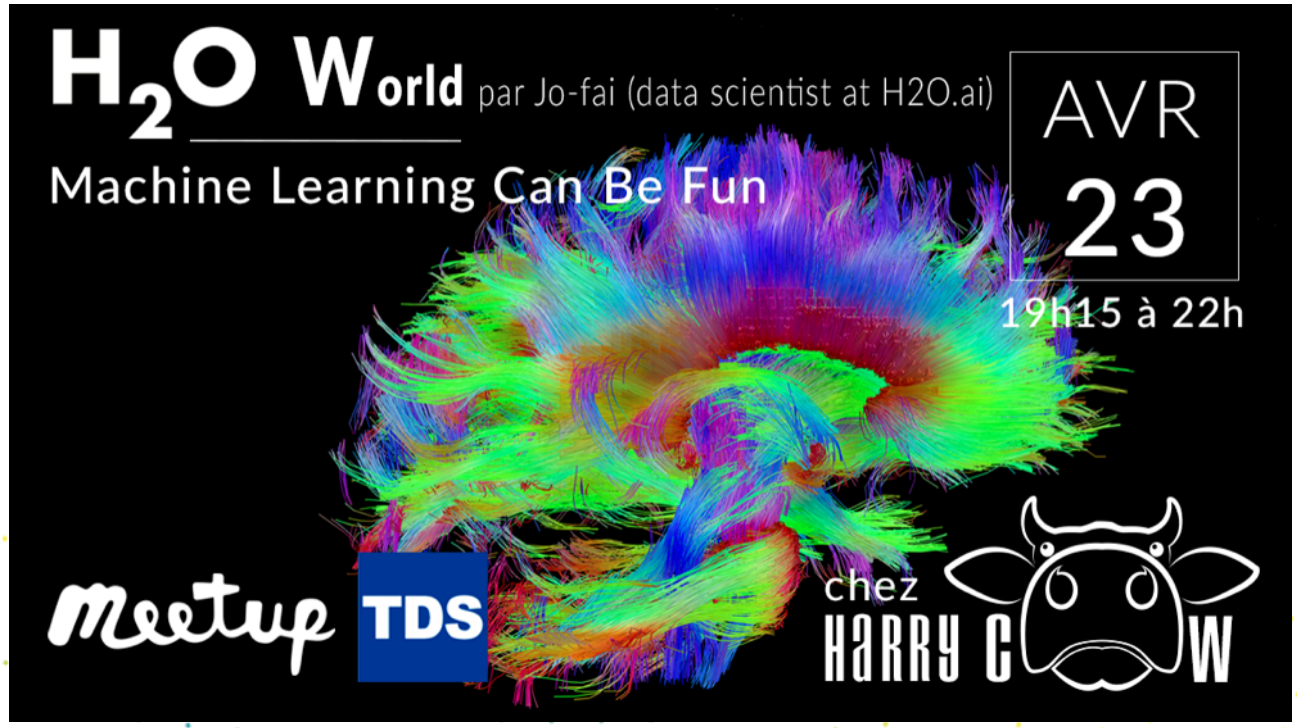
Scalable and Automatic Machine Learning with H_2O

Introduction, demos and a real-world use-case



Jo-fai (Joe) Chow

Data Scientist / Community Manager

joe@h2o.ai

@matlabulous

Agenda

- Talk 1: Introduction to H_2O
  - Company and People
  - H_2O Open Source ML Platform
  - Demos
    - H_2O on Hadoop (320 Cores)
    - AutoML
  - Other News
- Talk 2: Moneyball
  - From a proof-of-concept project to a multimillion-dollar contract

Company Overview

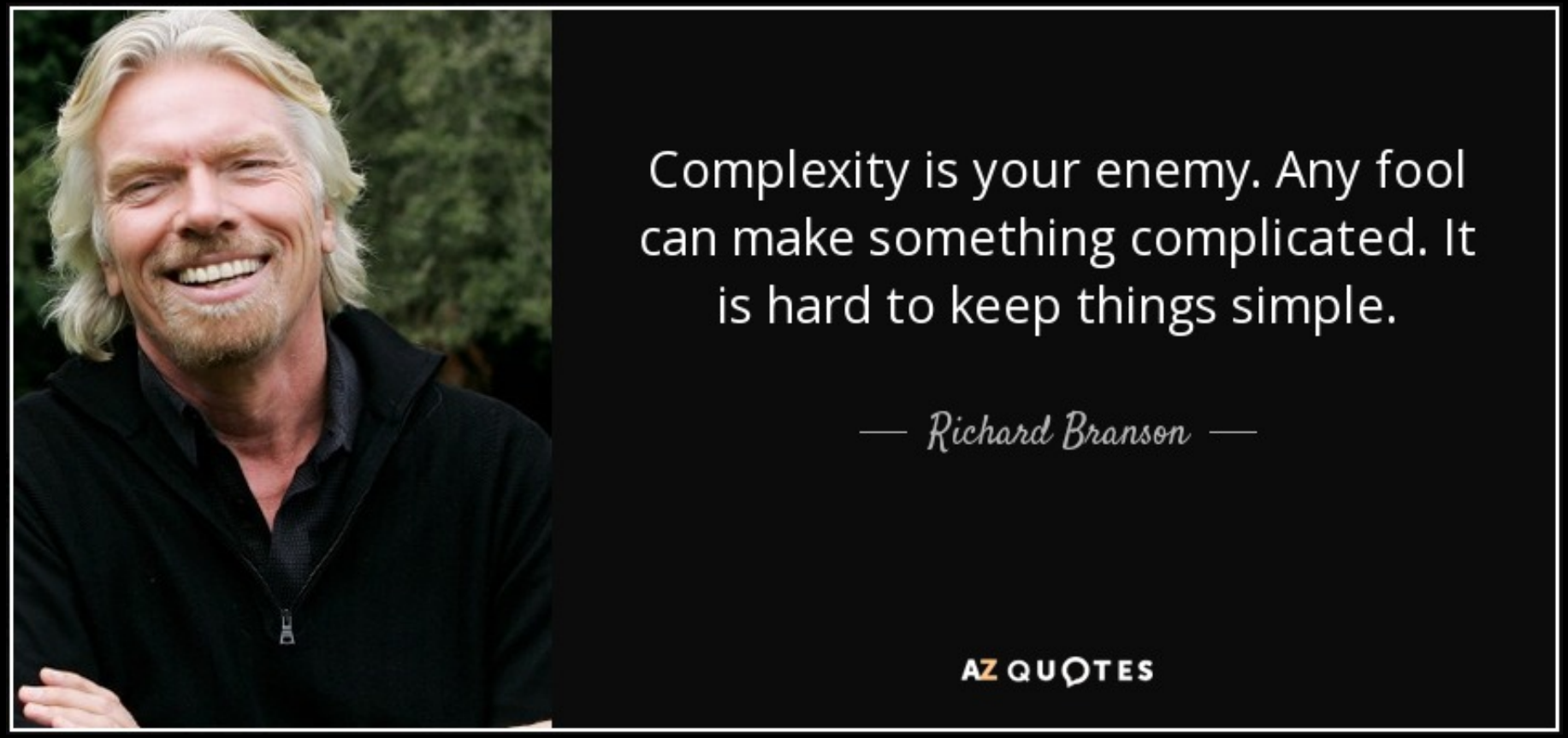
Founded	2012, Series C in Nov, 2017
Products	• Driverless AI – Automated Machine Learning • H_2O Open Source Machine Learning • Sparkling Water
Mission	Democratize AI. Do Good
Team	~100 employees • Distributed Systems Engineers doing Machine Learning • World-class visualization designers
Offices	Mountain View, London, Prague





H2O.ai

Our Mission:
Make Machine Learning Accessible to Everyone



Scientific Advisory Council

Dr. Trevor Hastie

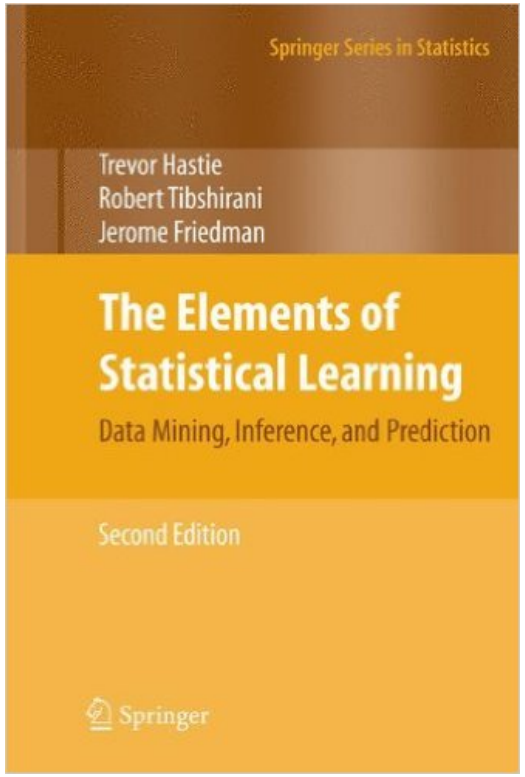
- John A. Overdeck Professor of Mathematics, Stanford University
- PhD in Statistics, Stanford University
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Co-author with John Chambers, *Statistical Models in S*
- Co-author, *Generalized Additive Models*

Dr. Robert Tibshirani

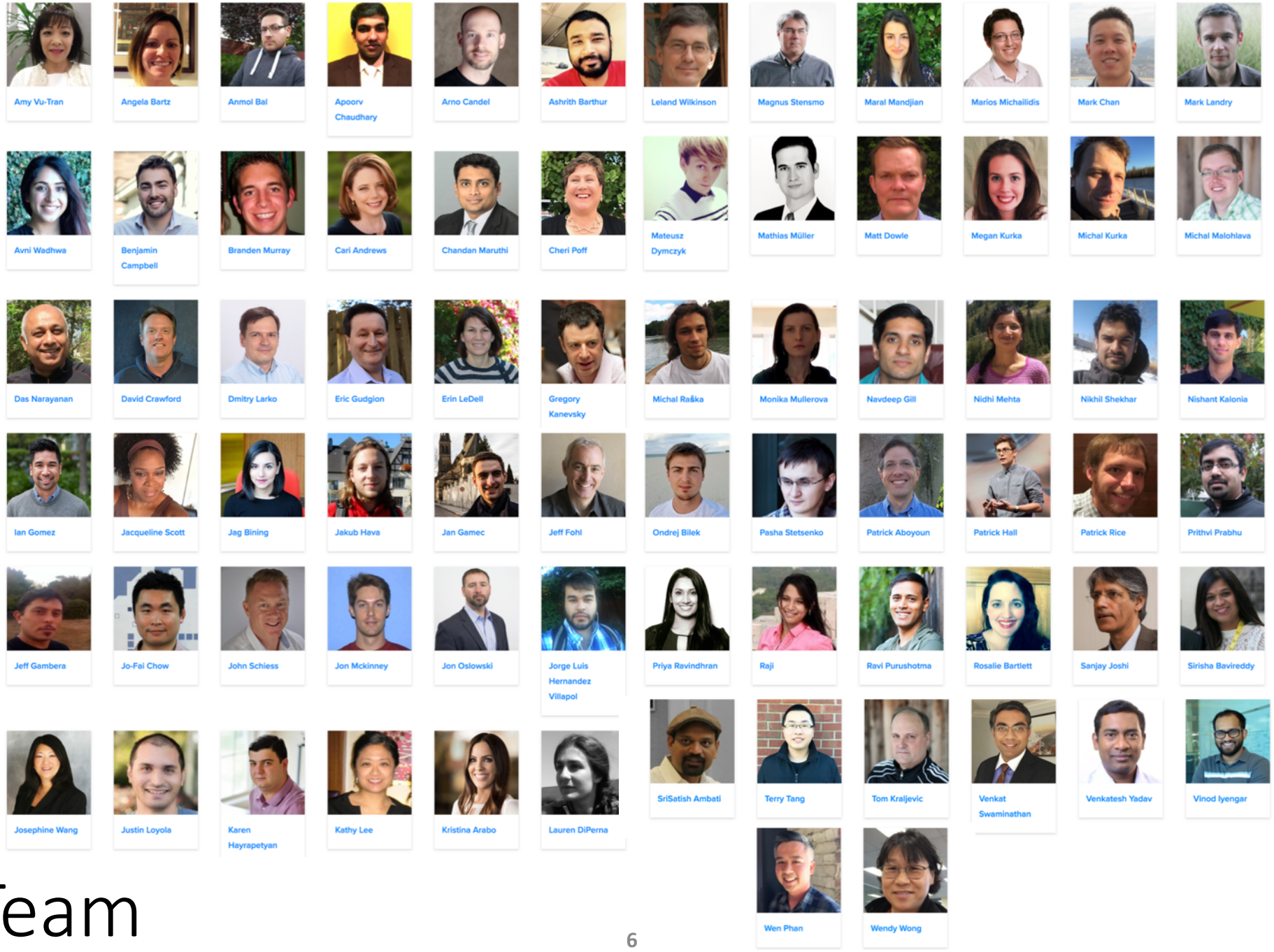
- Professor of Statistics and Health Research and Policy, Stanford University
- PhD in Statistics, Stanford University
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Author, *Regression Shrinkage and Selection via the Lasso*
- Co-author, *An Introduction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

Dr. Steven Boyd

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

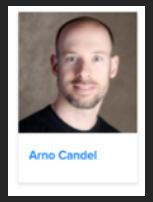


H2O.ai



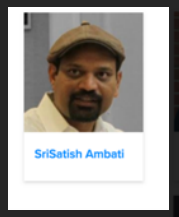
H_2O Team

H_2O.ai

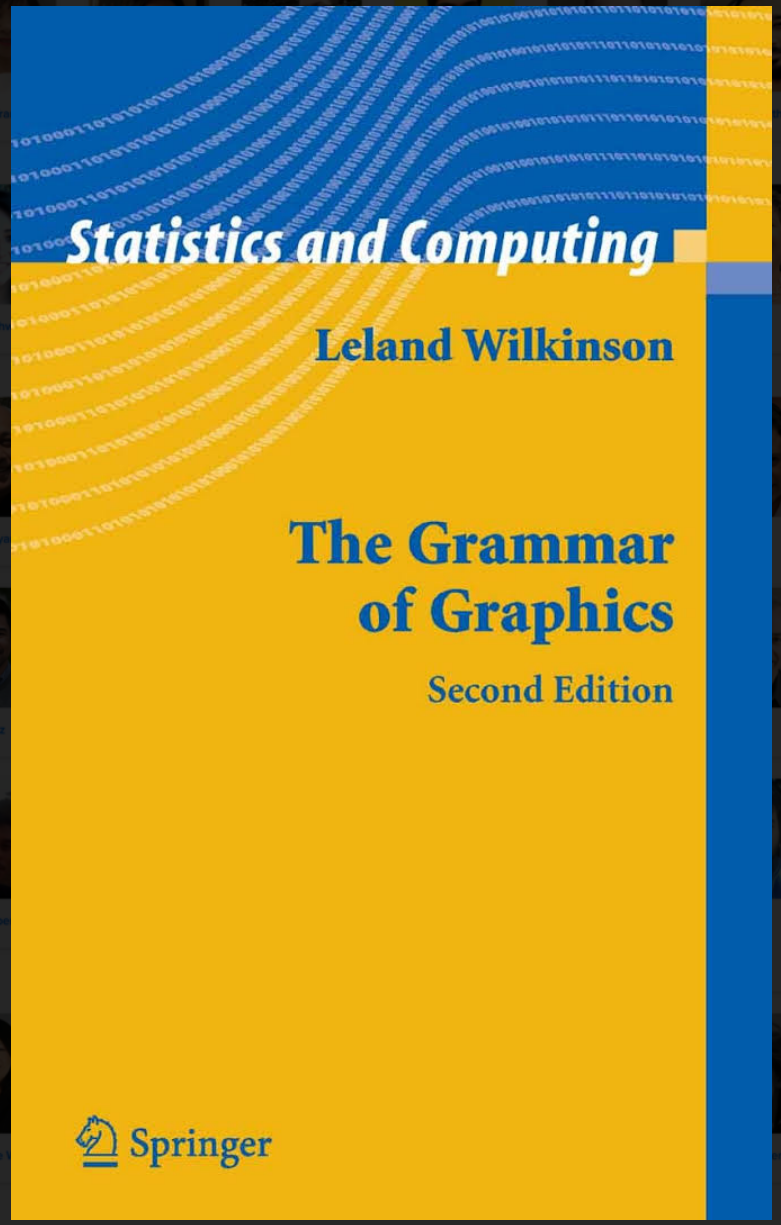


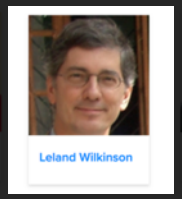
Arno Candel, CTO
Fortune's 2014 Big Data All-Star

Sri Ambati, Co-founder & CEO



H_2O Team





Origin of R Package `ggplot2`

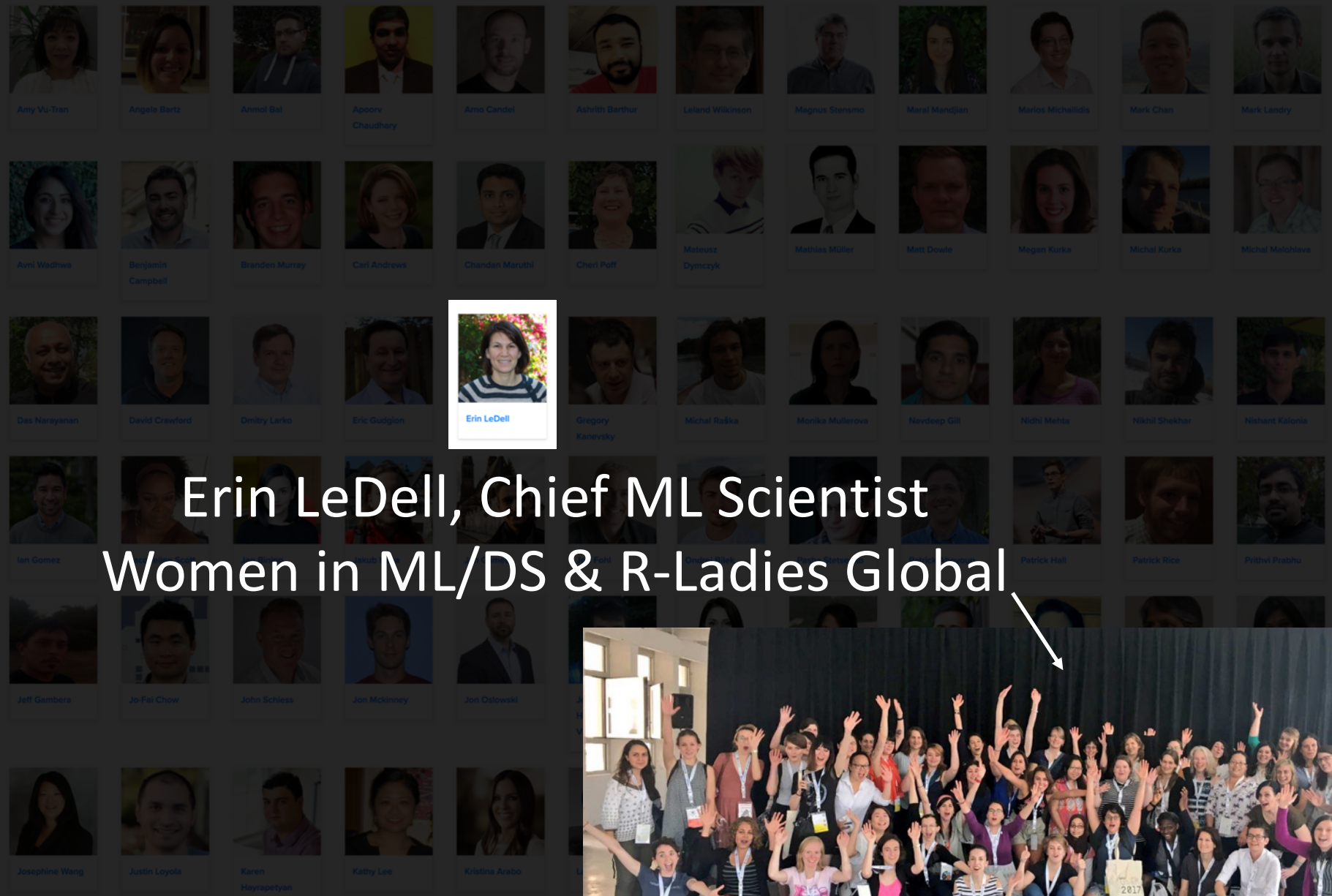
H_2O Team

H_2O.ai

data.table

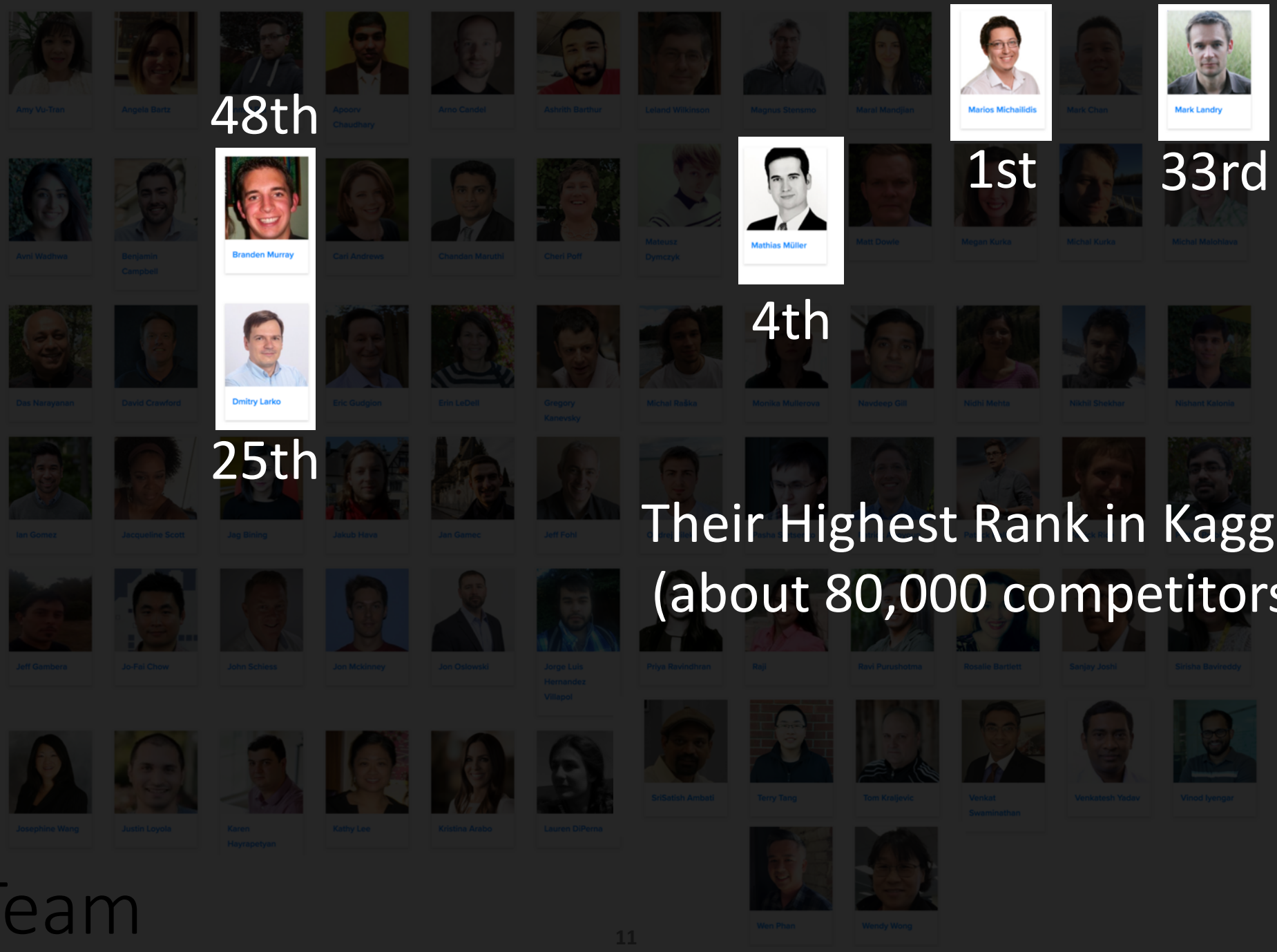
H_2O Team

H_2O.ai



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

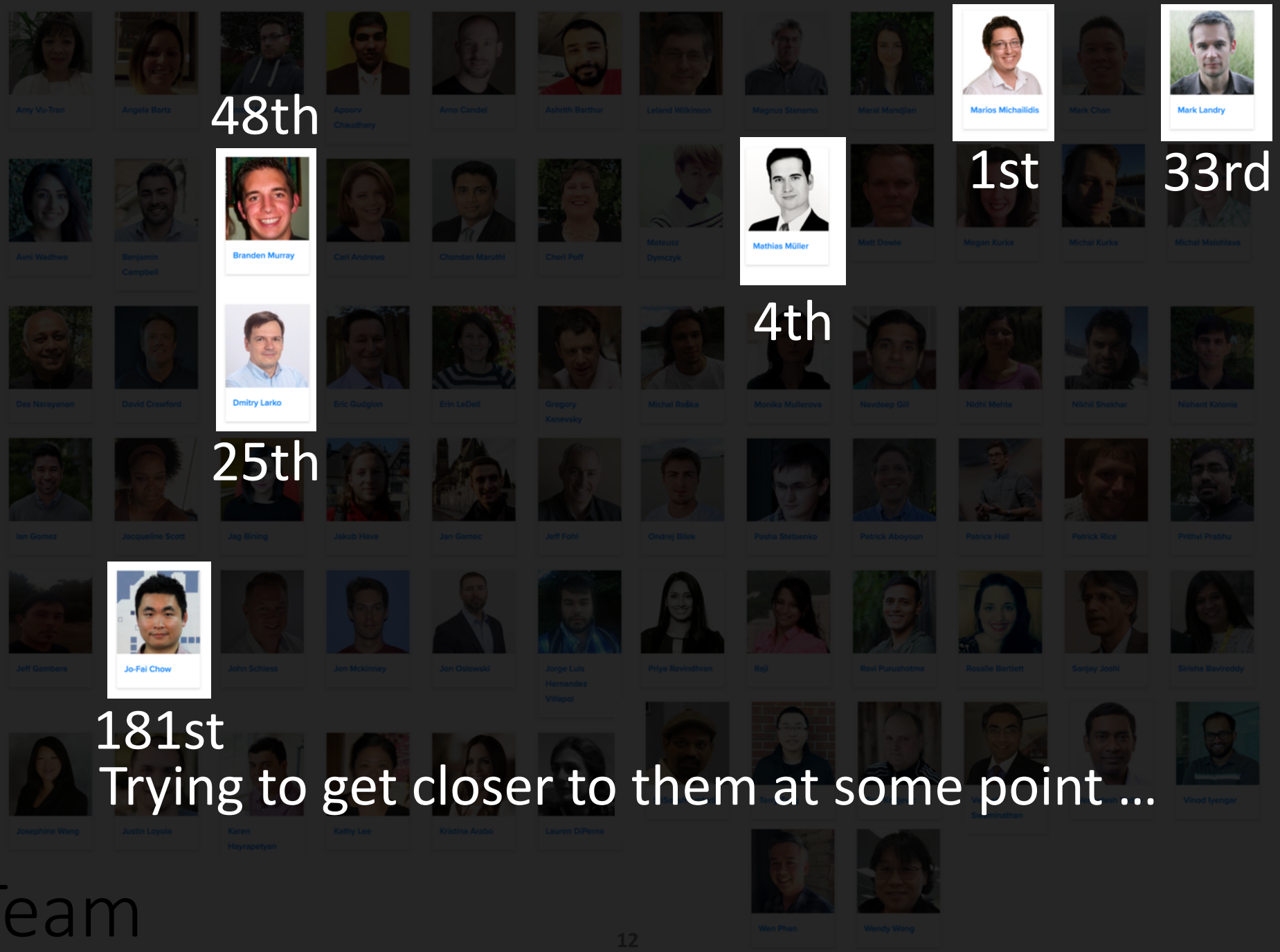
H_2O Team



Their Highest Rank in Kaggle
(about 80,000 competitors)

H_2O Team

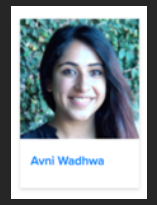
H_2O.ai



Trying to get closer to them at some point …

H_2O Team

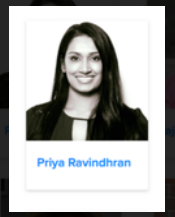
H_2O.ai



Avni

Bonsoir!

Joe



Priya

H_2O Team

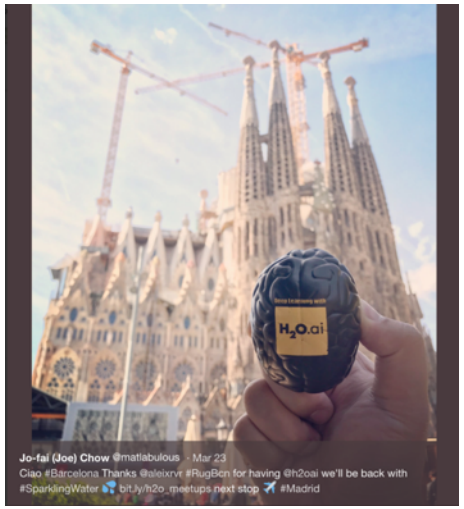
H_2O.ai

June 2017 - Present

H_2O Team in UK

Feb 2016 - Present

Joe's Roles at H_2O.ai



- Data Scientist /
  Sales Engineer /
  Speaker /
  Meetup Organiser /
  Community Evangelist
  (on paper)

- Unofficial Photographer of H_2O.ai SWAG
  (the travelling data scientist)

- H_2O.ai SWAG EMEA Distributor
  (please help yourself)

H2O.ai

Reminder: #360Selfie

Joe's Real Job at H_2O.ai

H_2O.ai

H_2O Products

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

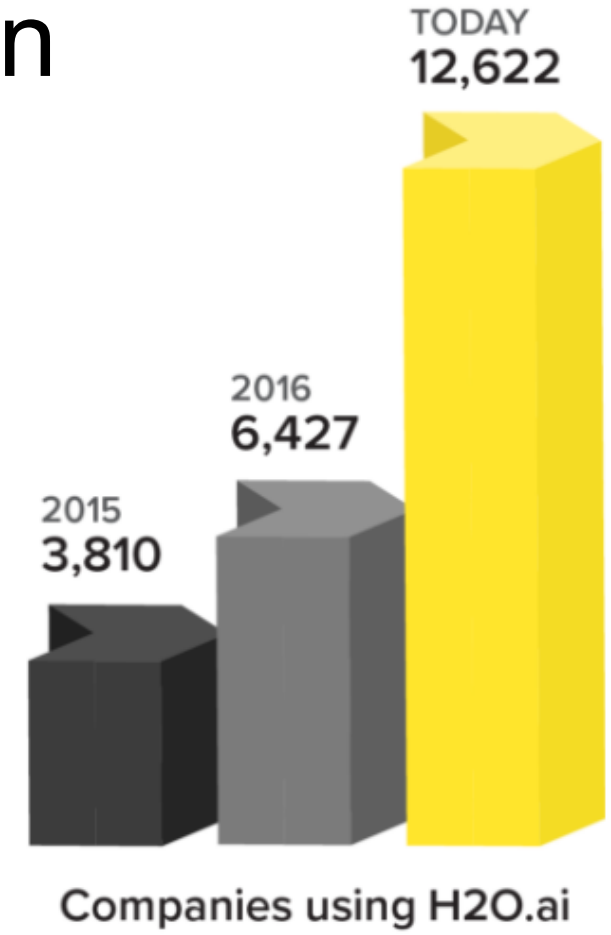
H2O AI Open Source Engine Integration with Spark

Lightning Fast machine learning on GPUs

Automatic feature engineering, machine learning and interpretability

Secure multi-tenant H2O clusters

Worldwide Community Adoption

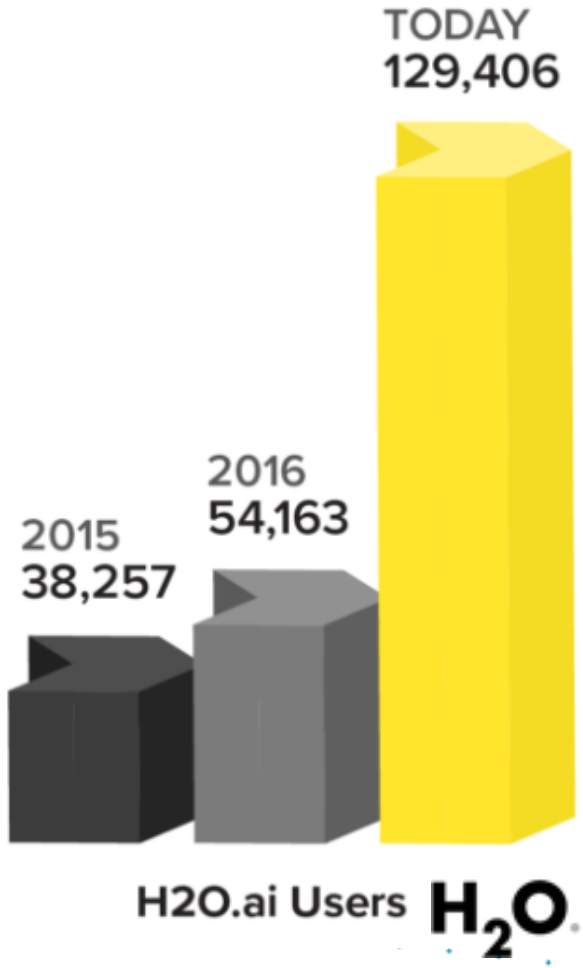


222 OF THE FORTUNE 500

8 OF TOP 10 BANKS

7 OF TOP 10 INSURANCE COMPANIES

4 OF TOP 10 HEALTHCARE COMPANIES



* DATA FROM GOOGLE ANALYTICS EMBEDDED IN THE END USER PRODUCT

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)

Platforms with H_2O integration

H_2O + KNIME Talk at KNIME Summit Mar 2017

H2O.ai Solution Leadership Across Verticals

H2O.ai

Community Expansion

Meetup

88,286	43	52	48	18
members	interested	Meetups	cities	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

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

H_2O Products

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

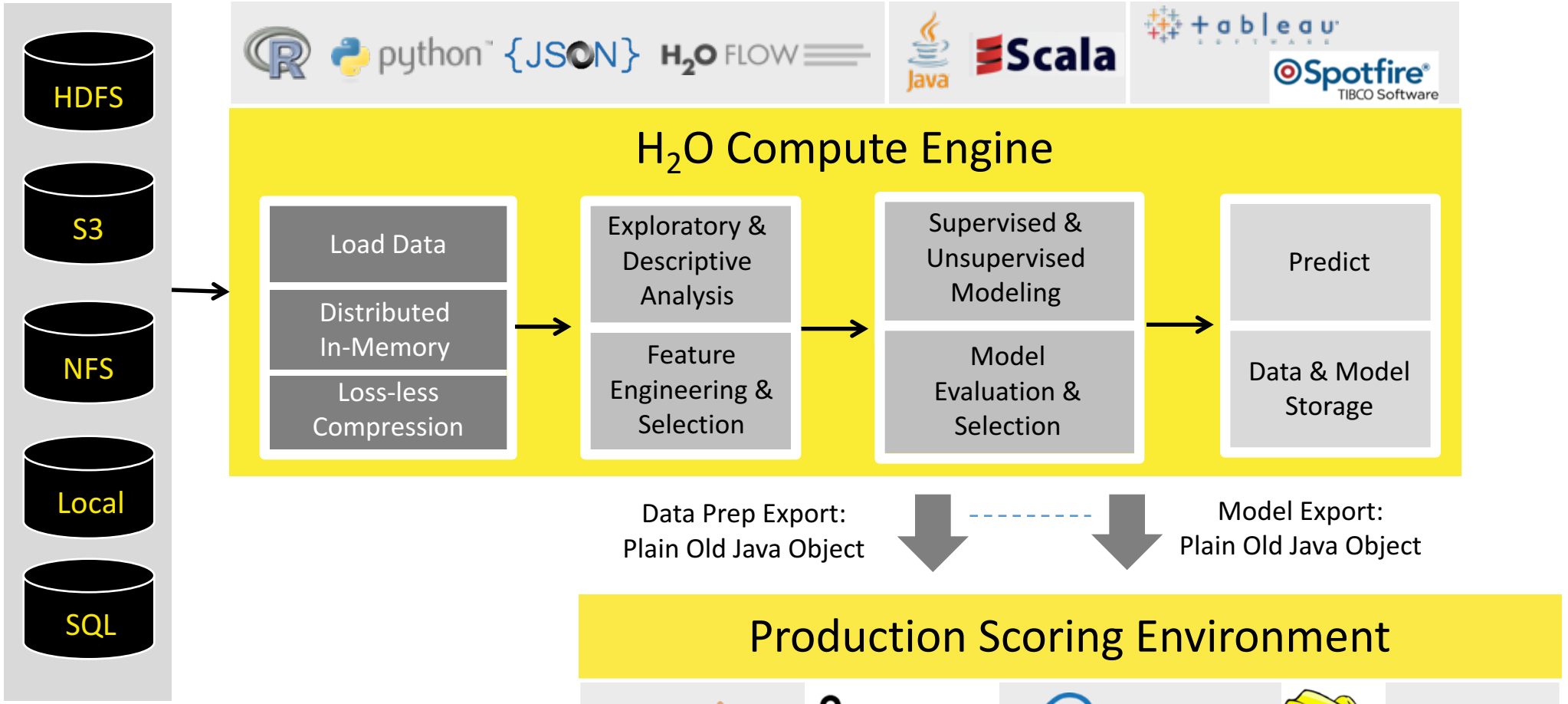
H2O AI Open Source Engine Integration with Spark

Lightning Fast machine learning on GPUs

Automatic feature engineering, machine learning and interpretability

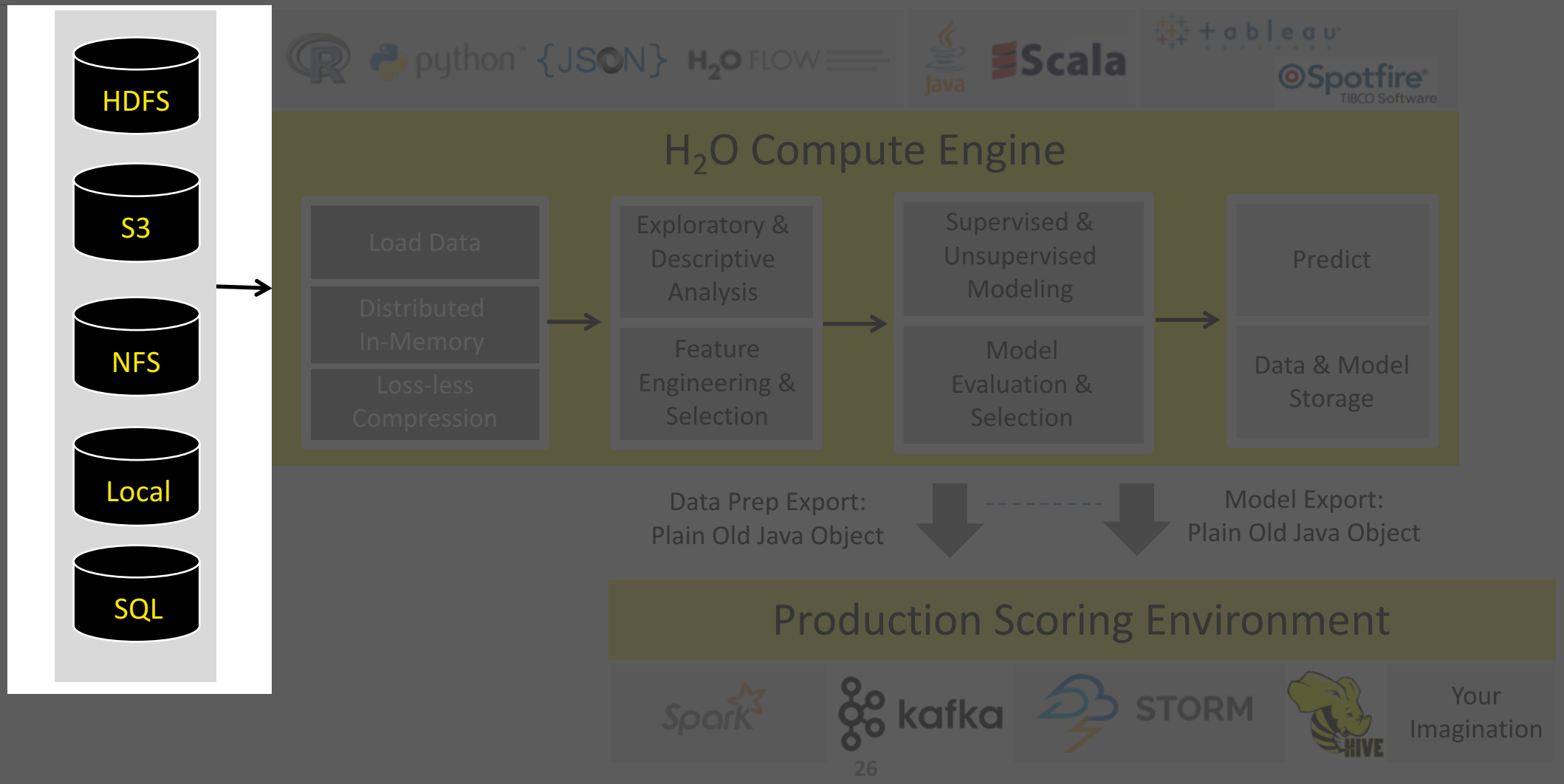
Secure multi-tenant H2O clusters

High Level Architecture



High Level Architecture

Import Data from Multiple Sources

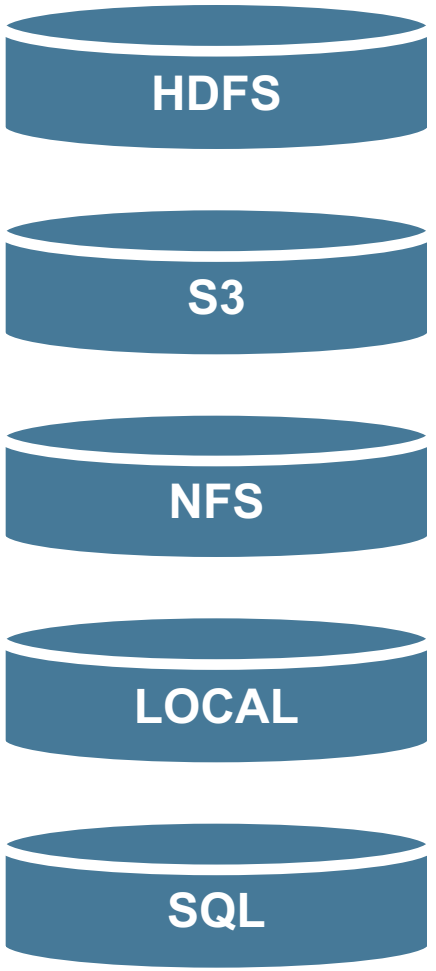


Supported Formats & Data Sources

9 Formats

File type or Folder of Files

- CSV
- XLS
- XLSX
- ORC*
- Hive*
- SVMLight
- ARFF
- Parquet
- Avro 1.8.0*



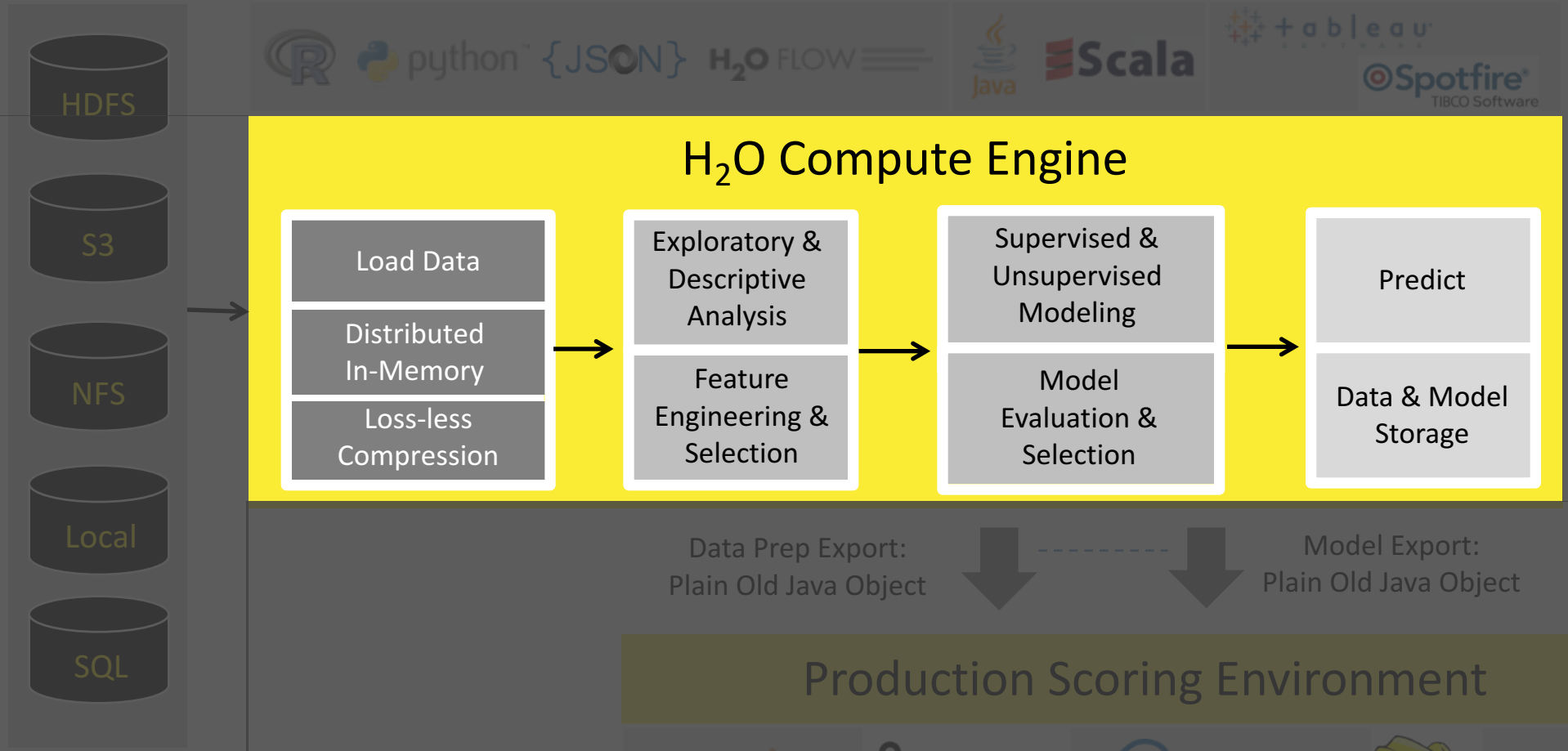
5 Sources

* 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

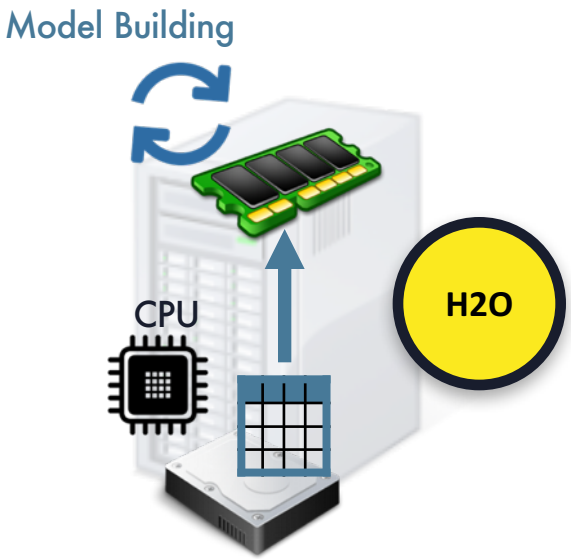
H2O.ai

High Level Architecture

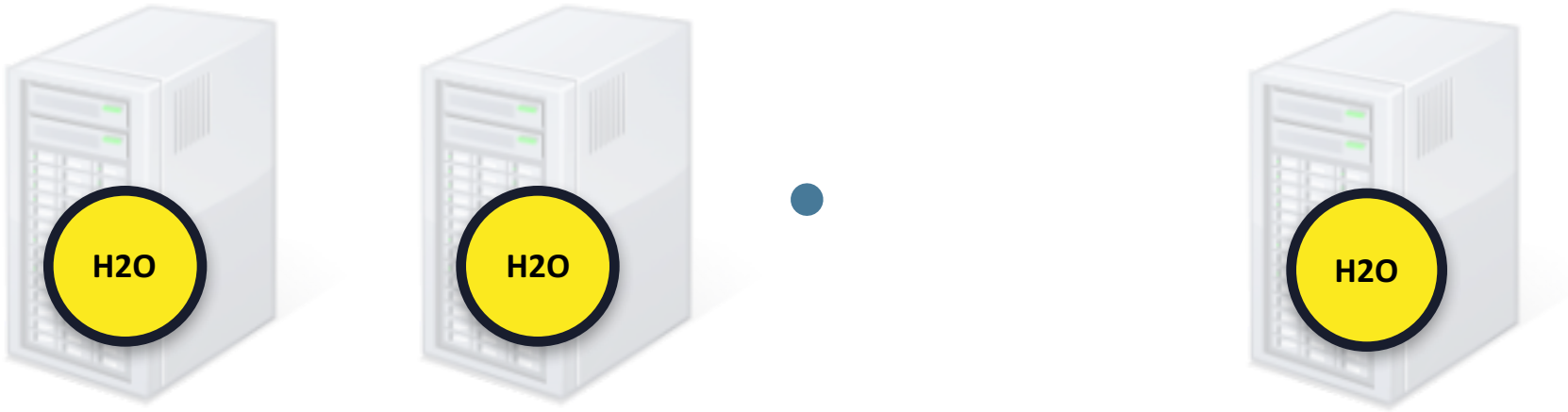
Fast, Scalable & Distributed Compute Engine Written in Java



H_2O Core

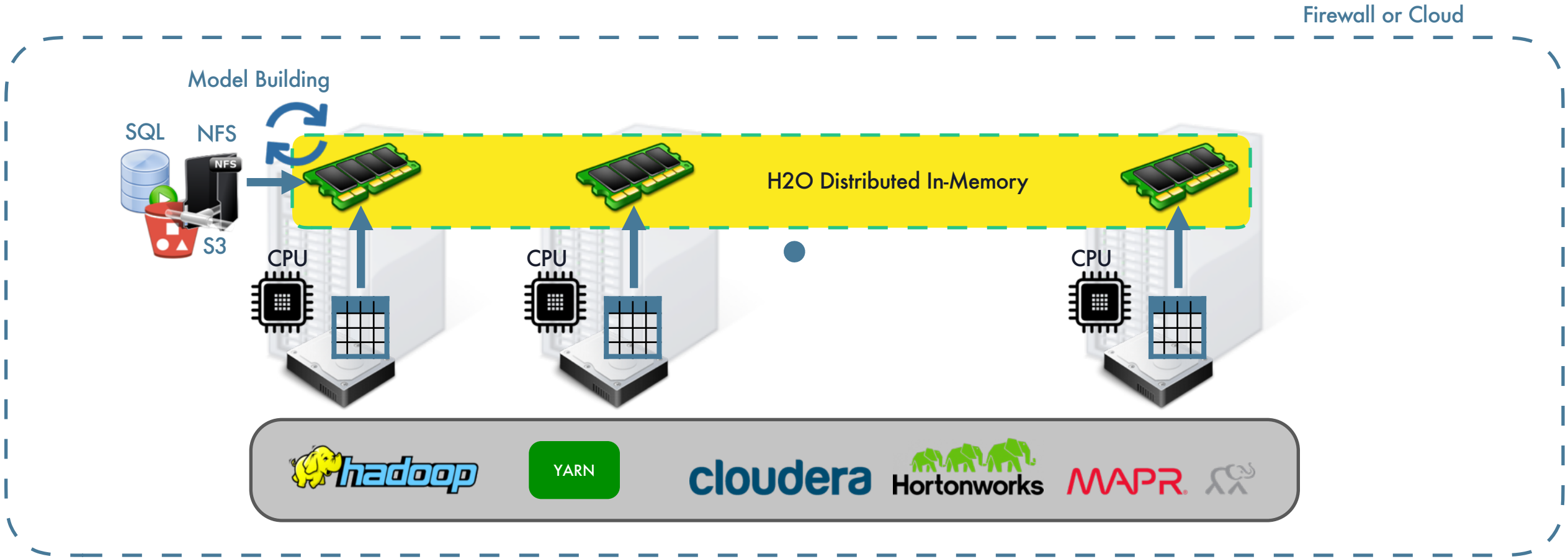


H_2O Core



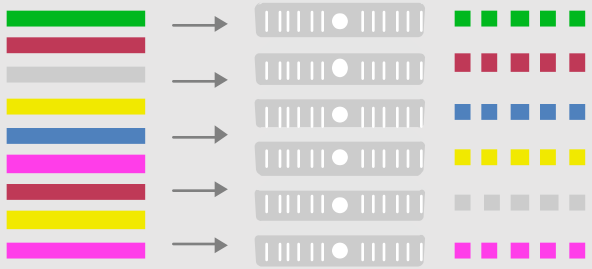
H_2O Core

H_2O Core

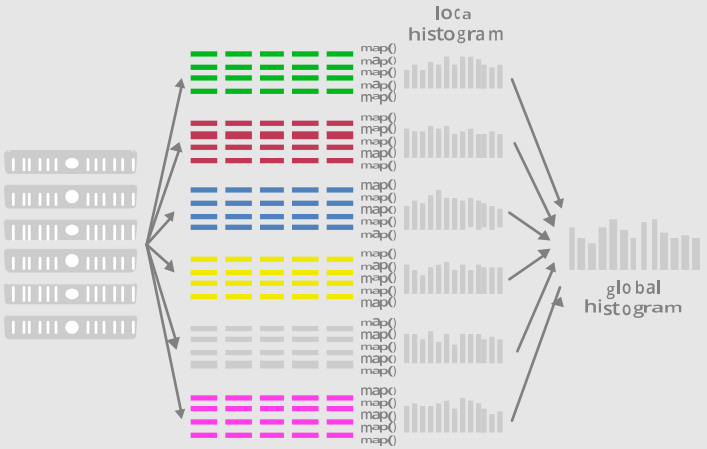

Firewall or Cloud
Model Building
SQL
NFS
NFS
S3
H2O Distributed In-Memory
CPU
CPU
CPU
hadoop
YARN
cloudera
Hortonworks
MAPR

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 H_2O: 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**

H_2O-3 Algorithms Overview

Supervised Learning

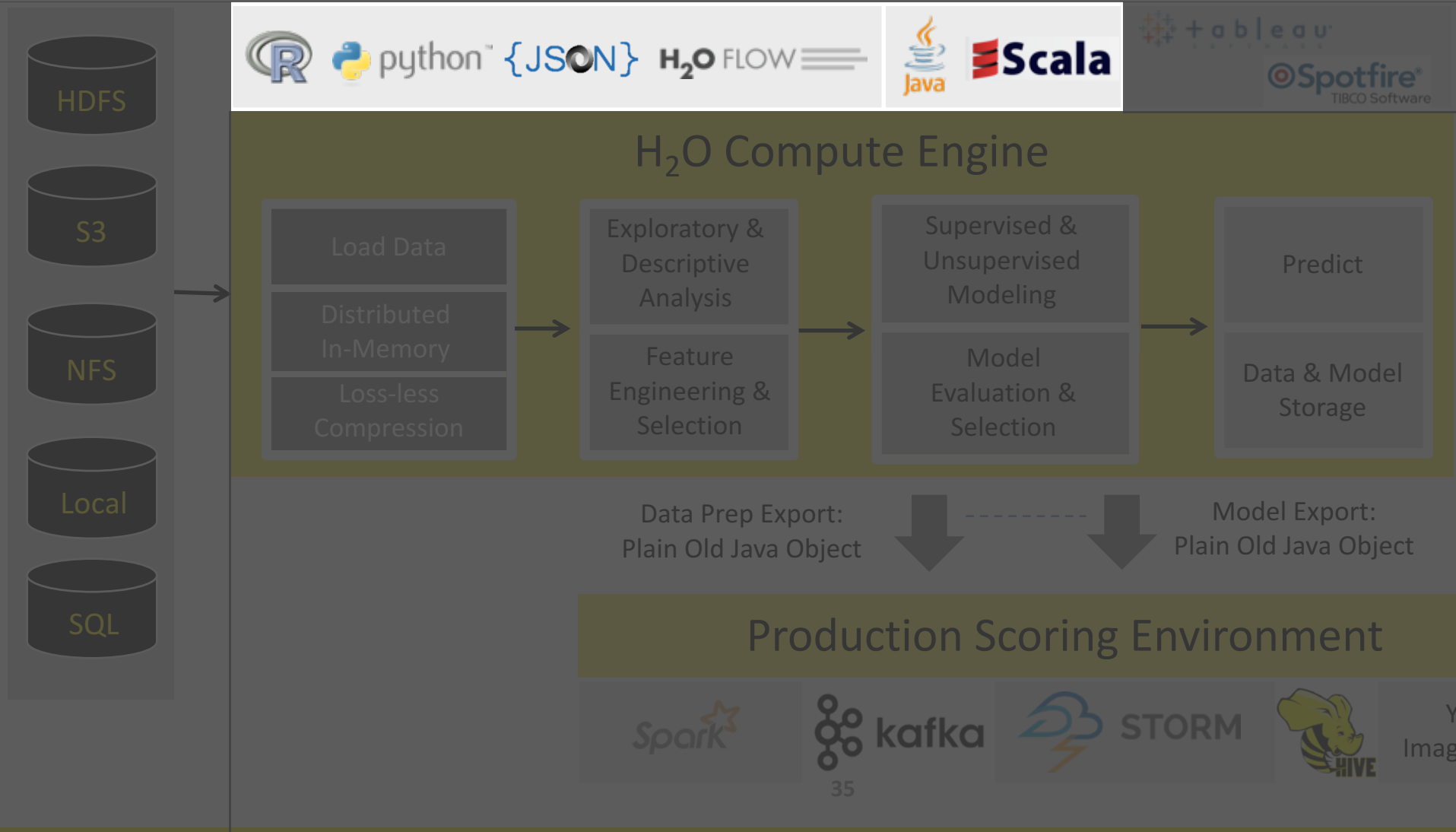
Statistical Analysis	• **Generalized Linear Models**: Binomial, Gaussian, Gamma, Poisson and Tweedie • **Naïve Bayes**
Ensembles	• **Distributed Random Forest**: Classification or regression models • **Gradient Boosting Machine**: Produces an ensemble of decision trees with increasing refined approximations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

Unsupervised Learning

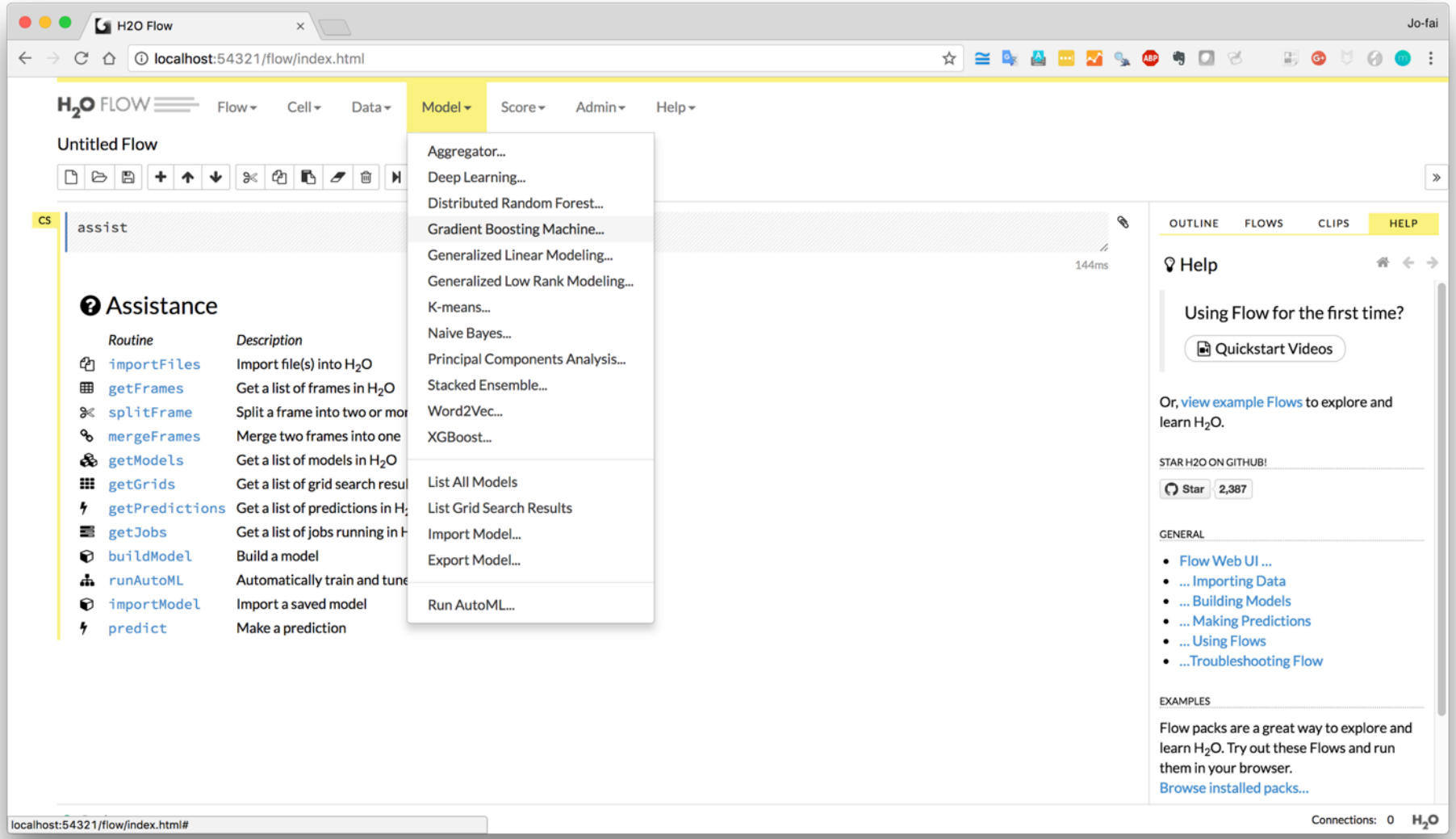
Clustering	• **K-means**: Partitions observations into k clusters/groups of the same spatial size. Automatically detect optimal k
Dimensionality Reduction	• **Principal Component Analysis**: Linearly transforms correlated variables to independent components • **Generalized Low Rank Models:** extend the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

High Level Architecture

Multiple Interfaces



H_2O Flow (Web) – First Demo



H2O.ai

Using H_2O with R and Python – Second Demo

~/Documents/repo_h2o/sales-engineering - master - RStudio Source Editor

credit_card_example.R

```r
# Credit Card Example

# Datasets:
# https://s3.amazonaws.com/h2o-training/credit_card/credit_card_train.csv
# https://s3.amazonaws.com/h2o-training/credit_card/credit_card_test.csv

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

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

# Look at datasets
summary(df_train)
summary(df_test)

# Define features and target
features = colnames(df_test)
target = "DEFAULT_PAYMENT_NEXT_MONTH"

# Train a GBM model
model_gbm = h2o.gbm(x = features,
                    y = target,
                    training_frame = df_train,
                    seed = 1234)
print(model_gbm)

# Use GBM model for making predictions
yhat_test = h2o.predict(model_gbm, newdata = df_test)
head(yhat_test)

# (Extra) Use H2O's AutoML
aml = h2o.automl(x = features,
                 y = target,
                 training_frame = df_train,
                 max_runtime_secs = 60,
                 seed = 1234)

# Print leaderboard
print(aml@leaderboard)

# Use best model for making predictions
best_model = aml@leader
yhat_test = h2o.predict(best_model, newdata = df_test)
head(yhat_test)
```

credit_card_example

localhost:8888/notebooks/credit_card_example.ipynb

jupyter credit_card_example Last Checkpoint: 5 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Help Python 2

Code CellToolbar

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/p7yt7_4n4fj1jlyq6g4qhfbw0000gn/T/tmpFdP3Av
  JVM stdout: /var/folders/4z/p7yt7_4n4fj1jlyq6g4qhfbw0000gn/T/tmpFdP3Av/h2o_jofaichow_started_from_python.out
  JVM stderr: /var/folders/4z/p7yt7_4n4fj1jlyq6g4qhfbw0000gn/T/tmpFdP3Av/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_id7iqa
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%
```

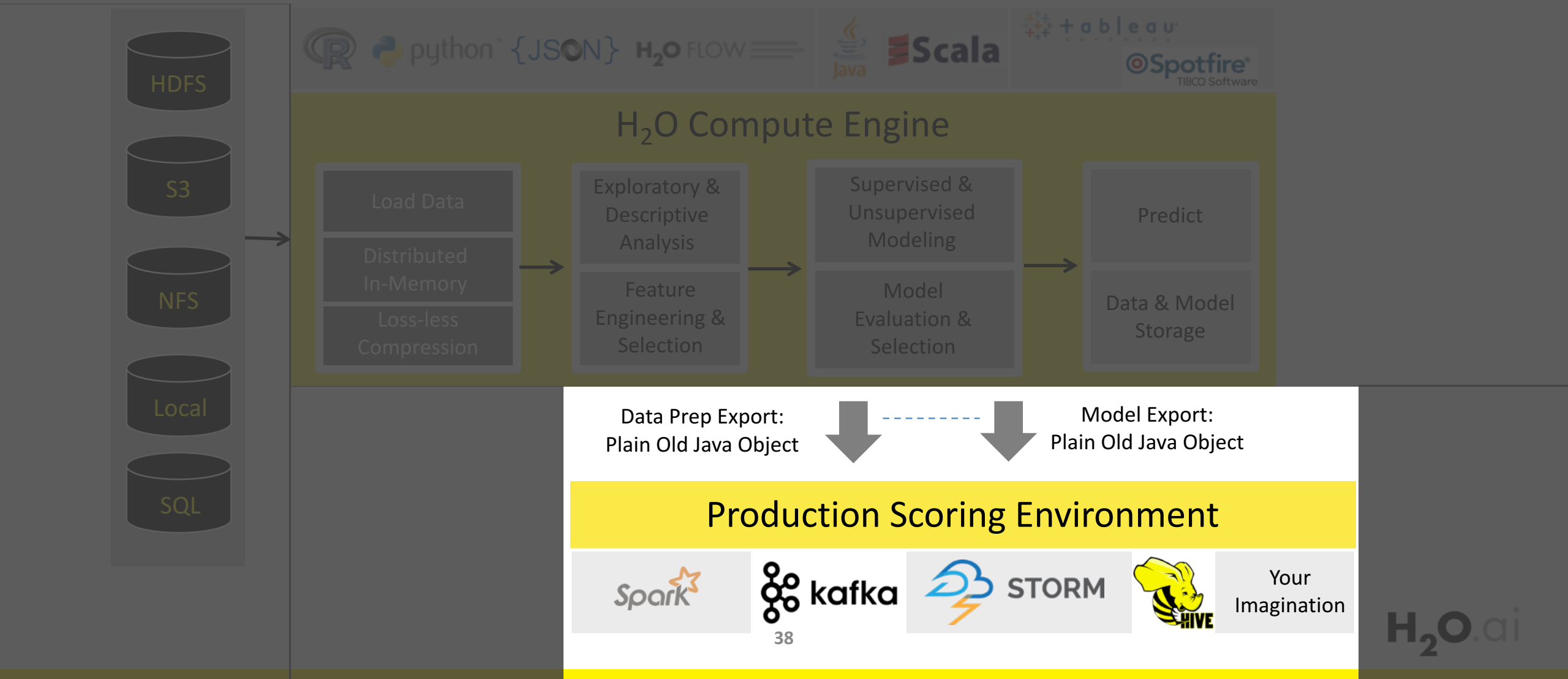
In [4]:
```
# Look at datasets
df_train.summary()
df_test.summary()
```

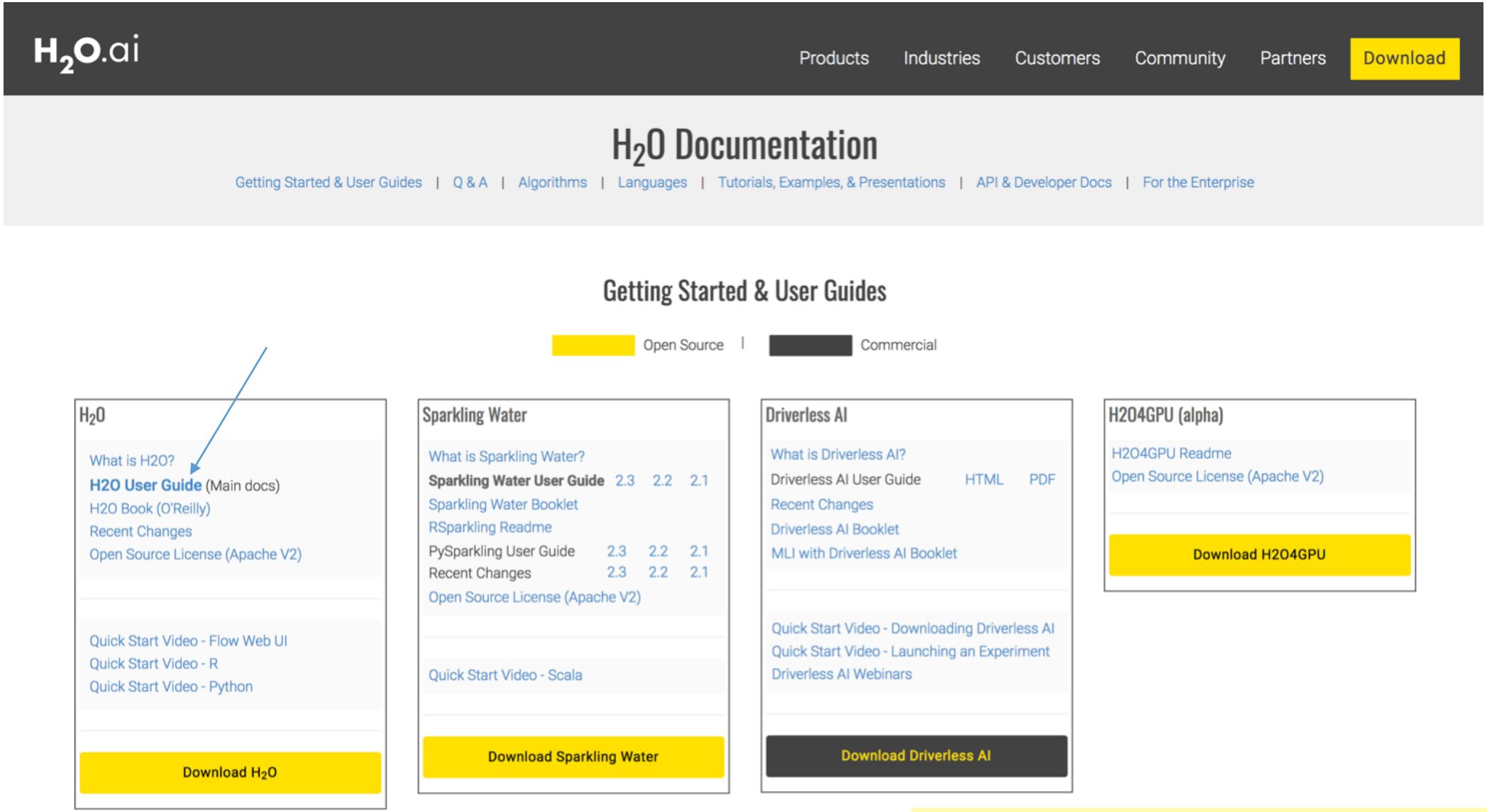
	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4
type	int	enum	int	int	int	int	int	int	int
mins	10000.0		0.0	0.0	21.0	-2.0	-2.0	-2.0	-2.0
mean	165471.466667		1.85	1.55578703704	35.4053240741	-0.00523148148148	-0.122361111111	-0.15537037037	-0.21060
maxs	1000000.0		6.0	3.0	79.0	8.0	8.0	8.0	8.0
sigma	128853.314839		0.779559696278	0.522505078476	9.27675421641	1.12668964211	1.20086854503	1.20727030901	1.17217
zeros	0		9	37	0	10563	11284	11309	11905
missing	0	0	0	0	0	0	0	0	0

H2O.ai

High Level Architecture

Export Standalone Models for Production





URL: **docs.h2o.ai**

H2O.ai

Demo:
H_2O on a 320-Core Hadoop Cluster

(Web Interface)

Higgs challenge

Higgs Boson Machine Learning Challenge

Use the ATLAS experiment to identify the Higgs boson

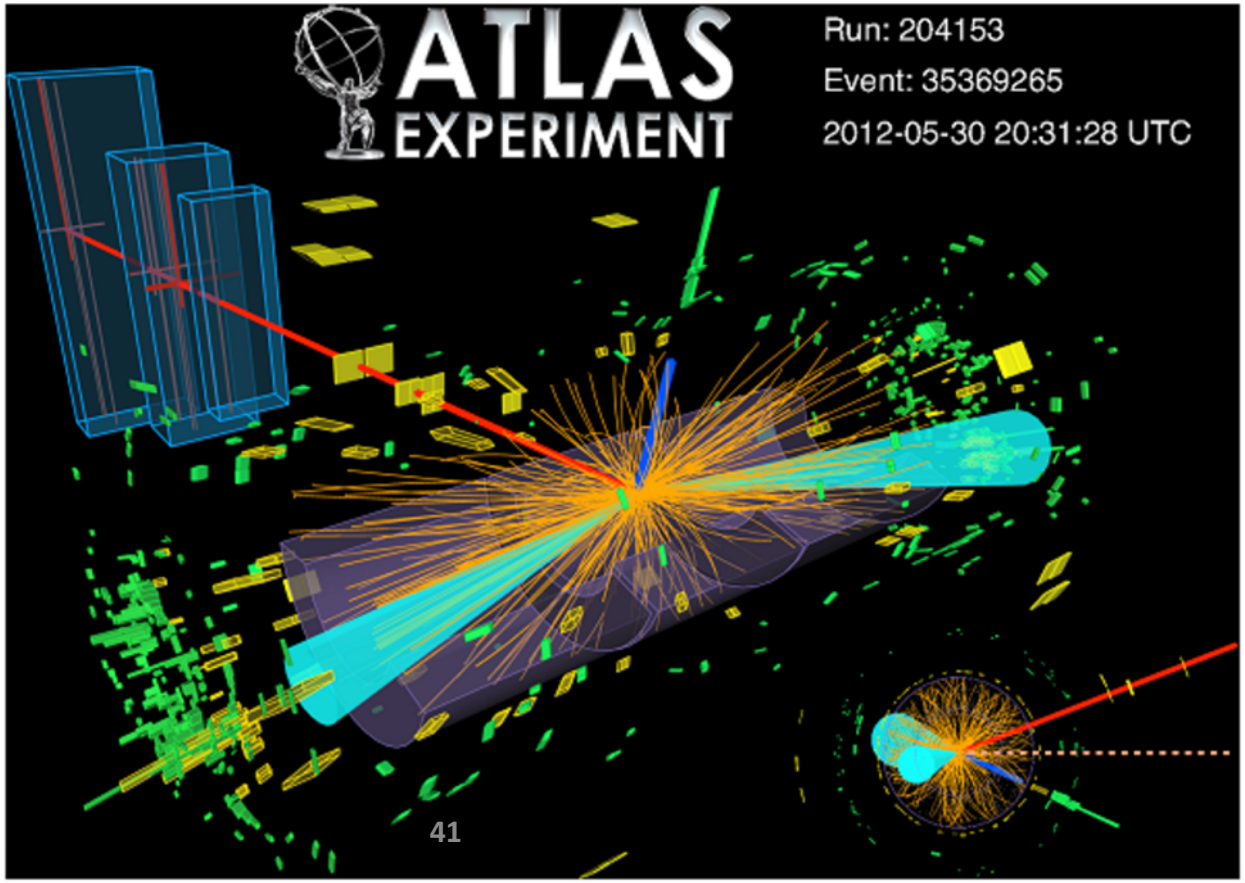
$13,000 · 1,785 teams · 3 years ago

Overview | Data | Discussion | Leaderboard | Rules | Team | My Submissions | Late Submission

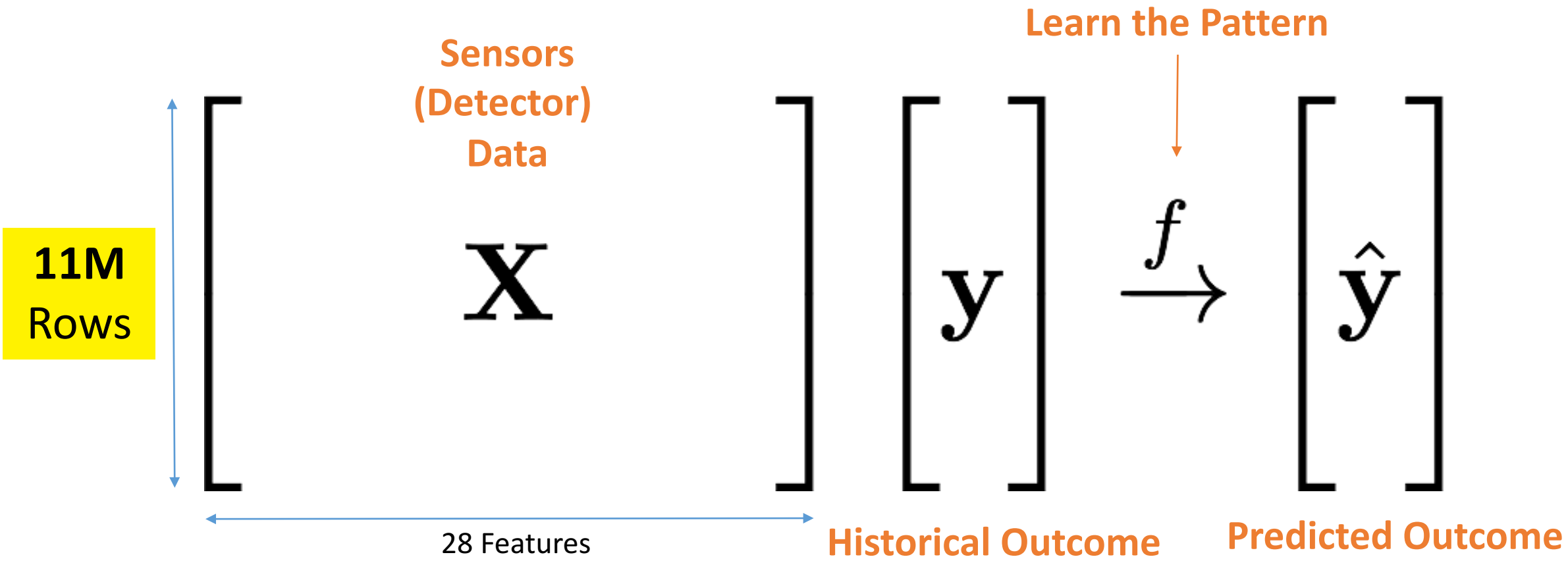
https://www.kaggle.com/c/higgs-boson

Overview

- Description
- Evaluation
- Prizes
- About The Sponsors
- Timeline
- Winners



Learning from Higgs Boson Machine Data



Raw Data Size: **7.48 GB**

11M Rows

Size (Raw): **7.48 GB**
Compressed: **2.00 GB** (≈ 27% of Raw)

HIGGS.hex

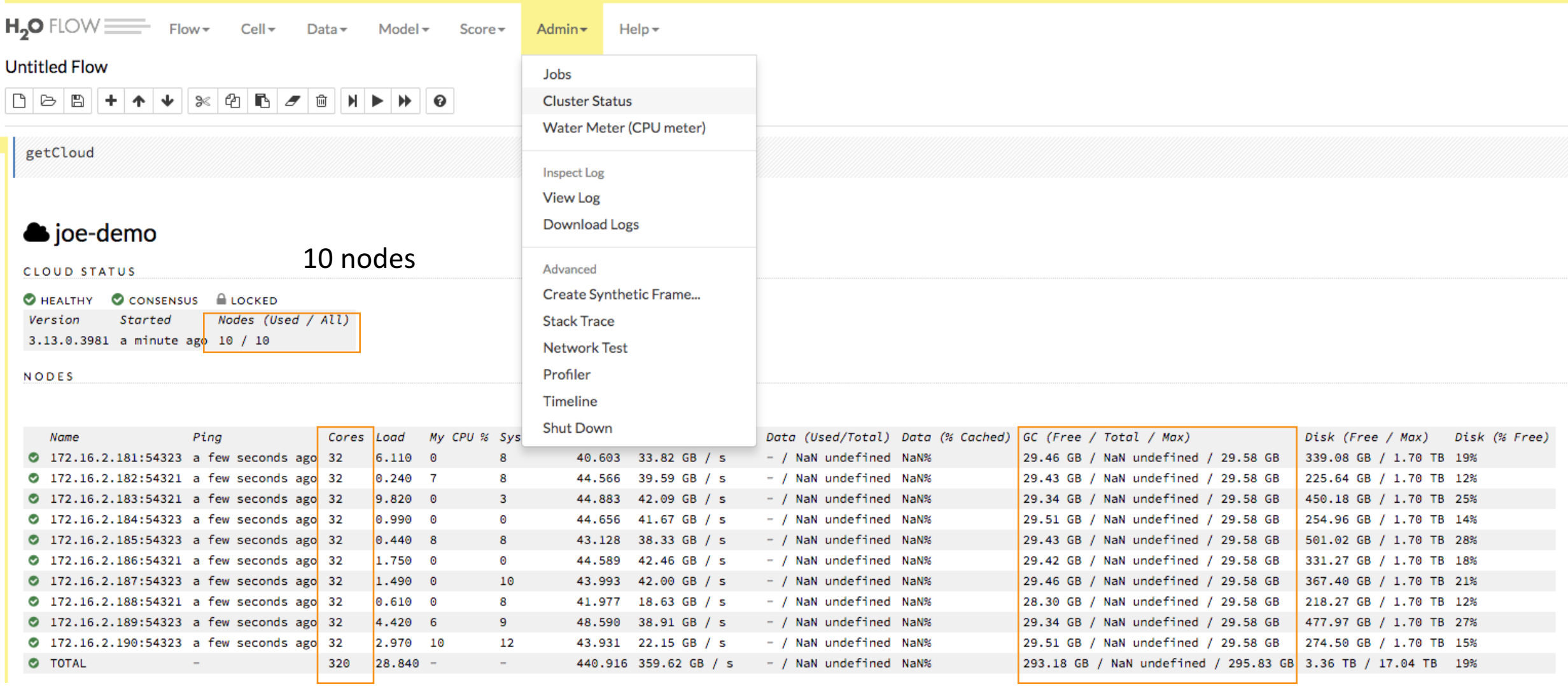
Actions: View Data | Split... | Build Model... | Predict | Download | Export

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	·	·

H2O.ai

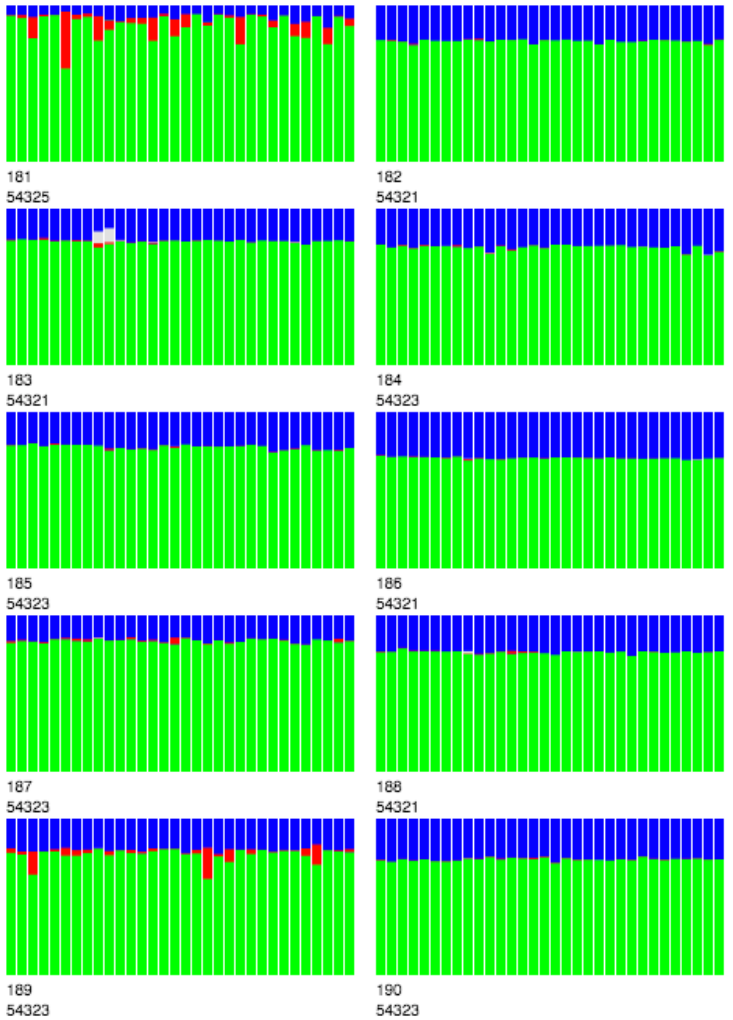


10 x 32 =
320 Cores

10 x 29.6 = 296
GB Memory

H_2O Water Meter (CPU Monitor)

10 x 32 = 320 Cores

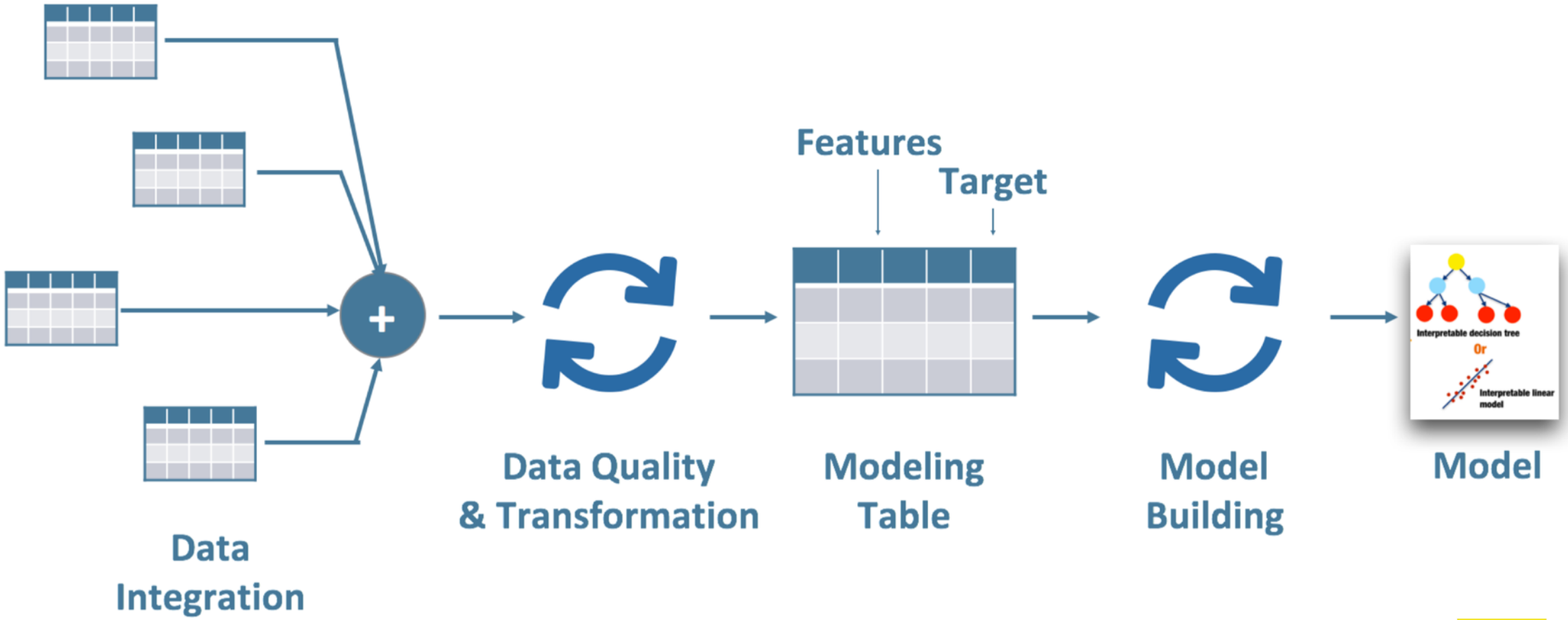


Demo: AutoML

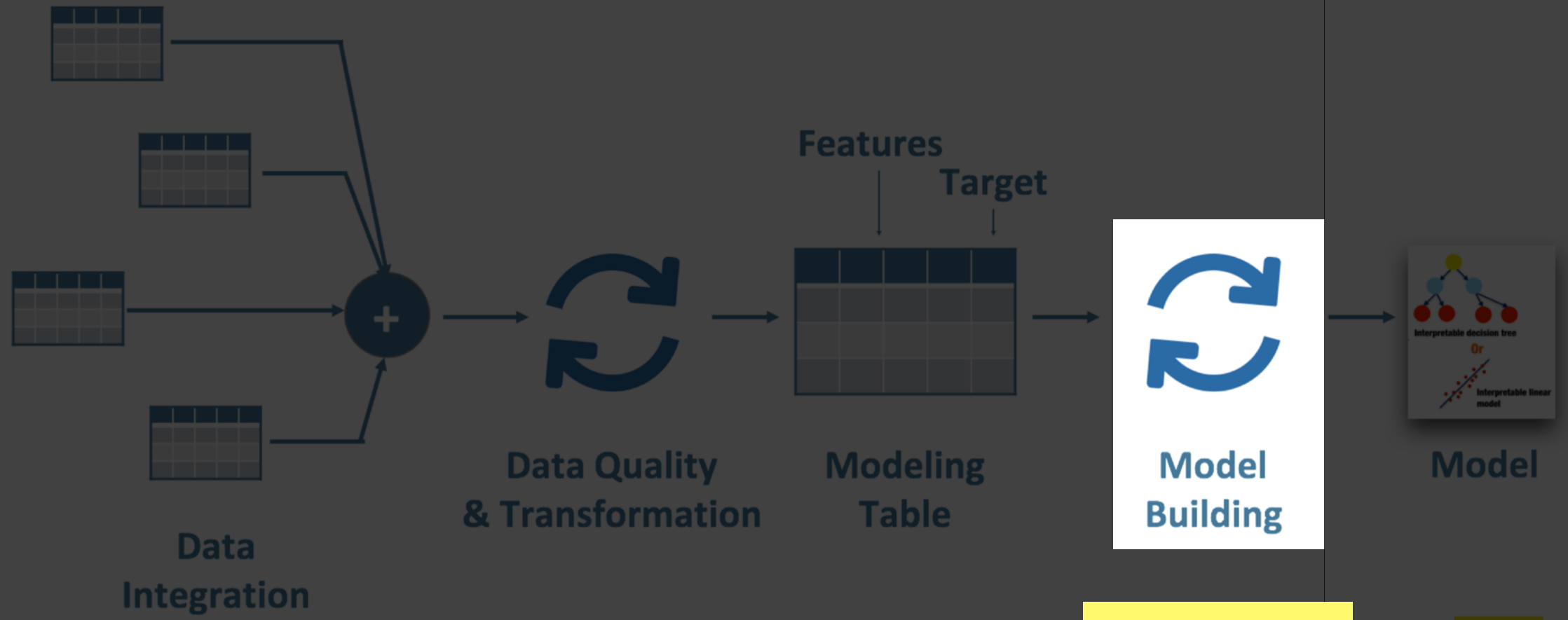
Automatic Machine Learning with H_2O

(R Interface)

Typical Enterprise Machine Learning Workflow



Typical Enterprise Machine Learning Workflow



AutoML

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

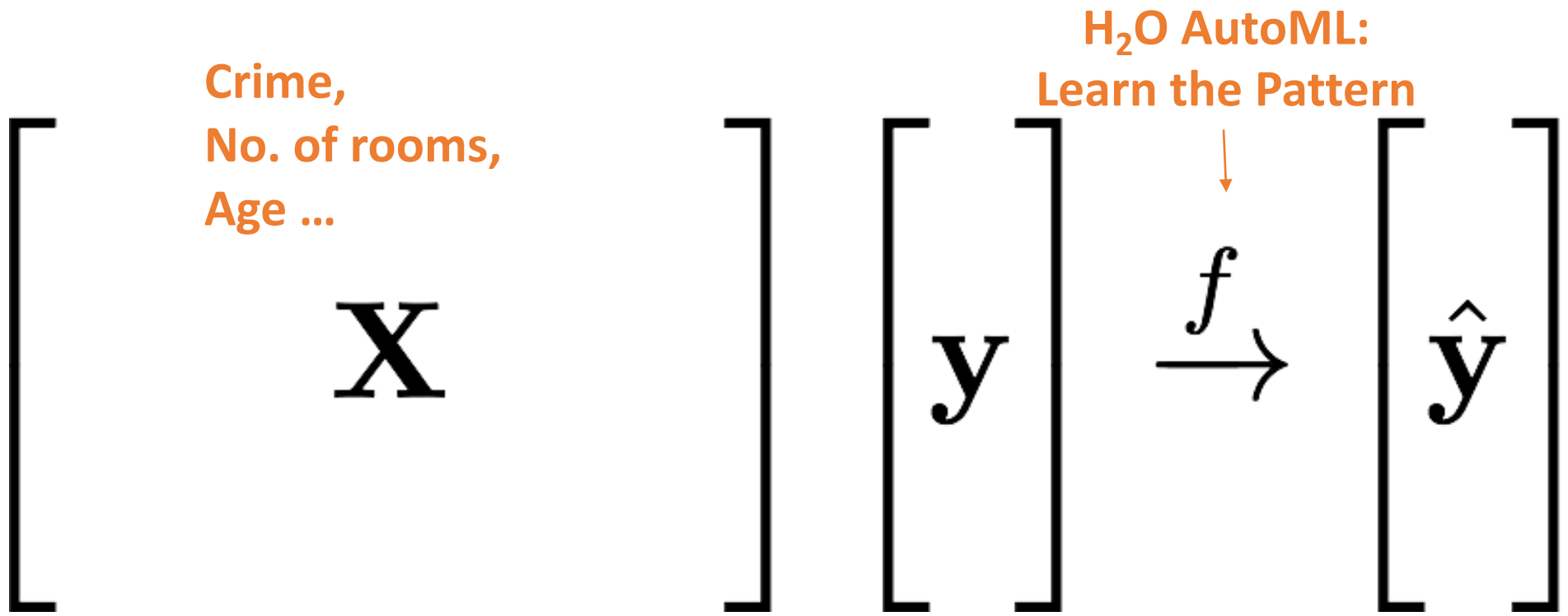
```
library(mlbench) # for dataset
data("BostonHousing")
dim(BostonHousing)
```

```
## [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

```r
# Start a local H2O cluster (JVM)
library(h2o)
h2o.init(nthreads = -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:
## ==============
##
## H2ORegressionModel: stackedensemble
## Model ID:  StackedEnsemble_BestOfFamily_0_AutoML_20180221_015244
## NULL
##
##
## H2ORegressionMetrics: stackedensemble
## ** Reported on training data. **
##
## MSE:  0.8307283
## RMSE:  0.911443
## MAE:  0.6676121
## RMSLE:  0.04544841
## Mean Residual Deviance :  0.8307283
##
##
## H2ORegressionMetrics: 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 H_2O News

Latest Developments

Events

H_2O 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

Automatic feature engineering, machine learning and interpretability

Secure multi-tenant H2O clusters

Algorithms on H_2O-3 (CPU)

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

Algorithms on H_2O4GPU (more to come)

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

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 H_2O 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:

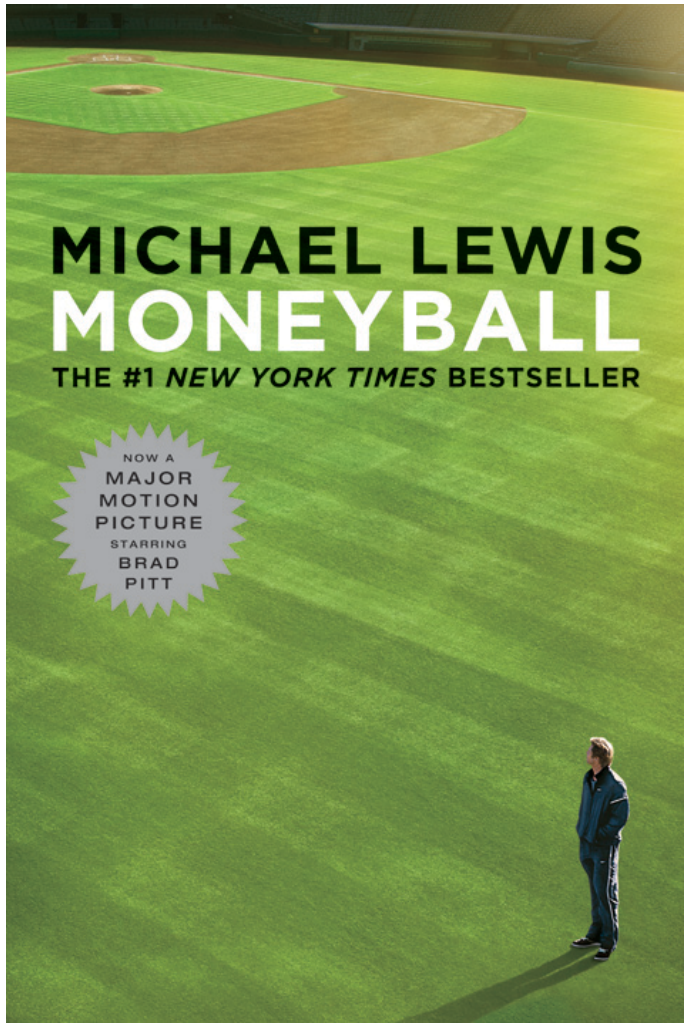
You do not ask me about the names of team and player involved.

(… 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

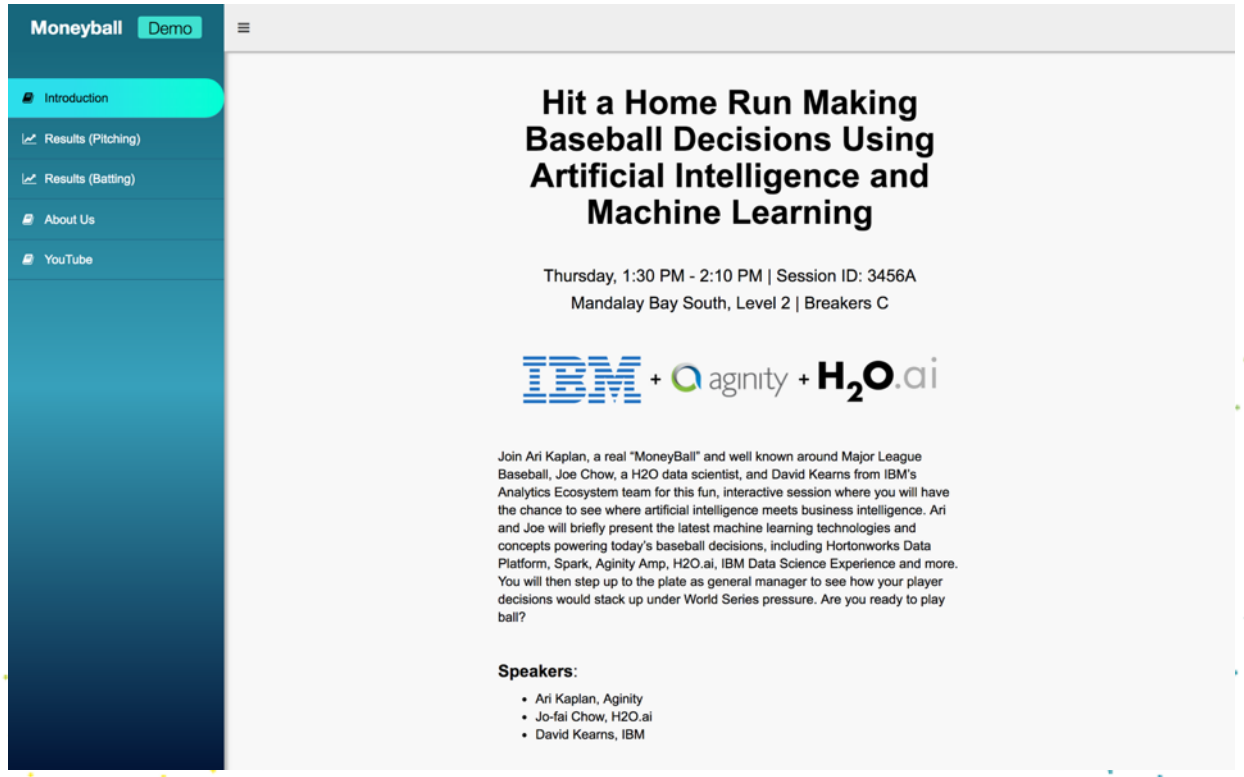
Peter Brand
(based on Paul DePodesta)

H2O.ai

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



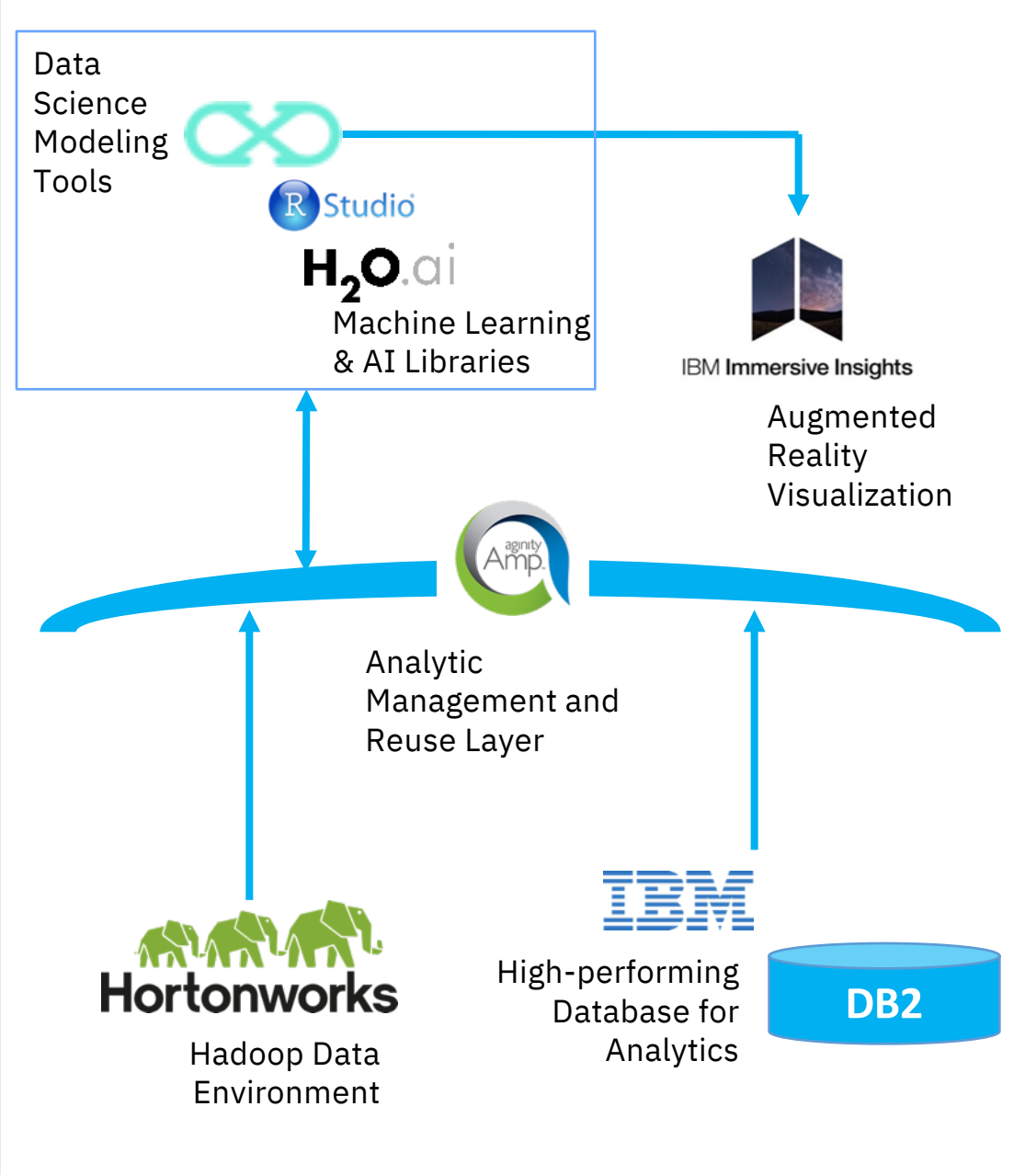
H2O.ai

The Architecture

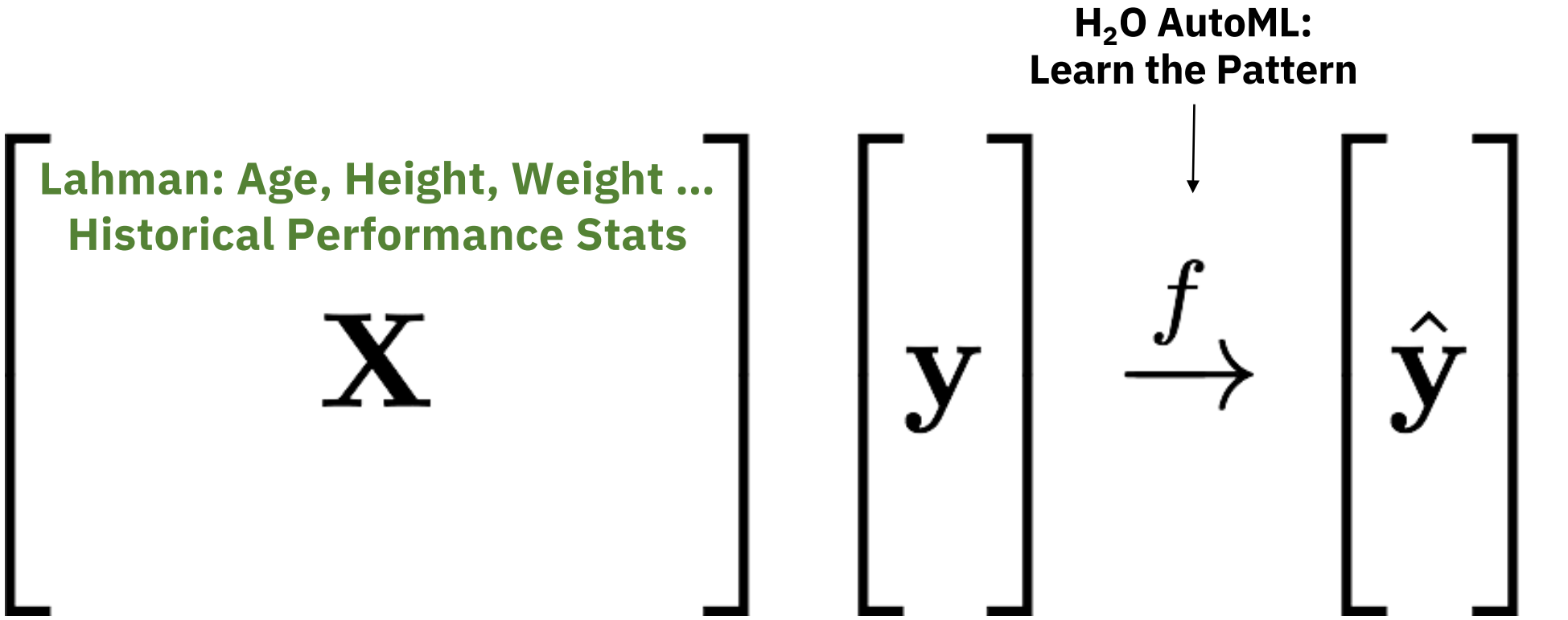
Enterprise Solution

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 H_2O
4. DSX and H_2O 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



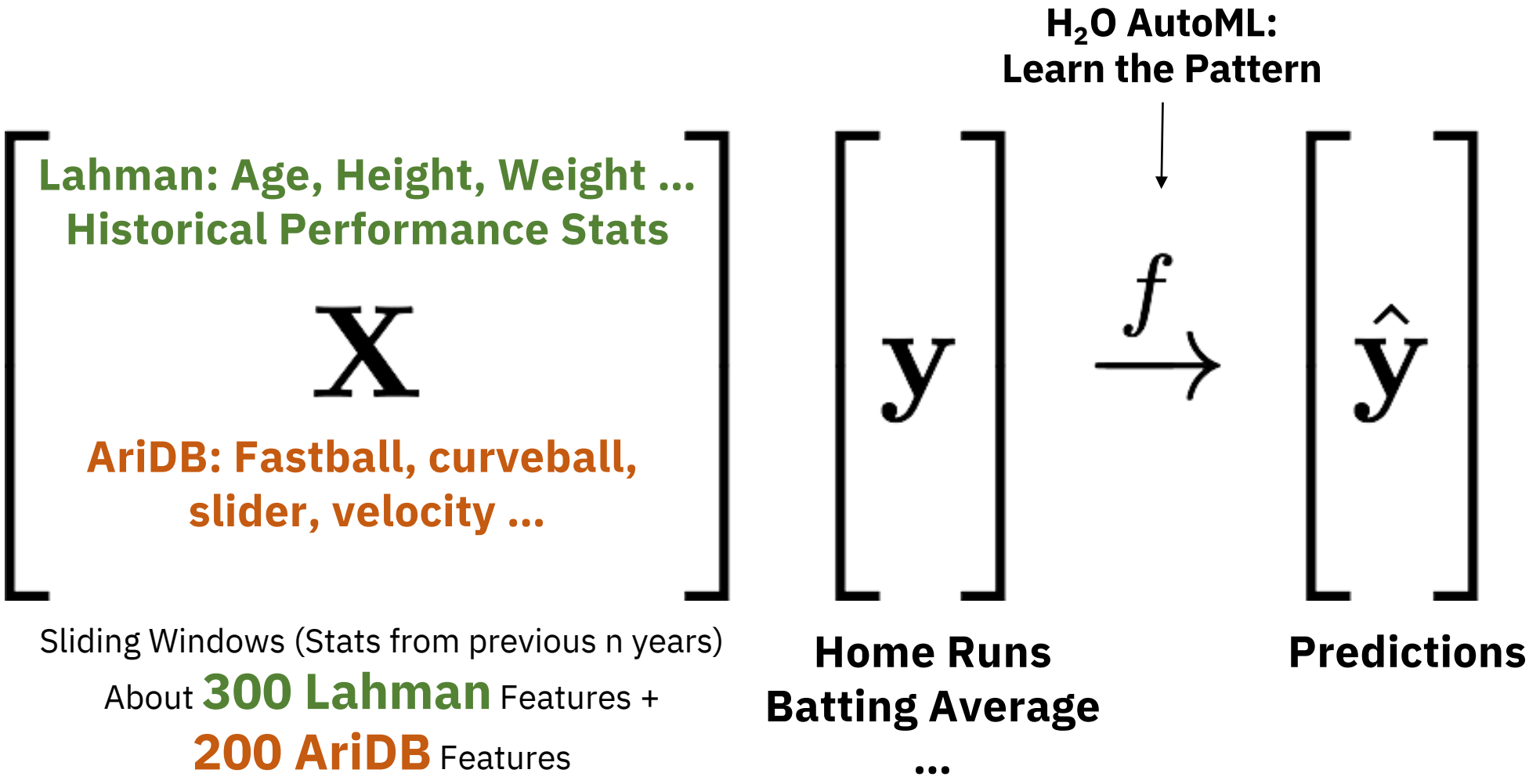
Sliding Windows (Stats from previous n years)

About **300 Lahman** Features

Home Runs
Batting Average
...

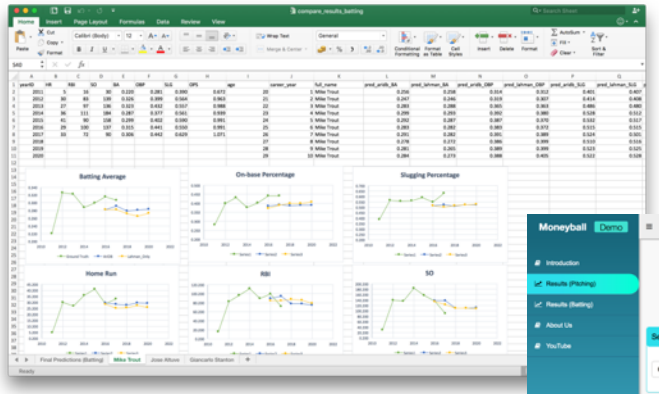
Predictions

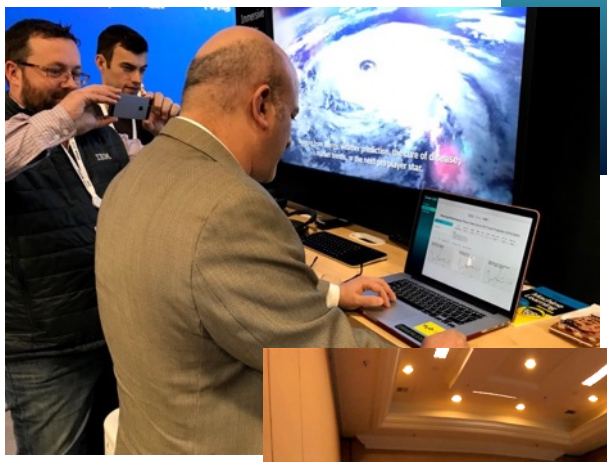
Approach Two: Learning from **Lahman** & **AriDB**



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

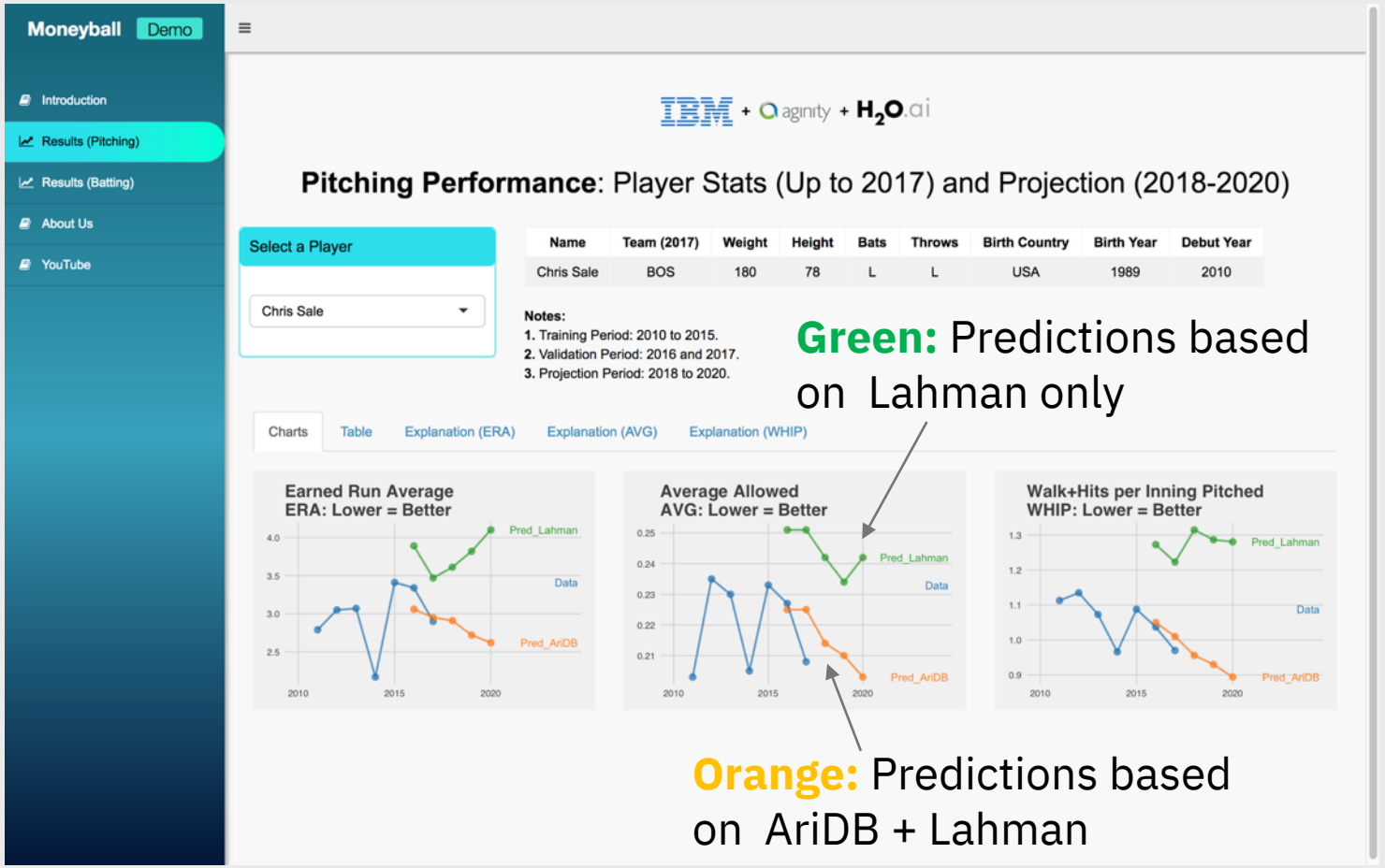


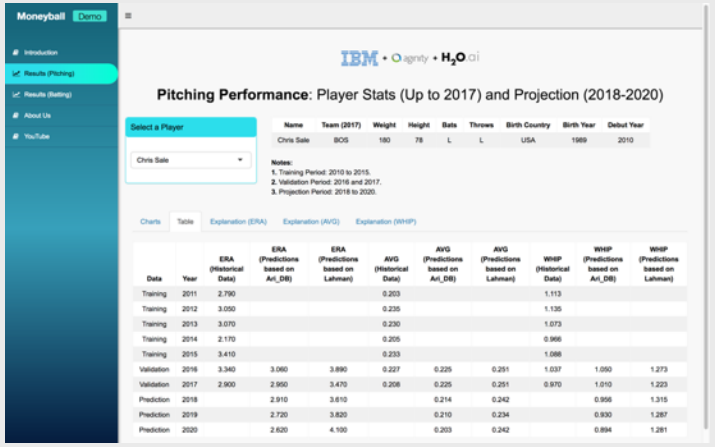


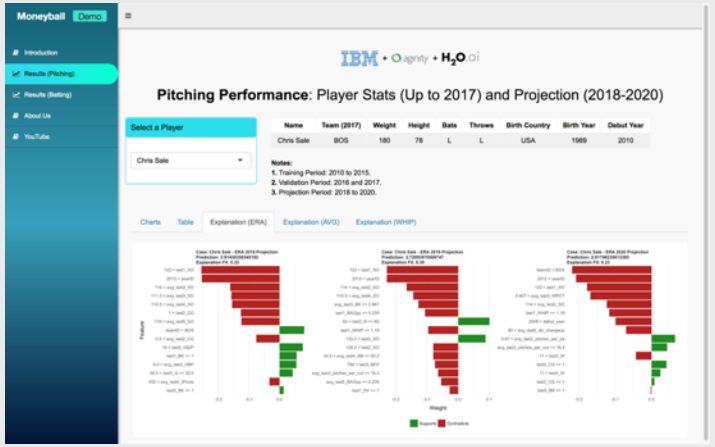
H2O.ai

Presentation

Shiny App







Acknowledgement

H2O.ai

Merci beaucoup!

- Organisers & Sponsors
  - Alexia Audevart
  - Christophe Regouby
  - HarryCow Coworking
- H_2O'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
  - github.com/woobe
- Please search/ask questions on **Stack Overflow**
  - Use the tag `**h2o**` (not h2 zero)