mlr3book

Marc Becker Martin Binder Bernd Bischl Natalie Foss Lars Kotthoff
Michel Lang Florian Pfisterer Nicholas G. Reich Jakob Richter
Patrick Schratz Raphael Sonabend Damir Pulatov

Table of contents

List of Figures

List of Tables

Preamble

```
set.seed(0)
```

Welcome to the Machine Learning in R 3 universe (mlr3verse), let us show you some of its magic. Before we begin, make sure you have installed mlr3 if you want to follow along, we recommend installing the full universe at once:

```
install.packages("mlr3verse")
```

You can also just install the base package:

```
install.packages("mlr3")
```

In this first example we'll show you the most basic use-case, train and predict.

```
library("mlr3")
task = tsk("penguins")
split = partition(task)
learner = lrn("classif.rpart", predict_type = "prob")

learner$train(task, row_ids = split$train)
learner$model
```

```
n= 231

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 231 129 Adelie (0.441558442 0.199134199 0.359307359)
  2) flipper_length< 207.5 145  44 Adelie (0.696551724 0.296551724 0.006896552)
    4) bill_length< 44.65 100   2 Adelie (0.980000000 0.020000000 0.000000000) *
    5) bill_length>=44.65 45   4 Chinstrap (0.066666667 0.911111111 0.022222222) *
  3) flipper_length>=207.5 86   4 Gentoo (0.011627907 0.034883721 0.953488372) *
```

```
predictions = learner$predict(task, row_ids = split$test)
predictions
```

```
<PredictionClassif> for 113 observations:
```

```
    row_ids     truth  response prob.Adelie prob.Chinstrap prob.Gentoo
          3    Adelie    Adelie  0.98000000     0.02000000  0.00000000
          4    Adelie    Adelie  0.98000000     0.02000000  0.00000000
          5    Adelie    Adelie  0.98000000     0.02000000  0.00000000
---
        341 Chinstrap    Adelie  0.98000000     0.02000000  0.00000000
        343 Chinstrap    Gentoo  0.01162791     0.03488372  0.95348837
        344 Chinstrap Chinstrap  0.06666667     0.91111111  0.02222222
```

```
predictions$score(msr("classif.acc"))
```

```
classif.acc
  0.9380531
```

Here we have picked the 'penguins' task (which is mlr3 language for dataset), randomly split the task into 67% training data and 33% testing data, trained a random forest on the training data to learn the probability of an observation falling into one of the outcome classes, showed the fitted model, and then made prediction on the test data, showed these predictions and evaluated the model using the accuracy measure.

Whilst mlr3 makes training and predicting easy, it also uses a unified interface to perform some very complex operations in just a few lines of code:

```
library(mlr3verse)
library(mlr3pipelines)
library(mlr3benchmark)

tasks = tsks(c("breast_cancer", "sonar"))
tuned_rf = auto_tuner(
    tnr("grid_search", resolution = 5),
    lrn("classif.ranger", num.trees = to_tune(200, 500)),
    rsmp("holdout")
)
tuned_rf = pipeline_robustify(NULL, tuned_rf, TRUE) %>>%
    po("learner", tuned_rf)
stack_lrn = ppl(
    "stacking",
    base_learners = lrns(c("classif.rpart", "classif.kknn")),
    lrn("classif.log_reg"))
stack_lrn = pipeline_robustify(NULL, stack_lrn, TRUE) %>>%
    po("learner", stack_lrn)

learners = c(tuned_rf, stack_lrn)
bm = benchmark(benchmark_grid(tasks, learners, rsmp("holdout")))
```

```
bma = bm$aggregate(msr("classif.acc"))[, c("task_id", "learner_id", "classif.acc")]
bma$learner_id = rep(c("RF", "Stack"), 2)
bma
```

```
         task_id learner_id classif.acc
1: breast_cancer         RF   0.9605263
2: breast_cancer      Stack   0.9122807
```