Package 'CGMissingDataR'

January 20, 2026

Title Missingness Benchmark (MICE Imputation, Random Forest, kNN)

Version 0.0.0.9000

Description Provides an R interface to a small Python benchmark that masks feature values, performs MICE-style imputation, trains Random Forest and kNN regressors, and reports MAPE and R2 across missingness rates.

License GPL (>= 2)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.3)

RoxygenNote 7.3.3

Imports reticulate

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

Config/testthat/edition 3

NeedsCompilation no

Language en-US

URL https://github.com/saraswatsh/CGMissingDataR

BugReports https://github.com/saraswatsh/CGMissingDataR/issues

Contents

run_missingness_benchmark 2

Index **3**

`run_missingness_benchmark` *Run missingness benchmark (Python-backed)*

Description

Loads a CSV, splits train/validation, masks *feature* values at various rates, imputes via an Iterative Imputer (MICE-style), trains Random Forest and kNN regressors, and returns MAPE and R2 per model and mask rate.

Usage

```
run_missingness_benchmark(
  data_path,
  target_col = "LBORRES",
  feature_cols = c("TimeSeries", "TimeDifferenceMinutes", "USUBJID"),
  mask_rates = c(0.05, 0.1, 0.2, 0.3, 0.4),
  test_size = 0.2,
  random_state = 42,
  imputer_random_state = 42,
  rf_n_estimators = 200,
  knn_k = 5
)
```

Arguments

`data_path`	Path to a CSV file.
`target_col`	Name of the target column.
`feature_cols`	Character vector of feature column names.
`mask_rates`	Numeric vector of missingness rates (0-1).
`test_size`	Validation split fraction.
`random_state`	Random seed for train/val splitting and model seeding.
`imputer_random_state`	Random seed for the iterative imputer.
`rf_n_estimators`	Number of trees for the random forest.
`knn_k`	Number of neighbors for kNN.

Details

This function is a thin R wrapper over the Python implementation shipped in `inst/python/CGMissingData`.

Value

A data.frame with columns MaskRate, Model, MAPE, R2.

Index

run_missingness_benchmark, 2