Package 'leakr'

October 26, 2025

Type Package

Title Data Leakage Detection Tools for Machine Learning

Version 0.1.0

Description Provides utilities to detect common data leakage patterns including train/test contamination, temporal leakage, and data duplication, enhancing model reliability and reproducibility in machine learning workflows. Generates diagnostic reports and visual summaries to support data validation. Methods based on best practices from Hastie, Tibshirani, and Friedman (2009, ISBN:978-0387848570).

Imports ggplot2, arrow, data.table, digest, htmltools, openxlsx, readxl, stringr, workflows, jsonlite

Suggests testthat (>= 3.0.0), caret, mlr3, tidymodels, knitr, rmarkdown

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

VignetteBuilder knitr

NeedsCompilation no

Author Cheryl Isabella Lim [aut, cre]

Maintainer Cheryl Isabella Lim <cheryl.academic@gmail.com>

Repository CRAN

Date/Publication 2025-10-26 18:50:02 UTC

Contents

`compile_report` *Enhanced report compilation with numeric severity scores*

Description

This function compiles a report with enhanced sorting, severity scoring, and detailed metadata, including configuration information.

Usage

```
compile_report(
  results,
  audit_data,
  config,
  show_config = FALSE,
  top_n = 10,
  report = "default"
)
```

Arguments

`results`	A list containing detection results.
`audit_data`	The audit data used for the report.
`config`	Configuration settings, including whether to use numeric severity scores.
`show_config`	Logical, whether to display the configuration used for report generation. Defaults to FALSE.
`top_n`	Numeric, the number of top results to display in the report. Defaults to 10.
`report`	A string indicating the type of report to generate. Defaults to "default".

Value

A `leakr_report` object containing the summary, evidence, and metadata for the report.

`format_detector_name` *Format detector names for display.*

Description

Format detector names by converting them to title case and separating words by spaces.

Usage

```
format_detector_name(detector_name)
```

Arguments

`detector_name`	A string to format, typically a detector name with underscores.

Value

A title-cased, space-separated string.

`grapes-or-or-grapes` *Null-coalescing operator for clean default value handling*

Description

Null-coalescing operator for clean default value handling

Usage

```
x %||% y
```

Arguments

x	First value to check
y	Fallback value if x is NULL

Value

x if not NULL, otherwise y

leakr *leakr: Data Leakage Detection for Machine Learning in R*

Description

leakr: Data Leakage Detection for Machine Learning in R

Details

The leakr package provides tools to automatically detect common data leakage patterns in machine learning workflows for tabular data. It identifies train/test contamination, target leakage, and duplicate rows with clear diagnostic reports and visualisations.

Key Features

- **Train/Test Contamination**: Detects ID overlaps and distributional shifts between training and test sets
- **Target Leakage**: Identifies features with suspicious correlations to the target variable
- **Duplication Detection**: Finds exact and near-duplicate rows
- **Clear Reports**: Generates severity-ranked diagnostics with actionable recommendations
- **Visualisations**: Creates diagnostic plots to highlight issues

Main Functions

- `leakr_audit`: Main function for comprehensive leakage detection
- `leakr_summarise`: Generate human-readable summaries
- `leakr_plot`: Create diagnostic visualisations

Built-in Detectors

- `train_test_contamination`: Checks for overlap between train/test sets
- `target_leakage`: Identifies suspicious feature-target relationships
- `duplication_detection`: Finds duplicate rows in datasets

Data Compatibility

Accepts `data.frame`, `tibble`, and `data.table` objects.

Quick Start

```
# Audit a dataset for leakage
library(leakr)
report <- leakr_audit(my_data, target = "outcome")

# View summary of issues found
```

```
leakr_summarise(report)

# Create diagnostic plots
leakr_plot(report)
```

Author(s)

Maintainer: Cheryl Isabella Lim <cheryl.academic@gmail.com>

See Also

- https://github.com/cherylisabella/leakr
- Report bugs at https://github.com/cherylisabella/leakr/issues

---

leakr_audit *Audit dataset for data leakage*

Description

This function audits a dataset for potential data leakage, running a series of predefined detectors and generating a comprehensive report with detailed findings.

Usage

```
leakr_audit(
  data,
  target = NULL,
  split = NULL,
  id = NULL,
  detectors = NULL,
  config = list()
)
```

Arguments

data	The dataset to be audited (data frame or tibble).
target	The target variable (optional). If NULL, no target variable is assumed.
split	The split variable used for training/test split (optional). If NULL, no split is assumed.
id	The unique identifier for each row (optional). If NULL, no id is used.
detectors	A vector of detector names to run (optional). If NULL, all available detectors will be used.
config	A list of configuration parameters for the audit. Defaults to an empty list.

Value

A leakr_report object containing the audit results, including summary, evidence, and metadata.

Examples

```
# Basic audit on iris dataset
report <- leakr_audit(iris, target = "Species")
print(report)
```

`leakr_create_snapshot` *Create data snapshots with improved metadata handling*

Description

Save data and metadata for reproducible leakage analysis with optimised performance.

Usage

```
leakr_create_snapshot(
  data,
  output_dir = file.path(tempdir(), "leakr_snapshots"),
  snapshot_name = NULL,
  metadata = list(),
  sample_for_hash = TRUE
)
```

Arguments

`data`	Data.frame to snapshot
`output_dir`	Directory for snapshot files
`snapshot_name`	Name for this snapshot
`metadata`	Additional metadata to store
`sample_for_hash`	Whether to sample large datasets for faster hashing

Value

Path to snapshot directory

leakr_export_data *Export data in various formats*

Description

Save processed data to different file formats with consistent behaviour.

Usage

```
leakr_export_data(data, file_path, format = "csv", verbose = TRUE, ...)
```

Arguments

data	Data.frame to export
file_path	Output file path
format	Output format: "csv", "excel", "rds", "json", "parquet"
verbose	Whether to show export messages
...	TODO: Add description

Value

Path to exported file (invisibly)

leakr_from_caret *Convert caret training objects to standard format*

Description

Extract data from caret train objects for leakage analysis.

Usage

```
leakr_from_caret(train_obj, original_data = NULL, target_name = "target")
```

Arguments

train_obj	caret train object
original_data	Original training data (if available)
target_name	Custom name for target variable (default: "target")

Value

List with data and metadata

leakr_from_mlr3 *Convert mlr3 Task objects to standard format*

Description

Extract data from mlr3 Task objects for leakage analysis.

Usage

```
leakr_from_mlr3(task, include_target = TRUE)
```

Arguments

task	mlr3 Task object (TaskClassif, TaskRegr, etc.)
include_target	Whether to include target variable in output

Value

List with data, target, and metadata

leakr_from_tidymodels *Convert tidymodels workflow to standard format*

Description

Extract data from tidymodels workflows for leakage analysis.

Usage

```
leakr_from_tidymodels(workflow, data)
```

Arguments

workflow	tidymodels workflow object
data	Original training data

Value

List with data and metadata

leakr_import *Import data from various sources for leakage analysis*

Description

Flexible data import function supporting multiple formats with automatic format detection and preprocessing for leakage analysis.

Usage

```
leakr_import(
  source,
  format = "auto",
  preprocessing = list(),
  encoding = "UTF-8",
  sheet = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

source	Path to data file, data.frame, or other supported object.
format	Data format: "auto", "csv", "excel", "rds", "json", "parquet", "tsv". If "auto", the format will be detected from the file extension.
preprocessing	List of preprocessing options to apply after import.
encoding	Character encoding for reading files. Default is "UTF-8".
sheet	Sheet name or index to read (for Excel files). Default is NULL.
verbose	Logical indicating whether to print progress messages. Default TRUE.
...	Additional arguments passed to specific import functions.

Value

Standardised data.frame suitable for leakage analysis

A standardized data.frame suitable for leakage analysis.

leakr_list_snapshots *List available snapshots with enhanced information*

Description

Display comprehensive information about available data snapshots.

Usage

```
leakr_list_snapshots(
  snapshots_dir = file.path(tempdir(), "leakr_snapshots"),
  include_metadata = TRUE
)
```

Arguments

snapshots_dir	Directory containing snapshots
include_metadata	Whether to load detailed metadata for each snapshot

Value

Data.frame with snapshot information

leakr_load_snapshot *Load data snapshot with enhanced validation*

Description

Restore data from a previously created snapshot with integrity checking.

Usage

```
leakr_load_snapshot(snapshot_path, format = "rds", verify_integrity = TRUE)
```

Arguments

snapshot_path	Path to snapshot directory
format	Format to load: "rds" (recommended), "csv"
verify_integrity	Whether to verify data integrity using hash

Value

Data.frame from snapshot

leakr_plot *Plot leakage detection results*

Description

Plot leakage detection results

Usage

```
leakr_plot(x, ...)
```

Arguments

x	Results from leakr_audit
...	TODO: Add description Plot leakage detection results

Value

A ggplot object

leakr_quick_import *Fast import with default preprocessing*

Description

Minimal quick import for typical user workflows. Uses leakr_import internally.

Usage

```
leakr_quick_import(source, ...)
```

Arguments

source	File path or data.frame
...	TODO: Add description

Value

Standardised data.frame

`leakr_summarise` *Enhanced summarise with better formatting*

Description

This function provides a formatted summary of the leakage audit report. It displays a summary of the leakage issues, including the severity and top issues detected. Optionally, it can also display configuration details used for the audit.

Usage

```
leakr_summarise(
  report,
  top_n = 10,
  show_config = FALSE,
  config = NULL,
  audit_data = NULL,
  detectors = NULL,
  libname = NULL,
  pkgname = NULL
)
```

Arguments

`report`	A `leakr_report` object from `leakr_audit()`.
`top_n`	Maximum number of issues to display in the summary. Defaults to 10.
`show_config`	Whether to display the configuration details used for the audit. Defaults to `FALSE`.
`config`	(Optional) A configuration list. This argument is not used directly in the function, but is referenced in the report metadata. Defaults to `NULL`.
`audit_data`	(Optional) The data used for auditing. This argument is not used directly in the function, but is part of the report metadata. Defaults to `NULL`.
`detectors`	(Optional) A vector of detectors used for the audit. This argument is not used directly in the function but is part of the report metadata. Defaults to `NULL`.
`libname`	(Optional) The name of the library. This is included for internal package functionality.
`pkgname`	(Optional) The name of the package. This is included for internal package functionality.

Value

An invisible `data.frame` summarizing the top `n` issues detected.

Examples

```
# Create and summarise a report
report <- leakr_audit(iris, target = "Species")
leakr_summarise(report, top_n = 5)
```

`list_registered_detectors` *List Registered Detectors*

Description

Returns the names of all detectors currently registered in the system. This is useful for checking which detectors are available.

Usage

```
list_registered_detectors()
```

Value

A character vector containing the names of all registered detectors.

Examples

```
list_registered_detectors()
```

`new_temporal_detector` *Create a new temporal detector*

Description

Create a new temporal detector

Usage

```
new_temporal_detector(time_col, lookahead_window = 1)
```

Arguments

`time_col`	Character. Name of the time column
`lookahead_window`	Numeric. Lookahead window size (default 1) Create a new temporal detector

Value

A temporal_detector object

A temporal_detector object

`new_train_test_detector` *Create a new train-test detector*

Description

Create a new train-test detector

Usage

```
new_train_test_detector(threshold = 0.1)
```

Arguments

`threshold`	TODO: Document Create a new train-test detector

Value

A train_test_detector object

`plot.detector_result` *Plot a detector_result object*

Description

Plot a detector_result object

Plot a detector_result object

Usage

```
## S3 method for class 'detector_result'
plot(x, palette = NULL, ...)
```

Arguments

`x`	TODO: Document
`palette`	TODO: Document
`...`	TODO: Document

Value

A ggplot object, invisibly. Printed if interactive

A ggplot object, invisibly. Printed if interactive

plot.udld_report *Plot a udld_report object*

Description

This function generates a bar plot of leakage issues detected by different detectors. The plot displays the count of issues by severity level for each detector in a `udld_report` object.

Usage

```
## S3 method for class 'udld_report'
plot(x, palette = NULL, ...)
```

Arguments

`x`	A `udld_report` object. This object contains the detectors and their associated issues.
`palette`	Optional. A `ggplot2` discrete palette for coloring the bars based on severity.
`...`	Additional arguments passed to `ggplot2` functions or other methods. These are typically used for customizing the plot further.

Value

A `ggplot` object, invisibly. The plot is printed if the session is interactive.

register_detector *Register a new detector*

Description

Register a new data leakage detector function

Usage

```
register_detector(name, fun, description = "")
```

Arguments

`name`	Name of the detector
`fun`	TODO: Add description
`description`	TODO: Add description

Value

Invisibly returns registration status

`run_detector` *Run a detector on data*

Description

Run a detector on data

Usage

```
run_detector(detector, data, split = NULL, id = NULL, config = list())
```

Arguments

`detector`	A detector object
`data`	Data frame to analyze
`split`	Split vector indicating train/test assignment (optional)
`id`	Optional ID column name
`config`	Optional configuration list

Value

A detector result object

A detector result object

Index