Predicting Loan Defaults with Logistic Regression

Sanzida Parvin

Executive Summary

Problem

It is very important for a bank to screen the potential good or bad clients for sanctioning any loan. The clients who will repay their loans in a timely manner are treated as good clients as they make profit for the bank. On the other hand, the clients who are not able to repay their loans can cause a big loss for the bank. Therefore, it is very crucial for any financial organization to identify the clients and predicts the loan defaults. In this study, logistic regression technique was implimented to predict the loan defaults.

Method

The study was executed in three steps: (i) Cleaning the raw data for the logistic regression analysis, (ii) Model selection using statistical tools and (iii) Profit calculation using different threshold levels.Briefly, after collecting the raw data, Some features engineering was done to reduce the unnecessary and redundant explanatory variables for preparing a cleaned version of the dataset. However, there were some missing values in the cleaned dataset. If any fields of values are empty or have some unusual values, then those are called missing values. Missing values should be treated in a proper way before running any analysis. For this analysis the missing values were imputed using statistical tools named ‘mice’. Finally got the cleaned final version if the input dataset which consists of 49,999 observations and 29 variables.After that some statistical analysis were performed to find out a suitable and effective model by which the good and bad loan applicants can be identified and the profit of the bank can be calculated based on the predicted good loans. The cleaned and final version of the input dataset was divided into two parts, test dataset and training dataset. Training dataset was used to get the suitable method,

and the test dataset was applied on that method to check the accuracy. Finally the profit was calculated by varying the thereshold values.

Results

Statistical calculations showed the overall accuracy level of the model, correctly predicted good and bad loans percentage and calculated the profit range based on the predicted good loans. By changing the threshold levels, different profits were calculated to identify the best threshold level that can make the highest profit with high accuracy and low percentage of incorrect predictions. The following table represents the profit of the predicted good loans as well as the different threshold levels for the good and bad loans.

```
##    Threshold TotalProfit Good Loans % Bad Loans %
## A       0.10  3023265.06        99.89        0.94
## B       0.20  3804192.20        97.78       10.67
## C       0.30  4353921.92        90.97       28.05
## D       0.40  4349920.54        80.65       47.32
## E       0.50  3611367.20        64.89       64.97
## F       0.60  2728642.18        48.40       79.73
## G       0.70  1779788.48        30.72       90.67
## H       0.80   596516.07        10.09       98.19
## I       0.90    63720.37         0.72      100.00
```

Introduction

The dataset ‘loans50k’ contains information about the loan applicants of banks. There are 50,000 loan applicants and 30 variables that contains different informations about the applicants in the dataset. The purpose of this project is to predict which applicants are likely to default on their loans using logistic regression. The given dataset was in a raw format and needs to clean and prepare it for the statistical analysis. The first approach of preparing the dataset is to look on it, remove any irrelevant rows and variables that has no effect on the prediction of default loans.

For instance, id, state as well as the rows that has no income were removed to clean the dataset prior to analysis.

Load Data

```
loans = read.csv("loans50k.csv")
#head(loans)
```

Preparing and Exploring the Data

The response variable ‘loan_status’ was prepared according to the instruction of the project. Then some feature engineering has done to reduce the categories inside a variable to make it more clear and do the error free analysis. The feature engineering includes,

1. New variables have created with reduced categories and the old one has removed, so that there would be no duplicate columns in the final dataset.

2. There also has some redundancy in variables like, ‘totalAcc’ and ‘openAcc’. ‘totalAcc’ indicates total number of credit lines in the file, which includes both open and closed accounts but the variable ‘openAcc’ is the number of open credit lines. Subtract the ‘openAcc’ from ‘totalAcc’ to get the number of closed accounts, which shows by the variable ‘closeAcc’ in the cleaned dataset. By doing this, the information will be more clear.

3. Another redundancy was ‘totalBcLim’(total credit limits of credit cards) which already includes in ‘totalRevLim’(sum of credit limits from all credit lines).

After doing those feature engineering, a much clear dataset was prepared. By exploring the summary of the ‘filtered_dataset’, found some missing values in the variables ‘revolRatio’, ‘bcOpen’ and ‘bcRatio’. The count of missing values of ‘bcOpen’ and ‘bcRatio’ was more than 300, so removing all the missing values could bias the result. To deal with the missing values here used the ‘mice’ package to impute them. Finally got the final cleaned version of the dataset ‘cleaned_dataset’ to apply the statistical tools for default loan analysis.

```
summary(filtered_dataset)
```

```
##      amount             term           rate          payment
##  Min.   : 1000           :    0   Min.   :0.0500   Min.   :  30.54
##  1st Qu.: 8000    36 months:25771   1st Qu.:0.1000   1st Qu.: 258.71
##  Median :12600    60 months: 8884   Median :0.1300   Median : 381.04
##  Mean   :14621                      Mean   :0.1318   Mean   : 440.96
##  3rd Qu.:20000                      3rd Qu.:0.1600   3rd Qu.: 578.68
##  Max.   :35000                      Max.   :0.2900   Max.   :1407.01
##
##       home           income                 verified
##         :    0   Min.   :   4000                :    0
##  MORTGAGE:17238   1st Qu.:  45500   Not Verified   :10236
##  OWN     : 3573   Median :  65000   Source Verified:14317
##  RENT    :13844   Mean   :  75221   Verified       :10102
##                   3rd Qu.:  90000
##                   Max.   :7446395
##
##                reason        debtIncRat      delinq2yr
##  debt_consolidation:21046   Min.   : 0.00   Min.   : 0.0000
##  credit_card       : 7843   1st Qu.:12.12   1st Qu.: 0.0000
##  home_improvement  : 2021   Median :17.87   Median : 0.0000
##  other             : 1571   Mean   :18.39   Mean   : 0.3367
##  major_purchase    :  619   3rd Qu.:24.30   3rd Qu.: 0.0000
##  medical           :  378   Max.   :61.21   Max.   :15.0000
##  (Other)           : 1177
##     inq6mth          openAcc          pubRec          revolRatio
##  Min.   :0.0000   Min.   : 1.00   Min.   : 0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.: 8.00   1st Qu.: 0.0000   1st Qu.:0.3600
##  Median :0.0000   Median :11.00   Median : 0.0000   Median :0.5400
##  Mean   :0.6968   Mean   :11.77   Mean   : 0.2438   Mean   :0.5372
##  3rd Qu.:1.0000   3rd Qu.:14.00   3rd Qu.: 0.0000   3rd Qu.:0.7200
##  Max.   :6.0000   Max.   :76.00   Max.   :19.0000   Max.   :1.2900
```

```
##                                     NA's   :15
##   totalPaid       totalBal       totalRevLim       accOpen24
## Min.   :    0   Min.   :     0   Min.   :     0   Min.   : 0.000
## 1st Qu.: 7081   1st Qu.:  29722   1st Qu.:  13500   1st Qu.: 3.000
## Median :12258   Median :  78579   Median :  23100   Median : 4.000
## Mean   :14812   Mean   : 137713   Mean   :  31158   Mean   : 4.742
## 3rd Qu.:20458   3rd Qu.: 207621   3rd Qu.:  38600   3rd Qu.: 6.000
## Max.   :58697   Max.   :2487317   Max.   :1060800   Max.   :50.000
##
##     avgBal          bcOpen         bcRatio          totalLim
## Min.   :    0   Min.   :    0   Min.   :  0.00   Min.   :    700
## 1st Qu.:  3110   1st Qu.:  1212   1st Qu.: 41.90   1st Qu.:  48443
## Median :  7223   Median :  3983   Median : 66.00   Median : 108925
## Mean   : 13117   Mean   :  8965   Mean   : 62.38   Mean   : 168758
## 3rd Qu.: 18159   3rd Qu.: 10603   3rd Qu.: 86.40   3rd Qu.: 246164
## Max.   :358652   Max.   :225313   Max.   :158.40   Max.   :2711061
##                  NA's   :360      NA's   :384
##  totalRevBal       totalIlLim      loan_status          emp_status
## Min.   :     0   Min.   :     0   Bad : 7581   job_tittle   : 1918
## 1st Qu.:  21453   1st Qu.:  14924   Good:27074   no_job_tittle:32737
## Median :  37639   Median :  31505
## Mean   :  49611   Mean   :  41766
## 3rd Qu.:  62405   3rd Qu.:  55689
## Max.   :1094819   Max.   :1027358
##
##     employed_length       risk_grade       closeAcc
##           : 1823   least_risk:15130   Min.   :  0.00
## 0-1 year    : 5124   moderate  :18371   1st Qu.:  7.00
## 10-10+ years:11511   risky     : 1154   Median : 13.00
## 2-5 years   : 9998                      Mean   : 14.25
## 6-9 years   : 6199                      3rd Qu.: 19.00
```

```
##                                Max.   :117.00
##
## [1] "revolRatio" "bcOpen"     "bcRatio"
#install.packages("mice")
library(mice)
## Warning: package 'mice' was built under R version 3.4.4
imputed_data = complete(mice(impute_dataset))
summary(imputed_data)
```

The final 'cleaned_dataset' has divided into 2 datasets. 80% of cleaned_dataset used as training and 20% as test dataset. The 'caTools' package was used to creat the test and training dataset. The training dataset has used to find out the best regression model and the test dataset has used to predict the proportion of bad and good loans by using contingency table.

```
#install.packages("caTools")
require(caTools)
## Warning: package 'caTools' was built under R version 3.4.4
set.seed(123)
sample = sample.split(cleaned_dataset,SplitRatio = 0.80)
training_data = subset(cleaned_dataset,sample ==TRUE)
test_data = subset(cleaned_dataset, sample==FALSE)

training_data = subset(training_data, select = -(totalPaid)) # Drop totalPaid from training dataset
summary(training_data)
summary(test_data)
```

First Model and Diagnostics

The first logistic regression model with all of the explanatory variables is given below,

```
loan_model = glm(loan_status~amount+term+rate+payment+home+income+verified+
                 reason+debtIncRat+delinq2yr+inq6mth+openAcc+pubRec+
                 revolRatio+totalBal+totalRevLim+accOpen24+avgBal+bcOpen+
                 bcRatio+totalLim+totalRevBal+totalIlLim+emp_status+
                 employed_length+risk_grade+closeAcc,
               data = training_data,family="binomial")
summary(loan_model)
```

The AIC of the first model is 26011. The predicted probablity and the overall accuracy of this model has been calculated and compare them with other models to find the best one.

Contingency table of the full model:

```
##       pred_goodbad
##         Bad Good  Sum
##   Bad   199 1291 1490
##   Good  178 5502 5680
##   Sum   377 6793 7170

## [1] "Overall Accuracy (Correctly predicted outcomes):  79.51"

## [1] "Correctly predicted bad loans:  52.79"
```

The percentage of correctly predicted outcomes is 79.51%

Correctly predicted bad loans: 52.79%

Based on the result, this is not an effective model for predicting true good or bad loans.

Improved Model and Diagnostics

To get a better outcome and a good model from the analysis, the number of Bad loans should be increased in order to balance the number of Good and Bad loans. Because a model can achieve high accuracy by simply classifying all loans as “Good”. For increasing the Bad loans rows, the over-sampling technique has used and make the dataset balanced.

Balanced Model

```
table(balanced_training_dataset$loan_status)

##
##   Bad  Good
## 21394 21394
```

From the balanced dataset, a new logistic regression mode and a contingency table has created to predict the overall accuracy of the model, correctly identified Bad loans and incorrectly identified Good loans. The original test dataset has used in the contingency table to compare it with the first model of unbalanced dataset.

Contingency Table from Balanced model,

```
##       pred_GooBad2
##         Bad Good  Sum
##   Bad   967  523 1490
##   Good 1992 3688 5680
##   Sum  2959 4211 7170

## [1] "Overall Accuracy (Correctly predicted outcomes):  64.92"

## [1] "Correctly predicted bad loans:  32.68"

## [1] "Incorrectly identified good loans:  12.42"
```

The AIC value and the over all accuracy of the new model indicates that this model is less efficient than the first one. Because the AIC value of the balanced model is greater than the first

one, also the over all accuracy is less in the balanced model, because the model is created from a balanced dataset.

Now, another model has created by automatic model selection method to compare with the previous ones.

Automatic Model Selection Using Step Function

The first step of automatic model selection is to use the VIF function to find out some extreme coliniarity between coefficients. If the VIF value is more than 10, then those explanatory variables should be removed one by one from the model and test it again to get a more effective model with no coliniarity.

```
new_model = glm(loan_status ~ amount+term+rate+payment+home+income+verified+
              reason+debtIncRat+delinq2yr+inq6mth+openAcc+pubRec+
              revolRatio+totalBal+totalRevLim+accOpen24+avgBal+bcOpen+
              bcRatio+totalLim+totalRevBal+totalIlLim+emp_status+
              employed_length+risk_grade+closeAcc,
            data = balanced_training_dataset, family="binomial")


library(car)
vif(new_model)

##                 GVIF Df GVIF^(1/(2*Df))
## amount        62.341410  1       7.895658
## term           7.849769  1       2.801744
## rate           4.603332  1       2.145538
## payment       52.805063  1       7.266709
## home           1.584268  2       1.121908
## income         1.939340  1       1.392602
## verified       1.266353  2       1.060813
## reason         1.365327 12       1.013059
## debtIncRat     1.590910  1       1.261313
```

```
## delinq2yr        1.100446  1       1.049022
## inq6mth          1.152294  1       1.073449
## openAcc          2.355102  1       1.534634
## pubRec           1.074019  1       1.036349
## revolRatio       3.490971  1       1.868414
## totalBal        92.355081  1       9.610155
## totalRevLim      7.059226  1       2.656920
## accOpen24        1.643453  1       1.281972
## avgBal           5.865565  1       2.421893
## bcOpen           2.663247  1       1.631946
## bcRatio          3.577021  1       1.891302
## totalLim        92.298026  1       9.607186
## totalRevBal     18.160778  1       4.261546
## totalIlLim      15.206496  1       3.899551
## emp_status      14.885119  1       3.858124
## employed_length 16.390292  4       1.418480
## risk_grade       3.309500  2       1.348778
## closeAcc         1.343459  1       1.159077
```

From the vif function it has found that 'amount, payment, totalBal, totalLim, totalRevBal, totalIlLim, emp_status and employed_length' variables have VIF values more than 10. Remove those coefficients one by one based on the high VIF value and checked again if the coliniarity has gone or not. After removing totalBal, amount, employed_length and totalRevBal the VIF values of all the explanatory variables found below 10. Then another model has created and the step wise automatic model selection has applied. Finally got a suitable model from the step function based on AIC value.

New model from step function after removing the large VIF variables

```
step_model = glm(loan_status ~ term+rate+payment+home+income+reason+
                 debtIncRat+delinq2yr+inq6mth+revolRatio+accOpen24+
                 bcOpen+bcRatio+totalLim+totalIlLim+emp_status+risk_grade+
```

```
                closeAcc,family="binomial",data=balanced_training_dataset)
summary(step_model)
```

Contingency table

```
##       GooBad_pred
##         Bad Good  Sum
##   Bad   976  514 1490
##   Good 1998 3682 5680
##   Sum  2974 4196 7170
```

```
## [1] "Overall Accuracy (Correctly predicted outcomes):  64.97"
```

```
## [1] "Correctly predicted bad loans:  65.5"
```

```
## [1] "Incorrectly identified good loans:  34.5"
```

```
## [1] "Correctly predicted good loans:  64.82"
```

AIC value of this model after removing large VIF is greater than the balanced model. Overall Accuracy 64.97%, correctly predicted good loans are 64.82% and correctly predicted bad loans are 65.5%.

Tuning the Predictions and Profit Analysis

After analyzing balanced model and autometic model selection with step function, the AIC value of balanced model is less compared to automatic model. Now calculate the total profit based on the best model.

Profit of Predicted Good Loans According to Balanced Model

```
profit = test_data$totalPaid - test_data$amount

balanced_probability = predict(balanced_model, newdata = test_data, type="response")
pred_GooBad2 = cut(balanced_probability, breaks = c(-Inf, 0.5, Inf), labels=c("Bad", "Good"))
```

```
profitgood = profit[pred_GooBad2 == "Good"]
print(paste('Total Profit: ', sum(profitgood)))

## [1] "Total Profit:  3606233.601471"
```

Changing Threshold

The model accuracy and the total profit of predicted good loans are checked by changing the threshold levels. For different threshold levels, we got different percentage of correctly identified good loans and different total profit.

```
## [1] "Overall Accuracy at threshold level  0.1  is  79.33 %"
## [1] "Correctly predicted bad loans:  0.94 %"
## [1] "Incorrectly identified good loans:  99.06 %"
## [1] "Correctly predicted good loans:  99.89 %"
## [1] "Total Profit:  3023265.06"
## [1] " "
## [1] "Overall Accuracy at threshold level  0.2  is  79.68 %"
## [1] "Correctly predicted bad loans:  10.67 %"
## [1] "Incorrectly identified good loans:  89.33 %"
## [1] "Correctly predicted good loans:  97.78 %"
## [1] "Total Profit:  3804192.2"
## [1] " "
## [1] "Overall Accuracy at threshold level  0.3  is  77.88 %"
## [1] "Correctly predicted bad loans:  27.99 %"
## [1] "Incorrectly identified good loans:  72.01 %"
## [1] "Correctly predicted good loans:  90.97 %"
## [1] "Total Profit:  4346672.56"
## [1] " "
## [1] "Overall Accuracy at threshold level  0.4  is  73.7 %"
## [1] "Correctly predicted bad loans:  47.32 %"
## [1] "Incorrectly identified good loans:  52.68 %"
## [1] "Correctly predicted good loans:  80.62 %"
```

```
## [1] "Total Profit:  4345876.34"
## [1] " "
## [1] "Overall Accuracy at threshold level  0.5  is  64.92 %"
## [1] "Correctly predicted bad loans:  64.9 %"
## [1] "Incorrectly identified good loans:  35.1 %"
## [1] "Correctly predicted good loans:  64.93 %"
## [1] "Total Profit:  3606233.6"
## [1] " "
## [1] "Overall Accuracy at threshold level  0.6  is  54.98 %"
## [1] "Correctly predicted bad loans:  79.73 %"
## [1] "Incorrectly identified good loans:  20.27 %"
## [1] "Correctly predicted good loans:  48.49 %"
## [1] "Total Profit:  2755051.58"
## [1] " "
## [1] "Overall Accuracy at threshold level  0.7  is  43.19 %"
## [1] "Correctly predicted bad loans:  90.87 %"
## [1] "Incorrectly identified good loans:  9.13 %"
## [1] "Correctly predicted good loans:  30.69 %"
## [1] "Total Profit:  1797176.79"
## [1] " "
## [1] "Overall Accuracy at threshold level  0.8  is  28.41 %"
## [1] "Correctly predicted bad loans:  98.26 %"
## [1] "Incorrectly identified good loans:  1.74 %"
## [1] "Correctly predicted good loans:  10.09 %"
## [1] "Total Profit:  601729.45"
## [1] " "
## [1] "Overall Accuracy at threshold level  0.9  is  21.32 %"
## [1] "Correctly predicted bad loans:  100 %"
## [1] "Incorrectly identified good loans:  0 %"
## [1] "Correctly predicted good loans:  0.69 %"
```

```
## [1] "Total Profit:  58404.59"
## [1] " "
```

It has shown from the different threshold level that, at threshold 0.3 the profit is high and correctly predicted good loans percentage is more than 90%.

Results Summary

The analysis of balanced model and autometic model selection, after removing the high VIF coefficients, shows that the AIC value of balanced model is less than the autometic stepwise model. This indicates, the balanced model is the best logistic regression model to calculate the profit for the bank as well as the percentage of correctly predicted good and default loans.

The over all accuracy and the total profit for the predicted good loans with different threshold levels are presented in the following table,

```
##    Threshold TotalProfit
## A       0.10  3023265.06
## B       0.20  3804192.20
## C       0.30  4353921.92
## D       0.40  4349920.54
## E       0.50  3611367.20
## F       0.60  2728642.18
## G       0.70  1779788.48
## H       0.80   596516.07
## I       0.90    63720.37
```

From the table it can be found that, at the threshold level of 0.3 the over all accuracy is more than 90%. Moreover, the total profit of all predicted good loans at the threshold level of 0.3 is higher compare to the profit of all other threshold levels.