Hyperparameter Optimization

Foundations, Algorithms, Best Practices and Open Challenges

Martin Binder
LMU, MCML

Slides:
https://tinyurl.com/mlr3vfs

Modern Machine Learning in R

R-packages:

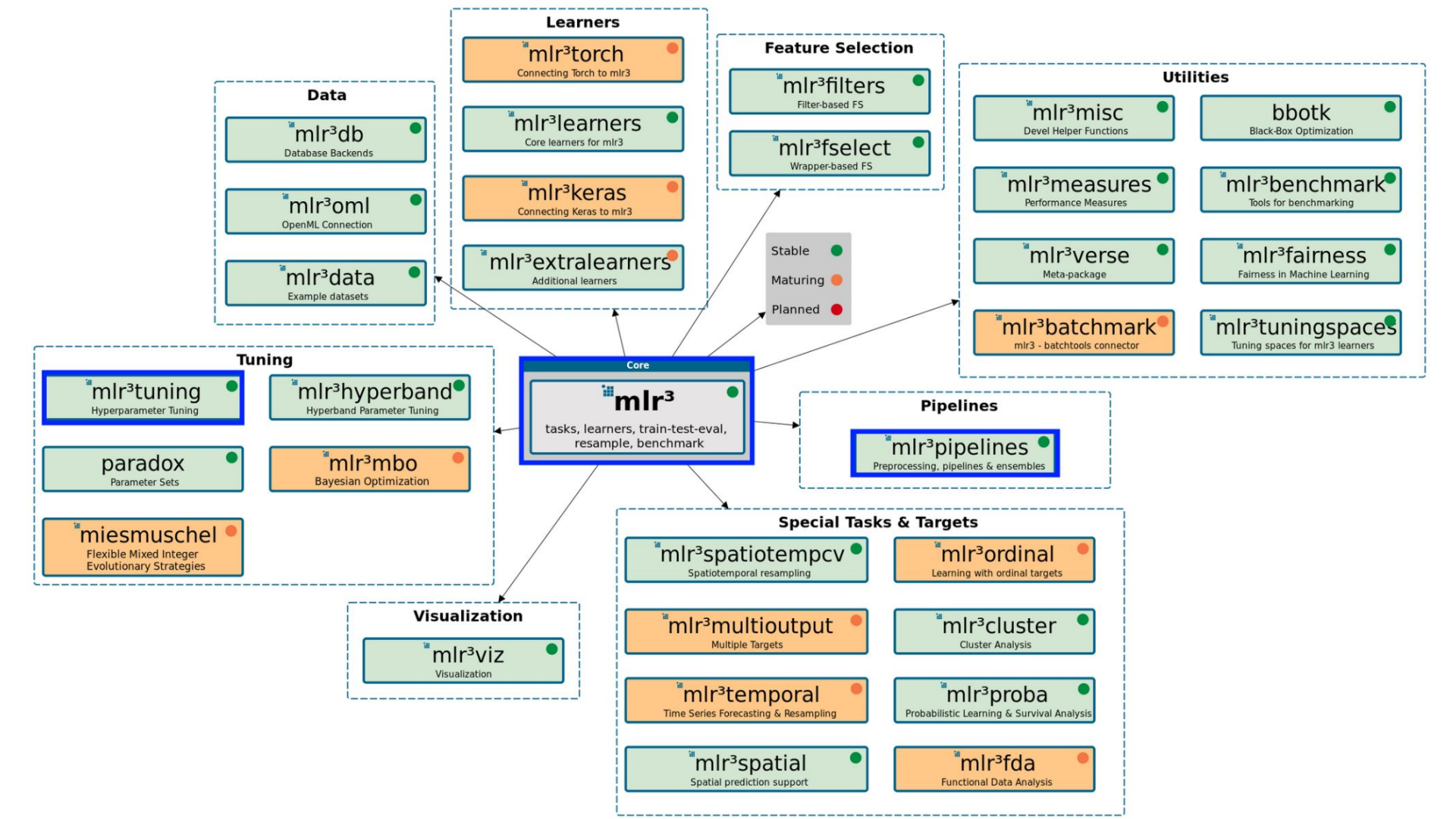
- **mlr3**
- **mlr3tuning**
- **mlr3pipelines**

(and others)

Modern Machine Learning in R

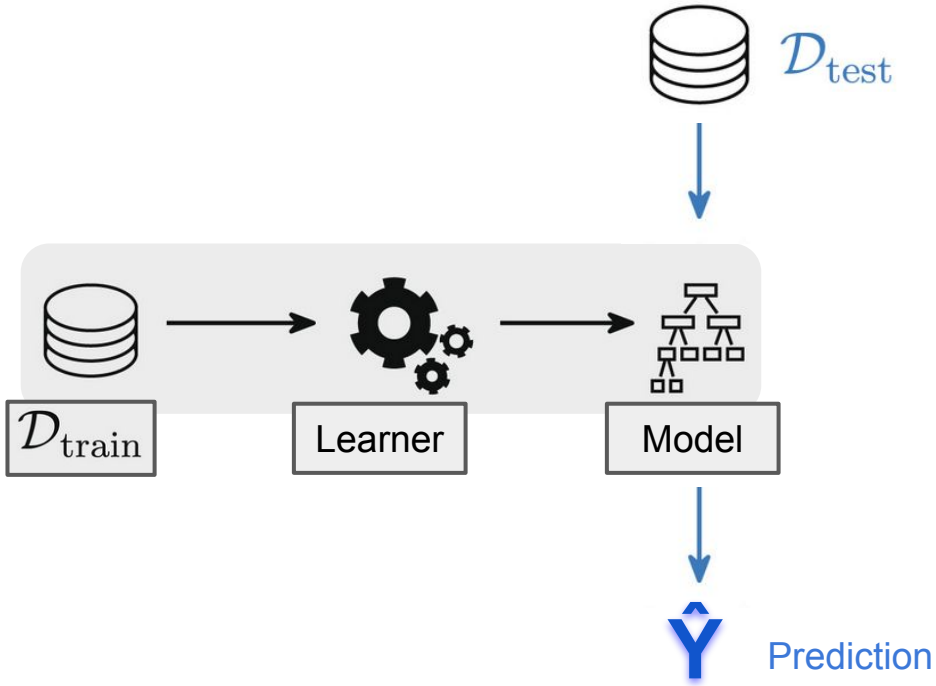
Principles of **`mlr3`**:

- Object oriented (R6)
  - algorithms are objects
  - ... with unified interfaces
  - ... that can be combined in a modular way
- Focus on AutoML
  - Hyperparameter-optimization
  - Machine-learning pipelines

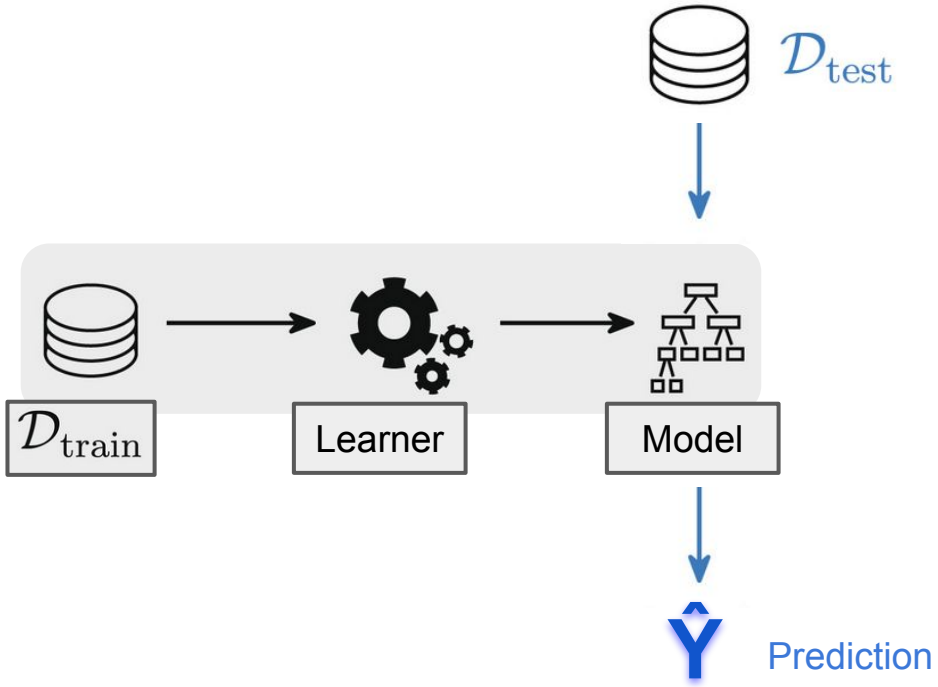

Learners
mlr³torch
Connecting Torch to mlr3
mlr³learners
Core learners for mlr3
mlr³keras
Connecting Keras to mlr3
mlr³extralearners
Additional learners
Feature Selection
mlr³filters
Filter-based FS
mlr³fselect
Wrapper-based FS
Utilities
mlr³misc
Devel Helper Functions
bbotk
Black-Box Optimization
mlr³measures
Performance Measures
mlr³benchmark
Tools for benchmarking
mlr³verse
Meta-package
mlr³fairness
Fairness in Machine Learning
mlr³batchmark
mlr3 - batchtools connector
mlr³tuningspaces
Tuning spaces for mlr3 learners
Data
mlr³db
Database Backends
mlr³oml
OpenML Connection
mlr³data
Example datasets
Stable
Maturing
Planned
Tuning
mlr³tuning
Hyperparameter Tuning
mlr³hyperband
Hyperband Parameter Tuning
paradox
Parameter Sets
mlr³mbo
Bayesian Optimization
miesmuschel
Flexible Mixed Integer
Evolutionary Strategies
Core
mlr³
tasks, learners, train-test-eval,
resample, benchmark
Pipelines
mlr³pipelines
Preprocessing, pipelines & ensembles
Visualization
mlr³viz
Visualization
Special Tasks & Targets
mlr³spatiotempcv
Spatiotemporal resampling
mlr³ordinal
Learning with ordinal targets
mlr³multioutput
Multiple Targets
mlr³cluster
Cluster Analysis
mlr³temporal
Time Series Forecasting & Resampling
mlr³proba
Probabilistic Learning & Survival Analysis
mlr³spatial
Spatial prediction support
mlr³fda
Functional Data Analysis

Machine Learning

Machine Learning

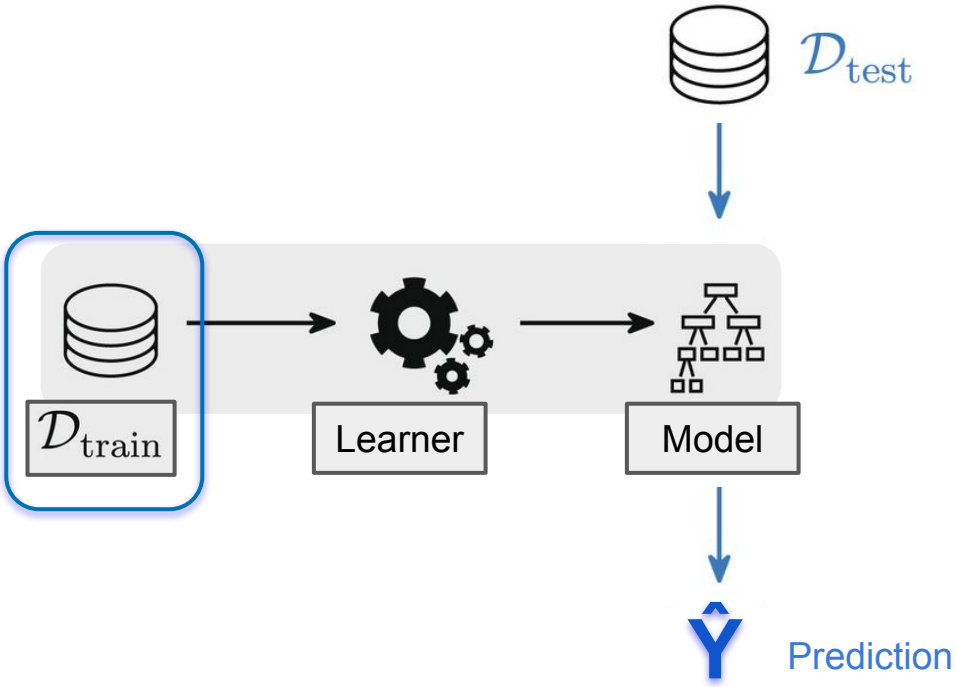


Machine Learning



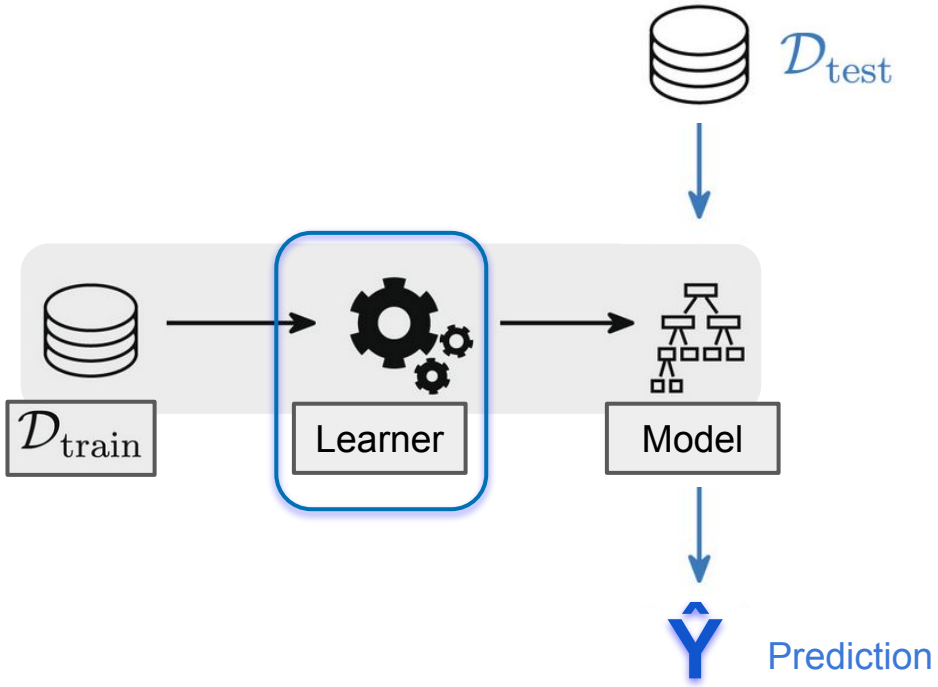

```r
data("german", package = "rchallenge")

library("mlr3")
```

Machine Learning



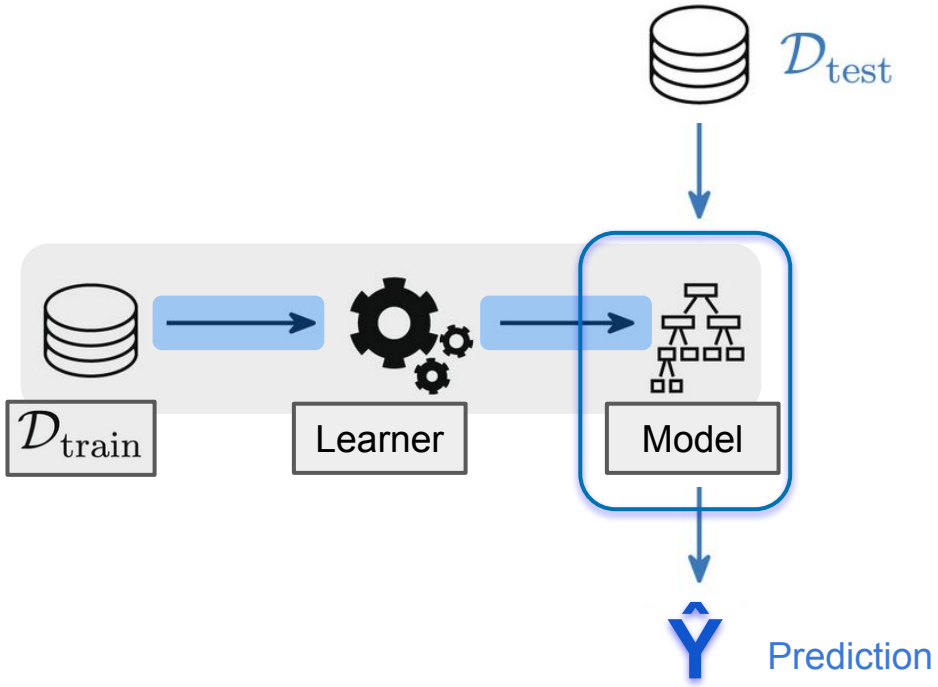

```r
data("german", package = "rchallenge")

library("mlr3")

task <- as_task_classif(
  german,
  target = "credit_risk"
)
```

Machine Learning



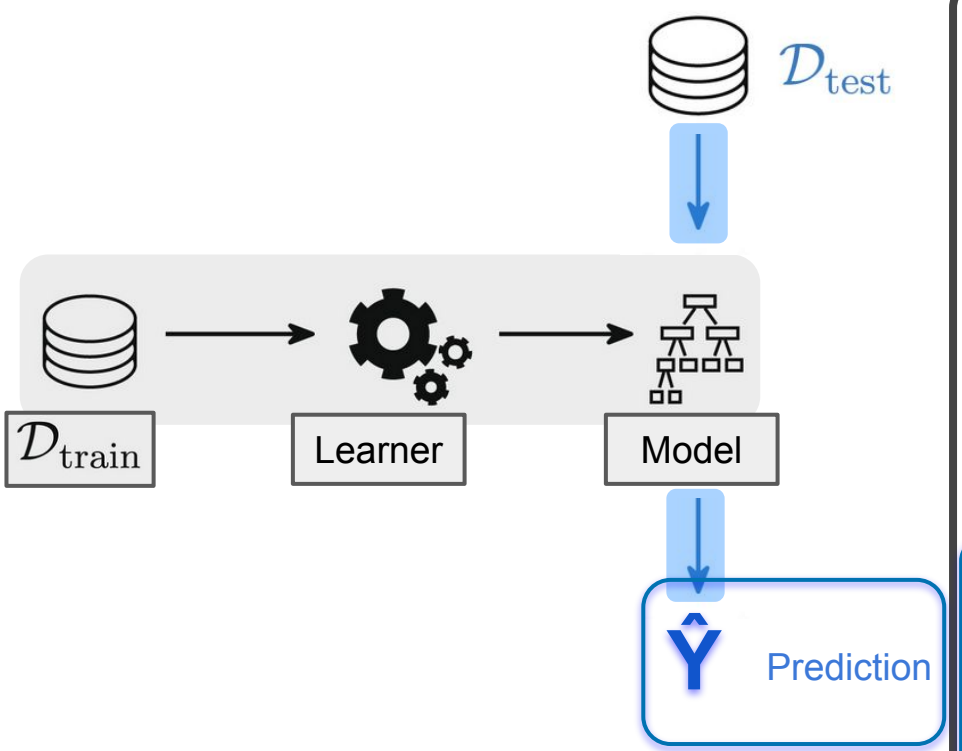

```r
data("german", package = "rchallenge")

library("mlr3")

task <- as_task_classif(
  german,
  target = "credit_risk"
)

library("mlr3learners")
learner <- lrn("classif.kknn")
```

Machine Learning




```r
data("german", package = "rchallenge")

library("mlr3")

task <- as_task_classif(
  german,
  target = "credit_risk"
)

library("mlr3learners")
learner <- lrn("classif.kknn")

splits <- partition(task)
learner$train(task, row_ids = splits$train)
```

Machine Learning

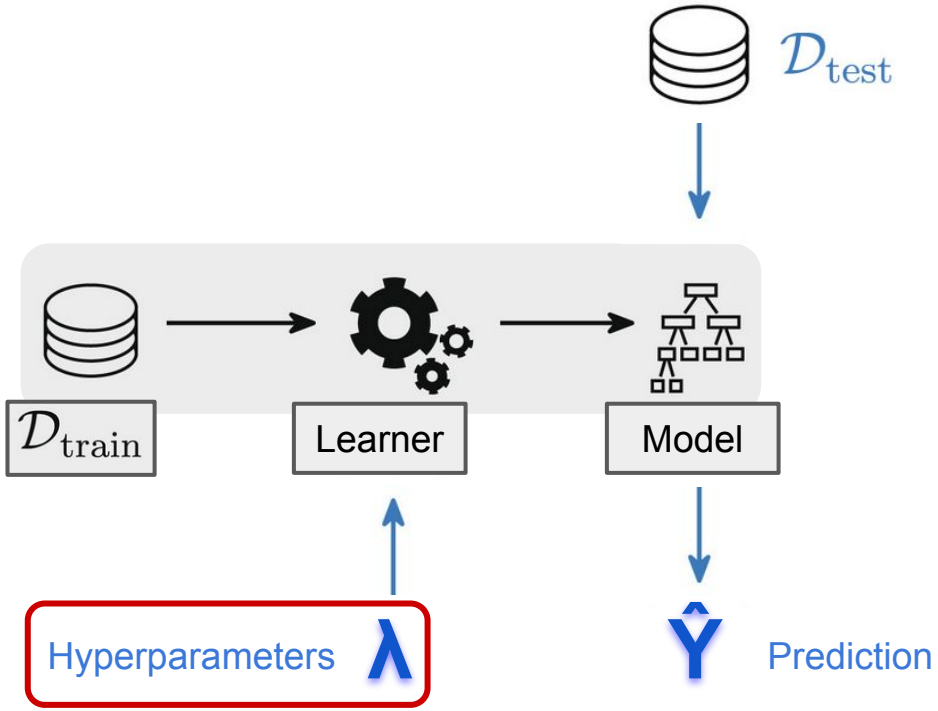



```
data("german", package = "rchallenge")

library("mlr3")

task <- as_task_classif(
  german,
  target = "credit_risk"
)

library("mlr3learners")
learner <- lrn("classif.kknn")

splits <- partition(task)
learner$train(task, row_ids = splits$train)

learner$predict(task, row_ids = splits$predict)
#> <PredictionClassif> for 1000 observations:
#>     row_ids truth response
#>           1  good     good
#>           2  good     good
#>           3  good     good
```

Machine Learning: It's Not That Easy

- Hyperparameters
- Preprocessing

Machine Learning -- Not That Easy (I)

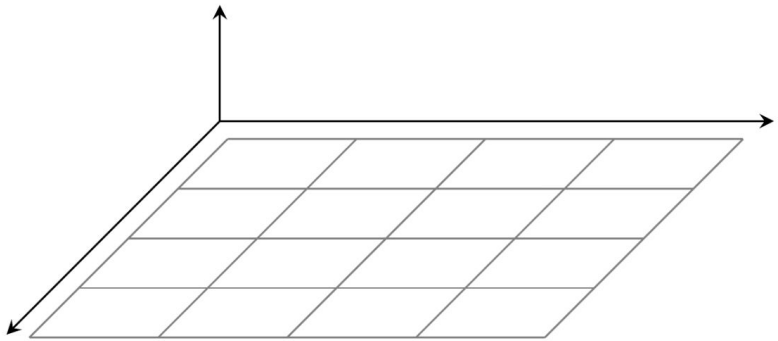


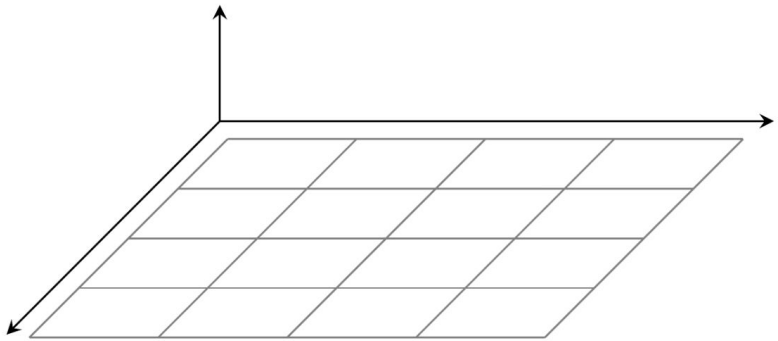

```
data("german", package = "rchallenge")

library("mlr3")

task <- as_task_classif(
  german,
  target = "credit_risk"
)

library("mlr3learners")
learner <- lrn("classif.kknn")

splits <- partition(task)
learner$train(task, row_ids = splits$train)

learner$predict(task, row_ids = splits$predict)
#> <PredictionClassif> for 1000 observations:
#>     row_ids truth response
#>           1  good     good
#>           2  good     good
#>           3  good     good
```

k = ?
other Methods?

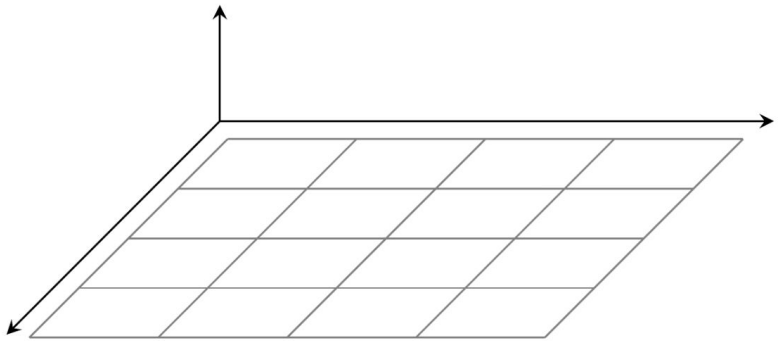
Hyperparameter Optimization

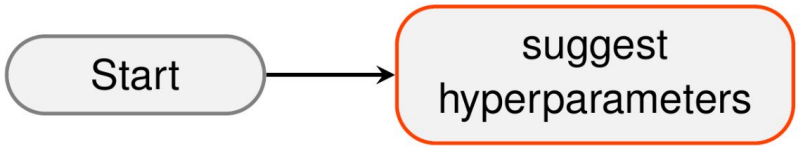
"Try out different values,
see what works best"

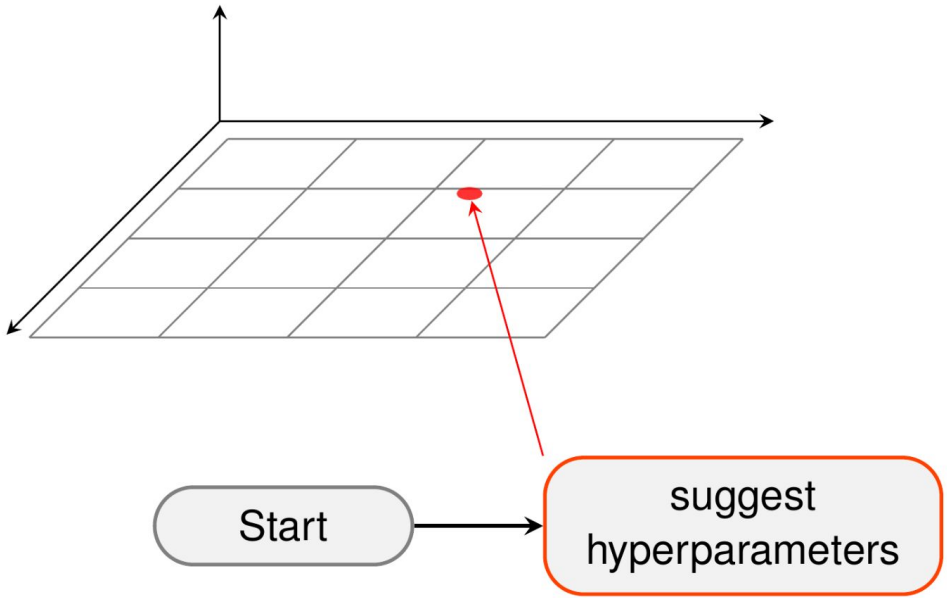


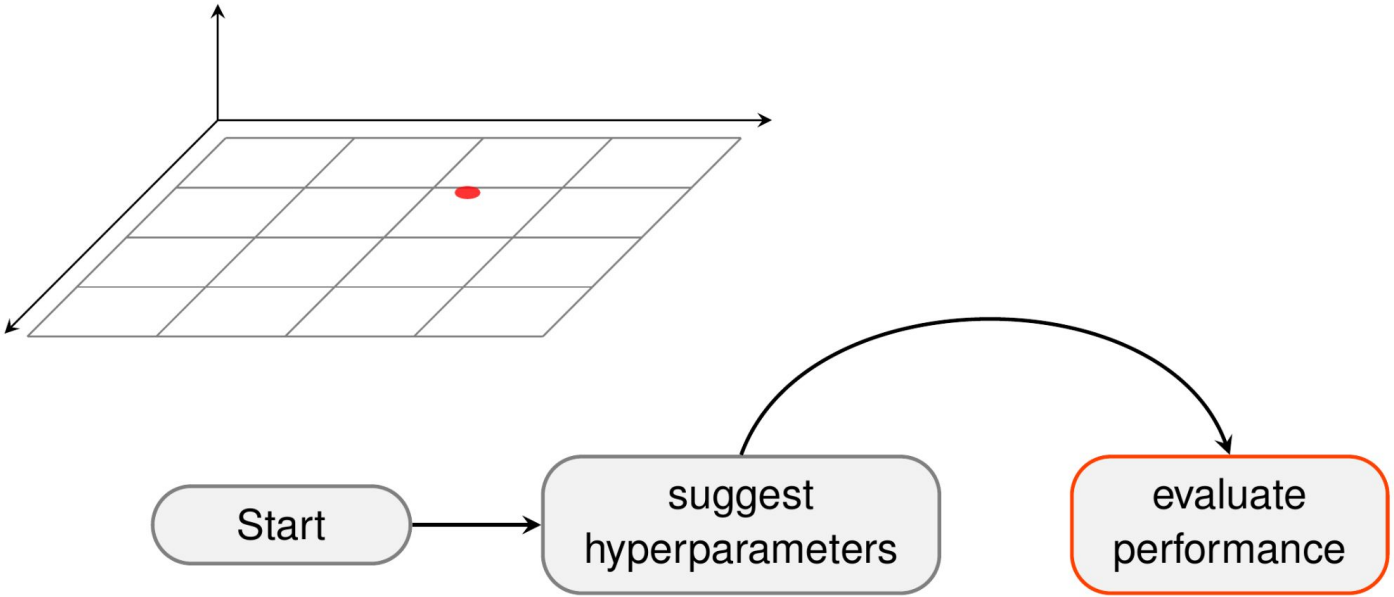


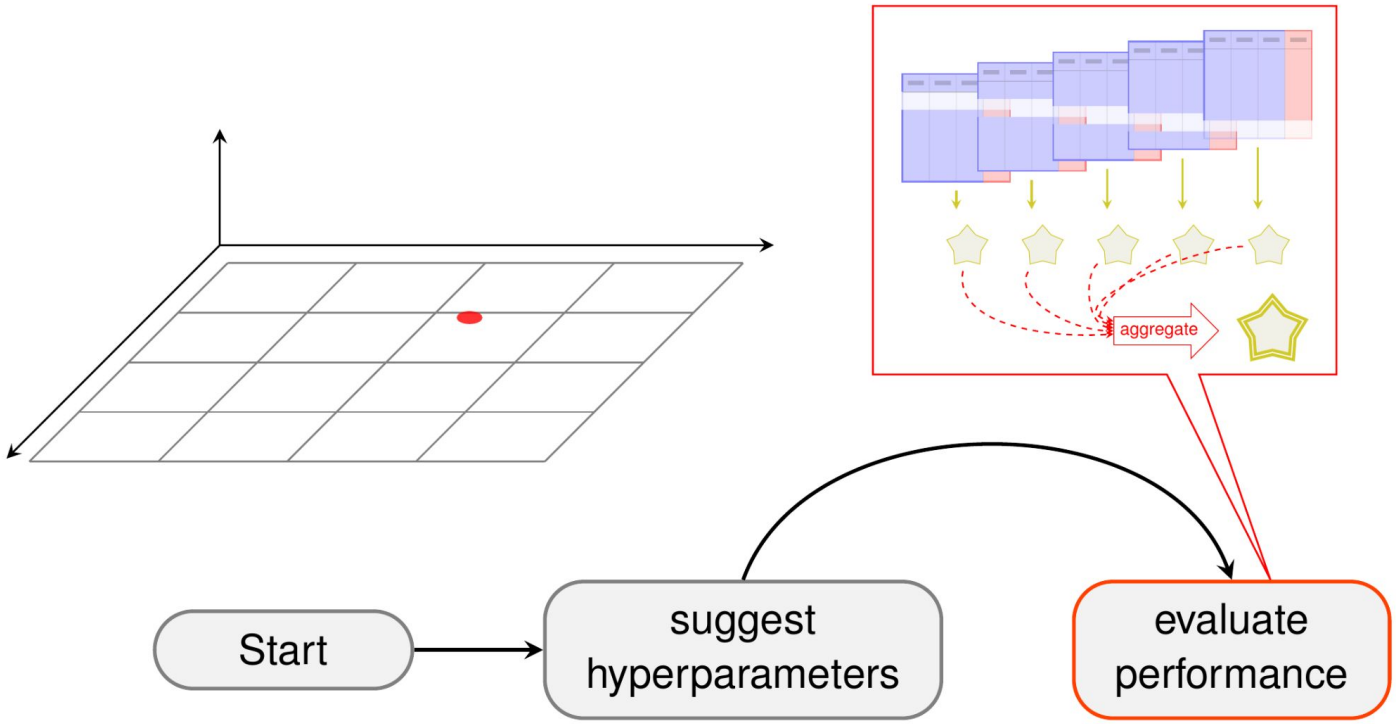
Start

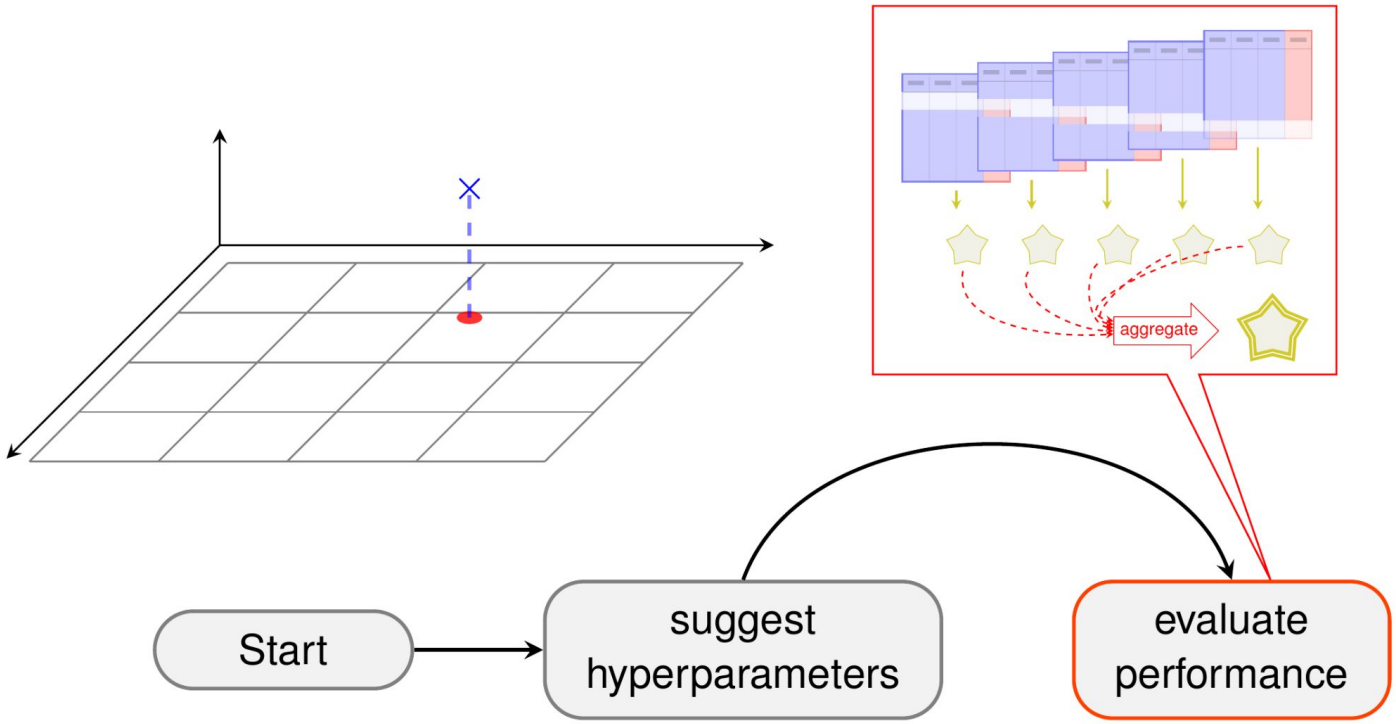


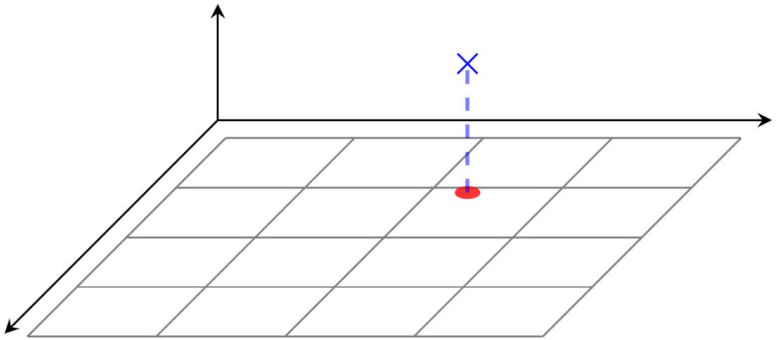

Start
suggest
hyperparameters

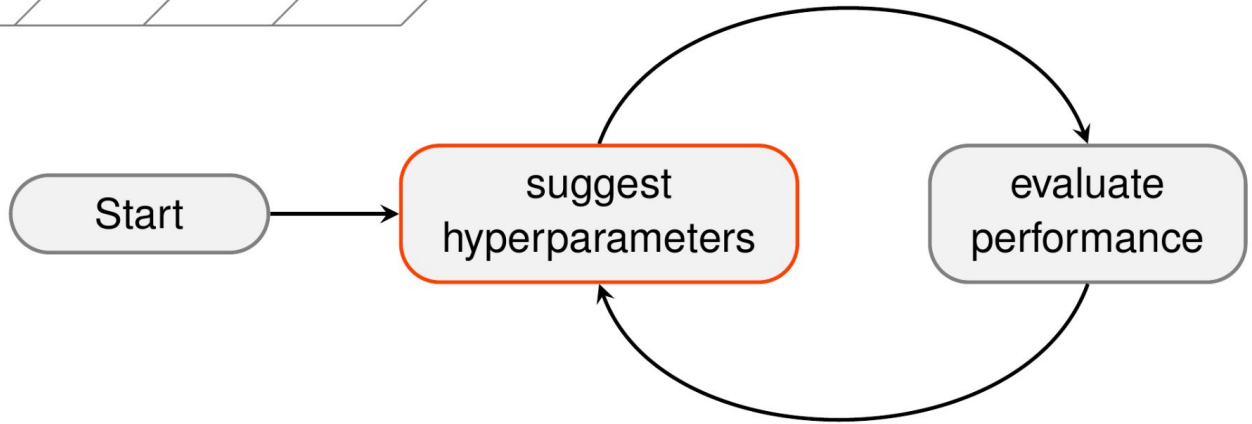

Start
suggest
hyperparameters

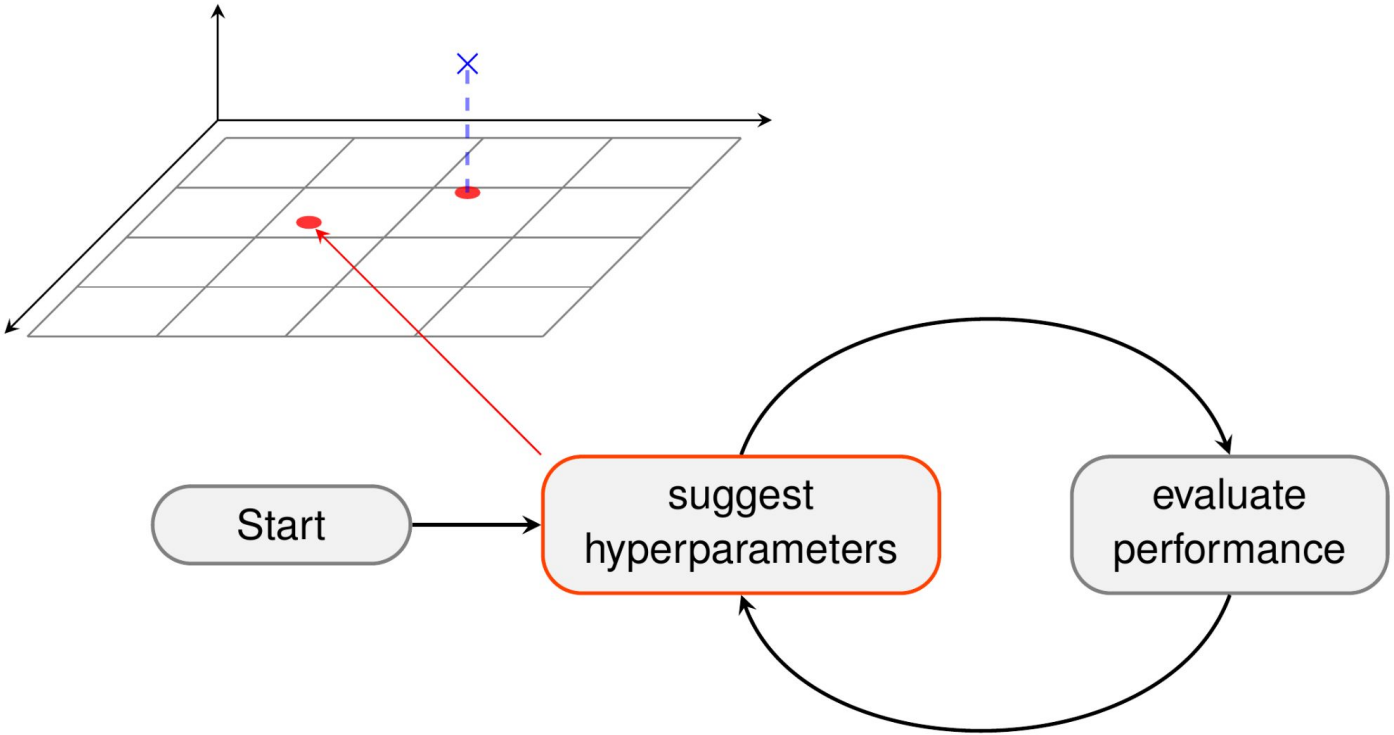

Start
suggest
hyperparameters
evaluate
performance

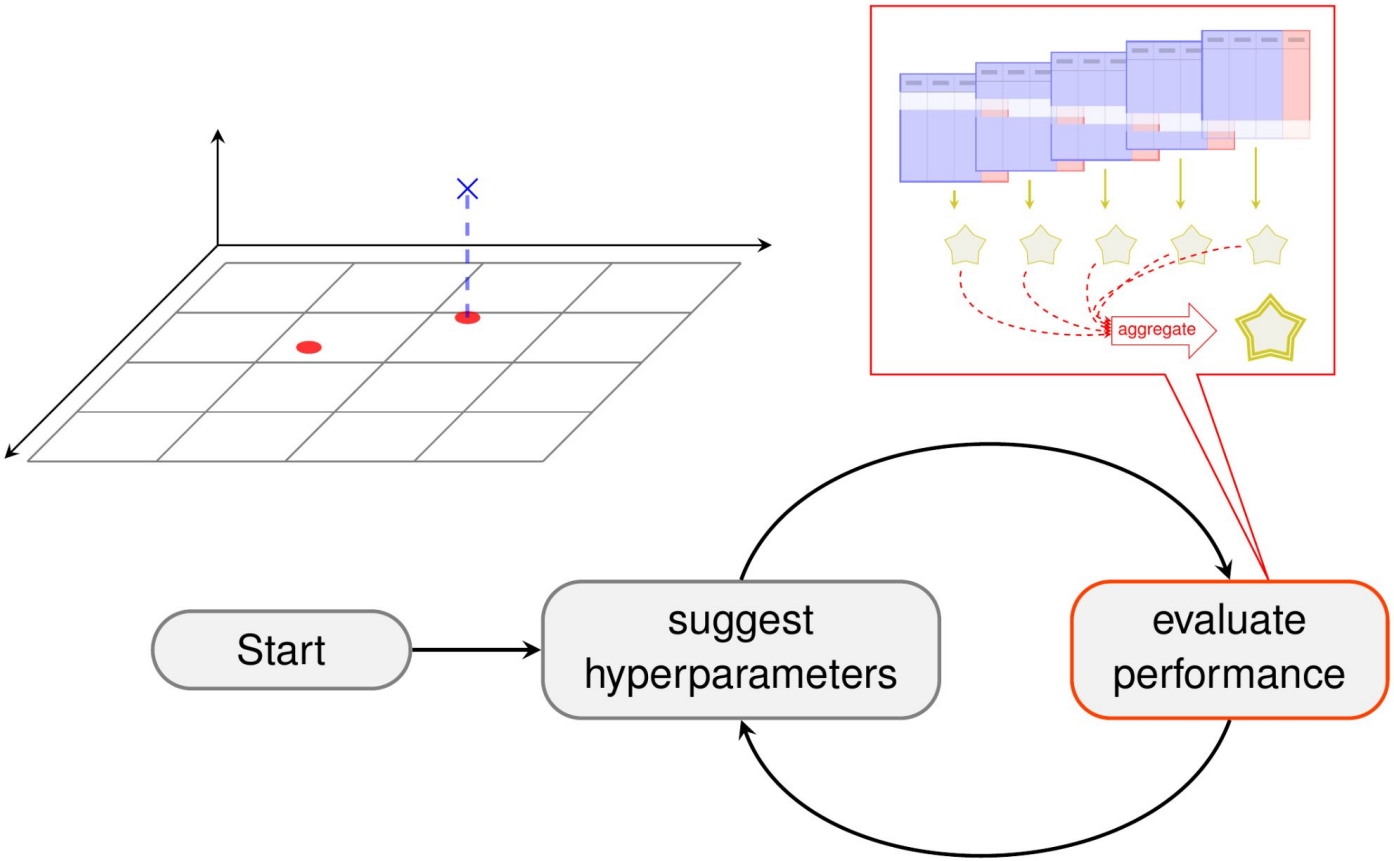

aggregate
Start
suggest
hyperparameters
evaluate
performance

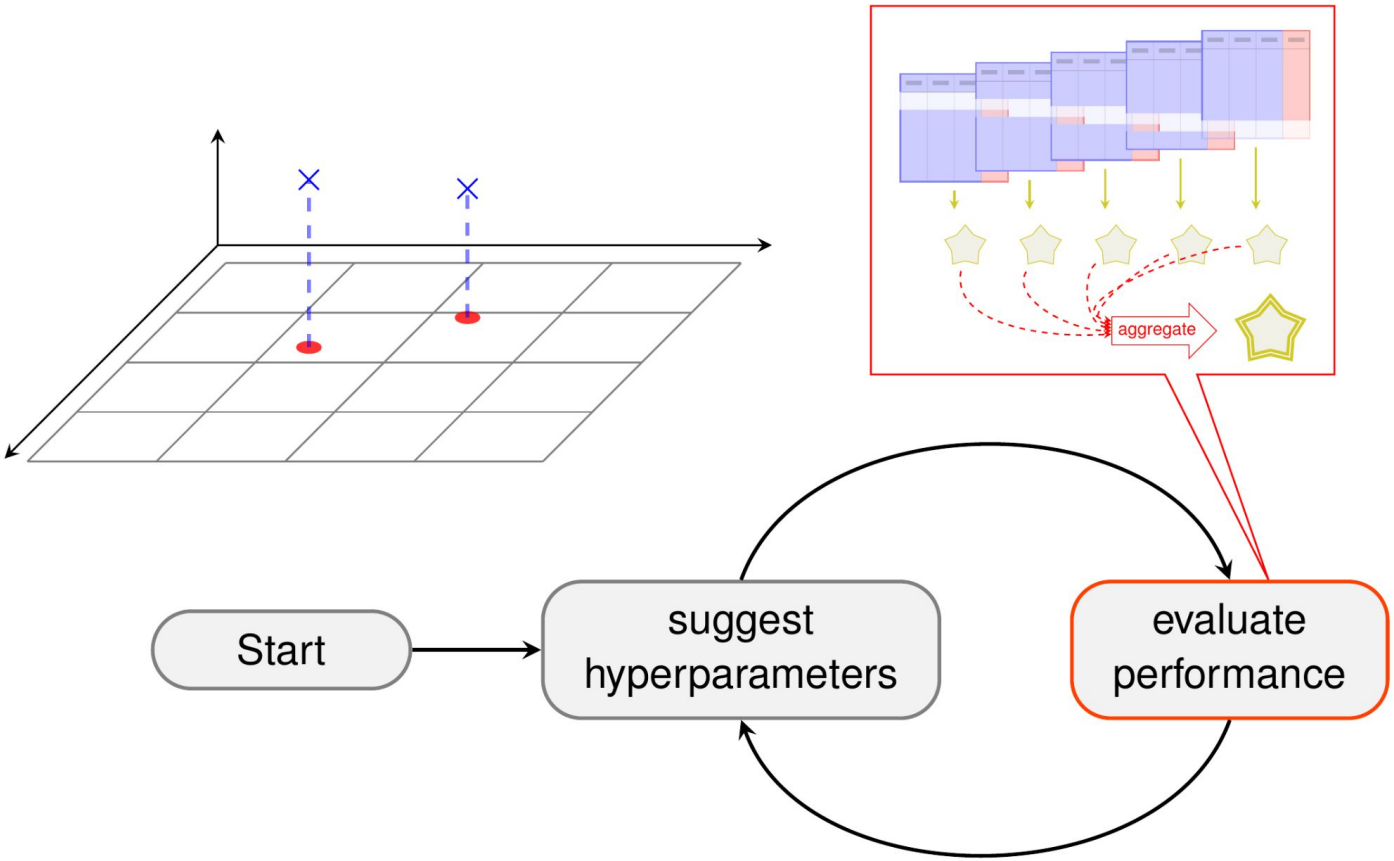

aggregate
Start
suggest
hyperparameters
evaluate
performance

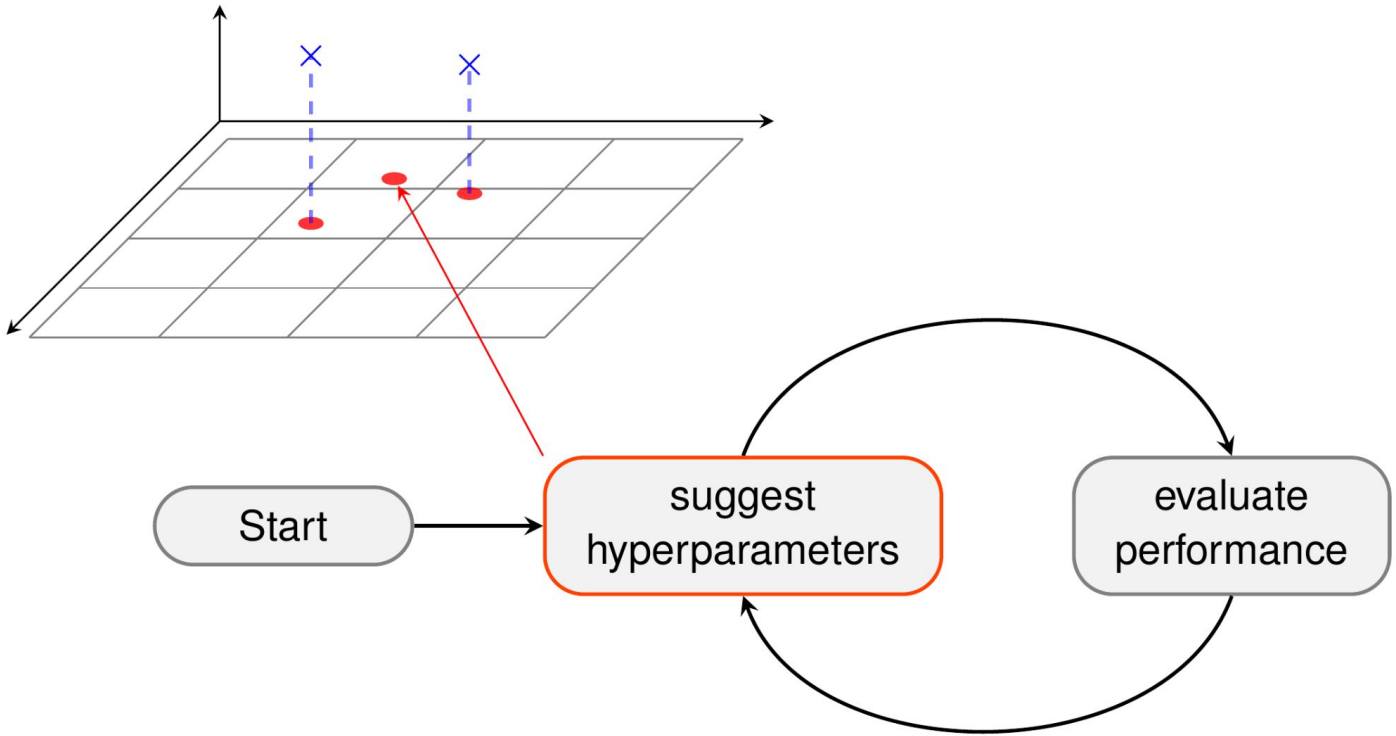


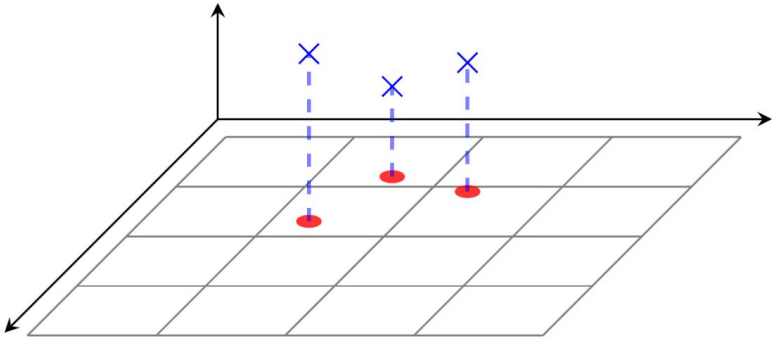

Start
suggest
hyperparameters
evaluate
performance

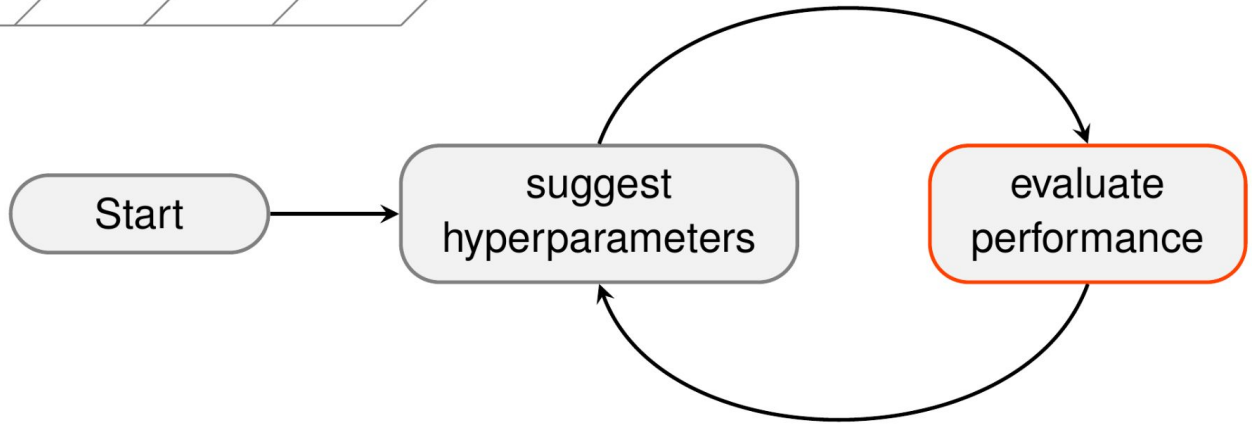

Start
suggest
hyperparameters
evaluate
performance

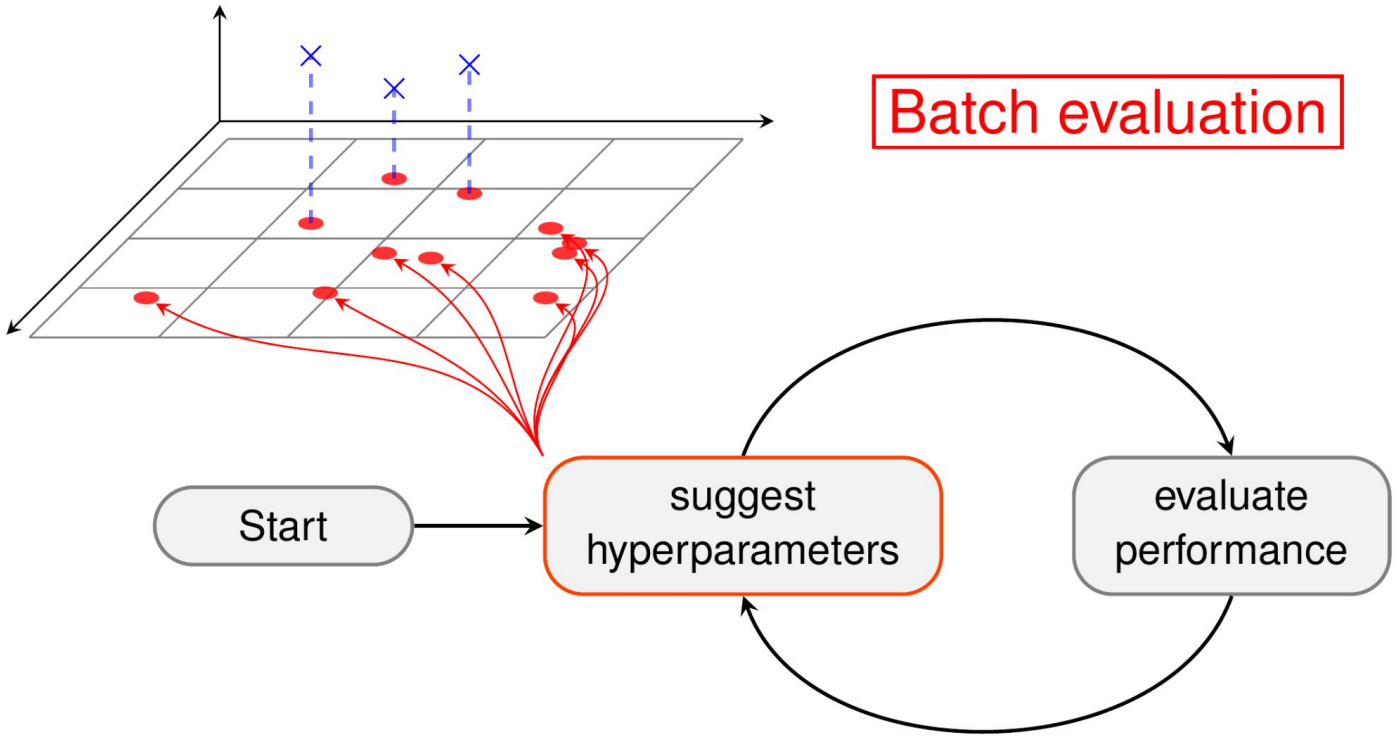

aggregate
Start
suggest
hyperparameters
evaluate
performance

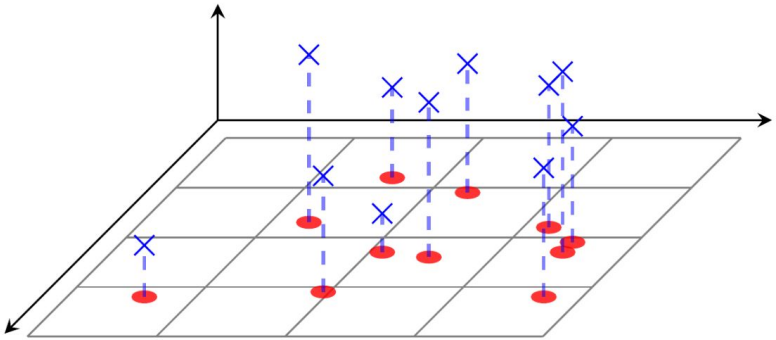

aggregate
Start
suggest
hyperparameters
evaluate
performance


Start
suggest
hyperparameters
evaluate
performance

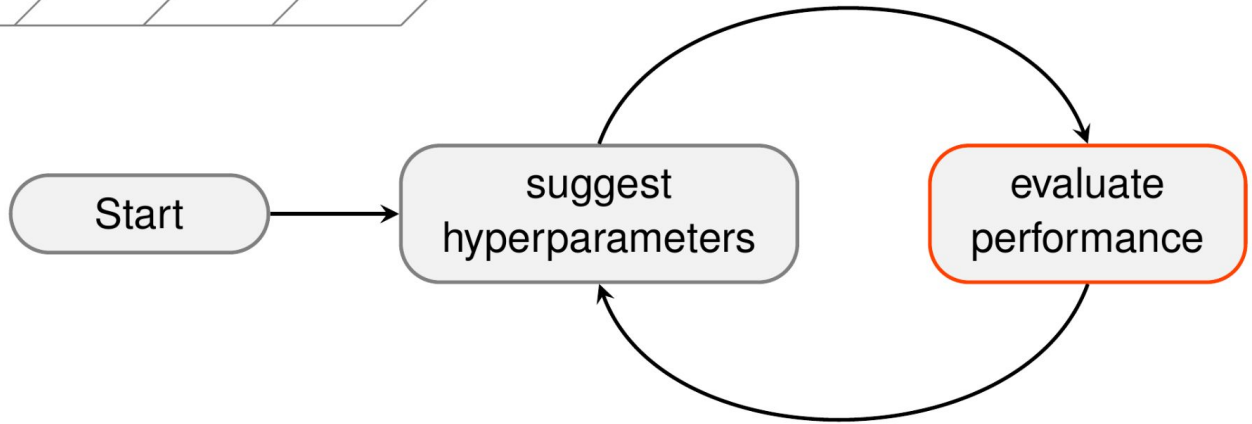


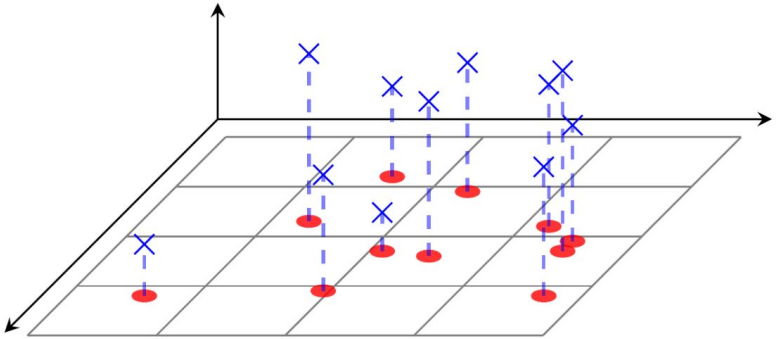

Start
suggest
hyperparameters
evaluate
performance

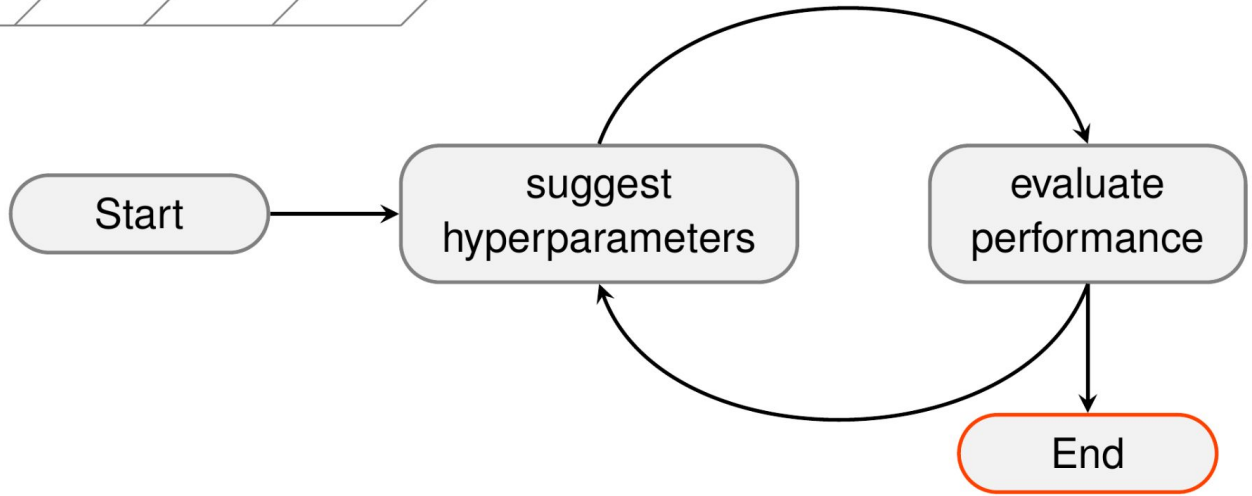

Batch evaluation
Start
suggest hyperparameters
evaluate performance

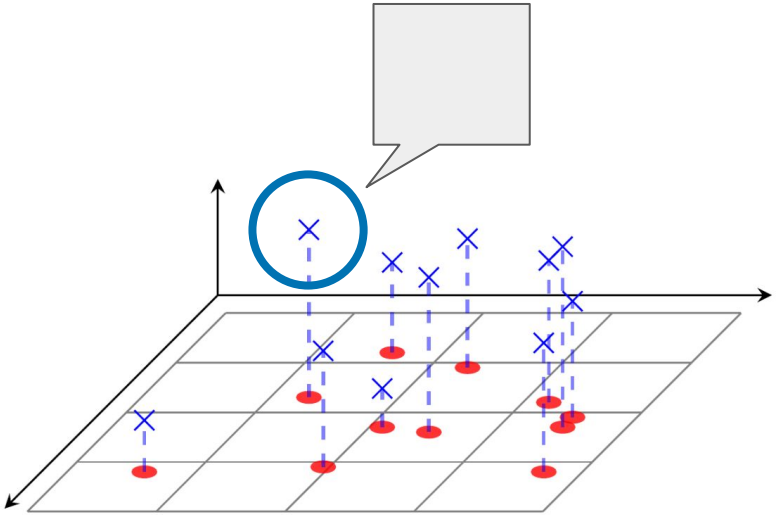


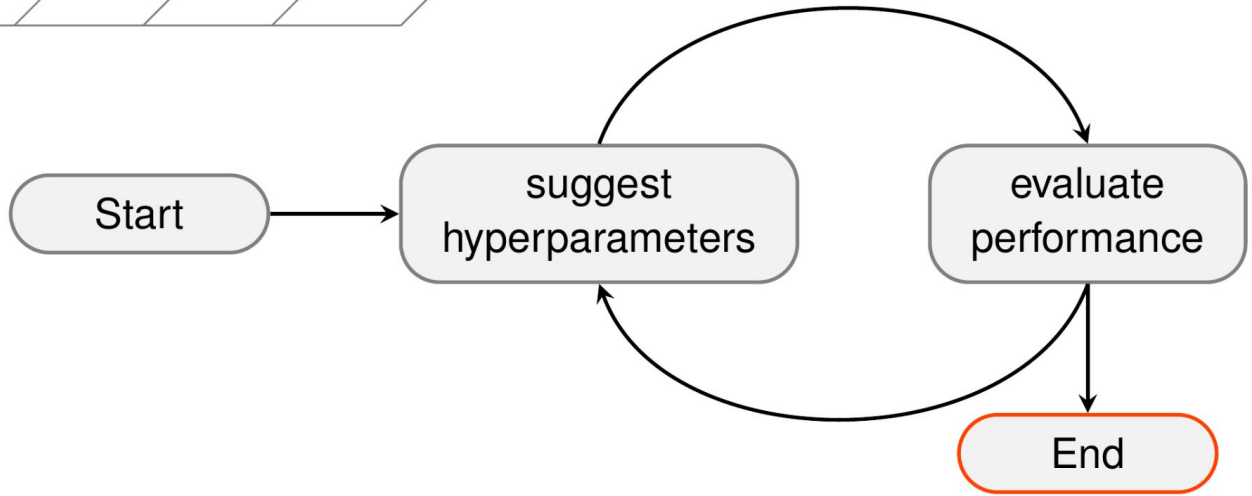
Batch evaluation






Start
suggest
hyperparameters
evaluate
performance
End




Start
suggest
hyperparameters
evaluate
performance
End

Tuning in mlr3

```
library("mlr3tuning")
lgr::get_logger("mlr3")$set_threshold("warn")
lgr::get_logger("bbotk")$set_threshold("warn")
```

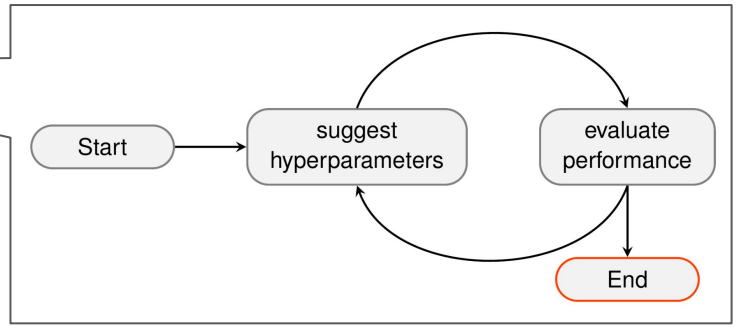
Tuning in mlr3

```
library("mlr3tuning")
lgr::get_logger("mlr3")$set_threshold("warn")
lgr::get_logger("bbotk")$set_threshold("warn")

learner$param_set$values$k <- to_tune(2, 100, logscale = TRUE)
```

Tuning in mlr3

```
library("mlr3tuning")
lgr::get_logger("mlr3")$set_threshold("warn")
lgr::get_logger("bbotk")$set_threshold("warn")

learner$param_set$values$k <- to_tune(2, 100, logscale = TRUE)
```

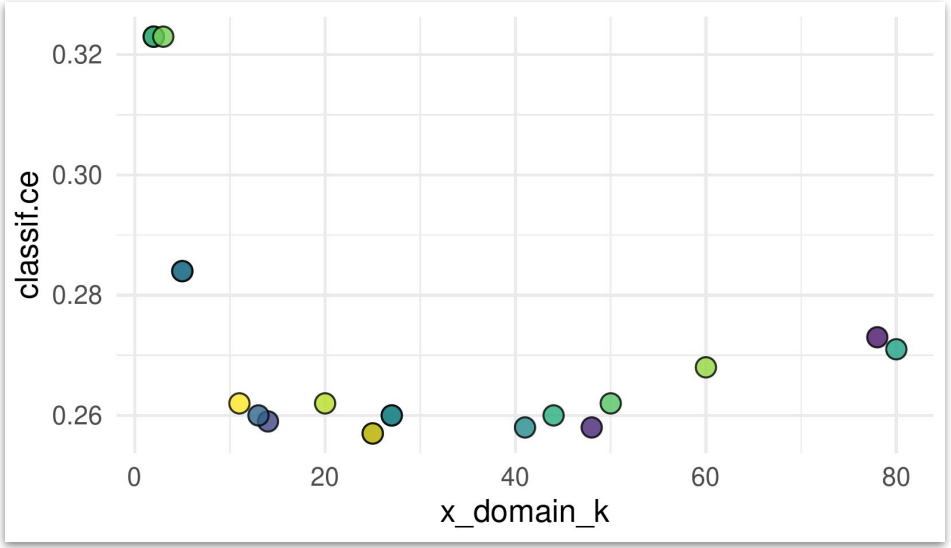
```
ti <- tune(
  tuner = tnr("random_search"),
  task = task,
  learner = learner,
  resampling = rsmp("cv", folds = 10),
  measures = msr("classif.ce"),
  term_evals = 20
)
```



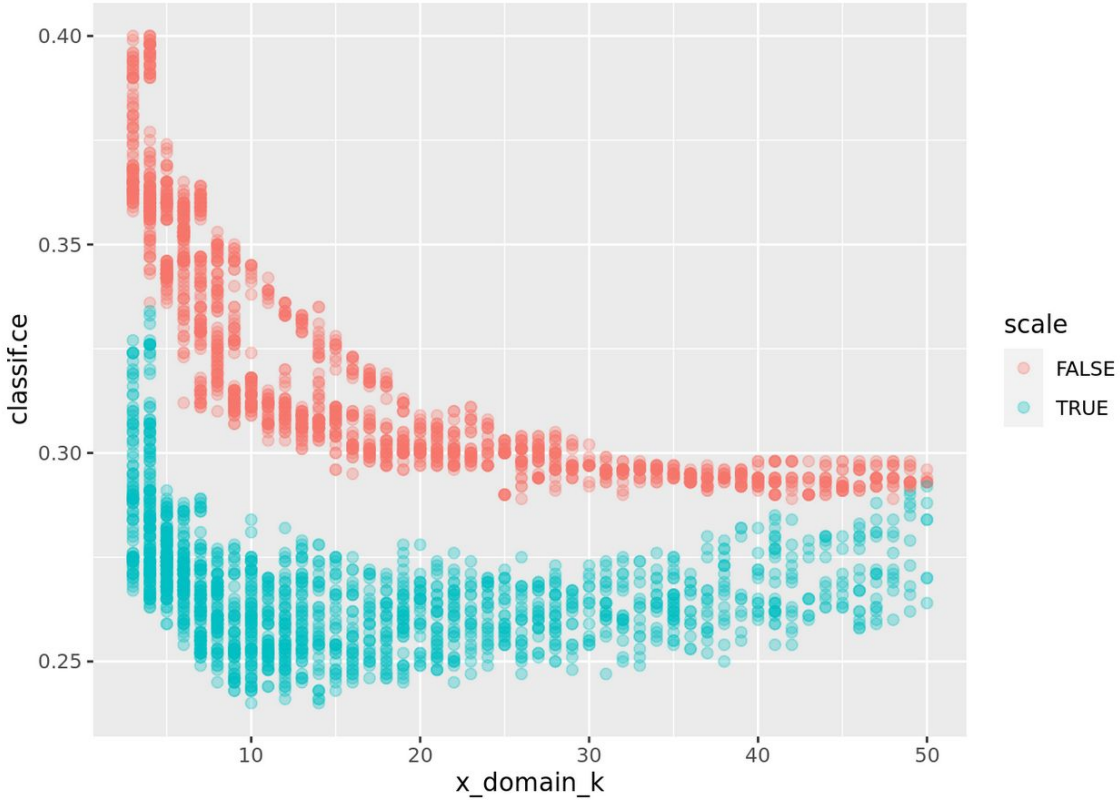
Tuning in mlr3

```
#> <TuningInstanceSingleCrit>
#> * State:  Optimized
#> * Objective: <ObjectiveTuning:classif.kknn_on_german>
#> * Search Space:
#>    id    class     lower    upper nlevels
#> 1:  k ParamDbl 0.6931472 4.615121     Inf
#> * Terminator: <TerminatorEvals>
#> * Result:
#>          k classif.ce
#> 1: 3.252434      0.257
#> * Archive:
#>           k classif.ce
#>  1: 3.252434      0.257
#>  2: 4.363433      0.273
#>  3: 3.888727      0.258
```

```
library("mlr3viz")
autoplot(ti, trafo = TRUE)
```



Tuning in mlr3

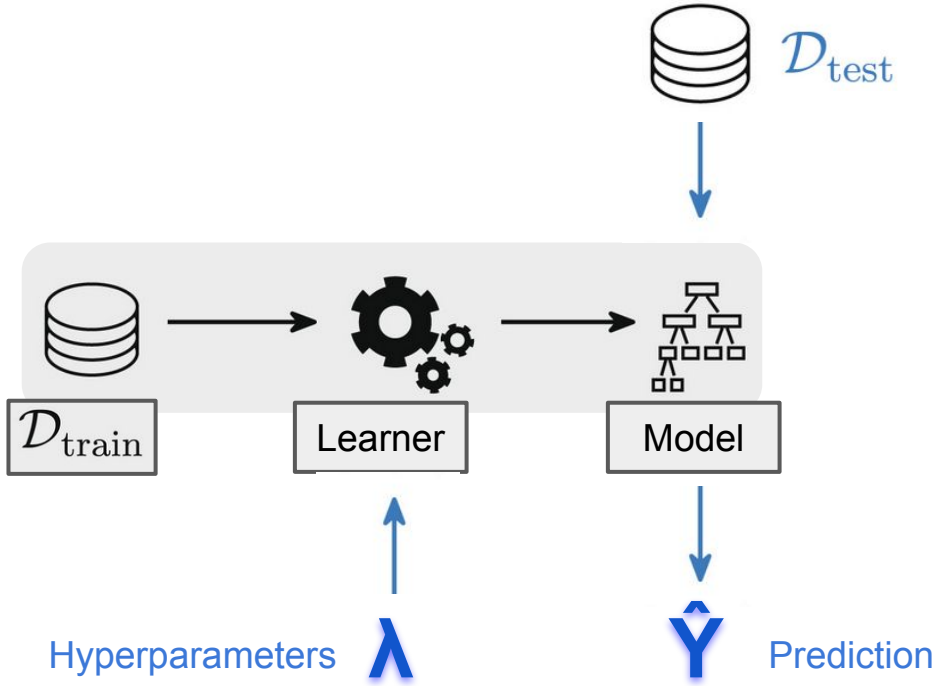


Tuning in mlr3

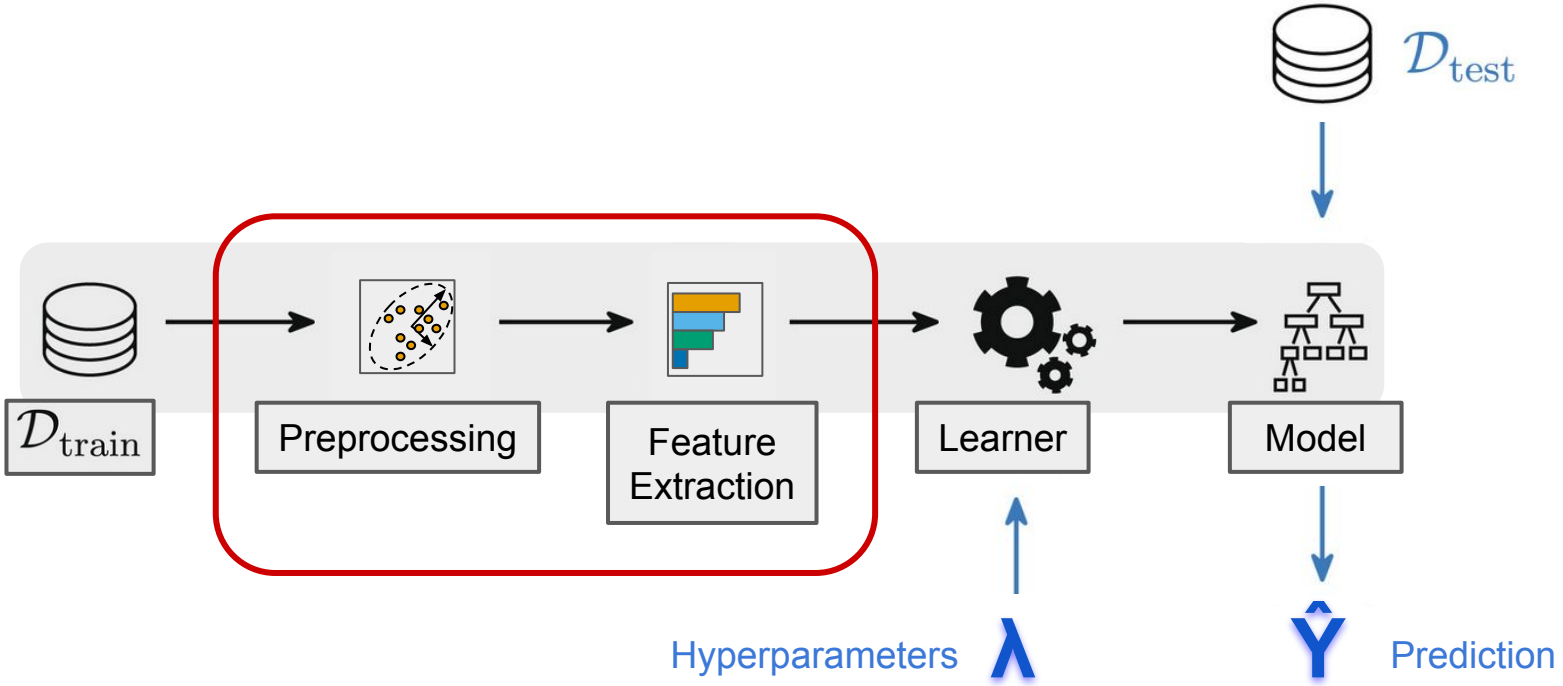
- Optimization algorithms are just objects, accessed through `tnr()`
- Common search spaces in the `mlr3tuningspaces` package
- Nested resampling for unbiased performance evaluation using the `AutoTuner`
- Transparent parallelization using the `future` package

Machine Learning:
It's Not That Easy

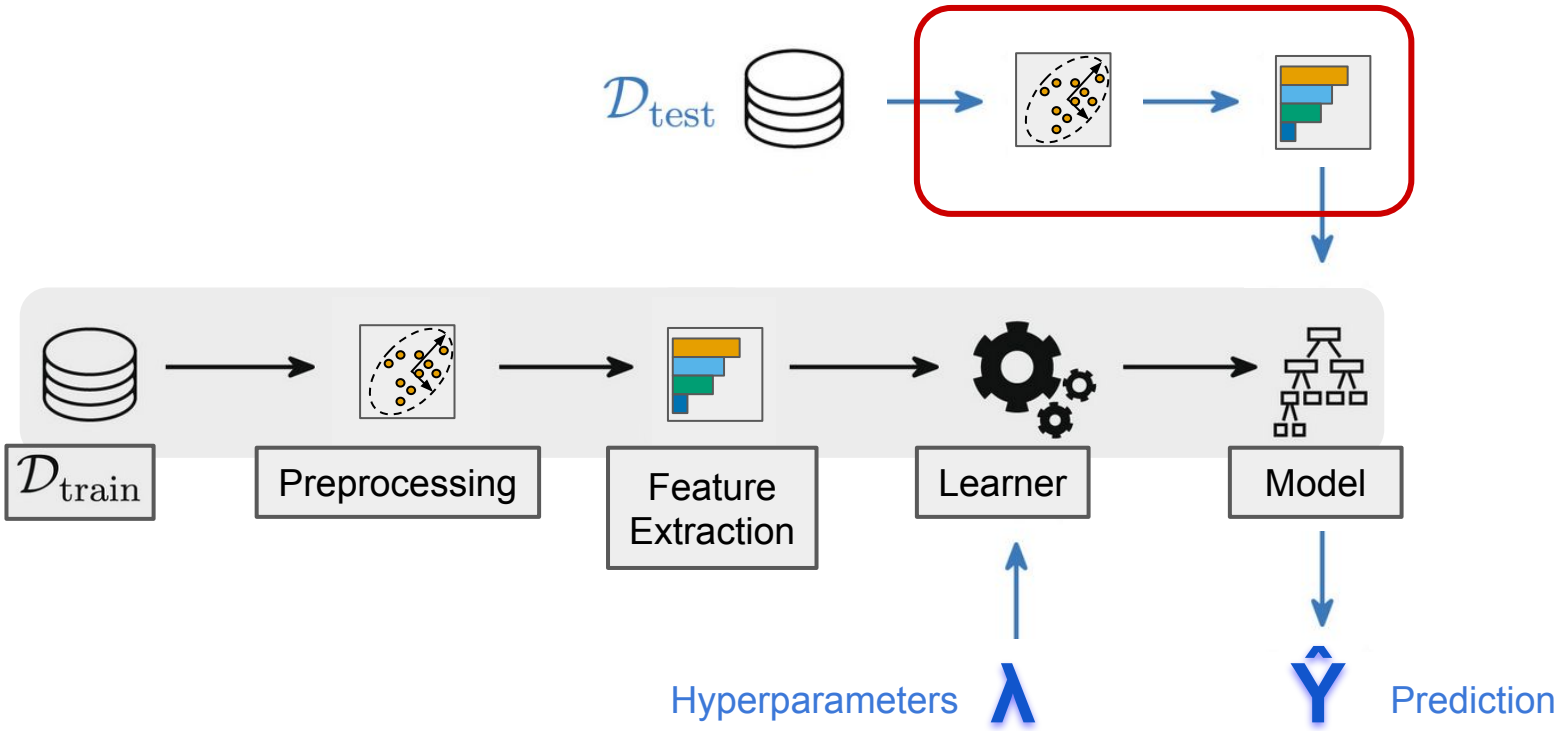
Machine Learning -- Not That Easy (II)



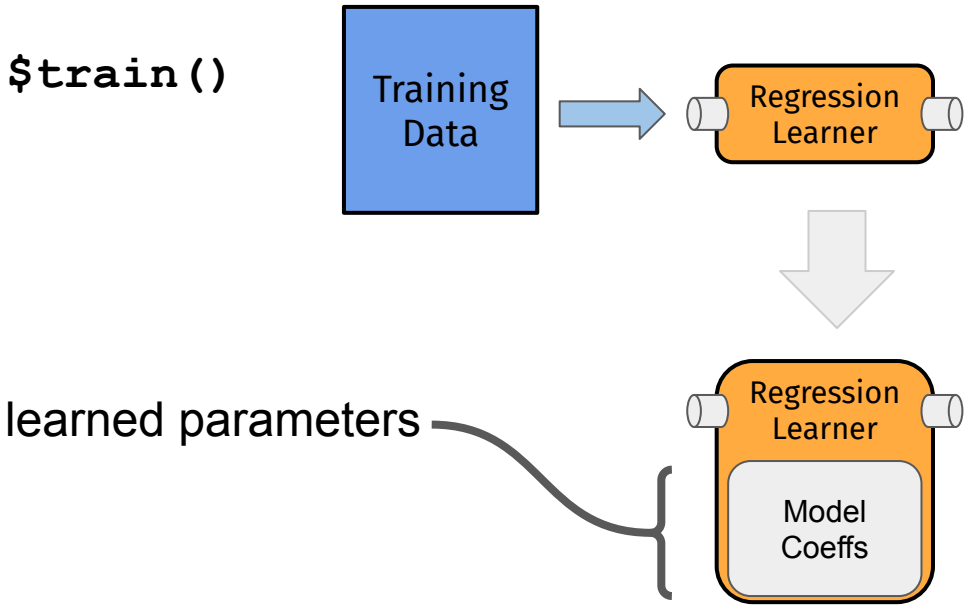
Machine Learning -- Not That Easy (II)



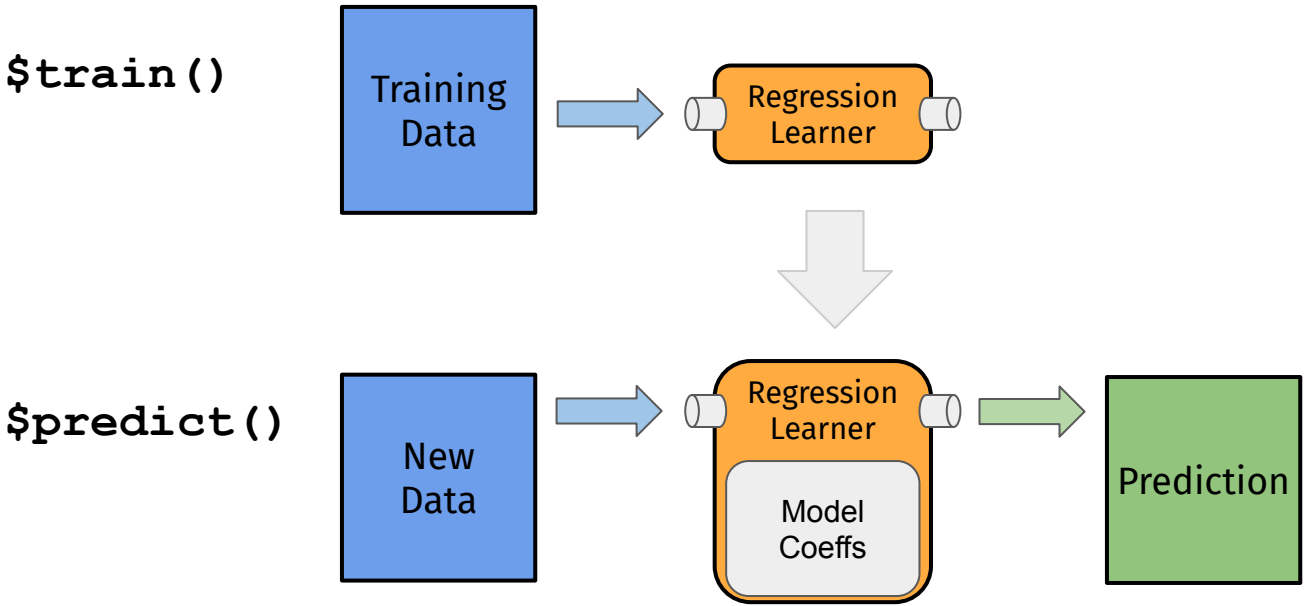
Machine Learning -- Not That Easy (II)



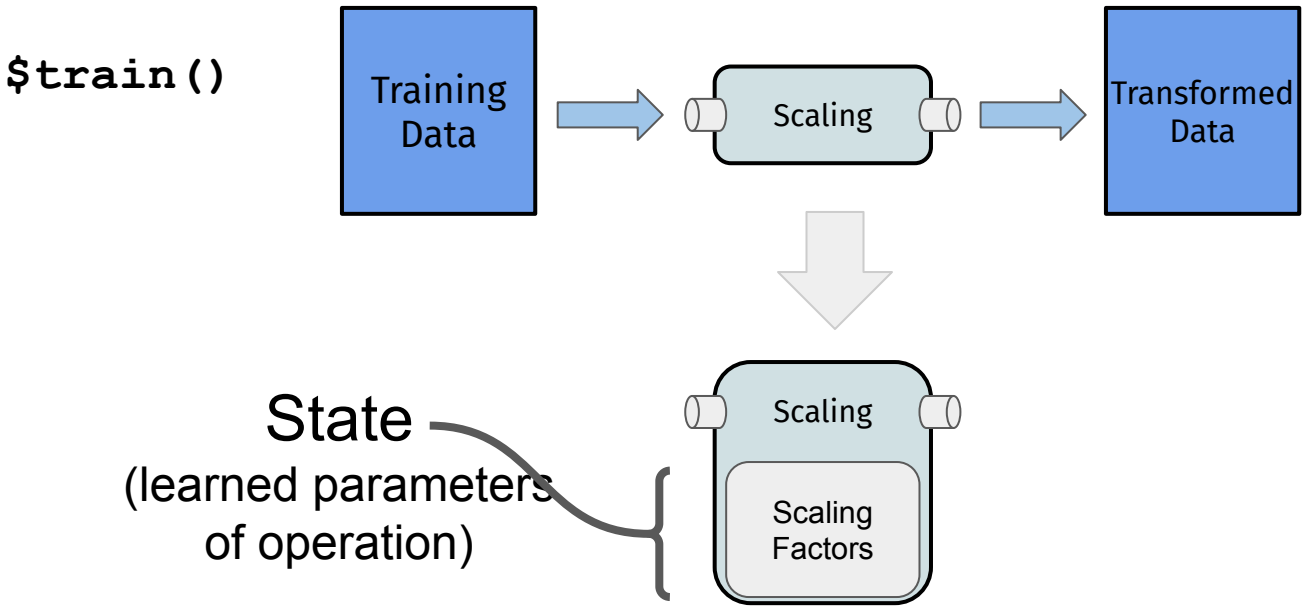
From Learner to Pipeline



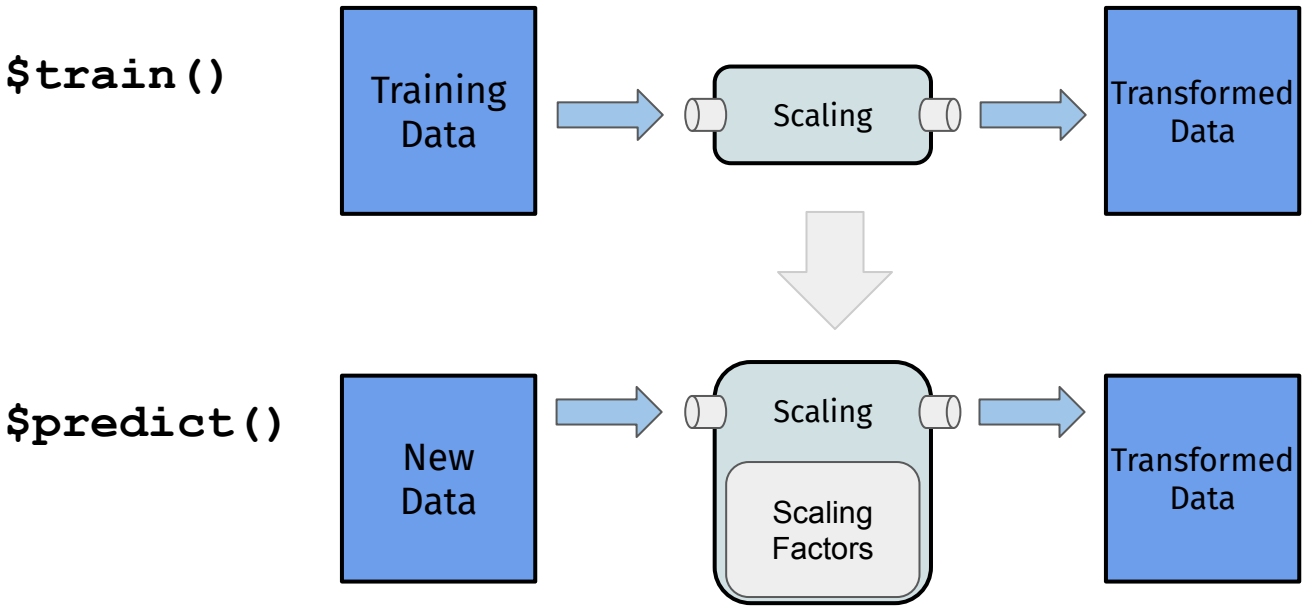
From Learner to Pipeline



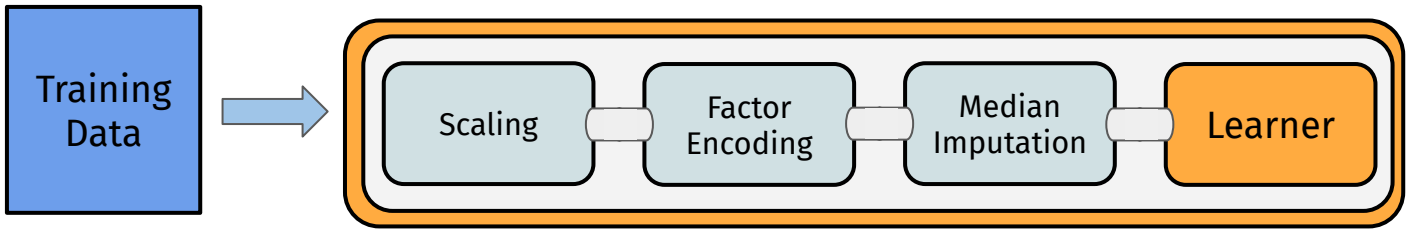
From Learner to Pipeline



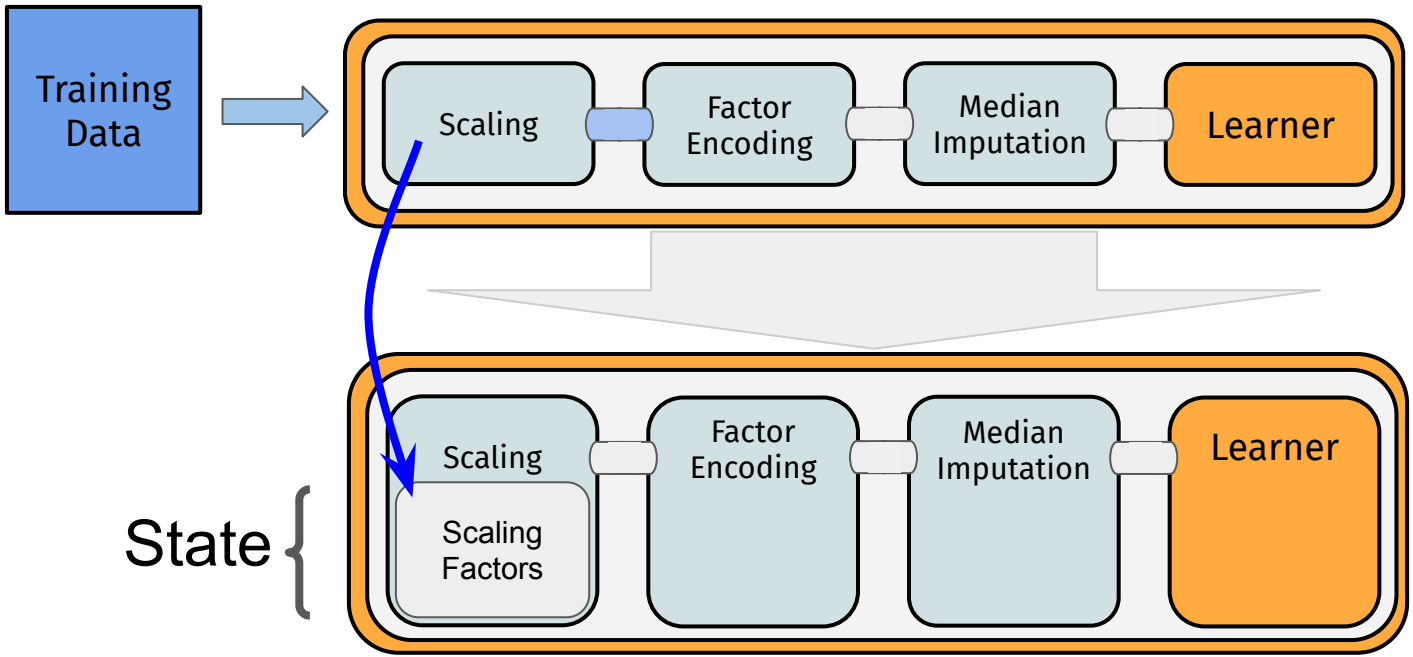
From Learner to Pipeline



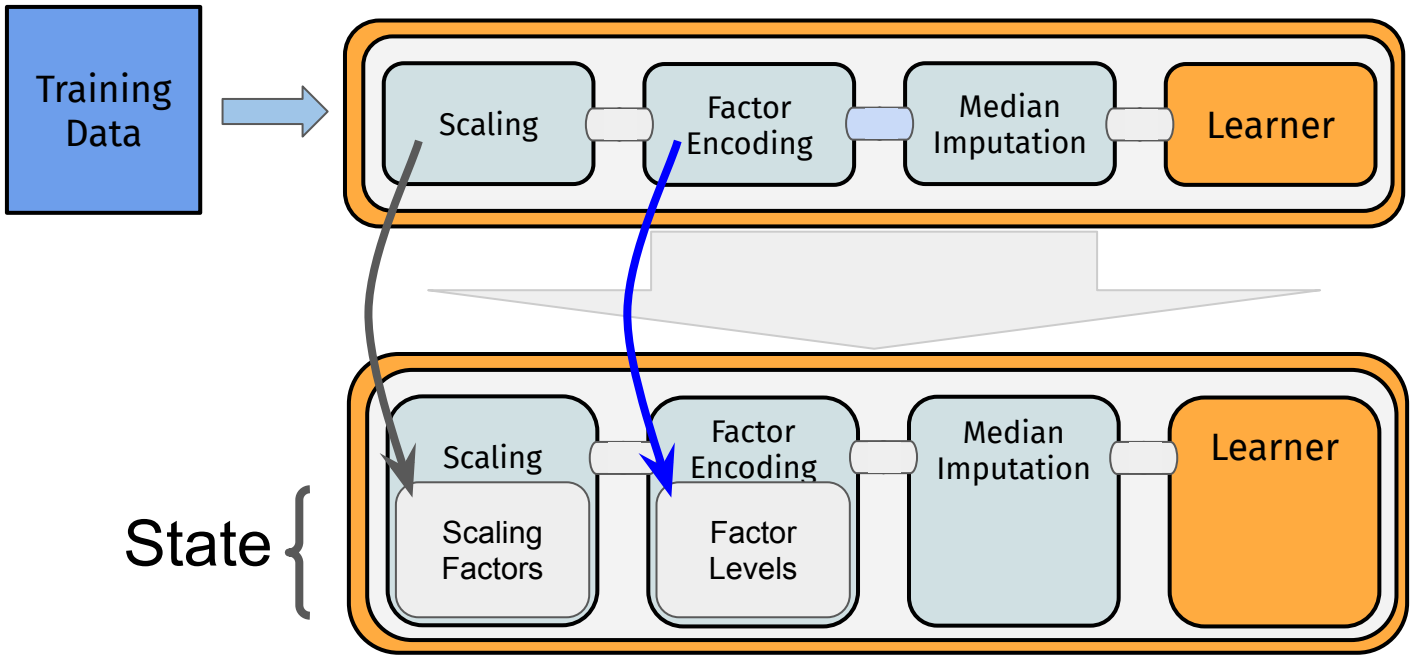
From Learner to Pipeline



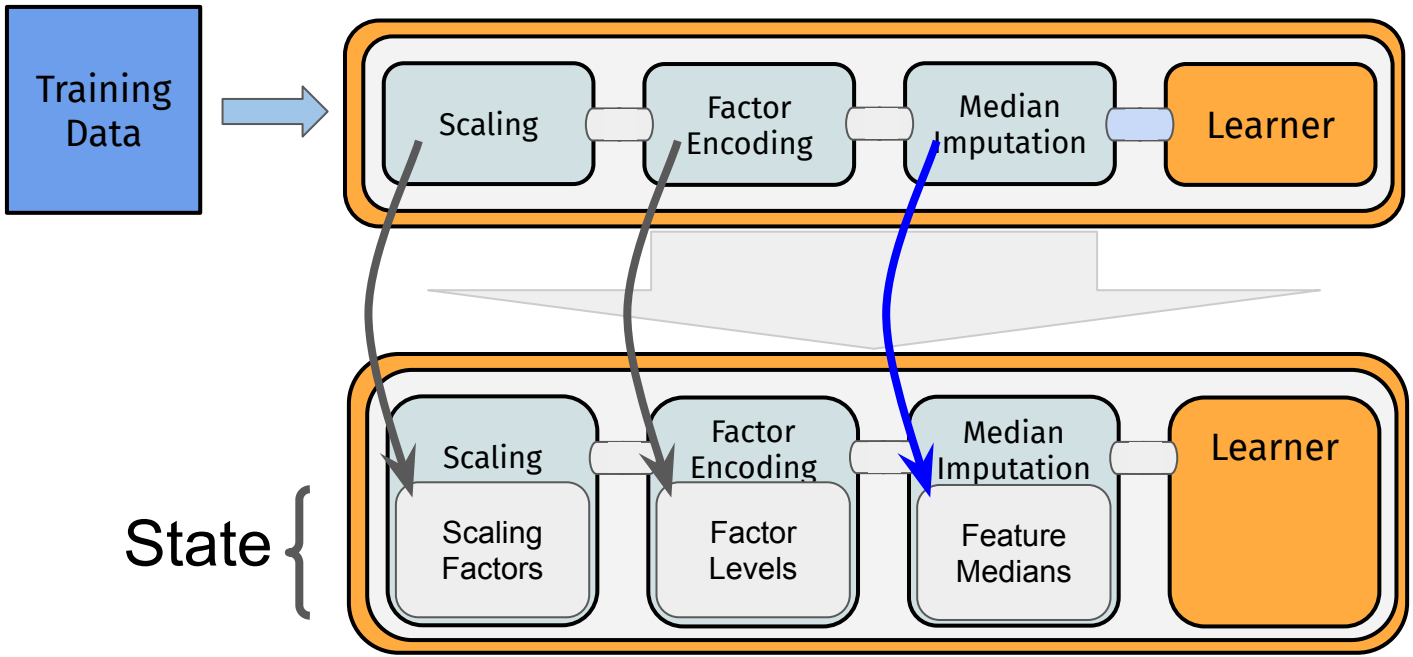
From Learner to Pipeline



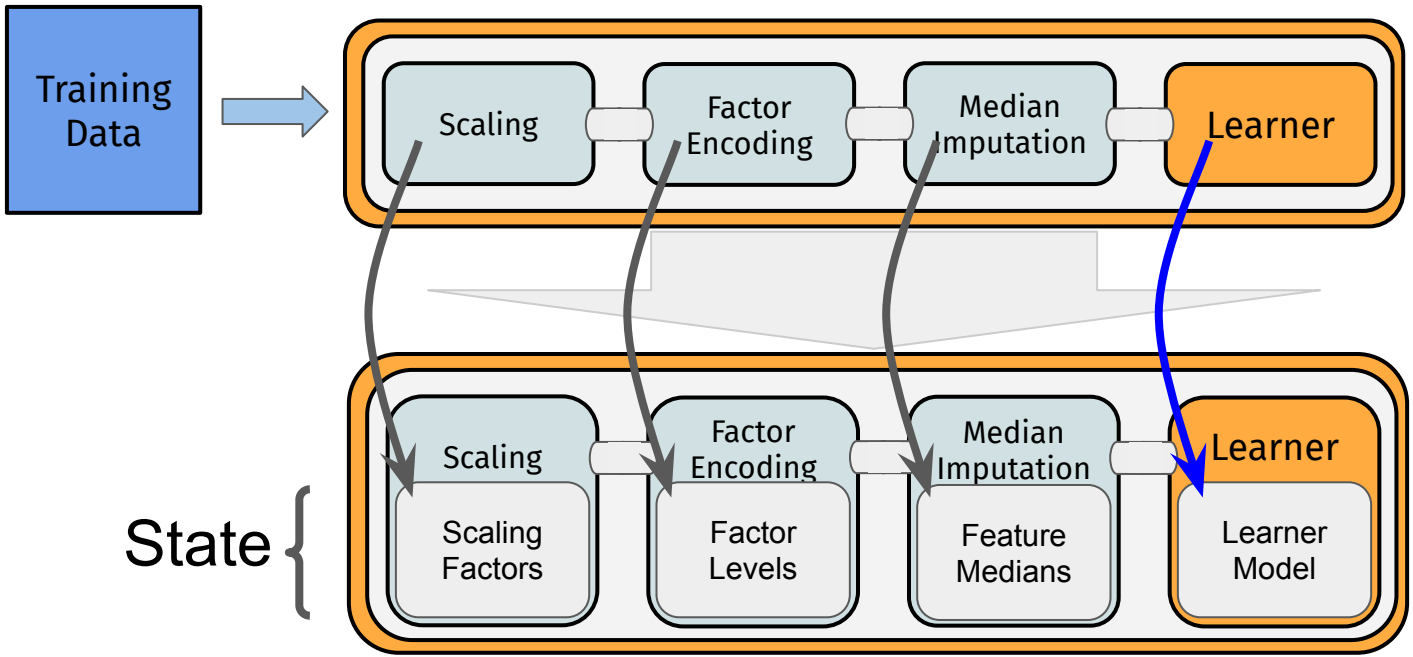
From Learner to Pipeline



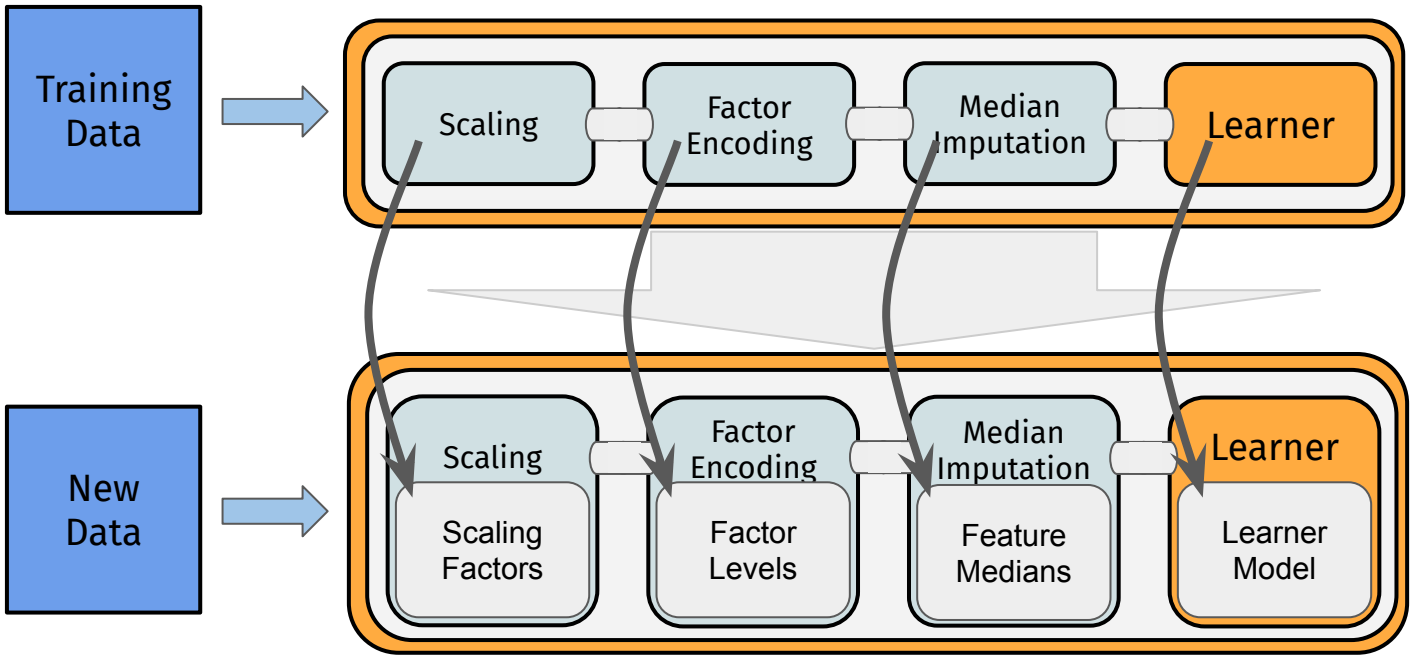
From Learner to Pipeline



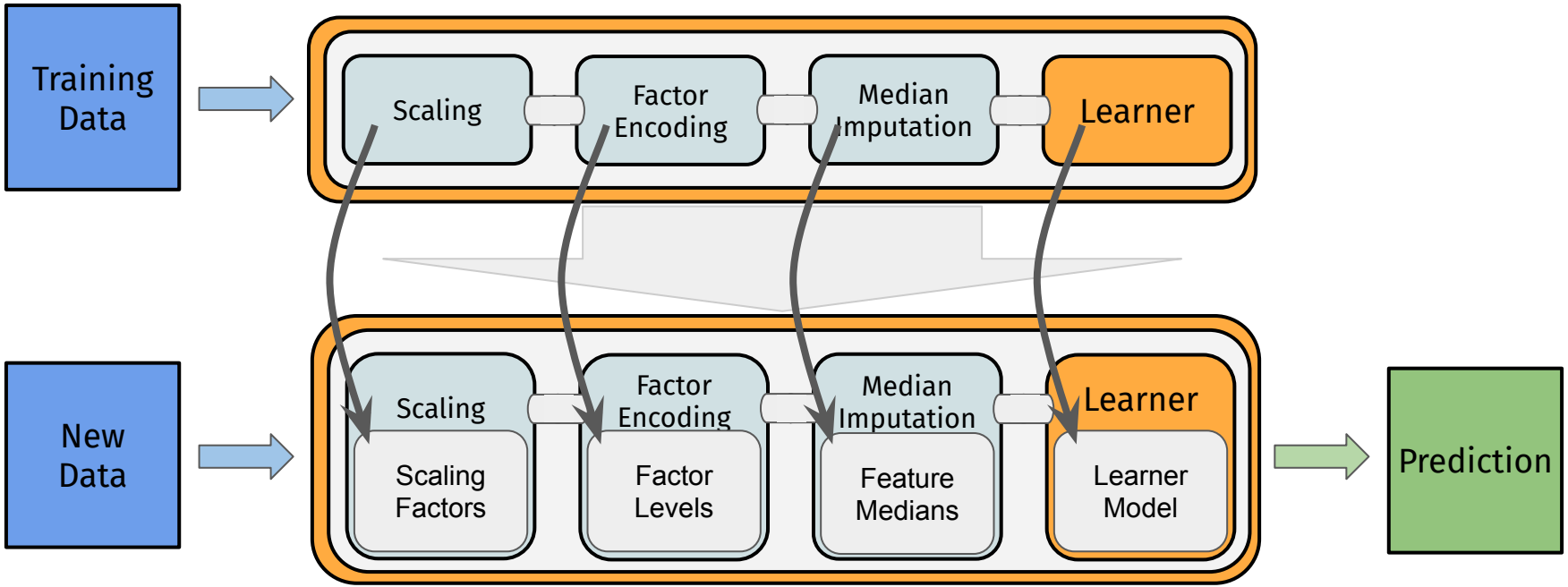
From Learner to Pipeline



From Learner to Pipeline



From Learner to Pipeline



Pipelines in mlr3: mlr3pipelines

```
library("mlr3pipelines")
```

Pipelines in mlr3: mlr3pipelines

```
library("mlr3pipelines")
library("mlr3filters")
pipeline <- as_learner(
  po("filter", flt("mim"), filter.nfeat = 3) %>>%
  lrn("classif.log_reg")
)
```

Pipelines in mlr3: mlr3pipelines

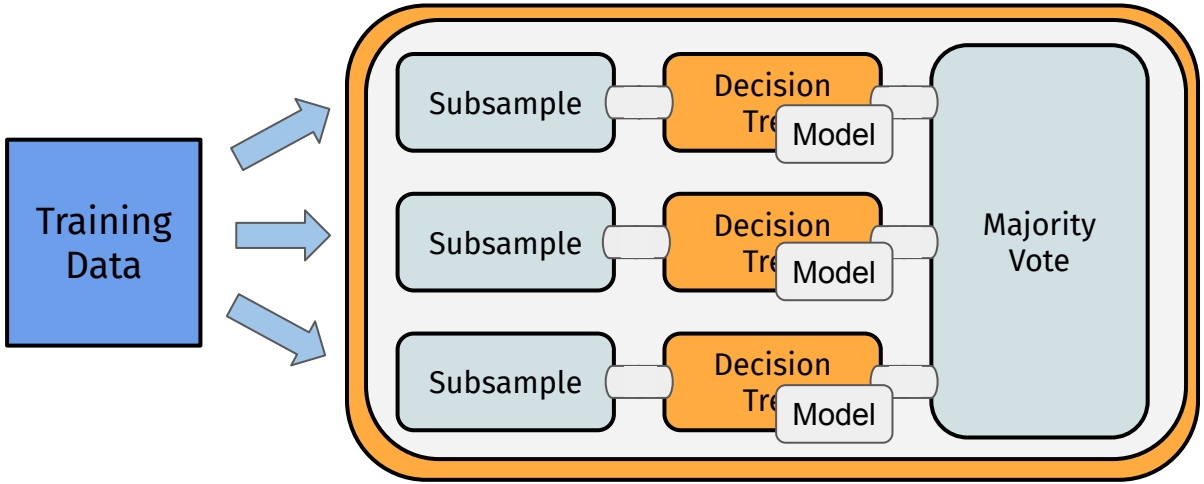
```
library("mlr3pipelines")
library("mlr3filters")
pipeline <- as_learner(
  po("filter", flt("mim"), filter.nfeat = 3) %>>%
  lrn("classif.log_reg")
)

pipeline$train(task)
```

Pipelines in mlr3: mlr3pipelines

```
library("mlr3pipelines")
library("mlr3filters")
pipeline <- as_learner(
  po("filter", flt("mim"), filter.nfeat = 3) %>>%
  lrn("classif.log_reg")
)

pipeline$train(task)

pipeline$base_learner()$model
#> Call:  stats::glm(formula = task$formula(), family = "binomial", data = data,
#>     model = FALSE)

#> Coefficients:
#>                                     (Intercept)
#>                                       -8.237e-01
#>                                          amount
#>                                       2.705e-05
#>                                        duration
#>                                       3.342e-02
#>                                 status... < 0 DM
#>                                       -5.020e-01
#>                           status0<= ... < 200 DM
#>                                       -1.100e+00
#> status... >= 200 DM / salary for at least 1 year
#>                                       -2.025e+00
```

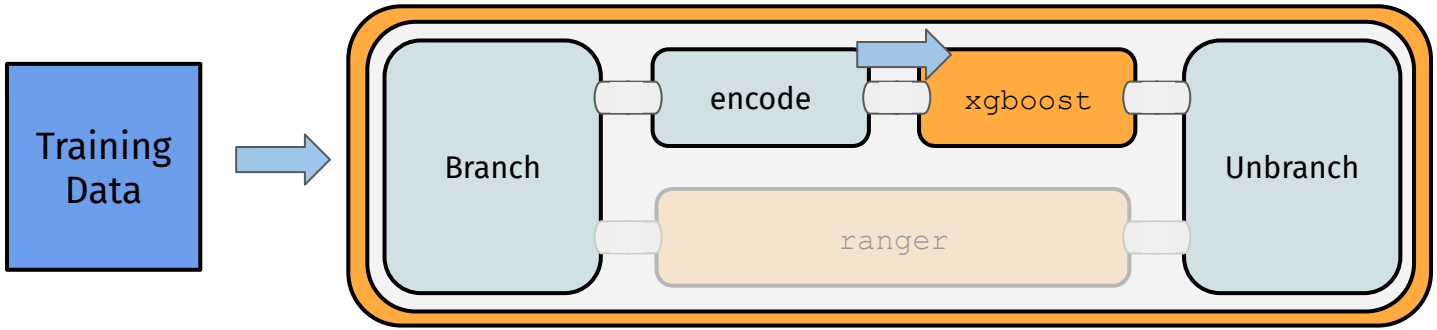
Pipelines in mlr3: mlr3pipelines

- Large library of common preprocessing operations
- Automatically avoids leaking test-set information into the training set
- Pipelines convert to `Learners`, work transparently with the rest of `mlr3`
- Advanced graph-structures: model ensembling



Pipelines in mlr3: mlr3pipelines

- Large library of common preprocessing operations
- Automatically avoids leaking test-set information into the training set
- Pipelines convert to **`Learners`**, work transparently with the rest of **`mlr3`**
- Advanced graph-structures: model ensembling
- Advanced graph-structures: alternative path branching -- control (and optimize) data-flow through hyperparameters



Next Steps

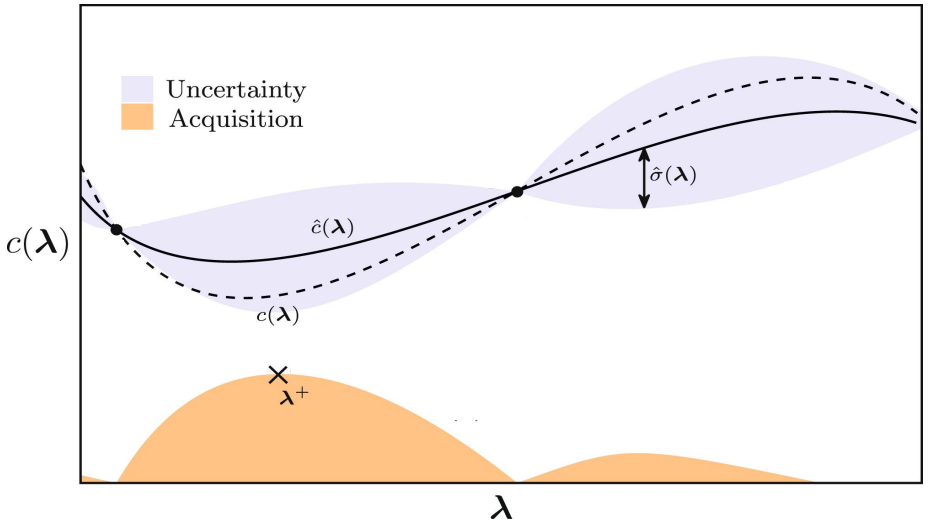
Going Beyond

More sophisticated optimization methods:

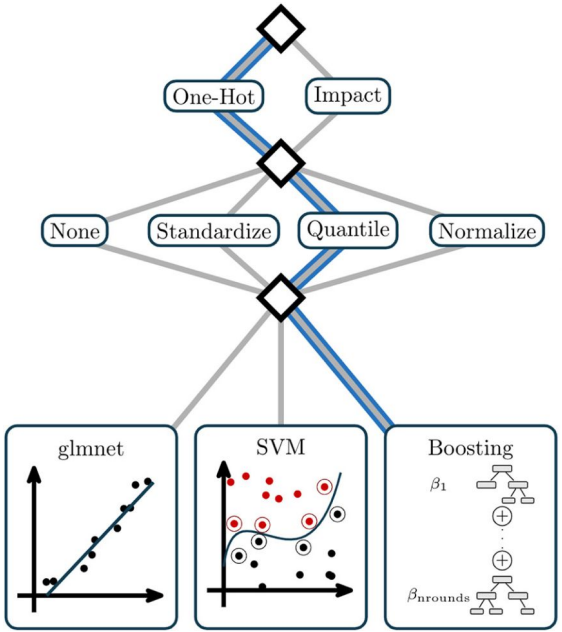
mlr³hyperband
Hyperband Parameter Tuning

mlr³mbo
Bayesian Optimization

miesmuschel
Flexible Mixed Integer Evolutionary Strategies



Optimizing pipeline-configurations and hyperparameters simultaneously



Learn More!

Free online `mlr3` book:

mlr3book.mlr-org.com

(physical copy is on the way)

- Examples from this talk: https://tinyurl.com/mlr3germancredit
- Slides: https://tinyurl.com/mlr3vfs