modelDown: automated website generator with interpretable documentation for predictive machine learning models

Kamil Romaszko[1], **Magda Tatarynowicz**[1], **Mateusz Urbański**[1], **and Przemysław Biecek**[1]

1 Faculty of Mathematics and Information Science, Warsaw University of Technology

Introduction

We live in times where machine learning is used almost everywhere. With the large number of accessible tools available, generation of predictive model for your own benefit is fairly straightforward.

The models may be simple to create and use, but most of the time they are not simple to understand. Larger models may contain thousands or millions of parameters. It can be challenging to see how the model works and how the inputs impact the obtained results. Numerous tools for explainable Artificial Intelligence can help, but it may be cumbersome and repetitive to use different tools every time you test the model.

In this paper, we introduce a package modelDown for the R language (R Core Team, 2018) that can help to eliminate these issues. modelDown automatically creates an easy to understand HTML website with model interpretations. You can automatically analyze, explore, and debug your predictive models in many ways with just a single command.

The modelDown package is designed in a similar way to the pkgdown (Wickham & Hesselberth, 2018) library. Both packages use templates to generate a static HTML website with detailed information. In the case of pkgdown, the information is about a selected package, while in the case of modelDown, it is about predictive models.

The modelDown package builds on the set of tools for model exploration implemented in the DALEX (Biecek, 2018) package. By creating explainer objects, DALEX provides a uniform interface for model agnostic exploration of models. Other R packages compatible with DALEX are also used to provide data, tables, and charts available on the website.

Example code

A basic example of using the package is as straightforward as below.

```
require("ranger")
require("breakDown")
require("DALEX")

devtools::install_github("MI2DataLab/modelDown")

# ranger
```

```
HR_ranger_model <- ranger(as.factor(left) ~ .,
                          data = HR_data, num.trees = 500,
                          classification = TRUE, probability = TRUE)

# glm
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")

# generating explainers
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)
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(explainer_ranger, list(explainer_glm1, explainer_glm2))
```
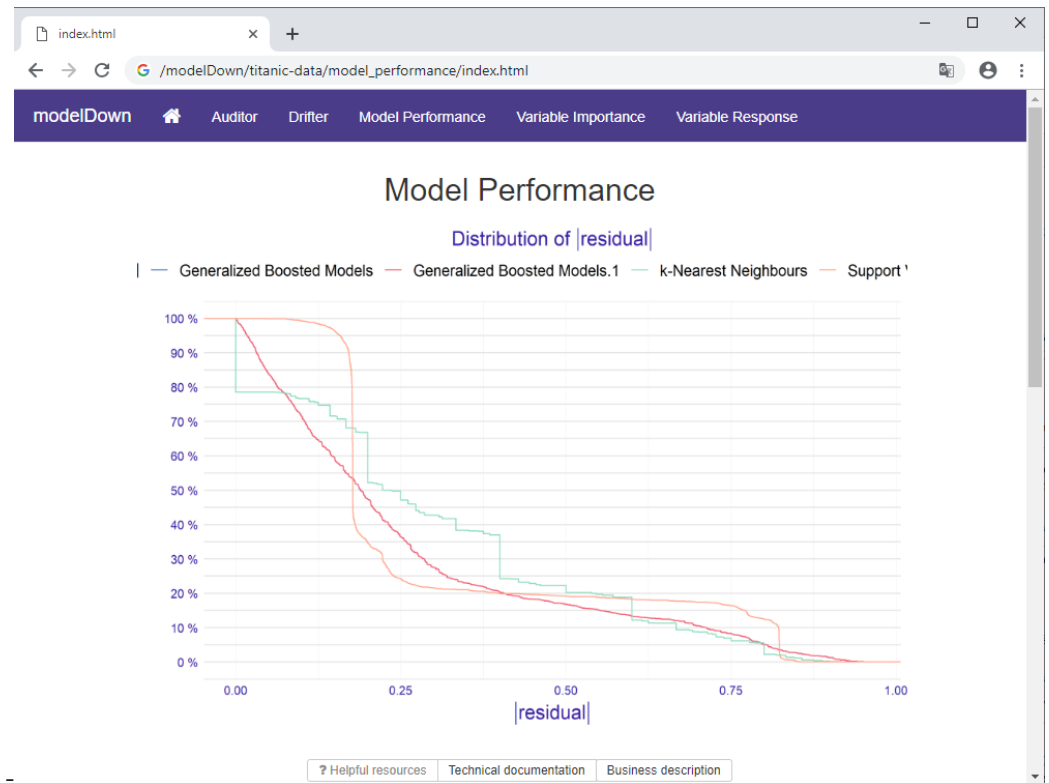
Available features

The modelDown package is highly modular. The primary outcome is a website with a set of tabs. Each tab explains a particular aspect of the model. Features are available across the tabs. By default, all the tabs are generated, and each of them can be parameterized.
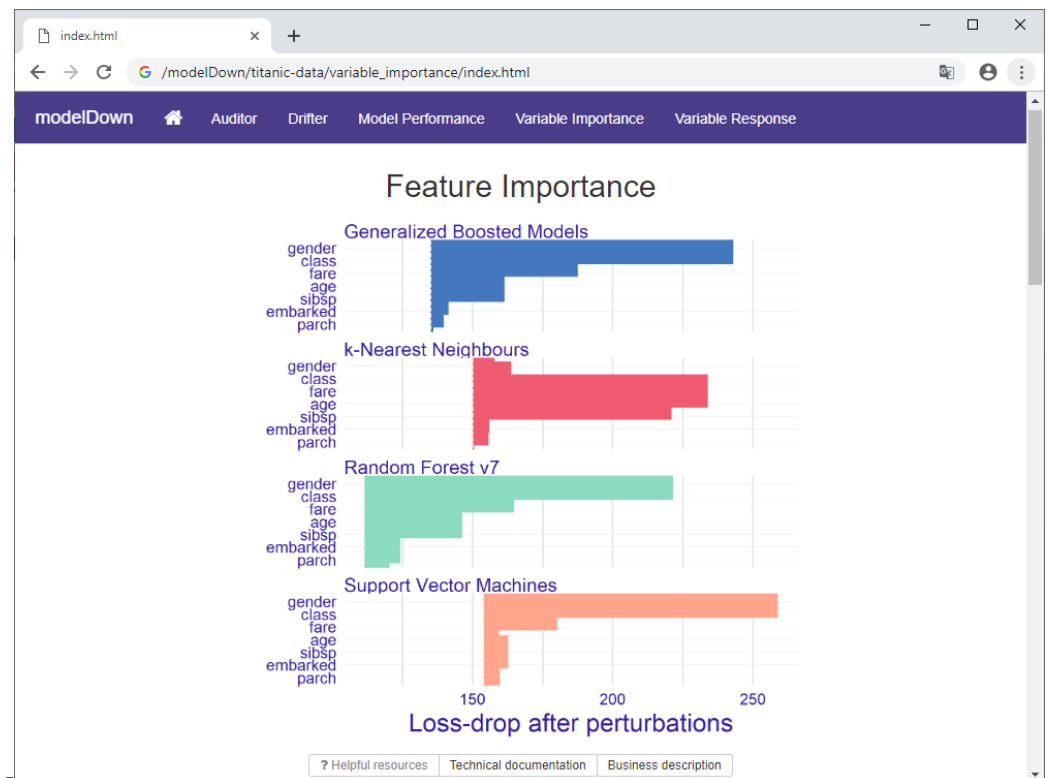
The 'Auditor' tab presents different ways of validating predictive models by analysis of residuals. All plots are generated by the auditor package (Gosiewska & Biecek, 2018). The plots may seem a bit complicated at first sight, so each plot has links with helpful explanations.

The 'Model performance' tab uses functions from DALEX package (Biecek, 2018). It shows contrastive comparisons of model predictive power by analyzing residuals. Read more about this type of model diagnostic in (Biecek & Burzykowski, 2019).

The 'Model performance' tab lets you compare different models. This applies for all tabs created with modelDown. This allows users not only to validate one model, but to pick the best model among many. Below we can see different distributions of residuals for the models.

The 'Feature importance' tab also uses functions from DALEX package. It shows how much impact each variable has when it comes to generating predictions. You can compare these model agnostic feature importance scores against domain knowledge in order to see if the model works in the expected way.

All examples presented in this paper are based on models generated with the Titanic dataset. In this example, three models that are considered picked gender as the most important variable. Only the k-Nearest Neighbors algorithm picked different features. This may suggest that this model does not give the best results for this data.

The 'Feature response' tab also uses functions from DALEX package. The main goal of this table is to present the link between average model response and the feature of interest. With the use of Partial Dependency Plots or Factor Merger Plots one can read how model predictions change with changing values of a selected single variable.

The last tab, 'drifter', uses functions from an R package with the same name (Biecek, 2019). You can test if there is concept drift or data drift in the data. You can also see how changes in the data distribution impact the model of interest.

All of this can be generated using one simple command, which is highly customizable. You can select different plot formats - either a raster png or vectorized (Wickham, Henry, Luciani, Decorde, & Lise, 2017), change each tab's individual settings and control visual aspects.

To boost reproducibility of models, all results from model exploration and explanation are saved in binary form with the archivist (Biecek & Kosinski, 2017) package. You can always look at them later, share models or explainers with collaborators, validate session info recorded when explainers were created, or reproduce selected explainers.

Summary

The package uses whisker html templates (de Jonge, 2013). Also, DALEX itself uses the extensible grammar of data visualization implemented in the ggplot2 package (Wickham, 2016). Because of that, as new packages compatible with DALEX explainers are developed, modelDown can be extended to use them. modelDown, with its simple usage and wide variety of information on the models, may be a powerful tool that helps you improve your model predictive power.

An example website generated with the modelDown package is at https://github.com/MI2DataLab/modelDown_example. It also contains a reproducible R code for model training, wrapping and transforming into an HTML website.

The modelDown is a part of DrWhy.AI (Warsaw University of Technology, 2018) collection of tools for automated training, exploration and maintenance for predictive models.

Acknowledgments

References

Biecek, P. (2018). DALEX: Explainers for complex predictive models in R. *Journal of Machine Learning Research*, *19*(84), 1–5. Retrieved from http://jmlr.org/papers/v19/18-416.html

Biecek, P. (2019). *drifter: Concept drift and concept shift detection for predictive models*. Retrieved from https://ModelOriented.github.io/drifter/

Biecek, P., & Burzykowski, T. (2019). *Predictive models: Visual exploration, explanation and debugging*. Retrieved from https://pbiecek.github.io/PM_VEE/

Biecek, P., & Kosinski, M. (2017). archivist: An R package for managing, recording and restoring data analysis results. *Journal of Statistical Software*, *82*(11), 1–28. doi:10.18637/jss.v082.i11

de Jonge, E. (2013). *whisker: Mustache for R, logicless templating*. Retrieved from https://CRAN.R-project.org/package=whisker

Gosiewska, A., & Biecek, P. (2018). *auditor: Model audit - verification, validation, and error analysis*. Retrieved from https://CRAN.R-project.org/package=auditor

R Core Team. (2018). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Warsaw University of Technology, M. @. (2018). DrWhy: Collection of tools for explainable AI (XAI). Retrieved April 14, 2019, from http://www.drwhy.ai/

Wickham, H. (2016). *Ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. Retrieved from http://ggplot2.org

Wickham, H., & Hesselberth, J. (2018). *pkgdown: Make static HTML documentation for a package*. Retrieved from https://CRAN.R-project.org/package=pkgdown

Wickham, H., Henry, L., Luciani, T. J., Decorde, M., & Lise, V. (2017). *svglite: An 'SVG' graphics device*. Retrieved from https://CRAN.R-project.org/package=svglite