

xai2cloud: the R package for automated cloud deployment of machine learning models with XAI REST API

Adam Rydelek¹ and Przemyslaw Biecek¹

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

DOI: [10.21105/joss.02344](https://doi.org/10.21105/joss.02344)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Pending Editor](#) ↗

Submitted: 15 June 2020

Published: 15 June 2020

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Introduction

Machine learning models are utilized in nearly every field of science and business. The growing number of excellent libraries for R and Python simplify and automate the process of building of ML models. It is becoming easier and easier to build a model locally, but sharing the model is still a challenge. Model deployment is an essential part of the life cycle of any machine learning model (Chapman et al., 1999). The goal of the `xai2cloud` library is to automate the last mile of the model life-cycle. The library works in a model-agnostic fashion, independently of the internal model structure. It transforms any predictive model into a RESTfull service automatically deployed in the cloud.

Created API allows not only to query the model on new points but also generates instance-level explanations of the reasons behind the value of a specific prediction. With the arising interest in eXplainable Artificial Intelligence (XAI) many tools for exploring machine learning models emerged. Among the most popular solutions SHAP (Lundberg & Lee, 2017), DALEX (Biecek, 2018), or the What-If (Wexler et al., 2019) should be mentioned. These libraries are very useful for building or debugging a model, but they are not integrated with the model deployment on production. The `xai2cloud` package treats explainability as an integral component of model functioning, offering the most popular XAI methods available with the same interface as model predictions.

There are also other libraries in R to explore the model, such as `modelStudio` (Baniecki & Biecek, 2019) (for precalculated statistics for serverless exploration), `archivist` (Biecek & Kosinski, 2017) (a database of meta-features of objects allowing for easier model searching and governance), `trackr` (Becker, Moore, & Lawrence, 2019) (focused on the reproducibility of models), `modelDown` (Romaszko, Tatarynowicz, Urbański, & Biecek, 2019) (automatically generated documentation for the model in HTML static page), `workflowr` (automation of model construction) and many others. But `xai2cloud` is the only package that allows you to interact with a live model and share models that are not frozen copies but working functions that can be combined with other cloud-based solutions.

The xai2cloud package

The `xai2cloud` is built on top of the `plumber` (Schloerke, 2020) and `analogsea` (Chamberlain, Wickham, & Chang, 2020) R packages. Those tools are combined with the tools for model exploration and explanation such as `ingredients` (Biecek, Baniecki, & Izdebski, 2020) and `iBreakDown` (Gosiewska & Biecek, 2019). It is easier to explore results visually than by raw numbers (Dulclercy & Joao, 2010) so the `xai2cloud` API generates also visual summaries.

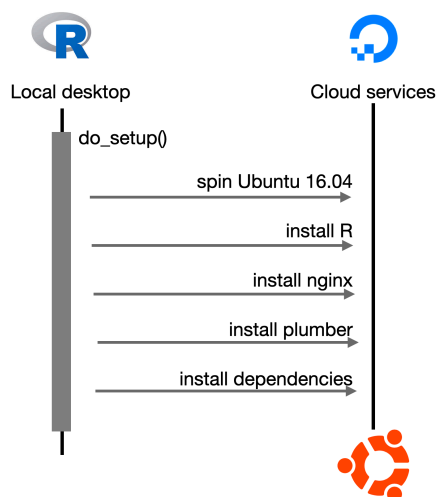
The `xai2cloud` package is developed for R (R Core Team, 2020) but the proposed architecture can also be replicated in other languages.

Despite cloud-based solutions being commonly associated with difficult setup and high entry threshold, `xai2cloud` is created with simplicity in mind. Configuration of the cloud environment is a one-time activity that is thoroughly explained at the package's [website](#).

The solution used for cloud computing is [DigitalOcean](#). The platform is affordable and presents an intuitive website interface to keep track of all currently running droplets. Droplet is a name given by DigitalOcean for their Linux servers running on top of cloud-based hardware. Configuration of a new server is done entirely through R using the `setup` feature of the `xai2cloud` package which creates a new droplet with R version 3.6.3 and all the required packages already installed.

After the initial setup, the deployment process is instant and intuitive. The package can deploy any model wrapped into an explainer with one R function. Explainers are adapters available for predictive models created using the DALEX package. Deployed model is subsequently available on the server with XAI features enabling thorough exploration.

1. Create a droplet



2. Add a model to the droplet

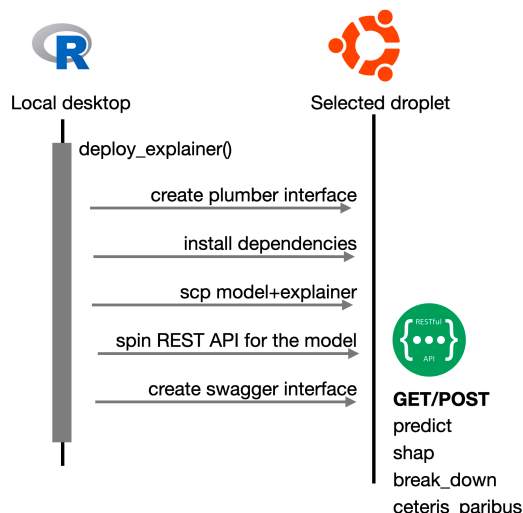


Figure 1: Architecture of two main functions in `xai2cloud` package. `do_setup` configures a new cloud environment, `deploy_explainer` adds model to the configured droplet.

Deployment example

The `xai2cloud` R package can be installed from GitHub using devtools with:

```
devtools::install_github('ModelOriented/xai2cloud')
```

Droplet setup is needed to start working with model deployment. The quick setup guide is available at the package's [website](#). Assuming the configuration has been completed, the deployment of any predictive models is rapid and effortless. Below is a code sample attached which produces an [example](#).

- 1. Create a model

```
library("ranger")
library("DALEX")
model <- ranger(survived~., data = titanic_imputed)
```

- 2. Wrap it into a DALEX explainer

```
exp_name <- explain(model,
  data = titanic_imputed[,-8],
  y = titanic_imputed$survived)
```

- 3. Check droplet's ID

```
library("analogsea")
my_droplets <- droplets()
```

- 4. Choose the correct droplets name - 'xai2cloudExamples' in this case

```
specific_droplet <- my_droplets$xai2cloudExamples
droplet_id <- specific_droplet$id
```

- 5. Deploy the explainer to the selected droplet

```
library("xai2cloud")
deploy_explainer(exp_name, model_package = 'ranger',
  droplet=droplet_id, port=8070, title="JOSS Example")
```

Features overview

The deployed explainer is active as an application compliant to the representational state transfer architecture operating on the droplet's server (Fielding & Taylor, 2000). It hosts six POST and GET hooks in total enabling the user to explore the model and its predictions. The features provided include not only a basic prediction of inputted data but also local model explanations:

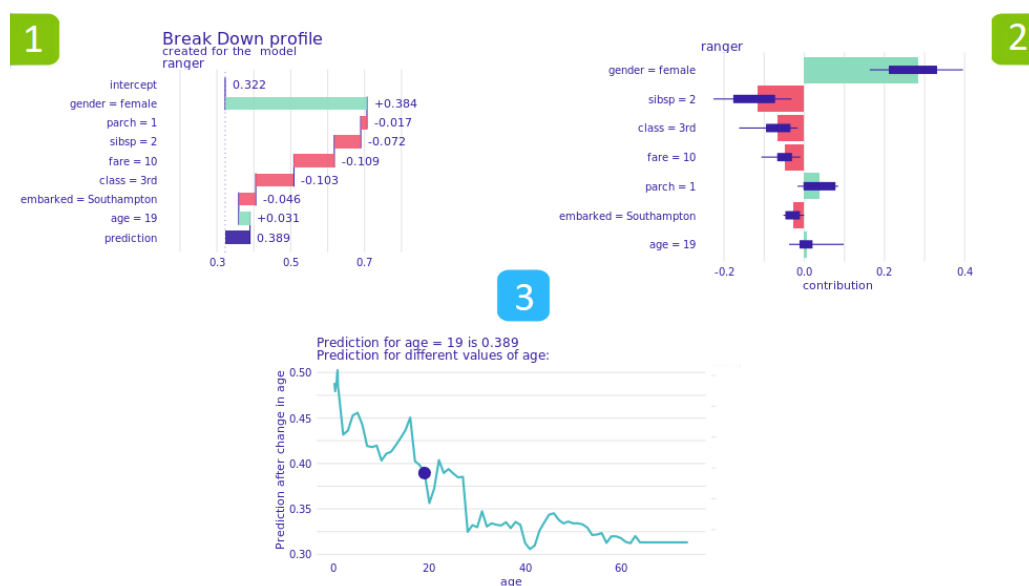


Figure 2: An example of Break Down (1), SHAP (2) and Ceteris Paribus (3) plots generated by the API for a Titanic ranger model

- The most basic feature provided by the API is the prediction hook. It returns an instant model's answer for any given observation.
- SHAP (Lundberg & Lee, 2017) plots offer another way of explaining the output of machine learning models by using classic Shapley values from game theory. The hook creates a plot enabling the user to take a look at each variable's contribution with a broader perspective.
- Break Down (Gosiewska & Biecek, 2019) plots present the contribution of each variable to the prediction. There are two hooks associated with this feature, one creates the plot and returns it as an image, the other one is a description consisting of crucial information that can be gathered from the Break-Down plot and is returned as a string.
- Ceteris Paribus plots from the `ingredients` package (Biecek, 2018) present model responses around a single point in the feature space. It enables the user to view possible changes in model predictions allowing for changes in a single variable while keeping all other features constant. The results are also available as both a chart and a text description summing up crucial conclusions.

Usage



Figure 3: Basic illustration of Swagger UI created for an example Titanic model using xai2cloud

The RESTful API created by xai2cloud can be utilized in various ways. There is a built-in Swagger User Interface (Figure 3) available at the droplet's server. Swagger is a set of open-source tools that enhance REST application building process. One of the tools available is Swagger UI which presents interactive API documentation. Therefore it can be used as a basic standalone solution for sharing predictive models results. It can be then easily used in much more complex projects. The xai2cloud package can deploy multiple models on one droplet without any further configuration, allowing users to create applications gathering model data from the whole organization. The package is built in a way to encourage using it as an external data source for projects concerning predictive modeling.

Summary

The `xai2cloud` package offers a new perspective on sharing the predictive models. It enables cloud deployment of artificial intelligence solutions while reducing the setup technological entry threshold to the minimum. Enabling users to view and understand the predictive model is the end goal of explainable artificial intelligence. This objective can be achieved using built-in local explanation features and plots from the `xai2cloud` package. The package is in constant development and the features list is going to expand over time. Updates and more examples can be found on [GitHub ModelOriented/xai2cloud repository](https://github.com/ModelOriented/xai2cloud) of the project and the [modeloriented.github.io/xai2cloud website](https://modeloriented.github.io/xai2cloud).

Acknowledgements

Work on this package was financially supported by the NCBR Grant POIR.01.01.01-00-0328/17.

References

- Baniecki, H., & Biecek, P. (2019). `modelStudio`: Interactive studio with explanations for ML predictive models. *Journal of Open Source Software*, 4(43), 1798. doi:[10.21105/joss.01798](https://doi.org/10.21105/joss.01798)
- Becker, G., Moore, S. E., & Lawrence, M. (2019). Trackr: A framework for enhancing discoverability and reproducibility of data visualizations and other artifacts in R. *Journal of Computational and Graphical Statistics*, 28(3), 644–658. doi:[10.1080/10618600.2019.1585259](https://doi.org/10.1080/10618600.2019.1585259)
- Biecek, P. (2018). DALEX: Explainers for complex predictive models in R. *Journal of Machine Learning Research*, 19.
- Biecek, P., Baniecki, H., & Izdebski, A. (2020). *Ingredients: Effects and importances of model ingredients*. Retrieved from <https://CRAN.R-project.org/package=ingredients>
- 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](https://doi.org/10.18637/jss.v082.i11)
- Chamberlain, S., Wickham, H., & Chang, W. (2020). *Analogsea: Interface to 'digital ocean'*. Retrieved from <https://CRAN.R-project.org/package=analogsea>
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (1999). *The crisp-dm 1.0 step-by-step data mining guide*. Retrieved from <ftp://ftp.software.ibm.com/software/analytics/spss/support/Modeler/Documentation/14/UserManual/CRISP-DM.pdf>
- Dulclerc, A., & Joao, T. (2010). Introduction of human perception in visualization. *International Journal of Imaging*, 4.
- Fielding, R. T., & Taylor, R. N. (2000). *Architectural styles and the design of network-based software architectures* (PhD thesis). University of California, Irvine.
- Gosiewska, A., & Biecek, P. (2019). Do not trust additive explanations. Retrieved from <https://arxiv.org/abs/1903.11420>
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S.

- Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems 30* (pp. 4765–4774). Curran Associates, Inc. Retrieved from <http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf>
- R Core Team. (2020). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from <https://www.R-project.org/>
- Romaszko, K., Tatarynowicz, M., Urbański, M., & Biecek, P. (2019). ModelDown: Automated website generator with interpretable documentation for predictive machine learning models. *Journal of Open Source Software*, 4(38), 1444. doi:[10.21105/joss.01444](https://doi.org/10.21105/joss.01444)
- Schloerke, B. (2020). *plumber: An API Generator for R*. Retrieved from <https://github.com/rstudio/plumber>
- Wexler, J., Pushkarna, M., Bolukbasi, T., Wattenberg, M., Viegas, F., & Wilson, J. (2019). The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, PP, 1–1. doi:[10.1109/TVCG.2019.2934619](https://doi.org/10.1109/TVCG.2019.2934619)