The breakDown package explains predictions from black-box models, such as random forest, xgboost, svm or neural networks (it works for lm and glm as well). As a result you gets decomposition of model prediction that can be attributed to particular variables.

The version 0.3 has a new function break\_down. It identifies pairwise interactions of variables. So if the model is not additive, then instead of seeing effects of single variables you will see effects for interactions.  
It’s easy to use this function. See an example below.  
  
HR is an artificial dataset. The break\_down function correctly identifies interaction between gender and age.

Breakdown Function with Example

This function implements decomposition of model predictions with identification of interactions. The complexity of this function is O(2\*p) for additive models and O(2\*p^2) for interactions. This function works in similar way to step-up and step-down greedy approaximations, the main difference is that in the fisrt step the order of variables is determied. And in the second step the impact is calculated.

break\_down(explainer, new\_observation, check\_interactions = TRUE,

keep\_distributions = FALSE)

 Arguments

|  |  |
| --- | --- |
| **explainer** | a model to be explained, preprocessed by function `DALEX::explain()`. |
| **new\_observation** | a new observation with columns that corresponds to variables used in the model |
| **check\_interactions** | the orgin/baseline for the `breakDown`` plots, where the rectangles start. It may be a number or a character "Intercept". In the latter case the orgin will be set to model intercept. |
| **keep\_distributions** | if TRUE, then the distribution of partial predictions is stored in addition to the average. |

Value

an object of the broken class

Examples

library("DALEX")

library("breakDown")

library("randomForest")

set.seed(1313)

# example with interaction

# classification for HR data

model <- [randomForest](http://www.rdocumentation.org/packages/randomForest/topics/randomForest)(status ~ . , data = HR)

new\_observation <- HRTest[1,]

data <- HR[1:1000,]

predict.function <- function(m,x) predict(m,x, type = "prob")[,1]

explainer\_rf\_fired <- [explain](http://www.rdocumentation.org/packages/DALEX/topics/explain)(model,

data = HR[1:1000,1:5],

y = HR$status[1:1000] == "fired",

predict\_function = function(m,x) predict(m,x, type = "prob")[,1],

label = "fired")

bd\_rf <- break\_down(explainer\_rf\_fired,

new\_observation,

keep\_distributions = TRUE)

bd\_rf

#> contribution

#> (Intercept) 0.386

#> \* hours = 42 0.231

#> \* salary = 2 -0.216

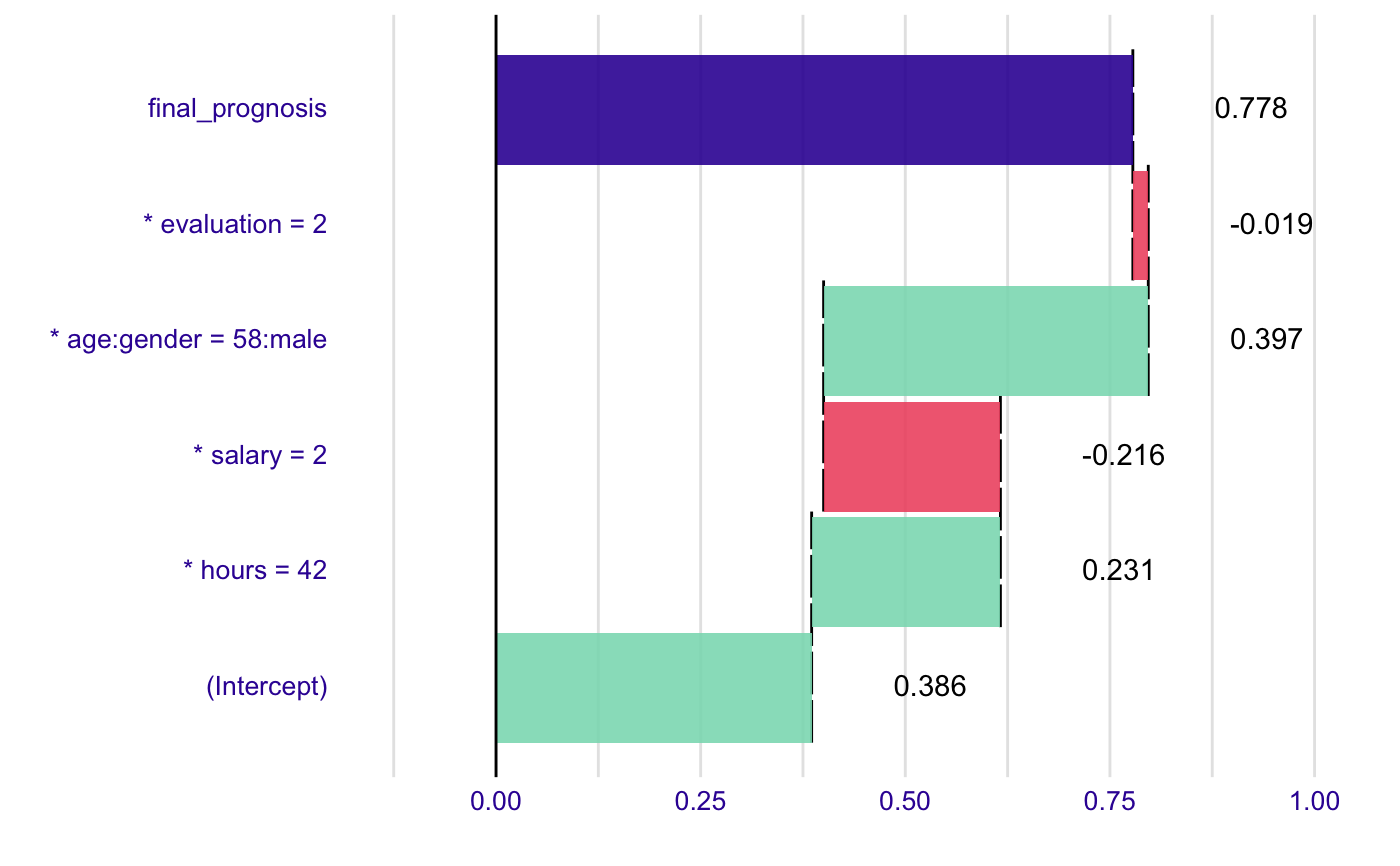
#> \* age:gender = 58:male 0.397

#> \* evaluation = 2 -0.019

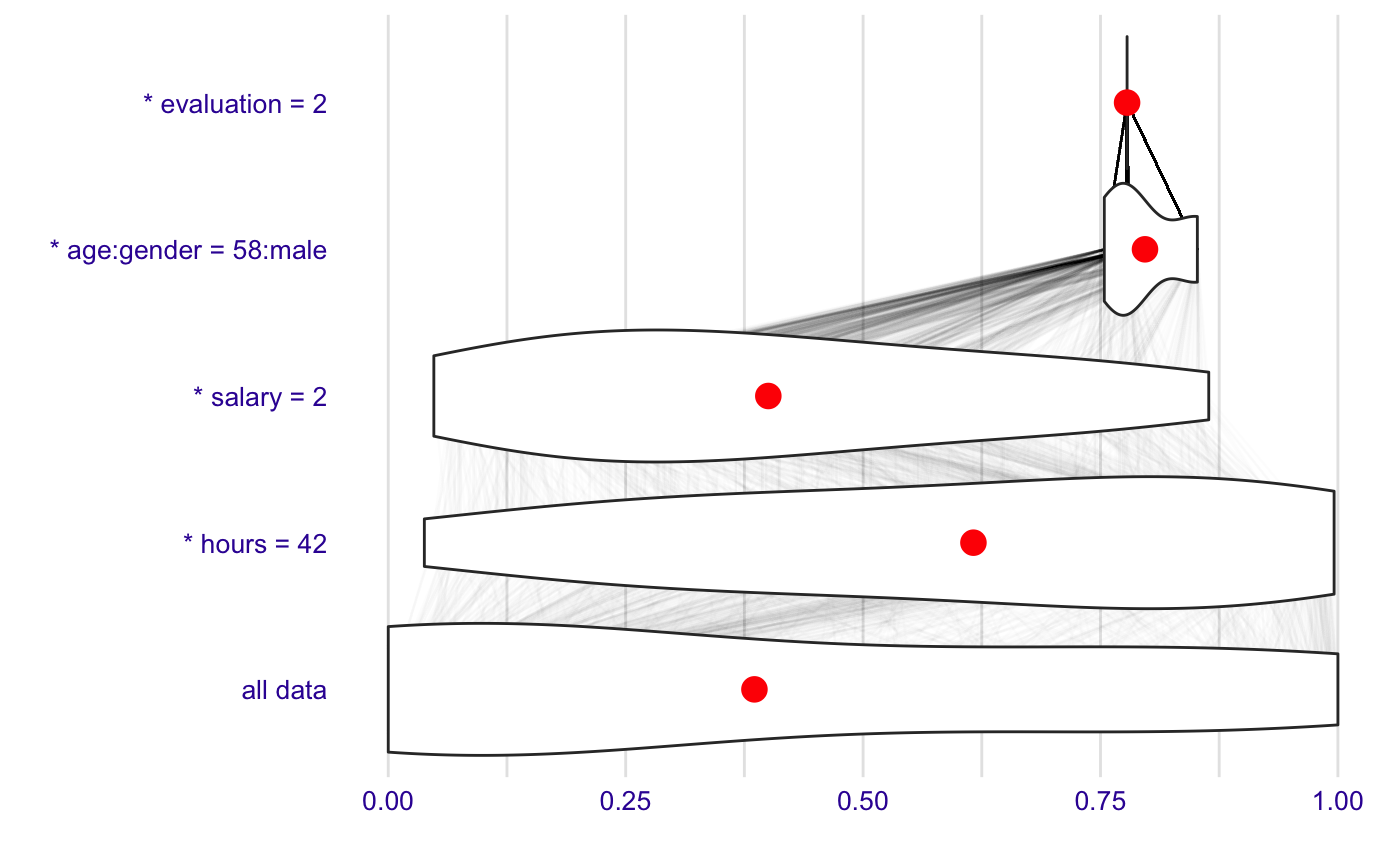
#> final\_prognosis 0.778

#> baseline: 0

plot(bd\_rf)



plot(bd\_rf, plot\_distributions = TRUE)



bd\_rf <- break\_down(explainer\_rf\_fired,

new\_observation,

check\_interactions = FALSE,

keep\_distributions = TRUE)

bd\_rf

#> contribution

#> (Intercept) 0.386

#> \* hours = 42 0.231

#> \* salary = 2 -0.216

#> \* evaluation = 2 0.006

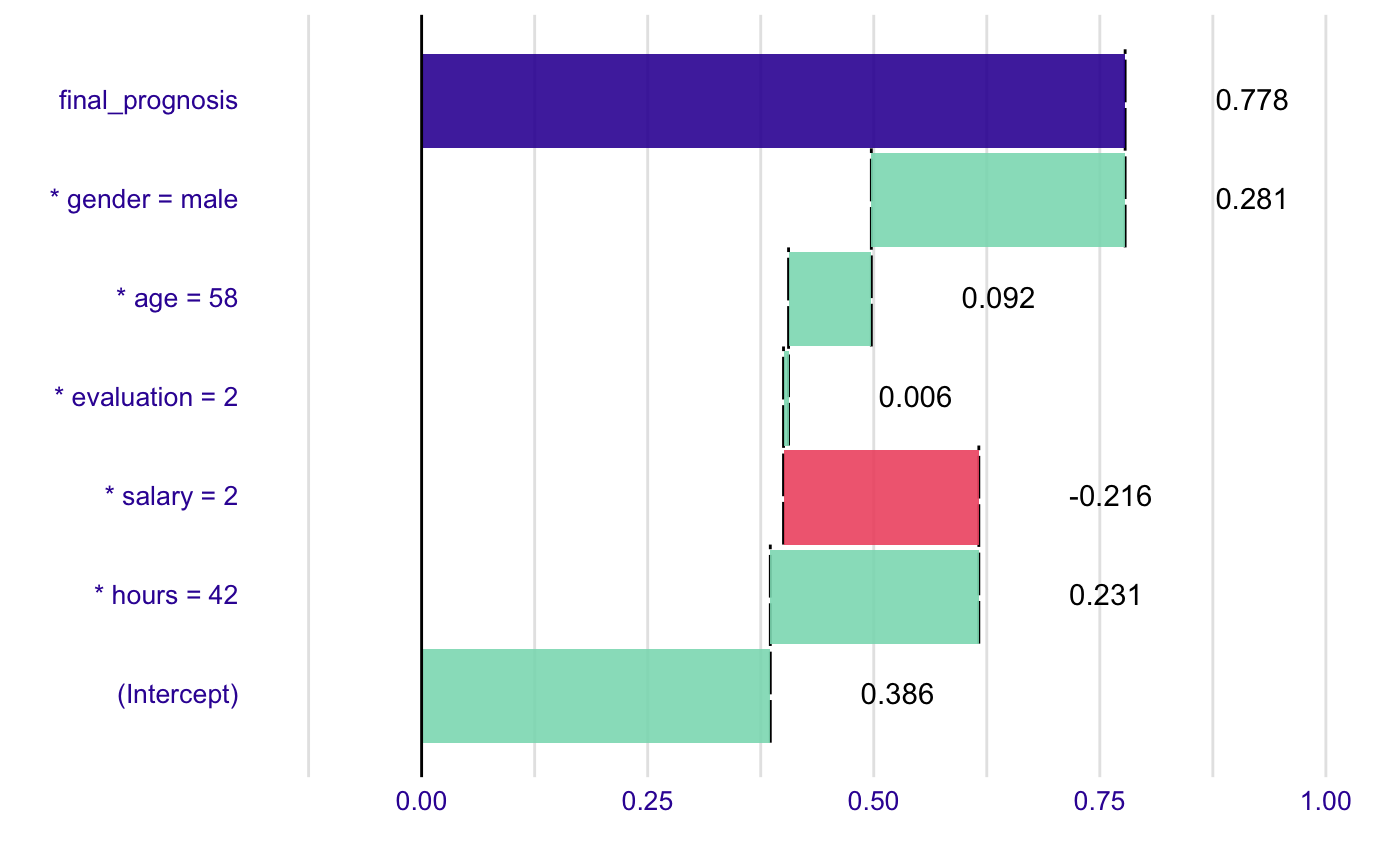
#> \* age = 58 0.092

#> \* gender = male 0.281

#> final\_prognosis 0.778

#> baseline: 0

plot(bd\_rf)



# example for regression - apartment prices

# here we do not have intreactions

model <- [randomForest](http://www.rdocumentation.org/packages/randomForest/topics/randomForest)(m2.price ~ . , data = apartments)

explainer\_rf <- [explain](http://www.rdocumentation.org/packages/DALEX/topics/explain)(model,

data = apartmentsTest[1:1000,2:6],

y = apartmentsTest$m2.price[1:1000],

label = "rf")

bd\_rf <- break\_down(explainer\_rf,

apartmentsTest[1,],

check\_interactions = FALSE,

keep\_distributions = TRUE)

bd\_rf

#> contribution

#> (Intercept) 3487.722

#> \* district = Srodmiescie 1034.737

#> \* surface = 130 -315.991

#> \* no.rooms = 5 -163.113

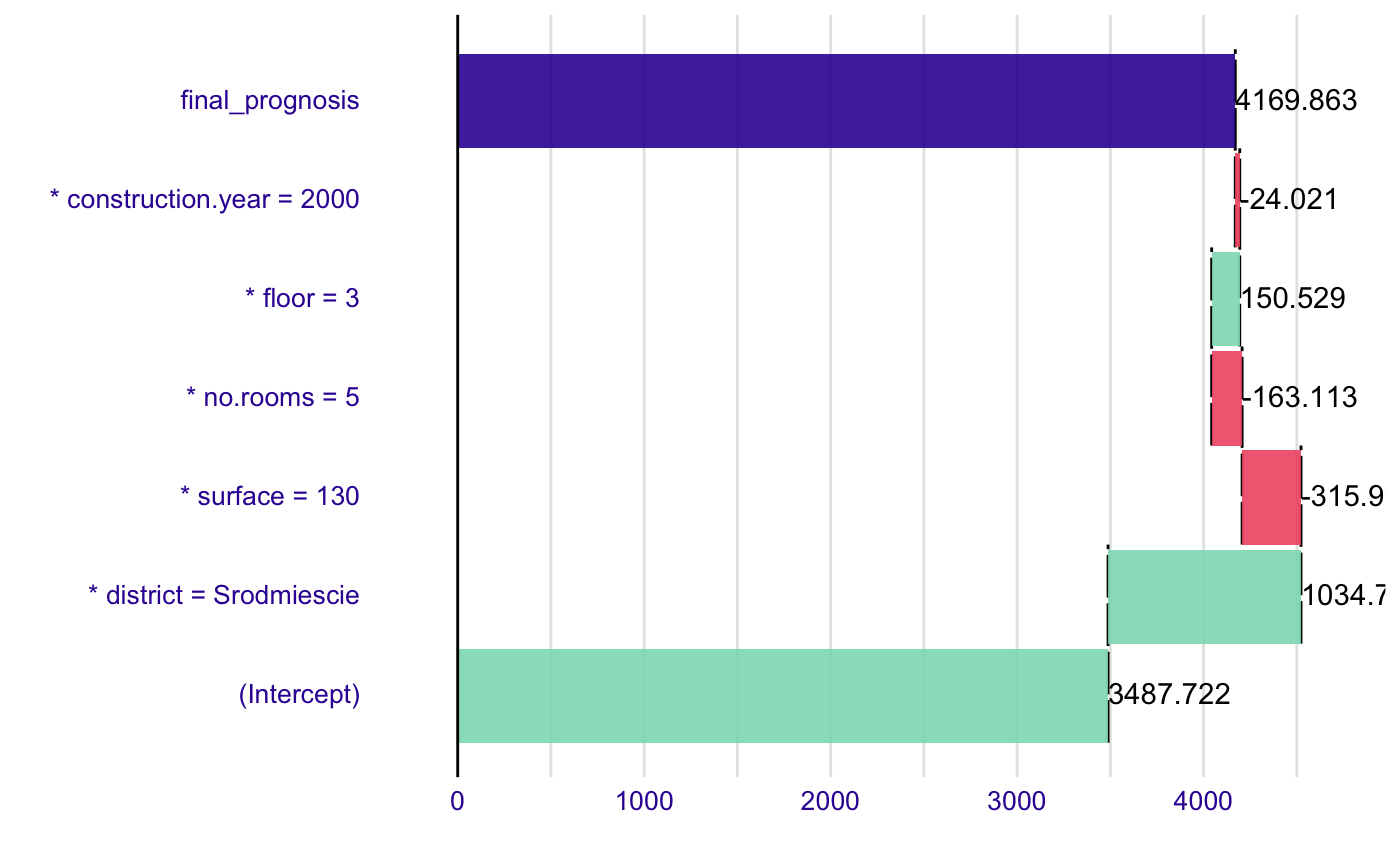
#> \* floor = 3 150.529

#> \* construction.year = 2000 -24.021

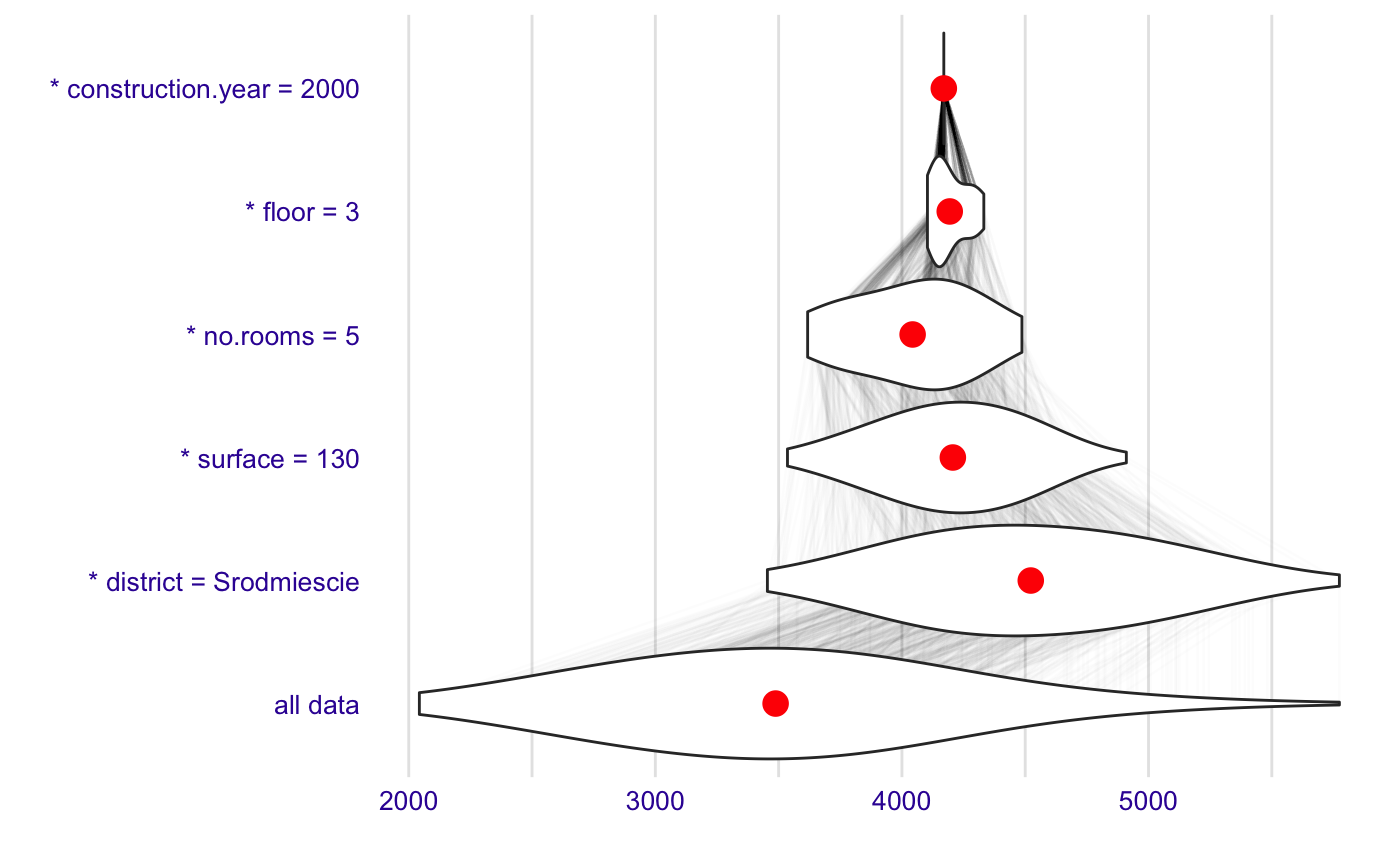
#> final\_prognosis 4169.863

#> baseline: 0

plot(bd\_rf)



plot(bd\_rf, plot\_distributions = TRUE)



#

# Create a model for classification

library("DALEX")

library("randomForest")

model <- randomForest(status ~ . , data = HR)

#

# Create a DALEX explainer

explainer\_rf\_fired <- explain(model,

data = HR, y = HR$status == "fired",

predict\_function = function(m,x) predict(m,x, type = "prob")[,1])

#

# Calculate variable attributions

new\_observation <- HRTest[1,]

library("breakDown")

bd\_rf <- break\_down(explainer\_rf\_fired,

new\_observation,

keep\_distributions = TRUE)

bd\_rf

#> contribution

#> (Intercept) 0.386

#> \* hours = 42 0.231

#> \* salary = 2 -0.216

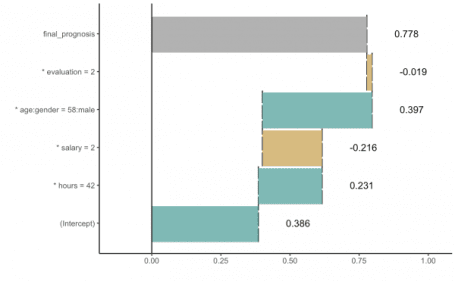
#> \* age:gender = 58:male 0.397

#> \* evaluation = 2 -0.019

#> final\_prognosis 0.778

#> baseline: 0

plot(bd\_rf)

Figure below shows that a single prediction was decomposed into 4 parts. One of them is related to the interaction between age and gender.  


BreakDown is a part of DALEXverse – collection of tools for visualisation, exploration and explanation of complex machine learning models.

Till the end of September I am visiting UC Davis and UC Berkeley. Happy to talk about DALEX explainers, XAI and related stuff.

