

# Package ‘mlr3viz’

February 19, 2020

**Title** Visualizations for 'mlr3'

**Version** 0.1.1

**Description** Provides visualizations for 'mlr3' objects such as tasks, predictions, resample results or benchmark results via the autoplot() generic of 'ggplot2'. The returned 'ggplot' objects are intended to provide sensible defaults, yet can easily be customized to create camera-ready figures. Visualizations include barplots, boxplots, histograms, ROC curves, and Precision-Recall curves.

**License** LGPL-3

**URL** <https://mlr3viz.mlr-org.com>, <https://github.com/mlr-org/mlr3viz>

**BugReports** <https://github.com/mlr-org/mlr3viz/issues>

**Depends** R (>= 3.1.0)

**Imports** checkmate, data.table, ggplot2, mlr3misc, utils

**Suggests** GGally, lgr, mlr3 (>= 0.1.6), mlr3filters, mlr3proba, precrec, rpart, survival, testthat

**Encoding** UTF-8

**LazyData** true

**NeedsCompilation** no

**RoxygenNote** 7.0.2

**Author** Michel Lang [cre, aut] (<<https://orcid.org/0000-0001-9754-0393>>)

**Maintainer** Michel Lang <[michellang@gmail.com](mailto:michellang@gmail.com)>

**Repository** CRAN

**Date/Publication** 2020-02-19 12:40:09 UTC

## R topics documented:

mlr3viz-package	2
as_precrec	2
autoplot.BenchmarkResult	3
autoplot.Filter	4

autoplot.PredictionClassif . . . . .	5
autoplot.PredictionRegr . . . . .	6
autoplot.ResampleResult . . . . .	7
autoplot.TaskClassif . . . . .	8
autoplot.TaskRegr . . . . .	9
autoplot.TaskSurv . . . . .	10
plot_learner_prediction . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

mlr3viz-package	<i>mlr3viz: Visualizations for 'mlr3'</i>
-----------------	---

---

## Description

Provides visualizations for 'mlr3' objects such as tasks, predictions, resample results or benchmark results via the autoplot() generic of 'ggplot2'. The returned 'ggplot' objects are intended to provide sensible defaults, yet can easily be customized to create camera-ready figures. Visualizations include barplots, boxplots, histograms, ROC curves, and Precision-Recall curves.

## Author(s)

**Maintainer:** Michel Lang <michellang@gmail.com> ([ORCID](#))

## See Also

Useful links:

- <https://mlr3viz.mlr-org.com>
- <https://github.com/mlr-org/mlr3viz>
- Report bugs at <https://github.com/mlr-org/mlr3viz/issues>

---

as_precrec	<i>Convert to 'precrec' Format</i>
------------	------------------------------------

---

## Description

Converts to a format which is understood by `precrec::evalmod()` of package **precrec**.

**Usage**

```
as_precrec(object)

## S3 method for class 'PredictionClassif'
as_precrec(object)

## S3 method for class 'ResampleResult'
as_precrec(object)

## S3 method for class 'BenchmarkResult'
as_precrec(object)
```

**Arguments**

```
object      :: any
              Object to convert.
```

**Value**

Object as created by `precrec::mmdata()`.

---

autoplot.BenchmarkResult

*Plot for BenchmarkResult*

---

**Description**

Generates plots for `mlr3::BenchmarkResult`, depending on argument type:

- "boxplot" (default): Boxplots of performance measures, one box per `mlr3::Learner` and one facet per `mlr3::Task`.
- "roc": ROC curve (1 - specificity on x, sensitivity on y). The `mlr3::BenchmarkResult` may only have a single `mlr3::Task` and a single `mlr3::ResampleResult`. Note that you can subset any `mlr3::BenchmarkResult` with its `$filter()` method (see examples). Requires package **precrec**.
- "prc": Precision recall curve. See "roc".

**Usage**

```
## S3 method for class 'BenchmarkResult'
autoplot(object, type = "boxplot", measure = NULL, ...)
```

**Arguments**

```
object      (mlr3::BenchmarkResult).
type        (character(1)):
              Type of the plot. See description.
measure     (mlr3::Measure).
...         (any): Additional arguments, passed down to the respective geom.
```

**Value**

`ggplot2::ggplot()` object.

**Examples**

```
library(mlr3)
library(mlr3viz)

tasks = tsks(c("spam", "pima", "sonar"))
learner = lrns(c("classif.featureless", "classif.rpart"), predict_type = "prob")
resampling = rsmps("cv")
object = benchmark(benchmark_grid(tasks, learner, resampling))

head(fortify(object))
autoplot(object)
autoplot(object$clone())$filter(task_ids = "spam", type = "roc")
autoplot(object$clone())$filter(task_ids = "pima", type = "prc")
```

---

autoplot.Filter	<i>Plot for Filter Scores</i>
-----------------	-------------------------------

---

**Description**

Generates plots for `mlr3filters::Filter`, depending on argument type:

- "barplot" (default): Bar plot of filter scores.

**Usage**

```
## S3 method for class 'Filter'
autoplot(object, type = "boxplot", n = Inf, ...)
```

**Arguments**

object	( <code>mlr3filters::Filter</code> ).
type	( <code>character(1)</code> ): Type of the plot. See description.
n	( <code>integer(1)</code> ) Only include the first n features with highest importance. Defaults to all features.
...	( <code>any</code> ): Additional argument, passed down to the respective geom.

**Value**

`ggplot2::ggplot()` object.

**Examples**

```
library(mlr3)
library(mlr3viz)
library(mlr3filters)

task = tsk("mtcars")
f = flt("correlation")
f$calculate(task)

head(fortify(f))
autoplot(f, n = 5)
```

---

```
autoplot.PredictionClassif
```

*Plot for PredictionClassif*

---

**Description**

Generates plots for `mlr3::PredictionClassif`, depending on argument type:

- "stacked" (default): Stacked barplot of true and estimated class labels.
- "roc": ROC curve (1 - specificity on x, sensitivity on y). Requires package **precrec**.
- "prc": Precision recall curve. Requires package **precrec**.

**Usage**

```
## S3 method for class 'PredictionClassif'
autoplot(object, type = "stacked", ...)
```

**Arguments**

object	( <code>mlr3::PredictionClassif</code> ).
type	(character(1)): Type of the plot. See description.
...	(any): Additional arguments, passed down to the respective geom.

**Value**

`ggplot2::ggplot()` object.

**Examples**

```
library(mlr3)
library(mlr3viz)

task = tsk("spam")
learner = lrn("classif.rpart", predict_type = "prob")
```

```

object = learner$train(task)$predict(task)

head(fortify(object))
autoplot(object)
autoplot(object, type = "roc")
autoplot(object, type = "prc")

```

---

```
autoplot.PredictionRegr
```

*Plot for PredictionRegr*

---

## Description

Generates plots for [mlr3::PredictionRegr](#), depending on argument type:

- "xy" (default): Scatterplot of true response vs predicted response. Additionally fits a linear model to visualize a possible trend.
- "histogram": Histogram of residuals  $r = y - \hat{y}$ .
- "residual": Plot of the residuals, with the response  $\hat{y}$  on the "x" and the residuals on the "y" axis.

## Usage

```

## S3 method for class 'PredictionRegr'
autoplot(object, type = "xy", ...)

```

## Arguments

object	( <a href="#">mlr3::PredictionRegr</a> ).
type	(character(1)): Type of the plot. See description.
...	(any): Additional arguments, passed down to the respective geom.

## Value

[ggplot2::ggplot\(\)](#) object.

## Examples

```

library(mlr3)
library(mlr3viz)

task = tsk("boston_housing")
learner = lrn("regr.rpart")
object = learner$train(task)$predict(task)

head(fortify(object))
autoplot(object)
autoplot(object, type = "histogram", binwidth = 1)

```

---

`autoplot.ResampleResult`*Plot for ResampleResult*

---

## Description

Generates plots for `mlr3::ResampleResult`, depending on argument type:

- "boxplot" (default): Boxplot of performance measures.
- "histogram": Histogram of performance measures.
- "roc": ROC curve (1 - specificity on x, sensitivity on y). The predictions of the individual `mlr3::Resamplings` are merged prior to calculating the ROC curve (micro averaged). Requires package `precrec`.
- "prc": Precision recall curve. See "roc".
- "prediction": Plots the learner prediction for a grid of points. Needs models to be stored. Set `store_models = TRUE` for `[mlr3::resample]`. For classification, we support tasks with exactly two features and learners with `predict_type=` set to "response" or "prob". For regression, we support tasks with one or two features. For tasks with one feature we can print confidence bounds if the predict type of the learner was set to "se". For tasks with two features the predict type will be ignored.

## Usage

```
## S3 method for class 'ResampleResult'  
autoplot(object, type = "boxplot", measure = NULL, predict_sets = "test", ...)
```

## Arguments

<code>object</code>	( <code>mlr3::ResampleResult</code> ).
<code>type</code>	( <code>character(1)</code> ): Type of the plot. See description.
<code>measure</code>	( <code>mlr3::Measure</code> ).
<code>predict_sets</code>	( <code>character()</code> ) Only for type set to "prediction". Which points should be shown in the plot? Can be a subset of ("train", "test") or empty.
<code>...</code>	(any): Additional arguments, passed down to the respective geom.

## Value

`ggplot2::ggplot()` object.

**Examples**

```

library(mlr3)
library(mlr3viz)

task = tsk("sonar")
learner = lrn("classif.rpart", predict_type = "prob")
resampling = rsmp("cv")
object = resample(task, learner, resampling)

head(fortify(object))

# Default: boxplot
autoplot(object)

# Histogram
autoplot(object, type = "histogram", bins = 30)

# ROC curve, averaged over resampling folds:
autoplot(object, type = "roc")

# ROC curve of joint prediction object:
autoplot(object$prediction(), type = "roc")

# Precision Recall Curve
autoplot(object, type = "prc")

# Prediction Plot
task = tsk("iris")$select(c("Sepal.Length", "Sepal.Width"))
resampling = rsmp("cv", folds = 3)
object = resample(task, learner, resampling, store_models = TRUE)
autoplot(object, type = "prediction")

```

---

autoplot.TaskClassif *Plot for Classification Tasks*

---

**Description**

Generates plots for [mlr3::TaskClassif](#), depending on argument type:

- "target" (default): Bar plot of the target variable (default).
- "duo": Passes data and additional arguments down to [GGally::ggduo\(\)](#). columnsX is target, columnsY is features.
- "pairs": Passes data and additional arguments down to [GGally::ggpairs\(\)](#). Color is set to target column.

**Usage**

```

## S3 method for class 'TaskClassif'
autoplot(object, type = "target", ...)

```



**Arguments**

object	(mlr3::TaskClassif).
type	(character(1)): Type of the plot. See description.
...	(any): Additional argument, possibly passed down to the underlying plot functions.

**Value**

ggplot2::ggplot() object.

**Examples**

```
library(mlr3)
library(mlr3viz)

task = mlr_tasks$get("iris")

head(fortify(task))
autoplot(task)
autoplot(task$clone())$select(c("Sepal.Length", "Sepal.Width")), type = "pairs")
autoplot(task, type = "duo")
```

---

autoplot.TaskRegr      *Plot for Regression Tasks*

---

**Description**

Generates plots for [mlr3::TaskRegr](#), depending on argument type:

- "target": Box plot of target variable (default).
- "pairs": Passes data and additional arguments down to [GGally::ggpairs\(\)](#). Color is set to target column.

**Usage**

```
## S3 method for class 'TaskRegr'
autoplot(object, type = "target", ...)
```

**Arguments**

object	(mlr3::TaskRegr).
type	(character(1)): Type of the plot. See description.
...	(any): Additional argument, passed down to the underlying geom or plot functions.

**Value**

`ggplot2::ggplot()` object.

**Examples**

```
library(mlr3)
library(mlr3viz)

task = mlr_tasks$get("mtcars")
task$select(c("am", "carb"))

head(fortify(task))
autoplot(task)
autoplot(task, type = "pairs")
```

---

autoplot.TaskSurv      *Plot for Survival Tasks*

---

**Description**

Generates plots for `mlr3proba::TaskSurv`, depending on argument type:

- "target": Calls `GGally::ggsurv()` on a `survival::survfit()` object.
- "duo": Passes data and additional arguments down to `GGally::ggduo()`. `columnsX` is target, `columnsY` is features.
- "pairs": Passes data and additional arguments down to `GGally::ggpairs()`. Color is set to target column.

**Usage**

```
## S3 method for class 'TaskSurv'
autoplot(object, type = "target", ...)
```

**Arguments**

object	( <code>mlr3proba::TaskSurv</code> ).
type	(character(1)): Type of the plot. Available choices:
...	(any): Additional argument, passed down to <code>\$formula</code> of <code>mlr3proba::TaskSurv</code> or the underlying plot functions.

**Value**

`ggplot2::ggplot()` object.

## Examples

```
library(mlr3)
library(mlr3viz)
library(mlr3proba)

task = mlr_tasks$get("lung")

head(fortify(task))
autoplot(task)
autoplot(task, rhs = "sex")
autoplot(task, type = "duo")
```

---

plot\_learner\_prediction

*Plot for Learner Predictions*

---

## Description

Generates a plot for the [mlr3::Prediction](#) of a single [mlr3::Learner](#) on a single [mlr3::Task](#).

- For classification we support tasks with exactly two features and learners with `predict_type` set to "response" or "prob".
- For regression we support tasks with one or two features. For tasks with one feature we print confidence bounds if the predict type of the learner was set to "se". For tasks with two features the predict type will be ignored.

Note that this function is a wrapper around `autoplot.ResampleResult()` for a temporary [mlr3::ResampleResult](#) using [mlr3::mlr\\_resamplings\\_holdout](#) with ratio 1 (all observations in training set).

## Usage

```
plot_learner_prediction(learner, task, grid_points = 100L, expand_range = 0)
```

## Arguments

learner	( <a href="#">mlr3::Learner</a> ).
task	( <a href="#">mlr3::Task</a> ).
grid_points	( <code>integer(1)</code> ) Resolution of the grid. For factors, ordered and logicals this value is ignored.
expand_range	( <code>numeric(1)</code> ) Expand the prediction range for numerical features.

## Value

`ggplot2::ggplot()` object.

**Examples**

```
library(mlr3)
library(mlr3viz)

task = tsk("pima")$select(c("age", "glucose"))
learner = lrn("classif.rpart", predict_type = "prob")
p = plot_learner_prediction(learner, task)
print(p)
```

# Index

as\_precrec, [2](#)  
autoplot.BenchmarkResult, [3](#)  
autoplot.Filter, [4](#)  
autoplot.PredictionClassif, [5](#)  
autoplot.PredictionRegr, [6](#)  
autoplot.ResampleResult, [7](#)  
autoplot.ResampleResult(), [11](#)  
autoplot.TaskClassif, [8](#)  
autoplot.TaskRegr, [9](#)  
autoplot.TaskSurv, [10](#)  
  
GGally::ggduo(), [8](#), [10](#)  
GGally::ggpairs(), [8–10](#)  
GGally::ggsurv(), [10](#)  
ggplot2::ggplot(), [4–7](#), [9–11](#)  
  
mlr3::BenchmarkResult, [3](#)  
mlr3::Learner, [3](#), [11](#)  
mlr3::Measure, [3](#), [7](#)  
mlr3::mlr\_resamplings\_holdout, [11](#)  
mlr3::Prediction, [11](#)  
mlr3::PredictionClassif, [5](#)  
mlr3::PredictionRegr, [6](#)  
mlr3::ResampleResult, [3](#), [7](#), [11](#)  
mlr3::Resampling, [7](#)  
mlr3::Task, [3](#), [11](#)  
mlr3::TaskClassif, [8](#), [9](#)  
mlr3::TaskRegr, [9](#)  
mlr3filters::Filter, [4](#)  
mlr3proba::TaskSurv, [10](#)  
mlr3viz (mlr3viz-package), [2](#)  
mlr3viz-package, [2](#)  
  
plot\_learner\_prediction, [11](#)  
precrc::evalmod(), [2](#)  
precrc::mmdata(), [3](#)  
  
survival::survfit(), [10](#)