DS705 Final Project - Predicting Loan Defaults

Pete Kelley

Aug 17, 2018

## Section 1 - Executive Summary

This report analyzes the creating and fine-tuning of a generalized linear model to maximize profit for the personal loan department. Eight models were created and compared to each other with measures such as prediction accuracy, maximum profit and other standard statistical measures of model fitness.

Data preparation included limiting input variables to only those available at the time a loan application was submitted, which would allow for partially or completely automating the loan approval process. The data was also evaluated and, where appropriate, normalized to reduce the influence of extreme values. Finally, data was converted into more useful forms such as the text string of "5 years" into the number 5 so it would be treated as part of a scale rather than as a label.

The results of the model were impressive and showed that the algorithm significantly improved the personal loan department profitability by limiting the number of unprofitable loan approvals.

Of the models, the best performing one increased the portfolio profit by 14 fold over the current process ($0.26M to $3.7M). Even after such a significant improvement, there is still more that can be done to maximize loan portfolio profitability. A perfect model would produce over 300% more profit by rejecting all bad loans and approving all good ones. The final model tuned for profit maximization using a threshold score of .73, that rejects 56% of bad loans and approves 78% of good loans.

#### Recommendations

* Use this algorithm to make automated accept or reject decisions for personal loans, allowing some discretion with accountability to loan officers when they go against the algorithm's suggestions.
* Conduct a follow-up study to determine if other methods or predictors do an even better job of differentiating good and bad loan applications.

A limitation of the scope of this project was that only linear models were used to predict if a loan should be approved or rejected; there may be other more effective models that would further maximize profit potential and they should be investigated.

## Section 2 - Introduction

This study uses a dataset of 30 variables across 50,000 loans to create a predictive model to help a bank identify which applicants are likely to default on their loans. The dataset was not ready to be evaluated and needed to be prepared for modeling. This paper will review data preparation, building and optimizing the model, and finally summarizing the results.

## Section 3 - Preparing and Cleaning the Data

The following was done to prepare the data for processing:

* Added a column for the response variable
* status column values of “Fully Paid” mapped to “Good”
* status column values of “Charged Off” mapped to “Bad”
* Removed rows that did not fit the parameters for a valid response variable
* Valid response rows were those that had reached end of life and were either paid in full or written off; all other rows were deleted
* Removed columns that are unknowable at the time a loan was requested
* Grade of the loan application, which is a result of the loan application decision process
  + Status of the loan, this was mapped to the response variable
* Kept but not used to create a model, only used to fine tune profitability
* Rate of the loan, which is a result of the loan application decision process
* TotalPaid is not knowable until the loan reaches end of life
* Removed loadID column which contains that loan application ID This left us with the following columns to explore and transform if needed:

## [1] "amount" "term" "rate" "payment" "employment"   
## [6] "length" "home" "income" "verified" "reason"   
## [11] "state" "debtIncRat" "delinq2yr" "inq6mth" "openAcc"   
## [16] "pubRec" "revolRatio" "totalAcc" "totalPaid" "totalBal"   
## [21] "totalRevLim" "accOpen24" "avgBal" "bcOpen" "bcRatio"   
## [26] "totalLim" "totalRevBal" "totalBcLim" "totalIlLim" "response"

## Section 4 - Exploring and Transforming the Data

Relative to data transformation, there were 7 types of data in the dataset.

* Data that was fine the way it was, no transformation required
* Data that needed to have unused factor levels removed
* Data that lent itself to a calculated or derived field
  + State was used to create a new column of Region; many of the states only had a few loans but I wanted to see if geography was a factor so I grouped the states by region
* Data stored as categorical with factor levels would be better represented as a continuous dimension
  + labels like “year 2” for a factor level has less information content than the number 2 in column year because of how 2 relates to 1 and 3
* Data that was skewed and needed a transformation
  + There were two primary transformations that were used the log base 10, and n-root transformations.
  + The n-root was used if the log transformation resulted in negative infinity values or it gave a better transformation as judged by the boxplot tails
  + The “n”" in the n-root was picked to balance the upper and lower trails of the boxplot.
* Data that was skewed but also did not lend itself to being transformed because of a very small range (0-6)
* Data that had too many factor levels to be of use
  + Employment had over 21k levels and was not going to be useful to the model.

Our data columns now look like this:

## [1] "amount" "term" "rate"   
## [4] "home" "verified" "reason"   
## [7] "state" "debtIncRat" "delinq2yr"   
## [10] "inq6mth" "openAcc" "pubRec"   
## [13] "revolRatio" "totalPaid" "totalRevLim"   
## [16] "avgBal" "bcRatio" "response"   
## [19] "logPayment" "years" "logIncome"   
## [22] "region" "logOpenAcc" "logTotalAcc"   
## [25] "rootTotalRevBal" "rootaccOpen24" "rootAvgBal"   
## [28] "rootbcOpen" "roottotalLim" "roottotalBal"   
## [31] "roottotalBcLim" "roottotalIlLim"

## Section 5 - The Logistic Selection for Model for Accuracy & Profit - Optimizing

The data was divided into a train and test dataset with 80% of the records going for training and 20% for testing. All of the model performance accuracy and profit statistics were obtained using the test dataset.

Eight logistic models were built and compared to one another to see which was the best at selecting loans to approve and reject. Two grading criteria were used.

1. Lowest model AIC
2. Highest accuracy at the 50% prediction value on the testing dataset

The first six models were built using the step function in the forward, backward and both directions. Two models were selected from each direction. One model that produced the best accuracy, and one model that produced the lowest AIC.

From these six models, the one produced by the backward and both directions were exactly the same so I discarded the two versions produced by the both direction.

Then I took the most accurate model and used it as the basis to create two second-order models, one for best accuracy and another one for lowest AIC.

Next I needed to see how the models compared to each other in approving loans to maximize profit. Assumptions: \* Approving Bad loan cost: Total paid - $500 – loan amount (this can be a positive number or a profit). \* Rejecting Good loan cost: Lost opportunity no monetary loss, the money is available to make another loan. \* Profit for loans made: Total paid – loan amount - $500 \* count(Bad loans)

Below is a tabular summary of the models comparing accuracy and profit when threshold is allowed to move, the percent improvement over the current process and the AIC and BIC.

Even the worst model produces 10 times better profit results than the current system does. Even after such a significant improvement, there is still more that can be done for profit maximization, the best model is far from perfect. It only captures 30% of the potential profit by granting bad loans that don’t provide more income than expenses and rejecting profitable loans.

To put the 30% into perspective, the current process was only capturing 2% of the available profit. The cost associated with making unprofitable bad loans negates significant profit.

Note: Max accuracy models don’t always have the highest accuracy when the threshold is allowed to move away from .5 they were selected with.

* fw means the model was created with the step method in the forward direction
* bw means the model was created with the step method in the backward direction
* sec means the model was created with the step method in both directions while looking for second order predictors
* Rows with ’\_acc’ indicate models with predictors chosen based on highest accuracy when the threshold cutoff point was held constant for all permutations of the step method creation process
* Rows with ’\_aic’ are the lowest AIC models created from the step method

## Acc Cutoff Max Acc Profit Cut Max Profit % Improved % Perfect  
## fw1\_acc: 31 0.785 70 2714052 1037.1 21.8   
## fw2\_aic: 46 0.793 77 3444349 1316.1 27.7   
## bw1\_acc: 47 0.794 74 3432764 1311.7 27.6   
## bw2\_aic: 44 0.793 77 3452207 1319.1 27.7   
## sec1\_acc: 47 0.794 75 3594744 1373.6 28.9   
## sec2\_aic: 54 0.796 73 3706857 1416.4 29.8   
## AIC BIC   
## fw1\_acc: 27114 27131  
## fw2\_aic: 26211 26508  
## bw1\_acc: 26210 26515  
## bw2\_aic: 26209 26497  
## sec1\_acc: 26084 26595  
## sec2\_aic: 25999 26921

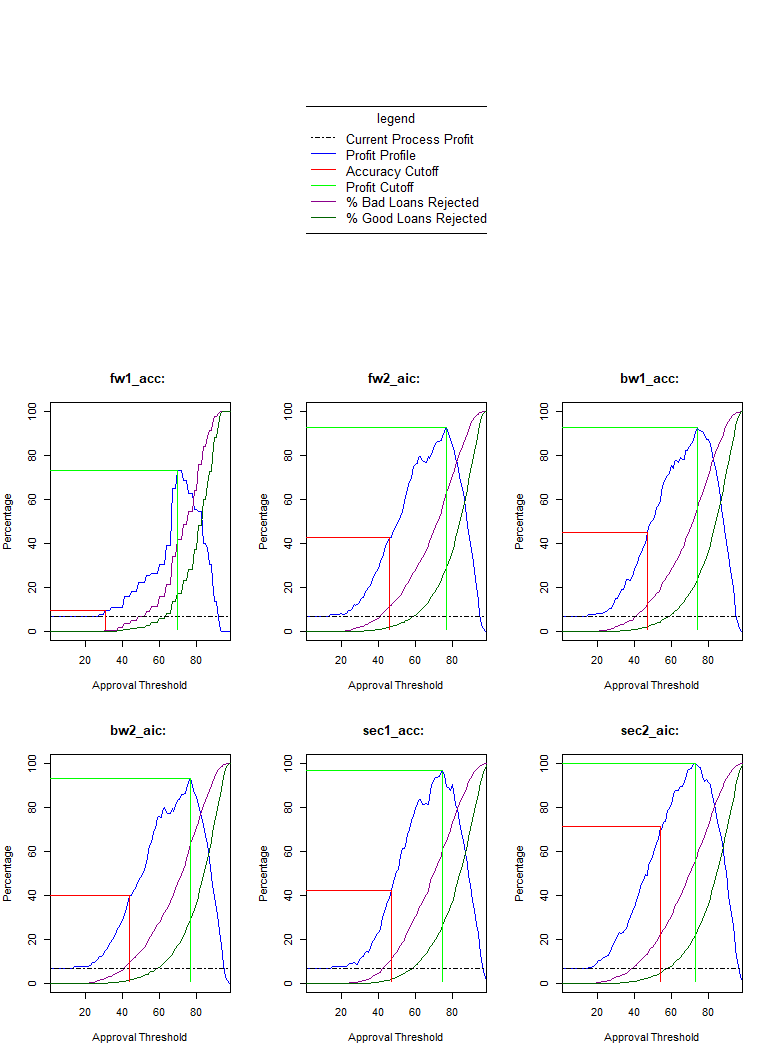
The following is a graphical representation of how profitability is affected by changing the algorithm approval threshold. The graphs also shows the algorithm approval threshold to achieve the highest accuracy.

Note: The highest accuracy is lower than the highest profit threshold for all of these models. Approving a few additional bad loans is made up by the profit from the additional good loans that are made.

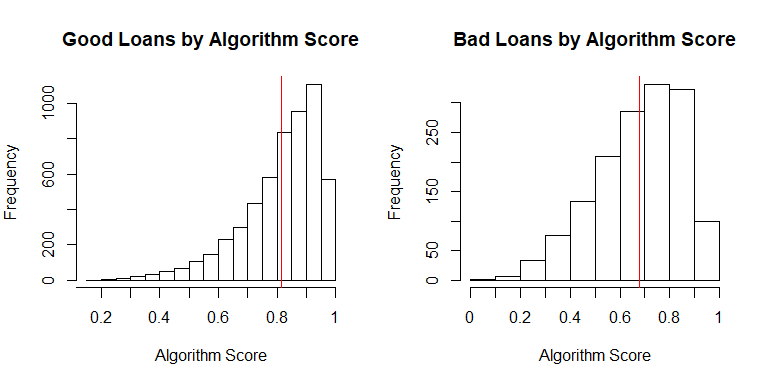
All graph y-axes are scaled so that 100% represents maximum profit for the most profitable model. This scaling allows the viewer to compare the profitability of each model to the best in class.

The current process profit for the test dataset is 7% of the best model max profit, and matches the result of all models if we set the threshold to approve 100% of the loans.

### @@@@ Model Profitability vs Algorithm Approval Threshold @@@@



Finally, here are two histograms showing the distribution of both good and bad loans versus algorithm score, with the mean value indicated by the red vertical line. Notice that even at very high threshold cutoff values there are still some bad loans that get approved.



## Section 6 - Results Summary

Where should we place the cutoff and how will that impact the bank?

To maximize return on loaned money, we would want to place the automatic cutoff level at 0.73.

For this portfolio of loans, that would represent an accuracy of 73% and a total profit of $3.71M which is an 1416% increase or an extra $3M in profit over what this portfolio of loans actually produced for the bank.

At the proposed threshold, the percent of bad loans rejected is 56%, and the percent of good loans accepted is 78%.

While this is a significant improvement there is still room to get better, a perfect model that approved 100% of good loans and rejected 100% of bad loans would outperform this model by 336% or, to put another way, would produce an additional $8.74M of profit.