The Journal of Open Source Software

mlr3: A modern object-oriented machine learning framework in R

Michel Lang[1, 2], **Martin Binder**[2], **Jakob Richter**[1], **Patrick Schratz**[2], **Florian Pfisterer**[2], **Stefan Coors**[2], **Quay Au**[2], **Giuseppe Casalicchio**[2], **Lars Kotthoff**[3], **and Bernd Bischl**[2]

1 TU Dortmund University **2** LMU Munich **3** University of Wyoming

DOI: 10.21105/joss.01903

Software

- Review
- Repository
- Archive

Editor: Yuan Tang

Reviewers:

- @nhejazi
- @osorensen

Submitted: 15 November 2019
Published: 11 December 2019

Summary

The R (R Core Team, 2019) package mlr3 and its associated ecosystem of extension packages implements a powerful, object-oriented and extensible framework for machine learning (ML) in R. It provides a unified interface to many learning algorithms available on CRAN, augmenting them with model-agnostic general-purpose functionality that is needed in every ML project, for example train-test-evaluation, resampling, preprocessing, hyperparameter tuning, nested resampling, and visualization of results from ML experiments. The package is a complete reimplementation of the mlr (Bischl et al., 2016) package that leverages many years of experience and learned best practices to provide a state-of-the-art system that is powerful, flexible, extensible, and maintainable. We target both **practitioners** who want to quickly apply ML algorithms to their problems and **researchers** who want to implement, benchmark, and compare their new methods in a structured environment. mlr3 is suitable for short scripts that test an idea, for complex multi-stage experiments with advanced functionality that use a broad range of ML functionality, as a foundation to implement new ML (meta-)algorithms (for example AutoML systems), and everything in between. Functional correctness is ensured through extensive unit and integration tests.

Several other general-purpose ML toolboxes exist for different programing languages. The most widely used ones are scikit-learn (Pedregosa et al., 2011) for Python , Weka (Hall et al., 2009) for Java, and mlj (Blaom, Kiraly, Lienart, & Vollmer, 2019) for Julia. The most important toolboxes for R are mlr, caret (Kuhn, 2008) and tidymodels (Kuhn & Wickham, 2019).

Lessons Learned from 6 Years of Machine Learning in R

The predecessor package mlr was first released to CRAN in 2013, with the core design and architecture dating back much further. As with most software, more code was added over time to integrate more ML algorithms, more approaches for feature selection or hyperparameter tuning, more methods to analyze trained models, and many other things. With each addition, the code base became larger and more difficult to test and maintain, in particular as changes in the dozens of packages that we integrated with mlr would break our code and prevent releases. Installing the package with all dependencies and a complete build with all tests would take hours – we had arrived at a point where adding **any** new functionality became a major undertaking. Further, some of the architectural and design decisions made it essentially impossible to support new cross-cutting functionality, for example ML pipelines, or using new R packages for better performance.

mlr3 takes these lessons learned to heart and now follows these design principles:

- Be modular and light on dependencies. The core `mlr3` package provides only the basic building blocks of ML: tasks, a few learners, resampling methods, and performance measures. Everything else can be installed and loaded separately through additional packages in the `mlr3` ecosystem, for example support for other kinds of data, methods for tuning hyperparameters, or integrations for additional ML packages.
- Leverage modern `R` packages, especially `data.table` for fast and efficient computations on rectangular data.
- Embrace `R6` for a clean object-oriented design, object state changes, and reference semantics.
- Defensive programming and type safety. All user input is checked with `checkmate` (Lang, 2017). Return types are documented and automatic type casting for "simplification" is avoided.

In addition, we simplified the API considerably by unifying container and result classes. Many result objects are now tabular by mixing `data.table`'s list-column feature with `R6` objects, which also allows for easy and efficient selection and "split-apply-combine" type operations.

Ecosystem

In addition to the main `mlr3` package, `mlr3learners` provides integrations to a careful selection of the most important ML algorithms and packages in `R`. Complex ML workflows (using directed acyclic graphs) that can incorporate preprocessing, (stacking) ensembles, alternative-branch execution, and much more can be built with the `mlr3pipelines` package. Funtionality for hyperparameter tuning and nested resampling of learners and complex pipelines is provided by the `mlr3tuning` package. `mlr3filters` integrates many feature filtering techniques and `mlr3db` allows direct use of databases as data sources for out-of-memory data. We are planning and working on many more packages; for example for Bayesian optimization, Hyperband, probabilistic regression, survival analysis, and spatial and temporal data. A complete list of existing and planned extension packages can be found on the mlr3 wiki.

`mlr3` and its ecosystem are documented in numerous manual pages and a comprehensive book (work in progress). All packages are licensed under GNU Lesser General Public License (LGPL-3).

Acknowledgments

This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibilities for its content.

This work was partly supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876, A3.

References

Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., Casalicchio, G., et al. (2016). Mlr: Machine learning in r. *Journal of Machine Learning Research*, *17*(170), 1–5. Retrieved from http://jmlr.org/papers/v17/15-066.html

Blaom, A., Kiraly, F., Lienart, T., & Vollmer, S. (2019). *Alan-turing-institute/mlj.jl: V0.5.3*. Zenodo. doi:10.5281/zenodo.3541506

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *ACM SIGKDD explorations newsletter*, *11*(1), 10–18. doi:10.1145/1656274.1656278

Kuhn, M. (2008). Building predictive models in r using the caret package. *Journal of Statistical Software, Articles*, *28*(5), 1–26. doi:10.18637/jss.v028.i05

Kuhn, M., & Wickham, H. (2019). *Tidymodels: Easily install and load the 'tidymodels' packages*. Retrieved from https://CRAN.R-project.org/package=tidymodels

Lang, M. (2017). checkmate: Fast Argument Checks for Defensive R Programming. *The R Journal*, *9*(1), 437–445. doi:10.32614/RJ-2017-028

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. Retrieved from http://jmlr.org/papers/v12/pedregosa11a.html

R Core Team. (2019). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/