Random-Forest-Carseats-sales.R

patriciamaya

2020-11-03

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.0.2
```

```
data("Carseats")
Data<- Carseats[sample(nrow(Carseats)),]
attach(Data)

# We introduce the variable High to solve a classification problem as opposed to a regression problem
High <- as.factor(ifelse(Sales >= 8, "YES", "No"))
Data <- data.frame(Data,High)
Data <- Data[,-1]
colnames(Data)[11] <- "Target"

# Random Forest Model
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.2
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
rf <- randomForest(Target ~ ., data = Data, mtry = sqrt(ncol(Data) - 1), ntree = 300, proximity = T, imp
rf
```

```
##
## Call:
##  randomForest(formula = Target ~ ., data = Data, mtry = sqrt(ncol(Data) -      1), ntree = 300, proxi
##                Type of random forest: classification
##                      Number of trees: 300
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 19.75%
## Confusion matrix:
##      No YES class.error
## No  209  27   0.1144068
## YES  52 112   0.3170732
```

```
importance(rf, type = 2)
```

```
##             MeanDecreaseGini
## CompPrice          22.320119
## Income             19.820320
## Advertising        23.587966
## Population         15.477340
```

```
## Price              43.356971
## ShelveLoc          29.652303
## Age                22.827596
## Education          10.263366
## Urban               2.361034
## US                  3.249562
```

```
#rf$proximity
rf$predicted
```

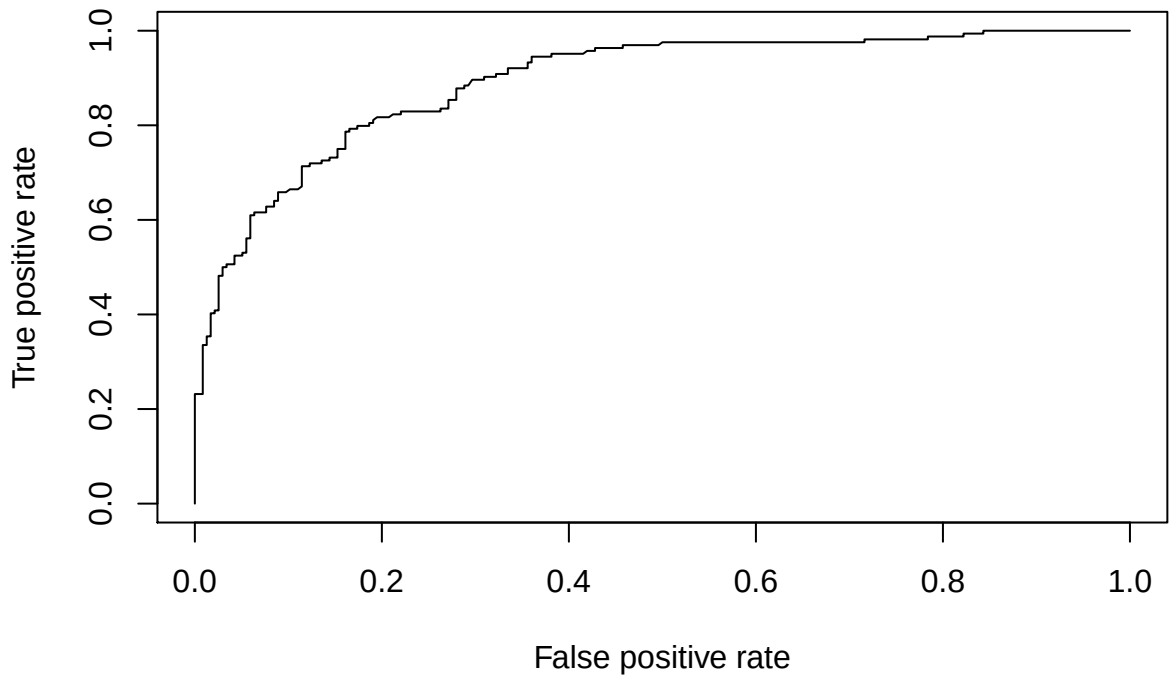
```
## 207 145 264 191 294 374 302   8 163 242 250 230  86 328 396 221 185  35 188 241
##  No YES  No YES YES  No YES YES  No YES  No YES  No  No YES YES  No  No  No YES
##  34  70 228 388  91 134 368 142 168 298 299 321 222  13 292 123 179 194 339 147
## YES  No  No  No  No  No YES  No  No  No  No  No  No  No YES  No YES YES  No  No
## 281  25 385 220 394 270 240  58  63  53 341 128 218 323 105 136 387 183 283 346
##  No YES YES YES  No  No  No  No  No  No YES  No  No YES  No YES  No  No YES  No
## 231 364 219 146  76  67  99 202  72 200 130   3 320 212  47 126  42  92 177  80
##  No YES YES YES  No  No YES  No  No  No  No YES  No YES YES YES  No  No  No  No
##  55 390  61 290 389 338 160  19 192 225   9 356 332 143  27 398  66  84  17  12
##  No  No YES  No YES  No  No YES YES  No  No  No YES  No YES  No  No  No  No YES
## 344 187 166 318 254  46 274 162 251  64 215 155 119 131 100 329  28   5 373 316
##  No YES  No  No  No  No YES  No  No YES  No YES  No YES  No  No  No  No  No  No
## 278 282  74 289 331 261  11  65 171  44 348 253 118 367 210 359 102 125  50 379
## YES YES YES  No  No YES  No YES  No  No YES  No  No  No  No  No  No  No  No  No
##  97 288 208 153 262 137 380 129 205 127 138 276 135  78  29 350 386 336 109 400
## YES  No  No YES  No  No  No  No YES YES  No  No  No YES  No YES  No  No  No YES
## 246 248 305 149 308 256 372 111  88 361  57 358 108 334 159 355 175 199 237 214
## YES  No YES  No  No YES  No YES YES YES YES YES  No  No YES  No  No  No YES  No
##  85 378 244  79 107 243 252 201 354  36 371   6  32 184  38 275  20 158 195 303
##  No  No  No  No  No  No  No  No YES YES YES YES  No  No  No YES  No YES YES  No
## 176 284  71 345 209 139 392 287 342  10 234 189 269 116 267  52 349  69 178 255
##  No  No YES  No YES YES  No  No  No  No YES  No  No  No YES  No YES YES YES YES
## 141 326 266 340 197 203 152 311 277 391 148  82  18  26 271 110 133  98  95 312
##  No YES  No YES  No  No YES  No  No  No YES  No YES YES YES YES  No  No  No  No
## 140 204 319 291 245 224 295 399  93 144 317 301 260  31  94 196 190 286 101 272
## YES  No YES YES  No  No YES  No  No  No YES  No  No YES  No  No YES YES  No  No
## 310 307 198 154  56 315  30   4 258  48 232  16 393 280 343  87 170 322 279 333
## YES  No  No  No  No YES  No  No  No  No  No  No  No  No YES  No YES  No YES  No
## 115 357 233  22 104 124 236 259 313 174  51 122 384 265 120 327 353 235 238   1
##  No  No YES YES  No  No  No  No  No  No  No YES  No  No  No  No YES YES  No  No
##  62 383 351 132  14 381  90  60 186  39 309 365 157 366  81 172 164 227 362 304
##  No  No YES  No YES  No  No  No  No  No  No YES  No  No YES YES  No  No  No YES
## 223 112  59 377 169  75  49 335 369 113 103 273  54  89  77 297  33 114  24  73
##  No  No  No YES  No  No  No YES YES YES  No YES  No  No YES YES YES  No  No  No
## 161 257 337 239 173 325 296  40   7  41  45 156 324 395 226 306 249 370 285 217
##  No  No  No  No YES  No  No  No  No  No  No YES YES  No  No  No  No YES  No  No
## 182 263 330 300 376  23 268 363 180 165 121 347 216 314 397 382  96 167  43 213
## YES  No YES YES  No  No  No  No  No  No  No  No  No  No  No  No  No  No YES YES
## 247 360  37  83 293 211  15 375 181 150 229 151 352  68 206 117   2 106  21 193
##  No  No YES YES YES  No YES  No  No YES  No YES YES YES  No  No YES  No  No  No
## Levels: No YES
```

```
#rf$votes

library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 4.0.2
```

```
score <- rf$votes[, 2]
pred <- prediction(score, Data$Target)
perf <- performance(pred, "tpr", "fpr")
plot(perf)
```




```
# How to find the best cut-off point in ROC curve
# The performance() function for ROC curve returns
# tpr, fpr and alpha-values (cut-off points). We need
# to write a function that receives these information and
# returns the best cut-off point
# Hence the input argument to the following function is perf

opt.cut <- function(perf){
  # mapply function applies the function FUN to all perf@x.values, perf@y.values,perf@alpha.values
  cut.ind <- mapply(FUN = function(x,y,p){d=(x-0)^2+(y-1)^2 # We compute the distance of all the points
  ind<- which(d==min(d)) # We find the index of the point that is closest to the corner
  c(recall = y[[ind]], specificity = 1-x[[ind]],cutoff = p[[ind]])},perf@x.values, perf@y.values,perf@a
}

BestcutOffPoint <- opt.cut(perf)
BestcutOffPoint
```

```
##                  [,1]
## recall      0.7926829
## specificity 0.8347458
## cutoff      0.4285714
```