Practical Techniques for Interpreting Machine Learning Models: Introductory Open Source Examples Using Python, H2O, and XGBoost

Patrick Hall, Navdeep Gill, Mark Chan
H2O.ai, Mountain View, CA

February 3, 2018

1 Description

This series of Jupyter notebooks uses open source tools such as Python, H2O, XGBoost, GraphViz, Pandas, and NumPy to outline practical explanatory techniques for machine learning models and results. The notebooks cover the following modeling and explanatory techniques, along with practical variants and concise visualizations thereof.

- **Monotonically constrained GBMs, partial dependence, and ICE**:

  – *Monotonic Gradient Boosting using XGBoost*
  – *Partial Dependence and ICE Plots*

  These notebooks use monotonicity constraints to train an explainable, and potentially regulator-approvable GBM model, by ensuring predictions only increase or only decrease for any change in a given input variable. Partial dependence plots and ICE plots are then used to analyze and investigate the global and local mechanisms of the monotonic GBM and verify its monotonic behavior.[1], [2]

- **Decision tree surrogate models, variable importance, and LOCO local feature importance**:

  – *Decision Tree Surrogates*
  – *Local Feature Importance and Reason Codes using LOCO*

  These notebooks use a decision tree surrogate model trained on the original inputs and predictions of a complex GBM and the variable importance and interactions displayed in the surrogate model to create an overall, approximate flowchart of the complex GBM's predictions. The global variable importance of the GBM can be compared to the surrogate model, to domain expertise, and to reasonable expectations to evaluate the trustworthiness of the GBM model and the generated explanations. To enhance local understanding of the complex GBM's behavior and to enhance the accountability of its predictions, a variant of the LOCO technique is then used to calculate the local contribution each input variable makes toward each model prediction. Local contributions are ranked to generate reason codes that describe, in plain English, the GBM's decision process for every prediction.[3], [1], [4]

- **LIME**:

  – *LIME*

  This notebook presents an educational, step-by-step implementation of the popular LIME technique and introduces a straightforward method of creating local samples for LIME that can be more appropriate for real-time scoring of new data in production applications. Once local samples have been generated, LIME will be used to understand local trends in the complex model's predictions. LIME will also be used to calculate the local contribution of each input variable toward each model prediction, and these contributions can be sorted to create reason codes – i.e. plain English explanations of every model prediction. LIME explanations will be validated to enhance trust in generated explanations using the local model's R^2 statistic and a ranked

predictions plot.[5]

- **Sensitivity Analysis**:

  – *Sensitivity Analysis*

  This notebook introduces sensitivity analysis, perhaps the most important validation technique for increasing trust in machine learning model predictions. Because machine learning model predictions can vary drastically for small changes in input variable values, especially outside of training input domains, it can be important to explicitly test model behavior on unseen data. This notebook investigates whether GBM model behavior and outputs remain stable when input data is intentionally perturbed.

2 Instructions and Dependencies

Docker instructions and a Dockerfile are available for Mac, Linux, and Windows 10 users to build an environment with all necessary dependencies for the notebook series. Manual installation instructions are also available.

3 Additional Code Resources

This series is an evolving body of work, and there are a few techniques that routinely come up in discussions about important explanatory techniques and are the highest priority approaches for inclusion in future introductory notebooks. These approaches and libraries include:

- **anchors** - New research from the inventors of LIME that uses rules to explain machine learning predictions.
- **eli5** - A popular Python library with implementations of LIME and treeinterpreter.
- **LIME** - The Python library written by the inventors of LIME.
- **RuleFit** - Jerome Friedman's R package for fitting interpretable rule ensembles.
- **Shapley explanations** - A promising new approach that unifies LIME, treeintepreter, and other pre-existing interpretability work.
- **Treeinterpreter** - The Python package authored by the inventor of treeinterpreter.

4 Recommended Reading

These papers summarize some of the most important issues in machine learning interpretability and are also approachable for less technical practitioners.

- *Ideas for Machine Learning Interpretability* by Patrick Hall, Wen Phan, and SriSatish Ambati[6]
- *Interpretability* by Fast Forward Labs[7]
- *The Mythos of Model Interpretability* by Zachary C. Lipton[8]
- *Towards A Rigorous Science of Interpretable Machine Learning* by Finale Doshi-Velez and Been Kim[9]

5 References

[1] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. ***The Elements of Statistical Learning***. Springer, New York, 2001. URL https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf.

[2] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1), 2015. URL https://arxiv.org/pdf/1309.6392.pdf.

[3] Mark W. Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. *Advances in Neural Information Processing Systems*, 1996. URL http://papers.nips.cc/paper/1152-extracting-tree-structured-representations-of-trained-networks.pdf.

[4] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association (just-accepted)*, 2017. URL http://www.stat.cmu.edu/~ryantibs/papers/conformal.pdf.

[5] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016. URL http://www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf.

[6] Patrick Hall, Wen Phan, and Sri Satish Ambati. Ideas on interpreting machine learning. *O'Reilly Ideas*, 2017. URL https://www.oreilly.com/ideas/ideas-on-interpreting-machine-learning.

[7] Fast Forward Labs. Interpretability. 2017. URL http://blog.fastforwardlabs.com/2017/08/02/interpretability.html.

[8] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint*, 2016. URL https://arxiv.org/pdf/1606.03490.pdf.

[9] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint*, 2017. URL https://arxiv.org/pdf/1702.08608.pdf.

6 About the Developers

The developers of this notebook series have spent years making machine learning projects successful in the regulated industry verticals of financial services and insurance. Interpretability, transparency, and accountability of predictive models and results are typically key differentiators in successful commercial applications of machine learning, and the developers recently used their combined experience to design and develop a first-of-its-kind interactive dashboard module for interpreting, debugging, and explaining sophisticated machine learning models, particularly those generated by the award-winning H2O Driverless AI expert system.

Patrick Hall is a senior director for data science products at H2O.ai and adjunct faculty in the Department of Decision Sciences at George Washington University. He is a frequent speaker on the topics of FAT/ML and explainable artificial intelligence (XAI) at conferences and on webinars.

Navdeep Gill is a software engineer and data scientist at H2O.ai. He has made important contributions to the popular open source H2O machine learning library and the newer open source h2o4gpu library. Navdeep also led a recent Silicon Valley Big Data Science Meetup about interpretable machine learning.

Mark Chan is a software engineer and customer data scientist at H2O.ai. He has contributed to the open source H2O library and to critical financial services customer products.

Address correspondence to phall@h2o.ai or file a GitHub issue.