

# Package: survivalisttoo (via r-universe)

June 5, 2026

**Type** Package

**Title** Model-Agnostic Survival Analysis

**Version** 0.1.0

**Date** 2026-04-29

**Description** Model-agnostic survival analysis: using any Machine learning algorithm for doing survival analysis.

**License** MIT + file LICENSE

**Imports** Rcpp

**Suggests** survival, glmnet, mlS3, knitr, rmarkdown, caret, elasticnet, fastICA

**LinkingTo** Rcpp

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**Encoding** UTF-8

**Repository** <https://techtonique.r-universe.dev>

**Date/Publication** 2026-05-02 19:24:05 UTC

**RemoteUrl** <https://github.com/Techtonique/survivalisttoo>

**RemoteRef** HEAD

**RemoteSha** 73c9183f8b9d521212575bf1ee6389defbdd118f

## Contents

survivalisttoo-package . . . . .	2
cox_gradient_boost . . . . .	2
predict.CoxGradientBoost . . . . .	3
print.CoxGradientBoost . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

survivalisttoo-package

*Model-Agnostic Survival Analysis*

---

### Description

Model-agnostic survival analysis: using any Machine learning algorithm for doing survival analysis.

### Package Content

Index: This package was not yet installed at build time.

### Maintainer

T. Moudiki <thierry.moudiki@gmail.com>

### Author(s)

T. Moudiki [aut, cre]

---

cox\_gradient\_boost

*Cox Gradient Boosting Model*

---

### Description

Fit a model-agnostic Cox-gradient boosting survival model

### Usage

```
cox_gradient_boost(  
  X,  
  time,  
  event,  
  regr_fun,  
  M = 100L,  
  nu = 0.1,  
  show_progress = TRUE,  
  ...  
)
```

**Arguments**

X	numeric matrix of covariates (n x p)
time	numeric vector of observed times
event	integer/logical vector (1 = event, 0 = censored)
regr_fun	function(X, y, ...) -> model with predict() method
M	number of boosting iterations (default 100)
nu	learning rate / shrinkage (default 0.1)
show_progress	logical; print a progress bar? (default TRUE)
...	extra arguments forwarded to regr_fun at every iteration

**Value**

object of class "CoxGradientBoost"

**Examples**

```
require(glmnet)
require(survival)

data(ovarian)

set.seed(42)
idx_train <- sample(nrow(ovarian), floor(0.75 * nrow(ovarian)))
df <- ovarian
train <- df[idx_train, ]; test <- df[-idx_train, ]

regr_lm <- function(X, y, ...) lm(y ~ ., data = data.frame(X, y = y))

fit_boost_lm <- survivalisttoo::cox_gradient_boost(train, train$futime,
train$fustat, regr_lm)

y_test <- Surv(test$futime, test$fustat)
(ci_blm <- glmnet::Cindex(predict(fit_boost_lm, test), y_test)) # C-index
```

---

predict.CoxGradientBoost

*Predict from a CoxGradientBoost model*

---

**Description**

Predict from a CoxGradientBoost model

**Usage**

```
## S3 method for class 'CoxGradientBoost'
predict(object, newdata, type = "lp", times = NULL, M = NULL, ...)
```

**Arguments**

object	a CoxGradientBoost object
newdata	numeric matrix or data.frame of new covariates
type	"lp" (default), "survival", or "cumhaz"
times	time grid for type != "lp"; NULL -> training event times
M	use only the first M base learners (early stopping probe); NULL -> all M (default)
...	unused

**Value**

type = "lp" -> numeric vector (length n\_new)  
 type = "survival" -> matrix (n\_new x length(times))  
 type = "cumhaz" -> matrix (n\_new x length(times))

---

```
print.CoxGradientBoost
```

*Print a CoxGradientBoost Object*

---

**Description**

Displays a concise summary of a fitted CoxGradientBoost model, including the base learner, number of boosting iterations, learning rate, and basic information about the training data.

**Usage**

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

**Arguments**

x	An object of class "CoxGradientBoost".
...	Further arguments passed to or from other methods (currently unused).

**Details**

This method prints key characteristics of the fitted model:

- The base learner used in boosting
- The number of boosting iterations ( $M$ )
- The learning rate ( $\nu$ )
- The number of observations
- The number of observed events
- The number of unique event times

**Value**

The input object *x*, invisibly.

**See Also**

[cox\\_gradient\\_boost](#), [predict.CoxGradientBoost](#)

# Index

## \* package

survivalisttoo-package, [2](#)

cox\_gradient\_boost, [2](#), [5](#)

predict.CoxGradientBoost, [3](#), [5](#)

print.CoxGradientBoost, [4](#)

survivalisttoo

(survivalisttoo-package), [2](#)

survivalisttoo-package, [2](#)