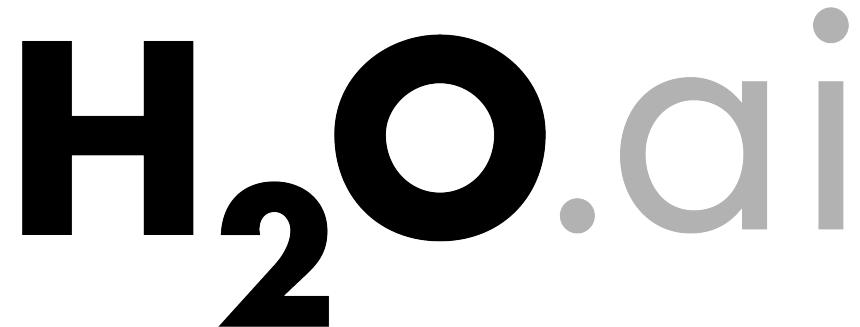


Introduction to Machine Learning with H₂O

- Introduction (20 Mins)
 - Company
 - H₂O Machine Learning Platform
- First Use Case (15 Mins)
 - Credit Card Dataset (Small CSV)
 - H₂O Quick Start (Web/R/Python)
- Second Use Case (20 Mins)
 - Freddie Mac Dataset (HDFS – 2.58 GB)
 - Higgs Boson Machine (HDFS – 7.48GB)
 - H₂O on Multi-Node Cluster
- Q & A (5 Mins)



Jo-fai (Joe) Chow
Data Scientist at H2O.ai
joe@h2o.ai

About H₂O.ai

Company Overview

Founded	2011 Venture-backed, debuted in 2012
Products	<ul style="list-style-type: none">• H₂O Open Source In-Memory AI Prediction Engine• Sparkling Water• Deep Water• Steam
Mission	Operationalize Data Science, and provide a platform for users to build beautiful data products
Team	<p>70 employees</p> <ul style="list-style-type: none">• Distributed Systems Engineers doing Machine Learning• World-class visualization designers
Headquarters	Mountain View, CA



H₂O.ai



Amy Vu-Tran



Amy Wang



Angela Bartz



Anmol Bal



Anna Candel



Aulrich Barthas



Maral Mandjarijan



Mark Chan



Mark Landry



Miroslava Dymotsky



Matt Dowle



Megan Kurkoski



Arkoch Chauhan



Ami Wadhwa



Beth Payne



Brandon Murray



Carl Andrews



Des Narayanan



Michal Kurko



Michal Molokhava



Navdeep Gill



Nidhi Mehta



Nikhil Shethkar



Nishant Kaleria



Daivid Chan



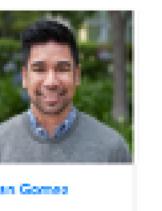
Dmitry Larko



Erin LaDell



Fonda Ingram



Ian Gomez



Jacqueline Scott



Pasha Serebrenko



Patrick Hall



Patrick Rice



Prithvi Prabhu



Ravi Purushotama



Raymond Peck



Jakub Hava



Jeff Fohl



Jeff Cambena



Jo-Hai Chow



Jon Olszewski



Josephine Wang



Sebastian Vidrio



Srikanth Ambati



Terence Ward



Tam Kraljevic



Tomas Nykodym



Venkatesh Yadav



Justin Loyola



Karen Hayrapetyan



Kimberly O'Shea



Lauren DiPenna



Leland Wilkinson



Magnus Stensmoen



Vinod Iyengar



Wien Pham



Wendy Wong



Joe (UK)

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*



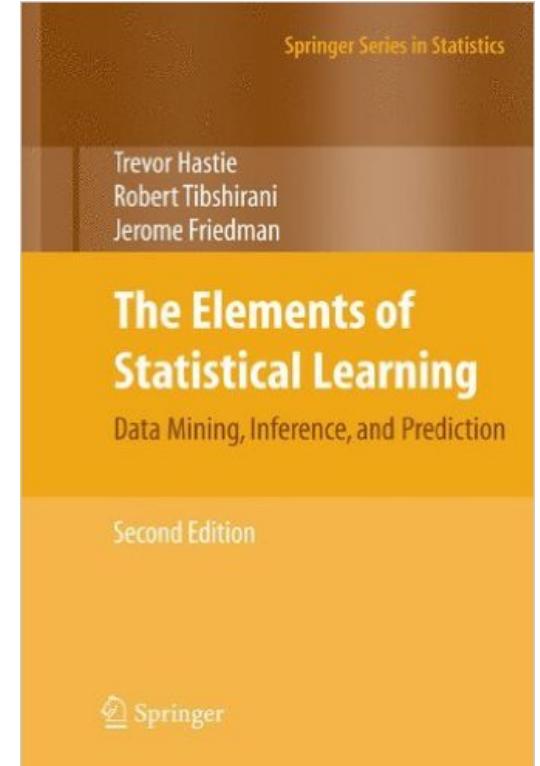
Dr. Robert Tibshirani

- 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*

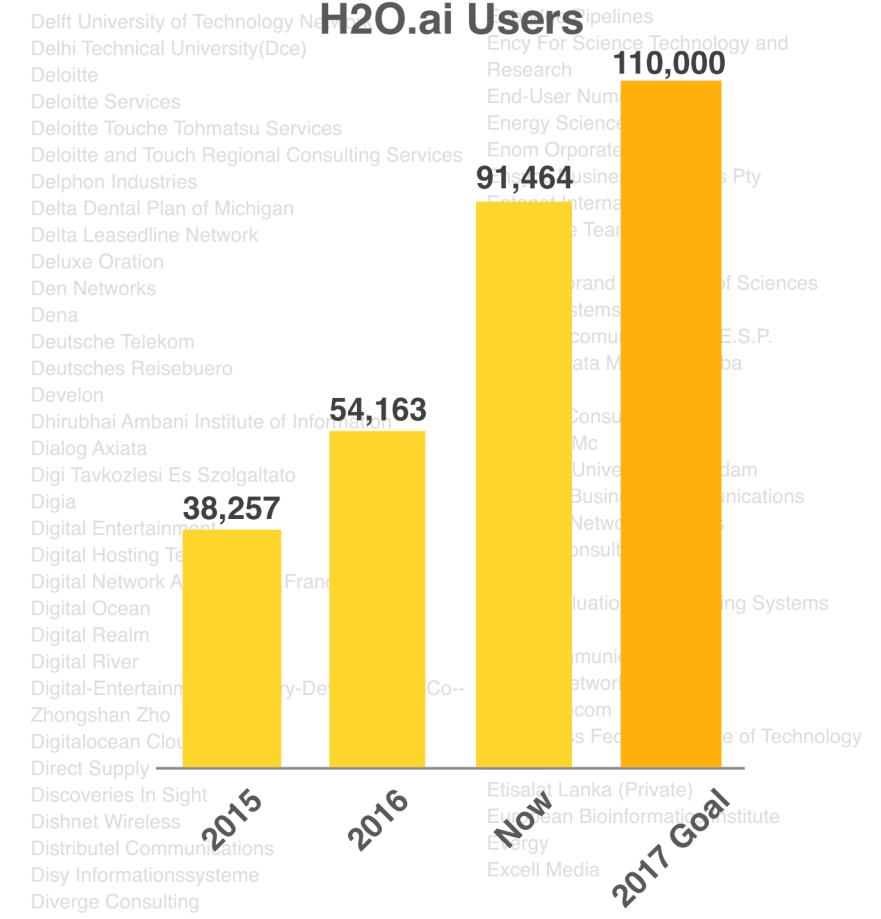
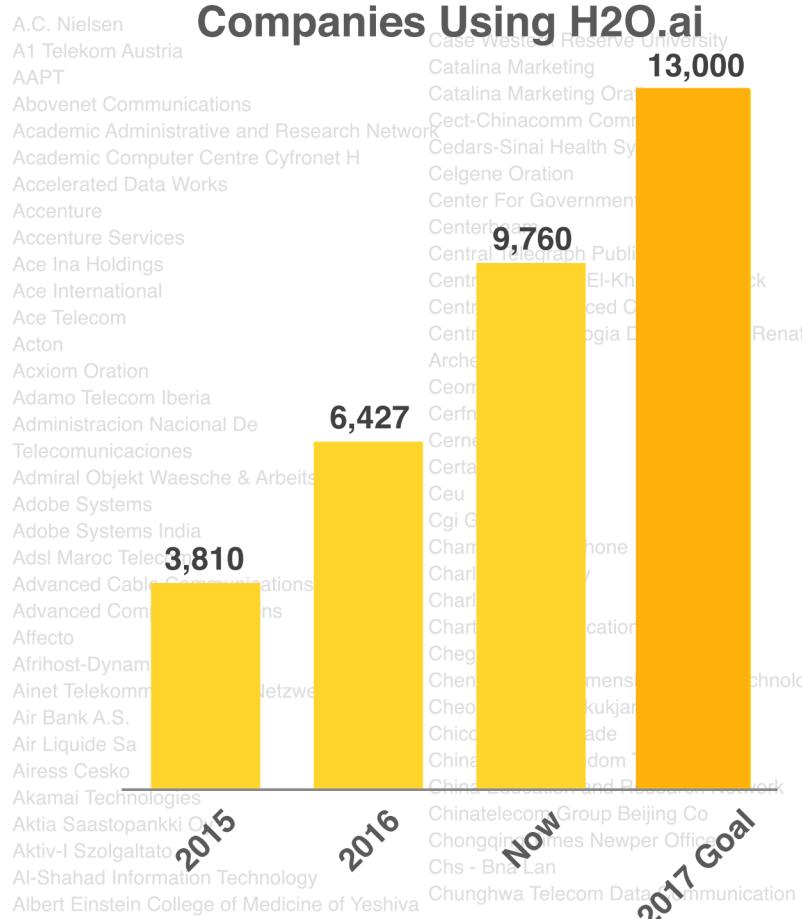


Dr. Steven Boyd

- 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 Community & Fortune 100 customers



Select Reference Customers:

"Overall customer satisfaction is very high." - Gartner

Select Customers



“Overall customer satisfaction is very high.” - Gartner

AI in Financial Services

Wholesale / Commercial Banking

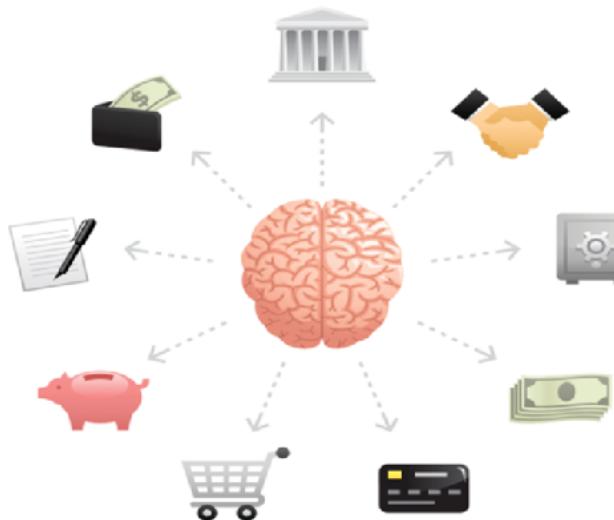
- Know Your Customers (KYC)
- Anti-Money Laundering (AML)

IT Infrastructure

- Security Cyberlake
- DoS Detection and Protection
- Master Data Management

Retail Banking

- Deposit Fraud
- Customer Churn Prediction
- Auto-Loan



Today's Use Case Examples

Card/Payments Business

- Transaction Frauds
- Real-time Targeting
- **Credit Risk Scoring**
- In-Context Promotion



Community Expansion

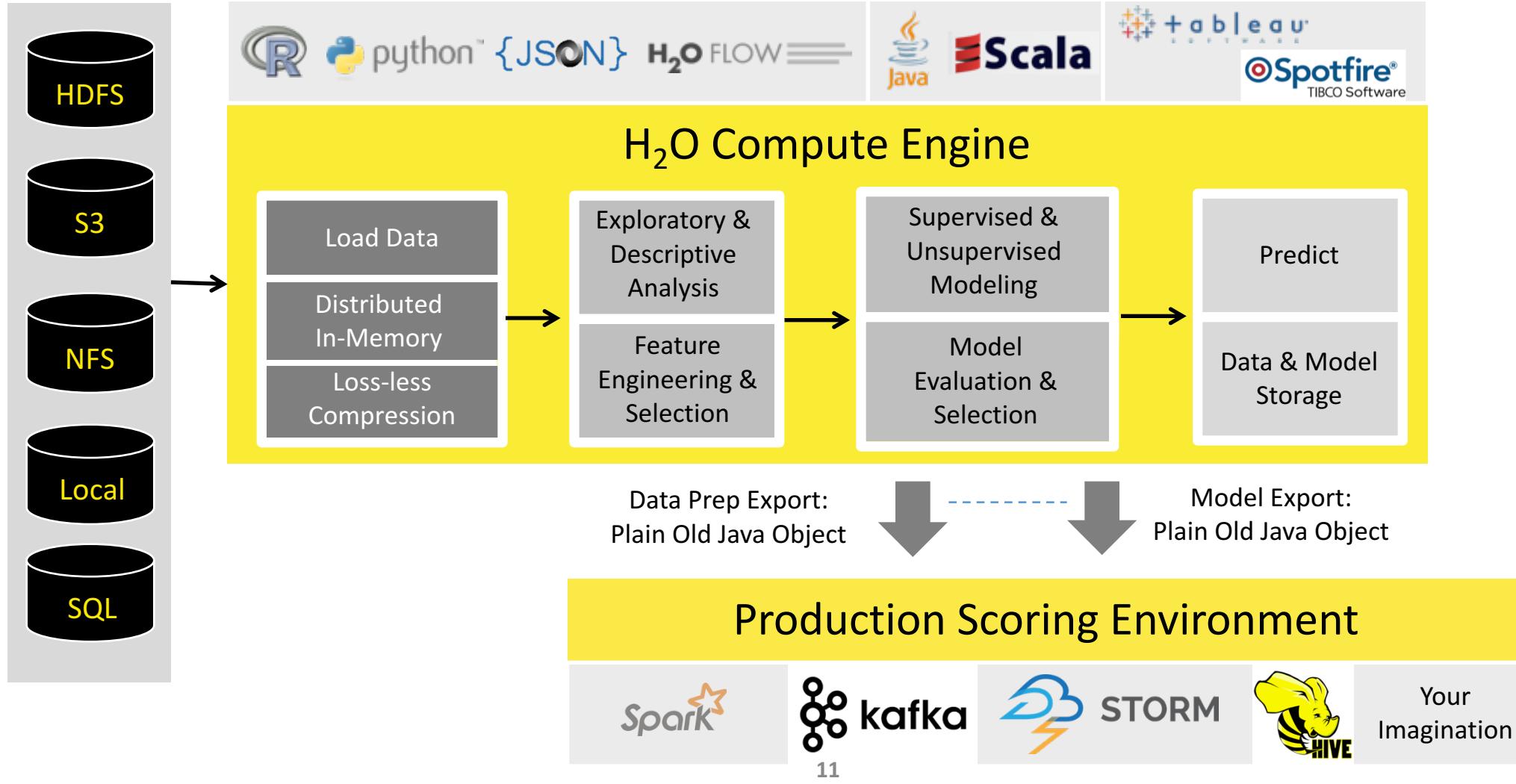


15 Meetups a month

7th Sep - London
12th Sep - Dublin

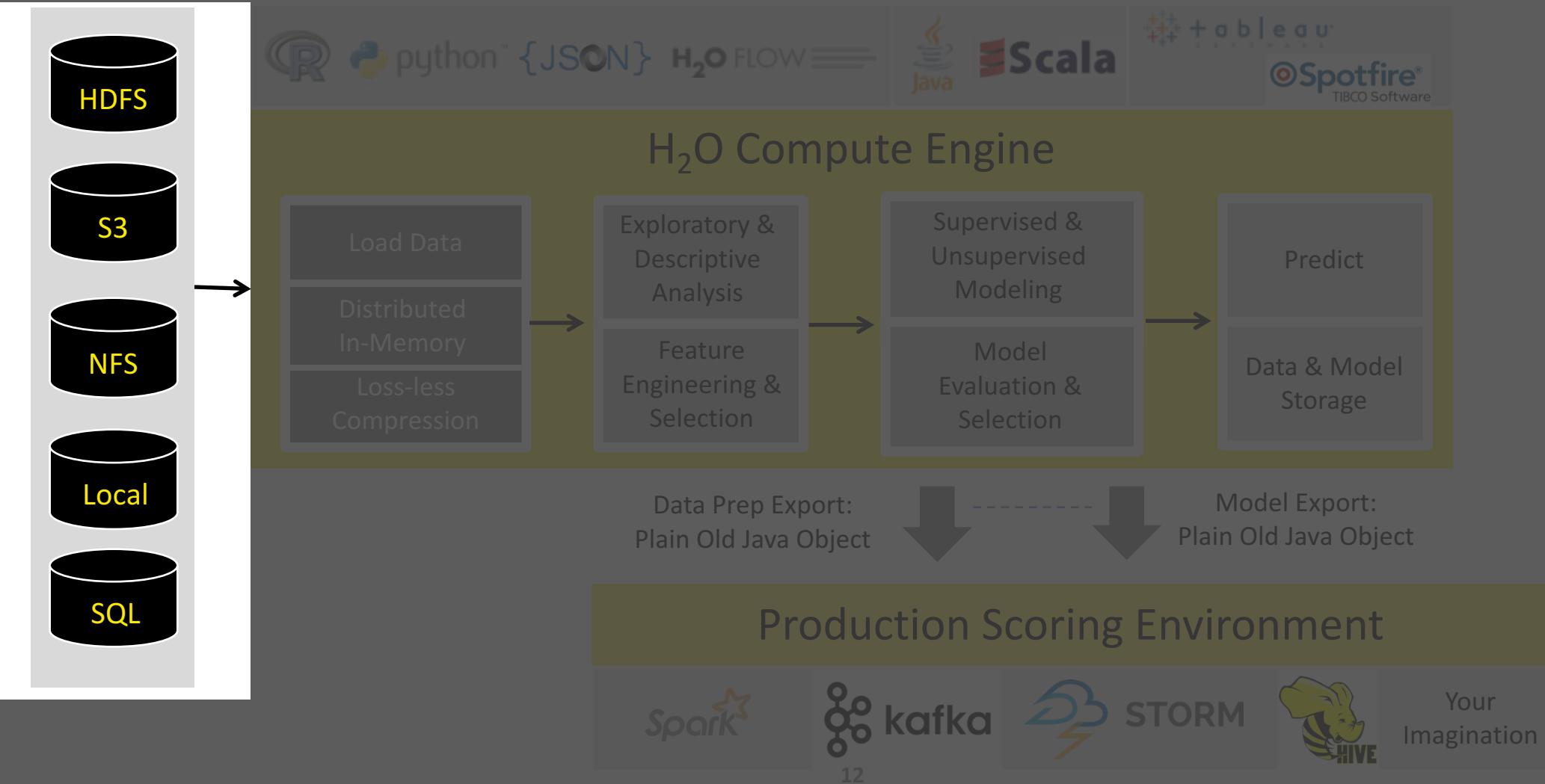
H₂O Machine Learning Platform

High Level Architecture



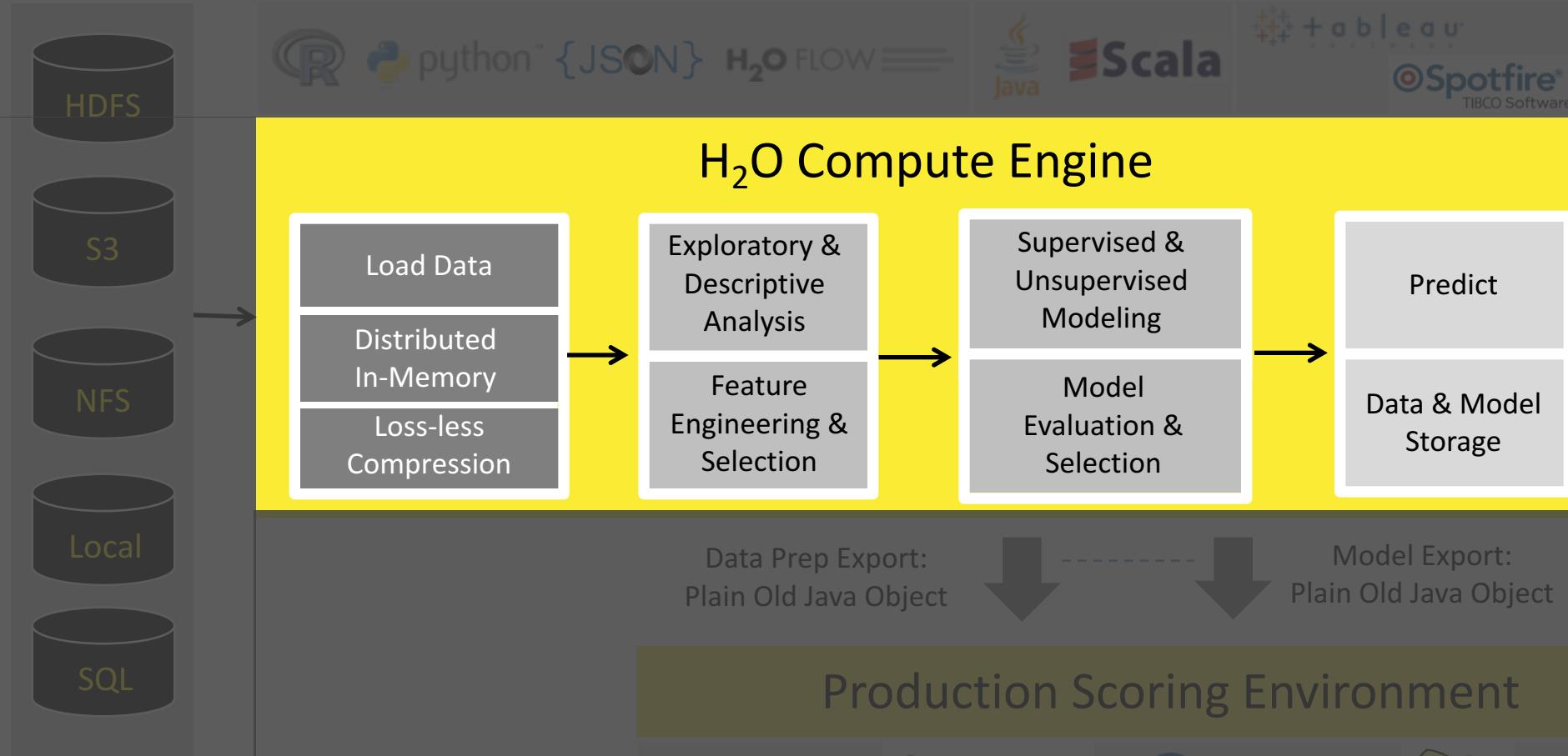
High Level Architecture

Import Data from
Multiple Sources



High Level Architecture

Fast, Scalable & Distributed
Compute Engine Written in
Java



Algorithms Overview

Supervised Learning

Statistical Analysis

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

Ensembles

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

Deep Neural Networks

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

Unsupervised Learning

Clustering

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

Dimensionality Reduction

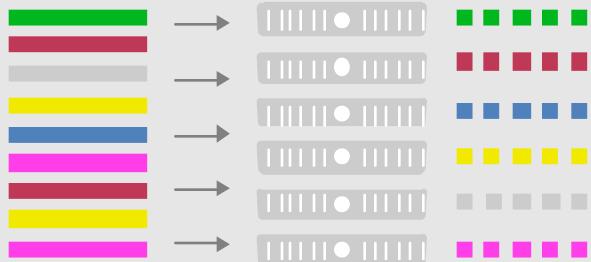
- **Principal Component Analysis:** Linearly transforms correlated variables to independent components
- **Generalized Low Rank Models:** extend the idea of PCA to handle arbitrary data consisting of numerical, Boolean, categorical, and missing data

Anomaly Detection

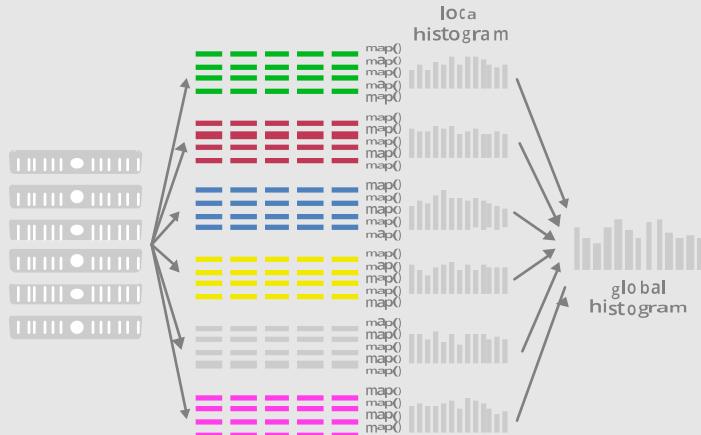
- **Autoencoders:** Find outliers using a nonlinear dimensionality reduction using deep learning

Distributed Algorithms

Foundation for Distributed Algorithms



Parallel Parse into **Distributed Rows**



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

Advantageous Foundation

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

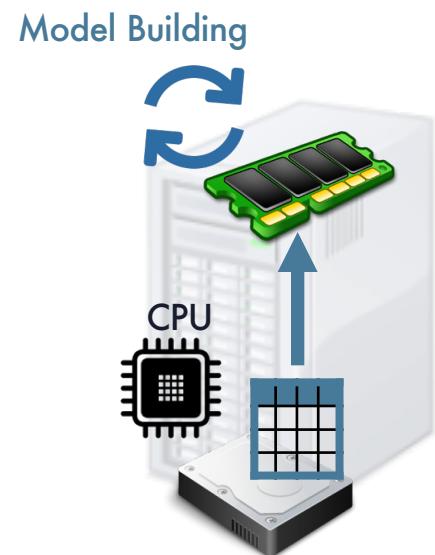
User Benefits

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

H₂O Core



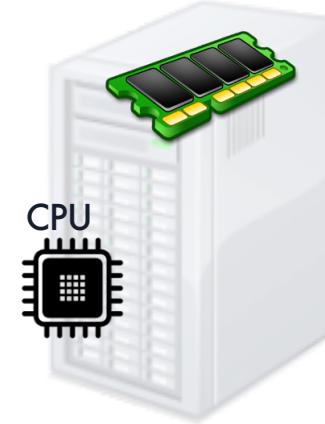
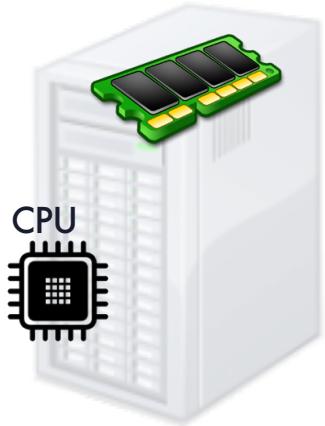
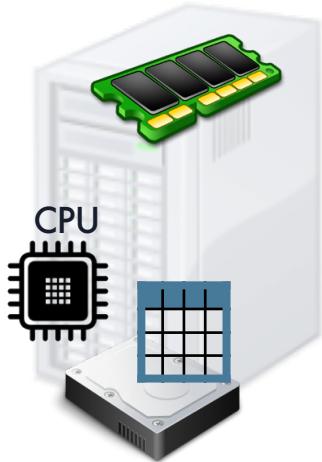
H₂O Core



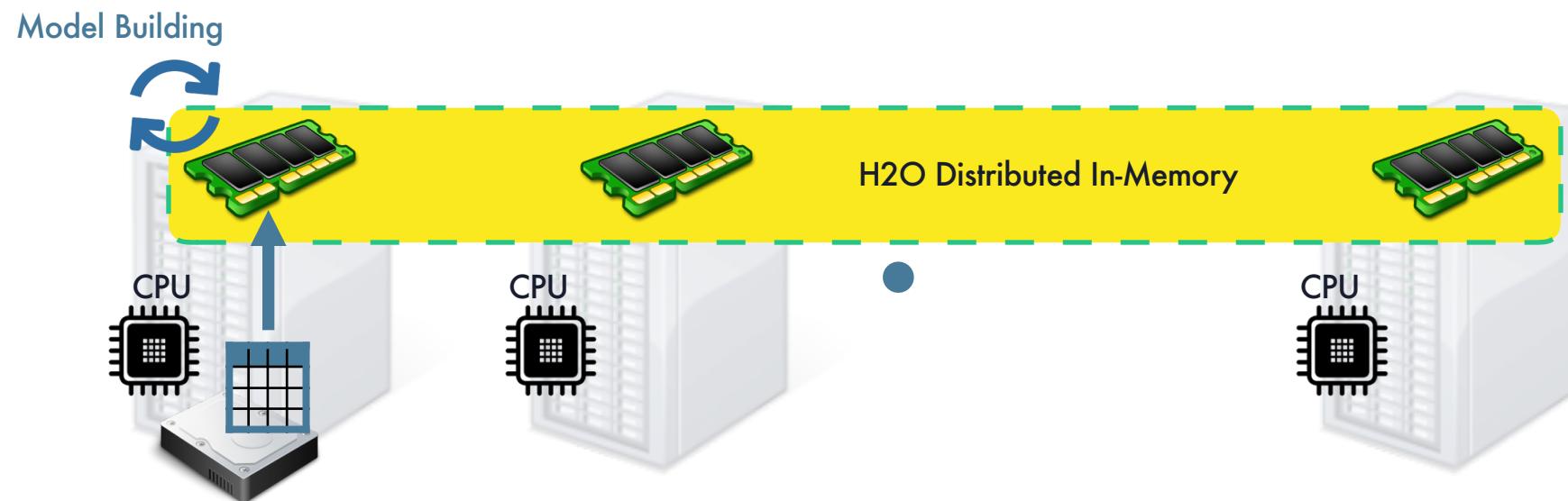
H₂O Core



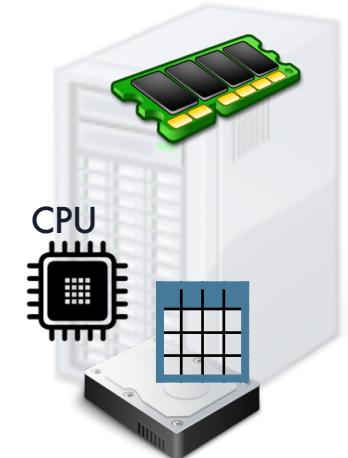
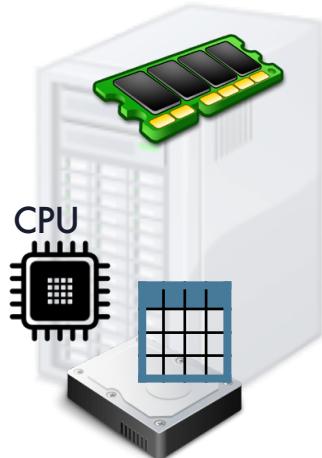
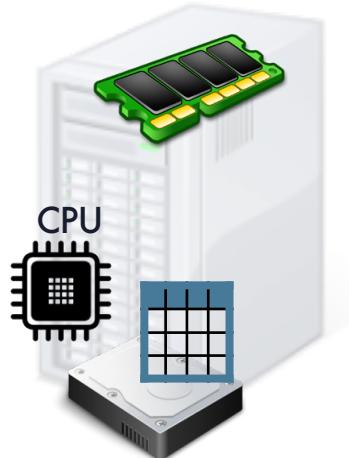
H₂O Core



H₂O Core



H₂O Core



YARN

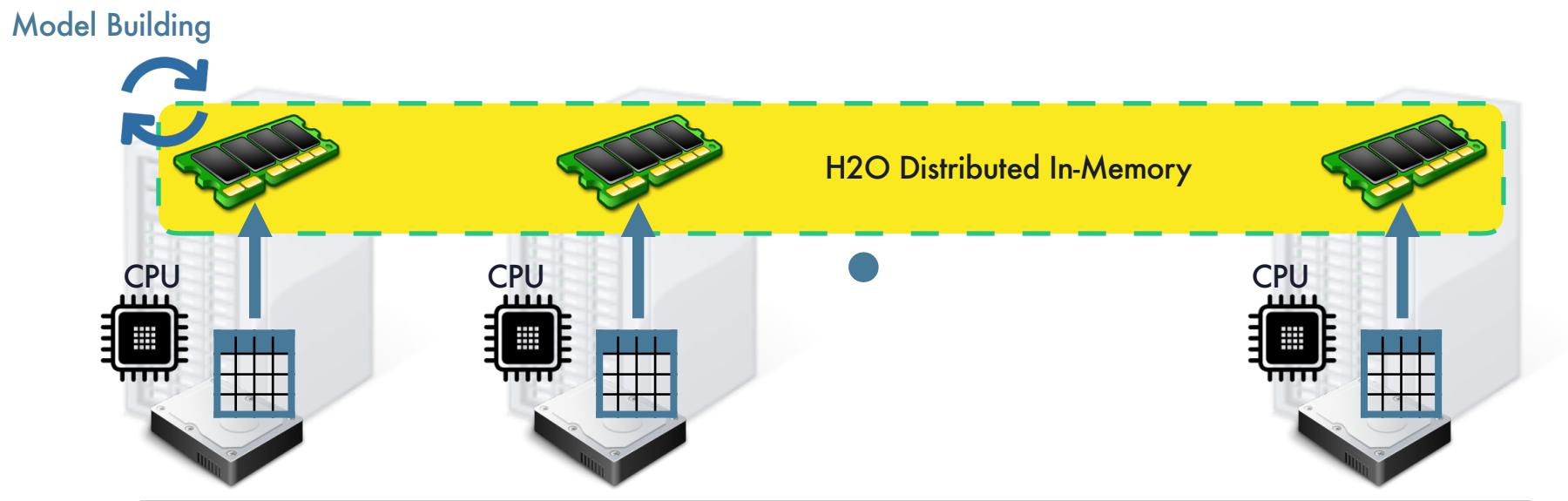
cloudera



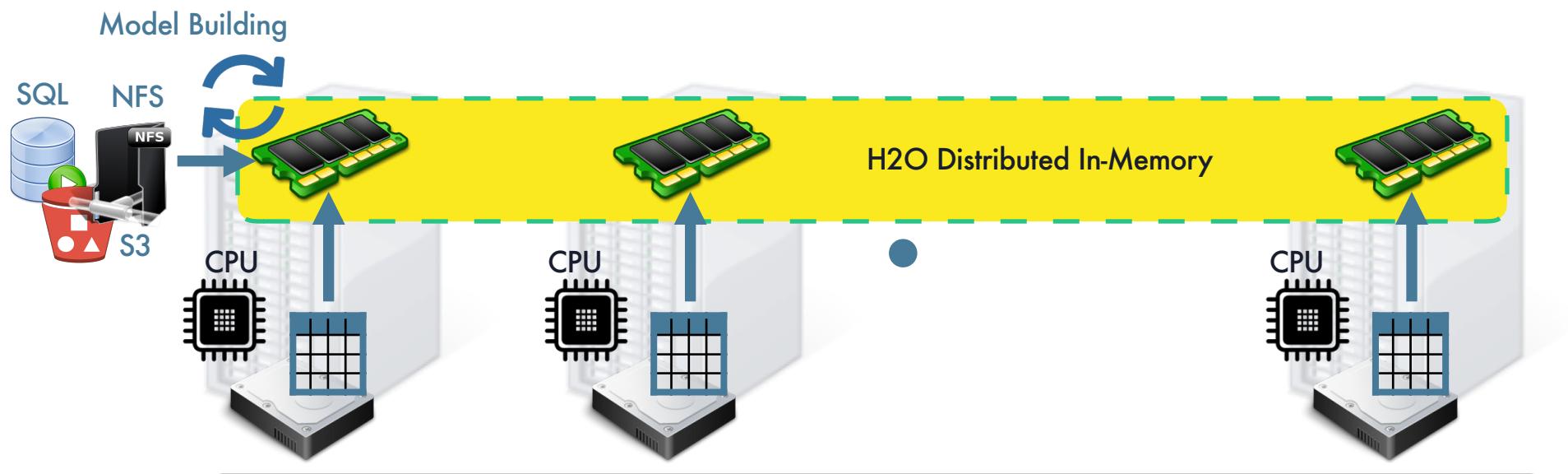
MAPR



H₂O Core



H₂O Core



YARN

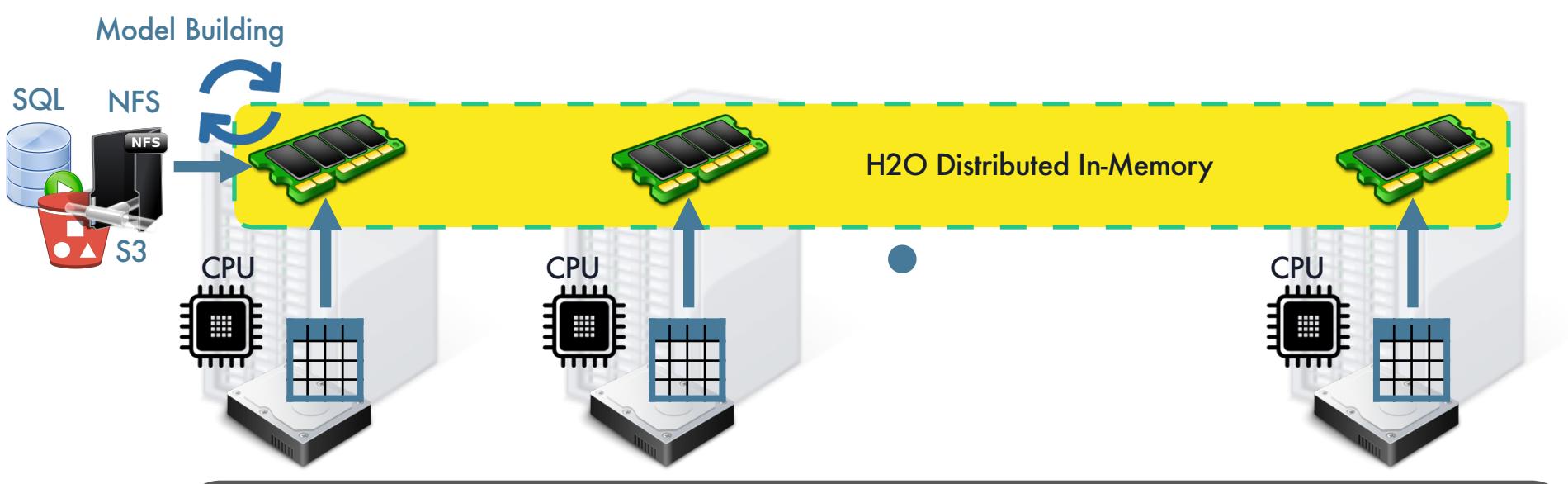
cloudera



MAPR



H₂O Core



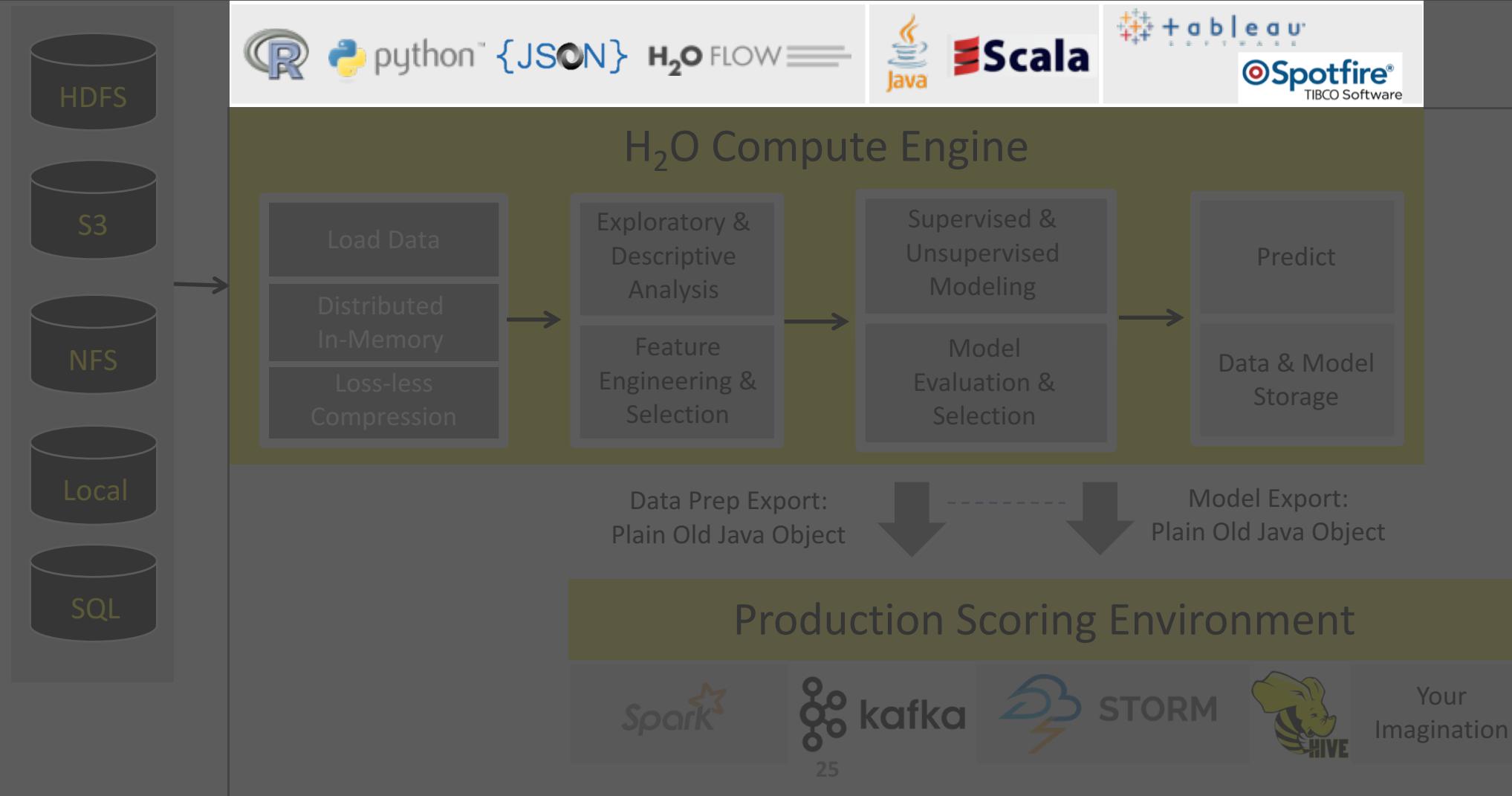
YARN

cloudera

Hortonworks

MAPR

High Level Architecture



H₂O Flow (Web)

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 top navigation bar includes "Flow", "Cell", "Data", "Model" (which is highlighted in yellow), "Score", "Admin", and "Help". A toolbar below the navigation bar contains various icons for file operations like import, export, and search.

The main workspace is titled "Untitled Flow" and contains a single step labeled "assist". To the left of the workspace is a sidebar titled "Assistance" which lists various H2O routines with their descriptions:

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

A context menu is open over the "assist" step, showing options such as 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..., and XGBoost... .

The right side of the interface features a "HELP" panel with sections for "Using Flow for the first time?", "Quickstart Videos", "Or, view example Flows to explore and learn H2O.", "STAR H2O ON GITHUB!", "GENERAL" (with links to Flow Web UI, Importing Data, Building Models, Making Predictions, Using Flows, and Troubleshooting Flow), and "EXAMPLES" (describing Flow packs and providing a link to Browse installed packs...). The bottom right corner shows "Connections: 0" and the H2O logo.

H₂O + R

```
# -----  
# Train a H2O Model  
# -----  
  
# Train three basic H2O models  
model_drf <- h2o.randomForest(x = features,  
.....y = target,  
.....model_id = "iris_random_forest",  
.....training_frame = d_iris)  
  
model_gbm <- h2o.gbm(x = features,  
.....y = target,  
.....model_id = "iris_gbm",  
.....training_frame = d_iris)  
  
model_dnn <- h2o.deeplearning(x = features,  
.....y = target,  
.....model_id = "iris_deep_learning",  
.....training_frame = d_iris)
```

H₂O + Python

Gradient Boosting Machines

```
# Build a Gradient Boosting Machines (GBM) model with default settings

# Import the function for GBM
from h2o.estimators.gbm import H2OGradientBoostingEstimator

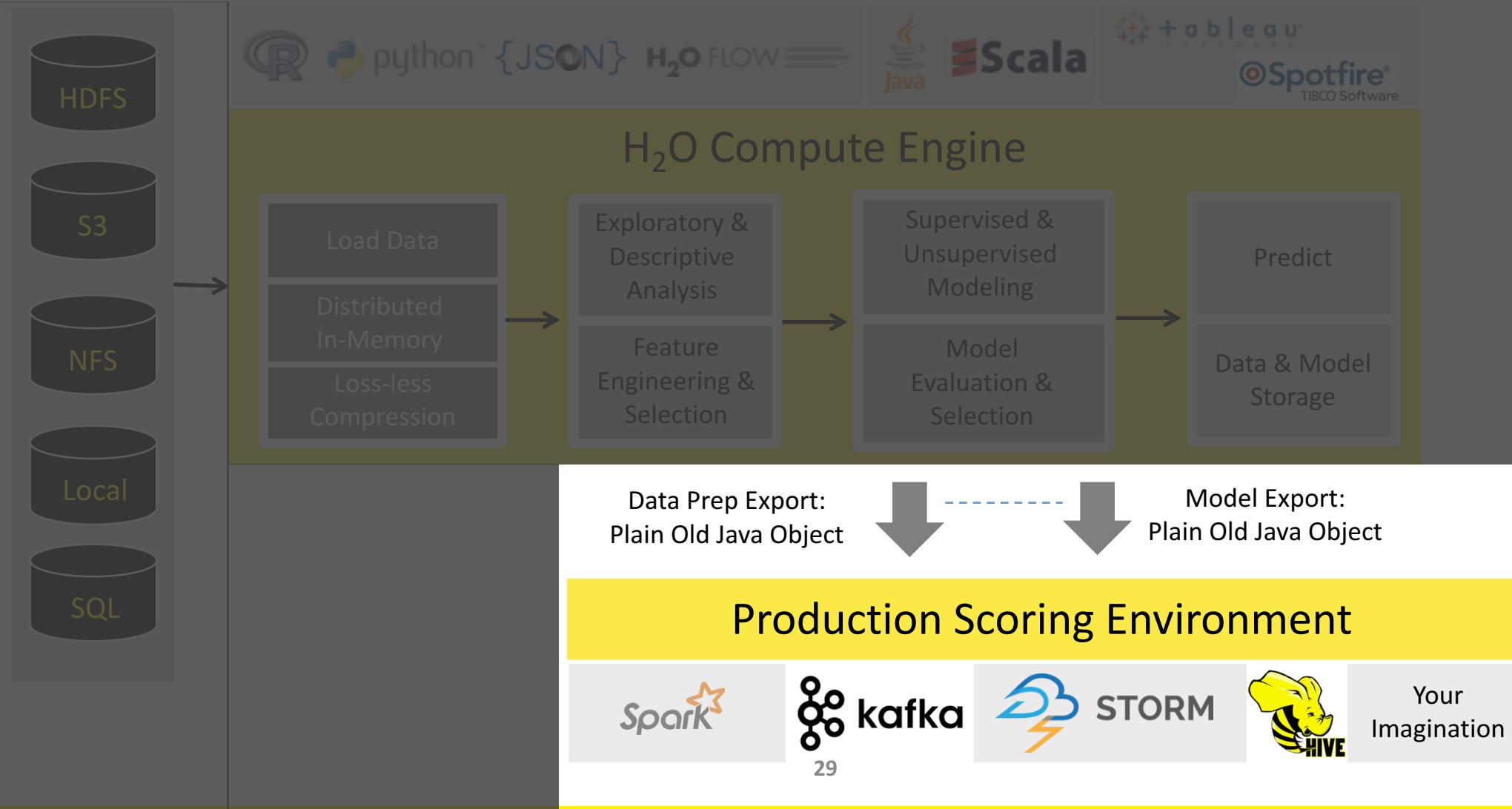
# Set up GBM for regression
# Add a seed for reproducibility
gbm_default = H2OGradientBoostingEstimator(model_id = 'gbm_default', seed = 1234)

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

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

High Level Architecture

Export Standalone Models
for Production



Getting Started & User Guides

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 Booklet](#)
[PySparkling Readme](#) 2.0 | 2.1 | 2.2
[RSparkling Readme](#)
[Open Source License \(Apache V2\)](#)

[Quick Start Video - Scala](#)

[Download Sparkling Water](#)

Steam

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

[Download Steam](#)

Deep Water (preview)

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

[Launch Deep Water AMI
\(choose p2.xlarge\)](#)

Q & A

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

For Supported Enterprise Customers
[Enterprise Support Web | Email](#)

Algorithms

Supervised Learning

Generalized Linear Modeling (GLM)	Tutorial	Booklet	Reference	Tuning
Gradient Boosting Machine (GBM)	Tutorial	Booklet	Reference	Tuning
Deep Learning	Tutorial	Booklet	Reference	Tuning
Distributed Random Forest	Tutorial	Booklet	Reference	Tuning
Naive Bayes	Tutorial	Booklet	Reference	Tuning
Stacked Ensembles	Tutorial	Booklet	Reference	Tuning
XGBoost	Tutorial	Booklet	Reference	Tuning

Unsupervised Learning

Generalized Low Rank Models (GLRM)	Tutorial	Reference
K-Means Clustering	Tutorial	Reference
Principal Components Analysis (PCA)	Tutorial	Reference

Miscellaneous

Word2vec	Tutorial	Reference
----------	--------------------------	---------------------------

TOP

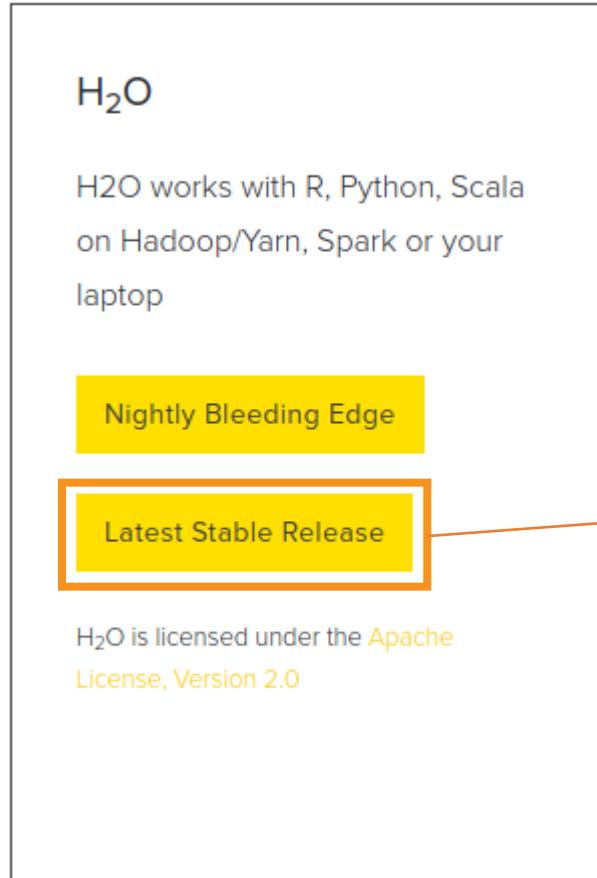
Use Case 1

- Install and start H₂O (Web/R/Python)
- Machine Learning Basics
- Import Credit Card dataset and train a model to predict default payments

H₂O Quick Start

www.h2o.ai/download

Prerequisite: Java version 1.7+



H₂O
Version 3.14.0.2

Fast Scalable Machine Learning API
For Smarter Applications

DOWNLOAD AND RUN INSTALL IN R INSTALL IN PYTHON INSTALL ON HADOOP USE FROM MAVEN

DOWNLOAD H₂O

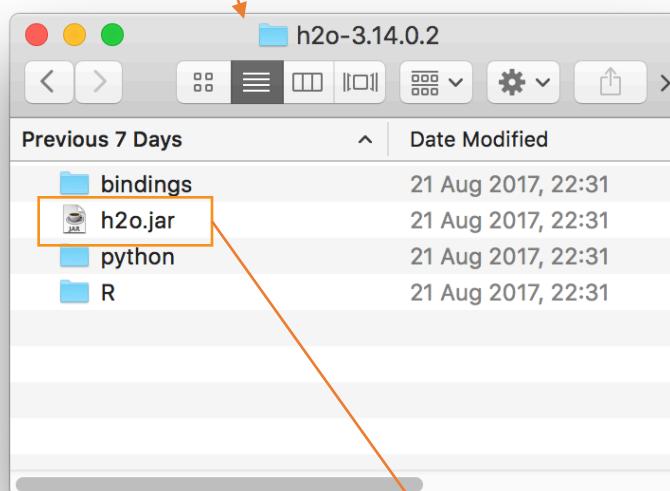
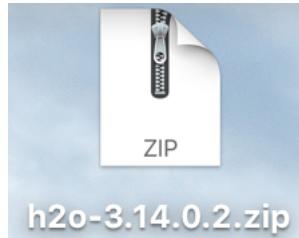
Get started with H₂O in 3 easy steps

1. Download H₂O. This is a zip file that contains everything you need to get started.
2. From your terminal, run:

```
cd ~/Downloads  
unzip h2o-3.14.0.2.zip  
cd h2o-3.14.0.2  
java -jar h2o.jar
```

3. Point your browser to <http://localhost:54321>

Install and Start H₂O Flow (Web Interface)



```
Jo-fais-MBP-2:h2o-3.14.0.2 jofaichow$ java -jar h2o.jar
```

```
Jo-fais-MBP-2:h2o-3.14.0.2 java -jar h2o.jar -- 175x73
[...]
INFO: Open H2O Flow in your web browser: http://192.168.1.70:54321
```

INFO: Open H2O Flow in your web browser: <http://192.168.1.70:54321>

H₂O Flow (Web Interface)

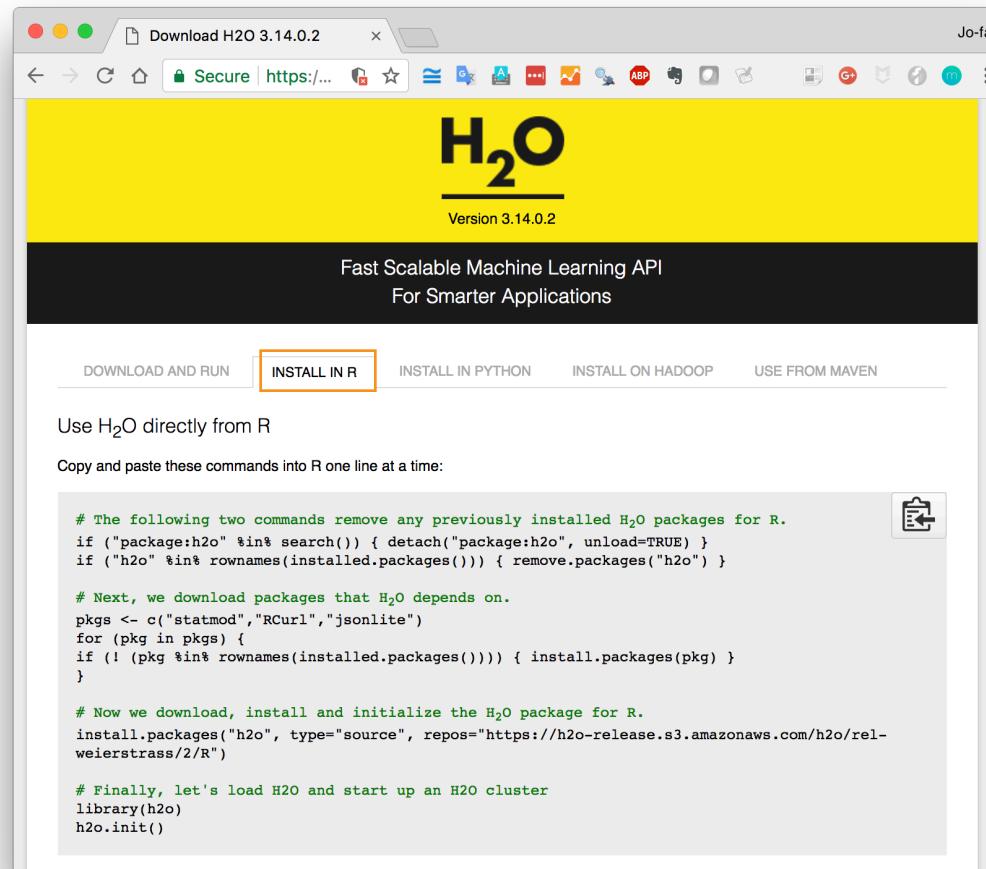
The screenshot shows the H2O Flow web interface running in a browser window. The title bar says "H2O Flow". The address bar shows "localhost:54321/flow/index.html". The top navigation menu includes "Flow", "Cell", "Data", "Model", "Score", "Admin", and "Help". The main content area is titled "Untitled Flow" and contains a toolbar with various icons. On the left, there's a sidebar titled "Assistance" with a list of routines:

Routine	Description
importFiles	Import file(s) into H ₂ O
getFrames	Get a list of frames in H ₂ O
splitFrame	Split a frame into two or more frames
mergeFrames	Merge two frames into one
getModels	Get a list of models in H ₂ O
getGrids	Get a list of grid search results in H ₂ O
getPredictions	Get a list of predictions in H ₂ O
getJobs	Get a list of jobs running in H ₂ O
buildModel	Build a model
runAutoML	Automatically train and tune many models
importModel	Import a saved model
predict	Make a prediction

The right side of the interface has tabs for "OUTLINE", "FLOWS", "CLIPS", and "HELP". The "HELP" tab is active. It contains sections for "Using Flow for the first time?", "Quickstart Videos", and "Or, view example Flows to explore and learn H₂O." An orange callout box points to the "view example Flows" link. Below these are sections for "STAR H2O ON GITHUB!" (with a star icon and 2,379 stars), "GENERAL" (with links to Flow Web UI, Importing Data, Building Models, Making Predictions, Using Flows, and Troubleshooting Flow), and "EXAMPLES" (with a note about Flow packs and a link to Browse installed packs...). The bottom status bar says "Ready" and "Connections: 0".

More Examples

Install and Start H₂O in R / Python



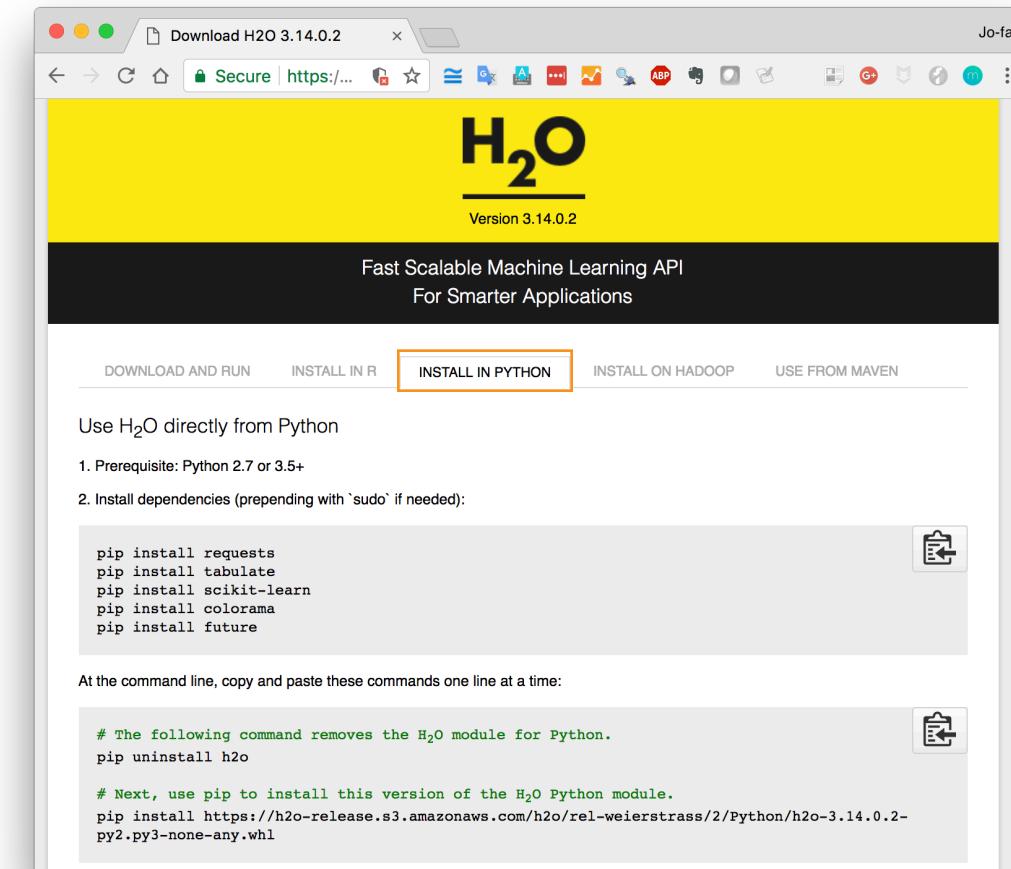
The screenshot shows the H₂O download page for version 3.14.0.2. The top navigation bar includes links for DOWNLOAD AND RUN, INSTALL IN R (which is highlighted), INSTALL IN PYTHON, INSTALL ON HADOOP, and USE FROM MAVEN. Below the navigation, there's a section titled "Use H₂O directly from R" with R code for installation and initialization. A clipboard icon is located next to the code block.

```
# The following two commands remove any previously installed H2O packages for R.
if ("package:h2o" %in% search()) { detach("package:h2o", unload=TRUE) }
if ("h2o" %in% rownames(installed.packages())) { remove.packages("h2o") }

# Next, we download packages that H2O depends on.
pkgs <- c("statmod", "RCurl", "jsonlite")
for (pkg in pkgs) {
  if (! (pkg %in% rownames(installed.packages()))) { install.packages(pkg) }
}

# Now we download, install and initialize the H2O package for R.
install.packages("h2o", type="source", repos="https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/R")

# Finally, let's load H2O and start up an H2O cluster
library(h2o)
h2o.init()
```



The screenshot shows the H₂O download page for version 3.14.0.2. The top navigation bar includes links for DOWNLOAD AND RUN, INSTALL IN R, INSTALL IN PYTHON (which is highlighted), INSTALL ON HADOOP, and USE FROM MAVEN. Below the navigation, there's a section titled "Use H₂O directly from Python" with instructions and command-line code. A clipboard icon is located next to the code block.

1. Prerequisite: Python 2.7 or 3.5+
2. Install dependencies (prepending with `sudo` if needed):

```
pip install requests
pip install tabulate
pip install scikit-learn
pip install colorama
pip install future
```

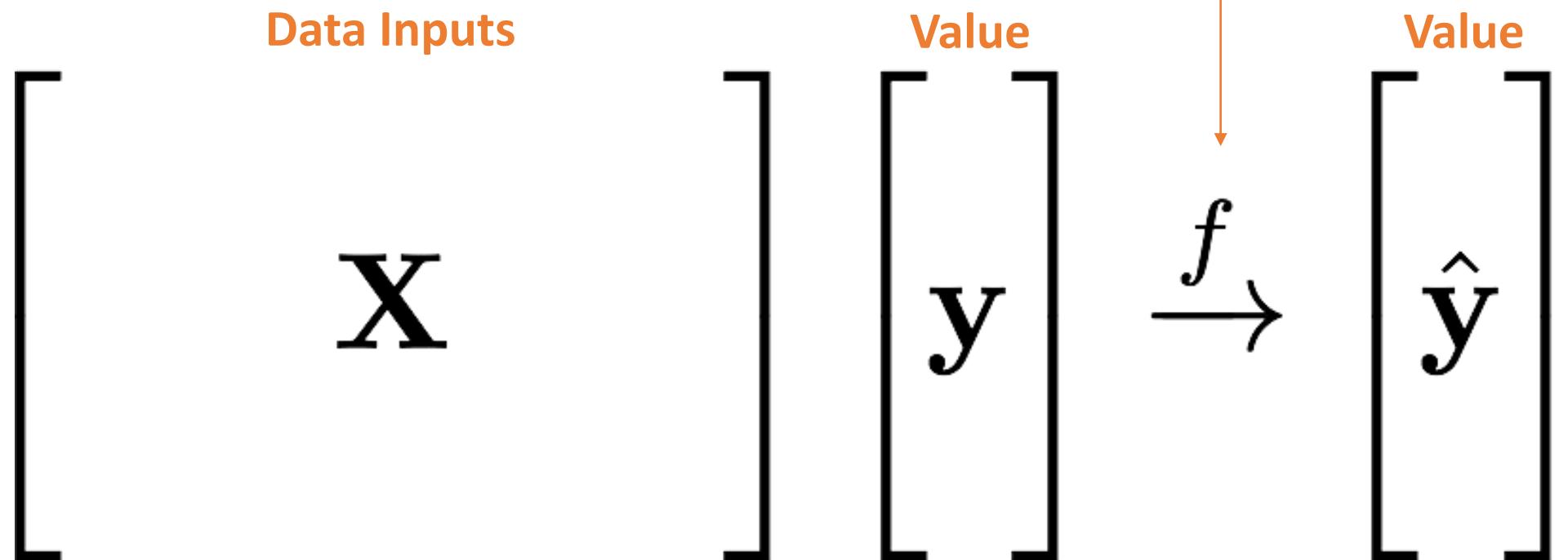
At the command line, copy and paste these commands one line at a time:

```
# The following command removes the H2O module for Python.
pip uninstall h2o

# Next, use pip to install this version of the H2O Python module.
pip install https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/Python/h2o-3.14.0.2-py2.py3-none-any.whl
```

Machine Learning Basics

Learning from Data



Learn the Pattern

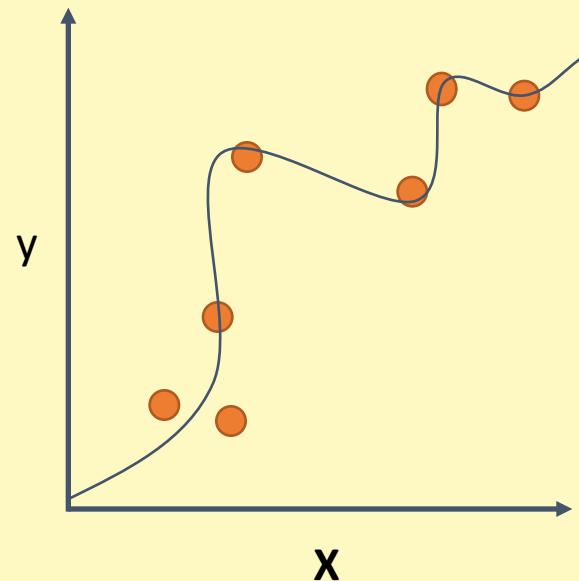
Historical
Value

Predicted
Value

$$f \rightarrow$$

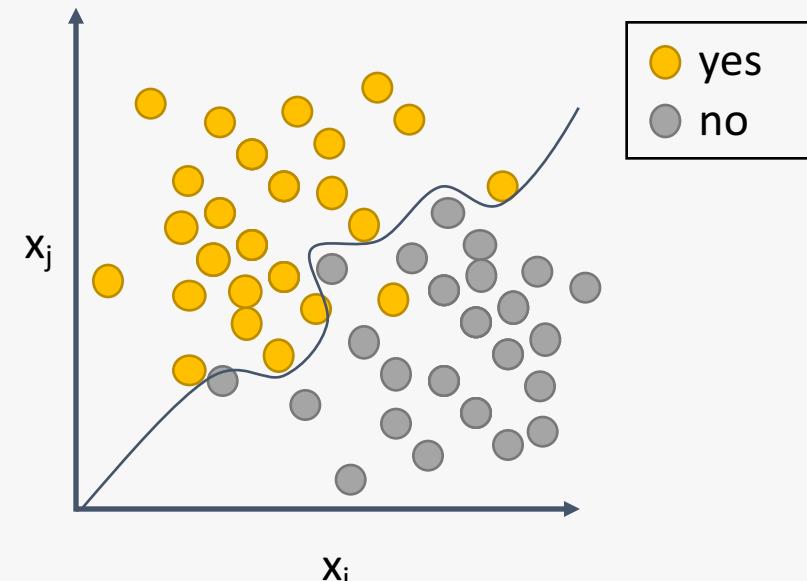
Supervised Learning

Regression:
How much will a customers spend?



H₂O algos:
Penalized Linear Models
Random Forest
Gradient Boosting
Neural Networks
Stacked Ensembles

Classification:
Will a customer make a purchase? Yes or No



H₂O algos:
Penalized Linear Models
Naïve Bayes
Random Forest
Gradient Boosting
Neural Networks
Stacked Ensembles

Credit Card Dataset

- Links:
 - https://github.com/woobe/h2o_tutorials/raw/master/datasets/credit_card_train.csv
 - https://github.com/woobe/h2o_tutorials/raw/master/datasets/credit_card_test.csv

credit_card_train.csv

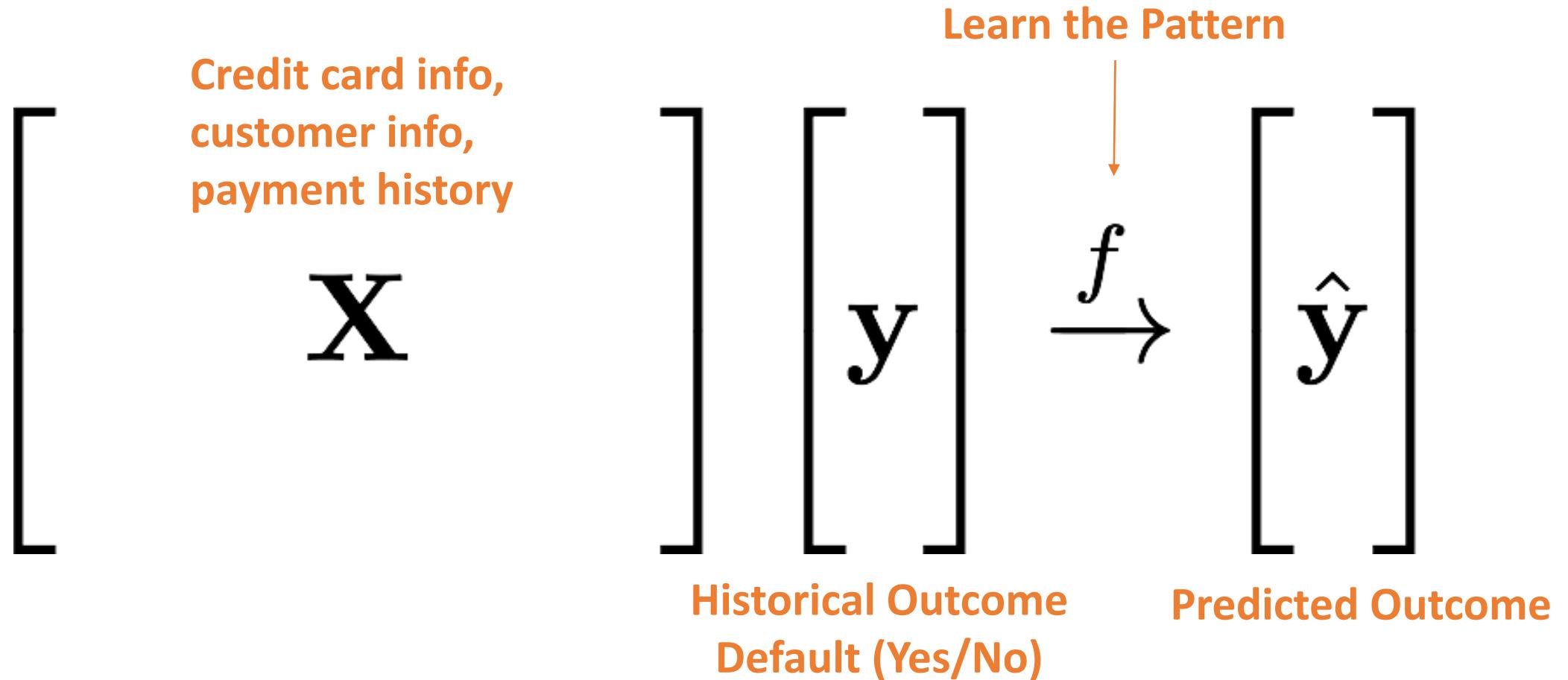
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	DEFAULT_PAYMENT_NEXT_MONTH	
2	20000	Female	2	1	24	2	2	-1	-2	-2	3913	3102	689	0	0	0	0	689	0	0	0	0	0	Yes	
3	120000	Female	2	2	26	-1	2	0	0	0	2682	1725	2682	3272	3455	3261	0	1000	1000	1000	0	0	2000	Yes	
4	90000	Female	2	2	34	0	0	0	0	0	29239	14027	13559	14331	14948	15549	1518	1500	1000	1000	1000	5000	No		
5	50000	Female	2	1	37	0	0	0	0	0	46990	48233	49291	28314	28959	29547	2000	2019	1200	1100	1069	1000	No		
6	50000	Male	2	1	57	-1	0	-1	0	0	8617	5670	35835	20940	19146	19131	2000	36681	10000	9000	689	679	No		
7	100000	Female	2	2	23	0	-1	-1	0	0	11876	380	601	221	-159	567	380	601	0	581	1687	1542	No		
8	140000	Female	3	1	28	0	0	2	0	0	11285	14096	12108	12211	11793	3719	3329	0	432	1000	1000	1000	No		
9	20000	Male	3	2	35	-2	-2	-2	-2	-1	0	0	0	0	0	0	0	0	0	13007	1122	0	No		
10	200000	Female	3	2	34	0	0	2	0	0	11073	9787	5535	2513	1828	3731	2306	12	50	300	3738	66	No		
11	630000	Female	2	2	41	-1	0	-1	-1	-1	12137	6500	6500	6500	6500	2870	1000	6500	6500	6500	2870	0	No		
12	70000	Male	2	2	30	1	2	2	0	0	2	65802	67369	65701	66782	36137	36894	3200	0	3000	3000	1500	0	Yes	
13	250000	Male	1	2	29	0	0	0	0	0	70007	67000	67000	67000	67000	55512	3000	3000	3000	3000	3000	3000	No		
14	50000	Female	3	2	41	-1	0	-1	-1	-1	12137	6500	6500	6500	6500	2870	1000	6500	6500	6500	2870	0	No		
15	20000	Male	1	2	34	0	0	2	0	0	11073	9787	5535	2513	1828	3731	2306	12	50	300	3738	66	No		
16	320000	Male	1	2	23	0	0	0	0	0	4744	7070	0	5398	6360	8292	3	1200	2045	2000	2000	2000	No		
17	360000	Female	1	2	27	1	-2	-1	-1	-1	-109	-425	259	-57	127	-189	19104	3200	0	1500	0	1650	0	Yes	
18	180000	Female	1	2	30	0	0	0	0	0	22541	16138	17163	17878	18931	19617	0	0	0	0	0	0	No		
19	130000	Female	3	1	47	-1	-1	-1	-1	-1	650	3415	3416	2040	30430	257	930	3000	1537	1000	2000	930	33764	No	
20	120000	Female	2	2	26	0	0	0	0	0	15329	16575	17496	17907	18375	11400	3	1432	1062	997	0	0	0	Yes	
21	70000	Female	2	2	27	-1	-1	-1	-1	-1	16646	17265	13266	15339	14307	36923	1	500	0	1000	0	1000	0	Yes	
22	450000	Female	1	2	23	0	0	0	0	0	47620	41810	36023	28967	29829	30046	195599	10358	10000	75940	20000	195599	50000	No	
23	90000	Male	1	2	23	0	0	0	-1	0	0	4744	7070	0	5398	6360	8292	0	0	0	0	0	0	No	
24	50000	Male	3	2	23	0	0	0	0	0	47620	41810	36023	28967	29829	30046	195599	10358	10000	75940	20000	195599	50000	No	
25	60000	Male	1	2	27	1	-2	-1	-1	-1	-109	-425	259	-57	127	-189	19104	3200	0	1500	0	1650	0	Yes	
26	50000	Female	3	2	30	0	0	0	0	0	22541	16138	17163	17878	18931	19617	0	0	0	0	0	0	No		
27	50000	Female	3	1	47	-1	-1	-1	-1	-1	650	3415	3416	2040	30430	257	930	3000	1537	1000	2000	930	33764	No	
28	50000	Male	1	2	26	0	0	0	0	0	15329	16575	17496	17907	18375	11400	46012	2007	3582	0	3601	0	1820	Yes	
29	230000	Female	1	2	27	-1	-1	-1	-1	-1	16646	17265	13266	15339	14307	36923	1	1432	1062	997	0	0	0	No	
30	100000	Male	1	2	32	0	0	0	0	0	93036	84071	82880	80958	78703	75589	3023	3511	3302	320	0	0	0	Yes	
31	50000	Female	2	2	54	-2	-2	-2	-2	-2	10929	4152	22722	7521	71439	8981	4152	22827	7521	7143	0	0	0	No	
32	500000	Male	1	1	58	-2	-2	-2	-2	-2	13709	5006	31130	3180	0	5293	5006	31178	3180	0	0	0	No		
33	160000	Male	1	2	30	-1	-1	-2	-2	-1	30265	-131	-527	-923	-1488	-1884	131	396	396	396	56	0	No		
34	280000	Male	2	1	40	0	0	0	0	0	186503	181328	180422	170410	173901	177413	8026	8060	6300	6300	640	0	No		
35	60000	Female	2	2	22	0	0	0	0	-1	15054	9806	11068	6026	-28335	18660	1500	1518	2043	0	0	0	No		
36	50000	Male	1	2	25	1	-1	-1	-2	-2	0	780	0	0	0	0	780	0	0	0	0	0	Yes		
37	280000	Male	1	2	31	-1	-1	2	-1	0	498	9075	4641	9976	17976	9477	9075	0	9976	800	0	0	No		
38	360000	Male	1	2	33	0	0	0	0	0	218668	221296	206895	628699	195969	179224	10000	7000	6000	6000	18884	0	No		
39	70000	Female	1	2	25	0	0	0	0	0	67521	66999	63949	63699	64718	65970	3000	4500	4042	250	0	0	No		
40	10000	Male	2	2	22	0	0	0	0	0	1977	2184	6002	2575	2620	2454	1500	2022	1000	0	0	0	No		

Data Inputs (x):
credit card limit balance,
customer info (sex, education, marriage, age),
payment history (six months)

Learn the Pattern

Historical Outcome (y):
Default Credit Card Payment after six months?
(Yes / No)

Learning from Credit Card Data



default_payment_test_data

credit_card_test.csv

Home Insert Page Layout Formulas Data Review View

Cut Copy Format

Calibri (Body) 12 A A Wrap Text General Conditional Formatting

Merge & Center Format as Table

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6				
2	50000	Male	1	2	37	0	0	0	0	0	0	64400	57069	57608	19394	19619	20024	2500	1815	657	1000	1000	800				
3	500000	Male	1	2	29	0	0	0	0	0	0	367965	412023	445007	542653	483003	473944	55000	40000	38000	20239	13750	13770				
4	260000	Female	1	2	51	-1	-1	-1	-1	-1	2	12261	21670	9966	8517	22287	13668	21818	9966	8583	22301	0	3640				
5	50000	Male	2	2	33	2	0	0	0	0	0	30518	29618	22102	22734	23217	23680	1718	1500	1000	1000	1000	716				
6	150000	Female	5	2	46	0	0	-1	0	0	-2	4463	3034	1170	1170	0	1013	1170	0	0	0	0	0				
7	20000	Male	1	2	24	0	0	0	0	0	0	17447	18479	19476	19865	20480	20063	1318	1315	704	928	912	1069				
8	130000	Female	2	1	51	-1	-1	-2	-2	-1	-1	99	0	0	0	0	0	0	0	0	2353	0	0				
9	320000	Male	2	2	29	2	2	2	2	2	2	58267	59246	60184	58622	62307	63526	2500	2500	0	4800	2400	1600				
10	50000	Male	3	2	25	-1	0	0	0	0	0	42838	37225	36087	9636	9590	10030	1759	1779	320	500	1000	1000				
11	130000	Female	1	1	35	0	0	0	-1	-1	-1	81313	117866	17740	1330	7095	1190	40000	5000	1330	7095	1190	2090				
12	20000	Male	3	2														0	1651	1000	2000	0	1500				
13	100000	Female	1	2														7555	0	0	0	0	0				
14	400000	Male	2	1														9677	11867	7839	14837	7959	5712				
15	180000	Male	1	1														4655	2690	2067	2142	2217	1000				
16	260000	Female	1	1														0	22500	0	969	1000	0				
17	140000	Male	2	1														3455					2602				
18	210000	Male	3	1														10478					10478				
19	370000	Male	1	2														15383					4699				
20	50000	Female	1	2	24	1	-2	-2	-2	-2	-2	-709	-709	-709	-2898	-3272	-3272	0					0				
21	180000	Female	1	2	29	-1	-1	-1	-2	-1	0	11386	199	0	0	17227	17042	199					5114				
22	120000	Male	2	2	26	0	0	0	0	0	0	107314	110578	113736	116000	119131	122135	5000					5000				
23	470000	Male	2	2	27	2	2	2	2	0	0	296573	303320	307843	479978	305145	309959	13000	11001	0	10484	10838	10367				
24	50000	Male	2	2	23	2	0	0	0	0	0	49758	48456	44116	21247	20066	18858	2401	2254	2004	704	707	1004				
25	20000	Male	2	2	23	1	2	0	0	2	0	20235	17132	16856	16875	13454	10104	0	1200	1000	0	1000	10000				
26	60000	Female	1	2	28	1	2	2	-2	-2	-1	21501	20650	0	0	0	2285	0	0	0	0	0	2285	0			
27	250000	Female	2	1	75	0	-1	-1	-1	-1	-1	52874	1631	1536	1010	5572	794	1631	1536	1010	5572	794	1184				
28	30000	Male	2	2	28	0	0	0	0	0	0	29242	29507	29155	25255	22001	0	5006	1244	851	955	0	0				
29	100000	Female	3	1	43	0	0	-2	-2	-2	-2	62170	0	0	0	0	0	0	0	0	0	0	0				
30	50000	Female	1	2	26	-1	-1	-1	-1	-1	-1	1156	316	316	316	316	316	316	316	316	316	316	316				
31	110000	Female	2	2	36	0	0	0	0	0	0	47819	48947	50330	50894	52175	53652	2200	2500	2000	2100	2500	2200				
32	180000	Male	2	2	29	1	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	0	0	0	0	0					
33	110000	Male	2	2	29	1	2	2	0	0	0	58362	56598	51908	48647	47862	47969	2500	0	2000	2000	1854	2000				
34	20000	Male	1	2	27	0	0	0	0	0	0	20571	19089	19658	19453	19108	18868	1323	1600	830	700	674	376				
35	140000	Female	2	2	29	0	0	0	0	0	0	20110	17102	18862	19996	21214	21085	3000	3000	3000	3500	2000	2000				
36	60000	Female	1	2	23	1	2	2	2	2	2	29332	28577	30805	31601	32349	32965	0	2709	1600	1400	1300	1200				
37	230000	Female	2	2	27	1	2	0	0	0	0	13668	12647	13135	10596	9218	5068	0	1064	423	313	1000	4641				
38	70000	Male	1	2	27	0	0	0	0	0	0	70119	68536	66601	29401	28949	29795	3600	1646	600	28468	1327	1000				
39	90000	Male	3	1	48	1	2	2	2	2	2	77604	73317	71334	67009	63228	59378	1700	4000	1600	1600	1500	4086				
40	30000	Female	2	1	43	2	2	2	2	2	2	28702	26622	24022	24268	20850	10622	1200	1608	0	0	0	800				

Data Inputs (x) only – no outcome data
(as if this is the data from new customers)

We use this dataset to make prediction

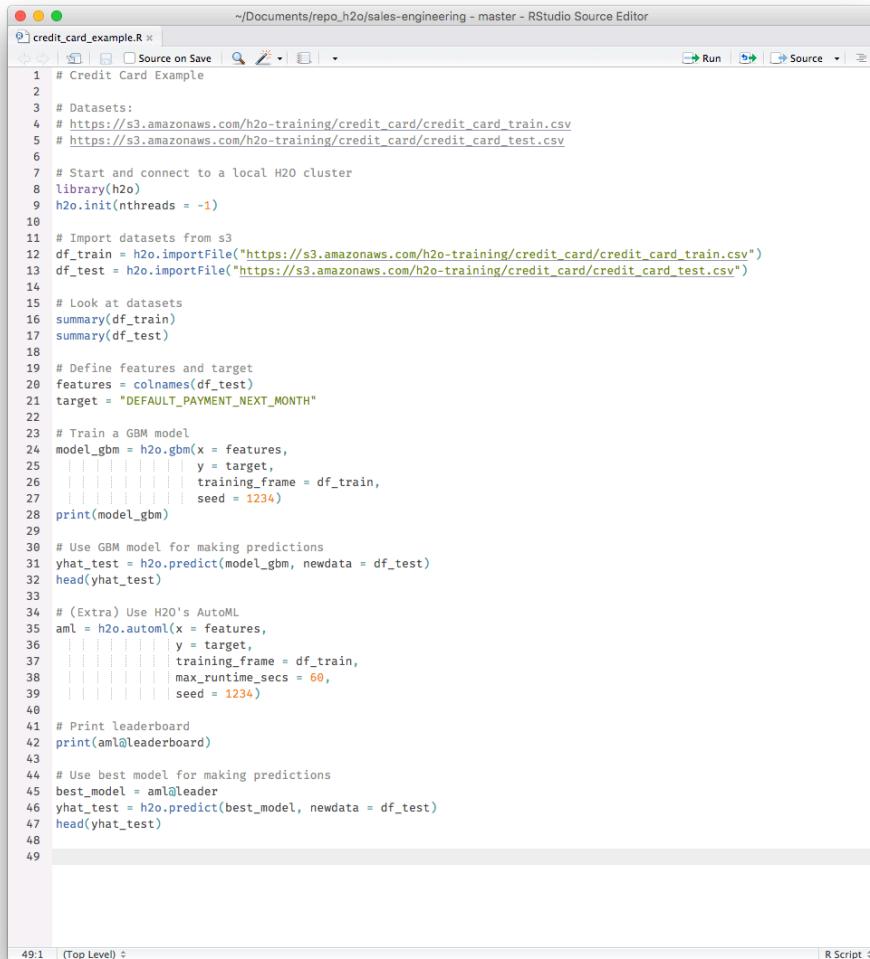
Make Predictions

Probability of Default Payments
(Decision Makers to Take Actions)

Live Demo 1

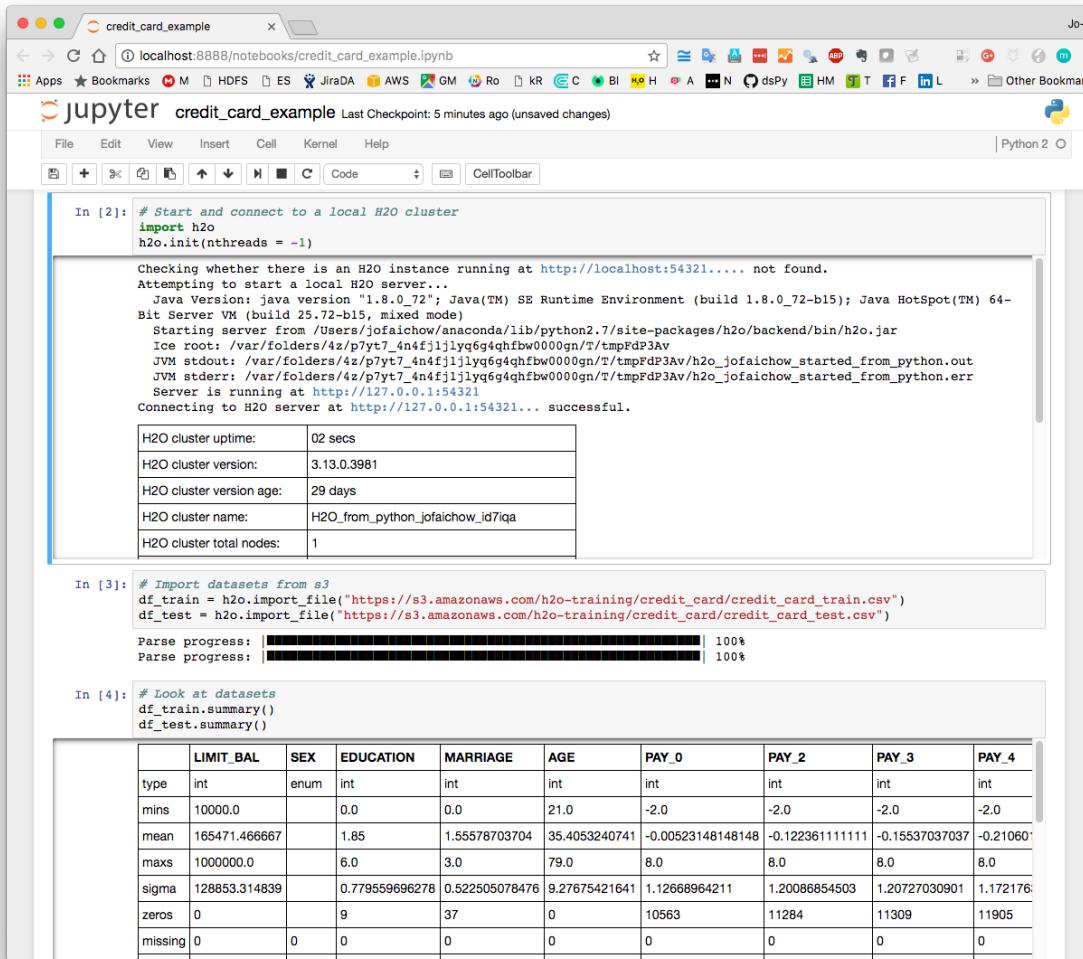
- Import and explore credit card dataset
- Train a gradient boosting model and make predictions
- Train many models using AutoML and make predictions using the best model

Using H₂O with R and Python



The screenshot shows the RStudio Source Editor window with the file `credit_card_example.R` open. The code is a script for a credit card example using the H2O library in R. It includes importing datasets from S3, connecting to a local H2O cluster, training a GBM model, and printing the leaderboard. The code is annotated with line numbers from 1 to 49.

```
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@leader
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 a credit card example. In cell [2], it attempts to start a local H2O server but finds one already running. It then prints information about the H2O cluster. In cell [3], it imports datasets from S3. In cell [4], it looks at the datasets by printing their summaries. The output for cell [2] shows the H2O cluster details:

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

The output for cell [3] shows the parse progress for the datasets:

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

The output for cell [4] shows the summary statistics for the datasets:

	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

Use Case 2

- Import large datasets directly from HDFS
- Train models on a multi-node cluster

www.h2o.ai/download



Version 3.14.0.2

Fast Scalable Machine Learning API
For Smarter Applications

DOWNLOAD AND RUN INSTALL IN R INSTALL IN PYTHON **INSTALL ON HADOOP** USE FROM MAVEN

Run H2O on Hadoop in just 3 steps

1. Download H2O for your version of Hadoop. This is a zip file that contains everything you need to get started.

```
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-cdh5.4.zip
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-cdh5.5.zip
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-cdh5.6.zip
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-cdh5.7.zip
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-cdh5.8.zip
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-cdh5.10.zip
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-hdp2.2.zip
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-hdp2.3.zip
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-hdp2.4.zip
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-hdp5.5.zip
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-mapr2.2.zip
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-mapr5.0.zip
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-mapr5.1.zip
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-mapr5.2.zip
wget https://h2o-release.s3.amazonaws.com/h2o/rel-weierstrass/2/h2o-3.14.0.2-lop4.2.zip
```

2. Unpack the zip file and launch a 6g instance of H2O:

```
unzip h2o-3.14.0.2-*.*.zip
cd h2o-3.14.0.2-*
hadoop jar h2odriver.jar -nodes 1 -mapperXmx 6g -output hdfsOutputDirName
```

Model Building



YARN

cloudera Hortonworks

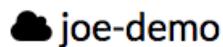
MAPR

Untitled Flow



CS

getCloud



CLOUD STATUS

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

NODES

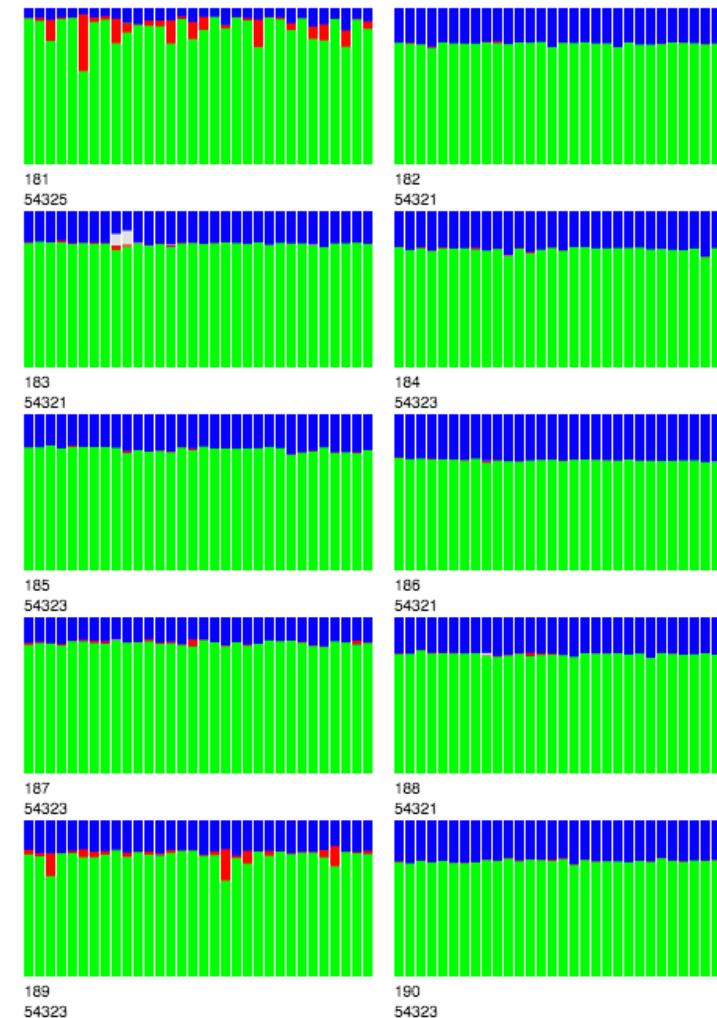
Name	Ping	Cores	Load	My CPU %	Sys	Shut Down	Data (Used/Total)	Data (% Cached)	GC (Free / Total / Max)	Disk (Free / Max)	Disk (% Free)
✓ 172.16.2.181:54323	a few seconds ago	32	6.110	0	8	-	40.603	33.82 GB / s	29.46 GB / NaN undefined / 29.58 GB	339.08 GB / 1.70 TB	19%
✓ 172.16.2.182:54321	a few seconds 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)

$10 \times 32 = 320$ Cores



Legend

Each bar represents one CPU.

Blue: idle time

Green: user time

Red: system time

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

Use Case 2 – Freddie Mac Loans



PERSPECTIVES

RESEARCH

BLOG

MEDIA ROOM

ABOUT

OUR BUSINESSES

Mortgage Rates

Insight

Outlook

Consumer Research

Indices

Additional Datasets

Single Family Loan-Level Dataset

As part of a larger effort to increase transparency, Freddie Mac is making available loan-level credit performance data on a portion of fully amortizing fixed-rate mortgages that the company purchased or guaranteed from 1999 to 2016.

The availability of this data will help investors build more accurate credit performance models in support of ongoing risk sharing initiatives highlighted by our regulator, the Federal Housing Finance Agency in the [2017 conservatorship scorecard \[PDF\]](#)

The dataset covers approximately 24.5 million fixed-rate mortgages (including HARP loans) originated between January 1, 1999 and June 30, 2016. Monthly loan performance data, including credit performance information up to and including property disposition, is being disclosed through December 31, 2016. Specific credit performance information in the dataset includes voluntary prepayments and loans that were Foreclosure Alternatives and REOs. Specific actual loss data in the dataset includes net sales proceeds, MI recoveries, non-MI recoveries, expenses, current deferred UPB, and due date of last paid installment.

The information in the historical dataset is unaudited and subject to change. Freddie Mac cannot guarantee the dataset is complete or error free. Read our [disclaimer](#). The historical dataset is not to be construed as securities disclosure.

In addition, Freddie Mac requires a [licensing agreement \[PDF\]](#) for commercial redistribution of the data in its Single-Family Loan-Level Dataset. Use of the dataset continues to be free for non-commercial, academic/research and for limited use, subject to the applicable terms and conditions.

Resources

General User Guide [PDF]

Additional Datasets

SINGLE-FAMILY LOAN-LEVEL DATASET

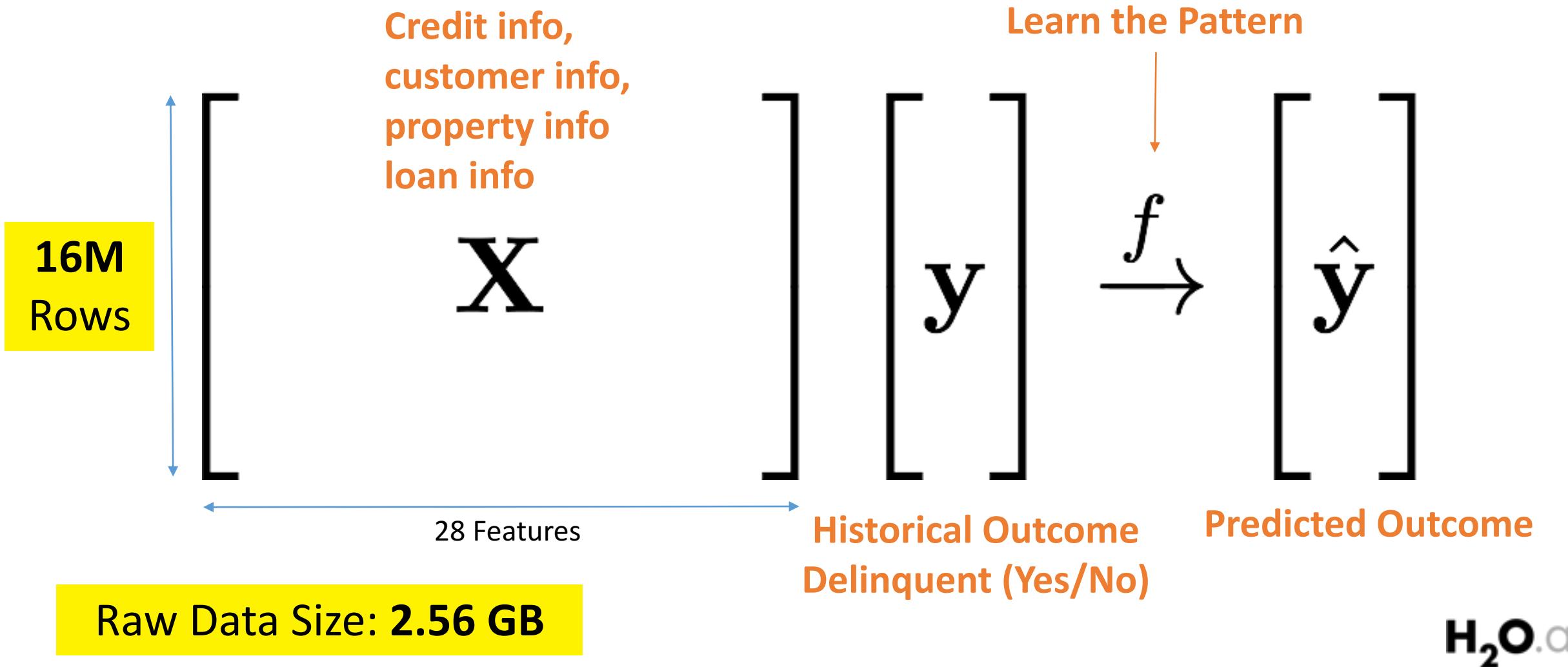
QUARTERLY REFINANCE STATISTICS

FEDERAL COST OF FUNDS INDEX

Authorized Licensees and Distributors



Learning from Freddie Mac (Loans) Data



16M Rows**Size (Raw): 2.58 GB****Compressed: 0.77 GB (\approx 30% of Raw)**

freddie_mac_loans

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

Rows	Columns	Compressed Size
16366856	29	769MB

COLUMN SUMMARIES

label	type	Missing	Zeros	+Inf	-Inf	min	max	mean	sigma	cardinality	Actions
credit_score	int	36902	0	0	0	300.0	850.0	731.9424	55.1077	·	Convert to enum
first_payment_date	int	0	0	0	0	199901.0	201609.0	200527.8795	391.7542	·	Convert to enum
first_time_homebuyer_flag	enum	3975291	10846414	0	0	0	1.0	0.1247	0.3304	2	Convert to numeric
maturity_date	int	0	0	0	0	200309.0	205405.0	203359.6542	493.3003	·	Convert to enum
msa	int	2327122	0	0	0	10180.0	49740.0	30497.4987	11351.0992	·	Convert to enum
mortgage_insurance_percentage	int	233246	13064396	0	0	0	55.0	4.7115	10.1318	·	Convert to enum
number_of_units	int	194	0	0	0	1.0	4.0	1.0251	0.2063	·	Convert to enum
occupancy_status	enum	0	730645	0	0	0	2.0	·	·	3	Convert to numeric
cltv	int	933	0	0	0	6.0	199.0	72.9324	17.0772	·	Convert to enum
dti	int	294607	0	0	0	1.0	65.0	33.7514	11.7248	·	Convert to enum
original_upb	int	0	0	0	0	5000.0	1403000.0	181878.1577	98894.4838	·	Convert to enum
original_loan-to-value_(ltv)	int	844	0	0	0	6.0	105.0	71.6317	16.6764	·	Convert to enum
original_interest_rate	real	0	0	0	0	2.2500	13.9500	6.0189	1.1074	·	·
channel	enum	0	679430	0	0	0	3.0	·	·	4	Convert to numeric
prepayment_penalty_mortgage_flag	enum	105587	16240319	0	0	0	1.0	0.0013	0.0359	2	Convert to numeric

Use Case 2 – Higgs Boson



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

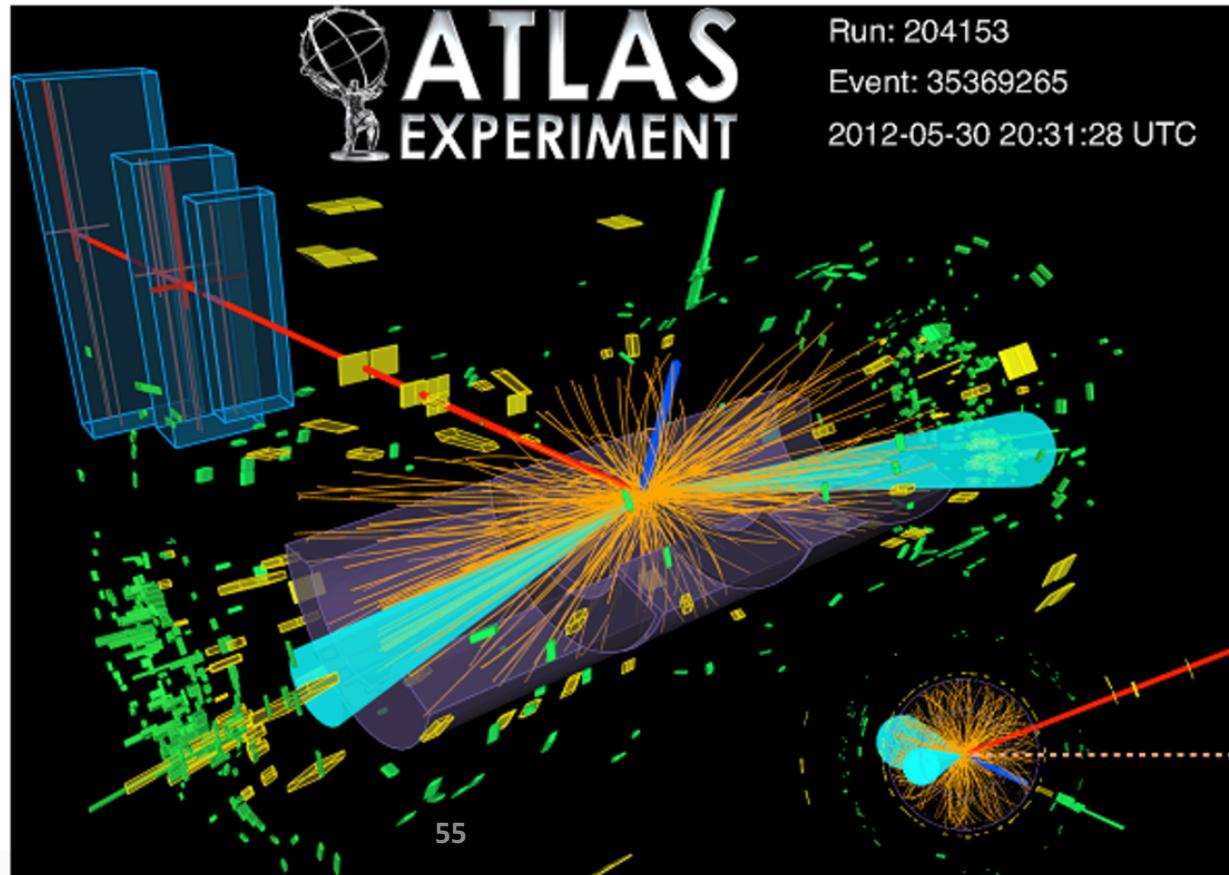
Evaluation

Prizes

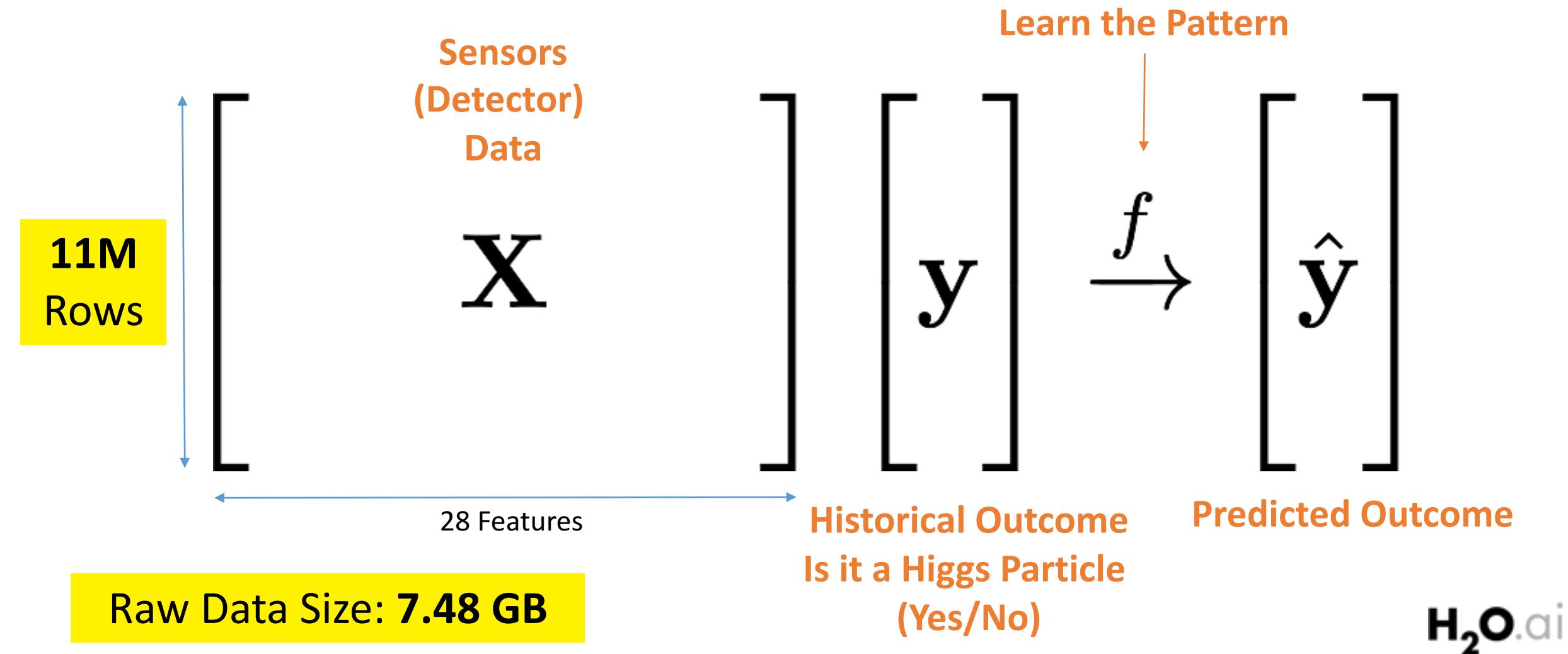
About The Sponsors

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

Use Case 2 Live Demo

- Import large datasets directly from HDFS
- Train models on a multi-node cluster

Q & A