Scalable & Responsible Automatic Machine Learning (AutoML)

Erin LeDell Ph.D.
@ledell

Agenda

- Automatic Machine Learning (AutoML)
- H2O AutoML
- Automatic Explainability
- Interpretability & Fairness
- Admissible AutoML

Intro to Automatic Machine Learning

Why AutoML?

“No free lunch”

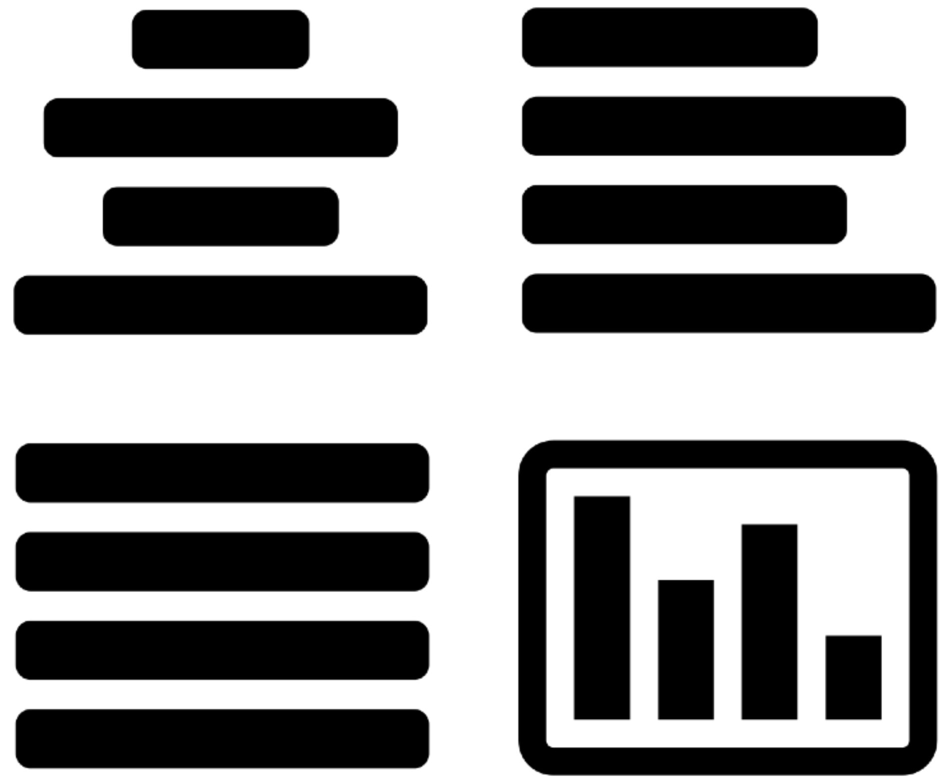
- No Free Lunch Theorem: All algorithms are the equivalent when averaged over all problems. In other words, there’s no single “best” algorithm.
- This is why we need to test many algorithms for any particular dataset/problem, and the purpose of AutoML is to automate & speed up this process.

https://en.wikipedia.org/wiki/No_free_lunch_theorem

Goals & Features of AutoML

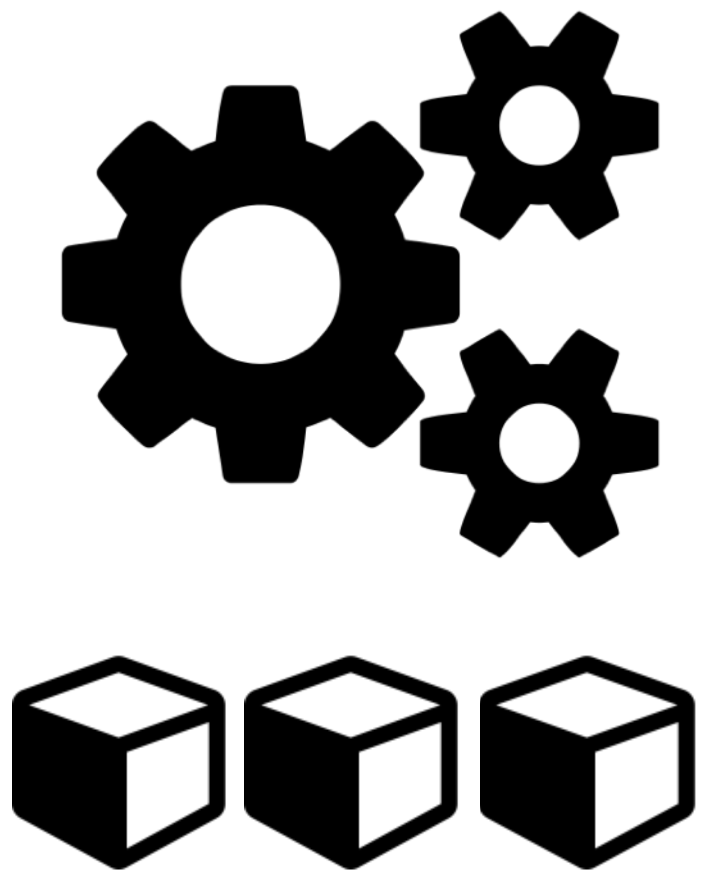
- 🏆 Train the best model in the least amount of time.
- 📉 Reduce the human effort & expertise required in machine learning.
- 📈 Improve the performance of machine learning models.
- 🔄 Increase reproducibility & establish a baseline for scientific research or applications.

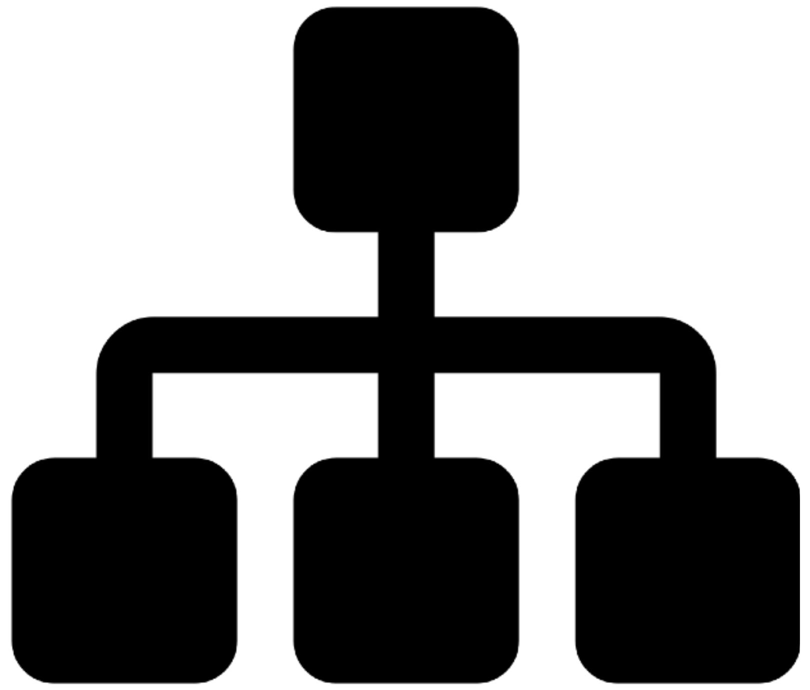
Aspects of Automatic Machine Learning



Data Prep

Model Generation





Ensembles

Different Flavors of AutoML

https://www.h2o.ai/blog/t

Most Visited docs.h2o.ai User Guide pydocs rdocs

Blog Communities Academics Documentation Downloads Contact Us

H2O.ai

Products Solutions Customers Partners Support Company Free Trial

August 15th, 2018

The different flavors of AutoML

Share **Category:** AutoML, Data Science, Driverless AI, H2O

By: Erin LeDell

In recent years, the demand for machine learning experts has outpaced the supply, despite the surge of people entering the field. To address this gap, there have been big strides in the development of user-friendly machine learning software (e.g. H2O, scikit-learn, keras). Although these tools have made it easy to train and evaluate machine learning models, there is still a good amount of data science knowledge that's required in order to create the *highest-quality* model, given your dataset. Writing the code to perform a hyperparameter search over many different types of algorithms can also be time consuming and repetitive work.

What is AutoML?

https://tinyurl.com/flavors-of-automl

H2O AutoML

H2O Machine Learning Platform

- Distributed (multi-core + multi-node) implementations of cutting edge ML algorithms.
- Core algorithms written in high performance Java.
- APIs available in R, Python, Scala; web GUI.
- Easily deploy models to production as pure Java code.
- Works on Hadoop, Spark, EC2, your laptop, etc.

H2O AutoML

- Basic data pre-processing (as in all H2O algos).
- Trains a random grid of GBMs, DNNs, GLMs, etc. using a carefully chosen hyper-parameter space.
- Individual models are tuned using cross-validation.
- Two Stacked Ensembles are trained ("All Models" ensemble & a lightweight "Best of Family" ensemble).
- Returns a sorted "Leaderboard" of all models.
- All models can be easily exported to production.

H2O AutoML in Python

Example

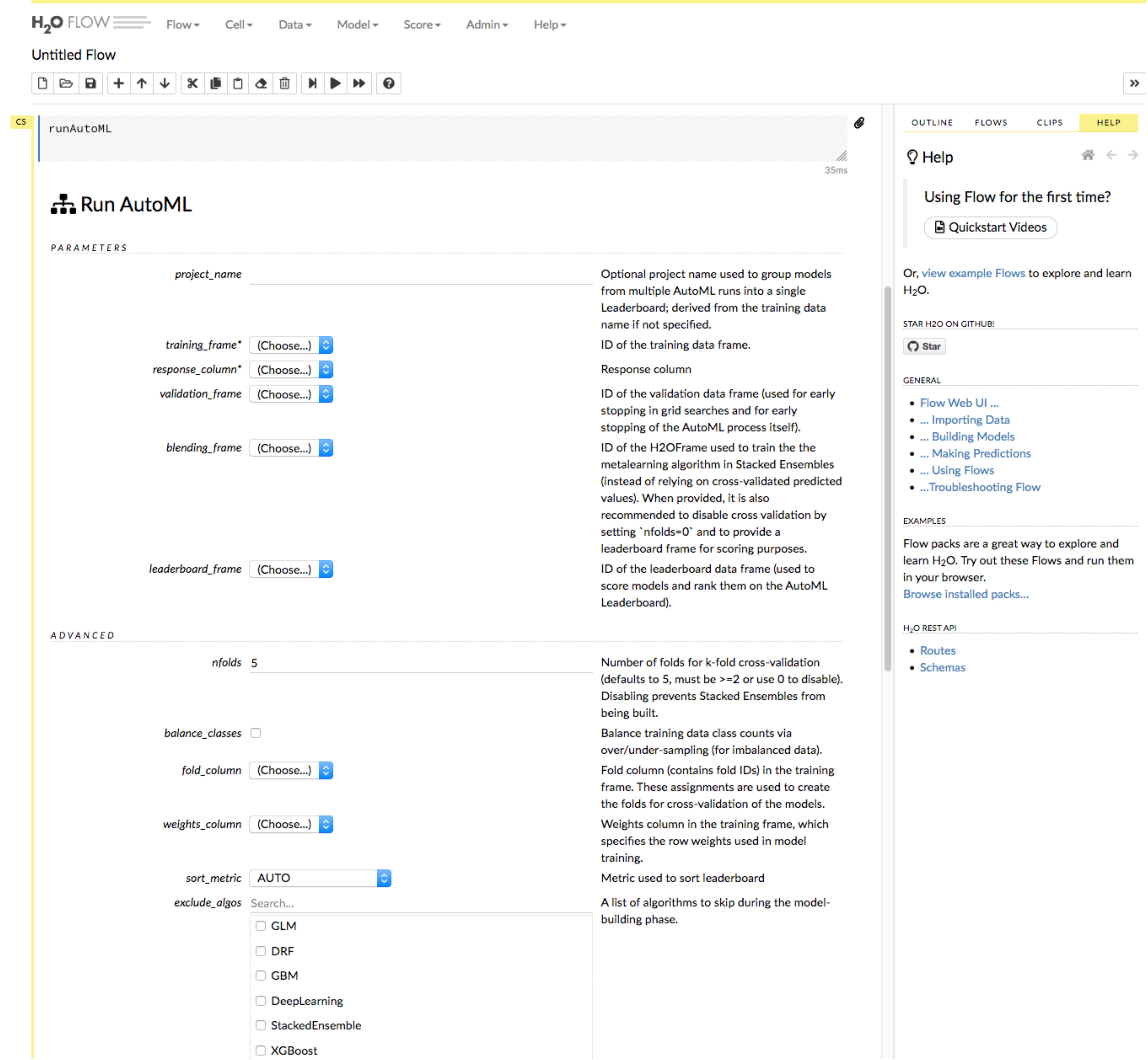
```python
import h2o
from h2o.automl import H2OAutoML
h2o.init()

train = h2o.import_file("train.csv")

aml = H2OAutoML(max_runtime_secs = 600)
aml.train(y = "response_colname",
          training_frame = train)

lb = aml.leaderboard
```

H2O AutoML in R

Example

```
library(h2o)

h2o.init()

train <- h2o.importFile("train.csv")


aml <- h2o.automl(y = "response_colname",
                  training_frame = train,
                  max_runtime_secs = 600)

lb <- aml@leaderboard
```

H2O AutoML in Flow (GUI)



H2O AutoML Leaderboard

	model_id	auc	logloss	aucpr	mean_per_class_error	rmse	mse
1	StackedEnsemble_AllModels_AutoML_20200709_004...	0.8378355	0.2866370	0.4481733	0.2498460	0.2913039	0.08485799
2	StackedEnsemble_BestOfFamily_AutoML_20200709_0...	0.8369381	0.2869531	0.4462222	0.2500683	0.2914670	0.08495302
3	XGBoost_3_AutoML_20200709_004415	0.8366588	0.2809896	0.4502926	0.2552901	0.2894478	0.08378002
4	GBM_4_AutoML_20200709_004415	0.8330289	0.2848382	0.4239271	0.2593298	0.2919957	0.08526147
5	GBM_3_AutoML_20200709_004415	0.8325824	0.2852444	0.4195761	0.2552272	0.2922670	0.08542002
6	GBM_2_AutoML_20200709_004415	0.8323248	0.2855498	0.4185351	0.2589230	0.2924915	0.08555129
7	GBM_1_AutoML_20200709_004415	0.8322315	0.2855884	0.4200573	0.2622791	0.2922375	0.08540278
8	XGBoost_1_AutoML_20200709_004415	0.8317490	0.2858897	0.4326282	0.2618297	0.2923182	0.08544993
9	GBM_5_AutoML_20200709_004415	0.8296069	0.2874258	0.4040567	0.2569593	0.2938664	0.08635746
10	XGBoost_2_AutoML_20200709_004415	0.8277037	0.2899311	0.4265391	0.2624847	0.2943874	0.08666391
11	DRF_1_AutoML_20200709_004415	0.8120043	0.3008964	0.3722857	0.2731671	0.2991530	0.08949252
12	GLM_1_AutoML_20200709_004415	0.6873574	0.3510707	0.2172795	0.3673990	0.3194751	0.10206432

Example Leaderboard for binary classification

Auto Explainability

H2O AutoML Explainability

- The new `h2o.explain()` interface automatically generates many explanations (annotated visualizations) for a single model or a group of models (e.g. AutoML leaderboard).
- Row-wise explanations are available via the `h2o.explain_row()` companion function.
- Visualizations are created with `ggplot2` in R and `matplotlib` in Python, and can be customized.

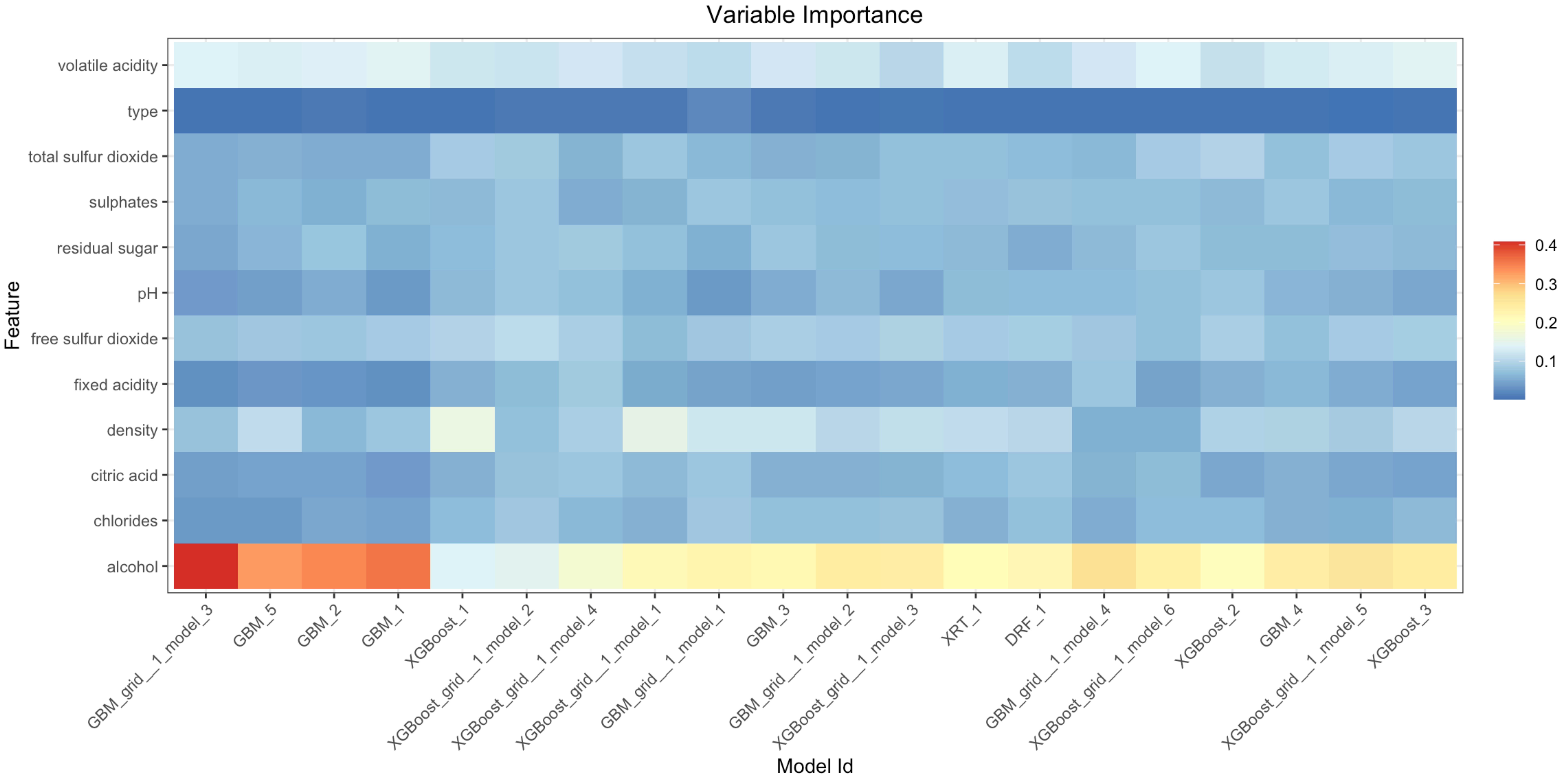
H2O AutoML Explainability

Automatic Explanations: 🔬 🛠️

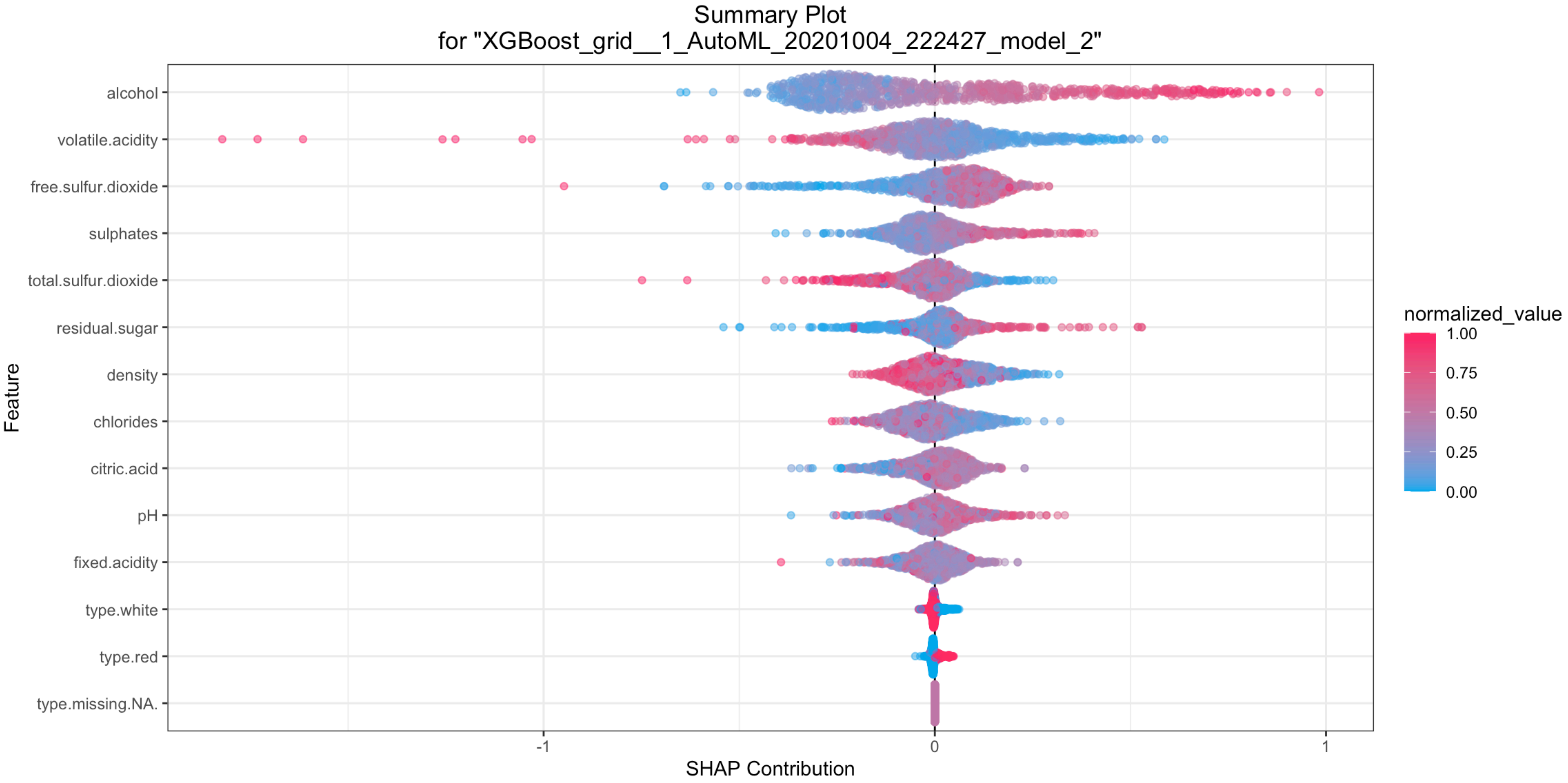
- Variable importance comparisons
- Model correlation heatmap
- SHAP contributions for tree-based models
- Partial dependence (PD) plots
- Individual Conditional Expectation (ICE) plots
- Residual Analysis

http://docs.h2o.ai/h2o/latest-stable/h2o-docs/explain.html

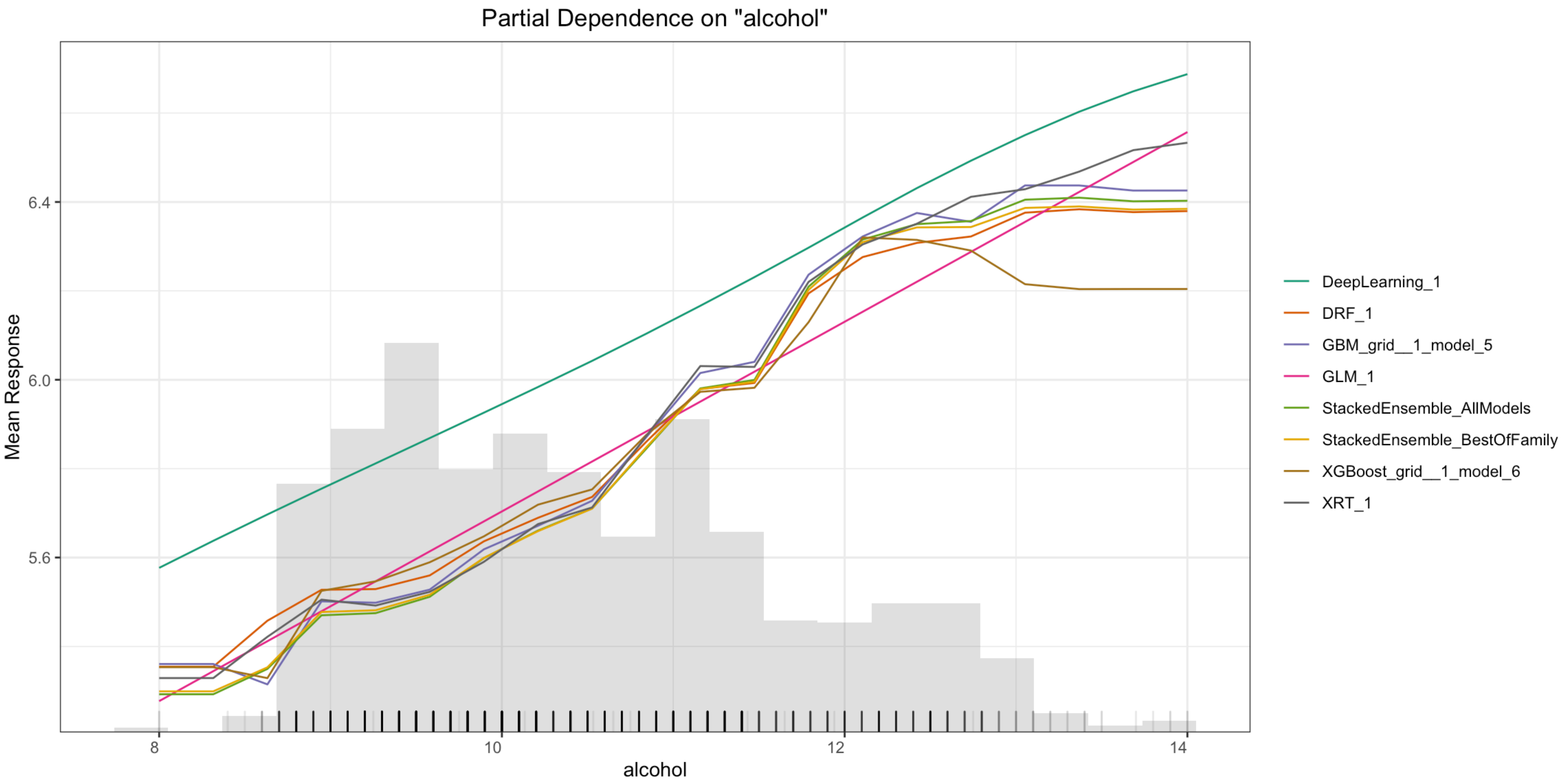
Variable Importance Heatmap



SHAP Summary



Partial Dependence (PD) Plots



Interpretability & Fairness

Interpretable ML

- 🧠 🔍 In Machine Learning, "interpretable" refers to techniques and models which are understandable by humans.
- 📊 🌲 Some models are inherently interpretable, such as Linear Models or Decision Trees.
- 🔑 🗃️ Note that this is not the same as "Explainable ML" which refers to techniques for helping to explain models (typically black-box models which are not interpretable).

Fairness in ML

What is "fairness" in ML? ⚖️

- Very hard question to answer. 🤔
- Describes a broad set of problems, not a specific approach or metric.

Why it's hard:

- Individual fairness vs. group fairness
- Law/policy is somewhat fuzzy – moving target...

Fairness in ML

- ⚖️ Fairness in ML refers to techniques for measuring and/or mitigating the disparate impact on subsets of people defined by "protected" attributes such as age, gender, race, etc.
- 👉 Note that the terms "disparate impact", "adverse impact" and "protected classes" both have colloquial and legal interpretations (and legal doctrine differs by country).

Protected Attributes for Lending in U.S.

What is credit discrimination?

The Equal Credit Opportunity Act makes it illegal for a creditor to discriminate in any aspect of credit transaction based on certain characteristics.

In addition, the Fair Housing Act makes many discrimination practices in home financing illegal.

It is illegal to:

- Refuse you credit if you qualify for it
- Discourage you from applying for credit
- Offer you credit on terms that are less favorable, like a higher interest rate, than terms offered to someone with similar qualifications
- Close your account

On the basis of:

- Race
- Color
- Religion
- National origin
- Sex (including sexual orientation and gender identity)
- Marital status
- Age
- Receiving money from public assistance
- Exercising in good faith your rights under the Consumer Credit Protection Act.

Consumer Financial Protection Bureau

- ⚖️ Several U.S. laws protect consumers from credit discrimination.
- 🏦 Lenders must not discriminate based on these attributes.

https://www.consumerfinance.gov/fair-lending/

Admissible Machine Learning

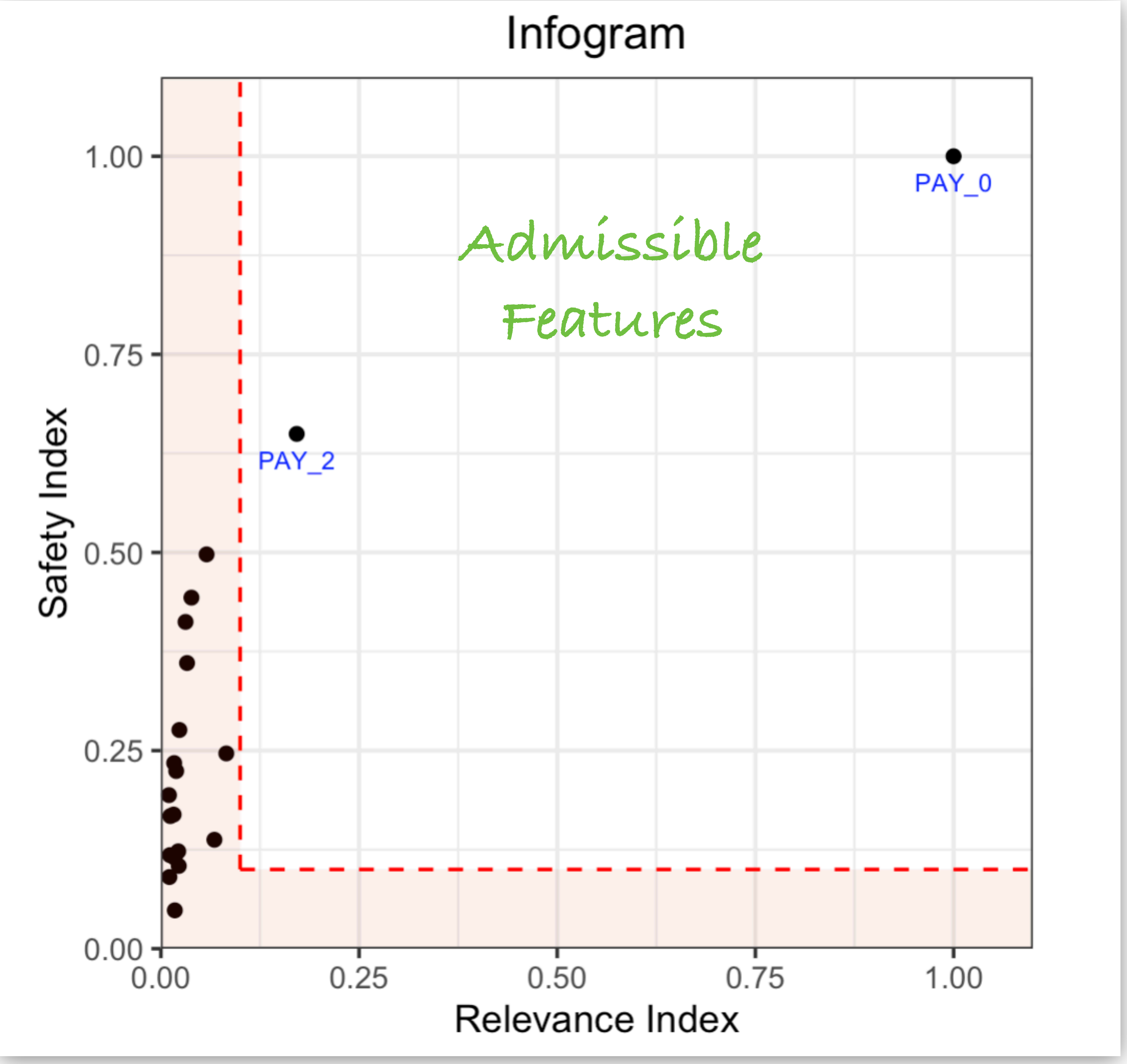
Goals and Features of Admissible ML

- 🙈 🙉 🙊 Addresses the issue of "Fairness through unawareness" (even if you remove protected variables, you could still have bias in your model).
- 😷 Filter out redundant and/or unsafe variables from your data prior to training models.
- 👷‍♀️ Tackle bias head-on vs. trying to debias models in a post-processing step or via exhaustive search.
- 👼 Easy to use; reduce barriers to safer ML.

Fair Infogram

😷 Identify unprotected variables that have high relevance and safety.

- The x-axis is relevance index, a measure of how much the variable drives the response (the more predictive, the higher the relevance).
- The y-axis is safety index, a measure measure of how much extra information the variable has that is not acquired through the protected variables.



Admissible AutoML

🤖 In the AutoML context, the goal is usually to maximize model performance 📈 within a fixed budget. 🏦

✂️ The reduced, admissible feature set allows us to train models faster and cover a larger search space for the same cost in AutoML systems.

Admissible AutoML is an automatic algorithmic risk-assessment method

🤖 😷

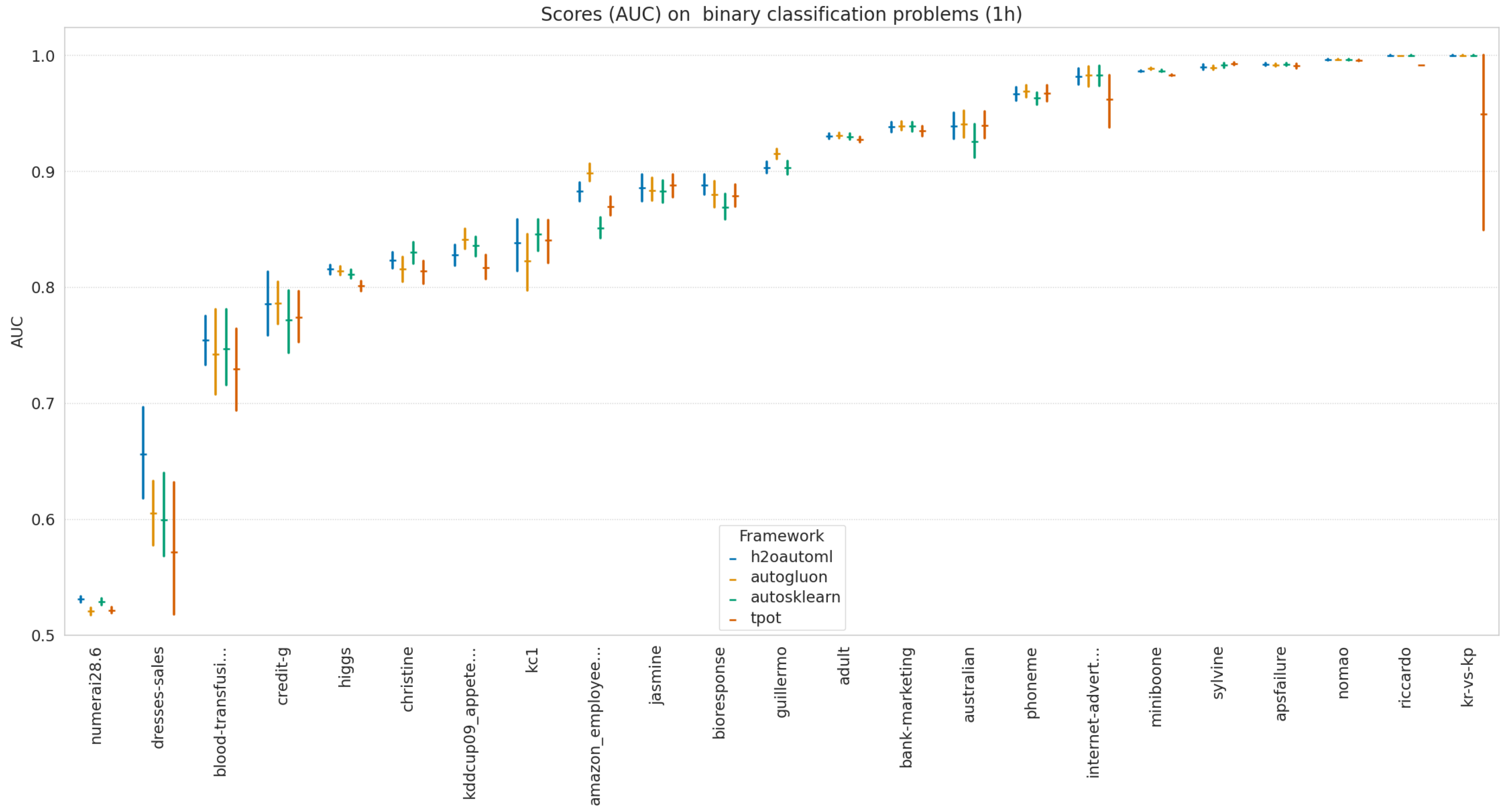
🔄 Automatically train models which reduce bias from protected variables by filtering out the inadmissible features.

Resources

H2O AutoML paper

The H2O AutoML paper was published at ICML 2020 AutoML Workshop

- *Official H2O AutoML paper*
- *Updated benchmarks*
- *Scalability study (100M rows)*
- *Stacking study*



ICML paper ⬇️ https://tinyurl.com/icml2020-h2o-paper

Learn H2O!

- Admissible ML Docs: https://tinyurl.com/h2o-admissible
- AutoML Docs: https://tinyurl.com/h2o-automl-docs
- AutoML tutorials: https://tinyurl.com/h2o-automl-tutorials
- Explainability: https://tinyurl.com/h2o-explain

H2O Resources

- Documentation: http://docs.h2o.ai
- Tutorials: https://github.com/h2oai/h2o-tutorials
- Slidedecks: https://github.com/h2oai/h2o-meetups
- Videos: https://www.youtube.com/user/0xdata
- Stack Overflow: https://stackoverflow.com/tags/h2o
- Google Group: https://tinyurl.com/h2ostream
- Gitter: http://gitter.im/h2oai/h2o-3
- Events & Meetups: http://h2o.ai/events

Thank you!

👋 @ledell on Github, Twitter