# Maximizing Efficiency and Profitability of Catalog Distribution

**Prepared for XYZ Direct Marketing, Inc.**

by

Daniel Rodriguez

Shawn Mire

Stephen McClure

MIS 6324

Dr. Zheng

November 14, 2012

# Maximizing Efficiency and Profitability of Catalog Distribution

**Prepared for XYZ Direct Marketing, Inc.**

# I. Executive summary

The objective of this project is to maximize revenue, and minimize catalog expense, ensuring that XYZ’s ROI is maximized.

**Background**

XYZ Direct Marketing has a customer base of 5 million households. If all households are sent catalogs (at a cost of $4 each), that is an expense of $20 million. The current response rate is about 30%, with an average catalog order amount of $16, with gross sales thus at about $24 million. With the large catalog expense, that leaves a net income of $4 million. Roughly $14 million of the mailing expense is money lost—a significant dollar amount that could be better spent elsewhere.

What if catalogs could be sent to just those 1.5 million customers who place orders? That would bring catalog expenses down to just $6 million, and since gross sales would remain constant, net income would surge to $18 million.

The business opportunity is to cut down on expenses by targeting a subset of those 5 million households. If catalogs were only sent to customers who are more likely to respond and order, then total catalog expenses would be lower and the response rate would be higher, although the response in absolute terms should be roughly the same. In other words, simply by targeting likely customers the bottom line could be improved. Since the company is concerned with lapsing customers, there is also an opportunity to determine whether targeting those customers results in responses.

**Methodology**

The business intelligence problem is thus to look at historical data to determine which attributes are shared by responding consumers, and then to look for those same attributes in future potential consumers, to predict who will more likely respond to catalog mailings. Those attributes should be identified which more strongly indicate responsiveness.

As there are a variety of predictive techniques, each one should be evaluated by splitting the historical data into a training set (to analyze and develop a model) and a testing set (to see how well the model works). The technique and model whose results most closely align with the testing set data will be the one we recommended to XYZ.

We investigated a variety of Business Intelligence techniques in SAS Institute’s Enterprise Miner.

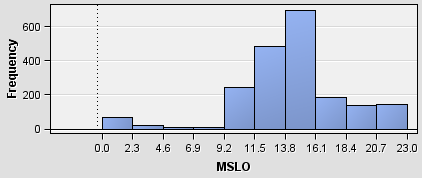
We trained and tested the various techniques with a dataset of 4,000 customers (2,000 each for training and testing), and compared the results for each model. These 4,000 customers provided a good sample:

* The overall response rate was 29.9% and that in the training dataset (584 orders out of 2000 observations.
* The test dataset had a comparable response rate of 28.4% (568 orders out of 2000 observations).
* This closely matches the response rate of the overall customer base, which is around 30%. The sample data, therefore there is no evidence of oversampling.

Before inserting the data into the models, we obtained statistics from Enterprise Miner’s *Explore* and *Graph Explore* nodes, and we pre-processed some data: we converted the LASD (Date since last order) to MSLO (Months since last order) using the *Transform Variables* node with the following formula:

MSLO = (2007 - INT(SUBSTRN(LASD,1,4)))\*12 + 2 - INT(SUBSTRN(LASD,5,6))

Given the company’s interest in the customers who made an order between 13 and 24 months ago, we determined that most of the customers are made an order between 9 and 23 months ago, with a median of 14 months ago, as shown in the histogram below:



Other important statistics about the data can be seen in the table below. Notable are the median NGIF (number of orders in the 24 months) of 8, and the median of RAMN (Total order amounts in dollars in the last 24 months) of 80.9. Also looking at the charts we note that both variables have a lot of outliers, mainly RAMN.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Variable** | **Mean** | **Median** | **Standard Deviation** | **non missing** | **Minimum** | **Maximum** |
| LASG | 16.86 | 15 | 16.62 | 2000 | 0 | 415 |
| MSLO | 14.11 | 14 | 4.15 | 2000 | 0 | 23 |
| NGIF | 10.29 | 8 | 8.81 | 2000 | 1 | 82 |
| RAMN | 107.83 | 80.9 | 126.75 | 2000 | 15 | 2200 |
| RFA1 | 2.02 | 2 | 1.11 | 2000 | 1 | 4 |
| RFA2 | 5.73 | 6 | 0.87 | 2000 | 4 | 7 |

Variable Key:

* LASG: Amount of last order
* MSLO: Months since last order until 2/1/2007
* NGIF: Number of orders in the last 24 months
* RAMN: Total revenue in last 24 months
* RFA1: Frequency of order; number of orders in last 24 months
* RFA2: Order amount category, as defined by XYZ, for the last order.

1=$0.01 to $1.99

2=$2.00 to $2.99

3=$3.00 to $4.99

4=$5.00 to $9.99

5=$10.00 to $14.99

6=$15.00 to $24.99

7=$25.00 and above

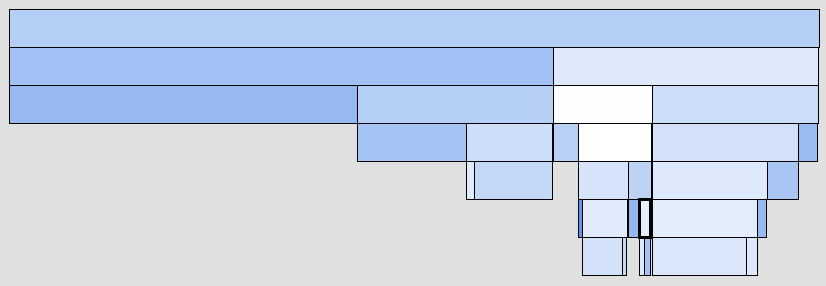
We trained 5 models and compared them using Enterprise Miner: Decision Tree, Logistic Regression, Neural Network, Memory Based Reasoning and Gradient Boosting. The selected model was Decision Tree. A summary of our findings appears in Part II.

# II. Comparing the 'Training' Models Using SAS Enterprise Miner

**Decision tree**

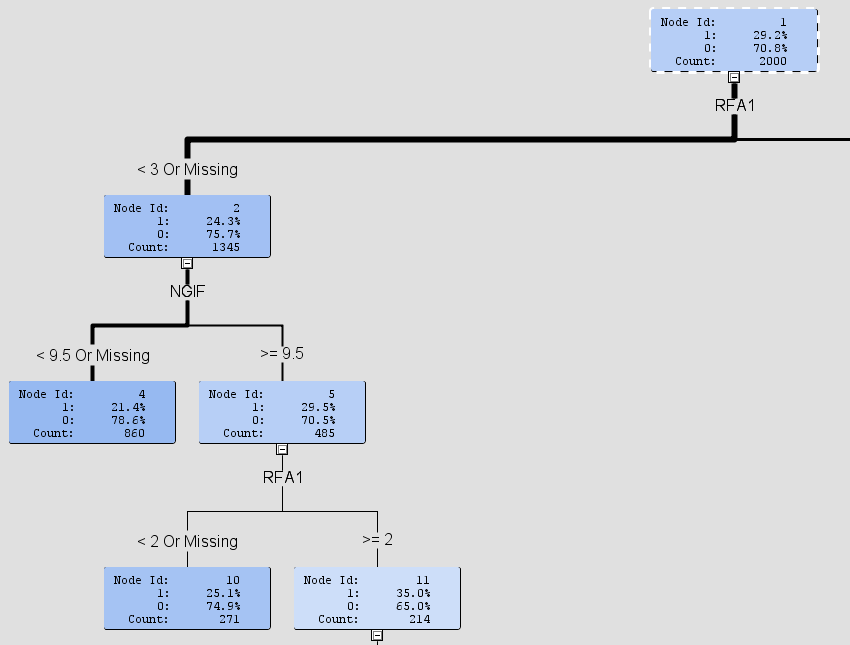
The decision tree was built to a depth of 6 and a minimum leaf size of 10. The first data attribute evaluated in the tree, which is the attribute that provides the most information gain, is RFA1 (Frequency of Order.) According to the model, those customers with fewer than three orders in the last 24 months will not place an order, with 75.7% probability. Those customers with three or more orders in the last 24 months are still likely to not place an order, but have a lower probability of not placing an order (60.8%), than those who had placed fewer than three orders in the past 24 months (75.7% probability).

The left hand side of the tree ends after 4 levels, while the right hand side continues on to six levels. This is shown graphically via the Tree Map:



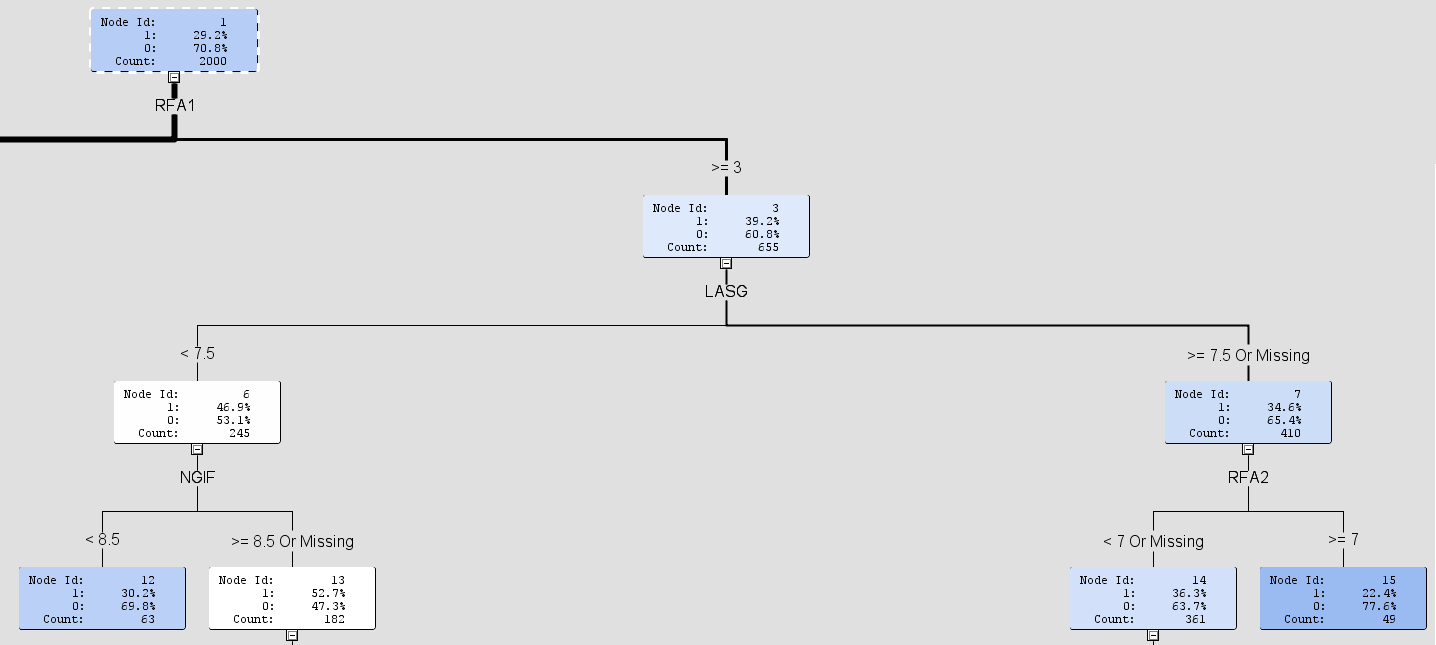
The top three levels below the root node of the decision tree are presented below as a series of if/then rules:

*Left-hand Side*



* **IF** RFA1 (Frequency of Order) is less than 3
  + **AND IF** NGIF (Number of orders in the last 24 months) is less than 9.5
    - **THEN** result is “No order” with a probability of 78.6%
  + **ELSE IF** NGIF (Number of orders in the last 24 months) is greater than 9.5
    - **AND IF** RFA1 (Frequency of Order) is less than 2
      * **THEN** result is “No order” with a probability of 74.9%
    - **ELSE IF** RFA1 (Frequency of Order) is greater than 2
      * **THEN** result is “No order” with a probability of 35%

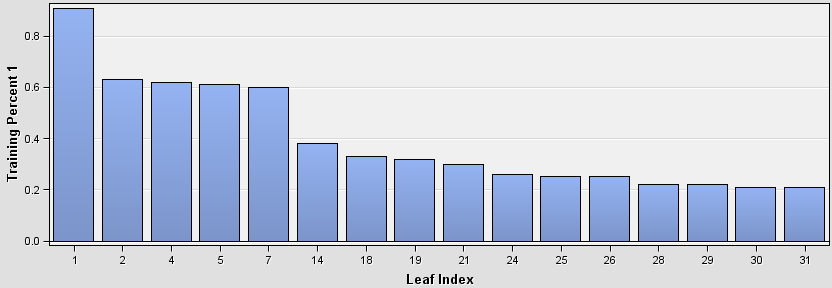
*Right-hand Side*



* **IF** RFA1 (Frequency of Order) is greater than or equal to 3
  + **AND IF** LASG (Amount of last order) is less than 7.5
    - **AND IF** NGIF (Number of orders in the last 24 months) is less than 8.5
      * **THEN** result is “No order” with a probability of 69.8%
    - **ELSE IF** NGIF (Number of orders in the last 24 months) is greater than or equal to 8.5
      * **THEN** result is “Order” with a probability of 52.7%
  + **ELSE IF** LASG (Amount of last order) is greater than or equal to 7.5
    - **AND IF** RFA2 (Order amount category) is less than 7
      * **THEN** result is “No order” with a probability of 63.7%
    - **ELSE IF** RFA2 (Order amount category) is greater than or equal to 7
      * **THEN** result is “No order” with a probability of 77.6%

The leaf with the highest percentage of responses (90.9%) occurs on level 5, and is on the attribute MSLO (Months Since Last Order Until 2/1/2007). But this node only has 11 observations (0.55% of the total data), so it is not very reliable.

The leaves with the next highest percentages of responses are just above 60% and occur at the lowest level of the tree on level 6.



In analyzing the tree for lapsing customers, we did note that MSLO (Months Since Last Order Until 2/1/2007) was evaluated three times near the bottom of the decision tree, in levels four, five, and six. It does not appear to be a very important data attribute given these three positions in the tree (only 485 observations are classified by it); but when it is evaluated, the child nodes do show a greater percentage of responses tend to come from customers with recent orders.

**Logistic regression**

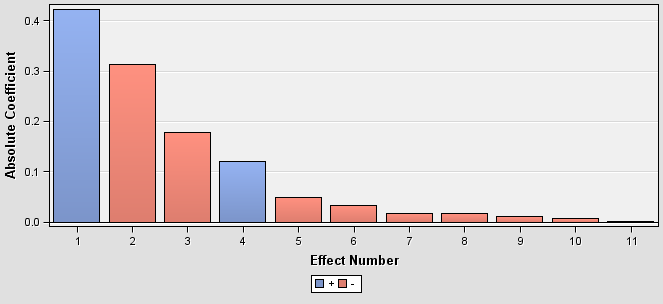
Since the target variable (Order) is a binary variable, we utilized logistic regression instead of a linear regression, just as Enterprise Miner does:

*“The Regression node automatically performs logistic regression if the target variable is a class variable that takes one of two values. If the target variable is a continuous variable, then the Regression node performs linear regression.”* [1]

Since RFA1 and RAF2 were defined as *ordinal* variables, Enterprise Miner divides these two variables into 6 variables (3 per variable). The results were as follows:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | DF | Estimate | Standard  Error | Wald Chi-Square | Pr > ChiSq | Standardized Estimate | Exp(Est) |
| RFA2        [4] | 1 | 0.4238 | 0.1635 | 6.72 | 0.0095 |  | 1.528 |
| RFA1        [1] | 1 | -0.3131 | 0.0954 | 10.77 | 0.001 |  | 0.731 |
| Intercept | 1 | -0.1789 | 0.236 | 0.57 | 0.4485 |  | 0.836 |
| RFA1        [3] | 1 | 0.1201 | 0.0994 | 1.46 | 0.2267 |  | 1.128 |
| RFA2        [6] | 1 | -0.049 | 0.0958 | 0.26 | 0.609 |  | 0.952 |
| MSLO | 1 | -0.033 | 0.0131 | 6.41 | 0.0113 | -0.0756 | 0.967 |
| RFA1        [2] | 1 | -0.0175 | 0.0972 | 0.03 | 0.8573 |  | 0.983 |
| RFA2        [5] | 1 | -0.0169 | 0.0982 | 0.03 | 0.8633 |  | 0.983 |
| LASG | 1 | -0.0102 | 0.00737 | 1.91 | 0.1669 | -0.0934 | 0.99 |
| NGIF | 1 | -0.00621 | 0.00949 | 0.43 | 0.5126 | -0.0302 | 0.994 |
| RAMN | 1 | 0.000883 | 0.000685 | 1.66 | 0.1972 | 0.0617 | 1.001 |

In the chart above, the smaller the size of the Pr > ChiSq value, the more important the coefficient is. Specifically, one would look for those values where it is lower than 0.05. As per the results, RFA1 [4] (Frequency of order - Four or more orders in the last 24 months) and RFA2 [1] (Frequency of order - One order in the last 24 months) are the most important, followed by MSLO (Months Since Last Order Until 2/1/2007).



**NN (Neural Network)**

The neural network model was built as a multilayer perceptron with 6 hidden units. The transfer function (activation function) used was logistic. The back propagation algorithm was used for training this model. Sample output is provided below; we sorted by percentage instead of posterior probability to see where the weight is higher.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Posterior  Probability Range** | **Number  of Events** | **Number  of non-events** | **Mean  posterior probability** | **Percentage** |
| 0.20-0.25 | 128 | 489 | 0.21968 | 30.85 |
| 0.30-0.35 | 93 | 195 | 0.32511 | 14.4 |
| 0.25-0.30 | 79 | 204 | 0.27339 | 14.15 |
| 0.15-0.20 | 42 | 175 | 0.18291 | 10.85 |
| 0.35-0.40 | 69 | 113 | 0.3722 | 9.1 |
| 0.40-0.45 | 44 | 66 | 0.42202 | 5.5 |
| 0.45-0.50 | 30 | 30 | 0.47374 | 3 |
| 0.10-0.15 | 10 | 49 | 0.12841 | 2.95 |
| 0.50-0.55 | 25 | 19 | 0.52735 | 2.2 |
| 0.05-0.10 | 2 | 32 | 0.07308 | 1.7 |
| 0.55-0.60 | 19 | 13 | 0.56956 | 1.6 |
| 0.60-0.65 | 14 | 11 | 0.61622 | 1.25 |
| 0.65-0.70 | 13 | 5 | 0.66894 | 0.9 |
| 0.00-0.05 | 0 | 12 | 0.03203 | 0.6 |
| 0.80-0.85 | 5 | 0 | 0.81889 | 0.25 |
| 0.70-0.75 | 3 | 2 | 0.71707 | 0.25 |
| 0.75-0.80 | 3 | 1 | 0.76444 | 0.2 |
| 0.90-0.95 | 3 | 0 | 0.92571 | 0.15 |
| 0.85-0.90 | 2 | 0 | 0.86868 | 0.1 |

**MBR (Memory Based Reasoning) using KNN (K-nearest neighbors)**

The memory based reasoning model was built with K set to 5, meaning that the observations would be divided into five groups based on the distance between the various attributes in each observation. The MBR method chosen was RD-Tree. Some sample output is provided below. Note that MBR does not accept ordinal inputs; therefore, only interval inputs are used by the algorithm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Posterior Probability Range** | **Number of Events** | **Number of non-events** | **Mean posterior probability** | **Percentage** |
| 0.95-1.00 | 9 | 0 | 1 | 0.45 |
| 0.85-0.90 | 7 | 22 | 0.88462 | 1.45 |
| 0.80-0.85 | 5 | 14 | 0.84211 | 0.95 |
| 0.75-0.80 | 56 | 7 | 0.8 | 3.15 |
| 0.70-0.75 | 2 | 6 | 0.75 | 0.4 |
| 0.65-0.70 | 5 | 16 | 0.66667 | 1.05 |
| 0.55-0.60 | 141 | 98 | 0.6 | 11.95 |
| 0.40-0.45 | 1 | 10 | 0.42105 | 0.55 |
| 0.35-0.40 | 220 | 300 | 0.4 | 26 |
| 0.15-0.20 | 137 | 544 | 0.2 | 34.05 |
| 0.05-0.10 | 1 | 11 | 0.08333 | 0.6 |
| 0.00-0.05 | 0 | 388 | 0 | 19.4 |

**Gradient Boosting**

The gradient boosting method is a machine learning algorithm (similar to Neural Network) that searches for an optimal partition of the data for a single target variable:

*“Gradient boosting is an approach that resamples the analysis data several times to generate results that form a weighted average of the resampled data set. Tree boosting creates a series of decision trees that form a single predictive model.”*[1]

The parameters used were:

* Maximum branch: 2, to create Binary trees
* Maximum depth: 10
* Leaf fraction: 0.01. That is 1% of the number of records (2000) = 20

Below is the Score Output for the Gradient Boosting method that shows the Importance of each variable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | Number of Splitting Rules | **Number   of Surrogates Rules** | Importance | H |
| RAMN | 726 | 543 | 1 | 0.29106 |
| NGIF | 306 | 841 | 0.92408 | 0.18413 |
| LASG | 296 | 731 | 0.88067 | 0.21962 |
| RFA2 | 55 | 469 | 0.66721 | 0.03797 |
| MSLO | 350 | 202 | 0.62413 | 0.22271 |
| RFA1 | 116 | 263 | 0.55754 | 0.12966 |

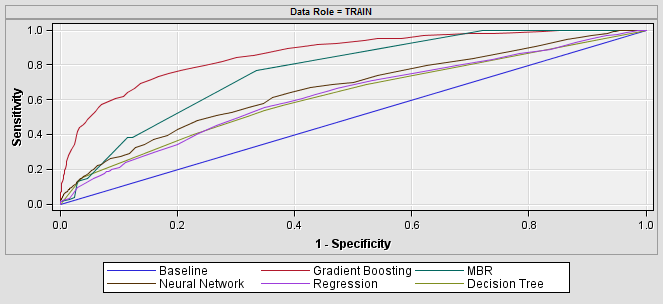
**Comparison**

Using the *Model Comparison* node on Enterprise Miner, it is possible to compare the different models used. Enterprise Miner selects a best/champion model; in our case, EM selected Gradient Boosting.

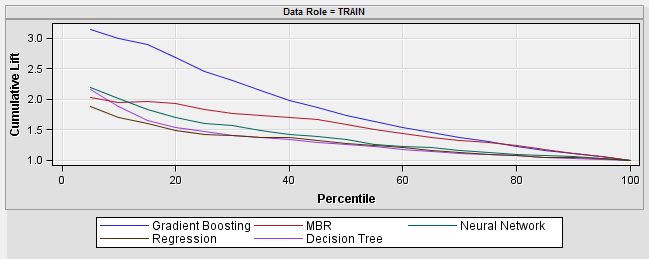
It is important to note on the table below that the misclassification rate (which is an indicator of the accuracy of the model) between the champion and the second is 5.35% while the difference between the champion and the worst model (Regression) is 9.56%. So the Gradient Boosting method is by far the best method of all tested.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model Description | Misclassification Rate | Average Squared Error | Maximum Absolute Error | Sum of Squared Errors |
| Gradient Boosting | 0.1970 | 0.142 | 0.912 | 570.920 |
| MBR | 0.2514 | 0.163 | 0.916 | 632.842 |
| Neural Network | 0.2740 | 0.189 | 0.926 | 758.933 |
| Decision Tree | 0.2740 | 0.195 | 0.786 | 780.699 |
| Regression | 0.2935 | 0.197 | 0.870 | 791.166 |

The Receiver Operating Curve (ROC) chart shows the sensitivity vs. the specificity of the different models used. Sensitivity is the proportion of target values that are predicted as a value of 1 and are actually equal to 1. Specificity is the proportion of observations that are predicted as a value of 0 and are actually equal to 0. [2. Page 279]. This is another indicator of accuracy and the general rule is that a higher value better, on that case Gradient Boosting is better.



Another indicator of accuracy is the cumulative lift. The cumulative lift is above 1.0 for all the models, this indicates a reasonably accurate model. However the best model is again Gradient Boosting.



From the results of each node it is possible to get the Confusion matrix for each model. To get an idea of what to expect we calculate the total number of 1s (positive response) and the total number of zeros (negative response). This helps us to know what to expect from the confusion matrix. Total 1s = 568 and Total 0s = 1432

|  |  |  |  |
| --- | --- | --- | --- |
| **Confusion Matrix** | | **Predicted** | |
| **Method** | **Actual class** | 1 | 0 |
| Decision tree | 1 | 111 | 473 |
| 0 | 62 | 1354 |
| Logistic Regression | 1 | 19 | 565 |
| 0 | 22 | 1394 |
| Neural Network | 1 | 87 | 497 |
| 0 | 51 | 1365 |
| MBR | 1 | 223 | 361 |
| 0 | 164 | 1252 |
| Gradient Boosting | 1 | 231 | 353 |
| 0 | 41 | 1375 |

Using the confusion matrix we can compare the accuracy of the training by adding the True Positives and True Negatives. As we can see on the table below the accuracy is better for the Gradient Boosting method with an accuracy of 80.3% which is 6.55% more than the second (MBR).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Decision tree | Logistic Regression | Neural Network | MBR | Gradient Boosting |
| Accuracy | 0.7325 | 0.7065 | 0.726 | 0.7375 | 0.803 |

Finally using the Confusion Matrix we can obtain of the Net Revenue that we could get by using each model. This gives us an idea of the profitability of each model in case of sending/not sending catalogs to the 2000 persons on the training dataset.

Again the best model is the Gradient Boosting because it has the most higher number of responses and also has the highest Net Revenue (Revenue of sending the catalogs minus the Loss from False Positives) and also has the lowest Loss from False Negatives, which is an opportunity cost not directly an expense.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Loss from False Negatives** | **Loss from False Positive** | **Catalogs sent** | **Responses** | **Revenue** | **Net Revenue** |
| Decision tree | $      5,676 | $     248 | 173 | 111 | $     1,332 | $     1,084 |
| Logistic Regression | $      6,780 | $       88 | 41 | 19 | $        228 | $       140 |
| Neural Network | $      5,964 | $     204 | 138 | 87 | $     1,044 | $       840 |
| MBR | $      4,332 | $     656 | 387 | 223 | $     2,676 | $     2,020 |
| Gradient Boosting | $      4,236 | $     164 | 272 | 231 | $     2,772 | $     2,608 |

This table is very important because it compares all of the results we obtained so far in an objective way. For example we can see that the Logistic Regression has the lower False Positive amount this can looks good at the beginning but the reason for that is that model only send 41 catalogs and this is not good for the revenue.

These results also offer a comparison between using and not using BI. We can calculate the revenue of not using BI and just send catalogs for the 2000 persons on the training dataset as follows:

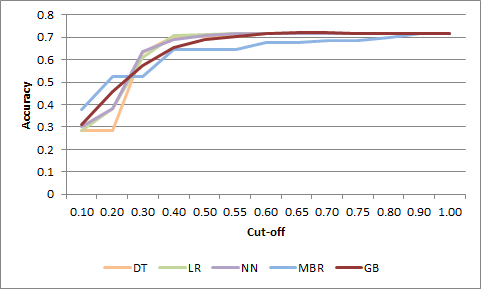
* Cost (send 2000 catalogs at $4 each) = $8000
* Revenue (for 586 positive responses with average revenue of $12) = $6816
* Net revenue = $-1184

As we can see if the company sent catalogs to all the 2000 persons and therefore generate the most number of responses (number of sales = 586) they generate a negative revenue. While using the worst model (Regression) it makes the company to win money ($140) and using the best model makes then win a lot more ($3,792). That suggests that if the sampling used to select the 2000 records is good the company is losing money on sending the catalogs.

# III. Compare the 'Scoring' Results Using SAS Enterprise Miner

The first thing that we need to define is the cutoff for the scores. The cutoff tells us when a probability is high enough to predict a consumer purchase, and send the catalog to that person. This decision is subjective, depending on the needs of the modeler. For our modeling, we have actual historical data available; this data would ordinarily be available months after sending the catalog mailing. Having historical data available allows us to test various cutoff levels, to determine which level yields the best results. Our initial approach was to begin with a cutoff that was more strict than 0.5 (random), such as 0.6.

By taking a few samples of the accuracy (using the Confusion Matrix), we generated the graph below. We can see that most of the models have a similar behavior, with relatively small improvements above a cutoff level of 0.6. Accuracy has a top limit of approximately 0.72. In conclusion, we were content with a cutoff level of 0.6. Finally we can see that Gradient Boosting has a more smoother curve, with fewer extreme changes than the others, but it has a little lower accuracy with a cutoff lower than 0.4.



With the cutoff selected, we can compare the results of the Confusion Matrix for each model. We can see that the utilizing the Logistic Regression model would result in no catalogs being distributed, as none of the probabilities for this model is higher than 0.6 (the maximum value is 0.578). Thus, we can discard this method.

|  |  |  |  |
| --- | --- | --- | --- |
| **Confusion Matrix** | | **Predicted** | |
| **Method** | **Actual class** | 1 | 0 |
| Decision tree | 1 | 73 | 495 |
| 0 | 68 | 1364 |
| Logistic Regression | 1 | 0 | 568 |
| 0 | 0 | 1432 |
| Neural Network | 1 | 29 | 539 |
| 0 | 30 | 1402 |
| MBR | 1 | 47 | 521 |
| 0 | 123 | 1309 |
| Gradient Boosting | 1 | 43 | 525 |
| 0 | 43 | 1389 |

Using this Confusion Matrix, it is possible to get the accuracy of each model by adding the true positives and true negatives of each model. By this measure, the best model is the Decision Tree with an accuracy of 71.85%; however, the difference is not much if we compare it with the other models. Gradient Boosting and Logistic Regression (previously discarded) have a variance of only 0.25 percentage points below that of Decision Tree. For reference, the difference of accuracy on the training dataset was of 6.55%. Only looking at the accuracy we can conclude that any of the four best models is good as the difference is not greater than 0.5 percentage points.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | DT | LR | NN | MBR | GM |
| **Total Correct Predictions** | 1437 | 1432 | 1431 | 1356 | 1432 |
| **Accuracy** | 71.85% | 71.6% | 71.55% | 67.8% | 71.6% |

Using the confusion matrix, we can do the same profit comparison as we did with the training dataset. On this case the best model is again the Decision Tree, which is unsurprising, given the above finding on accuracy.

We found that Net Revenue fell short of our expectations overall, with the Decision Tree yielding the maximum revenue of $604, far short of revenue that model yielded in the training dataset ($2,608). Revenues are summarized in the table below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Loss from False Negatives** | **Loss from False Positive** | **Catalogs sent** | **Responses** | **Revenue** | **Net Revenue** |
| Decision tree | 71.85% | $   5,940 | $     272 | 141 | 73 | $       876 | $     604 |
| Logistic Regression | 71.60% | $   6,816 | $        - | 0 | 0 | $          - | $        - |
| Neural Network | 71.55% | $   6,468 | $     116 | 59 | 29 | $       348 | $     232 |
| MBR | 67.80% | $   6,252 | $     188 | 170 | 47 | $       564 | $     376 |
| Gradient Boosting | 71.60% | $   6,300 | $     172 | 86 | 43 | $       516 | $     344 |

We wanted to find out the reason for this large difference in expected profit from the model (431%), since the accuracy was not that different (2.52%) between the results with training and testing. The following table compares the confusion matrix with training and testing; a plus sign indicates the value increased from training to testing, and a minus sign indicates that the value decreased.

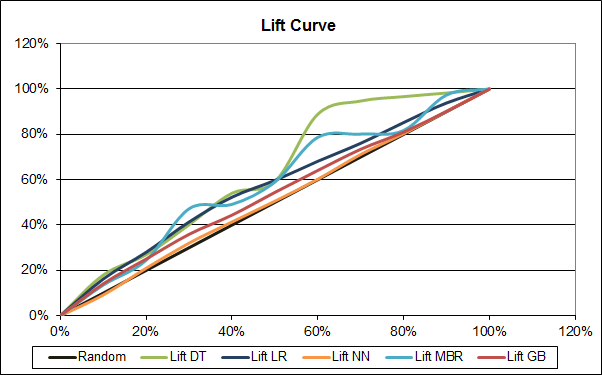
|  |  |  |  |
| --- | --- | --- | --- |
| **Confusion Matrix** | | **Predicted** | |
| **Method** | **Actual class** | 1 | 0 |
| Decision tree | 1 | - | + |
| 0 | + | + |
| Logistic Regression | 1 | - | + |
| 0 | - | + |
| Neural Network | 1 | - | + |
| 0 | - | + |
| MBR | 1 | - | + |
| 0 | - | + |
| Gradient Boosting | 1 | - | + |
| 0 | + | + |

We can see that both True Positives and False Positives decreased in most of the cases while both False Negatives and True Negatives increased. This means that fewer catalogs would be sent out if we used the people on the testing dataset than the people of the training dataset. This directly affects the revenue obtained.

This result was expected, as the model is optimized with the training dataset, and it is common that “real world” data will yield lower profit than the model. Regardless, four of the five tested models did yield a profit for XYZ.

Given the objective of profit maximization, we selected the Decision Tree method. It has the highest profit and the best response rate (51.77%), and also has the lower Loss from False Negatives (opportunity cost).

The lift curves for each model’s score results are shown below:



Based on the lift curves, the neural network is the worst model, as its selections ma are no better than just random guessing.

Gradient boosting has a slightly better lift curve than neural network, although it approaches random at about the 80th percentile.

Both the decision tree and the MBR models scoring data resulted in lift curves with a good distance above the random line. However, both curves are bumpy, and this is an indication of overfitting. [3, p 185]

The best model from the perspective of lift curves, therefore, is logistical regression. It is a steady, smooth curve, and extends further above the random line than the curves for neural network or gradient boosting.

Earlier we mentioned that the training and testing data was not oversampled. If the data were oversampled, the charts would need to be adjusted to match the original population. [4, p182-3]

# IV. Analysis and recommendations.

**Comparison of scoring results and training results**

A comparison of accuracy and net revenue between the training and testing dataset is shown below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy Training** | **Accuracy Testing** | **Accuracy Error** | **Net Revenue Training** | **Net Revenue Testing** | **Net Revenue Error** |
| Decision tree | 73.25% | 71.85% | 1.91% | $      1,084 | $     604 | 44.28% |
| Logistic Regression | 70.65% | 71.60% | -1.34% | $         140 | $        - | 100.00% |
| Neural Network | 72.60% | 71.55% | 1.45% | $         840 | $     232 | 72.38% |
| MBR | 73.75% | 67.80% | 8.07% | $      2,020 | $     376 | 81.39% |
| Gradient Boosting | 80.30% | 71.60% | 10.83% | $      2,608 | $     344 | 86.81% |

We recommend utilizing the Decision Tree model, which was selected because it better fits the testing dataset than do the other models.

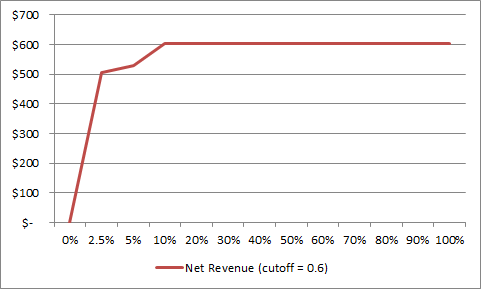
The table above illustrates the advantages of the Decision Tree model:

* It has a favorable accuracy rate in training and testing;
* It has a low accuracy error;
* While its training revenue was lower than MBR and Gradient Boosting, its testing revenue exceeded the other models, with a ret revenue error below the other models.
* Although there could be some “over-fitting” with the Decision Tree, as there was with the MBR model (evidenced by the lift curves with “bumps”), we recommend that the model be monitored for potential “over-fitting” upon implementation).
* Another indicator of over-fitting is the difference (error) between the training and testing on that case the best model is the Decision Tree.

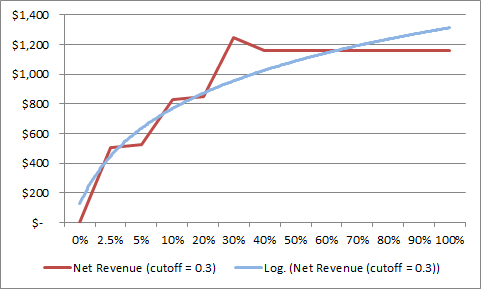
**Revenue and profit implications**

Revenue and profitability are a function of how many customers receive the mailing, the probability of purchase, and the cost of the mailing.

The chart below illustrates the predicted Net Revenue generated by the model. If we look at the whole sample we get a profit of $604, but If we sort the sample by probability of success and divide it into percentiles we can see where the weights of the profit are.



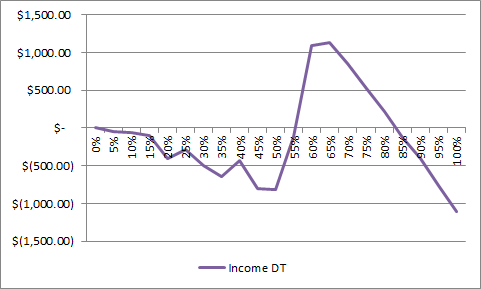
As we can see on chart above after the 10th percentile the difference on Net Revenue is null. We can say that after sending catalogs to the 10th percentile we maximize the result. The reason for that is that the cutoff is too high for the model; no more than 175 scores are higher than 0.6, by lowering the cutoff value to 0.3, we can see a different result. It is important to note that with a cutoff of 0.3 the accuracy decreases from 71.85% to 63%.



The Net Revenue chart above suggests that above the 40th percentile the Net Revenue is maximized, but looking at the Logarithmic Trend Line we can see that after the 60th percentile the increase is lower. If we look at the big picture is better to select the 60th of the sample to send the catalog, that is 1200 people.

After selecting the number of people and by already knowing the results of the 2000 customers we can calculate the money that we saved for the company.

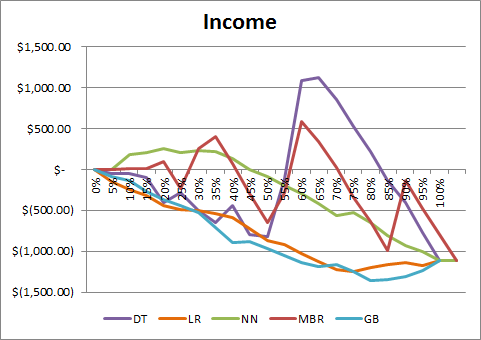
To do this we sorting the scoring data by probability of success and by calculating then we calculate the income (revenue less expenses) at each percentile.



The meaning of this chart is that by sending catalogs to the first 200 people (10th percentile) using the Decision Tree scores we get a loss of $-55.61 but if we send to 65th percentile of the sample we generate the an income of $1127.60.

Also with this chart we can confirm a prediction we made on section II. We predicted that if the company sent catalogs to the 2000 people on the data set they will lose money, and the value on the 100th percentile ($-1113.15) confirm that prediction

Finally with the results of the dataset, we can see if the model selected was the best model or if we make the wrong decision.



We can see that the Logistic Regression and Gradient Boosting models only generate losses while the MBR model is too random to generate a conclusion. The Neural Network model also generates positive income before the fifth percentile. Finally we can see that the Decision Tree is the best model if we are looking only for the higher income.

**Recommendations**

XYZ should start using some kind of classification system for their catalog mailings rather than sending catalogs to all customers, in order to reduce expenses. We believe that XYZ can significantly increase revenue from catalog mailings, and minimize cost, by optimizing distribution through the application of Business Intelligence techniques.

While both Decision Tree and Neural Network techniques generate a positive income from catalog mailings, we recommend utilizing the Decision Tree technique, as it has a higher Return to Risk ratio. Defining the risk as the standard deviation of the profit and the return as the highest profit, we found the following values:

* Risk of using Decision Tree = $615.75
* Return of using Decision Tree = $1127.60
* Return to Risk ratio of using Decision Tree = 1.83
* Risk of using Neural Network = $487.52
* Return of using Neural Network = $228.61
* Return to Risk ratio of using Neural Network = 0.46

While Neural Network’s risk is somewhat lower than that of Decision Tree, we believe that the revenue potential associated with the Decision Tree technique is worth the marginal additional risk. We recommend sending catalogs to the top 60-75% of the classified customers, as that percentile generates the higher income.

**Summary of learning**

* Developing models and choosing the best one is not enough; testing of the models is essential. In this project, we observed significant differences in the results of training vs. testing datasets. By XYZ distributing a catalog to all 2,000 people in the test dataset and recording the response, we are able to compare actual responses to what our models would have chosen. This comparison is completed before any operational changes are made, which allows for further learning and opportunities for revenue maximization and cost minimization.
  + We observed one case in which a winning model on the training dataset (Gradient Boosting) had significantly worse performance in the testing dataset. If that model had been chosen based only on the training results, we would have lost money.
* Since the test results were not as good as training led us to expect, it is important to keep trying to refine the models. We recommend that XYZ gather more historical data as well as explore using additional data attributes. Business Intelligence is a continuous process.
* Business Intelligence does not give exact answers; errors will be made by the models as they are based on learning and probabilities of success. In this project’s testing phase, we found the Decision Tree to be a good fit on some cases, but not as good as anticipated in other cases. For example, the training scores did not shown any income loss, across percentiles, while the testing results show that below the 55th percentile, losses occurred.
* We observed the importance of going beyond statistical measures, such as accuracy and lift. The “real-world” objectives of this project, revenue and profit maximization are far more important to XYZ than accuracy or lift.
* While software tools such as Enterprise Miner are essential for conducting Business Intelligence projects such as this one, there is still a very important role for management to subjectively review the results of models. Model results should be viewed through the lens of experience and industry knowledge.

# References

[1] SAS Institute Inc 2011. Getting Started with SAS® Enterprise Miner™ 7.1. Cary, NC: SAS Institute Inc. Available online: <<http://support.sas.com/documentation/cdl/en/emgsj/64144/PDF/default/emgsj.pdf>>

[2] Cerrito, Patricia B. 2006. Introduction to Data Mining Using SAS ® Enterprise Miner™. Cary, NC: SAS Institute Inc.

[3] Berry, Michael J.; Linoff, Gordon S.. Data Mining Techniques : For Marketing, Sales, and Customer Relationship Management (3rd Edition). Hoboken, NJ, USA: \*Wiley Computer Publishing, 2011. p 185. http://site.ebrary.com/lib/utdallas/Doc?id=10513818&ppg=223 Copyright © 2011. \*Wiley Computer Publishing. All rights reserved.

[4] ibid., p 182-3.