Package: icarm (via r-universe)

July 1, 2026

Title Interpretable Contextual-Accountable and Responsible Machine Learning

Version 0.1.0

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, 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, vip, 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 https://cran.r-universe.dev

Date/Publication 2026-06-30 20:40:10 UTC

RemoteUrl https://github.com/cran/icarm

RemoteRef HEAD

RemoteSha 3dbe08ce66055cd0516c95789d557617e45e471b

Contents

`icarm_audit` *Generate a JSON audit trail*

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` *Probability calibration diagnostics*

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` *Compare multiple icarm_models*

Description

Compare multiple icarm_models

Usage

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

Arguments

`models`	A named list of ‘icarm_model‘ objects.
`test_data`	A data frame for evaluation.
`outcome`	Character outcome column.
`protected`	Optional protected attribute for fairness.
`positive`	Positive class (binary).
`threshold`	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$train, model = "cart")
m2 <- icarm_fit(Species ~ ., sp$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` *Equity summary from a fairness report*

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` *Generate global model explanations*

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)
icarm_plot_importance(ex)
```

`icarm_explain_local` *Local explanation for individual observations*

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` *Group-level fairness audit*

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

`icarm_financial` *Synthetic Financial Loan Default Dataset*

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
m3 <- icarm_fit(Species ~ ., iris,
                model = "random_forest")

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

`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_plot_calibration` *Plot calibration curve*

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

`y_true`	Factor of true outcomes.
`y_pred`	Factor of predicted outcomes.
`title`	Optional title.

Value

A ggplot2 object.

Examples

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

`icarm_plot_fairness` *Plot group-level fairness metric*

Description

Plot group-level fairness metric

Usage

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

Arguments

`fairness`	An ‘icarm_fairness‘ from [icarm_fairness()].
`metric`	Character. Column to plot.
`title`	Optional title.
`ref_line`	Optional numeric reference line.

Value

A ggplot2 object.

Examples

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

`explainer`	An ‘icarm_explainer‘ from [icarm_explain()].
`n_features`	Max features to display (default 15).
`title`	Optional plot title.

Value

A ggplot2 object.

Examples

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

`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 [icarm_equalized_odds_curve()].
`title`	Optional title.

Value

A ggplot2 object.

`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_split` *Reproducible train/test split*

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` *Threshold sweep for binary classification*

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

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

`predict.icarm_model` *Predict from an icarm_model*

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` *Print an icarm_model*

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` *Summary of an icarm_model*

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