Scalable AutoML in R & Python using H2O

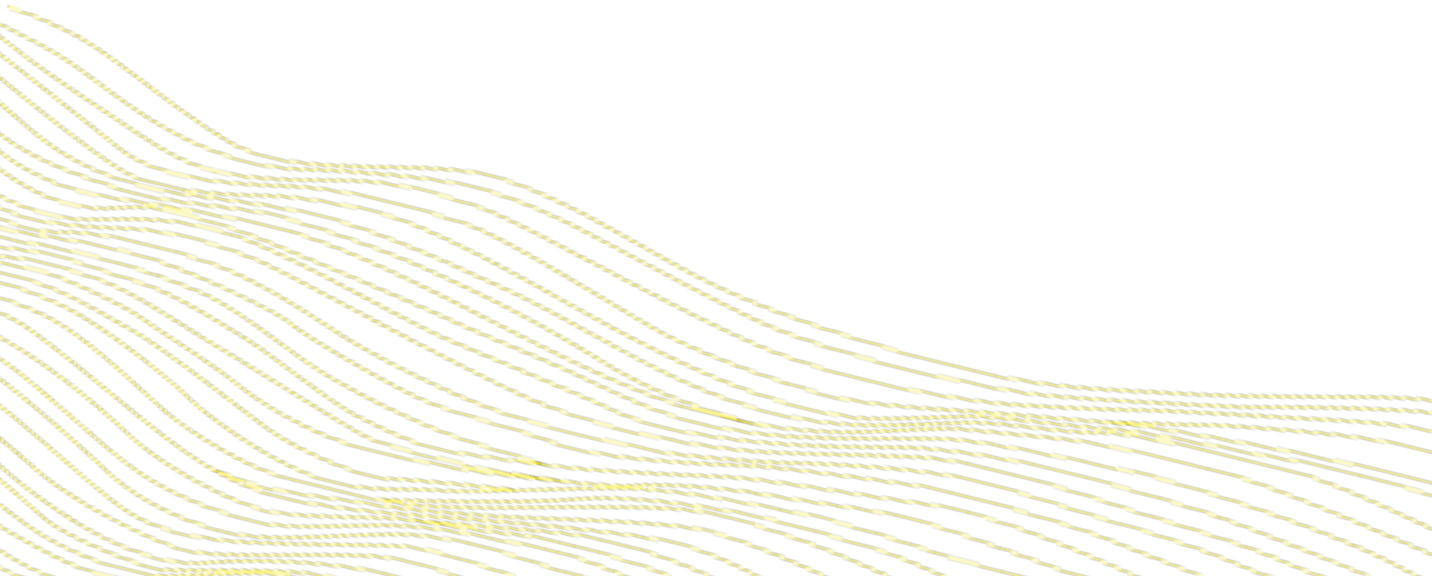
H2O.ai

Erin LeDell Ph.D.
@ledell

Agenda

- Automatic Machine Learning (AutoML)
- H2O AutoML
- Automatic Explainability
- New software for H2O AutoML:
  -  agua package in Tidymodels
  -  Wave UI for H2O AutoML

Intro to Automatic Machine Learning



Why AutoML?

"No free lunch"

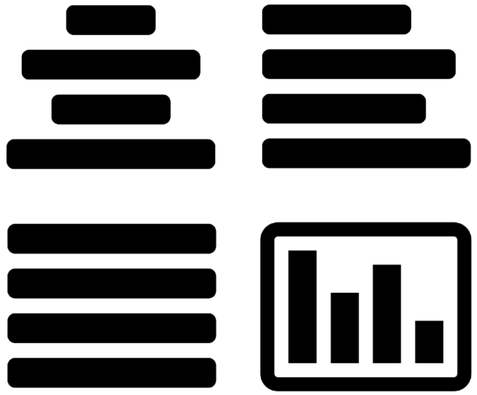
- No Free Lunch Theorem: All algorithms are the equivalent when averaged over all problems. In other words, there's no single "best" algorithm.
- This is why we need to test many algorithms for any particular dataset/problem, and the purpose of AutoML is to automate & speed up this process.

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

Goals & Features of AutoML

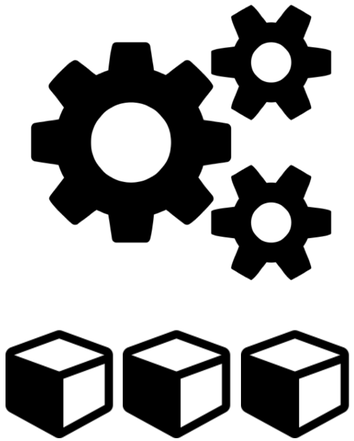
- 🏆 Train the best model in the least amount of time.
- 📉 Reduce the human effort & expertise required in machine learning.
- 📈 Improve the performance of machine learning models.
- 🔄 Increase reproducibility & establish a baseline for scientific research or applications.

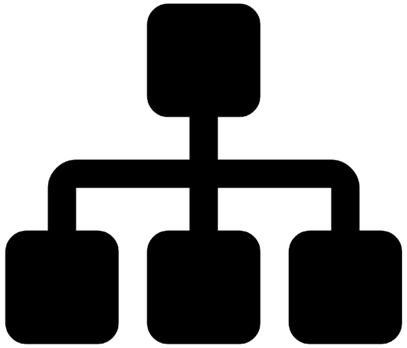
Aspects of Automatic Machine Learning



Data Prep

Model Generation





Ensembles

Different Flavors of AutoML

https://www.h2o.ai/blog/t 67% Search

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

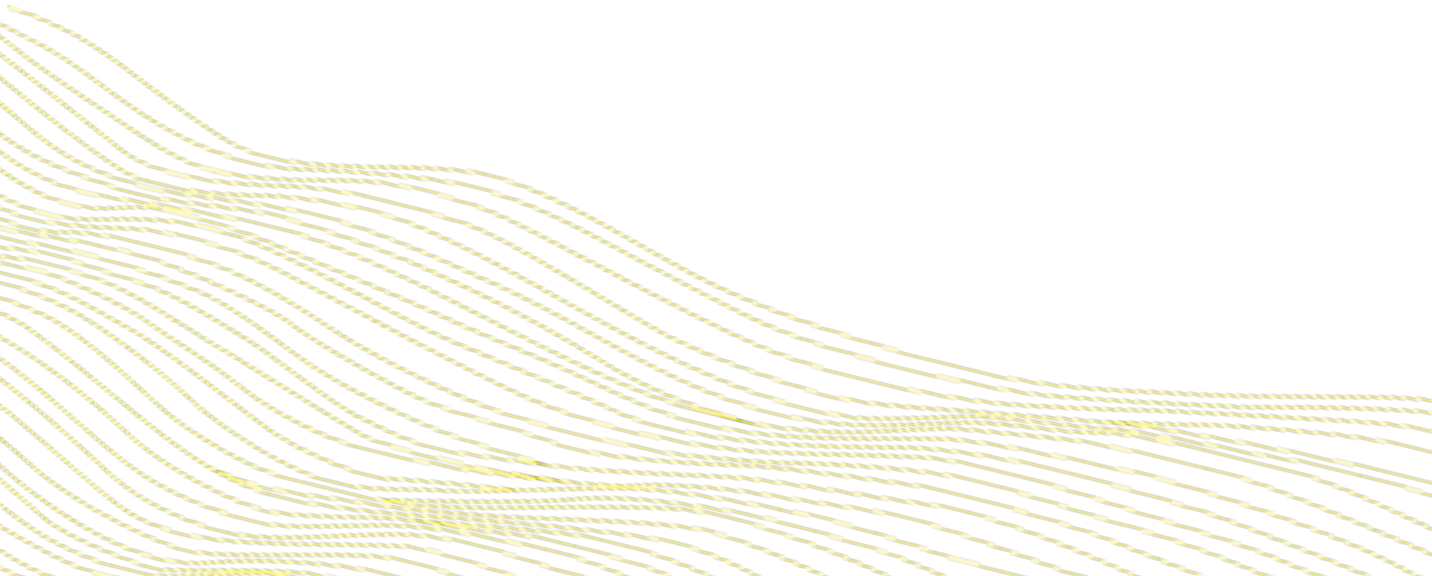
By: Erin LeDell

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?

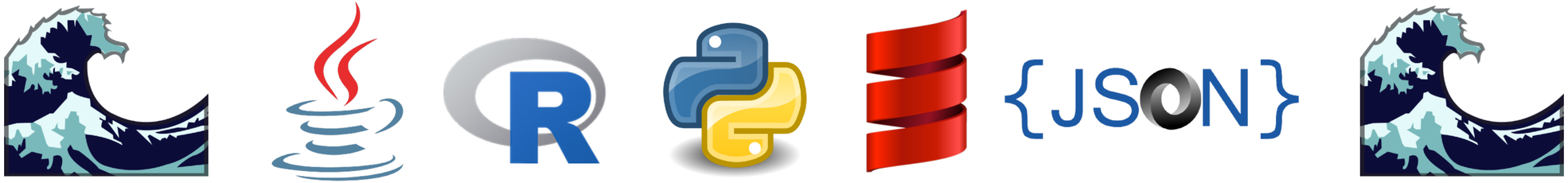
https://tinyurl.com/flavors-of-automl

H2O AutoML



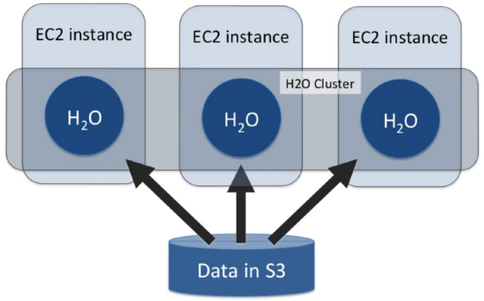
H2O Machine Learning Platform

- 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 Distributed Computing

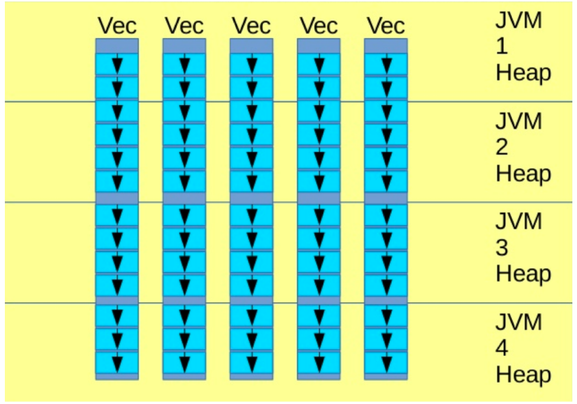
H2O Cluster



- Multi-node cluster with shared memory model.
- All computations in memory.
- Each node sees only some rows of the data.
- No limit on cluster size.

H2O Frame

- Distributed data frames (collection of vectors).
- Columns are distributed (across nodes) arrays.
- Works just like R's data.frame or Python Pandas DataFrame



Random Grid Search & Stacking

- Random Grid Search combined with Stacked Ensembles is a powerful combination.
- Ensembles perform particularly well if the models they are based on (1) are individually strong, and (2) make uncorrelated errors.
- Stacking uses a second-level metalearning algorithm to find the optimal combination of base learners.

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

```
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

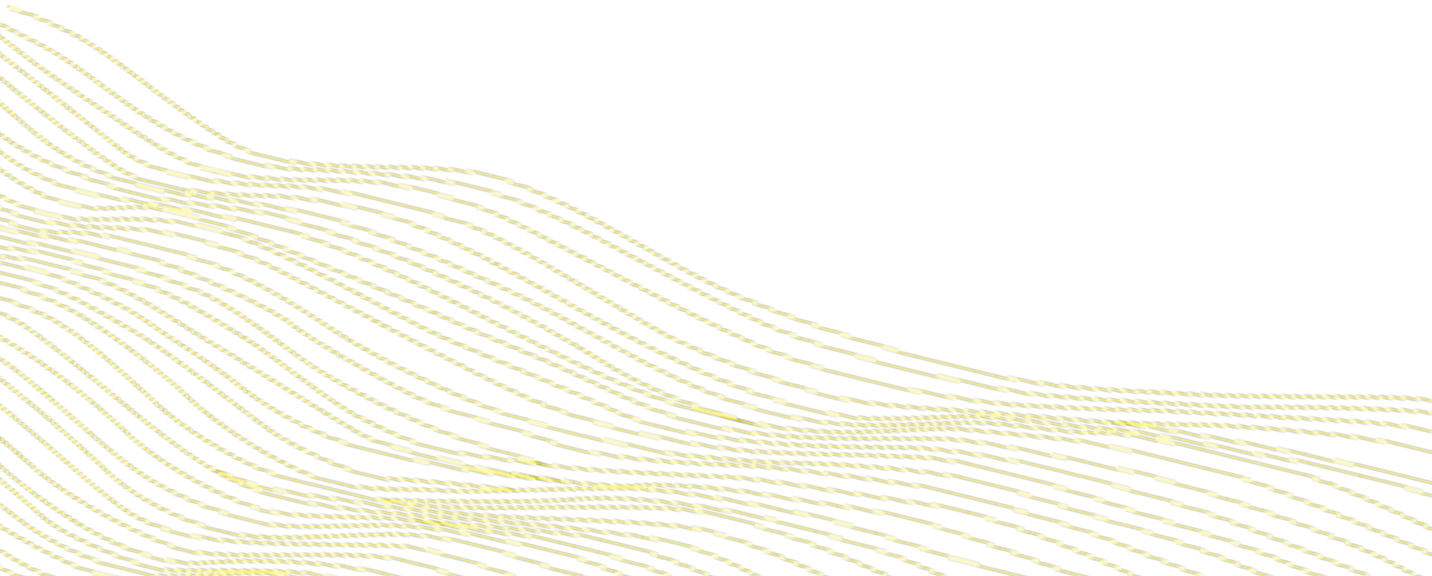
```
library(h2o)

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 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

AutoML Pro Tips!



AutoML Pro Tips: Customize

- Control time limit using `max_runtime_secs` or limit the number of models using `max_models`.
- You can turn off cross-validation for big datasets by setting `nfolds=0`. CV is required for Stacked Ensembles so that will be disabled.
- Turn on/off certain algorithms using `exclude_algos` or `include_algos`.

AutoML Pro Tips: Cluster memory

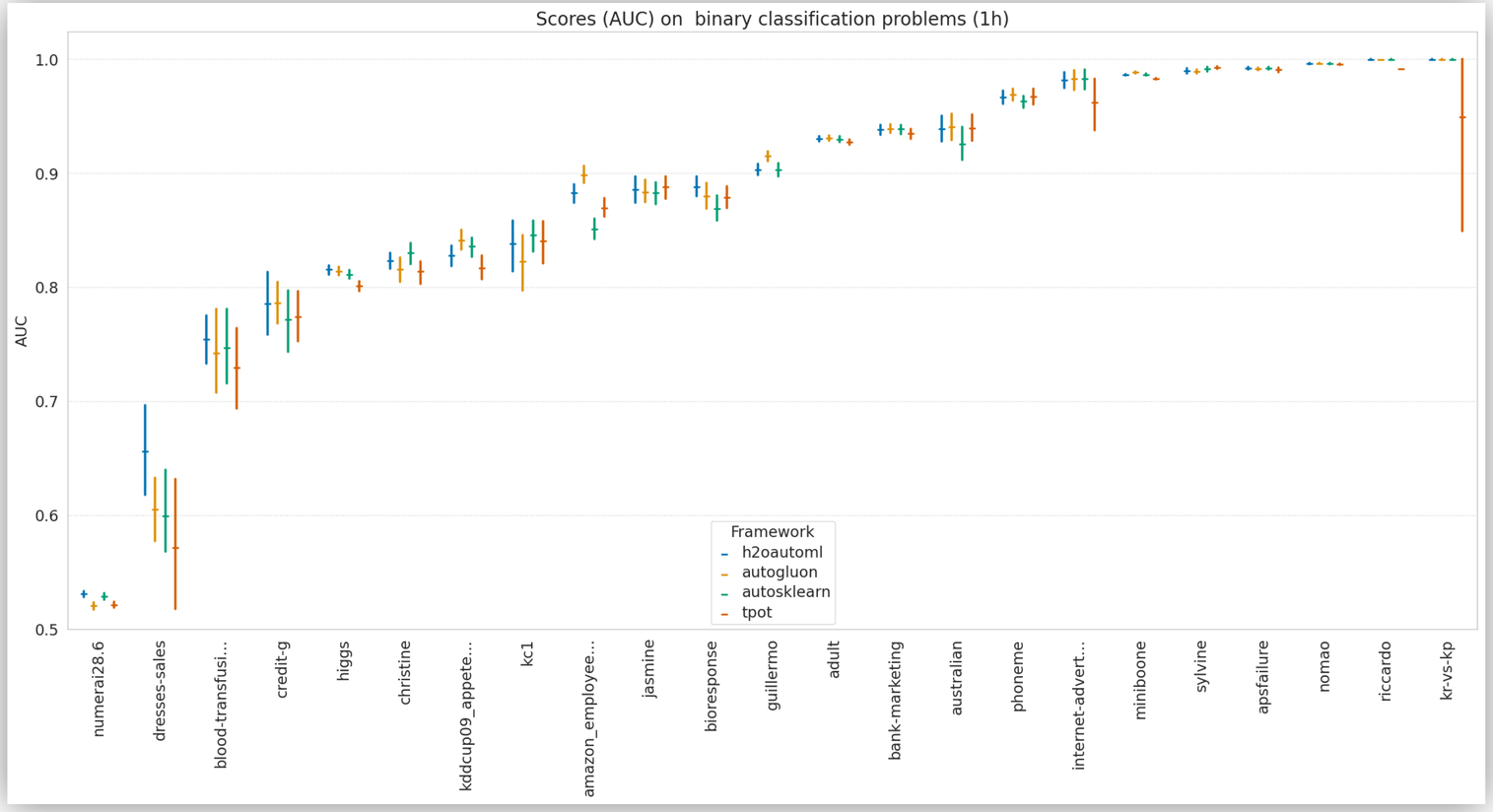
- Reminder: All H2O models are stored in H2O Cluster memory.
- Make sure to give the H2O Cluster a lot of memory if you're going to create hundreds or thousands of models.
- e.g. `h2o.init(max_mem_size = "80G")`

AutoML Pro Tips: Add More Models

- If you want to add (train) more models to an existing AutoML project, just make sure to use the same training set and `project_name`.
- If you set the same seed twice it will give you identical models as the first run (not useful), so change the seed or leave it unset.

H2O AutoML paper

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

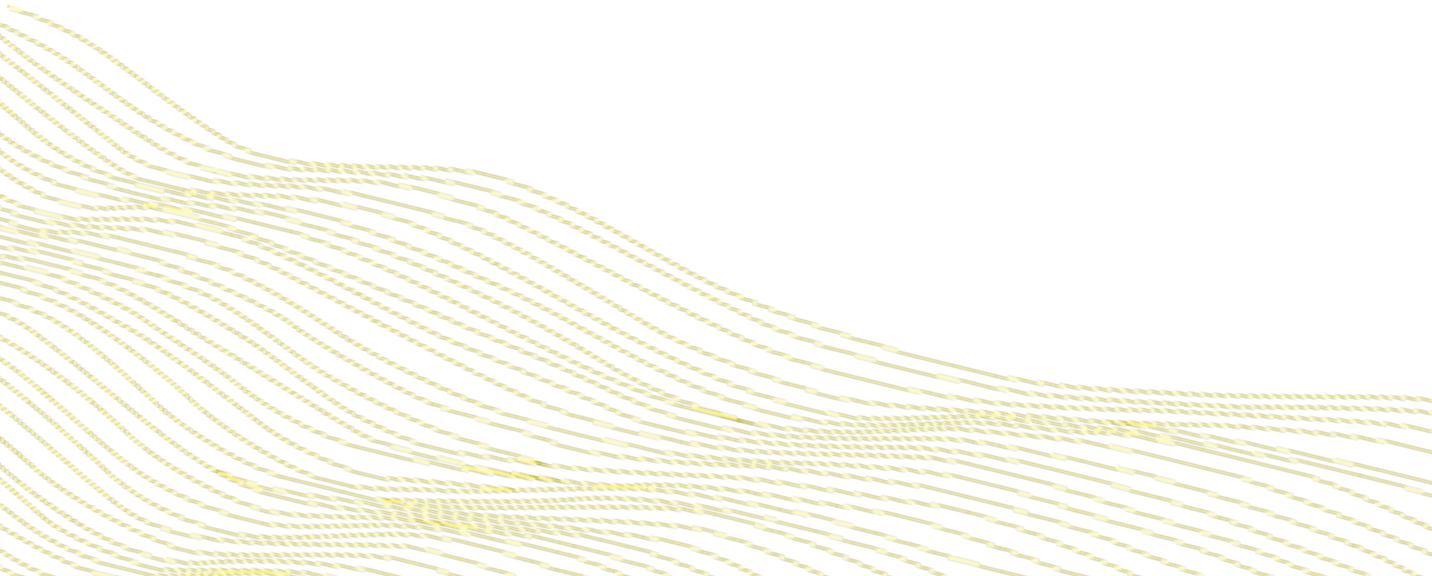


ICML paper ⬇️ https://tinyurl.com/icml2020-h2o-paper

Learn H2O AutoML!

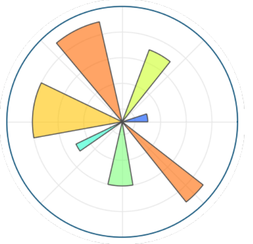
- Docs: https://tinyurl.com/h2o-automl-docs
- R & Py tutorials: https://tinyurl.com/h2o-automl-tutorials

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.



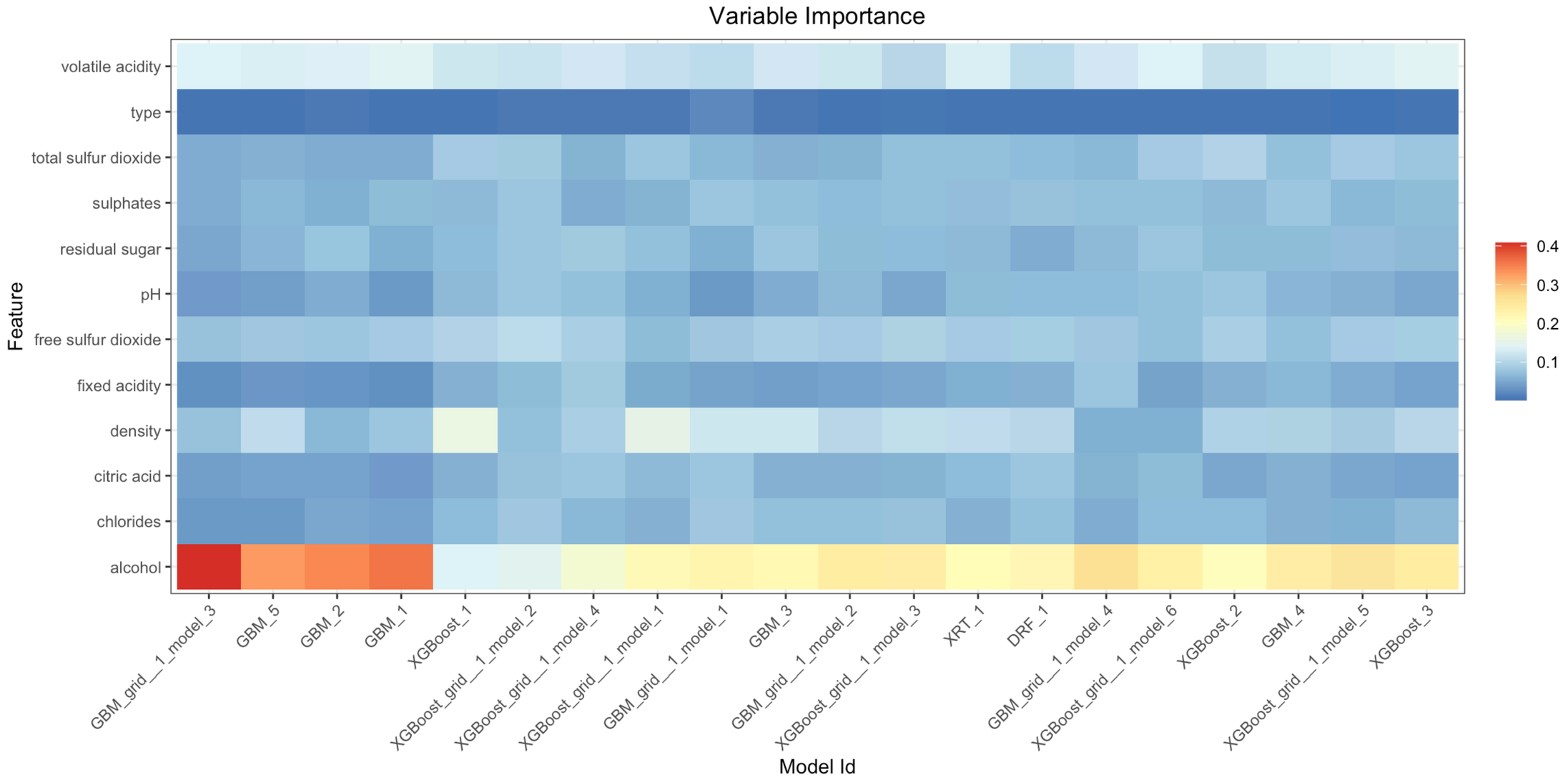
H2O AutoML Explainability

Automatic Explanations: 🔬 🛠️

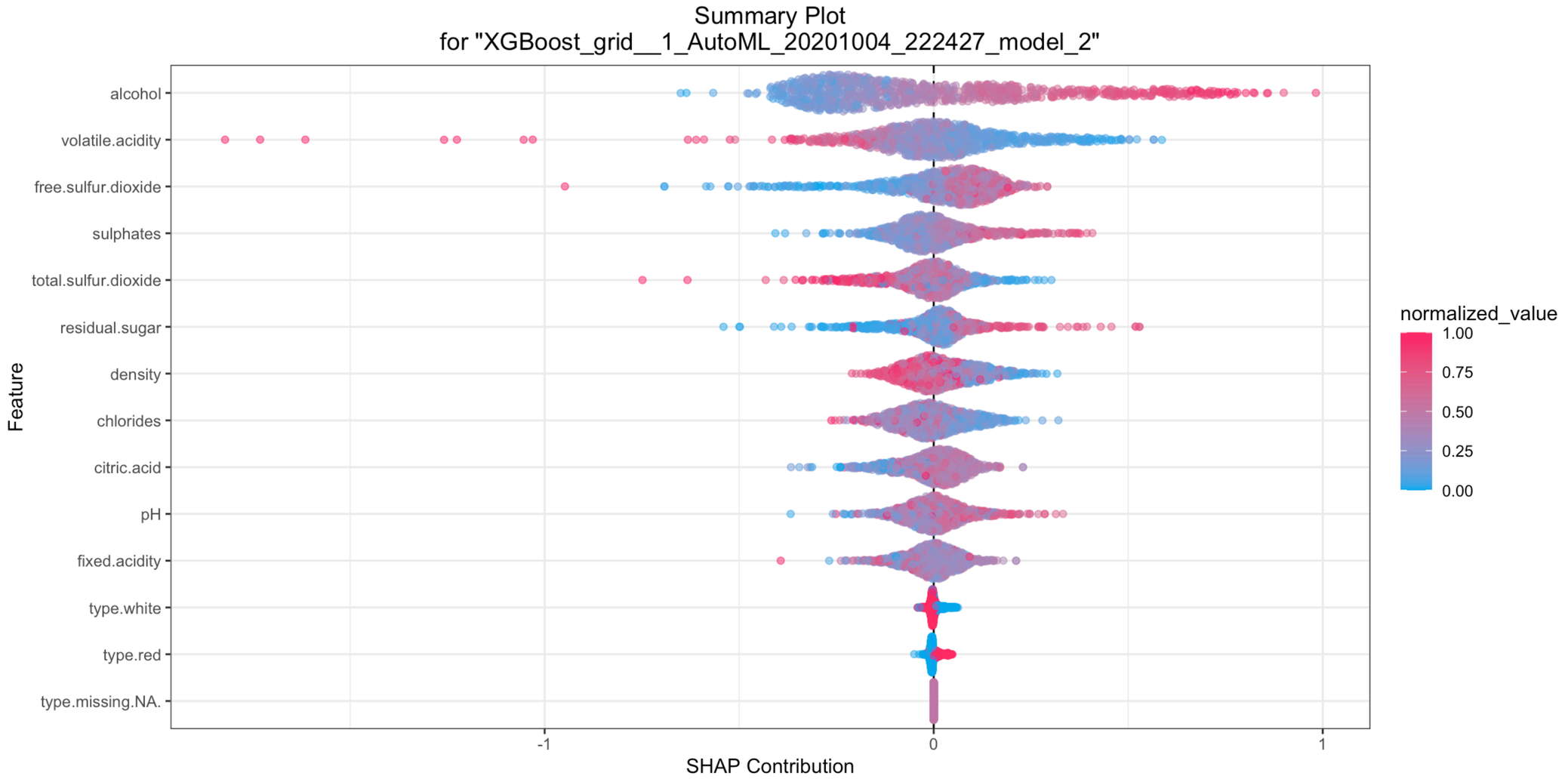
- Variable importance comparisons
- Model correlation heatmap
- SHAP contributions for tree-based models
- Partial dependence (PD) plots
- Individual Conditional Expectation (ICE) plots
- Residual Analysis

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

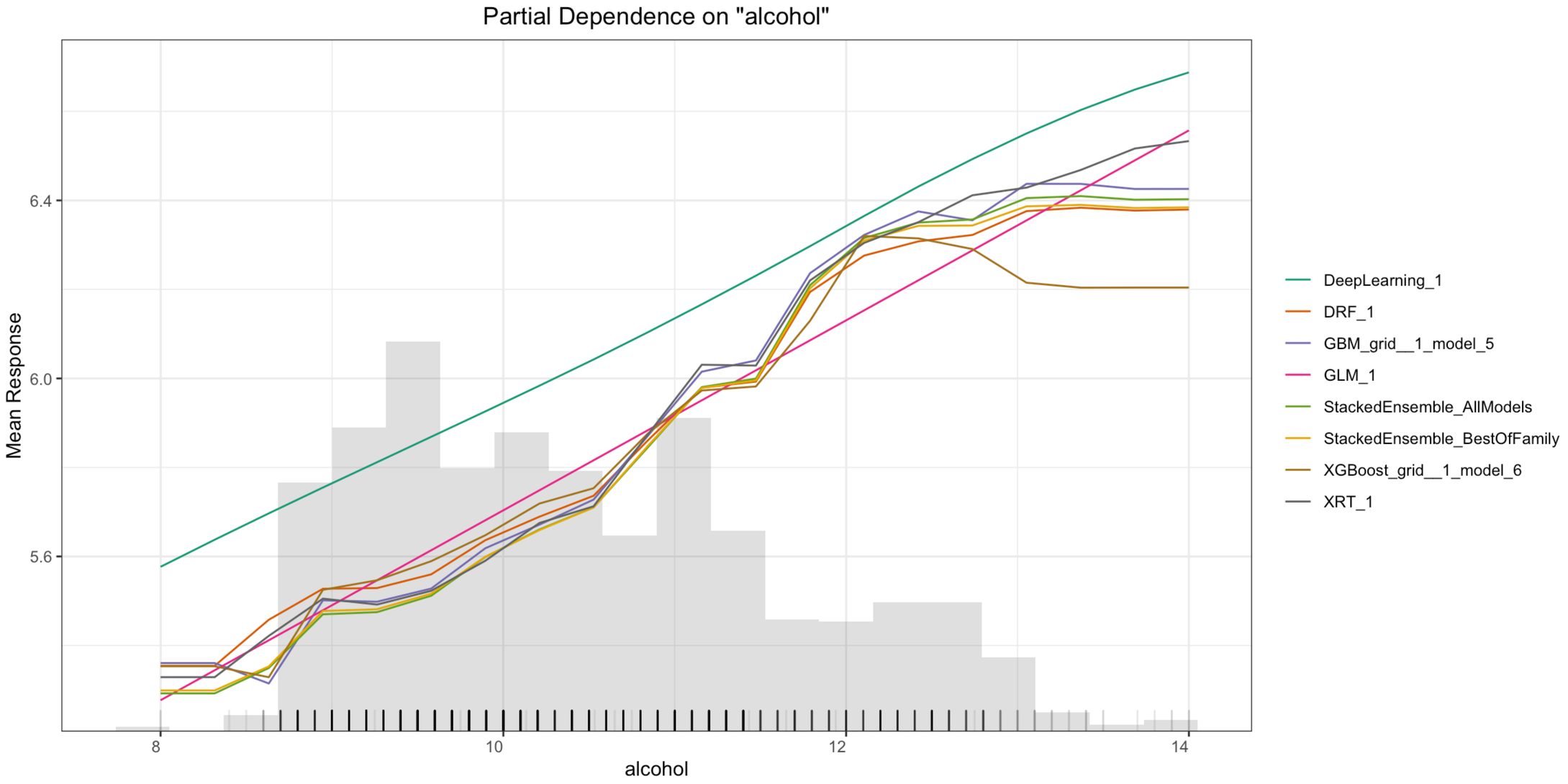
Variable Importance Heatmap



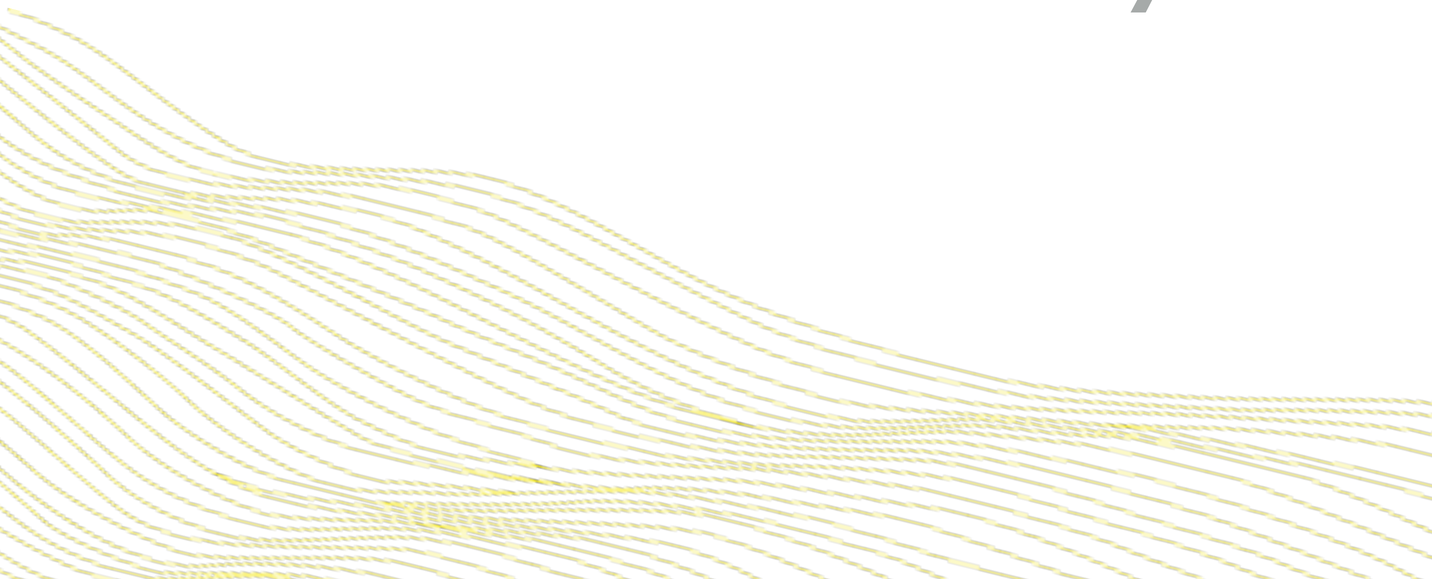
SHAP Summary



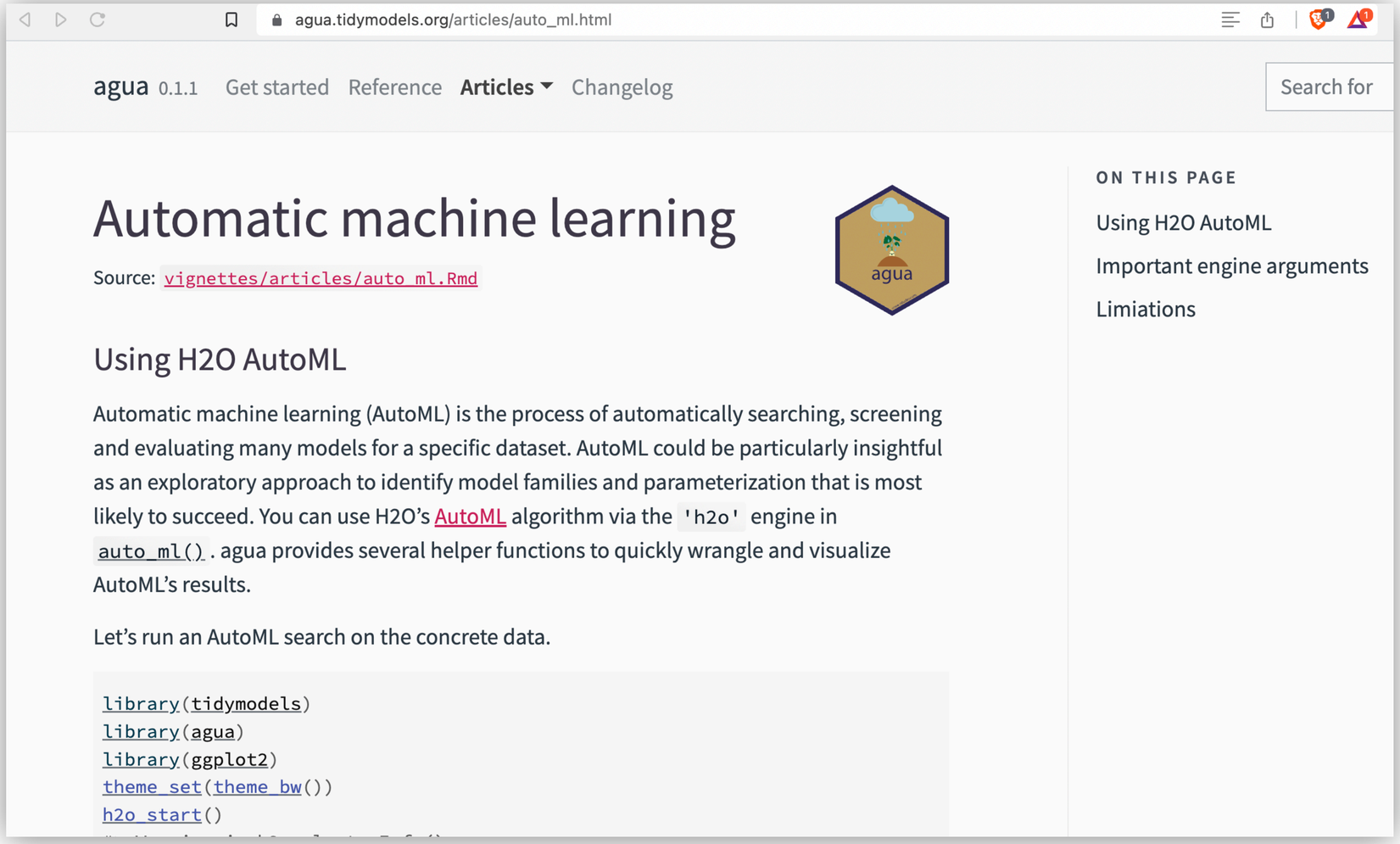
Partial Dependence (PD) Plots



agua: H2O in Tidymodels



agua: H2O AutoML in Tidymodels



- *New agua package for H2O in Tidymodels*
- *Finally AutoML in the Tidyverse!*
- *Leaderboard*
- *Auto-plotting*

https://agua.tidymodels.org/articles/auto_ml.html

agua: H2O AutoML in Tidymodels

```
# run for a maximum of 120 seconds
auto_spec <-
  auto_ml() %>%
  set_engine("h2o", max_runtime_secs = 120, seed = 1) %>%
  set_mode("regression")

normalized_rec <-
  recipe(compressive_strength ~ ., data = concrete_train) %>%
  step_normalize(all_predictors())

auto_wflow <-
  workflow() %>%
  add_model(auto_spec) %>%
  add_recipe(normalized_rec)
```

- *Specify the AutoML Run*
- *Add customized pre-processing steps*

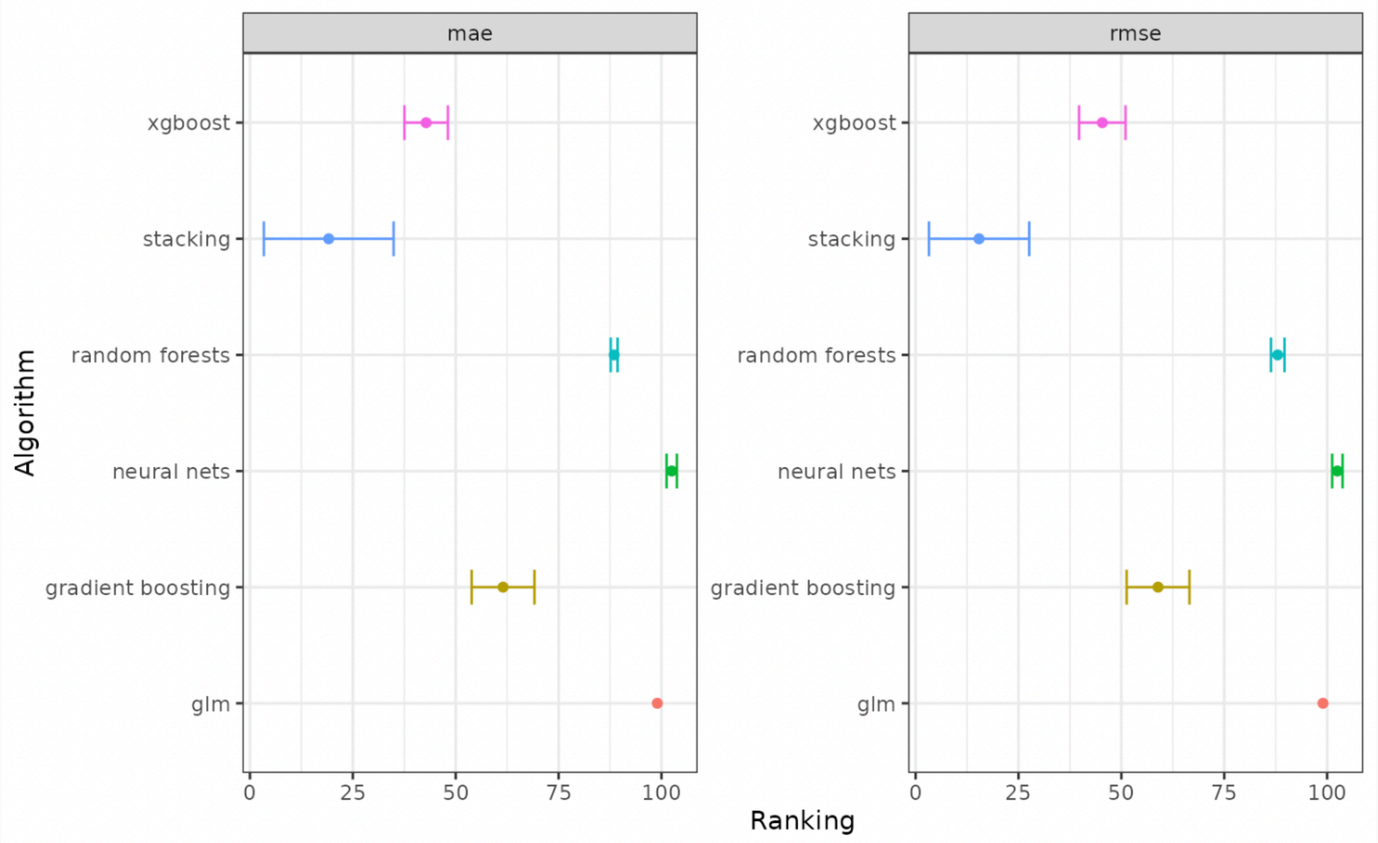
agua: H2O AutoML in Tidymodels

```
auto_fit <- fit(auto_wflow, data = concrete_train)

extract_fit_parsnip(auto_fit)
#> parsnip model object
#>
#> ═══════════════════ H2O AutoML Summary: 105 models ═══════════════════
#>
#>
#> ═══════════════════════════════ Leaderboard ═══════════════════════════════
#>                                                  model_id rmse  mse  mae
#> 1 StackedEnsemble_BestOfFamily_4_AutoML_1_20230209_174845 4.49 20.1 2.90
#> 2        XGBoost_grid_1_AutoML_1_20230209_174845_model_38 4.61 21.3 3.03
#> 3    StackedEnsemble_AllModels_2_AutoML_1_20230209_174845 4.62 21.4 3.04
#> 4    StackedEnsemble_AllModels_1_AutoML_1_20230209_174845 4.67 21.8 3.08
#> 5        XGBoost_grid_1_AutoML_1_20230209_174845_model_39 4.68 21.9 3.14
#> 6 StackedEnsemble_BestOfFamily_3_AutoML_1_20230209_174845 4.68 21.9 3.08
#>   rmsle mean_residual_deviance
#> 1 0.136                  20.1
#> 2 0.143                  21.3
#> 3 0.140                  21.4
#> 4 0.142                  21.8
#> 5 0.146                  21.9
#> 6 0.142                  21.9
```

- *Execute the AutoML Run*
- *View Leaderboard*

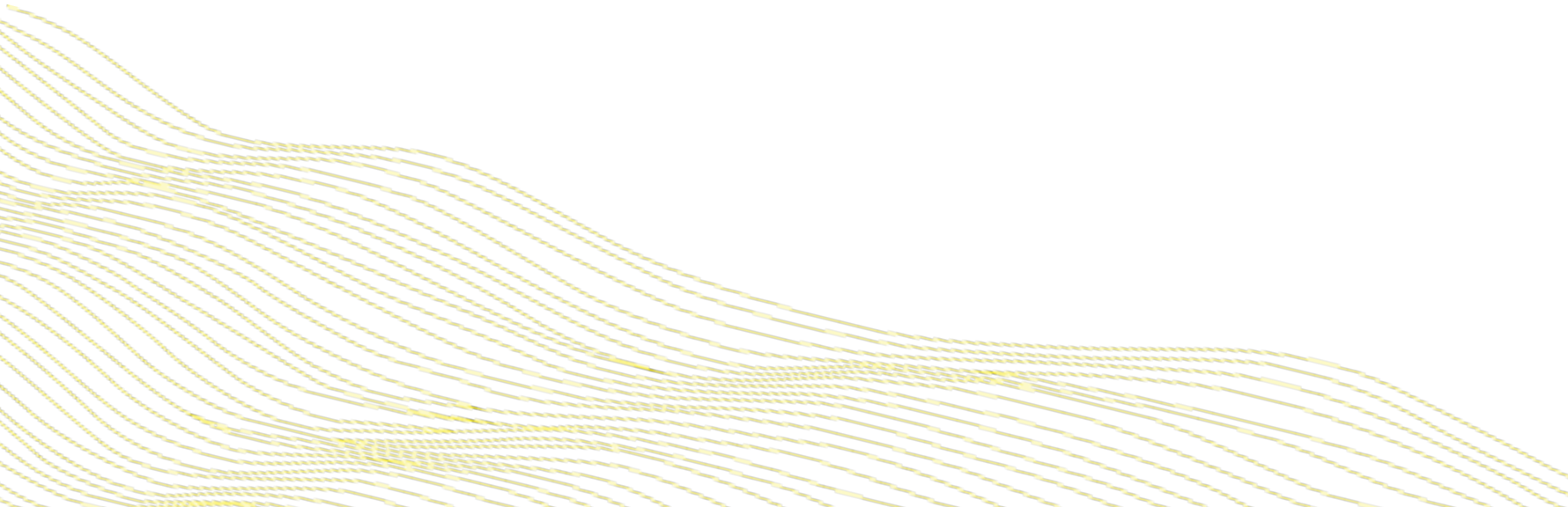
agua: H2O AutoML in Tidymodels

```
autoplot(auto_fit, type = "rank", metric = c("mae", "rmse")) +
  theme(legend.position = "none")
```

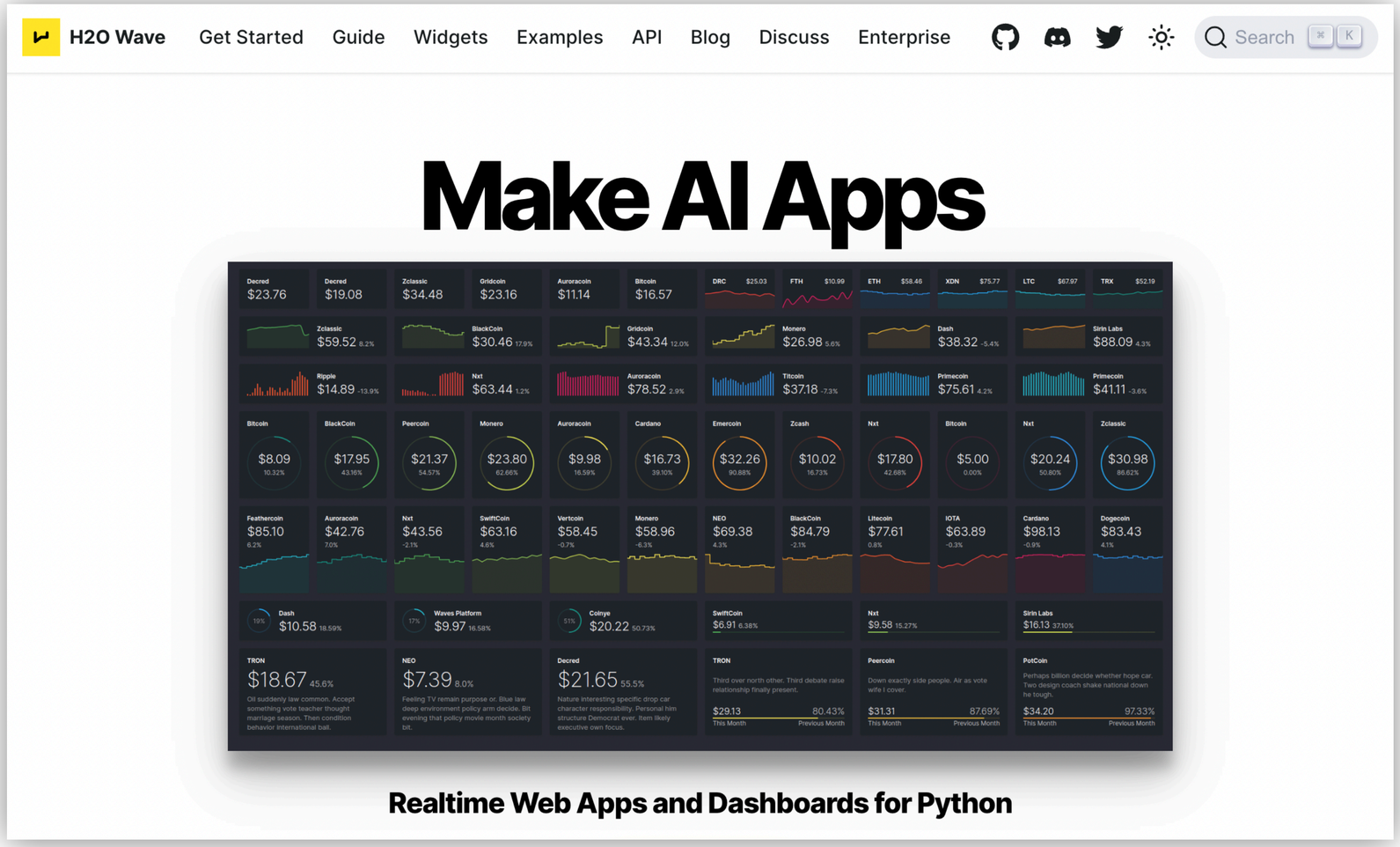


- *Autoplotting*
- *Compare performance of algorithms, visually*

H2O AutoML Wave app: A modern web GUI for AutoML



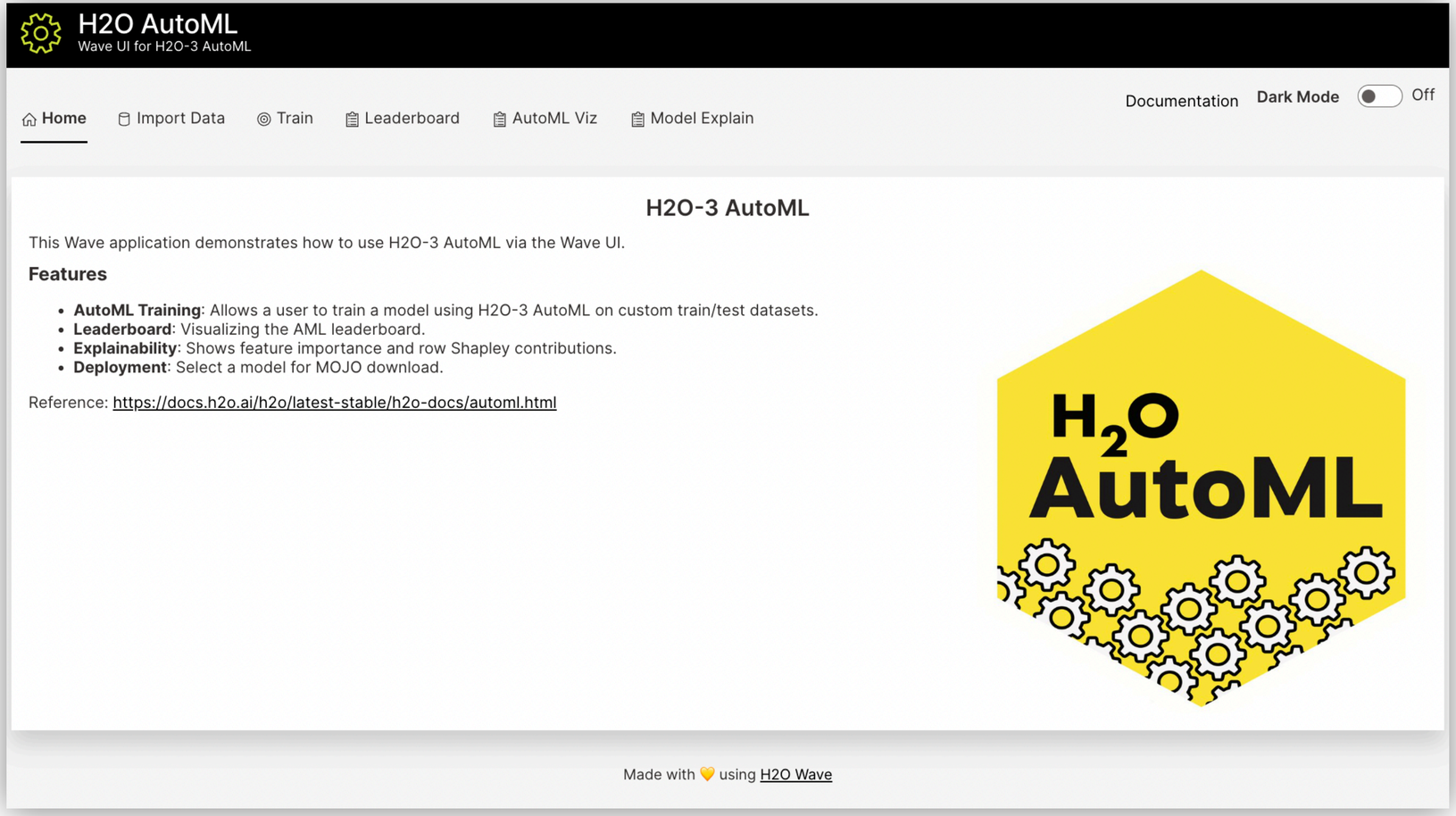
H2O Wave



- *Wave is a Python & R framework for developing realtime web apps*
- *Useful for making front-ends for ML projects*
- *Similar in function to Shiny (R) or Streamlit (Python)*

https://wave.h2o.ai/

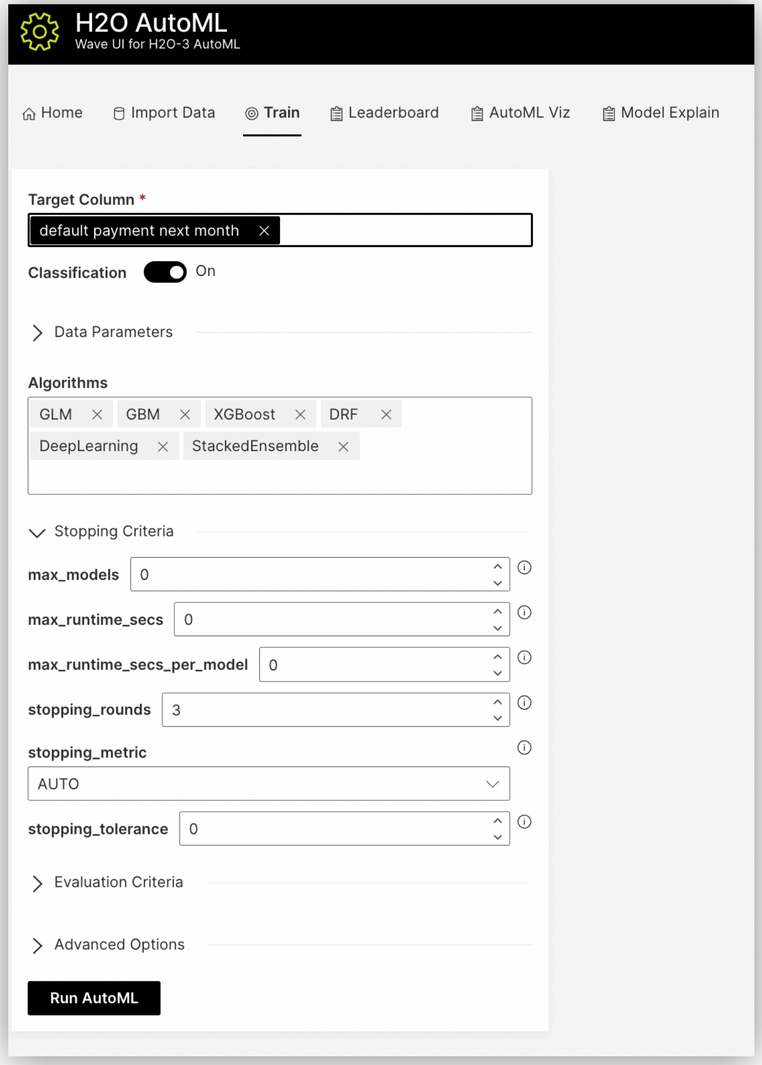
H2O AutoML Wave app

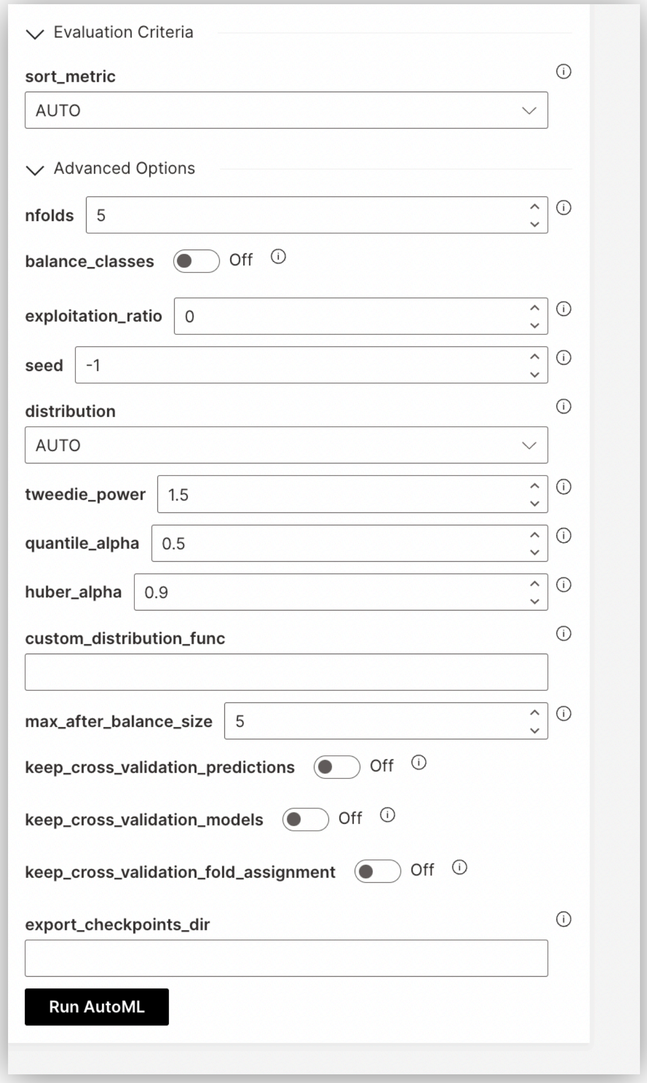


- *Modern GUI for H2O AutoML*
- *Import your data and Run AutoML*
- *View model metrics on a Leaderboard, and interactive Explainability plots*
- *Export any model*

https://github.com/h2oai/wave-h2o-automl

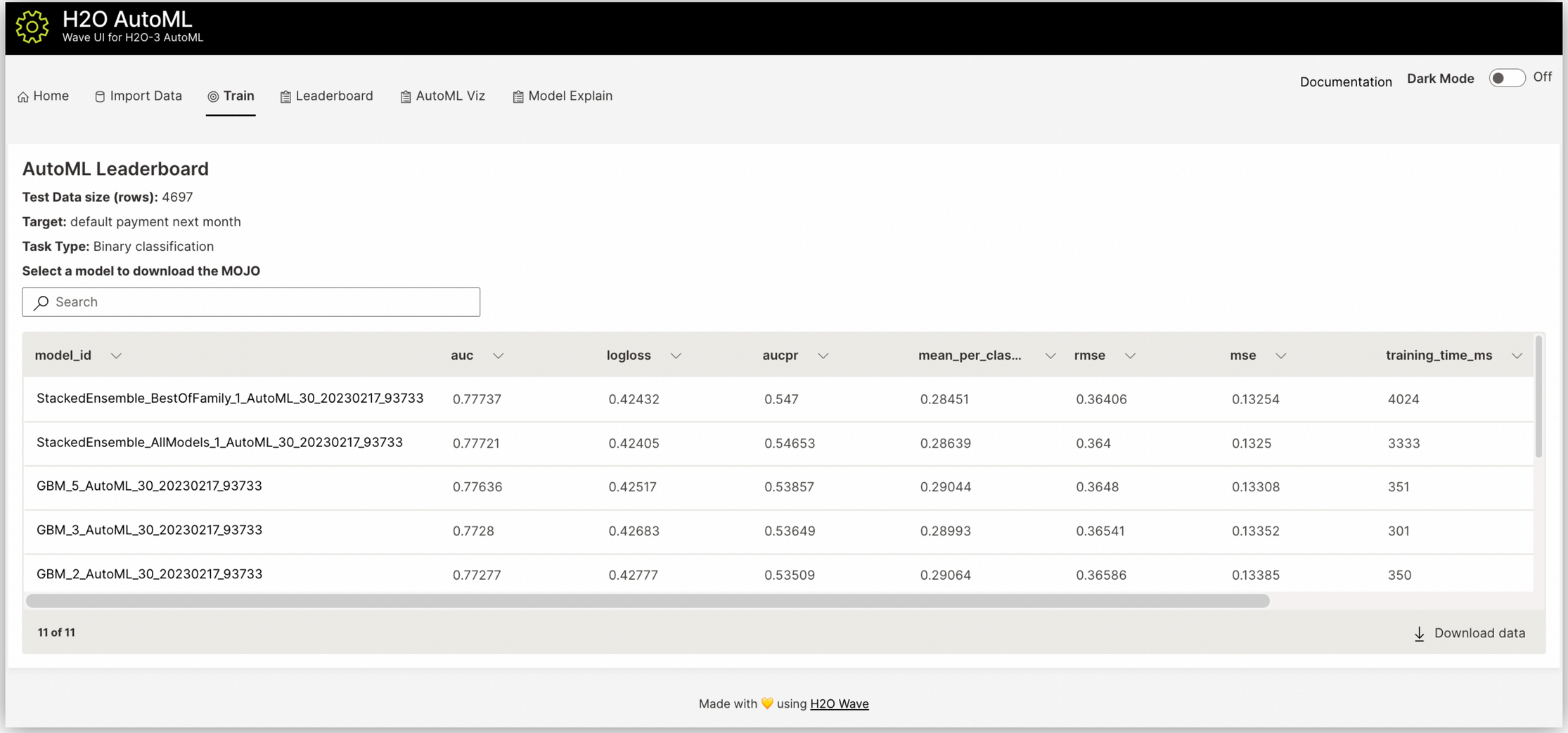
H2O AutoML Wave app



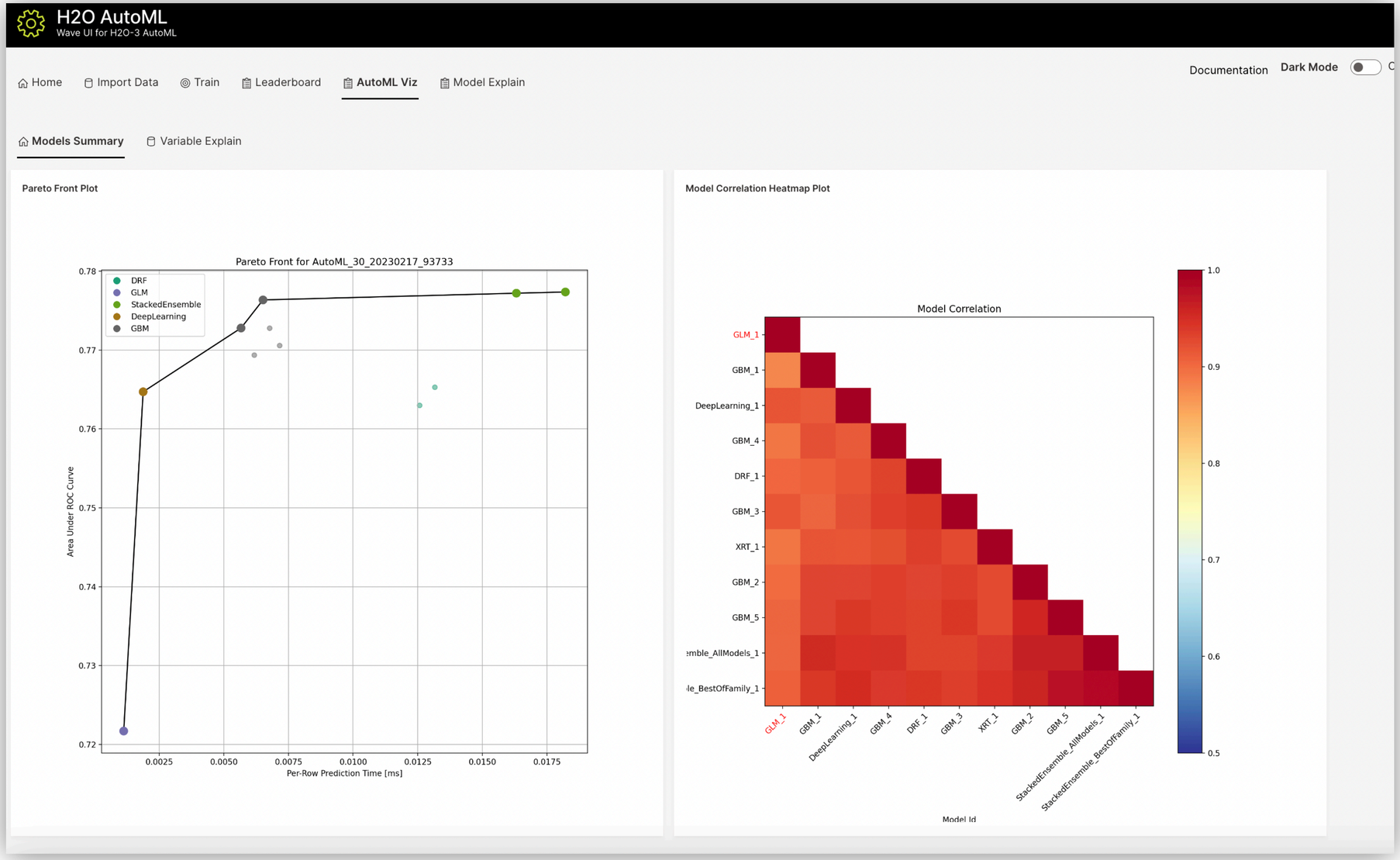


- *All the AutoML parameters are exposed to the user*
- *Specify how long you want AutoML to run (# of models or # seconds)*
- *Useful defaults, so you don't have to specify anything if you don't want to...*

H2O AutoML Wave app

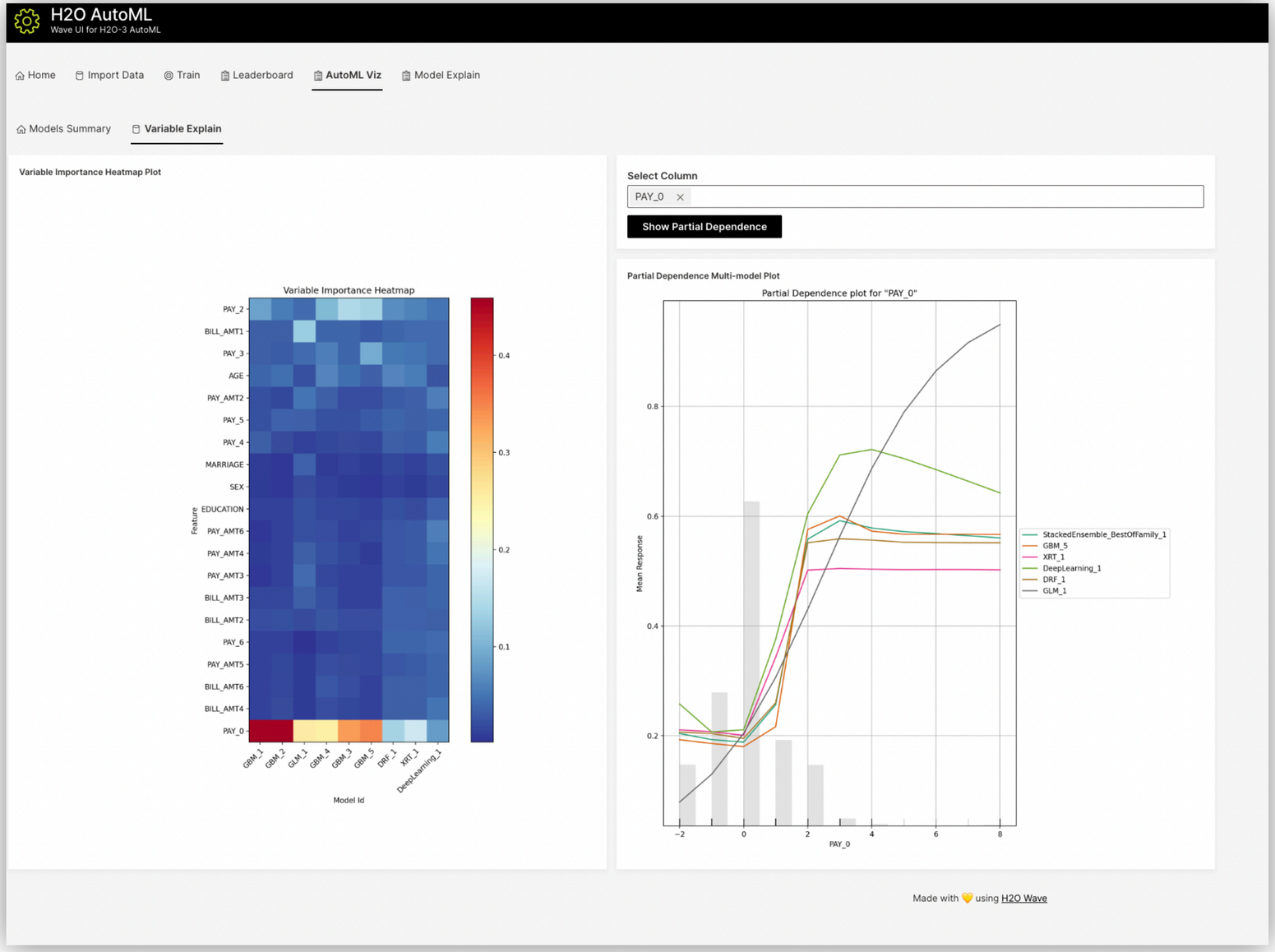


H2O AutoML Wave app



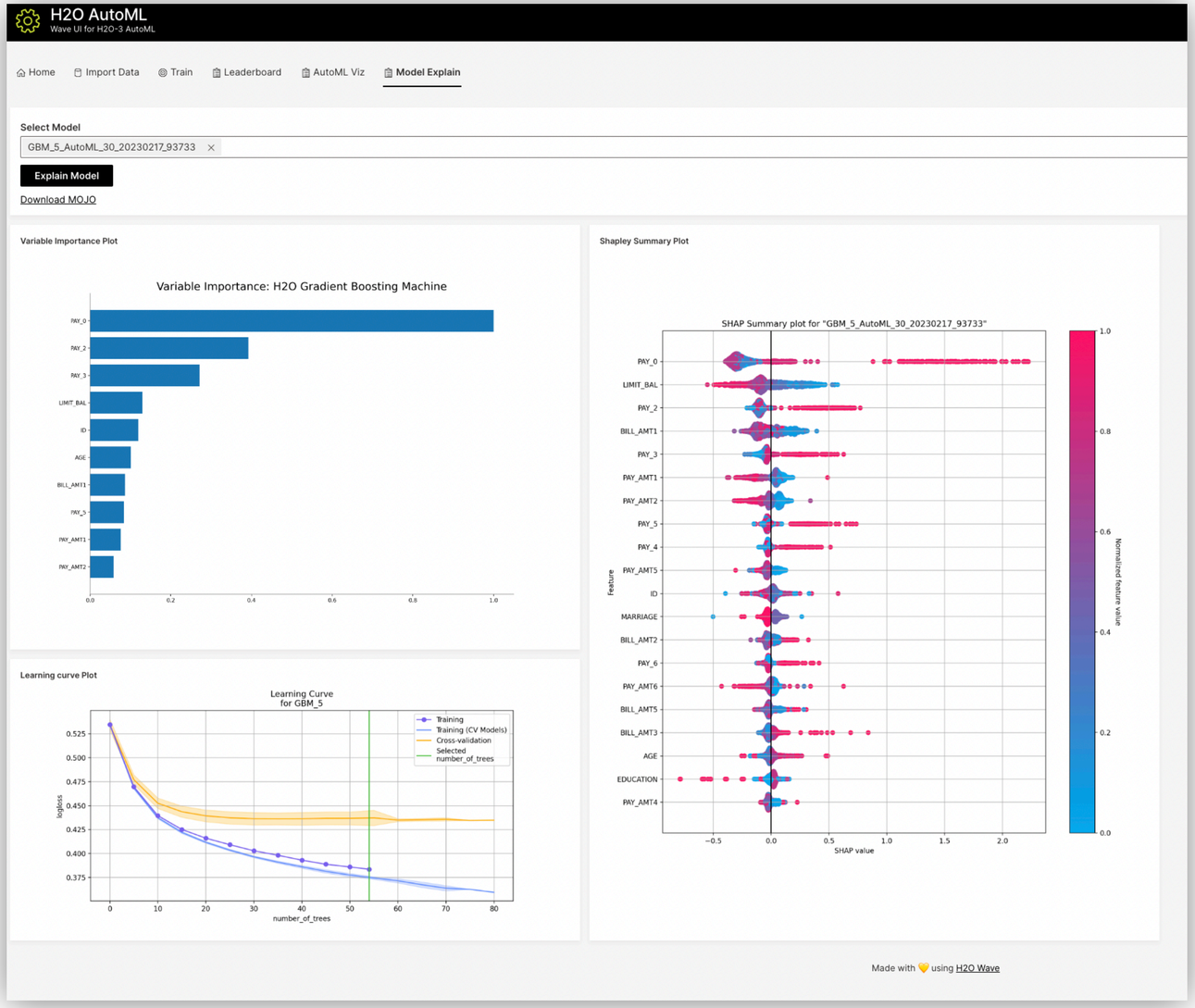
- *Pareto Front comparing model accuracy vs prediction speed for all models*
- *Model correlation plot*

H2O AutoML Wave app



- *Variable importance across all models*
- *Partial Dependence plot for any variable you choose (defaults to most important variable in the leader model). Shows PD for best model from each algorithm type.*

H2O AutoML Wave app



- *All the individual models can be explained with various visualizations*
- *Includes variable importance, learning curve, and SHAP summary plot*

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
- Events & Meetups: http://h2o.ai/events

Thank you!

@ledell on Github, Twitter

oss@ledell.org