

Package ‘mlr3viz’

January 8, 2020

Title Visualizations for 'mlr3'

Version 0.1.0

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>

Repository CRAN

Date/Publication 2020-01-08 17:10:02 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

Index	11
--------------	-----------

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 = rsmpl("cv", folds = 3)
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.
...	(any): 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 $r = y - \hat{y}$.

Usage

```

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

```

Arguments

object	(mlr3::PredictionRegr).
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".

Usage

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

Arguments

object	(mlr3::ResampleResult).
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)  
  
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")

```

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	(<code>mlr3::TaskClassif</code>).
type	(<code>character(1)</code>): 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	(<code>mlr3::TaskRegr</code>).
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	(<code>character(1)</code>): 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")
```

Index

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](#)

GGally::ggduo(), [8](#), [10](#)
GGally::ggpairs(), [8–10](#)
GGally::ggsurv(), [10](#)
ggplot2::ggplot(), [4–10](#)

mlr3::BenchmarkResult, [3](#)
mlr3::Learner, [3](#)
mlr3::Measure, [3](#), [7](#)
mlr3::PredictionClassif, [5](#)
mlr3::PredictionRegr, [6](#)
mlr3::ResampleResult, [3](#), [7](#)
mlr3::Resampling, [7](#)
mlr3::Task, [3](#)
mlr3::TaskClassif, [8](#)
mlr3::TaskRegr, [9](#)
mlr3filters::Filter, [4](#)
mlr3proba::TaskSurv, [10](#)
mlr3viz (mlr3viz-package), [2](#)
mlr3viz-package, [2](#)

precrc::evalmod(), [2](#)
precrc::mmdata(), [3](#)

survival::survfit(), [10](#)