Classification for Credit Defaulting

By Wabi Mposo

Executive summary

I decided to do a classification on predicting payment defaulting based on 26 factors originally. After inspecting the data, I transformed the data into a time series for a better data structure. This modified format changed the data dimension from 30,000 to 180,000 observations and 26 to 13 predictors. I ran three classifications models (logistic regression, randomforest, boosting), and the best one was randomforest, which achieved a 18% test error rate.

As I ran each regression model, it's worth to note the following:

- Overall bill_period and pay_period had zero influence on the percentage shift in the likelihood of payment defaulting for next month.
- As time in months of unpaid monthly credit bill elapsed for a particular bill amount, the probability of defaulting increased as well. Interestingly though, people with higher *limit_bal* (total credit balance), bill -amt and pay_amt had lower probability of default. This is consistent with the hypothesis that these people are probably wealthy.
- All three models performed better on the test data than on the validation test, but the margin was probably insignificant. However, to check that the margin was insignificant, that is due to random chance, was out of the scope of this study.

While I was happy about the performance of each model on average, I could had achieved even lower validation, test error if I had more time. I trialed with each model thrice and determined the best parameters based on educated guess from prior exposure to classification. In summary, with more time, I would have segmented the data into two segments: one for college and grad students and another for others. It seemed that the likelihood of defaulting should behave different given the other categories are the ones with the most financial power. An in-depth analysis for this classification will continue sometime soon.

Introduction

Consumer credit makes up a sizeable portion of the US credit market. Since the introduction of banks, consumer have used banks to finance their business and personal needs in the form of credit. What makes consumer credit so important is that virtually everyone in the USA relies on credit to make some investments. Investment made via credit is usually tax-deductible, and it comes with flexible repayment options. In general, credit borrowing is encouraged by the financial institutions, and it is healthy for the economy. There's however a concern of credit defaulting that banks deal must deal with. Credit defaulting hurts the economy and, in fact, is one of the leading causes of economic recessions. My purpose in not to determine factors that cause credit defaulting but to predict credit defaulting using the factors in the dataset.

Data for this project was gathered from UCI archives. Three classification methods were chosen to predict credit defaulting for next month payments. They were logistic regression, randomforest and boosting. My data set had many observations compared to the number of predictors especially since I decided to transform the file into a time series format from its original format.

Model Building

To build the model, we transformed the data into a time series format from its original format. The transformed model had 180,000 observations and reduced to 12 predictors. I divided the model into a training, validation (40,000) and testing (40,000) set. I included a validation, testing set to measure the reliability of our model. That is, we averaged the error rate from the validation and testing set to obtain mean error rate. Using each individual error rate (validation and testing, we calculated the variance of our error rates). We expect a strong prediction to have less varying error rate across multiple unseen data.

Model 1

I began my prediction modelling with a logistic regression. I used the logistic regression for predictive power but also for inferential learning about the factors in my data and how they are potential related to credit defaulting. I started with a regression model where all the variables were included. I regressed "default_nm" on all the variables. This yielded the following:

```
Coefficients: (10 not defined because of singularities)
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -8.758e-01  5.313e-02 -16.485  < 2e-16 ***
limit_bal               -2.693e-06  8.659e-08 -31.098  < 2e-16 ***
sexmale                  1.527e-01  1.660e-02   9.201  < 2e-16 ***
educhigh school          1.093e-02  2.564e-02   0.426 0.669813
educothers              -1.036e+00  9.781e-02 -10.588  < 2e-16 ***
educuniversity           2.365e-02  1.924e-02   1.229 0.218990
marriageothers          -2.247e-01  7.239e-02  -3.104 0.001906 **
marriagesingle          -2.199e-01  1.873e-02 -11.745  < 2e-16 ***
age                      3.159e-03  1.009e-03   3.130 0.001746 **
repay_periodrepay_0505  -1.797e-03  2.783e-02  -0.065 0.948521
repay_periodrepay_0605  -4.861e-02  2.792e-02  -1.741 0.081658 .
repay_periodrepay_0705  -9.610e-02  2.792e-02  -3.442 0.000577 ***
repay_periodrepay_0805  -1.238e-01  2.817e-02  -4.397 1.10e-05 ***
repay_periodrepay_0905  -1.503e-01  2.940e-02  -5.112 3.19e-07 ***
statuspay duly          -2.045e-01  2.991e-02  -6.837 8.07e-12 ***
statusrevolving credit  -3.777e-01  2.817e-02 -13.408  < 2e-16 ***
status1M-delay           6.712e-01  5.644e-02  11.892  < 2e-16 ***
status2M-delay           1.332e+00  3.248e-02  41.002  < 2e-16 ***
status3M-delay           1.573e+00  7.851e-02  20.037  < 2e-16 ***
status4M-delay           1.383e+00  1.354e-01  10.213  < 2e-16 ***
status5M-delay           1.255e+00  2.318e-01   5.413 6.21e-08 ***
status6M-11M-delay       2.054e+00  1.721e-01  11.936  < 2e-16 ***
pay_periodpay_amt2              NA         NA      NA       NA
pay_periodpay_amt3              NA         NA      NA       NA
pay_periodpay_amt4              NA         NA      NA       NA
pay_periodpay_amt5              NA         NA      NA       NA
pay_periodpay_amt6              NA         NA      NA       NA
pay_amt                 -1.042e-05  9.884e-07 -10.541  < 2e-16 ***
bill_periodbill_amt2            NA         NA      NA       NA
bill_periodbill_amt3            NA         NA      NA       NA
bill_periodbill_amt4            NA         NA      NA       NA
bill_periodbill_amt5            NA         NA      NA       NA
bill_periodbill_amt6            NA         NA      NA       NA
bill_amt                 1.742e-06  1.577e-07  11.045  < 2e-16 ***
```

I took from this that pay, bill period was not significant. Therefore, I generated prediction using all the predictors and compared to the model with pay, bill period excluded. Surprisingly, it seemed that the original model with all the predictors performed better across of the errors (training, validation and test). Below are the results:

All p predictors included	(p-2) predictors included

`cred_df_pred current default`
`current 29891 7115`
`default 1255 1739`
`Validation error rate: 0.209625`
	`cred_df_pred current default`
	`current 29891 7115`
	`default 1255 1739`
	`validation error rate: 0.2651`

`cred_df_pred current default`
`current 29937 7036`
`default 1222 1805`
`Test error rate: 0.20645`
	`cred_df_pred current default`
	`current 29937 7036`
	`default 1222 180`
	`Test error rate: 0.20645`

Error variance: 0.0008308505	Error variance: 0.04147181

In general, this logistic regression could only give us between 73 % to 80 % accuracy using “0.55” or greater as the cutoff point for defaulting (this cutoff point was used on classification models). If more time was allowed, then one interesting thing would be to find the optimal cutoff point to which the data seem to give the lowest validation or test error.

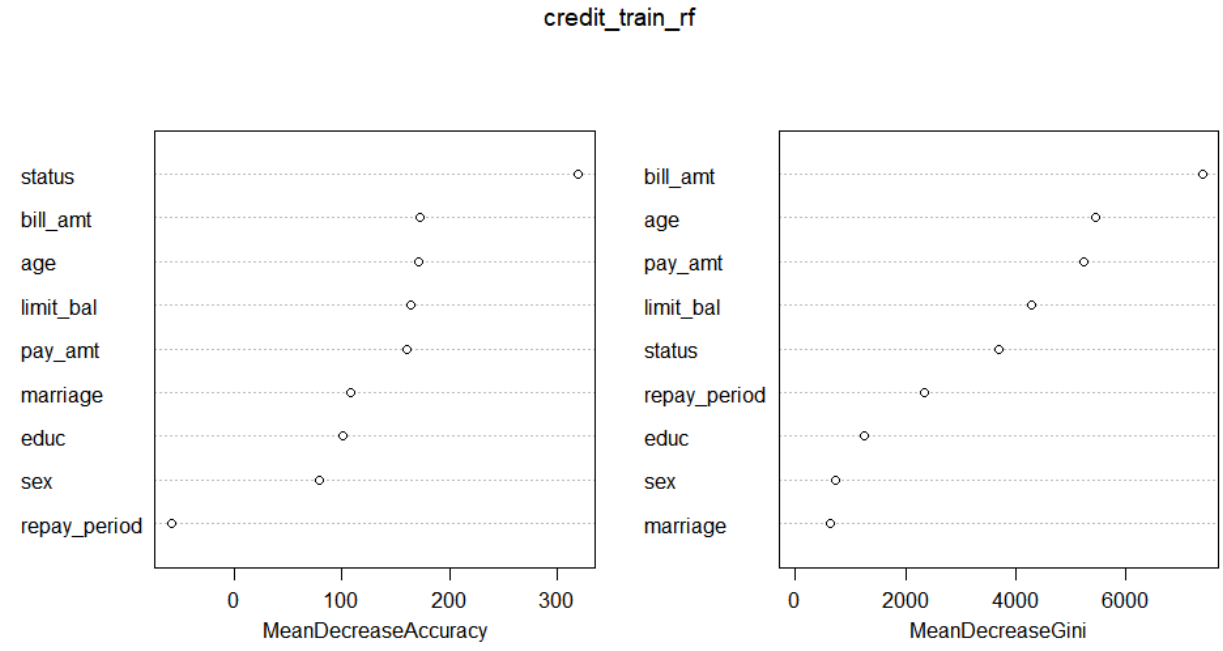
Model 2 (RandomForest)

I tried randomforest for a second classification model. Overall randomforest performed better than logistic regression and boosting. I was able to obtain a test error rate in the range of 18%. The greatest improve in the model occurred when I excluded “bill_period and “pay_period”, which were again classified as not important according to the importance function and importance plot. The following are the results from the best randomforest model. Other randomforest models resemble this format, except that they had a higher error-rate.

Validation Data	Test Data
credit_predict 0 1 0 29763 6224 1 1360 2653	credit_predict 0 1 0 29876 6178 1 1283 2663
Validation error: 0.1896	Test error: 0.186525

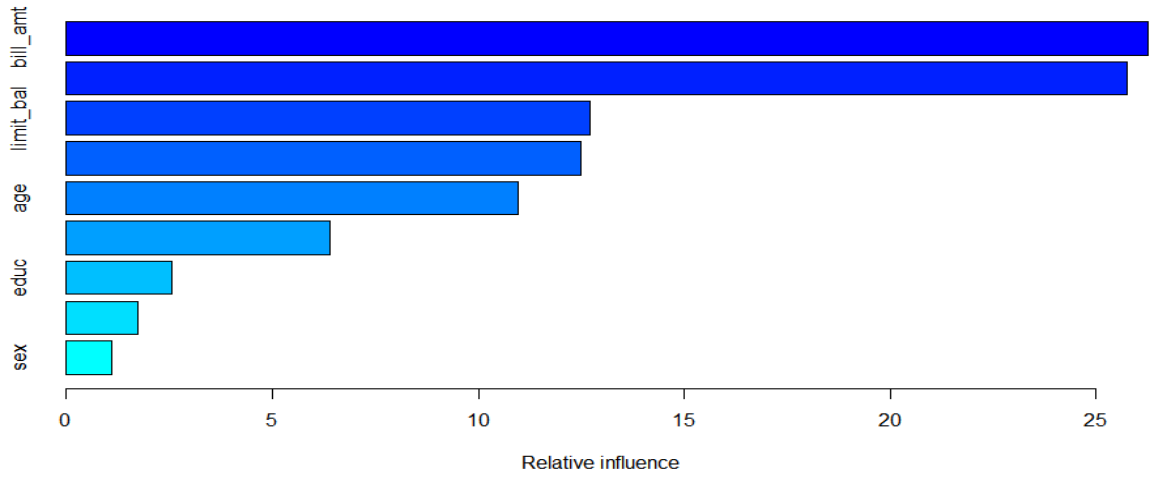
Variables ranked by importance

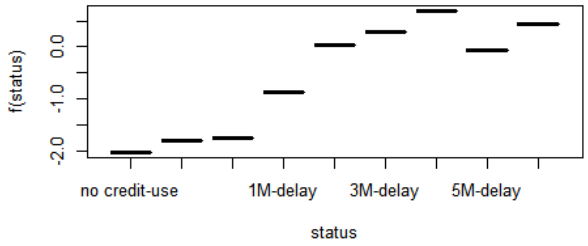
	0	1	MeanDecreaseAccuracy	MeanDecreaseGini
limit_bal	103.79003	147.45611	163.56621	4293.3584
sex	65.66066	69.67400	78.60128	729.8467
educ	81.53677	78.50610	100.87529	1248.2361
marriage	96.26191	69.87843	107.42161	644.3210
age	144.74271	114.73508	171.01829	5440.0215
repay_period	-27.87560	-66.65631	-59.01032	2343.6740
status	167.39955	274.39509	319.98240	3700.2501
pay_amt	116.11523	95.33494	160.71563	5224.5839
bill_amt	101.75466	117.62360	172.01709	7382.2947

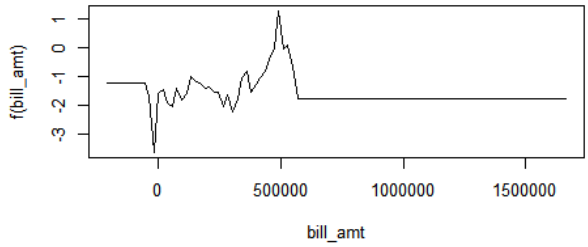


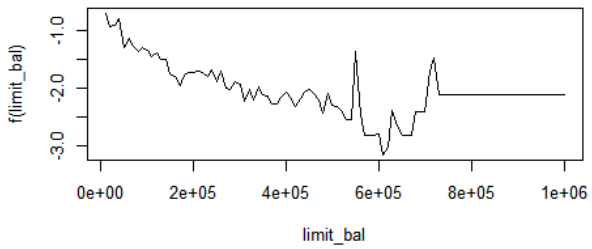
It is clear that limit_balance and sex are two most important variables. This is somewhat unexpected as one would normally think of status as being important. However, if one consider men and women borrowing and spending habits, then it makes sense that sex is an important predictor along of payment defaulting.

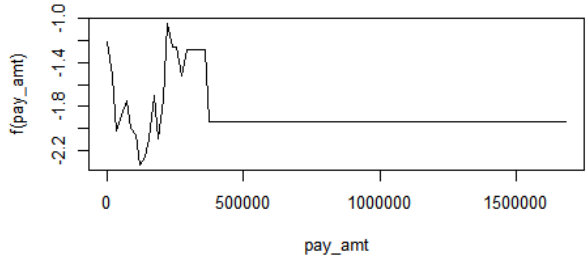
Model 3(Boosting)











Validation set	Test set

gbm.class 0 1 no 30026 6986 yes 1097 1891	gbm.class 0 1 no 30078 6910 yes 1081 1931
Validation error: 0.202075	Test error: 0.199775

I ran three boosting models, and all of them could not beat the benchmark established by randomforest of 18% error-rate. I selected the tuning parameters at random because choosing the optimal tuning parameters via train-control showed models with lower accuracy than what I was aiming for (80% or more). Playing with different tuning parameters on my own did not get me to outperform the benchmark set by randomforest, however, it allowed me to outperform the logistic regression. The average error-rate was upper 19% for different models. They seemed to all perform slightly better on the test set then the validation set. From the above plots, we can see that bill_amt and limit_balance were the most influential predictors. As with other models I used for classification, bill_period and pay_period had zero influence on the likelihood of default while other variables, however small or large the contribution, had non-zero influence on the likelihood of default. The four separate plotted graph an interesting story; however, I somewhat expected it from the result of the EDA. In summary, the repay status increased in time; that is, the number of unpaid months increased for a specific billing period, then percentage increased in the likelihood of defaulting also increased. However, there was a small drop in the curved for the 5-month unpaid status, which was confusing. I would expect the graph to have a diminish return shape. In line with the results from the EDA, the higher the limit balance correlated with a percentage decrease in the probability of defaulting. Bill, pay amount had a similar pattern. It is interesting to note that the pattern turned constant right after the $500,000 mark. In short, I concluded from this that higher bill amount resulted in negative percentage increase in the likelihood of defaulting since people with higher billing amount probably had higher credit-balance, which in turn meant that they were kind of wealthy.

Conclusion

For this classification project, I utilized three classification methods, which were a logistic regression, a randomforest and boosting. These three models suggested that bill period and pay period had zero influence in the percentage increase or decrease in the likelihood of defaulting. Other variables had non-zero influence. No one variable consistently ranked at the top, but *status*, *limit_bal* , *sex* and *pay_amt* were of large influence on the percentage swing in the probability of defaulting. I only managed to reach

18% from the three models with randomforest outperforming the remaining models. The winning model was the model that excluded the zero-influencing variables.

randomForest(default_nm~., data=credit_train[,-c(13,9,11)], mtyr=3, importance = TRUE)

However, I must point out that time was a factor in not achieving lower prediction error on the validation and test set. I am satisfied with the model given the time constraint and a 80% accuracy.

What I could have done better

- With more time, I would train more with the data to determine which cutoff value results in the optimal prediction accuracy for the logistic regression and boosting classification. I had chosen 0.55 because in 2005 the likelihood of defaulting was low because the economy was going well. Even though there was an easy access to credit liquidity, no one knew that the great recession was around the corner. However, I do feel that I could have selected the best cutoff point based on the which ever returned the lowest validation and test error.
- I would have tried more turning parameters to get the best boosting and randomforest model; however, I could only choose three. I chose them mainly at random due to my prior exposure to other classification problems.
- I would have segmented the data college and graduate students as one segment and the rest of the categories as the other segments. I think that the different segment might tell a different story especially since the other class seemed to be the class with more financial power.