Package: kronxNBC (via r-universe)

October 19, 2026

Type Package

Title Clock of Regimes Naive Bayes Classifier (Student-t)

Version 0.1.1

Description Computes and fits a heavy-tailed Student-t Naive Bayes classifier for non-stationary financial market regime analysis (Clock of Regimes, COR). The core innovation is a profile grid search over the degrees-of-freedom parameter nu that prevents numerical underflow and structural classification failures when identifying fat-tailed Stress regimes. Provides S3 methods for fitting, prediction, summarising, plotting, and parameter extraction.

License MIT + file LICENSE

Encoding UTF-8

Imports stats, graphics, utils, naivebayes

Suggests zoo, testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Oscar Linares [aut, cre]

Maintainer Oscar Linares <olinares@umich.edu>

Repository https://oismrg-saamlab.r-universe.dev

Date/Publication 2026-05-30 13:50:02 UTC

RemoteUrl https://github.com/cran/kronxNBC

RemoteRef HEAD

RemoteSha a9df2955ae18665e99d9facd373d28c70d801261

Contents

coef.student_t_naive_bayes 2
plot.student_t_naive_bayes 3
predict.student_t_naive_bayes 4
print.student_t_naive_bayes 5
student_t_naive_bayes 6
summary.student_t_naive_bayes 7
tables 8

Index **10**

`coef.student_t_naive_bayes` *Coefficient extraction for student_t_naive_bayes*

Description

Returns a data frame of fitted Student-t parameters (μ, σ, ν) organised by feature (rows) and class (column groups).

Usage

```
## S3 method for class 'student_t_naive_bayes'
coef(object, ...)
```

Arguments

`object`	A fitted `"student_t_naive_bayes"` object.
`...`	Additional arguments (currently unused).

Value

A data frame with p rows (features) and $3K$ columns, named `<class>:mu`, `<class>:sd`, `<class>:nu` for each of the K classes.

See Also

`student_t_naive_bayes`

Examples

```
set.seed(7)
X <- matrix(rnorm(120), 60, 2, dimnames = list(NULL, c("f1", "f2")))
y <- factor(rep(c("A", "B", "C"), 20))
m <- student_t_naive_bayes(X, y)
coef(m)
```

plot.student_t_naive_bayes *Plot method for student_t_naive_bayes*

Description

Draws per-feature Student-t density curves, one curve per class, for a fitted `"student_t_naive_bayes"` model.

Usage

```
## S3 method for class 'student_t_naive_bayes'
plot(
  x,
  which = NULL,
  ask = FALSE,
  legend = TRUE,
  legend.box = FALSE,
  arg.num = list(),
  prob = c("marginal", "conditional"),
  ...
)
```

Arguments

`x`	A fitted `"student_t_naive_bayes"` object.
`which`	Integer or character vector selecting which features to plot. `NULL` (default) plots all features.
`ask`	Logical. If `TRUE` the user is prompted before each plot. Default: `FALSE`.
`legend`	Logical. If `TRUE` (default) a legend is added to each plot.
`legend.box`	Logical. If `TRUE` the legend is drawn with a box. Default: `FALSE`.
`arg.num`	Named list of graphical parameters (e.g. `col`, `lty`) applied to the density lines. Defaults: sequential integer colours and solid lines.
`prob`	Character string: `"marginal"` (default) scales each density by the class prior; `"conditional"` plots the raw conditional density.
`...`	Additional graphical arguments passed to `plot`.

Value

Invisibly returns `NULL`.

See Also

`student_t_naive_bayes`

`predict.student_t_naive_bayes` *Predict method for student_t_naive_bayes*

Description

Computes MAP class assignments or posterior probabilities for new observations using a fitted `student_t_naive_bayes` model.

Usage

```
## S3 method for class 'student_t_naive_bayes'
predict(
  object,
  newdata = NULL,
  type = c("class", "prob"),
  threshold = 0.001,
  eps = 0,
  ...
)
```

Arguments

`object`	A fitted `"student_t_naive_bayes"` object.
`newdata`	Optional numeric matrix of new observations with the same named columns as the training matrix. If `NULL` the training data stored in `object` are used.
`type`	Character string: `"class"` (default) returns a factor of predicted class labels; `"prob"` returns a numeric matrix of posterior probabilities (rows = observations, columns = classes).
`threshold`	Minimum log-density floor applied after the `eps` check. Prevents $-\infty$ contributions from absorbing the posterior. Default: `0.001`.
`eps`	Densities at or below this value are replaced by `threshold`. Set to `0` (default) to use the machine minimum.
`...`	Additional arguments (currently unused).

Value

`type = "class"` A `factor` of length `nrow(newdata)` with levels matching the training classes.

`type = "prob"` A numeric matrix with `nrow(newdata)` rows and one column per class, containing softmax-normalised posterior probabilities.

See Also

`student_t_naive_bayes`

Examples

```
set.seed(1)
X <- matrix(rnorm(100), 50, 2, dimnames = list(NULL, c("f1", "f2")))
y <- factor(rep(c("A", "B"), 25))
m <- student_t_naive_bayes(X, y)
predict(m, type = "class")
predict(m, type = "prob")
```

`print.student_t_naive_bayes` *Print method for student_t_naive_bayes*

Description

Prints a concise summary of the model, including the call, prior probabilities, and the first five parameter tables.

Usage

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

Arguments

`x`	A fitted `"student_t_naive_bayes"` object.
`...`	Additional arguments (currently unused).

Value

Invisibly returns `x`.

See Also

`student_t_naive_bayes`, `summary.student_t_naive_bayes`

`student_t_naive_bayes` *Student-t Naive Bayes Classifier*

Description

Fits a Naive Bayes classifier where the per-class, per-feature likelihood is a scaled Student-t distribution. The degrees-of-freedom parameter ν is selected for every (class, feature) pair by a profile log-likelihood grid search over `nu_grid`, making the model robust to the fat-tailed returns that characterise financial Stress regimes.

Usage

```
student_t_naive_bayes(x, y, prior = NULL, nu_grid = c(3:30, 40, 60, 100), ...)
```

Arguments

`x`	A numeric matrix of predictors with named columns. Each column is one feature; each row is one observation.
`y`	A factor, character, or logical vector of class labels with length equal to `nrow(x)`. Must contain at least two distinct classes and at least two observations per class.
`prior`	Optional named numeric vector of prior class probabilities. Length must equal the number of class levels. Defaults to empirical class frequencies. Supplied values are normalised to sum to one.
`nu_grid`	Numeric vector of candidate degrees-of-freedom values used by the profile grid search. All values must be strictly greater than 2 (finite-variance requirement). Default: `c(3:30, 40, 60, 100)`.
	• Values near 3 capture extreme kurtosis typical of Stress regimes. • Values near 30+ approximate Gaussian behaviour for Calm regimes. • Including 60 and 100 provides a near-Gaussian safety net without imposing normality.
`...`	Additional arguments (currently unused).

Details

Classification rule (MAP)

Under the Naive Bayes conditional-independence assumption the posterior log-odds reduce to the Maximum A Posteriori rule:

$$\hat{k} = \arg\max_{k \in \{1,\ldots,K\}} \Big[\log \pi_k + \sum_{j=1}^{p} \log f_t\Big(x_j \,\Big|\, \mu_{kj},\, \sigma_{kj},\, \nu_{kj} \Big) \Big]$$

where the scaled Student-t density is

$$f_t(x \mid \mu, \sigma, \nu) = \frac{1}{\sigma}\, f_{t_\nu}\!\left(\frac{x - \mu}{\sigma} \right)$$

and π_k is the prior probability of class k.

Degrees-of-freedom grid search

For each (class k, feature j) pair the algorithm runs one IRLS step at every candidate $\nu \in$ `nu_grid` and retains the triplet $(\hat{\mu}, \hat{\sigma}, \hat{\nu})$ that maximises the profile log-likelihood. This discrete search avoids the numerical instability of continuous ν optimisation and is the mechanism that prevents log-likelihood underflow when scoring crisis observations in the Stress regime.

Value

An S3 object of class `"student_t_naive_bayes"` with components:

`data` List with elements `x` (training matrix) and `y` (training labels).

`levels` Character vector of class levels.

`params` Named list with $K \times p$ matrices `mu`, `sd`, and `nu` (K = classes, p = features).

`prior` Named numeric vector of prior probabilities.

`nu_grid` The `nu_grid` vector used during fitting.

`call` The matched call.

See Also

`predict.student_t_naive_bayes`, `tables.student_t_naive_bayes`, `coef.student_t_naive_bayes`

Examples

```
set.seed(42)
n  <- 150
y  <- factor(rep(c("Calm", "Stress", "Trend"), each = n / 3))
X  <- matrix(
  c(rnorm(50, 0, 1), rt(50, df = 4), rnorm(50, 1, 0.5),
    rnorm(50, 0, 1), rt(50, df = 4), rnorm(50, 1, 0.5)),
  nrow = n, ncol = 2,
  dimnames = list(NULL, c("ret", "vol"))
)
model <- student_t_naive_bayes(X, y)
print(model)
```

---

summary.student_t_naive_bayes

Summary method for student_t_naive_bayes

Description

Prints a high-level summary of a fitted `"student_t_naive_bayes"` model: sample size, feature count, ν grid range, and prior probabilities.

Usage

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

Arguments

object	A fitted "student_t_naive_bayes" object.
...	Additional arguments (currently unused).

Value

Invisibly returns object.

See Also

student_t_naive_bayes, print.student_t_naive_bayes

---

tables *Parameter tables for Naive Bayes classifiers*

Description

S3 generic that returns per-feature parameter tables for a fitted Naive Bayes model. For "student_t_naive_bayes" objects each table contains the fitted μ, σ, and ν parameters. For all other model classes the call is forwarded to naivebayes::tables().

Usage

```
tables(object, which = NULL, ...)

## S3 method for class 'student_t_naive_bayes'
tables(object, which = NULL, ...)

## Default S3 method:
tables(object, which = NULL, ...)
```

Arguments

object	A fitted Naive Bayes model. Supported classes: "student_t_naive_bayes" (this package) and any class accepted by tables.
which	Integer or character vector selecting which feature tables to return. NULL (default) returns all features.
...	Additional arguments passed to the method.

Value

A "naive_bayes_tables" object: a named list with one element per selected feature. For "student_t_naive_bayes" objects each element is a table with rows mu, sd, nu and one column per class level.

See Also

tables, student_t_naive_bayes

Examples

```
set.seed(3)
X <- matrix(rnorm(120), 60, 2, dimnames = list(NULL, c("ret", "vol")))
y <- factor(rep(c("Calm", "Stress", "Trend"), 20))
m <- student_t_naive_bayes(X, y)
tables(m)
tables(m, which = "ret")
tables(m, which = 2L)
```

Index

coef.student_t_naive_bayes, 2, *7*

factor, *4*

plot, *3*
plot.student_t_naive_bayes, 3
predict.student_t_naive_bayes, 4, *7*
print.student_t_naive_bayes, 5, *8*

student_t_naive_bayes, *2–5*, 6, *8, 9*
summary.student_t_naive_bayes, *5*, 7

table, *8*
tables, 8, *8, 9*
tables.student_t_naive_bayes, *7*