glmnet_model_04

```
rm(list=ls())
loan_feature_selected <- read.csv('/Users/fanyang/Documents/lendingclub/2018_12_21/loan_feature_selected.csv
',
                  header = TRUE, stringsAsFactors = FALSE)

loan <- loan_feature_selected
```

```
loan_feature_selected$next_pymnt_L <- ifelse(loan_feature_selected$next_pymnt_binary == '0', 'yes', 'no')
```

```
str(loan_feature_selected)
```

```
## 'data.frame':    601779 obs. of  31 variables:
##  $ X                     : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ next_pymnt_binary     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ loan_amnt             : int  3000 7000 10000 12500 17500 14000 15300 6000 16000 16000 ...
##  $ int_rate              : num  12.7 16 16 12.7 17.3 ...
##  $ installment           : num  67.8 170.1 243 282.4 223.7 ...
##  $ annual_inc            : num  80000 47004 29120 27000 40000 ...
##  $ dti                   : num  17.9 23.5 22.8 16 19.5 ...
##  $ delinq_2yrs           : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ inq_last_6mths        : int  0 1 1 3 1 2 2 0 0 1 ...
##  $ mths_since_last_delinq: int  38 188 68 188 188 188 56 20 188 188 ...
##  $ open_acc              : int  15 7 11 6 5 5 14 6 8 6 ...
##  $ pub_rec               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ revol_bal             : num  27783 17726 16158 10143 10724 ...
##  $ total_acc             : int  38 11 31 24 6 10 27 17 16 23 ...
##  $ pymnt_pct             : num  1.081 1.162 1.162 1.081 0.612 ...
##  $ tot_cur_bal           : num  81079 81079 81079 81079 81079 ...
##  $ total_rev_hi_lim      : num  24300 24300 24300 24300 24300 24300 24300 24300 24300 24300 ...
##  $ cr_his_days           : int  5813 2344 5416 3652 2040 2344 10165 2586 5539 6757 ...
##  $ term                  : chr  " 60 months" " 60 months" " 60 months" " 60 months" ...
##  $ grade                 : chr  "B" "C" "C" "B" ...
##  $ emp_length            : chr  "1 year" "8 years" "2 years" "1 year" ...
##  $ home_ownership        : chr  "RENT" "RENT" "RENT" "RENT" ...
##  $ verification_status   : chr  "Verified" "Not Verified" "Verified" "Verified" ...
##  $ addr_state            : chr  "OR" "NC" "FL" "IL" ...
##  $ state_mean_int        : chr  "lowmmedium" "mediumhigh" "lowmmedium" "lowmmedium" ...
##  $ purpose               : chr  "other" "debt_consolidation" "debt_consolidation" "debt_consolidation" ..
.
##  $ initial_list_status   : chr  "f" "f" "f" "f" ...
##  $ income_level          : chr  "mediumhigh" "lowmedium" "low" "low" ...
##  $ delinq_binary         : chr  "no" "no" "no" "no" ...
##  $ late_fee_binary       : chr  "no" "no" "no" "no" ...
##  $ next_pymnt_L          : chr  "yes" "yes" "yes" "yes" ...
```

correlation of numerical features

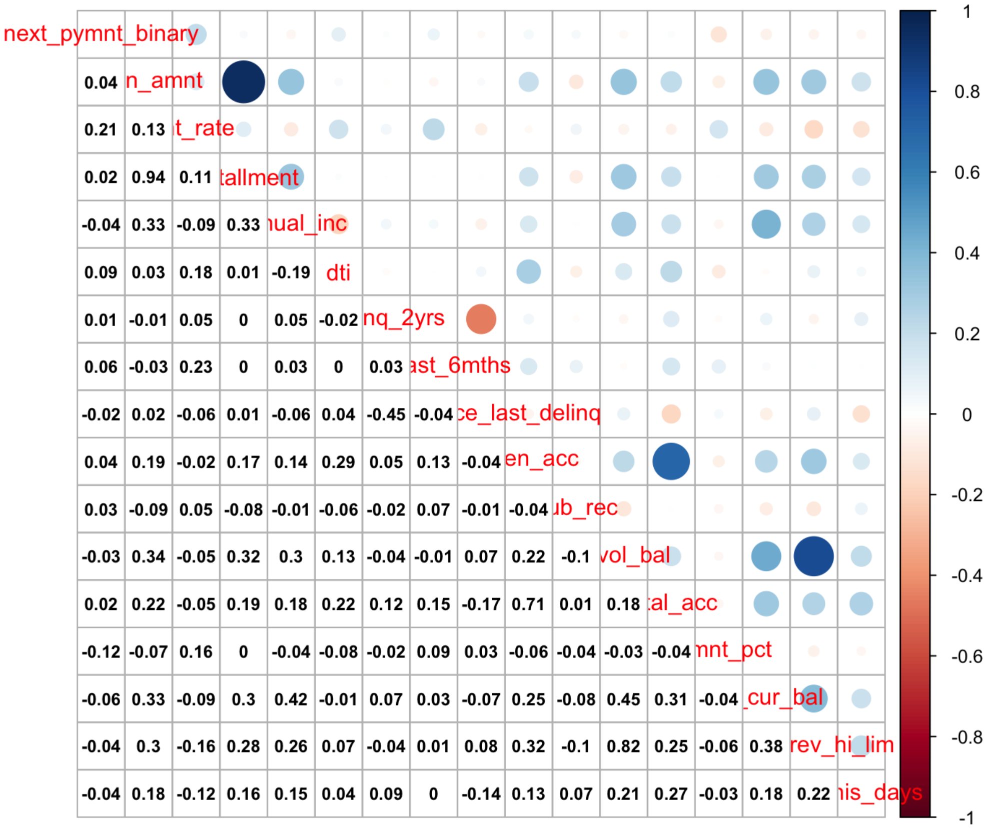
```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
correlations = cor(loan_feature_selected[, c('next_pymnt_binary',
                            'loan_amnt', 'int_rate', 'installment', 'annual_inc', 'dti', 'delinq_2yrs',
                            'inq_last_6mths', 'mths_since_last_delinq', 'open_acc', 'pub_rec', 'revol_bal',

                            'total_acc', 'pymnt_pct', 'tot_cur_bal', 'total_rev_hi_lim', 'cr_his_days')])

corrplot.mixed(correlations, lower.col = "black", number.cex=0.75)
```



prediction model by glmnet

```
# select features for modeling
# Tested the contribution of each feature to modeling before

loan.sub <- loan[,c('next_pymnt_binary',
                    'cr_his_days', 'loan_amnt', 'int_rate', 'installment', 'annual_inc',
                    'dti', 'inq_last_6mths', 'mths_since_last_delinq', 'open_acc', 'total_acc',
                    'pub_rec', 'revol_bal', 'pymnt_pct', 'tot_cur_bal', 'total_rev_hi_lim',
                    'term', 'grade', 'emp_length', 'home_ownership',
                    'state_mean_int' , 'initial_list_status', 'delinq_binary',
                    'late_fee_binary', 'verification_status')]
```

```
# split train and test dataset
train.ind <- sample(1:dim(loan.sub)[1], 0.7 * dim(loan.sub)[1])
train.sub <- loan.sub[train.ind, ]
test.sub <-  loan.sub[-train.ind, ]
```

```
# relevel categorical features
train.sub$state_mean_int <- relevel(as.factor(train.sub$state_mean_int), ref = 'low')

colnames(train.sub)
```

```
##  [1] "next_pymnt_binary"      "cr_his_days"
##  [3] "loan_amnt"              "int_rate"
##  [5] "installment"            "annual_inc"
##  [7] "dti"                    "inq_last_6mths"
##  [9] "mths_since_last_delinq" "open_acc"
## [11] "total_acc"              "pub_rec"
## [13] "revol_bal"              "pymnt_pct"
## [15] "tot_cur_bal"            "total_rev_hi_lim"
## [17] "term"                   "grade"
## [19] "emp_length"             "home_ownership"
## [21] "state_mean_int"         "initial_list_status"
## [23] "delinq_binary"          "late_fee_binary"
## [25] "verification_status"
```

```
# standardization of all numerical features
loan.sub.scale <- loan.sub
loan.sub.scale[, c(2,3,4,5,7,8,9,10,11,12,13,14,15,16)] <- scale(loan.sub.scale[, c(2,3,4,5,7,8,9,10,11,12,
13,14,15,16)])

train.sub.scale <- loan.sub.scale[train.ind, ]
test.sub.scale <-  loan.sub.scale[-train.ind, ]

train.ind = train.sub.scale[, -1]
train.ind <- model.matrix( ~., train.ind)
train.dep <- train.sub.scale[, 1]
```

```
library(glmnet)
```

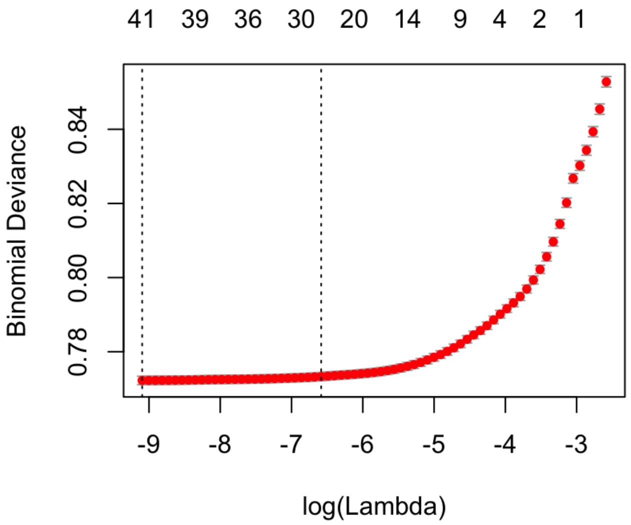
```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-16
```

```
logis.cvfit <- cv.glmnet(train.ind, train.dep, family = 'binomial')
```

```
plot(logis.cvfit)
```



```
# prediction
test.ind = test.sub.scale[, -1]
test.ind <- model.matrix( ~., test.ind)
test.dep <- test.sub.scale[, 1]
pred.cv <- predict(logis.cvfit, test.ind)
pred.cv <- as.numeric(pred.cv)
```

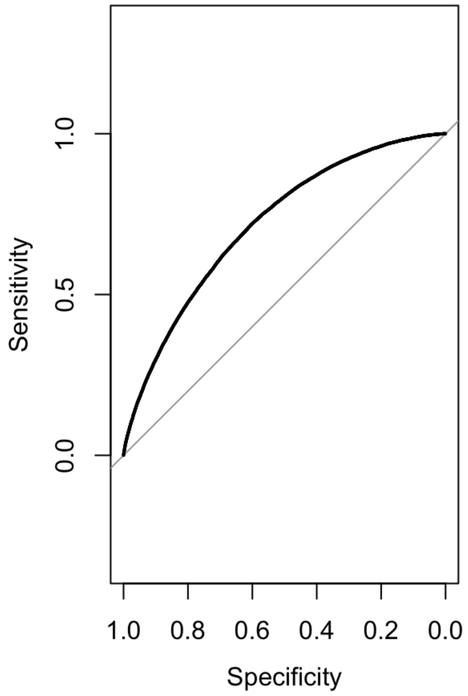
```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following object is masked from 'package:glmnet':
##
##     auc
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
par(mfrow = c(1, 2))
plot.roc(test.dep, pred.cv)
```



```
auc(test.dep, pred.cv)
```

```
## Area under the curve: 0.7191
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.