Stacked Ensembles in H2O

February 2017

Erin LeDell Ph.D.
Machine Learning Scientist

H2O.ai

Agenda

- Who/What is H2O?
- Ensemble Learning Overview
- Stacking / Super Learner
- Why Stacking?
- Grid Search & Stacking
- Stacking with Third-party Algos
- AutoML and Stacking

H2O.ai

H2O.ai, the Company

- Founded in 2012
- Stanford & Purdue Math & Systems Engineers
- Headquarters: Mountain View, California, USA

H2O, the Platform

- Open Source Software (Apache 2.0 Licensed)
- R, Python, Scala, Java and Web Interfaces
- Distributed algorithms that scale to “Big Data”

Scientific Advisory Council

Dr. Trevor Hastie

- John A. Overdeck Professor of Mathematics, Stanford University
- PhD in Statistics, Stanford University
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Co-author with John Chambers, *Statistical Models in S*
- Co-author, *Generalized Additive Models*

Dr. Robert Tibshirani

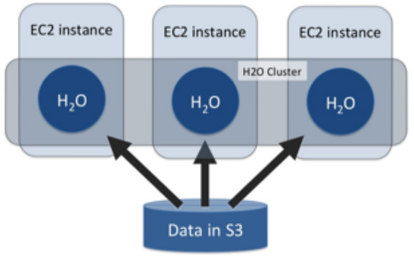
- Professor of Statistics and Health Research and Policy, Stanford University
- PhD in Statistics, Stanford University
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Author, *Regression Shrinkage and Selection via the Lasso*
- Co-author, *An Introduction to the Bootstrap*

Dr. Steven Boyd

- Professor of Electrical Engineering and Computer Science, Stanford University
- PhD in Electrical Engineering and Computer Science, UC Berkeley
- Co-author, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*
- Co-author, *Linear Matrix Inequalities in System and Control Theory*
- Co-author, *Convex Optimization*

H2O Distributed Computing

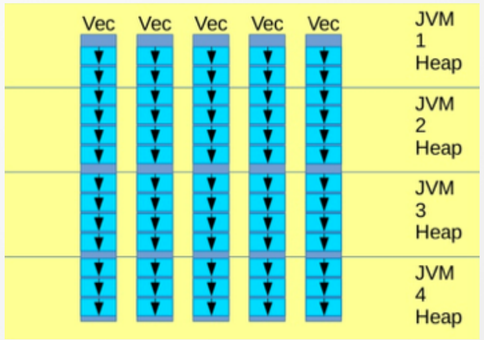
H2O Cluster



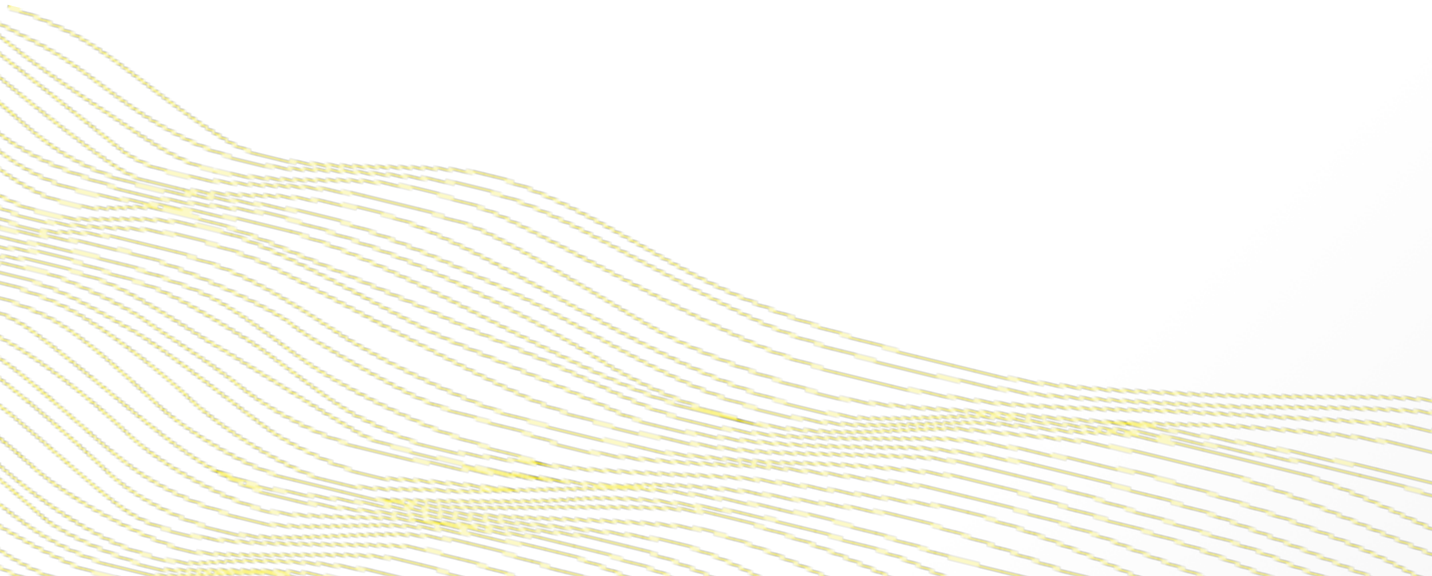
- Multi-node cluster with shared memory model.
- All computations in memory.
- Each node sees only some rows of the data.
- No limit on cluster size.

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



Introduction to Stacking



Ensemble Learning

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained by any of the constituent algorithms.

– Wikipedia

Common Types of Ensemble Methods

Bagging

- Reduces variance and increases accuracy
- Robust against outliers or noisy data
- Often used with Decision Trees (i.e. Random Forest)

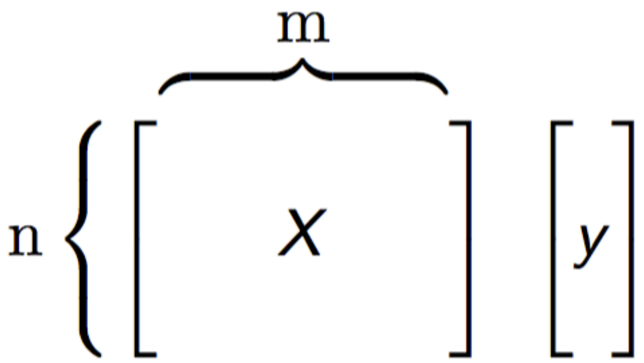
Boosting

- Also reduces variance and increases accuracy
- Not robust against outliers or noisy data
- Flexible – can be used with any loss function

Stacking

- Used to ensemble a diverse group of strong learners
- Involves training a second-level machine learning algorithm called a “metalearner” to learn the optimal combination of the base learners

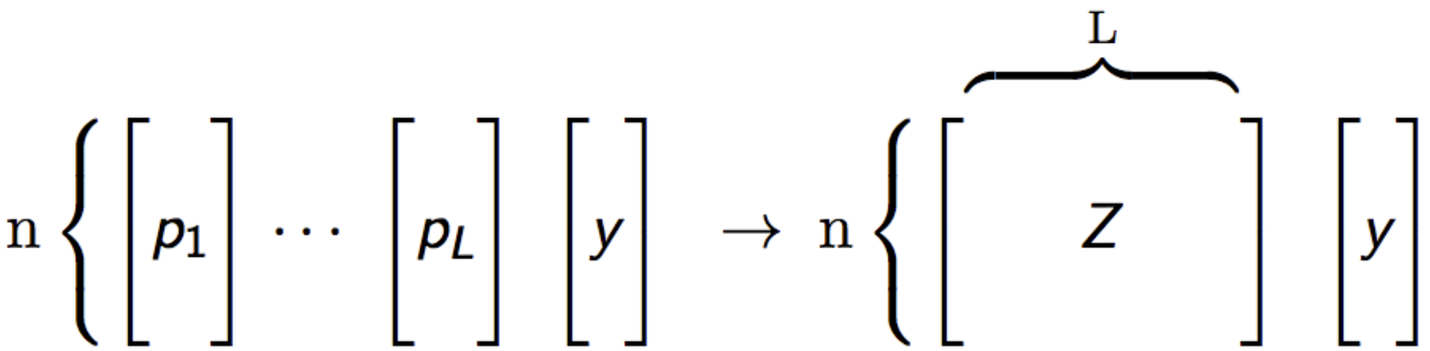
Stacking (aka Super Learner Algorithm)



"Level-zero" data

- Start with design matrix, X, and response, y
- Specify L base learners (with model params)
- Specify a metalearner (just another algorithm)
- Perform k-fold CV on each of the L learners

Stacking (aka Super Learner Algorithm)



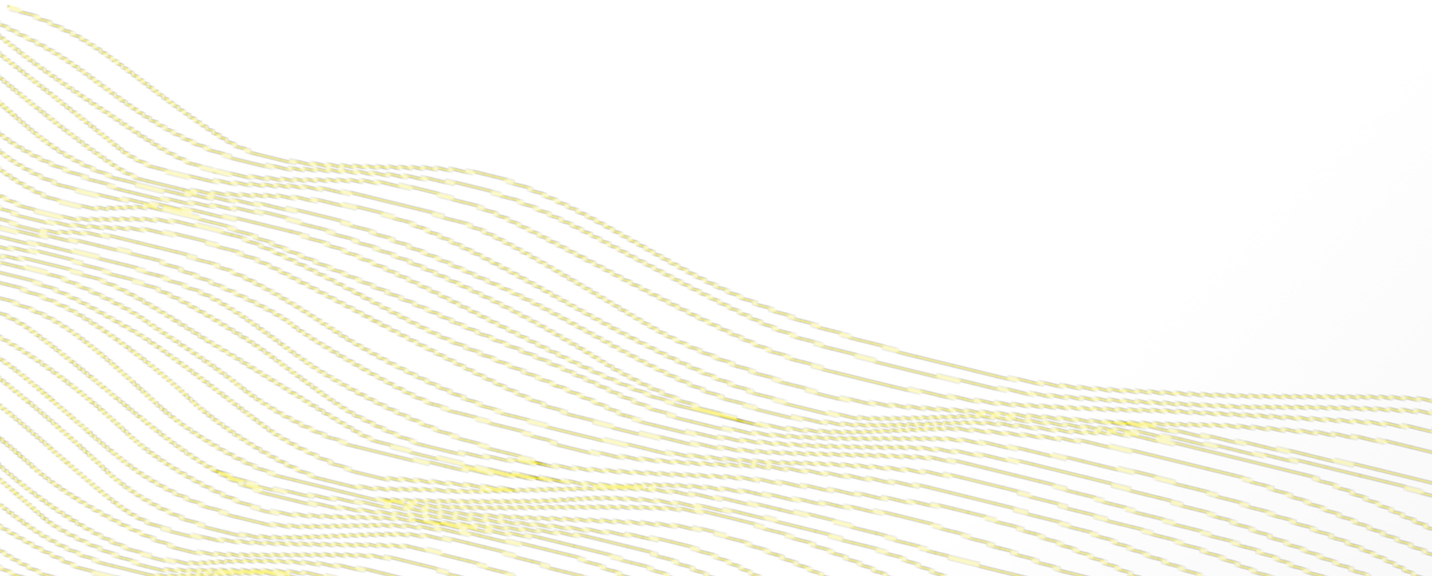
"Level-one" data

- Collect the predicted values from k-fold CV that was performed on each of the L base learners
- Column-bind these prediction vectors together to form a new design matrix, Z
- Train the metalearner using Z, y

Stacking vs. Parameter Tuning/Search

- A common task in machine learning is to perform model selection by specifying a number of models with different parameters.
- An example of this is Grid Search or Random Search.
- The first phase of the Super Learner algorithm is computationally equivalent to performing model selection via cross-validation.
- The latter phase of the Super Learner algorithm (the metalearning step) is just training another single model (no CV).
- With Stacking, your computation does not go to waste!

Why Stacked Ensembles?



How to Win Kaggle

#	Δrank	Team Name * in the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	↑1	Perfect Storm *	0.869558	128	Thu, 15 Dec 2011 05:35:00 (-3.2d)
2	↑4	Gxav *	0.869295	54	Thu, 15 Dec 2011 09:41:23 (-26.9h)
3	↑14	occupy *	0.869288	9	Thu, 20 Oct 2011 00:40:05
4	↑16	D'yakonov Alexander (MSU, Moscow, Russia)	0.869197	64	Thu, 15 Dec 2011 22:08:19 (-5.1d)
5	↓1	Indy Actuaries	0.869135	23	Thu, 15 Dec 2011 18:35:33 (-2.9d)
6	↑20	UCI_Combination	0.869097	19	Tue, 06 Dec 2011 06:41:59 (-3.5d)
7	↑42	vsh	0.869034	26	Thu, 15 Dec 2011 19:16:44
8	↓1	Xooma	0.868984	74	Thu, 15 Dec 2011 23:25:53 (-1.8h)
9	↓8	vsu	0.868942	14	Thu, 15 Dec 2011 14:02:51 (-0h)
10	↑12	cointegral	0.868913	2	Mon, 21 Nov 2011 12:24:20

https://www.kaggle.com/c/GiveMeSomeCredit/leaderboard/private

How to Win Kaggle

The big learning experience for me is how strong a team can be if the skills of its members complement each other. Rather like an ensemble in fact. None of us would have got in the top placings as individuals.

What we basically did was extract about 25-35 features from the original dataset, and applied an ensemble of five different methods; a regression random forest, a classification random forest, a feed-forward neural network with a single hidden layer, a gradient regression tree boosting algorithm, and a gradient classification tree boosting algorithm. The neural network was a pain to implement properly but improved things by a decent amount over the bagging and boosting based elements.

#17 | Posted 3 years ago

Alec Stephenson

Competition **1st** | Overall **642nd**
Posts **82** | Votes **55**
Joined **1 Sep '10** | Email User

Permalink

https://www.kaggle.com/c/GiveMeSomeCredit/forums/t/1166/congratulations-to-the-winners/7229#post7229

How to Win Kaggle

I used an ensemble of 15 models including GBMs, weighted GBMs, Random Forest, balanced Random Forest, GAM, weighted GAM (all with bernoulli/binomial error), SVM and bagged ensemble of SVMs.

I haven't try to fine tune each models individually but looked for diversity of fits.

My best score (0.89345, not in the private leaderboard as I haven't selected it in my final set) was an ensemble of 11 models which excluded the SVMs fits.

#18 | Posted 3 years ago

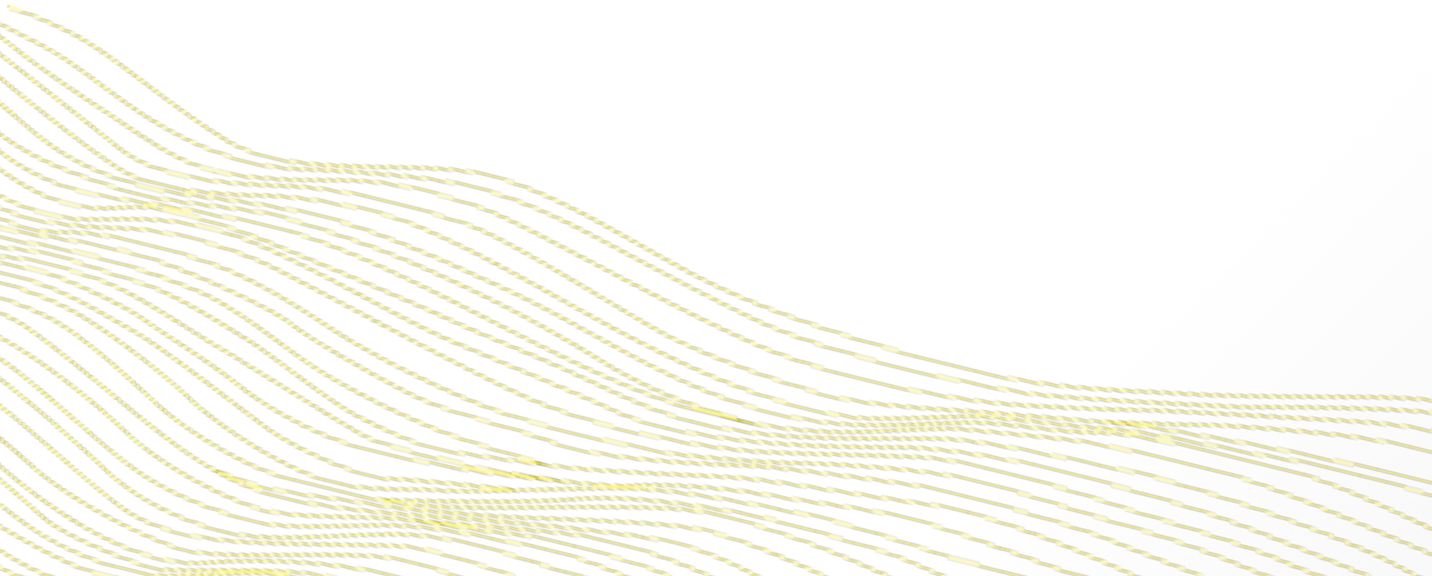
Xavier Conort

Competition **2nd** | Overall **33rd**
Posts **49** | Votes **94**
Joined **23 Sep '11** | Email User

Permalink

https://www.kaggle.com/c/GiveMeSomeCredit/forums/t/1166/congratulations-to-the-winners/7230#post7230

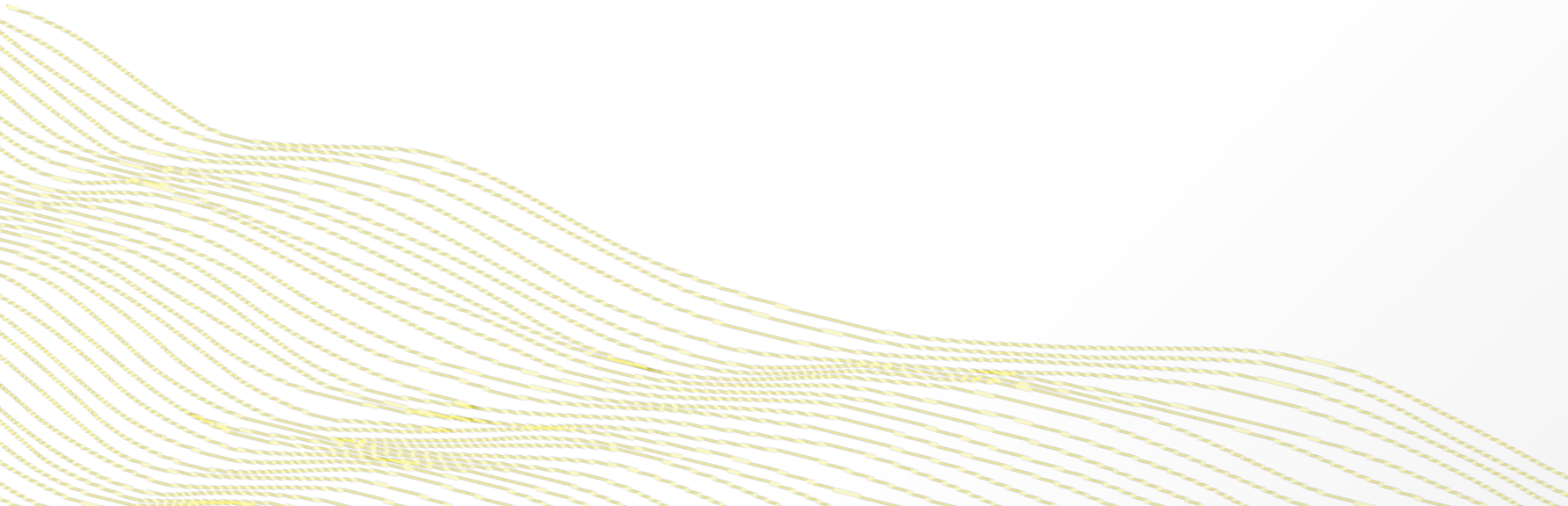
h2oEnsemble R package & Stacked Ensemble in h2o



Evolution of H2O Ensemble

- h2oEnsemble R package in 2015
- Ensemble logic ported to Java in late 2016
- Stacked Ensemble method in h2o in early 2017
  - R & Python APIs supported
  - In progress: Custom metalearners
  - In progress: MOJO for production use

Stacking with Random Grids



H2O Cartesian Grid Search

Example

```
hidden_opt <- list(c(200,200), c(100,300,100), c(500,500))
l1_opt <- c(1e-5,1e-7)
hyper_params <- list(hidden = hidden_opt, l1 = l1_opt)

grid <- h2o.grid(algorithm = "deeplearning",
                 hyper_params = hyper_params,
                 x = x, y = y,
                 training_frame = train,
                 validation_frame = valid)
```

H2O Random Grid Search

Example

```
search_criteria <- list(strategy = "RandomDiscrete",
                        max_runtime_secs = 600)

grid <- h2o.grid(algorithm = "deeplearning",
                 hyper_params = hyper_params,
                 search_criteria = search_criteria,
                 x = x, y = y,
                 training_frame = train,
                 validation_frame = valid)
```

Stacking with Random Grids (h2o R)

Example

```
# Create a list of all the base models
models <- c(gbm_models, rf_models, dl_models, glm_models)

# Let's stack!
fit <- h2o.stackedEnsemble(x = x, y = y,
                           selection_strategy="choose_all",
                           training_frame = train,
                           base_models = models)
```

Stacking with Random Grids (h2o Python)

Example

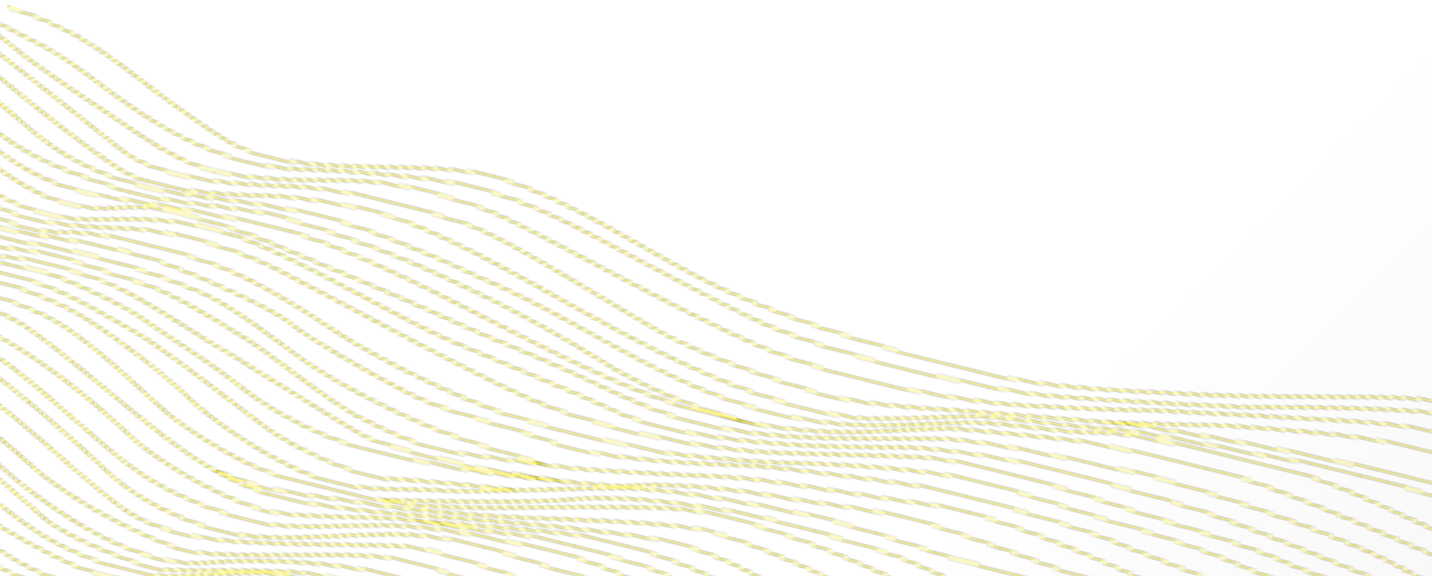
```
from h2o.estimators.stackedensemble \
 import H2OStackedEnsembleEstimator

# Let's stack!
stack = H2OStackedEnsembleEstimator( \
                selection_strategy="choose_all", \
                base_models=models)

stack.train(y=y, training_frame=train)
```

H2O Stacking Resources

H2O Stacked Ensembles docs & code demo:
http://tinyurl.com/h2o-stacked-ensembles

h2oEnsemble R package homepage on Github:
http://tinyurl.com/github-h2o-ensemble

Third-Party Integrations



Ensemble H2O with Anything

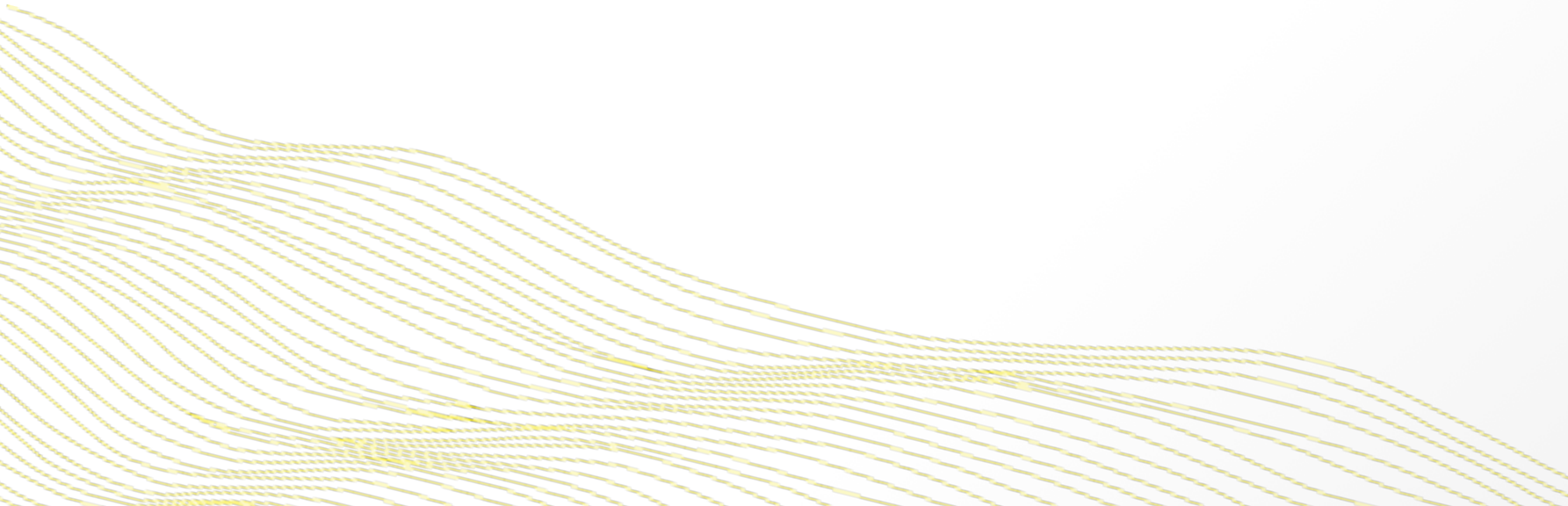
A powerful combo: H2O + XGBoost

- XGBoost will be available in the next major release of H2O, so you can use it with the Stacked Ensemble method
- https://github.com/h2oai/h2o-3/pull/699

Third party stacking with H2O:

- SuperLearner, subsemble, mlr & caret R packages support stacking with H2O for small/medium data

AutoML



H2O AutoML

- AutoML stands for “Automatic Machine Learning”
- The idea here is to remove most (or all) of the parameters from the algorithm, as well as automatically generate derived features that will aid in learning.
- Single algorithms are tuned automatically using a carefully constructed random grid search.
- Optionally, a Stacked Ensemble can be constructed.

Public code coming soon!

H2O Resources

- H2O Online Training: http://learn.h2o.ai
- H2O Tutorials: https://github.com/h2oai/h2o-tutorials
- H2O Meetup Materials: https://github.com/h2oai/h2o-meetups
- H2O Video Presentations: https://www.youtube.com/user/0xdata
- H2O Community Events & Meetups: https://h2o.ai/community

Thank you!

@ledell on Github, Twitter

erin@h2o.ai

http://www.stat.berkeley.edu/~ledell