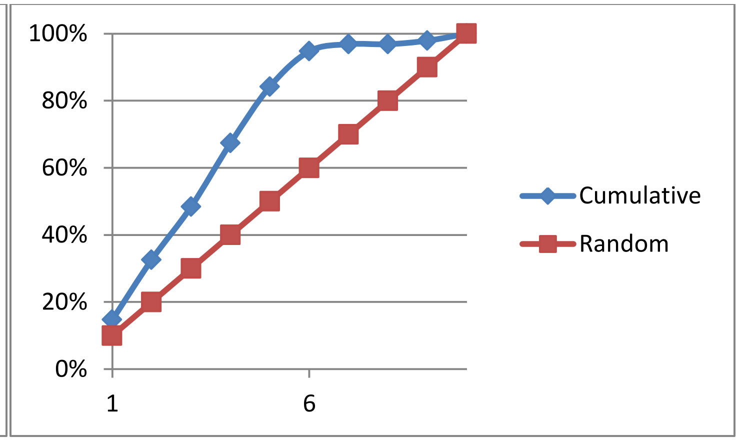


Model 2 vs TESTING SET

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

GROUP	Contacts	Responses	Rate	Theory	LIFT	Cumulative	RANDOM	Diff
1	20	14	14	9	1.50	15%	10%	5%
2	40	31	17	9	1.82	33%	20%	13%
3	61	46	15	9	1.60	48%	30%	18%
4	81	64	18	9	1.92	67%	40%	27%
5	101	80	16	9	1.71	84%	50%	34%
6	122	90	10	9	1.07	95%	60%	35%
7	142	92	2	9	0.21	97%	70%	27%
8	163	92	0	9	0.00	97%	80%	17%
9	183	93	1	9	0.11	98%	90%	8%
10	203	95	2	9	0.21	100%	100%	0%
					1.923158		KS=	35%





CONCLUSION:

First, we took our data set and divided it into a sample of data (70% of the data) to develop our models and a sample (30% of the data) to test our completed model for redundancy and overfitting.

After dividing the data we focused on developing our model. We selected a model with a dichotomous response variable to determine the probability that our response variable will be equal to 1. We used PROC FREQ to perform an EDA to determine which variables we can combine into base variables and dummy variables. Then we used PROC MEANS to explore our continuous variables for cut points to "cut" them into discrete variables. Next we used PROC LOGISTIC to fit the model as a logistic regression and evaluated the results using a lift table and lift graph. We repeated this step with a model provided by the "manager."

Overall, our results showed that both models were significant, even though individually some of the parameters in Model 2 were insignificant. Furthermore, we saw that both models had higher predictive lift values for every bucket and therefore every area of the model compared to randomly guessing. These results were supported by the high percent concordance and C values in both models. This showed that either model would do a fairly good job of predicting the probability of the response variable being equal to 1. Additionally, while each model would do a decent job, Model 1 consistently showed better metrics in model fit and predictive abilities.

After we learned that both models were significant and good predictors of probability, we tested both models against the test or out-sample data. We compared each model's associated lift values against the lift values they produced from their training sample data, and finally compared each model's lift data results and KS values against the other when both models were fitted to the testing out-sample data. Again, supporting the results we found from the training data (in-sample), we learned that Model 1 and Model 2 were fairly robust, most likely didn't suffer much from overfitting (although both did not perform quite as well with their lift values as when fitted to the training data). Even though both had a slight drop in lift value and KS value (Model 1 KS value was virtually identical to its training sample KS value), both were still fairly decent predictors of the probability that the response variable (credit approval) will equal 1 (credit was approved) compared to randomly guessing. Finally, we learned that Model 1 further distinguished itself as being the better model of the two when comparing the lift, percentage differences between buckets, and KS values of the two from the testing out-sample data.

CODE :

```
* creating the macro for data set;
%let PATH = /courses/u_northwestern.edu1/i_833463/c_3505/SAS_Data/;
%let NAME = MYDATA;
%let LIB = &NAME..;


libname &NAME. "&PATH." access=readonly;

%let INFILE = &LIB.credit_approval;
%let TEMPFILE = TEMPFILE;



* running proc freq to determine which variables have a small number (<=30)
        of observations in each category and thus can be combined into a base category,
        for variables with >30 observations in all categories the category with
        smallest number of observations is used as base;

proc freq data=&infile.;
tables A1 A4 A5 A6 A7 A9 A10 A12 A13 A16;
run;

* used to view continuous variables to determine cut points to
        convert into discrete variables;
proc means data=&infile. p5 p10 p25 p50 p75 p90 p95;
class A16;
var A2 A3 A8 A11 A14 A15;
run;


* creating the database divided into training and testing divisions,
        setting A16 as the dichotomous dependent response variable;
data &tempfile.;
        set &infile.;

label Y_train = "Received loan or no loan";
label A1 = "Living on own";
label A2 = "Yearly income";
label A3 = "Employers in past 10 years";
label A4 = "Have credit cards";
label A5 = "Have outstanding debt";
label A6 = "Number of rooms in house";
label A7 = "Married, separated, divorced, single";
label A8 = "Years remaining on mortgage";
```

```
label A9_t = "Own a car";
label A10 = "Own a house";
label A11 = "Number of dependents";
label A12 = "Employed";
label A13 = "Category title";
label A14 = "Continuous variable";
label A15 = "Another continuous variable";

* create training and testing data;
	u=uniform(123);
	if (u<0.7) then train=1;
		else train=0;



	if A16 = '+' then Y=1;
	else if A16 = '-' then Y=0;
	else Y=.;

	if (train=1) then Y_train=Y;
		else Y_train=.;

	* discretize continuous variables;
	if (A2<20) then A2_discrete=1;
	else if (A2<30) then A2_discrete=2;
	else if (A2<40) then A2_discrete=3;
	else A2_discrete=4;

	if (A3<1) then A3_discrete=1;
	else if (A3<4) then A3_discrete=2;
	else if (A3<8) then A3_discrete=3;
	else A3_discrete=4;

	if (A8<1) then A8_discrete=1;
	else if (A8<4) then A8_discrete=2;
	else if (A8<8) then A8_discrete=3;
	else A8_discrete=4;

	if (A11<2) then A11_discrete=1;
	else if (A11<5) then A11_discrete=2;
	else if (A11<10) then A11_discrete=3;
	else A11_discrete=4;

	if (A14<150) then A14_discrete=1;
	else if (A14<225) then A14_discrete=2;
	else if (A14<325) then A14_discrete=3;
```

```
        else A14_discrete=4;

        if (A15 < 1.5) then A15_discrete=1;
        else if (A15 < 250) then A15_discrete=2;
        else if (A15 < 1001) then A15_discrete=3;
        else A15_discrete=4;

* change variables to appropriate formats
        (continuous to discrete and categorical to dummy_variables)
        I'm combining any category with less than 31 observations and
        using them for the base;

        * A1 base is 'a';
        if (A1 ='b') then A1_b=1; else A1_b=0;
        label A1_b = "Living alone";
        label A1_a = "Living with others";

  * A4 base is 'l, y';
        if (A4 ='u') then A4_u=1; else A4_u=0;
        label A4_u = "Have credit cards";
        label A4_y = "No have credit cards";
        label A4_l = "Have debit cards";

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

  * A6 base 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 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 base is 'f';
        if (A9 ='t') then A9_t=1; else A9_t=0;
```

```
  * A10 base is 't';
        if (A10 ='f') then A10_f=1; else A10_f=0;

  * A12 base is 't';
        if (A12 ='f') then A12_f=1; else A12_f=0;

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

        * delete missing values;
        if (a1='?') or (a4='?') or (a5='?') or (a6='?') or (a7='?') or (a9='?') or (a10='?') or (a12='?') or (a13='?')
        or (a2=.) or (a3=.) or (a8=.) or (a11=.) or (a14=.) or (a15=.)
                then delete;
run;
*****************  IN-SAMPLE TESTING *******************;

* FIT THE MODEL (MODEL 1) TO TRAINING DATA;
        proc logistic data=&tempfile. 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_m A6_q A6_w A6_x
                A7_bb A7_ff A7_h A7_v
                A9_t A10_f A12_f A13_g / selection=backward;
                output out=model_data pred=yhat;
                title "Model 1 vs In-Sample Training data";
        run;


******* PROC RANK AND LIFT CHART FOR MODEL 1 vs TRAINING DATA*****;

* PROC RANK model divides values into groups (highest scores to
        lowest score decile);
        proc rank data=model_data out=training_scores descending groups=10;
        var yhat;
        ranks score_decile;
        where train=1;
        title "Model 1 vs In-Sample Training data";
        run;

* create lift chart;
        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 for training data;
		*95 represents the number of successes for the testing data;
		*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 #1: In-Sample Training Data Lift Chart';
	symbol1 color=green interpol=join w=2 value=dot height=1;
	symbol2 color=black interpol=join w= 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;


* FIT THE MANAGERS MODEL (MODEL 2) TO TRAINING DATA;
          proc logistic data=&tempfile. descending;
          model Y_train= A9_t A2 A3;
          output out=model_data2 pred=yhat;
          title "Model 2 vs In-Sample Training data";
          run;



******* PROC RANK AND LIFT CHART FOR MODEL 2 vs TRAINING DATA*****;

* PROC RANK model divides values into groups (highest scores to
          lowest score decile);
          proc rank data=model_data2 out=training_scores descending groups=10;
          var yhat;
          ranks score_decile;
          where train=1;
          title "Model 2 vs In-Sample Training data";
          run;

* create lift chart;
          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 for training data;
		*95 represents the number of successes for the testing data;
		*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= 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-SAMPLE TESTING *******************;


****** PROC RANK AND LIFT CHART FOR MODEL 1 vs TESTING DATA*****;

* PROC RANK model divides values into groups (highest scores to
	lowest score decile);
	proc rank data=model_data out=testing_scores descending groups=10;
```

```
        var yhat;
        ranks score_decile;
        where train=0;
        title "Model 1 vs Out-Sample Testing data";
        run;

* create lift chart;
        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;

                        *201 represents the number of successes for training data;
                        *95 represents the number of successes for the testing data;
                        *this value will need to be changed with different samples;
                        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 #1: Out-Sample Testing Data Lift Chart';
        symbol1 color=green interpol=join w=2 value=dot height=1;
        symbol2 color=black interpol=join w= 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;
******* PROC RANK AND LIFT CHART FOR MODEL 2 vs TESTING DATA*****;

* PROC RANK model divides values into groups (highest scores to
    lowest score decile);
    proc rank data=model_data2 out=testing_scores descending groups=10;
    var yhat;
    ranks score_decile;
    where train=0;
        title "Model 2 vs Out-Sample Testing data";
    run;

* create lift chart;
    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;

        *201 represents the number of successes for training data;
        *95 represents the number of successes for the testing data;
        *this value will need to be changed with different samples;
        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-Sample Testing Data Lift Chart';
    symbol1 color=green interpol=join w=2 value=dot height=1;
    symbol2 color=black interpol=join w= 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;



* used to view success of response varaible (Y=1) for training and testing samples;
proc freq data=&tempfile.;
    tables train*Y;
    run;
```

****** BINGO BONUS ***** I am going for all 20 available bingo bonus points (non-R related).**

1) labels and macro variables were entered into the first data step of this SAS program

An *excerpt* of the macro code from above is provided here so the instructor doesn't have to search for it. ☺

```
* creating the macro for data set;
%let PATH = /courses/u_northwestern.edu1/i_833463/c_3505/SAS_Data/;
%let NAME = MYDATA;
%let LIB = &NAME..;


libname &NAME. "&PATH." access=readonly;

%let INFILE = &LIB.credit_approval;
%let TEMPFILE = TEMPFILE;

*excerpt from data set with some of the label code
data &tempfile.;
	set &infile.;


label Y_train = "Credit approval";
label A1 = "Living on own";
label A2 = "Yearly income";
label A3 = "Employers in past 10 years";
label A4 = "Have credit cards";
label A5 = "Have outstanding debt";
label A6 = "Number of rooms in house";
label A7 = "Married, separated, divorced, single";
label A8 = "Years remaining on mortgage";
label A9_t = "Own a car";
label A10 = "Own a house";
label A11 = "Number of dependents";
label A12 = "Employed";
label A13 = "Category title";
label A14 = "Continuous variable";
label A15 = "Another continuous variable";
```

2) deploy model for Manager's Model 2

```
data chanceForLoan;
	set &tempfile.;
	TEMP = -3.6287 + 3.9836 * A9_t + 0.0227 * A2 + 0.0527 * A3;
	P2_TARGET = 1.0 / (1.0 + exp(-1*TEMP));
	title "Data set with deployed Manager's Model 2 for credit loan";
```

```
	run;

proc print data=chanceForLoan (obs=10);
	run;
```

3) replace missing values with mean of that variable

One approach is to use the mean of the variable in which the missing values were found. This is a quick, easy way to recover missing values and the estimated mean for that variable is not affected. Some drawbacks are that bias can be introduced if the missing values are not completely random and/or are associated with some event that's being reflected in the data. Furthermore, using the mean will reduce the overall variance and lower the standard error which can introduce even more bias into the results. Overall it seems that mean substitution is most effective when the number of missing values is small.

The pros of substituting for missing values is that we can still use the remaining portion of the observation's data (if it has more than one variable for that observation), it eliminates database attrition, reduces bias due to some variables having more missing values than others, and allows more effective weighting options. Some cons are that it can generate inconsistent data, distort relationships, and reduce variance of estimates.

Here is a sample code I could add to the data step in #2 to fix the missing data:

```
proc standard data=&tempfile. replace
	out=meanReplace;
	run;

proc print data=meanReplace (obs=25);
	run;
```