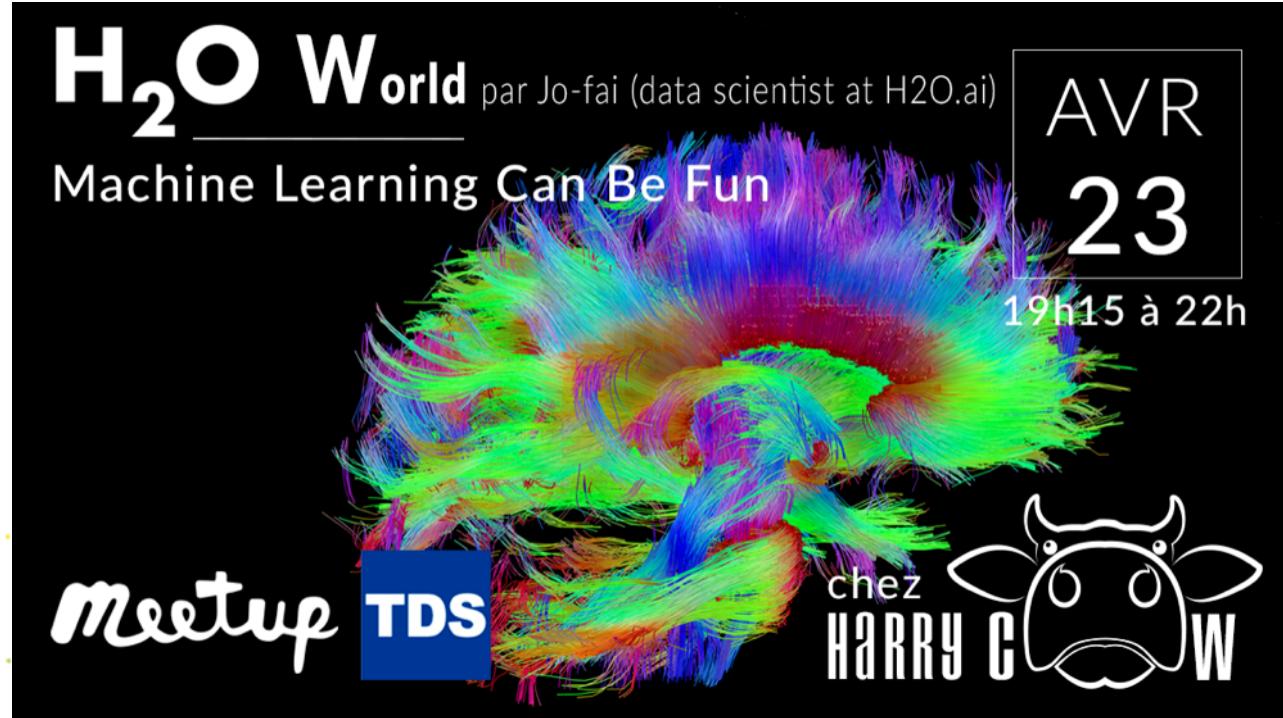


Scalable and Automatic Machine Learning with H₂O

Introduction, demos and a real-world use-case



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

Agenda

- Talk 1: Introduction to H₂O
 - Company and People
 - H₂O Open Source ML Platform
 - Demos
 - H₂O on Hadoop (320 Cores)
 - AutoML
 - Other News
- Talk 2: Moneyball
 - From a proof-of-concept project to a multimillion-dollar contract



Company Overview

Founded	2012, Series C in Nov, 2017
Products	<ul style="list-style-type: none">• Driverless AI – Automated Machine Learning• H₂O Open Source Machine Learning• Sparkling Water
Mission	Democratize AI. Do Good
Team	<p>~100 employees</p> <ul style="list-style-type: none">• Distributed Systems Engineers doing Machine Learning• World-class visualization designers
Offices	Mountain View, London, Prague



Our Mission: Make Machine Learning Accessible to Everyone



Complexity is your enemy. Any fool can make something complicated. It is hard to keep things simple.

— *Richard Branson* —

AZ QUOTES

Scientific Advisory Council



Dr. Trevor Hastie

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



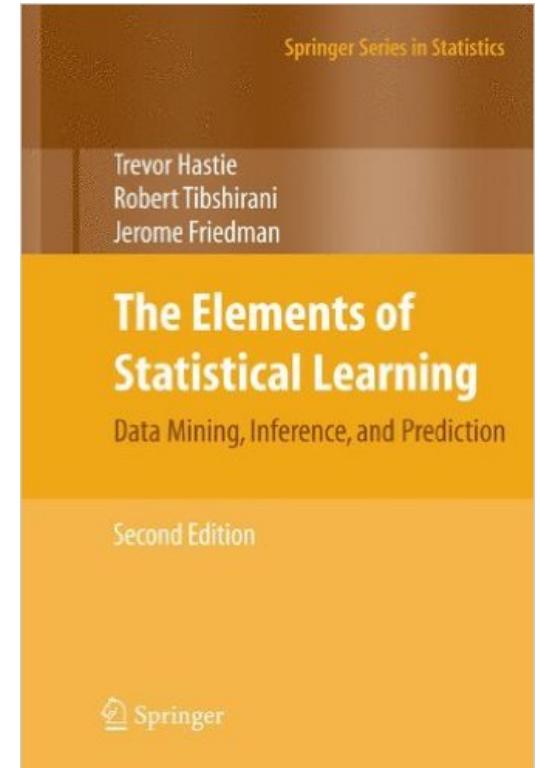
Dr. Robert Tibshirani

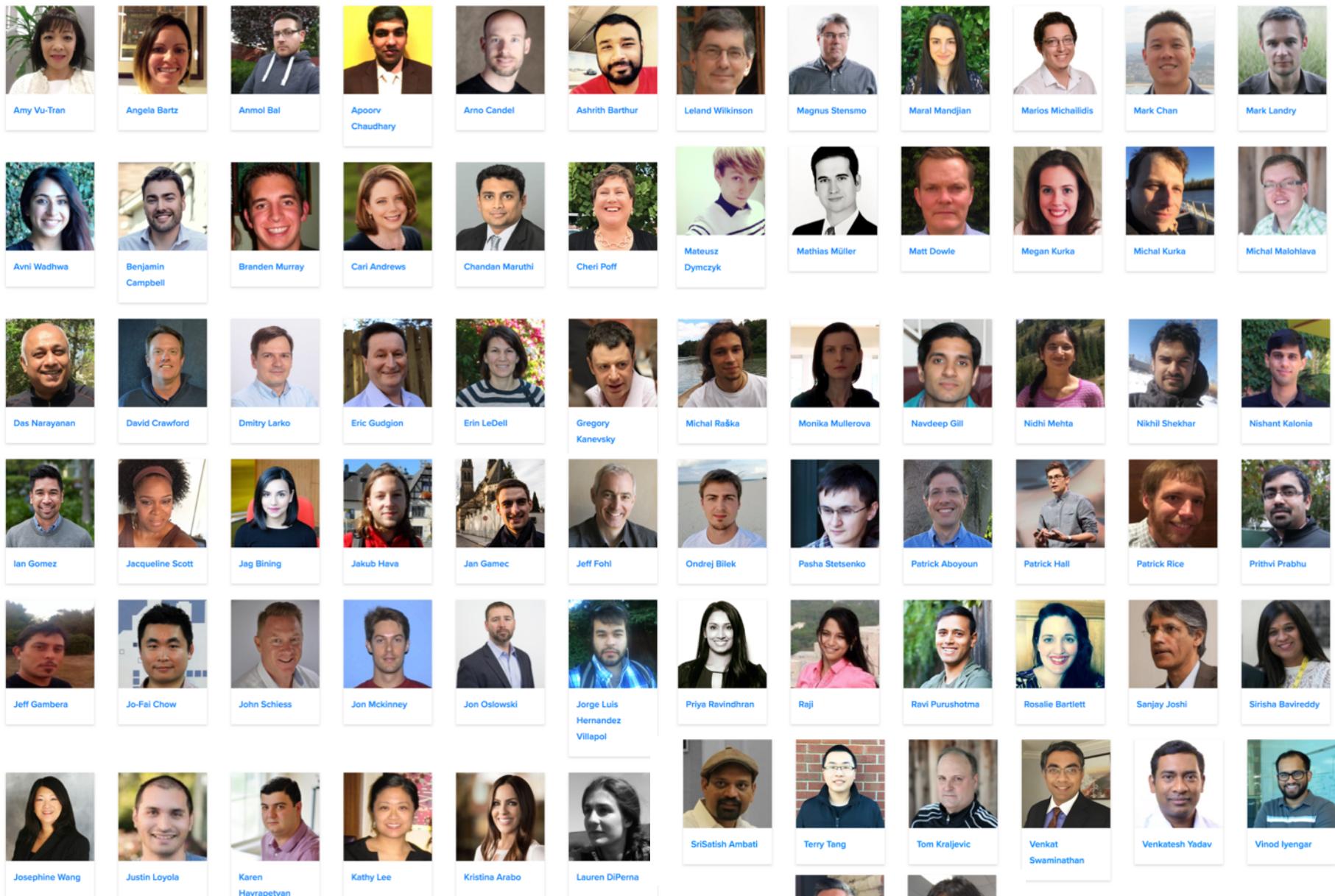
- Professor of Statistics and Health Research and Policy, Stanford University
- PhD in Statistics, Stanford University
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Author, *Regression Shrinkage and Selection via the Lasso*
- Co-author, *An Introduction to the Bootstrap*



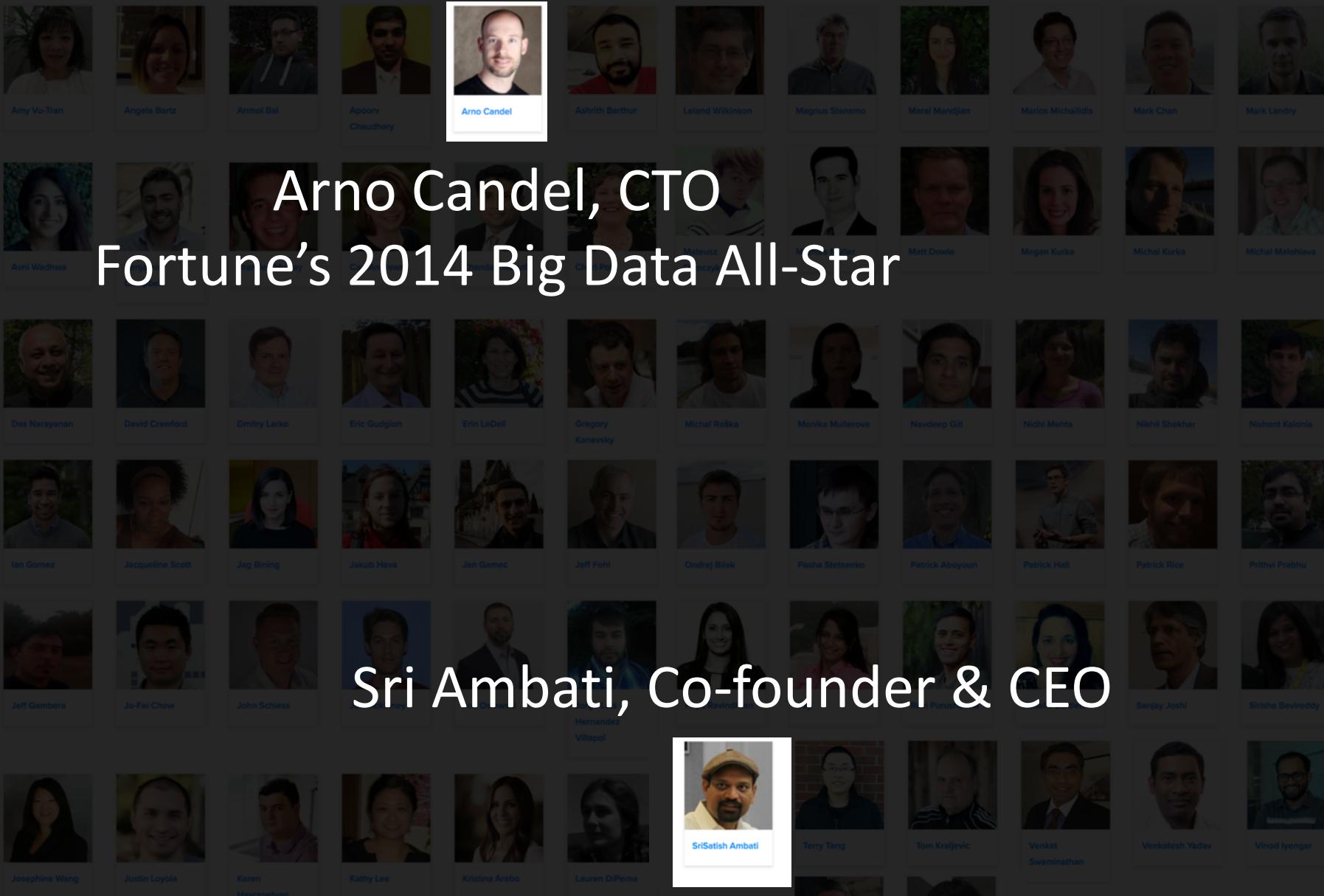
Dr. Steven Boyd

- Professor of Electrical Engineering and Computer Science, Stanford University
- PhD in Electrical Engineering and Computer Science, UC Berkeley
- Co-author, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*
- Co-author, *Linear Matrix Inequalities in System and Control Theory*
- Co-author, *Convex Optimization*



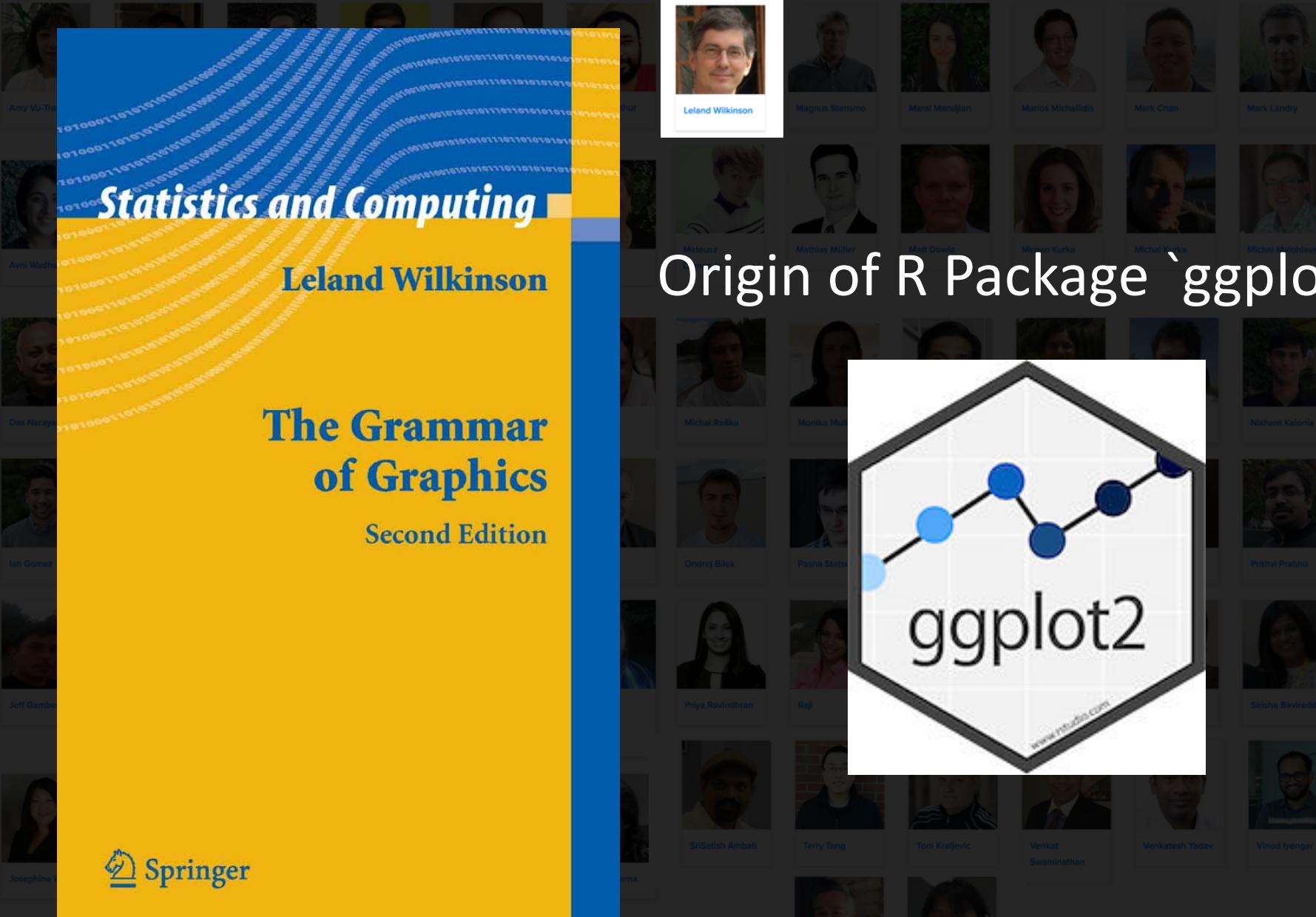


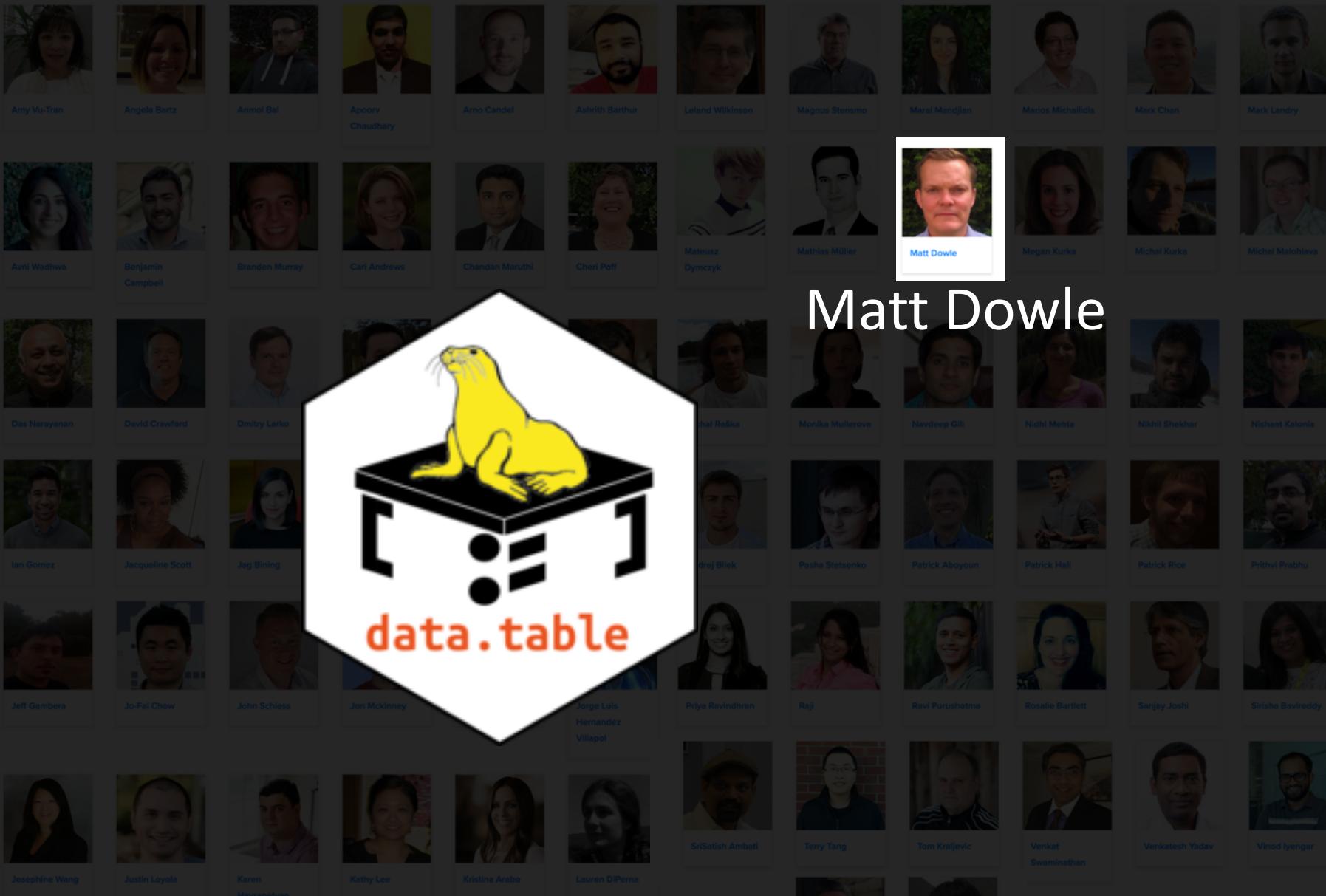
H₂O Team



H₂O Team

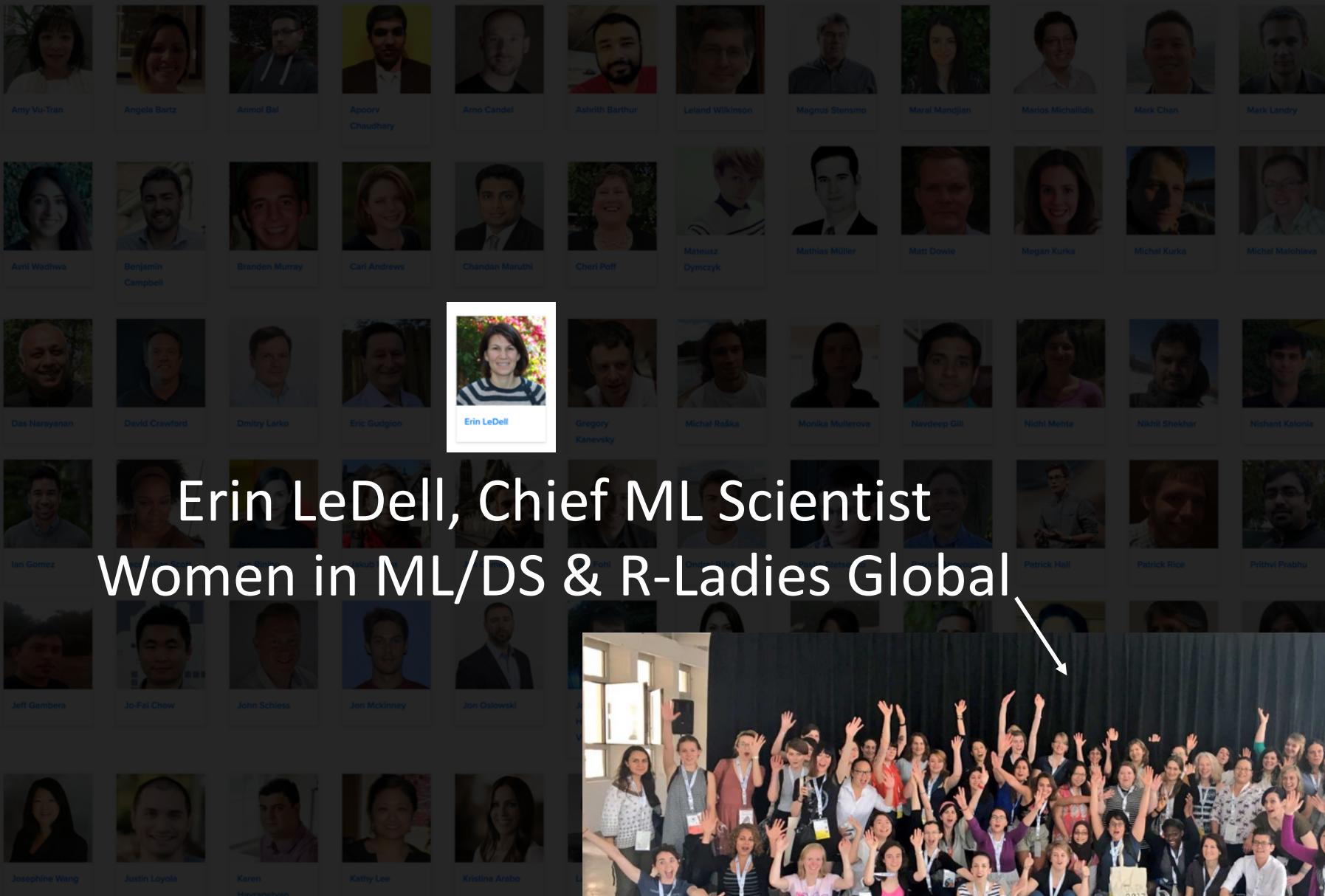
H₂O Team





Matt Dowle

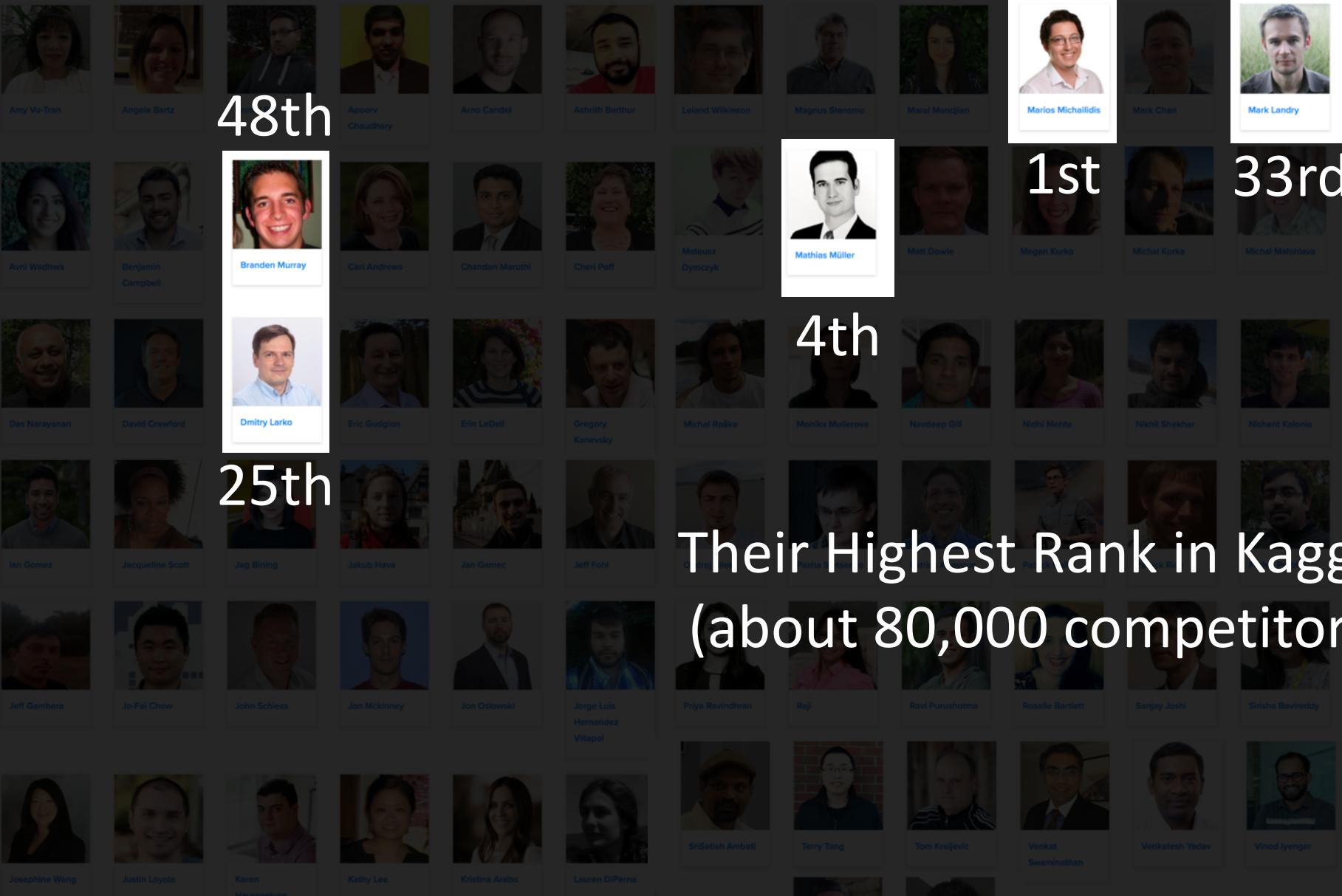
H₂O Team



Erin LeDell, Chief ML Scientist
Women in ML/DS & R-Ladies Global



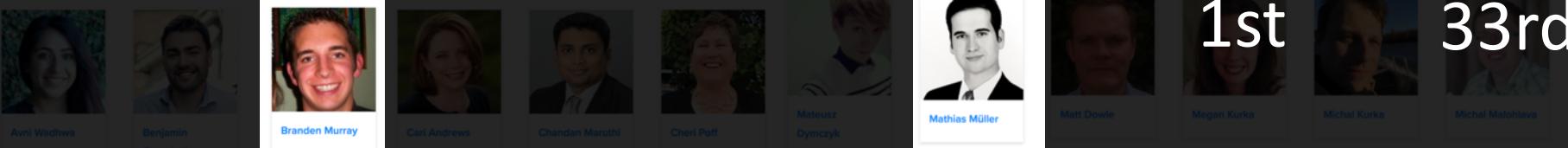
H₂O Team



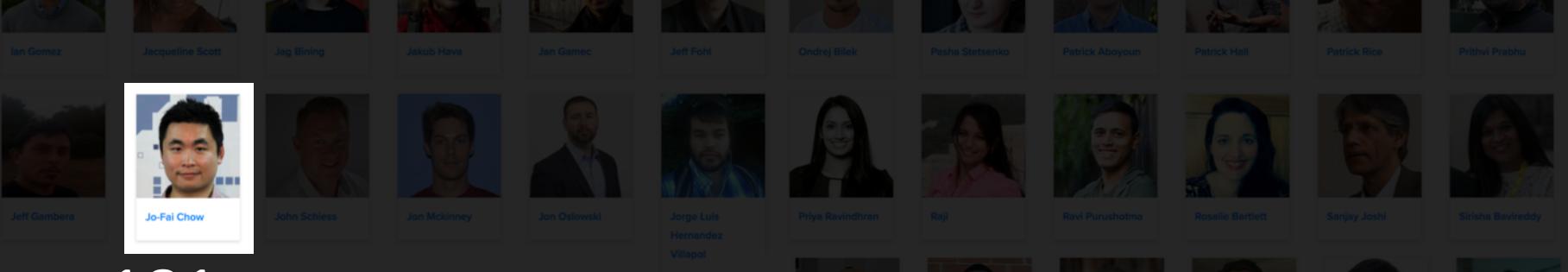
H₂O Team



48th



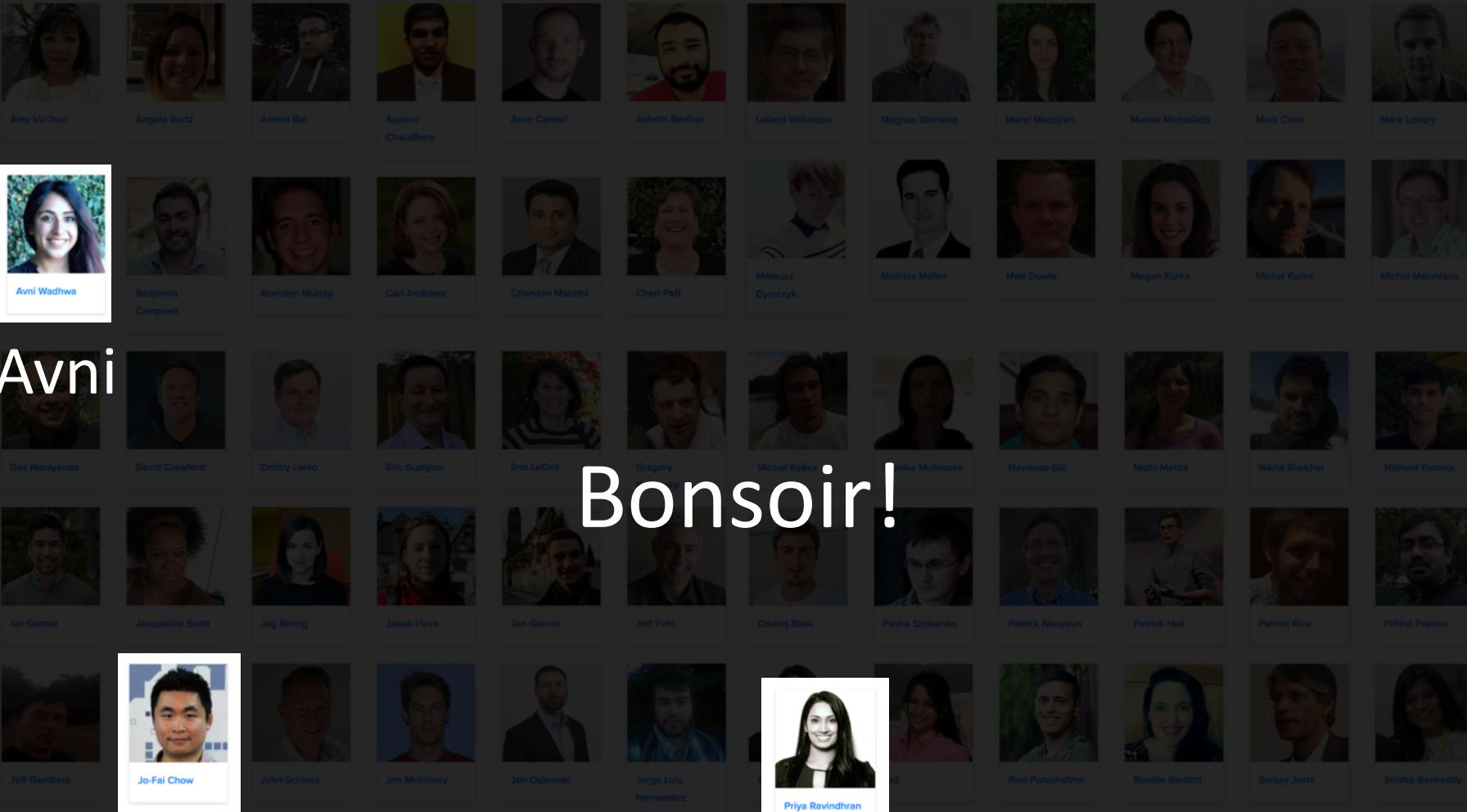
25th



181st

Trying to get closer to them at some point ...

H₂O Team



Avni
Bonsoir!



H₂O Team

H₂O Team

H₂O.ai

Feb 2016 - Present



H₂O Team in UK

Joe's Roles at H₂O.ai



- Data Scientist / Sales Engineer / Speaker / Meetup Organiser / Community Evangelist (on paper)
- Unofficial Photographer of H₂O.ai SWAG (the travelling data scientist)
- H₂O.ai SWAG EMEA Distributor (please help yourself)

Reminder: #360Selfie

Joe's Real Job at H₂O.ai

Jo-fai (Joe) Chow
@matlabulous

Another #FullHouse @h2oai #LondonAI
#meetup tonight. Thanks @MSFTRreactor for
the amazing venue and food! #OpenSource
#Community #MVPBuzz
#AroundTheWorldWithH2Oai #360Selfie 🇬🇧
cc our guest speakers @SKREDDY99
@cheukting_ho & Josh Warwick



7:15 PM - 12 Mar 2018 from London, England

Jo-fai (Joe) Chow
@matlabulous

Awesome #KNIMESummit2018
#KNIMESpringSummit in #Berlin. @knime
@Kurioos Marten here is our #360Selfie cc
@h2oai #AroundTheWorldWithH2Oai 🇩🇪
#OpenSource #MachineLearning
#Community 💪



1:54 PM - 7 Mar 2018 from Hotel Berlin

Jo-fai (Joe) Chow
@matlabulous

Thanks @ingnl for hosting @h2oai #meetup
in #Amsterdam last week. Tremendous
turnout and great discussions.
#AroundTheWorldWithH2Oai #360Selfie 🇳🇱
cc @fishnets88



7:15 AM - 26 Feb 2018 from Amsterdam, The Netherlands

H₂O Products



In-Memory, Distributed
Machine Learning Algorithms
with H2O Flow GUI



H2O AI Open Source Engine
Integration with Spark



Lightning Fast machine
learning on GPUs

DRIVERLESSAI

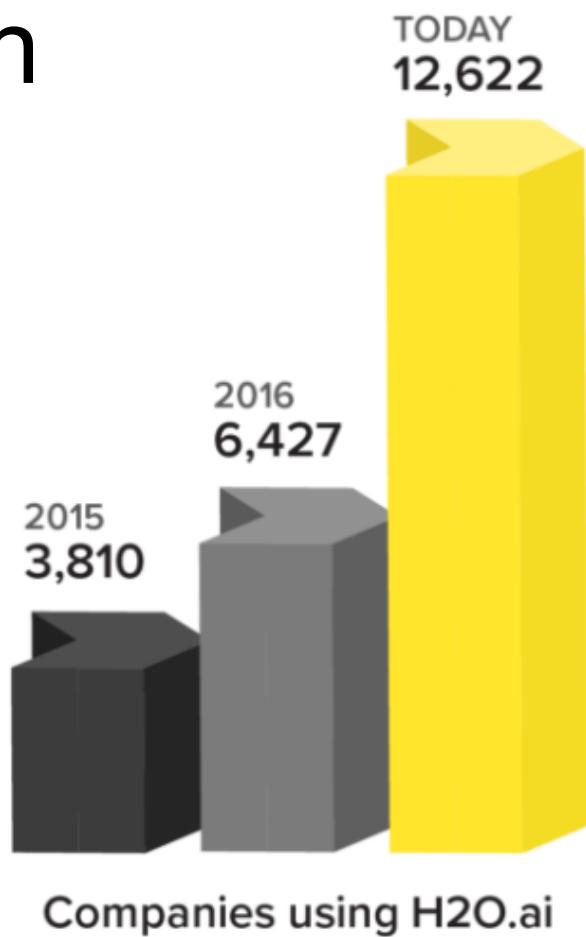
Automatic feature
engineering, machine
learning and interpretability

Steam

Secure multi-tenant H2O clusters



Worldwide Community Adoption



* DATA FROM GOOGLE ANALYTICS EMBEDDED IN THE END USER PRODUCT

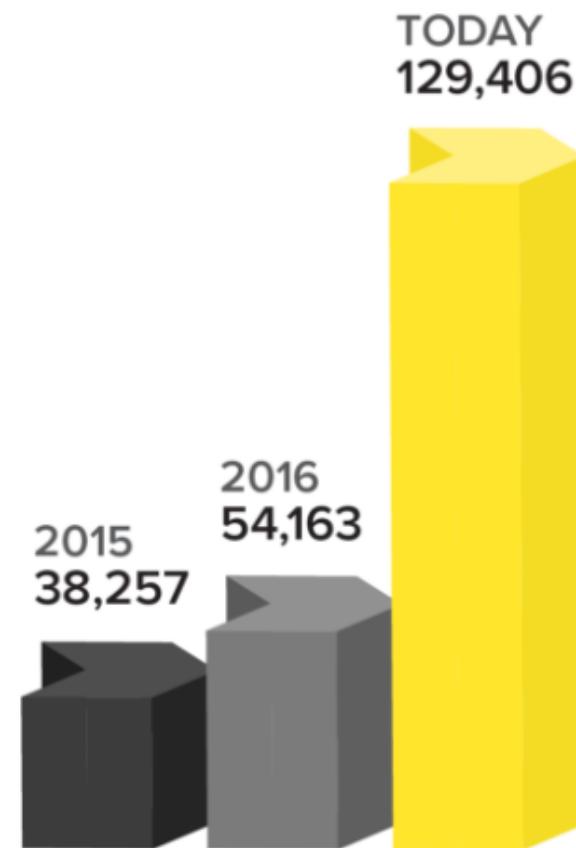
222 OF FORTUNE
THE 500



8 OF TOP 10
BANKS

7 OF TOP 10
INSURANCE COMPANIES

4 OF TOP 10
HEALTHCARE COMPANIES



H2O.ai **H₂O.ai**

Gartner names H2O as Leader with the most completeness of vision

- H2O.ai recognized as a **technology leader with most completeness of vision**
- H2O.ai was recognized for the mindshare, partner network and status as a **quasi-industry standard** for machine learning and AI.
- **H2O customers gave the highest overall score** among all the vendors for sales relationship and account management, customer support (onboarding, troubleshooting, etc.) and overall service and support.

Figure 1. Magic Quadrant for Data Science and Machine-Learning Platforms



Source: Gartner (February 2018)

As of January 2018

© Gartner, Inc

Platforms with H₂O integration



srisatish
@srisatish

Following

Replying to @BobMuenchen @knime @h2oai

@KNIME gained the ability to run @H2O.ai algorithms, so these two may be viewed as complementary, not competitors
#Ecosystem #OpenSource

3:32 PM - 2 Mar 2018



H₂O + KNIME Talk
at KNIME Summit
Mar 2017

1:54 PM - 7 Mar 2018 from Hotel Berlin

Figure 1. Magic Quadrant for Data Science and Machine-Learning Platforms



Source: Gartner (February 2018)

© Gartner, Inc

H₂O.ai

H2O.ai Solution Leadership Across Verticals



Community Expansion



meetup

88,286
members

43
interested

52
Meetups

48
cities
18
countries

Find out more: www.h2o.ai/community/



London Artificial Intelligence & Deep Learning

PRO

H2O Artificial Intelligence and Machine Learning -
39 groups

Location

London, United Kingdom

Members

5,869



Organizers

Ian Gomez and 1 other

[Schedule](#)[...](#)[Our group](#)[Meetups](#)[Members](#)[Photos](#)[Discussions](#)[More](#)

What we're about

🕒 Public Group

Welcome to the group. We're excited to bring you the latest happenings in AI, Machine Learning, Deep Learning, Data Science and Big Data.

Who are we? We're H2O.ai (<https://www.h2o.ai/>), creators of the world's leading open source deep learning and machine learning platform, used by more than 90,000 data scientists and 9,000 organizations around the world.

Past Meetups (10)

[See all](#)

12
MAR

Mon, Mar 12, 2018, 6:00 PM
#LondonAI Meetup with SK Reddy, Josh Warwick &...



You + 354 went

[Copy Meetup](#)

H₂O Products

H₂O.ai

In-Memory, Distributed
Machine Learning Algorithms
with H2O Flow GUI

Spark + H₂O
SPARKLING
WATER

H2O AI Open Source Engine
Integration with Spark

H₂O4GPU

Lightning Fast machine
learning on GPUs

DRIVERLESSAI

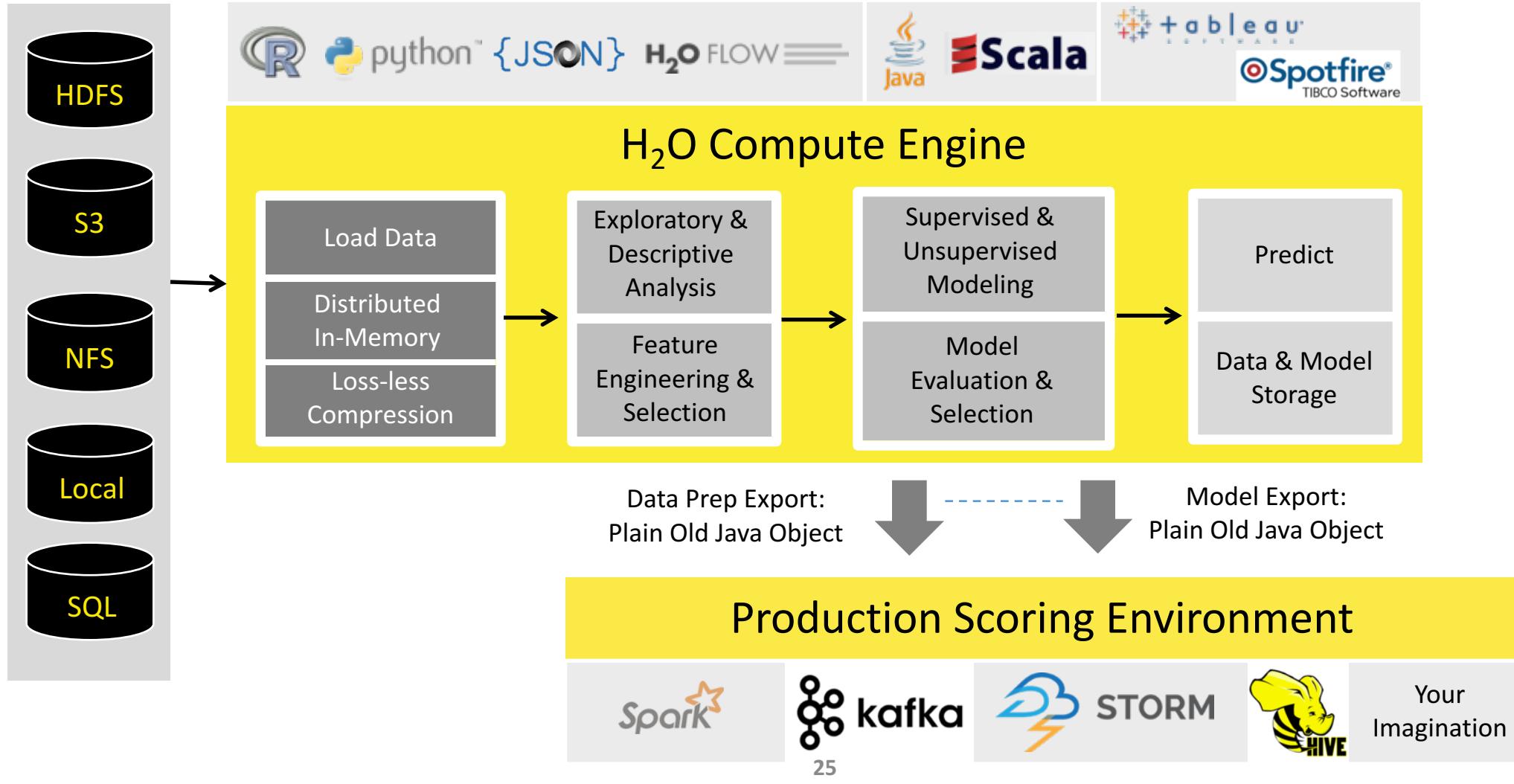
Automatic feature
engineering, machine
learning and interpretability

Steam

Secure multi-tenant H2O clusters

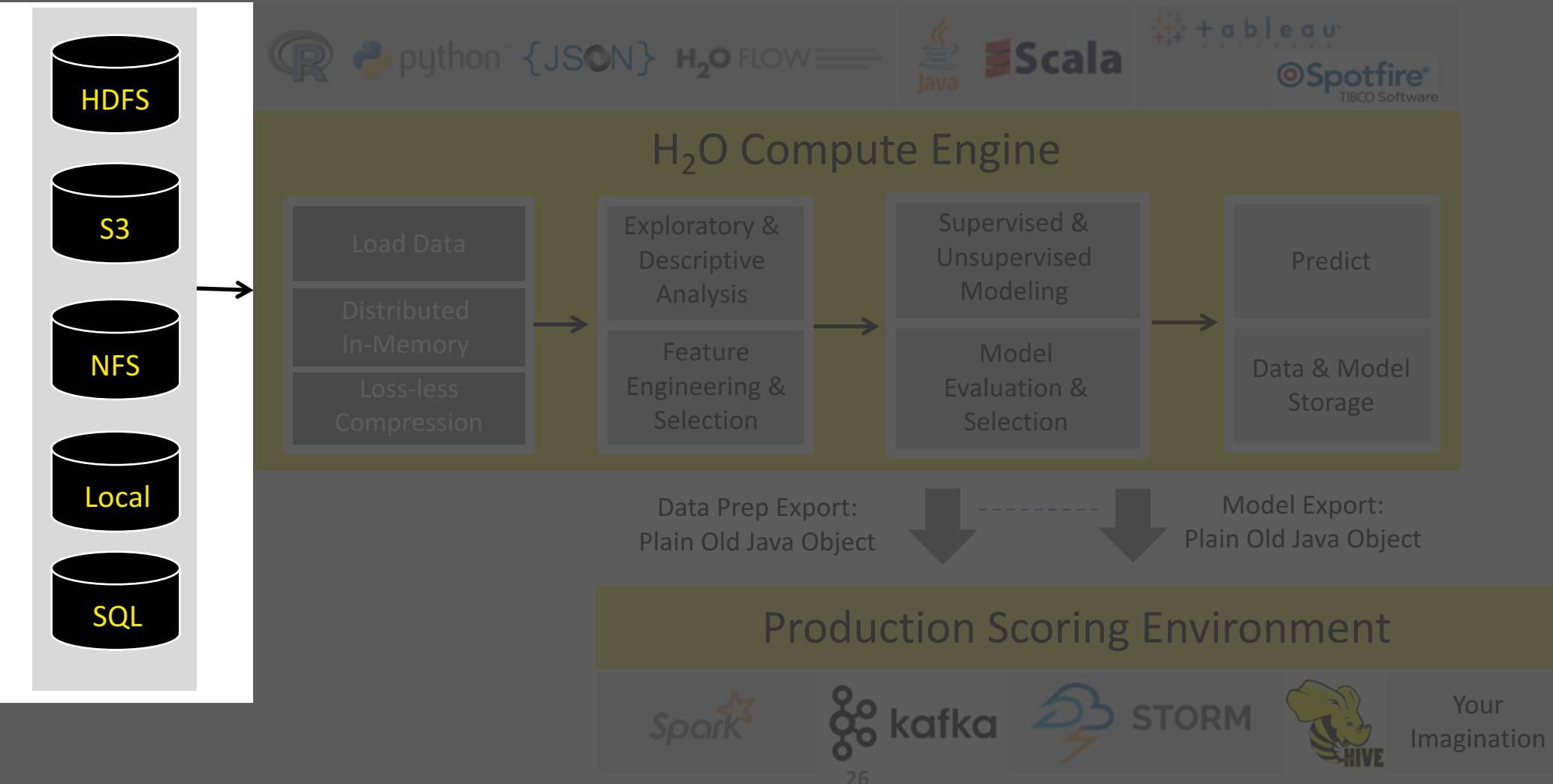
H₂O.ai

High Level Architecture

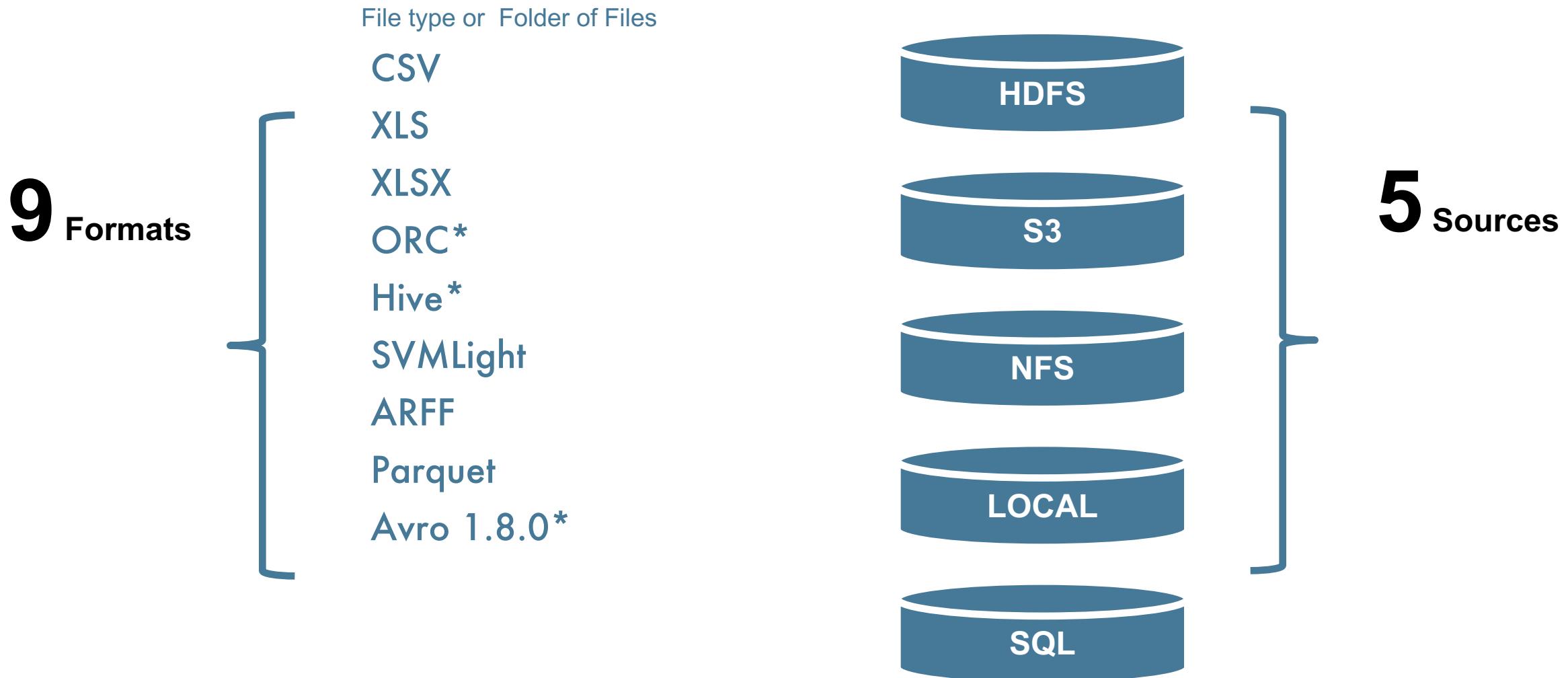


High Level Architecture

Import Data from
Multiple Sources



Supported Formats & Data Sources



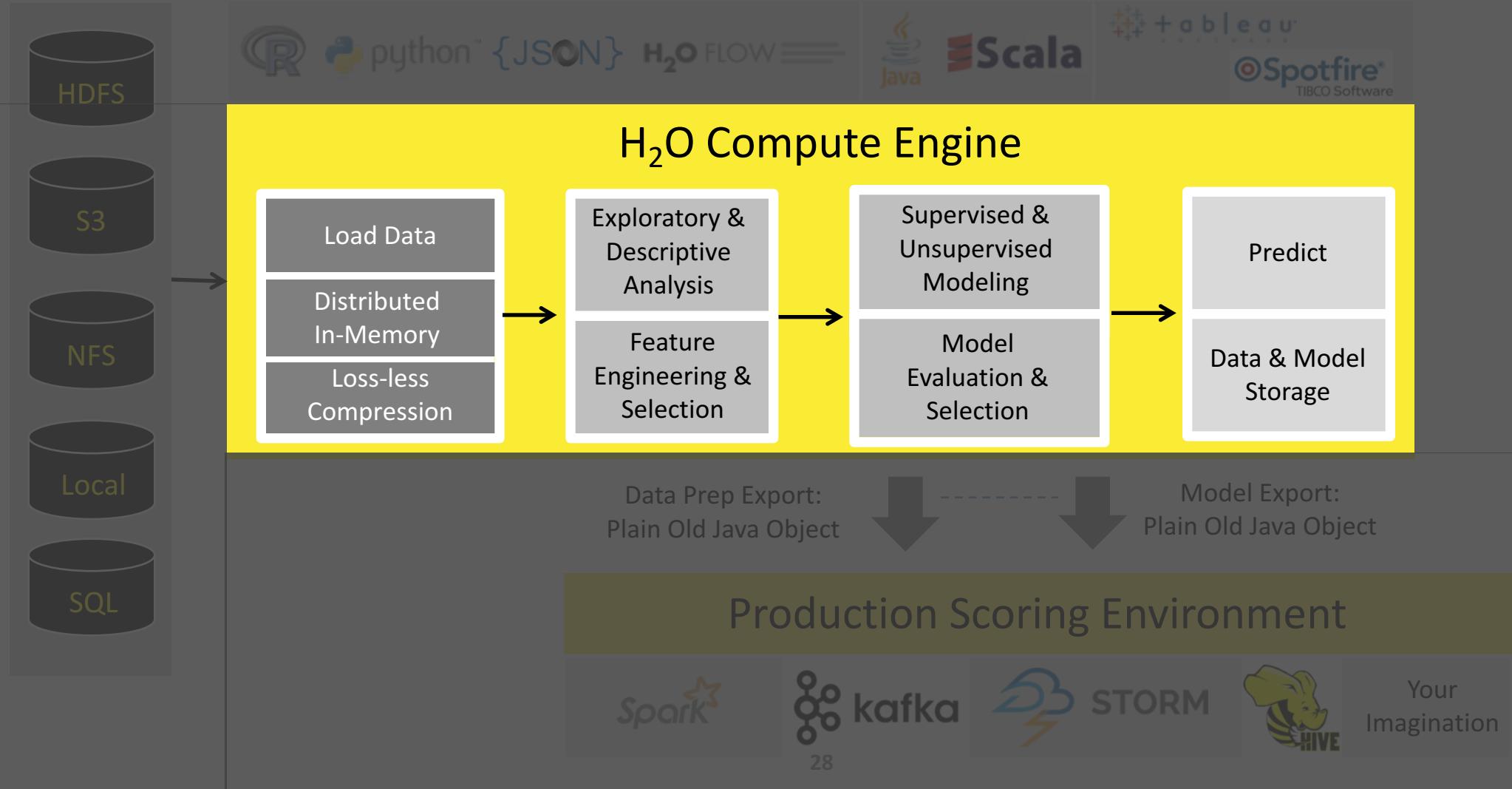
* 1. only if H2O is running as a Hadoop job

* 2. Hive files that are saved in ORC format

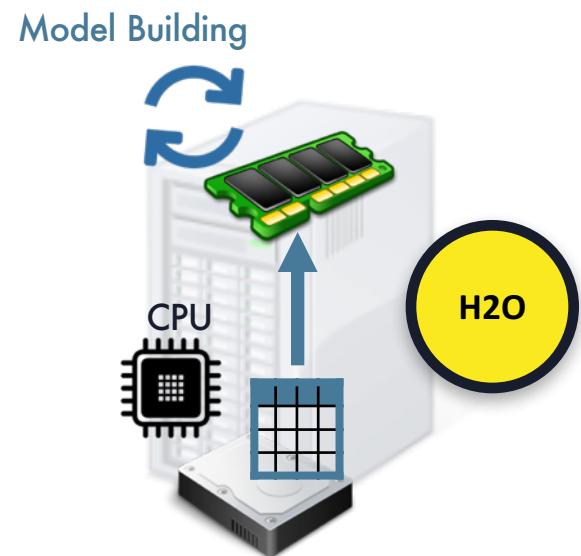
* 3. without multi-file parsing or column type modification

High Level Architecture

Fast, Scalable & Distributed
Compute Engine Written in
Java



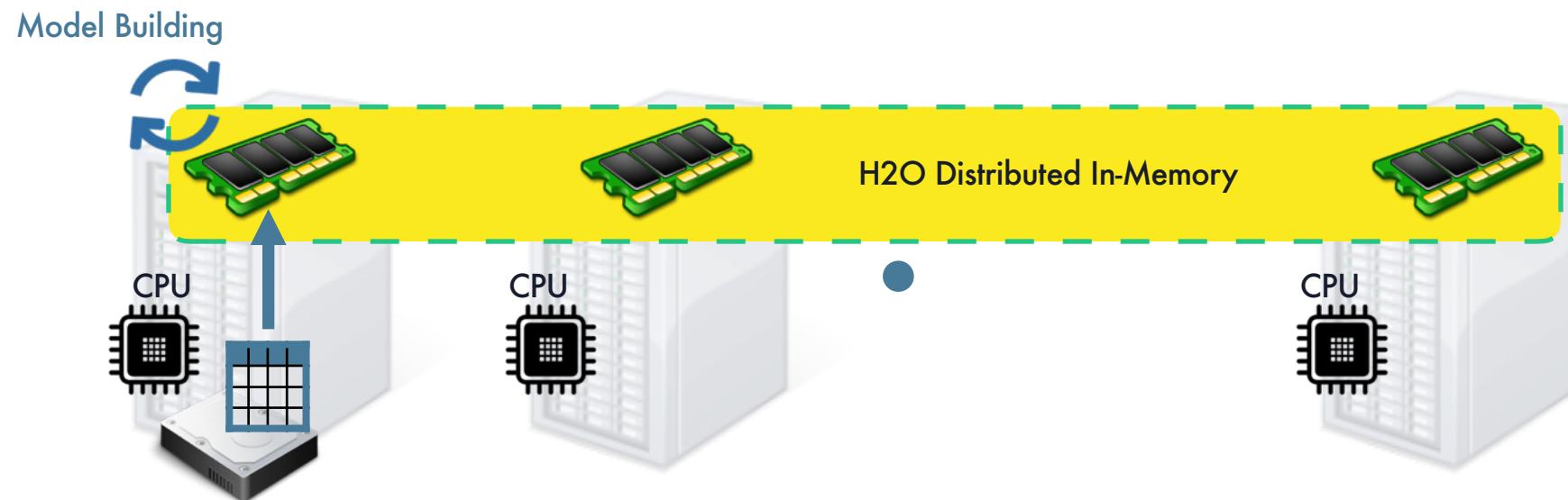
H₂O Core



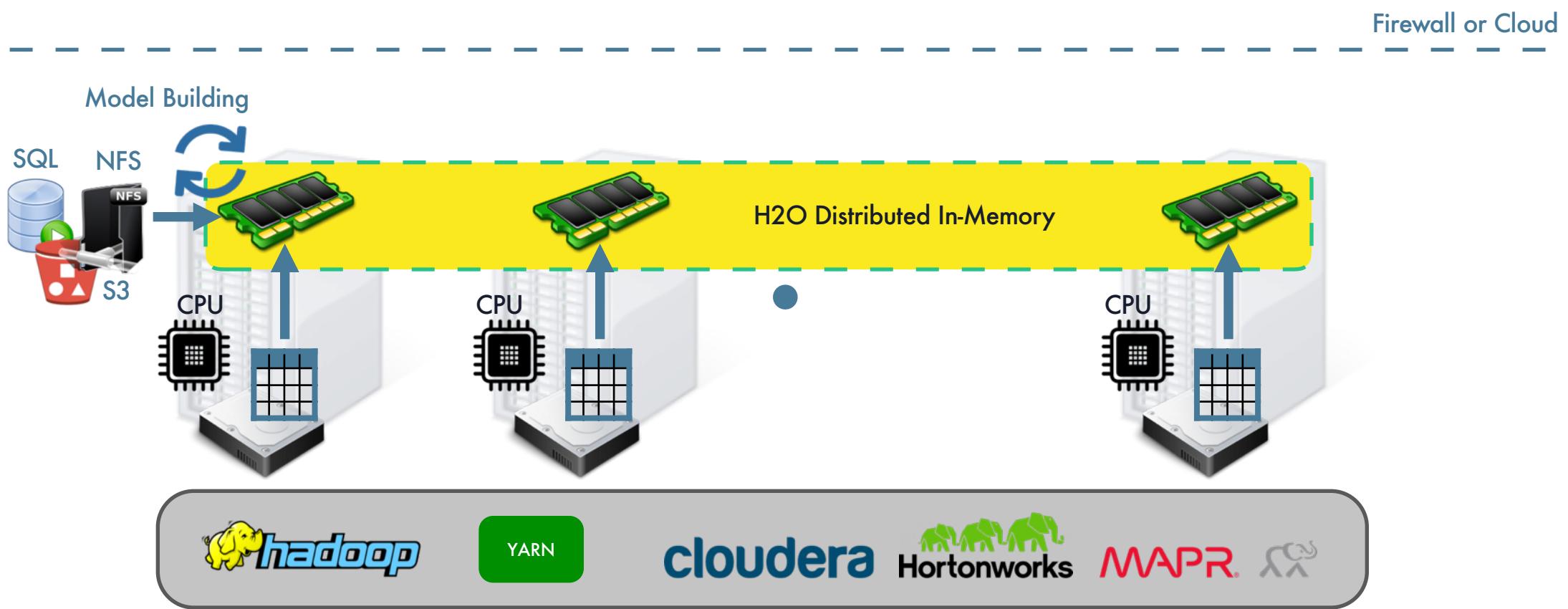
H₂O Core



H₂O Core

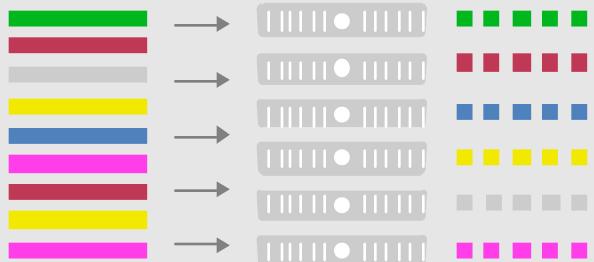


H₂O Core

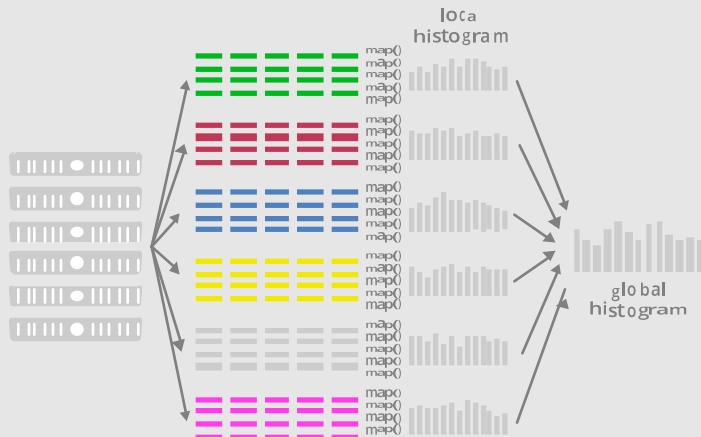


Distributed Algorithms

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-3 Algorithms Overview

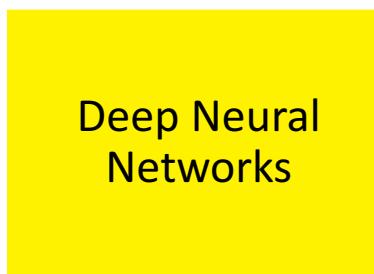
Supervised Learning



- **Generalized Linear Models:** Binomial, Gaussian, Gamma, Poisson and Tweedie
- **Naïve Bayes**



- **Distributed Random Forest:** Classification or regression models
- **Gradient Boosting Machine:** Produces an ensemble of decision trees with increasing refined approximations

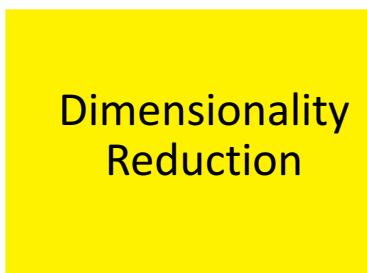


- **Deep learning:** Create multi-layer feed forward neural networks starting with an input layer followed by multiple layers of nonlinear transformations

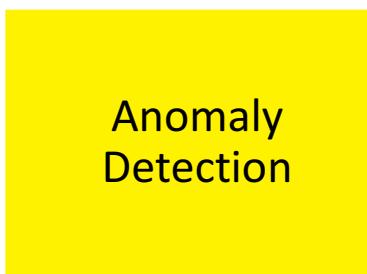
Unsupervised Learning



- **K-means:** Partitions observations into k clusters/groups of the same spatial size. Automatically detect optimal k

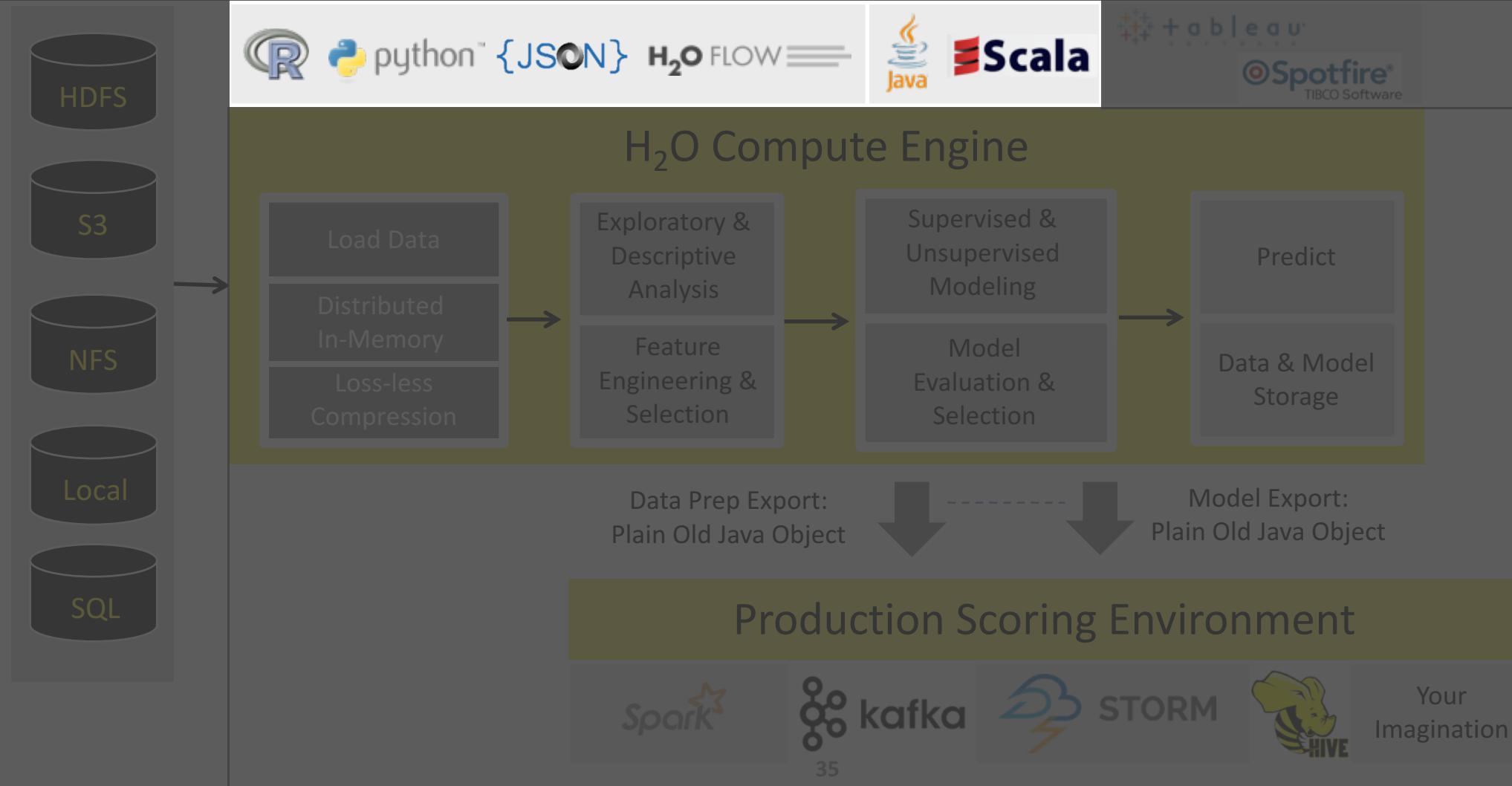


- **Principal Component Analysis:** Linearly transforms correlated variables to independent components
- **Generalized Low Rank Models:** extend the idea of PCA to handle arbitrary data consisting of numerical, Boolean, categorical, and missing data



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

High Level Architecture



H₂O Flow (Web) – First Demo

The screenshot shows the H2O Flow (Web) interface running in a browser window. The title bar reads "H2O Flow" and the address bar shows "localhost:54321/flow/index.html". The main menu bar includes "Model", "Score", "Admin", and "Help". The "Model" menu is currently open, displaying a list of modeling routines:

- Aggregator...
- Deep Learning...
- Distributed Random Forest...
- Gradient Boosting Machine...
- Generalized Linear Modeling...
- Generalized Low Rank Modeling...
- K-means...
- Naive Bayes...
- Principal Components Analysis...
- Stacked Ensemble...
- Word2Vec...
- XGBoost...

Below the menu, there's a sidebar titled "Assistance" listing various H2O routines with their descriptions:

Routine	Description
importFiles	Import file(s) into H2O
getFrames	Get a list of frames in H2O
splitFrame	Split a frame into two or more
mergeFrames	Merge two frames into one
getModels	Get a list of models in H2O
getGrids	Get a list of grid search results
getPredictions	Get a list of predictions in H2O
getJobs	Get a list of jobs running in H2O
buildModel	Build a model
runAutoML	Automatically train and tune
importModel	Import a saved model
predict	Make a prediction

The right side of the interface features a "Help" section with links to "Quickstart Videos" and "example Flows", and a "GENERAL" section with a list of links related to Flow and H2O.

Using H₂O with R and Python – Second Demo

The screenshot shows the RStudio Source Editor window with the file `credit_card_example.R` open. The code is an R script for a credit card example, demonstrating the use of the `h2o` package. It includes importing datasets from S3, defining features and target, training a GBM model, and printing the leaderboard. The code is well-structured with comments explaining each step.

```
1 # Credit Card Example
2
3 # Datasets:
4 # https://s3.amazonaws.com/h2o-training/credit_card/credit_card_train.csv
5 # https://s3.amazonaws.com/h2o-training/credit_card/credit_card_test.csv
6
7 # Start and connect to a local H2O cluster
8 library(h2o)
9 h2o.init(nthreads = -1)
10
11 # Import datasets from s3
12 df_train = h2o.importFile("https://s3.amazonaws.com/h2o-training/credit_card/credit_card_train.csv")
13 df_test = h2o.importFile("https://s3.amazonaws.com/h2o-training/credit_card/credit_card_test.csv")
14
15 # Look at datasets
16 summary(df_train)
17 summary(df_test)
18
19 # Define features and target
20 features = colnames(df_test)
21 target = "DEFAULT_PAYMENT_NEXT_MONTH"
22
23 # Train a GBM model
24 model_gbm = h2o.gbm(x = features,
25                      y = target,
26                      training_frame = df_train,
27                      seed = 1234)
28 print(model_gbm)
29
30 # Use GBM model for making predictions
31 yhat_test = h2o.predict(model_gbm, newdata = df_test)
32 head(yhat_test)
33
34 # (Extra) Use H2O's AutoML
35 aml = h2o.automl(x = features,
36                   y = target,
37                   training_frame = df_train,
38                   max_runtime_secs = 60,
39                   seed = 1234)
40
41 # Print leaderboard
42 print(aml@leaderboard)
43
44 # Use best model for making predictions
45 best_model = aml@leaderboard[[1]]
46 yhat_test = h2o.predict(best_model, newdata = df_test)
47 head(yhat_test)
48
49
```

The screenshot shows a Jupyter Notebook interface with the notebook `credit_card_example.ipynb` open. The notebook contains Python code for connecting to a local H2O cluster, importing datasets, and summarizing them. The output of the first cell shows the Java version, server startup logs, and cluster statistics. Subsequent cells show the import of datasets and their summaries. A table at the bottom displays the summary statistics for the dataset columns.

```
In [2]: # Start and connect to a local H2O cluster
import h2o
h2o.init(nthreads = -1)

Checking whether there is an H2O instance running at http://localhost:54321.... not found.
Attempting to start a local H2O server...
Java Version: java version "1.8.0_72"; Java(TM) SE Runtime Environment (build 1.8.0_72-b15); Java HotSpot(TM) 64-Bit Server VM (build 25.72-b15, mixed mode)
Starting server from /Users/jofaichow/anaconda/lib/python2.7/site-packages/h2o/backend/bin/h2o.jar
Ice root: /var/folders/4z/p7yt_4n4fjjlyg64qhfwbw000gn/T/tmpPdP3Av
JVM stdout: /var/folders/4z/p7yt_4n4fjjlyg64qhfwbw000gn/T/tmpPdP3Av/h2o_jofaichow_started_from_python.out
JVM stderr: /var/folders/4z/p7yt_4n4fjjlyg64qhfwbw000gn/T/tmpPdP3Av/h2o_jofaichow_started_from_python.err
Server is running at http://127.0.0.1:54321
Connecting to H2O server at http://127.0.0.1:54321... successful.

H2O cluster uptime: 02 secs
H2O cluster version: 3.13.0.3981
H2O cluster version age: 29 days
H2O cluster name: H2O_from_python_jofaichow_id7qa
H2O cluster total nodes: 1

In [3]: # Import datasets from s3
df_train = h2o.import_file("https://s3.amazonaws.com/h2o-training/credit_card/credit_card_train.csv")
df_test = h2o.import_file("https://s3.amazonaws.com/h2o-training/credit_card/credit_card_test.csv")

Parse progress: |██████████| 100%
Parse progress: |██████████| 100%

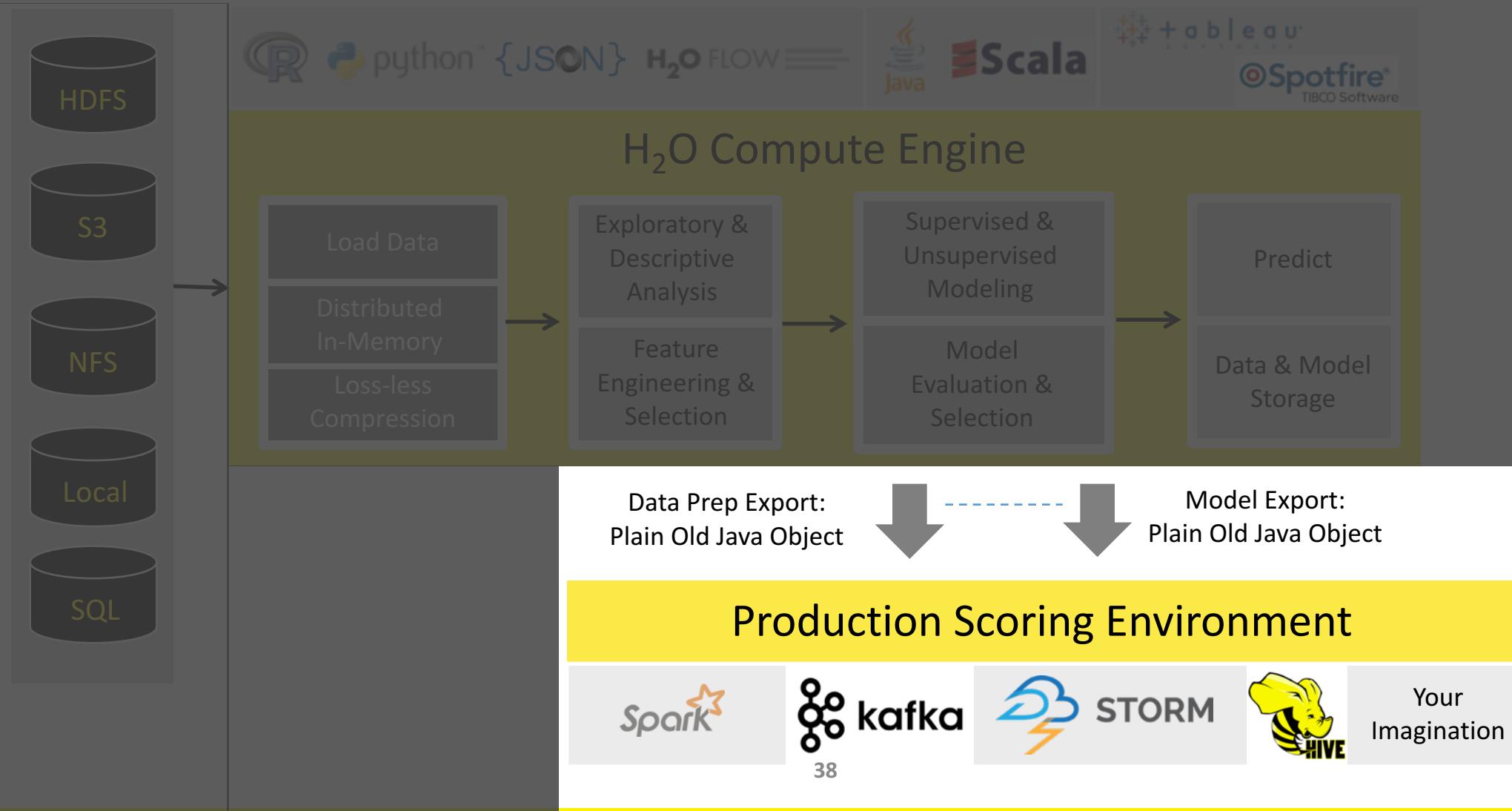
In [4]: # Look at datasets
df_train.summary()
df_test.summary()


```

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4
type	int	enum	int	int	int	int	int	int	int
mins	10000.0		0.0	0.0	21.0	-2.0	-2.0	-2.0	-2.0
mean	165471.466667		1.85	1.55578703704	35.4053240741	-0.00523148148148	-0.122361111111	-0.15537037037	-0.210601
maxs	1000000.0		6.0	3.0	79.0	8.0	8.0	8.0	8.0
sigma	128853.314839		0.779559696278	0.522505078476	9.27675421641	1.12668964211	1.20086854503	1.20727030901	1.172176
zeros	0		9	37	0	10563	11284	11309	11905
missing	0		0	0	0	0	0	0	0

High Level Architecture

Export Standalone Models
for Production



H₂O Documentation

[Getting Started & User Guides](#) | [Q & A](#) | [Algorithms](#) | [Languages](#) | [Tutorials, Examples, & Presentations](#) | [API & Developer Docs](#) | [For the Enterprise](#)

Getting Started & User Guides

 Open Source |  Commercial

H₂O

What is H₂O?
H₂O User Guide (Main docs)
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 User Guide 2.3 2.2 2.1
Sparkling Water Booklet
RSparkling Readme
PySparkling User Guide 2.3 2.2 2.1
Recent Changes 2.3 2.2 2.1
Open Source License (Apache V2)

Quick Start Video - Scala

[Download Sparkling Water](#)

Driverless AI

What is Driverless AI?
Driverless AI User Guide [HTML](#) [PDF](#)
Recent Changes
Driverless AI Booklet
MLI with Driverless AI Booklet

Quick Start Video - Downloading Driverless AI
Quick Start Video - Launching an Experiment
Driverless AI Webinars

[Download Driverless AI](#)

H₂O4GPU (alpha)

H₂O4GPU Readme
Open Source License (Apache V2)

[Download H₂O4GPU](#)

URL: [docs.h2o.ai](#)

Demo: H_2O on a 320-Core Hadoop Cluster

(Web Interface)



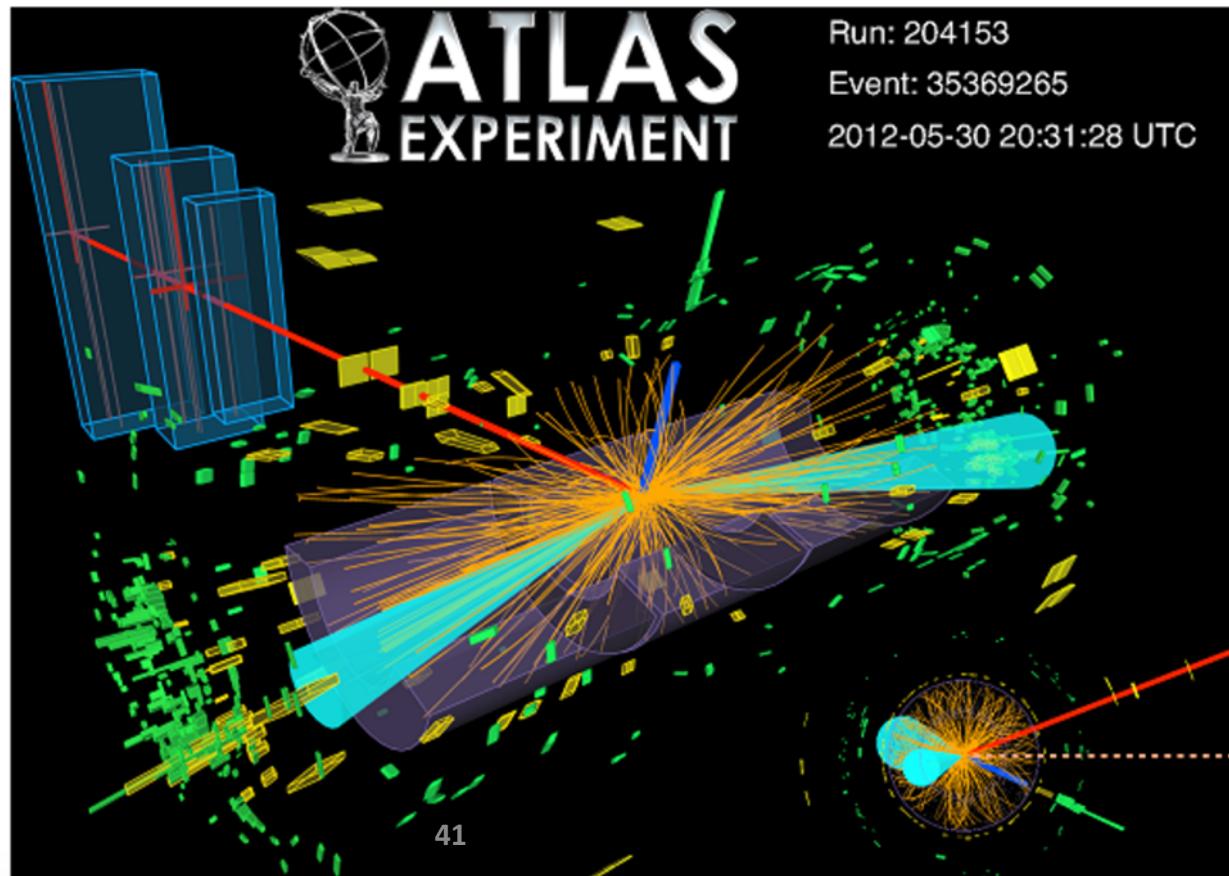
Higgs Boson Machine Learning Challenge

Use the ATLAS experiment to identify the Higgs boson

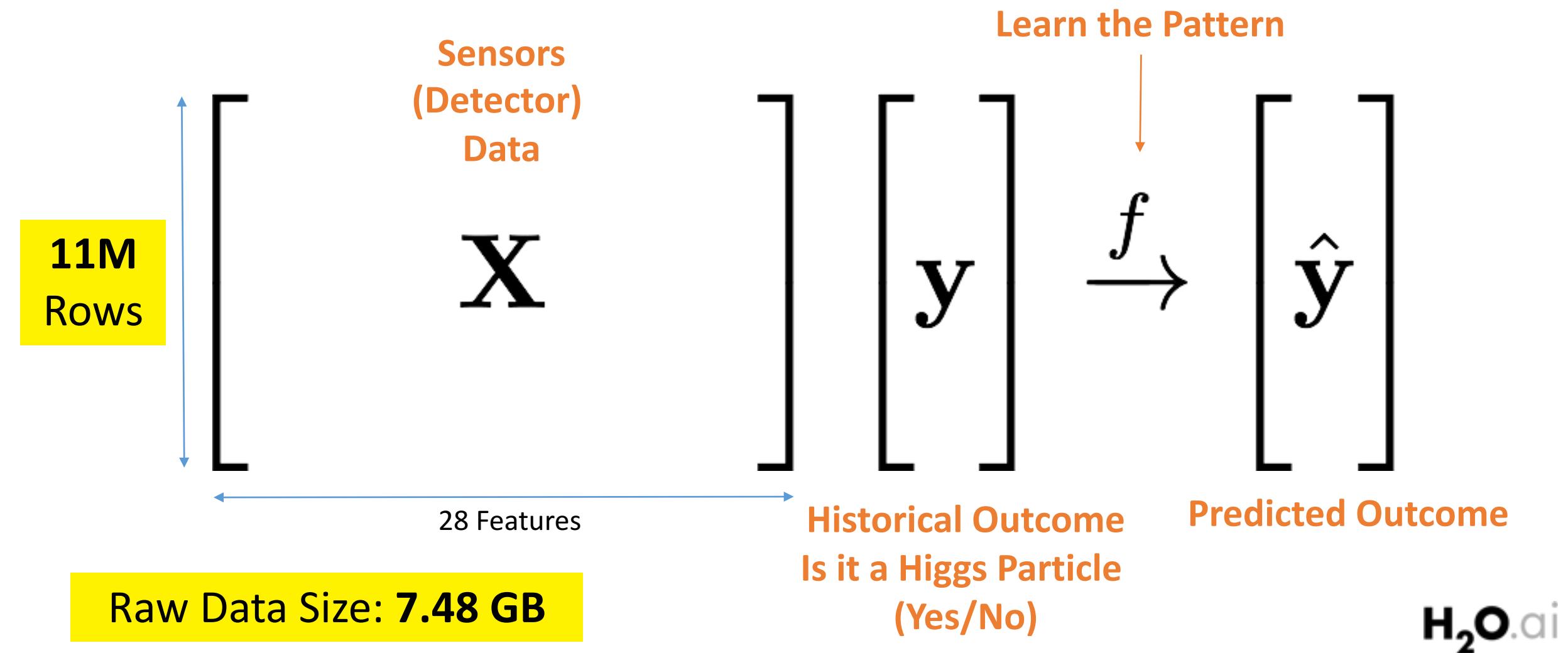
\$13,000 · 1,785 teams · 3 years ago

[Overview](#)[Data](#)[Discussion](#)[Leaderboard](#)[Rules](#)[Team](#)[My Submissions](#)[Late Submission](#)[Overview](#)

<https://www.kaggle.com/c/higgs-boson>

[Description](#)[Evaluation](#)[Prizes](#)[About The Sponsors](#)[Timeline](#)[Winners](#)

Learning from Higgs Boson Machine Data



11M Rows**Size (Raw): 7.48 GB****Compressed: 2.00 GB (\approx 27% of Raw)**

HIGGS.hex

Actions:

[View Data](#)[Split...](#)[Build Model...](#)[Predict](#)[Download](#)[Export](#)

Rows	Columns	Compressed Size
11000000	29	2GB

▼ COLUMN SUMMARIES

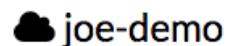
label	type	Missing	Zeros	+Inf	-Inf	min	max	mean	sigma	cardinality	Actions
C1	enum	0	5170877	0	0	0	1.0	0.5299	0.4991	2	Convert to numeric
C2	real	0	0	0	0	0.2747	12.0989	0.9915	0.5654	· ·	
C3	real	0	0	0	0	-2.4350	2.4349	-0.0	1.0088	· ·	
C4	real	0	0	0	0	-1.7425	1.7432	-0.0	1.0063	· ·	
C5	real	0	0	0	0	0.0002	15.3968	0.9985	0.6000	· ·	
C6	real	0	0	0	0	-1.7439	1.7433	0.0	1.0063	· ·	
C7	real	0	0	0	0	0.1375	9.9404	0.9909	0.4750	· ·	
C8	real	0	0	0	0	-2.9697	2.9697	-0.0	1.0093	· ·	
C9	real	0	0	0	0	-1.7412	1.7415	0.0	1.0059	· ·	
C10	real	0	5394611	0	0	0	2.1731	1.0	1.0278	· ·	
C11	real	0	0	0	0	0.1890	11.6471	0.9927	0.5000	· ·	
C12	real	0	0	0	0	-2.9131	2.9132	-0.0	1.0093	· ·	
C13	real	0	0	0	0	-1.7424	1.7432	-0.0	1.0062	· ·	
C14	real	0	5523912	0	0	0	2.2149	1.0	1.0494	· ·	
C15	real	0	0	0	0	0.2636	14.7090	0.9923	0.4877	· ·	
C16	real	0	0	0	0	-2.7297	2.7300	0.0	1.0087	· ·	
C17	real	0	0	0	0	-1.7421	1.7429	0.0	1.0063	· ·	
C18	real	0	6265240	0	0	0	2.5482	1.0	1.1937	· ·	
C19	real	0	0	0	0	0.3654	12.8826	0.9861	0.5058	· ·	
C20	real	0	0	0	0	-2.4973	2.4980	-0.0	1.0077	· ·	

Untitled Flow



CS

getCloud



joe-demo

10 nodes

CLOUD STATUS

HEALTHY CONSENSUS LOCKED

Version	Started	Nodes (Used / All)
3.13.0.3981	a minute ago	10 / 10

NODES

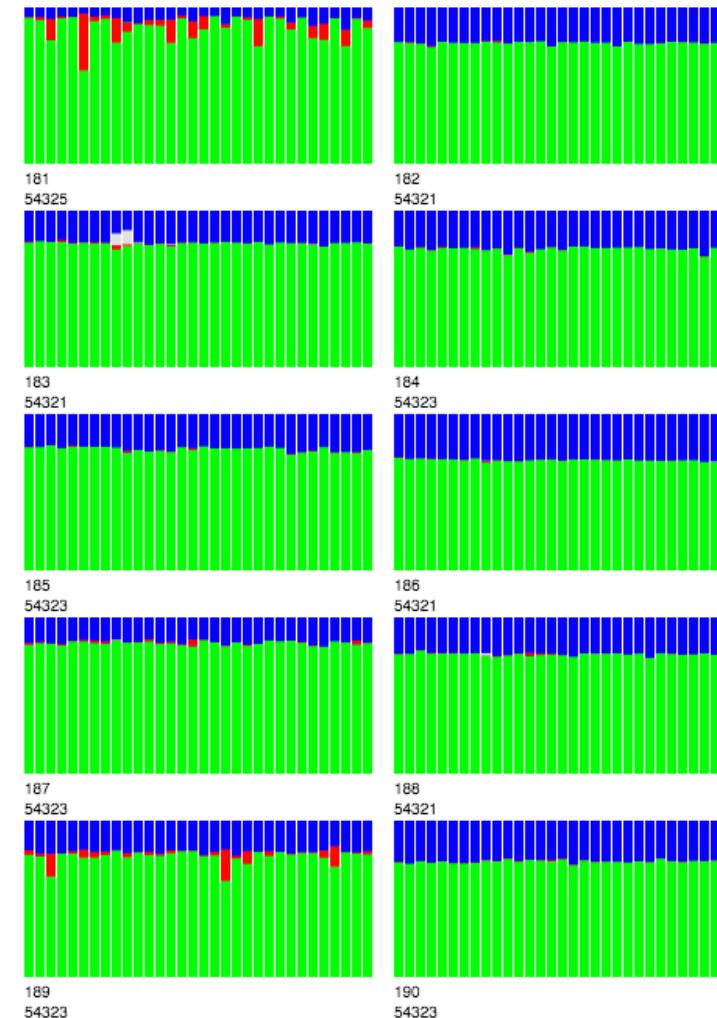
Name	Ping	Cores	Load	My CPU %	Sys	Shut Down	Data (Used/Total)	Data (% Cached)	GC (Free / Total / Max)	Disk (Free / Max)	Disk (% Free)
✓ 172.16.2.181:54323	a few seconds ago	32	6.110	0	8	-	40.603	33.82 GB / s	29.46 GB / NaN undefined / 29.58 GB	339.08 GB / 1.70 TB	19%
✓ 172.16.2.182:54321	a few seconds ago	32	0.240	7	8	-	44.566	39.59 GB / s	29.43 GB / NaN undefined / 29.58 GB	225.64 GB / 1.70 TB	12%
✓ 172.16.2.183:54321	a few seconds ago	32	9.820	0	3	-	44.883	42.09 GB / s	29.34 GB / NaN undefined / 29.58 GB	450.18 GB / 1.70 TB	25%
✓ 172.16.2.184:54323	a few seconds ago	32	0.990	0	0	-	44.656	41.67 GB / s	29.51 GB / NaN undefined / 29.58 GB	254.96 GB / 1.70 TB	14%
✓ 172.16.2.185:54323	a few seconds ago	32	0.440	8	8	-	43.128	38.33 GB / s	29.43 GB / NaN undefined / 29.58 GB	501.02 GB / 1.70 TB	28%
✓ 172.16.2.186:54321	a few seconds ago	32	1.750	0	0	-	44.589	42.46 GB / s	29.42 GB / NaN undefined / 29.58 GB	331.27 GB / 1.70 TB	18%
✓ 172.16.2.187:54323	a few seconds ago	32	1.490	0	10	-	43.993	42.00 GB / s	29.46 GB / NaN undefined / 29.58 GB	367.40 GB / 1.70 TB	21%
✓ 172.16.2.188:54321	a few seconds ago	32	0.610	0	8	-	41.977	18.63 GB / s	28.30 GB / NaN undefined / 29.58 GB	218.27 GB / 1.70 TB	12%
✓ 172.16.2.189:54323	a few seconds ago	32	4.420	6	9	-	48.590	38.91 GB / s	29.34 GB / NaN undefined / 29.58 GB	477.97 GB / 1.70 TB	27%
✓ 172.16.2.190:54323	a few seconds ago	32	2.970	10	12	-	43.931	22.15 GB / s	29.51 GB / NaN undefined / 29.58 GB	274.50 GB / 1.70 TB	15%
✓ TOTAL	-	320	28.840	-	-	-	440.916	359.62 GB / s	293.18 GB / NaN undefined / 295.83 GB	3.36 TB / 17.04 TB	19%

$$10 \times 32 = \\ 320 \text{ Cores}$$

$$10 \times 29.6 = 296 \\ \text{GB Memory}$$

H₂O Water Meter (CPU Monitor)

10 x 32 = 320 Cores



Legend

Each bar represents one CPU.

Blue: idle time

Green: user time

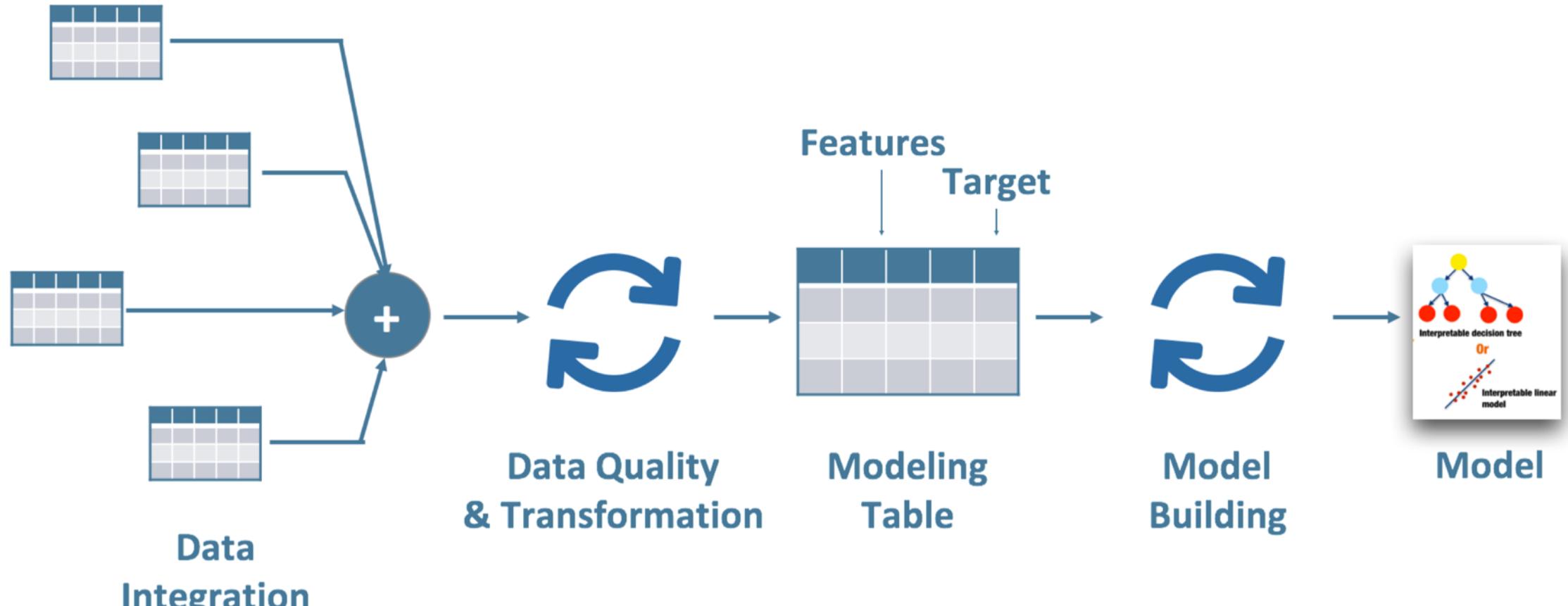
Red: system time

White: other time (e.g. i/o)

Demo: AutoML

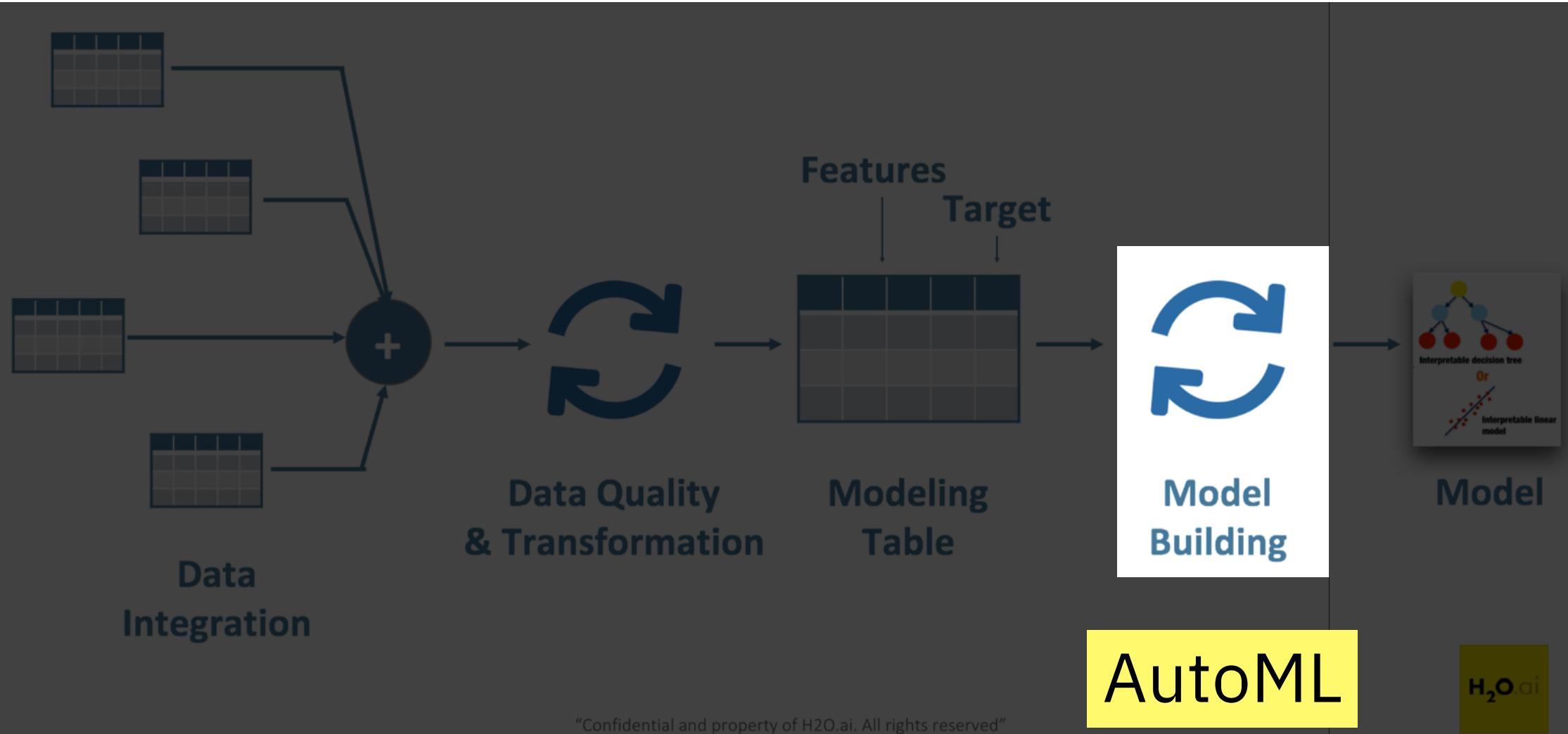
Automatic Machine Learning with H₂O
(R Interface)

Typical Enterprise Machine Learning Workflow



“Confidential and property of H2O.ai. All rights reserved”

Typical Enterprise Machine Learning Workflow



AutoML Output

The AutoML object includes a “leaderboard” of models that were trained in the process, ranked by a default metric based on the problem type (the second column of the leaderboard). In binary classification problems, that metric is AUC, and in multiclass classification problems, the metric is mean per-class error. In regression problems, the default sort metric is deviance. Some additional metrics are also provided, for convenience.

Here is an example leaderboard for a binary classification task:

model_id	auc	logloss
StackedEnsemble_0_AutoML_20170605_212658	0.776164	0.564872
GBM_grid_0_AutoML_20170605_212658_model_2	0.75355	0.587546
DRF_0_AutoML_20170605_212658	0.738885	0.611997
GBM_grid_0_AutoML_20170605_212658_model_0	0.735078	0.630062
GBM_grid_0_AutoML_20170605_212658_model_1	0.730645	0.67458
XRT_0_AutoML_20170605_212658	0.728358	0.629296
GLM_grid_0_AutoML_20170605_212658_model_1	0.685216	0.635137
GLM_grid_0_AutoML_20170605_212658_model_0	0.685216	0.635137

Regression Example: Boston Housing

Data Set Characteristics:

- Number of Instances: 506
- Number of Attributes: 13 numeric/categorical predictive
- Median Value (attribute 14) is the target
- Attribute Information (in order):
 - CRIM per capita crime rate by town
 - ZN proportion of residential land zoned for lots over 25,000 sq.ft.
 - INDUS proportion of non-retail business acres per town
 - CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
 - NOX nitric oxides concentration (parts per 10 million)
 - RM average number of rooms per dwelling
 - AGE proportion of owner-occupied units built prior to 1940
 - DIS weighted distances to five Boston employment centres
 - RAD index of accessibility to radial highways
 - TAX full-value property-tax rate per \$10,000
 - PTRATIO pupil-teacher ratio by town
 - B $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town
 - LSTAT % lower status of the population
 - MEDV Median value of owner-occupied homes in \$1000's
- Creator: Harrison, D. and Rubinfeld, D.L.
- Source: <http://archive.ics.uci.edu/ml/datasets/Housing>

Regression Example: Boston Housing

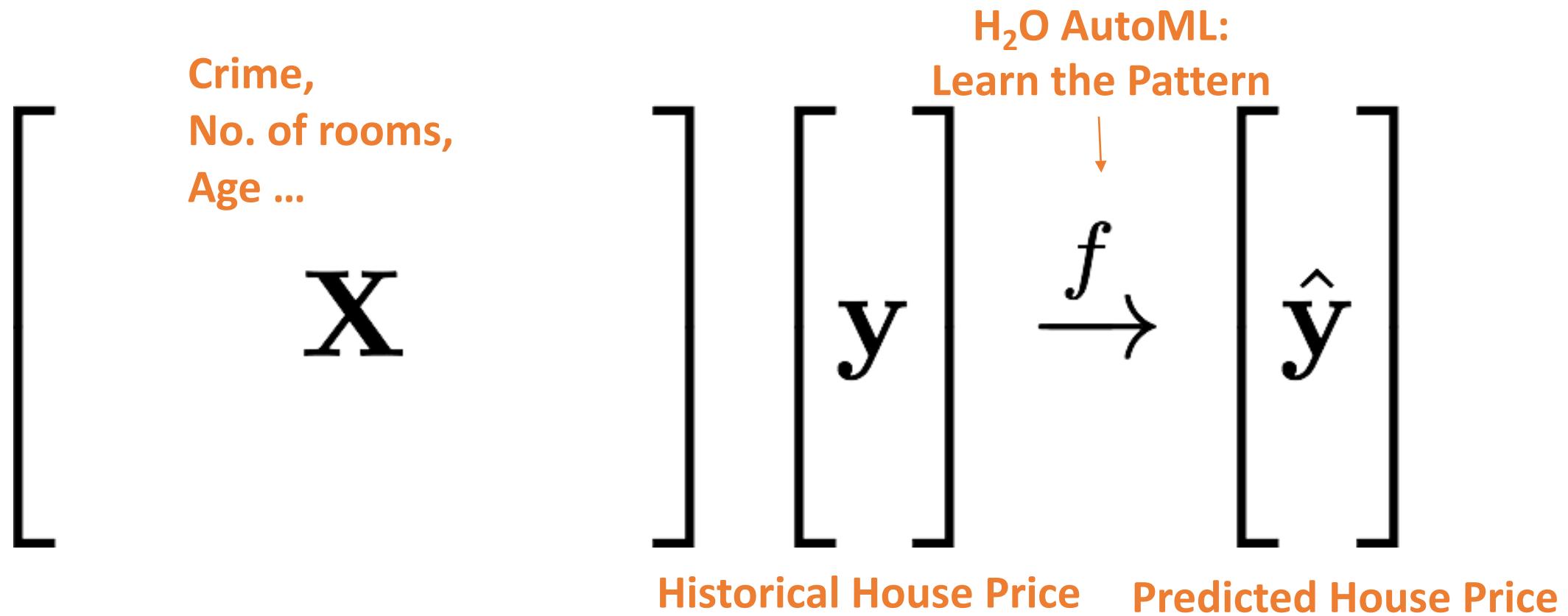
```
library(mlbench) # for dataset
data("BostonHousing")
dim(BostonHousing)

## [1] 506 14

# First six samples
knitr::kable(head(BostonHousing), format = "html")
```

crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv
0.00632	18	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
0.02985	0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7

Learning from Boston Housing Data



Boston Housing (Simple Split)

```
# Define features
features = setdiff(colnames(BostonHousing), "medv")
features

## [1] "crim"      "zn"        "indus"      "chas"       "nox"        "rm"        "age"
## [8] "dis"        "rad"       "tax"        "ptratio"    "b"          "lstat"

# Pick four random samples for test dataset
set.seed(1234)
row_test_samp = sample(1:nrow(BostonHousing), 4)

# Train
x_train = BostonHousing[-row_test_samp, features]
y_train = BostonHousing[-row_test_samp, "medv"]

# Test
x_test = BostonHousing[row_test_samp, features]
y_test = BostonHousing[row_test_samp, "medv"]
```

H2O AutoML

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

##
## H2O is not running yet, starting it now ...
##
## Note: In case of errors look at the following log files:
##       /var/folders/4z/p7yt7_4n4fj1jlyq6g4qhfbw0000gn/T//Rtmpptonn8/h2o_jofaichow_started_from_r.out
##       /var/folders/4z/p7yt7_4n4fj1jlyq6g4qhfbw0000gn/T//Rtmpptonn8/h2o_jofaichow_started_from_r.err
##
##
## Starting H2O JVM and connecting: ... Connection successful!
##
## R is connected to the H2O cluster:
##   H2O cluster uptime:      2 seconds 548 milliseconds
##   H2O cluster timezone:    Europe/Amsterdam
##   H2O data parsing timezone: UTC
##   H2O cluster version:     3.18.0.1
##   H2O cluster version age: 7 days, 21 hours and 16 minutes
##   H2O cluster name:        H2O_started_from_R_jofaichow_ygh539
##   H2O cluster total nodes: 1
##   H2O cluster total memory: 3.56 GB
##   H2O cluster total cores: 8
```

Prepare H2O Data Frames

```
# Prepare Data
h_train = as.h2o(BostonHousing[-row_test_samp,])
h_test = as.h2o(BostonHousing[row_test_samp,])

head(h_test)

##      crim   zn indus chas   nox     rm    age    dis   rad tax ptratio      b
## 1 0.01432 100   1.32   0 0.411 6.816 40.5 8.3248    5 256  15.1 392.90
## 2 0.36920   0   9.90   0 0.544 6.567 87.3 3.6023    4 304  18.4 395.69
## 3 0.04932  33   2.18   0 0.472 6.849 70.3 3.1827    7 222  18.4 396.90
## 4 0.26938   0   9.90   0 0.544 6.266 82.8 3.2628    4 304  18.4 393.39
##      lstat medv
## 1  3.95 31.6
## 2  9.28 23.8
## 3  7.53 28.2
## 4  7.90 21.6
```

Train Multiple H2O Models

```
# Train multiple H2O models with a simple API
# Stacked Ensembles will be created from those H2O models
# You tell H2O 1) how much time you have and/or 2) how many models do you want
model_automl = h2o.automl(x = features,
                           y = "medv",
                           training_frame = h_train,
                           nfolds = 5,
                           max_runtime_secs = 120, # time
                           max_models = 20,       # max models
                           stopping_metric = "RMSE",
                           seed = 1234)
```

H2O: AutoML Model Leaderboard

```
# Print out leaderboard
model_automl@leaderboard

##                                     model_id
## 1 StackedEnsemble_BestOfFamily_0_AutoML_20180221_015244
## 2             GBM_grid_0_AutoML_20180221_015244_model_0
## 3 StackedEnsemble_AllModels_0_AutoML_20180221_015244
## 4             GBM_grid_0_AutoML_20180221_015244_model_1
## 5             GBM_grid_0_AutoML_20180221_015244_model_3
## 6                 DRF_0_AutoML_20180221_015244
##   mean_residual_deviance      rmse      mae      rmsle
## 1          10.81557 3.288704 2.148544 0.140921
## 2          10.86044 3.295518 2.224282 0.145063
## 3          10.89855 3.301295 2.161700 0.141057
## 4          11.88445 3.447383 2.285338 0.145858
## 5          12.12041 3.481438 2.324986 0.148829
## 6          12.22679 3.496683 2.339066 0.148301
##
## [22 rows x 5 columns]
```

H2O: Model Leader

```
# Best Model (either an individual model or a stacked ensemble)
model_automl@leader

## Model Details:
## =====
##
## H2OResponseModel: stackedensemble
## Model ID: StackedEnsemble_BestOfFamily_0_AutoML_20180221_015244
## NULL
##
##
## H2OResponseMetrics: stackedensemble
## ** Reported on training data. **
##
## MSE: 0.8307283
## RMSE: 0.911443
## MAE: 0.6676121
## RMSLE: 0.04544841
## Mean Residual Deviance : 0.8307283
##
##
## H2OResponseMetrics: stackedensemble
## ** Reported on validation data. **
##
## MSE: 6.68755
## RMSE: 2.58603
```

H2O: Making Prediction

```
# Using the best model to make predictions on test set
yhat_test = h2o.predict(model_automl@leader, h_test)

# Create a new data frame to compare target (medv) and predictions
d_test = data.frame(x_test,
                     medv = y_test,
                     predict = as.data.frame(yhat_test),
                     row.names = NULL)
knitr::kable(d_test, format = "html")
```

crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv	predict
0.01432	100	1.32	0	0.411	6.816	40.5	8.3248	5	256	15.1	392.90	3.95	31.6	30.94657
0.36920	0	9.90	0	0.544	6.567	87.3	3.6023	4	304	18.4	395.69	9.28	23.8	22.84410
0.04932	33	2.18	0	0.472	6.849	70.3	3.1827	7	222	18.4	396.90	7.53	28.2	30.04268
0.26938	0	9.90	0	0.544	6.266	82.8	3.2628	4	304	18.4	393.39	7.90	21.6	21.96694

Other H₂O News

Latest Developments

Events

H₂O Products



In-Memory, Distributed
Machine Learning Algorithms
with H2O Flow GUI



H2O AI Open Source Engine
Integration with Spark



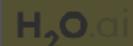
Lightning Fast machine
learning on GPUs

DRIVERLESSAI

Automatic feature
engineering, machine
learning and interpretability

Steam

Secure multi-tenant H2O clusters

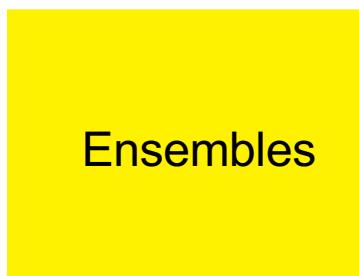


Algorithms on H₂O-3 (CPU)

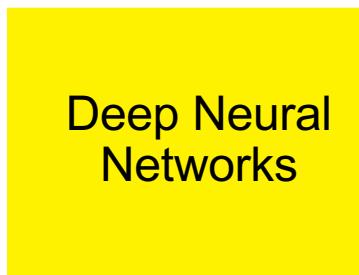
Supervised Learning



- Generalized Linear Models: Binomial, Gaussian, Gamma, Poisson and Tweedie
- Naïve Bayes



- Distributed Random Forest: Classification or regression models
- Gradient Boosting Machine: Produces an ensemble of decision trees with increasing refined approximations

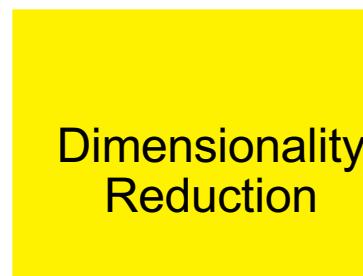


- Deep learning: Create multi-layer feed forward neural networks starting with an input layer followed by multiple layers of nonlinear transformations

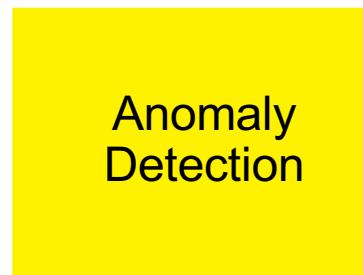
Unsupervised Learning



- K-means: Partitions observations into k clusters/groups of the same spatial size. Automatically detect optimal k



- 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



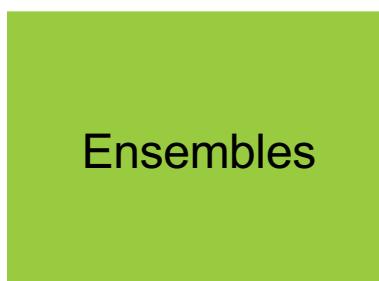
- Autoencoders: Find outliers using a nonlinear dimensionality reduction using deep learning

Algorithms on H₂O4GPU (more to come)

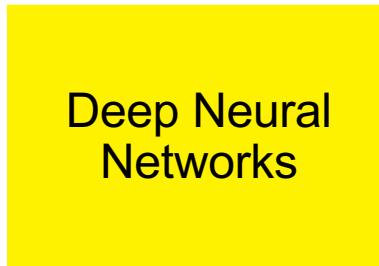
Supervised Learning



- Generalized Linear Models: Binomial, Gaussian, Gamma, Poisson and Tweedie
- Naïve Bayes



- Distributed Random Forest: Classification or regression models
- Gradient Boosting Machine: Produces an ensemble of decision trees with increasing refined approximations

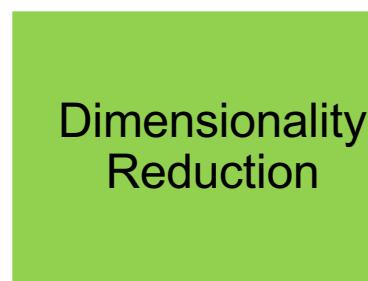


- Deep learning: Create multi-layer feed forward neural networks starting with an input layer followed by multiple layers of nonlinear transformations

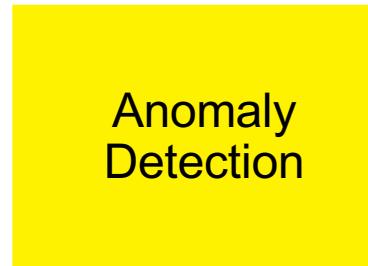
Unsupervised Learning



- K-means: Partitions observations into k clusters/groups of the same spatial size. Automatically detect optimal k



- 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



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

H2O4GPU now available in R

BY ERIN LEDELL ON MARCH 27, 2018 – 0 COMMENTS

In September, H2O.ai released a new open source software project for GPU machine learning called [H2O4GPU](#). The initial release (blog post [here](#)) included a Python module with a scikit-learn compatible API, which allows it to be used as a drop-in replacement for scikit-learn with support for GPUs on selected (and ever-growing) algorithms. We are proud to announce that the same collection of GPU algorithms is now available in R, and the `h2o4gpu` R package is available on [CRAN](#).



<https://github.com/h2oai/h2o4gpu>



Machine Learning

Title	Presenter(s)
Automatic and Interpretable Machine Learning in R with H2O and LIME	Jo-fai Chow 

Overview

General Data Protection Regulation (GDPR) is just around the corner. The regulation will become enforceable a week after the eRum conference (from 25 May 2018). Are you and your organization ready to explain your models?

This is a hands-on tutorial for R beginners. I will demonstrate the use of two R packages, h2o & LIME, for automatic and interpretable machine learning. Participants will be able to follow and build regression and classification models quickly with H2O's AutoML. They will then be able to explain the model outcomes with a framework called Local Interpretable Model-Agnostic Explanations (LIME).

References:

- Hall et al (2017): Ideas on interpreting machine learning.
- Ribeiro et al (2016): Introduction to Local Interpretable Model-Agnostic Explanations (LIME).
- Ribeiro et al (2016): "Why Should I Trust You?": Explaining the Predictions of Any Classifier.
- Wikipedia (2018): General Data Protection Regulation.

End of First Talk

Any Questions?

Making Multimillion-Dollar Decisions with H₂O AutoML, LIME and Shiny

My journey to a real Moneyball application

About Moneyball

The **first rule** of Moneyball:

You do not ask me about the names of team and player involved.

The **second rule** of Moneyball:

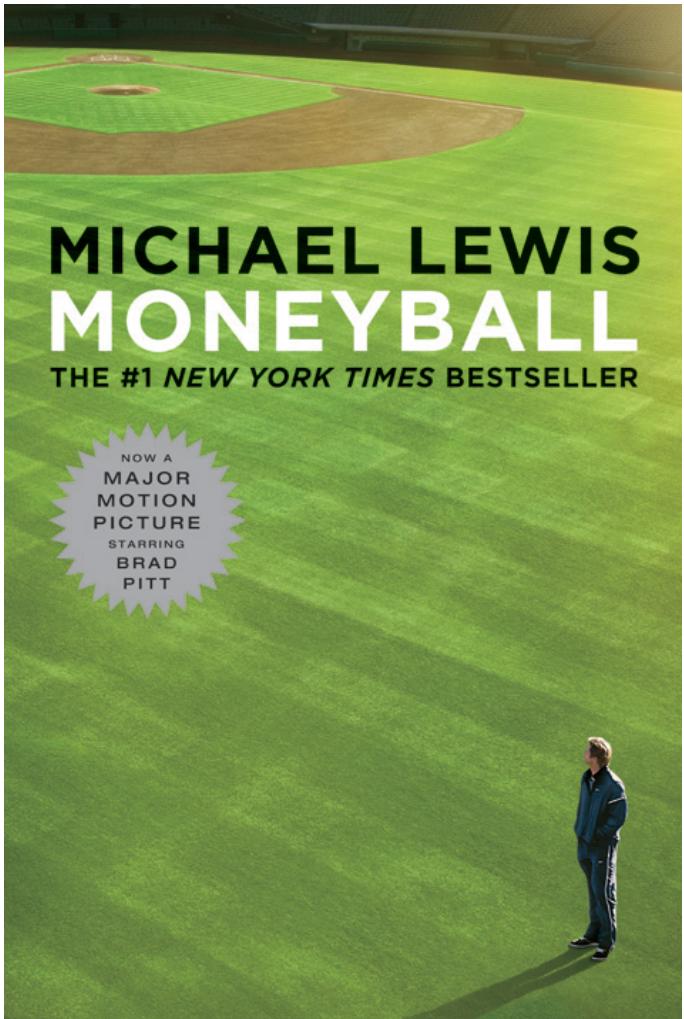
You do not ask me about the names of team and player involved.

(... for legal reasons ...)

The **third rule** of Moneyball:

If you happen to guess the names right, I can neither confirm nor deny.

About Moneyball



Billy Beane

Peter Brand
(based on Paul DePodesta)

Ari Kaplan – the Real “Moneyball” Guy

- The real characters in the movie (Billy Beane and Paul DePodesta) did not want to work with Hollywood.
- The filmmaker interviewed Ari instead and created the Paul DePodesta character based on Ari’s real-life story.
- Ari happens to work at Aginity so we have a real “Moneyball” guy for this demo.



A Proof-of-Concept Demo for IBM Think Conference



Moneyball [Demo](#)

- Introduction
- Results (Pitching)
- Results (Batting)
- About Us
- YouTube

Hit a Home Run Making Baseball Decisions Using Artificial Intelligence and Machine Learning

Thursday, 1:30 PM - 2:10 PM | Session ID: 3456A
Mandalay Bay South, Level 2 | Breakers C

IBM + aginity + H₂O.ai

Join Ari Kaplan, a real "MoneyBall" and well known around Major League Baseball, Joe Chow, a H2O data scientist, and David Kearns from IBM's Analytics Ecosystem team for this fun, interactive session where you will have the chance to see where artificial intelligence meets business intelligence. Ari and Joe will briefly present the latest machine learning technologies and concepts powering today's baseball decisions, including Hortonworks Data Platform, Spark, Aginity Amp, H2O.ai, IBM Data Science Experience and more. You will then step up to the plate as general manager to see how your player decisions would stack up under World Series pressure. Are you ready to play ball?

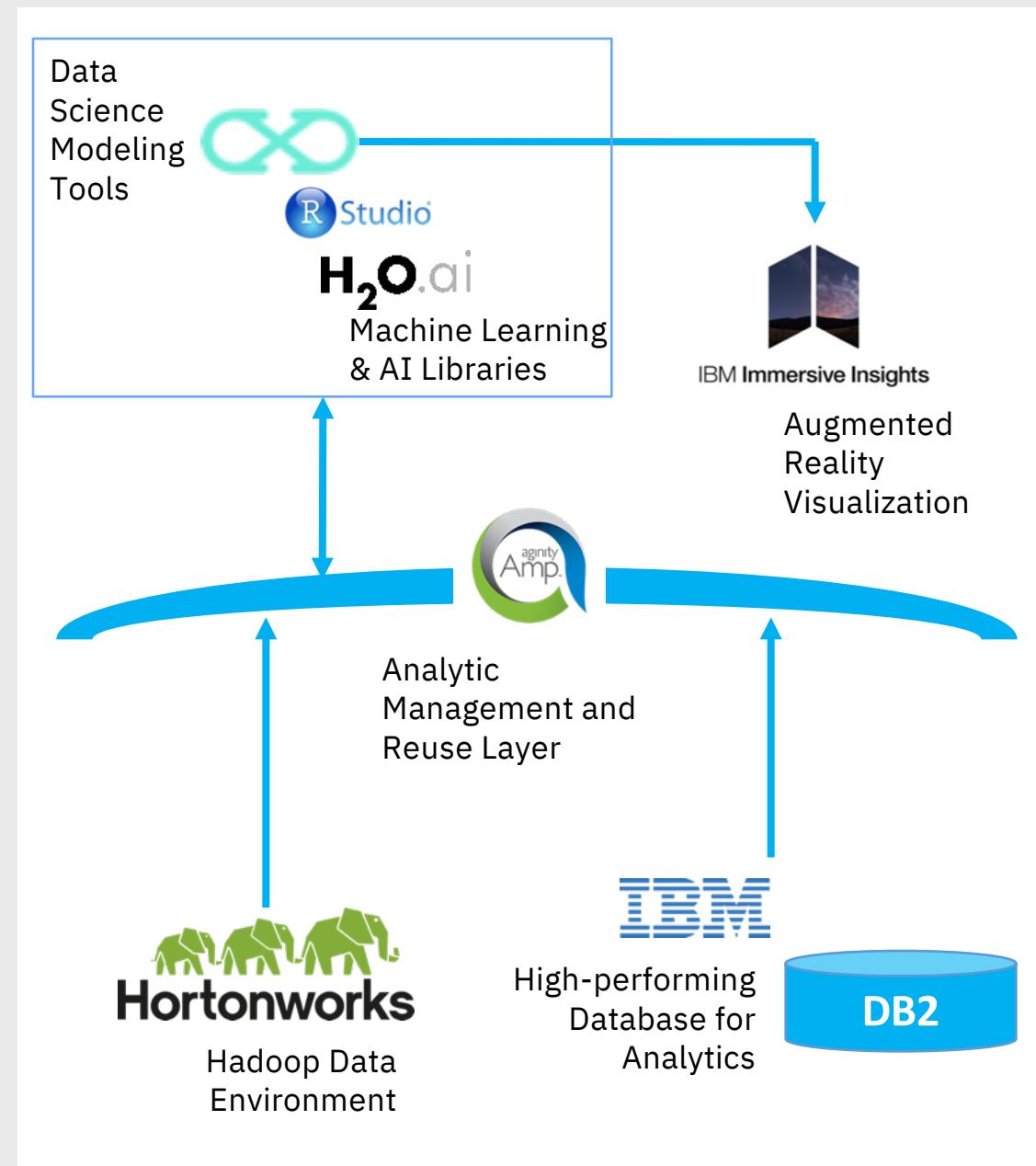
Speakers:

- Ari Kaplan, Aginity
- Jo-fai Chow, H2O.ai
- David Kearns, IBM

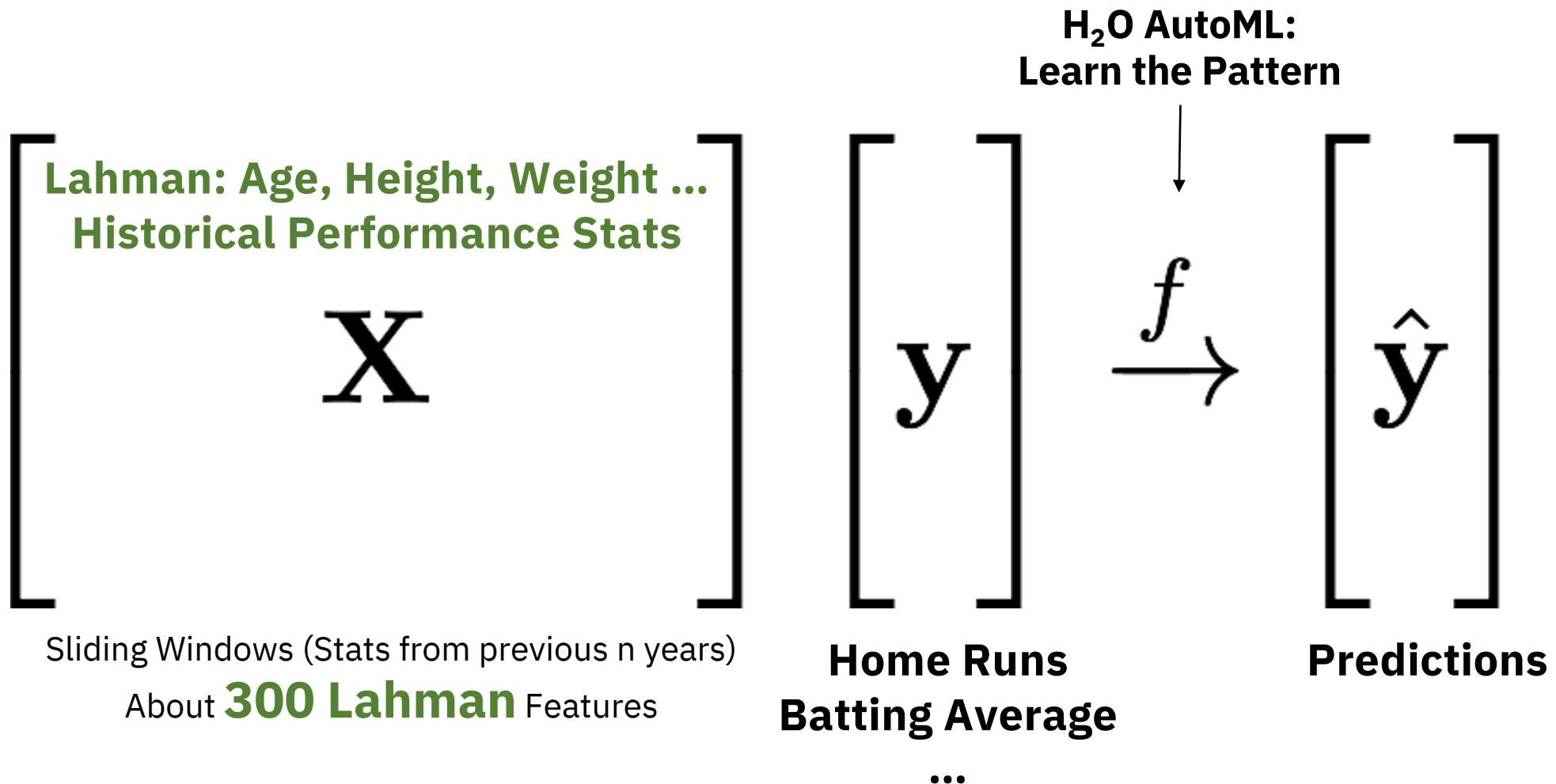
Enterprise Solution

The Workflow

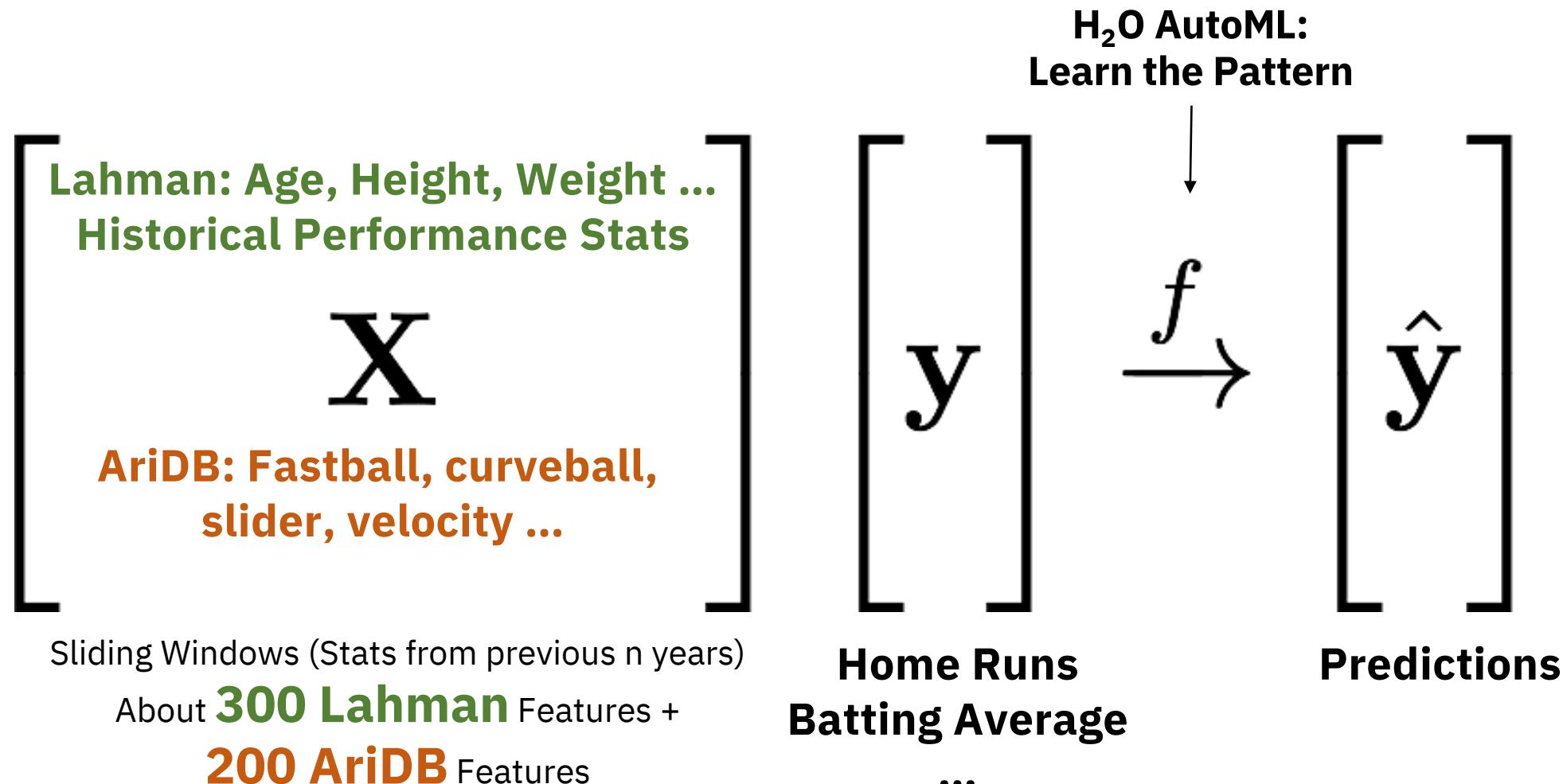
1. Data loaded into the databases
2. Connected diverse data sources to Amp
3. Amp used to create derived attributes and publish them and data to DSX and H₂O
4. DSX and H₂O to build and tweak statistical and machine learning models
5. Visualizations tested in Immersive Insights
6. Steps 4 and 5 repeated to get settled data
7. Statistical and machine learning models saved in Amp
8. Data exported to Immersive Insights for final visualizations



Approach One: Learning from **Lahman** only

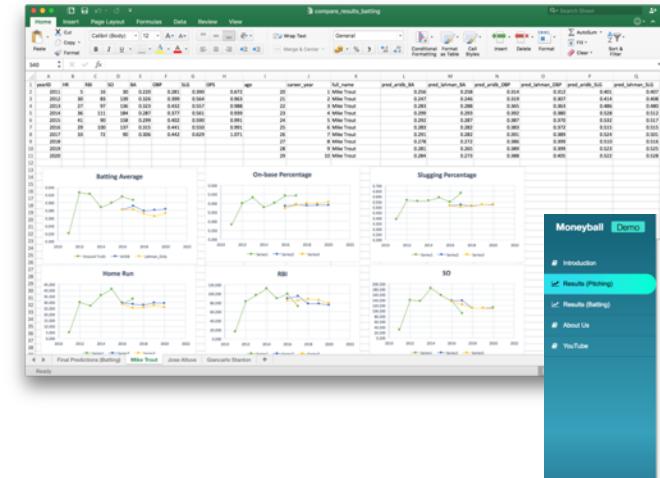


Approach Two: Learning from **Lahman** & **AriDB**



Timeline

- **March 19** – AutoML Predictions finalized. Initial presentation in Excel.
- **March 20** – Version 1 of Shiny app. Ari used to app to validate some players he had in mind and recommended one player to his team.
- **March 21** – Multimillion-dollar contract finalized.
- **March 22** – Moneyball presentation at IBM Think



Presentation Shiny App

IBM + aginity + H₂O.ai

Pitching Performance: Player Stats (Up to 2017) and Projection (2018-2020)

Select a Player

Name	Team (2017)	Weight	Height	Bats	Throws	Birth Country	Birth Year	Debut Year
Chris Sale	BOS	180	78	L	L	USA	1989	2010

Notes:

1. Training Period: 2010 to 2015.
2. Validation Period: 2016 and 2017.
3. Projection Period: 2018 to 2020.

Charts Table Explanation (ERA) Explanation (AVG) Explanation (WHIP)

Green: Predictions based on Lahman only

Orange: Predictions based on AriDB + Lahman

Moneyball Demo

IBM + aginity + H₂O.ai

Pitching Performance: Player Stats (Up to 2017) and Projection (2018-2020)

Select a Player

Name	Team (2017)	Weight	Height	Bats	Throws	Birth Country	Birth Year	Debut Year
Chris Sale	BOS	180	78	L	L	USA	1989	2010

Notes:

1. Training Period: 2010 to 2015.
2. Validation Period: 2016 and 2017.
3. Projection Period: 2018 to 2020.

Charts Table Explanation (ERA) Explanation (AVG) Explanation (WHIP)

Data	Year	ERA (Historical Data)	ERA (Predictions based on AriDB)	ERA (Predictions based on Lahman)	Avg (Historical Data)	Avg (Predictions based on AriDB)	Avg (Predictions based on Lahman)	WHIP (Historical Data)	WHIP (Predictions based on AriDB)	WHIP (Predictions based on Lahman)	
Training	2011	2.790							0.203		
Training	2012	3.050							0.235		
Training	2013	3.070							0.230		
Training	2014	2.170							0.205		
Training	2015	3.410							0.233		
Validation	2016	3.340	3.060	3.890	0.227	0.225	0.251	1.037	1.050	1.273	
Validation	2017	2.900	2.950	3.410	0.208	0.225	0.251	0.970	1.010	1.223	
Prediction	2018		2.910	3.810	0.214	0.242		0.956	1.215		
Prediction	2019		2.720	3.820	0.210	0.234		0.950	1.287		
Prediction	2020		2.620	4.100	0.203	0.242		0.894	1.281		

Moneyball Demo

IBM + aginity + H₂O.ai

Pitching Performance: Player Stats (Up to 2017) and Projection (2018-2020)

Select a Player

Name	Team (2017)	Weight	Height	Bats	Throws	Birth Country	Birth Year	Debut Year
Chris Sale	BOS	180	78	L	L	USA	1989	2010

Notes:

1. Training Period: 2010 to 2015.
2. Validation Period: 2016 and 2017.
3. Projection Period: 2018 to 2020.

Charts Table Explanation (ERA) Explanation (AVG) Explanation (WHIP)

Moneyball Demo

IBM + aginity + H₂O.ai

Batting Performance: Player Stats (Up to 2017) and Projection (2018-2020)

Select a Player

Name	Team (2017)	Weight	Height	Bats	Throws	Birth Country	Birth Year	Debut Year
Giancarlo Stanton	MA	245	78	R	R	USA	1989	2010

Notes:

1. Training Period: 2010 to 2015.
2. Validation Period: 2016 and 2017.
3. Projection Period: 2018 to 2020.

Charts (1/2) Charts (2/2) Table (1/2) Table (2/2) Exp. (BA) Exp. (HR) Exp. (RBI) Exp. (OBP) Exp. (SLG) Exp. (SO)

Acknowledgement



@DaithiOCiaran @arikaplan1 & @matlabulous are smoothly passing the mic back & forth to talk about their joint Moneyball project to a jam-packed room. #think2018 🎉
#machinelearning @Aginity @h2oai #DSX



9:04 PM - 22 Mar 2018



Merci beaucoup!

- Organisers & Sponsors
 - Alexia Audevert
 - Christophe Regouby
 - HarryCow Coworking
- H₂O's Mission
 - Democratize AI
 - Make Machine Learning Accessible to Everyone
- 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)