Loan Analysis Using Logistic Regression

Kunpei Peng

2/14/2021

Split data set and find accuracy of baseline model on test set

```
Loan = read.csv("Loans.csv")
str(Loan)
```

```
## 'data.frame':    9578 obs. of  14 variables:
##  $ CreditPolicy  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Purpose       : chr  "debt_consolidation" "credit_card" "debt_consolidation" "debt_consoli
dation" ...
##  $ IntRate       : num  0.119 0.107 0.136 0.101 0.143 ...
##  $ Installment   : num  829 228 367 162 103 ...
##  $ LogAnnualInc  : num  11.4 11.1 10.4 11.4 11.3 ...
##  $ Dti           : num  19.5 14.3 11.6 8.1 15 ...
##  $ Fico          : int  737 707 682 712 667 727 667 722 682 707 ...
##  $ DaysWithCrLine: num  5640 2760 4710 2700 4066 ...
##  $ RevolBal      : int  28854 33623 3511 33667 4740 50807 3839 24220 69909 5630 ...
##  $ RevolUtil     : num  52.1 76.7 25.6 73.2 39.5 51 76.8 68.6 51.1 23 ...
##  $ InqLast6mths  : int  0 0 1 1 0 0 0 0 1 1 ...
##  $ Delinq2yrs    : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ PubRec        : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ NotFullyPaid  : int  0 0 0 0 0 0 1 1 0 0 ...
```

```
View(Loan)

# Creating Training and Testing Sets
#install.packages("caTools")
library(caTools)

#Split Loan into train and test sets
set.seed(1234)
split = sample.split(Loan$NotFullyPaid, SplitRatio = 0.70)
Train = subset(Loan, split==TRUE)
Test = subset(Loan, split==FALSE)

#Baseline model accuracy on test set
Baseline_Accuracy <- (nrow(Test)-sum(Test$NotFullyPaid))/nrow(Test)
```

Build a logit model to predict NotFullyPaid using all other variables.

```
#Build model with training dataset
Loan.logit = glm(NotFullyPaid ~ ., data = Train, family=binomial)
summary(Loan.logit)
```

```
##
## Call:
## glm(formula = NotFullyPaid ~ ., family = binomial, data = Train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4651  -0.6141  -0.4914  -0.3670   2.5102
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 7.343e+00  1.548e+00   4.744 2.09e-06 ***
## CreditPolicy               -3.034e-01  1.009e-01  -3.008 0.002629 **
## Purposecredit_card         -5.774e-01  1.311e-01  -4.403 1.07e-05 ***
## Purposedebt_consolidation -4.260e-01  9.219e-02  -4.621 3.81e-06 ***
## Purposeeducational          1.278e-03  1.870e-01   0.007 0.994547
## Purposehome_improvement     1.044e-01  1.485e-01   0.703 0.481923
## Purposemajor_purchase      -3.583e-01  1.978e-01  -1.812 0.070044 .
## Purposesmall_business       4.947e-01  1.381e-01   3.582 0.000341 ***
## IntRate                     3.212e+00  2.080e+00   1.544 0.122508
## Installment                 1.144e-03  2.098e-04   5.453 4.95e-08 ***
## LogAnnualInc               -4.237e-01  7.141e-02  -5.932 2.99e-09 ***
## Dti                         3.990e-03  5.448e-03   0.732 0.463943
## Fico                       -7.357e-03  1.690e-03  -4.354 1.34e-05 ***
## DaysWithCrLine              1.088e-05  1.560e-05   0.698 0.485358
## RevolBal                    3.477e-06  1.143e-06   3.042 0.002350 **
## RevolUtil                   2.159e-03  1.529e-03   1.412 0.157951
## InqLast6mths                9.678e-02  1.632e-02   5.930 3.03e-09 ***
## Delinq2yrs                 -2.784e-02  6.436e-02  -0.433 0.665374
## PubRec                      2.541e-01  1.146e-01   2.217 0.026607 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5896.6  on 6704  degrees of freedom
## Residual deviance: 5476.2  on 6686  degrees of freedom
## AIC: 5514.2
##
## Number of Fisher Scoring iterations: 5
```

From the model outputs above, we can see that the following independent variables are significant: CreditPolicy, Purposecredit_card, Purposedebt_consolidation, Purposesmall_business, Installment, LogAnnualInc, Fico, RevolBal, RevolUtil, InqLast6mths, and PubRec.

Application A has a FICO credit socre of 700 while Application B has a FICO score of 710. What's Logit(A)-Logit(B)?

Since all else are the same besides the Fico scores, we can just calculate the difference between Logit(A) and Logit(B) by subtracting their Fico scores and times the result with the Fico coefficient from the logit model.

```
#Find coefficient for Fico variable from the Logit model
FICO_Coefficient <- Loan.logit$coefficients[["Fico"]]

#Multiply the difference in Fico score by the Fico coefficient from the model
LogitA.LogitB.diff <- (700-710)*FICO_Coefficient
LogitA.LogitB.diff
```

```
## [1] 0.07357483
```

Predict the probability of the test set loans not being paid back in full.

```
PredictedRisk <- predict(Loan.logit, type="response", newdata = Test)
Test$Predicted.Risk = PredictedRisk

library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
Test$PredictedRiskInt = ifelse(PredictedRisk < 0.5, 0, 1)

Actual.vs.Pred <- table(Test$NotFullyPaid, Test$PredictedRiskInt)
Actual.vs.Pred
```

```
##
##        0    1
##   0 2403   10
##   1  449   11
```

```
Loan.Pred.Test.Accuracy <- sum(diag(Actual.vs.Pred))/sum(Actual.vs.Pred)

Loan.Pred.Test.Accuracy
```

```
## [1] 0.8402367
```

This model accuracy of 83.78% is very close to the benchmark model accuracy of 83.99%.

What's the AUC?

```
#ROC curve
#install.packages("ROCR")
library(ROCR)
prediction = prediction(Test$Predicted.Risk, Test$NotFullyPaid)
performance(prediction, "auc")@y.values
```

```
## [[1]]
## [1] 0.6752311
```

The AUC of the model on the test set is about 0.6674, while the model accuracy on test set is about 83.78%. I think investors can use this as a temporary model for reference while developing more accurate models.

Use a logistic regression model to predict NotFullyPaid using only IntRate

```
# uild new logit model using interest rate as the only independent variable
Loan.logit.IR = glm(NotFullyPaid ~ IntRate, data = Train, family=binomial)
summary(Loan.logit.IR)
```

```
##
## Call:
## glm(formula = NotFullyPaid ~ IntRate, family = binomial, data = Train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0963  -0.6294  -0.5377  -0.4255   2.3197
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.8282     0.1704  -22.47   <2e-16 ***
## IntRate      17.1353     1.2795   13.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5896.6  on 6704  degrees of freedom
## Residual deviance: 5710.9  on 6703  degrees of freedom
## AIC: 5714.9
##
## Number of Fisher Scoring iterations: 4
```

From the model output above, we can see that IntRate is statistically significant when it was the only independent variable in the new logit model. However, it was not significant in the previous model built when other independent variables are included. My hypothesis is that the IntRate is highly correlated with one or more other variables in the previous model built, causing multi-collinearity.

Use the interest rate model to predict probability of NotFullyPaid on the test set

```
Prediction.IR <- predict(Loan.logit.IR, type="response", newdata = Test)
Test$Prediction.IR = Prediction.IR

#get the highest predicted probability of a loan not being paid back in full on the test set
max(Prediction.IR)
```

```
## [1] 0.4516973
```

If we use 0.5 as a threshold to determine if a customer will pay back in full or not, none of the customers on the testing set will have a probability higher than 0.5.

```
#calculate AUC
library(ROCR)
prediction.IR.vs.Actual = prediction(Prediction.IR, Test$NotFullyPaid)
performance(prediction.IR.vs.Actual, "auc")@y.values
```

```
## [[1]]
## [1] 0.6133313
```

The the current model is more robust than the previous model because the current model only has 1 independent variable, while the previous model has all other independent variables as input. The contrast between 0.6186 of AUC is so much worse than the 0.6674 of AUC from the more complicated model.

Using our logistic regression model to identify loans that are expected to be profitable:

Scenario 1: assume that we have a $10 investment with an annual interest rate of 6% pay back after 3 years, using continuous compounding of interest:

$$10 * e^{0.06*3} = 11.97217$$

Based on the calculation above, the expected returns (principle + interest) after 3 years at 6% payback would be $11.97-$10 = $1.97.

Note: If the investment is not paid back in full, the investor will have a negative balance on their returns because the investor will have no profit.

Computing the profit of a $1 investment in each loan

```
#Create new column for profit, assuming C = $1
Test$profit = 1*exp(Test$IntRate*3) - 1
Test$profit[Test$NotFullyPaid == 1] = -1
# max(test$profit)
MaxProfitC1 <- max(Test$profit)
MaxProfitC1
```

```
## [1] 0.8894769
```

The max profit amount is $0.889.

Alternative Investment Strategy

```
#create new data set for loans with high interest rates
HighInterest  <- subset(Test, Test$IntRate  >= 0.15)

#Mean profit for high interest rate loan
mean(HighInterest$profit)
```

```
## [1] 0.2691649
```

```
#Proportion of interest not paid back in full
sum(HighInterest$NotFullyPaid == 1)/nrow(HighInterest)
```

```
## [1] 0.2271715
```

What is the profit of an investor who invested $1 in each of these 100 lonas? How many of the 100 selected loans were not paid in full? How does this compare to the simple strategy?

```
# sort by predicted risk
HighInterest <- HighInterest[order(HighInterest$PredictedRisk),]

#Creating the SelectedLoan dataset
Threshold = sort(HighInterest$Predicted.Risk, decreasing=FALSE)[100]
SelectedLoans = subset(HighInterest, Predicted.Risk <= Threshold)
dim(SelectedLoans)
```

```
## [1] 100  18
```

```
#Total profit for an investor who invested $1 in each of the 100 loans
Total.Profit.for.investors <- sum(SelectedLoans$profit)
Total.Profit.for.investors
```

```
## [1] 36.35442
```

```
#Number of loans not paid back in full
table(SelectedLoans$NotFullyPaid)
```

```
##
##  0  1
## 84 16
```

As the above analyses shown, the profit of an investor who invested $1 in each of these loans would earn an expected profit of $31.24. Out of the 100 loans, 19 of them will not be paid in full. This simple strategy, yielding $31.24 profit seems perform better than investing in all loans, yielding $20.94 for $100 investment.

Note: Be wary of the assumptions behind this analysis:

The important assumptions of predictive modeling is that past is indicative of the future. In the context of financial situations, credit risk might fluctuate based on a person's living quality and job stability. If a person's life is heavily impacted by sudden change of events in factors such as job security, they might now be able to maintain a good credit score like they did in the past. To hedge this risk, an analyst can do further research in gathering more data or even survey a few of the customers who took out the loans to get a better understanding of their personal financial visions and plans.