

Package ‘live’

July 22, 2025

Type Package

Title Local Interpretable (Model-Agnostic) Visual Explanations

Version 1.5.13

Description Interpretability of complex machine learning models is a growing concern. This package helps to understand key factors that drive the decision made by complicated predictive model (so called black box model). This is achieved through local approximations that are either based on additive regression like model or CART like model that allows for higher interactions. The methodology is based on Tulio Ribeiro, Singh, Guestrin (2016) <doi:10.1145/2939672.2939778>. More details can be found in Staniak, Biecek (2018) <doi:10.32614/RJ-2018-072>.

URL <https://github.com/ModelOriented/live>

BugReports <https://github.com/ModelOriented/live/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Depends R (>= 3.0.2)

Suggests knitr, rmarkdown, testthat, glmnet, covr, DALEX, RWeka, mda, modeltools

VignetteBuilder knitr

Imports mlr, dplyr, breakDown, data.table, forestmodel, shiny, MASS, ggplot2, gower, e1071

NeedsCompilation no

Author Mateusz Staniak [cre, aut],
Przemysław Biecek [aut]

Maintainer Mateusz Staniak <mateusz.staniak@math.uni.wroc.pl>

Repository CRAN

Date/Publication 2020-01-15 06:30:17 UTC

Contents

add_predictions	2
euclidean_kernel	3
fit_explanation	4
gaussian_kernel	5
identity_kernel	6
live	6
live_shiny	7
local_approximation	7
local_permutation_importance	8
plot.live_explainer	10
plot.local_permutation_importance	11
print.live_explainer	11
print.live_explorer	12
print.local_permutation_importance	12
sample_locally	13
wine	14
Index	15

add_predictions	<i>Add black box predictions to generated dataset</i>
-----------------	---

Description

Add black box predictions to generated dataset

Usage

```
add_predictions(  
  to_explain,  
  black_box_model,  
  data = NULL,  
  predict_fun = predict,  
  hyperparams = list(),  
  ...  
)
```

Arguments

- to_explain List return by sample_locally function.
- black_box_model String with mlr signature of a learner or a model with predict interface.
- data Original data frame used to generate new dataset. Need not be provided when a trained model is passed in black_box_model argument.

predict_fun	Either a "predict" function that returns a vector of the same type as response or custom function that takes a model as a first argument, and data used to calculate predictions as a second argument and returns a vector of the same type as response. Will be used only if a model object was provided in the black_box argument.
hyperparams	Optional list of (hyper)parameters to be passed to mlr::makeLearner.
...	Additional parameters to be passed to predict function.

Value

list of class "live_explorer" consisting of

data	Dataset generated by sample_locally function with response variable.
target	Name of the response variable.
model	Black box model which is being explained.
explained_instance	Instance that is being explained.
sampling_method	Name of used sampling method
fixed_variables	Names of variables which were not sampled
sdeviations	Standard deviations of numerical variables

Examples

```
## Not run:
# Train a model inside add_predictions call.
local_exploration1 <- add_predictions(dataset_for_local_exploration,
                                     black_box_model = "regr.svm",
                                     data = wine)

# Pass trained model to the function.
svm_model <- svm(quality ~., data = wine)
local_exploration2 <- add_predictions(dataset_for_local_exploration,
                                     black_box_model = svm_model)

## End(Not run)
```

euclidean_kernel	<i>LIME kernel equal to the inverse of euclidean distance.</i>
------------------	--

Description

LIME kernel equal to the inverse of euclidean distance.

Usage

```
euclidean_kernel(explained_instance, simulated_instance)
```

Arguments

```
explained_instance
    explained instance
simulated_instance
    new observation
```

Value

```
numeric
```

fit_explanation	<i>Fit white box model to the simulated data.</i>
-----------------	---

Description

Fit white box model to the simulated data.

Usage

```
fit_explanation(
  live_object,
  white_box = "regr.lm",
  kernel = gaussian_kernel,
  standardize = FALSE,
  selection = FALSE,
  response_family = "gaussian",
  predict_type = "response",
  hyperpars = list()
)
```

Arguments

live_object	List return by add_predictions function.
white_box	String, learner name recognized by mlr package.
kernel	function which will be used to calculate distance between simulated observations and explained instance.
standardize	If TRUE, numerical variables will be scaled to have mean 0, variance 1 before fitting explanation model.
selection	If TRUE, variable selection based on glmnet implementation of LASSO will be performed.
response_family	family argument to glmnet (and then glm) function. Default value is "gaussian"

predict_type	Argument passed to mlr::makeLearner() argument "predict.type". Defaults to "response".
hyperpars	Optional list of values of hyperparameteres of a model.

Value

List of class "live_explainer" that consists of

data	Dataset used to fit explanation model (may have less column than the original)
model	Fitted explanation model
explained_instance	Instance that is being explained
weights	Weights used in model fitting
selected_variables	Names of selected variables

Examples

```
## Not run:
fitted_explanation <- fit_explanation(local_exploration1, "regr.lm", selection = TRUE)

## End(Not run)
```

gaussian_kernel	<i>LIME kernel from the original article with sigma = 1.</i>
-----------------	--

Description

LIME kernel from the original article with sigma = 1.

Usage

```
gaussian_kernel(explained_instance, simulated_instance)
```

Arguments

explained_instance	explained instance
simulated_instance	new observation

Value

numeric

identity_kernel	<i>LIME kernel that treats all observations as equally similar to observation of interest.</i>
-----------------	--

Description

LIME kernel that treats all observations as equally similar to observation of interest.

Usage

```
identity_kernel(explained_instance, simulated_instance)
```

Arguments

```
explained_instance
                        explained instance
simulated_instance
                        new observation
```

Value

```
numeric
```

live	<i>live: visualizing interpretable models to explain black box models.</i>
------	--

Description

This package aims to help locally fit and visualize interpretable models similarly to LIME methodology. Interface provided by mlr package is used. Tools are provided to create a simulated dataset of similar observations, fit chosen white box models (GLM and CART in particular) and visualize them. The methodology is based on Tulio Ribeiro, Singh, Guestrin (2016) <doi:10.1145/2939672.2939778>. More details can be found in Staniak, Biecek (2018) <doi:10.32614/RJ-2018-072>.

Important functions

[sample_locally](#) generates a dataset that will be used for local exploration. [add_predictions](#) adds black box model predictions to simulated dataset. [fit_explanation](#) fits a chosen white box model to simulated dataset. generic [plot](#) function visualizes fitted model. [local_approximation](#) function can be used with DALEX explainers to perform all the steps of local model exploration.

Example datasets

wine Data on wine quality taken from Modeling wine preferences by data mining from physico-chemical properties

live_shiny	<i>Function that starts a Shiny app which helps use LIVE.</i>
------------	---

Description

Function that starts a Shiny app which helps use LIVE.

Usage

```
live_shiny(train_data, black_box_model, target, explained_data = train_data)
```

Arguments

train_data	dataset from which observations will be sampled.
black_box_model	Pre-trained model with predict interface.
target	character, name of the response variable.
explained_data	Data frame with predictions to explain.

Value

shiny app

local_approximation	<i>Fit local model around the observation: shortcut for DALEX explainer objects</i>
---------------------	---

Description

Fit local model around the observation: shortcut for DALEX explainer objects

Usage

```
local_approximation(
  explainer,
  observation,
  target_variable_name,
  n_new_obs,
  local_model = "regr.lm",
  select_variables = F,
  predict_type = "response",
  kernel_type = gaussian_kernel,
  ...
)
```

Arguments

explainer	a model to be explained, preprocessed by the DALEX::explain function
observation	a new observation for which predictions need to be explained
target_variable_name	name of the response variable as a character
n_new_obs	Number of observation in the simulated dataset
local_model	Character specifying mlr learner to be used as a local model
select_variables	If TRUE, variable selection will be performed while fitting the local linear model
predict_type	Argument passed to mlr::makeLearner() argument "predict.type" while fitting the local model. Defaults to "response"
kernel_type	Function which will be used to calculate distances from simulated observation to explained instance
...	Arguments to be passed to sample_locally function

Value

object of class live_explainer. More details in fit_explanation function help.

Examples

```
## Not run:
data('wine')
library(randomForest)
library(DALEX)
rf <- randomForest(quality~., data = wine)
expl <- explain(rf, wine, wine$quality)
live_expl <- local_approximation(expl, wine[5, ], "quality", 500)

## End(Not run)
```

local_permutation_importance

Local permutation variable importance

Description

This function calculates local variable importance (variable drop-out) by finding top_n observations closest to the explained instance, performing permutation variable importance and using weighted mean square error as loss function with weights equal to 1 - Gower distances of the closest observations to the explained instance.

plot.live_explainer	<i>Plotting white box models.</i>
---------------------	-----------------------------------

Description

Plotting white box models.

Usage

```
## S3 method for class 'live_explainer'  
plot(x, type = "waterfall", ...)
```

Arguments

x	List returned by fit_explanation function.
type	Chr, "forest" or "waterfall" depending on which type of plot is to be created. if lm/glm model is used as interpretable approximation.
...	Additional parameters that will be passed to plot.broken or plot method. In particular, when number of features is large, top_features argument can be set in plot.broken.

Value

plot (ggplot2 or base)

Examples

```
## Not run:  
# Forest plot for regression  
plot(fitted_explanation1, type = "forest")  
# Waterfall plot  
plot(fitted_explanation1, type = "waterfall")  
# Plot decision tree  
plot(fitted_explanation2)  
  
## End(Not run)
```

```
plot.local_permutation_importance
```

Plot local permutation importance

Description

Plot local permutation importance

Usage

```
## S3 method for class 'local_permutation_importance'  
plot(x, ...)
```

Arguments

x	Object of class local_permutation_importance
...	Optional arguments, currently ignored

Value

ggplot2 object

```
print.live_explainer
```

Generic print function for live explainer

Description

Generic print function for live explainer

Usage

```
## S3 method for class 'live_explainer'  
print(x, ...)
```

Arguments

x	Object created using fit_explanation function
...	other arguments

print.live_explorer	<i>Generic print function for class live_explorer</i>
---------------------	---

Description

Generic print function for class live_explorer

Usage

```
## S3 method for class 'live_explorer'  
print(x, ...)
```

Arguments

x	Object created by sample_locally function or add_predictions function
...	Other arguments

print.local_permutation_importance	<i>Print method for local_permutation_importance class</i>
------------------------------------	--

Description

Print method for local_permutation_importance class

Usage

```
## S3 method for class 'local_permutation_importance'  
print(x, ...)
```

Arguments

x	Object of class local_permutation_importance
...	Optional arguments, currently ignored

sample_locally	<i>Generate dataset for local exploration.</i>
----------------	--

Description

Generate dataset for local exploration.

Usage

```
sample_locally(
  data,
  explained_instance,
  explained_var,
  size,
  method = "live",
  fixed_variables = NULL,
  seed = NULL,
  ...
)
```

Arguments

data	Data frame from which new dataset will be simulated.
explained_instance	One row data frame with the same variables as in data argument. Local exploration will be performed around this observation.
explained_var	Name of a column with the variable to be predicted.
size	Number of observations in a simulated dataset.
method	If "live", new observations will be created by changing one value per observation. If "permute", new observation will be created by permuting all columns of data. If "normal", numerical features will be sampled from multivariate normal distribution specified by ... arguments mu and Sigma.
fixed_variables	names or numeric indexes of columns which will not be changed while sampling.
seed	Seed to set before sampling. If NULL, results will not be reproducible.
...	Mean and covariance matrix for normal sampling method.

Value

list of class "live_explorer" consisting of

data	Dataset generated by sample_locally function with response variable.
target	Name of the response variable.
explained_instance	Instance that is being explained.

sampling_method Name of used sampling method
fixed_variables Names of variables which were not sampled
sdeviations Standard deviations of numerical variables

Examples

```
## Not run:  
dataset_for_local_exploration <- sample_locally(data = wine,  
                                                explained_instance = wine[5, ],  
                                                explained_var = "quality",  
                                                size = 50)  
  
## End(Not run)
```

wine	<i>Red wine characteristics and quality.</i>
------	--

Description

Popular dataset related to wine samples from north Portugal.

Usage

```
wine
```

Format

Data frame with 1599 rows and 12 columns.

References

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

Index

* datasets

wine, [14](#)

add_predictions, [2](#), [6](#)

euclidean_kernel, [3](#)

fit_explanation, [4](#), [6](#)

gaussian_kernel, [5](#)

identity_kernel, [6](#)

live, [6](#)

live_shiny, [7](#)

local_approximation, [6](#), [7](#)

local_permutation_importance, [8](#)

plot, [6](#)

plot.live_explainer, [10](#)

plot.local_permutation_importance, [11](#)

print.live_explainer, [11](#)

print.live_explorer, [12](#)

print.local_permutation_importance, [12](#)

sample_locally, [6](#), [13](#)

wine, [14](#)