Package 'DALEX2'

December 23, 2018

Title Descriptive mAchine Learning EXplanations

Version 0.9

Description Machine Learning models are widely used and have various applications in classification or regression tasks.
Due to increasing computational power, availability of new data sources and new methods, ML models are more and more complex.
Models created with techniques like boosting, bagging of neural networks are true black boxes.
It is hard to trace the link between input variables and model outcomes.
They are used because of high performance, but lack of interpretability is one of their weakest sides.
In many applications we need to know, understand or prove how input variables are used in the model and what impact do they have on final model prediction.
DALEX2 is a collection of tools that help to understand how complex predictive models are working.
DALEX2 is a part of DrWhy universe for tools for Explanation, Exploration and Visualisation for Predictive Models.

Depends R (>= 3.1)

License GPL

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests knitr, randomForest, tibble, testthat

URL https://ModelOriented.github.io/DALEX2/

BugReports https://github.com/ModelOriented/DALEX2/issues

NeedsCompilation no

Author Przemyslaw Biecek [aut, cre]

Maintainer Przemyslaw Biecek <przemyslaw.biecek@gmail.com>

Repository CRAN

Date/Publication 2018-12-23 16:20:14 UTC

R topics documented:

`apartments` *Apartments Data*

Description

Datasets `apartments` and `apartments_test` are artificial, generated form the same model. Structure of the dataset is copied from real-world dataset from `PBImisc` package, but they were generated in a way to mimic effect of Anscombe quartet for complex black box models.

Usage

```
data(apartments)
```

Format

a data frame with 1000 rows and 6 columns

Details

- m2.price - price per square meter
- surface - apartment area in square meters
- n.rooms - number of rooms (correlated with surface)
- district - district in which apartment is located, factor with 10 levels
- floor - floor
- construction.date - construction year

dragons *Dragon Data*

Description

Datasets `dragons` and `dragons_test` are artificial, generated form the same ground truth model, but with sometimes different data distridution.

Usage

```
data(dragons)
```

Format

a data frame with 2000 rows and 8 columns

Details

Values are generated in a way to: - have nonlinearity in year_of_birth and height - have concept drift in the test set

- year_of_birth - year in which the dragon was born. Negative year means year BC, eg: -1200 = 1201 BC
- year_of_discovery - year in which the dragon was found.
- height - height of the dragon in yards.
- weight - weight of the dragon in tons.
- scars - number of scars.
- colour - colour of the dragon.
- number_of_lost_teeth - number of teeth that the dragon lost.
- life_length - life length of the dragon.

explain.default *Create Model Explainer*

Description

Black-box models may have very different structures. This function creates a unified representation of a model, which can be further processed by various explainers.

Usage

```
explain.default(model, data = NULL, y = NULL,
  predict_function = yhat, link = I, ..., label = tail(class(model),
  1))

explain(model, data = NULL, y = NULL, predict_function = yhat,
  link = I, ..., label = tail(class(model), 1))
```

Arguments

model	object - a model to be explained
data	data.frame or matrix - data that was used for fitting. If not provided then will be extracted from the model
y	numeric vector with outputs / scores. Currently used only by variable_dropout() explainer.
predict_function	function that takes two arguments: model and new data and returns numeric vector with predictions
link	function - a transformation/link function that shall be applied to raw model predictions
...	other parameters
label	character - the name of the model. By default it's extracted from the 'class' attribute of the model

Details

Please NOTE, that the model is actually the only required argument. But some explainers may require that others will be provided too.

Value

An object of the class 'explainer'.

It's a list with following fields:

- model the explained model
- data the dataset used for training
- predict_function function that may be used for model predictions, shall return a single numerical value for each observation.
- class class/classes of a model
- label label, by default it's the last value from the class vector, but may be set to any character.

Examples

```
apartments_lm <- lm(m2.price ~ ., data = apartments)
apartments_lm_ex <- explain(apartments_lm, data = apartments, label = "apartments_lm")
apartments_lm_ex

 ## Not run:
library("breakDown2")
wine_lm_model4 <- lm(quality ~ pH + residual.sugar + sulphates + alcohol, data = wine)
wine_lm_explainer4 <- explain(wine_lm_model4, data = wine, label = "model_4v")
wine_lm_explainer4

library("randomForest")
wine_rf_model4 <- randomForest(quality ~ pH + residual.sugar + sulphates + alcohol, data = wine)
wine_rf_explainer4 <- explain(wine_rf_model4, data = wine, label = "model_rf")
wine_rf_explainer4

## End(Not run)
```

HR *Human Resources Data*

Description

Datasets HR and HR_test are artificial, generated form the same model. Structure of the dataset is based on a real data, from Human Resources department with information which employees were promoted, which were fired.

Usage

```
data(HR)
```

Format

a data frame with 10000 rows and 6 columns

Details

Values are generated in a way to: - have interaction between age and gender for the 'fired' variable - have non monotonic relation for the salary variable - have linear effects for hours and evaluation.

- gender - gender of an employee.
- age - gender of an employee in the moment of evaluation.
- hours - average number of working hours per week.
- evaluation - evaluation in the scale 2 (bad) - 5 (very good).
- salary - level of salary in the scale 0 (lowest) - 5 (highest).
- status - target variable, either 'fired' or 'promoted' or 'ok'.

loss_sum_of_squares *Loss Functions*

Description

Loss functions that can be used for model comparions ane feature importance estimation

Usage

```
loss_sum_of_squares(observed, predicted)
```

Arguments

observed	true observed lables
predicted	scores predicted with a model

Value

numeric model fit assessment. The lower - the better is the model

print.explainer *Prints Explainer Summary*

Description

Prints Explainer Summary

Usage

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

Arguments

x	a model expaliner created with the ‘explain‘ function
...	other parameters

Examples

```
apartments_lm <- lm(m2.price ~ ., data = apartments)
apartments_lm_ex <- explain(apartments_lm, data = apartments, label = "apartments_lm")
apartments_lm_ex

 ## Not run:
library("breakDown2")
wine_lm_model4 <- lm(quality ~ pH + residual.sugar + sulphates + alcohol, data = wine)
wine_lm_explainer4 <- explain(wine_lm_model4, data = wine, label = "model_4v")
wine_lm_explainer4

library("randomForest")
wine_rf_model4 <- randomForest(quality ~ pH + residual.sugar + sulphates + alcohol, data = wine)
wine_rf_explainer4 <- explain(wine_rf_model4, data = wine, label = "model_rf")
wine_rf_explainer4

## End(Not run)
```

---

theme_mi2 *MI^2 Theme*

Description

Themem oject. In order to avoid an additional dependency from ggplot2, we have here a raw vaerion of an theme object

Usage

```
theme_mi2()
```

Value

theme object that can be added to ggplot2 plots

---

yhat *Wrapper over the predict function*

Description

This function is just a wrapper over the predict function. It sets different default parameters for models from different classes. For example: for classification random Forest is forces the output to be probabilities not classes itself.

Usage

```
yhat(X.model, newdata, ...)

## S3 method for class 'lm'
yhat(X.model, newdata, ...)

## S3 method for class 'randomForest'
yhat(X.model, newdata, ...)

## Default S3 method:
yhat(X.model, newdata, ...)
```

Arguments

X.model	object - a model to be explained
newdata	data.frame or matrix - observations for prediction
...	other parameters that will be passed to the ‘predict‘ function

Details

We use the ‘yhat‘ name instead of ‘predict‘ since we need different defaults that the one set in the ‘predict‘ function.

Value

An numeric matrix of predictions. Can have more than one collumn.

Index