

Package: kfold (via r-universe)

October 11, 2026

Title Machine Learning for Runoff Prediction

Version 0.1.1

Description Machine learning In k-fold cross validation .

License MIT + file LICENSE

Depends R (>= 4.1.0)

Imports hydroGOF, ranger, randomForest, xgboost, data.table, magrittr,
Ipaper, plyr, furrr, future, future.apply, dplyr, tidysselect,
purrr

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Remotes rpkg/Ipaper

Encoding UTF-8

LazyData true

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev git make libharfbuzz-
dev libgit2-dev libicu-dev libpng-dev libuv1-dev libssl-dev libx11-dev zlib1g-dev

Repository <https://rpkg.r-universe.dev>

Date/Publication 2026-10-11 07:20:22 UTC

RemoteUrl <https://github.com/rpkg/kfold>

RemoteRef HEAD

RemoteSha 6f733224f655421c0b30d5ea6b345856f1dd867f

Contents

add_previous	2
chunk_stratified	2
feature_leads	3
GOT_list	3
kfold_calib	4
kfold_ml	4
NSE	6
predict.kfold	7
previous_tn	8

add_previous	<i>add_previous</i>
--------------	---------------------

Description

add_previous

Usage

add_previous(d, nlead = 12)

Arguments

- | | |
|-------|----------------------------|
| d | with the variable of Q_obs |
| nlead | the number of leads to add |

chunk_stratified	<i>Stratified k-fold split</i>
------------------	--------------------------------

Description

Splits observation indices into kfold groups, ensuring each group receives a representative range of the target variable y.

Usage

chunk_stratified(y, kfold = 5, seed = 1)

Arguments

- | | |
|-------|--|
| y | Numeric vector of target values used for stratified splitting. |
| kfold | Number of folds. |
| seed | Random seed (currently unused; seed is fixed internally). |

Value

A named list of length kfold, each element an integer vector of row indices belonging to that fold.

feature_leads	<i>Build lagged feature matrices for multiple lead times</i>
---------------	--

Description

Build lagged feature matrices for multiple lead times

Usage

```
feature_leads(data_full, leads = 1:12)
```

Arguments

data_full	A <code>data.table</code> / <code>data.frame</code> with columns <code>Q_obs</code> , <code>P</code> , <code>PET_Romanenko</code> , and <code>Q_sim</code> .
leads	Integer vector of lead times (in time steps) to build features for.

Value

A named list (one element per lead) of lists with `X` (feature matrix) and `Y` (response matrix).

GOT_list	<i>Compute GOF across multiple lead-time kfold objects</i>
----------	--

Description

Compute GOF across multiple lead-time kfold objects

Usage

```
GOT_list(list_kfold, list_test, ..., idcol = "lead")
```

Arguments

...	Ignored.
idcol	Column name for the lead-time id column.
objects	Named list of kfold objects (one per lead time).
ds_test	Named list of test datasets (one per lead time), each a list with <code>X</code> and <code>Y</code> .

Value

A `data.table` of GOF metrics with columns `lead` and `mode`.

kfold_calib	<i>kfold_calib</i>
-------------	--------------------

Description

Calibrate a model on a single train/validation split.

Usage

```
kfold_calib(X, Y, FUN = xgboost, index = NULL, ..., ratio_valid = 0.3)
```

Arguments

X	Feature matrix (rows = observations).
Y	Response matrix (rows = observations).
FUN	Model fitting function with signature FUN(x_train, y_train, ...).
index	Integer vector of validation row indices. If NULL, the first floor($n * \text{ratio_valid}$) rows are used.
...	Additional arguments forwarded to FUN.
ratio_valid	Fraction of rows used as validation when index = NULL.

kfold_ml	<i>kfold machine learning</i>
----------	-------------------------------

Description

kfold machine learning

Usage

```
kfold_ml(
  X,
  Y,
  kfold = 5,
  FUN,
  ...,
  fn_chunk = chunk_stratified,
  ind_lst = NULL,
  .progress = TRUE
)

kfold_rf(X, Y, kfold = 5, FUN = ranger, ntree = 500, importance = "none", ...)

kfold_xgboost(X, Y, kfold = 5, FUN = xgboost, nrounds = 500, ...)

kfold_lm(X, Y, kfold = 5, ...)
```

Arguments

<code>X</code>	Feature matrix (rows = observations).
<code>Y</code>	Response matrix (rows = observations).
<code>kfold</code>	Number of folds.
<code>FUN</code>	Model fitting function with signature <code>FUN(x_train, y_train, ...)</code> .
<code>...</code>	Additional arguments forwarded to <code>FUN</code> .
<code>fn_chunk</code>	Fold-splitting function; defaults to <code>chunk_stratified()</code> .
<code>ind_lst</code>	Optional list of validation row indices, one element per fold. If given, <code>kfold</code> and <code>fn_chunk</code> are ignored.
<code>.progress</code>	Show a progress bar during fold iteration.
<code>ntree</code>	Number of trees for <code>kfold_rf()</code> .
<code>importance</code>	Variable importance mode, one of 'none', 'impurity', 'impurity_corrected', 'permutation'. The 'impurity' measure is the Gini index for classification, the variance of the responses for regression and the sum of test statistics (see <code>splitrule</code>) for survival.
<code>nrounds</code>	Number of boosting iterations / rounds. Note that the number of default boosting rounds here is not automatically tuned, and different problems will have vastly different optimal numbers of boosting rounds.

See Also

`ranger::ranger()`, `xgboost::xgboost()`

Examples

```
set.seed(1)
n <- 100 ; p <- 2
X <- matrix(rnorm(n * p), n, p) # no intercept!
y <- as.matrix(rnorm(n))

## kfold
r_lm <- kfold_lm(X, y)
r_xgb <- kfold_xgboost(X, y)
# r_rf <- kfold_rf(X, y)

## 70%-30% split
r = kfold_calib(X, y, ratio_valid = 0.7, nrounds=500, verbose=FALSE)
r$gof
```

NSE

GOF

Description

Good of fitting

Usage

```
NSE(yobs, ysim, w, ...)
```

```
GOF(yobs, ...)
```

```
## Default S3 method:
```

```
GOF(
  yobs,
  ysim,
  w = NULL,
  include.cv = FALSE,
  include.r = TRUE,
  ...,
  idcol = "kfold",
  mode = "test"
)
```

```
## S3 method for class 'kfold'
```

```
GOF(yobs, test = NULL, ...)
```

Arguments

<code>yobs</code>	Numeric vector, observations
<code>ysim</code>	Numeric vector, corresponding simulated values
<code>w</code>	Numeric vector, weights of every points. If <code>w</code> included, when calculating mean, Bias, MAE, RMSE and NSE, <code>w</code> will be taken into considered.
<code>...</code>	Ignored.
<code>include.cv</code>	If true, <code>cv</code> will be included.
<code>include.r</code>	If true, <code>r</code> and <code>R2</code> will be included.
<code>idcol</code>	Column name for the id column when binding multi-column results.
<code>mode</code>	Label inserted into the mode column when computing multi-column GOF.
<code>test</code>	A list with <code>X</code> and <code>Y</code> for external testing. If <code>NULL</code> , GOF will be calculated

Value

- RMSE root mean square error
- NSE NASH coefficient
- MAE mean absolute error
- AI Agreement index (only good points ($w == 1$)) participate to calculate. See details in Zhang et al., (2015).
- Bias bias
- Bias_perc bias percentage
- n_sim number of valid obs
- cv Coefficient of variation
- R2 correlation of determination
- R pearson correlation
- pvalue pvalue of R

References

1. https://en.wikipedia.org/wiki/Coefficient_of_determination
2. https://en.wikipedia.org/wiki/Explained_sum_of_squares
3. https://en.wikipedia.org/wiki/Nash%E2%80%93Sutcliffe_model_efficiency_coefficient
4. Zhang Xiaoyang (2015), <http://dx.doi.org/10.1016/j.rse.2014.10.012>

Examples

```
yobs <- rnorm(100)
ysim <- yobs + rnorm(100) / 4
GOF(yobs, ysim)
```

predict.kfold	<i>predict for kfold object</i>
---------------	---------------------------------

Description

predict for kfold object

Usage

```
## S3 method for class 'kfold'
predict(object, newdata = NULL, ..., mode = "test")
```

Arguments

object	A kfold object returned by <code>kfold_ml()</code> .
newdata	New feature matrix for prediction. Required when mode = "test".
...	Additional arguments forwarded to the underlying model's predict method.
mode	Prediction mode: "train" (in-sample, hold-out fold masked), "valid" (out-of-fold only), or "test" (full new data).

previous_tn	<i>previous_tn</i>
-------------	--------------------

Description

Build a lagged matrix: column `t` is the original series, columns `t-1`, `t-2`, ... are progressively shifted (lagged) copies.

Usage

```
previous_tn(x, n = 7, prefix = "", ...)  
  
## Default S3 method:  
previous_tn(x, n = 7, prefix = "", ...)  
  
## S3 method for class 'data.frame'  
previous_tn(x, n = 7, ...)
```

Arguments

x	Numeric vector (default method) or <code>data.frame</code> to lag.
n	Number of lags to create.
prefix	Character prefix prepended to each column name.
...	Ignored.

Examples

```
set.seed(1)  
x <- rnorm(10)  
previous_tn(x, 7, "R1_")  
# data.frame  
d = data.frame(x)  
previous_tn(d)
```

Index

`add_previous`, [2](#)

`chunk_stratified`, [2](#)
`chunk_stratified()`, [5](#)

`feature_leads`, [3](#)

`GOF (NSE)`, [6](#)
`GOT_list`, [3](#)

`kfold_calib`, [4](#)
`kfold_lm(kfold_ml)`, [4](#)
`kfold_ml`, [4](#)
`kfold_ml()`, [8](#)
`kfold_rf(kfold_ml)`, [4](#)
`kfold_rf()`, [5](#)
`kfold_xgboost(kfold_ml)`, [4](#)

`NSE`, [6](#)

`predict.kfold`, [7](#)
`previous_tn`, [8](#)

`ranger::ranger()`, [5](#)

`xgboost::xgboost()`, [5](#)