Predictive Modelling for Bank Loan Accounts

Maria Shaikh

2023-03-16

Data Preprocessing 1. Reading the data and creating dummy columns

```
credit<- read.csv("creditv2.csv", header = TRUE)
credit$AMOUNT_REQUESTED <- as.numeric(sub(',','',as.character(credit$AMOUNT_REQUESTED),fixed=TRUE))
#
credit$AMOUNT_REQUESTED <- as.numeric(credit$AMOUNT_REQUESTED)
credit$NPV <- as.numeric(credit$NPV)
credit$PROFITABLE <- ifelse(credit$NPV>0,1,0)
#
credit$CHK_ACCT_0 <- ifelse(credit$CHK_ACCT==0,1,0)
credit$CHK_ACCT_1 <- ifelse(credit$CHK_ACCT==1,1,0)
credit$CHK_ACCT_2 <- ifelse(credit$CHK_ACCT==2,1,0)
credit$CHK_ACCT_3 <- ifelse(credit$CHK_ACCT==3,1,0)
#
credit$SAV_ACCT_0 <- ifelse(credit$SAV_ACCT==0,1,0)
credit$SAV_ACCT_1 <- ifelse(credit$SAV_ACCT==1,1,0)
credit$SAV_ACCT_2 <- ifelse(credit$SAV_ACCT==2,1,0)
credit$SAV_ACCT_3 <- ifelse(credit$SAV_ACCT==3,1,0)
credit$SAV_ACCT_4 <- ifelse(credit$SAV_ACCT==4,1,0)
#
credit$HISTORY_0 <- ifelse(credit$HISTORY==0,1,0)
credit$HISTORY_1 <- ifelse(credit$HISTORY==1,1,0)
credit$HISTORY_2 <- ifelse(credit$HISTORY==2,1,0)
credit$HISTORY_3 <- ifelse(credit$HISTORY==3,1,0)
credit$HISTORY_4 <- ifelse(credit$HISTORY==4,1,0)
#
credit$JOB_0 <- ifelse(credit$JOB==0,1,0)
credit$JOB_1 <- ifelse(credit$JOB==1,1,0)
credit$JOB_2 <- ifelse(credit$JOB==2,1,0)
credit$JOB_3 <- ifelse(credit$JOB==3,1,0)
#
credit$TYPE_0 <- ifelse(credit$TYPE==0,1,0)
credit$TYPE_1 <- ifelse(credit$TYPE==1,1,0)
credit$TYPE_2 <- ifelse(credit$TYPE==2,1,0)
credit$TYPE_3 <- ifelse(credit$TYPE==3,1,0)
credit$TYPE_4 <- ifelse(credit$TYPE==4,1,0)
credit$TYPE_5 <- ifelse(credit$TYPE==5,1,0)
credit$TYPE_6 <- ifelse(credit$TYPE==6,1,0)
#
#
credit$CREDIT_EXTENDED <- NULL
credit$OBS. <- NULL
credit$CHK_ACCT <- NULL
```

```
credit$SAV_ACCT <- NULL
credit$HISTORY <- NULL
credit$JOB <- NULL
credit$TYPE <- NULL
credit$NPV <- NULL
```

Scaling the data

```
fun <- function(x){
  a <- mean(x)
  b <- sd(x)
  (x - a)/(b)
}

M = data.frame(apply(credit,2,mean))
M = t(M)
M<-M[,-16]
SD = data.frame(apply(credit, 2, sd))
SD = t(SD)
SD<-SD[,-16]

credit[,1:15] <- apply(credit[,1:15], 2, fun)
credit[,17:41] <- apply(credit[,17:41], 2, fun)
```

Splitting data into training, tesrting and validation datasets

```
set.seed(123)

inTrain <- sample(nrow(credit), 0.6*nrow(credit))
#
dftrain <- data.frame(credit[inTrain,])
dftemp <- data.frame(credit[-inTrain,])
inVal <- sample(nrow(dftemp),0.5*nrow(dftemp))
dfvalidation <- data.frame(dftemp[inVal,])
dftest <- data.frame(dftemp[-inVal,])
dftemp <- NULL
```

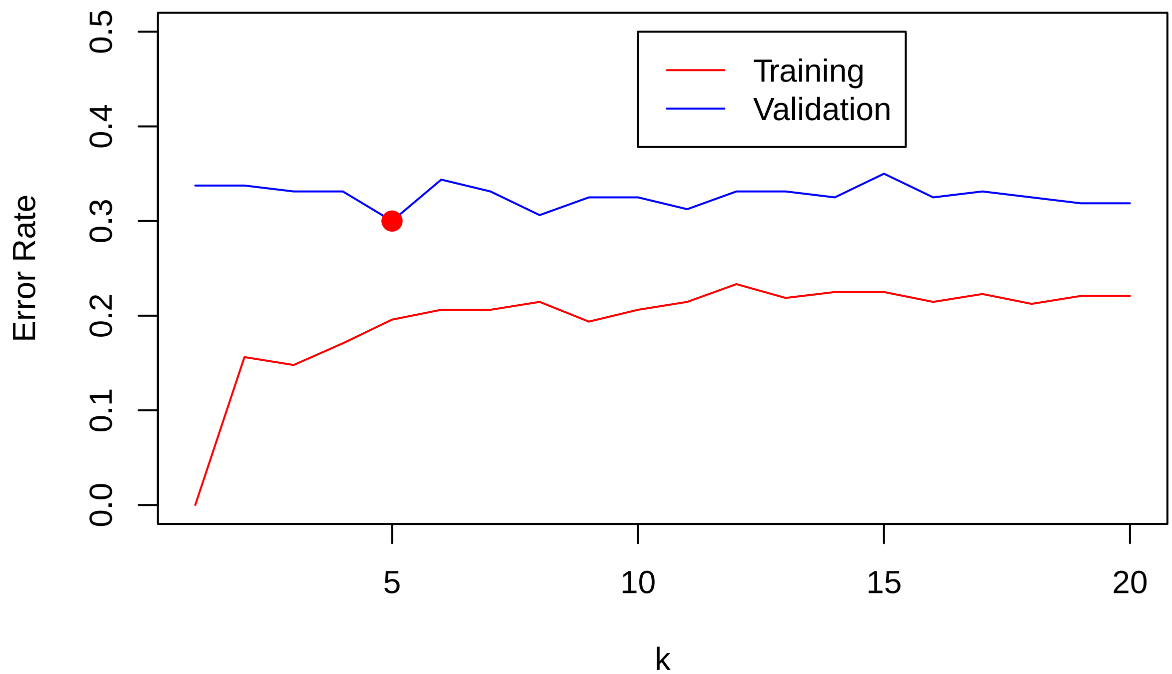
KNN Model

```
library(class)

train_input <- as.matrix(dftrain[,-16])
train_output <- as.vector(dftrain[,16])
validate_input <- as.matrix(dfvalidation[,-16])
test_input <- as.matrix(dftest[,-16])

kmax <- 20
ER1 <- rep(0,kmax)
ER2 <- rep(0,kmax)

set.seed(457)
for (i in 1:kmax){
  prediction <- knn(train_input, train_input,train_output, k=i)
  prediction2 <- knn(train_input, validate_input,train_output, k=i)
  prediction3 <- knn(train_input, test_input,train_output, k=i)
```

```
  # The confusion matrix for training data is:
  CM1 <- table( dftrain$PROFITABLE, prediction)
  # The training error rate is:
  ER1[i] <- (CM1[1,2]+CM1[2,1])/sum(CM1)
  # The confusion matrix for validation data is:
  CM2 <- table( dfvalidation$PROFITABLE, prediction2)
  ER2[i] <- (CM2[1,2]+CM2[2,1])/sum(CM2)
}


plot(c(1,kmax),c(0,0.5),type="n", xlab="k",ylab="Error Rate")
lines(ER1,col="red")
lines(ER2,col="blue")
legend(10, 0.5, c("Training","Validation"),lty=c(1,1), col=c("red","blue"))
z <- which.min(ER2)
cat("Minimum Validation Error k:", z)
```

```
## Minimum Validation Error k: 5
```

```
points(z,ER2[z],col="red",cex=2,pch=20)
```




```
# Scoring at optimal k
prediction <- knn(train_input, train_input,train_output, k=z)
prediction2 <- knn(train_input, validate_input,train_output, k=z)
prediction3 <- knn(train_input, test_input,train_output, k=z)
#
CM1 <- table( dftrain$PROFITABLE, prediction)
CM2 <- table( dfvalidation$PROFITABLE, prediction2)
CM3 <- table( dftest$PROFITABLE, prediction3)
CM1
```

```
##    prediction
##      0   1
##   0  62  67
```

```
##   1  27 324
```

```r
CM2
```

```
##    prediction2
##       0   1
##   0 18 38
##   1 12 92
```

```r
CM3
```

```
##    prediction3
##       0   1
##   0  18  31
##   1   8 103
```

```r
(ER1 <- (CM1[1,2]+CM1[2,1])/sum(CM1))
```

```
## [1] 0.1958333
```

```r
(ER2 <- (CM2[1,2]+CM2[2,1])/sum(CM2))
```

```
## [1] 0.3125
```

```r
(ER3 <- (CM3[1,2]+CM3[2,1])/sum(CM3))
```

```
## [1] 0.24375
```

```r
#PPV

PPVt = CM3[2,2]/(CM3[1,2]+CM3[2,2])
cat("PPV:", PPVt, "\n")
```

```
## PPV: 0.7686567
```

```r
# NPV
NPVt = (CM3[1,1]/(CM3[2,1]+CM3[1,1]))
cat("NPV:", NPVt, "\n")
```

```
## NPV: 0.6923077
```

Question 8

```r
X <- data.frame(AGE=27, NUM_CREDITS=1, DURATION=12, PRESENT_RESIDENT=1, EMPLOYMENT=1, NUM_DEPENDENTS=1,
                INSTALL_RATE=3,
                GUARANTOR=0, OTHER_INSTALL=0,
                OWN_RES=0, TELEPHONE=1,
                FOREIGN=0,
                REAL_ESTATE=0, AMOUNT_REQUESTED=4500,
                CHK_ACCT_1=1,CHK_ACCT_0=0, CHK_ACCT_2=0, CHK_ACCT_3=0,
                SAV_ACCT_0=0,SAV_ACCT_1=0,SAV_ACCT_2=0,SAV_ACCT_3=0,SAV_ACCT_4=1,
                HISTORY_0=0, HISTORY_1=1, HISTORY_2=0,HISTORY_3=0,HISTORY_4=0,
                JOB_0=0, JOB_1=0,JOB_2=1,JOB_3=0,
                TYPE_0=0,TYPE_1=0,TYPE_2=1,TYPE_3=0,TYPE_4=0,TYPE_5=0,TYPE_6=0)
XX <- as.matrix(X)
XX <- (XX-M)/SD
(prediction4 <- knn(train_input, XX, train_output, k=z, prob=T))
```

```
## [1] 0
## attr(,"prob")
## [1] 0.6
```

```
## Levels: 0 1
```

```r
#
(predicted.probability <- 1-attr(prediction4, "prob"))
```

```
## [1] 0.4
```

```r
predicted.probability
```

```
## [1] 0.4
```

ROC Curve: KNN model

```r
actualKNN <- dftest$PROFITABLE
#
# Predicted probability
prediction3 <- knn(train_input, test_input, train_output, k=z, prob=T)
#
predicted.probability <- attr(prediction3, "prob")
#
# This unfortunately returns the proportion of votes for the winning class - P(Success))
#
predicted.probability.knn <- ifelse(prediction3 ==1, predicted.probability, 1-predicted.probability)
#
par(pty="s")
library(pROC)
```

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

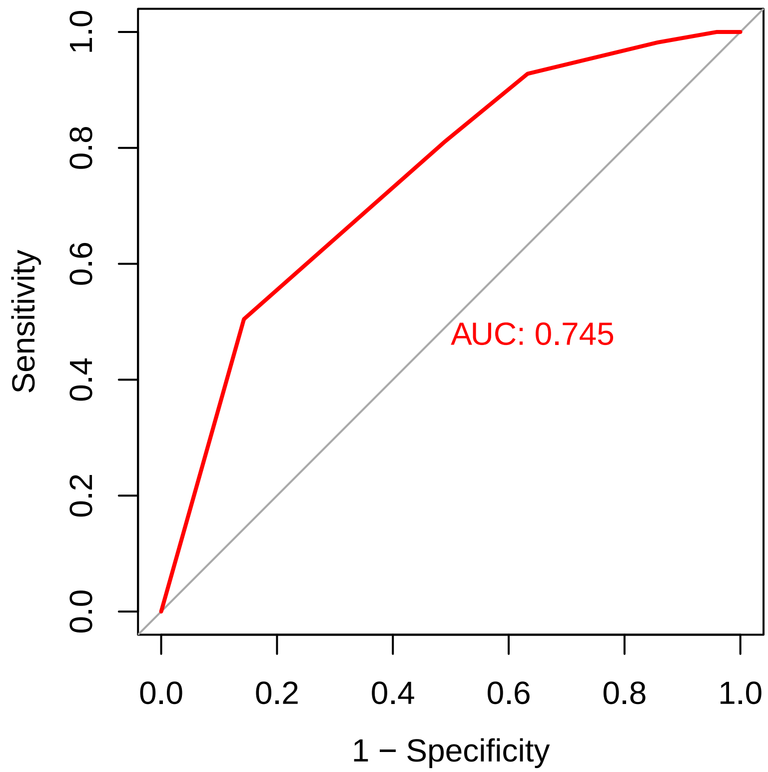
```
##
## Attaching package: 'pROC'
```

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

```r
roc_rose <- plot(roc(actualKNN, predicted.probability.knn), print.auc = TRUE,
                 legacy.axes=T, col = "red")
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```



Naive Bayes Model Data Preparation

```
credit <- read.csv("creditv2.csv")
#
#
credit$AMOUNT_REQUESTED <- as.numeric(credit$AMOUNT_REQUESTED)
credit$PROFITABLE <- ifelse(credit$NPV>0,1,0)
credit$OBS. <- NULL
credit$CREDIT_EXTENDED <- NULL
#

credit$CHK_ACCT <- as.factor(credit$CHK_ACCT)
credit$SAV_ACCT <- as.factor(credit$SAV_ACCT)
credit$NUM_CREDITS <- as.factor(credit$NUM_CREDITS)
credit$HISTORY <- as.factor(credit$HISTORY)
credit$PRESENT_RESIDENT <- as.factor(credit$PRESENT_RESIDENT)
credit$EMPLOYMENT <- as.factor(credit$EMPLOYMENT)
credit$JOB <- as.factor(credit$JOB)
credit$NUM_DEPENDENTS <- as.factor(credit$NUM_DEPENDENTS)
credit$RENT <- as.factor(credit$RENT)
credit$INSTALL_RATE <- as.factor(credit$INSTALL_RATE)
credit$GUARANTOR <- as.factor(credit$GUARANTOR)
credit$OTHER_INSTALL <- as.factor(credit$OTHER_INSTALL)
credit$OWN_RES <- as.factor(credit$OWN_RES)
credit$TELEPHONE <- as.factor(credit$TELEPHONE)
credit$FOREIGN <- as.factor(credit$FOREIGN)
credit$REAL_ESTATE <- as.factor(credit$REAL_ESTATE)
credit$TYPE <- as.factor(credit$TYPE)
#
credit$NPV <- NULL
```

```r
credit$PROFITABLE <- as.factor(credit$PROFITABLE)
```

Splitting the data

```r
set.seed(123)
inTrain <- sample(nrow(credit), 0.6*nrow(credit))
#
train <- data.frame(credit[inTrain,])
temp <- data.frame(credit[-inTrain,])
inVal <- sample(nrow(temp), 0.6*nrow(temp))
validation <- temp[inVal,]
test <- temp[-inVal,]
temp <- NULL
```

Running the Naive Bayes Model

```r
library(e1071)
model <- naiveBayes(PROFITABLE~., data=train)
model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##       0       1
## 0.26875 0.73125
##
## Conditional probabilities:
##    AGE
## Y       [,1]     [,2]
##   0 34.41085 11.72978
##   1 36.55556 11.69452
##
##    CHK_ACCT
## Y            0          1          2          3
##   0 0.44186047 0.35658915 0.04651163 0.15503876
##   1 0.21082621 0.23361823 0.07407407 0.48148148
##
##    SAV_ACCT
## Y            0          1          2          3          4
##   0 0.69767442 0.13953488 0.03875969 0.01550388 0.10852713
##   1 0.53276353 0.09686610 0.06837607 0.06552707 0.23646724
##
##    NUM_CREDITS
## Y             1           2           3           4
##   0 0.720930233 0.240310078 0.023255814 0.015503876
##   1 0.629629630 0.327635328 0.037037037 0.005698006
##
##    DURATION
## Y       [,1]     [,2]
##   0 24.58915 13.18238
```

```
##   1 18.89174 10.68054
##
##    HISTORY
## Y           0          1          2          3          4
##   0 0.03875969 0.12403101 0.62790698 0.06976744 0.13953488
##   1 0.01709402 0.02849003 0.53276353 0.07122507 0.35042735
##
##    PRESENT_RESIDENT
## Y          1         2         3         4
##   0 0.1162791 0.3488372 0.1627907 0.3720930
##   1 0.1481481 0.2962963 0.1424501 0.4131054
##
##    EMPLOYMENT
## Y           0          1          2          3          4
##   0 0.06976744 0.24806202 0.32558140 0.14728682 0.20930233
##   1 0.05982906 0.16809117 0.33048433 0.18803419 0.25356125
##
##    JOB
## Y           0          1          2          3
##   0 0.02325581 0.16279070 0.62790698 0.18604651
##   1 0.02849003 0.20227920 0.61253561 0.15669516
##
##    NUM_DEPENDENTS
## Y          1         2
##   0 0.8294574 0.1705426
##   1 0.8632479 0.1367521
##
##    RENT
## Y          0         1
##   0 0.7596899 0.2403101
##   1 0.8233618 0.1766382
##
##    INSTALL_RATE
## Y           1          2          3          4
##   0 0.08527132 0.20930233 0.13178295 0.57364341
##   1 0.14529915 0.26210826 0.16239316 0.43019943
##
##    GUARANTOR
## Y           0          1
##   0 0.95348837 0.04651163
##   1 0.94017094 0.05982906
##
##    OTHER_INSTALL
## Y          0         1
##   0 0.7984496 0.2015504
##   1 0.8290598 0.1709402
##
##    OWN_RES
## Y          0         1
##   0 0.4031008 0.5968992
##   1 0.2849003 0.7150997
##
##    TELEPHONE
## Y          0         1
```

```
##   0 0.6046512 0.3953488
##   1 0.5584046 0.4415954
##
##    FOREIGN
## Y           0          1
##   0 1.00000000 0.00000000
##   1 0.96866097 0.03133903
##
##    REAL_ESTATE
## Y          0         1
##   0 0.8062016 0.1937984
##   1 0.7037037 0.2962963
##
##    TYPE
## Y           0          1          2          3          4          5
##   0 0.05426357 0.32558140 0.04651163 0.15503876 0.25581395 0.06201550
##   1 0.05698006 0.18233618 0.11396011 0.19088319 0.33903134 0.03703704
##    TYPE
## Y           6
##   0 0.10077519
##   1 0.07977208
##
##    AMOUNT_REQUESTED
## Y       [,1]     [,2]
##   0 3614.264 3232.338
##   1 2910.028 2252.043
```

Getting the confusion matrix and sensitivity for the Naive Bayes model

```r
pred <- predict(model, newdata=test)
(CM <- table(test$PROFITABLE,pred))
```

```
##    pred
##      0  1
##   0 19 21
##   1  9 79
```

```r
(Sen <- (CM[2,2])/sum(CM[2,]))
```

```
## [1] 0.8977273
```

Question 14

```r
X <- data.frame(CHK_ACCT="1",SAV_ACCT="4", NUM_CREDITS="1", HISTORY="1", PRESENT_RESIDENT="1", EMPLOYME
#
X$SAV_ACCT <- as.factor(X$SAV_ACCT)
X$NUM_CREDITS <- as.factor(X$NUM_CREDITS)
X$HISTORY <- as.factor(X$HISTORY)
X$PRESENT_RESIDENT <- as.factor(X$PRESENT_RESIDENT)
X$EMPLOYMENT <- as.factor(X$EMPLOYMENT)
X$JOB <- as.factor(X$JOB)
X$NUM_DEPENDENTS <- as.factor(X$NUM_DEPENDENTS)
X$RENT <- as.factor(X$RENT)
X$INSTALL_RATE <- as.factor(X$INSTALL_RATE)
X$GUARANTOR <- as.factor(X$GUARANTOR)
X$OTHER_INSTALL <- as.factor(X$OTHER_INSTALL)
X$OWN_RES <- as.factor(X$OWN_RES)
```

```r
X$TELEPHONE <- as.factor(X$TELEPHONE)
X$FOREIGN <- as.factor(X$FOREIGN)
X$REAL_ESTATE <- as.factor(X$REAL_ESTATE)
X$TYPE <- as.factor(X$TYPE)
#
# credit$PROFITABLE <- as.factor(credit$PROFITABLE)
predicted.probability2 <- predict(model, newdata = X[1,], type="raw")
predicted.probability2
```

```
##              0         1
## [1,] 0.3747399 0.6252601
```

```r
(predclass <- predict(model, newdata = X[1,]))
```

```
## [1] 1
## Levels: 0 1
```

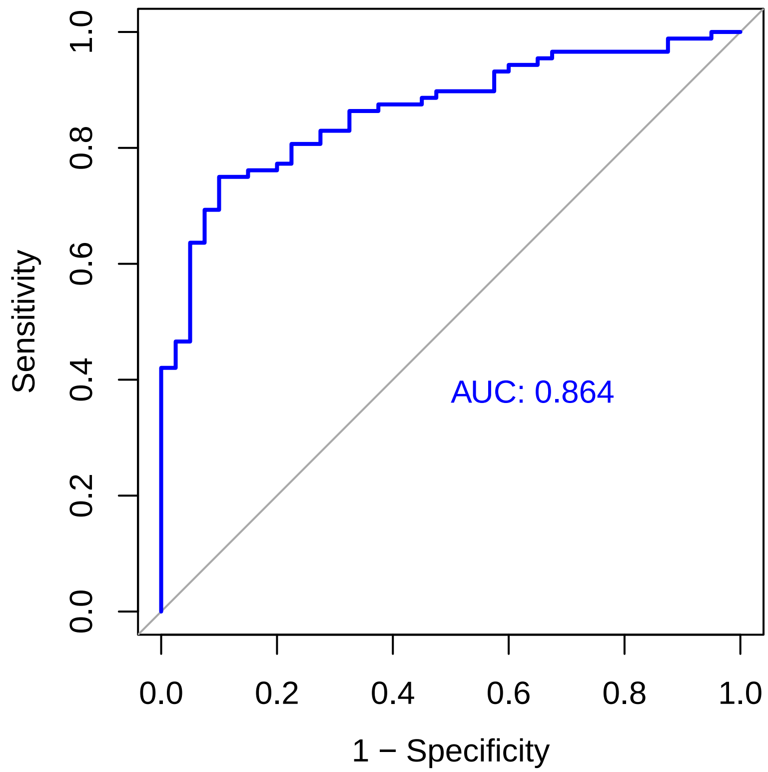
```r
#
#
actualNB <- test$PROFITABLE
pred <- predict(model, newdata = test, type="raw")
predicted.probability.NB <- pred[,2]
```

ROC Curve for Naive Bayes Model

```r
library(pROC)
par(pty="s")
roc_rose <- plot(roc(actualNB, predicted.probability.NB), print.auc = TRUE,
                 col = "blue", print.auc.y = .4, legacy.axes=T)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```



Logistic Regression

```
credit <- read.csv("creditv2.csv")
df_lr<- credit
df_lr$PROFITABLE <- ifelse(df_lr$NPV > 0, 1, 0)

df_lr$NPV <- NULL
df_lr$CREDIT_EXTENDED <- NULL

#converting categorical variables to factors
df_lr$CHK_ACCT <- as.factor(df_lr$CHK_ACCT)
df_lr$SAV_ACCT <- as.factor(df_lr$SAV_ACCT)
df_lr$HISTORY <- as.factor(df_lr$HISTORY)
df_lr$JOB <- as.factor(df_lr$JOB)
df_lr$TYPE <- as.factor(df_lr$TYPE)

#relevel to make sure profitable is the success class
df_lr$PROFITABLE <- factor(df_lr$PROFITABLE,levels=c("0","1"))

# Split data into training, validation, and test sets
set.seed(123)
inTrain <- sample(nrow(df_lr), 0.6*nrow(df_lr))
df_lr_train <- data.frame(df_lr[inTrain,])
df_lr_temp <- data.frame(df_lr[-inTrain,])
inVal <- sample(nrow(df_lr_temp),0.5*nrow(df_lr_temp))
df_lr_validation <- data.frame(df_lr_temp[inVal,])
df_lr_test <- data.frame(df_lr_temp[-inVal,])
df_lr_temp <- NULL

model_lr <- glm(PROFITABLE ~ AGE+CHK_ACCT+SAV_ACCT+NUM_CREDITS+DURATION+
```

```r
                 HISTORY+PRESENT_RESIDENT+EMPLOYMENT+JOB+NUM_DEPENDENTS+
                 RENT+INSTALL_RATE+GUARANTOR+OTHER_INSTALL+OWN_RES+TELEPHONE+
                 FOREIGN+REAL_ESTATE+TYPE+AMOUNT_REQUESTED, data = df_lr_train, family = "binomial")

#confusion matrix for test data
cutoff <- 0.5
actualLR <- df_lr_test$PROFITABLE
predicted.probability.LR <- predict(model_lr, type = "response", newdata = df_lr_test)
PredictedTest <- ifelse( predicted.probability.LR > cutoff, "1", "0")
PredictedTest <- factor(PredictedTest,levels=c("0","1"))
(confusionTest <- table(actualLR, PredictedTest))
```

```
##          PredictedTest
## actualLR   0   1
##         0  23  26
##         1   6 105
```

```r
error_rate<- (26+6)/(26+6+23+105)
error_rate
```

```
## [1] 0.2
```

ROC curves for all 3 models

```r
par(pty="s")
library(pROC)
roc_rose <- plot(roc(actualKNN, predicted.probability.knn), print.auc = TRUE, print.auc.y = .3,
                legacy.axes=T, col = "red")
```

```
## Setting levels: control = 0, case = 1
```

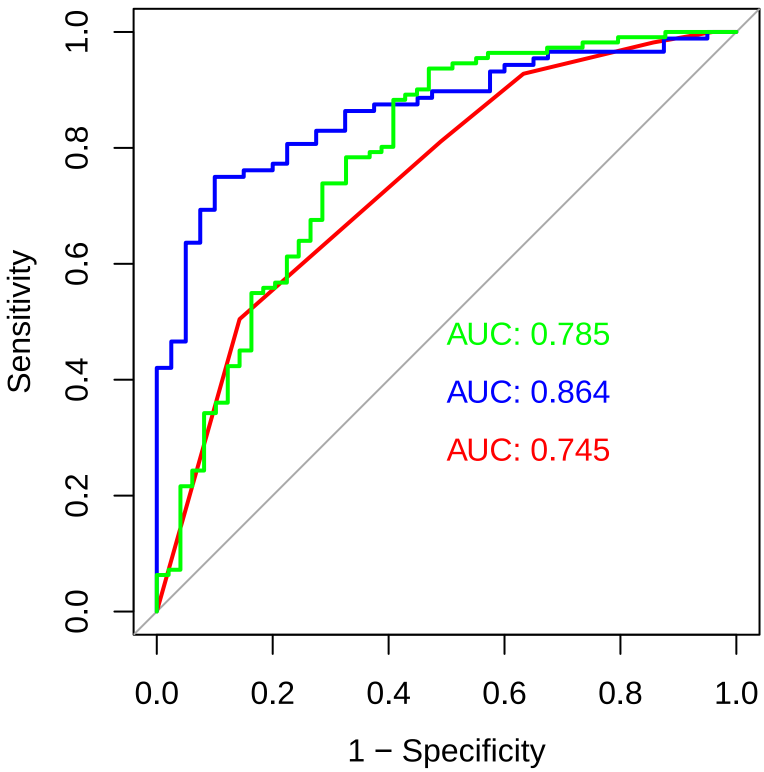
```
## Setting direction: controls < cases
```

```r
roc_rose <- plot(roc(actualNB, predicted.probability.NB), print.auc = TRUE,
                legacy.axes=T, col = "blue", add=TRUE, print.auc.y = .4)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
roc_rose <- plot(roc(actualLR, predicted.probability.LR), print.auc = TRUE, print.auc.y = .5,
                legacy.axes=T, col = "green", add=TRUE)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```



AUC: 0.785
AUC: 0.864
AUC: 0.745
Sensitivity
1 − Specificity
0.0
0.2
0.4
0.6
0.8
1.0

"'