

Package: mlfeaturer (via r-universe)

October 6, 2026

Type Package

Title Machine Learning Feature Engineering with R

Version 0.4.3

Date 2026-08-07

Description Utilities for scaling and transformation of features in the context of machine learning regression problems.

URL <https://tpetzoldt.github.io/mlfeaturer/>

License GPL (>= 2)

Depends R (>= 4.1.0)

Imports dplyr, stats, graphics, utils, methods, tidyr, purrr,
lubridate, knitr, jsonlite

Suggests keras3, nnet, quarto, rmarkdown, ggplot2, qualV

Encoding UTF-8

LazyData true

VignetteBuilder quarto

Config/roxygen2/version 8.0.0

Collate 'mlfeaturer-package.R' 'aaa_classes.R' 'as.list.R'
'create_preprocessed_data.R' 'daut4.R' 'get.R' 'get_id.R'
'inv_transform_data.R' 'is_keras_model.R' 'ml_evaluate.R'
'predict.R' 'residuals.R' 'rsquared.R' 'scale_data.R'
'select_and_transform.R' 'setAs.R' 'set_transformations.R'
'transform_data.R' 'utilities.R'

Config/pak/sysreqs libicu-dev

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

Date/Publication 2026-08-07 07:24:34 UTC

RemoteUrl <https://github.com/tpetzoldt/mlfeaturer>

RemoteRef HEAD

RemoteSha 0d715879bdbca95fb9bf6e03ef091a0a8e29dae4

Contents

mlfeaturer-package	2
as	3
as.list,feature_data-method	3
create_preprocessed_data	4
daut4	5
feature_data-class	6
feature_data_list-class	6
feature_params-class	7
get_data	7
get_id_all	8
get_id_test	9
get_id_train	10
get_x_all	10
get_x_test	11
get_x_train	12
get_y_all	13
get_y_test	13
get_y_train	14
inv_transform_data	15
list_or_NULL-class	16
ml_evaluate	16
predict,feature_data-method	17
residuals,feature_data-method	18
rsquared	19
scale_data	20
set_transformations	21
transform_data	22
Index	23

mlfeaturer-package	<i>Machine Learning and Feature Engineering for Regression Models</i>
--------------------	---

Description

This package provides preprocessing tools using S4 classes for data transformation, scaling, and inverse transformation.

Author(s)

Maintainer: Thomas Petzoldt <thomas.petzoldt@tu-dresden.de> ([ORCID](#))

Authors:

- Thomas Petzoldt <thomas.petzoldt@tu-dresden.de> ([ORCID](#))

See Also

Useful links:

- <https://tpetzoldt.github.io/mlfeaturer/>

as	<i>Convert feature_data to feature_data_list</i>
----	--

Description

Coerces a 'feature_data' object to a 'feature_data_list' object.

Coerces a 'feature_data' object to a standard R 'list' object.

Arguments

from A 'feature_data' object.

Value

A 'feature_data_list' object.

A 'list' object.

as.list, feature_data-method
<i>Convert feature_data to a list</i>

Description

Coerces a 'feature_data' object to a 'list' object.

Usage

```
## S4 method for signature 'feature_data'
as.list(x)
```

Arguments

x A 'feature_data' object.

Value

A 'list' object.

```
create_preprocessed_data
```

Create Preprocessed Data Object

Description

Constructs a ‘feature_data’ object, calculates scaling parameters, and optionally applies initial data transformations.

Usage

```
create_preprocessed_data(
  data,
  target_col,
  id_col = NULL,
  split_col = NULL,
  scale_option = c("train", "test", "all"),
  scale_method = c("zscore", "minmax", "none"),
  fun_transform = NULL,
  fun_inverse = NULL
)
```

Arguments

<code>data</code>	A data frame containing the raw data.
<code>target_col</code>	Character vector specifying the name of the target variable column.
<code>id_col</code>	Character vector specifying the name of the ID column (optional).
<code>split_col</code>	Character vector specifying the name of the column used for train/test split.
<code>scale_option</code>	Character string specifying the scaling option. Must be one of "train", "test", or "all". Defaults to "train".
<code>scale_method</code>	Character string specifying the scaling method. Must be one of "zscore" (standardization) or "minmax" (normalization). Defaults to "zscore".
<code>fun_transform</code>	A named list of functions to apply to the data <i>before</i> scaling. The names of the list elements should correspond to the columns to transform.
<code>fun_inverse</code>	A named list of functions representing the inverse transformations of ‘fun_transform’. These are applied during the ‘inverse_transform’ method.

Details

This function calculates scaling parameters (mean, standard deviation, min, max) based on the specified ‘scale_option’ and ‘scale_method’. It then creates a ‘feature_data’ object, storing the original data and the calculated parameters. If ‘fun_transform’ is provided, it applies the specified transformations to the data before creating the object.

If no ‘id_col’ is given, an additional column ‘id’ with row numbers is created.

If a ‘split_col’ is missing, an random ‘split’ column is created.

Value

A ‘feature_data’ object.

Examples

```
# Example usage (replace with your actual data and column names)
df <- data.frame(
  id = 1:10,
  x = runif(10),
  y = rnorm(10),
  z = 1:10,
  split = sample(c(TRUE, FALSE), 10, replace = TRUE)
)

transformations <- list(
  x = log,
  y = \(y) sqrt(5 + y),
  z = \(z) z^2
)

inverse_transformations <- list(
  x = exp,
  y = \(y) y^2 - 5,
  z = sqrt
)

prep_data <- create_preprocessed_data(
  df, id_col = "id", target_col = "y", split_col = "split",
  fun_transform = transformations, fun_inverse = inverse_transformations)
print(prep_data)
```

dauta4

*Growthrate of Algae***Description**

Growthrates of four different phytoplankton species, dependent on temperature and light.

Format

Data frame with the following columns:

species Phytoplankton species.

temperature Temperature in °C.

light Photosynthetic active radiation in

$$\mu E m^{-2} s^{-1}$$

growthrate Population growth rate in d^{-1}

Source

The example data were taken manually from the figures of Dauta et al. 1990.

References

Dauta, A, Devaux, J, Piquemal, F and Boumnich, L (1990) Growth Rate of Four Algae in Relation to Light and Temperature. Hydrobiologia 207: 221–26. doi:10.1007/BF00041459

feature_data-class	<i>Preprocessed Data</i>
--------------------	--------------------------

Description

An S4 class to store the preprocessed data and its associated parameters.

Slots

data Data frame containing the preprocessed data.
params ‘feature_params’ object storing the parameters used in preprocessing.

feature_data_list-class	<i>List Representation of Preprocessed Data</i>
-------------------------	---

Description

An S4 class to represent the preprocessed data as a list.

Slots

x_train Training data (X).
x_test Test data (X).
x_all All data (X).
y_train Training data (Y).
y_test Test data (Y).
y_all All data (Y).
x_train_scaled Scaled training data (X).
x_test_scaled Scaled test data (X).
x_all_scaled Scaled all data (X).

feature_params-class *Preprocessing Parameters*

Description

An S4 class to store the parameters used in the preprocessing pipeline.

Slots

id_col Character vector specifying the name of the ID column.

target_col Character vector specifying the name of the target variable column.

split_col Character vector specifying the name of the column used for train/test split.

scale_option Character string specifying the scaling option ("train", "test", "all").

scale_method Character string specifying the scaling method ("scale", "norm").

mean_vals Numeric vector storing the mean values for z-score scaling.

sd_vals Numeric vector storing the standard deviation values for z-score scaling.

min_vals Numeric vector storing the minimum values for minmax normalization.

max_vals Numeric vector storing the maximum values for minmax normalization.

t_mean_vals Numeric vector storing the mean of transformed values for z-score scaling.

t_sd_vals Numeric vector storing the standard deviation of transformed values for z-score scaling.

t_min_vals Numeric vector storing the minimum values of transformed for minmax normalization.

t_max_vals Numeric vector storing the maximum values of transformed for minmax normalization.

fun_transform List of functions for data transformation.

fun_inverse List of functions for inverse data transformation.

transformed Boolean value, 'TRUE' if object contains transformed values, 'NA' if it is the result of an inverse transformation.

get_data *Get All Data (X and Y)*

Description

Access complete data set from the preprocessed data.

Usage

```
get_data(object, ...)

## S4 method for signature 'feature_data'
get_data(
  object,
  prep = c("both", "scale", "transform", "none"),
  as_matrix = TRUE
)
```

Arguments

object	A 'feature_data' object.
...	Additional arguments (currently not used).
prep	Character argument if transformed data ("transform"), scaled data ("scale"), transformed and scaled data ("both") or original raw data ("none") will be returned.
as_matrix	logical TRUE if the function should return matrix, or a data frame or tibble otherwise.

Value

A data frame or matrix containing all data, including 'id_col' and 'split_col'.

Methods (by class)

- get_data(feature_data): Method for extracting all data.

get_id_all	<i>Get ID Columns for all data</i>
------------	------------------------------------

Description

Extracts ID columns (ID) from the preprocessed data.

Usage

```
get_id_all(object, ...)

## S4 method for signature 'feature_data'
get_id_all(object, as_matrix = TRUE)
```

Arguments

object	A 'feature_data' object.
...	Additional arguments (currently not used).
as_matrix	logical TRUE if the function should return matrix, or a data frame or tibble otherwise.

Value

A data frame or matrix containing the ID column(s).

Methods (by class)

- `get_id_all(feature_data)`: Method extracting id columns for all data

<code>get_id_test</code>	<i>Get ID Columns for test data</i>
--------------------------	-------------------------------------

Description

Extracts ID columns (ID) from the preprocessed data.

Usage

```
get_id_test(object, ...)

## S4 method for signature 'feature_data'
get_id_test(object, as_matrix = TRUE)
```

Arguments

<code>object</code>	A 'feature_data' object.
<code>...</code>	Additional arguments (currently not used).
<code>as_matrix</code>	logical TRUE if the function should return matrix, or a data frame or tibble otherwise.

Value

A data frame or matrix containing the ID column(s).

Methods (by class)

- `get_id_test(feature_data)`: Method for extracting id columns for the test data

get_id_train	<i>Get ID Columns for training data</i>
--------------	---

Description

Extracts ID columns (ID) from the preprocessed data.

Usage

```
get_id_train(object, ...)

## S4 method for signature 'feature_data'
get_id_train(object, as_matrix = TRUE)
```

Arguments

object	A 'feature_data' object.
...	Additional arguments (currently not used).
as_matrix	logical TRUE if the function should return matrix, or a data frame or tibble otherwise.

Value

A data frame or matrix containing the ID column(s).

Methods (by class)

- get_id_train(feature_data): Method for extracting id columns for the training data

get_x_all	<i>Get All Data (X)</i>
-----------	-------------------------

Description

Extracts all the data (X) from the preprocessed data.

Usage

```
get_x_all(object, ...)

## S4 method for signature 'feature_data'
get_x_all(
  object,
  prep = c("both", "scale", "transform", "none"),
  as_matrix = TRUE
)
```

Arguments

object	A 'feature_data' object.
...	Additional arguments (currently not used).
prep	Character argument if transformed data ("transform"), scaled data ("scale"), transformed and scaled data ("both") or original raw data ("none") will be returned.
as_matrix	logical TRUE if the function should return matrix, or a data frame or tibble otherwise.

Value

A data frame or matrix containing all the data (X).

Methods (by class)

- `get_x_all(feature_data)`: Method for extracting all data (X).

get_x_test	<i>Get Test Data (X)</i>
------------	--------------------------

Description

Extracts the test data (X) from the preprocessed data.

Usage

```
get_x_test(object, ...)

## S4 method for signature 'feature_data'
get_x_test(
  object,
  prep = c("both", "scale", "transform", "none"),
  as_matrix = TRUE
)
```

Arguments

object	A 'feature_data' object.
...	Additional arguments (currently not used).
prep	Character argument if transformed data ("transform"), scaled data ("scale"), transformed and scaled data ("both") or original raw data ("none") will be returned.
as_matrix	logical TRUE if the function should return matrix, or a data frame or tibble otherwise.

Value

A data frame or matrix containing the test data (X).

Methods (by class)

- `get_x_test(feature_data)`: Method for extracting test data (X).

<code>get_x_train</code>	<i>Get Training Data (X)</i>
--------------------------	------------------------------

Description

Extracts the training data (X) from the preprocessed data.

Usage

```
get_x_train(object, ...)

## S4 method for signature 'feature_data'
get_x_train(
  object,
  prep = c("both", "scale", "transform", "none"),
  as_matrix = TRUE
)
```

Arguments

<code>object</code>	A 'feature_data' object.
<code>...</code>	Additional arguments (currently not used).
<code>prep</code>	Character argument if transformed data ("transform"), scaled data ("scale"), transformed and scaled data ("both") or original raw data ("none") will be returned.
<code>as_matrix</code>	logical TRUE if the function should return matrix, or a data frame or tibble otherwise.

Value

A data frame or matrix containing the training data (X).

Methods (by class)

- `get_x_train(feature_data)`: Method for extracting training data (X).

get_y_all	<i>Get All Data (Y)</i>
-----------	-------------------------

Description

Extracts all the data (Y) from the preprocessed data.

Usage

```
get_y_all(object, ...)  
  
## S4 method for signature 'feature_data'  
get_y_all(  
  object,  
  prep = c("both", "scale", "transform", "none"),  
  as_matrix = TRUE  
)
```

Arguments

object	A 'feature_data' object.
...	Additional arguments (currently not used).
prep	Character argument if transformed data ("transform"), scaled data ("scale"), transformed and scaled data ("both") or original raw data ("none") will be returned.
as_matrix	logical TRUE if the function should return matrix, or a data frame or tibble otherwise.

Value

A data frame or matrix containing all the data (Y).

Methods (by class)

- get_y_all(feature_data): Method for extracting all data (Y).

get_y_test	<i>Get Test Data (Y)</i>
------------	--------------------------

Description

Extracts the test data (Y) from the preprocessed data.

Usage

```
get_y_test(object, ...)

## S4 method for signature 'feature_data'
get_y_test(
  object,
  prep = c("both", "scale", "transform", "none"),
  as_matrix = TRUE
)
```

Arguments

object	A 'feature_data' object.
...	Additional arguments (currently not used).
prep	Character argument if transformed data ("transform"), scaled data ("scale"), transformed and scaled data ("both") or original raw data ("none") will be returned.
as_matrix	logical TRUE if the function should return matrix, or a data frame or tibble otherwise.

Value

A data frame or matrix containing the test data (Y).

Methods (by class)

- `get_y_test(feature_data)`: Method for extracting test data (Y).

get_y_train	<i>Get Training Data (Y)</i>
-------------	------------------------------

Description

Extracts the training data (Y) from the preprocessed data.

Usage

```
get_y_train(object, ...)

## S4 method for signature 'feature_data'
get_y_train(
  object,
  prep = c("both", "scale", "transform", "none"),
  as_matrix = TRUE
)
```

Arguments

object	A 'feature_data' object.
...	Additional arguments (currently not used).
prep	Character argument if transformed data ("transform"), scaled data ("scale"), transformed and scaled data ("both") or original raw data ("none") will be returned.
as_matrix	logical TRUE if the function should return matrix, or a data frame or tibble otherwise.

Value

A data frame or matrix containing the training data (Y).

Methods (by class)

- `get_y_train(feature_data)`: Method for extracting training data (Y).

inv_transform_data	<i>Inverse Transform Data</i>
--------------------	-------------------------------

Description

Applies the inverse transformation to the data.

Usage

```
inv_transform_data(object, params, funs, ...)

## S4 method for signature 'feature_data,ANY,ANY'
inv_transform_data(object, params, funs, ...)

## S4 method for signature 'feature_data,missing,list'
inv_transform_data(object, params, funs, ...)

## S4 method for signature 'data.frame,feature_params,missing'
inv_transform_data(object, params, funs, ...)
```

Arguments

object	A 'feature_data' object.
params	A 'feature_params' object with transformation and scaling parameters
funs	A list of inverse data transformation functions.
...	Additional arguments (currently not used).

Value

A 'feature_data' object or a 'data.frame' with the inverse transformed data.

Methods (by class)

- `inv_transform_data(object = feature_data, params = ANY, funs = ANY)`: Method for inverse transforming data in a 'feature_data' object.
- `inv_transform_data(object = feature_data, params = missing, funs = list)`: Method for transforming data in a 'feature_data' object and user-defined functions (deprecated).
- `inv_transform_data(object = data.frame, params = feature_params, funs = missing)`: Method for a 'data frame'.

<code>list_or_NULL-class</code>	<i>Union Class of List or NULL</i>
---------------------------------	------------------------------------

Description

Class to slots that can be empty

<code>ml_evaluate</code>	<i>Evaluate Machine Learning Model Performance</i>
--------------------------	--

Description

Calculates goodness-of-fit metrics (MSE, RMSE, MAE, R2) on preprocessed data, supporting fine-grained control over feature and target preprocessing states.

Usage

```
ml_evaluate(object, ...)

## S4 method for signature 'feature_data'
ml_evaluate(
  object,
  model,
  subset = c("all", "test", "train"),
  xprep = c("both", "scale", "transform", "none"),
  yprep = c("both", "scale", "transform", "none"),
  to_original_scale = FALSE,
  ...
)
```

Arguments

object	A 'feature_data' object.
...	Additional arguments passed to 'predict()'.
model	A fitted machine learning model (e.g., Keras, nnet, lm).
subset	Data subset to evaluate ("all", "train", or "test").
xprep	Preprocessing state for predictor features ("both", "scale", "transform", "none").
yprep	Preprocessing state for target variable ("both", "scale", "transform", "none").
to_original_scale	Logical; if 'TRUE', evaluates performance on original physical units (overrides 'yprep' to "none" for target comparison).

Details

Bias (Mean Error) is calculated as $\text{mean}(\text{observed} - \text{predicted})$. Positive values indicate systematic underprediction by the model.

Value

A named numeric vector of evaluation metrics (MSE, RMSE, MAE, R2).

Methods (by class)

- `ml_evaluate(feature_data)`: Evaluate performance metrics for a 'feature_data' object.

predict,feature_data-method

Model Predictions with 'feature_data' Objects

Description

Makes predictions from a machine learning model with a 'feature_data' object.

Usage

```
## S4 method for signature 'feature_data'
predict(
  object,
  model,
  subset = c("all", "test", "train"),
  xprep = c("both", "scale", "transform", "none"),
  yprep = c("both", "scale", "transform", "none"),
  to_original_scale = FALSE,
  as_matrix = TRUE,
  ...
)
```

Arguments

object	A 'feature_data' object.
model	A fitted model object.
subset	Subset of the 'preproc' data set.
xprep	Character argument if transformed data ("transform"), scaled data ("scale"), transformed and scaled data ("both") or original raw data ("none") are used for x.
yprep	Character argument if transformed data ("transform"), scaled data ("scale"), transformed and scaled data ("both") or original raw data ("none") are used for y.
to_original_scale	Logical value to force re-transformation of predictions to the original scale.
as_matrix	logical TRUE if the function should return matrix, or a data frame or tibble otherwise.
...	Additional arguments (currently not used).

Details

Currently only certain Keras models can be used as fitted models.

Value

A matrix or vector with the predictions.

residuals,feature_data-method

Extract Model Residuals Using 'feature_data' Objects

Description

Extract residuals from a machine learning model compared to a 'feature_data' object.

Usage

```
## S4 method for signature 'feature_data'
residuals(
  object,
  model,
  subset = c("all", "test", "train"),
  xprep = c("both", "scale", "transform", "none"),
  yprep = c("both", "scale", "transform", "none"),
  to_original_scale = FALSE,
  as_matrix = TRUE,
  ...
)
```

Arguments

object	A 'feature_data' object.
model	A fitted model object.
subset	Subset of the 'preproc' data set.
xprep	Character argument if transformed data ("transform"), scaled data ("scale"), transformed and scaled data ("both") or original raw data ("none") are used for x.
yprep	Character argument if transformed data ("transform"), scaled data ("scale"), transformed and scaled data ("both") or original raw data ("none") are used for y.
to_original_scale	Logical value to force re-transformation of predictions to the original scale.
as_matrix	logical TRUE if the function should return matrix, or a data frame or tibble otherwise.
...	Additional arguments (currently not used).

Details

Currently only certain Keras models can be used as fitted models.

Value

A matrix or vector with the residuals.

rsquared	<i>Nonlinear Coefficient of Determination Using 'feature_data' Objects</i>
----------	--

Description

Estimates the nonlinear R2 from a machine learning model compared to a 'feature_data' object.

Usage

```
rsquared(object, ...)

## S4 method for signature 'feature_data'
rsquared(
  object,
  model,
  subset = c("all", "test", "train"),
  prep = c("both", "scale", "transform", "none"),
  ...
)
```

Arguments

object	A 'feature_data' object.
...	Additional arguments passed down to 'predict()'.
model	A fitted model object.
subset	Subset of the 'preproc' dataset ("all", "train", or "test").
prep	Character argument if transformed data ("transform"), scaled data ("scale"), transformed and scaled data ("both"), or original raw data ("none") will be returned.

Value

The coefficient of determination as an un-named scalar numeric value.

Methods (by class)

- `rsquared(feature_data)`: Method for estimating the coefficient of determination of a model for a 'feature_data' object.

scale_data

Scale Data

Description

Scales the data using the stored parameters.

Usage

```
scale_data(object, params, ...)
```

```
## S4 method for signature 'feature_data,missing'
scale_data(object, params, ...)
```

```
## S4 method for signature 'data.frame,feature_params'
scale_data(object, params, transformed = FALSE, ...)
```

Arguments

object	A 'feature_data' object or a data frame.
params	n 'feature_params' object with scaling parameters
...	Additional arguments (currently not used).
transformed	'TRUE' to use parameters of transformed data.

Value

A 'feature_data' object with the scaled data.

Methods (by class)

- `scale_data(object = feature_data, params = missing)`: Method for scaling data in a 'feature_data' object.
- `scale_data(object = data.frame, params = feature_params)`: Method for scaling data in a 'feature_data' object.

set_transformations	<i>Set Transformations</i>
---------------------	----------------------------

Description

Sets the transformation functions for a 'feature_data' object.

Usage

```
set_transformations(object, fun_transform = NULL, fun_inverse = NULL)

## S4 method for signature 'feature_data'
set_transformations(object, fun_transform = NULL, fun_inverse = NULL)
```

Arguments

<code>object</code>	A 'feature_data' object.
<code>fun_transform</code>	A named list of functions to apply to the data. The names should correspond to the columns, and the values should be the functions.
<code>fun_inverse</code>	A named list of functions representing the inverse transformations. The structure should be the same as 'fun_transform'.

Value

A 'feature_data' object with the updated transformation functions.

Methods (by class)

- `set_transformations(feature_data)`: Method for setting transformations in a 'feature_data' object.

transform_data

*Transform Data***Description**

Transforms the data using the provided functions.

Usage

```
transform_data(object, params, funs, ...)

## S4 method for signature 'feature_data,ANY,ANY'
transform_data(object, params, funs, ...)

## S4 method for signature 'feature_data,missing,list'
transform_data(object, params, funs, ...)

## S4 method for signature 'data.frame,feature_params,missing'
transform_data(object, params, funs, ...)

## S4 method for signature 'data.frame,missing,list'
transform_data(object, params, funs, ...)
```

Arguments

object	A 'feature_data' object or a data frame.
params	A 'feature_params' object with transformation and scaling parameters
funs	A list of data transformation functions .
...	Additional arguments (currently not used).

Value

A 'feature_data' object or a 'data.frame' with the transformed data.

Methods (by class)

- transform_data(object = feature_data, params = ANY, funs = ANY): Method for transforming data in a 'feature_data' object.
- transform_data(object = feature_data, params = missing, funs = list): Method for transforming data in a 'feature_data' object and user-defined functions (deprecated).
- transform_data(object = data.frame, params = feature_params, funs = missing): Method for a 'data frame'.
- transform_data(object = data.frame, params = missing, funs = list): Method for a 'data frame'.

Index

- * **data**
 - daut4, [5](#)
- as, [3](#)
- as.list, feature_data-method, [3](#)
- create_preprocessed_data, [4](#)
- daut4, [5](#)
- feature_data-class, [6](#)
- feature_data_list-class, [6](#)
- feature_params-class, [7](#)
- get_data, [7](#)
- get_data, feature_data-method
 - (get_data), [7](#)
- get_id_all, [8](#)
- get_id_all, feature_data-method
 - (get_id_all), [8](#)
- get_id_test, [9](#)
- get_id_test, feature_data-method
 - (get_id_test), [9](#)
- get_id_train, [10](#)
- get_id_train, feature_data-method
 - (get_id_train), [10](#)
- get_x_all, [10](#)
- get_x_all, feature_data-method
 - (get_x_all), [10](#)
- get_x_test, [11](#)
- get_x_test, feature_data-method
 - (get_x_test), [11](#)
- get_x_train, [12](#)
- get_x_train, feature_data-method
 - (get_x_train), [12](#)
- get_y_all, [13](#)
- get_y_all, feature_data-method
 - (get_y_all), [13](#)
- get_y_test, [13](#)
- get_y_test, feature_data-method
 - (get_y_test), [13](#)
- get_y_train, [14](#)
- get_y_train, feature_data-method
 - (get_y_train), [14](#)
- inv_transform_data, [15](#)
- inv_transform_data, data.frame, feature_params, missing-method
 - (inv_transform_data), [15](#)
- inv_transform_data, feature_data, ANY, ANY-method
 - (inv_transform_data), [15](#)
- inv_transform_data, feature_data, missing, list-method
 - (inv_transform_data), [15](#)
- list_or_NULL-class, [16](#)
- ml_evaluate, [16](#)
- ml_evaluate, feature_data-method
 - (ml_evaluate), [16](#)
- mlfeaturer (mlfeaturer-package), [2](#)
- mlfeaturer-package, [2](#)
- predict, feature_data-method, [17](#)
- residuals, feature_data-method, [18](#)
- rsquared, [19](#)
- rsquared, feature_data-method
 - (rsquared), [19](#)
- scale_data, [20](#)
- scale_data, data.frame, feature_params-method
 - (scale_data), [20](#)
- scale_data, feature_data, missing-method
 - (scale_data), [20](#)
- set_transformations, [21](#)
- set_transformations, feature_data-method
 - (set_transformations), [21](#)
- transform_data, [22](#)
- transform_data, data.frame, feature_params, missing-method
 - (transform_data), [22](#)
- transform_data, data.frame, missing, list-method
 - (transform_data), [22](#)

transform_data, feature_data, ANY, ANY-method
 (transform_data), [22](#)
transform_data, feature_data, missing, list-method
 (transform_data), [22](#)