Análise Discriminante Múltipla

Felipe N. S. Bezerra

10 de dezembro de 2018

Table of Contents

Avaliação - Estatística Multivariada IV Análise Discriminante Múltipla

Considere o conjunto Auto data do pacote ISLR do software R para desenvolver um modelo de predição para prever se um carro tem alta ou baixa quilometragem.

library(ISLR)  
library(foreign)  
library(MASS)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':  
##   
## select

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

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

library(biotools)

## Loading required package: rpanel

## Loading required package: tcltk

## Package `rpanel', version 1.1-4: type help(rpanel) for summary information

## Loading required package: tkrplot

## Loading required package: lattice

## Loading required package: SpatialEpi

## Loading required package: sp

## ---  
## biotools version 3.1

##

library(DiscriMiner)  
library(ggplot2)  
library(mvnormtest)

summary(Auto)

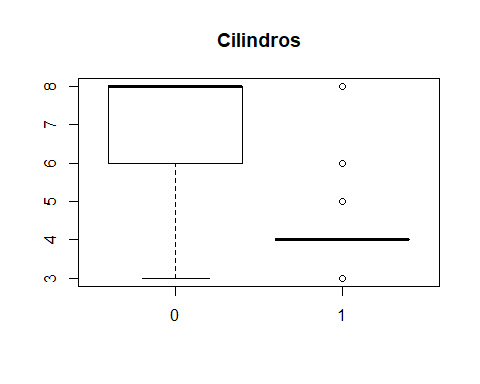
## mpg cylinders displacement horsepower   
## Min. : 9.00 Min. :3.000 Min. : 68.0 Min. : 46.0   
## 1st Qu.:17.00 1st Qu.:4.000 1st Qu.:105.0 1st Qu.: 75.0   
## Median :22.75 Median :4.000 Median :151.0 Median : 93.5   
## Mean :23.45 Mean :5.472 Mean :194.4 Mean :104.5   
## 3rd Qu.:29.00 3rd Qu.:8.000 3rd Qu.:275.8 3rd Qu.:126.0   
## Max. :46.60 Max. :8.000 Max. :455.0 Max. :230.0   
##   
## weight acceleration year origin   
## Min. :1613 Min. : 8.00 Min. :70.00 Min. :1.000   
## 1st Qu.:2225 1st Qu.:13.78 1st Qu.:73.00 1st Qu.:1.000   
## Median :2804 Median :15.50 Median :76.00 Median :1.000   
## Mean :2978 Mean :15.54 Mean :75.98 Mean :1.577   
## 3rd Qu.:3615 3rd Qu.:17.02 3rd Qu.:79.00 3rd Qu.:2.000   
## Max. :5140 Max. :24.80 Max. :82.00 Max. :3.000   
##   
## name   
## amc matador : 5   
## ford pinto : 5   
## toyota corolla : 5   
## amc gremlin : 4   
## amc hornet : 4   
## chevrolet chevette: 4   
## (Other) :365

(a) Crie uma variável binária, classmpg, que seja igual a 1 se o mpg for maior do que a mediana e 0, caso contrário. Você pode calcular a mediana de mpg no R usando a função median(). Note que talvez seja mais fácil usar o comando data.frame() para criar um conjunto de dados que contenha o classmpg e as demais variáveis do Auto data.

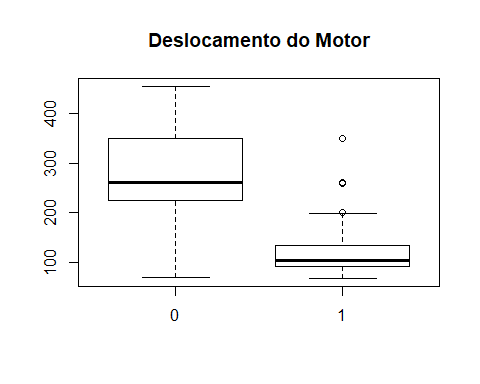
avaldata <- Auto[,-1]  
avaldata <- avaldata[,-8]  
avaldata$origin <- as.factor(avaldata$origin)  
  
scaleavaldata <- avaldata[,-7]  
scaleavaldata <- as.data.frame(scale(scaleavaldata))  
  
classmpg <- as.numeric(Auto$mpg>median(Auto$mpg))  
avaldata$classmpg <- as.factor(classmpg)  
scaleavaldata$origin <- avaldata$origin  
scaleavaldata$classmpg <- avaldata$classmpg

(b) Explore os dados graficamente para investigar a associação entre o classmpg e as demais variáveis. Quais variáveis parecem ser úteis para prever o classmpg? Pode usar boxplots para responder a essa questão.

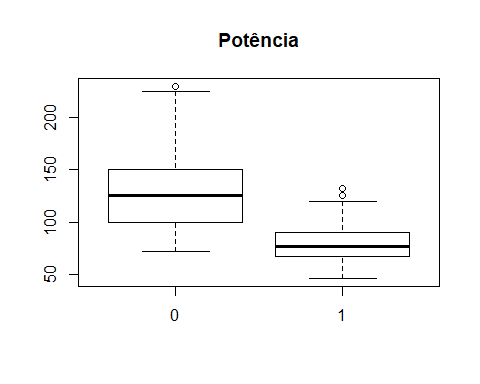
boxplot(avaldata$cylinders ~ avaldata$classmpg, main='Cilindros')



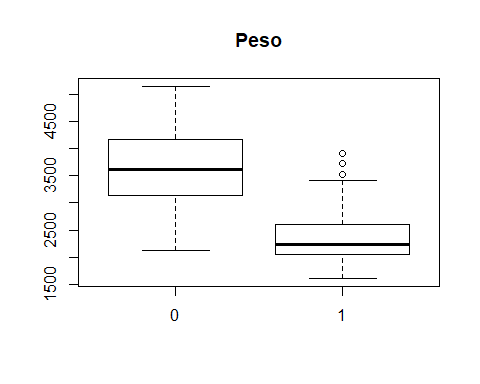
boxplot(avaldata$displacement ~ avaldata$classmpg, main='Deslocamento do Motor')



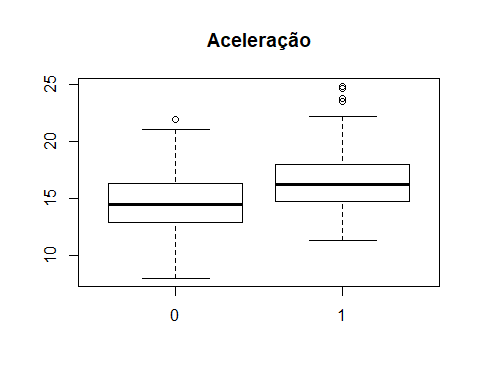
boxplot(avaldata$horsepower ~ avaldata$classmpg, main='Potência')



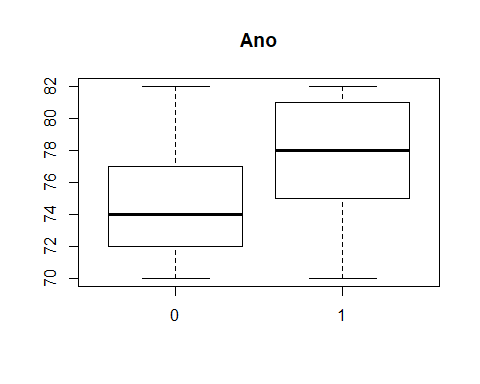
boxplot(avaldata$weight ~ avaldata$classmpg, main='Peso')



boxplot(avaldata$acceleration ~ avaldata$classmpg, main='Aceleração')



boxplot(avaldata$year ~ avaldata$classmpg, main='Ano')



aggregate(avaldata[,1:6], list(avaldata$classmpg), quantile)

## Group.1 cylinders.0% cylinders.25% cylinders.50% cylinders.75%  
## 1 0 3 6 8 8  
## 2 1 3 4 4 4  
## cylinders.100% displacement.0% displacement.25% displacement.50%  
## 1 8 70 225 261  
## 2 8 68 91 105  
## displacement.75% displacement.100% horsepower.0% horsepower.25%  
## 1 350 455 72.0 100.0  
## 2 134 350 46.0 67.0  
## horsepower.50% horsepower.75% horsepower.100% weight.0% weight.25%  
## 1 125.0 150.0 230.0 2124.00 3139.75  
## 2 76.5 90.0 132.0 1613.00 2045.00  
## weight.50% weight.75% weight.100% acceleration.0% acceleration.25%  
## 1 3607.00 4156.75 5140.00 8.000 12.950  
## 2 2229.00 2607.50 3900.00 11.300 14.700  
## acceleration.50% acceleration.75% acceleration.100% year.0% year.25%  
## 1 14.500 16.250 21.900 70 72  
## 2 16.200 17.925 24.800 70 75  
## year.50% year.75% year.100%  
## 1 74 77 82  
## 2 78 81 82

As variáveis “Cilindros” e “Deslocamento do Motor” parecem possuir maior acertividade em definir os veículos com maior consumo de combustível (classmpg = 0). Veículos com um número de cilindros diferente de 4 e Deslocamento do Motor com mais de 200 in percorrem menos milhas por galão; as demais faixas destas variáveis apresentam consumo de combustível variado. Veículos com maior potência, menor aceleração, mais pesados e mais antigos aparentam consumir mais combustível, mas tendência não se mostra tão definida quanto as duas variáveis citadas anteriormente.

(c) Divida os dados em duas amostras, uma de treino (75%) e outra de teste (25%).

set.seed(8)  
treinoaval <- scaleavaldata[sample(nrow(scaleavaldata), size = nrow(scaleavaldata) \* 0.75),]  
testeaval <- scaleavaldata[-sample(nrow(scaleavaldata), size = nrow(scaleavaldata) \* 0.75),]

(d) Obtenha e interprete as funções discriminantes para esse estudo. Verifique também as suposições da análise. Você utilizaria a análise discriminante linear ou quadrática para classificação?

Verificação do tamanho da amostra:

table(scaleavaldata$classmpg)

##   
## 0 1   
## 196 196

table(treinoaval$classmpg)

##   
## 0 1   
## 146 148

table(testeaval$classmpg)

##   
## 0 1   
## 50 48

Todos os grupos possuem mais de 20 observações (número aceitável para 6 variáveis independentes), inclusive nas amostras separadas para treino e teste.

Verificação de multicolinearidade:

cor(scaleavaldata[,1:6])

## cylinders displacement horsepower weight acceleration  
## cylinders 1.0000000 0.9508233 0.8429834 0.8975273 -0.5046834  
## displacement 0.9508233 1.0000000 0.8972570 0.9329944 -0.5438005  
## horsepower 0.8429834 0.8972570 1.0000000 0.8645377 -0.6891955  
## weight 0.8975273 0.9329944 0.8645377 1.0000000 -0.4168392  
## acceleration -0.5046834 -0.5438005 -0.6891955 -0.4168392 1.0000000  
## year -0.3456474 -0.3698552 -0.4163615 -0.3091199 0.2903161  
## year  
## cylinders -0.3456474  
## displacement -0.3698552  
## horsepower -0.4163615  
## weight -0.3091199  
## acceleration 0.2903161  
## year 1.0000000

As variáveis “Cilindros”e “Deslocamento do Motor” estão altamente correlacionadas (correlação superior a 0,95). Utilizar ambas implica em redundância, então apenas uma (Deslocamento do Motor) será utilizada no ajuste do modelo.

Normalidade multivariada das variáveis independentes:

mshapiro.test(t(scaleavaldata[,1:6]))

##   
## Shapiro-Wilk normality test  
##   
## data: Z  
## W = 0.86677, p-value < 2.2e-16

shapiro.test(scaleavaldata$cylinders)

##   
## Shapiro-Wilk normality test  
##   
## data: scaleavaldata$cylinders  
## W = 0.75066, p-value < 2.2e-16

shapiro.test(scaleavaldata$displacement)

##   
## Shapiro-Wilk normality test  
##   
## data: scaleavaldata$displacement  
## W = 0.88184, p-value < 2.2e-16

shapiro.test(scaleavaldata$horsepower)

##   
## Shapiro-Wilk normality test  
##   
## data: scaleavaldata$horsepower  
## W = 0.9041, p-value = 5.022e-15

shapiro.test(scaleavaldata$weight)

##   
## Shapiro-Wilk normality test  
##   
## data: scaleavaldata$weight  
## W = 0.94147, p-value = 2.602e-11

shapiro.test(scaleavaldata$acceleration)

##   
## Shapiro-Wilk normality test  
##   
## data: scaleavaldata$acceleration  
## W = 0.99187, p-value = 0.03053

shapiro.test(scaleavaldata$year)

##   
## Shapiro-Wilk normality test  
##   
## data: scaleavaldata$year  
## W = 0.94697, p-value = 1.223e-10

Segundo o teste de Shapiro-Wilk, nenhuma das variáveis independentes segue distribuição normal e, consequentemente, a amostra não obedece ao pressuposto de normalidade multivariada, requerido para que se prossiga com a análise discriminante.

Homogeneidade de variância/covariância:

boxM(data = scaleavaldata[,1:6], grouping = scaleavaldata$classmpg)

##   
## Box's M-test for Homogeneity of Covariance Matrices  
##   
## data: scaleavaldata[, 1:6]  
## Chi-Sq (approx.) = 404.08, df = 21, p-value < 2.2e-16

Não há igualdade entre as matrizes de variâncias e covariâncias da amostra, mesmo considerando aceitável um p-valor de 0,01 (mais comum para este teste).

Seleção de variável:

discPower(variables = treinoaval[,1:6], group = treinoaval$classmpg)

## correl\_ratio wilks\_lambda F\_statistic p\_value  
## cylinders 0.5610599 0.4389401 373.23890 0.000000e+00  
## displacement 0.5495676 0.4504324 356.26593 0.000000e+00  
## horsepower 0.4354455 0.5645545 225.22203 0.000000e+00  
## weight 0.5725786 0.4274214 391.16648 0.000000e+00  
## acceleration 0.1017403 0.8982597 33.07302 2.236779e-08  
## year 0.1996819 0.8003181 72.85494 7.771561e-16

Todas as variáveis contribuem significativamente para a discriminação dos grupos (embora com menor relevância se tratando da “Aceleração” e do “Ano do Modelo”).

Determinação da função discriminante:

discrim\_aval <- desDA(variables = treinoaval[, c("displacement", "horsepower", "weight", "acceleration", "year")],   
 group = treinoaval$classmpg)

Coeficientes das funções discriminantes:

discrim\_aval$discrivar

## DF1  
## constant -0.004868575  
## displacement -0.606862514  
## horsepower 0.467179185  
## weight -1.262871937  
## acceleration 0.033320876  
## year 0.477663363

Como há apenas dois grupos, só é necessária uma função discriminante.

Autovalores das funções discriminantes e variabilidade explicada:

discrim\_aval$values

## value proportion accumulated  
## DF1 1.699744 100 100

Como só há uma função discriminante, toda a variância do modelo é explicada por ela.

Matriz de fatores (correlação entre as variáveis explicativas e a função discriminante):

discrim\_aval$discor

## DF1  
## displacement -0.9342946  
## horsepower -0.8316493  
## weight -0.9536540  
## acceleration 0.4019941  
## year 0.5631741

A função discriminante obtida é mais fortemente ponderada pelo “Peso do Veídulo” e pelo “Deslocamento do motor”, e menos pela “Aceleração” e pelo “Ano do Modelo”.

Significância da função discriminante

summary(aov(discrim\_aval$scores ~ treinoaval$classmpg), test="Wilks")

## Df Sum Sq Mean Sq F value Pr(>F)   
## treinoaval$classmpg 1 496.3 496.3 496.3 <2e-16 \*\*\*  
## Residuals 292 292.0 1.0   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Dado o baixo p-valor, a função é considerada significante.

Centróide:

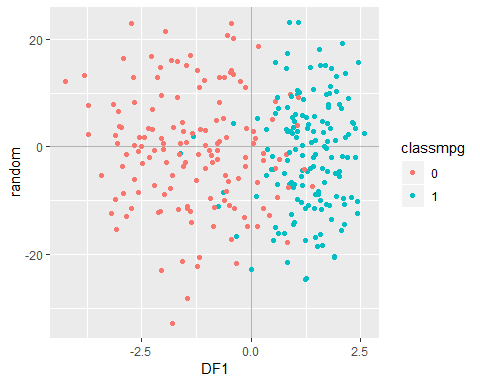
treinoaval$DF1 <- discrim\_aval$scores  
treinoaval %>% group\_by(classmpg) %>% summarise(C = mean(DF1))

## # A tibble: 2 x 2  
## classmpg C  
## <fct> <dbl>  
## 1 0 -1.31  
## 2 1 1.29

A função discriminante é desenvolvida de modo a tornar os valores do grupo dos veículos com maior consumo de combustível (classmpg = 0) negativos e do grupo dos veículos com menor consumo (classmpg = 1) positivos; isso fica explícito pelos valores discriminantes dos centróides de cada grupo

Scatterplot

treinoaval$random <- rnorm(294,mean=0,sd=10)  
  
library(ggplot2)  
ggplot(data = treinoaval, aes(x = DF1, y = random, colour = classmpg)) +  
 geom\_hline(yintercept = 0, colour="gray70") +  
 geom\_vline(xintercept = 0, colour="gray70") +  
 geom\_point()



A variável na ordenada é composta de números aleatórios, apenas para que seja mais claro visualizar a distribuição das observações ao longo do eixo dos valores discriminantes.

Como as matrizes de variâncias e covariâncias foram diferentes, é preferível que se utilize a análise discriminante quadrática, que, ao contrário da linear, não tem a igualdade das variâncias como um pressuposto.

(e) Compare a LDA e a QDA com relação à taxa de erro.

Análise Discriminante Linear

fitlda <- linDA(variables = treinoaval[, c("displacement", "horsepower", "weight", "acceleration", "year")],   
 group = treinoaval$classmpg)  
  
classiflda <- classify(fitlda, newdata = testeaval[, c("displacement", "horsepower", "weight", "acceleration", "year")])$pred\_class  
tablda <- table(classiflda, testeaval$classmpg)  
tablda

##   
## classiflda 0 1  
## 0 40 4  
## 1 10 44

acurlda <- (tablda[1,1] + tablda[2,2])/sum(tablda)  
acurlda

## [1] 0.8571429

Análise Discriminante Quadrática

fitqda <- quaDA(variables = treinoaval[, c("displacement", "horsepower", "weight", "acceleration", "year")],   
 group = treinoaval$classmpg)  
  
classidqda <- classify(fitqda, newdata = testeaval[, c("displacement", "horsepower", "weight", "acceleration", "year")])$pred\_class  
tabqda <- table(classidqda, testeaval$classmpg)  
tabqda

##   
## classidqda 0 1  
## 0 42 6  
## 1 8 42

acurqda <- (tabqda[1,1] + tabqda[2,2])/sum(tabqda)  
acurqda

## [1] 0.8571429

Embora ambas as formas de análise, linear e quadrática, obtiveram a mesma taxa de acerto, a análise linear classificou mais veículos (e, com isso, teve tanto mais acertos quanto mais erros) como pertencentes ao grupo 1, dos veículos com menor consumo de combustivel por distância percorrida.

(f) Faça uma regressão logística e avalie sua taxa de erro de acordo com alguma regra de classificação.

treinoaval$classmpg <- as.numeric(treinoaval$classmpg) -1  
testeaval$classmpg <- as.numeric(testeaval$classmpg) -1  
  
fitlgr <- glm(classmpg ~ cylinders + displacement + horsepower + weight + acceleration + year,   
 family = binomial(link = 'logit'), data = treinoaval)  
fitlgr

##   
## Call: glm(formula = classmpg ~ cylinders + displacement + horsepower +   
## weight + acceleration + year, family = binomial(link = "logit"),   
## data = treinoaval)  
##   
## Coefficients:  
## (Intercept) cylinders displacement horsepower weight   
## -1.0372 -0.1117 0.0140 -1.0437 -4.2250   
## acceleration year   
## 0.2057 1.7309   
##   
## Degrees of Freedom: 293 Total (i.e. Null); 287 Residual  
## Null Deviance: 407.6   
## Residual Deviance: 117.9 AIC: 131.9

p <- mean(treinoaval$classmpg)  
p

## [1] 0.5034014

log\_chances <- predict.glm(fitlgr, newdata = testeaval[,1:6])  
prob\_posteriori <- exp(log\_chances)/(1+exp(log\_chances))  
prob\_posteriori

## 4 6 9 16 20   
## 4.256488e-04 1.109317e-06 3.525055e-07 5.175486e-02 9.796627e-01   
## 23 26 30 34 38   
## 4.103425e-01 2.390383e-07 7.843469e-01 1.460671e-01 7.904104e-03   
## 40 48 64 66 71   
## 2.000467e-06 7.864827e-03 4.920782e-06 3.416848e-05 2.829175e-06   
## 72 79 81 83 91   
## 6.246914e-01 1.131481e-01 6.484706e-01 4.133476e-01 2.431765e-07   
## 99 102 103 114 117   
## 2.522955e-02 1.401101e-01 9.914192e-01 4.636109e-01 2.504000e-06   
## 120 122 124 126 130   
## 4.666267e-01 1.916331e-03 9.483533e-02 9.177989e-02 9.889975e-01   
## 132 135 140 143 148   
## 9.948718e-01 5.403969e-03 1.229081e-05 9.848276e-01 9.621781e-01   
## 153 158 160 162 164   
## 6.527269e-02 3.985728e-05 1.203096e-05 2.754726e-03 6.499481e-03   
## 170 174 176 180 181   
## 2.581886e-01 7.411023e-01 9.898541e-01 2.402838e-01 4.197662e-01   
## 185 191 193 194 197   
## 7.972359e-01 1.476874e-04 4.679183e-02 3.910688e-01 9.918156e-01   
## 198 202 204 206 210   
## 9.937274e-01 1.125765e-02 9.957097e-01 9.821947e-01 1.861601e-01   
## 217 218 237 242 244   
## 9.953849e-01 9.856165e-01 7.459566e-01 5.829211e-01 6.382601e-01   
## 245 247 249 257 258   
## 9.990002e-01 9.986951e-01 9.991933e-01 1.067969e-01 3.191861e-01   
## 259 260 276 281 282   
## 1.077128e-01 5.082163e-01 1.630177e-01 2.185866e-01 7.335051e-01   
## 284 290 293 294 296   
## 3.806778e-01 3.168307e-04 2.541490e-03 9.984874e-01 9.982196e-01   
## 297 298 301 302 309   
## 9.322352e-01 2.223958e-01 2.522532e-01 9.939049e-01 9.419837e-01   
## 317 321 322 324 326   
## 3.641363e-01 9.809160e-01 9.948039e-01 8.490122e-01 9.993669e-01   
## 329 334 345 351 353   
## 7.444493e-01 5.236528e-01 9.996807e-01 9.981223e-01 9.970773e-01   
## 361 363 367 373 374   
## 8.096391e-01 7.244041e-01 3.713607e-01 9.750190e-01 9.449422e-01   
## 383 387 394   
## 9.996243e-01 9.017276e-01 9.997223e-01

classiflgr <- ifelse(prob\_posteriori > p, 1, 0)   
classiflgr

## 4 6 9 16 20 23 26 30 34 38 40 48 64 66 71 72 79 81   
## 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 1   
## 83 91 99 102 103 114 117 120 122 124 126 130 132 135 140 143 148 153   
## 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 1 1 0   
## 158 160 162 164 170 174 176 180 181 185 191 193 194 197 198 202 204 206   
## 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 0 1 1   
## 210 217 218 237 242 244 245 247 249 257 258 259 260 276 281 282 284 290   
## 0 1 1 1 1 1 1 1 1 0 0 0 1 0 0 1 0 0   
## 293 294 296 297 298 301 302 309 317 321 322 324 326 329 334 345 351 353   
## 0 1 1 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1   
## 361 363 367 373 374 383 387 394   
## 1 1 0 1 1 1 1 1

tablgr <- table(classiflgr, testeaval$classmpg)  
tablgr

##   
## classiflgr 0 1  
## 0 44 7  
## 1 6 41

acurlrg <- (tablgr[1,1] + tablgr[2,2])/sum(tablgr)  
acurlrg

## [1] 0.8673469

treinoaval$classmpg <- as.factor(treinoaval$classmpg)  
testeaval$classmpg <- as.factor(testeaval$classmpg)

A regra de classificação considerou a própria probabilidade *a priori* de um veículo pertencer a determinado grupo. Veículos que tiveram uma probabilidade de pertencer ao grupo 1, segundo a regressão logística, maior que a proporção de veículos do grupo 1, foram classificados em tal grupo.

A taxa de acerto, neste caso, foi maior que o obtido da análise discriminante.

(g) Faça agora uma árvore de decisão e avalie sua taxa de erro.

set.seed(0)  
library(rpart)  
  
fitdtr <- rpart(classmpg ~ cylinders + displacement + horsepower + weight + acceleration + year + origin,  
 method="class", data = treinoaval)  
fitdtr

## n= 294   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 294 146 1 (0.496598639 0.503401361)   
## 2) displacement>=0.03906588 126 5 0 (0.960317460 0.039682540) \*  
## 3) displacement< 0.03906588 168 25 1 (0.148809524 0.851190476)   
## 6) weight>=-0.2178993 29 14 0 (0.517241379 0.482758621)   
## 12) year< 0.9556623 19 4 0 (0.789473684 0.210526316) \*  
## 13) year>=0.9556623 10 0 1 (0.000000000 1.000000000) \*  
## 7) weight< -0.2178993 139 10 1 (0.071942446 0.928057554)   
## 14) year< -0.6731187 31 9 1 (0.290322581 0.709677419)   
## 28) weight>=-0.7070666 9 3 0 (0.666666667 0.333333333) \*  
## 29) weight< -0.7070666 22 3 1 (0.136363636 0.863636364) \*  
## 15) year>=-0.6731187 108 1 1 (0.009259259 0.990740741) \*

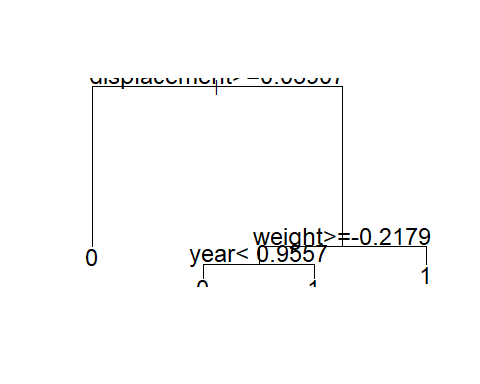
melhorCp = fitdtr$cptable[which.min(fitdtr$cptable[,"xerror"]),"CP"]  
melhorCp

## [1] 0.01027397

pfit <- prune(fitdtr, cp = melhorCp)  
pfit

## n= 294   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 294 146 1 (0.49659864 0.50340136)   
## 2) displacement>=0.03906588 126 5 0 (0.96031746 0.03968254) \*  
## 3) displacement< 0.03906588 168 25 1 (0.14880952 0.85119048)   
## 6) weight>=-0.2178993 29 14 0 (0.51724138 0.48275862)   
## 12) year< 0.9556623 19 4 0 (0.78947368 0.21052632) \*  
## 13) year>=0.9556623 10 0 1 (0.00000000 1.00000000) \*  
## 7) weight< -0.2178993 139 10 1 (0.07194245 0.92805755) \*

plot(pfit)  
text(pfit,use.n=FALSE,all=FALSE,cex=1.5)



classifdtr <- predict(pfit, testeaval[,1:7], type = 'class')  
tabdtr <- table(classifdtr, testeaval$classmpg)  
tabdtr

##   
## classifdtr 0 1  
## 0 45 5  
## 1 5 43

acurdtr <- (tabdtr[1,1] + tabdtr[2,2])/sum(tabdtr)  
acurdtr

## [1] 0.8979592

Este método de classificação obteve a maior taxa de acerto, embora não tão distante dos demais métodos.

(h) Utilize o método dos vizinhos mais próximos com k = 30 e avalie sua taxa de erro.

library(class)  
fitknn = knn(train = treinoaval[,1:7], test = testeaval[,1:7],   
 cl = treinoaval$classmpg, k = 30)  
tabknn <- table(fitknn, testeaval$classmpg)  
tabknn

##   
## fitknn 0 1  
## 0 41 5  
## 1 9 43

acurknn <- (tabknn[1,1] + tabknn[2,2])/sum(tabknn)  
acurknn

## [1] 0.8571429

o método de “k vizinhos mais próximos” obteve exatamente a mesma taxa de acerto das análises discriminantes linear e quadrática. Entretanto, tendendo a classificar mais observações como pertencentes à classe 1, mas não tanto quanto a LDA.