

Package ‘triageR’

September 5, 2026

Title Automated Machine Learning and AI Agent Tools for Clinical Prediction Modelling

Version 0.2.0

Description Provides a streamlined workflow for building, validating, and reporting clinical prediction models. Combines standard machine learning tools with an optional AI agent that recommends appropriate statistical methods, runs sensitivity analyses, and flags common pitfalls. Includes automated generation of reports aligned with TRIPOD+AI reporting guidance (Collins et al. (2024 <[doi:10.1136/bmj-2023-078378](https://doi.org/10.1136/bmj-2023-078378)>)) for reproducible, guideline-aligned research.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 8.0.0

URL <https://github.com/DevWebWacky/triageR>,
<https://devwebwacky.github.io/triageR/>

BugReports <https://github.com/DevWebWacky/triageR/issues>

Suggests knitr, rmarkdown, mlbench, missForest, ranger, spelling, testthat (>= 3.0.0), xgboost, MASS, aorsf, censored, shiny, bslib

Config/testthat/edition 3

Imports DALEX, dplyr, ellmer, ggplot2, mice, naniar, parsnip, pROC, quarto, recipes, survival, tibble, tidyr, workflows, yardstick

VignetteBuilder knitr

NeedsCompilation no

Author Uwakmfon Paul [aut, cre, cph]

Maintainer Uwakmfon Paul <uwakmfon31@gmail.com>

Depends R (>= 4.1.0)

Repository CRAN

Date/Publication 2026-09-05 02:20:02 UTC

Contents

tr_agent_review	2
tr_check_missing	3
tr_explain	3
tr_fit	4
tr_fit_survival	5
tr_impute	6
tr_launch_app	7
tr_load_clinical	7
tr_recommend_method	8
tr_sensitivity	9
tr_tripod_report	9
tr_validate	11
tr_validate_survival	12
Index	13

tr_agent_review	<i>Review a clinical modelling pipeline for common pitfalls</i>
-----------------	---

Description

Runs a series of rule-based checks for common clinical-ML pitfalls (class imbalance or low event rate, low events-per-variable, possible leakage, near-zero variance predictors), then optionally asks an LLM to summarize the findings in plain language. Works with both binary classification models (trriageR_model) and survival models (trriageR_survival_model).

Usage

```
tr_agent_review(data, model, use_agent = TRUE)
```

Arguments

data	The data frame used to fit the model.
model	A fitted triageR_model or triageR_survival_model object.
use_agent	Logical. If TRUE (default), also generates an AI-written plain-language summary of the findings via ellmer.

Value

A triageR_review object (list) containing a tibble of flags and, if use_agent = TRUE, an AI-generated summary.

Examples

```
## Not run:  
tr_agent_review(data, model)  
  
## End(Not run)
```

tr_check_missing	<i>Check missing data in a clinical dataset</i>
------------------	---

Description

Produces a summary table of missingness per column, plus a visual plot showing the pattern of missing data across the dataset.

Usage

```
tr_check_missing(data)
```

Arguments

data A data frame, typically the output of `tr_load_clinical()`.

Value

Invisibly returns a summary tibble (columns, n_missing, pct_missing). Also prints a summary to console and displays a missingness plot as a side effect.

Examples

```
df <- data.frame(a = c(1, NA, 3), b = c(4, 5, NA))  
tr_check_missing(df)
```

tr_explain	<i>Explain a clinical prediction model</i>
------------	--

Description

Generates variable importance or prediction explanations for a fitted `triageR_model`, using the DALEX framework under the hood.

Usage

```
tr_explain(  
  model,  
  method = c("permutation", "shap"),  
  newdata = NULL,  
  observation = 1  
)
```

Arguments

model	A fitted triageR_model object from tr_fit().
method	Character. One of "permutation" (global variable importance) or "shap" (SHAP values for a single prediction).
newdata	Optional data frame to explain predictions on. If NULL, uses the training data.
observation	Integer. Row index of the observation to explain, only used when method = "shap". Defaults to 1.

Value

A triageR_explanation object (list) containing the explanation result and a plot.

Examples

```
## Not run:
tr_explain(model, method = "permutation")
tr_explain(model, method = "shap", observation = 3)

## End(Not run)
```

tr_fit

Fit a clinical prediction model

Description

Fits a binary classification model using the parsnip/workflows framework. The user must specify the model engine explicitly.

Usage

```
tr_fit(
  data,
  outcome,
  engine = c("logistic_reg", "random_forest", "boost_tree"),
  predictors = NULL
)
```

Arguments

data	A data frame containing predictors and the outcome column.
outcome	Character. Name of the binary outcome column (must be a factor or coercible to one, with the event of interest as the second level).
engine	Character. Model engine to use. One of "logistic_reg", "random_forest", or "boost_tree".
predictors	Character vector of predictor column names. If NULL (default), all columns except outcome are used.

Value

A fitted triageR_model object — a list containing the fitted workflow, the engine used, and the outcome/predictor names.

Examples

```
set.seed(1)
df <- data.frame(
  age = round(rnorm(50, 55, 12)),
  sex = sample(c("M", "F"), 50, replace = TRUE),
  disease = sample(c(0, 1), 50, replace = TRUE)
)
model <- tr_fit(df, outcome = "disease", engine = "logistic_reg")
```

tr_fit_survival	<i>Fits a survival model using the censored/parsnip framework. The outcome must be specified as separate time and event columns (following survival::Surv() convention), and the user must explicitly choose an engine.</i>
-----------------	---

Description

Fits a survival model using the censored/parsnip framework. The outcome must be specified as separate time and event columns (following survival::Surv() convention), and the user must explicitly choose an engine.

Usage

```
tr_fit_survival(
  data,
  time_col,
  event_col,
  engine = c("cox_ph", "survival_rf"),
  predictors = NULL
)
```

Arguments

data	A data frame containing predictors, a time column, and an event column.
time_col	Character. Name of the column giving time to event or censoring.
event_col	Character. Name of the column indicating event status (1 = event occurred, 0 = censored), following standard survival analysis convention.
engine	Character. One of "cox_ph" (Cox Proportional Hazards) or "survival_rf" (Random Survival Forest via the aorsf engine).
predictors	Character vector of predictor column names. If NULL (default), all columns except time_col and event_col are used.

Value

A fitted `triarageR_survival_model` object (list) containing the fitted workflow, engine, and time/event/predictor column names.

Examples

```
if (requireNamespace("survival", quietly = TRUE)) {
  library(survival)
  lung_clean <- lung
  lung_clean$status <- lung_clean$status - 1 # convert 1/2 to 0/1
  lung_clean <- lung_clean[stats::complete.cases(lung_clean), ]
  model <- tr_fit_survival(lung_clean, time_col = "time",
                          event_col = "status", engine = "cox_ph")
}
```

tr_impute	<i>Impute missing values in a clinical dataset</i>
-----------	--

Description

Fills in missing values using either multiple imputation (mice) or a random-forest based approach (missForest). The method must be chosen explicitly, this function does not guess for you.

Usage

```
tr_impute(data, method = c("mice", "missForest"), m = 5, seed = 123)
```

Arguments

data	A data frame with missing values, typically the output of <code>tr_load_clinical()</code> .
method	Character. Either "mice" (default) or "missForest".
m	Integer. Number of multiple imputations to run if method = "mice". Defaults to 5. Ignored for "missForest".
seed	Integer. Random seed for reproducibility. Defaults to 123.

Value

A completed data frame with missing values filled in. If method = "mice", the first completed dataset is returned, and the full `mids` object is attached as an attribute ("mice_object") in case the user wants to inspect all imputations.

Examples

```
df <- data.frame(
  a = c(5, 7, 3, 9, 2, 8, 6, 4, 5, 7),
  b = c(1, 3, 2, 4, 5, 3, 2, NA, 1, 3)
)
completed <- tr_impute(df, method = "mice", m = 2)
```

tr_launch_app	<i>Launch the triageR Shiny app</i>
---------------	-------------------------------------

Description

Opens an interactive Shiny application for building, validating, and reporting clinical prediction models without writing R code directly. Supports data upload, model fitting across multiple engines, validation, explainability, automated pipeline review, and TRIPOD+AI report generation. The AI method-recommendation feature requires a configured Gemini API key (see `?tr_recommend_method`) and is optional.

Usage

```
tr_launch_app()
```

Value

Does not return; launches a Shiny application.

Examples

```
if (interactive()) {  
  tr_launch_app()  
}
```

tr_load_clinical	<i>Load and standardize a clinical dataset</i>
------------------	--

Description

Reads a flat clinical dataset (CSV or data frame) and returns it as a standardized triageR object with basic structure checks. This is the entry point for most triageR workflows.

Usage

```
tr_load_clinical(data, id_col = "id")
```

Arguments

data	A file path to a CSV, or an existing data frame.
id_col	Character. Name of the column identifying unique patients. Defaults to "id".

Value

A tibble of class triageR_data, with basic metadata attached.

Examples

```
df <- data.frame(
  id = 1:5,
  age = c(45, 62, 38, 71, 55),
  sex = c("F", "M", "F", "M", "F")
)
tr_load_clinical(df, id_col = "id")
```

tr_recommend_method	<i>Recommend an appropriate statistical or ML method (AI agent)</i>
---------------------	---

Description

Uses an LLM to inspect the structure of a clinical dataset and suggest an appropriate statistical or machine learning approach, with reasoning. This is an assistive tool, not a replacement for expert judgment.

Usage

```
tr_recommend_method(data, outcome, context = NULL)
```

Arguments

data	A data frame, typically the output of <code>tr_load_clinical()</code> .
outcome	Character. Name of the outcome column of interest.
context	Optional character string giving extra clinical context (e.g. "predicting 30-day readmission in heart failure patients").

Value

Invisibly returns the raw text recommendation (character string). Also prints the recommendation to console.

Examples

```
## Not run:
tr_recommend_method(data, outcome = "disease",
  context = "predicting diabetes onset in adults")

## End(Not run)
```

tr_sensitivity	<i>Run an automated sensitivity analysis battery</i>
----------------	--

Description

Re-fits a clinical prediction model under several robustness checks, complete-case vs imputed data, outlier exclusion, and subgroup consistency, and compares performance metrics across them.

Usage

```
tr_sensitivity(data, model, subgroup_col = NULL, outlier_sd = 3)
```

Arguments

data	The original (pre-imputation) data frame, containing the same predictors and outcome used in model.
model	A fitted triageR_model object from tr_fit().
subgroup_col	Optional character. Name of a categorical column (e.g. "sex") to check subgroup consistency across.
outlier_sd	Numeric. Number of standard deviations beyond which a numeric predictor value is considered an outlier and excluded in the outlier-robustness check. Defaults to 3.

Value

A triageR_sensitivity object (list) with a comparison tibble of metrics across all sensitivity scenarios.

Examples

```
## Not run:
tr_sensitivity(data, model, subgroup_col = "sex")

## End(Not run)
```

tr_tripod_report	<i>Generate a TRIPOD+AI-aligned clinical model report</i>
------------------	---

Description

Renders a reproducible report summarizing model fit, validation, sensitivity analysis, and pipeline review, aligned with TRIPOD+AI reporting guidance. Supports both binary classification and survival models. This is a drafting aid, not a certified compliance tool.

Usage

```
tr_tripod_report(
  model,
  model_type = c("classification", "survival"),
  validation = NULL,
  review = NULL,
  sensitivity = NULL,
  recommendation = NULL,
  output_file = file.path(tempdir(), "trriageR_report"),
  format = c("html", "docx")
)
```

Arguments

<code>model</code>	A fitted <code>trriageR_model</code> or <code>trriageR_survival_model</code> object.
<code>model_type</code>	Character. Either "classification" (default) or "survival". Must match the type of model supplied.
<code>validation</code>	Optional validation object: a <code>trriageR_validation</code> (from <code>tr_validate()</code>) for classification models, or a <code>trriageR_survival_validation</code> (from <code>tr_validate_survival()</code>) for survival models.
<code>review</code>	Optional <code>trriageR_review</code> object from <code>tr_agent_review()</code> .
<code>sensitivity</code>	Optional <code>trriageR_sensitivity</code> object from <code>tr_sensitivity()</code> . Not currently supported for survival models.
<code>recommendation</code>	Optional character string from <code>tr_recommend_method()</code> .
<code>output_file</code>	Character. File path (without extension) to save the report to. Defaults to a file in <code>tempdir()</code> . Set explicitly (e.g. <code>file.path("my_folder", "report")</code>) to save elsewhere.
<code>format</code>	Character. Either "html" (default) or "docx".

Value

Invisibly returns the path to the rendered report file.

Examples

```
## Not run:
tr_tripod_report(model, validation = val, review = rev,
  output_file = file.path(tempdir(), "report"))

## End(Not run)
```

`tr_validate`*Validate a clinical prediction model*

Description

Computes discrimination (AUC, sensitivity, specificity) and calibration metrics for a fitted `triageR_model`. Can validate on new (external/holdout) data, or fall back to the training data with a clear warning.

Usage

```
tr_validate(  
  model,  
  newdata = NULL,  
  threshold = 0.5,  
  calibration_method = c("binned", "smooth")  
)
```

Arguments

<code>model</code>	A fitted <code>triageR_model</code> object from <code>tr_fit()</code> .
<code>newdata</code>	Optional data frame to validate on. If <code>NULL</code> (default), validation runs on the original training data, with a warning.
<code>threshold</code>	Numeric. Probability threshold for classifying the positive class. Defaults to 0.5.
<code>calibration_method</code>	Character. One of "binned" (groups patients into risk deciles, the standard clinical approach) or "smooth" (loess-smoothed calibration curve, more granular). Defaults to "binned".

Value

A `triageR_validation` object (list) containing a metrics tibble and the underlying predictions, invisibly printed as a summary.

Examples

```
set.seed(1)  
df <- data.frame(  
  age = round(rnorm(50, 55, 12)),  
  sex = sample(c("M", "F"), 50, replace = TRUE),  
  disease = sample(c(0, 1), 50, replace = TRUE)  
)  
model <- tr_fit(df, outcome = "disease", engine = "logistic_reg")  
tr_validate(model, newdata = df)
```

tr_validate_survival *Validate a clinical survival model*

Description

Computes the concordance index (C-index) for a fitted `trriageR_survival_model` — the survival-analysis equivalent of AUC, measuring how well the model ranks patients by risk. Can validate on new (external/holdout) data, or fall back to the training data with a clear warning.

Usage

```
tr_validate_survival(model, newdata = NULL)
```

Arguments

<code>model</code>	A fitted <code>trriageR_survival_model</code> object from <code>tr_fit_survival()</code> .
<code>newdata</code>	Optional data frame to validate on. If <code>NULL</code> (default), validation runs on the original training data, with a warning.

Value

A `trriageR_survival_validation` object (list) containing the C-index and supporting details.

Examples

```
if (requireNamespace("survival", quietly = TRUE)) {
  library(survival)
  lung_clean <- lung
  lung_clean$status <- lung_clean$status - 1
  lung_clean <- lung_clean[stats::complete.cases(lung_clean), ]
  model <- tr_fit_survival(lung_clean, time_col = "time",
    event_col = "status", engine = "cox_ph")
  tr_validate_survival(model, newdata = lung_clean)
}
```

Index

tr_agent_review, [2](#)
tr_check_missing, [3](#)
tr_explain, [3](#)
tr_fit, [4](#)
tr_fit_survival, [5](#)
tr_impute, [6](#)
tr_launch_app, [7](#)
tr_load_clinical, [7](#)
tr_recommend_method, [8](#)
tr_sensitivity, [9](#)
tr_tripod_report, [9](#)
tr_validate, [11](#)
tr_validate_survival, [12](#)