

Package ‘ReSurv’

September 15, 2026

Type Package

Depends R (>= 4.1.0)

Title Machine Learning Models for Predicting Claim Counts

Version 1.1.0

Description Prediction of claim counts using the feature based development factors introduced in the manuscript Hiabu M., Hofman E. and Pittarello G. (2023) <[doi:10.48550/arXiv.2312.14549](https://doi.org/10.48550/arXiv.2312.14549)>. Implementation of Neural Networks, Extreme Gradient Boosting, and Cox model with splines to optimise the partial log-likelihood of proportional hazard models.

URL <https://github.com/edhofman/ReSurv>,
<https://edhofman.github.io/ReSurv/>

BugReports <https://github.com/edhofman/ReSurv/issues>

License GPL (>= 2)

Imports stats, dplyr (>= 1.1.0), actuar, fastDummies, data.table, purrr, tidyr, ggplot2, lubridate, survival, SynthETIC, xgboost

Encoding UTF-8

Suggests bshazard, clmplus, knitr, torch, rmarkdown, rpart, testthat (>= 3.0.0)

VignetteBuilder knitr, rmarkdown

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

Author Emil Hofman [aut, cre, cph],
Gabriele Pittarello [aut, cph] (ORCID:
<<https://orcid.org/0000-0003-3360-5826>>),
Munir Hiabu [aut, cph] (ORCID: <<https://orcid.org/0000-0001-5846-667X>>)

Maintainer Emil Hofman <emil_hofman@hotmail.dk>

Repository CRAN

Date/Publication 2026-09-15 08:50:02 UTC

Contents

data_generator	2
IndividualDataPP	4
plot.ReSurvPredict	7
predict.ReSurvFit	8
predictReserve	9
print.ReSurvCV	9
print.summaryReSurvPredict	10
ReSurv	10
ReSurvCV	13
Score_Reserving	15
summary.ReSurvPredict	16
Index	17

data_generator	<i>Individual data generator</i>
----------------	----------------------------------

Description

This function generates individual claims data on the requested time scale using the SynthETIC package. This simple function allows to simulate from a sand-box to test out the ReSurv approach. Some parameters of the simulation can be changed.

Usage

```
data_generator(  
  ref_claim = 2e+05,  
  time_unit = 1/360,  
  years = 4,  
  random_seed = 1964,  
  period_exposure = 200,  
  period_frequency = 0.2,  
  scenario = 1  
)
```

Arguments

- ref_claim integer, reference claim size.
- time_unit Numeric, fraction of a year per output period, for example 1/360 (the default) for days, 1/12 for months, or 1 for years.
- years integer, number of years to be simulated.
- random_seed integer, random seed for replicable code.
- period_exposure integer, volume (number of policies) underwritten each period.

period_frequency	numeric, expected frequency in each period.
scenario	character or numeric, one of the scenarios described in the accompanying manuscript. Possible choices are 'alpha' (0), 'beta' (1), 'gamma' (2), 'delta' (3), 'epsilon' (4). Our simulated data are constituted of a mix of short tail claims (claim_type 0) and claims with longer resolution (claim_type 1). We chose the parameter of the simulator to resemble a mix of property damage (claim_type 0) and bodily injuries (claim_type 1). each scenario has distinctive characteristics. Scenario Alpha is a mix of claim_type 0 and claim_type 1 with same number of claims volume at each accident period. Differently from scenario Alpha, in scenario Beta the volumes of claim_type 1 are decreasing in the most recent accident periods. In scenario Gamma we add an interaction between claim_type 1 and accident period: in a real world setting this can be motivated by a change in consumer behavior or company policies resulted in different reporting patterns over time. In scenario Delta, we introduce a seasonality effect dependent on the accident period for claim_type 0 and claim_type 1. In the real word, scenario Delta resembles seasonal changes in the workforce composition. Scenario Epsilon does not satisfy the proportionality assumption. Additional scenarios are "zeta" (5), with age, property value and business-use covariates, and "eta" (6), with business-use and accident-period effects.

Value

Individual claims data. It contains the following columns:

- claim_number: Policy ID.
- claim_type: Type of claim (0 or 1) in scenarios alpha through epsilon. Zeta instead includes age, property_value, and business_use; eta includes business_use.
- AP: Accident period
- RP: Reporting period.

References

Avanzi, B., Taylor, G., Wang, M., & Wong, B. (2021). SynthETIC: an individual insurance claim simulator with feature control. *Insurance: Mathematics and Economics*, 100, 296-308.

Hiabu, M., Hofman, E., & Pittarello, G. (2023). A machine learning approach based on survival analysis for IBNR frequencies in non-life reserving. *arXiv preprint arXiv:2312.14549*.

Examples

```
input_data_0 <- data_generator(
  random_seed = 1964,
  scenario = "alpha",
  time_unit = 1,
  years = 2,
  period_exposure = 100)
```

Description

This function pre-processes the data for the application of a ReSurv model.

Usage

```
IndividualDataPP(
  data,
  id = NULL,
  continuous_features = NULL,
  categorical_features = NULL,
  accident_period,
  calendar_period,
  input_time_granularity = "months",
  output_time_granularity = "quarters",
  years = NULL,
  calendar_period_extrapolation = FALSE,
  continuous_features_spline = NULL,
  degrees_cf = 3,
  degrees_of_freedom_cf = 4,
  degrees_cp = 3,
  degrees_of_freedom_cp = 4
)
```

Arguments

<code>data</code>	<code>data.frame</code> , for the individual reserving. The number of development periods can be larger than the number of accident periods.
<code>id</code>	character, data column that contains the policy identifier. If NULL (default), we assume that each row is an observation. We assume that each observation can only have one reporting time, if not null we take the reporting time of the first row for each id.
<code>continuous_features</code>	character, continuous features columns to be scaled.
<code>categorical_features</code>	character, categorical features columns to be one-hot encoded.
<code>accident_period</code>	character, it contains the name of the column in data corresponding to the accident period.
<code>calendar_period</code>	character, it contains the name of the column in data corresponding to the calendar period.

input_time_granularity	character, time unit of the input data. Granularity supported: • "days": the input data are daily. • "months": the input data are monthly. • "quarters": the input data are quarterly • "semesters": six-month periods. • "years": the input data are yearly. Default to months.
output_time_granularity	character, time unit of the output data. The granularity supported is the same as for the input data: • "days": the output data will be on a daily scale. • "months": the output data will be on a monthly scale. • "quarters": the output data will be on a quarterly scale. • "semesters": six-month periods. • "years": the output data will be on yearly scale. The output granularity must be equal to or coarser than the input granularity. Also, the output granularity must be consistent with the input granularity, meaning that the time conversion must be possible. E.g., it is possible to group quarters to years. Quarters can also be grouped to semesters. Default to quarters.
years	numeric, number of development years in the study.
calendar_period_extrapolation	logical, whether a spline for calendar extrapolation should be considered in the cox model fit. Default is 'FALSE'.
continuous_features_spline	character, names of continuous features to model with splines; NULL uses linear terms. Use "AP_i" for a remapped accident-period feature.
degrees_cf	numeric, degrees of the spline for smoothing continuous features.
degrees_of_freedom_cf	numeric, degrees of freedom of the splines for smoothing continuous features.
degrees_cp	numeric, degrees of the spline for smoothing the calendar period effect.
degrees_of_freedom_cp	numeric, degrees of freedom of the splines for smoothing the calendar period effect.

Details

Accident and reporting periods are indexed from one. Development time is $DP_i = RP_i - AP_i + 1$; reverse development time is $DP_{rev_i} = DP_{max} - DP_i + 1$, and truncation time is $TR_i = AP_i - 1$. Training retains observed rows with $DP_{rev_i} > TR_i$.

The conversion factor is the input time unit divided by the output time unit (for example, 1/3 for months to quarters). Accident and calendar periods are grouped using ceiling; development periods also account for the position of the accident period within each output period. Days use a 360-day year for the development horizon.

Value

An IndividualDataPP list containing `training.data` (observed rows), `full.data` (all encoded rows), and `data_information` (conversion factor, input/output formulas, feature names, time units, horizon, and original column names).

After pre-processing, we provide a standard encoding for the time components. This regards the output in `training.data`. In the ReSurv notation:

- AP_i: Input granularity accident period.
- AP_o: Output granularity accident period.
- DP_i: Input granularity development period in forward time.
- DP_rev_i: Input granularity development period in reverse time.
- DP_rev_o: Output granularity development period in reverse time.
- TR_i: Input granularity truncation time.
- TR_o: Output granularity truncation time.
- I: event indicator, under this framework is equal to one for each entry.

References

Hiabu, M., Hofman, E., & Pittarello, G. (2023). A machine learning approach based on survival analysis for IBNR frequencies in non-life reserving. arXiv preprint arXiv:2312.14549.

Examples

```
input_data_0 <- data_generator(
  random_seed = 1964,
  scenario = "alpha",
  time_unit = 1,
  years = 2,
  period_exposure = 100)

individual_data <- IndividualDataPP(data = input_data_0,
  categorical_features = "claim_type",
  continuous_features = "AP",
  accident_period = "AP",
  calendar_period = "RP",
  input_time_granularity = "years",
  output_time_granularity = "years",
  years = 2)
```

plot.ReSurvPredict *Plot of the development factors*

Description

Plots the development factors by group code.

Usage

```
## S3 method for class 'ReSurvPredict'
plot(
  x,
  granularity = "input",
  group_code = 1,
  color_par = "royalblue",
  linewidth_par = 2.5,
  ylim_par = NULL,
  ticks_by_par = NULL,
  base_size_par = NULL,
  title_par = NULL,
  x_text_par = NULL,
  plot.title.size_par = NULL,
  ...
)
```

Arguments

x	"ReSurvPredict" object specifying hazard and development factors.
granularity	character, either "input" for input_time_granularity or "output" for output_time_granularity.
group_code	numeric: Identifier for the group that will be plotted. Default is 1. The code identifiers can be find in the ReSurvPredict\$long_triangle_format_out list. Depending on the granularity of interest, it will be either in ReSurvPredict\$long_triangle_format_out for input_time_granularity or ReSurvPredict\$long_triangle_format_out\$output_granularity for output_time_granularity.
color_par	character: ggplot2 Colour of the line plot. Default is 'royalblue'. Optional.
linewidth_par	numeric: Line plot width. Optional.
ylim_par	numeric: Highest intercept on the y-axis (development factors). The default is the highest predicted development factor. Optional.
ticks_by_par	numeric: gap between each x-axis label (development period). Default is 2. Optional.
base_size_par	numeric: base size of the plot. Default is 5. See base_size in the ?theme_bw documentation. Optional.
title_par	character: Title of the plot. Optional.

`x_text_par` character: Text on the x-axis. Optional.
`plot.title.size_par` numeric: size of the plot title. Default is 20. See size in the ?element_text documentation. Optional.
`...` Other arguments to be passed to Plot. Optional.

Value

ggplot2 of the development factors

predict.ReSurvFit	<i>Predict IBNR frequency</i>
-------------------	-------------------------------

Description

This function predicts the results from the ReSurv fits.

Usage

```
## S3 method for class 'ReSurvFit'
predict(
  object,
  newdata = NULL,
  lower_triangular_output = FALSE,
  minimal_output = FALSE,
  check_value = 1.85,
  ...
)
```

Arguments

`object` A fitted ReSurvFit object.
`newdata` An optional IndividualDataPP object using the same features and time scale as the fitted data. NULL predicts for the fitted data.
`lower_triangular_output` logical, if set to TRUE we add the predicted lower triangle in input and output granularity to the predict.ReSurvFit output.
`minimal_output` logical, if set to TRUE return a reduced prediction object.
`check_value` Retained for compatibility; currently ignored.
`...` Additional arguments to pass to the predict function.

Value

A ReSurvPredict object with fitted predictions, long triangle outputs, predicted counts, and optional lower-triangle outputs.

predictReserve	<i>Predict deterministic reserve table</i>
----------------	--

Description

Predict deterministic reserve table

Predict deterministic reserve table from a ReSurvFit object

Usage

```
predictReserve(object, ...)
```

```
## S3 method for class 'ReSurvFit'
```

```
predictReserve(object, granularity = c("output", "input"), ...)
```

Arguments

object A ReSurvFit object.

... Additional arguments passed to predict.ReSurvFit().

granularity Character. Either "output" or "input".

Value

A data.table with accident period AP, development period DP, calendar period CP, and predicted claim count IBNR. These are claim counts, not monetary reserves.

A data.table with columns AP, DP, CP, IBNR.

See Also

[predict.ReSurvFit](#)

print.ReSurvCV	<i>Print method for ReSurvCV objects</i>
----------------	--

Description

Print method for ReSurvCV objects

Usage

```
## S3 method for class 'ReSurvCV'
```

```
print(x, ...)
```

Arguments

x An object of class ReSurvCV.

... Additional arguments, currently ignored.

```
print.summaryReSurvPredict
```

Print summary of IBNR predictions

Description

Gives overview of IBNR predictions

Usage

```
## S3 method for class 'summaryReSurvPredict'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

<code>x</code>	"ReSurvPredict" object specifying hazard and development factors.
<code>digits</code>	numeric, number of digits to print for IBNR and Likelihood.
<code>...</code>	Other arguments to be passed to print.

Value

print of summary of predictions

ReSurv	<i>Fit ReSurv models on the individual data.</i>
--------	--

Description

This function fits and computes the reserves for the ReSurv models

Usage

```
ReSurv(
  IndividualDataPP,
  hazard_model = "COX",
  tie = "efron",
  baseline = "spline",
  continuous_features_scaling_method = "minmax",
  random_seed = 1,
  hparameters = list(),
  percentage_data_training = 0.8,
  grouping_method = "exposure",
  check_value = 1.85,
  eta = 0.5,
  simplifier = TRUE
```

```

)

## Default S3 method:
ReSurv(
  IndividualDataPP,
  hazard_model = "COX",
  tie = "efron",
  baseline = "spline",
  continuous_features_scaling_method = "minmax",
  random_seed = 1,
  hparameters = list(),
  percentage_data_training = 0.8,
  grouping_method = "exposure",
  check_value = 1.85,
  eta = 0.5,
  simplifier = TRUE
)

## S3 method for class 'IndividualDataPP'
ReSurv(
  IndividualDataPP,
  hazard_model = "COX",
  tie = "efron",
  baseline = "spline",
  continuous_features_scaling_method = "minmax",
  random_seed = 1,
  hparameters = list(),
  percentage_data_training = 0.8,
  grouping_method = "exposure",
  check_value = 1.85,
  eta = 0.5,
  simplifier = TRUE
)

```

Arguments

IndividualDataPP	IndividualDataPP object to use for the ReSurv fit.
hazard_model	character, hazard model supported from our package, must be provided as a string. The model can be chosen from: • "COX": Standard Cox model for the hazard. • "NN": Deep Survival Neural Network. • "XGB": eXtreme Gradient Boosting.
tie	Handling of ties in the Cox fit, passed to <code>survival::coxph()</code> : "efron" (default), "breslow", or "exact". NN and XGB fitting use Efron handling.
baseline	Retained for compatibility. The baseline is estimated from the risk sets using the specified eta.

continuous_features_scaling_method	Scaling method: "minmax" or "standard".
random_seed	integer, random seed set for reproducibility
hparameters	list, hyperparameters for the machine learning models. It will be disregarded for the cox approach.
percentage_data_training	numeric, fraction in (0, 1] of observed data used for training; the remainder is used for validation in the NN and XGB models.
grouping_method	Retained for compatibility; currently ignored.
check_value	Retained for compatibility; currently ignored.
eta	Numeric in [0, 1], controlling the baseline estimate and hazard-to-development-factor conversion. Default is 0.5.
simplifier	logical, kept for compatibility. The simplified forecast frame is always used.

Details

The model fit uses the theoretical framework of Hiabu et al. (2023), that relies on the correspondence between hazard models and development factors:

Neural networks use the native R torch backend. Install its runtime with `torch::install_torch()` before fitting a neural network.

The ReSurv package assumes proportional hazard models. Given an i.i.d. sample $\{y_i, x_i\}_{i=1, \dots, n}$ the individual hazard at time t is:

$$\lambda_i(t) = \lambda_0(t)e^{y_i(x_i)}$$

Composed of a baseline $\lambda_0(t)$ and a proportional effect $e^{y_i(x_i)}$.

Currently, the implementation allows to optimize the partial likelihood (concerning the proportional effects) using one of the following statistical learning approaches:

- **COX**
- **Neural Networks**
- **eXtreme Gradient Boosting**

Value

A `ReSurvFit` list containing `model.out` (design matrix and fitted backend model), `hazard_frame` (risk scores, baseline, hazards, development factors and survival probabilities), `data_information` (preprocessing and reserving metadata), and `fit_information` (model name, training loss `is_lkh`, validation loss `os_lkh`, and `eta`). Use `predict()` or `predictReserve()` to obtain claim-count predictions.

References

- Hiabu, M., Hofman, E., & Pittarello, G. (2023). A machine learning approach based on survival analysis for IBNR frequencies in non-life reserving. arXiv preprint arXiv:2312.14549.
- Therneau, T. M., & Lumley, T. (2015). Package `~survival~`™. R Top Doc, 128(10), 28-33.

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., & Kluger, Y. (2018). DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. BMC medical research methodology, 18(1), 1-12.

Chen, T., He, T., Benesty, M., & Khotilovich, V. (2019). Package `xgboost`™. R version, 90, 1-66.

Examples

```
input_data_0 <- data_generator(
  random_seed = 1964,
  scenario = "alpha",
  time_unit = 1,
  years = 4,
  period_exposure = 100)

individual_data <- IndividualDataPP(data = input_data_0,
  categorical_features = "claim_type",
  continuous_features = "AP",
  accident_period = "AP",
  calendar_period = "RP",
  input_time_granularity = "years",
  output_time_granularity = "years",
  years=4)

resurv_fit_cox <- ReSurv(individual_data,
  hazard_model = "COX",
  eta = 0)
```

ReSurvCV

K-fold cross-validation of a ReSurv model

Description

K-fold cross-validation of a ReSurv model

Usage

```
ReSurvCV(
  IndividualDataPP,
  model,
  hparameters_grid,
  folds,
  random_seed,
  continuous_features_scaling_method = "minmax",
```

```

print_every_n = 1L,
nrounds = NULL,
early_stopping_rounds = NULL,
epochs = 1,
parallel = FALSE,
ncores = 1,
num_workers = 0,
verbose = FALSE,
verbose.cv = FALSE,
cv_data_subsample = 1
)

```

Arguments

IndividualDataPP	An object of class IndividualDataPP.
model	Character. Either "NN" or "XGB".
hparameters_grid	Named list defining the hyperparameter grid.
folds	Integer. Number of folds.
random_seed	Integer. Random seed.
continuous_features_scaling_method	Character. Scaling method for continuous features.
print_every_n	Integer. Passed to XGBoost.
nrounds	Integer. Number of XGBoost boosting rounds.
early_stopping_rounds	Integer. XGBoost early stopping.
epochs	Integer. Number of NN epochs.
parallel	Logical. Retained for compatibility; execution is sequential and setting this to TRUE produces a warning.
ncores	Integer. Retained for compatibility; currently ignored.
num_workers	Deprecated for the native torch backend. Ignored.
verbose	Logical. Print model fitting output.
verbose.cv	Logical. Print CV progress.
cv_data_subsample	Fraction of training rows used for cross-validation, in (0, 1]. Values greater than 1 and at most 100 are interpreted as percentages. The default 1 uses all rows; use 0.01 for one percent.

Value

An object of class ReSurvCV containing `out.cv` (all combinations and mean training and validation losses), `out.cv.best.oos` (the row with smallest validation loss), and `hparameters.best`, suitable for the `hparameters` argument of `ReSurv()`. Cross-validation does not refit the final model.

Score_Reserving	<i>Score reserving predictions</i>
-----------------	------------------------------------

Description

Compare predictions with realized claim counts in the lower triangle.

Usage

```
Score_Reserving(
  models,
  newdata,
  scoring_metrics = c("EI", "R-tot", "R-cell-wise", "R-cal-wise"),
  granularity = c("output", "input"),
  chain_ladder = TRUE,
  clmplus_benchmark = NULL,
  ...
)
```

Arguments

<code>models</code>	A fitted <code>ReSurvFit</code> or a named list of models. At least one entry must be a <code>ReSurvFit</code> ; other entries must support <code>predictReserve()</code> .
<code>newdata</code>	Realized claims, either individual data with the accident and reporting columns used for fitting, or an aggregate table with AP, DP, CP, and counts in actual, I, or IBNR. Aggregate periods must be on the requested granularity scale.
<code>scoring_metrics</code>	Character vector selecting "EI", "R-tot", "R-cell-wise", "R-cal-wise", or "CRPS". CRPS is available only for <code>ReSurvFit</code> models.
<code>granularity</code>	Time scale for scoring: "output" (default) or "input".
<code>chain_ladder</code>	Logical; include the aggregate chain-ladder benchmark.
<code>clmplus_benchmark</code>	Optional character vector containing any of "ac" or "apc". Requested models are fitted with the 'clmplus' package to the same aggregate triangle as the chain-ladder benchmark. 'NULL' (the default) disables these benchmarks.
<code>...</code>	Additional arguments passed to <code>predictReserve()</code> .

Value

A list of score tables, one per requested metric, with class `Score_Reserving` and a `granularity` attribute.

summary.ReSurvPredict *Summary of IBNR predictions*

Description

Gives overview of IBNR predictions

Usage

```
## S3 method for class 'ReSurvPredict'  
summary(object, granularity = "input", ...)
```

Arguments

object	"ReSurvPredict" object specifying hazard and development factors.
granularity	character, specify if which granularity the summary should be on. • "input" • "output" Default is "input".
...	Other arguments to be passed to summary.

Value

Summary of predictions

Index

`data_generator`, [2](#)

`IndividualDataPP`, [4](#)

`plot.ReSurvPredict`, [7](#)

`predict.ReSurvFit`, [8](#), [9](#)

`predictReserve`, [9](#)

`print.ReSurvCV`, [9](#)

`print.summaryReSurvPredict`, [10](#)

`ReSurv`, [10](#)

`ReSurvCV`, [13](#)

`Score_Reserving`, [15](#)

`summary.ReSurvPredict`, [16](#)