PROJECT 2 – GIVE ME SOME CREDIT

Scott Mitchell

Executive Summary

For my final project I chose to model data from a completed Kaggle competition called ``Give Me Some Credit.'' Link: https://www.kaggle.com/c/GiveMeSomeCredit. The goal of this competition was to predict whether or not a person will experience severe payment delinquency on outstanding debt. The competition was judged on AUC. For my project I decided to use Decision Tree, Bagging, Random Forest, and Gradient Boosting to make my predictions and compare the results of each.

The final outcomes of my modeling attempts on the Kaggle submission data were as follows:

PREDICTION METHOD	AUC (TEST ON TRAINING DATA)	AUC (KAGGLE TEST DATA)	KAGGLE RK
GRADIENT BOOSTING	0.8548	0.8550	618
RANDOM FOREST	0.8478	0.8533	636
BAGGING	0.8353	0.8387	697
DECISION TREE	0.7874	0.7874	780

I was pleased with my attempts as the best Kaggle AUC was .8696 which was less than 1.5 percentage points better than my attempt.

Project Learnings

Though getting the various prediction algorithms to run on my data was certainly a challenge unto itself I had a couple of other important takeaways from doing the project. The biggest lesson I learned was how to deal with binary prediction outputs where one of the prediction options occurs infrequently. For instance, in the 150,000 rows of data for training only 6.7% indicated that a person had been seriously delinquent on payments. As will be shown this low positive response rate can make classification predictions difficult because models will tend to almost never predict a positive response. This is why a probability prediction using regression is more appropriate in this situation. The low positive response rate also made the EDA difficult as it was hard to visually discern whether or not a particular sub-segment of predictors was producing more or fewer positive responses. Another important lesson I learned was that cleaning data is not only identifying NA values. In reviewing the data I was able to see that some predictor entries where data was present did not conform to reasonable expectations given the parameters of the predictors - you could say that data appeared to be “wrong.” Understanding what & where those non-conforming values can aid significantly in the predictive modeling process.

Cleaning the Data

Here are some of the observations and decisions made during the cleaning process:

- Age - All observations were 21 or more except for one that was 0. Assume 0 is actually NA and code as 999.
- Monthly Income - Value is missing 20% of the time. Code missing values as 99999999.
- Debt Ratio - This column is dependent on monthly income, but in some cases monthly income is missing where debt ratio is provided. In some instances where Monthly Income is NA, debt ratio is 0. Does this truly indicate debt ratio is 0 or does it mean debt ratio is missing? Decision to code debt ratio as 999999.99 wherever Monthly Income is missing.
- Number of Dependents - Value is missing 2.6% of the time.
- Number of Times Past Due (3 Predictors) - There is a large gap in the data values in all 3 Past Due categories. For instance, in the "Number of Times 60-89 Days Past Due" there are no observations after 12 until 98 where there are 264 observations. Thus, it is highly likely that 98 holds a different meaning than 98 times past due. For instance, 98 may indicate an NA value.

The decision to handle NA values in each category by coding them with a specific numeric value distinct from any other value in that column was made knowing that MLR and Logistic Regression were not being used. It was determined that giving the NA values their own distinct category was potentially more useful as a tree splitting mechanism than trying to guess at actual values.

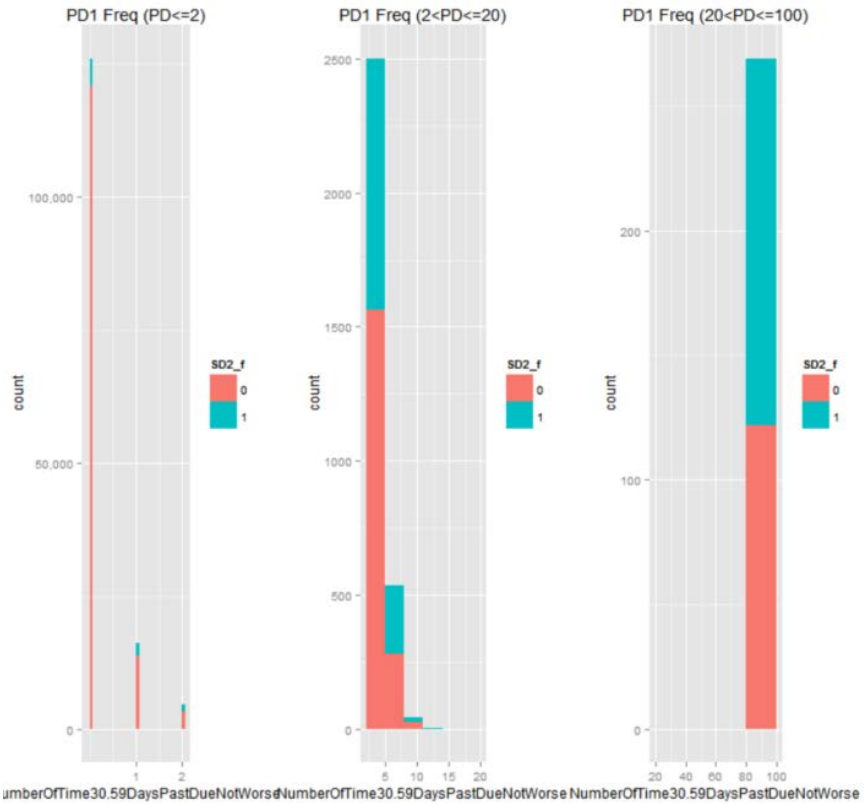
Exploratory Data Analysis

The most difficult challenge with EDA for an infrequent positive response on a binary variable was being able to visually identify where a positive response was relatively more frequent. For instance, if, on average, there is a 6.7% chance of a positive response but for a particular segment of the data the response rate rises to 9% this is a 34% increase, which is material. However, discerning this rise from visual methods can be tricky. As such my main method of visually exploring the data was to create histograms for the predictors and insert relevant breaks in the predictor values and color them with the delinquency response.

Here are some of the more interesting charts:

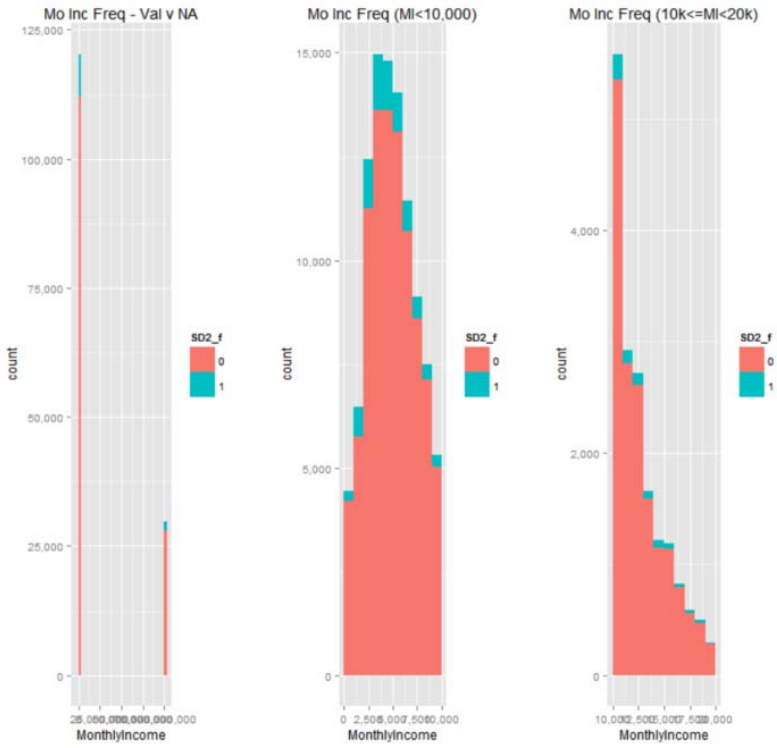
<u>Number of Times 30 to 59 Days Past Due</u>

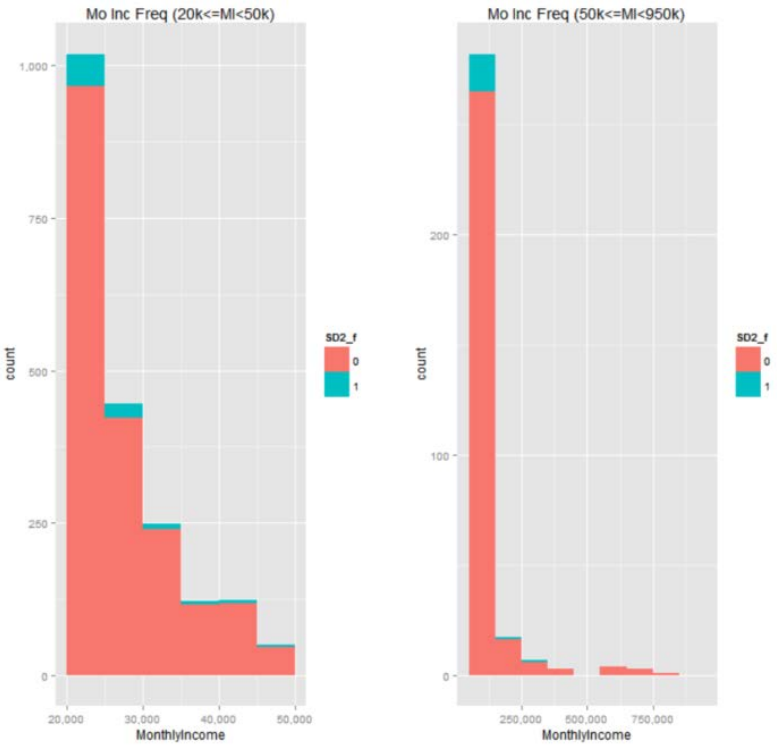
The below chart shows that delinquency rises with number of times past due.



Monthly Income

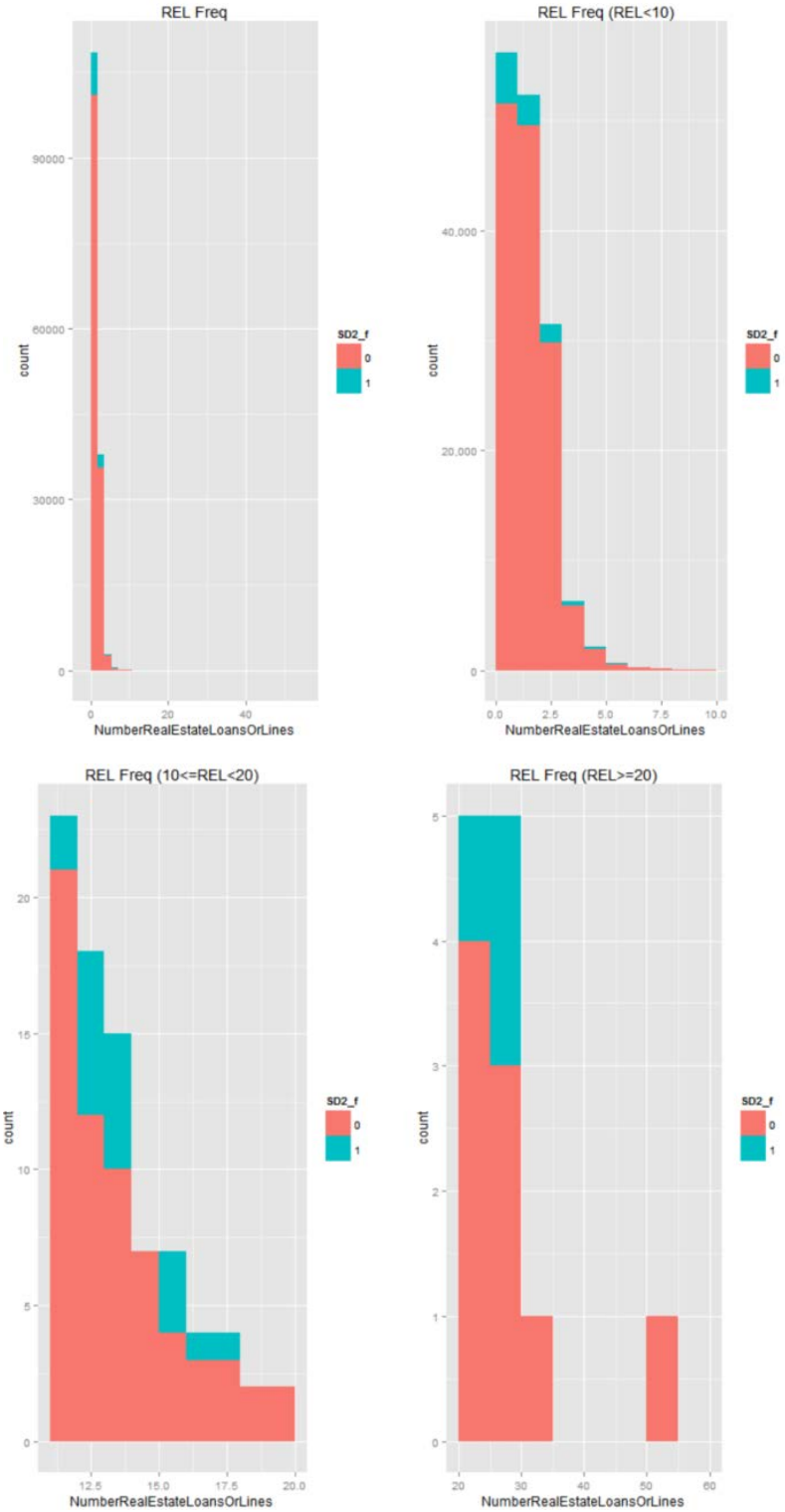
I was anticipating that the delinquency rate would fall sharply with increasing monthly income but it is hard to discern any significant change in delinquency rates across monthly income intervals until the largest values.





Number of Real Estate Loans

It is interesting here to note that have 0 real estate loans appears to have a higher delinquency rate than 1 to 10.



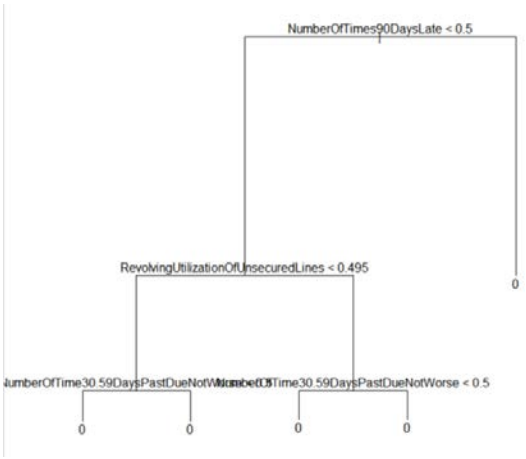
Prediction Models

Tree Models

It was interesting to compare the regression tree results to a classification tree (see below). With the infrequent positive response rate, the classification tree simply predicts 0 for all values. This is not particularly useful.

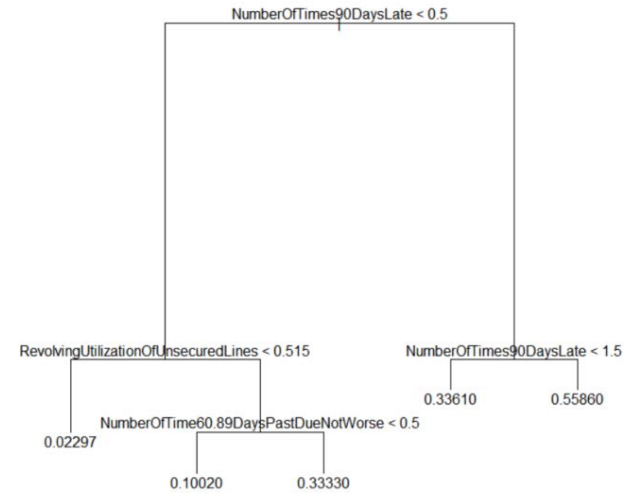
Classification Tree

```
Classification tree:
tree(formula = SD2_f ~ . - SeriousDlqin2yrs, data = train_cs_data)
Variables actually used in tree construction:
[1] "NumberOfTimes90DaysLate"              "RevolvingUtilizationOfUnsecuredLines" "NumberOfTime30.59DaysPastDueNotWorse"
Number of terminal nodes:  5
Residual mean deviance:  0.3822 = 28660 / 75000
Misclassification error rate: 0.06668 = 5001 / 75000
```



Regression Tree

```
Regression tree:
tree(formula = SeriousDlqin2yrs ~ . - SD2_f, data = train_cs_data)
Variables actually used in tree construction:
[1] "NumberOfTimes90DaysLate"              "RevolvingUtilizationOfUnsecuredLines" "NumberOfTime60.89DaysPastDueNotWorse"
Number of terminal nodes:  5
Residual mean deviance:  0.05184 = 3888 / 75000
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.55860 -0.02297 -0.02297  0.00000 -0.02297  0.97700
```



Bagging Model

```
> print(bag_SD2)

Call:
 randomForest(formula = SeriousDlqin2yrs ~ ., data = train_cs_data_2,      mtry = 11, importance = TRUE, ntree = 250)
               Type of random forest: regression
                     Number of trees: 250
No. of variables tried at each split: 10

          Mean of squared residuals: 0.05190855
                    % Var explained: 16.59
> round(importance(bag_SD2),2)
                                      %IncMSE IncNodePurity
RevolvingUtilizationOfUnsecuredLines   64.52        638.46
age                                    46.49        609.67
NumberOfTime30.59DaysPastDueNotWorse   51.28        184.94
DebtRatio                              47.03        538.64
MonthlyIncome                          43.21        695.54
NumberOfOpenCreditLinesAndLoans        69.91        401.98
NumberOfTimes90DaysLate                86.81        608.83
NumberRealEstateLoansOrLines           45.19        149.61
NumberOfTime60.89DaysPastDueNotWorse   64.70        127.94
NumberOfDependents                     31.74        193.61
```

Random Forest Model

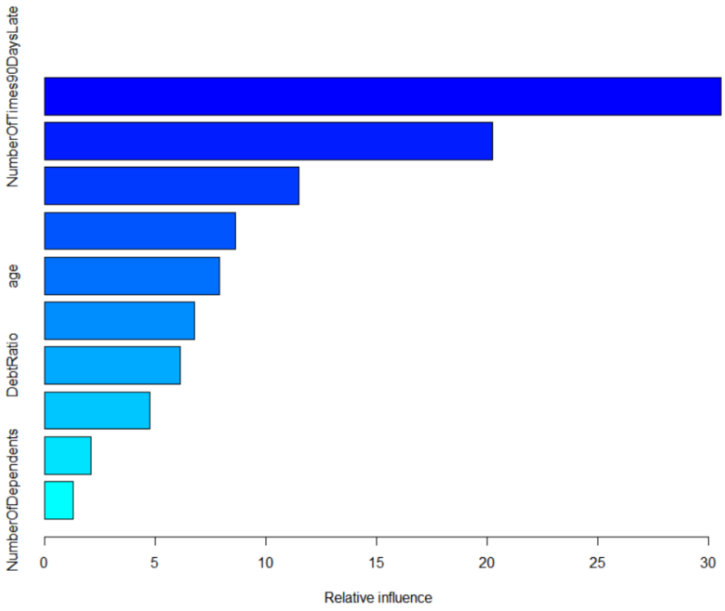
```
> print(rf_SD2)

Call:
 randomForest(formula = SeriousDlqin2yrs ~ ., data = train_cs_data_2,      mtry = sqrt(10), importance = TRUE, ntree = 250)
               Type of random forest: regression
                     Number of trees: 250
No. of variables tried at each split: 3

          Mean of squared residuals: 0.05011239
                    % Var explained: 19.48
> round(importance(rf_SD2),2)
                                      %IncMSE IncNodePurity
RevolvingUtilizationOfUnsecuredLines   52.34        615.68
age                                    34.42        480.17
NumberOfTime30.59DaysPastDueNotWorse   46.97        236.94
DebtRatio                              28.24        450.62
MonthlyIncome                          29.21        533.58
NumberOfOpenCreditLinesAndLoans        54.69        352.83
NumberOfTimes90DaysLate                81.27        433.25
NumberRealEstateLoansOrLines           34.09        145.45
NumberOfTime60.89DaysPastDueNotWorse   62.43        224.00
NumberOfDependents                     19.26        170.07
```

Gradient Boosting Model

```
> print(gbm_SD2)
gbm(formula = SeriousDlqin2yrs ~ ., distribution = "bernoulli",
    data = train_cs_data_2, n.trees = 250, interaction.depth = 5,
    shrinkage = 0.2)
A gradient boosted model with bernoulli loss function.
250 iterations were performed.
There were 10 predictors of which 10 had non-zero influence.
> summary(gbm_SD2)
                                                                 var   rel.inf
NumberOfTimes90DaysLate                           NumberOfTimes90DaysLate 30.586914
RevolvingUtilizationOfUnsecuredLines RevolvingUtilizationOfUnsecuredLines 20.248934
NumberOfTime30.59DaysPastDueNotWorse NumberOfTime30.59DaysPastDueNotWorse 11.495113
NumberOfTime60.89DaysPastDueNotWorse NumberOfTime60.89DaysPastDueNotWorse  8.637406
age                                                                  age  7.902176
MonthlyIncome                                                MonthlyIncome  6.789796
DebtRatio                                                        DebtRatio  6.150806
NumberOfOpenCreditLinesAndLoans            NumberOfOpenCreditLinesAndLoans  4.786181
NumberRealEstateLoansOrLines                  NumberRealEstateLoansOrLines  2.102649
NumberOfDependents                                      NumberOfDependents  1.300023
```



Predictions

Regression Tree

Test Results on the Training Data

```
> summary(reg_tree_pred_SD2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.02297 0.02297 0.02297 0.06649 0.10020 0.55860
```

```
> auc(test_cs_data$SeriousDlqin2yrs,reg_tree_pred_SD2)
Area under the curve: 0.7874
```

Results on Kaggle Test Data

```
> summary(kag_reg_tree_pred_SD2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.02297 0.02297 0.02297 0.06719 0.10020 0.55860
```

Kaggle AUC = .7874

Bagging

Test Results on the Training Data

```
> summary(bagging_pred_SD2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.00240 0.01980 0.07495 0.07813 0.99550
```

```
> auc(test_cs_data_2$SeriousDlqin2yrs, bagging_pred_SD2)
Area under the curve: 0.8353
```

Results on Kaggle Test Data

```
> summary(kag_bagging_pred_SD2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.00240 0.01960 0.07494 0.07820 0.99310
```

Kaggle AUC = .8353

Random Forest

Test Results on Training Data

```
> summary(rf_pred_SD2)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
0.0003033 0.0052100 0.0206800 0.0700900 0.0711500 0.9022000
> auc(test_cs_data_2$SeriousDlqin2yrs, rf_pred_SD2)
Area under the curve: 0.8478
```

Results on Kaggle Test Data

```
> summary(kag_rf_pred_SD2)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
0.0002045 0.0051530 0.0205300 0.0702000 0.0711700 0.9022000
```

Kaggle AUC = .8478

Gradient Boosting

Test Results on Training Data

```
> summary(gbm_prob_pred_SD2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.01935 0.01935 0.02295 0.06825 0.06205 0.81430
```

```
> auc(test_cs_data_2$SeriousDlqin2yrs, gbm_prob_pred_SD2)
Area under the curve: 0.8548
```

Results on Kaggle Test Data

```
> summary(kag_gbm_pred_SD2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.01935 0.01935 0.02295 0.06837 0.06205 0.81430
```

Kaggle AUC = .8550