LoanRiskAnalysis_LoanStatus

Liang Tan

Read data

```
loan <- read.csv("loan.csv", stringsAsFactors = FALSE)
loanT <- loan
num.NA <- sort(sapply(loan, function(x) sum(is.na(x))), decreasing = TRUE)
remain.col = names(num.NA)[(num.NA < 0.8 * dim(loan)[1])]
delete.col = names(num.NA)[(num.NA >= 0.8 * dim(loan)[1])]
delete.col
```

```
##  [1] "dti_joint"               "annual_inc_joint"
##  [3] "il_util"                 "mths_since_rcnt_il"
##  [5] "open_acc_6m"             "open_il_6m"
##  [7] "open_il_12m"             "open_il_24m"
##  [9] "total_bal_il"            "open_rv_12m"
## [11] "open_rv_24m"             "max_bal_bc"
## [13] "all_util"                "inq_fi"
## [15] "total_cu_tl"             "inq_last_12m"
## [17] "mths_since_last_record"
```

Feature engineering and selection

Target variable pretreatment. I am only interested in making predictions for loans with status as "Current" or "Issued".

```
loan$loan_status <- gsub("Does not meet the credit policy. Status:", "", loan$loan_status)
loan <- subset(loan, !loan_status %in% c("Current", "Issued"))
loan$loan_status_binary <- with(loan, ifelse(loan_status == "Fully Paid", 1,
    0))
```

User feature selection
addr_state, emp_title, member_id, zip_code is removed
emp_length, home_ownership is reserved

```
# encode home_ownership
loan$home_ownership <- ifelse(loan$home_ownership %in% c("ANY", "NONE", "OTHER"),
    "OTHER", loan$home_ownership)
# encode state information with the help of int_rate
int_state <- by(loan, loan$addr_state, function(x) {
    return(mean(x$int_rate))
})

loan$state_mean_int <- ifelse(loan$addr_state %in% names(int_state)[which(int_state <=
    quantile(int_state, 0.25))], "low", ifelse(loan$addr_state %in% names(int_state)[which(int_state <=
    quantile(int_state, 0.5))], "lowmedium", ifelse(loan$addr_state %in% names(int_sta
    quantile(int_state, 0.75))], "mediumhigh", "high")))
select.features_1 <- c("home_ownership", "state_mean_int")
```

Financial feature selection
combine annual_inc and annual_inc_joint, dti and dti_joint, verification_status and verification_status_joint based on joint condition

```
loan$dti <- ifelse(!is.na(loan$dti_joint), loan$dti_joint, loan$dti)
loan$annual_inc <- ifelse(!is.na(loan$annual_inc_joint), loan$annual_inc_joint,
    loan$annual_inc)
loan$annual_inc[which(is.na(loan$annual_inc))] <- median(loan$annual_inc, na.rm = T)
loan$verification_status <- ifelse(loan$application_type == "JOINT", loan$verification_status_joint,
    loan$verification_status)
select.features_2 <- c("dti", "annual_inc", "verification_status")
```

Credit scores feature selection
inq_fi, inq_last_12m is removed for over 80% NA values.
The earliest_cr_line and last_credit_pull_d are removed for irrelavant.

credit lines feature selection
all_util, open_acc_6m, total_cu_tl, open_il_6m, open_il_12m, open_il_24m, open_rv_12m, open_rv_24m, max_bal_bc, mths_since_last_record, il_util, mths_since_rcnt_il, total_bal_il, max_bal_bc are removed for over 80% NA values
policy_code and url are removed for irrelavance
total_acc, tot_cur_bal, open_acc, acc_now_delinq, delinq_2yrs, mths_since_last_delinq, collections_12_mths_ex_med, tot_coll_amt, pub_rec, mths_since_last_major_derog, revol_util, total_rev_hi_lim are reserved

```
# mean and median are similar so I use mean for na
loan$total_acc[which(is.na(loan$total_acc))] <- mean(loan$total_acc, na.rm = T)
# mean of tot_cur_bal is more influenced by large value so I use median
loan$tot_cur_bal[which(is.na(loan$tot_cur_bal))] <- median(loan$tot_cur_bal,
    na.rm = T)
# mean and median are similar so I use mean for na
loan$open_acc[which(is.na(loan$open_acc))] <- mean(loan$open_acc, na.rm = T)
# acc_now_delinq is int number, so I use median for na
loan$acc_now_delinq[which(is.na(loan$acc_now_delinq))] <- median(loan$acc_now_delinq,
    na.rm = T)
# delinq_2yrs is int number, so I use median for na
loan$delinq_2yrs[which(is.na(loan$delinq_2yrs))] <- median(loan$delinq_2yrs,
    na.rm = T)
# mths_since_last_delinq is int number, so I use median for na
loan$mths_since_last_delinq[which(is.na(loan$mths_since_last_delinq))] <- median(loan$mths_since_last_de
    na.rm = T)
# collections_12_mths_ex_med is int number, so I use median for na
loan$collections_12_mths_ex_med[which(is.na(loan$collections_12_mths_ex_med))] <- median(loan$collectio
    na.rm = T)
# tot_coll_amt is int number, so I use median for na
loan$tot_coll_amt[which(is.na(loan$tot_coll_amt))] <- median(loan$tot_coll_amt,
    na.rm = T)
# pub_rec is int number, so I use median for na
loan$pub_rec[which(is.na(loan$pub_rec))] <- median(loan$pub_rec, na.rm = T)
# mths_since_last_major_derog is int number, so I use median for na
loan$mths_since_last_major_derog[which(is.na(loan$mths_since_last_major_derog))] <- median(loan$mths_si
    na.rm = T)
# mean and median is similar so I use mean for revol_util na values
loan$revol_util[which(is.na(loan$revol_util))] <- mean(loan$revol_util, na.rm = T)
# total_rev_hi_lim is int number, so I use median for na
loan$total_rev_hi_lim[which(is.na(loan$total_rev_hi_lim))] <- median(loan$total_rev_hi_lim,
```

```r
    na.rm = T)

select.features_3 <- c("total_acc", "tot_cur_bal", "open_acc", "acc_now_delinq",
    "delinq_2yrs", "mths_since_last_delinq", "collections_12_mths_ex_med", "tot_coll_amt",
    "pub_rec", "mths_since_last_major_derog", "revol_util", "total_rev_hi_lim")
```

loan feature selection
desc, id, title, issue_d, are removed
loan_amnt, application_type, purpose, term and initial_list_status are reserved

```r
select.features_4 <- c("loan_amnt", "application_type", "purpose", "term", "initial_list_status")
```

loan payment feature selection
last_pymnt_amnt, last_pymnt_d, next_pymnt_d, total_pymnt, total_pymnt_inv, total_rec_int, total_rec_late_fee, total_rec_prncp are inrrelative here
installment, funded_amnt, funded_amnt_inv, pymnt_plan, recoveries collection_recovery_fee, out_prncp, out_prncp_inv are reserved

```r
select.features_5 <- c("installment", "funded_amnt", "funded_amnt_inv", "pymnt_plan",
    "recoveries", "collection_recovery_fee", "out_prncp", "out_prncp_inv")
```

grade and int_rate are used as well

```r
select.features <- c(select.features_1, select.features_2, select.features_3,
    select.features_4, select.features_5, "grade", "int_rate", "loan_status_binary")
loan <- loan[select.features]
```

scale all numeric variables

```r
select.features.num <- names(loan[, sapply(loan[, 1:32], is.numeric)])
loan.scale <- loan
loan.scale[, select.features.num] <- scale(loan.scale[, select.features.num])
```

check the level of all category variables

```r
select.features.cate <- names(loan.scale[, sapply(loan.scale, is.character)])
n_levels <- sort(sapply(loan.scale[select.features.cate], function(x) {
    nlevels(as.factor(x))
}), decreasing = TRUE)
print(n_levels)
```

```
##             purpose               grade      home_ownership
##                  14                   7                   4
##      state_mean_int verification_status    application_type
##                   4                   3                   2
##                term initial_list_status          pymnt_plan
##                   2                   2                   2
```

train, test split

```r
set.seed(1)
train.ind <- sample(1:dim(loan)[1], 0.8 * dim(loan)[1])
train.sub <- loan.scale[train.ind, ]
test.sub <- loan.scale[-train.ind, ]
```

model train

```r
logis.mod <- glm(loan_status_binary ~ ., train.sub, family = "binomial")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(logis.mod)
```

```
##
## Call:
## glm(formula = loan_status_binary ~ ., family = "binomial", data = train.sub)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -8.49    0.00    0.00    0.00    8.49
##
## Coefficients:
##                                      Estimate Std. Error    z value
## (Intercept)                        -7.164e+13  1.534e+06 -4.670e+07
## home_ownershipOTHER                 2.981e+14  4.976e+06  5.990e+07
## home_ownershipOWN                   3.907e+13  5.365e+05  7.283e+07
## home_ownershipRENT                 -7.615e+12  3.482e+05 -2.187e+07
## state_mean_intlow                   3.496e+13  5.209e+05  6.711e+07
## state_mean_intlowmedium             4.111e+13  4.410e+05  9.323e+07
## state_mean_intmediumhigh           -3.050e+13  4.572e+05 -6.671e+07
## dti                                -6.653e+13  1.646e+05 -4.043e+08
## annual_inc                          1.402e+14  1.599e+05  8.768e+08
## verification_statusSource Verified -4.286e+13  3.782e+05 -1.133e+08
## verification_statusVerified        -1.939e+13  3.782e+05 -5.127e+07
## total_acc                           4.558e+13  2.034e+05  2.241e+08
## tot_cur_bal                         5.485e+13  1.846e+05  2.972e+08
## open_acc                           -3.836e+13  2.064e+05 -1.859e+08
## acc_now_delinq                      1.364e+12  1.453e+05  9.389e+06
## delinq_2yrs                        -1.092e+13  1.693e+05 -6.450e+07
## mths_since_last_delinq              1.293e+12  1.762e+05  7.338e+06
## collections_12_mths_ex_med         -6.929e+12  1.446e+05 -4.793e+07
## tot_coll_amt                        1.071e+14  1.771e+06  6.050e+07
## pub_rec                            -3.622e+09  1.477e+05 -2.453e+04
## mths_since_last_major_derog         4.265e+12  1.615e+05  2.641e+07
## revol_util                         -3.855e+13  1.691e+05 -2.279e+08
## total_rev_hi_lim                    9.175e+12  1.705e+05  5.381e+07
## loan_amnt                          -5.946e+13  2.183e+06 -2.724e+07
## application_typeJOINT               5.049e+14  3.357e+07  1.504e+07
## purposecredit_card                  3.787e+13  1.277e+06  2.966e+07
## purposedebt_consolidation          -8.095e+12  1.248e+06 -6.485e+06
## purposeeducational                 -1.634e+14  3.808e+06 -4.292e+07
## purposehome_improvement            -3.527e+13  1.365e+06 -2.584e+07
## purposehouse                        6.697e+13  2.160e+06  3.100e+07
## purposemajor_purchase              -3.597e+13  1.524e+06 -2.360e+07
## purposemedical                     -1.767e+14  1.822e+06 -9.698e+07
## purposemoving                      -2.046e+14  2.011e+06 -1.017e+08
## purposeother                       -1.624e+14  1.366e+06 -1.189e+08
## purposerenewable_energy            -5.239e+13  4.611e+06 -1.136e+07
## purposesmall_business              -3.418e+14  1.616e+06 -2.115e+08
## purposevacation                    -1.099e+14  2.189e+06 -5.022e+07
## purposewedding                      9.182e+13  2.069e+06  4.438e+07
## term 60 months                      1.387e+14  9.217e+05  1.505e+08
## initial_list_statusw               -7.860e+13  3.238e+05 -2.427e+08
## installment                         1.093e+14  1.122e+06  9.738e+07
```

```
## funded_amnt                           -4.508e+14  2.818e+06 -1.600e+08
## funded_amnt_inv                        4.302e+14  1.304e+06  3.298e+08
## pymnt_plany                           -3.127e+14  2.742e+07 -1.141e+07
## recoveries                            -1.122e+15  2.476e+05 -4.530e+09
## collection_recovery_fee                3.438e+14  2.465e+05  1.395e+09
## out_prncp                              6.014e+15  8.759e+07  6.866e+07
## out_prncp_inv                         -7.056e+15  8.759e+07 -8.055e+07
## gradeB                                -2.244e+14  6.258e+05 -3.586e+08
## gradeC                                -2.133e+14  8.928e+05 -2.389e+08
## gradeD                                -2.870e+14  1.192e+06 -2.407e+08
## gradeE                                -3.409e+14  1.511e+06 -2.257e+08
## gradeF                                -3.712e+14  1.925e+06 -1.928e+08
## gradeG                                -5.212e+14  2.441e+06 -2.135e+08
## int_rate                              -8.092e+13  4.969e+05 -1.628e+08
##                                       Pr(>|z|)
## (Intercept)                             <2e-16 ***
## home_ownershipOTHER                     <2e-16 ***
## home_ownershipOWN                       <2e-16 ***
## home_ownershipRENT                      <2e-16 ***
## state_mean_intlow                       <2e-16 ***
## state_mean_intlowmedium                 <2e-16 ***
## state_mean_intmediumhigh                <2e-16 ***
## dti                                     <2e-16 ***
## annual_inc                              <2e-16 ***
## verification_statusSource Verified      <2e-16 ***
## verification_statusVerified             <2e-16 ***
## total_acc                               <2e-16 ***
## tot_cur_bal                             <2e-16 ***
## open_acc                                <2e-16 ***
## acc_now_delinq                          <2e-16 ***
## delinq_2yrs                             <2e-16 ***
## mths_since_last_delinq                  <2e-16 ***
## collections_12_mths_ex_med              <2e-16 ***
## tot_coll_amt                            <2e-16 ***
## pub_rec                                 <2e-16 ***
## mths_since_last_major_derog             <2e-16 ***
## revol_util                              <2e-16 ***
## total_rev_hi_lim                        <2e-16 ***
## loan_amnt                               <2e-16 ***
## application_typeJOINT                   <2e-16 ***
## purposecredit_card                      <2e-16 ***
## purposedebt_consolidation               <2e-16 ***
## purposeeducational                      <2e-16 ***
## purposehome_improvement                 <2e-16 ***
## purposehouse                            <2e-16 ***
## purposemajor_purchase                   <2e-16 ***
## purposemedical                          <2e-16 ***
## purposemoving                           <2e-16 ***
## purposeother                            <2e-16 ***
## purposerenewable_energy                 <2e-16 ***
## purposesmall_business                   <2e-16 ***
## purposevacation                         <2e-16 ***
## purposewedding                          <2e-16 ***
## term 60 months                          <2e-16 ***
```

```
## initial_list_statusw                     <2e-16 ***
## installment                              <2e-16 ***
## funded_amnt                              <2e-16 ***
## funded_amnt_inv                          <2e-16 ***
## pymnt_plany                              <2e-16 ***
## recoveries                               <2e-16 ***
## collection_recovery_fee                  <2e-16 ***
## out_prncp                                <2e-16 ***
## out_prncp_inv                            <2e-16 ***
## gradeB                                   <2e-16 ***
## gradeC                                   <2e-16 ***
## gradeD                                   <2e-16 ***
## gradeE                                   <2e-16 ***
## gradeF                                   <2e-16 ***
## gradeG                                   <2e-16 ***
## int_rate                                 <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:  245712  on 221711  degrees of freedom
## Residual deviance: 4163402  on 221657  degrees of freedom
## AIC: 4163512
##
## Number of Fisher Scoring iterations: 25
```

evaluate model

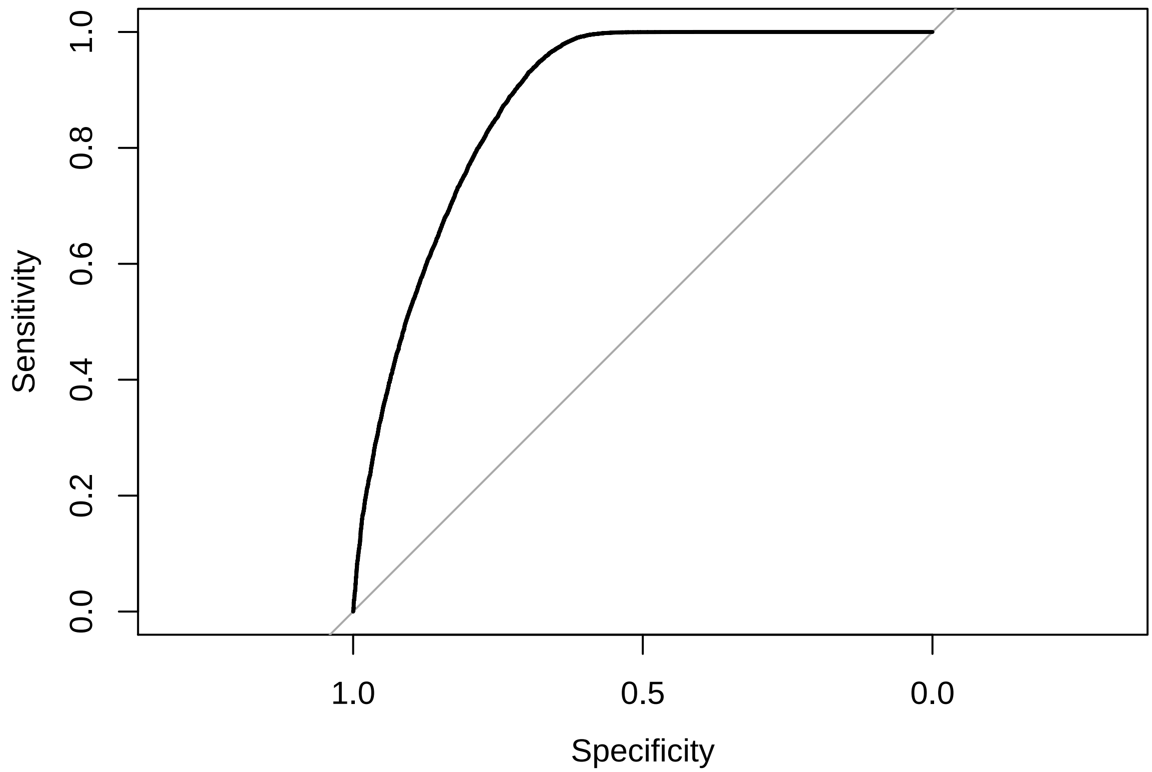
```
library(pROC)
```

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

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

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

```
pred <- predict(logis.mod, test.sub[, 1:32])
plot.roc(test.sub$loan_status_binary, pred)
```

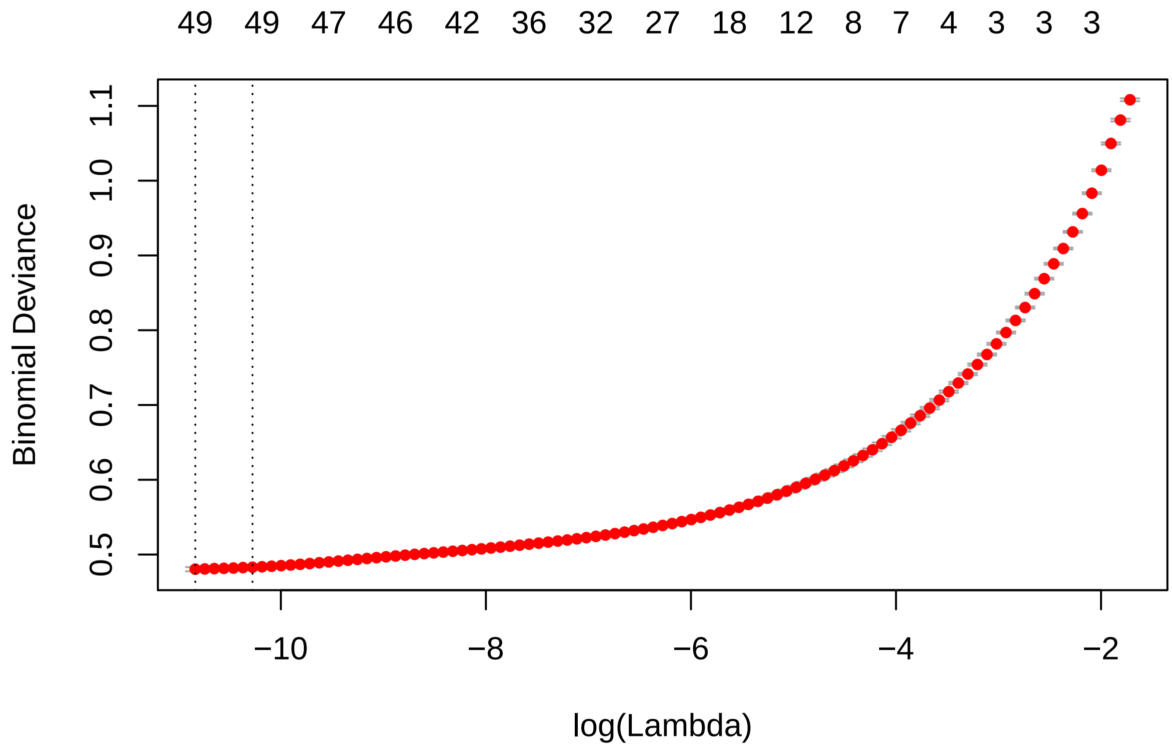


The model seems not good so I will add regularization to make it better

```
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-13

##
## Attaching package: 'glmnet'

## The following object is masked from 'package:pROC':
##
##     auc
```

```
test.matrix <- model.matrix(~., test.sub[, 1:32])
ind <- train.sub[, 1:32]
ind <- model.matrix(~., ind)
dep <- train.sub[, "loan_status_binary"]
# Use cross validation to tune parameters
logis.cvfit <- cv.glmnet(ind, dep, family = "binomial")
plot(logis.cvfit)
```

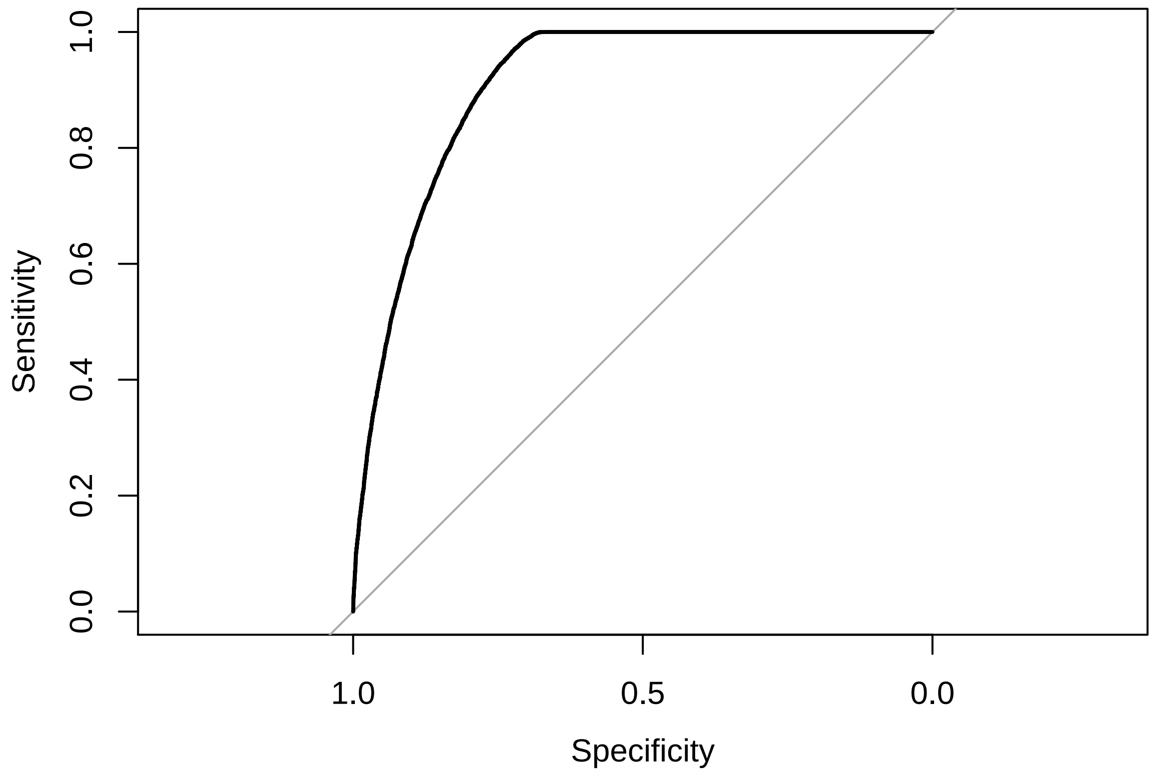



```r
print(paste("The optimus lambda for model is", logis.cvfit$logis.cvfit$lambda.1se))
```

```
## [1] "The optimus lambda for model is "
```

```r
cv.pred <- predict(logis.cvfit, s = logis.cvfit$lambda.1se, newx = test.matrix)
plot.roc(test.sub$loan_status_binary, cv.pred)
```

```
## Warning in roc.default(x, predictor, plot = TRUE, ...): Deprecated use
## a matrix as predictor. Unexpected results may be produced, please pass a
## numeric vector.
```



coefficients with this model is

```
print(coef(logis.cvfit, s = "lambda.1se"))
```

```
## 56 x 1 sparse Matrix of class "dgCMatrix"
##                                            1
## (Intercept)                    -2.740352e+01
## (Intercept)                     .
## home_ownershipOTHER             2.035858e-01
## home_ownershipOWN               2.828198e-03
## home_ownershipRENT             -9.589400e-02
## state_mean_intlow               1.208202e-01
## state_mean_intlowmedium         1.733891e-01
## state_mean_intmediumhigh       -2.864538e-02
## dti                            -2.579556e-01
## annual_inc                      3.045841e-01
## verification_statusSource Verified -1.135178e-01
## verification_statusVerified    -1.747696e-02
## total_acc                       8.265866e-02
## tot_cur_bal                     1.047427e-01
## open_acc                       -8.300373e-02
## acc_now_delinq                  6.117109e-04
## delinq_2yrs                    -2.236134e-02
## mths_since_last_delinq          2.622951e-02
## collections_12_mths_ex_med     -2.162677e-02
## tot_coll_amt                    5.831161e-03
## pub_rec                        -3.269206e-02
## mths_since_last_major_derog     2.807576e-03
## revol_util                     -7.486259e-02
## total_rev_hi_lim                1.904605e-02
```

```
## loan_amnt                        -1.046023e-01
## application_typeJOINT              .
## purposecredit_card               -1.058032e-01
## purposedebt_consolidation        -1.404312e-01
## purposeeducational               -2.926918e-01
## purposehome_improvement          -1.808174e-01
## purposehouse                      1.206670e-01
## purposemajor_purchase            -7.535288e-02
## purposemedical                   -3.793945e-01
## purposemoving                    -3.113796e-01
## purposeother                     -2.366905e-01
## purposerenewable_energy          -1.632192e-01
## purposesmall_business            -6.116772e-01
## purposevacation                  -5.824432e-02
## purposewedding                    2.769736e-01
## term 60 months                   -4.868740e-01
## initial_list_statusw             -3.789032e-01
## installment                      -1.803778e-01
## funded_amnt                      -5.171280e-01
## funded_amnt_inv                   7.449352e-01
## pymnt_plany                        .
## recoveries                       -1.205841e+02
## collection_recovery_fee            .
## out_prncp                        -2.461443e+01
## out_prncp_inv                      .
## gradeB                           -1.751439e-01
## gradeC                           -3.071449e-01
## gradeD                           -2.969414e-01
## gradeE                           -2.175125e-01
## gradeF                             .
## gradeG                           -1.877935e-02
## int_rate                         -3.442488e-01
```