Package 'TrustworthyMLR'

May 7, 2026

Type Package

Title Stability and Robustness Evaluation for Machine Learning Models

Version 0.1.0

Description Provides tools for evaluating the trustworthiness of machine learning models in production and research settings. Computes a Stability Index that quantifies the consistency of model predictions across multiple runs or resamples, and a Robustness Score that measures model resilience under small input perturbations. Designed for data scientists, ML engineers, and researchers who need to monitor and ensure model reliability, reproducibility, and deployment readiness.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Ali Hamza [aut, cre]

Maintainer Ali Hamza <ahamza.msse25mcs@student.nust.edu.pk>

Repository CRAN

Date/Publication 2026-02-20 10:40:36 UTC

Contents

classification_stability *Stability Index for Classification Models*

Description

Computes the stability of classification predictions across multiple runs. For classification, stability is measured as the average agreement between pairs of runs, adjusted for chance (similar to Cohen's Kappa but extended for multiple runs).

Usage

```
classification_stability(class_matrix)
```

Arguments

class_matrix	A matrix or data.frame where each row represents an observation and each column represents a predicted class (factor or character) from a single model run.

Value

A numeric scalar between 0 and 1, where 1 indicates perfect consistency and 0 indicates consistency no better than chance.

Examples

```
# Simulate classification predictions from 3 runs
preds <- data.frame(
    run1 = c("A", "A", "B", "C"),
    run2 = c("A", "B", "B", "C"),
    run3 = c("A", "A", "B", "C")
)
classification_stability(preds)
```

plot_robustness *Plot Robustness Decay Curve*

Description

Visualizes how model performance (robustness) decreases as the level of input noise increases. This "decay curve" is a powerful tool for understanding the sensitivity threshold of a machine learning model.

Usage

```
plot_robustness(
  predict_fn,
  X,
  levels = seq(0, 0.3, by = 0.05),
  n_rep = 5L,
  ...
)
```

Arguments

predict_fn	A function that accepts a numeric matrix and returns a numeric vector of predictions.
X	A numeric matrix or data.frame of input features.
levels	A numeric vector of noise levels to evaluate. Default is seq(0, 0.3, by = 0.05).
n_rep	Number of repetitions for each noise level. Default is 5L.
...	Additional arguments passed to plot.

Value

A data.frame with columns noise_level and robustness_score.

Examples

```
# Simple model
pred_fn <- function(X) X %*% c(1, -1)
X <- matrix(rnorm(200), ncol = 2)

# Plot decay
plot_robustness(pred_fn, X, main = "Model Robustness Decay")
```

---

plot_stability *Plot Stability of Model Predictions*

Description

Creates a visualization showing the variability of model predictions across multiple runs. This helps identify whether instability is uniform across the dataset or concentrated on specific observations.

Usage

```
plot_stability(predictions_matrix, type = c("range", "sd"), ...)
```

Arguments

predictions_matrix	A numeric matrix or data.frame where each row represents an observation and each column represents predictions from a single model run or resample.
type	Character string indicating what the error bars represent. Either "range" (default) or "sd" (standard deviation).
...	Additional arguments passed to plot.

Details

The plot displays the mean prediction for each observation with error bars representing the range (minimum and maximum) or standard deviation of predictions across runs.

Value

No return value, called for side effects (plotting).

Examples

```
# Simulate predictions from 5 model runs
set.seed(42)
base_predictions <- sort(rnorm(50))
predictions <- matrix(
    rep(base_predictions, 5) + rnorm(250, sd = 0.2),
    ncol = 5
)

plot_stability(predictions, main = "Model Prediction Stability")
```

---

robustness_score *Robustness Score Under Input Perturbation*

Description

Evaluates the robustness of a machine learning model by measuring how much its predictions change when small amounts of noise are added to the input data. A robustness score of 1 indicates that predictions are completely unaffected by perturbations, while values near 0 indicate high sensitivity to input noise.

Usage

```
robustness_score(predict_fn, X, noise_level = 0.05, n_rep = 10L)
```

Arguments

predict_fn	A function that accepts a numeric matrix (observations in rows, features in columns) and returns a numeric vector of predictions with length equal to nrow(X).
X	A numeric matrix or data.frame of input features. Rows are observations and columns are features. Must contain at least two rows and no missing values.
noise_level	A positive numeric scalar controlling the magnitude of Gaussian noise added to each feature, expressed as a fraction of the feature's standard deviation. Default is 0.05 (5 percent).
n_rep	A positive integer specifying the number of perturbation repetitions. Default is 10L.

Details

Gaussian noise proportional to each feature's standard deviation is added to the input data. The magnitude of the noise is controlled by noise_level. Predictions on the perturbed data are compared to baseline predictions using normalised mean squared error. The process is repeated n_rep times and the average score is returned.

Value

A numeric scalar between 0 and 1, where 1 indicates perfect robustness and values near 0 indicate high sensitivity to noise.

Examples

```
# A simple linear prediction function
pred_fn <- function(X) X %*% c(1, 2, 3)
set.seed(42)
X <- matrix(rnorm(300), ncol = 3)
robustness_score(pred_fn, X, noise_level = 0.05, n_rep = 10)

# A constant prediction function is perfectly robust
const_fn <- function(X) rep(5, nrow(X))
robustness_score(const_fn, X)
```

---

stability_index *Stability Index for Model Predictions*

Description

Computes a Stability Index that quantifies the consistency of machine learning model predictions across multiple runs or resamples. A stability index of 1 indicates perfectly consistent predictions, while values closer to 0 indicate high variability across runs.

Usage

```
stability_index(predictions_matrix)
```

Arguments

`predictions_matrix`
: A numeric matrix or data.frame where each row represents an observation and each column represents predictions from a single model run or resample. Must contain at least two columns and no missing values.

Details

The index is calculated by comparing the mean per-observation variance across runs to the overall variance of all predictions. Low per-observation variance relative to overall variance indicates that the model produces consistent results regardless of the specific training run or resample.

Value

A numeric scalar between 0 and 1, where 1 indicates perfect stability (identical predictions across all runs) and values near 0 indicate high instability.

Examples

```
# Simulate predictions from 5 model runs for 100 observations
set.seed(42)
base_predictions <- rnorm(100)
predictions <- matrix(
  rep(base_predictions, 5) + rnorm(500, sd = 0.1),
  ncol = 5
)
stability_index(predictions)

# Perfectly stable predictions yield an index of 1
stable_preds <- matrix(rep(1:10, 3), ncol = 3)
stability_index(stable_preds)
```

Index