

Package: nbsurv (via r-universe)

October 8, 2026

Type Package

Title Conditional Naive Bayes Survival Modelling for Right-Censored Data

Version 0.5.1

Description Fits conditional naive Bayes survival models for right-censored outcomes using inverse-probability of censoring weighting. The package provides model fitting, prediction, resampling-based evaluation, cross-validation, hyper-parameter tuning, and permutation variable importance utilities for horizon-specific survival prediction. The model is the censored naive Bayes classifier of Wolfson et al. (2015) <doi:10.1002/sim.6526>, which combines the marginal Kaplan-Meier survivor function with horizon-specific class-conditional covariate densities and inverse-probability-of-censoring weights. Resampling evaluation uses the inverse-probability-of-censoring-weighted Brier score of Gerds and Schumacher (2006) <doi:10.1002/bimj.200610301>.

License GPL-3

Encoding UTF-8

Depends R (>= 4.1.0)

Imports graphics, survival, stats, utils

Suggests knitr, pkgload, pec, quarto, ranger, rmarkdown, testthat

VignetteBuilder knitr

URL <https://github.com/ielbadisy/nbsurv>

BugReports <https://github.com/ielbadisy/nbsurv/issues>

Config/testthat/edition 3

RoxygenNote 7.3.2

Repository <https://ielbadisy.r-universe.dev>

Date/Publication 2026-08-31 14:41:40 UTC

RemoteUrl <https://github.com/ielbadisy/nbsurv>

RemoteRef HEAD

RemoteSha 5dc8f04338d75edebf494f7bf4382726716e6c94

Contents

calibration_plot_nbsurv	2
cv_nbsurv	3
nbsurv	5
nbsurv-class	6
plot.varimp_nbsurv	7
predict.nbsurv	8
varimp_nbsurv	9
Index	11

calibration_plot_nbsurv
<i>Calibration Plot for a nbsurv Model</i>

Description

Assesses calibration of a nbsurv model at a single prediction horizon by grouping observations into quantile bins of predicted event probability and comparing the group mean prediction against the Kaplan-Meier observed event probability within each bin.

Usage

```
calibration_plot_nbsurv(object, newdata, horizon, n_groups = 10, ...)
```

Arguments

object	A fitted "nbsurv" model.
newdata	A data frame containing the outcome and predictors.
horizon	A single numeric prediction horizon.
n_groups	Number of quantile-based calibration groups.
...	Additional graphical arguments passed to plot().

Value

Invisibly returns a data frame with columns mean_pred (mean predicted event probability per group), observed (observed event probability from Kaplan-Meier), and n (group size). The calibration plot is drawn as a side effect.

Examples

```
library(survival)

lung <- stats::na.omit(lung)
lung$status <- as.integer(lung$status == 2)
fit <- nbsurv(Surv(time, status) ~ age + sex, data = lung)
calibration_plot_nbsurv(fit, newdata = lung, horizon = 300)
```

Description

Provides resampling-based utilities for evaluating, cross-validating, and tuning nbsurv survival models using horizon-specific Brier score, integrated Brier score (IBS), and concordance.

The integrated Brier score summarises calibration over a range of horizons:

$$\text{IBS} = \frac{1}{\tau - t_1} \int_{t_1}^{\tau} \text{BS}(t) dt$$

approximated by the trapezoidal rule over the supplied times.

Usage

```
cv_nbsurv(
  formula,
  data,
  folds = 5,
  times = NULL,
  seed = NULL,
  scale = TRUE,
  laplace = 1,
  min_sd = 0.05,
  time_grid = NULL,
  eps = 1e-06,
  cov_structure = c("diagonal", "full"),
  shrinkage = 0.2,
  time_smooth = FALSE,
  bandwidth = NULL
)
```

```
evaluate_nbsurv(object, newdata, times, metrics = c("brier", "concordance"), ibs = FALSE)
```

```
tune_nbsurv(
  formula,
  data,
  param_grid,
  folds = 5,
  times = NULL,
  metric = c("brier", "concordance"),
  maximize = NULL,
  seed = NULL,
  eps = 1e-06
)
```

Arguments

formula	A model formula with a <code>survival::Surv()</code> response.
data	A data frame containing outcome and predictors.
folds	Number of cross-validation folds.
times	Prediction horizons to evaluate. If <code>NULL</code> , event-time quantiles are used.
seed	Optional random seed for reproducible fold assignment.
scale	Logical; if <code>TRUE</code> , continuous predictors are standardized.
laplace	Laplace smoothing constant for categorical likelihoods.
min_sd	Lower bound used for Gaussian standard deviations.
time_grid	Optional prediction grid stored with each fitted model.
eps	Small numerical constant used in probability clipping and IPCW calculations.
cov_structure	Passed to each fold's <code>nbsurv</code> fit; see there.
shrinkage	Passed to each fold's <code>nbsurv</code> fit; see there.
time_smooth	Passed to each fold's <code>nbsurv</code> fit; see there.
bandwidth	Passed to each fold's <code>nbsurv</code> fit; see there.
object	A fitted "nbsurv" model.
newdata	Evaluation data containing the same outcome and predictors as the training formula.
metrics	Performance metrics to compute. One or both of "brier" and "concordance".
ibs	Logical; if <code>TRUE</code> and "brier" is in metrics, the integrated Brier score (trapezoidal rule over times) is attached as <code>attr(result, "ibs")</code> .
param_grid	A data frame with columns <code>scale</code> , <code>laplace</code> , <code>min_sd</code> , and <code>time_grid</code> .
metric	The metric used to choose the best hyper-parameter setting.
maximize	Logical; if <code>TRUE</code> , larger metric values are preferred.

Value

`evaluate_nbsurv()` returns a data frame of horizon-specific metrics. When `ibs = TRUE` the integrated Brier score is attached via `attr(result, "ibs")`. `cv_nbsurv()` returns a list with fold-level and mean cross-validation summaries. `tune_nbsurv()` returns a list with all candidate results and the best row.

Examples

```
library(survival)

lung <- stats::na.omit(lung)
lung$status <- as.integer(lung$status == 2)
fit <- nbsurv(Surv(time, status) ~ age + sex, data = lung)
res <- evaluate_nbsurv(fit, newdata = lung[1:50, ], times = c(100, 200, 400), ibs = TRUE)
res
attr(res, "ibs")

cv_nbsurv(Surv(time, status) ~ age + sex, data = lung, folds = 3, times = c(100, 200))
```

nbsurv

*Fit a Conditional Naive Bayes Survival Model***Description**

Fits a conditional naive Bayes model for right-censored survival data. The model combines the marginal Kaplan-Meier survival estimate with horizon-specific likelihood terms estimated among subjects who fail before the horizon and subjects known to survive beyond it.

Usage

```
nbsurv(
  formula,
  data,
  scale = TRUE,
  laplace = 1,
  min_sd = 0.05,
  time_grid = NULL,
  eps = 1e-06,
  cov_structure = c("diagonal", "full"),
  shrinkage = 0.2,
  time_smooth = FALSE,
  bandwidth = NULL
)
```

Arguments

<code>formula</code>	A model formula with a <code>survival::Surv()</code> response.
<code>data</code>	A data frame containing the outcome and predictors.
<code>scale</code>	Logical; if TRUE, continuous predictors are standardized.
<code>laplace</code>	Laplace smoothing constant for categorical likelihoods.
<code>min_sd</code>	Lower bound used for Gaussian standard deviations.
<code>time_grid</code>	Optional numeric vector of prediction horizons stored with the fit.
<code>eps</code>	Small numerical constant used to clip probabilities away from 0 and 1.
<code>cov_structure</code>	"diagonal" (default): continuous predictors are modeled as conditionally independent (the classic naive Bayes assumption), each with its own class-conditional Gaussian. "full": continuous predictors are modeled jointly as a single multivariate Gaussian per class (with covariance shrunk toward diagonal by <code>shrinkage</code>), relaxing the independence assumption between them. Categorical predictors are always treated as conditionally independent regardless of this setting. Requires 2+ continuous predictors to have any effect; with only one, "full" is numerically identical to "diagonal". Empirically improves the IPCW Brier score (calibration) when continuous predictors are correlated, with a smaller and less consistent effect on concordance (ranking) - see <code>vignette("nbsurv-workflow")</code> .

shrinkage	Shrinkage weight toward the diagonal covariance, in <code>'[0, 1]'</code> , used only when <code>cov_structure = "full"</code> . <code>'0'</code> uses the raw sample covariance (more variance, especially at small <code>'n'</code>); <code>'1'</code> is equivalent to <code>"diagonal"</code> . Default <code>'0.2'</code> .
time_smooth	Logical, default FALSE. If TRUE, class-conditional statistics (means, covariances, categorical probabilities) are estimated once at every point of <code>time_grid</code> at fit time, then combined across horizons via Nadaraya-Watson kernel regression (Gaussian kernel, width <code>bandwidth</code>) when <code>predict()</code> is called at a given horizon, instead of being re-estimated from scratch, independently, at each requested horizon. This borrows statistical strength from neighboring horizons, reducing estimation variance where one of the two horizon-defined classes (subjects who already failed vs. subjects known to survive past it) is sparse - typically near the earliest or latest horizons requested - while still allowing genuinely time-varying effects (unlike a proportional-hazards model, which shares a single coefficient across all of time). A convex (weight-normalized) combination of the per-grid-point statistics is used, so smoothed covariance matrices stay positive semi-definite and smoothed categorical probabilities stay valid automatically, with no extra correction needed. Increases fit time (every grid point's statistics are computed once, up front) but not predict time. Empirically (see <code>'NEWS.md'</code>) trades ranking robustness (concordance) for calibration at extreme horizons as <code>'bandwidth'</code> widens, and vice versa as it narrows - tune <code>'bandwidth'</code> via tune_nbsurv rather than relying on the default for a specific dataset.
bandwidth	Kernel bandwidth (in the same units as time), used only when <code>time_smooth = TRUE</code> . Defaults to one quarter of the fitted <code>time_grid</code> 's range when NULL.

Value

An object of class `"nbsurv"`.

Examples

```
library(survival)

lung <- stats::na.omit(lung)
lung$status <- as.integer(lung$status == 2)
fit <- nbsurv(
  Surv(time, status) ~ age + sex,
  data = lung
)

predict(fit, newdata = lung[1:3, ], times = c(100, 200, 300))
```

nbsurv-class

Conditional Naive Bayes Survival Model Object

Description

`nbsurv` objects are created by [nbsurv](#) and store the training data, outcome, feature types, scaling information, and the Kaplan-Meier estimates required for prediction.

Usage

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

## S3 method for class 'nbsurv'
plot(x, times = NULL, n_curves = 5, ...)
```

Arguments

x	A fitted "nbsurv" model object.
times	Optional vector of horizons used for plotting.
n_curves	Number of training-set curves to display in the plot method.
...	Additional arguments passed to the underlying plotting functions.

Value

print.nbsurv is called for its side effect of printing a concise summary of the fitted model (formula, number of training rows, predictors, prediction grid size, covariance structure, and time-smoothing settings) to the console; it returns its argument x invisibly.

plot.nbsurv is called for its side effect of drawing predicted survival curves for the first n_curves training observations over times; it returns, invisibly, the matrix of predicted survival probabilities used for the plot (one row per displayed observation, one column per element of times).

plot.varimp_nbsurv	<i>Visualization Helpers for nbsurv Results</i>
--------------------	---

Description

Base-graphics plot methods for the result objects returned by [varimp_nbsurv](#), [cv_nbsurv](#), and [tune_nbsurv](#).

plot.varimp_nbsurv() draws a horizontal bar chart of permutation importance, sorted by increasing importance (largest bar at top).

plot.cv_nbsurv() draws the mean cross-validation metric against the evaluation horizons.

plot.tune_nbsurv() draws the mean tuning metric for every candidate row of param_grid, highlighting the selected best candidate in red.

Usage

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

## S3 method for class 'cv_nbsurv'
plot(x, metric = c("brier", "concordance"), ...)

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

Arguments

<code>x</code>	A "varimp_nbsurv", "cv_nbsurv", or "tune_nbsurv" object.
<code>metric</code>	For <code>plot.cv_nbsurv()</code> : which mean cross-validation metric to plot against time. One of "brier" or "concordance".
<code>...</code>	Additional arguments passed to the underlying base graphics function (barplot , matplot , or plot).

Value

The input object, invisibly. Called for its plotting side effect.

Examples

```
library(survival)

lung <- stats::na.omit(lung)
lung$status <- as.integer(lung$status == 2)
fit <- nbsurv(Surv(time, status) ~ age + sex + ph.ecog, data = lung)

vi <- varimp_nbsurv(fit, newdata = lung, times = c(100, 200, 400), n_repeats = 5, seed = 1)
plot(vi)

cvfit <- cv_nbsurv(Surv(time, status) ~ age + sex, data = lung, folds = 3, times = c(100, 200, 300))
plot(cvfit)

grid <- expand.grid(scale = TRUE, laplace = c(0.5, 1, 2), min_sd = 0.05)
tf <- tune_nbsurv(
  Surv(time, status) ~ age + sex, data = lung,
  param_grid = grid, folds = 3, times = c(100, 200)
)
plot(tf)
```

predict.nbsurv

Predict Survival Probabilities from a nbsurv Model

Description

Computes horizon-specific survival or event probabilities from a fitted conditional naive Bayes survival model. The returned survival curves are forced to be non-increasing across increasing time horizons.

Usage

```
## S3 method for class 'nbsurv'
predict(object, newdata, times, type = c("survival", "event"), ...)

predictSurvProb(object, ...)
```

```
## S3 method for class 'nbsurv'
predictSurvProb(object, newdata, times, ...)
```

Arguments

object	A fitted "nbsurv" model.
newdata	A data frame of predictor values.
times	Numeric vector of prediction horizons.
type	Either "survival" for survival probabilities or "event" for event probabilities.
...	Unused additional arguments.

Value

A matrix with one row per observation in newdata and one column per prediction horizon.

Examples

```
library(survival)

lung <- stats::na.omit(lung)
lung$status <- as.integer(lung$status == 2)
fit <- nbsurv(Surv(time, status) ~ age + sex, data = lung)
predict(fit, newdata = lung[1:2, ], times = c(100, 200))
predict(fit, newdata = lung[1:2, ], times = c(100, 200), type = "event")
```

varimp_nbsurv

Permutation Variable Importance for nbsurv Models

Description

Computes permutation-based variable importance: for each predictor, its values are independently shuffled `n_repeats` times, [evaluate_nbsurv](#) is recomputed on the permuted data, and importance is the resulting degradation in the chosen metric, averaged over times and repeats. Predictors the model relies on more heavily degrade performance more when permuted.

Usage

```
varimp_nbsurv(
  object,
  newdata,
  times,
  metric = c("brier", "concordance"),
  n_repeats = 10,
  seed = NULL
)
```

Arguments

object	A fitted "nbsurv" model.
newdata	Evaluation data containing the same outcome and predictors as the training formula.
times	Prediction horizons averaged over when computing the metric.
metric	The metric to degrade under permutation. One of "brier" (higher = worse) or "concordance" (higher = better).
n_repeats	Number of independent permutations per feature; importance is the mean degradation across repeats.
seed	Optional random seed for reproducible permutations.

Value

A data frame (class "varimp_nbsurv") with one row per predictor, sorted by decreasing importance, with columns feature, baseline (the unpermuted metric, averaged over times), permuted (the metric after permuting that feature, averaged over times and n_repeats), and importance (the degradation: permuted - baseline for "brier", baseline - permuted for "concordance", so higher is always more important).

Examples

```
library(survival)

lung <- stats::na.omit(lung)
lung$status <- as.integer(lung$status == 2)
fit <- nbsurv(Surv(time, status) ~ age + sex + ph.ecog, data = lung)
varimp_nbsurv(fit, newdata = lung, times = c(100, 200, 400), n_repeats = 5, seed = 1)
```

Index

barplot, [8](#)

calibration_plot_nbsurv, [2](#)

cv_nbsurv, [3](#), [7](#)

evaluate_nbsurv, [9](#)

evaluate_nbsurv (cv_nbsurv), [3](#)

matplot, [8](#)

nbsurv, [4](#), [5](#), [6](#)

nbsurv-class, [6](#)

plot, [8](#)

plot.cv_nbsurv (plot.varimp_nbsurv), [7](#)

plot.nbsurv (nbsurv-class), [6](#)

plot.tune_nbsurv (plot.varimp_nbsurv), [7](#)

plot.varimp_nbsurv, [7](#)

predict.nbsurv, [8](#)

predictSurvProb (predict.nbsurv), [8](#)

print.nbsurv (nbsurv-class), [6](#)

print.varimp_nbsurv (varimp_nbsurv), [9](#)

tune_nbsurv, [6](#), [7](#)

tune_nbsurv (cv_nbsurv), [3](#)

varimp_nbsurv, [7](#), [9](#)