Assignment 6

Pranav Vinod

3/21/2022

library(DAAG)

## Loading required package: lattice

library(datasets)  
library(ggplot2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(psych)

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

library(caret)  
library(pROC)

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

##   
## Attaching package: 'pROC'

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

library(nnet)  
# Read the data  
data <- read.csv("/Users/pranavvinod/downloads/A6DATA.txt", header = F)  
data$V5 <- as.factor(data$V5)  
  
summary(data) # 762 forged banknotes and 610 genuine banknotes

## V1 V2 V3 V4   
## Min. :-7.0421 Min. :-13.773 Min. :-5.2861 Min. :-8.5482   
## 1st Qu.:-1.7730 1st Qu.: -1.708 1st Qu.:-1.5750 1st Qu.:-2.4135   
## Median : 0.4962 Median : 2.320 Median : 0.6166 Median :-0.5867   
## Mean : 0.4337 Mean : 1.922 Mean : 1.3976 Mean :-1.1917   
## 3rd Qu.: 2.8215 3rd Qu.: 6.815 3rd Qu.: 3.1793 3rd Qu.: 0.3948   
## Max. : 6.8248 Max. : 12.952 Max. :17.9274 Max. : 2.4495   
## V5   
## 0:762   
## 1:610   
##   
##   
##   
##

set.seed(222)  
  
# Partitioning the data  
indices <- sample(1:nrow(data), size = nrow(data)\*0.5)  
train <- data[indices,]  
test <- data[-indices,]  
  
# Data description  
summary(train) # 381 forged banknotes and 305 genuine banknotes

## V1 V2 V3 V4   
## Min. :-7.0364 Min. :-13.773 Min. :-5.2133 Min. :-7.7853   
## 1st Qu.:-1.8180 1st Qu.: -2.502 1st Qu.:-1.2489 1st Qu.:-2.0381   
## Median : 0.3345 Median : 1.883 Median : 0.7679 Median :-0.5168   
## Mean : 0.3549 Mean : 1.673 Mean : 1.5917 Mean :-1.0993   
## 3rd Qu.: 2.6627 3rd Qu.: 6.623 3rd Qu.: 3.3064 3rd Qu.: 0.4385   
## Max. : 6.5633 Max. : 12.952 Max. :17.9274 Max. : 2.4495   
## V5   
## 0:381   
## 1:305   
##   
##   
##   
##

summary(test) # 381 forged banknotes and 305 genuine banknotes

## V1 V2 V3 V4   
## Min. :-7.0421 Min. :-13.678 Min. :-5.2861 Min. :-8.5482   
## 1st Qu.:-1.7448 1st Qu.: -1.017 1st Qu.:-1.8191 1st Qu.:-2.6168   
## Median : 0.5404 Median : 2.872 Median : 0.3065 Median :-0.7160   
## Mean : 0.5126 Mean : 2.172 Mean : 1.2035 Mean :-1.2840   
## 3rd Qu.: 2.9713 3rd Qu.: 6.937 3rd Qu.: 3.1330 3rd Qu.: 0.3408   
## Max. : 6.8248 Max. : 12.378 Max. :17.6772 Max. : 2.1625   
## V5   
## 0:381   
## 1:305   
##   
##   
##   
##

# Create model for 50:50 split  
m1 <- glm(formula = V5 ~., data = train, family = 'binomial')

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

summary(m1)

##   
## Call:  
## glm(formula = V5 ~ ., family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.21019 0.00000 0.00000 0.00008 2.34895   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 9.5831 3.7680 2.543 0.01098 \*   
## V1 -9.2371 3.3643 -2.746 0.00604 \*\*  
## V2 -3.7759 1.2996 -2.905 0.00367 \*\*  
## V3 -5.2246 1.8005 -2.902 0.00371 \*\*  
## V4 0.9020 0.8068 1.118 0.26358   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 942.561 on 685 degrees of freedom  
## Residual deviance: 21.758 on 681 degrees of freedom  
## AIC: 31.758  
##   
## Number of Fisher Scoring iterations: 13

# we can see that V4 is not a significant variable for predictions, so we drop it  
m1.best <- glm(formula = V5 ~.-V4, data = train, family = 'binomial')

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

summary(m1.best)

##   
## Call:  
## glm(formula = V5 ~ . - V4, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.34409 0.00000 0.00000 0.00013 2.03675   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 8.094 2.613 3.098 0.001951 \*\*   
## V1 -8.616 2.688 -3.205 0.001349 \*\*   
## V2 -3.948 1.174 -3.364 0.000770 \*\*\*  
## V3 -5.217 1.585 -3.291 0.000999 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 942.561 on 685 degrees of freedom  
## Residual deviance: 23.373 on 682 degrees of freedom  
## AIC: 31.373  
##   
## Number of Fisher Scoring iterations: 12

# Final equation : log(p) = 8.094 - 8.616\*V1 - 3.948\*V2 - 5.217\*V3  
  
# Confusion matrices   
p1 <- predict(m1.best, train, type = 'response')  
pred1 <- ifelse(p1>0.5,1,0)  
cm1 <- table(pred1, train$V5) # for training data  
cm1

##   
## pred1 0 1  
## 0 377 4  
## 1 4 301

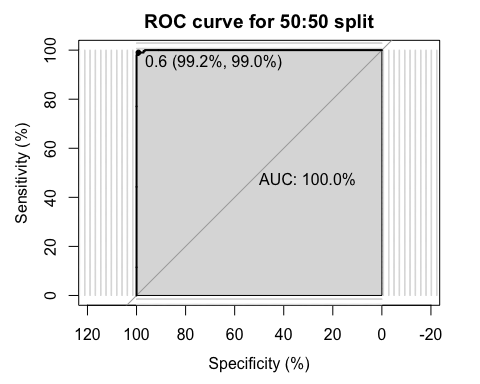
p1 <- predict(m1.best, test, type = 'response')  
pred1 <- ifelse(p1>0.5,1,0)  
cm2 <- table(pred1, test$V5) # for testing data  
cm2

##   
## pred1 0 1  
## 0 377 3  
## 1 4 302

# Misclassification Error/Accuracy  
errorTrain <- (377+301)/(377+301+8) # 98.8% / 1.2%  
errorTest <- (377+302)/(377+301+7) # 99.1% / 1.9%  
  
# Specificity, sensitivity and accuracy  
Spec1 <- cm1[1]/(cm1[1]+cm1[3]) # 0.989  
sens1 <- cm1[4]/(cm1[2]+cm1[4]) #0.986  
   
Spec2 <- cm2[1]/(cm2[1]+cm2[3]) # 0.992  
sens2 <- cm2[4]/(cm2[2]+cm2[4]) # 0.986  
  
# ROC Curve  
  
r <- multiclass.roc(test$V5, p1, percent = TRUE)

## Setting direction: controls < cases

roc <- r[['rocs']]  
r1 <- roc[[1]]  
plot.roc(r1,print.auc = T,  
 print.thres = T,  
 auc.polygon = T,  
 grid = c(0.1,0.2),  
 main = "ROC curve for 50:50 split")



r1

##   
## Call:  
## roc.default(response = response, predictor = predictor, levels = X, percent = percent, direction = ..1, auc = FALSE, ci = FALSE)  
##   
## Data: predictor in 381 controls (response 0) < 305 cases (response 1).  
## Area under the curve not computed.

##########################################################################################################  
  
# 60:40 split  
  
# Partitioning the data  
indices <- sample(1:nrow(data), size = nrow(data)\*0.6)  
# indices <- sample(2, nrow(data), replace = F, prob = c(0.8, 0.2))  
train <- data[indices,]  
test <- data[-indices,]  
  
# Data description  
summary(train) # 463 forged banknotes and 360 genuine banknotes

## V1 V2 V3 V4   
## Min. :-7.0421 Min. :-13.678 Min. :-5.2861 Min. :-8.5482   
## 1st Qu.:-1.8389 1st Qu.: -1.053 1st Qu.:-1.5067 1st Qu.:-2.3660   
## Median : 0.5415 Median : 2.665 Median : 0.5721 Median :-0.5776   
## Mean : 0.4453 Mean : 2.069 Mean : 1.3519 Mean :-1.1636   
## 3rd Qu.: 2.8542 3rd Qu.: 6.721 3rd Qu.: 3.2086 3rd Qu.: 0.4194   
## Max. : 6.8248 Max. : 12.730 Max. :17.5795 Max. : 2.4495   
## V5   
## 0:463   
## 1:360   
##   
##   
##   
##

summary(test) # 299 forged banknotes and 250 genuine banknotes

## V1 V2 V3 V4   
## Min. :-6.7526 Min. :-13.773 Min. :-5.2159 Min. :-7.6612   
## 1st Qu.:-1.6706 1st Qu.: -2.212 1st Qu.:-1.6643 1st Qu.:-2.4927   
## Median : 0.3798 Median : 1.846 Median : 0.6312 Median :-0.6125   
## Mean : 0.4164 Mean : 1.702 Mean : 1.4661 Mean :-1.2338   
## 3rd Qu.: 2.6946 3rd Qu.: 6.870 3rd Qu.: 3.0895 3rd Qu.: 0.3521   
## Max. : 6.5633 Max. : 12.952 Max. :17.9274 Max. : 2.1625   
## V5   
## 0:299   
## 1:250   
##   
##   
##   
##

# Create model for 60:40 split  
m2 <- glm(formula = V5 ~., data = train, family = 'binomial')

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

summary(m2)

##   
## Call:  
## glm(formula = V5 ~ ., family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.76362 0.00000 0.00000 0.00008 2.12447   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 8.7644 3.1305 2.800 0.00511 \*\*  
## V1 -9.0661 3.3630 -2.696 0.00702 \*\*  
## V2 -4.8932 1.7711 -2.763 0.00573 \*\*  
## V3 -6.2577 2.3136 -2.705 0.00684 \*\*  
## V4 -0.6383 0.5575 -1.145 0.25223   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1127.996 on 822 degrees of freedom  
## Residual deviance: 16.985 on 818 degrees of freedom  
## AIC: 26.985  
##   
## Number of Fisher Scoring iterations: 13

# we can see that V4 is not a significant variable for predictions, so we drop it  
m2.best <- glm(formula = V5 ~.-V4, data = train, family = 'binomial')

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

summary(m2.best)

##   
## Call:  
## glm(formula = V5 ~ . - V4, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.65242 0.00000 0.00000 0.00029 2.45575   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 8.021 2.458 3.263 0.001101 \*\*   
## V1 -7.770 2.471 -3.144 0.001664 \*\*   
## V2 -4.055 1.230 -3.297 0.000976 \*\*\*  
## V3 -5.226 1.619 -3.227 0.001250 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1128.00 on 822 degrees of freedom  
## Residual deviance: 18.37 on 819 degrees of freedom  
## AIC: 26.37  
##   
## Number of Fisher Scoring iterations: 13

# Final equation : p = 8.021 - 7.770\*V1 - 4.055\*V2 - 5.226\*V3  
  
# Confusion matrices   
p2 <- predict(m2.best, train, type = 'response')  
pred2 <- ifelse(p2>0.5,1,0)  
cm1 <- table(pred2, train$V5) # for training data  
cm1

##   
## pred2 0 1  
## 0 461 1  
## 1 2 359

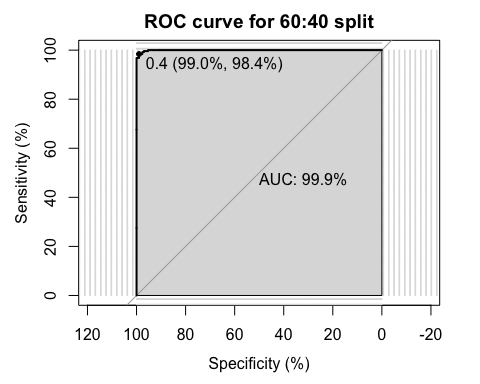
p2 <- predict(m2.best, test, type = 'response')  
pred2 <- ifelse(p2>0.5,1,0)  
cm2 <- table(pred2, test$V5) # for testing data  
cm2

##   
## pred2 0 1  
## 0 296 6  
## 1 3 244

# Misclassification Error/Accuracy  
errorTrain <- (461+359)/(461+359+3) # 99.6%  
errorTest <- (296+244)/(294+244+9) # 98.7%  
  
# Specificity, sensitivity and accuracy  
Spec1 <- cm1[1]/(cm1[1]+cm1[3]) # 0.997  
sens1 <- cm1[4]/(cm1[2]+cm1[4]) #0.994  
  
Spec2 <- cm2[1]/(cm2[1]+cm2[3]) # 0.98  
sens2 <- cm2[4]/(cm2[2]+cm2[4]) # 0.987  
  
r <- multiclass.roc(test$V5, p2, percent = TRUE)

## Setting direction: controls < cases

roc <- r[['rocs']]  
r2 <- roc[[1]]  
plot.roc(r2,print.auc = T,  
 print.thres = T,  
 auc.polygon = T,  
 grid = c(0.1,0.2),  
 main = "ROC curve for 60:40 split")



r2

##   
## Call:  
## roc.default(response = response, predictor = predictor, levels = X, percent = percent, direction = ..1, auc = FALSE, ci = FALSE)  
##   
## Data: predictor in 299 controls (response 0) < 250 cases (response 1).  
## Area under the curve not computed.

#######################################################################################################  
  
# 70:30 split  
  
# Partitioning the data  
indices <- sample(1:nrow(data), size = nrow(data)\*0.7)  
train <- data[indices,]  
test <- data[-indices,]  
  
# Data description  
summary(train) # 552 forged banknotes and 408 genuine banknotes

## V1 V2 V3 V4   
## Min. :-7.0364 Min. :-13.678 Min. :-5.2861 Min. :-8.5482   
## 1st Qu.:-1.7073 1st Qu.: -1.415 1st Qu.:-1.7574 1st Qu.:-2.6204   
## Median : 0.5446 Median : 2.693 Median : 0.5297 Median :-0.6649   
## Mean : 0.4940 Mean : 2.253 Mean : 1.2287 Mean :-1.2781   
## 3rd Qu.: 2.9550 3rd Qu.: 7.203 3rd Qu.: 3.0526 3rd Qu.: 0.3777   
## Max. : 6.8248 Max. : 12.952 Max. :17.5795 Max. : 2.4495   
## V5   
## 0:552   
## 1:408   
##   
##   
##   
##

summary(test) # 210 forged banknotes and 202 genuine banknotes

## V1 V2 V3 V4 V5   
## Min. :-7.0421 Min. :-13.773 Min. :-5.151 Min. :-7.8719 0:210   
## 1st Qu.:-2.0463 1st Qu.: -2.631 1st Qu.:-1.293 1st Qu.:-1.9502 1:202   
## Median : 0.3088 Median : 1.302 Median : 0.779 Median :-0.4709   
## Mean : 0.2933 Mean : 1.152 Mean : 1.791 Mean :-0.9902   
## 3rd Qu.: 2.6275 3rd Qu.: 5.667 3rd Qu.: 3.493 3rd Qu.: 0.4339   
## Max. : 5.9374 Max. : 12.606 Max. :17.927 Max. : 2.1353

# Create model for 70:30 split  
m3 <- glm(formula = V5 ~., data = train, family = 'binomial')

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

summary(m3)

##   
## Call:  
## glm(formula = V5 ~ ., family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.786 0.000 0.000 0.000 1.561   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 15.6440 6.5219 2.399 0.0165 \*  
## V1 -16.7027 7.1441 -2.338 0.0194 \*  
## V2 -8.6264 3.7213 -2.318 0.0204 \*  
## V3 -11.2348 4.8598 -2.312 0.0208 \*  
## V4 -0.9746 0.8665 -1.125 0.2607   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1309.161 on 959 degrees of freedom  
## Residual deviance: 14.982 on 955 degrees of freedom  
## AIC: 24.982  
##   
## Number of Fisher Scoring iterations: 14

# we can see that V4 is not a significant variable for predictions, so we drop it  
m3.best <- glm(formula = V5 ~.-V4, data = train, family = 'binomial')

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

summary(m3.best)

##   
## Call:  
## glm(formula = V5 ~ . - V4, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.664 0.000 0.000 0.000 1.894   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 13.152 4.732 2.779 0.00545 \*\*  
## V1 -13.198 4.675 -2.823 0.00476 \*\*  
## V2 -6.698 2.382 -2.813 0.00491 \*\*  
## V3 -8.758 3.115 -2.812 0.00493 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1309.161 on 959 degrees of freedom  
## Residual deviance: 16.401 on 956 degrees of freedom  
## AIC: 24.401  
##   
## Number of Fisher Scoring iterations: 14

# Final equation : p = 13.152 - 13.198\*V1 - 6.698\*V2 - 8.758\*V3  
  
# Confusion matrices   
p3 <- predict(m3.best, train, type = 'response')  
pred3 <- ifelse(p3>0.5,1,0)  
cm1 <- table(pred3, train$V5) # for training data  
cm1

##   
## pred3 0 1  
## 0 550 2  
## 1 2 406

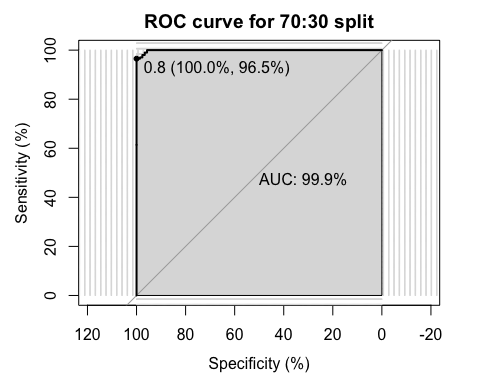
p3 <- predict(m3.best, test, type = 'response')  
pred3 <- ifelse(p3>0.5,1,0)  
cm2 <- table(pred3, test$V5) # for testing data  
cm2

##   
## pred3 0 1  
## 0 207 6  
## 1 3 196

# Misclassification Error/Accuracy  
errorTrain <- (550+406)/(550+406+4) # 99.5%  
errorTest <- (207+196)/(207+196+9) # 97.8%  
  
# Specificity, sensitivity and accuracy  
Spec1 <- cm1[1]/(cm1[1]+cm1[3]) # 0.996  
sens1 <- cm1[4]/(cm1[2]+cm1[4]) #0.995  
  
Spec2 <- cm2[1]/(cm2[1]+cm2[3]) # 0.971  
sens2 <- cm2[4]/(cm2[2]+cm2[4]) # 0.984  
  
r <- multiclass.roc(test$V5, p3, percent = TRUE)

## Setting direction: controls < cases

roc <- r[['rocs']]  
r3 <- roc[[1]]  
plot.roc(r3,print.auc = T,  
 print.thres = T,  
 auc.polygon = T,  
 grid = c(0.1,0.2),  
 main = "ROC curve for 70:30 split")



r3

##   
## Call:  
## roc.default(response = response, predictor = predictor, levels = X, percent = percent, direction = ..1, auc = FALSE, ci = FALSE)  
##   
## Data: predictor in 210 controls (response 0) < 202 cases (response 1).  
## Area under the curve not computed.

##############################################################################################  
  
# 80:20  
  
# Partitioning the data  
indices <- sample(1:nrow(data), size = nrow(data)\*0.8)  
train <- data[indices,]  
test <- data[-indices,]  
  
# Data description  
summary(train) # 606 forged banknotes and 491 genuine banknotes

## V1 V2 V3 V4   
## Min. :-6.9599 Min. :-13.773 Min. :-5.2613 Min. :-8.5482   
## 1st Qu.:-1.7599 1st Qu.: -1.767 1st Qu.:-1.5443 1st Qu.:-2.3530   
## Median : 0.4690 Median : 2.345 Median : 0.5755 Median :-0.5828   
## Mean : 0.4396 Mean : 1.900 Mean : 1.3666 Mean :-1.1880   
## 3rd Qu.: 2.8297 3rd Qu.: 6.795 3rd Qu.: 3.1392 3rd Qu.: 0.3612   
## Max. : 6.8248 Max. : 12.952 Max. :17.9274 Max. : 2.4495   
## V5   
## 0:606   
## 1:491   
##   
##   
##   
##

summary(test) # 156 forged banknotes and 119 genuine banknotes

## V1 V2 V3 V4 V5   
## Min. :-7.0421 Min. :-13.459 Min. :-5.286 Min. :-7.7581 0:156   
## 1st Qu.:-1.8116 1st Qu.: -1.473 1st Qu.:-1.681 1st Qu.:-2.5884 1:119   
## Median : 0.5394 Median : 2.237 Median : 0.686 Median :-0.6164   
## Mean : 0.4102 Mean : 2.012 Mean : 1.521 Mean :-1.2063   
## 3rd Qu.: 2.7184 3rd Qu.: 6.894 3rd Qu.: 3.398 3rd Qu.: 0.4375   
## Max. : 6.0919 Max. : 12.118 Max. :17.593 Max. : 2.1547

# Create model for 80:20 split  
m4 <- glm(formula = V5 ~., data = train, family = 'binomial')

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

summary(m4)

##   
## Call:  
## glm(formula = V5 ~ ., family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.62609 0.00000 0.00000 0.00011 2.24154   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 7.9667 1.9259 4.137 3.52e-05 \*\*\*  
## V1 -8.8730 2.2385 -3.964 7.38e-05 \*\*\*  
## V2 -4.6885 1.1662 -4.020 5.81e-05 \*\*\*  
## V3 -5.9333 1.4916 -3.978 6.96e-05 \*\*\*  
## V4 -0.6959 0.4094 -1.700 0.0892 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1508.687 on 1096 degrees of freedom  
## Residual deviance: 35.159 on 1092 degrees of freedom  
## AIC: 45.159  
##   
## Number of Fisher Scoring iterations: 13

# we can see that V4 is not a significant variable for predictions, so we drop it  
m4.best <- glm(formula = V5 ~.-V4, data = train, family = 'binomial')

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

summary(m4.best)

##   
## Call:  
## glm(formula = V5 ~ . - V4, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.57011 0.00000 0.00000 0.00048 2.18446   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 7.4386 1.7124 4.344 1.40e-05 \*\*\*  
## V1 -7.5205 1.7492 -4.300 1.71e-05 \*\*\*  
## V2 -3.8352 0.8651 -4.433 9.28e-06 \*\*\*  
## V3 -4.9067 1.1231 -4.369 1.25e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1508.687 on 1096 degrees of freedom  
## Residual deviance: 38.244 on 1093 degrees of freedom  
## AIC: 46.244  
##   
## Number of Fisher Scoring iterations: 12

# Final equation : p = 7.438 - 7.52\*V1 - 3.835\*V2 - 4.90\*V3  
  
# Confusion matrices   
p4 <- predict(m4.best, train, type = 'response')  
pred4 <- ifelse(p4>0.5,1,0)  
cm1 <- table(pred4, train$V5) # for training data  
cm1

##   
## pred4 0 1  
## 0 602 5  
## 1 4 486

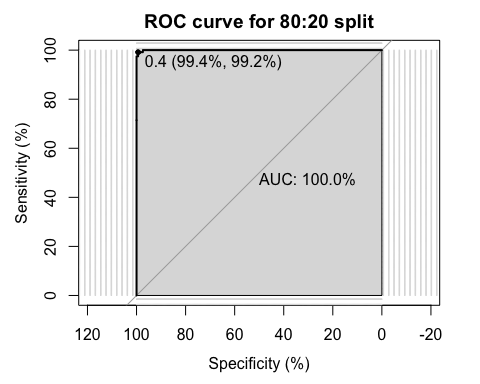
p4 <- predict(m4.best, test, type = 'response')  
pred4 <- ifelse(p4>0.5,1,0)  
cm2 <- table(pred4, test$V5) # for testing data  
cm2

##   
## pred4 0 1  
## 0 155 2  
## 1 1 117

# Misclassification Error/Accuracy  
errorTrain <- (602+486)/(602+486+9) # 99.1%  
errorTest <- (155+117)/(155+117+3) # 98.9%  
  
# Specificity, sensitivity and accuracy  
Spec1 <- cm1[1]/(cm1[1]+cm1[3]) # 0.991  
sens1 <- cm1[4]/(cm1[2]+cm1[4]) #0.991  
  
Spec2 <- cm2[1]/(cm2[1]+cm2[3]) # 0.987  
sens2 <- cm2[4]/(cm2[2]+cm2[4]) # 0.991  
  
r <- multiclass.roc(test$V5, p4, percent = TRUE)

## Setting direction: controls < cases

roc <- r[['rocs']]  
r4 <- roc[[1]]  
plot.roc(r4,print.auc = T,  
 print.thres = T,  
 auc.polygon = T,  
 grid = c(0.1,0.2),  
 main = "ROC curve for 80:20 split")



r4

##   
## Call:  
## roc.default(response = response, predictor = predictor, levels = X, percent = percent, direction = ..1, auc = FALSE, ci = FALSE)  
##   
## Data: predictor in 156 controls (response 0) < 119 cases (response 1).  
## Area under the curve not computed.

#########################################################################################################