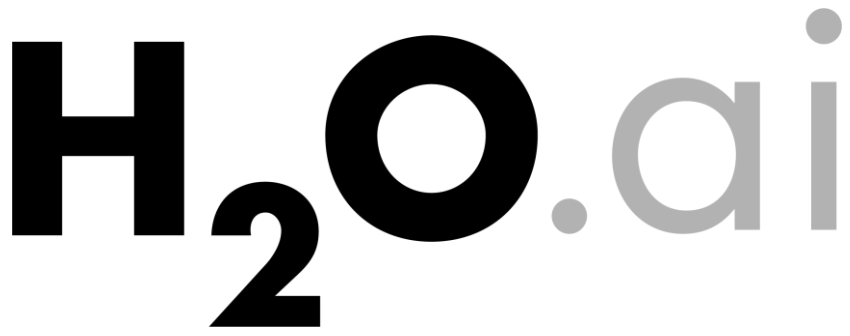
Latest Developments in H_2O



Jo-fai (Joe) Chow

Data Scientist

joe@h2o.ai

@matlabulous

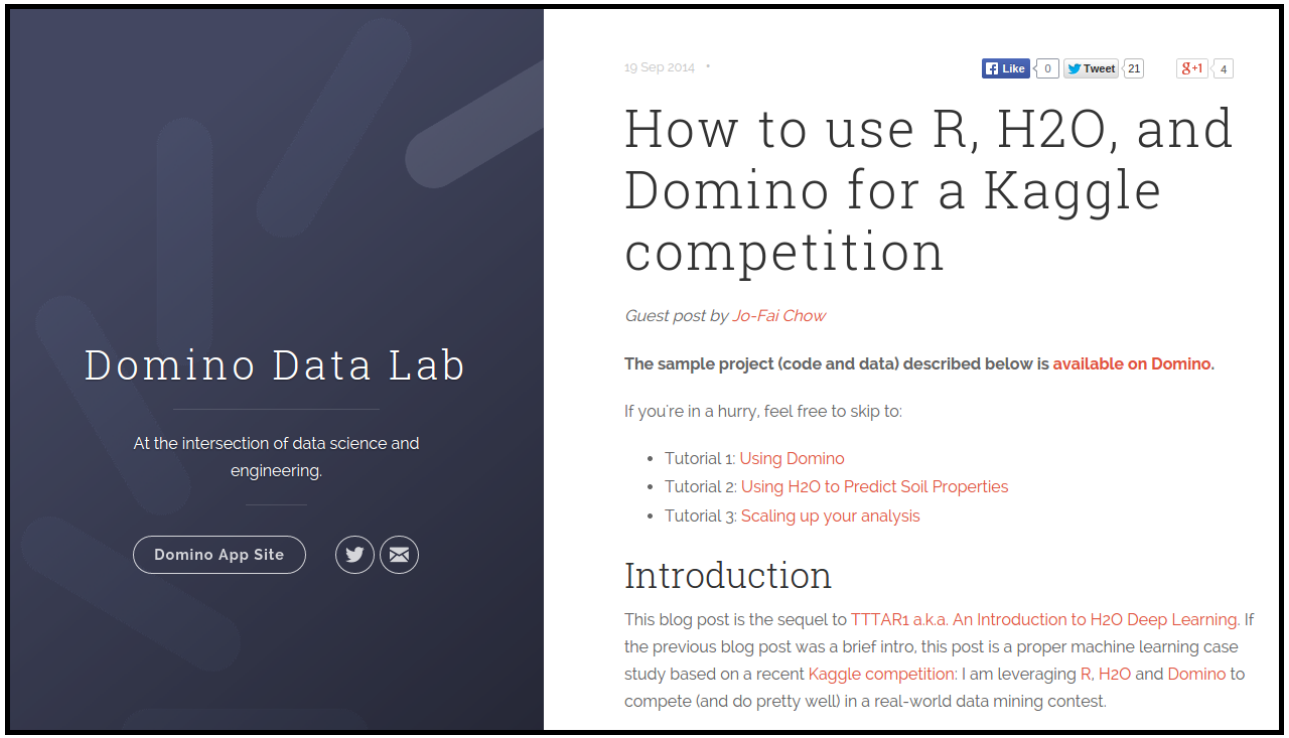
H_2O at Booking.com – Amsterdam

6th April, 2017

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 H_2O in 2014
- Data Scientist
  - 2015
  - Virgin Media (UK)
  - Domino Data Lab (Silicon Valley, US)
  - 2016 – Present
  - H_2O.ai (Silicon Valley, US)

About Me



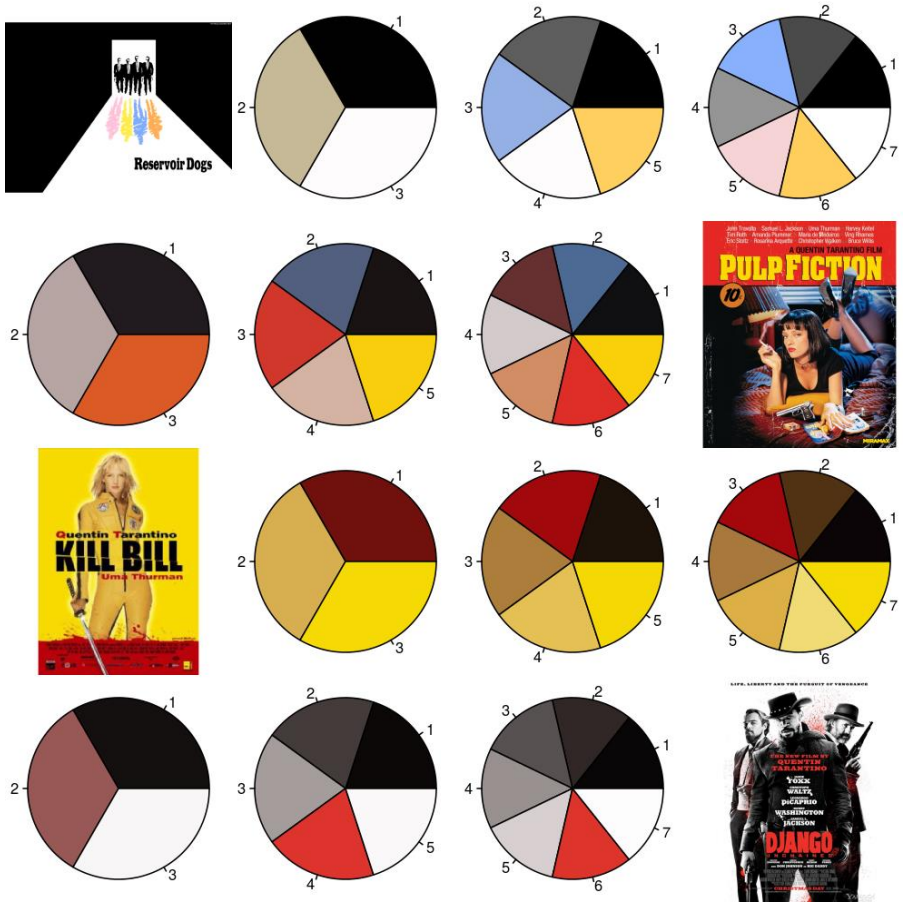
R + H_2O + Domino for Kaggle

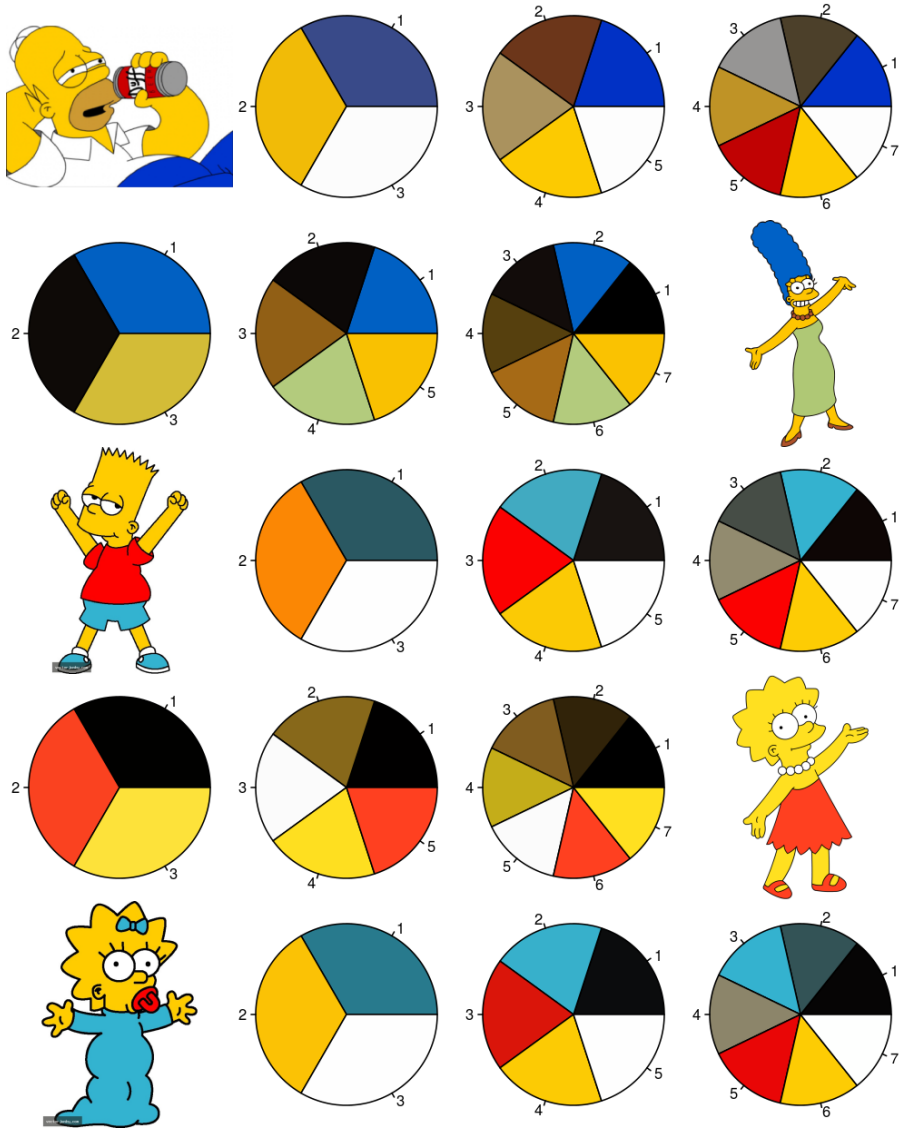
Guest Blog Post for Domino & H_2O (2014)

- The Long Story
  - bit.ly/joe_kaggle_story

H_2O.ai

About Me





Developing R Packages for Fun
rPlotter (2014)

H2O.ai

Agenda

- About H_2O.ai
  - Company
  - Machine Learning Platform
- What's New?
  - Deep Water
  - H_2O + xgboost
  - Stacked Ensembles
  - Auto Machine Learning
  - Model Interpretation
  - Community
- PyData Conference

Booking.com

About H_2O.ai

Company Overview

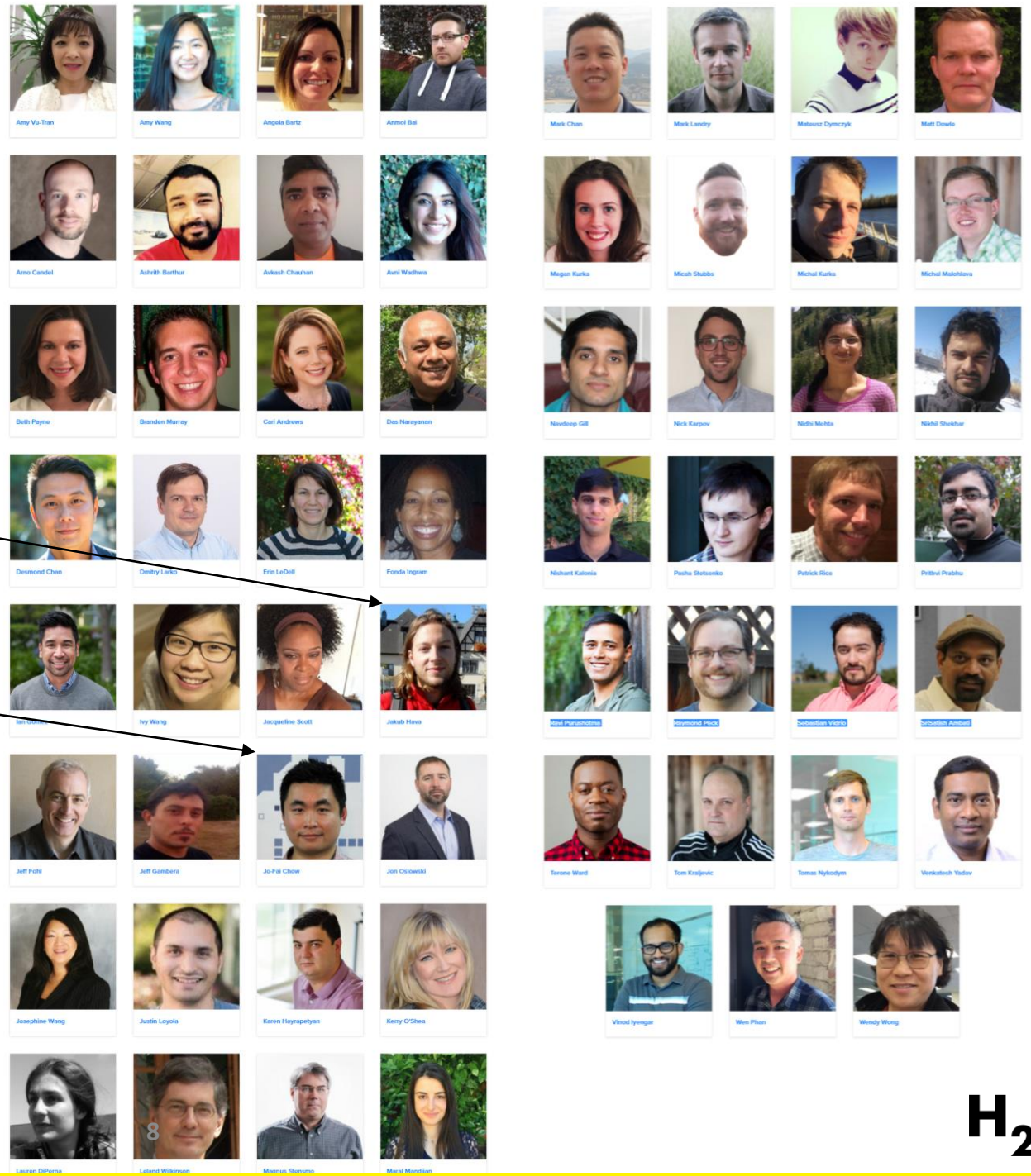
Founded	2011 Venture-backed, debuted in 2012
Products	• H_2O Open Source In-Memory AI Prediction Engine • Sparkling Water • Steam
Mission	Operationalize Data Science, and provide a platform for users to build beautiful data products
Team	70 employees • Distributed Systems Engineers doing Machine Learning • World-class visualization designers
Headquarters	Mountain View, CA




Our Team

Kuba

Joe



H2O.ai

Scientific Advisory Council

Dr. Trevor Hastie

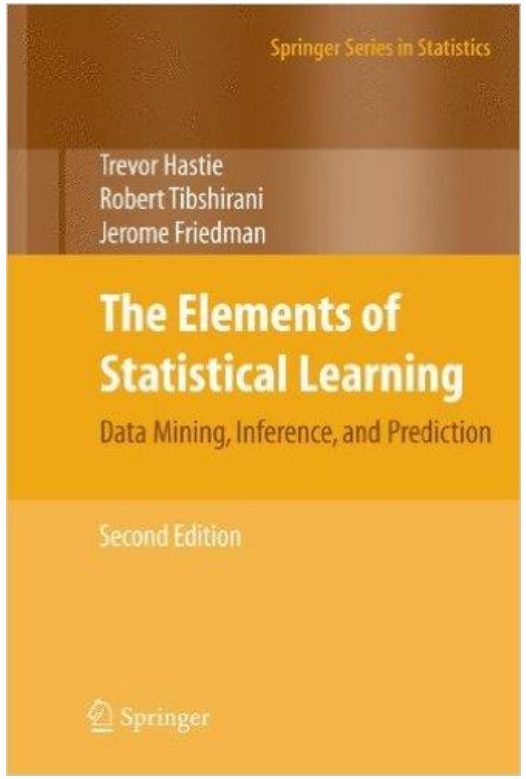
- John A. Overdeck Professor of Mathematics, Stanford University
- PhD in Statistics, Stanford University
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Co-author with John Chambers, *Statistical Models in S*
- Co-author, *Generalized Additive Models*

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.ai

wenphan
@wenphan

Following

So much brain power in one place: @ArnoCandel and Stanford profs. Boyd, Tibs, and Hastie. Hacking algos at @h2oai HQ

H2O.ai

Figure 1. Magic Quadrant for Data Science Platforms

Source: Gartner (February 2017)

H2O.ai recognized for completeness of vision and ability to execute

We are thrilled to be named a Visionary among the 16 vendors included in Gartner's 2017 Magic Quadrant for Data Science Platforms. As a Visionary we believe we are positioned highest in Ability to Execute for companies of our size and scale.

Since 2011, our mission has been to democratize data science through open source AI and deep learning. Today, H2O.ai is focused on bringing AI to enterprises with a growing community of more than 8,500 organizations that depend on H2O for mission critical applications. H2O.ai was recently named CB Insights AI 100 and is used by 107 of the Fortune 500 companies.

Disclaimer: This graphic was published by Gartner, Inc. as part of a larger research document and should be evaluated in the context of the entire document. The Gartner document is available upon request from H2O.ai.

http://www.h2o.ai/gartner-magic-quadrant/

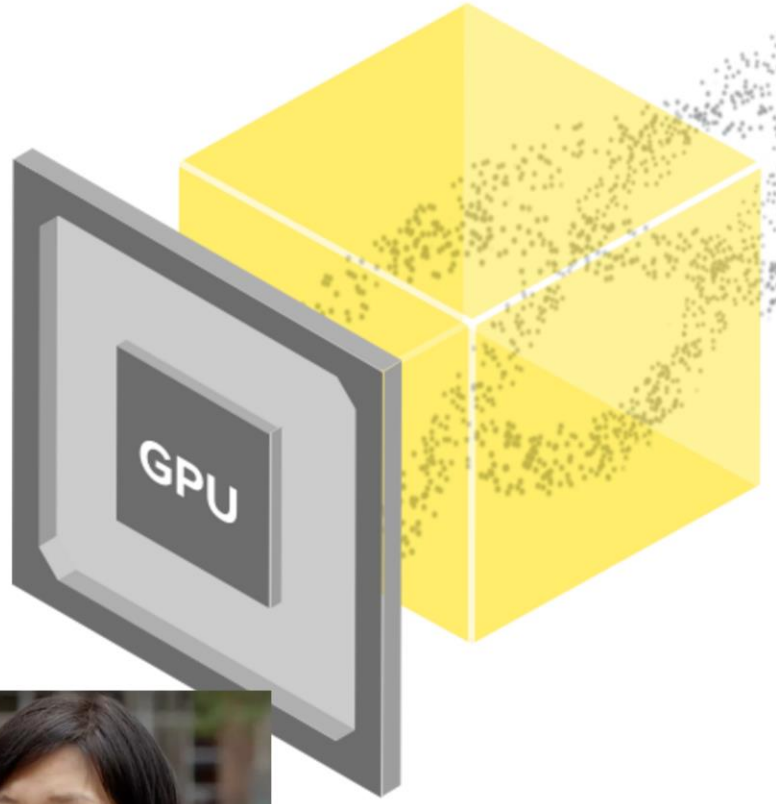
H2O.ai

Check out our website h2o.ai

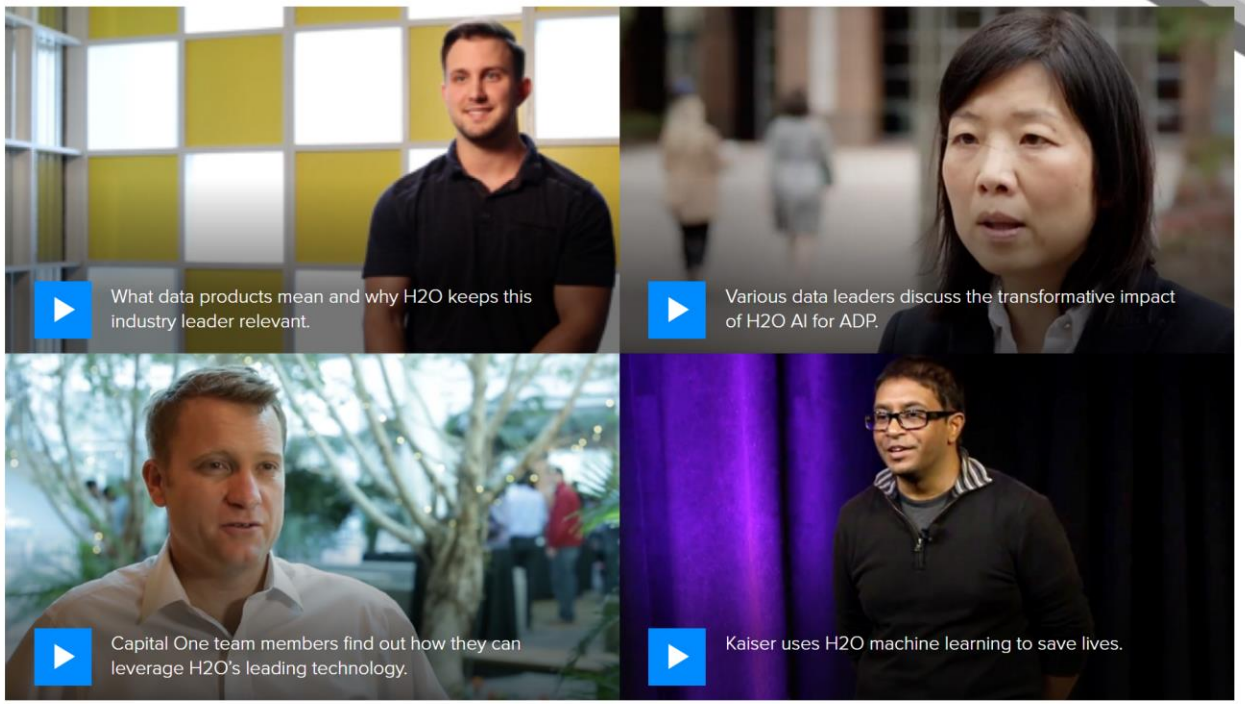
World Record Performance for AI

H2O.ai is accelerating both machine learning and deep learning on GPUs, providing enterprises opportunities to build better models and enable new use cases.

SEE DEEP WATER

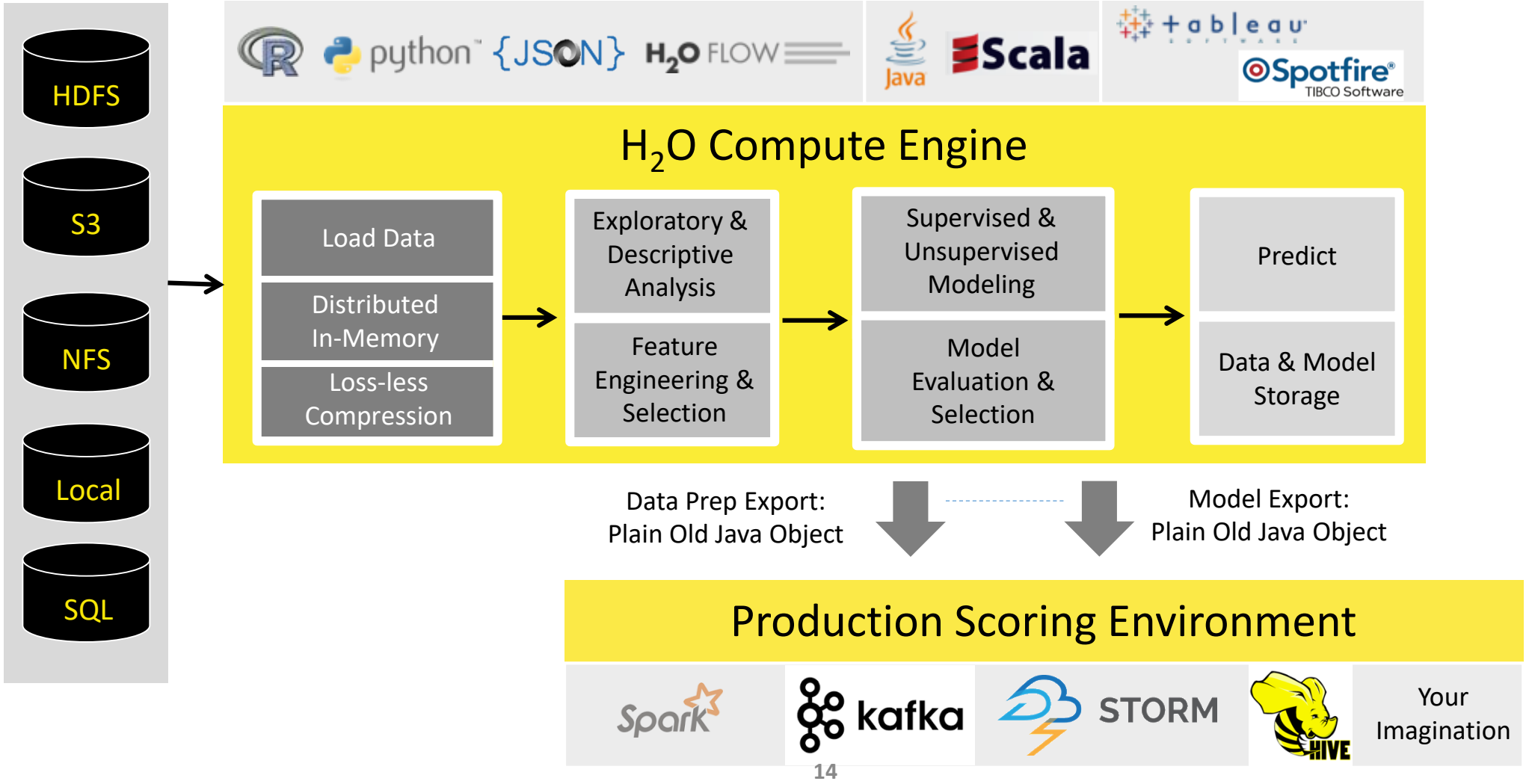


H2O in Action



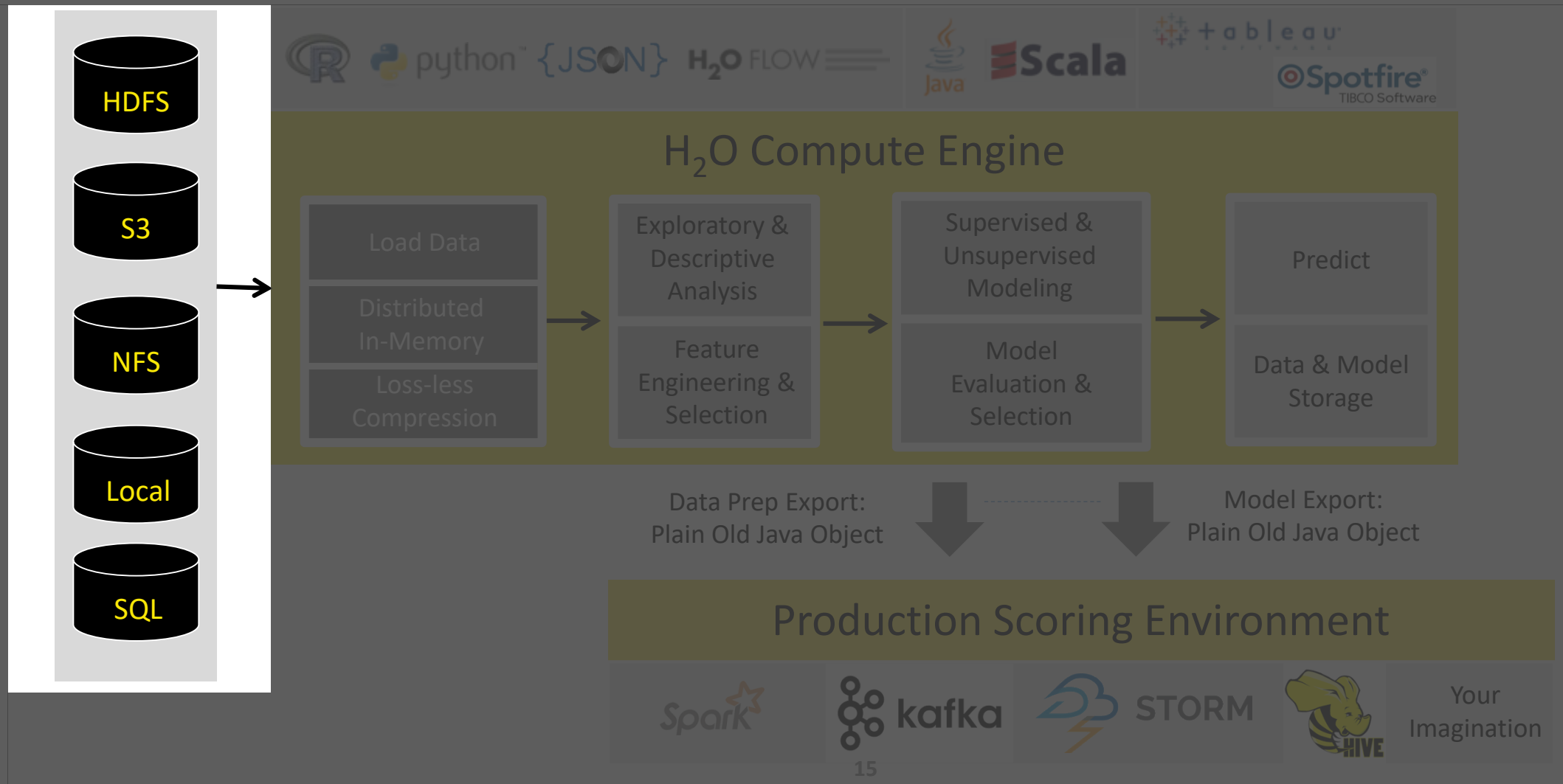
H_2O Machine Learning Platform

High Level Architecture



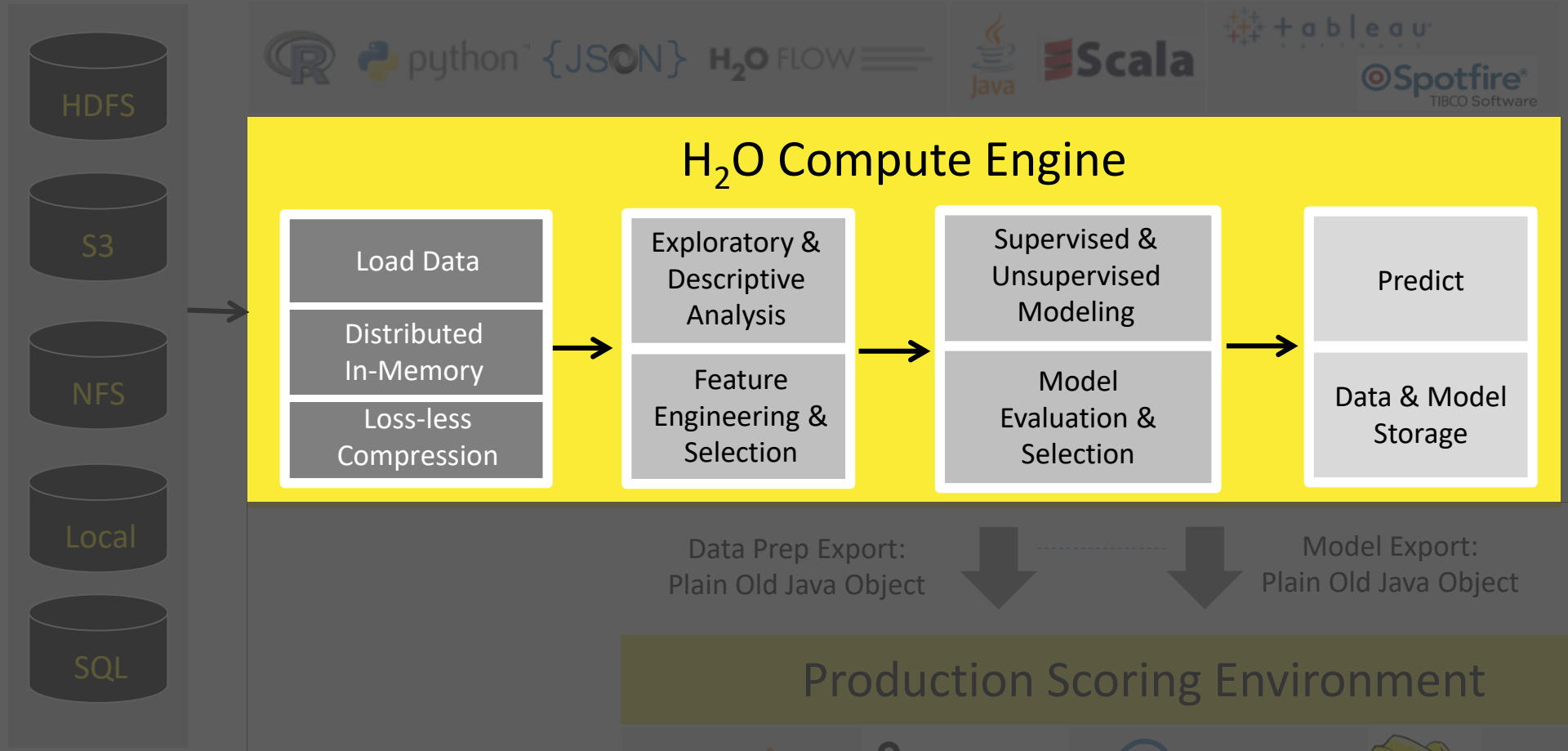
High Level Architecture

Import Data from Multiple Sources



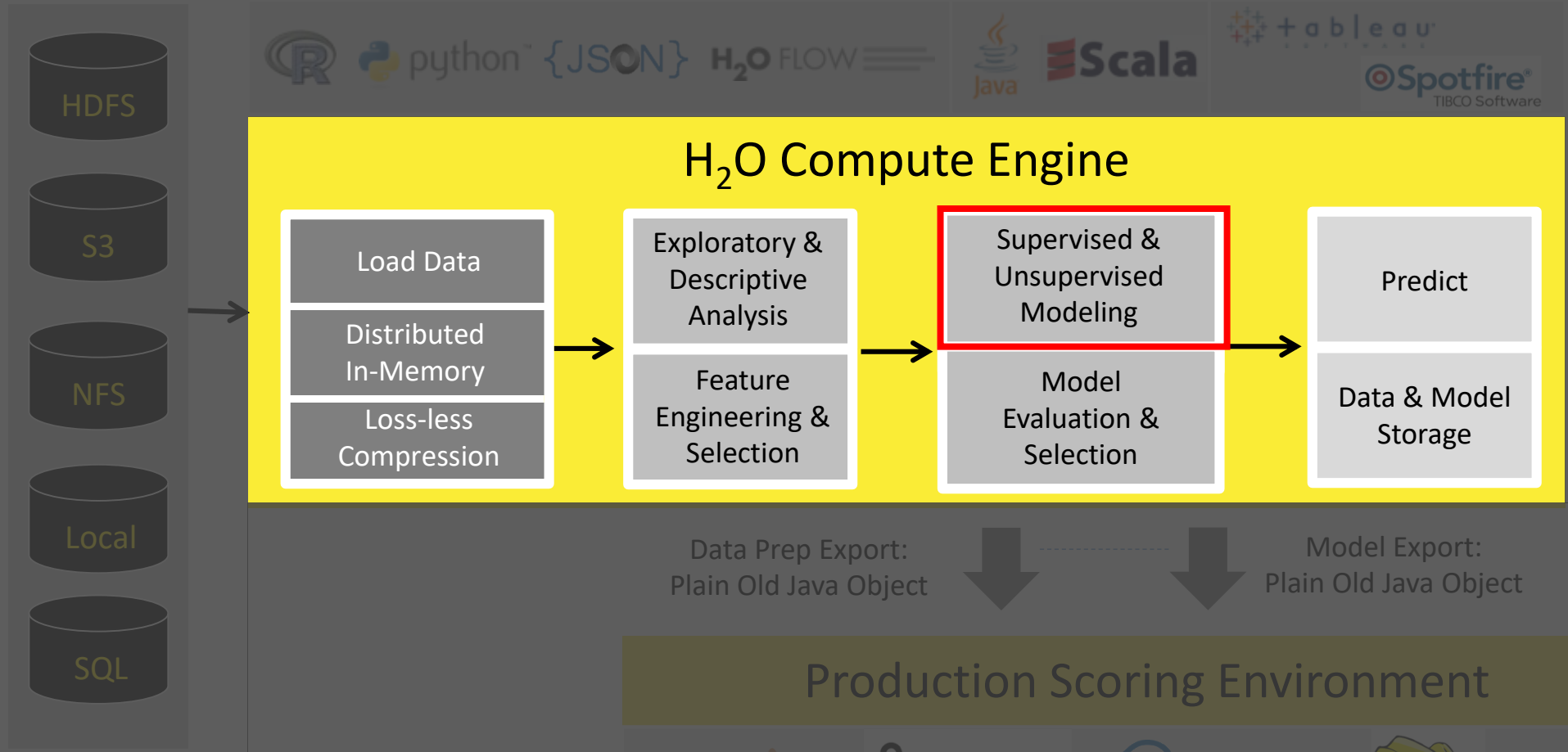
High Level Architecture

Fast, Scalable & Distributed Compute Engine Written in Java



High Level Architecture

Fast, Scalable & Distributed Compute Engine Written in Java



Algorithms Overview

Supervised Learning

Statistical Analysis	• **Generalized Linear Models**: Binomial, Gaussian, Gamma, Poisson and Tweedie • **Naïve Bayes**
Ensembles	• **Distributed Random Forest**: Classification or regression models • **Gradient Boosting Machine**: Produces an ensemble of decision trees with increasing refined approximations
Deep Neural Networks	• **Deep learning**: Create multi-layer feed forward neural networks starting with an input layer followed by multiple layers of nonlinear transformations

Unsupervised Learning

Clustering	• **K-means**: Partitions observations into k clusters/groups of the same spatial size. Automatically detect optimal k
Dimensionality Reduction	• **Principal Component Analysis**: Linearly transforms correlated variables to independent components • **Generalized Low Rank Models:** extend the idea of PCA to handle arbitrary data consisting of numerical, Boolean, categorical, and missing data
Anomaly Detection	• **Autoencoders**: Find outliers using a nonlinear dimensionality reduction using deep learning

H_2O Deep Learning in Action

116M rows, 6GB CSV file
800+ predictors (numeric + categorical)

airlines_all_selected_cols.hex

Actions: View Data | Split... | Build Model... | Predict | Download | Export

Rows	Columns	Compressed Size
116695259	12	2GB

Job

Run Time 00:00:36.712
Remaining Time 00:00:17.188
Type Model
Key deeplearning-dd2f42f7-81f7-42e8-9d98-e34437309828
Description DeepLearning
Status RUNNING
Progress 69%

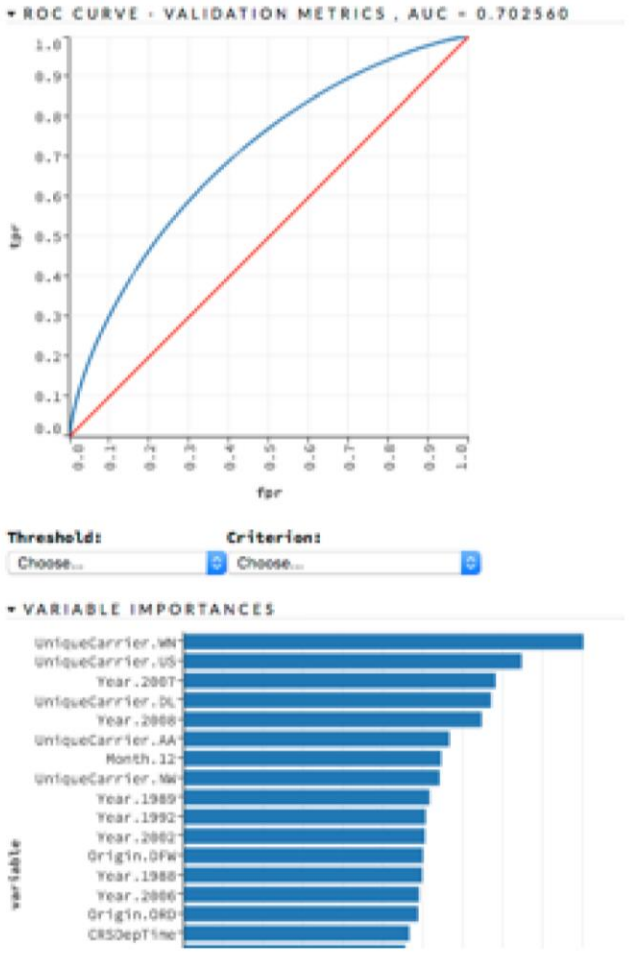
Iterations: 12. Epochs: 0.628821. Speed: 2,243,735 samples/sec. Estimated time left: 21.849 sec

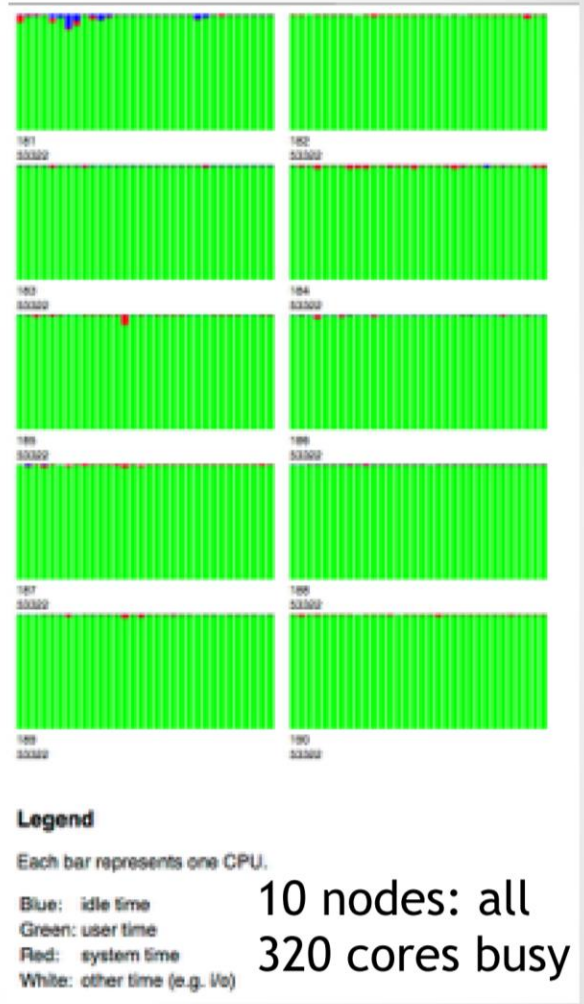
Actions: View | Cancel Job

model trained in <1 min:
2M+ samples/second

▾ OUTPUT - STATUS OF NEURON LAYERS (PREDICTING ISDEPDELAYED, 2-CLASS CLASSIFICATION, BERNOULLI DISTRIBUTION, CROSSENTROPY LOSS, 17,462 WEIGHTS/BIASES, 221.3 KB, 106,585,365 TRAINING SAMPLES, MINI-BATCH SIZE 1)

layer	units	type	dropout	l1	l2	mean_rate	rate_RMS	momentum	mean_weight	weight_RMS	mean_bias	bias_RMS
1	807	Input	0									
2	20	Rectifier	0	0	0	0.0493	0.2020	0	-0.0021	0.2111	-0.9139	1.0036
3	20	Rectifier	0	0	0	0.0197	0.0227	0	-0.1033	0.5362	-1.3988	1.5259
4	20	Rectifier	0	0	0	0.0517	0.0446	0	-0.1575	0.3068	-0.6846	0.6846
5	20	Rectifier	0	0	0	0.0761	0.0844	0	-0.0374	0.2275	-0.2647	0.2481
6	2	Softmax		0	0	0.0161	0.0083	0	0.0741	0.7260	0.4269	0.2056





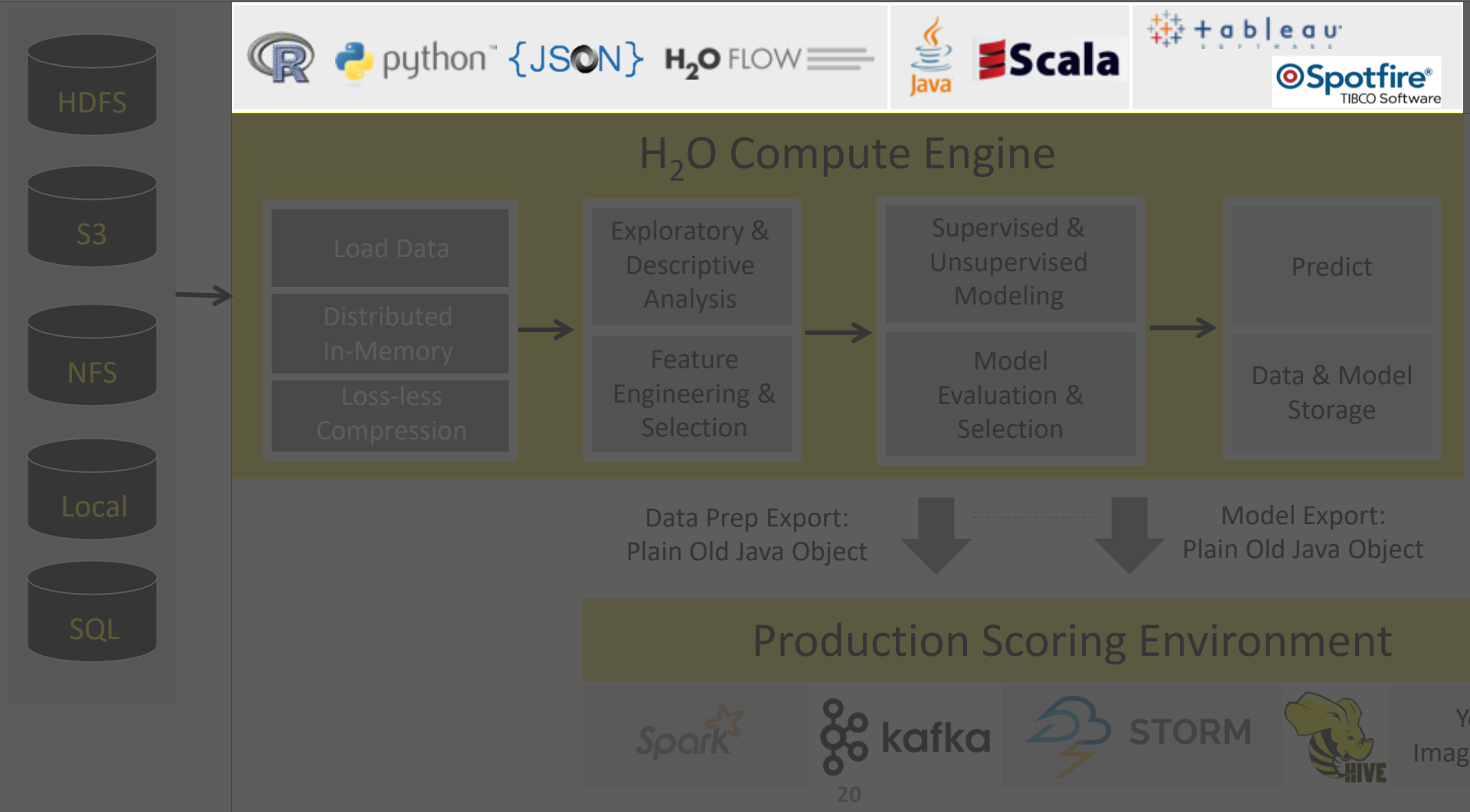
10 nodes: all
320 cores busy

real-time, interactive
model inspection in Flow

H_2O.ai **Deep Learning Model**

High Level Architecture

Multiple Interfaces



H_2O + R

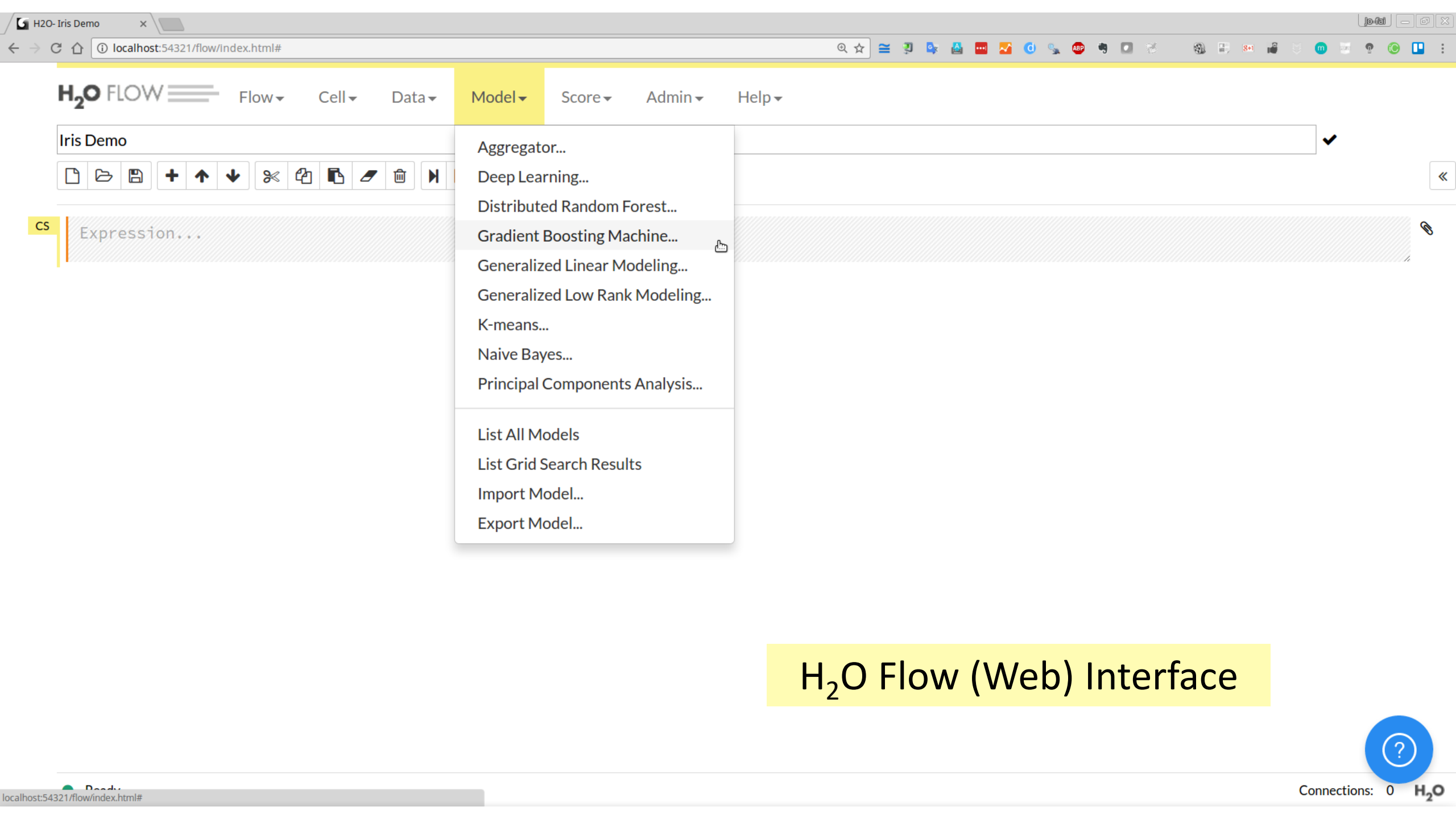
```
# ------------------------------------------------------------
# Train a H2O Model
# ------------------------------------------------------------

# Train three basic H2O models
model_drf <- h2o.randomForest(x = features,
                              y = target,
                              model_id = "iris_random_forest",
                              training_frame = d_iris)

model_gbm <- h2o.gbm(x = features,
                     y = target,
                     model_id = "iris_gbm",
                     training_frame = d_iris)

model_dnn <- h2o.deeplearning(x = features,
                              y = target,
                              model_id = "iris_deep_learning",
                              training_frame = d_iris)
```

H_2O + Python

Gradient Boosting Machines

```
# Build a Gradient Boosting Machines (GBM) model with default settings

# Import the function for GBM
from h2o.estimators.gbm import H2OGradientBoostingEstimator

# Set up GBM for regression
# Add a seed for reproducibility
gbm_default = H2OGradientBoostingEstimator(model_id = 'gbm_default', seed = 1234)

# Use .train() to build the model
gbm_default.train(x = features,
                  y = 'quality',
                  training_frame = wine_train)
```

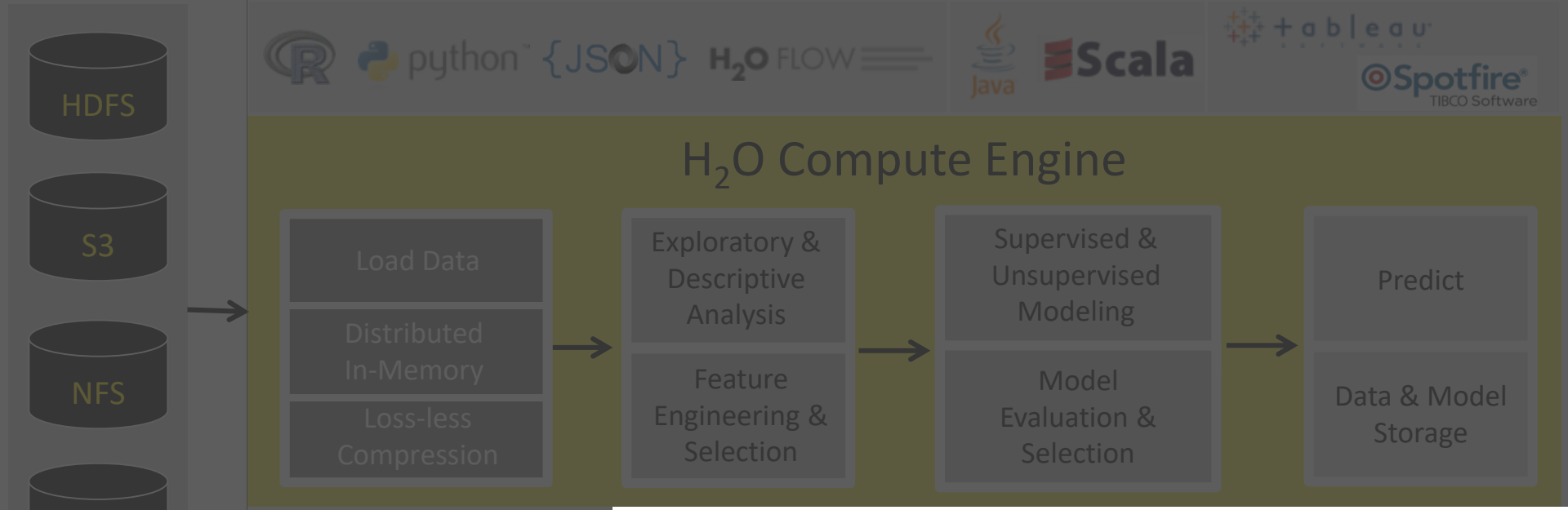
```
gbm Model Build progress: |████████████████████████████████████████████████| 100%
```



H_2O Flow (Web) Interface

High Level Architecture

Export Standalone Models for Production



Data Prep Export:
Plain Old Java Object

Model Export:
Plain Old Java Object

Production Scoring Environment

Spark | kafka | STORM | HIVE | Your Imagination

docs.h2o.ai

Languages

R

- Quick Start Video - R
- R Package Docs
- R Booklet
- Examples and Demos
- R FAQ
- Ensemble R Package Readme
- RSparkling Readme
- Migrating from H2O-2

Python

- Quick Start Video - Python
- Python Module Docs
- Python Booklet
- Examples and Demos
- Python FAQ
- PySparkling Readme 2.0 | 1.6
- skutil Docs

Java

- POJO and MOJO Model Javadoc
- H2O Core Javadoc
- H2O Algorithms Javadoc

Scala

Sparkling Water API	2.0	1.6
Sparkling Water Scaladoc	2.0	1.6
H2O Scaladoc	2.11	2.10

Tutorials, Examples, & Presentations

Tutorials and Blogs

- H2O Tutorials HTML | PDF
- H2O Blogs
- H2O University

Use Case Examples

Chicago crime prediction	R	Python	ScalaSW	PySW
Airlines delays prediction	R	Python	ScalaSW	PySW
Lending Club loan prediction	R	Python	ScalaSW	PySW
Ham or Spam	R	Python	ScalaSW	PySW
Prediction with prostate dataset	R	Python	ScalaSW	PySW

Presentations

- H2O Meetups
- H2O World 2014 Videos
- H2O World 2015 Videos
- Open Tour Chicago Videos
- Open Tour NYC Videos
- Open Tour Dallas Videos

New Developments


H2O.ai
Products
Industries
Customers
Community
Documentation
Download
Deep Water
H2O.ai
Caffe
mxnet
TensorFlow

Deep Water

Next-Gen Distributed Deep Learning with H_2O

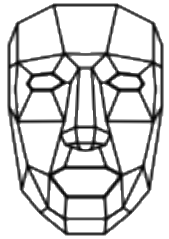
One Interface - GPU Enabled - Significant Performance Gains

Inherits All H_2O Properties in Scalability, Ease of Use and Deployment

TensorFlow Caffe H_2O.ai mxnet

Deep Water

H_2O integrates with existing GPU backends for significant performance gains



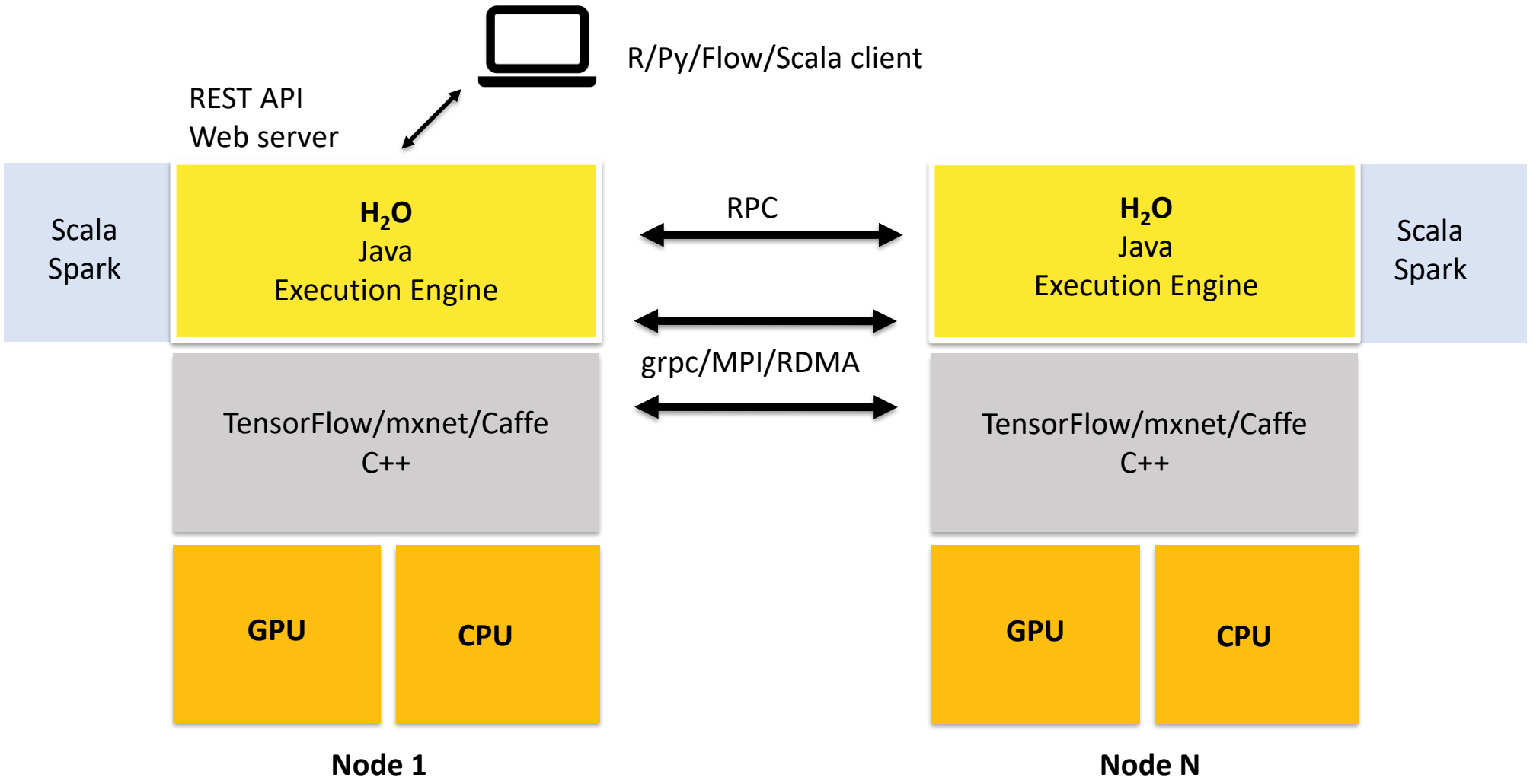
Convolutional Neural Networks enabling **Image, video, speech recognition**

Recurrent Neural Networks enabling **natural language processing, sequences, time series,** and more

Hybrid Neural Network Architectures enabling **speech to text translation, image captioning, scene parsing** and more

H_2O.ai

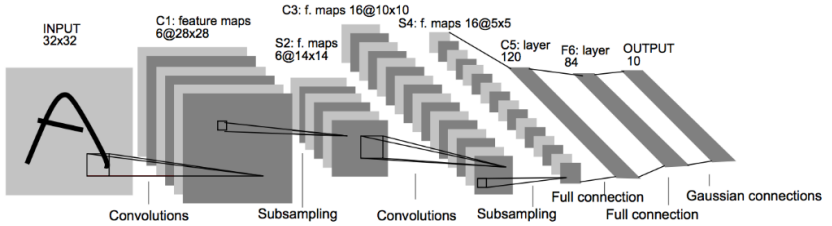
Deep Water Architecture



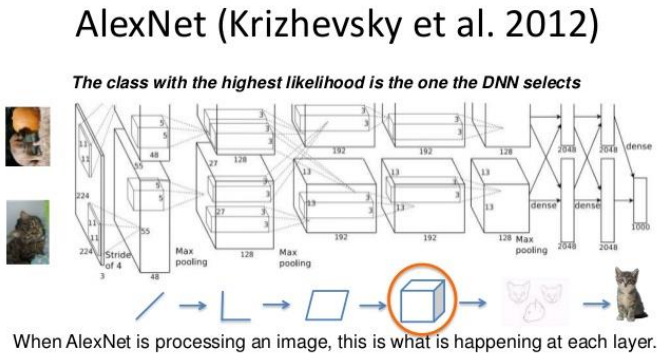
H2O.ai

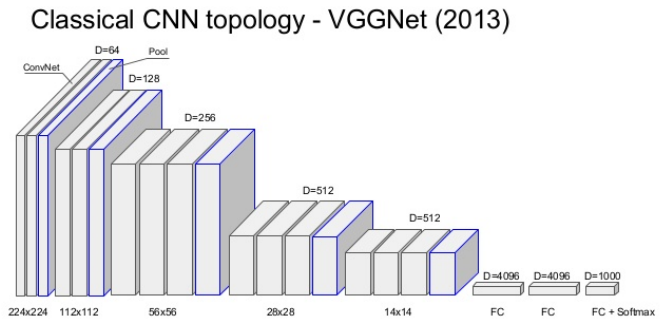
Available Networks in Deep Water

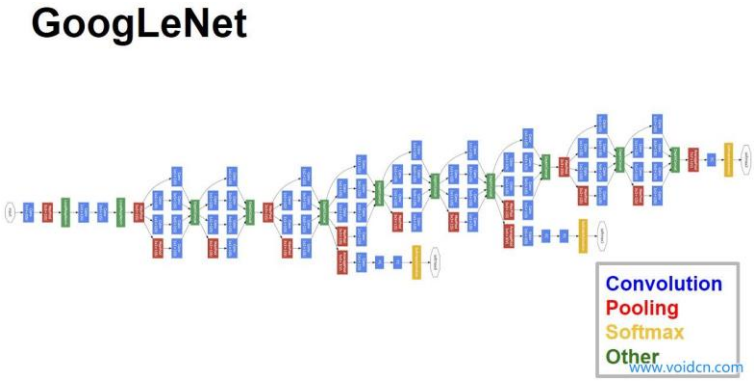
- LeNet
- AlexNet
- VGGNet
- Inception (GoogLeNet)
- ResNet (Deep Residual Learning)
- Build Your Own

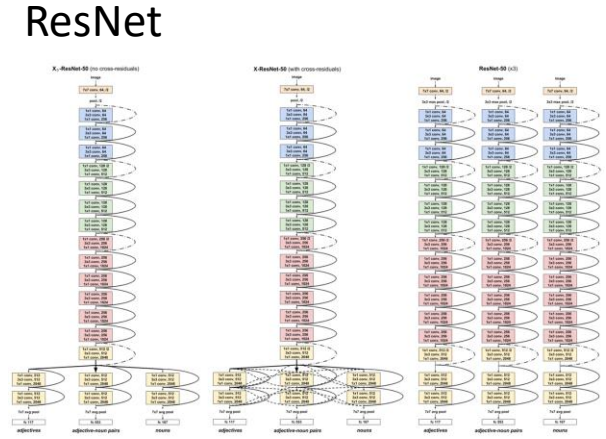


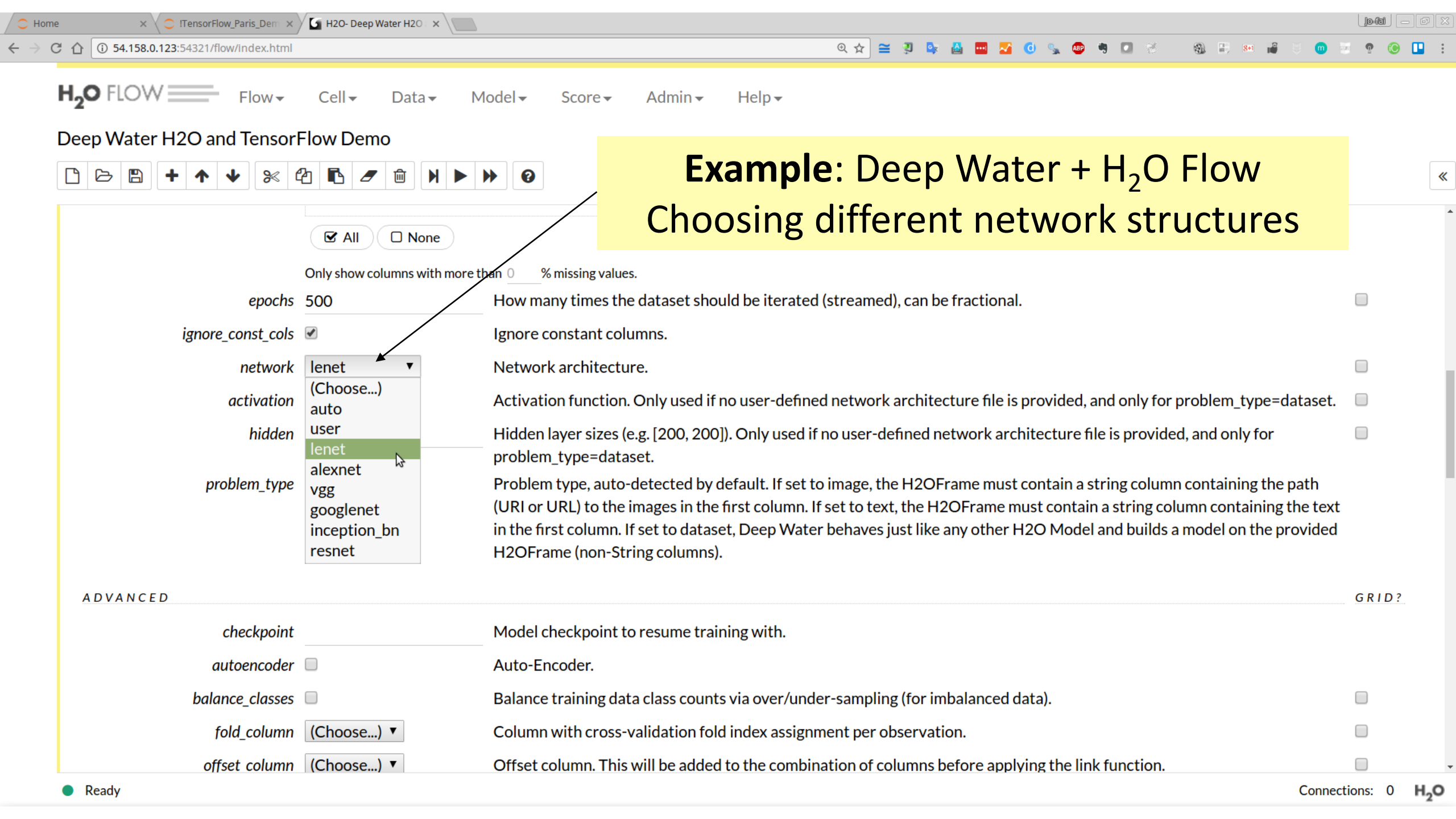
CNN called LeNet by Yann LeCun (1998)

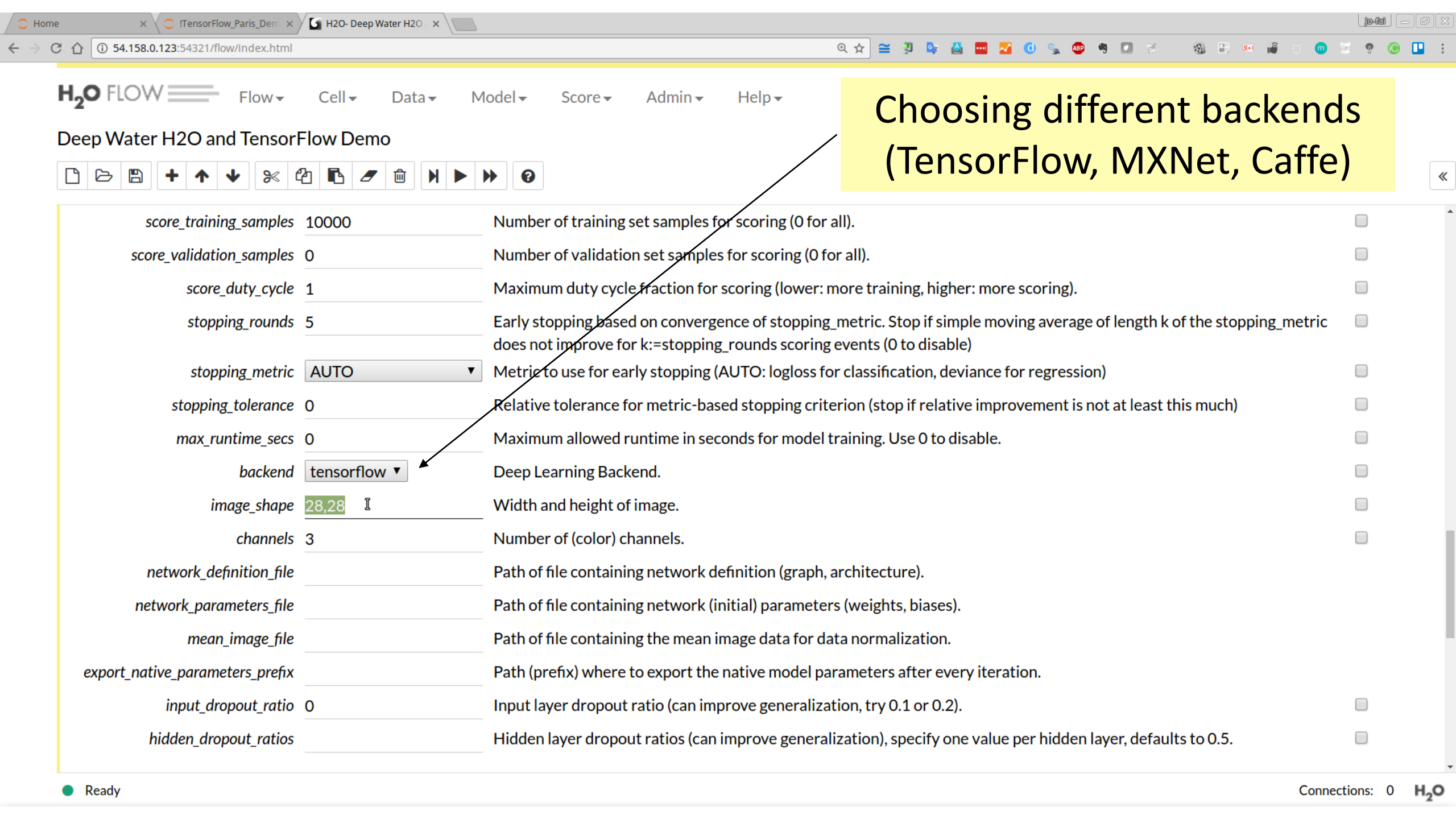









Home
ITensorFlow_Paris_Dem
H2O- Deep Water H2O
54.158.0.123:54321/flow/index.html
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Deep Water H2O and TensorFlow Demo
Example: Deep Water + H2O Flow
Choosing different network structures
All
None
Only show columns with more than 0 % missing values.
epochs
500
How many times the dataset should be iterated (streamed), can be fractional.
ignore_const_cols
Ignore constant columns.
network
lenet
Network architecture.
activation
Activation function. Only used if no user-defined network architecture file is provided, and only for problem_type=dataset.
hidden
Hidden layer sizes (e.g. [200, 200]). Only used if no user-defined network architecture file is provided, and only for problem_type=dataset.
problem_type
Problem type, auto-detected by default. If set to image, the H2OFrame must contain a string column containing the path (URI or URL) to the images in the first column. If set to text, the H2OFrame must contain a string column containing the text in the first column. If set to dataset, Deep Water behaves just like any other H2O Model and builds a model on the provided H2OFrame (non-String columns).
(Choose...)
auto
user
lenet
alexnet
vgg
googlenet
inception_bn
resnet
ADVANCED
GRID?
checkpoint
Model checkpoint to resume training with.
autoencoder
Auto-Encoder.
balance_classes
Balance training data class counts via over/under-sampling (for imbalanced data).
fold_column
(Choose...)
Column with cross-validation fold index assignment per observation.
offset_column
(Choose...)
Offset column. This will be added to the combination of columns before applying the link function.
Ready
Connections: 0
H2O


Home
ITensorFlow_Paris_Dem
H2O- Deep Water H2O
54.158.0.123:54321/flow/index.html
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Deep Water H2O and TensorFlow Demo
Choosing different backends
(TensorFlow, MXNet, Caffe)
score_training_samples 10000 Number of training set samples for scoring (0 for all).
score_validation_samples 0 Number of validation set samples for scoring (0 for all).
score_duty_cycle 1 Maximum duty cycle fraction for scoring (lower: more training, higher: more scoring).
stopping_rounds 5 Early stopping based on convergence of stopping_metric. Stop if simple moving average of length k of the stopping_metric does not improve for k:=stopping_rounds scoring events (0 to disable)
stopping_metric AUTO Metric to use for early stopping (AUTO: logloss for classification, deviance for regression)
stopping_tolerance 0 Relative tolerance for metric-based stopping criterion (stop if relative improvement is not at least this much)
max_runtime_secs 0 Maximum allowed runtime in seconds for model training. Use 0 to disable.
backend tensorflow Deep Learning Backend.
image_shape 28,28 Width and height of image.
channels 3 Number of (color) channels.
network_definition_file Path of file containing network definition (graph, architecture).
network_parameters_file Path of file containing network (initial) parameters (weights, biases).
mean_image_file Path of file containing the mean image data for data normalization.
export_native_parameters_prefix Path (prefix) where to export the native model parameters after every iteration.
input_dropout_ratio 0 Input layer dropout ratio (can improve generalization, try 0.1 or 0.2).
hidden_dropout_ratios Hidden layer dropout ratios (can improve generalization), specify one value per hidden layer, defaults to 0.5.
Ready
Connections: 0
H2O

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

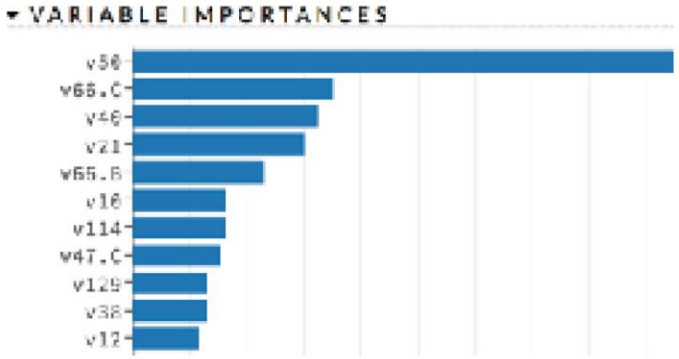
Select an algorithm: XGBoost

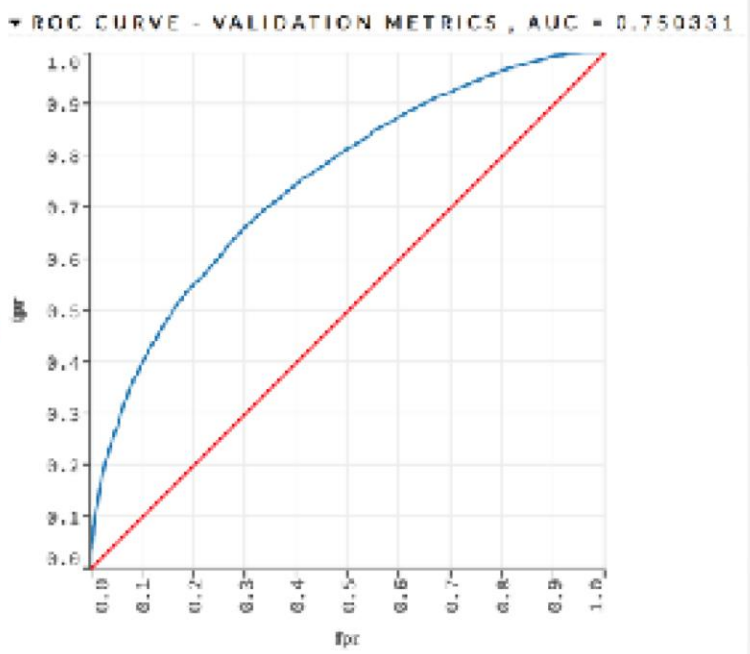
booster gbtree

reg_lambda 1

reg_alpha (Choose...)

backend auto / gpu / cpu






```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 375.20                 Driver Version: 375.20                    |
|-------------------------------+----------------------+----------------------+
| 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 / 180W |   2040MiB /  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
```

H2O.ai

Kaggle dataset: BNP Paribas Cardif Claims Management

Stacked Ensembles

Stacked Ensembles in H2O

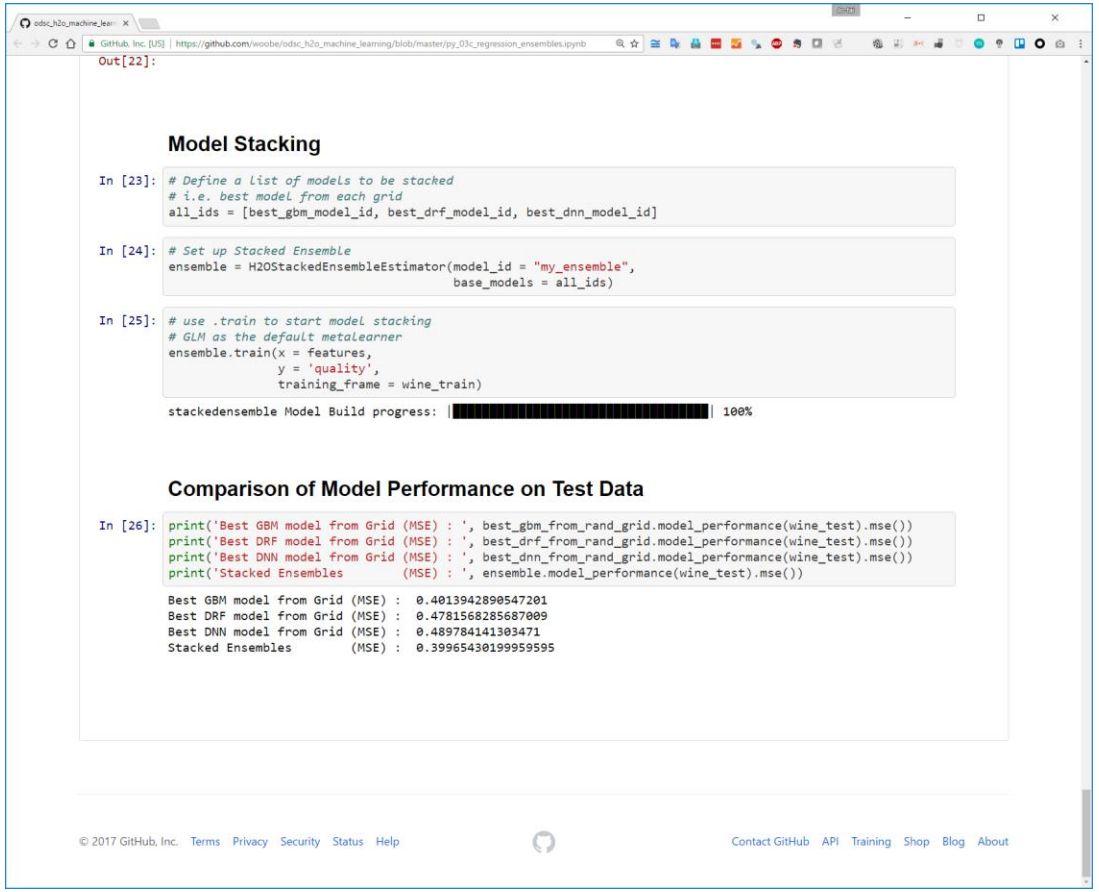
https://github.com/h2oai/h2o-meetups/blob/master/2017_02_23_Metis_SF_Sacked_Ensembles_Deep_Water/stacked_ensembles_in_h2o_feb2017.pdf

February 2017

Erin LeDell Ph.D.
Machine Learning Scientist

H2O.ai

Available in both Python and R



Out[22]:

Model Stacking

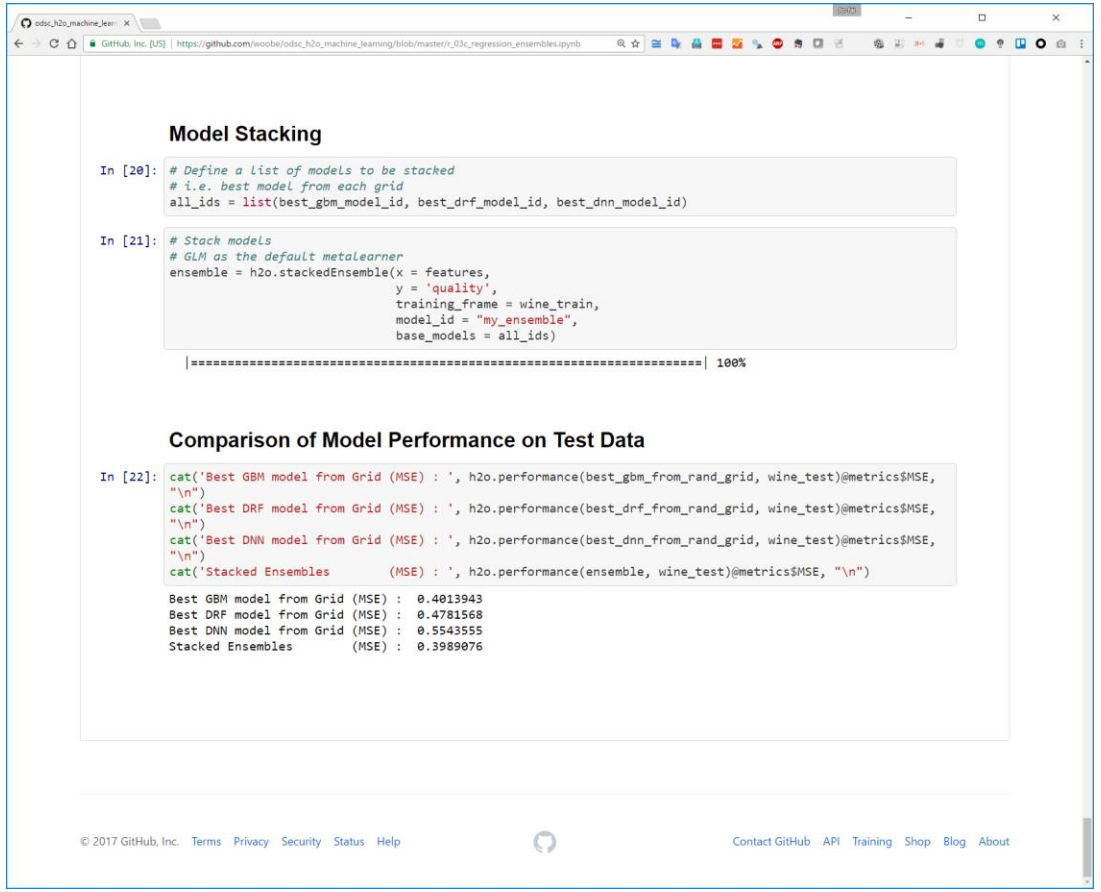
```
In [23]: # Define a list of models to be stacked
         # i.e. best model from each grid
         all_ids = [best_gbm_model_id, best_drf_model_id, best_dnn_model_id]

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) :  0.4781568285687009
Best DNN model from Grid (MSE) :  0.489784141303471
Stacked Ensembles        (MSE) :  0.39965430199959595
```

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)@metrics$MSE,
         "\n")
         cat('Best DRF model from Grid (MSE) : ', h2o.performance(best_drf_from_rand_grid, wine_test)@metrics$MSE,
         "\n")
         cat('Best DNN model from Grid (MSE) : ', h2o.performance(best_dnn_from_rand_grid, wine_test)@metrics$MSE,
         "\n")
         cat('Stacked Ensembles        (MSE) : ', h2o.performance(ensemble, wine_test)@metrics$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
```


https://github.com/woobe/odsc_h2o_machine_learning

H2O.ai

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!

Model Interpretation

https://www.oreilly.com/ideas/ideas-on-interpreting-machine-learning

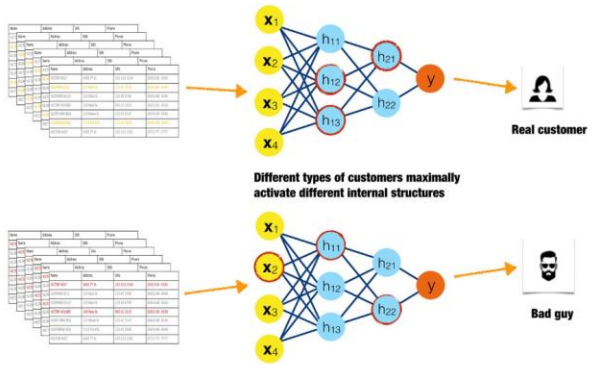
Ideas on interpreting machine learning

Mix-and-match approaches for visualizing data and interpreting machine learning models and results.

By Patrick Hall, Wen Phan and SriSatish Ambati, March 15, 2017

You've probably heard by now that machine learning algorithms can use big data to predict whether a donor will give to a charity, whether an infant in a NICU will develop sepsis, whether a customer will respond to an ad, and on and on. Machine learning can even drive cars and predict elections. ... Err, wait. Can it? I believe it can, but these recent high-profile hiccups should leave everyone who works with data (big or not) and machine learning algorithms asking themselves some very hard questions: do I understand my data? Do I understand the model and answers my machine learning algorithm is giving me? And do I trust these answers?

Unfortunately, the complexity that bestows the extraordinary predictive



Inputs activating different neurons in a neural network. (source: Image courtesy of Patrick Hall and the h2o.ai team, used with permission)

H2O.ai

Correlation Graphs

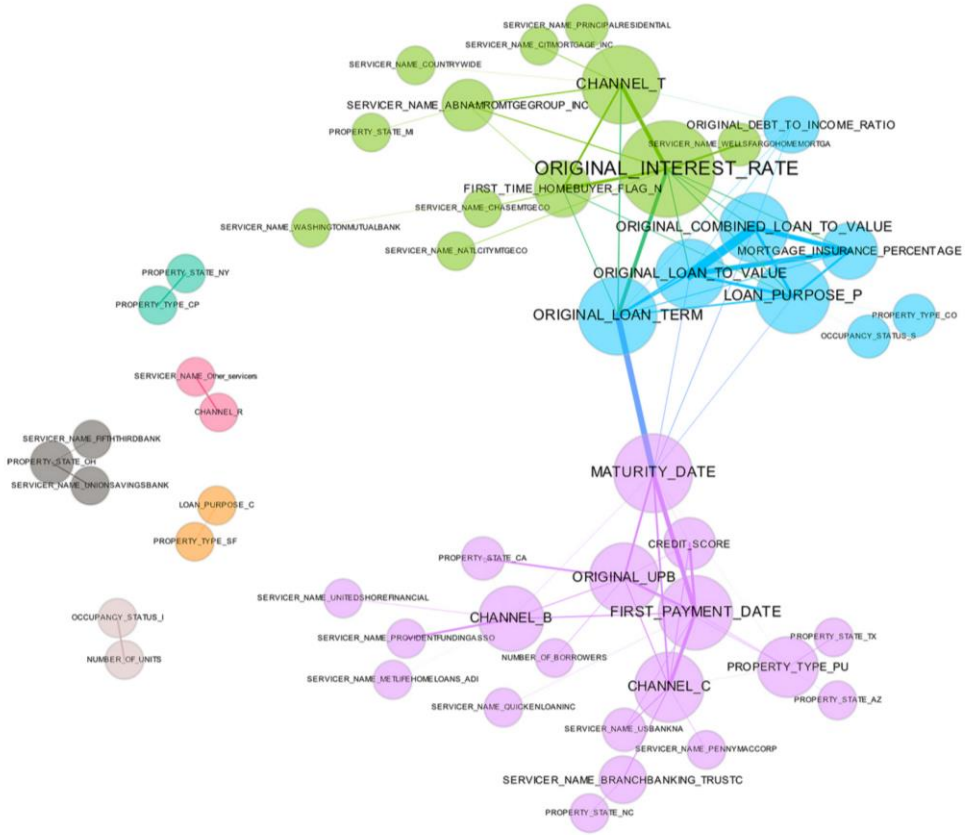


Figure 3. A correlation graph representing loans made by a large financial firm. Figure courtesy of Patrick Hall and the H2O.ai team.

Partial dependence plots

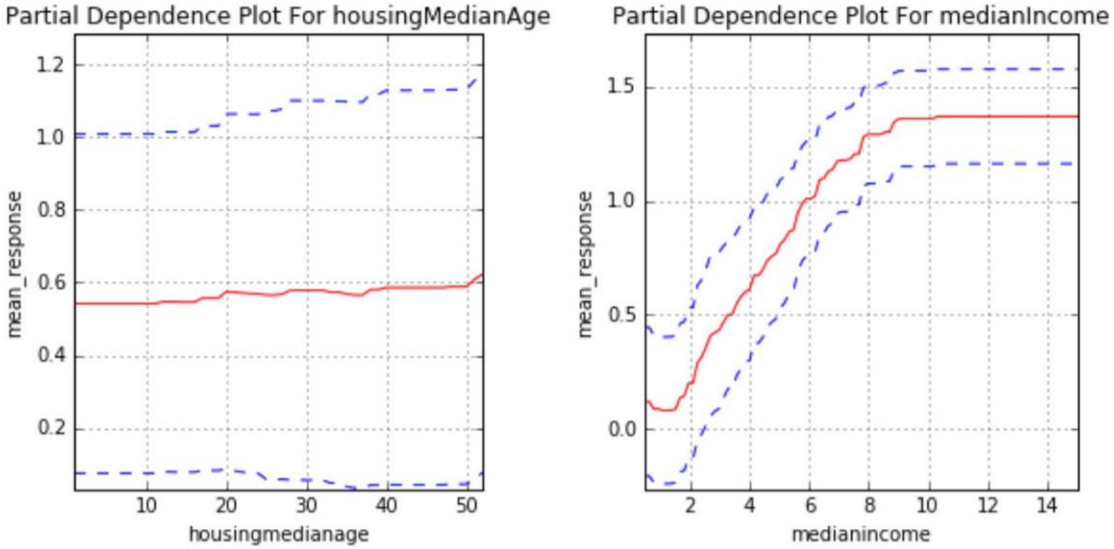


Figure 5. One-dimensional partial dependence plots from a gradient boosted tree ensemble model of the well-known California housing data set. Image courtesy Patrick Hall and the H2O.ai team.

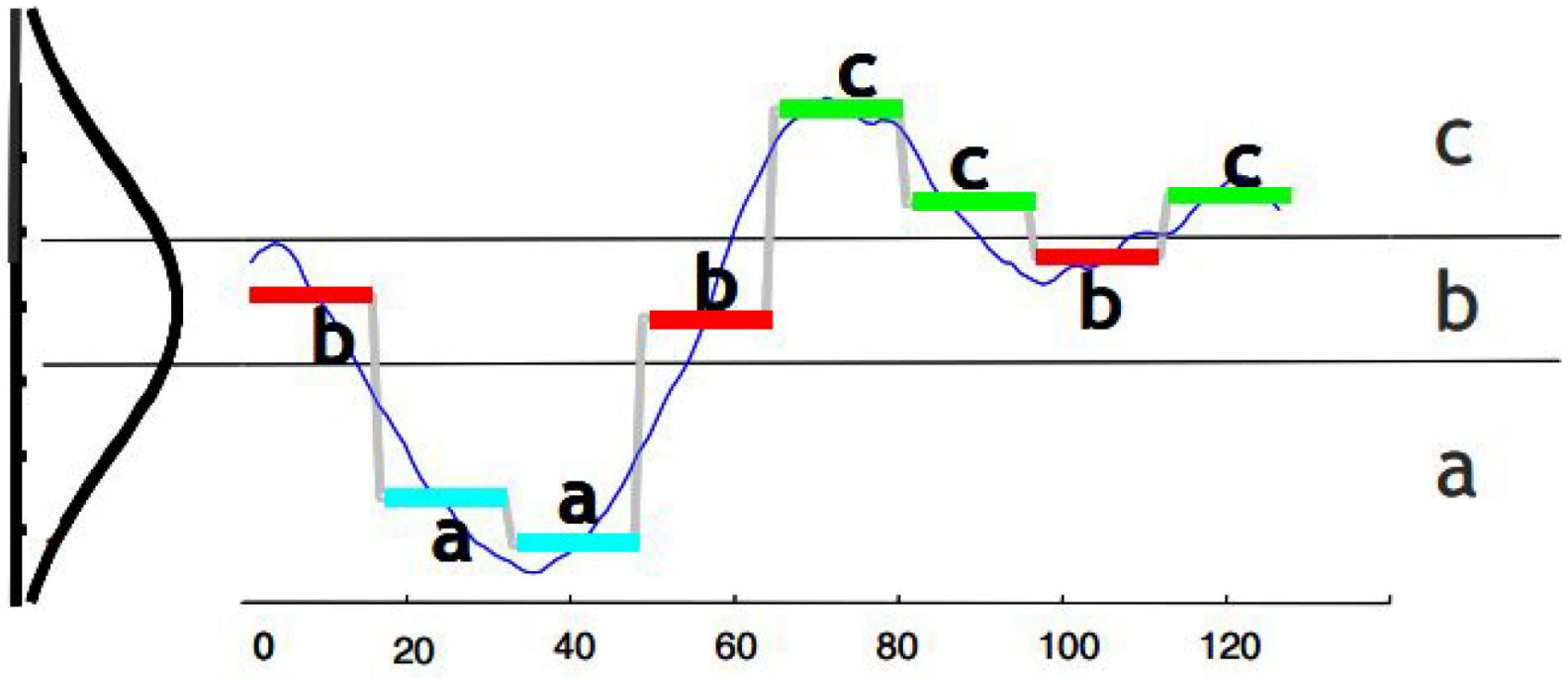
H2O.ai

Time Series in H_2O

ISAX

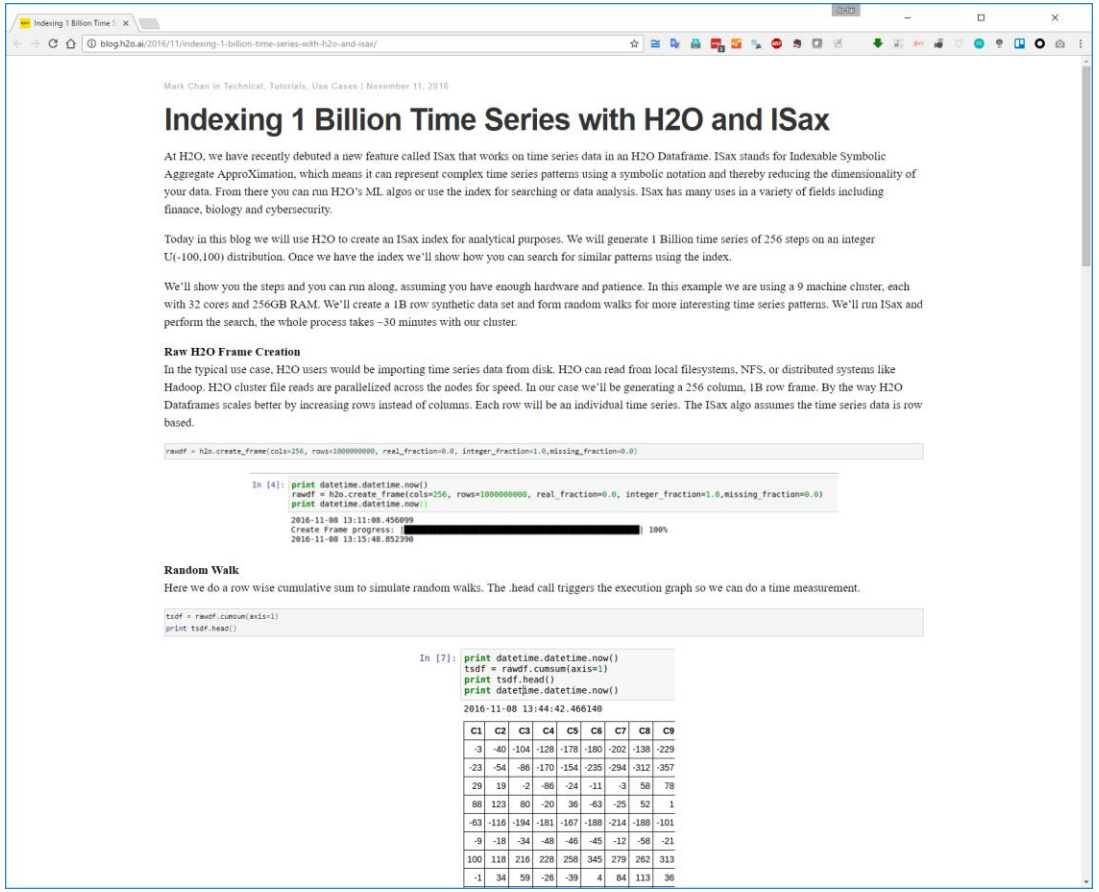
- Time series data compression algorithm, implemented on H2O's distributed architecture
- Find groups of similar time series patterns in one billion
  - http://blog.h2o.ai/2016/11/indexing-1-billion-time-series-with-h2o-and-isax/
- ISAX applied to fields such as IoT, Finance, BioInformatics,Image/Sound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

ISAX



Blog Post

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



Mark Chan in Technical, Tutorials, Use Cases | November 11, 2016

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 column, 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 13:11:08.456099
Create Frame progress: [##################################################] 100%
2016-11-08 13:15:48.852390
```

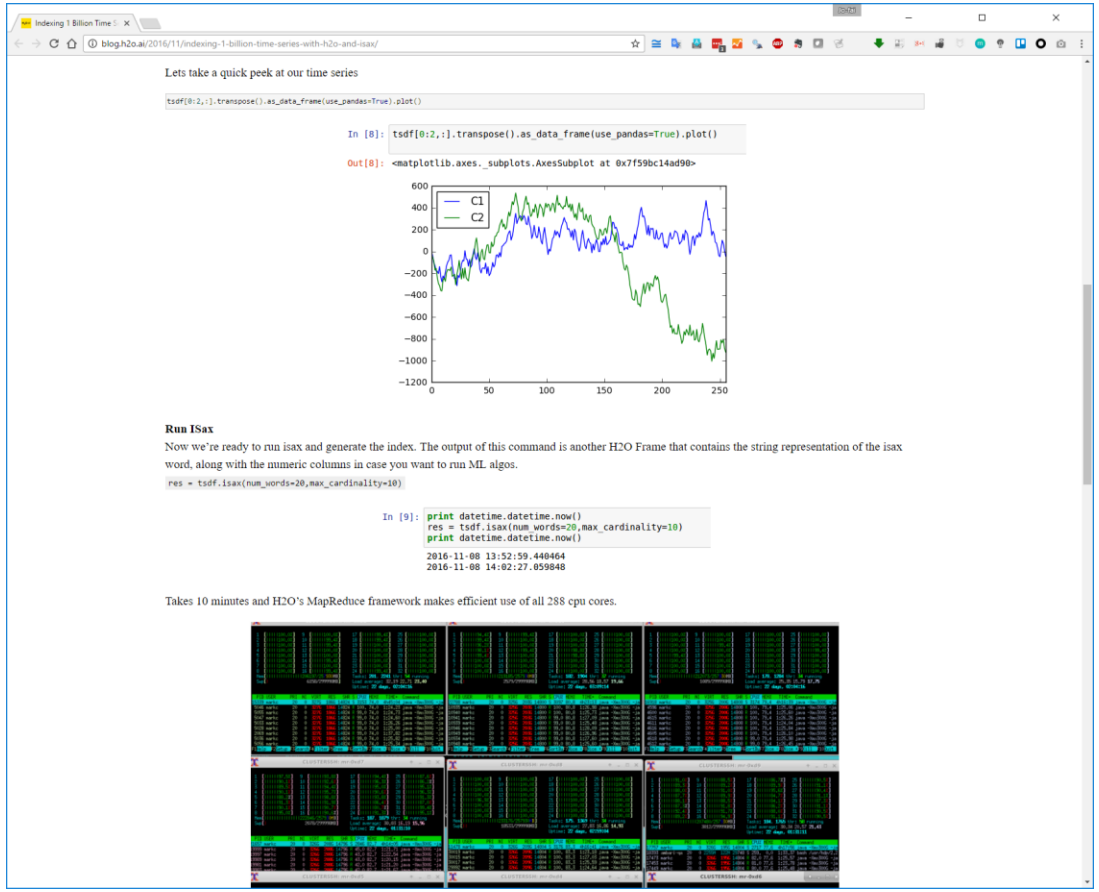
Random Walk

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 14:02:27.059848
```

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

H2O.ai

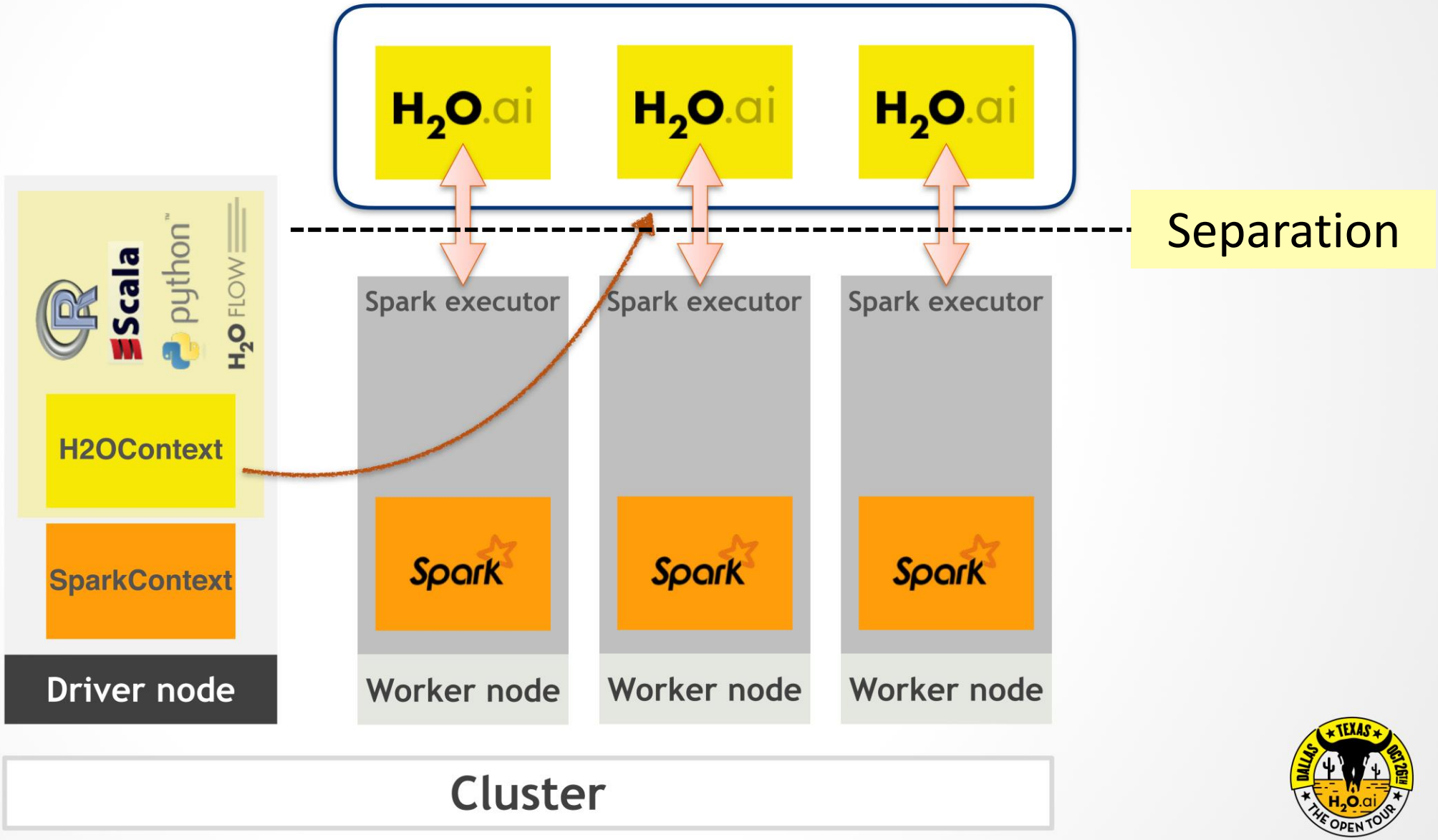
Sparkling Water

H_2O's integration with Spark – latest developments

External H_2O Backend for Sparkling Water

- High Availability Mode
  - Separating Spark and H_2O
  - while preserving same API
- Advantages
  - H_2O does not crash when Spark executor goes down
  - Better resource management since resources can be planned per tool
- Disadvantages
  - Transfer overhead between Spark and H_2O processes
  - Under measurement with cooperation of a customer
- Links
  - https://github.com/h2oai/sparkling-water/blob/master/doc/backends.md

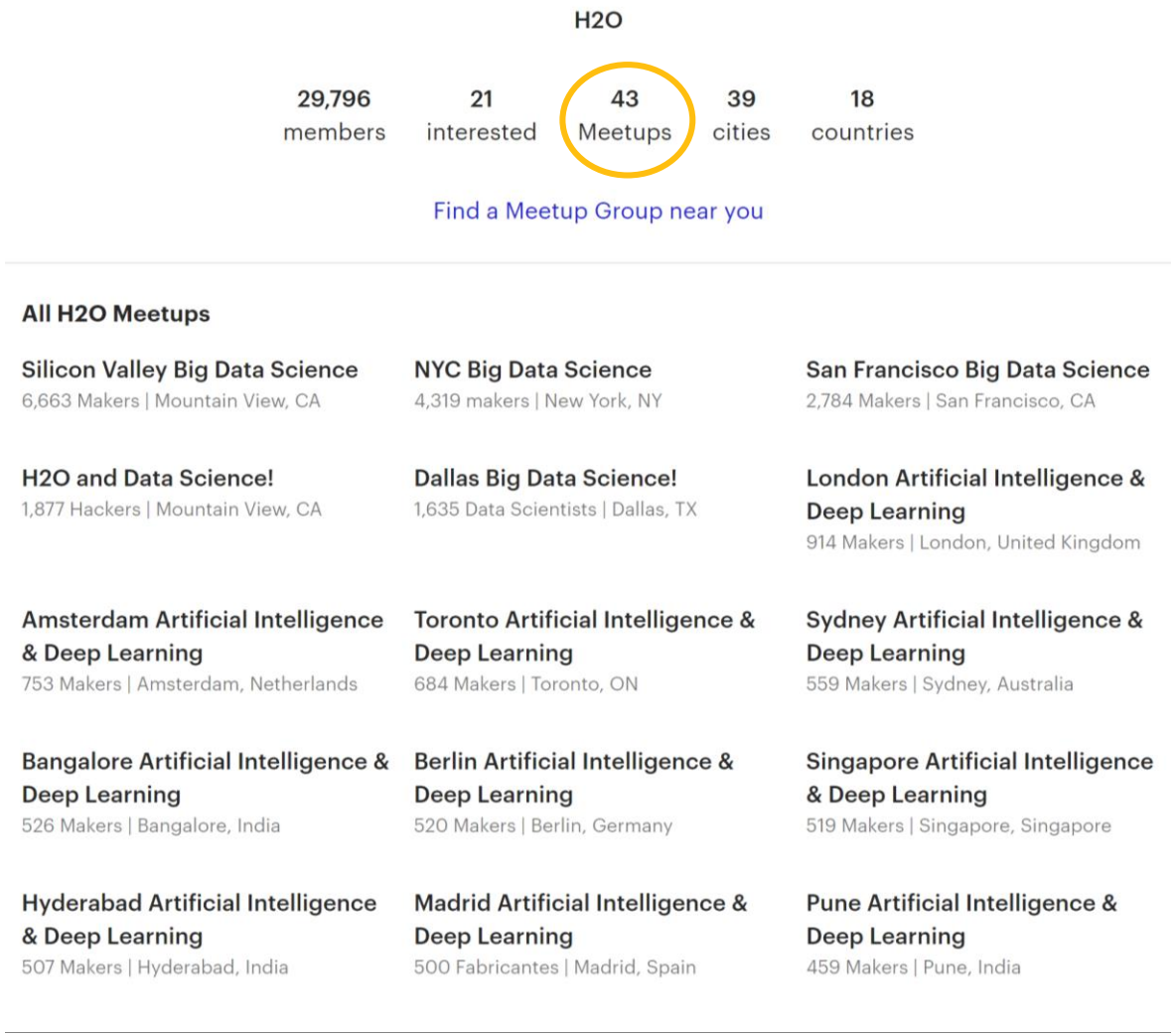
High Availability



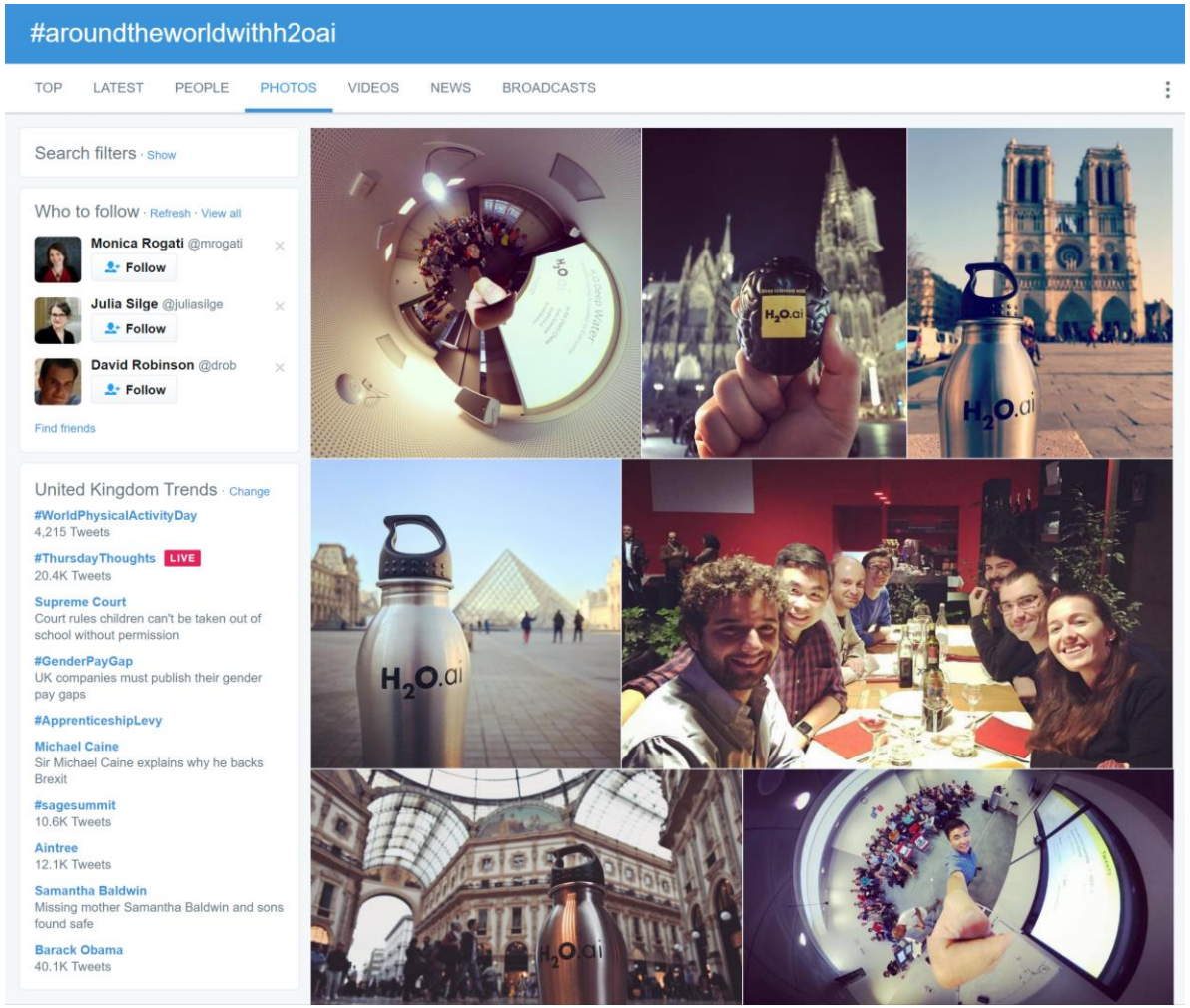
H_2O Community

H_2O Meetup Groups

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



#AroundTheWorldWithH2Oai



Search

Ask Question

h2o

search

Advanced Search Tips

1,130 results

relevance newest votes active

3 votes

1 answer

Q: Subsetting in H2O R

I have a **h2o** object. The standard R for subset sub1<-trans[trans$Type==1,] I tried the same in **h2o**. It is not working sub1<-trans[trans$Type==1,] I also tried sub1<-h2o.exec(trans[trans$Type==1,]) note* trans is a **h2o** data Object. Any idea to do it in **h2o**? Thanks …

r subset h2o

asked Nov 28 '14 by chee.work.stuff

5 votes

4 answers

Q: h2o implementation in R

I am learning **h2o** package now, I installed **h2o** package from CRAN and couln't run this code &#35&#35 To import small iris data file from H\ :sub:`2`\ O's package irisPath = system.file("extdata … ", "iris.csv", package="**h2o**") iris.hex = h2o.importFile(localH2O, path = irisPath, key = "iris.hex") I am getting the below error, Error in h2o.importFile(localH2O, path = irisPath, key …

r h2o

asked Aug 22 '16 by varun

0 votes

1 answer

Q: MAPE metric at h2O

What is correct way to implement MAPE under **h2o** framework? I am interested to convert below function to **h2o** concept def mape(a, b): mask = a <> 0 return (np.fabs(a - b)/a)[mask].mean() …

python python-3.x pandas dataframe h2o

asked Mar 29 by SpanishBoy

1 vote

Q: H2O Python module build problems

This is the error encountered when running ./gradlew build in the **h2o**-3 repository. :**h2o**-py:buildDistFound packages: ['**h2o**', 'h2o.backend', 'h2o.estimators', 'h2o.grid', 'h2o.model', 'h2o.schemas … : invalid command

results found containing **h2o**

Hot Network Questions

- What can the United Nations do after the chemical attack in Syria?
- Does a LED also emit light when conducting in avalanche mode?
- Can a high voltage line kill a person without touching it?
- How do I compute the mean from ASCII file data in bash?

PyData Amsterdam 2017

PyData

HOME VENUE ABOUT SPONSORS SCHEDULE

Conference Schedule

View past PyData event schedules here.

Tutorial Sessions — Friday April 7, 2017

	Big Room	Small Room
8:30	Registration	
9:00	From Fourier to deep convnets Ivo Everts	Pandas from the Inside / "Big Pandas" Stephen Simmons
12:00	Lunch	
13:00	Deep learning - advanced techniques Geoff French	So You Want to Be a Python Expert? James Powell
16:00	Hacking Space	Introduction to Machine Learning with H2O and Python Jo-fai Chow
18:30	PyData Amsterdam Meetup RSVP separately (Space is Limited!)	
22:00		

General Sessions — Saturday April 8, 2017

	Big Room	Small Room
8:00	Breakfast & Registration	
9:00	Opening Notes	
9:15	Ethical Machine Learning: Creating Fair Models in an Unjust World Katharine Jarmul	
10:00	Knowledge Repository Matthew Wardrop, Dan Frank	Creativity and AI: Deep Neural Nets "Going Wild" Roelof Pieters
10:45	Break	
11:00	Successfully applying Bayesian statistics to A/B testing in your business Ruben Mak	Detecting Clickbaits using Machine Learning Abhishek Thakur
11:45	Deep Reinforcement Learning: theory, intuition, code Maxim Lapan	Making contract documents fully searchable at KPN Gianluigi Bardelloni
12:30	Lunch	
13:30	Diagnosing Machine Learning Models Lucas Javier Bernardi	A Pythonic Tour of Neo4j and the Cypher Query Language Nigel Small
14:15	Training a TensorFlow model to detect lung nodules on CT scans Mark-Jan Harte	A practical guide to speed up your application with Asyncio Niels Denissen
15:00	Break	
15:15	Bayesian optimization with Scikit-Optimize Gilles Louppe	Different Strategies of Scaling H2O Machine Learning on Apache Spark Jakub Hava

H_2O Tutorial
Friday – 4:00 pm
(All Materials will be available on GitHub)

Lucas' Talk
Saturday – 1:30 pm

Sparkling Water Talk
Saturday – 3:15 pm

H_2O.ai

Thanks!

- Organizers & Sponsors

Booking.com

- Find us at PyData Conference
  - Live Demos

- Code, Slides & Documents
  - bit.ly/h2o_meetups
  - docs.h2o.ai

- Contact
  - joe@h2o.ai
  - @matlabulous
  - github.com/woobe

- Please search/ask questions on **Stack Overflow**
  - Use the tag `**h2o**` (not H2 zero)

H2O.ai