Package: modelDown (via r-universe)

November 1, 2024

Title Make Static HTML Website for Predictive Models

Version 1.1

Description Website generator with HTML summaries for predictive
models. This package uses 'DALEX' explainers to describe global
model behavior. We can see how well models behave (tabs: Model
Performance, Auditor), how much each variable contributes to
predictions (tabs: Variable Response) and which variables are
the most important for a given model (tabs: Variable
Importance). We can also compare Concept Drift for pairs of
models (tabs: Drifter). Additionally, data available on the
website can be easily recreated in current R session. Work on
this package was financially supported by the NCN Opus grant
2017/27/B/ST6/01307 at Warsaw University of Technology, Faculty
of Mathematics and Information Science.

Depends R (>= 3.4.0)

License Apache License 2.0

Encoding UTF-8

LazyData true

Imports DALEX (>= 1.0), auditor (>= 0.3.0), ggplot2 (>= 3.1.0),
whisker (>= 0.3-2), DT (>= 0.4), kableExtra (>= 0.9.0), psych
(>= 1.8.4), archivist (>= 2.1.0), svglite (>= 1.2.1), devtools
(>= 2.0.1), breakDown (>= 0.1.6), drifter (>= 0.2.1)

Suggests ranger, testthat, useful, covr

RoxygenNote 7.1.1

URL https://github.com/ModelOriented/modelDown

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

Repository https://modeloriented.r-universe.dev

RemoteUrl https://github.com/modeloriented/modeldown

RemoteRef HEAD

RemoteSha c1f230791c2c6412e7239b98e11db5bba05e8248

Contents

---

modelDown	*Generates a website with HTML summaries for predictive models*

Description

Generates a website with HTML summaries for predictive models

Usage

```
modelDown(
  ...,
  modules = c("auditor", "drifter", "model_performance", "variable_importance",
    "variable_response"),
  output_folder = "output",
  repository_name = "repository",
  should_open_website = interactive()
)
```

Arguments

...	one or more explainers created with `DALEX::explain()` function. Pair of explainer could be provided to check drift of models
modules	modules that should be included in the website
output_folder	folder where the website will be saved
repository_name	
	name of local archivist repository that will be created
should_open_website	
	should generated website be automatically opened in default browser

Details

Additional arguments that could by passed by name:

- remote_repository_path Path to remote repository that stores folder with archivist repository. If not provided, links to local repository will be shown.

- device Device to use. Tested for "png" and "svg", but values from `ggplot2::ggsave` function should be working fine. Defaults to "png".

- vr.vars variables which will be examined in Variable Response module. Defaults to all variables. Example vr.vars = c("var1", "var2")

- vr.type types of examinations which will be conducteed in Variable Response module. Defaults to "partial". Example vr.type = c("partial", "conditional", "accumulated")

Author(s)

Przemysław Biecek, Magda Tatarynowicz, Kamil Romaszko, Mateusz Urbański

Examples

```
if(FALSE){
require("ranger")
require("breakDown")
require("DALEX")


# Generate simple modelDown page
HR_data_selected <- HR_data[1000:3000,]
HR_glm_model <- glm(left~., HR_data_selected, family = "binomial")
explainer_glm <- explain(HR_glm_model, data=HR_data_selected, y = HR_data_selected$left)

modelDown::modelDown(explainer_glm,
                       modules = c("model_performance", "variable_importance",
                                   "variable_response"),
                       output_folder = tempdir(),
                       repository_name = "HR",
                       device = "png",
                       vr.vars= c("average_montly_hours"),
                       vr.type = "partial")

# More complex example with all modules
HR_ranger_model <- ranger(as.factor(left) ~ .,
                data = HR_data, num.trees = 500, classification = TRUE, probability = TRUE)
explainer_ranger <- explain(HR_ranger_model,
                       data = HR_data, y = HR_data$left, function(model, data) {
 return(predict(model, data)$prediction[,2])
}, na.rm=TRUE)

# Two glm models used for drift detection
HR_data1 <- HR_data[1:4000,]
HR_data2 <- HR_data[4000:nrow(HR_data),]
HR_glm_model1 <- glm(left~., HR_data1, family = "binomial")
HR_glm_model2 <- glm(left~., HR_data2, family = "binomial")
explainer_glm1 <- explain(HR_glm_model1, data=HR_data1, y = HR_data1$left)
explainer_glm2 <- explain(HR_glm_model2, data=HR_data2, y = HR_data2$left)

modelDown::modelDown(list(explainer_glm1, explainer_glm2),
  modules = c("auditor", "drifter", "model_performance", "variable_importance",
              "variable_response"),
  output_folder = tempdir(),
  repository_name = "HR",
  remote_repository_path = "some_user/remote_repo_name",
  device = "png",
  vr.vars= c("average_montly_hours", "time_spend_company"),
  vr.type = "partial")
}
```

Index

modelDown,