imodels: a python package for fitting interpretable models

Chandan Singh[*1], Keyan Nasseri[†1], and Bin Yu[1, 2]

1 EECS Department, University of California, Berkeley **2** Statistics Department, University of California, Berkeley

Summary

`imodels` is a Python package for concise, transparent, and accurate predictive modeling. It provides users a simple interface for fitting and using state-of-the-art interpretable models, all compatible with scikit-learn (Pedregosa et al., 2011). These models can often replace black-box models while improving interpretability and computational efficiency, all without sacrificing predictive accuracy. In addition, the package provides a framework for developing custom tools and rule-based models for interpretability.

Statement of need

Recent advancements in machine learning have led to increasingly complex predictive models, often at the cost of interpretability. There is often a need for models which are inherently interpretable (Murdoch et al., 2019; Rudin, 2018), particularly in high-stakes applications such as medicine, biology, and political science. In these cases, interpretability can ensure that models behave reasonably, identify when models will make errors, and make the models more trusted by domain experts. Moreover, interpretable models tend to be much more computationally efficient then larger black-box models.

Despite the development of many methods for fitting interpretable models (Molnar, 2020), implementations for such models are often difficult to find, use, and compare to one another. `imodels` aims to fill this gap by providing a simple unified interface and implementation for many state-of-the-art interpretable modeling techniques.

Features

Interpretable models can take various forms. section shows four possible forms a model in the `imodels` package can take. Each form constrains the final model in order to make it interpretable, but there are different methods for fitting the model which differ in their biases and computational costs. The `imodels` package contains implementations of various such methods and also useful functions for recombining and extending them.

Rule sets consist of a set of rules which each act independently. There are different strategies for deriving a rule set, such as Skope-rules (Skope Collaboration, 2021) or Rulefit (Friedman et al., 2008). Rule lists are composed of a set of rules which act in sequence, and include models such as Bayesian rule lists (Letham et al., 2015) or the oneR algorithm (Holte, 1993). Rule trees are similar to rule lists, but allow branching after rules. This includes models such as CART decision trees (Breiman et al., 1984). Algebraic models take a final form of simple algebraic expressions, such as supersparse linear integer models (Ustun & Rudin, 2016).

---

[*]Equal contribution
[†]Equal contribution

Examples of different supported model forms. The bottom of each box shows predictions of the corresponding model as a function of X_1 and X_2.

Acknowledgements

References

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Friedman, J. H., Popescu, B. E., & others. (2008). Predictive learning via rule ensembles. *Annals of Applied Statistics*, *2*(3), 916–954.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, *11*(1), 63–90.

Letham, B., Rudin, C., McCormick, T. H., Madigan, D., & others. (2015). Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *Annals of Applied Statistics*, *9*(3), 1350–1371.

Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, *116*(44), 22071–22080.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & others. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Rudin, C. (2018). Please stop explaining black box models for high stakes decisions. *arXiv Preprint arXiv:1811.10154*, *1*.

Skope Collaboration. (2021). Skope-rules. In *GitHub repository*. GitHub. https://github.com/scikit-learn-contrib/skope-rules

Ustun, B., & Rudin, C. (2016). Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, *102*(3), 349–391.