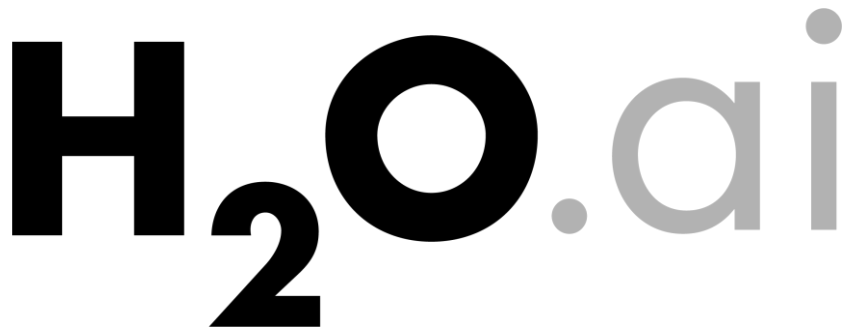
Introduction to Machine Learning with H_2O Flow



Jo-fai (Joe) Chow

Data Scientist

joe@h2o.ai

@matlabulous

H_2O at <RE/START> Rotterdam

10th April, 2017`

About Me

- Civil (Water) Engineer
  - 2010 – 2015
  - Consultant (UK)
    - Utilities
    - Asset Management
    - Constrained Optimization
  - Industrial PhD (UK)
    - Infrastructure Design Optimization
    - Machine Learning + Water Engineering
    - Discovered H_2O in 2014
- Data Scientist
  - 2015
  - Virgin Media (UK)
  - Domino Data Lab (Silicon Valley, US)
  - 2016 – Present
  - H_2O.ai (Silicon Valley, US)

About Me

Jo-fai Chow

woobe

Civil Engineer turned Data Scientist

H2O.ai
United Kingdom
jofai.chow@gmail.com
http://www.jofaichow.co.uk/

Organizations

Overview | Repositories 47 | Stars 402 | Followers 140 | Following 29

Popular repositories — Customize your pinned repositories

blenditbayes
Code used in my blog "Blend it like a Bayesian!"
R ★ 77 82

deepr
An R package to streamline the training, fine-tuning and predicting processes for deep learning based on 'darch' and 'deepnet'.
R ★ 41 16

rPlotter
Wrapper functions that make plotting in R a lot easier for beginners.
R ★ 30 4

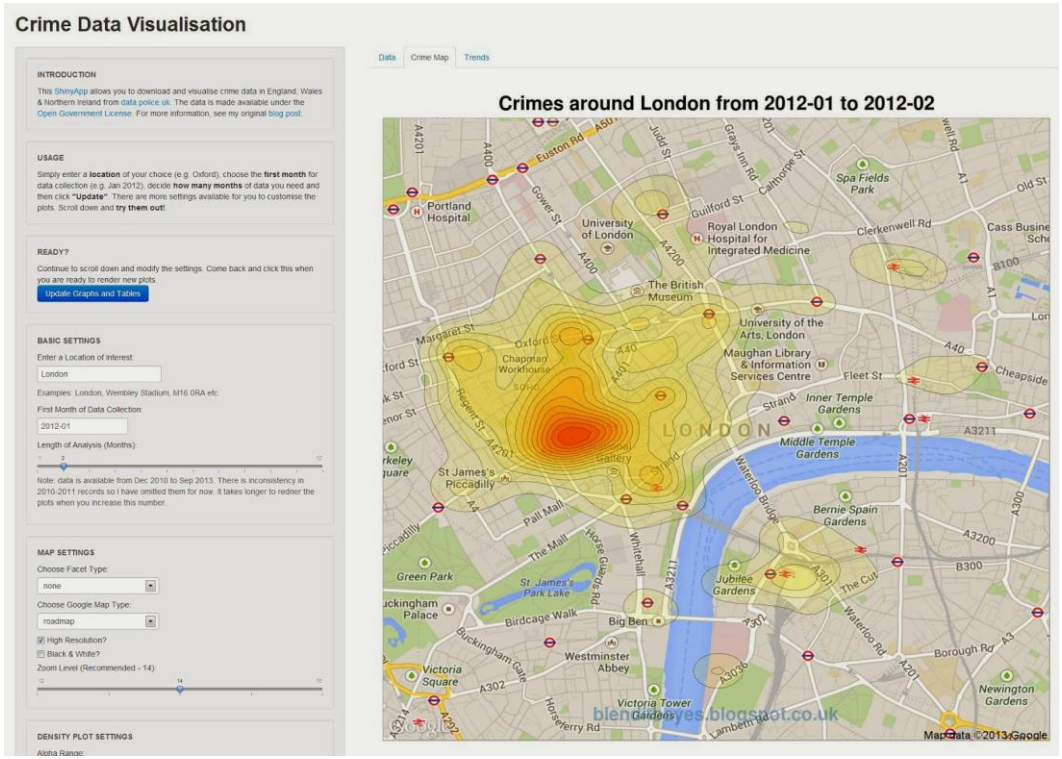
rCrimemap
This is the next generation of CrimeMap!
R ★ 22 8

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

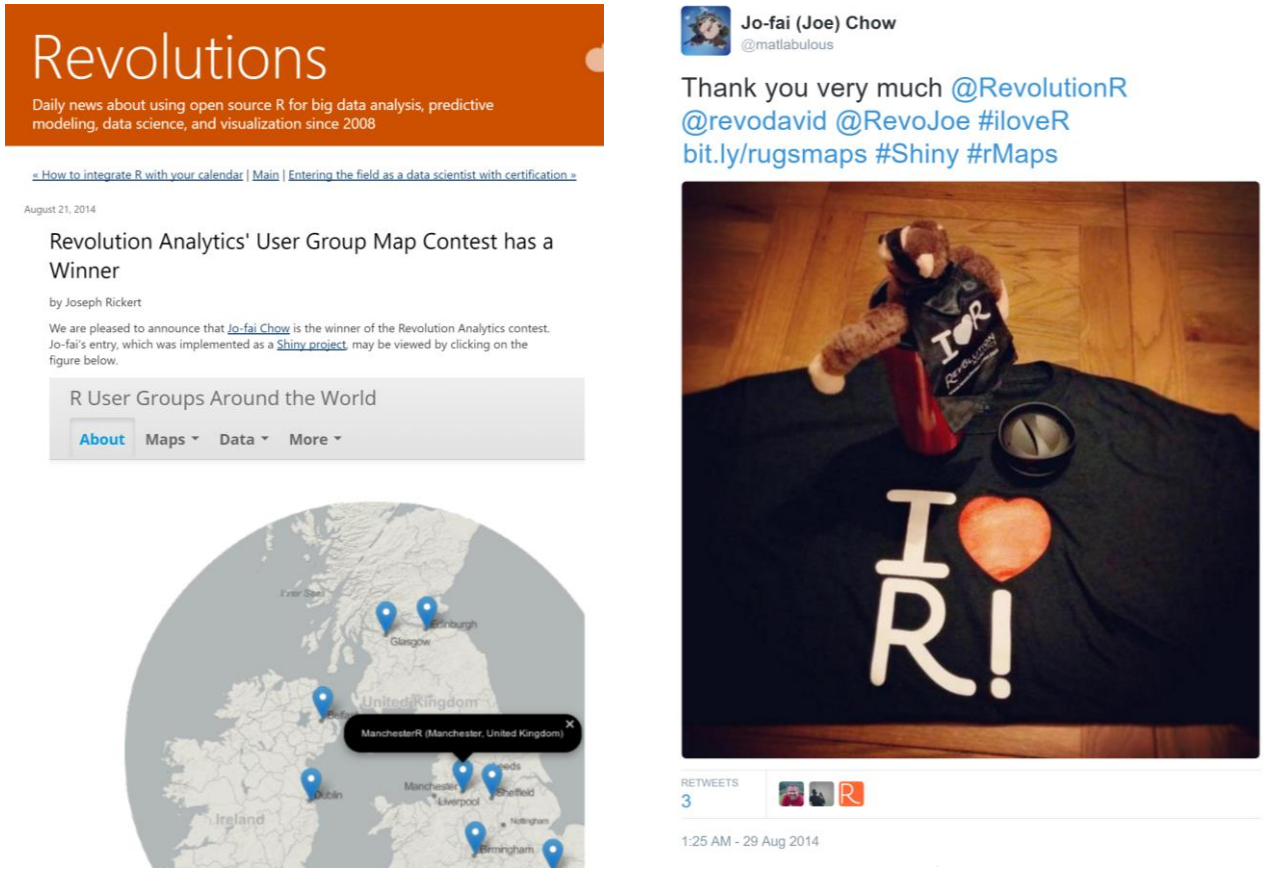
rApps
Repository for my R (Shiny) web applications.
R ★ 16 37

H2O.ai

About Me



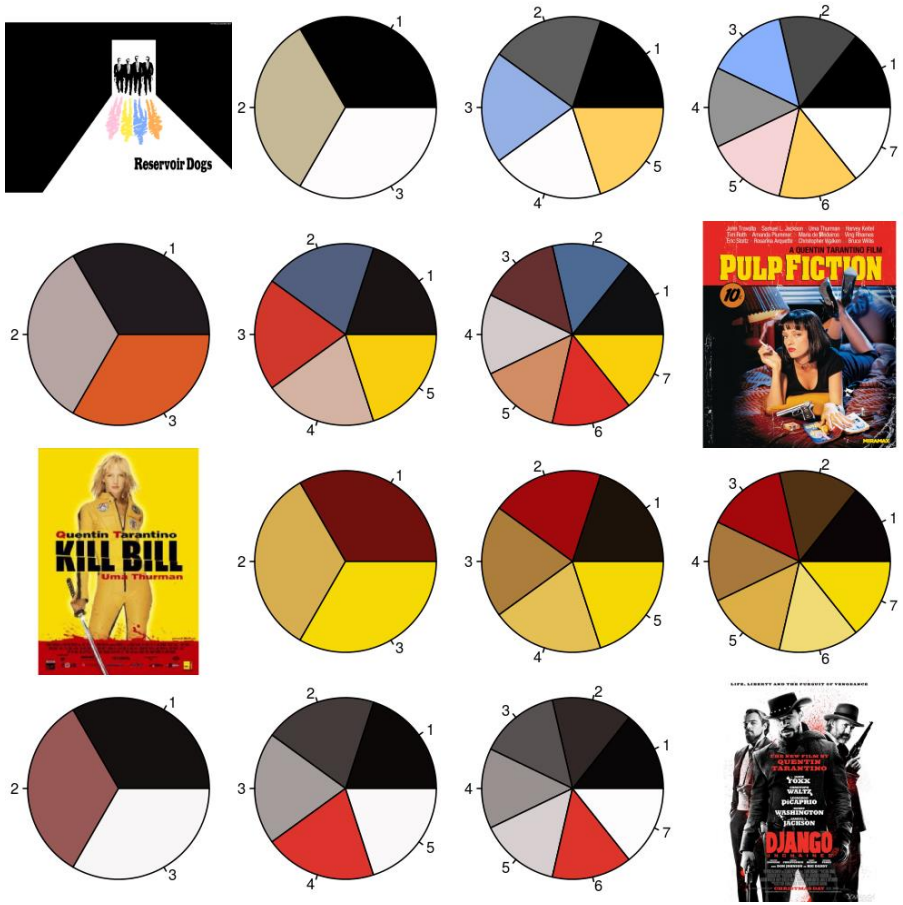
My First Data Viz & Shiny App Experience
CrimeMap (2013)



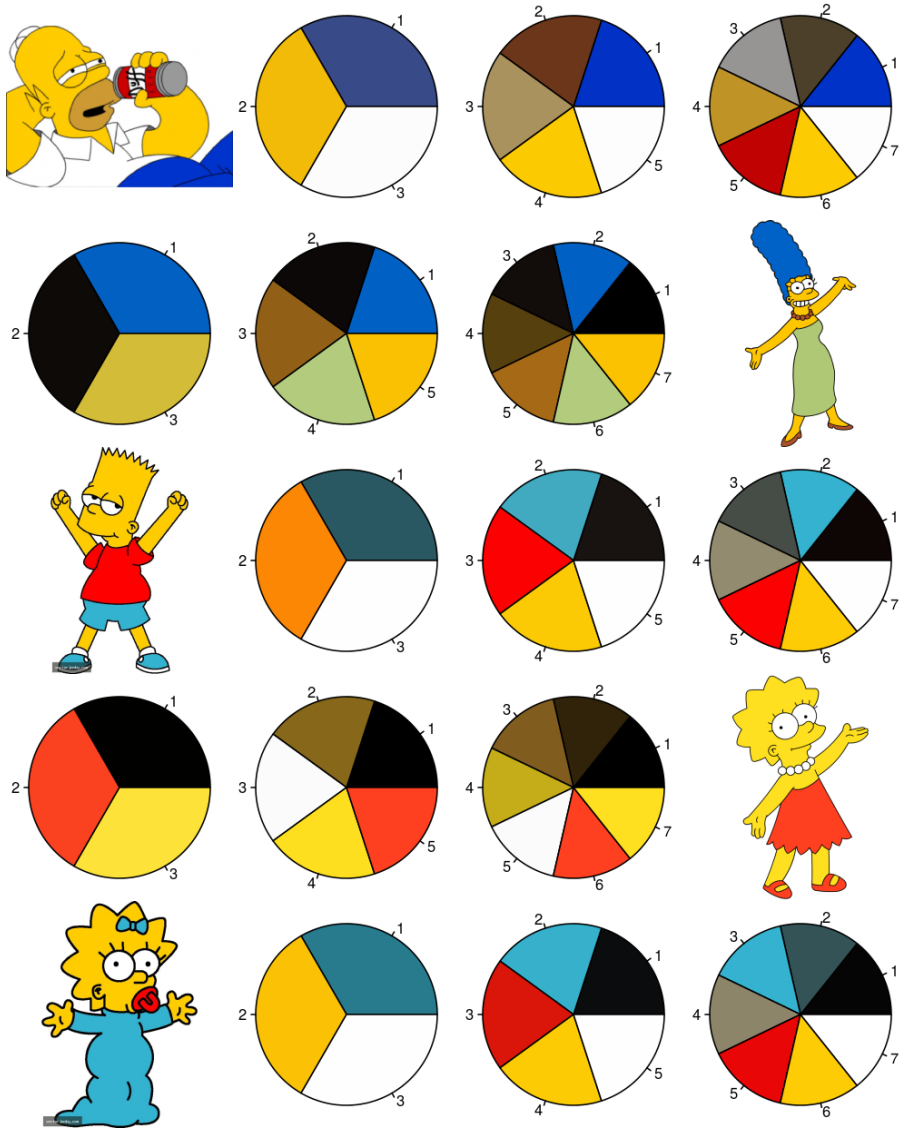
Revolution Analytics' Data Viz Contest
RUGSMAPS (2014)

H2O.ai

About Me

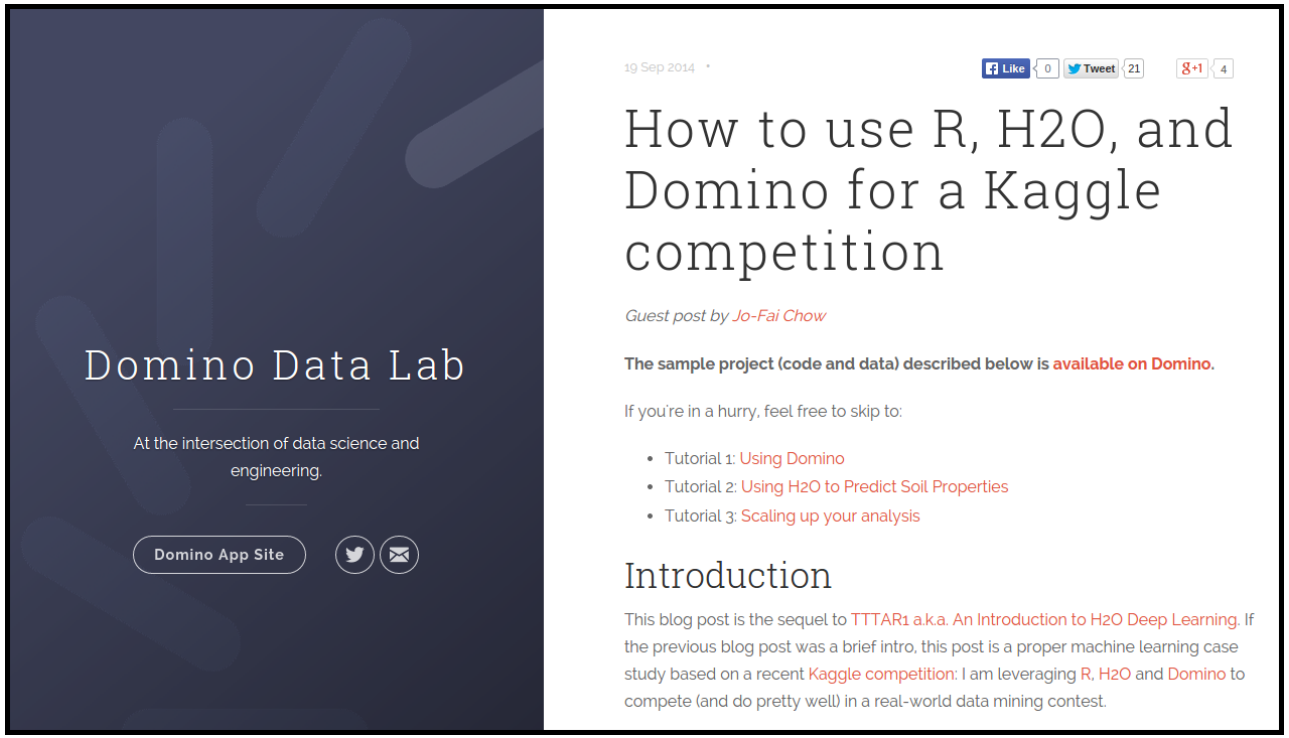


Developing R Packages for Fun
rPlotter (2014)



H2O.ai

About Me



R + H_2O + Domino for Kaggle

Guest Blog Post for Domino & H_2O (2014)

- The Long Story
  - bit.ly/joe_kaggle_story

H_2O.ai

Agenda

- About H_2O.ai
  - Company
  - Machine Learning Platform
- H_2O Flow (Web Interface) Example
  - Classification (Iris Dataset)
- H_2O with Python
  - Overview & Resource

<RE/START>

About H_2O.ai

Company Overview

Founded	2011 Venture-backed, debuted in 2012
Products	• H_2O Open Source In-Memory AI Prediction Engine • Sparkling Water • Steam
Mission	Operationalize Data Science, and provide a platform for users to build beautiful data products
Team	70 employees • Distributed Systems Engineers doing Machine Learning • World-class visualization designers
Headquarters	Mountain View, CA




Our Team

Joe

H2O.ai

Scientific Advisory Council

Dr. Trevor Hastie

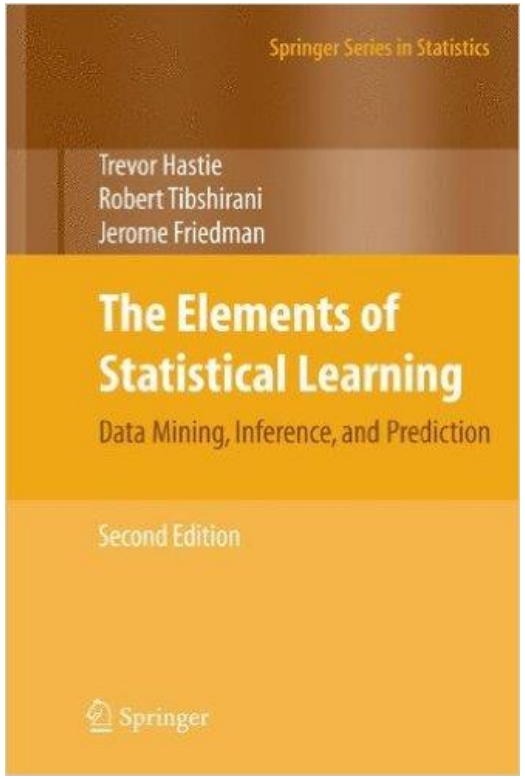
- John A. Overdeck Professor of Mathematics, Stanford University
- PhD in Statistics, Stanford University
- Co-author, *The Elements of Statistical Learning: Prediction, Inference and Data Mining*
- Co-author with John Chambers, *Statistical Models in S*
- Co-author, *Generalized Additive Models*

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

wenphan
@wenphan

Following

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

Figure 1. Magic Quadrant for Data Science Platforms

Source: Gartner (February 2017)

H2O.ai recognized for completeness of vision and ability to execute

We are thrilled to be named a Visionary among the 16 vendors included in Gartner's 2017 Magic Quadrant for Data Science Platforms. As a Visionary we believe we are positioned highest in Ability to Execute for companies of our size and scale.

Since 2011, our mission has been to democratize data science through open source AI and deep learning. Today, H2O.ai is focused on bringing AI to enterprises with a growing community of more than 8,500 organizations that depend on H2O for mission critical applications. H2O.ai was recently named CB Insights AI 100 and is used by 107 of the Fortune 500 companies.

Disclaimer: This graphic was published by Gartner, Inc. as part of a larger research document and should be evaluated in the context of the entire document. The Gartner document is available upon request from H2O.ai.

http://www.h2o.ai/gartner-magic-quadrant/

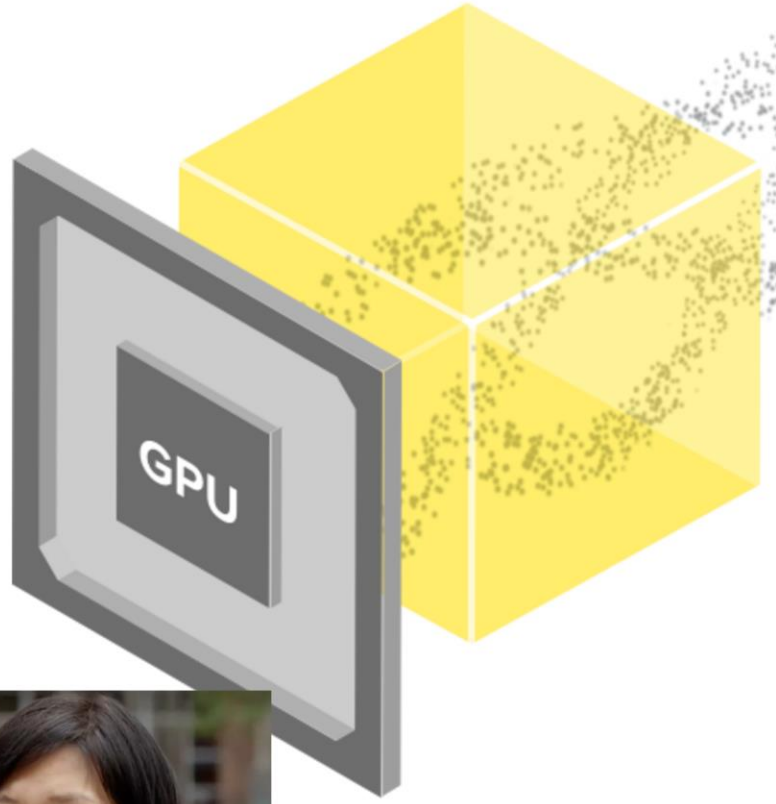
H2O.ai

Check out our website h2o.ai

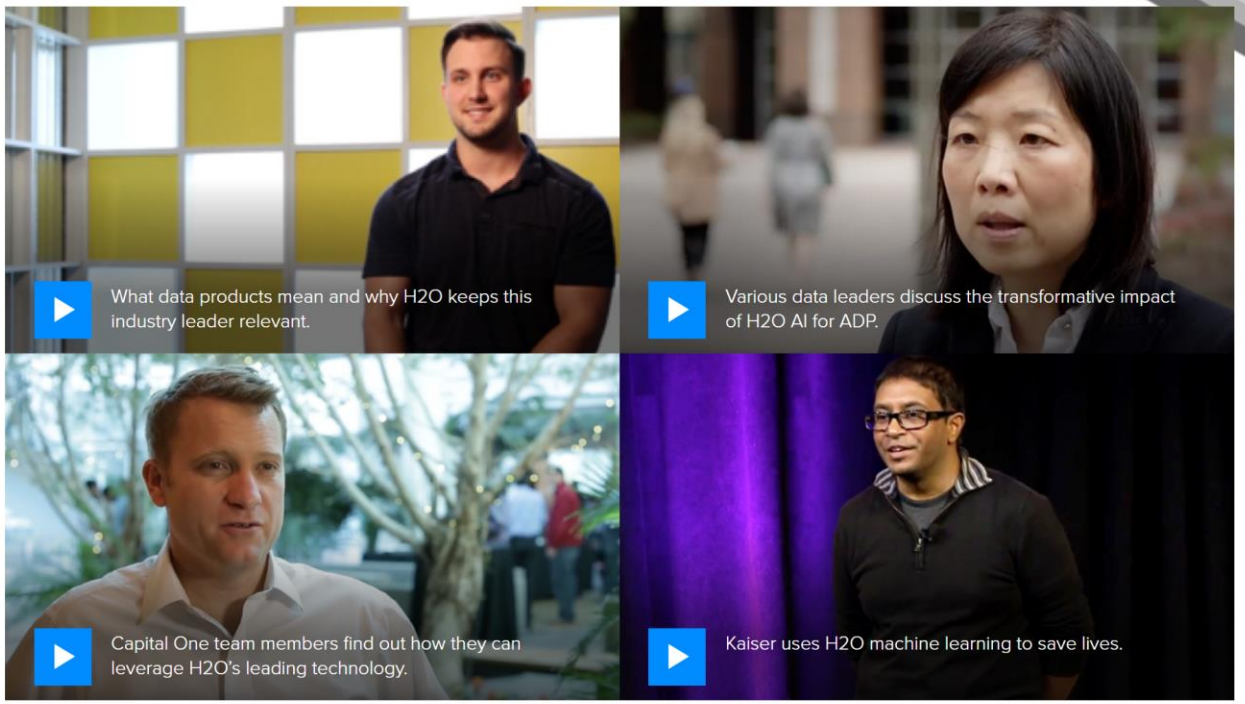
World Record Performance for AI

H2O.ai is accelerating both machine learning and deep learning on GPUs, providing enterprises opportunities to build better models and enable new use cases.

SEE DEEP WATER

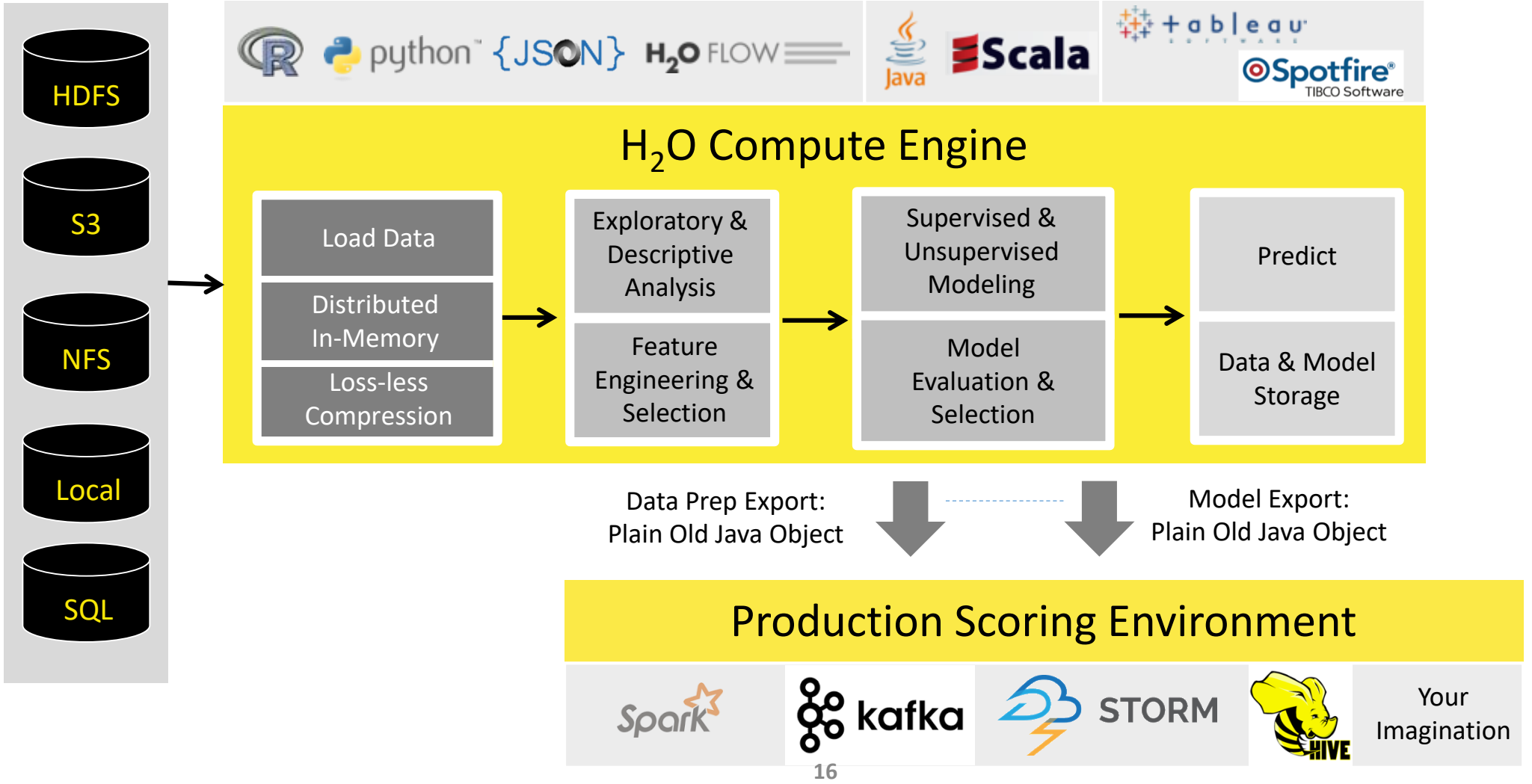


H2O in Action



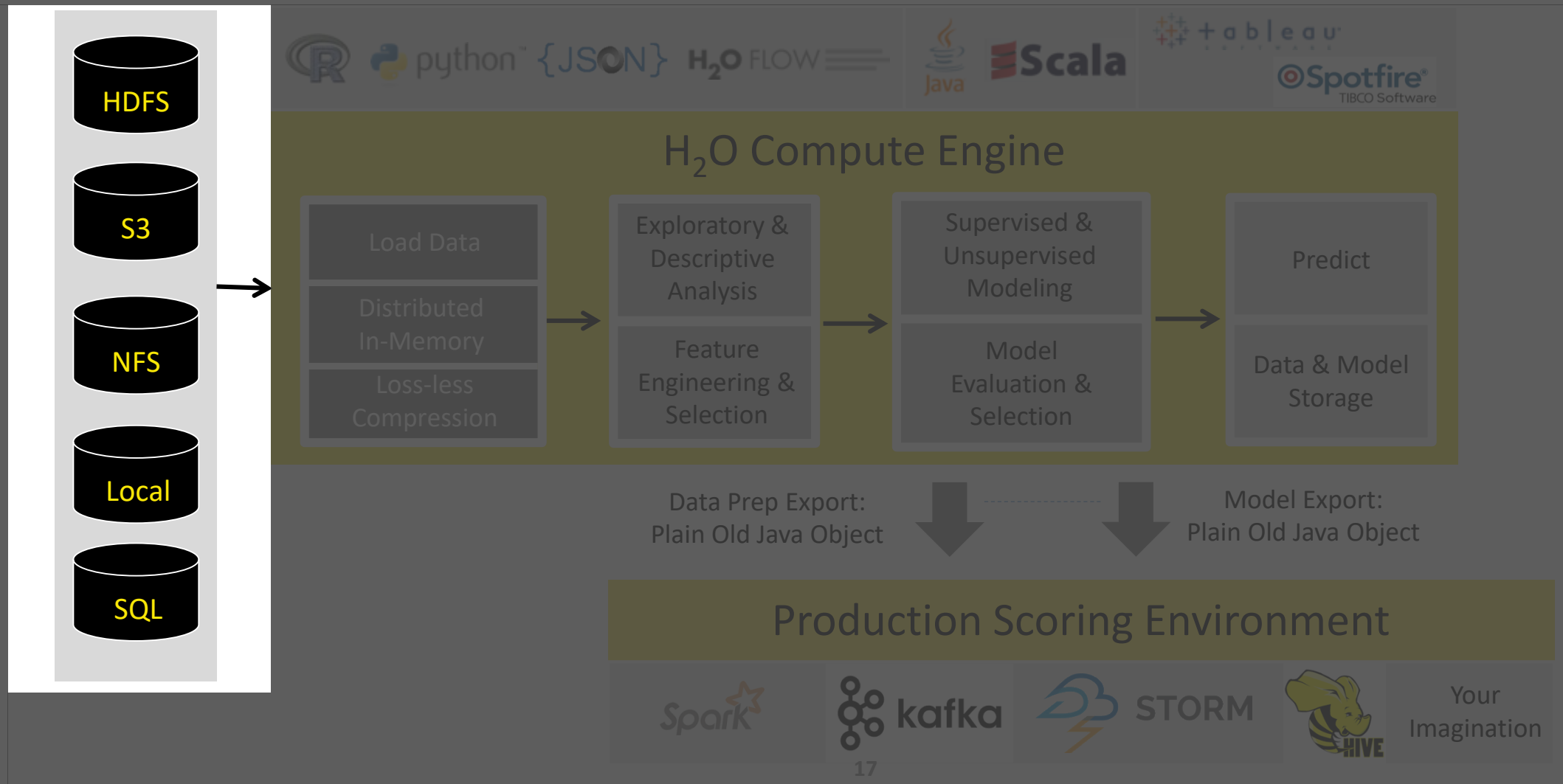
H_2O Machine Learning Platform

High Level Architecture



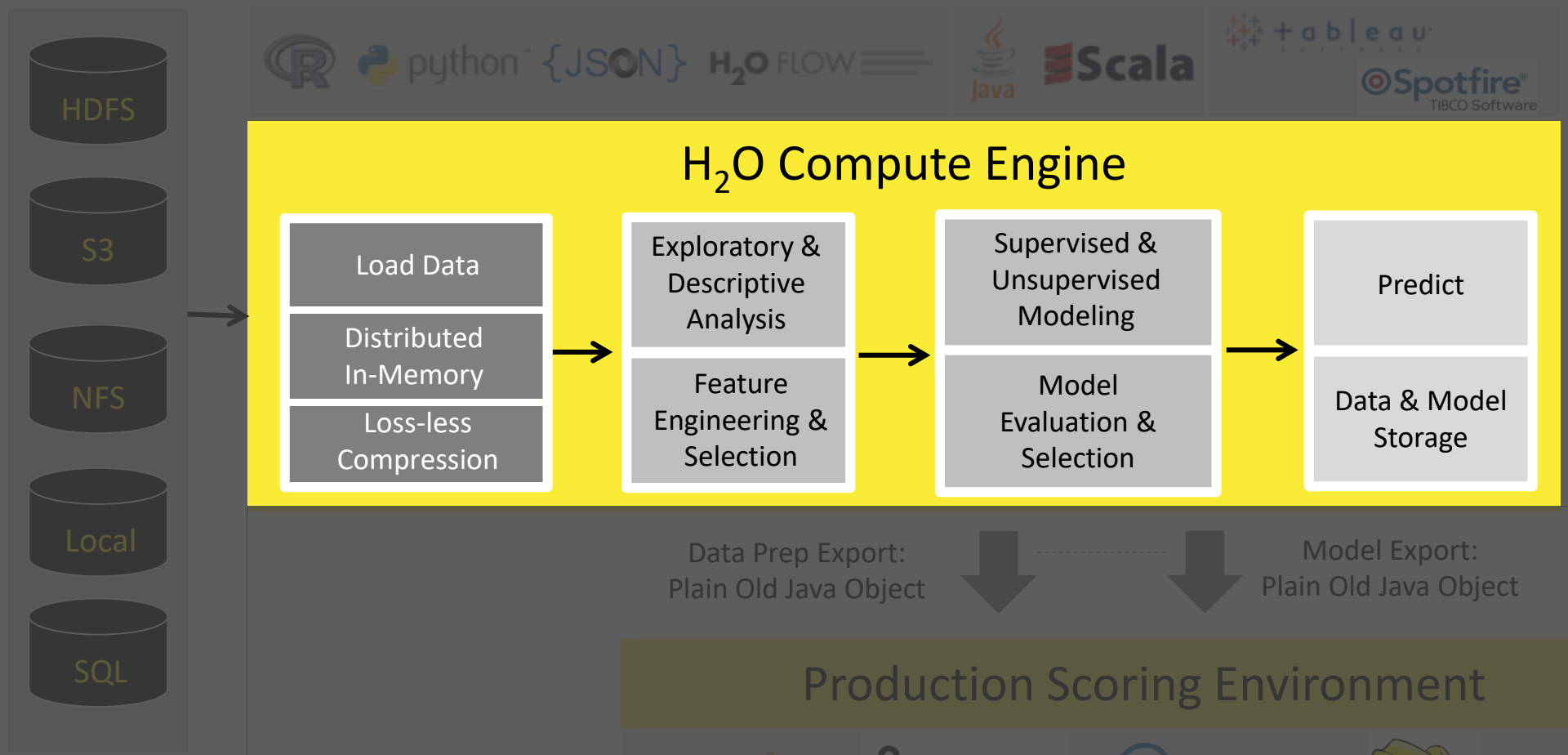
High Level Architecture

Import Data from Multiple Sources



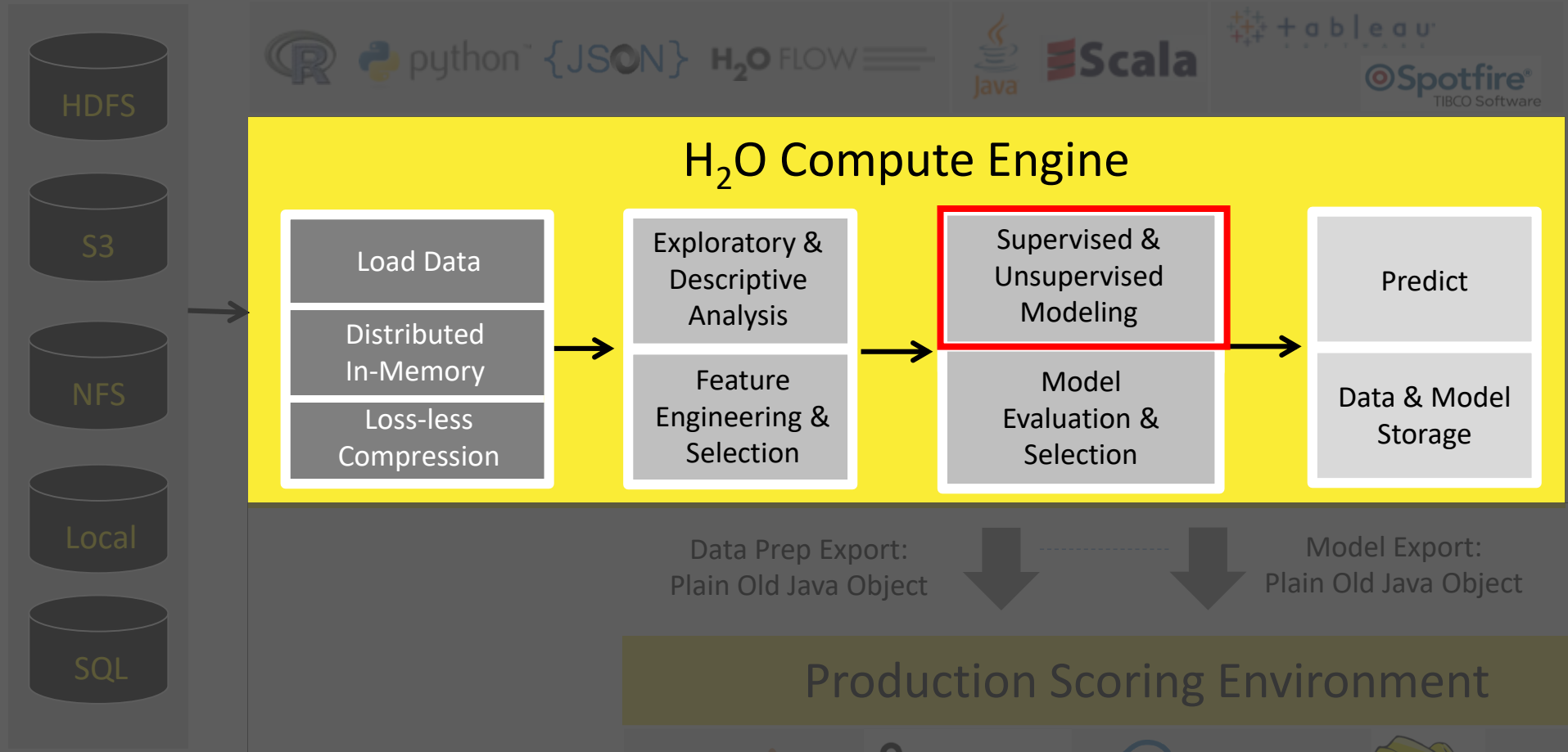
High Level Architecture

Fast, Scalable & Distributed Compute Engine Written in Java



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

H_2O Deep Learning in Action

116M rows, 6GB CSV file
800+ predictors (numeric + categorical)

airlines_all_selected_cols.hex

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

Rows	Columns	Compressed Size
116695259	12	2GB

Job

Run Time 00:00:36.712
Remaining Time 00:00:17.188
Type Model
Key deeplearning-dd2f42f7-81f7-42e8-9d98-e34437309828
Description DeepLearning
Status RUNNING
Progress 69%

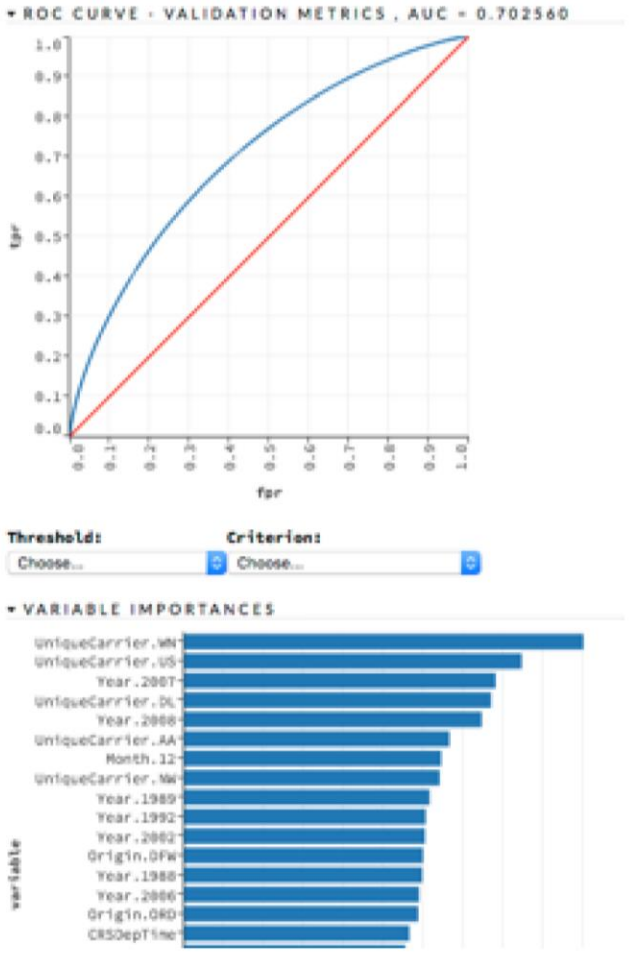
Iterations: 12. Epochs: 0.628821. Speed: 2,243,735 samples/sec. Estimated time left: 21.849 sec

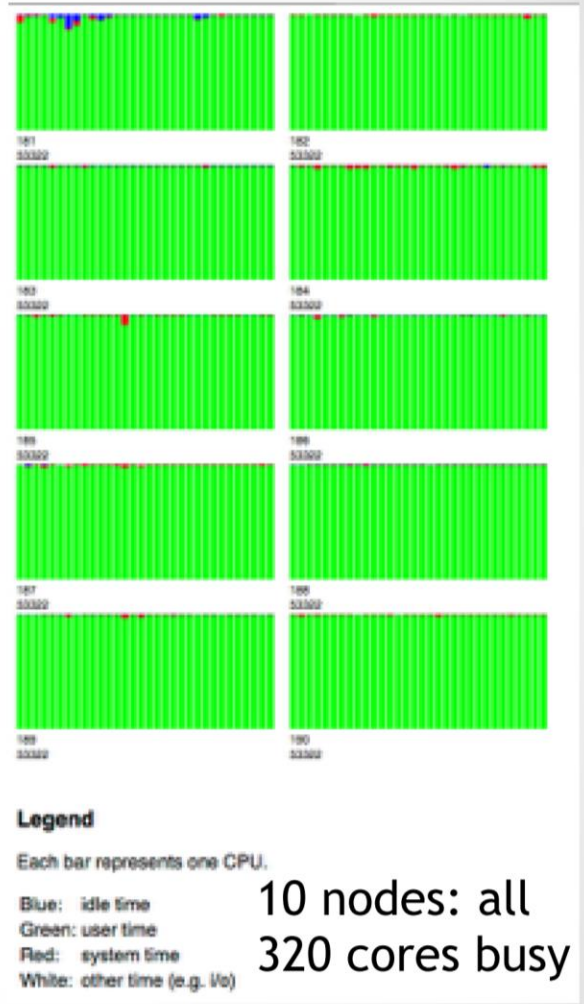
Actions: View | Cancel Job

model trained in <1 min:
2M+ samples/second

▾ OUTPUT - STATUS OF NEURON LAYERS (PREDICTING ISDEPDELAYED, 2-CLASS CLASSIFICATION, BERNOULLI DISTRIBUTION, CROSSENTROPY LOSS, 17,462 WEIGHTS/BIASES, 221.3 KB, 106,585,365 TRAINING SAMPLES, MINI-BATCH SIZE 1)

layer	units	type	dropout	l1	l2	mean_rate	rate_RMS	momentum	mean_weight	weight_RMS	mean_bias	bias_RMS
1	807	Input	0									
2	20	Rectifier	0	0	0	0.0493	0.2020	0	-0.0021	0.2111	-0.9139	1.0036
3	20	Rectifier	0	0	0	0.0197	0.0227	0	-0.1033	0.5362	-1.3988	1.5259
4	20	Rectifier	0	0	0	0.0517	0.0446	0	-0.1575	0.3068	-0.6846	0.6846
5	20	Rectifier	0	0	0	0.0761	0.0844	0	-0.0374	0.2275	-0.2647	0.2481
6	2	Softmax		0	0	0.0161	0.0083	0	0.0741	0.7260	0.4269	0.2056





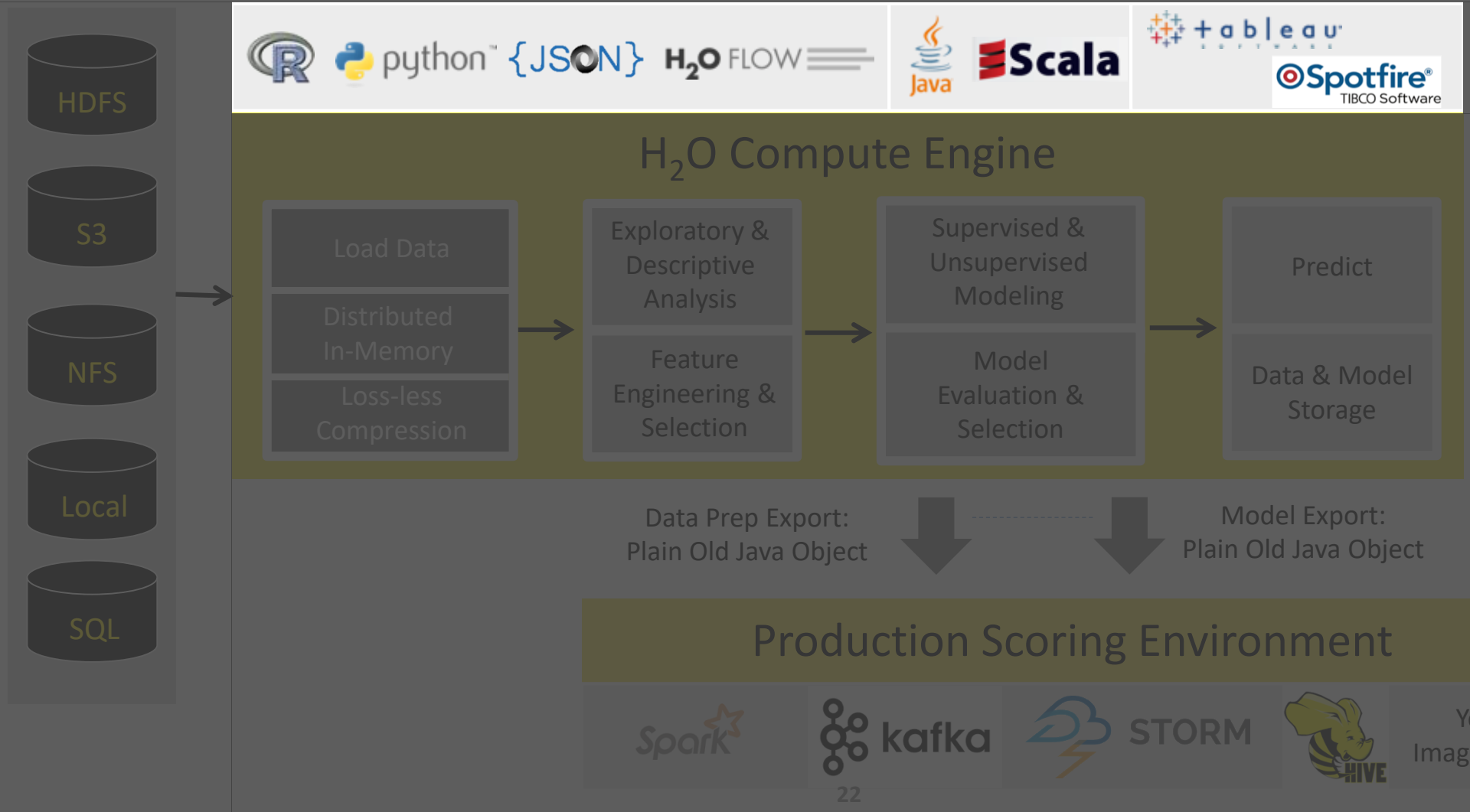
10 nodes: all
320 cores busy

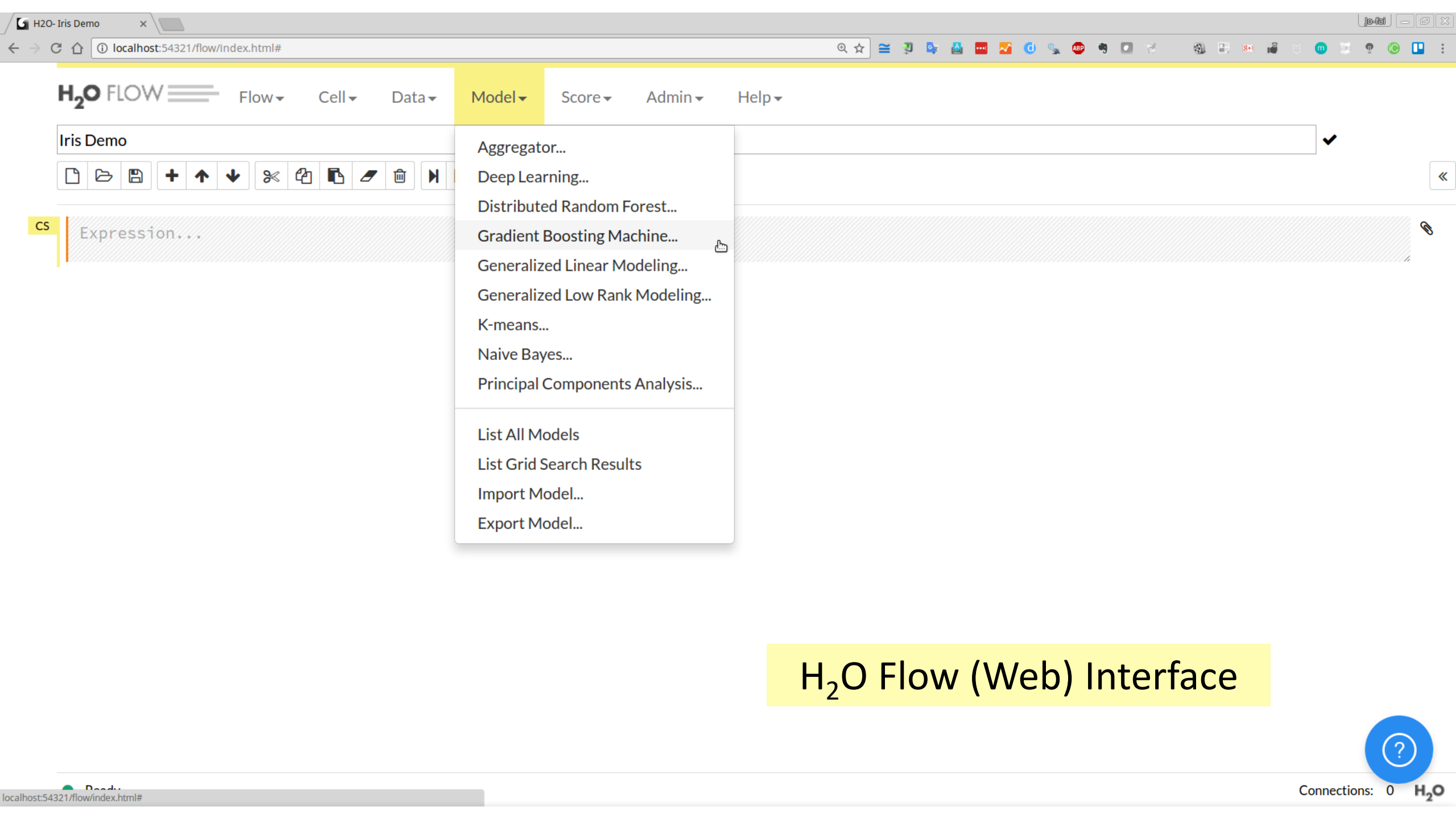
real-time, interactive
model inspection in Flow

H_2O.ai **Deep Learning Model**

High Level Architecture

Multiple Interfaces





H_2O Flow (Web) Interface

H_2O + 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)
```

H_2O + Python

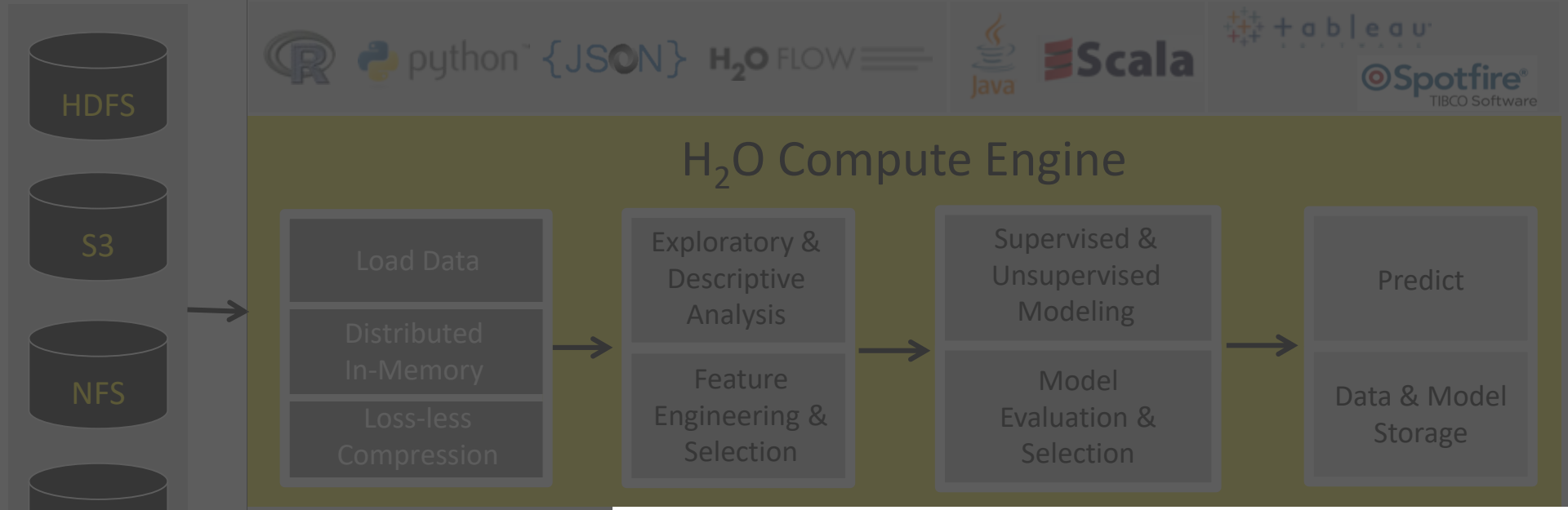
Gradient Boosting Machines

```
# 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



Data Prep Export:
Plain Old Java Object

Model Export:
Plain Old Java Object

Production Scoring Environment

Spark | kafka | STORM | HIVE | Your Imagination

docs.h2o.ai

Languages

R

- Quick Start Video - R
- R Package Docs
- R Booklet
- Examples and Demos
- R FAQ
- Ensemble R Package Readme
- RSparkling Readme
- Migrating from H2O-2

Python

- Quick Start Video - Python
- Python Module Docs
- Python Booklet
- Examples and Demos
- Python FAQ
- PySparkling Readme 2.0 | 1.6
- skutil Docs

Java

- POJO and MOJO Model Javadoc
- H2O Core Javadoc
- H2O Algorithms Javadoc

Scala

Sparkling Water API	2.0	1.6
Sparkling Water Scaladoc	2.0	1.6
H2O Scaladoc	2.11	2.10

Tutorials, Examples, & Presentations

Tutorials and Blogs

- H2O Tutorials HTML | PDF
- H2O Blogs
- H2O University

Use Case Examples

Chicago crime prediction	R	Python	ScalaSW	PySW
Airlines delays prediction	R	Python	ScalaSW	PySW
Lending Club loan prediction	R	Python	ScalaSW	PySW
Ham or Spam	R	Python	ScalaSW	PySW
Prediction with prostate dataset	R	Python	ScalaSW	PySW

Presentations

- H2O Meetups
- H2O World 2014 Videos
- H2O World 2015 Videos
- Open Tour Chicago Videos
- Open Tour NYC Videos
- Open Tour Dallas Videos

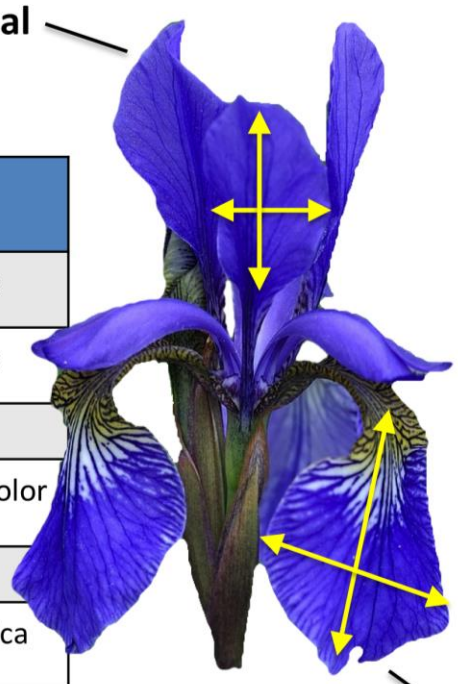
Classification Example

Iris Dataset

Simple Demo – Iris

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica



Class labels
(targets)

Features
(attributes, measurements, dimensions)

H2O.ai

Start a Local H_2O Cluster

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

DOWNLOAD H_2O

Get started with H_2O in 3 easy steps

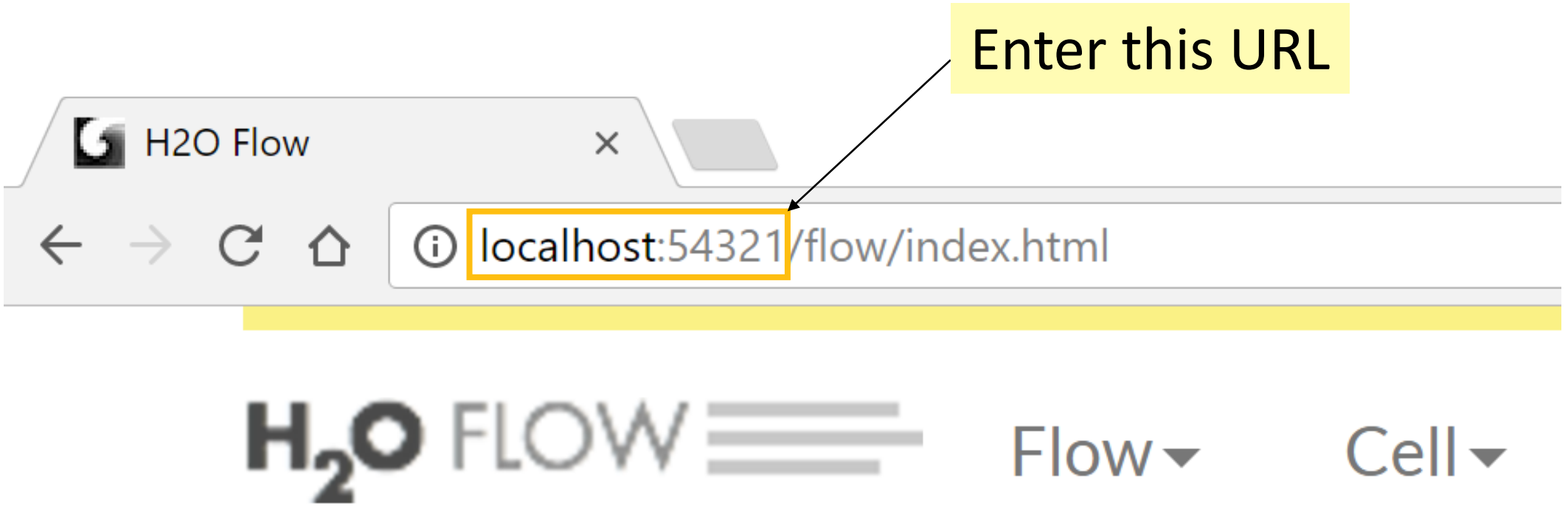
1. Download H_2O. This is a zip file that contains everything you need to get started.

2. From your terminal, run:

```
cd ~/Downloads
unzip h2o-3.10.4.3.zip
cd h2o-3.10.4.3
java -jar h2o.jar
```

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

Connect to H_2O Flow



NOTE: Localhost:54321 is the same as 127.0.0.1:54321

H2O.ai

H2O FLOW Flow Cell Data Model Score Admin Help

Untitled Flow

CS

```
assist
```

235ms

Assistance

	Routine	Description
	importFiles	Import file(s) into H_2O
	getFrames	Get a list of frames in H_2O
	splitFrame	Split a frame into two or more frames
	mergeFrames	Merge two frames into one
	getModels	Get a list of models in H_2O
	getGrids	Get a list of grid search results in H_2O
	getPredictions	Get a list of predictions in H_2O
	getJobs	Get a list of jobs running in H_2O
	buildModel	Build a model
	importModel	Import a saved model
	predict	Make a prediction

Check out the examples

OUTLINE FLOWS CLIPS HELP

Help

Using Flow for the first time?

Quickstart Videos

Or, view example Flows to explore and learn H_2O.

STAR H2O ON GITHUB!

Star 1,897

GENERAL

- Flow Web UI ...
- ... Importing Data
- ... Building Models
- ... Making Predictions
- ... Using Flows
- ...Troubleshooting Flow

EXAMPLES

Flow packs are a great way to explore and learn H_2O. Try out these Flows and run

Ready

Connections: 0 H2O

MD

GBM Tutorial

The purpose of this tutorial is to walk new users through a GBM analysis in H2O Flow.

Those who have never used H2O before should refer to Getting Started for additional instructions on how to run H2O Flow.

Getting Started

This tutorial uses a publicly available data set that can be found at:

http://archive.ics.uci.edu/ml/datasets/Arrhythmia.

The original data are the Arrhythmia data set made available by UCI Machine Learning repository. They are composed of 452 observations and 279 attributes.

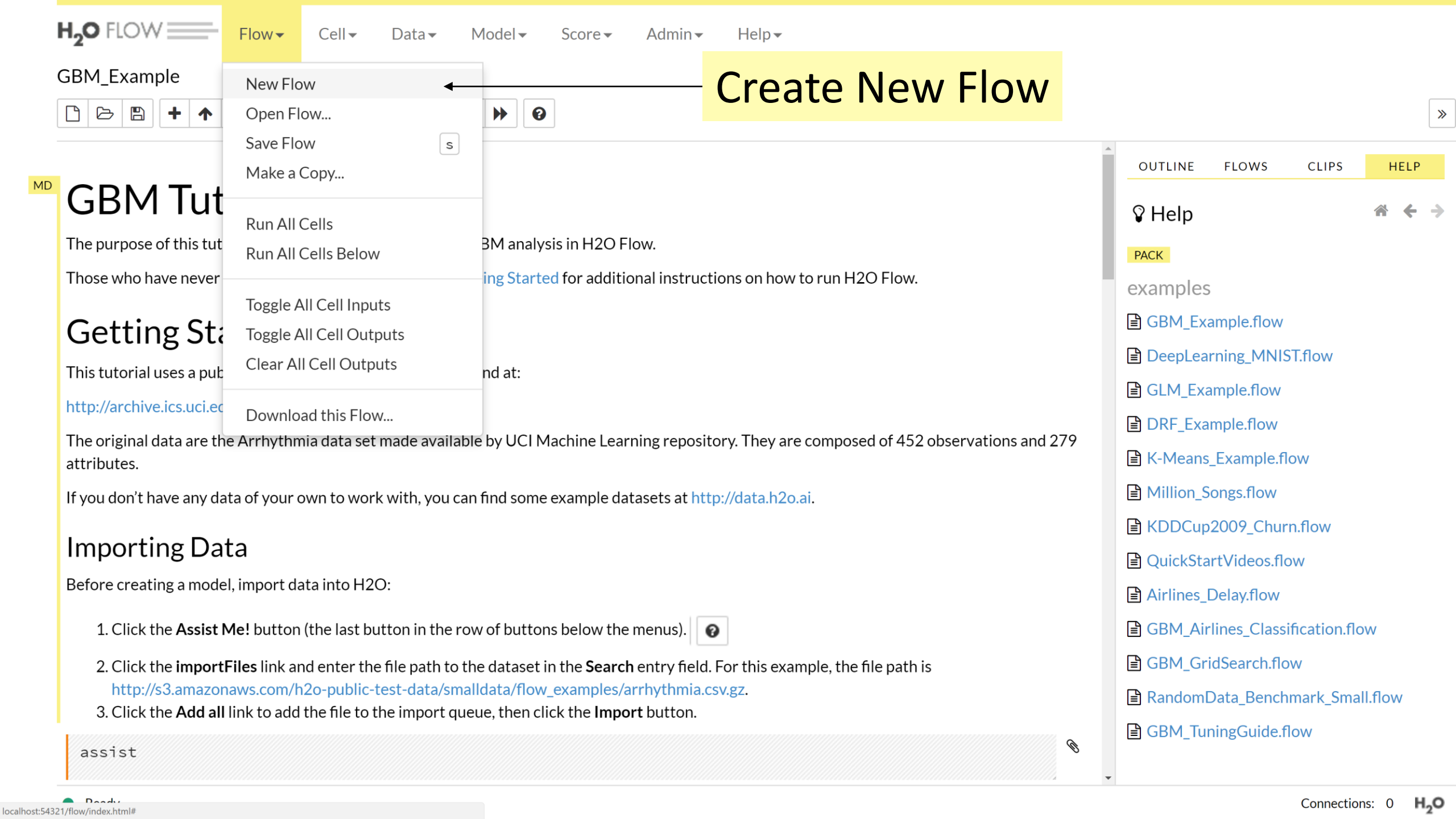
If you don't have any data of your own to work with, you can find some example datasets at http://data.h2o.ai.

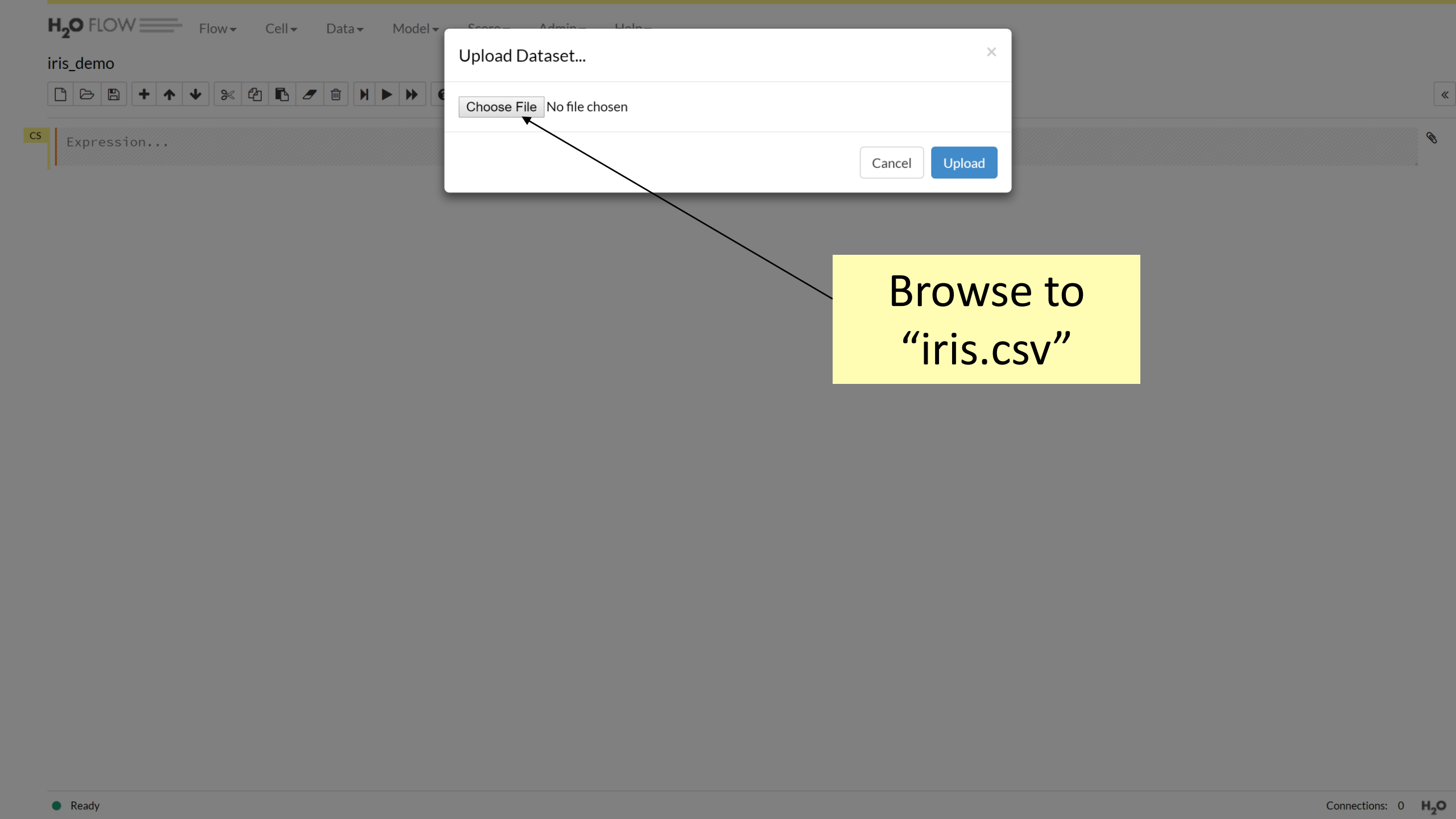
Importing Data

Before creating a model, import data into H2O:

1. Click the **Assist Me!** button (the last button in the row of buttons below the menus).
2. Click the **importFiles** link and enter the file path to the dataset in the **Search** entry field. For this example, the file path is http://s3.amazonaws.com/h2o-public-test-data/smalldata/flow_examples/arrhythmia.csv.gz.
3. Click the **Add all** link to add the file to the import queue, then click the **Import** button.

```
assist
```


H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
GBM_Example
New Flow
Open Flow...
Save Flow
s
Make a Copy...
Run All Cells
Run All Cells Below
Toggle All Cell Inputs
Toggle All Cell Outputs
Clear All Cell Outputs
Download this Flow...
Create New Flow
MD
GBM Tut
The purpose of this tut
BM analysis in H2O Flow.
Those who have never
ing Started for additional instructions on how to run H2O Flow.
Getting Sta
This tutorial uses a pub
nd at:
http://archive.ics.uci.ed
The original data are the Arrhythmia data set made available by UCI Machine Learning repository. They are composed of 452 observations and 279 attributes.
If you don't have any data of your own to work with, you can find some example datasets at http://data.h2o.ai.
Importing Data
Before creating a model, import data into H2O:
1. Click the Assist Me! button (the last button in the row of buttons below the menus).
2. Click the importFiles link and enter the file path to the dataset in the Search entry field. For this example, the file path is http://s3.amazonaws.com/h2o-public-test-data/smalldata/flow_examples/arrhythmia.csv.gz.
3. Click the Add all link to add the file to the import queue, then click the Import button.
assist
OUTLINE
FLOWS
CLIPS
HELP
Help
PACK
examples
GBM_Example.flow
DeepLearning_MNIST.flow
GLM_Example.flow
DRF_Example.flow
K-Means_Example.flow
Million_Songs.flow
KDDCup2009_Churn.flow
QuickStartVideos.flow
Airlines_Delay.flow
GBM_Airlines_Classification.flow
GBM_GridSearch.flow
RandomData_Benchmark_Small.flow
GBM_TuningGuide.flow
Connections: 0
H2O
localhost:54321/flow/index.html#


H2O FLOW
Flow
Cell
Data
Model
iris_demo
CS
Expression...
Upload Dataset...
Choose File
No file chosen
Cancel
Upload
Browse to
“iris.csv”
Ready
Connections: 0

H2O FLOW Flow Cell Data Model Score Admin Help

File uploaded successfully!

iris_demo

```
Expression...
```

CS

```
setupParse source_frames: [ "iris.csv"]
```

173ms

Setup Parse

PARSE CONFIGURATION

Sources iris.csv

ID Key_Frame__iris1.hex

Parser CSV

Separator ,: '44'

Column Headers Auto

First row contains column names

First row contains data

Options Enable single quotes as a field quotation character

Delete on done

EDIT COLUMN NAMES AND TYPES

Search by column name...

1	sepal_length	Numeric	5.1	4.9	4.7	4.6	5	5.4	4.6	5	4.4
2	sepal_width	Numeric	3.5	3	3.2	3.1	3.6	3.9	3.4	3.4	2.9
3	petal_length	Numeric	1.4	1.4	1.3	1.5	1.4	1.7	1.4	1.5	1.4
4	petal_width	Numeric	0.2	0.2	0.2	0.2	0.2	0.4	0.3	0.2	0.2
5	species	Enum	setosa	setosa	setosa	setosa	setosa	setosa	setosa	setosa	setosa

Previous page Next page

Parse

Click “Parse”

Ready

Connections: 0 H2O

H2O FLOW Flow Cell Data Model Score Admin Help

iris_demo

Parse

CS

```
parseFiles
  source_frames: ["iris.csv"]
  destination_frame: "Key_Frame__iris1.hex"
  parse_type: "CSV"
  separator: 44
  number_columns: 5
  single_quotes: false
  column_names: ["sepal_length","sepal_width","petal_length","petal_width","species"]
  column_types: ["Numeric","Numeric","Numeric","Numeric","Enum"]
  delete_on_done: true
  check_header: 1
  chunk_size: 4194304
```

1.1s

Job

Run Time	00:00:00.12
Remaining Time	00:00:00.0
Type	Frame
Key	Key_Frame__iris1.hex
Description	Parse
Status	DONE
Progress	100%
	Done.
Actions	View

Click “View”

Ready

Connections: 0 H2O

iris_demo

Key Key_Frame__iris1.hex
Description Parse
Status DONE
Progress 100%
Done.
Actions View

CS

```
getFrameSummary "Key_Frame__iris1.hex"
```

80ms

Key_Frame__iris1.hex

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

Rows	Columns	Compressed Size
150	5	2KB

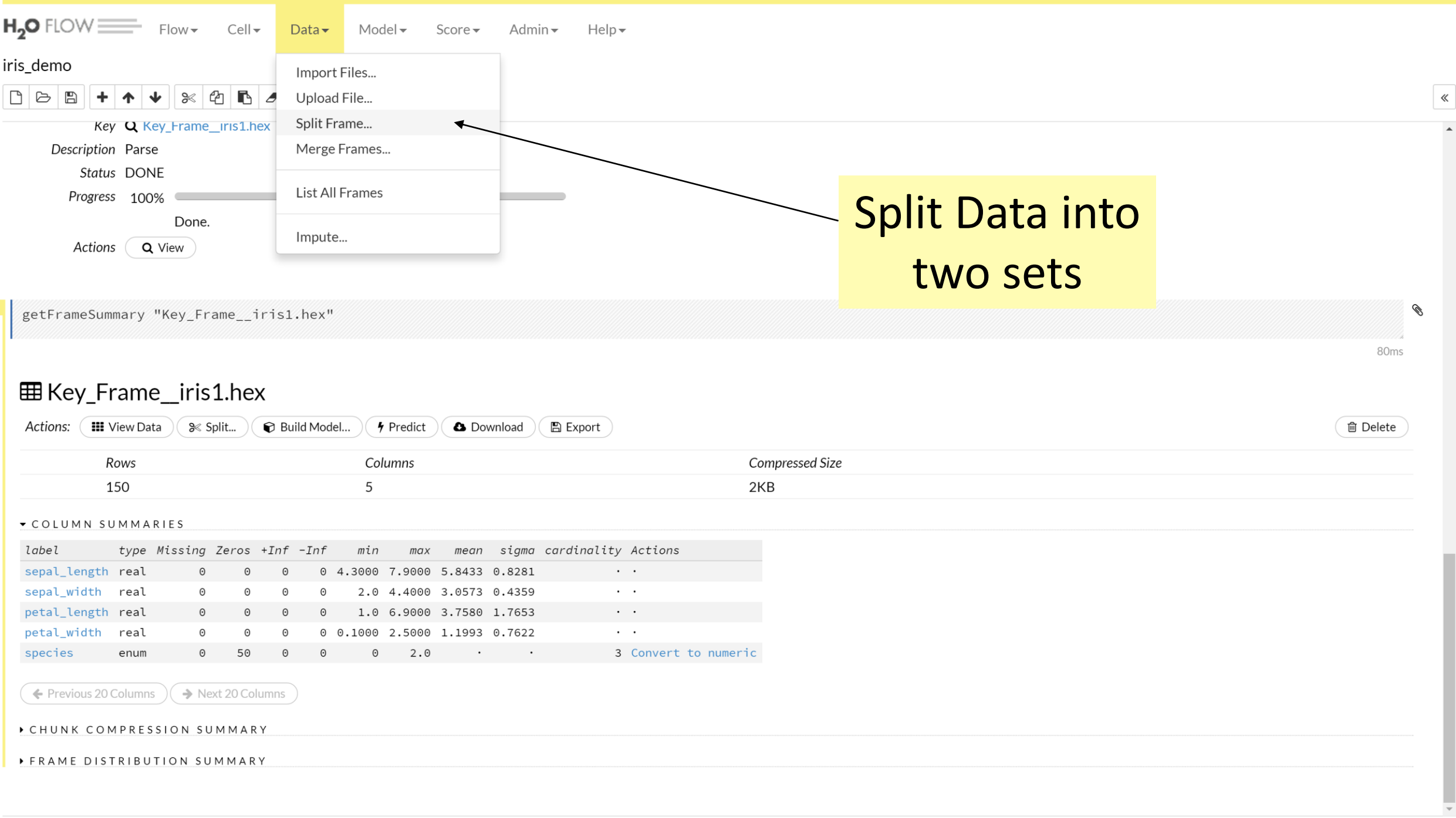
COLUMN SUMMARIES

label	type	Missing	Zeros	+Inf	-Inf	min	max	mean	sigma	cardinality	Actions
sepal_length	real	0	0	0	0	4.3000	7.9000	5.8433	0.8281	·	·
sepal_width	real	0	0	0	0	2.0	4.4000	3.0573	0.4359	·	·
petal_length	real	0	0	0	0	1.0	6.9000	3.7580	1.7653	·	·
petal_width	real	0	0	0	0	0.1000	2.5000	1.1993	0.7622	·	·
species	enum	0	50	0	0	0	2.0	·	·	3	Convert to numeric

Previous 20 Columns | Next 20 Columns

CHUNK COMPRESSION SUMMARY

FRAME DISTRIBUTION SUMMARY


H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
iris_demo
Import Files...
Upload File...
Split Frame...
Merge Frames...
List All Frames
Impute...
Key Key_Frame__iris1.hex
Description Parse
Status DONE
Progress 100%
Done.
Actions View
Split Data into two sets
CS
getFrameSummary "Key_Frame__iris1.hex"
80ms
Key_Frame__iris1.hex
Actions: View Data Split... Build Model... Predict Download Export Delete
Rows 150
Columns 5
Compressed Size 2KB
COLUMN SUMMARIES
label type Missing Zeros +Inf -Inf min max mean sigma cardinality Actions
sepal_length real 0 0 0 0 4.3000 7.9000 5.8433 0.8281 · ·
sepal_width real 0 0 0 0 2.0 4.4000 3.0573 0.4359 · ·
petal_length real 0 0 0 0 1.0 6.9000 3.7580 1.7653 · ·
petal_width real 0 0 0 0 0.1000 2.5000 1.1993 0.7622 · ·
species enum 0 50 0 0 0 2.0 · · 3 Convert to numeric
Previous 20 Columns
Next 20 Columns
CHUNK COMPRESSION SUMMARY
FRAME DISTRIBUTION SUMMARY
localhost:54321/flow/index.html#
Connections: 0
H2O

H2O FLOW Flow Cell Data Model Score Admin Help

iris_demo

CHUNK COMPRESSION SUMMARY

FRAME DISTRIBUTION SUMMARY

```
splitFrame
```

21ms

80% & 20% Split

Split Frame

Frame: Key_Frame__iris.hex

Splits:	Ratio	Key
	0.8	train
	0.20	test

Add a new split

Seed: 1234

Create

CS

```
splitFrame "Key_Frame__iris.hex", [0.8], ["train","test"], 1234
```

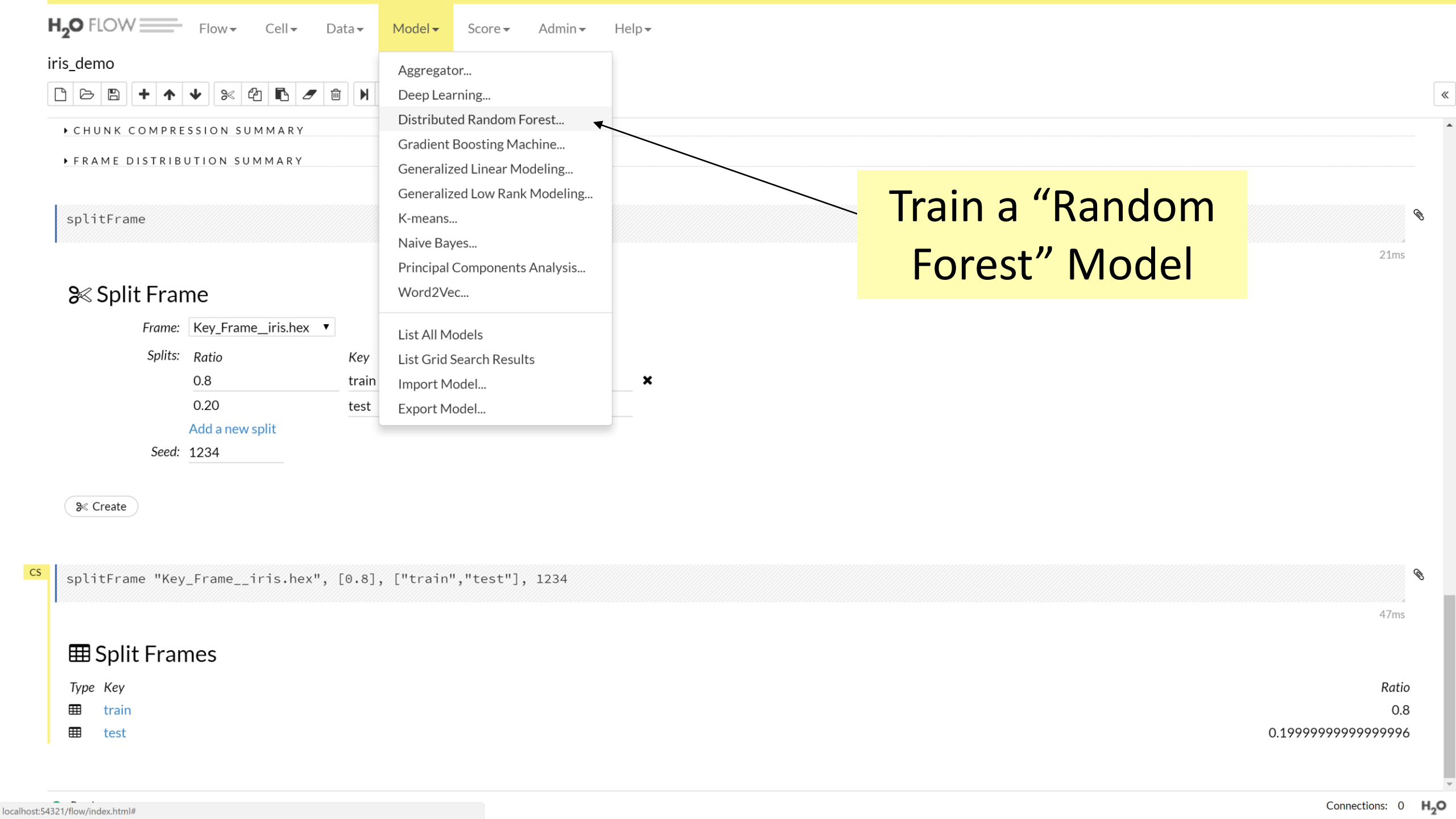
47ms

Split Frames

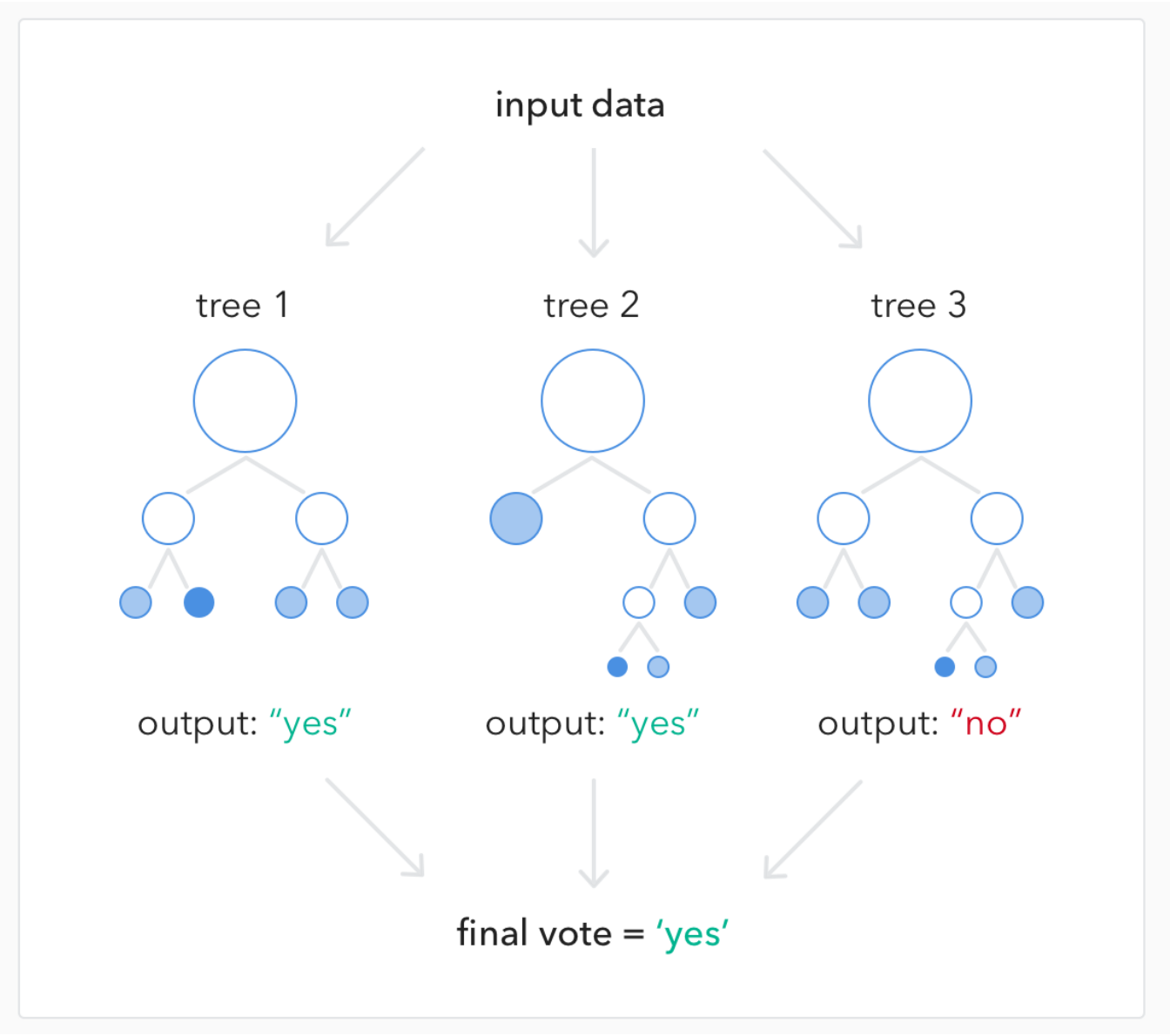
Type	Key	Ratio
	train	0.8
	test	0.19999999999999996

Ready

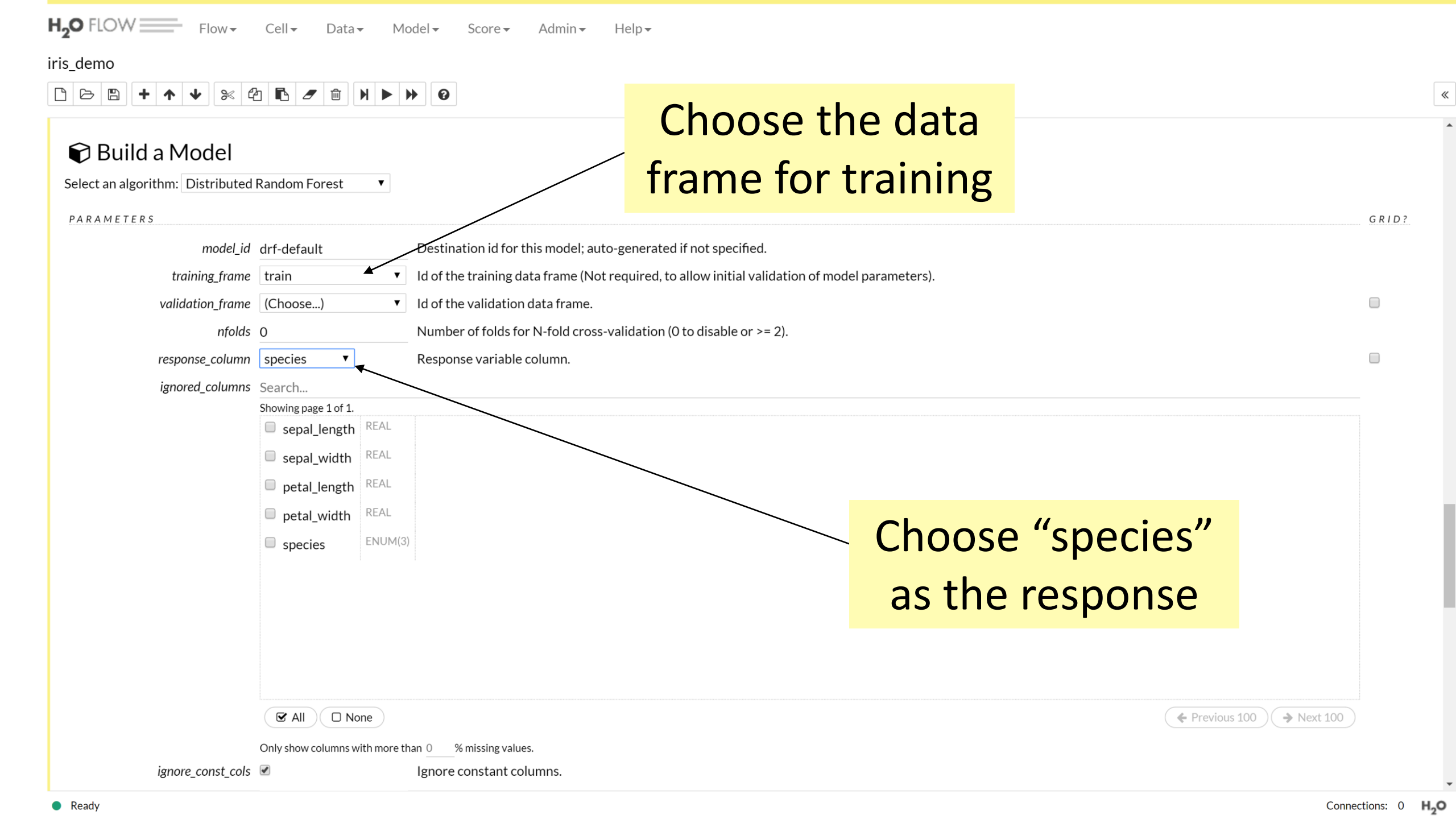
Connections: 0 H2O


H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
iris_demo
Aggregator...
Deep Learning...
Distributed Random Forest...
Gradient Boosting Machine...
Generalized Linear Modeling...
Generalized Low Rank Modeling...
K-means...
Naive Bayes...
Principal Components Analysis...
Word2Vec...
List All Models
List Grid Search Results
Import Model...
Export Model...
CHUNK COMPRESSION SUMMARY
FRAME DISTRIBUTION SUMMARY
splitFrame
21ms
Train a “Random Forest” Model
Split Frame
Frame: Key_Frame__iris.hex
Splits: Ratio Key
0.8 train
0.20 test
Add a new split
Seed: 1234
Create
CS
splitFrame "Key_Frame__iris.hex", [0.8], ["train","test"], 1234
47ms
Split Frames
Type Key Ratio
train 0.8
test 0.19999999999999996
Connections: 0

Random Forest



Random Forest (Source: https://cdn.auth0.com/blog/machine-learning-1/random_forest.png)


H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
iris_demo
Build a Model
Select an algorithm: Distributed Random Forest
PARAMETERS
GRID?
model_id drf-default Destination id for this model; auto-generated if not specified.
training_frame train Id of the training data frame (Not required, to allow initial validation of model parameters).
validation_frame (Choose...) Id of the validation data frame.
nfolds 0 Number of folds for N-fold cross-validation (0 to disable or >= 2).
response_column species Response variable column.
ignored_columns Search...
Showing page 1 of 1.
sepal_length REAL
sepal_width REAL
petal_length REAL
petal_width REAL
species ENUM(3)
All
None
Previous 100
Next 100
Only show columns with more than 0 % missing values.
ignore_const_cols Ignore constant columns.
Choose the data frame for training
Choose “species” as the response
Ready
Connections: 0

H2O FLOW Flow Cell Data Model Score Admin Help

iris_demo

Parameters (leave them as default for now)

ntrees	50	Number of trees.
max_depth	20	Maximum tree depth.
min_rows	1	Fewest allowed (weighted) observations in a leaf.
nbins	20	For numerical columns (real/int), build a histogram of (at least) this many bins, th
seed	-1	Seed for pseudo random number generator (if applicable)
mtries	-1	Number of variables randomly sampled as candidates at each split. If set to -1, defaults to sqrt{p} for classification and p/3 for regression (where p is the # of predictors
sample_rate	0.6320000290870667	Row sample rate per tree (from 0.0 to 1.0)

ADVANCED GRID?

score_each_iteration	☐	Whether to score during each iteration of model training.
score_tree_interval	0	Score the model after every so many trees. Disabled if set to 0.
fold_column	(Choose...)	Column with cross-validation fold index assignment per observation.
offset_column	(Choose...)	Offset column. This will be added to the combination of columns before applying the link function.
weights_column	(Choose...)	Column with observation weights. Giving some observation a weight of zero is equivalent to excluding it from the dataset; giving an observation a relative weight of 2 is equivalent to repeating that row twice. Negative weights are not allowed.
balance_classes	☐	Balance training data class counts via over/under-sampling (for imbalanced data).
max_confusion_matrix_size	20	[Deprecated] Maximum size (# classes) for confusion matrices to be printed in the Logs
max_hit_ratio_k	0	Max. number (top K) of predictions to use for hit ratio computation (for multi-class only, 0 to disable)
nbins_top_level	1024	For numerical columns (real/int), build a histogram of (at most) this many bins at the root level, then decrease by factor of two per level
nbins_cats	1024	For categorical columns (factors), build a histogram of this many bins, then split at the best point. Higher values can lead to more overfitting.
r2_stopping	1.7976931348623157e+	r2_stopping is no longer supported and will be ignored if set - please use stopping_rounds, stopping_metric and stopping_tolerance instead. Previous version of H2O would stop making trees when the R^2 metric equals or exceeds this
stopping_rounds	0	Early stopping based on convergence of stopping_metric. Stop if simple moving average of length k of the stopping_metric does not improve for k:=stopping_rounds scoring events (0 to disable)
stopping_metric	AUTO	Metric to use for early stopping (AUTO: logloss for classification, deviance for regression)
stopping_tolerance	0.001	Relative tolerance for metric-based stopping criterion (stop if relative improvement is not at least this much)

Ready

Connections: 0 H2O

iris_demo

max_confusion_matrix_size	20	[Deprecated] Maximum size (# classes) for confusion matrices to be printed in the Logs
max_hit_ratio_k	0	Max. number (top K) of predictions to use for hit ratio computation (for multi-class only, 0 to disable)
nbins_top_level	1024	For numerical columns (real/int), build a histogram of (at most) this many bins at the root level, then decrease by factor of two per level
nbins_cats	1024	For categorical columns (factors), build a histogram of this many bins, then split at the best point. Higher values can lead to more overfitting.
r2_stopping	1.7976931348623157e+	r2_stopping is no longer supported and will be ignored if set - please use stopping_rounds, stopping_metric and stopping_tolerance instead. Previous version of H2O would stop making trees when the R^2 metric equals or exceeds this
stopping_rounds	0	Early stopping based on convergence of stopping_metric. Stop if simple moving average of length k of the stopping_metric does not improve for k:=stopping_rounds scoring events (0 to disable)
stopping_metric	AUTO	Metric to use for early stopping (AUTO: logloss for classification, deviance for regression)
stopping_tolerance	0.001	Relative tolerance for metric-based stopping criterion (stop if relative improvement is not at least this much)
max_runtime_secs	0	Maximum allowed runtime in seconds for model training. Use 0 to disable.
checkpoint		Model checkpoint to resume training with.
col_sample_rate_per_tree	1	Column sample rate per tree (from 0.0 to 1.0)
min_split_improvement	0.00001	Minimum relative improvement in squared error reduction for a split to happen
histogram_type	AUTO	What type of histogram to use for finding optimal split points
categorical_encoding	AUTO	Encoding scheme for categorical features

EXPERT GRID?

build_tree_one_node	☐	Run on one node only; no network overhead but fewer cpus used. Suitable for small datasets.
sample_rate_per_class		Row sample rate per tree per class (from 0.0 to 1.0)
binomial_double_trees	☐	For binary classification: Build 2x as many trees (one per class) - can lead to higher accuracy.
col_sample_rate_change_per_level	1	Relative change of the column sampling rate for every level (from 0.0 to 2.0)

Build Model

Click “Build Model”

iris_demo

build_tree_one_node	☐	Run on one node only; no network overhead but fewer cpus used. Suitable for small datasets.
sample_rate_per_class		Row sample rate per tree per class (from 0.0 to 1.0)
binomial_double_trees	☐	For binary classification: Build 2x as many trees (one per class) - can lead to higher accuracy.
col_sample_rate_change_per_level	1	Relative change of the column sampling rate for every level (from 0.0 to 2.0)

Build Model

CS

```
buildModel 'drf', {"model_id":"drf-default","training_frame":"train","nfolds":0,"response_column":"species","ignored_columns":
[],"ignore_const_cols":true,"ntrees":50,"max_depth":20,"min_rows":1,"nbins":20,"seed":-1,"mtries":-1,"sample_rate":0.6320000290870667,"score_each_iteration":false,"sc
ore_tree_interval":0,"balance_classes":false,"max_confusion_matrix_size":20,"max_hit_ratio_k":0,"nbins_top_level":1024,"nbins_cats":1024,"r2_stopping":1.7976931348623
157e+308,"stopping_rounds":0,"stopping_metric":"AUTO","stopping_tolerance":0.001,"max_runtime_secs":0,"checkpoint":"","col_sample_rate_per_tree":1,"min_split_improvem
ent":0.00001,"histogram_type":"AUTO","categorical_encoding":"AUTO","build_tree_one_node":false,"sample_rate_per_class":
[],"binomial_double_trees":false,"col_sample_rate_change_per_level":1}
```

1.1s

Job

Run Time	00:00:00.452
Remaining Time	00:00:00.0
Type	Model
Key	drf-default
Description	DRF
Status	DONE
Progress	100%

Done.

Actions View

H2O FLOW Flow Cell Data Model Score Admin Help

iris_demo

Model

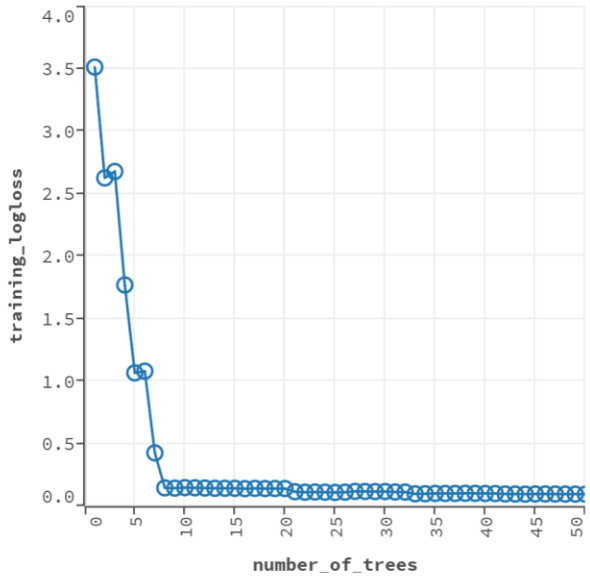
Model ID: drf-default

Algorithm: Distributed Random Forest

Actions: Refresh | Predict... | Download POJO | Download Model Deployment Package | Export | Inspect | Delete

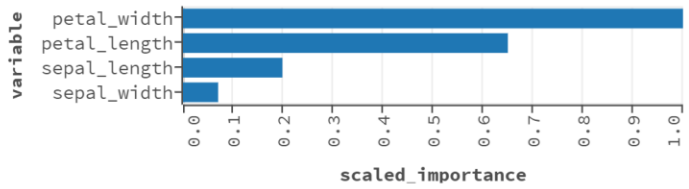
MODEL PARAMETERS

SCORING HISTORY - LOGLOSS



Logloss decreasing with no. of trees = Model getting better

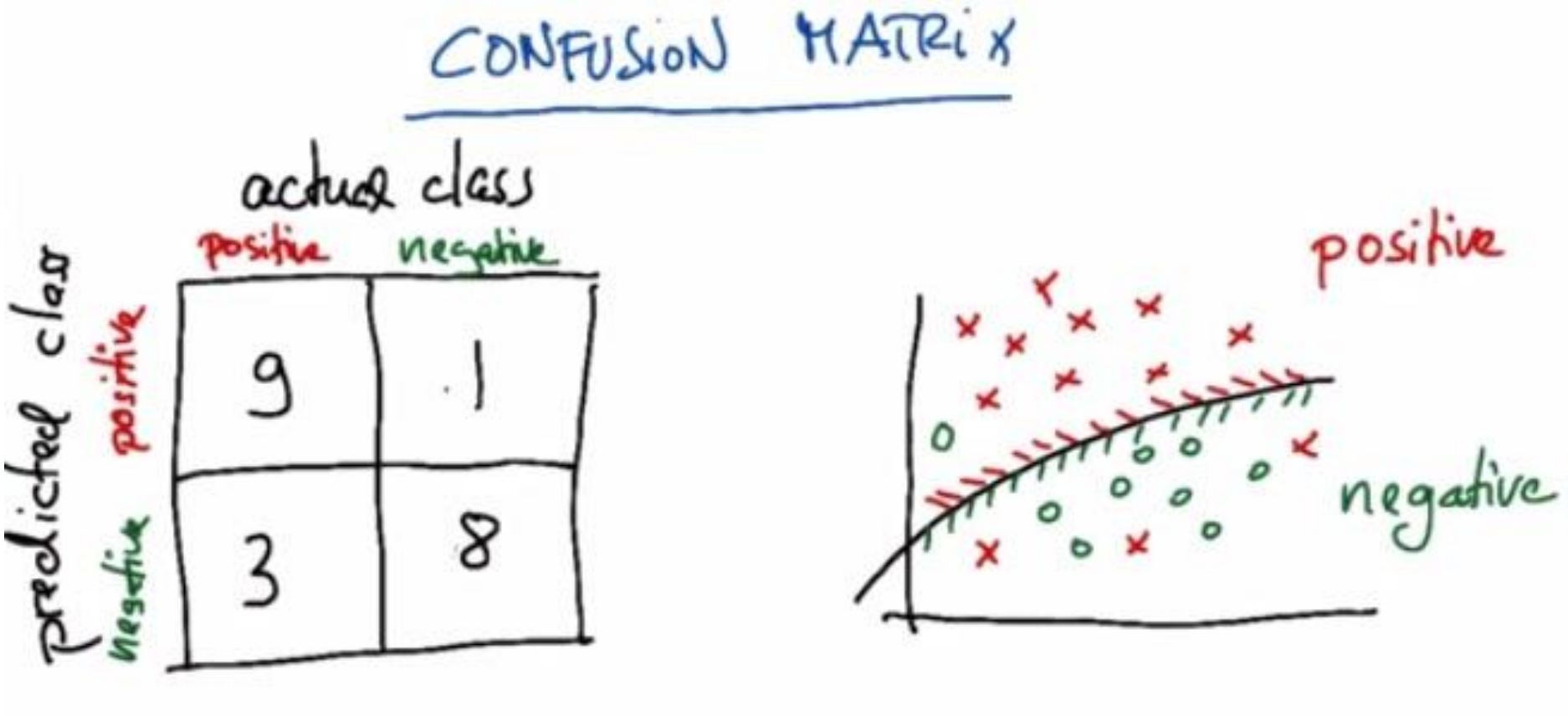
VARIABLE IMPORTANCES



Ready

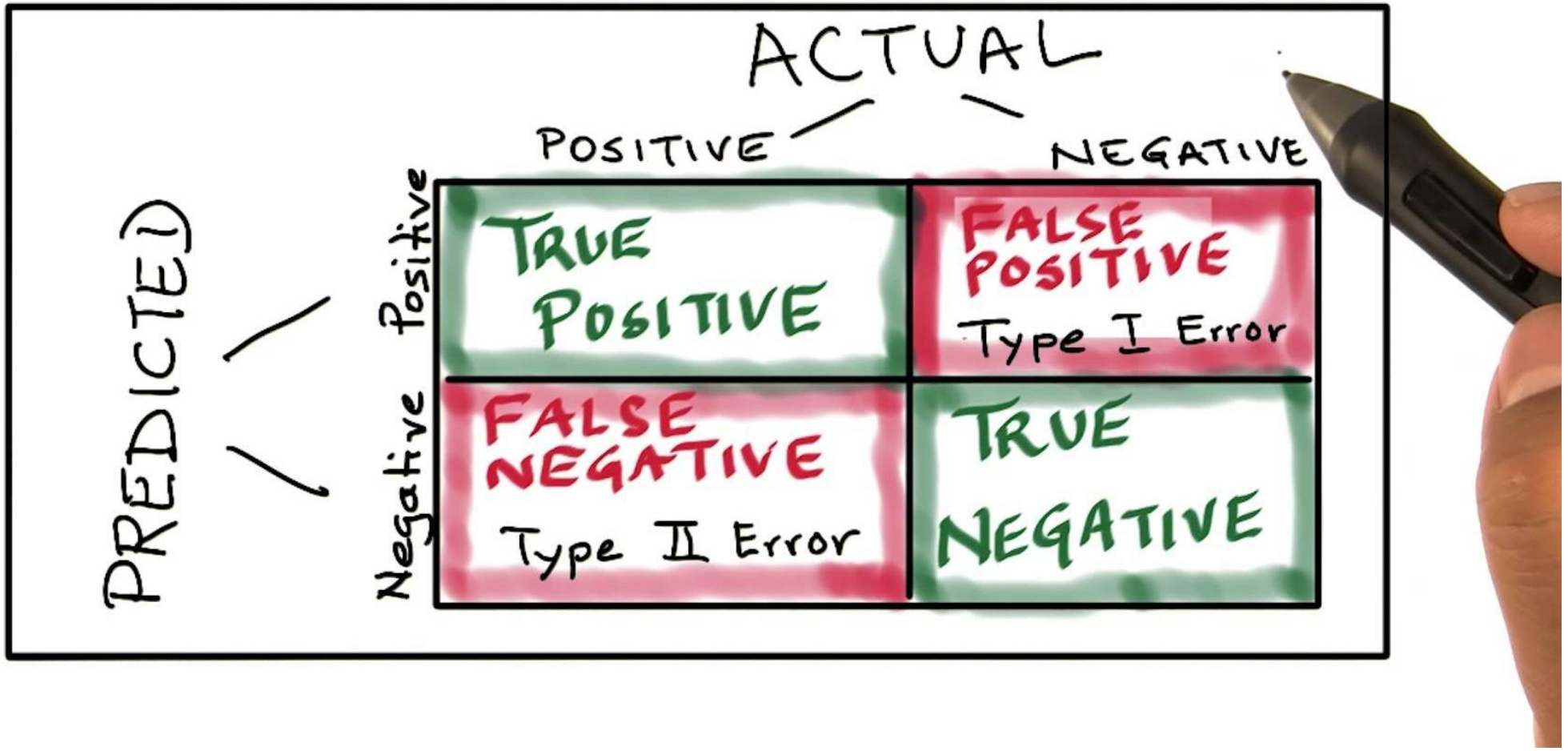
Connections: 0 H2O

Classification Performance – Confusion Matrix



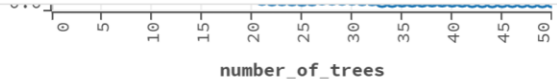
H2O.ai

Confusion Matrix

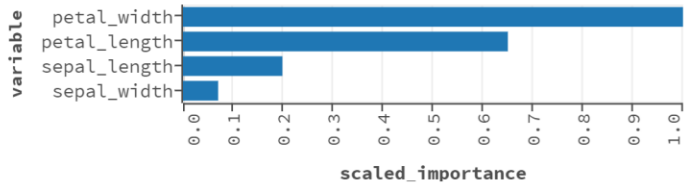


H2O.ai

iris_demo



VARIABLE IMPORTANCES



TRAINING METRICS - CONFUSION MATRIX VERTICAL: ACTUAL; ACROSS: PREDICTED

	setosa	versicolor	virginica	Error	Rate
setosa	36	0	0	**0**	**0 / 36**
versicolor	0	38	2	**0.0500**	**2 / 40**
virginica	0	3	44	**0.0638**	**3 / 47**
Total	**36**	**41**	**46**	**0.0407**	**5 / 123**

OUTPUT

OUTPUT - MODEL SUMMARY

OUTPUT - SCORING HISTORY

OUTPUT - TRAINING_METRICS

OUTPUT - TRAINING_METRICS - TOP-3 HIT RATIOS

OUTPUT - VARIABLE IMPORTANCES

PREVIEW POJO

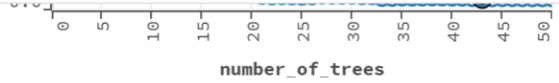
</> Preview POJO

H2O FLOW Flow Cell Data Model Score Admin Help

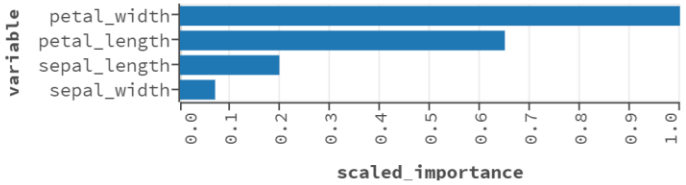
iris_demo

Predict...
Partial Dependence Plots...
List All Predictions

Make Predictions



VARIABLE IMPORTANCES



TRAINING METRICS - CONFUSION MATRIX VERTICAL: ACTUAL; ACROSS: PREDICTED

	setosa	versicolor	virginica	Error	Rate
setosa	36	0	0	**0**	**0 / 36**
versicolor	0	38	2	**0.0500**	**2 / 40**
virginica	0	3	44	**0.0638**	**3 / 47**
Total	**36**	**41**	**46**	**0.0407**	**5 / 123**

OUTPUT

OUTPUT - MODEL SUMMARY

OUTPUT - SCORING HISTORY

OUTPUT - TRAINING_METRICS

OUTPUT - TRAINING_METRICS - TOP-3 HIT RATIOS

OUTPUT - VARIABLE IMPORTANCES

PREVIEW POJO

</> Preview POJO

H2O FLOW Flow Cell Data Model Score Admin Help

iris_demo

	setosa	versicolor	virginica	Error	Rate
setosa	36	0	0	0	0 / 36
versicolor	0	38	2	0.0500	2 / 40
virginica	0	3	44	0.0638	3 / 47
Total	36	41	46	0.0407	5 / 123

▸ OUTPUT

▸ OUTPUT - MODEL SUMMARY

▸ OUTPUT - SCORING HISTORY

▸ OUTPUT - TRAINING_METRICS

▸ OUTPUT - TRAINING_METRICS - TOP-3 HIT RATIOS

▸ OUTPUT - VARIABLE IMPORTANCES

▾ PREVIEW POJO

</> Preview POJO

CS

```
predict
```

71ms

Predict

Name: prediction-drf

Model: drf-default

Frame: test

Actions: Predict

Select “Test” Frame

Ready

Connections: 0 H2O

iris_demo

CS

```
predict model: "drf-default", frame: "test", predictions_frame: "prediction-drf"
```

116ms

Prediction

Actions: Inspect

PREDICTION

model	drf-default
model_checksum	-5212176637003146240
frame	test
frame_checksum	-3534762742338534912
description	·
model_category	Multinomial
scoring_time	1491775918327
predictions	prediction-drf
MSE	0.034303
RMSE	0.185212
nobs	27
r2	0.926014
logloss	0.097069
mean_per_class_error	0.144444

Combine predictions with frame

PREDICTION - TOP-3 HIT RATIOS

k	*hit_ratio*
1	0.9259
2	1.0
3	1.0

PREDICTION - CM

iris_demo

model	drf-default
model_checksum	-5212176637003146240
frame	test
frame_checksum	-3534762742338534912
description	·
model_category	Multinomial
scoring_time	1491775918327
predictions	prediction-drf
MSE	0.034303
RMSE	0.185212
nobs	27
r2	0.926014
logloss	0.097069
mean_per_class_error	0.144444

Combine predictions with frame

▾ PREDICTION - TOP-3 HIT RATIOS

k	hit_ratio
1	0.9259
2	1.0
3	1.0

▾ PREDICTION - CM

▾ PREDICTION - CM - CONFUSION MATRIX

setosa	versicolor	virginica	Error	Rate
14	0	0	0	0 / 14
0	9	1	0.1000	1 / 10
0	1	2	0.3333	1 / 3
14	10	3	0.0741	2 / 27

Confusion Matrix for Predictions

H2O FLOW Flow Cell Data Model Score Admin Help

iris_demo

CS

```
getFrameData "prediction-drf"
```

48ms

prediction-drf

DATA

Previous 20 Columns Next 20 Columns

Row	predict	setosa	versicolor	virginica
1	setosa	0.9986	0	0.0014
2	setosa	0.9986	0	0.0014
3	setosa	0.9595	0.0392	0.0013
4	setosa	0.9986	0	0.0014
5	setosa	0.9587	0.0399	0.0014
6	setosa	0.9986	0	0.0014
7	setosa	0.9986	0	0.0014
8	setosa	0.9986	0	0.0014
9	setosa	0.9986	0	0.0014
10	setosa	0.9986	0	0.0014
11	setosa	0.9986	0	0.0014
12	setosa	0.9986	0	0.0014
13	setosa	0.9791	0.0196	0.0013
14	setosa	0.9986	0	0.0014
15	versicolor	0	0.9773	0.0227
16	versicolor	0	0.9986	0.0014
17	versicolor	0	0.8151	0.1849
18	versicolor	0	0.9985	0.0015
19	versicolor	0	0.9986	0.0014
20	versicolor	0	0.9986	0.0014
21	virginica	0	0.2946	0.7054
22	versicolor	0	0.9986	0.0014
23	versicolor	0.0188	0.9422	0.0390
24	versicolor	0	0.9985	0.0015

Outputs with predicted labels and probabilities

Ready

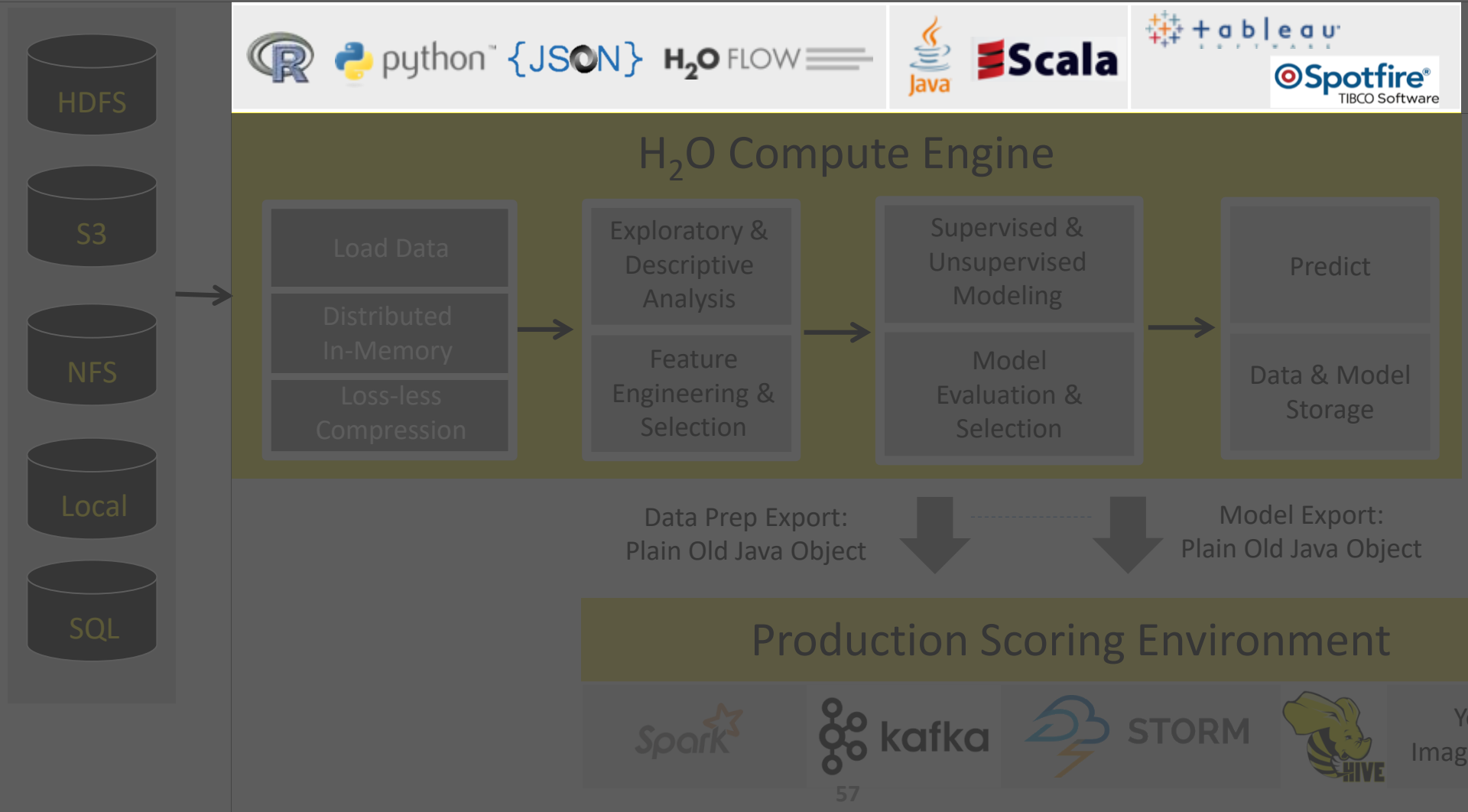
Connections: 0 H2O

H_2O with Python

PyData Amsterdam Tutorial

High Level Architecture

Multiple Interfaces



Slides and Code Examples:

bit.ly/joe_h2o_tutorials

Distributed Random Forest

```
# Build a Distributed Random Forest (DRF) model with default settings

# Import the function for DRF
from h2o.estimators.random_forest import H2ORandomForestEstimator

# Set up DRF for regression
# Add a seed for reproducibility
drf_default = H2ORandomForestEstimator(model_id = 'drf_default', seed = 1234)

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

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)
```

Thanks!

- Organizers

<RE/START>

- 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)

H2O.ai