

H2O AutoML: Scalable Automatic Machine Learning



NYC Big Data Science Meetup

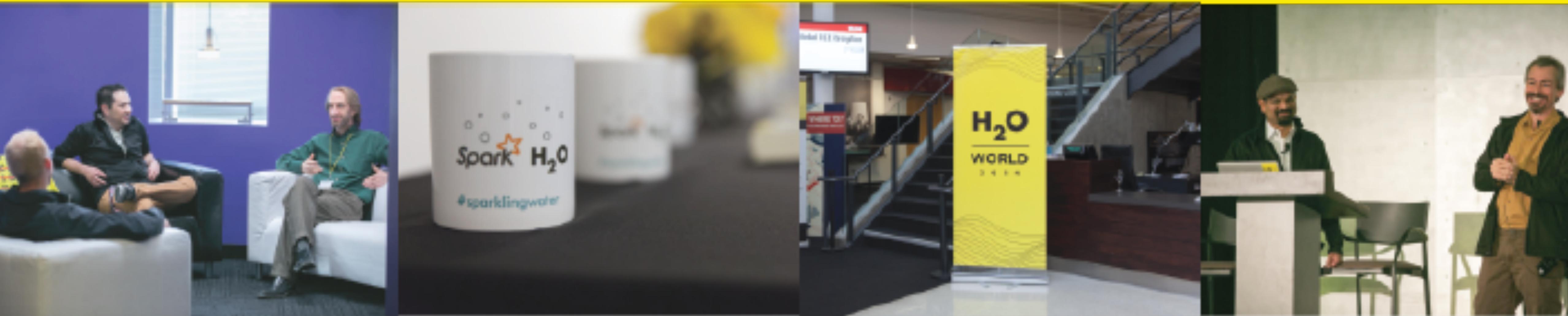
June 2017

H₂O.ai

Erin LeDell Ph.D.
H2O.ai

Introduction

- Machine Learning Scientist at H2O.ai, in Mountain View, California, USA
- Ph.D. in Biostatistics with Designated Emphasis in Computational Science and Engineering from UC Berkeley (focus on Machine Learning)
- Worked as a data scientist at several startups



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*

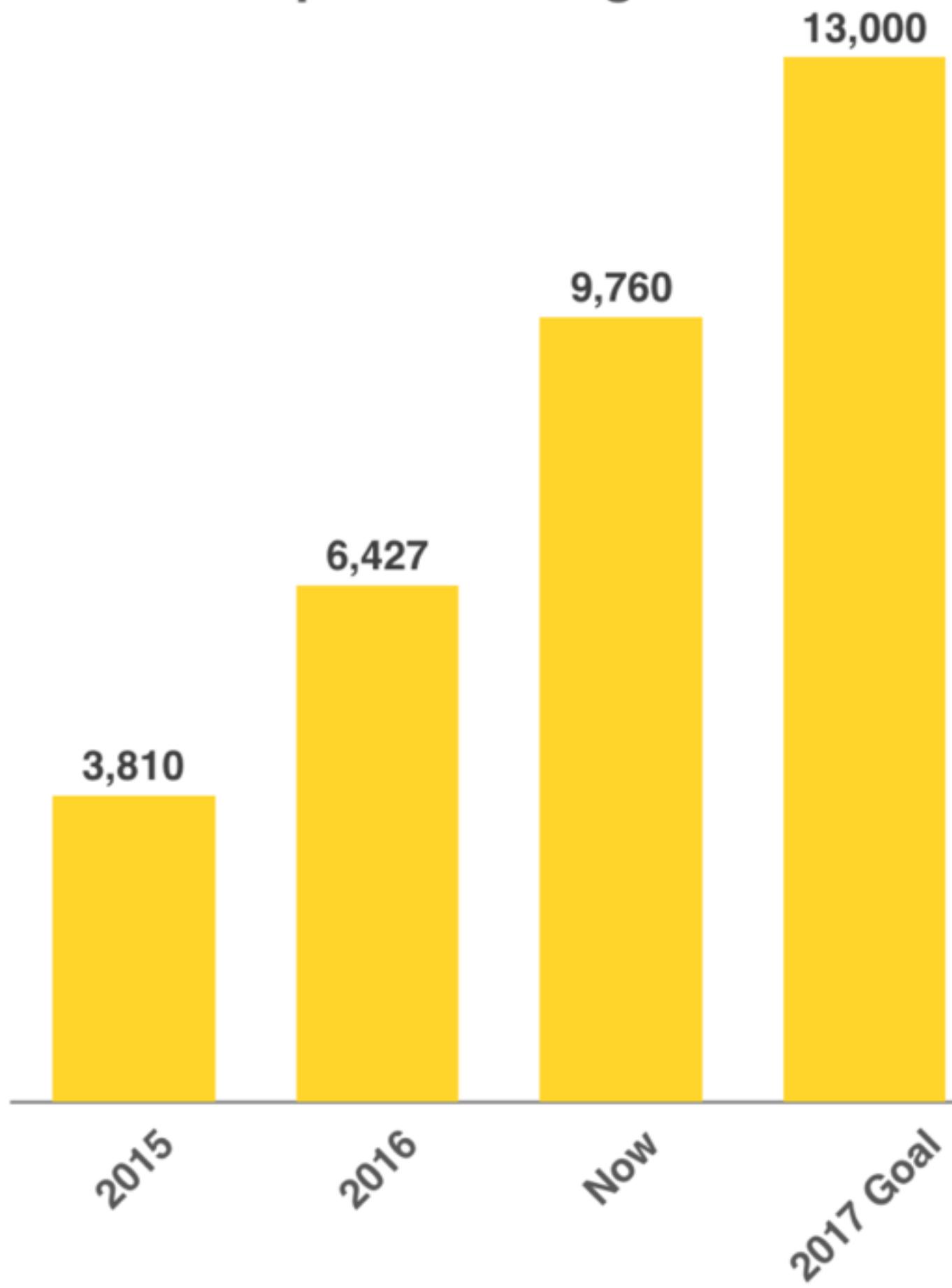


Dr. Steven Boyd

- 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 User Community

Companies Using H2O.ai



169 OF THE **500** FORTUNE

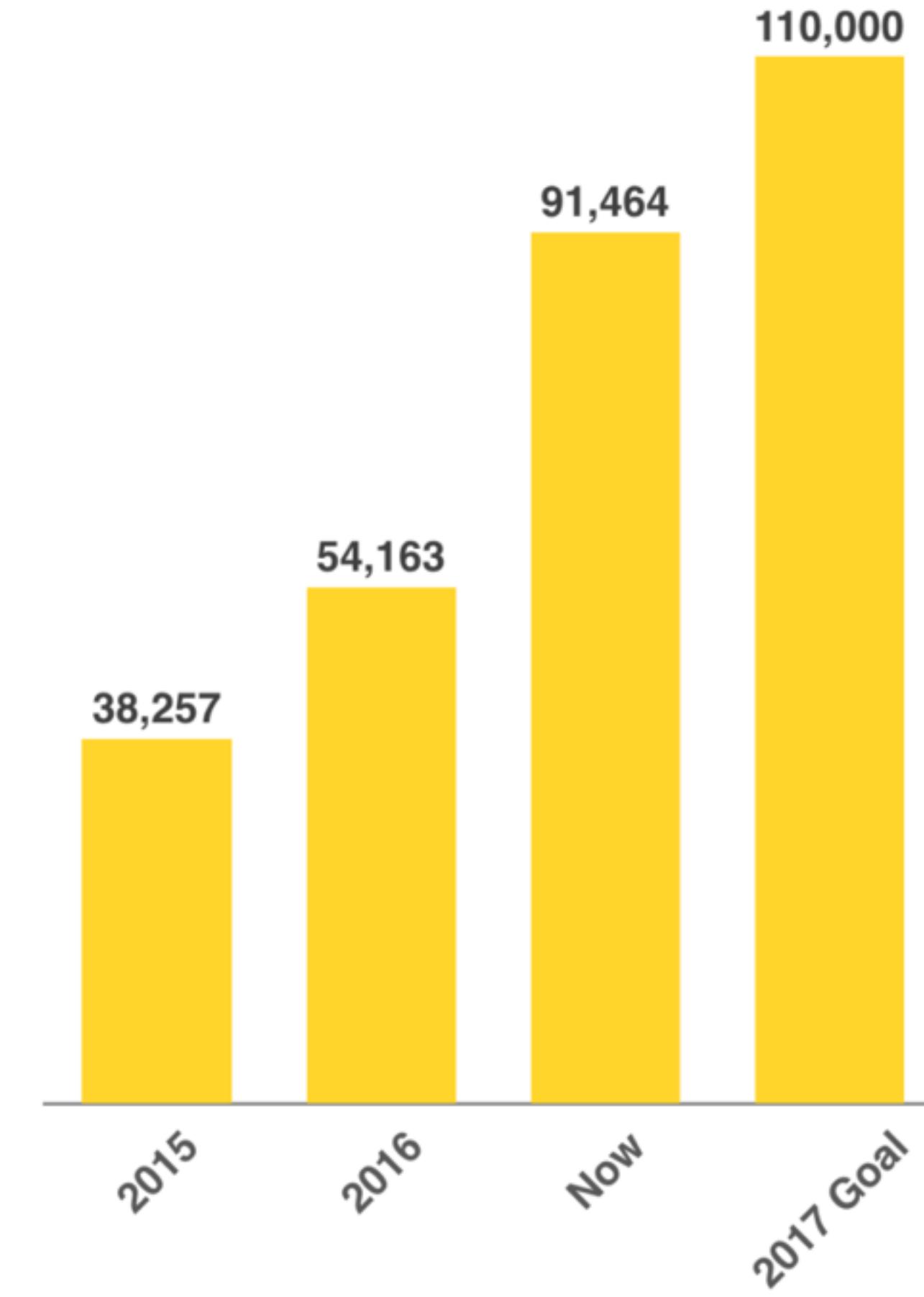


8 OF TOP 10 BANKS

7 OF TOP 10 INSURANCE COMPANIES

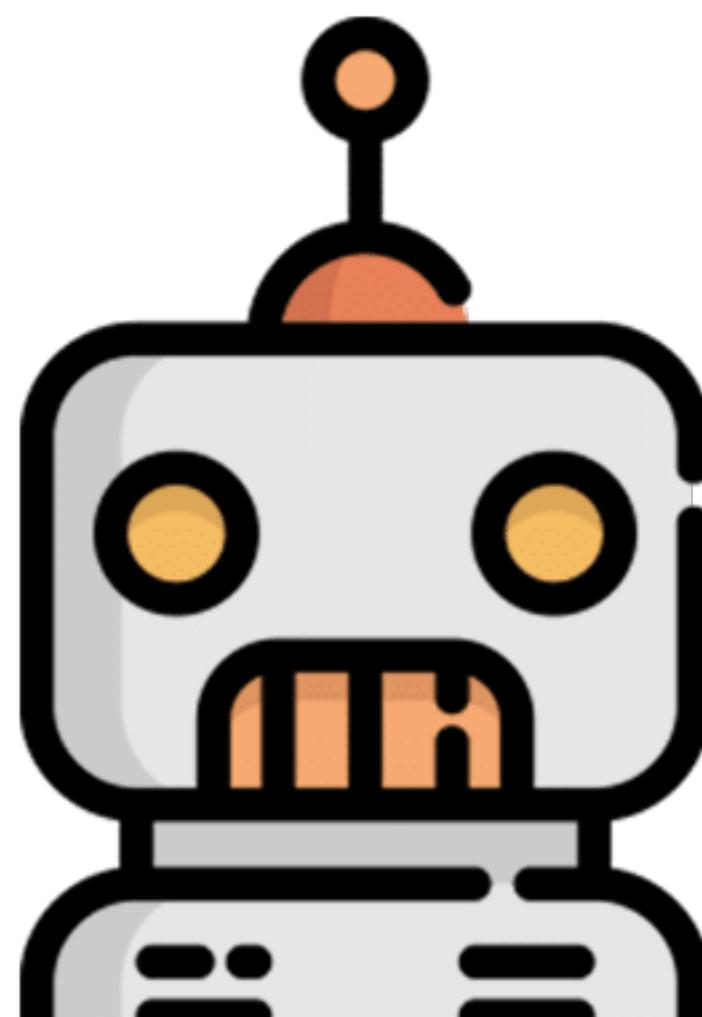
4 OF TOP 10 HEALTHCARE COMPANIES

H2O.ai Users

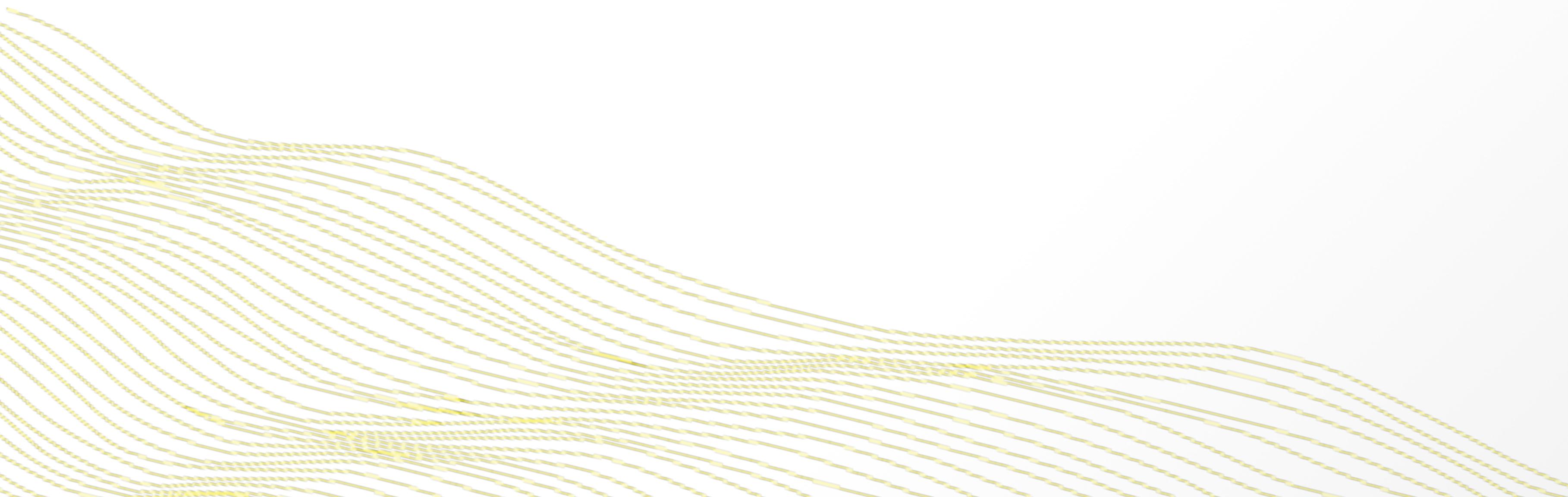


Agenda

- H2O Platform Overview
- Intro to Automatic Machine Learning (AutoML)
- Bayesian Hyperparameter Optimization
- Random Grid Search
- Stacked Ensembles & Ensemble Selection
- H2O's AutoML



H2O Platform



H2O Platform Overview

- Distributed implementations of cutting edge ML algorithms.
- Core algorithms written in high performance Java.
- APIs available in R, Python, Scala, REST/JSON.
- Interactive Web GUI called H2O Flow.
- Easily deploy models to production with H2O Steam.



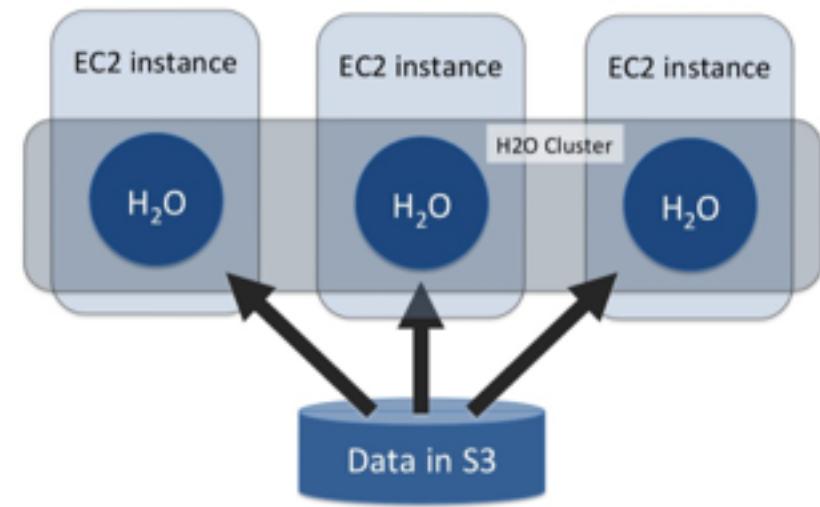
H2O Platform Overview

- Write code in high-level language like R (or use the web GUI) and output production-ready models in Java.
- To scale, just add nodes to your H2O cluster.
- Works with Hadoop, Spark and your laptop.



H2O Distributed Computing

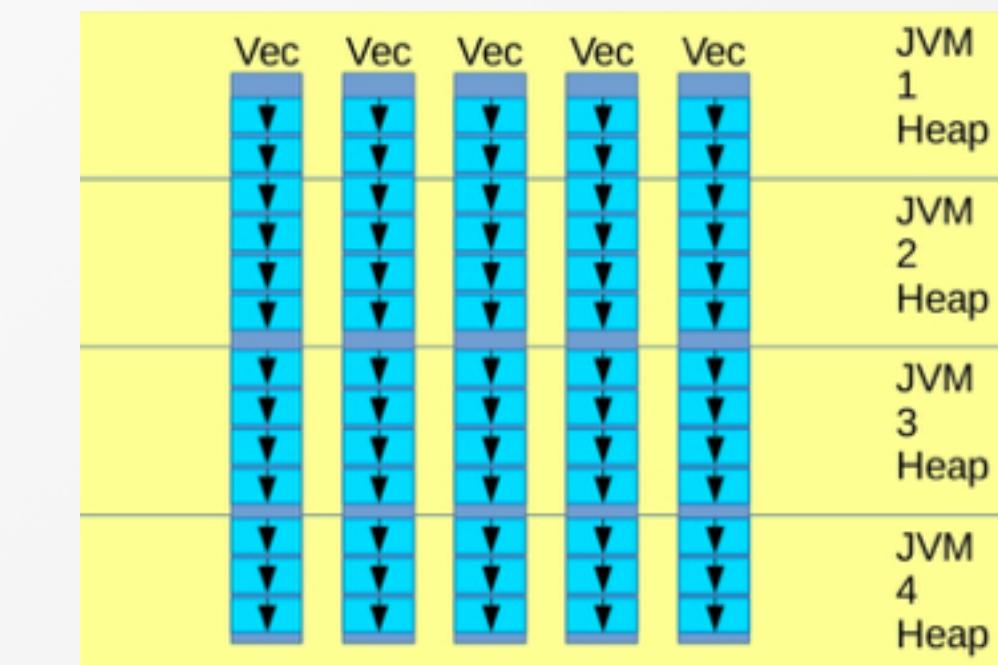
H2O Cluster



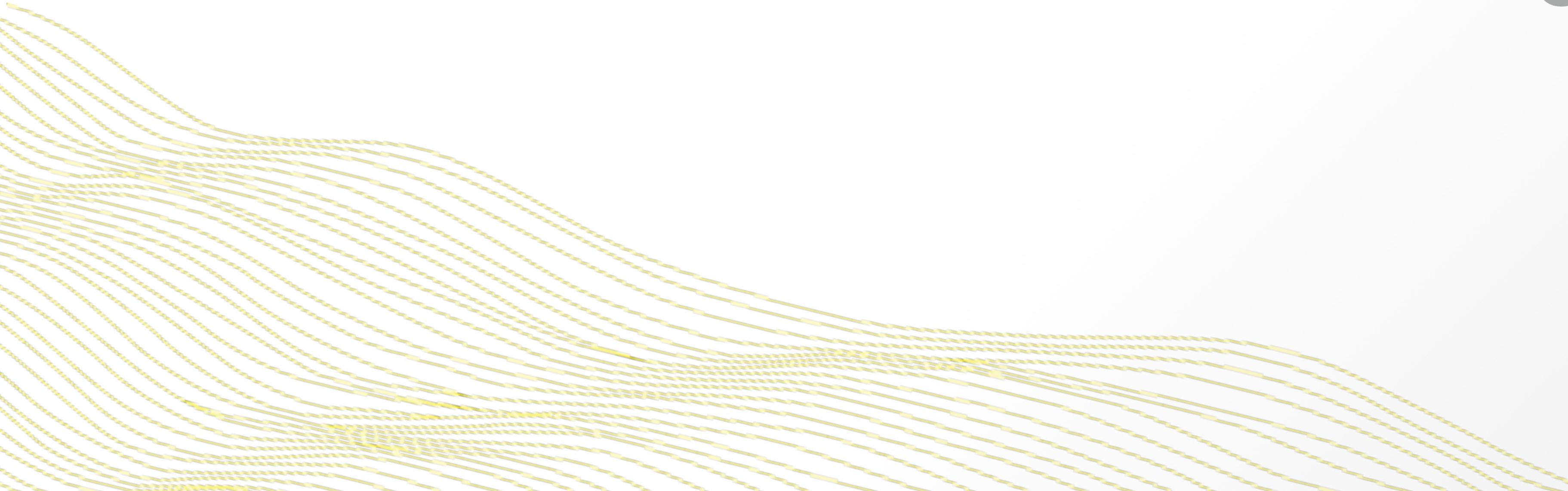
- Multi-node cluster with shared memory model.
- All computations in memory.
- Each node sees only some rows of the data.
- No limit on cluster size.

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`



Intro to Automatic Machine Learning



Aspects of Automatic Machine Learning

Data Preprocessing

- Imputation, one-hot encoding, standardization
 - Feature selection and/or feature extraction (e.g. PCA)
 - Count/Label/Target encoding of categorical features
-

Model Generation

- Cartesian grid search or random grid search
 - Bayesian Hyperparameter Optimization
 - Individual models can be tuned using a validation set
-

Ensembles

- Ensembles often out-perform individual models
- Stacking / Super Learning (Wolpert, Breiman)
- Ensemble Selection (Caruana)

Bayesian Optimization of Hyperparameters



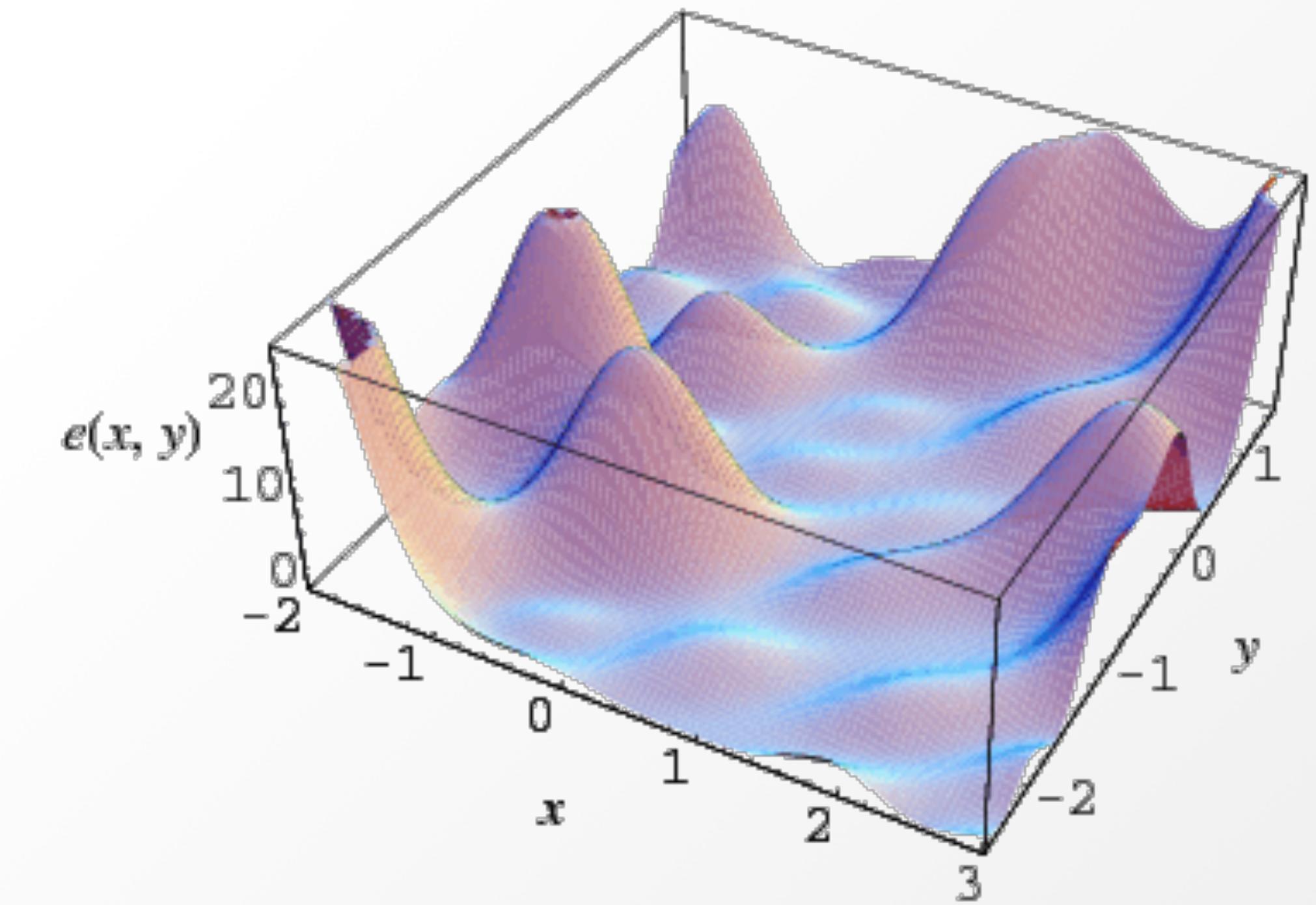
Bayesian Optimization

- Bayesian Hyperparameter Optimization consists of developing a statistical model of the function mapping hyperparameter values to the objective (e.g. AUC, MSE), evaluated on a validation set.
- Different approaches based on: Gaussian Processes, Tree Structured Parzen Estimator, Random Forest

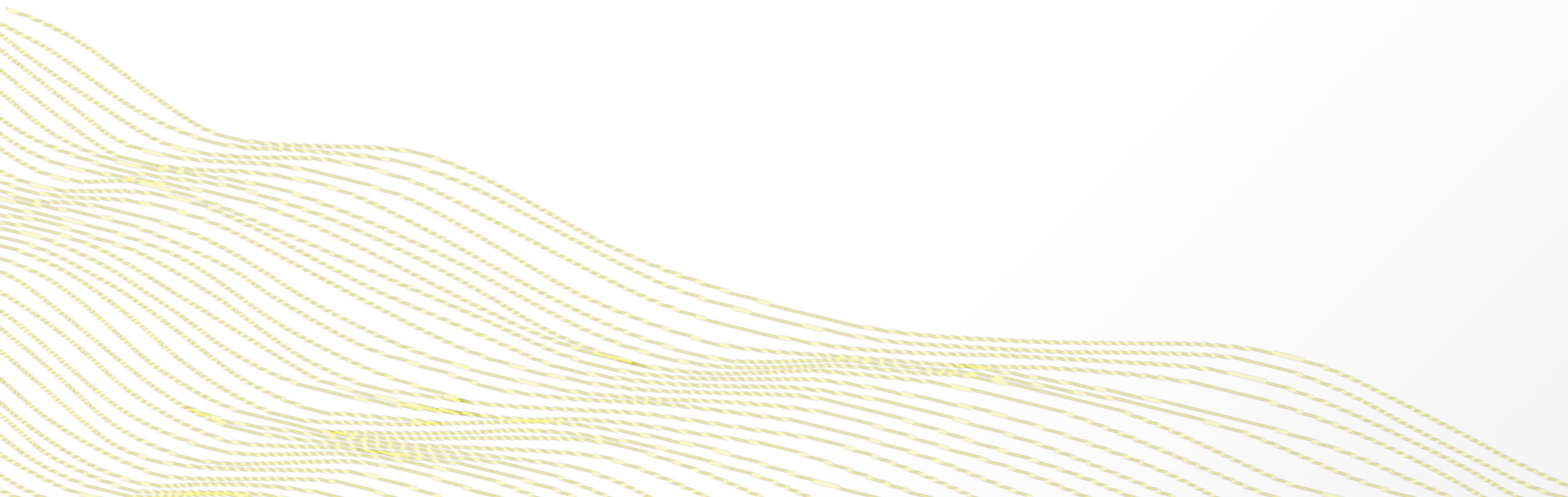
AKA “Sequential Model-based Optimization (SMBO)”

Hyperparameter Optimization Software

- SigOpt (SaaS)
- MOE (C++/Python OSS)
- Scikit-Optimize (Python OSS)
- SMAC (Java OSS)
- Hyperopt (Python OSS)
- Spearmint (Python non-commercial OSS)



Random Grid Search & Stacked Ensembles



Random Grid Search & Stacking

- In Bengio & Bergstra's 2012 JMLR paper, "Random Search for Hyper-Parameter Optimization", it shows that Random Search is far more efficient than a full (aka Cartesian) Grid Search
- 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 (aka Super Learner Algorithm)

$$n \left\{ \begin{bmatrix} x \\ \vdots \\ x \end{bmatrix} \right\} \begin{bmatrix} y \\ \vdots \\ y \end{bmatrix}$$

“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)

$$n \left\{ \begin{bmatrix} p_1 \\ \vdots \\ p_L \end{bmatrix} \cdots \begin{bmatrix} p_1 \\ \vdots \\ p_L \end{bmatrix} \begin{bmatrix} y \end{bmatrix} \right\} \rightarrow n \left\{ \underbrace{\begin{bmatrix} \quad & \quad & \quad \\ \quad & \quad & \quad \\ z & & \end{bmatrix}}_L \begin{bmatrix} y \end{bmatrix} \right\}$$

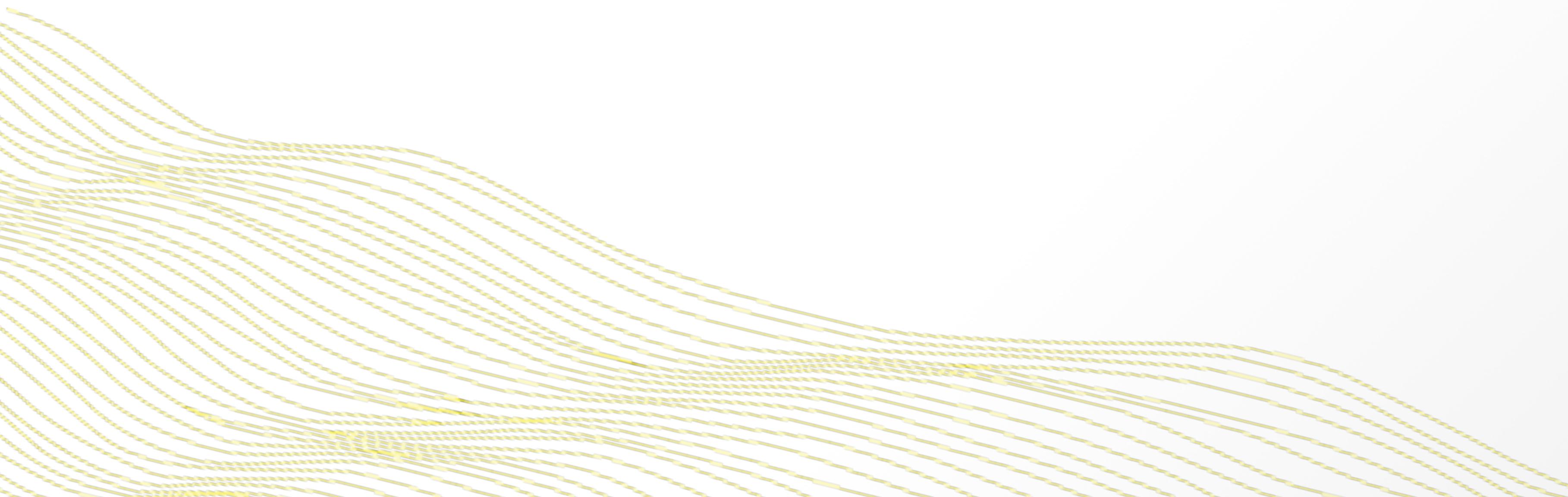
"Level-one"
data

- 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 Ensemble Selection

- Stacking uses all the given models (good and bad) and uses a second-level metalearning algorithm to find the optimal combination of base learners.
- With Ensemble Selection, rather than combine good and bad models in an ensemble, forward stepwise selection is used to find a subset of models that, when averaged together, yield the best performance.

H2O AutoML



H2O AutoML

- Very simple interface to define the data, response column and stopping strategy.
- Basic data pre-processing (as in all H2O algos).
- Trains a random grid of GBMs, DNNs, GLMs, etc. using a carefully chosen parameter space; individual models are tuned using a validation set.
- A Stacked Ensemble is trained using all models.
- Returns a sorted “Leaderboard” of all models.

Available on the Bleeding Edge release of H2O!

H2O AutoML Leaderboard

model_id	auc	logloss
StackedEnsemble_0_AutoML_20170605_212658	0.776164	0.564872
GBM_grid_0_AutoML_20170605_212658_model_2	0.75355	0.587546
DRF_0_AutoML_20170605_212658	0.738885	0.611997
GBM_grid_0_AutoML_20170605_212658_model_0	0.735078	0.630062
GBM_grid_0_AutoML_20170605_212658_model_1	0.730645	0.67458
XRT_0_AutoML_20170605_212658	0.728358	0.629296
GLM_grid_0_AutoML_20170605_212658_model_1	0.685216	0.635137
GLM_grid_0_AutoML_20170605_212658_model_0	0.685216	0.635137

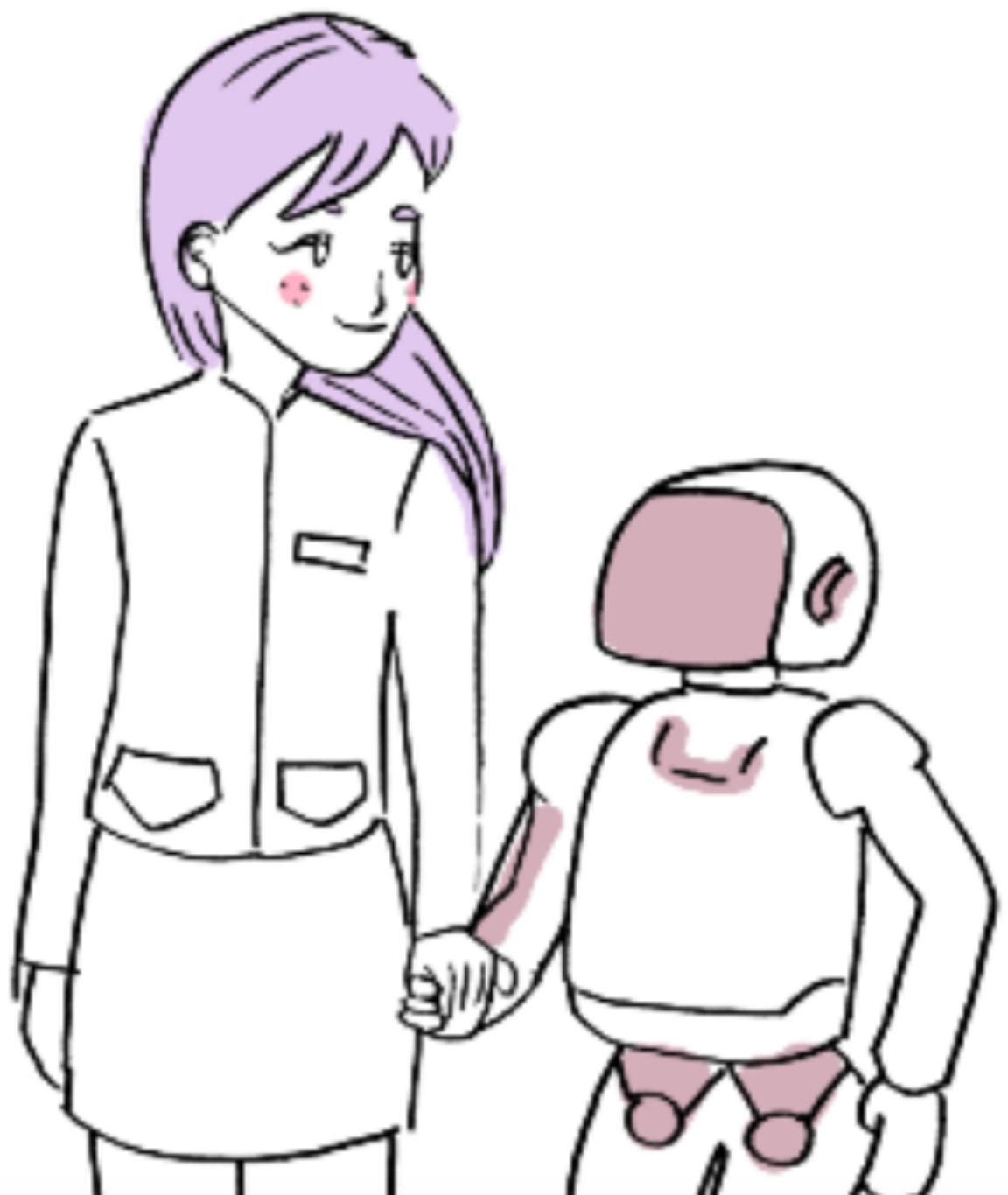
Example Leaderboard for binary classification

H2O AutoML: Active Development

- In Development: Adding third-party algorithms to the AutoML process.
 - XGBoost
 - TensorFlow, Caffe, MXNet via H2O Deep Water
- Currently does not include any automated feature engineering, but may add in the future.
- More info at [AutoML JIRA: Open & In-progress tickets](#)

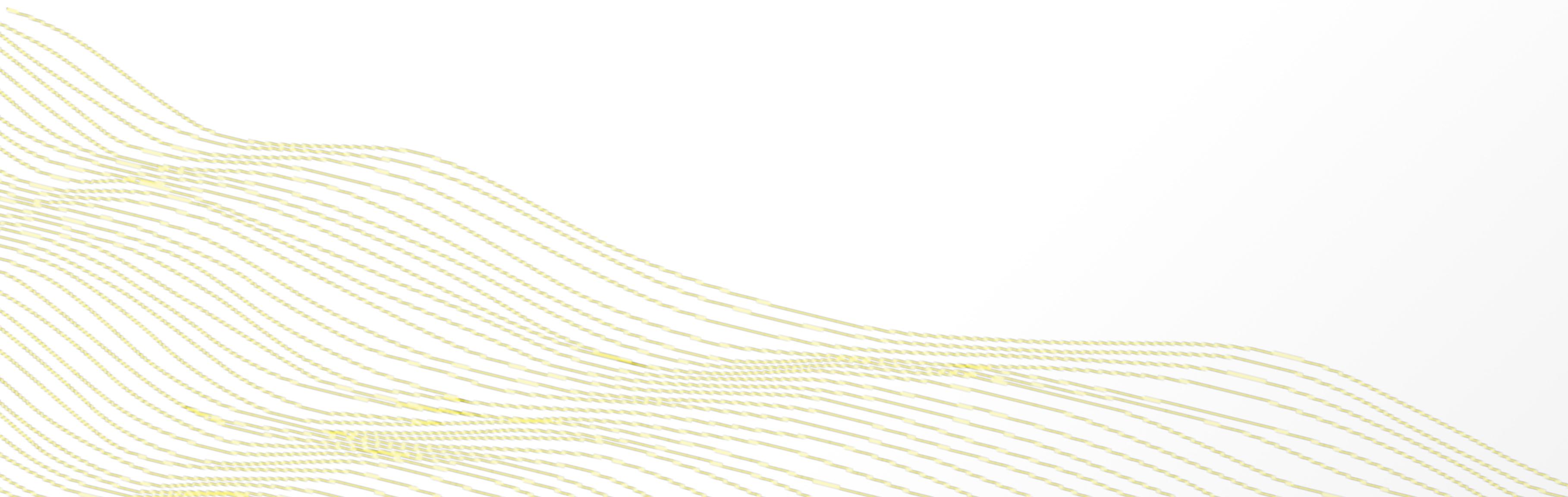
H2O AutoML R & Python Demos

- Documentation, R and Python code examples are available in the H2O User Guide at <http://docs.h2o.ai>.
- Live demo time!



```
aml <- h2o.automl(x = x, y = y,  
                   training_frame = train,  
                   leaderboard_frame = test,  
                   max_runtime_secs = 30)
```

Appendix



H2O Deep Water

- Native implementation of Deep Learning models for GPU-optimized backends (mxnet, Caffe, TensorFlow, etc.)
- Easy to use interface from H2O to use state-of-the-art Deep Learning models
 - Extends the H2O platform to include Convolutional Neural Nets (CNNs) and Recurrent Neural Nets (RNNs) including LSTMs

<https://github.com/h2oai/deepwater>

H2O Resources

- Documentation: <http://docs.h2o.ai>
- Tutorials: <https://github.com/h2oai/h2o-tutorials>
- Slidedecks: <https://github.com/h2oai/h2o-meetups>
- Video Presentations: <https://www.youtube.com/user/0xdata>
- Events & Meetups: <http://h2o.ai/events>



Thank you!

@ledell on Github, Twitter
erin@h2o.ai

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