

Package ‘icarm’

July 9, 2026

Title Interpretable Contextual-Accountable and Responsible Machine Learning

Version 0.2.0

Description A general-purpose framework for Interpretable Contextual-Accountable and Responsible Machine Learning (ICARM) that works with any clean tabular data across any application domain including healthcare, finance, social science, business, and education. Automatically detects whether a prediction task is binary classification, multi-class classification, or regression from the target variable type. Provides a unified entry point `icarm_fit()` supporting both interpretable learners (Classification and Regression Trees (CART), logistic regression, linear regression, Generalized Additive Models (GAM)) and extended learners (random forest, 'XGBoost', Support Vector Machines (SVM)) with consistent interfaces for global and local model explanation including approximate SHapley Additive exPlanations (SHAP) values and Partial Dependence Profiles (PDPs), learning curve diagnostics, group-level fairness auditing across protected attributes, probability calibration, threshold analysis, multi-model comparison, reproducible JavaScript Object Notation (JSON) audit trails, and accountability scorecards. The contextual accountability framing emphasises that algorithmic fairness and interpretability requirements depend on the deployment domain and must be evaluated accordingly. Extends the 'civic.icarm' framework (Awe 2025) <<https://cran.r-project.org/package=civic.icarm>> to general-purpose applications beyond civic and political education.

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

Depends R (>= 4.1.0)

Imports stats, utils, rpart, ggplot2, dplyr, tidyr, tibble, purrr, rlang, jsonlite, digest

Suggests randomForest, xgboost, e1071, mgcv, glmnet, nnet, DALEX, pROC, testthat, covr

Config/testthat/edition 3

LazyData true

RoxygenNote 7.3.3

NeedsCompilation no

Author Olushina Olawale Awe [aut, cre],
Ludwigsburg University of Education [fnd]

Maintainer Olushina Olawale Awe <olawaleawe@gmail.com>

Repository CRAN

Date/Publication 2026-07-09 10:20:11 UTC

Contents

icarm_audit	3
icarm_calibrate	4
icarm_compare	4
icarm_equalized_odds_curve	5
icarm_equity_summary	6
icarm_explain	7
icarm_explain_local	7
icarm_fairness	8
icarm_financial	9
icarm_fit	10
icarm_learning_curve	11
icarm_medical	12
icarm_metrics	13
icarm_pdp	14
icarm_plot_calibration	14
icarm_plot_comparison	15
icarm_plot_confusion	15
icarm_plot_fairness	16
icarm_plot_importance	17
icarm_plot_learning_curve	17
icarm_plot_pdp	18
icarm_plot_roc_groups	19
icarm_plot_shap	19
icarm_plot_thresholds	20
icarm_racism_survey	20
icarm_scorecard	21
icarm_shap	22
icarm_split	23
icarm_thresholds	24
predict.icarm_model	24
print.icarm_model	25
summary.icarm_model	26

Index

27

icarm_audit	<i>Generate a JSON audit trail</i>
-------------	------------------------------------

Description

Generate a JSON audit trail

Usage

```
icarm_audit(  
  object,  
  metrics = NULL,  
  fairness = NULL,  
  notes = NULL,  
  analyst = NULL,  
  path = NULL  
)
```

Arguments

object	An 'icarm_model'.
metrics	Named numeric vector from [icarm_metrics()].
fairness	An 'icarm_fairness' from [icarm_fairness()].
notes	Character analyst notes.
analyst	Character analyst name.
path	File path to write JSON (optional).

Value

Invisibly, the JSON string.

Examples

```
m <- icarm_fit(Species ~ ., iris)  
trail <- icarm_audit(m, analyst = "O. O. Awe")  
cat(trail)
```

icarm_calibrate	<i>Probability calibration diagnostics</i>
-----------------	--

Description

Probability calibration diagnostics

Usage

```
icarm_calibrate(object, data, outcome, positive = NULL, n_bins = 10L)
```

Arguments

object	An 'icarm_model' (binary only).
data	A data frame.
outcome	Character outcome column.
positive	Positive class label.
n_bins	Number of bins (default 10).

Value

An object of class 'icarm_calibration'.

Examples

```
m <- icarm_fit(
  Petal.Width ~ Sepal.Length + Sepal.Width, iris,
  model = "linear")
# calibration only for binary:
data(icarm_medical)
m2 <- icarm_fit(readmitted ~ ., icarm_medical)
cal <- icarm_calibrate(m2, icarm_medical, "readmitted", "Yes")
print(cal)
```

icarm_compare	<i>Compare multiple icarm_models</i>
---------------	--------------------------------------

Description

Compare multiple icarm_models

Usage

```
icarm_compare(
  models,
  test_data,
  outcome,
  protected = NULL,
  positive = NULL,
  threshold = 0.5
)
```

Arguments

<code>models</code>	A named list of ‘icarm_model’ objects.
<code>test_data</code>	A data frame for evaluation.
<code>outcome</code>	Character outcome column.
<code>protected</code>	Optional protected attribute for fairness.
<code>positive</code>	Positive class (binary).
<code>threshold</code>	Decision threshold (binary, default 0.5).

Value

A tibble of class ‘icarm_comparison’.

Examples

```
sp <- icarm_split(iris, stratify = "Species")
m1 <- icarm_fit(Species ~ ., sp$train, model = "cart")
if (requireNamespace("nnet", quietly = TRUE)) {
  m2 <- icarm_fit(Species ~ ., sp$train, model = "multinomial")
  cmp <- icarm_compare(list(CART = m1, Multinom = m2),
    sp$test, outcome = "Species")
  print(cmp)
}
```

icarm_equalized_odds_curve

Equalized odds curves across thresholds

Description

Equalized odds curves across thresholds

Usage

```
icarm_equalized_odds_curve(
  object,
  data,
  outcome,
  protected,
  positive = NULL,
  thresholds = seq(0.05, 0.95, 0.05)
)
```

Arguments

object	An 'icarm_model' (binary only).
data	A data frame.
outcome	Character outcome column.
protected	Character protected attribute column.
positive	Positive class label.
thresholds	Numeric threshold vector.

Value

A tibble with threshold, group, tpr, fpr, tnr.

icarm_equity_summary	<i>Equity summary from a fairness report</i>
----------------------	--

Description

Equity summary from a fairness report

Usage

```
icarm_equity_summary(fairness)
```

Arguments

fairness	An 'icarm_fairness' from [icarm_fairness()].
----------	--

Value

A named list of scalar equity indicators.

icarm_explain	<i>Generate global model explanations</i>
---------------	---

Description

Generate global model explanations

Usage

```
icarm_explain(object, data = NULL, label = NULL)
```

Arguments

object	An 'icarm_model' from [icarm_fit()].
data	Optional data frame for DALEX explainer.
label	Optional label for DALEX explainer.

Value

An object of class 'icarm_explainer'.

Examples

```
m <- icarm_fit(Species ~ ., iris)
ex <- icarm_explain(m)
print(ex)
print(icarm_plot_importance(ex))
```

icarm_explain_local	<i>Local explanation for individual observations</i>
---------------------	--

Description

Local explanation for individual observations

Usage

```
icarm_explain_local(explainer, newdata, n_features = 10L)
```

Arguments

explainer	An 'icarm_explainer' from [icarm_explain()].
newdata	A data frame of observations to explain.
n_features	Max features to show (default 10).

Value

A list of tibbles, one per row of newdata.

Examples

```
m <- icarm_fit(Species ~ ., iris)
ex <- icarm_explain(m)
icarm_explain_local(ex, iris[1:2, ])
```

icarm_fairness	<i>Group-level fairness audit</i>
----------------	-----------------------------------

Description

Group-level fairness audit

Usage

```
icarm_fairness(
  object,
  data,
  outcome,
  protected,
  positive = NULL,
  threshold = 0.5
)
```

Arguments

object	An ‘icarm_model’.
data	A data frame with outcome and protected column.
outcome	Character. Outcome column name.
protected	Character. Protected attribute column name.
positive	Positive class (binary).
threshold	Decision threshold (binary, default 0.5).

Value

A tibble of class ‘icarm_fairness’.

Examples

```
m <- icarm_fit(Species ~ ., iris)
idf <- iris
idf$size <- factor(ifelse(idf$Sepal.Length > 5.8, "large", "small"))
icarm_fairness(m, idf, "Species", "size")
```

icarm_financial	<i>Synthetic Financial Loan Default Dataset</i>
-----------------	---

Description

A synthetic dataset of 1,000 loan applicants with financial and demographic variables. Suitable for binary classification (predicting loan default) and fairness auditing across gender and ethnicity — a classic algorithmic fairness benchmark.

Usage

```
icarm_financial
```

Format

A tibble with 1,000 rows and 12 variables:

age Integer. Applicant age (20-75).
income Numeric. Annual income (USD).
credit_score Integer. Credit score (300-850).
loan_amount Numeric. Requested loan amount (USD).
loan_term Integer. Loan term in months.
employment_status Factor. Employment category.
home_owner Factor. Yes or No.
num_accounts Integer. Number of credit accounts.
debt_ratio Numeric. Debt-to-income ratio (0-1).
gender Factor. male or female.
ethnicity Factor. Ethnicity category.
default Factor. Yes (defaulted) or No.

Source

Synthetic data generated by the icarm team.

Examples

```
data(icarm_financial)
m <- icarm_fit(default ~ credit_score + income +
               loan_amount + debt_ratio,
               icarm_financial, model = "logistic",
               positive = "Yes")
```

icarm_fit

*Fit an ICARM model on any tabular data***Description**

Single unified entry point for all icarm modelling. Automatically detects the prediction task from your target variable and supports both interpretable and extended (black-box) model families.

****Task auto-detection:**** | Target type | Task | |—|—| | numeric / integer | regression | | factor / character, 2 levels | binary classification | | factor / character, 3+ levels | multi-class classification |

****Interpretable models (ICARM-compliant):**** - "cart" — Classification/Regression Tree (rpart) - "logistic" — Logistic regression (binary) - "logistic_l1" — L1-penalised logistic (glmnet) - "linear" — Linear regression - "gam" — Generalised Additive Model (mgcv) - "multinomial" — Multinomial logistic (nnet)

****Extended models (requires post-hoc explanation):**** - "random_forest" — Random forest (randomForest) - "xgboost" — Gradient boosting (xgboost) - "svm" — Support vector machine (e1071)

Usage

```
icarm_fit(
  formula,
  data,
  task = "auto",
  model = "auto",
  seed = 2025L,
  positive = NULL,
  cart_control = NULL,
  ...
)
```

Arguments

formula	A model formula, e.g. 'outcome ~ .' or 'outcome ~ x1 + x2'.
data	A 'data.frame' or 'tibble'.
task	One of "auto" (default), "binary", "multiclass", or "regression".
model	Character. Model type. Use "auto" for CART (default), or specify any model from the list above.
seed	Integer random seed for reproducibility (default 2025).
positive	Positive class label for binary classification.
cart_control	Optional [rpart::rpart.control()] for CART.
...	Additional arguments passed to the underlying fitter.

Value

An S3 object of class 'icarm_model' with full provenance.

Examples

```
# Works on any data -- task auto-detected
m1 <- icarm_fit(Species ~ ., iris)           # multiclass
m2 <- icarm_fit(Sepal.Length ~ ., iris)      # regression

# Extended models (randomForest package required)
if (requireNamespace("randomForest", quietly = TRUE)) {
  m3 <- icarm_fit(Species ~ ., iris,
                  model = "random_forest")
}

# Built-in datasets
data(icarm_medical)
m4 <- icarm_fit(readmitted ~ ., icarm_medical,
                 model = "cart")
```

icarm_learning_curve *Learning Curve Diagnostics*

Description

Trains models on progressively larger subsets of the data and records training and validation error at each size, repeated over `cv_folds` random splits. Use the resulting curve to diagnose high bias (both errors high and close) versus high variance (large gap between train and validation errors).

Usage

```
icarm_learning_curve(
  formula,
  data,
  model = "auto",
  sizes = seq(0.1, 0.9, by = 0.1),
  cv_folds = 5L,
  seed = 2025L,
  ...
)
```

Arguments

<code>formula</code>	A model formula.
<code>data</code>	A data frame.
<code>model</code>	Character model type (default "auto"). See icarm_fit .
<code>sizes</code>	Numeric vector of training-set fractions (default 0.1 to 0.9).
<code>cv_folds</code>	Integer. Repetitions per size (default 5).
<code>seed</code>	Integer seed (default 2025).
<code>...</code>	Additional arguments forwarded to icarm_fit .

Value

An `icarm_learning_curve` object with elements `curve` (tibble of training and validation errors at each size), `metric`, and `task`.

Examples

```
lc <- icarm_learning_curve(Species ~ ., iris,
                           model = "cart", cv_folds = 3L,
                           sizes = seq(0.2, 0.8, by = 0.2))
print(lc)
```

icarm_medical

Synthetic Medical Readmission Dataset

Description

A synthetic dataset of 500 hospital patients with clinical and administrative variables, designed for binary classification (predicting 30-day readmission) and fairness auditing across gender and insurance type.

Usage

```
icarm_medical
```

Format

A tibble with 500 rows and 12 variables:

age Integer. Patient age (18-90).
gender Factor. male or female.
bmi Numeric. Body mass index.
systolic_bp Integer. Systolic blood pressure (mmHg).
diastolic_bp Integer. Diastolic blood pressure (mmHg).
glucose_level Integer. Fasting glucose (mg/dL).
smoker Factor. Yes or No.
diabetes Factor. Yes or No.
insurance Factor. Public, Private, or None.
num_prior_visits Integer. Prior hospital visits.
length_of_stay Integer. Current stay in days.
readmitted Factor. Yes (readmitted within 30 days) or No.

Source

Synthetic data generated by the icarm team.

Examples

```
data(icarm_medical)
m <- icarm_fit(readmitted ~ ., icarm_medical,
               model = "logistic", positive = "Yes")
```

icarm_metrics

*Compute performance metrics for any task***Description**

Compute performance metrics for any task

Usage

```
icarm_metrics(y_true, y_pred, y_prob = NULL, positive = NULL, type = "auto")
```

Arguments

y_true	True outcome values.
y_pred	Predicted values.
y_prob	Numeric probability for positive class (binary, for AUC).
positive	Positive class label (binary classification).
type	One of "auto", "binary", "multiclass", "regression".

Value

A named numeric vector of metrics.

Examples

```
# Classification
y <- factor(c("yes", "no", "yes", "yes", "no"))
yhat <- factor(c("yes", "no", "no", "yes", "no"))
icarm_metrics(y, yhat, positive = "yes")

# Regression
icarm_metrics(c(1,2,3,4,5), c(1.1,2.2,2.9,4.1,4.8))

# Multiclass
m <- icarm_fit(Species ~ ., iris)
icarm_metrics(iris$Species, predict(m, iris))
```

icarm_pdp	<i>Partial Dependence Profile</i>
-----------	-----------------------------------

Description

Computes the Partial Dependence Plot (PDP) for a single feature by marginalising over all other features. At each grid point the feature is set to a fixed value and predictions are averaged across all observations. The 10th-90th percentile band shows individual-level variability.

Usage

```
icarm_pdp(object, data, feature, n_grid = 20L, seed = 2025L)
```

Arguments

- object An icarm_model from [icarm_fit](#).
- data A data frame used for marginalisation.
- feature Character. Name of the feature to profile.
- n_grid Integer. Grid resolution for numeric features (default 20).
- seed Integer seed (default 2025).

Value

An icarm_pdp object with element pdp (a tibble with columns feature_value, mean_pred, lower, upper).

Examples

```
m <- icarm_fit(Sepal.Length ~ ., iris, model = "linear")
pdp <- icarm_pdp(m, iris, feature = "Petal.Length")
print(pdp)
```

icarm_plot_calibration	<i>Plot calibration curve</i>
------------------------	-------------------------------

Description

Plot calibration curve

Usage

```
icarm_plot_calibration(calibration, title = NULL)
```

Arguments

calibration	An 'icarm_calibration' from [icarm_calibrate()].
title	Optional title.

Value

A ggplot2 object.

icarm_plot_comparison *Plot multi-model comparison*

Description

Plot multi-model comparison

Usage

```
icarm_plot_comparison(  
  comparison,  
  metrics = c("accuracy", "f1", "max_tpr_gap", "min_dp_ratio"),  
  title = NULL  
)
```

Arguments

comparison	An 'icarm_comparison' from [icarm_compare()].
metrics	Character vector of metric columns.
title	Optional title.

Value

A ggplot2 object.

icarm_plot_confusion *Plot confusion matrix*

Description

Plot confusion matrix

Usage

```
icarm_plot_confusion(y_true, y_pred, title = NULL)
```

Arguments

<code>y_true</code>	Factor of true outcomes.
<code>y_pred</code>	Factor of predicted outcomes.
<code>title</code>	Optional title.

Value

A ggplot2 object.

Examples

```
m <- icarm_fit(Species ~ ., iris)
yhat <- predict(m, iris)
print(icarm_plot_confusion(iris$Species, yhat))
```

<code>icarm_plot_fairness</code>	<i>Plot group-level fairness metric</i>
----------------------------------	---

Description

Plot group-level fairness metric

Usage

```
icarm_plot_fairness(fairness, metric = "acc", title = NULL, ref_line = NULL)
```

Arguments

<code>fairness</code>	An 'icarm_fairness' from [icarm_fairness()].
<code>metric</code>	Character. Column to plot.
<code>title</code>	Optional title.
<code>ref_line</code>	Optional numeric reference line.

Value

A ggplot2 object.

Examples

```
m <- icarm_fit(Species ~ ., iris)
idf <- iris
idf$size <- factor(ifelse(idf$Sepal.Length > 5.8, "large", "small"))
f <- icarm_fairness(m, idf, "Species", "size")
print(icarm_plot_fairness(f, metric = "acc"))
```

`icarm_plot_importance` *Plot feature importance*

Description

Plot feature importance

Usage

```
icarm_plot_importance(explainer, n_features = 15L, title = NULL)
```

Arguments

<code>explainer</code>	An 'icarm_explainer' from <code>[icarm_explain()]</code> .
<code>n_features</code>	Max features to display (default 15).
<code>title</code>	Optional plot title.

Value

A ggplot2 object.

Examples

```
m <- icarm_fit(Species ~ ., iris)
ex <- icarm_explain(m)
print(icarm_plot_importance(ex))
```

`icarm_plot_learning_curve`
Plot a Learning Curve

Description

Plots training and validation error against training set size, with +/- 1 SD bands. A large gap indicates high variance (overfitting); both curves high and close together indicates high bias (underfitting).

Usage

```
icarm_plot_learning_curve(lc, title = NULL)
```

Arguments

<code>lc</code>	An <code>icarm_learning_curve</code> from icarm_learning_curve .
<code>title</code>	Optional plot title.

Value

A ggplot2 object.

Examples

```
lc <- icarm_learning_curve(Species ~ ., iris, model = "cart",
                           cv_folds = 3L,
                           sizes = seq(0.2, 0.8, by = 0.2))
p <- icarm_plot_learning_curve(lc)
```

icarm_plot_pdp

Plot a Partial Dependence Profile

Description

Plots the Partial Dependence Profile for one feature. For numeric features a line with a 10th-90th percentile ribbon is shown; for categorical features a dot-and-errorbar chart is shown.

Usage

```
icarm_plot_pdp(pdp, title = NULL)
```

Arguments

pdp	An icarm_pdp from icarm_pdp .
title	Optional plot title.

Value

A ggplot2 object.

Examples

```
m <- icarm_fit(Sepal.Length ~ ., iris, model = "linear")
pdp <- icarm_pdp(m, iris, feature = "Petal.Length")
p <- icarm_plot_pdp(pdp)
```

icarm_plot_roc_groups *Plot per-group ROC curves*

Description

Plot per-group ROC curves

Usage

```
icarm_plot_roc_groups(eoc_tbl, title = NULL)
```

Arguments

eoc_tbl	A tibble from <code>[icarm_equalized_odds_curve()]</code> .
title	Optional title.

Value

A ggplot2 object.

icarm_plot_shap *Beeswarm SHAP plot*

Description

Displays SHAP values as a beeswarm-style scatter. Each point is one observation; the x-axis shows the SHAP value; colour encodes the feature value (blue = low, red = high).

Usage

```
icarm_plot_shap(shap, n_features = 10L, title = NULL)
```

Arguments

shap	An icarm_shap from icarm_shap .
n_features	Max features to show (default 10).
title	Optional plot title.

Value

A ggplot2 object.

Examples

```
m <- icarm_fit(Sepal.Length ~ ., iris, model = "linear")
shap <- icarm_shap(m, iris[1:30, ], n_samples = 20L)
p <- icarm_plot_shap(shap)
```

icarm_plot_thresholds *Plot threshold performance curves*

Description

Plot threshold performance curves

Usage

```
icarm_plot_thresholds(  
  thresholds_tbl,  
  metrics = c("accuracy", "recall", "precision", "f1"),  
  title = NULL  
)
```

Arguments

`thresholds_tbl` A tibble from [icarm_thresholds()].
`metrics` Character vector of metric columns.
`title` Optional title.

Value

A ggplot2 object.

icarm_racism_survey *Synthetic Racism and Civic Participation Survey*

Description

A synthetic dataset of 150 individuals capturing experiences of racism, policing, migration, and civic participation across multiple demographic groups. Suitable for fairness auditing, regression, binary, and multi-class classification.

Usage

```
icarm_racism_survey
```

Format

A tibble with 150 rows and 16 variables:

age Integer. Age in years (18-75).
gender Factor. male, female, diverse.
hair_color Factor. blonde, brown, black, other.
skin_color Factor. light, medium, dark.
relationship_status Factor. single, engaged, married.
racism_impact Integer 0-10. Perceived racism impact.
police_stop Factor. 0 times, 1 time, 2 or more.
migrant_status Factor. Yes or No.
income Numeric. Monthly income (EUR).
education_level Ordered factor. Education level.
employment_status Factor. Employment category.
area_type Factor. Urban or Rural.
religion Factor. Religion category.
language_proficiency Ordered factor. Language level.
number_of_friends Integer 0-10.
political_orientation Integer 0-3 (left to right).

Source

Synthetic data generated by the icarm team.

Examples

```
data(icarm_racism_survey)
m <- icarm_fit(racism_impact ~ ., icarm_racism_survey,
               model = "linear")
```

icarm_scorecard

Generate a full accountability scorecard

Description

Generate a full accountability scorecard

Usage

```
icarm_scorecard(
  object,
  test_data,
  outcome,
  protected = NULL,
  positive = NULL,
  analyst = NULL,
  project = "icarm",
  path = NULL
)
```

Arguments

object	An 'icarm_model'.
test_data	Data frame of test data.
outcome	Character outcome column.
protected	Optional protected attribute column.
positive	Positive class (binary).
analyst	Character analyst name.
project	Character project name.
path	Optional JSON output path.

Value

Invisibly, the scorecard list.

Examples

```
sp <- icarm_split(iris, stratify = "Species")
m <- icarm_fit(Species ~ ., sp$train)
iris_test <- sp$test
iris_test$size <- factor(
  ifelse(iris_test$Sepal.Length > 5.8, "large", "small"))
icarm_scorecard(m, iris_test, outcome="Species",
  protected="size", project="Iris Demo")
```

icarm_shap

SHAP-style local feature attributions

Description

Computes approximate SHAP (SHapley Additive exPlanations) values using a marginal interventional approach. For each observation and each feature the SHAP value is the average change in prediction when that feature is set to the observed value versus its background distribution. Works with any `icarm_model` without additional packages.

Usage

```
icarm_shap(object, data, n_samples = 50L, max_obs = 100L, seed = 2025L)
```

Arguments

object	An icarm_model from icarm_fit .
data	A data frame of observations to explain.
n_samples	Integer. Background sample size (default 50).
max_obs	Integer. Maximum observations to explain (default 100).
seed	Integer seed for reproducibility (default 2025).

Value

An icarm_shap object with elements shap (long-format tibble of SHAP values), importance (features ranked by mean |SHAP|), and baseline (mean prediction over the background sample).

Examples

```
m <- icarm_fit(Sepal.Length ~ ., iris, model = "linear")
shap <- icarm_shap(m, iris[1:20, ], n_samples = 20L)
print(shap)
```

icarm_split	<i>Reproducible train/test split</i>
-------------	--------------------------------------

Description

Reproducible train/test split

Usage

```
icarm_split(data, prop = 0.75, seed = 2025L, stratify = NULL)
```

Arguments

data	A data.frame or tibble.
prop	Proportion for training (default 0.75).
seed	Integer seed (default 2025).
stratify	Optional column name for stratified split.

Value

A named list with train, test, seed, prop.

Examples

```
splits <- icarm_split(iris, prop = 0.8, stratify = "Species")
nrow(splits$train)
```

icarm_thresholds	<i>Threshold sweep for binary classification</i>
------------------	--

Description

Threshold sweep for binary classification

Usage

```
icarm_thresholds(
  y_true,
  y_prob,
  positive = NULL,
  thresholds = seq(0.1, 0.9, 0.05)
)
```

Arguments

y_true	Factor of true class labels.
y_prob	Numeric probability vector for positive class.
positive	Positive class label.
thresholds	Numeric vector of thresholds to evaluate.

Value

A tibble with one row per threshold.

Examples

```
set.seed(42)
y <- factor(sample(c("yes", "no"), 200, replace = TRUE))
p <- runif(200)
thr <- icarm_thresholds(y, p, positive = "yes")
print(icarm_plot_thresholds(thr))
```

predict.icarm_model	<i>Predict from an icarm_model</i>
---------------------	------------------------------------

Description

Predict from an icarm_model

Usage

```
## S3 method for class 'icarm_model'
predict(object, newdata, type = c("class", "prob"), threshold = 0.5, ...)
```


Arguments

object	An 'icarm_model'.
newdata	A data frame for prediction.
type	For classification: "class" or "prob". For regression: ignored.
threshold	Decision threshold for binary (default 0.5).
...	Ignored.

Value

Factor vector, probability matrix, or numeric vector.

Examples

```
m <- icarm_fit(Species ~ ., iris)
predict(m, iris[1:5, ], type = "class")
```

print.icarm_model	<i>Print an icarm_model</i>
-------------------	-----------------------------

Description

Print an icarm_model

Usage

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

Arguments

x	An icarm_model object.
...	Further arguments passed to or from other methods.

Value

Invisibly returns the icarm_model object x. Called for its side effect of printing a formatted summary to the console.

summary.icarm_model	<i>Summary of an icarm_model</i>
---------------------	----------------------------------

Description

Summary of an icarm_model

Usage

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

Arguments

object	An icarm_model object.
...	Further arguments passed to or from other methods.

Value

Invisibly returns the summary of the underlying fitted model object. Called for its side effect of printing a detailed model summary to the console.

Index

* datasets

- icarm_financial, [9](#)
- icarm_medical, [12](#)
- icarm_racism_survey, [20](#)

- icarm_audit, [3](#)
- icarm_calibrate, [4](#)
- icarm_compare, [4](#)
- icarm_equalized_odds_curve, [5](#)
- icarm_equity_summary, [6](#)
- icarm_explain, [7](#)
- icarm_explain_local, [7](#)
- icarm_fairness, [8](#)
- icarm_financial, [9](#)
- icarm_fit, [10](#), [11](#), [14](#), [23](#)
- icarm_learning_curve, [11](#), [17](#)
- icarm_medical, [12](#)
- icarm_metrics, [13](#)
- icarm_pdp, [14](#), [18](#)
- icarm_plot_calibration, [14](#)
- icarm_plot_comparison, [15](#)
- icarm_plot_confusion, [15](#)
- icarm_plot_fairness, [16](#)
- icarm_plot_importance, [17](#)
- icarm_plot_learning_curve, [17](#)
- icarm_plot_pdp, [18](#)
- icarm_plot_roc_groups, [19](#)
- icarm_plot_shap, [19](#)
- icarm_plot_thresholds, [20](#)
- icarm_racism_survey, [20](#)
- icarm_scorecard, [21](#)
- icarm_shap, [19](#), [22](#)
- icarm_split, [23](#)
- icarm_thresholds, [24](#)
- predict.icarm_model, [24](#)
- print.icarm_learning_curve
 (icarm_learning_curve), [11](#)
- print.icarm_model, [25](#)
- print.icarm_pdp (icarm_pdp), [14](#)

- print.icarm_shap (icarm_shap), [22](#)

- summary.icarm_model, [26](#)