Project 2

Clayton Chan

2024-11-11

```r
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
df <- read.csv("~/Downloads/Loan_default.csv", stringsAsFactors = T)
df <- df[,-1] #Drop ID as it is not necessary
df$Education <- factor(df$Education,levels = c("High School","Bachelor's","Master's","PhD")) #Reorder L
df$MaritalStatus <- factor(df$MaritalStatus,levels = c("Single","Married","Divorced"))
```

```r
set.seed(123)
idx <- sample(29653)
balance <- df[df$Default == 0,]
df <- rbind(df[df$Default == 1,],balance[idx,]) #Balance dataset
```

```r
set.seed(123)
index <- createDataPartition(df$Default, p=.8, list=FALSE, times=1)
train <- df[index,]
test <- df[-index,] #Create train and test split
```

```r
logreg <- glm(Default~.,family = 'binomial',data = train)
summary(logreg)
```

```
##
## Call:
## glm(formula = Default ~ ., family = "binomial", data = train)
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               1.593e+00  8.305e-02  19.182  < 2e-16 ***
## Age                      -3.962e-02  7.099e-04 -55.817  < 2e-16 ***
## Income                   -8.329e-06  2.573e-07 -32.371  < 2e-16 ***
## LoanAmount                3.993e-06  1.452e-07  27.508  < 2e-16 ***
## CreditScore              -7.954e-04  6.421e-05 -12.387  < 2e-16 ***
## MonthsEmployed           -9.871e-03  2.988e-04 -33.036  < 2e-16 ***
## NumCreditLines            9.820e-02  9.103e-03  10.787  < 2e-16 ***
## InterestRate              6.928e-02  1.577e-03  43.947  < 2e-16 ***
## LoanTerm                 -5.801e-04  6.009e-04  -0.965 0.334382
```

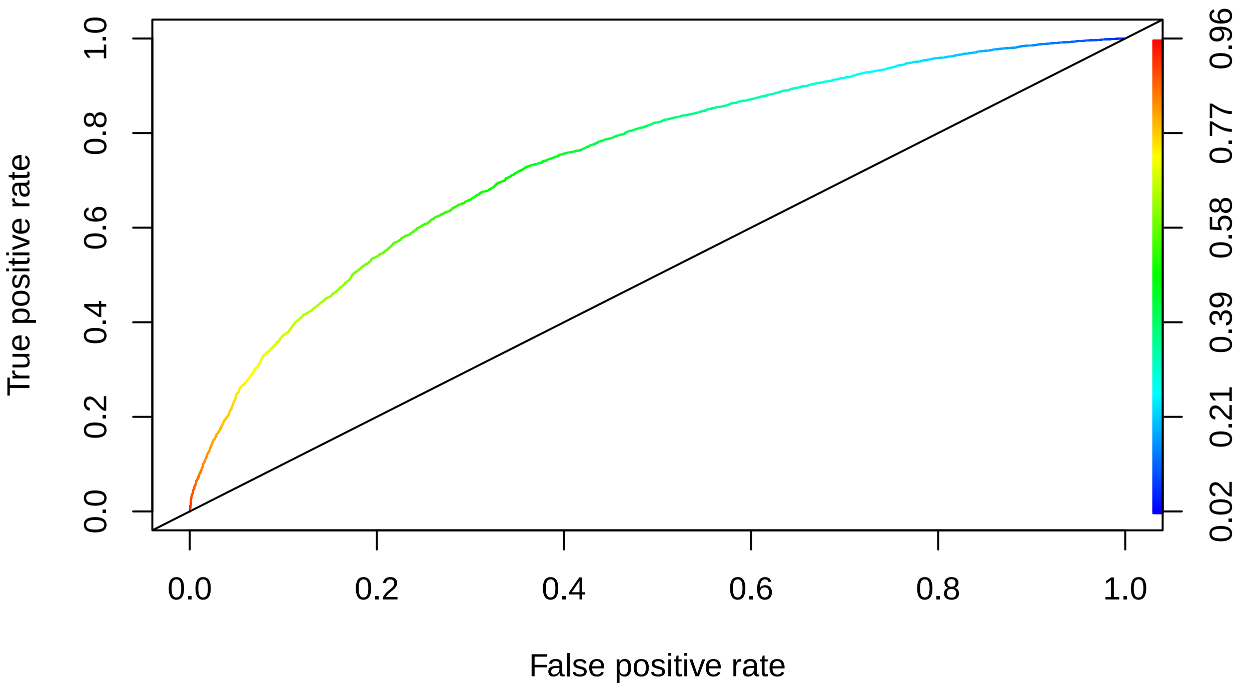
```
## DTIRatio                    2.543e-01  4.435e-02   5.734 9.80e-09 ***
## EducationBachelor's         -8.470e-02  2.826e-02  -2.997 0.002730 **
## EducationMaster's           -2.208e-01  2.875e-02  -7.680 1.59e-14 ***
## EducationPhD                -2.886e-01  2.882e-02 -10.015  < 2e-16 ***
## EmploymentTypePart-time      2.985e-01  2.953e-02  10.108  < 2e-16 ***
## EmploymentTypeSelf-employed  2.211e-01  2.952e-02   7.489 6.92e-14 ***
## EmploymentTypeUnemployed     4.582e-01  2.914e-02  15.722  < 2e-16 ***
## MaritalStatusMarried        -1.505e-01  2.524e-02  -5.964 2.47e-09 ***
## MaritalStatusDivorced        3.344e-02  2.459e-02   1.360 0.173847
## HasMortgageYes              -1.605e-01  2.038e-02  -7.873 3.46e-15 ***
## HasDependentsYes            -2.389e-01  2.040e-02 -11.710  < 2e-16 ***
## LoanPurposeBusiness          9.817e-02  3.178e-02   3.089 0.002008 **
## LoanPurposeEducation        -2.331e-03  3.192e-02  -0.073 0.941792
## LoanPurposeHome             -1.250e-01  3.262e-02  -3.832 0.000127 ***
## LoanPurposeOther             1.387e-02  3.198e-02   0.434 0.664489
## HasCoSignerYes              -2.847e-01  2.040e-02 -13.955  < 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: 65774  on 47445  degrees of freedom
## Residual deviance: 56187  on 47421  degrees of freedom
## AIC: 56237
##
## Number of Fisher Scoring iterations: 3
```

```r
probs <- predict(logreg, test, type = 'response')
pred <- ifelse(probs>=.5,1,0)
confusionMatrix(data = factor(pred),reference = factor(test$Default))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4022 1886
##          1 1908 4044
##
##                Accuracy : 0.6801
##                  95% CI : (0.6716, 0.6885)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.3602
##
##  Mcnemar's Test P-Value : 0.7332
##
##             Sensitivity : 0.6782
##             Specificity : 0.6820
##          Pos Pred Value : 0.6808
##          Neg Pred Value : 0.6794
##              Prevalence : 0.5000
##          Detection Rate : 0.3391
##    Detection Prevalence : 0.4981
```

```
##       Balanced Accuracy : 0.6801
##
##        'Positive' Class : 0
##
```

```
library(ROCR)
pred_m1 <- prediction(probs, test$Default)
roc_curve <- performance(pred_m1, "tpr","fpr")
plot(roc_curve, colorize=T)
abline(0, 1)
```




```
auc_ROCR <- performance(pred_m1, measure = "auc")
(auc_ROCR <- auc_ROCR@y.values[[1]]) #Plot ROC curve and AUC
```

```
## [1] 0.7439145
```

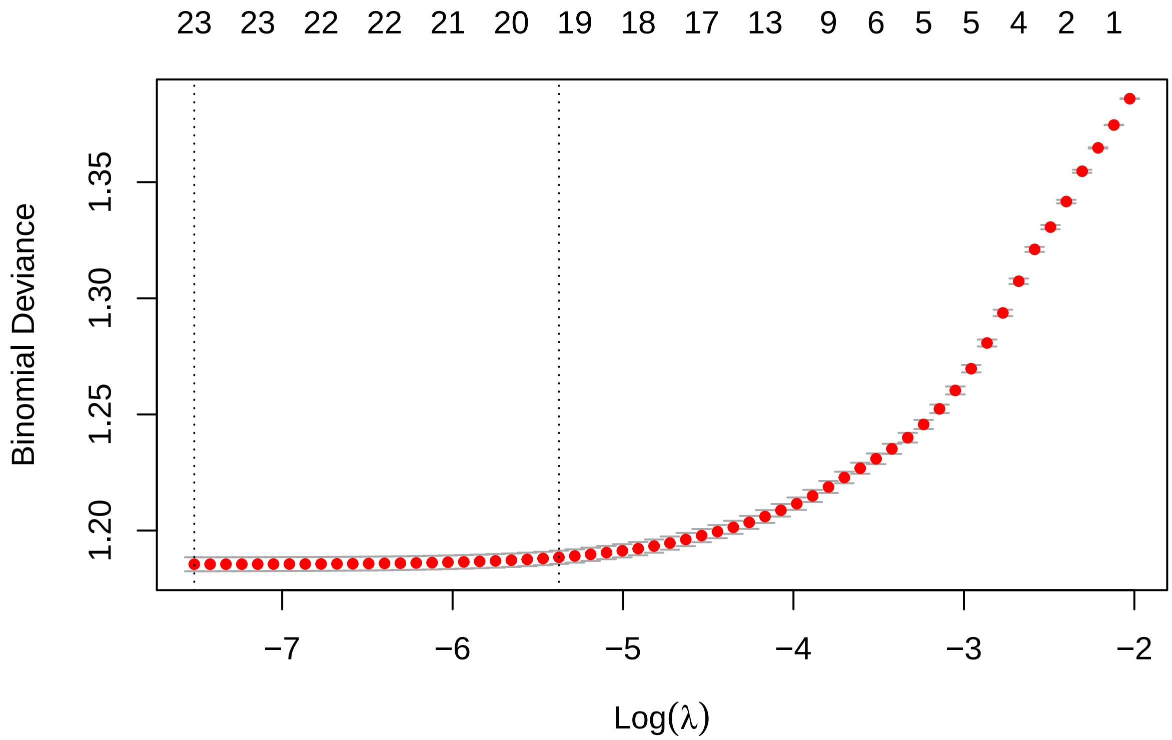
```
library(glmnet)
```

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

```
## Loaded glmnet 4.1-8
```

```
x <- model.matrix(Default~., data = train)[,-1]
glmmod <- glmnet(x, y=as.factor(train$Default), alpha=1, family="binomial") #Fit LASSO
```

```
cv.glmmod <- cv.glmnet(x, y=as.factor(train$Default), alpha=1, family="binomial")
plot(cv.glmmod)
```




```r
(best.lambda <- cv.glmmod$lambda.min) #Find a lambda that balances accurate predictions and parsimony
```

```
## [1] 0.0005444213
```

```r
glmmod <- glmnet(x, y=as.factor(train$Default), alpha=1, family="binomial", lambda = 0.0265)
coef(glmmod) #After experimenting .0265 balances parsimony and accuracy really well
```

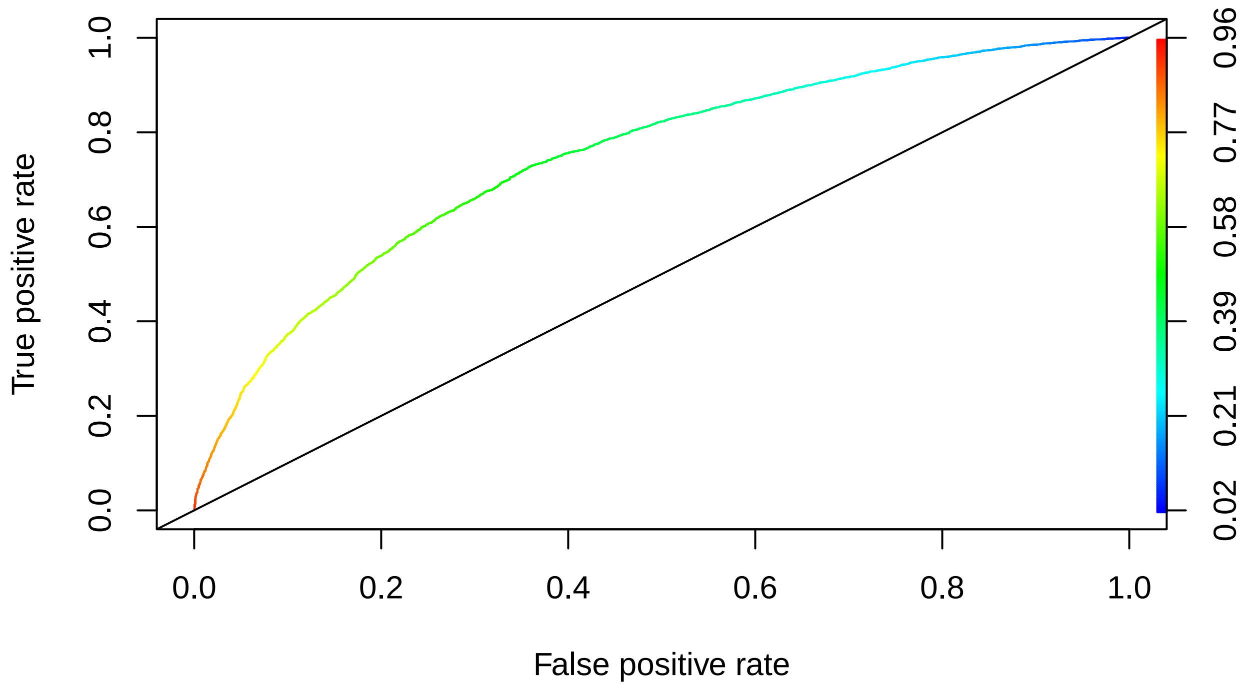
```
## 25 x 1 sparse Matrix of class "dgCMatrix"
##                                         s0
## (Intercept)                   9.205784e-01
## Age                          -2.908833e-02
## Income                       -4.910185e-06
## LoanAmount                    2.126419e-06
## CreditScore                   .
## MonthsEmployed               -5.733888e-03
## NumCreditLines                .
## InterestRate                  4.699555e-02
## LoanTerm                      .
## DTIRatio                      .
## EducationBachelor's           .
## EducationMaster's             .
## EducationPhD                  .
## EmploymentTypePart-time       .
## EmploymentTypeSelf-employed   .
## EmploymentTypeUnemployed      .
## MaritalStatusMarried          .
## MaritalStatusDivorced         .
## HasMortgageYes                .
## HasDependentsYes              .
```

```
## LoanPurposeBusiness           .
## LoanPurposeEducation          .
## LoanPurposeHome               .
## LoanPurposeOther              .
## HasCoSignerYes              -3.533869e-02
```

```r
xtest <- model.matrix(Default~., data = test)[,-1]
probs <- predict(glmmod, newx = xtest, type = 'response')
pred <- ifelse(probs>=.5,1,0)
confusionMatrix(data = factor(pred),reference = factor(test$Default))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3943 1897
##          1 1987 4033
##
##                Accuracy : 0.6725
##                  95% CI : (0.664, 0.681)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.345
##
##  Mcnemar's Test P-Value : 0.1533
##
##             Sensitivity : 0.6649
##             Specificity : 0.6801
##          Pos Pred Value : 0.6752
##          Neg Pred Value : 0.6699
##              Prevalence : 0.5000
##          Detection Rate : 0.3325
##    Detection Prevalence : 0.4924
##       Balanced Accuracy : 0.6725
##
##        'Positive' Class : 0
##
```

```r
pred_m2 <- prediction(probs, test$Default)
roc_curve <- performance(pred_m1, "tpr","fpr")
plot(roc_curve, colorize=T)
abline(0, 1)
```




```r
auc_ROCR <- performance(pred_m2, measure = "auc")
(auc_ROCR <- auc_ROCR@y.values[[1]])
```

```
## [1] 0.7315365
```