

mlr3tuning :: CHEAT SHEET

The **mlr3tuning** package provides hyperparameter tuning for mlr3.



Intro

The **mlr3tuning** package is an extension for the **mlr3** package which provides R6 classes for hyperparameter tuning.

Terminator

The Terminator determines when to stop the tuning. The package provides four Terminator classes:

```
terminator = TerminatorClockTime$new()
```

Terminate after a given time

```
terminator = TerminatorEvals$new()
```

Terminate after a given amount of iterations

```
terminator = TerminatorPerfReached$new()
```

Terminate after a specific performance is reached

```
terminator = TerminatorStagnation$new()
```

Terminate when tuning does not improve

```
terminator = TerminatorCombo$new()
```

A combination of the above in an ALL or ANY fashion

```
terminator = term(key, ...)
```

Get terminator by *key* and construct terminator with settings (...) in one go.

ParamterSet

The ParamSet defines the hyperparamter to tune and the tuning space.

```
tune_ps = ParamSet$new(list(
  ParamInt$new(id, lower, upper)))
```

The *id* (character value) refers to the hyperparameter of the learner. The *lower* and *upper* parameter define the bounds of the tuning space.

TuningInstance

The TuningInstance specifies a general search scenario.

```
instance = TuningInstance$new(
  task,
  learner,
  resampling,
  param_set,
  terminator)
```

The TuningInstance is constructed by supplying a *task*, *learner*, *resampling*, *param set* and *terminator*.

Tuner

The Tuner describes the tuning strategy. The package provides three algorithms:

```
tuner = TunerGridSearch$new()
```

Grid search

```
tuner = TunerRandomSearch$new()
```

Random Search

```
tuner = TunerGenSA$new()
```

Generalized Simulated Annealing

```
tuner = trn(key, ...)
```

Get tuner by *key* and construct tuner with settings (...) in one go.

Automatic Tuning

The AutoTuner wraps a learner and augments it with an automatic tuning for a given set of hyperparameters.

```
at = AutoTuner$new(
  learner,
  resampling,
  measures,
  tune_ps,
  terminator,
  tuner)
```