Introduction to Machine Learning

Working Group "Computational Statistics" – Bernd Bischl et al.

Code demo for ROC

Data

We use the Breast Cancer data set from UCI database and try to predict the class of the cancer.
This is an **unbalanced data set**, and we manipulate the data set further to make it even more unbalanced.
The data looks like that:

```
library("dplyr")
library("mlbench")
library("mlr3")
library("mlr3learners")
library("mlr3viz")

data("BreastCancer")

target_label <- "Class"

# delete one column with missing values
bc <- BreastCancer[, -c(1, 7)]

table(BreastCancer$Class) / nrow(BreastCancer)
```

```
##
##    benign malignant
##     0.655     0.345
```

```
# transform all factors to numeric, dangerous simplification but ok here
# (don't tell the stats profs...!)
mut <- bc[, -9] %>%
  mutate_all(as.character) %>%
  mutate_all(as.numeric)
bc_data <- cbind(mut, bc[, target_label])
colnames(bc_data) <- c(colnames(mut), target_label)
# make it more unbalanced and remove 60% of the "malignant" class instances
bc_data <- bc_data %>%
  filter(
    # always keep non-malignant
    (Class != "malignant") |
      # randomly discard non-malignant with probability .6
      ((Class == "malignant") & (runif(nrow(bc_data)) < .4))
  )

head(bc_data)
```

```
##   Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
## 1            5         1          1             1            2
## 2            5         4          4             5            7
## 3            3         1          1             1            2
## 4            6         8          8             1            3
## 5            4         1          1             3            2
## 6            8        10         10             8            7
```

```
##   Bl.cromatin Normal.nucleoli Mitoses     Class
## 1           3               1       1    benign
## 2           3               2       1    benign
## 3           3               1       1    benign
## 4           3               7       1    benign
## 5           3               1       1    benign
## 6           9               7       1 malignant
```

```
table(bc_data[, target_label]) / sum(table(bc_data[, target_label]))
```

```
##
##    benign malignant
##     0.834     0.166
```

We split the data again in train and test:

```
# Data split
set.seed(1337)
train_size <- 3 / 4
train_indices <- sample(
  x = seq(1, nrow(bc_data), by = 1),
  size = ceiling(train_size * nrow(bc_data)), replace = FALSE
)
bc_train <- bc_data[ train_indices, ]
bc_test <- bc_data[ -train_indices, ]

task <- TaskClassif$new(
  id = "bc_task", backend = bc_train,
  target = target_label
)
```

Models

We check the performance of three classifiers:

Logistic regression

```
# logreg
learner_logreg <- lrn("classif.log_reg", predict_type = "prob")
learner_logreg$train(task)
pred_logreg <- learner_logreg$predict_newdata(newdata = bc_test)
pred_logreg$score(list(msr("classif.ce"), msr("classif.auc")))
```

```
##  classif.ce classif.auc
##      0.0438      0.9939
```

```
pred_logreg$confusion
```

```
##            truth
## response    benign malignant
##   benign       107         5
##   malignant      1        24
```

k-NN

```
# knn
learner_knn <- lrn("classif.kknn", k = 5, predict_type = "prob")
learner_knn$train(task)
pred_knn <- learner_knn$predict_newdata(newdata = bc_test)
pred_knn$score(list(msr("classif.ce"), msr("classif.auc")))
```

```
##  classif.ce classif.auc
##       0.073       0.956
```

```
pred_knn$confusion
```

```
##            truth
## response    benign malignant
##   benign       106         8
##   malignant      2        21
```

Featureless

.. a fairly stupid learner that simply predicts the majority of the two classes for each observation.

```
# learner that uses simple majority vote for classification
learner_stupid <- lrn("classif.featureless",
  method = "mode", predict_type = "prob"
)

learner_stupid$train(task)
pred_stupid <- learner_stupid$predict_newdata(newdata = bc_test)
pred_stupid$score(list(msr("classif.ce"), msr("classif.auc")))
```

```
##  classif.ce classif.auc
##       0.212       0.500
```

```
pred_stupid$confusion
```

```
##            truth
## response    benign malignant
##   benign       108        29
##   malignant      0         0
```

By looking at the confusion matrices we see that the problem is now, that even the stupid approach yields a reasonable mmce performance. Thus, we need additional measure: Let's compare the logistic regression and the stupid learner in terms of sensitivity[1] and specificity[2] (check if you can compute these values by hand):

ROC Curve Evaluation

[1] Also called *true positive rate* or *recall*.
[2] Also called *true negative rate*

```
pred_logreg$confusion
```

```
##            truth
## response    benign malignant
##   benign       107         5
##   malignant      1        24
```

```
pred_logreg$score(list(msr("classif.sensitivity"), msr("classif.specificity")))
```

```
## classif.sensitivity classif.specificity
##               0.991               0.828
```

```
pred_stupid$confusion
```

```
##            truth
## response    benign malignant
##   benign       108        29
##   malignant      0         0
```

```
pred_stupid$score(list(msr("classif.sensitivity"), msr("classif.specificity")))
```

```
## classif.sensitivity classif.specificity
##                   1                   0
```

A specificity of 0 means that all ill persons would be told they are healthy, which is certainly not what the test is intended for. On the other hand can we do better with the logistic regression in terms of those measures? Remember with our classification methods we try to estimate the posterior probabilities. Until now in the case of two classes we classified the observation as the first class if its posterior probability is greater or equal to 50% and otherwise as the second class. So what happens if we move this threshold?

```
pred_logreg$set_threshold(0.01)
pred_logreg$confusion
```

```
##            truth
## response    benign malignant
##   benign       108        20
##   malignant      0         9
```

```
pred_logreg$score(list(msr("classif.sensitivity"), msr("classif.specificity")))
```

```
## classif.sensitivity classif.specificity
##                1.00                0.31
```

```
pred_logreg$set_threshold(0.7)
pred_logreg$confusion
```

```
##            truth
## response    benign malignant
##   benign       106         5
##   malignant      2        24
```

```
pred_logreg$score(list(msr("classif.sensitivity"), msr("classif.specificity")))
```

```
## classif.sensitivity classif.specificity
##               0.981               0.828
```

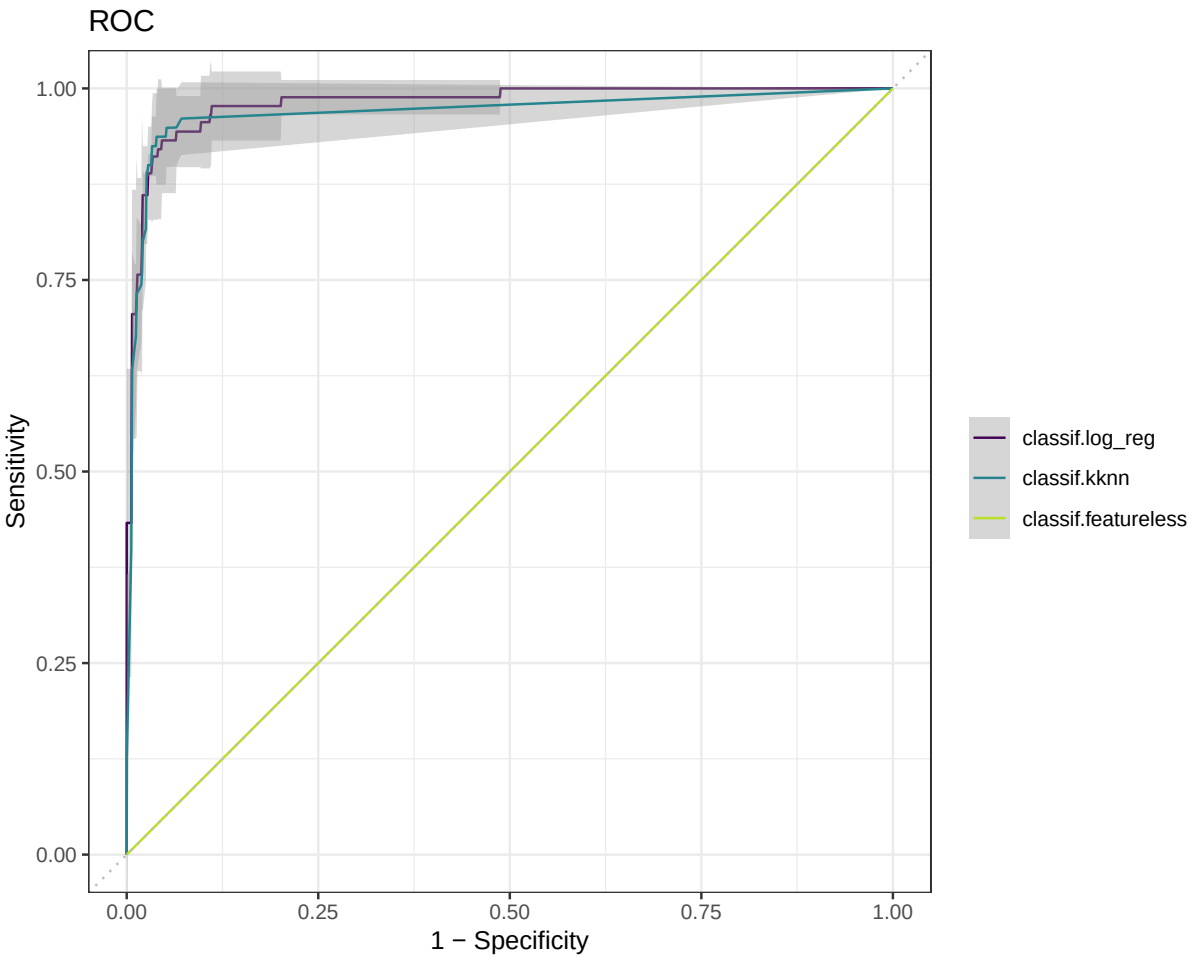
We see that in our case with a higher threshold value the specificity improves and the sensitivity degrades and vice versa. We can investigate this relationship with ROC curves. Compare the ROC curves:

```
task <- TaskClassif$new(
  id = "bc_all_task", backend = bc_data,
  target = target_label
)
learners <- list(learner_logreg, learner_knn, learner_stupid)

design <- benchmark_grid(
  tasks = task,
  learners = learners,
  resamplings = rsmp("cv", folds = 3)
)
bmr <- benchmark(design)
```

```
## INFO  [10:41:02.326] Benchmark with 9 resampling iterations
## INFO  [10:41:02.363] Applying learner 'classif.log_reg' on task 'bc_all_task' (iter 1/3)
## INFO  [10:41:02.401] Applying learner 'classif.log_reg' on task 'bc_all_task' (iter 2/3)
## INFO  [10:41:02.425] Applying learner 'classif.log_reg' on task 'bc_all_task' (iter 3/3)
## INFO  [10:41:02.450] Applying learner 'classif.kknn' on task 'bc_all_task' (iter 1/3)
## INFO  [10:41:02.483] Applying learner 'classif.kknn' on task 'bc_all_task' (iter 2/3)
## INFO  [10:41:02.517] Applying learner 'classif.kknn' on task 'bc_all_task' (iter 3/3)
## INFO  [10:41:02.552] Applying learner 'classif.featureless' on task 'bc_all_task' (iter 1/3)
## INFO  [10:41:02.586] Applying learner 'classif.featureless' on task 'bc_all_task' (iter 2/3)
## INFO  [10:41:02.600] Applying learner 'classif.featureless' on task 'bc_all_task' (iter 3/3)
## INFO  [10:41:02.627] Finished benchmark
```

```
autoplot(bmr, type = "roc")
```


ROC
Sensitivity
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
1 − Specificity
classif.log_reg
classif.kknn
classif.featureless