Package 'SplitWise'

July 21, 2025

Type Package

Title 'SplitWise': Hybrid Stepwise Regression with Single-Split Dummy Encoding

Version 1.0.0

Description Implements 'SplitWise', a hybrid regression approach that transforms numeric variables into either single-split (0/1) dummy variables or retains them as continuous predictors. The transformation is followed by stepwise selection to identify the most relevant variables. The default 'iterative' mode adaptively explores partial synergies among variables to enhance model performance, while an alternative 'univariate' mode applies simpler transformations independently to each predictor. For details, see Kurbucz et al. (2025) <doi:10.48550/arXiv.2505.15423>.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 3.5.0)

Imports rpart, stats

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

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>

Repository CRAN

Date/Publication 2025-05-28 16:00:02 UTC

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

## S3 method for class 'splitwise_lm'
print(x, ...)

## S3 method for class 'splitwise_lm'
summary(object, ...)
```

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 `"iterative"` or `"univariate"`. Default = `"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 summary.lm.
x	A "splitwise_lm" object returned by splitwise.
object	A "splitwise_lm" object returned by splitwise.

Value

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

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

Functions

- print(splitwise_lm): Prints a summary of the splitwise_lm object.
- summary(splitwise_lm): Provides a detailed summary, including how dummies were created.

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)
```

Index