

Interactive Studio for Explanatory Model Analysis



Overview

The `modelStudio` package **automates the explanatory analysis of machine learning predictive models**. It generates advanced interactive model explanations in the form of a **serverless HTML site** with only one line of code. This tool is model-agnostic, therefore compatible with most of the black-box predictive models and frameworks (e.g. `mlr/mlr3`, `xgboost`, `caret`, `h2o`, `parsnip`, `tidymodels`, `scikit-learn`, `lightgbm`, `keras/tensorflow`).

The main `modelStudio()` function computes various (instance and model-level) explanations and produces a **customisable dashboard**, which consists of multiple panels for plots with their short descriptions. It is possible to easily **save** the dashboard and **share** it with others. Tools for [Explanatory Model Analysis](#) unite with tools for Exploratory Data Analysis to give a broad overview of the model behavior.



The `modelStudio` package is a part of the [DrWhy.AI](#) universe.

Installation

```
# Install from CRAN:
install.packages("modelStudio")

# Install the development version from GitHub:
devtools::install_github("ModelOriented/modelStudio")
```

Simple demo

```
library("DALEX")
library("ranger")
library("modelStudio")

# fit a model
model <- ranger(score ~., data = happiness_train)

# create an explainer for the model
explainer <- explain(model,
  data = happiness_test,
  y = happiness_test$score,
  label = "Random Forest")

# make a studio for the model
modelStudio(explainer)
```

Save the output in the form of a HTML file - [Demo Dashboard](#).



R & Python examples [more](#)

The `modelStudio()` function uses `DALEX` explainers created with `DALEX::explain()` or `DALEXtra::explain_*`.

```
# packages for the explainer objects
install.packages("DALEX")
install.packages("DALEXtra")
```

mlr [dashboard](#)

Make a studio for the regression `ranger` model on the `apartments` data.

```
▼ code

# load packages and data
library(mlr)
library(DALEXtra)
library(modelStudio)

data <- DALEX::apartments

# split the data
index <- sample(1:nrow(data), 0.7*nrow(data))
train <- data[index,]
test <- data[-index,]

# fit a model
task <- makeRegrTask(id = "apartments", data = train, target = "m2.price")
learner <- makeLearner("regr.ranger", predict.type = "response")
model <- train(learner, task)

# create an explainer for the model
explainer <- explain_mlr(model,
  data = test,
  y = test$m2.price,
  label = "mlr")

# pick observations
new_observation <- test[1:2,]
rownames(new_observation) <- c("id1", "id2")

# make a studio for the model
modelStudio(explainer, new_observation)
```

xgboost [dashboard](#)

Make a studio for the classification `xgboost` model on the `titanic` data.

```
► code
```

The `modelStudio()` function uses `daLEX` explainers created with `daLEX.Explainer()`.

```
:: package for the Explainer object
pip install daLEX -U
```

Use `pickle` Python module and `reticulate` R package to easily make a studio for a model.

```
# package for pickle load
install.packages("reticulate")
```

scikit-learn [dashboard](#)

Make a studio for the regression `Pipeline` `SVR` model on the `fifa` data.

```
► code
```

lightgbm [dashboard](#)

Make a studio for the classification `Pipeline` `LGBMClassifier` model on the `titanic` data.

```
► code
```

Save & share

Save `modelStudio` as a HTML file using buttons on the top of the RStudio Viewer or with `r2d3::save_d3_html()`.



Interactive Studio for Random Forest Model

Citations

If you use `modelStudio`, please cite our [JOSS](#) article:

```
@article{baniecki2019modelstudio,
  title = {{modelStudio: Interactive Studio with Explanations for ML Predictive M
  author = {Hubert Baniecki and Przemysław Biecek},
  journal = {Journal of Open Source Software},
  year = {2019},
  volume = {4},
  number = {43},
  pages = {1798},
  url = {https://doi.org/10.21105/joss.01798}
}
```

For a description and evaluation of the Interactive EMA process, refer to our [DAMI](#) article:

```
@article{baniecki2023grammar,
  title = {The grammar of interactive explanatory model analysis},
  author = {Hubert Baniecki and Dariusz Parzych and Przemysław Biecek},
  journal = {Data Mining and Knowledge Discovery},
  year = {2023},
  pages = {1--37},
  url = {https://doi.org/10.1007/s10618-023-00924-w}
}
```

More resources

- Introduction to the plots: [Explanatory Model Analysis: Explore, Explain, and Examine Predictive Models](#)
- Vignettes: [perks and features](#), [R & Python examples](#), [modelStudio in R Markdown HTML](#)
- Changelog: [NEWS](#)
- Conference poster: [ML in PL 2019](#)

Acknowledgments

Work on this package was financially supported by the National Science Centre (Poland) grant 2016/21/B/ST6/02176 and National Centre for Research and Development grant POIR.01.01.01-00-0328/17.