Machine Learning for spatiotemporal data using {mlr3}

Marc Becker, Patrick Schratz et al.

2021-09-10, OpenDataScience Europe Conference

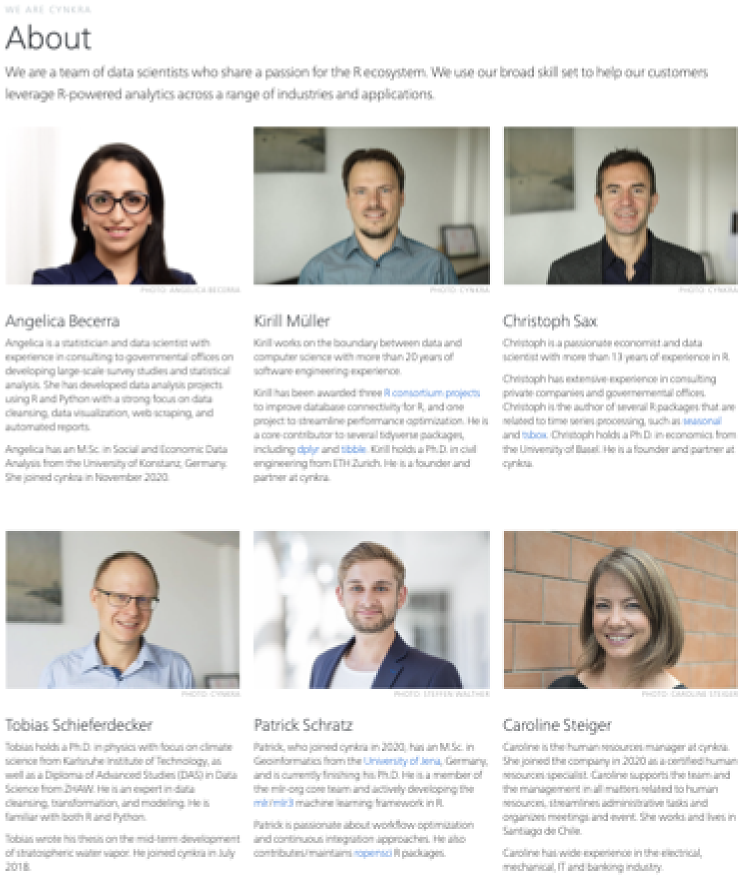
Who I am

- M.Sc. Geoinformatics
- Previously researcher at University of **Jena** and LMU **Munich**
- Now R consultant in Zurich, Switzerland
- PhD Candidate (Environmental modelling)

- Unix & R enthusiast
- Gitea (https://gitea.io) contributor
- Member of mlr-org core team; Machine learning in R
  - mlr3 - https://github.com/mlr-org/mlr3
  - mlr - https://github.com/mlr-org/mlr

Where I work

- Swiss-based R consulting company (Zurich), founded in 2018 - www.cynkra.com
- 5 - 10 people from 7 different countries
- Strong Free and Open-Source (FOSS) philosophy
- RStudio Certified Partner



WE ARE CYNKRA

About

We are a team of data scientists who share a passion for the R ecosystem. We use our broad skill set to help our customers leverage R-powered analytics across a range of industries and applications.

PHOTO: ANGELICA BECERRA

Angelica Becerra

Angelica is a statistician and data scientist with experience in consulting to governmental offices on developing large-scale survey studies and statistical analysis. She has developed data analysis projects using R and Python with a strong focus on data cleansing, data visualization, web scraping, and automated reports.

Angelica has an M.Sc. in Social and Economic Data Analysis from the University of Konstanz, Germany. She joined cynkra in November 2020.

PHOTO: CYNKRA

Kirill Müller

Kirill works on the boundary between data and computer science with more than 20 years of software engineering experience.

Kirill has been awarded three R consortium projects to improve database connectivity for R, and one project to streamline performance optimization. He is a core contributor to several tidyverse packages, including dplyr and tibble. Kirill holds a Ph.D. in civil engineering from ETH Zurich. He is a founder and partner at cynkra.

PHOTO: CYNKRA

Christoph Sax

Christoph is a passionate economist and data scientist with more than 13 years of experience in R.

Christoph has extensive experience in consulting private companies and governemental offices. Christoph is the author of several R packages that are related to time series processing, such as seasonal and tsbox. Christoph holds a Ph.D. in economics from the University of Basel. He is a founder and partner at cynkra.

PHOTO: CYNKRA

Tobias Schieferdecker

Tobias holds a Ph.D. in physics with focus on climate science from Karlsruhe Institute of Technology, as well as a Diploma of Advanced Studies (DAS) in Data Science from ZHAW. He is an expert in data cleansing, transformation, and modeling. He is familiar with both R and Python.

Tobias wrote his thesis on the mid-term development of stratospheric water vapor. He joined cynkra in July 2018.

PHOTO: STEFFEN WALTHER

Patrick Schratz

Patrick, who joined cynkra in 2020, has an M.Sc. in Geoinformatics from the University of Jena, Germany, and is currently finishing his Ph.D. He is a member of the mlr-org core team and actively developing the mlr/mlr3 machine learning framework in R.

Patrick is passionate about workflow optimization and continuous integration approaches. He also contributes/maintains ropensci R packages.

PHOTO: CAROLINE STEIGER

Caroline Steiger

Caroline is the human resources manager at cynkra. She joined the company in 2020 as a certified human resources specialist. Caroline supports the team and the management in all matters related to human resources, streamlines administrative tasks and organizes meetings and event. She works and lives in Santiago de Chile.

Caroline has wide experience in the electrical, mechanical, IT and banking industry.

1. mlr3 Overview

mlr3: Overview

- Why do we want to use mlr3?
- Key principles of mlr3

Code available at

https://gist.github.com/pat-s/ae290bd6dd8c2970c7aa0baf200483c4

Slides

https://my.cynkra.com/connect/talks/opendatascience-eu-2021/

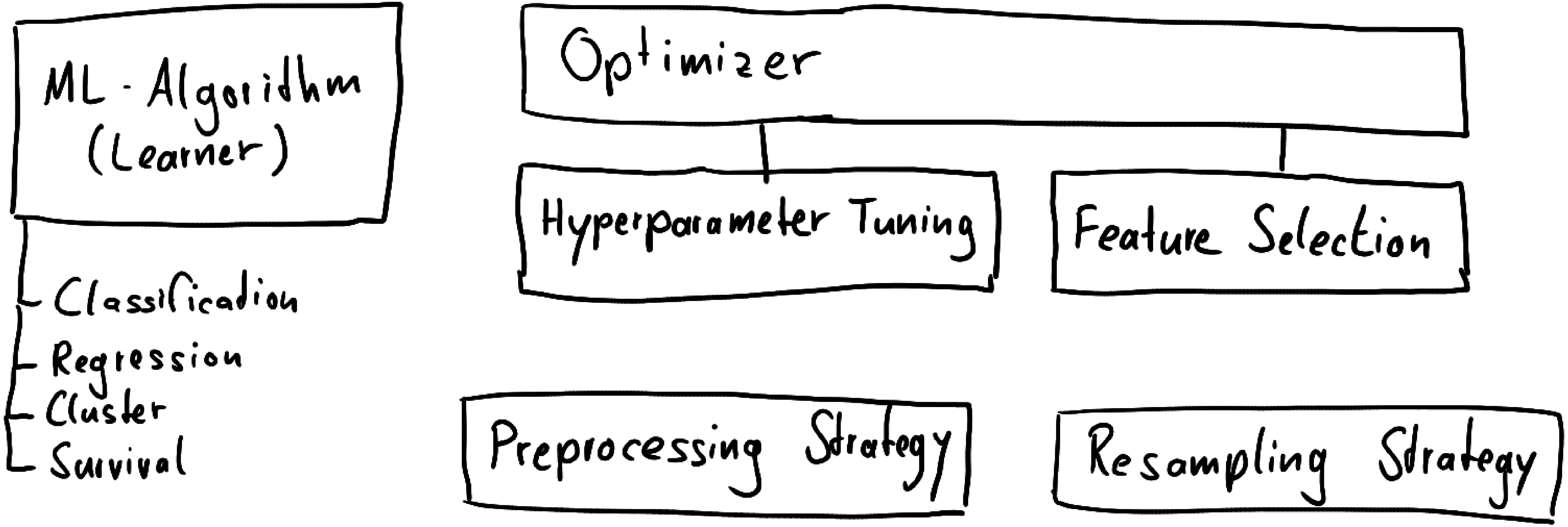
? Why use mlr3

Users want to efficiently **train**/**predict**/**benchmark**

- many **methods**
- on many **datasets**
- using different **tuning methods**
- using different **feature selection methods**
- preferably using the **same syntax**

→ *Design principles* of {mlr3}

mlr3: Overview



Motivation: Make benchmarking easy!

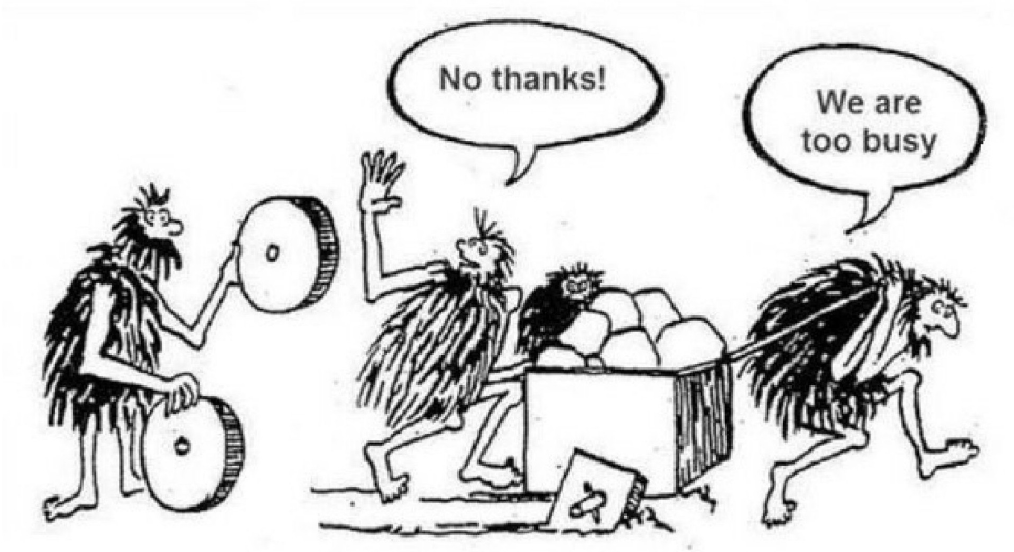
By unifying

- interfaces to **train** and **predict** methods,
- interfaces to learner **hyperparameters** and **optimizers** (tuning),
- **resampling** (performance estimation),
- **preprocessing** independently from the data,
- **parallelization**, and
- **error handling**

Source: https://giphy.com/gifs/nba-warriors-golden-state-xUPGck7rzlAftbFZza

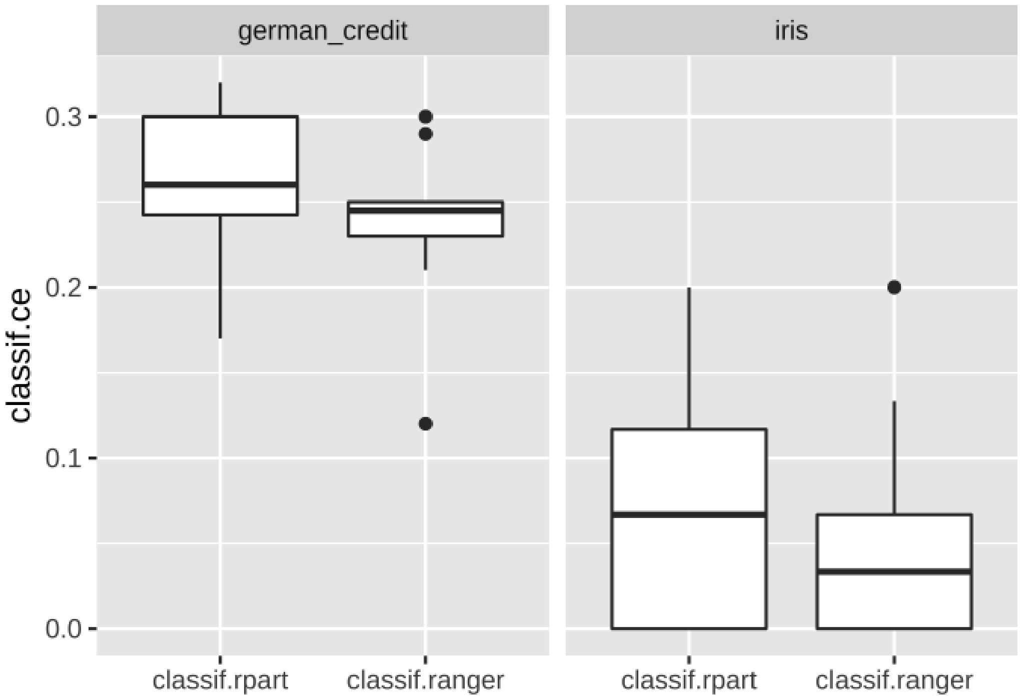
Is it worth to "learn" mlr3?

- Avoid making mistakes by relying on **tested functionality**
    - Predefined performance measures
    - Resampling
- **Easily scale up** your benchmark
    - Integrated parallelization
    - Benchmarking functions
- New methods can be easily integrated into the {mlr3verse}



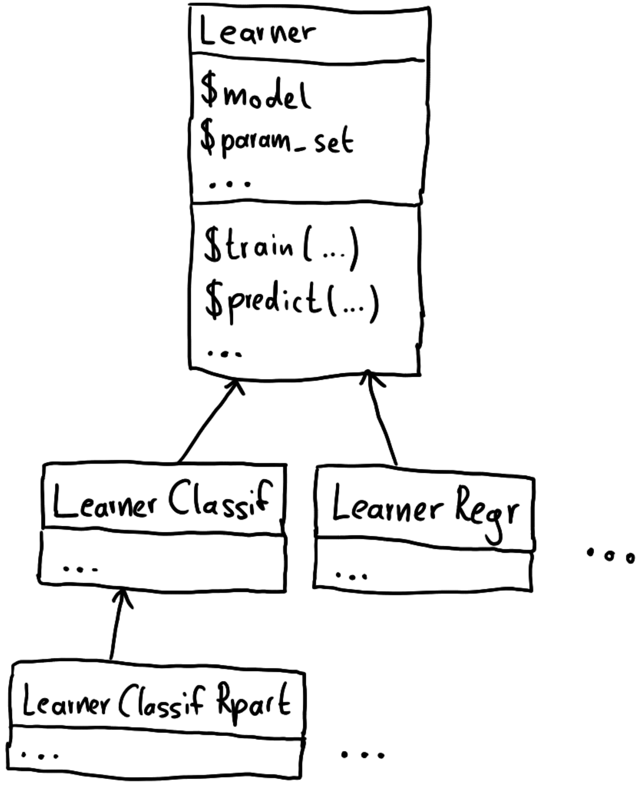
mlr3 in a nutshell

```r
library("mlr3verse", quietly = TRUE)
set.seed(42)

# example tasks
tasks <- tsks(c("iris", "german_credit"))
# from {mlr3learners}
learners <- lrns(c("classif.rpart",
  "classif.ranger"))

# run a cross-val
bmg <- benchmark_grid(
  tasks, learners,
  rsmp("cv")
)
bmr <- benchmark(bmg)

# visualize by classification error
autoplot(bmr, measure = msr("classif.ce"))
```

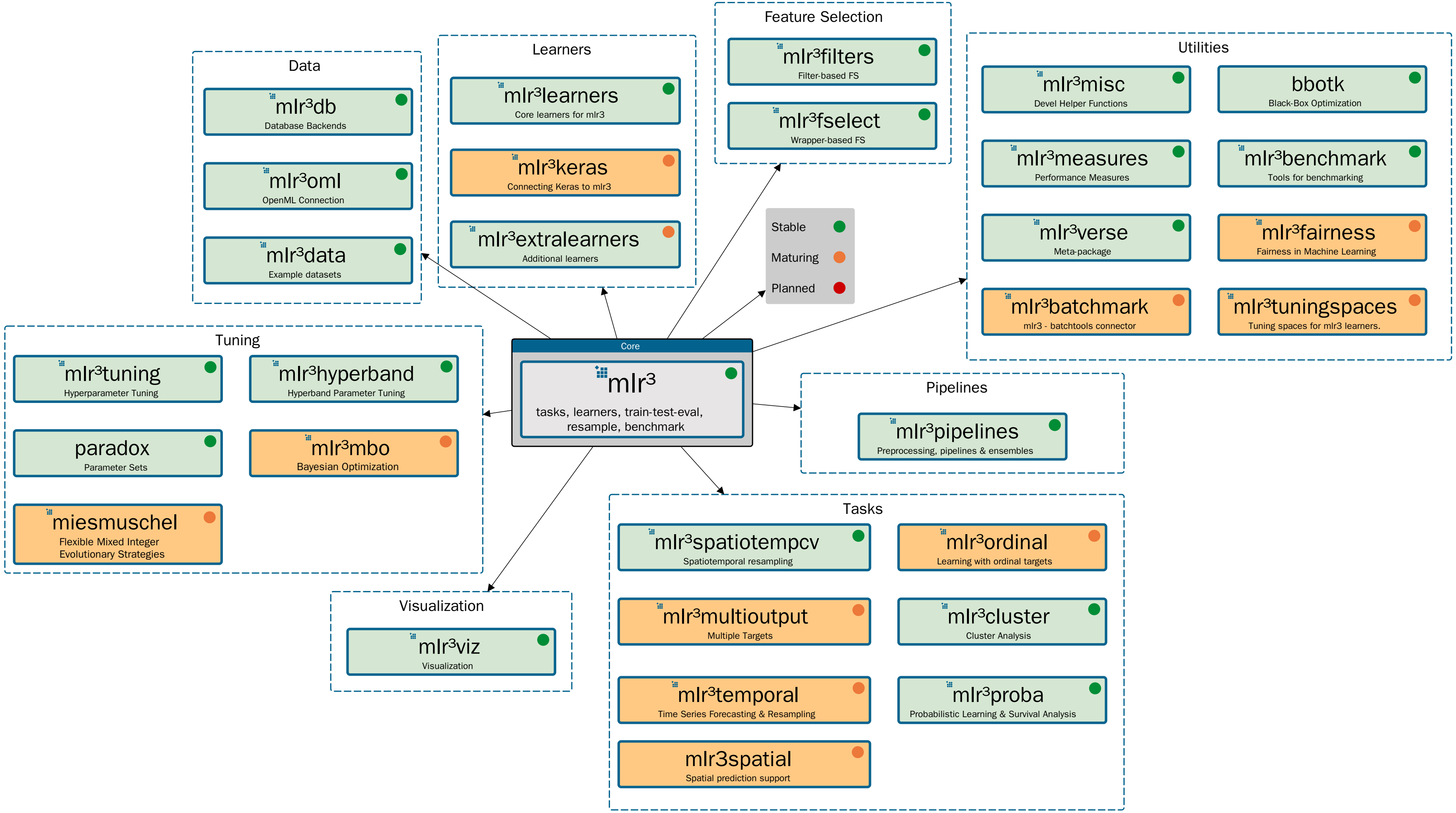


Principles of mlr3

- Overcome limitations of S3 with the help of **{R6}**
  - Truly object-oriented: data and methods live in the same object
  - Make use of inheritance
  - Make slight use of reference semantics
- Embrace **{data.table}**, both for arguments and internally
  - Fast operations for tabular data
  - List columns to arrange complex objects in tabular structure
- Be **light on dependencies**:
  - `{R6}`, `{data.table}`, `{lgr}`
  - Plus some of our own packages (`{backports}`, `{checkmate}`)
  - Special packages are loaded from mlr3 extension libraries



The mlr3verse



2. mlr3 + spatiotemporal data

mlr3 + spatiotemporal data

- How does mlr3 help in spatiotemporal/environmental/ecological modelling?
- What things do I need to be aware of?
- What is still missing?
- Can I contribute?

mlr3 + spatiotemporal data

There are currently two packages for spatiotemporal analysis in mlr3:

{mlr3spatiotempcv}

→ Spatiotemporal **resampling methods** (for cross-validation)

{mlr3spatial}

→ Spatial **DataBackends** and (parallelized) **prediction** support

Planned but unfinished (and currently unmaintained): mlr3temporal. Please reach out to us if you have knowledge in this area and think about contributing 🤝

2.1 mlr3spatial

mlr3spatial

{mlr3spatial} is new and not on CRAN yet

What's inside the tin?

- ✔ `DataBackendRaster` for ({terra}, {raster}, {stars})
- ✔ `DataBackendVector` for {sf})
- ✔ Parallel (future-based) predictions via `<learner>$predict()`
- ✔ Memory-aware chunked predictions

mlr3spatial

Predict the cadmium concentration from the `l7data` dataset (see `?stars::L7_ETMs`).

```
library("mlr3")
library("mlr3learners")
library("mlr3spatial")

tif <- system.file("tif/L7_ETMs.tif",
  package = "stars"
)
l7data <- stars::read_stars(tif)

# create mlr3 backend from sf data
backend <- as_data_backend(l7data)
```

- Load required packages
- Load the `L7` data
- Create a `DataBackendSpatial`

mlr3spatial

```r
# create a "Random Forest" learner and train it
learner <- lrn("regr.ranger")
task <- as_task_regr(backend, target = "layer.1")

rows_train <- sample(1:task$nrow, 1000)
rows_pred <- setdiff(1:task$nrow, rows_train)

learner$train(task, row_ids = rows_train)
```

- Create a `TaskRegr` with `layer1` as the response
- Train a Random Forest learner ({ranger} package) on a subset of the data (1000 obs.)

Usually one does not split a raster file into train and test - often the train set is composed from point observations and a raster is used for predictions into unknown space.

mlr3spatial

Also available as vignette "Getting Started".

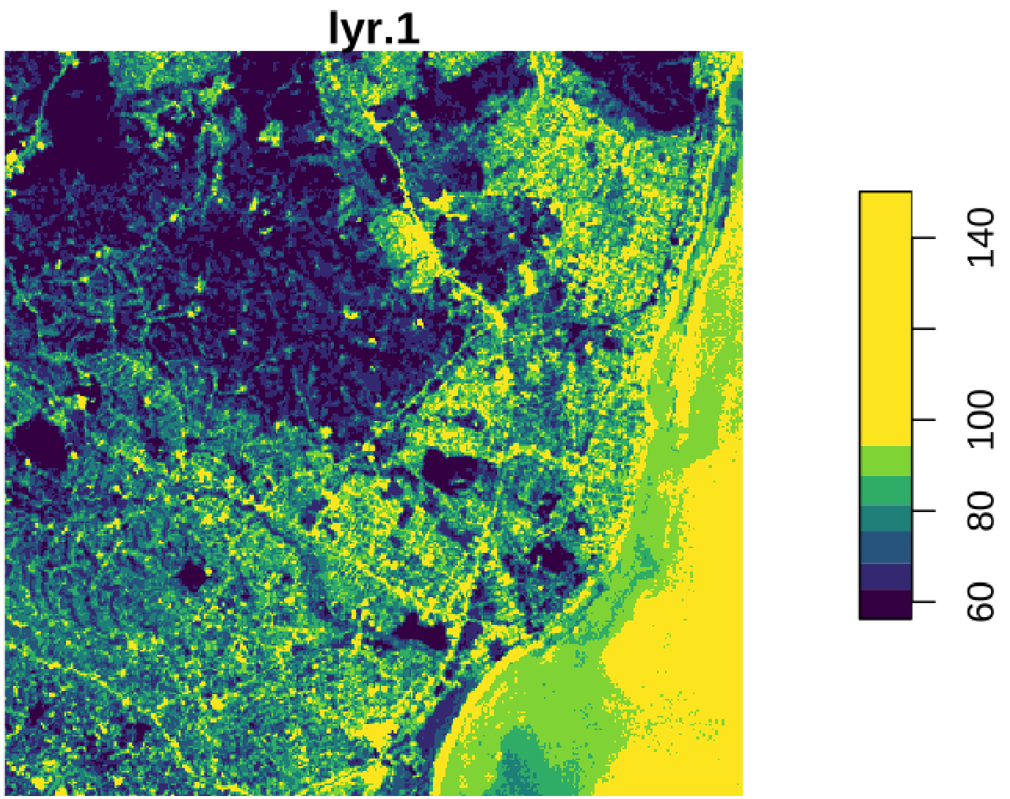
```
# set the output file and predict with the learner
pred <- predict_spatial(task, learner, format = "stars")
```

```
pred
```

```
## stars object with 2 dimensions and 1 attribute
## attribute(s):
##              Min.  1st Qu.   Median    Mean  3rd Qu.    Max.
## lyr.1  56.29713 67.00904 78.60435 78.9765 89.28517 150.165
## dimension(s):
##   from  to  offset delta                      refsys point values x/y
## x    1 349  288776  28.5 UTM Zone 25, Southern Hem... FALSE   NULL [x]
## y    1 352 9120761 -28.5 UTM Zone 25, Southern Hem... FALSE   NULL [y]
```

mlr3spatial

```
plot(pred, col = c("#440154FF", "#443A83FF", "#31688EFF",
  "#21908CFF", "#35B779FF", "#8FD744FF", "#FDE725FF"))
```



mlr3spatial

Parallel predictions

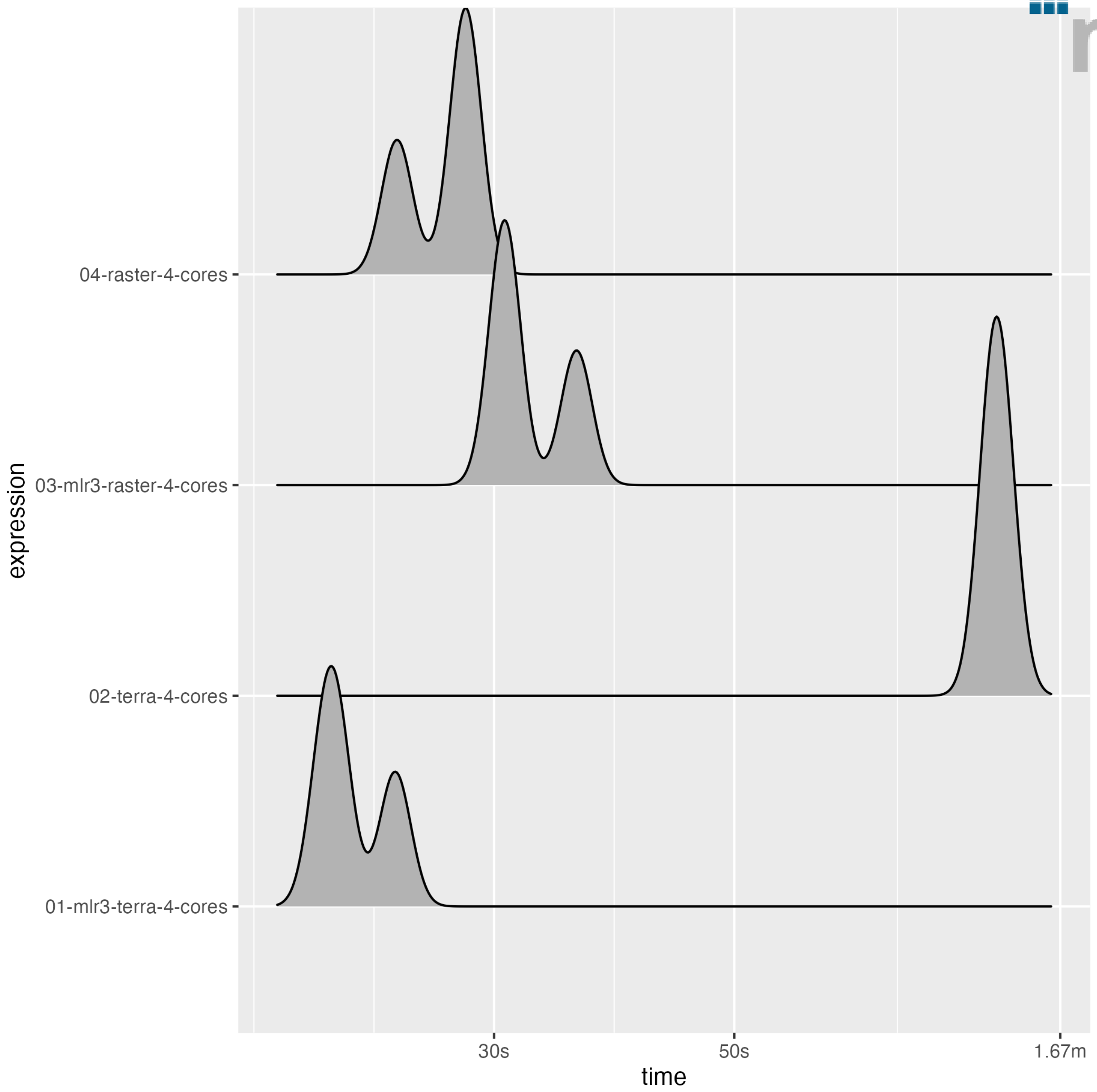
Often spatial predictions take quite some time due to the amount of points to be predicted. Especially in the field of remote sensing this can be **millions** of points and more.

While some spatial classes come with built-in parallelization, {mlr3} provides a more efficient and generalized methodology to speed up such large prediction tasks.

Check out this benchmark 👉

Source: https://mlr3spatial.mlr-org.com/articles/benchmark.html

- 500 MB file on disk
- ~ 25 Mio. values
  - `demo_stack_spatraster(500)`



2.2 mlr3spatiotempcv

mlr3spatiotempcv

- Spatiotemporal resampling methods for {mlr3}
- Aims to simplify/structure the jungle of spatiotemporal resampling methods

✔ Generic `ggplot2::autoplot()` for all methods

- Upcoming paper (JSS)

✔ Currently wraps **8** resampling methods from **4** packages

- {blockCV}
- {sperrorest}
- {CAST}
- {skmeans}

mlr3spatiotempcv

Spatiotemporal performance estimations - Essentials

→ Non-spatial resampling methods **overestimate** model performace due to **spatial autocorrelation** betweeen train and test data

! There is **no single best** method, the choice of the method should be **target-oriented** (what do I want to predict?)

? There is a debate whether spatiotemporal resampling methods **might be too pessimistic**

→ Ongoing research

mlr3spatiotempcv

Example:

- Spatial cross-validation with Random Forest ;
- Predicting **landslide** events (0/1) in Ecuador.

```
library("mlr3spatiotempcv")

# create 'sf' object from example data
data_sf <- sf::st_as_sf(ecuador, coords = c("x", "y"), crs = 32717)
```

mlr3spatiotempcv

mlr3spatiotempcv

```r
# create ClassifST task
task <- TaskClassifST$new("ecuador_sf", backend = data_sf,
  target = "slides", positive = "TRUE"
)
print(task)
```

```
## <TaskClassifST:ecuador_sf> (751 x 11)
## * Target: slides
## * Properties: twoclass
## * Features (10):
##   - dbl (10): carea, cslope, dem, distdeforest, distroad, distslidespast, hcurv, log.carea,
##     slope, vcurv
## * Coordinates:
##                X       Y
##   1: 712882.5 9560002
##   2: 715232.5 9559582
##   3: 715392.5 9560172
##   4: 715042.5 9559312
##   5: 715382.5 9560142
##  ---
## 747: 714472.5 9558482
## 748: 713142.5 9560992
## 749: 713322.5 9560562
```

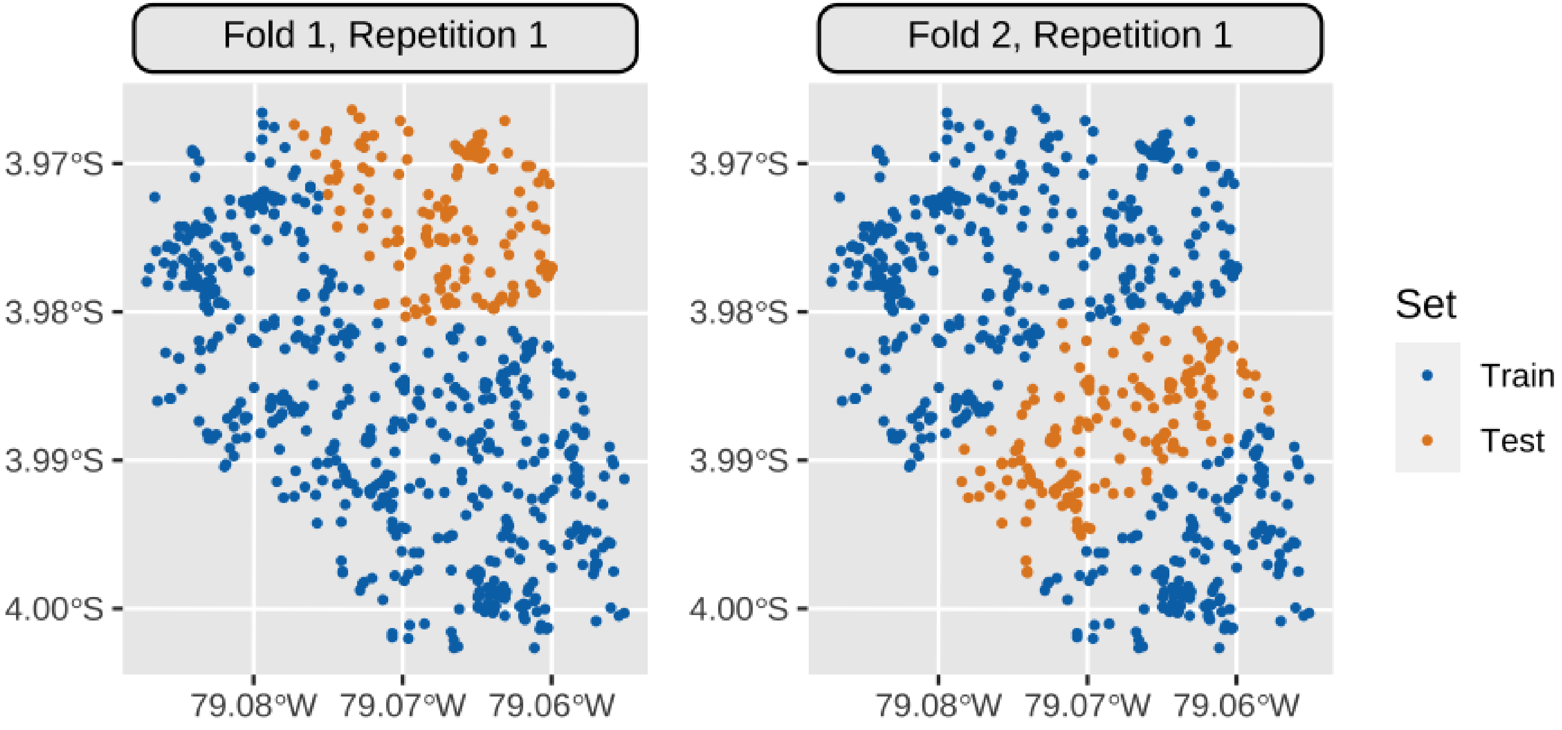
mlr3spatiotempcv

```r
library("mlr3learners")
library("ranger")
task <- tsk("ecuador")

learner <- lrn("classif.ranger", predict_type = "prob")
resampling_sp <- rsmp("repeated_spcv_coords",
  folds = 4, repeats = 2
)

rr_sp <- resample(
  task = task,
  learner = learner,
  resampling = resampling_sp
)

rr_sp$aggregate(measures = msr("classif.ce"))
```

```
## classif.ce
##  0.3585072
```

mlr3spatiotempcv

```
autoplot(resampling_sp, task, fold_id = c(1:2), size = 0.7) *
  ggplot2::scale_y_continuous(breaks = seq(-3.97, -4, -0.01)) *
  ggplot2::scale_x_continuous(breaks = seq(-79.06, -79.08, -0.01))
```



mlr3spatiotempcv

More resources

- See the **"Spatiotemporal Analysis"** chapter in the mlr3book (https://mlr3book.mlr-org.com/special-tasks.html#spatiotemporal)
- Function reference of {mlr3spatiotempcv}: https://mlr3spatiotempcv.mlr-org.com/reference/index.html
- Literature: Roberts et al. 2017, Schratz et al. 2019

mlr3spatiotempcv

What about (spatio)-temporal methods?

- Two methods (`"sptcv_cstf"` and `"sptcv_cluto"`) support both space and time
- Spatiotemporal resampling is non-trivial due to the involvment of multiple dimensions
- We would love to see help/contributions from the community for {mlr3temporal}

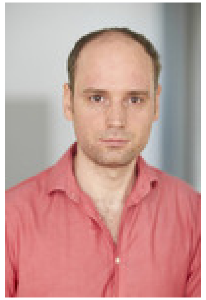
Acknowledgements

Thanks to **Marc Becker** for his help developing mlr3 spatial packages.

Thanks to mlr-org's GitHub sponsors (especially **OpenGeoHub** and **cynkra**).

Thanks to **you** for being interested in / using mlr3!

Bernd Bischl



Michel Lang

Lars Kothoff

Jakob Richter

Martin Binder

Patrick Schratz

Flo Pfisterer

Marc Becker

L. Schneider

R. Sonabend