LendingClub Probability of Default

Seung Hun Ham

August 24, 2017

1. Importing Data

```r
library(tidyverse)
library(smbinning)
library(InformationValue)

raw_data <- read_csv("LoanStats3b.csv")
```

2. Cleaning Data

```r
# Only including loans with 36 months term
my_data <- filter(raw_data, term == "36 months")

# Excluding variables with 1 unique value
my_data <- my_data[,sapply(my_data, function(x) length(unique(x)) > 1)]

# Excluding certain variables
excluding_vars <- names(my_data) %in%
  c("grade", "sub_grade", "emp_title", "issue_d", "desc", "title", "last_pymnt_d",
    "last_credit_pull_d", "last_pymnt_amnt", "recoveries", "total_rec_prncp",
    "total_pymnt", "total_pymnt_inv", "collection_recovery_fee",
    "total_rec_late_fee", "total_rec_int")
my_data <- my_data[!excluding_vars]

# Making loan_status = 1 for defaulted loans and 0 for current loans
my_data <- mutate(my_data, loan_status = ifelse(loan_status == "Charged Off", 1, 0))

# Converting character variables to numeric
my_data[,"int_rate"] <-
  as.numeric(sapply(my_data[,"int_rate"], function(x) sub("%","",x)))
my_data[,"revol_util"] <-
  as.numeric(sapply(my_data[,"revol_util"], function(x) sub("%","",x)))
my_data[,"earliest_cr_line"] <-
  lapply(my_data[,"earliest_cr_line"],
         function(x) as.numeric(difftime(as.Date(Sys.Date(), format='%d/%m/%y'),
                                         as.Date(paste("01", x, sep="-"), "%d-%b-%y"),
                                         unit="weeks")/52.25))
my_data[,"zip_code"] <-
  as.numeric(sapply(my_data[,"zip_code"], function(x) as.numeric(substring(x,1,3))))

# Recategorize some character variables
my_data <-
  mutate(my_data, emp_length =
           ifelse(emp_length %in% c("1 year","2 years","3 years"), "1-3 years",
           ifelse(emp_length %in% c("4 year","5 years","6 years"), "4-6 years",
           ifelse(emp_length %in% c("7 year","8 years","9 years"), "7-9 years",
           emp_length))))

my_data <-
  mutate(my_data, purpose = ifelse(purpose %in% c("renewable_energy","house","vacation"),
                                   "other", purpose))

# Change character columns to factor
chac_colnames = colnames(my_data[sapply(my_data, is.character)])
my_data[chac_colnames] <- lapply(my_data[chac_colnames], factor)

# Change numeric columns to factor if number of unique values < 10
to_factor = sapply(my_data, function(x) is.numeric(x) & length(unique(x)) < 10)
to_factor[which(colnames(my_data)=="loan_status")] = FALSE
my_data[,to_factor] <- lapply(my_data[,to_factor], factor)

# Change na values to 0
my_data[is.na(my_data)] <- 0

# Check data integrity
x <- data.frame(colnames(my_data), sapply(my_data, function(x) length(unique(x))),
                sapply(my_data, class), sapply(my_data, function(x) sum(is.na(x))))
```

3. Defining Training and Test Sets

```
set.seed(12345)

input_default_set <- my_data[my_data$loan_status == 1, ]
input_current_set <- my_data[my_data$loan_status == 0, ]

train_default_row = sample(1:nrow(input_default_set), 0.7*nrow(input_default_set))
train_current_row = sample(1:nrow(input_current_set), 0.7*nrow(input_default_set))

train_default_set <- input_default_set[train_default_row, ]
test_default_set <- input_default_set[-train_default_row, ]

train_current_set <- input_current_set[train_current_row, ]
test_current_set <- input_current_set[-train_current_row, ]

train_data = rbind(train_default_set, train_current_set)
test_data = rbind(test_default_set, test_current_set)
```

4. Picking Variables Using Information Values

```
y_index <- which(names(train_data)=="loan_status")
iv_df <- data.frame(VARS=colnames(train_data)[-y_index], IV=numeric(ncol(train_data)-1))
```

```r
data_type = sapply(train_data, class)

for(var in colnames(train_data)[-y_index]){
  if(data_type[var] == "factor"){
    smb <- smbinning.factor(as.data.frame(train_data), y="loan_status", x=var)
  } else {
    smb <- smbinning(as.data.frame(train_data), y="loan_status", x=var)
  }

  if(class(smb) != "character"){
    iv_df[iv_df$VARS == var, "IV"] <- smb$iv
  }
}

iv_df <- iv_df[order(-iv_df$IV), ]
head(iv_df)
```

```
##                VARS     IV
## 4          int_rate 0.2626
## 8        annual_inc 0.1114
## 64 tot_hi_cred_lim 0.0839
## 33  bc_open_to_buy 0.0714
## 32     avg_cur_bal 0.0713
## 29     tot_cur_bal 0.0662
```

From the above results, I found the below top 5 loan attributes predictive of default:

- Interest rate on the loan
- Borrower's annual income
- Borrower's total high credit limit
- Borrower's total open to buy on revolving bankcards
- Borrower's average current balance of all accounts

5. Logistic Regression Model

```r
logit_model <-
  glm(loan_status ~ int_rate + annual_inc + tot_hi_cred_lim + bc_open_to_buy + avg_cur_bal,
      family=binomial(link='logit'),
      data=train_data)

predicted <- plogis(predict(logit_model, test_data))

optCutOff <- optimalCutoff(test_data$loan_status, predicted)[1]

summary(logit_model)
```

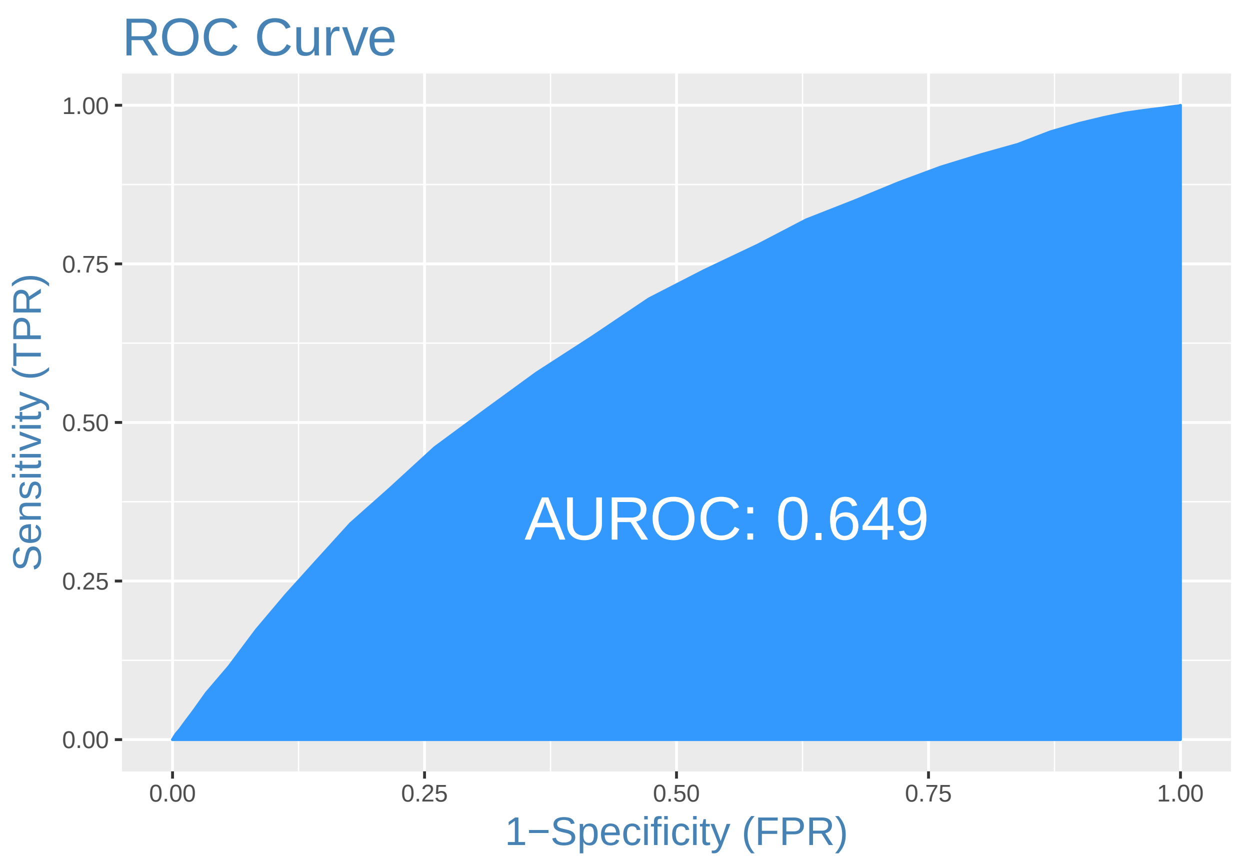
```
##
## Call:
## glm(formula = loan_status ~ int_rate + annual_inc + tot_hi_cred_lim +
##     bc_open_to_buy + avg_cur_bal, family = binomial(link = "logit"),
##     data = train_data)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.9101  -1.1254   0.2913   1.1007   2.8826
##
## Coefficients:
##                      Estimate    Std. Error z value             Pr(>|z|)
## (Intercept)     -1.2234222148  0.0606754001 -20.163 < 0.0000000000000002
## int_rate         0.1184911308  0.0037311375  31.757 < 0.0000000000000002
## annual_inc      -0.0000047136  0.0000003914 -12.044 < 0.0000000000000002
## tot_hi_cred_lim  0.0000005887  0.0000002045   2.879              0.00399
## bc_open_to_buy  -0.0000057945  0.0000013739  -4.218     0.00002467928527
## avg_cur_bal     -0.0000144803  0.0000020763  -6.974     0.00000000000308
##
## (Intercept)     ***
## int_rate        ***
## annual_inc      ***
## tot_hi_cred_lim **
## bc_open_to_buy  ***
## avg_cur_bal     ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 35478  on 25591  degrees of freedom
## Residual deviance: 33494  on 25586  degrees of freedom
## AIC: 33506
##
## Number of Fisher Scoring iterations: 4
```

```r
misClassError(test_data$loan_status, predicted, threshold = optCutOff)
```

```
## [1] 0.0464
```

```r
plotROC(test_data$loan_status, predicted)
```



I can assess the accuracy of the model in the below ways:

- The model summary result shows significance of each variable used.
- Misclassification error on test set is low.
- The model has fair amount of area under ROC curve.