Binary Classification Modeling Final Deliverable Using Logistic Regression to Build Credit Scores Lauren Cohen & Lillian King

Supervised by Michael Frankel

Kennesaw State University

Submitted 05/02/2022

to fulfill the requirements for STAT4330

**Executive Summary**

The purpose of this report is to create a credit scoring model to maximize the profitability of extending credit to customers. The original dataset contained over 1.2 million customers with over 300 predictors for each customer. SAS was used for the entirety of this project. The customers' delinquency on payments was used to create the binary predictor of whether they were considered a credit risk. The first task in the project was to re-code the data since there were many missing values and coded values for the variables. The median was used to impute missing and coded values since many of the variables were severely right skewed.

Variables with large proportions of missing or coded data were removed to avoid imputing a significant portion of the values. Variables were then clustered to further reduce the number of predictors to be used in the model.

In the next phase of the project, variable transformations were performed to create additional variables to see whether different forms of variables yielded significant results in the final model. Some transformations which were included were calculating the odds of the variables and creating discrete indicators for variables which were continuous. Two primary methods were used in creating the transformations: an automatic process using a procedure in SAS and a manual process which was left up to the model creator.

In the last phase of the project, the logistic regression was run on the variables and the different transformed versions of the variables which were in the dataset. The main statistics used to assess model performance were the percent of concordant observations and the KS statistic. From this logistic regression, only the top 3 predictors were chosen to remain in the model to reduce model complexity. The model was validated against 10 validation datasets to assess the consistency of the model. The profitability was then calculated by looking at the number of people the model correctly predicted to not default, and each of these customers on average generated a revenue of $250. Of those that do default but are predicted to be low risk customers, these represent an average loss of half of the customer's credit limit. Using this model, it was determined that every 1,000 customers who are extended credit yield an average profit of $105,750.33. In addition to this model, the final model which included the original number of significant variables was considered. This model yielded an average profit of $111,504.77 per 1,000 customers. It is important to see that using the less complex model results in approximately $6,000 less profit per 1,000 customers. This model is more interpretable and easier to present to a client and is easier to implement and maintain.

**Introduction**

This research paper describes the process and results of developing a binary classification model, using Logistic Regression, to generate Credit Risk Scores. These scores are then used to maximize a profitability function.

The data for this project came from a Sub-Prime lender. Three datasets were provided:

* CPR. 1,462,955 observations and 338 variables. Each observation represents a unique customer. This file contains all the potential predictors of credit performance. The variables have differing levels of completeness.
* PERF. 17,244,104 observations and 18 variables. This file contains the post hoc performance data for each customer, including the response variable for modeling – DELQID.
* TRAN. 8,536,608 observations and 5 variables. This file contains information on the transaction patterns of each customer.

Each file contains a consistent “MATCHKEY” variable which was used to merge the datasets. The process for the project included:

Assignment of Data Cleansing and Dependent Variable Imputation

Odds, Correlation

and Plots

Multicollinearity

assessment using Regression and VIF

Discretization and transformations

Sampling

Model Development

Model Evaluation

Data Discovery

Variable

Preparation

Modeling

Each of these processes will be discussed in turn.

**Data Discovery**

The purpose of the first phase of the project is to reduce the number of variables in the original dataset before the modeling phase. The CPR and PERF datasets are merged together by MATCHKEY to yield a dataset with 1,743,505 observations and 356 variables. The PERF dataset contains a variable DELQID (the number of cycles that a customer is late on payment), which is used to assign a binary variable "GOODBAD" indicating whether a customer is considered a credit risk. Since the PERF dataset contains monthly data for customers during the observation period, the maximum value of DELQID is chosen to represent each customer. The reason for this conservative approach is explained further below. Values for DELQID range from 0-7, where 0 means the customer is too new to rate, 1 means the customer is in the current pay period, 2 means the customer is one cycle late, 3 means the customer is two cycles late, and so on. For example, if a customer is tracked for twelve months in the performance period, and at some point, during the performance period that customer is four cycles late on payment, their DELQID is assigned as 5, even if by the end of the performance period they are back to being one cycle late on payment. The "GOODBAD" variable is then assigned a value of 1, "bad", if DELQID is 3 or greater (two or more cycles late) for a customer. A table of the distribution of GOODBAD is shown below in Table 1.

Table 1. Distribution of GOODBAD Variable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GOODBAD | Frequency | Percent | Cumulative Frequency | Cumulative Percent |
| 0 | 1034829 | 82.43 | 1034829 | 82.43 |
| 1 | 220600 | 17.57 | 1255429 | 100.00 |

One can see that 17.57% of the observations are labeled "bad." A DELQID of 3 corresponds to a customer being 60 days late on payment. This is chosen as the cutoff point between a "good" or "bad" customer because credit card companies can make large profits on customers who are continually only 1 cycle late on payment. Once a customer is two or more cycles late, they generally tend to fall further and further behind on payments. One implication of assigning the DELQID variable to the maximum value of a customer's DELQID during their performance period is that the type II error may increase. Customers who may have gone 3 cycles late but caught up on their payments and finished the performance period paying on time are still assigned a "1" and considered a bad customer. There will be customers who are predicted to default and denied a credit card, even though they will pay off their accounts. The alternative would be extending credit cards to too many customers, and losing money on the ones who default, which would result in a type I error. The safer route will be taken, to try to reduce the probability of giving a credit card to a customer that will default.

During the merge of the CPR and PERF datasets, some of the observations between the two datasets did not match. There were 17,230 observations without DELQID. These are the number of customers who are in the CPR file, but not the PERF file. Since these customers will not have the binary indicator "GOODBAD", these observations are deleted from the dataset. There were also 470,846 customers without an age. These are the number of customers who are in the PERF file, but not the CPR file. These customers have none of the predictors that will be used to assess whether the customer is good or bad, so these observations are also deleted. All the performance variables except for DELQID and credit limit are dropped for the cleaning process

since these variables should not be changed. The dataset now contains 1,255,429 observations and 341 variables.

**Variable Reduction**

Of these 341 variables, there are many missing and coded values. Variables whose values end in a 93 or above are coded in one of several ways: invalid past due, invalid high credit, invalid balance, etc. For example, a variable like age, whose values only span two-digit numbers, have coded values for 93-99. A variable such as CRDPTH (the age of the oldest trade), which takes up to four-digit numbers, will have coded values for 9993-9999. These coded values must be replaced with meaningful values since the coded values greatly skew the data. The two main methods of imputation which were considered were replacing these values with the median or mean of the variable. Due to the size and volume of the variables and data, more advanced techniques such as a regressed imputation were not chosen. Many of the variables in the dataset have a histogram which is skewed, such as the example of the PDEROG (Number of Public/Rerecord Items) variable in Figure 1 below.

Figure 1. Histogram of PRDEROG Variable

Chart

Description automatically generated

Because of the skewed graph, and number of coded values, the mean and the median shown in Table 2 are very different for the PDEROG variable. The median is 2 more times different and is representative of the actual data values for customers, while the mean is almost 2 times greater than the median because the coded values greatly affect the mean. For the variables with a significant portion of coded values or with highly skewed distributions, the means will be much higher than the medians. Because of this, the imputation method chosen in this project will be to use the median, as it is more representative of the data. Table 3 shows the new descriptive statistics for the PDEROG variable after imputations with the median. One can see that the mean is much lower post imputation. Note that the median which is used for imputation is the median of the non-coded data (the overall median was 4 with the coded values, but when ignoring these coded values, the median is only 2 and this is used for imputed values).

Table 2. Descriptive Statistics of PDEROG Variable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Minimum | Maximum | Median | Mean | Number Non-Coded |
| 1 | 98 | 1.0 | 1.6249656 | 556577 |

Table 3. Descriptive Statistics of PDEROG Variable after Imputation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Minimum | Maximum | Median | Mean | Number |
| 1 | 92.0 | 1.0 | 1.6188618 | 1255429 |

A macro is used to go through each of the variables to decide which will remain in the dataset. One of the inputs in the macro is the limit for the percentage of coded values within each variable. For example, if the limit is set to .2, then any variable with 20% or more coded values will be dropped from the dataset. As the limit decreases, fewer and fewer variables will remain, as the cutoff is becoming more restrictive. Figure 2 below shows the number of variables remaining when the limit is varied between .1 to .7.

Figure 2. Graph Depicting Limits for Percent Missing Used in Macro

Chart, line chart

Description automatically generated

One can see that there is not a great change in the number of variables remaining when the limit is set between .1 to .3. The same can be seen once the limit exceeds .6, as the graph is leveling off. Between the .4 and .5 limits, the .4 limit is chosen for convenience as this yields roughly half the variables (164) that were in the original dataset of 341, whereas .5 yields 237 values. While 237 variables is still a significant difference from 341, using .4 would ensure more accurate results. One other reason for choosing .4 over .5 is that this implies that each variable remaining in the dataset had at least 60% non-coded values. In other words, any variable that has more than 40% coded values is dropped from the dataset.

The other input in the macro was a number which determined how many standard deviations an observation was allowed to be from the mean before imputing the value. For example, if the maximum was allowed to be four, then any value within four standard deviations of the mean remained the same, but a value greater than four standard deviations from the mean was imputed to the median. The purpose of this portion of the macro is to avoid outliers in the dataset as the selected variables move to the modeling phase. Note that changing this value had no impact on how many variables would remain, so the limit of .4 (from the previous section) was kept constant as it was desired to retain 164 variables. Several values were tried for the maximum standard deviations "MSTD" value to see the effect on the remaining 164 variables. The chart below in Table 4 shows the effects of changing the value of MSTD.

Table 4. Chart Showing effects of Changing MSTD in Macro

|  |  |
| --- | --- |
| Change in MSTD | Number of Variables Affected |
| 4 to 5 | 124 |
| 4 to 6 | 143 |
| 5 to 6 | 130 |

For example, when the value for MSTD was changed from four to five, the maximum value for 124 of the 164 variables was affected. Since most of the variables are very right skewed, changing MSTD will only affect values greater than the mean, so it is desired to look at how the maximum values of each variable are affected. Table 5 below shows how some of the maximum values for individual variables are affected when different values of MSTD are chosen (max4 shows the maximum value for the variable when MSTD=4, max 5 shows the maximum value for the variable when MSTD=5, and max6 shows the maximum value for the variable when MSTD=6).

Table 5. How Variables Are Affected by Changing MSTD Value

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | max4 | max5 | max6 |
| COLLS | 7 | 9 | 11 |
| LOCINQS | 12 | 14 | 17 |
| INQAGE | 24 | 24 | 24 |
| FININQS | 13 | 16 | 18 |
| BKP | 1 | 1 | 1 |
| BKPOP | 1 | 1 | 1 |
| PRIMNQS | 67 | 80 | 91 |
| BNKINGS | 27 | 32 | 37 |

In the table above, one can see that some variables such as age, INQAGE, BKP, BKPOP are not affected by changing MSTD from four to six. Note that BKP and BKPOP are binary variables, so they are unaffected since there is not enough variability in the values of these variables. Other variables, however, such as COLLS, have different maximum values depending on which value of MSTD is chosen. As MSTD is increased, fewer extreme values are imputed, and more of the original values are left the same. One concern in leaving MSTD too low is that we are replacing too many values with the median. By leaving MSTD too high, there is a possibility that the variables contain many outliers which will make it more difficult to create a model with these variables. To explore this further, Figure 3 and Figure 4 below show the histograms of two variables, DCAGE and CRATE79, respectively, for different values of MSTD.

Figure 3. Histograms of DCAGE Variable at Different MSTD Values

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generated

MSTD= 4 (top left), MSTD=5 (top right), MSTD = 6(bottom)

In the figure above, we can see that the maximum value in the DCAGE variable (age of oldest/dept store acct) is increasing for increasing values of MSTD. This is again what is expected as the greater MSTD becomes, the greater the requirement for the value to get imputed. One can see from these graphs that even as MSTD increases, it is only affecting a small portion of the values. In the top left histogram of Figure 3, the cutoff point for DCAGE is 485 when MSTD is set to 4. In the top right histogram of Figure 3, the cutoff point for DCAGE is 579 when MSTD is set to 5. This is a fairly large increase in the maximum, but there are very few values which fall in this interval, as indicated by the low density in the right tail of the graph. Figure 4 below shows a similar trend for the CRATE79 variable (the number of accounts/currently bad debt), which has much less variability in its values. When MSTD is set to 4, the maximum CRATE79 value is 15, and when MSTD is set to 5, the maximum CRATE79 value is 18. Again, however, there is very low frequency among these values that are cut off when increasing the value of MSTD. Only two variables are explained in detail with graphics, but a similar trend is seen for most of the variables. Since increasing the value for MSTD affects only a small percentage of the values, the value of four will be chosen for MSTD. For modeling purposes, this should help get rid of some of the outliers in the data, by replacing them with the median.

Figure 4. Histograms of CRATE79 Variable at Different MSTD Values

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generated

MSTD= 4 (top left), MSTD=5 (top right), MSTD = 6(bottom)

**Variable Clustering**

The next step in the process is to further reduce the number of variables. This will be done by determining the VIF of each variable and deleting any variables with a VIF of greater than 10. There are currently 165 variables in the dataset. There is one variable in our dataset, beacon, which contains all missing values. This value was not picked up with the macro since there were no coded values; however, this variable was manually removed since it provides no information. One additional variable, age, will be removed since this variable cannot be used to assess credit. 163 variables will enter the VIF stage.

VIF's are variance inflation factors and signify when there may be multicollinearity among variables. Generally, there is cause for concern when the VIF is greater than 10. When calculating VIF's, each variable is run in a regression as a linear combination of the other variables. The resulting R2 of this regression is used in calculating VIF's (1/(1-R2)). A VIF greater than 10 implies that the R2 of the regression was greater than .9, signifying that the variable was a linear combination of other variables in the model. In looking at the VIF's from the regression model with the 47 predictors, none of the variables have a VIF greater than 10.

Table 6 below shows the variables with the greatest VIF's. Generally, two variables that are correlated will both display large VIF's. In Figure 6, the correlations between the three variables with largest VIF's are shown. The variables OT3PTOT and MOSOPEN are slightly correlated (correlation = .66040), which could explain why these variables exhibit the largest VIF's. BRADB may not be as correlated because it may be more correlated with other variables in this list. The final dataset to take into the discretization process now contains 44 variables.

Table 6. Variables and Corresponding VIF's

Table

Description automatically generated

Figure 5. Correlations for Variables with Largest VIF's

Table

Description automatically generated

After this, they can be clustered. Now there are 48 observations, of which 4 variables are DELQID, GOODBAD, CRELIM and MATCHKEY. These variables will be removed from the dataset for the VIF portion since these are the target variables and unique identifier for each observation, respectively. With the remaining 44 variables, the clustering can begin. This will be done by clustering the variables into groups to see which variables are most similar to one another. Once in clusters, the most representative variable will be chosen from the cluster to be kept in the dataset. To determine how many clusters will be chosen, a graphic in Figure 6 shows the percent variation explained for different numbers of clusters. In the graph below, one can see that as there are more clusters, the percent variation explained increases. When each variable is assigned its own cluster, 100% of the variation is explained. Once there are clusters with two or more variables, some information is lost. Figure 7 shows a magnified version of Figure 6. In this graph, one can see a steady increase in the percent variation explained as the number of clusters increase.

With around 26 clusters, the graph begins to level off, as the percent explained is 86.51%. 26 clusters will be chosen as 86.51% of variation explained is suitable to again reduce the number of starting variables (159) to less than a quarter.

Figure 6. Graph of Proportion Variation Explained at Different Numbers of Clusters

Chart, line chart

Description automatically generated

Figure 7. Magnified Graph of Figure 6 Chart, line chart

Description automatically generated

Once the clusters are created, the variable with the lowest (1-R2) ratio is chosen. This ratio is a function of both the R2 of a variable with its own cluster (the higher the R2 the more collinear this variable is with its cluster) and the R2 of the variable with the next cluster. The table in Table 6 below shows the makeup of some of the clusters. Again, the variable which has the lowest ratio within the group is chosen from the group to remain in the model. For example, from cluster 1, the variable OT12PTOT is chosen.

This will be done by clustering the variables into groups to see which variables are most similar to one another. Once in clusters, the most representative variable will be chosen from the cluster to be kept in the dataset. To determine how many clusters will be chosen, a graphic in Figure 5 shows the percent variation explained for different numbers of clusters.

Table 6. Several Clusters with Variables and Their (1-R2) Ratios

Table

Description automatically generated

In looking at a correlation table below in Figure 7, it is apparent that variables within the same cluster are highly correlated. OT24PTOT, OT12PTOT, OT3PTOT, and TPOPEN were all in cluster 1 (as seen in Table 6 above). The correlations for these variables among one another are all .46 or above. The variable CPAF29 is also included in Figure 7 to show that a variable from cluster 2 has a very weak correlation to these variables in cluster 1 (apparent by the correlations ranging between 0.16 to .07).

Figure 8. Correlations Between Select Variables in Clusters 1 and 2

Table

Description automatically generated

Once the variables are selected from the clusters, the dataset has 65 predictor variables (equal to the number of clusters). In addition to these variables, the DELQID, GOODBAD, CRELIM and MATCHKEY variables are added back into the dataset to bring the total to 69 variables. As one final check on the remaining variables, a linear regression is run to inspect the VIF's among the variables. VIF's are variance inflation factors and signify when there may be multicollinearity among variables. Generally, there is cause for concern when the VIF is greater than 10. When calculating VIF's, each variable is run in a regression as a linear combination of the other variables. The resulting R2 of this regression is used in calculating VIF's (1/(1-R2)). A VIF greater than 10 implies that the R2 of the regression was greater than .9, signifying that the variable was a linear combination of other variables in the model. In looking at the VIF's from the regression model with the 26 predictors, none of the variables have a VIF greater than 10.

Table 7 below shows the variables with the greatest VIF's. Generally, two variables that are correlated will both display large VIF's. In Figure 8, the correlations between the three variables with largest VIF's are shown. It can be seen that the variables TOPEN and BRTRADES are slightly correlated (correlation = .64985). This can explain why these variables exhibit the largest VIF's. Since a VIF of 10 was chosen as a cutoff point, no variables will be removed. The final dataset to take into the discretization process now contains 44 variables.

Table 7. Variables and Corresponding VIF's

Table

Description automatically generated

Figure 9. Correlations for Variables with Largest VIF's

Table

Description automatically generated

**Variable Preparation**

In the next step of the project, the variables undergo a transformation process. Each variable will be transformed using two methods (discretization process 1 and discretization process 2) to create an ordinal ranking for the values of each variable. Discretization process 1 was done first, but for clarity reasons, discretization process 2 will be discussed first.

Discretization process 2 is done using the rank procedure in SAS. The rank procedure will attempt to bin each variable into 10 (as chosen by the user) bins of equal size. In other words, each bin should theoretically have 10% of the values. This binning process will be explained in further detail in the next section.

Discretization process 1 is done in a manual fashion, in which the user creates the bins in a more logical manner. For example, a variable such as age may be binned in intervals of 10 years (ages 15-24, ages 25-34, ages 35-44, etc.). The bins don't necessarily have to be of equal frequency in this process, but it is more logical to have bins of equal interval widths excluding the first and last bins which were subject to change. Once the bins are created in each of the processes, the average default rate is calculated within each of the bins.

This default rate is then used to calculate the odds ratio within each bin. The odds ratio is calculated by dividing the default rate by the quantity (1 minus the default rate). Take for example a bin which has a default rate of .2. The odds ratio for this bin would be calculated by dividing .2 by .8, which is .25. In other words, a customer in this bin is 4 times more likely to not default than to default. The last variable that will be created is the log of the odds ratio. A log transformation is used since a logistic function will be used to model the data. In all, 21 of the 26 original predictor variables will undergo 2 discretization processes, in which each process bins the variable and calculates the odds and log of the odds. The reason 5 of the variables did not go under the two processes is because they were already discretized. Each of the 21 variables will have up to 6 additional transformations of the variable created. In the next section, these binning processes will be discussed in further detail.

In the binning process for discretization process 2, each variable should be split into bins of equal frequency. Due to the discrete nature of many of the variables, this is not actually possible for every variable, and since the variables have already been discretized in discretization process 1, many ranks may remain unrepresented as they have already been collapsed. Table 8 shows the original distribution of the BRWCRATE variable (bank review/worst current rating), after going through the first discretization process. Very few of the values are rank 0, which are all binned into bin 1. The remaining values are then binned together into bin 2, 3, and 4 (named here 7, 8, and 9, respectively). At the 90th percentile, the rank procedure attempts to create a bin. In the case for this variable, the value of "0" ended in the first percentile (cumulative percent is 0 in Table 8), so it was binned with rank 7 (as seen in Table 9). When using the rank procedure, many discrete variables (with distributions similar to the BRWCRATE variable) were transformed into binary variables after the binning process.

Figure 10: Original Distribution of BRWCRATE Variable before Discretization 1

Graphical user interface

Description automatically generated

Table 8. Distribution of BRWCRATE Variable before Binning with Discretization Process 2

Table

Description automatically generated

Table 9. Distribution of BRWCRATE Variable after Binning

Table

Description automatically generated

One issue that arose for some variables is depicted in Table 10 below for the variable LAAGE (age of last activity). The distribution lies between two ranks, and therefore it was not discretized. This occurred with five variables, and they did not go through either discretization processes. Because these variables cannot be ranked, there is no need to put them through the second process.

Table 11 depicts the probability of default for LAAGE variable. The variables with a similar distribution to LAAGE typically showed the same probability of defaults, ranging from .16 to .18.

Table 10. Distribution of LAAGE variable

Table

Description automatically generated

Table 11. Probability of Default of LAAGE variable

A picture containing background pattern

Description automatically generated

One other thing which the discretization process 2 considers is the probability of defaults within each of the ranks after doing the rank procedure. For example, in Table 12, the probability for default differs little between ranks 1 and 2, and 6 and 7 (as shown in bold). As part of the SAS code in the macro which runs the rank procedure, a t-test is performed to detect a significant difference in the probability of default between consecutive ranks. For the TROPENEX variable (number of open trades included/closed narratives), there was no statistically significant difference between these ranks, so they are combined into bin 2 and bin 7 as seen in bold in Table 13. Originally there were 10 ranks, and once ranks 1, 2, 6, and 7 collapse, there are 8 bins which remain.

Table 12. Distribution of TROPENEX Variable with Discretization Process 2

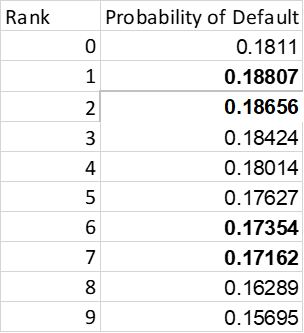


Table 13. Distribution of TROPENEX Variable after Ranks Collapse

Table

Description automatically generated

When looking at the TROPENEX variable and the probability of default within each bin, there is not a great deal of separation between the bins. Figure 11 shows a graphical representation of Table 13. This variable is likely not an overall great predictor for a customer defaulting because of two main reasons. The first reason, as stated above, is that the overall spread between the lowest probability of default in bin 9 and the highest probability of default in bin 2 is only 18.7%. The average probability of default is again .1752, and the numbers in Table 13 are close to this value. The second reason is that it would be ideal to see a more linear trend in one direction than what is seen in Figure 11. The probability of default decreases from bin 8 to bin 9 and increases slightly in bins 0 and 2. It will be difficult to use a variable which has a bouncing pattern similar to the one seen in Figure 11 since this variable does not have a definitive trend or relationship among the bins and probability of default.

Figure 11. Probability of Default by Bin in TROPENEX Variable

Chart, line chart

Description automatically generated

In discretization process 1, there was more flexibility in defining the bins. The PRMINQ2 variable (number of promissory inquiries in the past six months) is looked at in further detail below. Figure 10 shows the original distribution of the PRMINQ2 variable. One can see that the distribution is heavily skewed to the right. It was decided to bin this variable in intervals of 5, and every value of PRMINQ2 less than 5 was grouped into the first bin and any value greater than 20 was grouped in the last bin since these values had low frequency counts compared to other bins.

Figure 12. Distribution of PRMINQ2 VariableGraphical user interface, chart

Description automatically generated

Table 14 below shows the results of binning the PRMINQ2 variable as described above. The associated probabilities of default are given for each bin. The probability of default barely differs in bins 2-5. Bin 1 will be further investigated to find a good cutoff point for the bins.

Table 14. Binning of PRMINQ2 in Intervals of 5

|  |  |  |
| --- | --- | --- |
| Bin | Total in Bin | Probability of Default |
| 1 | 215752 | 0.23673 |
| 2 | 346720 | 0.19145 |
| 3 | 319313 | 0.16109 |
| 4 | 189612 | 0.14162 |
| 5 | 184032 | 0.13506 |

Bin 1 from Table 14 contained values of PRMINQ2 from 0 to 5 with a default rate close to 24%. In looking at the remaining values, it was noticed that the default rate for customers in the 2-5 range for PRMINQ2 had a default rate of average about 15%, as was seen in the rest of the bins. Due to this, in discretization process 2, the bins were split into more bins separated from 1-10 evenly grouped by 1. Table 15 shows the new bins, along with the probability of default within the bins. There was not much separation among the customers with their number of promissory inquiries in the past six months. Thus, customers had roughly the same probability of default but decreasing rates as the number of promissory inquiries in the past six months increases. Figure 13 shows the binning process for this variable when discretization process 2 was used.

Table 15. New Binning of PRMINQ2

|  |  |  |
| --- | --- | --- |
| Bin | Number of Observations | Probability of Default |
| 1 | 111221 | 0.24662 |
| 2 | 104531 | 0.22620 |
| 3 | 129017 | 0.20431 |
| 4 | 144553 | 0.18674 |
| 5 | 73150 | 0.17809 |
| 6 | 140460 | 0.16863 |
| 7 | 124390 | 0.15708 |
| 8 | 145856 | 0.14826 |
| 9 | 142304 | 0.13406 |
| 10 | 139935 | 0.13733 |

Figure 13. Probability of Default by Bin in PRMINQ2 Variable for Discretization Process 2

Chart, line chart

Description automatically generated

Another example comparing a variable and the differences between discretization processes 1 and 2 is shown with the FININQS variable (number of financial inquiries in last 24 months). In discretization process 2, six ranks were created seen in Table 16 as defined by the macro procedure. In discretization process 1 shown in Table 17, the original values were further binned into three bins. The associated probabilities of default are in Table 16 and Table 17.

Table 16. Binning of FININQS Variable in Discretization Process 2

Table

Description automatically generated

Table 17. Binning of FININQS Variable in Discretization Process 1

|  |  |
| --- | --- |
| Value | Probability of Default |
| 1 | 0.16270 |
| 2 | 0.18172 |
| 3 | 0.17898 |

The difference between the two processes is the number of bins. Discretization 1 was decided to have three bins, whereas discretization process 2 split the ranks into six groups having rank 1 grouped with rank 2 and rank 7 and 8 grouped with rank 6. The trend of customers defaulting is shown to be average as the ones who are more likely to default are the middle ranks from 4-6 according to the discretization process 2. In Table 17 we can see that the binning makes the FININQS variable average across the bins which doesn’t show the more details as seen in Table 16. From Table 16 we can see that rank 6 has the highest probability of default with 0.3538, but this nor specific ranking can be seen in Table 17. As a result, some information is lost when the values are collapsed into three bins in discretization process 1 which lessens the variable being a good predictor.

Discretization process 2 is easier to use since it is a more automated process, but discretization process 1 is also useful as it allows the user to manually adjust and make clear relationships which may have been overlooked in the other discretization process.

These discretization techniques are useful in converting the variables into a more understandable ranked form, especially for variables which weren't already highly discrete (only had ~10 or less possible values to begin with). In all, there are 156 variables in the dataset at this point (including GOODBAD, MATCHKEY, CRELIM and DELQID). The next step will be to create a logistic regression and decide which predictors remain in the final model.

**Logistic Regression**

All the predictor variables are put into a logistic regression with the GOODBAD variable as the response variable. A stepwise selection approach to this regression was not performed due to the sheer size of the dataset. Instead, a backward selection process was performed to ensure that every variable was in the model in the beginning, and then removed based off a cutoff point at the .05 level of significance. The least significant variable is removed in each step until all remaining variables are significant. Some output from the logistic regression is shown below in Table 18.

Table 18. Top 3 Variables in Final Model after Backward Selection Logistic Regression

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Parameter | DF | Estimate | Standard  Error | Wald Chi-Square | Pr >  ChiSq | VIF |
| INTERCEPT | 1 | -0.9202 | 0.0163 | 3180.7441 | <.0001 | 0 |
| LODSEQBRADB | 1 | 0.8382 | 0.00325 | 66601.2260 | <.0001 | 1.31485 |
| ORDBRWCRATE | 1 | 0.9232 | 0.00715 | 16670.9451 | <.0001 | 1.11547 |
| LODSEQTPCTSTAT | 1 | 0.3936 | 0.00536 | 5392.7055 | <.0001 | 1.33135 |

The coefficients in the model are interpretable just like in a regular regression model. For instance, the LODSEQBRADB variable is the log odds of the BRADB (average debt burden/bank revenue accounts) variable. The coefficient of LODSEQBRADB is found by taking the log of the odds of default from the BRADB variable (e^x = 0.8382: where x is the coefficient for the BRADB variable); this resulting numbers is the odds ratio. For every one unit increase in the BRADB variable (for each debt burden/bank revenue account the customer has), a customer is 0.8382 times more likely to default.

Table 18 only contains the top three variables from the model. It was desired to keep the top three variables from the model for the purposes of having a simple model. The variables are sorted in order of the largest chi-square statistic, which is representative of the most significant variables in the model. Before getting to the final model of three variables, multicollinearity was addressed as seen with VIF analysis below.

In Table 19, The top 17 variables are shown since the variable BRWCRATE (bank review/worst current rating) appears twice among the top 18, both in its raw form and one of the transformed forms of the variable. When inspecting the VIF's for the top 18 variables in Table 19, it is evident that the BRWCRATE variable and its transformation (ORDEQBRWCRATE) are redundant. When removing the BRWCRATE variable (seen in Table 20), the VIF's are now almost all much less than 10, and there is no problem for multicollinearity. It was decided to remove the BRWCRATE variable since the original form of the variable had the lower chi-square statistic. This process was continued until we got to the final model of three variables with all VIF less than 10 as seen in Table 18 thus having no multicollinearity in the final model.

Table 19. VIF's with Top 18 Variables Table 20. VIF's with Top 17 Variables



The full model which was created from the backward selection process contained over 70 variables. Again, the model which will be chosen as the reduced model will only contain the top three variables from the full model. One of the statistics which is used in assessing performance of this model is the c statistic. This statistic represents the percentage of concordant pairs in the model. To calculate this, the dataset is split in two datasets: one which contains the non-defaults (GOODBAD = 0) and the other which contains the defaults (GOODBAD = 1). The predicted probabilities of defaults are calculated, based off the logistic regression, for each observation in both datasets. Each observation is now paired with every observation in the other dataset, and a concordant pair corresponds to a pair of observations in which the predicted probability of default, for an observation with GOODBAD=0, is less than the predicted probability of default for an observation with GOODBAD=1. That is, ideally the model should predict lower probabilities of default for customers who did not default, than for customers that do default, and this is a metric to assess how well the model is doing this. Table 21 shows the percent concordant in the full model. Table 22 shows the percent concordant in the reduced model. The value of 83.5 in the full model is a satisfactory percent concordant for the full model. More importantly, by only choosing 3 variables from the full model, the percent concordant does not decrease greatly (80.0%).

Table 21. Percent Concordant in Full Model

|  |  |
| --- | --- |
| Percent Concordant | 83.5 |
| Percent Discordant | 16.5 |
| Percent Tied | 0.0 |
| Pairs | 102476447472 |

Table 22. Percent Concordant in Reduced Model

|  |  |
| --- | --- |
| Percent Concordant | 80.0 |
| Percent Discordant | 19.5 |
| Percent Tied | 0.6 |
| Pairs | 102476447472 |

The other main metric used to assess model performance is the KS statistic. To calculate the KS statistic, the dataset is split into ten bins of equal frequency (deciles). The score, which is 1000\*P(default), is used to determine the cutoff points for each bin and are sorted from highest to lowest bins in Table 23. Within each bin, the percentage of the overall good and bad customers is calculated. The cumulative percentage of good and bad customers are then calculated across the deciles. The KS statistic is the maximum separation between the cumulative percent of the good and bad customers. In the table below, the maximal KS of 44.72 occurs in the 6th decile.

Among customers in the top 6 deciles, the model can identify 77% of all the defaulting customers, and 32% of the non-defaulting customers. The larger the KS, the better the model is at discerning good from bad customers, and ideally the maximal KS should occur within the first few deciles as this means that the model can easily discern between customers in the highest score ranges. The lift is also calculated, which is how much more likely the model is to predict a default within any of the deciles. For example, in the first decile, the model is 3.4 times more likely to predict a default than it would without a model. A graphical representation of the KS is shown in Figure 12. Graphically, the KS is the maximum difference between the solid and dashed lines of the graph, which occurs between the 2nd and 3rd deciles.

Table 23. KS Calculations



Figure 12. KS Chart

**Cost of Simplicity**

The three variable model which has been chosen contains none of the variables in their raw form. Two of the variables retained are from discretization process 2 and the last variable (ORDBRWCRATE) was created from discretization process 1. A model will be created with the same 3 variables, but the 3 variables which were created during the discretization processes will be used in their raw form. This model will represent the best 3 predictors without any transformations performed on them. The percent concordant of the model is shown in Table 24 below. The percent concordant is lower than that of the full and original 3 variable model by nearly 10%. Also, the c-statistic is less than 0.80 at 0.722 which shows it is not a good model. The average profit is calculated in a similar way as before, and the hit chart is shown in Table 25. The average profit per customer in the simple model is $86.95, and the profit per 1,000 customers scored is $86,900.42.

Table 24. Percent Concordant in Simple Model

|  |  |
| --- | --- |
| Percent Concordant | 72.2 |
| Percent Discordant | 27.8 |
| Percent Tied | 0.0 |
| Pairs | 102476447472 |

Table 25. Hit Chart for Simple Model

|  |  |  |  |
| --- | --- | --- | --- |
| Percent Valid1 | Percent Valid2 | Percent Type I Error | Percent Type II Error |
| 10.68 | 60.52 | 6.90 | 21.91 |

**Conclusion**

The first model which was created contained all the significant predictors from the backward selection process and had a concordant percentage of 83.5. By choosing only the top 3 variables from this model, the concordant percentage remained high and dropped only slightly to 80.0. The average profit per 1,000 customers scored in this three variable model was $105,750.33. An even less complex model was created to eliminate the variables which were created in the manual discretization processes. Instead of using these transformed variables, the same 10 variables were only used in their raw, original forms. In this model, the concordant percentage was a bit lower at 72.2, and the profit per 1,000 customers was $86,900.42. This is interesting since it is much easier to leave the variables in their original form than to explain the transformations to a client. For a difference in profit of just over $19,000 (or $19.00 per individual customer scored), it is actually more beneficial to transform the variables from the original state in order to reap more profit. Table 26 shows these comparisons between the models.

Table 26. Comparison of Final Models

|  |  |  |
| --- | --- | --- |
|  | Reduced Model | Simplicity Model |
| Percent Concordant | 80.0 | 72.2 |
| Profit Per 1000 Customers | $105,750.33 | $86,900.42 |

Overall looking at the three different models (final model, reduced model and simplicity model), the one that would be the most key to use in the reduced model. Although the variables are in the transformed state it gains $19,000 more dollars in the long run. Also compared to the Final model, the reduced is more efficient as even though is returns about $6,000 less in profit, since the final model has over 70 variables including original and transformed, companies would have to spend way more money to be able to receive the necessities from the customers to even run the model which would lose a lot of profit.