Michael Verwijst
3/27/2018

Optimal Quote Pricing at a Major Steel Distributor

Business Understanding

The CRU Group is the leading authority for the world of metals and mining. The price indices that the CRU Group generates establish benchmarks on which buyers and sellers can confidently base transactions. CRU's US Midwest Hot Rolled Coil Price Index is the leading steel benchmark in North America and is continuously used by steel mills and steel distributors to price steel products appropriately and fairly.

As a major distributor of steel products in the US, the success of the company relies heavily on The US Midwest Hot Rolled Coil Price Index (referred to from here on out as the **CRU**). The price that steel products are purchased and sold at is directly related to the CRU price, and thus has a major impact on sales dollars, cost of goods sold, and profitability of the company.

In the realm of steel distributors, a quote to a customer for steel products often competes heavily on price. Since the lowest quoted price frequently wins in a multi-vendor scenario, there is a crucial need to price competitively. The CRU changes daily and can vary significantly, so quoting competitively and aggressively can be a challenge. While winning a quote is crucial to generate revenue, it is more important for the Sales Team to maximize profitability, which adds another layer of difficulty in the quoting process.

Project Objectives

The objectives of this project are to:
- Identify the variables that hold predictive weight in determining if a quote will be won or lost
- Experiment with various tools to learn and get a better understanding of the many predictive modeling techniques available
- Utilize Feature Engineering techniques to improve the predictive performance of the selected model
- Achieve accuracy of greater than 80%, as required by the business in order to feel confident in deploying a predictive model into production

Expected Outcomes

The expected outcomes of this project are to:
- Provide the Sales Team a better understanding of what is impacting the quote win rate
- Highlight areas of focus in the quoting process to win more quotes and capture more revenue
- Build a pricing tool that the Sales Team can lean on to select the optimal price for a steel product at any given time (and current CRU price)

Data Understanding

The original data set for this project includes altered Quote data for the main branch of a major Steel Distributor for the years 2017 and 2018. The dataset was scrubbed of sensitive information and modified to obscure confidential operations of the business.

Structure – 61,548 Records … 18 Columns:

- QuoteDet: Quote Reference
- QuoteDate: Date the Quote was sent
- SalesOrdNbr: If Quote won=Sales Order Number… if lost=left blank (**altered**)
- TonGroup: Tonnage Group (product group) of quoted item
- LineCode: Delivery Type quoted
- ProdNbr: Product Code Reference
- PriceExt: Total Price on Quote (**altered**)
- MarginExt: Total Margin on Quote (**altered**)
- Gp%: Gross Profit (Margin/Sales) of quote (**altered**)
- FrtExt: Freight Cost added to Quote
- WgtExt: Total Weight Quoted
- CutDesc: If/How unit will need to be cut
- OurCost: Cost of individual unit quoted
- BasePrice: Price of individual unit quoted
- UnitWgt: Weight of individual Unit quoted
- CustPn: Customer PartNbr Reference
- LowMargin: Indicates if Quote is flagged as low margin
- CustNbr: Customer Reference Number (**altered**)

This same data had been used in two previous projects with similar goals in mind, but the outcomes had unsatisfactory results. The first project attempted to provide an analysis of the average price of winning quotes compared to the average price of losing quotes as it related to different weight groups. The results were inconclusive as they often showed losing quotes having a lower average price than wining quotes, which was inadequate as defined by the business. Because of the ever-changing CRU price, the average price comparison project did not work as it required only using a few weeks of data to compare prices effectively. Additionally, the deployment method used was too slow, and was not usable.

The second project was a classification model using logistic regression to predict if a quote would be won or lost. The initial results of that project produced accuracy and recall rates of greater than 85%. However, fields were later discovered in the data that signified a quote was never actually sent and that a quote was simply entered to produce a sales order. Both of these fields needed to be eliminated in order to produce meaningful prediction results. Doing so significantly reduced the classification results to a recall and accuracy rate of around 65%, a level that was unacceptable by the business. Both projects were deemed unsuccessful and abandoned.

Data Preparation

The original data set has many categorical variables, which were converted into binary features (1 or 0). Most notably this includes the target variable (Quote_Won), if a part was cut or not, if freight was charged to the customer, if the part had a customer part number reference, and if the quote was flagged as low margin.

The business identified weight groups in the first project, as described in Data Understanding, which would allow for the comparison of quote pricing at different weight groups. This project uses those same weight groups, and the data is binned using the same logic. For example, the total weight quoted is binned into meaningful package sizes (<100 lbs, <500, <1k, <5k, <10k, <40k, and >40k) since these increments change how an order would be shipped. Additionally, Gross Profit% is binned based on specific target ranges that the sales team tracks.

There are a number of new variables that were added to enrich the data set. Throughout this project and previous project modeling iterations of this data set, many new variables were tested to improve the accuracy of the prediction model. 'Previous Customer Incumbency' and the 'Historic Win Rate' of a customer over the past two years were the variables with the largest impact on prediction results by far, and thus are used in this project. Doing so requires the addition of historic quote data for all branches and tonnage groups across the organization to be summed and joined in to the original data set.

In order to address the issue of the CRU price constantly fluctuating, this project introduces the concept of CWT:CRU Ratios. CWT:CRU Ratios calculate the Price per 100 pounds (CWT) and divides it by the CRU price at the time of the quote. Doing so requires bringing in historic CRU Prices. For example, a quote with a Price per CWT of $100 while the CRU price was at $350, has a CWT:CRU Ratio of .286; while a quote with a Price per CWT of $200 when the CRU price was at $700, also has a CWT:CRU Ratio of .286. This concept reflects that, while the actual quoted price is $100 different, it is effectively the same price as it relates to the CRU and can be compared as such. This is a pivotal aspect of this project, as it removes the variability of the CRU and allows for historic quotes to be compared to one another on a level playing field.

In order to produce effective Ratio Groups and price ranges, it was discovered that the data need to be analyzed at each specific Tonnage Group rather than in total. There is a significant price variation for products in different Tonnage Groups, so they need to be split and analyzed separately. For the remainder of this analysis, this project will look only at Tonnage Group 070 for a single branch, which includes 6,126 records over 2 years of history.

Modeling

Initial modeling for this project was done using Salford Predictive Modeler. The results from the TreeNet and Logistic Regression models indicated that the statistically relevant variables (p-values < .05) holding the most predictive weight were the historic 'Customer Win Ratio', 'CWT:CRU Ratio', and the Weight Groups. The more complex models, like TreeNets, did produce better results; but the output shown below using logistic regression was not far behind. For the sake of simplicity and re-creating the model in R later, it was decided to use logistic regression for the remainder of this project.

Model Summary

Model error measures

Name	Learn
LogLikelihood	-2,829.56277
LogLikelihood (constant model)	-3,650.56671
Average LogLikelihood (Negative)	0.46197
Chi-Sq P-Value	0.00000
ROC (Area Under Curve)	0.81244
Variance of ROC (Area Under Curve)	0.00004
Lower Confidence Limit ROC	0.80071
Upper Confidence Limit ROC	0.82416
Lift	2.81268
McFadden's Rho-Squared	0.22490
K-S Stat	0.46062
Misclass Rate Overall (Raw)	0.21012
Balanced Error Rate (Simple Average over classes)	0.27359
Class. Accuracy (Baseline threshold)	0.74922

Model Coefficients

Variable	Coefficients	S.E.	T-Value	P-Value
Constant	-4.51148	0.65996	-6.836	0.00000
WGT_0_100	2.40577	0.6609	3.6402	0.00027
WGT_100_500	2.46291	0.65749	3.746	0.00018
WGT_500_1000	2.30664	0.66065	3.4914	0.00048
WGT_1K_5K	2.10537	0.65959	3.192	0.00141
WGT_5K_10K	1.95950	0.68918	2.8432	0.00447
WGT_10K_40K	1.85058	0.71757	2.5789	0.00991
WGT_40K_	0.11409	0	0	0
CUST_WINRATIO	7.23916	0.22709	31.878	0.00000
CWT_CRU_RATIO	-1.10885	0.21423	-5.1759	0.00000

Actual Class	Total Class	Percent Correct	Predicted Classes 0 N = 3986	Predicted Classes 1 N = 2139
0	4,390	77.90%	**3,420**	970
1	1,735	67.38%	566	**1,169**
Total:	6,125			
Average:		72.64%		
Overall % Correct:		74.92%		
Specificity		77.90%		
Sensitivity/Recall		67.38%		
Precision		54.65%		
F1 statistic		60.35%		

Initial results using the new features discussed in previous sections showed a substantial improvement compared to the analysis in previous projects. The area under the ROC curve was .812, with a Lift of 2.81. The Confusion Matrix showed a Recall of almost 68% and an Average accuracy of nearly 75%. The precision rate was only 55%, which is cause for concern as only 970 of the 2139 Win predictions were actually Quote Wins. The biased nature of the data set is the likely culprit, as only approximately 18% of all quotes were won, while 82% are lost.

A major pitfall of the previous quote analysis projects discussed in the Data Understanding section was that there was not enough data to provide meaningful results. This was due mainly to the CRU price constantly changing and resulting in a model trying to produce results for only a short time frame. In addition to using the CWT:CRU Ratios, this project uses the concepts or Binning and Mirror Data to address this obstacle and to expand the data set.

After calculating the CWT:CRU Ratio for each quote, the concept of binning is used to split the quotes into 6 Ratio Groups with equal number of quotes in each group (See Appendix A for R Code). These groups are arbitrary; they will be re-calculated and different for each tonnage group analyzed. This is simply a method to group quotes at different levels of historic pricing.

A key concept to understand in this analysis is that, a quote is NOT always won or lost based solely on price. However, all else equal, it is fair to assume that a quote won at Ratio Group 3 would have won at the lower priced Ratio Groups 2 and 1 as well. Similarly, a quote lost at Ratio Group 3 would have lost at the higher priced Ratio Groups 4, 5, and 6 as well. By using this logic, Mirror Data are produced to create duplicate records with everything except the CWT:CRU Ratio the same (see R Code below). This use of Mirror Data allows for the significant expansion of data while increasing the predictive power of the CWT:CRU ratio (price). An additional benefit to this method is that it keeps roughly the same percentage split between quotes won and lost, as to avoid skewing results.

Using the Binning and Mirror Data techniques, the 070 tonnage group data set expands from 6,126 records to 19,878 records. This not only provides the logistic regression model more data to analyze, but the techniques used puts more emphasis on the CWT:CRU Ratio, and thus, the quote price. After applying these methods to the data, the new results produced using logistic regression are below:

Model Summary

Model error measures

Name	Learn
LogLikelihood	-6,890.92775
LogLikelihood (constant model)	-11,891.43001
Average LogLikelihood (Negative)	0.34664
Chi-Sq P-Value	0.00000
ROC (Area Under Curve)	0.91155
Variance of ROC (Area Under Curve)	0.00001
Lower Confidence Limit ROC	0.90714
Upper Confidence Limit ROC	0.91596
Lift	3.27615
McFadden's Rho-Squared	0.42051
K-S Stat	0.66580
Misclass Rate Overall (Raw)	0.14312
Balanced Error Rate (Simple Average over classes)	0.16955
Class. Accuracy (Baseline threshold)	0.83334

Actual Class	Total Class	Percent Correct	Predicted Classes 0 N = 12890	Predicted Classes 1 N = 6989
0	14,201	83.72%	**11,889**	2,312
1	5,678	82.37%	1,001	**4,677**
Total:	19,879			
Average:		83.05%		
Overall % Correct:		83.33%		
Specificity		83.72%		
Sensitivity/Recall		82.37%		
Precision		66.92%		
F1 statistic		73.85%		

The results using the Binning and Mirror Data techniques show further improvement compared to the initial analysis. The area under the ROC curve is now .912, with a Lift of 3.276. The Confusion Matrix now shows a Recall of 82% and an Average accuracy of 83%. The precision rate is still only 67%, which is still cause for concern with this model and will need to be understood by the business going forward.

The logistic regression model using the same target and independent variables was reproduced in R Studio, with nearly identical results. See R Code in Appendix A.

Evaluation

Prior to project kick-off, the business defined the project objectives as well as results thresholds. It was determined that, classification model results must achieve greater than 80% accuracy in determining if a quote was won or lost. Previous projects did not meet this requirement, and thus, we're not approved to move past testing.

In the initial modeling phase of this project, as discussed above, the logistic regression model achieved a recall rate of 68% and an accuracy of 75%. Although much improved from previous project results, this still did not meet the business requirements of 80% accuracy.

After exploring Binning and Mirror Data techniques, the classification model now achieves a recall rate of 82% and an average accuracy of 83%. These results do meet the business requirements and have received the stamp of approval from the business to move forward into the Deployment phase of the project.

Deployment

The output of the Logistic Regression Classification model is a probability of each quote being classified as a Win or a Loss. Using R Studio, these probabilities are applied as a new field in the data set along with the Ratio Group each quote (original and mirrored) belongs to. The quote data is then pivoted based on the calculated Ratio Group, and an average win probability calculated, as shown below:

Branch 1 - TonGroup 070			
Row Labels	**Count of Quotes**	**Sum of Win Probability**	**Average Win Probability**
a) 0.0215 - 0.114	2,391	1,250	52.27%
b) 0.114 - 0.1317	2,760	1,195	43.30%
c) 0.1317 - 0.1512	3,166	1,142	36.06%
d) 0.1512 - 0.1821	3,605	1,063	29.47%
e) 0.1822 - 0.251	4,094	885	21.62%
f) 0.251 - 4.0211	3,863	144	3.73%
Grand Total	**19879**	**5678**	**28.56%**

This summary shows a clear separation of the probability of winning a quote at each Ratio Group, or price range. Logically, this makes sense as higher relative prices would generally result in less chance of winning the quote. Taking this summary view a step further, the data can be pivoted on the Weight Group that each quote belongs to as well (see Appendix B for full output).

Wgt Group	Ratio Group	Count of Quote	Sum of Quote_Won	Sum of Win_Probability	**Avg Win Probability**
a. 0-100	a) 0.0215 - 0.114	376	368	321	**85.48%**
a. 0-100	b) 0.114 - 0.1317	403	363	326	**81.00%**
a. 0-100	c) 0.1317 - 0.1512	430	345	326	**75.75%**
a. 0-100	d) 0.1512 - 0.1821	513	306	339	**66.01%**
a. 0-100	e) 0.1822 - 0.251	804	250	376	**46.79%**
a. 0-100	f) 0.251 - 4.0211	891	158	102	**11.40%**
b. 100-500	a) 0.0215 - 0.114	799	744	561	**70.26%**
b. 100-500	b) 0.114 - 0.1317	894	677	547	**61.21%**
b. 100-500	c) 0.1317 - 0.1512	1,104	540	543	**49.23%**
b. 100-500	d) 0.1512 - 0.1821	1,401	368	521	**37.18%**
b. 100-500	e) 0.1822 - 0.251	1,584	202	388	**24.47%**
b. 100-500	f) 0.251 - 4.0211	1,295	67	37	**2.88%**

This summary again shows the different probabilities across the **Ratio** Groups, but it also shows clear separation of average win probabilities between the different **Weight** Groups. From the above snapshot of the first two weight groups, it can be seen that quotes priced in the .0215 - .114 Ratio Groups have an 85.48% probability of winning if the Total Weight quoted is under 100 pounds; but only a 70.26% probability of winning if the Total Weight quoted is between 100-500 pounds. Logically, this makes sense as well, as customers generally expect lower prices at higher weights, thus the lower probability of winning at the same Ratio Groups.

The Ratio Group and Weight Group average win probability logic is then built into a Quote Pricing Tool. An Excel Spreadsheet is used to provide an easily accessible tool that can be quickly reproduced and distributed. The input simply asks for today's CRU Price, the Total Cost, and the Total Weight of the Quote to be entered by the Salesman. Behind the scenes, the tool calculates the CWT:CRU Ratio and references the corresponding probabilities for the Weight Group that each particular quote falls into. By multiplying the CWT:CRU ratio groups by the current CRU price entered, the Quote Pricing Tool's output is a range of prices with the probability of winning that quote at each price level, shown below:

		Weight Group	CWT:CRU Ratio				CWT Price			Calculations at each WgtGrp and Ratio			
Quote Details - Input		**Wgt Group**	**RatioGroup**	**Low**	**Mean**	**High**	**Low**	**Mean**	**High**	**Win Probability**	Cost per CWT	Price per CWT at Mean	GP%
CRU Price	$ 758	d. 1k-5k	a) 0.0215 - 0.114	0.0215	0.1036	0.1140		**$ 78.53**	**$ 86.41**	**28.50%**	$ 74.17	$ 78.53	5.6%
CostExt	$ 821.77		b) 0.114 - 0.1317	0.1140	0.1228	0.1317	$ 86.41	**$ 93.08**	$ 99.83	**20.50%**	$ 74.17	$ 93.08	20.3%
WgtExt	1,108		c) 0.1317 - 0.1512	0.1317	0.1413	0.1512	$ 99.83	**$ 107.11**	$ 114.61	**15.07%**	$ 74.17	$ 107.11	30.8%
			d) 0.1512 - 0.1821	0.1512	0.1651	0.1821	$ 114.61	**$ 125.15**	$ 138.03	**10.37%**	$ 74.17	$ 125.15	40.7%
			e) 0.1822 - 0.251	0.1822	0.2113	0.2510	$ 138.11	**$ 160.17**	$ 190.26	**6.01%**	$ 74.17	$ 160.17	53.7%
	d. 1k-5k		f) 0.251 - 4.0211	0.2510	0.5169	4.0211	**$ 190.26**	**$ 391.81**		**0.17%**	$ 74.17	$ 391.81	81.1%

The above example shows a quote with a total material cost of $821.77 and a total weight of 1,108 pounds, while the CRU price is $758. The tool gives the probability of winning the quote at six different price ranges. Since the total cost of material will remain the same regardless of the price charged, the Gross Margin and Gross Profit (GP%) can also be calculated for each price range.

The concept is clear, at lower prices there is a higher probability of winning a quote. But being able to actually see the differences in probability of winning at different price ranges and weight groups allows the Sales Team to visualize and understand the impact of changing the price in multiple scenarios. With the addition of the GP% column, the Salesman can quickly weigh the risk of a lower win probability against the benefit of higher gross profit at each price range. In a fast-paced and competitive market, the capability to quote aggressively while keeping profitability in mind is very important.

Taking this analysis one step further, it is also possible to automatically select the optimal price for any given quote. This is done by multiplying the Win Probability by the Margin (Price – Cost) at each Ratio Group. The result is an expected profitability at each price range; the range with the highest expected profitability is the price that should be quoted. This step requires a level of confidence in the win probabilities being calculated by the logistic regression model, a level that the business is not ready for just yet. However, the opportunity is available and could prove to be useful down the road.

Calculations at each WgtGrp and Ratio					
Win Probability	Cost per CWT	Price per CWT at Mean	GP%	Expected Profitability	Suggested Price Range
28.50%	$ 74.17	$ 78.53	5.6%	$ 1.24	
20.50%	$ 74.17	$ 93.08	20.3%	$ 3.88	
15.07%	$ 74.17	$ 107.11	30.8%	$ 4.96	
10.37%	$ 74.17	$ 125.15	40.7%	$ 5.29	***
6.01%	$ 74.17	$ 160.17	53.7%	$ 5.17	
0.17%	$ 74.17	$ 391.81	81.1%	$ 0.53	

Appendix A – R Code

```
library(WriteXLS)
library(splitstackshape)
library(dplyr)
library(ggplot2)
library(mltools)
library(DMwR)

#Import Dataset and create variable 'mydata'
mydata <- Quote_Pricing_Branch1_070
round(mydata$CWT.CRU.Ratio, digits=6)

#Determine # of CWT:CRU Ratio Bins
NbrBins <- 6
#Define Bin Min, Max, and Mean
binsMean <- tapply(mydata$CWT.CRU.Ratio, cut_number(mydata$CWT.CRU.Ratio, NbrBins),mean)
bins <- binsMean
binsMin <- tapply(mydata$CWT.CRU.Ratio, cut_number(mydata$CWT.CRU.Ratio, NbrBins),min)
binsMax <- tapply(mydata$CWT.CRU.Ratio, cut_number(mydata$CWT.CRU.Ratio, NbrBins),max)
binsMean <- as.vector(binsMean)
binsMin <- as.vector(binsMin)
binsMax <- as.vector(binsMax)
bins
binsMean
binsMin
binsMax

#Define variables for Bin Values
aMax <- round(binsMax[1], digits = 4)
bMax <- round(binsMax[2], digits = 4)
cMax <- round(binsMax[3], digits = 4)
dMax <- round(binsMax[4], digits = 4)
eMax <- round(binsMax[5], digits = 4)
fMax <- round(binsMax[6], digits = 4)

aMin <- round(binsMin[1], digits = 4)
bMin <- round(binsMin[2], digits = 4)
cMin <- round(binsMin[3], digits = 4)
dMin <- round(binsMin[4], digits = 4)
eMin <- round(binsMin[5], digits = 4)
fMin <- round(binsMin[6], digits = 4)
```

```
aMean <- round(binsMean[1], digits = 4)
bMean <- round(binsMean[2], digits = 4)
cMean <- round(binsMean[3], digits = 4)
dMean <- round(binsMean[4], digits = 4)
eMean <- round(binsMean[5], digits = 4)
fMean <- round(binsMean[6], digits = 4)

#Create NewRatioGroups Column
mydata <- within(mydata, NewRatioGroups <-
  ifelse(Quote_Won==0,
    ifelse(CWT.CRU.Ratio<bMin, capture.output(cat(bMean,cMean,dMean,eMean,fMean,sep=',')),
    ifelse(CWT.CRU.Ratio<cMin, capture.output(cat(cMean,dMean,eMean,fMean,sep=',')),
    ifelse(CWT.CRU.Ratio<dMin, capture.output(cat(dMean,eMean,fMean,sep=',')),
    ifelse(CWT.CRU.Ratio<eMin, capture.output(cat(eMean,fMean,sep=',')),
    ifelse(CWT.CRU.Ratio<eMin, capture.output(cat(fMean,sep=','))
      ,""))))),
  ifelse(Quote_Won==1,
    ifelse(CWT.CRU.Ratio<bMin, "",
    ifelse(CWT.CRU.Ratio<cMin, capture.output(cat(aMean,sep=',')),
    ifelse(CWT.CRU.Ratio<dMin, capture.output(cat(aMean,bMean,sep=',')),
    ifelse(CWT.CRU.Ratio<eMin, capture.output(cat(aMean,bMean,cMean,sep=',')),
    ifelse(CWT.CRU.Ratio<fMin, capture.output(cat(aMean,bMean,cMean,dMean,sep=',')),
    ifelse(CWT.CRU.Ratio>=fMin,capture.output(cat(aMean,bMean,cMean,dMean,eMean,sep=','))
      ,"")))))),"")) )

#Copy original CWT:CRU Ratio to the NewRatioGroup Columns
mydata$Ratio_Expanded <- paste(mydata$CWT.CRU.Ratio, mydata$NewRatioGroups, sep=",")
gsub("\\,$", "",mydata$Ratio_Expanded)

#Create Mirrored Records using the csplit function on NewRatioGroups
mydata <- cSplit(mydata, "Ratio_Expanded", ",", direction="long")

#Create new RatioGroup_Expanded record
mydata <- within(mydata, RatioGroup_Expanded <-
   ifelse(Ratio_Expanded<bMin, capture.output(cat('a)',aMin,'-',aMax,sep=' ')),
   ifelse(Ratio_Expanded<cMin, capture.output(cat('b)',bMin,'-',bMax,sep=' ')),
   ifelse(Ratio_Expanded<dMin, capture.output(cat('c)',cMin,'-',cMax,sep=' ')),
   ifelse(Ratio_Expanded<eMin, capture.output(cat('d)',dMin,'-',dMax,sep=' ')),
   ifelse(Ratio_Expanded<fMin, capture.output(cat('e)',eMin,'-',eMax,sep=' ')),
   ifelse(Ratio_Expanded>=fMin,capture.output(cat('f)',fMin,'-',fMax,sep=' '))
        ,"")))))))
mydata <- within(mydata, RatioGroup_Min <-
```

```
   ifelse(Ratio_Expanded<bMin, aMin,
   ifelse(Ratio_Expanded<cMin, bMin,
   ifelse(Ratio_Expanded<dMin, cMin,
   ifelse(Ratio_Expanded<eMin, dMin,
   ifelse(Ratio_Expanded<fMin, eMin,
   ifelse(Ratio_Expanded>=fMin,fMin
       ,"")))))))
mydata <- within(mydata, RatioGroup_Max <-
   ifelse(Ratio_Expanded<bMin, aMax,
   ifelse(Ratio_Expanded<cMin, bMax,
   ifelse(Ratio_Expanded<dMin, cMax,
   ifelse(Ratio_Expanded<eMin, dMax,
   ifelse(Ratio_Expanded<fMin, eMax,
   ifelse(Ratio_Expanded>=fMin,fMax
       ,"")))))))
mydata <- within(mydata, RatioGroup_Mean <-
   ifelse(Ratio_Expanded<bMin, aMean,
   ifelse(Ratio_Expanded<cMin, bMean,
   ifelse(Ratio_Expanded<dMin, cMean,
   ifelse(Ratio_Expanded<eMin, dMean,
   ifelse(Ratio_Expanded<fMin, eMean,
   ifelse(Ratio_Expanded>=fMin,fMean
       ,"")))))))

#Logistic Regression model using the below variables
model <- glm(formula= Quote_Won ~ Cust_WinRatio + Ratio_Expanded
          + mydata$`Wgt_0-100` + mydata$`Wgt_100-500`
          + mydata$`Wgt_500-1000` + mydata$`Wgt_1k-5k`
          + mydata$`Wgt_5k-10k`
          + mydata$`Wgt_10k-40k` + mydata$`Wgt_40k+`
          + mydata$`Cut?` + mydata$`Contract?`
          , data=mydata, family=binomial)

#Verify the coefficients are statistically relevant
summary(model)

#Add the Win_Probability (fitted model) to the original data set
mydata <- cbind(mydata, Win_Probability = fitted(model))
round(mydata$Win_Probability, digits=4)
```

#Subset of data to Write to CSV

```
mydata1 <- mydata[,c("QuoteDet","QuoteDate","Quote_Won","WgtGroup",
                "RatioGroup_Expanded","Win_Probability",
                "RatioGroup_Min","RatioGroup_Max","RatioGroup_Mean" )]
```

#Write Prediction Results to CSV - prepare for copy into Pricing Tool

```
write.csv(mydata1, "c:/Users/mverwijst/Documents/0. Miscellaneous/Analytics/Quote Pricing/Branch1_070_Mirrored_Predictions.csv")
```

Appendix B – Weight Group Table

Wgt Group	Ratio Group	Count of Quote	Sum of Quote_Won	Sum of Win_Probability	Avg Win Probability
a. 0-100	a) 0.0215 - 0.114	376	368	321	85.48%
a. 0-100	b) 0.114 - 0.1317	403	363	326	81.00%
a. 0-100	c) 0.1317 - 0.1512	430	345	326	75.75%
a. 0-100	d) 0.1512 - 0.1821	513	306	339	66.01%
a. 0-100	e) 0.1822 - 0.251	804	250	376	46.79%
a. 0-100	f) 0.251 - 4.0211	891	158	102	11.40%
b. 100-500	a) 0.0215 - 0.114	799	744	561	70.26%
b. 100-500	b) 0.114 - 0.1317	894	677	547	61.21%
b. 100-500	c) 0.1317 - 0.1512	1,104	540	543	49.23%
b. 100-500	d) 0.1512 - 0.1821	1,401	368	521	37.18%
b. 100-500	e) 0.1822 - 0.251	1,584	202	388	24.47%
b. 100-500	f) 0.251 - 4.0211	1,295	67	37	2.88%
c. 500-1,00	a) 0.0215 - 0.114	354	252	145	41.08%
c. 500-1,00	b) 0.114 - 0.1317	475	154	137	28.83%
c. 500-1,00	c) 0.1317 - 0.1512	578	94	126	21.76%
c. 500-1,00	d) 0.1512 - 0.1821	624	44	99	15.91%
c. 500-1,00	e) 0.1822 - 0.251	648	21	62	9.60%
c. 500-1,00	f) 0.251 - 4.0211	626	8	3	0.53%
d. 1k-5k	a) 0.0215 - 0.114	661	307	188	28.50%
d. 1k-5k	b) 0.114 - 0.1317	786	164	161	20.50%
d. 1k-5k	c) 0.1317 - 0.1512	849	86	128	15.07%
d. 1k-5k	d) 0.1512 - 0.1821	865	43	90	10.37%
d. 1k-5k	e) 0.1822 - 0.251	856	17	51	6.01%
d. 1k-5k	f) 0.251 - 4.0211	844	3	1	0.17%
e. 5k-10k	a) 0.0215 - 0.114	120	41	23	19.00%
e. 5k-10k	b) 0.114 - 0.1317	111	11	13	11.94%
e. 5k-10k	c) 0.1317 - 0.1512	116	5	11	9.30%
e. 5k-10k	d) 0.1512 - 0.1821	117	3	9	7.57%
e. 5k-10k	e) 0.1822 - 0.251	115	-	4	3.64%
e. 5k-10k	f) 0.251 - 4.0211	118	-	0	0.10%
f. 10k-40k	a) 0.0215 - 0.114	57	20	8	14.20%
f. 10k-40k	b) 0.114 - 0.1317	63	6	8	11.95%
f. 10k-40k	c) 0.1317 - 0.1512	60	2	6	9.69%
f. 10k-40k	d) 0.1512 - 0.1821	59	-	4	6.68%
f. 10k-40k	e) 0.1822 - 0.251	61	-	2	4.08%
f. 10k-40k	f) 0.251 - 4.0211	63	-	0	0.21%
g. 40k+	a) 0.0215 - 0.114	24	3	2	9.36%
g. 40k+	b) 0.114 - 0.1317	28	3	3	9.14%
g. 40k+	c) 0.1317 - 0.1512	29	3	2	7.74%
g. 40k+	d) 0.1512 - 0.1821	26	-	1	4.76%
g. 40k+	e) 0.1822 - 0.251	26	-	1	2.69%
g. 40k+	f) 0.251 - 4.0211	26	-	0	0.04%