The Journal of Open Source Software

mlr3extralearners: Expanding the mlr3 Ecosystem with Community-Driven Learner Integration

Sebastian Fischer [2,3], **John Zobolas** [4], **Raphael Sonabend** [15], **Marc Becker** [2,3], **Michel Lang** [1], **Martin Binder**[2,3], **Lennart Schneider** [2,3], **Lukas Burk** [2,3,5,6], **Patrick Schratz** [18], **Byron C. Jaeger** [13], **Stephen A Lauer** [7], **Lorenz A. Kapsner** [8], **Maximilian Mücke** [2], **Zezhi Wang** [9], **Damir Pulatov** [14], **Keenan Ganz** [10], **Henri Funk** [3,11,12], **Liana Harutyunyan**[17], **Pierre Camilleri** [16], **Philipp Kopper** [3], **Andreas Bender** [2,3], **Baisu Zhou** [19], **Niko German** [2], **Lona Koers** [2], **Anna Nazarova**[2], and **Bernd Bischl** [2,3]

1 TU Dortmund University, Germany **2** Department of Statistics, LMU Munich, Germany **3** Munich Center for Machine Learning (MCML), Germany **4** Department of Cancer Genetics, Institute for Cancer Research, Oslo University Hospital, Norway **5** Leibniz Institute for Prevention Research and Epidemiology (BIPS), Bremen, Germany **6** Faculty of Mathematics and Computer Science, University of Bremen, Germany **7** Certilytics, Inc., 9200 Shelbyville Rd, Louisville, KY, 40222, USA **8** Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany **9** Department of Statistics and Finance/International Institute of Finance, School of Management, University of Science and Technology of China, Hefei, Anhui, China **10** School of Environmental and Forest Sciences, University of Washington, Seattle **11** Department of Geography, LMU Munich, Germany **12** Statistical Consulting Unit StaBLab, LMU Munich, Germany **13** Wake Forest University School of Medicine, Department of Biostatistics and Data Science, Division of Public Health Sciences Winston-Salem, North Carolina **14** University of North Carolina Wilmington **15** OSPO Now **16** multi, 8 passage Brûlon, 75012 PARIS, France **17** ServiceTitan, Inc., Glendale, California **18** devXY GmbH **19** Department of Computer Science, University of Tübingen

DOI: 10.21105/joss.08331

Software

- Review
- Repository
- Archive

Editor: Richard Littauer

Reviewers:

- @ritika-giri
- @kelly-sovacool

Submitted: 24 March 2025
Published: 17 November 2025

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

Summary

The mlr3 ecosystem is a versatile toolbox for machine learning (ML) in R (R Core Team, 2019) that is targeted towards both practitioners and researchers (Bischl et al., 2024). The core mlr3 package (Lang et al., 2019) defines the standardized interface for ML, but its goal is not to implement algorithms. This is, e.g., done by the mlr3learners extension (Lang, Au, et al., 2024) that connects 21 stable learning algorithms from various R packages to the mlr3 ecosystem that serve as a good starting point for many ML tasks. In addition, mlr3extralearners is a *community-driven* package that integrates many more methods. The package currently wraps **147 different ML algorithms** from many different R packages, for tasks such as classification, regression, and survival analysis. This enables users to seamlessly access and utilize these learners directly within their workflows. One of the strengths of mlr3 is the design and implementation of large-scale benchmark experiments. For example, datasets for such experiments can be easily obtained from the OpenML[1] repository (Vanschoren et al., 2014) via the mlr3oml package (Lang & Fischer, 2024). Furthermore, strong support for parallelization, including execution on high-performance computing clusters via batchtools (Lang et al., 2017) and its mlr3 integration mlr3batchmark (Becker & Lang, 2024), is available and well documented (Fischer et al., 2024). In combination, these tools allow for large-scale empirical investigations, which has, for example, been used to collect and analyze data about hyperparameter landscapes of ML algorithms (Binder et al., 2020). An overview of all mlr3

[1]https://openml.org

learners, including those introduced through mlr3extralearners, is available on the mlr3 website[2].

Beyond accessibility, mlr3extralearners also allows mlr3 users to easily connect their own algorithms to the interface. This **enriches each learner with extensive metadata** about its hyperparameter space, prediction types, and other key attributes. Furthermore, mlr3extralearners includes robust mechanisms for **quality assurance**, such as regular automated sanity checks and verification tests that ensure learner parameters are consistent and up-to-date with the latest versions of their underlying R packages. In order to allow the integration of learners that are not available on CRAN, the package is hosted on the mlr R-universe[3]. By providing a standardized interface and comprehensive metadata for each learner, mlr3extralearners enhances the FAIRness (findability, accessibility, interoperability, and reusability) of ML algorithms within the R ecosystem (Wilkinson et al., 2016).

Statement of Need

ML often requires practitioners to navigate a diverse array of modeling problems, each with unique demands such as predictive performance, prediction speed, interpretability, or compatibility with specific data types and tasks. To address this challenge, packages like mlr3's predecessor mlr (Bischl et al., 2016), caret (Kuhn, 2008), and more recently parsnip (Kuhn & Vaughan, 2024) from the tidymodels ecosystem (Kuhn & Wickham, 2020) were designed to provide unified interfaces for simplifying model experimentation. For instance, parsnip provides a clean and consistent way to define models, enabling users to experiment with different algorithms without dealing with the nuances of underlying package syntax. Similarly, the mlr3 ecosystem aims to streamline model selection and experimentation, making it a versatile toolbox for ML in R.

Within this ecosystem, mlr3extralearners plays a crucial role by providing a comprehensive collection of external ML algorithms integrated into the mlr3 framework. This ensures that users can access a wide variety of learners to meet their needs and choose the most appropriate algorithm for their particular problem. While connecting new learners to mlr3 is straightforward and can be done on a per-need basis, integrating them into mlr3extralearners benefits the broader community by avoiding redundant effort and ensuring accessibility for all users. Additionally, contributions to mlr3extralearners are reviewed by the package maintainers, providing a layer of quality assurance. This review process ensures that integrated learners work as expected and adhere to the high standards of the mlr3 ecosystem.

Beyond its utility for users, mlr3extralearners also offers significant advantages for developers of ML packages. By integrating a new algorithm into the mlr3 ecosystem, developers can immediately make their methods accessible to a wider audience. This integration facilitates seamless tuning (Becker et al., 2024) and preprocessing (Binder et al., 2021) through the broader mlr3 framework, enhancing the usability and impact of their work.

Features

The core functionality of mlr3extralearners is to integrate new learners into the mlr3 ecosystem, allowing users to access a wide array of learning algorithms through a unified syntax and standardized interface. However, the advantages of mlr3extralearners go well beyond simple integration.

Metadata

One core feature of the mlr3 ecosystem is that it annotates learners with extensive metadata.

[2]https://mlr-org.com/learners.html
[3]https://mlr-org.r-universe.dev

- **Hyperparameter management**: The hyperparameter spaces of learners are defined using `ParamSet` objects from the `paradox` package (Lang, Bischl, et al., 2024). Each hyperparameter is explicitly typed, with annotations for feasible values. This ensures valid configurations and simplifies tasks like hyperparameter tuning.
- **Task and prediction types**: Learners are categorized with respect to their task type (e.g. as classification, regression or survival analysis (Sonabend et al., 2021)) and prediction types (e.g. probabilities or response predictions). This allows users to easily identify suitable learners for their specific modeling tasks.
- **Standardized properties**: Learners are annotated with detailed attributes, including the types of features they can process and their support for functionalities such as feature selection, importance scoring, handling missing values, or monitoring performance on a separate validation set during training among others. This allows users to have a clear understanding of a learner's capabilities and limitations and assess if it aligns with the specific requirements of their workflows, reducing trial-and-error and streamlining the modeling process.

Functional Correctness

Integrating learners from diverse `R` packages poses challenges, on the one hand because changes in upstream APIs need to be reflected in `mlr3extralearners` and on the other hand because we want to ensure a high level of quality of algorithms connected to `mlr3`. `mlr3extralearners` addresses both points through automated checks:

- **Interface consistency**: The package regularly verifies that each learner adheres to the expected interface of the latest released version of its upstream function. When new parameters are introduced or existing ones changed or removed, the tests fail until the parameter sets are updated accordingly.
- **Automated testing**: In general, writing unit tests for ML algorithms is challenging, because of edge-cases, numeric errors, and the fact that the input to these algorithms can be arbitrary datasets. `mlr3extralearners` is aimed at addressing these challenges, and performs regular automated tests on all learners. These tests include sanity checks that, e.g., verify that the learners produce sensible predictions for simple randomly generated datasets. Furthermore, the tests also validate the learners' metadata annotations, such as whether a learner can actually handle missing values or is able to produce importance scores. In the past, these tests have detected bugs in some upstream packages and we have subsequently notified their authors.

Simplified Integration of New Learners

To streamline the addition of new learners, `mlr3extralearners` provides robust support tools:

- **Code templates**: Predefined templates are available for both the learner implementation and associated test files. Contributors can utilize these templates through an `R` function that accepts learner metadata and generates new `R` code files based on the templates. This approach pre-fills as much information as possible, minimizing the input required from the contributor. Note that these templates can also be used when learners are only used locally for specific projects and not contributed to `mlr3extralearners`.
- **Guides and resources**: The package website[4] contains an extensive tutorial, as well as a curated list of common issues encountered during learner integration, making the process accessible for contributors of all experience levels. Additionally, every integrated learner includes a simple example of usage in the documentation, ensuring that users can quickly understand how to utilize the learner effectively within the `mlr3` ecosystem.

[4]https://mlr3extralearners.mlr-org.com

Community Impact and Future Directions

mlr3extralearners is a direct result of the contributions from a diverse community of authors and developers. The authors of this paper themselves have been actively involved in integrating learners, providing quality assurance, and maintaining the package's infrastructure. Their contributions, such as the addition of learners for specialized tasks like survival analysis and high-dimensional data, highlight the impact that thoughtful integration has on the mlr3 ecosystem. This ongoing effort illustrates the transformative potential of **community-driven development**, ensuring that mlr3extralearners continues to grow as a dynamic and inclusive repository for ML algorithms.

Future work will also focus on expanding the ecosystem with more deep learning methods through mlr3torch (Fischer & Binder, 2025), which aims to seamlessly integrate deep learning models and neural network architectures within the mlr3 framework.

Acknowledgements

Sebastian Fischer is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 460135501 (NFDI project MaRDI). John Zobolas received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101016851, project PANCAIM.

References

Becker, M., & Lang, M. (2024). *mlr3batchmark: Batch experiments for 'mlr3'*. https://doi.org/10.32614/cran.package.mlr3batchmark

Becker, M., Lang, M., Richter, J., Bischl, B., & Schalk, D. (2024). *mlr3tuning: Hyperparameter optimization for 'mlr3'*. https://doi.org/10.32614/cran.package.mlr3tuning

Binder, M., Pfisterer, F., & Bischl, B. (2020). Collecting empirical data about hyperparameters for data driven AutoML. *Democratizing Machine Learning Contributions in AutoML and Fairness*, 93.

Binder, M., Pfisterer, F., Lang, M., Schneider, L., Kotthoff, L., & Bischl, B. (2021). mlr3pipelines - Flexible Machine Learning Pipelines in R. *Journal of Machine Learning Research*, *22*(184), 1–7. http://jmlr.org/papers/v22/21-0281.html

Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., Casalicchio, G., & Jones, Z. M. (2016). mlr: Machine Learning in R. *Journal of Machine Learning Research*, *17*(170), 1–5.

Bischl, B., Sonabend, R., Kotthoff, L., & Lang, M. (Eds.). (2024). *Applied Machine Learning Using mlr3 in R*. CRC Press. ISBN: 9781032507545

Fischer, S., & Binder, M. (2025). *mlr3torch: Deep Learning with 'mlr3'*. https://doi.org/10.32614/cran.package.mlr3torch

Fischer, S., Lang, M., & Becker, M. (2024). Large-Scale Benchmarking. In B. Bischl, R. Sonabend, L. Kotthoff, & M. Lang (Eds.), *Applied machine learning using mlr3 in R*. CRC Press. https://mlr3book.mlr-org.com/large-scale_benchmarking.html

Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software, Articles*, *28*(5), 1–26. https://doi.org/10.18637/jss.v028.i05

Kuhn, M., & Vaughan, D. (2024). *Parsnip: A common API to modeling and analysis functions*. https://doi.org/10.32614/cran.package.parsnip

Kuhn, M., & Wickham, H. (2020). *Tidymodels: A collection of packages for modeling and machine learning using tidyverse principles.* https://www.tidymodels.org

Lang, M., Au, Q., Coors, S., Schratz, P., & Becker, M. (2024). *mlr3learners: Recommended learners for 'mlr3'.* https://doi.org/10.32614/cran.package.mlr3learners

Lang, M., Binder, M., Richter, J., Schratz, P., Pfisterer, F., Coors, S., Au, Q., Casalicchio, G., Kotthoff, L., & Bischl, B. (2019). mlr3: A Modern Object-Oriented Machine Learning Framework in R. *Journal of Open Source Software*, *4*(44), 1903. https://doi.org/10.21105/JOSS.01903

Lang, M., Bischl, B., Richter, J., Sun, X., & Binder, M. (2024). *paradox: Define and Work with Parameter Spaces for Complex Algorithms*. https://doi.org/10.32614/cran.package.paradox

Lang, M., Bischl, B., & Surmann, D. (2017). batchtools: Tools for R to work on batch systems. *Journal of Open Source Software*, *2*(10), 135. https://doi.org/10.21105/joss.00135

Lang, M., & Fischer, S. (2024). *mlr3oml: Connector between 'mlr3' and 'OpenML'*. https://CRAN.R-project.org/package=mlr3oml

R Core Team. (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. https://doi.org/10.32614/r.manuals

Sonabend, R., Király, F. J., Bender, A., Bischl, B., & Lang, M. (2021). mlr3proba: an R package for machine learning in survival analysis. *Bioinformatics*, *37*(17), 2789–2791. https://doi.org/10.1093/BIOINFORMATICS/BTAB039

Vanschoren, J., Van Rijn, J. N., Bischl, B., & Torgo, L. (2014). OpenML: Networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, *15*(2), 49–60.

Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., Silva Santos, L. B. da, Bourne, P. E., & others. (2016). The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, *3*(1), 1–9. https://doi.org/10.1038/s41597-019-0009-6