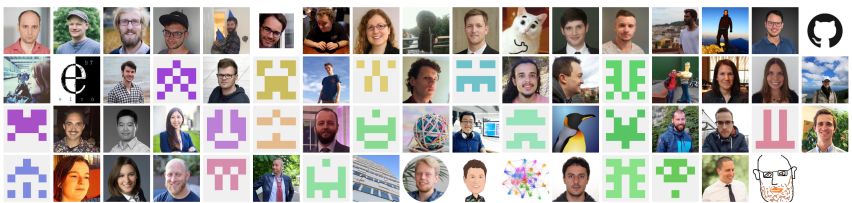
mlr: Machine Learning in R

mlr core team, represented by Lars Kotthoff
larsko@uwyo.edu



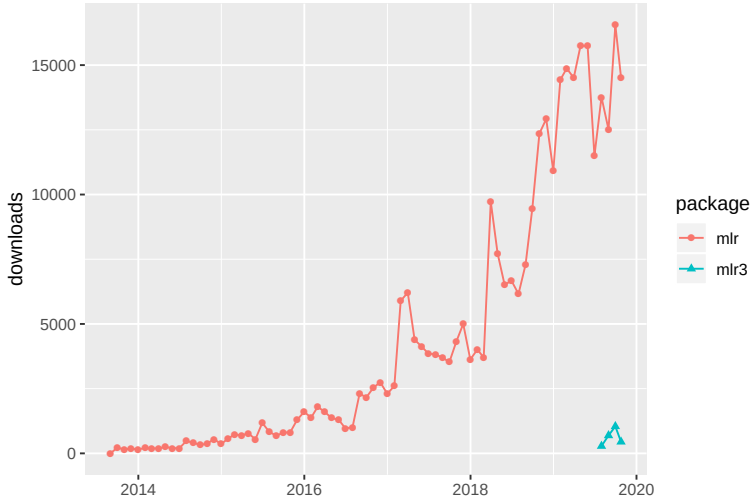
ODSC West, 01 November 2019

Overview

- unified interface to machine learning algorithms in R with a common infrastructure for ML tasks
- 89 classification, 59 regression, 12 survival, 10 clustering, 3 multi-label learners
- added value: benchmarking, feature selection through wrappers, cost-sensitive classification, hyperparameter tuning...

Some History

- first release in 2013, 68 contributors, 1,300 stars on Github
- thousands of individual contributions, several Google Summer of Code projects



mlr3: The Next Generation

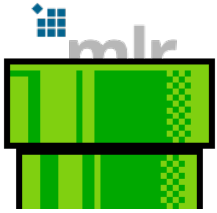
- mlr large monolithic package, integrated with hundreds of other R packages
  - → difficult to maintain and extend
  - → hundreds of dependencies to install
  - → changes in other learners (without tests) broke tests in mlr and prevented releases
  - → long build and test times
  - → R ecosystem evolved
- mlr3 completely redesigned and reimplemented from the ground up as ecosystem of modular small packages
- mlr now in maintenance-only mode

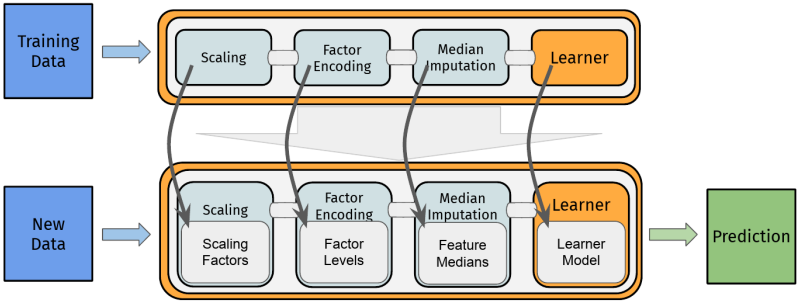
mlr3 Features

- (almost) everything you loved in mlr
- support for spatial and temporal data
- database backends
- integration with Bayesian optimization, Hyperband, racing, OpenML
- planned: visualizations, probabilistic learning, functional data analysis, deep learning

mlr3pipelines

- powerful DSL for ML graphs
- preprocessing, feature selection, ensembles, stacking...
- graphs themselves treated as Learners: can be trained, benchmarked, tuned...
- build your own automated data analysis, data cleaning solutions, AutoML systems...

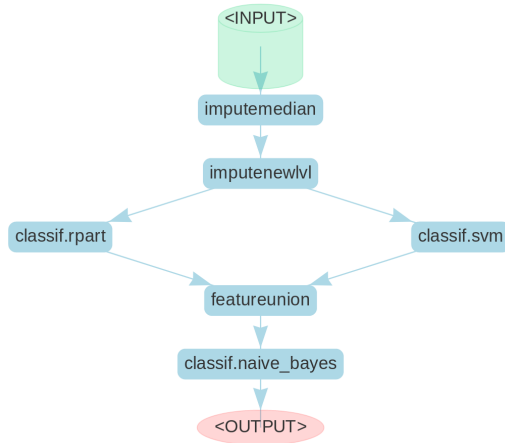




30 Second Example: Defining a Graph

```
library(mlr3)
library(mlr3learners)
library(mlr3pipelines)

stack = po("imputemedian") %>>%
  po("imputenewlvl") %>>%
  list(
    po("learner_cv", lrn("classif.rpart")),
    po("learner_cv", lrn("classif.svm"))
  ) %>>%
  po("featureunion") %>>%
  lrn("classif.naive_bayes")
```



30 Second Example: Tuning a Graph

```
library(paradox)
library(mlr3tuning)

params = ParamSet$new(list(
  ParamFct$new("classif.svm.kernel", levels = c("polynomial", "radial")),
  ParamDbl$new("classif.svm.gamma", lower = -15, upper = 15),
  ParamInt$new("classif.svm.degree", lower = 1, upper = 5)))$
    add_dep("classif.svm.degree", "classif.svm.kernel",
            CondEqual$new("polynomial"))
params$trafo = function(x, param_set)
  { x$classif.svm.gamma = 2^x$classif.svm.gamma; x }

instance = TuningInstance$new(
  mlr_tasks$get("sonar"), stack, rsmp("cv", folds = 5),
  msr("classif.acc"), params, term("evals", n_evals = 50)
)

tnr("random_search")$tune(instance)
```

Further Information

- https://mlr3.mlr-org.com/
- mlr3 book (work in progress) https://mlr3book.mlr-org.com/
- contributions welcome https://github.com/mlr-org/mlr3
- slack https://mlr-org.slack.com/
- tutorial at 11am