

Package ‘RBPcurve’

July 21, 2025

Type Package

Title The Residual-Based Predictiveness Curve

Version 1.3

Date 2025-04-11

URL <https://github.com/giuseppec/RBPcurve>

BugReports <https://github.com/giuseppec/RBPcurve/issues>

Description The RBP curve is a visual tool to assess the performance of prediction models.

License GPL-3

Depends R ($\geq 3.0.2$), mlr (≥ 2.11)

Imports BBmisc (≥ 1.11), checkmate ($\geq 1.8.2$), shape, TeachingDemos

Suggests testthat, mlbench

RoxygenNote 7.3.2

NeedsCompilation no

Author Giuseppe Casalicchio [cre, aut] (ORCID:
<<https://orcid.org/0000-0001-5324-5966>>),
Bernd Bischl [ctb] (ORCID: <<https://orcid.org/0000-0001-6002-6980>>)

Maintainer Giuseppe Casalicchio <giuseppe.casalicchio@stat.uni-muenchen.de>

Repository CRAN

Date/Publication 2025-04-11 11:20:05 UTC

Contents

addGoodCalib	2
addPEV	3
addPrevalence	4
addRates	4
addWellCalib	5
makeRBPObj	6
plotRBPcurve	7

Index	10
--------------	-----------

addGoodCalib

Visualizes a measure for good calibration on the RBP curve.

Description

The integral of the RBP curve is a measure for good calibration. If the sum of the two integrals (below and above the RBP curve) is close to 0, good calibration is satisfied and the prevalence is close to the average predicted probabilities.

Usage

```
addGoodCalib(
  obj,
  plot.values = TRUE,
  show.info = TRUE,
  col = grDevices::rgb(0, 0, 0, 0.25),
  border = NA,
  ...
)
```

Arguments

obj	[RBPObj] Data container for RBP curve.
plot.values	[logical(1)] Whether the values of the corresponding measure should be added to the plot? Default is FALSE.
show.info	[logical(1)] Print more information for the respective measure on console? Default is TRUE.
col	[vector(1)] Color for filling the polygon, as in polygon . Default is “grey”.
border	[vector(1)] Color to draw the borders, as in polygon . Default is NA to omit borders.
...	[any] Passed to polygon .

Value

[invisible(NULL)].

addPEV	<i>Visualize the PEV on the RBP curve.</i>
--------	--

Description

The PEV measure is the difference between the conditional expectation of the predicted probabilities (conditional on the two groups that are determined by the target variable). The PEV measure can be visually obtained by the RBP curve, namely by the difference of the two areas that are Highlighted with addPEV.

Usage

```
addPEV(  
  obj,  
  plot.values = TRUE,  
  show.info = TRUE,  
  text.col = "black",  
  col = rgb(0, 0, 0, 0.25)  
)
```

Arguments

obj	[RBPObj] Data container for RBP curve.
plot.values	[logical(1)] Whether the values of the corresponding measure should be added to the plot? Default is FALSE.
show.info	[logical(1)] Print more information for the respective measure on console? Default is TRUE.
text.col	[character(1) numeric(1)] Text color, used when plot.values = TRUE, otherwise ignored. Default is "black".
col	[character(1) numeric(1)] A specification for the plotting color.

Value

[invisible(NULL)].

addPrevalence	<i>Visualizes the prevalence on the RBP curve.</i>
---------------	--

Description

The prevalence is the proportion of a population having a specific condition. In binary classification, the condition refers to whether the target variable has the value 1, that is, whether the target variable corresponds to the positive class.

Usage

```
addPrevalence(obj, plot.values = TRUE, digits = 3L, col = "grey")
```

Arguments

obj	[RBPObj] Data container for RBP curve.
plot.values	[logical(1)] Whether the values of the corresponding measure should be added to the plot? Default is FALSE.
digits	[numeric(1)] Indicates the number of decimal places for the values that are plotted when plot.values = TRUE. Default is 3L.
col	[character(1) numeric(1)] A specification for the plotting color.

Value

[invisible(NULL)].

addRates	<i>Visualizes the TPR and FPR on the RBP curve.</i>
----------	---

Description

For a given threshold tresh, the true positive rate (TPR) and the false positive rate (FPR) can be visually assessed by the RBP curve by the intersection of the RBP curve with the horizontal lines at -tresh and 1 - tresh, respectively.

Usage

```
addRates(
  obj,
  plot.values = TRUE,
  digits = 3L,
  col = "black",
  thresh = obj$prev,
  thresh.label = "thresh"
)
```

Arguments

obj	[RBPObj] Data container for RBP curve.
plot.values	[logical(1)] Whether the values of the corresponding measure should be added to the plot? Default is FALSE.
digits	[numeric(1)] Indicates the number of decimal places for the values that are plotted when plot.values = TRUE. Default is 3L.
col	[character(1) numeric(1)] A specification for the plotting color.
thresh	[numeric(1)] Threshold that is used to compute the true positive and false positive rate. Default is prevalence.
thresh.label	[character(1)] The label for the threshold that is plotted when plot.values = TRUE.

Value

[invisible(NULL)].

addWellCalib

Visualizes a measure for well calibration on the RBP curve.

Description

A measure for a well calibrated model can be obtained by grouping the predicted probabilities via deciles yielding 10 groups. The equally colored areas belong to a specific group. When each of the two equally colored areas are similar, the model is well calibrated.

Usage

```
addWellCalib(
  obj,
  plot.values = TRUE,
  subplot.control = list(diff = TRUE),
  col = shape::greycol(10L, interval = c(0.3, 1)),
  pos = NULL
)
```

Arguments

obj	[RBPObj] Data container for RBP curve.
plot.values	[logical(1)] Whether the values of the corresponding measure should be added to the plot? Default is FALSE.
subplot.control	[list] A named list of arguments that will be passed to barplot . Additionally, you can set diff = TRUE to plot differences of the equally colored areas or diff = FALSE to directly plot the areas of the equally colored areas in juxtaposed bars.
col	[character numeric] A specification for the the plotting color for the areas.
pos	[list] A named List that determines the x and y positioning of a subplot that compares the areas in additional barplots (see subplot). Can be NA for no additional subplot. Default is pos = NULL for an auto positioning in the topleft quadrant.

Value

A matrix that contains the average of the “probabilities within deciles” conditional on Y.

makeRBPObj	Create data container for RBP curve.
------------	--------------------------------------

Description

Must be created for all subsequent plot function calls.

Usage

```
makeRBPObj(pred, y, positive = NULL)
```

Arguments

pred	[numeric] Predicted probabilities for each observation.
y	[numeric factor] Class labels of the target variable. Either a numeric vector with values 0 or 1, or a factor with two levels.
positive	[character(1)] Set positive class label for target variable which is transformed as 1 to compute. Only needed when y is a "factor".

Value

Object members:

n [numeric(1)] Number of observations.
 pred [numeric(n)] Predicted probabilities.
 y [numeric(n)] Target variable having the values 0 and 1.
 positive [character(1)] Positive class label of target variable. Only present when y is a factor.
 e0 [numeric(1)] Average of the predicted probabilities conditional on y=0.
 e1 [numeric(1)] Average of the predicted probabilities conditional on y=1.
 pev [numeric(1)] Proportion of explained variation measure. Computed as e1-e0.
 tpr [numeric(1)] True positive rate.
 fpr [numeric(1)] False positive rate.
 prev [numeric(1)] Prevalence.
 one.min.prev [numeric(1)] One minus the value of the prevalence.
 axis.x [numeric(n)] Values for the X-Axis of the RBP curve.
 axis.y [numeric(n)] Values for the Y-Axis of the RBP curve.

plotRBPCurve

Plot residual-based predictiveness (RBP) curve.

Description

plots the RBP curve

Usage

```
plotRBPCurve(
  obj,
  main = "RBP Curve",
  xlab = "Cumulative Percentage",
  ylab = "Estimated Residuals",
  type = "l",
  ylim = c(-1, 1.2),
  x.adj = c(NA, -0.5),
  y.adj = c(NA, NA),
  cond.axis = FALSE,
  title.line = ifelse(cond.axis, 3, 2),
  add = FALSE,
  ...
)
```

Arguments

obj	[RBPObj] Data container for RBP curve.
main	[character(1)] An overall title for the plot.
xlab	[character(1)] Label for X-axis. Default is "Cumulative Percentage".
ylab	[character(1)] Label for Y-axis. Default is "Estimated Residuals".
type	[character(1)] The plot type that should be drawn, see plot for all possible types. Default is type = "l" for lines.
ylim	[numeric(2)] Limits for Y-axis. Default is c(-1, 1.1).
x.adj	[numeric(2)] Adjustment for the X-axis.
y.adj	[numeric(2)] Adjustment for the Y-axis.
cond.axis	[logical(1)] Should an additional axis be plotted reflecting residuals conditional on y? Default is FALSE.
title.line	[integer(1)] Where to plot the title, see title .
add	[logical(1)] Should RBP plot be added to current plot? Default is FALSE.
...	[any] Passed to plot or lines , depending on add.

Examples

```
# Download data
mydata = getTaskData(pid.task)
head(mydata)

# Build logit model and plot RBP curve
mylogit <- glm(diabetes ~ ., data = mydata, family = "binomial")
y = mydata$diabetes
pred1 = predict(mylogit, type="response")
obj1 = makeRBPObj(pred1, y)
plotRBPCurve(obj1, cond.axis = TRUE, type = "b")

## Not run:
# Build logit model using mlr and plot RBP curve
task = pid.task
lrn = makeLearner("classif.logreg", predict.type = "prob")
tr = train(lrn, task)
pred2 = getPredictionProbabilities(predict(tr, task))
obj2 = makeRBPObj(pred2, y)
plotRBPCurve(obj2, cond.axis = TRUE, type = "b", col = 2)

## End(Not run)
```

Index

`addGoodCalib`, [2](#)
`addPEV`, [3](#)
`addPrevalence`, [4](#)
`addRates`, [4](#)
`addWellCalib`, [5](#)

`barplot`, [6](#)

`lines`, [8](#)

`makeRBPObj`, [6](#)

`plot`, [8](#)
`plotRBPCurve`, [7](#)
`polygon`, [2](#)

`RBPObj`, [2–6](#), [8](#)
`RBPObj (makeRBPObj)`, [6](#)

`subplot`, [6](#)

`title`, [8](#)