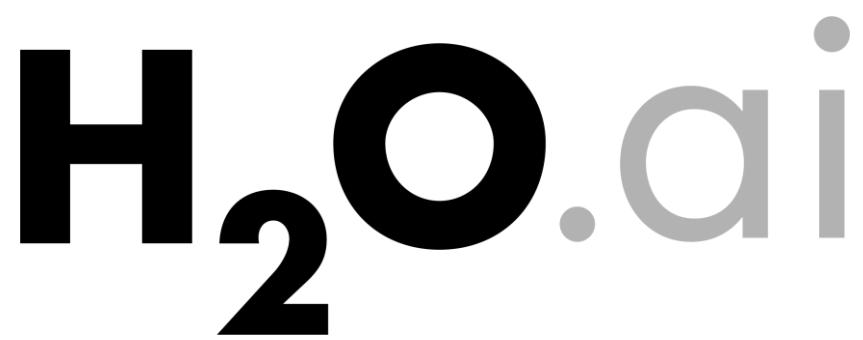


H_2O at Berlin R Meetup

Introduction to ML with H_2O and Deep Water



Jo-fai (Joe) Chow

Data Scientist

joe@h2o.ai

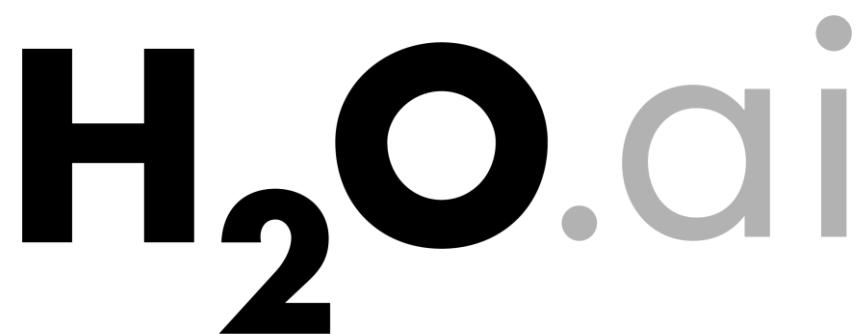
@matlabulous

Berlin R at Adjust.com

23rd May, 2017

H₂O at Berlin R Meetup

Introduction to ML with H₂O and Deep Water



Jo-fai (Joe) Chow

Data Scientist

joe@h2o.ai

@matlabulous

All slides, data and code examples

http://bit.ly/h2o_meetups

Agenda

- Introduction
 - Company
 - Why H₂O?
 - H₂O Machine Learning Platform
- Deep Water
 - Motivation / Benefits
 - GPU Deep Learning Demo
- Sparkling Water
 - Kuba's Talk
 - Sparkling Water = H₂O + Spark



About Me

- Civil (Water) Engineer
 - 2010 – 2015
 - Consultant (UK)
 - Utilities
 - Asset Management
 - Constrained Optimization
 - EngD (Industrial PhD) (UK)
 - Infrastructure Design Optimization
 - Machine Learning + Water Engineering
 - Discovered H₂O in 2014
 - Data Scientist
 - 2015 – 2016
 - Virgin Media (UK)
 - Domino Data Lab (Silicon Valley)
 - 2016 – Present
 - H₂O.ai (Silicon Valley)
 - How?
 - bit.ly/joe_kaggle_story

About Me – I ❤️ R

Search GitHub

Pull requests Issues Gist

Overview Repositories 48 Stars 418 Followers 152 Following 31

Popular repositories

Customize your pinned repositories

blenditbayes
Code used in my blog "Blend it like a Bayesian!"
R ★ 79 ⚡ 81

deepr
An R package to streamline the training, fine-tuning and predicting processes for deep learning based on 'darch' and 'deepnet'.
R ★ 43 ⚡ 16

rPlotter
Wrapper functions that make plotting in R a lot easier for beginners.
R ★ 32 ⚡ 4

rCrimemap
This is the next generation of CrimeMap!
R ★ 22 ⚡ 8

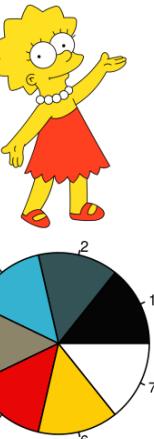
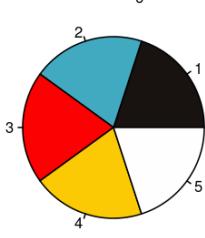
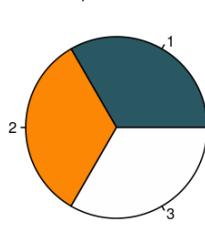
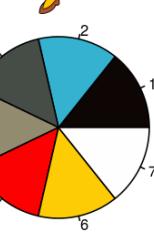
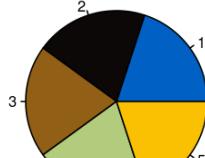
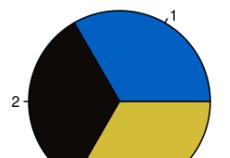
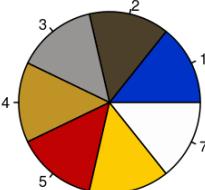
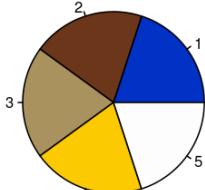
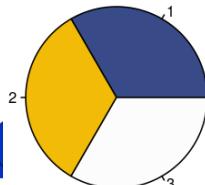
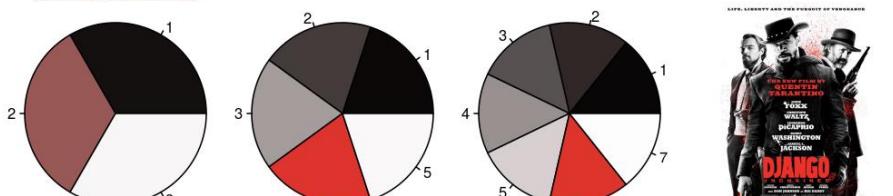
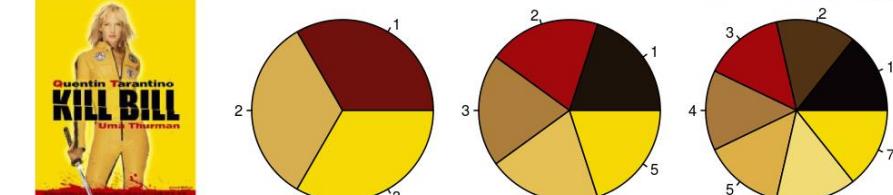
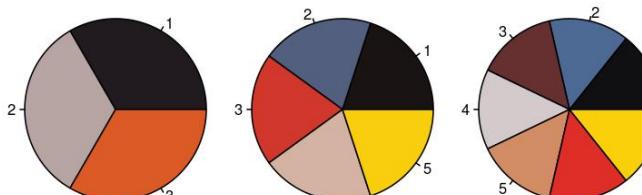
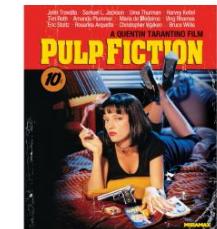
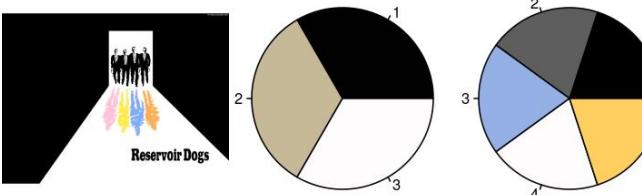
rugsmaps
This app is my submission to the visualization contest held by Revolution Analytics.
R ★ 19 ⚡ 18

rApps
Repository for my R (Shiny) web applications.
R ★ 16 ⚡ 38

396 contributions in the last year Contribution settings ▾

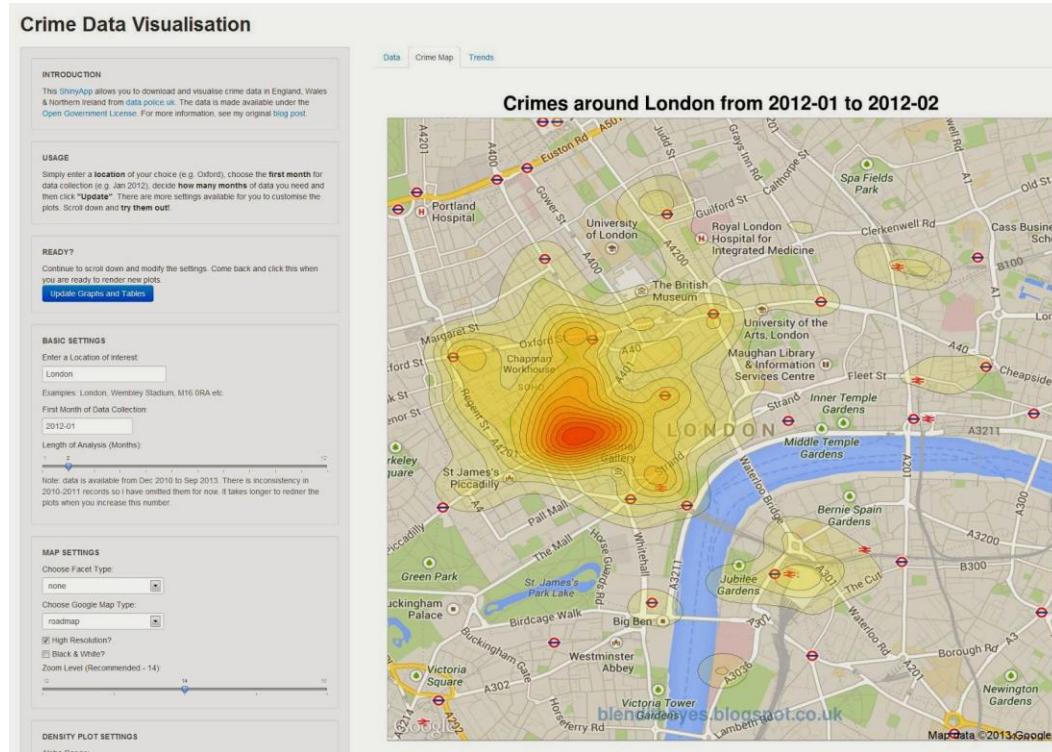
The screenshot shows a GitHub profile for the user 'jofai'. The profile picture is a circular map with a blue and white heatmap overlay. The user's name is 'Jo-fai Chow' and their handle is 'woobe'. Below the name, it says 'Civil Engineer turned Data Scientist'. The profile includes sections for 'H2O.ai' (United Kingdom, email, website), 'Organizations' (H2O.ai), and 'Contributions' (396 in the last year). The 'Popular repositories' section lists six repositories, all of which have an 'R' icon next to them, indicating they are R packages. These repositories are: 'blenditbayes', 'deepr', 'rPlotter', 'rCrimemap', 'rugsmaps', and 'rApps'. Each repository card shows its name, a brief description, the 'R' icon, the number of stars, and the number of contributions (⚡). The entire list of repositories is circled with a yellow oval.

About Me – I ❤ Colours

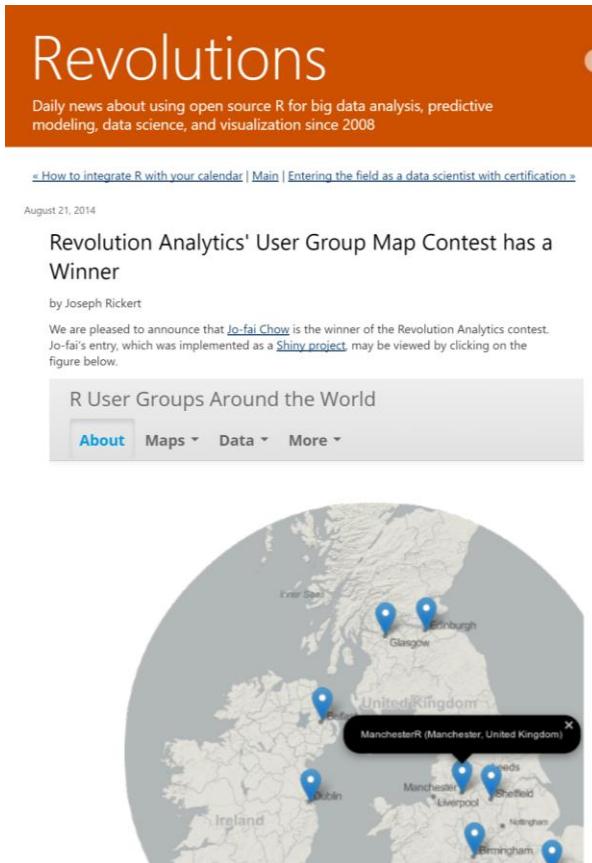


Developing R Packages for Fun
[rPlotter](#) (2014)

About Me – I ❤ DataViz



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



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

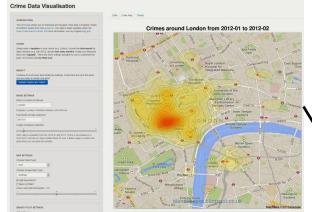


Jo-fai (Joe) Chow
@matlabulous

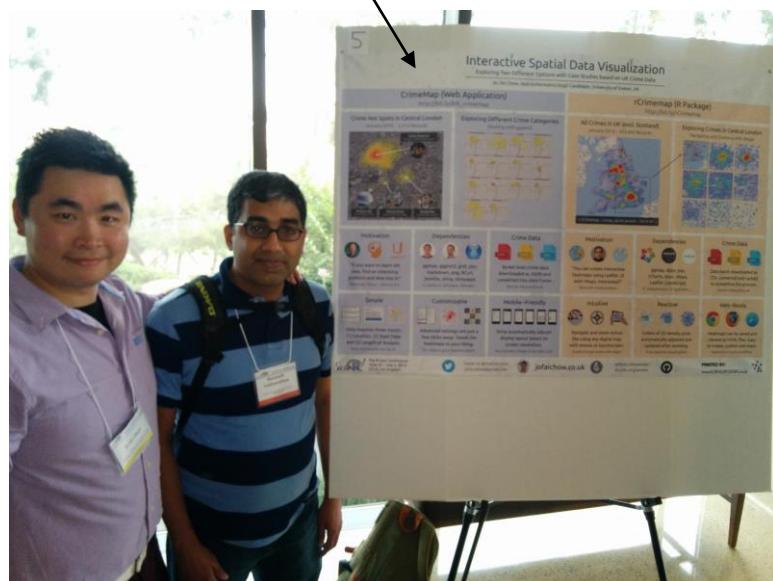
Thank you very much @RevolutionR
@revodavid @RevoJoe #iloveR
bit.ly/rugsmaps #Shiny #rMaps



useR! 2014



CrimeMap -> poster



H2O.ai @h2oai Following
#linus for #rstat #user2014 #JohnChambers



John Chambers mentioned H₂O



Jo-fai (Joe) Chow
@matlabilous

Replies to @h2oai

Hi @srishatish @ArnoCandel and every1
@hexadata thx 4 making and open-sourcing
the powerful #H2O shd hv tried it during (not
after) #user2014

LIKES
2

1:41 PM · 28 Jul 2014

H₂O.ai

About Me – I ❤️ Kaggle

The screenshot shows a blog post on the Domino Data Lab website. The post is titled "How to use R, H2O, and Domino for a Kaggle competition" and is a guest post by Jo-Fai Chow. It was published on 19 Sep 2014. The post includes social sharing icons for Facebook, Twitter, and Google+, and a link to the sample project on Domino. The content discusses using Domino, H2O, and R for a Kaggle competition, providing links to three tutorials. The introduction mentions it's a sequel to a previous post and describes the machine learning case study.

Domino Data Lab
At the intersection of data science and engineering.
Domino App Site | [Twitter](#) | [Email](#)

19 Sep 2014 • 0 Likes | 21 Tweets | 4 G+1

How to use R, H2O, and Domino for a Kaggle competition

Guest post by [Jo-Fai Chow](#)

The sample project ([code](#) and [data](#)) described below is [available on Domino](#).

If you're in a hurry, feel free to skip to:

- [Tutorial 1: Using Domino](#)
- [Tutorial 2: Using H2O to Predict Soil Properties](#)
- [Tutorial 3: Scaling up your analysis](#)

Introduction

This blog post is the sequel to [TTTAR1](#) a.k.a. [An Introduction to H2O Deep Learning](#). If the previous blog post was a brief intro, this post is a proper machine learning case study based on a recent [Kaggle competition](#): I am leveraging [R](#), [H2O](#) and [Domino](#) to compete (and do pretty well) in a real-world data mining contest.

R + H₂O + Domino for Kaggle
[Guest Blog Post for Domino & H₂O \(2014\)](#)

- The Long Story
 - bit.ly/joe_kaggle_story

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• Deep Water• Steam
Mission	Operationalize Data Science, and provide a platform for users to build beautiful data products
Team	<p>70 employees</p> <ul style="list-style-type: none">• Distributed Systems Engineers doing Machine Learning• World-class visualization designers
Headquarters	Mountain View, CA





Amy Vu-Tran



Amy Wang



Angela Bartz



Anmol Bal



Amao Candel



Aulrich Barthaz



Maral Mandjarijan



Mark Chan



Mark Landry



Mateusz Dymotyki



Matt Dowle



Megan Kuras



Arvindh Chauhan



Anni Wadhwa



Beth Payne



Brandon Murray



Carl Andrews



Das Narayanan



Michal Kurka



Michal Moloshko



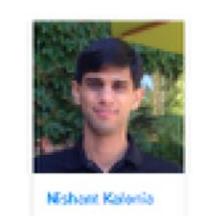
Navdeep Gill



Nidhi Mehta



Nikhil Shekhar



Nishant Kaleria



Daivid Chan



Dmitry Larko



Erin LaDell



Fonda Ingram



Ian Gomez



Jacqueline Scott



Pasha Satsenko



Patrick Hall



Patrick Rice



Prithvi Prabhu



Ravi Purushothama



Raymond Peck



Jakub Hava



Jeff Fohl



Jeff Cambena



Jo-Hai Chow



Jon Olszewski



Josephine Wang



Sebastian Vidrio



Srikanth Ambati



Terence Ward



Tam Kraljevic



Tomas Nykodym



Venkatesh Yedda



Justin Loyola



Karen Hayrapetyan



Kimberly O'Shea



Lauren DiPerna



Leland Wilkinson



Magnus Stenske



Vinod Iyengar



Wien Pham



Wendy Wong



Kuba (Prague)

Joe (London)

H₂O.ai Offers AI Open Source Platform Product Suite to Operationalize Data Science with Visual Intelligence

This Meetup

Framework For Data Interpretation and
Build Powerful Data Products

100% Open Source



In-Memory, Distributed
Machine Learning
Algorithms with Speed and
Accuracy

Deep Water

State-of-the-art
Deep Learning on GPUs with
TensorFlow, MXNet or Caffe
with the ease of use of H2O



H2O Integration with Spark.
Best Machine Learning on
Spark.

Steam

Operationalize and
Streamline Model Building,
Training and Deployment
Automatically and Elastically

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*



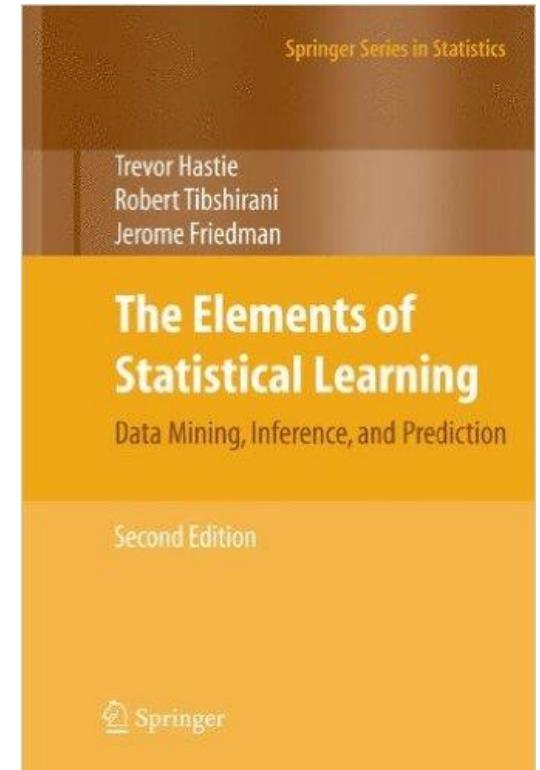
Dr. Robert Tibshirani

- 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*





wenphan
@wenphan

Following

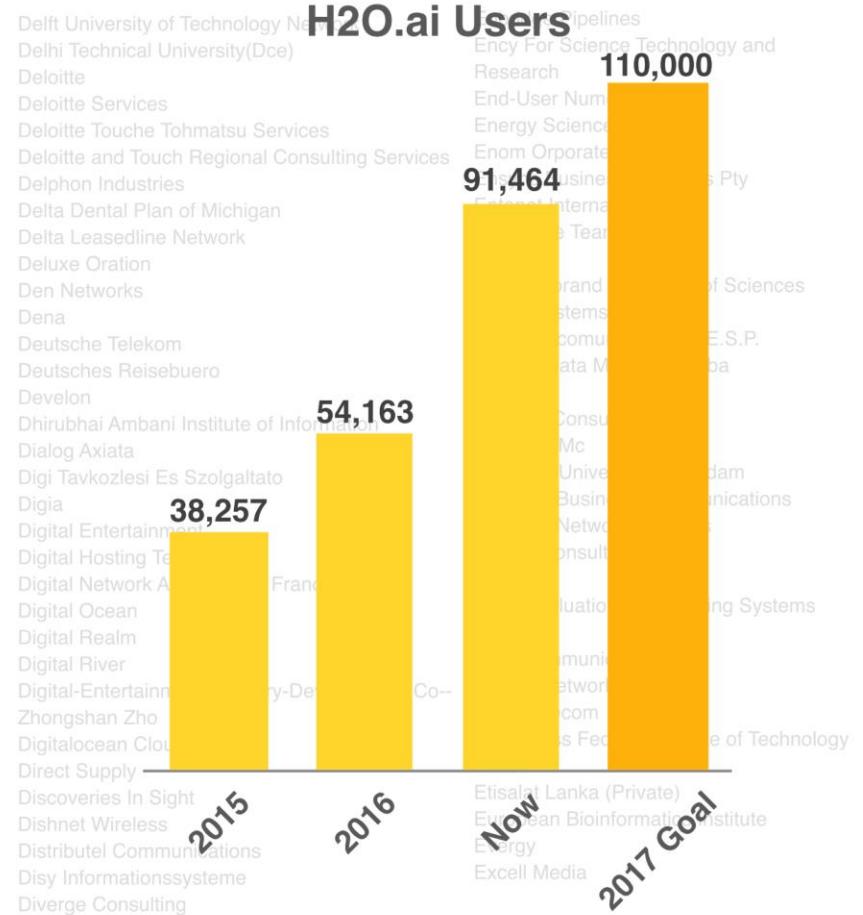
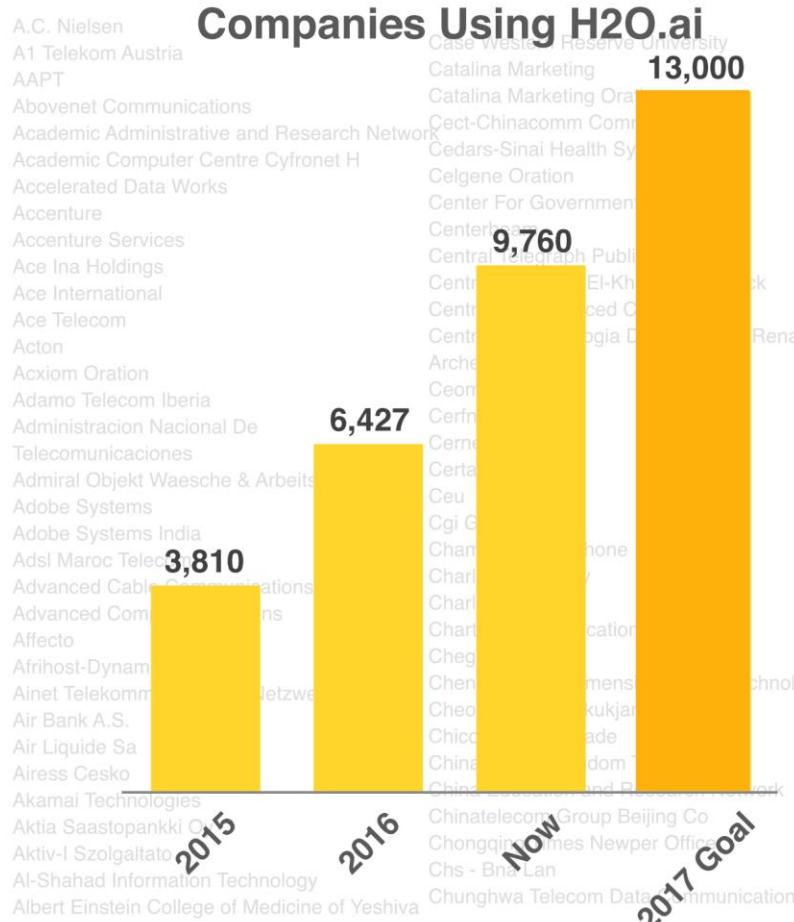


So much brain power in one place:
[@ArnoCandel](#) and Stanford profs. Boyd,
Tibs, and Hastie. Hacking algos at [@h2oai](#)
HQ



Arno (CTO)

H2O Community & Fortune 100 customers



Select Reference Customers:

"Overall customer satisfaction is very high." - Gartner

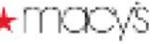
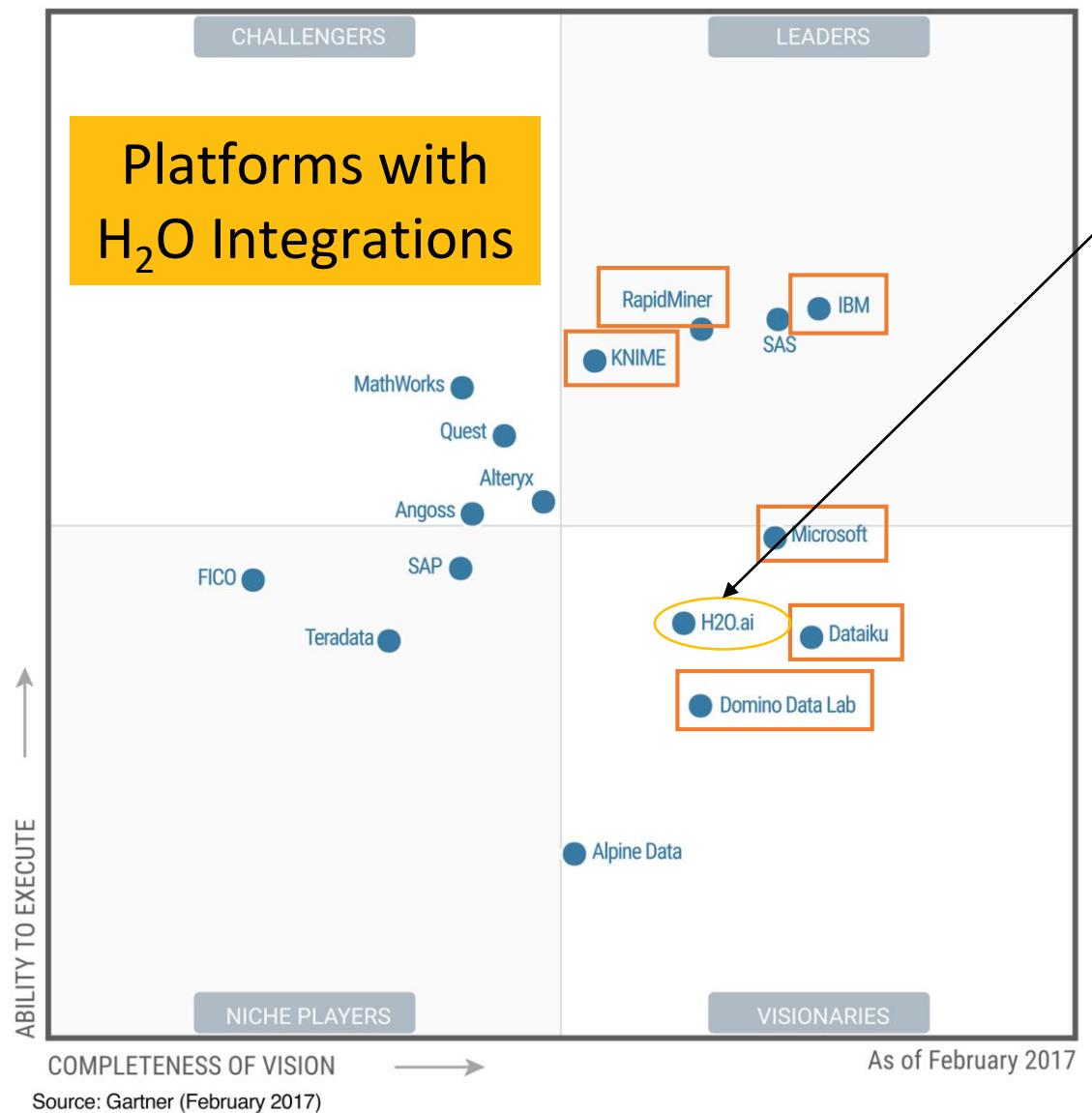


Figure 1. Magic Quadrant for Data Science Platforms



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.

Check out our website h2o.ai

H2O In Action

ADP



Various data leaders discuss the transformative impact of H2O AI for ADP.

Insurance



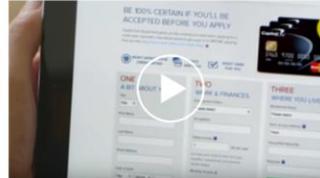
What data products mean and why H2O keeps this industry leader relevant.

Progressive



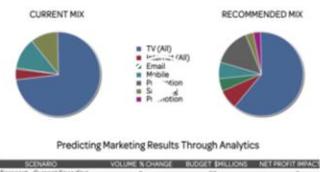
See how Progressive uses H2O predictive analytics for User-based Insurance (UBI).

Capital One



Capital One uses H2O machine learning for various use cases.

MarketShare



H2O predictive analytics helps boost the impact and results of digital marketing.

Kaiser



Kaiser uses H2O machine learning to save lives.

Zurich Insurance



Zurich turned to H2O as a strategic differentiator for commercial insurance.

Comcast



Comcast uses H2O to improve customer experience.

McKesson



McKesson discusses the adoption of artificial intelligence in healthcare.

Macy's



Macy's uses H2O for personalized site recommendations.

Transamerica



Transamerica turns to H2O to develop an insurance recommendation platform.

Paypal



Paypal turned to H2O Deep Learning for fraud detection and customer churn.

eBay



eBay chose H2O for open source machine learning.

Cisco



Cisco uses H2O to build a scalable model factory to improve sales and marketing.

Nielsen Catalina Solutions



H2O helps the country's largest TV behavior analytics company optimize ad performance.

#AroundTheWorldWithH2Oai

A year ago I couldn't even imagine myself attending #StrataHadoop thx @h2oai for the oppor... ift.tt/1UjtrG7



#AroundTheWorldWithH2Oai @h2oai #paris
bit.ly/h2o_meetups #museedulouvre #twitter
bit.ly/2g6m9tb



@h2oai's very first #meetup in #Warsaw.
Thx @DominikBatorski & Wit Jakuczun for the connection & opportunities
(bit.ly/h2o_warsaw_1)



First @h2oai #rstats #meetup in #Poznan!
Many thanks to @mberesewicz
@adolfoalvarez #PAZUR
slideshare.net/JofaiChow/h2o-... Next stop:
#London



Good evening #Cologne 🇩🇪
#AroundTheWorldWithH2Oai
#CologneCathedral #Germany #twitter
bit.ly/2nJTxJG



Thanks #ViennaR #meetup for having us
@h2oai see you next time with more 🌎
next stop #Amsterdam #PyData
#AroundTheWorldWithH2Oai ✈️



Helping #Refugees by teaching them basic
#machinelearning skills @h2oai workshop
@Restart_Network #Rotterdam 🇳🇱
#AroundTheWorldWithH2Oai



Thx @DataScienceMi #datasciencemilan
@h2oai bit.ly/h2o_milan_1
#AroundTheWorldWithH2Oai #...
ift.tt/2dTKWDh



Ciao #Barcelona Thanks @aleixrvr #RugBcn
for having @h2oai we'll be back with
#SparklingWater 🍷 bit.ly/h2o_meetups next
stop ✈️ #Madrid



My 2nd @h2oai talk at #skillsmatter
#codenode bit.ly/joe_h2o_talk2 #twitter
ift.tt/20nUQeu



"When one drinks @h2oai, one must not forget where it comes from." Thank you
@Stanford for @h2oai & #useR2016 💦



Thx @fishnets88 & everyone
@GoDataDriven for hosting
@pydataamsterdam @h2oai tutorial
yesterday. Very cool office 😊
#AroundTheWorldWithH2Oai



Thanks @datatons @quierodata
@manuellamelas @lailaelqadi for organizing
the first @h2oai #Madrid #meetup we will be
back 🙏 next stop #Vienna

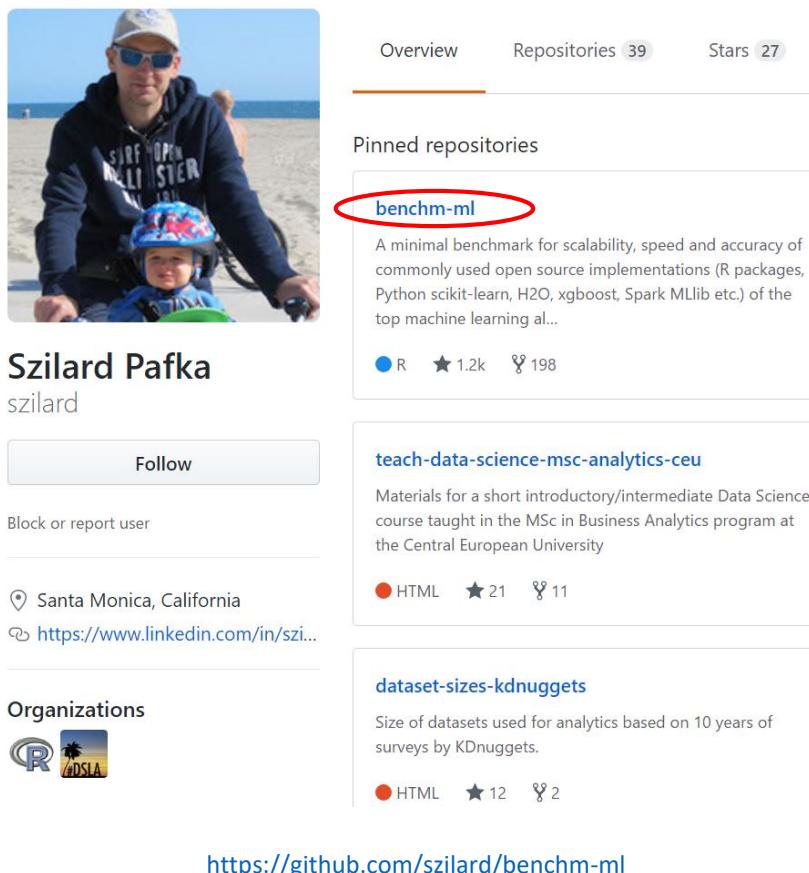


Thanks everyone for coming to my @h2oai
@BigData_LDN talk today, all our
conf/meetup slides -> github.com/h2oai/h2o-meet... #fullhouse 🙏



Why H₂O?

Szilard Pafka's ML Benchmark



A screenshot of Szilard Pafka's GitHub profile. It shows his profile picture, name, bio, and pinned repositories. The pinned repository 'benchm-ml' is highlighted with a red oval.

Szilard Pafka
szilard

[Follow](#)

Block or report user

Santa Monica, California

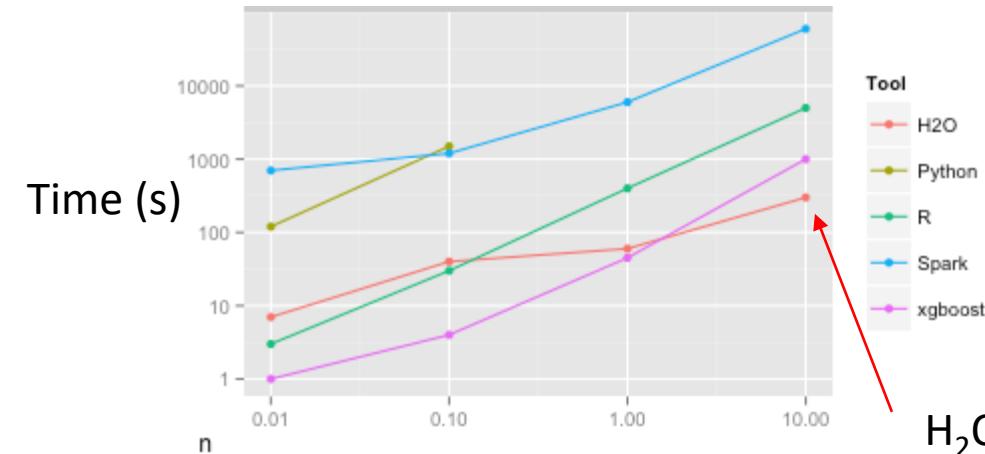
<https://www.linkedin.com/in/szilard-pafka/>

Organizations

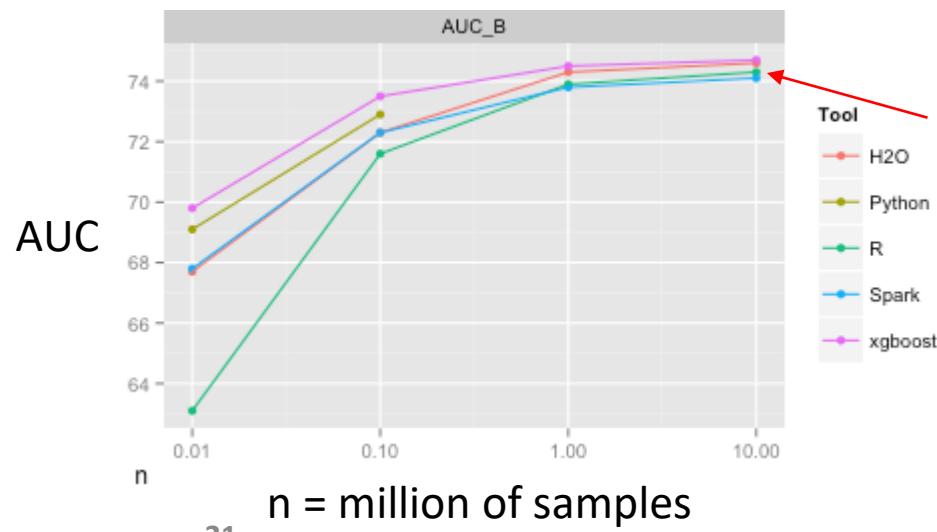
 RDSLA

<https://github.com/szilard/benchm-ml>

Gradient Boosting Machine Benchmark



H₂O is fastest at 10M samples



H₂O is as accurate as others at 10M samples

H₂O for Kaggle Competitions

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

Triskelion | 01.02.2015

[READ MORE](#)

“I did really like H2O’s deep learning implementation in R, though - the interface was great, the back end extremely easy to understand, and it was scalable and flexible. Definitely a tool I’ll be going back to.”

Kaggle challenge
2nd place winner
Colin Priest

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)

[READ MORE](#)

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

H₂O.ai

H₂O for Academic Research

European Journal of Operational Research

Available online 22 October 2016

In Press, Accepted Manuscript — Note to users



Innovative Applications of O.R.

Deep neural networks, gradient-boosted trees, random forests:
Statistical arbitrage on the S&P 500

Christopher Krauss^{1,a}, Xuan Anh Do^{1,a}, Nicolas Huck^{1,b}.

Received 15 April 2016, Revised 22 August 2016, Accepted 18 October 2016, Available online 22 October 2016

Highlights

- Latest machine learning techniques are deployed in a statistical arbitrage context.
- Deep neural networks, gradient-boosted trees, and random forests are considered.
- An equal-weighted ensemble of these techniques produces the best performance.
- Daily returns are substantial though declining over time.
- The system is especially effective at times of financial turmoil.

<http://www.sciencedirect.com/science/article/pii/S0377221716308657>

Cornell University Library

We gratefully acknowledge support from the Simons Foundation and member institutions

arXiv.org > physics > arXiv:1509.01199

Search or Article-id (Help | Advanced search) All papers ▾ Go!

Physics > Physics and Society

Inferring Passenger Type from Commuter Eigentravel Matrices

Erika Fille Legara, Christopher Monterola

(Submitted on 25 Aug 2015)

A sufficient knowledge of the demographics of a commuting public is essential in formulating and implementing more targeted transportation policies, as commuters exhibit different ways of traveling. With the advent of the Automated Fare Collection system (AFC), probing the travel patterns of commuters has become less invasive and more accessible. Consequently, numerous transport studies related to human mobility have shown that these observed patterns allow one to pair individuals with locations and/or activities at certain times of the day. However, classifying commuters using their travel signatures is yet to be thoroughly examined. Here, we contribute to the literature by demonstrating a procedure to characterize passenger types (Adult, Child/Student, and Senior Citizen) based on their three-month travel patterns taken from a smart fare card system. We first establish a method to construct distinct commuter matrices, which we refer to as eigentravel matrices, that capture the characteristic travel routines of individuals. From the eigentravel matrices, we build classification models that predict the type of passengers traveling. Among the models explored, the gradient boosting method (GBM) gives the best prediction accuracy at 76%, which is 84% better than the minimum model accuracy (41%) required vis-à-vis the proportional

Download:

- PDF
- Other formats (license)

Current browse context: physics.soc-ph
< prev | next >
new | recent | 1509

Change to browse by: cs cs.CY physics physics.data-an stat stat.AP stat.ML

References & Citations

- INSPIRE HEP (refers to | cited by)
- NASA ADS

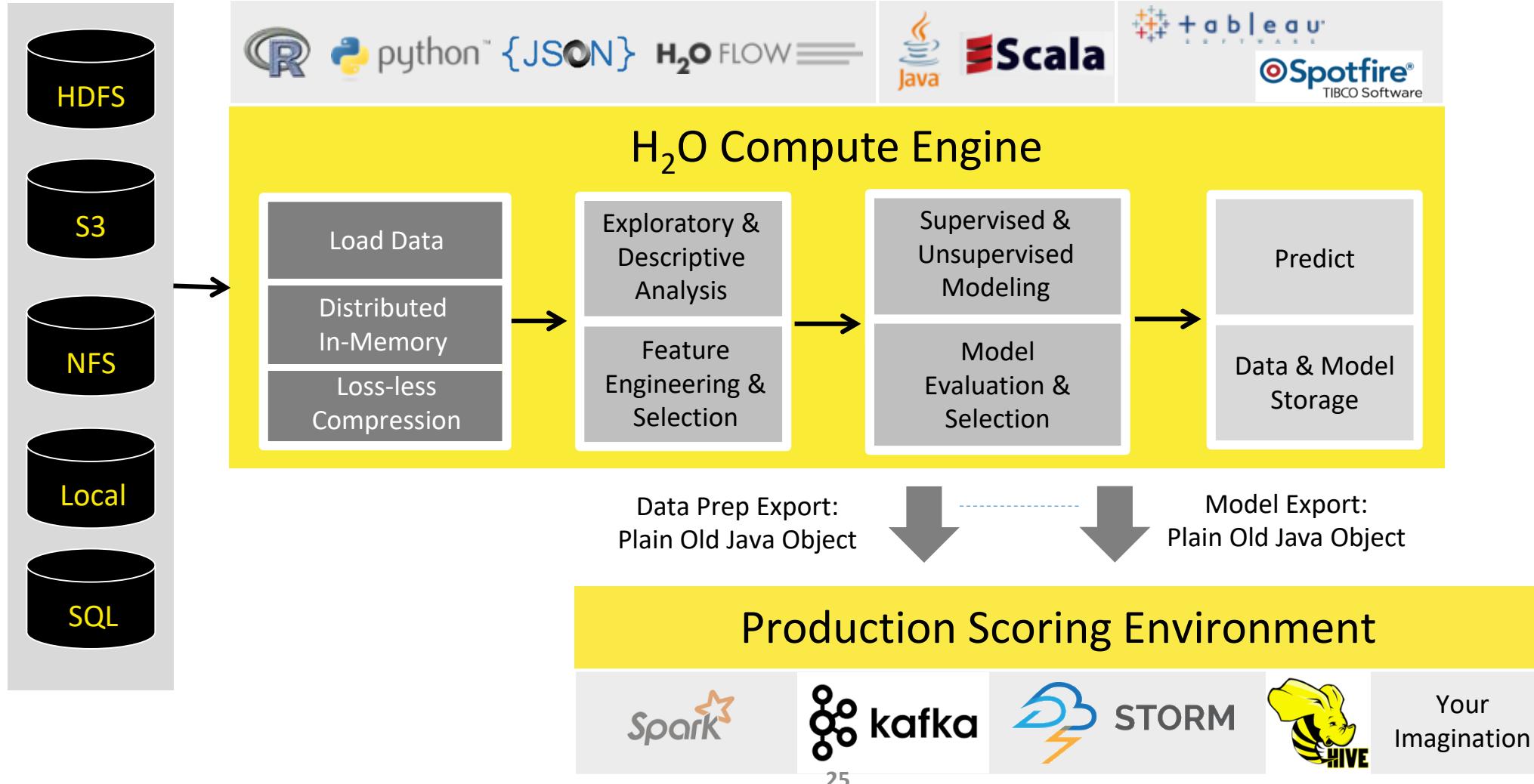
Bookmark (what is this?)



<https://arxiv.org/abs/1509.01199>

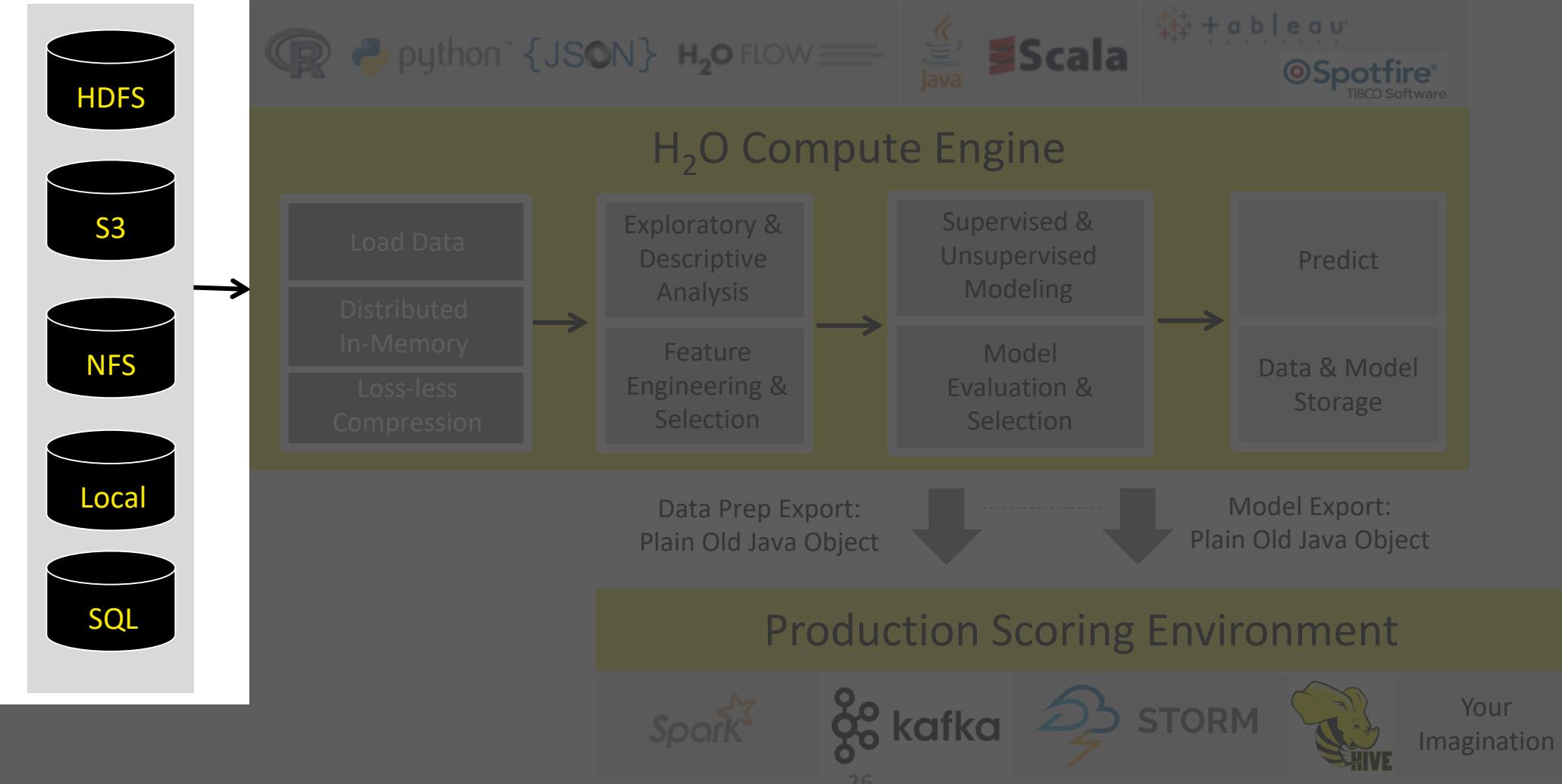
H₂O Machine Learning Platform

High Level Architecture



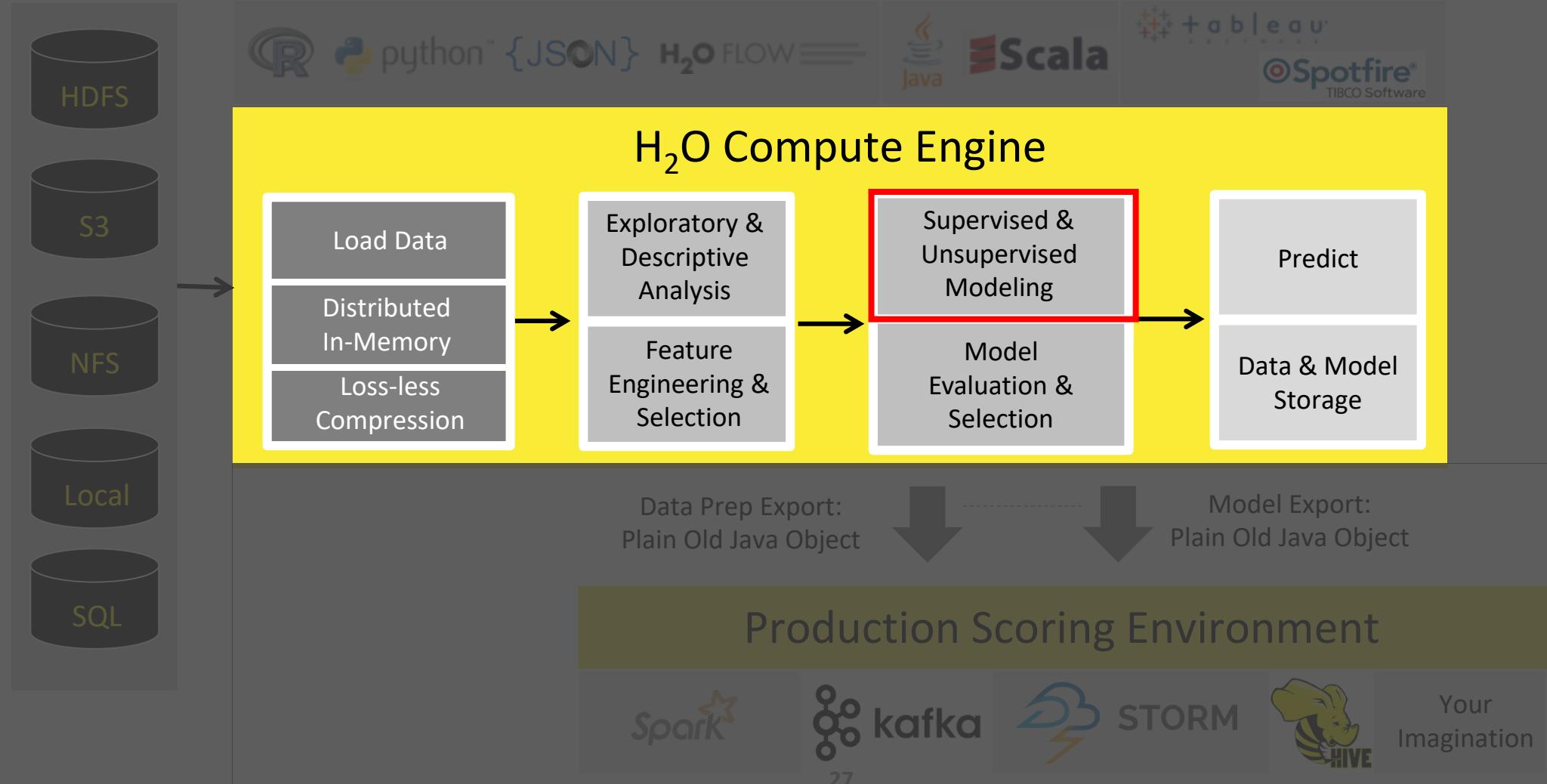
High Level Architecture

Import Data from
Multiple Sources



High Level Architecture

Fast, Scalable & Distributed
Compute Engine Written in
Java



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

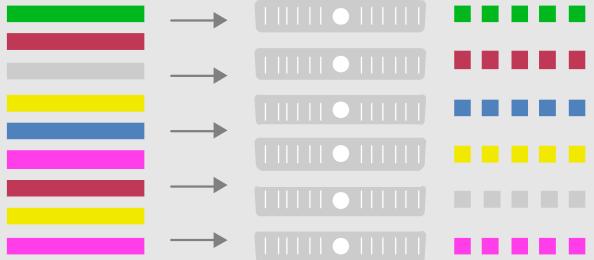
- **Principal Component Analysis:** Linearly transforms correlated variables to independent components
- **Generalized Low Rank Models:** extend the idea of PCA to handle arbitrary data consisting of numerical, Boolean, categorical, and missing data

Anomaly Detection

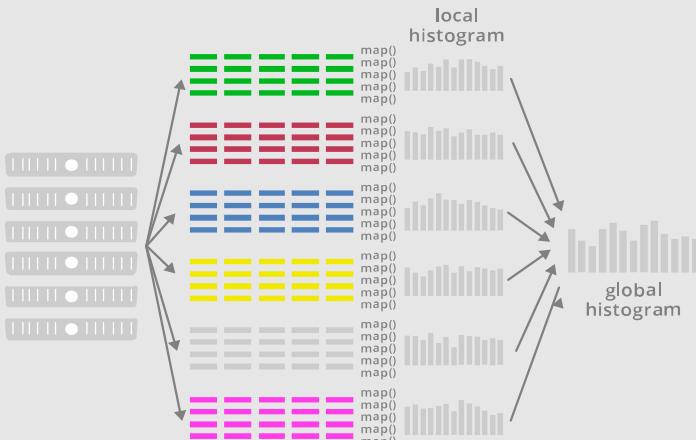
- **Autoencoders:** Find outliers using a nonlinear dimensionality reduction using deep learning

Distributed Algorithms

Foundation for Distributed Algorithms



Parallel Parse into **Distributed Rows**



Fine Grain Map Reduce Illustration: Scalable
Distributed Histogram Calculation for GBM

Advantageous Foundation

- Foundation for In-Memory Distributed Algorithm Calculation - **Distributed Data Frames** and **columnar compression**
- All algorithms are distributed in H₂O: GBM, GLM, DRF, Deep Learning and more. Fine-grained map-reduce iterations.
- **Only enterprise-grade, open-source distributed algorithms in the market**

User Benefits

- “Out-of-box” functionalities for all algorithms (**NO MORE SCRIPTING**) and uniform interface across all languages: R, Python, Java
- **Designed for all sizes of data sets, especially large data**
- **Highly optimized Java code for model exports**
- **In-house expertise for all algorithms**

H₂O 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 Q 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

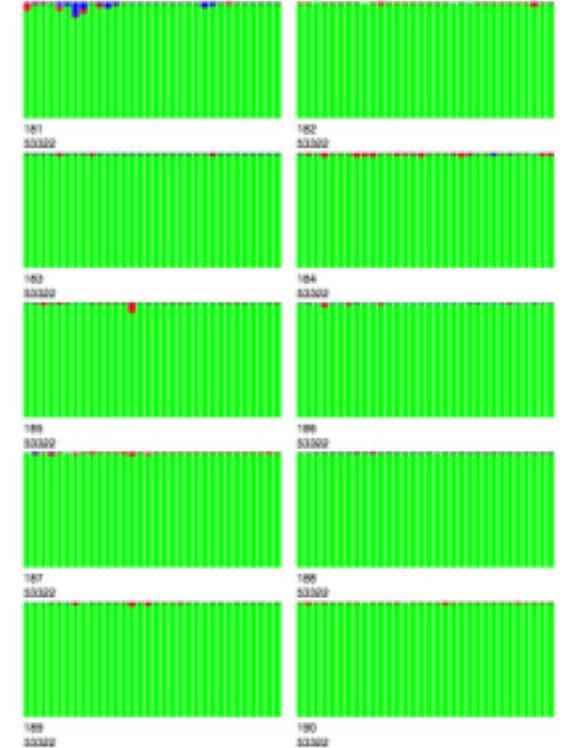
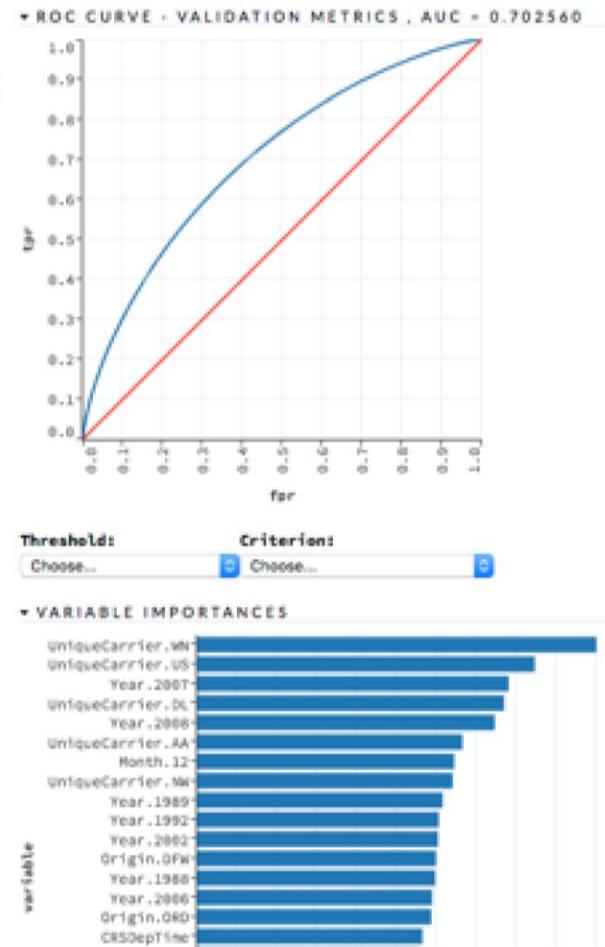
* OUTPUT - STATUS OF NEURON LAYERS (PREDICTING ISDELAYED, 2-CLASS CLASSIFICATION, BERNoulli DISTRIBUTION, CROSSENTROPY LOSS, 17,462 WEIGHTS/BIASES, 221.3 KB, 106,585,385 TRAINING SAMPLES, MINI-BATCH SIZE 1)

layer	units	type	dropout	l1	l2	mean_rate	rate_RMS	momentum	weight_RMS	mean_weight	weight_RMS	mean_bias	bias_RMS
1	887	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	0.0157	0.0227	0	-0.1833	0.5362	-1.3988	1.5259	
4	20	Rectifier	0	0	0	0.0517	0.0446	0	-0.1575	0.3068	-0.8846	0.6046	
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	0.0161	0.0083	0	0.0741	0.7268	0.4269	0.2056	

H₂O.ai

Deep Learning Model

real-time, interactive
model inspection in Flow



Legend

Each bar represents one CPU.

Blue: idle time

Green: user time

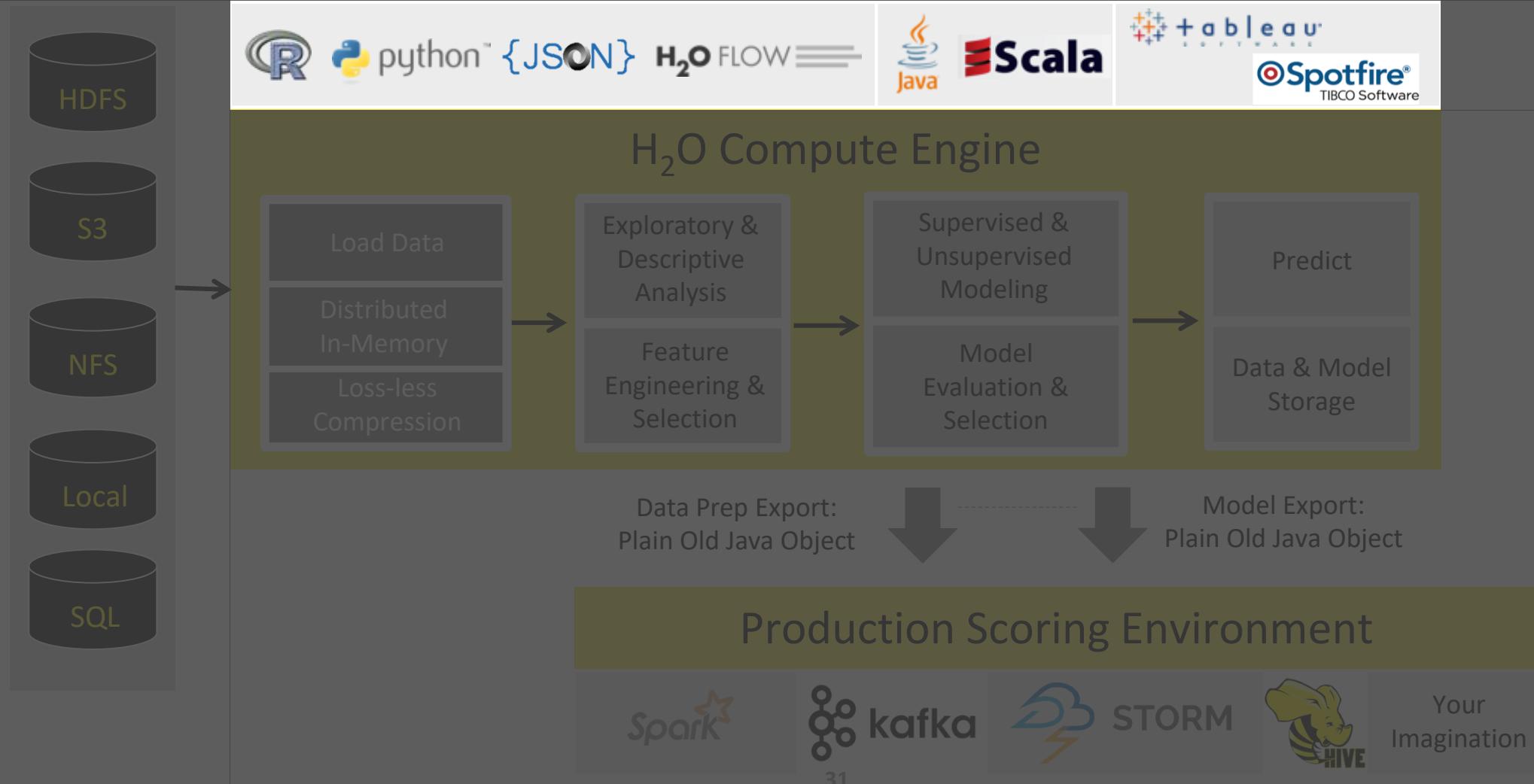
Red: system time

White: other time (e.g. Io)

10 nodes: all
320 cores busy



High Level Architecture



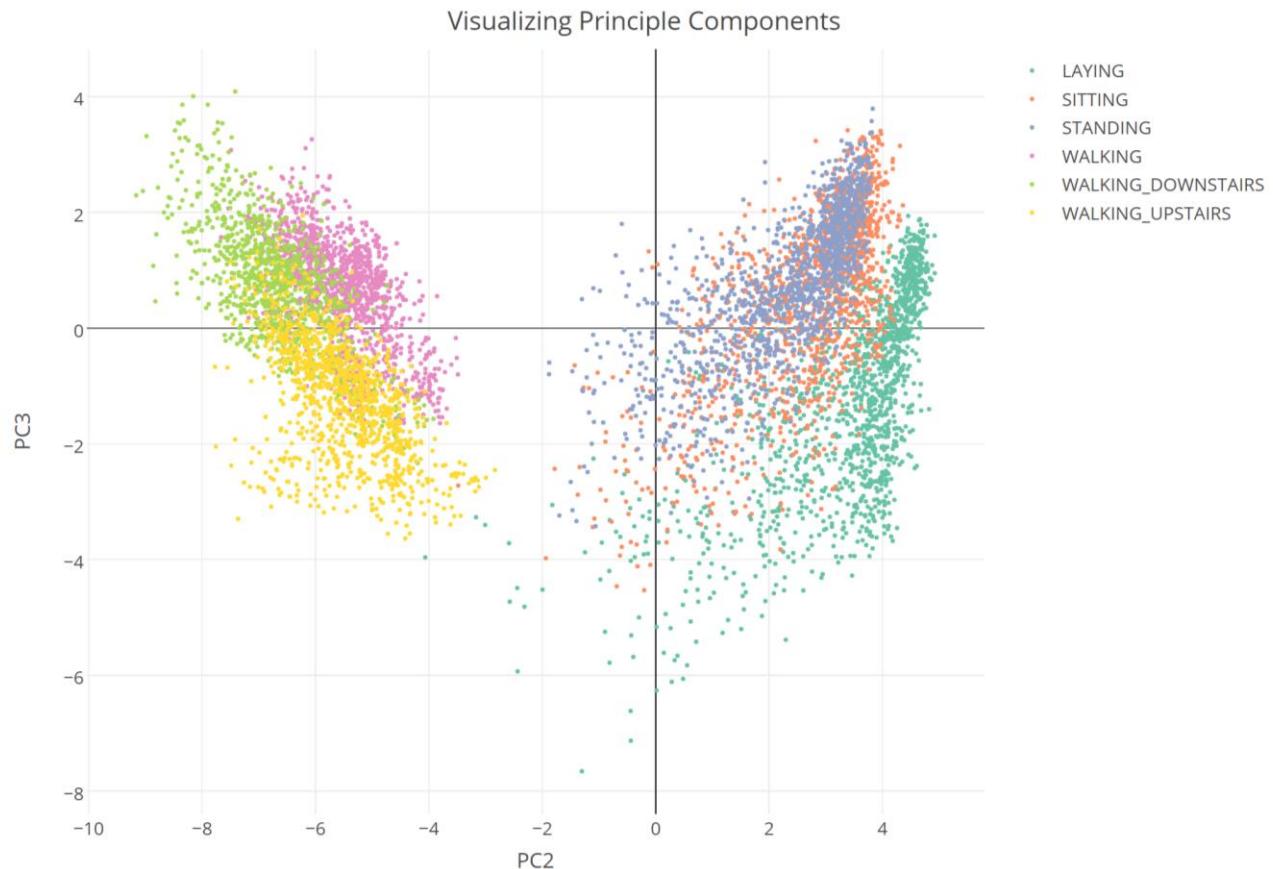
H₂O + R

```
# -----  
# 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)
```

Use H₂O with Other R Packages

1. Train a PCA model with H₂O
2. Visualize PCs with plotly
3. Deploy as a Shiny app

```
p <- plot_ly(data = d_pca, x = ~PC2, y = ~PC3, color = ~activity,
              type = "scatter", mode = "markers", marker = list(size = 3)) %>%
  layout(title = "Visualizing Principle Components")
p
```



From the graph above, we can see that:

- it could be difficult to distinguish between **Standing** and **Sitting** as there are large overlaps in their sensor data.
- **Laying** has its own cluster so it should be easy to classify.
- **Walking, Walking Upstairs** and **Walking Downstairs** are understandably closer to each other yet they are quite different to **Sitting**, **Standing** and **Laying**.

https://github.com/woobe/h2o_demos/tree/master/human_activity_recognition_with_smartphones

H2O Iris Demo x jo-fai

localhost:54321/flow/index.html#

H2O FLOW ≡ Flow ▼ Cell ▼ Data ▼ Model ▼ Score ▼ Admin ▼ Help ▼

Iris Demo

CS Expression...

File Folder Copy New Up Down Delete Edit Save Delete Run

Model... 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



H₂O + 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

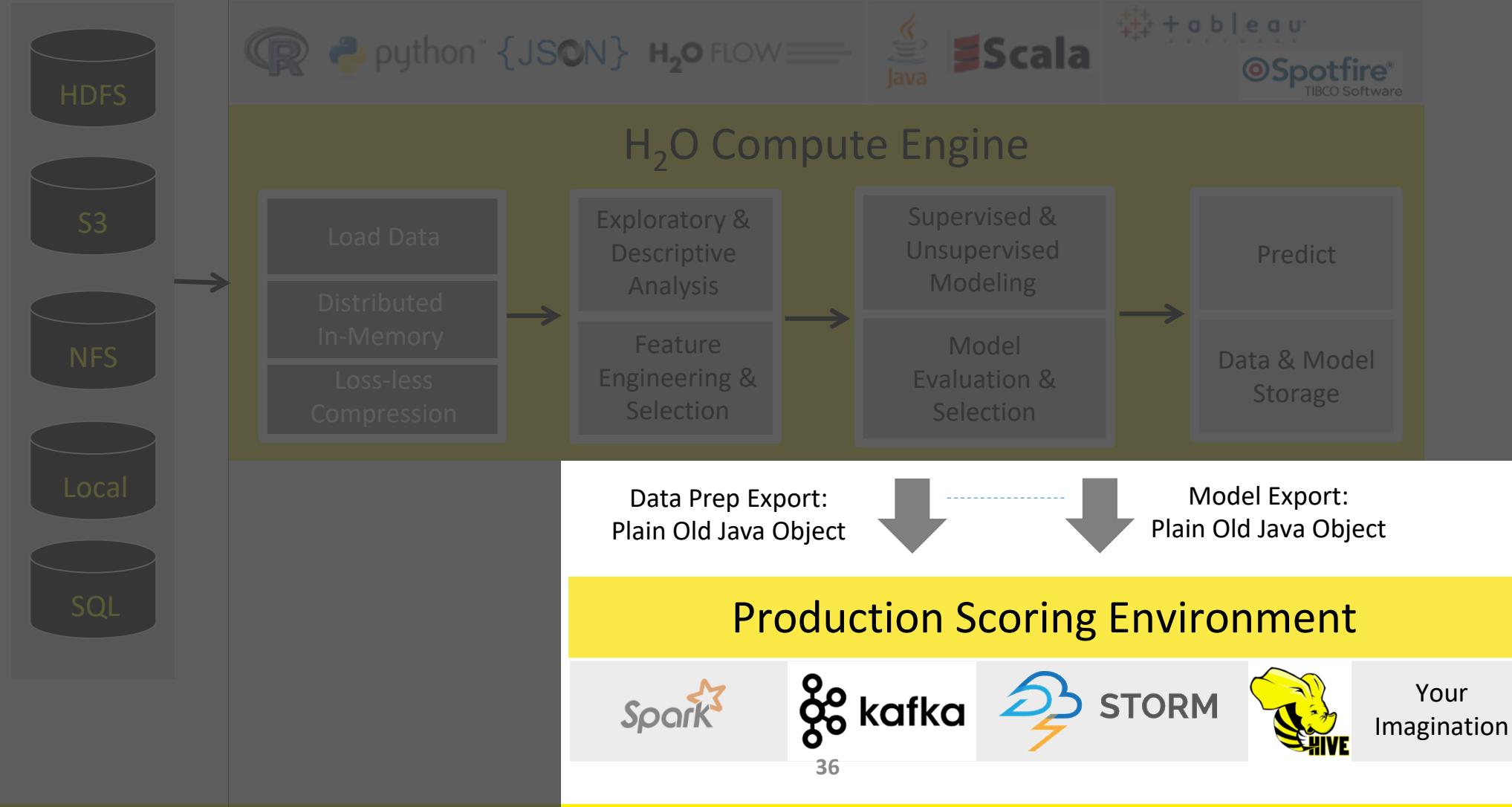
# Set up GBM for regression
# Add a seed for reproducibility
gbm_default = H2OGradientBoostingEstimator(model_id = 'gbm_default', seed = 1234)

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

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

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](#)

H₂O Tutorials

- Introduction to Machine Learning with H₂O
 - Basic Extract, Transform and Load (ETL)
 - Supervised Learning
 - Parameters Tuning
 - Model Stacking
- GitHub Repository
 - bit.ly/joe_h2o_tutorials
 - Both R & Python Code Examples Included



Jo-fai (Joe) Chow
@matlabulous

Replies to @matlabulous @h2oai @pydataamsterdam

Thx @fishnets88 & everyone
@GoDataDriven for hosting
@pydataamsterdam @h2oai tutorial
yesterday. Very cool office 😎
#AroundTheWorldWithH2Oai

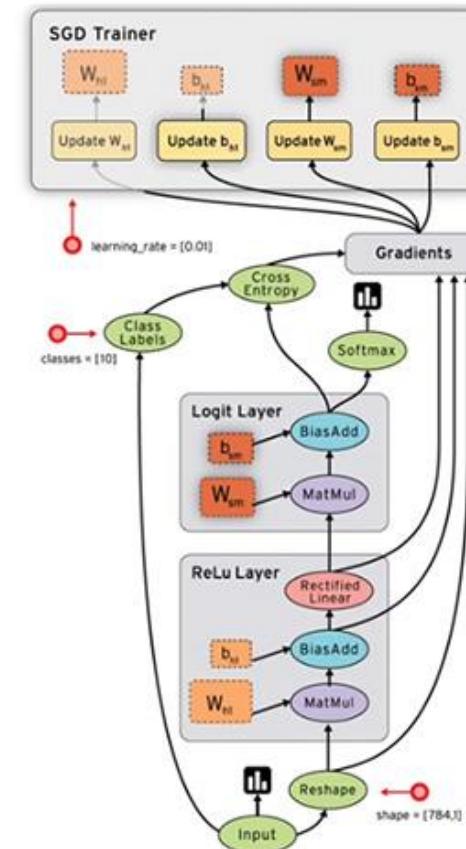


Deep Water

H₂O.ai Caffe  mxnet  TensorFlow

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 ...



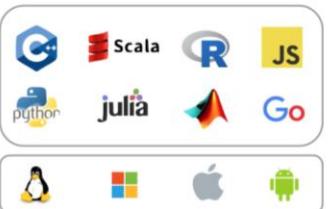
<https://github.com/tensorflow/tensorflow>



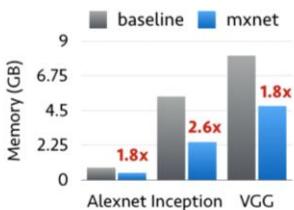
dmlc mxnet for Deep Learning

build passing docs latest license Apache 2.0

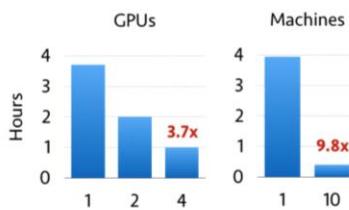
Portable



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>

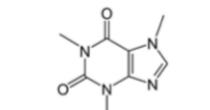
Caffe

- 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 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 Q_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

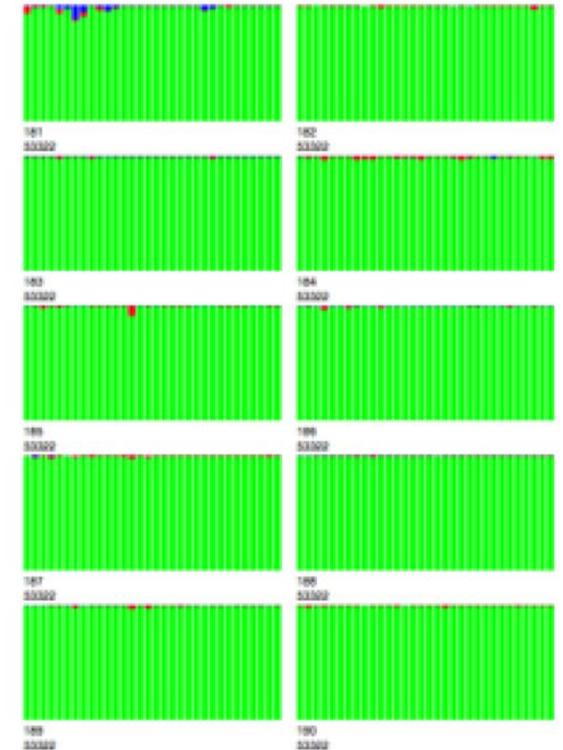
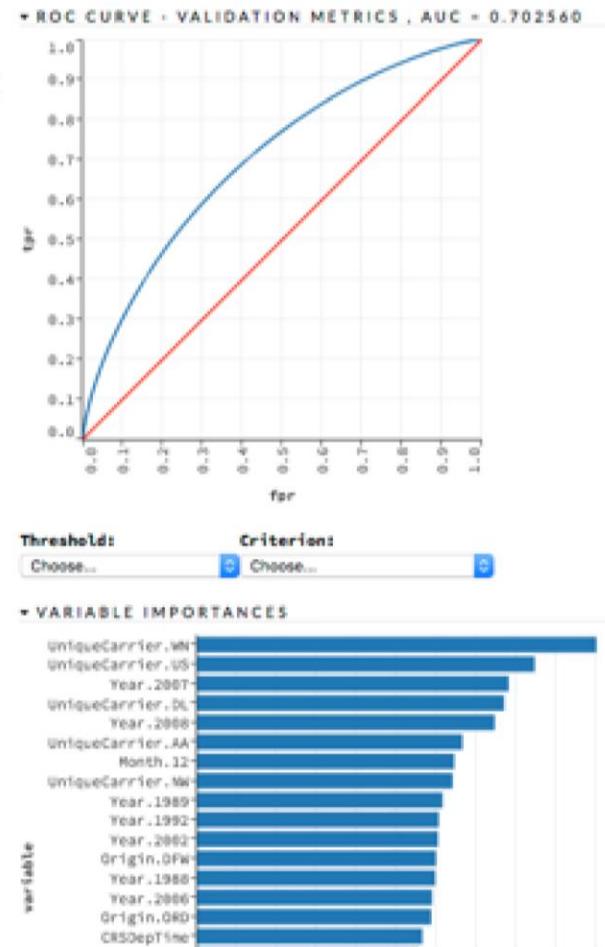
* OUTPUT - STATUS OF NEURON LAYERS (PREDICTING ISDELAYED, 2-CLASS CLASSIFICATION, BERNoulli DISTRIBUTION, CROSSENTROPY LOSS, 17,462 WEIGHTS/BIASES, 221.3 KB, 106,585,385 TRAINING SAMPLES, MINI-BATCH SIZE 1)

layer	units	type	dropout	l1	l2	mean_rate	rate_rms	momentum	mean_weight	weight_rms	mean_bias	bias_rms
1	887	Input	0									
2	20	Rectifier	0	0	0	0.0493	0.2029	0	-0.0021	0.2111	-0.9139	1.0036
3	20	Rectifier	0	0	0	0.0157	0.0227	0	-0.1833	0.5362	-1.3988	1.5259
4	20	Rectifier	0	0	0	0.0517	0.0446	0	-0.1575	0.3068	-0.8846	0.6046
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	0.0161	0.0083	0	0.0741	0.7268	0.4269	0.2056

H₂O.ai

Deep Learning Model

real-time, interactive
model inspection in Flow



10 nodes: all
320 cores busy



Both TensorFlow and H₂O are widely used

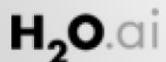
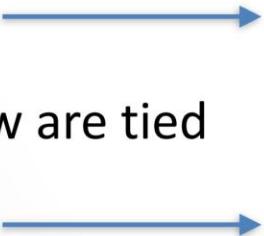
The usage of Hadoop/Big Data tools grew to 39%, up from 29% in 2015 (and 17% in 2014), driven by Apache Spark, MLlib (Spark Machine Learning Library) and H2O.

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!

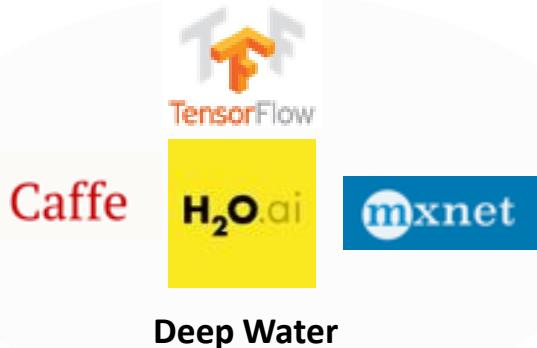


Deep Water

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**



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



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

H₂O

H₂O works with R, Python, Scala on Hadoop/Yarn, Spark or your laptop

The best machine learning platform on Spark and Hadoop

Interface using R, Python or intuitive web UI - Flow

Open Source Software, Apache 2.0 licensed

Data agnostic support for all common data and file types

Nanofast scoring engine

Export models as code or binary model files

[Download H₂O Latest Stable Release](#)

[Download H₂O Nightly Bleeding Edge](#)

H₂O is licensed under the [Apache License, Version 2.0](#)

H₂O with GPU-Enabled Machine Learning

H₂O Deep Water Deep Learning with MXNet, TensorFlow and Caffe XGBoost Gradient Boosting

Requirements:

- Ubuntu 16.04
- CUDA 8.0
- cuDNN 5.1

H₂O: wget
<http://s3.amazonaws.com/h2o-deepwater/public/nightly/latest/h2o.jar>
java -jar h2o.jar

Python: pip install
<http://s3.amazonaws.com/h2o-deepwater/public/nightly/latest/h2o-3.11.0-py2.py3-none-any.whl>

R: wget
http://s3.amazonaws.com/h2o-deepwater/public/nightly/latest/h2o_3.11.0.tar.gz
R CMD INSTALL
h2o_3.11.0.tar.gz

Docker image: Installation instruction:
<https://github.com/h2oai/deep-water#pre-release-docker-image>

Sparkling Water

H₂O Sparkling Water works with Spark 1.6 and Spark 2.0

The best machine learning platform on Spark and Hadoop

H₂O - the killer app for Spark

Seamlessly transition back and forth between Spark and H₂O

Open Source Software, Apache 2.0 licensed

Use Scala or Python to build models

Power of Spark combined with the speed of H₂O

All the features of H₂O included (Flow - UI, model export etc.)

[Download Sparkling Water 2.1](#)

[Download Sparkling Water 2.0](#)

Sparkling Water is licensed under the [Apache License, Version 2.0](#)

Steam

Build and deploy smart applications

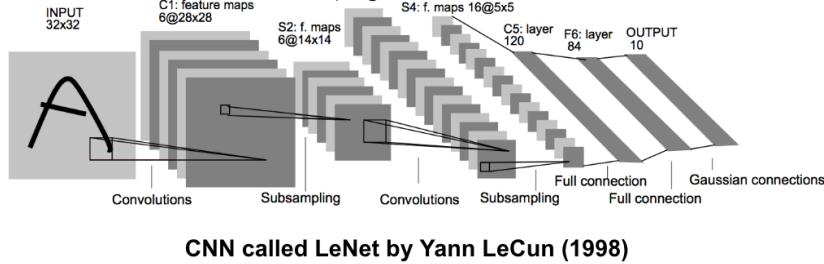
The best machine learning platform on Spark and Hadoop

Start H₂O Clusters on Yarn

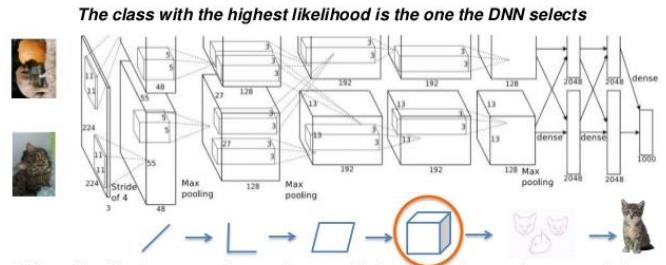
www.h2o.ai/download/

Available Networks in Deep Water

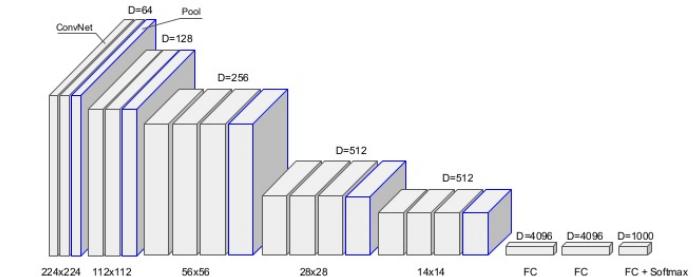
- LeNet
- AlexNet
- VGGNet
- Inception (GoogLeNet)
- ResNet (Deep Residual Learning)
- Build Your Own



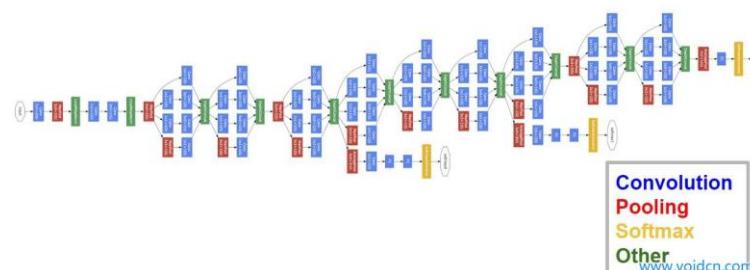
AlexNet (Krizhevsky et al. 2012)



Classical CNN topology - VGGNet (2013)

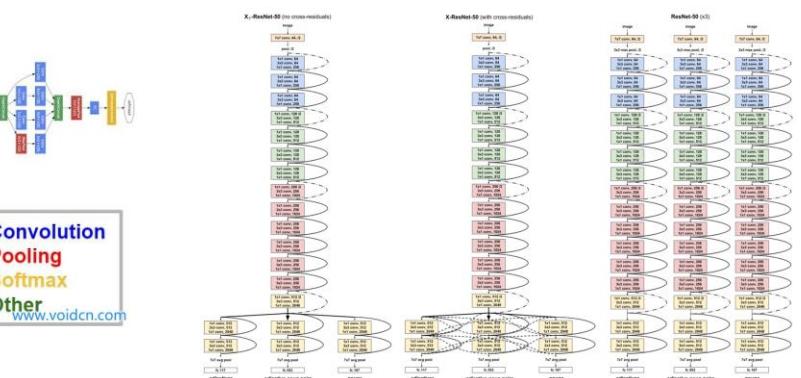


GoogLeNet



48

ResNet



Unified Interface (Deep Water + R)

```
# Train a LeNet with basic parameters and MXNet
model_mxnet <- h2o.deepwater(x = path,
                               y = response,
                               training_frame = df,
                               epochs = 300,
                               learning_rate = 1e-3,
                               image_shape = c(28, 28),
                               channels = 3,
                               backend = "mxnet",
                               network = "lenet")
```

Choosing different
network structures
and backends

Deep Water H2O and TensorFlow Demo



All None

Only show columns with more than % 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).

lenet
(Choose...) auto user
lenet
alexnet vgg googlenet inception_bn resnet

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 + 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

 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_incep...	Add credit card default risk model, update other notebooks.	3 months ago
 deeplearning_cat_dog_mouse_lenet....	Add credit card default risk model, update other notebooks.	
 deeplearning_cat_dog_mouse_lenet...	Add back model.plot() and scoring history.	
 deeplearning_cifar10_vgg.ipynb	Rename notebooks.	
 deeplearning_credit_card_default_ri...	changing the name of deeplearning_credit_card_default_risk_prediction...	
 deeplearning_ensemble_boston_ho...	Ensemble demo using GBM, DRF and Deep Water (#676)	
 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...	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.ip...	Add missing file.	3 months ago
 deeplearning_tensorflow_cat_dog_...	Add tensorflow example (#529)	2 months ago
 deeplearning_tensorflow_mnist.ipynb	Added MNIST example for TensorFlow	a month ago

Deep Water Example notebooks

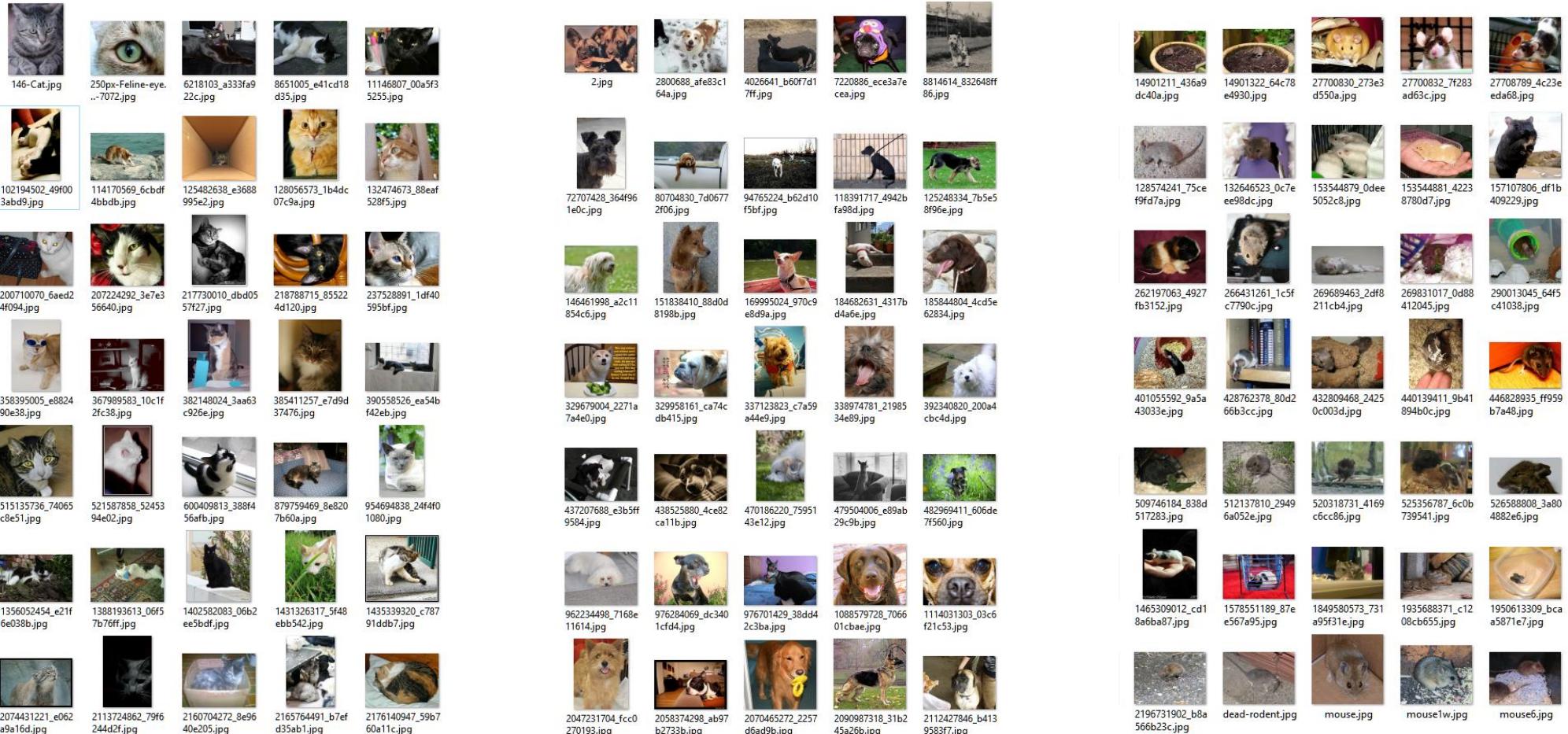
<https://github.com/h2oai/h2o-3/tree/master/examples/deeplearning/notebooks>

Deep Water Cat/Dog/Mouse Demo

Deep Water R Demo

- H₂O + MXNet + TensorFlow
 - Dataset – Cat/Dog/Mouse
 - MXNet & TensorFlow as GPU backend
 - Train LeNet (CNN) models
 - R Demo
- 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

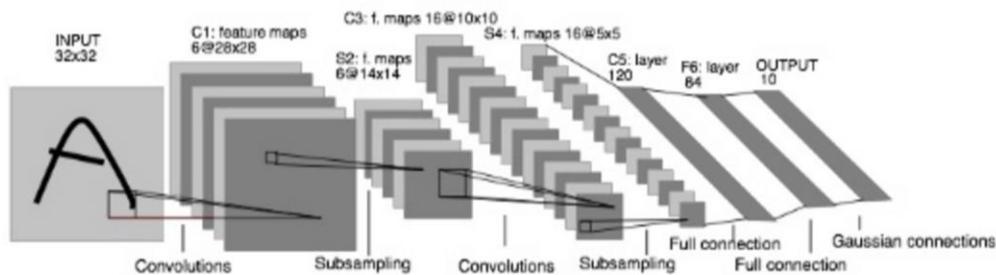
Live Demo if Possible

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]

```
# Train a LeNet with basic parameters and MXNet
model_mxnet <- h2o.deepwater(x = path,
                               y = response,
                               training_frame = df,
                               epochs = 300,
                               learning_rate = 1e-3,
                               image_shape = c(28, 28),
                               channels = 3,
                               backend = "mxnet",
                               network = "lenet")
```

```
# Train a LeNet with basic parameters and TensorFlow
model_tf <- h2o.deepwater(x = path,
                           y = response,
                           training_frame = df,
                           epochs = 300,
                           learning_rate = 1e-3,
                           image_shape = c(28, 28),
                           channels = 3,
                           backend = "tensorflow",
                           network = "lenet")
```

Easy Switch

Train a CNN (LeNet) Model on GPU

The image shows two terminal windows on a Linux system (Ubuntu 16.04 LTS) running on a GRID K520 GPU.

Top Terminal:

- Output of `gpustat -cp`:
- GPU [0]: GRID K520, Temperature: 34°C, Utilization: 76%, Memory Usage: 3806 / 4036 MB, Process: java/1357(3804M)
- Annotation: "Using GPU for training" points to the GPU utilization information.

Bottom Terminal:

- Output of `top`:
- Processor usage for cores 1-8: 15.2%, 26.2%, 14.0%, 11.1%, 10.8%, 11.0%, 10.5%, 14.5%.
- Memory usage: Mem 9.09G/14.7G, Swap 0K/0K.
- System status: Tasks: 50, 122 thr; 2 running, Load average: 0.33 0.21 0.15, Uptime: 06:54:06.

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),  
g) image size  
e  
                      channels=3)  
## provide expected (or matching)  
## 3 for color, 1 for monochrom
```

Point it to the custom
network structure file

Model

Note: Overfitting is expected as we only use a very small datasets to demonstrate the APIs only

Model Details:

=====

H20MultinomialModel: 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

H20MultinomialMetrics: 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

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.

- Latest Nightly Build

- <https://s3.amazonaws.com/h2o-deepwater/public/nightly/latest/h2o.jar>

- 100% Open Source

- The party will get bigger!



Other H₂O Developments

- H₂O + xgboost [[Link](#)]
- Stacked Ensembles [[Link](#)]
- Automatic Machine Learning [[Link](#)]
- Time Series [[Link](#)]
- High Availability Mode in Sparkling Water [[Link](#)]
- word2vec [[Link](#)]

H2O.ai
@h2oai

Following

Excited to be part of the #GPU Open Analytics Initiative w/ @ContinuumIO @MapD bit.ly/2pU0bks #GOAI

10 RETWEETS 20 LIKES

5:12 PM - 8 May 2017 from Middletown, NJ

Danke!

- Organizers & Sponsors
 - Berry Boessenkool & Berlin R
 - Nikola Chochkov & Adjust.com
- Code, Slides & Documents
 - bit.ly/h2o_meetups
 - 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)

