Package ‘CatPredi’

May 8, 2026

Title Optimal Categorisation of Continuous Variables in Prediction Models

Version 2.0

Description Allows the user to categorise a continuous predictor variable in a logistic or a Cox proportional hazards regression setting, by maximising the discriminative ability of the model. I Barrio, I Arostegui, MX Rodriguez-Alvarez, JM Quintana (2015) <doi:10.1177/0962280215601873>. I Barrio, MX Rodriguez-Alvarez, L Meira-Machado, C Esteban, I Arostegui (2017) <`https://www.idescat.cat/sort/sort411/41.1.3.barrio-etal.pdf`>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Imports CPE, doParallel, foreach, ggplot2, graphics, mgcv, rgenoud, rms, stats, survival

NeedsCompilation no

Author Irantzu Barrio [aut, cre],
Maria Xose Rodriguez-Alvarez [aut],
Inmaculada Arostegui [ctb],
Diana Marcela Perez [ctb]

Maintainer Irantzu Barrio <`irantzu.barrio@ehu.eus`>

Repository CRAN

Date/Publication 2026-05-08 07:42:31 UTC

Contents

`catpredi` *Function to obtain optimal cut points to categorise a continuous predictor variable in a logistic regression model*

Description

Returns an object with the optimal cut points to categorise a continuous predictor variable in a logistic regression model

Usage

```
catpredi(
  formula,
  cat.var,
  cat.points = 1,
  data,
  method = c("addfor", "genetic", "backaddfor"),
  range = NULL,
  correct.AUC = FALSE,
  control = controlcatpredi(),
  ...
)
```

Arguments

`formula`	An object of class `formula` giving the model to be fitted in addition to the continuous covariate is aimed to categorise. This argument allows the user to specify whether the continuous predictor should be categorised in a univariable context, or in presence of other covariates or cofounders, i.e in a multiple logistic regression model. For instance, `Y ~ 1` indicates that the categorisation should be done in a univariable setting, with Y being the response variable. If the predictor variable is aimed to be categorised in a multivariable setting, this argument allows to specify whether the covariates should be modelled using linear or non linear effects. In the latest, the effects are estimated using the **mgcv** package.
`cat.var`	Name of the continuous variable to categorise.
`cat.points`	Number of cut points to look for.
`data`	Data frame containing all needed variables.
`method`	The algorithm selected to search for the optimal cut points. `"addfor"` if the AddFor algorithm is choosen, `"backaddfor"` if the BackAddFor algorithm is selected and `"genetic"` otherwise.

`range`	The range of the continuous variable in which to look for the cut points. By default `NULL`, i.e, all the range.
`correct.AUC`	A logical value. If `TRUE` the bias corrected AUC is estimated.
`control`	Output of the `controlcatpredi` function.
`...`	Further arguments for passing on to the function `genoud` of the package **rgenoud**.

Value

Returns an object of class `"catpredi"` with the following components:

call The matched call.

method The algorithm selected in the call.

formula The model formula used in the call.

cat.var Name of the continuous variable to categorise.

data The data frame used in the call.

correct.AUC Logical value indicating whether bias-corrected AUC was used.

results A list containing estimated cut points, AUC and bias-corrected AUC for each method.

control The control parameters used in the call.

Author(s)

Irantzu Barrio, Maria Xose Rodriguez-Alvarez, Inmaculada Arostegui, Javier Roca-Pardinas and Xabier Amutxastegi.

References

I Barrio, J Roca-Pardinas and I Arostegui (2021). Selecting the number of categories of the lymph node ratio in cancer research: A bootstrap-based hypothesis test. *Statistical Methods in Medical Research*, 30(3), 926-940.

I Barrio, I Arostegui, M.X Rodriguez-Alvarez and J.M Quintana (2017). A new approach to categorising continuous variables in prediction models: proposal and validation. *Statistical Methods in Medical Research*, 26(6), 2586-2602.

S.N Wood (2006). Generalized Additive Models: An Introduction with R. Chapman and Hall/CRC.

See Also

`controlcatpredi`, `comp.cutpoints`, `plot.catpredi`, `summary.catpredi`

Examples

```
library(CatPredi)
## Not run:
set.seed(127)
#Simulate data
n = 100
#Predictor variable
```

```
xh <- rnorm(n, mean = 0, sd = 1)
xd <- rnorm(n, mean = 1.5, sd = 1)
x <- c(xh, xd)
#Response
y <- c(rep(0,n), rep(1,n))
#Covariate
zh <- rnorm(n, mean=1.5, sd=1)
zd <- rnorm(n, mean=1, sd=1)
z <- c(zh, zd)
# Data frame
df <- data.frame(y = y, x = x, z = z)

# Select optimal cut points using the AddFor algorithm
res.addfor <- catpredi(formula = y ~ z, cat.var = "x", cat.points = 2,
                       data = df, method = "addfor", range=NULL, correct.AUC=FALSE,
                       control=controlcatpredi(grid=20))

# Select optimal cut points using the BackAddFor algorithm
res.backaddfor <- catpredi(formula = y ~ z, cat.var = "x", cat.points = 3,
                        data = df, method = "backaddfor", range=NULL, correct.AUC=FALSE)

## End(Not run)
## Not run:
  set.seed(127)
  #Simulate data
  n = 200
  #Predictor variable
  xh <- rnorm(n, mean = 0, sd = 1)
  xd <- rnorm(n, mean = 1.5, sd = 1)
  x <- c(xh, xd)
  #Response
  y <- c(rep(0,n), rep(1,n))
  #Covariate
  zh <- rnorm(n, mean=1.5, sd=1)
  zd <- rnorm(n, mean=1, sd=1)
  z <- c(zh, zd)
  # Data frame
  df <- data.frame(y = y, x = x, z = z)

  # Select optimal cut points using the AddFor algorithm
  res.addfor <- catpredi(formula = y ~ z, cat.var = "x", cat.points = 3,
                        data = df, method = "addfor", range=NULL, correct.AUC=FALSE)

  # Select optimal cut points using the BackAddFor algorithm
  res.backaddfor <- catpredi(formula = y ~ z, cat.var = "x", cat.points = 3,
                        data = df, method = "backaddfor", range=NULL, correct.AUC=FALSE)

## End(Not run)
```

`catpredi.survival` *Function to obtain optimal cut points to categorise a continuous predictor variable in a Cox proportional hazards regression model*

Description

Returns an object with the optimal cut points to categorise a continuous predictor variable in a Cox proportional hazards regression model

Usage

```
catpredi.survival(
  formula,
  cat.var,
  cat.points = 1,
  data,
  method = c("addfor", "genetic", "backaddfor"),
  conc.index = c("cindex", "cpe"),
  range = NULL,
  correct.index = FALSE,
  control = controlcatpredi.survival(),
  ...
)
```

Arguments

`formula`	An object of class `formula` giving the model to be fitted in addition to the continuous covariate is aimed to categorise. The response must be a survival object as returned by the `Surv` function. This argument allows the user to specify whether the continuous predictor should be categorised in a univariable context, or in presence of other covariates or cofounders, i.e in a multiple Cox proportional hazards regression model. For instance, Surv(SurvT,SurvS)~1 indicates that the categorisation should be done in a univariable setting.
`cat.var`	Name of the continuous variable to categorise.
`cat.points`	Number of cut points to look for.
`data`	Data frame containing all needed variables.
`method`	The algorithm selected to search for the optimal cut points. `"addfor"` if the AddFor algorithm is choosen, `"backaddfor"` if the BackAddFor algorithm is selected and `"genetic"` otherwise.
`conc.index`	The concordance probability estimator selected for maximisation purposes. "cindex" if the c-index concordance probability is choosen and "cpe" otherwise. The c-index and CPE are estimated using the **rms** and **CPE** packages, respectively.
`range`	The range of the continuous variable in which to look for the cut points. By default `NULL`, i.e, all the range.
`correct.index`	A logical value. If TRUE the bias corrected concordance probability is estimated.

`control`	Output of the `controlcatpredi.survival` function.
`...`	Further arguments for passing on to the function `genoud` of the package **rgenoud**.

Value

Returns an object of class `"catpredi.survival"` with the following components:

call The matched call.

method The algorithm selected in the call.

formula an object of class `formula` giving the model to be fitted in addition to the continuous covariate is aimed to categorise.

cat.var name of the continuous variable to categorise.

data the data frame with the variables used in the call.

correct.index The logical value used in the call.

results a list with the estimated cut points, concordance probability and bias corrected concordance probability.

control the control parameters used in the call. When the c-index concordance probability is choosen, a list with the following components is obtained for each of the methods used in the call:

"cutpoints" Estimated optimal cut points.

"Cindex" Estimated c-index.

"Cindex.cor" Estimated bias corrected c-index. When the CPE concordance probability is choosen, a list with the following components is obtained for each of the methods used in the call:

"cutpoints" Estimated optimal cut points.

"CPE" Estimated CPE.

"CPE.cor" Estimated bias corrected CPE.

Author(s)

Irantzu Barrio and Maria Xose Rodriguez-Alvarez

References

I Barrio, M.X Rodriguez-Alvarez, L Meira-Machado, C Esteban and I Arostegui (2017). Comparison of two discrimination indexes in the categorisation of continuous predictors in time-to-event studies. *SORT*, 41:73-92

M Gonen and G Heller (2005). Concordance probability and discriminatory power in proportional hazards regression. *Biometrika*, 92:965-970.

F Harrell (2001). Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis. Springer.

See Also

`controlcatpredi.survival`, `comp.cutpoints.survival`, `plot.catpredi.survival`, `catpredi`

Examples

```
library(CatPredi)
library(survival)
set.seed(123)
#Simulate data
n = 500
tauc = 1
X <- rnorm(n=n, mean=0, sd=2)
SurvT <- exp(2*X + rweibull(n = n, shape=1, scale = 1))   + rnorm(n, mean=0, sd=0.25)
# Censoring time
CensTime <- runif(n=n, min=0, max=tauc)
# Status
SurvS <- as.numeric(SurvT <= CensTime)
# Data frame
dat <- data.frame(X = X, SurvT = pmin(SurvT, CensTime), SurvS = SurvS)

# Select optimal cut points using the AddFor algorithm
res <- catpredi.survival (formula= Surv(SurvT,SurvS)~1, cat.var="X", cat.points = 2,
                        data = dat, method = "addfor", conc.index = "cindex", range = NULL,
                            correct.index = FALSE)
```

`comp.cutpoints` *Selection of optimal number of cut points*

Description

Compares two objects of class `"catpredi"`.

Usage

```
comp.cutpoints(obj1, obj2, V = 100)
```

Arguments

`obj1`	An object inheriting from class `"catpredi"` for k number of cut points
`obj2`	An object inheriting from class `"catpredi"` for k+1 number of cut points
`V`	Number of bootstrap resamples. By default V=100

Value

This function returns an object of class `"comp.cutpoints"` with the following components:

AUC.cor.diff the difference of the bias corrected AUCs for the two categorical variables.

icb.auc.diff bootstrap based confidence interval for the bias corrected AUC difference.

Author(s)

Irantzu Barrio, Maria Xose Rodriguez-Alvarez and Inmaculada Arostegui.

References

I Barrio, I Arostegui, M.X Rodriguez-Alvarez and J.M Quintana (2017). A new approach to categorising continuous variables in prediction models: proposal and validation. *Statistical Methods in Medical Research*, 26(6), 2586-2602.

See Also

`catpredi`

Examples

```
library(CatPredi)
set.seed(127)
#Simulate data
n = 100
#Predictor variable
xh <- rnorm(n, mean = 0, sd = 1)
xd <- rnorm(n, mean = 1.5, sd = 1)
x <- c(xh, xd)
#Response
y <- c(rep(0,n), rep(1,n))
# Data frame
df <- data.frame(y = y, x = x)


  # Select 2 optimal cut points using the AddFor algorithm. Correct the AUC
    res.backaddfor.k2 <- catpredi(formula = y ~ 1, cat.var = "x", cat.points = 2,
                          data = df, method = "backaddfor", range=NULL, correct.AUC=TRUE,
                                  control=controlcatpredi(grid=100))
  # Select 3 optimal cut points using the AddFor algorithm. Correct the AUC
    res.backaddfor.k3 <- catpredi(formula = y ~ 1, cat.var = "x", cat.points = 3,
                          data = df, method = "backaddfor", range=NULL, correct.AUC=TRUE,
                                  control=controlcatpredi(grid=100))

  # Select optimal number of cut points
    comp <-  comp.cutpoints(res.backaddfor.k2, res.backaddfor.k3, V = 100)
```

---

comp.cutpoints.survival

Selection of optimal number of cut points

Description

Compares two objects of class `"catpredi.survival"`

Usage

```
comp.cutpoints.survival(obj1, obj2, V = 100)
```

Arguments

`obj1`	An object inheriting from class `"catpredi.survival"` for k number of cut points
`obj2`	An object inheriting from class `"catpredi.survival"` for k+1 number of cut points
`V`	Number of bootstrap resamples. By default V=100

Value

This function returns an object of class "comp.cutpoints.survival" with the following components:

CI.cor.diff the difference of the bias corrected concordance probability for the two categorical variables.

icb.CI.diff bootstrap based confidence interval for the bias corrected concordance probability difference.

Author(s)

Irantzu Barrio and Maria Xose Rodriguez-Alvarez.

References

I Barrio, M.X Rodriguez-Alvarez, L Meira-Machado, C Esteban and I Arostegui (2017). Comparison of two discrimination indexes in the categorisation of continuous predictors in time-to-event studies. *SORT*, 41:73-92

See Also

`catpredi.survival`

Examples

```
library(CatPredi)
library(survival)
set.seed(123)

#Simulate data
n = 300
tauc = 1
X <- rnorm(n=n, mean=0, sd=2)
SurvT <- exp(2*X + rweibull(n = n, shape=1, scale = 1))   + rnorm(n, mean=0, sd=0.25)
# Censoring time
CensTime <- runif(n=n, min=0, max=tauc)
# Status
SurvS <- as.numeric(SurvT <= CensTime)
# Data frame
dat <- data.frame(X = X, SurvT = pmin(SurvT, CensTime), SurvS = SurvS)

  # Select 2 optimal cut points using the AddFor algorithm. Correct the c-index
   res.k2 <- catpredi.survival (formula= Surv(SurvT,SurvS)~1, cat.var="X", cat.points = 2,
```

```
                                data = dat, method = "addfor", conc.index = "cindex",
                                range = NULL, correct.index = TRUE)
  # Select 3 optimal cut points using the AddFor algorithm. Correct the c-index
   res.k3 <- catpredi.survival (formula= Surv(SurvT,SurvS)~1, cat.var="X", cat.points = 3,
                                data = dat, method = "addfor", conc.index = "cindex",
                                range = NULL, correct.index = TRUE)
  # Select optimal number of cut points
    comp <-  comp.cutpoints.survival(res.k2, res.k3, V = 100)
```

`compare.AUC.ht` *A bootstrap-based hypothesis test to select the best number of categories for a continuous predictor variable in a logistic regression model*

Description

Compares two objects of class "catpredi" to evaluate the significance of the improvement in model performance (in terms of the AUC) by adding k+1 cut-off points to the predictor variable.

Usage

```
compare.AUC.ht(
  obj1,
  obj2,
  level = 0.95,
  nb = 100,
  parallel = TRUE,
  plot = TRUE
)
```

Arguments

`obj1`	An object inheriting from class `"catpredi"` for k number of cut points.
`obj2`	An object inheriting from class `"catpredi"` for k+1 number of cut points.
`level`	The confidence level required for the hypothesis test. By default level = 0.95.
`nb`	Number of bootstrap resamples. By default nb = 100
`parallel`	A logical value. if TRUE the bootstrap is processed in parallel.
`plot`	A logical value. if TRUE the density plot for the bootstrap statistic is provided.

Value

This function returns an object of class `"compare.AUC.ht"` with the following components:

t.null test statistic, with the difference of the AUCs for the two objects.

t.boot a vector with the `nb` bootstrap statistics.

t.null empirical `level`-percentile of the bootstrap statistics vector.

Author(s)

Irantzu Barrio, Inmaculada Arostegui, Javier Roca-Pardinas and Xabier Amutxastegi.

References

I Barrio, J Roca-Pardinas and I Arostegui (2021). Selecting the number of categories of the lymph node ratio in cancer research: A bootstrap-based hypothesis test. *Statistical Methods in Medical Research*, 30(3), 926-940.

See Also

`catpredi`, `comp.cutpoints`

Examples

```
library(CatPredi)
## Not run:
  set.seed(127)
  #Simulate data
  n = 100
  #Predictor variable
  xh <- rnorm(n, mean = 0, sd = 1)
  xd <- rnorm(n, mean = 1.5, sd = 1)
  x <- c(xh, xd)
  #Response
  y <- c(rep(0,n), rep(1,n))
  # Data frame
  df <- data.frame(y = y, x = x)
  # Select 2 optimal cut points using the AddFor algorithm. Correct the AUC
    res.addfor.k2 <- catpredi(formula = y ~ 1, cat.var = "x", cat.points = 2,
                            data = df, method = "addfor", range=NULL, correct.AUC=TRUE,
                             control=controlcatpredi(grid=20))
  # Select 3 optimal cut points using the AddFor algorithm. Correct the AUC
    res.addfor.k3 <- catpredi(formula = y ~ 1, cat.var = "x", cat.points = 3,
                            data = df, method = "addfor", range=NULL, correct.AUC=TRUE,
                             control=controlcatpredi(grid=20))
    comp <-  comp.cutpoints(res.addfor.k2, res.addfor.k3, V = 10)

  # Select 1 optimal cut points using the BackAddFor algorithm.
    res.backaddfor.k1 <- catpredi(formula = y ~ 1, cat.var = "x", cat.points = 1,
                          data = df, method = "backaddfor", range=NULL, correct.AUC=FALSE)
  # Select 2 optimal cut points using the BackAddFor algorithm.
    res.backaddfor.k2 <- catpredi(formula = y ~ 1, cat.var = "x", cat.points = 2,
                          data = df, method = "backaddfor", range=NULL, correct.AUC=FALSE)
  # Test if k=1 cut-off points is enough to categorise x
    comp.k1.k2 <-  compare.AUC.ht(res.backaddfor.k1, res.backaddfor.k2)

## End(Not run)
```

`controlcatpredi` *Control function*

Description

Function used to set several parameters to control the selection of the optimal cut points in a logistic regression model

Usage

```
controlcatpredi(
  min.p.cat = 1,
  grid = 100,
  B = 50,
  eps = 0.001,
  b.method = c("ncoutcome", "coutcome"),
  print.gen = 0
)
```

Arguments

`min.p.cat`	Set the minimun number of individuals in each category
`grid`	Grid size for the AddFor and BackAddFor algorithms
`B`	Number of bootstrap replicates for the AUC bias correction procedure
`eps`	An argument for the BackAddFor algorithm, indicating whether the improvement between iterations is considered significant
`b.method`	Allows to specify whether the bootstrap resampling should be done considering or not the outcome variable. The option "ncoutcome" indicates that the data is resampled without taking into account the response variable, while "coutcome" indicates that the data is resampled in regard to the response variable
`print.gen`	Corresponds to the argument print.level of the `genoud` function of the package **rgenoud**.

Value

A list with components for each of the possible arguments.

Author(s)

Irantzu Barrio, Maria Xose Rodriguez-Alvarez, Inmaculada Arostegui, Javier Roca-Pardinas and Xabier Amutxastegi.

References

Mebane Jr, W. R., & Sekhon, J. S. (2011). Genetic optimization using derivatives: the rgenoud package for R. *Journal of Statistical Software* 42**11**, 1-26.

See Also

catpredi

`controlcatpredi.survival`

Control function

Description

Function used to set several parameters to control the selection of the optimal cut points in a Cox proportional hazards regression model.

Usage

```
controlcatpredi.survival(
  min.p.cat = 5,
  grid = 100,
  B = 50,
  b.method = c("ncoutcome", "coutcome"),
  print.gen = 0
)
```

Arguments

`min.p.cat`	Set the minimun number of individuals in each category.
`grid`	Grid size for the AddFor algorithm.
`B`	Number of bootstrap replicates for the AUC bias correction procedure.
`b.method`	Allows to specify whether the bootstrap resampling should be done considering or not the outcome variable. The option "ncoutcome" indicates that the data is resampled without taking into account the response variable, while "coutcome" indicates that the data is resampled in regard to the response variable.
`print.gen`	Corresponds to the argument print.level of the `genoud` function of the package **rgenoud**.

Value

A list with components for each of the possible arguments.

Author(s)

Irantzu Barrio and Maria Xose Rodriguez-Alvarez.

References

Mebane Jr, W. R., & Sekhon, J. S. (2011). Genetic optimization using derivatives: the rgenoud package for R. *Journal of Statistical Software* 42**11**, 1-26.

See Also

`controlcatpredi.survival`

`plot.catpredi` *Plot the optimal cut points.*

Description

Plots the relationship between the predictor variable is aimed to categorise and the response variable based on a GAM model. Additionally, the optimal cut points obtained with the catpredi() function are drawn on the graph.

Usage

```
## S3 method for class 'catpredi'
plot(x, ...)
```

Arguments

`x`	An object of type catpredi.
`...`	Additional arguments to be passed on to other functions. Not yet implemented.

Value

This function returns the plot of the relationship between the predictor variable and the outcome.

Author(s)

Irantzu Barrio, Maria Xose Rodriguez-Alvarez and Inmaculada Arostegui.

References

I Barrio, I Arostegui, M.X Rodriguez-Alvarez and J.M Quintana (2017). A new approach to categorising continuous variables in prediction models: proposal and validation. *Statistical Methods in Medical Research*, 26(6), 2586-2602.

I Barrio, J Roca-Pardinas and I Arostegui (2021). Selecting the number of categories of the lymph node ratio in cancer research: A bootstrap-based hypothesis test. *Statistical Methods in Medical Research*, 30(3), 926-940.

See Also

`catpredi`

Examples

```
library(CatPredi)
## Not run:
set.seed(127)
#Simulate data
n = 100
#Predictor variable
xh <- rnorm(n, mean = 0, sd = 1)
xd <- rnorm(n, mean = 1.5, sd = 1)
x <- c(xh, xd)
#Response
y <- c(rep(0,n), rep(1,n))
# Data frame
df <- data.frame(y = y, x = x)

# Select optimal cut points using the AddFor algorithm
res.backaddfor <- catpredi(formula = y ~ 1, cat.var = "x", cat.points = 3,
                     data = df, method = "backaddfor", range = NULL, correct.AUC = FALSE)
# Plot
plot(res.backaddfor)

## End(Not run)
```

---

plot.catpredi.survival

Plot the optimal cut points.

Description

Plots the functional form of the predictor variable we want to categorise. Additionally, the optimal cut points obtained with the catpredi.survival() function are drawn on the graph.

Usage

```
## S3 method for class 'catpredi.survival'
plot(x, ...)
```

Arguments

`x`	An object of type catpredi.survival.
`...`	Additional arguments to be passed on to other functions. Not yet implemented.

Value

This function returns the plot of the relationship between the predictor variable and the outcome.

Author(s)

Irantzu Barrio and Maria Xose Rodriguez-Alvarez.

References

I Barrio, M.X Rodriguez-Alvarez, L Meira-Machado, C Esteban and I Arostegui (2017). Comparison of two discrimination indexes in the categorisation of continuous predictors in time-to-event studies. *SORT*, 41:73-92

See Also

`catpredi.survival`

Examples

```
library(CatPredi)
library(survival)
set.seed(123)
#Simulate data
n = 500
tauc = 1
X <- rnorm(n=n, mean=0, sd=2)
SurvT <- exp(2*X + rweibull(n = n, shape=1, scale = 1))   + rnorm(n, mean=0, sd=0.25)
# Censoring time
CensTime <- runif(n=n, min=0, max=tauc)
# Status
SurvS <- as.numeric(SurvT <= CensTime)
# Data frame
dat <- data.frame(X = X, SurvT = pmin(SurvT, CensTime), SurvS = SurvS)

# Select optimal cut points using the AddFor algorithm
res <- catpredi.survival (formula= Surv(SurvT,SurvS)~1, cat.var="X", cat.points = 2,
                      data = dat, method = "addfor", conc.index = "cindex", range = NULL,
                         correct.index = FALSE)
# Plot
plot(res)
```

`summary.catpredi` *Summary method for catpredi objects*

Description

Produces a summary of a catpredi object. The following are printed: the call to the catpredi() function; the estimated optimal cut points obtained with the method selected and the estimated AUC and bias corrected AUC (if the argument correct.AUC is TRUE) for the categorised variable.

Usage

```
## S3 method for class 'catpredi'
summary(object, digits = 4, ...)
```

Arguments

`object`	An object of class catpredi as produced by catpredi()
`digits`	.
`...`	Further arguments passed to or from other methods.

Value

Returns an object of class "summary.catpredi" with the same components as the catpredi function (see `catpredi`). plus:

fit.gam fitted model according to the model specified in the call, based on the function `gam` of the package **mgcv**.

Author(s)

Irantzu Barrio, Maria Xose Rodriguez-Alvarez and Inmaculada Arostegui.

References

I Barrio, I Arostegui, M.X Rodriguez-Alvarez and J.M Quintana (2017). A new approach to categorising continuous variables in prediction models: proposal and validation. *Statistical Methods in Medical Research*, 26(6), 2586-2602.

I Barrio, J Roca-Pardinas and I Arostegui (2021). Selecting the number of categories of the lymph node ratio in cancer research: A bootstrap-based hypothesis test. *Statistical Methods in Medical Research*, 30(3), 926-940.

See Also

`catpredi`

Examples

```
library(CatPredi)
set.seed(127)
#Simulate data
n = 200
#Predictor variable
xh <- rnorm(n, mean = 0, sd = 1)
xd <- rnorm(n, mean = 1.5, sd = 1)
x <- c(xh, xd)
#Response
y <- c(rep(0,n), rep(1,n))
#Covariate
zh <- rnorm(n, mean=1.5, sd=1)
zd <- rnorm(n, mean=1, sd=1)
z <- c(zh, zd)
# Data frame
df <- data.frame(y = y, x = x, z = z)

# Select optimal cut points using the AddFor algorithm
```

```
res.backaddfor <- catpredi(formula = y ~ z, cat.var = "x", cat.points = 2,
                          data = df, method = "backaddfor", range=NULL, correct.AUC=FALSE)
# Summary
summary(res.backaddfor)
```

---

summary.catpredi.survival

Summary method for objects of class "catpredi.survival"

Description

Produces a summary of a "catpredi.survival" object. The following are printed: the call to the catpredi.survival() function; the estimated optimal cut points obtained with the method and concordance probability estimator selected and the estimated and bias corrected concordance probability for the categorised variable (whenever the argument correct.index is set to TRUE).

Usage

```
## S3 method for class 'catpredi.survival'
summary(object, digits = 4, ...)
```

Arguments

`object`	An object of class "catpredi.survival" as produced by catpredi.survival()
`digits`	.
`...`	Further arguments passed to or from other methods.

Value

Returns an object of class "summary.catpredi.survival" with the same components as the catpredi.survival function (see `catpredi.survival`).

Author(s)

Irantzu Barrio and Maria Xose Rodriguez-Alvarez.

References

I Barrio, M.X Rodriguez-Alvarez, L Meira-Machado, C Esteban and I Arostegui (2017). Comparison of two discrimination indexes in the categorisation of continuous predictors in time-to-event studies. *SORT*, 41:73-92

See Also

`catpredi.survival`

Examples

```
library(CatPredi)
library(survival)
set.seed(123)
#Simulate data
n = 500
tauc = 1
X <- rnorm(n=n, mean=0, sd=2)
SurvT <- exp(2*X + rweibull(n = n, shape=1, scale = 1))   + rnorm(n, mean=0, sd=0.25)
# Censoring time
CensTime <- runif(n=n, min=0, max=tauc)
# Status
SurvS <- as.numeric(SurvT <= CensTime)
# Data frame
dat <- data.frame(X = X, SurvT = pmin(SurvT, CensTime), SurvS = SurvS)

# Select optimal cut points using the AddFor algorithm
res <- catpredi.survival (formula= Surv(SurvT,SurvS)~1, cat.var="X", cat.points = 2,
                       data = dat, method = "addfor", conc.index = "cindex", range = NULL,
                           correct.index = FALSE)
# Summary
summary(res)
```

Index