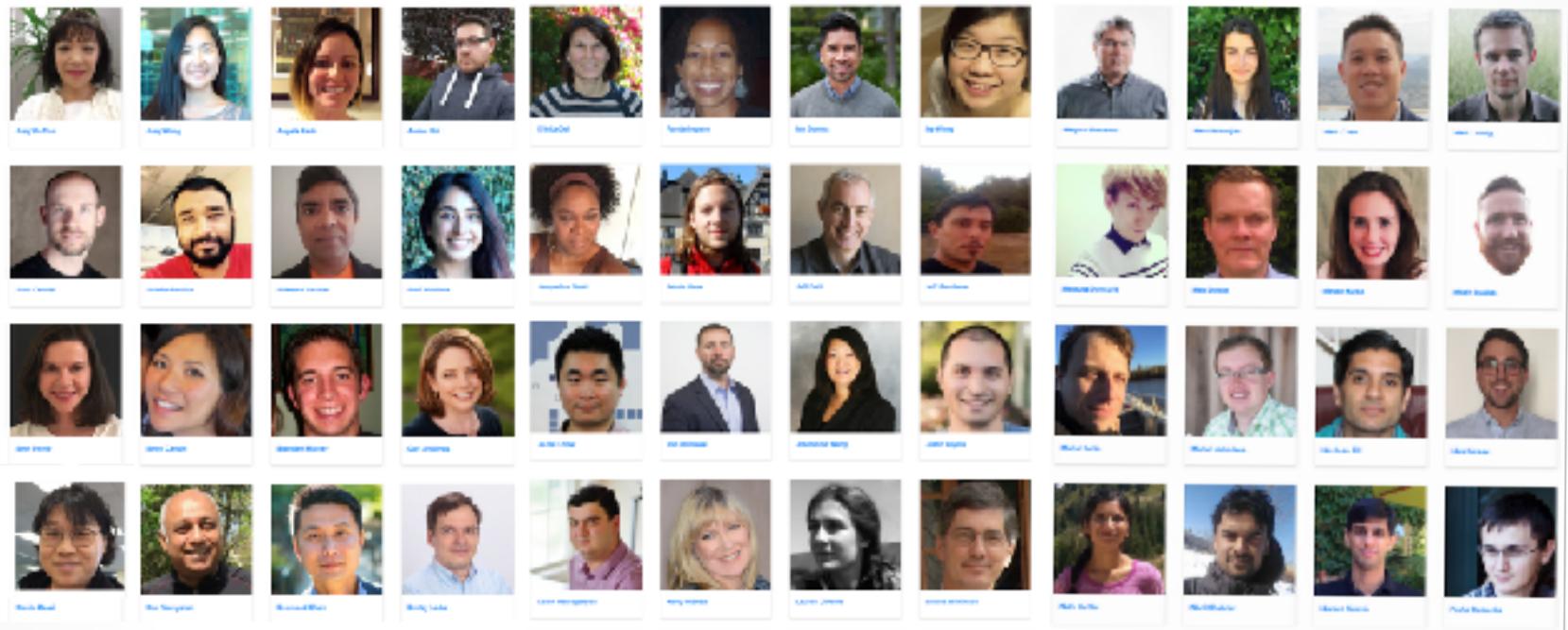


Deep Learning is Like Water: It's Everywhere!

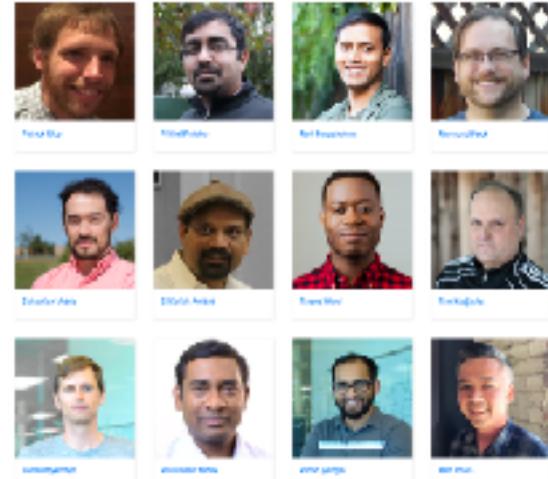
Arno Candel, PhD
CTO, H2O.ai
@ArnoCandel

AI By The Bay, San Francisco
March 6, 2017

Meet the H2O Makers



H₂O.ai



H₂O.ai

H₂O.ai

Software Product: H2O - AI for Business Transformation

- Distributed Data Frame with Scalable Execution Engine
- Distributed Algorithms *Deep Learning, GBM, RF, GLM, K-Means, PCA, ...*
- Apache v2 Open Source (github.com/h2oai)

H2O is Easy to Use and Deploy

- h2o.ai/download and run anywhere, immediately
- Client APIs: R, Python, Java, Scala, REST, Flow GUI
- Spark (cf. [Sparkling Water](#)), Hadoop, Bare Metal
- Productionize with auto-generated Java/C++ scoring code

Spark + H₂O

SPARKLING
WATER

H₂O.ai

H2O.ai - At the Core of AI



100 STARTUPS USING ARTIFICIAL INTELLIGENCE TO TRANSFORM INDUSTRIES

CONVERSATIONAL AI/ BOTS



VISION



AUTO



ROBOTICS



CYBERSECURITY



BUSINESS INTELLIGENCE & ANALYTICS



CORE AI



AD, SALES, CRM



HEALTHCARE



FINTECH & INSURANCE

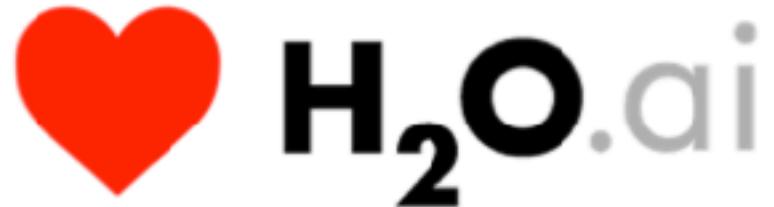


OTHER



H2O.ai - Loved By The Best

107 out of Fortune 500



8 out of 10

Banks

7 out of 10

Insurance

4 out of 10

Healthcare

H2O.ai - Visionary in 2017 Gartner MQ for Data Science



H2O Deep Water got (Tech) Crunch'ed

TC News Video Events Crunchbase [Search](#)

CRUNCHBOARD Put Your Job In Front Of 14M TechCrunch Readers Post Your Job Today →

Artificial Intelligence Water Software deep learning H2O.ai

H2O's Deep Water puts deep learning in the hands of enterprise users

Posted Jan 26, 2017 by [John Mannes \(@JohnMannes\)](#)

[Next](#)

To complement existing offerings like Sparkling Water and Steam, [H2O.ai](#) is releasing Deep Water, a new tool to help businesses make deep learning a part of everyday operations.

Deep Water will open up new possibilities for the TensorFlow, MXNet and Caffe communities to

H2O.ai - Highlighted by Fortune Magazine



ARTIFICIAL INTELLIGENCE

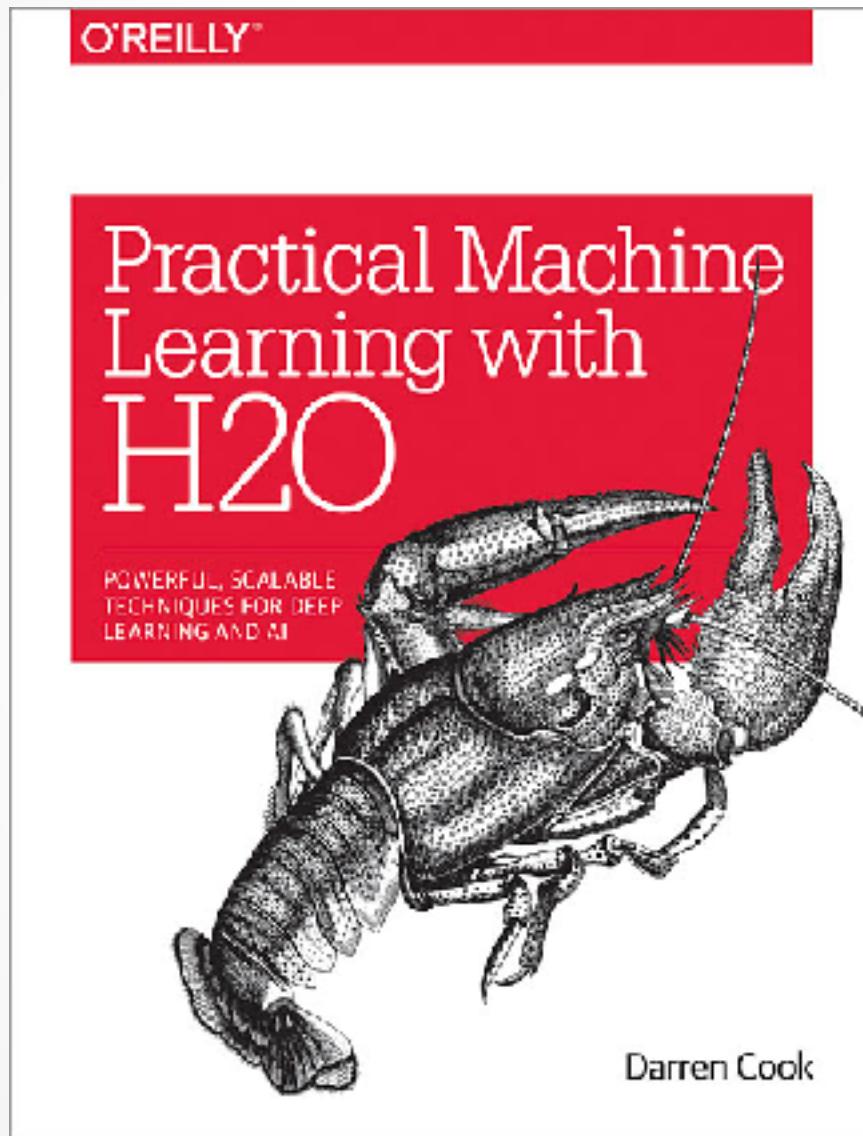
Here are 50 Companies Leading the AI Revolution

Brian O'Keefe,
Nicolas Rapp
Feb 23, 2017



<http://fortune.com/2017/02/23/artificial-intelligence-companies/>

H2O Book - Written by the Community



**Powerful, Scalable
Techniques for Deep
Learning and AI**

brand new: Dec 2016

H2O.ai Customer Love

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



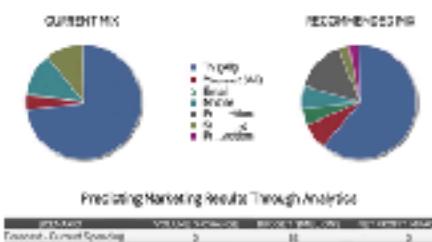
See how Progressive uses H2O predictive analytics for User-based Insurance (UBI).

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.

H2O.ai Customer Love

Hospital Corporation of America



HCA uses H2O to predict patient outcomes in real-time.

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



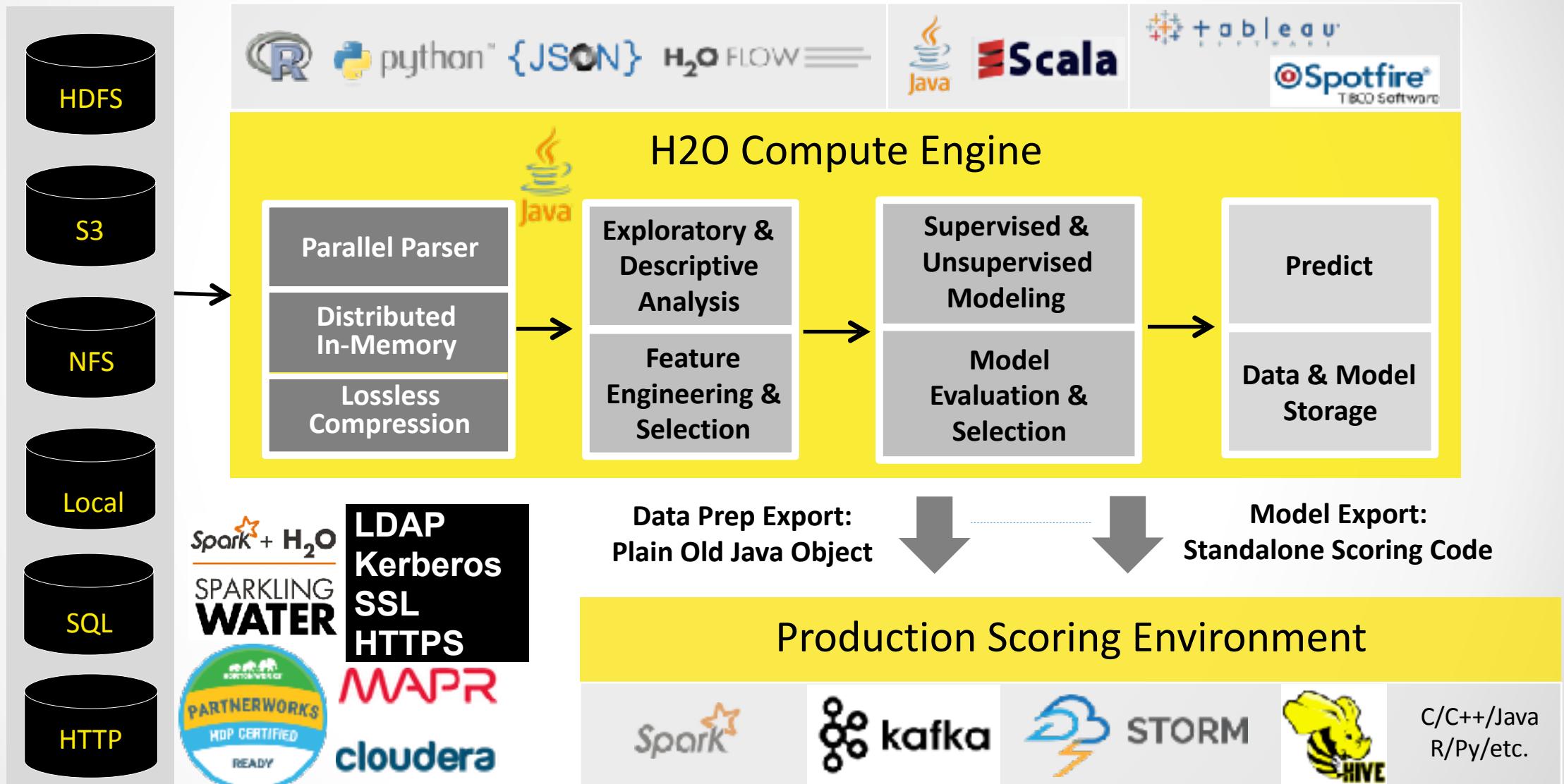
Cisco uses H2O to build a scalable model factory to improve sales and marketing.

Nielsen Catalina Solutions



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

High Level Architecture of H2O



Native APIs: Java, Scala — REST APIs: R, Python, Flow, JavaScript, Java



{ REST }

```
library(h2o)
h2o.init()
h2o.deeplearning(x=1:4,y=5,as.h2o(iris))
```

Scala

```
import _root_.hex.deeplearning.DeepLearning
import _root_.hex.deeplearning.DeepLearningParameters
val dlParams = new DeepLearningParameters()
dlParams._train = iris.hex
dlParams._response_column = 'Species
val dl = new DeepLearning(dlParams)
val dlModel = dl.trainModel.get
```

H₂O.ai



```
import h2o
from h2o.estimators.deeplearning import H2ODeepLearningEstimator
h2o.init()
dl = H2ODeepLearningEstimator()
dl.train(x=list(range(1,4)), y="Species", training_frame=iris.hex)
```

Spark + H₂O

SPARKLING
WATER



H₂O FLOW

Built-in interactive GUI and
notebook - no coding needed!

Job

Run Time 00:00:58.963

Remaining Time 00:00:04.726

Type Model

Key [deeplearning-dd2f42f2-81f7-42a8-9d98-a34437309828](#)

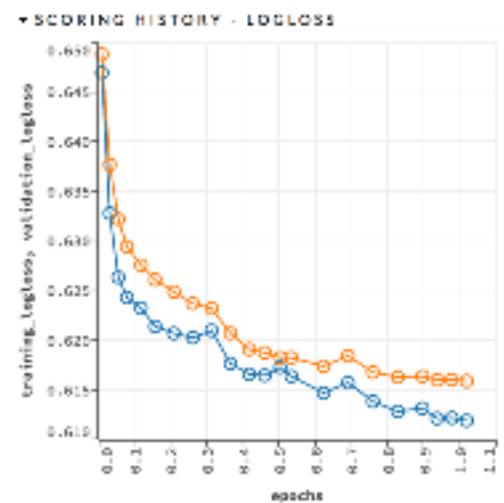
Description DeepLearning

Status RUNNING

Progress 93%

Iterations: 21. Epochs: 0.094657. Speed: 2,634,276 samples/sec. Estimated time left: 7.460 sec

Actions [View](#) [Cancel Job](#)



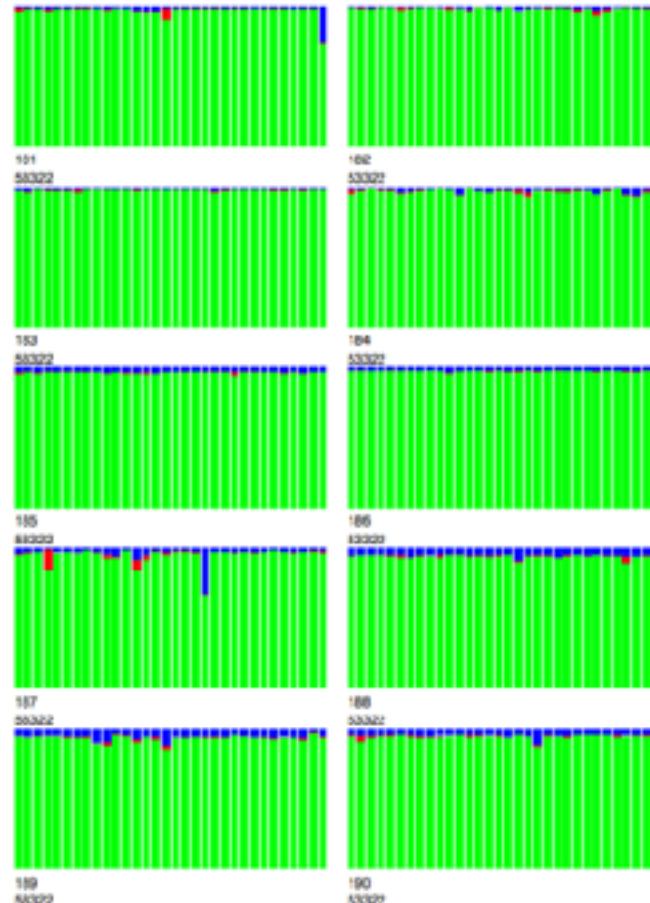
All heavy lifting is done by the backend!

Live Demo of Distributed Deep Learning in H2O

Airline dataset: 116M flights in the U.S. over 20 years

CHUNK COMPRESSION SUMMARY					
chunk_type	chunk_name	count	count_percentage	size	size_percentage
CBL	Constant Integers	2622	15.2853	384.8 KB	8.0118
CBS	Bits	1438	8.3391	34.8 KB	0.8282
CXT	Zero Sparse Integers	7	0.0406	275.6 KB	0.0159
C1	1-Byte Integers	1	0.0056	63.1 KB	0.0046
CIN	1-Byte Integers (w/o NAs)	4824	26.8151	358.4 MB	21.3781
CIS	1-Byte Fractions	72	0.4175	5.6 MB	0.3383
C2	2-Byte Integers	6480	49.1765	1.28 GB	77.6330

FRAME DISTRIBUTION SUMMARY				
	size	number_of_rows	number_of_chunks_per_column	number_of_chunks
172.16.2.181:53322	176.6 MB	12167007.0	143.0	1716.0
172.16.2.182:53322	168.5 MB	11633924.0	143.0	1716.0
172.16.2.183:53322	168.2 MB	11613228.0	143.0	1716.0
172.16.2.184:53322	168.1 MB	11595969.0	143.0	1716.0
172.16.2.185:53322	169.4 MB	11612917.0	143.0	1716.0
172.16.2.186:53322	168.8 MB	11622587.0	143.0	1716.0
172.16.2.187:53322	167.9 MB	11628626.0	143.0	1716.0
172.16.2.188:53322	169.9 MB	11599977.0	143.0	1716.0
172.16.2.189:53322	169.8 MB	11601393.0	143.0	1716.0
172.16.2.190:53322	168.6 MB	11618868.0	143.0	1716.0
mean	169.3 MB	11669525.9880	143.7030	1724.4800
min	167.9 MB	11595969.0	143.0	1716.0
max	176.6 MB	12167007.0	150.0	1800.0
stddev	2.5 MB	165511.9476	2.1030	25.2800
total	1.65 GB	116695259.0	1437.0	17244.0



10x in-memory compression vs CSV

All cluster CPU cores are busy

Brand-new: H2O XGBoost Integration (Gradient Boosting)

Why XGBoost?

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

GPU support (for small/medium data)

Efficient on **sparse** data

Why integrate into H2O?

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

Real-time model status (var imp, metrics)

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

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

Live Demo of GPU Gradient Boosting in H2O

Build a Model

Select an algorithm: XGBoost

booster: gbtree

reg_lambda: 1

reg_alpha: (Choose...) auto

backend: gpu

NVIDIA-SMI 375.28 Driver Version: 375.28

GPU Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC
Fan Temp	Perf Pwr. Usage/Cap	Memory-Usage	GPU-Util	Compute M.
8 GeForce GTX 1080	Off	0000:02:00.0	On	N/A
27% 43C	P2 83W / 188W	2840MiB / 8145MiB	94%	Default

VARIABLE IMPORTANCES

Variable	Importance
v50	High
v65.C	Medium-High
v46	Medium-High
v21	Medium-High
v65.B	Medium-Low
v18	Medium-Low
v114	Medium-Low
v47.C	Medium-Low
v129	Medium-Low
v38	Medium-Low
v12	Low

ROC CURVE - VALIDATION METRICS , AUC = 0.750331

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

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

Deep Water: Best Open-Source Deep Learning

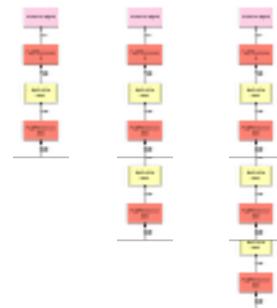
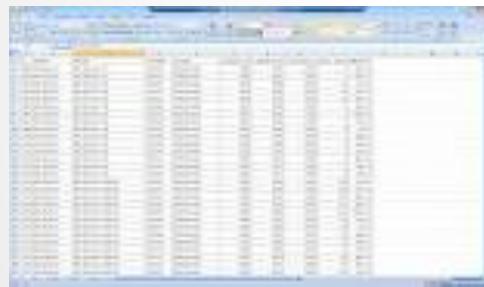
Enterprise Deep Learning for Business Transformation

Deep Water = THE Deep Learning Platform	H2O integrates the top open-source DL tools	
Native GPU support	  is up to 100x faster than	
Enterprise Ready	Easy to train and deploy, interactive, scalable, etc. Flow, R, Python, Spark/Scala, Java, REST, POJO, Steam	
New Big Data Use Cases (previously impossible or difficult in H2O)	Image - social media, manufacturing, healthcare, ... Video - UX/UI, security, automotive, social media, ... Sound - automotive, security, call centers, healthcare, ... Text - NLP, sentiment, security, finance, fraud, ... Time Series - security, IoT, finance, e-commerce, ...	

Deep Water Brings State-Of-The-Art Deep Learning on GPUs to H2O

H2O Deep Learning: simple multi-layer networks, CPUs

1-5 layers
MBs/GBs of data

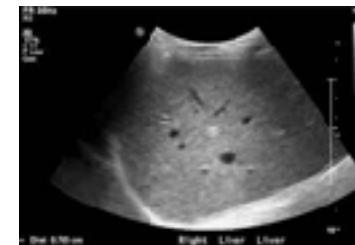


Limited to business analytics,
statistical models (CSV data)

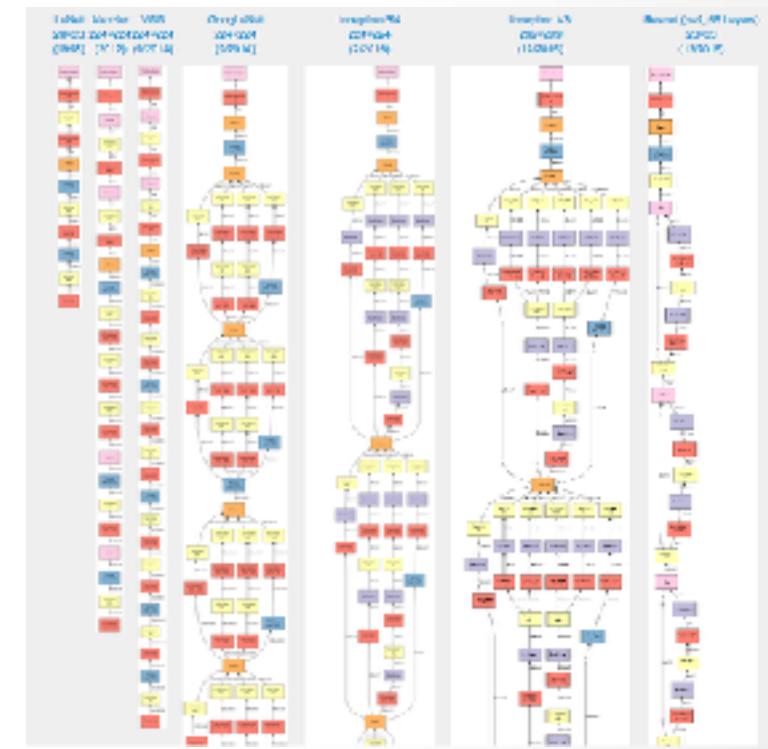


H2O Deep Water: arbitrary networks, CPUs or GPUs

1-1000 layers
GBs/TBs of data



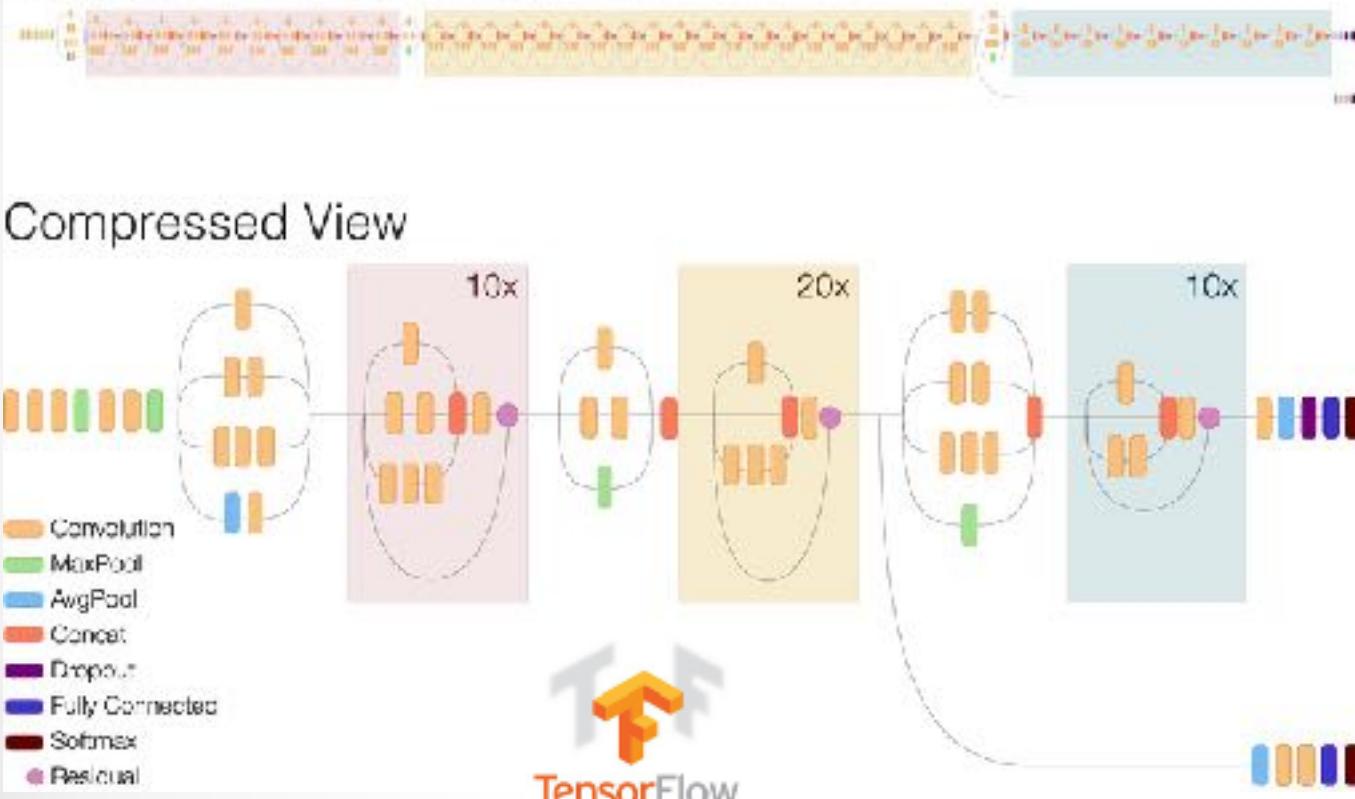
Large networks for big data
(e.g. image 1000x1000x3 -> 3m inputs per observation)



Open-Source - Leverage Community Code, Data and Models

Best Image Classifier as of Aug 2016: Google + Microsoft Hybrid Architecture

Inception Resnet V2 Network



<https://research.googleblog.com/2016/08/improving-inception-and-image.html>

open-source implementation

```
stem_2_1x1 = Conv(data=concat1, num_filter=num_4_1,
stem_2_7x1 = Conv(data=stem_2_1x1, num_filter=num_4_
    suffix='_conv_2')
stem_2_1x7 = Conv(data=stem_2_7x1, num_filter=num_4_
    suffix='_conv_3')
stem_2_3x3 = Conv(data=stem_2_1x7, num_filter=num_4_
concat2 = mx.sym.Concat(*[stem_1_3x3, stem_2_3x3], 1
pool2 = mx.sym.Pooling(data=concat2, kernel=(3, 3),
    name='%s_pool2' % ('max',
stem_3_3x3 = Conv(data=concat2, num_filter=num_5_1,
    suffix='_conv_1', withRelu=False)

concat3 = mx.sym.Concat(*[pool2, stem_3_3x3], name=
bn1 = mx.sym.BatchNorm(data=concat3, name='%s_bn1'
act1 = mx.sym.Activation(data=bn1, act_type='relu')

return act1

def InceptionResnetV2A(data,
    num_1_1,
    num_2_1, num_2_2,
    num_3_1, num_3_2, num_3_3,
    proj,
    name,
    scaleResidual=True):
    import mxnet as mx
    init = data

    a1 = Conv(data=data, num_filter=num_1_1, name='%s_a1'
    a2 = Conv(data=a1, num_filter=num_2_1, name='%s_a2'
    a2 = Conv(data=a2, num_filter=num_2_2, kernel=(3, 3)

    a3 = Conv(data=a2, num_filter=num_3_1, name='%s_a3'
    a3 = Conv(data=a3, num_filter=num_3_2, kernel=(3, 3)
    a3 = Conv(data=a3, num_filter=num_3_3, kernel=(3, 3)
```



H2O takes DL model definition as input

Live Demo of Deep Water for Image Classification on GPUs

```
frame = h2o.import_file(PATH+"/bigdata/laptop/deepwater/imagenet/cat_dog_mouse.csv")
print(frame.head(5))
```

Parse progress: |██████████| 100%

C1	C2
bigdata/laptop/deepwater/imagenet/cat/102194502_49f003ahd9.jpg	cat
bigdata/laptop/deepwater/imagenet/cat/11146807_00a5f35255.jpg	cat
bigdata/laptop/deepwater/imagenet/cat/1140846215_70e3261868.jpg	cat

```
nclassees = frame[1].nlevels()[0]
get_symbol(nclassees).save("/tmp/symbol_inception_resnet_v2-py.json")
model = H2ODDeepWaterEstimator(epcchs=20, learning_rate=1e-3, learning_rate_annealing=1e-5,
                               mini_batch_size=16,
                               network_definition_file="/tmp/symbol_inception_resnet_v2-py.json",
                               image_shape=[299,299],
                               channels=3)
model.train(x=[0],y=1, training_frame=frame)
model.show()
```

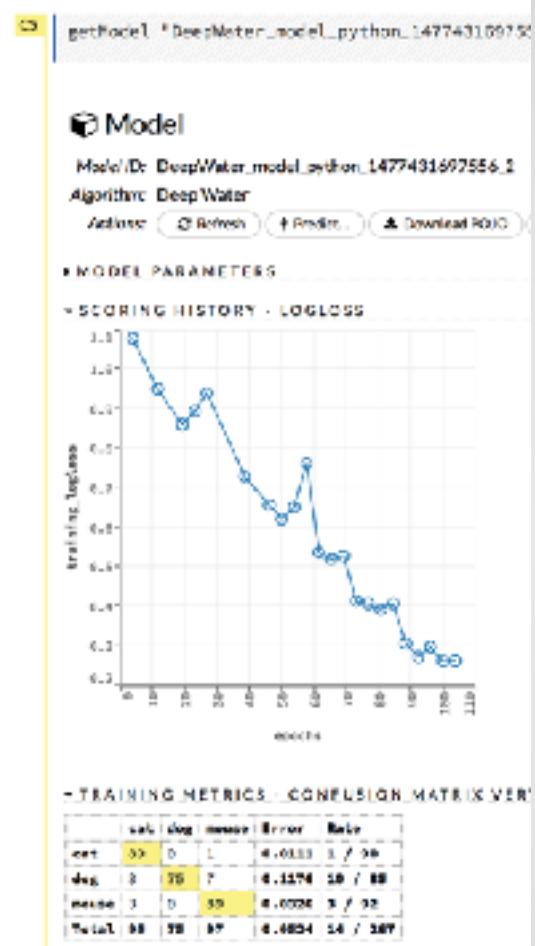
deeptwater Model Build progress: |██████████| 100%

Model Details

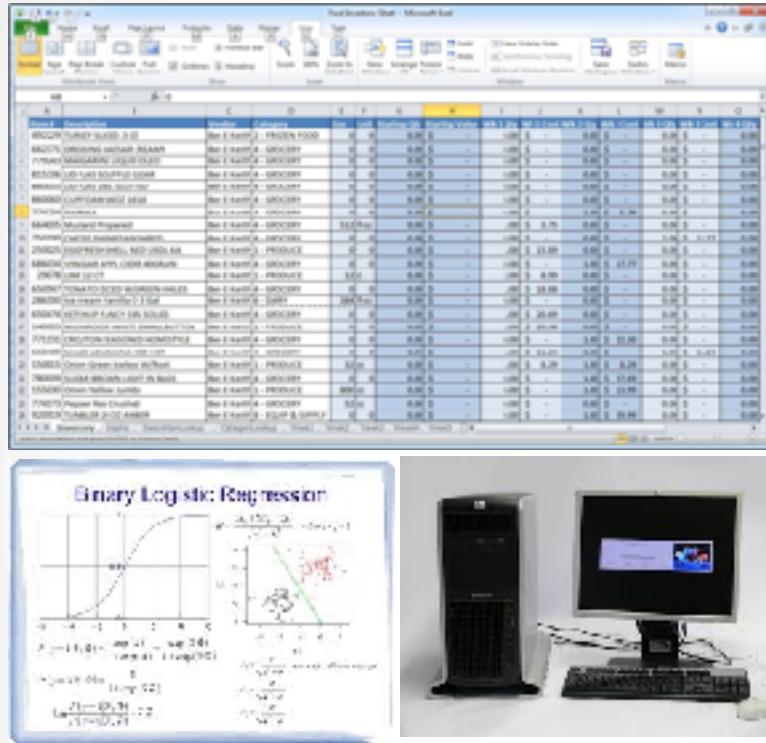
H2ODDeepWaterEstimator : Deep Water

Model Key: DeepWater_model_python_1477179782032_5

Status of Deep Learning Model: user, 116.1 MB, predicting C2, 3-class classification, 5,632 training samples, mini-batch size 16



Things are Changing Quickly



Yesterday: Small Data (<GB)

Data + Skills

H₂O.ai are good for business



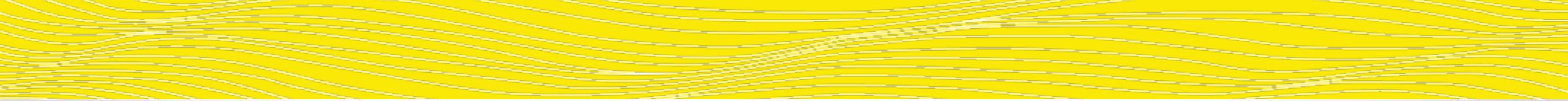
Today: Big Data (TeraBytes, ExaBytes)

Data + Machine Learning

ARE the business

$$\begin{aligned} \langle \phi_n | \phi_m \rangle &= \left\langle \phi_n \left| \int dx |\psi\rangle \langle \psi| \right. \right\rangle \quad (4.1-1) \\ \phi_n(x) &= C_n |\phi_n\rangle \xrightarrow{\text{def}} \phi_n''(x) = \phi_n/x^2 \quad (4.1-2) \\ \langle \phi_n | \phi_m \rangle &= \int_{-\infty}^{\infty} dx |\phi_n(x)|^2 = \int_{-\infty}^{\infty} dx \frac{1}{x^2} = L \cdot \frac{2}{L} = 1 \quad (4.1-3) \\ \langle \phi_n | \phi_n \rangle &= \left\langle \phi_n \left| \int dx |\psi\rangle \langle \psi| \right. \right\rangle \quad (4.1-4) \\ \langle \phi_n | \phi_n \rangle &= \int_{-\infty}^{\infty} dx \phi_n''(x) \phi_n(x) \quad (4.1-5) \quad \text{from } (4.1-2) \\ \langle \phi_n | \phi_n \rangle &= \frac{1}{2} \int_{-\infty}^{\infty} dx e^{-ikx} e^{ikx} = 1 \quad (4.1-6) \quad \text{from } (4.1-5) \end{aligned}$$





Challenges With AI and Deep Learning

The Hype and Reality of AI

CEO: “We will transform our business with AI”

Management: “Hire someone to give us AI”

Senior Data Scientist: “I should look into AI”

Junior Data Scientist: “I use TensorFlow all the time”

High School Kid: “I did my internship on Deep Learning”

Average Joe: “I want a self-driving car (and keep my job)”

Stanford Professors:

“focus on interpretability, start with simple models!”

H2O.ai Stanford Advisors

Sri/CEO

Boyd

Hastie

Tibshirani



me

Which Open-Source AI Platform to Use?



Caffe



H₂O.ai



H₂O.ai



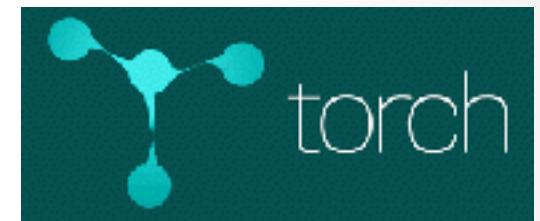
NVIDIA DIGITS
Interactive Deep Learning GPU Training System



theano



Chainer



OpenAI



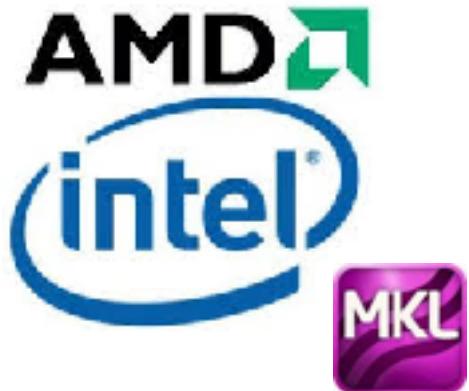
Which Programming Language To Use?



Which Hardware To Use?



Analog/Neuromorphic



Who Does the Work and on What Infrastructure?



Data Lake?
Micro-Services?



Cloud? Which? On Premise?

When is the Model Good Enough?

Crowd sourcing?



kaggle

Trust a Genius?



Internal Bake-Off?



Back to the Drawing Board!

What problem are you solving in the first place?

What problem should/could you be solving instead?

Do you need AI, Deep Learning or just a simple model?

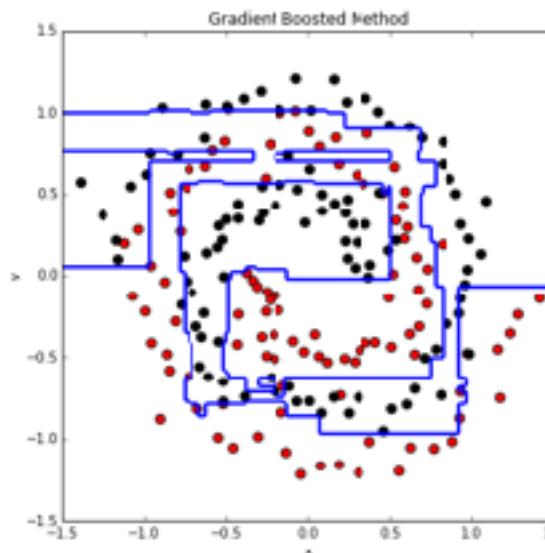
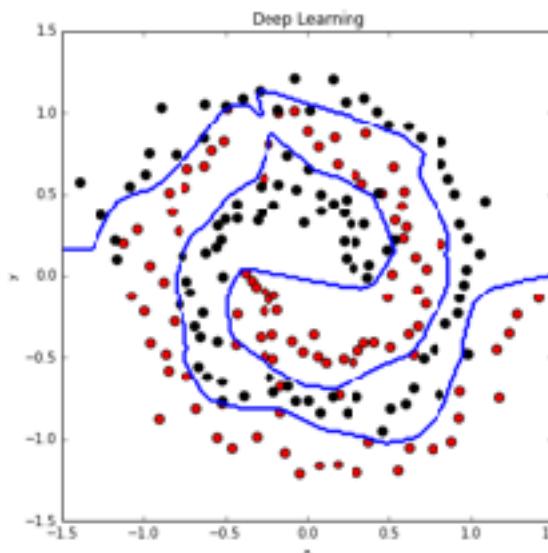
How can you characterize the model?

What can you learn from the model?

How can you improve the models? More, better data?

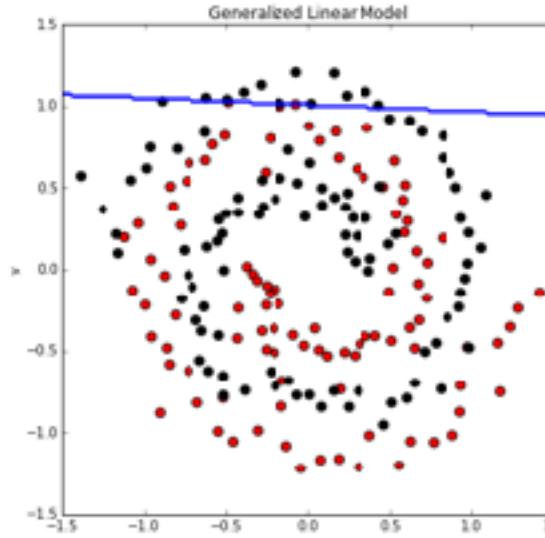
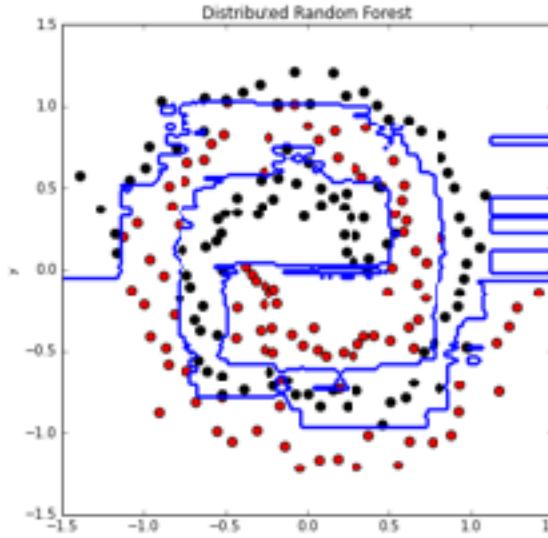
Do you need AI, Deep Learning or just a Simple Model?

Deep Learning



Gradient Boosting
Machine

Distributed
Random Forest



Generalized
Linear Modeling

Future Of AI: Or What's Left for Humans to Do?

