Curso Livre II

Aluno: Manuel Ferreira Junior

Matricula: 20180008601

Modulo III

Carregamento dos packages:

```
install.packages('mlr')
install.packages('mlbench')
install.packages('rpart.plot')
library(mlr)
library(mlbench)
library(rpart.plot)
```

Tarefa :

1. Obtenha um conjunto de dados de câncer de mama usando os seguintes comandos (precisa instalar o pacote mlbench):

```
data(BreastCancer, package = "mlbench")
df = BreastCancer
df$Id = NULL
cl = df$Class
indx = sapply(df, is.factor)
df[indx] = lapply(df[indx], function(x) as.numeric(as.character(x)))
df$Class = cl
df$Bare.nuclei = NULL
```

2. Com os dados carregados, separe-os em conjunto de treinamento e de teste e use a biblioteca mlr como vimos:

```
set.seed(0)
trein_index <- sample(1:nrow(df), 0.8 * nrow(df))
test_index <- setdiff(1:nrow(df), trein_index)
trein <- df[trein_index,]
test <- df[test_index,]
```

3. Crie a tarefa de classificação (a variável alvo é "Class")

```
> taskclf <- makeClassifTask(data=trein, target = 'Class', positive =
'benign')
```

4. Treine os modelos que vimos na primeira aula (e outros, se desejar)

Decision Tree Classifier

```
> tree <- makeLearner('classif.rpart', predict.type = 'prob')
> tree_trein <- train(learner = tree, task = taskclf)
> rpart.plot(tree_trein$learner.model, roundint = F)
```

K-Nearest Neighbors (KNN)

```
> knn <- makeLearner('classif.knn')
> knn_trein <- train(learner = knn, task = taskclf)
```

Logistic Regression

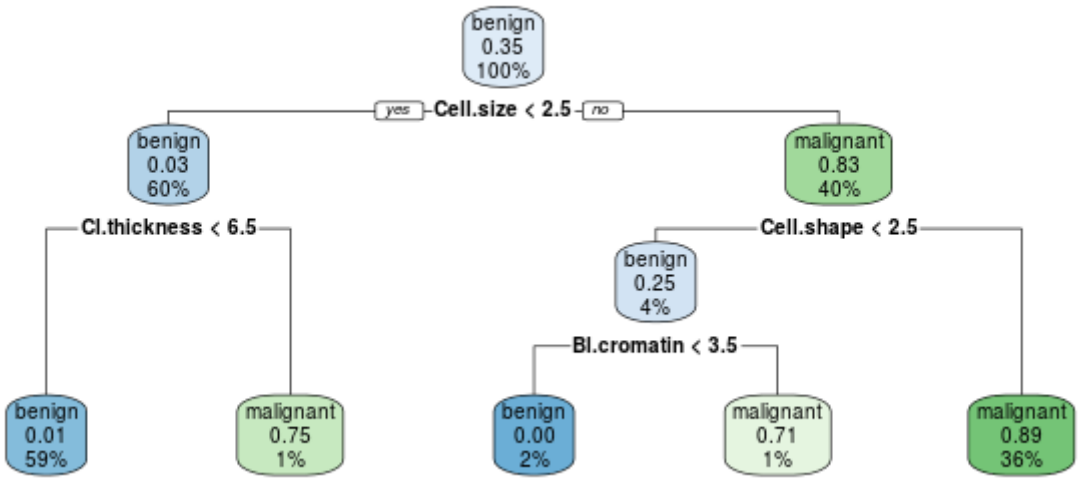
```
> reg_log <- makeLearner('classif.logreg', predict.type = 'prob')
> reg_log_trein <- train(learner = reg_log, task = taskclf)
```

5. Avalie os modelos treinados usando o conjunto de teste e veja qual teve melhor desempenho
   Decision Tree Classifier

```
> tree_predict <- predict(tree_trein, newdata = test)
> calculateROCMeasures(tree_predict)
           predicted
true        benign    malignant
  benign    96        4         tpr: 0.96 fnr: 0.04
  malignant 5         35        fpr: 0.12 tnr: 0.88
            ppv: 0.95 for: 0.1  lrp: 7.68 acc: 0.94
            fdr: 0.05 npv: 0.9  lrm: 0.05 dor: 168


Abbreviations:
tpr - True positive rate (Sensitivity, Recall)
fpr - False positive rate (Fall-out)
fnr - False negative rate (Miss rate)
tnr - True negative rate (Specificity)
ppv - Positive predictive value (Precision)
for - False omission rate
lrp - Positive likelihood ratio (LR+)
fdr - False discovery rate
npv - Negative predictive value
acc - Accuracy
lrm - Negative likelihood ratio (LR-)
dor - Diagnostic odds ratio
```

```
> calculateConfusionMatrix(tree_predict)
           predicted
true        benign malignant -err.-
  benign        96         4      4
  malignant      5        35      5
  -err.-         5         4      9
```



ppv (Precision) : 0.95

tpr (Recall): 0.96

acc (Acuracy): 0.94

K-Nearest Neighbors (KNN)

```
> knn_predict <- predict(knn_trein, newdata = test)
> calculateROCMeasures(knn_predict)
           predicted
true        benign    malignant
  benign    98        2         tpr: 0.98 fnr: 0.02
  malignant 5         35        fpr: 0.12 tnr: 0.88
            ppv: 0.95 for: 0.05 lrp: 7.84 acc: 0.95
            fdr: 0.05 npv: 0.95 lrm: 0.02 dor: 343


  Abbreviations:
  tpr - True positive rate (Sensitivity, Recall)
  fpr - False positive rate (Fall-out)
  fnr - False negative rate (Miss rate)
  tnr - True negative rate (Specificity)
  ppv - Positive predictive value (Precision)
  for - False omission rate
  lrp - Positive likelihood ratio (LR+)
  fdr - False discovery rate
  npv - Negative predictive value
  acc - Accuracy
```

```
lrm - Negative likelihood ratio (LR-)
dor - Diagnostic odds ratio
> calculateConfusionMatrix(knn_predict)
           predicted
true        benign malignant -err.-
  benign        98         2      2
  malignant      5        35      5
  -err.-         5         2      7
```

ppv (Precision) : 0.95

tpr (Recall): 0.98

acc (Acuracy): 0.95

Logistic Regression

```
> reg_log_predict <- predict(reg_log_trein, newdata = test)
> calculateROCMeasures(reg_log_predict)
           predicted
true        benign    malignant
  benign    100       0         tpr: 1    fnr: 0
  malignant 9         31        fpr: 0.22 tnr: 0.78
            ppv: 0.92 for: 0    lrp: 4.44 acc: 0.94
            fdr: 0.08 npv: 1    lrm: 0    dor: Inf


Abbreviations:
tpr - True positive rate (Sensitivity, Recall)
fpr - False positive rate (Fall-out)
fnr - False negative rate (Miss rate)
tnr - True negative rate (Specificity)
ppv - Positive predictive value (Precision)
for - False omission rate
lrp - Positive likelihood ratio (LR+)
fdr - False discovery rate
npv - Negative predictive value
acc - Accuracy
lrm - Negative likelihood ratio (LR-)
dor - Diagnostic odds ratio
> calculateConfusionMatrix(reg_log_predict)
           predicted
true        benign malignant -err.-
  benign       100         0      0
  malignant      9        31      9
  -err.-         9         0      9
```

ppv (Precision) : 0.92 \ **tpr (Recall):** 1 \ **acc (Acuracy):** 0.94

Todos :

Modelo\medida	ppv	tpr	acc
DecisionTree	0.95	0.96	0.94
KNN	0.95	0.98	0.95
LogisticReg	0.92	1	0.94

A partir dos resultados acima, é possível identificar que a regressão logística é o melhor modelo para esse problema, apresentando uma taxa de cobertura de 1, consequentemente descontando na precisão do modelo, porém é o modelo que cobre maior taxa de cobertura das classes.