Package 'clinicalfair'

April 2, 2026

Title Algorithmic Fairness Assessment for Clinical Prediction Models

Version 0.1.0

Description Post-hoc fairness auditing toolkit for clinical prediction models. Unlike in-processing approaches that modify model training, this package evaluates existing models by computing group-wise fairness metrics (demographic parity, equalized odds, predictive parity, calibration disparity), visualizing disparities across protected attributes, and performing threshold-based mitigation. Supports intersectional analysis across multiple attributes and generates audit reports useful for fairness-oriented auditing in clinical AI settings.
Methods described in Obermeyer et al. (2019) <doi:10.1126/science.aax2342> and Hardt, Price, and Srebro (2016) <doi:10.48550/arXiv.1610.02413>.

License MIT + file LICENSE

URL https://github.com/CuiweiG/clinicalfair

BugReports https://github.com/CuiweiG/clinicalfair/issues

Depends R (>= 4.1.0)

Imports cli (>= 3.4.0), dplyr (>= 1.1.0), ggplot2 (>= 3.4.0), rlang (>= 1.1.0), stats, tibble (>= 3.1.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Config/testthat/edition 3

Language en-US

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

NeedsCompilation no

Author Cuiwei Gao [aut, cre, cph]

Maintainer Cuiwei Gao <48gaocuiwei@gmail.com>

Repository CRAN

Date/Publication 2026-04-02 20:10:09 UTC

Contents

`autoplot.fairness_metrics` *Plot fairness metrics disparity*

Description

Plot fairness metrics disparity

Usage

```
## S3 method for class 'fairness_metrics'
autoplot(object, type = c("disparity", "roc", "calibration"), ...)
```

Arguments

`object`	A `fairness_metrics` object.
`type`	Plot type: `"disparity"` (default), `"roc"`, or `"calibration"`.
`...`	Additional arguments (unused).

Value

A ggplot object.

Examples

```
set.seed(42)
fd <- fairness_data(
  predictions = c(runif(100, 0.2, 0.8), runif(100, 0.3, 0.9)),
  labels = c(rbinom(100, 1, 0.3), rbinom(100, 1, 0.5)),
  protected_attr = rep(c("A", "B"), each = 100)
)
fm <- fairness_metrics(fd)
autoplot(fm)
```

---

compas_sim *Simulated COMPAS-like recidivism prediction data*

Description

A simulated dataset reflecting the documented racial disparities in recidivism prediction algorithms, based on published statistics from the ProPublica investigation (Angwin et al. 2016).

Usage

```
compas_sim
```

Format

A data frame with 1000 rows and 3 columns:

risk_score Predicted recidivism risk (numeric, 0–1).

recidivism Actual recidivism outcome (binary, 0/1).

race Racial group: White or Black (character).

Source

Simulated. Based on patterns from Angwin et al. (2016) "Machine Bias" and Obermeyer et al. (2019) doi:10.1126/science.aax2342.

Examples

```
data(compas_sim)
fd <- fairness_data(compas_sim$risk_score, compas_sim$recidivism,
                    compas_sim$race)
fairness_metrics(fd)
```

`fairness_data` *Create a fairness evaluation data object*

Description

Bundles predictions, labels, and protected attributes into a standardized container for fairness analysis.

Usage

```
fairness_data(
  predictions,
  labels,
  protected_attr,
  threshold = 0.5,
  reference_group = NULL
)
```

Arguments

`predictions`	Numeric vector of predicted probabilities or risk scores (between 0 and 1).
`labels`	Binary integer vector of true outcomes (0 or 1).
`protected_attr`	Character or factor vector identifying the protected group membership (e.g., race, sex, age group).
`threshold`	Decision threshold for converting probabilities to binary predictions. Default 0.5.
`reference_group`	Name of the reference (privileged) group. If `NULL`, the group with the highest selection rate is used.

Value

A `fairness_data` object (list) with standardized components: `predictions`, `labels`, `protected`, `threshold`, `predicted_class`, `reference_group`, `groups`, `n`, `prevalence`.

Examples

```
set.seed(42)
fd <- fairness_data(
  predictions = runif(200),
  labels = rbinom(200, 1, 0.3),
  protected_attr = sample(c("GroupA", "GroupB"), 200, replace = TRUE)
)
fd
```

`fairness_metrics` *Compute fairness metrics across groups*

Description

Calculates a comprehensive set of group-wise and comparative fairness metrics from a `fairness_data` object, with optional bootstrap confidence intervals.

Usage

```
fairness_metrics(
  data,
  metrics = c("selection_rate", "tpr", "fpr", "ppv", "accuracy", "auc", "brier"),
  ci = FALSE,
  n_boot = 2000L,
  ci_level = 0.95
)
```

Arguments

`data`	A fairness_data object.
`metrics`	Character vector of metrics to compute. Default computes all available metrics. Options: `"selection_rate"`, `"tpr"`, `"fpr"`, `"ppv"`, `"accuracy"`, `"auc"`, `"brier"`.
`ci`	Logical; if `TRUE`, compute bootstrap confidence intervals for each metric. Default `FALSE`.
`n_boot`	Number of bootstrap replicates when `ci = TRUE`. Default 2000.
`ci_level`	Confidence level for the interval. Default 0.95.

Details

Fairness is assessed by comparing metric values across groups. A ratio of 1.0 or difference of 0.0 indicates perfect parity. Common thresholds: ratio in $[0.8, 1.25]$ (four-fifths rule, EEOC guidelines) or difference < 0.05.

When `ci = TRUE`, percentile bootstrap confidence intervals are computed by resampling within each group. This accounts for sampling variability and is recommended when reporting fairness metrics for regulatory or publication purposes.

Value

A `fairness_metrics` object (tibble) with columns: `group`, `metric`, `value`, `ratio` (vs reference group), `difference` (vs reference group). When `ci = TRUE`, additional columns `ci_lower` and `ci_upper` are included.

References

Obermeyer Z, et al. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453. doi:10.1126/science.aax2342

Examples

```
set.seed(42)
fd <- fairness_data(
  predictions = c(runif(100, 0.2, 0.8), runif(100, 0.3, 0.9)),
  labels = c(rbinom(100, 1, 0.3), rbinom(100, 1, 0.5)),
  protected_attr = rep(c("A", "B"), each = 100)
)
fairness_metrics(fd)

# With bootstrap CIs
fairness_metrics(fd, ci = TRUE, n_boot = 500)
```

---

fairness_report *Generate a fairness summary report*

Description

Generate a fairness summary report

Usage

```
fairness_report(data, metrics = NULL)
```

Arguments

data	A fairness_data object.
metrics	A fairness_metrics object. If NULL, computed automatically.

Value

A fairness_report (list) with $summary, $flags, $recommendation.

Examples

```
set.seed(42)
fd <- fairness_data(
  predictions = c(runif(100, 0.2, 0.8), runif(100, 0.3, 0.9)),
  labels = c(rbinom(100, 1, 0.3), rbinom(100, 1, 0.5)),
  protected_attr = rep(c("A", "B"), each = 100)
)
fairness_report(fd)
```

`intersectional_fairness` *Compute intersectional fairness metrics*

Description

Evaluates fairness across combinations of multiple protected attributes (e.g., race x sex), revealing disparities hidden by single-attribute analysis.

Usage

```
intersectional_fairness(
  predictions,
  labels,
  ...,
  threshold = 0.5,
  min_group_size = 10L
)
```

Arguments

`predictions`	Numeric vector of predicted probabilities.
`labels`	Binary integer vector of true outcomes.
`...`	Two or more named vectors of protected attributes. Names become the attribute labels.
`threshold`	Decision threshold. Default 0.5.
`min_group_size`	Minimum number of observations required per intersectional group. Groups below this threshold are dropped with a warning. Default 10.

Value

A `fairness_metrics` object with intersectional groups. Groups with fewer than `min_group_size` observations are excluded.

References

Buolamwini J, Gebru T (2018). Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. *Conference on Fairness, Accountability and Transparency*.

Examples

```
set.seed(42)
n <- 400
intersectional_fairness(
  predictions = runif(n),
  labels = rbinom(n, 1, 0.3),
  race = sample(c("White", "Black"), n, replace = TRUE),
```

```
  sex = sample(c("Male", "Female"), n, replace = TRUE)
)
```

`plot_calibration` *Plot calibration curves by group*

Description

Assesses whether predicted probabilities match observed event rates within each protected group.

Usage

```
plot_calibration(data, n_bins = 10L)
```

Arguments

`data`	A fairness_data object.
`n_bins`	Number of calibration bins. Default 10.

Value

A ggplot object.

Examples

```
data(compas_sim)
fd <- fairness_data(compas_sim$risk_score, compas_sim$recidivism,
                    compas_sim$race)
plot_calibration(fd)
```

`plot_roc` *Plot ROC curves by group*

Description

Plot ROC curves by group

Usage

```
plot_roc(data)
```

Arguments

`data`	A fairness_data object.

Value

A ggplot object.

Examples

```
set.seed(42)
fd <- fairness_data(
  predictions = c(runif(100, 0.2, 0.8), runif(100, 0.3, 0.9)),
  labels = c(rbinom(100, 1, 0.3), rbinom(100, 1, 0.5)),
  protected_attr = rep(c("A", "B"), each = 100)
)
plot_roc(fd)
```

---

threshold_optimize *Optimize thresholds for fairness*

Description

Finds group-specific decision thresholds that maximize accuracy subject to a fairness constraint, or minimize disparity subject to a minimum accuracy constraint.

Usage

```
threshold_optimize(
  data,
  objective = c("equalized_odds", "demographic_parity"),
  min_accuracy = 0.5,
  grid_resolution = 0.01
)
```

Arguments

`data`	A fairness_data object.
`objective`	`"equalized_odds"` (default): minimize TPR/FPR disparity across all groups. `"demographic_parity"`: equalize selection rates.
`min_accuracy`	Minimum acceptable overall accuracy. Default 0.5.
`grid_resolution`	Step size for the threshold grid search. Default 0.01 (99 candidate thresholds). Smaller values give finer-grained optimization at modest computational cost.

Details

This implements post-processing threshold adjustment, the simplest and most transparent mitigation strategy. Each group receives its own threshold to equalize the chosen fairness criterion.

For `"equalized_odds"`, the algorithm computes a pooled target TPR and FPR across all groups at the original threshold, then optimizes every group (including the reference) to match the pooled target. This avoids the asymmetry of fixing the reference group threshold while only adjusting others.

For clinical applications, group-specific thresholds are interpretable and auditable, unlike in-processing methods that modify the model itself.

Value

A `fairness_mitigation` object (list) with: `$thresholds` (named numeric, one per group), `$before` and `$after` (fairness_metrics objects), `$accuracy_before` and `$accuracy_after`.

References

Hardt M, Price E, Srebro N (2016). Equality of Opportunity in Supervised Learning. *NeurIPS*.

Examples

```
data(compas_sim)
fd <- fairness_data(compas_sim$risk_score, compas_sim$recidivism,
                    compas_sim$race)
mit <- threshold_optimize(fd)
mit
```

Index