Package 'SplitWise'

April 12, 2025

Type Package

Title Hybrid Stepwise Regression with Single-Split Dummy Encoding

Version 0.1.0

Author Marcell T. Kurbucz [aut, cre], Nikolaos Tzivanakis [aut], Nilufer Sari Aslam [aut], Adam Sykulski [aut]

Maintainer Marcell T. Kurbucz <m.kurbucz@ucl.ac.uk>

Description Implements a hybrid regression approach that allows numeric variables to be transformed into either single-split (0/1) dummy variables or retained as continuous predictors. This transformation is followed by stepwise selection to identify the most significant variables. Additionally, the package offers an 'iterative' mode designed to detect partial synergies among variables, enhancing model performance.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 3.5.0)

Imports rpart, stats

LazyData true

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Contents

splitwise *SplitWise Regression*

Description

Transforms each numeric variable into either a single-split dummy or keeps it linear, then runs `stats::step()` for stepwise selection. The user can choose a simpler univariate transformation or an iterative approach.

Usage

```
splitwise(
  formula,
  data,
  transformation_mode = c("univariate", "iterative"),
  direction = c("backward", "forward", "both"),
  minsplit = 5,
  criterion = c("AIC", "BIC"),
  exclude_vars = NULL,
  verbose = FALSE,
  trace = 1,
  steps = 1000,
  k = 2,
  ...
)
```

Arguments

`formula`	A formula specifying the response and (initial) predictors, e.g. `mpg ~ .`.
`data`	A data frame containing the variables used in `formula`.
`transformation_mode`	Either `"univariate"` or `"iterative"`.
`direction`	Stepwise direction: `"backward"`, `"forward"`, or `"both"`.
`minsplit`	Minimum number of observations in a node to consider splitting. Default = 5.
`criterion`	Either `"AIC"` or `"BIC"`. Default = `"AIC"`. **Note**: If you choose `"BIC"`, you typically want `k = log(nrow(data))` in stepwise.
`exclude_vars`	A character vector naming variables that should be forced to remain linear (i.e., no dummy splits allowed). Default = `NULL`.
`verbose`	Logical; if `TRUE`, prints debug info in transformation steps. Default = `FALSE`.
`trace`	If positive, `step()` prints info at each step. Default = 1.
`steps`	Maximum number of steps for `step()`. Default = 1000.
`k`	Penalty multiple for the number of degrees of freedom (used by `step()`). E.g. 2 for AIC, `log(n)` for BIC. Default = 2.
`...`	Additional arguments passed to `step()` or the iterative function.

Value

An S3 object of class `c("splitwise_lm", "lm")`, storing:

`splitwise_info`	List containing transformation decisions, final data, and call.

Examples

```
# Load the mtcars dataset
data(mtcars)

# Univariate transformations (AIC-based, backward stepwise)
model_uni <- splitwise(
  mpg ~ .,
  data               = mtcars,
  transformation_mode = "univariate",
  direction           = "backward",
  trace               = 0
)
summary(model_uni)

# Iterative approach (BIC-based, forward stepwise)
# Note: typically set k = log(nrow(mtcars)) for BIC in step().
model_iter <- splitwise(
  mpg ~ .,
  data               = mtcars,
  transformation_mode = "iterative",
  direction           = "forward",
  criterion           = "BIC",
  k                   = log(nrow(mtcars)),
  trace               = 0
)
summary(model_iter)
```