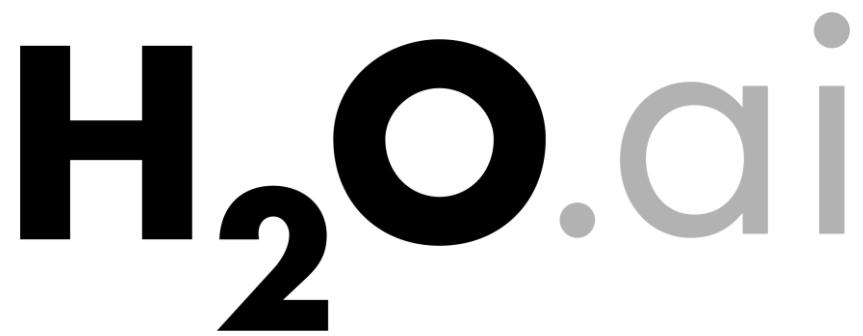


Latest Developments in H₂O



Jo-fai (Joe) Chow

Data Scientist

joe@h2o.ai

@matlabulous

Data Science for IoT Meetup – London
9th March, 2017

What's New

- From Kaggle to H₂O
 - Joe's Story
- ODSC Masterclass Summit
 - Training Materials on GitHub
- Stacked Ensembles
- Automatic Machine Learning
- Time Series
- Sparkling Water
- Deep Water
- H₂O + xgboost
- Community Events
- Discussions
 - H₂O on ARM
- Other News

From Kaggle to H₂O

The true story of a civil engineer turned data geek

About Me

- Civil (Water) Engineer
2010 – 2015
 - Consultant (UK)
 - Utilities
 - Asset Management
 - Constrained Optimization
 - Industrial PhD (UK)
 - Infrastructure Design Optimization
 - Machine Learning + Water Engineering
 - **Discovered H₂O in 2014**

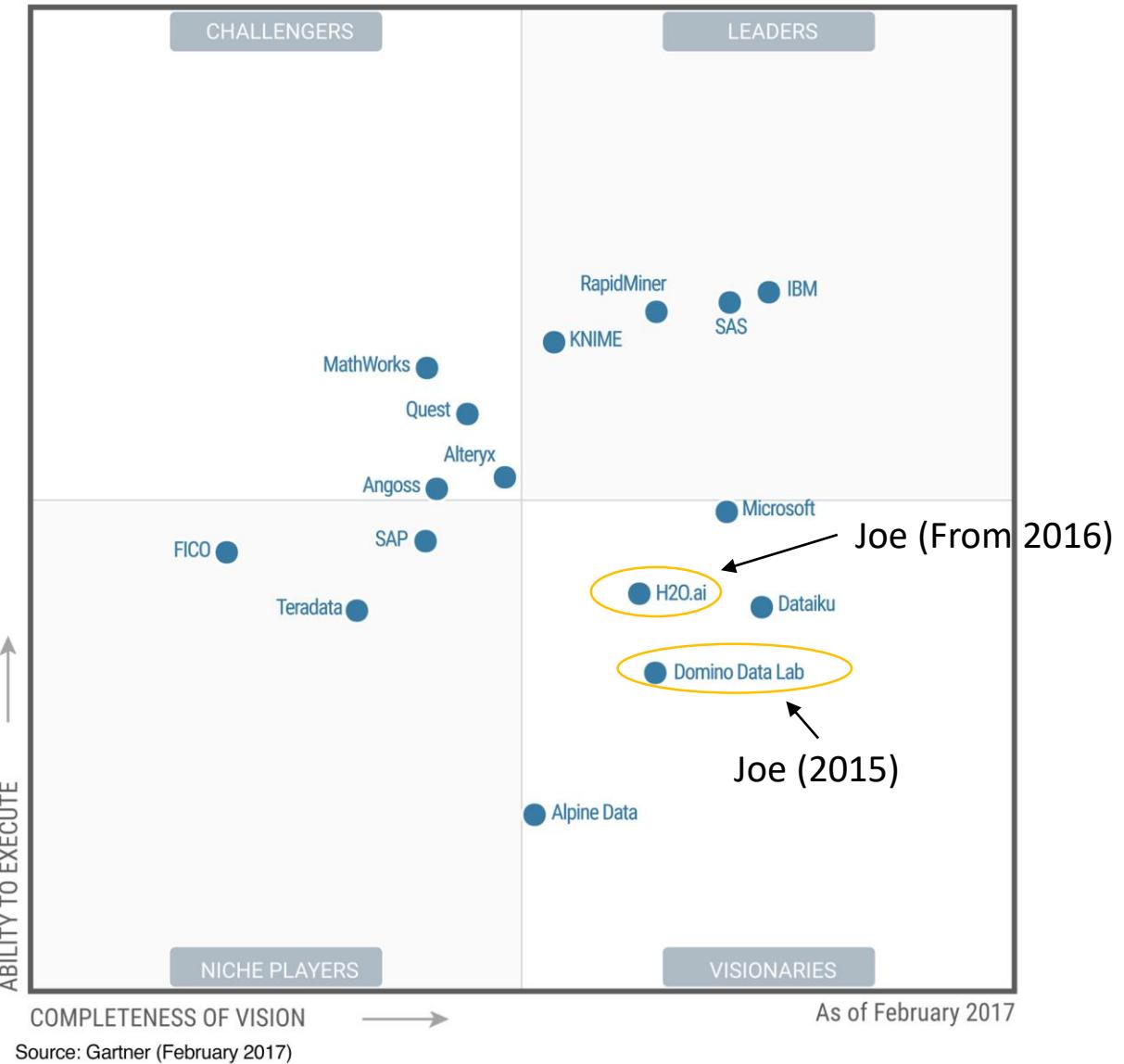
- Data Scientist
2015
 - Virgin Media (UK)
 - Domino Data Lab (Silicon Valley)
- 2016 – Present**
 - H₂O.ai (Silicon Valley)

Joe the Outlier

- Slides

- https://github.com/h2oai/h2o-meetups/blob/master/2017_02_28_SV_BigDataScience/2017_02_28_SVBDS_Kaggle_Story.pdf

Figure 1. Magic Quadrant for Data Science Platforms



ODSC Masterclass Summit

Introduction to Machine Learning & Deep Learning with H₂O

New Training Materials

- In both Python and R
 - Based on [Oxford IoT Course](https://github.com/h2oai/h2o-meetups/tree/master/2017_03_01_ODSC_Masterclass_Summit)
- Machine Learning with H₂O
 - Basic Extract, Transform and Load (ETL)
 - Supervised Learning
 - Parameters Tuning
 - Stacking
- Deep Learning with H₂O
 - MNIST Example
 - Outlier Detection
- Materials
 - https://github.com/h2oai/h2o-meetups/tree/master/2017_03_01_ODSC_Masterclass_Summit



Improving Model Performance (Step-by-Step)

Model Settings	MSE (CV)	MSE (Test)
GBM with default settings	N/A	0.4551
GBM with manual settings	N/A	0.4433
Manual settings + cross-validation	0.4502	0.4433
Manual + CV + early stopping	0.4429	0.4287
CV + early stopping + full grid search	0.4378	0.4196
CV + early stopping + random grid search	0.4227	0.4047
Stacking best two from random grid search	N/A	0.3997

Stacked Ensembles

Stacked Ensembles in H2O



February 2017

Erin LeDell Ph.D.
Machine Learning Scientist

https://github.com/h2oai/h2o-meetups/blob/master/2017_02_23_Metis_SF_Stacked_Eensemles_Deep_Water/stacked_ensembles_in_h2o_feb2017.pdf

H₂O.ai

Available in both Python and R

```
Model Stacking

In [23]: # Define a list of models to be stacked
# i.e. best model from each grid
all_ids = [best_gbm_model_id, best_drf_model_id, best_dnn_model_id]

In [24]: # Set up Stacked Ensemble
ensemble = H2OStackedEnsembleEstimator(model_id = "my_ensemble",
                                         base_models = all_ids)

In [25]: # use .train to start model stacking
# GLM as the default metalearner
ensemble.train(x = features,
                y = 'quality',
                training_frame = wine_train)

stackedensemble Model Build progress: |██████████| 100%
```

Comparison of Model Performance on Test Data

```
In [26]: print('Best GBM model from Grid (MSE) : ', best_gbm_from_rand_grid.model_performance(wine_test).mse())
print('Best DRF model from Grid (MSE) : ', best_drf_from_rand_grid.model_performance(wine_test).mse())
print('Best DNN model from Grid (MSE) : ', best_dnn_from_rand_grid.model_performance(wine_test).mse())
print('Stacked Ensembles (MSE) : ', ensemble.model_performance(wine_test).mse())

Best GBM model from Grid (MSE) : 0.4013942890547201
Best DRF model from Grid (MSE) : 0.478156285687009
Best DNN model from Grid (MSE) : 0.489784141303471
Stacked Ensembles (MSE) : 0.39965430199959595
```

© 2017 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About

```
Model Stacking

In [20]: # Define a list of models to be stacked
# i.e. best model from each grid
all_ids = list(best_gbm_model_id, best_drf_model_id, best_dnn_model_id)

In [21]: # Stack models
# GLM as the default metalearner
ensemble = h2o.stackedEnsemble(x = features,
                                y = 'quality',
                                training_frame = wine_train,
                                model_id = "my_ensemble",
                                base_models = all_ids)

|=====| 100%
```

Comparison of Model Performance on Test Data

```
In [22]: cat('Best GBM model from Grid (MSE) : ', h2o.performance(best_gbm_from_rand_grid, wine_test)$MSE,
      "\n")
cat('Best DRF model from Grid (MSE) : ', h2o.performance(best_drf_from_rand_grid, wine_test)$MSE,
      "\n")
cat('Best DNN model from Grid (MSE) : ', h2o.performance(best_dnn_from_rand_grid, wine_test)$MSE,
      "\n")
cat('Stacked Ensembles (MSE) : ', h2o.performance(ensemble, wine_test)$MSE, "\n")

Best GBM model from Grid (MSE) : 0.4013943
Best DRF model from Grid (MSE) : 0.4781568
Best DNN model from Grid (MSE) : 0.5543555
Stacked Ensembles (MSE) : 0.3989076
```

© 2017 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About

https://github.com/woobe/odsc_h2o_machine_learning

Automatic Machine Learning (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.

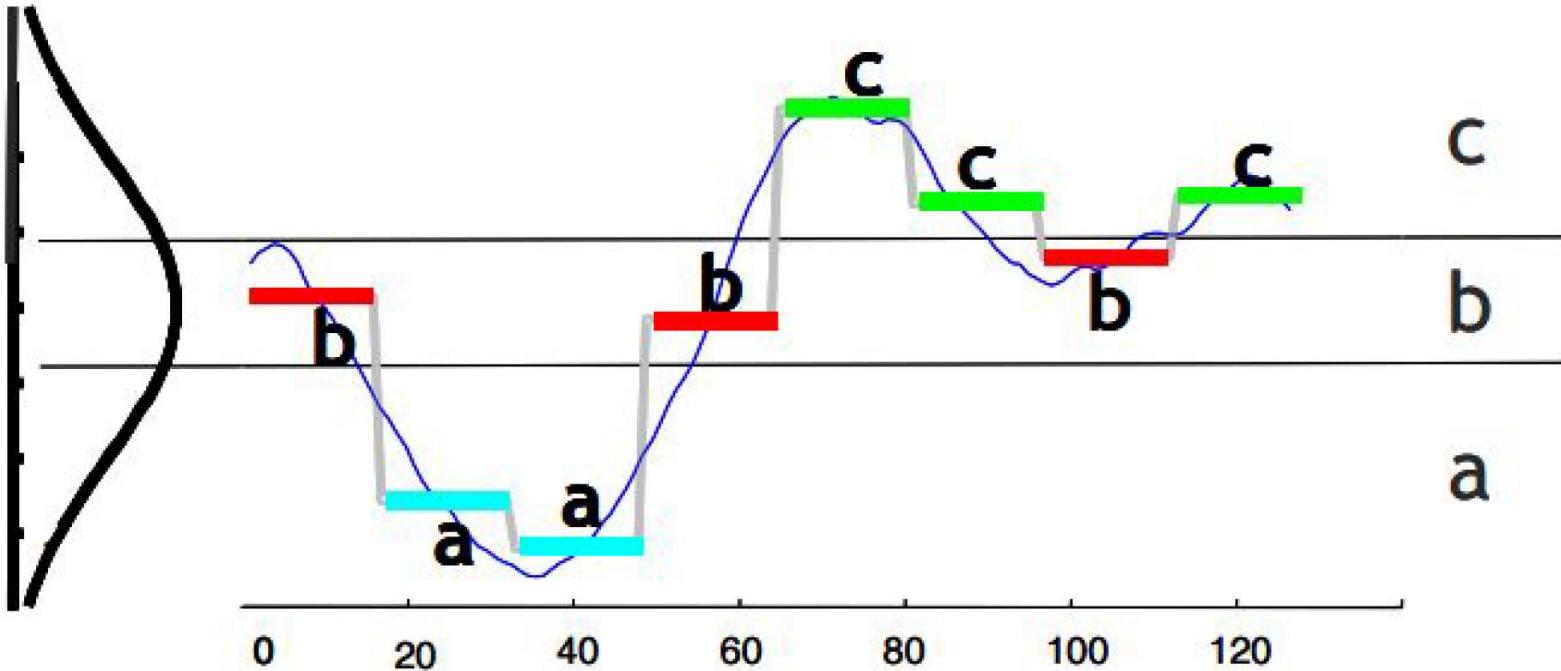
Public code coming soon!

Time Series in H₂O

ISAX

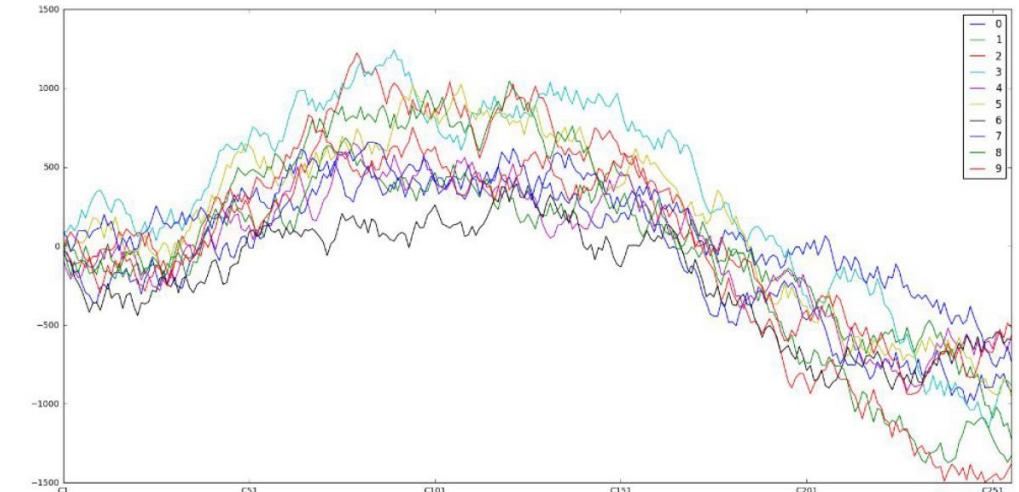
- Time series data compression algorithm, implemented on H2O's distributed architecture
- Find groups of similar time series patterns in one billion
 - <http://blog.h2o.ai/2016/11/indexing-1-billion-time-series-with-h2o-and-isax/>
- ISAX applied to fields such as IoT, Finance, Bioinformatics, Image/Sound processing.
 - <http://www.cs.ucr.edu/~eamonn/SAX.htm>
- Compress your time series data and run:
 - Clustering
 - Classification
 - Anomaly Detection
 - Predictive analytics

ISAX



ISAX

- Highly scalable, tested on 2TB of data, 1Billion time series w/ 256 points (columns)
 - Leverages H2O's MapReduce framework
- Accessible from R or Python
- Image below, a group of similar time series was found from 1 Billion. Search took ~5 minutes. Indexing took 10 minutes.



Time Series at H2O

- More time series features are coming to H2O soon. Keep checking back!
 - Pivoting
 - Rolling Window

Blog Post

<http://blog.h2o.ai/2016/11/indexing-1-billion-time-series-with-h2o-and-isax/>

Indexing 1 Billion Time Series with H2O and ISax

At H2O, we have recently debuted a new feature called ISax that works on time series data in an H2O Dataframe. ISax stands for Indexable Symbolic Aggregate APPROXimation, which means it can represent complex time series patterns using a symbolic notation and thereby reducing the dimensionality of your data. From there you can run H2O's ML algos or use the index for searching or data analysis. ISax has many uses in a variety of fields including finance, biology and cybersecurity.

Today in this blog we will use H2O to create an ISax index for analytical purposes. We will generate 1 Billion time series of 256 steps on an integer U(100,100) distribution. Once we have the index we'll show how you can search for similar patterns using the index.

We'll show you the steps and you can run along, assuming you have enough hardware and patience. In this example we are using a 9 machine cluster, each with 32 cores and 256GB RAM. We'll create a 1B row synthetic data set and form random walks for more interesting time series patterns. We'll run ISax and perform the search, the whole process takes ~30 minutes with our cluster.

Raw H2O Frame Creation

In the typical use case, H2O users would be importing time series data from disk. H2O can read from local filesystems, NFS, or distributed systems like Hadoop. H2O cluster file reads are parallelized across the nodes for speed. In our case we'll be generating a 256 columns, 1B row frame. By the way H2O Dataframes scales better by increasing rows instead of columns. Each row will be an individual time series. The ISax algo assumes the time series data is row based.

```
rawdf = h2o.create_frame(cols=256, rows=1000000000, real_fraction=0.0, integer_fraction=1.0, missing_fraction=0.0)

In [4]: print(datetime.datetime.now())
rawdf = h2o.create.frame(cols<256, rows=1000000000, real_fraction=0.0, integer_fraction=1.0, missing_fraction=0.0)
print(datetime.datetime.now())

2016-11-08 13:11:00.456099
Create Frame progress: [██████████] 100%
2016-11-08 13:15:40.852390
```

Random Walk

Here we do a row wise cumulative sum to simulate random walks. The .head call triggers the execution graph so we can do a time measurement.

```
tsdf = rawdf.cumsum(axis=1)
print tsdf.head()

In [7]: print(datetime.datetime.now())
tsdf = rawdf.cumsum(axis=1)
print tsdf.head()
print(datetime.datetime.now())

2016-11-08 13:44:42.466140
[ C1 C2 C3 C4 C5 C6 C7 C8 C9 ]
[-3 -40 -104 -128 -178 -180 -202 -138 -229
 -23 -54 -86 -170 -154 -235 -294 -312 -357
 29 19 -2 -86 -24 -11 -3 58 78
 88 123 80 -20 36 -63 -25 52 1
 -63 -116 -194 -181 -167 -188 -214 -188 -101
 -9 -18 -34 -48 -46 -45 -12 -58 -21
 100 118 216 228 258 345 279 262 313
 -1 34 59 -26 -39 4 84 113 36]
```

Lets take a quick peek at our time series

```
tsdf[0:2,:].transpose().as_data_frame(use_pandas=True).plot()

In [8]: tsdf[0:2,:].transpose().as_data_frame(use_pandas=True).plot()

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x7f59bc14ad90>
```

Run ISax

Now we're ready to run isax and generate the index. The output of this command is another H2O Frame that contains the string representation of the isax word, along with the numeric columns in case you want to run ML algos.

```
res = tsdf.isax(num_words=20,max_cardinality=10)

In [9]: print(datetime.datetime.now())
res = tsdf.isax(num_words=20,max_cardinality=10)
print(datetime.datetime.now())

2016-11-08 13:52:59.440464
2016-11-08 14:02:27.059648
```

Takes 10 minutes and H2O's MapReduce framework makes efficient use of all 288 cpu cores.

More Slides & Code Examples

- https://github.com/h2oai/h2o-meetups/tree/master/2017_02_16_TimeSeries_Meetup
- <https://github.com/h2oai/h2o-3/blob/master/h2o-core/src/main/java/water/rapids/ast/prims/reducers/AstCumu.java>
- <https://github.com/h2oai/h2o-3/blob/master/h2o-core/src/main/java/water/rapids/ast/prims/timeseries/AstLsax.java>

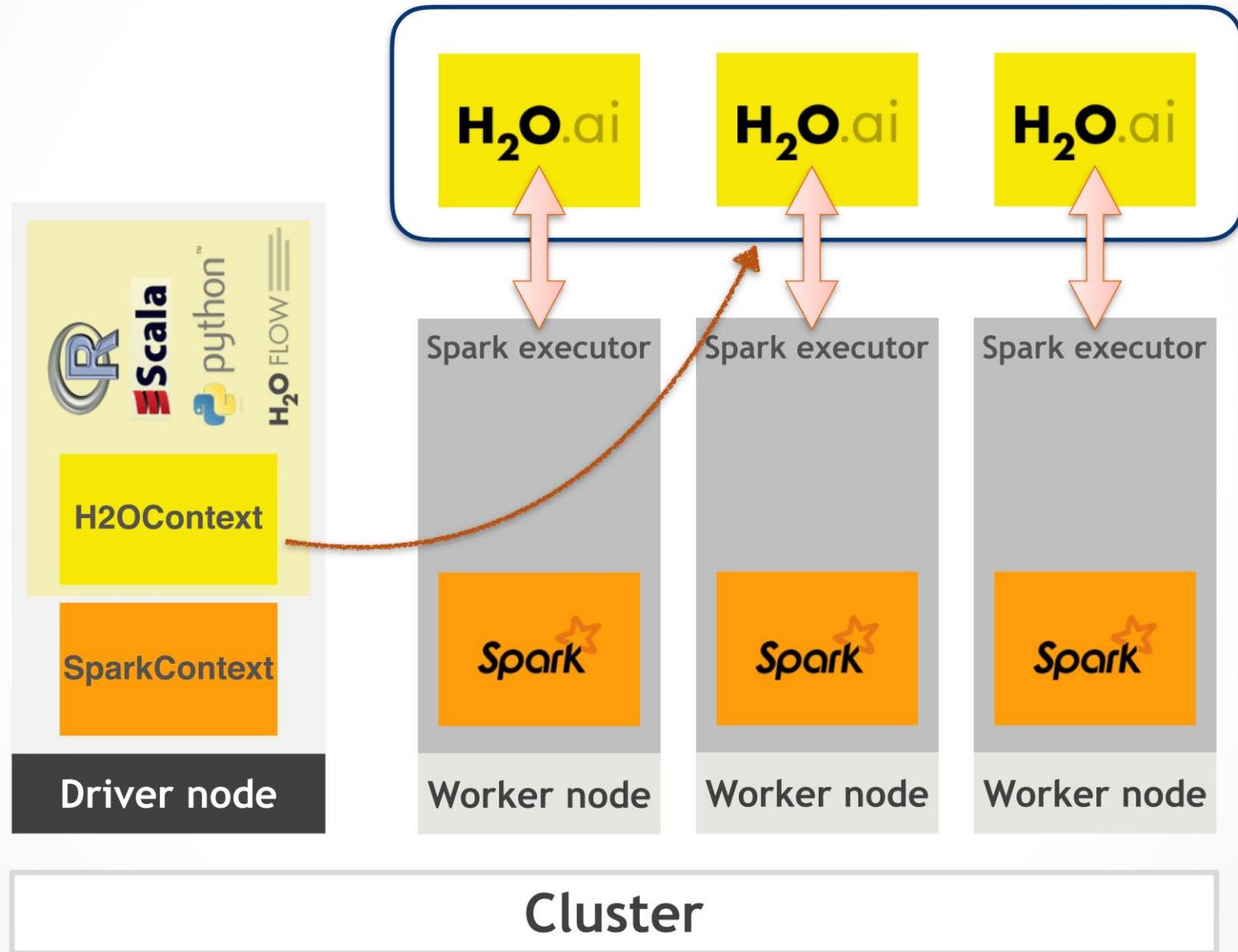
Sparkling Water

H₂O's integration with Spark – latest developments

External H₂O Backend for Sparkling Water

- High Availability Mode
 - Separating Spark and H₂O
 - while preserving same API
- Advantages
 - H₂O does not crash when Spark executor goes down
 - Better resource management since resources can be planned per tool
- Disadvantages
 - Transfer overhead between Spark and H₂O processes
 - Under measurement with cooperation of a customer
- Links
 - <https://github.com/h2oai/sparkling-water/blob/master/doc/backends.md>

High Availability



Deep Water

H₂O's integration with TensorFlow, MXNet and Caffe – latest developments

Latest in Deep Water

- MXNet
 - Works on most stuff
- TensorFlow
 - Works on most stuff
- Caffe
 - Works on Multi-GPU
 - Does not score yet
- Latest deepwater branch in h2o-3
 - <https://github.com/h2oai/h2o-3/tree/deepwater>

H_2O + xgboost

Brand-new: H2O XGBoost Integration (Gradient Boosting)

Why XGBoost?

Competitive **accuracy** and **speed** (great for Kaggle)

GPU support (for small/medium data)

Efficient on **sparse** data

Why integrate into H2O?

Ease of use (**Flow** GUI, R/Py APIs)

Real-time model status (var imp, metrics)

Efficient **data preprocessing** (sparse, categorical)

Integration into **H2O ecosystem** (modeling, deployment, support)

Live Demo of GPU Gradient Boosting in H2O

Build a Model

Select an algorithm: XGBoost

booster gbtree

reg_lambda 1

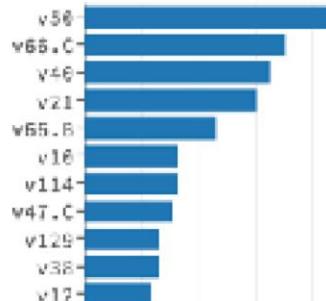
reg_alpha (Choose...)

auto

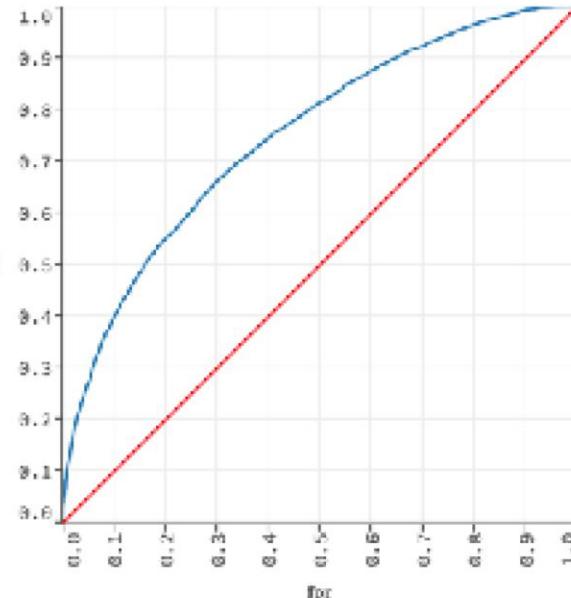
backend gpu

cpu

VARIABLE IMPORTANCES



* ROC CURVE - VALIDATION METRICS , AUC = 0.750331



```
+-----+  
| NVIDIA-SMI 375.28      Driver Version: 375.28 |  
+-----+  
| GPU  Name     Persistence-M| Bus-Id     Disp.A  Volatile Uncorr. ECC |  
| Fan  Temp  Perf  Pwr.Usage/Cap| Memory-Usage  GPU-Util  Compute M. |  
|-----+-----+-----+-----+-----+-----+-----+  
|  0  GeForce GTX 1080     Off  | 0000:02:00.0  On   | N/A  |  
| 27% 43C    P2    83W / 188W | 2848MiB / 8145MiB | 94%   Default |  
+-----+
```

```
gbm = h2o.get_model("h2o-gbm")
xgb = h2o.get_model("h2o-xgboost")
print("H2O GBM:      Validation AUC=%r" % gbm.auc(valid=True))
print("H2O XGBoost: Validation AUC=%r" % xgb.auc(valid=True))

H2O GBM:      Validation AUC=0.7517126133633125
H2O XGBoost: Validation AUC=0.750331001716736
```

Community Events in Europe



H₂O Events in Europe - 2016

London: 13

Amsterdam: 1

Exeter: 1

Warsaw: 1

Paris: 3

Budapest: 2

Milan: 1

Cities: 7
Events: 22



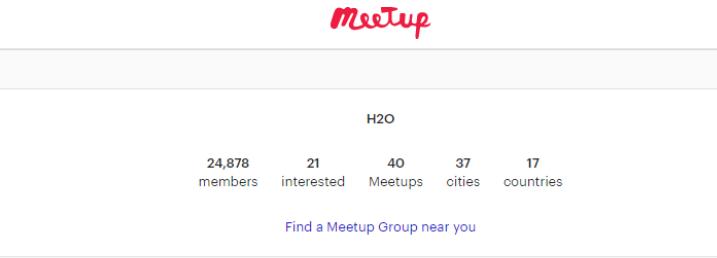
H₂O Events in Europe - 2017

Scheduled so far ...

Cities: >17
Events: >22

H₂O Meetup Groups

- www.meetup.com/topics/h2o/all/
- Need your help ☺



All H2O Meetups			
Silicon Valley Big Data Science 6,560 Makers Mountain View, CA	NYC Big Data Science 4,098 makers New York, NY	San Francisco Big Data Science 2,679 Makers San Francisco, CA	
H2O and Data Science! 1,810 Hackers Mountain View, CA	Dallas Big Data Science! 1,579 Data Scientists Dallas, TX	London Artificial Intelligence & Deep Learning 580 Makers London, United Kingdom	
Amsterdam Artificial Intelligence & Deep Learning 534 Makers Amsterdam, Netherlands	Tel Aviv Artificial Intelligence & Deep Learning 390 Makers Tel Aviv-Yafo, Israel	Berlin Artificial Intelligence & Deep Learning 365 Makers Berlin, Germany	
Singapore Artificial Intelligence & Deep Learning 381 Makers Singapore, Singapore	Sydney Artificial Intelligence & Deep Learning 374 Makers Sydney, Australia	Madrid Artificial Intelligence & Deep Learning 364 Makers Madrid, Spain	
San Diego Artificial Intelligence & Deep Learning 331 Makers San Diego, CA	Toronto Artificial Intelligence & Deep Learning 326 Makers Toronto, ON	Paris Artificial Intelligence & Deep Learning 328 Makers Paris, France	
Hyderabad Artificial Intelligence & Deep Learning 313 Makers Hyderabad, India	Washington D.C. Artificial Intelligence & Deep Learning 294 Makers Washington, DC	Pune Artificial Intelligence & Deep Learning 283 Makers Pune, India	
Dublin Artificial Intelligence & Deep Learning 272 Makers Dublin, Ireland	Bangalore Artificial Intelligence & Deep Learning 269 Makers Bangalore, India	Boston Artificial Intelligence & Deep Learning 264 Makers Boston, MA	
Seattle Artificial Intelligence & Deep Learning 226 Makers Seattle, WA	São Paulo Artificial Intelligence & Deep Learning 224 Makers São Paulo, Brazil	Mumbai Artificial Intelligence & Deep Learning 219 Makers Mumbai, India	
Chicago Fast, Scalable Big Data Machine Learning With H2O 213 H2O Explorers Chicago, IL	Chicago Artificial Intelligence & Deep Learning 197 Makers Chicago, IL	Chennai Artificial Intelligence & Deep Learning 177 Members Chennai, India	
Atlanta Artificial Intelligence & Deep Learning 176 Makers Atlanta, GA	Delhi Artificial Intelligence & Deep Learning 171 Members Delhi, India	Denver Artificial Intelligence & Deep Learning 156 Members Denver, CO	
Irvine Artificial Intelligence & Deep Learning 150 Members Irvine, CA	Taipei Artificial Intelligence & Deep Learning 148 Members Taipei, Taiwan	Tokyo Artificial Intelligence & Deep Learning 119 Makers Tokyo, Japan	
Moscow Artificial Intelligence & Deep Learning 109 Members Moscow, Russia	Los Angeles Artificial Intelligence & Deep Learning 109 Makers Los Angeles, CA	Hartford Artificial Intelligence & Deep Learning 70 Makers Hartford, CT	
Beijing Artificial Intelligence & Deep Learning 62 Members Beijing, China	Charlotte Artificial Intelligence & Deep Learning 56 Makers Charlotte, NC	Kolkata Artificial Intelligence & Deep Learning 51 Makers Kolkata, India	

H₂O on ARM

H₂O on ARM

- New Challenge
 - Use ARM device for MOJO (scoring only – not model training)
 - C++ instead of Java
- Discussion
 - Data Collection
 - Inputs? (IoT Applications)
 - Outputs? (How to represent)

ARM mbed

Developer Resources | Partners | Cloud

Hardware Documentation Code Questions Forum | jofaichow Compiler

Boards » FRDM-K64F

FRDM-K64F

The Freedom-K64F is an ultra-low-cost development platform for Kinetis K64, K63, and K24 MCUs.



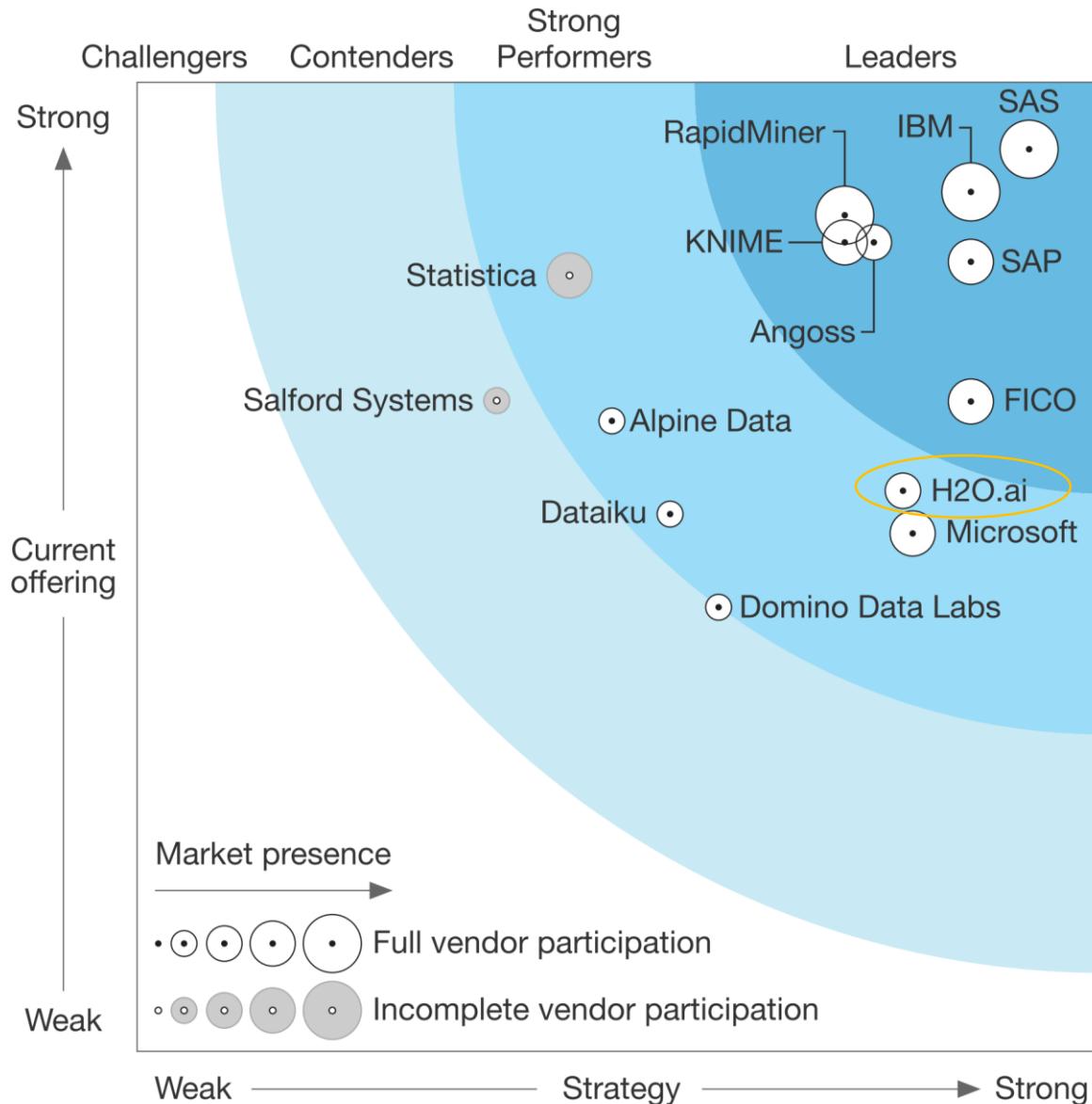
Overview

The Flagship FRDM-K64F has been designed by NXP in collaboration with mbed for prototyping all sorts of devices, especially those requiring optimized size and price points. The board is well sized for connected applications, thanks to its power efficient Kinetis K64 MCU featuring an ARM® Cortex®-M4 core running up to 120MHz and embedding 1024KB Flash, 256KB RAM and lots of peripherals (16-bit ADCs, DAC, Timers) and interfaces (Ethernet, USB Device Crystal-less and Serial). The Kinetis K64 MCU family remains fully software, hardware and development tool compatibility with Kinetis MCU and Freedom board families. It is packaged as a development board including extension headers compatible with Arduino R3 shields and includes a built-in USB Debug and Flash Programmer.

Table of Contents

1. Overview
2. MCU Features
3. Board Features
4. Board Block Diagram
5. Board Pinout
6. PC Configuration
7. Firmware Update
8. Get Started with mbed
9. Flash a project binary
10. Open existing Project
11. Create new Project

Other News



FORRESTER® RESEARCH

The Forrester Wave™

Go to Forrester.com to download the Forrester Wave tool for more detailed product evaluations, feature comparisons, and customizable rankings.

107 OF THE FORTUNE 500 LOVE H₂O

8 OF TOP 10
BANKS

7 OF TOP 10
INSURANCE COMPANIES

4 OF TOP 10
HEALTHCARE COMPANIES

Thanks!

- Slides
 - bit.ly/h2o_meetups
- Contact
 - joe@h2o.ai
 - [@matlabulous](https://twitter.com/matlabulous)
 - github.com/woobe



H₂O.ai

Making Machine Learning
Accessible to Everyone

Photo credit: Virgin Media