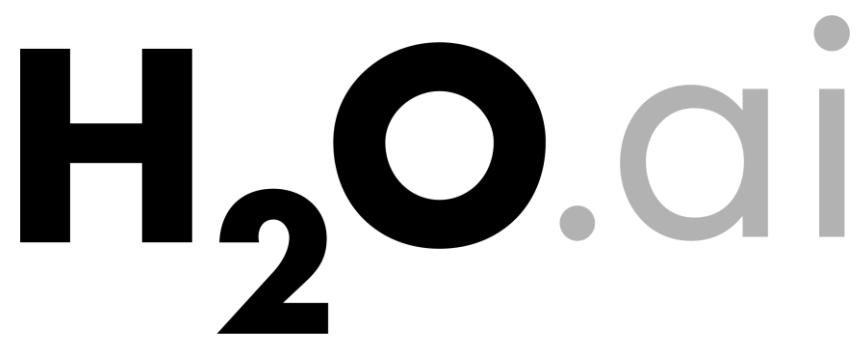


H₂O Deep Water

Making Deep Learning Accessible to Everyone



Jo-fai (Joe) Chow
Data Scientist
joe@h2o.ai
[@matlabulous](https://twitter.com/matlabulous)

RugBcn Barcelona R Users Group
22nd March, 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 H₂O in 2014
- Data Scientist
 - From 2015
 - Virgin Media (UK)
 - Domino Data Lab (Silicon Valley)
 - H₂O.ai (Silicon Valley)
- How?
 - bit.ly/joe_kaggle_story

I ❤️ R



Jo-fai Chow

woobe

Civil Engineer turned Data Scientist

H2O.ai

United Kingdom

jofai.chow@gmail.com

<http://www.jofaichow.co.uk/>

Organizations



Overview Repositories 47 Stars 402 Followers 140 Following 29

Popular repositories

blenditbayes

Code used in my blog "Blend it like a Bayesian!"

R ★ 77 ⚡ 82

deepr

An R package to streamline the training, fine-tuning and predicting processes for deep learning based on 'darch' and 'deepnet'.

R ★ 41 ⚡ 16

rPlotter

Wrapper functions that make plotting in R a lot easier for beginners.

R ★ 30 ⚡ 4

rCrimemap

This is the next generation of CrimeMap!

R ★ 22 ⚡ 8

rugsmaps

This app is my submission to the visualization contest held by Revolution Analytics.

R ★ 19 ⚡ 18

Apps

Repository for my R (Shiny) web applications.

R ★ 16 ⚡ 37

H₂O.ai



Crime Data Visualisation

INTRODUCTION
This ShinyApp allows you to download and visualise crime data in England, Wales & Northern Islands from data.police.uk. The data is made available under the Open Government License. For more information, see my original blog post.

USAGE
Simply enter a location of your choice (e.g. Oxford), choose the first month for data collection (e.g. Jan 2012), decide how many months of data you need and then click "update". There are some more settings available for you to customise the plots. Scroll down and try them out!

READY?
Continue to scroll down and modify the settings. Come back and click this when you are ready to render new plots.
[Update Graphics and Tables](#)

BASIC SETTINGS
Enter a Location of Interest:

Examples: London, Wembley Stadium, M16 GRA etc.
First Month of Data Collection:

Length of Analysis (Months):

Note: Data is available from Dec 2010 to Sep 2013. There is inconsistency in 2010-2011 records so I have omitted them for now. It takes longer to render the plots when you increase this number.

MAP SETTINGS
Choose Facet Type:
 none
 choropleth
Choose Google Map Type:
 roadmap
 satellite
 High Resolution?
 Black & White?
Zoom Level (Recommended - 14):

DENSITY PLOT SETTINGS
Alpha Range:



My First Data Viz & Shiny App Experience
[CrimeMap \(2013\)](#)

Revolutions
Daily news about using open source R for big data analysis, predictive modeling, data science, and visualization since 2008

[« How to integrate R with your calendar](#) | [Main](#) | [Entering the field as a data scientist with certification »](#)

August 21, 2014

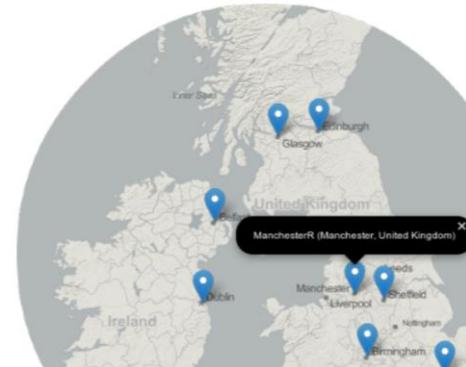
Revolution Analytics' User Group Map Contest has a Winner

by Joseph Rickert

We are pleased to announce that [Jo-fai Chow](#) is the winner of the Revolution Analytics contest. Jo-fai's entry, which was implemented as a [Shiny project](#), may be viewed by clicking on the figure below.

R User Groups Around the World

[About](#) [Maps](#) [Data](#) [More](#)



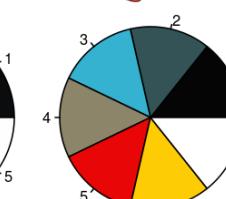
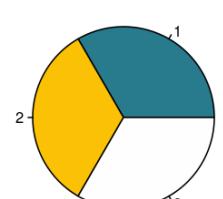
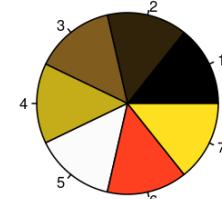
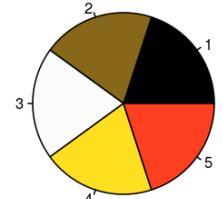
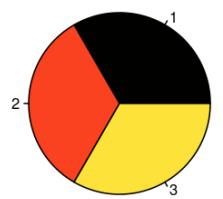
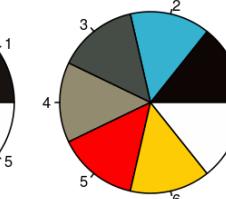
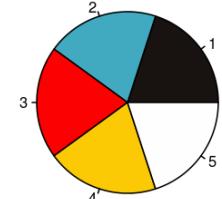
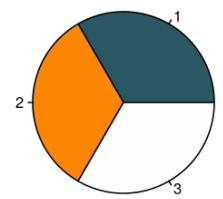
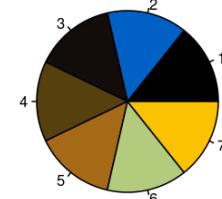
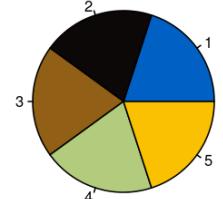
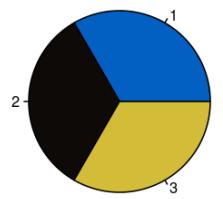
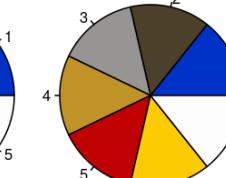
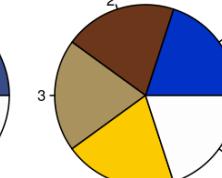
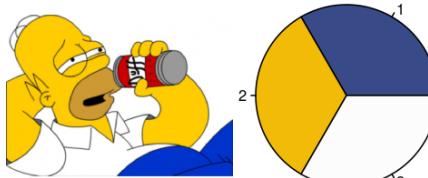
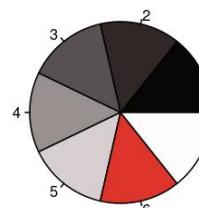
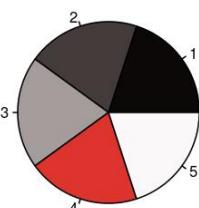
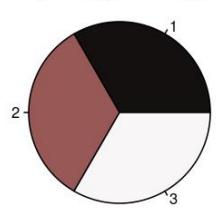
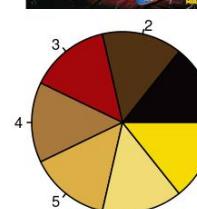
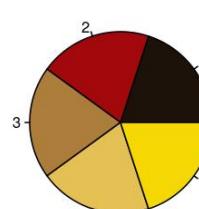
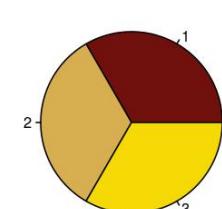
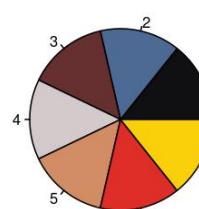
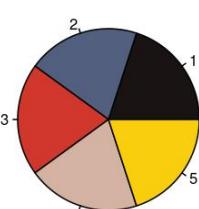
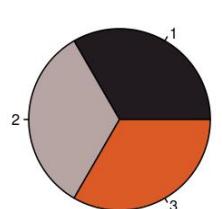
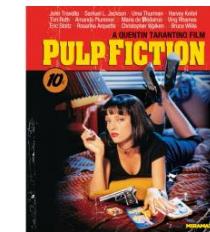
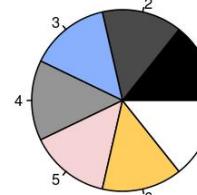
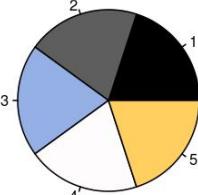
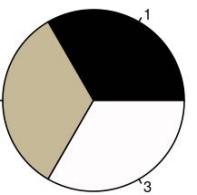
 **Jo-fai (Joe) Chow**
@matlabulous

Thank you very much @RevolutionR
@revodavid @RevoJoe #iloveR
[bit.ly/rugsmaps](#) #Shiny #rMaps



Revolution Analytics' Data Viz Contest
[RUGSMAPS \(2014\)](#)

I ❤ R



Developing R Packages for Fun
[rPlotter](#) (2014)

Agenda

- About H₂O.ai
 - Company
 - Machine Learning Platform
- Deep Learning Tools
 - TensorFlow, MXNet, Caffe, H₂O
- Deep Water
 - Motivation / Benefits
 - Interfaces / Demo
- Other News
- Conclusions

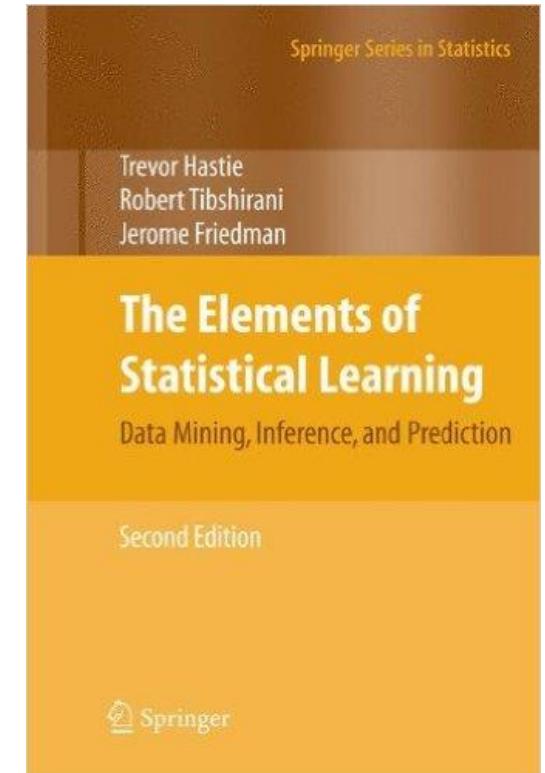


About H₂O.ai

Company Overview

| | |
|---------------------|---|
| Founded | 2011 Venture-backed, debuted in 2012 |
| Products | <ul style="list-style-type: none">• H₂O 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 <ul style="list-style-type: none">• Distributed Systems Engineers doing Machine Learning• World-class visualization designers |
| Headquarters | Mountain View, CA |





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*

Figure 1. Magic Quadrant for Data Science Platforms



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

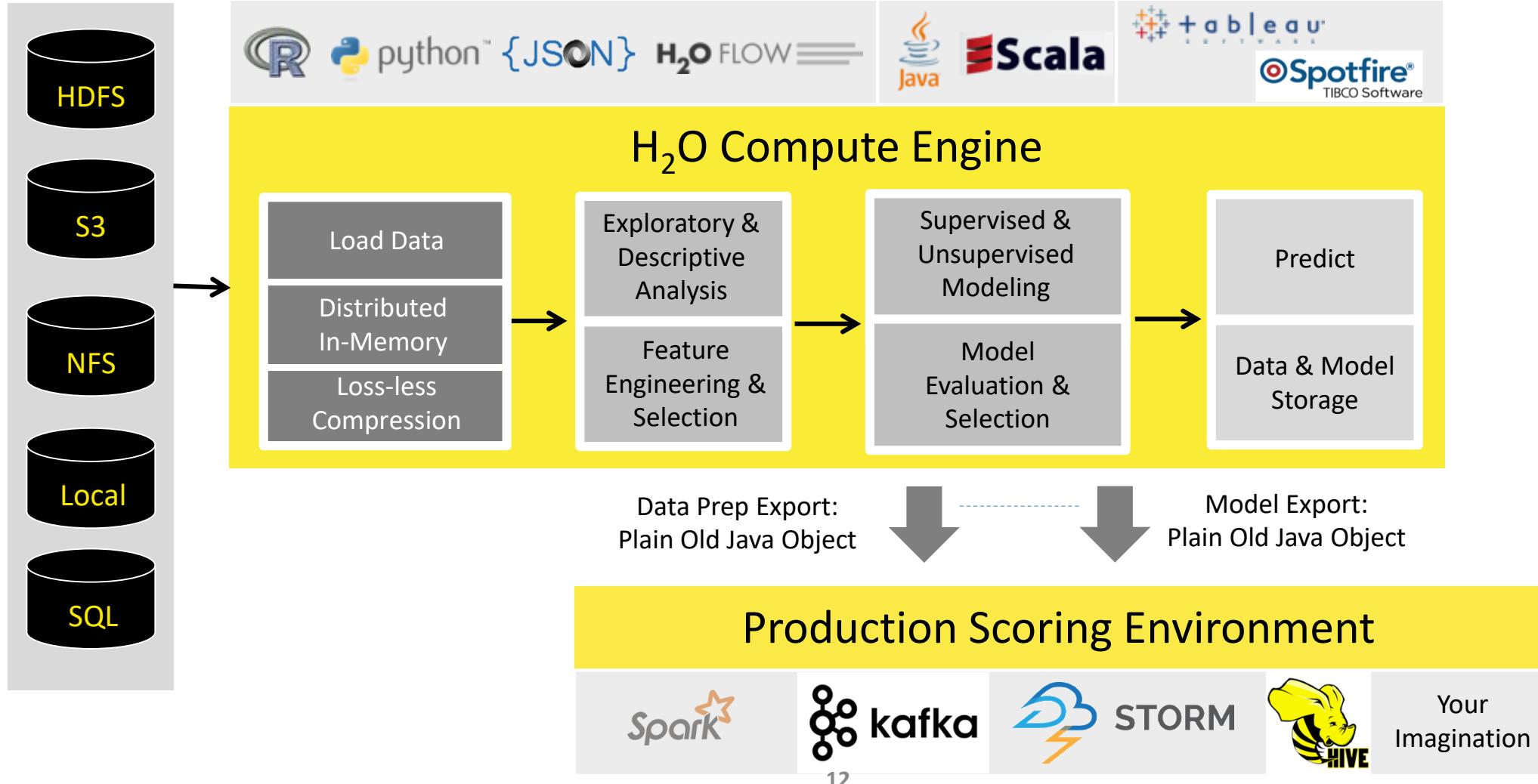
We are thrilled to be named a Visionary among the 16 vendors included in Gartner's 2017 Magic Quadrant for Data Science Platforms. As a Visionary we believe we are positioned highest in Ability to Execute for companies of our size and scale.

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.

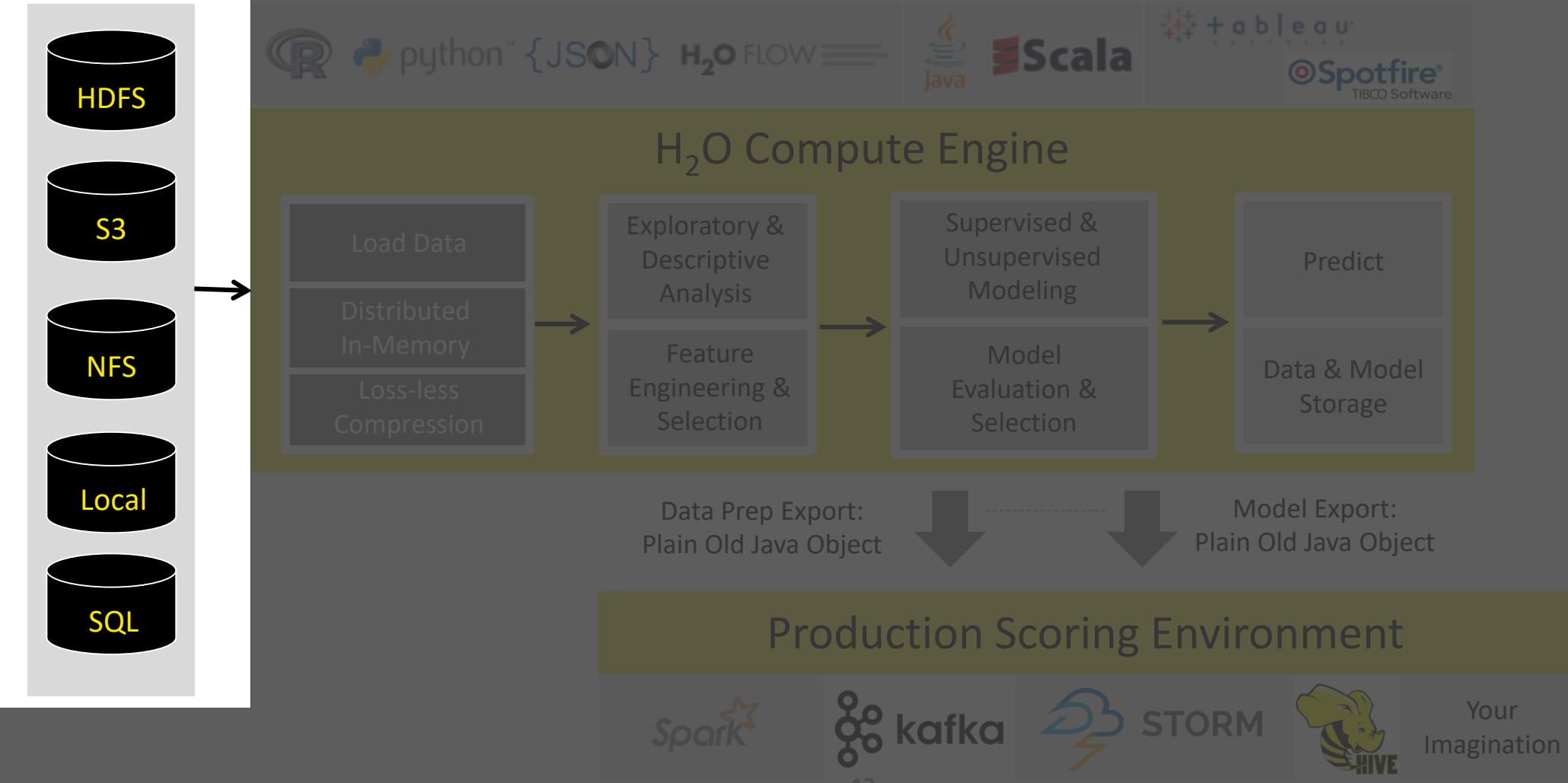
H₂O Machine Learning Platform

High Level Architecture



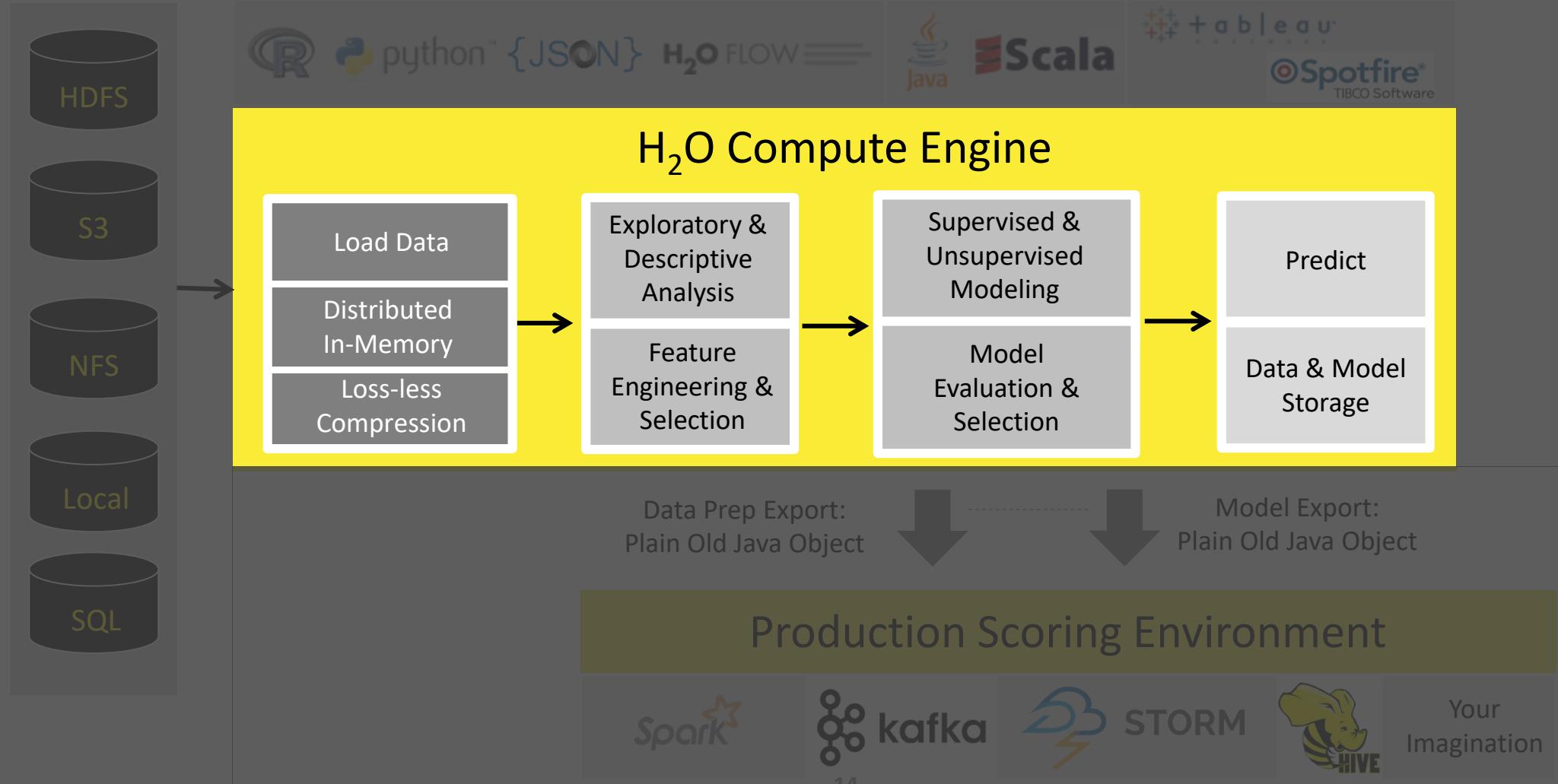
High Level Architecture

Import Data from
Multiple Sources



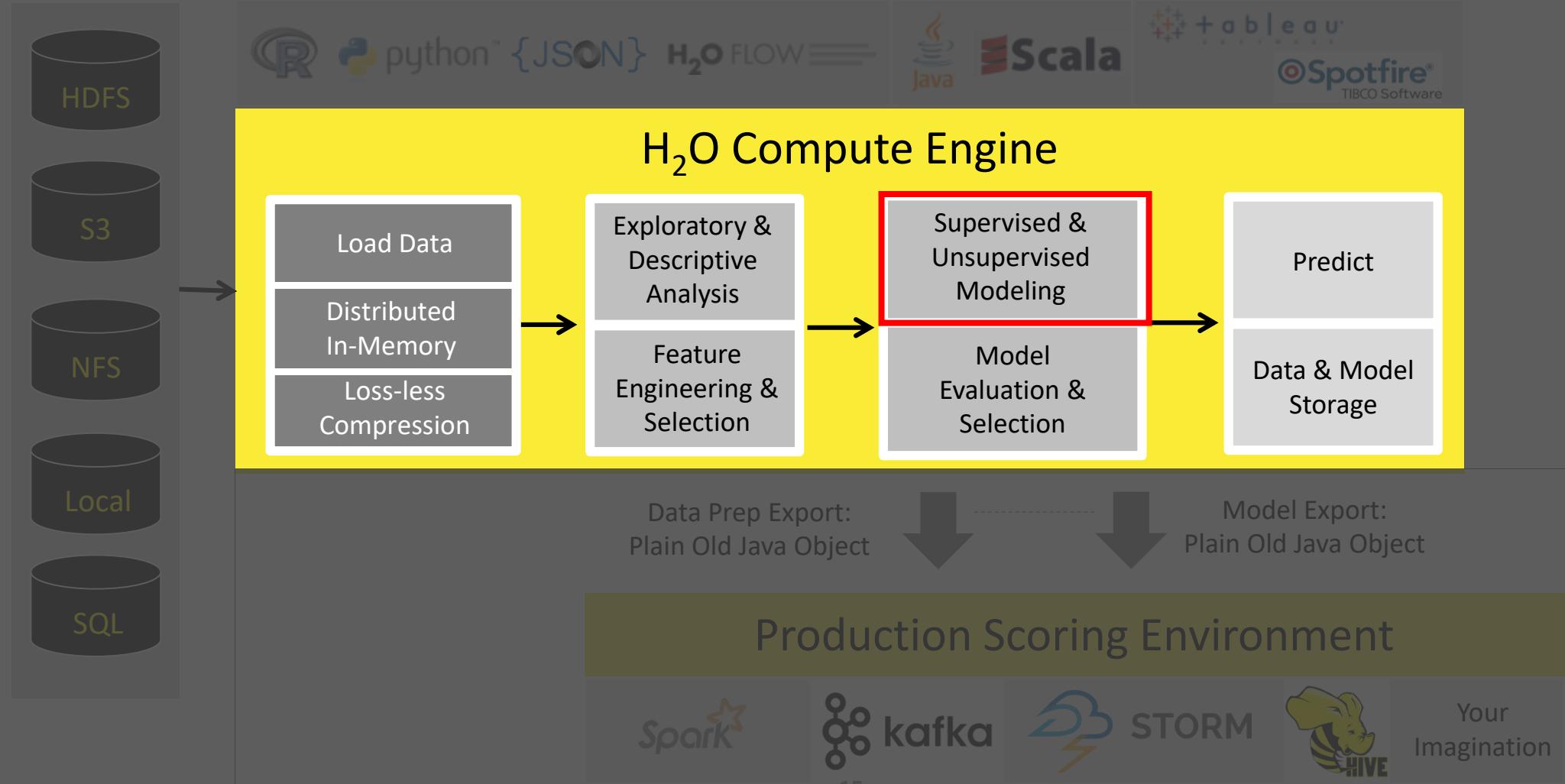
High Level Architecture

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

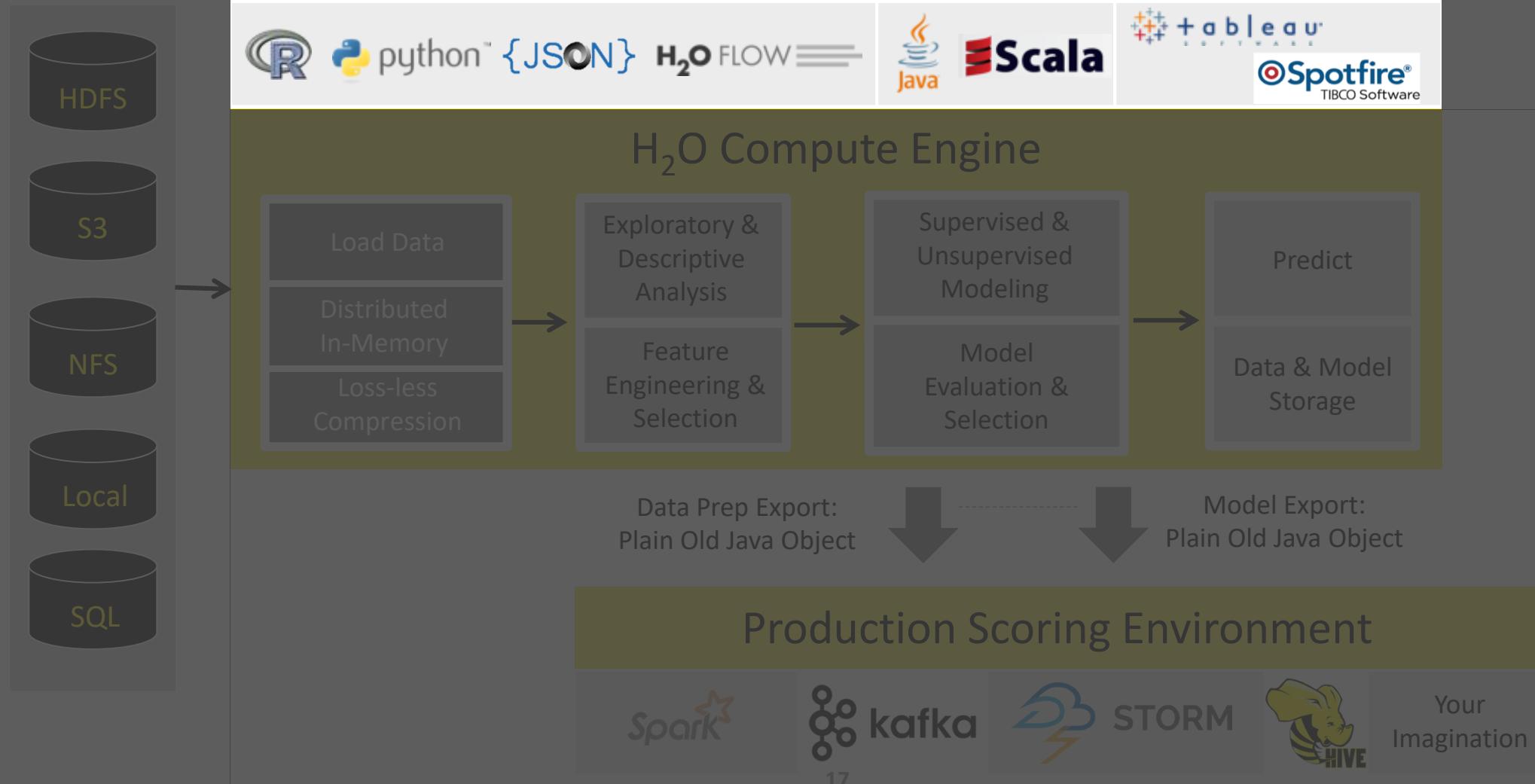
Dimensionality Reduction

- **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

High Level Architecture



H₂O + R

```
# Start and connect to a local H2O cluster  
library(h2o)  
h2o.init(ntreads = -1)
```

Package ‘h2o’ from CRAN
or H₂O’s website

```
# Import data from a R data frame  
data(iris)  
d_iris <- as.h2o(iris)
```

Start a local H₂O (Java
Virtual Machine) cluster

```
# Define Targets and Features  
target <- "Species"  
features <- setdiff(colnames(d_iris), c("Species"))
```

Simple ‘iris’ example

H₂O + 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)
```



Flow ▾ Cell ▾ Data ▾

Model ▾ Score ▾ Admin ▾ Help ▾

Iris Demo



CS

Expression...

- Aggregator...
- Deep Learning...
- Distributed Random Forest...
- Gradient Boosting Machine... 🕒
- Generalized Linear Modeling...
- Generalized Low Rank Modeling...
- K-means...
- Naive Bayes...
- Principal Components Analysis...

- List All Models
- List Grid Search Results
- Import Model...
- Export Model...

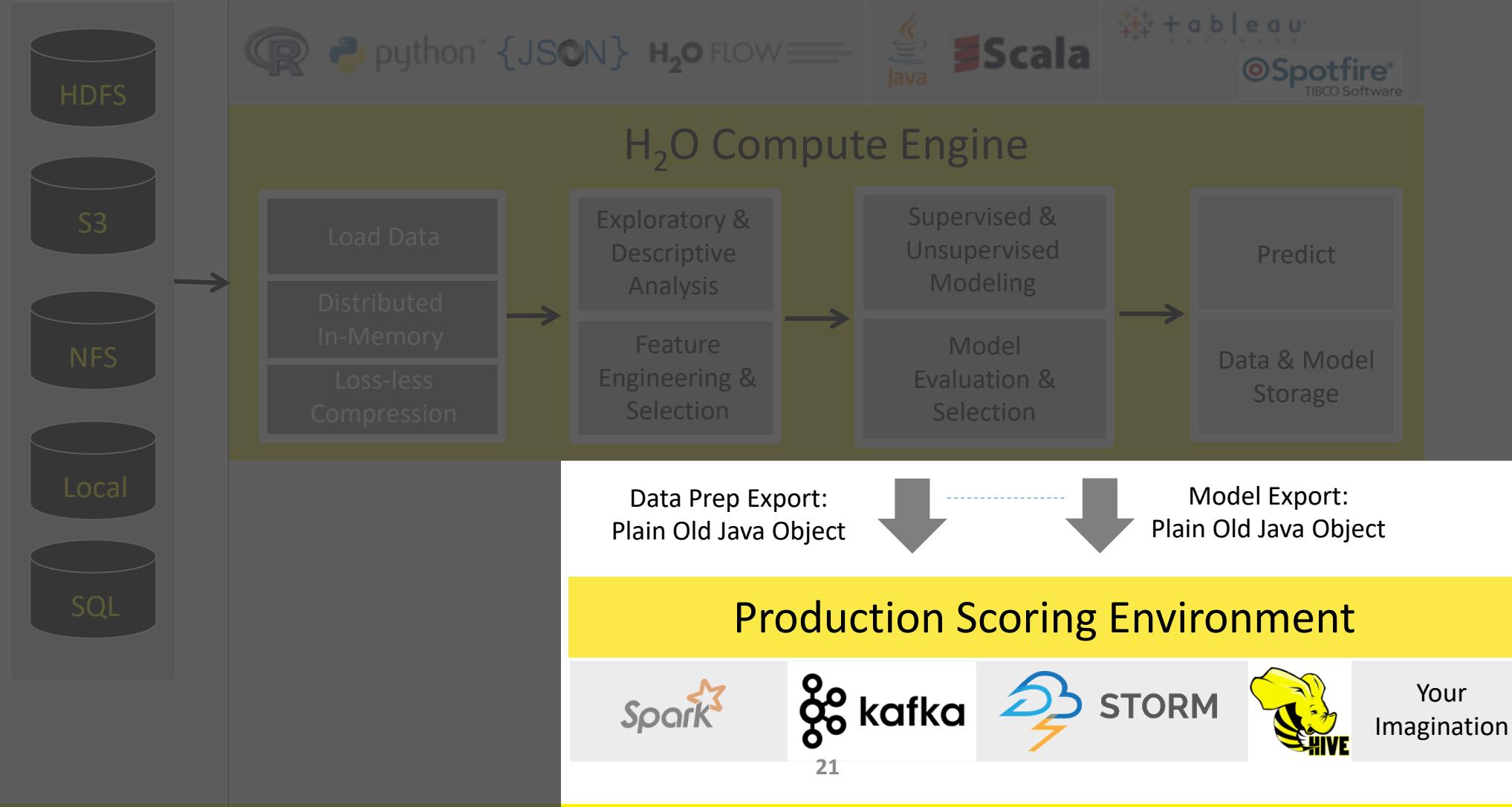
H₂O Flow (Web) Interface



Connections: 0 H₂O

High Level Architecture

Export Standalone Models
for Production



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 Training Materials

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

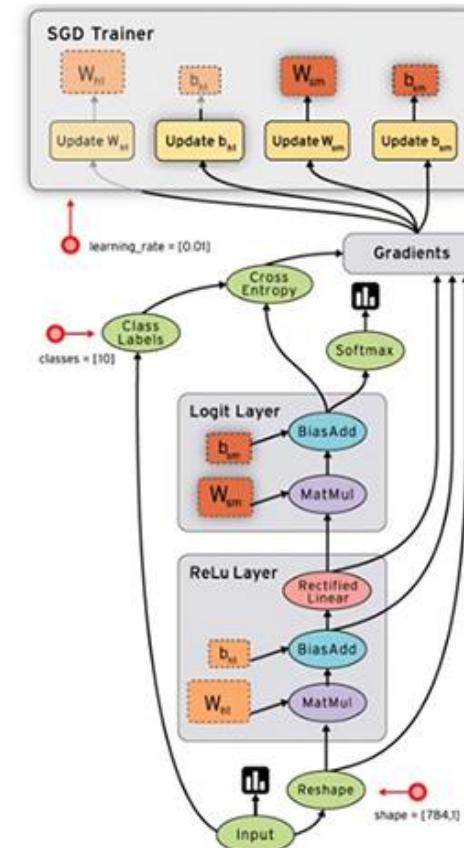


Open-Source Deep Learning Tools

TensorFlow, mxnet, Caffe and H₂O Deep Learning

TensorFlow

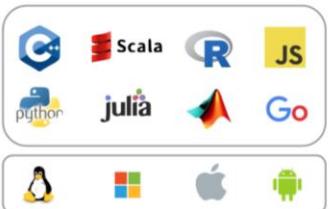
- Open source machine learning framework by Google
- Python / C++ API
- TensorBoard
 - Data Flow Graph Visualization
- Multi CPU / GPU
 - v0.8+ distributed machines support
- Multi devices support
 - desktop, server and Android devices
- Image, audio and NLP applications
- **HUGE** Community
- Support for Spark, Windows ...



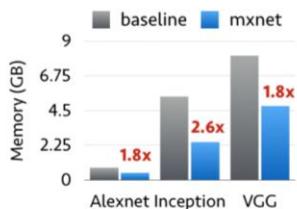
dmlc mxnet for Deep Learning

build passing docs latest license Apache 2.0

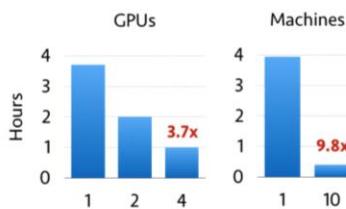
Portable



Efficient



Scalable



MXNet is a deep learning framework designed for both *efficiency* and *flexibility*. It allows you to *mix* the *flavours* of symbolic programming and imperative programming to *maximize* efficiency and productivity. In its core, a dynamic dependency scheduler that automatically parallelizes both symbolic and imperative operations on the fly. A graph optimization layer on top of that makes symbolic execution fast and memory efficient. The library is portable and lightweight, and it scales to multiple GPUs and multiple machines.

MXNet is also more than a deep learning project. It is also a collection of *blue prints and guidelines* for building deep learning system, and interesting insights of DL systems for hackers.

MXNet now chosen by Amazon as Deep Learning Framework

By Geneva Clark | 2016-11-24

Share this magazine



Amazon has announced that it has chosen MXNet as its deep learning framework of choice for its web services(AWS). Amazon extensively uses machine learning in areas like fraud detection, abusive review detection, and book classification. Amazon also uses it in application areas such as text and speech recognition, autonomous drones etc...

<https://github.com/dmlc/mxnet>

<https://www.zeolearn.com/magazine/amazon-to-use-mxnet-as-deep-learning-framework>

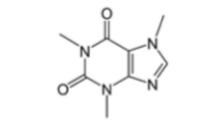
Caffe

- Convolution Architecture For Feature Extraction (CAFFE)
- Pure C++ / CUDA architecture for deep learning
- Command line, Python and MATLAB interface
- Model Zoo
 - Open collection of models

DIY Deep Learning for Vision: a Hands-On Tutorial with Caffe



| | Maximally accurate | Maximally specific |
|----------|--------------------|--------------------|
| espresso | 2.23192 | |
| coffee | 2.19914 | |
| beverage | 1.93214 | |
| liquid | 1.89367 | |
| fluid | 1.85519 | |



caffe.berkeleyvision.org



github.com/BVLC/caffe



Evan Shelhamer, Jeff Donahue, Jon Long,
Yangqing Jia, and Ross Girshick

Look for further
details in the
outline notes



H₂O Deep Learning

CIFAR-10 Competition
Winners: Interviews with Dr.
Ben Graham, Phil Culliton, &
Zygmunt Zajac
Triskelion | 01.02.2015

[READ MORE](#)

Kaggle challenge
2nd place winner
Colin Priest

[READ MORE](#)

for creating this corpus. . .
do not contain Spanish sent-
is a widespread major langu-
reason was to create a corp
tasks. These tasks are com

Completed • Knowledge • 161 teams

Denoising Dirty Documents

Mon 1 Jun 2015 – Mon 5 Oct 2015 (3 months ago)

“For my final competition submission I used an ensemble of models, including 3 deep learning models built with R and h2o.”

H₂O.ai



Both TensorFlow and H₂O are widely used

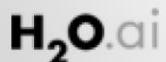
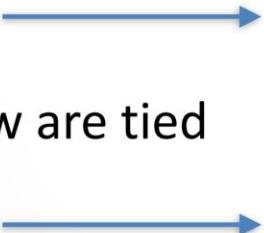
The usage of Hadoop/Big Data tools grew to 39%, up from 29% in 2015 (and 17% in 2014), driven by Apache Spark, MLlib (Spark Machine Learning Library) and H2O.

See also

- KDnuggets interview with Spark Creator Matei Zaharia
- KDnuggets interview with Arno Candel, H2O.ai on How to Quick Start Deep Learning with H2O

<http://www.kdnuggets.com>

H2O and TensorFlow are tied



TensorFlow, **MXNet**, **Caffe** and **H₂O DL**
democratize the power of deep learning.

H₂O platform democratizes artificial
intelligence & big data science.

There are other open source deep learning libraries like Theano and Torch too.
Let's have a party, this will be fun!

Deep Water

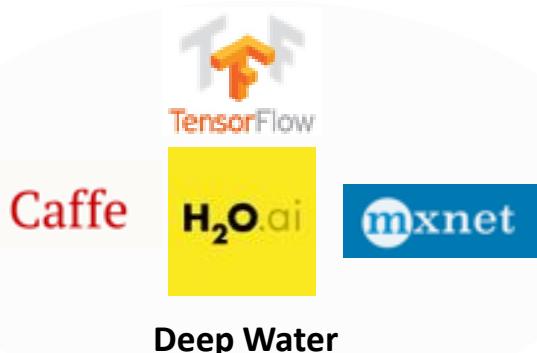
H₂O.ai Caffe  mxnet  TensorFlow

Deep Water

Next-Gen Distributed Deep Learning with H₂O

One Interface - GPU Enabled - Significant Performance Gains

Inherits All H₂O Properties in Scalability, Ease of Use and Deployment



H₂O integrates with existing **GPU** backends
for **significant performance gains**



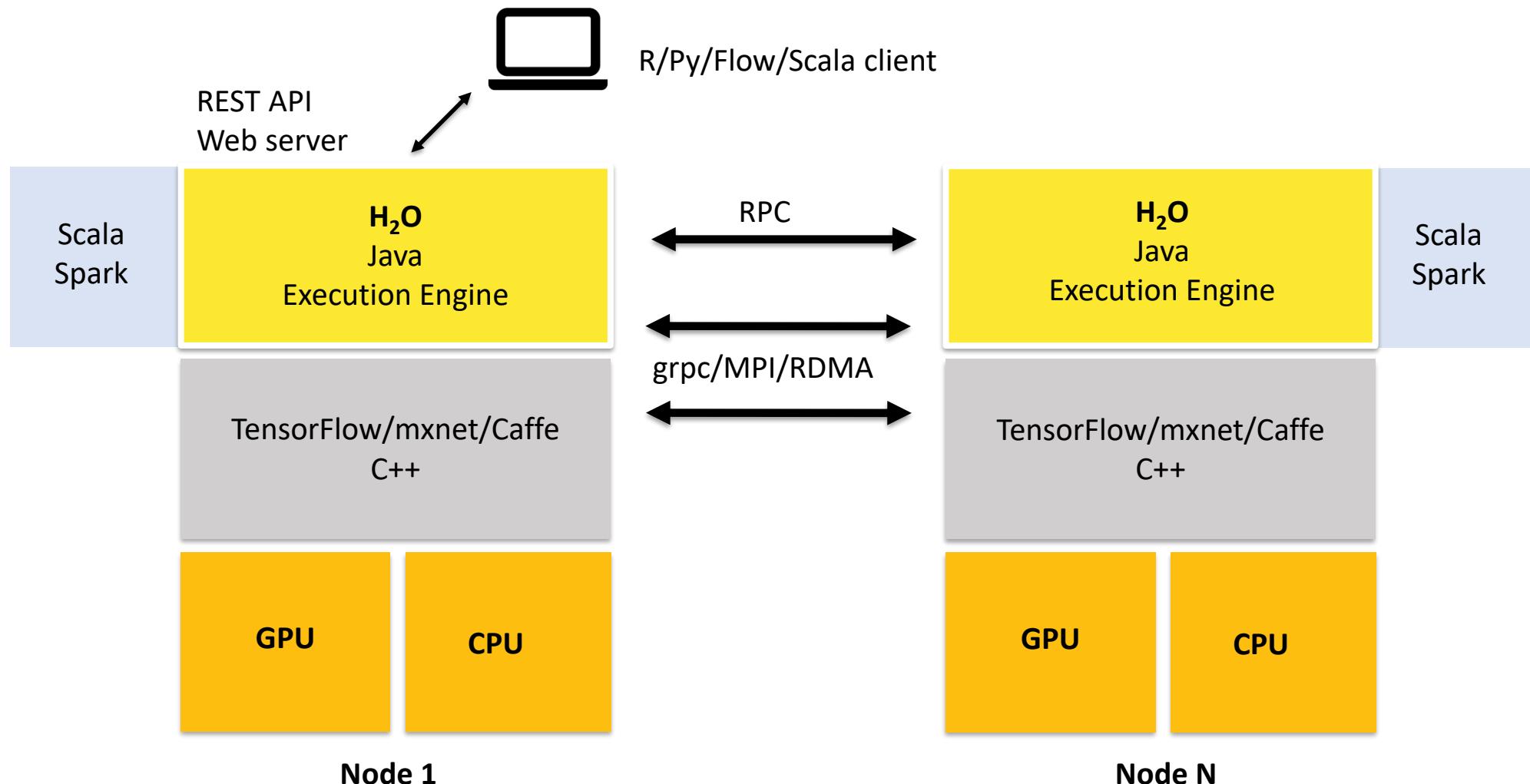
Convolutional Neural Networks enabling
Image, video, speech recognition



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

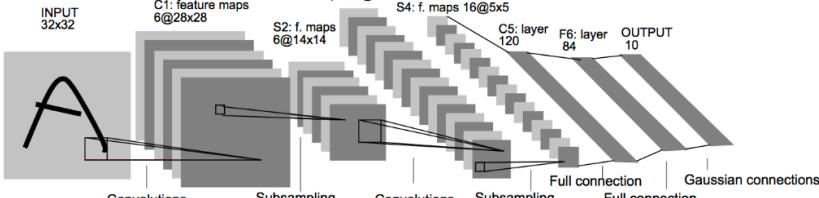
Hybrid Neural Network Architectures
enabling **speech to text translation, image captioning, scene parsing** and more

Deep Water Architecture



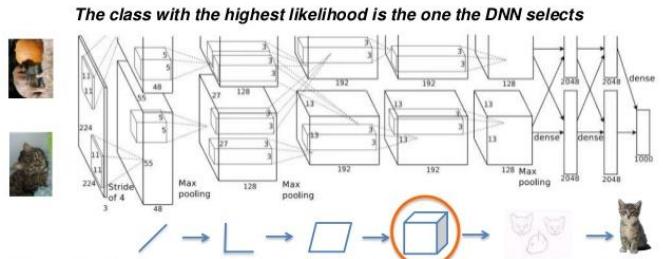
Available Networks in Deep Water

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



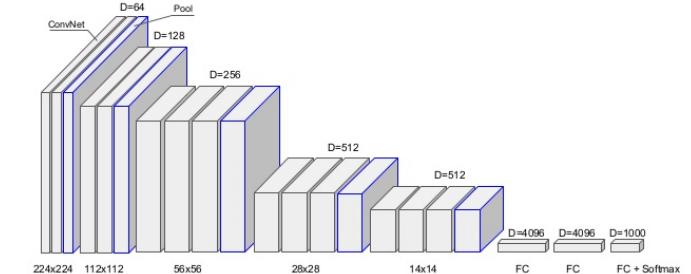
CNN called LeNet by Yann LeCun (1998)

AlexNet (Krizhevsky et al. 2012)

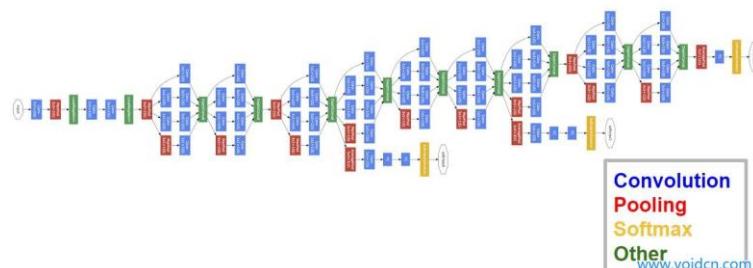


When AlexNet is processing an image, this is what is happening at each layer.

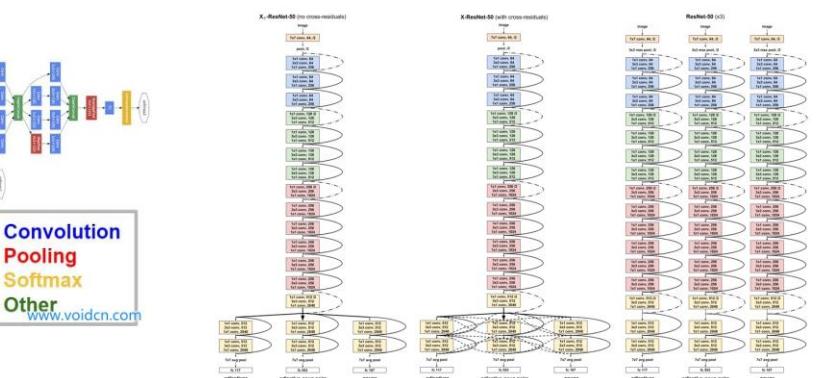
Classical CNN topology - VGGNet (2013)



GoogLeNet



ResNet



Deep Water H2O and TensorFlow Demo



All None

Only show columns with more than % missing values.

epochs

How many times the dataset should be iterated (streamed), can be fractional.

ignore_const_cols

Ignore constant columns.

network

Network architecture.

activation

Activation function. Only used if no user-defined network architecture file is provided, and only for problem_type=dataset.

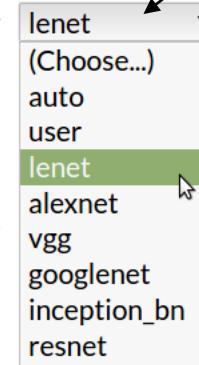
hidden

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).

Example: Deep Water + H₂O Flow Choosing different network structures



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

Column with cross-validation fold index assignment per observation.

offset_column

Offset column. This will be added to the combination of columns before applying the link function.



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). | <input type="checkbox"/> |
| score_validation_samples | 0 | Number of validation set samples for scoring (0 for all). | <input type="checkbox"/> |
| score_duty_cycle | 1 | Maximum duty cycle fraction for scoring (lower: more training, higher: more scoring). | <input type="checkbox"/> |
| 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) | <input type="checkbox"/> |
| stopping_metric | AUTO | Metric to use for early stopping (AUTO: logloss for classification, deviance for regression) | <input type="checkbox"/> |
| stopping_tolerance | 0 | Relative tolerance for metric-based stopping criterion (stop if relative improvement is not at least this much) | <input type="checkbox"/> |
| max_runtime_secs | 0 | Maximum allowed runtime in seconds for model training. Use 0 to disable. | <input type="checkbox"/> |
| backend | tensorflow ▾ | Deep Learning Backend. | <input type="checkbox"/> |
| image_shape | 28,28 | Width and height of image. | <input type="checkbox"/> |
| channels | 3 | Number of (color) channels. | <input type="checkbox"/> |
| network_definition_file | | Path of file containing network definition (graph, architecture). | <input type="checkbox"/> |
| network_parameters_file | | Path of file containing network (initial) parameters (weights, biases). | <input type="checkbox"/> |
| mean_image_file | | Path of file containing the mean image data for data normalization. | <input type="checkbox"/> |
| export_native_parameters_prefix | | Path (prefix) where to export the native model parameters after every iteration. | <input type="checkbox"/> |
| input_dropout_ratio | 0 | Input layer dropout ratio (can improve generalization, try 0.1 or 0.2). | <input type="checkbox"/> |
| hidden_dropout_ratios | | Hidden layer dropout ratios (can improve generalization), specify one value per hidden layer, defaults to 0.5. | <input type="checkbox"/> |

Unified Interface (Deep Water + R)

```
model <- h2o.deepwater(x=path, y=response,  
                        training_frame=df, epochs=50,  
                        learning_rate=1e-3, network = "lenet")  
model
```

Choosing different network structures

Unified Interface (Deep Water + Python)

Choosing different network structures

```
: model = H2ODeepWaterEstimator(epochs      = 500,  
                               network       = "lenet",  
                               image_shape  = [28,28],  ## provide image size  
                               channels     = 3,  
                               backend       = "tensorflow",  
                               model_id     = "deepwater_tf_simple")  
  
model.train(x = [0], # file path e.g. xxx/xxx/xxx.jpg  
            y = 1, # label cat/dog/mouse  
            training_frame = frame)  
  
model.show()
```

Change backend to
“mxnet”, “caffe” or “auto”

```
deepwater Model Build progress: |██████████| 100%  
Model Details  
=====
```

H2ODeepWaterEstimator : Deep Water
Model Key: deepwater_tf_simple

Easy Stacking with other H₂O Models

Model Stacking

Now we have three different models, we are ready to carry out model stacking.

```
In [47]: # Create a list to include all the models for stacking  
models <- list(model_dw, model_gbm, model_drf)
```

```
In [48]: # Define a metalearner (one of the H2O supervised machine learning algorithms)  
metalearner <- "h2o.glm.wrapper"
```

```
In [49]: # Use h2o.stack() to carry out metalearning  
stack <- h2o.stack(models = models,  
                    response_frame = h_train$medv,  
                    metalearner = metalearner)
```

```
[1] "Metalearning"
```

```
In [50]: # Finally, we evaluate the predictive performance on the ensemble as well as individual models.  
h2o.ensemble_performance(stack, newdata = h_test)
```

```
Base learner performance, sorted by specified metric:  
learner      MSE  
1 h2o_deepwater 8.377644  
2      h2o_gbm 8.106541  
3      h2o_drf 7.443517
```

```
H2O Ensemble Performance on <newdata>:  
-----
```

```
Family: gaussian
```

```
Ensemble performance (MSE): 5.80436983051916
```

Ensemble of Deep Water, Gradient Boosting
Machine & Random Forest models

H₂O, Sparkling Water, Steam, & Deep Water Documentation

[Getting Started](#)[Data Science Algorithms](#)[Languages](#)[Tutorials, Examples, & Presentations](#)[For Developers](#)[For the Enterprise](#)

docs.h2o.ai

Getting Started



H₂O

[What is H₂O?](#)
[H₂O User Guide](#)
[H₂O Book \(O'Reilly\)](#)
[Recent Changes](#)
[Open Source License \(Apache V2\)](#)

[Quick Start Video - Flow Web UI](#)
[Quick Start Video - R](#)
[Quick Start Video - Python](#)

[Download H₂O](#)

Sparkling Water

[What is Sparkling Water?](#)
[Sparkling Water Booklet](#)
[PySparkling Readme 2.0 | 1.6](#)
[RSparkling Readme](#)
[Open Source License \(Apache V2\)](#)

[Quick Start Video - Scala](#)
[Quick Start Video - Python](#)

[Download Sparkling Water](#)

Steam

[What is Steam?](#)
[Steam User Guide](#)
[Recent Changes](#)
[Open Source License \(AGPL\)](#)

[Download Steam](#)

Deep Water (preview)

[Deep Water Readme](#)
[Deep Water AMI Guide](#)
[Open Source License \(Apache V2\)](#)

[Launch Deep Water AMI
\(choose g2.2xlarge\)](#)

Q & A

[FAQ](#)
[Community Forum](#)
[h2ostream Google Group](#)
[Issue Tracking \(JIRA\)](#)
[Gitter](#)
[Stack Overflow](#)
[Cross Validated](#)

For Supported Enterprise Customers

[Enterprise Support Web | Email](#)

| | | | |
|---|--|-----|-----------------------------------|
|  mstensmo | changing the name of deeplearning_credit_card_default_risk_prediction... | ... | Latest commit 5568350 11 days ago |
| .. | | | |
|  images | Add cat/dog/mouse lenet example. | | 3 months ago |
|  README.md | Update README.md | | 2 months ago |
|  deeplearning_anomaly_detection.ipynb | Update notebooks, introduce local paths to ~/h2o-3/ | | 3 months ago |
|  deeplearning_benchmark_mnist.ipynb | Update lenet test to remove all. Update MNIST benchmark with comments. | | 3 months ago |
|  deeplearning_cat_dog_mouse_inception.ipynb | Add credit card default risk model, update other notebooks. | | |
|  deeplearning_cat_dog_mouse_lenet.ipynb | Add credit card default risk model, update other notebooks. | | |
|  deeplearning_cat_dog_mouse_lenet.ipynb | Add back model.plot() and scoring history. | | |
|  deeplearning_cifar10_vgg.ipynb | Rename notebooks. | | |
|  deeplearning_credit_card_default_risk.ipynb | changing the name of deeplearning_credit_card_default_risk_prediction... | | |
|  deeplearning_ensemble_boston_housing.ipynb | Ensemble demo using GBM, DRF and Deep Water (#676) | | 17 days ago |
|  deeplearning_grid_iris.ipynb | Add two new notebooks: Lenet for R and iris grid for python | | 3 months ago |
|  deeplearning_grid_iris_R.ipynb | Update R py notebook. | | 3 months ago |
|  deeplearning_image_reconstruction.ipynb | Update notebooks, introduce local paths to ~/h2o-3/ | | 3 months ago |
|  deeplearning_mnist_convnet.ipynb | Update notebooks, introduce local paths to ~/h2o-3/ | | 3 months ago |
|  deeplearning_mnist_introduction.ipynb | Add missing file. | | 3 months ago |
|  deeplearning_tensorflow_cat_dog.ipynb | Add tensorflow example (#529) | | 2 months ago |
|  deeplearning_tensorflow_mnist.ipynb | Added MNIST example for TensorFlow | | a month ago |

<https://github.com/h2oai/h2o-3/tree/master/examples/deeplearning/notebooks>

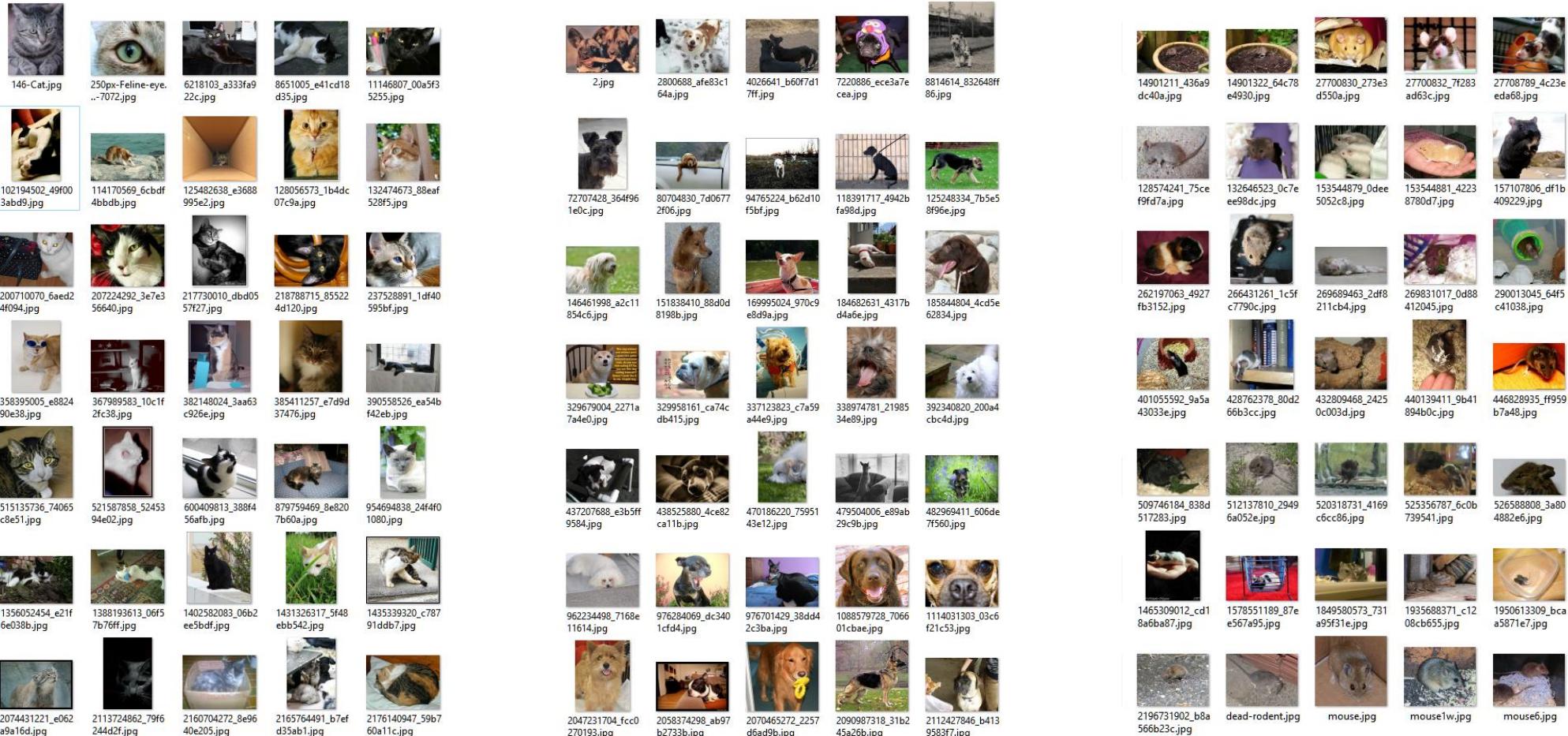
Deep Water Example notebooks

Deep Water Cat/Dog/Mouse Demo

Deep Water H₂O + MXNet Demo

- H₂O + MXNet
 - Dataset – Cat/Dog/Mouse
 - MXNet as GPU backend
 - Train LeNet (CNN) models
 - R Demo (Jupyter Notebook)
 - [Link](#)
- Code and Data
 - github.com/h2oai/deepwater

Data – Cat/Dog/Mouse Images



Data – CSV

| | A | B |
|----|---|-----|
| 1 | bigdata/laptop/deepwater/imagenet/cat/102194502_49f003abd9.jpg | cat |
| 2 | bigdata/laptop/deepwater/imagenet/cat/11146807_00a5f35255.jpg | cat |
| 3 | bigdata/laptop/deepwater/imagenet/cat/1140846215_70e326f868.jpg | cat |
| 4 | bigdata/laptop/deepwater/imagenet/cat/114170569_6cbdf4bbdb.jpg | cat |
| 5 | bigdata/laptop/deepwater/imagenet/cat/1217664848_de4c7fc296.jpg | cat |
| 6 | bigdata/laptop/deepwater/imagenet/cat/1241603780_5e8c8f1ced.jpg | cat |
| 7 | bigdata/laptop/deepwater/imagenet/cat/1241612072_27ececbdef.jpg | cat |
| 8 | bigdata/laptop/deepwater/imagenet/cat/1241613138_ef1d82973f.jpg | cat |
| 9 | bigdata/laptop/deepwater/imagenet/cat/1244562192_35becd66bd.jpg | cat |
| 10 | bigdata/laptop/deepwater/imagenet/cat/125482638_e3688995e2.jpg | cat |
| 11 | bigdata/laptop/deepwater/imagenet/cat/128056573_1b4dc07c9a.jpg | cat |
| 12 | bigdata/laptop/deepwater/imagenet/cat/12945197_75e607e355.jpg | cat |
| 13 | bigdata/laptop/deepwater/imagenet/cat/132474673_88eaf528f5.jpg | cat |
| 14 | bigdata/laptop/deepwater/imagenet/cat/1350530984_ecf3039cf0.jpg | cat |
| 15 | bigdata/laptop/deepwater/imagenet/cat/1351606235_c9fbef634.jpg | cat |
| 16 | bigdata/laptop/deepwater/imagenet/cat/1356052454_e21f6e038b.jpg | cat |
| 17 | bigdata/laptop/deepwater/imagenet/cat/1388193613_06f57b76ff.jpg | cat |

Deep Water – Basic Usage

Start and Connect to H2O Cluster

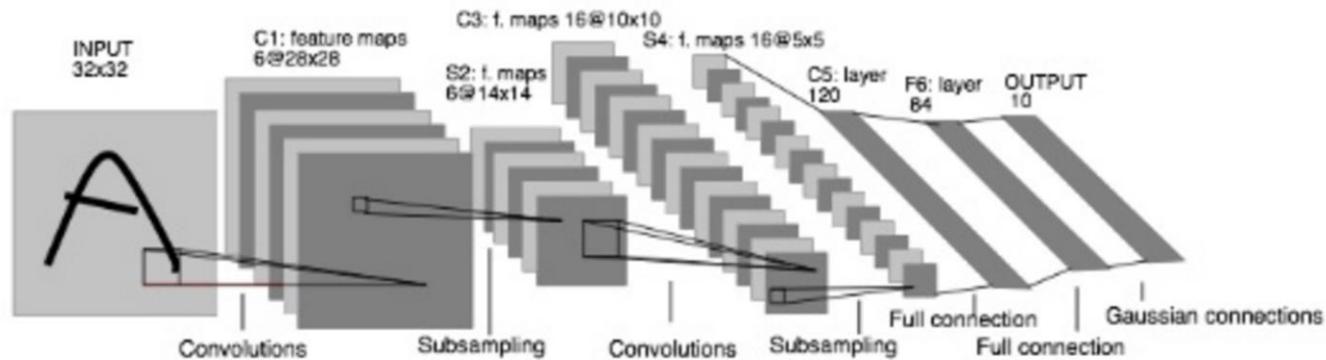
```
library(h2o)
h2o.init(nthreads=-1)
```

Import CSV

```
df <- h2o.importFile("/home/ubuntu/h2o-3/bigdata/laptop/deepwater/imagenet/cat_dog_mouse.csv")
print(head(df))
path = 1 ## must be the first column
response = 2
```

```
|=====| 100%
          C1  C2
1  bigdata/laptop/deepwater/imagenet/cat/102194502_49f003abd9.jpg  cat
2  bigdata/laptop/deepwater/imagenet/cat/11146807_00a5f35255.jpg  cat
3  bigdata/laptop/deepwater/imagenet/cat/1140846215_70e326f868.jpg  cat
4  bigdata/laptop/deepwater/imagenet/cat/114170569_6cbdf4bbdb.jpg  cat
5  bigdata/laptop/deepwater/imagenet/cat/1217664848_de4c7fc296.jpg  cat
6  bigdata/laptop/deepwater/imagenet/cat/1241603780_5e8c8f1ced.jpg  cat
```

Train a CNN (LeNet) Model on GPU



LeNet: a layered model composed of convolution and subsampling operations followed by a holistic representation and ultimately a classifier for handwritten digits. [Yann LeCun; LeNet]

We'll use a GPU to train such a LeNet model in seconds

To build a LeNet image classification model in H2O, simply specify `network = "lenet"`:

```
model <- h2o.deepwater(x=path, y=response,  
                        training_frame=df, epochs=50,  
                        learning_rate=1e-3, network = "lenet")
```

Train a CNN (LeNet) Model on GPU

Model

Model Details:

=====

```
H2OMultinomialModel: deepwater
Model ID: DeepWater_model_R_1477378862430_2
Status of Deep Learning Model: lenet, 1.6 MB, predicting C2, 3-class classif
s, mini-batch size 32
    input_neurons      rate momentum
1           2352  0.000986  0.990000
```

H2OMultinomialMetrics: deepwater

** Reported on training data. **

** Metrics reported on full training frame **

Training Set Metrics:

=====

Extract training frame with `h2o.getFrame("cat_dog_mouse.hex_sid_95f8_1")`

MSE: (Extract with `h2o.mse`) 0.131072

RMSE: (Extract with `h2o.rmse`) 0.3620386

Logloss: (Extract with `h2o.logloss`) 0.4176429

Mean Per-Class Error: 0.1165104

Confusion Matrix: Extract with `h2o.confusionMatrix(<model>,train = TRUE)`

=====

Confusion Matrix: vertical: actual; across: predicted

| | cat | dog | mouse | Error | Rate |
|--------|-----|-----|-------|--------|------------|
| cat | 75 | 4 | 11 | 0.1667 | = 15 / 90 |
| dog | 4 | 75 | 6 | 0.1176 | = 10 / 85 |
| mouse | 3 | 3 | 86 | 0.0652 | = 6 / 92 |
| Totals | 82 | 82 | 103 | 0.1161 | = 31 / 267 |

Deep Water – Custom Network

If you'd like to build your own LeNet network architecture, then this is easy as well. In this example script, we are using the 'mxnet' backend. Models can easily be imported/exported between H2O and MXNet since H2O uses MXNet's format for model definition.

```
In [5]: get_symbol <- function(num_classes = 1000) {  
  library(mxnet)  
  data <- mx.symbol.Variable('data')  
  # first conv  
  conv1 <- mx.symbol.Convolution(data = data, kernel = c(5, 5), num_filter = 20)  
  
  tanh1 <- mx.symbol.Activation(data = conv1, act_type = "tanh")  
  pool1 <- mx.symbol.Pooling(data = tanh1, pool_type = "max", kernel = c(2, 2), stride = c(2, 2))  
  
  # second conv  
  conv2 <- mx.symbol.Convolution(data = pool1, kernel = c(5, 5), num_filter = 50)  
  tanh2 <- mx.symbol.Activation(data = conv2, act_type = "tanh")  
  pool2 <- mx.symbol.Pooling(data = tanh2, pool_type = "max", kernel = c(2, 2), stride = c(2, 2))  
  # first fullc  
  flatten <- mx.symbol.Flatten(data = pool2)  
  fc1 <- mx.symbol.FullyConnected(data = flatten, num_hidden = 500)  
  tanh3 <- mx.symbol.Activation(data = fc1, act_type = "tanh")  
  # second fullc  
  fc2 <- mx.symbol.FullyConnected(data = tanh3, num_hidden = num_classes)  
  # loss  
  lenet <- mx.symbol.SoftmaxOutput(data = fc2, name = 'softmax')  
  return(lenet)  
}
```

Configure custom
network structure
(MXNet syntax)

```
In [7]: nclasses = h2o.nlevels(df[,response])  
network <- get_symbol(nclasses)  
cat(network$as.json(), file = "/tmp/symbol_lenet-R.json", sep = '')
```

Saving the custom network
structure as a file

Train a Custom Network

```
model = h2o.deepwater(x=path, y=response, training_frame = df,
                      epochs=500, ## early stopping is on by default and might trigger before
                      network_definition_file="/tmp/symbol_lenet-R.json", ## specify the model
                      image_shape=c(28,28), ## provide expected (or matching
g) image size
                      channels=3) ## 3 for color, 1 for monochrom
e
```

Model

Model Details:

=====

H2OMultinomialModel: deepwater

Model Key: DeepWater_model_R_1477378862430_3

Status of Deep Learning Model: user, 1.6 MB, predicting C2, 3-class classifiers, mini-batch size 32

 input_neurons rate momentum

 1 2352 0.004409 0.990000

H2OMultinomialMetrics: deepwater

 ** Reported on training data. **

 ** Metrics reported on full training frame **

Training Set Metrics:

=====

Extract training frame with `h2o.getFrame("cat_dog_mouse.hex_sid_95f8_1")`

MSE: (Extract with `h2o.mse`) 0.03078524

RMSE: (Extract with `h2o.rmse`) 0.1754572

Logloss: (Extract with `h2o.logloss`) 0.1154222

Mean Per-Class Error: 0.03366487

Confusion Matrix: Extract with `h2o.confusionMatrix(<model>,train = TRUE)`

=====

Confusion Matrix: vertical: actual; across: predicted

| | cat | dog | mouse | Error | Rate |
|--------|-----|-----|-------|--------|-----------|
| cat | 88 | 2 | 0 | 0.0222 | = 2 / 90 |
| dog | 2 | 82 | 1 | 0.0353 | = 3 / 85 |
| mouse | 1 | 3 | 88 | 0.0435 | = 4 / 92 |
| Totals | 91 | 87 | 89 | 0.0337 | = 9 / 267 |

Other News

Other H₂O Developments

- H₂O + xgboost [[Link](#)]
- Stacked Ensembles [[Link](#)]
- Automatic Machine Learning [[Link](#)]
- Time Series [[Link](#)]
- High Availability Mode in Sparkling Water [[Link](#)]
- Model Interpretation [[Link](#)]
- Previous Talk
 - https://github.com/h2oai/h2o-meetups/tree/master/2017_03_09_London_IoT_Meetup

Conclusions

Project “Deep Water”

- H₂O + TF + MXNet + Caffe
 - a powerful combination of widely used open source machine learning libraries.
- All Goodies from H₂O
 - inherits all H₂O properties in scalability, ease of use and deployment.
- Unified Interface
 - allows users to build, stack and deploy deep learning models from different libraries efficiently.

- 100% Open Source
 - the party will get bigger!



Thanks!

- Organizers & Sponsors
 - Barcelona R Users Group
 - H2O.ai
- Code, Slides & Documents
 - bit.ly/h2o_meetups
 - bit.ly/h2o_deepwater
 - docs.h2o.ai
- Contact
 - joe@h2o.ai
 - [@matlabulous](https://twitter.com/matlabulous)
 - github.com/woobe
- Please search/ask questions on **Stack Overflow**
 - Use the tag `h2o` (not H2 zero)