Package 'deriva'

August 3, 2026

Title Tidy Drift Detection for Monitored Machine Learning Models

Version 0.1.0

Description Detects concept drift and data drift in streams produced by deployed machine learning models, using a tidy interface that composes with the 'tidymodels' ecosystem. Detectors are specified, fitted on a baseline period, and advanced over new batches of observations, returning tibbles annotated with warning and drift flags. A catalogue of 22 sequential drift detectors is provided. Error-based methods include the Drift Detection Method (DDM) of Gama et al. (2004) <doi:10.1007/978-3-540-28645-5_29>, the Early Drift Detection Method (EDDM) of Baena-Garcia et al. (2006), the Hoeffding's inequality based Drift Detection Methods (HDDM) of Frias-Blanco et al. (2015) <doi:10.1109/TKDE.2014.2345382>, and the Exponentially Weighted Moving Average (EWMA) chart of Ross et al. (2012) <doi:10.1016/j.patrec.2011.08.019>. Distribution-based methods include Adaptive Windowing (ADWIN) of Bifet and Gavalda (2007) <doi:10.1137/1.9781611972771.42>, Kolmogorov-Smirnov Windowing (KSWIN) of Raab et al. (2020) <doi:10.1016/j.neucom.2019.11.111>, and the Page-Hinkley test of Page (1954) <doi:10.1093/biomet/41.1-2.100>.

License MIT + file LICENSE

URL https://github.com/bonijoao/deriva

BugReports https://github.com/bonijoao/deriva/issues

Encoding UTF-8

RoxygenNote 7.3.3

Imports cli, generics, rlang, stats, tibble, vctrs

Suggests ggplot2, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author João Paulo Assis Bonifácio [aut, cre] (ORCID:
<https://orcid.org/0009-0009-2335-8391>),
Geraldo Magela da Cruz Pereira [aut] (ORCID:
<https://orcid.org/0000-0001-6280-4870>),
Pedro Mambelli Fernandes [aut] (ORCID:
<https://orcid.org/0009-0001-1695-279X>)

Maintainer João Paulo Assis Bonifácio <jpab.27@hotmail.com>

Repository CRAN

Date/Publication 2026-08-03 17:40:07 UTC

Contents

`add_prediction_error` *Build a drift signal from model predictions*

Description

Bridge from tidymodels: takes the output of `augment()` on a fitted workflow/model and adds a `.error` column — the signal drift detectors consume. Classification (factor/character truth): 0/1 mismatch against `estimate` (default column `.pred_class`). Regression (numeric truth): absolute error against `estimate` (default column `.pred`).

Usage

```
add_prediction_error(data, truth, estimate = NULL, ...)
```

Arguments

`data`	A data frame with truth and prediction columns.
`truth`	Unquoted name of the true outcome column.
`estimate`	Unquoted name of the prediction column. Defaults to `.pred_class` (classification) or `.pred` (regression), following tidymodels conventions.
`...`	Not used.

Value

`data` as a tibble with a `.error` column added.

Examples

```
d <- tibble::tibble(truth = c(1, 2, 3), .pred = c(1, 1, 5))
add_prediction_error(d, truth = truth)
```

---

advance *Advance a fitted drift detector over a new batch*

Description

Feeds a new batch of observations (any size, including 1 — stream mode) to the detector and returns a NEW fitted object with the engine state advanced and the annotated batch appended to the history. The original object is not modified. This is the only way to persist state; see `augment()` for a read-only preview.

Usage

```
advance(object, ...)

## S3 method for class 'drift_detector_fit'
advance(object, new_data, ...)
```

Arguments

`object`	A `drift_detector_fit`.
`...`	Passed to methods.
`new_data`	A data frame with the new batch, in temporal order, containing the same signal column used in `fit()`.

Details

Why not `update()`: in the tidymodels ecosystem `update()` on a spec means "change hyperparameters", so deriva defines its own verb.

Value

A new `drift_detector_fit`.

Examples

```
base <- sim_drift_stream(n_pre = 100, n_post = 0, seed = 1)
f0 <- fit(drift_detector("ddm"), base, signal = error)
f1 <- advance(f0, sim_drift_stream(n_pre = 0, n_post = 50, seed = 2))
```

augment.drift_detector_fit

Annotated observations from a fitted drift detector

Description

With `new_data = NULL`, returns the accumulated history (baseline + advanced batches) annotated with `.warning`, `.drift` and `.phase`. With `new_data`, returns a READ-ONLY preview: the batch annotated from the current state, WITHOUT persisting it — use `advance()` to persist.

Usage

```
## S3 method for class 'drift_detector_fit'
augment(x, new_data = NULL, ...)
```

Arguments

`x`	A `drift_detector_fit`.
`new_data`	Optional data frame with a new batch to preview.
`...`	Not used.

Value

A tibble.

Examples

```
base <- sim_drift_stream(n_pre = 100, n_post = 0, seed = 1)
f0 <- fit(drift_detector("ddm"), base, signal = error)
augment(f0)
```

autoplot.drift_detector_fit

Plot the monitored signal with drift markings

Description

Plots the running mean of the signal over the full history, with the baseline/stream boundary (labelled "training ends"), warning points (orange) and drift points (red vertical lines, the first one labelled with its index). Requires ggplot2 (Suggests).

Usage

```
## S3 method for class 'drift_detector_fit'
autoplot(object, ...)
```

Arguments

`object`	A `drift_detector_fit`.
`...`	Not used.

Value

A ggplot object.

`detect_drift` *Detect drift in a signal column (one-shot shortcut)*

Description

Layer-3 convenience: runs a detector over an existing signal column and returns the data annotated with `.warning` / `.drift`. For an explicit baseline and persistent state, use the full object path: `drift_detector()` + `fit()` + `advance()`.

Usage

```
detect_drift(data, .col, method = "ddm", ...)
```

Arguments

`data`	A data frame in temporal order.
`.col`	Unquoted name of the signal column.
`method`	Name of a registered method (default `"ddm"`).
`...`	Hyperparameters forwarded to `drift_detector()`.

Details

For `"ddm"`, the warm-up is governed by `min_instances`: the first `min_instances - 1` observations get `NA` flags.

Value

`data` as a tibble with `.warning` and `.drift` columns added.

Examples

```
s <- sim_drift_stream(seed = 42)
detect_drift(s, .col = error, method = "ddm")
```

drift_detector *Specify a drift detector*

Description

Creates an inert detector specification (analogous to a parsnip model spec). Nothing is computed until `fit()` is called on a baseline period.

Usage

```
drift_detector(method = "ddm", ...)
```

Arguments

`method`	Name of a registered detection method, e.g. `"ddm"`.
`...`	Method hyperparameters overriding the defaults (e.g. `min_instances = 50` for `"ddm"`). Unknown parameters error.

Value

A `drift_detector` specification object.

Examples

```
drift_detector("ddm", min_instances = 50)
```

fit.drift_detector *Fit a drift detector on a baseline period*

Description

Runs the detector over the baseline data — the period where the monitored model is considered stable — so it learns the reference ("normal") level. The returned object is immutable: feed new batches with `advance()`.

Usage

```
## S3 method for class 'drift_detector'
fit(object, data, signal, ...)
```

Arguments

`object`	A `drift_detector()` specification.
`data`	A data frame with the baseline period, in temporal order.
`signal`	Unquoted name of the signal column (0/1 errors for error-based methods such as `"ddm"`).
`...`	Not used.

Value

A `drift_detector_fit` object.

Examples

```
base <- sim_drift_stream(n_pre = 100, n_post = 0, seed = 1)
fit(drift_detector("ddm"), base, signal = error)
```

`glance.drift_detector_fit` *One-row summary of a fitted detector*

Description

One-row summary of a fitted detector

Usage

```
## S3 method for class 'drift_detector_fit'
glance(x, ...)
```

Arguments

`x`	A `drift_detector_fit`.
`...`	Not used.

Value

A 1-row tibble: `method`, `n_obs`, `n_warning`, `n_drift`, `first_drift` (NA if no drift detected).

`sim_dist_stream` *Simulate a continuous stream with a known distribution-shift point*

Description

Generates a numeric stream drawn from `N(mean_pre, sd_pre)` for the first `n_pre` observations and `N(mean_post, sd_post)` afterwards. Companion to `sim_drift_stream()` for distribution-based detectors (e.g. `"kswin"`).

Usage

```
sim_dist_stream(
  n_pre = 500,
  n_post = 500,
  mean_pre = 0,
  mean_post = 3,
  sd_pre = 1,
  sd_post = 1,
  seed = NULL
)
```

Arguments

n_pre, n_post	Observations before / after the shift point.
mean_pre, mean_post	Means before / after the shift.
sd_pre, sd_post	Standard deviations before / after the shift.
seed	Optional integer for set.seed().

Value

A tibble with t (index), value (numeric) and drift_true (logical: TRUE after the shift point).

Examples

```
sim_dist_stream(n_pre = 100, n_post = 100, mean_post = 3, seed = 42)
```

---

sim_drift_stream *Simulate a binary error stream with a known drift point*

Description

Generates a stream of 0/1 classifier errors whose error rate jumps from p_pre to p_post after n_pre observations. Useful for testing and validating drift detectors against a known ground truth.

Usage

```
sim_drift_stream(
  n_pre = 500,
  n_post = 500,
  p_pre = 0.05,
  p_post = 0.3,
  seed = NULL
)
```

Arguments

n_pre, n_post	Number of observations before / after the drift point.
p_pre, p_post	Error probability before / after the drift point.
seed	Optional integer; if supplied, set.seed() is called for reproducibility.

Value

A tibble with columns t (index), error (0/1) and drift_true (logical ground truth: TRUE after the drift point).

Examples

```
sim_drift_stream(n_pre = 100, n_post = 100, seed = 42)
```

---

tidy.drift_detector_fit

Drift points of a fitted detector

Description

Drift points of a fitted detector

Usage

```
## S3 method for class 'drift_detector_fit'
tidy(x, ...)
```

Arguments

x	A drift_detector_fit.
...	Not used.

Value

A tibble with one row per detected drift: index (position in the history) and phase.

Index