hmeq data analysis using logistic regression

Sungho Moon

8/1/2021

Data description

The data set HMEQ reports characteristics and delinquency information for 5,960 home equity loans. A home equity loan is a loan where the obligor uses the equity of his or her home as the underlying collateral. The data set has the following characteristics:

BAD: 1 = applicant defaulted on loan or seriously delinquent; 0 = applicant paid loan LOAN: Amount of the loan request

MORTDUE: Amount due on existing mortgage

VALUE: Value of current property

REASON: DebtCon = debt consolidation; HomeImp = home improvement

JOB: Occupational categories

YOJ: Years at present job

DEROG: Number of major derogatory reports

DELINQ: Number of delinquent credit lines

CLAGE: Age of oldest credit line in months

NINQ: Number of recent credit inquiries

CLNO: Number of credit lines

DEBTINC: Debt-to-income ratio

Loading libraries

```r
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
```

```r
library(PerformanceAnalytics)
```

```
## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend
```

```r
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
library(lmtest)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(ROCR)
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
library(dgof)
```

```
##
## Attaching package: 'dgof'

## The following object is masked from 'package:stats':
##
##     ks.test
```

```r
library(usdm)
```

```
## Loading required package: sp

## Loading required package: raster

##
## Attaching package: 'raster'

## The following objects are masked from 'package:MASS':
##
##     area, select

## The following object is masked from 'package:dplyr':
##
##     select

##
## Attaching package: 'usdm'

## The following object is masked from 'package:car':
##
##     vif
```

Loading data

```r
hmeq.ori <- read.csv("hmeq.csv"); head(hmeq.ori)
```

```
##   BAD LOAN MORTDUE  VALUE  REASON    JOB  YOJ DEROG DELINQ     CLAGE NINQ CLNO
## 1   1 1100   25860  39025 HomeImp  Other 10.5     0      0  94.36667    1    9
## 2   1 1300   70053  68400 HomeImp  Other  7.0     0      2 121.83333    0   14
## 3   1 1500   13500  16700 HomeImp  Other  4.0     0      0 149.46667    1   10
```

```
## 4   1 1500      NA     NA                  NA     NA      NA         NA   NA   NA
## 5   0 1700   97800 112000 HomeImp Office  3.0      0       0   93.33333    0   14
## 6   1 1700   30548  40320 HomeImp  Other  9.0      0       0 101.46600    1    8
##    DEBTINC
## 1       NA
## 2       NA
## 3       NA
## 4       NA
## 5       NA
## 6 37.11361
```

BAD = 1 means delinquency (default) of one's loan.

Data Cleaning

Omit rows including NA and empty entries

```
hmeq <- na.omit(hmeq.ori)
clean <- c()
for(i in 1:ncol(hmeq)){
  if("" %in% unique(hmeq[,i])){
    clean[i] <- i
  }
  else{
    clean[i] <- 0
  }
}
clean <- unique(clean[clean>0])
for(i in 1:length(clean)){
  hmeq <- subset(hmeq,subset = hmeq[,clean[i]]!="")
}
```

Set categorical variables as factors

```
for(i in c(1,5,6)){
  hmeq[,i] <- as.factor(hmeq[,i])
}
```

Exploratory Data Analysis

```
length(c(2:4,7:13))
```

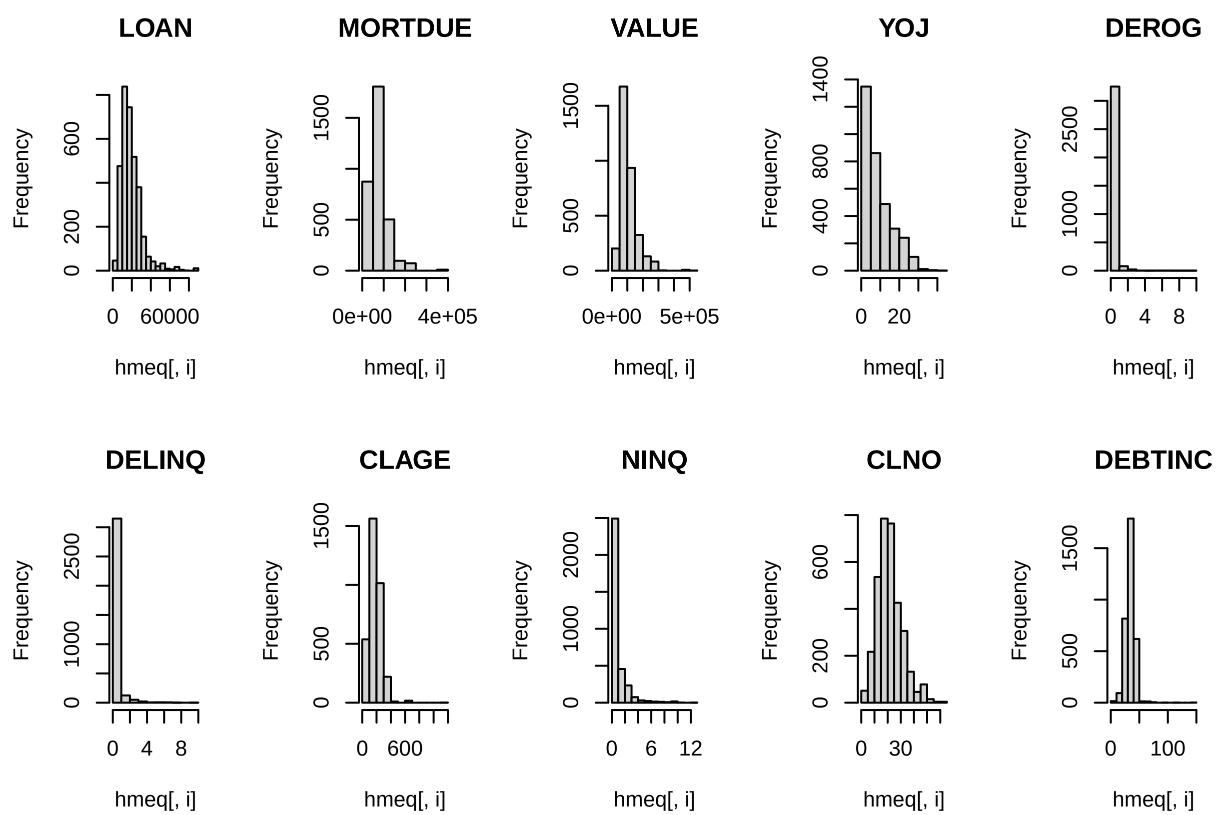
```
## [1] 10
```

```
par(mfrow=c(2,5))
for(i in c(2:4,7:13)){
  print(colnames(hmeq)[i])
```

```
  print(summary(hmeq[,i]))
  print(shapiro.test(hmeq[,i])[2])
  hist(hmeq[,i],main=colnames(hmeq)[i])
}
```

```
## [1] "LOAN"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1700   12000   17000   19154   23825   89900
## $p.value
## [1] 6.474179e-50

## [1] "MORTDUE"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5076   49351   67278   76250   92987  399412
## $p.value
## [1] 2.340408e-47

## [1] "VALUE"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   21144   71235   94454  107501  122339  512650
## $p.value
## [1] 8.49089e-50

## [1] "YOJ"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    3.00    7.00    9.11   13.00   41.00
## $p.value
## [1] 8.628721e-40

## [1] "DEROG"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.1468  0.0000 10.0000
## $p.value
## [1] 1.186515e-78

## [1] "DELINQ"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.2788  0.0000 10.0000
## $p.value
## [1] 7.784152e-75

## [1] "CLAGE"
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##    0.4867  118.6879  176.7420  180.9937  230.4022 1168.2336
## $p.value
## [1] 1.613498e-38

## [1] "NINQ"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   1.037   2.000  13.000
## $p.value
## [1] 2.679141e-62
```

```
## [1] "CLNO"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   16.00   21.00   22.11   27.00   64.00
## $p.value
## [1] 4.939999e-28

## [1] "DEBTINC"
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##   0.8381  29.3626  35.1295  34.1354  39.0876 144.1890
## $p.value
## [1] 1.181944e-46
```




```r
par(mfrow=c(1,1))
```

It seems like the scales of continuous variables are quite different. We meed scaling. All the Shapiro-Wilk tests of normality for each continuous variable have very small p-values. Thus we reject null hypothesis. Data are not distributed normally.

Dealing with outliers

```r
for(i in c(2:4,7:13)){
  outliers<-boxplot(hmeq[,i],plot=FALSE)$out
  hmeq <- hmeq[-which(hmeq[,i] %in% outliers),]
}
```

Delete columns with 98% or more duplicate values

```
hmeq <- hmeq[,-c(8,9)]
```

Data Scaling

Min-Max Normalization for continuous variables

```
normalize <- function(X){
  for(i in 1:ncol(X)){
    X[,i] <- (X[,i]-min(X[,i]))/(max(X[,i])-min(X[,i]))
  }
  return(X)
}
hmeq.n <- normalize(hmeq[,c(2:4,7:11)])
hmeq.n <- cbind(hmeq.n,hmeq[,c(1,5:6)])
```

```
head(hmeq)
```

Before scaling

```
##    BAD LOAN MORTDUE  VALUE  REASON    JOB YOJ     CLAGE NINQ CLNO  DEBTINC
## 6    1 1700   30548  40320 HomeImp  Other   9 101.46600    1    8 37.11361
## 8    1 1800   28502  43034 HomeImp  Other  11  88.76603    0    8 36.88489
## 20   0 2300  102370 120953 HomeImp Office   2  90.99253    0   13 31.58850
## 26   1 2400   34863  47471 HomeImp    Mgr  12  70.49108    1   21 38.26360
## 27   0 2400   98449 117195 HomeImp Office   4  93.81177    0   13 29.68183
## 35   0 2900  103949 112505 HomeImp Office   1  96.10233    0   13 30.05114
```

```
head(hmeq.n)
```

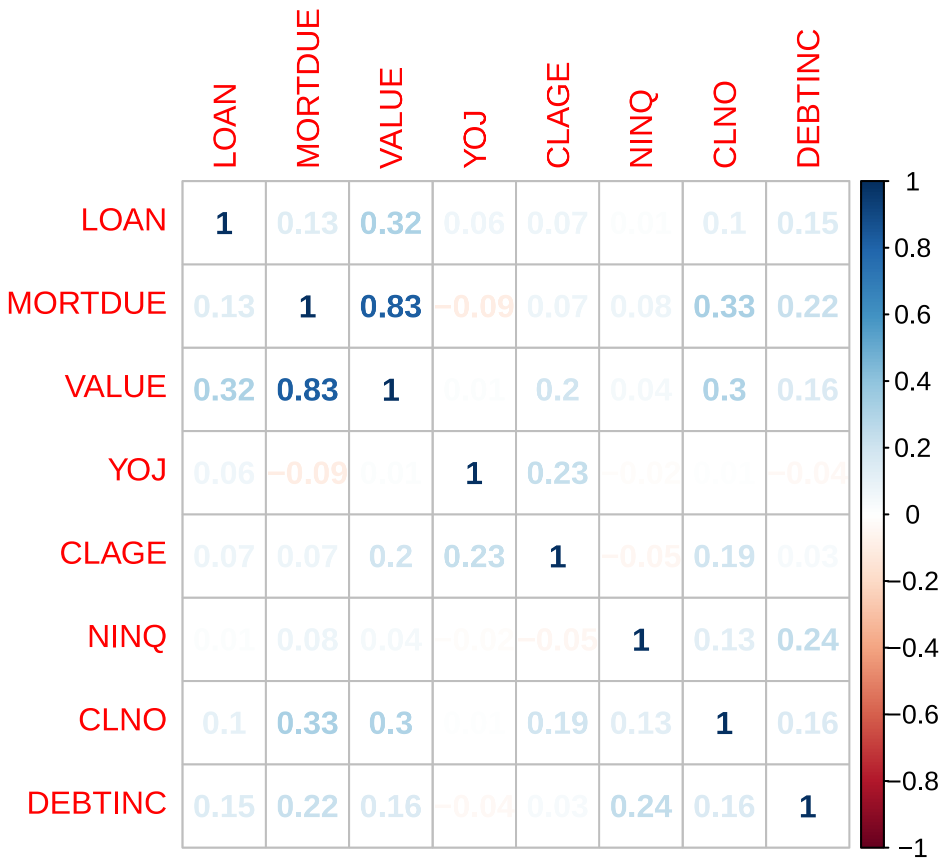
After scaling

```
##           LOAN   MORTDUE     VALUE        YOJ     CLAGE NINQ  CLNO   DEBTINC
## 6  0.000000000 0.1732848 0.1152355 0.30000000 0.2267271  0.2 0.150 0.6226247
## 8  0.002538071 0.1593660 0.1315449 0.36666667 0.1910495  0.0 0.150 0.6160672
## 20 0.015228426 0.6618865 0.5997885 0.06666667 0.1973043  0.0 0.275 0.4642183
## 26 0.017766497 0.2026395 0.1582085 0.40000000 0.1397102  0.2 0.475 0.6555951
## 27 0.017766497 0.6352121 0.5772053 0.13333333 0.2052243  0.0 0.275 0.4095534
## 35 0.030456853 0.6726283 0.5490214 0.03333333 0.2116591  0.0 0.275 0.4201416
##    BAD  REASON    JOB
## 6    1 HomeImp  Other
## 8    1 HomeImp  Other
## 20   0 HomeImp Office
## 26   1 HomeImp    Mgr
## 27   0 HomeImp Office
## 35   0 HomeImp Office
```

Checking Multicollinearity

Checking correlation plot

```r
cont.var <- hmeq.n[,-c(9:11)]
M <- cor(cont.var)
corrplot(M, method="number")
```



Checking VIF

```r
vif(cont.var)
```

```
##   Variables      VIF
## 1      LOAN 1.211352
## 2   MORTDUE 3.807156
## 3     VALUE 4.020883
## 4       YOJ 1.082964
## 5     CLAGE 1.162206
## 6      NINQ 1.079784
## 7      CLNO 1.179997
## 8   DEBTINC 1.146369
```

MORTDUE and VALUE have quite high VIF compared to other variables. They may be highly correlated. Adding an interaction term may help the model fit better.

Model fitting

Divide data into training / test dataset(7:3)

```
set.seed(2416)
sub <- sample(nrow(hmeq.n),floor(nrow(hmeq.n)*0.7))
train <- hmeq.n[sub,]
test <- hmeq.n[-sub,]
```

May not be necessary ATM but will use the train/test data later

Fit a Model

```
fit.train <- glm(BAD~(.)+MORTDUE:VALUE,data=train,family=binomial)
summary(fit.train)
```

Baseline category for dummy variables: JOB=Mgr, REASON=DebtCon

```
##
## Call:
## glm(formula = BAD ~ (.) + MORTDUE:VALUE, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2226  -0.3031  -0.1749  -0.0973   3.8054
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.7520     0.9140   0.823 0.410648
## LOAN           -3.6874     1.0472  -3.521 0.000429 ***
## MORTDUE        -6.6555     2.7265  -2.441 0.014645 *
## VALUE          -8.0901     2.5701  -3.148 0.001645 **
## YOJ            -1.2908     0.7581  -1.703 0.088635 .
## CLAGE          -4.2000     0.8851  -4.745 2.08e-06 ***
## NINQ            1.2850     0.5735   2.241 0.025038 *
## CLNO           -0.8885     0.7739  -1.148 0.250958
## DEBTINC         4.0817     0.9442   4.323 1.54e-05 ***
## REASONHomeImp  -0.4430     0.3319  -1.335 0.181955
## JOBOffice      -0.4544     0.5962  -0.762 0.445995
## JOBOther        0.1909     0.4516   0.423 0.672555
## JOBProfExe      0.4694     0.5028   0.933 0.350589
## JOBSales        1.8099     0.7368   2.457 0.014025 *
## JOBSelf        -0.2632     1.2653  -0.208 0.835199
## MORTDUE:VALUE  14.7787     2.6797   5.515 3.48e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 566.12  on 1503  degrees of freedom
## Residual deviance: 443.36  on 1488  degrees of freedom
## AIC: 475.36
##
## Number of Fisher Scoring iterations: 7
```

There are some insignificant variables, and the fit of the model may not the best. Thus, stepwise variable selection (purposeful variable selection later) is required.

Stepwise variable selection (fit1)

```r
step(fit.train)
```

```
## Start:  AIC=475.36
## BAD ~ (LOAN + MORTDUE + VALUE + YOJ + CLAGE + NINQ + CLNO + DEBTINC +
##     REASON + JOB) + MORTDUE:VALUE
##
##                 Df Deviance    AIC
## - CLNO           1   444.70 474.70
## - REASON         1   445.20 475.20
## <none>               443.36 475.36
## - JOB            5   453.36 475.36
## - YOJ            1   446.51 476.51
## - NINQ           1   448.16 478.16
## - LOAN           1   456.43 486.43
## - DEBTINC        1   464.77 494.77
## - CLAGE          1   470.97 500.97
## - MORTDUE:VALUE  1   471.55 501.55
##
## Step:  AIC=474.7
## BAD ~ LOAN + MORTDUE + VALUE + YOJ + CLAGE + NINQ + DEBTINC +
##     REASON + JOB + MORTDUE:VALUE
##
##                 Df Deviance    AIC
## - JOB            5   454.19 474.19
## - REASON         1   446.29 474.29
## <none>               444.70 474.70
## - YOJ            1   448.24 476.24
## - NINQ           1   449.30 477.30
## - LOAN           1   459.37 487.37
## - DEBTINC        1   465.15 493.15
## - MORTDUE:VALUE  1   476.28 504.28
## - CLAGE          1   476.30 504.30
##
## Step:  AIC=474.19
## BAD ~ LOAN + MORTDUE + VALUE + YOJ + CLAGE + NINQ + DEBTINC +
##     REASON + MORTDUE:VALUE
##
##                 Df Deviance    AIC
## <none>               454.19 474.19
## - REASON         1   456.56 474.56
```

```
## - NINQ           1   457.43 475.43
## - YOJ            1   458.82 476.82
## - LOAN           1   471.30 489.30
## - DEBTINC        1   478.01 496.01
## - CLAGE          1   484.15 502.15
## - MORTDUE:VALUE  1   485.62 503.62
```

```
##
## Call:  glm(formula = BAD ~ LOAN + MORTDUE + VALUE + YOJ + CLAGE + NINQ +
##     DEBTINC + REASON + MORTDUE:VALUE, family = binomial, data = train)
##
## Coefficients:
##   (Intercept)           LOAN        MORTDUE          VALUE            YOJ
##        0.9226        -3.9969        -7.6846        -7.1988        -1.5301
##         CLAGE           NINQ        DEBTINC  REASONHomeImp  MORTDUE:VALUE
##       -4.2653         1.0330         4.0974        -0.4895        14.5722
##
## Degrees of Freedom: 1503 Total (i.e. Null);  1494 Residual
## Null Deviance:       566.1
## Residual Deviance: 454.2     AIC: 474.2
```

```r
fit.train.step <- glm(formula = BAD ~ LOAN + MORTDUE + VALUE + YOJ + CLAGE + NINQ +
                DEBTINC + REASON + MORTDUE:VALUE, family = binomial, data = train)
summary(fit.train.step)
```

```
##
## Call:
## glm(formula = BAD ~ LOAN + MORTDUE + VALUE + YOJ + CLAGE + NINQ +
##     DEBTINC + REASON + MORTDUE:VALUE, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1723  -0.3096  -0.1884  -0.1036   3.7175
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.9226     0.7999   1.153  0.24871
## LOAN           -3.9969     0.9983  -4.004 6.24e-05 ***
## MORTDUE        -7.6846     2.4732  -3.107  0.00189 **
## VALUE          -7.1988     2.3902  -3.012  0.00260 **
## YOJ            -1.5301     0.7454  -2.053  0.04011 *
## CLAGE          -4.2653     0.8619  -4.949 7.47e-07 ***
## NINQ            1.0330     0.5626   1.836  0.06634 .
## DEBTINC         4.0974     0.9053   4.526 6.01e-06 ***
## REASONHomeImp  -0.4895     0.3250  -1.506  0.13210
## MORTDUE:VALUE  14.5722     2.4609   5.921 3.19e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 566.12  on 1503  degrees of freedom
## Residual deviance: 454.19  on 1494  degrees of freedom
```

```
## AIC: 474.19
##
## Number of Fisher Scoring iterations: 7
```

Here, REASONHomeImp & NINQ seems to be insignificant. Later, remove REASONHomeImp and re-fit the model, to see if it improves the goodness-of-fit or not.

Re-fitting the model (fit2)

```
fit.train.step2 <- glm(formula = BAD ~ LOAN + MORTDUE + VALUE + YOJ + CLAGE + NINQ +
                        DEBTINC + MORTDUE:VALUE, family = binomial, data = train)
summary(fit.train.step2)
```

```
##
## Call:
## glm(formula = BAD ~ LOAN + MORTDUE + VALUE + YOJ + CLAGE + NINQ +
##     DEBTINC + MORTDUE:VALUE, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2692  -0.3076  -0.1927  -0.1092   3.6960
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.5802     0.7754   0.748 0.454322
## LOAN           -3.5292     0.9476  -3.725 0.000196 ***
## MORTDUE        -7.4464     2.4571  -3.031 0.002441 **
## VALUE          -6.7930     2.3357  -2.908 0.003634 **
## YOJ            -1.6850     0.7384  -2.282 0.022488 *
## CLAGE          -4.1345     0.8569  -4.825 1.40e-06 ***
## NINQ            1.1314     0.5537   2.043 0.041026 *
## DEBTINC         3.9542     0.8901   4.443 8.89e-06 ***
## MORTDUE:VALUE  13.8576     2.3864   5.807 6.36e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 566.12  on 1503  degrees of freedom
## Residual deviance: 456.56  on 1495  degrees of freedom
## AIC: 474.56
##
## Number of Fisher Scoring iterations: 7
```

Here, newly fitted model seems to be not so much different in terms of the goodness-of-fit. Since one explanatory variable (dummy) is removed, the p-value of intercept term has increased. On the other hand, statistical significance of NINQ has been stabilized, as its p-value decreased below a conventional significance level (0.05).

Model Diagnostic

Perform likelihood ratio test for fit1 vs. fit2.

```r
lrtest(fit.train.step,fit.train.step2)
```

```
## Likelihood ratio test
##
## Model 1: BAD ~ LOAN + MORTDUE + VALUE + YOJ + CLAGE + NINQ + DEBTINC +
##     REASON + MORTDUE:VALUE
## Model 2: BAD ~ LOAN + MORTDUE + VALUE + YOJ + CLAGE + NINQ + DEBTINC +
##     MORTDUE:VALUE
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  10 -227.10
## 2   9 -228.28 -1 2.3651     0.1241
```

Since the p-value of the LRT is greater than the conventional significance level, null hypothesis cannot be rejected. Stick with fit1 (considering goodness-of-fit). Yet, more discussion is required regarding the model's predictive power.

Confusion Matrix

```r
train.prob1 <- predict(fit.train.step, type="response")
train.pred1 <- ifelse(train.prob1>0.5,"1","0")
table(train.pred1,train$BAD)
```

Confusion matrix of fit1

```
##
## train.pred1    0    1
##           0 1434   66
##           1    0    4
```

```r
mean(train.pred1==train$BAD)
```

```
## [1] 0.956117
```

```r
train.prob2 <- predict(fit.train.step2, type="response")
train.pred2 <- ifelse(train.prob2>0.5,"1","0")
table(train.pred2,train$BAD)
```

Confusion matrix of fit2

```
##
## train.pred2    0    1
##           0 1433   65
##           1    1    5
```

```r
mean(train.pred2==train$BAD)
```

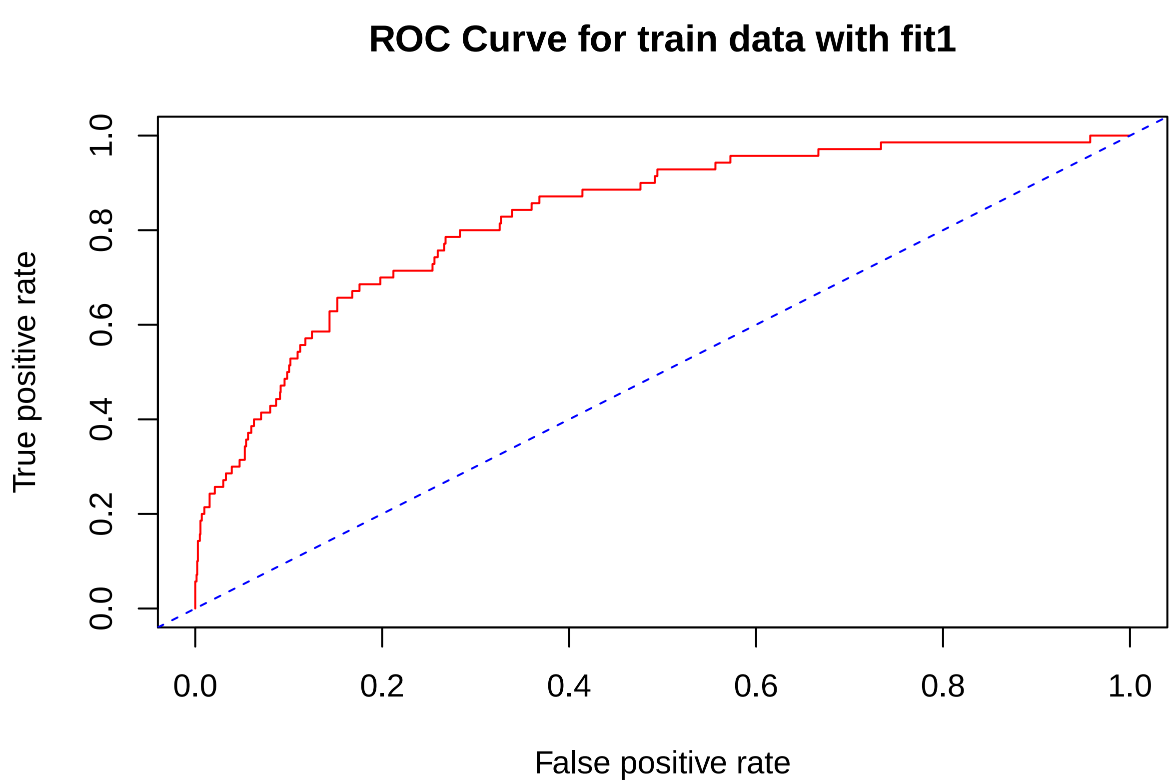
```
## [1] 0.956117
```

Comparing the confusion matrices of fit1 and fit2, there exists no big difference. The numbers of wrong predictions from fit1 and fit2 are equal. TPRs(True Positive Rate) are equal.

ROC Curve and AUC

ROC & AUC (using ROCR package) of fit.step

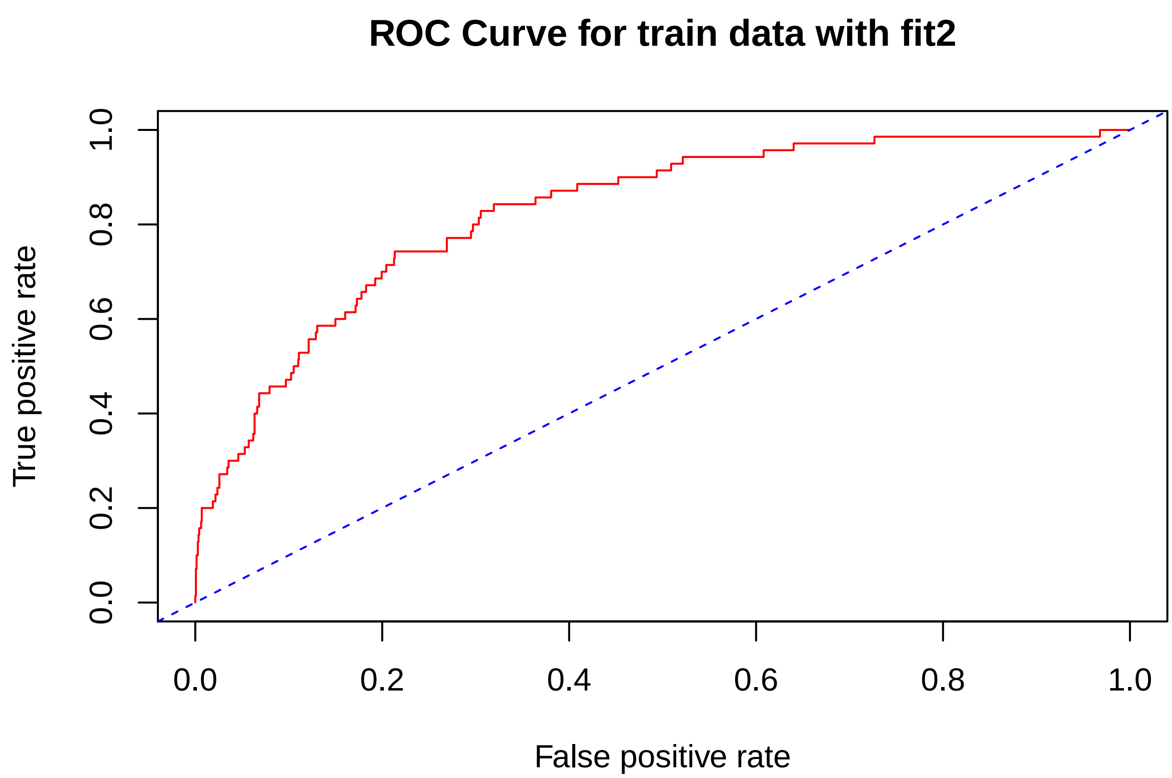
```r
train.roc1 <- prediction(train.prob1,train$BAD)
plot(performance(train.roc1,"tpr","fpr"),col="red",main="ROC Curve for train data with fit1")
abline(0,1,lty=8,col="blue")
```




```r
train.auc1 <- performance(train.roc1,"auc")
slot(train.auc1,"y.values")
```

```
## [[1]]
## [1] 0.8282128
```

```r
train.roc2 <- prediction(train.prob2,train$BAD)
plot(performance(train.roc2,"tpr","fpr"),col="red",main="ROC Curve for train data with fit2")
abline(0,1,lty=8,col="blue")
```




```r
train.auc2 <- performance(train.roc2,"auc")
slot(train.auc2,"y.values")
```

```
## [[1]]
## [1] 0.8269576
```

AUC_fit1 is greater than AUC_fit2 by 0.0013. Fit1 has a slightly better predictive power than fit2.

KS Statistic (Kolmogorov-Smirnov)

```r
train.perf <- performance(train.roc1, "tpr", "fpr")
train.ks <- max(train.perf@y.values[[1]]-train.perf@x.values[[1]])
train.ks
```

```
## [1] 0.5179319
```

KS Statistic is 0.5295. Good enough discriminatory power to distinguish “BAD” and “GOOD”

Conclusion

Final model:

```
summary(fit.train.step)
```

```
##
## Call:
## glm(formula = BAD ~ LOAN + MORTDUE + VALUE + YOJ + CLAGE + NINQ +
##     DEBTINC + REASON + MORTDUE:VALUE, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1723  -0.3096  -0.1884  -0.1036   3.7175
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.9226     0.7999   1.153  0.24871
## LOAN           -3.9969     0.9983  -4.004 6.24e-05 ***
## MORTDUE        -7.6846     2.4732  -3.107  0.00189 **
## VALUE          -7.1988     2.3902  -3.012  0.00260 **
## YOJ            -1.5301     0.7454  -2.053  0.04011 *
## CLAGE          -4.2653     0.8619  -4.949 7.47e-07 ***
## NINQ            1.0330     0.5626   1.836  0.06634 .
## DEBTINC         4.0974     0.9053   4.526 6.01e-06 ***
## REASONHomeImp  -0.4895     0.3250  -1.506  0.13210
## MORTDUE:VALUE  14.5722     2.4609   5.921 3.19e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 566.12  on 1503  degrees of freedom
## Residual deviance: 454.19  on 1494  degrees of freedom
## AIC: 474.19
##
## Number of Fisher Scoring iterations: 7
```

Probability of the default ("BAD") is:

Pr(Y=1) = 1 / (1+e^x)

where x is the attribute vector defined as:

X=−(0.9226−3.9969∗LOAN−7.6846∗MORTDUE−7.1988∗VALUE−1.5301∗YOJ−4.2653∗

CLAGE+1.033∗NINQ+4.0974∗DEBTINC−0.4895∗REASONHomeImp+14.5722∗MORTDUEiVALUE)

Example 1 (using min-max normalized data) Suppose one with almost all the median attributes. That is, LOAN = 0.5, MORTDUE = 0.5, VALUE = 0.5, YOJ = 0.5, CLAGE = 0.5, NINQ = 0.6, DEBTINC = 0.5, REASON = 1 (HomeImp), MORTDUE:VALUE = 0.25. By inserting the values in the function, the resulting output is

$$\mathbf{X=-(0.9226-3.9969*0.5-7.6846*0.5-7.1988*0.5-1.5301*0.5-4.2653*0.5+1.033*}$$

$$\mathbf{0.6+4.0974*0.5-0.4895*1+14.5722*0.25)=5.5932}$$

$$\mathbf{Pr(Y=1) = 1 / (1+e\hat{\ }x) = 1 / (1 + e\hat{\ }(5.5932)) = 0.0037}$$

This implies that one with the given median attributes would have 0.37%

chance of resulting delinquency(default).

Example 2 (using min-max normalized data) Suppose one with (intuitively) bad attributes. That is, LOAN = 0.9, MORTDUE = 0.9, VALUE = 0.8, YOJ = 0.1, CLAGE = 0.1, NINQ = 0.2, DEBTINC = 0.8, REASON = 0 (not HomeImp), MORTDUE:VALUE = 0.81. By inserting the values in the function, the resulting output is

$$\mathbf{X=-(0.9226-3.9969*0.9-7.6846*0.59-7.1988*0.8-1.5301*0.1-4.2653*0.1+1.033*0.2+}$$

$$\mathbf{4.0974*0.8-0.4895*0+14.5722*0.81)=-1.740898}$$

$$\mathbf{Pr(Y=1) = 1 / (1+e\hat{\ }x) = 1 / (1+e(-1.740898)) = 0.850811}$$

This implies that one with such (intuitively) bad attributes would have

85.08% chance of resulting delinquency(default).